SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

September 2026, Vol. 69, Iss. 9, 192205:1–192205:16
https://doi.org/10.1007/s11432-025-4959-7

RV-LIO: reinforced-vector-based LiDAR-inertial odometry for weak-feature environments

Chunyan WANG[1,2], Weiyi KONG[2], Yitao LU[2], Wei DONG[1,3,4,5*],
Zhongkui LI[6] & Fang DENG[1,3,4,5]

[1]*Beijing Institute of Technology, Zhuhai, Zhuhai 519088, China*
[2]*School of Aerospace Engineering, Beijing Institute of Technology, Beijing 100081, China*
[3]*National Key Laboratory of Autonomous Intelligent Unmanned Systems, Beijing Institute of Technology, Beijing 100081, China*
[4]*Beijing Key Laboratory of Lightweight Intelligent Systems, Beijing Institute of Technology, Beijing 100081, China*
[5]*School of AI, Beijing Institute of Technology, Beijing 100081, China*
[6]*School of Advanced Manufacturing and Robotics, Peking University, Beijing 100871, China*

Received 11 April 2025/Revised 3 September 2025/Accepted 24 October 2025/Published online 13 August 2026

Abstract This paper studies the light detection and ranging (LiDAR)-inertial odometry (LIO) invalidation problem for unmanned aerial vehicles (UAVs) in weak-feature environments. A reinforced-vector-based LIO (RV-LIO) system is developed to tackle this challenge. First, valid localization is proposed as a criterion based on constraint vector distribution in weak-feature environments. Two boundary conditions are derived from the relation between vector distribution and state observability. Second, a reinforced vector module is designed to extract and reinforce special constraint vectors through iterative singular value decomposition and noise coefficient adjustment, respectively. Third, a repaired vector module is built to capture and correct certain constraints by incorporating prior knowledge of tunnel surroundings. Both modules serve as subsystems of the filter process in the LIO system. Finally, Gazebo simulation and real experiment results are provided to verify the effectiveness of the proposed system.

Keywords LiDAR-inertial odometry, weak-feature environments, feature extraction, reinforced vector, unmanned aerial vehicles

Citation Wang C Y, Kong W Y, Lu Y T, et al. RV-LIO: reinforced-vector-based LiDAR-inertial odometry for weak-feature environments. Sci China Inf Sci, 2026, 69(9): 192205, https://doi.org/10.1007/s11432-025-4959-7

1 Introduction

In recent years, unmanned aerial vehicles (UAVs) have been increasingly used in many complex applications [1–3]. Among these, mapping and localization are key research topics that aim to support UAV navigation, obstacle avoidance, and target tracking [4].

The proposition of the simultaneous localization and mapping (SLAM) issue dates back to Smith and Cheeseman's seminal work in 1986 [5]. At that time, an early solution was obtained using an extended Kalman filter by casting the problem as an optimal filtering one. Dissanayake et al. [6] further developed an SLAM solution for robotics applications. Then, Durrant-Whyte and Bailey [7] proposed a probabilistic framework to solve the problem and introduced various algorithmic improvements, which have since become classic frameworks to address this issue. Thereafter, reliable SLAM in two-dimensional (2D) environments using wheeled robots equipped with lasers and encoders became feasible. However, despite the above marvelous progress throughout the past 20 years, current SLAM algorithms may still fail to cope with possible featureless environments, especially for indoor UAV applications [8, 9]. Because of the lack of global information (e.g., provided by global navigation satellite systems) or locally accurate measurements (e.g., wheel encoders), degenerated environments may confuse pose calculations and cause irreversible errors [10, 11].

Existing studies in three-dimensional (3D) light detection and ranging (LiDAR) SLAM are typically based on the structure of LiDAR odometry and mapping (LOAM) proposed by Zhang and Singh [12]. The three main modules in these algorithms are feature extraction, odometry calculation, and mapping. Subsequent studies, such as LeGO-LOAM [13], mainly kept the original framework but attempted to decrease the computation load in feature extraction. In addition, a good initial pose provided by the inertial measurement unit (IMU) can decrease

* Corresponding author (email: dong@bit.edu.cn)

the number of iterative closest point [14] in odometry calculation, as demonstrated in LION [15] and LIOM [16]. Furthermore, numerous optimization methods have been developed to accelerate mapping and matching processes, such as LINS [17], LILI-LOM [18], and LIO-SAM [19]. Some key improvements in feature storage structure and odometry calculation have been made in FAST-LIO2 [20], and researchers have robustly achieved more accurate and faster performance. In all these studies, the scan's pose is optimized based on the fact that each scan's pose is unique in the local map. More specifically, only the correct posture and position can bring the current scan's points into the known point cloud map perfectly.

Compared with vision-based methods [21–23], 3D LiDAR provides a direct, dense, and accurate environmental perception [24–26]. Nonetheless, in some specific environments, such as coal mines, tunnels, and chimneys [27, 28], partial unobservability issues occur due to the descent of localization solvability. Many existing studies on SLAM have considered and analyzed such partially observable and completely unobservable situations [29, 30], yet only a few can solve this challenge effectively. One possible solution is the use of external infrastructure to obtain a global position in this situation. In GPS-denied environments, external measurements from ultra-wideband (UWB) systems are introduced to develop a fusion framework to obtain global position estimates and mitigate error accumulation [31, 32]. In addition, camera information and the change in tunnel direction are utilized to limit the effect of similar textures in tunnel environments [33]. However, the LiDAR-inertial odometry (LIO) problem in featureless and structured environments without direction change or any external infrastructure remains open.

In recent years, distributed optimization and graph-based SLAM have become core directions in multirobot perception for advancing robustness, real-time performance, and efficiency. Specifically, Kimera-Multi [34] is recognized as the first fully distributed dense metric-semantic SLAM using graduated nonconvexity-based distributed pose graph optimization (PGO) and peer-to-peer communication to achieve accuracy comparable to centralized solutions. Meanwhile, Zhong et al. [35] developed a distributed collaborative LiDAR SLAM framework for a robotic swarm (DCL-SLAM) that leveraged LiDAR Iris descriptors and distributed PGO to improve localization in complex environments. DOOR-SLAM [36] enhanced outlier robustness via distributed pairwise consistent measurement maximization to reject spurious loops without raw sensor data. For 3D LiDAR real-time collaboration, RDC-SLAM [37] optimized data transmission and map registration for sparse networks. In graph modeling, DDF-SAM [38] adopted constrained factor graphs and Bayesian approximation to reduce computational complexity. The four-dimensional radar-based pose graph SLAM [39] integrated ego-velocity preintegration and Gaussian processes to improve dynamic-environment robustness. Karam et al. [40] used cubic splines and plane classification to improve indoor mapping robustness. Additionally, Doherty et al. [41] sparsified pose graphs via Laplacian spectral preservation to accelerate large-scale PGO while ensuring accuracy. However, the above results paid little attention to weak-feature environments, which inevitably affect their feature extraction and degrade localization performance.

In this paper, a reinforced-vector-based LIO (RV-LIO) system is proposed to address the failure of conventional LIO in weak-feature environments. Both Gazebo simulation and a real UAV experiment are performed to sufficiently verify the effectiveness of the proposed approach. The main contributions of this work are threefold.

(1) The proposed RV-LIO system includes an activating criterion, a reinforced vector module, and a repaired vector module. To the best of our knowledge, this work is the first to address the odometry failure challenge in straight weak-feature environments using only LiDAR-inertial sensors in aerial robotics.

(2) Based on the formulated constraint vector distribution model, a distribution criterion for system activation is given to differentiate between normal and abnormal situations. Unlike previous observability analyses [29, 30], this work explicitly reveals the relation between the observability and the vector distribution model, with two boundary conditions for local observability in the studied environments.

(3) Iterative singular value decomposition (SVD) and noise coefficient adjustment methods are developed in the design of the reinforced vector module to capture and reinforce special constraint vectors (SCVs), respectively. Meanwhile, the repaired vector module pulls certain constraint vectors into the desired correct directions. Together, these two modules enable robust and effective odometry performance in challenging weak-feature environments.

2 Problem statement and preliminaries

Considering that FAST-LIO2 is considered a mainstream framework for its fast and robust performance, this work basically inherits its structure. A group of accumulated points sampled from 3D LiDAR over a time period is described as a scan. Once a scan's odometry is known, its points can be registered into a local map. When a subsequent scan is obtained, the IMU can provide an initial pose of the current LiDAR through integration. Then, the pose is optimized by matching the scan's points to the local map and updating the system state using an iterated error state Kalman filter (IESKF).

The state $\mathbf{x}$ used in the algorithm contains the IMU's posture ${}^G\mathbf{R}_I$, position ${}^G\mathbf{p}_I$, and velocity ${}^G\mathbf{v}_I$ with respect to the fixed (global) coordinate, the angular velocity bias $\mathbf{b}_{\boldsymbol{\omega}}$ and the linear acceleration bias $\mathbf{b}_{\mathbf{a}}$ in IMU measurement, LiDAR's extrinsic attitude ${}^I\mathbf{R}_L$ and position ${}^I\mathbf{p}_L$ with respect to the IMU coordinate, and the gravity acceleration ${}^G\mathbf{g}$. The state set and its manifold $\mathcal{M}$ are defined as

$$\mathcal{M} \triangleq SO(3) \times \mathbb{R}^{12} \times SO(3) \times \mathbb{R}^{6},$$
$$\mathbf{x} \triangleq \left[{}^G\mathbf{R}_I^{\mathrm{T}} \ \ {}^G\mathbf{p}_I^{\mathrm{T}} \ \ {}^G\mathbf{v}_I^{\mathrm{T}} \ \ \mathbf{b}_{\boldsymbol{\omega}}^{\mathrm{T}} \ \ \mathbf{b}_{\mathbf{a}}^{\mathrm{T}} \ \ {}^I\mathbf{R}_L^{\mathrm{T}} \ \ {}^I\mathbf{p}_L^{\mathrm{T}} \ \ {}^G\mathbf{g}^{\mathrm{T}} \right]^{\mathrm{T}} \in \mathcal{M}.$$

Hereafter, the notations k, υ, and j represent the scan, iteration, and point indexes, respectively. Specifically, the subscript k denotes the kth frame of the LiDAR pose; thus, $k+1$ refers to the next frame. The superscript υ represents the iteration number; thus, $\upsilon+1$ indicates the next iteration. The symbols $\widehat{*}$ and $\widetilde{*}$ denote the posterior result and the bias between the current result and the ground truth, respectively. The state $\widehat{\mathbf{x}}_k^{\upsilon}$ updated in IESKF is

$$\widehat{\mathbf{x}}_k^{\upsilon+1} = \widehat{\mathbf{x}}_k^{\upsilon} \boxplus \left(-\mathbf{K}\mathbf{z}_k^{\upsilon} - (\mathbf{I} - \mathbf{K}\mathbf{H}) (\mathbf{J}^{\upsilon})^{-1} (\widehat{\mathbf{x}}_k^{\upsilon} \boxminus \widehat{\mathbf{x}}_k) \right), \tag{1}$$

where $\mathbf{K} = \left(\mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{H} + \widehat{\mathbf{P}}^{-1} \right)^{-1} \mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}$, $\mathbf{z}$ denotes the estimators' results, the encapsulation operators $\boxplus$ and $\boxminus$ indicate addition and subtraction on the manifold $\mathcal{M}$, $\mathbf{P}$ represents the covariance, $\mathbf{H}$ represents the estimator Jacobian matrix, $\mathbf{R}$ represents the noise matrix in the estimators, and $\mathbf{J}$ represents the Jacobian matrix of the error linearization model given by

$$\begin{aligned} \mathbf{x}_k \boxminus \widehat{\mathbf{x}}_k^0 &= (\widehat{\mathbf{x}}_k^{\upsilon} \boxplus \widetilde{\mathbf{x}}_k^{\upsilon}) \boxminus \widehat{\mathbf{x}}_k = \widehat{\mathbf{x}}_k^{\upsilon} \boxminus \widehat{\mathbf{x}}_k^0 + \mathbf{J}^{\upsilon}\widetilde{\mathbf{x}}_k^{\upsilon} \\ &\sim \mathcal{N}\left(\mathbf{0}, \widehat{\mathbf{P}}_k\right), \end{aligned} \tag{2}$$

with $\mathbf{x}_k = (\widehat{\mathbf{x}}_k^{\upsilon} \boxplus \widetilde{\mathbf{x}}_k^{\upsilon})$ denoting the true value. Moreover, the estimator design has the following structure:

$$\begin{aligned} \mathbf{z}_j^{\upsilon} = \mathbf{h}_j(\widehat{\mathbf{x}}_k^{\upsilon}, \mathbf{0}) &= \mathbf{u}_j^{\mathrm{T}} \left({}^G\widehat{\mathbf{T}}_{I_k}^{\upsilon} \widehat{\mathbf{T}}_{L_k}^{\upsilon} \mathbf{p}_j - {}^G\mathbf{q}_j \right) \\ &= \mathbf{u}_j^{\mathrm{T}} \left[{}^G\widehat{\mathbf{R}}_{I_k}^{\upsilon} \left(\widehat{\mathbf{R}}_{L_k}^{\upsilon} \mathbf{p}_j + \widehat{\mathbf{p}}_{L_k}^{\upsilon} \right) + {}^G\widehat{\mathbf{p}}_{I_k}^{\upsilon} - {}^G\mathbf{q}_j \right]. \end{aligned} \tag{3}$$

Here, the IMU's pose ${}^G\mathbf{T}_{I_k} = \left({}^I\mathbf{R}_{I_k}, {}^I\mathbf{p}_{I_k}\right)$ and extrinsic parameters $\widehat{\mathbf{T}}_{L_k}^{\upsilon} = \left({}^G\mathbf{R}_{L_k}, {}^G\mathbf{p}_{L_k}\right)$ represent LiDAR's pose with respect to IMU. Moreover, $\mathbf{u}_j$ denotes the normal vector of the corresponding plane, and ${}^G\mathbf{q}_j$ represents the point lying on this plane [20]. Then, one can obtain $\mathbf{H}$ by composing the Jacobian derivatives of all $\mathbf{h}_j\left(\widehat{\mathbf{x}}_k^{\upsilon} \boxplus \widetilde{\mathbf{x}}_k^{\upsilon}, {}^L\mathbf{n}_j\right)$ with respect to $\widetilde{\mathbf{x}}_k^{\upsilon}$, where ${}^L\mathbf{n}_j$ denotes the LiDAR measurement noise.

The iteration scheme of IESKF [20] is as follows.

(1) Initialize $\widehat{\mathbf{x}}_k^0$ and $\widehat{\mathbf{P}}_k$ through IMU forward propagation.

(2) For any integer $\upsilon \geqslant 0$, compute $\mathbf{J}^{\upsilon}$ from (2).

(3) Match the scan using the current pose $\widehat{\mathbf{x}}^{\upsilon}$ with the map.

(4) Calculate the estimator residual $\mathbf{z}_k^{\upsilon}$ and Jacobian $\mathbf{H}_k^{\upsilon}$ using (3).

(5) Update the state $\widehat{\mathbf{x}}_k^{\upsilon}$ using (1) and go to step (2).

Notably, the update expression (1) implies that the estimator noise $\mathbf{R}$ affects the coefficient in the current gain $\mathbf{K}$.

Remark 1. Each element of $\mathbf{R}$ denotes the estimator noise of each point. A larger estimator noise implies less feasibility of the specific estimator. In particular, estimators with a lower block in $\mathbf{R}$ lead to higher $\mathbf{K}$ and larger weights for the estimator's result compared with the current prediction in the next update.

Consider a weak-feature environment where error drift can happen. Figure 1(a) illustrates an extreme case. The environment is completely smooth and structured, where each horizontal section along the axis of a tunnel is completely the same. Consequently, the position solution of a UAV in this specific direction is not unique. The correct position 1 can be mistaken for position 2, provided that their estimators' results are completely the same.

Meanwhile, in the real case shown in Figure 1(b), because the wall and floor are hardly smooth, the points sampled from them include noise. In addition, a few features (e.g., beams, pillars, and lights) exist along the vertical axis. However, due to the drift in IMU integration or misleading convergence trend caused by unsmooth points, the localization of position 1 can drift to the incorrect position 2 after iteration. Then, the few remaining constraints from the vertical features are unavailable as the wrong position misleads the matching principle in the local map. For the following optimization, the situation worsens to a case similar to the extreme one.

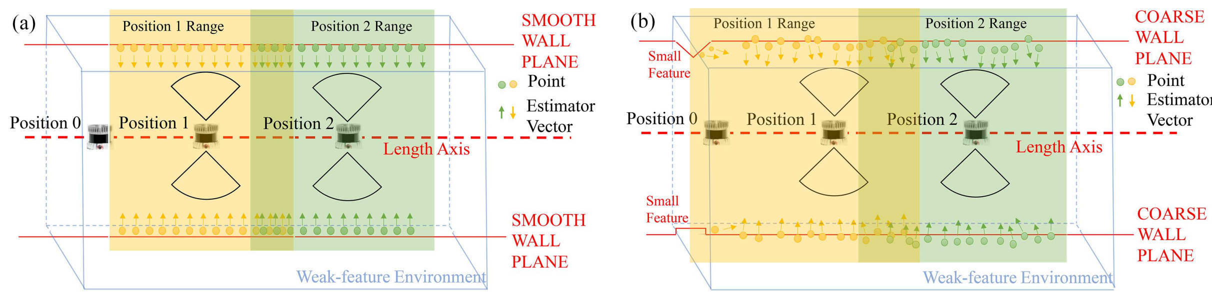

Figure 1 (Color online) Illustration of extreme and real situations in weak-feature environments. A 2D cross-section is sampled along the length axis of the tunnel to simplify the odometry process in (a) the extreme case with completely smooth walls and (b) the real case with small features and coarse walls.

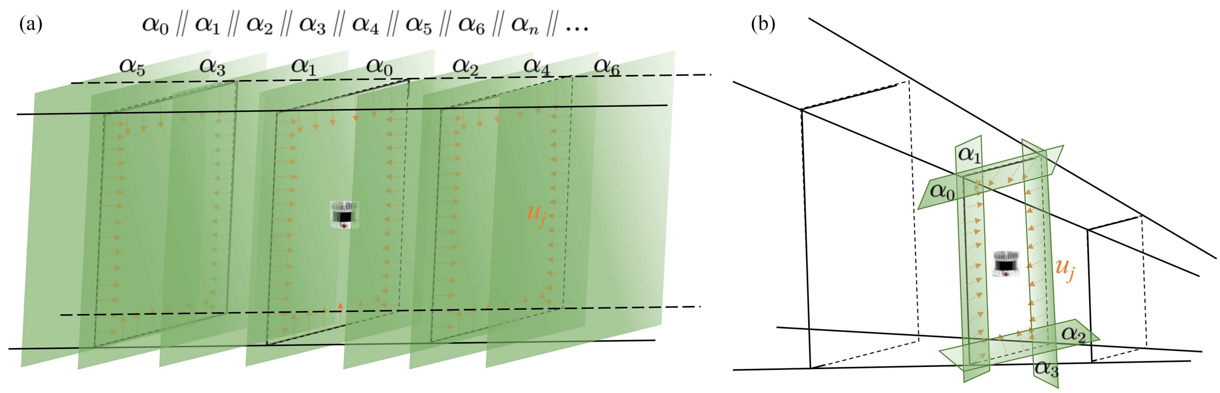

Figure 2 (Color online) Constraint vector illustration in abnormal and normal situations. The symbol // represents a parallel relationship. Each red arrow represents the direction of one constraint vector $\mathbf{u}_j$, which has been defined previously as the normal vector of the corresponding plane. In the abnormal situation (a), all vectors lie on the green parallel planes. In the normal situation (b), the vectors lie on different planes (e.g., the green non-parallel planes α_0, α_1, α_2, and α_3).

From the above analysis, the conventional LIO is easily invalidated in weak-feature environments. Thus, we aim to develop an improved LIO system to address this challenge.

3 Main results

3.1 Vector distribution criterion

From (3), the residual obtained from the estimator is actually the distance between the new scan's point and the corresponding known plane it could belong to in the local map. The constraint direction of the estimator is the normal vector of this plane (i.e., the constraint vector). As mentioned previously, state optimization aims to put the scan in a correct position and posture. Supposing that the current posture is accurate, the remaining task is to translate points to correct positions. The performance of this translation is evaluated using the estimator residual, with smaller residuals indicating a better fit between the new scan points and the corresponding known planes. In the IESKF, the scan points are iteratively translated along the directions of these planes' normal vectors. Larger residuals correspond to longer necessary movement distances. Eventually, the process stops until all the scan's points find their appropriate plane.

The constraint vector distribution for the extreme case in Figure 1(a) is shown in Figure 2(a), where all vectors are coplanar. In this case, there exists α such that $\forall \mathbf{u}_j \subseteq \alpha$. Note that the involved vectors are only direction-specific. Apparently, the UAV's poses within α_0 can be solved correctly using constraint vectors. However, the plane's location along the tunnel axis cannot be determined. The detailed analysis is given in Subsection 3.2. At this point, the following criterion can be introduced. Note that the criterion below is necessary but not sufficient because it considers most vectors instead of all vectors, due to the reason revealed in Figure 1(b).

Proposition 1. If there exists a plane α such that most constraint vectors $\mathbf{u}_i$ belong to it, the error drift could happen with high probability at this point.

The error drift is essentially a result of unobservability. In [42], the SLAM system was considered observable if there were no two points $\mathbf{x}_0$ and $\mathbf{x}_1 \in \mathcal{M}$ that were indistinguishable in the estimators. However, distinguishing two points with long distances was not necessary for an SLAM system, as estimator corrections are often performed around the initial pose. Thus, the concept of locally weak observability was proposed in [43] to describe this particular observability.

Definition 1. For one point $\mathbf{x}_0$ in $\mathcal{M}$, if any other state $\mathbf{x}_1 \in \mathcal{V}$ can be distinguished in the estimators, the system is locally weakly observable at point $\mathbf{x}_0$, where $\mathcal{V}$ denotes an open neighborhood of $\mathbf{x}_0$ in domain $\mathcal{M}$.

Define $\mathcal{O}$ as the observability matrix whose rows are constructed from the gradients of the Lie derivatives of each residual calculation h_i [43]. Then, $\mathcal{O}$ is indeed the estimator matrix $\mathbf{H}$. According to (3), matrix $\mathbf{H}$ is given by

$$\mathbf{H}=\begin{bmatrix} \frac{\partial h_1}{\partial\left({}^G\tilde{\mathbf{R}}^v_{I_k}\right)} & \frac{\partial h_1}{\partial\left({}^G\tilde{\mathbf{p}}^v_{I_k}\right)} & \mathbf{0}_{3\times12} & \frac{\partial h_1}{\partial\left({}^G\tilde{\mathbf{R}}^v_{L_k}\right)} & \frac{\partial h_1}{\partial\left({}^G\tilde{\mathbf{p}}^v_{L_k}\right)} \\ \frac{\partial h_2}{\partial\left({}^G\tilde{\mathbf{R}}^v_{I_k}\right)} & \frac{\partial h_2}{\partial\left({}^G\tilde{\mathbf{p}}^v_{I_k}\right)} & \mathbf{0}_{3\times12} & \frac{\partial h_2}{\partial\left({}^G\tilde{\mathbf{R}}^v_{L_k}\right)} & \frac{\partial h_2}{\partial\left({}^G\tilde{\mathbf{p}}^v_{L_k}\right)} \\ & & \vdots & & \\ \frac{\partial h_n}{\partial\left({}^G\tilde{\mathbf{R}}^v_{I_k}\right)} & \frac{\partial h_n}{\partial\left({}^G\tilde{\mathbf{p}}^v_{I_k}\right)} & \mathbf{0}_{3\times12} & \frac{\partial h_n}{\partial\left({}^G\tilde{\mathbf{R}}^v_{L_k}\right)} & \frac{\partial h_n}{\partial\left({}^G\tilde{\mathbf{p}}^v_{L_k}\right)} \end{bmatrix}. \tag{4}$$

Lemma 1 is provided for observability analysis.

Lemma 1. The system is locally weakly observable with respect to specific error states if the corresponding block in $\mathcal{O}$ has full column rank at $\mathbf{x}_0$.

Note that the estimator in IESKF only focuses on the error states (i.e., the bias) instead of the full system states.

3.2 Observability analysis and validation

Comparing the two cases shown in Figure 2, the supposed plane α exists in the first case. All normal vectors used in the estimators belong to α. Then, two linearly independent vectors $\mathbf{u_a}$ and $\mathbf{u_b}$ are selected as bases. Using the Gram-Schmidt method, orthogonal bases $\eta_\mathbf{a}$ and $\eta_\mathbf{b}$ can further be obtained. Therefore, all vectors in α can be described by

$$\mathbf{u}_j \in \mathrm{span}\{\eta_\mathbf{a}, \eta_\mathbf{b}\} = \mathrm{span}\{\mathbf{u_a}, \mathbf{u_b}\}. \tag{5}$$

For the SLAM system, extrinsic parameters are often considered as constants. $\tilde{\mathbf{p}}^v_{L_k}$ and $\tilde{\mathbf{R}}^v_{L_k}$ are not optimized except in initial calibration. This paper focuses on the locally weak observability of the error states ${}^G\tilde{\mathbf{R}}^v_{I_k}$ and ${}^G\tilde{\mathbf{p}}^v_{I_k}$.

3.2.1 *Observability of posture*

The block related to ${}^G\tilde{\mathbf{R}}^v_{I_k}$ is extracted from $\mathcal{O}$ (i.e., $\mathbf{H}$ in (4)):

$$\mathcal{O}_{{}^G\tilde{\mathbf{R}}^v_{I_k}} = \begin{bmatrix} \frac{\partial h_1}{\partial\left({}^G\tilde{\mathbf{R}}^v_{I_k}\right)} \\ \frac{\partial h_2}{\partial\left({}^G\tilde{\mathbf{R}}^v_{I_k}\right)} \\ \vdots \\ \frac{\partial h_n}{\partial\left({}^G\tilde{\mathbf{R}}^v_{I_k}\right)} \end{bmatrix} = \begin{bmatrix} \mathbf{u}_1^{\mathrm{T}}\left(-{}^G\hat{\mathbf{R}}^v_{I_k}\,{}^G\tilde{\mathbf{R}}^v_{I_k}\,{}^I\mathbf{p}_1\right)^\wedge \\ \mathbf{u}_2^{\mathrm{T}}\left(-{}^G\hat{\mathbf{R}}^v_{I_k}\,{}^G\tilde{\mathbf{R}}^v_{I_k}\,{}^I\mathbf{p}_2\right)^\wedge \\ \vdots \\ \mathbf{u}_n^{\mathrm{T}}\left(-{}^G\hat{\mathbf{R}}^v_{I_k}\,{}^G\tilde{\mathbf{R}}^v_{I_k}\,{}^I\mathbf{p}_n\right)^\wedge \end{bmatrix}, \tag{6}$$

where ${}^I\mathbf{p}_i = \widehat{\mathbf{R}}^v_{L_k}\mathbf{p}_i + \widehat{\mathbf{p}}^v_{L_k}$ denotes this point's position in the IMU frame and the superscript $^\wedge$ denotes the antisymmetric operation for a vector $\mathbf{a} = [a_1, a_2, a_3]^{\mathrm{T}}$ as

$$\mathbf{a}^\wedge = [[0, a_3, -a_2]^{\mathrm{T}}, [a_3, 0, -a_1]^{\mathrm{T}}, [-a_2, a_1, 0]^{\mathrm{T}}]. \tag{7}$$

Then, a sufficient and unnecessary condition is provided to determine whether ${}^G\tilde{\mathbf{R}}^v_{I_k}$ is locally weakly observable.

Condition 1. There exist no fewer than two non-parallel corresponding planes such that each can contain no fewer than two points in the estimators.

Consider a corresponding plane α_0 providing no fewer than two points for the estimators. The block of α_0 in $\mathcal{O}_{{}^G\tilde{\mathbf{R}}^{\upsilon}_{I_k}}$ is

$$\mathcal{O}_{\alpha_0}=\begin{bmatrix}\mathbf{u}_{\alpha_0}^{\mathrm{T}}\left(-{}^G\hat{\mathbf{R}}^{\upsilon}_{I_k}{}^G\tilde{\mathbf{R}}^{\upsilon}_{I_k}{}^I\mathbf{p}_1\right)^{\wedge}\\ \mathbf{u}_{\alpha_0}^{\mathrm{T}}\left(-{}^G\hat{\mathbf{R}}^{\upsilon}_{I_k}{}^G\tilde{\mathbf{R}}^{\upsilon}_{I_k}{}^I\mathbf{p}_2\right)^{\wedge}\\ \vdots\\ \mathbf{u}_{\alpha_0}^{\mathrm{T}}\left(-{}^G\hat{\mathbf{R}}^{\upsilon}_{I_k}{}^G\tilde{\mathbf{R}}^{\upsilon}_{I_k}{}^I\mathbf{p}_n\right)^{\wedge}\end{bmatrix}. \tag{8}$$

Elementary row operations are performed to obtain $\mathcal{O}_{s\alpha_0}$ as

$$\mathcal{O}_{s\alpha_0}=\begin{bmatrix}\mathbf{u}_{\alpha_0}^{\mathrm{T}}\left(-{}^G\hat{\mathbf{R}}^{\upsilon}_{I_k}{}^G\tilde{\mathbf{R}}^{\upsilon}_{I_k}{}^I\mathbf{p}_1\right)^{\wedge}\\ \mathbf{u}_{\alpha_0}^{\mathrm{T}}\left(-{}^G\hat{\mathbf{R}}^{\upsilon}_{I_k}{}^G\tilde{\mathbf{R}}^{\upsilon}_{I_k}\left({}^I\mathbf{p}_2-{}^I\mathbf{p}_1\right)\right)^{\wedge}\\ \vdots\\ \mathbf{u}_{\alpha_0}^{\mathrm{T}}\left(-{}^G\hat{\mathbf{R}}^{\upsilon}_{I_k}{}^G\tilde{\mathbf{R}}^{\upsilon}_{I_k}\left({}^I\mathbf{p}_n-{}^I\mathbf{p}_1\right)\right)^{\wedge}\end{bmatrix}. \tag{9}$$

Because the points $\mathbf{p}_1,\mathbf{p}_2,\ldots,\mathbf{p}_n$ are coplanar, the vectors $\mathbf{p}_1$, $\mathbf{p}_2-\mathbf{p}_1,\ldots,\mathbf{p}_n-\mathbf{p}_1$ are also coplanar. In addition, because any two points $\mathbf{p}_s$ and $\mathbf{p}_m$ accumulated from LiDAR and the origin cannot be located on one line, there are at least two vectors that are not collinear. Similar to (5), let the vectors $\mathbf{p}_s$ and $\mathbf{p}_m-\mathbf{p}_s$ be selected as bases. Then, all vectors on this plane can be described by these two bases. Consequently, the rank of $\mathcal{O}_{\alpha 0}$ is

$$\operatorname{rank}\left(\mathcal{O}_{\alpha_0}\right)=\operatorname{rank}\left(\mathcal{O}_{s\alpha_0}\right)=2. \tag{10}$$

Remark 2. Actually, the sampled points $\mathbf{p}_1,\mathbf{p}_2,\ldots,\mathbf{p}_n$ may not be located on their corresponding plane and thus may not be coplanar. One reason is that the matching process may make mistakes. Another reason is that the point's local position contains noise and bias. These reasons can artificially increase the rank of the observability matrix, leading to misleading observability.

Furthermore, the dimension of the null space of $\mathcal{O}_{\alpha_0}$ can be calculated as

$$\dim(N(\mathcal{O}_{\alpha_0}))=n-\operatorname{rank}(\mathcal{O}_{\alpha_0})=1. \tag{11}$$

The unique basis in $N(\mathcal{O}_{\alpha_0})$ is $\mathbf{u}_{\boldsymbol{\alpha_0}}$ because

$$\mathbf{u}_{\alpha_0}^{\mathrm{T}}\left(-{}^G\hat{\mathbf{R}}^{\upsilon}_{I_k}{}^G\tilde{\mathbf{R}}^{\upsilon}_{I_k}{}^I\mathbf{p}_1\right)^{\wedge}\mathbf{u}_{\alpha_0}=0, \tag{12}$$

considering that $\left(-{}^G\hat{\mathbf{R}}^{\upsilon}_{I_k}{}^G\tilde{\mathbf{R}}^{\upsilon}_{I_k}{}^I\mathbf{p}_1\right)^{\wedge}$ is antisymmetric. Following procedures similar to those in (6)–(12), for another plane α_1 non-parallel to α_0, the unique basis in the null space of another block $\mathcal{O}_{\alpha_1}$ is $\mathbf{u}_{\alpha_1}$, where $\mathcal{O}_{\alpha_1}$ is the corresponding block in $\mathcal{O}_{{}^G\tilde{\mathbf{R}}^{\upsilon}_{I_k}}$. Because of the non-parallel relation, the normal vectors $\mathbf{u}_{\alpha_0}$ and $\mathbf{u}_{\alpha_1}$ are linearly independent. Then, the dimensions of $N\left(\mathcal{O}_{\alpha_0}\right)\cap N\left(\mathcal{O}_{\alpha_1}\right)$ and $N\left(\mathcal{O}_{{}^G\tilde{\mathbf{R}}^{\upsilon}_{I_k}}\right)$ are both 0. This suggests that $\mathcal{O}_{{}^G\tilde{\mathbf{R}}^{\upsilon}_{I_k}}$ has full column rank. Therefore, the observability matrix $\mathcal{O}_{{}^G\tilde{\mathbf{R}}^{\upsilon}_{I_k}}$ satisfies the rank condition in Lemma 1—that is, the error state ${}^G\tilde{\mathbf{R}}^{\upsilon}_{I_k}$ is locally weakly observable.

Condition 1 is easily satisfied under real situations, provided that the quantity of points accumulated from LiDAR in a scan commonly exceeds 1000. This is supported by the performance of currently widely used LiDAR models, such as Livox MiD-70, MiD-100, Ouster OS0, OS1, and Velodyne HDL series, all of which can obtain feature point clouds exceeding 1000. Moreover, in indoor spaces, these LiDAR models can acquire relatively accurate echoes for all point clouds, ensuring that the available features also meet the required quantity. In addition, except for a few specific structures like completely round and smooth pipes, the environment often contains more than two non-parallel planes, such as walls, that can be used in estimators.

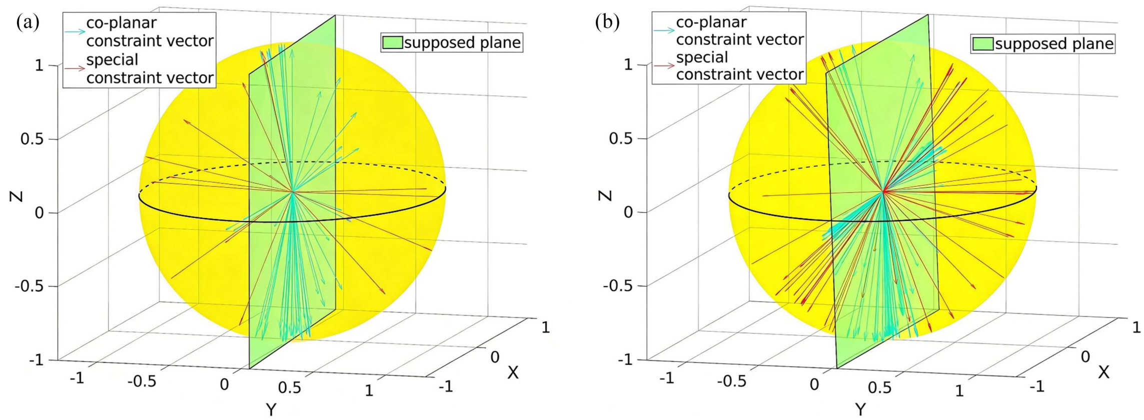

Figure 3 (Color online) Validation of the proposed criterion. (a) Normal vector distribution; (b) vector distribution in an invalidation point.

3.2.2 *Observability of position*

Similarly, the block related to ${}^{G}\tilde{\mathbf{p}}^{v}_{L_k}$ is extracted from $\mathcal{O}$—that is, $\mathbf{H}$ defined in (4):

$$\mathcal{O}_{{}^{G}\tilde{\mathbf{p}}^{v}_{I_k}} = \begin{bmatrix} \frac{\partial h_1}{\partial\left({}^{G}\tilde{\mathbf{p}}^{v}_{I_k}\right)} \\ \frac{\partial h_2}{\partial\left({}^{G}\tilde{\mathbf{p}}^{v}_{I_k}\right)} \\ \vdots \\ \frac{\partial h_n}{\partial\left({}^{G}\tilde{\mathbf{p}}^{v}_{I_k}\right)} \end{bmatrix} = \begin{bmatrix} \mathbf{u}_1^{\mathrm{T}} \\ \mathbf{u}_2^{\mathrm{T}} \\ \vdots \\ \mathbf{u}_n^{\mathrm{T}} \end{bmatrix}. \tag{13}$$

From (5), the row rank of block $\mathcal{O}_{{}^{G}\tilde{\mathbf{p}}^{v}_{I_k}}$ in Figure 2(a) is no more than 2. Therefore, blocks with a size of $n \times 3$ cannot satisfy the full column rank condition of locally weak observability in Lemma 1. In Figure 2(b), three normal vectors ($\mathbf{u}_m, \mathbf{u}_n$, and $\mathbf{u}_k$) that are not coplanar can make the rank of $[\mathbf{u}_m\ \mathbf{u}_n\ \mathbf{u}_k]^{\mathrm{T}}$ be 3. Then, the column rank of $\mathcal{O}_{{}^{G}\tilde{\mathbf{p}}^{v}_{I_k}}$ is also 3 because its rank has to be higher than that of the extracted block. In this case, the error state is locally weakly observable. Therefore, the condition that the error state ${}^{G}\tilde{\mathbf{p}}^{v}_{L_k}$ is locally weakly observable is provided below.

Condition 2. There are at least three corresponding planes whose normal vectors are linearly independent.

Overall, in the normal situation in Figure 2(b), the pose solution is unique at any point of the tunnel because of the constraints of the vertical components. Conditions 1 and 2, as well as the criterion in Proposition 1, are all satisfied. In contrast, multiple solutions appear in the abnormal situation in Figure 2(a). Condition 2 and the criterion are not satisfied. This completes the proof of Proposition 1.

To validate the criterion, the vector distribution is recorded at two different typical regions. Figure 3(b) shows the distribution in an outside environment with diverse features, whereas Figure 3(a) shows the distribution in a certain indoor weak-feature region where error drift happens. Apparently, most vectors in Figure 3(a) are not coplanar and can introduce various constraints along the tunnel axis, whereas most vectors in Figure 3(b) tend to fit onto one plane. This comparison indicates that the proposed criterion can distinguish between two situations effectively, forming the basis of subsequent remedies.

3.3 Reinforced vector module design

Given the conditions in Subsection 3.2, the key to distinguishing normal and abnormal cases is identifying the supposed fitting plane. In the fitting process, the SVD technique is utilized. For brevity, we only introduce the necessary procedure, and interested readers can refer to Mandel [44] for details. First, the average point $(\bar{x}, \bar{y}, \bar{z})$ is considered to lie on the fitting plane. Then, using the average point and all available points, the coefficient matrix $\mathbf{A} \in \mathbb{R}^{n\times 3}$ can be constructed [44]. Thereafter, one can decompose $\mathbf{A}$ as

$$\mathbf{A} = \mathbf{D\Sigma V}^{\mathrm{T}}, \tag{14}$$

where $\mathbf{D} \in \mathbb{R}^{n\times n}$ and $\mathbf{V} \in \mathbb{R}^{3\times 3}$ are both unitary matrices and $\mathbf{\Sigma} \in \mathbb{R}^{n\times 3}$ is a diagonal matrix. Finally, the last column of $\mathbf{V}$ is taken as the normal vector of the fitting plane. Using the normal vector and the average point, the fitting plane can be reconstructed.

To evaluate the fitting result, we define the coplanarity distance

$$d_i = \mathbf{u}_i \cdot \mathbf{n}_\alpha, \tag{15}$$

where $\mathbf{u}_i \in \mathbb{R}^3$ denotes the i-th constraint vector, $\mathbf{n}_\alpha \in \mathbb{R}^3$ denotes the normal vector of the supposed plane α, and $\cdot$ denotes the dot product of vectors to get the projection distance of $\mathbf{u}_i$ to α.

Unlike most plane-fitting problems, the constraint vectors are all specified only by their directions. Therefore, vectors can be moved to a shared origin $O(0,0,0)$ on the fitting plane. When the normal vector $\mathbf{n}_\alpha \in \mathbb{R}^3$ is obtained, the fitting plane α can be constructed as

$$\mathbf{n}_\alpha : [a,b,c]^{\mathrm{T}} \Rightarrow \alpha : ax + by + cz = 0. \tag{16}$$

The matrix $\mathbf{A}$ in (14) stands for the following coefficient matrix:

$$\mathbf{A} \cdot \mathbf{n}_\alpha = \mathbf{0} \in \mathbb{R}^n. \tag{17}$$

By the shared origin O of vectors, $\mathbf{A}$ can be simplified as

$$\mathbf{A} = \begin{bmatrix} x_1 - \bar{x}, y_1 - \bar{y}, z_1 - \bar{z} \\ x_2 - \bar{x}, y_2 - \bar{y}, z_2 - \bar{z} \\ x_3 - \bar{x}, y_3 - \bar{y}, z_3 - \bar{z} \\ \dots \\ x_n - \bar{x}, y_n - \bar{y}, z_n - \bar{z} \end{bmatrix} = \begin{bmatrix} x_1, y_1, z_1 \\ x_2, y_2, z_2 \\ x_3, y_3, z_3 \\ \dots \\ x_n, y_n, z_n \end{bmatrix}. \tag{18}$$

At this point, we can introduce Algorithm 1.

Algorithm 1 Abnormal case detection.

Input: All corresponding planes used in each point's estimator $\{a,b,c|\exists : ax+by+cz+1=0\}$; threshold ϵ_0;
1: Normalize all vectors $[a,b,c]^{\mathrm{T}}$ as $\mathbf{u}_i$;
2: Compose the matrix $\mathbf{A}$ by adding rows $\mathbf{u}_i$ using (18);
3: Perform SVD on $\mathbf{A}$ using (14);
4: Take the last column of $\mathbf{V}$;
5: Construct plane α using (16);
6: Calculate d_i for each vector using (15);
if $d_i \leqslant \epsilon_0$ holds for all vectors **then**
7: Completely abnormal situation, exit;
else
8: Determine SCVs ($\mathbf{u}_s$) with $d_i > \epsilon_0$;
end if
Output: Conclusion on whether all vectors satisfy $d_i \leqslant \epsilon_0$; plane α and SCVs ($\mathbf{u}_s$).

Suppose that some SCVs remain effective in the next iteration, indicating that the situation is not the extreme case. The next task is to identify and retain as many useful vectors as possible, rather than ignoring them. According to Remark 1, to fully utilize these SCVs, their corresponding noise value in $\mathbf{R}$ should be appropriately shortened to maximize a specific estimator's effect. This approach limits the loss of effective vectors in the current iteration. The detailed strategy is to minimize the noise value and multiply the coefficient k_0 as

$$K = k_0 \times \frac{size(\mathbf{u}_s)}{size(\mathbf{u}_c)}. \tag{19}$$

Here, $size$ denotes the overall quantity of specific vectors, and k_0 is a tunable parameter for different situations. Then, Algorithm 2 is provided to reinforce the constraint vectors.

3.4 Repaired vector module design

In this section, minor corrections are applied to the coplanar estimators' vectors. According to prior knowledge, constraint vectors generated by surrounding walls in straight weak-feature environments are coplanar. However, these vectors may be contaminated by noise, leading to deviations from the truly correct direction. Therefore, those vectors that can come from coplanar constraints must be modified to avoid optimizing the pose in the wrong direction. The repaired constraint vector, adjusted by a forced bias to pull the deviated vector back toward the correct direction, is rewritten as

Algorithm 2 Reinforcement of constraint vectors.

Input: All corresponding planes used in each point's estimator $\{a, b, c|\exists : ax + by + cz + 1 = 0\}$, the last calculated α and $\mathbf{u}_s$ from Algorithm 1, coefficient k_0, threshold ϵ_0;
1: Normalize all $[a, b, c]^{\mathrm{T}}$ vectors as $\mathbf{u}_i$;
2: Initialize $\alpha_0 = \alpha$ and $k = 0$;
3: **repeat**
4: Compute d_i with α_k for all vectors except $\mathbf{u}_s$ using (15);
5: Delete rows in $\mathbf{A}$ from $\mathbf{u}_s$ using (18);
6: Execute steps 3–8 in Algorithm 1;
7: Take the output α as α_{k+1};
8: $k = k + 1$;
9: **until** $d_i \leqslant \epsilon_0$ for all vectors except $\mathbf{u}_s$;
10: Determine the estimator $\mathbf{z}_i$ where $\mathbf{u}_s$ comes;
11: Calculate the coefficient K using (19) and update block $\mathbf{R}$;
Output: Updated matrix $\mathbf{R}$.

$$\mathbf{u}' = (\mathbf{u} \cdot \mathbf{n}_\alpha) \cdot \mathbf{n}_\alpha. \tag{20}$$

This correction ensures that subsequent pose optimization is based on reliable constraint information, enhancing the accuracy and stability of the overall system. The entire workflow is given in Algorithm 3, including both reinforcement and repair of the original odometry.

Algorithm 3 RV-LIO.

Input: Each point's estimator in FAST-LIO2, threshold ϵ_0 and ϵ_1 with $\epsilon_1 \leqslant \epsilon_0$, the output from algorithm 1;
if $d_i \leqslant \epsilon_0$ for any vector **then**
1: Exit;
else
2: Execute Algorithm 2;
end if
3: Check d_i for all vectors using (15);
if there are certain vectors $\mathbf{u}_c$ such that $d_i \leqslant \epsilon_1$ **then**
4: Repair vectors near the plane and obtain $\mathbf{u}'$ using (20);
5: Determine the estimator $\mathbf{z}_i$ where $\mathbf{u}_c$ comes;
6: Update rows in $\mathbf{H}$ with $\mathbf{u}'$ and recalculate residual $\mathbf{z}_i$;
end if
7: Go to the original IESKF in odometry;
Output: The improved odometry.

Remark 3. Notably, the RV-LIO system, like other LIO-based approaches, is inherently limited in completely smooth pipeline environments with entirely featureless surfaces. Specifically, in the case of completely circular cross-sections, the axial attitude remains unobservable within the algorithm. In another scenario involving axial translation in such featureless environments, the system also fails to achieve observability. This is fundamentally due to the absence of identifiable features that can constrain axial movement or attitude, leaving the degrees of freedom unconstrained. However, such extreme scenarios are rare in practical applications, with most real-world environments having distinguishing characteristics.

Remark 4. Recent learning-based SLAM methods that leverage neural radiance fields (NeRFs), 3D Gaussian splatting, or deep regression have significantly advanced dense reconstruction by jointly optimizing camera pose and scene appearance. These approaches benefit from data-driven priors, enabling robust tracking in texture-rich environments by hallucinating geometry in occluded regions and producing photorealistic renderings even from sparse viewpoints. However, their performance degrades considerably in weakly textured or geometrically feature-scarce scenes. For instance, Turki et al. [45] observed that NeRF-based reconstructions suffer from scale ambiguity when depth or pose initialization is inconsistent. Similarly, Sucar et al. [46] noted that neural implicit SLAM systems exhibit metric scale drift in the absence of loop closures. Cho et al. [47] also highlighted that learned LiDAR odometry accumulates scale errors due to insufficient structural cues. These limitations are intrinsic to learning-based methods: implicit representations can silently stretch or compress the absolute scale as long as the depth reprojection error remains below the optimization threshold, leaving the metric gauge unconstrained. Even with additional loss terms or postprocessing refinement, the metric scale is only loosely preserved. In contrast, our approach enforces metric scale explicitly through direct geometric constraints. As such, directly comparing our method with learning-based approaches would conflate appearance-driven reconstruction accuracy with metric-aware localization performance. Therefore, in the following section, to ensure a fair evaluation of scale consistency under metrically grounded references, the proposed RV-LIO is benchmarked against several existing methods that enforce scale through explicit observation constraints.

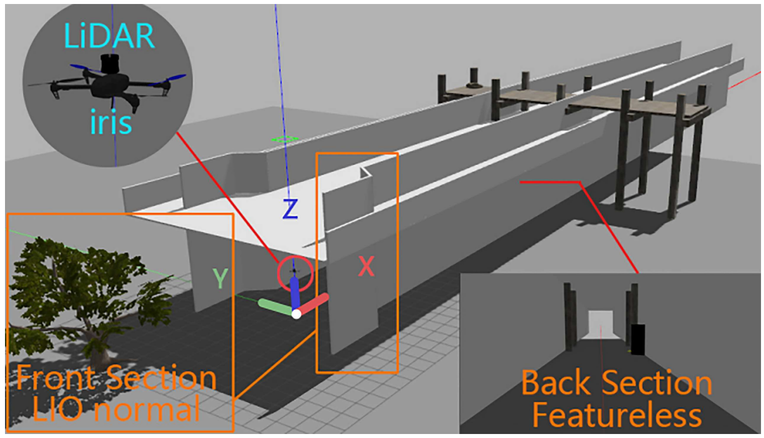

Figure 4 (Color online) Gazebo simulation environment.

4 Simulation and experimental results

4.1 Accuracy evaluation in Gazebo simulation

The simulation environment was based on Gazebo 9. The system setup is shown in Figure 4, which includes an Ouster OS1-32 LiDAR installed on the top of a quadrotor Iris and a 10.45 m $\times$ 7.8 m $\times$ 100.5 m weak-feature building. Data processing was performed using ROS Noetic Ninjemys. All simulations were executed on a laptop equipped with an Intel Core i7-13700H CPU, NVIDIA GeForce RTX 3050Ti GPU, and 256 GB DDR4 RAM, ensuring sufficient computational resources for real-time data processing. In this simulation, a UAV was used to explore the tunnel using both RV-LIO and other existing methods (e.g., FAST-LIO2 [20], Faster-LIO [48], Point-LIO [49], LINS [17], LeGO-LOAM [13], A-LOAM [12], and LIO-SAM [19]). The environment could be divided into front and back parts. Some complex items like trees and unstructured walls were set near the entrance of the tunnel. Most algorithms could work normally in the front part because of its diverse features, but comparatively fewer features in the back part of the tunnel invalidated the existing algorithms. According to the commands from the offboard node, Iris successively flew over the following waypoints: $[0, 0, 0]$, $[0, 0, 2.5]$, $[30, 0, 2.5]$, $[30, 0, 4.5]$, and $[50, 0, 4.5]$. The marked shock in this process was caused by the switch among states, such as the transition from hovering to forward acceleration. These shocks pose challenges to localization in tunnel environments.

To determine suitable design parameters adapted to the characteristics of tunnel environments, pre-experimental calibration was employed. Specifically, the system was first operated over a short segment of the tunnel along the aforementioned waypoints, during which key metrics related to localization performance and state transition stability were sampled and recorded. By comparing the algorithm's performance under different candidate values based on the sampled data, the optimal parameters that best adapted to the current environment were selected. According to the above procedure, the design parameters in the simulation were determined as $\epsilon_0 = 0.510303$, $\epsilon_1 = 0.11303$, and $k_0 = 695.07$.

The localization outputs of the odometries are compared in Figure 5(a), where the ground truth is given by Gazebo. The figure shows that the proposed odometry worked effectively in the back part. The maximum bias of RV-LIO was about 0.15 m, whereas the original algorithm experienced degradation in the middle part of the tunnel scenario. This degradation could be attributed to the environmental characteristics and noise impact. As shown in Figure 4, in the middle section of the tunnel, feature constraints at both ends were insufficient. In addition, though there was valid information in the middle segment, noise overwhelmed the few correct features and their direction vectors.

As an additional evaluation criterion, the root mean square errors (RMSEs) of absolute trajectory errors (ATEs) and relative pose errors (RPEs) are computed and reported in Table 1. RV-LIO increased the accuracy of existing studies in the simulations, attributed to its ability to address the aforementioned issues. By strengthening the role of direction vectors, RV-LIO effectively amplified the influence of correct features. Meanwhile, using the constraint vector correction mechanism, it avoided erroneous pose correction caused by noise to a certain extent, thus achieving lower errors.

The mapping results of RV-LIO and its baseline algorithm (FAST-LIO2) are further compared in Figure 5(b). The error drift and disorder were obvious in the conventional odometry's map, which stemmed from the baseline's vulnerability to feature scarcity and noise. When correct features were overwhelmed, it failed to maintain pose

Table 1 RMSEs of ATE and RPE in the simulations. × denotes that the system totally failed or the odometry went beyond the measurement range. The best results are in bold.

	RV-LIO	FAST-LIO2	Faster-LIO	Point-LIO	LINS	LeGO-LOAM	A-LOAM	LIO-SAM
ATE (m)	**0.055**	4.776	1.311	2.728	14.266	8.564	14.480	×
RPE of translation (m)	**0.024**	0.113	0.039	0.702	2.256	0.640	0.124	×
RPE of rotation (rad)	**0.002**	**0.002**	0.009	0.049	0.064	0.041	0.004	×

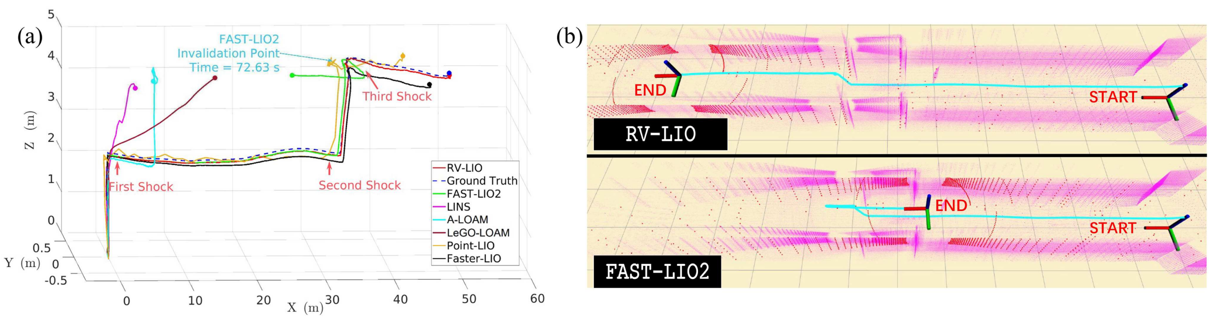

Figure 5 (Color online) Trajectories and mapping comparisons in the simulations. The shock represents the switch of motion states. (a) Trajectories of RV-LIO and existing methods; (b) comparison of the mapping results of RV-LIO and the baseline algorithm (FAST-LIO2).

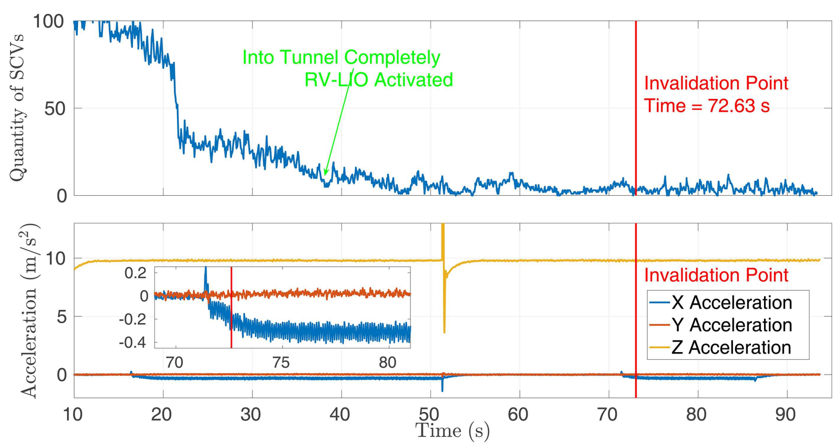

Figure 6 (Color online) Profiles of SCVs and IMU acceleration in Gazebo simulation.

accuracy, leading to distorted mapping. In contrast, our proposed odometry exhibited consistent mapping, ensuring stable pose estimation even in challenging conditions.

The profiles of SCVs' quantity and IMU are presented in Figure 6. There exist diverse SCVs (i.e., $\mathbf{u}_s$ before 39.28 s), under which the odometry worked well. However, as these features diminished, the baseline algorithm became unstable. This was because the diminishing SCVs, combined with noise, left the baseline without sufficient reliable constraints, making it prone to errors. Consequently, the error rose in 72.63 s because of a wrong initial pose caused by the sudden movement, as evident in the acceleration profile along the x-axis. In contrast, RV-LIO, by enhancing direction vectors and correcting constraints, remained robust even with fewer SCVs, avoiding such error surges.

To further clarify the contribution of each module in RV-LIO, additional analyses of their individual impacts were conducted. The reinforced vector module played a crucial role in enhancing performance. By strengthening the role of direction vectors, it effectively amplified the influence of correct features—the key factor behind the significant reduction in trajectory errors compared with the baseline. This was particularly evident in the middle tunnel segment, where the reinforced vectors ensured that the algorithm maintained a stable reference for pose estimation even with sparse valid features. Meanwhile, the repaired vector module contributed to improved robustness. In scenarios with rough wall surfaces (simulated by introducing irregular reflection noise) or radar sensors with large depth measurement noise, this module prevented the filter from converging toward erroneous directions. For instance, when noise levels spiked in the simulation, the repaired vectors corrected deviated constraints, avoiding abrupt error increases that plagued the baseline.

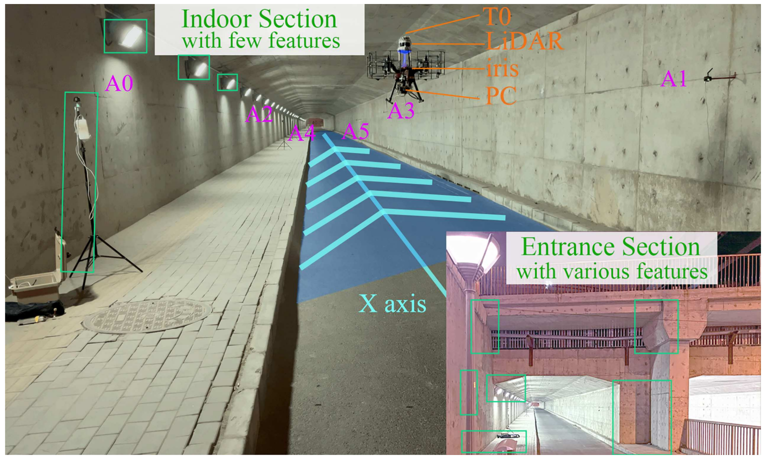

Figure 7 (Color online) Real experiment environment. A0–A5 represent the UWB anchors, and T0 represents the UWB target. For safety, the drone was manually controlled by the pilot at a speed of 0.6 m/s and a height of 1.85 m. Sensor measurements were collected.

4.2 Accuracy evaluation in real experiments

The platform in the real experiments comprised an Ouster OS0-32 LiDAR, an ICM-40609D IMU, an onboard computer, six Nooploop UWB anchors, and one UWB tag. The Ouster OS0-32 LiDAR featured a 360° horizontal field of view and a $-15°$ to $+15°$ vertical field of view. It had a maximum detection range of 120 m, with a point rate of up to 2.4 million points per second and a frame rate of 10 Hz. The ranging accuracy was ± 3 cm at 10 m, and the angular resolutions were 0.35° (horizontal) and 0.35° (vertical). The ICM-40609D IMU, a six-axis sensor, had the following relevant parameters: gyroscope noise of 4.5 mdps/$\sqrt{\text{Hz}}$, gyroscope bias stability TC of ± 10 mdps/°C, and gyroscope sensitivity error of $\pm 0.5\%$. The acceleration noise was 100 μg/$\sqrt{\text{Hz}}$, the acceleration bias stability TC was ± 0.15 mg/°C, and the acceleration sensitivity error was $\pm 0.5\%$. The Nooploop UWB system, which consisted of six anchors and one tag, offered a positioning accuracy of ± 10 cm in 3D space. It operated in the 6.5 GHz frequency band, with a communication range of up to 100 m in line-of-sight conditions and an update rate of 10 Hz.

The 5.8 m $\times$ 5.0 m $\times$ 225.5 m tunnel in Figure 7 was selected as the experimental environment. Similar to the earlier findings, various features in the front part made the odometry work normally, whereas error drift occurred in the back part because of the sparse features in the weak-feature region. Meanwhile, shocks induced by the motion state change resulting from manual control also posed challenges to localization. The flight distance was chosen as it exceeded the existing algorithms' capabilities to handle error drift. The design parameters were set to $\epsilon_0 = 0.290713, \epsilon_1 = 0.29071$, and $k_0 = 0.053$. Six anchors in the UWB positioning system were deployed at four corners and two midpoints of the longer sides.

Because of the anchors' layout and positioning algorithm [50, 51], the resolved accuracy in the length axis was much higher than that in the width axis. This discrepancy stemmed from the geometric distribution of the UWB anchors: their placement along the length of the experimental area created a more robust triangulation network for position estimation in this axis, whereas the relatively sparse coverage in the width direction limited the system's ability to resolve fine positional details. The positioning algorithm, optimized for scenarios with elongated anchor layouts, further enhanced precision in the length axis by leveraging more redundant measurements and stronger signal strengths from the aligned anchors.

In addition, for the weak-feature shape, odometry invalidation only occurred in the length axis (i.e., the x-axis in Figure 7). This axis-specific failure was tied to the environmental characteristics of weak-feature regions: the lack of distinct structural cues along the length direction reduced the odometry's ability to constrain drift, whereas the presence of occasional lateral features (even in sparse form) helped maintain stability in the width axis. Consequently, the primary source of error in odometry performance, and thus the key metric for evaluation, was concentrated in the x-axis. Therefore, the position output of UWB along the x-axis was taken as the reference value to be compared with the odometry results.

During actual operation, the program exhibited efficient resource utilization: memory consumption remained below 40% of the available capacity, which ensured a sufficient margin to handle sudden data spikes or concurrent tasks without causing memory bottlenecks. This is crucial for maintaining consistent performance in dynamic tunnel environments, where sensor data streams (e.g., point clouds from LiDAR and motion data from IMU) are continuously processed. In terms of CPU usage, the program was restricted to no more than four cores. This not only prevented excessive CPU occupation, which could lead to thermal throttling or system slowdowns, but also

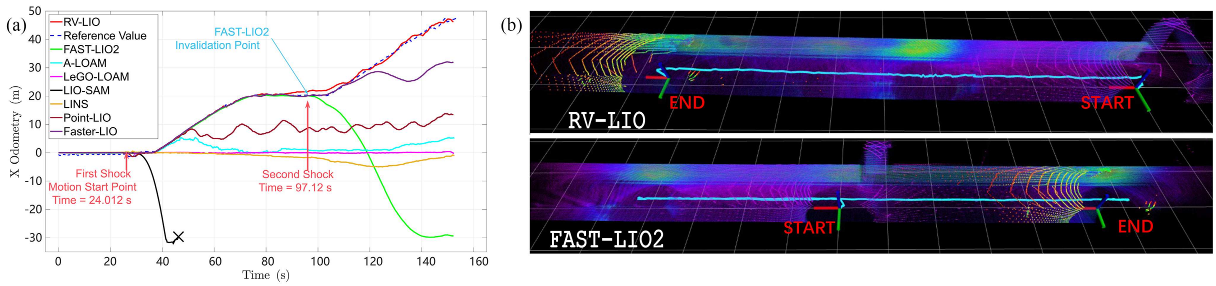

Figure 8 (Color online) Comparisons of odometry and mapping results in a real experiment. (a) Odometry comparison of RV-LIO and existing methods; (b) mapping results of RV-LIO and the baseline algorithm (FAST-LIO2).

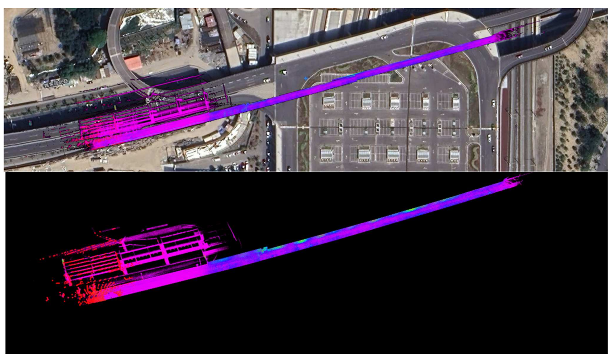

Figure 9 (Color online) Large-scale mixed-scene experiment results.

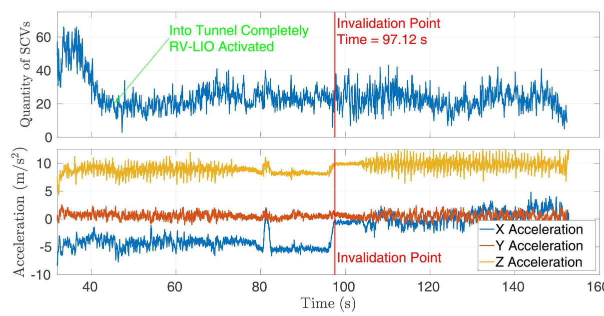

Figure 10 (Color online) Profiles of SCVs and IMU acceleration in real experiments.

allowed the remaining cores to handle other critical operations, such as communication with the UWB system or execution of control algorithms for the UAV. Such controlled resource usage underscores the practical feasibility of the proposed system for real-time applications, as it balances computational demands with the hardware constraints of onboard computers.

Localization and mapping comparisons are shown in Figure 8. The error drift made the other odometries generate distorted values and become completely invalid at the first or second shock time. In contrast, the proposed odometry coincided with the reference value given by UWB until the termination time. The maximum bias of the proposed RV-LIO remained below 0.43 m compared with 10.21 m for Faster-LIO, the best among the other odometries. In addition, the map of RV-LIO was mainly accurate with satisfactory performance, whereas the map of the baseline odometry had significant disorder and overlay.

We evaluated five other sequences collected in the weak-feature region. Besides the default 60 m region shown in Figure 8, the algorithm was tested in the 80, 100, 150, and 200 m tunnel regions. Specifically, to check the activation module, a large-scale experiment with a length of 350 m was additionally conducted that included both a featureless weak-feature scene and a normal scene with various features. These six sequences either had a good reference position truth provided by Linktrack P-B UWB (tag-anchor position mode) or a precise distance measurement to the starting position collected by P452 UWB (tag-tag distance mode). The theoretical accuracy of P-B positioning was 0.25 m in our measurement range of 100 m. The theoretical distance measurement accuracy of P452 (manufactured by Time Domain Corporation) was 0.1 m in its effective measurement range of 500 m. The RMSEs of the odometry positioning result or its point-to-start distance are evaluated and reported in Table 2. RV-LIO was more effective than the existing methods in all six sequences. The map built by RV-LIO in the large-scale experiment is shown in Figure 9, showing small end-to-end drift (i.e., 0.51 m) and a good agreement with the satellite map.

The quantities of SCVs are further shown in Figure 10. The constraint vectors tended to be in one plane after 42.35 s, when the proposed RV-LIO was activated at this time. As shown in Figure 8(a), the conventional odometry became unstable and eventually led to failure at 97.12 s due to the inaccurate initial pose given by the IMU integration between the long-lasting hovering and the quick forward motion. In contrast, the proposed system ensured high positioning accuracy. The experimental results demonstrate that RV-LIO can handle error drift in weak-feature regions effectively.

Table 2 RMSEs (m) of ATE in real experiments. Sequences 1–3 had a good reference truth from the positioning result of Linktrack P-B, while Sequences 4–6 used the distance measurement of P452 as the reference truth distance to the starting position for its good performance for long distances. Sequence 6 included a 208 m tunnel and an outside pedestrian zone of 150 m. The best results are in bold.

Sequence	Length (m)	RV-LIO	FAST-LIO2	Faster-LIO	Point-LIO	LINS	LeGO-LOAM	A-LOAM	LIO-SAM
1	60	**0.7319**	1.7283	2.4531	5.6737	5.9970	27.5595	2.4643	12.7977
2	80	**0.9777**	2.9681	3.6239	9.2544	34.5393	6.2136	7.0921	24.8367
3	100	**0.9435**	4.8152	4.8576	11.4708	55.2898	12.1923	6.6798	35.3012
4	150	**2.2998**	48.7945	51.3355	47.8678	31.1861	4.5768	16.2713	49.4389
5	200	**3.2465**	55.6961	50.4919	57.3539	47.7416	7.1413	20.5467	57.4887
6	350	**3.7829**	72.3849	52.7366	70.4964	29.5449	37.1874	15.7811	48.1362

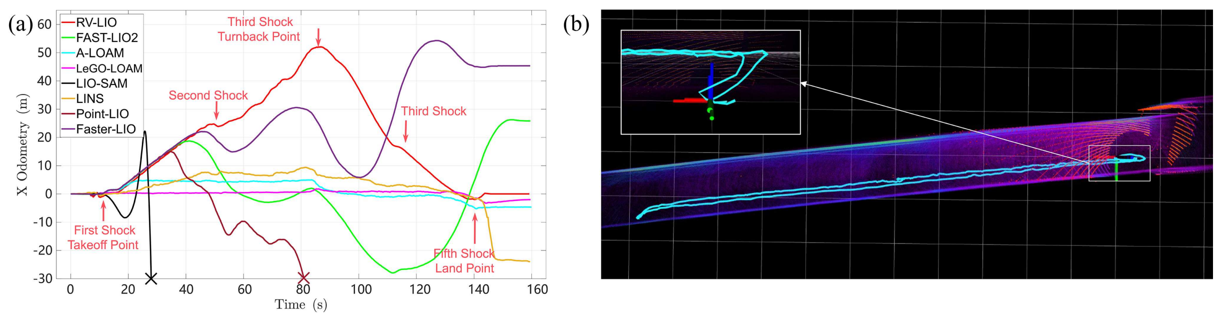

Figure 11 (Color online) Odometry and mapping results in loopback motion where the drone was controlled to fly into the tunnel, turn back, and land on the takeoff zone. In the odometry results (a), comparisons of RV-LIO and existing methods are shown. In the mapping results (b) of RV-LIO, the larger and smaller axes represent the starting and terminal points, respectively.

4.3 Accuracy evaluation in loopback motion experiment

To further validate the accuracy and effectiveness of the proposed RV-LIO, we performed an extensive experiment in loopback motion in terms of the accumulated error. Under a pilot's control, a drone first flew to a spot about 50 m away from the entrance, then turned around and flew back to the takeoff zone. Considering that the loopback algorithm can enhance accuracy significantly, the starting and terminal points of RV-LIO were compared directly without the loopback module to evaluate the accumulated error.

The odometry results are presented in Figure 11(a). Most existing methods failed before the farthest point because of shocks. Considering that only the RV-LIO worked without failure and coincided with the known loopback motion, its mapping results are provided in Figure 11(b). The initial starting and terminal points were $[0, 0, 0]$ and $[0.0595, -0.0075, 0.0405]$, respectively. These results further demonstrate that RV-LIO can mitigate odometry failure in state-of-the-art algorithms and work effectively with little accumulated error.

5 Conclusion

This paper proposes a new RV-LIO system to tackle odometry invalidation in straight tunnel-like environments. A necessary criterion is provided to activate the system when error drift occurs with rigorous observability problems. The reinforced vector module is designed as the core of the entire system to improve localization accuracy. Meanwhile, the repaired vector module further enhances system robustness. Both Gazebo simulation and real experiments are conducted to validate the effectiveness of the proposed system. The results indicate that RV-LIO can be activated at the correct region to prevent error drift and ensure high localization and mapping accuracy. Considering that the parameters are sensitive in RV-LIO, some adaptive algorithms may be implemented in the future to improve our approach. Moreover, some dynamic measurements in aerial robotics may be considered to enhance its robustness and effectiveness in extreme cases with no features.

Acknowledgements This work was supported in part by National Natural Science Foundation of China (Grant Nos. 62373055, 62403052, 62425301, W2511069), Fundamental and Interdisciplinary Disciplines Breakthrough Plan of the Ministry of Education of China (Grant No. JYB2025XDXM206), and Young Scientists Foundation of CSAA (Guidance Navigation and Control, GNC) (Grant No. CSAA-YSF2025-GNC-20).

References

1 Usamentiaga R. Semiautonomous pipeline inspection using infrared thermography and unmanned aerial vehicles. IEEE Trans Ind Inf, 2024, 20: 2540–2550

2 Hua C S, Qi J T, Shang H, et al. Detection of collapsed buildings with the aerial images captured from UAV. Sci China Inf Sci, 2016, 59: 032102

3 Wang C, Wang W, Dong W, et al. Multiple-stage spatial-temporal cooperative guidance without time-to-go estimation. Chin J Aeronautics, 2024, 37: 399–416

4 Lyu Y, Pan Q, Hu J W, et al. Obstacle avoidance under relative localization uncertainty. Sci China Inf Sci, 2019, 62: 084201

5 Smith R C, Cheeseman P. On the representation and estimation of spatial uncertainty. Int J Robot Res, 1986, 5: 558–567

6 Dissanayake G, Durrant-Whyte H, Bailey T. A computationally efficient solution to the simultaneous localisation and map building (SLAM) problem. In: Proceedings of 2000 IEEE International Conference on Robotics and Automation, 2000. 1009–1014

7 Durrant-Whyte H, Bailey T. Simultaneous localization and mapping: part I. IEEE Robot Automat Mag, 2006, 13: 99–110

8 Chen Y, Wu Y, Zhang Z, et al. Image-based visual servoing of unmanned aerial manipulators for tracking and grasping a moving target. IEEE Trans Ind Inf, 2022, 19: 8889–8899

9 Li Q, Queralta J P, Gia T N, et al. Multi-sensor fusion for navigation and mapping in autonomous vehicles: accurate localization in urban environments. Un Sys, 2020, 08: 229–237

10 Zhang S, Cui R, Yan W, et al. Dual-layer path planning with pose SLAM for autonomous exploration in GPS-denied environments. IEEE Trans Ind Electron, 2023, 71: 4976–4986

11 Hou X L, Li Z Y, Pan Q. Autonomous navigation of a multirotor robot in GNSS-denied environments for search and rescue. Sci China Inf Sci, 2023, 66: 139203

12 Zhang J, Singh S. LOAM: LiDAR odometry and mapping in real-time. In: Proceedings of Robotics: Science and Systems, 2014. 1–9

13 Shan T, Englot B. LeGO-LOAM: lightweight and ground-optimized lidar odometry and mapping on variable terrain. In: Proceedings of 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2018. 4758–4765

14 Segal A, Haehnel D, Thrun S. Generalized-ICP. In: Proceedings of Robotics: Science and Systems, 2009. 1–8

15 Tagliabue A, Tordesillas J, Cai X, et al. LION: lidar-inertial observability-aware navigator for vision-denied environments. In: Proceedings of International Symposium on Experimental Robotics, 2020. 380–390

16 Ye H, Chen Y, Liu M. Tightly coupled 3D lidar inertial odometry and mapping. In: Proceedings of 2019 International Conference on Robotics and Automation, 2019. 3144–3150

17 Qin C, Ye H, Pranata C E, et al. LINS: a lidar-inertial state estimator for robust and efficient navigation. In: Proceedings of 2020 IEEE International Conference on Robotics and Automation, 2020. 8899–8906

18 Li K, Li M, Hanebeck U D. Towards high-performance solid-state-LiDAR-inertial odometry and mapping. IEEE Robot Autom Lett, 2021, 6: 5167–5174

19 Shan T, Englot B, Meyers D, et al. LIO-SAM: tightly-coupled LiDAR inertial odometry via smoothing and mapping. In: Proceedings of 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2020. 5135–5142

20 Xu W, Cai Y, He D, et al. FAST-LIO2: fast direct LiDAR-inertial odometry. IEEE Trans Robot, 2022, 38: 2053–2073

21 Qin T, Li P, Shen S. VINS-Mono: a robust and versatile monocular visual-inertial state estimator. IEEE Trans Robot, 2018, 34: 1004–1020

22 Zhang H, Jin L, Ye C. An RGB-D camera based visual positioning system for assistive navigation by a robotic navigation aid. IEEE CAA J Autom Sin, 2021, 8: 1389–1400

23 Yang X, Yuan Z, Zhu D, et al. Robust and efficient RGB-D SLAM in dynamic environments. IEEE Trans Multimedia, 2020, 23: 4208–4219

24 Yang B B, Huang Z Y, Li Y J, et al. Hybrid3D: learning 3D hybrid features with point clouds and multi-view images for point cloud registration. Sci China Inf Sci, 2023, 66: 172101

25 Júnior G P C, Rezende A M C, Miranda V R F, et al. EKF-LOAM: an adaptive fusion of LiDAR SLAM with wheel odometry and inertial data for confined spaces with few geometric features. IEEE Trans Autom Sci Eng, 2022, 19: 1458–1471

26 Huang H, Zhang M, Li L, et al. GTSCalib: generalized target segmentation for target-based extrinsic calibration of non-repetitive scanning LiDAR and camera. IEEE Trans Automat Sci Eng, 2024, 22: 3648–3660

27 Nielsen K, Hendeby G. Survey on 2D lidar feature extraction for underground mine usage. IEEE Trans Automat Sci Eng, 2023, 20: 981–994

28 Quenzel J, Nieuwenhuisen M, Droeschel D, et al. Autonomous MAV-based indoor chimney inspection with 3D laser localization and textured surface reconstruction. J Intell Robot Syst, 2019, 93: 317–335

29 Lee K W, Wijesoma W S, Guzman J I. On the observability and observability analysis of SLAM. In: Proceedings of 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2006. 3569–3574

30 Huang G P, Mourikis A I, Roumeliotis S I. Observability-based rules for designing consistent EKF SLAM estimators. Int J Robotics Res, 2010, 29: 502–528

31 Hu C, Wang Y X, Wang R, et al. An improved radar detection and tracking method for small UAV under clutter environment. Sci China Inf Sci, 2019, 62: 029306

32 Wang S, Wang S. UWB radar imaging method for the longwall retreat and creep displacements measurement in underground coal mine. IEEE Trans Automat Sci Eng, 2024, 21: 4573–4584

33 Ozaslan T, Loianno G, Keller J, et al. Autonomous navigation and mapping for inspection of penstocks and tunnels with MAVs. IEEE Robot Autom Lett, 2017, 2: 1740–1747

34 Tian Y, Chang Y, Arias F H, et al. Kimera-Multi: robust, distributed, dense metric-semantic SLAM for multi-robot systems. IEEE Trans Robot, 2022, 38: 2022–2038

35 Zhong S, Qi Y, Chen Z, et al. DCL-SLAM: a distributed collaborative LiDAR SLAM framework for a robotic swarm. IEEE Sens J, 2023, 24: 4786–4797

36 Lajoie P Y, Ramtoula B, Chang Y, et al. DOOR-SLAM: distributed, online, and outlier resilient SLAM for robotic teams. IEEE Robot Autom Lett, 2020, 5: 1656–1663

37 Xie Y, Zhang Y, Chen L, et al. RDC-SLAM: a real-time distributed cooperative SLAM system based on 3D LiDAR. IEEE Trans Intell Transp Syst, 2021, 23: 14721–14730

38 Cunningham A, Paluri M, Dellaert F. DDF-SAM: fully distributed SLAM using constrained factor graphs. In: Proceedings of 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2010. 3025–3030

39 Li X, Zhang H, Chen W. 4D radar-based pose graph SLAM with ego-velocity pre-integration factor. IEEE Robot Autom Lett, 2023, 8:

5124–5131
40 Karam S, Lehtola V, Vosselman G. Simple loop closing for continuous 6DOF LIDAR&IMU graph SLAM with planar features for indoor environments. ISPRS J Photogrammetry Remote Sens, 2021, 181: 413–426
41 Doherty K J, Rosen D M, Leonard J J. Spectral measurement sparsification for pose-graph SLAM. In: Proceedings of 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2022. 1–8
42 Hermann R, Krener A. Nonlinear controllability and observability. IEEE Trans Automat Contr, 1977, 22: 728–740
43 Kelly J, Sukhatme G S. Visual-inertial sensor fusion: localization, mapping and sensor-to-sensor self-calibration. Int J Robotics Res, 2011, 30: 56–79
44 Mandel J. Use of the singular value decomposition in regression analysis. Am Statistician, 1982, 36: 15–24
45 Turki H, Ramanan D, Satyanarayanan M. Mega-NERF: scalable construction of large-scale NeRFs for virtual fly-throughs. In: Proceedings of 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022. 12922–12931
46 Sucar E, Liu S, Ortiz J, et al. iMAP: implicit mapping and positioning in real-time. In: Proceedings of 2021 IEEE/CVF International Conference on Computer Vision, 2021. 6229–6238
47 Cho Y, Kim G, Kim A. Unsupervised geometry-aware deep LiDAR odometry. In: Proceedings of 2020 IEEE International Conference on Robotics and Automation, 2020. 2145–2152
48 Bai C, Xiao T, Chen Y, et al. Faster-LIO: lightweight tightly coupled lidar-inertial odometry using parallel sparse incremental voxels. IEEE Robot Autom Lett, 2022, 7: 4861–4868
49 He D, Xu W, Chen N, et al. Point-LIO: robust high-bandwidth light detection and ranging inertial odometry. Adv Intell Syst, 2023, 5: 2200459
50 Dong W, Li Y, Sheng X, et al. Trajectory estimation of a flying robot with a single ranging beacon and derived velocity constraints. IEEE Trans Ind Electron, 2022, 70: 5024–5033
51 Zhao S, Huang B, Liu F. Localization of indoor mobile robot using minimum variance unbiased FIR filter. IEEE Trans Automat Sci Eng, 2016, 15: 410–419