UA-LIO: An Uncertainty-Aware LiDAR-Inertial Odometry for Autonomous Driving in Urban Environments

Qi Wu, Xieyuanli Chen, *Member, IEEE*, Xiangyu Xu, Xinliang Zhong,
Xingwei Qu, *Student Member, IEEE*, Songpengcheng Xia, *Member, IEEE*, Guoqing Liu,
Liu Liu, Wenxian Yu, *Senior Member, IEEE*, and Ling Pei, *Senior Member, IEEE*

***Abstract*— Odometry estimation is a fundamental technique for many robotics and autonomous driving applications. Recent advances using light detection and ranging (LiDAR) and inertial measurement unit (IMU) sensor-fusion methods show promising potential in odometry estimation by combining the precise perception of LiDAR and the high-frequency motion estimation from IMU sensors. In this article, we propose a novel uncertainty-aware LiDAR-inertial odometry (LIO) algorithm designed for autonomous vehicles operating in urban driving environments. Unlike previous approaches that use point-to-point or point-to-plane methods for updates, in this article, we employ a distribution-to-distribution approach for updates. Each point sampled on a surface is modeled as a Gaussian distribution, and the covariance is estimated from the decomposed eigenvalues by considering the correlation between the current point and its surrounding points. The estimated covariance makes the update module aware of match quality, allowing it to ignore poorly matched points and focus on well-matched ones, thereby improving odometry accuracy. In addition, it eliminates *Z*-axis drift in long-term odometry estimation by using ground plane information and linearly adjusts pose uncertainty based on optimized pose values. This unified approach to managing uncertainty is essential for the system's long-term stability and accuracy. We thoroughly evaluated our method using multiple publicly available datasets. The experimental results show that our method is accurate and reliable in dynamic urban environments and achieves state-of-the-art LIO performance with fast speed and strong generalization ability. We will release the code of our method here: https://github.com/Gatsby23/UA-LIO.git.**

***Index Terms*— Autonomous driving, iterated error-state Kalman filter (IESKF), light detection and ranging (LiDAR)-inertial odometry (LIO), simultaneous localization and mapping (SLAM), state estimation.**

Received 4 September 2024; revised 28 November 2024; accepted 9 January 2025. Date of publication 21 February 2025; date of current version 7 March 2025. This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant 62273229 and in part by the Science and Technology Commission of Shanghai Municipality under Grant 24TS1402600 and Grant 24TS1402800. The Associate Editor coordinating the review process was Dr. Yan Zhuang. *(Corresponding author: Ling Pei.)*

Qi Wu, Songpengcheng Xia, Guoqing Liu, Wenxian Yu, and Ling Pei are with the School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: ling.pei@sjtu.edu.cn).

Xieyuanli Chen is with the College of Intelligence Science and Technology, National University of Defense Technology, Changsha 410073, China.

Xiangyu Xu is with ByteDance Inc., Beijing 100098, China.

Xinliang Zhong is with Alibaba Group, Hangzhou 311121, China.

Xingwei Qu is with the Department of Computer Science, Technical University of Munich, 80333 München, Germany.

Liu Liu is with the School of Computer Science and Information Engineering, Hefei University of Technology, Hefei 230002, China.

Digital Object Identifier 10.1109/TIM.2025.3544286

I. Introduction

ODOMETRY or ego-motion estimation is a fundamental task for many robotics and autonomous driving applications. It provides the pose information of the robot or vehicle online and plays an important role in downstream tasks, such as simultaneous localization and mapping (SLAM) [1], [2], [3], [4], global localization [5], and path planning [6], [7]. Recently, multisensor fusion approaches have shown advances in the odometry estimation task under urban driving environments. The approaches using inertial measurement unit (IMU) together with 3-D light detection and ranging (LiDAR) sensors have gained a lot of interest due to their accurate and reliable performance under GNSS-denied environments.

The fusion of these two sensors enhances accuracy by compensating for the deficiencies of each modality. The core concept of these algorithms is to use LiDAR data to correct IMU-induced drift. Based on the used constraint function, they are categorized as either loosely coupled or tightly coupled. Loosely coupled strategies estimate the vehicle's pose by integrating independent estimates from the two sensors, with LiDAR-inertial odometry LIO-SAM as a notable example that employs pose graph optimization. In contrast, tightly coupled approaches enhance localization accuracy through direct observations. For example, LIO-Mapping employs optimization techniques, while FAST-LIO2 utilizes filtering methods. Benefiting from the precise sensing capabilities of LiDAR and the high-frequency motion estimation provided by IMU sensors, filter-based LIO systems have become increasingly favored for their ability to maintain high accuracy while meeting low computational demands through correctly matched LiDAR features. Although current LIO algorithms have demonstrated good accuracy, they often overlook the issue of mismatched outliers, which is a common problem in LIO systems.

In practical urban environments, LIO algorithms primarily face two challenges: pose drift caused by mismatched features during the update process, and *Z*-axis drift caused by sparse vertical observations from LiDAR. First, regarding the mismatch issue, researchers generally attribute it to the presence of dynamic objects in the scene, which violate the static environment assumption of SLAM systems, leading to significant trajectory drift. Although deep learning methods have been employed to identify and exclude these dynamic

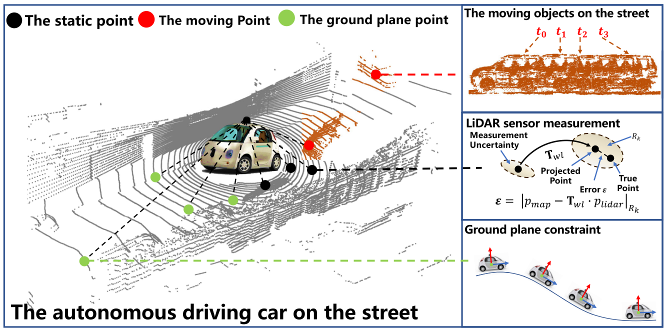


Fig. 1. Odometry estimation in a dynamic urban driving environment. Our method models the uncertainty of each LiDAR point to mitigate mismatches during the matching process, and leveraging the ground plane information to achieve accurate LiDAR odometry results.

objects to reduce mismatches and enhance pose accuracy, the inherent lack of texture information in LiDAR data makes it exceedingly challenging to completely avoid mismatches in complex urban scenarios. Second, concerning the Z-axis drift, some researchers attempt to directly enhance the Z-axis accuracy using ground information provided by LiDAR. However, this approach can lead to inconsistencies in the system state within tightly coupled LIO systems, potentially resulting in decreased system performance or even failure. Thus, effectively tackling these two issues is vital for enhancing the stability and precision of LIO algorithms.

In this article, we have developed an uncertainty-aware LIO algorithm to tackle two major challenges encountered when deploying in urban environments, as shown in Fig. 1. For the first issue, we introduce an update module that incorporates distribution-to-distribution constraints. Each LiDAR point is treated as a sample point on a locally sampled plane, and its relationship with surrounding points is modeled as a Gaussian distribution. By incorporating the uncertainty derived from this distribution into the update module, this method enhances the correction of correctly matched point pairs and reduces the negative impacts of incorrect matches. For the second issue, considering that the Gauss–Newton method and iterative Kalman filtering are equivalent in nonlinear optimization problems, we treat the ground constraint optimization process as an update process of the extended Kalman filter. By analyzing the changes in the pose before and after optimization and the constraint equations in the optimization process, we can approximate the gain matrix during the update process and thus estimate the pose uncertainty after optimization. Adjusting the pose uncertainty effectively enhances the stability of the LIO system, making long-distance stable operation in urban environments possible.

In conclusion, we make three key claims.

1) We propose a novel LIO algorithm system that significantly enhances the localization accuracy over long trajectories in large urban environments. This algorithm will be open-sourced to make a contribution to the autonomous driving algorithm community.
2) Our algorithm addresses the common issue of mismatched features in autonomous driving scenarios by utilizing a proposed covariance estimation module. This module estimates covariance based on the decomposed eigenvalue from a Gaussian distribution of the surrounding geometric information. The proposed module reduces attention to mismatched point pairs and increases attention to correctly matched pairs, thereby enhancing the accuracy of the LIO system.
3) To mitigate z-axis drift over long trajectories in autonomous driving scenarios, we not only use ground plane information to suppress drift but also propose a unified management of pose uncertainty for pose optimization. This unified management is key to maintaining system stability over long-term trajectories.

II. Related Work

In this section, we review works that use IMU, LiDAR, and LiDAR-inertial for the odometry task in real urban environments. Both IMU [8] and LiDAR [2], [9] sensors have been widely used individually in pose estimation for robotics and autonomous driving applications. IMU is small, cheap, and with high sensor frame rates thus has been applied to obtain pose information for mobile systems [8]. It is however influenced by accumulated drift and usually combined with additional sensors such as LiDAR to achieve accurate odometry estimation. Using LiDAR data to estimate the status of the sensor is a classic topic in robotics and autonomous driving. One of the primary methods is the iterative closest point (ICP) algorithm [10]. It aligns two point clouds together and obtains the relative transformation between them. However, the sparsity of the LiDAR data would naturally cause a mismatch during the data association between consecutive scans. To tackle this issue, Zhang and Singh [2] propose LOAM, which uses the point-to-edge and the point-to-plane distance to optimize the ICP error and achieve more accurate odometry estimates. Other variants of LOAM, such as LeGO-LOAM [9], a lightweight and ground-optimized algorithm, utilize ground plane information to inhibit Z-axis drift in a two-step optimization-based pose estimation. This type of method uses only LiDAR data, which is vulnerable to featureless environments.

Combining LiDAR and IMU for odometry becomes the trend for more reliable odometry by fusing the individual advantages and compensating for the disadvantages of each sensor. Several LiDAR-inertial-based odometry methods have been proposed [11], [12] and can be typically divided into two groups: the optimized-based algorithms [13], [14], [15] and the filter-based algorithms [16], [17]. For the optimization algorithm [12], [18], the LIO-SAM proposed by Shan et al. [12] performs the best. It uses the sensor odometry estimates to build the factor graph and correct the pose via pose graph optimization. The filter-based algorithms consist of two steps. They first use IMU propagation result to predict the current status of the sensor system and then use the LiDAR measurements to correct the biases. The FAST-LIO2 by Xu et al. [19] achieves state-of-the-art performance in LIO, which uses the flat features of the scan together with the point-to-plane error distances to correct the biases. Both algorithms rely on the point cloud registration results for updating the pose, which

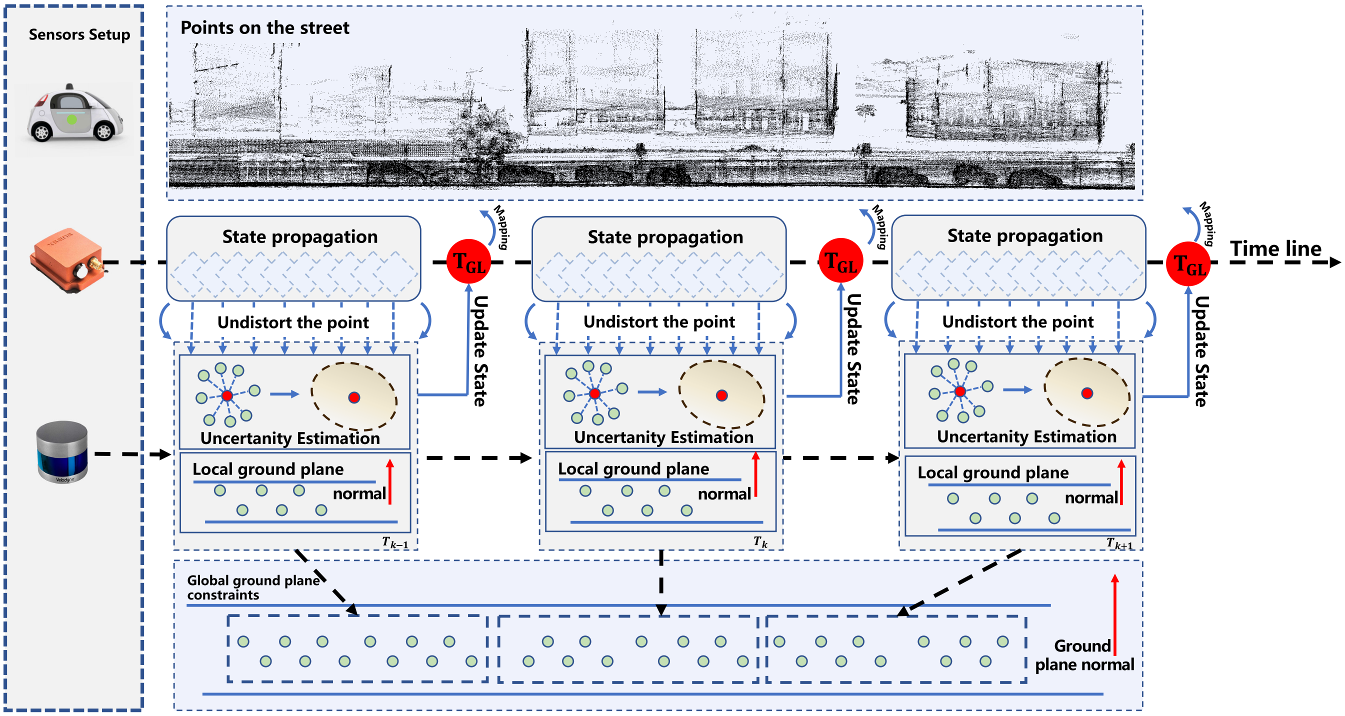


Fig. 2. Overview of the proposed LIO method. Our method is based on the IESKF-based framework to fuse the LiDAR observations with the IMU pose estimates. The main process of our method is to use the proposed covariance to improve the poses' accuracy within the IESKF framework. It then uses the ground plane constraints to eliminate the z-axis drift and further optimize the odometry results.

typically works well in static environments while failing in a dynamic world.

To tackle the problem of mismatches in urban driving scenarios, there are multiple works [20], [21], [22], [23], [24], [25] using semantic information to filter out the dynamic object to improve LiDAR-based pose estimation. For example, the semantic-enhanced LiDAR SLAM by Chen et al. [1] uses the multiclass semantics from a neural network to filter out the moving pedestrians, cars and other potentially moving objects on each LiDAR scan. However, this method relies on the multiclass semantics and may wrongly remove some useful information such as parked cars. To tackle this issue, they later propose LiDAR moving object segmentation [21], [22], [23], [24] that distinctions between the truly moving objects and the potentially moving objects. These methods only use LiDAR data, which may fail in featureless environments. The most related work to ours is by Qian et al. [26], which exploits the moving object semantics to improve LIO results. However, it relies on a hand-crafted model, which may need careful parameter adjustment and its implementation is not publicly available. However, no algorithm can completely eliminate dynamic information in the scene. Even if dynamic information is entirely removed, mismatches will still occur due to the inherent lack of texture in the laser data. There have been some works leveraging covariance to mitigate these issues [27], [28], [29], [30], but most of them struggle to operate effectively over large-scale and long-distance trajectories.

Unlike all existing methods, our approach focuses more on the uncertainty-aware odometry estimation under dynamic urban environments and explicitly exploits ground information to enhance the LIO and obtain more accurate pose estimation.

III. Method

The overview of our system is shown in Fig. 2. Our method uses a filter-based framework (see Section III-A) to fuse the LiDAR observations with the IMU pose estimates. It has two main modules: the uncertainty-aware enhanced odometry estimation module (see Section III-C), and the ground plane optimization module (see Section III-D). We will first review the iterated error-state Kalman filter (IESKF) framework as applied to LiDAR-Inertial systems and provide an in-depth examination of these two specific modules.

A. Background and Notion

Before diving into our proposed method, we first briefly introduce notations used in LIO and the IESKF-based pipeline.

We define the true state $\mathbf{x}$ of the IMU body frame at timestamp k as

$$\mathbf{x}_k = \left[{}^{G}\mathbf{R}_I^\top \;\; {}^{G}\mathbf{P}_I^\top \;\; {}^{I}\mathbf{R}_L^\top \;\; {}^{I}\mathbf{P}_L^\top \;\; {}^{G}\mathbf{v}_I^\top \;\; \mathbf{b}_\omega^\top \;\; \mathbf{b}_a^\top \;\; {}^{G}\mathbf{g}_I^\top \right]^\top. \quad (1)$$

The left superscript represents the coordinate and the right subscript represents the body frame of the current sensor. The G represents the first IMU frame as the global coordinate. ${}^{G}\mathbf{R}_I^\top \in SO(3)$ and ${}^{G}\mathbf{P}_I^\top \in \mathbb{R}^3$ represents the orientation and position of the IMU body frame in the global coordinate. The transformation from the LiDAR frame to the IMU frame is represented as $\left[{}^{I}\mathbf{R}_L \; {}^{I}\mathbf{P}_L \right]$, where the ${}^{I}\mathbf{R}_L \in SO(3)$ and the ${}^{I}\mathbf{P}_L \in \mathbb{R}^3$. The $\mathbf{v}_I$ represents the global velocity of the IMU body. The $\mathbf{b}_w$ and $\mathbf{b}_a$ are the IMU biases modeled as the random walk process, which is further described in Section III-C. The ${}^{G}\mathbf{g}$ is the gravity under the global frame.

Here, we give a brief description of the IESKF. Different from the extended Kalman filter algorithm, the IESKF is the recursive form of the Gauss–Newton algorithm. It defines the propagation of the IMU estimate as the predicted state $\hat{\mathbf{x}}$ with the process noise $\boldsymbol{\mu}$ and the true state $\mathbf{x}$ without the noise disturbance. The measurements of the LiDAR observations are defined as $\mathbf{y}_k = \mathbf{h}(\hat{\mathbf{x}}_k, \zeta)$. So the update step is to minimize the equation below

$$\min_{\mathbf{x}_k} \left|\left|\mathbf{x}_k - \hat{\mathbf{x}}_k\right|\right|^2_{\mathbf{P}_k} + \left|\left|h\left(\hat{\mathbf{x}}_k, \mathbf{0}\right)\right|\right|_{(\mathbf{J}_k \zeta \mathbf{J}_k^{\top})^{-1}}. \tag{2}$$

For the propagation part, the body state at time k is denoted as x_k. The predicted state $\hat{\mathbf{x}}_{k+1}$ at time $k+1$ is $f(\hat{\mathbf{x}}_k, \boldsymbol{\mu})$ and the true state $\mathbf{x}_{k+1}$ is $f(\hat{\mathbf{x}}_k, \mathbf{0})$. The function $f(\cdot)$ represents the kinematic of the nonlinear discrete-time system. So the error state $\delta\mathbf{x}$ at time $k+1$ is represented as

$$\delta\mathbf{x}_{k+1} = \mathbf{x}_{k+1} - \hat{\mathbf{x}}_{k+1} \tag{3}$$

$$\simeq \mathbf{F}_k \delta\mathbf{x}_k + \mathbf{G}_k \tag{4}$$

where the matrix $\mathbf{F}_k$ and the matrix $\mathbf{G}_k$ is the Jacobian matrix of the kinematic model with respect to the error state and the noise vector.

The covariance is updated as

$$\hat{\mathbf{P}}_{k+1} = \mathbf{F}_k \mathbf{P}_k \mathbf{F}_k^{\mathrm{T}} + \mathbf{G}_k \mathbf{Q} \mathbf{G}_k^{\mathrm{T}} \tag{5}$$

where the $\mathbf{Q}$ is the process noise covariance matrix.

For the update part, the linearization point is $\mathbf{x}_k^i$ at each iterations and the start point $\mathbf{x}_{k+1}^0 = \hat{\mathbf{x}}_{k+1}$. The right superscript represents each iteration i. So we get

$$\mathbf{K}_{k+1}^i = \left(\hat{\mathbf{P}}_{k+1}^{-1} + \mathbf{H}^{\mathrm{T}} \zeta^{-1} \mathbf{H}\right)^{-1} \mathbf{H}^{\top} \zeta^{-1} \tag{6}$$

$$\delta\mathbf{x}_{k+1}^i = \mathbf{K}_{k+1}^i \left(\mathbf{H}_k^i \left(\delta\mathbf{x}_{k+1}^i\right) - \mathbf{h}(\mathbf{x}_k, \mathbf{0})\right) \tag{7}$$

$$\mathbf{x}_{k+1}^{i+1} = \hat{\mathbf{x}}_{k+1}^{i+1} + \delta\mathbf{x}_{k+1}^{i+1} \tag{8}$$

where the $\mathbf{H}$ is the Jacobian of the error state at each linearization point. After convergence, the covariance is

$$\mathbf{P}_{k+1} = (\mathbf{I} - \mathbf{K}\mathbf{H})\hat{\mathbf{P}}_{k+1} \tag{9}$$

where $\mathbf{P}_{k+1}$ and $\mathbf{x}_{k+1}$ are then used as the initial guess for the next iteration.

B. Distribution-Based Awareness of Point Uncertainties

In this section, we will provide a detailed description of how to make each LiDAR point-matched pair to be aware of its uncertainty. Inspired by [28] and [31], we assume that each LiDAR point is sampled at the surface as the center of a local plane. The position of each LiDAR point is the center of this plane and we describe this sampling as a Gaussian distribution: ${}^{L}\mathbf{p}_i \sim \mathcal{N}({}^{L}\mathbf{p}_i, \Sigma_i)$. The covariance matrix Σ_i describes the correlation between the current point and its 20 surrounding points. As shown in Fig. 3, considering that each point is a sample of a local plane on an object's surface, the correct paired points should have the same plane normal. More specifically, the uncertainty along the plane normal vector should be lower. So we normalize the covariance matrix by replacing its eigenvalue with $(1, 1, \epsilon)$, where ϵ is a small value (we set it as $1\mathrm{e}^{-3}$ in this article).

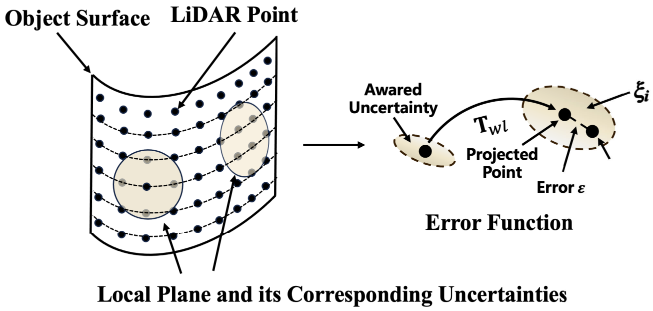


Fig. 3. Estimation of uncertainty for each LiDAR point.

C. Uncertainty-Aware-Enhanced Odometry Estimation

There are two steps in our filter-based odometry method, odometry propagation from the IMU and pose correction using the LiDAR observation. We build our method upon the IESKF framework. We inherit the pose propagation part but change the pose correction by integrating our aware uncertainties into the updation modules. Different from existing methods that use point-to-plane error metrics in updation modules, our approach utilizes distribution-to-distribution error metrics, incorporating uncertainties derived from Section III-B, to achieve more robust results.

We select the ith point from the current scan as ${}^{L}\mathbf{p}_i$, and $\mathbf{\Sigma}_i$ is its corresponding uncertainty. The associated map point in the online built local map is represented as ${}^{G}\mathbf{p}_j$. The observation function $\mathbf{y}_{p_i}$ is defined as follows:

$$\begin{aligned} \mathbf{y}_{p_i} &= {}^{G}\mathbf{p}_j - {}^{G}\mathbf{T}_I \cdot {}^{I}\mathbf{T}_L \cdot {}^{L}\mathbf{p}_i \\ \mathbf{y}_{p_i} &\sim \left(0, \mathbf{\Sigma}_j + {}^{G}\mathbf{T}_I {}^{I}\mathbf{T}_L \mathbf{\Sigma}_i {}^{I}\mathbf{T}_L^{\top} {}^{G}\mathbf{T}_I^{\top}\right) \end{aligned} \tag{10}$$

where ${}^{G}\mathbf{T}_I$ is the matrix form of the $[{}^{G}\mathbf{R}_I^{\top}, {}^{G}\mathbf{P}_I^{\top}]$ and the ${}^{I}\mathbf{T}_L$ is the matrix form of the $[{}^{I}\mathbf{R}_L^{\top}, {}^{I}\mathbf{P}_L^{\top}]$. We use the KD-Tree to do the association, and align the current scan with the online built local point cloud map.

Combined the uncertainty from (10), the ith row the matrix $\mathbf{H}$ derived from the ith residual is calculated as

$$\mathbf{H}_i = \boldsymbol{\xi_i} \left[{}^{G}\mathbf{R}_I^{\top} \left[{}^{I}\mathbf{p}\right]_{\times} \quad \mathbf{I}_{3\times3} {}^{G}\mathbf{R}_I^{\top} {}^{I}\mathbf{R}_L^{\top} \left[{}^{L}\mathbf{p}\right]_{\times} \quad {}^{G}\mathbf{R}_I^{\top} \right]. \tag{11}$$

The matrix $\boldsymbol{\xi_i}$ is the result of performing Cholesky decomposition on the matrix $(\mathbf{\Sigma}_j + {}^{G}\mathbf{T}_I {}^{I}\mathbf{T}_L \mathbf{\Sigma}_i {}^{I}\mathbf{T}_L^{\top} {}^{G}\mathbf{T}_I^{\top})$. By substituting $\mathbf{H}$ to (6)–(9), we obtain an enhanced odometry estimates.

D. Ground Plane Optimization

Besides using the aware uncertainties for rejecting the outliers of paired features to improve the pose correction, our method further exploits the ground plane information to eliminate the z-axis drift in a back-end manner. Inspired by the works [32], [33], we built the ground-plane constraints upon the pose-graph optimization in the same sliding window $\{\mathcal{S}_{k-N}, \ldots, \mathcal{S}_{k-i} \ldots, \mathcal{S}_k\}$. We assume that autonomous cars

TABLE I
IMU PARAMETERS OF URBANNAV AND ECMD DATASETS

Parameter	Value
IMU rate	400 [HZ]
Gyroscope noise density	1.0270904839 × 10^{-02} [rad/s/$\sqrt{\text{Hz}}$]
Gyroscope random walk	9.1355383994 × 10^{-05} [rad/s^2/$\sqrt{\text{Hz}}$]
Accelerometer noise density	1.1197412605 × 10^{-02} [m/s^2/$\sqrt{\text{Hz}}$]
Accelerometer random walk	1.1751767903 × 10^{-04} [m/s^3/$\sqrt{\text{Hz}}$]
Magnitude of gravity	9.805 [m/s^2]

TABLE II
IMU PARAMETERS USED IN REAL-WORLD CAMPUS-LIKE SCENARIOS

Parameter	Value
IMU rate	120 [HZ]
Gyroscope noise density	1.5717 × 10^{-04} [rad/s/$\sqrt{\text{Hz}}$]
Gyroscope random walk	1 × 10^{-05} [rad/s^2/$\sqrt{\text{Hz}}$]
Accelerometer noise density	4.1893 × 10^{-04} [m/s^2/$\sqrt{\text{Hz}}$]
Accelerometer random walk	1.932 × 10^{-05} [m/s^3/$\sqrt{\text{Hz}}$]
Magnitude of gravity	9.805 [m/s^2]

TABLE III
IMU PARAMETERS USED IN SIMULATED SCENARIOS

Parameter	Value
IMU rate	120 [HZ]
Gyroscope noise density	1 × 10^{-03} [rad/s/$\sqrt{\text{Hz}}$]
Gyroscope random walk	1 × 10^{-05} [rad/s^2/$\sqrt{\text{Hz}}$]
Accelerometer noise density	1 × 10^{-02} [m/s^2/$\sqrt{\text{Hz}}$]
Accelerometer random walk	1 × 10^{-04} [m/s^3/$\sqrt{\text{Hz}}$]
Magnitude of gravity	9.805 [m/s^2]

always touch to the ground, and that the ground is always flat without significant fluctuations. The constraint function can be defined as $\mathbf{y}_{\text{op}_i} = \mathbf{y}_{\text{gp}_i} + \mathbf{y}_{\text{po}_i}$. Within the past ith scan $\mathcal{S}_{k-i}$ in the sliding window, we defined the ground plane constraint function $\mathbf{y}_{\text{gp}_i}$ as

$$
\begin{aligned}
{}^{k-i}\mathbf{n}^{\top} &= {}^{k-i}\mathbf{R}_{k-N} \cdot {}^{k-N}\mathbf{n}^{\top} && (12)\\
{}^{k-i}d &= {}^{k-N}d -{}^{k-N}\,\mathbf{P}_{k-i} \cdot{}^{k-i}\,\mathbf{n}^{\top}. && (13)
\end{aligned}
$$

The minimal parameterization of the ground plane normal information is: $\boldsymbol{\tau}(\pi) = \begin{bmatrix}\phi & \psi & d\end{bmatrix}$ and for each scan $\mathcal{S}_{k-i}$, it is represented as

$$
\begin{aligned}
\boldsymbol{\tau}(\pi)_{k-i} &= \left[\arctan\left(\tfrac{n_y}{n_x}\right), \arctan\left(\tfrac{n_z}{|\boldsymbol{n}|}\right), d\right] && (14)\\
\mathbf{y}_{\text{gp}_i} &= \boldsymbol{\tau}(\pi)_{k-N} - \boldsymbol{\tau}(\pi)_{k-i} && (15)
\end{aligned}
$$

which projects the global plane information into the $(k-i)$th coordinate, aligns the z-axis to reduces the drift. The $\mathbf{y}_{\text{po}_i} = [\mathbf{y}_{\theta}, \mathbf{y}_{\text{P}}]$ is defined as follows:

$$
\begin{aligned}
\mathbf{y}_{\theta} &= {}^{I_{i-1}}\mathbf{R}_{I_i}^{-1}\, {}^{G}\mathbf{R}_{I_{i-1}}^{-1}\, {}^{G}\mathbf{R}_{I_i} && (16)\\
\mathbf{y}_{\text{P}} &= {}^{I_{i-1}}\mathbf{P}_{I_i} - {}^{G}\mathbf{R}_{I_{i-1}}^{\top}\left({}^{G}\mathbf{P}_{I_i} - {}^{G}\mathbf{P}_{I_{i-1}}\right). && (17)
\end{aligned}
$$

In order to calculate the covariance matrix corresponding to the optimized pose, we treat the optimized pose as the true state $\mathbf{x}$ and the preoptimized pose as the predicted state $\hat{\mathbf{x}}$ in Section III-A and get the error state $\delta\mathbf{x}$ thorough (4). The corresponding covariance $\hat{\mathbf{P}}_{k+1}$ is the concatenation of the covariances corresponding to each preoptimized pose in the sliding window. The linearized matrix $\mathbf{H}$ is defined as $\mathbf{H} = [\mathbf{H}_{\text{gp}}^{\top}\ \mathbf{H}_{\theta}^{\top}\ \mathbf{H}_{\text{P}}^{\top}]^{\top}$. The $\mathbf{H}_{\text{gp}}$ is

$$
\begin{aligned}
\mathbf{H}_{\text{gp}} &= \begin{bmatrix}\mathbf{0}_{3\times 6} & \cdots & \dfrac{\partial \mathbf{y}_{\text{gp}}}{\partial \mathbf{R}} & \dfrac{\partial \mathbf{y}_{\text{gp}}}{\partial \mathbf{P}}\end{bmatrix} && (18)\\
\mathbf{H}_{\theta} &= \begin{bmatrix}\mathbf{0}_{3\times 6} & \cdots & \mathbf{I}_{3\times 3} & \mathbf{0}_{3\times 3}\end{bmatrix} && (19)\\
\mathbf{H}_{\text{P}} &= \begin{bmatrix}\mathbf{0}_{3\times 6} & \cdots & \mathbf{0}_{3\times 3} & {}^{I_{i-1}}\mathbf{R}_{G}\end{bmatrix}. && (20)
\end{aligned}
$$

In order to simplify the equation, we denote the symbol ${}^{k-i}\mathbf{n}$ as $\mathbf{n}'$ and the ${}^{k-i}d$ as d'. According to the chain the rule, the $(\partial \mathbf{y}_{\text{gp}}/\partial \mathbf{R})$ is (21) and (22), as shown at the bottom of the next page.

The $(\partial \mathbf{y}_{\text{gp}}/\partial \mathbf{P})$ is

$$
\frac{\partial \mathbf{y}_{\text{gp}}}{\partial \mathbf{P}} = \frac{\partial \mathbf{y}_{\text{gp}}}{\partial d'}\frac{\partial d'}{\partial \mathbf{P}} = \begin{bmatrix} 0 & 0 & 0\\ 0 & 0 & 0\\ -n'_x & -n'_y & -n'_z \end{bmatrix}. \qquad (23)
$$

By substituting the $\mathbf{H}$, $\delta\mathbf{x}$, and the $\hat{\mathbf{P}}_{k+1}$ into (6)–(9), we obtain the consistent covariance of the optimized state. From practical testing, the rotation estimation itself is relatively accurate, so we only adjusted the position in the optimization process.

$$
\begin{aligned}
\frac{\partial \mathbf{y}_{\text{gp}}}{\partial \mathbf{R}} &= \frac{\partial \mathbf{y}_{\text{gp}}}{\partial \mathbf{n}'}\frac{\partial \mathbf{n}'}{\partial \mathbf{R}} && (21)\\
&= \underbrace{\begin{bmatrix} -\dfrac{n'_y}{n'^2_x + n'^2_y} & \dfrac{n'_x}{n'^2_x + n'^2_y} & 0\\ -\dfrac{n'_z n'_x}{\left(n'^2_x + n'^2_y + n'^2_z\right)^{\frac{3}{2}}} & -\dfrac{n'_z n'_y}{\left(n'^2_x + n'^2_y + n'^2_z\right)^{\frac{3}{2}}} & \dfrac{\left|\mathbf{n}'\right|^2 - n'^2_z}{\left(n'^2_x + n'^2_y + n'^2_z\right)^{\frac{3}{2}}}\\ -\mathbf{P}_x & -\mathbf{P}_y & -\mathbf{P}_z \end{bmatrix}}_{\frac{\partial \mathbf{y}_{\text{gp}}}{\partial \mathbf{n}'}}\\
&\quad -\underbrace{\mathbf{R}[\mathbf{n}]_{\times}}_{\frac{\partial \mathbf{n}'}{\partial \mathbf{R}}} && (22)
\end{aligned}
$$

TABLE IV
LIO RESULTS COMPARISON ON THE ECMD DATASET

Algorithms	Easy (620.078m)				Medium (805.216m)				Difficult (153.147m)			
	RMSE (m)	MAX (m)	MEAN (m)	STD (m)	RMSE (m)	MAX (m)	MEAN (m)	STD (m)	RMSE (m)	MAX (m)	MEAN (m)	STD (m)
DLO*	1.374	2.734	1.243	0.586	2.401	5.133	2.117	1.132	0.517	2.951	0.405	0.321
FAST-LIO2	1.519	2.650	1.409	0.565	1.749	3.044	1.643	0.599	0.576	2.928	0.413	0.401
FASTER-LIO	1.650	3.019	1.533	0.610	1.988	3.634	1.850	0.727	0.574	2.921	0.410	0.401
VoxelMap++	-	-	-	-	1.626	3.874	1.491	0.647	0.624	**2.851**	0.481	0.398
Point-LIO	1.263	2.661	1.134	0.555	1.965	4.125	1.698	0.988	0.562	2.898	0.399	0.395
PV-LIO	2.199	4.318	1.952	1.013	1.608	2.932	1.470	0.649	0.620	2.943	0.470	0.404
LIO-SAM	1.145	2.661	1.066	0.416	1.704	3.726	1.530	0.751	0.565	2.893	0.402	0.397
UA-LIO	**0.952**	**1.754**	**0.905**	**0.296**	**1.547**	**2.900**	**1.454**	**0.526**	**0.468**	2.969	**0.347**	**0.314**

Note: **Bold** values indicate the best result per column. ”-” denotes cases where the algorithm did not complete the course (drifted).

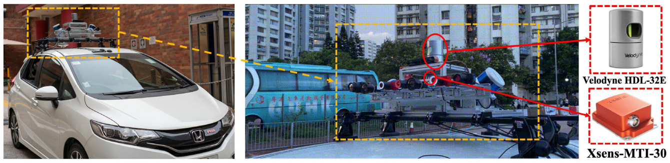


Fig. 4. Autonomous driving suite used in the UrbanNav and the ECMD dataset, including a Velodyne HDL-32E LiDAR sensor and an Xsens MTI-30 IMU.

IV. EXPERIMENTAL EVALUATION

Our experimental results are designed to support our claims that our method.

1) Achieving state-of-the-art performance in real-world urban autonomous driving environments, especially for the long-term journey trajectory and demonstrating good generalization across different scenarios.
2) Successfully reducing the impact of mismatches on the accuracy of LiDAR odometry through aware uncertainties.
3) Effectively suppressing the Z-axis drift issue and ensuring long-term operability by adjusting the uncertainty corresponding to the optimized pose.

The experimental results in this article well support these viewpoints.

A. Datasets and Experimental Setups

To comprehensively evaluate our method, we conducted extensive tests using public datasets, self-collected datasets, and simulation datasets. For the public datasets, we tested our algorithm on the UrbanNav [34] and ECMD [35] datasets, both of which were collected in Hong Kong at different times using the same autonomous driving suite. As shown in Fig. 4, this suite includes a LiDAR sensor (Velodyne-32) and an IMU (MTI-300). As mentioned earlier, these datasets contain a significant number of dynamic objects, reflecting common scenarios in autonomous driving, which pose substantial challenges to the LIO algorithm. However, the difficulty of each dataset varies: the ECMD dataset contains a large number of pedestrians, and since it is currently impossible to completely eliminate dynamic objects through deep learning, this dataset effectively tests our algorithm's robustness against mismatches caused by dynamic objects. On the other hand, the UrbanNav dataset, with numerous dynamic vehicles and driving distances exceeding 3 km, allows us to validate the robustness of our algorithm over long distances. Ground truths for all these datasets were generated using high-precision inertial navigation devices integrated with GPS positioning.

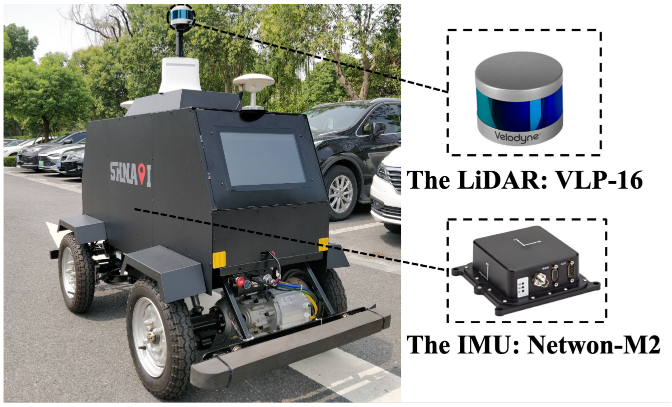


Fig. 5. Platform we used in the campus environment.

To validate the generalization capability of our algorithm and its performance in real-world scenarios, we conducted data collection and algorithm testing using a self-collected dataset on the campus of Shanghai Jiao Tong University, as shown in Fig. 5. We collected low dynamic, medium dynamic, and high dynamic sequences based on the varying pedestrian traffic at different times on campus. The final test results demonstrate that our algorithm performs exceptionally well in real-world road environments, exhibiting strong generalization capabilities.

In the following experiments, we will compare our proposed method with various open-sourced LiDAR-inertial systems. Considering that our proposed LIO core module primarily consists of point-to-point constraints and the covariance estimation, we will compare it with similar algorithms such as Point-LIO [36], VoxelMap++ [29], and PV-LIO [37]. Point-LIO is based on the point-to-point constraint, while VoxelMap++ and PV-LIO also consider covariance to enhance localization quality. In addition, we will compare it with the classical methods, particularly for long-term journeys, like Direct LiDAR Odometry [28], a loosely coupled LiDAR-inertial algorithm; FAST-LIO2 [11], a tightly coupled IESKF-based sensor fusion algorithm and its more efficient variant [16]; and LIO-SAM [12], an optimization-based LIO algorithm. For all comparative algorithms, we adjusted the

TABLE V
LIO RESULTS COMPARISON ON THE REAL-WORLD ENVIRONMENTS

Algorithms	Low				Middle				High			
	RMSE	MAX	MEAN	STD	RMSE	MAX	MEAN	STD	RMSE	MAX	MEAN	STD
DLO	0.817	**2.368**	0.677	0.457	1.113	3.038	0.915	0.634	0.922	2.012	**0.808**	0.444
Fast-LIO2	0.890	2.416	0.816	0.355	0.873	**2.337**	0.791	**0.371**	1.036	2.669	0.900	0.513
Faster-LIO	1.015	2.254	0.912	0.445	drift	drift	drift	drift	drift	drift	drift	drift
Point-LIO	0.890	2.554	0.823	**0.355**	1.523	3.802	1.322	0.756	0.680	1.408	0.650	**0.200**
UA-LIO	**0.683**	2.512	**0.454**	0.389	**0.707**	2.395	**0.587**	0.393	**0.622**	**1.183**	0.575	0.237

Note: **Bold** values indicate the best result per row. ”drift” denotes cases where the algorithm did not complete the course (drifted).

TABLE VI
LIO RESULTS COMPARISON ON THE URBANNAV DATASET

Algorithms	Medium (3641.810m)				Deep (4506.624m)			
	RMSE (m)	MAX (m)	MEAN (m)	STD (m)	RMSE (m)	MAX (m)	MEAN (m)	STD (m)
DLO*	5.467	11.554	4.751	2.704	-	-	-	-
FAST-LIO2	8.321	15.815	7.451	3.704	6.165	14.827	5.448	2.885
FASTER-LIO	10.002	21.081	8.947	4.470	6.810	16.669	6.089	3.069
Point-LIO	10.955	23.577	10.017	4.434	13.608	33.419	11.282	7.607
VoxelMap++	-	-	-	-	-	-	-	-
PV-LIO	-	-	-	-	-	-	-	-
LIO-SAM	10.578	16.521	6.121	3.897	10.500	27.793	8.009	6.790
UA-LIO	**2.591**	**3.014**	**1.554**	**0.532**	**3.902**	**7.275**	**1.885**	**1.102**

Note: **Bold** values indicate the best result per column. ”-” denotes cases where the algorithm did not complete the course (drifted).

extrinsic parameters between the LiDAR and IMU and the intrinsic parameters of the IMU for different datasets, setting them as outlined in Tables I–III, while maintaining the other parameters at their default values. The experimental results presented in Tables IV–VI demonstrate the state-of-the-art performance of our algorithm.

We follow these methods to use the absolute trajectory error (APE) to evaluate the algorithms' performance. We selected four metrics, RMSE, MEAN, MAX, and STD, to evaluate the algorithms, and utilized the open-sourced tool evo [38] for computation. We run all the algorithms on an Intel i9 13900KF@3.0 GHZ with 32 cores with 32-GB RAM and an Nvidia RTX 4090 with 24-GB RAM.

B. LIO Performance

In the first experiment, we show the comparison of LIO performance between our proposed method with other LIO methods on the ECMD dataset [34] and UrbanNav dataset [34], which supports that our proposed method achieves the state-of-the-art LIO performance and generalizes well to different environments.

1) ECMD Dataset: For the ECMD dataset, we show the LIO results on three sequences in Table IV. The trajectories in the Dense street part, while not long, contain numerous dynamic objects, representing the primary challenge faced by LIO algorithms in urban bustling scenarios. Based on the number of dynamic objects, we selected three sequences from the dataset: Dense street night easy e, Dense street day medium circle a, and Dense street day difficult. These were used to test our algorithm and are simplified to easy, medium, and difficult in the following. Since the dataset involves bumpy road conditions, which cause rapid variations in ground information that contradict the flat terrain assumption presented in Section III-D, we have frozen the ground optimization module in this case.

As shown in Table IV, the RMSE values indicate that our algorithm performs the best in all sequences and the STD values indicate that our algorithm is more robust than other algorithms. Although the MAX value of our algorithm performance on the difficult sequence is slightly inferior to other algorithms, considering that this metric is primarily used to evaluate the algorithm performance in the worst case scenario and its stability, our result is only slightly lower than the optimal algorithm, indicating that our algorithm remains stable even in extreme situations without crashing. Fig. 6 provides a more intuitive display that our algorithm is better than others. This robustness is primarily attributed to the uncertainty-aware distribution-to-distribution update module, which effectively handles challenging urban scenarios with dynamic objects and frequent mismatches.

2) UrbanNav Dataset: We also provide the quantitative analysis of LIO performance for different algorithms

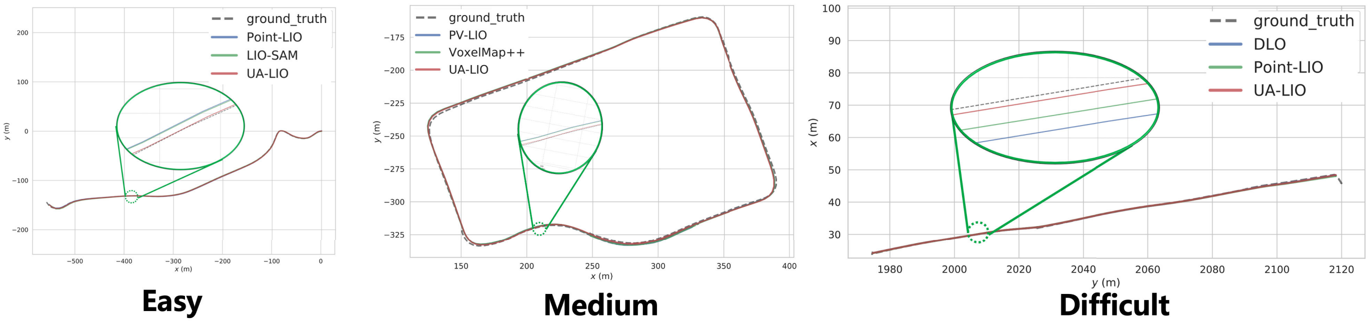


Fig. 6. xy-trajectory qualitative results of the top three ranked LIO algorithms on ECMD datasets.

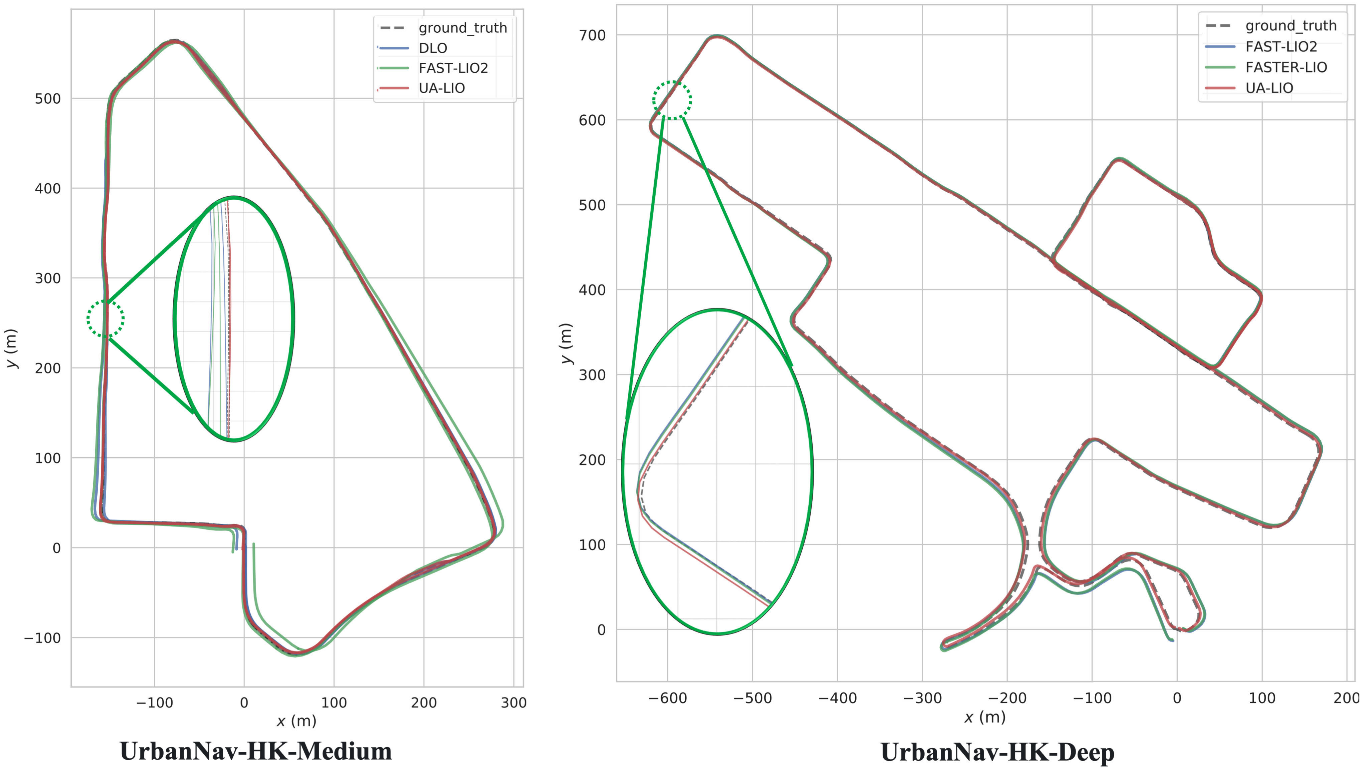


Fig. 7. xy-trajectory qualitative results of the top three ranked LIO algorithms on the UrbanNav datasets.

in Table VI. The UN-hk-medium sequence covers a typical urban canyon of Hong Kong. Its involves high-rise buildings, numerous dynamic objects, which is a challenge to the LiDAR sensors. The UN-hk-deep sequence was collected in a narrow street with numerous moving objects while fewer static landmarks. To better visualize and understand the results, we show the xy-trajectory results in Fig. 7 and z-trajectory results in Figs. 8 and 9. As can be seen in Fig. 7, our method generates accurate trajectories consistently in all four sequences. Especially in the UrbanNav hk-meidum sequence, the odometry trajectory estimated by our approach is very close to ground truths even when other methods fail due to the moving objects. The z-trajectory results in Figs. 8 and 9 shows that benefiting from the devised ground-plane optimization, our method has much less error in the z-axis compared to other methods.

3) Real World Dataset: As shown in Table V, we conducted data collection in real road environments on the campus of Shanghai Jiao Tong University. Unlike the 32-line LiDAR sensor used in previous public datasets, we employed a 16-line LiDAR sensor and categorized the collected sequences into low-traffic, medium-traffic, and high-traffic sequences based on the number of dynamic objects. The experimental results demonstrate that our algorithm exhibits strong generalization capabilities and performs excellently across various real-world scenarios.

C. Ablation Study

In this section, we primarily discuss the impact of different modules on the performance of the algorithm. Specifically, we analyze the uncertainty-awared module in the odometry estimation module and the importance of the covariance adjustment after the ground plane optimization during long-term tracking. The prior module is to effectively reduce trajectory drift caused by mismatches, while the latter ensures

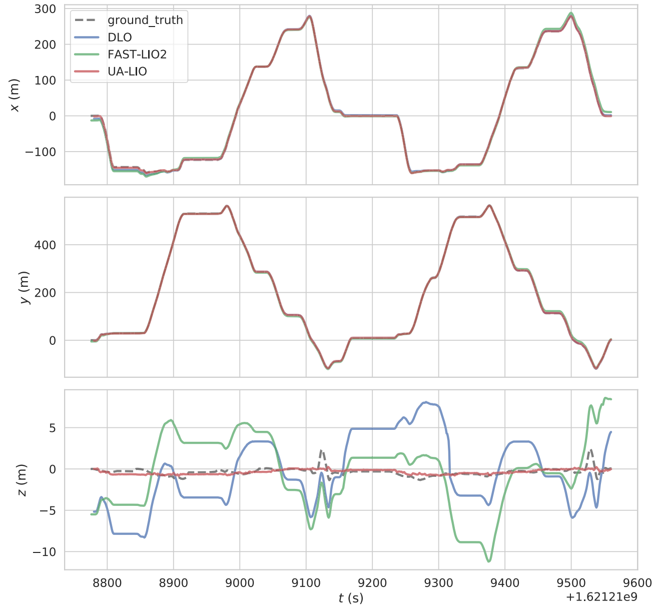


Fig. 8. Results of xyz-axis in the UrbanNav medium sequence.

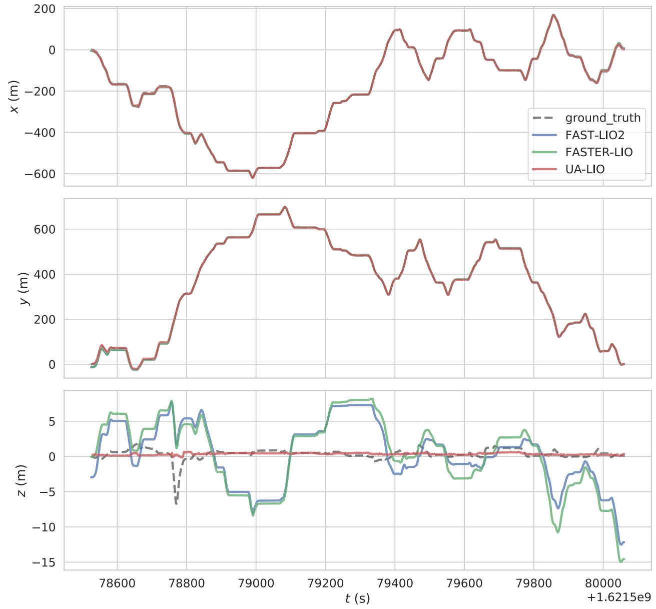


Fig. 9. Results of xyz-axis in the UrbanNav deep sequence.

the stability of the algorithm during long-distance operations. To validate these points, we used various datasets. To disable the uncertainty-aware module, we set $\boldsymbol{\xi_i}$ in (11) to an identity matrix, indicating that all matches contribute equally to the IMU bias update. For validating the covariance adjustment module, we referenced the loosely coupled approach commonly used in integrated navigation, scaling the covariance at the pose correspondences by a factor of ten, either increasing or decreasing it, represented as Ground† and Ground*, respectively.

To evaluate the impact of different modules on the algorithm's performance, we conducted a comprehensive ablation study. The proposed algorithm is designated as UA-LIO. To facilitate the ablation study, we define specific variants:

TABLE VII
ABLATION STUDY ON THE PROPOSED MODULES

COV	Ground	Ground†	Ground*	APE (m)
✗	✗	✗	✗	7.995
✗	✓	✗	✗	8.209
✗	✗	✓	✗	-
✗	✗	✗	✓	-
✓	✗	✗	✗	3.894
✓	✓	✗	✗	**2.481**

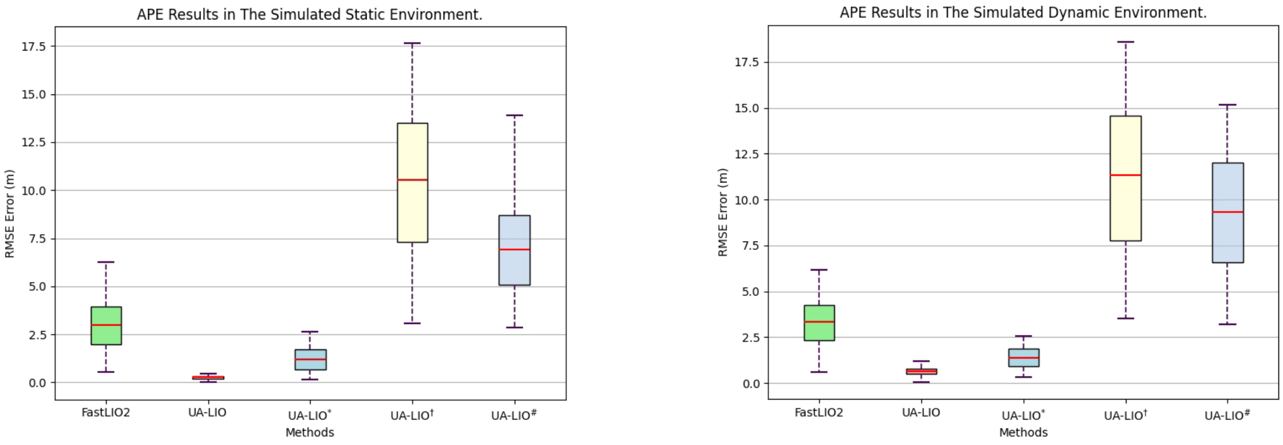
UA-LIO* represents the algorithm incorporating only the covariance estimation module, UA-LIO† denotes the version including only the ground constraint module, and UA-LIO$^\#$ refers to the configuration where both modules are excluded. We had the vehicle follow the exact same trajectory in both purely static and dynamic scenarios. The presence of dynamic objects increases the likelihood of mismatches, which further highlights the advantages of our module. To achieve this, we conducted simulation-based tests using the AirSim simulator, as depicted in Fig. 10, which allowed the vehicle to navigate through the same urban area under both conditions. The noise parameters of the IMU in the simulation are shown in Table III. As shown in Fig. 11, the test results indicate that even in purely static scenarios, the comparison algorithm FASTLIO2 still exhibits significant drift. This supports the point we claimed in Section I: mismatches are not caused only by dynamic objects. The inherent limitation of LiDAR point, capturing only positional information without texture details, also leads to mismatches. The uncertainty-aware distribution-to-distribution update module, however, can effectively reduce the impact of such mismatches.

Table VII demonstrates that both the uncertainty-aware distribution-to-distribution update module and ground constraints significantly improve the odometry accuracy of the LIO algorithm in urban scenarios. However, in the absence of uncertainty considerations, the inclusion of the ground constraint module results in inferior performance compared to its exclusion. This phenomenon can be attributed to the dependency of ground constraints on the accuracy of the initial pose estimation. Without the incorporation of uncertainty information, estimated poses lack sufficient precision, thereby undermining the effectiveness of the ground constraint module and leading to suboptimal outcomes.

D. Runtime Results

In this section, we demonstrate that our algorithm is capable of real-time operation. The average processing time is obtained by measuring the execution time across multiple trials on real environment datasets to ensure robustness and accuracy in our performance evaluation. The most time-consuming components of our method are the ground normal estimation module, the covariance estimation module, and the update module, while the propagation step requires only 2 ms. For the covariance estimation module, we leverage OpenMP for parallel processing, reducing the computation time to approximately

Fig. 10. Simulated scenarios created in AirSim with added dynamic vehicles.



(a)

(b)

Fig. 11. Comparison of APE in simulated environments. (a) Simulated static environment. (b) Simulated dynamic environment.

2 ms per scan. The ground estimation module takes around 7 ms per scan. The distribution-to-distribution update module averages 45 ms per scan. As a result, the total average processing time of our algorithm is 56 ms, which is faster than the frame rate of the LiDAR sensor, ensuring real-time performance.

V. Conclusion

In this article, we presented a novel uncertainty-aware LIO method for autonomous driving in urban environments. Our method leverages the proposed odometry combined with our uncertainty estimation module to filter out mismatched features and utilize ground plane information, thereby enhancing LIO performance. We evaluated our approach across various datasets and compared it with other existing methods. The experimental results suggest that our algorithm achieves state-of-the-art performance in terms of odometry accuracy and generalizes well across different urban environments.

In the future, we will explore more robust LiDAR odometry estimation algorithms and work toward developing a more universal ground model to handle variations in ground information, thereby enabling the application of ground constraints under changing terrain conditions.

References

[1] X. Chen, A. Milioto, E. Palazzolo, P. Giguère, J. Behley, and C. Stachniss, "SuMa++: Efficient LiDAR-based semantic SLAM," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Nov. 2019, pp. 4530–4537. [Online]. Available: http://www.ipb.uni-bonn.de/wp-content/papercite-data/pdf/chen2019iros.pdf

[2] J. Zhang and S. Singh, "LOAM: LiDAR odometry and mapping in real-time," in *Proc. Robot., Sci. Syst. X*, Jul. 2014, pp. 1–9.

[3] J. Wang, Z. Meng, and L. Wang, "Efficient probabilistic approach to range-only SLAM with a novel likelihood model," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–12, 2021.

[4] J. Chang, N. Dong, and D. Li, "A real-time dynamic object segmentation framework for SLAM system in dynamic scenes," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–9, 2021.

[5] X. Chen, T. Läbe, L. Nardi, J. Behley, and C. Stachniss, "Learning an overlap-based observation model for 3D LiDAR localization," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2020, pp. 4602–4608. [Online]. Available: https://www.ipb.uni-bonn.de/pdfs/chen2020iros.pdf

[6] Y. Wang et al., "Elastic and efficient LiDAR reconstruction for large-scale exploration tasks," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2021, pp. 5035–5041.

[7] Y.-Q. Liu, F. Jin, K.-F. Dong, J.-L. Song, W.-Q. Mo, and Y.-J. Hui, "Eccentric optimization of multisensor for SLAM-integrated navigation," *IEEE Trans. Instrum. Meas.*, vol. 72, pp. 1–8, 2023.

[8] M. Kok, J. D. Hol, and T. B. Schön, "Using inertial sensors for position and orientation estimation," *Found. Trends Signal Process.*, vol. 11, pp. 1–153, Nov. 2017, doi: 10.1561/2000000094.

[9] T. Shan and B. Englot, "LeGO-LOAM: Lightweight and ground-optimized LiDAR odometry and mapping on variable terrain," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 4758–4765.

[10] P. J. Besl and N. D. McKay, "A method for registration of 3D shapes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 2, pp. 239–256, Feb. 1992. [Online]. Available: http://www-evasion.inrialpes.fr/people/Franck.Hetroy/Teaching/ProjetsImage/2007/Bib/besl_mckay-pami1992.pdf

[11] W. Xu and F. Zhang, "FAST-LIO: A fast, robust LiDAR-inertial odometry package by tightly-coupled iterated Kalman filter," *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 3317–3324, Apr. 2021.

[12] 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. (IROS)*, Oct. 2020, pp. 5135–5142.

[13] H. Ye, Y. Chen, and M. Liu, "Tightly coupled 3D LiDAR inertial odometry and mapping," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2019, pp. 3144–3150.

[14] J. Zhang, W. Wen, F. Huang, X. Chen, and L.-T. Hsu, "Coarse-to-fine loosely-coupled LiDAR-inertial odometry for urban positioning and mapping," *Remote Sens.*, vol. 13, no. 12, p. 2371, Jun. 2021.

[15] P. Lyu, B. Wang, J. Lai, S. Bai, M. Liu, and W. Yu, "A factor graph optimization method for high-precision IMU based navigation system," *IEEE Trans. Instrum. Meas.*, vol. 72, 2023, Art. no. 9509712.

[16] 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 Robot. Autom. Lett.*, vol. 7, no. 2, pp. 4861–4868, Apr. 2022.

[17] C. Qin, H. Ye, C. E. Pranata, J. Han, S. Zhang, and M. Liu, "LINS: A LiDAR-inertial state estimator for robust and efficient navigation," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2020, pp. 8899–8906.

[18] S. Liang, Z. Cao, C. Wang, and J. Yu, "Hierarchical estimation-based LiDAR odometry with scan-to-map matching and fixed-lag smoothing," *IEEE Trans. Intell. Vehicles*, vol. 8, no. 2, pp. 1607–1623, Feb. 2023.

[19] 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.

[20] N. Li, Y. Yao, X. Xu, Y. Peng, Z. Wang, and H. Wei, "An efficient LiDAR SLAM with angle-based feature extraction and voxel-based fixed-lag smoothing," *IEEE Trans. Instrum. Meas.*, vol. 73, pp. 1–13, 2024.

[21] X. Chen et al., "Moving object segmentation in 3D LiDAR data: A learning-based approach exploiting sequential data," *IEEE Robot. Autom. Lett.*, vol. 6, no. 4, pp. 6529–6536, Oct. 2021. [Online]. Available: http://www.ipb.uni-bonn.de/pdfs/chen2021ral-iros.pdf

[22] B. Mersch, X. Chen, I. Vizzo, L. Nunes, J. Behley, and C. Stachniss, "Receding moving object segmentation in 3D LiDAR data using sparse 4D convolutions," *IEEE Robot. Autom. Lett.*, vol. 7, no. 3, pp. 7503–7510, Jul. 2022. [Online]. Available: https://www.ipb.uni-bonn.de/wp-content/papercite-data/pdf/mersch2022ral.pdf

[23] X. Chen et al., "Automatic labeling to generate training data for online LiDAR-based moving object segmentation," 2022, *arXiv:2201.04501*.

[24] J. Sun et al., "Efficient spatial–temporal information fusion for LiDAR-based 3D moving object segmentation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2022, pp. 11456–11463.

[25] G. He, Q. Zhang, and Y. Zhuang, "Online semantic-assisted topological map building with LiDAR in large-scale outdoor environments: Toward robust place recognition," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–12, 2022.

[26] C. Qian, Z. Xiang, Z. Wu, and H. Sun, "RF-LIO: Removal-first tightly-coupled LiDAR inertial odometry in high dynamic environments," 2022, *arXiv:2206.09463*.

[27] L. He, W. Li, Y. Guan, and H. Zhang, "IGICP: Intensity and geometry enhanced LiDAR odometry," *IEEE Trans. Intell. Vehicles*, vol. 9, no. 1, pp. 541–554, Jan. 2024.

[28] K. Chen, B. T. Lopez, A.-A. Agha-Mohammadi, and A. Mehta, "Direct LiDAR odometry: Fast localization with dense point clouds," *IEEE Robot. Autom. Lett.*, vol. 7, no. 2, pp. 2000–2007, Apr. 2022.

[29] C. Wu et al., "Voxelmap++: Mergeable voxel mapping method for online LiDAR (-inertial) odometry," *IEEE Robot. Autom. Lett.*, vol. 9, no. 1, pp. 427–434, Nov. 2023.

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

[31] K. Koide, M. Yokozuka, S. Oishi, and A. Banno, "Voxelized GICP for fast and accurate 3D point cloud registration," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2021, pp. 11054–11059.

[32] K. Koide, J. Miura, and E. Menegatti, "A portable three-dimensional LiDAR-based system for long-term and wide-area people behavior measurement," *Int. J. Adv. Robotic Syst.*, vol. 16, no. 2, pp. 1285–1292, Mar. 2019.

[33] L. Ma, C. Kerl, J. Stückler, and D. Cremers, "CPA-SLAM: Consistent plane-model alignment for direct RGB-D SLAM," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2016, pp. 1285–1291.

[34] L.-T. Hsu et al., "UrbanNav: An open-sourced multisensory dataset for benchmarking positioning algorithms designed for urban areas," in *Proc. ION GNSS+, Int. Tech. Meeting Satell. Division Inst. Navigat.*, Oct. 2021, pp. 226–256.

[35] W. Wen et al., "UrbanLoco: A full sensor suite dataset for mapping and localization in urban scenes," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2020, pp. 2310–2316.

[36] D. He, W. Xu, N. Chen, F. Kong, C. Yuan, and F. Zhang, "Point-LIO: Robust high-bandwidth light detection and ranging inertial odometry," *Adv. Intell. Syst.*, vol. 5, no. 7, Jul. 2023, Art. no. 2200459.

[37] B. Jiang and S. Shen, "A LiDAR-inertial odometry with principled uncertainty modeling," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2022, pp. 13292–13299.

[38] M. Grupp. (2017). *Evo: Python Package for the Evaluation of Odometry and SLAM*. [Online]. Available: https://github.com/MichaelGrupp/evo

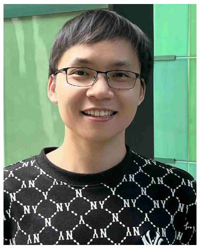
Qi Wu received the bachelor's degree in electrical engineering and automation from Chongqing University of Posts and Telecommunications, Chongqing, China, in 2016, and the master's degree from Beijing University of Posts and Telecommunications, Beijing, China, in 2019. He is currently pursuing the Ph.D. degree with Shanghai Jiao Tong University, Shanghai, China.

Xieyuanli Chen (Member, IEEE) received the bachelor's degree in electrical engineering and automation from Hunan University, Changsha, China, in 2015, the master's degree in robotics from the National University of Defense Technology (NUDT), Changsha, in 2017, and the Ph.D. degree in robotics from Photogrammetry and Robotics Laboratory, University of Bonn, Bonn, Germany, in 2022.

He joined the College of Intelligence Science and Technology, NUDT, in 2022, where he is now an Associate Professor.

Xiangyu Xu received the bachelor's and master's degrees from Wuhan University, Wuhan, China, in 2019.

He is a Senior Algorithm Engineer with ByteDance Inc., Beijing, China. His main research interests include visual-inertial odometry, robotics, and reconstruction algorithms.

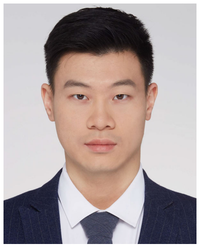


Xinliang Zhong received the bachelor's and master's degrees from Beijing Institute of Technology, Beijing, China, in 2019.

He is a Senior Algorithm Engineer with Banma Network, Alibaba Group, Hangzhou, China. His main research interests include intelligent driving systems and hardware in the loop.

Xingwei Qu (Student Member, IEEE) received the B.S. degree in automation science from Beihang University, Wuhan, China, in 2015. He is currently pursuing the Ph.D. degree with The University of Manchester, Manchester, U.K.

His current research interests include machine learning, multisensor fusion, large language models, and music generation.



Songpengcheng Xia (Member, IEEE) received the B.S. degree in navigation engineering from Wuhan University, Wuhan, China, in 2019. He is currently pursuing the Ph.D. degree with Shanghai Jiao Tong University, Shanghai, China.

His current research interests include machine learning, multisensor fusion, sensor-based human activity recognition, and human pose estimation with wearables.

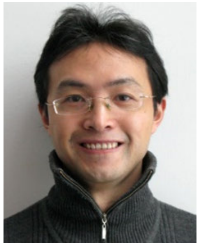
Guoqing Liu received the bachelor's degree from the School of Automation, Harbin University of Science and Technology, Harbin, China, in 2018, and the master's degree from the Faculty of Robot Science and Engineering, Northeastern University, Shenyang, China, in 2021.

Liu Liu received the B.S. degree from Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2015, and the Ph.D. degree from the University of Science and Technology of China, Hefei, China, in 2020.

He was a Post-Doctoral Researcher at Shanghai Jiao Tong University, Shanghai, China, from 2020 to 2022. He is currently an Associate Professor with Hefei University of Technology, Hefei, China. His research interests include computer vision, deep learning, and embodied AI.

Wenxian Yu (Senior Member, IEEE) received the B.S., M.S., and Ph.D. degrees from the National University of Defense Technology, Changsha, China, in 1985, 1988, and 1993, respectively.

From 1996 to 2008, he was a Professor at the College of Electronic Science and Engineering, National University of Defense Technology, where he was also the Deputy Head and an Assistant Director of the National Key Laboratory of Automatic Target Recognition. From 2009 to 2011, he was the Executive Dean of the School of Electronic, Information, and Electrical Engineering, Shanghai Jiao Tong University, Shanghai, China, where he is currently a Yangtze River Scholar Distinguished Professor and the Head. His research interests include remote sensing information processing, automatic target recognition, and multisensor data fusion.



Ling Pei (Senior Member, IEEE) received the Ph.D. degree from Southeast University, Nanjing, China, in 2007.

From 2007 to 2013, he was a Specialist Research Scientist at the Finnish Geospatial Research Institute, Espoo, Finland. He is currently a Professor with the School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai, China. He has authored or co-authored over 100 scientific papers. He is also an inventor of 25 patents and pending patents. His main research interests include indoor/outdoor seamless positioning, ubiquitous computing, wireless positioning, bio-inspired navigation, context-aware applications, location-based services, and navigation of unmanned systems.

Dr. Pei was a recipient of Shanghai Pujiang Talent in 2014.