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

# A Robust LiDAR-Inertial Multi Constraint-Based Localization for Agricultural Environments

Narayan Longani[1], and Gon-Woo Kim[2] *Member, IEEE,*

***Abstract*—Accurate state estimation is essential for an autonomous agricultural robot's reliable operations. The effectiveness of state estimation is influenced by a number of factors, such as sensor-fusion algorithms, the environment, and sensor quality. When the robot traverses in large-scale scenarios, the distance travelled and high-speed mobility produce a drift in the estimation process and should be carefully considered. Moreover, the time-varying noise in sensors affects odometry accuracy further; this is especially noticeable in long travel. This research work is related to the multi-constraints-based state estimation in large unstructured environments with uneven terrain, with a focus on agricultural applications. Using LiDAR-IMU based fusion, our goal is to provide a reliable & accurate localization solution in complex environments like agricultural fields. Furthermore, the agricultural environments become more challenging due to the uneven terrain and lack of features. Our research proposes a hybrid framework which combines factor graph-based optimization & adaptive Kalman filtering to address these challenges in complex environments. Furthermore, performance evaluation is conducted on self-collected datasets from agricultural environments as well as on open-access datasets such as GRACO & KITTI.**

***Index Terms*—Localization, Sensor Fusion, LiDAR-Inertial SLAM, Hybrid Pose Estimation, Time-Varying Noise.**

## I. Introduction

INTEGRATION of robotics in the agricultural sector has increased immensely in recent years due to significant developments in technology. In recent years, it was possible to develop autonomous vehicles in the agriculture sector by incorporating specialised sensors [1]. These sensors help estimate the robot's position and map its surroundings [2], [3]. However, individual sensors have their own set of limitations. In Inertial Navigation Systems (INS), errors can accumulate as inertial measurement unit (IMU) sensor measurements are integrated over time [4]. GPS signals may experience obstruction from high buildings or trees. Visual Odometry (VO) relies on computing camera poses at each time step based on previous estimates. The VO may fail due to blocked camera lenses, insufficient lighting, or insufficient detected feature points [5]. Consequently, reliance on a single sensor often results in inaccurate estimations. These problems can be solved by combining data from multiple sensors into a process using sensor fusion, which produces more accurate results than any one sensor could [6], [7].

Recent simultaneous localization and mapping (SLAM) estimation frameworks are based on feature matching. The more features we can extract, the more accurately we can estimate the pose [8]–[11]. In agricultural environments, we experience many different types of challenges. These environments usually are comprised of trees, grass, and crops, which unfortunately render the detected features unreliable. Consider the edge features [8] obtained from LiDAR scans in agricultural fields. During the feature matching process, the low number of edge features from leaves can significantly impact accuracy. Our experiments also showed that the number of edge features in an agricultural field are comparatively lower than the planar features. The limited number of detected features results in increased error and can sometimes lead to odometry failure [12]. Further, uneven terrain, which is common in agricultural environments, can cause drift in IMU odometry. This often leads to estimation failure due to the abrupt variations of terrains [6], [7], [13]. Recent research has shown considerable interest in enhancing the robustness of state estimation within SLAM frameworks, particularly through high-frequency motion estimation in conjunction with an IMU. Integrating Light Detection and Ranging (LiDAR) with an IMU has become a prominent research focus. LiDAR is less sensitive to changes in ambient conditions and can directly capture three-dimensional environmental data, whereas IMU can provide data at much higher frequencies than LiDAR. Long-term state estimation can have substantial cumulative errors due to IMU's susceptibility to bias and sensor noise [14]. Point cloud maps, created using LiDAR and photogrammetry, have valuable applications in agriculture, particularly in crop monitoring, terrain mapping, and autonomous navigation in GPS-denied areas. High-precision 3D point cloud maps help assess crop health and growth patterns over time, as highlighted by Zhao [15], enabling better monitoring and timely interventions for improved yield management. Additionally, terrain mapping using point clouds provides detailed digital terrain models (DTMs), essential for water management and erosion control, as demonstrated by Sharma et al. [16]. Xu further emphasizes using vehicle-mounted LiDAR for terrain elevation extraction, enhancing navigation in dynamic environments [17]. In GPS-denied areas, point cloud-based SLAM techniques, as explored by Donoso et al. [18], enable real-time navigation for autonomous farming machinery. Moreover, Lucas et al. show that point clouds can aid in vegetation classification, benefiting land-use planning. Overall, point cloud mapping enhances agricultural efficiency and decision-making [19].

Building upon the preceding discussion, we aim to develop

Received 28 February 2025; accepted 13 July 2025. Date of publication 1 August 2025; date of current version 15 August 2025. This article was recommended for publication by Associate Editor M. Hanheide and Editor S. Behnke upon evaluation of the reviewers' comments. This research was supported in part by Innovative Human Resource Development for Local Intellectualization program through the Institute of Information & Communications Technology Planning & Evaluation(IITP) grant funded by the Korea government(MSIT) (IITP-2025-RS-2020-II201462, 50) and in part by the National Research Foundation (NRF) funded by the Korean government (MSIT) (RS-2024-00421129, 50). *(Corresponding author: Gon-Woo Kim).*

[1]Narayan Longani is with the Department of Intelligent Systems and Robotics, Chungbuk National University, Cheongju 28644, South Korea, and also with the Department of Electrical Engineering, COMSATS University Islamabad, Abbottabad 22060, Pakistan (e-mail: narayan@chungbuk.ac.kr).

[2]Gon-Woo Kim is with the Department of Intelligent Systems and Robotics, Chungbuk National University, Cheongju 28644, South Korea (e-mail: gwkim@cbnu.ac.kr).

Digital Object Identifier (DOI): 10.1109/LRA.2025.3595047

a LiDAR-Inertial SLAM system that integrates both loosely and tightly coupled approaches. The main contributions of this work can be summarized as follows:

- Development of a hybrid LiDAR-Inertial framework that incorporates tightly coupled and loosely coupled strategies, particularly designed for complex environments.
- Utilization of an adaptive Kalman filter to effectively accommodate uncertainties stemming from unexpected environmental motion and time-varying noise.
- Addressing the challenges encountered by LiDAR-Inertial SLAM in unstructured agricultural environments, further characterized by uneven terrain.
- Comprehensive evaluations conducted across a range of datasets featuring different robots and environmental conditions, ensuring comprehensive validation of the proposed framework.

The paper is organized as follows: Section II provides a comprehensive review of LiDAR IMU fusion methods. A detailed description of the proposed framework is given in Section III. Section IV covers experimental evaluation, and Section V investigates the real-world agriculture field issues. Section VI serves as the paper's conclusion section.

## II. Related Work

In recent years, several frameworks have emerged for the development of LiDAR-Inertial systems. These methods were originally divided into two categories: loosely coupled and tightly coupled. However, recent frameworks frequently combine elements from both categories, resulting in mixed approaches. Several frameworks have also employed hybrid optimization techniques to further improve performance and robustness.

### A. Loosely Coupled:

Loosely coupled techniques involve two sensors working independently to determine their respective motion constraints, which eventually merge [20]. In the IMU-aided LOAM, the orientation and translation determined by the IMU serve as priors for optimization [21]. Another study [13] combined IMU data with pre-existing maps & pose estimates derived from a lidar-based Gaussian particle filter. Notwithstanding the computational efficiency usually associated with loosely coupled fusion [22], [23], information loss may occur from the decoupling of LiDAR & inertial constraints.

### B. Tightly Coupled:

Tightly coupled approaches fall into two main categories: optimization-based [9], [24], [25], and Kalman filter (EKF)-based [23], [26]. They directly integrate lidar and inertial measurements. Local trajectory optimization was carried out in [27] by utilising the combined constraints from the lidar and IMU data. Furthermore, constraints from LiDAR plane data and inertial pre-integration are handled by LIPS [28], which was introduced by [4] and uses graph optimization. Another noteworthy framework is LIO-mapping (LIOM), created by [29], which refines final poses and maps by combining a novel rotation-constrained mapping technique with graph optimization. For real-time applications, however, the process of creating constraints and batch optimization within a local map window turns out to be time-consuming.

### C. Coupled-Decoupled Approach:

Loosely coupled methods [8], [11] have gained popularity due to their low computational cost, simplicity, and scalability, while tightly coupled methods [30] offer superior accuracy and robust performance. However, extending tightly coupled methods to new environments, integrating additional sensors, or implementing modality-switching schemes can be challenging due to their reliance on specific sensor configurations and calibration pipelines. Furthermore, handling sensor failures may be more difficult in tightly coupled frameworks if a single estimation engine is used. In contrast, loosely coupled methods may improve resilience by distributing the risk of sensor failure across multiple independent estimation engines. Each approach presents its trade-offs. A hybrid strategy, such as the one employed in [31], seeks to balance accuracy, robustness, computational efficiency, and scalability.

### D. Hybrid Optimization:

A hybrid strategy that combines factor graph & Kalman filter is presented in [11], [32], [33], which helps to address several issues with SLAM procedures, especially in large-scale situations where there is a chance of motion distortion & drift during fast motion. LiDAR-inertial odometry (LIO) via Kalman filter and factor graph optimization (LIOFILO) is a technique that uses LIO via these methods to provide high-frequency, high-precision, real-time odometry [32].

## III. Proposed Framework

### A. System Overview

The overall architecture of the system is shown in Fig. 1. The global fixed reference frame is denoted as the world frame $\boldsymbol{W}$, which serves as a stationary reference, typically initialized at the robot's starting position. The body frame $\boldsymbol{B}$ represents the coordinate frame of the robot, and coincides with the IMU frame. Various sensor coordinate frames and the body frame are illustrated in Fig. 2. The state vector is represented as:

$$x = [p^T, \delta^T, v^T, \omega^T]^T \in \mathbb{R}^{12} \tag{1}$$

where $p = [x, y, z]$, $\delta = [\varphi, \theta, \psi]$ represent position and orientation of the robot where $v = [v_x, v_y, v_z]$ and $\omega = [\omega_x, \omega_y, \omega_z]$ represent linear and the angular velocities respectively. The main components of this framework are adaptive Kalman filter (AKF) based IMU Odometry Estimation, LiDAR distortion removal, Feature Extraction, and hybrid optimization. The factor graph-based LiDAR estimates pose, $\hat{x}_{\mathrm{L}}^{\mathrm{POSE}}$, by fusing factors from scan matching, $\hat{x}_{\mathrm{SCAN}}$, loop closures, $\hat{x}_{\mathrm{LOOP}}$, and IMU motion prior as constraint, $\hat{x}_{\mathrm{IMU}}$. The posterior [34] over the LiDAR poses given the various constraints, is expressed as:

$$p\left(\hat{x}_L^{\mathrm{pose}} \mid \hat{x}_{\mathrm{SCAN}}, \hat{x}_{\mathrm{LOOP}}, \hat{x}_{\mathrm{IMU}},\right) \propto \prod_i p\left(\hat{x}_i \mid \hat{x}_L^{POSE}\right) \tag{2}$$

where $\hat{x}_i = \{\hat{x}_{\mathrm{SCAN}}, \hat{x}_{\mathrm{LOOP}}, \hat{x}_{\mathrm{IMU}}\}$ represents a pose-level constraint in the graph. To refine IMU odometry information and support initialization, we integrate an adaptive Kalman filter. The AKF predicts and corrects IMU-based estimates,

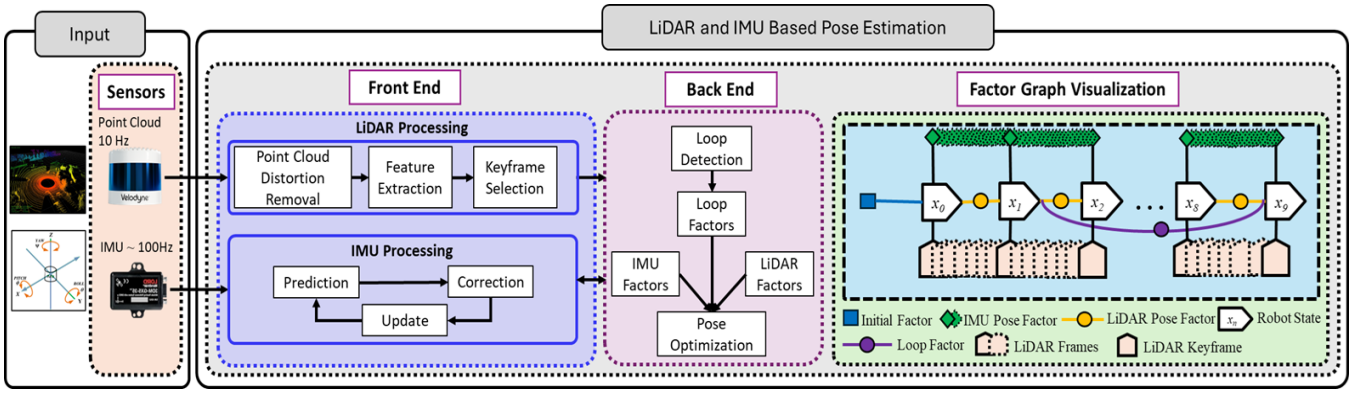



Fig. 1: This figure illustrates the overall SLAM framework architecture from data flow from 3D LiDAR and IMU to the refined pose of the robot. The figure consists of three main components: a front-end for IMU and LiDAR processing, a back-end for pose optimization through multi-sensor fusion, and visualization of the factor graph for LiDAR pose refinement.

incorporating LiDAR-derived pose updates only during the correction phase. To avoid double-counting, only pose-level information is exchanged rather than the raw sensor data. This modular fusion architecture ensures estimator independence and prevents statistical redundancy. Building upon this factorization framework, the Bayesian posterior of the IMU estimator, conditioned on high-rate raw IMU data $z_{\text{IMU}}$ and low-rate LiDAR pose corrections $\hat{x}_{\text{L}}^{\text{POSE}}$ can be expressed as:

$$p\left(\hat{x}_{\text{IMU}} \mid z_{\text{IMU}}, \hat{x}_L^{\text{pose}}\right) \propto p\left(\hat{x}_{\text{IMU}} \mid z_{\text{IMU}}\right) \cdot p\left(\hat{x}_L^{\text{pose}} \mid \hat{x}_{\text{IMU}}\right) \quad (3)$$

This formulation shows that LiDAR information influences the IMU estimator only at the pose level, avoiding raw measurement reuse and eliminating statistical redundancy. The system comprises two distinct estimators-one based on AKF and the other on factor graph optimization-each operating on separate data paths while serving complementary roles.

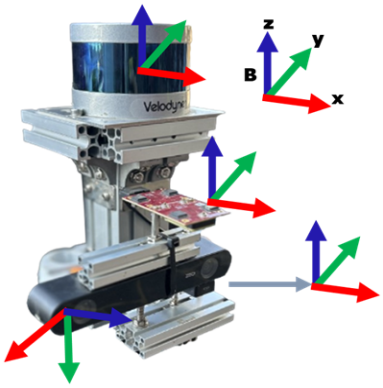



Fig. 2: The 3D coordinate frames of the LiDAR, stereo camera, and IMU sensors are shown in relation to the platform's body frame. The body reference frame $B$ is provided in the top-right corner, using a standard axis-color convention (X: red, Y: green, Z: blue). Each sensor's local frame is annotated for clarity, enabling consistent spatial alignment.

### B. IMU Odom Factors:

The IMU provides measurement of linear acceleration and the angular velocity as follows.

$$\begin{aligned} \hat{\omega}_t &= \omega_t + b_t^{\omega} + \eta_t^{\omega} \\ \hat{a}_t &= R_t^{WB}(a_t - g) + b_t^{a} + \eta_t^{a} \end{aligned} \quad (4)$$

where $\hat{\boldsymbol{\omega}}_t$ and $\hat{\boldsymbol{a}}_t$ are the raw IMU measurements in $\boldsymbol{B}$ at time $\mathbf{t}$. The $b$ and $\eta$ are bias and noise, respectively. $\mathbf{R}^{WB}$ is the rotation matrix from $\boldsymbol{W}$ to $\boldsymbol{B}$, $\mathbf{g}$ is the constant gravity vector in $\boldsymbol{W}$. Initially, IMU pose is predicted using raw sensor data, then these measurements are refined to determine the optimal states. Using a maximum a priori estimation technique, the Sage-Husa Adaptive Kalman filter [11], [35]–[37] maximises filter performance by utilising recent measurements to lessen the impact of older measurements on the estimation process. The filter initiates by adhering to the conventional Extended Kalman Filter steps, then updating the Q and R matrices. This adaptive adjustment enables the filter to dynamically adapt the gain K, particularly beneficial in scenarios where noise statistics fluctuate over time. This improves the estimation accuracy by giving the most recent measurements more weight. For systems with time-varying characteristics, the expected values of process noise $w_k$ and measurement noise $v_k$ are defined as

$$\begin{aligned} E[w_k] &= q_k \\ E[v_k] &= r_k \end{aligned} \quad (5)$$

where $E[\cdot]$ denotes the mean value operator. Unlike traditional Kalman filtering that assumes zero-mean noise, the Sage-Husa filter accommodates non-zero mean noise distributions, making it more robust in real-world applications. Now, the filter proceeds in three main steps as:

*1) Prediction Step:*

$$\begin{aligned} \hat{x}_{k+1|k} &= \phi_{k+1|k}\hat{x}_{k|k} + q_{k+1} \\ P_{k+1|k} &= \phi_{k+1|k}P_{k|k}\phi_{k+1|k}^T + Q_k \end{aligned} \quad (6)$$

*2) Correction Step:*

$$\begin{aligned} K_{k+1} &= P_{k+1|k}H_{k+1}^T\left[H_{k+1}P_{k+1|k}H_{k+1}^T + R_{k+1}\right]^{-1} \\ P_{k+1|k+1} &= \left[I - K_{k+1}H_{k+1}\right]P_{k+1|k} \\ \tilde{z}_{k+1} &= z_{k+1} - H_{k+1}\hat{x}_{k+1|k} - r_{k+1} \\ \hat{x}_{k+1|k+1} &= \hat{x}_{k+1|k} + K_{k+1}\tilde{z}_{k+1} \end{aligned} \quad (7)$$

*3) Adaptive Step:*

$$\begin{aligned}
\hat{r}_{k+1} &= (1-d_k)\hat{r}_k + d_k(z_{k+1} - H_{k+1}\hat{x}_{k+1|k}) \\
\hat{R}_{k+1} &= (1-d_k)\hat{R}_k + d_k(\tilde{z}_{k+1}\tilde{z}_{k+1}^T - H_{k+1}P_{k+1|k}H_{k+1}^T) \\
\hat{q}_{k+1} &= (1-d_k)\hat{q}_k + d_k(\hat{x}_{k+1} - \phi_{k+1|k}\hat{x}_k) \\
Q_{k+1} &= (1-d_k)Q_k + d_k(K_{k+1}\tilde{z}_{k+1}\tilde{z}_{k+1}^T K_{k+1}^T + P_{k+1} \\
&\quad - \phi_{k+1|k}P_{k|k}\phi_{k+1|k}^T)
\end{aligned} \tag{8}$$

The parameter $d$ is known as the attenuation factor, given by

$$d_k = \frac{1-b}{1-b^{k+1}} \quad (0 < b < 1) \tag{9}$$

The notations used are summarized in Table I. The forgetting factor $b$ is typically determined empirically, with values between 0.95 and 0.99, enabling online estimation of the noise covariances $Q$ and $R$. The Sage-Husa adaptive Kalman filter iteratively updates these matrices based on innovation sequences [38], [39]. Although adaptive filtering refines IMU-based pose estimates by adjusting noise covariances in real-time, it remains susceptible to drift over time, particularly in uneven agricultural terrains. To mitigate this, we incorporate pose-level corrections from LiDAR odometry to constrain the IMU estimation. Since the raw IMU and LiDAR measurements are never reused across pipelines, cross-correlation between estimators remains minimal. A similar strategy to avoid double-counting was adopted in [40], where only processed priors are shared. Since the estimation paths operate on distinct observables and exchange only pose-level corrections, cross-correlation is minimized and handled implicitly through adaptive covariance updates within the AKF.

TABLE I: Summary of Variables Used in Sage-Husa Adaptive Kalman Filter

| Symbol(s) | Description |
|---|---|
| $\hat{x}_{k+1\|k}$, $\hat{x}_{k+1\|k+1}$ | Predicted & updated state estimates |
| $P_{k+1\|k}$, $P_{k+1\|k+1}$ | Predicted & updated state covariances |
| $q_k$, $r_k$ | Process & measurement noise mean |
| $Q_k$ | Process noise covariance |
| $R_k$ | Measurement noise covariance |
| $\hat{q}_k$, $\hat{r}_k$ | Estimated means of $q_k$ & $r_k$ |
| $z_k$, $\tilde{z}_k$ | Measurement & innovation (residual) |
| $\Phi_{k+1\|k}$, $H_k$ | State transition & Measurement matrix |
| $K_k$, $I$ | Kalman gain & Identity matrix |
| $d_k$, $b$ | Attenuation & forgetting factors |

### C. LiDAR Factors:

The process of obtaining the LiDAR factor consists of LiDAR distortion removal, Features Extraction, and Optimization. The LiDAR data is also utilized to build the point cloud map. The LiDAR distortion removal is performed on the LiDAR sensor's point cloud first. Here, the data is segmented, outliers are eliminated, and the points are remapped into a range image [13], [30]. Next, feature extraction is performed. Usually, edge and planar features are extracted from the LiDAR point cloud [8]. These features are classified based on the continuity of the local curvature using (10). If the local curvature calculated is less than the threshold, then it is a planar feature; else edge feature. The local curvature-based planar features extraction can be visualized in Fig. 3.

$$c = \frac{1}{|S||X_{(k,i)}^L|} \sum_{j\in S \quad j\neq i} (X_{(k,i)}^L - X_{(k,j)}^L) \tag{10}$$

Based on our experiments across various environments, we observed that planar features significantly outnumbered edge features in agricultural environments such as rice and wheat fields. This motivates our decision to use planar features for LiDAR pose estimation.

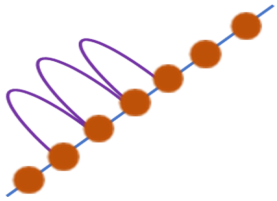


Fig. 3: Planar features represented by the closed points

Table II summarizes the average number of extracted edge and planar features per LiDAR frame across different datasets. Now, the features extracted at time $t$ compose a LiDAR frame $F_t^p$ in frame $\boldsymbol{B}$. Now the generation of the point cloud map & the lidar factors using the $F_t^p$ is described as follows.

TABLE II: Average Feature Points Extracted per Frame Across Different Environments. ATP = Average Total Points, AEP = Average Edge Points, APP = Average Planar Points.

| Sequence Name | ATP | AEP | APP |
|---|---|---|---|
| Rice_Field_1348 | 8315 | 178 | 2540 |
| Rice_Field_1354 | 8660 | 195 | 2480 |
| Wheat_Field_1405 | 8570 | 302 | 2600 |
| Wheat_Field_1411 | 8760 | 301 | 2481 |

*1) Point Cloud Map:* Point cloud map is constructed with the $n$ most recent keyframes [24], as opposed to optimising the transformation between successive scans. Relative transformation is used to obtain the keyframes ${}'F_t^p$ frame in $\boldsymbol{W}$. After that, these sub-keyframes are combined to create the voxel map, $M_t^p$. $M_t^p$ is the down-sampled version of the transformed planar features in $\boldsymbol{W}$, by removing duplicate features that are contained in the same voxel cell. This voxel map, $M_t^p$, also serves as the reference for scan matching against the newly acquired LiDAR frame $F_{t+1}^P$. The scan-matching process aligns $F_{t+1}^P$ to $M_t^P$, following the robust and computationally efficient method described in [8]. The relationship between the voxel map and the transformed planar features is:

$$M_t^p = {}'F_t^p \cup {}'F_{t-1}^p \ldots \cup {}'F_{t-n}^p \tag{11}$$

*2) Pose Estimation:* The pose using Lidar's data is estimated using transformed features, ${}'F_t^P$ in frame $\boldsymbol{W}$. For each feature point in ${}'F_t^P$, corresponding features are identified in the voxel map $M_t^p$. The distance, $d_k$, between a current feature point and its corresponding point in $M_t^p$ is computed using:

$$d_k = \frac{\left|(p_{t+1,k} - p_{t,u}) \cdot [(p_{t,u} - p_{t,v}) \times (p_{t,u} - p_{t,w})]\right|}{\left\|(p_{t,u} - p_{t,v}) \times (p_{t,u} - p_{t,w})\right\|} \tag{12}$$

For a feature point $p_{t+1,k}$ in ${}'F_{t+1}^p$, and the points $\{p_{t,u}, p_{t,v}, p_{t,w}\}$ forms the corresponding planar patch in $M_t^p$. Then, optimization is used to minimize the sum of distances:

$$\min_{T_{i+1}} \left( \sum_{p_{i+1,k} \in {}'F_{i+1}^p} d_k \right) \tag{13}$$

This process ensures that the robot's pose is accurately estimated by leveraging planar features, minimizing errors caused by misalignment, and optimizing iteratively.

### D. Loop Closure Factor:

Loop closure helps in correcting the drift of the robot. Our approach is to develop a two-step loop detection process. We start with constructing a compact 2D polar representation of 3D keyframe cloud [41]. Each keyframe is discretized into a polar grid $(r_i, \theta_j)$ composed of radial rings, $N_r$ and angular sectors, $N_s$. Now, each point $p^k = (x_k, y_k, z_k)$ is converted to polar coordinates $(\theta_k, r_k)$ and assigned to a corresponding section of $P_{ij}$. Now the each section of $P_{ij}$ consists of all the points that fall within the radial bound of $r_{i-1} \leq r_k < r_i$ and angular bounds $\theta_{j-1} \leq \theta_k < \theta_j$. Then, a unique $a_{ij}$ is assigned to each section, which is the maximum euclidean distance of the points in the keyframe cloud $p^n_{ij}$.

Now we have a compact representation of the keyframe cloud. To enhance robustness and compactness, a Binary Pattern operator is applied as defined in algorithm 1. This provides a robust representation of local structures. Initially, to identify the loop candidates, we compute the cosine similarity between the current keyframe descriptor and all prior keyframe descriptors. Additionally, a root-shifting mechanism is applied to improve rotational invariance in descriptor matching. For the current keyframe cloud $F_{t+1}$, the graph state $x_r$ is one of the returned candidates. Then, fine-grained scan matching is performed between $F_{t+1}$ and a local window of keyframes $\{F_{r-m}, ..., F_r, ..., F_{r+m}\}$. If it satisfies the matching threshold, then it is added as a loop closure constraint.

## IV. Experiments And Results

We conducted a comprehensive series of experiments to rigorously assess the efficacy of our proposed framework, both qualitatively and quantitatively. Our proposed LiDAR-Inertial framework underwent rigorous comparison with established methods, including Fast-LIO, A-Loam, and Lego-LOAM. All tests were performed on a dedicated system featuring an AMD Ryzen 7 5800 16-Core Processor and 16 GB RAM, running Ubuntu 20.04 with ROS. Noteworthy, all algorithms were executed concurrently on identical datasets, ensuring a fair and consistent comparison. We have used the root-mean-squared-error (RMSE) as a primary metric for evaluation.

**Algorithm 1:** Binary Pattern Based Descriptor

**Input:** Image of Key-frame Cloud
**Output:** Binary Pattern Based Descriptor
**Procedure: Binary Pattern Computation**
**for** *x, y in* `image` **do**
  Extract the centre pixel
  Initialize `code` $\leftarrow 0$;
  **for** $i \leftarrow 0$ **to** 7 **do**
    Compute neighbor coordinates:
    $dx \leftarrow \cos(2\pi i/8)$, $dy \leftarrow \sin(2\pi i/8)$;
    Extract neighbor pixel:
    `neighbor` $\leftarrow image[x+dx, y+dy]$;
    Update binary code:
    `code` += (`neighbor` > `center`) $\ll (7-i)$;
  **end**
  Assign the computed code:
  $image[x, y] \leftarrow$ `code`;
**end**

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} |P_{GT} - P_{EST}|^2} \tag{14}$$

Where $P_{\text{GT}}$ represents the ground truth position and $P_{\text{EST}}$ represents the estimated position in the XY-plane.

### A. Evaluation on Self-Collected Agriculture Dataset

We have evaluated our framework on the self-collected datasets in the agricultural environments. For validation, we meticulously collected distinct datasets spanning both rice and wheat fields environments as shown in Fig. 4. A mini sensor platform was designed to rigidly place multiple sensors on the tractor as in Fig. 5. The mini sensor platform utilized in this study comprises a ZED 2 Stereo Camera, Velodyne VLP-16 LiDAR, a MicroStrain 3DM-GX5-25 IMU, and the TDR2000 GPS. To establish ground truth for evaluation purposes, we have used the Novatel GPS PWR 7D system. A few odometry plots for the agriculture fields based on the proposed and SOTA frameworks are in the Fig. 6a and in Fig.6b.

### B. Evaluation on Public Datasets

We have also evaluated the performance of our framework on public datasets like the KITTI raw dataset [42] and the GRACO dataset [43]. For the ease of the representation, we have used KRS for "Kitti Raw Sync 2011 09 26" and GO

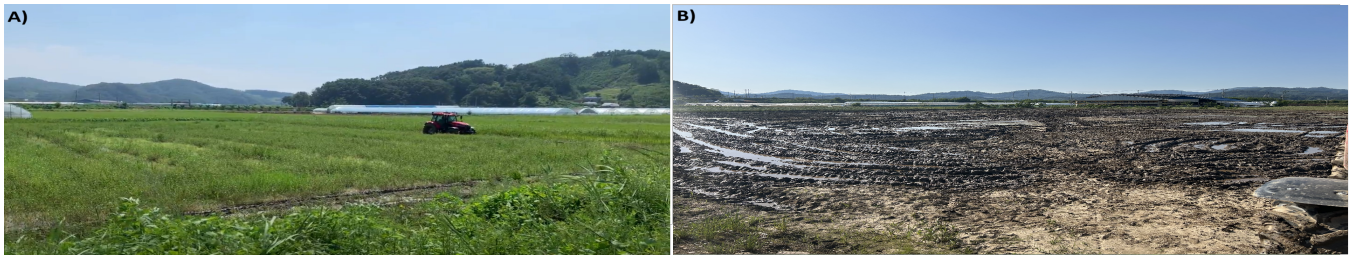



Fig. 4: Visualization of the challenging agriculture field environments used during data collection. The rice field (in A) includes dense vegetation with a waterlogged surface, making feature extraction and motion estimation more difficult. The wheat field (in B) exhibited similar challenges, with the additional complexity of both waterlogged (soft) and dry (hard) soil regions, leading to surface discontinuities and significant mechanical vibrations.

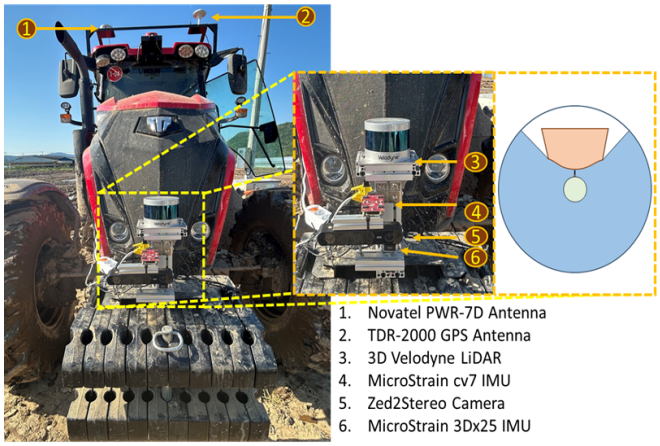



Fig. 5: Mini multi-sensor platform rigidly mounted at the front of a tractor for data acquisition in unstructured agricultural fields such as rice and wheat. GPS antennas were mounted on top of the tractor.

for "GRACO Ground Outdoor". The odometry plots GO 01 and KRS 009 are in Fig. 6c and in Fig. 6d respectively.

The RMSE based results are summarized in Table III, which provides a comprehensive evaluation on the self-collected agricultural datasets and public datasets. The term Fail in Table III denotes a failure in pose estimation when the RMSE $\geqslant$ 11m. Our proposed framework demonstrates improved accuracy over other LiDAR-Inertial methods in several cases. To highlight the performance, we have included $I_{SOTA}$ in Table III, which quantifies the percentage improvement of our method over the best-performing SOTA method for each sequence. The percentage improvement is calculated as:

$$I_{SOTA} = \left( \frac{RMSE_{SOTA} - RMSE_{Our\ Work}}{RMSE_{SOTA}} \right) \times 100\% \qquad (15)$$

## V. Discussion

In this research, we presented a LiDAR–IMU-based pose estimation framework tailored for unstructured environments with uneven terrain, such as agricultural fields. Through real-time experiments, particularly in wheat and rice fields, we observed several challenges in reliable pose estimation. A major issue is the reduced effectiveness of LiDAR-based odometry due to limited field of view (FOV) and feature sparsity in such open, low-structured environments.

### A. Initial Pose Estimation Inconsistency:

In real-time deployment, we also encountered initialization problems. One such instance involved the robot initially estimating backwards motion while it was moving in a forward direction. An inaccurate pose estimated by the sensors is the cause of this disparity. This issue resembles problems in tightly coupled systems, where inaccurate estimation from one sensor can negatively impact the entire state estimate. Fig. 7b visualizes this problem during the first few seconds of motion.

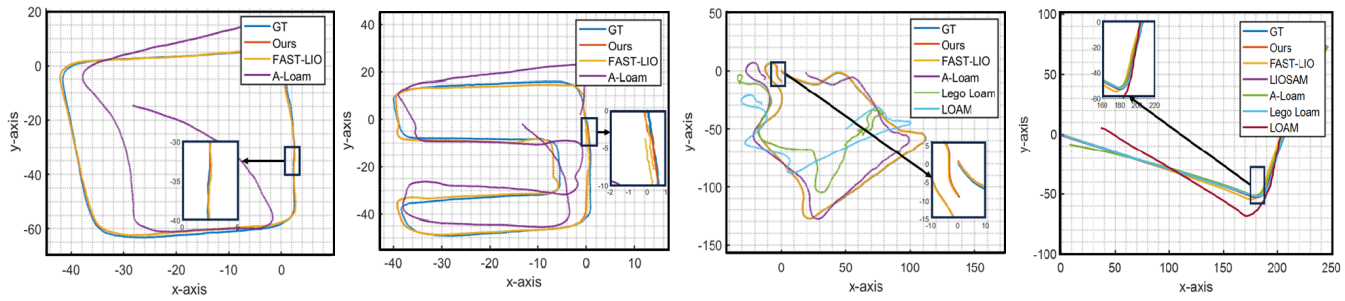
### B. Estimation Near Obstructive Structures:

As illustrated in Fig. 5, the placement of the mini sensor platform on the tractor was carefully selected to maximize the FOV while adhering to the structural and functional constraints of the tractor. The placement was determined in consultation with the tractor's design team, and the platform was mounted at the front of the vehicle. However, this position makes the rear side obstructed by the tractor's structural components, resulting in occluded regions and missing environmental information. Further, when the tractor traversed areas near external obstructions—such as sheds or huts—the LiDAR's data was further reduced. These occlusions degraded the feature extraction process, often leading to pose estimation failures. For example, Fig. 7a illustrates how operating near obstructive structures can result in drift and eventual failure in LiDAR odometry.

Our proposed framework overcomes these limitations & consistently delivers reliable pose estimates even in such degraded conditions. This is reflected in the quantitative comparisons presented in Table III, where our method achieves accuracy comparable to state-of-the-art (SOTA) methods on public datasets, also significantly outperforming on our self-collected agricultural data. However, the current framework does not explicitly model agricultural-specific factors such as seasonal changes in crop structure or vegetation-specific LiDAR processing. Addressing these aspects remains an open challenge. Future work will explore adaptive feature extraction and environment-specific filtering techniques to further improve localization's accuracy under dynamic conditions.

## VI. Conclusion

Precise position robot estimation in an agricultural environment is a difficult task. The main challenges are the unstructured environment and the highly distorted IMU

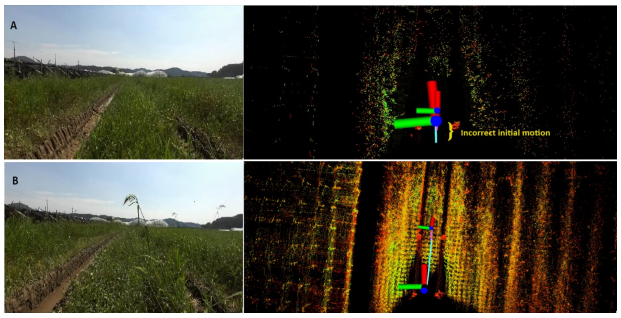



(a) Rice Field 1342 (b) Rice Field 1358 (c) GO 01 (d) KRS 0009

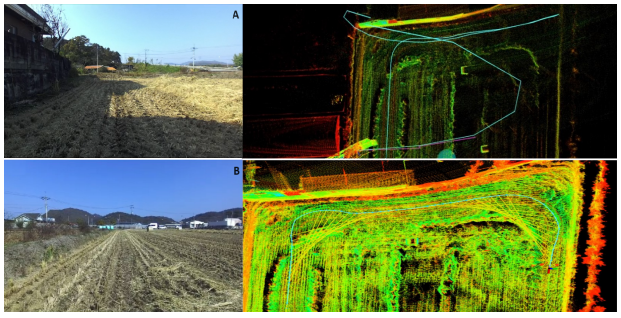
Fig. 6: Comparison of Our Framework With Existing Frameworks on Self-Collected Agriculture Dataset and Public Dataset.

TABLE III: Evaluation on Self-Collected Agricultural Datasets (Rice and Wheat Fields) and Public Datasets (GRACO and KITTI Raw). The lowest RMSE per sequence is bolded, and the minimum RMSE from the SOTA is highlighted grey. If a SOTA method achieves the best RMSE, it is bolded and shaded in grey. When our method outperforms all others, the $I_{SOTA}$ column shows the percentage improvement over the best baseline, marked with an upward arrow (↑), and no improvement is represented with "–". A sequence is labelled as a "Fail" if RMSE ⩾ 11 m.

| | Sequence Name | Our Work | Fast-LIO | LIO-SAM | A-Loam | Lego Loam | LOAM | $I_{SOTA}$(%) |
|---|---|---|---|---|---|---|---|---|
| **Rice Fields** | 134243 | **2.0636** | 2.0957 | Fail | Fail | Fail | Fail | 1.53 ↑ |
| | 134846 | 3.1786 | **3.0932** | Fail | Fail | Fail | Fail | — |
| | 135407 | **2.4246** | 2.5189 | Fail | Fail | Fail | Fail | 3.74 ↑ |
| | 135859 | **2.7927** | 2.9539 | Fail | Fail | Fail | Fail | 5.46 ↑ |
| | 140521 | **2.0863** | 2.1299 | Fail | Fail | Fail | Fail | 2.05 ↑ |
| | 141104 | **2.5436** | 2.6966 | Fail | Fail | Fail | Fail | 5.67 ↑ |
| **Wheat Fields** | A | **0.6941** | 3.6786 | Fail | 1.7515 | Fail | Fail | 60.37 ↑ |
| | B | **0.46919** | 1.1663 | 0.4950 | 2.1132 | 3.7158 | 1.4072 | 5.21 ↑ |
| | C | 0.6291 | 0.70309 | 0.66937 | **0.54337** | 7.2853 | 1.5107 | — |
| **KITTI Raw Dataset (KRS)** | 0009 | 0.9941 | 2.9574 | 0.9096 | 2.6325 | **0.8569** | Fail | — |
| | 0059 | **2.2506** | 3.0987 | 2.2507 | 2.6808 | 2.429 | 4.3251 | 0.004 ↑ |
| | 0084 | **2.2819** | 3.0858 | 2.2906 | 2.3131 | 2.3733 | 3.1127 | 0.38 ↑ |
| | 0095 | **2.356** | 2.4431 | 2.3574 | 2.4407 | 2.6606 | Fail | 0.059 ↑ |
| | 0104 | **2.2495** | 2.3457 | 2.2494 | 2.4346 | 2.7828 | 4.3403 | — |
| **GRACO Dataset (GO)** | GO 01 | 0.5766 | 0.6134 | **0.31453** | 1.4333 | Fail | Fail | — |
| | GO 02 | 0.8643 | 0.8717 | **0.83209** | 2.6999 | Fail | Fail | — |
| | GO 03 | **0.7024** | 0.7236 | 1.1245 | 4.8562 | Fail | Fail | 2.93 ↑ |
| | GO 04 | 0.51414 | **0.4971** | 0.7291 | 2.0164 | Fail | Fail | — |
| | GO 05 | **0.3617** | 0.5553 | 1.98020 | 4.0807 | Fail | Fail | 34.86 ↑ |
| | GO 06 | **0.5145** | 0.6069 | 0.92172 | 4.6208 | Fail | Fail | 15.23 ↑ |




(a) A: An example of a pose estimation failing in an agricultural environment because of limiting features and blocking structures. B: demonstrating how well our framework works to overcome these challenges




(b) A: A real-time implementation illustration of an initialization issue that shows backwards motion even though the robot is moving ahead. B: An illustration of how our framework solves this problem by ensuring forward motion and pose estimation in agricultural environments.

Fig. 7: Enhancing Initialization and Addressing Odometry Failures in Agricultural Environments

readings caused by the uneven terrain. To address these challenges, we present a novel hybrid optimization-based state estimation framework utilizing LiDAR and the IMU sensor. Our framework uses a constraint-based methodology to enhance sensor performance in challenging environments, particularly where terrain irregularities affect sensor accuracy. This hybrid network ensures robust overall performance, even when one sensor's pose estimation is degraded, as the other sensor compensates for this deficiency. Additionally, the implementation of an adaptive Kalman filter further supports pose estimation in uneven terrain by effectively utilizing IMU data. We validated the proposed framework using a self-collected dataset from agricultural environments, such as rice fields, and confirmed its effectiveness across various scenarios. To ensure comprehensive validation, we also tested the framework on well-known public datasets like GRACO and the KITTI Raw synced dataset. The results demonstrate the robustness and adaptability of our approach in diverse and challenging conditions, making it a viable solution for precise robotic positioning in agricultural fields.

## References

[1] L. F. Oliveira, A. P. Moreira, and M. F. Silva, "Advances in agriculture robotics: A state-of-the-art review and challenges ahead," *Robotics*, vol. 10, no. 2, p. 52, 2021.

[2] A. S. Aguiar, F. N. dos Santos, J. B. Cunha, H. Sobreira, and A. J. Sousa, "Localization and mapping for robots in agriculture and forestry: A survey," *Robotics*, vol. 9, no. 4, p. 97, 2020.

[3] M. Zaffar, S. Ehsan, R. Stolkin, and K. M. Maier, "Sensors, slam and long-term autonomy: A review," in *2018 NASA/ESA Conference on Adaptive Hardware and Systems (AHS)*. IEEE, 2018, pp. 285–290.

[4] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, "Imu preintegration on manifold for efficient visual-inertial maximum-a-posteriori estimation," in *Proceedings of Robotics: Science and Systems*, Rome, Italy, July 2015.

[5] Z. Javed and G.-W. Kim, "Omni-directional visual-lidar slam for multi-camera system," *The Journal of Korea Robotics Society*, vol. 17, no. 3, pp. 353–358, 2022.

[6] M. Magnusson, A. Lilienthal, and T. Duckett, "Scan registration for autonomous mining vehicles using 3d-ndt," *Journal of Field Robotics*, vol. 24, no. 10, pp. 803–827, 2007.

[7] S. Kohlbrecher, O. Von Stryk, J. Meyer, and U. Klingauf, "A flexible and scalable slam system with full 3d motion estimation," in *2011 IEEE international symposium on safety, security, and rescue robotics*. IEEE, 2011, pp. 155–160.

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

[9] T. Qin, P. Li, and S. Shen, "Vins-mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 1004–1020, 2018.

[10] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "Orb-slam: a versatile and accurate monocular slam system," *IEEE transactions on robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[11] G. P. C. Júnior, A. M. Rezende, V. R. Miranda, R. Fernandes, H. Azpúrua, A. A. Neto, G. Pessin, and G. M. Freitas, "Ekf-loam: an adaptive fusion of lidar slam with wheel odometry and inertial data for confined spaces with few geometric features," *IEEE Transactions on Automation Science and Engineering*, vol. 19, no. 3, pp. 1458–1471, 2022.

[12] 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.

[13] 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)*. IEEE, 2018, pp. 4758–4765.

[14] W. Zhen, S. Zeng, and S. Soberer, "Robust localization and localizability estimation with a rotating laser scanner," in *2017 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2017, pp. 6240–6245.

[15] B. Zhao, S. Liu, X. Zhao, L. Zhu, T. Zhang, Z. Han, and W. Zhang, "Research on the construction method of high precision 3d point cloud map for agricultural environments," *INMATEH - Agricultural Engineering*, vol. 72, no. 1, pp. 391–401, 2024.

[16] R. Sharma, Z. Xu, R. Sugumaran, and S. Oliveira, "Parallel landscape driven data reduction & spatial interpolation algorithm for big lidar data," *ISPRS international journal of geo-information*, vol. 5, no. 6, p. 97, 2016.

[17] P. Xu, S. Liu, D. Zhao, X. Wu, Z. Lin, and C. Zhang, "Extracting terrain elevation information in front of the vehicle based on vehicle-mounted lidar in dynamic environments," *Measurement Science and Technology*, vol. 36, no. 1, p. 016009, 2024.

[18] F. Donoso, K. Austin, and P. McAree, "Three new iterative closest point variant-methods that improve scan matching for surface mining terrain," *Robotics and Autonomous Systems*, vol. 95, pp. 117–128, 2017.

[19] C. Lucas, W. Bouten, Z. Koma, W. D. Kissling, and A. C. Seijmonsbergen, "Identification of linear vegetation elements in a rural landscape using lidar point clouds," *Remote Sensing*, vol. 11, no. 3, 2019.

[20] J. Tang, Y. Chen, X. Niu, L. Wang, L. Chen, J. Liu, C. Shi, and J. Hyyppä, "Lidar scan matching aided inertial navigation system in gnss-denied environments," *Sensors*, vol. 15, no. 7, pp. 16 710–16 728, 2015.

[21] S. Lynen, M. W. Achtelik, S. Weiss, M. Chli, and R. Siegwart, "A robust and modular multi-sensor fusion approach applied to mav navigation," in *2013 IEEE/RSJ international conference on intelligent robots and systems*. IEEE, 2013, pp. 3923–3929.

[22] M. Li, B. H. Kim, and A. I. Mourikis, "Real-time motion tracking on a cellphone using inertial sensing and a rolling-shutter camera," in *2013 IEEE International Conference on Robotics and Automation*. IEEE, 2013, pp. 4712–4719.

[23] Z. Huai and G. Huang, "Robocentric visual–inertial odometry," *The International Journal of Robotics Research*, vol. 41, no. 7, pp. 667–689, 2022.

[24] S. Leutenegger, P. Furgale, V. Rabaud, M. Chli, K. Konolige, and R. Siegwart, "Keyframe-based visual-inertial slam using nonlinear optimization," *Proceedings of Robotis Science and Systems (RSS) 2013*, 2013.

[25] G. Hyeonjae, L. Dongjae, S. Kwanhyung, A. Seungwook, and K. Ayoung, "Gnss-lidar-inertial state estimation system for mapping and autonomous navigation in unstructured environments," *Journal of Robotics Society*, vol. 18, no. 1, pp. 72–81, 2023.

[26] G. P. Huang, N. Trawny, A. I. Mourikis, and S. I. Roumeliotis, "Observability-based consistent ekf estimators for multi-robot cooperative localization," *Autonomous Robots*, vol. 30, pp. 99–122, 2011.

[27] C. Park, P. Moghadam, S. Kim, A. Elfes, C. Fookes, and S. Sridharan, "Elastic lidar fusion: Dense map-centric continuous-time slam," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 1206–1213.

[28] P. Geneva, K. Eckenhoff, Y. Yang, and G. Huang, "Lips: Lidar-inertial 3d plane slam," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 123–130.

[29] H. Ye, Y. Chen, and M. Liu, "Tightly coupled 3d lidar inertial odometry and mapping," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 3144–3150.

[30] 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)*. IEEE, 2020, pp. 5135–5142.

[31] S. Zhao, H. Zhang, P. Wang, L. Nogueira, and S. Scherer, "Super odometry: Imu-centric lidar-visual-inertial estimator for challenging environments," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 8729–8736.

[32] J. Tang, X. Zhang, Y. Zou, Y. Li, and G. Du, "A high-precision lidar-inertial odometry via kalman filter and factor graph optimization," *IEEE Sensors Journal*, 2023.

[33] D. He, H. Li, and J. Yin, "Ligo: A tightly coupled lidar-inertial-gnss odometry based on a hierarchy fusion framework for global localization with real-time mapping," *IEEE Transactions on Robotics*, 2025.

[34] F. Dellaert, "Factor graphs and gtsam: A hands-on introduction," *Georgia Institute of Technology, Tech. Rep*, vol. 2, no. 4, 2012.

[35] K.-V. Yuen, Y.-S. Liu, and W.-J. Yan, "Estimation of time-varying noise parameters for unscented kalman filter," *Mechanical Systems and Signal Processing*, vol. 180, p. 109439, 2022.

[36] D. Wang, H. Zhang, and B. Ge, "Adaptive unscented kalman filter for target tacking with time-varying noise covariance based on multi-sensor information fusion," *Sensors*, vol. 21, no. 17, p. 5808, 2021.

[37] H. Ahmed, I. Ullah, U. Khan, M. B. Qureshi, S. Manzoor, N. Muhammad, M. U. Shahid Khan, and R. Nawaz, "Adaptive filtering on gps-aided mems-imu for optimal estimation of ground vehicle trajectory," *Sensors*, vol. 19, no. 24, p. 5357, 2019.

[38] M. Nilsson, "Kalman filtering with unknown noise covariances," in *Reglermöte 2006*, 2006.

[39] D. Simon, *Optimal state estimation: Kalman, H infinity, and nonlinear approaches*. John Wiley & Sons, 2006.

[40] L. Bai, Y. Ge, and H. Wymeersch, "Belief propagation-based target handover in distributed integrated sensing and communication," *arXiv preprint arXiv:2506.23118*, 2025.

[41] G. Kim and A. Kim, "Scan context: Egocentric spatial descriptor for place recognition within 3d point cloud map," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 4802–4809.

[42] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.

[43] Y. Zhu, Y. Kong, Y. Jie, S. Xu, and H. Cheng, "Graco: A multimodal dataset for ground and aerial cooperative localization and mapping," *IEEE Robotics and Automation Letters*, vol. 8, no. 2, pp. 966–973, 2023.