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

R^2-LIO: Real-time and Robust LiDAR-Inertial Odometry in Dynamic Environments

Changjun Gu[1], Ziyi Huang[1], Gan Sun[3], Jiahua Dong[4], Jiaxu Leng[2] and Xinbo Gao[2*]

***Abstract*—LiDAR-Inertial Odometry (LIO) is crucial for robot navigation and autonomous exploration. Most existing methods rely on the assumption of a static environment, indiscriminately using all LiDAR measurements for localization. However, LiDAR data acquired in urban scenes often contain dynamic objects such as vehicles and pedestrians, which can adversely affect localization accuracy—particularly when using solid-state LiDAR with a relatively narrow field of view. To address this issue, we propose a novel real-time and robust solid-state LiDAR-Inertial Odometry (R^2-LIO) framework that removes dynamic objects to improve the localization accuracy and robustness. Specifically, we design a dynamic point removal mechanism based on voxel state changes, which removes dynamic points and preserves most static points to effectively reduce interference from dynamic objects. In addition, we introduce a line search mechanism into the Error State Iterated Kalman Filter (ESIKF) to improve the localization accuracy. Experimental results on the challenging YULAN and HeLiPR datasets show that R^2-LIO surpasses existing methods, verifying its effectiveness in improving the localization accuracy and robustness.**

***Index Terms*—Simultaneous Localization and Mapping (SLAM), LiDAR Moving Object Removal, State Estimation.**

I. Introduction

Simultaneous Localization and Mapping (SLAM) is crucial for robot navigation and autonomous exploration [1]. In SLAM systems, LiDAR and inertial measurement unit (IMU) are often fused to achieve robust localization, where LiDAR provides geometric constraints and the IMU offers motion priors for accurate and efficient state estimation. However, traditional mechanical LiDAR is costly and bulky, which hinders large-scale deployment. Recently, solid-state LiDAR has emerged as a lightweight, low-cost, and high-performance alternative. Therefore, this paper focuses on solid-state LiDAR–inertial localization.

[1]Changjun Gu and Ziyi Huang are with the School of Automation, Chongqing University of Posts and Telecommunications, Chongqing, China.
[2]Xinbo Gao and Jiaxu Leng are with the School of Computer Science and Technology, Chongqing University of Posts and Telecommunications, Chongqing, China.
[3]Gan Sun is with the College of Automation Science and Engineering, South China University of Technology, Guangzhou, China.
[4]Jiahua Dong is with the Mohamed bin Zayed University of Artificial Intelligence, Abu Dhabi, UAE.
*The corresponding author is Prof. Xinbo Gao (e-mail: gaoxb@cqupt.edu.cn).
This work was supported by the National Natural Science Foundation of China under Grants No. 62303083, U23A20318, 62221005 and 62441601, in part by the Science and Technology Research Program of Chongqing Municipal Education Commission (Grant No. KJQN202300626), and in part by the Science and Technology Innovation Key R&D Program of Chongqing (Grant No. CSTB2023TIAD-STX0016).

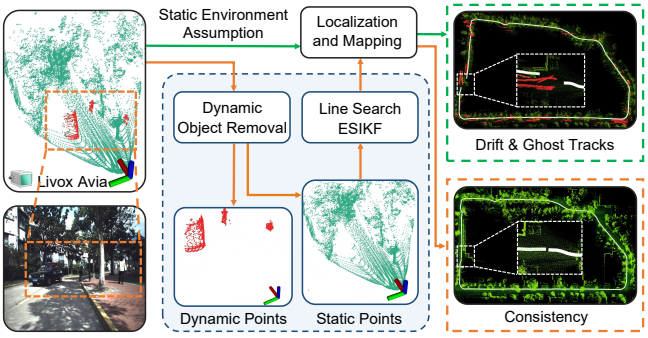

Fig. 1. Comparison between baseline and R^2-LIO. Existing solid-state LiDAR–inertial localization methods typically assume a static environment and directly rely on raw LiDAR points and IMU measurements for localization. In contrast, R^2-LIO removes dynamic objects and integrates a line search ESIKF to enhance localization accuracy. Green lines illustrate existing methods, while orange lines represent R^2-LIO.

As shown in Fig. 1, most existing solid-state LiDAR–inertial localization methods assume a static environment and tightly fuse raw LiDAR points with IMU measurements. For instance, FAST-LIO [2] and FAST-LIO2 [3] implement this strategy through an iterated Kalman filter framework and organize data efficiently with an incremental k-d tree. VoxelMap [4] and VoxelMap++ [5] further enhance LiDAR–inertial odometry by using voxel-based maps with planar features for efficient and robust pose estimation. Recent methods have started incorporating visual information to further improve localization accuracy [6]. For example, FAST-LIVO [7] and FAST-LIVO2 [8] consist of LIO and VIO modules: the LIO module reconstructs 3D structures, while the VIO module provides color information [9]. Both modules contribute to state estimation.

Despite advances in solid-state LiDAR–inertial localization, existing methods still face significant challenges, as LiDAR data collected in urban environments often contain dynamic objects [10], [11]. There are two main issues: (1) **Dynamic Objects Degrade Localization Accuracy:** LiDAR data in urban environments often contain dynamic objects such as vehicles and pedestrians, which introduce false occupancy in maps and reduce localization accuracy. The issue is more severe for solid-state LiDAR, whose limited field of view causes moving objects to occupy a larger portion of each scan, leading to unstable pose estimation. (2) **Reduced Constraints due to Limited Static Feature Points:** After removing dynamic objects, the limited number of remaining

static points can result in insufficient constraints for state optimization, exacerbating ill-conditioning and causing convergence failure. To overcome the above challenges in solid-state LiDAR–inertial localization (as shown in Fig. 1), we propose a Real-time and Robust solid-state LiDAR-Inertial Odometry (R^2-LIO), a novel method that removes dynamic objects to improve localization accuracy and robustness. Specifically, we introduce a real-time and efficient dynamic point removal module in the front end, which leverages IMU-predicted states and motion-compensated point clouds to maintain a local voxel occupancy probability map. Dynamic points are effectively removed based on temporal changes in voxel states, providing more accurate observations for back-end pose optimization. In addition, to improve the efficiency of state optimization caused by a limited number of static feature points, we incorporate a line search mechanism into the Error State Iterated Kalman Filter (ESIKF), using the Armijo condition to adaptively adjust the state update step size, which enhances optimization efficiency and improves localization accuracy. We evaluate our method on the challenging YULAN [12] and HeLiPR datasets. Experimental results show that our proposed method consistently outperforms state-of-the-art (SOTA) methods. In summary, our contributions are as follows:

- We propose R^2-LIO, a novel framework that removes dynamic objects to enhance localization accuracy and robustness. To the best of our knowledge, this is an earlierly attempt to explicitly account for dynamic objects in solid-state LiDAR–inertial localization.
- We introduce a line search mechanism into the error state iterated Kalman filter to improve the optimization efficiency and localization accuracy.
- Experimental results on multiple public datasets with dynamic objects show that R^2-LIO surpasses existing LIO methods in localization accuracy and robustness, demonstrating the effectiveness of our proposed method.

II. Related Works

In this section, we briefly review related work on LiDAR-inertial odometry and LiDAR-inertial odometry for dynamic environments.

A. LiDAR-Inertial Odometry

Existing methods are typically classified into two categories: loosely coupled methods and tightly coupled methods. Loosely coupled methods utilize IMU measurements as motion priors for scan registration. For example, LOAM-Livox [13] adapts LOAM [14] to Livox sensors by redesigning feature extraction and mapping to handle non-repetitive scans. Tightly coupled methods, on the other hand, deeply fuse LiDAR and IMU data, significantly enhancing state estimation performance. For example, FAST-LIO [2] tightly couples LiDAR and IMU via ESIKF with efficient Kalman gain computation, while FAST-LIO2 [3] removes explicit feature extraction and employs an incremental kd-tree for faster updates. To improve efficiency, Faster-LIO [15] proposes a data structure based on incremental voxels, significantly accelerating computation while maintaining accuracy. Voxel-Map [4] introduces a probabilistic adaptive voxel mapping method for LIO, enhancing registration accuracy through a probabilistic representation of the environment. To improve localization accuracy in complex or degenerate environments (e.g., corridors, tunnels), visual information is often incorporated to capture texture features. R2LIVE [16] and R3LIVE [17] introduce a parallel design of LIO and VIO modules, where the LIO module provides geometric structure information to assist the VIO module, and visual landmarks are leveraged in the back end for graph-based optimization.

Fast-LIVO [7] and Fast-LIVO2 [8] combine LiDAR, camera, and IMU measurements within a single ESIKF, which can be updated using both LiDAR and visual observations. SR-LVIO [18] employs sweep reconstruction to improve state estimation efficiency, thereby enhancing localization accuracy. These methods achieve accurate localization for autonomous driving and robotic applications. However, they typically assume a static environment, meaning that all objects in the scene are stationary. As a result, their accuracy can degrade in urban scenarios containing dynamic objects such as vehicles and pedestrians.

B. LiDAR-Inertial Odometry for Dynamic Environments

Removing dynamic objects from LiDAR scans effectively improves both localization accuracy and robustness. Existing methods are typically divided into three categories: label- or consistency-based methods, occupancy-map-based methods, and semantic segmentation methods. Label- or consistency-based checking methods typically identify dynamic objects by detecting points in the current scan that differ from those in previous scans. For example, STATIC-LIO [19] employs terrain-assisted dynamic point removal to enhance localization accuracy, while RF-LIO [20] utilizes multi-resolution range images to identify and eliminate dynamic objects, thereby improving both localization accuracy and robustness. Occupancy-map-based methods use grids to represent the environment, enabling static and dynamic objects to be distinguished. For example, DRR-LIO [21] projects the point cloud onto a 2D grid and leverages height statistics to differentiate dynamic from static points, removing dynamic points from the current scan to improve LiDAR–inertial localization accuracy. Semantic segmentation methods leverage semantic information to distinguish between static and dynamic objects, allowing the removal of dynamic points from the map and improving localization performance. For example, SuMa++ [22] removes dynamic objects using RangeNet++ [23] segmentation, thereby enhancing both localization and map accuracy.

However, most existing methods focus on mechanically spinning omnidirectional LiDAR–inertial localization in dynamic environments, and lack efficient solutions for solid-state LiDAR–inertial localization, which is characterized by irregular scanning patterns and a relatively narrow field of view.

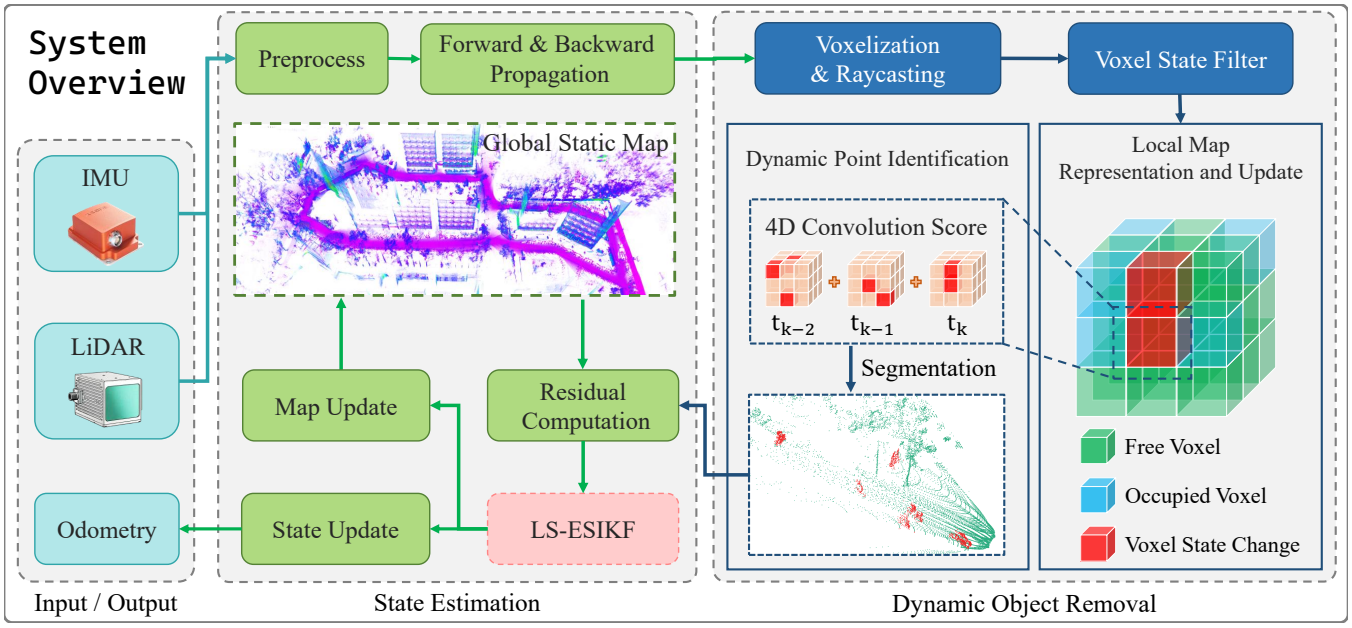

Fig. 2. Overview of R^2-LIO. LiDAR and IMU measurements are first processed through preprocessing and forward–backward propagation. The Dynamic Object Removal (DOR) module then filters dynamic points from the current scan. Finally, state estimation is performed using the line-search-enhanced Error-State Iterated Kalman Filter (LS-ESIKF).

In this work, we address the aforementioned limitations with a novel dynamic point removal method integrated into state optimization. The key idea is to remove dynamic points while preserving most static points, thereby reducing interference from moving objects.

III. METHOD

An overview of the R^2-LIO architecture is shown in Fig. 2. The framework integrates two key modules: Dynamic Object Removal (DOR) and Line-Search-Enhanced ESIKF (LS–ESIKF). DOR module tracks voxel state changes over the past few seconds of LiDAR observations to extract and discard dynamic points from the current LiDAR scan, while the remaining static points are used for LiDAR observation updates. LS–ESIKF introduces a line search mechanism into the conventional ESIKF to improve the stability and convergence of state updates, thereby enhancing overall localization accuracy.

A. Problem Definition

LiDAR-Inertial SLAM estimates the robot trajectory $\mathbf{x}_{1:t}$ and the environment map $\mathcal{M}$ through the integration of LiDAR scans and IMU measurements. In our formulation, we explicitly remove dynamic points from LiDAR observations before state estimation, leading to the following optimization problem:

$$\hat{\mathbf{x}}_{1:t}, \hat{\mathcal{M}} = \arg\max_{\mathbf{x}_{1:t}, \mathcal{M}} p\Big(\mathbf{x}_{1:t}, \mathcal{M} \Big| \bigcup_{\tau \in [1,t]} ({}^{L}\mathcal{P}_{\tau} \backslash {}^{L}\mathcal{P}_{\tau}^{\text{dyn}}),\, \mathcal{U}_{1:t}\Big), \tag{1}$$

where $\hat{\mathbf{x}}_{1:t}$ and $\hat{\mathcal{M}}$ denote the estimated robot trajectory and environment map, respectively; τ is the current time step; ${}^{L}\mathcal{P}_{\tau}$ represents the LiDAR observation at time step τ containing dynamic points; and ${}^{L}\mathcal{P}_{\tau}^{\text{dyn}}$ represents the set of dynamic points filtered out from ${}^{L}\mathcal{P}_{\tau}$ at time step τ.

B. Dynamic Object Removal (DOR)

DOR module consists of three main components: voxelization and ray casting, local dynamic map representation and update, and dynamic point identification. First, voxelization discretizes the 3D scene into voxels, while ray casting emits rays to determine which voxels are occupied. Next, the local dynamic map is updated to facilitate dynamic point detection. Finally, the dynamic points ${}^{L}\mathcal{P}^{\text{dyn}}$ are identified and removed from the current LiDAR scan ${}^{L}\mathcal{P}_{\tau}$.

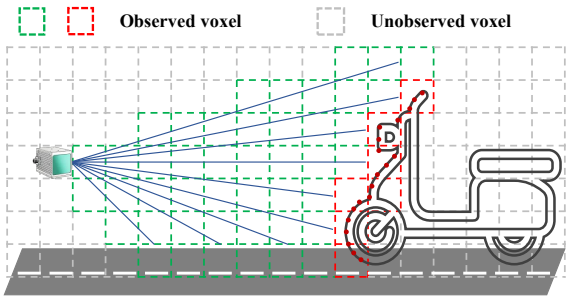

Fig. 3. Illustration of Voxelization and Raycasting.

1) Voxelization and Ray Casting: At time step τ, following FAST-LIO2, we employ the preprocessing and forward propagation steps to estimate the LiDAR pose $({}^{G}R_{L}, {}^{G}t_{L})$ in the global frame G, and apply backward propagation to obtain the undistorted LiDAR point cloud ${}^{L}\mathcal{P}_{\tau} = {}^{L}p_i \in \mathbb{R}^3$. The point cloud is subsequently projected into the global frame according to:

$${}^{G}\mathcal{P}_{\tau} = \left\{ {}^{G}R_{L}\, {}^{L}\mathcal{P}_{\tau} + {}^{G}t_{L} \right\}. \tag{2}$$

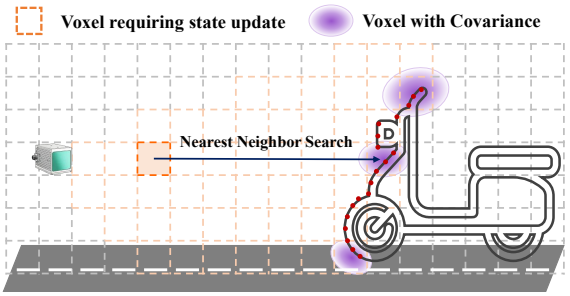

Fig. 4. Illustration of state update for observed voxels.

Next, the global point cloud is discretized into voxels, with each point assigned the coordinates of its corresponding voxel, denoted as $v({}^{G}p_i)$. At time step τ, the LiDAR frame is represented by the set of occupied voxels:

$$\mathcal{V}_\tau = \left\{ v({}^{G}p_i) \;\middle|\; {}^{G}p_i \in {}^{G}P_\tau \right\}. \tag{3}$$

After voxelization, we perform ray casting to refine the observation space $\mathcal{V}_\tau$, as illustrated in Fig. 3. Specifically, rays are emitted from the LiDAR-origin voxel toward each occupied voxel $\mathcal{V}_\tau^{\text{occ}}$, while all other voxels along the ray path are marked as free $\mathcal{V}_\tau^{\text{free}}$. The refined observation space of the current LiDAR scan is then represented as $\mathcal{V}_\tau^{\text{obs}}$:

$$\mathcal{V}_\tau^{\text{obs}} = \mathcal{V}_\tau^{\text{occ}} \cup \mathcal{V}_\tau^{\text{free}}. \tag{4}$$

2) Local Dynamic Map Representation and Update: The local dynamic map $\mathcal{M}_{loc}$ is the set of voxels observed by the LiDAR over the past 5 seconds, each voxel $v_i \in \mathcal{M}_{loc}$ is represented as:

$$v_i = (\mathbf{s}, t_{\text{obs}}, \Delta, D), \tag{5}$$

where $\mathbf{s} = [p_{\text{occ}}, p_{\text{free}}]^\top$ is the voxel state vector with $p_{\text{occ}} + p_{\text{free}} = 1$. A voxel is considered occupied or free if its probability exceeds 0.99. t_{obs} denotes the most recent observation time; voxels not updated for more than 5 seconds are removed. Δ is the state-change flag, set to true if the voxel changes from free to occupied. D is the dynamic flag; if true, the voxel is considered dynamic.

We update the local dynamic map using the latest $\mathcal{V}_\tau^{\text{obs}}$. For the each voxel $v_i \in \mathcal{V}_\tau^{\text{obs}}$ at time step τ, if v_i does not exist in the map $\mathcal{M}_{loc}$, it is initialized as $v_i = \left([0.5, 0.5]^\top, t_\tau, false, false\right)$ and directly inserted into the map. Otherwise, only update its observation time t_{obs} to τ. State updates are performed using the recursive HMM filter [24], which recursively refines the voxel states as follows:

$$\mathbf{s}_{i,\tau} = \eta_{i,\tau} \underbrace{\begin{bmatrix} \mathcal{L}_{v_i}^o & 0 \\ 0 & 1 - \mathcal{L}_{v_i}^o \end{bmatrix}}_{\mathbf{B}_{i,\tau}} \underbrace{\begin{bmatrix} 1-\epsilon & \epsilon \\ \epsilon & 1-\epsilon \end{bmatrix}}_{\mathbf{A}} \mathbf{s}_{i,\tau-1}, \tag{6}$$

where $\mathbf{A}$ denotes the state transition matrix, with ϵ representing a small probability of state change, reflecting the high likelihood of the voxel remaining in its previous state. $\mathbf{B}_{i,\tau}$ is the conditional observation probability matrix, which updates the voxel state based on the current observation. $\eta_{i,\tau}$ is a normalization factor ensuring that $p_{\text{occ}} + p_{\text{free}} = 1$. The likelihood $\mathcal{L}_{v_i}^o$ is computed based on the Euclidean distance d_i from voxel v_i to the nearest occupied voxel:

$$\mathcal{L}_{v_i}^o = \exp\Big(- \frac{d_i^2}{2\sigma_{d_i}^2} \Big), \tag{7}$$

where σ_i represents the uncertainty of the distance measurement. Specifically, when $d_i = 0$, the likelihood is $\mathcal{L}_{v_i}^o = 1$. To compute the distance d_i and its associated uncertainty $\sigma_{d_i}^2$, as illustrated in Fig. 4, we consider v_i as the voxel to be updated and v_i^{occ} as its nearest occupied voxel, with coordinates q_i and q_o in the global frame, respectively. The distance d_i and the uncertainty $\sigma_{d_i}^2$ are then computed as:

$$d_i = \|\mathbf{q_i} - \mathbf{q_o}\|, \quad \sigma_{d_i}^2 = \mathbf{J}\, \mathbf{\Sigma}_{q_o}\, \mathbf{J}^\top, \tag{8}$$

where $\mathbf{J} = \frac{\partial d_i}{\partial \mathbf{q_o}} = \frac{\mathbf{q_i} - \mathbf{q_o}}{|\mathbf{q_i} - \mathbf{q_o}|}$ is the Jacobian of the distance with respect to the coordinates of the occupied voxel, and $\mathbf{\Sigma}_{q_o}$ denotes the covariance of the occupied voxel in the global frame, which is computed as follows:

$$\mathbf{\Sigma}_{\mathbf{q}_o} = {}^{G}\mathbf{R}_L \mathbf{\Sigma}^{G}\mathbf{R}_L^T + {}^{G}\mathbf{R}_L \left[{}^{L}\mathbf{q}_o\right]_\times \mathbf{\Sigma}_{\mathbf{R}} \left[{}^{L}\mathbf{q}_o\right]_\times^T {}^{G}\mathbf{R}_L^T + \mathbf{\Sigma}_{\mathbf{t}}, \tag{9}$$

where $\mathbf{\Sigma}_{\mathbf{R}}$ and $\mathbf{\Sigma}_{\mathbf{t}}$ denote the uncertainties of the LiDAR rotation and translation, respectively; ${}^{L}\mathbf{q}_o$ represents the coordinates of the occupied voxel in the LiDAR frame, and $\mathbf{\Sigma}$ is the corresponding covariance.

3) Dynamic Point Identification: Following [25], candidate dynamic voxels in the current frame are first identified as those with $\Delta(v_i) =$ true, forming the set $\mathcal{V}$cand. For each $v_i \in \mathcal{V}^{\text{cand}}$, a 4D convolution score $S(v_i)$ is computed (see Algorithm 1, lines 5–11). Voxels with $S(v_i)$ exceeding the predefined dynamic threshold θ_{dyn} are classified as dynamic, forming the dynamic voxel set $\mathcal{V}_{\text{dyn}}$ and setting their dynamic flag $D(v_i)$ to true. Points within the dynamic voxel set $\mathcal{V}_{\text{dyn}}$ are treated as dynamic points and removed from the current scan for LiDAR update.

C. State Estimation With Line Search ESIKF (LS-ESIKF)

The system state at time τ, denoted as $\mathbf{x}_\tau$, is estimated by minimizing a Maximum a Posteriori (MAP) cost function that combines LiDAR point-to-plane constraints and an IMU prior constraint:

$$\min_{\mathbf{x}_\tau} J(\mathbf{x}_\tau) = \sum_{i \in {}^{L}\mathcal{P}^{\text{sta}}} \sum_{j \in M} \left\| \mathbf{r}_{i,j}^{\text{pt2pl}} \right\|_{\mathbf{\Omega}_{i,j}^{\text{pt2pl}}}^2 + \left\| \mathbf{r}_\tau^{\text{prior}} \right\|_{\mathbf{\Omega}_\tau^{\text{prior}}}^2, \tag{10}$$

where $J(\mathbf{x}_\tau)$ is the objective function to be minimized with respect to the state $\mathbf{x}_\tau$.The first term, $\mathbf{r}_{i,j}^{\text{pt2pl}}$, is the point-to-plane residual, which measures the geometric error between the LiDAR static points and the corresponding map points. $\mathbf{\Omega}_{i,j}^{\text{pt2pl}}$ is the information matrix associated with the point-to-plane residual, reflecting the uncertainty along the surface normal direction. The second term, $\mathbf{r}_\tau^{\text{prior}}$, is the prior residual derived from IMU integration. The weighted squared norm is defined as $\|\mathbf{r}\|_{\mathbf{\Omega}}^2 = \mathbf{r}^\top \mathbf{\Omega} \mathbf{r}$.

When the number of LiDAR static points available for point-to-plane constraints is small, the cost function is weakly

constrained along certain directions, which may lead to unreliable gradients or excessively large update steps, resulting in unstable state estimation. To address this issue, we introduce a line search mechanism [26] into the ESIKF. Specifically, at each iteration, the state is updated along the search direction determined by the current gradient, and the step size is evaluated using the Armijo condition to ensure sufficient decrease:

$$\hat{\mathbf{x}}_{\tau,j+1} = \begin{cases} \hat{\mathbf{x}}_{\tau,j} \boxplus \alpha \tilde{\mathbf{x}}_{\tau,j}, \text{if formula (12) holds} \\ \text{otherwise, } \alpha \leftarrow \lambda\alpha \text{ and re-evaluate} \end{cases} \tag{11}$$

The Armijo condition is defined as:

$$J(\hat{\mathbf{x}}_{\tau,j} \boxplus \alpha \tilde{\mathbf{x}}_{\tau,j}) \leq J(\hat{\mathbf{x}}_{\tau,j}) + \alpha c \nabla J(\hat{\mathbf{x}}_{\tau,j})^\top \tilde{\mathbf{x}}_{\tau,j}, \tag{12}$$

where $\nabla J(\hat{\mathbf{x}}_{\tau,j})$ denotes the gradient of the cost function at the current state $\hat{\mathbf{x}}_{\tau,j}$. The step size α is initialized to 1.0 and determined via line search. It is iteratively reduced by a factor $\lambda \in (0, 1.0)$ until the Armijo condition is met or a predefined iteration limit is reached, ensuring that each update yields a decrease in the cost function. By incorporating LS-ESIKF, the optimization remains stable even when the number of static points is limited, while enhancing estimation accuracy and accelerating convergence.

Algorithm 1: R^2-LIO

Input : Posterior state $\mathbf{x}_{\tau-1}$ and covariance $\mathbf{P}_{\tau-1}$; IMU data $\mathcal{I}_{\tau-1:\tau}$, LiDAR scan $\mathcal{P}_\tau$.
Output: Posterior state $\mathbf{x}_\tau$ and covariance $\mathbf{P}_\tau$.
Forward propagation to predict $\hat{\mathbf{x}}_\tau$ and $\hat{\mathbf{P}}_\tau$;
Backward propagation for $\mathcal{P}_\tau$ motion compensation;
// Dynamic Object Removal
$\mathcal{V}_\tau^{\text{occ}} \leftarrow$ voxelize($^L\mathcal{P}_\tau$), Ray-cast to obtain $\mathcal{V}_\tau^{\text{free}}$;
Update observed voxels $\mathcal{V}_\tau^{\text{obs}} \in \mathcal{M}_{\text{loc}}$ using (6)
for *each* $v_i \in \mathcal{V}_\tau^{obs}$ **do**
  for $t = \tau - 2$ *to* τ **do**
    for *each* $v_j \in \mathcal{N}(v_i)$ **do**
      if v_j*: free* $\rightarrow$ *occupied at* t **then**
        $S(v_i) \leftarrow S(v_i) + 1$
      else
        $S(v_i) \leftarrow S(v_i) - 1$
Remove LiDAR points in v_i with $S(v_i) > \theta_{\text{dyn}}$;
// LiDAR Update with LS-ESIKF
$\alpha \leftarrow 1$, $j \leftarrow 0$, $\hat{\mathbf{x}}_{\tau,0} \leftarrow \hat{\mathbf{x}}_\tau$
repeat
  Construct MAP cost and solve for $\tilde{\mathbf{x}}_{\tau,j}$ using (10);
  while (12) *is not satisfied* **do**
    $\alpha \leftarrow \lambda\alpha$
  $\hat{\mathbf{x}}_{\tau,j+1} \leftarrow \hat{\mathbf{x}}_{\tau,j} \boxplus \alpha \tilde{\mathbf{x}}_{\tau,j}$
  $j \leftarrow j + 1$
until $\|\tilde{\mathbf{x}}_{\tau,j} \boxminus \tilde{\mathbf{x}}_{\tau,j-1}\| < \epsilon_{opt}$

IV. Experiments

The experimental setup is detailed in this section, after which the proposed method is assessed on both public datasets and our self-collected dataset.

A. Experimental Setup

Implementation Details: The proposed algorithm is implemented in C++ and built upon the ROS Noetic middleware framework. All evaluations were carried out on a desktop platform operating Ubuntu 20.04 and configured with an Intel i5-13500 processor and 64 GB RAM. In the DOR module, the voxelization parameter is set to 0.25, and the maximum ray-casting distance is 30 m. In the LIO module, the Armijo step-size scaling factor λ is set to 0.6, and the parameter c is set to 1×10^{-3}.

Datasets: The localization performance of R^2-LIO was evaluated on two public datasets and our private dataset. The YULAN [12] dataset was collected using a low-speed vehicle equipped with a Livox Avia LiDAR (10 Hz) and an internal IMU (BMI088, 200 Hz), making it a benchmark for static or low-dynamic environments. In contrast, the HeLiPR [27] dataset was captured in highly dynamic urban scenarios using a real-vehicle platform carrying a Livox Avia LiDAR (10 Hz) and an MTi-300 IMU (100 Hz), serving as a benchmark for evaluating localization accuracy and robustness in highly dynamic scenarios. Our private dataset was collected using a quadruped robot equipped with an NVIDIA AGX Orin computing unit and a Livox Avia LiDAR with a built-in IMU (BMI088, 200 Hz), consisting of three sequences used for qualitative evaluation of localization and mapping performance. The data collection platform is illustrated in Fig. 5.

Evaluation Metrics: To rigorously assess global trajectory consistency, the root mean square error (RMSE) of the Absolute Trajectory Error (ATE) is reported on the public datasets to quantify the overall deviation of the estimated trajectory from the ground truth. For the private dataset, where complete ground-truth annotations are unavailable, the End-to-End Error (E2E) is adopted to characterize the accumulated drift between the beginning and the end of each trajectory, thereby reflecting localization performance in real-world deployment scenarios.

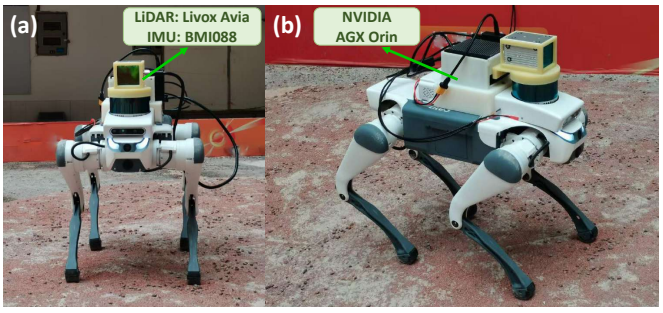

Fig. 5. Our platform for collecting the private dataset.

TABLE I
RMSE (M) ON THE YULAN AND HELIPR DATASETS; E2E (M) ON THE PRIVATE DATASET.

Sequence	FAST-LIO2	Faster-LIO	iG-LIO	Super-LIO	Ours
1017_01	0.65	0.26	0.29	0.27	**0.14**
1017_02	0.72	0.57	0.63	0.62	**0.56**
1017_03	1.00	0.66	0.85	0.47	**0.37**
1017_04	1.14	1.16	**0.72**	1.08	1.06
1017_05	0.39	0.25	0.23	**0.22**	**0.22**
1017_06	1.06	0.84	**0.44**	0.76	0.72
DCC05	4.05	4.48	9.07	3.88	**3.43**
DCC06	3.04	2.22	2.12	1.98	**1.45**
KAIST05	4.51	2.83	×	2.88	**2.29**
KAIST06	4.49	2.49	2.33	2.13	**1.60**
Riverside05	4.96	8.28	×	5.48	**4.79**
Riverside06	12.22	9.34	×	8.83	**7.89**
Private_01	0.17	0.11	0.12	**0.08**	**0.08**
Private_02	0.65	0.49	0.56	0.46	**0.43**
Private_03	3.35	3.84	4.53	3.17	**2.91**

× denotes the system totally failed.

TABLE II
ABLATION STUDIES ON THE HELIPR DATASET.

Sequence	Super-LIO	Ours (w/o DOR)	Ours (w/o LS)	Ours
DCC05	3.88	3.66	3.58	**3.43**
DCC06	1.98	1.68	1.71	**1.45**
KAIST05	2.88	2.81	3.08	**2.29**
KAIST06	2.13	2.11	2.02	**1.60**
Riverside05	5.48	5.30	5.11	**4.79**
Riverside06	8.83	8.20	8.08	**7.89**
Mean	4.19	3.96	3.93	**3.58**

B. Localization Accuracy Evaluation

To quantitatively evaluate the localization performance of the proposed R^2-LIO, we compared it with several SOTA methods, e.g., FAST-LIO2 [3], Faster-LIO [15], iG-LIO [28], and Super-LIO [29].

On the YULAN dataset, R^2-LIO achieved the lowest RMSE in sequences `1017_01`, `1017_02`, `1017_03`, and `1017_05`, with reductions of approximately 46%–79%, 2%–22%, 21%–63%, and 0%–44% compared to the baseline methods, respectively. For sequences `1017_04` and `1017_06`, iG-LIO performed best, mainly due to its GICP observation update strategy, which outperforms the point-to-plane update in certain scenarios. However, when considering only point-to-plane update methods, R^2-LIO still maintains the best performance on these two sequences, with error reductions of approximately 1.9%–8.6% and 5.3%–32.1%. Overall, R^2-LIO demonstrates strong accuracy and robustness in YULAN, which primarily consists of static or low-dynamic environments.

On the more challenging HeLiPR dataset, R^2-LIO achieved the lowest RMSE across all six sequences. Compared to other methods, the error reductions for each sequence are approximately: `DCC05`, 12%–24%; `DCC06`, 27%–52%; `KAIST05`, 19%–49%; `KAIST06`, 25%–64%; `Riverside05`, 3%–42%; and `Riverside06`, 11%–36%. These significant improvements indicate that R^2-LIO exhibits stronger robustness and higher localization performance under dynamic scene conditions, highlighting the contribution of the proposed DOR module and LS-ESIKF in mitigating dynamic disturbances.

Furthermore, we conducted additional evaluations on our self-collected private sequences. Results show that R^2-LIO achieved the lowest localization error in `Private_01`, `Private_02`, and `Private_03`. Here, `Private_01` represents a static scene, while `Private_02` and `Private_03` include lightly dynamic objects. Although the performance gains are limited in these relatively simple scenarios, R^2-LIO consistently achieves the best performance, demonstrating its reliability and generalizability in real-world applications.

C. Ablation Study

An ablation study was conducted on the HeLiPR dataset to analyze the contribution of each core component. The experiments use APE (m) as the evaluation metric, with the best-performing comparative method, Super-LIO, serving as the reference. Three configurations were compared: without DOR module, without LS-ESIKF, and the full system with all modules.

Effect of DOR: Removing DOR module resulted in increased errors across all six sequences. The APE increased by 6.7%, 15.9%, 22.7%, 31.9%, 10.6%, and 3.9%, respectively, with an average increase of 10.6% (from 3.58 m to 3.96 m). The error growth was most pronounced on `KAIST05` and `KAIST06`, which contain a large number of dynamic objects. This indicates that dynamic points introduce erroneous observation constraints, while DOR module effectively improves the reliability of observations in dynamic environments.

Effect of LS-ESIKF: Removing LS-ESIKF also led to increased errors. The APE increased by 4.4%, 17.9%, 34.5%, 26.3%, 6.7%, and 2.4%, with an average increase of 9.8% (from 3.58 m to 3.93 m). This trend is more evident on `KAIST05` and `KAIST06`. In such scenarios, the reduction in the number of valid static points weakens the geometric constraints. LS-ESIKF enhances the stability of iterative convergence by adaptively controlling the step size under reduced geometric constraints.

Overall Analysis: With all modules enabled, the average APE decreases by 14.6% compared to Super-LIO (from 4.19 m to 3.58 m). These results indicate that DOR module improves observation quality, LS-ESIKF enhances the stability of state updates, and their combination delivers consistent and sustained performance improvements.

D. Evaluation of DOR

To assess the contribution of the DOR module in dynamic environments, we conducted experiments on four sequences, namely `1017_01` and `1017_02` from the YULAN, and `Dynamic03` and `Dynamic04` from the M3DGR [30]. As shown in Fig. 6, a1–a4 show representative color images

Fig. 6. Dynamic point detection results on YULAN (`101701`, `101702`) and M3DGR (`Dynamic03`, `Dynamic04`). (a1–a4) Color images. (b1–b4) LiDAR segmentation results (green: static, red: dynamic).

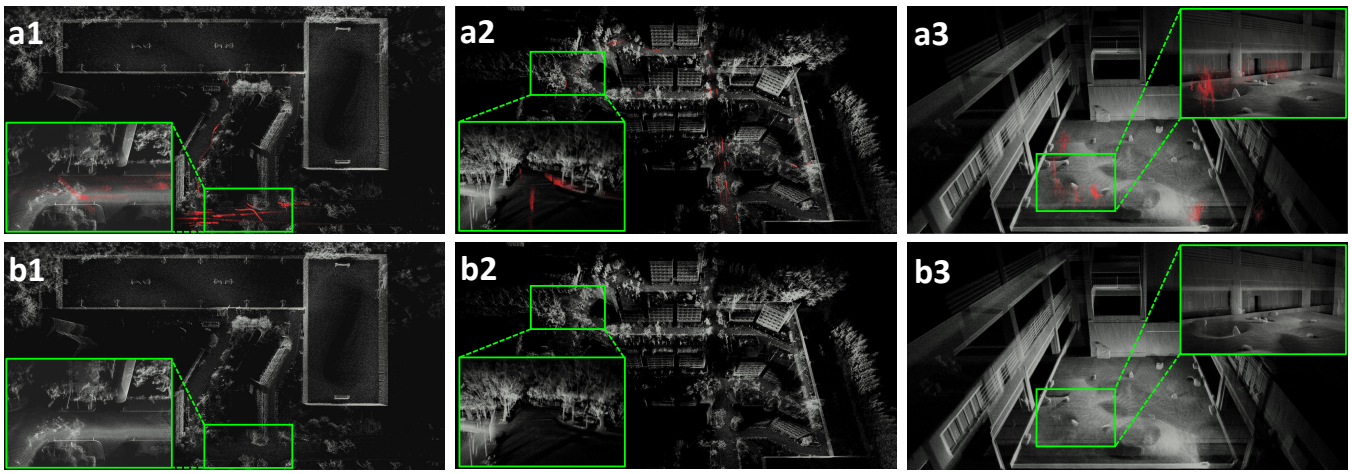

Fig. 7. Mapping results on YULAN `1017_05`, `1017_06`, and `Private_04`. (a1, a2, a3) show the maps before dynamic object removal, while (b1, b2, b3) show the results after removal. Green boxes indicate enlarged local regions for comparison.

from each sequence, while b1–b4 present the corresponding LiDAR-based dynamic segmentation results, where green points denote static points and red points denote dynamic points.

In the `1017_01` (a1, b1) and `1017_02` (a2, b2) sequences of YULAN, the proposed method accurately detects typical dynamic entities in urban environments, while preserving the integrity of static structures including buildings and road surfaces. In the `Dynamic03` (a3, b3) and `Dynamic04` (a4, b4) sequences of M3DGR, although the scenes are more dynamic and contain more distant targets, the method consistently identifies dynamic objects and effectively reduces the misclassification of static background points. Overall, the proposed approach demonstrates strong stability and robustness across different dynamic scenarios.

E. Evaluation of Mapping

To investigate how DOR module affects mapping quality, we conducted comparative evaluations on three sequences: `1017_05`, `1017_06`, and `Private_04`. The visual results are shown Fig. 7. The subsections (a)–(c) correspond to the three sequences. For each sequence, (a1, b1, c1) present the maps generated before dynamic object removal, whereas (a2, b2, c2) show the corresponding maps after removal. Before dynamic object removal (a1, b1, c1), the maps exhibit ghosting and motion artifacts, especially in areas with frequent pedestrian or vehicle movements, leading to blurred structures and reduced geometric consistency. After removing dynamic objects (a2, b2, c2), structural boundaries become clearer and motion artifacts are significantly reduced. The enlarged views further show improved structural continuity, resulting in more accurate and stable maps.

F. Run Time Analysis

Table III presents the average per-frame runtime of the proposed R^2-LIO on the YULAN and HeLiPR datasets. The processing time ranges from 10.61 ms to 13.10 ms for YULAN, and from 10.65 ms to 15.36 ms for HeLiPR. Considering that the typical LiDAR frame interval is around 100 ms, our method demonstrates sufficient real-time capability, ensuring that each point cloud can be processed within its acquisition period.

TABLE III
AVERAGE PROCESSING TIME (MS).

YULAN		HeLiPR	
Sequence	Runtime	Sequence	Runtime
1017_01	11.84	DCC05	15.36
1017_02	13.10	DCC06	15.17
1017_03	12.04	KAIST05	12.96
1017_04	12.23	KAIST06	12.95
1017_05	10.61	Riverside05	11.12
1017_06	11.22	Riverside06	10.65

V. CONCLUSION

In this work, we present R^2-LIO, an efficient solid-state LiDAR–inertial localization framework that integrates dynamic object removal with state estimation based on a line search ESIKF to enhance localization accuracy. By maintaining a local dynamic voxel map, our method effectively removes dynamic points while preserving most static points, without relying on prior assumptions about dynamic objects or scene structures. Additionally, we introduce a line search mechanism that mitigates ill-conditioning and improves convergence under limited static observations. Evaluations on publicly available datasets demonstrate that R^2-LIO delivers leading pose estimation accuracy in both highly dynamic and static scenarios, highlighting its robustness and practical effectiveness.

REFERENCES

[1] C. Gu, Z. Hou, Y. Chen, J. Dong, and X. Gao, "Information entropy-assisted hierarchical framework for unknown environments exploration," in *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 13237–13243, 2025.

[2] 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.

[3] 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.

[4] 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.

[5] C. Wu, Y. You, Y. Yuan, X. Kong, Y. Zhang, Q. Li, and K. Zhao, "Voxelmap++: Mergeable voxel mapping method for online lidar (-inertial) odometry," *IEEE Robotics and Automation Letters*, vol. 9, no. 1, pp. 427–434, 2023.

[6] C. Gu, L. Wang, J. Leng, and X. Gao, "Mis-vio: Deep multi-modal visual-inertial localization with cross-modality interaction and selection," *iOptics*, vol. 1, no. 2, p. 100019, 2025.

[7] 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)*, pp. 4003–4009, 2022.

[8] 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.

[9] L. Wang, C. Gu, J. Yang, and Y. Chen, "Sef-vio: Self-attention-based multimodal fusion for neural visual-inertial odometry," in *2025 40th Youth Academic Annual Conference of Chinese Association of Automation (YAC)*, pp. 2195–2201, 2025.

[10] H. Lim, S. Hwang, and H. Myung, "Erasor: Egocentric ratio of pseudo occupancy-based dynamic object removal for static 3d point cloud map building," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 2272–2279, 2021.

[11] H. Lim, L. Nunes, B. Mersch, X. Chen, J. Behley, H. Myung, and C. Stachniss, "Erasor2: Instance-aware robust 3d mapping of the static world in dynamic scenes," in *Robotics: Science and Systems (RSS)*, 2023.

[12] Y. Liu, Y. Fu, M. Qin, Y. Xu, B. Cui, K. Liu, F. Chen, W. Tao, M. Vlaminck, B. Goossens, P. Z. H. Sun, and H. Zhao, "Standard datasets for autonomous navigation and mapping: A full-stack construction methodology," *IEEE Transactions on Intelligent Vehicles*, vol. 9, no. 9, pp. 5773–5796, 2024.

[13] J. Lin and F. Zhang, "Loam livox: A fast, robust, high-precision lidar odometry and mapping package for lidars of small fov," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3126–3131, 2020.

[14] J. Zhang, S. Singh, *et al.*, "Loam: Lidar odometry and mapping in real-time.," in *Proceedings of the Robotics: Science and systems (RSS), 2014*.

[15] C. Bai, T. Xiao, Y. Chen, H. Wang, F. Zhang, and X. Gao, "Faster-lio: Lightweight tightly coupled lidar-inertial odometry using parallel sparse incremental voxels," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4861–4868, 2022.

[16] J. Lin, C. Zheng, W. Xu, and F. Zhang, "R2live: A robust, real-time, lidar-inertial-visual tightly-coupled state estimator and mapping," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 7469–7476, 2021.

[17] 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)*, pp. 10672–10678, 2022.

[18] Z. Yuan, J. Deng, R. Ming, F. Lang, and X. Yang, "Sr-livo: Lidar-inertial-visual odometry and mapping with sweep reconstruction," *IEEE Robotics and Automation Letters*, vol. 9, no. 6, pp. 5110–5117, 2024.

[19] X. Duan, Q. Hu, M. Ai, P. Zhao, M. Wu, J. Li, and C. Xiong, "Static-lio: A sliding window and terrain-assisted dynamic points removal lidar inertial odometry," *Information Fusion*, vol. 121, p. 103132, 2025.

[20] C. Qian, Z. Xiang, Z. Wu, and H. Sun, "Rf-lio: Removal-first tightly-coupled lidar inertial odometry in high dynamic environments," *arXiv preprint arXiv:2206.09463*, 2022.

[21] Y. Wang, W. Yao, B. Zhang, J. Fu, J. Yang, and G. Sun, "Drr-lio: A dynamic-region-removal-based lidar inertial odometry in dynamic environments," *IEEE Sensors Journal*, vol. 23, no. 12, pp. 13175–13185, 2023.

[22] X. Chen, A. Milioto, E. Palazzolo, P. Giguère, J. Behley, and C. Stachniss, "Suma++: Efficient lidar-based semantic slam," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4530–4537, 2019.

[23] A. Milioto, I. Vizzo, J. Behley, and C. Stachniss, "Rangenet ++: Fast and accurate lidar semantic segmentation," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4213–4220, 2019.

[24] R. J. Elliott, L. Aggoun, and J. B. Moore, *Hidden Markov Models: Estimation and Control*, vol. 29. Springer Science & Business Media, 2008.

[25] V. Bhandari, J. James, T. Phillips, and P. R. McAree, "Moving object segmentation in point cloud data using hidden markov models," *arXiv preprint arXiv:2410.18638*, 2024.

[26] L. Armijo, "Minimization of functions having lipschitz continuous first partial derivatives," *Pacific Journal of mathematics*, vol. 16, no. 1, pp. 1–3, 1966.

[27] M. Jung, W. Yang, D. Lee, H. Gil, G. Kim, and A. Kim, "Helipr: Heterogeneous lidar dataset for inter-lidar place recognition under spatiotemporal variations," *The International Journal of Robotics Research*, vol. 43, no. 12, pp. 1867–1883, 2024.

[28] Z. Chen, Y. Xu, S. Yuan, and L. Xie, "ig-lio: An incremental gicp-based tightly-coupled lidar-inertial odometry," *IEEE Robotics and Automation Letters*, vol. 9, no. 2, pp. 1883–1890, 2024.

[29] L. Wang, X. Zhang, C. Li, D. He, Y. Pan, and J. Yi, "Super-lio: A robust and efficient lidar-inertial odometry system with a compact mapping strategy," *IEEE Robotics and Automation Letters*, vol. 11, no. 3, pp. 2666–2673, 2026.

[30] D. Zhang, J. Zhang, Y. Sun, T. Li, H. Yin, H. Xie, and J. Yin, "Towards robust sensor-fusion ground slam: A comprehensive benchmark and a resilient framework," in *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 8894–8901, 2025.