LiDAR-Inertial 3D SLAM with Plane Constraint for Multi-story Building

Jiashi Zhang1*, Yuzhu Su2*, Chengyang Zhang1, Jianxiang Jin1,3, Jun Wu1,3, Rong Xiong1,3, Qiuguo Zhu1,2,3†

Abstract—The ubiquitous planes and structural consistency are the most apparent features of indoor multi-story buildings compared with outdoor environments. In this paper, we propose a tightly coupled LiDAR-Inertial 3D SLAM framework with plane features for the multi-story building. The framework we proposed is mainly composed of three parts: tightly coupled LiDAR-Inertial odometry, extraction of structural representative planes, and factor graph optimization. By building a local map and inertial measurement unit (IMU) pre-integration, we get LiDAR scan-to-local-map matching and IMU measurements, respectively. Minimize the joint cost function to obtain the LiDAR-Inertial odometry information. Once a new keyframe is added to the graph, all the planes of this keyframe that can represent structural features are extracted to find the constraint between different poses and stories. A keyframe-based factor graph is conducted with the constraint of planes, and LiDAR-Inertial odometry for keyframe poses refinement. The experimental results show that our algorithm has outstanding performance in accuracy compared with the state-of-the-art algorithms.

I. INTRODUCTION

With the development of environmental perception capabilities, the scenarios can be explored are expanding from 2D to 3D by drone and legged robot. Accurate state estimation and mapping are the basic premises for applying robots toward to the real world. Facing indoor environments, especially multi-story buildings, the robot must obtain a globally consistent pose estimation on different floors. Otherwise the point clouds of different floors will overlap or be deflected, which cannot be used for autonomous navigation of robots. How to make the robot obtain globally consistent pose estimation on different floors is the focus and difficulty of SLAM in multi-story buildings.

A 3D LiDAR based on scanning mechanism has the advantages of textureless, invariant to the illumination, and broad horizontal of view (FOV) of 360°, which is generally used in indoor environments[1], [2]. Under normal circumstances, LiDAR-aided SLAM mainly uses extracting corner points and surf points method [3], [4], Normal Distributions Transform (NDT)[5] scan matching, or floor extraction[6] methods to achieve SLAM for a single floor. Although many algorithms implement SLAM by extracting planes in indoor environments, most only use plane constraints in the odometry part and achieve accurate SLAM algorithms by finding the scan-to-scan plane correspondence. These algorithms can achieve good results in scenes with a single indoor floor or a relatively small number of floors. However, when the robot explores from bottom to top in a multi-story building, the existing algorithms cannot achieve accurate state estimation on the robot’s 6-DOF, due to long-distance and loop closure does not work. In the multi-story SLAM, due to the consistency of structure between different floors, some planes on different floors can represent the same building structure. When the robot observes the same plane on different stories, it can correct the current pose. Here we call these planes structural representative planes (SRP).

This paper uses SRPs to build global constraints in different stories. Our framework has three parts: (1) tightly coupled LiDAR-Inertial odometry, (2) extraction of representative planes of the structure, and (3) factor graph optimization. The odometry is obtained by jointly optimizing the relative pose of the scan-to-local-map and the IMU pre-integration measurements. According to the odometry information, all
the SRP will be extracted as candidates for the global plane constraint once a new keyframe is selected. Transform the global SRP to the keyframe coordinate system, and construct the global constraint relationship between keyframes according to the direction of planes’ normals and the distance to the coordinate origin. Add odometry information and constraint information from planes to the factor graph, perform global optimization, and get the accurate pose of each keyframe.

The main contributions of this paper are summarized as follows:

- We propose the method of finding and constructing the global constraints of SRP in the multi-story blocks to achieve accurate 6-DOF state estimation of the robot when the loop closure is not possible.
- We propose a tightly coupled LiDAR-Inertial, keyframe-based SLAM framework to get the dense 3D point cloud maps of multi-story blocks.
- We validate the algorithm using the data collected from Velodyne VLP-16 and Xsens Mti-300 mounted on a real quadruped robot (Jueying Robot). Compared with the method without SRP, better results are obtained.

II. RELATED WORK

LiDAR Inertial odometry 3D LiDAR and IMU have been widely used in SLAM, both indoors and outdoors. The fusion methods of LiDAR and IMU are mainly divided into two categories: loosely coupled and tightly coupled. In the field of loosely coupled, LOAM [3] is a classic loosely coupled framework. It uses the orientation calculated by the IMU de-skew the point cloud and as prior information in the optimization process. The same method is also applied to its variants LeGO-LOAM [7], LIO-Mapping [8] implemented LiDAR-Inertial tightly coupled algorithm by optimizing the cost function that includes both LiDAR and inertial measurements. However, the optimization process is carried out in a sliding window, so the time-consuming calculations make it impossible to maintain real-time performance. In their follow-up work, R-LINS [9], they use iterated-ESKF for the first time to achieve LiDAR-Inertial tightly coupled fusion and propose an iterated Kalman filter [10] to reduce wrong matchings in each iteration. A tightly coupled framework based on iterated Kalman filter is presented in [11], similar to R-LINS. An incremental kd-tree data structure is adopted to ensure cumulative updates and dynamic balance to ensure fast and robust LiDAR mapping. LIO-SAM [4] proposed by Shan T optimizes the measurements of LiDAR and IMU by factor graph, and at the same time, estimates the bias of the IMU.

SLAM related to plane features Whether in vision-based SLAM or LiDAR-based SLAM, plane-related features are widely used to improve state estimation accuracy. In LiDAR-based SLAM, LOAM [3] proposed extracting feature points from planar surface patches and sharp edges based on curvature calculation and improved the iterative closest point (ICP) [12] method based on the extracted feature points demonstrating the superb LiDAR odometry effect. Koide K [6] realize SLAM in a large-scale environment by detecting the ground, assuming that the indoor environment is a single flat floor. But this assumption is not applicable in all scenes and can only limit the height on the z-axis. LIPS [13] extract the plane in the three-axis direction of the point cloud, not only the ground plane, and combine the plane and IMU measurements in a graph-based framework. At the same time, the closeset point (CP) is used to represent the plane to solve the singularity. π-LSAM, an indoor environment SLAM system using planes as landmarks, is proposed by Zhou L [14]. They adopt plane adjustment (PA) as the back-end to optimize plane parameters and poses of keyframes, similar to bundle adjustment (BA) in visual SLAM. Their subsequent work [15] extended this by using first-order Taylor expansion to replace the Levenberg Marquardt (LM) [16] method. To achieve faster computational speed, they define the integrated cost matrix (ICM) for each plane and achieve outstanding SLAM effects in a single-layer indoor environment. All of the above frameworks use a single LiDAR or a loosely coupled method of LiDAR and IMU as the front-end. On the contrary, we use a tightly coupled LiDAR-Inertial method as the front-end, which can obtain a more accurate prior pose of the keyframe, making it more precise when looking for the corresponding between the planes.

III. LIDAR-INERTIAL ODOMETRY

The Lidar-Inertial odometry, which is adapted from [8], maintains two sliding windows for building local map and optimizing states. Although it cannot run in real time, it can calculate an accurate pose transformation between two keyframes.

A. IMU Pre-integration

The LiDAR and IMU reference frames at time \( t \) are noted \( L \) and \( I \), respectively. The state \( X^W_t \) of IMU to be estimated in the world frame \( W \) and the extrinsic matrix \( T^t_f \) from IMU to LiDAR are written as:

\[
X^W_t = \begin{bmatrix} p^W_t^I & q^W_t^I & t^I \\ q^W_t^I & T^W_t^I & b^I_a & b^I_g \\ \end{bmatrix}^T
\]

where \( p^W_t^I, q^W_t^I, \) and \( q^W_t^I \) are the position, velocity, and orientation of IMU in the world frame \( W \) at time \( t \). \( b^I_a \) and \( b^I_g \) are the bias of accelerometer and gyroscope of IMU.

Let \( t_i \) and \( t_j \) be the starting time and ending time of a raw LiDAR scan \( \mathcal{J}_i \), respectively, so the pre-integration measurements \( \Delta p_{ij}, \Delta v_{ij}, \Delta q_{ij} \) of IMU from time \( t_i \) to \( t_j \) are computed as:

\[
\Delta p_{ij} = \sum_{k=i}^{j-1} \left[ \Delta v_{ik} \Delta t + \frac{1}{2} \Delta R_{ik} (\hat{a}_k - b_{ik} - n_k) \Delta t \right]
\]

\[
\Delta v_{ij} = \sum_{k=i}^{j-1} \Delta R_{ik} (\hat{a}_k - b_{ik} - n_k) \Delta t
\]

\[
\Delta q_{ij} = \prod_{k=i}^{j-1} \delta q_k = \prod_{k=i}^{j-1} \left\{ \frac{1}{2} \Delta t \left( \hat{\omega}_k - b_{wk} - n_w \right) \right\}
\]
Readers can refer to [17] for the detailed derivation of Eq. (2).

B. Scan Deskewing and Feature Extraction

Due to the relative movement between the laser and the robot, there will be motion distortion for the raw LiDAR output $S_j$, where $S_j$ represents the point cloud starting from time $t_i$ to time $t_j$. Every point $x(t) \in S_j$ is transformed to the correct position by linear interpolation to $T_{ij}$ according to its timestamp, where $t \in [t_i, t_j]$. $T_{ij}^{p}$ is obtained by IMU pre-integration and extrinsic matrix $T_{ij}^{f}$, and the undistorted scan is represented by $S_j$.

To improve the efficiency of calculation, only the feature points that can reflect the characteristics of the surrounding environment are selected to find the relative pose of the LiDAR. Here we use the method of extracting feature points located on sharp edges and planar surfaces proposed by LiDAR. Here we take the method of extracting feature points from $S_j$ as denoted as $S^{e}_{j}$ and $S^{p}_{j}$, respectively.

C. LiDAR Relative Measurements

When the new feature points $S^{e}_{j}$ and $S^{p}_{j}$ are extracted, the measurements of LiDAR need to be found to jointly perform the optimization with IMU.

1) Building Local Map: Since the points of a single scan are not dense enough, to obtain more accurate LiDAR measurements, we use a sliding window to construct a local map. The sliding window contains $n$ LiDAR frames from time $t_{i-1}$ to time $t_{i-n}$. Since we have extracted planar points and edge points separately, we transform $\{S^{e}_{i-n}, ..., S^{e}_{i-2}, S^{e}_{i-1}\}$ and $\{S^{p}_{i-n}, ..., S^{p}_{i-2}, S^{p}_{i-1}\}$ to frame $L_{i-1}$ respectively with $\{T^{e}_{i-n}, ..., T^{e}_{i-2}, T^{e}_{i-1}\}$ and $\{T^{p}_{i-n}, ..., T^{p}_{i-2}, T^{p}_{i-1}\}$ to obtain two feature local maps, $M^{e}_{i-1}$ and $M^{p}_{i-1}$.

2) Scan Matching: The relationship between the feature points and the local maps at time $t_i$ are calculated by the point-line and the point-plane distances. First, transform the feature points $S^{e}_{i}$ and $S^{p}_{i}$ to frame $L_{i-1}$. The prediction transformation $T^{f}_{i-1}$ used here is obtained through IMU pre-integration and extrinsic matrix $T_{ij}^{f}$. Here we take the plane points as an example. For each transformed plane point $x^{p}_{j}$, find the nearest $m$ points in $M^{p}_{i-1}$ to fit a plane in the frame $L_{i-1}$ and express in Hesse normal form:

$$x^{T}n_{p}-d_{p}=0$$

where $n_{p}$ is the unit normal vector of plane, and $d_{p}$ is the distance from plane to the origin of frame $L_{i-1}$. So for each plane point $x^{p}_{j} \in S^{p}_{j}$, the residual is expressed as the point-plane distance:

$$r_{\beta}(T^{f}_{i-1}) = (T^{p}_{i-1}x^{e}_{j} + n^{p}_{i-1})^{T}n_{p} - d_{p}$$

Similar to the calculation method of the plane point, the Hesse normal form can also describe the line in $\mathbb{R}^2$.

For each edge point, the residual is represented as the point-line distance:

$$r_{\beta}(x^{e}_{j}) = (T^{e}_{i-1}x^{e}_{j} + n^{e}_{i-1})^{T}n_{e} - d_{e}$$

D. Front-End Optimization

We build a cost function including IMU measurements and LiDAR measurements jointly and optimize all the states in the sliding window iteratively. For a sliding window of size $n$ at time $t_i$, the states need to be optimized is $X_i = [T^{e}_{i-n}, ..., T^{e}_{i-(n-1)}]$, and the final cost function is described as:

$$\min_{X_{i}} \frac{1}{2} \sum_{a \in \{i-n,...,i-1\}} \|r_{\beta}(z_{\alpha+1}^{\alpha}, X_{i})\|_{C^{a}_{i,a+1}}^{2} + \sum_{\beta \in \{i-n,...,i-1\}} \|r_{\beta}(X_{i})\|_{C^{\alpha}_{i}}^{2} + \sum_{\gamma \in \{i-n,...,i-1\}} \|r_{\gamma}(X_{i})\|_{C^{\gamma}_{i}}^{2}$$

where $\|X\|^{2}_{C} = X^{T}CX$ and $r_{\beta}(X_{i})$ is the residual of IMU measurements, which is defined in[8]. $r_{\beta}(X_{i})$ and $r_{\beta}(X_{i})$ are the residuals of planar points matching and edge points matching. $C^{a}_{i,a+1}, C^{\alpha}_{i}, C^{\gamma}_{i}$ represent the covariance matrix. This non-linear least squares problem is solved using the Levenberg–Marquardt algorithm[16].

Fig. 2. System overview of our algorithm.
IV. SRP CONSTRAINT AND GRAPH OPTIMIZATION

In this part, we extract keyframes based on the LiDAR-Inertial odometry and extract all SRP from the LiDAR scan in the keyframe coordinate system, find the correspondence in the entire graph and construct constraints as demonstrated in Fig. 3.

A. SRP Extraction

For the calculation efficiency, we select keyframes as vertices of the factor graph according to the odometry of the front-end. Since we are using a LiDAR based on scanning mechanism, the change of the yaw angle does not affect the selection of keyframes. The new keyframe will be selected only when the distance between the new frame and the previous keyframe exceeds 1 m or the pitch angle or roll angle exceeds 10°.

We extract all SRP from the corrected LiDAR scan $\mathcal{S}_i$ for each newly added keyframe $K_i$. Here we define the plane as $\pi(n, d)$ through the Hesse normal form described by Eq. (3). $n = [n_x, n_y, n_z]^T$ represents the unit normal vector of the plane, and $d$ represents the distance from the coordinate origin of $K_i$ to the plane. Next, apply RANSAC[18] to extract planes for $\mathcal{S}_i$, but not all planes are reserved for building constraints, but only those planes that can represent the structure of the building (e.g., ground, walls, etc.) are selected. Here we adopt the following strategies for the extraction of SRP:

- Keep all the planes with more than $N$ points (Here, we set $N$ to 400).
- According to the normal vector of the extracted plane, three planes containing the most points and almost orthogonal are retained.
- Use 80% of the points in $\mathcal{S}_i$ to extract the plane, and the remaining points default to the unextractable points.

In a multi-story building, the walls between different floors are likely to be on the same plane in space, but small planes such as doors and cabinets are usually not associated between different floors. Therefore, we use the RANSAC algorithm to extract planes according to the number of inliers from large to small, and extract the most obvious planes first. The plane of the ground can be used to constrain the change of the Z-axis of the robot within the same floor and during stair climbing. If the first 80% cannot find three orthogonal SRPs, it is considered that there are no SRPs in the remaining 20%. We build constraints using already found SPRs (maybe 1 or 2). Too many planes are extracted will increase the uncertainty of the RANSAC process and cause mismatches in the plane matching process. Here we only use three orthogonal planes to obtain the precise pose of the LiDAR with 6-DOF. At the same time, fewer edges will be constructed in the factor graph to reduce the calculation time.

B. SRP Global Constraint

To construct the global constraint, all SRP extracted from keyframe $K_i$ will be checked whether they have appeared in the previous keyframes. Here we denote all the planes added to the graph as $\Pi = \{\pi_{K_0}^0, \ldots, \pi_{K_i}^i, \ldots, \pi_{K_{i-1}}^{i-1}\}$, and the SRP under the $K_i$ frame as $\Pi^{K_i} = \{\pi_{K_i}^1, \ldots, \pi_{K_i}^{K_i}\}$. First, according to the optimized results $T_{K_m}^W, m \in \{1, \ldots, i-1\}$ and the front-end odometry $T_{K_{i-1}}^W$, the planes in $\Pi$ are transformed to the frame of keyframe $K_i$.

$$T_{K_m}^{K_i} = T_{K_{i-1}}^W T_{K_m}^W T_{K_{i-1}}^{K_m} = \begin{bmatrix} R_{K_i}^K & p_{K_i}^K \\ 0 & 1 \end{bmatrix}$$

(7)

For all $\pi_{K_i}^m(n^K, d^K) \in \Pi^{K_i}, m \in \{1, \ldots, K_i\}$, calculate the angle $\delta \theta$ between its normal vector $n^K$ and $n_{K_i}$ and the distance $\delta d$ between $d^K$ and $d_{K_i}$. Once $\delta \theta$ and $\delta d$ are lower than the preset threshold, add a plane edge to the factor graph. Otherwise, it’s considered a new plane and added to $\Pi$.

C. Graph optimization

When the LiDAR-Inertial odometry and SRP construct the constraints between keyframes, the SLAM problem is expressed in a factor graph. The vertices of the graph represent states of being optimized, and the edges represent the constraints formed by the sensors’ measurements, as
shown in Fig. 3. Following [19], [20], the maximum likelihood estimation problem is expressed as a nonlinear least-squares problem:

$$F(x) = \sum_{(i,j) \in \mathcal{E}} e(x_i, x_j, z_{ij})^T \Omega_{ij} e(x_i, x_j, z_{ij})$$  \hspace{1cm} (8)

where $x$ represents all states to be optimized and $x_i, x_j \in x$, $z_{ij}$ and $\Omega_{ij}$ represent the mean and the information matrix of a constraint between $x_i$ and $x_j$, $\mathcal{E}$ is the set of pairs of indices for which the constraint exist, and $e(x_i, x_j, z_{ij})$ is the error function between $x_i$, $x_j$ and $z_{ij}$. Eq. (8) is minimized by Gauss-Newton or Levenberg-Marquardt algorithm.

V. EXPERIMENTS

A. Experimental Settings

To verify the versatility of the algorithm, we conduct experiments in different buildings. We use the Jueying Mini robot (Fig. 4) equipped with Velodyne VLP-16 and Xsens Mti-300 to collect data from multiple sets of multi-story scenes. The LiDAR is fixed on the head of Jueying, and the IMU is assembled at the center of mass. Since there are currently no publicly available datasets of LiDAR and IMU for indoor multi-story scenes, we used Jueying to collect actual data in two buildings and named them Building A and Building B, respectively. Building A is a five-story building in the shape of a long corridor, and Building B is a six-story building with two long corridor-shaped scattered on the left and right. Our algorithm is tested on a PC with Intel Core i7-7567U, 16G memory.

B. Results and Analysis

We compared the state-of-the-art SLAM algorithms based on multi-sensor fusion, including Fast-LIO2 [11], LIO-SAM [4] and LOAM [3]. Due to the unique experimental scene, we cannot obtain the ground truth of the robot motion. At the same time, we set the robot’s starting point and ending point to be the same when collecting data to calculate the relative position and orientation deviation.

Overview The performance of Ours, Fast-LIO2 and LIO-SAM on the Building A and Building B datasets are shown in Fig. 5. We can see that Ours with SRP constraint is better than the others on both datasets because of plane constraints. When the 16-line LiDAR moves horizontally, the height estimation will produce more significant deviations, especially in degraded scenarios such as corridors. Despite the aid of IMU, there will still be cumulative errors, which is seen more clearly in Fast-LIO2 and LIO-SAM. Ours without SRP optimizes each state in the sliding window, which consumes more time, so the effect of height estimation is better, but in the end, it does not return to the starting point as well. The other two algorithms did not return to the starting point in the end due to the lack of performing loop closure. Trajectory Fig. 6 shows the trajectories of two datasets. LIO-SAM fails in Building A and Building B, so we did not plot its trajectory. Although we do not have global ground truth, we can see in Fig. 6(b) and Fig. 6(d) that in the staircase on the left of Building A and Building B, the other three algorithms drift a lot. Still, after adding plane constraints, ours can maintain the consistency of different floors. Table I provides the relative deviations of translation and rotation. Since accurate state estimation is achieved on other stories, our algorithm can return to the starting point without loop closure. Because of the same planes used to construct constraints, both translation and rotation are almost consistent with the starting point.

VI. CONCLUSION AND FUTURE WORK

This paper proposes a SLAM algorithm for indoor multi-story scenes with a plane as the main feature. To reduce the possibility of plane mis-matching, we use the tightly coupled LiDAR and IMU as the front-end. By plane matching and constraints building, the robot can eliminate the cumulative
error in different stories, and achieve an effect similar to "dimensionality reduction." Experiments show that our algorithm can significantly improve the state estimation and increase the accuracy of both localization and mapping. This improvement will boost robot performance in tasks such as indoor autonomous navigation and detection, which is extremely important for applications in indoor service and rescue robotics. However, the current plane matching process relies heavily on front-end odometry. In the future, it may be necessary to combine features unique to the plane to make the process more robust and fast.

REFERENCES

[1] 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), 2010, pp. 5376–5382.

[2] Q. Zou, Q. Sun, L. Chen, B. Nie, and Q. Li, “A Comparative Analysis of LiDAR SLAM-Based Indoor Navigation for Autonomous Vehicles,” IEEE Transactions on Intelligent Transportation Systems, pp. 1–15, 2021, early access.

[3] J. Zhang and S. Singh, “LOAM: Lidar Odometry and Mapping in Real-time.” in Robotics: Science and Systems, vol. 2, no. 9, 2014.

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

[5] M. Magnusson, A. Lilienthal, and T. Duckett, “Scan registration for autonomous mining vehicles using 3D-NDT,” Journal of Field Robotics, vol. 24, no. 10, pp. 803–827, 2007.

[6] K. Koide, J. Miura, and E. Menegatti, “A portable three-dimensional LiDAR-based system for long-term and wide-area people behavior measurement,” International Journal of Advanced Robotic Systems, vol. 16, no. 2, p. 172988141919424132, 2019.

[7] T. Shan and B. Englot, “LeGO-LOAM: Lightweight and Ground-Optimized Lidar Odometry and Mapping on Variable Terrain,” in 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2018, pp. 4758–4765.

[8] H. Ye, Y. Chen, and M. Liu, “Tightly Coupled 3D Lidar Inertial Odometry and Mapping,” in 2019 International Conference on Robotics and Automation (ICRA), 2019, pp. 3144–3150.

[9] C. Qin, H. Ye, C. E. Pranata, J. Han, S. Zhang, and M. Liu, “LINS: A Lidar-Inertial State Estimator for Robust and Efficient Navigation,” in 2020 IEEE Conference on Robotics and Automation (ICRA), 2020, pp. 8899–8906.

[10] B. Bell and F. Cathey, “The iterated Kalman filter update as a Gaussian-Newton method,” IEEE Transactions on Automatic Control, vol. 38, no. 2, pp. 294–297, 1993.

[11] W. Xu, Y. Cui, D. He, J. Lin, and F. Zhang, “FAST-LIO2: Fast Direct LiDAR-Inertial Odometry,” IEEE Transactions on Robotics, pp. 1–21, 2022, early access.

[12] S. Rusinkiewicz and M. Levoy, “Efficient variants of the ICP algorithm,” in Proceedings Third International Conference on 3-D Digital Imaging and Modeling, 2001, pp. 145–152.

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

[14] L. Zhou, T. Wang, and M. Kaess, “t-LSAM: LiDAR Smoothing and Mapping With Planes,” in 2021 IEEE International Conference on Robotics and Automation (ICRA), 2021, pp. 5751–5757.

[15] L. Zhou, D. Koppel, and M. Kaess, “LiDAR SLAM With Plane Adjustment for Indoor Environment,” IEEE Robotics and Automation Letters, vol. 6, no. 4, pp. 7073–7080, 2021.

[16] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, “G:O: A general framework for graph optimization,” in 2011 IEEE International Conference on Robotics and Automation, 2011, pp. 3607–3613.

[17] G. Grisetti, R. Kümmerle, C. Stachniss, and W. Burgard, “A Tutorial on Graph-Based SLAM,” IEEE Intelligent Transportation Systems Magazine, vol. 2, no. 4, pp. 31–43, 2010.