LIC-Fusion 2.0: LiDAR-Inertial-Camera Odometry with Sliding-Window Plane-Feature Tracking

Xingxing Zuo[1,2], Yulin Yang[3], Patrick Geneva[3], Jiajun Lv[2], Yong Liu[2], Guoquan Huang[3], Marc Pollefeys[1,4]

Abstract*— Multi-sensor fusion of multi-modal measurements from commodity inertial, visual and LiDAR sensors to provide robust and accurate 6DOF pose estimation holds great potential in robotics and beyond. In this paper, building upon our prior work (i.e., LIC-Fusion), we develop a sliding-window filter based LiDAR-Inertial-Camera odometry with online spatiotemporal calibration (i.e., LIC-Fusion 2.0), which introduces a novel sliding-window plane-feature tracking for efficiently processing 3D LiDAR point clouds. In particular, after motion compensation for LiDAR points by leveraging IMU data, low-curvature planar points are extracted and tracked across the sliding window. A novel outlier rejection criteria is proposed in the plane-feature tracking for high quality data association. Only the tracked planar points belonging to the same plane will be used for plane initialization, which makes the plane extraction efficient and robust. Moreover, we perform the observability analysis for the LiDAR-IMU subsystem and report the degenerate cases for spatiotemporal calibration using plane features. While the estimation consistency and identified degenerate motions are validated in Monte-Carlo simulations, different real-world experiments are also conducted to show that the proposed LIC-Fusion 2.0 outperforms its predecessor and other state-of-the-art methods.

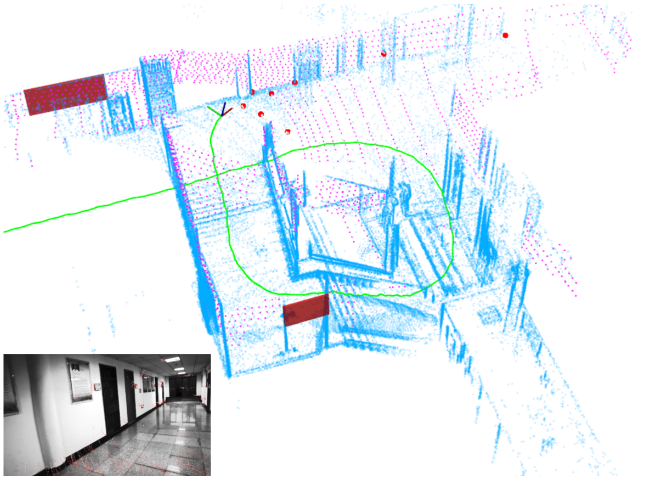

Fig. 1: The proposed LIC-Fusion 2.0 with sliding-window plane-feature tracking. The stably tracked SLAM plane landmarks from the LiDAR and SLAM point landmarks from the camera are colored in red. High curvature LiDAR points in blue, accumulated from a series of LiDAR scans, are shown to visualize the surroundings only. Magenta points are extracted planar points from the latest LiDAR scan. The estimated trajectory is marked in green.

I. Introduction and Related Work

Accurate and robust 3D localization is essential for autonomous robots to perform high-level tasks such as autonomous driving, inspection, and delivery. LiDAR, camera, and Inertial Measurements Unit (IMU) are among the most popular sensor choices for 3D pose estimation [1–5]. Since each sensor modality has its virtues and inherent shortcomings, a proper multi-sensor fusion algorithm aiming at leveraging the “best” of each sensor modality is expected to have a substantial performance gain in both estimation accuracy and robustness. For this reason, Zhang and Singh [1] proposed a graph optimization based laser-visual-inertial localization and mapping method following a multilayer processing pipeline, in which the IMU data for prediction, a visual-inertial coupled estimator for motion estimation, and LiDAR based scan matching is integrated to further improve the motion estimation and reconstruct the map. In contrast to [1], our prior LIC-Fusion [5] follows a lightweight filtering pipeline, which also enables spatial and temporal calibrations between the un-synchronized sensors. In [2], a depth association algorithm for visual features from LiDAR measurements is developed, which is particularly suitable for autonomous driving scenarios. Shao et al. [4] fused stereo visual-inertial odometry and LiDAR scan matching within a graph optimization framework, in which, after detecting loop closures from images, iterative closet point (ICP) of LiDAR data is performed to find the loop closure constraints.

Substantial research efforts have been devoted on processing 3D LiDAR measurements to find the relative pose between two LiDAR scans. To achieve this, ICP [6] is among the most widely used algorithms to compute the relative motion from two point clouds. However, traditional ICP can easily get poor results when applied on registering two 3D LiDAR scans, which have vertical sparsity and ring structure. To cope with the sparsity in LiDAR scans, in [7], raw LiDAR points are converted into line segments, and the closest points from two line segments are minimized iteratively. Similarly, in the well-known LOAM algorithm [8], the registration of LiDAR scans leverage the implicit geometrical constraints (point-to-plane and point-to-line distance) to perform “feature” based ICP. This algorithm is more robust and efficient since only a few selected points with high/low curvatures are processed. However, both ICP and LOAM provide con-

This work was partially supported by the Department of Computer Science at ETHz and the National Natural Science Foundation of China under Grant 61836015. Zuo was partially supported by the Chinese Scholarship Committee, Yang was partially supported by the University of Delaware Doctoral Fellowship, and Geneva by the Delaware Space Grant College and Fellowship Program (NASA Grant NNX15AI19H). (Yong Liu is the corresponding author, email: yongliu@iipc.zju.edu.cn)

[1] Department of Computer Science, ETH Zürich, Switzerland.

[2] Institute of Cyber-System and Control, Zhejiang University, Hangzhou, China.

[3] Robot Perception and Navigation Group, University of Delaware, Newark, DE 19716, USA.

[4] Microsoft Mixed Reality and Artificial Intelligence Lab, Zürich, Switzerland.

straints *only* between two consecutive scans, and it is hard to accurately model the relative pose uncertainty. An alternative approach is to directly extract features (e.g., planes) and construct a feature-based SLAM problem [9]. However, not only is large-scale plane extraction often computationally intensive, but also plane-feature data association (e.g., based on Mahalanobis distance test) needs ad-hoc parameter tuning in cluttered environments.

To address these issues, building upon our prior work of LIC-fusion 1.0, we propose a novel plane-feature tracking algorithm to efficiently process the LiDAR measurements and then optimally integrated it into a sliding-window filter-based multi-sensor fusion framework (see the overview of the system in Fig. 1). In particular, after removing the motion distortion for LiDAR points, during the current sliding window, we only extract and track planar points associated with certain planes. Only tracked planar points will be used for plane feature initialization, which makes the plane extraction more efficient and robust. While abundant of works exist on observability analysis of visual-inertial systems with point features [10, 11], we perform observability analysis for the proposed lidar-inertial-visual system and identify degenerate cases for online calibration with plane features. The main contributions of this work can be summarized as follows:

- We develop a novel sliding-window plane-feature tracking algorithm that allows for tracking 3D environmental plane features across multiple LiDAR scans within a sliding-window. This tracking algorithm is optimally integrated into our prior tightly-coupled fusion framework: LIC-Fusion [5]. For the proposed plane tracking, a novel outlier rejection criterion is advocated, which allows for robust matching by taking into account the transformation uncertainty between LiDAR frames. The system can model the uncertainties of LiDAR measurements reasonably, which eliminates the inconsistent-prone ICP for LiDAR scan matching.
- We perform an in-depth observability analysis of the LiDAR-inertial-camera system with plane features and identify the degenerate cases that cause the system to have additional unobservable directions.
- We conduct extensive experiments of the proposed LIC-Fusion 2.0 on a series of Monte-Carlo simulations and real-world datasets, which verifies both the consistency and accuracy of the proposed system.

II. LIC-Fusion 2.0 Problem Formulation

A. State Vector

In addition to LIC-Fusion's [5] original state containing IMU state $\mathbf{x}_I$, camera clones $\mathbf{x}_C$, LiDAR clones $\mathbf{x}_L$, and spatial-temporal calibration of IMU-CAM $\mathbf{x}_{calib\_C}$ and LiDAR-IMU $\mathbf{x}_{calib\_L}$, we store environmental visual ${}^G\mathbf{x}_f$ and LiDAR landmarks ${}^A\mathbf{x}_\pi$. These features are "long lived" and through frequent matching can limit estimation drift. The state vector is:

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_I^\top \ \mathbf{x}_{calib\_C}^\top \ \mathbf{x}_{calib\_L}^\top \ \mathbf{x}_C^\top \ \mathbf{x}_L^\top \ {}^G\mathbf{x}_f^\top \ {}^A\mathbf{x}_\pi^\top \end{bmatrix}^\top \tag{1}$$

where

$$\mathbf{x}_I = \begin{bmatrix} {}^{I_k}_G\bar{q}^\top \ \mathbf{b}_g^\top \ {}^G\mathbf{v}_{I_k}^\top \ \mathbf{b}_a^\top \ {}^G\mathbf{p}_{I_k}^\top \end{bmatrix}^\top \tag{2}$$

$$\mathbf{x}_{calib\_C} = \begin{bmatrix} {}^C_I\bar{q}^\top \ {}^C\mathbf{p}_I^\top \ t_{dC} \end{bmatrix}^\top \tag{3}$$

$$\mathbf{x}_{calib\_L} = \begin{bmatrix} {}^L_I\bar{q}^\top \ {}^L\mathbf{p}_I^\top \ t_{dL} \end{bmatrix}^\top \tag{4}$$

$$\mathbf{x}_C = \begin{bmatrix} {}^{I_{c_0}}_G\bar{q}^\top \ {}^G\mathbf{p}_{I_{c_0}}^\top \cdots {}^{I_{c_{m-1}}}_G\bar{q}^\top \ {}^G\mathbf{p}_{I_{c_{m-1}}}^\top \end{bmatrix}^\top \tag{5}$$

$$\mathbf{x}_L = \begin{bmatrix} {}^{I_{l_0}}_G\bar{q}^\top \ {}^G\mathbf{p}_{I_{l_0}}^\top \cdots {}^{I_{l_{n-1}}}_G\bar{q}^\top \ {}^G\mathbf{p}_{I_{l_{n-1}}}^\top \end{bmatrix}^\top \tag{6}$$

$${}^G\mathbf{x}_f = \begin{bmatrix} {}^G\mathbf{p}_{f_0}^\top \ {}^G\mathbf{p}_{f_1}^\top \cdots {}^G\mathbf{p}_{f_{g-1}}^\top \end{bmatrix}^\top \tag{7}$$

$${}^A\mathbf{x}_\pi = \begin{bmatrix} {}^A\mathbf{p}_{\pi_0}^\top \ {}^A\mathbf{p}_{\pi_1}^\top \cdots {}^A\mathbf{p}_{\pi_{h-1}}^\top \end{bmatrix}^\top \tag{8}$$

In the above, $\{I_k\}$ is the local IMU frame at time instant t_k. ${}^{I_k}_G\bar{q}$ is a unit quaternion in JPL format [12], which represents 3D rotation ${}^{I_k}_G\mathbf{R}$ from $\{G\}$ to $\{I_k\}$. ${}^G\mathbf{v}_{I_k}$, ${}^G\mathbf{p}_{I_k}$ denote the velocity and position of IMU in $\{G\}$. Moreover, $\mathbf{b}_g$ and $\mathbf{b}_a$ are the gyro and accelerator biases that corrupt the IMU measurements respectively. The system error state for x is defined as $\tilde{x} = x - \hat{x}$ where $\hat{x}$ is the current estimate[1]. For details on the calibration parameters please see the original LIC-Fusion paper [5]. Additionally, we include environmental visual features, ${}^G\mathbf{p}_f$, represented in the global frame of reference, and store environmental plane features represented in an anchored frame $\{A\}$. The plane is represented by the closest point [9, 13], and the anchored representation can avoid the singularity when the norm of ${}^G\mathbf{p}_\pi$ approaches zero. These long-lived planar features will be tracked in incoming LiDAR scans using the proposed tracking algorithm until they are lost.

B. Point-to-Plane Measurement Model

Considering a LiDAR planar point measurement, ${}^L\mathbf{p}_f$, that is sampled on the plane ${}^A\mathbf{p}_\pi$. We can define the point-to-plane distance measurement model:

$$\mathbf{z}_\pi = \frac{{}^L\mathbf{p}_\pi^\top}{\left\|{}^L\mathbf{p}_\pi\right\|}({}^L\mathbf{p}_f - \mathbf{n}_f) - \left\|{}^L\mathbf{p}_\pi\right\| \tag{9}$$

where $\mathbf{n}_f \sim \mathcal{N}(\mathbf{0}, \sigma_f^2\mathbf{I}_3)$. With a slight abuse of notation, by defining ${}^Ld = \left\|{}^L\mathbf{p}_\pi\right\|$ and ${}^L\mathbf{n} = {}^L\mathbf{p}_\pi / \left\|{}^L\mathbf{p}_\pi\right\|$, a plane ${}^A\mathbf{p}_\pi$ can be transformed into the local frame by:

$$\begin{bmatrix} {}^L\mathbf{n} \\ {}^Ld \end{bmatrix} = \begin{bmatrix} {}^L_A\mathbf{R} & 0 \\ -{}^A\mathbf{p}_L^\top & 1 \end{bmatrix} \begin{bmatrix} {}^A\mathbf{n} \\ {}^Ad \end{bmatrix} \tag{10}$$

C. LiDAR Plane Feature Update

Analogous to point features [14], we divide all the tracked plane features from the LiDAR point clouds into "MSCKF" and "SLAM" based on the track length. Note that the sliding-window-based plane tracking will be explained in detail in Section III-B. Given a series of LiDAR point measurements collected over the whole sliding window of the plane feature

[1] $\tilde{x}$ holds for velocity, position, bias, except for the quaternion, which follows: $\tilde{\bar{q}} \simeq \left[\frac{1}{2}\delta\boldsymbol{\theta}^\top \ 1\right]^\top \otimes \hat{\bar{q}}$, where $\otimes$ denotes quaternion multiplication [12], and $\delta\boldsymbol{\theta}$ is the corresponding error state.

${}^A\mathbf{p}_{\pi_j}$, we can linearize the measurement $\mathbf{z}_f^{(j)}$ in Eq. (9) at current estimates of ${}^A\mathbf{p}_{\pi_j}$ and the states $\mathbf{x}$ as:

$$\mathbf{r}_f^{(j)} = \mathbf{0} - \mathbf{z}_f^{(j)} \simeq \mathbf{H}_x^{(j)}\tilde{\mathbf{x}} + \mathbf{H}_\pi^{(j)A}\tilde{\mathbf{p}}_{\pi_j} + \mathbf{H}_n^{(j)}\mathbf{n}_f^{(j)} \tag{11}$$

where $\mathbf{n}_f^{(j)}$ denotes the stacked noise vector. $\mathbf{H}_x^{(j)}$, $\mathbf{H}_\pi^{(j)}$ and $\mathbf{H}_n^{(j)}$ are the stacked Jacobians with respect to pose states, the plane landmark and the measurement noise, respectively. Analytical forms of $\mathbf{H}_x^{(j)}, \mathbf{H}_\pi^{(j)}, \mathbf{H}_n^{(j)}$ can be found out in our companion technical report [15].

If ${}^A\mathbf{p}_{\pi_j}$ is a MSCKF plane landmark, the nullspace operation [16] is performed to remove the dependency on ${}^A\mathbf{p}_{\pi_j}$ by projection onto the left nullspace $\mathbf{N}$:

$$\mathbf{N}^\top\mathbf{r}_f^{(j)} = \mathbf{N}^\top\mathbf{H}_x^{(j)}\tilde{\mathbf{x}} + \mathbf{N}^\top\mathbf{H}_\pi^{(j)A}\tilde{\mathbf{p}}_{\pi_j} + \mathbf{N}^\top\mathbf{H}_n^{(j)}\mathbf{n}_f^{(j)} \tag{12}$$

$$\Rightarrow\ \mathbf{r}_{fo}^{(j)} = \mathbf{H}_{xo}^{(j)}\tilde{\mathbf{x}} + \mathbf{n}_o^{(j)} \tag{13}$$

Due to the special structure that $\mathbf{H}_n^{(j)}\mathbf{H}_n^{(j)\top} = \mathbf{I}_n$ the measurement covariance is still isotropic and thus the nullspace operation is valid (i.e. $\sigma_f^2\mathbf{N}^\top\mathbf{H}_n^{(j)}\mathbf{H}_n^{(j)\top}\mathbf{N} = \sigma_f^2\mathbf{I}_n$). By stacking the residuals and Jacobians of all MSCKF plane landmarks, we obtain:

$$\mathbf{r}_{fo} = \mathbf{H}_{xo}\tilde{\mathbf{x}} + \mathbf{n}_o \tag{14}$$

This stacked system can then update the state and covariance using the standard EKF update equations.

If ${}^A\mathbf{p}_{\pi_j}$ is a SLAM plane landmark that already exists in the state, we can directly update its estimate and the state using Eq. (11). To determine whether a plane feature with a long track length should be initialized into the state as a SLAM feature, we note that planes constrain the current state estimate based on their normals. In the case that three planes that are not parallel to each other are observed, then the current state estimate can be well constrained [17]. Thus, we opt to insert "informative" planes whose normal directions are significantly different from the planes currently being estimated (in our implementation, we only insert planes whose normal directions have greater than ten degrees difference). After augmenting a plane feature into the state vector, future LiDAR scans can also match to it.

III. Sliding-Window LiDAR Plane Tracking

A. Motion Compensation for Raw LiDAR Points

Since the raw LiDAR points are deteriorated by motion distortion, we can remove the distortion by utilizing the high-frequency IMU pose estimation. When propagating IMU state, we save the propagated IMU poses at each timestep into a buffer, which can then be used to remove the distortion. Since LiDAR points occur at a higher frequency than IMU, we perform linear interpolate between each of these buffered poses to the corresponding time of each LiDAR ray. For orientation, we perform $SO(3)$ interpolation similar to [18], while linear interpolate between the two positions. Using this pose, we transform all 3D points into the pose at the sweep start time, eliminating the motion distortion.

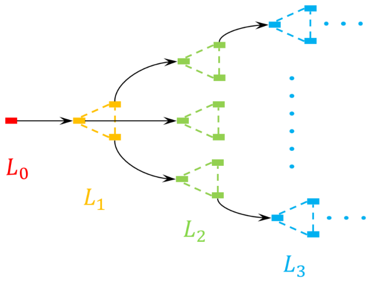

Fig. 2: A plane landmark tracked across multiple LiDAR frames $\{L_0, L_1, L_2, L_3, \cdots\}$ within a sliding window. A planar point in the last LiDAR scan is associated with a triangle consisting of three planar points in its subsequent LiDAR scan. All the tracked planar points are assumed to be sampled from the same plane landmark.

B. Planar Landmark Tracking

We now explain how we perform temporal planar feature tracking across sequential undistorted LiDAR scans. We first extract planar points from each LiDAR scan using the method proposed in [8], where low-curvature points are classified as being on some environmental planar surface. A planar point indexed by i in LiDAR frame $\{L_a\}$ will be tracked in the latest LiDAR frame $\{L_b\}$ by finding its nearest neighbour point j after projection into $\{L_b\}$. We then find another two points (indexed by k, l), which are the nearest points to j on the same scan ring and the adjacent scan rings, respectively. These three points (j, k, l) are guaranteed to be non-collinear and form a planar patch corresponding to planar point i. If the distance between the projected i and j or distances between any two points $\in \{j, k, l\}$ are larger than a given threshold, we will reject to associate i to (j, k, l), and thus lose track of this planar LiDAR feature. An overview of this approach is shown in Algorithm 1 and an additional outlier rejection scheme is presented in the following section. To prevent the reuse of information, we employ a simple strategy that a planar point can only be matched to a single common plane feature.

Algorithm 1 LiDAR Plane Tracking Procedure

```
Extract planar points from $\{L_b\}$
Project prior planar points from $\{L_a\}$ into $\{L_b\}$, find the nearest corresponding point to each in $\{L_b\}$.
for all $(\mathbf{p}_i, \mathbf{p}_j) \in$ projected plane points do
    Find two closest points $\mathbf{p}_k$, $\mathbf{p}_l$ in $\{L_b\}$
    Ensure $\mathbf{p}_k$ scan ring is the same
    Ensure $\mathbf{p}_l$ scan rings is the adjacent
    Ensure that selected points are not already used
    if $|\mathbf{p}_n - \mathbf{p}_m| < d \quad \forall(n, m) \in (i, j, k, l)$ then
        Compute plane normal ${}^b\mathbf{n}_2$ transformed into $\{L_a\}$
        Compute measurement covariance matrix $\mathbf{P}_{\pi n}$
        if $\chi^2(\mathbf{z}_n, \mathbf{H}, \mathbf{P}_{\pi n}) == Pass$ then
            $\mathbf{p}_j, \mathbf{p}_k, \mathbf{p}_l$ are measurements of $\mathbf{p}_i$'s plane
            $\mathbf{p}_j, \mathbf{p}_k, \mathbf{p}_l$ will be tracked into the next scan
        end if
    end if
end for
```

C. Normal-based Plane Data Association

We now discuss our novel plane normal-based data association method, which rejects invalid plane associations based on the calculated plane normal. Consider the case that we have extracted a plane on the floor next to a vertical wall. If the tracking algorithm discussed in the previous section is used, then points that are *near* the bottom of the wall would be classified as being on the same plane as floor points due to purely relying on 3D distance. This can have huge implications on the estimation accuracy due to incorrectly saying that the wall and floor are the same plane even though their normal directions should be perpendicular to each other.

To handle this, we propose leveraging the current state uncertainty and the uncertainty of the planar points to perform a Mahalanobis distance test between the normal vectors of the candidate match. Specifically, we have a possible planar match of the points, $({}^{L_a}\mathbf{p}_{fm}, {}^{L_a}\mathbf{p}_{fn}, {}^{L_a}\mathbf{p}_{fo})$ in frame $\{L_a\}$, and $({}^{L_b}\mathbf{p}_{fg}, {}^{L_b}\mathbf{p}_{fh}, {}^{L_b}\mathbf{p}_{fi})$ in frame $\{L_b\}$. We define a synthetic measurement $\mathbf{z}_n$ reflecting the "parallelarity" between the two normal vectors of each of these planes as:

$$\mathbf{z}_n = \lfloor {}^{L_a}\mathbf{n}_1 \rfloor {}^{L_a}_{L_b}\mathbf{R}\, {}^{L_b}\mathbf{n}_2 \tag{15}$$

$${}^{L_a}\mathbf{n}_1 = \lfloor {}^{L_a}\mathbf{p}_{fn} - {}^{L_a}\mathbf{p}_{fm} \rfloor ({}^{L_a}\mathbf{p}_{fo} - {}^{L_a}\mathbf{p}_{fm}) \tag{16}$$

$${}^{L_b}\mathbf{n}_2 = \lfloor {}^{L_b}\mathbf{p}_{fh} - {}^{L_b}\mathbf{p}_{fg} \rfloor ({}^{L_b}\mathbf{p}_{fi} - {}^{L_n}\mathbf{p}_{g}) \tag{17}$$

We can define two simplified stacked "states" as:

$$\mathbf{p}_{n1} = \begin{bmatrix} {}^{L_a}\mathbf{p}_{fm}^\top & {}^{L_a}\mathbf{p}_{fn}^\top & {}^{L_a}\mathbf{p}_{fo}^\top \end{bmatrix}^\top \tag{18}$$

$$\mathbf{p}_{n2} = \begin{bmatrix} {}^{L_b}\mathbf{p}_{fg}^\top & {}^{L_b}\mathbf{p}_{fh}^\top & {}^{L_b}\mathbf{p}_{fi}^\top \end{bmatrix}^\top \tag{19}$$

The corresponding covariances of $\mathbf{p}_{n1}$ and $\mathbf{p}_{n2}$ can be computed from LiDAR points noises and denoted as $\mathbf{P}_{n1} = \mathbf{P}_{n2} = \sigma_f^2 \mathbf{I}_{\mathbf{p}_{n1}}$. The Mahalanobis distance d_z of $\mathbf{z}_n$ can be computed as:

$$d_z = \mathbf{z}_n^\top \mathbf{P}_{\pi n}^{-1} \mathbf{z}_n \tag{20a}$$

$$\begin{aligned}\mathbf{P}_{\pi n} = & \frac{\partial \tilde{\mathbf{z}}_n}{\partial \tilde{\mathbf{p}}_{n1}} \mathbf{P}_{n1} \left(\frac{\partial \tilde{\mathbf{z}}_n}{\partial \tilde{\mathbf{p}}_{n1}} \right)^\top + \frac{\partial \tilde{\mathbf{z}}_n}{\partial \tilde{\mathbf{p}}_{n2}} \mathbf{P}_{n2} \left(\frac{\partial \tilde{\mathbf{z}}_n}{\partial \tilde{\mathbf{p}}_{n2}} \right)^\top \\ & + \frac{\partial \tilde{\mathbf{z}}_n}{\partial {}^{L_a}_{L_b}\delta\boldsymbol{\theta}} \mathbf{P}_{ori} \left(\frac{\partial \tilde{\mathbf{z}}_n}{\partial {}^{L_a}_{L_b}\delta\boldsymbol{\theta}} \right)^\top \end{aligned} \tag{20b}$$

where $\mathbf{P}_{ori}$ is the known covariance of relative rotation ${}^{L_a}_{L_b}\mathbf{R}$ based on the current EKF covariance and the Jacobians:

$$\frac{\partial \tilde{\mathbf{z}}_n}{\partial \tilde{\mathbf{p}}_{n1}} = -\lfloor {}^{L_a}_{L_b}\mathbf{R}\, {}^{L_b}\mathbf{n}_2 \rfloor \frac{\partial {}^{L_a}\tilde{\mathbf{n}}_1}{\partial \tilde{\mathbf{p}}_{n1}} \tag{21a}$$

$$\frac{\partial \tilde{\mathbf{z}}_n}{\partial \tilde{\mathbf{p}}_{n2}} = \lfloor {}^{L_a}\mathbf{n}_1 \rfloor {}^{L_a}_{L_b}\mathbf{R} \frac{\partial {}^{L_b}\tilde{\mathbf{n}}_2}{\partial \tilde{\mathbf{p}}_{n2}} \tag{21b}$$

Based on the Mahalanobis distance test, we can reject incorrect temporal planar tracks. Note that this check can only be performed once we have more than two sequential LiDAR frames, see Fig. 2 for illustrating the measurements on the same plane while across multiple LiDAR frames.

D. Planar Landmark Initialization

If a plane landmark ${}^{L_a}\mathbf{p}_{\pi_j}$ can be tracked across several LiDAR frames, we will initialize this plane landmark in the oldest LiDAR frame $\{L_a\}$ with all its valid planar point observations, denoted as set $\mathcal{P}_{fj}$, within the sliding window. A planar point observation ${}^{L_x}\mathbf{p}_{fm_i}^{(j)} = {}^{L_x}\mathbf{p}_{f_i}^{(j)} + \mathbf{n}_{f_i}^{(j)}$ is the i_{th} measurement in $\mathcal{P}_{fj}$, with $\mathbf{n}_{f_i}^{(j)}$ is the measurement noise. We compute the distance between ${}^{L_x}\mathbf{p}_{f_i}^{(j)}$ and ${}^{L_a}\mathbf{p}_{\pi_j}$ as:

$$\mathbf{z}_{f_i}^{(j)} = \frac{{}^{L_a}\mathbf{p}_{\pi_j}^\top}{\left\| {}^{L_a}\mathbf{p}_{\pi_j} \right\|} \left({}^{L_a}_{L_x}\mathbf{R} \left({}^{L_x}\mathbf{p}_{fm_i}^{(j)} - \mathbf{n}_{f_i}^{(j)} \right) + {}^{L_a}\mathbf{p}_{L_x} \right) - \left\| {}^{L_a}\mathbf{p}_{\pi_j} \right\| \tag{22}$$

By stacking Eq. (22) and constructing a linear system, we can compute the initial guess for plane normal vector ${}^{L_a}\hat{\mathbf{p}}_{\pi_j}/||{}^{L_a}\hat{\mathbf{p}}_{\pi_j}||$ and plane distance scalar $||{}^{L_a}\hat{\mathbf{p}}_{\pi_j}||$. The initial guess of the plane landmark can be further refined by minimizing following cost function:

$${}^{L_a}\mathbf{p}_{\pi_j}^* = \underset{{}^{L_a}\mathbf{p}_{\pi_j}}{\arg\min} \sum_{i=1}^{n} \left\| \tilde{\mathbf{z}}_{f_i}^{(j)} \right\|_{\frac{\mathbf{I}}{\sigma_f^2}}^2 \tag{23}$$

where n is the amount of observations in $\mathcal{P}_{fj}$. The entire proposed LIC-Fusion 2.0 LiDAR processing pipeline can be seen in Algorithm 2.

IV. Observability Analysis

The observability analysis of IMU-CAM navigation system with online calibration has been studied extensively in literature [10, 11, 19], however, the analysis for LiDAR-IMU navigation with online calibration using plane features is still missing. In addition, since the calibration between IMU-CAM and LiDAR-IMU calibration are relatively independent, previously identified degenerate motions for VINS calibration cannot be directly applied to LiDAR-IMU cases with plane features. Hence, in this paper, we focus on the subsystem of LIC-Fusion 2.0 with LiDAR-IMU only and study specifically the degenerate cases for online spatial-temporal LiDAR-IMU calibration using plane features. In particular, the observability matrix $\mathbf{M}(\mathbf{x})$ is given by:

$$\mathbf{M}(\mathbf{x}) = \begin{bmatrix} \left(\mathbf{H}_{\mathbf{x},1} \boldsymbol{\Phi}_{(1,1)} \right)^\top & \dots & \left(\mathbf{H}_{\mathbf{x},k} \boldsymbol{\Phi}_{(k,1)} \right)^\top \end{bmatrix}^\top \tag{24}$$

where $\mathbf{H}_{\mathbf{x},k}$ represents the measurement Jacobians at timestep k. The right null space of $\mathbf{M}(\mathbf{x})$, denoted by $\mathbf{N}$, indicates the unobservable directions of the underlying system.

A. State Vector and State Transition Matrix

As in our previous work [11], we have already studied the observability for IMU-CAM subsystem with online calibration and point features, this analysis will only focus on LiDAR-IMU system with online calibration and plane features. Hence, with closest point representation for plane feature, the state vector with a plane feature and LiDAR-IMU calibration can be written as:

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_I^\top & \mathbf{x}_{calib\_L}^\top & {}^{G}\mathbf{p}_\pi^\top \end{bmatrix}^\top \tag{25}$$

The state transition matrix can be written as:

$$\boldsymbol{\Phi}_{(k,1)} = \begin{bmatrix} \boldsymbol{\Phi}_I & \mathbf{0}_{15\times7} & \mathbf{0}_{15\times3} \\ \mathbf{0}_{7\times15} & \boldsymbol{\Phi}_{calib\_L} & \mathbf{0}_{7\times3} \\ \mathbf{0}_{3\times15} & \mathbf{0}_{3\times7} & \boldsymbol{\Phi}_\pi \end{bmatrix} \tag{26}$$

Where $\boldsymbol{\Phi}_I$ denotes the IMU state transition matrix [10]. $\boldsymbol{\Phi}_{calib\_L} = \mathbf{I}_7$ and $\boldsymbol{\Phi}_\pi = \mathbf{I}_3$. Note that without loss of generality, we represent the plane feature in the global frame $\{G\}$. We only consider one plane in our state vector, for the more planes cases please refer to our technical report [15].

Algorithm 2 LIC-Fusion 2.0 LiDAR Processing Pipeline

```
Propagation:
  • Propagate the state forward in time by IMU measurements
  • Buffer propagated poses for LiDAR scan motion compensation
Update: Given an incoming LiDAR Scan,
  • Clone the corresponding IMU pose.
  • Remove motion distortion for the scan as Sec. III-A
  • Extract and track planar points as Sec. III-B.
  • For SLAM plane landmarks, use the tracked planar points to compute the residuals & measurement Jacobians, and perform EKF update [Eq. (11)].
  • For planar points that tracked across the sliding window or lost track in the current scan:
      – Query its associated observations over the sliding window.
      – Check the association validity by Mahalanobis gating test as Sec. III-C.
      – Construct the residual vectors and the Jacobians in Eq. (22) with all the verified observations.
      – Determine whether the plane landmark should be a SLAM landmark by checking the track length and the normal vector “parallelity” to the existing SLAM plane landmarks.
      – If it should be a SLAM plane, add it to the state vector and augment the state covariance matrix. Otherwise, treat it as a MSCKF feature.
  • Stack the residuals and Jacobians of all MSCKF plane landmarks, and perform EKF update [Eq. (14)]
Management of States:
  • SLAM plane landmarks that have lost track are marginalized out.
  • SLAM plane landmarks anchored in the frame that needs to be marginalized are moved to the newest frame.
  • Marginalize the cloned pose corresponding to the oldest LiDAR frame in the sliding window state.
```

B. Measurement Jacobians and Observability Matrix

Following the observability methodology in [10], we construct the k-th block of the observability matrix as:

$$\mathbf{M}_k = \mathbf{H}_\pi \begin{bmatrix} {}^L_I\mathbf{R}{}^I_G\hat{\mathbf{R}} & \mathbf{0}_{3\times1} \\ \mathbf{0}_{1\times3} & 1 \end{bmatrix} \times$$
$$\begin{bmatrix} \Gamma_{\pi11} & \mathbf{0}_3 & \mathbf{0}_3 & \Gamma_{\pi14} & \mathbf{0}_3 & \Gamma_{\pi16} & \mathbf{0}_3 & \Gamma_{\pi18} & \Gamma_{\pi19} \\ \Gamma_{\pi21} & {}^G\mathbf{n}^\top & {}^G\mathbf{n}^\top\Delta t_k & \Gamma_{\pi24} & \Gamma_{\pi25} & \Gamma_{\pi26} & \Gamma_{\pi27} & \Gamma_{\pi28} & \Gamma_{\pi29} \end{bmatrix}$$

where $\Gamma_{\pi ij}, i \in \{1,2\}, j \in \{1\ldots9\}$ can be found in [15].

For LiDAR aided INS, if the state vector contains IMU state, spatial/temporal LiDAR-IMU calibration and a plane feature, the system will have at least 7 unobservable directions as $\mathbf{N}^{(\pi)}$.

$$\mathbf{N}^{(\pi)} = \begin{bmatrix} \mathbf{N}_1^{(\pi)} & \mathbf{N}_{2:4}^{(\pi)} & \mathbf{N}_{5:6}^{(\pi)} & \mathbf{N}_7^{(\pi)} \end{bmatrix} \tag{27}$$

$$= \begin{bmatrix} {}^{I_1}_G\hat{\mathbf{R}}{}^G\mathbf{g} & \mathbf{0}_3 & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & {}^{I_1}_G\hat{\mathbf{R}}{}^G\hat{\mathbf{n}}_\pi \\ -\lfloor {}^G\hat{\mathbf{p}}_{I_1} \rfloor {}^G\mathbf{g} & {}^G\hat{\mathbf{R}}_\pi & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} \\ -\lfloor {}^G\hat{\mathbf{v}}_{I_1} \rfloor {}^G\mathbf{g} & \mathbf{0}_3 & {}^G\hat{\mathbf{n}}_1^\perp & {}^G\hat{\mathbf{n}}_2^\perp & \mathbf{0}_{3\times1} \\ \mathbf{0}_{13\times1} & \mathbf{0}_{13\times3} & \mathbf{0}_{13\times1} & \mathbf{0}_{13\times1} & \mathbf{0}_{13\times1} \\ -\lfloor {}^G\hat{d}_\pi {}^G\hat{\mathbf{n}}_\pi \rfloor {}^G\mathbf{g} & {}^G\hat{\mathbf{n}}_\pi\mathbf{e}_3^\top & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} \end{bmatrix}$$

where ${}^G\mathbf{R}_\pi = \begin{bmatrix} {}^G\mathbf{n}_1^\perp & {}^G\mathbf{n}_2^\perp & {}^G\mathbf{n} \end{bmatrix}$. The $\mathbf{N}_1^{(\pi)}$ relates to the global yaw around the gravity direction, $\mathbf{N}_{2:4}^{(\pi)}$ relate to the aided INS sensor platform, $\mathbf{N}_{5:6}^{(\pi)}$ relates to the velocity parallel to the plane and $\mathbf{N}_7^{(\pi)}$ relates to the rotation around the plane normal direction.

Given 3D random motions, $\mathbf{\Gamma}_{\pi16}$, $\mathbf{\Gamma}_{\pi18}$, $\mathbf{\Gamma}_{\pi26}$, $\mathbf{\Gamma}_{\pi27}$ and $\mathbf{\Gamma}_{\pi28}$ tend to have full column rank and make both the spatial and temporal calibration between LiDAR-IMU observable.

C. Degenerate Cases Analysis for LiDAR-IMU Calibration

Given the LiDAR-IMU navigation system with plane features, the online calibration will suffer from degenerate cases that make the calibration parameters to be unobservable. These degenerate cases can be affected by (1) plane structure and (2) system motion. In this section, we will use one-plane case with several degenerate motions to illustrate our findings (see Table. I). Two-plane or three-plane cases will be also included in our companion technique report. Note that the one-plane case refers to the cases when there is only one plane or all planes in the state vector are parallel. We have identified the following degenerate motions for the LiDAR-IMU calibration:

- If the system undergoes pure translation, the rigid transformation (including orientation and translation) between LiDAR-IMU will be unobservable with unobservable subspace as:

$$\mathbf{N}_{8:11}^{(\pi)} = \begin{bmatrix} \mathbf{0}_{15\times1} & \mathbf{0}_{15\times3} \\ {}^L_I\mathbf{R}{}^{I_1}_G\mathbf{R}{}^G\mathbf{n} & \mathbf{0}_3 \\ \mathbf{0}_{3\times1} & {}^L_I\mathbf{R}{}^{I_1}_G\mathbf{R}{}^G\mathbf{R}_\pi \\ 0 & 0 \\ \mathbf{0}_{3\times1} & \mathbf{e}_3^\top{}^G\mathbf{n} \end{bmatrix} \tag{28}$$

- If rotating with the fixed axis as ${}^L\mathbf{k}$, the translation between LiDAR-IMU is not observable along the rotation axis with unobservable directions as $\mathbf{N}_{12}^{(\pi)}$. Note that if the rotation axis is perpendicular to the plane direction, we will have an extra unobservable direction $\mathbf{N}_{13}^{(\pi)}$.

$$\mathbf{N}_{12:13}^{(\pi)} = \begin{bmatrix} \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} \\ {}^{I_1}_G\mathbf{R}{}^I_L\mathbf{R}{}^L\mathbf{k} & \mathbf{0}_{3\times1} \\ \mathbf{0}_{12\times1} & \mathbf{0}_{12\times1} \\ {}^L\mathbf{k} & {}^L\mathbf{k} \\ \mathbf{0}_{4\times1} & \mathbf{0}_{4\times1} \end{bmatrix} \tag{29}$$

- Similar to IMU-CAM calibration, if the system undergoes motions with constant ${}^I\boldsymbol{\omega}$ and ${}^I\mathbf{v}$ or constant ${}^I\boldsymbol{\omega}$ and ${}^G\mathbf{a}$, the LiDAR-IMU temporal calibration will also be unobservable with unobservable directions as $\mathbf{N}_{14}^{(\pi)}$ and $\mathbf{N}_{15}^{(\pi)}$, respectively. In addition, for one-plane case, we have an extra degenerate motion (${}^G\boldsymbol{\omega} \parallel {}^G\mathbf{n}$ and

${}^{G}\mathbf{n} \perp {}^{G}\mathbf{v}_I$) for time offset as $\mathbf{N}^{(\pi)}_{16}$.

$$\mathbf{N}^{(\pi)}_{14:16} = \begin{bmatrix} \mathbf{0}_{6\times 1} & \mathbf{0}_{6\times 1} & \mathbf{0}_{6\times 1} \\ \mathbf{0}_{3\times 1} & {}^{G}\mathbf{a}_I & \mathbf{0}_{3\times 1} \\ \mathbf{0}_{6\times 1} & \mathbf{0}_{6\times 1} & \mathbf{0}_{6\times 1} \\ {}^{L}_{I}\mathbf{R}{}^{I}\boldsymbol{\omega} & {}^{L}_{I}\mathbf{R}{}^{I}\boldsymbol{\omega} & \mathbf{0}_{3\times 1} \\ -{}^{L}_{I}\mathbf{R}{}^{I}\mathbf{v} & \mathbf{0}_{3\times 1} & \mathbf{0}_{3\times 1} \\ -1 & -1 & 1 \\ \mathbf{0}_{3\times 1} & \mathbf{0}_{3\times 1} & \mathbf{0}_{3\times 1} \end{bmatrix} \tag{30}$$

It can be seen that many of these degenerate motions for LiDAR-IMU coincide with the results of IMU-CAM calibration a few with additional directions. Pure translation will cause both the orientation and translation of LiDAR-IMU extrinsic calibration unobservable, whereas for IMU-CAM calibration just the translation is unobservable. In addition, one-plane case will also introduce extra unobservable directions, such as t_{dL}, if ${}^{G}\boldsymbol{\omega} \parallel {}^{G}\mathbf{n}$ and ${}^{G}\mathbf{n} \perp {}^{G}\mathbf{v}$. Note that any combination shown in Table I would also be degenerate.

TABLE I: Summary of degenerate motions for LiDAR-IMU calibration with one-plane feature.

One Plane / Parallel Planes	**Unobservable**
Pure Translation	${}^{L}_{I}\mathbf{R}$, ${}^{L}\mathbf{p}_I$
1-axis Rotation	${}^{L}\mathbf{p}_I$ along rotation axis
Constant ${}^{I}\omega$ and ${}^{I}\mathbf{v}$	t_{dL}, ${}^{L}\mathbf{p}_I$
Constant ${}^{I}\omega$ and ${}^{G}\mathbf{a}$	t_{dL}, ${}^{L}\mathbf{p}_I$
${}^{G}\boldsymbol{\omega} \parallel {}^{G}\mathbf{n}$ and ${}^{G}\mathbf{n} \perp {}^{G}\mathbf{v}$	t_{dL}

V. Simulation Results

We first verify our proposed system and observability analysis in simulation. A virtual room with structural planes (Fig. 3) is simulated [20, 21]. IMU measurements, LiDAR points, sparse image features, perturbation to the initial states, and noises to measurements are generated according to configuration shown in Table II. We first evaluate the proposed system with and without online LiDAR-IMU calibration by 12 Monte-Carlo runs, where absolute trajectory error (ATE) and Normalized Estimation Error Squared (NEES) are used to quantify accuracy and consistency, respectively.

The results are shown in Table III, where the "true" denotes the system starts with ground truth calibration parameters, while the "bad" indicates the system is initialized with perturbed calibration. The results suggest that the proposed system with online calibration can achieve consistent and accurate pose estimation. In comparison, the system will output inconsistent pose estimation (much larger ATE and NEES) if it starts from perturbed initial states and runs without online calibration. Furthermore, LIC-Fusion 2.0 also outperforms its IMU-CAM (IC) subsystem (which only fuses IMU and camera measurements). During the simulation, 14.61 MSCKF plane landmarks and 1.60 SLAM plane landmarks are used for update on average every scan.

TABLE II: Simulation setup parameters.

Parameter	**Value**	**Parameter**	**Value**
Cam Freq. (hz)	10	IMU Freq. (hz)	200
LiDAR Freq. (hz)	7	LiDAR Point Noise (m)	0.03
Gyro. White Noise	1.6968e-04	Gyro. Rand. Walk	1.9393e-05
Accel. White Noise	2.0000e-3	Accel. Rand. Walk	3.0000e-3
Pixel Proj. (px)	1	Timeoff (s)	0.01
Rot. LtoI (rad)	0.001	Pos. IinL (m)	0.01
Max Num. SLAM Point	12	Max Num. SLAM Plane	8
Num. Clones Image	11	Num. Clones LiDAR	8

TABLE III: Averaged ATE and NEES over 12 simulation runs with and without online calibration. Note that "true" means ground truth calibration while "bad" means the perturbed calibration and IC refers to IMU-camera subsystem only.

IMU Model	**ATE (deg)**	**ATE (m)**	**Ori. NEES**	**Pos. NEES**
true w/ calib	0.118	0.020	2.210	0.185
bad w/ calib	0.129	0.021	2.216	0.221
bad w/o calib	0.148	0.024	2.677	0.246
true w/o calib	0.122	0.021	2.233	0.208
IC true w/o calib	0.159	0.027	2.237	0.314

TABLE IV: Parameters used in our real world experiments.

Parameter	**Value**	**Parameter**	**Value**
Cam Freq. (hz)	20	IMU Freq. (hz)	400
LiDAR Freq. (hz)	10	Image Res. (px)	1920×1200
Num. Clones Image	11	Num. Clones LiDAR	8
Max Num. SLAM Point	20	Max Num. SLAM Plane	8

We further examine a degenerate motion (1-axis rotation motion) identified for online LiDAR-IMU calibration. With the same trajectory shown in Fig. 3, we remove orientation roll and pitch changes allowing only the yaw to change. The spatial-temporal calibration between LiDAR-IMU over 6 runs with online calibration are shown in Fig. 3. All calibration parameters except the z component of ${}^{L}\mathbf{p}_I$ converge nicely with shrunken uncertainty bounds. Because the sensor is rotating around z-axis (yaw only orientation), hence, the z-component of ${}^{L}\mathbf{p}_I$ is observable. Therefore, the results support our degenerate motion analysis, see Table I.

VI. Real-world Experimental Results

We further validate the proposed LIC-Fusion 2.0. using our multi-sensor platform that consists of a Velodyne VLP-16, an Xsens IMU, and a global-shutter monocular camera (see Fig. 4). All sensors publish asynchronously, with all time offsets estimated online with the zero as the initial guesses. The image processing pipeline is based on our prior work OpenVINS [21], while the LiDAR processing pipeline is proposed in this work. Note that IMU is necessary as the base sensor while, by design, the LiDAR and camera can be turned on/off without affecting performance. Videos are recorded when generating experimental results[2].

A. Teaching Building Sequences

The proposed system is first evaluated on data (Fig. 4 and Table IV) collected within a teaching building at Zhejiang University. Since we started and ended in the same position when collecting data, the start-to-end drift (supposed to be zero) is used for system performance evaluation (see Fig. 5). The averaged start and end errors of 5 runs tested on 7 sequences are shown in Table V. In the experiments, we compare the proposed plane landmarks enhanced LiDAR-IMU-CAM odometry (*LIC-Fusion 2.0*) with its subsystems (IMU-CAM system: *OpenVINS*, LiDAR-IMU system: *Proposed-LI*) and the other state-of-the-art algorithms, such as the LiDAR odometry, (*LOAM* [8]), the tightly-coupled LiDAR-Inertial odometry and mapping method (*LIO-MAP* [22]), and our prior work (*LIC-Fusion* [5]). Due to aggressive motion, degraded structures, lighting changes, some algorithms fail

[2] https://www.youtube.com/watch?v=waE5nepxD-Q, https://drive.google.com/open?id=1cLczzQVpsgtRQhuCXAHOO563gFJSZckX

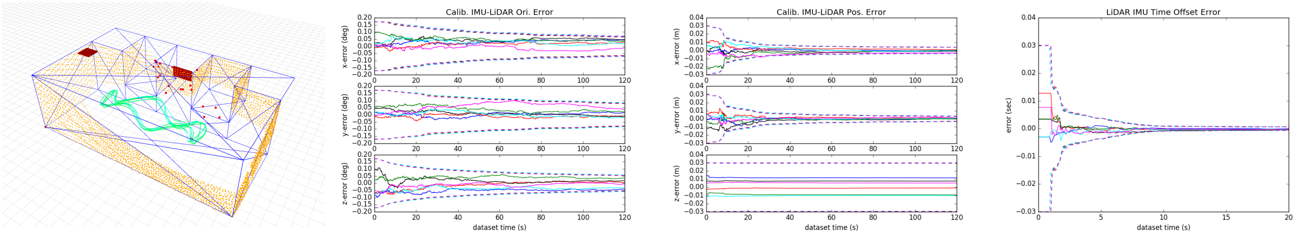

Fig. 3: Left: the simulated room with structure planes (blue), 16-beam LiDAR points (yellow), SLAM point landmarks (red dot), SLAM plane landmarks (red patch), estimated (green) and ground truth (cyan) trajectories. Middle and right: calibration errors and 3 sigma bounds for 6 typical runs with different initial state perturbations for 1-axis rotation motion (yaw only). z-component of ${}^{L}\mathbf{p}_I$ is not observable and does not converge at all.

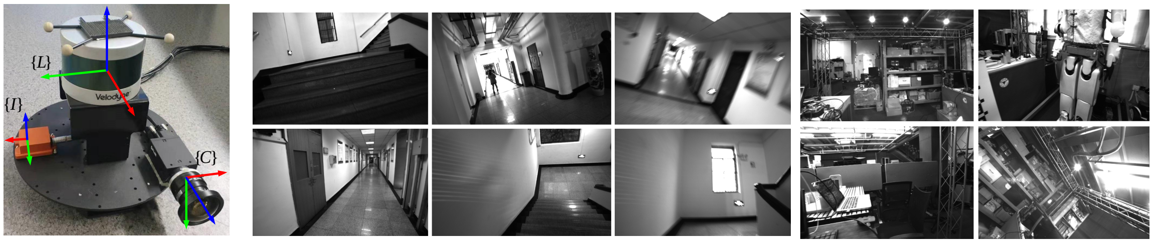

Fig. 4: Left: Sensor suite with a Velodyne VPL-16, Xsens IMU, and a monocular camera. Middle: Snapshots of Teaching Building sequences. Right: Snapshots of Vicon Room sequences.

to work on certain sequences. In the Table V, we omit severe failures marked by "-" when the norm of final drift is larger than 30 meters. In Seq 1, the camera-based OpenVINS fails to track visual features due to huge camera exposure changes when we go upstairs under poor lighting conditions. The proposed-LI subsystem has a larger drift on Seq 3 and Seq 6, in which the sensor suite traversed long corridors with only parallel planes observed. LIO-MAP also fails on Seq 3 with long corridors even with a maintained global map. In general, compared to other algorithms, the proposed LIC-Fusion 2.0 is more robust and can achieve higher accuracy on most sequences. Note that in a typical indoor scenario of Seq 5, there are 18.81 MSCKF planes and 2.09 SLAM planes used for the update on average.

B. Vicon Room Sequences

Data sequences collected within a VICON are also used for system evaluation. Clutters in the environment (shown in Fig. 4) pose challenges for data associations of LiDAR points. The averaged ATE [23] are computed with the provided ground truth to compare the LIC-Fusion 2.0, OpenVINS-IC, Proposed-LI, LOAM, LIO-MAP, and LIC-Fusion. The results are shown in Table. VI and Fig. 5, the cases with transitional errors more than 20 meters are marked with "-". The proposed LIC-Fusion 2.0 with reliable data associations over the sliding window outperforms the other algorithms. We appreciate the help from the authors of LIO-MAP [22] for parameters tuning to achieve better accuracy. However, LIO-MAP still fails on some sequences due to error-prone data association in clutter environment and lack of time synchronization between LiDAR and IMU.

The results demonstrate that LIC-Fusion 2.0 with the novel temporal plane tracking and online spatial/temporal calibration can achieve better accuracy than existing LiDAR-IMU-CAM fusion algorithms. We further examine the computational cost (shown in Fig. 6) of the main stages when running it on Seq 6 on a desktop computer with Intel i7-8086k CPU@4.0GHz. The averaged processing time for its IMU-CAM subsystem is 0.0168 seconds, and for its LiDAR-IMU subsystem is 0.0402 seconds. Thus LIC-Fusion 2.0 is suitable for real-time applications in this indoor scenario.

VII. Conclusions and Future Work

In this paper, we have developed a robust and efficient sliding-window plane-feature tracking algorithm to process 3D LiDAR point cloud measurements. We integrated this tracking algorithm into our prior LIC-Fusion estimator resulting in LIC-Fusion 2.0 with improved performance. In particular, during the proposed plane-feature tracking, we have advocated a new outlier rejection criteria to improve feature matching quality by taking to account the uncertainty of the LiDAR frame transformations. Additionally, we have investigated the observability properties of the linearized LIC system model in-depth and identified the degenerate cases for spatial-temporal LiDAR-IMU calibration with plane features. The proposed approach has been validated in both simulated and real-world datasets and shown to achieve better accuracy than the state-of-the-art algorithms. In the future, sliding-window edge-feature tracking in LiDAR scans will be investigated.

References

[1] J. Zhang and S. Singh. "Laser – visual – inertial odometry and mapping with high robustness and low drift". In: *Journal of Field Robotics* 35.8 (2018), pp. 1242–1264.

[2] J. Graeter, A. Wilczynski, and M. Lauer. "LIMO: Lidar-Monocular Visual Odometry". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2018, pp. 7872–7879.

[3] G. Wan, X. Yang, R. Cai, H. Li, Y. Zhou, H. Wang, and S. Song. "Robust and precise vehicle localization based on multi-sensor fusion in diverse city scenes". In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2018, pp. 4670–4677.

[4] W. Shao, S. Vijayarangan, C. Li, and G. Kantor. "Stereo visual inertial lidar simultaneous localization and mapping". In: *arXiv preprint arXiv:1902.10741* (2019).

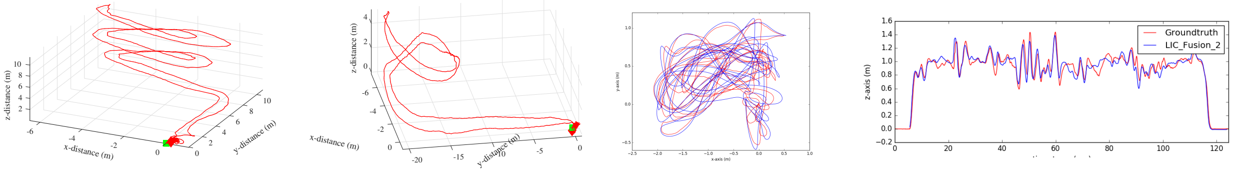

Fig. 5: Left Two: Estimated trajectories with LIC-Fusion 2.0 on Teaching Bulding Seq 1 and 2. Right Two: Estimated trajectories with LIC-Fusion 2.0 on Vicon Room Seq 2 overlaid with ground truth.

TABLE V: Averaged Start-to-End drift Error of 5 runs on Teaching Building Sequences (unit meters). The lengths for Seq1 - Seq 7 are around 108, 124, 237, 195, 85, 140, 83 meters, respectively. Note that estimated trajectories on Seq 1 and 2 are shown in Fig. 5.

Methods	**Seq 1**	**Seq 2**	**Seq 3**	**Seq 4**	**Seq 5**	**Seq 6**	**seq7**
LIC-Fusion 2.0	0.213, 0.074, 0.338	0.136, -0.107, -0.140	**0.689, -0.404, -0.172**	**0.456, 0.122, -0.322**	**0.054, -0.168, -0.027**	**0.025, -0.654, 0.199**	1.911, 0.226, -0.166
OpenVINS-IC	-, -, -	-1.765,-1.149,-0.836	3.917, 3.552, -0.475	3.181, -0.595, -1.372	-1.093,-0.083,-0.362	-0.085,-3.223,-0.143	-2.312, 1.562, 0.247
Proposed-LI	0.401, -0.195, 0.655	0.203, 0.503, 0.037	-, -, -	0.164,22.251,0.502	1.542, -2.110, 0.342	-, -, -	1.242, -0.462, -0.530
LOAM	0.831, -5.145, -0.607	**-0.059, -0.065, 0.073**	-3.418, 3.938, -21.364	-0.933, -8.395, 0.098	-9.014, 1.084, -0.300	-0.130, 0.461, 2.960	1.612, 0.000, -2.867
LIO-MAP	**-0.104, 0.057, 0.092**	-0.019, -0.423, 0.223	-, -, -	0.471, -0.215, -1.37	0.147, 0.017, -0.232	0.206, 0.125, 1.530	**0.019, -0.039, -0.142**
LIC-Fusion	-0.740, 0.0401, 0.222	0.293, 0.984, -0.656	1.216, 1.831, -0.465	-1.117, 0.607, 0.529	-0.382, -2.248, -0.905	-3.295, -1.934, 0.585	-0.912, -0.847, 0.377

TABLE VI: Averaged ATE of 5 runs on Vicon Room Sequences (units degrees/meters). The lengths for Seq 1 - Seq 6 are 42.62, 84.16, 33.92, 53.14, 49.74, 87.87 meters, respectively. Note that estimated trajectory on Seq 2 is shown in Fig. 5

Methods	**Seq 1**	**Seq 2**	**Seq 3**	**Seq 4**	**Seq 5**	**Seq 6**	**Average**
LIC-Fusion 2.0	2.537 / 0.097	1.870 / 0.145	**1.940** / **0.101**	**2.081** / 0.116	**2.710** / 0.104	**3.320** / **0.113**	**2.410** / **0.113**
OpenVINS-IC	2.625 / **0.094**	**1.741** / 0.177	3.131 / 0.273	2.404 / **0.115**	2.962 / 0.129	3.953 / 0.129	2.803 / 0.153
Proposed-LI	2.333 / 0.199	3.325 / 0.444	2.810 / 0.306	5.335 / 0.272	3.332 / 0.440	4.866 / 0.412	3.667 / 0.345
LOAM	5.880 / 0.156	6.414 / **0.134**	15.384 / 0.333	6.354 / 0.150	5.542 / 0.140	7.095 / 0.188	7.778 / 0.183
LIO-MAP	- / -	5.608 / 0.214	- / -	- / -	4.890 / 0.170	12.862 / 0.238	7.786 / 0.207
LIC-Fusion	**2.345** / 0.097	1.879 / 0.173	1.973 / 0.104	- / -	2.743 / **0.100**	3.788 / 0.131	2.546 / 0.121

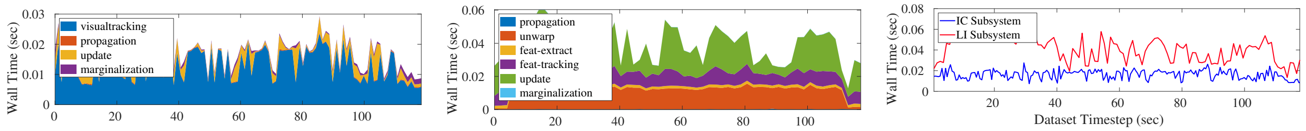

Fig. 6: The run time analysis of the proposed system.

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

[6] P. J. Besl and N. D. McKay. "Method for registration of 3-D shapes". In: *Sensor fusion IV: control paradigms and data structures*. Vol. 1611. International Society for Optics and Photonics. 1992, pp. 586–606.

[7] M. Velas, M. Spanel, and A. Herout. "Collar line segments for fast odometry estimation from velodyne point clouds". In: *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2016, pp. 4486–4495.

[8] J. Zhang and S. Singh. "LOAM: Lidar Odometry and Mapping in Real-time." In: *Robotics: Science and Systems*. Vol. 2. 2014, p. 9.

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

[10] J. A. Hesch, D. G. Kottas, S. L. Bowman, and S. I. Roumeliotis. "Consistency Analysis and Improvement of Vision-aided Inertial Navigation". In: *IEEE Transactions on Robotics* 30.1 (2014), pp. 158–176. ISSN: 1941-0468.

[11] Y. Yang, P. Geneva, K. Eckenhoff, and G. Huang. "Degenerate motion analysis for aided ins with online spatial and temporal sensor calibration". In: *IEEE Robotics and Automation Letters* 4.2 (2019), pp. 2070–2077.

[12] N. Trawny and S. I. Roumeliotis. "Indirect Kalman filter for 3D attitude estimation". In: *University of Minnesota, Dept. of Comp. Sci. & Eng., Tech. Rep* 2 (2005), p. 2005.

[13] Y. Yang, P. Geneva, X. Zuo, K. Eckenhoff, Y. Liu, and G. Huang. "Tightly-coupled aided inertial navigation with point and plane features". In: *International Conference on Robotics and Automation (ICRA)*. IEEE. Montreal, Canada, May 2019, pp. 6094–6100.

[14] M. Li and A. I. Mourikis. "Optimization-based estimator design for vision-aided inertial navigation". In: *Robotics: Science and Systems*. Berlin Germany. 2013, pp. 241–248.

[15] X. Zuo, Y. yang, P. Geneva, J. Lv, Y. Liu, G. Huang, and M. Pollefeys. "Technique Report of LIC-Fusion 2.0 with Temporal Plane Tracking". In: *Ethz, Dept. of Comp. Sci., Tech. Rep* 1 (2020). Available: http://udel.edu/~ghuang/papers/tr_lic2.pdf.

[16] Y. Yang, J. Maley, and G. Huang. "Null-Space-based Marginalization: Analysis and Algorithm". In: *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Vancouver, Canada, 2017, pp. 6749–6755.

[17] Y. Yang and G. Huang. "Aided inertial navigation with geometric features: Observability analysis". In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2018, pp. 2334–2340.

[18] S. Ceriani, C. Sánchez, P. Taddei, E. Wolfart, and V. Sequeira. "Pose interpolation SLAM for large maps using moving 3D sensors". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2015, pp. 750–757.

[19] Y. Yang and G. Huang. "Observability Analysis of Aided INS With Heterogeneous Features of Points, Lines, and Planes". In: *IEEE Transactions on Robotics* 35.6 (2019), pp. 1399–1418. ISSN: 1941-0468.

[20] P. Geneva, K. Eckenhoff, Y. Yang, and G. Huang. "LIPS: LiDAR-Inertial 3D Plane SLAM". In: *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*. Madrid, Spain, 2018.

[21] P. Geneva, K. Eckenhoff, W. Lee, Y. Yang, and G. Huang. "OpenVINS: A Research Platform for Visual-Inertial Estimation". In: *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*. Paris, France, 2020.

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

[23] Z. Zhang and D. Scaramuzza. "A tutorial on quantitative trajectory evaluation for visual (-inertial) odometry". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2018, pp. 7244–7251.