RGB-D SLAM in Indoor Environments With STING-Based Plane Feature Extraction

Qinxuan Sun, Jing Yuan, *Member, IEEE*, Xuebo Zhang, *Member, IEEE*, and Fengchi Sun

***Abstract*—In this paper, the RGB-D camera-based simultaneous localization and mapping (SLAM) of indoor environments is addressed using plane features. The plane parameter space (PPS) is defined for a compact representation of planes in the Cartesian space. The statistical information grid (STING) structure is constructed in the PPS to extract plane features. The plane association graph is developed to determine the correspondences between the plane features from two successive scans. The RGB-D camera pose is directly calculated using the matched plane features if they can provide sufficient constraints for the pose estimation. Otherwise, a novel STING-based scan matching method is developed in the PPS to achieve a full six degrees of freedom camera pose estimation. The proposed method uses only the plane features independent of any other features to estimate the RGB-D camera poses and can thus be suitable for some challenging scenes. The experimental results demonstrate that the proposed plane feature-based RGB-D SLAM can achieve high accuracy and robustness in both on-board and hand-held applications.**

***Index Terms*—Indoor environment mapping, plane feature, RGB-D camera, robot vision, six degrees of freedom (6-DoF) camera pose estimation.**

I. Introduction

RECENTLY, there is increasing demand for mobile robots to perform their given tasks in indoor industrial and office environments, which is significant for intelligent manufacture and service. Three-dimensional (3-D) mapping is of great importance for autonomous navigation of a mobile robot. It provides a prerequisite for localization [1], [2], navigation [3], [4], and path planning [5], [6]. The commercial launch of low-cost and light-weight RGB-D cameras, such as the Microsoft Kinect [7], Asus Xtion, and Carmine, offers an attractive alternative to other sensors, such as monocular vision sensors [8] and laser range finders [9], [10] for building 3-D maps of indoor environments. Motion estimation of the robot using an RGB-D camera mainly includes scan-matching-based methods [11]–[17] and feature-based methods [18]–[22].

The scan matching generally recovers the robot motion by calculating a transformation that best aligns two successive scans. Based on the error metrics, the scan-matching-based methods can be divided into three major groups, including geometric error-based scan matching [11], [12], photometric error-based scan matching [13]–[15], and their combination [16], [17]. The geometric error-based scan matching uses the geometric distance as an error metric. In [11], the consecutive functions, which were defined by extended Gaussian images created from two successive scans, were correlated via spherical harmonic analysis, resulting in a three degrees of freedom (3-DoF) rotation estimate. Then, the iterative closest point (ICP) algorithm was applied to refine the resultant rotation and simultaneously obtain the 3-DoF translation. Regarding photometric error-based scan matching [13]–[15], it is based on the photo-consistency assumption that a point in the world coordinate system observed by a camera at different poses yields the same brightness in the image. The combination of geometric and photometric errors was used in [16] and [17] for estimation of RGB-D camera poses. However, all the aforementioned scan matching methods may fail to track the camera pose when two successive scans are far apart in orientation, which usually causes problems in data association.

The feature-based methods need to extract features such as points [18], [19], edges [20], and planar patches [21], [22] from successive scans. Then, unknown correspondences between features are determined to estimate the robot pose. The point feature-based RGB-D simultaneous localization and mapping (SLAM) systems (e.g., [18]) extracted the point features from the RGB images as landmarks to compute the robot motion. The point feature-based methods may fail when the robot cannot extract sufficient point features in textureless scenes. The plane features, however, are less affected by the texture information and are more suitable for describing the spatial structure of indoor environments. In addition, they could also provide some semantic information crucial for scene interpretation, which may facilitate the robot performing various tasks.

In this paper, we propose a statistical information grid (STING) based plane extraction (STING-PE) algorithm and a plane feature-based RGB-D camera pose estimation method. Specifically, the plane parameter space (PPS) is defined, and the transformation between the PPS and Cartesian space is es-

Manuscript received February 24, 2017; revised August 30, 2017; accepted November 3, 2017. Date of publication November 14, 2017; date of current version June 12, 2018. Recommended by Technical Editor Prof. X. Chen. This work was supported in part by the Natural Science Foundation of China under Grant 61573196 and Grant 61573195, and in part by the Natural Science Foundation of Tianjin under Grant 15JCYBJC18800 and Grant 16JCYBJC18300. *(Corresponding author: Jing Yuan.)*

The authors are with the College of Computer and Control Engineering, Nankai University, Tianjin 300071, China (e-mail: sunqinxuan@outlook.com; nkyuanjing@gmail.com; zhangxuebo@nankai.edu.cn; fengchisun@nankai.edu.cn).

This paper has supplementary downloadable multimedia material available at http://ieeexplore.ieee.org provided by the authors.

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

Digital Object Identifier 10.1109/TMECH.2017.2773576

tablished. Then, the STING structure is constructed by dividing the PPS into multilevel grid cells with different resolutions. The points in each cell are approximately represented by a Gaussian distribution. The plane features are extracted via a top-down search of the STING structure. Then, the plane feature matching is fulfilled based on the plane association graph (PAG), whose nodes and edges represent planes and their geometric relationships. The matched plane features are directly used to estimate the RGB-D camera pose if the plane features can provide 6-DoF constraint for camera motion (6-DoF constraint case). Degeneracy occurs when the plane feature can provide only 5-DoF or 3-DoF constraints for camera motion (5-DoF and 3-DoF constraint cases). For the degenerate cases, a STING-based scan matching (STING-SM) method is developed in the PPS to provide extra constraints and calculate the 6-DoF camera pose. The main contributions and advantages of this paper are as follows.

1) The proposed method only uses the plane features. It provides a powerful alternative to the point feature-based RGB-D SLAM systems when few point features are extracted in textureless environments or the RGB-D camera is pointed to an area outside of the valid depth range.
2) The planes in the Cartesian space are represented as points in the PPS, which are organized in a hierarchical grid structure (STING). The plane extraction is executed via a top-down search in the STING, which is robust to the scale of planes and can facilitate real-time performance.
3) The plane features may not provide full 6-DoF constraint for camera motion estimation. When the degenerate cases (5-DoF and 3-DoF cases) are identified, STING-SM is performed to provide the extra 1-DoF (or 3-DoF) constraint by aligning two scans in the PPS, which brings the three constraint cases into a unified framework.

The rest of the paper is organized as follows. The system overview is presented in Section II. The plane extraction and matching based on the STING structure are discussed in Section III. The plane feature-based RGB-D camera pose estimation is addressed in Section IV. A thorough experimental evaluation is presented in Section V. Conclusions are presented in Section VI.

II. System Overview

In this paper, the points and planes are represented in three different spaces. We use the left superscripts C, P, and R to represent the Cartesian space, PPS, and RGB space, respectively. For plane matching and RGB-D camera pose estimation, we consider two successive scans, i.e., the current scan and the reference scan, which are indicated by the right subscripts c and r, respectively. For an overview of the proposed method, the following notations are used:

${}^C\Omega$	Points in the Cartesian space;
${}^C\Omega_\pi$	Local planes in the Cartesian space;
${}^P\Omega$	Points in the PPS;
O	Cells in the STING structure;
$\Pi_c(\Pi_r)$	Planes extracted from the current (reference) scan;
$\Pi_{mc}(\Pi_{mr})$	Matched planes;
$\mathbf{R}, \mathbf{t}$	Rotation and translation between the current and reference frames.

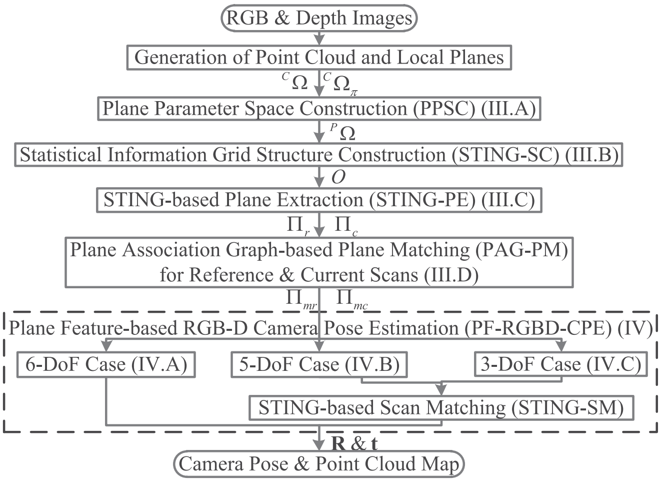

RGB & Depth Images
Generation of Point Cloud and Local Planes
${}^C\Omega$ ${}^C\Omega_\pi$
Plane Parameter Space Construction (PPSC) (III.A)
${}^P\Omega$
Statistical Information Grid Structure Construction (STING-SC) (III.B)
O
STING-based Plane Extraction (STING-PE) (III.C)
Π_r Π_c
Plane Association Graph-based Plane Matching (PAG-PM) for Reference & Current Scans (III.D)
Π_{mr} Π_{mc}
Plane Feature-based RGB-D Camera Pose Estimation (PF-RGBD-CPE) (IV)
6-DoF Case (IV.A)
5-DoF Case (IV.B)
3-DoF Case (IV.C)
STING-based Scan Matching (STING-SM)
R & t
Camera Pose & Point Cloud Map

Fig. 1. System overview.

A systematic overview of the proposed method is shown in Fig. 1. The input to the entire system is the depth and RGB images used for pose estimation at each time step (as the current scan) and those at the previous time step (as the reference scan). The output is the estimate of the camera pose.

First, ${}^P\Omega$ is obtained by the PPS construction (PPSC) module with ${}^C\Omega$ and ${}^C\Omega_\pi$ as inputs. Cells O in the STING are then constructed using ${}^P\Omega$ through the STING structure construction (STING-SC) module. Planes Π_c and Π_r are extracted from current and reference scans, respectively, by the STING-PE module and matched by the PAG-based plane matching (PAG-PM) module, which outputs the matched planes Π_{mc} and Π_{mr}. Afterward, the plane feature-based RGB-D camera pose estimation (PF-RGBD-CPE) module computes the camera pose by aligning Π_{mc} and Π_{mr}. When the degenerate cases (5-DoF and 3-DoF) are identified, the STING-SM module is activated to provide the extra constraints of the camera motion, yielding the complete 6-DoF pose estimate.

III. Plane Extraction and Matching

A. Construction of the PPS

For a point ${}^C\mathbf{p} \in \mathbb{R}^3$ in the Cartesian coordinate system, if ${}^C\mathbf{p}$ satisfies ${}^C\mathbf{n}^T\,{}^C\mathbf{p} + {}^C d = 0$, it is on the plane represented by ${}^C\pi = [{}^C\mathbf{n}^T, {}^C d]^T$, where ${}^C\mathbf{n} = [{}^C n_x, {}^C n_y, {}^C n_z]^T \in \mathbb{R}^3$ is the unit normal of the plane and ${}^C d \in \mathbb{R}$ is the vertical distance from the origin to the plane. The PPS is defined based on the plane parameters. The coordinates of a point in the PPS are denoted by ${}^P\mathbf{p} = [{}^P\theta, {}^P\varphi, {}^P d]^T \in \mathbb{R}^3$, ${}^P\theta \in [0, \pi]$, ${}^P\varphi \in (-\pi, \pi]$, where

$$\begin{cases} {}^P\theta = \arccos\left({}^C n_z\right) \\ {}^P\varphi = \operatorname{atan2}\left({}^C n_y, {}^C n_x\right) \\ {}^P d = {}^C d \end{cases}. \tag{1}$$

As shown in Fig. 2(a), a point in the PPS corresponds to a plane in the Cartesian space. In indoor environments, the point cloud data are acquired from surfaces, which are normally locally planar. Let ${}^C\Omega = \{{}^C\mathbf{p}_i, i = 1, \ldots, N\}$ be a set of points in the Cartesian coordinate system and

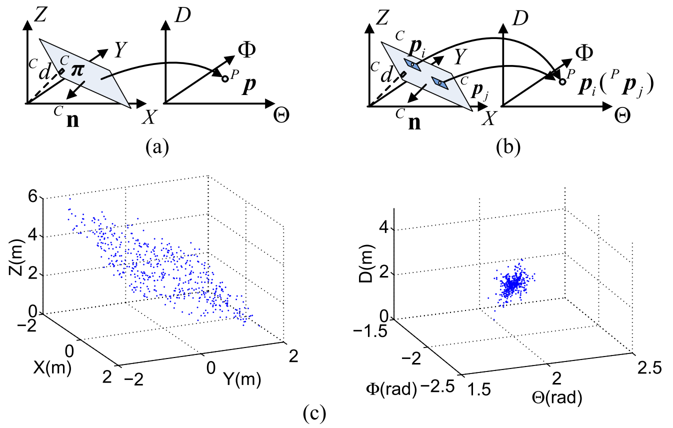


Fig. 2. (a) Correspondence between a plane ${}^C\pi$ (left) in the Cartesian space and a point ${}^P\mathbf{p}$ (right) in the PPS. (b) Correspondence between the local planes of two points ${}^C\mathbf{p}_i$ and ${}^C\mathbf{p}_j$ (left) in the Cartesian space and two points ${}^P\mathbf{p}_i$ and ${}^P\mathbf{p}_j$ (right) in the PPS. (c) Correspondence between plane measurement (left) in the Cartesian space and the points (right) in the PPS.

${}^C\Omega_\pi = \{{}^C\pi_i, i = 1, \ldots, N\}$ be the corresponding local planes estimated by [23]. They are projected into the PPS via (1) to yield ${}^P\Omega = \{{}^P\mathbf{p}_i, i = 1, \ldots, N\}$.

On the other hand, if the coordinates of a point ${}^P\mathbf{p}_i$ in the PPS are given, its corresponding plane ${}^C\pi_i$ in the Cartesian coordinate system can be computed by

$$ {}^C\pi_i = \begin{bmatrix} {}^C\mathbf{n}_i \\ {}^C d_i \end{bmatrix} = \begin{bmatrix} \sin{}^P\theta_i \cos{}^P\varphi_i \\ \sin{}^P\theta_i \sin{}^P\varphi_i \\ \cos{}^P\theta_i \\ {}^P d_i \end{bmatrix}. \tag{2}$$

For two points ${}^C\mathbf{p}_i$ and ${}^C\mathbf{p}_j$ lying on the same plane in the Cartesian space, as shown in Fig. 2(b), the corresponding points ${}^P\mathbf{p}_i$ and ${}^P\mathbf{p}_j$ in the PPS share the same coordinates. However, the measured points in the Cartesian coordinate system are normally affected by sensor noise. As a result, the coordinates of these points in the PPS are not exactly equal to one another. Fig. 2(c) shows a real plane represented in the Cartesian coordinate system, which is composed of measured points affected by sensor noise, as well as the corresponding points in the PPS. Although those points in the PPS are not completely coincident, they tend to follow a concentrated distribution. Therefore, plane extraction and modeling in the Cartesian space can be converted into a problem of fitting a distribution for the points in the PPS. To this end, we extend the basic STING structure [24] to the 3-D case and use it to organize the points in the PPS.

The mapping between a plane in the Cartesian space and a point in the PPS is a one-to-one mapping given ${}^P\theta \in [0, \pi], {}^P\varphi \in (-\pi, \pi]$. A singularity occurs when ${}^C\mathbf{n} = [0, 0, \pm 1]^T$. However, the singularity can be completely eliminated in theory by rotating the camera coordinate system before the plane is projected into the PPS such that the parameters of the rotated plane satisfy ${}^C\mathbf{n} \neq [0, 0, \pm 1]^T$. In the implementation, there are many local planes in the Cartesian space. We find a common rotation for all the planes to generate a mapping between two spaces far from the singularity. The rotation is obtained by a simple PCA analysis on the normal vectors of the planes. Define a matrix C_R as

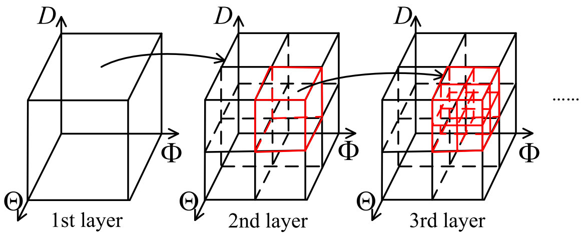


Fig. 3. Three highest levels of the STING structure built in the PPS.

$$C_R = \frac{1}{N}\sum_{i=1}^{N} {}^C\mathbf{n}_i \, {}^C\mathbf{n}_i^T. \tag{3}$$

The eigenvalue decomposition of C_R is computed by $C_R = \mathbf{Q}_R \mathbf{\Lambda}_R \mathbf{Q}_R^T$, where $\mathbf{\Lambda}_R = \text{diag}\{\lambda_{R,1}, \lambda_{R,2}, \lambda_{R,3}\}$ and $\mathbf{Q}_R = [\mathbf{q}_{R,1}, \mathbf{q}_{R,2}, \mathbf{q}_{R,3}]$ denote the eigenvalues and the corresponding eigenvectors, respectively. Assuming that $\lambda_{R,1} \geq \lambda_{R,2} \geq \lambda_{R,3}$, rotation R_{PCA} is defined as

$$R_{\text{PCA}} = \begin{bmatrix} \dfrac{\mathbf{q}_{R,1} \times \mathbf{q}_{R,3}}{\|\mathbf{q}_{R,1} \times \mathbf{q}_{R,3}\|} & \mathbf{q}_{R,1} & \mathbf{q}_{R,3} \end{bmatrix}^T. \tag{4}$$

The normal vectors of the planes are left-multiplied by R_{PCA} before the planes ${}^C\Omega_\pi$ are projected into the PPS. Likewise, the normal vectors of the planes are left-multiplied by R_{PCA}^T after the points ${}^P\Omega$ are projected back into the Cartesian space, such that the original plane parameters ${}^C\Omega_\pi$ can be obtained. Note that, the aforementioned process is performed independent of any other calculation in either space. Furthermore, the impact caused by a singularity is eliminated in actual experiments on both public datasets and real scenes by performing this scheme.

B. STING Structure

The basic STING structure has been proposed in [24] for spatial data mining. The spatial area is divided into multilevel grid cells corresponding to different resolutions. Each cell at a high level is partitioned into several cells at the next lower level. A certain kind of distribution is assigned to the data in each cell. The distribution parameters of the bottom-level cells are calculated directly from the data inside. The parameters of the higher-level cells can be easily obtained from those of the lower-level cells. However, the basic STING structure in [24] was developed for representing and fitting a distribution of scalars, and it cannot be directly applied to represent vector parameters of a high-dimensional distribution. Therefore, we extend the basic STING to make it suitable for computing the vector parameters in this paper.

As shown in Fig. 3,we divide the PPS into a hierarchical grid structure. For the data points in each cell, a Gaussian distribution is adopted to fit their coordinates ${}^P\mathbf{p} = [{}^P\theta, {}^P\varphi, {}^P d]^T \in \mathbb{R}^3$ in the PPS as well as the coordinates ${}^R\mathbf{p} = [{}^R r, {}^R g, {}^R b]^T \in \mathbb{R}^3$ of their corresponding color image pixels in the RGB space.

We assume that the STING structure has L levels ($L = 5$ in our implementation) and that each cell of the lth level ($l = 1, \ldots, L - 1$) corresponds to the union of areas of its N_s

children ($N_s = 8$ in our implementation) at the $(l+1)$th level. Fig. 3 shows the spatial relations between adjacent levels.

The cells in the constructed STING structure are denoted by $O = \{o_{lk}\}, k = 1, 2, \ldots, (N_s)^{l-1}, l = 1, 2, \ldots, L$. The quintuple $o_{lk} = ({}^P\mathbf{m}_{lk}, {}^R\mathbf{m}_{lk}, {}^P\mathbf{S}_{lk}, {}^R\mathbf{S}_{lk}, c_{lk})$ represents the kth cell at the lth level, where ${}^P\mathbf{m}_{lk}$ and ${}^R\mathbf{m}_{lk}$ are the means of the Gaussian distributions in the PPS and in the RGB space, respectively, ${}^P\mathbf{S}_{lk}$ and ${}^R\mathbf{S}_{lk}$ are their covariance matrices, and c_{lk} is the number of the data points in the cell. Meanwhile, the jth child of o_{lk} is denoted by $o_{lkj} = ({}^P\mathbf{m}_{\mathrm{lkj}}, {}^R\mathbf{m}_{\mathrm{lkj}}, {}^P\mathbf{S}_{\mathrm{lkj}}, {}^R\mathbf{S}_{\mathrm{lkj}}, c_{\mathrm{lkj}})$, $j = 1, 2, \ldots, N_s$. The point numbers c_{lk} and c_{lkj} satisfy $c_{lk} = \sum_{j=1}^{N_s} c_{lkj}$. For ${}^P\mathbf{p}$ and ${}^R\mathbf{p}$, we extend the method of the basic STING to fit the vector parameters. ${}^P\mathbf{m}_{lk}$ and ${}^P\mathbf{S}_{lk}$ can be calculated as

$$ {}^P\mathbf{m}_{lk} = \frac{1}{c_{lk}} \sum_{j=1}^{N_s} {}^P\mathbf{m}_{lkj} c_{lkj} \tag{5} $$

$$ {}^P\mathbf{S}_{lk} = \frac{1}{c_{lk}} \sum_{j=1}^{N_s} c_{lkj} \left({}^P\mathbf{S}_{lkj} + {}^P\mathbf{m}_{lkj} {}^P\mathbf{m}_{lkj}^{\;T} \right) - {}^P\mathbf{m}_{lk} {}^P\mathbf{m}_{lk}^{\;T} \tag{6} $$

${}^R\mathbf{m}_{lk}$ and ${}^R\mathbf{S}_{lk}$ can be computed likewise.

C. STING Based Plane Extraction

If sufficient points concentrate in a small area in the PPS, their corresponding local planes in the Cartesian space are highly likely to be on the same plane, as shown in Fig. 2(b). Based on this fact, the STING-PE method is proposed. With the hierarchical STING structure in hand, STING-PE is fulfilled by a top-down search whose purpose is to find the cells containing sufficient data points that follow a concentrated distribution in the PPS. For such a cell, the extracted plane feature can be calculated using $o = ({}^P\mathbf{m}, {}^R\mathbf{m}, {}^P\mathbf{S}, {}^R\mathbf{S}, \quad c)$. Let $P_\pi = ({}^C\pi, {}^R\mathbf{m}, {}^R\mathbf{S}, \quad c)$ denote the plane feature, where ${}^C\pi = [{}^C\mathbf{n}^T, {}^C d]^T$ is calculated from ${}^P\mathbf{m} = [{}^P\theta, {}^P\varphi, {}^P d]^T$ via (2), ${}^R\mathbf{m}$ and ${}^R\mathbf{S}$ represent the mean and covariance matrix of the RGB value of the points on the plane, respectively, and c is the number of the points on the plane.

Algorithm 1 gives the entire STING-PE process. In lines 2–6, each cell of O is initialized as "not relevant". In lines 7–11, cells at the l_0th level with sufficient points inside are labeled "relevant". In lines 12–27, for the "relevant" cells at the lth level, their children containing sufficient inside points that follow a concentrated distribution are extracted, and their parameters o_{lkj} are used to describe a plane P_π (line 19). The children with sufficient scattered points are labeled "relevant" (line 21). Because o_{lkj} is on the $(l+1)$th level, when traversing the lower $(l+1)$th level, o_{lkj} will be treated as a "relevant" cell.

The threshold ε_n in Algorithm 1 is determined by experiments and $\varepsilon_n = 500$ can achieve fairly good performance. The threshold ε_s is set to be 0.01 in our experiments.

By means of the STING-PE algorithm, planes $\Pi = \{P_{\pi,i} = ({}^C\pi_i, {}^R\mathbf{m}_i, {}^R\mathbf{S}_i, \quad c_i), i = 1, \ldots, N_\pi\}$ are extracted from the current scan, where N_π is the number of planes.

Algorithm 1: STING-PE.

```
inputs:
    STING structure $O$.
outputs:
    Plane set
    $\Pi = \{P_{\pi,i} = ({}^C\pi_i, {}^R\mathbf{m}_i, {}^R\mathbf{S}_i, \quad c_i), i = 1, \cdots, N_\pi\}$.
 1: Start from the $l_0$-th level. Set $i = 0, \Pi = \emptyset$.
 2: for $l = l_0, \cdots, L-1$ do
 3:     for $k = 1, \cdots, (N_s)^{l-1}$ do
 4:         Label the cell $o_{lk}$ as "not relevant".
 5:     end for
 6: end for
 7: for $k = 1, \cdots, (N_s)^{l_0-1}$ do
 8:     if $c_{l_0k} > \varepsilon_n$ then
 9:         Label the cell $o_{l_0k}$ as "relevant".
10:     end if
11: end for
12: for $l = l_0, \cdots, L-1$ do
13:     for $k = 1, \cdots, (N_s)^{l-1}$ do
14:         if the cell $o_{lk}$ is labeled "relevant" then
15:             for $j = 1, \cdots, N_s$ do
16:                 if $c_{lkj} > \varepsilon_n$ then
17:                     if $\lambda_{lkj} < \varepsilon_s$ then
18:                         $i \leftarrow i + 1$.
19:                         Add $P_{\pi,i} = ({}^C\pi_i, {}^R\mathbf{m}_i, {}^R\mathbf{S}_i, c_i)$
                             to set $\Pi$.
20:                     else
21:                         Label the cell $o_{lkj}$ as "relevant".
22:                     end if
23:                 end if
24:             end for
25:         end if
26:     end for
27: end for
```

D. PAG-Based Plane Matching

The PAG-PM is employed to set up correspondences between two plane sets extracted from two successive scans, respectively. The PAG is a graph built for each scan, wherein nodes represent the extracted planes and edges represent the geometric relationships between the planes.

Consider two plane sets $\Pi_c = \{P_{\pi c,i} = ({}^C\pi_{c,i}, {}^R\mathbf{m}_{c,i}, {}^R\mathbf{S}_{c,i}, c_{c,i}), i = 1, \ldots, N_{\pi c}\}$ and $\Pi_r = \{P_{\pi r,k} = ({}^C\pi_{r,k}, {}^R\mathbf{m}_{r,k}, {}^R\mathbf{S}_{r,k}, c_{r,k}), k = 1, \cdots, N_{\pi r}\}$ extracted from the current and reference scan, respectively. The geometric relationship between two planes can be classified into two categories, nonparallel and parallel. The relationship between two nonparallel planes can be measured by the angle between their normal vectors, and that between two parallel planes by their vertical distance. For two planes $P_{\pi c,i}, P_{\pi c,j} \in \Pi_c$, the angle $\alpha_{c,ij} \in [0, \pi]$ between their normal vectors is

$$ \alpha_{c,ij} = \arccos({}^C\mathbf{n}_{c,i}^T \, {}^C\mathbf{n}_{c,j}). \tag{7} $$

$P_{\pi c,i}$ and $P_{\pi c,j}$ are regarded as parallel if $\alpha_{c,ij} < \varepsilon_{\alpha 1}$ ($\varepsilon_{\alpha 1} = 15°$ in our implementation), and nonparallel otherwise. The vertical distance between parallel planes $P_{\pi c,i}$ and $P_{\pi c,j}$ is

$$d_{c,ij} = \left|{}^{C}d_{c,i} - {}^{C}d_{c,j}\right|. \tag{8}$$

Then, we build the PAG $G_c = (V_c, E_c)$ for Π_c, where V_c=$\{v_{c,i} = P_{\pi c,i}, i = 1, \ldots, N_{\pi c}\}$ is the set of nodes, and $E_c = \{e_{c,ij}, i, j = 1, \ldots, N_{\pi c}, i \neq j\}$ is the set of edges. The edge $e_{c,ij}$ is defined as

$$e_{c,ij} = (\omega_{c,ij}, \alpha_{c,ij}, d_{c,ij}), i, j = 1, \ldots, N_{\pi c}, i \neq j \tag{9}$$

where $\omega_{c,ij} \in \{\text{parallel,not parallel}\}$ is an enumeration variable

$$\omega_{c,ij} = \begin{cases} \text{parallel} & \text{if } \alpha_{c,ij} < \varepsilon_{\alpha 1} \\ \text{not parallel} & \text{otherwise} \end{cases} \tag{10}$$

and $\alpha_{c,ij}$ can be calculated by (7). The distance $d_{c,ij}$ satisfies

$$d_{c,ij} = \begin{cases} \left|{}^{C}d_{c,i} - {}^{C}d_{c,j}\right| & \text{if } \omega_{c,ij} = \text{parallel} \\ 0 & \text{otherwise} \end{cases}. \tag{11}$$

The PAG $G_r = (V_r, E_r)$ for the plane set Π_r in the reference scan is constructed likewise.

For two edges $e_{c,ij} \in E_c$ and $e_{r,kl} \in E_r$ in two PAGs $G_c = (V_c, E_c)$ and $G_r = (V_r, E_r)$, respectively, we define the relationship between $e_{c,ij}$ and $e_{r,kl}$ as

$$\begin{cases} e_{c,ij} = e_{r,kl} & \text{if } \omega_{c,ij} = \omega_{r,kl} \text{ and } |\alpha_{c,ij} - \alpha_{r,kl}| < \varepsilon_{\alpha 2} \\ & \quad \text{and } |d_{c,ij} - d_{r,kl}| < \varepsilon_d \\ e_{c,ij} \neq e_{r,kl} & \text{otherwise} \end{cases}. \tag{12}$$

The thresholds in (12) are chosen as $\varepsilon_{\alpha 2} = 5°$ and $\varepsilon_d = 0.06$ m in the experiments.

For any two nodes $v_{c,i} \in V_c$ and $v_{r,k} \in V_r$, we define the similarity $s(v_{c,i}, v_{r,k})$ between $v_{c,i}$ and $v_{r,k}$ by

$$s(v_{c,i}, v_{r,k}) = s_{\text{col}}(v_{c,i}, v_{r,k}) + s_{\text{geo}}(v_{c,i}, v_{r,k}) \tag{13}$$

where $s_{\text{col}}(v_{c,i}, v_{r,k})$ and $s_{\text{geo}}(v_{c,i}, v_{r,k})$ represent the color and geometric similarity, respectively. The color similarity $s_{\text{col}}(v_{c,i}, v_{r,k})$ is defined as the Bhattacharyya distance between the color distributions of two planes as follows:

$$\begin{aligned} s_{\text{col}}(v_{c,i}, v_{r,k}) = {} & \frac{1}{8}\left({}^{R}\mathbf{m}_{c,i} - {}^{R}\mathbf{m}_{r,k}\right)^{T} {}^{R}\mathbf{S}^{-1} \left({}^{R}\mathbf{m}_{c,i} - {}^{R}\mathbf{m}_{r,k}\right) \\ & + \frac{1}{2}\ln\left(\frac{\left|{}^{R}\mathbf{S}\right|}{\sqrt{\left|{}^{R}\mathbf{S}_{c,i}\right| \cdot \left|{}^{R}\mathbf{S}_{r,k}\right|}}\right) \end{aligned} \tag{14}$$

where ${}^{R}\mathbf{S} = \frac{{}^{R}\mathbf{S}_{c,i} + {}^{R}\mathbf{S}_{r,k}}{2}$. The geometric similarity $s_{\text{geo}}(v_{c,i}, v_{r,k})$ is defined by the sum of color similarity of the nodes connected to $v_{c,i}$ and $v_{r,k}$ by similar edges [25]

$$\begin{aligned} & s_{\text{geo}}(v_{c,i}, v_{r,k}) \\ & = \frac{1}{\left|I_{v_{c,i}|v_{r,k}}\right|} \sum_{t=1}^{\left|I_{v_{c,i}|v_{r,k}}\right|} s_{\text{col}}\left(I_{v_{c,i}|v_{r,k}}[t], I_{v_{r,k}|v_{c,i}}[t]\right) \end{aligned} \tag{15}$$

where $I_{v_{c,i}|v_{r,k}}$ and $I_{v_{r,k}|v_{c,i}}$ are two index sets. For the nodes $\{v_{c,j}, j = 1, \ldots, N_{\pi c}, j \neq i\}$ and $\{v_{r,l}, l = 1, \ldots, N_{\pi r}, l \neq k\}$, if $e_{c,ij}$ and $e_{r,kl}$ satisfy $e_{c,ij} = e_{r,kl}$, then $v_{c,j}$ and $v_{r,l}$ are pushed into $I_{v_{c,i}|v_{r,k}}$ and $I_{v_{r,k}|v_{c,i}}$, respectively. In other words, the tth element $I_{v_{c,i}|v_{r,k}}[t]$ in $I_{v_{c,i}|v_{r,k}}$ and $I_{v_{r,k}|v_{c,i}}[t]$ in $I_{v_{r,k}|v_{c,i}}$ represent the nodes connected with $v_{c,i}$ and $v_{r,k}$ by the similar edges in G_c and G_r, respectively. $|I_{v_{c,i}|v_{r,k}}|$ and $|I_{v_{r,k}|v_{c,i}}|$ represent the numbers of the elements in $I_{v_{c,i}|v_{r,k}}$ and $I_{v_{r,k}|v_{c,i}}$, respectively, and satisfy $|I_{v_{c,i}|v_{r,k}}| = |I_{v_{r,k}|v_{c,i}}|$. Each pair of planes from Π_c and Π_r is examined by the similarity (13) to find correspondence between them. The number of planes extracted from one single frame is relatively small. Therefore, the traversal of the PAGs constructed from two successive scans is timesaving.

Algorithm 2: PF-RGBD-CPE.

```
inputs:
    Matched plane set $\Pi_{mc}$ and $\Pi_{mr}$.
outputs:
    The transformation between two successive scans $\mathbf{R}$ and $\mathbf{t}$.
Compute matrix $\mathbf{H}$ and its SVD.
if $\lambda_1 \geq \lambda_2 \geq \lambda_3 > 0$ then
    // 6-DoF case
    Compute $\mathbf{R}$ and $\mathbf{t}$ using (19) and (20), respectively.
else
    if $\lambda_1 \geq \lambda_2 > \lambda_3 = 0$ then
        // 5-DoF case
        Compute $\mathbf{R}$ and $\mathbf{t}_1$ using (22) and (24), respectively.
        Compute $\mathbf{t}_2$ by minimizing (29).
        $\mathbf{t} = \mathbf{t}_1 + \mathbf{t}_2$.
    else
        // 3-DoF case
        Compute $\mathbf{R}_1$ and $\mathbf{t}_1$ using (31) and (33), respectively.
        Compute $\mathbf{R}_2$ and $\mathbf{t}_2$ by minimizing (46).
        $\mathbf{R} = \mathbf{R}_2\mathbf{R}_1$ and $\mathbf{t} = \mathbf{t}_1 + \mathbf{t}_2$.
    end if
end if
```

IV. RGB-D Camera Pose Estimation

In this section, the PE-RGBD-CPE method is proposed to estimate the 6-DoF RGB-D camera pose $\mathbf{R}$ and $\mathbf{t}$. Consider two sets of matched planes $\Pi_{mc} = \{P_{\pi c,i} = ({}^{C}\pi_{c,i}, {}^{R}\mathbf{m}_{c,i}, {}^{R}\mathbf{S}_{c,i},\ c_{c,i}), i = 1, \ldots, N_{\pi}\}$ and Π_{mr} = $\{P_{\pi r,i} = ({}^{C}\pi_{r,i}, {}^{R}\mathbf{m}_{r,i}, {}^{R}\mathbf{S}_{r,i},\ c_{r,i}), i = 1, \ldots, N_{\pi}\}$, where $\{P_{\pi c,i}, P_{\pi r,i}\}$ is a pair of matched planes, and N_{π} is the number of the matched plane pairs.

The matched planes can provide three kinds of constraints for the camera pose estimation. To distinguish them, a matrix $\mathbf{H} = \sum_{i=1}^{N_{\pi}} {}^{C}\mathbf{n}_{c,i} {}^{C}\mathbf{n}_{r,i}^{T}$ is defined [26] and the singular value decomposition (SVD) is performed for $\mathbf{H}$ as follows:

$$\mathbf{H} = \mathbf{U}\mathbf{\Lambda}\mathbf{V}^{T} = \lambda_1\mathbf{u}_1\mathbf{v}_1^{T} + \lambda_2\mathbf{u}_2\mathbf{v}_2^{T} + \lambda_3\mathbf{u}_3\mathbf{v}_3^{T} \tag{16}$$

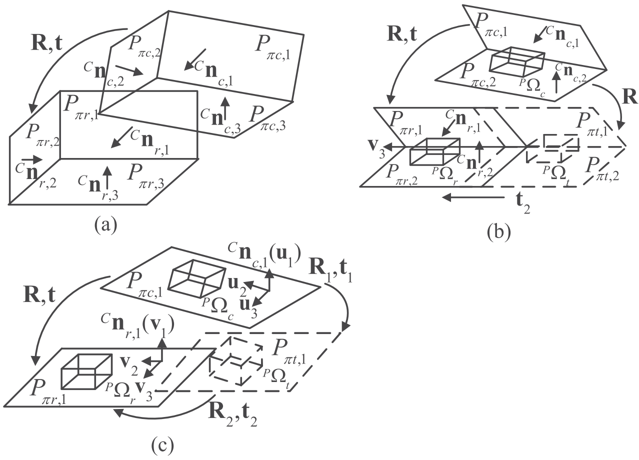


Fig. 4. Three examples corresponding to three cases of constraints: (a) three nonparallel plane pairs $\{P_{\pi c,i}, P_{\pi r,i}\}, i = \{1,2,3\}$ from the current scan and the reference scan, respectively, for the 6-DoF case, (b) two nonparallel plane pairs $\{P_{\pi c,i}, P_{\pi r,i}\}, i = \{1,2\}$ for the 5-DoF case, and (c) one plane pair $\{P_{\pi c,1}, P_{\pi r,1}\}$ for the 3-DoF case.

where $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3]$ and $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3]$ are both orthonormal matrices, and $\mathbf{\Lambda} = \mathrm{diag}\{\lambda_1, \lambda_2, \lambda_3\}$ $(\lambda_1 \geq \lambda_2 \geq \lambda_3)$. Because $P_{\pi c,i}$ is corresponding to $P_{\pi r,i}$, for any $i \in \{1,2,\ldots,N_\pi\}$ and $j \in \{1,2,3\}$, if ${}^C\mathbf{n}_{c,i}^T\mathbf{u}_j = 0$ holds, then we have ${}^C\mathbf{n}_{r,i}^T \mathbf{v}_j = 0$.

The three kinds of constraint cases can be distinguished online according to the singularity values of $\mathbf{H}$. For all the three cases, $\mathbf{R}$ and $\mathbf{t}$ are computed by minimizing (17) and (18), respectively,

$$J_R(\mathbf{R}) = \sum_{i=1}^{N_\pi} \left\| {}^C\mathbf{n}_{r,i} - \mathbf{R} \cdot {}^C\mathbf{n}_{c,i} \right\|^2 \tag{17}$$

$$J_t(\mathbf{t}) = \sum_{i=1}^{N_\pi} \left({}^C d_{r,i} - \left({}^C d_{c,i} + {}^C\mathbf{n}_{r,i}^T \mathbf{t} \right) \right)^2. \tag{18}$$

In what follows, the detailed method is discussed for three different cases, respectively. Accordingly, the pseudo-code is given in Algorithm 2.

A. 6-DoF Constraint Case

If $\mathbf{H}$ is nonsingular, at least three pairs of nonparallel planes are matched between Π_{mc} and Π_{mr}, as shown in Fig. 4(a). The rotation $\mathbf{R}$ to minimize (17) can be calculated by [26]

$$\mathbf{R} = \mathbf{U}\mathbf{V}^T. \tag{19}$$

And $\mathbf{t}$ to minimize (18) can be obtained through the least-square method

$$\mathbf{t} = \left(\mathbf{A}^T\mathbf{A}\right)^{-1}\mathbf{A}^T\mathbf{d} \tag{20}$$

where

$$\mathbf{A} = \begin{bmatrix} {}^C\mathbf{n}_{r,1}^T \\ {}^C\mathbf{n}_{r,2}^T \\ \cdots \\ {}^C\mathbf{n}_{r,\mathrm{N}_\pi}^T \end{bmatrix} \quad \mathbf{d} = \begin{bmatrix} {}^C d_{r,1} - {}^C d_{c,1} \\ {}^C d_{r,2} - {}^C d_{c,2} \\ \cdots \\ {}^C d_{r,\mathrm{N}_\pi} - {}^C d_{c,\mathrm{N}_\pi} \end{bmatrix}. \tag{21}$$

B. 5-DoF Constraint Case

If $\mathbf{H}$ is singular and $\lambda_1 \geq \lambda_2 > \lambda_3 = 0$, then $\{{}^C\mathbf{n}_{c,i}, i = 1,2,\ldots,N_\pi\}$ $(\{{}^C\mathbf{n}_{r,i}, i = 1,2,\ldots,N_\pi\})$ are coplanar and vertical to the $\mathbf{u}_3(\mathbf{v}_3)$. Whereas, in practical applications, λ_3 may not be exactly equal to zero due to noise. In our implementation, λ_3 is regarded as zero when it satisfies $\lambda_2 > 10\lambda_3$. In Fig. 4(b), two correspondences are established between two pairs of nonparallel planes from Π_{mc} and Π_{mr}. In this case, $\mathbf{R}$ is computed by [26]

$$\mathbf{R} = \begin{cases} \mathbf{U}\mathbf{V}^T & \text{if} \quad \det(\mathbf{U}\mathbf{V}^T) = 1 \\ \mathbf{U}'\mathbf{V}^T & \text{if} \quad \det(\mathbf{U}\mathbf{V}^T) = -1 \end{cases} \tag{22}$$

where $\mathbf{U}' = [\mathbf{u}_1, \mathbf{u}_2, -\mathbf{u}_3]$.

The translation vector $\mathbf{t}$ cannot be directly solved by (20) because $\det(\mathbf{A}^T\mathbf{A}) = 0$. Note that, the translation along the direction of $\mathbf{v}_3$ does not change the plane parameters of $P_{\pi r,1}$ and $P_{\pi r,2}$. Therefore, the translation along $\mathbf{v}_3$ cannot be constrained between these two scans. Let the component $\mathbf{t}_2$ of the translation along $\mathbf{v}_3$ be zero, and define the objective function $J_{1,t}(\mathbf{t})$ as

$$J_{1,t}(\mathbf{t}) = \sum_{i=1}^{N_\pi} \left({}^C d_{r,i} - \left({}^C d_{c,i} + {}^C\mathbf{n}_{r,i}^T \mathbf{t} \right) \right)^2 + \left(\mathbf{v}_3^T \mathbf{t} \right)^2. \tag{23}$$

Then, $\mathbf{t}_1$ can be solved by minimizing $J_{1,t}(\mathbf{t})$ with the least-square method as follows:

$$\mathbf{t}_1 = \left(\mathbf{A}_1^T\mathbf{A}_1\right)^{-1}\mathbf{A}_1^T\mathbf{d}_1 \tag{24}$$

where

$$\mathbf{A}_1 = \begin{bmatrix} \mathbf{v}_3^T \\ {}^C\mathbf{n}_{r,1}^T \\ \cdots \\ {}^C\mathbf{n}_{r,\mathrm{N}_\pi}^T \end{bmatrix} \quad \mathbf{d}_1 = \begin{bmatrix} 0 \\ {}^C d_{r,1} - {}^C d_{c,1} \\ \cdots \\ {}^C d_{r,\mathrm{N}_\pi} - {}^C d_{c,\mathrm{N}_\pi} \end{bmatrix}. \tag{25}$$

The resultant transformation $\mathbf{R}$ and $\mathbf{t}_1$ can align the corresponding planes in Π_{mc} and Π_{mr}. Until now, the translation $\mathbf{t}_2 = \mu\mathbf{v}_3$ along the $\mathbf{v}_3$ direction has not been determined. In what follows, we calculate it to yield the complete 6-DoF camera pose estimate.

Denote the point sets in the PPS (obtained in Section III-A) of the current and reference scans by ${}^P\Omega_c = \{{}^P\mathbf{p}_{c,i}, i = 1,\ldots,N_c\}$ and ${}^P\Omega_r = \{{}^P\mathbf{p}_{r,i}, i = 1,\ldots,N_r\}$, respectively, as shown in Fig. 4(b). Let ${}^P\Omega_t = \{{}^P\mathbf{p}_{t,i}, i = 1,\ldots,N_c\}$ be the point set transformed from ${}^P\Omega_c$ by $\mathbf{R}$ and $\mathbf{t}_1$. For any ${}^P\mathbf{p}_{t,i} \in {}^P\Omega_t$, its local plane in the Cartesian space is ${}^C\pi_{t,i} = [{}^C\mathbf{n}_{t,i}, {}^C d_{t,i}]^T$. Assuming that $\mathbf{t}_2$ is applied to ${}^P\mathbf{p}_{t,i}$ to generate ${}^P\mathbf{p}_{t',i}$, whose local plane in the Cartesian space is ${}^C\pi_{t',i} = [{}^C\mathbf{n}_{t',i}, {}^C d_{t',i}]^T$. Then we have

$${}^C\mathbf{n}_{t',i} = {}^C\mathbf{n}_{t,i} \tag{26}$$

$${}^C d_{t',i} = {}^C d_{t,i} + {}^C\mathbf{n}_{t,i}^T \mathbf{t}_2 = {}^C d_{t,i} + \mu \cdot {}^C\mathbf{n}_{t,i}^T \mathbf{v}_3. \tag{27}$$

From (26) and (27), if ${}^C\mathbf{n}_{t,i}$ is perpendicular to $\mathbf{v}_3$, $\mathbf{t}_2$ has no effect on both ${}^C\mathbf{n}_{t',i}$ and ${}^C d_{t',i}$. Therefore, we set a threshold ε_{sub} to exclude the points in ${}^P\Omega_t$ that make little contribution to the computation of $\mathbf{t}_2$, yielding a subset of ${}^P\Omega_t$ denoted by ${}^P\Omega_{t,\text{sub}} = \{{}^P\mathbf{p}_{t,i}, i =$

$1, 2, \ldots N_{t,\mathrm{sub}} | {}^{P}\mathbf{p}_{t,i} \in {}^{P}\Omega_t, |{}^{C}\mathbf{n}_{t,i}^{T}\mathbf{v}_3| > \varepsilon_{sub}\}$. In the experiments, $\varepsilon_{\mathrm{sub}} = 0.5$ produces satisfactory results.

To estimate the translation $\mathbf{t}_2$, a STING-SM method is proposed. The bottom level cells $\{o_{Lk}, k = 1, 2, \ldots, (N_s)^{L-1}\}$ in the STING structure (built in Section III-B) of the reference scan are used to fit a local Gaussian distribution of the points ${}^{P}\Omega_r$ in the PPS. The translation $\mathbf{t}_2 = \mu\mathbf{v}_3$ is applied to the point set ${}^{P}\Omega_{t,\mathrm{sub}}$ in the PPS using (26) and (27), yielding a point set ${}^{P}\Omega_{t',\mathrm{sub}} = \{{}^{P}\mathbf{p}_{t',i}, i = 1, \ldots, N_{t,\mathrm{sub}}\}$. It is assumed that for each $i \in \{1, \ldots, N_{t,\mathrm{sub}}\}$, ${}^{P}\mathbf{p}_{t',i}$ falls into the k_ith cell o_{Lk_i}. According to the local Gaussian distribution of the cell o_{Lk_i}, the probability of measuring a point in the location of ${}^{P}\mathbf{p}_{t',i}$ can be computed by

$$\begin{aligned} & p\left({}^{P}\mathbf{p}_{t,i}, \mu\right) \\ & = \frac{1}{\rho_1} \exp \frac{-\left({}^{P}\mathbf{p}_{t',i} - {}^{P}\mathbf{m}_{Lk_i}\right)^{T} {}^{P}\mathbf{S}_{Lk_i}{}^{-1} \left({}^{P}\mathbf{p}_{t',i} - {}^{P}\mathbf{m}_{Lk_i}\right)}{2} \end{aligned} \tag{28}$$

where ρ_1 is a normalization factor. Then, Newton's algorithm is used to solve μ by minimizing

$$f(\mu) = -\sum_{i=1}^{N_{t,\mathrm{sub}}} p\left({}^{P}\mathbf{p}_{t,i}, \mu\right). \tag{29}$$

Therefore, the translation $\mathbf{t}_2 = \mu\mathbf{v}_3$ can be determined. The 6-DoF transformation between the two successive scans is thus obtained only based on the matched plane features and the STING structure, without requirement for any other feature extraction process. Equations (22) and (24) provide a close-form solution to $\mathbf{R}$ and $\mathbf{t}_1$. As for the calculation of $\mathbf{t}_2$, only one variable μ, rather than a 6-DoF pose, needs to be computed with the nonlinear optimization. As a result, the computation process is largely simplified.

C. 3-DoF Constraint Case

If $\mathbf{H}$ is singular and $\lambda_1 > \lambda_2 = \lambda_3 = 0$, then $\{{}^{C}\mathbf{n}_{c,i}, i = 1, 2, \ldots, N_\pi\}$ $(\{{}^{C}\mathbf{n}_{r,i}, i = 1, 2, \ldots, N_\pi\})$ are along the $\mathbf{u}_1(\mathbf{v}_1)$ direction. In Fig. 4(c), only one plane correspondence is established between the current and reference scans. In this case, the rotation about the $\mathbf{v}_1$ and the translation along the $\mathbf{v}_2$ and $\mathbf{v}_3$ cannot be constrained.

Because the rotation about $\mathbf{v}_1$ cannot be constrained, there exist infinite rotation matrices that can minimize (17). To obtain a unique solution, we redefine a new objective function

$$J_{2,R}(\mathbf{R}) = \sum_{i=1}^{N_\pi} \left\|{}^{C}\mathbf{n}_{r,i} - \mathbf{R} \cdot {}^{C}\mathbf{n}_{c,i}\right\|^2 + \|\mathbf{v}_2 - \mathbf{R}\mathbf{u}_2\|^2. \tag{30}$$

Likewise, define the matrix $\mathbf{H}_1 = \sum_{i=1}^{N_\pi} {}^{C}\mathbf{n}_{c,i}{}^{C}\mathbf{n}_{r,\mathrm{i}}^{T} + \mathbf{u}_2\mathbf{v}_2^{T} = \mathbf{H} + \mathbf{u}_2\mathbf{v}_2^{T}$ and find its SVD $\mathbf{H}_1 = \mathbf{U}_1\mathbf{\Lambda}_1\mathbf{V}_1^{T}$. Then, the rotation matrix $\mathbf{R}_1$ that minimizes (30) can be calculated by

$$\mathbf{R}_1 = \begin{cases} \mathbf{U}_1\mathbf{V}_1^{T} & \text{if } \det(\mathbf{U}_1\mathbf{V}_1^{T}) = 1 \\ \mathbf{U}_1'\mathbf{V}_1^{T} & \text{if } \det(\mathbf{U}_1\mathbf{V}_1^{T}) = -1 \end{cases} \tag{31}$$

where $\mathbf{U}_1'$ is defined in the same way as $\mathbf{U}'$ in the 5-DoF case.

To calculate the translation vector $\mathbf{t}_1$, we define

$$J_{2,t}(\mathbf{t}) = \sum_{i=1}^{N_\pi} \left({}^{C}d_{r,i} - \left({}^{C}d_{c,i} + {}^{C}\mathbf{n}_{r,i}^{T}\mathbf{t}\right)\right)^2 + \left(\mathbf{v}_2^{T}\mathbf{t}\right)^2 + \left(\mathbf{v}_3^{T}\mathbf{t}\right)^2. \tag{32}$$

And the least-square method is utilized to obtain $\mathbf{t}_1$ by minimizing $J_{2,t}(\mathbf{t})$

$$\mathbf{t}_1 = \left(\mathbf{A}_2^{T}\mathbf{A}_2\right)^{-1}\mathbf{A}_2^{T}\mathbf{d}_2 \tag{33}$$

where

$$\mathbf{A}_2 = \begin{bmatrix} \mathbf{v}_2^{T} \\ \mathbf{v}_3^{T} \\ {}^{C}\mathbf{n}_{r,1}^{T} \\ \cdots \\ {}^{C}\mathbf{n}_{r,\mathrm{N}_\pi}^{T} \end{bmatrix} \mathbf{d}_2 = \begin{bmatrix} 0 \\ 0 \\ {}^{C}d_{r,1} - {}^{C}d_{c,1} \\ \cdots \\ {}^{C}d_{r,\mathrm{N}_\pi} - {}^{C}d_{c,\mathrm{N}_\pi} \end{bmatrix}. \tag{34}$$

Furthermore, an additional transformation $(\mathbf{R}_2, \mathbf{t}_2)$ needs to be calculated to obtain the complete 6-DoF transformation, as shown in Fig. 4(c). We denote $\mathbf{R}_2$ and $\mathbf{t}_2$ by a compact form $\mathbf{w} = [\phi, x, y]^{T}$, where $\phi \in [0, 2\pi)$ is the rotation angle around $\mathbf{v}_1$, $x, y \in R$ are the components of $\mathbf{t}_2$ along $\mathbf{v}_2$ and $\mathbf{v}_3$, respectively. Then, $\mathbf{R}_2$ and $\mathbf{t}_2$ can be represented by

$$\mathbf{R}_2 = \begin{bmatrix} v_{1,x}^2 \cdot k\phi + c\phi & v_{1,x}v_{1,y} \cdot k\phi - v_{1,z} \cdot s\phi & v_{1,x}v_{1,z} \cdot k\phi + v_{1,y} \cdot s\phi \\ v_{1,x}v_{1,y} \cdot k\phi + v_{1,z} \cdot s\phi & v_{1,y}^2 \cdot k\phi + c\phi & v_{1,y}v_{1,z} \cdot k\phi - v_{1,x} \cdot s\phi \\ v_{1,x}v_{1,z} \cdot k\phi - v_{1,y} \cdot s\phi & v_{1,y}v_{1,z} \cdot k\phi + v_{1,x} \cdot s\phi & v_{1,z}^2 \cdot k\phi + c\phi \end{bmatrix} \tag{35}$$

$$\mathbf{t}_2 = x\mathbf{v}_2 + y\mathbf{v}_3 \tag{36}$$

where $s\phi = \sin\phi$, $c\phi = \cos\phi$, $k\phi = 1 - \cos\phi$ and $\mathbf{v}_1 = [v_{1,x}, v_{1,y}, v_{1,z}]^{T}$.

In this case, (26) and (27) become

$${}^{C}\mathbf{n}_{t',i} = \mathbf{R}_2 \cdot {}^{C}\mathbf{n}_{t,i} \tag{37}$$

$${}^{C}d_{t',i} = {}^{C}d_{t,i} + \left(\mathbf{R}_2 \cdot {}^{C}\mathbf{n}_{t,i}\right)^{T}\mathbf{t}_2. \tag{38}$$

Assuming that the variation of $\mathbf{w}$ is small, $\mathbf{R}_2$ can be approximated by

$$\mathbf{R}_2 = \begin{bmatrix} 1 & -v_{1,z}\phi & v_{1,y}\phi \\ v_{1,z}\phi & 1 & -v_{1,x}\phi \\ -v_{1,y}\phi & v_{1,x}\phi & 1 \end{bmatrix}. \tag{39}$$

The Jacobian matrix of ${}^C\pi_{t',i}$ with respect to $\mathbf{w}$ is

$$\frac{\partial^C \pi_{t',i}}{\partial \mathbf{w}} = \begin{bmatrix} \frac{\partial^C \mathbf{n}_{t',i}}{\partial \mathbf{w}} \\ \frac{\partial^C d_{t',i}}{\partial \mathbf{w}} \end{bmatrix}$$

$$= \begin{bmatrix} {}^C n_z v_{1,y} - {}^C n_y v_{1,z} & 0 & 0 \\ {}^C n_x v_{1,z} - {}^C n_z v_{1,x} & 0 & 0 \\ {}^C n_y v_{1,x} - {}^C n_x v_{1,y} & 0 & 0 \\ 0 & {}^C \mathbf{n}_{t,i}^T \mathbf{v}_2 & {}^C \mathbf{n}_{t,i}^T \mathbf{v}_3 \end{bmatrix} \quad (40)$$

where ${}^C\mathbf{n}_{t,i} = [{}^C n_x, {}^C n_y, {}^C n_z]^T$. $\frac{\partial^C \pi_{t',i}}{\partial \mathbf{w}}$ represents the variation of the plane parameters ${}^C\pi_{t,i}$ caused by a small transformation $\Delta\mathbf{w}$. Namely,

$$\begin{bmatrix} {}^C\mathbf{n}_{t',i} - {}^C\mathbf{n}_{t,i} \\ {}^C d_{t',i} - {}^C d_{t,i} \end{bmatrix} = \frac{\partial^C \pi_{t',i}}{\partial \mathbf{w}} \Delta\mathbf{w}. \quad (41)$$

Squaring (41) results in

$$D\left({}^C\mathbf{n}_{t,i}\right) = \Delta\mathbf{w}^T \left(\frac{\partial^C \pi_{t',i}}{\partial \mathbf{w}}\right)^T \left(\frac{\partial^C \pi_{t',i}}{\partial \mathbf{w}}\right) \Delta\mathbf{w}$$
$$= \Delta\mathbf{w}^T \Psi \Delta\mathbf{w} \quad (42)$$

where $\Psi = (\frac{\partial^C \pi_{t',i}}{\partial \mathbf{w}})^T (\frac{\partial^C \pi_{t',i}}{\partial \mathbf{w}})$ is a symmetric and positive semidefinite matrix. Its eigenvalue decomposition is computed by $\Psi = \mathbf{Q}\mathbf{\Lambda}_\psi \mathbf{Q}^T$, where $\mathbf{\Lambda}_\psi = \mathrm{diag}\{\lambda_{\psi,1}, \lambda_{\psi,2}, \lambda_{\psi,3}\}$ and $\mathbf{Q} = [\mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3]$ denote the eigenvalues and the corresponding eigenvectors, respectively,

$$\lambda_{\psi,1} = \left\|\mathbf{v}_1 \times {}^C\mathbf{n}_{t,i}\right\|^2, \mathbf{q}_1 = [1,0,0]^T \quad (43)$$

$$\lambda_{\psi,2} = \left({}^C\mathbf{n}_{t,i}^T\mathbf{v}_2\right)^2 + \left({}^C\mathbf{n}_{t,i}^T\mathbf{v}_3\right)^2,$$

$$\mathbf{q}_2 = \frac{1}{\rho_2}\left[0, {}^C\mathbf{n}_{t,i}^T\mathbf{v}_2, {}^C\mathbf{n}_{t,i}^T\mathbf{v}_3\right]^T \quad (44)$$

$$\lambda_{\psi,3} = 0, \mathbf{q}_3 = \frac{1}{\rho_3}\left[0, {}^C\mathbf{n}_{t,i}^T\mathbf{v}_3, -{}^C\mathbf{n}_{t,i}^T\mathbf{v}_2\right]^T \quad (45)$$

where ρ_2 and ρ_3 are the normalization factors. The eigenvalue $\lambda_{\psi,j}, j \in \{1,2,3\}$ is proportional to the rate of the change of ${}^C\pi_{t,i}$, which is caused by the transformation specified by $\mathbf{q}_j$. Note that, a singularity of Ψ exists because $\lambda_{\psi,3}$ is equal to zero. As a result, transformation specified by $\mathbf{q}_3$ cannot cause a change of ${}^C\pi_{t,i}$. Thus, to prevent $\lambda_{\psi,1}$ and $\lambda_{\psi,2}$ from being zero, the subset ${}^P\Omega_{t,sub} = \{{}^P\mathbf{p}_{t,i}, i = 1,2,\ldots N_{t,sub} | {}^P\mathbf{p}_{t,i} \in {}^P\Omega_t, |{}^C\mathbf{n}_{t,i}^T\mathbf{v}_2| > \varepsilon_{sub}, |{}^C\mathbf{n}_{t,i}^T\mathbf{v}_3| > \varepsilon_{sub}\}$ is chosen to exclude the points in ${}^P\Omega_t$ which lead to ${}^C\mathbf{n}_{t,i}^T\mathbf{v}_2 = 0$ and ${}^C\mathbf{n}_{t,i}^T\mathbf{v}_3 = 0$.

Similar to the case of 5-DoF constraint, $f(\mathbf{w})$ can be defined as

$$f(\mathbf{w}) = -\sum_{i=1}^{N_{t,\mathrm{sub}}} p\left({}^P\mathbf{p}_{t,i}, \mathbf{w}\right) \quad (46)$$

where $p({}^P\mathbf{p}_{t,i}, \mathbf{w})$ is defined in the same way as (28), where ${}^P\Omega_{t',\mathrm{sub}}$ is obtained via (37) and (38).

TABLE I
RECALL RATE AND PRECISION RATE OF THE TWO METHODS ON FIVE IMAGE SEQUENCES

		STING-PE	RANSAC-PE
Fr1/xyz	Recall rate	98.6%	93.5%
	Precision rate	99.0%	95.1%
Fr2/desk	Recall rate	97.3%	92.7%
	Precision rate	99.3%	90.2%
Fr1/room	Recall rate	98.5%	93.5%
	Precision rate	97.0%	87.3%
Fr3/cabinet	Recall rate	100%	100%
	Precision rate	100%	100%
Fr2/pioneer360	Recall rate	89.4%	76.2%
	Precision rate	91.2%	82.2%

V. EXPERIMENTAL EVALUATION

In this section, the STING-PE method in the PPS is first compared with the most widely used RANSAC algorithm performed in the Cartesian space. Then, the proposed PF-RGBD-CPE method is run as the RGB-D visual odometry (VO) on the Freiburg RGB-D benchmark [27] and is compared with the plane-point method [22] and the RGBD-ICP method [28]. We also apply g2o [29] to the PF-RGBD-CPE method as a backend optimizer to perform the map correction, which yields a complete SLAM system. It is then compared with the RGBD-ICP + sparse bundle adjustment (SBA) SLAM system [28] and ElasticFusion [17] on the same RGB-D benchmark. Finally, a real-world experiment using a Pioneer 3-DX mobile robot equipped with a Microsoft Kinect 1.0 is performed in a laboratory environment, using the proposed PF-RGBD-CPE as the VO.

A. Plane Extraction Experiment

To test the performance of the STING-PE, we compared it with the RANSAC-based plane extraction (RANSAC-PE). In our experiment, the distance threshold of the RANSAC is set to 0.01 m, and the inlier number threshold is set to 500. Note that these thresholds are chosen as the best ones among many trials.

For the five image sequences chosen from the Freiburg dataset, we manually label the planes in each sequence. The two methods are employed on the five sequences. The recall rate and precision rate are used to evaluate the performance of the two methods. The recall rate is the fraction of the labeled planes that have been extracted over all the labeled planes. Precision is the fraction of extracted planes that are labeled over all the extracted planes. The recall and precision rates are calculated on each sequence and are shown in Table I.Note that the STING-PE yields higher recall and precision rates than the RANSAC-PE on the sequences except for Fr3/cabinet. On the Fr3/cabinet sequence, the recall and precision rates of both methods are 100% because the scene merely consists of several planar surfaces.

Table II shows the average computation time of the two methods for each frame of the five image sequences. The test platform is a PC with an Intel Pentium G2020 CPU at 2.9 GHz and 4 GB RAM. It can be seen that the STING-PE method is generally less time-consuming than the RANSAC-PE method.

TABLE II
AVERAGE COMPUTATION TIME OF THE TWO METHODS FOR EACH FRAME OF FIVE IMAGE SEQUENCES

	STING-PE	RANSAC-PE
Fr1/xyz	85.1 ms	116.0 ms
Fr2/desk	171.8 ms	286.5 ms
Fr1/room	128.2 ms	168.3 ms
Fr3/cabinet	113.9 ms	163.1 ms
Fr2/pioneer360	249.7 ms	325.0 ms

TABLE III
COMPARISON OF THREE CAMERA POSE ESTIMATION METHODS

		PF-RGBD-CPE	Plane-point	RGB-D ICP
Fr1/xyz	ATE	0.0381 m	0.0513 m	0.0539 m
	RPE	0.0224 m, 0.77°	0.0259 m, 0.96°	0.0233 m, 1.11°
Fr2/desk	ATE	0.0987 m	0.127 m	0.305 m
	RPE	0.0484 m, 1.57°	0.0507 m, 1.70°	0.0515 m, 1.81°
Fr1/room	ATE	0.284 m	0.341 m	1.53 m
	RPE	0.0418 m, 1.81°	0.0668 m, 2.44°	0.365 m, 21.4°
Fr3/cabinet	ATE	0.0709 m	0.760 m	1.33 m
	RPE	0.0113 m, 1.02°	0.134 m, 12.1°	0.599 m, 19.6°
Fr2/pioneer360	ATE	0.5102 m	Failed	Failed
	RPE	0.1357 m, 1.55°		

TABLE IV
OCCURRENCE PERCENTAGE OF THREE KINDS OF CONSTRAINTS IN THE FIVE IMAGE SEQUENCES

	Fr1/xyz	Fr2/desk	Fr1/room	Fr3/cabinet	Fr2/pioneer360
6-DoF	44.6%	53.8%	44.7%	48.5%	2.1%
5-DoF	40.9%	35.1%	32.5%	47.4%	20.6%
3-DoF	14.5%	11.1%	22.8%	4.1%	77.3%

B. Comparison of Different Camera Pose Estimation Methods

The proposed PF-RGBD-CPE method estimates the incremental camera motion and VO can be calculated by the integration of the incremental motion. The accuracy of our method is compared with that of two other VO algorithms: the plane-point method [22] and the RGBD-ICP method [28]. The plane-point method used both planes and points as primitives in a RANSAC framework to determine correspondences and compute the sensor pose. In the RGBD-ICP method, both sparse FAST feature points and a dense point cloud were used to calculate the sensor pose via the ICP algorithm. Table III gives the experimental results of the three methods, where the root mean square errors (RMSEs) of both the absolute trajectory error (ATE) and the relative pose error (RPE) are adopted as the error metric. From Table III, the PF-RGBD-CPE method outperforms the other two in terms of both the ATE and RPE. The percentage of the 6-DoF, 5-DoF, and 3-DoF constraint cases in every sequence is shown in Table IV. It can be seen that, insufficient constraint cases are likely to occur in the indoor environments because the RGB-D cameras commonly have a narrow field of view and a quite limited depth range. The results demonstrate that STING-SM is performed as a necessary component of the PF-RGBD-CPE.

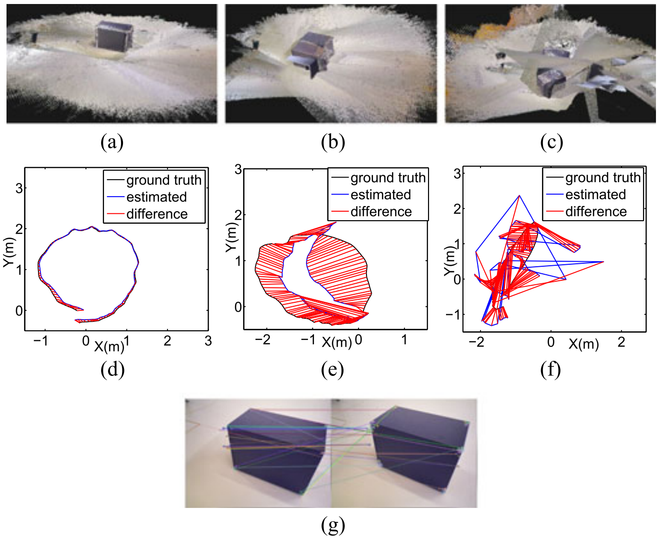
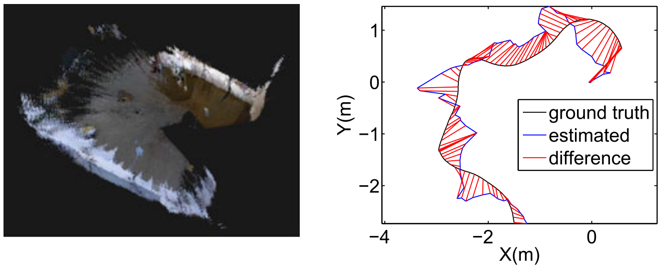


Fig. 5. Point cloud maps (a)–(c) and the visualizations of the ATE (d)–(f) on Fr3/cabinet for PF-RGBD-ICP method, plane-point method, and RGBD-ICP method, respectively. The trajectories estimated by either the plane-point method or the RGBD-ICP method have large offset with respect to the ground truth in (e) and (f), while the trajectory estimated by the PF-RGBD-CPE method is close to the ground truth in (d). (g) is the extracted and matched SURF point features from two successive frames.

In the following, we further show some detailed results on the image sequences Fr3/cabinet and Fr2/pioneer360.

For the sequence Fr3/cabinet, the handheld Asus Xtion moves around a cabinet with little texture and structure. Fig. 5 (a)–(f) shows the generated point cloud maps and the visualizations of the ATE for three methods. In Fig. 5(c), RGBD-ICP cannot restore the contours of the environment because few point features can be correctly matched between two successive frames, as shown in Fig. 5(g). Additionally, the offset of the trajectory estimated by the plane-point method from the ground truth is very large because this method heavily relies on the point features when the matched plane features cannot provide enough constraints. In contrast, our method performs well, as shown in Fig. 5(a) and (d). In this case, the plane features turn out to be a powerful alternative to the point features.

For the image sequence Fr2/pioneer360, the Kinect was mounted on the top of a Pioneer robot, which was controlled by a joystick to wander around an industrial hall. The area of the industrial hall is quite large compared with the office environment. As a result, the depth measurement is usually missing or noisy because of the quadratic growth of the depth uncertainty in RGB-D cameras [30]. Therefore, it is a very challenging scene for a SLAM system. From Table III, only the PF-RGBD-CPE method can build the environment map successfully. The point cloud map and the visualization of the ATE are shown in Fig. 6(a) and (b), respectively. Because the area of the hall is large, most of the SURF features detected in the RGB images frequently go out of the depth range of the Kinect. Thus, the association between these point features in the RGB image and



(a) (b)

Fig. 6. Point cloud map (a) and the visualization of the ATE (b) on Fr2/pioneer360 for the PF-RGBD-CPE method.

TABLE V
Mean and Standard Deviation of the Computation Time for Each Frame of Five Image Sequences

	PF-RGBD-CPE	Plane-point	RGB-D ICP
Fr1/xyz	198.7 ± 173.2 ms	366.1 ± 60.7 ms	446.3 ± 72.7 ms
Fr2/desk	259.3 ± 187.4 ms	423.8 ± 79.3 ms	506.4 ± 108.3 ms
Fr1/room	248.6 ± 175.5 ms	333.7 ± 51.1 ms	439.3 ± 95.4 ms
Fr3/cabinet	175.1 ± 76.2 ms	276.3 ± 49.2 ms	386.8 ± 51.4 ms
Fr2/pioneer360	401.4 ± 184.2 ms		

their counterparts in the 3-D point cloud cannot be established. As a result, both the plane-point and RGBD-ICP methods, which rely on sufficient 3-D point features, fail to build the map of the hall.

For the three methods, the mean and standard deviation of the computation time of each frame is shown in Table V. All the experiments are run on a PC with an Intel Pentium G2020 CPU at 2.9 GHz and 4 GB RAM. As can be seen from Table V that the average runtime of our method is less than that of the other two, though the standard deviation of our method is larger, because the 5-DoF and 3-DoF cases in our method are generally more time-consuming than the 6-DoF case due to the additional scan matching process.

C. Comparison of Different SLAM Methods

Using the PF-RGBD-CPE as the front-end and the g2o [29] as the backend pose graph optimizer, a complete SLAM system is constructed and compared with the state-of-the-art ElasticFusion [17] as well as the SLAM algorithm proposed in [28]. This experiment is conducted just to demonstrate that the PF-RGBD-CPE method can work well in an entire SLAM system with any pose graph optimization as the backend optimizer. ElasticFusion is a map-centric approach to dense SLAM. It maintains a fused surfel-based dense map of the environment and performs fusing and tracking simultaneously. The loop closure was achieved via the randomized fern encoding technique, and the map correction was fulfilled by nonrigid deformation. The SLAM in [28] was performed using RGBD-ICP as the frontend and the SBA [31] as the backend optimizer.

The ATE RMSEs of three SLAM systems are shown in Table VI, where the results of ElasticFusion have been published in [17]. Our method outperforms the other two except for on the Fr1/room sequence, on which ElasticFusion achieves the best result in term of the ATE RMSE. However, ElasticFusion fails to track the camera pose on the sequences Fr3/cabinet and Fr2/pioneer360. The photometric and geometric frame-to-model tracking of ElasticFusion is likely to fail for a textureless and simple structure, such as the Fr3/cabinet scene. For Fr2/pioneer360, the camera sometimes points to an area outside of the range of valid depths, which results in failures for dense tracking methods. For more details, see [17].

TABLE VI
Comparison of Three SLAM Systems

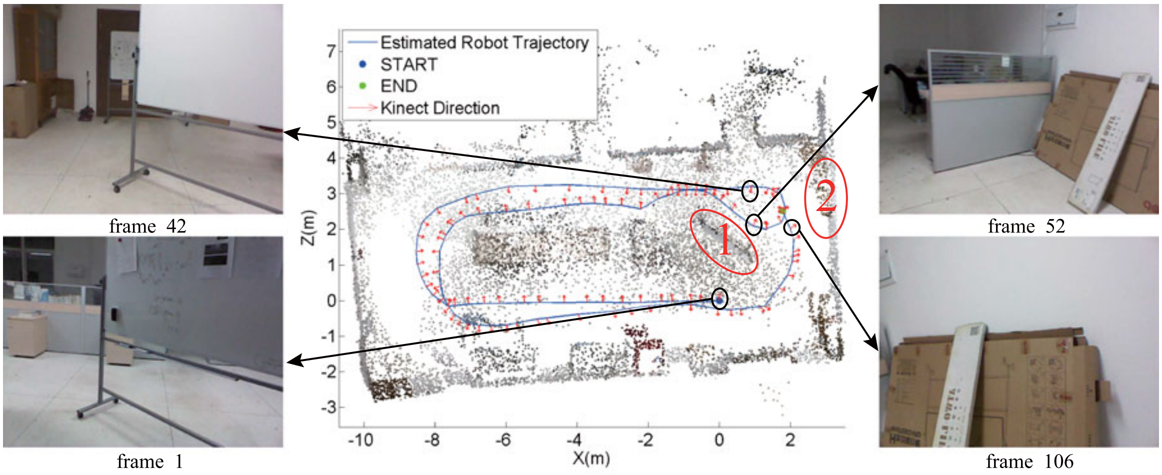
	PF-RGBD-CPE + g2o	ElasticFusion	RGBD-ICP + SBA
Fr1/xyz	0.011 m	0.011 m	0.014 m
Fr2/desk	0.053 m	0.071 m	0.113 m
Fr1/room	0.083 m	0.068 m	1.322 m
Fr3/cabinet	0.032 m	Failed	1.152 m
Fr2/pioneer360	0.209 m	Failed	Failed

D. Real-World Robot Experiment

In this experiment, the proposed method is run as an RGB-D VO without any backend optimization in a real world scene. The size of the laboratory is approximately 12.0 m × 5.2 m. The Kinect is mounted on the Pioneer 3-DX mobile robot 1.14 m above the ground, pointing to the right side of the robot. The PF-RGBD-CPE runs on an onboard computer (Intel Pentium Dual T2390 CPU at 1.86 GHz, 3G RAM). The length of the trajectory is approximately 46.7 m, covering two loops around the laboratory. During the first loop, the Kinect points to the tables in the middle of the room, and during the second loop it points to the surrounding walls.

The estimated trajectory of the mobile robot is shown in Fig. 7,where the built point cloud map is projected onto the X- and Z-axes of the global coordinate system (i.e., the camera coordinate system of the first frame), and the generated 3-D point cloud map is shown in Fig. 8. In frame 1, a white board is observed by the robot, which is marked with a red ellipse labeled "1" in Figs. 7 and Fig. 8(a)–(c). After traveling approximately 19.9 m around the room, the robot re-observes the same white board in frame 42. From Fig. 8, we can see that the observation of the same white board in frame 1 almost completely overlaps with the one in frame 42 in the point cloud map. Likewise, the robot observes a white carton in frame 52 and again in frame 106 after traveling approximately 23.8 m. The white carton is marked in Figs. 7 and 8(a)–(c) with a red ellipse labeled "2". Note from Fig. 8(a)–(c) that only a small offset occurs after the incremental mapping process of a distance of 24 m using the PF-RGBD-CPE method as a frame-to-frame registration technique.

In summary, the PF-RGBD-CPE method yields good results in both incremental mapping and complete SLAM. Especially when the environment is lacking in texture or the RGB-D sensor is pointed to an area outside of the valid depth values, the PE-RGBD-CPE method can provide an alternative to the existing point-feature-based SLAM technique.



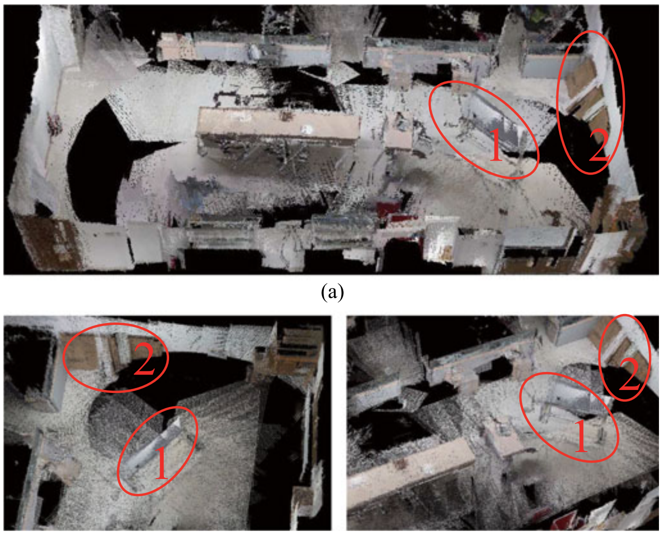
frame 42

frame 1

frame 52

frame 106

Fig. 7. Estimated trajectory of the mobile robot and the point cloud map projected onto the X- and Z-axes of the global coordinate system. The white board seen at frame 1 and frame 42 is marked by the red ellipse labeled “1”, and the white carton seen at frame 52 and frame 106 is marked by the red ellipse labeled “2” in Fig. 8.



(a)

(b)

(c)

Fig. 8. Three-dimensional point cloud map generated by the PF-RGBD-CPE method as a frame-to-frame registration technique. The labels “1” and “2” indicate the white board and the white carton, respectively, which are circled with the same labels in Fig. 7. (a) Panoramic view of the generated map. (b) and (c) The zoom-in local views of the map.

VI. Conclusion

In this paper, an RGB-D camera pose estimation method aiming at 3-D indoor environment mapping has been proposed. The STING-PE and PAG-PM have been presented. Then, the matched plane features have been used to calculate the RGB-D camera pose. When the plane matches fail to provide sufficient constraints for the camera pose estimation, a STING-SM method has been developed to offer extra constraints and achieve full 6-DoF camera pose estimation.

The proposed method is applicable not only to 3-D mapping using a hand-held RGB-D sensor, but also to SLAM with a mobile robot or an air vehicle. Experimental results have demonstrated that the proposed method compares favorably with other VO and SLAM techniques. It performs well even for texture-less indoor scenes.

References

[1] S. Park and K. S. Roh, “Coarse-to-fine localization for a mobile robot based on place learning with a 2-D range scan,” *IEEE Trans. Robot.*, vol. 32, no. 3, pp. 528–544, Jun. 2016.

[2] K. Wang, Y. Liu, and L. Li, “A simple and parallel algorithm for real-time robot localization by fusing monocular vision and odometry/AHRS sensors,” *IEEE/ASME Trans. Mechatronics*, vol. 19, no. 4, pp. 1447–1457, Aug. 2014.

[3] J. Yoo and J. Kim, “Gaze control-based navigation architecture with a situation-specific preference approach for humanoid robots,” *IEEE/ASME Trans. Mechatronics*, vol. 20, no. 5, pp. 2425–2436, Oct. 2015.

[4] F. Aghili and C. Su, “Robust relative navigation by integration of ICP and adaptive kalman filter using laser scanner and IMU,” *IEEE/ASME Trans. Mechatronics*, vol. 21, no. 4, pp. 2015–2026, Aug. 2016.

[5] H. Yu, K. Meier, M. Argyle, and R. W. Beard, “Cooperative path planning for target tracking in urban environments using unmanned air and ground vehicles,” *IEEE/ASME Trans. Mechatronics*, vol. 20, no. 2, pp. 541–552, Apr. 2015.

[6] R. Valencia, M. Morta, J. Andrade-Cetto, and J. M. Porta, “Planning reliable paths with pose SLAM,” *IEEE Trans. Robot.*, vol. 29, no. 4, pp. 1050–1059, Aug. 2013.

[7] M. Stommel, M. Beetz, and W. Xu, “Model-free detection, encoding, retrieval, and visualization of human poses from Kinect data,” *IEEE/ASME Trans. Mechatronics*, vol. 20, no. 2, pp. 865–875, Apr. 2015.

[8] S. Weiss *et al.*, “Monocular vision for long-term micro aerial vehicle state estimation: A compendium,” *J. Field Robot.*, vol. 30, no. 5, pp. 803–831, 2013.

[9] H. Kretzschmar, C. Stachniss, and G. Grisetti, “Efficient information theoretic graph pruning for graph-based SLAM with laser range finders,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, San Francisco, CA, USA, 2011, pp. 865–871.

[10] K. Pathak, A. Birk, N. Vaskevicius, and J. Popinga, “Fast registration based on noisy planes with unknown correspondences for 3-D mapping,” *IEEE Trans. Robot.*, vol. 26, no. 3, pp. 424–441, Jun. 2010.

[11] P. Osteen, J. Owens, and C. Kessens, “Online egomotion estimation of RGB-D sensors using spherical harmonics,” in *Proc. IEEE Int. Conf. Robot. Autom.*, Saint Paul, MN, USA, 2012, pp. 1679–1684.

[12] 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, 2013, pp. 2305–2310.

[13] C. Kerl, J. Sturm, and D. Cremers, “Robust odometry estimation for RGB-D cameras,” in *Proc. IEEE Int. Conf. Robot. Autom.*, Karlsruhe, Germany, 2013, pp. 3748–3754.

[14] F. Steinbrucker, J. Sturm, and D. Cremers, "Real-time visual odometry from dense RGB-D images," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, Barcelona, Spain, 2011, pp. 719–722.

[15] S. Klose, P. Heise, and A. Knoll, "Efficient compositional approaches for real-time robust direct visual odometry from RGB-D data," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, Tokyo, Japan, 2013, pp. 1100–1106.

[16] D. Gutierrez-Gomez, W. Mayol-Cuevas, and J. Guerrero, "Inverse depth for accurate photometric and geometric error minimization in RGB-D dense visual odometry," in *Proc. IEEE Int. Conf. Robot. Autom.*, Seattle, WA, USA, 2015, pp. 83–89.

[17] 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, 2016.

[18] 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.

[19] K. Yousif, A. Bab-hadiashar, and R. Hoseinnezhad, "Real-time RGB-D registration and mapping in texture-less environments using ranked order statistics," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, Chicago, IL, USA, 2014, pp. 2654–2660.

[20] I. Dryanovski, C. Jaramillo, and J. Xiao, "Incremental registration of RGB-D images," in *Proc. IEEE Int. Conf. Robot. Autom.*, Saint Paul, MN, USA, 2012, pp. 1685–1690.

[21] T. Lee, S. Lim, S. Lee, S. An, and S. Oh, "Indoor mapping using planes extracted from noisy RGB-D sensors," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, Vilamoura, Portugal, 2012, pp. 1727–1733.

[22] Y. Taguchi, Y. D. Jian, S. Ramalingam, and C. Feng, "Point-plane SLAM for hand-held 3-D sensors," in *Proc. IEEE Int. Conf. Robot. Autom.*, Karlsruhe, Germany, 2013, pp. 5182–5189.

[23] S. Holzer, R. B. Rusu, M. Dixon, S. Gedikli, and N. Navab, "Adaptive neighborhood selection for real-time surface normal estimation from organized point cloud data using integral images," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, Vilamoura, Portugal, 2012, pp. 2684–2689.

[24] W. Wang, J. Yang, and R. Muntz, "STING: A statistical information grid approach to spatial data mining," in *Proc. 23rd Conf. Very Large Data Bases*, San Francisco, CA, USA, 1997, pp. 186–195.

[25] G. Jeh and J. Widom, "Simrank: A measure of structural-context similarity," in *Proc. Eighth ACM SIGKDD Int. Conf. Knowledge Discovery Data Mining*, New York, NY, USA, 2002, pp. 538–543.

[26] K. S. Arun, T. S. Huang, and S. D. Blostein, "Least-squares fitting of two 3-D point sets," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-9, no. 5, pp. 698–700, Sep. 1987.

[27] 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. Robot. Syst.*, Vilamoura, Portugal, 2012, pp. 573–580.

[28] P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox, "RGB-D mapping: Using kinect-style depth cameras for dense 3-D modeling of indoor environments," *Int. J. Robot. Res.*, vol. 31, no. 5, pp. 647–663, 2012.

[29] R. Kmmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "G2o: A general framework for graph optimization," in *Proc. IEEE Int. Conf. Robot. Autom.*, Shanghai, China, 2011, pp. 3607–3613.

[30] 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, 2012.

[31] M. Lourakis and A. Argyros, "SBA: A software package for generic sparse bundle adjustment," *ACM Trans. Math. Softw.*, vol. 36, no. 1, pp. 1–30, 2009.

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 working toward the Ph.D. degree in control theory and control engineering.

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

Jing Yuan (M'12) 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 an Associate Professor. His current research interests include robotic control, target tracking, and simultaneous localization and mapping.

Xuebo Zhang (M'12) 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 an Associate Professor. His current research interests include mobile robotics, motion planning, and visual servoing.

Fengchi Sun received the B.Sc. and M.Sc. degrees in automation from the Shandong University of Science and Technology, Qingdao, China, in 1994 and 1998, respectively, and the Ph.D. degree in control theory and control engineering from Nankai University, Tianjin, China, in 2003.

He is currently an Associate Professor with the College of Software, Nankai University. His current research interests include autonomous mobile robots and embedded systems.