Sensors **2015**, *15*, 12816-12833; doi:10.3390/s150612816

OPEN ACCESS

sensors

ISSN 1424-8220

www.mdpi.com/journal/sensors

Article

Tightly-Coupled Stereo Visual-Inertial Navigation Using Point and Line Features

Xianglong Kong, Wenqi Wu *, Lilian Zhang and Yujie Wang

College of Mechatronics and Automation, National University of Defense Technology, Changsha 410073, China; E-Mails: kongxianglong51@gmail.com (X.K.); lilian-zhang@hotmail.com (L.Z.); yjwang@nudt.edu.cn (Y.W.)

* Author to whom correspondence should be addressed; E-Mail: wenqiwu_lit@hotmail.com; Tel./Fax: +86-731-8457-6305 (ext. 8216).

Academic Editor: Vittorio M.N. Passaro

Received: 13 April 2015 / Accepted: 27 May 2015 / Published: 1 June 2015

Abstract: This paper presents a novel approach for estimating the ego-motion of a vehicle in dynamic and unknown environments using tightly-coupled inertial and visual sensors. To improve the accuracy and robustness, we exploit the combination of point and line features to aid navigation. The mathematical framework is based on trifocal geometry among image triplets, which is simple and unified for point and line features. For the fusion algorithm design, we employ the Extended Kalman Filter (EKF) for error state prediction and covariance propagation, and the Sigma Point Kalman Filter (SPKF) for robust measurement updating in the presence of high nonlinearities. The outdoor and indoor experiments show that the combination of point and line features improves the estimation accuracy and robustness compared to the algorithm using point features alone.

Keywords: vision-aided inertial navigation; point and line features; trifocal geometry; tightly-coupled

1. Introduction

Reliable navigation in dynamic and unknown environments is a key requirement for many applications, particularly for autonomous ground, underwater and air vehicles. The most common sensor modality used to tackle this problem is the Inertial Measurement Unit (IMU). However, inertial

navigation systems (INS) are proved to drift over time due to error accumulation [1]. In the last decades, the topic of vision-aided inertial navigation has received considerable attention in the research community, thanks to some important advantages [2–9]. Firstly, the integrated system can operate in environments where GPS is unreliable or unavailable. Secondly, the complementary frequency responses and noise characteristics of vision and inertial sensors address the respective limitations and deficiencies [10]. In particular, fast and highly dynamic motions can be precisely tracked by an IMU in a short time, and thus the problem of scale ambiguity and large latency in vision can be settled to a certain extent. On the other hand, the low-frequency drift in the inertial measurements can be significantly controlled by visual observations. Furthermore, both cameras and IMUs are low cost, light-weight and low power-consumption devices, which make them ideal for many payload-constrained platforms. Corke [10] has presented a comprehensive introduction of these two sensory modalities from a biological and an engineering perspective.

The simplest fusion scheme for a vision-aided inertial navigation system (VINS) uses separate INS and visual blocks, and fuses information in a loosely-coupled approach [10]. For instance, some methods fuse the inertial navigation solution with the relative pose estimation between consecutive image measurements [11–14]. Tightly-coupled methods in contrast process the raw information of both sensors in a single estimator, thus all the correlations between them are considered, leading to higher accuracy [15,16]. The most common tightly-coupled scheme augments the 3D feature positions in the filter state, and concurrently estimates the motion and structure [2]. However, this method suffers from high computational complexity, as the dimension of the state vector increases with the number of the observed features. To address this problem, Mourikis [15] proposed an EKF-based algorithm which maintains a sliding window of poses in the filter state, and make use of the tracked features to impose constraints on these poses. The shortcomings of this approach are twofold: (1) the space complexity is high, because it needs to store all the tracked features; (2) it requires a reconstruction of the 3D position of the tracked feature points, which are not necessary in navigation tasks. To overcome these shortcomings, Hu [9] developed a sliding window odometry using the monocular camera geometry constraints among three images as measurements, resulting in a tradeoff between accuracy and computational cost.

While the vision-aided inertial navigation has been extensively studied, and a considerable amount of work has also been dedicated to processing visual observations of point features [2,4,5,7], on the contrary, much less work has been aimed at exploring line features. In fact, line primitives and point primitives provide complementary information about the image [17]. There are many scenes (e.g., wall corners, stairwell edges, *etc.*) where the point primitive matches are unreliable while the line primitives are well matched, due to multi-pixel support [6].

On the other hand, points are crucial as they give more information than lines. For instance, there are no pose constraints imposed by line correspondences from two views, while there are well-known epipolar geometry constraints for point correspondences from two views [18].

In this paper, we propose a method that combines point and line features for navigation aiding in a simple and unified framework. Our algorithm can deal with any mixed combination of point and line correspondences utilizing trifocal geometry across two stereo views. In the implementation, the inertial sensors are tightly-coupled within feature tracking to improve the robustness and tracking speed. Meanwhile, the drifts of inertial sensors are greatly reduced by using the constraints imposed in the

tracked features. Leveraging both of the complementary characteristics of the inertial and visual sensors and the complementary characteristics between point and line features, the proposed algorithm demonstrates improved performance and robustness.

The remainder of this paper is organized as follows: we describe the mathematical model of the VINS in Section 2, and then develop our estimator in Section 3. Experimental results are given in Section 4. Finally, Section 5 contains some conclusions and suggests several directions for future work.

2. Mathematical Formulation

2.1. Notations and Convention

We denote scalars in italic lower case letters (e.g., a), denote vectors in lower case letters with boldface non-italic (e.g., $\mathbf{p}$), and denote matrices in upper case letters with bold font (e.g., $\mathbf{R}$). If a vector or matrix describes the relative pose of one reference frame with respect to another, we combine subscript letters to designate the frames, e.g., $\mathbf{p}_{WI}$ represents the translation vector from the origin of the frame $\{W\}$ to the origin of the frame $\{I\}$, and $\mathbf{R}_{WI}$ represents the direction cosine matrix of frame $\{I\}$ in the reference frame $\{W\}$. The six degrees of freedom transform between two reference frames can be represented as a translation followed by a rotation:

$$\mathbf{t}^W = \mathbf{p}_{WI}^W + \mathbf{R}_{WI}\mathbf{t}^I \tag{1}$$

In the remaining Sections, unit quaternions are also used to describe the relative orientation of two reference frames, e.g., $\bar{q}_{WI}$ represents the orientation of frame $\{I\}$ in frame $\{W\}$.

Finally, to represent projective geometry, it is simpler and more symmetric to introduce homogeneous coordinates, which provides a scale invariant representation for point and line in the Euclidean plane. In this paper, vectors in homogeneous coordinate form are expressed by an underline, e.g., $\underline{\mathbf{m}} = \begin{pmatrix} u & v & w \end{pmatrix}^T$ represents the point $\mathbf{m} = \begin{pmatrix} u' & v' \end{pmatrix}^T$ in the Euclidean plane, with $u' = u/w, v' = v/w, w \neq 0$.

2.2. System Model

The evolving IMU state is described by the vector:

$$\mathbf{x}_{IMU}(t) = \begin{bmatrix} \left(\mathbf{p}_{WI}^W\right)^T & \left(\bar{q}_{WI}\right)^T & \left(\mathbf{v}_{WI}^W\right)^T & \left(\mathbf{b}_g(t)\right)^T & \left(\mathbf{b}_a(t)\right)^T \end{bmatrix}^T \tag{2}$$

where $\mathbf{p}_{WI}^W(t)$ denotes the position of IMU in the world frame $\{W\}$; $\bar{q}_{WI}(t)$ is the unit quaternion of the IMU frame $\{I\}$ in the world frame; $\mathbf{v}_{WI}^W$ is the linear velocity of the IMU in the world frame; $\mathbf{b}_g(t)$ and $\mathbf{b}_a(t)$ are the IMU gyroscope and accelerometer biases, respectively.

In this work, we model the biases $\mathbf{b}_g(t)$ and $\mathbf{b}_a(t)$ as a Gaussian random walk process, driven by the white, zero-mean noise vectors $\mathbf{n}_{gw}$ and $\mathbf{n}_{aw}$, with covariance matrices $\mathbf{Q}_{gw}$ and $\mathbf{Q}_{aw}$ respectively. The time evolution of the IMU state is given by the following equation [2]:

$$\dot{\mathbf{p}}_{WI}^W = \mathbf{v}_{WI}^W, \quad \dot{\bar{q}}_{WI} = \frac{1}{2}\Omega\left(\boldsymbol{\omega}_{WI}^I\right)\bar{q}_{WI}, \quad \dot{\mathbf{v}}_{WI}^W = \mathbf{a}_{WI}^W, \quad \dot{\mathbf{b}}_g = \mathbf{n}_{gw}, \quad \dot{\mathbf{b}}_a = \mathbf{n}_{aw} \tag{3}$$

where $\Omega\left(\boldsymbol{\omega}_{WI}^{I}\right)$ is the quaternion multiplication matrix:

$$\Omega\left(\boldsymbol{\omega}_{WI}^{I}\right)=\begin{bmatrix} 0 & -\left(\boldsymbol{\omega}_{WI}^{I}\right)^{T} \\ \boldsymbol{\omega}_{WI}^{I} & -\left[\boldsymbol{\omega}_{WI}^{I}\times\right] \end{bmatrix} \tag{4}$$

which relates the time rate of change of the unit quaternion to the angular velocity; $\boldsymbol{\omega}_{WI}^{I}$ is the angular velocity of the IMU with respect to the world frame, and $\mathbf{a}_{WI}^{W}$ is the acceleration of the IMU with respected to the world frame expressed in the world frame. The measured angular velocity and linear acceleration from are:

$$\boldsymbol{\omega}_{m}=\boldsymbol{\omega}_{WI}^{I}+\mathbf{b}_{g}+\mathbf{n}_{g} \tag{5}$$

$$\mathbf{a}_{m}=\mathbf{R}^{T}\left(\overline{q}_{WI}\right)\left(\mathbf{a}_{WI}^{W}-\mathbf{g}^{W}\right)+\mathbf{b}_{a}+\mathbf{n}_{a} \tag{6}$$

where $\mathbf{R}\left(\overline{q}_{WI}\right)$ is the direction cosine matrix corresponding to the unit quaternion $\overline{q}_{WI}$, $\mathbf{n}_{g}$ and $\mathbf{n}_{a}$ are measurements noises of gyroscope and accelerometer, which are assumed to be zero-mean Gaussian noise with covariance matrices $\mathbf{Q}_{g}$ and $\mathbf{Q}_{a}$, respectively. Note that we do not consider the Earth's rotation rate in the gyroscope measurement, because it is small enough relative to the noise and bias of the low-cost gyroscope.

2.3. Measurement Model

2.3.1. Camera Model

In this Section, we consider the standard perspective camera model, which is commonly used in the computer vision applications. Let $\mathbf{K}$ denote the intrinsic camera parameters matrix which can be obtained by calibrating. A mapping between the 3D homogeneous point $\underline{\mathbf{M}}=\begin{bmatrix} M_1 & M_2 & M_3 & M_4 \end{bmatrix}^{T}$ in space and the homogeneous image pixel coordinates $\underline{\mathbf{m}}=\begin{bmatrix} u & v & 1 \end{bmatrix}^{T}$ can be given by:

$$\underline{\mathbf{m}}\propto\mathbf{K}\cdot\left[\mathbf{R}\,|\,\mathbf{t}\right]\underline{\mathbf{M}}=\mathbf{P}\underline{\mathbf{M}} \tag{7}$$

where $\propto$ means equality up to scale, and $\mathbf{P}=\mathbf{K}\cdot\left[\mathbf{R}\,|\,\mathbf{t}\right]$ is 3 × 4 camera matrix, with $\mathbf{R}$ and $\mathbf{t}$ representing pose of the camera with respect to the world reference frame. Similarly, a mapping between a 3-space line represented as a Plücker matrix $\mathbf{L}$ and the homogenous image line $\mathbf{l}$ is given by [18]:

$$\left[\mathbf{l}\times\right]=\mathbf{PLP}^{T} \tag{8}$$

2.3.2. Review of the Trifocal Tensor

A trifocal tensor is a 3 × 3 × 3 array of numbers that describes the geometric relations among three views. It depends only on the relative motion between the different views and is independent of scene structures. Assuming that the camera matrices of three views are $\mathbf{P}_{1}=\left[\mathbf{I}\,|\,\mathbf{0}\right]$, $\mathbf{P}_{2}=\left[\mathbf{A}\,|\,\mathbf{a}_{4}\right]$, $\mathbf{P}_{3}=\left[\mathbf{B}\,|\,\mathbf{b}_{4}\right]$, the entries of the trifocal tensor can be derived accordingly using the standard matrix-vector notation [18]:

$$\mathbf{T}_i = \mathbf{a}_i \mathbf{b}_4^T - \mathbf{a}_4 \mathbf{b}_i^T \tag{9}$$

where $\mathbf{a}_i$ and $\mathbf{b}_i$ denote the i-th column of the camera matrices $\mathbf{P}_2$ and $\mathbf{P}_3$, respectively.

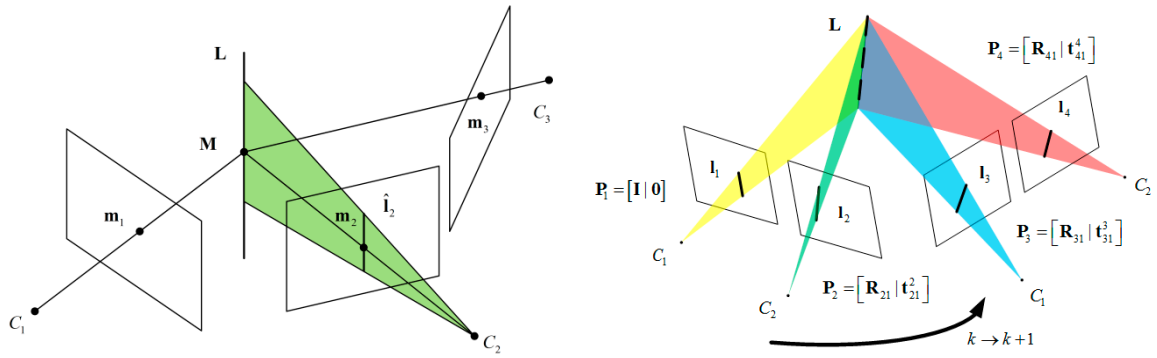

Figure 1. (**a**) The point-line-point correspondence among three views; (**b**) Stereo geometry for two views and line-line-line configuration.

Once the trifocal tensor is computed, we can use of it to map a pair of matched points $\underline{\mathbf{m}}_1 \leftrightarrow \underline{\mathbf{m}}_2$ in the first and second views into the third view, using the homography between the first view and the third view induced by a line in the second image [18]. As shown in Figure 1a, a line in second view defines a plane in space, and this plane induces a homography between the first view and third view. As recommended by Hartley [18], the line $\hat{\mathbf{l}}_2$ is chosen as the line perpendicular to the epipolar line. The transfer procedure is summarized as follows [18]:

(1) Compute the epipolar line $\mathbf{l}_e = \mathbf{F}_{21}\underline{\mathbf{m}}_1$, where $\mathbf{F}_{21}$ is the fundamental matrix between the first and second views.

(2) Compute the line $\hat{\mathbf{l}}_2$ which passes through $\underline{\mathbf{m}}_2$ and is perpendicular to $\mathbf{l}_e$. If $\mathbf{l}_e = \begin{bmatrix} l_{e1} & l_{e2} & l_{e3} \end{bmatrix}^T$ and $\underline{\mathbf{m}}_2 = \begin{bmatrix} m_{21} & m_{22} & 1 \end{bmatrix}^T$, then $\hat{\mathbf{l}}_2 = \begin{bmatrix} l_{e2} & -l_{e1} & -m_{21}l_{e2} + m_{22}l_{e1} \end{bmatrix}^T$.

(3) The transferred point is $\hat{\underline{\mathbf{m}}}_3 = \left(\sum_i m_{1i} \mathbf{T}_i^T \right) \hat{\mathbf{l}}_2$.

Similarly, it is possible to transfer a pair of matched lines $\mathbf{l}_2 \leftrightarrow \mathbf{l}_3$ in the second and third views into the first view according to the line transfer equation [18]:

$$\hat{l}_{1i} = \mathbf{l}_2^T \mathbf{T}_i \mathbf{l}_3 \tag{10}$$

2.3.3. Stereo Vision Measurement Model via Trifocal Geometry

In this Section, we exploit the trifocal geometry of stereo vision to deduce the measurement model. We depict the stereo camera configuration of two consecutive frames in Figure 1b. For the sake of clarity, we only provide the geometrical relations of lines. The camera matrices of the stereo image pair at the previous time step can be represented in canonical form as:

$$\mathbf{P}_1 = \begin{bmatrix} \mathbf{I} \mid \mathbf{0} \end{bmatrix}, \mathbf{P}_2 = \begin{bmatrix} \mathbf{R}_{21} \mid \mathbf{t}_{21}^2 \end{bmatrix} \tag{11}$$

where $\mathbf{R}_{21} = \mathbf{R}_0$ and $\mathbf{t}_{21}^2 = \mathbf{t}_0$ encode the rigid transform of the rig which are known after calibration. The camera matrices of the successive stereo image pairs are defined as:

$$\mathbf{P}_3 = \left[\mathbf{R}_{31} \mid \mathbf{t}_{31}^3\right], \mathbf{P}_4 = \left[\mathbf{R}_{41} \mid \mathbf{t}_{41}^4\right] \tag{12}$$

For simplicity, we assume that the IMU frame of reference coincides with the camera frame of reference. Thus, the terms in Equation (12) can be expressed as follows:

$$\mathbf{R}_{31} = \mathbf{R}_{WI}^T \mathbf{R}_{WI_1} \tag{13}$$

$$\mathbf{t}_{31}^3 = \mathbf{R}_{WI}^T \left(\mathbf{p}_{WI_1}^W - \mathbf{p}_{WI}^W\right) \tag{14}$$

$$\mathbf{R}_{41} = \mathbf{R}_{43}\mathbf{R}_{31} = \mathbf{R}_0 \mathbf{R}_{WI}^T \mathbf{R}_{WI_1} \tag{15}$$

$$\mathbf{t}_{41}^4 = \mathbf{t}_{43}^4 + \mathbf{R}_{43}\mathbf{t}_{31}^3 = \mathbf{t}_0 + \mathbf{R}_0 \mathbf{R}_{WI}^T \left(\mathbf{p}_{WI_1}^W - \mathbf{p}_{WI}^W\right) \tag{16}$$

where $\mathbf{R}_{WI_1}$ and $\mathbf{p}_{WI_1}^W$ are the pose of IMU corresponding to the last time the image pair captured.

Two trifocal tensors, $\mathcal{T}_L = \left\{\mathbf{T}_i^L\right\}$ relating the previous image pair to the current left image and $\mathcal{T}_R = \left\{\mathbf{T}_i^R\right\}$ relating to the current right image can be determined according to Equation (9) using the camera matrices Equations (11) and (12):

$$\mathcal{T}_L = \mathcal{T}\left(\mathbf{P}_1, \mathbf{P}_2, \mathbf{P}_3\right) = \mathcal{T}\left(\mathbf{R}_0, \mathbf{t}_0, \mathbf{R}_{WI}, \mathbf{p}_{WI}^W, \mathbf{R}_{WI_1}, \mathbf{p}_{WI_1}^W\right) \tag{17}$$

$$\mathcal{T}_L = \mathcal{T}\left(\mathbf{P}_1, \mathbf{P}_2, \mathbf{P}_4\right) = \mathcal{T}\left(\mathbf{R}_0, \mathbf{t}_0, \mathbf{R}_{WI}, \mathbf{p}_{WI}^W, \mathbf{R}_{WI_1}, \mathbf{p}_{WI_1}^W\right) \tag{18}$$

From the corresponding point set $\left\{\mathbf{m}_1 \leftrightarrow \mathbf{m}_2 \leftrightarrow \mathbf{m}_3 \leftrightarrow \mathbf{m}_4\right\}$ and the point transfer relations among the triplets, the following non-linear functions can be defined:

$$h_1\left(\mathcal{T}_L\left(\mathbf{R}_0, \mathbf{t}_0, \mathbf{R}_{WI}, \mathbf{p}_{WI}^W, \mathbf{R}_{WI_1}, \mathbf{p}_{WI_1}^W\right), \mathbf{m}_1, \mathbf{m}_2, \mathbf{m}_3\right) = \mathbf{0}_{2\times 1} \tag{19}$$

$$h_1\left(\mathcal{T}_R\left(\mathbf{R}_0, \mathbf{t}_0, \mathbf{R}_{WI}, \mathbf{p}_{WI}^W, \mathbf{R}_{WI_1}, \mathbf{p}_{WI_1}^W\right), \mathbf{m}_1, \mathbf{m}_2, \mathbf{m}_4\right) = \mathbf{0}_{2\times 1} \tag{20}$$

where $h_1(\cdot)$ denotes the pixel differences between the transferred point and the measured point.

For line measurements, we also need a formulation to compare the transferred lines with the measured lines. Because of the aperture problem [19], only the measurement components which are orthogonal to the transferred line can be used for correction. In [3,17], the line-point is chosen as observation, which is defined as the closest point on the line segment to the image origin. Accordingly, the error function is defined as the differences between the measured and transferred line-points, which is similar to the error function of point features. However, when the lines pass through the origin, the orientation error of the lines cannot be revealed by this error function. Thus, we choose the signed distances between the endpoints of the measured line segment to the transferred line as observation. Suppose that $\underline{\mathbf{s}}^a$ and $\underline{\mathbf{s}}^b$ are the end points of the line segment measured in the first view. We denote the line transferred from the second and third views by $\hat{\mathbf{l}} = \left(\hat{l}_1, \hat{l}_2, \hat{l}_3\right)^T$. The signed distances between the end points of the measured line segment and the transferred line make up the line observation function:

$$\mathbf{d}_1 = \begin{bmatrix} \underline{\mathbf{s}}^a \cdot \hat{\mathbf{l}} / \sqrt{\left(\hat{l}_1\right)^2 + \left(\hat{l}_2\right)^2} \\ \underline{\mathbf{s}}^b \cdot \hat{\mathbf{l}} / \sqrt{\left(\hat{l}_1\right)^2 + \left(\hat{l}_2\right)^2} \end{bmatrix} = \mathbf{0}_{2\times 1} \tag{21}$$

Similarly, the line observation function concerning the first, second, and fourth views is defined as:

$$\mathbf{d}_2 = \begin{bmatrix} \underline{\mathbf{s}}^a \cdot \hat{\mathbf{l}}' / \sqrt{\left(\hat{l}_1'\right)^2 + \left(\hat{l}_2'\right)^2} \\ \underline{\mathbf{s}}^b \cdot \hat{\mathbf{l}}' / \sqrt{\left(\hat{l}_1'\right)^2 + \left(\hat{l}_2'\right)^2} \end{bmatrix} = \mathbf{0}_{2\times 1} \tag{22}$$

where $\hat{\mathbf{l}}' = \left(\hat{l}_1', \hat{l}_2', \hat{l}_3'\right)^T$ is the line transferred from the second and fourth views.

As we process the point and line measurements in a unified manner after defining the corresponding error, we define the observation model in a single function:

$$\mathbf{z} = h\left(\mathcal{T}_L\left(\mathbf{R}_0, \mathbf{t}_0, \mathbf{R}_{WI}, \mathbf{p}_{WI}^W, \mathbf{R}_{WI_1}, \mathbf{p}_{WI_1}^W\right), \mathcal{T}_R\left(\mathbf{R}_0, \mathbf{t}_0, \mathbf{R}_{WI}, \mathbf{p}_{WI}^W, \mathbf{R}_{WI_1}, \mathbf{p}_{WI_1}^W\right), \left\{\mathbf{f}_1, \mathbf{f}_2, \mathbf{f}_3, \mathbf{f}_4\right\}\right) = \mathbf{0}_{4\times 1} \tag{23}$$

where $\left\{\mathbf{f}_1, \mathbf{f}_2, \mathbf{f}_3, \mathbf{f}_4\right\}$ denotes the general feature correspondences among the four views, $\mathcal{T}_L$ and $\mathcal{T}_R$ encode the motion information between the successive stereo image pairs, and the function $h(\cdot)$ defines the observations based on the feature type.

3. Estimator Description

3.1. Structure of the State Vector

As can be seen in the previous Section, the measurement models are implicit relative-pose measurements, which relate the system state at two different time instants (*i.e.*, the current time and the previous time when image pair is captured). However, the "standard" Kalman filter formulation requires that the measurements employed for the state update be independent of any previous filter states. The problem can be addressed by augment the state vector to include a history of IMU pose when last image pair is recorded. With these state augmentations, the measurements are only related to the current state, and thus, a Kalman filter framework can be applied. The augmented nominal state is given by:

$$\hat{\mathbf{x}} = \begin{bmatrix} \hat{\mathbf{x}}_{IMU}^T & \left(\hat{\mathbf{p}}_{WI_1}^W\right)^T & \left(\hat{\bar{q}}_{I_1}^W\right)^T \end{bmatrix}^T \tag{24}$$

where $\hat{\mathbf{x}}_{IMU}^T$ is the nominal state of IMU, which can be obtained by integrating Equation (3) without considering the noise term; $\left(\hat{\mathbf{p}}_{WI_1}^W\right)^T$ and $\left(\hat{\bar{q}}_{I_1}^W\right)^T$ denotes the nominal-state pose of the IMU at time when the last image pair is recorded. The augmented error state is defined accordingly:

$$\delta\mathbf{x} = \begin{bmatrix} \delta\mathbf{x}_{IMU} & \left(\delta\mathbf{p}_{WI_1}^W\right)^T & \left(\delta\boldsymbol{\theta}^{I_1}\right)^T \end{bmatrix}^T \tag{25}$$

where $\delta\mathbf{x}_{IMU}$ is the IMU error-state defined as:

$$\delta\mathbf{x}_{IMU} = \begin{bmatrix} \left(\delta\mathbf{p}_{WI}^{W}\right)^{T} & \left(\delta\boldsymbol{\theta}^{I}\right)^{T} & \left(\delta\mathbf{v}_{WI}^{W}\right)^{T} & \delta\mathbf{b}_{g}^{T} & \delta\mathbf{b}_{a}^{T} \end{bmatrix}^{T} \tag{26}$$

The standard additive error definition is used for the position, velocity and biases, while for the orientation error $\delta\boldsymbol{\theta}^{I}$, the multiplicative error definition is applied:

$$\bar{q}_{I}^{W} = \hat{\bar{q}}_{I}^{W} \otimes \begin{bmatrix} 1 & \frac{1}{2}\left(\delta\boldsymbol{\theta}^{I}\right)^{T} \end{bmatrix}^{T} \tag{27}$$

where the symbol $\otimes$ denotes quaternion multiplication. With the above error definition, the true-state may be expressed as a suitable composition of the nominal and the error-states:

$$\mathbf{x} = \hat{\mathbf{x}} \oplus \delta\mathbf{x} \tag{28}$$

where $\oplus$ means a generic composition.

3.2. Filter Propagation

The continuous-time IMU error-state model may be given as a single matrix error equation:

$$\delta\dot{\mathbf{x}}_{IMU} = \mathbf{F}_{IMU}\delta\mathbf{x}_{IMU} + \mathbf{G}_{IMU}\mathbf{n}_{IMU} \tag{29}$$

where:

$$\mathbf{F}_{IMU} = \begin{bmatrix} \mathbf{0}_{3\times3} & \mathbf{I}_{3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & -\left\lfloor\left(\boldsymbol{\omega}_{WI}^{I}(t) - \mathbf{b}_{g}(t)\right)\times\right\rfloor & \mathbf{0}_{3\times3} & -\mathbf{I}_{3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & -\mathbf{R}\left(\bar{q}_{WI}(t)\right)\left\lfloor\left(\mathbf{a}_{m}(t) - \mathbf{b}_{a}(t)\right)\times\right\rfloor & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & -\mathbf{R}\left(\bar{q}_{WI}(t)\right) \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \end{bmatrix} \tag{30}$$

$$\mathbf{G}_{IMU} = \begin{bmatrix} \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ -\mathbf{I}_{3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & -\mathbf{R}\left(\bar{q}_{WI}(t)\right) & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{I}_{3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{I}_{3} \end{bmatrix} \tag{31}$$

$$\mathbf{n}_{IMU} = \begin{bmatrix} \left(\mathbf{n}_{g}\right)^{T} & \left(\mathbf{n}_{a}\right)^{T} & \left(\mathbf{n}_{gw}\right)^{T} & \left(\mathbf{n}_{aw}\right)^{T} \end{bmatrix}^{T} \tag{32}$$

Since the past pose is unchanged during the filter prediction step, its corresponding derivatives are zero:

$$\dot{\hat{\mathbf{p}}}_{WI_1}^{W} = \mathbf{0}, \dot{\hat{\bar{q}}}_{I_1}^{W} = \mathbf{0} \tag{33}$$

$$\delta\dot{\mathbf{p}}_{WI_1}^{W} = \mathbf{0}, \delta\dot{\boldsymbol{\theta}}^{I_1} = \mathbf{0} \tag{34}$$

Combining Equations (29) and (34), the continuous-time augmented error state equation is given by:

$$\delta\dot{\mathbf{x}} = \mathbf{F}_{c}\delta\mathbf{x} + \mathbf{G}_{c}\mathbf{n}_{IMU} \tag{35}$$

where:

$$\mathbf{F}_c = \begin{bmatrix} \mathbf{F}_{IMU} & \mathbf{0}_{15\times 6} \\ \mathbf{0}_{6\times 15} & \mathbf{0}_{6\times 6} \end{bmatrix} \tag{36}$$

$$\mathbf{G}_c = \begin{bmatrix} \mathbf{G}_{IMU} \\ \mathbf{0}_{6\times 6} \end{bmatrix} \tag{37}$$

where $\mathbf{F}_{IMU}$ and $\mathbf{G}_{IMU}$ are defined in Equations (30) and (31).

Each time a new IMU measurement is received, the nominal state prediction is performed by numerical integration of the kinematic Equations (3) and (33). In order to obtain the error covariance, we compute the discrete-time state transition matrix:

$$\mathbf{\Phi}_k = \mathbf{\Phi}(t_{k+1}, t_k) = \exp\left(\int_{t_k}^{t_{k+1}} \mathbf{F}_c(\tau)\, d\tau\right) \tag{38}$$

The elements of $\mathbf{\Phi}_k$ can be computed analytically following similar derivation as [20]. The state transition matrix is slightly different from [21], because the rates of filter prediction and filter update are different in our case.

The noise covariance matrix $\mathbf{Q}_d$ of the discrete-time system is evaluated by:

$$\mathbf{Q}_d = \int_{t_k}^{t_{k+1}} \mathbf{\Phi}(t_{k+1}, \tau)\mathbf{G}_c\mathbf{Q}_c\mathbf{G}_c^T\mathbf{\Phi}^T(t_{k+1}, \tau) d\tau \tag{39}$$

The predicted covariance is then obtained as:

$$\mathbf{P}_{k+1|k} = \mathbf{\Phi}_k\mathbf{P}_{k|k}\mathbf{\Phi}_k^T + \mathbf{Q}_d \tag{40}$$

3.3. Measurement Update

Since the measurement model is highly nonlinear, we employ statistical linearization for measurement updating, which is generally more accurate than the first order Taylor series expansion [22]. Specifically, the Sigma Point approach is applied. First, the following sets of sigma points are selected:

$$\begin{aligned} \mathcal{X}^{(0)} &= \mathbf{0}_{27\times 1}, \\ \mathcal{X}^{(i)} &= \left(\sqrt{(n+\lambda)\mathbf{P}_{k+1|k}}\right)_i, \quad i = 1, \ldots, n, \\ \mathcal{X}^{(i)} &= -\left(\sqrt{(n+\lambda)\mathbf{P}_{k+1|k}}\right)_i, \quad i = n+1, \ldots, 2n \end{aligned} \tag{41}$$

where $n = 21$ is the dimension of the state, the parameter $\lambda = \alpha^2(n+\kappa) - n$ with tuning parameters α, κ, $\left(\sqrt{\mathbf{P}}\right)_i$ indicates the i th column of the matrix square-root of the covariance matrix $\mathbf{P}$. We define the following weights for the sigma points:

$$\begin{aligned} W_m^{(0)} &= \frac{\lambda}{\lambda+n}, \\ W_c^{(0)} &= \frac{\lambda}{\lambda+n} + \left(1-\alpha^2+\beta\right), \\ W_m^{(i)} &= W_c^{(i)} = \frac{1}{2(\lambda+n)}, \quad i = 1, 2, \ldots, 2n \end{aligned} \tag{42}$$

where β is related to the higher order moments of the distribution [23] (a good starting guess is $\beta = 2$ for Gaussian distribution).

The predicted measurement vector is determined by propagating individual sigma point through the nonlinear observation function $h(\cdot)$ defined in Equation (23):

$$\mathcal{Z}_j^{(i)} = h\left(\mathcal{T}_L\left(\hat{\mathbf{x}}_{k+1} \oplus \mathcal{X}^{(i)}\right), \mathcal{T}_R\left(\hat{\mathbf{x}}_{k+1} \oplus \mathcal{X}^{(i)}\right), \left\{\mathbf{f}_1, \mathbf{f}_2, \mathbf{f}_3, \mathbf{f}_4\right\}_j\right) \tag{43}$$

The mean and covariance are computed as:

$$\hat{\mathcal{Z}}_j = \sum_{i=0}^{i=2L} W_m^{(i)} \mathcal{Z}_j^{(i)} \tag{44}$$

$$\mathbf{P}^{z_j z_j} = \sum_{i=0}^{i=2L} W_c^{(i)} \left[\mathcal{Z}_j^{(i)} - \hat{\mathcal{Z}}_j\right]\left[\mathcal{Z}_j^{(i)} - \hat{\mathcal{Z}}_j\right]^T \tag{45}$$

$$\mathbf{P}^{xz_j} = \sum_{i=0}^{i=2L} W_c^{(i)} \left[\mathcal{X}^{(i)} - \mathbf{0}_{27\times 1}\right]\left[\mathcal{Z}_j^{(i)} - \hat{\mathcal{Z}}_j\right]^T \tag{46}$$

where $\mathbf{P}^{z_j z_j}$ and $\mathbf{P}^{xz_j}$ are the predicted measurement covariance matrix and the state-measurement cross-covariance matrix, respectively.

The filter gain is given as follows:

$$\mathbf{K}_j = \mathbf{P}^{xz_j}\left(\mathbf{P}^{z_j z_j} + \mathbf{R}_j\right)^{-1} \tag{47}$$

where $\mathbf{R}_j$ is the measurement noise covariance matrix.

Then, the error state and error covariance are updated using the normal Kalman filter equation:

$$\delta\mathbf{x}_{k+1|k+1} = \delta\mathbf{x}_{k+1|k} + \mathbf{K}_j\left(\mathbf{0} - \hat{\mathcal{Z}}_j\right) \tag{48}$$

$$\mathbf{P}_{k+1|k+1} = \mathbf{P}_{k+1|k} - \mathbf{K}_j \mathbf{P}^{z_j z_j} \mathbf{K}_j^T \tag{49}$$

After measurement update, the estimated state $\delta\mathbf{x}_{k+1|k+1}$ is then used to correct nominal state $\hat{\mathbf{x}}_{k+1}$.

Finally, replace old state by current state and revise the corresponding error covariance:

$$\hat{\mathbf{x}}_{k+1} = \mathbf{T}_n \hat{\mathbf{x}}_{k+1}, \delta\mathbf{x}_{k+1} = \mathbf{T}_e \delta\mathbf{x}_{k+1}, \mathbf{P}_{k+1|k+1} = \mathbf{T}_e \mathbf{P}_{k+1|k+1} \mathbf{T}_e^T \tag{50}$$

with:

$$\mathbf{T}_n = \begin{bmatrix} \mathbf{I}_7 & \mathbf{0}_{7\times 9} & \mathbf{0}_{7\times 7} \\ \mathbf{0}_{9\times 7} & \mathbf{I}_9 & \mathbf{0}_{9\times 7} \\ \mathbf{I}_7 & \mathbf{0}_{7\times 9} & \mathbf{0}_{7\times 7} \end{bmatrix}, \mathbf{T}_e = \begin{bmatrix} \mathbf{I}_6 & \mathbf{0}_{6\times 9} & \mathbf{0}_{6\times 6} \\ \mathbf{0}_{9\times 6} & \mathbf{I}_9 & \mathbf{0}_{9\times 6} \\ \mathbf{I}_6 & \mathbf{0}_{6\times 9} & \mathbf{0}_{6\times 6} \end{bmatrix} \tag{51}$$

4. Experimental Results and Discussion

4.1. Outdoor Experiment

We evaluate the proposed method using the publicly available KITTI Vision Benchmark Suite [24], which provides several multi-sensor datasets with ground truth. The selected dataset was captured in a residential area from the experimental vehicle, equipped with a GPS/IMU localization unit with RTK

correction signals (OXTS RT3003), and a stereo rig with two grayscale cameras (PointGrey Flea2). The duration is about 440 s, with a traveling distance of about 3600 m, and the average speed is about 29 km/h. All the sensors are rigidly mounted on top of the vehicle. The intrinsic parameters of the cameras and the transformation between the cameras and GPS/IMU were well calibrated. Moreover, the cameras and GPS/IMU are manually synchronized, with sampling rates of 10 Hz and 100 Hz, respectively.

The announced gyroscope and accelerometer bias specifications are 36 deg/h (1σ) and 1 mg (1σ), respectively. The resolution of stereo images is 1226 × 370 pixels, with 90° field view. For the position ground truth, we use the trajectory of the GPS/IMU output, with open sky localization errors less than 5 cm.

4.1.1. Feature Detection, Tracking, and Outlier Rejection

For point features, the fast corner detection (FAST) algorithm [25] was used for feature extraction, and matching was carried out by normalized cross-correlation. The main advantage of the FAST detector compared to others is the better trade-off between accuracy and efficiency. In order to reduce the computational complexity and to guarantee the well distribution of the image features, we choose a subset of the matched point features by means of bucketing [26]: Divide the image into several non-overlapping rectangles, and maintain a maximal number of feature points in each rectangle.

We extract lines using EDlines detector [27] in the scale space, which can give accurate results in real-time. Then we employ the method described in [28] for line matching. The lines are described local appearances by the so-called Line Band Descriptor (LBD) similar to SIFT [29] for point features, and are matched by exploiting the local appearance similarities and geometric consistencies [28]. The average execution time of line matching between views is about 56 ms with Intel Core i5 2.6 GHz processors running the non-optimized C++ code. Figure 2 shows a sample image from the dataset with extracted points and lines. As can be seen, both point and line features are rich in the selected sequence.

In order to reject mismatched features and features located on independently moving objects (e.g., the running car), we employ a chi-square test [30] for the measurement residuals. We compute the Mahalanobis distance:

$$\upsilon_j = \left(\mathbf{0} - \hat{Z}_j\right)^T \left(\mathbf{P}^{z_j z_j} + \mathbf{R}_j\right)^{-1} \left(\mathbf{0} - \hat{Z}_j\right) \tag{52}$$

where $\left(\mathbf{0} - \hat{Z}_j\right)$ is the measurement residual, and $\mathbf{P}^{z_j z_j} + \mathbf{R}_j$ is the covariance of the measurement residual. The rejection threshold is usually chosen by an empirical evaluation of reliability of feature matching. We set the threshold to 12 in the experiment. The feature measurements whose residuals exceed the threshold are discarded.

Figure 2. Sample image with extracted point (red) and line (green) features.

4.1.2. Experimental Results

In this Section, we compare the performance of our algorithm with the following methods: (1) GPS/IMU localization unit, with open sky localization errors less than 5 cm; (2) VINS using only point feature; (3) pure inertial-only navigation solution; (4) pure stereo visual odometry [31].

The trajectory estimation results of different algorithms with the ground truth data are shown in Figure 3. The corresponding 3D position errors are depicted in Figure 4. The overall root-mean-square errors (RMSE) are shown in Table 1. It can be found that the inertial-only navigation suffers from error accumulation and is not reliable for long-term operation; Secondly, the result of pure stereo visual odometry is inferior, specially where the vehicle turns, and the error grows super-linearly owing to the inherent bias in stereo visual odometry; Thirdly, the combining of inertial navigation and stereo vision with point feature alone can reduce the drift rate effectively, and the additional information from line measurements results in better performance. Note that the jumps from 80 s to 100 s are caused by ground truth errors. It also shows the advantage of the proposed method in cluttered urban environments where the GPS information is less reliable.

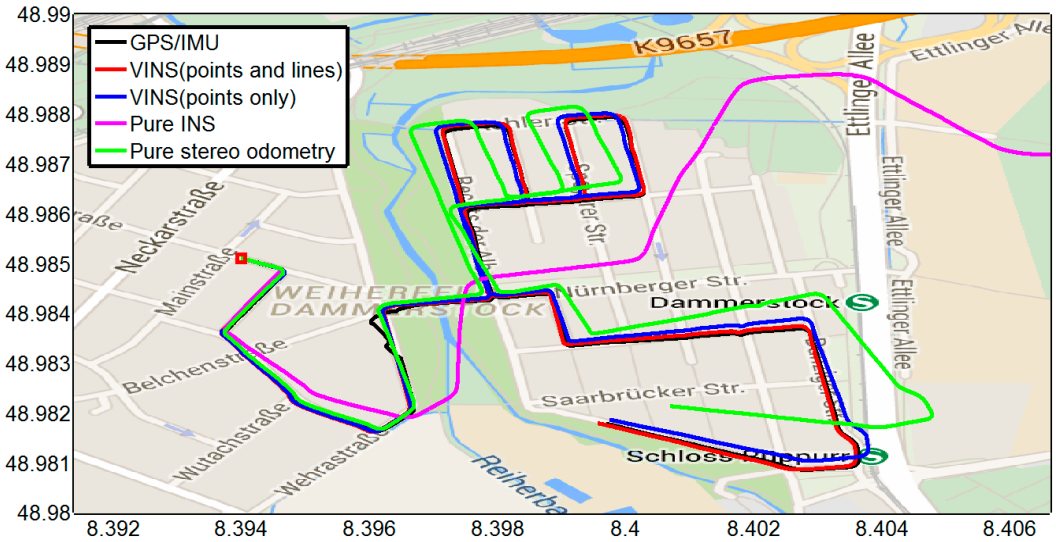

Figure 3. The motion trajectory plot on Google Maps. The initial position is denoted by a red square.

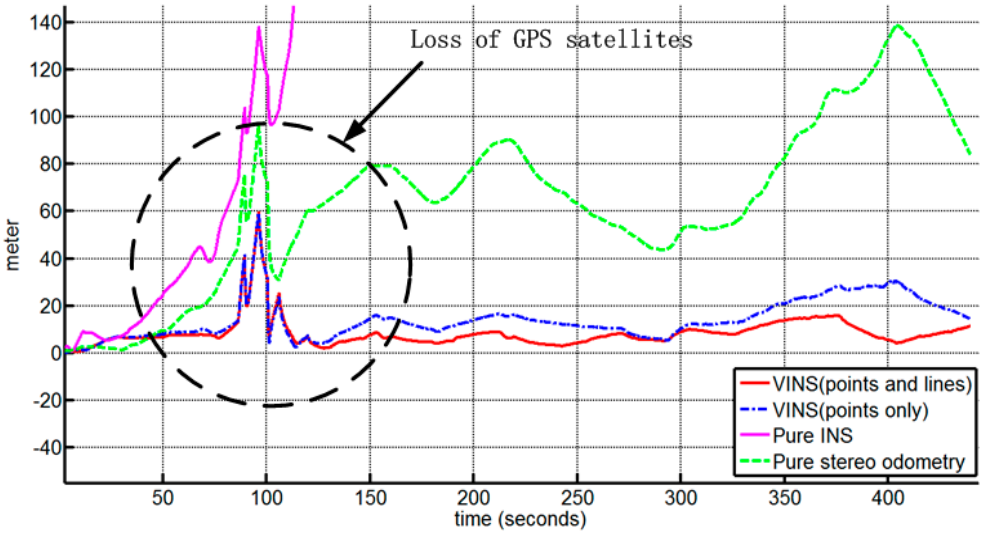

Figure 4. 3D Position Errors of different solutions.

Table 1. The overall RMSE of the outdoor experiment.

Methods	Position RMSE (m)	Orientation RMSE (deg)
VINS (points and lines)	10.6338	0.8313
VINS (points only)	16.4150	0.9126
Pure INS	2149.9	2.0034
Pure stereo odometry	72.6399	8.1809

We demonstrate the velocity and attitude deviations of the proposed method with the corresponding 3σ bounds in Figures 5 and 6, which verify that the velocity and attitude estimates are consistent. Note that the standard deviations of the roll and pitch angle errors are bounded, while the standard deviation of the yaw angle error grows over time. This is consistent with the observable property of the VINS system, which indicates that the yaw angle is unobservable [8]. The yaw angle error is bounded under 5° due to the accuracy of the gyroscopes in the experiment.

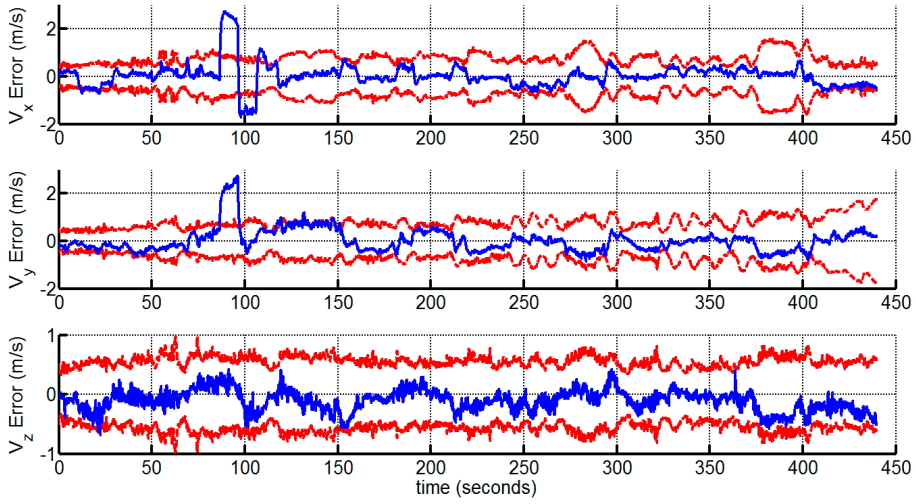

Figure 5. The velocity estimation errors and 3σ bounds (the large deviations around 100th second is due to the ground truth errors).

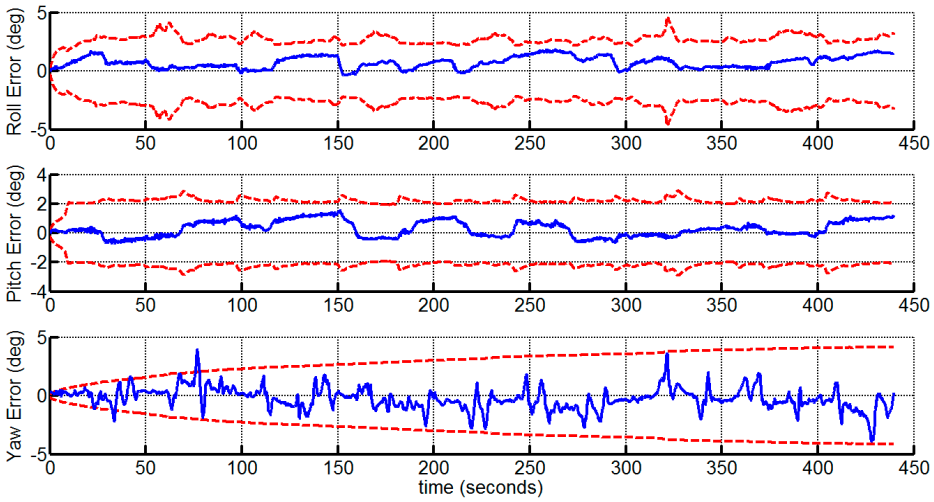

Figure 6. The attitude estimation errors and 3σ bounds.

Finally, the estimates of the gyroscope and accelerometer biases are depicted in Figure 7. All the estimated biases converge quickly to some reasonable ranges, meaning a practical estimation and allowing the compensation of the INS.

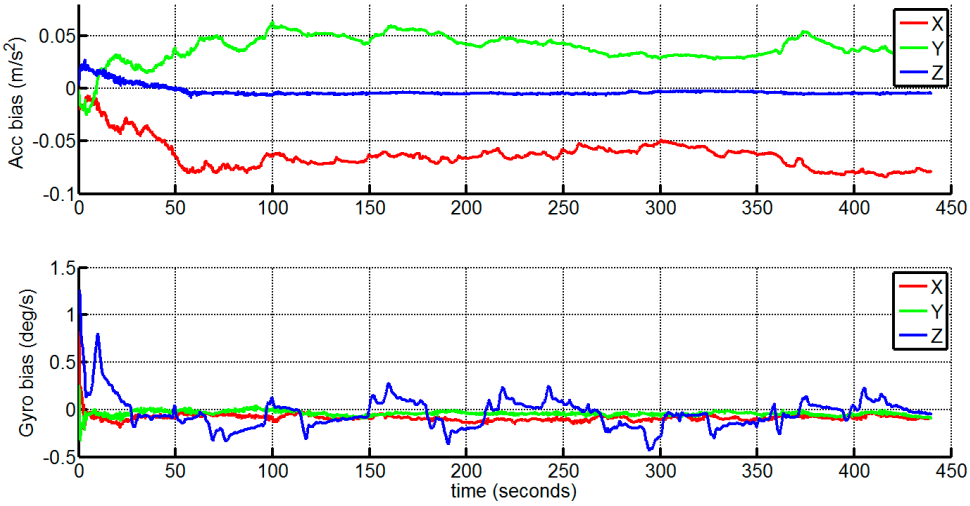

Figure 7. Estimated gyroscope and accelerometer bias.

4.2. Indoor Experiment

To demonstrate the robustness of our algorithm in a textureless structured environment, we perform indoor experiments in a corridor scenario with textureless walls which lead to very few points being tracked in some frames. The test rig consists a PointGrey Bumblebee2 stereo pair, a Xsens MTi unit, and a laptop for data recording (Figure 8a). The accuracy specifications and sampling rates of the sensors are listed in Table 2. The relative pose of the IMU and the camera are well calibrated prior to the experiment using the method proposed in [32], and keep unchanged during the experiment. The actual motion of the pushcart is a move along with the corridor, and then return to the initial point. The full length of the path is about 82 m.

Table 2. The accuracy specifications and sampling rates of the sensors.

Sensors	Accuracies	Sampling Rates
IMU	Gyro bias stability (1 σ): 1°/s Accelerometer bias stability: 0.02 m/s^2	100 Hz
Stereo Camera	Resolution: 640 × 480 pixels Focus length: 3.8 mm Field of view: 70° Base line: 12 cm	12 Hz

In Figure 8b, we show the bird's eye view of the estimated trajectories. It is obvious that the combination of point and line features leads to much better performance than the use of point features alone in this scenario. The reason is that the point features are few or not well distributed over the image in some frames, leading to a bad orientation estimation. In Figure 8c, a plot of the number of inlier point and line features per frame is shown, which clearly demonstrates the superiority of combining both feature types under such circumstances.

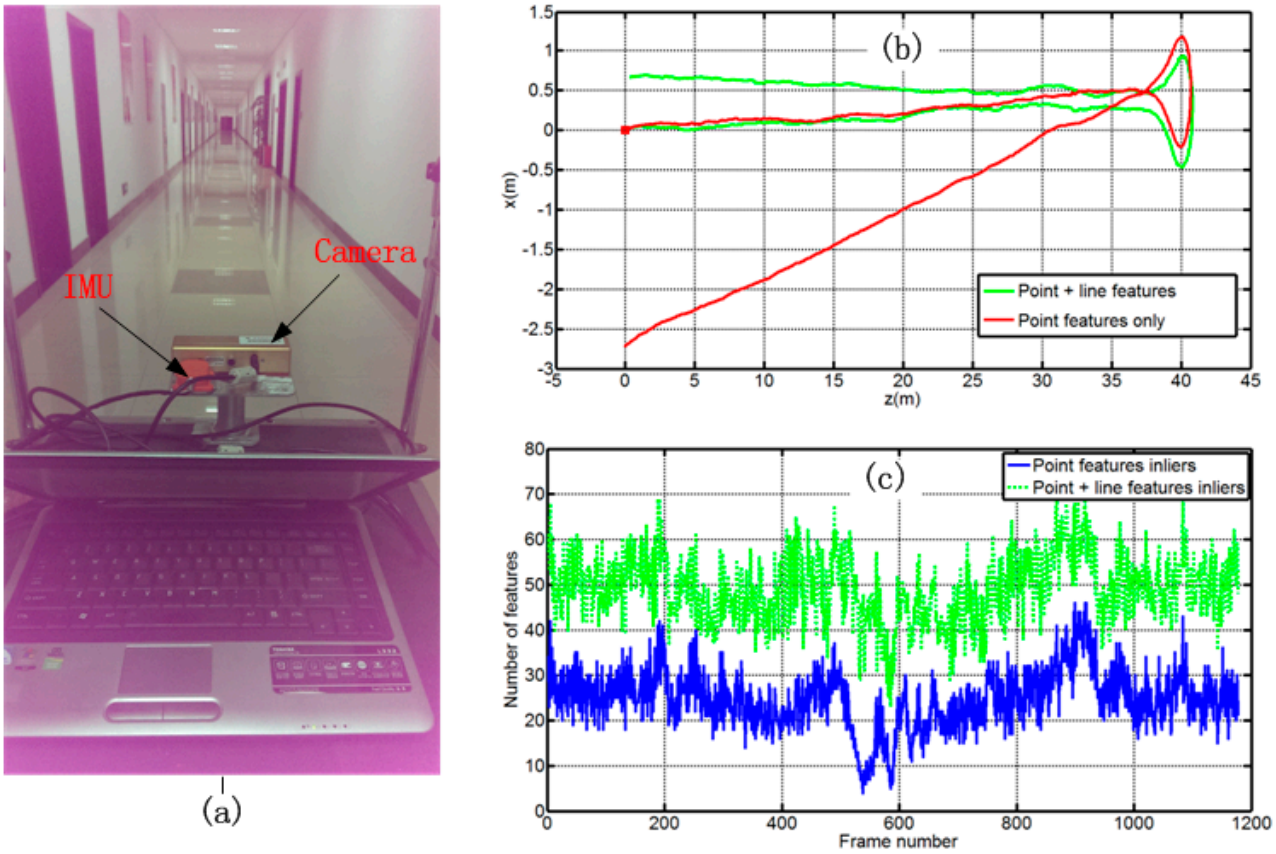

Figure 8. Performance in low-textured indoor environment: (**a**) Experimental setup and experimental scene; (**b**) Top view of estimated trajectories; (**c**) The number of point and line inliers used to estimate the motion.

5. Conclusions/Outlook

This paper presents a tightly-coupled vision-aided inertial navigation algorithm, which exploits point and line features to aid navigation in a simple and unified framework. The measurement models of the point and line features are derived, and incorporated into a single estimator. The outdoor experimental results show that the proposed algorithm performs well in cluttered urban environments. The overall RMSE of position and orientation is about 10.6 m and 0.83°, respectively, over a path of up to about 4 km in length. The indoor experiment demonstrates the better performance and robustness of combining both point and line features in textureless structured environments. The proposed approach which combines both feature types can deal with different types of environments with a slight increase in computational cost.

As part of future work, we aim to improve the proposed approach, by taking advantage of the structural regularity of man-made environments, such as Manhattan-world scenes, *i.e.*, scenes that lines should be orthogonal or parallel to each other [33]. Unlike ordinary lines, the Manhattan-world lines encode the global orientation information, which can be used to eliminate the accumulated orientation errors, and further suppress the position drifts.

Acknowledgments

This work was supported by Research Fund for the Doctoral Program of Higher Education of China (Grant No. 2012.4307.110006) and the New Century Excellent Talents in University of China (Grant No. NCET-07-0225).

Author Contributions

Xianglong Kong, Wenqi Wu and Lilian Zhang conceived the idea and designed the experiments; Xianglong Kong and Yujie Wang performed the experiments; Xianglong Kong analyzed the data and wrote the paper.

Conflicts of Interest

The authors declare no conflict of interest.

References

1. Titterton, D.H.; Weston, J.L. *Srapdown Inertial Navigation Technology*, 2nd ed.; The Institution of Electrical Engineers: London, UK, 2004.
2. Kelly, J.; Sukhatme, G.S. Visual-inertial sensor fusion: Localization, mapping and sensor-to-sensor self-calibration. *Int. J. Robot. Res.* **2011**, *30*, 56–79.
3. Feng, G.H.; Wu, W.Q.; Wang, J.L. Observability analysis of a matrix kalman filter-based navigation system using visual/inertial/magnetic sensors. *Sensors* **2012**, *12*, 8877–8894.
4. Indelman, V.; Gurfil, P.; Rivlin, E.; Rotstein, H. Real-time vision-aided localization and navigation based on three-view geometry. *IEEE Trans. Aerosp. Electron. Syst.* **2012**, *48*, 2239–2259.

5. Weiss, S.; Achtelik, M.W.; Lynen, S.; Chli, M.; Siegwart, R. Real-time onboard visual-inertial state estimation and self-calibration of mavs in unknown environments. In Proceeding of the IEEE International Conference on Robotics and Automation, St. Paul, MN, USA, 14–18 May 2012; pp. 957–964.
6. Kottas, D.G.; Roumeliotis, S.I. Efficient and consistent vision-aided inertial navigation using line observations. In Proceeding of the IEEE International Conference on Robotics and Automation, Karlsruhe, Germany, 6–10 May 2013; pp. 1540–1547.
7. Li, M.Y.; Mourikis, A.I. High-precision, consistent EKF-based visual-inertial odometry. *Int. J. Robot. Res.* **2013**, *32*, 690–711.
8. Hesch, J.A.; Kottas, D.G.; Bowman, S.L.; Roumeliotis, S.I. Camera-imu-based localization: Observability analysis and consistency improvement. *Int. J. Robot. Res.* **2014**, *33*, 182–201.
9. Hu, J.-S.; Chen, M.-Y. A sliding-window visual-imu odometer based on tri-focal tensor geometry. In Proceeding of the IEEE International Conference on Robotics and Automation, Hong Kong, China, 31 May–7 June 2014; pp. 3963–3968.
10. Corke, P.; Lobo, J.; Dias, J. An introduction to inertial and visual sensing. *Int. J. Robot. Res.* **2007**, *26*, 519–535.
11. Roumeliotis, S.I.; Johnson, A.E.; Montgomery, J.F. Augmenting inertial navigation with image-based motion estimation. In Proceeding of the IEEE International Conference on Robotics and Automation, Washington, DC, USA, 12–18 May 2002; pp. 4326–4333.
12. Diel, D.D.; DeBitetto, P.; Teller, S. Epipolar constraints for vision-aided inertial navigation. In Proceeding of the Seventh IEEE Workshops on Application of Computer Vision, Breckenridge, CO, USA, 5–7 Janury 2005; pp. 221–228.
13. Tardif, J.-P.; George, M.; Laverne, M. A new approach to vision-aided inertial navigation. In Proceeding of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Taipei, Taiwan, 18–22 October 2010; pp. 4146–4168.
14. Sirtkaya, S.; Seymen, B.; Alatan, A.A. Loosely coupled kalman filtering for fusion of visual odometry and inertial navigation. In Proceeding of the 16th International Conference on Information Fusion (FUSION), Istanbul, Turkey, 9–12 July 2013; pp. 219–226.
15. Mourikis, A.; Roumeliotis, S.I. A multi-state constraint kalman filter for vision-aided inertial navigation. In Proceeding of the IEEE Inernational Conference in Robotics and Automation, Roma, Italy, 10–14 April 2007; pp. 3565–3572.
16. Leutenegger, S.; Furgale, P.T.; Rabaud, V.; Chli, M.; Konolige, K.; Siegwart, R. Keyframe-based visual-inertial slam using nonlinear optimization. In Proceeding of the Robotics: Science and Systems, Berlin, Germany, 24–28 June 2013.
17. Zhang, L. Line Primitives and Their Applications in Geometric Computer Vision. Ph.D. Thesis, Kiel University, Kiel, Germany, 15 August 2013.
18. Hartley, R.; Zisserman, A. *Multiple View Geometry in Computer Vision*, 2nd ed.; Cambridge University Press: Cambridge, UK, 2004.
19. Sola, J.; Vidal-Calleja, T.; Civera, J.; Montiel, J.M.M. Impact of landmark parametrization on monocular ekf-slam with points and lines. *Int. J. Comput. Vision.* **2012**, *97*, 339–368.

20. Weiss, S.; Siegwart, R. Real-time metric state estimation for modular vision-inertial systems. In Proceeding of the IEEE International Conference on Robotics and Automation (ICRA), Shanghai, China, 9–13 May 2011; pp. 4531–4537.
21. Ford, T.J.; Hamilton, J. A new positioning filter: Phase smoothing in the position domain. *Navigation* **2003**, *50*, 65–78.
22. Van Der Merwe, R. Sigma-Point Kalman Filters for Probabilistic Inference in Dynamic State-Space Models. Ph.D. Thesis, Oregon Health & Science University, Portland, OR, USA, 9 April 2004.
23. Julier, S.J. The scaled unscented transformation. In Proceeding of the American Control Conference, Anchorage, AK, USA, 8–10 May 2002; pp. 4555–4559.
24. Geiger, A.; Lenz, P.; Urtasun, R. Are we ready for autonomous driving? The kitti vision benchmark suite. In Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition, Rhode Island, Greece, 16–21 June 2012; pp. 3354–3361.
25. Rosten, E.; Drummond, T. Machine learning for high-speed corner detection. In *Computer Vision–Eccv 2006*; Leonardis, A., Bischof, H., Pinz, A., Eds.; Springer-Verlag: Berlin/Heidelberg, Germany, 2006; Volume 3951, pp. 430–443.
26. Zhang, Z.; Deriche, R.; Faugeras, O.; Luong, Q.-T. A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. *Artif. Intell.* **1995**, *78*, 87–119.
27. Akinlar, C.; Topal, C. Edlines: A real-time line segment detector with a false detection control. *Pattern Recognit. Lett.* **2011**, *32*, 1633–1642.
28. Zhang, L.; Koch, R. Line matching using appearance similarities and geometric constraints. In *Pattern Recognition*; Pinz, A., Pock, T., Bischof, H., Leberl, F., Eds.; Springer: Berlin/Heidelberg, Germany, 2012; Volume 7476, pp. 236–245.
29. Lowe, D.G. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.* **2004**, *60*, 91–110.
30. Bar-Shalom, Y.; Li, X.R.; Kirubarajan, T. *Estimation with Applications to Tracking and Navigation: Theory Algorithms and Software*; John Wiley & Sons: Hoboken, NJ, USA, 2004.
31. Geiger, A.; Ziegler, J.; Stiller, C. Stereoscan: Dense 3D reconstruction in real-time. In Proceeding of the IEEE Intelligent Vehicles Symposium, Baden-Baden, Germany, 5–9 Junuary 2011; pp. 963–968.
32. Furgale, P.; Rehder, J.; Siegwart, R. Unified temporal and spatial calibration for multi-sensor systems. In Proceeding of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Tokyo, Japan, 3–8 November 2013; pp. 1280–1286.
33. Coughlan, J.M.; Yuille, A.L. Manhattan world: Compass direction from a single image by bayesian inference. In Proceeding of the IEEE International Conference on Computer Vision, Kerkyra, Greece, 20–27 September 1999; pp. 941–947.