Camera Intrinsic Parameters Estimation by Visual–Inertial Odometry for a Mobile Phone With Application to Assisted Navigation

Lingqiu Jin, He Zhang, and Cang Ye, *Senior Member, IEEE*

***Abstract*—The increasing computing and sensing capabilities of modern mobile phones have spurred research interests in developing new visual–inertial odometry (VIO) techniques to turn a smartphone into a self-contained vision-aided inertial navigation system for various applications. Smartphones nowadays use cameras with optical image stabilization (OIS) technology to reduce image blurs. However, the mechanism may result in varying camera intrinsic parameters (CIP), which must be taken into account in VIO computation. In this article, we first develop a linear model to relate the CIP with the inertial measurement unit measured acceleration. Based on the model, we introduce a new VIO method, called CIP-VMobile, which treats CIP as state variables and tightly couples them with other state variables in a graph optimization process to estimate the optimal state. The method uses the linear model to construct a factor graph and uses the linear-model-computed values as initial CIP estimates to speed up the VIO computation and attain a better pose estimation result. Simulation and experimental results with an iPhone 7 validate the method's efficacy. Based on CIP-VMobile, we fabricated a robotic navigation aid (RNA) based on an iPhone 7 for assisted navigation. Experimental results with the RNA demonstrate CIP-VMobile's promise in real-world navigation applications.**

***Index Terms*—6-DOF camera pose estimation, robotic navigation aid (RNA), simultaneous localization and mapping (SLAM), visual–inertial odometry (VIO).**

I. Introduction

As MOBILE phones rapidly improve their computing and sensing power, there is a growing interest in the research community in using a smartphone to solve computer vision and autonomous navigation problems. In the area of robotics, a smartphone can be used as a platform to build a self-contained navigation system as it is equipped with the needed computing resources and sensors, including camera, inertial measurement unit (IMU), GPS receiver, etc. A smartphone-based solution is cost-effective, compact in size, and highly portable. Recently, a number of smartphone-based simultaneous localization and mapping (SLAM) methods have been developed for virtual/augmented reality [1]–[6] as well as autonomous navigation [7]–[13]. These SLAM methods couple the camera (visual) and IMU (inertial) data to estimate the device pose, and they fall under the category of visual–inertial odometry (VIO) approach. In the recent robotics literature, VIOs [14]–[16] have been extensively explored for robot pose estimation and three-dimensional (3-D) mapping. Some of the VIO methods have been translated and implemented on smartphones and achieved the real-time pose estimation performance [2], [9]. Pose estimation accuracy becomes a critical factor that determines if the smartphone-based VIO can be applied to virtual reality/augmented reality (VR/AR) and robotics applications. One missing link in the existing works is that the variation of the camera intrinsic parameters (CIP) caused by the optical image stabilization (OIS) [17] mechanism of the smartphone camera is not factored into the SLAM computation.

OIS has now become a mainstay feature of most smartphones. The OIS mechanism aims to reduce hand-shake blurs caused by involuntary hand tremors during image capturing. Due to the use of a small imaging sensor, a smartphone's camera required a longer exposure time than a traditional camera and is, thus, sensitive to hand tremors, which can alter the optical path of the object being imaged during the exposure time and results in a blurred image. To tackle this issue, an actuator is used to shift the lens barrel to counteract the optical path movement. Currently, the most widespread actuator is based on the voice coil motor (VCM) [18], [19], which produces a force by running a current through the coil winding amid the magnetic field. While it improves the image and video quality, the OIS mechanism results in varying CIP, which may result in unwanted pose estimation error if not considered by a SLAM method.

In this article, we introduce a new VIO method for pose estimation of a modern smartphone. The proposed method treats the CIP of the phone's camera as state variables and tightly couples them with the other state variables (including the camera poses, velocity, and IMU bias) in a graph optimization process to solve the state estimation problem. As part of the state variables,

Manuscript received January 11, 2020; revised March 25, 2020; accepted April 29, 2020. Date of publication May 25, 2020; date of current version August 13, 2020. This work was supported in part by the National Institute of Biomedical Imaging and Bioengineering and in part by the National Eye Institute of the NIH under Award R01EB018117. Recommended by Technical Editor G. M. Clayton and Senior Editor X. Chen. *(Corresponding author: Cang Ye.)*

The authors are with the Computer Science Department, Virginia Commonwealth University, Richmond, VA 23284 USA (e-mail: jinl@mymail.vcu.edu; hzhang8@vcu.edu; cye@vcu.edu).

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

Digital Object Identifier 10.1109/TMECH.2020.2997606

the CIP values are re-estimated at each iteration of the successive linearization and approximation process of the VIO, resulting in a more accurate pose estimation result. A linear model relating the IMU-measured acceleration to the CIP is created and used to constrain the CIP estimation. Using the model, the CIP values are computed first from the camera's acceleration and then used as the initial CIP to start the graph SLAM computation. The initial CIP values speed up the iterative VIO computation and improve the pose estimation accuracy. To the best of authors' knowledge, the proposed method is the first in its kind in the literature. To investigate the VIO's real-world application possibility, we developed and fabricated a robotic navigation aid (RNA) based on an iPhone 7 for assisted wayfinding for a blind traveler and carried out experiments to validate the VIO method by using the RNA prototype as a SLAM platform.

The remainder of this article is organized as follows. Section II gives an overview of the related works in the literature. Section III presents the RNA prototype and its software system. Section IV details the proposed VIO method. Section V presents the experimental results. Finally, Section VI concludes this article.

II. Related Work

A. Related Work in VIO

Among the rich literature on SLAM [14]–[16], [20]–[22], the tightly coupled VIO approach jointly fuses the raw measurements of a camera and an IMU through state filtering or batch optimization. For state filtering, the most commonly used strategy is the extended Kalman filter (EKF). MSCKF [14] is an EKF-based VIO method. It maintains several camera poses in the state vector and computes a multiconstraint state update by using the visual measurements of the same features observed at these poses. Li *et al.* [9] implement the MSCKF method on a mobile phone. The effect of the rolling-shutter camera is modeled by using the camera's translational and rotational velocities. In recent work [11], [12], camera intrinsics are added into the EKF state vector to allow the filtering process to update the intrinsics online so as to improve the camera model for more accurate pose estimation.

For batch optimization, a graph model is usually adopted for pose estimation. OKVIS [16] is a graph-optimization-based VIO that searches for the optimal states to minimize a nonlinear cost function for its graph by a repeated process of approximation and linearization of the function. To keep the computation low, OKVIS uses keyframes for graph construction and performs graph optimization only for the nodes within a sliding-window. These measures substantially reduce the amount of imaging data to be processed for real-time pose estimation. To deal with the cases where the system starts under motion and/or the IMU's initial bias is not trivial, Tong *et al.* [15] propose a similar method, called VINS-Mono. VINS-Mono is capable of initializing the visual–inertial system with good initial estimates for the state variables including the visual scale, initial attitude (with reference to the gravity direction), velocity, and gyroscope bias. In [2], the VINS-Mono method is transplanted to a mobile phone for localization of the phone for augmented reality application and renamed as VINS-Mobile. However, the OIS-induced variation of CIP is not considered. In this article, we extend the VINS-Mobile method by treating CIP as variables in the graph optimization process. The extended method is called CIP-VMobile. Compared to the existing methods [11], [12], the proposed method presents the following different features.

1) It is the first graph-optimization-based VIO that is capable of estimating the CIP of a camera with OIS mechanism and it allows to update the poses for all previous keyframes in the sliding window (while an EKF-based approach only allows to update the current keyframe).
2) It uses a model to compute the CIP values from the accelerometer readings and uses the values as the initial estimates of CIP to speed up VIO computation and reduce the pose estimation error.
3) It has been validated with a real smartphone camera with OIS function, but the existing methods have not yet.

B. Related Work in RNAs

In the literature, several vision-based RNAs have been introduced to assist blind people in wayfinding. Monocular camera [23], stereo-cameras [24], RGB-D cameras [25], and 3-D time-of-flight camera [26] have been used in these RNAs to perform visual-SLAM to estimate the camera pose. These RNAs require an off-board computer, such as a server [20], a laptop [24]–[26], or a tablet computer [27] to process a large amount of camera data for navigational decision making. The need for off-board computing resources has hampered the practical use of the RNAs. The CIP-VMobile method allows for real-time and accurate pose estimation with a mobile phone, making a self-contained and highly portable RNA possible. In this article, we apply CIP-VMobile to assistive navigation for the blind. The embodiment is a new RNA (described in Section III) for assisted wayfinding in large-scale indoor spaces. In this article, the RNA is used as a platform to evaluate the CIP-VMobile method's performance in pose estimation.

III. Robocane Prototype and Software System

The RNA is depicted in Fig. 1. It uses an iPhone 7 as the sensing and computing platforms. The phone's rear camera and IMU (iNEMO inertial module LSM6DSM) are used as the imaging and motion sensors to form a visual–inertial system. The camera produces 640 $\times$ 480 images at a rate of 30 Hz and the IMU provides inertial data (three-axis acceleration and three-axis rotation) at 100 Hz. The phone is installed on a white cane by using a 3-D printed housing. The phone is connected to a Bluno Nano board (via Bluetooth) that controls the active rolling tip (ART) at the front-end of the cane by using the RNA control circuit (RCC) and the dc motor drive (Faulhaber drive MCBL3002S). The ART is used to convey the desired direction of travel (DDT) to the user by steering the cane into the DDT [28]. The ART consists of a rolling tip, an electromagnetic clutch, and a gearhead-motor-encoder assembly. The Bluno Nano controls the clutch via its GPIO port and communicates with the motor drive via its RS232 port. To indicate the DDT, the clutch is engaged to allow the motor to

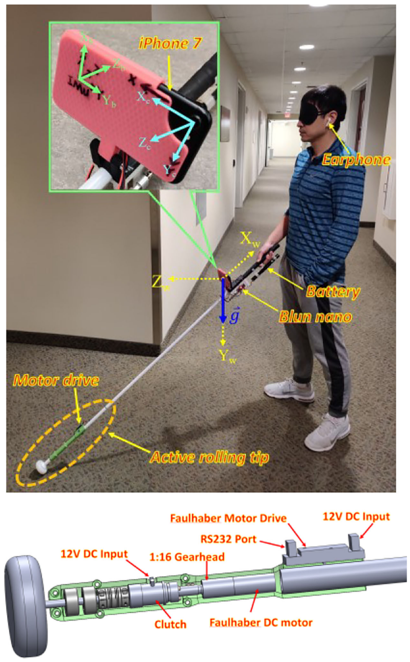

Fig. 1. Top: RNA prototype with the coordinate systems. Bottom: Solidworks drawing of the ART. The coordinate systems for IMU, camera, and the world are denoted $\{B\}$ (or $X_bY_bZ_b$), $\{C\}$(or $X_cY_cZ_c$), and $\{W\}$(or $X_wY_wZ_w$), respectively. The initial $\{B\}$ at the beginning of a navigation task is taken as the world coordinate system $\{W\}$ after performing a rotation around Y_b to align X_b with the gravity vector $\vec{g}$. In this article, super scripts b and c are used to indicate a variable in $\{B\}$ and $\{C\}$, respectively. The transformation matrix between $\{B\}$ and $\{C\}$ is precalibrated and denoted $\mathbf{T}_c^b = [\mathbf{R}_c^b;\ \mathbf{t}_c^b]$, where $\mathbf{R}_c^b$ is rotation and $\mathbf{t}_c^b$ is translation.

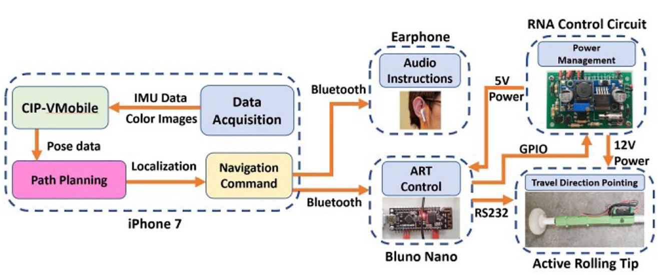

Fig. 2. Software pipeline of the RNA.

drive the rolling tip and steers the cane into the targeted heading direction. When the clutch is disengaged, the ART turns itself into a regular rolling tip of a standard white cane. A user intent detection interface is devised to automatically set the ART in an appropriate mode. The mechanism is omitted for conciseness. The RCC board consists of circuits for controlling the clutch and two mini vibration motors whose vibration patterns will be used to alert the blind user of certain events. The board also performs voltage conversion and provides power supply to the onboard electronics. The RNA prototype weighs 900 grams. The weight can be reduced by using a lighter battery and motor assembly.

The pipeline of the RNA software system is shown in Fig. 2. The CIP-VMobile module acquires imaging and inertial data from the phone's camera and IMU and estimates the 6-DOF device pose, based on which the path planning module determines the RNA's location and heading on a prestored 2-D floorplan and plans the shortest path to the destination. The point-of-interest graph [29] is used to find the path and generate a navigational command. The navigational command, such as "turn left" and "keep going," is conveyed to the blind traveler by using the Bluetooth earphone. The desired turn angle (the difference between the current heading and next heading) is sent to the Bluno Nano to control the ART and guide the user to move along the planned path toward the destination.

IV. CIP-VMobile

In our earlier work [30], we have demonstrated that VINS-Mono performs better than the other state-of-the-art VIO methods in indoor navigation. Since VINS-Mobile [2] is the mobile version of VINS-Mono, it is natural for us to develop the CIP-VMobile method based on the framework of VINS-Mobile. In this article, we extend VINS-Mobile by adding CIP into the state vector for online estimation. Moreover, we build a linear model relating the CIP to the acceleration data and use it to constrain the solution to the optimization problem. CIP-VMobile has two major modules: front-end feature tracking and back-end state estimation. The front-end module is the same as that of VINS-Mobile, while the back-end module is different, and it is described in the following.

The state estimation problem can be illustrated by a factor graph model [31]. A factor graph 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 encodes a measurement model defined by a probabilistic distribution function (PDF) of these variables. We denote the set of variables up to m nodes by Θ_m. The graph is denoted by $G_m = (\mathcal{F}_m,\ \Theta_m, \mathcal{E}_m)$, where the variable node $\theta_i \in \Theta_m$ represents an unknown random variable to be estimated; factor node $f_i \in \mathcal{F}_m$ represents the variable's PDF; and edges $\varepsilon_{ij} \in \mathcal{E}_m$ indicates the connection/relation between nodes f_i and $\boldsymbol{\theta_j}$. The joint PDF of the graph G_m is factorized by

$$\mathrm{pdf}\,(G_m) = \prod_i f_i\,(\theta_i). \tag{1}$$

Assuming a Gaussian measurement model, f_i can be computed by

$$f_i\,(\theta_i) \propto \exp\left(-\frac{1}{2}\|\mathbf{r}_i\|^2\right) = \exp\left(-\frac{1}{2}\mathbf{e}_i^T\boldsymbol{\Sigma}_i^{-1}\mathbf{e}_i\right). \tag{2}$$

Here, $\|\mathbf{r}\|_i{}^2$ is the squared Mahalanobis distance; $\mathbf{e}_i = h_i(\theta_i) - \mathbf{z}_i$ is the residual vector, representing the difference between the estimated measurement $h_i(\theta_i)$ and the actual measurement $\mathbf{z}_i$; and $\boldsymbol{\Sigma}_i$ is the covariance matrix. $\boldsymbol{r_i} = \sqrt[2]{\boldsymbol{\Sigma}_i^{-1}}\mathbf{e}_i$ is called the normalized residual vector. The measurement model f_i is a constraint for the estimation of θ_i. The solution to the state estimation problem is to find the optimal value Θ_m^* that maximizes $\mathrm{pdf}(G_m)$

$$\Theta_m^* = \underset{\Theta_m}{\operatorname{argmax}} \prod_i f_i\,(\theta_i). \tag{3}$$

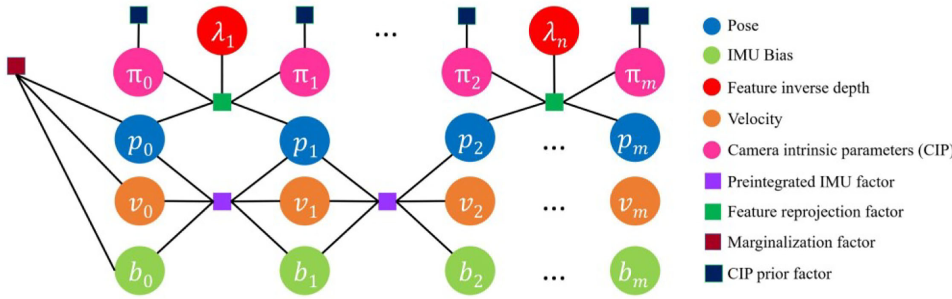

Fig. 3. Example of factor graph structure: circle and square stand for variable node and factor node, respectively.

This is equivalent to the nonlinear least-square (LS) solution

$$\Theta_m^* = \underset{\Theta_m}{\operatorname{argmax}} \left(-\sum \log f(\Theta_m) \right) = \underset{\Theta_m}{\operatorname{argmin}} \left(\sum_{i=1}^{m} \|\mathbf{r}_i\|^2 \right). \tag{4}$$

CIP-VMobile uses keyframes to estimate the poses. A sliding window with m keyframes ($m = 10$ in this article) is used to keep the computational cost low. At the time when the kth keyframe is captured, the state vector of the VIO problem is defined as $\Theta_k = \{\mathbf{x}_{b_k}^w, \mathbf{x}_{b_{k-1}}^w, \ldots, \mathbf{x}_{b_{k-m}}^w, \zeta_{\boldsymbol{k}}, \zeta_{k-1}, \ldots, \zeta_{\boldsymbol{k-m}}, \lambda_1, \lambda_2, \ldots, \lambda_n\}$. Here, $\mathbf{x}_{b_k}^w = \{\mathbf{t}_{b_k}^w, \mathbf{v}_{b_k}^w, \mathbf{q}_{b_k}^w, \mathbf{b}_a, \mathbf{b}_g\}$ is the IMU's motion state consisting of the translation, velocity, rotation, accelerometer bias, and gyroscope bias, $\zeta_{\boldsymbol{k}} = \{f_x^k, f_y^k, o_x^k, o_y^k\}$ is the camera intrinsic vector including focal length and principal point for the kth keyframe, and $\lambda_i (i = 1, \ldots, n)$ denotes the estimated inverse depth of the ith visual feature. $\mathbf{R}_{b_k}^w$ is the rotation matrix corresponding to quaternion $\mathbf{q}_{b_k}^w$. $\mathbf{x}_{b_k}^w$ consists of three variable nodes $\Psi_{\boldsymbol{k}} = \{\mathbf{t}_{b_k}^w, \mathbf{q}_{b_k}^w\}$, $\boldsymbol{V_k} = \{\mathbf{v}_{b_k}^w\}$, and $\boldsymbol{B_k} = \{\mathbf{b}_a, \mathbf{b}_g\}$ in the factor graph. The measurement constraints between the three nodes and other variable nodes are encoded in the connected factor nodes. One example factor graph is shown in Fig. 3. The graph has four types of factor nodes: preintegrated IMU factor, feature reprojection (FR), marginalization factor, and CIP prior factor. The LS solution of the factor graph is given by

$$\Theta_m^* = \underset{\Theta_m}{\operatorname{argmin}} \left(\|\mathbf{r}_0\|^2 + \sum_i \left\| {}^{\mathbf{IMU}}\mathbf{r}_{\mathbf{i,i+1}} \right\|^2 + \sum_{ij} \left\| {}^{\mathbf{FR}}\mathbf{r}_{\mathbf{ij}}{}^2 \right\| + \sum_i \left\| {}^{\mathbf{CIP}}\mathbf{r}_{\mathbf{ij}} \right\|^2 \right) \tag{5}$$

where $\mathbf{r}_0$, ${}^{\mathbf{IMU}}\mathbf{r}_{\mathbf{i,i+1}}$, ${}^{\mathbf{FR}}\mathbf{r}_{\mathbf{ij}}$, and ${}^{\mathbf{CIP}}\mathbf{r}_{\mathbf{ij}}$ are the normalized residual vectors related to the factors of marginalization, preintegrated IMU, FR, and CIP prior, respectively. Subscripts i and j represent the ith and jth keyframes, respectively. In this article, corner points are detected by using the Shi and Tomasi [32] corner detector and used as the visual features. The details of the preintegrated IMU and marginalization factors are referred to [15]. The FR factor and the CIP factor are described as follows.

1) FR Factor: Let the kth feature point that was first observed at the ith keyframe be denoted as $\mathbf{p}_{\boldsymbol{i}} = [u_i, v_i,\ 1]^{\mathrm{T}}$, where (u_i, v_i) are the feature coordinates in C_i. The estimated inverse depth for the feature point is λ. Note that we drop subscript k for simplicity. If the feature point is tracked onto the jth keyframe as $\mathbf{p_j} = [u_j, v_j,\ 1]^{\mathrm{T}}$, the reprojected visual feature from C_i to C_j is computed as $\mathbf{p}'_{\boldsymbol{j}} = [x'_j, y'_j, z'_j]^T = \mathbf{R}_{c_i}^{c_j} \frac{\pi_i^{-1}(\mathbf{p}_i)}{\lambda} + \mathbf{t}_{c_i}^{c_j}$, where $\mathbf{R}_{c_i}^{c_j} = (\mathbf{R}_{b_j}^w \mathbf{R}_c^b)^T \mathbf{R}_{b_i}^w \mathbf{R}_c^b$, $\mathbf{t}_{c_i}^{c_j} = (\mathbf{R}_{b_j}^w \mathbf{R}_c^b)^T (\mathbf{R}_{b_i}^w \mathbf{t}_{\mathrm{c}}^{\mathrm{b}} + \mathbf{t}_{b_i}^w - \mathbf{R}_{b_j}^w \mathbf{t}_{\mathrm{c}}^{\mathrm{b}} - \mathbf{t}_{b_j}^w)$, and $\pi_i^{-1}(\mathbf{p}_i)$ is the inverse camera perspective transformation of $\mathbf{p}_i$ that is given by $n_i(\mathbf{p}_{\boldsymbol{i}}) = \pi_i^{-1}(\mathbf{p}_{\boldsymbol{i}}) = [(u_i - o_x^i)/f_x^i\ (v_i - o_y^i)/f_y^i\ 1]^T$. Then, the residual vector $\mathbf{e_{ij}}$ of the FR factor is defined by $\mathbf{e_{ij}} = ((\mathbf{p}'_{\boldsymbol{j}}/z'_j) - \pi_j^{-1}(\mathbf{p_j}))_2$, where $(\cdot)_2$ represents the first two entries of the vector. The covariance matrix is defined as a diagonal matrix $\mathbf{\Sigma_{ij}} = \mathrm{diag}[\sigma_\gamma^2/(f_x^j)^2,\ \ \sigma_\gamma^2/(f_y^j)^2]$. ($\sigma_\gamma = 1.5$ pixels in this article). The Jacobians of $\mathbf{e_{ij}}$ with respect to pose variables $\Psi_{\boldsymbol{i}}$, $\Psi_{\boldsymbol{j}}$ and inverse depth λ are defined in [15], and the Jacobians with respect to CIP ζ_i and ζ_i are given by

$$\frac{\partial \mathbf{e_{ij}}}{\partial \zeta_{\boldsymbol{i}}} = \frac{\partial \mathbf{e_{ij}}}{\partial \mathbf{p}'_{\boldsymbol{j}}} \cdot \frac{\partial \mathbf{p}'_{\boldsymbol{j}}}{\partial n_i} \cdot \frac{\partial n_i}{\partial \zeta_i} \tag{6}$$

and

$$\frac{\partial \mathbf{e_{ij}}}{\partial \zeta_{\boldsymbol{j}}} = -\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \frac{o_x^j - u_j}{(f_x^j)^2} & -\frac{1}{f_x^j} & 0 \\ \frac{o_y^j - v_j}{(o_y^j)^2} & 0 & -\frac{1}{f_y^j} \\ 0 & 0 & 0 \end{bmatrix} \tag{7}$$

where

$$\frac{\partial \mathbf{e_{ij}}}{\partial \mathbf{p}'_{\boldsymbol{j}}} = \begin{bmatrix} \frac{1}{z'_j}, 0, \frac{-x'_j}{(z'_j)^2} \\ 0, \frac{1}{z'_j}, \frac{-y_j{}'}{(z'_j)^2} \end{bmatrix},$$

$$\frac{\partial \mathbf{p}'_{\boldsymbol{j}}}{\partial n_i} = \mathbf{R}_{c_i}^{c_j}/\lambda,$$

and

$$\frac{\partial n_i}{\partial \zeta_i} = \begin{bmatrix} \frac{o_x^i - u_i}{(f_x^i)^2} & -\frac{1}{f_x^i} & 0 \\ \frac{o_y^i - v_i}{(f_y^i)^2} & 0 & -\frac{1}{f_y^i} \\ 0 & 0 & 0 \end{bmatrix}.$$

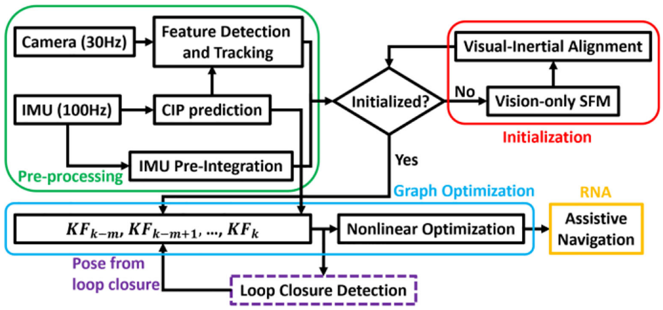

Fig. 4. Diagram of the proposed CIP-VMobile method.

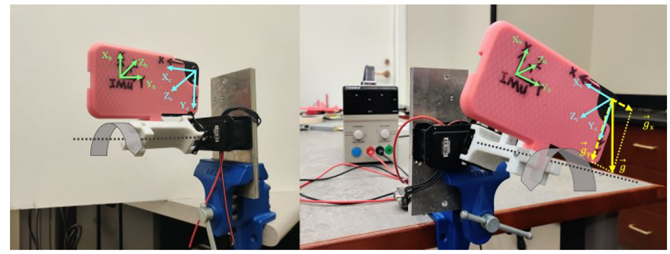

Fig. 5. Data collection setup: By changing the mounting location, the actuator turned the iPhone around its *x*/*y*/*z*-axis.

2) CIP Prior Factor: For a VCM-based OIS smartphone camera [18], [19], the lens is connected to mechanical support that is anchored to the chassis by springs. The springs allow for translation and/or rotation of the lens, resulting in varying CIP. According to Hooke's law, the extension/compression of the springs is linearly proportional to the exerted force. As the force is linearly related to the acceleration that can be measured by the accelerometers, we use a linear model to estimate the CIP based on accelerometer data. As a result, the CIP for the kth keyframe is given by

$$\boldsymbol{\eta_k} = L\left(\mathbf{a_k}\right) = (\alpha),\ \mathbf{a_k} + \boldsymbol{\beta} \tag{8}$$

where $\mathbf{a_k}$ is the accelerometer reading, $\cdot,\cdot$ stands for element-wise multiplication, $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are the coefficients whose values are determined by experiment (described in Section V-A) The residual vector of the CIP prior factor is then given by $\mathbf{e}_k = \widehat{\eta}_{\boldsymbol{k}} - \eta_{\boldsymbol{k}}$. The computation of the covariance matrix $\boldsymbol{\Sigma_k}$ is given in Section V-A. The Jacobian matrix is a 3-D identity matrix.

The proposed CIP-VMobile method is depicted in Fig. 4. Just like VINS-Mobile, CIP-VMobile can detect a loop closure by matching the visual features of the current keyframe with the visual features of a template image frame in a prestored imaging database. Once a loop closure is detected, the pose for the keyframe will be fixed to the value computed by the loop closure detection method. This way, the accumulative pose error of the current keyframe is reset and the pose errors for the other keyframes in the sliding windows may be significantly reduced by the graph optimization process. It is noted that loop closure detection can be computationally expensive as it requires computing the descriptors of the current keyframe's visual features and comparing them with those of each of the template images in the database. In this article, we compare CIP-VMobile with VINS-Mobile without loop closure detection.

A theoretical analysis on the convergence of the basic pose graph optimization problem is given in [33], where an estimate of the convergence domain and the region within which the minimum is unique is provided. One can view CIP-VMobile as an extension of the basic pose graph optimization problem, i.e., VINS-Mono/VINS-Mobile adds velocity-nodes and bias-nodes as well as the related edges to the graph while CIP-VMobile adds CIP-nodes and the related edges to the graph. As CIP is linearly related to acceleration, the CIP-nodes can be viewed as acceleration-nodes. A CIP-node is much less complicated than a velocity-node because the acceleration data comes directly from the IMU measurement while the velocity value is the integral of the acceleration data and it is also related to the IMU pose. Therefore, adding CIP-nodes causes less complication to the convergence issue than adding velocity-nodes to the graph. It has been proved in [33] that adding edges to the graph does not reduce the convergence radius. Therefore, it is reasonable to believe that CIP-VMobile will behave well in convergence if VINS-Mono/VINS-Mobile behaves well.

V. Experiments

We first characterized the camera and IMU of an iPhone 7 that is used for this article by experiments. Based on the experimental data, we derived a linear model that relates the CIP to the IMU-measured acceleration of the camera. Also, the statistical properties (noise density and bias random walk) of the IMU were obtained by the characterization study. Then, we carried out both simulations and experiments to compare the pose estimation performance of CIP-VMobile with that of VINS-Mobile. Finally, we further evaluated the proposed method's performance in assisted wayfinding application in a real-world environment by using the RNA as an experimental platform.

A. Calibration

As shown in Fig. 5, we mounted the iPhone 7 on a Dynamixel EX-106 servo actuator via a 3-D printed bracket, which allowed us to rotate the phone around its $x/y/z$ axis from 0° to 180° (with a step-size of 3°). At each step, the three-axis accelerometer reading $\mathbf{a} = [a_x, a_y, a_z]^{\boldsymbol{T}}$ and the camera's CIP were obtained and paired. A total of 100 data-pairs were acquired for each step. The CIP values were determined by camera calibration [34]. The acquired data were plotted in Fig. 6, which clearly indicates that the CIP values are linearly related to the acceleration. Based on our observation, the focal lengths along x-axis and y-axis are almost the same. Therefore, we ignore the difference and let $f_c = f_x = f_y$. The coefficients ($\boldsymbol{\alpha}$, $\boldsymbol{\beta}$) of the linear CIP-acceleration (CIPA) model denoted $L(\mathbf{a})$ can be obtained by

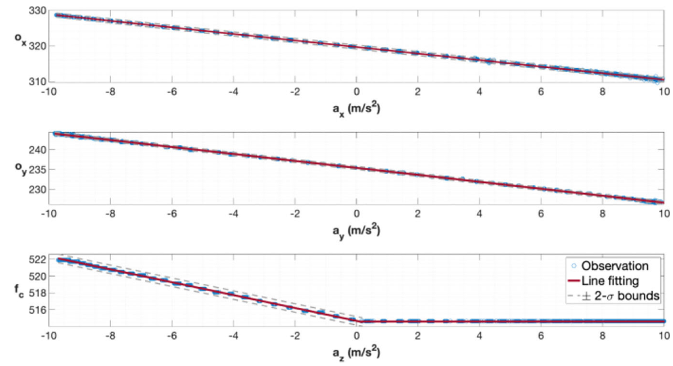

Fig. 6. Parametric fitting for the CIP's linear model. $o_x/o_y/f_c$ unit: pixel.

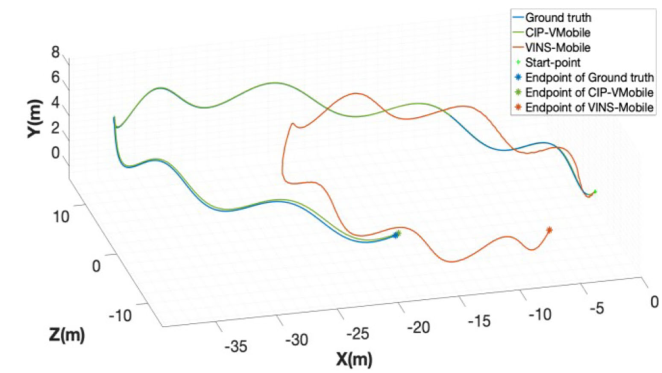

Fig. 7. Performance Comparison using simulated data.

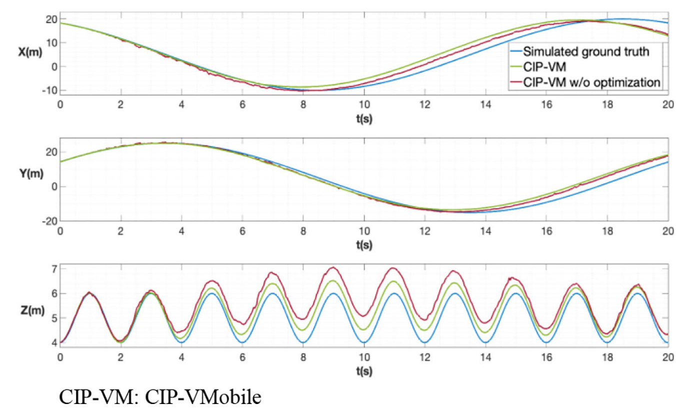

Fig. 8. Estimated trajectories: The *X*-, *Y*-, and *Z*-coordinates of the estimated trajectory are more accurate if the CIP optimization of the method is enabled. CIP-VM: CIP-VMobile.

TABLE I
IMU NOISE AND RANDOM WALK BIAS

	Noise density	Random walk
accelerometer	$5.59\times10^{-3}\frac{m}{s^2}\frac{1}{\sqrt{Hz}}$	$3.19\times10^{-4}\frac{m}{s^3}\frac{1}{\sqrt{Hz}}$
gyroscope	$9.37\times10^{-4}\frac{rad}{s}\frac{1}{\sqrt{Hz}}$	$2.0\times10^{-6}\frac{rad}{s^2}\frac{1}{\sqrt{Hz}}$

line-fitting. The resulted model is given by

$$f_c = \begin{cases} -0.6806 * a_z + 514.7183, & a_z < 0 \\ 514.7183, & a_z \geq 0 \end{cases} \tag{9}$$

$$o_x = -0.8549 \ * \ a_x + 319.6476 \tag{10}$$

$$o_y = -0.8817 \ * \ a_y + 235.2553. \tag{11}$$

The covariance of the CIP prior factor $\mathbf{\Sigma}$ is a diagonal matrix defined as $\mathbf{\Sigma} = \mathrm{diag}(\frac{1}{n-1}\sum_{i=1}^{n-1}\langle\delta_i,\delta_i\rangle)$, where n is the total number of the data points and δ_i is the fitting error for the ith data point. The line-fitting result produces $\mathbf{\Sigma} = \mathrm{diag}(0.08585, 0.0392, 0.064)$. The small variances indicate an accurate CIPA mode. The use of the CIPA model in the proposed method constrains the adjustment of the CIP values in the vicinity of the model-computed CIP values, making the method converge faster.

We employed the Allan variance analysis to estimate the statistical properties of the iPhone's IMU. The results are tabulated in Table I.

B. Simulation Results

We employed the open-source code [35] to generate simulated visual–inertial data for a simulated run by moving the iPhone in a sinuous trajectory (about 120 m). The IMU's statistical properties and the values of the CIP are generated based on the calibration results in Section V-A. Projection of visual features were made by using a virtual camera with the corresponding CIP. The standard deviation of a visual feature measurement σ_γ is set to 1.5 pixels. We ran CIP-VMobile and VINS-Mobile on the simulated data. The estimated trajectories are compared against the ground truth in Fig. 7. Clearly, CIP-VMobile demonstrates a superior pose estimation performance over VINS-Mobile: its trajectory closely tracks the ground truth while that of VINS-Mobile diverges quickly from the ground truth.

To demonstrate that the graph optimization process can effectively refine the CIPA-computed CIP, we purposefully degraded the CIPA model by increasing the noise to 3 pixels (about ten times). This significantly decreased the accuracy of the CIP data. We first ran CIP-VMobile on the simulated data and repeated the simulation by disabling the method's CIP optimization/refinement function (i.e., simply used the CIP computed by the CIPA model). Fig. 8 compares the trajectories generated by the two simulation runs to the ground truth. The CIP-VMobile-generated trajectory is much closer to the ground truth. This was due to the fact that the CIP optimization procedure of CIP-VMobile refined the CIP (see Fig. 9) for each keyframe during the graph optimization process, resulting in a more accurate camera model and, thus, more accurate pose estimation result.

Finally, our simulation results also showed that CIP-VMobile resulted in larger errors in tracking the intrinsic parameters when the CIPA model was not used to produce the initial CIP estimates. The reason behind this was that without using the model the method started with some bad CIP values, which increase the chance for the method to be terminated prematurely.

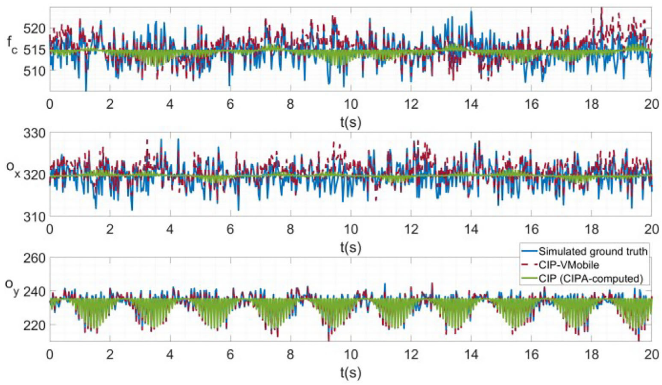

Fig. 9. CIP refinement by CIP-VMobile: The CIP optimization procedure substantially refined the CIPA-computed parameters and make them much closer to the ground truth values.

Fig. 10. Snapshot (panoramic view) of the lab environment for experiment.

TABLE II
EPENs (%) FOR EXPERIMENTS WITH A HANDHELD I PHONE

Experiment	1	2	3	4	5	6	7	8	9	10	Avg
CIP-VM	0.89	1.34	0.89	1.11	1.30	0.95	0.53	0.73	1.95	0.89	1.06
VINS-M	1.05	3.41	1.37	1.26	1.90	1.29	1.05	0.84	2.42	1.34	1.62

CIP-VM: CIP-VMobile, VINS-M: VINS-Mobile.

The use of the CIPA model can effectively alleviate the problem of premature termination problem and reduce CIP estimation error. In addition, it can reduce the iteration number of the graph optimization procedure and speed up the computation.

C. Experimental Results With Hand-Held iPhone

We carried out ten experiments in our laboratory (see Fig. 10) by hand-holding the iPhone 7 and walking in a looped trajectory (i.e., the starting point and the endpoint is the same) at a normal walking speed ($\sim$0.6 m/s). The length of the trajectory for each experiment is about 20 m. At the beginning of each experiment, we rotated the iPhone significantly to excite the visual–inertial system to allow for a good system initialization. The endpoint position error norm (EPEN) in a percentage of the path-length is used as the metric of pose estimation accuracy. It can be seen that CIP-VMobile consistently outperformed VINS-Mobile. The results of the experiments are tabulated in Table II. On average, CIP-VMobile reduced the EPEN error by 34.6%. Fig. 11 compares the trajectories estimated by the two methods for experiment 1 against the ground truth trajectory. It can be seen that the CIP-VMobile generated trajectory tracks the ground truth better than that of VINS-Mobile. The root mean squares of the point-to-point position errors of CIP-VMobile and VINS-Mobile are 0.109 m and 0.148 m, respectively, indicating that the former has an overall better pose estimation accuracy.

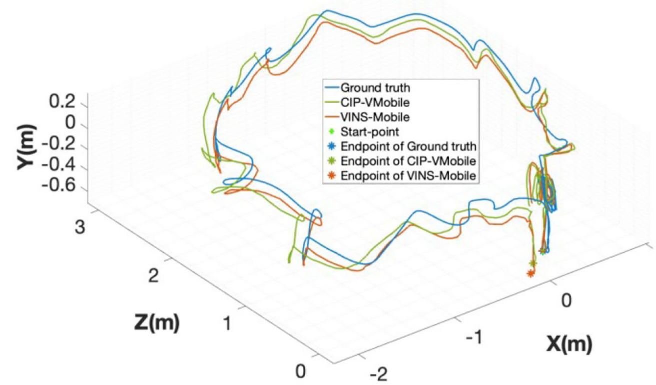

Fig. 11. Comparison of trajectories for experiment 1.

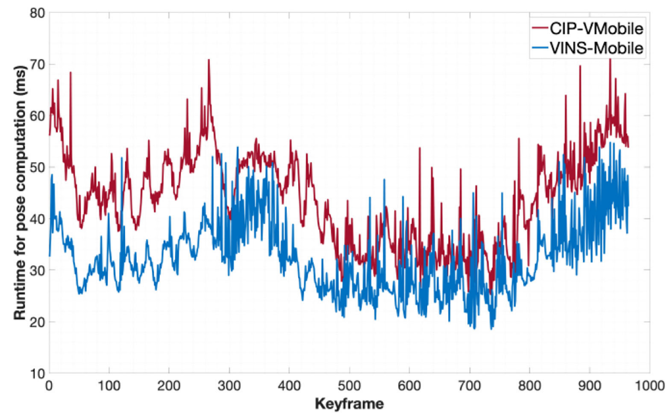

Fig. 12. Comparison of the runtimes for CIP-VMobile and VINS-Mobile.

D. Runtime Analysis

We ran CIP-VMobile and VINS-Mobile on a laptop computer (Intel Core i7-8550U, 16 GB memory, Ubuntu 16.04 LTS 64-bit OS). The results showed that both methods could compute pose in real time. Taking the first experiment in Table II for instance, the runtimes of the two methods are compared in Fig. 12. It can be seen that the runtime for CIP-VMobile to compute a pose is larger than that of VINS-Mobile. On average, the runtimes for CIP-VMobile and VINS-Mobile are 44.0 ms and 32.4 ms, respectively. With respect to the implementation with a smartphone, VINS-Mobile achieved a real-time pose computation performance ($\sim$23 per pose computation) on an iPhone 7 [2]. Therefore, it is anticipated that CIP-VMobile can run in real time on the same smartphone platform. It is noted that VINS-Mobile uses simplified linear algebra libraries to save computational cost when implemented with an iPhone 7, resulting in a faster speed than our laptop implementation.

TABLE III
EPENs (%) FOR EXPERIMENTS WITH THE RNA PROTOTYPE

Experiment	1	2	3	4	Avg
CIP-VMobile	1.07	1.57	1.21	1.59	1.46
VINS-Mobile	1.31	1.69	1.38	1.85	1.64

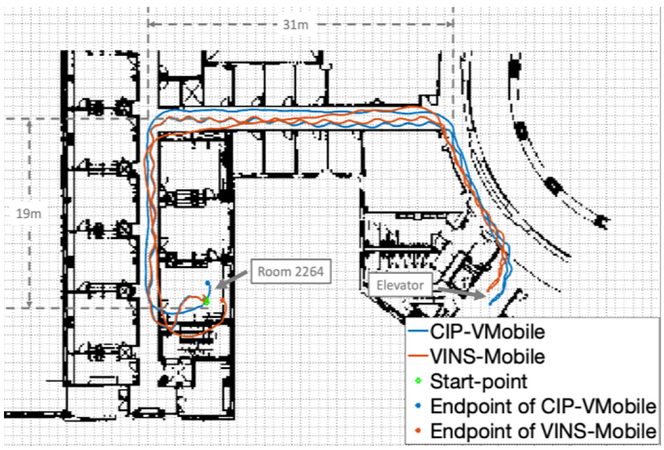

Fig. 13. Comparison of trajectories for experiment 3.

E. Experimental Results With the RNA

To validate the CIP-VMobile method in the real world, we carried out experiments with the RNA prototype in the hallways of the East Engineering Building on campus. In each experiment, the RNA user walked from Room 2264 to the elevator and returned to the starting point. He swung the RNA when walking to mimic the way a blind person uses a traditional white cane. The results are tabulated in Table III. It can be seen that CIP-VMobile achieved a smaller EPEN than VINS-Mobile in all of the four experiments. On average, CIP-VMobile reduces the EPEN by $\sim$11%. Fig. 13 compares the trajectories estimated by the two methods for experiment 3, from which it can be observed that CIP-VMobile resulted in a more accurate trajectory than VINS-Mobile. This is evidenced by that the trajectory of VINS-Mobile collides with the walls at several locations, but no collision occurs along the CIP-VMobile estimated trajectory.

VI. Conclusion

We have presented a new VIO method, called CIP-VMobile, for pose estimation of a modern smartphone with a camera that uses OIS to reduce image blurs. The proposed method uses a linear model to estimate the camera's CIP based on the accelerometer data and refines the estimated CIP in the graph optimization process. The model and the graph-based refinement are complimentary one another and they work together to ensure more accurate CIP and pose estimation: an accurate and precise model can speed up the CIP estimation process while the graph-based refinement can kick in as needed if the model is less accurate/precise. The combination reduces the proposed method's reliance on a perfect model and enhances its reliability for real-world applications. Simulation results and experimental results with an iPhone 7 demonstrate that the proposed method can substantially improve the pose estimation performance of the state-of-the-art mobile-phone-based VIO method. Based on CIP-VMobile, we designed and fabricated an RNA prototype for assisted navigation in large indoor spaces. Experimental results with the RNA validate the method's efficacy in pose estimation for assisted wayfinding. The proposed method can be applied to any mobile devices that use an OIS camera for device pose estimation.

In the future, we will further develop the software system for the RNA and carry out thorough experiments with blind human subjects to test the RNA's reliability in real-world wayfinding scenarios and investigate the user acceptance of the RNA.

Acknowledgment

The content is solely the responsibility of the authors and does not necessarily represent the official views of the funding agencies.

References

[1] K. Georg and D. Murray, "Parallel tracking and mapping on a camera phone," in *Proc. IEEE Int. Symp. Mixed Augmented Reality*, 2009, pp. 83–86.
[2] P. Li, T. Qin, B. Hu, F. Zhu, and S. Shen, "Monocular visual-inertial state estimation for mobile augmented reality," in *Proc. IEEE Int. Symp. Mixed Augmented Reality*, 2017, pp. 11–21.
[3] S. Thomas, J. Engel, and D. Cremers, "Semi-dense visual odometry for AR on a smartphone," in *Proc. IEEE Int. Symp. Mixed Augmented Reality*, 2014, pp. 145–150.
[4] W. Fan *et al.*, "Real-time motion tracking for mobile augmented/virtual reality using adaptive visual-inertial fusion," *Sensors*, vol. 17, no. 5, 2017, Art. no. 1037.
[5] J. Piao and S. Kim, "Adaptive monocular visual–inertial SLAM for real-time augmented reality applications in mobile devices," *Sensors*, vol. 17, no. 11, 2017, Art. no. 2567.
[6] L. Porzi, E. Ricci, T. A. Ciarfuglia, and M. Zanin, "Visual-inertial tracking on android for augmented reality applications," in *Proc. IEEE Workshop Environ. Energy Struct. Monitoring Syst.*, 2012, pp. 35–41.
[7] W. Winterhalter, F. Fleckenstein, B. Steder, L. Spinello, and W. Burgard, "Accurate indoor localization for RGB-D smartphones and tablets given 2D floor plans," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2015.
[8] R. Faragher and R. Harle, "SmartSLAM-an efficient smart-phone indoor positioning system exploiting machine learning and opportunistic sensing," in *Proc. Int. Tech. Meeting Satell. Division Inst. Navigat.*, 2013, pp. 1006–1019.
[9] M. Li, B. H. Kim, and A. I. Mourikis, "Real-time motion tracking on a cellphone using inertial sensing and a rolling-shutter camera," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2013, pp. 4712–4719.
[10] R. Tapu, B. Mocanu, A. Bursuc, and T. Zaharia, "A smartphone-based obstacle detection and classification system for assisting visually impaired people," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, 2013, pp. 444–451.
[11] M. Li H. Yu, X. Zheng, and A. I. Mourikis, "High-fidelity sensor modeling and self-calibration in vision-aided inertial navigation," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2014, pp. 409–416.
[12] S. Tso and S. Jan, "Observability analysis and performance evaluation of EKF-based visual-inertial odometry with online intrinsic camera parameter calibration," *IEEE Sens. J.*, vol. 19, no. 7, pp. 2695–2703, Apr. 2019.
[13] M. Klingensmith, I. Dryanovski, S. Srinivasa, and J. Xiao, "Chisel: Real-time dense reconstruction on a mobile device using spatially hashed truncated signed distance fields," *Robot., Sci. Syst.*, 2015.
[14] A. I. Mourikis and S. I. Roumeliotis, "A multi-state constraint Kalman filter for vision-aided inertial navigation," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2007, pp. 3565–3572.
[15] T. Qin, P. Li, and S. Shen, "VINS-Mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Trans. Robot.*, vol. 34, no. 4, pp. 1004–1020, Aug. 2018.

[16] S. Leutenegger, P. Furgale, V. Rabaud, M. Chli, K. Konolige, and R. Siegwart, “Keyframe-based visual-inertial slam using nonlinear optimization,” *Robot., Sci. Syst.*, 2013.
[17] X. Liu *et al.*, MEMS-based optical image stabilization, U.S. Patent Appl. 8 855 476, Oct. 2014.
[18] R. J. Topliss *et al.*, Voice coil motor optical image stabilization wires, U.S. Patent 10 063 752, Aug. 2018.
[19] R. J. Topliss, VCM OIS actuator module, U. S. Patent 9 134 503, Sept. 2015.
[20] H. Jo, H. Min Cho, S. Jo, and E. Kim, “Efficient grid-based rao–blackwellized particle filter slam with interparticle map sharing,” *IEEE/ASME Trans. Mechatron.*, vol. 23, no. 2, pp. 714–724, Apr. 2018.
[21] J. Du, W. Sheng, and M. Liu, “A human–robot collaborative system for robust three-dimensional mapping,” *IEEE/ASME Trans. Mechatron.*, vol. 23, no. 5, pp. 2358–2368, Oct. 2018.
[22] Q. Sun, J. Yuan, X. Zhang, and F. Sun, “RGB-D slam in indoor environments with sting-based plane feature extraction,” *IEEE/ASME Trans. Mechatron.*, vol. 23, no. 3, pp. 1071–1082, Jun. 2018.
[23] J. Tardif, Y. Pavlidis, and K. Daniilidis, “Monocular visual odometry in urban environments using an omnidirectional camera,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2008, pp. 2531–2538.
[24] V. Pradeep, G. Medioni, and J. Weiland, “Robot vision for the visually impaired,” in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Workshops*, 2010, pp. 15–22.
[25] Y. H. Lee and G. Medioni, “RGB-D camera based navigation for the visually impaired,” *Robot., Sci. Systems*, 2011.
[26] H. Zhang and C. Ye, “An indoor navigation aid for the visually impaired,” in *Proc. IEEE Int. Conf. Robot. Biomimetics*, 2016, pp. 467–472.
[27] B. Li *et al.*, “Vision-based mobile indoor assistive navigation aid for blind people,” *IEEE Trans. Mobile Comput.*, vol. 18, no. 3 pp. 702–714, Mar. 2019.
[28] H. Zhang and C. Ye, “Human-robot interaction for assisted wayfinding of a robotic navigation aid for the blind,” in *Proc. Int. Conf. Human Syst. Interact.*, 2019, pp. 137–142.
[29] H. Zhang and C. Ye, “An indoor wayfinding system based on geometric features aided graph SLAM for the visually impaired,” *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 25, no. 9, pp. 1592–1604, Sep. 2017.
[30] H. Zhang, L. Jin, H. Zhang, and C. Ye “A comparative analysis of visual-inertial slam for assisted wayfinding of the visually impaired,” in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, 2019, pp. 210–217.
[31] V. Indelman, S. Williams, M. Kaess, and F. Dellaert, “Information fusion in navigation systems via factor graph based incremental smoothing,” *Robot. Auton. Syst.*, vol. 61, no. 8, pp. 721–738, 2013.
[32] J. Shi and C. Tomasi, “Good features to track,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 1994, pp. 593–600.
[33] L, Carlone, “A convergence analysis for pose graph optimization via gauss-newton methods,” in *Proc. IEEE Int. Conf. Robot. Autom.*, 2013, pp. 957–964.
[34] Z. Zhang, “A flexible new technique for camera calibration,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 11, pp. 1330–1334, Nov. 2000.
[35] 2018. [Online]. Available: https://github.com/HeYijia/vio_data_simulation