Hindawi
Journal of Sensors
Volume 2018, Article ID 4801584, 14 pages
https://doi.org/10.1155/2018/4801584

Research Article

An Indoor Navigation System Based on Stereo Camera and Inertial Sensors with Points and Lines

Bo Yang, Xiaosu Xu, Tao Zhang, Yao Li, and Jinwu Tong

Key Laboratory of Micro-Inertial Instrument and Advanced Navigation Technology, Ministry of Education, School of Instrument Science and Engineering, Southeast University, Nanjing 210096, China

Correspondence should be addressed to Xiaosu Xu; xxs@seu.edu.cn

Received 20 December 2017; Revised 25 May 2018; Accepted 4 June 2018; Published 5 July 2018

Academic Editor: Frederick Mailly

An indoor navigation system based on stereo camera and inertial sensors with points and lines is proposed to further improve the accuracy and robustness of the navigation system in complex indoor environments. The point and line features, which are fast extracted by ORB method and line segment detector (LSD) method, are both employed in this system to improve its ability to adapt to complex environments. In addition, two different representations of lines are adopted to improve the efficiency of the system. Besides stereo camera, an inertial measurement unit (IMU) is also used in the system to further improve its accuracy and robustness. An estimator is designed to integrate the camera and IMU measurements in a tightly coupled approach. The experimental results show that the performance of the proposed navigation system is better than the point-only VINS and the vision-only navigation system with points and lines.

1. Introduction

Indoor navigation technique, which has been widely applied in the field of mobile robot (e.g., home service robot) or unmanned aerial vehicle (UAV) system [1, 2], has received considerable attention in the past few years. One major challenge of the indoor navigation system is the unavailability of the global position system (GPS) signal in indoor environment. Therefore, many other sensors have been applied in the system such as sonar [3], odometry [4], light detection and ranging (LiDAR) [2], camera [5], and inertial measurement unit (IMU) [6]. With the recent development in vision-based techniques, cameras used as sensors make the vision-based navigation system more and more attractive [7]. Direct method and feature-based method are two main methods for indoor navigation system. While the direct method estimates the motion by minimizing the photometric error [8], we deal with feature-based method in this paper. For conventional feature-based visual navigation system, feature extraction methods are applied to extract the point features from images collected by the camera. After matching these point features between two frames, the pose of the system can be estimated through ego-motion estimation methods (e.g., perspective-n-point (PnP)) or by minimizing the reprojection error [9, 10]. Note that only point features are used in the conventional feature-based visual navigation system. However, in some low-textured scenarios exemplified by man-made or indoor environments, it is hard to extract enough key points by suitable point feature extraction method. As a result, the performance of conventional feature-based method will decrease, which thus implies that other complementary features need to be explored for further performance improvement. In recent years, line feature has received more and more attention since lines or line segments provide significantly more information than points in encoding the structure of the surrounding environment [11]. They are usually abundant in human-made scenarios, which are characterized by regular structures rich in edges and linear shapes [12]. In addition, recent advances in the line segment detection method have made it possible to extract line features fast and accurately. Therefore, in some low-textured environments, the line features can be used in the feature-based visual navigation system to improve

the navigation accuracy [11, 13]. In this paper, we choose the stereo camera to acquire images of indoor environments; thereafter, points and lines features are all extracted from these images. In addition, the scale information of the features can be estimated by stereo camera because of the small scale of the indoor environments.

Although the camera can be used efficiently in navigation system, it is not robust to motion blur induced by rapid motion [14] and the vision-based navigation system often fails in rapid motion scenarios. In order to make the system more robust, a common solution is fusing camera with inertial sensors, which leads to the visual-inertial navigation system (VINS) [6, 15]. Inertial sensors usually contain three orthogonal gyroscopes and accelerometers to measure the angular velocity and acceleration of the carrier and estimate the carrier's motion in high frequency. However, it suffers from the accumulated error and relatively large measurement uncertainty at slow motion [16]. It can be found that visual and inertial measurements offer complementary properties which make them particularly suitable for fusion [17], since inertial sensors have a relatively low uncertainty at high velocity, whereas cameras can track features very accurately at low velocity and less accurately with increasing velocity [16]. Consequently, VINS works better than vision-only or pure inertial navigation system. Without loss of generality, the VINS can be classified into two methods, filtering-based method and optimization-based method. Although the accuracy of optimization approach is better than the filtering approach, it is computationally expensive. Considering the extra computational cost in line feature extraction, we adopt the filtering approach which usually uses the extended Kalman filter (EKF) to estimate the pose of the system.

In this paper, we present a visual-inertial navigation system with point and line features for indoor environments. The stereo camera is combined with an inertial sensor in a tightly coupled filtering approach to overcome the defect in single sensor. The point and line features are both used in the proposed system, such that this system can perform well in low-textured environments such as indoor environments. In addition, the robustness of this navigation system can be improved by these two schemes. The performance of this system is also tested in an experiment which uses visual-inertial benchmark dataset.

The rest of the paper is organized as follows. The related work is described simply in Section 2. In Section 3, we present the IMU model based on the inertial measurements and the representations of point and line features. The estimator and the implementation of the algorithm are described in Section 4. Experiments are conducted in Section 5 to demonstrate the performance of the proposed system. Finally, conclusions are drawn in Section 6.

2. Related Work

Compared to the simultaneous localization and mapping (SLAM) system, the visual navigation system does not have loop closures, but in a large part, it follows the SLAM paradigm (especially the front-end of the SLAM). For feature-based method, pioneering work has been carried out in [18]. It presents a monocular camera SLAM system which uses sparse key points and EKF to estimate the camera motion. In [19], it points out that the monocular camera suffers from the "scale drift" problem; however, the stereo camera can efficiently overcome this issue. In [20], it presents a stereo visual SLAM system which uses Harris corner detector to extract point features and EKF to estimate the pose. More recently, the ORB-SLAM has been proposed in [21, 22], which supports all monocular, stereo, and RGB-D cameras. In ORB-SLAM, the ORB point features [23] have been used for the first time and the accuracy of the system is very high. However, the systems mentioned above would reduce their performance in low-textured environment with insufficient point features. Therefore, massive efforts are devoted to the line segment detection method and its application in visual navigation and SLAM system.

A linear-time line segment detector has been proposed in [24, 25] with a high detection accuracy. In [13], a line-based monocular 2D SLAM system has been also presented, which incorporates vertical and horizontal lines to estimate the motion with EKF. In addition, another line-based system has been proposed in [11] to improve the efficiency of the system by adopting two different representations of a line segment. However, the line feature only lends itself to the structured environment, which implies that the system performance is likely to be degraded in complex environment by employing the single line feature. Taking into account the complementarity between point and line features, a combination of both features has beneficial effects on the robustness and accuracy of the navigation system. Recently, visual navigation systems with points and line segments have been proposed in many works [26–28], which reveals superior performance in a wide variety of real-time scenarios. However, the visual-only system has its own limitation and the VINS is developed to overcome it.

The filtering-based VINS is particularly relevant to our work. A vision-aided inertial navigation system based on a multistate constraint Kalman filter has been proposed in [6, 15], presenting a novel system model which does not contain the feature position in state vector. Therefore, the computational overhead of the method is low. However, this system relies on large storage capacity, which makes it unavailable in some applications. A consistent VINS based on EKF has been proposed in [29]. In this paper, the author proposes a new method which improves the consistency of the system based on analyzing the observability of the VINS. An autonomous navigation system combines the stereo camera, and IMU has been proposed in [30], which improves the positioning accuracy by considering both the near and the far point features. However, all these methods mentioned above use point features only. Line-based VINS has been proposed in [31, 32], which use straight lines as features. These systems exhibit superior reconstruction performance against point-based navigation system in line-rich environment. However, in our work, we mainly focus on the VINS combining both point and line features.

3. IMU Model and Feature Representation

In this section, we will describe the IMU model and the camera model which contain the point and line features. To begin with, we define the reference coordinate frames which would be applied in the rest of the paper.

The navigation coordinate frame $\{N\}$ (also can be referred as the world coordinate frame) is chosen as the East-North-Up (ENU) coordinate frame in this paper (the origin of this coordinate frame is fixed in the first frame of the system, and the x-, y-, and z-axes are aligned with the east, north, and up of the system's first frame, resp.). The pose information of the system is all characterized with respect to in this coordinate frame. In addition, the navigation coordinate frame is fixed with the first frame of the system such that the effect of the earth rotation can be ignored (cf. (1) and (2)).

The IMU coordinate frame $\{I\}$ and the camera coordinate frame $\{C\}$ are fixed with the IMU and the stereo camera, respectively. Their origins are located at the center of the IMU and the optical center of the camera. The x-, y-, and z-axes of the IMU coordinate frame are aligned with the right, front, and up of the IMU, respectively. The z-axis of camera coordinate frame points aligns with the optical axis. There are two camera coordinate frames, which are denoted in terms of left camera coordinate frame C_{L} and the right camera coordinate frame C_{R}. The transition matrix between them is obtained though camera calibration, and the C_{L} is chosen as the reference camera coordinate frame $\{C\}$.

In addition, for the sake of simplicity, in Sections 3 and 4, we assume that the IMU coordinate frame $\{I\}$ and the camera coordinate frame $\{C\}$ (left camera coordinate frame) are coincident (in practice, the transition matrix between these two coordinate frames can be extrinsically calibrated following the method proposed in [33]). To avoid confusion, in Sections 3 and 4, we use the other notation $\{B\}$ (body coordinate frame) to represent both coordinate frames (in practice, one of the $\{I\}$ and $\{C\}$ can be selected as the body coordinate frame).

3.1. IMU Model. The angular velocity $\omega_{\mathrm{m}}^{\mathrm{b}}$ and the specific force $f_{\mathrm{m}}^{\mathrm{b}}$ of the system in the body coordinate frame can be measured by the IMU. These measurements include the angular velocity and the acceleration information with noise [34]:

$$\omega_{\mathrm{m}}^{\mathrm{b}} = \omega^{\mathrm{b}} + \varepsilon^{\mathrm{b}} + w_{\mathrm{g}}^{\mathrm{b}}, \tag{1}$$

$$f_{\mathrm{m}}^{\mathrm{b}} = \mathbf{C}_{\mathrm{n}}^{\mathrm{b}}(a^{\mathrm{n}} - \mathbf{g}) + \nabla^{\mathrm{b}} + w_{\mathrm{a}}^{\mathrm{b}}, \tag{2}$$

where ω^{b} denotes the true angular velocity in body coordinate frame, a^{n} denotes the true acceleration in navigation coordinate frame, ε^{b} and ∇^{b} are the constant drifts of the gyroscopes and the accelerometers in body coordinate frame, respectively, $w_{\mathrm{g}}^{\mathrm{b}}$ and $w_{\mathrm{a}}^{\mathrm{b}}$ are the random noises of the gyroscopes and the accelerometers in body coordinate frame, which can be modeled as uncorrelated zero-mean white Gaussian noise, $\mathbf{C}_{\mathrm{n}}^{\mathrm{b}}$ denotes the rotation matrix from the navigation coordinate frame to the body coordinate frame, and $\mathbf{g}$ is the gravity vector.

The estimates of the attitude information $\widehat{Q}$ (described in the form of unit quaternion), the velocity $\hat{v}^{\mathrm{n}}$, and the position $\hat{p}^{\mathrm{n}}$ can be updated by the IMU measurements as follows [34]:

$$\begin{aligned}
\dot{\widehat{Q}} &= \frac{1}{2}M'\left(\omega_{\mathrm{m}}^{\mathrm{b}} - \hat{\varepsilon}^{\mathrm{b}}\right)\widehat{Q}, \\
\dot{\hat{v}}^{\mathrm{n}} &= \widehat{\mathbf{C}}_{\mathrm{b}}^{\mathrm{n}}\left(f_{\mathrm{m}}^{\mathrm{b}} - \hat{\nabla}^{\mathrm{b}}\right) + \mathbf{g}, \\
\dot{\hat{p}}^{\mathrm{n}} &= \hat{v}^{\mathrm{n}},
\end{aligned} \tag{3}$$

where $\widehat{\mathbf{C}}_{\mathrm{b}}^{n}$ denotes the estimated rotation matrix from the body coordinate frame to the navigation coordinate frame, $\hat{\varepsilon}^{\mathrm{b}}$ and $\hat{\nabla}^{\mathrm{b}}$ are the estimates of the constant drifts of the gyroscopes and the accelerometers, which can be obtained by the calibration method [35], and

$$M'(\omega) = \begin{bmatrix} 0 & -\omega_x & -\omega_y & -\omega_z \\ \omega_x & 0 & \omega_z & -\omega_y \\ \omega_y & -\omega_z & 0 & \omega_x \\ \omega_z & \omega_y & -\omega_x & 0 \end{bmatrix}. \tag{4}$$

In addition, the unit quaternion Q, the attitudes θ, and the rotation matrixes $\mathbf{C}_{\mathrm{b}}^{\mathrm{n}}$ and $\mathbf{C}_{\mathrm{n}}^{\mathrm{b}}$ can be interchangeable [34].

In addition, ε^{b} and ∇^{b} can be modeled as follows:

$$\begin{aligned}
\dot{\varepsilon}^{\mathrm{b}} &= 0, \\
\dot{\nabla}^{\mathrm{b}} &= 0.
\end{aligned} \tag{5}$$

According to the analysis above, the IMU state vector can be described as follows:

$$\mathbf{X}_{\mathrm{I}} = \begin{bmatrix} Q^T & v^{\mathrm{n}T} & p^{\mathrm{n}T} & \varepsilon^{\mathrm{b}T} & \nabla^{\mathrm{b}T} \end{bmatrix}^T. \tag{6}$$

In practice, the angular velocity $\omega_{\mathrm{m}}^{\mathrm{b}}$ and the specific force $f_{\mathrm{m}}^{\mathrm{b}}$ are sampled by IMU at discrete times, so (3) should be calculated in a discrete method. Therefore, we assume that the $\omega_{\mathrm{m}}^{\mathrm{b}}$ and $f_{\mathrm{m}}^{\mathrm{b}}$ are constant between the two sampling times; thereafter, (3) can be discretized in an easy way.

3.2. Feature Representation. The point and line features are extracted from the images collected by the stereo camera. In this subsection, we will describe the representations of the point and line features.

3.2.1. Point Features. Owing to small scale of the indoor environments, the depth of the most points in images can be estimated by stereo camera through the baseline between the left and right cameras, so point features can be coded with a Cartesian coordinate frame as follows:

$$P_{\mathrm{p}}^{\mathrm{b}} = \begin{bmatrix} X & Y & Z \end{bmatrix}^T, \tag{7}$$

where $P_{\mathrm{p}}^{\mathrm{b}}$ denotes the position of point features in body coordinate frame and it can be estimated from the pixel coordinate value of the point features as follows:

$$\widehat{P}_{\mathrm{p}}^{\mathrm{b}} = \begin{bmatrix} \dfrac{b(u_{\mathrm{L}} - u_0)}{d} & \dfrac{b(v_{\mathrm{L}} - v_0)}{d} & \dfrac{bf}{d} \end{bmatrix}^T, \tag{8}$$

where b is the baseline between the left camera and the right camera, $d = u_{\mathrm{L}} - u_{\mathrm{R}}$ is the disparity, u_{L} and v_{L} are the pixel coordinate value of the point features in the left camera image, u_0 and v_0 are the pixel coordinate value of the optical center in the left camera image, and f is the focal length. In addition, u_0, v_0, and f are the intrinsic parameters of the camera which can be obtained by the camera calibration method [36].

The position of point features in navigation coordinate frame $P_{\mathrm{p}}^{\mathrm{n}}$ is also important in the proposed system, and the estimated position can be computed as follows:

$$\widehat{P}_{\mathrm{p}}^{\mathrm{n}} = \widehat{\mathbf{C}}_{\mathrm{b}}^{\mathrm{n}} \widehat{P}_{\mathrm{p}}^{\mathrm{b}} + \widehat{p}^{\mathrm{n}}. \tag{9}$$

This process is only executed when the point feature is detected in the first time. In this section, for the sake of simplicity, the rotation matrix between IMU coordinate frame and camera coordinate frame $C_{\mathrm{I}}^{\mathrm{C}}$, $C_{\mathrm{C}}^{\mathrm{I}}$ and the relative position between these two coordinate frames are ignored, but in practice, they should be considered.

According to the analysis above, the point feature state vector can be described as follows:

$$\mathbf{X}_{\mathrm{p}} = \begin{bmatrix} P_{\mathrm{p},1}^{\mathrm{n}} & P_{\mathrm{p},2}^{\mathrm{n}} & \cdots & P_{\mathrm{p},M}^{\mathrm{n}} \end{bmatrix}^T, \tag{10}$$

where $P_{\mathrm{p},i}^{\mathrm{n}}$ denotes the position of the ith point features in navigation coordinate frame and M is the number of the point features and this number is a variable because of the different images collected by the camera.

3.2.2. Line Features. Different from the points represented by $(X, Y, Z)^T$ in three-dimensional spaces, lines are parameterized in 4 DOFs. In this paper, we introduce two different representations of lines, the Plücker coordinate frame representation and the orthonormal representation [11, 37]. The Plücker coordinate frame representation is initialized when the line features are detected in the first time, and the orthonormal representation is used in the estimator.

The Plücker coordinate frame representation is determined by two points on the line. Assuming the homogeneous coordinate frames of these two points are $P_{X1} = (x_1, y_1, z_1, w_1)^T$ and $P_{X2} = (x_2, y_2, z_2, w_2)^T$, the Plücker matrix $\mathbf{L}$ can be determined as follows:

$$\mathbf{L} = P_{X2} P_{X1}^T - P_{X1} P_{X2}^T. \tag{11}$$

The Plücker matrix $\mathbf{L}$ is a 4×4 antisymmetric matrix in which the diagonal elements are zero and there are six nonzero elements in this matrix. The Plücker coordinate frame is a 6×1 vector composed by these six nonzero elements as below:

$$\mathscr{L} = \begin{bmatrix} \tilde{P}_{X1} \times \tilde{P}_{X2} \\ w_1 \tilde{P}_{X2} - w_2 \tilde{P}_{X1} \end{bmatrix} = \begin{bmatrix} n \\ v \end{bmatrix}, \tag{12}$$

where $\tilde{P}_{X1}$ and $\tilde{P}_{X2}$ are the 3×1 inhomogeneous coordinate frames of the two points. n is orthogonal to the interpretation plane containing the line and the origin, and v is the direction of the line. The relationship between the Plücker matrix $\mathbf{L}$ and the Plücker coordinate frame $\mathscr{L}$ is expressed as

$$\mathbf{L} = \begin{bmatrix} \lfloor n \times \rfloor & v \\ -v^T & 0 \end{bmatrix}, \tag{13}$$

where $\lfloor \cdot \times \rfloor$ denotes the skew-symmetric cross-product matrix with a vector and the $\lfloor n \times \rfloor$ is defined as

$$\lfloor n \times \rfloor = \begin{bmatrix} 0 & -n_3 & n_2 \\ n_3 & 0 & -n_1 \\ -n_2 & n_1 & 0 \end{bmatrix}. \tag{14}$$

The transformation of the Plücker coordinate frame in different reference coordinate frames (e.g., navigation coordinate frame and body coordinate frame) can be described as follows [38]:

$$\begin{aligned} \begin{bmatrix} n^{\mathrm{b}} \\ v^{\mathrm{b}} \end{bmatrix} &= \mathscr{H}_{\mathrm{n}}^{\mathrm{b}} \begin{bmatrix} n^{\mathrm{n}} \\ v^{\mathrm{n}} \end{bmatrix}, \\ \mathscr{H}_{\mathrm{n}}^{\mathrm{b}} &= \begin{bmatrix} \mathbf{C}_{\mathrm{n}}^{\mathrm{b}} & \left\lfloor \mathbf{t}_{\mathrm{n}}^{\mathrm{b}} \times \right\rfloor \mathbf{C}_{\mathrm{n}}^{\mathrm{b}} \\ 0 & \mathbf{C}_{\mathrm{n}}^{\mathrm{b}} \end{bmatrix}, \end{aligned} \tag{15}$$

$$\begin{aligned} \begin{bmatrix} n^{\mathrm{n}} \\ v^{\mathrm{n}} \end{bmatrix} &= \mathscr{H}_{\mathrm{b}}^{\mathrm{n}} \begin{bmatrix} n^{\mathrm{b}} \\ v^{\mathrm{b}} \end{bmatrix}, \\ \mathscr{H}_{\mathrm{b}}^{\mathrm{n}} &= \begin{bmatrix} \mathbf{C}_{\mathrm{b}}^{\mathrm{n}} & \lfloor \mathbf{t}_{\mathrm{b}}^{\mathrm{n}} \times \rfloor \mathbf{C}_{\mathrm{b}}^{\mathrm{n}} \\ 0 & \mathbf{C}_{\mathrm{b}}^{\mathrm{n}} \end{bmatrix}, \end{aligned} \tag{16}$$

where $\mathbf{t}_{\mathrm{n}}^{\mathrm{b}}$ denotes the translation vector from navigation coordinate frame to the body coordinate frame.

The 3D lines represented by Plücker coordinate frames in body coordinate frame can also be projected onto the image plane as follows:

$$\begin{aligned} l^{\mathrm{b}} &= \mathscr{K} n^{\mathrm{b}}, \\ \mathscr{K} &= \begin{bmatrix} f_v & 0 & 0 \\ 0 & f_u & 0 \\ -f_v c_u & f_u c_v & f_u f_v \end{bmatrix}, \end{aligned} \tag{17}$$

where l^{b} is the 2D line in the image plane represented by pixel coordinate frame. It can be found that only n^{b} is used in this projection process.

According to the analysis above, six elements are used in the Plücker coordinate frame representation while the line's DOFs are four. If using this representation in estimator, superfluous elements will cost more computation. In addition, the orthogonal constraint between the n and v, that is, $n^T v = 0$, will decrease the numerical stability, and thus, we need to use another representation method in the estimator.

The orthonormal representation uses minimum 4 parameters to represent a line, and this representation can be derived from the Plücker coordinate frame representation [37].

We define a matrix $\mathbf{S} = [n \mid v]$ and decompose this matrix by QR decomposition as

$$\mathrm{QR}(\mathbf{S}) = \mathbf{U} \begin{bmatrix} w_1 & 0 \\ 0 & w_2 \\ 0 & 0 \end{bmatrix} \tag{18}$$

and set

$$\mathbf{W} = \begin{bmatrix} w_1 & -w_2 \\ w_2 & w_1 \end{bmatrix}. \tag{19}$$

The 3D line can be represented by $(\mathbf{U}, \mathbf{W})$ where $\mathbf{U} \in \mathrm{SO}(3), \mathbf{W} \in \mathrm{SO}(2)$. This means the $\mathbf{U}$ and $\mathbf{W}$ are three- and two-dimensional rotation matrices, respectively. Thus, the matrix $\mathbf{U}$ can be updated accordingly by a vector containing 3 parameters such as Euler angles $\theta_{\mathrm{l}} = \begin{bmatrix} \theta_{\mathrm{l},1} & \theta_{\mathrm{l},2} & \theta_{\mathrm{l},3} \end{bmatrix}^T$ and $\mathbf{W}$ can be updated by a scalar parameter $\theta_{\mathrm{l}} \in (0, \pi/2)$ as follows:

$$\begin{aligned} \mathbf{U}^* &= \mathbf{U}\mathbf{R}(\theta_{\mathrm{l}}), \\ \mathbf{R}(\theta_{\mathrm{l}}) &= \mathbf{R}_x(\theta_{\mathrm{l},1}), \mathbf{R}_y(\theta_{\mathrm{l},2}), \mathbf{R}_z(\theta_{\mathrm{l},3}), \\ \mathbf{W}^* &= \begin{bmatrix} \cos\theta_{\mathrm{l}} & -\sin\theta_{\mathrm{l}} \\ \sin\theta_{\mathrm{l}} & \cos\theta_{\mathrm{l}} \end{bmatrix} \mathbf{W}, \end{aligned} \tag{20}$$

where $\mathbf{R}_x(\theta_1), \mathbf{R}_y(\theta_2)$, and $\mathbf{R}_z(\theta_3)$ denote the three-dimensional rotation matrices.

We can define a 4×1 increment vector $\boldsymbol{\sigma} = [\theta_{\mathrm{I}}^T, \theta_{\mathrm{l}}]^T$ which can be used in the estimator presented in the next section to represent the error of line features. The detailed discussions will be provided in Section 4.

In addition, the Plücker coordinate frame can be transferred from the orthonormal representation as follows:

$$\mathscr{L} = \begin{bmatrix} w_1 u_1 \\ w_2 u_2 \end{bmatrix}, \tag{21}$$

where u_i denotes the ith column of $\mathbf{U}$.

4. Estimator Establishment and Algorithm Implementation

In this section, an EKF estimator is presented. The IMU measurements along with the point and line features mentioned in Section 3 are combined based on this estimator in a tightly coupled approach. In addition, we will elaborate the feature detection methods, the feature initialization, and some other processing steps.

4.1. Error-State Equation. In our work, the EKF estimator uses the error model, which is defined as follows:

$$\mathbf{X}^T = \begin{bmatrix} \boldsymbol{\delta}\mathbf{X}_{\mathrm{I}}^T & \boldsymbol{\delta}\mathbf{X}_{\mathrm{p}}^T & \boldsymbol{\sigma}_{\mathrm{l}}^T \end{bmatrix}, \tag{22}$$

where $\boldsymbol{\delta}\mathbf{X}_{\mathrm{I}}^T$, $\boldsymbol{\delta}\mathbf{X}_{\mathrm{p}}^T$, and $\boldsymbol{\sigma}_{\mathrm{l}}^T$ are the error-state vectors of IMU, point features, and line features, respectively.

The IMU error-state vector can be defined as

$$\boldsymbol{\delta}\mathbf{X}_{\mathrm{I}}^T = \begin{bmatrix} \delta\theta_{\mathrm{I}}^T & \delta v^{\mathrm{n}T} & \delta p^{\mathrm{n}T} & \delta\varepsilon^{\mathrm{b}T} & \delta\nabla^{\mathrm{b}T} \end{bmatrix}, \tag{23}$$

where δv^{n}, δp^{n}, $\delta\varepsilon^{\mathrm{b}}$, and $\delta\nabla^{\mathrm{b}}$ are the velocity error, the position error, the error of gyroscope constant drift, and the error of accelerometer constant drift, respectively. The aforementioned error is formulated as $\delta x = x - \hat{x}$ where x is the true value and $\hat{x}$ is the estimated value. $\delta\theta_{\mathrm{I}}$ is the attitude error computed from the error quaternion δQ which is defined as

$$\delta Q \approx \begin{bmatrix} 1 \\ \dfrac{\theta_{\mathrm{I}}^T}{1} \end{bmatrix}^T. \tag{24}$$

The relationship between the true quaternion Q and the estimated quaternion $\widehat{Q}$ is defined as $Q = \widehat{Q} \otimes \delta Q$, where $\otimes$ is the quaternion multiplication [34].

Similarly, the error-state vector of point features can be defined as

$$\boldsymbol{\delta}\mathbf{X}_{\mathrm{p}}^T = \begin{bmatrix} \boldsymbol{\delta}\mathbf{P}_{\mathrm{p},1}^{\mathrm{n}} & \boldsymbol{\delta}\mathbf{P}_{\mathrm{p},2}^{\mathrm{n}} & \cdots & \boldsymbol{\delta}\mathbf{P}_{\mathrm{p},M}^{\mathrm{n}} \end{bmatrix}, \tag{25}$$

where $\delta P_{\mathrm{p},i}^{\mathrm{n}}$ is the position error of the ith point features in navigation coordinate frame which is defined as $\delta P_{\mathrm{p},i}^{\mathrm{n}} = P_{\mathrm{p},i}^{\mathrm{n}} - \widehat{P}_{\mathrm{p},i}^{\mathrm{n}}$. In addition, $\boldsymbol{\delta}\dot{\mathbf{X}}_{\mathrm{p}} = 0$.

The error-state vector of the line features can be defined as follows:

$$\boldsymbol{\sigma}_{\mathrm{l}}^{\boldsymbol{T}} = \begin{bmatrix} \boldsymbol{\sigma}_1^{\mathrm{n}\boldsymbol{T}} & \boldsymbol{\sigma}_2^{\mathrm{n}\boldsymbol{T}} & \cdots & \boldsymbol{\sigma}_N^{\mathrm{n}\,\boldsymbol{T}} \end{bmatrix}, \tag{26}$$

where $\boldsymbol{\sigma}_j^{\mathrm{n}}$ denotes the error vector of the jth line features in navigation coordinate frame and N is the number of the line features. It is worth to mention that N is a variable because of the different images collected by the camera. In addition, $\dot{\boldsymbol{\sigma}}_{\mathrm{l}} = 0$.

According to the analysis above, the error-state equation can be established as follows:

$$\begin{bmatrix} \delta\dot{\mathbf{X}}_{\mathrm{I}} \\ \delta\dot{\mathbf{X}}_{\mathrm{P}} \\ \dot{\sigma}_{\mathrm{l}} \end{bmatrix} = F \begin{bmatrix} \delta\mathbf{X}_{\mathrm{I}} \\ \delta\mathbf{X}_{\mathrm{P}} \\ \sigma_{\mathrm{l}} \end{bmatrix} + w,$$

$$F = \begin{bmatrix} F_{\mathrm{I}} & 0_{15\times 3M} & 0_{15\times 4N} \\ 0_{3M\times 15} & 0_{3M\times 3M} & 0_{3M\times 4N} \\ 0_{4N\times 15} & 0_{4N\times 3M} & 0_{4N\times 4N} \end{bmatrix}, \quad (27)$$

$$w = \begin{bmatrix} w_{\mathrm{I}} \\ 0_{3M\times 4N} \\ 0_{3M\times 4N} \end{bmatrix},$$

where w_{I} is the process noise assumed to be uncorrelated white zero-mean Gaussian noise the covariance matrix of which is Q_w and

$$F_{\mathrm{I}} = \begin{bmatrix} -\left\lfloor \left(\omega_{\mathrm{m}}^{\mathrm{b}} - \varepsilon^{\mathrm{b}}\right)\times \right\rfloor & 0_{3\times 3} & 0_{3\times 3} & -I_3 & 0_{3\times 3} \\ -\mathbf{C}_{\mathrm{b}}^{\mathrm{n}} \left\lfloor \left(f_{\mathrm{m}}^{\mathrm{b}} - \nabla^{\mathrm{b}}\right)\times \right\rfloor & 0_{3\times 3} & 0_{3\times 3} & 0_{3\times 3} & -\mathbf{C}_{\mathrm{b}}^{\mathrm{n}} \\ 0_{3\times 3} & I_3 & 0_{3\times 3} & 0_{3\times 3} & 0_{3\times 3} \\ 0_{6\times 3} & 0_{6\times 3} & 0_{6\times 3} & 0_{6\times 3} & 0_{6\times 3} \end{bmatrix}. \quad (28)$$

Owing to the estimator is executed in the discrete time, and the differential equation needs to be transformed into the discrete equation as follows:

$$\begin{aligned} \mathbf{X}_k &= \Phi_{k,k-1}\mathbf{X}_{k-1} + w_{k-1}, \\ \Phi_{k,k-1} &\approx I + \mathrm{FT}, \end{aligned} \quad (29)$$

where T is the sampling period and k denotes the kth frame.

In addition, in any environment, the stereo camera is used to capture a series of fixed images. According to the point and line feature extraction methods, for a series of fixed images, the extracted points and lines are also fixed and we assume that the matched features are all true positive, and thus, there is no random noise for these features. Actually, they only contain the measurement noise the differential of which equal to zero. Therefore, in (27), there is no noise of point and line features.

4.2. Measurement Equation. The features' reprojection errors in the image plane are selected as the measurement. For the sake of illustration, we present the reprojection errors by considering the case where a single point feature and a single line feature are present.

We first define $P_{\mathrm{p},i}^{\mathrm{n}} = \begin{bmatrix} X_i & Y_i & Z_i \end{bmatrix}^T$ as the ith point feature which can be observed by the camera in the kth frame. The reprojection error of the point feature $z_{\mathrm{p},i}$ can be defined as follows:

$$\begin{aligned} \hat{P}_{\mathrm{p},i}^{\mathrm{b}} &= \hat{\mathbf{C}}_{\mathrm{n}}^{\mathrm{b}}\left(P_{\mathrm{p},i}^{\mathrm{n}} - \hat{p}_{\mathrm{k}}^{\mathrm{n}}\right) = \begin{bmatrix} X_i^{\mathrm{b}} & Y_i^{\mathrm{b}} & Z_i^{\mathrm{b}} \end{bmatrix}^T, \\ \hat{P}_{\mathrm{p},i}{}' &= \begin{bmatrix} \hat{u}_i \\ \hat{v}_i \end{bmatrix} = \begin{bmatrix} f_u \dfrac{X_i^{\mathrm{b}}}{Z_i^{\mathrm{b}}} + u_0 \\ f_v \dfrac{Y_i^{\mathrm{b}}}{Z_i^{\mathrm{b}}} + v_0 \end{bmatrix}, \\ z_{\mathrm{p},i} &= P_{\mathrm{p},i}{}' - \hat{P}_{\mathrm{p},i}{}' = \begin{bmatrix} u_i - \hat{u}_i \\ v_i - \hat{v}_i \end{bmatrix}, \end{aligned} \quad (30)$$

where $\hat{p}_k^{\mathrm{n}}$ is the system's position in the kth frame, $\hat{P}_{\mathrm{p},i}^{\mathrm{b}}$ is the point feature's estimated position in the body coordinate frame, $\hat{P}_{\mathrm{p},i}{}'$ is the point feature's reprojection position in the image plane, and $(\hat{u}_i, \hat{v}_i)$ is its pixel value. $P_{\mathrm{p},i}{}'$ is the point feature's observation value in the image plane, and (u_i, v_i) is its pixel value.

We define $\mathscr{L}_j^{\mathrm{n}} = \left[n_j^{\mathrm{n}T} v_j^{\mathrm{n}T}\right]^T$ as the jth line feature which can be observed by the camera in the kth frame. The reprojection error of the line feature $z_{\mathrm{l},j}$ can be defined as follows:

$$\hat{\mathscr{L}}_j^{\mathrm{b}} = \begin{bmatrix} \hat{n}_j^{\mathrm{b}} \\ \hat{v}_j^{\mathrm{b}} \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{C}}_{\mathrm{n}}^{\mathrm{b}} & \left\lfloor \hat{\mathbf{t}}_{\mathrm{n}}^{\mathrm{b}} \times \right\rfloor \hat{\mathbf{C}}_{\mathrm{n}}^{\mathrm{b}} \\ 0 & \hat{\mathbf{C}}_{\mathrm{n}}^{\mathrm{b}} \end{bmatrix} \mathscr{L}_j^{\mathrm{n}}, \quad (31)$$

$$\hat{l}_j^{\mathrm{b}} = \begin{bmatrix} f_v & 0 & 0 \\ 0 & f_u & 0 \\ -f_v c_u & f_u c_v & f_u f_v \end{bmatrix} \hat{n}_j^{\mathrm{b}} = \begin{bmatrix} \hat{l}_{j,1}^{\mathrm{b}} \\ \hat{l}_{j,2}^{\mathrm{b}} \\ \hat{l}_{j,3}^{\mathrm{b}} \end{bmatrix}, \quad (32)$$

$$z_{\mathrm{l},j} = \begin{bmatrix} \dfrac{x_{j,\mathrm{s}}^T \cdot \hat{l}_j^{\mathrm{b}}}{\sqrt{\left(\hat{l}_{j,1}^{\mathrm{b}}\right)^2 + \left(\hat{l}_{j,2}^{\mathrm{b}}\right)^2}} & \dfrac{x_{j,\mathrm{e}}^T \cdot \hat{l}_j^{\mathrm{b}}}{\sqrt{\left(\hat{l}_{j,1}^{\mathrm{b}}\right)^2 + \left(\hat{l}_{j,2}^{\mathrm{b}}\right)^2}} \end{bmatrix}^T, \quad (33)$$

where $\hat{\mathscr{L}}_j^{\mathrm{b}}$ is the line feature's estimated position in the body coordinate frame, $\hat{l}_j^{\mathrm{b}}$ is the point feature's reprojection position in the image plane. In addition, when the line feature is observed by the camera, we can obtain the pixel value of the start-point and end-point of the line feature. In (33), $x_{j,\mathrm{s}}^T$ and $x_{j,\mathrm{e}}^T$ denote the start-point and end-point of the line feature's observation value in the image plane. Note that $x_{j,\mathrm{s}}^T$ and $x_{j,\mathrm{e}}^T$ are represented by the homogeneous coordinate frames which are 3×1 vectors (the first two elements are the pixel value, and the third element is 1).

According to the analysis above, the measurement vector can be selected as follows:

$$\mathbf{Z}^T = \left[z_{\mathrm{p},1}, z_{\mathrm{p},2}, \ldots, z_{\mathrm{p},M}, z_{\mathrm{l},1}, z_{\mathrm{l},2}, \ldots, z_{\mathrm{l},N}\right]. \tag{34}$$

The error-state equation can be established as follows:

$$\mathbf{Z} = \mathbf{H}\mathbf{X} + \eta, \tag{35}$$

where η is the measurement noise assumed to be uncorrelated white zero-mean Gaussian process the covariance matrix of which is $\mathbf{R}_\eta$ and $\mathbf{H}$ is the measurement Jacobian matrix formulated as:

$$\mathbf{H} = \begin{bmatrix} \mathbf{H}_{\mathrm{p},1}^T & \cdots & \mathbf{H}_{\mathrm{p},M}^T & \mathbf{H}_{\mathrm{l},1}^T & \cdots & \mathbf{H}_{\mathrm{l},N}^T \end{bmatrix}^T, \tag{36}$$

where the individual matrixes $\mathbf{H}_{\mathrm{p},i}$ and $\mathbf{H}_{\mathrm{l},j}$ can be computed as

$$\begin{aligned}
\mathbf{H}_{\mathrm{p},i} &= J_{z,i}\begin{bmatrix} \left\lfloor \mathbf{C}_{\mathrm{n}}^{\mathrm{b}}\left(P_{\mathrm{p},i}^{\mathrm{n}} - p^{\mathrm{n}}\right)\times \right\rfloor & 0_{3\times3} & -\mathbf{C}_{\mathrm{n}}^{\mathrm{b}} & 0_{3\times6} & \cdots & \mathbf{C}_{\mathrm{n}}^{\mathrm{b}} & \cdots \end{bmatrix}, \\
\mathbf{H}_{\mathrm{l},j} &= J_{\xi,j}\begin{bmatrix} -\left\lfloor \mathbf{C}_{\mathrm{n}}^{\mathrm{b}} n^{\mathrm{n}}\times \right\rfloor - \left\lfloor \left\lfloor \mathbf{t}_{\mathrm{n}}^{\mathrm{b}}\times \right\rfloor \mathbf{C}_{\mathrm{n}}^{\mathrm{b}} v^{\mathrm{n}}\times \right\rfloor & 0_{3\times3} & -\left\lfloor \mathbf{C}_{\mathrm{n}}^{\mathrm{b}} v^{\mathrm{n}}\times \right\rfloor & 0_{3\times6} & \cdots & J_{\delta,j} & \cdots \\ -\left\lfloor \mathbf{C}_{\mathrm{n}}^{\mathrm{b}} v^{\mathrm{n}}\times \right\rfloor & & 0_{3\times3} & 0_{3\times3} & 0_{3\times6} & & \end{bmatrix}, \\
J_{z,i} &= \begin{bmatrix} \dfrac{f_u}{Z_i^{\mathrm{b}}} & 0 & \dfrac{-f_u X_i^{\mathrm{b}}}{\left(Z_i^{\mathrm{b}}\right)^2} \\ 0 & \dfrac{f_v}{Z_i^{\mathrm{b}}} & \dfrac{-f_v Y_i^{\mathrm{b}}}{\left(Z_i^{\mathrm{b}}\right)^2} \end{bmatrix}, \\
J_{\delta,j} &= \mathscr{H}_{\mathrm{n}}^{\mathrm{b}}\begin{bmatrix} -\left\lfloor w_{j,1}u_{j,1}\times \right\rfloor & -w_{j,2}u_{j,1} \\ -\left\lfloor w_{j,2}u_{j,2}\times \right\rfloor & -w_{j,1}u_{j,2} \end{bmatrix}, \\
J_{\xi,j} &= \frac{1}{l_{\mathrm{n}}}\begin{bmatrix} u_{j,1} - \dfrac{l_{j,1}^{\mathrm{b}} e_{j,1}}{l_{\mathrm{n}}^2} & v_{j,1} - \dfrac{l_{j,2}^{\mathrm{b}} e_{j,1}}{l_{\mathrm{n}}^2} & 1 \\ u_{j,2} - \dfrac{l_{j,1}^{\mathrm{b}} e_{j,2}}{l_{\mathrm{n}}^2} & v_{j,2} - \dfrac{l_{j,2}^{\mathrm{b}} e_{j,2}}{l_{\mathrm{n}}^2} & 1 \end{bmatrix}\left[\mathscr{K} 0_{3\times3}\right],
\end{aligned} \tag{37}$$

where $e_{j,1} = x_{j,\mathrm{s}}^T l_j^{\mathrm{b}}$, $e_{j,2} = x_{j,\mathrm{e}}^T l_j^{\mathrm{b}}$, $l_{\mathrm{n}} = \sqrt{\left(l_{j,1}^{\mathrm{b}}\right)^2 + \left(l_{j,2}^{\mathrm{b}}\right)^2}$, $x_{j,\mathrm{s}}^T = \begin{bmatrix} u_{j,1} & v_{j,1} & 1 \end{bmatrix}$, and $x_{j,\mathrm{e}}^T = \begin{bmatrix} u_{j,2} & v_{j,2} & 1 \end{bmatrix}$.

The measurement equation can also be written as the discrete form:

$$\mathbf{Z}_k = \mathbf{H}_k\mathbf{X}_k + \eta_k. \tag{38}$$

4.3. Estimator and Algorithm Implementation. According to the analysis in the previous section, it can be found that the error-state equation and the measurement equation are nonlinear. In order to deal with the linearization error, we use the EKF estimator to fuse the inertial measurement and the images observed by the stereo camera. As presented in the previous section, these nonlinear functions have been linearized.

Based on the linearized error-state and measurement equations, the EKF estimator can be executed with the following two steps:

(i) Predict

$$\begin{aligned}
\mathbf{X}_{k+1,k} &= \Phi_{k+1,k}\mathbf{X}_k, \\
\mathbf{P}_{k+1,k} &= \Phi_{k+1,k}\mathbf{P}_k\Phi_{k+1,k}^T + \mathbf{Q}_{w,k},
\end{aligned} \tag{39}$$

where $\mathbf{P}_k$ is the covariance matrix of the state vector.

(ii) Update

$$\begin{aligned}
\mathbf{K}_k &= \boldsymbol{P}_{k+1,k}\mathbf{H}_k^T\left(\mathbf{H}_k\mathbf{P}_{k+1,k}\mathbf{H}_k^T + \mathbf{R}_{\eta,k}\right)^{-1}, \\
\mathbf{X}_{k+1,k+1} &= \mathbf{X}_{k+1,k} + \mathbf{K}_k(\mathbf{Z}_k - \mathbf{H}_k\mathbf{X}_{k+1,k}), \\
\mathbf{P}_{k+1,k+1} &= (I - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_{k+1,k},
\end{aligned} \tag{40}$$

where $\mathbf{K}_k$ is the Kalman gain matrix.

The covariance matrices $\mathbf{P}_k$, $\mathbf{Q}_w$, and $\mathbf{R}_\eta$ in the above equations are empirically initialized based on the characteristics of the sensors. Specifically, $\mathbf{Q}_w$ and $\mathbf{R}_\eta$ are initialized, respectively, associated with the random noise of the inertial

sensors and the scales of the feature space. $\mathbf{P}_k$ is initialized concerning the constant drifts of the inertial sensors and the scales of the feature space.

Finally, we summarize the whole algorithm. In this system, static base alignment is achieved for obtaining the initial attitude angle by following the method in [30]. Subsequently, analogous to the inertial navigation system [34], the IMU measurement is used to update the pose and the velocity of the system by solving (3).

On the other hand, the images are collected by the stereo camera; after performing the feature detection and initialization which will be discussed in the next subsection, the error-state and measurement equations can be established by (22), (23), (24), (25), (26), (27), (28), (29), (30), (31), (32), (33), (34), (35), (36), (37), and (38) with the IMU measurement and the update result of the inertial navigation. In the last step, the EKF estimator is executed by (39) and (40). In addition, the covariance matrix of the EKF should be augmented when new point or line features are observed.

4.4. Feature Detection and Initialization. In terms of the feature detection, the ORB method [23] is used to extract point features from the images. This method deals with the orientation problem of the FAST method and generates a very fast binary descriptor based on BRIEF to describe the detected point features. Therefore, this method can detect and describe the point features in a fast, efficient, and accurate manner. In [21, 22], the ORB method was used in the visual SLAM system, which resulted in an excellent performance.

On the other hand, the LSD method [24, 25] is used to extract the line features from the images. Its mechanism is to merge pixels with the similar gradient direction. This method can extract the lines with subpixel accuracy in linear time. In addition, the detected lines are described by a binary descriptor, line band descriptor (LBD) [39], which allows accurate matching between two frames or two cameras (left and right).

The nearest-neighbor method is used to match the features between the two frames based on the descriptor of features. In order to reduce the false matches, the random sample consensus (RANSAC) method is used to filter out the outliers for both point features and line features.

Once a new feature is detected and described, we next perform binocular matching. For a point feature in the left image, its counterpart in the right image can be acquired along the parallel epipolar line in the right image. The Hamming distance is used to measure the similarity of point features with a predefined threshold. For the line feature, we exploit the matching method proposed in [40].

The unsuccessfully matched features will be discarded. After matching the point and line features between the two images, the positions of the point features and the two endpoints of the line features can be obtained by (8). The representations of line features in the 3D space can be computed by (12). The point features and lines in the navigation coordinate frame can be computed by (9) and (16). With the above-mentioned feature initialization, the initial positions of the features in the navigation system are obtained and are thus used to compute the reprojection error in the measurement equation. In addition, after executing one filtering process, the error of the features' positions in the navigation coordinate frame can be estimated after feature filtering. Therefore, these positions can be updated in the filtering processes and become progressively accurate.

5. Experiments and Results

In this section, experiments are conducted to test the effectiveness of the proposed indoor navigation system. We assume our system is tailored for any structured indoor environment where the geometric scale information can be calculated by the stereo camera and the object textures can be characterized by either point or line features. We use EuRoC dataset [41] for evaluating our approach and use root-mean-square error (RMSE) for performance measure. In the comparative study, we compare the proposed navigation system (PL-VINS), with the StVO-PL [26] and the system based on multistate constraint Kalman filter (MSCKF) [15]. The StVO-PL is a state-of-the-art visual navigation system using stereo camera with points and lines, while the MSCKF is an EKF-based visual-inertial navigation system with high accuracy. In implementation, we directly use the publicly available source code for both StVO-PL and MSCKF methods. For the sake of consistency, we repeat all the competing approaches ten times on all the sequences and the final experimental results are obtained by averaging the accuracy scores.

5.1. Experiment Description. The EuRoC dataset contains 11 sequences which are categorized into two types both recorded in three indoor environments. The first type of the dataset contains the first five sequences which are recorded in a machine hall (MH). The second type of the dataset contains the remaining six sequences recorded in Vicon room 1 (V1) and Vicon room 2 (V2) with an approximate size of $8\,\mathrm{m} \times 8.4\,\mathrm{m} \times 4\,\mathrm{m}$. All eleven sequences include the stereo images and the IMU measurements containing the gyroscopes, accelerometers data along with the ground truth. More specifically, the stereo images are recorded by a stereo camera, MT9V034 with $2 \times 20\,\mathrm{Hz}$ while the IMU measurements are recorded by a MEMS inertial sensor, ADIS16448 with 200 Hz. Besides, the ground truths are provided by Leica MS50 and VICON. All these equipment are mounted on a micro air vehicle (MAV) which flies in the indoor environment to record the data. More details about the EuRoC dataset can be found in [41]. In addition, these 11 sequences provide varying challenges relevant to the speed of the MAV, illumination, texture, and so forth. In some sequences, especially for the sequence Vicon room 2 03, the environment is low-textured with few visual contents.

According to the description of the EuRoC dataset, it can be found that these sequences are suitable for testing the proposed indoor navigation system because of the visual-inertial data and the low-textured indoor environments. In addition, to the best of our knowledge, this dataset is the only

dataset which provides visual-inertial data and ground truth in indoor environments.

In order to test the algorithms quantitatively, the transformation error matrix between two coordinate frames can be computed as follows:

$$\boldsymbol{\delta T} = \widehat{\mathbf{T}}^{-1}\mathbf{T},$$
$$\mathbf{T} = \begin{bmatrix} \mathbf{C}_n^b & \mathbf{t}_n^b \\ 0_{1\times 3} & 1 \end{bmatrix}, \tag{41}$$

where $\mathbf{T}$ is the transformation matrix between two coordinate frames provided by the ground truth, $\widehat{\mathbf{T}}$ is the estimation transformation matrix from the navigation system, and $\boldsymbol{\delta T}$ is the error matrix. The translation errors and the rotation errors can be obtained from this error matrix.

The RMSE of translation and rotation can also be computed for comparison of different systems:

$$\text{RMSE(Transal.)} = \left(\frac{1}{m}\sum_{i=1}^{m}\|\text{trans}(\boldsymbol{\delta T}_i)\|^2\right)^{1/2},$$
$$\text{RMSE(Rot.)} = \left(\frac{1}{m}\sum_{i=1}^{m}\|\text{Rot}(\boldsymbol{\delta T}_i)\|^2\right)^{1/2}, \tag{42}$$

where m is the number of the frames, $\text{trans}(\boldsymbol{\delta T}_i)$ is the 3×1 translation error vector, and $\text{Rot}(\boldsymbol{\delta T}_i)$ is the 3×1 rotation error vector which are represented by the Euler angles.

5.2. Results and Discussion

5.2.1. Translation Errors. Extensive evaluations on 11 sequences are carried out by making use of three different navigation systems mentioned above. First, we analyze the translation errors of the dataset. Figures 1 and 2 show the translation errors of different methods on sequences machine hall 04 and Vicon room 2 02. Table 1 gives the relative x, y, z, and whole translation RMSE errors of all 11 sequences. In Table 1, the "T" column means the whole translation RMSE errors. "MH," "V1," and "V2," respectively, indicate the "machine hall," the "Vicon room 1," and the "Vicon room 2." "MH01" implies the sequence 01 in machine hall, and another four sequences named from "MH02" to "MH05" are also used in our evaluations. Besides, we also use the respective three sequences in V1 and V2 in our experiments, namely, "V1 01," "V1 02," "V1 03," "V2 01," "V2 02," and "V2 03." In addition, in order to demonstrate the robot movement in some environment, we give the ground truth trajectories and the trajectories estimated by the proposed method as shown in Figure 3.

It can be seen from Figures 1 and 2 and Table 1 that the proposed navigation system reports higher accuracy than MSCKF and StVO-PL in terms of translation errors. The StVO-PL achieves the lowest positioning accuracy with the largest x, y, z, and whole translation RMSE errors. For most of the sequences, the translation RMSE errors of the proposed method are better than those of the MSCKF. In addition, the proposed navigation system is more robust than the other two systems. The error curves of the

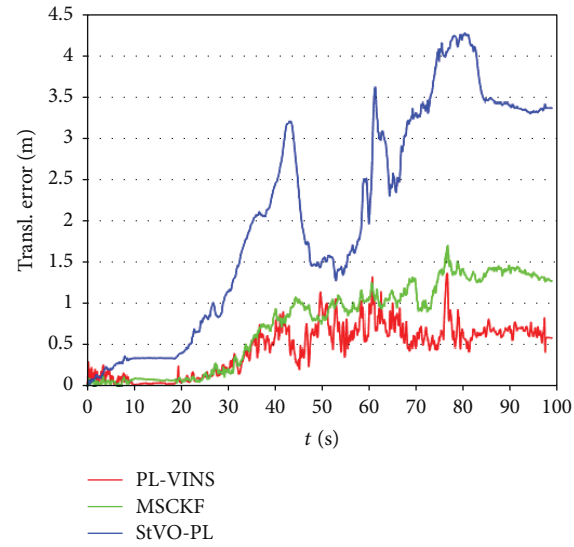


Figure 1: Translation errors of machine hall 04.

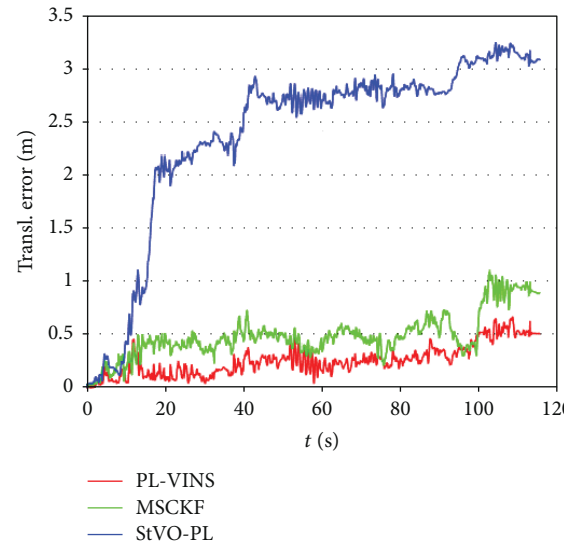


Figure 2: Translation errors of Vicon room 2 02.

proposed method are smoother than those of the StVO-PL. On sequence Vicon room 2 03, both the StVO-PL and MSCKF underperform while the proposed system exhibits superior performance.

It is observed that in terms of the sequences with substantial point features generated, point-based VINS performs well such that the line features provide limited performance improvements. In some cases, particularly, the measurement noise of the line features and the incorrect matches could lead to a slight worse performance such as the results in sequence machine hall 02. It is also shown that our approach outperforms MSCKF in the case when the features are insufficient. Thus, in specific cases, the performance of the proposed system is likely to be compromised by the measurement noise and the false matches; however, most of the results

Table 1: Translation errors (RMSE) in meters of the EuRoC dataset.

	VINS (point and lines) (m)				MSCKF (m)				StVO-PL (m)			
	x	y	z	T	x	y	z	T	x	y	z	T
MH 01	0.105	0.112	0.116	0.193	0.352	0.139	0.071	0.386	0.813	0.271	0.388	0.941
MH 02	0.082	0.168	0.108	0.216	0.134	0.126	0.078	0.201	0.929	0.251	0.202	0.984
MH 03	0.306	0.137	0.172	0.377	0.174	0.491	0.124	0.535	1.669	1.069	0.271	2.001
MH 04	0.323	0.322	0.307	0.551	0.629	0.665	0.114	0.923	0.865	1.365	1.853	2.458
MH 05	0.376	0.335	0.209	0.546	0.569	0.326	0.645	0.920	0.672	2.232	0.677	2.423
V1 01	0.286	0.101	0.109	0.261	0.797	1.004	0.717	0.322	0.797	1.003	0.718	1.468
V1 02	0.081	0.085	0.175	0.211	0.181	0.146	0.055	0.239	0.250	0.379	0.392	0.599
V1 03	0.202	0.247	0.134	0.347	0.291	0.357	0.396	0.608	0.264	1.823	0.453	1.897
V2 01	0.098	0.171	0.045	0.202	0.206	0.214	0.238	0.381	0.971	0.388	0.962	1.421
V2 02	0.096	0.246	0.120	0.289	0.412	0.203	0.264	0.531	1.519	0.968	1.766	2.522
V2 03	0.529	0.649	0.567	1.011	—	—	—	—	—	—	—	—

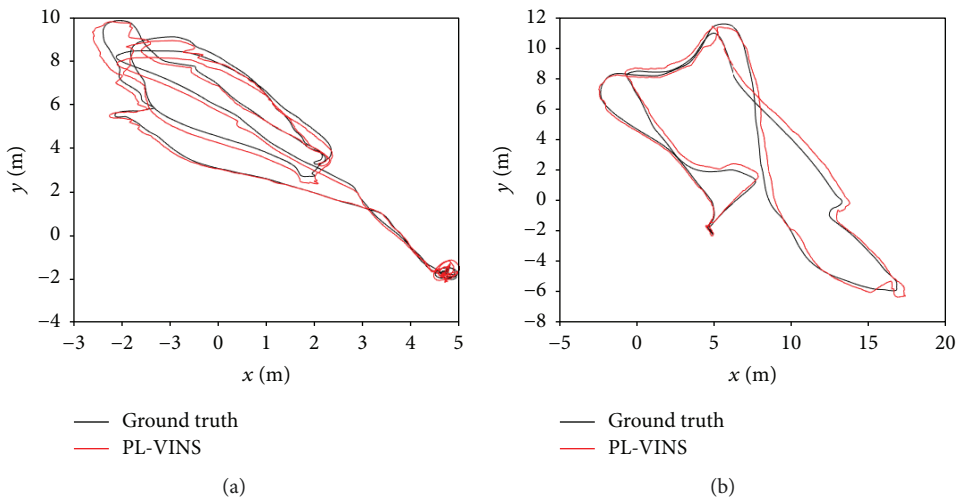


Figure 3: (a) Ground truth, the trajectories estimated by our algorithm on MH 02 sequence, which is viewed from the gravity direction. (b) Ground truth, the trajectories estimated by our algorithm on MH 05 sequence, which is viewed from the gravity direction.

significantly demonstrate the advantage of our system against other competing system.

It is also observed from figures that the error of the StVO-PL increases over time. Since StVO-PL is a vision-based navigation system, serious accumulated error is produced, which significantly impairs the performance of the system. By contrast, the proposed method and the MSCKF are visual-inertial navigation system, which enables reducing the error accumulation. Besides, both of them achieve very high positioning accuracy. Furthermore, compared to the MSCKF, the proposed method utilizes both the line features and point features. The line features abundant in indoor or man-made environments can use edges and linear shapes in environments. Thus, in some complex low-textured environments such as sequence Vicon room 2 03, the proposed system can still extract enough features to estimate the motion, while other systems such as MSCKF suffer from the lack of point features, which contributes to better positioning accuracy and the robustness of the proposed system.

5.2.2. Rotation Errors. In this subsection, we analyze the rotation errors of the EuRoC dataset. Similarly, as for sequences machine hall 05 and Vicon room 1 03, the rotation errors of three systems along with time are shown in Figures 4 and 5. The relative x, y, z, and whole rotation RMSE errors of all 11 sequences are shown in Table 2. In Table 2, the "R" column denotes the whole rotation RMSE errors.

The results of the rotation is similar to the results of the translation. It can be seen from the two figures and the table that the rotation accuracy of the StVO-PL is much worse than the other two systems with large errors of roll, pitch, and yaw angles. Despite the smooth motion in some sequences, the StVO-PL system does not perform well, due to the accumulated error of the visual navigation system. As for the other two navigation systems, the proposed navigation system has the relatively small rotation errors. However, in some sequences, the rotation errors of these two systems are very close, probably due to the smooth movement. For the challenging sequences with sudden angle changes, the

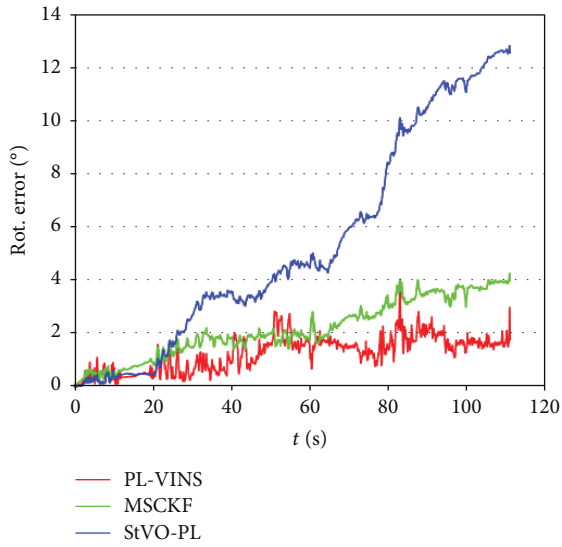


Figure 4: Rotation errors of the machine hall 05.

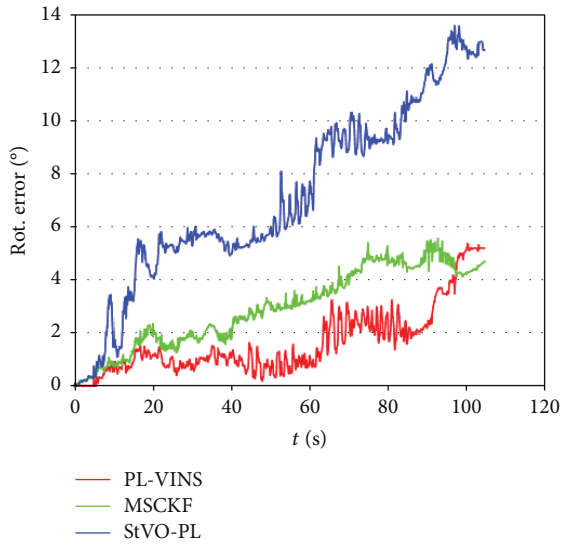


Figure 5: Rotation errors of the Vicon room 1 03.

proposed navigation system performs better than the MSCKF. In addition, the robustness of the proposed navigation is also better than other two systems. Both StVO-PL and MSCKF fail for the sequence Vicon room 2 03 which has many low-textured scenes, but the proposed navigation system executes successfully. As analogous to the translation errors, our method reports higher rotation errors than the MSCKF in some scenario, which can be caused by the measurement noise and false matches of line features.

Finally, it can be observed from Tables 1 and 2 that our approach has been evaluated on all the eleven video clips which provides different challenges, and the experimental results demonstrate that our method beats the other competing techniques on ten video clips, which considerably suggests the consistent superiority of our method.

To summarize, the proposed navigation system performs much better than the visual-only navigation system with points and lines and better than the point-only VINS. The reason is that the inertial sensors can aid the visual sensors to improve the performance of the navigation system. What is more, the datasets used in our work are all recorded in man-made environments which contain abundant line features and in some sequences; these environments contain some low-textured senses which lack enough point features. Therefore, the application of line feature can improve the performance of the system to a certain degree.

5.2.3. Qualitative Evaluations of Both Features in Different Environment. Figures 6 and 7 qualitatively illustrate the matched point and line features between two frames in the dark environment and in the environment with some curves.

It is shown in Figure 6 that it is difficult to extract substantial point features and line features due to insufficient visual contents in the dark environment. To be specific, the point features are distributed intensively while the line patterns are scattered sparsely. Despite the significant variance in visual distribution, combining point and line features sufficiently encode the most image textures, which indicates the strong correlation between the two features.

It is observed in Figure 7 that line features can be extracted on the smooth surfaces or bar-shaped objects except some curves with high curvature. By contrast, the point features can be extracted on these curves with high curvature yet fail to be mined on the smooth surfaces or bar-shaped objects. Based on this observation, we argue that there exists substantial complementarity between these two features. Therefore, fusing the point and line features significantly contributes to the performance improvements.

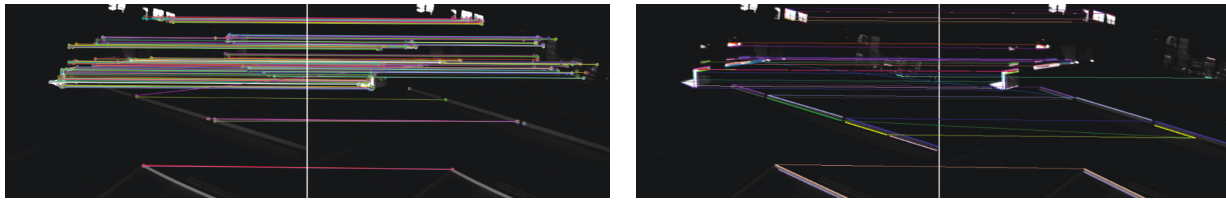
In addition, sufficient correct matches are observed from these two figures, which thus guarantees the estimation accuracy. Subsequently, the estimated position errors of both features can be derived and revised accordingly. Therefore, the accuracy of these positions will be progressively accurate.

6. Conclusions

In this paper, we present an indoor navigation system combining stereo camera and inertial sensors with point and line features. The stereo images and the IMU measurements are fused by the EKF estimator for more accurate and robust pose estimation. Furthermore, the point and line features are all extracted from the stereo images. This scheme can further improve the robustness and accuracy of the system and makes it perform well in complex indoor environment such as low-texture or human-made scenes. Experiments based on eleven sequences provided by the EuRoC dataset are conducted to test the proposed system. The results show that the proposed navigation system performs better than the point-only VINS or the vision-only navigation system with points and lines in indoor environment, especially in more complex and challenging scenarios. So, the proposed navigation system can be applied to the complex indoor environment with high accuracy and robustness.

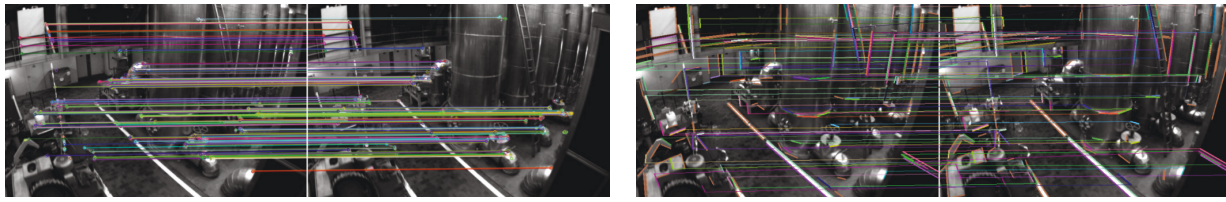
Table 2: Rotation errors (RMSE) in degrees of the EuRoC dataset.

	VINS (point and lines) (°)				MSCKF (°)				StVO-PL (°)			
	Roll	Pitch	Yaw	R	Roll	Pitch	Yaw	R	Roll	Pitch	Yaw	R
MH 01	0.852	0.442	0.768	1.229	0.419	0.250	1.383	1.467	2.452	1.825	4.021	5.051
MH 02	1.349	1.169	1.338	2.231	1.049	2.377	0.676	2.685	1.973	1.636	4.569	5.239
MH 03	0.961	0.507	0.525	1.207	1.084	0.681	1.683	2.071	2.525	4.439	3.963	5.954
MH 04	0.918	1.612	1.734	1.271	0.314	0.869	1.970	2.170	4.936	1.423	3.985	6.503
MH 05	1.658	1.121	1.988	1.411	0.452	0.966	2.179	2.426	1.081	2.621	6.056	6.687
V1 01	1.326	1.267	2.193	2.869	2.234	2.322	0.908	3.348	2.744	2.644	3.130	4.931
V1 02	0.374	0.877	1.301	1.613	0.553	0.371	1.761	1.882	3.138	3.646	3.771	6.113
V1 03	1.544	0.517	1.362	2.123	1.179	0.549	3.041	3.307	2.841	4.004	6.178	7.892
V2 01	0.613	0.489	0.879	1.178	0.631	0.518	0.682	1.064	3.413	2.742	2.808	5.202
V2 02	0.888	0.841	1.138	1.671	0.599	0.697	1.867	2.081	2.096	2.391	5.184	6.081
V2 03	2.087	2.661	2.878	4.441	—	—	—	—	—	—	—	—



(a) (b)

Figure 6: (a) Matched point features between two frames in the dark environment. (b) Matched line features between two frames in the dark environment.



(a) (b)

Figure 7: (a) Matched point features between two frames when there are some curves in environment. (b) Matched line features between two frames when there are some curves in environment.

In addition, the MEMS inertial sensors and the camera are all the low-cost and passive sensors, so, this system can be used in most environments and suitable for the low-cost autonomous navigation system. On the other hand, this system does not include the loop closure detection, so, its accuracy may be no higher than the visual-inertial SLAM system. In addition, we do not consider the features which cannot obtain the scale information by stereo camera because of the far distance, so, the performance in outdoor environment will be worse. In future work, we will further improve this system in these two fields.

Conflicts of Interest

The authors declare no conflict of interest.

Acknowledgments

This study is supported in part by the National Natural Science Foundation of China (Grant nos. 61473085, 51775110, and 51375088), Fundamental Research Funds for the Central Universities (2242015R30031), and key laboratory fund of the Ministry of Public Security based on large data structure (2015DSJSYS002).

References

[1] M. Jung and J.-B. Song, "Robust mapping and localization in indoor environments," *Intelligent Service Robotics*, vol. 10, no. 1, pp. 55–66, 2017.

[2] G. A. Kumar, A. K. Patil, R. Patil, S. S. Park, and Y. H. Chai, "A LiDAR and IMU integrated indoor navigation system for

UAVs and its application in real-time pipeline classification," *Sensors*, vol. 17, no. 6, p. 1268, 2017.

[3] M. W. M. G. Dissanayake, P. Newman, S. Clark, H. F. Durrant-Whyte, and M. Csorba, "A solution to the simultaneous localization and map building (SLAM) problem," *IEEE Transactions on Robotics*, vol. 17, no. 3, pp. 229–241, 2001.

[4] W. Chen and T. Zhang, "An indoor mobile robot navigation technique using odometry and electronic compass," *International Journal of Advanced Robotic Systems*, vol. 14, no. 3, article 1729881417711643, 2017.

[5] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "MonoSLAM: real-time single camera SLAM," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, pp. 1052–1067, 2007.

[6] A. I. Mourikis and S. Roumeliotis, "A multi-state constraint Kalman filter for vision-aided inertial navigation," in *2007 IEEE International Conference on Robotics and Automation*, pp. 3565–3572, Roma, Italy, April 2007.

[7] A. Ben-Afia, V. Gay-Bellile, A.-C. Escher et al., "Review and classification of vision-based localisation techniques in unknown environments," *IET Radar, Sonar & Navigation*, vol. 8, no. 9, pp. 1059–1072, 2014.

[8] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 3, pp. 611–625, 2018.

[9] F. Fraundorfer and D. Scaramuzza, "Visual odometry: part I: the first 30 years and fundamentals," *IEEE Robotics and Automation Magazine*, vol. 18, no. 4, pp. 80–92, 2011.

[10] F. Fraundorfer and D. Scaramuzza, "Visual odometry : part II: matching, robustness, optimization, and applications," *IEEE Robotics and Automation Magazine*, vol. 19, no. 2, pp. 78–90, 2012.

[11] G. Zhang, J. H. Lee, J. Lim, and I. H. Suh, "Building a 3-D line-based map using stereo SLAM," *IEEE Transactions on Robotics*, vol. 31, no. 6, pp. 1364–1377, 2015.

[12] R. Gomez-Ojeda, J. Briales, and J. Gonzalez-Jimenez, "PL-SVO: semi-direct monocular visual odometry by combining points and line segments," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4211–4216, Daejeon, South Korea, October 2016.

[13] G. Zhang and I. Suh, "Building a partial 3D line-based map using a monocular SLAM," in *2011 IEEE International Conference on Robotics and Automation*, pp. 1497–1502, Shanghai, China, May 2011.

[14] L. Clement, V. Peretroukhin, J. Lambert, and J. Kelly, "The battle for filter supremacy: a comparative study of the multi-state constraint Kalman filter and the sliding window filter," in *2015 12th Conference on Computer and Robot Vision (CRV)*, pp. 23–30, Halifax, NS, Canada, June 2015.

[15] M. Li and A. I. Mourikis, "High-precision, consistent EKF-based visual-inertial odometry," *International Journal of Robotics Research*, vol. 32, no. 6, pp. 690–711, 2013.

[16] P. Corke, J. Lobo, and J. Dias, "An introduction to inertial and visual sensing," *The International Journal of Robotics Research*, vol. 26, no. 6, pp. 519–535, 2007.

[17] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale, "Keyframe-based visual–inertial odometry using nonlinear optimization," *The International Journal of Robotics Research*, vol. 34, no. 3, pp. 314–334, 2015.

[18] A. J. Davison, "Real-time simultaneous localisation and mapping with a single camera," in *2003. Proceedings. Ninth IEEE International Conference on Computer Vision*, pp. 1403–1410, Nice, France, October 2003.

[19] L. M. Paz, P. Pinies, J. D. Tardos, and J. Neira, "Large-scale 6-DOF SLAM with stereo-in-hand," *IEEE Transactions on Robotics*, vol. 24, no. 5, pp. 946–957, 2008.

[20] A. J. Davison and D. W. Murray, "Simultaneous localization and map-building using active vision," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 865–880, 2002.

[21] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "ORB-SLAM: a versatile and accurate monocular SLAM system," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[22] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: an open-source SLAM system for monocular, stereo, and RGB-D cameras," *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.

[23] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: an efficient alternative to SIFT or SURF," in *2011 International Conference on Computer Vision*, pp. 2564–2571, Barcelona, Spain, November 2011.

[24] R. G. von Gioi, J. Jakubowicz, J.-M. Morel, and G. Randall, "LSD: a fast line segment detector with a false detection control," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 4, pp. 722–732, 2010.

[25] R. G. von Gioi, J. Jakubowicz, J.-M. Morel, and G. Randall, "LSD: a line segment detector," *Image Processing on Line*, vol. 2, pp. 35–55, 2012.

[26] R. Gomez-Ojeda and J. Gonzalez-Jimenez, "Robust stereo visual odometry through a probabilistic combination of points and line segments," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2521–2526, Stockholm, Sweden, May 2016.

[27] A. Pumarola, A. Vakhitov, A. Agudo, A. Sanfeliu, and F. Moreno-Noguer, "PL-SLAM: real-time monocular visual SLAM with points and lines," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4503–4508, Singapore, Singapore, May 2017.

[28] R. Gomez-Ojeda, D. Zuñiga-Noël, F.-A. Moreno, D. Scaramuzza, and J. Gonzalez-Jimenez, "PL-SLAM: A stereo SLAM system through the combination of points and line segments," 2017, http://arxiv.org/abs/1705.09479.

[29] G. Huang, M. Kaess, and J. Leonard, "Towards consistent visual-inertial navigation," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4926–4933, Hong Kong, China, May 2014.

[30] Z. Xian, X. Hu, and J. Lian, "Fusing stereo camera and low-cost inertial measurement unit for autonomous navigation in a tightly-coupled approach," *Journal of Navigation*, vol. 68, no. 3, pp. 434–452, 2015.

[31] D. Kottas and S. Roumeliotis, "Efficient and consistent vision-aided inertial navigation using line observations," in *2013 IEEE International Conference on Robotics and Automation*, pp. 1540–1547, Karlsruhe, Germany, May 2013.

[32] H. Yu and A. Mourikis, "Vision-aided inertial navigation with line features and a rolling-shutter camera," in *2013 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 892–899, Karlsruhe, Germany, September 2015.

[33] C. Guo and I. Roumeliotis, "IMU-RGBD camera 3D pose estimation and extrinsic calibration: observability analysis

and consistency improvement," in *2013 IEEE International Conference on Robotics and Automation*, pp. 2935–2942, Karlsruhe, Germany, May 2013.

[34] D. H. Titterton and J. L. Weston, *Strapdown Inertial Navigation Technology*, Lavenham Press Ltd, London, UK, 2nd edition, 2004.

[35] J. Rohac, M. Sipos, and J. Simanek, "Calibration of low-cost triaxial inertial sensors," *IEEE Instrumentation & Measurement Magazine*, vol. 18, no. 6, pp. 32–38, 2015.

[36] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330–1334, 2000.

[37] A. Bartoli and P. Sturm, "Structure-from-motion using lines: representation, triangulation and bundle adjustment," *Computer Vision and Image Understanding*, vol. 100, no. 3, pp. 416–441, 2005.

[38] A. Bartoli and P. Sturm, "The 3D line motion matrix and alignment of line reconstructions," *International Journal of Computer Vision*, vol. 57, no. 3, pp. 159–178, 2004.

[39] L. Zhang and R. Koch, "An efficient and robust line segment matching approach based on LBD descriptor and pairwise geometric consistency," *Journal of Visual Communication and Image Representation*, vol. 24, no. 7, pp. 794–805, 2013.

[40] D.-M. Woo, D.-C. Park, S.-S. Han, and S. Beack, "2D line matching using geometric and intensity data," in *2009. AICI '09. International Conference on Artificial Intelligence and Computational Intelligence*, pp. 99–103, Shanghai, China, November 2009.

[41] M. Burri, J. Nikolic, P. Gohl et al., "The EuRoC micro aerial vehicle datasets," *International Journal of Robotics Research*, vol. 35, no. 10, pp. 1157–1163, 2016.