Technique Report of 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]

[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.

Tech Report-2020-LICFusion2
Last Updated - March 13, 2020

Contents

1 Problem Formulation

1.1 State Vector

Besides the IMU states $\mathbf{x}_I$, the IMU clones at the time instants of receiving camera and LiDAR measurements $\mathbf{x}_C$ and $\mathbf{x}_L$, we also estimate the extrinsics between camera and IMU $\mathbf{x}_{calib\_C}$ and the extrinsics between LiDAR and IMU $\mathbf{x}_{calib\_L}$ as [1]. We keep m and n clones in the sliding window corresponding to images and LiDAR scans, respectively. Furthermore, we also incorporate the stably tracked SLAM point landmarks ${}^G\mathbf{x}_f$ and SLAM plane landmarks ${}^A\mathbf{x}_\pi$ into the state vector. These SLAM features are "long lived" and through frequent matching can limit estimation drift. The point landmark are denoted in global frame $\{G\}$, while the plane landmark are denoted in its anchor frame $\{A\}$, which is the LiDAR frame whether it was firstly observed. To limit the computational cost of the system, we keep point and plane landmarks in the state vector up to maximums of g and h, respectively. In summary, the state vector is:

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

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 [2], which represents 3D rotation ${}^{I_k}_G\mathbb{R}$ from $\{I_k\}$ to $\{G\}$. ${}^G\mathbf{v}_{I_k}$, ${}^G\mathbf{p}_{I_k}$ denotes 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].

Furthermore, In Eq. 3 and Eq. 4, t_{dL} and t_{dC} are the time offsets between LiDAR - IMU, and Camera - IMU. IMU is the base sensor in our framework, we need to know the time offsets between IMU and the other sensors. The reported timestamp t_C and t_L from camera and LiDAR have the following relationship with the IMU clock t_I:

$$t_I = t_C + t_{dC} \tag{9a}$$

$$t_I = t_L + t_{dL} \tag{9b}$$

Once a LiDAR scan received at time $t_{L_{k+1}}$ corresponding to IMU clock $t_{I_{k+1}}$, we will propagate the IMU state $\mathbf{x}_I$ from time t_{I_k} to $t_{I_{k+1}}$. And clone the IMU pose in the propagated $\mathbf{x}_I$ at time $t_{I_{k+1}}$ into the state

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

vector. Since the cloned pose is a function of t_{dL}, once the cloned pose is updated, t_{dL} can be also updated [1, 3]. Analogously, the time offset between camera and IMU t_{dC} can be updated with the cloned poses at receiving images.

We additionally store 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 [4, 5], and the anchored representation can avoid the singularity when the norm of ${}^G\mathbf{p}_\pi$ approaches zero [4]. These long-lived planar features will be tracked in incoming LiDAR scans using the proposed tracking algorithm until they are lost.

With the above notation, the relative pose $\{{}^{L_b}_{L_a}\mathbf{R}, {}^{L_b}\mathbf{p}_{L_a}\}$ between two LiDAR frames L_a and L_b at time instants t_a and t_b in IMU clock can be computed by IMU clones $\{{}^{I_a}_{G}\mathbf{R}, {}^{G}\mathbf{p}_{I_a}\}$, $\{{}^{I_b}_{G}\mathbf{R}, {}^{G}\mathbf{p}_{I_b}\}$, and extrinsics $\{{}^{L}_{I}\mathbf{R}, {}^{L}\mathbf{p}_{I}\}$:

$$ {}^{L_b}_{L_a}\mathbf{R} = {}^{L}_{I}\mathbf{R}{}^{I_b}_{G}\mathbf{R}\left({}^{L}_{I}\mathbf{R}{}^{I_a}_{G}\mathbf{R}\right)^\top \tag{10a} $$

$$ {}^{L_b}\mathbf{p}_{L_a} = {}^{L}_{I}\mathbf{R}{}^{I_b}_{G}\mathbf{R}\left({}^{I_a}_{G}\mathbf{R}^{\top I}\mathbf{p}_L + {}^G\mathbf{p}_{I_a} - {}^G\mathbf{p}_{I_b}\right) + {}^L\mathbf{p}_I \tag{10b} $$

Furthermore, a LiDAR point ${}^{L_a}\mathbf{p}_f$ in frame L_a can be transformed into L_b through

$$ {}^{L_b}\mathbf{p}_f = {}^{L_b}_{L_a}\mathbf{R}{}^{L_a}\mathbf{p}_f + {}^{L_b}\mathbf{p}_{L_a} \tag{11} $$

1.2 Point-to-Plane Measurement Model

We leverage the high-frequency IMU measurement for prediction. And the image processing pipeline is the same with MSCKF [6, 7] and OpenVINS [8]. To summarize the image pipeline in brief, after triangulating point landmarks from its observations tracked in multiple images, the residual based on reprojection error will be used to update the related states in the state vector. Few point landmarks with a long track length will be initialized in the state vector as SLAM point landmarks, most point landmarks with short track length will not added into the state vector as MSCKF point landmarks. MSCKF point landmarks will not be dropped after updating with their observations. Here we only present the plane landmarks based LiDAR processing pipeline into details.

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

$$ \mathbf{z}_\pi = \frac{{}^L\mathbf{p}_\pi^\top}{\left\|{}^L\mathbf{p}_\pi\right\|}({}^L\mathbf{p}_f - \mathbf{n}_f) - \left\|{}^L\mathbf{p}_\pi\right\| \tag{12} $$

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:

$$ \begin{bmatrix} {}^L\mathbf{n} \\ {}^Ld \end{bmatrix} = \begin{bmatrix} {}^L_A\mathbf{R} & 0 \\ -{}^A\mathbf{p}_L^\top & 1 \end{bmatrix} \begin{bmatrix} {}^A\mathbf{n} \\ {}^Ad \end{bmatrix} \tag{13} $$

1.3 LiDAR Plane Feature Update

1.3.1 MSCKF Plane Feature

Analogous to point features [7], we divide all the tracked plane features from the LiDAR pointclouds into "MSCKF" and "SLAM" based on the track length. Note that the sliding-window-based plane tracking is explained in detail in [9]. Considering we have a series of measurements collected over the whole sliding window of the plane feature ${}^{A}\mathbf{p}_{\pi_j}$, we can linearize the measurements $\mathbf{z}_f^{(j)}$ in Eq. (12) at current estimates of ${}^{A}\mathbf{p}_{\pi_j}$ and the states $\mathbf{x}$ as:

$$\mathbf{r}_f^{(j)} = \mathbf{0} - \mathbf{z}_f^{(j)} \simeq \mathbf{H}_x^{(j)}\tilde{\mathbf{x}} + \mathbf{H}_\pi^{(j)}{}^{L_b}\tilde{\mathbf{p}}_{\pi_j} + \mathbf{H}_n^{(j)}\mathbf{n}_f^{(j)} \tag{14}$$

where $\mathbf{n}^{(j)}$ 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.

Considering only one measurement $\mathbf{z}_{f_i}^{(j)}$ induced by the planar LiDAR point observement ${}^{L_a}\mathbf{p}_i$ of the plane feature ${}^{L_b}\mathbf{p}_{\pi_j}$, we can linearize the measurement as follow. Firstly, $\mathbf{H}_{x_i}^{(j)}$ is the Jacobian of $\mathbf{z}_{f_i}^{(j)}$ with respect to the IMU clones $\mathbf{x}_{Ia} = [{}^{I_a}_{G}\bar{q}^\top, {}^{G}\mathbf{p}_{I_a}^\top]^\top$, $\mathbf{x}_{Ib} = [{}^{I_b}_{G}\bar{q}^\top, {}^{G}\mathbf{p}_{I_b}^\top]^\top$, and extrinsics $\mathbf{x}_{LI} = [{}^{L}_{I}\bar{q}^\top, {}^{L}\mathbf{p}_I^\top]^\top$ in the state vector. $\mathbf{H}_{x_i}^{(j)}$ is composed by

$$\frac{\partial \tilde{\mathbf{z}}_{f_i}^{(j)}}{\partial \tilde{\mathbf{x}}_{Ia}} = \frac{{}^{L_b}\mathbf{p}_{\pi_j}^\top}{\left\|{}^{L_b}\mathbf{p}_{\pi_j}\right\|} * \left[{}^{L}_{I}\mathbf{R}{}^{I_b}_{G}\mathbf{R}{}^{I_a}_{G}\mathbf{R}^\top \lfloor {}^{L}_{I}\mathbf{R}^\top({}^{L}\mathbf{p}_I - {}^{L_a}\mathbf{p}_{f_i})\rfloor \quad {}^{L}_{I}\mathbf{R}{}^{I_b}_{G}\mathbf{R}\right] \tag{15}$$

$$\frac{\partial \tilde{\mathbf{z}}_{f_i}^{(j)}}{\partial \tilde{\mathbf{x}}_{I_b}} = \frac{{}^{L_b}\mathbf{p}_{\pi_j}^\top}{\left\|{}^{L_b}\mathbf{p}_{\pi_j}\right\|} * \left[{}^{L}_{I}\mathbf{R} * \lfloor {}^{I_b}_{G}\mathbf{R}{}^{I_a}_{G}\mathbf{R}^\top{}^{L}_{I}\mathbf{R}^\top\left({}^{L_a}\mathbf{p}_{f_i} - {}^{L}\mathbf{p}_I\right) - {}^{I_b}_{G}\mathbf{R}\left({}^{G}\mathbf{p}_{I_b} - {}^{G}\mathbf{p}_{I_a}\right)\rfloor \quad -{}^{L}_{I}\mathbf{R}{}^{I_b}_{G}\mathbf{R}\right] \tag{16}$$

$$\frac{\partial \tilde{\mathbf{z}}_{f_i}^{(j)}}{\partial \tilde{\mathbf{x}}_{LI}} = \frac{{}^{L_b}\mathbf{p}_{\pi_j}^\top}{\left\|{}^{L_b}\mathbf{p}_{\pi_j}\right\|} * \left[{}^{L_b}_{L_a}\mathbf{R}\lfloor {}^{L}\mathbf{p}_I - {}^{L_a}\mathbf{p}_{f_i}\rfloor + \lfloor {}^{L_b}_{L_a}\mathbf{R}\left({}^{L_a}\mathbf{p}_{f_i} - {}^{L}\mathbf{p}_I\right) - {}^{L}_{I}\mathbf{R}{}^{I_b}_{G}\mathbf{R}\left({}^{G}\mathbf{p}_{I_b} - {}^{G}\mathbf{p}_{I_a}\right)\rfloor \quad \mathbf{I} - {}^{L_b}_{L_a}\mathbf{R}\right] \tag{17}$$

Then,

$$\mathbf{H}_{\pi_i}^{(j)} = \frac{\partial \tilde{\mathbf{z}}_{f_i}^{(j)}}{\partial {}^{L_b}\tilde{\mathbf{p}}_{\pi_j}} = \frac{{}^{L_b}\mathbf{p}_{f_i}^\top}{{}^{L_b}d_{\pi_j}} - \left(\frac{{}^{L_b}\mathbf{p}_{f_i}^\top{}^{L_b}\mathbf{n}_{\pi_j}}{{}^{L_b}d_{\pi_j}} + 1\right){}^{L_b}\mathbf{n}_{\pi_j}^\top \tag{18}$$

and

$$\mathbf{H}_{n_i}^{(j)} = \frac{\partial \tilde{\mathbf{z}}_{f_i}^{(j)}}{\partial {}^{L_a}\mathbf{n}_{f_i}} = \frac{{}^{L_b}\mathbf{p}_{\pi_j}^\top}{\left\|{}^{L_b}\mathbf{p}_{\pi_j}\right\|} * {}^{I_b}_{I_a}\mathbf{R} \tag{19}$$

If ${}^{A}\mathbf{p}_{\pi_j}$ is a MSCKF plane landmark, the nullspace operation [10] is performed to remove the dependency on ${}^{A}\mathbf{p}_{\pi_j}$ by projection onto the left nullspace $\mathbf{N}$ of :

$$\mathbf{N}^\top\mathbf{r}_f^{(j)} = \mathbf{N}^\top\mathbf{H}_x^{(j)}\tilde{\mathbf{x}} + \mathbf{N}^\top\mathbf{H}_\pi^{(j)}{}^{L_b}\tilde{\mathbf{p}}_\pi + \mathbf{N}^\top\mathbf{H}_n^{(j)}\mathbf{n}_f^{(j)} \tag{20}$$

$$\Rightarrow \mathbf{r}_{fo}^{(j)} = \mathbf{H}_{xo}^{(j)}\tilde{\mathbf{x}} + \mathbf{n}_o^{(j)} \tag{21}$$

Due to the special structure that $\mathbf{H}_n^{(j)}\mathbf{H}_n^{(j)\top} = \mathbf{I}_n$ the measurement covariance is still is isotropic and thus the nullspace operation is still 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:

$$\mathbf{r}_{fo} = \mathbf{H}_{xo}\tilde{\mathbf{x}} + \mathbf{n}_o \tag{22}$$

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

1.3.2 SLAM Plane Feature

If ${}^{L_a}\mathbf{p}_\pi$ is a SLAM plane landmark that already exists in the state, we can directly update its estimate and the state using Eq. (14). 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 [11]. 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). Along with augmenting the SLAM plane feature into state vector, we also need to augment the state covariance matrix with the plane feature's initial covariance and cross-correlation with the other states, which is analogous to the SLAM point feature [12]. Since ${}^{L_a}\mathbf{p}_\pi$ is anchored in local LiDAR reference frame, once frame L_a needs to be removed from the sliding window, we will change its anchor frame to the newest LiDAR frame L_b in the sliding window by:

$$
{}^{L_b}\mathbf{p}_\pi = {}^{L_b}\mathbf{n}_\pi {}^{L_b}d_\pi = {}^{L_b}_{L_a}\mathbf{R}^{L_a}\mathbf{n}\left({}^{L_b}\mathbf{p}_{L_a}^\top {}^{L_b}_{L_a}\mathbf{R}^{L_a}\mathbf{n}_\pi + {}^{L_a}d_\pi\right) \tag{23}
$$

And the related states are $\mathbf{x}_{Ia} = [{}^{I_a}_G\bar{q}^\top, {}^G\mathbf{p}_{I_a}^\top]^\top$, $\mathbf{x}_{Ib} = [{}^{I_b}_G\bar{q}^\top, {}^G\mathbf{p}_{I_b}^\top]^\top$, extrinsics $\mathbf{x}_{LI} = [{}^L_I\bar{q}^\top, {}^L\mathbf{p}_I^\top]^\top$, and the plane feature ${}^{L_a}\mathbf{p}_\pi$.

$$
\frac{\partial^{L_b}\tilde{\mathbf{p}}_{\pi_j}}{\partial^{L_a}\tilde{\mathbf{p}}_{\pi_j}} = \frac{1}{{}^{L_a}d_\pi}\left[\left({}^{L_a}d_\pi + {}^{L_b}\mathbf{p}_{L_a}^\top {}^{L_b}_{L_a}\mathbf{R}^{L_a}\mathbf{n}_\pi\right){}^{L_b}_{L_a}\mathbf{R} + {}^{L_b}_{L_a}\mathbf{R}^{L_a}\mathbf{n}_\pi {}^{L_b}\mathbf{p}_{L_a}^\top {}^{L_b}_{L_a}\mathbf{R}\right]\left(\mathbf{I} - {}^{L_a}\mathbf{n}_\pi {}^{L_a}\mathbf{n}_\pi^\top\right) + {}^{L_b}_{L_a}\mathbf{R}^{L_a}\mathbf{n}_\pi {}^{L_a}\mathbf{n}_\pi^\top \tag{24}
$$

where $\{{}^{L_b}_{L_a}\mathbf{R}, {}^{L_b}\mathbf{p}_{L_a}\}$ can be computed as Eq. 10. With a slight abuse of notation, $\mathbf{x}_{ab} = [{}^{L_b}_{L_a}\bar{q}^\top, {}^{L_b}\mathbf{p}_{L_a}^\top]^\top$, we can compute Jacobians as

$$
\frac{\partial^{L_b}\tilde{\mathbf{p}}_{\pi_j}}{\partial\tilde{\mathbf{x}}_{ab}} = \begin{bmatrix} \lfloor {}^{L_b}\mathbf{p}_\pi \rfloor + {}^{L_b}_{L_a}\mathbf{R}^{L_a}\mathbf{n}^{L_b}\mathbf{p}_{L_a}^\top \lfloor {}^{L_b}_{L_a}\mathbf{R}^{L_a}\mathbf{n}\rfloor & {}^{L_b}_{L_a}\mathbf{R}^{L_a}\mathbf{n}\left({}^{L_b}_{L_a}\mathbf{R}^{L_a}\mathbf{n}\right)^\top \end{bmatrix} \tag{25}
$$

and

$$
\frac{\partial\tilde{\mathbf{x}}_{ab}}{\partial\tilde{\mathbf{x}}_{I_a}} = \begin{bmatrix} -{}^L_I\mathbf{R}{}^{I_b}_G\mathbf{R}{}^{I_a}_G\mathbf{R}^\top & \mathbf{0} \\ {}^L_I\mathbf{R}{}^{I_b}_G\mathbf{R}{}^{I_a}_G\mathbf{R}^\top \lfloor {}^L_I\mathbf{R}^\top {}^L\mathbf{p}_I \rfloor & {}^L_I\mathbf{R}{}^{I_b}_G\mathbf{R} \end{bmatrix} \tag{26}
$$

$$
\frac{\partial\tilde{\mathbf{x}}_{ab}}{\partial\tilde{\mathbf{x}}_{I_b}} = \begin{bmatrix} {}^L_I\mathbf{R} & \mathbf{0} \\ -{}^L_I\mathbf{R}\lfloor {}^{I_b}_G\mathbf{R}\left({}^{I_a}_G\mathbf{R}^\top {}^L_I\mathbf{R}^\top {}^L\mathbf{p}_I + {}^G\mathbf{p}_{I_b} - {}^G\mathbf{p}_{I_a}\right)\rfloor & -{}^L_I\mathbf{R}{}^{I_b}_G\mathbf{R} \end{bmatrix} \tag{27}
$$

$$
\frac{\partial\tilde{\mathbf{x}}_{ab}}{\partial\tilde{\mathbf{x}}_{LI}} = \begin{bmatrix} \mathbf{I} - {}^{L_b}_{L_a}\mathbf{R} & \mathbf{0} \\ \lfloor {}^{L_b}_{L_a}\mathbf{R}^L\mathbf{p}_I - {}^L_I\mathbf{R}{}^{I_b}_G\mathbf{R}\left({}^G\mathbf{p}_{I_b} - {}^G\mathbf{p}_{I_a}\right)\rfloor - {}^{L_b}_{L_a}\mathbf{R}\lfloor {}^L\mathbf{p}_I\rfloor & \mathbf{I} - {}^{L_b}_{L_a}\mathbf{R} \end{bmatrix} \tag{28}
$$

2 Observability Analysis

The observability analysis of vision-aided-inertial navigation system with online calibration has been studied extensively in iteratures [11, 13, 14], however, the analysis for LiDAR-aided-Inertial navigation with online calibration using plane features is still missing. In addition, since the calibration between IMU-CAM and IMU-LiDAR calibration are independent, previously identified degenerate motions for VINS calibration cannot be directly applied to IMU-LiDAR cases with plane features. Hence, in this paper, we focus on the subsystem of LIC-Fusion 2.0 with IMU and LiDAR only and study specifically the degenerate cases for online spatial-temporal IMU-LiDAR calibration using plane features. In particular, the observability matrix $\mathbf{M}(\mathbf{x})$ is given by:

$$\mathbf{M}(\mathbf{x}) = \begin{bmatrix} \mathbf{H}_{\mathbf{x},1}\mathbf{\Phi}_{(1,1)} \\ \vdots \\ \mathbf{H}_{\mathbf{x},k}\mathbf{\Phi}_{(k,1)} \end{bmatrix} \tag{29}$$

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

2.1 State Vector and State Transition Matrix

As in our previous work [15], the closest point of the plane is the point with the smallest distance from the plane to the origin. The closet point for plane can be written as:

$$\mathbf{p}_\pi = d_\pi \mathbf{n}_\pi \tag{30}$$

Hence, the state vector with a CP plane in the state vector can be written as:

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_I^\top & \mathbf{x}_{calib_L}^\top & {}^G\mathbf{p}_\pi^\top \end{bmatrix}^\top \tag{31}$$

Note that we further denote $\mathbf{x}_{calib_L} = \begin{bmatrix} \mathbf{x}_{LI}^\top & t_{dL} \end{bmatrix}^\top$ where $\mathbf{x}_{LI}$ represents the rigid transformation $({}^L_I\bar{q}, {}^L\mathbf{p}_I)$ between IMU and LiDAR. The state transition matrix can be written as:

$$\mathbf{\Phi}_{(k,1)} = \begin{bmatrix} \mathbf{\Phi}_I & \mathbf{0}_{15\times 7} & \mathbf{0}_{15\times 3} \\ \mathbf{0}_{7\times 15} & \mathbf{\Phi}_{calib\_L} & \mathbf{0}_{7\times 3} \\ \mathbf{0}_{3\times 15} & \mathbf{0}_{3\times 7} & \mathbf{\Phi}_\pi \end{bmatrix} \tag{32}$$

Where $\mathbf{\Phi}_I$ denotes the IMU state transition matrix [13, 16]. $\mathbf{\Phi}_{calib\_L} = \mathbf{I}_7$ and $\mathbf{\Phi}_\pi = \mathbf{I}_3$. Note that without loss of generality for analysis, we represent the plane feature in the global frame $\{G\}$.

2.2 Point-to-Plane Measurement

We use the point-to-plane distance as the measurement for the plane as:

$$\mathbf{z}_\pi = \frac{{}^L\mathbf{p}_\pi^\top \left({}^L\mathbf{p}_f - \mathbf{n}_f\right)}{\left\|{}^L\mathbf{p}_\pi\right\|} - \left\|{}^L\mathbf{p}_\pi\right\| \tag{33}$$

where ${}^L\mathbf{p}_f^\top$ is a LiDAR point from the plane, $\mathbf{n}_f$ is the measurement noise. If we denote ${}^Ld = \left\|{}^L\mathbf{p}_\pi\right\|$ and ${}^L\mathbf{n} = \frac{{}^L\mathbf{p}_\pi}{\left\|{}^L\mathbf{p}_\pi\right\|}$, we can write the transformation of plane from $\{G\}$ to $\{L\}$ as:

$$\begin{bmatrix} {}^L\mathbf{n} \\ {}^Ld \end{bmatrix} = \begin{bmatrix} {}^L_I\mathbf{R} & \mathbf{0}_{3\times 1} \\ {}^L\mathbf{p}_I^\top {}^L_I\mathbf{R} & 1 \end{bmatrix} \begin{bmatrix} {}^I_G\mathbf{R} & \mathbf{0}_{3\times 1} \\ -{}^G\mathbf{p}_I^\top & 1 \end{bmatrix} \begin{bmatrix} {}^G\mathbf{n} \\ {}^Gd \end{bmatrix} \tag{34}$$

Note that ${}^I_G\mathbf{R}$ and ${}^G\mathbf{p}_I$ is also a function of time offset. The measurement Jacobians w.r.t. to the state vector can be written as:

$$\mathbf{H}_{\mathbf{x}}^{(\pi)} = \frac{\partial \tilde{\mathbf{z}}_\pi}{\partial \tilde{\mathbf{x}}} = \begin{bmatrix} \frac{\partial \tilde{\mathbf{z}}_\pi}{\partial \tilde{\mathbf{x}}_I} & \frac{\partial \tilde{\mathbf{z}}_\pi}{\partial \tilde{\mathbf{x}}_{LI}} & \frac{\partial \tilde{\mathbf{z}}_\pi}{\partial \tilde{t}_{dL}} & \frac{\partial \tilde{\mathbf{z}}_\pi}{\partial {}^G\tilde{\mathbf{p}}_\pi} \end{bmatrix} \tag{35}$$

According to the chain rule, the Jacobians can be computed:

$$\frac{\partial \tilde{\mathbf{z}}_\pi}{\partial {}^L\tilde{\mathbf{p}}_\pi} = \frac{{}^L\mathbf{p}_f^\top}{{}^Ld_\pi}\left(\mathbf{I}_3 - {}^L\mathbf{n}_\pi{}^L\mathbf{n}_\pi^\top\right) - {}^L\mathbf{n}_\pi^\top \tag{36}$$

$$\tag{37}$$

$$\frac{\partial {}^L\tilde{\mathbf{p}}_\pi}{\partial \begin{bmatrix}{}^L\tilde{\mathbf{n}}^\top & {}^L\tilde{d}\end{bmatrix}^\top} = \begin{bmatrix}{}^Ld\mathbf{I}_3 & {}^L\mathbf{n}\end{bmatrix} \tag{38}$$

$$\frac{\partial \begin{bmatrix}{}^L\tilde{\mathbf{n}}^\top & {}^L\tilde{d}\end{bmatrix}^\top}{\partial \tilde{\mathbf{x}}_I} = \begin{bmatrix}{}^L_I\mathbf{R}\lfloor {}^I_G\mathbf{R}{}^G\mathbf{n}\rfloor & \mathbf{0}_3 \\ {}^L\mathbf{p}_{LI}^\top{}^L_I\mathbf{R}\lfloor {}^I_G\mathbf{R}{}^G\mathbf{n}\rfloor & {}^G\mathbf{n}^\top\end{bmatrix} \tag{39}$$

$$\frac{\partial \begin{bmatrix}{}^L\tilde{\mathbf{n}}^\top & {}^L\tilde{d}\end{bmatrix}^\top}{\partial \tilde{\mathbf{x}}_{LI}} = \begin{bmatrix}\lfloor {}^L_I\mathbf{R}{}^I_G\mathbf{R}{}^G\mathbf{n}\rfloor & \mathbf{0}_3 \\ {}^L\mathbf{p}_I^\top\lfloor {}^L_I\mathbf{R}{}^I_G\mathbf{R}{}^G\mathbf{n}\rfloor & {}^G\mathbf{n}^\top{}^I_G\mathbf{R}^\top{}^L_I\mathbf{R}^\top\end{bmatrix} \tag{40}$$

$$\frac{\partial \begin{bmatrix}{}^L\tilde{\mathbf{n}}^\top & {}^L\tilde{d}\end{bmatrix}^\top}{\partial \tilde{t}_{dL}} = \begin{bmatrix}{}^L_I\mathbf{R}\lfloor {}^I_G\mathbf{R}{}^G\mathbf{n}\rfloor{}^I\boldsymbol{\omega} \\ {}^L\mathbf{p}_I^\top{}^L_I\mathbf{R}\lfloor {}^I_G\mathbf{R}{}^G\mathbf{n}\rfloor{}^I\boldsymbol{\omega} - {}^G\mathbf{v}_I^\top{}^G\mathbf{n}\end{bmatrix} \tag{41}$$

$$\frac{\partial \begin{bmatrix}{}^L\tilde{\mathbf{n}}^\top & {}^L\tilde{d}\end{bmatrix}^\top}{\partial {}^G\tilde{\mathbf{p}}_\pi} = \begin{bmatrix}\frac{1}{{}^Gd}{}^L_I\mathbf{R}{}^I_G\mathbf{R}\left(\mathbf{I}_3 - {}^G\mathbf{n}{}^G\mathbf{n}^\top\right) \\ \frac{1}{{}^Gd}\left({}^L\mathbf{p}_I^\top{}^L_I\mathbf{R}{}^I_G\mathbf{R} - {}^G\mathbf{p}_I^\top\right)\left(\mathbf{I}_3 - {}^G\mathbf{n}{}^G\mathbf{n}^\top\right) + {}^G\mathbf{n}^\top\end{bmatrix} \tag{42}$$

For simplicity, we define:

$$\mathbf{H}_\pi = \frac{\partial \tilde{\mathbf{z}}_\pi}{\partial {}^L\tilde{\mathbf{p}}_\pi}\frac{\partial {}^L\tilde{\mathbf{p}}_\pi}{\partial \begin{bmatrix}{}^L\tilde{\mathbf{n}}^\top & {}^L\tilde{d}\end{bmatrix}^\top} \tag{43}$$

Therefore, we can get the overall measurement Jacobians as:

$$\mathbf{H}_\mathbf{x} = \mathbf{H}_\pi \begin{bmatrix}{}^L_I\mathbf{R}{}^I_G\mathbf{R} & \mathbf{0}_3 \\ \mathbf{0}_{1\times 3} & 1\end{bmatrix}\begin{bmatrix}\mathbf{H}_{11} & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{H}_{16} & \mathbf{0}_3 & \mathbf{H}_{18} & \mathbf{H}_{19} \\ \mathbf{H}_{21} & {}^G\mathbf{n}^\top & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{H}_{26} & \mathbf{H}_{27} & \mathbf{H}_{28} & \mathbf{H}_{29}\end{bmatrix} \tag{44}$$

where we have:

$$\mathbf{H}_{11} = \lfloor {}^G\mathbf{n}\rfloor {}^G_I\mathbf{R} \tag{45}$$

$$\mathbf{H}_{16} = \lfloor {}^G\mathbf{n}\rfloor {}^G_I\mathbf{R}{}^I_L\mathbf{R} \tag{46}$$

$$\mathbf{H}_{18} = \lfloor {}^G\mathbf{n}\rfloor {}^G_{I_k}\mathbf{R}{}^I\boldsymbol{\omega} \tag{47}$$

$$\mathbf{H}_{19} = \frac{1}{{}^Gd}{}^L_I\mathbf{R}{}^I_G\mathbf{R}\left(\mathbf{I}_3 - {}^G\mathbf{n}{}^G\mathbf{n}^\top\right) \tag{48}$$

$$\mathbf{H}_{21} = {}^L\mathbf{p}_I^\top{}^L_I\mathbf{R}\lfloor {}^I_G\mathbf{R}{}^G\mathbf{n}\rfloor \tag{49}$$

$$\mathbf{H}_{22} = {}^L\mathbf{p}_I^\top\lfloor {}^L_I\mathbf{R}{}^I_G\mathbf{R}{}^G\mathbf{n}\rfloor \tag{50}$$

$$\mathbf{H}_{27} = {}^G\mathbf{n}^\top{}^I_G\mathbf{R}^\top{}^L_I\mathbf{R}^\top \tag{51}$$

$$\mathbf{H}_{28} = {}^L\mathbf{p}_I^\top{}^L_I\mathbf{R}\lfloor {}^I_G\mathbf{R}{}^G\mathbf{n}\rfloor{}^I\boldsymbol{\omega} - {}^G\mathbf{n}^\top{}^G\mathbf{v}_I \tag{52}$$

$$\mathbf{H}_{29} = \frac{1}{{}^Gd}\left({}^L\mathbf{p}_I^\top{}^L_I\mathbf{R}{}^I_G\mathbf{R} - {}^G\mathbf{p}_I^\top\right)\left(\mathbf{I}_3 - {}^G\mathbf{n}{}^G\mathbf{n}^\top\right) + {}^G\mathbf{n}^\top \tag{53}$$

2.3 Observability Analysis

Following the observability analysis in [17], we can construct the k-th block of the observability matrix as:

$$\begin{aligned}\mathbf{M}_k &= \mathbf{H}_{\mathbf{x}k}\mathbf{\Phi}_{(k,0)} \\ &= \mathbf{H}_\pi \begin{bmatrix} {}^L_I\mathbf{R}{}^I_G\hat{\mathbf{R}} & \mathbf{0}_{3\times1} \\ \mathbf{0}_{1\times3} & 1 \end{bmatrix} \begin{bmatrix} \Gamma_{\pi11} & \mathbf{0}_3 & \mathbf{0}_3 & \Gamma_{\pi14} & \mathbf{0}_3 & \Gamma_{\pi16} & \mathbf{0}_3 & \Gamma_{\pi18} & \Gamma_{\pi19} \\ \Gamma_{\pi21} & {}^G\mathbf{n}^\top & {}^G\mathbf{n}^\top\Delta t_k & \Gamma_{\pi24} & \Gamma_{\pi25} & \Gamma_{\pi26} & \Gamma_{\pi27} & \Gamma_{\pi28} & \Gamma_{\pi29} \end{bmatrix}\end{aligned} \tag{54}$$

where we have:

$$\Gamma_{11} = \mathbf{H}_{11}\mathbf{\Phi}_{I11} = \lfloor {}^G\mathbf{n} \rfloor {}^G_{I_0}\mathbf{R} \tag{55}$$

$$\Gamma_{14} = \mathbf{H}_{11}\mathbf{\Phi}_{I14} = -\lfloor {}^G\mathbf{n} \rfloor {}^G_{I_k}\mathbf{R}\mathbf{J}_r(t_k)\Delta t_k \tag{56}$$

$$\Gamma_{16} = \mathbf{H}_{16} = \lfloor {}^G\mathbf{n} \rfloor {}^G_{I_k}\mathbf{R}{}^I_L\mathbf{R} \tag{57}$$

$$\Gamma_{18} = \mathbf{H}_{18} = \lfloor {}^G\mathbf{n} \rfloor {}^G_{I_k}\mathbf{R}{}^{I_k}\boldsymbol{\omega} \tag{58}$$

$$\Gamma_{19} = \mathbf{H}_{19} = \frac{1}{{}^Gd}{}^L_I\mathbf{R}{}^{I_k}_G\mathbf{R}\left(\mathbf{I}_3 - {}^G\mathbf{n}{}^G\mathbf{n}^\top\right) \tag{59}$$

$$\Gamma_{21} = \mathbf{H}_{21}\mathbf{\Phi}_{I11} + {}^G\mathbf{n}^\top\mathbf{\Phi}_{I21} \tag{60}$$

$$= {}^L\mathbf{p}_I^\top {}^L_I\mathbf{R}\lfloor {}^{I_k}_G\mathbf{R}{}^G\mathbf{n} \rfloor {}^{I_k}_{I_0}\mathbf{R} - {}^G\mathbf{n}^\top \lfloor {}^G\mathbf{p}_{I_k} - {}^G\mathbf{p}_{I_0} - {}^G\mathbf{v}_{I_0}\Delta t_k + \frac{1}{2}{}^G\mathbf{g}\Delta t_k^2 \rfloor {}^G_{I_0}\mathbf{R} \tag{61}$$

$$\Gamma_{24} = \mathbf{H}_{21}\mathbf{\Phi}_{I14} + {}^G\mathbf{n}^\top\mathbf{\Phi}_{I24} = -{}^L\mathbf{p}_I^\top {}^L_I\mathbf{R}\lfloor {}^{I_k}_G\mathbf{R}{}^G\mathbf{n} \rfloor \mathbf{J}_r(t_k)\Delta t_k + {}^G\mathbf{n}^\top {}^G_{I_0}\mathbf{R}\Xi_4 \tag{62}$$

$$\Gamma_{25} = {}^G\mathbf{n}^\top\mathbf{\Phi}_{I25} = -{}^G\mathbf{n}^\top {}^G_{I_0}\mathbf{R}\Xi_2 \tag{63}$$

$$\Gamma_{26} = \mathbf{H}_{26} = {}^L\mathbf{p}_I^\top \lfloor {}^L_I\mathbf{R}{}^{I_k}_G\mathbf{R}{}^G\mathbf{n} \rfloor \tag{64}$$

$$\Gamma_{27} = \mathbf{H}_{27} = {}^G\mathbf{n}^\top {}^I_G\mathbf{R}^\top {}^L_I\mathbf{R}^\top \tag{65}$$

$$\Gamma_{28} = \mathbf{H}_{28} = {}^L\mathbf{p}_I^\top {}^L_I\mathbf{R}\lfloor {}^I_G\mathbf{R}{}^G\mathbf{n} \rfloor {}^I\boldsymbol{\omega} - {}^G\mathbf{n}^\top {}^G\mathbf{v}_I \tag{66}$$

$$\Gamma_{29} = \mathbf{H}_{29} = \frac{1}{{}^Gd}\left({}^L\mathbf{p}_I^\top {}^L_I\mathbf{R}{}^I_G\mathbf{R} - {}^G\mathbf{p}_I^\top\right)\left(\mathbf{I}_3 - {}^G\mathbf{n}{}^G\mathbf{n}^\top\right) + {}^G\mathbf{n}^\top \tag{67}$$

Lemma 1. *For LiDAR aided INS, if the state vector contains IMU state, spatial/temporal calibration between IMU-LiDAR and a plane feature, the system will have at least 7 unobservable directions as* $\mathbf{N}^{(\pi)}$.

$$\mathbf{N}^{(\pi)} = \begin{bmatrix} {}^{I_1}_G\hat{\mathbf{R}}{}^G\mathbf{g} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & {}^{I_1}_G\hat{\mathbf{R}}{}^G\hat{\mathbf{n}}_\pi \\ -\lfloor {}^G\hat{\mathbf{p}}_{I_1} \rfloor {}^G\mathbf{g} & {}^G\hat{\mathbf{n}}_1^\perp & {}^G\hat{\mathbf{n}}_2^\perp & {}^G\hat{\mathbf{n}}_\pi & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} \\ -\lfloor {}^G\hat{\mathbf{v}}_{I_1} \rfloor {}^G\mathbf{g} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & {}^G\hat{\mathbf{n}}_1^\perp & {}^G\hat{\mathbf{n}}_2^\perp & \mathbf{0}_{3\times1} \\ \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} \\ \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} \\ \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} \\ \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ -\lfloor {}^G\hat{d}_\pi {}^G\hat{\mathbf{n}}_\pi \rfloor {}^G\mathbf{g} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & {}^G\hat{\mathbf{n}}_\pi & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} \end{bmatrix} \tag{68}$$

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 build full rank columns and make both the spatial and temporal calibration between LiDAR-IMU observable.

2.4 Degenerate Cases Analysis for LiDAR-IMU Calibration

Given the LiDAR aided IMU system with plane features, the online calibration will suffer some degenerate cases that can cause the calibration parameters to be unobservable.

2.4.1 One-Plane Case

One-plane case refers to the cases there is only one plane or parallel planes in the state vector. We have identified the following degenerate motions for the LiDAR-IMU calibration:

- Pure Translation: if the system undergoes pure translation, the rigid transformation between LiDAR-IMU will be unobservable. The unobservable directions can be written as:

$$\mathbf{N}^{\pi}_{8:11} = \begin{bmatrix} \mathbf{0}_{15\times1} & \mathbf{0}_{15\times3} \\ {}^L_I\mathbf{R}^{I_1}_G\mathbf{R}^G\mathbf{n} & \mathbf{0}_3 \\ \mathbf{0}_{3\times1} & {}^L_I\mathbf{R}^{I_1}_G\mathbf{R}^G\mathbf{R}_\pi \\ 0 & 0 \\ \mathbf{0}_{3\times1} & \mathbf{e}_3^\top{}^G\mathbf{n} \end{bmatrix} \tag{69}$$

  where ${}^G\mathbf{R}_\pi = \begin{bmatrix} {}^G\mathbf{n}_1^\perp & {}^G\mathbf{n}_2^\perp & {}^G\mathbf{n} \end{bmatrix}$ denotes the plane orientation.

- One-axis rotation: if the system undergoes one-axis rotation, with fixed rotation axis as ${}^L\mathbf{k}$, the translation between LiDAR-IMU are not observable with unobservable directions as $\mathbf{N}^\pi_{12}$. Note that if the rotation axis is perpendicular to the plane direction, we will have an extra unobservable direction as $\mathbf{N}^\pi_{13}$.

$$\mathbf{N}^{\pi}_{12:13} = \begin{bmatrix} \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} \\ {}^G_{I_1}\mathbf{R}^I_L\mathbf{R}^L\mathbf{k} & \mathbf{0}_{3\times1} \\ \mathbf{0}_{12\times1} & \mathbf{0}_{12\times1} \\ {}^L\mathbf{k} & {}^L\mathbf{k} \\ \mathbf{0}_{4\times1} & \mathbf{0}_{4\times1} \end{bmatrix} \tag{70}$$

- Similar to IMU-CAM calibration, if the system undergoes motions with constant ${}^I\boldsymbol{\omega}\&{}^I\mathbf{v}$ or constant ${}^I\boldsymbol{\omega}\&{}^G\mathbf{a}$, the IMU-LiDAR temporal calibration will also be unobservable with unobservable directions as $\mathbf{N}^\pi_{14}$ or $\mathbf{N}^\pi_{15}$, respectively. In addition, for one-plane case, we have an extra degenerate motion (${}^G\boldsymbol{\omega} \parallel {}^G\mathbf{n}$ and ${}^G\mathbf{n} \perp {}^G\mathbf{v}_I$[2]) for time offset as $\mathbf{N}^\pi_{16}$.

$$\mathbf{N}^{(\pi)}_{14:16} = \begin{bmatrix} \mathbf{0}_{6\times1} & \mathbf{0}_{6\times1} & \mathbf{0}_{6\times1} \\ \mathbf{0}_{3\times1} & {}^G\mathbf{a}_I & \mathbf{0}_{3\times1} \\ \mathbf{0}_{6\times1} & \mathbf{0}_{6\times1} & \mathbf{0}_{6\times1} \\ {}^L_I\mathbf{R}^I\boldsymbol{\omega} & {}^L_I\mathbf{R}^I\boldsymbol{\omega} & \mathbf{0}_{3\times1} \\ -{}^L_I\mathbf{R}^I\mathbf{v} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} \\ -1 & -1 & 1 \\ \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} & \mathbf{0}_{3\times1} \end{bmatrix} \tag{71}$$

[2] " $\parallel$ " and " $\perp$ " denote parallel and perpendicular relationship, respectively.

2.4.2 Two-Plane Case

Two-Plane case refers to the cases that there are two intersected planes or all the planes have parallel intersections in the state vector. We first modify our state vector as:

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_I^\top & \mathbf{x}_{calib_L}^\top & {}^G\mathbf{p}_{\pi 1}^\top & {}^G\mathbf{p}_{\pi 2}^\top \end{bmatrix}^\top \tag{72}$$

We have identified the following degenerate motions for the LiDAR-IMU calibration:

- Pure translation: if the system undergoes pure translation, the translation of rigid transformation between LiDAR-IMU will be unobservable. The unobservable directions can be written as:

$$\mathbf{N}^{\pi}_{17:19} = \begin{bmatrix} \mathbf{0}_{15\times 3} \\ \mathbf{0}_3 \\ {}^L_I\mathbf{R}^{I_1}_G\mathbf{R} \\ 0 \\ \mathbf{e}_3^\top {}^G\mathbf{n}_{\pi 1} {}^G\mathbf{R}_{\pi 1}^\top \\ \mathbf{e}_3^\top {}^G\mathbf{n}_{\pi 2} {}^G\mathbf{R}_{\pi 2}^\top \end{bmatrix} \tag{73}$$

  where ${}^G\mathbf{R}_{\pi i} = \begin{bmatrix} {}^G\mathbf{n}_{\pi i1}^\perp & {}^G\mathbf{n}_{\pi i2}^\perp & {}^G\mathbf{n}_{\pi i} \end{bmatrix}, i = \{1, 2\}$ denotes the plane orientation.

- One-axis rotation: if the system undergoes one-axis rotation, with fixed rotation axis as ${}^L\mathbf{k}$, the translation between LiDAR-IMU are not observable with unobservable directions as $\mathbf{N}^{\pi}_{20}$. Note that if the rotation axis is parallel to both planes' intersection direction (${}^G_{I_1}\mathbf{R}^I_L\mathbf{R}^L\mathbf{k} \parallel \lfloor {}^G\mathbf{n}_{\pi 1} \rfloor {}^G\mathbf{n}_{\pi 2}$), we will have an extra unobservable direction as $\mathbf{N}^{\pi}_{21}$.

$$\mathbf{N}^{\pi}_{20:21} = \begin{bmatrix} \mathbf{0}_{3\times 1} & \mathbf{0}_{3\times 1} \\ {}^G_{I_1}\mathbf{R}^I_L\mathbf{R}^L\mathbf{k} & \mathbf{0}_{3\times 1} \\ \mathbf{0}_{12\times 1} & \mathbf{0}_{12\times 1} \\ {}^L\mathbf{k} & {}^L\mathbf{k} \\ \mathbf{0}_{7\times 1} & \mathbf{0}_{7\times 1} \end{bmatrix} \tag{74}$$

- If the system undergoes motions with constant ${}^I\boldsymbol{\omega}\&{}^I\mathbf{v}$ or constant ${}^I\boldsymbol{\omega}\&{}^G\mathbf{a}$, the IMU-LiDAR temporal calibration will also be unobservable with unobservable directions as $\mathbf{N}^{\pi}_{22}$ or $\mathbf{N}^{\pi}_{23}$, respectively. In addition, for two-plane case, we have an extra degenerate motion (constant ${}^I\boldsymbol{\omega}$ and ${}^G\mathbf{v}_I \parallel \lfloor {}^G\mathbf{n}_{\pi 1} \rfloor {}^G\mathbf{n}_{\pi 2}$) for time offset as $\mathbf{N}^{\pi}_{24}$.

$$\mathbf{N}^{(\pi)}_{22:24} = \begin{bmatrix} \mathbf{0}_{6\times 1} & \mathbf{0}_{6\times 1} & \mathbf{0}_{6\times 1} \\ \mathbf{0}_{3\times 1} & {}^G\mathbf{a}_I & \mathbf{0}_{3\times 1} \\ \mathbf{0}_{6\times 1} & \mathbf{0}_{6\times 1} & \mathbf{0}_{6\times 1} \\ {}^L_I\mathbf{R}^I\boldsymbol{\omega} & {}^L_I\mathbf{R}^I\boldsymbol{\omega} & {}^L_I\mathbf{R}^I\boldsymbol{\omega} \\ -{}^L_I\mathbf{R}^I\mathbf{v} & \mathbf{0}_{3\times 1} & \mathbf{0}_{3\times 1} \\ -1 & -1 & -1 \\ \mathbf{0}_{6\times 1} & \mathbf{0}_{6\times 1} & \mathbf{0}_{6\times 1} \end{bmatrix} \tag{75}$$

2.4.3 Multiple-Plane Case

Multiple-Plane case refers to the cases that there are at least three intersecting planes with unparalleled plane intersections in the state vector. We first modify our state vector as:

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_I^\top & \mathbf{x}_{calib_L}^\top & {}^G\mathbf{p}_{\pi 1}^\top & {}^G\mathbf{p}_{\pi 2}^\top & {}^G\mathbf{p}_{\pi 3}^\top \end{bmatrix}^\top \tag{76}$$

We have identified the following degenerate motions for the LiDAR-IMU calibration:

- Pure translation: if the system undergoes pure translation, the translation of rigid transformation between LiDAR-IMU will be unobservable. The unobservable directions can be written as:

$$\mathbf{N}_{25:27}^{\pi} = \begin{bmatrix} \mathbf{0}_{15\times 3} \\ \mathbf{0}_3 \\ {}_I^L\mathbf{R}_G^{I_1}\mathbf{R} \\ 0 \\ \mathbf{e}_3^\top {}^G\mathbf{n}_{\pi 1} {}^G\mathbf{R}_{\pi 1}^\top \\ \mathbf{e}_3^\top {}^G\mathbf{n}_{\pi 2} {}^G\mathbf{R}_{\pi 2}^\top \\ \mathbf{e}_3^\top {}^G\mathbf{n}_{\pi 3} {}^G\mathbf{R}_{\pi 3}^\top \end{bmatrix} \tag{77}$$

  where ${}^G\mathbf{R}_{\pi i} = \begin{bmatrix} {}^G\mathbf{n}_{\pi i1}^\perp & {}^G\mathbf{n}_{\pi i2}^\perp & {}^G\mathbf{n}_{\pi i} \end{bmatrix}, i = \{1, 2, 3\}$ denotes the plane orientation.

- One-axis rotation: if the system undergoes one-axis rotation, with fixed rotation axis as ${}^L\mathbf{k}$, the translation between LiDAR-IMU are not observable with unobservable directions as $\mathbf{N}_{28}^{\pi}$.

$$\mathbf{N}_{28}^{\pi} = \begin{bmatrix} \mathbf{0}_{3\times 1} \\ {}_{I_1}^G\mathbf{R}_L^I\mathbf{R}^L\mathbf{k} \\ \mathbf{0}_{12\times 1} \\ {}^L\mathbf{k} \\ \mathbf{0}_{10\times 1} \end{bmatrix} \tag{78}$$

- If the system undergoes motions with constant ${}^I\boldsymbol{\omega}\&{}^I\mathbf{v}$ or constant ${}^I\boldsymbol{\omega}\&{}^G\mathbf{a}$, the IMU-LiDAR temporal calibration will also be unobservable with unobservable directions as $\mathbf{N}_{29}^{\pi}$ or $\mathbf{N}_{30}^{\pi}$, respectively.

$$\mathbf{N}_{29:30}^{(\pi)} = \begin{bmatrix} \mathbf{0}_{6\times 1} & \mathbf{0}_{6\times 1} \\ \mathbf{0}_{3\times 1} & {}^G\mathbf{a}_I \\ \mathbf{0}_{6\times 1} & \mathbf{0}_{6\times 1} \\ {}_I^L\mathbf{R}^I\boldsymbol{\omega} & {}_I^L\mathbf{R}^I\boldsymbol{\omega} \\ -{}_I^L\mathbf{R}^I\mathbf{v} & \mathbf{0}_{3\times 1} \\ -1 & -1 \\ \mathbf{0}_{9\times 1} & \mathbf{0}_{9\times 1} \end{bmatrix} \tag{79}$$

Table 1: Summary of Degenerate Motions for VINS Calibration

One Plane / Parallel Planes	Unobservable
Pure Translation	${}^{L}_{I}\mathbf{R}$, ${}^{L}\mathbf{p}_I$
1-axis Rotation	${}^{L}\mathbf{p}_I$
Constant ${}^{I}\omega$ and ${}^{I}\mathbf{v}$	t_{dI}, ${}^{L}\mathbf{p}_I$
Constant ${}^{I}\omega$ and ${}^{G}\mathbf{a}$	t_{dI}, ${}^{L}\mathbf{p}_I$
${}^{G}\boldsymbol{\omega} \parallel {}^{G}\mathbf{n}$ and ${}^{G}\mathbf{n} \perp {}^{G}\mathbf{v}_I$	t_{dI}
Two Planes with Intersection	**Unobservable**
Pure Translation	${}^{L}\mathbf{p}_I$
1-axis Rotation	${}^{L}\mathbf{p}_I$
Constant ${}^{I}\omega$ and ${}^{I}\mathbf{v}$	t_{dI}, ${}^{L}\mathbf{p}_I$
Constant ${}^{I}\omega$ and ${}^{G}\mathbf{a}$	t_{dI}, ${}^{L}\mathbf{p}_I$
Constant ${}^{I}\boldsymbol{\omega}$ and ${}^{G}\mathbf{v}_I \parallel \lfloor {}^{G}\mathbf{n}_{\pi 1} \rfloor {}^{G}\mathbf{n}_{\pi 2}$	t_{dI}
Multiple Planes	**Unobservable**
Pure Translation	${}^{L}\mathbf{p}_I$
1-axis Rotation	${}^{L}\mathbf{p}_I$
Constant ${}^{I}\omega$ and ${}^{I}\mathbf{v}$	t_{dI}, ${}^{L}\mathbf{p}_I$
Constant ${}^{I}\omega$ and ${}^{G}\mathbf{a}$	t_{dI}, ${}^{L}\mathbf{p}_I$

References

[1] 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*. Macau, China, Nov. 2019.

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

[3] M. Li and A. I. Mourikis. "Online temporal calibration for camera–IMU systems: Theory and algorithms". In: *The International Journal of Robotics Research* 33.7 (2014), pp. 947–964.

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

[5] 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 (Dec. 2019), pp. 399–1418.

[6] A. I. Mourikis and S. I. Roumeliotis. "A multi-state constraint Kalman filter for vision-aided inertial navigation". In: *Proceedings 2007 IEEE International Conference on Robotics and Automation*. IEEE. 2007, pp. 3565–3572.

[7] 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.

[8] 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*. Paris, France, 2020.

[9] X. Zuo, Y. yang, P. Geneva, J. Lv, Y. Liu, G. Huang, and M. Pollefeys. "LIC-Fusion 2.0: LiDAR-Inertial-Camera Odometry with Sliding-Window Plane-Feature Tracking". In: *A Submission of Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2020, p. 2020.

[10] 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*. Vancouver, Canada, 2017, pp. 6749–6755.

[11] Y. Yang and G. Huang. "Aided Inertial Navigation with Geometric Features: Observability Analysis". In: *Proc. of the IEEE International Conference on Robotics and Automation*. Brisbane, Australia, 2018.

[12] M. Li. "Visual-inertial odometry on resource-constrained systems". PhD thesis. UC Riverside, 2014.

[13] 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.

[14] Y. Yang, P. Geneva, K. Eckenhoff, and G. Huang. “Degenerate Motion Analysis for Aided INS with Online Spatial and Temporal Calibration”. In: *IEEE Robotics and Automation Letters (RA-L)* 4.2 (2019), pp. 2070–2077.

[15] 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.

[16] Y. Yang, B. P. W. Babu, C. Chen, G. Huang, and L. Ren. “Analytic Combined IMU Integrator for Visual-Inertial Navigation”. In: *Proc. of the IEEE International Conference on Robotics and Automation*. Paris, France, 2020.

[17] J. Hesch, D. Kottas, S. Bowman, and S. Roumeliotis. “Consistency Analysis and Improvement of Vision-aided Inertial Navigation”. In: *IEEE Transactions on Robotics* PP.99 (2013), pp. 1–19.