Plane-Aided Visual-Inertial Odometry for Pose Estimation of a 3D Camera based Indoor Blind Navigation

He Zhang
hxzhang1@ualr.edu
Cang Ye
cxye@ualr.edu

System Engineering Department
University of Arkansas at Little Rock, Arkansas, USA

Abstract

The classic visual-inertial odometry (VIO) method estimates a moving camera's 6-DOF pose relative to its starting point by fusing the camera's ego-motion measured by a visual odometry (VO) and the motion measured by an inertial measurement unit (IMU). The VIO attempts to updates the estimates of the IMU's biases at each step by using the VO's output so as to improve the accuracy of IMU measurement. This approach works only if an accurate VO output can be identified and used. However, there is no reliable method that can be used to evaluate the accuracy of the VO.

In this paper, a new VIO method is introduced for pose estimation of a robotic navigation aid (RNA) that uses a 3D time-of-flight camera for perception. The method, called plane-aided visual-inertial odometry (PAVIO), extracts planes from the 3D point cloud of the current camera view and track them onto the next camera view by using the IMU's measurement. The tracking result is used to accept the VO output only if it is accurate. The accepted VO outputs, the information of the extracted planes, and the IMU's measurements over time are used to create a factor graph. By optimizing the graph, the method improves the estimation accuracy of the IMU bias and reduces the camera's pose error. Experimental results with the RNA validate the effectiveness of the proposed method.

1 Introduction

Visual impairment reduces a person's independent mobility and severely deteriorates the quality of life. According to the World Health Organization, there are ~285 million people with visual impairment, of which 39 million are blind. The visually impaired community is growing due to the aging population. Therefore, there is a dire need to develop new mobility tools that may help visual impaired move around independently. A number of Robotic Navigation Aids (RNAs) [1], [2], [3], [4], [5], [6], [7], [8], have been introduced to guide the blind in indoor environments. Among these RANs, vision based systems are becoming more and more popular because the cameras, including monocular camera [2], stereo-cameras [3], [4], RGB-D cameras [5], [6] or 3D time-of-flight (TOF) cameras [7], [8], used in these RNAs can provide the needed information of the environments for navigation, including 6-DOF device pose (position and orientation) estimation and 3D object/obstacle detection. The pose information of an RNA can be used to build a 3D map for an unknown environment, locate the blind traveller in the environment, and guide the traveller to the destination.

The problem of camera pose estimation is also known as visual simultaneous localization and mapping (SLAM). The state-of-the-art visual SLAM algorithms [9], [10], [11], [12] have demonstrated their effectiveness in earlier research. However, they are effective only in a feature-rich environment. Their performances may be compromised when the operating environment is feature-sparse. To address this problem, two approaches have been taken in the literature.

The first is to use the geometric features (e.g., planes, lines) of the operating environment to limit the pose estimation error as they are ubiquitous in a man-made environment. Plane features have been used in EKF-SLAM [13] and pose-graph SLAM [14], [15] to reduce pose error accumulated by visual odometry (VO). In [7], [16], floor plane and wall lines were extracted from the point cloud data of a 3D TOF camera and used to reduce pose error of an RNA for wayfinding in an unknown indoor environment. However, geometric features are often insufficient to provide a full correction to the 6-DOF pose. For instance, at least three intersecting planes are needed to reduce the 6-DOF pose estimation error. However, only one or two planes are observable most of the time when using the RNA [7], [16] in the indoor environments.

The second is to use an additional sensor, usually an inertial measurement unit (IMU) or a gyro. In the robotics community, the combination of a camera and an IMU has become standard equipment for SLAM. The SLAM method based on a Camera-IMU suite is termed visual inertial odometry (VIO). The state-of-the-art VIO [17], [18], [19] uses VO to periodically update the estimates of the IMU's biases and integrate IMU's measurements for pose estimation when VO malfunctions. A more efficient way to incorporate IMU data is to fuse IMU's measurement with visual observation by incremental smoothing. However, VO may fail or produce large error in a feature-sparse environment. While VO failure may be detected, it is difficult to detect the latter with a monocular camera based VIO. The estimates of the IMU biases may become inaccurate in the former case and wrong in the latter case, resulting in a large pose estimation error. To overcome this problem, we propose a new method, called Plane-Aided VIO (PAVIO), for pose estimation of an RNA that uses a 3D TOF camera and an IMU for wayfinding. The method extracts plane features from the camera's point cloud data and tracks these features over the camera's data frames to associate plane features between the data frames. It then uses the plane correspondence information to reduce the pose estimation error by using a factor graph that integrates the pose changes estimated by VO and IMU and the information of the associated planes.

2 Navigation system and notation

As depicted in Fig. 1, the RNA uses a TOF camera (SwissRanger SR4000) for 3D perception and an IMU (VN 100 of VectorNav Technologies, LLC) for motion measurement. The SR4000 has a resolution of 176×144 pixels and a field-of-view of 43.6°×34.6°. It produces imaging data, each of which consists of an intensity image and a depth image, at about 25 fps. The IMU measures the angular velocity and acceleration at 200 Hz. In this paper, a new VIO method is developed to estimate the IMU's Pose Change (PC) by incorporating the IMU's measurement, PC estimated by the SR4000-based Visual Odometry (VO) [20], and geometry features (i.e., planes) extracted from the camera's point cloud data. The VO estimates the PC between two keyframes. The first data frame of the camera is taken as the first keyframe. A subsequent keyframe is defined as one with a substantial translation (>0.1 meter) or rotation (>2°) from the previous keyframe.

Keyframe i contains intensity image i (called image i for simplicity) and depth image i. There are multiple IMU measurements between two keyframes. These inertial measurements are integrated to produce another Pose Change Estimate (PCE). The two PCEs together with the plane features are used to create a factor graph [21] for estimating the IMU's pose in the world coordinate system. The use of the plane features helps to reduce pose estimation error. The proposed method is detailed in sections 3 and 4.

The coordinate systems of the IMU and camera, $X_sY_sZ_s$ and $X_cY_cZ_c$, of the RNA are defined in Fig. 1. The world coordinate system $X_wY_wY_w$ is defined as the IMU's coordinate system at the first keyframe. Throughout the paper, a matrix is represented by a bold capital letter and a vector a bold lowercase letter. The right superscript of a letter represents the coordinate system, in which the variable is expressed, while a left superscript indicates the type of variable. For example, the camera transformation matrix from keyframe i to keyframe j is denoted by $\mathbf{T}_{\mathbf{ij}}^{\mathbf{c}}$. Similarly, the transformation matrix from $X_wY_wY_w$ to $X_sY_sZ_s$ at keyframe i is denoted $\mathbf{T}_{\mathbf{i}}^{\mathbf{ws}}$. We use Riemannian geometry notation to describe a rigid body's pose. Let the IMU's pose at keyframe i be denoted $\boldsymbol{\xi}_{\boldsymbol{i}}^{\boldsymbol{w}} = [(\boldsymbol{\omega}_{\boldsymbol{i}}^{\boldsymbol{w}})^{\boldsymbol{T}}, (\boldsymbol{t}_{\boldsymbol{i}}^{\boldsymbol{w}})^{\boldsymbol{T}}]^{\boldsymbol{T}}$, where $\boldsymbol{\omega}_{\boldsymbol{i}}^{\boldsymbol{w}} \in \mathbb{R}^{\mathbf{3}}$ and $\boldsymbol{t}_{\boldsymbol{i}}^{\boldsymbol{w}} \in \mathbb{R}^{\mathbf{3}}$ are the IMU's orientation and position in $X_wY_wY_w$, respectively. The relation between $\boldsymbol{\omega}_{\boldsymbol{i}}^{\boldsymbol{w}}$ and the rotation matrix $\mathbf{R}_{\mathbf{i}}^{\mathbf{ws}} \in SO(3)$ is determined by the exponential/logarithm map [19]. Specifically, $\mathbf{R}_{\mathbf{i}}^{\mathbf{ws}} = \exp((\boldsymbol{\omega}_{\boldsymbol{i}}^{\boldsymbol{w}})^{\wedge})$ and $\boldsymbol{\omega}_{\boldsymbol{i}}^{\boldsymbol{w}} = (\log(\mathbf{R}_{\mathbf{i}}^{\mathbf{ws}}))^{\vee}$, where $(\boldsymbol{\omega}_{\boldsymbol{i}}^{\boldsymbol{w}})^{\wedge} \in \mathfrak{so}(3)$ is the corresponding Lie algebra element of $\boldsymbol{\omega}_{\boldsymbol{i}}^{\boldsymbol{w}}$, the *hat* operator $(\cdot)^{\wedge}$ maps a vector to a skew symmetric matrix in $\mathfrak{so}(3)$, and the *vee* operator $(\cdot)^{\vee}$ is the inverse operation. The transformation matrix from $X_sY_sZ_s$ to $X_wY_wY_w$ is $\mathbf{T}_{\mathbf{i}}^{\mathbf{ws}} = \begin{bmatrix} \mathbf{R}_{\mathbf{i}}^{\mathbf{ws}} & \boldsymbol{t}_{\boldsymbol{i}}^{\boldsymbol{w}} \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} \exp(\boldsymbol{\omega}_{\boldsymbol{i}}^{\boldsymbol{w}})^{\wedge} & \boldsymbol{t}_{\boldsymbol{i}}^{\boldsymbol{w}} \\ 0 & 1 \end{bmatrix}$ and the IMU pose $\boldsymbol{\xi}_{\boldsymbol{i}}^{\boldsymbol{w}}$ can be computed from $\mathbf{T}_{\mathbf{i}}^{\mathbf{ws}}$ by

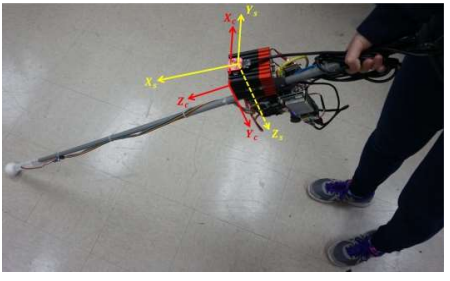


Figure 1: RNA and coordinate systems

$$\boldsymbol{\xi}_{\boldsymbol{j}}^{\boldsymbol{w}} = \begin{bmatrix} \boldsymbol{\omega}_{\boldsymbol{j}}^{\boldsymbol{w}} \\ \boldsymbol{t}_{\boldsymbol{j}}^{\boldsymbol{w}} \end{bmatrix} = \begin{bmatrix} \log(\exp((\boldsymbol{\omega}_{\boldsymbol{i}}^{\boldsymbol{w}})^{\wedge})\exp((\boldsymbol{\omega}_{\boldsymbol{ij}}^{\boldsymbol{s}})^{\wedge}))^{\vee} \\ \mathbf{R}_{\mathbf{i}}^{\mathbf{ws}}\boldsymbol{t}_{\boldsymbol{ij}}^{\boldsymbol{s}} + \boldsymbol{t}_{\boldsymbol{i}}^{\boldsymbol{w}} \end{bmatrix} \tag{1}$$

Let $\boldsymbol{x}_{\boldsymbol{i}}^{\boldsymbol{w}} = [(\boldsymbol{\xi}_{\boldsymbol{i}}^{\boldsymbol{w}})^{\boldsymbol{T}}, (\boldsymbol{v}_{\boldsymbol{i}}^{\boldsymbol{w}})^{\boldsymbol{T}}]^{\boldsymbol{T}}$ denote the IMU's state and $\boldsymbol{b}_{\boldsymbol{i}} = [(\boldsymbol{b}_{\boldsymbol{i}}^{\boldsymbol{\omega}})^{\boldsymbol{T}}, (\boldsymbol{b}_{\boldsymbol{i}}^{\boldsymbol{a}})^{\boldsymbol{T}}]^{\boldsymbol{T}}$ its calibration parameters, where $\boldsymbol{v}_{\boldsymbol{i}}^{\boldsymbol{w}} \in \mathbb{R}^{\mathbf{3}}$ is the IMU's linear velocity, and $\boldsymbol{b}_{\boldsymbol{i}}^{\boldsymbol{\omega}}$ and $\boldsymbol{b}_{\boldsymbol{i}}^{\boldsymbol{a}}$ are the bias of the gyroscope and accelerometer, respectively. At keyframe i, a set of planes are detected and described in $X_sY_sY_s$ as $\boldsymbol{l}_{\boldsymbol{ik}}^{\boldsymbol{s}} = [\boldsymbol{n}_{\boldsymbol{ik}}^{\boldsymbol{s}}, d_{ik}^{s}]^{T}$ for $k = 1, \cdots, K$, where $\boldsymbol{n}_{\boldsymbol{ik}}^{\boldsymbol{s}}$ and d_{ik}^{s} are the plane's normal vector and the distance from the origin of $X_sY_sY_s$ to the plane, respectively. The plane's representation in $X_wY_wZ_w$ is given by:

$$\boldsymbol{l}_{\boldsymbol{ik}}^{\boldsymbol{w}} = \boldsymbol{g}(\mathbf{T}_{\mathbf{i}}^{\mathbf{sw}}, \boldsymbol{l}_{\boldsymbol{ik}}^{\boldsymbol{s}}) = \begin{bmatrix} \boldsymbol{n}_{\boldsymbol{ik}}^{\boldsymbol{w}} \\ d_{ik}^{w} \end{bmatrix} = \begin{bmatrix} (\mathbf{R}_{\mathbf{i}}^{\mathbf{sw}})^{\boldsymbol{T}}\boldsymbol{n}_{\boldsymbol{ik}}^{\boldsymbol{s}} \\ \boldsymbol{t}_{\boldsymbol{i}}^{\boldsymbol{sw}}\boldsymbol{n}_{\boldsymbol{ik}}^{\boldsymbol{s}} + d_{ik}^{s} \end{bmatrix} \tag{2}$$

3 Factor graph formulation

In this paper, we use a factor graph to model the SLAM problem. A factor graph [21], [22] is a bipartite graph consisting of nodes and edges. There are two types of nodes: variable nodes and factor nodes. A variable node represents the random variables to be estimated

and a factor node the probabilistic properties of these variables. We denote the set of variables up to m keyframes by $\Theta_m = \{\mathcal{X}_m, \mathcal{B}_m, \mathcal{L}_m\}$, where $\mathcal{X}_m = \{\boldsymbol{x_i^w}\}$, $\mathcal{B}_m = \{\boldsymbol{b_i}\}$ and $\mathcal{L}_m = \{\boldsymbol{l_i^w}\}$ for $i = 1, \cdots, m$. The graph is denoted by $G_m = (\mathcal{F}_m, \Theta_m, \mathcal{E}_m)$, where variable node $\boldsymbol{\theta_i} \in \Theta_m$ represents an unknown random variable to be estimated; factor node $f_i \in \mathcal{F}_m$ represents the variable's probabilistic information; and edges $\varepsilon_{ij} \in \mathcal{E}_m$ indicates the connection/relation between nodes f_i and θ_j. G_m defines the factorization of a function given by:

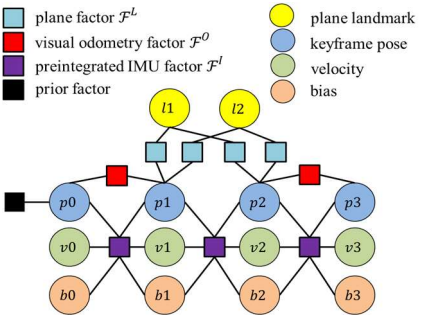


Figure 2: Factor graph structure

$$f(\Theta_m) = \prod_i f_i(\boldsymbol{\theta_i}) \tag{3}$$

Assuming a Gaussian measurement model, f_i, can be computed by:

$$f_i(\boldsymbol{\theta_i}) \propto exp(-\frac{1}{2} r_i^2) \tag{4}$$

Here, $r_i^2 = \boldsymbol{e_i^T \Sigma_i^{-1} e_i}$ is the squared Mahalanobis distance. The residual error $\boldsymbol{e_i}$ is computed by $\boldsymbol{e_i} = h_i(\boldsymbol{\theta_i}) - \boldsymbol{z_i}$, where $h_i(\boldsymbol{\theta_i})$ is the estimated measurement; $\boldsymbol{z_i}$ is the actual measurement; and $\boldsymbol{\Sigma_i}$ is the covariance matrix. The solution to the SLAM problem is to find the optimum value Θ_m^* that maximizes $f(\Theta_m)$ in (3):

$$\Theta_m^* = \underset{\Theta_k}{\operatorname{argmax}} \prod_i f_i(\boldsymbol{\theta_i}) \tag{5}$$

This is equivalent to the nonlinear Least-Square (LS) solution:

$$\Theta_m^* = \underset{\Theta_m}{\operatorname{argmax}}(-\sum \log f(\Theta_m)) = \underset{\Theta_m}{\operatorname{argmin}}(\textstyle\sum_{i=1}^{m} r_i^2) \tag{6}$$

In the factor graph, factor nodes f_i consists of PC measurements of both VO and IMU and the biases of the IMU [19]. In this paper, we extend Forster's VIO method [19] by adding plane features as landmarks to the factor graph to reduce pose estimation error. A factor graph example in our case is shown in Fig. 2. The LS problem is given by:

$$\Theta_m^* = \underset{\Theta_k}{\operatorname{argmin}} \left(r_o^2 + \textstyle\sum_{ij} \left({}^{c}\boldsymbol{r_{ij}} \right)^2 + \sum_{kl} ({}^{s}\boldsymbol{r_{kl}})^2 + \sum_{pq} \left({}^{l}\boldsymbol{r_{pq}} \right)^2 \right) \tag{7}$$

where $\boldsymbol{r_o}$, ${}^{c}\boldsymbol{r_{ij}}$, ${}^{s}\boldsymbol{r_{kl}}$ and ${}^{l}\boldsymbol{r_{pq}}$ are the residual errors related to the factors of prior, VO, IMU and plane measurements, and ${}^{c}\boldsymbol{\Sigma_{pq}}$, ${}^{s}\boldsymbol{\Sigma_{pq}}$ and ${}^{l}\boldsymbol{\Sigma_{pq}}$ are the corresponding covariance matrices. Computation of $\boldsymbol{r_o}$ and ${}^{s}\boldsymbol{r_{kl}}$ for the IMU factors $\mathcal{F}^I$ follows Forster's VIO method [19]. The error functions and covariance matrices related to $\mathcal{F}^O$ and $\mathcal{F}^L$ are detailed as follows.

3.1 Visual odometry factor $\mathcal{F}^O$

VO estimates PC between keyframes i and j by extracting visual features from image i and tracking them onto image j. In this work, SIFT features [23] are used for VO. Note that we use the keypoints only. Given the PC $\hat{\mathbf{T}}_{\mathbf{ij}}^{\mathbf{c}}$ (calculated from the IMU measurements), a space correspondence search is employed to find the matched features between the two images and the RANSAC process [20] is used to find inliers, from which PCE is computed. Finally, the Bundle Adjustment (BA) method [12] is applied to the inliers to refine the PCE. We observed from our experiments that the PCE produced by BA could be inaccurate when the inlier number is low. Fig. 3 shows a case with 8 inliers. The point

cloud data of the two keyframes are registered based on the estimated PC. The misalignment of the wall surfaces indicates an inaccurate PCE. In this work, we propose to use the associated planes between the two keyframes to verify the PCE accuracy of BA and thus determine rejection or acceptance of the PCE. The proposed VO method is illustrated in Algorithm 1. The method computes $[\mathbf{T}_{ij}^{c}, \mathbf{\Sigma}_{ij}^{c}]$ (PCE and the covariance matrix) and $\boldsymbol{L}$ (associated plane-pairs) between key-frames i and j. If it returns with 0 (success), $[\mathbf{T}_{ij}^{c}, \mathbf{\Sigma}_{ij}^{c}]$ is used to construct the factor graph. Otherwise, $\boldsymbol{L}$ is used to add plane-nodes to the factor graph. The PlaneDetection function (in line 4) extracts planes from keyframe i by a RANSAC plane-fitting process if the PlaneAssociation function does not find all associated planes in the previous step. Otherwise, it simply inherits the planes. It is noted that the algorithm returns 1 if the inlier number N is smaller than 12. This is because that the SR4000-based VO requires at least 12 matched points to compute an accurate PCE [13]. The rest of the algorithm is self-explanatory. Some technical details of plane factor, plan association and plane consistency check are given in the following sections.

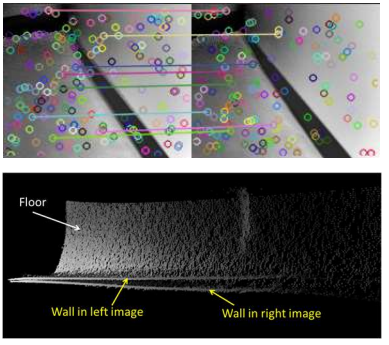


Figure 3: Top: matched visual features (8 inliers); Bottom: misalignment of the wall surfaces due to PCE error.

Algorithm 1 The proposed VO:

1: reset $[\mathbf{T}_{ij}^{c}, \mathbf{\Sigma}_{ij}^{c}], \boldsymbol{L}$
2: $\Theta = \text{FeatureMatch}(\text{I}_\text{i}, \text{D}_\text{i}, \text{I}_\text{j}, \widehat{\mathbf{T}}_{ij}^{c})$
3: $[\widetilde{\mathbf{T}}_{ij}^{c}, \text{N}, \text{S}] = \text{RANSAC\_InliersDetection}(\Theta)$
4: $l_i^c = \text{PlaneDetection}(\text{D}_\text{i})$
5: if l_i^c is not empty
6: $\quad \boldsymbol{L} = \text{PlaneAssociation}(\boldsymbol{l}_i^c, \text{I}_\text{i}, \text{D}_\text{i}, \text{I}_\text{j}, \text{D}_\text{j}, \widehat{\mathbf{T}}_{ij}^{c})$
7: ***if*** N < 12
8: $\quad$ ***return 1***
9: $[\mathbf{T}_{ij}^{c}, \mathbf{\Sigma}_{ij}^{c}] = \text{BA}(\widetilde{\mathbf{T}}_{ij}^{c}, S)$
10: ***if*** $\boldsymbol{L}$ is ***empty***
11: $\quad$ ***return* 0**
12: $\text{E}_\text{r} = \text{PlaneConsistencyCheck}(\boldsymbol{L}, [\mathbf{T}_{ij}^{c}, \mathbf{\Sigma}_{ij}^{c}])$
13: ***if*** $\text{E}_\text{r} < \chi^2_{3,0.9}$
14: $\quad$ ***return* 0**
15: ***else***
16: $\quad$ ***return 1***

$\text{I}_\text{i}, \text{D}_\text{i}$: Image and Depth Image of keyframe i
$\text{I}_\text{j}, \text{D}_\text{j}$: Image and Depth Image of keyframe j
Θ: matched SIFT features, S: a set of inliers
N: number of inliers

3.2 Plane factor $\mathcal{F}^{L}$

Let the k^{th} plane detected at keyframe i (with pose $\boldsymbol{\xi}_i^w$) be denoted by $\boldsymbol{l}_{ik}^{c} = [\boldsymbol{n}_{ik}^{c}, d_{ik}^{c}]^T$ in $X_cY_cZ_c$. In this work, the plane's covariance matrix $\mathbf{\Sigma}_{ik}^{c}$ is computed by the method in [24] and its variables and factors for graph construction are calculated according to [21]. The elements of $\boldsymbol{n}_{ik}^{c}$ represent a point on the surface of sphere S^2—a manifold consisting of all the unit vectors in $\mathbb{R}^3$: $S^2 = \{\boldsymbol{n} \in \mathbb{R}^3\}$, where $\boldsymbol{n} = [n_x, n_y, n_z]^T$ is a unit vector. Each point in the space of S^2 corresponds to a normal vector of a plane. The tangent space $T_{\boldsymbol{n}}(S^2)$ at a point $\boldsymbol{n}$ of S^2 is composed of 3D vectors $\boldsymbol{\varphi}$: $T_{\boldsymbol{n}}(S^2) \triangleq \{\boldsymbol{\varphi} \in \mathbb{R}^3 | \, \boldsymbol{n}^T\boldsymbol{\varphi} = 0\}$. A basis $\mathbf{B}_\mathbf{n} = [\boldsymbol{b}_1 | \boldsymbol{b}_2]$ for $T_{\boldsymbol{n}}(S^2)$ is computed by $\boldsymbol{b}_1 = \boldsymbol{b}'/||\boldsymbol{b}'||$ and $\boldsymbol{b}_2 = \boldsymbol{n} \times \boldsymbol{b}_1$, where $\boldsymbol{b}' = \boldsymbol{n} \times \boldsymbol{a}$. To ensure that $\boldsymbol{a}$ is not parallel to $\boldsymbol{n}$, we set $\boldsymbol{a}$ to $[1,0,0]^T$, $[0,1,0]^T$ or $[0,0,1]^T$ if n_x, n_y or n_z dominates the other two elements. A vector $\boldsymbol{\varphi}$ in $T_{\boldsymbol{n}}(S^2)$ is represented by $\boldsymbol{\varphi} = \mathbf{B}_\mathbf{n}\boldsymbol{\rho}$, where $\boldsymbol{\rho} \in \mathbb{R}^2$ is the 2D coordinate in the tangent plane with basis $\mathbf{B}_\mathbf{n}$. Using the local tangent space, $\mathbf{\Sigma}_{ik}^{c}$ can be re-written as:

$$\mathbf{S}_{\mathbf{ik}}^{\mathbf{c}} = \begin{bmatrix} \mathbf{B}_{\boldsymbol{n}_{ik}^{c}}^{T} \boldsymbol{\Sigma}_{\boldsymbol{n}_{ik}^{c}} \mathbf{B}_{\boldsymbol{n}_{ik}^{c}} & \mathbf{0}_{2\times 1} \\ \mathbf{0}_{1\times 2} & \sigma_{d_{ik}^{c}}^{2} \end{bmatrix}. \tag{8}$$

The covariance of the plane factor (in $X_s Y_s Z_s$) is then computed by ${}^{l}\boldsymbol{\Sigma}_{\boldsymbol{ik}} = \mathbf{J}\mathbf{S}_{\mathbf{ik}}^{\mathbf{c}}\mathbf{J}^{T}$. Here, the jacobian matrix $\mathbf{J}$ can be derived from $\boldsymbol{l}_{\boldsymbol{ik}}^{\boldsymbol{s}} = \boldsymbol{g}(\mathbf{T}^{\boldsymbol{cs}}, \boldsymbol{l}_{\boldsymbol{ik}}^{\boldsymbol{c}})$ and is given by

$$\mathbf{J} = \begin{bmatrix} \mathbf{B}_{\boldsymbol{n}_{ik}^{s}}^{T} (\mathbf{R}^{\boldsymbol{cs}})^{T} \mathbf{B}_{\boldsymbol{n}_{ik}^{c}} & \mathbf{B}_{\boldsymbol{n}_{ik}^{c}}^{T} \boldsymbol{t}^{\boldsymbol{cs}} \\ \mathbf{0}_{1\times 2} & 1 \end{bmatrix}. \tag{9}$$

The predicted measurement of plane $\boldsymbol{l}_{\boldsymbol{ik}}^{\boldsymbol{s}}$ is given by $\hat{\boldsymbol{l}}_{\boldsymbol{ik}}^{\boldsymbol{s}} = \left[\hat{\boldsymbol{n}}_{\boldsymbol{ik}}^{\boldsymbol{s}}, \hat{d}_{ik}^{s}\right]^{T} = \boldsymbol{g}(\mathbf{T}_{\mathbf{i}}^{\mathbf{ws}}, \boldsymbol{l}_{\boldsymbol{ik}}^{\boldsymbol{w}})$. The error vector of the plane factor is calculated by

$${}^{l}\boldsymbol{e}_{\boldsymbol{ik}} = [\boldsymbol{\rho}_{\boldsymbol{ik}}^{T}, \hat{d}_{ik}^{s} - d_{ik}^{s}]^{T}, \tag{10}$$

where $\boldsymbol{\rho}_{\boldsymbol{ik}} = \mathbf{B}_{\hat{\boldsymbol{n}}_{\boldsymbol{ik}}^{s}}^{T} \frac{\theta}{\sin(\theta)} (\boldsymbol{n}_{\boldsymbol{ik}}^{\boldsymbol{s}} - \hat{\boldsymbol{n}}_{\boldsymbol{ik}}^{\boldsymbol{s}} * \cos(\theta))$ with $\theta = \cos^{-1}((\hat{\boldsymbol{n}}_{\boldsymbol{ik}}^{\boldsymbol{s}})^{T} \boldsymbol{n}_{\boldsymbol{ik}}^{\boldsymbol{s}})$ [25]

4 Plane association and consistency check

4.1 Plane Association

Assume $\hat{\mathbf{T}}_{\mathbf{ij}}^{\mathbf{c}}$ (i.e., the camera's PC measured by the IMU) is close to the true value. $\boldsymbol{l}_{\boldsymbol{ik}}^{\boldsymbol{c}}$ (the k^{th} plane observed at keyframe i) can be tracked into keyframe j as $\hat{\boldsymbol{l}}_{\boldsymbol{jk}}^{\boldsymbol{c}} = \left[\hat{\boldsymbol{n}}_{\boldsymbol{jk}}^{\boldsymbol{c}}, \hat{d}_{jk}^{c}\right]^{T} = \boldsymbol{g}(\hat{\mathbf{T}}_{\mathbf{ij}}^{\mathbf{c}}, \boldsymbol{l}_{\boldsymbol{ik}}^{\boldsymbol{c}})$ to speed up the search for its associated plane $\boldsymbol{l}_{\boldsymbol{jk}}^{\boldsymbol{c}}$ at keyframe j. The distances between all points of keyframe j and $\hat{\boldsymbol{l}}_{\boldsymbol{jk}}^{\boldsymbol{c}}$ are computed. Those points with a distance below $2\sigma_{ik}$ are treated as data points on plane $\hat{\boldsymbol{l}}_{\boldsymbol{jk}}^{\boldsymbol{c}}$. ($\sigma_{ik}$ is the plane fitting error of plane $\boldsymbol{l}_{\boldsymbol{ik}}^{\boldsymbol{c}}$.) Some of these data points are on plane $\boldsymbol{l}_{\boldsymbol{jk}}^{\boldsymbol{c}}$ because plane $\boldsymbol{l}_{\boldsymbol{jk}}^{\boldsymbol{c}}$ is in the neighbourhood of $\hat{\boldsymbol{l}}_{\boldsymbol{jk}}^{\boldsymbol{c}}$. In this work, a RANSAC plane-fitting process is used to extract plane $\boldsymbol{l}_{\boldsymbol{jk}}^{\boldsymbol{c}}$ from the data points. After the RANSAC process, $\boldsymbol{l}_{\boldsymbol{jk}}^{\boldsymbol{c}}$ is extended by adding data points satisfying the distance criteria. Finally, the plane's parameters are recomputed. If the number of data points of $\boldsymbol{l}_{\boldsymbol{jk}}^{\boldsymbol{c}}$ is large enough, the pair $\{\boldsymbol{l}_{\boldsymbol{ik}}^{\boldsymbol{c}}, \boldsymbol{l}_{\boldsymbol{jk}}^{\boldsymbol{c}}\}$ taken as a successful match.

4.2 Plane consistency check (PCC)

After plane association, a set of matched plane-pairs $\boldsymbol{L} = \cup_{k}\{\boldsymbol{l}_{\boldsymbol{ik}}^{\boldsymbol{c}}, \boldsymbol{l}_{\boldsymbol{jk}}^{\boldsymbol{c}}\}$ is obtained. PCC is to evaluate, based on the PCE of VO, how good the match between each plane-pair is. The evaluation result determines if the VO's estimate should be accepted. Given the PCE $\{\mathbf{T}_{\mathbf{ij}}^{\mathbf{c}}, \boldsymbol{\Sigma}_{\mathbf{ij}}^{\mathbf{c}}\}$ from VO, the prediction of plane $\boldsymbol{l}_{\boldsymbol{ik}}^{\boldsymbol{c}}$ for keyframe j is $\tilde{\boldsymbol{l}}_{\boldsymbol{jk}}^{\boldsymbol{c}} = \boldsymbol{g}(\mathbf{T}_{\mathbf{ij}}^{\mathbf{c}}, \boldsymbol{l}_{\boldsymbol{ik}}^{\boldsymbol{c}})$. The goodness of match between $\boldsymbol{l}_{\boldsymbol{jk}}^{\boldsymbol{c}}$ and $\tilde{\boldsymbol{l}}_{\boldsymbol{jk}}^{\boldsymbol{c}}$ is evaluated by the angle between $\tilde{\boldsymbol{n}}_{\boldsymbol{jk}}^{\boldsymbol{c}}$ and $\boldsymbol{n}_{\boldsymbol{jk}}^{\boldsymbol{c}}$ and the value of $\delta_d = \tilde{d}_{jk}^{c} - d_{jk}^{c}$. If the angle is larger than 5° or δ_d>0.1 m, the PCE is rejected. If the angle is less than 0.5° and δ_d<0.01 m, the PCE is accepted. For the rest, we compute the covariance of $\tilde{\boldsymbol{l}}_{\boldsymbol{jk}}^{\boldsymbol{c}}$ by:

$$\tilde{\mathbf{S}}_{\mathbf{jk}}^{\mathbf{c}} = \left(\frac{\partial \tilde{\boldsymbol{l}}_{\boldsymbol{jk}}^{\boldsymbol{c}}}{\partial \tilde{\boldsymbol{\xi}}_{\boldsymbol{ij}}^{\boldsymbol{c}}}\right) \tilde{\boldsymbol{\Sigma}}_{\mathbf{ij}}^{\mathbf{c}} \left(\frac{\partial \tilde{\boldsymbol{l}}_{\boldsymbol{jk}}^{\boldsymbol{c}}}{\partial \tilde{\boldsymbol{\xi}}_{\boldsymbol{ij}}^{\boldsymbol{c}}}\right)^{T} + \left(\frac{\partial \tilde{\boldsymbol{l}}_{\boldsymbol{jk}}^{\boldsymbol{c}}}{\partial \boldsymbol{l}_{\boldsymbol{ik}}^{\boldsymbol{c}}}\right) \mathbf{S}_{\mathbf{ik}}^{\mathbf{c}} \left(\frac{\partial \tilde{\boldsymbol{l}}_{\boldsymbol{jk}}^{\boldsymbol{c}}}{\partial \boldsymbol{l}_{\boldsymbol{ik}}^{\boldsymbol{c}}}\right)^{T} \tag{11}$$

where $\frac{\partial \tilde{\boldsymbol{l}}_{jk}^{c}}{\partial \tilde{\boldsymbol{\xi}}_{ij}^{c}} = \begin{bmatrix} -\mathbf{B}_{\tilde{\boldsymbol{n}}_{jk}^{c}}^{T} \tilde{\mathbf{R}}_{ij}^{c} [\boldsymbol{n}_{jk}^{c}]_{\times} & \mathbf{0}_{2\times 3} \\ \mathbf{0}_{1\times 3} & [\tilde{\boldsymbol{n}}_{jk}^{c}]^{T} \end{bmatrix}$ and $\frac{\partial \tilde{\boldsymbol{l}}_{jk}^{c}}{\partial \boldsymbol{l}_{ik}^{c}} = \begin{bmatrix} \mathbf{B}_{\tilde{\boldsymbol{n}}_{jk}^{c}}^{T} \tilde{\mathbf{R}}_{ij}^{c} \mathbf{B}_{\boldsymbol{n}_{ik}^{c}} & \mathbf{B}_{\boldsymbol{n}_{ik}^{c}}^{T} \tilde{\boldsymbol{t}}_{ij}^{c} \\ \mathbf{0}_{1\times 2} & 1 \end{bmatrix}$. $[\boldsymbol{n}]_{\times}$ is the skew matrix of $\boldsymbol{n}$. We use error $\boldsymbol{r}(\tilde{\boldsymbol{n}}_{jk}^{c}, \boldsymbol{n}_{jk}^{c}) = \mathbf{B}_{\tilde{\boldsymbol{n}}_{jk}^{c}}^{T} \boldsymbol{n}_{jk}^{c}$ [21] to describe the difference between $\tilde{\boldsymbol{n}}_{jk}^{c}$ and $\boldsymbol{n}_{jk}^{c}$. An error vector $\boldsymbol{e}_{k}$ for the plane pair $\{\boldsymbol{l}_{ik}^{c}, \boldsymbol{l}_{jk}^{c}\}$ is computed as $\boldsymbol{e}_{k} = \begin{bmatrix} \boldsymbol{r}(\tilde{\boldsymbol{n}}_{jk}^{c}, \boldsymbol{n}_{jk}^{c}) \\ \tilde{d}_{jk}^{c} - d_{jk}^{c} \end{bmatrix}$, and the covariance matrix of $\boldsymbol{e}_{k}$ is given by:

$$\boldsymbol{\Sigma}_{\boldsymbol{e}_{k}} = \left(\frac{\partial \boldsymbol{e}_{k}}{\partial \tilde{\boldsymbol{l}}_{jk}^{c}}\right) \tilde{\mathbf{S}}_{jk}^{c} \left(\frac{\partial \boldsymbol{e}_{k}}{\partial \tilde{\boldsymbol{l}}_{jk}^{c}}\right)^{T} + \left(\frac{\partial \boldsymbol{e}_{k}}{\partial \boldsymbol{l}_{jk}^{c}}\right) \mathbf{S}_{jk}^{c} \left(\frac{\partial \boldsymbol{e}_{k}}{\partial \boldsymbol{l}_{jk}^{c}}\right)^{T} \tag{12}$$

where $\frac{\partial \boldsymbol{e}_{k}}{\partial \tilde{\boldsymbol{l}}_{jk}^{c}} = \begin{bmatrix} [\boldsymbol{n}_{jk}^{c}]^{T} \left(\frac{\partial \tilde{\boldsymbol{b}}_{1}}{\partial \tilde{\boldsymbol{n}}_{jk}^{c}}\right) & 0 \\ [\boldsymbol{n}_{jk}^{c}]^{T} \left(\frac{\partial \tilde{\boldsymbol{b}}_{2}}{\partial \tilde{\boldsymbol{n}}_{jk}^{c}}\right) & 0 \\ \mathbf{0}_{1\times 2} & 1 \end{bmatrix}$, with $\frac{\partial \tilde{\boldsymbol{b}}_{1}}{\partial \tilde{\boldsymbol{n}}_{jk}^{c}} = \mathbf{J}_{\boldsymbol{b}'} [-\boldsymbol{a}]_{\times} \mathbf{B}_{\tilde{\boldsymbol{n}}_{jk}^{c}}$. $\mathbf{B}_{\tilde{\boldsymbol{n}}_{jk}^{c}} = [\tilde{\boldsymbol{b}}_{1} | \tilde{\boldsymbol{b}}_{2}]$, and $\mathbf{J}_{\boldsymbol{b}'} = \frac{1}{\|\boldsymbol{b}'\|^{\frac{3}{2}}} \begin{bmatrix} b_{y}'^{2} + b_{z}'^{2} & -b_{x}' b_{y}' & -b_{x}' b_{z}' \\ -b_{x}' b_{y}' & b_{z}'^{2} + b_{x}'^{2} & -b_{y}' b_{z}' \\ -b_{x}' b_{z}' & -b_{y}' b_{z}' & b_{x}'^{2} + b_{y}'^{2} \end{bmatrix}$ where $\boldsymbol{b}' = [b_{x}', b_{y}', b_{z}']^{T} = \tilde{\boldsymbol{n}}_{jk}^{c} \times \boldsymbol{a}$ (refer to section 3.2 on how to determine $\boldsymbol{a}$). $\frac{\partial \tilde{\boldsymbol{b}}_{2}}{\partial \tilde{\boldsymbol{n}}_{jk}^{c}} = [-\tilde{\boldsymbol{b}}_{1}]_{\times} \mathbf{B}_{\tilde{\boldsymbol{n}}_{jk}^{c}} + [\tilde{\boldsymbol{n}}_{jk}^{c}]_{\times} \frac{\partial \tilde{\boldsymbol{b}}_{1}}{\partial \tilde{\boldsymbol{n}}_{jk}^{c}}$ and $\frac{\partial \boldsymbol{e}_{k}}{\partial \boldsymbol{l}_{jk}^{c}} = \begin{bmatrix} \mathbf{B}_{\tilde{\boldsymbol{n}}_{jk}^{c}}^{T} \mathbf{B}_{\boldsymbol{n}_{jk}^{c}} & \mathbf{0}_{2\times 1} \\ \mathbf{0}_{1\times 2} & -1 \end{bmatrix}$. The squared Mahalanobis distance is computed by $r_{k}^{2} = \boldsymbol{e}_{k}^{T} \boldsymbol{\Sigma}_{\boldsymbol{e}_{k}}^{-1} \boldsymbol{e}_{k}$. $\mathrm{E_r} = max(r_{k}^{2})$ is then used to evaluate the plane consistency. If $\mathrm{E_r} < \chi_{3,0.9}^{2} = 6.25$, the PCE is accepted, and otherwise rejected.

5 Experiments

We used the RNA to collect data to validate the efficacy of the proposed SLAM method. Seven datasets were obtained from the human subject experiments in the environments with lobbies, hallways and/or stairways in two buildings on campus. These datasets were acquired from the sensors by using the on-board computer (up boards computer) and processed offline by a desktop computer. When collecting the data, the subject swung the RNA and walked in a normal walking speed (average speed: 0.6 m/s) to imitate the way a blind person uses a white cane. In each of the experiments, the subject walked along a looped path and returned to the starting point. We use the End Point Error Norm (EPEN) of the trajectory as the performance metric. The EPENs (both absolute value and percentage of path-length) produced by PAVIO for the seven datasets are tabulated in Table I, where the results from the state-of-the-art VIO method [19] (using the inlier threshold N=12) are compared. The percentage values allow us to compute the statistical results (mean and standard deviation over the seven datasets) of the methods and compare their overall performances.

From table 1, it can be seen that: (1) PAVIO outperforms VIO in most of the cases; (2) PAVIO has an overall much better performance (mean EPEN: 2.63%) in pose estimation

than VIO (mean EPEN: 6.06%); (3) PAVIO has a more stable performance (standard deviation of EPEN: 1.3%) than VIO (standard deviation: 8.22), meaning that it is more robust to the variation of operating environments. As shown in table 1, the VIO's performance may be improved by adjusting the value of N. However, PAVIO still exhibits a better overall performance (in term of pose estimation accuracy and robustness) than VIO. Fig. 4 shows the trajectories generated by these methods over the point cloud map built by PAVIO for each experiment. It can be seen that PAVIO's trajectories are more accurate. Fig. 5 shows the percentage of times the standard VO's results (step 9) are accepted/rejected by PCC for datasets 5 and 6. As the environment of dataset 5 was less feature-rich than that of dataset 6, PCC rejected standard VO's results more often, resulting in a lower acceptance rate (and a higher rejection rate) of VO results.

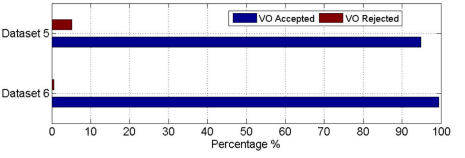


Figure 5: Accepted/rejected standard VO results of dataset 5 and 6

Dataset	Carpet	Path-length (meters)	EPEN (meters, %)				
			VIO (N=5)	VIO (N=12)	VIO (N=20)	VIO (N=30)	PAVIO
1	Patterned	118	3.60, 3.05	3.60, 3.05	3.61, 3.05	3.29, 2.79	**3.23, 2.74**
2	Patterned	128	2.26, 1.77	3.32, 2.59	2.09, 1.63	2.38, 1.86	**1.97, 1.54**
3	Solid-colored	116	22.40, 19.31	**4.51, 3.89**	7.22, 6.22	7.41, 6.39	4.57, 3.94
4	Solid-colored	70	16.77, 23.96	17.21, 24.59	4.56, 6.51	15.71, 22.44	**3.31, 4.73**
5	Solid-colored	116	3.59, 3.09	4.26, 3.67	6.65, 5.73	5.25, 4.53	**1.77, 1.53**
6	Patterned	115	**1.11, 0.97**	1.41, 1.23	1.33, 1.16	2.51, 2.18	1.57, 1.37
7	Patterned	133	7.38, 5.55	4.56, 3.43	3.45, 2.59	5.91, 4.44	**3.43, 2.58**
Average (mean / standard deviation)			8.24 / 9.35	6.06 / 8.22	3.84 / 2.26	6.38, 7.26	2.63, 1.3

Table 1: Comparison for the Final End Position Errors

6 Conclusion and Future Work

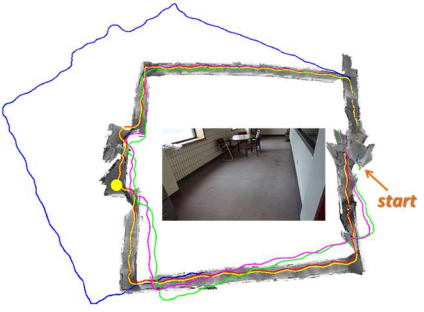
We have presented a new SLAM method for pose estimation of a robotic navigation aid (RNA) that uses a 3D time-of-flight camera and an IMU for assistive navigation. The method, called plane-aided visual-inertial odometry (PAVIO), extracts planes from the 3D point data of the camera current frame and track them onto the next camera frame by using the IMU's inertial measurements. The tracking result is used to evaluate the pose change estimation (PCE) of the visual odometry (VO) and accept the PCE only if it is accurate. The accepted VO outputs, the information of the extracted planes, and the IMU's measurements over time are used to create a factor graph. By optimizing the graph, the method improves the estimation accuracy of the IMU bias and reduces the pose estimation error for the RNA. Experimental results with the RNA demonstrate that the proposed method outperforms the state-of-the-art VIO method in pose estimation in term of both accuracy and repeatability. The proposed method can be used to produce more accurate 3D map of the operating environment for object/obstacle detection and locate a visually impaired traveller in the environment for assistive navigation of the visually impaired. It can also be applied to autonomous navigation of mobile robots.

7 Acknowledgements

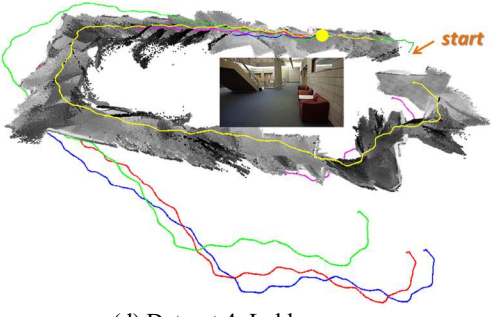
This work was supported by the NIBIB and NEI of the NIH under award R01EB018117.

(a) Dataset 1: Lobby and Hallway

(b) Dataset 2: Hallway A

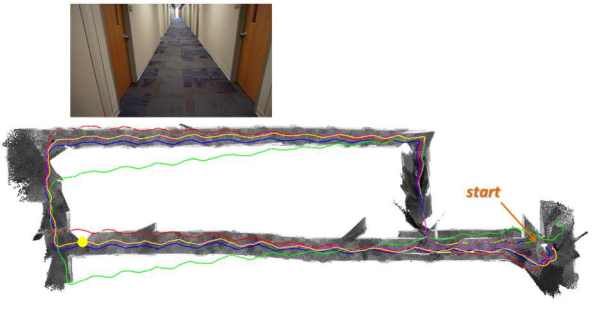


(c) Dataset 3: Hallway B

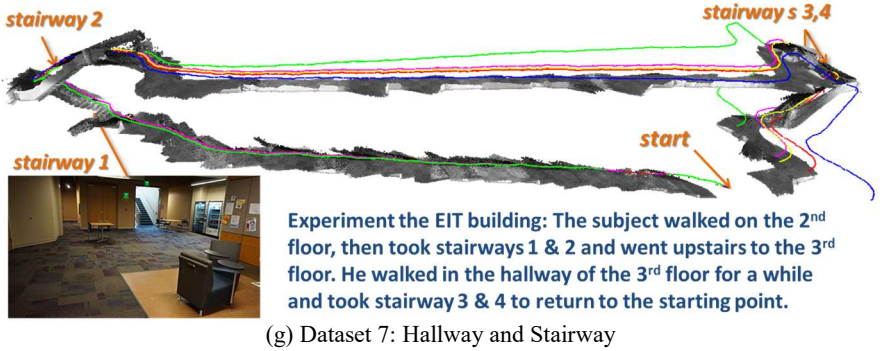


(d) Dataset 4: Lobby

(e) Dataset 5: Hallway C



(f) Dataset 6: Hallway D



(g) Dataset 7: Hallway and Stairway

Figure 4. Trajectories and 3D Map of each experiment: the trajectories produced by VIO (N=5), VIO (N=12), VIO (N=20), VIO (N=20) and PAVIO are plotted in blue, red, purple, green, and yellow, respectively. The yellow dot shows the location where the snapshot was taken.

References

[1] J. A. Hesch and S. I. Roumeliotis. Design and analysis of a portable indoor localization aid for the visually impaired. *Int. J. Robot. Res*., vol. 29, no. 11, pp. 1400-1415, 2010.

[2] S. Treuillet, E. Royer, T. Chateau, M. Dhome, J.M. Lavest. Body Mounted Vision System for Visually Impaired Outdoor and Indoor Wayfinding Assistance. in *CVHI*, 2007.

[3] J. M. Saez, F. Escolano, and A. Penalver. First steps towards stereo-based 6DOF SLAM for the visually impaired. *IEEE Int. Conf. Comput. Vision Pattern Recognition*., pp. 23-23, 2005.

[4] V. Pradeep, G. Medioni, and J. Weiland. Robot vision for the visually impaired. in *Proc. IEEE Int. Conf. Comput. Vision Pattern Recogn*, pp. 15-22, 2010.

[5] Y. H. Lee and G. Medioni. RGB-D camera based navigation for the visually impaired. in *Proc. Workshop RGB-D: Adv. Reasoning With Depth Camera*, pp. 1-6, 2011.

[6] H. Takizawa, S. Yamaguchi, M. Aoyagi, N. Ezaki and S. Mizuno. Kinect cane: An assistive system for the visually impaired based on three-dimensional object recognition. in *IEEE/SICE International Symposium on System Integration (SII)*, pp. 740-745, 2012.

[7] H. Zhang and C.Ye. An Indoor Wayfinding Method for Robotic Navigation Aids. in *Proc. IEEE Int. Conf. Robotics and Biomimetics*, 2016.

[8] C. Ye, H. Soonhac, X. Qian, and W. Wu. Co-Robotic Cane: A New Robotic Navigation Aid for the Visually Impaired. *IEEE Systems, Man, and Cybernetics Magazine* 2, no. 2: 33-42, 2016.

[9] J. Engel, T. Schöps, and D. Cremers. LSD-SLAM: Large-scale direct monocular SLAM. in *Proc. European Conference on Computer Vision*, pp. 834–849, 2014.

[10] F. Endres, J. Hess, N. Engelhard, J. Sturm, D. Cremers, and W. Burgard. An evaluation of the RGB-D SLAM system. in *Proc IEEE Int. Conf. Robotics and Automation*, pp 1691-1696, 2012.

[11] J. Engel, V. Koltun, and D. Cremers. Direct sparse odometry. *arXiv:1607.02565* ,2016.

[12] R. Mur-Artal., JMM. Montiel. and J.D. Tardós. Orb-slam: a versatile and accurate monocular slam system. *IEEE Transactions on Robotics*, 31(5), pp.1147-1163. 2015.

[13] C. Ye, S. Hong, and A. Tamjidi. 6-DOF pose estimation of a robotic navigation aid by tracking visual and geometric features, *IEEE Trans. Autom. Sci. Eng*., vol. 12(4), pp. 1169-1180. 2015.

[14] A. Trevor, John Rogers, and H. Christensen. Planar surface SLAM with 3D and 2D sensors. in *Proc. IEEE Int. Conf. Robot. Autom*., pp. 3041-3048, 2012.

[15] M. Dou, L. Guan, J.-M. Frahm, and H. Fuchs. Exploring High-Level Plane Primitives for Indoor 3D Reconstruction with a Hand-held RGB-D Camera. in *Proc. Computer Vision-ACCV Workshops*, vol 7729, pp. 94-108. 2012.

[16] H. Zhang and C. Ye. An Indoor Wayfinding System based on Geometric Features Aided Graph SLAM for the Visually Impaired. IEEE Transactions on Neural Systems and Rehabilitation Engineering, pp(99): 1-1, 2017. (DOI: 10.1109/TNSRE.2017.2682265)

[17] T. Lupton and S. Sukkarieh. Visual-inertial-aided navigation for high dynamic motion in built environments without initial conditions. *IEEE Transactions on Robotics*, 28(1):61–76, 2012.

[18] V. Indelman, S. Williams, M. Kaess, and F. Dellaert. Information fusion in navigation systems via factor graph based incremental smoothing. *Int. Journal of Robotics and Autonomous Systems* (RAS), 61(8):721–738, 2013.

[19] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza. On-Manifold Preintegration for Real-Time Visual--Inertial Odometry. *IEEE Transactions on Robotics*, 2016.

[20] C. Ye and M. Bruch. A visual odometry method based on the SwissRanger SR-4000. in *Proc. Unmanned Syst. Technol. XII Conf. SPIE Defense, Security, and Sensing Symp*., vol.7692. 2010.

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

[22] F. Dellaert, and K. Michael. Square Root SAM: Simultaneous localization and mapping via square root information smoothing. *International Journal of Robotics Research* 25, no. 12 1181-1203, 2006.

[23] D. G. Lowe. Distinctive Image Features From Scale-Invariant Keypoints. *Int. J. Computer Vision*, vol. 60, no. 2, pp. 91-110, 2004.

[24] J. W. Weingarten, G. Gruener, and R. Siegwart. Probabilistic plane fitting in 3D and an application to robotic mapping. *in IEEE International Conference on Robotics and Automation,* (ICRA) vol. 1, pp. 927-932, 2004.

[25] http://ani.stat.fsu.edu/~anuj/CVPR_Tutorial/Part2.pdf