# LIPS: LiDAR-Inertial 3D Plane SLAM

Patrick Geneva\*, Kevin Eckenhoff†, Yulin Yang†, and Guoquan Huang†

*Abstract*— This paper presents the formalization of the closest point plane representation and an analysis of its incorporation in 3D indoor simultaneous localization and mapping (SLAM). We present a singularity free plane factor leveraging the closest point plane representation, and demonstrate its fusion with inertial preintegration measurements in a graph-based optimization framework. The resulting LiDAR-inertial 3D plane SLAM (LIPS) system is validated both on a custom made LiDAR simulator and on a real-world experiment.

## I. INTRODUCTION

Accurate and robust indoor localization and mapping is a fundamental requirement for many applications of autonomous robots. Indoor environments are typically rich in structural information, such as lines and planes, that should be exploited to achieve high-accuracy simultaneous localization and mapping (SLAM). Although indoor environments prevent the use of GPS for localization, inertial navigation systems (INS) aided with exteroceptive sensors such as cameras [1, 2], light detection and ranging (LiDAR) sensors [3, 4], and even sonars [5], have shown to be effective.

Recently LiDAR sensing technology has reduced in weight and size, allowing for portable and handheld use, delivering up to 2.2 million data points per second of the surrounding environment. LiDARs have proven to be effective in textureless and low-light environments, typical of normal office and building interiors, providing measurements with high signal-to-noise ratios (SNR). Fundamentally, LiDAR sensors do not rely on the lighting or texture properties that other sensors such as cameras require. One of the challenging aspects of LiDAR sensors is how to process their large amount of unordered 3D point data for inclusion in estimation. One of the conventional ways is to use an iterative closest point (ICP) solver to find the relative transformation between poses. Using ICP to only recover the relative poses, prevents the inclusion of highly informative environmental primitives, such as planes, in state estimation.

The first challenge of using planar primitives for estimation is their parameterization [6]. The most common representation is that of the plane's normal direction vector and a distance scalar, which is known as the *Hesse form*. Since the Hesse form is an over-parameterization, it would suffer from a singular information matrix in least-squares

optimization if no extra care is taken. To avoid this over-parameterization, spherical coordinates containing 2 angles (horizontal and vertical angle) and a distance scalar are commonly used as its error state representation. Although this is minimal, it suffers from ambiguities when the vertical angle equals $\pm\frac{\pi}{2}$. More recently, Kaess [6] proposed the use of a unit quaternion and its singularity free 3 degree-of-freedom (DOF) multiplicative error state, whose geometric connection to the plane is unclear, and whose numerical stability may not be optimal. By contrast, in this work, we advocate the use of the *closest point* (CP) representation which is defined by the closest point on the plane to the origin of a given frame of reference. The CP representation not only captures all geometric plane information, but also minimally represents the plane with a simple additive error state operation, yielding numerical advantages.

Specifically, leveraging the CP representation for planes, we fuse plane primitive measurements from a 3D LiDAR and motion information from an inertial measurement unit (IMU) (in the form of continuous IMU preintegration [7, 8]) through graph optimization for 3D indoor SLAM. The key contributions of this paper are the following:

- Formulation of the closest point (CP) plane representation, analysis of its singularities, and its use as a plane representation and error state in 3D plane SLAM.
- Design of a novel LiDAR-Inertial 3D Plane SLAM (LIPS) system with a robust relative plane anchor factor for graph-based optimization, effectively overcoming the singularity issue of the CP representation.
- Development of a generalized LiDAR simulator for evaluating LiDAR-aided localization algorithms, which is open sourced to better benefit the community.[1]
- Validation of the proposed LIPS system through Monte-Carlo simulations and a real-world experiment.

The rest of the paper is structured as follows: After reviewing the related work in the next section, we formulate the LiDAR-inertial plane SLAM (LIPS) problem in Section III, and describe the continuous IMU preintegration in Section IV. In Section V we present in detail how to determine the plane factors that are used in graph-based optimization. In Section VI we validate the proposed LIPS system on synthetic and real experiments. Finally, the paper concludes in Section VII along with possible future research.

## II. RELATED WORK

Over the past three decades, we have witnessed tremendous progress in SLAM and a multitude of different algo-

\*Dept. of Computer and Information Sciences, University of Delaware, Newark, DE 19716, USA. Email: pgeneva@udel.edu
†Dept. of Mechanical Engineering, University of Delaware, Newark, DE 19716, USA. Email: {keck, yuyang, ghuang}@udel.edu

[1]https://github.com/rpng/lips

rithms have been developed [9]. In what follows, we only review the planar SLAM literature closely related to our work.

## A. SLAM with Planar Primitives

Within the literature involving range based sensors, planar features have shown to improve point cloud registration accuracy [10–13]. In the field of LiDAR based odometry, a state-of-the-art method is LOAM [14], which sequentially registers extracted planar and edge features to an incrementally built global map, and demonstrates impressive efficiency and accuracy. Proencca et al. [15] recently presented a planar odometry method that used a modified Hesse plane representation but during plane matching they defined the cost function as the difference between two points residing on the planes. While this was not used as a planar representation, we will more formally present it as the "closest point" representation in Section V-A. That being said, instead of solving the conventional point cloud registration problem (e.g., iterative closest point (ICP)), we seek to solve the SLAM problem (i.e., to probabilistically estimate both the historical robot poses and environment structure) with a high level geometric primitive: plane features.

One of the first uses of planes in SLAM was by Weingarten et al. [16] who extracted planes from incoming point clouds through a breadth-first region growing algorithm and fused similar planes using a Mahalanobis-distance test. Pathak et al. [17] extended this work by presenting a plane correspondence algorithm that maximized geometric scene consistency and allowed for realtime performance.

Trevor et al. [18] combined lines, planes, and odometry measurements in a graph-based framework, but used the overparametrized Hesse form during optimization. Taguchi et al. [19] introduced a handheld RGBD point-plane SLAM system, provides analysis of degeneracy issues, and presented a RANSAC based approach to the feature correspondence problem. Salas et al. [20] leveraged the creation of a dense planar map to allow for simple localization of incoming RGBD sensor readings through the direct projection of the dense planar map into the camera frame. Kaess [6] presented the unit quaternion plane representation, proposed a relative plane formulation for improved convergence in batch optimization, and demonstrated simple planar 3D mapping with a handheld RGBD sensor. Based on this work, Hsiao et al. [21, 22] performed keyframe-based dense planar SLAM and achieved higher estimation accuracy due to the additional plane constraints and recently incorporated discrete inertial preintegration.

More recently, Zhang et al. [23] introduced a fast plane segmentation and map refinement step that improved real-time performance and constructed map quality. Ma et al. [24] used a RGBD camera to perform direct alignment to planar keyframes and optimized a global graph using an expectation-maximization framework. All these works have tried to solve the same basic question: *How does one optimally leverage high level plane primitives in SLAM?* In this work, we propose the use of the CP representation, and combine inertial information provided by an IMU to perform robust 3D SLAM.

## B. Inertial-aided Planar SLAM

Conventionally, IMUs have been used to provide 3D pose predictions for LiDAR registration methods. We instead look to tightly fuse inertial and planar primitive measurements in our state estimation. Hesch et al. [3] first proposed a LiDAR-aided inertial EKF that used a 2D LiDAR for indoor mapping. Unlike the proposed method, they assumed that all planes extracted were orthogonal in the environment and also performed frequent stops to prevent drift in the unobservable z-direction. Additionally, since plane orientations were assumed orthogonal to each other and known, only the distance to the plane was estimated. Guo et al. [25] investigated the observability properties of an IMU-RGBD system and constructed an observability-constrained Kalman filter. They leveraged both point and planar features and only used the plane orientation as a measurement losing the information captured in the distance to the plane while avoiding singularity issues. We look to incorporate the inertial information in a graph-based system, allowing for loop closures over long distances. The use of planar primitives for loop closures is desirable due to the large size and abundance of planar features in the environment. We leverage the inertial measurements to better constrain the low frequency LiDAR sensor in highly dynamic scenarios. Additionally, unlike these prior works, we incorporate the full 3 DOF plane measurements in estimation without prior assumptions.

## III. PROBLEM FORMULATION

Graph-based SLAM has prevailed because of its ability to relinearize all nonlinear measurements and thus provide accurate results [26, 27]. We therefore employ such formulation for our proposed LiDAR-inertial plane SLAM (LIPS).

### A. Graph-based Optimization

As a robot follows a trajectory it collects a set of measurements, $\mathbf{z}$, of the environment around it through sensors such as IMUs, LiDARs, and RGBD cameras. These measurements either directly or indirectly measure the underlying state, $\mathbf{x}$, of the robot. Formulated as a graph-based optimization problem, we represent historical robot states, $\mathbf{x}_i$, as nodes with measurements, $\mathbf{z}_i$, as edges/factors connected to their involved nodes. Under the assumption of independent Gaussian noise corruption on our measurements, we formulate the Maximum Likelihood Estimation (MLE) problem as the following nonlinear least-squares problem [27, 28]:

$$\hat{\mathbf{x}} = \operatorname*{argmin}_{\mathbf{x}} \sum_i ||\mathbf{r}_i(\mathbf{x})||^2_{\mathbf{P}_i} \qquad (1)$$

where $\mathbf{r}_i$ is the zero-mean residual associated with measurement $\mathbf{z}_i$, $\mathbf{P}_i$ is the measurement covariance, and $||\mathbf{v}||^2_{\mathbf{P}} = \mathbf{v}^\top \mathbf{P}^{-1} \mathbf{v}$ is the energy norm. This problem can be solved iteratively by linearizing about the current estimate, $\hat{\mathbf{x}}$, and
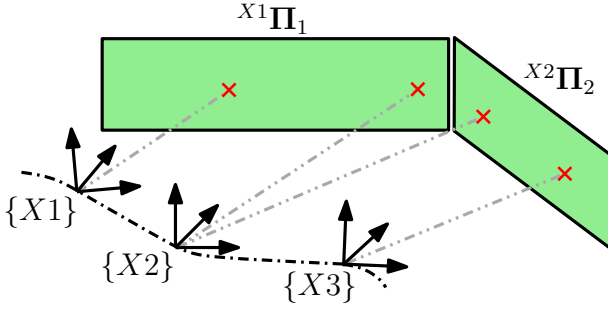
Fig. 1: Pictorial view an example trajectory and measurements that are included in the proposed LIPS optimization. Two planes are being estimated in different anchor poses. In the case of $^{X1}\mathbf{\Pi}_1$, it was first seen from the $X1$ state so it will be estimated in the $\{X1\}$ frame of reference while $^{X2}\mathbf{\Pi}_2$ will be estimated in the $\{X2\}$ frame of reference.

defining a new optimization problem in terms of the *error state*, $\widetilde{\mathbf{x}}$:

$$\widetilde{\mathbf{x}}^* = \underset{\widetilde{\mathbf{x}}}{\operatorname{argmin}} \sum_i ||\mathbf{r}_i(\hat{\mathbf{x}}) + \mathbf{J}_i\widetilde{\mathbf{x}}||^2_{\mathbf{P}_i} \tag{2}$$

where $\mathbf{J}_i = \frac{\partial \mathbf{r}_i(\hat{\mathbf{x}} \boxplus \check{\mathbf{x}})}{\partial \check{\mathbf{x}}}$ is the Jacobian of $i$-th residual with respect to the error state. We define the generalized update operation, $\boxplus$, which maps the error state to the full state. After solving the linearized system, the current state estimate is updated as $\mathbf{x} = \hat{\mathbf{x}} \boxplus \widetilde{\mathbf{x}}$.

### B. The Proposed LIPS System

In the proposed LIPS, we define the robot state at each time as the following $16 \times 1$ vector:

$$\mathbf{x}_I = \begin{bmatrix} I_G\bar{q}^\top & \mathbf{b}_g^\top & {}^G\mathbf{v}_I^\top & \mathbf{b}_a^\top & {}^G\mathbf{p}_I^\top \end{bmatrix}^\top \tag{3}$$

where the quaternion $^I_G\bar{q}$ represents the rotation, $^I_G\mathbf{R}$, from global frame $\{G\}$ to the IMU frame $\{I\}$, the velocity $^G\mathbf{v}_I$ is of the IMU seen from the global frame, position $^G\mathbf{p}_I$ is the IMU position seen in the global frame, and $\mathbf{b}_g$ and $\mathbf{b}_a$ are the gyroscope and accelerometer biases respectively. We can define the minimal representation error state as:

$$\widetilde{\mathbf{x}}_I = \begin{bmatrix} I_G\delta\boldsymbol{\theta}^\top & \widetilde{\mathbf{b}}_g^\top & {}^G\widetilde{\mathbf{v}}_I^\top & \widetilde{\mathbf{b}}_a^\top & {}^G\widetilde{\mathbf{p}}_I^\top \end{bmatrix}^\top \tag{4}$$

The position, velocity, and biases can all be updated using the conventional additive error model (i.e., $^G\widetilde{\mathbf{p}}_I = {}^G\mathbf{p}_I - {}^G\hat{\mathbf{p}}_I$). Special care needs to be taken to use a multiplicative error model for the quaternion defined by:

$$\delta\bar{q} = \bar{q} \otimes \hat{\bar{q}}^{-1} \simeq \begin{bmatrix} \frac{1}{2}\delta\boldsymbol{\theta}^\top & 1 \end{bmatrix}^\top \tag{5}$$

where $\otimes$ is the quaternion multiplication operator [29].

The total state to be estimated, $\mathbf{x}$, consists of the $m$ historical IMU states and $k$ planar primitives:

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_{I(1)} & \cdots & \mathbf{x}_{I(m)} & \mathbf{\Pi}_1 & \cdots & \mathbf{\Pi}_k \end{bmatrix} \tag{6}$$

where $\mathbf{x}_{I(i)}$ is the state of the IMU at timestep $i$ and $\mathbf{\Pi}_j$ refers to the $j$th plane. To perform this estimation, we first determine the continuous IMU preintegration factors (see Section IV) and 3D plane factors from LiDAR measurements (see Section V), thereby building a factor graph for optimization. In particular, planes are first extracted from the

point cloud and then compressed into the CP representation (see Section V-C). Figure 1 illustrates the overall LIPS system. These measurements are added to the graph and optimized using the iSAM2 [30] implementation available in the GTSAM [31] nonlinear optimization library. Note that, while not used in our small-scale experiments (see Section VI-D), an advantage of the relative information provided by the IMU preintegration is the ability to perform LiDAR cloud unwarping during high speed maneuvers. The final cost function of the LIPS system can be described as:

$$\hat{\mathbf{x}} = \underset{\mathbf{x}}{\operatorname{argmin}} \left[ \sum_i ||\mathbf{r}_{Ii}(\mathbf{x})||^2_{\mathbf{R}_{Ii}} + \sum_j ||\mathbf{r}_{\Pi j}(\mathbf{x})||^2_{\mathbf{R}_{\Pi j}} \right] \tag{7}$$

where $\mathbf{r}_{Ii}(\mathbf{x})$ and $\mathbf{r}_{\Pi j}(\mathbf{x})$ are the zero mean residuals associated with the continuous preintegration and anchored CP planes measurements, respectively. While, $\mathbf{R}_{Ii}$ and $\mathbf{R}_{\Pi j}$ are the covariances of the continuous preintegration and anchor CP plane measurements, respectively.

### IV. CONTINUOUS IMU PREINTEGRATION FACTOR

Low cost MEMS based IMU sensors have seen great popularity as they provide direct readings of the evolution of the robot's state and provide robustness in dynamic situations which other exteroceptive sensors alone might fail. To provide information between the low frequency LiDAR sensor readings, we leverage our previous work in continuous preintegration [7, 8] which allows for large amounts of IMU readings within the interval to be compounded into a single highly informative constraint. Continuous preintegration is based on *closed-form* expressions of the IMU measurement dynamics rather than the discrete approximations used in previous works [32]. We model the linear acceleration and angular velocity inertial measurements as:

$$\boldsymbol{\omega}_m = \boldsymbol{\omega} + \mathbf{b}_w + \mathbf{n}_w \tag{8}$$

$$\mathbf{a}_m = \mathbf{a} + \mathbf{b}_a + \mathbf{n}_a + {}^I_G\mathbf{R}^G\mathbf{g} \tag{9}$$

where $^G\mathbf{g}$ is the gravity in the global frame, $\boldsymbol{\omega}$ is the angular velocity, $\mathbf{a}$ is the linear acceleration, and $\mathbf{n}_w$, $\mathbf{n}_a$ are the continuous measurement noises. The underlying standard IMU dynamics are given by [33]:

$$^I_G\dot{\bar{q}} = \frac{1}{2}\boldsymbol{\Omega}(\boldsymbol{\omega}_m - \mathbf{b}_w - \mathbf{n}_w)^I_G\bar{q} \tag{10}$$

$$\dot{\mathbf{b}}_w = \mathbf{n}_{wb} \tag{11}$$

$$^G\dot{\mathbf{v}}_k = {}^G_k\mathbf{R}(\mathbf{a}_m - \mathbf{b}_a - \mathbf{n}_a) - {}^G\mathbf{g} \tag{12}$$

$$\dot{\mathbf{b}}_a = \mathbf{n}_{ab} \tag{13}$$

$$^G\dot{\mathbf{p}}_k = {}^G\mathbf{v}_k \tag{14}$$

where $\mathbf{n}_{wb}$, $\mathbf{n}_{ab}$ are the random walk noises and $\boldsymbol{\Omega}(\cdot)$ is:

$$\boldsymbol{\Omega}(\boldsymbol{\omega}) = \begin{bmatrix} -\lfloor\boldsymbol{\omega}\times\rfloor & \boldsymbol{\omega} \\ -\boldsymbol{\omega}^\top & 0 \end{bmatrix} \tag{15}$$

The key idea of preintegration is to factorize the resulting integration of equations (10)-(14) between two LiDAR

timesteps [32]:

$$^G\mathbf{p}_{k+1} = {}^G\mathbf{p}_k + {}^G\mathbf{v}_k\Delta T - \frac{1}{2}{}^G\mathbf{g}\Delta T^2 + {}_k^G\mathbf{R}^k\boldsymbol{\alpha}_{k+1} \quad (16)$$

$$^G\mathbf{v}_{k+1} = {}^G\mathbf{v}_k - {}^G\mathbf{g}\Delta T + {}_k^G\mathbf{R}^k\boldsymbol{\beta}_{k+1} \quad (17)$$

$$_G^{k+1}\bar{q} = {}_k^{k+1}\bar{q} \otimes {}_G^k\bar{q} \quad (18)$$

where $\Delta T$ is the difference between the bounding LiDAR pose timestamps $(t_k, t_{k+1})$ and $^k\boldsymbol{\alpha}_{k+1}$, $^k\boldsymbol{\beta}_{k+1}$ are defined by the following integrations of the IMU measurements:

$$^k\boldsymbol{\alpha}_{k+1} = \int_{t_k}^{t_{k+1}} \int_{t_k}^{s} {}_u^k\mathbf{R}\left(\mathbf{a}_m - \mathbf{b}_a - \mathbf{n}_a\right) du\,ds \quad (19)$$

$$^k\boldsymbol{\beta}_{k+1} = \int_{t_k}^{t_{k+1}} {}_u^k\mathbf{R}\left(\mathbf{a}_m - \mathbf{b}_a - \mathbf{n}_a\right) du \quad (20)$$

We note that the preintegrated measurements, $^k\boldsymbol{\alpha}_{k+1}$, $^k\boldsymbol{\beta}_{k+1}$, $_k^{k+1}\bar{q}$ are dependent on the *true* biases. This dependency is addressed through a first order Taylor series expansion about the current bias estimates $\bar{\mathbf{b}}_w$ and $\bar{\mathbf{b}}_a$ at time $t_k$:

$$^k\boldsymbol{\alpha}_{k+1} \simeq {}^k\breve{\boldsymbol{\alpha}}_{k+1} + \left.\frac{\partial\boldsymbol{\alpha}}{\partial\mathbf{b}_a}\right|_{\bar{\mathbf{b}}_a}\Delta\mathbf{b}_a + \left.\frac{\partial\boldsymbol{\alpha}}{\partial\mathbf{b}_w}\right|_{\bar{\mathbf{b}}_w}\Delta\mathbf{b}_w \quad (21)$$

$$^k\boldsymbol{\beta}_{k+1} \simeq {}^k\breve{\boldsymbol{\beta}}_{k+1} + \left.\frac{\partial\boldsymbol{\beta}}{\partial\mathbf{b}_a}\right|_{\bar{\mathbf{b}}_a}\Delta\mathbf{b}_a + \left.\frac{\partial\boldsymbol{\beta}}{\partial\mathbf{b}_w}\right|_{\bar{\mathbf{b}}_w}\Delta\mathbf{b}_w \quad (22)$$

$$_k^{k+1}\bar{q} \simeq \bar{q}(\Delta\mathbf{b}_w)^{-1} \otimes {}_k^{k+1}\breve{q} \quad (23)$$

where $^k\breve{\boldsymbol{\alpha}}_{k+1}$, $^k\breve{\boldsymbol{\beta}}_{k+1}$, $_k^{k+1}\breve{q}$ are the preintegrated measurements evaluated at the current bias estimates. In particular, $_k^{k+1}\breve{q}$ can be found using the zeroth order quaternion integrator [29]. We define the quaternion which models multiplicative orientation corrections due to changes in the linearized bias as:

$$\bar{q}(\Delta\mathbf{b}_w) = \begin{bmatrix} \frac{\boldsymbol{\theta}}{||\boldsymbol{\theta}||}\sin\frac{||\boldsymbol{\theta}||}{2} \\ \cos\frac{||\boldsymbol{\theta}||}{2} \end{bmatrix}, \quad \boldsymbol{\theta} = \left.\frac{\partial\bar{q}}{\partial\mathbf{b}_w}\right|_{\bar{\mathbf{b}}_w}\left(\mathbf{b}_{w(k)} - \bar{\mathbf{b}}_w\right)$$

where $\Delta\mathbf{b}_w := \mathbf{b}_{w(k)} - \bar{\mathbf{b}}_w$ and $\Delta\mathbf{b}_a := \mathbf{b}_{a(k)} - \bar{\mathbf{b}}_a$ are the differences between the true biases and the bias estimate used as the linearization point. The new preintegration measurements can now be computed *once* and changes in the bias estimates can be taken into account through the above Taylor series. The final measurement residual is as follows:

$$r_I(\mathbf{x}) = \begin{bmatrix} 2\text{vec}\left({}_G^{k+1}\bar{q} \otimes {}_G^k\bar{q}^{-1} \otimes {}_k^{k+1}\breve{q}^{-1} \otimes \bar{q}(\Delta\mathbf{b}_w)\right) \\ \mathbf{b}_{w(k+1)} - \mathbf{b}_{w(k)} \\ \begin{pmatrix} {}_G^k\mathbf{R}\left({}^G\mathbf{v}_{k+1} - {}^G\mathbf{v}_k + {}^G\mathbf{g}\Delta T\right) \\ -{}^k\breve{\boldsymbol{\beta}}_{k+1} - \left.\frac{\partial\boldsymbol{\beta}}{\partial\mathbf{b}_a}\right|_{\bar{\mathbf{b}}_a}\Delta\mathbf{b}_a - \left.\frac{\partial\boldsymbol{\beta}}{\partial\mathbf{b}_w}\right|_{\bar{\mathbf{b}}_w}\Delta\mathbf{b}_w \end{pmatrix} \\ \mathbf{b}_{a(k+1)} - \mathbf{b}_{a(k)} \\ \begin{pmatrix} {}_G^k\mathbf{R}\left({}^G\mathbf{p}_{k+1} - {}^G\mathbf{p}_k - {}^G\mathbf{v}_k\Delta T + \frac{1}{2}{}^G\mathbf{g}\Delta T^2\right) \\ -{}^k\breve{\boldsymbol{\alpha}}_{k+1} - \left.\frac{\partial\boldsymbol{\alpha}}{\partial\mathbf{b}_a}\right|_{\bar{\mathbf{b}}_a}\Delta\mathbf{b}_a - \left.\frac{\partial\boldsymbol{\alpha}}{\partial\mathbf{b}_w}\right|_{\bar{\mathbf{b}}_w}\Delta\mathbf{b}_w \end{pmatrix} \end{bmatrix}$$

where $\text{vec}(\cdot)$ returns the vector portion of the quaternion (i.e., the top three elements) and the bias errors are the difference between biases in the bounding states.

We use combined continuous preintegration factors that included both the inertial and bias errors together and relate to the full 15 degree-of-freedom state (see Equation 3). This combined continuous preintegration factor better models the measurement error state dynamics due to bias drift over the integration interval. The analytical Jacobians needed for graph optimization, bias Jacobians, and *closed-form* preintegrated measurements are included in the preintegration technical report [34]. To find the covariance of the above residual, continuous IMU error state dynamics noise characterization is performed, for details we refer the reader to the continuous preintegration paper [8].

## V. DETERMINING 3D PLANE FACTOR

In this section, we present the closest point (CP) plane representation, derivation of the plane factor for use in graph optimization, and our approach to computing the closest point measurement mean and covariance from a given set of points.

### A. Closest Point Plane Representation

We first explain a unique plane representation that is well suited for range based sensors. This "closest point (CP)" representation can be described as a 3D point that resides on the plane and is the *closest* to the current frame's origin. The benefit of this representation is that it is already in its minimal representation and is singularity free if special care is taken to select the frame of reference it is defined from. By representing the plane as a single 3D point, we also have a simple additive error model when updating the parameter during optimization. This CP representation can be described using the Hesse normal vector $^G\mathbf{n}$ and distance scalar $^Gd$:

$$^G\boldsymbol{\Pi} = {}^G\mathbf{n}\,{}^Gd \quad (24)$$

$$\begin{bmatrix} {}^G\mathbf{n} \\ {}^Gd \end{bmatrix} = \begin{bmatrix} {}^G\boldsymbol{\Pi}/\left\|{}^G\boldsymbol{\Pi}\right\| \\ \left\|{}^G\boldsymbol{\Pi}\right\| \end{bmatrix} \quad (25)$$

It is important to point out that without special care, this representation still has a singularity when the value of $^Gd$ approaches zero. Any plane $^G\boldsymbol{\Pi}$ that intersects our frame of reference ($\{G\}$ in this case) will be represented as the same zero vector regardless of the plane's orientation since the closest point on that plane is at the origin. Nevertheless, we argue that this singularity is well suited in the case of plane estimation using range based sensors (e.g., LiDAR and RGBD cameras) since planes extracted from these sensors will not be ill-defined if they are represented in the frame they are extracted from. The singularity in practice would only arise if we transform a local CP plane, $^L\boldsymbol{\Pi}$, into a frame where the plane intersects its origin (see Figure 2). It was also noted in [14] that planes extracted from range based sensors that are close to intersecting the sensor frame should be consider "unreliable" if found and discarded.
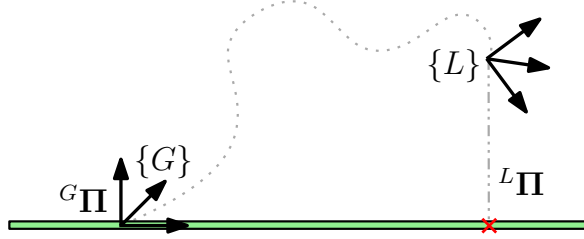
Fig. 2: A visual representation of the closest point on the plane. Also shown is an example of a local plane parameter $^L\mathbf{\Pi}$ that is well defined, while the global plane representation $^G\mathbf{\Pi}$ is ill-defined.
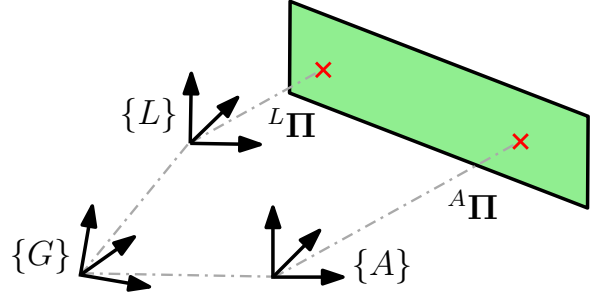


Fig. 3: Pictorial view of a closest point plane representation seen in the local $\{L\}$ frame which can be transformed into its anchor frame $\{A\}$ and vice versa.

## B. Anchor Plane Factor

To overcome the aforementioned singularity issue of the CP representation, we parameterize the plane in the first observation frame, guaranteeing that the distance to the plane will be non-zero (from here forward this will be denoted the "anchor" frame/state). As seen in Figure 3, the transform of the plane representation from one frame to another is *not* a direct 3D point transformation, and instead requires the calculation of the CP in the new frame. Using the Hesse plane representation we can map a plane represented in the anchor frame $\{A\}$ into the local frame $\{L\}$ as:

$$\begin{bmatrix} ^L\mathbf{n} \\ ^L d \end{bmatrix} = \begin{bmatrix} ^L_A\mathbf{R} & 0 \\ -^A\mathbf{p}_L^\top & 1 \end{bmatrix} \begin{bmatrix} ^A\mathbf{n} \\ ^A d \end{bmatrix} \tag{26}$$

where $^L_A\mathbf{R}$ is the relative rotation between the local and anchor LiDAR frames, $^A\mathbf{p}_L$ is the position of the local LiDAR frame seen from the anchor LiDAR frame.

The transform represents the rotation of the anchor plane normal vector into the local frame, and the subtraction of the distance between the two frames projected onto the anchor plane normal. Using the definition of the CP representation (24), we can use the above $\{^A\mathbf{n}, ^A d\}$ relation to obtain:

$$^L\mathbf{\Pi}(\mathbf{x}) = \left( ^L_A\mathbf{R}\,^A\mathbf{n} \right)\left( ^A d - ^A\mathbf{p}_L^\top\,^A\mathbf{n} \right) \tag{27}$$

For a given plane measurement, $^L\widehat{\mathbf{\Pi}}$, we have the following residual for use in graph optimization (see (7)):

$$r_\Pi(\mathbf{x}) = {}^L\mathbf{\Pi}(\mathbf{x}) - {}^L\widehat{\mathbf{\Pi}} \tag{28}$$

The analytical Jacobians needed for graph optimization are included in the accompanying technical report [35] and have been omitted here for space efficiency.

## C. Point to Plane Compression

To get the local CP measurement, we fit planes to the incoming point clouds from the LiDAR sensor. To find subsets of the unordered point cloud that correspond to planes, RANSAC or other plane segmentation methods can be used. We model each point measurement $^L\mathbf{p}_{mi}$ as a true measurement $^L\mathbf{p}_i$ being corrupted by some zero mean Gaussian noise:

$$^L\mathbf{p}_{mi} = {}^L\mathbf{p}_i + \mathbf{n}_p, \quad \mathbf{n}_p \sim \mathcal{N}(0, \mathbf{R}_d) \tag{29}$$

We look to first compress the extracted subset of points into a *local* CP primitive and matching covariance that can be used

in optimization. We can start by formulating a weighted least squared optimization problem where we seek to minimize the point-to-plane distances between extracted points and the local CP measurement $^L\mathbf{\Pi}$:

$$^L\mathbf{\Pi}^* = \operatorname*{argmin}_{^L\mathbf{\Pi}} \sum_i \left\| \frac{^L\mathbf{\Pi}^\top}{\|^L\mathbf{\Pi}\|}\,^L\mathbf{p}_{mi} - \|^L\mathbf{\Pi}\| \right\|^2_{W_i^{-1}} \tag{30}$$

where $W_i$ is the inverse variance of the noise that corrupts the $i$th measurement. In practice, we also introduce a robust Huber loss to minimize the effect of outliers during optimization (see [36]). We minimize the above cost function using the Gauss-Newton method of iterative linearization of the residual about the current best estimate. Formally, we solve for the correction, $^L\widetilde{\mathbf{\Pi}}$, to our linearization point $^L\widehat{\mathbf{\Pi}}$:

$$^L\widetilde{\mathbf{\Pi}} = -\left( \sum_i \mathbf{J}_i^\top\,W_i\,\mathbf{J}_i \right)^{-1} \left( \sum_i \mathbf{J}_i^\top W_i\,r_i({}^L\widehat{\mathbf{\Pi}}) \right)$$

where we have the following:

$$\mathbf{J}_i = \frac{^L\mathbf{p}_{mi}^\top}{\|^L\widehat{\mathbf{\Pi}}\|} - \left( ^L\mathbf{p}_{mi}^\top\,^L\widehat{\mathbf{\Pi}} \right)\frac{^L\widehat{\mathbf{\Pi}}^\top}{\|^L\widehat{\mathbf{\Pi}}\|^3} - \frac{^L\widehat{\mathbf{\Pi}}^\top}{\|^L\widehat{\mathbf{\Pi}}\|} \tag{31}$$

$$W_i = \left( \frac{^L\widehat{\mathbf{\Pi}}^\top}{\|^L\widehat{\mathbf{\Pi}}\|}\,\mathbf{R}_d\,\frac{^L\widehat{\mathbf{\Pi}}}{\|^L\widehat{\mathbf{\Pi}}\|} \right)^{-1} \tag{32}$$

Additionally we can calculate the covariance matrix of the final local CP $^L\widehat{\mathbf{\Pi}}$ as the following:

$$\mathbf{P}_\Pi = \left( \sum_i \mathbf{J}_i^\top\,W_i\,\mathbf{J}_i \right)^{-1} \tag{33}$$

In summary, we compress each of the extracted subsets of the point cloud into *local* CP plane, $^L\widehat{\mathbf{\Pi}}$, representations, which are then directly used to construct plane factors (28). We found that using a robust Huber loss function on the plane factors led to lower sensitivity to poor plane measurement compression performance.

## VI. Experimental Results

### A. LiDAR-Inertial Simulator

To evaluate the feasibility of the proposed system, a custom LiDAR IMU simulator was developed. A 2D floor plan was created and extruded vertically to create a Manhattan world environment (we note, for clarity, that the CP

TABLE I: Realistic parameters used in simulation.

| Simulation Parameter | Value |
|---|---|
| Gyroscope Noise Density | $0.005\ rad/s\sqrt{Hz}$ |
| Gyroscope Random Walk | $4.0\text{e-}06\ rad/s^2\sqrt{Hz}$ |
| Accelerometer Noise Density | $0.01\ m/s^2\sqrt{Hz}$ |
| Accelerometer Random Walk | $2.0\text{e-}04\ m/s^3\sqrt{Hz}$ |
| LiDAR Point Deviation | 1 and 3 $cm$ |
| LiDAR Angular Resolution | $0.25°$ |
| LiDAR Zenith Angles | $3.2°,0.0°,-3.2°,-6.4°,$ $-9.5°,-12.5°,-15.4°,-18.3°$ |
| Rotation LiDAR to IMU | [-1,0,0;0,1,0;0,0,-1] |
| Position IMU in LiDAR | [0;0.04;-0.06] $m$ |
| Global Gravity | [0,0,9.81] $m/s^2$ |
| LiDAR / IMU Sensor Rate | 5 / 800 $Hz$ |

TABLE II: Average RMSE over 80 Monte-Carlo simulations at different LiDAR noise values.

| | Closest Point | | Quaternion [6] | |
|---|---|---|---|---|
| Units | m | deg | m | deg |
| 1 cm | 0.005 | 0.027 | 0.016 | 0.081 |
| 3 cm | 0.012 | 0.057 | 0.033 | 0.126 |

are shown in Figure 4 and Table II. The proposed CP representation and anchor plane factors were able to localize in the planar environment with high accuracy at different levels of LiDAR sensor noise. The simulations were done in realtime with the plane correspondences known and solved using iSAM2. The large non-zero orientation error towards the beginning is due to the sensors remaining stationary for a period after initialization with only a small number of far away planes constraining the orientation.

The beginning of the simulated trajectory has limited amounts of loop closures due to the entering and leaving of rooms causing the estimation error to increase as one would normally see in odometry systems (see Figure 5). After 300 seconds the trajectory re-enters the long hallway and returns towards the starting position re-observing previously seen sections of the hallway. As seen in Figure 4, loop closures with previous planes rapidly decrease the estimator error towards zero.

### C. Plane Representation Comparison

To evaluate the effect of using the CP representation, we compare against the state-of-the-art that leverages the quaternion and its minimal error state [6]. We compressed the sets of point clouds into the quaternion representation using the same methodology used for CP, in which we perform a minimization on the point-to-plane distances. We implemented the "relative quaternion" factor proposed by Kaess, which is directly comparable to our "anchored" CP representation (whose analytical Jacobians can be found in our technical report [35]). As shown in Table II, the CP representation yielded improved accuracy over its quaternion counterpart. While the results presented here were generated using the iSAM2 solver, we found that during full batch optimization the quaternion representation converged slower compared to CP. Our conjecture for these results is that the CP-based measurement model has a better linear Gaussian approximation than the quaternion parametrization, and thus provides better numerical performance.

### D. Real-World Experiments

To further validate that the proposed LIPS system can be realized on physical sensors, the issue of plane correspondences needed to be addressed. While there have been works on matching planes in 3D space [19, 21], a simple Mahalanobis-distance test between incoming local CPs and existing planes was sufficient for our small scale experiments. Point clouds were first processed offline using RANSAC plane segmentation, available in the Point Cloud Library (PCL) [37], to find planar subsets. The measurement compression and estimator were able to run in realtime, but
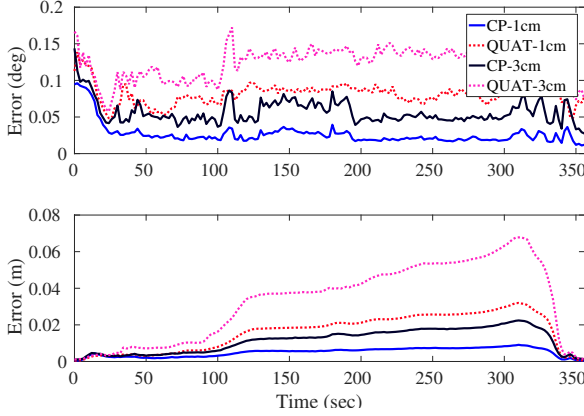


Fig. 4: RMSE results from 80 Monte-Carlo simulations showing the achievable accuracy of the proposed LIPS system running in realtime. We show RMSE results for both CP and quaternion representations.

representation can handle arbitrary plane orientations). A set of ordered 3D control points are used to create a 3D spline trajectory through the environment (see Figure 5 for the generated trajectory). Using analytic spline differentiation, the true IMU measurements can be obtained at any time along the trajectory. At a given LiDAR sensing frequency, rays were generated using the intrinsic LiDAR sensor model defined by the angular resolution and vertical zenith angles. Generated rays are then intersected with all planes in the environment and all ray-plane intersections are found. The final step performs invalidation of intersections that should not be generated due to occlusions by enforcing that each ray should *only* hit the plane that is *closest* to the LiDAR frame.

The simulation parameters shown in Table I are modeled after a Quanergy M8 sensor with a ADIS16448 IMU rigidly attached. Following the conventional inertial model, IMU measurements have additive discrete bias and white noise terms corrupting the true value of each measurement axis. The noise corrupting the generated 3D LiDAR points is modeled as an additive white noise to each measurement axis.

### B. Monte-Carlo Simulations

80 Monte-Carlo simulations of the LIPS system were performed at varying LiDAR noise values, whose results
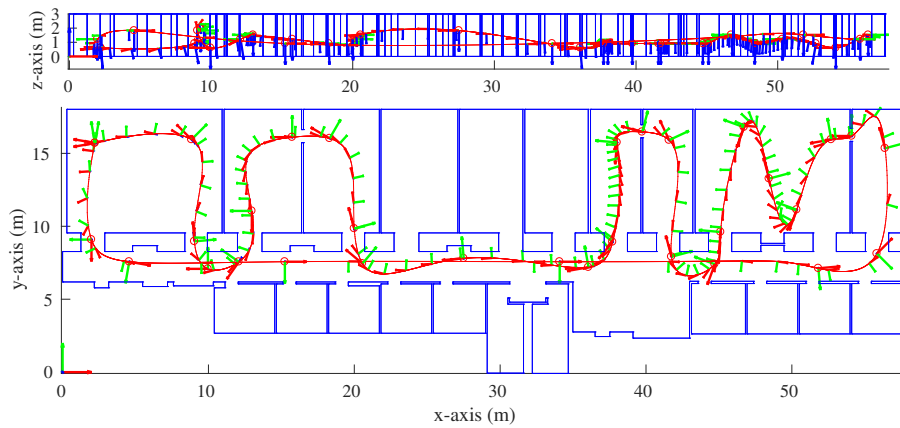
Fig. 5: Generated 180 meter long simulation trajectory through a 3D environment. The original 2D floor plan (bottom) has been extruded, and a spline was fitted to control points to generate a complete trajectory. The trajectory starts in the top left corner and weaves in and out of rooms with varying height from the floor before finally re-entering the hallway and returning back to the starting location.

for real-world applications the RANSAC plane extraction will need to be substituted or have its execution time decreased.

In this test, planar objects were placed around the LiDAR sensor to allow for easy RANSAC extraction, to avoid degenerate motions [38], and to ensure that the LiDAR was fully constrained in all degrees of freedom (see Figure 6). An eight channel Quanergy M8 LiDAR operating at 10Hz was used with a Microstrain 3DM-GX3-25 IMU attached to the bottom of the LiDAR operating at 500Hz. We manually estimated the LiDAR to IMU extrinsic transformation but this could easily be added into the factor graph for online estimation. To evaluate the estimation drift, the sensor unit was moved in front of the planar surfaces and returned to the same starting location. As seen in Figure 7, after a 30 meter trajectory distance, the difference between the start and end poses was 1.5 cm corresponding to 0.05% error over the trajectory length.

## VII. Conclusions and Future Work

We have presented and leveraged a novel CP planar representation in a LiDAR-inertial SLAM system, allowing for singularity free optimization of the full planar state (both orientation and distance). The proposed anchor plane factor was then combined with continuous preintegration to allow for a robust graph-based SLAM system to be constructed. This system was evaluated on a custom made LiDAR simulator on a 180 meter trajectory and shown to achieve impressive localization accuracy at different noise levels when compared to another representation. Finally, a proof-of-concept real-world experiment was conducted and shown to achieve similar localization accuracy. Moving forward, one of the drawbacks of any LiDAR SLAM system are the possible cases where the extracted planes do not fully constrain the pose estimate. We look to include other feature information from camera sensors and expand our experiential analysis to evaluate full building localization accuracy.

## References

[1] M. Li and A. I. Mourikis. "High-precision, consistent EKF-based visual-inertial odometry". In: *The International Journal of Robotics Research* 32.6 (2013), pp. 690–711.

[2] K. Eckenhoff, P. Geneva, and G. Huang. "Direct visual-inertial navigation with analytical preintegration". In: *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2017, pp. 1429–1435.

[3] J. A. Hesch, F. M. Mirzaei, G. L. Mariottini, and S. I. Roumeliotis. "A Laser-aided Inertial Navigation System (L-INS) for human localization in unknown indoor environments". In: *2010 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2010, pp. 5376–5382.

[4] J. Zhang and S. Singh. "Enabling aggressive motion estimation at low-drift and accurate mapping in real-time". In: *International Conference on Robotics and Automation*. IEEE. 2017, pp. 5051–5058.

[5] Y. Yang and G. Huang. "Acoustic-inertial underwater navigation". In: *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2017, pp. 4927–4933.

[6] M. Kaess. "Simultaneous localization and mapping with infinite planes". In: *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2015, pp. 4605–4611.

[7] K. Eckenhoff, P. Geneva, and G. Huang. "High-Accuracy Preintegration for Visual-Inertial Navigation". In: *Proc. of International Workshop on the Algorithmic Foundations of Robotics*. San Francisco, CA, Dec. 2016.

[8] K. Eckenhoff, P. Geneva, and G. Huang. "Continuous Preintegration Theory for Graph-based Visual-Inertial Navigation". In: *CoRR* abs/1805.02774 (2018). arXiv: 1805.02774. URL: http://arxiv.org/abs/1805.02774.

[9] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. D. Reid, and J. J. Leonard. "Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age". In: *IEEE Transactions on Robotics* 32.6 (2016), pp. 1309–1332.

[10] D. Viejo and M. Cazorla. "3D plane-based egomotion for SLAM on semi-structured environment". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2007, pp. 2761–2766.

[11] K. Pathak, A. Birk, N. Vaskevicius, M. Pfingsthorn, S. Schwertfeger, and J. Poppinga. "Online three-dimensional SLAM by registration of large planar surface segments and closed-form pose-graph relaxation". In: *Journal of Field Robotics* 27.1 (2010), pp. 52–84.

[12] C. Ulaş and H. Temeltaş. "A fast and robust scan matching algorithm based on ML-NDT and feature extraction". In: *2011 International Conference on Mechatronics and Automation (ICMA)*. IEEE. 2011, pp. 1751–1756.

[13] J. Xiao, B. Adler, and H. Zhang. "3D point cloud registration based on planar surfaces". In: *2012 IEEE Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*. IEEE. 2012, pp. 40–45.
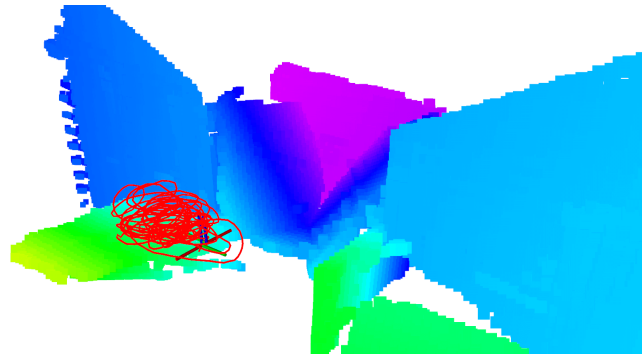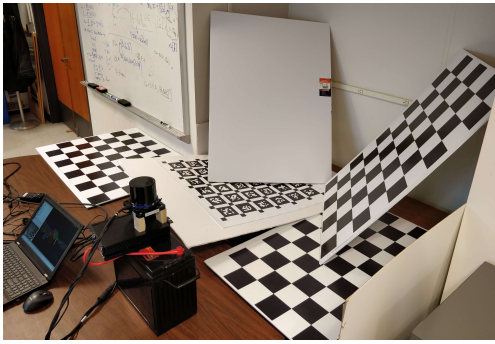
Fig. 6: Experimental environment that the proposed localization operated in (left) and reconstructed depth map of the planar surfaces (right).
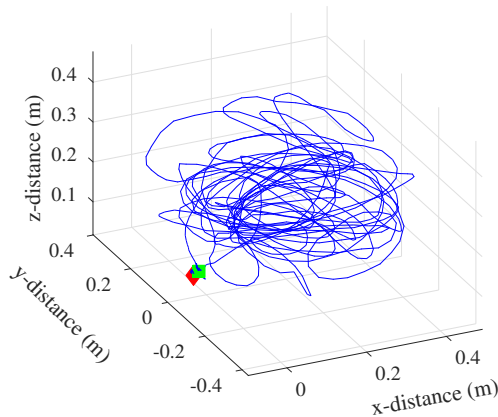


Fig. 7: 3D trajectory of the sensor during the experiment. Total path length was 30 meters with the difference between the starting and end position being 1.5cm. The green square and red diamond denote the start and end of the trajectory, respectively.

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

[15] P. F. Proença and Y. Gao. "Probabilistic combination of noisy points and planes for rgb-d odometry". In: *Conference Towards Autonomous Robotic Systems*. Springer. 2017, pp. 340–350.

[16] J. Weingarten and R. Siegwart. "3D SLAM using planar segments". In: *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2006, pp. 3062–3067.

[17] K. Pathak, A. Birk, N. Vaskevicius, and J. Poppinga. "Fast registration based on noisy planes with unknown correspondences for 3-D mapping". In: *IEEE Transactions on Robotics* 26.3 (2010), pp. 424–441.

[18] A. J. Trevor, J. G. Rogers, and H. I. Christensen. "Planar surface SLAM with 3D and 2D sensors". In: *2012 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2012, pp. 3041–3048.

[19] Y. Taguchi, Y.-D. Jian, S. Ramalingam, and C. Feng. "Point-plane SLAM for hand-held 3D sensors". In: *2013 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2013, pp. 5182–5189.

[20] R. F. Salas-Moreno, B. Glocken, P. H. Kelly, and A. J. Davison. "Dense planar SLAM". In: *2014 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE. 2014, pp. 157–164.

[21] M. Hsiao, E. Westman, G. Zhang, and M. Kaess. "Keyframe-based dense planar SLAM". In: *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2017, pp. 5110–5117.

[22] M. Hsiao, E. Westman, and M. Kaess. "Dense Planar-Inertial SLAM with Structural Constraints". In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. 2018.

[23] L. Zhang, D. Chen, and W. Liu. "Point-plane SLAM based on line-based plane segmentation approach". In: *2016 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. IEEE. 2016, pp. 1287–1292.

[24] L. Ma, C. Kerl, J. Stückler, and D. Cremers. "CPA-SLAM: Consistent plane-model alignment for direct RGB-D SLAM". In: *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2016, pp. 1285–1291.

[25] C. X. Guo and S. I. Roumeliotis. "IMU-RGBD camera navigation using point and plane features". In: *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2013, pp. 3164–3171.

[26] G. Grisetti, R. Kummerle, C. Stachniss, and W. Burgard. "A Tutorial on Graph-Based SLAM". In: *IEEE Intelligent Transportation Systems Magazine* 2.4 (2010), pp. 31–43.

[27] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard. "g²o: A general framework for graph optimization". In: *2011 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2011.

[28] K. Eckenhoff, L. Paull, and G. Huang. "Decoupled, Consistent Node Removal and Edge Sparsification for Graph-based SLAM". In: *in Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. Oct. 2016.

[29] N. Trawny and S. I. Roumeliotis. *Indirect Kalman Filter for 3D Attitude Estimation*. Tech. rep. University of Minnesota, Dept. of Comp. Sci. & Eng., Mar. 2005.

[30] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. J. Leonard, and F. Dellaert. "iSAM2: Incremental smoothing and mapping using the Bayes tree". In: *The International Journal of Robotics Research* 31.2 (2012), pp. 216–235.

[31] F. Dellaert. *Factor graphs and GTSAM: A hands-on introduction*. Tech. rep. Georgia Institute of Technology, 2012.

[32] T. Lupton and S. Sukkarieh. "Visual-Inertial-Aided Navigation for High-Dynamic Motion in Built Environments Without Initial Conditions". In: 28.1 (Feb. 2012), pp. 61–76.

[33] A. B. Chatfield. *Fundamentals of High Accuracy Inertial Navigation*. Reston, VA: American Institute of Aeronautics and Astronautics, Inc., 1997.

[34] K. Eckenhoff, P. Geneva, and G. Huang. *Continuous Preintegration Theory for Visual-Inertial Navigation*. Tech. rep. RPNG-2018-CPI. Available: http://udel.edu/~ghuang/papers/tr_cpi.pdf. University of Delaware, 2018.

[35] P. Geneva, K. Eckenhoff, Y. Yang, and G. Huang. *LIPS: Lidar Inertial 3D Plane SLAM*. Tech. rep. RPNG-2018-LIPS. Available: http://udel.edu/~ghuang/papers/tr_lips.pdf. University of Delaware, 2018.

[36] E. Eade. "Gauss-Newton / Levenberg-Marquardt Optimization". In: *Technical Report* (2013).

[37] R. B. Rusu and S. Cousins. "3d is here: Point cloud library (pcl)". In: *2011 IEEE International Conference on Robotics and automation (ICRA)*. IEEE. 2011, pp. 1–4.

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