MSC-LIO: An MSCKF-Based LiDAR-Inertial Odometry with Same-Plane-Point Tracking

Tisheng Zhang, Man Yuan, Linfu Wei, Hailiang Tang*, and Xiaoji Niu

Abstract—**The multi-state constraint Kalman filter (MSCKF) has been proven to be more efficient than graph optimization for visual-based odometry while with similar accuracy. However, it has not yet been properly considered and studied for LiDAR-based odometry. In this paper, we propose a novel tightly coupled LiDAR-inertial odometry based on the MSCKF framework, named MSC-LIO. An efficient LiDAR same-plane-point (LSPP) tracking method, without explicit feature extraction, is present for frame-to-frame data associations. The tracked LSPPs are employed to build an LSPP measurement model, which constructs a multi-state constraint. Besides, we propose an effective point-velocity-based LiDAR-IMU time-delay (LITD) estimation method, which is derived from the proposed LSPP tracking method. To validate the effectiveness and robustness of the proposed method, we conducted extensive experiments on both public datasets and real-world environments. The results demonstrate that the proposed MSC-LIO yields higher accuracy and efficiency than the state-of-the-art methods. The ablation experiment results indicate that the data-association efficiency is improved by nearly 3 times using the LSPP tracking method, and the proposed LITD estimation method can effectively and accurately estimate the LITD. Besides, the MSC-LIO was implemented on an edge device and demonstrated excellent real-time performance.**

Index Terms—**State estimation, light detection and ranging (LiDAR)-inertial odometry, multi-state constraint Kalman filter (MSCKF), multi-sensor fusion navigation.**

I. Introduction

L IGHT detection and ranging (LiDAR) has played an increasingly important role in autonomous vehicles and robots, mainly due to its spatial perception capabilities and the rapid development of low-cost solid-state LiDARs. Meanwhile, the inertial measurement unit (IMU) can be employed for navigation independently, and the inertial navigation system (INS) can output high-frequency poses. Hence, the INS can be adopted to correct the motion distortion of point clouds. For this

reason, the LiDAR and the IMU have been integrated to construct the LiDAR-inertial odometry (LIO) for more accurate pose estimation. LIOs can be categorized into optimization-based LIOs and filter-based LIOs according to the type of the state estimator. Generally, graph optimization is more accurate than filters but less efficient [1]. The multi-state constraint Kalman filter (MSCKF) [2] has been proven to achieve similar accuracy to graph optimization while more efficient in visual-inertial odometry (VIO) [3], [4]. Recently, some studies [5], [6], [7] have explored the application of MSCKF in LIOs and proved that the MSCKF achieves higher accuracy than other filters for LiDAR-inertial navigation. Nevertheless, the data-association method and the LiDAR measurement model in MSCKF-based LIOs should still be further studied for efficient and accurate state estimation.

A. Related Works

1) Optimization-Based Methods

PGO-LIOM [8] developed a tightly coupled LIO with a gradient-free optimization method, which achieved low time consumption. The tightly-coupled LIO proposed in [9] built a local map using LiDAR feature points within a local window and employed LiDAR measurements and IMU pre-integration to optimize the states within the window. LIO-SAM [10] constructed a LIO based on a factor graph and used a sliding window-based scan-matching approach to achieve real-time performance. GIVL-SLAM [11] designed a sliding-window-based factor graph optimization (FGO) estimator to fuse IMU measurements, LiDAR features, visual features, and global navigation satellite system (GNSS) [12] observations. LIPS [13] proposed an anchor plane factor for graph-based optimization that could be associated with multiple LiDAR frames. VILENS [14] tracked plane and line features from LiDAR frames and then constructed plane and line landmark factors for graph optimization. However, the features in [11] and [12] were explicitly extracted, which was inefficient and not suitable for unstructured environments. BA-LINS [15] constructed a frame-to-frame (F2F) LiDAR BA measurement model within the FGO framework. The BA measurements were constructed with same-plane points from keyframes within a sliding window, and the same-plane points were obtained by direct point-cloud processing. However, the above methods adopt optimization frameworks, requiring multiple iterations and leading to poor efficiency.

Unlike VIOs, LIOs can eliminate the need to estimate extra landmark states with large linearization errors. Hence, filters should achieve the same accuracy as optimization for LIOs.

2) Filter-Based Methods

LINS [16] employed an iterated error-state Kalman filter (IESKF) to construct a tightly-coupled LIO, yielding higher efficiency than optimization-based LIO-SAM [10]. FAST-LIO [17] proposed a tightly coupled LIO using the IESKF and introduced a new formula with reduced computational complexity for computing the Kalman gain. On the basis of FAST-LIO [17], FAST-LIO2 [18] developed an incremental k-d tree to represent the global map efficiently and directly register raw points to the map, enabling more robust scan registration in unstructured scenes. The adaptive voxel map proposed in [19] consisted of voxels, each containing a plane feature that probabilistically represents the environment. The voxel map was integrated into the IESKF to construct a LiDAR(-inertial) odometry with high robustness and efficiency. VoxelMap++ [20] expanded on the work in [19] by consolidating coplanar features into larger planes, thereby reducing uncertainty in the overall map and improving state estimation accuracy. Additionally, the efficiency of plane covariance estimation was enhanced through least squares estimation. The filter-based methods mentioned above use the IESKF for state estimation, exhibiting higher efficiency than optimization-based methods. However, they all employ the frame-to-map (F2M) association, leading to incorrect absolute measurements and inconsistent state estimation.

In contrast, the F2F association constructs relative measurements and ensures consistent state estimation. Consequently, LIOs with the F2F association can be tightly coupled with other absolute positioning sensors, such as the GNSS and ultrawideband (UWB) [21]. In [22], the consistent state estimation for LIO was achieved by scan-to-scan matching within the IESKF framework, and thus the global positioning system (GPS) measurements were seamlessly fused to achieve localization and mapping in a large-scale environment. Inv-LIO1 [23] proposed an invariant extended Kalman filter state estimator for LIO and found the feature correspondence between two consecutive scans by scan-to-scan match method using extracted features and IMU measurements. LIC-Fusion [6] efficiently fused IMU measurements and LiDAR features (plane and edge features) within the MSCKF framework. The extracted features in the current LiDAR frame were tracked back to the previous frame to construct the F2F measurement model. The above methods only constructed the F2F associations between two LiDAR frames. In contrast, LIC-Fusion 2.0 [7] adopted a normal-based method to associate plane features and stored plane features as SLAM plane landmarks. Each plane landmark could be associated with multiple frames, achieving a multi-state F2F constraint. Besides, LIC-Fusion 2.0 proposed a plane-feature tracking method for higher efficiency, which was designed specifically for spinning LiDARs. However, the explicit extraction of plane features in LIC-Fusion 2.0 demands significant computational resources and is only effective in structured scenarios.

The multi-state relative pose constraints in BA-LINS [15] and LIC-Fusion 2.0 [7] are similar to the bundle adjustment (BA) in visual multiple-view geometry [24], which has been proven to be more accurate in LiDAR mapping by BALM [25].

BALM is designed for LiDAR mapping rather than odometry. BA-LINS is an optimization-based method exhibiting poor efficiency. Although LIC-Fusion 2.0 is based on the MSCKF framework, it requires the explicit feature extraction of plane features, which increases computational costs notably. Besides, LIC-Fusion 2.0 augments plane landmark states into the state vector, resulting in poor state-estimation efficiency. Hence, the high-efficiency potential of the MSCKF have not yet been fully employed in the existing LIOs, and the LiDAR measurement model still should be further studied for accurate MSCKF state estimation.

B. Main Contributions

In this study, we propose an MSCKF-based LIO with LiDAR same-plane point (LSPP) tracking, named MSC-LIO. We first track the LSPPs in the latest LiDAR keyframe to construct an LSPP cluster. Then, the LSPP clusters are utilized for plane fitting and constructing the F2F LSPP measurement model. Finally, the IMU measurements and LSPP measurements are tightly integrated within the MSCKF framework. The main contributions of this study are as follows:

1) We present an MSCKF-based LIO that tightly integrates the IMU measurements and LSPP measurements with online calibration of LiDAR-IMU spatiotemporal parameters.

2) An F2F LSPP tracking method without explicit feature extraction is proposed. The LSPPs are tracked frame by frame within a sliding window, yielding improved data-association efficiency.

3) We propose a novel method to estimate the LiDAR-IMU time delay (LITD) based on the LSPP velocity. The point velocity can be approximately calculated by the LSPP tracking.

4) We conduct comprehensive experiments on both public datasets and real-world environments to evaluate the performance of the proposed MSC-LIO. The results demonstrate that the proposed method outperforms the state-of-the-art (SOTA) methods in terms of accuracy and efficiency.

II. SYSTEM OVERVIEW

The system overview of the proposed MSC-LIO is shown in Fig. 1. We adopt an INS-centric processing pipeline. The initial position and heading angle of the IMU are initialized to zero, while the roll and pitch angles are determined by the accelerometer measurements and gravity acceleration. If a zero-velocity state is detected during initialization, we can also obtain an initial gyroscope bias using the mean gyroscope measurements. Once the system is initialized and the IMU input is received, the INS mechanization is performed, and the state vector and its covariance are forward propagated. In the meantime, the INS pose is stored for further point-cloud preprocessing. When a LiDAR frame is received, the point-cloud distortion is corrected with the high-frequency INS pose. Besides, the INS prior pose is also employed for keyframe selection. If the relative motion or time interval between the current and the previous keyframe exceeds the set threshold, the current frame is considered a keyframe.

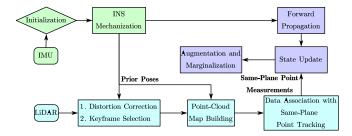When a keyframe is selected, all non-keyframes between the

Fig. 1. The system overview of the proposed MSC-LIO.

previous keyframe and the current keyframe are projected and merged into the current keyframe to construct the keyframe point-cloud map. Subsequently, the LSPP tracking is performed between the historical keyframe point-cloud maps and the current. More specifically, the LSPP candidates in the historical keyframe point-cloud maps are utilized to track the LSPPs in the current keyframe point-cloud map. The LSPP tracking is conducted by searching for nearest neighboring points instead of explicit plane feature extraction. The tracked LSPPs are employed to construct the F2F LSPP measurement model. Then, the LSPP measurements are utilized to update the IMU state, the keyframe states, and the LiDAR-IMU spatiotemporal parameters within the MSCKF framework. Finally, the current keyframe pose state is augmented into the MSCKF state vector. The marginalization is conducted when the sliding window exceeds its maximum length.

III. MSCKF-BASED LiDAR-INERTIAL ESTIMATOR

A. State Vector

The error state vector of the MSCKF includes the IMU state $\delta \boldsymbol{x}_I$, the LiDAR-IMU extrinsic parameter state $\delta \boldsymbol{x}_I^b$, the LiDAR-IMU time delay state δt_d, and the keyframe states $\delta \boldsymbol{x}_i (i = 0, 1, \cdots, N-1)$, where N is the length of the sliding window. The error state $\delta \boldsymbol{x}$ can be written as

$$\delta \boldsymbol{x} = [\delta \boldsymbol{x}_I, \delta \boldsymbol{x}_I^b, \delta t_d, \delta \boldsymbol{x}_0, \delta \boldsymbol{x}_1, \cdots, \delta \boldsymbol{x}_{N-1}]^T, \quad (1)$$

where $\delta \boldsymbol{x}_I$, $\delta \boldsymbol{x}_I^b$, and $\delta \boldsymbol{x}_i$ are expressed as

$$\delta \boldsymbol{x}_I = [\delta \boldsymbol{\theta}_{b_N}^w, \delta \boldsymbol{p}_{b_N}^w, \delta \boldsymbol{v}^w, \delta \boldsymbol{b}_g, \delta \boldsymbol{b}_a], \quad (2)$$

$$\delta \boldsymbol{x}_I^b = [\delta \boldsymbol{\theta}_I^b, \delta \boldsymbol{p}_I^b], \quad (3)$$

$$\delta \boldsymbol{x}_i = [\delta \boldsymbol{\theta}_{b_i}^w, \delta \boldsymbol{p}_{b_i}^w]. \quad (4)$$

Here, w, b, l represent the world frame, the IMU frame, and the LiDAR frame, respectively; $\delta \boldsymbol{\theta}_{b_N}^w$, $\delta \boldsymbol{p}_{b_N}^w$ and $\delta \boldsymbol{v}^w$ denote the errors of current IMU attitude, position, and velocity, respectively; $\delta \boldsymbol{b}_g$ and $\delta \boldsymbol{b}_a$ denote the bias errors of the gyroscope and accelerometer, respectively; $\delta \boldsymbol{\theta}_I^b$ and $\delta \boldsymbol{p}_I^b$ denote the errors of LiDAR-IMU extrinsic parameters; $\delta \boldsymbol{\theta}_{b_i}^w$ and $\delta \boldsymbol{p}_{b_i}^w$ denote the IMU pose errors at the time of keyframe i, which is denoted as KF_i. The relationship between the true state $\boldsymbol{x}$, estimated state $\hat{\boldsymbol{x}}$, and error state $\delta \boldsymbol{x}$ is

$$\boldsymbol{x} = \hat{\boldsymbol{x}} \boxplus \delta \boldsymbol{x}. \quad (5)$$

For the attitude error $\delta \boldsymbol{\theta}$, the operator $\boxplus$ is given by

$$\mathbf{R} = \hat{\mathbf{R}} \mathrm{Exp}(\delta \boldsymbol{\theta}) \approx \hat{\mathbf{R}}(\mathbf{I} + (\delta \boldsymbol{\theta}) \times), \quad (6)$$

where $\mathbf{R}$ and $\hat{\mathbf{R}}$ denote the true and estimated rotation matrix, respectively; Exp is the exponential map [26]; $(\cdot) \times$ denotes the skew-symmetric matrix of the vector belonging to $\mathbb{R}^3$ [27]. For other states, the operator $\boxplus$ is equivalent to Euclidean addition, i.e., $\boldsymbol{a} = \hat{\boldsymbol{a}} + \delta \boldsymbol{a}$. The LITD t_d is employed to adjust the LiDAR time t_{LiDAR} for synchronization with IMU time t_{IMU}:

$$t_{IMU} = t_{LiDAR} + t_d. \quad (7)$$

When receiving the IMU measurement, INS mechanization is conducted to update the IMU pose and velocity, and the standard error-state Kalman filter (ESKF) prediction formula [28] is employed for the forward propagation of the MSCKF error state and its covariance.

B. LiDAR Same-plane-point Measurement Model

The tracked LiDAR same-plane point clusters are used to construct the LSPP measurement model. The LSPP cluster with index j is denoted as $S_j = \{\boldsymbol{p}_k^l \in \mathbb{R}^3 \mid k = n_i^j \in [0, N], i = 0, 1, \cdots, n_j - 1\}$, where $\boldsymbol{p}_k^l$ represents the same-plane point belonging to KF_k and n_j is the number of same-plane points in S_j. All the same-plane points are expressed in the LiDAR frame.

Considering an LSPP cluster S_j, each same-plane point is first projected onto the world frame with the LiDAR-IMU extrinsic parameters and the IMU pose as

$$\boldsymbol{p}_k^w = \hat{\mathbf{R}}_{b_k}^w (\hat{\mathbf{R}}_l^b \boldsymbol{p}_k^l + \hat{\boldsymbol{p}}_l^b) + \hat{\boldsymbol{p}}_{b_k}^w. \quad (8)$$

Then, the plane fitting is conducted with the projected points by solving an overdetermined linear equation [29]. The normalized normal vector of the plane is denoted as $\boldsymbol{n}$, and d is the distance that satisfies the following equation:

$$\boldsymbol{n}^T \boldsymbol{p}^w + d = 0, \quad (9)$$

where $\boldsymbol{p}^w$ is a point on the plane in the w frame. The LSPP measurement is the mean of the sum of squares of the point-to-plane distances:

$$\boldsymbol{z}^j = \frac{1}{n_j} \sum_k (\boldsymbol{n}^T \boldsymbol{p}_k^w + d)^2. \quad (10)$$

C. LiDAR Measurement Update

In the proposed method, only the LSPP clusters containing the point from the current keyframe are used for the update. The same-plane points in S_j belong to the same plane, and thus point-to-plane distances are zeros in the absence of errors. Hence, the residual $\boldsymbol{r}^j$ can be written as

$$\boldsymbol{r}^j = 0 - \boldsymbol{z}^j \approx \mathbf{H}_x^j \delta \boldsymbol{x} + \boldsymbol{n}_r^j, \quad (11)$$

where $\mathbf{H}_x^j$ is the Jacobian w.r.t to $\delta \boldsymbol{x}$, and $\boldsymbol{n}_r^j \in \boldsymbol{N}(0, \boldsymbol{\Sigma}_\Gamma^j)$ is the noise. The adaptive covariance $\boldsymbol{\Sigma}_\Gamma^j$ will be detailed in Section III.A.

The residual $\boldsymbol{r}^j$ is the function of the poses and the

LiDAR-IMU extrinsic parameters. Therefore, we can derive the corresponding analytical Jacobians using the error-perturbation method [30]. The Jacobians w.r.t the keyframe states not associated with the measurement z^j are zeros, and the Jacobians w.r.t the pose errors $\{\delta\theta_{b_k}^w, \delta p_{b_k}^w\}$ can be formulated as

$$\frac{\partial r^j}{\partial \delta\theta_{b_k}^w} = -\mathbf{J}_k^j \hat{\mathbf{R}}_{b_k}^w (\hat{\mathbf{R}}_l^b p_k^l + \hat{p}_l^b)\times \quad (12)$$

$$\frac{\partial r^j}{\partial \delta p_{b_k}^w} = \mathbf{J}_k^j, \quad (13)$$

where

$$\mathbf{J}_k^j = \frac{2}{n_j}(n^T p_k^w + d)n^T. \quad (14)$$

Similarly, the Jacobians w.r.t the LiDAR-IMU extrinsic errors $\{\delta\theta_l^b, \delta p_l^b\}$ can be formulated as

$$\frac{\partial r^j}{\partial \delta\theta_l^b} = -\sum_k \mathbf{J}_k^j \hat{\mathbf{R}}_{b_k}^w \hat{\mathbf{R}}_l^b (p_k)\times, \quad (15)$$

$$\frac{\partial r^j}{\partial \delta p_l^b} = \sum_k \mathbf{J}_k^j \hat{\mathbf{R}}_{b_k}^w. \quad (16)$$

The time delay t_d is not considered in this section and will be discussed in Section III.B. With the analytical Jacobians, the MSCKF error state and covariance can be updated with the standard ESKF update formula [28].

D. State Augmentation and Marginalization

The proposed MSC-LIO does not explicitly extract plane features, eliminating the need for null-space projection. Thus, the adopted MSCKF method only requires additional state augmentation and marginalization compared with the standard Kalman filter with fixed states. When receiving a new LiDAR keyframe, the IMU state is corrected, and then the IMU pose state is augmented into the state vector. Meanwhile, the covariance $\mathbf{P}_{n\times n}$ is augmented. The Jacobian of the augmented pose state w.r.t the state vector is $\mathbf{J}_{6\times n}$. The augmented covariance can be written as

$$\mathbf{P}_{(n+6)\times(n+6)} = \begin{bmatrix}\mathbf{I}_{n\times n}\\\mathbf{J}_{6\times n}\end{bmatrix}\mathbf{P}_{n\times n}\begin{bmatrix}\mathbf{I}_{n\times n}\\\mathbf{J}_{6\times n}\end{bmatrix}^T = \begin{bmatrix}\mathbf{P} & \mathbf{PJ}^T\\\mathbf{JP}^T & \mathbf{JPJ}^T\end{bmatrix}. (17)$$

The state and covariance of the oldest keyframe will be directly deleted when the sliding window exceeds its maximum length, that is, the marginalization [1].

IV. LiDAR Same-Plane-Point Tracking

A. F2F Same-plane-point Tracking Method

The proposed LSPP tracking is performed within the keyframe point-cloud maps, which are downsampled by a voxel-grid filter (with a default voxel size of 0.5 m). Since the plane features are not explicitly extracted, all points in the point-cloud maps are candidates for same-plane points. The specific procedure for LSPP tracking is illustrated in Fig. 2. For each LSPP cluster S_j, the newest point is projected onto the

current keyframe point-cloud map, and then five neighboring
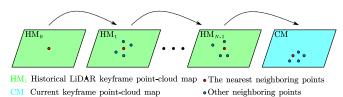

Fig. 2. An illustration of the plane-based same-plane point tracking.

points $\{p_m \mid m = 1,2,\cdots,5\}$ are searched. The five points are used to fit a plane (n_j, d_j). If each point-to-plane distance is less than the set threshold (0.1 m), the tracking is considered valid. In this case, the nearest neighboring point is added to S_j for tracking in subsequent keyframes, and the plane thickness [15] is calculated as

$$\Gamma = \frac{1}{5}\sum_{m=1}^5 (n_l^T p_m + d_l)^2. \quad (18)$$

For S_j, the covariance of the plane thickness $\boldsymbol{\Sigma}_\Gamma^j$ and the adaptive standard deviation (STD) of the point-to-plane distance σ^j [15] can be obtained by

$$\boldsymbol{\Sigma}_\Gamma^j = \frac{1}{n_j}\sum_k (\Gamma_k^2)\mathbf{I}, \quad \sigma^j = \sqrt[4]{0.5\boldsymbol{\Sigma}_\Gamma^j}. \quad (19)$$
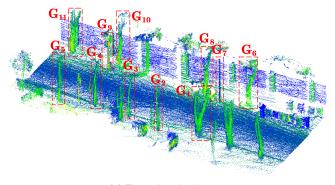
To ensure the validity of the LSPP measurement, only the LSPP clusters that contain at least 5 points and are associated with the current keyframe are considered as candidates for valid measurements. The points in S_j are projected onto the w frame and employed to fit a plane, and the point-to-plane distances are calculated. S_j is used to construct the LSPP measurement model only when all the point-to-plane distances are less than $3\sigma^j$.

Fig. 3. shows the results of the proposed tracking method, and each LSPP cluster is represented by a distinct color. There are many tree trunks and other pole-like objects on both sides of the road, marked with G_i in the point-cloud map and tracking schematics. Most of the same-plane points associated with tree trunks and pole-like objects in Fig. 3 are successfully tracked, indicating the effectiveness of the LSPP tracking. Notably, only the LSPP clusters associated with all keyframes in the sliding window are shown in Fig. 3. For clarity, the LSPP clusters associated with the right wall are not shown in Fig. 3 (c).
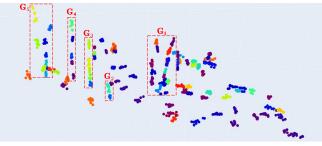
To ensure the stability of MSC-LIO, we supplement the LSPP clusters with new points. The current keyframe point-cloud map is voxelized with a much larger voxel size than 0.5 m, and only the point closest to each voxel center is retained to ensure uniform spatial distribution of the supplementary points. The retained points are employed to construct new LSPP clusters to ensure the robustness of the F2F association. Besides, the clusters not associated with the newest five keyframes will be removed. An overview of the tracking method is provided in Fig. 4.

B. LiDAR-IMU Time-Delay Estimation Based on Same-Plane-Point Velocity
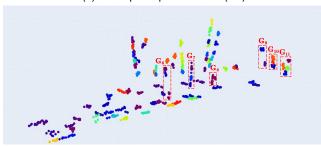
The traditional LITD estimation requires the angular rate and

(a) The point-cloud map.

(b) Same-plane point clusters (left).

(c) Same-plane point clusters (right).

Fig. 3. The tracked same-plane point clusters. Different clusters are represented by different colors. Those clusters associated with tree trunks and other pole-like objects are marked with G_i, to demonstrate the effectiveness of the tracking.

velocity of the IMU to compensate for each keyframe pose [31]. In contrast, our method only requires the point velocity to compensate for LiDAR points, which is simpler and more direct. The velocity of a plane point can be calculated by the relative motion between two keyframes. The LiDAR frames of the two adjacent keyframes are denoted as L_1 and L_2, and the plane point $\mathbf{P}$ is observed in both keyframes, as depicted in Fig. 5 (a). The coordinates of $\mathbf{P}$ in L_1 and L_2 are $\boldsymbol{p}^{L_1}$ and $\boldsymbol{p}^{L_2}$, respectively, and the velocity of $\mathbf{P}$ can be obtained by

$$\boldsymbol{v}_p = \boldsymbol{v}_p^{L_1} = \boldsymbol{v}_p^{L_2} = \frac{\boldsymbol{p}^{L_2} - \boldsymbol{p}^{L_1}}{\Delta t}, \quad (20)$$

where Δt is the time interval. However, the points observed in the two keyframes may not be the same due to the downsampling of point clouds, as shown in Fig. 5 (b). Hence, when conducting LSPP tracking, the distances between same-plane points are constrained within a threshold to ensure

the accuracy of the point velocity calculated by Equation (20).

For $\quad S_j = \{\boldsymbol{p}_k^l \in \mathbb{R}^3 \mid k = n_i^j \in [0, N], i = 0, 1, \cdots, n_j - 1\}$,
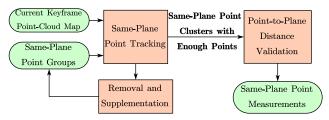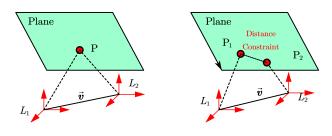

Fig. 4. The overview of the same-plane-point tracking method.

(a) The same point.   (b) Downsampled points.

Fig. 5. Point velocity calculation.

$\boldsymbol{p}_k^l$ belongs to the keyframe KF_k, and the velocity of $\boldsymbol{p}_k^l$ is denoted as $\boldsymbol{v}_{P_k}$. The estimated LITD of KF_k is $\hat{t}_{d_k}$, and the estimated KF_k pose $\{\hat{\mathbf{R}}_{b_k}^w, \hat{\boldsymbol{p}}_{b_k}^w\}$ is the IMU pose at $t_{IMU_k} = t_{LiDAR_k} + \hat{t}_{d_k}$. Specifically, the estimated LITD of the current keyframe KF_N is denoted as $\hat{t}_d = \hat{t}_{d_N}$. Hence, the difference between $\hat{t}_{d_k}$ and $\hat{t}_d$ is employed to compensate for $\boldsymbol{p}_k^l$ as follows:

$$\hat{\boldsymbol{p}}_k^l = \boldsymbol{p}_k^l - \boldsymbol{v}_{P_k}(\hat{t}_d - \hat{t}_{d_k}). \quad (21)$$

Then, $\hat{\boldsymbol{p}}_k^l$ is employed to construct the LSPP measurement by formula (8) and (10). The Jacobian w.r.t the LITD error is

$$\frac{\partial \boldsymbol{r}^j}{\partial \delta t_d} = -\sum_k \hat{\mathbf{J}}_k^j \hat{\mathbf{R}}_{b_k}^w \hat{\mathbf{R}}_l^b \boldsymbol{v}_{P_k}. \quad (22)$$

V. EXPERIMENTS AND RESULTS

A. Implementation and Datasets

We implement the proposed MSC-LIO in C++ and Robots Operating System (ROS). We compare the proposed MSC-LIO with other SOTA LIOs, including LIO-SAM (without loop closure) [10], FAST-LIO2 [18] and FF-LINS [21]. LIO-SAM and FF-LINS are optimization-based systems, while FAST-LIO2 is a filter-based system. The LiDAR-IMU extrinsic parameters and time delay are assumed unknown for all systems. The sliding window size for MSC-LIO and FF-LINS are all set to 10 to ensure a fair comparison. All systems are implemented on a desktop PC (AMD R7-3700X).

The used public datasets are the *MCD* [32] and *WHU-Helmet* [33] datasets. Six *KTH* sequences of the *MCD* dataset are selected, with a total trajectory length of 6786 m. The *KTH* sequences were collected using a handheld setup, which was

equipped with a spinning LiDAR, i.e., Ouster OS1, and a MEMS IMU. The *WHU-Helmet* dataset was collected by a
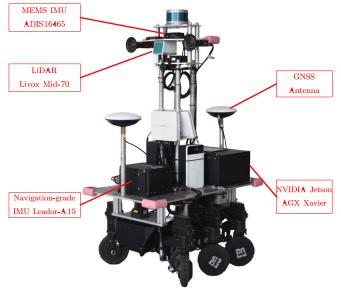

Fig. 6. The robot used for real-world experiments.

helmet-based system, which was equipped with a solid-state LiDAR, i.e., Livox AVIA, and a MEMS IMU. In the *WHU-Helmet* dataset, four sequences collected by Livox AVIA are employed, with a total trajectory length of 3656 m. Both the *MCD* and *WHU-Helmet* datasets provide high-precision reference truth.

To further evaluate MSC-LIO in real-world scenarios, we also conducted real-world experiments on a wheeled robot with a maximum speed of 1.5 m/s, as depicted in Fig. 6. The sensors used include a solid-state LiDAR, i.e., Livox Mid-70, and a MEMS IMU ADIS16465. The sensors are well-synchronized by using the pulse signal from the GNSS receiver. The integrated navigation solution of a navigation-grade [30] IMU Leador-A15 and high-precision GNSS real-time kinematic (RTK) serves as the ground truth, with position and attitude accuracy of 0.02 m and 0.01 deg, respectively [34]. Eight sequences with a total length of 15248 m were collected by the wheeled robot and denoted as the *RobNav* dataset. The eight experimental scenes include various structured and unstructured environments. In addition, dynamic objects, such as pedestrians, bicycles, and vehicles, make it challenging to achieve robust positioning. Four sequences among them are shown in Fig. 7. The information on the public datasets and the real-world Robnav dataset is summarized in TABLE I.

B. Evaluation of the Accuracy

1) Public MCD Dataset

The absolute translation errors (ATEs) are calculated using evo [35], as presented in TABLE II. LIO-SAM has a significant error on the *kth_nigth_01* sequence, mainly due to its inability to extract effective features in small indoor scenes. Due to the presence of many loop closures in the *MCD* dataset, FAST-LIO2 based on F2M association can establish associations with the self-built map, resulting in better accuracy compared to FF-LINS based on F2F association. MSC-LIO constructs an F2F multi-state constraint and thus achieves

higher accuracy compared to FF-LINS. Besides, MSC-LIO achieves the best accuracy on three sequences and the smallest

Fig. 7. Four scenes of the real-world experiments. Different colors denote different sequences.

TABLE I
DATASETS DESCRIPTIONS

Dataset	*MCD*	*WHU-Helmet*	*RobNav*
LiDAR Type	Spinning	Solid-state	Solid-state
LiDAR Rate	10Hz	10Hz	10Hz
LiDAR Line	64	6	1
IMU Rate	400Hz	600Hz	200Hz
Total Length	6786 m	3656 m	15248 m

TABLE II
ATEs ON THE *MCD* DATASET

ATE (m)	LIO-SAM	FAST-LIO2	FF-LINS	MSC-LIO
kth_day_06	0.84	0.46	0.57	**0.27**
kth_day_09	1.09	**0.17**	0.83	0.34
kth_day_10	1.22	**0.41**	0.51	0.45
kth_night_01	13.19	0.52	1.12	**0.33**
kth_night_04	0.47	**0.20**	0.38	0.29
kth_night_05	0.95	0.41	0.27	**0.25**
RMS	5.46	0.38	0.67	**0.33**

TABLE III
ATEs ON THE *WHU-HELMET* DATASET
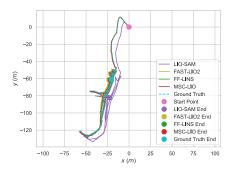
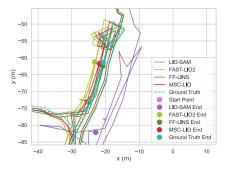| ATE (m)	LIO-SAM	FAST-LIO2	FF-LINS	MSC-LIO
mall	0.55	**0.32**	0.69	0.48
residence	**0.35**	1.03	0.43	0.45
street	1.06	0.90	0.97	**0.60**
subway	28.43	2.39	2.34	**1.82**
RMS	14.23	1.39	1.33	**1.01**

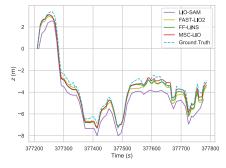root-mean-square (RMS) error on all six sequences.

2) Public WHU-Helmet Dataset

The ATEs are calculated on the *WHU-Helmet* dataset, as shown in TABLE III. LIO-SAM nearly fails in the indoor scene of the *subway* sequence, resulting in a great ATE. Benefiting from the multi-state constraints of LSPP measurements, the proposed MSC-LIO achieves the highest accuracy in the *street* and *subway* sequences. Moreover, the ATE RMS of the four sequences indicates that MSC-LIO outperforms other SOTA systems in terms of accuracy.

The results of the *residence* sequence are shown in Fig. 8. In Fig. 8 (a), the whole trajectory of LIO-SAM exhibits a significant drift compared to the ground truth trajectory, while the trajectories of other systems are well aligned with the truth.

(a) The whole trajectory.　　　　(b) The endpoint.　　　　(c) The height (z-axis) change.

Fig. 8. Results on the *WHU-Helmet residence* sequence.

Fig. 8 (b) and (c) show that near the endpoint, FF-LINS and MSC-LIO are closer to the ground truth than FAST-LIO2, which benefits from the consistent F2F association. MSC-LIO exhibits a smaller error than FF-LINS, especially in the z-axis, as MSC-LIO constructs accurate multi-state constraints.

3) *Real-World Experiments*

The absolute rotation errors (AREs) and ATEs are calculated, as presented in TABLE IV. LIO-SAM fails to run on the real-world experiments, i.e., the *RobNav* dataset, because the used Livox Mid-70 point clouds are sparse and difficult to extract features effectively. In contrast, other systems do not explicitly extract features, and thus they can run successfully. The AREs of FAST-LIO2 are mostly larger than 3 degrees because it cannot estimate the LiDAR-IMU extrinsic parameters. In contrast, FF-LINS and MSC-LIO can estimate the extrinsic parameters, resulting in much smaller AREs. MSC-LIO achieves the highest accuracy in most sequences, and the RMSs of ATEs and AREs are both at optimal levels.

The results of the *luojia_square* sequence are shown in Fig. 9. Fig. 9 (a) and (b) show that the trajectory of MSC-LIO is better aligned with the ground truth, while FF-LINS and FAST-LIO2 exhibit significant drifts in the horizontal and elevation directions, respectively. The attitude error is shown in Fig. 9 (c). FAST-LIO2 has large errors in the roll, pitch, and yaw angles due to the F2M association that may introduce incorrect observability. With consistent F2F association, FF-LINS and MSC-LIO exhibit high accuracy in the observable roll and pitch angles, though the yaw angle may diverge for unobservability. MSC-LIO exhibits higher accuracy in the yaw angle with a maximum error of 0.72 degrees because of the multi-state constraints of LSPP measurements.

C. Evaluation of the Efficiency

We compared the efficiency of MSC-LIO with FAST-LIO2, FF-LINS, and MSC-LIO w/o Tracking on the *RobNav* dataset. MSC-LIO w/o Tracking does not use the proposed LSPP tracking; instead, its data association works as follows. The current keyframe point clouds are sampled and voxel downsampled. Then, the obtained points are projected onto each historical keyframe point-cloud map within the sliding window. Finally, planes are fitted by searching for neighboring points, and the nearest neighboring points from all keyframes are employed to construct the LSPP measurements. The running time of data association and state estimation is recorded, as shown in TABLE V.

1) *Efficiency of Data Association*

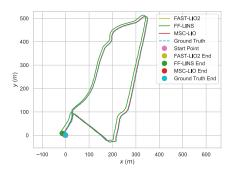Data association includes F2F or F2M feature association.
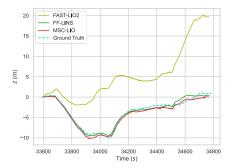
TABLE IV
AREs AND ATEs ON THE *RobNav* DATASET

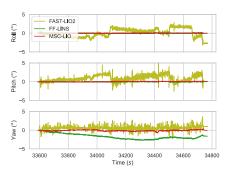| ARE / ATE (deg / m)	FAST-LIO2	FF-LINS	MSC-LIO
campus	3.55 / 4.42	**0.41** / 1.51	0.50 / **1.01**
building	3.13 / 3.12	0.65 / 1.90	**0.40** / **1.03**
playground	2.84 / 1.59	0.77 / 1.27	**0.55** / **0.75**
park	3.24 / 4.00	**0.90** / 1.44	0.97 / **1.25**
cs_campus	3.68 / 4.38	0.93 / 2.04	**0.84** / **1.67**
luojia_square	3.47 / 5.18	0.88 / 3.88	**0.22** / **0.92**
east_lake	3.20 / 4.49	1.48 / 8.39	**0.88** / **3.28**
library	3.28 / 2.73	**0.37** / **1.77**	0.39 / 1.93
RMS	3.31 / 3.90	0.86 / 3.58	**0.65** / **1.67**

For MSC-LIO, it consists of the preprocessing of LSPP clusters (removal and supplementation) and LSPP tracking. FAST-LIO2 and MSC-LIO spend less time in data association according to TABLE V. The reason is FF-LINS and MSC-LIO w/o Tracking perform nearest neighboring point search and plane fitting with all historical keyframes within the sliding window. In contrast, FAST-LIO2 only matches the current frame with the map, and MSC-LIO only matches the latest points in the LSPP clusters with the current keyframe. Although the preprocessing of LSPP clusters consumes extra time, the data association efficiency of MSC-LIO is still significantly improved, with the time spent approximately one-third of MSC-LIO w/o Tracking.

In fact, the data association time of MSC-LIO is shorter than FAST-LIO2 for the whole sequences, as MSC-LIO only processes the LiDAR frames at each keyframe moment, while FAST-LIO2 performs F2M association at each LiDAR frame moment. Due to the keyframe selection strategy, the average keyframe interval is 3-4 times of LiDAR frames on the *RobNav* dataset. This is why the running time for single data association and state estimation of MSC-LIO is longer than FAST-LIO2, but the total running time of MSC-LIO is shorter in TABLE V.

2) *Efficiency of State Estimation*

State estimation includes the prediction and update, or factor graph optimization. For MSCKF, it also includes extra state augmentation and marginalization. As shown in TABLE V, the state estimation time of the optimization-based FF-LINS is much longer than other filter-based methods. The state estimation time of MSC-LIO w/o Tracking and MSC-LIO is similar, as the number of plane features is close. The state

(a) The whole trajectory. (b) The height (z-axis) change. (c) The attitude error.

Fig. 9. Results on the *RobNav luojia_square* sequence.

TABLE V
THE AVERAGE RUNNING TIME PER KEYFRAME AND THE TOTAL TIME OF DATA ASSOCIATION AND STATE ESTIMATION

	FAST-LIO2			FF-LINS			MSC-LIO w/o Tracking			MSC-LIO		
	t_{DA} (ms)	t_{Est} (ms)	t_{Total} (s)	t_{DA} (ms)	t_{Est} (ms)	t_{Total} (s)	t_{DA} (ms)	t_{Est} (ms)	t_{Total} (s)	t_{DA} (ms)	t_{Est} (ms)	t_{Total} (s)
campus	5.2	0.7	65	17.1	38.7	198	15.3	10.8	92	5.5	11.2	59
building	5.1	0.7	106	15.8	42.1	58	14.8	11.0	153	5.0	10.6	92
playground	7.8	0.9	84	17.5	46.6	194	14.7	8.7	71	8.0	12.9	63
park	4.6	0.6	65	17.1	34.1	169	14.5	11.3	84	4.0	9.6	45
cs_campus	4.9	0.6	102	15.9	41.0	140	13.5	9.3	134	5.1	10.0	89
luojia_square	5.0	0.7	68	17.8	40.7	209	14.5	10.9	91	5.5	11.1	59
east_lake	4.3	0.7	72	15.9	32.4	269	12.7	8.8	109	3.5	7.9	59
library	5.2	0.7	95	18.3	42.7	316	15.2	11.0	135	5.5	11.3	87
Average	5.3	0.7	82	16.9	39.8	194	14.4	10.2	108	5.3	10.6	69

Here, t_{DA} and t_{Est} represents the average running time of data association and state estimation for each keyframe, respectively; t_{Total} represents the total time of data association and state estimation for the entire sequence.

estimation efficiency of FAST-LIO2 is higher than MSCKF-based methods, as its state vector dimension is lower. In contrast, the state vectors of MSCKF-based methods contain the keyframe states within the sliding window, and extra state augmentation and marginalization are also required.

D. Ablation Experiments

1) The Impact of the Same-Plane Point Tracking

In Section III.C, it has been demonstrated that the LSPP tracking improves the efficiency of data association. In this part, we will further evaluate its impact on accuracy. The statistical results of the number of planes, AREs, and ATEs on the *RobNav* dataset are shown in TABLE VI. The number of planes refers to the average number of LSPP clusters selected as valid measurements at each keyframe selection moment. The proposed LSPP tracking method improves the accuracy, with an average reduction of 17.7% in AREs and 31.8% in ATEs, while the average number of planes remains relatively unchanged. The improvement in accuracy stems not from the change in the number of planes but from the more uniform distribution of the tracked same-plane points.

2) The Impact of the Proposed LiDAR-IMU Time Delay Estimation Method

In this section, the impact of the LITD estimation method on accuracy is evaluated. The traditional LITD estimation methods use the LITD to compensate for keyframe poses, as detailed in [31]. MSC-LIO with the traditional LITD estimation method is denoted as MSC-LIO-PTD, and MSC-LIO without LITD estimation is denoted as MSC-LIO w/o TD. The *MCD* dataset has a relatively larger LITD, while the LITDs in the

WHU-Helmet and *RobNav* datasets are less than 2 ms. Hence, the *MCD* dataset is employed to evaluate the effect of LITD

TABLE VI
IMPACT OF SAME-PLANE POINT TRACKING ON THE NUMBER OF PLANES, AREs AND ATEs

	MSC-LIO w/o Tracking		MSC-LIO	
	plane number	ARE / ATE (deg / m)	plane number	ARE/ATE (deg / m)
campus	350	**0.36** / 1.15	**356**	0.50 / **1.01**
building	**361**	0.85 / 1.97	341	**0.40** / **1.03**
playground	267	0.75 / 0.88	**390**	**0.55** / **0.75**
park	**369**	1.34 / 2.18	291	**0.97** / **1.25**
cs_campus	305	**0.52** / **1.33**	**313**	0.84 / 1.67
luojia_square	**360**	0.38 / 1.97	343	**0.22** / **0.92**
east_lake	**285**	1.03 / 5.34	242	**0.88** / **3.28**
library	**360**	0.53 / **1.81**	359	**0.39** / 1.93
RMS	**334**	0.79 / 2.45	332	**0.65** / **1.67**
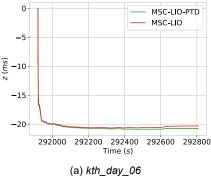
TABLE VII
IMPACT OF THE TIME-DELAY ESTIMATION METHODS ON ATEs
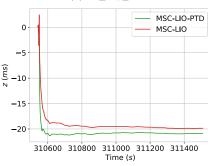
ATE (m)	MSC-LIO w/o TD	MSC-LIO-PTD	MSC-LIO
kth_day_06	0.78	**0.25**	0.27
kth_day_09	1.14	0.36	**0.34**
kth_day_10	2.65	0.46	**0.45**
kth_night_01	1.60	**0.33**	**0.33**
kth_night_04	1.13	0.35	**0.29**
kth_night_05	0.71	0.26	**0.25**
RMS	1.49	0.34	**0.33**

estimation methods.

Fig. 10 depicts the estimated LITDs on the *kth_day_06* and *kth_night_01* sequence. The estimated LITDs are close and have converged, verifying the feasibility of the proposed LITD

estimation method. The ATEs with different LITD estimation methods are shown in TABLE VII. The ATEs increase

(a) *kth_day_06*

(b) *kth_night_01*

Fig. 10. The LiDAR-IMU time delays of MSC-LIO-PTD and MSC-LIO. MSC-LIO-PTD uses the time delay to compensate for the pose of each keyframe.

significantly when the large LITD is not estimated. This is mainly because the LITD causes inaccuracy in the IMU poses used for point cloud distortion correction, resulting in low accuracy of LSPP measurements. The ATEs of MSC-LIO-PTD and MSC-LIO are similar for all sequences, indicating that the accuracy of the proposed point-velocity-based LITD estimation method is comparable to the traditional method. However, the proposed method is much simpler and more direct, as it employs only point velocity to compensate for the LiDAR points rather than using both the angular rate and velocity to compensate for all keyframe poses within the sliding window.

E. Real-Time Performance

To further test the real-time performance of MSC-LIO, we also implemented the MSC-LIO on an edge device NVIDIA jetson AGX Xavier, as shown in Fig. 6. The AGX Xavier features an 8-core ARM v8.2, 64-bit CPU with 8MB L2 and 4MB L3 cache, along with 16GB of 256-bit LPDDR4x memory running at 2133MHz, providing a bandwidth of 137GB/s. We collected two extra sequences in real-world environments by the wheeled robot shown in Fig. 6. One is the *information_campus* sequence (2012 s and 2565 m), and the other is the *academic_building* sequence (1152 s and 1457 m), as depicted in Fig. 11. The ground truth is obtained from the integrated navigation solution of Leador-A15 and GNSS-RTK as mentioned above.

We run the proposed MSC-LIO on the edge device NVIDIA Jetson AGX Xavier in real-time mode. The estimated trajectories are well-aligned with the ground truth, as shown in Fig. 12. The real-time performance including accuracy and

efficiency is shown in TABLE VIII. The small AREs and ATEs indicate that the proposed MSC-LIO exhibits high accuracy
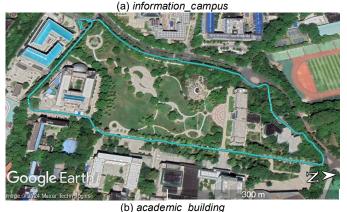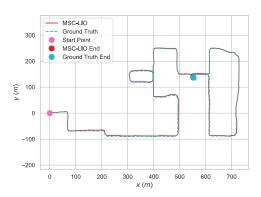

(a) *information_campus*

(b) *academic_building*
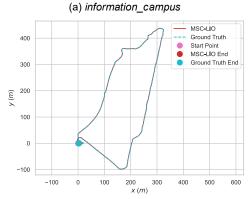
Fig. 11. The testing scenes of the real-time experiment.

(a) *information_campus*

(b) *academic_building*

Fig. 12. Estimated trajectories on the real-time experiments.

when running in real-time mode on the AGX Xavier. The total running times of MSC-LIO are less than one-fifth of the total

duration of the sequences. Besides, the average processing time per LiDAR frame is calculated by dividing the total running

TABLE VIII
REAL-TIME PERFORMANCE OF MSC-LIO

sequence	ARE / ATE (deg / m)	total running time (s)	average processing time per frame (s)
information_campus	0.50 / 1.47	393	0.020
academic_building	0.49 / 0.73	196	0.017

time by the total number of LiDAR frames (the LiDAR frame rate is 10 Hz in the real-time experiment). On the edge device AGX Xavier, the average processing times per LiDAR frame of the *information_campus* and *academic_building* are 0.02 s and 0.017 s, respectively, exhibiting superior efficiency. In summary, the proposed MSC-LIO demonstrates excellent real-time performance on the edge device NVIDIA Jetson AGX Xavier.

VI. CONCLUSION

This paper proposes a tightly coupled LIO within the MSCKF framework. An F2F same-plane-point tracking method is designed to improve data association efficiency, and the same-plane-point measurement model constructs a multi-state constraint. In addition, we propose a simpler and more direct point-velocity-based LiDAR-IMU time-delay estimation method based on the same-plane-point tracking. We conducted extensive experiments on both public datasets and real-world environments. The experimental results demonstrate that the proposed MSC-LIO outperforms SOTA systems in terms of accuracy and efficiency. Future work involves integrating the absolute positioning sources, such as the GNSS and the UWB, to achieve drift-free localization in large-scale outdoor and indoor environments.

REFERENCES

[1] G. Huang, "Visual-Inertial Navigation: A Concise Review," in *2019 International Conference on Robotics and Automation (ICRA)*, Montreal, QC, Canada: IEEE, May 2019, pp. 9572–9582.

[2] A. I. Mourikis and S. I. Roumeliotis, "A Multi-Constraint Kalman Filter for Vision-aided Inertial Navigation," in *Proceedings 2007 IEEE International Conference on Robotics and Automation*, Rome, Italy: IEEE, Apr. 2007, pp. 3565–3572.

[3] P. Geneva, K. Eckenhoff, W. Lee, Y. Yang, and G. Huang, "OpenVINS: A Research Platform for Visual-Inertial Estimation," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, Paris, France: IEEE, May 2020, pp. 4666–4672.

[4] K. Sun *et al.*, "Robust Stereo Visual Inertial Odometry for Fast Autonomous Flight," *IEEE Robot. Autom. Lett.*, vol. 3, no. 2, pp. 965–972, Apr. 2018.

[5] W. Liu, Z. Li, W. Li, P. Gardoni, H. Du, and M. A. Sotelo, "Time Delay Error Online Correction of LiDAR-IMU System Through MSCKF Integrated DLRNN Method," *IEEE/ASME Trans. Mechatron.*, vol. 29, no. 3, pp. 1878–1890, Jun. 2024.

[6] X. Zuo, P. Geneva, W. Lee, Y. Liu, and G. Huang, "LIC-Fusion: LiDAR-Inertial-Camera Odometry," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Macau, China: IEEE, Nov. 2019, pp. 5848–5854.

[7] X. Zuo *et al.*, "LIC-Fusion 2.0: LiDAR-Inertial-Camera Odometry with Sliding-Window Plane-Feature Tracking," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Las Vegas, NV, USA: IEEE, Oct. 2020, pp. 5112–5119.

[8] H. Shen, Q. Zong, B. Tian, X. Zhang, and H. Lu, "PGO-LIOM: Tightly Coupled LiDAR-Inertial Odometry and Mapping via Parallel and Gradient-Free Optimization," *IEEE Trans. Ind. Electron.*, vol. 70, no. 11, pp. 11453–11463, Nov. 2023.

[9] H. Ye, Y. Chen, and M. Liu, "Tightly Coupled 3D Lidar Inertial Odometry and Mapping," in *2019 International Conference on Robotics and Automation (ICRA)*, Montreal, QC, Canada: IEEE, May 2019, pp. 3144–3150.

[10] T. Shan, B. Englot, D. Meyers, W. Wang, C. Ratti, and D. Rus, "LIO-SAM: Tightly-coupled Lidar Inertial Odometry via Smoothing and Mapping," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Las Vegas, NV, USA: IEEE, Oct. 2020, pp. 5135–5142.

[11] X. Wang, X. Li, H. Yu, H. Chang, Y. Zhou, and S. Li, "GIVL-SLAM: A Robust and High-Precision SLAM System by Tightly Coupled GNSS RTK, Inertial, Vision, and LiDAR," *IEEE/ASME Trans. Mechatron.*, pp. 1–12, 2024.

[12] T. Fan, T. Zhang, H. Zhang, J. Mo, and X. Niu, "A double sideband combined tracking method for Galileo E5 AltBOC signals," *Satell. Navig.*, vol. 4, no. 1, p. 27, Dec. 2023.

[13] P. Geneva, K. Eckenhoff, Y. Yang, and G. Huang, "LIPS: LiDAR-Inertial 3D Plane SLAM," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Madrid: IEEE, Oct. 2018, pp. 123–130.

[14] D. Wisth, M. Camurri, S. Das, and M. Fallon, "Unified Multi-Modal Landmark Tracking for Tightly Coupled Lidar-Visual-Inertial Odometry," *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 1004–1011, Apr. 2021.

[15] H. Tang, T. Zhang, L. Wang, M. Yuan, and X. Niu, "BA-LINS: A Frame-to-Frame Bundle Adjustment for LiDAR-Inertial Navigation," 2024, *arXiv:2401.11491*: arXiv:2401.11491.

[16] 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 *2020 IEEE International Conference on Robotics and Automation (ICRA)*, Paris, France: IEEE, May 2020, pp. 8899–8906.

[17] 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.

[18] 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.

[19] 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.

[20] 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, Jan. 2024.

[21] H. Tang, T. Zhang, X. Niu, L. Wang, L. Wei, and J. Liu, "FF-LINS: A Consistent Frame-to-Frame Solid-State-LiDAR-Inertial State Estimator," *IEEE Robot. Autom. Lett.*, vol. 8, no. 12, pp. 8525–8532, Dec. 2023.

[22] Y. Wu and J. Zhao, "A Robust and Precise LiDAR-Inertial-GPS Odometry and Mapping Method for Large-Scale Environment," *IEEE/ASME Trans. Mechatron.*, vol. 27, no. 6, pp. 5027–5036, Dec. 2022.

[23] P. Shi, Z. Zhu, S. Sun, X. Zhao, and M. Tan, "Invariant Extended Kalman Filtering for Tightly Coupled LiDAR-Inertial Odometry and Mapping," *IEEE/ASME Trans. Mechatron.*, vol. 28, no. 4, pp. 2213–2224, Aug. 2023.

[24] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*, 2. edition, 17. printing. Cambridge: Cambridge Univ. Press, 2018.

[25] Z. Liu and F. Zhang, "BALM: Bundle Adjustment for Lidar Mapping," *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 3184–3191, Apr. 2021.

[26] C. Hertzberg, R. Wagner, U. Frese, and L. Schröder, "Integrating Generic Sensor Fusion Algorithms with Sound State Representations through Encapsulation of Manifolds," Jul. 06, 2011, *arXiv:1107.1119*.

[27] J. Solà, "Quaternion kinematics for the error-state Kalman filter," Nov. 03, 2017, *arXiv:1711.02508*: arXiv:1711.02508.

[28] T. D. Barfoot, *State Estimation for Robotics*, 1st ed. Cambridge: Cambridge Univ. Press, 2017.

[29] H. Tang, X. Niu, T. Zhang, L. Wang, and J. Liu, "LE-VINS: A Robust Solid-State-LiDAR-Enhanced Visual-Inertial Navigation System for Low-Speed Robots," *IEEE Trans. Instrum. Meas.*, vol. 72, pp. 1–13, 2023.

[30] Paul Groves, *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems*. Boston: Artech House, 2008.

[31] X. Zuo *et al.*, "Technique Report of LIC-Fusion 2.0: LiDAR-Inertial-Camera Odometry with Sliding-Window Plane-Feature Tracking," Dept. of Comp. Sci, Ethz, Tech. Rep 1, 2020.

[32] T.-M. Nguyen *et al.*, "MCD: Diverse Large-Scale Multi-Campus Dataset for Robot Perception," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

[33] J. Li *et al.*, "WHU-Helmet: A Helmet-Based Multisensor SLAM Dataset for the Evaluation of Real-Time 3-D Mapping in Large-Scale GNSS-Denied Environments," *IEEE Trans. Geosci. Remote Sensing*, vol. 61, pp. 1–16, 2023.

[34] T. Zhang, M. Yuan, L. Wang, H. Tang, and X. Niu, "A Robust and Efficient IMU Array/GNSS Data Fusion Algorithm," *IEEE Sensors J.*, 2024.

[35] Michael Grupp, "evo." [Online]. Available: https://github.com/MichaelGrupp/evo

Xiaoji Niu received the bachelor's and Ph.D. degrees from the Department of Precision Instruments, Tsinghua University, Beijing, China, in 1997 and 2002, respectively.

He was a Postdoctoral Researcher with the University of Calgary, Calgary, AB, Canada, and worked as a Senior Scientist with SiRF Technology Inc., Shanghai, China. He is currently a Professor with the GNSS Research Center, Wuhan University, Wuhan, China. He has authored or coauthored more than 90 academic articles and owns 28 patents. He leads a multi-sensor navigation group focusing on GNSS/INS integration, low-cost navigation sensor fusion, and its new applications.

Tisheng Zhang received the B.Sc. and Ph.D. degrees in communication and information systems from Wuhan University, Wuhan, China, in 2008 and 2013, respectively.

From 2018 to 2019, he was a Postdoctoral Researcher with The Hong Kong Polytechnic University, Hong Kong. He is an Associate Professor with the GNSS Researches Center, Wuhan University. His research interests include GNSS receiver and multi-sensor deep integration.

Man Yuan received the B.E. (Hons.) degree in communication engineering from Wuhan University, Wuhan, China, in 2023, where he is currently pursuing the master's degree in navigation, guidance, and control with the GNSS Research Center.

His primary research interests include GNSS/INS integrations and LiDAR-based navigation.

Linfu Wei received the B.E. and M.E. (Hons.) degrees from Wuhan University, Wuhan, China, in 2021 and 2024, respectively.

His primary research interests include LiDAR-inertial odometry, UWB-based navigation, and multi-sensor fusion.

Hailiang Tang received the M.E. and Ph.D. degrees from Wuhan University, Wuhan, China, in 2020 and 2023, respectively.

He is currently a Postdoctoral Fellow with the GNSS Research Center, Wuhan University. His current research interests include visual and LiDAR SLAM, autonomous robotics systems, GNSS/INS integration technology, and deep learning.