This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

Stereo Visual Inertial Pose Estimation Based on Feedforward and Feedbacks

Shengyang Chen, Yurong Feng, Chih-Yung Wen, Yajing Zou, and Wu Chen

***Abstract*—In this article, we present a stereo visual inertial pose estimation method based on feedforward and feedbacks. Compared to the widely used filter-based or optimization-based approaches, the proposed method only stores the most recent pose and measurements and thus can achieve fast processing. A gradient decreased feedback, a roll-pitch feedforward, and a bias estimation feedback, are introduced to fuse the vision and the inertial measurements. This system, which is called feedforward and feedback based visual inertial system (FVIS), is evaluated on the popular European robotics challenge micro aerial vehicle (EuRoC MAV) dataset. FVIS achieves high accuracy and robustness with respect to existing visual inertial simultaneous localization and mapping (SLAM) approaches. FVIS has also been implemented and tested on a unmanned aerial vehicle (UAV) platform. The source code developed during this study is available publicly.**

***Index Terms*—Control system, image motion analysis, inertial navigation, multisensor systems, robot vision systems, SLAM, sensor fusion, state estimation.**

I. Introduction

FUSING the measurement of inertial measurement unit (IMU) and an additional camera extends a monocular visual simultaneous localization and mapping (SLAM) system to a stereo visual inertial system. Such systems are widely used in unmanned aerial vehicle (UAV) [1], [2] applications, and the advantages of the system can be categorized into the following three aspects.

1) Robustness: The pose between consecutive visual frames can be estimated by the IMU. Also, when visual tracking is lost, the IMU can maintain the pose output within a short period.
2) Accuracy: More measurements are fused during pose estimation, increasing accuracy.
3) Scale-consistency: Depth information can be extracted directly from stereo images without any motion. The system scale is consistent, and the initialization can be achieved in one shot.

However, these advantages come with a price. The visual information handled by the stereo visual inertial system is twice as large as that of the monocular system. Also, a typical IMU sensor delivers a data rate on the order of 100 Hz; the modern feature-based viSLAM systems adopt either the filter-based framework or the optimization-based framework to fuse all of these data.

Both of these frameworks are based on finding the most appropriate prediction to satisfy every single measurement (e.g., from visual cameras or IMU). As the amount of input data increases, more computational power is required. For a robotic system, particularly a Low size weight and power system, such as a UAV, the high performance pose estimators will consume too many computational resources.

This study aims to solve the accurate visual inertial fusion problem for resource limited systems. We present a novel visual inertial system based on feedforward and feedbacks. This framework combines the advantages of contemporary frameworks along with several independent models to decouple the stereo visual inertial fusing process. An ideal measurement model is assumed, which means that the errors induced by other settings, such as inaccurate camera calibration, were neglected. The accuracy of the ego-motion estimation only depends on the accuracy of the landmark (LM) position and the quality of the initial guess. Therefore, a better initial guess from IMU and well maintained positions of LMs are closely related to the accuracy of the system.

Fig. 1 shows the simplified system architecture, which includes one feedforward (highlighted in green) and two feedbacks (highlighted in red). The motion estimation unit is based on the standard IMU propagation model, in which a one-step gradient-based filter (Madgwick filter) is included, which fuses the accelerometer and the gyroscope to achieve an accurate and robust orientation estimation. Then, the IMU states are fed into the classical vision estimation pipeline as the correction of the initial guess for pose refinement. After pose refinement, the biases of the accelerometer and the gyroscope are estimated and fed back to the IMU propagation model. In the conventional framework, keyframe technology is used in the front end to restrain the accumulation of drift [3]. In feedforward and

Manuscript received 18 August 2022; revised 4 December 2022 and 26 February 2023; accepted 8 April 2023. Recommended by Technical Editor C. Liu and Senior Editor K. J. Kyriakopoulos. This work was supported by the Resilient Urban PNT Infrastructure to Support Safety of UAV Remote Sensing in Urban Regions, RISUD, PolyU and RCUAS, PolyU. *(Corresponding author: Chih-Yung Wen.)*

Shengyang Chen, Yurong Feng, and Chih-Yung Wen are with the Department of Mechanical Engineering and Department of Aeronautical and Aviation Engineering, Hong Kong Polytechnic University, Hong Kong 999077, Hong Kong (e-mail: 17903070r@connect.polyu.hk; yu-rong.feng@connect.polyu.hk; cywen@polyu.edu.hk).

Yajing Zou and Wu Chen are with the Department of Land Surveying and Geo-Informatics, Hong Kong Polytechnic University, Kowloon 999077, Hong Kong (e-mail: rick.zou@connect.polyu.hk; lswuchen@polyu.edu.hk).

This article has supplementary material provided by the authors and color versions of one or more figures available at https://doi.org/10.1109/TMECH.2023.3272208.

Digital Object Identifier 10.1109/TMECH.2023.3272208

1083-4435 © 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: Brown University. Downloaded on November 09,2023 at 19:02:02 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

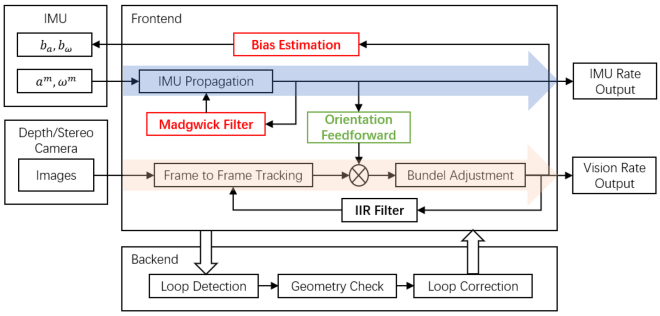


Fig. 1. Simplified system architecture of FVIS.

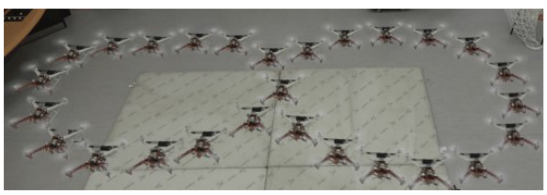


Fig. 2. Composite image of the field test, in which the UAV flies figure-8 trajectory with FVIS.

feedback based visual inertial system (FVIS), the continuous fusion of measurements via an infinite impulse response (IIR) filter ensures high localization accuracy in the current framework. The front end only contains the states of the current frame and the previous frame and can achieve fast processing. The back end of the system is a loop closure module which allows the system to relocalize itself in large scenarios.

Finally, the proposed method is verified by public dataset and in the real-world environment. Evaluation results show that the accuracy of the proposed system is comparable to those of existing methods. In addition, the proposed method is applied to a resource limited UAV system, which is demonstrated to achieve on-board localization and control in an indoor environment (Fig. 2). The primary contributions of this work include the following.

1) A novel pose estimation framework that fuses the visual and inertial measurement through designed feedforward and feedbacks. The proposed method achieved accurate pose estimation with an 80% computational budget when compared to conventional filter-based or optimization-based approaches.
2) Performance evaluation on both the EuRoC UAV dataset and real world experiments. The benchmark evaluation results are compared to top ranked existing algorithms. The implementation of the proposed method is open-sourced.

II. Related Work

A. Filter-Based and Optimization-Based vSLAM

The SLAM problem was modeled based on the Markov assumption and the sensor measurement models [4]. Typically, such a problem can be solved using the extended Kalman filter (EKF). The pose estimation and map recovery are performed by the interactive evolution of the probability densities over the feature depth and the camera pose in the EKF framework [5]. With the development of SLAM theory, visual pose estimation was later modeled as a pose graph optimization on a manifold [6]. In the pose graph optimization, the camera poses and LMs are presented as vertices, and the corresponding measurements are presented as edges. Based on the gradient distributions, the optimizer adjusts the vertices in their neighborhood region to minimize the cost function of edges. To improve local consistency, most of the modern vSLAM works consider information from multiple frames [7], [8]. Consequently, the number of states in the EKF and the object function in the optimizer increase; the processing time becomes longer. The proposed method breaks this performance-processing time tradeoff with a recursive LM update strategy.

B. Monocular vSLAM or RGB-D SLAM

In the monocular vSLAM system, an unavoidable challenge is scale recovery. Among all monocular vSLAM systems, the scale is handled by an independent process with several alternative solutions, e.g., IMU integrated structure from motion (SFM) [9], predefined object pattern [10], and geometric perspective estimation [11]. However, all of these solutions are either based on the preliminary message of the environment or require the camera motion. Apart from monocular cameras, there are currently many red green blue and depth (RGB-D) cameras that are available at reasonable prices. Although different in their sensing modes [12], all of these cameras can sense depth information of a scene immediately without the need for motion. This advantageous feature promotes the studies of RGB-D SLAM. Because depth information can be extracted from every frame, RGB-D SLAM can be modeled as a point cloud registration problem. The egomotion is estimated by an iterative closest point process [13]. Kinect-fusion [14] and dense visual odometry (DVO) [15] are the most notable works of this type. Other depth visual SLAM systems can be regarded as the extensions of their monocular versions, such as visual inertial system with multi-sensor fusion (VINS-Fusion) [8], oriented fast and rotated brief SLAM version 2 (ORBSLAM2) [16] and semi-direct visual-inertial odometry (SVO) 2.0 [17]. In these works, the measurement of another camera and the stereo constraints were added into the optimization-based frameworks. As more information is integrated, these RGB-D SLAM methods are more robust than their monocular counterparts. Thanks to the high quality of the depth image, the RGB-D SLAM can directly output the dense representation of the environment.

C. Visual Inertial Fusion Technologies

Visual inertial SLAM (viSLAM) fuses the inertial measurement to the vSLAM system to increase its accuracy and robustness [18]. Early attempts at applying visual inertial fusion were performed by Diel et al. [19] and Oskiper et al. [20]. During these early stages of research, the visual camera and IMU presented their own perception results and these results were then fused by a filter. Such a system is called a loosely coupled system. In contrast, the tightly coupled system includes both the IMU states

Authorized licensed use limited to: Brown University. Downloaded on November 09,2023 at 19:02:02 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

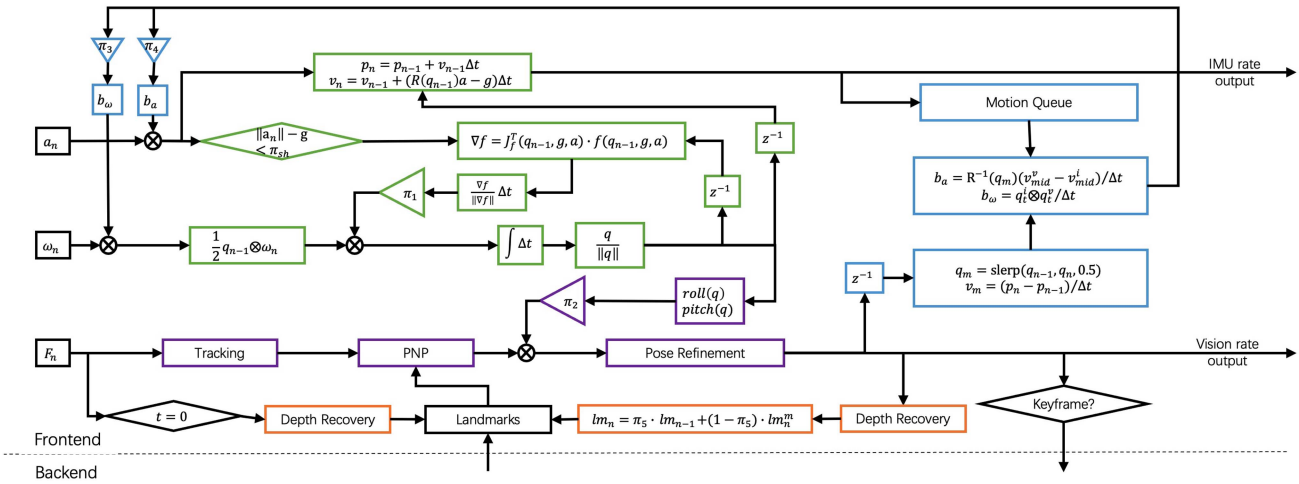


Fig. 3. Frontend workflow.

and visual measurement in their estimation framework, which are innovated together. In multi-state constraint Kalman filter (MSCKF) [5] and robust visual inertial odometry (ROVIO) [7], the nominal state, together with its covariance matrix, is updated in the IMU propagation process and innovated when a visual measurement arrives. The greatest advantage of tightly coupled systems is that the egomotion between camera frames can be recovered with the IMU propagation model. So the systems are able to output real-time egomotion estimation in high frequency. Currently, the tightly coupled systems are widely used in aggressive robotic applications such as UAV navigation. The next evolution of visual inertial fusion is the "preintegration" theory. Because optimization-based systems have become popular, the results of preintegration can be integrated into graph-based optimizers. Most optimization-based methods introduced in the previous subsection have their visual inertial approaches, such as visual-inertial version of direct sparse odometry (VI-DSO) [21] and visual-inertial version of large-scale direct SLAM - (VI LSD-SLAM) [22]. Open keyframe-based visual-inertial SLAM (OKVIS) [3], and visual inertial system with monocular camera (VINSMono) [9] are the benchmarks for viSLAM studies.

D. Towards Fast Processing

For better accuracy of the estimated pose, the vSLAM system tends to fuse information from more frames and requires longer processing time. To achieve real-time processing, the modern methods, such as Stereo odometry relying on feature tracking SLAM (SOFT-SLAM) [23], [24] (Rank 1 in Kitti Odometry benchmark) and Direct sparse odometry lite (DSOL) [25], split the problem into two separate components, tracking (front end) and mapping (back end) and handle them in parallel on the multi-threads. The front end of SOFT-SLAM consists of a two-step (rotation-translation) pose estimator and can process the frames in real-time. The back end of it handles all key frames and generates a global consistency trajectory. By applying a task-based, fine-grained parallelism pipeline, the processing speed of DSOL is five times faster than its original version [26].

III. Front End

This section describes the algorithm pipeline of front end. The front end can be roughly divided into four modules. These four modules are colored in Fig. 3 as follows:

1) IMU propagation with Madgwick feedback (in green);
2) visual estimation with orientation feedforward (in purple);
3) interframe bias estimation feedback (in blue);
4) depth recovery and IIR filter (in orange).

These modules will be introduced in sequence as follows.

A. IMU Propagation With Madgwick Feedback

1) IMU Sensor Model: The micro-electromechanical system (MEMS) inertial measurement unit (IMU) consists of a three-axis accelerometer and a three-axis gyroscope. The simplified IMU model, which neglects the nonlinear and scale factor terms of the comprehensive measurement model, is widely used in pose estimation. In this model, the measured angular velocity $\boldsymbol{\omega}^m$ and $\boldsymbol{a}^m$ acceleration can be described as

$$\boldsymbol{\omega}^m = \boldsymbol{\omega}^{\text{real}} + \boldsymbol{b}_\omega + \boldsymbol{n}_\omega \qquad \boldsymbol{n}_\omega \sim \mathcal{N}\left(\boldsymbol{0}, \boldsymbol{\sigma}_\omega^2\right) \tag{1}$$

$$\boldsymbol{a}^m = \boldsymbol{a}^{\text{real}} + \boldsymbol{b}_a + \boldsymbol{n}_a \qquad \boldsymbol{n}_a \sim \mathcal{N}\left(\boldsymbol{0}, \boldsymbol{\sigma}_a^2\right) \tag{2}$$

$\boldsymbol{\omega}^{\text{real}}$ and $\boldsymbol{a}^{\text{real}}$ are real angular velocity and acceleration, respectively. $\boldsymbol{n}_\omega$ and $\boldsymbol{n}_a$ refer to the intrinsic noises of the sensor, which follow the Gaussian distributions. The biases ($\boldsymbol{b}_\omega$ and $\boldsymbol{b}_a$) are affected by the temperature and change over time.

2) IMU Propagation: For two consecutive IMU readouts at time instants t_{n-1} and t_n, position, velocity, and orientation states can be propagated by

$$\boldsymbol{q}_n = \boldsymbol{q}_{n-1} + \frac{1}{2}\left(\boldsymbol{q}_{n-1} \otimes \boldsymbol{\omega}_n\right)\Delta t \tag{3}$$

$$\boldsymbol{p}_n = \boldsymbol{p}_{n-1} + \boldsymbol{v}_{n-1}\Delta t \tag{4}$$

$$\boldsymbol{v}_n = \boldsymbol{v}_{n-1} + \left(\boldsymbol{R}(\boldsymbol{q}_{n-1})\boldsymbol{a}_n - \boldsymbol{g}\right)\Delta t \tag{5}$$

$$\Delta t = t_n - t_{n-1} \tag{6}$$

Authorized licensed use limited to: Brown University. Downloaded on November 09,2023 at 19:02:02 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

where angular velocity $\boldsymbol{\omega}_n$ and acceleration $\boldsymbol{a}_n$ refer to the nominal state, which has been compensated with the corresponding bias ($\boldsymbol{\omega}_n = \boldsymbol{\omega}_{m,n} - \boldsymbol{b}_\omega$ and $\boldsymbol{a}_n = \boldsymbol{a}_{m,n} - \boldsymbol{b}_a$). The estimation of the biases will be shown in Section III-C. In (3), $\otimes$ refers to the multiplication operation of the quaternions.

3) Madgwick Feedback: To obtain an attitude estimation of high accuracy, the Madgwick feedback of attitude estimation from the accelerometer to the orientation propagation is used. When the visual inertial sensor is in uniform motion or stays steady, the field of gravity and the field of acceleration measurement should be aligned

$$\boldsymbol{a} = {}^i_w\boldsymbol{q}^* \otimes \boldsymbol{g} \otimes {}^i_w\boldsymbol{q} = \boldsymbol{R}({}^i_w\boldsymbol{q}) \cdot \boldsymbol{g} = \boldsymbol{R}(\boldsymbol{q}^*) \cdot \boldsymbol{g} \tag{7}$$

where $\boldsymbol{q}^*$ is the conjugation of the quaternion, $\boldsymbol{R}({}^i_w\boldsymbol{q})$ and $\boldsymbol{R}(\boldsymbol{q}^*)$ are the rotation matrix forms of quaternions ${}^i_w\boldsymbol{q}$ and $\boldsymbol{q}^*$, respectively. In this scenario, by aligning the orientations of these two fields, acceleration can be fed back to the orientation estimation. In detail, when the norm of incoming acceleration is near the magnitude of gravity, (i.e., $||\boldsymbol{a}|| - ||\boldsymbol{g}|| < \pi_{sh}$), the orientation estimation from the accelerometer can be presented by

$$\boldsymbol{q}^{est} = \underset{\boldsymbol{q}}{\operatorname{argmin}}\, \boldsymbol{f}(\boldsymbol{q}, \boldsymbol{g}, \boldsymbol{a})$$
$$\boldsymbol{f}(\boldsymbol{q}, \boldsymbol{g}, \boldsymbol{a}) = \boldsymbol{R}(\boldsymbol{q}^*) \cdot \boldsymbol{g} - \boldsymbol{a}. \tag{8}$$

This optimization problem can be solved by the gradient descent algorithm

$$\boldsymbol{q}_{k+1} = \boldsymbol{q}_k - u\frac{\nabla \boldsymbol{f}(\boldsymbol{q}_k, \boldsymbol{g}, \boldsymbol{a})}{\parallel \nabla \boldsymbol{f}(\boldsymbol{q}_k, \boldsymbol{g}, \boldsymbol{a}) \parallel}, k = 0, 1, 2 \ldots k \tag{9}$$

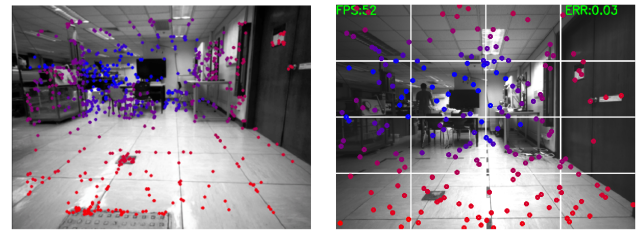
$$\nabla \boldsymbol{f}(\boldsymbol{q}_k, \boldsymbol{g}, \boldsymbol{a}) = \boldsymbol{J}_f^T(\boldsymbol{q}_k) \cdot \boldsymbol{f}(\boldsymbol{q}_k, \boldsymbol{g}, \boldsymbol{a}). \tag{10}$$

Equation (9) shows the result of orientation estimation $\boldsymbol{q}_{k+1}$ after k iterations, which is based on an initial guess orientation $\boldsymbol{q}_0$ and step size u. The gradient of the function can be calculated by its Jacobian and guessed value [(10)]. For the East north up (ENU) coordinate system, the normalized gravity in the world frame can be presented by ${}^w\boldsymbol{g} = (0, 0, 1)^T$. Thus, the objective function $\boldsymbol{f}$ and its Jacobian $\boldsymbol{J}_f^T$ can be derived as follows:

$$\boldsymbol{f}(\boldsymbol{q}_k, \boldsymbol{g}, \boldsymbol{a}) = \begin{bmatrix} 2q_wq_y - 2q_xq_z - a_x \\ -2q_wq_x - 2q_yq_z - a_y \\ 2q_x^2 + 2q_y^2 - 1 - a_z \end{bmatrix} \tag{11}$$

$$\boldsymbol{J}_f^T(\boldsymbol{q}_k) = \begin{bmatrix} 2q_y & -2q_z & 2q_w & -2q_x \\ -2q_x & -2q_w & -2q_z & -2q_y \\ 0 & 4q_x & 4q_y & 0 \end{bmatrix}. \tag{12}$$

Inspired by Madgwick et al. [27], the fusion of orientation estimation from the accelerometer to the propagation model can be performed by fusion of the derivative of orientation $\dot{\boldsymbol{q}}$. We do not use the $\boldsymbol{q}$ obtained from (9). But instead, we take the $\frac{\nabla \boldsymbol{f}(\boldsymbol{q},\boldsymbol{g},\boldsymbol{a})}{\|\nabla \boldsymbol{f}(\boldsymbol{q},\boldsymbol{g},\boldsymbol{a})\|}$ part from the first iteration as an approximation of the orientation derivative from the accelerometer $\dot{\boldsymbol{q}}_a$. The $\frac{1}{2}\boldsymbol{q}_{n-1} \otimes \boldsymbol{\omega}_n$ term in (3) is the orientation derivative from the gyroscope. Combining both items with a fusing weighting factor π, the orientation derivative and fused orientation propagation model can be presented by



(a) (b)

Fig. 4. Features distribution of conventional and improved workflows. (a) Conventional workflow. (b) Improved workflow.

$$\dot{\boldsymbol{q}} = \frac{1}{2}\boldsymbol{q} \otimes \boldsymbol{\omega} + \pi\frac{\nabla \boldsymbol{f}(\boldsymbol{q}_{n-1}, \boldsymbol{g}, \boldsymbol{a})}{\parallel \nabla \boldsymbol{f}(\boldsymbol{q}_{n-1}, \boldsymbol{g}, \boldsymbol{a}) \parallel} \tag{13}$$

$$\boldsymbol{q}_n = \boldsymbol{q}_{n-1} + \dot{\boldsymbol{q}}\Delta t \tag{14}$$

B. Visual Estimation With Orientation Feedforward

1) Feature Detection, Tracking. and 3-D–2-D Pose Estimation: Several improvements are added to improve robustness and quality of the tracking-based implementation. Detection begins with extracting all features from the image. Then, all of these features are assigned to 16 predefined regions based on their distributions in the image plane. In every region, the features are scored with their Harris index [28] and sorted accordingly. Following close feature checking, the 15 top score features are selected and added to the feature list. The close feature checking rules out the feature when it is too near a selected feature (Fig. 4).

For the first frame, the feature becomes LMs after gaining its depth information. For the following frames, after histogram equalization of the image, the features are tracked by the Lucas–Kanade optical flow [29]. After successfully tracking features between consecutive frames, a random sample consensus (RANSAC) P3P [30] algorithm is used to estimate the camera pose and to rules out outliers.

2) Roll and Pitch Feedforward: As introduced in the previous subsection, with the gradient decrease feedback, the IMU can provide accurate orientation estimation. Compared to the yaw direction, estimations in pitch and roll directions are more accurate because the gravity components in these two directions are larger than in the z-axis direction. Therefore, we trust the estimation in these two directions and introduce the pitch and roll feedforward prior to the bundle adjustment process.

3) Pose Refinement: Prior to output of the estimated result, a small scale optimization is conducted. The optimizer consists of the following three kinds of elements: 1) the pose vertex of the current frame; 2) fixed 3-D point vertices of LM position in the current frame and 3) reprojection edges. The optimizer refined the pose by minimizing reprojection errors. The refined pose is the vision rate output of the system. Besides, the refined pose is also used to estimate the bias of IMU.

Authorized licensed use limited to: Brown University. Downloaded on November 09,2023 at 19:02:02 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

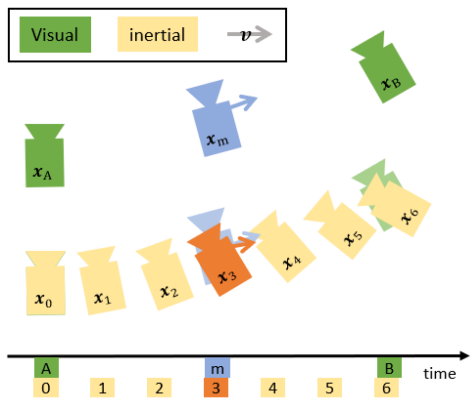


Fig. 5. Bias estimation.

C. Interframe Bias Estimation Feedback

The biases, which act as errors, accumulate along with IMU propagation, and they induce drifts in the estimation process. The gyroscope bias induces orientation drift, while the velocity drift is accumulated by the accelerometer bias and further induces drift in position. The drift compensation and more accurate egomotion estimation can be achieved by estimating the biases and feeding them back to the IMU propagation model.

Prior to the derivation of bias estimation, some temporarily used notations are introduced. Taking Fig. 5 as an example, two frames are captured at times t_A and t_B. In the interval between these two frames, there exist seven IMU inputs at time $t_0 \cdots t_6$. The relevant vision states and IMU states are presented by ${}_v\boldsymbol{x}_A$, ${}_v\boldsymbol{x}_B$ and ${}_i\boldsymbol{x}_0 \cdots {}_i\boldsymbol{x}_6$. Δt_A^B refers to the time interval between t_A and t_B.

The interframe bias estimation is based on the following three assumptions.

1) The biases remain constant in the interframe section.
2) The white noise of the IMU is ignored ($\boldsymbol{n}_\omega = \boldsymbol{n}_a = 0$).
3) The vision state and IMU state are aligned at the beginning with the same initial state (${}_v\boldsymbol{x}_A = {}_i\boldsymbol{x}_0$).

In addition, in real situations, orientation and velocity drifts are coupled. However, in the interframe bias estimation scenario, the coupled term can be neglected because the magnitudes of these drifts are small in a short period of time. Thus, the orientation drift and velocity drift are solved separately.

For orientation drift and gyroscope bias estimation, the vision states are considered to be accurate and drift-free and can be propagated by

$$ {}_v\boldsymbol{q}_n = {}_v\boldsymbol{q}_{n-1} + \frac{1}{2}\left({}_v\boldsymbol{q}_{n-1} \otimes \boldsymbol{\omega}_n\right)\Delta t_{n-1}^n. \tag{15} $$

However, biases exist in the IMU propagation and affect the IMU related state. Thus, (3) becomes

$$ {}_i\boldsymbol{q}_n = {}_i\boldsymbol{q}_{n-1} + \frac{1}{2}({}_i\boldsymbol{q}_{n-1} \otimes (\boldsymbol{\omega}_n + \boldsymbol{b}_\omega))\Delta t_{n-1}^n. \tag{16} $$

Because Δt is small, this equation can be approximated by IMU states and a bias-induced turbulence

$$ {}_i\boldsymbol{q}_n \approx {}_i\boldsymbol{q}_{n-1} + \frac{1}{2}({}_i\boldsymbol{q}_{n-1} \otimes \boldsymbol{\omega}_n)\Delta t_{n-1}^n + \frac{1}{2}\left({}_i\boldsymbol{q}_{n-1} \otimes \boldsymbol{b}_\omega\right)\Delta t_{n-1}^n. \tag{17} $$

Considering the initial state assumption (${}_i\boldsymbol{q}_{n-1} = {}_v\boldsymbol{q}_{n-1}$) and recursive integration over time, (17) can be approximated as

$$ \begin{aligned} {}_i\boldsymbol{q}_n &\approx {}_v\boldsymbol{q}_n + \frac{1}{2}\left({}_v\boldsymbol{q}_{n-1} \otimes \boldsymbol{b}_\omega\right)\Delta t_0^n \\ &\approx {}_v\boldsymbol{q}_n + \frac{1}{2}\left({}_v\boldsymbol{q}_n \otimes \boldsymbol{b}_\omega\right)\Delta t_0^n. \end{aligned} \tag{18} $$

Multiplying ${}_v\boldsymbol{q}_n^{-1}$ on both sides yields

$$ {}_v\boldsymbol{q}_n^{-1} \otimes {}_i\boldsymbol{q}_n \approx 1 + \frac{1}{2}\boldsymbol{b}_\omega \Delta t_0^n \tag{19} $$

and the gyroscope bias can be solved as

$$ \boldsymbol{b}_\omega = \frac{2\left({}_v\boldsymbol{q}_n^{-1} \otimes {}_i\boldsymbol{q}_n - 1\right)}{\Delta t_0^n}. \tag{20} $$

Similarly, for the velocity drift and accelerometer bias, the corresponding vision and IMU states are

$$ {}_v\boldsymbol{v}_n = {}_v\boldsymbol{v}_{n-1} + \left(\boldsymbol{R}({}_v\boldsymbol{q}_{n-1})\boldsymbol{a}_n - \boldsymbol{g}\right)\Delta t_{n-1}^n \tag{21} $$

$$ {}_i\boldsymbol{v}_n = {}_i\boldsymbol{v}_{n-1} + \left(\boldsymbol{R}({}_i\boldsymbol{q}_{n-1})\left(\boldsymbol{a}_n + \boldsymbol{b}_a\right) - \boldsymbol{g}\right)\Delta t_{n-1}^n. \tag{22} $$

The IMU state can be approximated by

$$ \begin{aligned} {}_i\boldsymbol{v}_n &\approx {}_i\boldsymbol{v}_{n-1} + \left(\boldsymbol{R}({}_i\boldsymbol{q}_{n-1})\boldsymbol{a}_n - \boldsymbol{g}\right)\Delta t_{n-1}^n + \boldsymbol{R}({}_i\boldsymbol{q}_{n-1})\boldsymbol{b}_a\Delta t_{n-1}^n \\ &\approx {}_v\boldsymbol{v}_n + \boldsymbol{R}({}_v\boldsymbol{q}_{n-1})\boldsymbol{b}_a\Delta t_0^n \\ &\approx {}_v\boldsymbol{v}_n + \boldsymbol{R}({}_v\boldsymbol{q}_n)\boldsymbol{b}_a\Delta t_0^n. \end{aligned} \tag{23} $$

Rearranging the above equation and multiplying $\boldsymbol{R}({}_v\boldsymbol{q}_n^{-1})$ on both sides yields

$$ \boldsymbol{R}\left({}_v\boldsymbol{q}_n^{-1}\right)\left({}_i\boldsymbol{v}_n - {}_v\boldsymbol{v}_n\right) \approx \boldsymbol{b}_a\Delta t_0^n \tag{24} $$

and the accelerometer bias can be solved as

$$ \boldsymbol{b}_a = \frac{\boldsymbol{R}\left({}_v\boldsymbol{q}_n^{-1}\right)\left({}_i\boldsymbol{v}_n - {}_v\boldsymbol{v}_n\right)}{\Delta t_0^n}. \tag{25} $$

Equations (20) and (25) are used to estimate the gyroscope bias and accelerometer bias, respectively. As shown in Fig. 5, the gyroscope bias is estimated from the derived visual state $\boldsymbol{x}_B$ and the relevant IMU state $\boldsymbol{x}_6$. However, the velocity cannot be derived from a single measurement. Thus, the accelerometer bias is estimated from an interpolated middle state $\boldsymbol{x}_m$ and the corresponding IMU state $\boldsymbol{x}_3$. The middle state contains velocity and orientation parts. The velocity $\boldsymbol{v}_m$ can be calculated by the difference of the position ($\boldsymbol{p}$), and the orientation $\boldsymbol{q}_m$ can be derived from the spherical linear interpolation (slerp) operation

$$ \boldsymbol{v}_m = \frac{\boldsymbol{p}_B - \boldsymbol{p}_A}{\Delta t_A^B} \tag{26} $$

$$ \boldsymbol{q}_m = slerp(\boldsymbol{q}_A, \boldsymbol{q}_B, 0.5). \tag{27} $$

D. Depth Recovery and IIR Filter

Some stereo cameras, such as Kinect or D435i, contain embedded vision processors and can output the rectified depth

Authorized licensed use limited to: Brown University. Downloaded on November 09,2023 at 19:02:02 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

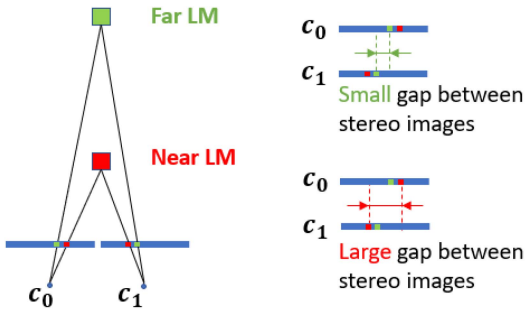


Fig. 6. Disparity of LM on the image plane.

images. The depth information can be directly read out from the depth image. For conventional stereo camera pairs, a sparse depth recovery is performed. The Lucas–Kanade optical flow [29] extracts features pairs from stereo images. Then, a triangulation process [31] recovers the depth of these pairs.

As shown in Fig. 6, a far LM and a near LM are observed by the c_0 camera and c_1 camera. By aligning the image planes of the two cameras (see the right part of Fig. 6), the disparity of a certain LM in the image plane exhibits a negative correlation with its distance from the camera. For the near LM, a large gap exists between the corresponding pairs on two image planes. However, because the Lucas–Kanade optical flow can only apply to a limited region on the image plane, the large gap leads to tracking failure. Therefore, the near LMs cannot recover their depth information. To address this issue, we improve the workflow with the initial guess and dummy depth technology, as follows.

1) Reproject LMs into the c_1 frame as an initial guess. For newly detected features, the initial guesses are their positions in the c_0 frame.
2) Apply the Lucas–Kanade optical flow from the c_0 frame to the c_1 frame.
3) Triangulate the point pairs. For those untracked features, dummy depths with random depth values are given.

Therefore, every feature will gain its depth information from a single capture. For the near LMs, although their depth information is false in the first loop, there is a high success rate of determining the correct positions in the following loops. Then, the triangulation can be performed correctly.

A popular way to improve measurement accuracy is to apply the IIR filter. Thus, by continuously fusing independent sensor measurements of the system states, the uncertainty level of these states will be reduced. In a stereo vision system, we can get the depth measurements of the LMs in every capture. A certain LM may be captured in different frames. Besides, these measurements are independent of each other. We apply such a filter to maintain the position information of the LMs. While comparing to other types of measurement fusion algorithm (such as the depth filter in direct sparse odometry (DSO) [26]), the IIR filter relies only on the current measurement and the previous measurement. The greatest advantage of adopting the IIR filter is that all the measurements throughout the observation lifespan are considered and fused while the number of system state maintains small.

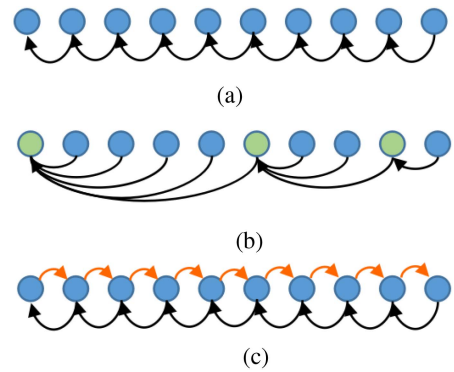


Fig. 7. Vision-based pose estimation workflows. (a) Frame to frame pose estimation. (b) frame to keyframe pose estimation. (c) Frame to frame pose estimation with IIR filter.

In implementation of the IIR, the LMs can first be projected into the view of camera c_0 (or camera c_1 in Fig. 6)

$$ {}_{c_0}\boldsymbol{lm}_{n-1} = {}^{c_0}_{w}\boldsymbol{T}^{est} \cdot \boldsymbol{lm}_{n-1} \tag{28} $$

and IIR filter is adopted next

$$ {}_{c_0}\boldsymbol{lm}_{n} = \pi \cdot {}_{c_0}\boldsymbol{lm}_{n-1} + (1-\pi) \cdot {}_{c_0}\boldsymbol{lm}^{m}_{n}. \tag{29} $$

We compare three different methods in Fig. 7. In the frame to frame (F2F) workflow, the LM position only depends on the last measurement, and the uncertainty will accumulate quickly along the workflow. To inhibit the fast accumulation of drift, the frame to key frame workflow selects a frame as the key frame and uses it to estimate the following frames. The uncertainty level remains the same between the two key frames. In the IIR filter-based workflow, the previous measurements are fused with new measurements in every frame, and thus use all information throughout the LM lifespan (from the first detection to disappearance).

IV. Back End

The back end of the FVIS is a loop closing module. If the motion between the latest frame and last key frame is larger than a threshold, the latest frame will become new key frame and be added to the back end.

A. Loop Detection

Similar to VINS-Mono and ORBSLAM, direct index bag of words (DBoW) is used to find a candidate loop from the key frame database [32]. DBoW extracts descriptors from numerous images and then clusters the representative descriptors as words. Descriptors on the queried key frame will be transferred to a word vector and then compared with previous key frames by calculating the similarity score.

B. Geometry Check

If the environment contains similar textures or objects, the wrong loop candidate may be detected. Therefore, a geometry check should be performed to avoid perceptual aliasing. The relative motion between the loop candidate and the current

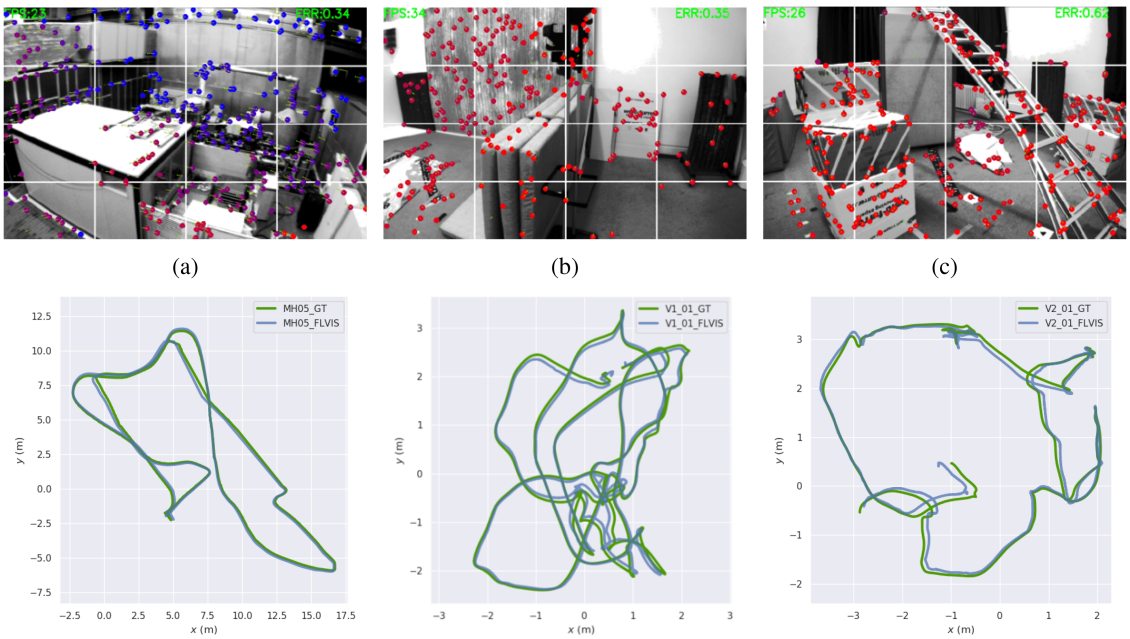
Authorized licensed use limited to: Brown University. Downloaded on November 09,2023 at 19:02:02 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

TABLE I
COMPARISON OF THE RESULTS OF DIFFERENT USING DIFFERENT VISLAM IMPLEMENTATIONS IN EUROC MAV DATASETS

Sequence	Length	APE_{trans} / RPE_{rot}					
		FVIS	**VINS-Fusion**	**MSCKF**	**Open-vins**	**Basalt**	**ORB-SLAM3**
MH01	80.96	0.131/0.025	0.268/0.094	n.a.	0.138/0.070	0.09/0.0017	0.027/0.0020
MH02	73.22	0.126/0.026	0.202/0.028	0.059/0.172	0.204/0.036	0.051/0.0016	0.033/0.0020
MH03	127.07	0.399/0.050	0.356/0.050	0.334/0.050	0.248/0.053	0.085/0.0023	0.035/0.0026
MH04	91.10	0.45/0.037	0.445/0.113	0.327/0.037	0.585/0.096	n.a.	0.061/0.0019
MH05	96.98	0.412/0.035	0.379/0.098	0.392/0.059	0.96/0.094	0.127/0.0017	0.082/0.0018
V101	60.12	0.11/0.011	0.135/0.009	0.106/0.024	0.103/0.009	0.042/0.0022	0.038/0.0027
V102	74.68	0.122/0.012	0.138/0.013	0.149/0.052	0.132/0.012	0.056/0.0045	0.014/0.0054
V103	75.10	0.139/0.013	0.124/0.013	0.179/0.084	0.102/0.011	0.072/0.0051	0.024/0.0067
V201	36.58	0.255/0.011	0.141/0.015	0.106/0.023	0.093/0.014	0.035/0.0022	0.032/0.0026
V202	81.47	0.3/0.017	0.205/0.033	0.215/0.050	0.178/0.033	0.064/0.0047	0.012/0.0058
V203	83.52	n.a.	0.378/0.037	n.a.	0.19/0.040	n.a.	0.021/0.0069
Mean	-	0.244/0.023	0.252/0.045	0.207/0.061	0.267/0.043	0.069/0.003	0.035/0.004

Note: MH: Machine hall sequence; V: Vicon room sequence. The final results of each dataset were the averages of five runs. If the method fails in any of the five runs, the result will be mark as n.a. FVIS, VINS-FSION, MSCKF, and open-vins output are recorded in real-time. The Basalt and ORB-SLAM3 output are generated after the end of the trail.



(a) (b) (c)

(d) (e) (f)

Fig. 8. FVIS results on EuRoC MAV datasets. (a)–(c) are the scenes of the MH05, V101, and V201 sequences, respectively. (d)–(f) are the comparison of the trajectories (FVIS) and GT; (a)(d), (b)(e), and (c)(f) are the FVIS results for the corresponding EuRoC MAV sequences.

key frame is calculated by RANSAC perspective-n-point (PnP). Then, the inlier features are selected by ratio testing, cross matching, and RANSAC filtering.

C. Loop Correction

If the loop candidate passes the geometry test, pose-graph optimization is performed to correct the key frame poses along the loop. This process involves the following two types of error: 1) adjacent error and 2) loop error

$$\boldsymbol{e}_{m,n} = \ln\left({}^{m}_{c}\boldsymbol{T}_n \cdot {}^{w}_{c}\boldsymbol{T}_m^{-1} \cdot {}^{w}_{c}\boldsymbol{T}_n\right) \tag{30}$$

where, for the adjacent error, m and n are adjacent key frame indices; and for loop error, they represent the end indices of the loop. The objective function of pose graph optimization is

$$\sum_{a} \boldsymbol{e}_{a,a+1}^{T}\Omega_{a,a+1}^{-1}\boldsymbol{e}_{a,a+1} + \sum_{l} \rho_h\left(\boldsymbol{e}_{l1,l2}^{T}\Omega_{l1,l2}^{-1}\boldsymbol{e}_{l1,l2}\right) \tag{31}$$

where $\boldsymbol{e}_{a,a+1}$, $\boldsymbol{e}_{l1,l2}$, and Ω are the adjacent error, the loop error, and the covariance matrix, respectively. ρ_h is the Huber norm to mitigate the effect of possible outliers. Again, the Gauss–Newton method in g2o is applied to minimize the above equation.

V. EXPERIMENTAL RESULTS AND DISCUSSIONS

We evaluate the proposed method using both public datasets and field test results. The accuracy of the system is described by

Authorized licensed use limited to: Brown University. Downloaded on November 09,2023 at 19:02:02 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

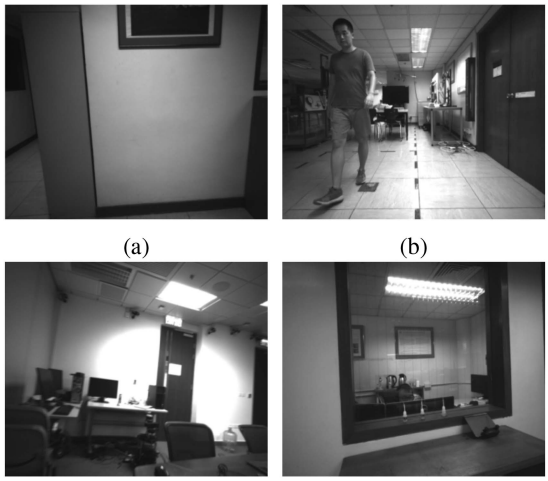


Fig. 9. Snapshots with (a) poorly-detailed features, (b) pedestrian, (c) large overexposure area, and (d) glass window, presenting some challenging scenarios in the real-world environment.

TABLE II
PROCESSING TIME ON DIFFERENT HARDWARE PLATFORMS

		HP1	HP2	HP3	HP4
Frontend time (ms)	Tracking	13.7	10.7	6.6	2.7
	Pose estimation	3.8	2.5	1.8	1.1
	Feature detection	43.7	24.0	14.6	7.1
	Depth recovery-stereo	25.3	17.0	18.3	3.9
	Depth recovery-depth	1	1	1	1
Front end rate (FPS)	Open-vins stereo	8	12	20	43
	VINS-Fusion stereo	9	13	21	41
	FVIS stereo	11	15	24	67
	FVIS depth	16	20	42	83

Note: Data are the mean of 25 iterations for all hardware platform (HP); HP1: UP-board (x5-z8350, 4G RAM); HP2: LattePanda 864s (m3-8100Y, 8G RAM); HP3: Laptop (i5-7200U, 8G RAM); HP4: Intel NUC (i7-8665U, 16G RAM).

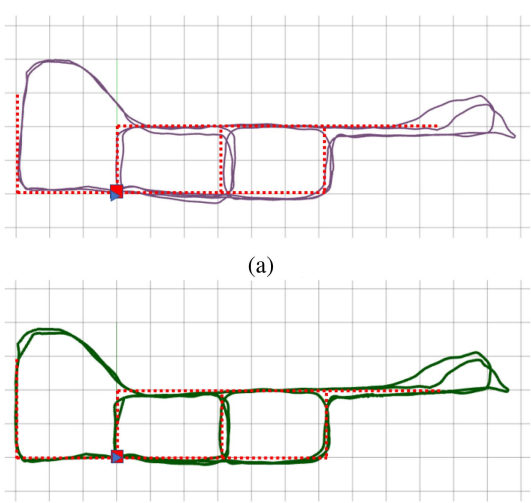


Fig. 10. Trajectories with/without loop closure in the laboratory scenario. The start and end points of the trajectories are marked as red rectangle and blue triangle; the grid size of the figure is $1\,m \times 1\,m$. (a) Without loop closure. (b) With loop closure.

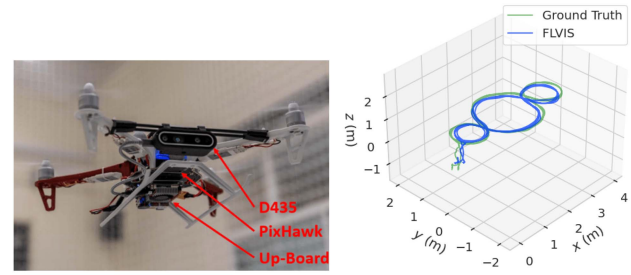


Fig. 11. Field experiment. (a) UAV experimental platform. (b) Trajectory of loop closure-disabled FVIS on the UAV platform.

TABLE III
CASE STUDY OF LOOP CONTRIBUTIONS TO THE SYSTEM

Sequence	APE_{trans} / RPE_{rot}			
	MH01	MH02	v101	v201
Case Study 1	0.131/0.025	0.123/0.026	0.11/0.011	0.122/0.012
Case Study 2	0.258/0.026	0.182/0.027	0.143/0.011	0.271/0.013
Case Study 3	0.145/0.029	0.126/0.034	0.117/0.020	0.136/0.022

Note: Case Study 1: Enable all four modules; Case Study 2: Disable IIR filter; Case Study 3: disable all IMU related loops

the rmse of translation and rotation drifts. Both the absolute pose error (APE) and the one-step relative pose error (RPE) cases are considered [33].

A. EuRoC Mav Dataset

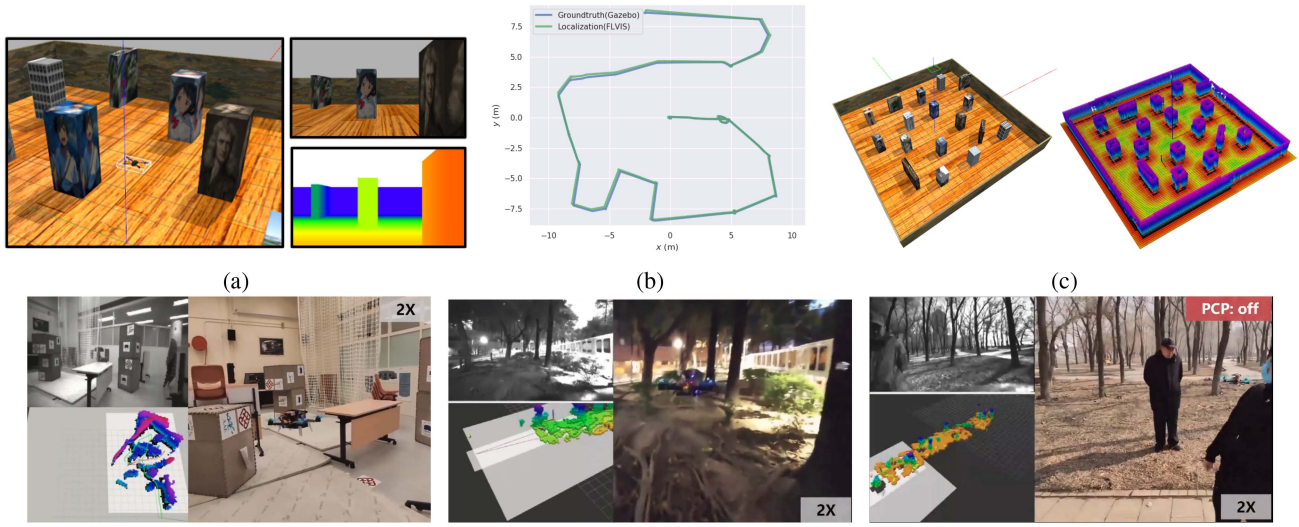
The proposed system is first evaluated using the European robotics challenge micro aerial vehicle (EuRoC MAV) visual inertial datasets [34]. These datasets are collected using a UAV mounted visual inertial sensor, which consists of a stereo module (Aptina MT9V034 global shutter, WVGA monochrome, 20 frames per second (FPS)) and a hardware synchronized MEMS IMU (ADIS16448, angular rate and acceleration, 200 Hz). The ground truths (GT) of these datasets are provided by a VICON motion capture system (VICON) (6-D pose, vicon room sequences) and a Leica MS50 (3-D position, machine hall sequence). The datasets are run on a standard laptop (CPU: Core i5-7200 U; RAM: 8 GB).

For the comparison group, the notable work of the EKF-based (MSCKF [5] and open-vins [35]) and optimization-based (VINS-Fusion [8]) methods are selected. All these four algorithms are run in the stereo visual inertial mode and disables the loop closure function. We also test the Basalt [36], and oriented fast and Rotated brief SLAM version 3 (ORB-SLAM3) [37] as a reference. These two methods are known for their powerful relocalization modules and global map management. Selected results are shown in Fig. 8, and the APEs of translation (APE_{trans}) and RPEs of rotation (RPE_{rot}) for all sequences are listed in Table I.

As shown in Table I, Basalt and ORB-SLAM3 outperform four other implementations as they manage a global map and apply map-level optimization at the back end. These methods

Authorized licensed use limited to: Brown University. Downloaded on November 09,2023 at 19:02:02 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.



(a) (b) (c)

(d) (e) (f)

Fig. 12. FVIS empowered UAV navigation framework in the simulation environment and in real-world environments. (a), (b), and (c) are in the simulation environment. (a) UAV flies in the simulator and the onboard image and depth image are the inputs of the FVIS. (b) Comparison of the estimated pose and the ground truth. (c) Bird view of the reconstructed map and the simulated ground truth. (d), (e), and (f) are in real-world environments. (d) UAV flies in the lab environment. (e) UAV flies in the bush at night. (f) UAV flies in the crowded park—Upper left: camera view; bottom left: path and reconstructed surroundings. right: 3rd person view.

require high computation power and a lot of memory overhead. Besides, Basalt and ORB-SLAM3 exist a time delay between the input and the output because of the time-consuming global optimization. To our best knowledge, there is no reported successful case that uses these two approaches in UAV applications. Therefore, we focus on the comparison of the first four approaches. All approaches achieve high accuracies (i.e., $APE_{\text{trans}}/length < 1\%$ and $RPE_{\text{rot}} < 0.05$). Among them, FVIS achieved top APE_{trans} for two sequences and top RPE_{rot} for six sequences, however it failed in one sequence; open-vins has the best APE_{trans} performance in most of the room sequences, and the performance decreases in the machine hall sequences; MSCKF exhibits a balance performance in both room and machine hall sequences, however it failed in two sequences. The vins-fusion has good accuracy when considering the RPE_{rot}. Consider the mean value of the error, FVIS ranks two in APE_{trans} and rank one in RPE_{rot}. In summary, FVIS outperforms or is comparable to VINS-Fusion, MSCKF, and open-vins with respect to most machine hall sequences and vicon room sequences. Because every stereo image is considered to have equal variability in our proposed FVIS method, the nonuniform exposure of the stereo images affects the accuracy of the system (e.g., in the V1 03 and V2 03 sequences). As shown in Fig. 8, good agreement between the estimated trajectory and the ground truth can be observed.

B. Typical Processing Time of FVIS in Different Platforms

The FVIS algorithm is tested in several hardware platforms, varying from an embedded x5-z8350 processor to a powerful i7-8665 U platform. Processing times are measured based on the MH 05 sequences in the EuRoC dataset (stereo mode) and the real-world lab sequences (depth mode). The average processing time for each part of the algorithm is listed in the upper part of Table II. The processing time of FVIS is compared with the conventional optimization-based system (VINS-Fusion) and filter-based system (open-vins). As shown in the lower part of Table II, FVIS runs faster than VINS-Fusion and open-vins. In addition, with the VPU integrated sensor, which can output the depth directly, FVIS can achieve higher FPS. FVIS is demonstrated to be suitable for resource-limited robotic systems.

C. Contributions of Modules

In this subsection, we discuss how designed modules contribute to the system. We conducted experiments by enabling/disabling certain modules for comparison. By default (Case Study 1), we enable all four modules. In Case Study 2, we disable the IIR filter, which means FVIS degenerates to a F2F work flow. In Case Study 3, we disable all IMU related modules (Madgwick filter, orientation feedforward, and bias estimation feedback). The stereo visual inertial system of FVIS degenerates to a stereo visual system. We tested all these cases in the EuRoC sequences and the results are shown in the Table III.

By comparing Case Studies 1 and 2, IIR filter is shown to play a critical role in the pose estimation with a reduction of 20% to 50% absolute translation error. Additionally, comparing the results of Case Studies 1 and 3 indicate that the fusion of IMU measurement (by the Madgwick filter and orientation feedforward) can also improve system accuracy.

D. Field Tests

1) Real-World Environment With D435i Sensors: A real-world test using the RealSense D435i sensor was conducted. The sensor contains two global shutter cameras (resolution: 640x480, image rate: 30 Hz), an IMU (Bosch BMI055, accelerometer update rate: 250 Hz, gyroscope update rate: 200 Hz,

Authorized licensed use limited to: Brown University. Downloaded on November 09,2023 at 19:02:02 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

IMU update rate: 450 Hz with linear interpolation) and an integrated vision processing unit that can directly output the depth image. In contrast to the previous test, the algorithm here directly uses the depth output of the camera instead of triangulation from the stereo images, leading to reduced processing time. The real-world test contains two sequences (i.e., with and without loop closure thread enabled), which cover some challenging scenarios (Fig. 9). The complete video clips can be found in the supplemental material.

Fig. 10 shows the estimated trajectories in the laboratory with and without the loop closure thread enabled. The red dotted line along the grid in Fig. 10 represents the designed walking trajectory marked by the dark tape on the floor of the laboratory. In this test, the vi sensor moves along the designed trajectory in the laboratory for several turns and finally returns to where the test started. When the sensor passes the same place for the second time, the loop closure thread identifies the scenario and optimizes the loop poses between two passes. As shown in the Fig. 10, for the no loop closure trajectory, certain observable drifts occur, causing the trajectory to be unable to return to the starting point. In contrast, the trajectory with loop closure is much more consistent than that without loop closure.

2) FVIS in Autonomous UAV Navigation: For aerial vehicle applications, FVIS is applied on a UAV platform to achieve autonomous feedback control in a VICON motion capture laboratory. The loop closure thread is not enabled in the test. As shown in Fig. 11(a), the UAV platform consists of a forward looking depth camera (D435i), an IMU integrated fly controller (Pixhawk), and an on-board computer (Up-Board). The UAV is commanded to follow a predefined trajectory of three circles with radius of 0.5 m, 1.5 m, and 0.5 m. The ground truth is captured by the VICON system. The estimated trajectory of the FVIS is shown in Fig. 11(b). Results show good agreement between the estimated trajectory and ground truth. The total length of the trajectory is 52.1 m and the APE of the FVIS is 0.17 m.

Besides, as a localization kit, FVIS has been integrated into the UAV navigation framework. This navigation framework consists of our in-house mapping kit [38] and planning kit [39] and has been tested in simulation and in real-world environments. In the simulation, the input of the FVIS is the rendered colored image and depth image [Fig. 12(a)]. The output of the FVIS is used for position control of the UAV and map reconstruction. The trajectory length is 83 m, and the APE (translation) is 0.3 m [Fig. 12(b)]. The reconstructed map can accurately describe the setup of the simulation world [Fig. 12(c)]. The field tests consist of various scenarios [Fig. 12(d)–(f)], such as in the lab environment, in the dark night condition, and in a public park. In all these tests, the FLVIS, with mapping and planning kits, runs on the onboard computer (Latte Panda 864). The FVIS runs in the stereo visual inertial mode. More test videos can be found in the supplementary materials.

VI. Conclusion and Future Works

In this article, we propose a new stereo visual inertial framework, called FVIS. The new FVIS replaces the conventional fusion models (e.g., EKF) with a new control-based strategy. In the FVIS, the objective of fusion is no longer to determine the most appropriate pose guess to satisfy every single measurement (visual and IMU) in a certain period of time. Instead, the FVIS provides a good initial pose value and maintains more reliable LM positions. We then compare the FVIS with other existing open source implementations on public datasets. The superior performance of the FVIS is described, and the FVIS is also demonstrated in a resource-limited UAV platform. The implementation of the FVIS has been made open source for the benefit of the viSLAM community. Like all other local estimators, FVIS has unavoidable long-term drift accumulation. We plan to integrate FVIS with global navigation satellite system (GNSS) in the future. After aligning the local trajectory with the global trajectory, GNSS measurements will correct the long-term drift. Furthermore, FVIS will be a reliable guess in the receiver autonomous integrity monitoring module. Besides, we will keep trying to improve the system's robustness. The improvement may include dynamic object removal and robust tracking toward aggressive motion.

Acknowledgment

The authors would like to thank J. Chang, W. Zhou, and H. Chen from MAV LAB, PolyU for their support with hardware, video recording, and experimentation.

References

[1] C. Fu, A. Sarabakha, E. Kayacan, C. Wagner, R. John, and J. M. Garibaldi, "Input uncertainty sensitivity enhanced nonsingleton fuzzy logic controllers for long-term navigation of quadrotor UAVs," *IEEE/ASME Trans. Mechatron.*, vol. 23, no. 2, pp. 725–734, Apr. 2018.

[2] R. Duan, D. P. Paudel, C. Fu, and P. Lu, "Stereo orientation prior for UAV robust and accurate visual odometry," *IEEE/ASME Trans. Mechatron.*, vol. 27, no. 5, pp. 3440–3450, Oct. 2022.

[3] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale, "Keyframe-based visual–inertial odometry using nonlinear optimization," *Int. J. Robot. Res.*, vol. 34, no. 3, pp. 314–334, 2015.

[4] S. Thrun, "Probabilistic robotics," *Commun. ACM*, vol. 45, no. 3, pp. 52–57, 2002.

[5] A. I. Mourikis and S. I. Roumeliotis, "A multi-state constraint Kalman filter for vision-aided inertial navigation," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2007, pp. 3565–3572.

[6] G. Grisetti, R. Kümmerle, C. Stachniss, U. Frese, and C. Hertzberg, "Hierarchical optimization on manifolds for online 2D and 3D mapping," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2010, pp. 273–278.

[7] M. Bloesch, S. Omari, M. Hutter, and R. Siegwart, "Robust visual inertial odometry using a direct ekf-based approach," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2015, pp. 298–304.

[8] T. Qin, J. Pan, S. Cao, and S. Shen, "A general optimization-based framework for local odometry estimation with multiple sensors," 2019, *arXiv:1901.03638*.

[9] T. Qin, P. Li, and S. Shen, "VINS-mono: A robust and versatile monocular visual-inertial state estimator," *IEEE Trans. Robot.*, vol. 34, no. 4, pp. 1004–1020, Aug. 2018.

[10] B. Pfrommer and K. Daniilidis, "TAGSLAM: Robust slam with fiducial markers," 2019, *arXiv:1910.00679*.

[11] X. Wang, H. Zhang, X. Yin, M. Du, and Q. Chen, "Monocular visual odometry scale recovery using geometrical constraint," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2018, pp. 988–995.

[12] S. Giancola, M. Valenti, and R. Sala, *A Survey on 3D Cameras: Metrological Comparison of Time-of-Flight, Structured-Light and Active Stereoscopy Technologies*. Cham, Switzerland: Springer, 2018.

[13] P. J. Besl and N. D. McKay, "A method for registration of 3-D shapes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 2, pp. 239–256, 1992.

Authorized licensed use limited to: Brown University. Downloaded on November 09,2023 at 19:02:02 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

[14] S. Izadi et al., "Kinectfusion: Real-time 3D reconstruction and interaction using a moving depth camera," in *Proc. 24th Annu. ACM Symp. User interface Softw. Technol.*, 2011, pp. 559–568.

[15] C. Kerl, J. Stuckler, and D. Cremers, "Dense continuous-time tracking and mapping with rolling shutter RGB-D cameras," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 2264–2272.

[16] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: An open-source SLAM system for monocular, stereo, and RGB-D cameras," *IEEE Trans. Robot.*, vol. 33, no. 5, pp. 1255–1262, Oct. 2017.

[17] C. Forster, Z. Zhang, M. Gassner, M. Werlberger, and D. Scaramuzza, "SVO: Semidirect visual odometry for monocular and multicamera systems," *IEEE Trans. Robot.*, vol. 33, no. 2, pp. 249–265, Apr. 2017.

[18] C. N. Taylor, M. J. Veth, J. F. Raquet, and M. M. Miller, "Comparison of two image and inertial sensor fusion techniques for navigation in unmapped environments," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 47, no. 2, pp. 946–958, Apr. 2011.

[19] D. D. Diel, P. DeBitetto, and S. Teller, "Epipolar constraints for vision-aided inertial navigation," in *Proc. IEEE 7th Workshops Appl. Comput. Vis.*, 2005, pp. 221–228.

[20] T. Oskiper, Z. Zhu, S. Samarasekera, and R. Kumar, "Visual odometry system using multiple stereo cameras and inertial measurement unit," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2007, pp. 1–8.

[21] L. Von Stumberg, V. Usenko, and D. Cremers, "Direct sparse visual-inertial odometry using dynamic marginalization," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2018, pp. 2510–2517.

[22] V. Usenko, J. Engel, J. Stückler, and D. Cremers, "Direct visual-inertial odometry with stereo cameras," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2016, pp. 1885–1892.

[23] I. Cvišić, J. Ćesić, I. Marković, and I. Petrović, "Soft-SLAM: Computationally efficient stereo visual simultaneous localization and mapping for autonomous unmanned aerial vehicles," *J. Field Robot.*, vol. 35, no. 4, pp. 578–595, 2018.

[24] I. Cvišić, I. Marković, and I. Petrović, "SOFT2: Stereo visual odometry for road vehicles based on a point-to-epipolar-line metric," *IEEE Trans. Robot.*, vol. 39, no. 1, pp. 273–288, Feb. 2023.

[25] C. Qu, S. S. Shivakumar, I. D. Miller, and C. J. Taylor, "DSOL: A fast direct sparse odometry scheme," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2022, pp. 10587–10594.

[26] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 3, pp. 611–625, Mar. 2017.

[27] S. O. Madgwick, A. J. Harrison, and R. Vaidyanathan, "Estimation of IMU and marg orientation using a gradient descent algorithm," in *Proc. IEEE Int. Conf. Rehabil. Robot.*, 2011, pp. 1–7.

[28] C. G. Harris et al., "A combined corner and edge detector," in *Proc. Alvey Vis. Conf.*, 1988, vol. 15, no. 50, pp. 10–5244.

[29] J.-Y. Bouguet et al., "Pyramidal implementation of the affine lucas kanade feature tracker description of the algorithm," 2001. [Online]. Available: http://robots.stanford.edu/cs223b04/algotracking.pdf

[30] X.-S. Gao, X.-R. Hou, J. Tang, and H.-F. Cheng, "Complete solution classification for the perspective-three-point problem," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 8, pp. 930–943, Aug. 2003.

[31] R. I. Hartley and P. Sturm, "Triangulation," *Comput. Vis. Image Understanding*, vol. 68, no. 2, pp. 146–157, 1997.

[32] D. Gálvez-López and J. D. Tardós, "Bags of binary words for fast place recognition in image sequences," *IEEE Trans. Robot.*, vol. 28, no. 5, pp. 1188–1197, Oct. 2012.

[33] M. Grupp, "EVO: Python package for the evaluation of odometry and SLAM," 2017. [Online]. Available: https://github.com/MichaelGrupp/evo

[34] M. Burri et al., "The EuRoC micro aerial vehicle datasets," *Int. J. Robot. Res.*, vol. 35, no. 10, pp. 1157–1163, 2016.

[35] P. Geneva, K. Eckenhoff, W. Lee, Y. Yang, and G. Huang, "Openvins: A research platform for visual-inertial estimation," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2020, pp. 4666–4672.

[36] V. Usenko, N. Demmel, D. Schubert, J. Stückler, and D. Cremers, "Visual-inertial mapping with non-linear factor recovery," *IEEE Robot. Automat. Lett.*, vol. 5, no. 2, pp. 422–429, Apr. 2020.

[37] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. Montiel, and J. D. Tardós, "ORB-SLAM3: An accurate open-source library for visual, visual-inertial and multi-map SLAM," *IEEE Trans. Robot.*, vol. 37, no. 6, pp. 1874–1890, 2021.

[38] S. Chen, H. Chen, C.-W. Chang, and C.-Y. Wen, "Multilayer mapping kit for autonomous UAV navigation," *IEEE Access*, vol. 9, pp. 31493–31503, 2021.

[39] H. Chen, P. Lu, and C. Xiao, "Dynamic obstacle avoidance for UAVs using a fast trajectory planning approach," in *Proc. IEEE Int. Conf. Robot. Biomimetics*, 2019, pp. 1459–1464.

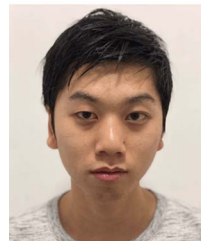
Shengyang Chen received the B.E. degree in aeronautical engineering from Northwestern Polytechnical University, Xi.an, China, in 2013, the M.Sc. degree in mechatronics from the University of Siegen, Siegen, Germany, in 2017, and the Ph.D. degree in mechanical engineering from the Hong Kong Polytechnic University, Hong Kong, in 2021.

He is currently a Senior Engineer with Huawei.

Yurong Feng received the B.E. degree in aeronautical engineering from Xi'an Jiaotong University, Xi'an, China, in 2013, and the M.S. degree in aviation engineering from The Hong Kong Polytechnic University, Hong Kong, where he is currently working toward the Ph.D. degree with the MAV/UAV Laboratory, Department of Aeronautical and Aviation Engineering.

Chih-Yung Wen received the B.S. degree in mechanical engineering from National Taiwan University, Taipei, Taiwan, in 1986, and the M.S. and Ph.D. degrees in aeronautics from the California Institute of Technology (Caltech), Pasadena, CA, USA, in 1989 and 1994, respectively.

He is currently a Chair Professor with the Department of Aeronautical and Aviation Engineering, The Hong Kong Polytechnic University, Hong Kong. His research interests include hypersonic aero-thermodynamics, shock/droplet and shock/bubble interactions, detonation, flow control by plasma actuators on a delta wing, and UAVs and MAVs.



Yajing Zou received the B.S. and M.S. degrees in land surveying and remote sensing from Wuhan University, Wuhan, China, in 2013 and 2016, respectively, and the Ph.D. degree in navigation and positioning from the Hong Kong Polytechnic University, Hong Kong, in 2022.

He is currently a Research Engineer with AutoX.

Wu Chen received the B.S. degree in physics from the University of Science and Technology of China, Hefei, China, and the Ph.D. degree in land surveying from Newcastle University, Newcastle, U.K., in 1992.

In 2000, he joined the Hong Kong Polytechnic University, Hong Kong, where he is currently a Professor with the Department of Land Surveying and Geoinformatics. He has actively worked on GNSS-related research for more than 30 years. His research interests include Geodesy and Geodynamics, seamless positioning technologies, indoor positioning, navigation and integrity, GNSS positioning and applications, system integration, GNSS performance evaluation, regional GPS networks, wireless sensor network positioning, and airborne Lidar applications.

Authorized licensed use limited to: Brown University. Downloaded on November 09,2023 at 19:02:02 UTC from IEEE Xplore. Restrictions apply.