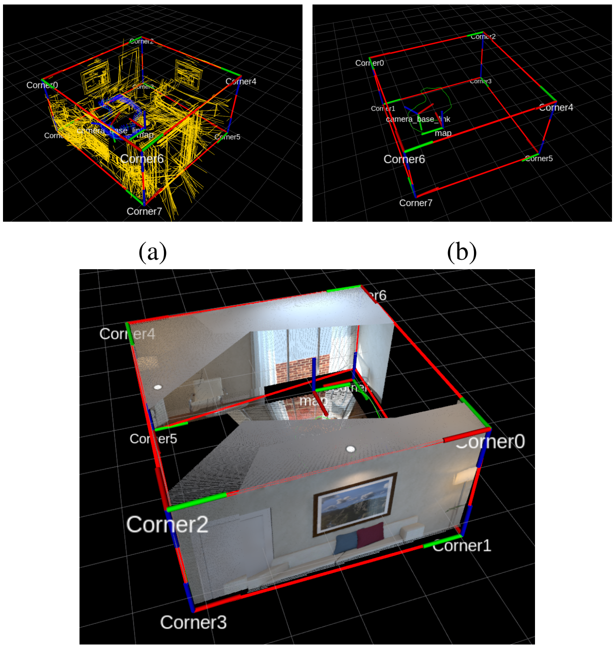
Low-Drift RGB-D SLAM with Room Reconstruction Using Scene Understanding

Zefeng Ye[1], Xin Jiang[2*], *Member, IEEE*, Yunhui Liu[1], *Fellow, IEEE*

***Abstract*— Room reconstruction task is very important for robot motion planning and navigation. Existing indoor dense mapping algorithms are inefficient in cluttered and occlusion environments because the reconstructed building environment consists of unmeaningful plane fragments. In this paper, we present an architecture for online, incremental room reconstruction which combines an accurate RGB-D SLAM and room layout understanding. We proposed an efficient scene understanding method, which detects room's corners to infer the wireframes and layout planes of room from single RGB-D image, even if the parts of the room are occluded. Moreover, the 3D global features (wireframes and layout planes of the building) can also improve the accuracy of state estimation, especially in geometric indoor environments. These 3D global features are treated as global consistent landmarks, it efficiently bounds the trajectory drift with the travel length increasing. On a public ICL-NUIM dataset, our algorithm achieves higher accuracy than other state-of-arts, and it also builds a geometrically meaningful map.**

I. INTRODUCTION

In robotics, simultaneous localization and mapping (SLAM) is an effiecient way to map an unknown environment while estimating a robots pose within it. Reliable autonomous navigation requires accurate knowledge of the robots pose and the surrounding environment. Traditional approaches of SLAM rely on low-level and local geometric features such as points [1], lines [2], and surface patches [3] to reconstruct the metric 3-D structure of a scene. The complex indoor scenarios present significant challenges to these local-feature based approaches: the cluttered building leads the appearance-based visual features hard to be tracked by multi-view, it significantly degenerates the SLAM system based on graph optimization method [4]. Recently, [5] and [6] proposed high-level geometric features (cuboid, quadratic) to recognize the objects. The recognized object also benefits the state estimation to achieve a more robust and accurate state estimation. However, it is not enough for the robot to perform navigation tasks.



(a)

(b)

(c)

Fig. 1: Example result of geometric SLAM map with wireframes and layout planes of building.

On the other hand, scene understanding methods can provide more geometrically meaningful features for room reconstruction. In [7] the room layout is parameterized as 3D box, it then samples rays from vanishing points and using surface labels to refine the hypothesized room model. The most closest work to ours is [8], the author represented the room as a set of corners of building, and select the best subset corners by the orientation map. However, most of these methods cannot achieve real-time performance. There are many works [9], [10] which combined the scene understanding methods with state estimation.

* Corresponding author

[1] T Stone Robotics Institute, The Chinese University of Hong Kong, Hong Kong, China

[2] Department of Mechanical Engineering and Automation, Harbin Institute of Technology, Shenzhen 518055, China

They have demonstrated that parsing scene can significantly benefit the 3D reconstruction tasks. Moreover, the scene understanding methods seem capable of effortlessly generating more geometrically meaningful map among such structural environments by exploiting larger-scale (global or semi-global) structural features (such as the wireframe of the building and the large layout planes).

The goal of this paper is to take the advantage of scene understanding to improve the SLAM system in both state estimation and geometric mapping. We propose an effective scene understanding method, which samples a set of orthonormal corners where line segments intersect from a single RGB image and utilizes the depth map to select the best subset corners. According to the Manhattan assumption, we also infer the 3D wireframes and layout planes of buildings as global features for further uses in SLAM system. These global features provide additional geometric constraints to improve camera pose estimation as well as the geometrically meaningful 3D map. Our work makes the following contributions to the state of the art:

- We propose an efficient scene understanding method to detect wireframes and layout planes of building from RGB-D image, even if there are occlusions in the scene.
- We treat the wireframes and layout planes as global features and integrate them with points-based SLAM to improve the accuracy and robustness.
- A geometrically more meaningful map can be obtained from the proposed method.

In the following section II, we describe the single RGB-D image scene understanding, which provides line of wireframe and supposed plane measurements for SLAM. In Section III, we introduce the formulation of RGB-D SLAM with multi-feature measurements. Experiments on a public ICL-NUIM dataset is presented in Section IV. Finally, we conclude in Section V.

II. Single RGB-D image Scene Understanding

In indoor environments, most man-made structures have regular shapes, especially those containing lines, corners, and large planes. Building models can be generated by connecting line segments to create corners, and linking corners to create wireframes and layout planes. This section shows that we represent the building room as a set of geometric wireframe lines and layout planes. The goal is to infer the 3D positions of wireframe lines and planes from 2D images for further SLAM uses.

The process of the single image scene understanding can be summarized into three steps: 1). generating a set of building's corners from clustered line segments; 2). selecting the best subset of them according to the depth map; 3). the potential layout planes can be inferred through the corner characteristic. The details will be described in the following section.

A. Corner Proposal

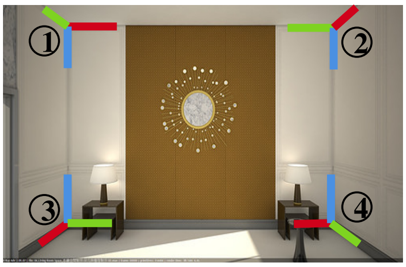
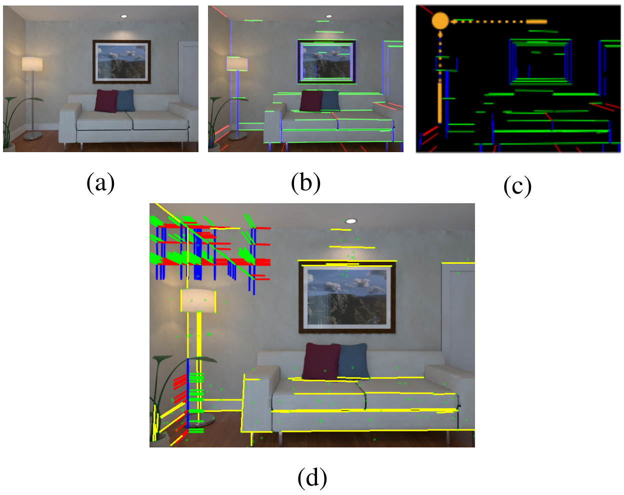


Fig. 2: Four types of corners of ordinary buildings. 1: Top left corner; 2: Top right Corner; 3: Bottom left corner; 4: Bottom right corner.


(a)

(b)

(c)

(d)

Fig. 3: Generating corner hypotheses.

Following [8], a room normally consists of four types corner, as shown in Fig. 2 The process of generating hypotheses is illustrated in Fig.3. We start by detecting line segments [11] and cluster them by three vanish points [12] (Fig. 3b). Then we use RANSAC to sample the clustered lines to form corners, as shown in Fig. 3c. A corner consists of three orthonormal lines, we only need two of them to define a corner. Different from the method in [8], in which sample of all possible corners are taken randomly. Instead, we primarily sample more stable top corners (1 and 2 types), where there are few occlusions. Based on the top corners, we can easily infer the corresponding bottom corners (3 and 4 types) (Fig. 3d).

B. Proposal Evaluation With Depth Map

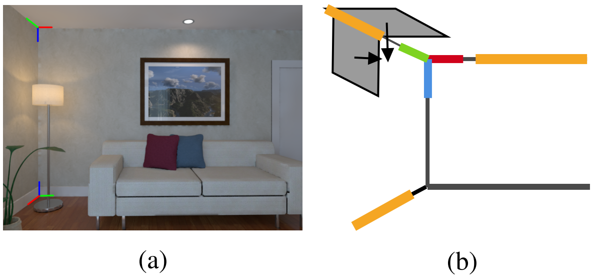

(a) (b)

Fig. 4: The best corners selected with depth map.

After sampling many corner candidates, we use depth map to cross-validate these candidates and select the best subset, as shown in Fig. 4a. There are several non real-time evaluation functions that have been proposed, such as ground boundary detection [13], surface labels [7], and orientation map [8]. We propose a more efficient evaluation method, which utilizes the depth map to fast filter out the sampled lines with small curvature. It means that the corners only can be formed by the lines of wireframe, which are the intersection lines between two orthonormal wall planes, as shown in Fig .4b.

C. Infer 3D Dense Layout Planes

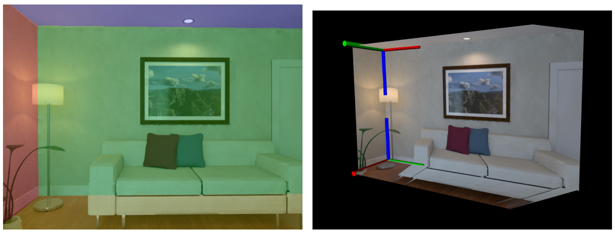

(a) Labeling map (b) 3D Layout Planes

Fig. 5: Infer 3D layout plane.

In indoor environments, most man-made structures have regular shapes, especially those have large enough plane features. Due to the limitation of depth detection from low cost sensors like RGB-D cameras, we can infer potential 3D layout planes by the information of extracted wireframe. Firstly, based on the Manhattan-assumption and selected corners, we can generate the labeling map of layout planes in 2D, as shown in Fig. 5a. Then, we reversely project the top corners (1 and 2 types) as top 3D wireframe junctions, and we also use the co-planar knowledge between the 3D wireframe junctions and layout planes to infer the parameters of planes. Due to always existing occlusions at the bottom, the depth information of bottom 3D wireframe junctions should be supposed by back-projection the corner's pixel onto the 3D inferred layout planes:

$$P = \frac{-d}{n^T\left(K^{-1}p\right)} K^{-1} p \qquad (1)$$

where the inferred layout plane can be represented as $\pi = (\mathbf{n}, d) \in \mathbb{R}^4$, and $\mathbf{n} \in \mathbb{R}^3$ and d are the normal vector and distance to origin respectively. K is the camera's calibration matrix, p and P are the pixel and 3D position respectively. Finally, we use the labeled pixels to back-projection them to corresponding planes and it generates a 3D dense layout planes, as shown in Fig. 5b.

III. RGB-D SLAM WITH SCENE UNDERSTANDING CONSTRAINTS

This section introduces the RGB-D SLAM using wireframes and layout planes to achieve globally accurate state estimation.

A. Fundamentals

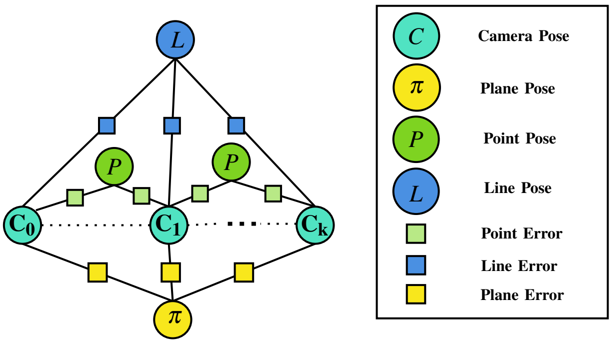


Fig. 6: Factor Graph

The factor graph of SLAM is shown in Figure 6. The circles and recetangles mean the nodes and edges in factor graph. As the afrontmention, the L lines of wireframe and the layout plane landmarks are observed by more camera node C_i. The representation of cost function for the tightly-coupled RGB-D SLAM with heterogeneous geometric features (points, lines, planes) is defined as following:

$$\mathbf{x}_i^* = \arg\min\Big(\sum_{p_k} \left\|\mathbf{E}_p(k)\right\|_{\Sigma_p} + \sum_{l_k} \|\mathbf{E}_l(k)\|_{\Sigma_l} \\ + \sum_{\pi_k} \|\mathbf{E}_\pi(k)\|_{\Sigma_\pi}\Big) \tag{2}$$

where $\mathbf{E}_p$, $\mathbf{E}_l$, $\mathbf{E}_\pi$ errors represent for camera w.r.t. visual point, camera w.r.t. line, and camera w.r.t. plane features respectively. Σ_p, Σ_l, and Σ_π are the corresponding information matrix of different error measurements. The detailed definitions of variables and errors terms are in the following section.

B. Geometric information: Point, Line, and Plane

From each frame image, we utilize geometric point, line, and plane measurements to tackle traking problem.

*1) **Point measurement:*** The point reprojection error constrains the minimum geometric error between a projected mappoint and a corresponding measured keypoint. Consider the kth feature that is observed in the ith image, the residual is defined as:

$$\left\|\mathbf{E}_p(k)\right\|_{\Sigma_p} = \left(\mathbf{x}^k - \pi\left(\mathbf{P}_{C_i}^k\right)\right)^T \Sigma_p \left(\mathbf{x}^k - \pi\left(\mathbf{P}_{C_i}^k\right)\right) \tag{3}$$

where π represents a projection function, $\mathbf{x}^k$ is the $\mathbf{k}th$ keypoint location in the image frame, $\mathbf{P}_{\mathrm{C_i}}^k$ is the $\mathbf{k}th$ mappoint in $\mathbf{i}th$ camera coordinates, the information matrix is Σ_p.

*2) **Line measurement:*** In our system, a 4-DoF straight line in 3D space is treated as an infinite line and we adopt two parameterizations for a 3D line as in [14]. One is the Plcker line coordinates $\mathbf{L} = (\mathbf{n}^T, \mathbf{d}^T)^T \in \mathbb{R}^6$ used for transformation and projection due to its geometric intuitive, where $\mathbf{n} \in \mathbb{R}^3$ is the normal of the plane determined by the line's endpoints and the coordinate origin, and $\mathbf{d} \in \mathbb{R}^3$ is the line direction. Another one is the orthonormal representation $(U, W) \in \mathbb{R}^4$ used for optimization and updating due to its compactness. The reprojection error is:

$$\|\mathbf{E}_l(l_k)\|_{\Sigma_l} = \mathbf{e}_l(l_k)^T \Sigma_l \mathbf{e}_l(l_k) \\ \mathbf{e}_l(l_i) = \begin{bmatrix} d\left(\mathbf{s}_{l_k}, \mathbf{l}_{l_k}\right) \\ d\left(\mathbf{e}_{l_k}, \mathbf{l}_{l_k}\right) \end{bmatrix}, \quad d(\mathbf{s}, 1) = \frac{\mathbf{s}^\top \mathbf{l}}{\sqrt{l_1^2 + l_2^2}} \tag{4}$$

where $\mathbf{s}_{l_k}$ and $\mathbf{e}_{l_k}$ are the endpoints of kth lines, respectively. $\mathbf{l}_{l_k}$ is the re-projection of lines on the image plane.

*3) **Plane measurement:*** We construct a global map that consists of plane features in the scenes. A 3-DoF plane is represented as an infinite plane. Similar with line, the over-parametrized homogeneous representation is $\pi = (\mathbf{n}^T, d)^T \in \mathbb{R}^4$ for transformation and projection, where $\mathbf{n} \in \mathbb{R}^3$ is the normal vector of the plane, and d is the nearest distance between plane and origin. We also utilize the spherical coordinate plane representation as in [14] $(cos\phi cos\theta, cos\phi sin\theta, d) \in \mathbb{R}^3$ to do plane updates during optimization, where ϕ and θ are the azimuth and elevation angles of the plane normal $\mathbf{n} \in \mathbb{R}^3$ respectively. The reprojection error is:

$$\|\mathbf{E}_\pi(k)\| = e_\pi(k)^T \Sigma_\pi e_\pi(k) \\ e_\pi(k) = \pi_c^k - \mathbf{T}_{cw}^{-\top} \pi_w^k \tag{5}$$

where π_w^k is the kth plane parameter in the world coordinate system and π_c^k is the corresponding plane observation in the camera coordinate system.

C. Data Association

The point association is intuitive because the point can be matched with their descriptor as used in [1]. Similar with point association, line also has LBD descriptor for matching [2]. For plane association, we use the following three geometry information for plane matching: the difference between plane normals, plane distance to each other, and projection overlapping between planes. Then the best match is selected based on a weighted sum of them.

IV. Experiments

A. Implementation details

We implemented our SLAM based on the feature point based ORB SLAM2 [1], augmented with our wireframes and layout planes. The graph optimization is impleted in g2o with L-M non-linear algorithm [15]. The whole system pipline as shown in Fig 7. For the final dense map generation, we reversely project pixels in the labeled image regions onto the layout plane landmarks.

B. Single Image Scene Understanding Result

We first show the single image scene understanding results. Some examples of building reconstruction are shown in Fig. 8. We can see that the scene understanding can obtain more geometric information of room compared with traditional RANSAC-based plane feature extration method [16]. Although the background wall has occlusion, our method can still provide a complete geometric point cloud of walls.

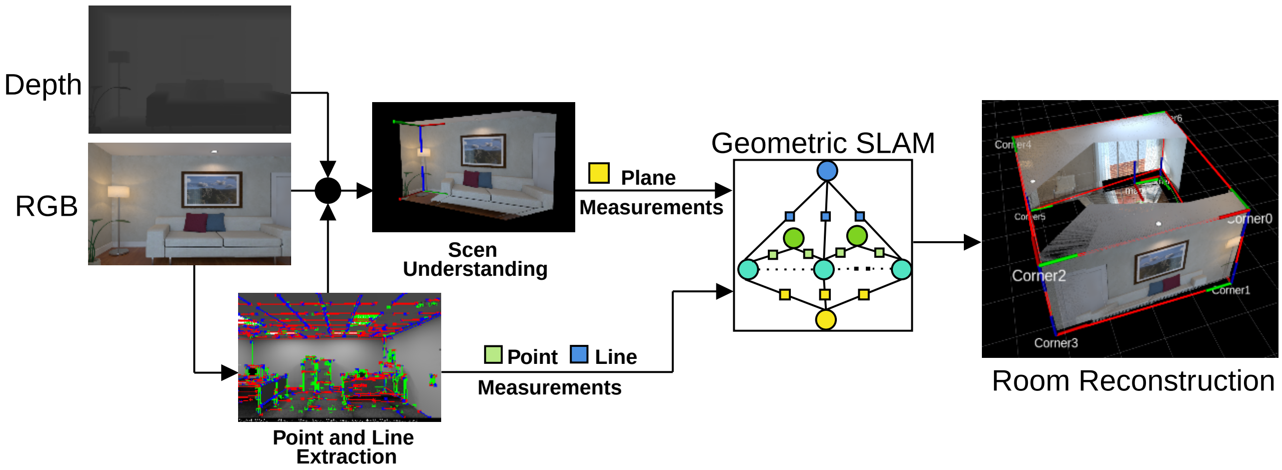


Fig. 7: System overview.

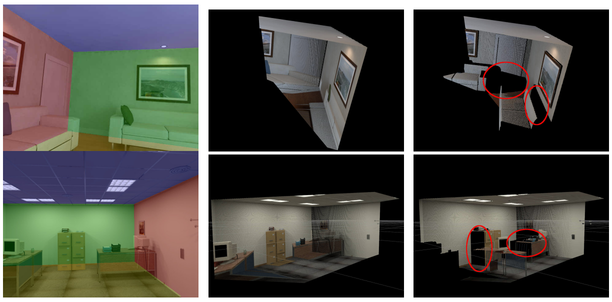


Fig. 8: Single image results. First column represents labelled map; Second column represents 3D layout planes; Third column is the plane detected by trandi-tional RANSAC extration.

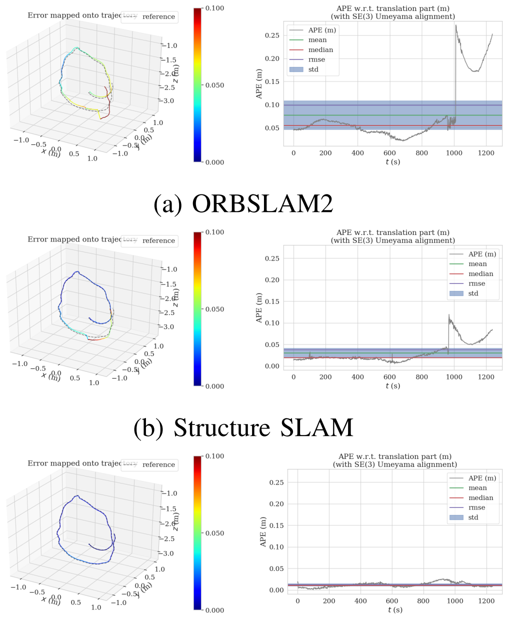
C. SLAM Result

We then evaluate the our SLAM tracking and mapping performance on public datasets ICL-NUIM [17].

Methods	ORBSLAM2	Structure SLAM	Ours
ICL living kt1	0.034	0.013	**0.012**
ICL living kt2	0.017	0.019	**0.012**
ICL office kt2	0.041	0.027	**0.024**
ICL office kt3	0.099	0.038	**0.013**

TABLE I: Absolute Trajectory Error (ATE) [m] in ICL-NUIM Datasets

1) Quantitative Result: In this experiment, we mainly evaluate the quantitative result of SLAM system. We show the trajectory accuracy of camera pose comparison with ORB SLAM2 [1] and Structure SLAM [18], shown in Fig. 9. The figure illustrates that our augmented SLAM system can efficiently bound the drift with the travel length increasing. More comparison results are summarized in Table. I, we can see that in most of the scenarios, the augmented line and plane landmark constraints in the SLAM system (Structure SLAM [18] and ours) improve the camera pose estimation. The reason is that the line and



(a) ORBSLAM2

(b) Structure SLAM

(c) Ours

Fig. 9: The trajectory results on ICL-NUIM Office kt3 dataset. The left column shows the estimated trajectory against with the ground truth. The right column shows the Absolute Trajectory Error (ATE).

plane features can provide more rotation constraints to reduce the drift [19]. In addition, our SLAM system has lower drift than Structure SLAM [18], due to global feature constraints (wireframes and layout planes of building). The global features have long-range visibility properties, our SLAM system can associate more frames with the wireframe and layout plane landmarks to reduce the final drift.

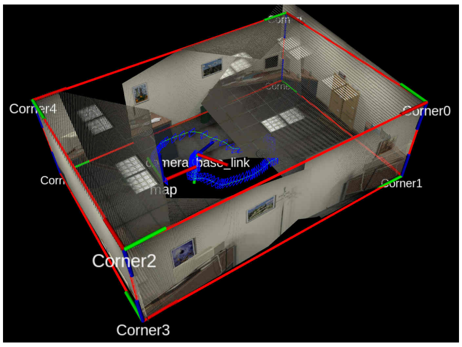


Fig. 10: The room reconstruction result on ICl-NUIM office room.

		Length	Width	Height
Living Room	Estimated	4.958	4.928	2.528
	Ground Truth	4.980	4.940	2.500
	Error	0.022	0.012	0.028
Office Room	Estimated	7.414	5.051	2.402
	Ground Truth	7.400	5.000	2.400
	Error	0.014	0.051	0.002

TABLE II: The room reconstruction error [m].

2) Quality Result: In this experiment, we mainly evaluate the 3D mapping accuracy by comparing our estimated 3D building model with the ground truth. We evaluate the estimated living room model (Fig. 1) and office room (Fig. 10). The numerical results are shown in Table 2. Although the estimated room's scale has around 2 cm error, the map with wireframe and layout planes can provide enough geometric information for robot navigation.

V. CONCLUSIONS

In this paper, we have presented a low-drift RGB-D SLAM system augmented by efficient scene understanding method. For cluttered scenarios involving occlusions, we propose a fast scene understanding method to detect wireframe and layout planes of buildings. The wireframe and layout planes also are treated as global features to integrate into the point feature based SLAM system for accuracy and robustness. Compared to low-level features based SLAM, our SLAM system can build a more geometrically meaningful map for robot navigation.

In the future, we will take more general corner types (e.g, Convex corner, occlusion corner, and corridor turning corner) into account to produce a more complete geometric map.

References

[1] R. Mur-Artal and J. D. Tardós, "Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras," *IEEE transactions on robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.

[2] A. Pumarola, A. Vakhitov, A. Agudo, A. Sanfeliu, and F. Moreno-Noguer, "Pl-slam: Real-time monocular visual slam with points and lines," in *2017 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2017, pp. 4503–4508.

[3] 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*. IEEE, 2013, pp. 5182–5189.

[4] I. Aloise, B. Della Corte, F. Nardi, and G. Grisetti, "Systematic handling of heterogeneous geometric primitives in graph-slam optimization," *IEEE Robotics and Automation Letters*, vol. 4, no. 3, pp. 2738–2745, 2019.

[5] S. Yang and S. Scherer, "Cubeslam: Monocular 3-d object slam," *IEEE Transactions on Robotics*, vol. 35, no. 4, pp. 925–938, 2019.

[6] L. Nicholson, M. Milford, and N. Sünderhauf, "Quadricslam: Dual quadrics from object detections as landmarks in object-oriented slam," *IEEE Robotics and Automation Letters*, vol. 4, no. 1, pp. 1–8, 2018.

[7] V. Hedau, D. Hoiem, and D. Forsyth, "Recovering the spatial layout of cluttered rooms," in *2009 IEEE 12th International Conference on Computer Vision*, 2009, pp. 1849–1856.

[8] D. C. Lee, M. Hebert, and T. Kanade, "Geometric reasoning for single image structure recovery," in *2009 IEEE conference on computer vision and pattern recognition*. IEEE, 2009, pp. 2136–2143.

[9] S. Yang, Y. Song, M. Kaess, and S. Scherer, "Pop-up slam: Semantic monocular plane slam for low-texture environments," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 1222–1229.

[10] S. Yang and S. Scherer, "Monocular object and plane slam in structured environments," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3145–3152, 2019.

[11] R. G. Von Gioi, J. Jakubowicz, J.-M. Morel, and G. Randall, "Lsd: a line segment detector," *Image Processing On Line*, vol. 2, pp. 35–55, 2012.

[12] J.-P. Tardif, "Non-iterative approach for fast and accurate vanishing point detection," in *2009 IEEE 12th International Conference on Computer Vision*. IEEE, 2009, pp. 1250–1257.

[13] S. Yang, D. Maturana, and S. Scherer, "Real-time 3d scene layout from a single image using convolutional neural networks," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 2183–2189.

[14] Y. He, J. Zhao, Y. Guo, W. He, and K. Yuan, "Pl-vio: Tightly-coupled monocular visual–inertial odometry using point and line features," *Sensors*, vol. 18, no. 4, p. 1159, 2018.

[15] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "g 2 o: A general framework for graph optimization," in *2011 IEEE International Conference on Robotics and Automation*. IEEE, 2011, pp. 3607–3613.

[16] R. Schnabel, R. Wahl, and R. Klein, "Efficient ransac for point-cloud shape detection," in *Computer graphics forum*, vol. 26, no. 2. Wiley Online Library, 2007, pp. 214–226.

[17] A. Handa, T. Whelan, J. McDonald, and A. J. Davison, "A benchmark for rgb-d visual odometry, 3d reconstruction and slam," in *2014 IEEE international conference on Robotics and automation (ICRA)*. IEEE, 2014, pp. 1524–1531.

[18] Y. Li, N. Brasch, Y. Wang, N. Navab, and F. Tombari, "Structure-slam: Low-drift monocular slam in indoor environments," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 6583–6590, 2020.

[19] P. Kim, B. Coltin, and H. J. Kim, "Visual odometry with drift-free rotation estimation using indoor scene regularities." in *BMVC*, vol. 2, no. 6, 2017, p. 7.