Plane-Edge-SLAM: Seamless Fusion of Planes and Edges for SLAM in Indoor Environments

Qinxuan Sun, Jing Yuan, *Member, IEEE*, Xuebo Zhang, *Senior Member, IEEE*, and Feng Duan, *Member, IEEE*

***Abstract*— Planes and edges are attractive features for simultaneous localization and mapping (SLAM) in indoor environments because they can be reliably extracted and are robust to illumination changes. However, it remains a challenging problem to seamlessly fuse two different kinds of features to avoid degeneracy and accurately estimate the camera motion. In this article, a plane-edge-SLAM system using an RGB-D sensor is developed to address the seamless fusion of planes and edges. Constraint analysis is first performed to obtain a quantitative measure of how the planes constrain the camera motion estimation. Then, using the results of the constraint analysis, an adaptive weighting algorithm is elaborately designed to achieve seamless fusion. Through the fusion of planes and edges, the solution to motion estimation is fully constrained, and the problem remains well-posed in all circumstances. In addition, a probabilistic plane fitting algorithm is proposed to fit a plane model to the noisy 3-D points. By exploiting the error model of the depth sensor, the proposed plane fitting is adaptive to various measurement noises corresponding to different depth measurements. As a result, the estimated plane parameters are more accurate and robust to the points with large uncertainties. Compared with the existing plane fitting methods, the proposed method definitely benefits the performance of motion estimation. The results of extensive experiments on public data sets and in real-world indoor scenes demonstrate that the plane-edge-SLAM system can achieve high accuracy and robustness.**

***Note to Practitioners*—This article is motivated by the robust localization and mapping for mobile robots. We suggest a novel simultaneous localization and mapping (SLAM) approach fusing the plane and edge features in indoor scenes (plane-edge-SLAM). This newly proposed approach works well in the textureless or dark scenes and is robust to the sensor noise. The experiments are carried out in various indoor scenes for mobile robots, and the results demonstrate the robustness and effectiveness of the proposed framework. In future work, we will address the fusion of other high-level features (for example, 3-D lines) and the active exploration of the environments.**

***Index Terms*— Plane fitting, RGB-D camera, six-degree-of-freedom (6-DoF) camera motion estimation.**

I. Introduction

SIMULTANEOUS localization and mapping (SLAM) is one of the fundamental problems for mobile rbots navigating in indoor environments [1]–[4]. Planes are inherently attractive features for SLAM in indoor environments that mainly include man-made structures. Compared with point features, planes are more robust to illumination changes and are demonstrated to perform well in low-texture regions [5], [6]. They can also provide structural and semantic information of the environments, which can be used in applications, such as path planning [7], [8], target tracking [9], place recognition [10], [11], and robot manipulation [12].

Many researchers have exploited the advantages of planes in localization and navigation of mobile robots in indoor environments [13]–[15]. The weakness of estimating the camera motion using only plane features lies in the possibility of ill-posedness because there exist spatial configurations of planes that cannot fully constrain the solution. The degenerate configurations of planes have been studied in [5], [10], [14], and [16]. In our previous work [5], the degeneracy was detected by singular value decomposition (SVD) of a matrix constructed by the plane normals. The singularity in the solution to pose estimation was then eliminated by a scan-matching process in the plane parameter space. The degenerate configurations were also discussed in [10], where the five-degree-of-freedom (5-DoF) pose hypothesis was generated sequentially from planar surface segments using the extended Kalman filter, under the assumption that typical indoor scenes contain at least two dominant nonparallel planar surfaces. The other translational DoF was determined by a voting scheme using planar surfaces and line segments. However, the assumption does not always hold true because, in some circumstances, the robot only observes parallel planes in indoor scenes. In [16], a random sample consensus (RANSAC) framework was adopted to generate camera pose hypotheses using both point-to-point and plane-to-plane correspondences. The nondegenerate configurations of points and planes were discussed. In summary, the aforementioned works only empirically considered how much a plane contributed to the estimation of the camera pose.

Manuscript received April 18, 2019; revised July 13, 2020 and September 2, 2020; accepted October 19, 2020. Date of publication November 4, 2020; date of current version October 6, 2021. This article was recommended for publication by Associate Editor C. Yang and Editor K. Saitou upon evaluation of the reviewers' comments. This work was supported in part by the Natural Science Foundation of China under Grant 62073178 and Grant 61573196, in part by the Fundamental Research Funds for the Central Universities, Nankai University, under Grant 63206026, and in part by the major basic research projects of the Natural Science Foundation of Shandong Province under Grant ZR2019ZD07. *(Corresponding author: Jing Yuan.)*

The authors are with the College of Artificial Intelligence, Nankai University, Tianjin 300350, China, and also with the Tianjin Key Laboratory of Intelligent Robotics, Nankai University, Tianjin 300350, China (e-mail: nkyuanjing@gmail.com).

This article has supplementary downloadable material available at https://ieeexplore.ieee.org, provided by the authors. This includes three multimedia MP4 format videos, which show the experiments presented in Sections VI-E and VI-F. This material is 24.7 MB in size.

Color versions of one or more of the figures in this article are available online at https://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TASE.2020.3032831

In the work of [14], an observability analysis was performed for the linearized-aided inertial navigation system (INS) with heterogeneous geometric features (point, lines, and planes). However, the analysis is based on an INS framework and cannot be directly applied to the scan-alignment for a depth sensor. To the best of our knowledge, a thorough quantitative analysis on how a single plane or a set of planes constrain the pose estimation of an RGB-D camera has never been given.

In case the extracted planes are insufficient to fully constrain the motion estimation, additional information is required. In [17], an inertial measurement unit (IMU) was used to deal with the degeneracy problem in RGB-D SLAM. Though the IMU information can impose additional constraints on the motion estimation, the calibration of the two sensors and the drift of IMU data need to be considered. Another way to handle the degeneracy problem is to combine other features. The most widely used one is the visual feature extracted from RGB images [13], [18], [19], which is a preferable choice in the textured scenes under good lighting conditions. In the work of [18], point features and planar segments were combined in the camera tracking and map construction. In [13], planes were used together with the point, and line features to implement scan registration, and the back-end optimization was implemented in [19] to achieve a complete SLAM system. However, in textureless scenes, the visual features can hardly be extracted. Furthermore, the extraction of visual features is very sensitive to the changes in illumination conditions and fails to work in low-light conditions. In many applications, mobile robots are required to work in dark environments, such as patrolling a building at night and working underground in a coal mine. Therefore, the ability to navigate in dark environments is indispensable for mobile robots.

Apart from the point feature, the edge is another kind of popular feature. The edges present structure information of the environment and have been shown to have good performance in RGB-D SLAM or visual odometry (VO) systems [20]–[22]. The Canny-VO system proposed in [20] achieved the robust tracking by a 3-D–2-D edge alignment based on the nearest neighbor fields. In [21], a robust edge-based SLAM system was built, and a local sliding window optimization over the keyframes was used to refine the depth of edges, the calibration parameters and the camera poses. The edge-based RGB-D SLAM system in [22] was proposed for dynamic environments, and a static weighting method was designed for edge-points to indicate the likelihood of each point being part of the static scene. The above-mentioned works demonstrate the efficiency of edge features. For an RGB-D sensor, two types of edges are available, i.e., the edges extracted from RGB images and the ones from depth images. Similar to the image feature points, the RGB edges still suffer from sensitivity to the changes in illumination conditions. Therefore, we use the depth edge information [23] to disambiguate the solutions when planes cannot fully constrain the motion estimation. Note that, calculating the camera poses using the edges extracted from two successive frames is essentially a 3-D curve registration problem. It has been illustrated in [24] that indiscriminately using all the points measured from a curve for registration will inordinately slow down the convergence of estimation or even find a wrong solution. The works of [25] and [26] analyzed the stability of the estimated transformation and selected the points to maximize stability. A normal-space sampling (NSS) method was proposed for the widely used iterative closest point (ICP) algorithm in [27]. The rationale of NSS was to sample enough constraints in the normal space to determine all the components of transformation. The approach in [24] extended the NSS and the proposed dual-NSS (DNSS) to sample points in both translational and rotational normal spaces such that the translational and rotational components are properly constrained. For all the aforementioned studies, the points cannot be quantitatively evaluated and selected to constrain some specific dimensions of the 6-D space of rigid transformation. Furthermore, when the point features are combined with other kinds of features, the constraints provided by each kind of feature need to be taken into account separately in the feature selection, which has not been addressed in previous studies. As a result, the existing methods are not capable of dealing with our situation, in which some of the components are already strongly constrained by plane features, while the others are unconstrained. To fully fuse the edge information with the planes, the edge-points need to be thoroughly evaluated and selected to determine the components of motion in the subspace that cannot be constrained by planes. This problem has remained unexplored in previous methods.

In this article, we develop an RGB-D SLAM system to address the aforementioned issues. A seamless fusion of planes and edges is proposed for an accurate and robust motion estimation, which is achieved based on constraint analysis and an adaptive weighting algorithm. The constraint analysis is performed to analyze how the planes constrain motion estimation. The constrained subspace of motion is explicitly represented, and a quantitative measure of the constraint strength provided by planes is given. Using the results of constraint analysis, an adaptive weighting algorithm is designed to automatically assign different weights to edge-points according to their constraints on the motion in the subspace that the planes cannot constrain. In addition, a probabilistic plane fitting algorithm is proposed, which improves the accuracy of the fitted plane model and further benefits the plane-based motion estimation. The error model of a depth sensor is exploited to consider the effect of measurement noises varying with the location of 3-D points. In this way, the plane fitting process is less affected by the points with large measurement noises, and the estimated planes are more fitted to the more accurately measured points.

The original contributions are summarized as follows.

1) A seamless fusion of planes and edges is proposed to fully constrain the motion estimation. The constraints provided by both planes and edges are utilized in the fusion process. The seamless fusion makes the problem of motion estimation remain well-posed in all circumstances and also gives a new perspective to the feature fusion problem.
2) An analysis is performed on the constraints of planes on the estimation of the camera motion, and an explicit representation of the constrained motion subspace is derived. A quantitative measure of the constraint strength on a given

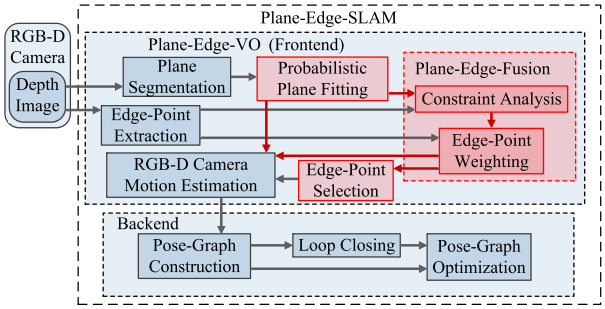


Fig. 1. Overview of the plane-edge-SLAM system. The red boxes contain main contributions of this article.

camera motion is presented. The analysis results can be used in not only the fusion of planes and edges but also the identification of singular solutions to the motion estimation.

3) A probabilistic plane fitting algorithm is proposed to fit a plane model using noisy points extracted from a depth image. The estimated plane model is more adaptive to measurement noises, which benefits the plane-based estimation of the camera motion.

The rest of the article is organized as follows. The system overview is presented in Section II. The probabilistic plane fitting algorithm is developed in Section III. The seamless fusion of planes and edges is presented in Section IV. The loop closing and pose-graph optimization are executed in Section V. Thorough experimental evaluations are presented in Section VI. Conclusions are drawn in Section VII.

II. System Overview

The architecture overview of the plane-edge-SLAM system is shown in Fig. 1, which is composed of two parts: the frontend (plane-edge-based VO and plane-edge-VO) and the backend. In the plane-edge-VO, planar segments and edges are extracted by the approaches proposed in [5] and [23], respectively. It needs to be pointed out that both the two features are extracted from depth images captured by an RGB-D camera, as shown in Fig. 1. In the probabilistic plane fitting module, a plane model is fitted to the noisy points in each planar segment. The planes and edges are then fused in the plane-edge-fusion module, which consists of two submodules: the constraint analysis and the edge-point weighting. In the constraint analysis module, an explicit representation of the constrained motion subspace is derived. Using the results of constraint analysis, the edge-point weighting module adaptively assigns different weights to edge-points. The output of edge-point weighting is fed into two modules afterward. The first is the edge-point selection module, which excludes edge-points with small weights. The second is the RGB-D camera motion estimation module, in which the weights are used to balance two kinds of residuals in the cost function.

The backend constructs the pose graph in the process of incremental motion estimation and searches for possible loop closures in the pose graph. Once a loop closure is detected, an additional constraint is added, and then the pose graph is optimized to achieve a complete SLAM system.

III. Probabilistic Plane Fitting

In order to extract a plane from the point cloud captured by an RGB-D camera, two steps are involved, i.e., segmenting planar regions from the point cloud and fitting an infinite-plane model to each planar region. The fitted plane is essentially a high-level representation of the raw data points, and its parameters (the normal of the plane and the vertical distance from the origin to the plane) are used in the calculation of camera motion. Thus, the accuracy of plane fitting has a significant impact on the results of motion estimation. In this article, segmentation of the planar regions is performed by the statistical information grid (STING)-based method proposed in our previous work [5].

The most widely used method to fit a plane model to a set of noisy points is the least squares (LS) method [28]–[30]. The cost function of the LS method is the sum of vertical distances from each point to the plane. In LS fitting, the point-to-plane distance of each point contributes equally to the cost function. In other words, the confidence in the measurement of each point is assumed to be the same regardless of the location of the point. However, as verified in previous researches [31]–[33], for a depth sensor, uncertainty of the measurement of a 3-D point varies along with its location. If the points with a large uncertainty are treated equally with those with a small uncertainty, the estimated plane model will be inaccurate, which will further decrease the precision of the motion estimate. Therefore, when fitting a plane, two aspects should be considered. First, the estimated plane model should be more fitted to the points that are measured more accurately. Second, the uncertainty needs to be propagated from the depth and pixel measurements to the point-to-plane distances. In this section, a probabilistic plane fitting method is designed to address the two aspects.

We first segment a set of 3-D points that comes from a planar surface in the scene from a point cloud by the STING-based plane segmentation method [5]. The point set is denoted by $\{\boldsymbol{p}_{\pi j}\}_{j=1,\ldots,N_{p\pi}}$, where $\boldsymbol{p}_{\pi j} \in \mathbb{R}^3$ is the location of a 3-D point on the planar surface and $N_{p\pi}$ represents the number of points. Assuming that the measurement noise of $\boldsymbol{p}_{\pi j}$ follows a zero-mean Gaussian distribution with covariance $\boldsymbol{C}_{p_{\pi j}}$, which is propagated from the variances of the depth measurement and pixel coordinates, respectively, to $\boldsymbol{C}_{p_{\pi j}}$

$$\begin{aligned}\boldsymbol{C}_{p_{\pi j}} = {} & (\boldsymbol{K}^{-1}\tilde{\boldsymbol{u}}_j)\sigma_{zj}^2(\boldsymbol{K}^{-1}\tilde{\boldsymbol{u}}_j)^T \\ & + (z_j\boldsymbol{k}_1)\sigma_{uj}^2(z_j\boldsymbol{k}_1)^T + (z_j\boldsymbol{k}_2)\sigma_{vj}^2(z_j\boldsymbol{k}_2)^T \quad (1)\end{aligned}$$

where $\boldsymbol{K}$ is the intrinsic matrix of a depth camera and $\boldsymbol{k}_i, i = 1, 2, 3$ is the ith column of $\boldsymbol{K}^{-1}$. $\tilde{\boldsymbol{u}}_j$ is the homogeneous representation of the pixel coordinates $\boldsymbol{u}_j = [u_j, v_j]^T$ corresponding to $\boldsymbol{p}_{\pi j}$ and $\sigma_{uj}^2, \sigma_{vj}^2$ are the variances of u_j, v_j, respectively (σ_{uj} and σ_{vj} are both set to 1/2 pixel in the experiments). z_j and σ_{zj}^2 are the depth value of the point $\boldsymbol{p}_{\pi j}$ and its variance, respectively. For different kinds of depth sensors, e.g., depth sensors based on structured light (SL) [33], [34] and time-of-flight [32], [35], the measurement noise is modeled differently and σ_{zj}^2 is assigned accordingly. It is worth pointing out that the error model of any kind of depth sensor can be involved in the proposed probabilistic

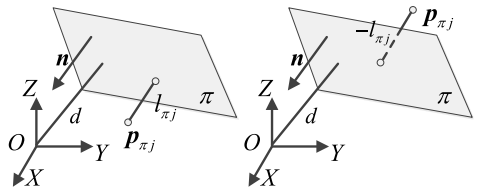


Fig. 2. Signed distance $l_{\pi j}$ from a 3-D point $\boldsymbol{p}_{\pi j}$ to the plane $\boldsymbol{\pi}$. $l_{\pi j}$ equals zero when $\boldsymbol{p}_{\pi j}$ lies exactly on $\boldsymbol{\pi}$. (Left) $l_{\pi j}$ is positive when $-\boldsymbol{n}^T\boldsymbol{p}_{\pi j} < d$, i.e., $\boldsymbol{p}_{\pi j}$ lies at the side of the plane $\boldsymbol{\pi}$, where the origin of the coordinate frame lies. (Right) Otherwise, $l_{\pi j}$ is negative.

plane fitting method, which does not rely on specific models of measurement noises. In our experiments, the error model proposed in [33] is adopted for the quantitative evaluations in Sections VI-A–VI-E, because the TUM data set is collected using the SL-based depth sensor [36] and the depth noise of Microsoft Kinect 1.0 is simulated in the ICL-NUIM data set [37]. In the real-world experiments in Section VI-F, we use the error model proposed in [35] for the Microsoft Kinect 2.0.

The fitted plane is denoted by $\boldsymbol{\pi} = [\boldsymbol{n}^T, d]^T$, where $\boldsymbol{n}$ is the unit normal of the plane (pointing to the origin of the camera coordinate frame) and d is the vertical distance from the origin to the plane. Define the signed distance $l_{\pi j}$ from $\boldsymbol{p}_{\pi j}$ to $\boldsymbol{\pi}$ as shown in Fig. 2

$$l_{\pi j} = \boldsymbol{n}^T\boldsymbol{p}_{\pi j} + d. \tag{2}$$

$l_{\pi j}$ follows a zero-mean Gaussian distribution with variance

$$\sigma_{l\pi j}^2 = \frac{\partial l_{\pi j}}{\partial \boldsymbol{p}_{\pi j}}^T \boldsymbol{C}_{p_{\pi j}} \frac{\partial l_{\pi j}}{\partial \boldsymbol{p}_{\pi j}} = \boldsymbol{n}^T\boldsymbol{C}_{p_{\pi j}}\boldsymbol{n}. \tag{3}$$

The squared Mahalanobis distance $D_l^2(l_{\pi j}, \boldsymbol{n}, d)$ of $l_{\pi j}$ is

$$D_l^2(l_{\pi j}, \boldsymbol{n}, d) = \frac{l_{\pi j}^2}{\sigma_{l\pi j}^2} = \frac{(\boldsymbol{n}^T\boldsymbol{p}_{\pi j} + d)^2}{\boldsymbol{n}^T\boldsymbol{C}_{p_{\pi j}}\boldsymbol{n}}. \tag{4}$$

The estimate of $\boldsymbol{\pi} = [\boldsymbol{n}^T, d]^T$ is computed by minimizing

$$E(\boldsymbol{n}, d) = \sum_{j=1}^{N_{p\pi}} D_l^2(l_{\pi j}, \boldsymbol{n}, d). \tag{5}$$

By taking the partial derivative of $E(\boldsymbol{n}, d)$ with respect to d and setting it to zero, the estimate of d can be obtained by

$$d^* = -\boldsymbol{n}^T\boldsymbol{p}_G(\boldsymbol{n}) \tag{6}$$

where $\boldsymbol{p}_G(\boldsymbol{n})$ is the weighted centroid of all the points.

$$\boldsymbol{p}_G(\boldsymbol{n}) = \frac{\sum_{j=1}^{N_{p\pi}} c_j(\boldsymbol{n})\boldsymbol{p}_{\pi j}}{\sum_{j=1}^{N_{p\pi}} c_j(\boldsymbol{n})} \tag{7}$$

$$c_j(\boldsymbol{n}) = (\boldsymbol{n}^T\boldsymbol{C}_{p_{\pi j}}\boldsymbol{n})^{-1}. \tag{8}$$

By substituting (6) into (5), $E(\boldsymbol{n}, d)$ can be rewritten as

$$E(\boldsymbol{n}) = \boldsymbol{n}^T\boldsymbol{S}(\boldsymbol{n})\boldsymbol{n} \tag{9}$$

$$\boldsymbol{S}(\boldsymbol{n}) = \sum_{j=1}^{N_{p\pi}} c_j(\boldsymbol{n})(\boldsymbol{p}_{\pi j} - \boldsymbol{p}_G(\boldsymbol{n}))(\boldsymbol{p}_{\pi j} - \boldsymbol{p}_G(\boldsymbol{n}))^T. \tag{10}$$

From a geometrical point of view, $\boldsymbol{S}(\boldsymbol{n})$ is a weighted scatter matrix with a weighted centroid $\boldsymbol{p}_G(\boldsymbol{n})$, which gives information about the dispersion of the noisy points around $\boldsymbol{p}_G(\boldsymbol{n})$. The weight $c_j(\boldsymbol{n})$, as defined in (8), is the reciprocal of $\sigma_{l\pi j}^2$, which is propagated from the measurement covariance $\boldsymbol{C}_{p_{\pi j}}$. From (8) and (10), we know that a point with a large uncertainty corresponds to a small weight $c_j(\boldsymbol{n})$, and thus, has a small influence on the minimization of $E(\boldsymbol{n})$. With $\boldsymbol{n}$ being a unit vector, the minimum of (9) is obtained when $\boldsymbol{n}$ equals the eigenvector of $\boldsymbol{S}(\boldsymbol{n})$ corresponding to the smallest eigenvalue. However, the minimization of (9) cannot be solved analytically because $\boldsymbol{S}(\boldsymbol{n})$ is a function of $\boldsymbol{n}$. When $c_j(\boldsymbol{n})$ is set to 1, the minimization of $E(\boldsymbol{n})$ degrades to the general LS problem.

Although the minimization of (9) can be iteratively solved by nonlinear optimization, it is time consuming and likely to get trapped into local minima. In addition, because $\boldsymbol{n}$ is constrained on the unit sphere in $\mathbb{R}^3$, the optimization needs further constraints on the variable. Therefore, we substitute $S(\boldsymbol{n})$ in (9) with $S(\boldsymbol{n}_{\text{LS}})$, where $\boldsymbol{n}_{\text{LS}}$ is the LS estimate of $\boldsymbol{n}$, i.e., the eigenvector corresponding to the smallest eigenvalue of $\boldsymbol{S}_{\text{LS}}$

$$\boldsymbol{S}_{\text{LS}} = \sum_{j=1}^{N_{p\pi}} (\boldsymbol{p}_{\pi j} - \boldsymbol{p}_{G\_\text{LS}})(\boldsymbol{p}_{\pi j} - \boldsymbol{p}_{G\_\text{LS}})^T \tag{11}$$

$$\boldsymbol{p}_{G\_\text{LS}} = \frac{1}{N_{p\pi}} \sum_{j=1}^{N_{p\pi}} \boldsymbol{p}_{\pi j}. \tag{12}$$

Then, the solution $\boldsymbol{n}^*$ can be obtained by

$$\boldsymbol{n}^* = \arg\min_{n}\ \boldsymbol{n}^T\boldsymbol{S}(\boldsymbol{n}_{\text{LS}})\boldsymbol{n}. \tag{13}$$

Since $S(\boldsymbol{n}_{\text{LS}})$ is independent of $\boldsymbol{n}$, (13) can be solved analytically and the solution $\boldsymbol{n}^*$ is the eigenvector corresponding to the smallest eigenvalue of $\boldsymbol{S}(\boldsymbol{n}_{\text{LS}})$. Although we may sacrifice a little optimality due to the substitution of $\boldsymbol{S}(\boldsymbol{n})$ with $\boldsymbol{S}(\boldsymbol{n}_{\text{LS}})$, it yields a better solution than the LS fitting and much faster performance than the iterative method. As aforementioned, $\boldsymbol{S}(\boldsymbol{n}_{\text{LS}})$ is a weighted scatter matrix of the measured points and $c_j(\boldsymbol{n}_{\text{LS}})$ is computed by the measurement covariance $\boldsymbol{C}_{p_{\pi j}}$ as well as the LS estimate $\boldsymbol{n}_{\text{LS}}$ of the normal. The point with a small uncertainty along the $\boldsymbol{n}_{\text{LS}}$ direction will be assigned a large weight, as in (8). Therefore, the plane estimated by our algorithm will be more fitted to the points with small uncertainties along the $\boldsymbol{n}_{\text{LS}}$ direction and less affected by points with large measurement noises. In comparison, in the LS method, all the measured points are treated equally in the computation of $\boldsymbol{S}_{\text{LS}}$, as in (11). The estimation result is prone to be inaccurate due to the influence of points with large measurement noises.

The uncertainty of the plane parameters can be represented by the covariance estimated by the inverse of the Hessian matrix of $E(\boldsymbol{n})$ evaluated at $\boldsymbol{n}^*, d^*$ [38]

$$\boldsymbol{C}_{\pi}^{-1} = \left[\begin{array}{cc} \dfrac{\partial^2 E}{\partial \boldsymbol{n}^2} & \dfrac{\partial^2 E}{\partial d \partial \boldsymbol{n}} \\ \dfrac{\partial^2 E}{\partial \boldsymbol{n} \partial d} & \dfrac{\partial^2 E}{\partial d^2} \end{array}\right]\Bigg|_{n^*, d^*} = \sum_{j=1}^{N_{p\pi}} c_j \left[\begin{array}{cc} \boldsymbol{p}_{\pi j}\boldsymbol{p}_{\pi j}^T & \boldsymbol{p}_{\pi j} \\ \boldsymbol{p}_{\pi j}^T & 1 \end{array}\right]. \tag{14}$$

IV. Seamless Fusion of Planes and Edges

In this section, the seamless fusion of planes and edges is presented. The motion of the RGB-D camera between two successive frames is estimated by scan alignment fusing the information of planes and edges, as presented in Section IV-A.

Although the planes can be reliably extracted and associated, the degeneracy in the plane-based motion estimation is unlikely to be avoided because of the narrow field of view and the limited depth range of the RGB-D camera [5]. The plane-based motion estimation problem may be ill-posed for some spatial configurations of planes (for instance, all the extracted planes are parallel). Thus, an explicit representation for the constrained subspace of camera motion estimate and a quantitative measure of the constraint strength on a given motion are essential prerequisites to fully and seamlessly fuse the planes with other features to fulfil the motion estimation. To this end, a thorough analysis of how the planes constrain the motion estimation is presented in Section IV-B.

In our method, the edges [23] are used to disambiguate the motion estimation when the planes cannot provide sufficient constraints. Two types of edges extracted from the depth images in [23] are used in this article, i.e., the occluding edges and the high curvature edges. Note that the RGB edges extracted from the images in [23] are also suitable to be used in our method in good lighting conditions. However, similar to the visual feature points, the RGB edges are sensitive to changes in illumination conditions and are not able to work in the dark indoor environments. Hence, we only use depth edges in our implementation. Then, a weight that is adaptively assigned to each edge-point is computed in Section IV-C fusing the information of planes and edges.

A. Plane-Edge-Fusion-Based Scan Alignment

The camera motion between two successive frames is calculated via an ICP-like scan alignment, in which two different kinds of primitives, i.e., planes and edge-points, are involved. The residual errors of the two kinds of primitives are defined in different spaces and are inappropriate to be treated equally in the overall cost function. In our method, the contribution of each edge-point to the cost function is tuned adaptively according to the constraints provided by planes. Thus, the information of edges is fused with the planes to determine all the components of camera motion. In each iteration, the planes and edge-points observed in the current frame are associated with those in the reference frame, respectively, using the nearest-neighbor approach [39], [40]. Then, a transformation between the two successive frames is solved by minimizing a cost function composed of residual errors between each primitive and its correspondence. Specifically, the matched planes are denoted by $\{{}^c\boldsymbol{\pi}_i, {}^r\boldsymbol{\pi}_i\}_{i=1,\ldots,N_\pi}$ and matched edge-points by $\{{}^c\boldsymbol{p}_k, {}^r\boldsymbol{p}_k\}_{k=1,\ldots,N_p}$, where $\boldsymbol{\pi}_i$ represent the ith plane, $\boldsymbol{p}_k$ represents the coordinates of the kth edge-point, the presuperscripts c and r denote the current and reference frames, respectively, and N_π and N_p are numbers of the plane and edge-point pairs, respectively. The camera motion is represented by $\boldsymbol{\xi} = [\boldsymbol{t}^T, \boldsymbol{\omega}^T]^T \in \mathbb{R}^6$. The exponential of $\boldsymbol{\omega}^\wedge \in \mathfrak{so}(3)$ ($\boldsymbol{\omega}^\wedge$ is the skew-symmetric matrix associated with $\boldsymbol{\omega} \in \mathbb{R}^3$) is a 3D rotation denoted by $\boldsymbol{R} \in \mathbb{SO}(3)$. And $\boldsymbol{t} \in \mathbb{R}^3$ is the translation. The cost function for the plane-edge-based motion estimation is designed as

$$F(\boldsymbol{\xi}) = \sum_{i=1}^{N_\pi} \boldsymbol{e}_{\pi i}^T \boldsymbol{\Omega}_{\pi i} \boldsymbol{e}_{\pi i} + W_p \sum_{k=1}^{N_p} w_{pk} \boldsymbol{e}_{pk}^T \boldsymbol{\Omega}_{pk} \boldsymbol{e}_{pk}. \quad (15)$$

The residual vectors $\boldsymbol{e}_{\pi i}$ and $\boldsymbol{e}_{pk}$ measure how well the estimated motion $\boldsymbol{\xi}$ aligns the planes and edge-points, respectively. $\boldsymbol{e}_{\pi i}$ is calculated by

$$\boldsymbol{e}_{\pi i} = {}^c\boldsymbol{\pi}_i - T({}^r\boldsymbol{\pi}_i, \boldsymbol{\xi}) \quad (16)$$

where $T({}^r\boldsymbol{\pi}_i, \boldsymbol{\xi})$ represents the plane transformed from ${}^r\boldsymbol{\pi}_i$ by $\boldsymbol{\xi}$, which is computed by

$$T({}^r\boldsymbol{\pi}_i, \boldsymbol{\xi}) = \begin{bmatrix} \boldsymbol{R} & \boldsymbol{0}_{3\times 1} \\ -\boldsymbol{t}^T\boldsymbol{R} & 1 \end{bmatrix} \begin{bmatrix} {}^r\boldsymbol{n}_i \\ {}^r d_i \end{bmatrix}. \quad (17)$$

$\boldsymbol{\Omega}_{\pi i}$ is the information matrix of the residual vector $\boldsymbol{e}_{\pi i}$ and is computed by the inverse of ${}^c\boldsymbol{C}_{\pi i} + {}^r\boldsymbol{C}_{\pi i}$, where $\boldsymbol{C}_{\pi i}$ is the covariance matrix of $\boldsymbol{\pi}_i$ and is given by (14). Likewise, the residual vector for edge-points $\boldsymbol{e}_{pk}$ is calculated by

$$\boldsymbol{e}_{pk} = {}^c\boldsymbol{p}_k - T({}^r\boldsymbol{p}_k, \boldsymbol{\xi}) \quad (18)$$

where $T({}^r\boldsymbol{p}_k, \boldsymbol{\xi})$ is the edge-point transformed from ${}^r\boldsymbol{p}_k$ by $\boldsymbol{\xi}$

$$T({}^r\boldsymbol{p}_k, \boldsymbol{\xi}) = \boldsymbol{R} \cdot {}^r\boldsymbol{p}_k + \boldsymbol{t}. \quad (19)$$

The information matrix $\boldsymbol{\Omega}_{pk}$ is given by the inverse of ${}^c\boldsymbol{C}_{pk} + {}^r\boldsymbol{C}_{pk}$, where $\boldsymbol{C}_{pk}$ is the covariance matrix of $\boldsymbol{p}_k$, and will be calculated in Section IV-C. In each iteration of ICP, (15) is minimized by the Gauss–Newton method to solve the transformation that best aligns the two scans.

The key innovation of the cost function (15) is the introduction of the weights W_p and $w_{pk}, k = 1, \ldots, N_p$, which is the essential part of the plane-edge-fusion framework. Note that the weights are not manually set. They are automatically computed according to the constraints provided by the planes and edges on the motion estimation, and the computation method is presented in Section IV-C. The adaptive weighting algorithm plays a key role in the seamless fusion of planes and edges. W_p is used to balance the contribution of two different kinds of features to the overall cost function in the optimization. w_{pk} is computed for each edge-point according to the constraints provided by this edge-point and the planes and is used to provide enough constraints to determine all the components of camera motion. In this way, the degeneracy in the plane-based motion estimation can be eliminated, and the problem remains well-conditioned, i.e., the camera motion, along with each dimension of the motion space, can be determined with certainty.

B. Constraint Analysis for Planes

If the robot only extracts the plane features and uses them as the primitives in scan matching, the calculation of the robot motion may suffer from degeneracy. In this case, additional information is needed to fully constrain the solution. To this end, the singular solutions to the problem should be identified beforehand, and a quantitative measure of the constraint

strength should be made available. In this section, we first demonstrate that a single pair of matched planes can only partially constrain the 6-DoF camera motion, and the unconstrained subspace is explicitly represented. Then, the analysis process is extended to the case of multiple plane pairs. The unconstrained subspace is spanned by multiple basis vectors, which are obtained by an eigenvalue decomposition (EVD) process, and the constraint is quantitatively measured by the eigenvalues.

We first consider a single pair of planes $\{{}^c\boldsymbol{\pi}_i, {}^r\boldsymbol{\pi}_i\}$. The Jacobian of $\boldsymbol{e}_{\pi i}$ with respect to $\boldsymbol{\xi}$ can be computed by

$$\boldsymbol{J}_{\pi i} = \frac{\partial \boldsymbol{e}_{\pi i}}{\partial \boldsymbol{\xi}} = \begin{bmatrix} \boldsymbol{0}_{3\times 3} & (\boldsymbol{R}\cdot {}^r\boldsymbol{n}_i)^\wedge \\ (\boldsymbol{R}\cdot {}^r\boldsymbol{n}_i)^T & -\boldsymbol{t}^T(\boldsymbol{R}\cdot {}^r\boldsymbol{n}_i)^\wedge \end{bmatrix}. \tag{20}$$

The camera motion that cannot be constrained by $\{{}^c\boldsymbol{\pi}_i, {}^r\boldsymbol{\pi}_i\}$ lies in the null space of $\boldsymbol{J}_{\pi i}$, which is denoted by $\text{null}(\boldsymbol{J}_{\pi i})$

$$\text{null}(\boldsymbol{J}_{\pi i}) = \{\boldsymbol{\xi} \in \mathbb{R}^6 \mid \boldsymbol{J}_{\pi i}\boldsymbol{\xi} = \boldsymbol{0}\} = \begin{bmatrix} \mu_1\boldsymbol{t}_1 + \mu_2\boldsymbol{t}_2 \\ \mu_3\boldsymbol{R}\cdot {}^r\boldsymbol{n}_i \end{bmatrix} \tag{21}$$

where $\boldsymbol{t}_1$ and $\boldsymbol{t}_2$ are orthogonal unit vectors spanning the plane vertical to $\boldsymbol{R}\cdot{}^r\boldsymbol{n}_i$, and $\mu_1, \mu_2, \mu_3 \in \mathbb{R}$. It can be seen from (21) that only three DoFs of the camera motion can be constrained by $\{{}^c\boldsymbol{\pi}_i, {}^r\boldsymbol{\pi}_i\}$.

Then, we consider the case of multiple plane pairs $\{{}^c\boldsymbol{\pi}_i, {}^r\boldsymbol{\pi}_i\}_{i=1,\ldots,N_\pi}$. The variation of $\boldsymbol{e}_{\pi i}$ caused by an infinitesimal change in the camera motion $d\boldsymbol{\xi}$ is represented by $d\boldsymbol{e}_{\pi i} = \boldsymbol{J}_{\pi i}d\boldsymbol{\xi}$. Summing the squared variations of residuals over the set of plane pairs $\{{}^c\boldsymbol{\pi}_i, {}^r\boldsymbol{\pi}_i\}_{i=1,\ldots,N_\pi}$ results in

$$\sum_{i=1}^{N_\pi} d\boldsymbol{e}_{\pi i}^T\boldsymbol{\Omega}_{\pi i}d\boldsymbol{e}_{\pi i} = d\boldsymbol{\xi}^T\left(\sum_{i=1}^{N_\pi}\boldsymbol{J}_{\pi i}^T\boldsymbol{\Omega}_{\pi i}\boldsymbol{J}_{\pi i}\right)d\boldsymbol{\xi}. \tag{22}$$

We denote $\boldsymbol{\Psi}_\pi = \sum_{i=1}^{N_\pi}\boldsymbol{J}_{\pi i}^T\boldsymbol{\Omega}_{\pi i}\boldsymbol{J}_{\pi i}$ and it contains information about the distribution of the Jacobians of residuals over all the matched planes. The EVD of $\boldsymbol{\Psi}_\pi$ can be computed by

$$\boldsymbol{\Psi}_\pi = \boldsymbol{Q}_\pi\boldsymbol{\Lambda}_\pi\boldsymbol{Q}_\pi^T = \sum_{l=1}^{6}\lambda_{\pi l}\boldsymbol{q}_{\pi l}\boldsymbol{q}_{\pi l}^T \tag{23}$$

where $\lambda_{\pi l}, l = 1, \ldots, 6$ are the eigenvalues of $\boldsymbol{\Psi}_\pi$, arranged in nonincreasing order, and $\boldsymbol{q}_{\pi l}, l = 1, \ldots, 6$ are the corresponding eigenvectors. Note that $\boldsymbol{q}_{\pi l}$ forms a basis in the 6-D space of a rigid motion, and $\lambda_{\pi l}$ indicates a measure of the constraint strength provided by the matched planes to the camera motion along $\boldsymbol{q}_{\pi l}$. For example, applying a transformation in the direction of $\boldsymbol{q}_{\pi 1}$ corresponding to the largest eigenvalue $\lambda_{\pi 1}$ will cause the largest change in the residuals, while the direction of $\boldsymbol{q}_{\pi 6}$ corresponding to the smallest eigenvalue $\lambda_{\pi 6}$ will cause the smallest change. The degeneracy occurs when there exists at least one zero eigenvalue $\lambda_{\pi l} = 0(\exists l)$. All the eigenvectors corresponding to the zero eigenvalues span the null space of $\boldsymbol{\Psi}_\pi$, in which the pose $\boldsymbol{\xi}$ cannot be constrained.

The above-mentioned analysis offers a fundamental to the fusion of planes and edges. In a seamless fusion, the contributions of planes and edges to the overall cost function should be complementary to each other. That is, for a given camera motion $\boldsymbol{\xi}$, if it cannot be constrained by the planes, the edge-points that can strongly constrain $\boldsymbol{\xi}$ should be adaptively assigned large weights. Therefore, it is of great importance to providing a quantitative measure of the constraint strength along each dimension of the motion space. In Section IV-C, the strength of the constraint provided by the edge-points along each direction $\boldsymbol{q}_{\pi l}, l = 1, \ldots, 6$ is measured, and the weights W_p and $w_{pk}, k = 1, \ldots, N_p$ in (15) are computed by fusing the information provided by planes and edges.

The matrix $\boldsymbol{\Psi}_\pi$ in (23) can also be used directly to identify the degenerate configurations of planes. In order to relate the degenerate cases to spatial configurations of planes, the null space of $\boldsymbol{\Psi}_\pi$ is described by the parameters of planes with the Jacobians $\boldsymbol{J}_{\pi i}$ calculated at a given camera pose. Specifically, at the pose $\boldsymbol{\xi} = \boldsymbol{0}$

$$\boldsymbol{\Psi}_\pi\mid_{\xi=0} = \sum_{i=1}^{N_\pi}\begin{bmatrix} \boldsymbol{\Omega}_{ddi}\cdot{}^r\boldsymbol{n}_i{}^r\boldsymbol{n}_i^T & {}^r\boldsymbol{n}_i\boldsymbol{\Omega}_{ndi}^T{}^r\boldsymbol{n}_i^\wedge \\ {}^r\boldsymbol{n}_i^{\wedge T}\boldsymbol{\Omega}_{ndi}{}^r\boldsymbol{n}_i^T & {}^r\boldsymbol{n}_i^{\wedge T}\boldsymbol{\Omega}_{nni}{}^r\boldsymbol{n}_i^\wedge \end{bmatrix} \tag{24}$$

where $\boldsymbol{\Omega}_{nni}$, $\boldsymbol{\Omega}_{ndi}$, and $\boldsymbol{\Omega}_{ddi}$ are the top-left 3×3, top-right 3×1, and bottom-right 1×1 submatrices of $\boldsymbol{\Omega}_{\pi i}$, respectively. Define the matrix $\boldsymbol{M} = \sum_{i=1}^{N_\pi}{}^r\boldsymbol{n}_i{}^c\boldsymbol{n}_i^T$ [5] and compute its SVD as

$$\boldsymbol{M} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{V}^T = \lambda_1\boldsymbol{u}_1\boldsymbol{v}_1^T + \lambda_2\boldsymbol{u}_2\boldsymbol{v}_2^T + \lambda_3\boldsymbol{u}_3\boldsymbol{v}_3^T \tag{25}$$

where $\boldsymbol{U} = [\boldsymbol{u}_1, \boldsymbol{u}_2, \boldsymbol{u}_3]$ and $\boldsymbol{V} = [\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3]$ are 3×3 orthonormal matrices, and $\boldsymbol{\Lambda} = \text{diag}\{\lambda_1, \lambda_2, \lambda_3\}$ with $\lambda_1 \geq \lambda_2 \geq \lambda_3$. Two cases of degeneracy are summarized as follows.

1) If $\lambda_1 \geq \lambda_2 > \lambda_3 = 0$, ${}^r\boldsymbol{n}_i^T\boldsymbol{u}_3 = 0$ holds true for all $i = 1, \ldots, N_\pi$. In this case, the 1-DoF camera motion $\boldsymbol{\xi}_{1D} = [\mu\boldsymbol{u}_3^T, \boldsymbol{0}^T]^T$ $(\mu \in \mathbb{R})$ satisfies $\boldsymbol{\xi}_{1D}^T\Psi_\pi\mid_{\xi=0}\boldsymbol{\xi}_{1D} = 0$. In other words, $\boldsymbol{\xi}_{1D}$ cannot be constrained by the matched planes. This degenerate case corresponds to the configuration that the normal vectors are coplanar.
2) Likewise, if $\lambda_1 > \lambda_2 = \lambda_3 = 0$, ${}^r\boldsymbol{n}_i$ $(i = 1, \ldots, N_\pi)$ satisfies ${}^r\boldsymbol{n}_i^T\boldsymbol{u}_2 = 0$, ${}^r\boldsymbol{n}_i^T\boldsymbol{u}_3 = 0$ and ${}^r\boldsymbol{n}_i \times \boldsymbol{u}_1 = 0$. In this case, the 3-DoF motion $\boldsymbol{\xi}_{3D} = [\mu_2\boldsymbol{u}_2^T + \mu_3\boldsymbol{u}_3^T, \mu_1\boldsymbol{u}_1^T]^T$ $(\mu_1, \mu_2, \mu_3 \in \mathbb{R})$ cannot be constrained by the matched planes. This degenerate case corresponds to the configuration that the normal vectors of planes are collinear.

Note that the cases 1) and 2) correspond to the 5-DoF and 3-DoF constraint cases discussed in our previous work [5], respectively. In [5], if the degeneracy is detected, the 5-DoF (3-DoF) motion is determined by the planes and the extra 1-DoF (3-DoF) is solved by a scan matching process. Differently, in this article, the motion along each dimension of the 6-D space is constrained by the fused information provided by both planes and edges. It is theoretically superior to the previous method [5] and the reason is as follows. In most circumstances, there exist l_1 and l_2 such that $0 < \lambda_{\pi l_1} \ll \lambda_{\pi l_2}$. In this case, the calculated motion has a large uncertainty along $\boldsymbol{q}_{\pi l_1}$ direction, which was not considered in [5]. In contrast, in this article, the edge-points that strongly constrain the motion along $\boldsymbol{q}_{\pi l_1}$ direction will be automatically assigned a high weight. Thus, the overall problem will become well-posed.

C. Computation of Adaptive Weights

In this section, the covariance ${}^c\boldsymbol{C}_{pk}$ for each edge-point ${}^c\boldsymbol{p}_k$ is estimated and the weights W_p and w_{pk} in (15) are adaptively computed based on analysis results in Section IV-B.

The computation of ${}^c\boldsymbol{C}_{pk}$ is different from that of ${}^c\boldsymbol{C}_{p_{\pi j}}$, which denotes the covariance of a point ${}^c\boldsymbol{p}_{\pi j}$ lying on a plane. Dryanovski *et al.* [41] demonstrated that the uncertainty of points around object edges could not be accurately modeled by the random error model in [33], which is only well suitable for describing the uncertainty of a point lying on a flat surface. The covariance ${}^c\boldsymbol{C}_{pk}$ is estimated using the edge-points in the spherical neighborhood of ${}^c\boldsymbol{p}_k$ with a given radius (the radius is set to 0.1 m in our implementation). It is illustrated in the Appendix that using ${}^c\boldsymbol{C}_{pk}$ in the computation of F_{pk}, the residual vector $\boldsymbol{e}_{pk}$ along the local edge direction of ${}^c\boldsymbol{p}_k$ will have the least contribution to F_{pk}. This conclusion characterizes the property of edge-points that they provide a little constraint on the camera motion along the edge direction. The property will be taken into account when computing the constraint strength provided by the edge-points, along each direction $\boldsymbol{q}_{\pi l}, l = 1, \ldots, 6$.

In the following, details about how to compute the weights W_p and w_{pk} are given. The Jacobian of the residual $\boldsymbol{e}_{pk}$ with respect to $\boldsymbol{\xi}$ can be calculated by

$$\boldsymbol{J}_{pk} = \frac{\partial \boldsymbol{e}_{pk}}{\partial \boldsymbol{\xi}} = [-\boldsymbol{I}_{3\times 3} \quad (\boldsymbol{R} \cdot {}^r\boldsymbol{p}_k)^\wedge]. \tag{26}$$

Similar to the constraint analysis on planes, a matrix $\boldsymbol{\Psi}_{pk} = \boldsymbol{J}_{pk}^T \boldsymbol{\Omega}_{pk} \boldsymbol{J}_{pk}$ is defined and the effect of the covariance matrix ${}^c\boldsymbol{C}_{pk}$ is involved in the information matrix defined by $\boldsymbol{\Omega}_{pk} = ({}^c\boldsymbol{C}_{pk} + {}^r\boldsymbol{C}_{pk})^{-1}$. Then, compute $\lambda_{pkl} = \boldsymbol{q}_{\pi l}^T \boldsymbol{\Psi}_{pk} \boldsymbol{q}_{\pi l}$. The value of λ_{pkl} gives a quantitative measure of the constraint strength provided by $\{{}^c\boldsymbol{p}_k, {}^r\boldsymbol{p}_k\}$ on the $\boldsymbol{q}_{\pi l}$ direction, which is used together with the constraint strength (measured by $\lambda_{\pi l}$) provided by planes to compute the weight w_{pk}

$$w_{pk} = \sum_{l=1}^{6} \frac{\nu_{pkl}}{\nu_{\pi l}} = \sum_{l=1}^{6} \frac{\lambda_{pkl} \Big/ \sum_{k=1}^{N_p} \lambda_{pkl}}{\exp\left(\alpha \sqrt{\frac{\lambda_{\pi l}}{\lambda_{\pi 1}}}\right)}. \tag{27}$$

The weight w_{pk} in (27) is adaptively computed by the constraints on motions along all the basis vectors $\boldsymbol{q}_{\pi l}, l = 1, \ldots, 6$ provided by both the planes and edge-points. Specifically, for each direction $\boldsymbol{q}_{\pi l}$, the numerator ν_{pkl} represents the proportion of the constraint from one edge-point pair $\{{}^c\boldsymbol{p}_k, {}^r\boldsymbol{p}_k\}$ among all the pairs $\{{}^c\boldsymbol{p}_k, {}^r\boldsymbol{p}_k\}_{k=1,\ldots,N_p}$. ν_{pkl} is divided by $\nu_{\pi l}$ which varies within the range $[1, e^\alpha]$. Note that if $\boldsymbol{q}_{\pi l}$ lies in the null space of $\boldsymbol{\Psi}_\pi$, i.e., $\lambda_{\pi l} = 0$ (planes cannot constrain the estimation of the motion along $\boldsymbol{q}_{\pi l}$), the denominator $\nu_{\pi l}$ equals 1 and ν_{pkl} is directly added to w_{pk}. And if $\lambda_{\pi l} > 0$ (planes can constrain the estimation of the motion along $\boldsymbol{q}_{\pi l}$ and the constraint strength is quantified by $\lambda_{\pi l}$), the value of $\nu_{\pi l}$ is greater than 1 and ν_{pkl} is reduced (divided by $\nu_{\pi l} > 1$) before being added to w_{pk}. Thus, the contribution of $\{{}^c\boldsymbol{p}_k, {}^r\boldsymbol{p}_k\}$ to w_{pk} is decreased according to the constraint provided by planes along $\boldsymbol{q}_{\pi l}$. We adopt the exponential function in the denominator because it increases faster as $(\lambda_{\pi l}/\lambda_{\pi 1})^{1/2}$ increases. In other words, as the constraint $\lambda_{\pi l}$ along $\boldsymbol{q}_{\pi l}$ increases, the weight w_{pk} decreases faster and the contribution of the corresponding edge-point pair $\{{}^c\boldsymbol{p}_k, {}^r\boldsymbol{p}_k\}$ will be more restrained. In this way, the weight w_{pk} can adaptively fuse the information of the planes and edges. The impact of w_{pk} on the overall cost function (15) is that the edge-points constraining the motion in the null space of $\boldsymbol{\Psi}_\pi$ contribute more to (15).

The weight w_{pk} is also used in the selection of edge-points. Because the amount of edge-points is much larger than that of planes, using all the edge-points will greatly increase the computational load. Furthermore, the edge-points with small weights w_{pk} have little contribution to motion estimation. Therefore, exclusion of the edge-points with small weights has little effect on the accuracy of motion estimation. The selection of edge-points can largely increase the real-time performance without affecting the accuracy of the algorithm. In our implementation, the weight w_{pk} is thresholded by 0.01, and extensive experiments in different scenes demonstrate that this threshold yields satisfactory performance.

As regards the coefficient $\alpha \geq 0$, it can be preset by users according to the application requirements. If α is set to zero, the weight of each edge-point is 1. In this case, the contribution of each edge-point to the cost function (15) is not affected by the planes. The larger α is, the more the contribution of edge-points will be reduced. Therefore, if a large number of planes are extracted from the scenes, a large α is more suitable. On the contrary, in the scenes where the planes cannot be stably extracted, α should be set relatively small. In our experiments, α is set to 1.

The weight W_p is used to balance the contributions of two different kinds of primitives. Because $\boldsymbol{\pi}_i$ and $\boldsymbol{p}_k$ are defined in different spaces, the two terms of (15) may vary greatly in magnitude. The weight W_p is computed by normalizing the magnitude of the second term (corresponding to the edge-points) of (15) with respect to the first term (corresponding to the planes)

$$W_p = \frac{\sum_{l=1}^{6} \lambda_{\pi l}}{\sum_{l=1}^{6} \sum_{k=1}^{N_p} w_{pk} \lambda_{pkl}}. \tag{28}$$

V. Loop Closing and Pose-Graph Optimization

Essentially, the motion estimator proposed in Section IV is a VO for RGB-D cameras. Though it is sufficiently accurate in a short period of time, the problem of error accumulation in the VO is unavoidable. Therefore, the loop-closure detection and backend pose-graph optimization of an SLAM system are necessary.

Because the proposed motion estimator is sufficiently accurate and mainly aims at the indoor environments with a relatively limited area, we simply search loop closure candidates in a spherical neighborhood around the current position to detect a loop closure [42], [43]. Once a loop closure is detected, a new edge is added to the pose graph. The accumulating errors can be corrected by solving a nonlinear LS minimization problem. The g2o framework [44] is applied to optimize the pose graph.

VI. Experimental Evaluation

In this section, the proposed probabilistic plane fitting and plane-edge-based camera motion estimation are evaluated by extensive experiments. The TUM [36] and ICL-NUIM [37] RGB-D benchmarks are used in the assessments. The TUM

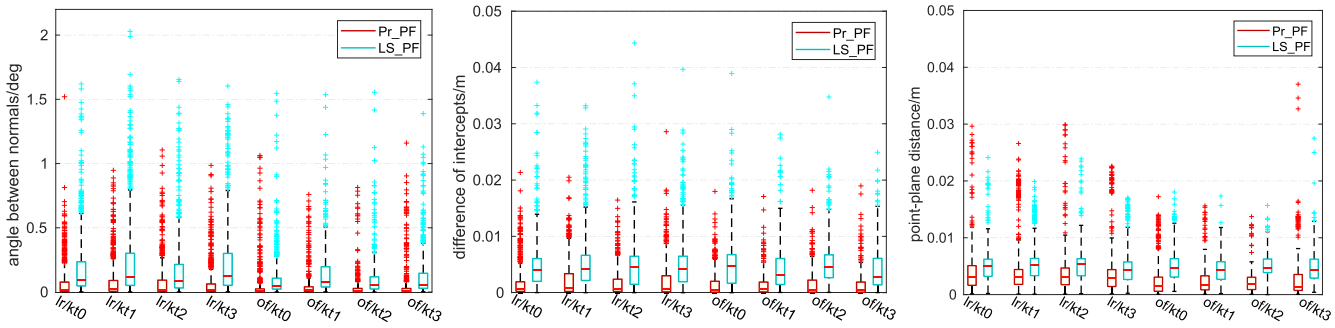


Fig. 3. Comparison of accuracy of detected planes in terms of different evaluation metrics. (a) Angle between normals. (b) Difference of intercepts. (c) Point-plane distance.

benchmark provides image sequences captured in indoor environments with time-synchronized ground truth from an external motion capture system. And the ICL-NUIM data set includes a collection of hand-held RGB-D camera sequences within synthetically generated environments. The test platform for all the experiments is an onboard computer with an Intel Core i5-3230M CPU at 2.6 GHz and 3.8-GB RAM.

The performance of the proposed probabilistic plane fitting is compared with the widely used LS method in Section VI-A. The adaptive weighting strategy in plane-edge-fusion is evaluated in Section VI-B. In Section VI-C, the plane-edge-VO is compared with three VO methods. In Section VI-D, the plane-edge-SLAM system is implemented and is compared with seven state-of-the-art SLAM systems. In Section VI-E, the point-cloud map constructed based on the estimated trajectory is evaluated and compared with three plane-based SLAM systems. In Section VI-F, the plane-edge-VO is run online in three different kinds of real-world indoor scenes to demonstrate the efficiency and robustness of our method.

A. Experiments on Plane Fitting

In this section, the probabilistic plane fitting method proposed in Section III is evaluated. We use the synthetic RGB-D data set ICL-NUIM [37] to obtain the ground truth of the detected planes. The planes are first extracted from the 3-D models of the synthetic scene and then used as the ground truth for subsequent quantitative evaluations. Then, plane models are fitted using the points on the planes with simulated Kinect 1.0 sensor noise by the proposed probabilistic plane fitting (referred to as Pr_PF) method and the LS plane fitting (referred to as LS_PF) method, respectively. Three evaluation metrics are used to compare the estimated planes and the ground truth, i.e., the angle between normals, the difference of intercepts and the average distance from the ideal points to the estimated plane model. The comparison results are shown in Fig. 3. It can be seen clearly that Pr_PF obtains better results in terms of three metrics on all the sequences in ICL-NUIM.

To further demonstrate the effect of the fitted plane model to the accuracy of the motion estimation, the two-plane fitting methods are used in the plane-edge-VO, respectively, and are run on five image sequences from the TUM benchmark.

TABLE I

COMPARISON OF ATE RMSE BETWEEN PLANE-EDGE-VOS WITH Pr_PF AND LS_PF, RESPECTIVELY

	Pr_PF	LS_PF
fr1/desk	**0.028m**	0.055m
fr1/plant	**0.048m**	0.061m
fr2/desk	**0.051m**	0.089m
fr3/office	**0.053m**	0.069m
fr3/str	**0.030m**	0.052m

The root-mean-square error (RMSE) of the absolute trajectory error (ATE) is calculated for each image sequence, as shown in Table I. It can be seen clearly from Table I that the Pr_PF significantly improves the accuracy of the motion estimation, compared with the LS_PF. The Pr_PF method takes account of the measurement errors that are correlated with the coordinates of 3-D points. As a result, the Pr_PF is less affected by measurements with large uncertainties, and the resultant plane model can provide a more accurate estimate for the ego-motion of the camera.

B. Experiments on Plane-Edge-Fusion

In this section, the plane-edge-VOs with and without adaptive weighting scheme, respectively, are compared. In the plane-edge-VO with uniform weighting, the weights of edge-points are not adaptively computed and all set to 1 in the optimization process. The RMSE of ATE is computed to evaluate the accuracy, which is shown in Table II. We can see that when the adaptive weighting scheme is applied, the accuracy of plane-edge-VO is significantly improved. By adaptively weighting the edge-points, the information from both planes and edges is fused, and the problem of camera motion estimation becomes well-posed, as discussed in Section IV-C. In contrast, for the plane-edge-VO with uniform weighting, the problem is likely to be ill-posed. Therefore, the plane-edge-VO with the adaptive weighting scheme yields better results in terms of accuracy.

The modulewise average runtime of each scan-alignment is computed on the image sequences, and the results are plotted in Fig. 4(a), which shows that the motion estimation with adaptive weighting is much time saving than that with uniform weighting. The corresponding detailed statistics for

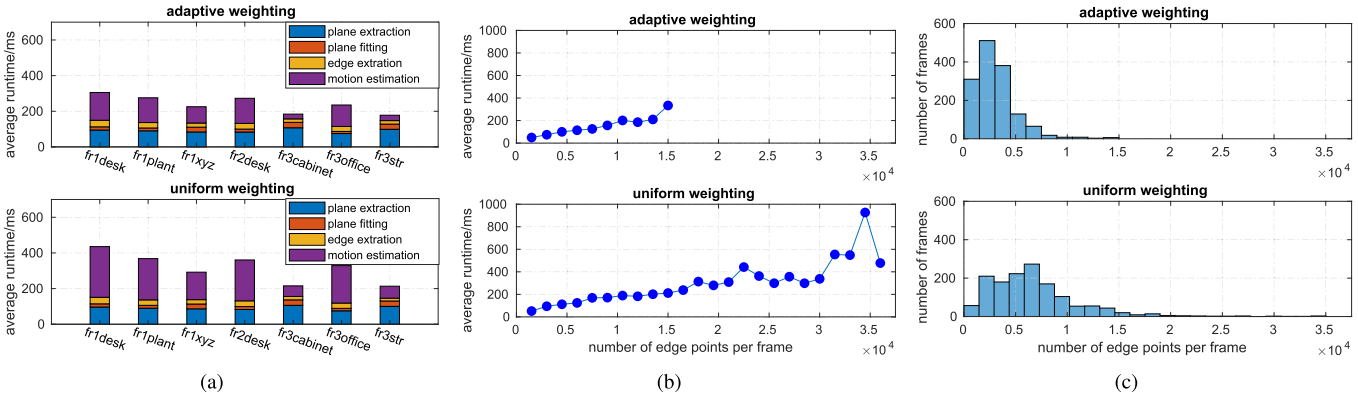


Fig. 4. (a) Modulewise average runtime of each scan alignment of plane-edge-VO with (top) adaptive weighting and (bottom) uniform weighting, respectively. (b) Average runtime of each scan-alignment w.r.t. the number of edge-points used in motion estimation. (c) Histogram of the quantity of edge-points.

TABLE II
ATE RMSES OF PLANE-EDGE-VOS WITH ADAPTIVE WEIGHTING AND UNIFORM WEIGHTING, RESPECTIVELY

	adaptive weighting	uniform weighting
fr1/xyz	**0.031m**	0.056m
fr1/desk	**0.028m**	0.062m
fr1/plant	**0.048m**	0.068m
fr2/desk	**0.051m**	0.106m
fr3/office	**0.053m**	0.095m
fr3/str	**0.030m**	0.061m
fr3/cabinet	**0.063m**	0.079m

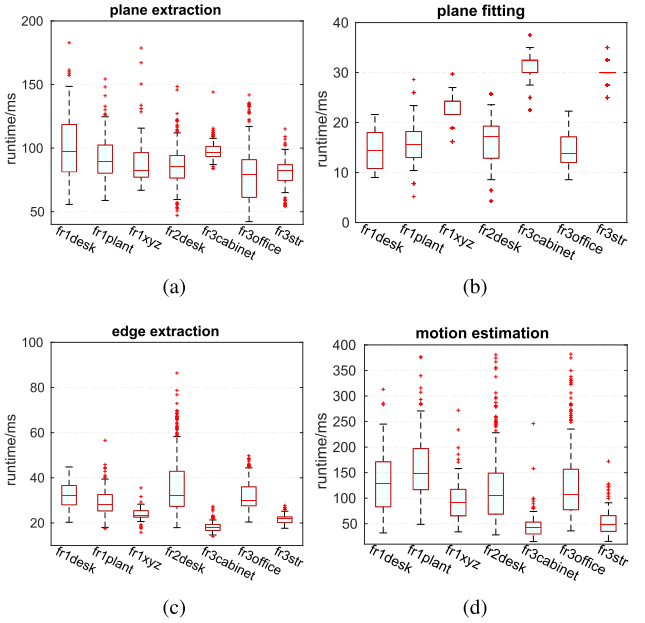


Fig. 5. Statistics of modulewise runtime for the plane-edge-VO with adaptive weighting scheme. (a) Plane extraction. (b) Plane fitting. (c) Edge extraction. (d) Motion estimation.

the plane-edge-VO with adaptive weighting is presented by the boxplot in Fig. 5. Furthermore, the relationship between the runtime and the number of edge-points is shown in Fig. 4(b), which indicates that the average runtime increases along with the increase of the number of edge-points. The histogram in Fig. 4(c) gives the number of frames as a function of the number of edge-points per frame. It can be seen that much fewer edge-points are involved in motion estimation for the plane-edge-VO with adaptive weighting. Because through the edge-point selection described in Section IV-C, a large number of edge-points (above 50% with the threshold being set to 0.01) are excluded by thresholding the weight w_{pk} of each edge-point. As a result, the computational load is largely reduced. Because excluded edge-points have little contribution to the motion estimation, the exclusion of them has little influence on the accuracy of the VO.

C. Evaluation of Visual Odometry

In this section, the plane-edge-VO is compared with three other VO systems using geometric features, i.e., CPA-VO [45], STING-VO [5], and Canny-VO [20]. The VO system is achieved by a frame-to-frame incremental scan-alignment *without the loop closure and graph optimization*. The CPA-VO tracks the camera motion toward a reference frame and a global plane model in an expectation–maximization (EM) framework. For the STING-VO, the camera poses are calculated directly by plane features extracted from two successive frames. When the planes cannot fully constrain the pose estimate, an STING-based scan-alignment is performed to offer remaining constraints [5]. Both CPA-VO and STING-VO use plane features in the camera tracking process. The Canny-VO is an efficient RGB-D VO system that is achieved by aligning the Canny edge features extracted from the images. Tables III and IV present the comparison results between the plane-edge-VO and the other three VO algorithms. The results of the Canny-VO and CPA-VO have been reported in [20] and [45], respectively. Note that the EM tracking in the CPA-VO is implemented with a GPU to support the real-time computation of the algorithm.

The results in Table III show that the performance of the plane-edge-VO is better than the other three state-of-the-art VO systems in terms of ATE, except on the fr2/desk sequence. Both the plane-edge-VO and STING-VO use planes as a high-level representation of the raw data and use parameters of planes to estimate the camera motion. In contrast, the CPA-VO uses the dense image information and aligns it with the global

TABLE III

COMPARISON OF ATE RMSE BETWEEN FOUR VO ALGORITHMS: PLANE-EDGE-VO, CPA-VO [45], STING-VO [5], AND CANNY-VO [20]

	plane-edge-VO	CPA-VO	STING-VO	Canny-VO
fr1/desk	**0.028m**	0.030m	0.041m	0.044m
fr1/plant	**0.048m**	0.073m	0.070m	0.059m
fr2/desk	0.051m	0.095m	0.098m	**0.037m**
fr3/office	**0.053m**	0.076m	0.089m	0.085m
fr3/str	**0.030m**	0.036m	0.040m	0.031m

TABLE IV

COMPARISON OF ATE RMSE BETWEEN THREE VO ALGORITHMS: PLANE-EDGE-VO, CANNY-VO [20], AND STING-VO [5]

	plane-edge-VO	Canny-VO	STING-VO
fr1/xyz	**0.006m/0.68deg**	0.019m/1.12deg	0.022m/0.77deg
fr1/floor	**0.009m/0.53deg**	0.010m/0.82deg	0.011m/0.68deg
fr2/xyz	**0.002m/0.21deg**	0.004m/0.31deg	0.004m/0.34deg
fr2/rpy	**0.002m/0.20deg**	0.004m/0.32deg	0.004m/0.30deg
fr2/desk	**0.005m/0.50deg**	0.008m/0.45deg	0.048m/1.57deg
fr3/cabinet	**0.006m/0.91deg**	0.036m/1.63deg	0.011m/1.02deg
fr3/office	**0.004m/0.42deg**	0.010m/0.50deg	0.009m/0.50deg

plane model via a direct image alignment. It is the reason that CPA-VO needs the GPU to support real-time computation. We can see from Table III that though the plane-edge-VO uses plane parameters to represent the raw data, rather than the dense image information, it can achieve a higher accuracy of the estimated trajectory in most image sequences than the CPA-VO. In addition, the plane-edge-VO presents superior performance over the STING-VO in terms of ATE and relative pose error (RPE), as shown in Tables III and IV. As illustrated in Section IV-B, in the STING-VO, the motion estimation problem may be ill-conditioned even though a full 6-DoF solution can be determined by planes. In this case, the final solution may suffer from a large uncertainty in some DoFs. In contrast, this issue is skillfully addressed in the proposed plane-edge-VO algorithm through a seamless fusion of planes and edges. As a result, better results can be achieved. We also compare the plane-edge-VO with Canny-VO in terms of ATE and RPE. It can be seen from Tables III and IV that the plane-edge-VO also shows superiority over Canny-VO that aligns edges with estimating the camera motion.

D. Evaluation of SLAM

In this section, the proposed plane-edge-SLAM system is compared with seven state-of-the-art SLAM systems: ORB-SLAM2 [46], RGBD-SLAM [42], PL-SLAM [47], edgeSLAM [48], CPA-SLAM [45], ElasticFusion [49], and STING-SLAM [5]. The ORB-SLAM2 [46] and RGBD-SLAM [42] are among the most popular point feature-based SLAM systems. The PL-SLAM [47], edgeSLAM [48], CPA-SLAM [45], and STING-SLAM [5] are SLAM systems using geometric features. The ElasticFusion [49] is a map fusion-based SLAM framework. The experimental results are given in Table V, where the results of ORB-SLAM2 are obtained using the open-source implementation, and those of other systems have been reported in their respective publications. It can be seen clearly that our method compares favorably to other state-of-the-art SLAM methods and is obviously superior to other systems in the textureless scenes. The estimated trajectories compared against ground truth trajectories on two image sequences are plotted in Fig. 6 and compared with ORB-SLAM2, which is widely acknowledged as the most efficient and accurate open-source SLAM system. Furthermore, the trajectories and the point-cloud maps estimated by plane-edge-SLAM in textureless scenes fr3/str_ntex_near and fr3/str_ntex_far are shown in Fig. 7(a) and (b), respectively, where the ORB-SLAM2 fails to track the camera because insufficient visual features can be extracted.

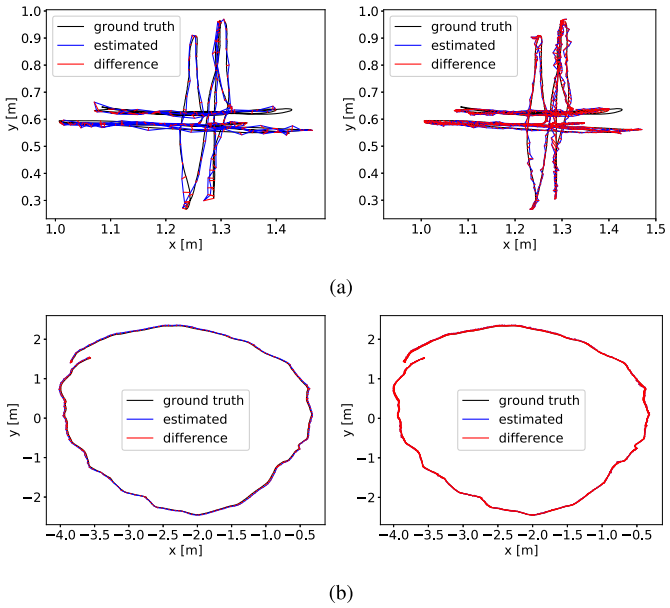


Fig. 6. Trajectories estimated by (left) plane-edge-SLAM and (right) ORB-SLAM2, respectively, compared against ground truth trajectories. (a) fr1/xyz. (b) fr3/nstr_tex_near.

E. Evaluation on Map Quality

In this section, the map of plane-edge-SLAM is compared with that of three planes feature-based SLAM: SA-SHAGO [19], STING-SLAM [5], and point-plane-SLAM [50]. To quantitatively evaluate the quality of the map, we use the synthetic RGB-D data set ICL-NUIM [37]. The synthetically generated *living room (lr/kt0)* scene in the ICL-NUIM data set has an associated 3-D polygonal model, which allows evaluation of the accuracy of map reconstruction. The map quality is evaluated by the mean distance from each point to the nearest surface in the ground truth 3-D model, as is recommended in [37]. The first two columns of Table VI list the comparison results of the four SLAM methods. It is obvious that the plane-edge-SLAM shows superior performance in terms of both qualities of the map and the accuracy of the trajectory. The heat map of the reconstruction is shown in Fig. 8, in which the areas that are less accurately reconstructed are more highlighted, which intuitively illustrates the quality of all the areas in the constructed point-cloud map.

Apart from the quantitative evaluation on the quality of the map, we run the four plane-based SLAM systems in two

TABLE V
COMPARISON OF SLAM IN TERMS OF RMSE OF ATE

	plane-edge-SLAM	ORB-SLAM2	RGBD-SLAM	PL-SLAM	EdgeSLAM	ElasticFusion	CPA-SLAM	STING-SLAM
fr1/xyz	**0.010m**	0.013m	0.012m	0.012m	0.013m	0.011m	0.011m	0.011m
fr1/desk	0.020m	**0.016m**	0.026m	–	–	0.020m	0.018m	0.030m
fr1/plant	**0.015m**	0.017m	0.059m	–	–	0.022m	0.029m	0.027m
fr2/xyz	0.008m	**0.004m**	0.015m	0.004m	0.005m	0.011m	0.014m	0.010m
fr2/desk	0.030m	**0.009m**	0.057m	–	0.017m	0.071m	0.046m	0.053m
fr3/office	0.015m	**0.010m**	0.413m	0.019m	–	0.017m	0.025m	0.034m
fr3/nst_tex_near	**0.015m**	0.019m	0.026m	0.020m	–	0.016m	0.016m	0.018m
fr3/str_tex_far	0.014m	0.015m	0.033m	0.009m	**0.006m**	0.013m	–	0.009m
fr3/str_ntex_near	**0.018m**	failed	0.034m	–	0.083m	0.021m	–	0.037m
fr3/str_ntex_far	**0.029m**	failed	0.068m	–	0.067m	0.030m	–	0.060m

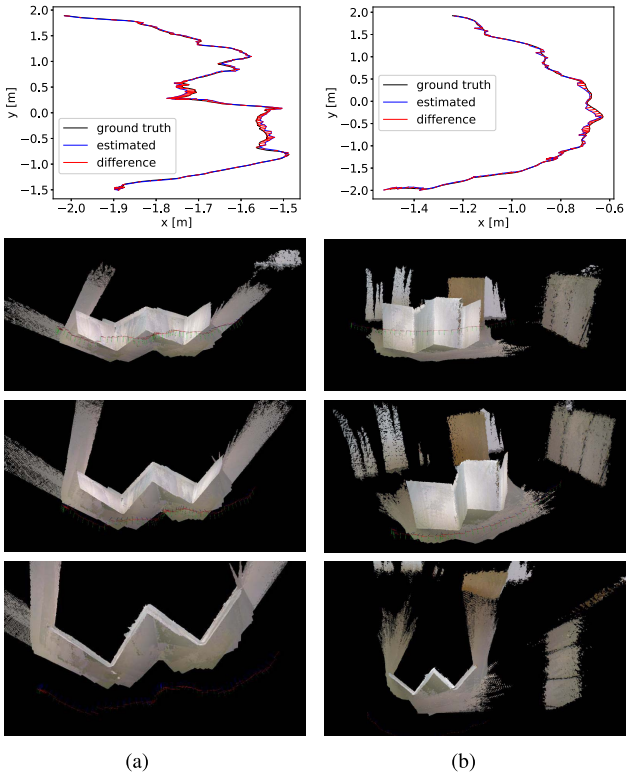


Fig. 7. Estimated trajectories and the generated point-cloud maps by plane-edge-SLAM on the sequences (a) fr3/str_ntex_near and (b) fr3/str_ntex_far, respectively.

TABLE VI
COMPARISON OF FOUR PLANE-BASED SLAM SYSTEMS

	map (lr/kt0)	ATE (lr/kt0)	ATE (fr3/tex)	ATE (fr3/ntex)
plane-edge-SLAM	**0.014m**	**0.015m**	**0.014m**	**0.029m**
SA-SHAGO	0.024m	0.032m	0.032m	0.440m
STING-SLAM	0.054m	0.057m	0.056m	0.066m
point-plane-SLAM	0.106m	0.143m	0.066m	0.383m

scenes with a similar structure from the TUM data set, one of which has high texture (fr3/str_tex_far), and the other has low texture (fr3/str_ntex_far). The RMSEs of ATE are listed in the last two columns of Table VI, and it can be seen that the plane-edge-SLAM achieves the best results on both sequences. In addition, in the textureless scene, the SA-SHAGO and

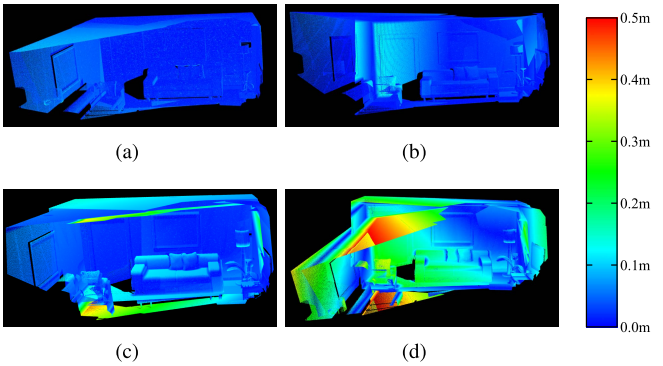


Fig. 8. Heat maps of the reconstructions for four methods on the lr/kt0 sequence of the ICL_NUIM benchmark. The reconstruction error is indicated by the color bar at the right-hand side. (a) Plane-edge-SLAM. (b) SA-SHAGO. (c) STING-SLAM. (d) Point-plane-SLAM.

point-plane-SLAM methods cannot localize the camera accurately. Because both the SA-SHAGO and point-plane-SLAM systems rely on the visual features to localize the camera when plane features are insufficient to constrain the pose estimation [19], [50]. And the sequence fr3/str_ntex_far is captured in a textureless scene, and little visual features can be extracted. As a result, both systems perform poorly in this scene. In contrast, the proposed plane-edge-SLAM and the STING-SLAM do not rely on the visual features, and thus, can get favorable results in both textured and textureless scenes. The estimated trajectories and the generated point-cloud maps of the four methods on the sequence fr3/str_ntex_far are shown in Fig. 9. The experimental video for this section is also included in the submitted multimedia files.

F. Real-World Experiments

In this section, we test the proposed plane-edge-VO in three different kinds of real-world scenes, i.e., a laboratory, a large-scale corridor and challenging illumination scenes, respectively. The results are presented in Sections IV-F.1–IV-F.3, and the corresponding experimental processes are recorded in the videos, which are submitted as Supplementary Materials. No backend optimization is performed during the localization and mapping process such that the performance of the plane-edge-based camera motion estimation can be fully presented.

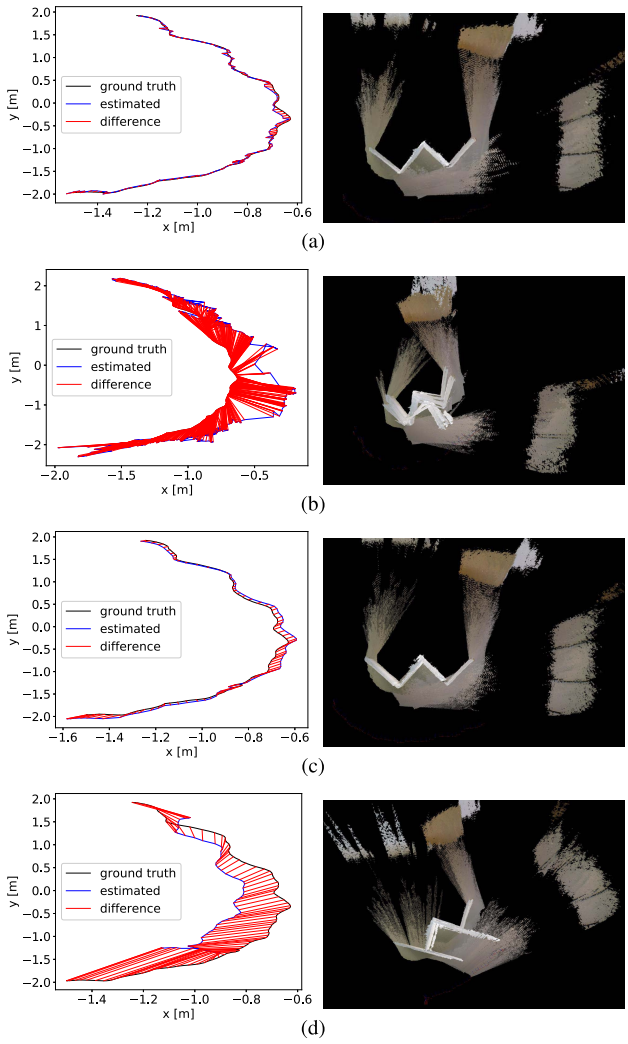


Fig. 9. Trajectories and point-cloud maps estimated by the four plane SLAM systems on the fr3/str_ntex_far sequence of the TUM benchmark. (a) Plane-edge-SLAM. (b) SA-SHAGO. (c) STING-SLAM. (d) Point-plane-SLAM.

1) Laboratory Scene: In the first experiment, the robot is joysticked around a laboratory, as is shown in Fig. 10(a). The size of the laboratory is 12.00 m × 8.40 m, and the width of the corridor is 2.35 m. It can be seen from Fig. 10(a) that the 3-D model of the environment is well constructed through an incremental camera motion estimation, which is achieved by a frame-to-frame registration based on planes and edge-points. The results demonstrate that the proposed plane-edge-based method can achieve accurate motion estimation in real-time robot navigation. Four zoomed-in views of the point-cloud map are given in Fig. 10(b)–(e). Note that the presented point-cloud map is simply created by downsampled scan data without any point of cloud fusion. It further demonstrates the accuracy of the proposed method.

In addition, in the attached experimental video, we can see that when the robot is moving through the corridor, a person accidentally passes by (xxmin xxs∼xxmin xxs in the video "video_real_world.mp4"). As shown from the trajectory of the robot, only the estimation of the translational motion along the direction of the corridor is affected by the moving person. The estimate of the other 5-DoF motion is not affected,

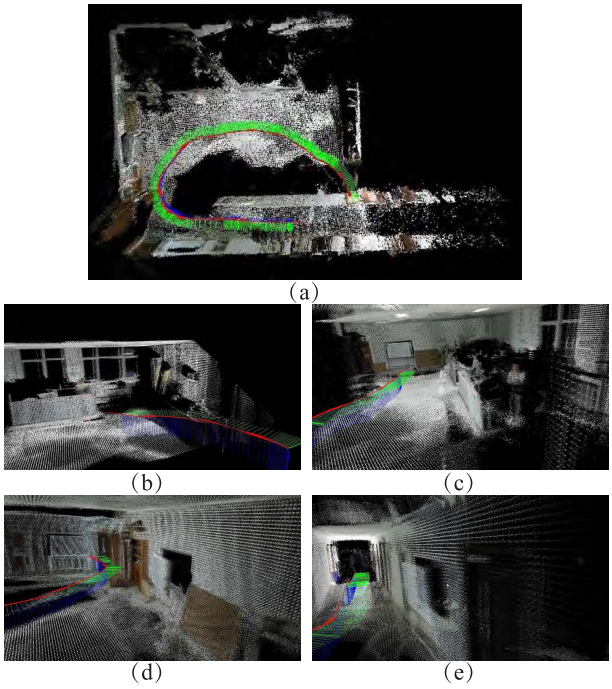


Fig. 10. Point-cloud map generated by the plane-edge-VO algorithm in a laboratory. The x-, y-, and z-axes of the camera coordinate system are represented by green, blue, and red line segments, respectively. (a) Panoramic view of the generated map. (b)–(e) Four zoomed-in views of the map.

and the reason is as follows. The estimate of the other five DoFs is strongly constrained by the two nonparallel planes extracted from the corridor scene (the floor and the walls). And the plane extraction and fitting are not affected by the nonplanar dynamic objects in the environments. However, the translational DoF along the direction of the corridor cannot be determined by planes, and the edge information is needed to constrain the estimate of this translational motion. However, edge extraction is affected by the moving object. As a result, the estimation of the translation along the corridor is influenced. In summary, though the plane-edge-VO is not completely robust to the dynamic environments, the impact of the moving object is relatively small because the plane extraction and fitting are robust to the nonplanar moving objects.

2) Large-Scale Corridor Scene: In the second experiment, we test our method in a large-scale corridor environment, as shown in Fig. 11(a). The size of the environment is approximately 80 m × 60 m. The robot is joysticked along the corridor and then is back to the start position. We can see from Fig. 11(a) that the accumulated offset is very small after traveling a distance of about 280 m, without any optimization on the estimated trajectory. It strongly demonstrates the accuracy of the proposed method. Four images captured by the camera and the corresponding zoomed-in views of the point-cloud map are shown in Fig. 11(b)–(e), and the positions of the robot when capturing the images are labeled in the panoramic view in Fig. 11(a).

3) Challenging Illumination Conditions: The plane-edge-VO is also performed in scenes under varying and low illumination conditions, respectively. During the experiment under

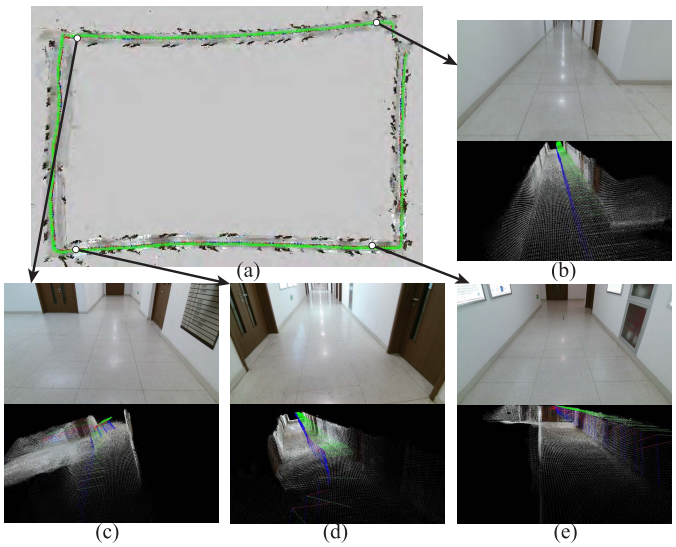


Fig. 11. Point-cloud map generated by the plane-edge-VO algorithm in a large-scale environment. (a) Panoramic view of the generated map. (b)–(e) Four images captured at different positions which are labeled in (a), and the corresponding zoomed-in views of the map.

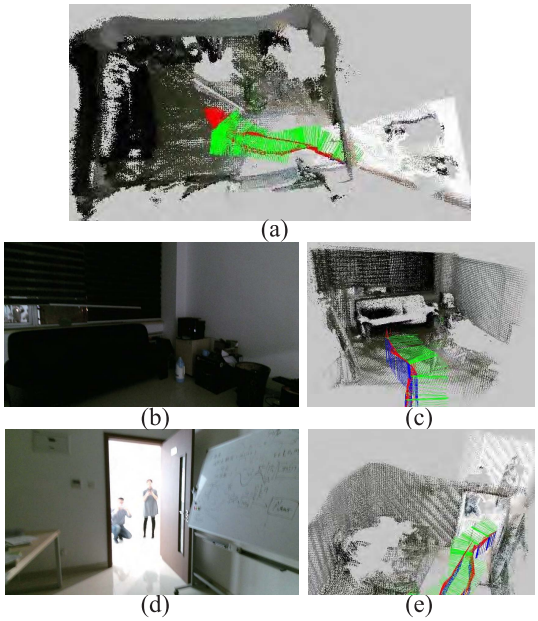


Fig. 12. Point-cloud map generated by the plane-edge-VO algorithm under varying illumination. (a) Panoramic view of the generated map. (b) and (c) Image of the dark office and the corresponding zoomed-in view of the map. (d) and (e) Image of the strong light from the corridor and the corresponding zoomed-in view of the map.

varying illumination condition, the robot is joysticked into a dark room with the light OFF and then back to a corridor in the normal lighting condition, as shown in Fig. 12(b) and (d). From the panoramic view Fig. 12(a) and the zoomed-in views Fig. 12(c) and (e), we can see that the plane-edge-VO can achieve good results under varying illumination condition. Because our method utilizes only geometric features extracted from the depth images, rather than visual features extracted from the RGB images, it is robust to the changes of the lighting condition.

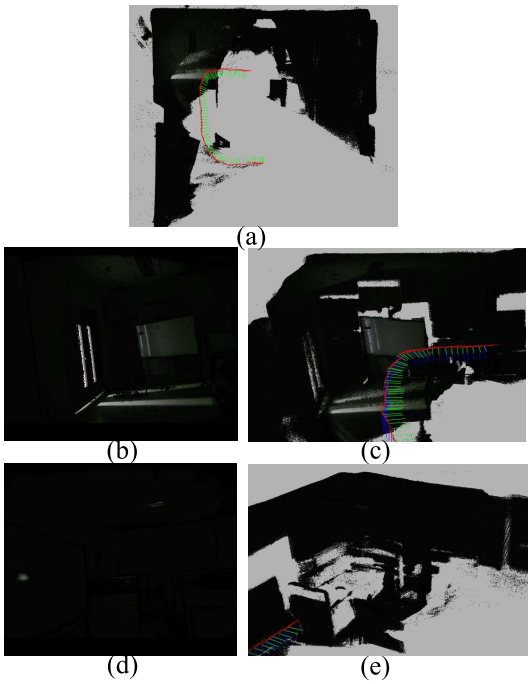


Fig. 13. Point-cloud map generated by the plane-edge-VO algorithm in a completely dark scene. (a) Panoramic view of the generated map. (b) and (d) Two RGB images. (c) and (e) Corresponding zoomed-in views in the map.

To further demonstrate the performance of our system in low illumination conditions, an experiment is conducted in a completely dark scene. The Kinect 2.0 is hand-held through the darkroom, and the point-cloud map is generated along with the tracking of the camera. The results are shown in Fig. 13. We have also run ORB-SLAM2 [46], RGBD-SLAM [42], and SA-SHAGO [19] in the same environment. But all of them fail to track the camera because they cannot extract visual features from such a dark background. It is worthwhile to point out that under good illumination conditions, our method can also be combined with the visual feature-based SLAM to further enhance the performance of the system.

VII. Conclusion

In this article, the plane-edge-SLAM system has been developed with the plane-edge-fusion and probabilistic plane fitting. The constraint analysis for planes is a prerequisite part of the fusion of planes and edges because it provides a quantitative measure of the constraint strength. The analysis result can also be used to identify the singular solutions to the motion estimation problem in any plane-based SLAM system since it gives an explicit representation of the unconstrained subspace of motion. An adaptive weighting algorithm is elaborately designed for the seamless fusion of planes and edges. The weights of the edge-points are adaptively computed based on the quantitative measure of constraint strength on the motion along each dimension of the motion space. To the best of our knowledge, it is the first time that the result of the constraint analysis is used in the fusion of planes and edges. It can provide a novel point of view for the problem of information fusion.

A probabilistic plane fitting algorithm has been proposed to fit a plane model to the measured points. The plane fitting is adaptive to various measurement noises corresponding to the different depth values by exploiting the error model of the depth sensor. Thus, the estimated plane model is more accurate and robust to large measurement noises. The fitted plane is further used in the estimation of the camera motion, and the accuracy of the motion estimation can be largely improved. Furthermore, the proposed probabilistic method can be easily extended to other sensors, such as the binocular vision sensor, the laser scanner, the 3-D lidar sensors.

In addition, the extraction of the two features, i.e., planes and depth edges, is inherently insensitive to the illumination changes, and no RGB information is involved in the processes of both plane fitting and motion estimation. As a result, the plane-edge-SLAM system is definitely effective in completely dark environments, which is demonstrated in real-world experiments. Therefore, the plane-edge-SLAM is shown to be an attractive complement to the state-of-the-art visual SLAM system in the indoor scenes that are challenging for the vision sensors.

APPENDIX
INFLUENCE OF EDGE-POINT COVARIANCE ON $F(\boldsymbol{\xi})$

In this appendix, we illustrate that using the ${}^c\boldsymbol{C}_{pk}$ in the computation of F_{pk}, the residual vector $\boldsymbol{e}_{pk}$ along the local edge direction of ${}^c\boldsymbol{p}_k$ has the least contribution to F_{pk}.

The eigenvalues of ${}^c\boldsymbol{C}_{pk}$ are denoted by γ_j (arranged in nonincreasing order and $j = 1, 2, 3$), and the corresponding eigenvectors (unit vectors) are denoted by $\boldsymbol{w}_j$. Since ${}^c\boldsymbol{C}_{pk}$ is estimated using the edge-points in the neighborhood of ${}^c\boldsymbol{p}_k$, it is obvious that $\boldsymbol{w}_1$ is the local edge direction at the point ${}^c\boldsymbol{p}_k$ and $\gamma_1 \gg \gamma_2 \geq \gamma_3$. Thus,

$$\frac{1}{\gamma_3} \geq \frac{1}{\gamma_2} \gg \frac{1}{\gamma_1}. \tag{29}$$

Then, we know that

$$\boldsymbol{\Omega}_{pk} = {}^c\boldsymbol{C}_{pk}^{-1} = \frac{1}{\gamma_1}\boldsymbol{w}_1\boldsymbol{w}_1^T + \frac{1}{\gamma_2}\boldsymbol{w}_2\boldsymbol{w}_2^T + \frac{1}{\gamma_3}\boldsymbol{w}_3\boldsymbol{w}_3^T. \tag{30}$$

F_{pk} can be written as

$$\begin{aligned} F_{pk} &= \boldsymbol{e}_{pk}^T\boldsymbol{\Omega}_{pk}\boldsymbol{e}_{pk} \\ &= \boldsymbol{e}_{pk}^T\left(\frac{1}{\gamma_1}\boldsymbol{w}_1\boldsymbol{w}_1^T + \frac{1}{\gamma_2}\boldsymbol{w}_2\boldsymbol{w}_2^T + \frac{1}{\gamma_3}\boldsymbol{w}_3\boldsymbol{w}_3^T\right)\boldsymbol{e}_{pk}. \end{aligned} \tag{31}$$

Because the eigenvectors $\boldsymbol{\omega}_j (j = 1, 2, 3)$ form a basis of $\mathbb{R}^3$. The residual vector $\boldsymbol{e}_{pk}$ can be represented by

$$\boldsymbol{e}_{pk} = e_1\boldsymbol{w}_1 + e_2\boldsymbol{w}_2 + e_3\boldsymbol{w}_3 \tag{32}$$

where e_j is the projected component of $\boldsymbol{e}_{pk}$ onto $\boldsymbol{\omega}_j$. Then, F_{pk} is calculated by

$$F_{pk} = \frac{1}{\gamma_1}e_1^2 + \frac{1}{\gamma_2}e_2^2 + \frac{1}{\gamma_3}e_3^2. \tag{33}$$

As can be seen from (29) and (33), the projected component of $\boldsymbol{e}_{pk}$ onto $\boldsymbol{\omega}_1$ will have the least contribution to F_{pk} among all the three components.

REFERENCES

[1] H. Gao, X. Zhang, J. Wen, J. Yuan, and Y. Fang, "Autonomous indoor exploration via polygon map construction and graph-based SLAM using directional endpoint features," *IEEE Trans. Autom. Sci. Eng.*, vol. 16, no. 4, pp. 1531–1542, Oct. 2019.

[2] J. Cheng, C. Wang, and M. Q.-H. Meng, "Robust visual localization in dynamic environments based on sparse motion removal," *IEEE Trans. Autom. Sci. Eng.*, vol. 17, no. 2, pp. 658–669, Apr. 2020.

[3] Q. Liang and M. Liu, "An automatic site survey approach for indoor localization using a smartphone," *IEEE Trans. Autom. Sci. Eng.*, vol. 17, no. 1, pp. 191–206, Jan. 2020.

[4] J. Wen, X. Zhang, H. Gao, J. Yuan, and Y. Fang, "CAE-RLSM: Consistent and efficient redundant line segment merging for online feature map building," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 7, pp. 4222–4237, Jul. 2020.

[5] Q. Sun, J. Yuan, X. Zhang, and F. Sun, "RGB-D SLAM in indoor environments with STING-based plane feature extraction," *IEEE/ASME Trans. Mechatronics*, vol. 23, no. 3, pp. 1071–1082, Jun. 2018.

[6] S. Yang, Y. Song, M. Kaess, and S. Scherer, "Pop-up SLAM: Semantic monocular plane SLAM for low-texture environments," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Daejeon, South Korea, Oct. 2016, pp. 1222–1229.

[7] M. Wu, S.-L. Dai, and C. Yang, "Mixed reality enhanced user interactive path planning for omnidirectional mobile robot," *Appl. Sci.*, vol. 10, no. 3, p. 1135, Feb. 2020.

[8] J. Yuan, F. Sun, and Y. Huang, "Trajectory generation and tracking control for double-steering tractor–trailer mobile robots with on-axle hitching," *IEEE Trans. Ind. Electron.*, vol. 62, no. 12, pp. 7665–7677, Dec. 2015.

[9] J. Yuan, H. Chen, F. Sun, and Y. Huang, "Multisensor information fusion for people tracking with a mobile robot: A particle filtering approach," *IEEE Trans. Instrum. Meas.*, vol. 64, no. 9, pp. 2427–2442, Sep. 2015.

[10] R. Cupec, E. K. Nyarko, D. Filko, A. Kitanov, and I. Petrovic, "Place recognition based on matching of planar surfaces and line segments," *Int. J. Robot. Res.*, vol. 34, nos. 4–5, pp. 674–704, Apr. 2015.

[11] J. Yuan *et al.*, "A novel approach to image-sequence-based mobile robot place recognition," *IEEE Trans. Syst., Man, Cybern. Syst.*, early access, Dec. 12, 2019, doi: 10.1109/TSMC.2019.2956321.

[12] H. Lin, T. Zhang, Z. Chen, H. Song, and C. Yang, "Adaptive fuzzy Gaussian mixture models for shape approximation in robot grasping," *Int. J. Fuzzy Syst.*, vol. 21, no. 4, pp. 1026–1037, Jun. 2019.

[13] F. Nardi, B. Della Corte, and G. Grisetti, "Unified representation and registration of heterogeneous sets of geometric primitives," *IEEE Robot. Autom. Lett.*, vol. 4, no. 2, pp. 625–632, Apr. 2019.

[14] Y. Yang and G. Huang, "Observability analysis of aided INS with heterogeneous features of points, lines, and planes," *IEEE Trans. Robot.*, vol. 35, no. 6, pp. 1399–1418, Dec. 2019.

[15] H. Zhang and C. Ye, "Plane-aided visual-inertial odometry for 6-DOF pose estimation of a robotic navigation aid," *IEEE Access*, vol. 8, pp. 90042–90051, 2020.

[16] Y. Taguchi, Y.-D. Jian, S. Ramalingam, and C. Feng, "SLAM using both points and planes for hand-held 3D sensors," in *Proc. IEEE Int. Symp. Mixed Augmented Reality (ISMAR)*, Atlanta, GA, USA, Nov. 2012, pp. 321–322.

[17] H. Cho, S. Yeon, H. Choi, and N. Doh, "Detection and compensation of degeneracy cases for IMU-Kinect integrated continuous SLAM with plane features," *Sensors*, vol. 18, no. 4, pp. 935–943, 2018.

[18] X. Zhang, W. Wang, X. Qi, Z. Liao, and R. Wei, "Point-plane SLAM using supposed planes for indoor environments," *Sensors*, vol. 19, no. 17, p. 3795, Sep. 2019.

[19] I. Aloise, B. D. Corte, F. Nardi, and G. Grisetti, "Systematic handling of heterogeneous geometric primitives in graph-SLAM optimization," *IEEE Robot. Autom. Lett.*, vol. 4, no. 3, pp. 2738–2745, Jul. 2019.

[20] Y. Zhou, H. Li, and L. Kneip, "Canny-VO: Visual odometry with RGB-D cameras based on geometric 3-D–2-D edge alignment," *IEEE Trans. Robot.*, vol. 35, no. 1, pp. 184–199, Feb. 2019.

[21] F. Schenk and F. Fraundorfer, "RESLAM: A real-time robust edge-based SLAM system," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2019, pp. 154–160.

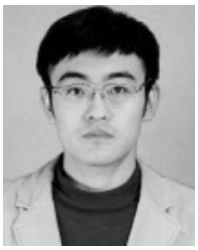
[22] S. Li and D. Lee, "RGB-D SLAM in dynamic environments using static point weighting," *IEEE Robot. Autom. Lett.*, vol. 2, no. 4, pp. 2263–2270, Oct. 2017.

[23] C. Choi, A. J. B. Trevor, and H. I. Christensen, "RGB-D edge detection and edge-based registration," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Tokyo, Japan, Nov. 2013, pp. 1568–1575.

[24] T.-H. Kwok, "DNSS: Dual-normal-space sampling for 3-D ICP registration," *IEEE Trans. Autom. Sci. Eng.*, vol. 16, no. 1, pp. 241–252, Jan. 2019.
[25] D. Simon, "Fast and accurate shape-based registration," Ph.D. dissertation, Robot. Inst., Carnegie Mellon Univ., Pittsburgh, PA, USA, Dec. 1996.
[26] N. Gelfand, L. Ikemoto, S. Rusinkiewicz, and M. Levoy, "Geometrically stable sampling for the ICP algorithm," in *Proc. 4th Int. Conf. 3-D Digit. Imag. Model. (3DIM)*, 2003, pp. 260–267.
[27] S. Rusinkiewicz and M. Levoy, "Efficient variants of the ICP algorithm," in *Proc. 3rd Int. Conf. 3-D Digit. Imag. Model.*, 2001, pp. 145–152.
[28] J. Wang, M. Garratt, and S. Anavatti, "Dominant plane detection using a RGB-D camera for autonomous navigation," in *Proc. 6th Int. Conf. Autom., Robot. Appl. (ICARA)*, Queenstown, New Zealand, Feb. 2015, pp. 456–460.
[29] J. Biswas and M. Veloso, "Planar polygon extraction and merging from depth images," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Vilamoura, Portugal, Oct. 2012, pp. 3859–3864.
[30] A. Trevor, S. Gedikli, R. Rusu, and H. Christensen, "Efficient organized point cloud segmentation with connected components," in *Proc. 3rd Workshop Semantic Perception Mapping Explor. (SPME)*, Karlsruhe, Germany, 2013, pp. 1–6.
[31] D. Belter, M. Nowicki, and P. Skrzypczynski, "Improving accuracy of feature-based RGB-D SLAM by modeling spatial uncertainty of point features," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2016, pp. 1279–1284.
[32] A. Belhedi, V. Gay-Bellile, A. Bartoli, K. Hamrouni, P. Sayd, and S. Bourgeois, "Noise modelling in time-of-flight sensors with application to depth noise removal and uncertainty estimation in three-dimensional measurement," *IET Comput. Vis.*, vol. 9, no. 6, pp. 967–977, Dec. 2015.
[33] K. Khoshelham and S. O. Elberink, "Accuracy and resolution of kinect depth data for indoor mapping applications," *Sensors*, vol. 12, no. 2, pp. 1437–1454, Feb. 2012.
[34] J.-H. Park, Y.-D. Shin, J.-H. Bae, and M.-H. Baeg, "Spatial uncertainty model for visual features using a kinect sensor," *Sensors*, vol. 12, no. 7, pp. 8640–8662, Jun. 2012.
[35] P. Fankhauser, M. Bloesch, D. Rodriguez, R. Kaestner, M. Hutter, and R. Siegwart, "Kinect v2 for mobile robot navigation: Evaluation and modeling," in *Proc. Int. Conf. Adv. Robot. (ICAR)*, Jul. 2015, pp. 388–394.
[36] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of RGB-D SLAM systems," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Vilamoura, Portugal, Oct. 2012, pp. 573–580.
[37] A. Handa, T. Whelan, J. McDonald, and A. J. Davison, "A benchmark for RGB-D visual odometry, 3D reconstruction and SLAM," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2014, pp. 1524–1531.
[38] K. Pathak, N. Vaskevicius, and A. Birk, "Revisiting uncertainty analysis for optimum planes extracted from 3D range sensor point-clouds," in *Proc. IEEE Int. Conf. Robot. Autom.*, Kobe, Japan, May 2009, pp. 2035–2040.
[39] Y. Chen and G. Medioni, "Object modeling by registration of multiple range images," in *Proc. IEEE Int. Conf. Robot. Autom.*, Sacramento, CA, USA, 2002, pp. 145–155.
[40] A. Segal, D. Haehnel, and S. Thrun, "Generalized-ICP," in *Proc. Robot., Sci. Syst.*, Seattle, WA, USA, 2009.
[41] I. Dryanovski, R. G. Valenti, and J. Xiao, "Fast visual odometry and mapping from RGB-D data," in *Proc. IEEE Int. Conf. Robot. Autom.*, Karlsruhe, Germany, May 2013, pp. 2305–2310.
[42] F. Endres, J. Hess, J. Sturm, D. Cremers, and W. Burgard, "3-D mapping with an RGB-D camera," *IEEE Trans. Robot.*, vol. 30, no. 1, pp. 177–187, Feb. 2014.
[43] C. Kerl, J. Sturm, and D. Cremers, "Dense visual SLAM for RGB-D cameras," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Tokyo, Japan, Nov. 2013, pp. 2100–2106.
[44] R. Kummerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "G^2o: A general framework for graph optimization," in *Proc. IEEE Int. Conf. Robot. Autom.*, Shanghai, China, May 2011, pp. 3607–3613.
[45] L. Ma, C. Kerl, J. Stuckler, and D. Cremers, "CPA-SLAM: Consistent plane-model alignment for direct RGB-D SLAM," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Stockholm, Sweden, May 2016, pp. 1285–1291.
[46] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras," *IEEE Trans. Robot.*, vol. 33, no. 5, pp. 1255–1262, Oct. 2017.
[47] A. Pumarola, A. Vakhitov, A. Agudo, A. Sanfeliu, and F. Moreno-Noguer, "PL-SLAM: Real-time monocular visual SLAM with points and lines," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 4503–4508.
[48] S. Maity, A. Saha, and B. Bhowmick, "Edge SLAM: Edge points based monocular visual SLAM," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops (ICCVW)*, Oct. 2017, pp. 2408–2417.
[49] T. Whelan, R. F. Salas-Moreno, B. Glocker, A. J. Davison, and S. Leutenegger, "ElasticFusion: Real-time dense SLAM and light source estimation," *Int. J. Robot. Res.*, vol. 35, no. 14, pp. 1697–1716, Dec. 2016.
[50] Y. Taguchi, Y.-D. Jian, S. Ramalingam, and C. Feng, "Point-plane SLAM for hand-held 3D sensors," in *Proc. IEEE Int. Conf. Robot. Autom.*, Karlsruhe, Germany, May 2013, pp. 5182–5189.

Qinxuan Sun received the B.Sc. degree in electronic information engineering from the Beijing University of Aeronautics and Astronautics, Beijing, China, in 2013, and the M.Sc. degree in control theory and control engineering from Nankai University, Tianjin, China, in 2016, where she is currently pursuing the Ph.D. degree.

Her current research interests include mobile robot navigation and simultaneous localization and mapping.



Jing Yuan (Member, IEEE) received the B.Sc. degree in automatic control and the Ph.D. degree in control theory and control engineering from Nankai University, Tianjin, China, in 2002 and 2007, respectively.

He has been with the Department of Automation, Nankai University, since 2007, where he is currently a Professor. His current research interests include robotic control, target tracking, and simultaneous localization and mapping.

Xuebo Zhang (Senior Member, IEEE) received the B.Eng. degree in automation from Tianjin University, Tianjin, China, in 2006, and the Ph.D. degree in control theory and control engineering from Nankai University, Tianjin, in 2011.

He has been with the Institute of Robotics and Automatic Information System, Nankai University, where he is currently a Professor. His current research interests include mobile robotics, motion planning, visual servoing, and simultaneous localization and mapping.

Dr. Zhang is a Technical Editor of the IEEE/ASME TRANSACTIONS ON MECHATRONICS and an Associate Editor of the *ASME Journal of Dynamic Systems, Measurement, and Control*.

Feng Duan (Member, IEEE) received the B.E. and M.E. degrees in mechanical engineering from Tianjin University, Tianjin, China, in 2002 and 2004, respectively, and the M.S. and Ph.D. degrees in precision engineering from The University of Tokyo, Tokyo, Japan, in 2006 and 2009, respectively.

He is currently a Professor with Nankai University, Tianjin, China. His research interests include cellular manufacture systems, rehabilitation robots, and brain–machine interfaces.