宁波大学
NINGBO UNIVERSITY

本科毕业设计（论文）

外文翻译

题目：面向轻量化定位与地图构建方法研究

学　　院	信息科学与工程学院
专　　业	计算机科学与技术
班　　级	22 计算机一班
学　　号	226002618
学生姓名	李杰
指导教师	彭成斌
开题日期	2025 年 12 月 25 日

<文献翻译一：原文>

OpenVINS：A Research Platform for Visual-Inertial Estimation[1]

Patrick Geneva, Kevin Eckenhoff, Woosik Lee, Yulin Yang, and Guoquan Huang

Abstract— In this paper, we present an open platform, termed OpenVINS, for visual-inertial estimation research for both the academic community and practitioners from industry. The open sourced codebase provides a foundation for researchers and engineers to quickly start developing new capabilities for their visual-inertial systems. This codebase has out of the box support for commonly desired visual-inertial estimation features, which include: (i) on-manifold sliding window Kalman filter, (ii) online camera intrinsic and extrinsic calibration, (iii) camera to inertial sensor time offset calibration, (iv) SLAM landmarks with different representations and consistent First-Estimates Jacobian (FEJ) treatments, (v) modular type system for state management, (vi) extendable visual-inertial system simulator, and (vii) extensive toolbox for algorithm evaluation. Moreover, we have also focused on detailed documentation and theoretical derivations to support rapid development and research, which are greatly lacked in the current open sourced algorithms. Finally, we perform comprehensive validation of the proposed OpenVINS against state-of-the-art open sourced algorithms, showing its competing estimation performance.

• Open source: https://github.com/rpng/open_vins

• Documentation: https://docs.openvins.com

I. INTRODUCTION

Autonomous robots and consumer-grade mobile devices such as drones and smartphones are becoming ubiquitous, in part due to a large increase in computing ability and a simultaneous reduction in power consumption and cost. To endow these robots and mobile devices with the ability to perceive and understand their contextual locations within local environments, which is desired in many different applications from mobile AR/VR to autonomous navigation, visual-inertial navigation systems (VINS) are often used to provide accurate motion estimates by fusing the data from on-board camera and inertial sensors [1].

[1] Geneva P, Eckenhoff K, Lee W, et al. OpenVINS: A Research Platform for Visual-Inertial Estimation[C]. 2020 IEEE International Conference on Robotics and Automation (ICRA), 2020: 4666-4672.

Developing a working VINS algorithm from scratch has proven to be challenging, and in the robotics research community, this has shown to be a significant hurdle for researchers due to the lack of VINS codebases that have comprehensive documentation and detailed derivations for which even users with little background can learn and extend a current state-of-the-art work to address their problems at hand. While there are several open sourced visual-inertial codebases [2]–[8], they are not developed for extensibility and lack proper documentation and evaluation tools, which, in our experience, are crucial for rapid development and deep understanding, thus accelerating VINS research and development in the field. Moreover, these systems have many hard-coded assumptions or features that require an intricate understanding of the codebases in order to adapt them to the sensor systems at hand. This, along with inadequate documentation and support, limits their wide adoption in different applications.

To fill the aforementioned void in the community and to promote the VINS research in robotics and beyond, in this paper, we present an extendable, open sourced codebase that is particularly designed for researchers and practitioners with either limited or extensive background knowledge of state estimation. We provide the necessary documentation, tools, and theory for those who are even new to visual-inertial estimation, and term this collection of utilities as OpenVINS (OV). This codebase has been the foundation of many of the recent visual-inertial estimation projects in our group at the University of Delaware, which include multi-camera [9], multi-IMU [10], visual-inertial moving object tracking [11], [12], Schmidt-based visual-inertial SLAM [13], [14], point-plane and point-line visual-inertial navigation [15], [16], among others [17]–[19]. We summarize the key functionality of the different components in OpenVINS as follows:

• ov_core – Contains 2D image sparse visual feature tracking; linear and Gauss-Newton feature triangulation methods; visual-inertial simulator for arbitrary number of cameras and frequencies; and fundamental manifold math operations and utilities.

• ov_eval – Contains trajectory alignment; plotting utilities for trajectory accuracy and consistency evaluation; Monte-Carlo evaluation of different accuracy metrics; and utility for recording ROS topics to file.

• ov_msckf – Contains the extendable modular Extended Kalman Filter (EKF)-based sliding window visual-inertial estimator with on-manifold type system for flexible state representa-

tion. Features include: First-Estimates Jacobians (FEJ) [20]–[22], IMU-camera time offset calibration [23], camera intrinsics and extrinsic online calibration [24], standard MSCKF [25], and 3D SLAM landmarks of different representations.

In what follows we describe our generalized modular on-manifold EKF-based estimator which, in its simplest form, estimates the current state of a camera-IMU pair. We then introduce the implemented features that provide the foundation for researchers to quickly build and extend on. Note that what we present here is only a brief introduction to the feature set and readers are referred to our thorough documentation website. We also provide an evaluation of the proposed EKF-based solution in simulations and then on real-world datasets, clearly demonstrating its competing performance against other open sourced algorithms.

II. ON-MANIFOLD MODULAR EKF

The state vector of our visual-inertial system consists of the current inertial navigation state, a set of c historical IMU pose clones, a set of m environmental landmarks, and a set of w cameras' extrinsic and intrinsic parameters.

$$\mathbf{x}_k = [\mathbf{x}_I^\top \quad \mathbf{x}_C^\top \quad \mathbf{x}_M^\top \quad \mathbf{x}_W^\top \quad {}^{C}t_I]^\top \tag{1}$$

$$\mathbf{x}_I = \left[{}^{I_k}_{G}\bar{q}^\top \quad {}^{G}\mathbf{p}_{I_k}^\top \quad {}^{G}\mathbf{v}_{I_k}^\top \quad \mathbf{b}_{\omega_k}^\top \quad \mathbf{b}_{a_k}^\top \right]^\top \tag{2}$$

$$\mathbf{x}_C = \left[{}^{I_{k-1}}_{G}\bar{q}^\top \quad {}^{G}\mathbf{p}_{I_{k-1}}^\top \quad \cdots \quad {}^{I_{k-c}}_{G}\bar{q}^\top \quad {}^{G}\mathbf{p}_{I_{k-c}}^\top \right]^\top \tag{3}$$

$$\mathbf{x}_M = \left[{}^{G}\mathbf{p}_{f_1}^\top \quad \cdots \quad {}^{G}\mathbf{p}_{f_m}^\top \right]^\top \tag{4}$$

$$\mathbf{x}_W = \left[{}^{I}_{C_1}\bar{q}^\top \quad {}^{C_1}\mathbf{p}_I^\top \quad \boldsymbol{\zeta}_0^\top \cdots \quad {}^{I}_{C_w}\bar{q}^\top \quad {}^{C_w}\mathbf{p}_I^\top \quad \boldsymbol{\zeta}_w^\top \right]^\top \tag{5}$$

where ${}^{I_k}_{G}\bar{q}$ is the unit quaternion parameterizing the rotation $\mathbf{R}\left({}^{I_k}_{G}\bar{q}\right) = {}^{I_k}_{G}\mathbf{R}$ from the global frame of reference $\{G\}$ to the IMU local frame $\{I_k\}$ at time k[26], $\mathbf{b}_\omega$ and $\mathbf{b}_a$ are the gyroscope and accelerometer biases, and ${}^{G}\mathbf{v}_{I_k}$ and ${}^{G}\mathbf{p}_{I_k}$ are the velocity and position of the IMU expressed in the global frame, respectively. The inertial state $\mathbf{x}_I$ lies on the manifold defined by the product of the unit quaternions $\mathbb{H}$ with the vector space $\mathbb{R}^{12}$ (i.e. $\mathcal{M} = \mathbb{H} \times \mathbb{R}^{12}$) and has 15 total degrees of freedom (DOF).

For vector variables, the "boxplus" and "boxminus" operations, which map elements to and from a given manifold [27], equate to simple addition and subtraction of their vectors. For quaternions, we define the quaternion boxplus operation as:

$$\bar{q}_1 \boxplus \delta\boldsymbol{\theta} \triangleq \begin{bmatrix} \frac{\delta\boldsymbol{\theta}}{2} \\ 1 \end{bmatrix} \otimes \bar{q}_1 \simeq \bar{q}_2 \tag{6}$$

Note that although we have defined the orientations using the left quaternion error, it is not limited to this and any onmanifold representation in practice can be used (e.g., [28]).

The map of environmental landmarks $\mathbf{x}_M$ contains global 3D positions only for simplicity, while in practice we offer support for different representations (e.g. inverse MSCKF [25], full inverse depth [29], and anchored 3D position [30]).

The calibration vector $\mathbf{x}_W$ contains the camera intrinsics $\boldsymbol{\zeta}$, consisting of focal length, camera center, and distortion parameters, and the camera-IMU extrinsics, i.e., the spatial transformation (relative pose) from the IMU to each camera. Since we consider synchronized camera clocks, we include a single time offset ${}^{C}t_I$ between the IMU and the camera clock in the calibration vector.

A. Propagation

The inertial state $\mathbf{x}_I$ is propagated forward using incoming IMU measurements of linear accelerations ${}^{I}\mathbf{a}_m$ and angular velocities ${}^{I}\boldsymbol{\omega}_m$ based on the following generic nonlinear IMU kinematics propagating the state from timestep $k-1$ to k [31]:

$$\mathbf{x}_k = f(\mathbf{x}_{k-1}, {}^{I}\mathbf{a}_m, {}^{I}\boldsymbol{\omega}_m, \mathbf{n}) \tag{7}$$

where $\mathbf{n}$ contains the zero-mean white Gaussian noise of the IMU measurements along with random walk bias noise. This state estimate is evaluated at the current estimate:

$$\hat{\mathbf{x}}_{k|k-1} = f(\hat{\mathbf{x}}_{k-1|k-1}, {}^{I}\mathbf{a}_m, {}^{I}\boldsymbol{\omega}_m, \mathbf{0}) \tag{8}$$

where $\cdot$ denotes the estimated value and the subscript $k \mid k-1$ denotes the predicted estimate at time k given the measurements up to time $k-1$. The state covariance matrix is propagated typically by linearizing the nonlinear model at the current estimate:

$$\mathbf{P}_{k|k-1} = \boldsymbol{\Phi}_{k-1}\mathbf{P}_{k-1|k-1}\boldsymbol{\Phi}_{k-1}^{\top} + \mathbf{Q}_{k-1} \tag{9}$$

where $\boldsymbol{\Phi}_{k-1}$ and $\mathbf{Q}_{k-1}$ are respectively the system Jacobian and discrete noise covariance matrices [25]. The clones $\mathbf{x}_C$, environmental features $\mathbf{x}_M$, and calibration $\mathbf{x}_W$ states do not evolve with time and thus the corresponding state Jacobian entries are identity with zero propagation noise and allow for exploitation of the sparsity for computational savings.

B. On-Manifold Update

Consider the following nonlinear measurement function:

$$\mathbf{z}_{m,k} = h(\mathbf{x}_k) + \mathbf{n}_{m,k} \tag{10}$$

where we have the measurement noise $\mathbf{n}_{m,k} \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_{m,k})$. For the standard EKF update, one linearizes the above equation at the current state estimate. In our case, as in the indirect EKF [26], we linearize (10) with respect to the current zero-mean error state (i.e. $\tilde{\mathbf{x}} = \mathbf{x} \boxminus \hat{\mathbf{x}} \sim \mathcal{N}(\mathbf{0}, \mathbf{P})$):

$$\mathbf{z}_{m,k} = h(\hat{\mathbf{x}}_{k|k-1} \boxplus \tilde{\mathbf{x}}_{k|k-1}) + \mathbf{n}_{m,k} \tag{11}$$
$$= h(\hat{\mathbf{x}}_{k|k-1}) + \mathbf{H}_k \tilde{\mathbf{x}}_{k|k-1} + \mathbf{n}_{m,k} \tag{12}$$
$$\Rightarrow \tilde{\mathbf{z}}_{m,k} = \mathbf{H}_k \tilde{\mathbf{x}}_{k|k-1} + \mathbf{n}_{m,k} \tag{13}$$

where $\mathbf{H}_k$ is the measurement Jacobian computed as follows:

$$\mathbf{H}_k = \left. \frac{\partial h(\hat{\mathbf{x}}_{k|k-1} \boxplus \tilde{\mathbf{x}}_{k|k-1})}{\partial \tilde{\mathbf{x}}_{k|k-1}} \right|_{\tilde{\mathbf{x}}_{k|k-1}=\mathbf{0}} \tag{14}$$

Using this linearized measurement model, we can now perform the following standard EKF update to ensure the updated states remain on-manifold:

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} \boxplus \mathbf{K}_k \left(\mathbf{z}_{m,k} - h(\hat{\mathbf{x}}_{k|k-1}) \right) \tag{15}$$
$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K}_k \mathbf{H}_k \mathbf{P}_{k|k-1} \tag{16}$$
$$\mathbf{K}_k = \mathbf{P}_{k|k-1} \mathbf{H}_k^\top \left(\mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^\top + \mathbf{R}_{m,k} \right)^{-1} \tag{17}$$

III. OPENVINS RESEARCH PLATFORM

A. Type-based Index System

At the core of the OpenVINS library is the type-based index system. Inspired by graph-based optimization frameworks such as GTSAM [32], we abstract away from the user the need to directly manipulate the covariance and instead provide the tools to automatically manage the state and its covariance. This offers many benefits such as reduced implementation time and being less prone to development errors due to explicit state and covariance access.

Each state variable "type" has internally the location of where it is in the error state which is automatically updated during initialization, cloning, or marginalization operations which affect variable ordering. A type is defined by its covariance location, its current estimate and its error state size. The current value does not have to be a vector, but could be a matrix in the case of an $\mathbb{SO}(3)$ rotation representation. The error state for all types is a vector and thus a type will need to define the boxplus mapping between its error state and its manifold representation (i.e. the update function).

```
class Type {
protected:
    // Current best estimate
    Eigen :: MatrixXd _value;
    // Index of error state in covariance
    int _id = -1;
    // Dimension of error state
    int _size = -1;
    // Vector correction, how to update
    void update (const Eigen ::VectorXd dx);
};
```

One of the main advantages of this type system is that it reduces the complexity of adding new features by allowing the user to construct sparse Jacobians. Instead of constructing a Jacobian for all state elements, the "sparse" Jacobian needs to only include the state elements that the measurement is a function of. This both saves computation in the cases where a measurement is a function of only a few state elements and allows for measurement functions to be state agnostic as long as their involved state variables are present.

B. State Variable Initialization

Based on a set of linearized measurement equations (13), we aim to optimally compute the initial estimate of a new state variable and its covariance and correlations with the existing state variables. As a motivating example, we here describe how to initialize a new SLAM landmark ${}^{G}\mathbf{p}_f$, whose key logic can be used for any new state variable and is generalized to any type within the codebase. As in [33] we first perform QR decomposition (e.g., using computationally efficient in-place Givens rotations) to separate the linear system (13) into two subsystems: (i) one that depends on the new state (i.e., ${}^{G}\mathbf{p}_f$), and (ii) the other that does not.

$$\tilde{\mathbf{z}}_{m,k} = [\mathbf{H}_x \quad \mathbf{H}_f] \begin{bmatrix} \tilde{\mathbf{x}}_k \\ {}^{G}\tilde{\mathbf{p}}_f \end{bmatrix} + \mathbf{n}_{m,k} \tag{18}$$

$$\Rightarrow \begin{bmatrix} \tilde{\mathbf{z}}_{m1,k} \\ \tilde{\mathbf{z}}_{m2,k} \end{bmatrix} = \begin{bmatrix} \mathbf{H}_{x1} & \mathbf{H}_{f1} \\ \mathbf{H}_{x2} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{x}}_k \\ {}^{G}\tilde{\mathbf{p}}_f \end{bmatrix} + \begin{bmatrix} \mathbf{n}_{f1} \\ \mathbf{n}_{f2} \end{bmatrix} \tag{19}$$

where $\mathbf{n}_{fi} \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_{fi}), i \in \{1,2\}$. Note that in the above expression $\tilde{\mathbf{z}}_{m1,k}$ and $\tilde{\mathbf{z}}_{m2,k}$ are orthonormally transformed measurement residuals, not the direct partitions of $\tilde{\mathbf{z}}_{m,k}$. With the top transformed linearized measurement residual $\tilde{\mathbf{z}}_{m1,k}$ in (19), we now perform efficient EKF update to initialize the state estimate of ${}^{G}\hat{\mathbf{p}}_{f}$ and its covariance and correlations to $\mathbf{x}_k$ [see (15)], which will then be augmented to the current state and covariance matrix.

$$
\begin{aligned}
{}^{G}\hat{\mathbf{p}}_{f} &= {}^{G}\hat{\mathbf{p}}_{f} \boxplus \mathbf{H}_{f1}^{-1}\tilde{\mathbf{z}}_{m1,k} && (20)\\
\mathbf{P}_{xf} &= -\mathbf{P}_{k}\mathbf{H}_{x1}^{\top}\mathbf{H}_{f1}^{-\top} && (21)\\
\mathbf{P}_{ff} &= \mathbf{H}_{f1}^{-1}\left(\mathbf{H}_{x1}\mathbf{P}_{k}\mathbf{H}_{x1}^{\top} + \mathbf{R}_{f1}\right)\mathbf{H}_{f1}^{-\top} && (22)
\end{aligned}
$$

It should be noted that a full-rank $\mathbf{H}_{f1}$ is needed to perform the above initialization, which normally is the case if enough measurements are collected (i.e., delayed initialization). Note also that to utilize all available measurement information, we also perform EKF update using the bottom measurement residual $\tilde{\mathbf{z}}_{m2,k}$ in (19), which essentially is equivalent to the Multi-State Constraint Kalman Filter (MSCKF) [25] update with nullspace projection [34].

C. Landmark Update

We generalize the landmark measurement model as a series of nested functions to encompass different feature parameterizations such as 3D position and inverse depth and so on. Assuming a visual feature that has been tracked over the sliding window of stochastic clones [35], we can write the visual-bearing measurements (i.e., pixel coordinates) as the following series of nested functions:

$$
\begin{aligned}
\mathbf{z}_{m,k} &= h(\mathbf{x}_k) + \mathbf{n}_{m,k} && (23)\\
&= h_d\left(\mathbf{z}_{n,k}, \boldsymbol{\zeta}\right) + \mathbf{n}_{m,k} && (24)\\
&= h_d\left(h_p\left({}^{C_k}\mathbf{p}_f\right), \boldsymbol{\zeta}\right) + \mathbf{n}_{m,k} && (25)\\
&= h_d\left(h_p\left(h_t\left({}^{G}\mathbf{p}_f, {}^{C_k}_{G}\mathbf{R}, {}^{G}\mathbf{p}_{C_k}\right)\right), \boldsymbol{\zeta}\right) + \mathbf{n}_{m,k} && (26)
\end{aligned}
$$

where $\mathbf{z}_{m,k}$ is the raw uv pixel coordinate; $\mathbf{n}_{m,k}$ the raw pixel noise and typically assumed to be zero-mean white Gaussian; $\mathbf{z}_{n,k}$ is the normalized undistorted uv measurement; ${}^{C_k}\mathbf{p}_f$ is the landmark position in the current camera frame; ${}^{G}\mathbf{p}_f$ is the landmark position in the global frame and depending on its representation may also be a function of state elements; and $\left\{{}^{C_k}_{G}\mathbf{R}, {}^{G}\mathbf{p}_{C_k}\right\}$ denotes the current camera pose (position and orientation) in the global frame.

The measurement functions h_d, h_p, and h_t correspond to the intrinsic distortion, projection, and transformation functions and the corresponding measurement Jacobians can be computed through a simple chain rule. Note that we compute the errors on the raw uv pixels to allow for calibration of the camera intrinsics ζ and that the function h_d can be changed to support any camera model (e.g., radial-tangential and equidistant). We refer readers to the documentation website for the details of these measurement functions.

D. Online Calibration

We perform online spatiotemporal calibration of the camera-IMU time offset and extrinsic transformation, and camera intrinsics. Looking at the landmark measurement (26), one can simply take the derivative with respect to the desired variables that they wish to calibrate online. In this case we will have additional Jacobians for the intrinsic ζ in function h_d and $\{{}^{C}_{I}\mathbf{R},\ {}^{C}\mathbf{p}_I\}$ extrinsics that the global pose $\left\{{}^{C_k}_{G}\mathbf{R},\ {}^{G}\mathbf{p}_{C_k}\right\}$ is a function of. For derivations and Jacobian results, we refer the reader to our documentation.

We also co-estimate the time offset between the camera and IMU, which can commonly exist in low-cost devices due to sensor latency, clock skew, or data transmission delays. Consider the time ${}^{C}t$ as expressed in the camera clock is related to the same instant represented in the IMU clock, ${}^{I}t$, by a time offset ${}^{C}t_I$:

$$ {}^{I}t = {}^{C}t + {}^{C}t_I \tag{27}$$

This offset is unknown and estimated online. We refer the reader to [23] for further details.

E. Codebase Documentation

It is our belief that the documentation of this work in itself is one of the main contributions to the research community. Both researchers and practitioners with little background in estimation may struggle to grasp the core theoretical concepts and important implementation details when it comes to

visual-inertial estimation algorithms. To bridge this gap the documentation of this codebase takes as much of a priority as new features that could improve the estimation performance. As compared to existing open sourced systems with limited documentation, we focus on providing

additional dedicated derivation pages on how different parts of the code are derived and interact. The in-code and page documentation is automatically generated from the codebase using Doxygen [36] which is then post-processed using m.css [37] to provide high quality search functionality and mobile friendly layout. This tight-coupling of our documentation and derivations within the codebase also ensures that the documentation is up to date and that developers can easily find answers.

IV. VISUAL-INERTIAL SIMULATOR

We now detail how our simulator generates visual-inertial measurements. We note that this simulator can be easily extended to include other measurements besides the inertial and visual-bearing measurements presented below.

A. B-Spline Interpolation

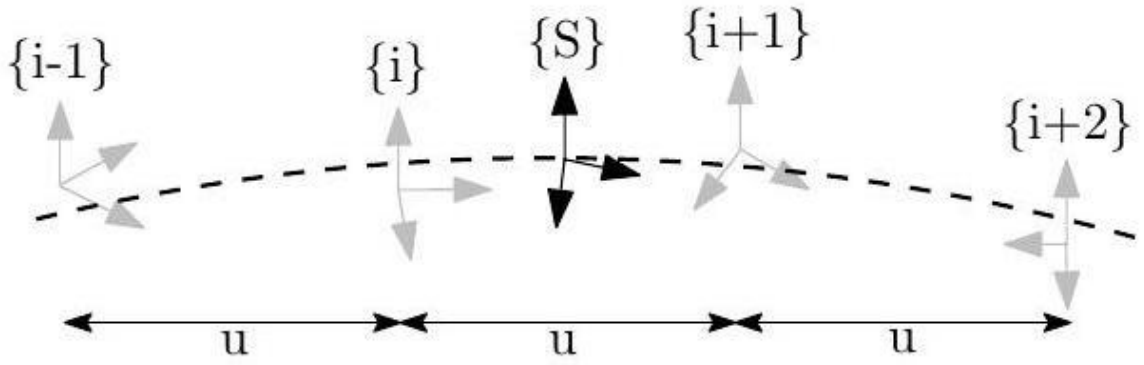

Fig. 1: Illustrate the B-spline interpolation to a pose ${}^{G}_{S}\mathbf{T}$ which is bounded by four control poses.

At the center of the simulator is an $\mathbb{SE}(3)$ B-spline which allows for the calculation of the pose, velocity, and accelerations at any given timestep along a given trajectory. We follow the work of Patron-Perez et al. [38] and Mueggler et al. [39] in which given a series of temporally uniformly distributed "control point" poses, the pose $\{S\}$ at a given timestep t_s can be interpolated by:

$$ {}^{G}_{S}\mathbf{T}\big(u(t_s)\big) = {}^{G}_{i-1}\mathbf{T}\mathbf{A}_0\mathbf{A}_1\mathbf{A}_2 \tag{28} $$

$$ \mathbf{A}_j = \exp\left(B_j(u(t))\,{}^{i-1+j}_{i+j}\mathbf{\Omega}\right) \tag{29} $$

$$ {}^{i-1}_{i}\mathbf{\Omega} = \log\left({}^{G}_{i-1}\mathbf{T}^{-1}\,{}^{G}_{i}\mathbf{T}\right) \tag{30} $$

where $B_j(u(t))$ are our spline interpolation constants, $\exp(\cdot), \log(\cdot)$ are the $\mathbb{SE}(3)$ matrix exponential and $\log$ arithm, and the frame notations are shown in Figure 1. Equation (28) can be interpreted as compounding the fraction portions of the bounding poses to the first pose ${}^{G}_{i-1}\mathbf{T}$. It is then simple to take the time derivative to allow the computation of the velocity and acceleration at any point. The only needed input into the simulator is a pose trajectory which we uniformly sample to construct control points for the B-spline. This B-spline is then used to both generate the inertial measurements while also providing the pose information needed to generate visual-bearing measurements.

B. Inertial Measurements

To incorporate inertial measurements from an IMU sensor, we can leverage the continuous nature and C^2-continuity of

our cubic B-spline. To obtain the true measurements from our $\mathrm{SE}(3)$ B-spline we can do the following:

$$ {}^{I}\boldsymbol{\omega}(t) = \mathrm{vee}({}^{G}_{I}\mathbf{R}(u(t))^{\top}\,{}^{G}_{I}\dot{\mathbf{R}}(u(t))) \tag{31} $$

$$ {}^{I}\mathbf{a}(t) = {}^{G}_{I}\mathbf{R}(u(t))^{\top G}\ddot{\mathbf{p}}_{I}(u(t)) \tag{32} $$

where vee($\cdot$) returns the vector portion of the skewsymmetric matrix. These are then corrupted using the random walk biases and corresponding white noises.

C. Visual-Bearing Measurement

After creating the B-spline trajectory we generate environmental landmarks that can be later projected into the synthetic camera frames. To generate these landmarks, we increment along the spline at a fixed interval and ensure that all cameras see enough landmarks in the map. If there are not enough landmarks in the given camera frame, we generate new landmarks by sending out random rays from the camera and assigning a random depth. Landmarks are then added to the map so that they can be projected into future frames. We generate landmarks' visual measurements by projecting them into the current frame. Projected landmarks are limited to being within the field of view, in front, and close in distance to the camera. Pixel noise can be directly added to the true pixel values.

V. BENCHMARKS

A. Simulation Results

With the proposed visual-inertial simulator, we evaluate the proposed online calibration and the consistency of our MSCKF estimator, which is implemented based on the First Estimate Jacobians (FEJ)-EKF [21], [22]. In particular, the system is run with a monocular camera, a window size of 11, a maximum of 100 feature tracks per frame, and a maximum of 50 SLAM landmarks kept in the state, [1] along with VIO feature tracks that are processed by the MSCKF update. The camera is simulated at 10 Hz while the IMU is simulated at 400 Hz . We inject one pixel noise and the IMU noise characteristics of an ADIS16448 MEMS IMU. To simulate bad initial calibration values, we randomly initialize the calibration values using the prior distribution values of the estimator. This ensures that during Monte-Carlo simulation we have both different measurement noises and initial calibration values for each run.

As summarized in Table I, the average Absolute Trajectory Error (ATE) and Normalized Estimation Error Squared (NEES) for each different scenario shows that when performing online calibration, estimation accuracy does not degrade if we are given the true calibration; while in the case that we have bad initial guesses, the estimator remains consistent and is able to estimate with reasonable accuracy. A representative run with uncertainty bounds is shown in Figure 3. When calibration is disabled and a bad initial guess is used, the NEES becomes large due to not modeling the uncertainty that these calibration parameters have, and in many cases the estimate diverges. We also plot the first ten and sixty

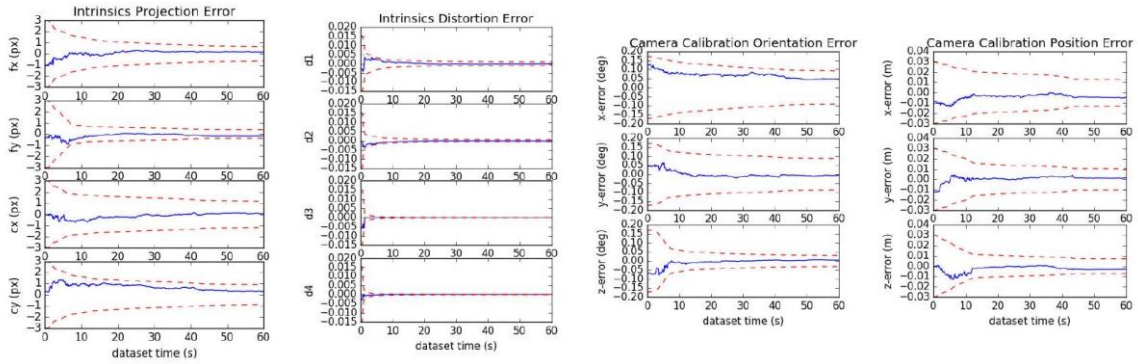

Fig. 2: Camera intrinsic projection and distortion along with extrinsic orientation and positions parameters error (blue-solid) and 3σ bounds (red-dashed) for a representative run. Note that we only plot the first sixty seconds of the dataset.

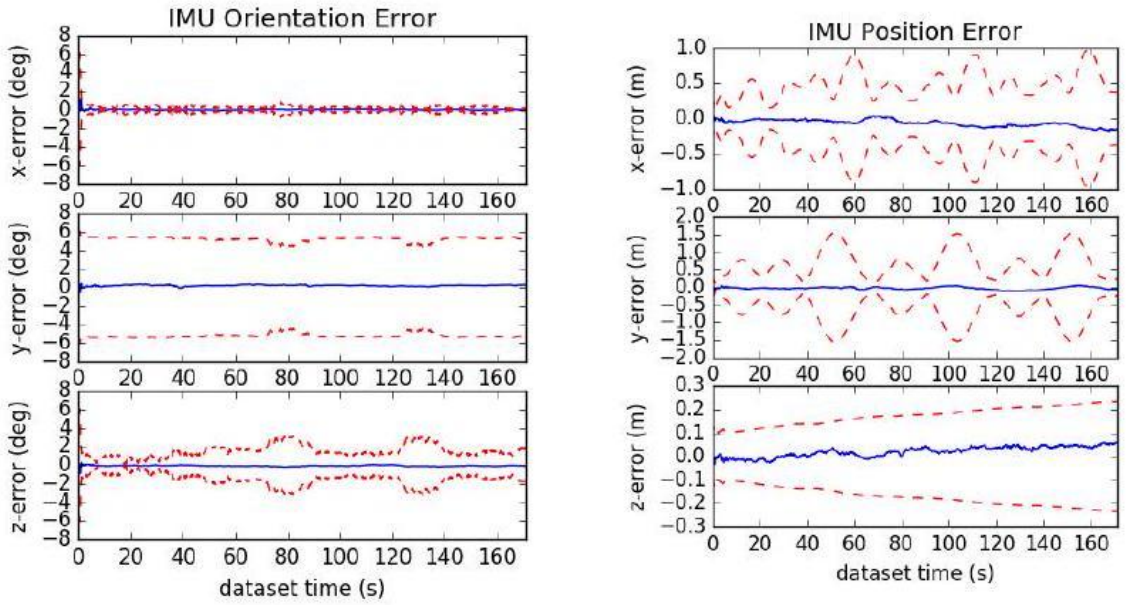

Fig. 3: IMU pose errors (blue-solid) and 3σ bounds (reddashed) for a representative run of the proposed method with SLAM landmarks and online calibration.

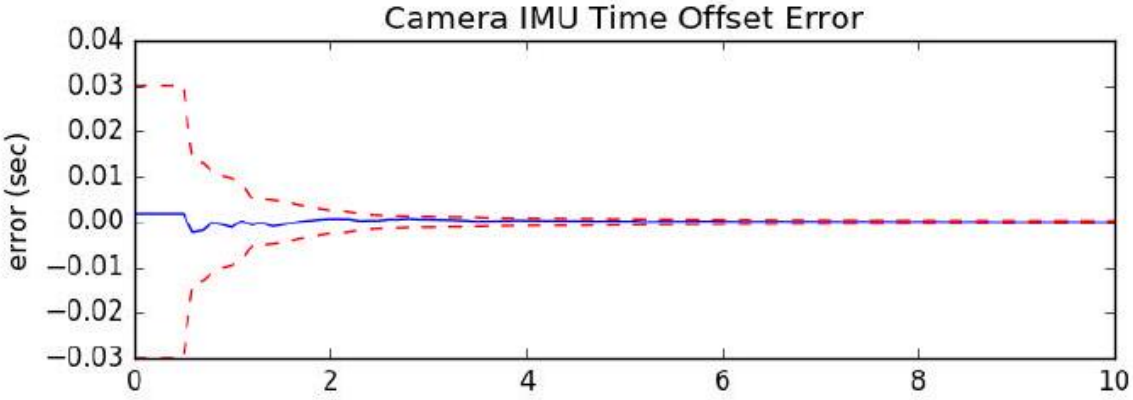

Fig. 4: Camera to IMU time offset error (blue-solid) and 3σ bounds (red-dashed) for a representative run.

seconds of all calibration parameters of a representative run in Figures 2 and 4, showing that these parameters rapidly converge from their initially poor guesses.

B. Real-World Comparison

We evaluate the proposed visual-inertial FEJ-MSCKF estimator with and without SLAM landmarks on the Vicon room scenarios from the EurocMav dataset [41] which provides both 20 Hz stereo images, 200 Hz ADIS16448 MEMS IMU measurements, and optimized groundtruth trajectories. It should be noted that we have recalculated the V1_01_easy groundtruth due to the original having incorrect orientation values

and have provided this corrected groundtruth trajectory to the community on our documentation website. All methods were run with the configuration files from their open sourced repositories with each algorithm being run ten times on each dataset to compensate for some randomness inherent to the visual front-ends. In this benchmarking test, we eval-

TABLE I: Average ATE and NEES over twenty runs with true or bad calibration, with and without online calibration.

	ATE (deg)	ATE (m)	Ori. NEES	Pos. NEES
true w/ calib	0.212	0.134	2.203	1.880
true w/o calib	0.200	0.128	2.265	1.909
bad w/ calib	0.218	0.139	2.235	2.007
bad w/o calib	5.432	508.719	9.159	1045.174

uate the following state-of-the-art visual-inertial estimation algorithms:

OKVIS [2] - Keyframe-based fixed-lag smoother which optimizes arbitrarily spaced keyframe poses connected with inertial measurement factors and environmental landmarks. A fixed window size was enforced to ensure computational feasibility with the focus on selective marginalization to allow for problem sparsity.

VINS-Fusion VIO [3] - Extension of the original VINSMono [42] sliding optimization-based method that leverages IMU preintegration which is then loosely coupled with a secondary pose-graph optimization. VINS-Fusion extends the original codebase to support stereo cameras.

Basalt VIO [4] - Stereo keyframe-based fix-lag smoother with custom feature tracking frontend with focus on extracting relevant information from the VIO for later offline visual-inertial mapping.

R-VIO [5] - Robocentric MSCKF-based algorithm which estimates in a local frame and updates the global frame through a composition step. The direction of gravity is also estimated within the filter.

ROVIO [6] - We use the ROVIO implementation within maplab [43], which is a monocular iterative EKF-based approach that performs minimization on the direct image intensity patches allowing for tracking of non-corner features such as high gradient lines.

ICE-BA [7] - Stereo incremental bundle adjustment (BA) method which optimizes both a local siding window and global optimization problem in parallel. They exploited the sparseness of their formulation and introduced a relative marginalization procedure.

S-MSCKF [8] - An open sourced implementation of original

TABLE II: Ten runs mean absolute trajectory error (ATE) for each algorithm in units of degree/meters. Note that V2_03 dataset is excluded due the inability for some algorithms to run on it. Green denotes the best, while blue is second best.

	V1_01_easy	V1_02_medium	V1_03_difficult	V2_01_easy	V2_02_medium	Average
mono_ov_slam	0.699 / 0.058	1.675 / 0.076	2.542 / 0.063	0.773 / 0.124	1.538 / 0.074	1.445 / 0.079
mono_ov_vio	0.642 / 0.076	1.766 / 0.096	2.391 / 0.344	1.164 / 0.121	1.248 / 0.106	1.442 / 0.148

mono_okvis	0.823 / 0.090	2.082 / 0.146	4.122 / 0.222	0.826 / 0.117	1.704 / 0.197	1.911 / 0.154
mono_rovioli	2.249 / 0.153	1.635 / 0.131	3.253 / 0.158	1.455 / 0.106	1.678 / 0.153	2.054 / 0.140
mono_rvio	0.994 / 0.094	2.288 / 0.129	1.757 / 0.147	1.735 / 0.144	1.690 / 0.233	1.693 / 0.149
mono_vinsfusion_vio	1.199 / 0.064	3.542 / 0.103	5.934 / 0.202	1.585 / 0.073	2.370 / 0.079	2.926 / 0.104
stereo_ov_slam	0.856 / 0.061	1.813 / 0.047	2.764 / 0.059	1.037 / 0.056	1.292 / 0.047	1.552 / 0.054
stereo_ov_vio	0.905 / 0.061	1.767 / 0.056	2.339 / 0.057	1.106 / 0.053	1.151 / 0.048	1.454 / 0.055
stereo_basalt	0.654 / 0.035	2.067 / 0.059	2.017 / 0.085	0.981 / 0.046	0.888 / 0.059	1.321 / 0.057

stereo_iceba	0.909 / 0.059	2.574 / 0.120	3.206 / 0.137	1.819 / 0.128	1.212 / 0.116	1.944 / 0.112
stereo_okvis	0.603 / 0.039	1.963 / 0.079	4.117 / 0.122	0.834 / 0.075	1.201 / 0.092	1.744 / 0.081
stereo_smsckf	1.108 / 0.086	2.147 / 0.121	3.918 / 0.198	1.181 / 0.083	2.142 / 0.164	2.099 / 0.130
stereo_vinsfusion_vio	1.073 / 0.054	2.695 / 0.089	3.643 / 0.132	2.499 / 0.071	2.006 / 0.074	2.383 / 0.084

TABLE III: Relative pose error (RPE) for different segment lengths for each algorithm variation over all datasets in units of degree/meters. Note that V2_03 dataset is excluded due the inability for some algorithms to run on it.

	8m	16m	24m	**32m**	40m	48m
mono_ov_slam	0.661 / 0.074	0.802 / 0.086	0.979 / 0.097	1.061 / 0.105	1.145 / 0.120	1.289 / 0.122
mono_ov_vio	0.826 / 0.094	1.039 / 0.106	1.215 / 0.111	1.283 / 0.132	1.342 / 0.151	1.425 / 0.184
mono_okvis	0.662 / 0.107	0.870 / 0.161	1.031 / 0.190	1.225 / 0.213	1.384 / 0.240	1.603 / 0.251

mono_rovioli	1.136 / 0.095	1.585 / 0.135	1.847 / 0.184	2.078 / 0.226	2.218 / 0.263	2.402 / 0.295
mono_rvio	0.705 / 0.130	0.902 / 0.160	1.029 / 0.183	1.074 / 0.213	0.991 / 0.227	1.077 / 0.232
mono_vinsfu-sion_vio	0.940 / 0.070	1.298 / 0.103	1.680 / 0.118	1.822 / 0.146	1.833 / 0.153	1.860 / 0.171
stereo_ov_slam	0.685 / 0.069	0.876 / 0.080	1.064 / 0.087	1.169 / 0.087	1.275 / 0.098	1.488 / 0.105
stereo_ov_vio	0.722 / 0.068	0.892 / 0.077	1.089 / 0.087	1.218 / 0.088	1.342 / 0.101	1.489 / 0.106
stereo_basalt	0.538 / 0.063	0.576 / 0.070	0.649 / 0.078	0.715 / 0.086	0.647 / 0.097	0.758 / 0.111
stereo_iceba	0.955 / 0.096	1.227 / 0.114	1.415 / 0.120	1.658 / 0.152	1.856 / 0.173	1.803 / 0.180
stereo_okvis	0.611 / 0.066	0.772 / 0.089	0.916 / 0.103	1.089 / 0.119	1.173 / 0.136	1.404 / 0.141
stereo_smsckf	1.084 / 0.098	1.462 / 0.136	1.578 / 0.159	1.667 / 0.187	1.901 / 0.200	2.134 / 0.217
stereo_vinsfu-sion_vio	0.946 / 0.057	1.357 / 0.079	1.721 / 0.097	1.928 / 0.111	1.935 / 0.125	1.805 / 0.132

MSCKF [25] paper with stereo feature tracking and a focus on high-speed motion scenarios.

Note that we evaluate only the VIO portion of these codebases (i.e., not the non-realtime backend pose graph thread output of VINS-Fusion [3] and visual-inertial mapping of Basalt [4]), as one could simply append a pose graph optimizer after any of these odometry methods to improve long-term accuracy.

Table II shows the average ATE of all methods for each dataset. It is clear that the addition of SLAM landmarks in our OpenVINS greatly reduces the drift in the monocular case, while it has a smaller impact on the stereo performance; and more importantly, OpenVINS is able to perform competitively to other methods. We additionally compared the Relative Pose Error (RPE) of all methods. Shown in Table III, our monocular system clearly outperforms the current open sourced codebases, with our stereo system being able to perform second to Basalt. While we did not evaluate perframe timing rigorously, we found that Basalt outperformed all other algorithms, with our proposed method being limited by the visual-frontend implementation from OpenCV [44] and SLAM feature update equally. On the first EurocMav dataset we could process at 2.7x/4.3x and 1.2x/1.9x realtime

for our monocular SLAM/VIO, and stereo SLAM/VIO, respectively, on an Intel(R) Xeon(R) CPU E3-1505M v6 @ 3.00 GHz processor in single threaded execution.

VI. CONCLUSION AND FUTURE WORK

In this paper we have presented our OpenVINS (OV) system as a platform for the research community. At the core we provide the visual processing frontend, full visual-inertial simulator, and modular on-manifold EKF. In particular, we have implemented the FEJ-based MSCKF with and without SLAM landmarks and demonstrated the competing performance of our estimator. We have heavily documented the project to allow for researchers and practitioners to quickly build on top of this work with minimal estimation theory background. In the future we plan to expand our system to provide a sliding window optimization-based estimator leveraging our closed-form preintegration [45]. We are also interested in integrating visual-inertial mapping and perception capabilities into OpenVINS.

References

[1] G. Huang, "Visual-inertial navigation: A concise review," in Proc. International Conference on Robotics and Automation, Montreal, Canada, May 2019.

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

[3] T. Qin, J. Pan, S. Cao, and S. Shen, "A general optimization-based framework for local odometry estimation with multiple sensors," CoRR, vol. abs/1901.03638, 2019.

[4] V. C. Usenko, N. Demmel, D. Schubert, J. Stückler, and D. Cremers, "Visual-inertial mapping with non-linear factor recovery," CoRR, vol. abs/1904.06504, 2019.

[5] Z. Huai and G. Huang, "Robocentric visual-inertial odometry," International Journal of Robotics Research, Apr. 2019, (to appear).

[6] M. Bloesch, M. Burri, S. Omari, M. Hutter, and R. Siegwart, "Iterated extended kalman filter based visual-inertial odometry using direct photometric feedback," The International Journal of Robotics Research, vol. 36, no. 10, pp. 1053-1072, 2017.

[7] H. Liu, M. Chen, G. Zhang, H. Bao, and Y. Bao, "Ice-ba: Incremental, consistent and efficient bundle adjustment for visual-inertial slam," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 1974-1982.

[8] K. Sun, K. Mohta, B. Pfrommer, M. Watterson, S. Liu, Y. Mulgaonkar, C. J. Taylor, and V. Kumar, "Robust stereo visual inertial odometry for fast autonomous flight," IEEE Robotics and Automation Letters, vol. 3, no. 2, pp. 965-972, April 2018.

[9] K. Eckenhoff, P. Geneva, J. Bloecker, and G. Huang, "Multi-camera visual-inertial navigation with online intrinsic and extrinsic calibration," in Proc. International Conference on Robotics and Automation, Montreal, Canada, May 2019.

[10] K. Eckenhoff, P. Geneva, and G. Huang, "Sensor-failure-resilient multi-imu visual-inertial navigation," in Proc. International Conference on Robotics and Automation, Montreal, Canada, May 2019.

[11] K. Eckenhoff, Y. Yang, P. Geneva, and G. Huang, "Tightly-coupled visual-inertial localization and 3D rigid-body target tracking," IEEE Robotics and Automation Letters (RA-L), vol. 4, no. 2, pp. 1541-1548, 2019.

[12] K. Eckenhoff, P. Geneva, N. Merrill, and G. Huang, "Schmidt-ekfbased visual-inertial moving object tracking," in Proc. of the IEEE International Conference on Robotics and Automation, Paris, France, 2020.

[13] P. Geneva, K. Eckenhoff, and G. Huang, "A linear-complexity EKF for visual-inertial navigation with loop closures," in Proc. International Conference on Robotics and Automation, Montreal, Canada, May 2019.

[14] P. Geneva, J. Maley, and G. Huang, "An efficient schmidt-ekf for 3D visual-inertial SLAM," in Proc. Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, June 2019, (accepted).

[15] Y. Yang, P. Geneva, X. Zuo, K. Eckenhoff, Y. Liu, and G. Huang, "Tightly-coupled aided inertial navigation with point and plane features," in Proc. International Conference on Robotics and Automation, Montreal, Canada, May 2019.

[16] Y. Yang, P. Geneva, K. Eckenhoff, and G. Huang, "Visual-inertial navigation with point and line features," Macau, China, Nov. 2019, (accepted).

[17] X. Zuo, P. Geneva, W. Lee, Y. Liu, and G. Huang, "LIC-Fusion: Lidar-inertial-camera odometry," Macau, China, Nov. 2019, (accepted).

[18] X. Zuo, P. Geneva, Y. Yang, W. Ye, Y. Liu, and G. Huang, "Visualinertial localization with prior lidar map constraints," IEEE Robotics and Automation Letters (RA-L), 2019, (to appear).

[19] Y. Yang, P. Geneva, K. Eckenhoff, and G. Huang, "Degenerate motion analysis for aided INS with online spatial and temporal calibration," IEEE Robotics and Automation Letters (RA-L), vol. 4, no. 2, pp. 20702077, 2019.

[20] G. Huang, A. I. Mourikis, and S. I. Roumeliotis, "Analysis and improvement of the consistency of extended Kalman filter-based SLAM," in Proc. of the IEEE International Conference on Robotics and Automation, Pasadena, CA, May 19-23 2008, pp. 473-479.

[21] -, "A first-estimates Jacobian EKF for improving SLAM consistency," in Proc. of the 11th International Symposium on Experimental Robotics, Athens, Greece, July 14-17, 2008.

[22] -, "Observability-based rules for designing consistent EKF SLAM estimators," International Journal of Robotics Research, vol. 29, no. 5, pp. 502-528, Apr. 2010.

[23] M. Li and A. I. Mourikis, "Online temporal calibration for CameraIMU systems: Theory and algorithms," International Journal of Robotics Research, vol. 33, no. 7, pp. 947-964, June 2014.

[24] M. Li, H. Yu, X. Zheng, and A. I. Mourikis, "High-fidelity sensor modeling and self-calibration in vision-aided inertial navigation," in IEEE International Conference on Robotics and Automation (ICRA), May 2014, pp. 409-416.

[25] A. I. Mourikis and S. I. Roumeliotis, "A multi-state constraint Kalman filter for vision-aided inertial navigation," in Proceedings of the IEEE International Conference on Robotics and Automation, Rome, Italy, Apr. 10-14, 2007, pp. 3565-3572.

[26] N. Trawny and S. I. Roumeliotis, "Indirect Kalman filter for 3D attitude estimation," University of Minnesota, Dept. of Comp. Sci. & Eng., Tech. Rep., Mar. 2005.

[27] C. Hertzberg, R. Wagner, U. Frese, and L. Schröder, "Integrating generic sensor fusion algorithms with sound state representations through encapsulation of manifolds," Information Fusion, vol. 14, no. 1, pp. 57-77, 2013.

[28] K. Wu, T. Zhang, D. Su, S. Huang, and G. Dissanayake, "An invariant-ekf vins algorithm for improving consistency," in Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Sept 2017, pp. 1578-1585.

[29] J. Civera, A. Davison, and J. Montiel, "Inverse depth parametrization for monocular SLAM," IEEE Transactions on Robotics, vol. 24, no. 5, pp. 932-945, Oct. 2008.

[30] M. K. Paul, K. Wu, J. A. Hesch, E. D. Nerurkar, and S. I. Roumeliotis, "A comparative analysis of tightly-coupled monocular, binocular, and stereo VINS," in Proc. of the IEEE International Conference on Robotics and Automation, Singapore, July 2017, pp. 165-172.

[31] A. B. Chatfield, Fundamentals of High Accuracy Inertial Navigation. AIAA, 1997.

[32] F. Dellaert, "Factor graphs and gtsam: A hands-on introduction," Georgia Institute of Technology, Tech. Rep., 2012.

[33] M. Li, "Visual-inertial odometry on resource-constrained systems," Ph.D. dissertation, UC Riverside, 2014.

[34] Y. Yang, J. Maley, and G. Huang, "Null-space-based marginalization: Analysis and algorithm," in Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems, Vancouver, Canada, Sept. 24-28, 2017, pp. 6749-6755.

[35] S. I. Roumeliotis and J. W. Burdick, "Stochastic cloning: A generalized framework for processing relative state measurements," in Proceedings of the IEEE International Conference on Robotics and Automation, Washington, DC, May 11-15, 2002, pp. 1788-1795.

[36] D. Van Heesch, "Doxygen: Source code documentation generator tool," URL: http://www.doxygen.org, 2008.

[37] V. Vondruš, "m.css: A no-nonsense, no-javascript css framework and pelican theme for content-oriented websites," URL: https://mcss. mosra.cz/, 2018.

[38] A. Patron-Perez, S. Lovegrove, and G. Sibley, "A spline-based trajectory representation for sensor fusion and rolling shutter cameras," International Journal of Computer Vision, vol. 113, no. 3, pp. 208219, 2015.

[39] E. Mueggler, G. Gallego, H. Rebecq, and D. Scaramuzza, "Continuous-time visual-inertial odometry for event cameras," IEEE Transactions on Robotics, pp. 1-16, 2018.

[40] M. Li and A. I. Mourikis, "Optimization-based estimator design for vision-aided inertial navigation," in Robotics: Science and Systems, Berlin, Germany, June 2013, pp. 241-248.

[41] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. W. Achtelik, and R. Siegwart, "The euroc micro aerial vehicle datasets," The International Journal of Robotics Research, vol. 35, no. 10, pp. 1157-1163, 2016.

[42] T. Qin, P. Li, and S. Shen, "VINS-Mono: A robust and versatile monocular visual-inertial state estimator," IEEE Transactions on Robotics, vol. 34, no. 4, pp. 1004-1020, 2018.

[43] T. Schneider, M. Dymczyk, M. Fehr, K. Egger, S. Lynen, I. Gilitschenski, and R. Siegwart, "Maplab: An open framework for research in visual-inertial mapping and localization," IEEE Robotics and Automation Letters, vol. 3, no. 3, pp. 1418-1425, July 2018.

[44] OpenCV Developers Team, "Open source computer vision (OpenCV) library," Available: http://opencv.org.

[45] K. Eckenhoff, P. Geneva, and G. Huang, "Closed-form preintegration methods for graph-based visual-inertial navigation," International Journal of Robotics Research, vol. 38, no. 5, pp. 563-586, 2019.

〈文献翻译一：译文〉

OpenVINS：一种视觉惯性估计研究平台

Patrick Geneva 、 Kevin Eckenhoff 、 Woosik Lee 、 Yulin Yang 和 Guoquan Huang

摘要——本文提出一个名为 OpenVINS 的开放平台，旨在为学术界和工业界从业者提供视觉惯性估计研究的解决方案。该开源代码库为研究人员和工程师快速开发视觉惯性系统新功能奠定了基础，内置支持以下常用功能：(i)流形滑动窗口卡尔曼滤波器，（ii）在线相机本征与外在校准，（iii）相机与惯性传感器时间偏移校准，（iv）采用不同表征方式的 SLAM 地标及统一的初值雅可比矩阵（FEJ）处理，(v)模块化状态管理系统，（vi）可扩展视觉惯性系统仿真器，（vii）丰富的算法评估工具箱。此外，我们还特别注重详细文档和理论推导，以支持快速开发与研究——这些在现有开源算法中普遍缺失。最后，我们通过与最先进开源算法的全面对比验证，证明了 OpenVINS 在视觉惯性估计性能上的竞争优势。

- 开源：https://github.com/rpng/open_vins
- 文档：https://docs.openvins.com

一、引言

自主机器人和消费级移动设备（如无人机和智能手机）正变得无处不在，这主要得益于计算能力的大幅提升，以及功耗和成本的同步降低。为了让这些机器人和移动设备具备感知和理解其在局部环境中的位置信息的能力——从移动 AR/VR 到自主导航等众多应用场景都对此有需求——视觉惯性导航系统（VINS）常被用来通过融合车载摄像头和惯性传感器的数据，提供精确的运动估计[1]。

从零开始开发实用的视觉惯性导航系统（VINS）算法已被证明极具挑战性。在机器人研究领域，由于缺乏具备完整文档和详细推导过程的 VINS 代码库，即便是技术基础薄弱的研究人员也难以学习和扩展现有前沿成果来解决实际问题，这已成为科研人员面临的重要障碍。虽然目前存在多个开源视觉惯性代码库[2]-[8]，但这些代码库并非为可扩展性设计，且缺乏完善的文档说明和评估工具——根据我们的经验，这些要素对于快速开发和深入理解至关重要，从而加速该领域的 VINS 研究与开发进程。此外，这些系统存在大量硬编码假设或功能特性，需要对代码库进行复杂理解才能适配特定传感器系统。这种特性，加上文档支持不足，限制了其在不同应用场景中的广泛应用。

为填补该领域研究空白并推动视觉惯性系统（VINS）在机器人技术及其他领域的应用，本文提出了一套可扩展的开源代码库，特别针对具有有限或深厚状态估计背景知识

的研究人员和实践者。我们为视觉惯性估计领域的新手提供必要文档、工具及理论支撑，并将这套工具集命名为 OpenVINS（OV）。该代码库已成为特拉华大学研究团队近期多个视觉惯性估计项目的基石，涵盖多相机[9]、多惯性测量单元[10]、视觉惯性移动目标追踪[11][12]、基于施密特算法的视觉惯性 SLAM[13][14]、点面与点线视觉惯性导航[15][16]等项目[17]-[19]。现将 OpenVINS 各组件的核心功能总结如下：

• ov_core – 包含二维图像稀疏视觉特征跟踪；线性与高斯-牛顿特征三角化方法；适用于任意数量摄像头及频率的视觉惯性模拟器；以及基本流形数学运算与工具。

• eval 工具包 – 包含轨迹对齐功能；用于轨迹精度与一致性评估的绘图工具；不同精度指标的蒙特卡洛评估；以及将 ROS 话题记录至文件的实用功能。

•msckf——包含可扩展模块化扩展卡尔曼滤波器（EKF）滑动窗口视觉惯性估计器，采用流形型系统实现灵活状态表示。功能包括：雅可比矩阵初值估计（FEJ）[20][21][22]、IMU-相机时间偏移校准[23]、相机本征与外在在线校准[24]、标准 MSCKF [25]，以及不同表示形式的 3DSLAM 地标。

下文我们将介绍一种基于广义模态流形 EKF 的估计器，其最简形式可实现相机-惯性测量单元（IMU）组合的实时状态估计。随后我们将详细说明实现的核心功能模块，为研究者快速搭建和扩展系统奠定基础。需要说明的是，本文仅对功能模块进行简要介绍，读者可访问我们详尽的文档网站获取完整资料。我们通过仿真测试和真实数据集验证，清晰展示了该 EKF 方案相较于其他开源算法的性能优势。

二、ON-多模模块化 EKF

本视觉惯性系统的状态向量由当前惯性导航状态、一组历史 IMU 姿态克隆、一组环境地标以及一cmw组相机的外在与内在参数组成。

$$\mathbf{x}_k = [\mathbf{x}_I^\top \quad \mathbf{x}_C^\top \quad \mathbf{x}_M^\top \quad \mathbf{x}_W^\top \quad {}^C t_I]^\top \tag{1}$$

$$\mathbf{x}_I = \left[{}^{I_k}_{G}\bar{q}^\top \quad {}^G\mathbf{p}_{I_k}^\top \quad {}^G\mathbf{v}_{I_k}^\top \quad \mathbf{b}_{\omega_k}^\top \quad \mathbf{b}_{a_k}^\top\right]^\top \tag{2}$$

$$\mathbf{x}_C = \left[{}^{I_{k-1}}_{G}\bar{q}^\top \quad {}^G\mathbf{p}_{I_{k-1}}^\top \quad \cdots \quad {}^{I_{k-c}}_{G}\bar{q}^\top \quad {}^G\mathbf{p}_{I_{k-c}}^\top\right]^\top \tag{3}$$

$$\mathbf{x}_M = \left[{}^G\mathbf{p}_{f_1}^\top \quad \cdots \quad {}^G\mathbf{p}_{f_m}^\top\right]^\top \tag{4}$$

$$\mathbf{x}_W = \left[{}^{I}_{C_1}\bar{q}^\top \quad {}^{C_1}\mathbf{p}_I^\top \quad \boldsymbol{\zeta}_0^\top \cdots \quad {}^{I}_{C_w}\bar{q}^\top \quad {}^{C_w}\mathbf{p}_I^\top \quad \boldsymbol{\zeta}_w^\top\right]^\top \tag{5}$$

其中，单位四元数参数化了从全局参考系到 IMU 局部参考系的旋转，时间 t 为时刻 t，分别为陀螺仪和加速度计的偏置量，以及 IMU 在全局参考系中表达的速度和位置。惯性状态位于由单位四元数与向量空间（即）的乘积所定义的流形上，具有 15 个自由度（DOF）。 ${}^{I_k}_{G}\bar{q}\mathbf{R}\left({}^{I_k}_{G}\bar{q}\right) = {}^{I_k}_{G}\mathbf{R}\{G\}\{I_k\}k[26], \mathbf{b}_\omega\mathbf{b}_a\,{}^G\mathbf{v}_{I_k}\,{}^G\mathbf{p}_{I_k}\mathbf{x}_I\mathbb{H}\mathbb{R}^{12}\mathcal{M} = \mathbb{H} \times \mathbb{R}^{12}$

对于向量变量，“boxplus”和“boxminus”运算（将元素映射至给定流形[27]并从该流形映射回）等同于其向量的简单加减运算。对于四元数，我们将四元数 boxplus 运算定义为：

$$\bar{q}_1 \boxplus \delta\boldsymbol{\theta} \triangleq \begin{bmatrix} \frac{\delta\boldsymbol{\theta}}{2} \\ 1 \end{bmatrix} \otimes \bar{q}_1 \simeq \bar{q}_2 \tag{6}$$

需注意，尽管我们已采用左四元数误差来定义方向，但在实际应用中并不限于此，任何流形表示均可使用（例如[28]）。

环境地标图仅包含简化处理的全球三维位置$\mathbf{x}_M$，实际应用中我们支持多种表示形式（如逆 MSCKF [25]、完整逆深度[29]和锚定三维位置[30]）。

校准向量包含相机固有$\mathbf{x}_W \boldsymbol{\zeta}\, {}^C t_I$参数（由焦距、相机中心和畸变参数组成）以及相机-IMU 外在参数（即 IMU 到各相机的空间变换，即相对姿态）。由于我们采用同步相机时钟，因此在校准向量中加入了 IMU 与相机时钟之间的单一时间偏移量。

A. 传播

惯性状态通过基于$\mathbf{x}_I \, {}^I\mathbf{a}_m \, {}^I\boldsymbol{\omega}_m k-1k$以下通用非线性 IMU 运动学的线性加速度和角速度的 IMU 测量值进行前向传播，该运动学将状态从时间步传递至[31]：

$$\mathbf{x}_k = f(\mathbf{x}_{k-1}, {}^I\mathbf{a}_m, {}^I\boldsymbol{\omega}_m, \mathbf{n}) \tag{7}$$

其中包含$\mathbf{n}$IMU 测量值的零均值白高斯噪声以及随机游走偏差噪声。该状态估计值在当前估计值处进行评估：

$$\hat{\mathbf{x}}_{k|k-1} = f(\hat{\mathbf{x}}_{k-1|k-1}, {}^I\mathbf{a}_m, {}^I\boldsymbol{\omega}_m, \mathbf{0}) \tag{8}$$

其中·表示估计值，下标表示在给定时间点（截至时间$k \mid k-1kk-1$点的测量数据）下的预测估计值。状态协方差矩阵通常通过在线性化当前估计值处的非线性模型进行传播：

$$\mathbf{P}_{k|k-1} = \boldsymbol{\Phi}_{k-1}\mathbf{P}_{k-1|k-1}\boldsymbol{\Phi}_{k-1}^{\mathsf{T}} + \mathbf{Q}_{k-1} \tag{9}$$

其中和分别$\boldsymbol{\Phi}_{k-1}\mathbf{Q}_{k-1}\mathbf{x}_C\mathbf{x}_M\mathbf{x}_W$为系统雅可比矩阵与离散噪声协方差矩阵[25]。克隆体、环境特征及校准状态均不随时间演化，因此相应状态雅可比矩阵的元素为单位矩阵，传播噪声为零，可利用稀疏性实现计算节省。

B. 曲面更新

考虑以下非线性测量函数：

$$\mathbf{z}_{m,k} = h(\mathbf{x}_k) + \mathbf{n}_{m,k} \tag{10}$$

其中存在测量噪声。对于标准 EKF 更新$\mathbf{n}_{m,k} \sim \mathcal{N}\left(\mathbf{0}, \mathbf{R}_{m,k}\right)\tilde{\mathbf{x}} = \mathbf{x} \boxminus \hat{\mathbf{x}} \sim \mathcal{N}(\mathbf{0}, \mathbf{P})$，需在当前状态估计处对上述方程进行线性化处理。在本研究中，如同间接 EKF [26]所述，我们对公式（10）进行线性化处理，以当前零均值误差状态（即）为变量：

$$\mathbf{z}_{m,k} = h(\hat{\mathbf{x}}_{k|k-1} \boxplus \tilde{\mathbf{x}}_{k|k-1}) + \mathbf{n}_{m,k} \tag{11}$$

$$= h(\hat{\mathbf{x}}_{k|k-1}) + \mathbf{H}_k \tilde{\mathbf{x}}_{k|k-1} + \mathbf{n}_{m,k} \tag{12}$$

$$\Rightarrow \tilde{\mathbf{z}}_{m,k} = \mathbf{H}_k \tilde{\mathbf{x}}_{k|k-1} + \mathbf{n}_{m,k} \tag{13}$$

其中测量$\mathbf{H}_k$雅可比矩阵按以下方式计算：

$$\mathbf{H}_k = \left.\frac{\partial h(\hat{\mathbf{x}}_{k|k-1} \boxplus \tilde{\mathbf{x}}_{k|k-1})}{\partial \tilde{\mathbf{x}}_{k|k-1}}\right|_{\tilde{\mathbf{x}}_{k|k-1}=\mathbf{0}} \tag{14}$$

通过采用这种线性化测量模型，我们现在可以执行以下标准 EKF 更新，以确保更新后的状态始终保留在流形上：

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} \boxplus \mathbf{K}_k \left(\mathbf{z}_{m,k} - h(\hat{\mathbf{x}}_{k|k-1})\right) \tag{15}$$

$$\mathbf{P}_{k|k} = \mathbf{P}_{k|k-1} - \mathbf{K}_k \mathbf{H}_k \mathbf{P}_{k|k-1} \tag{16}$$

$$\mathbf{K}_k = \mathbf{P}_{k|k-1} \mathbf{H}_k^\top \left(\mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^\top + \mathbf{R}_{m,k}\right)^{-1} \tag{17}$$

三、 OPENVINS 研究平台

A. 基于类型的索引系统

OpenVINS 库的核心是基于类型的索引系统。受 GTSAM [32]等图优化框架的启发，我们让用户无需直接操作协方差，而是提供自动管理状态及其协方差的工具。这带来了诸多优势，例如缩短实现时间，以及减少因显式访问状态和协方差而产生的开发错误。

每个状态变量的“类型”内部都存储着其在误差状态中的位置信息，这些信息会在初始化、克隆或边缘化操作（这些操作会影响变量排序）时自动更新。类型由其协方差位置、当前估计值和误差状态大小共同定义。当前值不必是向量形式，但在旋转表示的情况下可以是矩阵。所有类型的误差状态均为向量形式，因此每个类型都需要定义其误差状态与流形表示之间的$\mathbb{SO}(3)$盒加映射（即更新函数）。

```
class Type {
protected:
    // Current best estimate
    Eigen :: MatrixXd _value;
    // Index of error state in covariance
    int _id = -1;
    // Dimension of error state
    int _size = -1;
    // Vector correction, how to update
```

```
    void update (const Eigen ::VectorXd dx);
};
```

这种系统架构的核心优势在于，它通过让用户构建稀疏雅可比矩阵，显著降低了新增功能的复杂度。相较于为所有状态变量构建完整雅可比矩阵，“稀疏”雅可比矩阵只需包含测量值所依赖的状态变量。这种设计不仅在测量仅涉及少数状态变量时大幅节省计算资源，还使得测量函数能够保持状态无关特性——只要相关状态变量存在，测量函数就能不受具体状态变量限制。

B. 状态变量初始化

基于一组线性化测量方程（13），我们的目标是通过最优计算新状态变量的初始估计值及其与现有状态变量的协方差和相关性。作为示例，本文将介绍如何初始化新的SLAM 地标点，其核心逻辑可应用于任何新状态变量，并能通过代码库实现类型泛化。参照文献[33]的方法 ${}^{G}\mathbf{p}_f$ ${}^{G}\mathbf{p}_f$，我们首先进行 QR 分解（例如采用计算效率高的原地吉文斯旋转），将线性系统（13）拆分为两个子系统：(i)依赖新状态的子系统（即），以及（ii）不依赖新状态的子系统。

$$\tilde{\mathbf{z}}_{m,k} = [\mathbf{H}_x \quad \mathbf{H}_f] \begin{bmatrix} \tilde{\mathbf{x}}_k \\ {}^{G}\tilde{\mathbf{p}}_f \end{bmatrix} + \mathbf{n}_{m,k} \tag{18}$$

$$\Rightarrow \begin{bmatrix} \tilde{\mathbf{z}}_{m1,k} \\ \tilde{\mathbf{z}}_{m2,k} \end{bmatrix} = \begin{bmatrix} \mathbf{H}_{x1} & \mathbf{H}_{f1} \\ \mathbf{H}_{x2} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{x}}_k \\ {}^{G}\tilde{\mathbf{p}}_f \end{bmatrix} + \begin{bmatrix} \mathbf{n}_{f1} \\ \mathbf{n}_{f2} \end{bmatrix} \tag{19}$$

其中。需$\mathbf{n}_{fi} \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_{fi}), i \in \{1,2\}\tilde{\mathbf{z}}_{m1,k}\tilde{\mathbf{z}}_{m2,k}\tilde{\mathbf{z}}_{m,k}\tilde{\mathbf{z}}_{m1,k}$ ${}^{G}\hat{\mathbf{p}}_f\mathbf{x}_k$注意，在上述表达式中，和是正交归一化变换后的测量残差，而非的直接分解。基于（19）式中的顶部线性化测量残差，我们通过高效 EKF 更新来初始化和的状态估计及其协方差与相关性[参见（15）]，随后将其扩展为当前状态与协方差矩阵。

$${}^{G}\hat{\mathbf{p}}_f = {}^{G}\hat{\mathbf{p}}_f \boxplus \mathbf{H}_{f1}^{-1}\tilde{\mathbf{z}}_{m1,k} \tag{20}$$

$$\mathbf{P}_{xf} = -\mathbf{P}_k\mathbf{H}_{x1}^{\top}\mathbf{H}_{f1}^{-\top} \tag{21}$$

$$\mathbf{P}_{ff} = \mathbf{H}_{f1}^{-1}\left(\mathbf{H}_{x1}\mathbf{P}_k\mathbf{H}_{x1}^{\top} + \mathbf{R}_{f1}\right)\mathbf{H}_{f1}^{-\top} \tag{22}$$

需要指出的是，执行上述初始化需要$\mathbf{H}_{f1}\tilde{\mathbf{z}}_{m2,k}$完整秩，这通常在收集足够测量数据时成立（即延迟初始化）。还需注意的是，为充分利用所有可用测量信息，我们还采用公式（19）中的底部测量残差进行 EKF 更新，这本质上等同于采用零空间投影[34]的多状态约束卡尔曼滤波器（MSCKF）[25]更新。

C. 地标更新

我们将标志测量模型推广为一系列嵌套函数，以涵盖不同的特征参数化方式，例如三维位置和逆深度等。假设在随机克隆的滑动窗口[35]中已追踪到视觉特征，我们可以将承载视觉特征的测量值（即像素坐标）表示为以下嵌套函数序列：

$$\mathbf{z}_{m,k} = h(\mathbf{x}_k) + \mathbf{n}_{m,k} \tag{23}$$

$$= h_d(\mathbf{z}_{n,k}, \boldsymbol{\zeta}) + \mathbf{n}_{m,k} \tag{24}$$

$$= h_d(h_p({}^{C_k}\mathbf{p}_f), \boldsymbol{\zeta}) + \mathbf{n}_{m,k} \tag{25}$$

$$= h_d\left(h_p\left(h_t({}^{G}\mathbf{p}_f, {}^{C_k}_{G}\mathbf{R}, {}^{G}\mathbf{p}_{C_k})\right), \boldsymbol{\zeta}\right) + \mathbf{n}_{m,k} \tag{26}$$

其中，原始$\mathbf{z}_{m,k}\mathbf{n}_{m,k}\mathbf{z}_{n,k}$ ${}^{C_k}\mathbf{p}_f$ ${}^{G}\mathbf{p}_f\{{}^{C_k}_{G}\mathbf{R}, {}^{G}\mathbf{p}_{C_k}\}$uv 像素坐标表示原始像素噪声（通常假设为零均值白高斯分布）；归一化无畸变 uv 测量值；当前相机帧中的标志点位置；全局坐标系中的标志点位置（根据其表示方式可能还包含状态元素函数）；以及表示当前相机在全局坐标系中的姿态（位置与方向）。

测量函数对应于相机的固有畸$h_d, h_p h_t \zeta h_d$变、投影和变换函数，其对应的测量雅可比矩阵可通过简单的链式法则计算得出。需要说明的是，我们通过计算原始 uv 像素的误差来实现相机固有参数的校准，且该函数可灵活调整以适配不同相机模型（如径向-切向型和等距型）。关于这些测量函数的具体实现细节，读者可查阅文档网站获取详细说明。

D. 在线校准

我们对相机-惯性测量单元（IMU）的时间偏移量、外在变换及相机内禀参数进行在线时空校准。通过地标测量（26）数据，可直接对需要在线校准的变量求导。在此情况下，我们将获得内禀参数与全局姿态相关外在参数的附加雅可比矩阵。关于推导过程及雅可比矩阵结果，读者可参阅我们的技术文档。$\zeta h_d\{{}^{C}_{I}\mathbf{R}, {}^{C}\mathbf{p}_I\}\{{}^{C_k}_{G}\mathbf{R}, {}^{G}\mathbf{p}_{C_k}\}$

我们还对相机与惯性测量单元（IMU）之间的时间偏移进行联合估计，这种偏移常见于低成本设备中，通常由传感器延迟、时钟偏差或数据传输延迟引起。相机时钟所表示的时间与 IMU 时钟所表示的同一时刻之间，存在一个时间偏移：${}^{C}t\,{}^{I}t\,{}^{C}t_I$

$${}^{I}t = {}^{C}t + {}^{C}t_I \tag{27}$$

该偏移量未知，需在线估算。更多细节请参阅文献[23]。

E. 代码库文档

我们坚信，这项工作的文献记录本身就是对研究界的重要贡献。对于缺乏估计背景的研究人员和实践者而言，当涉及核心理论概念和重要实施细节时，可能难以理解。

视觉惯性估计算法。为弥补这一技术空白，本代码库的文档编写与新功能开发同等重要，所有能提升性能的改进都会优先推进。相较于文档简陋的开源系统，我们特别打造了详尽的代码推导页面，清晰展示各模块的实现原理与协同机制。代码库中的文档通过Doxygen[36]自动生成，再经 m.css[37]后处理，实现智能搜索和移动端友好界面。这种代码与文档的深度耦合，既保证了文档的实时更新，也让开发者能轻松获取所需技术解答。

四、视觉惯性模拟器

现详细阐述本模拟器如何生成视觉-惯性测量数据。需指出的是，该模拟器可轻松扩展以纳入除下文所述惯性测量与视觉方位测量之外的其他测量数据。

A. B 样条插值

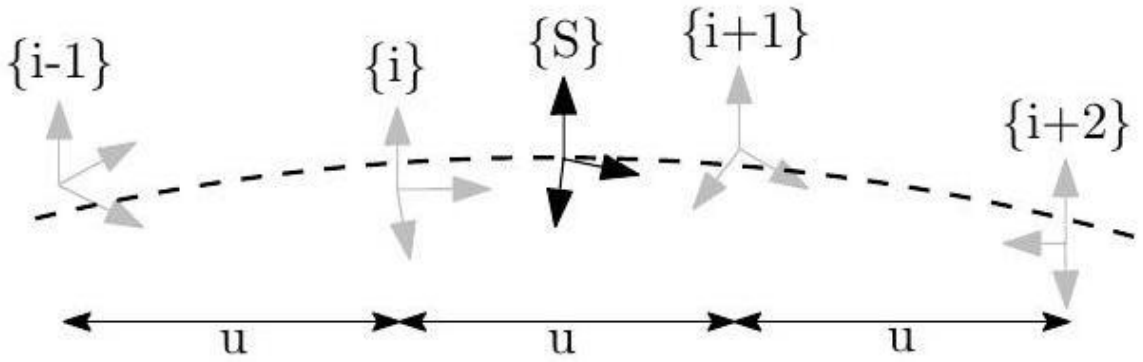

图 1：展示由四个控制姿态限定的 B 样条插值到姿态的过程。$\underset{S}{G}\mathbf{T}$

该模拟器的核心是一个 B 样条，可$\mathbb{SE}(3)\{S\}t_s$沿给定轨迹在任意时间步长计算姿态、速度和加速度。我们遵循 Patron-Perez 等人[38]和 Mueggler 等人[39]的研究方法，即给定一系列时间均匀分布的“控制点”姿态时，可通过以下方式插值计算特定时间步长的姿态：

$$\underset{S}{G}\mathbf{T}\big(u(t_s)\big) = \underset{i-1}{G}\mathbf{T}\mathbf{A}_0\mathbf{A}_1\mathbf{A}_2 \tag{28}$$

$$\mathbf{A}_j = \exp\left(B_j(u(t))\,\underset{i+j}{\overset{i-1+j}{}}\boldsymbol{\Omega}\right) \tag{29}$$

$$\underset{i}{\overset{i-1}{}}\boldsymbol{\Omega} = \log\left(\underset{i-1}{G}\mathbf{T}^{-1}\,\underset{i}{G}\mathbf{T}\right) \tag{30}$$

其中，我们$B_j(u(t))\exp(\cdot),\log(\cdot)\mathbb{SE}(3)\log\underset{i-1}{G}\mathbf{T}$的样条插值常数、矩阵指数和算术运算符的符号表示如图 1 所示。公式（28）可理解为将边界位姿的分数部分累加到初始位姿上。通过计算其时间导

数，即可轻松获得任意点的速度和加速度。模拟器仅需输入位姿轨迹数据，我们通过均匀采样该轨迹来构建 B 样条的控制点。该 B 样条不仅用于生成惯性测量数据，同时还能提供生成视觉方位测量所需的位置信息。

B. 惯性测量

为整合来自惯性测量单元（IMU）传感器的惯性测量数据，我们可以利用其连续性特征及-连续性C^2

我们的三次 B 样条。为从 B 样条中获取真实测量值，可执行$\mathrm{SE}(3)$以下操作：

$$^{I}\boldsymbol{\omega}(t) = \mathrm{vee}({}^{G}_{I}\mathbf{R}(u(t))^{\top}\,{}^{G}_{I}\dot{\mathbf{R}}(u(t))) \tag{31}$$

$$^{I}\mathbf{a}(t) = {}^{G}_{I}\mathbf{R}(u(t))^{\top G}\ddot{\mathbf{p}}_{I}(u(t)) \tag{32}$$

其中 vee（·）返回斜对称矩阵的向量部分。随后这些向量部分通过随机游走偏差及相应白噪声进行扰动。

C. 视觉-方位测量

在生成 B 样条轨迹后，我们会创建环境地标点，这些地标点可后续投影到合成相机帧中。生成地标点时，我们沿样条曲线以固定间隔递增，并确保所有相机在地图中都能捕捉到足够地标点。若某相机帧中的地标点不足，我们通过从相机发射随机光线并赋予随机深度值来生成新地标点。这些地标点随后被添加到地图中，以便投射到后续帧中。我们通过将地标点投影到当前帧来生成其视觉测量值。投影后的地标点需满足三个条件：位于视野范围内、处于前方且距离相机较近。像素噪声可直接叠加到真实像素值上。

五、基准

A. 模拟结果

通过所提出的视觉惯性模拟器，我们评估了在线校准方法及基于第一估计雅可比矩阵（FEJ）- EKF [21][22]实现的 MSCKF 估计器的一致性。具体而言，系统采用单目相机运行，窗口尺寸为 11，每帧最多保留 100 条特征轨迹，状态中最多保留 50 个 SLAM 地标，并通过 MSCKF 更新处理 VIO 特征轨迹。相机以 10Hz 频率模拟，惯性测量单元（IMU）以 400Hz 频率模拟。我们注入单像素噪声 ^{1}，并采用 ADIS16448 MEMS 惯性测量单元的噪声特性。为模拟初始校准值的偏差，我们使用估计器的先验分布值随机初始化校准参数。这确保在蒙特卡洛模拟过程中，每次运行都具有不同的测量噪声和初始校准值。

如表 I 所示，各场景的平均绝对轨迹误差（ATE）与归一化估计误差平方（NEES）表明：在线校准时，若获得真实校准数据，估计精度不会下降；而当初始猜测存在偏差时，估计器仍能保持稳定，

可获得合理精度的估计结果。图 3 展示了包含不确定度边界的典型运行案例。当校准功能被禁用且初始猜测存在偏差时，由于未对校准参数的不确定性进行建模， NEES 值会显著增大，且多数情况下估计结果会出现发散现象。我们还绘制了前 10 次和 60 次

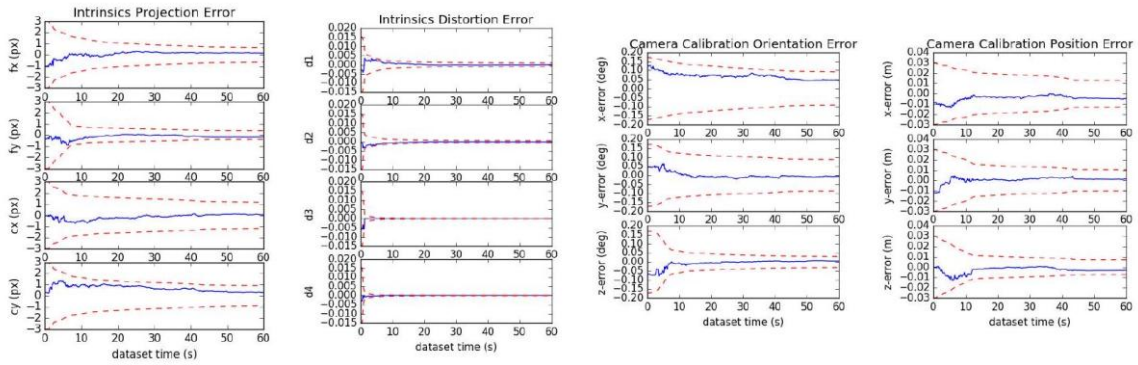

图 2：代表性运行中相机固有投影与畸变以及外在方向与位置参数误差（蓝色实线）与边界（红色虚线）的对比。需注意3σ，我们仅绘制了数据集的前 60 秒。

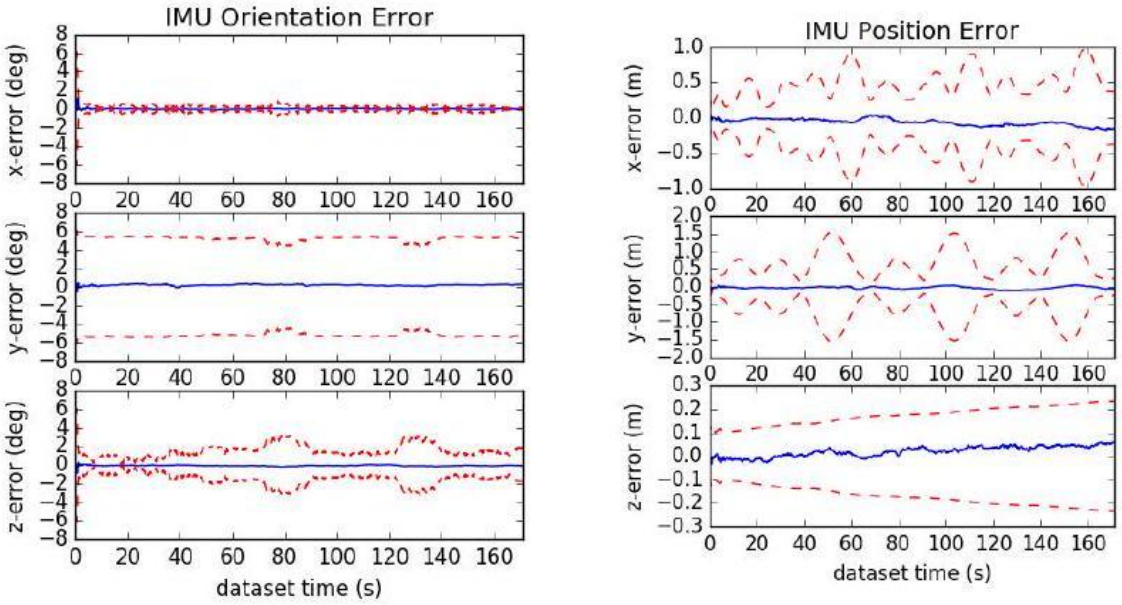

图 3：采用 SLAM 地标和在线校准的所3σ提方法代表性运行中 IMU 姿态误差（蓝色实线）与边界（红色虚线）

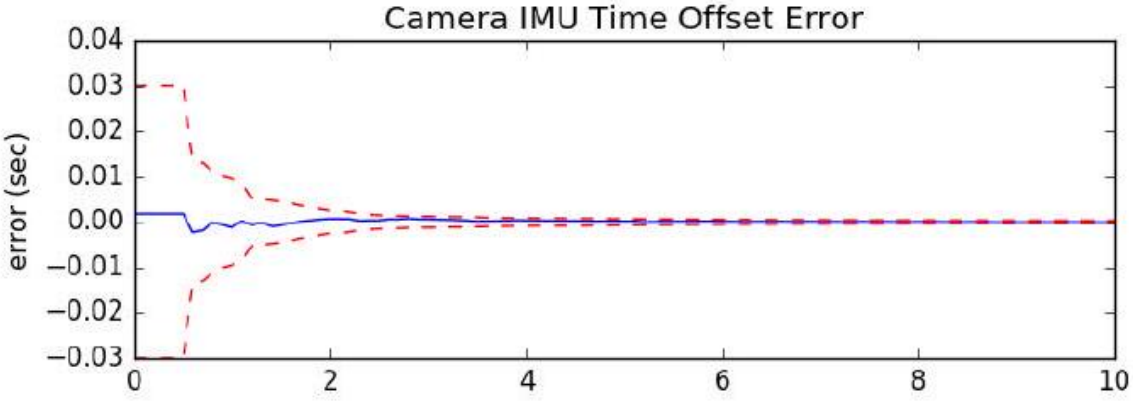

图 4：代表性运行中相机至 IMU 时间偏移误差（蓝色实3σ线）及边界（红色虚线）

图 2 和图 4 中代表性运行的所有校准参数的秒数，显示这些参数从初始较差的猜测值迅速收敛。

B. 现实世界对比

我们基于 EurocMav 数据集[41]中的 Vicon 房间场景，评估了采用与不采用 SLAM 地标点的视觉惯性 FEJ - MSCKF 估计器。该数据集包含 20Hz 立体图像、200Hz ADIS16448 MEMS 惯性测量单元（IMU）测量数据以及优化后的真实轨迹。需要说明的是，由于原始 V1_01_easy 存在方向值错误，我们已重新计算并修正了真实轨迹数据，并在文档网站上向社区公开了校正后的轨迹。所有方法均采用开源仓库中的配置文件运行，每个算法在每个数据集上运行十次以补偿视觉前端固有的随机性。在本次基准测试中，我们评估了

表 I：在真实或不良校准条件下，采用在线校准与非在线校准方式，二十次运行的平均 ATE（平均绝对误差）与 NEES 。

	ATE（度）	ATE（毫摩尔）	Ori. NEES	NEES
真值/校准值	0.212	0.134	2.203	1.880
真值（无校准）	0.200	0.128	2.265	1.909
带口径的坏的	0.218	0.139	2.235	2.007
无校准的坏的	5.432	508.719	9.159	1045.174

采用以下最先进的视觉惯性估计算法：

OKVIS [2]——一种基于关键帧的固定滞后平滑算法，通过优化任意间隔的关键帧姿态，并结合惯性测量因子与环境地标信息。该算法强制采用固定窗口尺寸以确保计算可行性，同时采用选择性边缘化策略以实现问题稀疏性。

VINS-Fusion VIO [3] - 基于原始 VINSMono [42]滑动优化方法的扩展，该方法利用惯性测量单元（IMU）预积分技术，并与次级姿态图优化进行松散耦合。VINS-Fusion 扩展了原始代码库以支持立体相机。

Basalt VIO [4] - 基于立体关键帧的固定滞后平滑器，配备定制化特征跟踪前端，专注于从 VIO 中提取相关信息，以供后续离线视觉惯性测绘使用。

R-VIO[5]——一种基于机器人中心 MSCKF 的算法，该算法在局部坐标系中进行估计，并通过合成步骤更新全局坐标系。重力方向也在滤波器内进行估计。

ROVIO [6] -我们采用 MapLab[43]中的 ROVIO 实现，这是一种基于单目迭代 EKF 的方法，通过对直接图像强度区域进行最小化处理，能够追踪非角点特征（如高梯度线）。

ICE-BA[7]——一种立体增量束调整（BA）方法，可并行优化局部侧窗与全局优化问题。该方法利用其公式结构的稀疏性，并引入了一种相对边际化程序。

S- MSCKF [8] - 原始开源实现

表 II：十次运行的平均绝对轨迹误差（ATE）（单位：度/米），各算法数据。需注意 V2_03 数据集因部分算法无法运行而被排除。绿色表示最优，蓝色次优。

	V1_01_easy	V1_02_medium	V1_03_difficult	V2_01_easy	V2_02_medium	平均
单超声波	0.699 / 0.058	1.675 / 0.076	2.542 / 0.063	0.773 / 0.124	1.538 / 0.074	1.445 / 0.079
单体卵黄蛋白	0.642 / 0.076	1.766 / 0.096	2.391 / 0.344	1.164 / 0.121	1.248 / 0.106	1.442 / 0.148
单视点	0.823 / 0.090	2.082 / 0.146	4.122 / 0.222	0.826 / 0.117	1.704 / 0.197	1.911 / 0.154

单核小体	2.249 / 0.153	1.635 / 0.131	3.253 / 0.158	1.455 / 0.106	1.678 / 0.153	2.054 / 0.140
单重音节	0.994 / 0.094	2.288 / 0.129	1.757 / 0.147	1.735 / 0.144	1.690 / 0.233	1.693 / 0.149
单体血管融合	1.199 / 0.064	3.542 / 0.103	5.934 / 0.202	1.585 / 0.073	2.370 / 0.079	2.926 / 0.104
立体超声定位	0.856 / 0.061	1.813 / 0.047	2.764 / 0.059	1.037 / 0.056	1.292 / 0.047	1.552 / 0.054
立体视网膜血管	0.905 / 0.061	1.767 / 0.056	2.339 / 0.057	1.106 / 0.053	1.151 / 0.048	1.454 / 0.055

基性岩	0.654 / 0.035	2.067 / 0.059	2.017 / 0.085	0.981 / 0.046	0.888 / 0.059	1.321 / 0.057
立体冰巴	0.909 / 0.059	2.574 / 0.120	3.206 / 0.137	1.819 / 0.128	1.212 / 0.116	1.944 / 0.112
立体视标	0.603 / 0.039	1.963 / 0.079	4.117 / 0.122	0.834 / 0.075	1.201 / 0.092	1.744 / 0.081
立体声短信加密	1.108 / 0.086	2.147 / 0.121	3.918 / 0.198	1.181 / 0.083	2.142 / 0.164	2.099 / 0.130
立体视觉融合	1.073 / 0.054	2.695 / 0.089	3.643 / 0.132	2.499 / 0.071	2.006 / 0.074	2.383 / 0.084

表 III：各算法变体在所有数据集上不同区段长度的相对姿态误差（RPE），单位为度/米。需注意，V2_03 数据集因部分算法无法运行而被排除。

	8m	16m	24m	**32m**	40m	48m

单超声波	0.661 / 0.074	0.802 / 0.086	0.979 / 0.097	1.061 / 0.105	1.145 / 0.120	1.289 / 0.122
单体卵黄蛋白	0.826 / 0.094	1.039 / 0.106	1.215 / 0.111	1.283 / 0.132	1.342 / 0.151	1.425 / 0.184
单视点	0.662 / 0.107	0.870 / 0.161	1.031 / 0.190	1.225 / 0.213	1.384 / 0.240	1.603 / 0.251
单核小体	1.136 / 0.095	1.585 / 0.135	1.847 / 0.184	2.078 / 0.226	2.218 / 0.263	2.402 / 0.295
单重音节	0.705 / 0.130	0.902 / 0.160	1.029 / 0.183	1.074 / 0.213	0.991 / 0.227	1.077 / 0.232
单体血管融合	0.940 / 0.070	1.298 / 0.103	1.680 / 0.118	1.822 / 0.146	1.833 / 0.153	1.860 / 0.171
立体超声定位	0.685 / 0.069	0.876 / 0.080	1.064 / 0.087	1.169 / 0.087	1.275 / 0.098	1.488 / 0.105
立体视网膜血管	0.722 / 0.068	0.892 / 0.077	1.089 / 0.087	1.218 / 0.088	1.342 / 0.101	1.489 / 0.106
基性岩	0.538 / 0.063	0.576 / 0.070	0.649 / 0.078	0.715 / 0.086	0.647 / 0.097	0.758 / 0.111
立体冰巴	0.955 / 0.096	1.227 / 0.114	1.415 / 0.120	1.658 / 0.152	1.856 / 0.173	1.803 / 0.180
立体视标	0.611 / 0.066	0.772 / 0.089	0.916 / 0.103	1.089 / 0.119	1.173 / 0.136	1.404 / 0.141
立体声短信加密	1.084 / 0.098	1.462 / 0.136	1.578 / 0.159	1.667 / 0.187	1.901 / 0.200	2.134 / 0.217

立体视觉融合	0.946 / 0.057	1.357 / 0.079	1.721 / 0.097	1.928 / 0.111	1.935 / 0.125	1.805 / 0.132

MSCKF [25]论文采用立体特征追踪技术，重点关注高速运动场景。

需要说明的是，我们仅评估这些代码库的视觉惯性里程（VIO）部分（即不包括 VINS-Fusion[3]的非实时后端姿态图线程输出及 Basalt[4]的视觉惯性地图），因为只需在这些里程法中的任意一种后添加姿态图优化器，即可提升长期精度。

表 II 展示了各方法在各数据集上的平均绝对时间误差（ATE）。显然，在 OpenVINS 中加入 SLAM 地标显著降低了单目系统的漂移，而对立体视觉性能的影响较小；更重要的是，OpenVINS 的性能已能与其他方法相媲美。我们还对比了各方法的相对姿态误差（RPE）。如表 III 所示，我们的单目系统明显优于当前开源代码库，其立体视觉系统性能仅次于 Basalt。虽然我们未对逐帧时间精度进行严格评估，但发现 Basalt 的表现优于所有其他算法，而我们提出的方法受限于 OpenCV[44]的视觉前端实现和 SLAM 特征更新能力。在首个 EurocMav 数据集上，我们实现了实时处理能力。
2.7x/4.3x1.2x/1.9x

分别在单线程执行模式下，于 Xeon® CPU E3-1505M v6 3.00 GHz 处理器Intel(R)上完成单目 SLAM/VIO 与立体 SLAM/VIO 任务。

六、结论与展望

本文中，我们向研究界展示了 OpenVINS（OV）系统这一创新平台。该系统核心包含视觉处理前端、完整视觉惯性模拟器及模块化流形 EKF 。特别值得一提的是，我们实现了基于 FEJ 的 MSCKF（含/不含 SLAM 地标），并验证了该估计器的性能优势。我们通过详尽的文档说明，使研究人员和从业者无需深入估计理论背景即可快速搭建系统。未来计划将系统扩展为基于滑动窗口优化的估计器，充分利用我们的闭式预积分方法[45]。同时，我们正致力于将视觉惯性测绘与感知功能整合至 OpenVINS 系统。

参考文献

[1] 黄刚，《视觉-惯性导航：简明综述》，载于《国际机器人与自动化会议论文集》，加拿大蒙特利尔，2019 年 5 月。

[2] S. Leutenegger、S. Lynen、M. Bosse、R. Siegwart 和 P. Furgale 合著的《基于关键帧的视觉惯性里程计非线性优化方法》，发表于《国际机器人研究杂志》2015 年第 34 卷第 3 期，第 314-334 页。

[3] T. Qin、J. Pan、S. Cao 和 S. Shen，《基于通用优化的多传感器局部里程计估计框架》，CoRR，卷号 abs/1901.03638,2019 年。

[4] V. C. Usenko、N. Demmel、D. Schubert、J. Stuckler 和 D. Cremers，《基于非线性因子恢复的视觉-惯性映射》，CoRR，卷号 abs/1904.06504,2019 年。

[5] 胡志与黄刚，《机器人中心视觉惯性里程计》，《国际机器人研究杂志》，2019 年 4 月（待刊）。

[6] M. Bloesch、M. Burri、S. Omari、M. Hutter 与 R. Siegwart 合著的《基于迭代扩展卡尔曼滤波器的视觉惯性里程计技术——采用直接光度反馈方案》，发表于《国际机器人研究杂志》2017 年第 36 卷第 10 期，页码 1053-1072。

[7] 刘浩、陈明、张刚、鲍浩和鲍勇合著的《Ice-ba：视觉惯性 SLAM 的增量式、一致且高效的束调整方法》，收录于《IEEE 计算机视觉与模式识别会议论文集》2018 年版，第 1974-1982 页。

[8] 孙科、莫塔·K、普弗罗默·B、瓦特森·M、刘·S、穆尔高恩卡尔·Y、泰勒·C·J、库马尔·V 合著的《鲁棒立体视觉惯性里程计在快速自主飞行中的应用》，发表于《IEEE 机器人与自动化快报》2018 年 4 月第 3 卷第 2 期，第 965-972 页。

[9] K. Eckenhoff、P. Geneva、J. Bloecker 和 G. Huang，《多相机视觉惯性导航的在线本征与外在校准》，载于《国际机器人与自动化会议论文集》，加拿大蒙特利尔，2019 年 5 月。

[10] K. Eckenhoff、P. Geneva 和 G. Huang，《具有传感器故障容错能力的多惯性测量单元视觉惯性导航》，载于《国际机器人与自动化会议论文集》，加拿大蒙特利尔，2019 年 5 月。

[11] K. Eckenhoff、Y. Yang、P. Geneva 和 G. Huang，《紧密耦合的视觉-惯性定位与三维刚体目标跟踪》，《IEEE 机器人与自动化快报（RA-L）》，第 4 卷第 2 期，第 1541-1548 页，2019 年。

[12] K. Eckenhoff、P. Geneva、N. Merrill 和 G. Huang，“基于 Schmidt-ekf 的视觉惯性移动目标跟踪，”收录于《IEEE 国际机器人与自动化会议论文集》，法国巴黎，2020 年。

[13]P.Jenève、K.Eckenhoff 和 G.Huang 合著的《基于线性复杂度的视觉惯性导航闭环 EKF 》，收录于 2019 年 5 月加拿大蒙特利尔国际机器人与自动化会议论文集。

[14] P. Geneva、J. Maley 和 G. Huang，“一种高效的三维视觉惯性 SLAM 施密特-扩展卡尔曼滤波算法，”计算机视觉与模式识别会议（CVPR）论文集，加州长滩，2019 年 6 月，（已接受）。

[15] 杨宇、P. 日内瓦、左旭、K. 埃肯霍夫、刘宇和黄刚，《基于点与平面特征的紧密耦合辅助惯性导航》，载于《国际机器人与自动化会议论文集》，加拿大蒙特利尔，2019 年 5 月。

[16] 杨宇、P. 日内瓦、K. 埃肯霍夫与 G. 黄，《基于点与线特征的视觉-惯性导航》，中国澳门，2019 年 11 月（已录用）。

[17] 齐泽、P. 吉内瓦、W. 李、Y. 刘和 G. 黄，《激光雷达-惯性相机里程计：激光雷达-惯性相机里程计（LIC-Fusion）》，中国澳门，2019 年 11 月，（已接受）。

[18] 齐晓、杨鹏、杨阳、叶伟、刘勇和黄刚，《基于激光雷达地图先验约束的视觉惯性定位》，IEEE 机器人与自动化快报（RA-L），2019 年（待刊）。

[19] 杨 Y、日内瓦 P、埃肯霍夫 K 与黄 G 合著的《基于在线时空校准的辅助 INS 退化运动分析》，发表于《IEEE 机器人与自动化快报（RA-L）》2019 年第 4 卷第 2 期，页码 2070-2077。

[20] 黄 G、穆里基斯 A.I.与鲁梅利奥蒂斯 S.I.合著的《基于扩展卡尔曼滤波的 SLAM 一致性分析与改进》，收录于 2008 年 5 月 19-23 日于美国加州帕萨迪纳市举行的 IEEE 国际机器人与自动化会议论文集，第 473-479 页。

[21]，《用于提升 SLAM 一致性的初步估计雅可比 EKF 》，收录于《第十一届国际实验机器人研讨会论文集》，希腊雅典，2008 年 7 月 14-17 日。

[22]，《基于可观测性的规则设计一致 EKF SLAM 估计器》，《国际机器人研究杂志》，第 29 卷第 5 期，第 502-528 页，2010 年 4 月。

[23] 李明与 A·I·穆里基斯合著的《相机惯性测量单元系统的在线时间标定：理论与算法》，载于《国际机器人研究杂志》2014 年 6 月第 33 卷第 7 期，第 947-964 页。

[24] 李明、余浩、郑晓和 A·I·穆里基斯合著的《视觉辅助惯性导航中的高保真传感器建模与自校准》，收录于 2014 年 5 月 IEEE 国际机器人与自动化会议（ICRA），第 409-416 页。

[25] A. I. Mourikis 与 S. I. Roumeliotis 合著的《视觉辅助惯性导航的多状态约束卡尔曼滤波器》，收录于《IEEE 国际机器人与自动化会议论文集》（2007 年 4 月 10-14 日，意大利罗马），第 3565-3572 页。

[26] N. Trawny 与 S. I. Roumeliotis 合著的《三维姿态估计的间接卡尔曼滤波器》，明尼苏达大学计算机科学与工程系技术报告，2005 年 3 月。

[27] C. 赫茨伯格、R. 瓦格纳、U. 弗雷斯与 L. 施罗德合著的《通过流形封装将通用传感器融合算法与声音状态表征整合》，载于《信息融合》2013 年第 14 卷第 1 期，第 57-77 页。

[28] 吴科、张涛、苏丹、黄松和迪萨纳亚克·G. 合著的《一种用于提升一致性的不变量-EKF VINS 算法》，收录于《IEEE/RSJ 国际智能机器人与系统会议论文集》2017 年 9 月刊，第 1578-1585 页。

[29] J. Civera、A. Davison 和 J. Montiel，《单目 SLAM 的逆深度参数化》，《IEEE 机器人学汇刊》，第 24 卷第 5 期，第 932-945 页，2008 年 10 月。

[30] M. K. Paul、吴凯、J. A. Hesch、E. D. Nerurkar 和 S. I. Roumeliotis 合著的《紧密耦合单目、双眼及立体视觉神经刺激系统（VINS）的比较分析》，收录于《IEEE 国际机器人与自动化会议论文集》（2017 年 7 月，新加坡），第 165-172 页。

[31] A. B. Chatfield，《高精度惯性导航基础》， AIAA ，1997 年。

[32] F. Dellaert，《因子图与 gtsam：实践入门》，佐治亚理工学院技术报告，2012 年。

[33] 李明，《资源受限系统中的视觉-惯性里程计》，博士论文，加州大学河滨分校，2014 年。

[34] 杨宇、马雷和黄刚合著的《基于零空间的边缘化：分析与算法》，收录于 2017 年 9 月 24 日至 28 日在加拿大温哥华举行的 IEEE/RSJ 国际智能机器人与系统会议论文集，第 6749-6755 页。

[35] S. I. Roumeliotis 与 J. W. Burdick 合著的《随机克隆：处理相对状态测量的通用框架》，收录于《IEEE 国际机器人与自动化会议论文集》（2002 年 5 月 11-15 日，华盛顿特区），第 1788-1795 页。

[36] D. Van Heesch，“Doxygen: 源代码文档生成工具，” 网址: http://www.doxygen.org，2008.

[37] V. Vondrus，“m.css: 一个简洁实用、不依赖 JavaScript 的 CSS 框架及 Pelican 主题，专为内容导向型网站设计，” URL: https://mcss. mosra.cz/，2018.

[38] A. Patron-Perez、S. Lovegrove 和 G. Sibley，《基于样条的轨迹表示方法在传感器融合与滚动快门相机中的应用》，《国际计算机视觉杂志》，第 113 卷第 3 期，第 208-219 页，2015 年。

[39] E. Mueggler、G. Gallego、H. Rebecq 和 D. Scaramuzza，《事件相机的连续时间视觉惯性里程计》，《IEEE 机器人学汇刊》，第 1-16 页，2018 年。

[40] 李明与 A·I·穆里基斯合著的《基于优化的视觉辅助惯性导航估计器设计》，载于《机器人学：科学与系统》（2013 年 6 月，德国柏林），第 241-248 页。

[41] M. Burri、J. Nikolic、P. Gohl、T. Schneider、J. Rehder、S. Omari、M. W. Achtelik 与 R. Siegwart 合著的《欧洲微型飞行器数据集》，载于《国际机器人研究杂志》2016 年第 35 卷第 10 期，第 1157-1163 页。

[42] T. Qin、P. Li 和 S. Shen，《VINS-Mono：一种稳健且多功能的单目视觉-惯性状态估计器》，《IEEE 机器人学汇刊》，第 34 卷第 4 期，第 1004-1020 页，2018 年。

[43] T. 施耐德、M. 迪姆奇克、M. 费尔、K. 埃格、S. 林恩、I. 吉利琴斯基和 R. 西格瓦特合著的《Maplab：视觉惯性测绘与定位研究的开放框架》，发表于《IEEE 机器人与自动化快报》2018 年 7 月第 3 卷第 3 期，第 1418-1425 页。

[44] OpenCV 开发者团队，《开源计算机视觉（OpenCV）库》，可获取地址：http://opencv.org。

[45] K. Eckenhoff、P. Geneva 和 G. Huang，《基于图的视觉惯性导航的闭式预积分方法》，《国际机器人研究杂志》，第 38 卷，第 5 期，第 563-586 页，2019 年。

〈文献翻译二：原文〉

A General Optimization-based Framework for Local Odometry Estimation with Multiple Sensors[2]

Tong Qin, Jie Pan, Shaozu Cao, and Shaojie Shen

Abstract

Nowadays, more and more sensors are equipped on robots to increase robustness and autonomous ability. We have seen various sensor suites equipped on different platforms, such as stereo cameras on ground vehicles, a monocular camera with an IMU (Inertial Measurement Unit) on mobile phones, and stereo cameras with an IMU on aerial robots. Although many algorithms for state estimation have been proposed in the past, they are usually applied to a single sensor or a specific sensor suite. Few of them can be employed with multiple sensor choices. In this paper, we proposed a general optimization-based framework for odometry estimation, which supports multiple sensor sets. Every sensor is treated as a general factor in our framework. Factors which share common state variables are summed together to build the optimization problem. We further demonstrate the generality with visual and inertial sensors, which form three sensor suites (stereo cameras, a monocular camera with an IMU, and stereo cameras with an IMU). We validate the performance of our system on public datasets and through real-world experiments with multiple sensors. Results are compared against other state-of-the-art algorithms. We highlight that our system is a general framework, which can easily fuse various sensors in a pose graph optimization. Our implementations are open source [1].

I. Introduction

[2] Qin T, Pan J, Cao S, et al. A General Optimization-based Framework for Local Odometry Estimation with Multiple Sensors[R]. arXiv preprint arXiv:1901.03638, 2019.

Real-time 6-DoF (Degrees of Freedom) state estimation is a fundamental technology for robotics. Accurate state estimation plays an important role in various intelligent applications, such as robot exploration, autonomous driving, VR (Virtual Reality) and AR (Augmented Reality). The most common sensors we use in these applications are cameras. A large number of impressive vision-based algorithms for pose estimation has been proposed over the last decades, such as [1]-[5]. Besides cameras, the IMU is another popular option for state estimation. The IMU can measure acceleration and angular velocity at a high frequency, which is necessary for low-latency pose feedback in real-time applications. Hence, there are numerous research works fusing vision and IMU together, such as [6]-[12]. Another popular sensor used in state estimation is LiDAR. LiDAR-based approaches [13] achieve accurate pose estimation in a confined local environment. Although a lot of algorithms have been proposed in the past, they are usually applied to a single input sensor or a specific sensor suite.

Recently, we have seen platforms equipped with various sensor sets, such as stereo cameras on ground vehicles, a monocular camera with an IMU on mobile phones, stereo

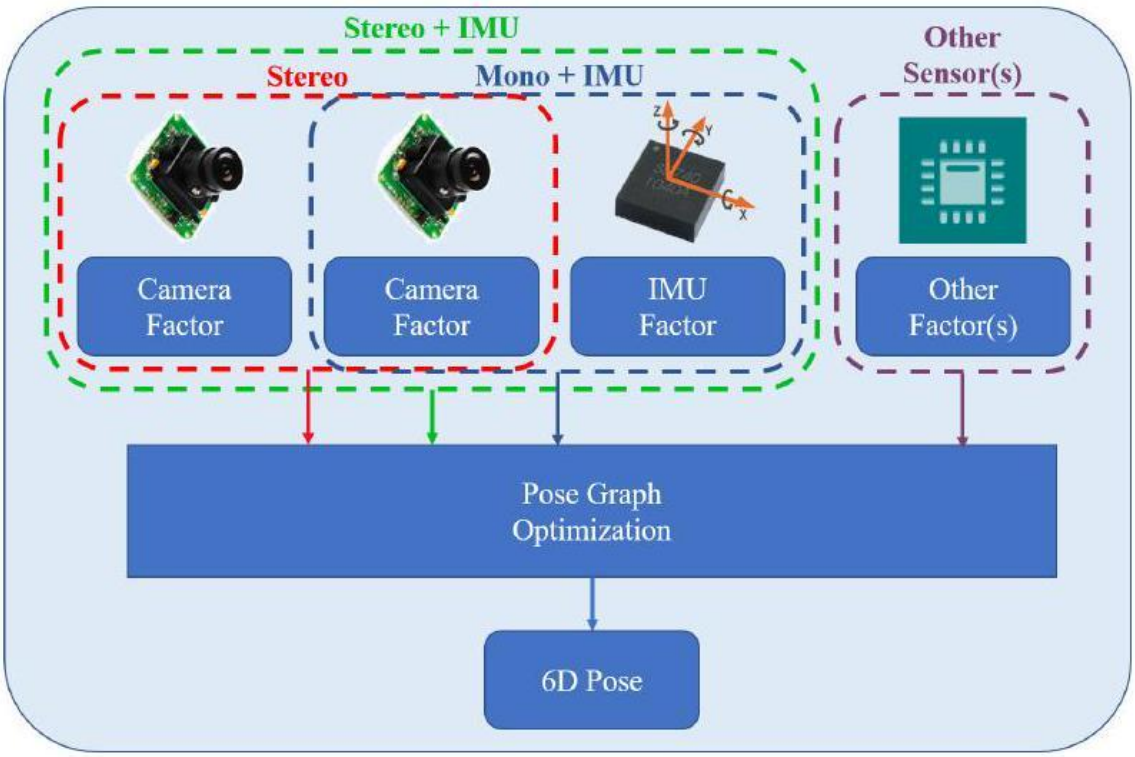

Fig. 1. An illustration of the proposed framework for state estimation, which supports multiple sensor choices, such as stereo cameras, a monocular camera with an IMU, and stereo cameras

with an IMU. Each sensor is treated as a general factor. Factors which share common state variables are summed together to build the optimization problem.
cameras with an IMU on aerial robots. However, as most traditional algorithms were designed for a single sensor or a specific sensor set, they cannot be ported to different platforms. Even for one platform, we need to choose different sensor combinations in different scenarios. Therefore, a general algorithm which supports different sensor suites is required. Another practical requirement is that in case of sensor failure, an inactive sensor should be removed and an alternative sensor should be added into the system quickly. Hence, a general algorithm which is compatible with multiple sensors is in need.

In this paper, we propose a general optimization-based framework for pose estimation, which supports multiple sensor combinations. We further demonstrate it with visual and inertial sensors, which form three sensor suites (stereo cameras, a monocular camera with an IMU, and stereo cameras with an IMU). We can easily switch between different sensor combinations. We highlight the contribution of this paper as follows:

- a general optimization-based framework for state estimation, which supports multiple sensors.
- a detailed demonstration of state estimation with visual and inertial sensors, which form different sensor suites (stereo cameras, a monocular camera + an IMU, and stereo cameras + an IMU).
- an evaluation of the proposed system on both public datasets and real experiments.
- open-source code for the community.

II. Related Work

State estimation has been a popular research topic over the last decades. A large number of algorithms focus on accurate 6-DoF pose estimation. We have seen many impressive approaches that work with one kind of sensor, such as visual-based methods [1]-[5], LiDAR-based methods [13], RGB-D based methods [14]. and event-based methods [15]. Approaches work with a

monocular camera is hard to achieve 6-DoF pose estimation, since absolute scale cannot be recovered from a single camera. To increase the observability and robustness, multiple sensors which have complementary properties are fused together.

There are two trends of approaches for multi-sensor fusion. One is filter-based methods, the other is optimizationbased methods. Filter-based methods are usually achieved by EKF (Extended Kalman Filter). Visual and inertial measurements are usually filtered together for 6-DoF state estimation. A high-rate inertial sensor is used for state propagation and visual measurements are used for the update in [9, 16]. MSCKF [6, 7] was a popular EKF-based VIO (Visual Inertial Odometry), which maintained several camera poses and leveraged multiple camera views to form the multi-constraint update. Filter-based methods usually linearize states earlier and suffer from error induced by inaccurate linear points. To overcome the inconsistency caused by linearized error, observability constrained EKF [17] was proposed to improve accuracy and consistency. An UKF (Unscented Kalman Filter) algorithm was proposed in [18], where visual, Li DAR and GPS measurements were fused together. UKF is an externsion of EKF without analytic Jacobians. Filterbased methods are sensitive to time synchronization. Any late-coming measurements will cause trouble since states cannot be propagated back in filter procedure. Hence, special ordering mechanism is required to make sure that all measurements from multiple sensors are in order.

Optimization-based methods maintain a lot of measurements and optimize multiple variables at once, which is also known as Bundle Adjustment (BA). Compared with filterbased method, optimization-based method have advantage in time synchronization. Because the big bundle serves as a nature buffer, it can easily handle the case when measurments from multiple sensors come in disorder. Optimization-based algorithms also outperform the filter-based algorithms in term of accuracy at the cost of computational complexity. Early optimization solvers, such as G2O [19], leveraged the Gauss-Newton and Levenberg-Marquardt approaches to solve the problem. Although the sparse structure was employed in optimization solvers, the complexity grown quadratically with the number of states and measurements. In order to achieve real-time performance, some algorithms have explored incremental solvers, while others bounded the

size of the pose graph. iSAM2 [20] was an efficient incremental solver, which reused the previous optimization result to reduce computation when new measurements came. The optimization iteration only updated a small part of states

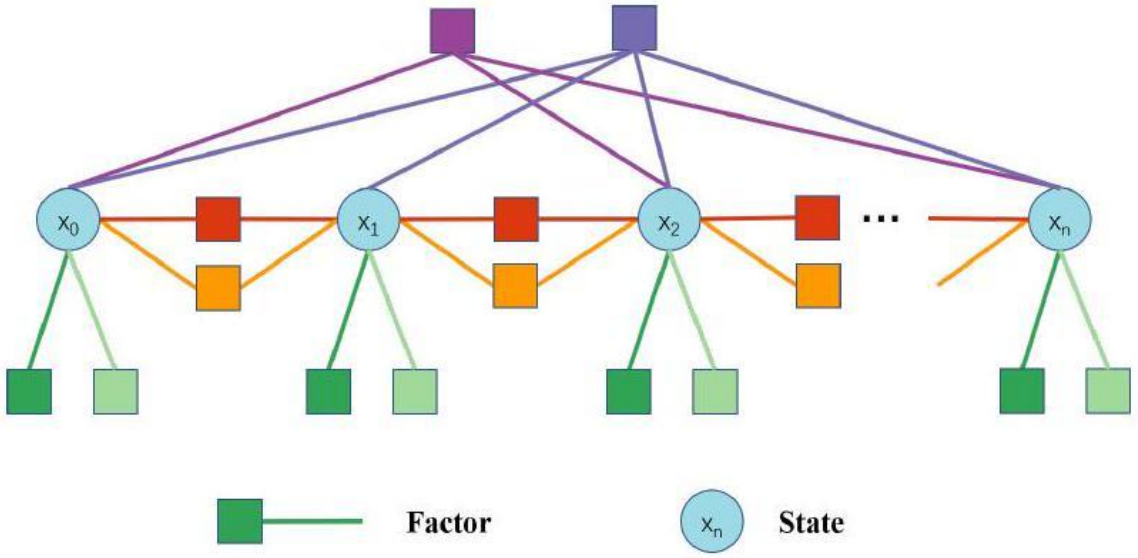

Fig. 2. A graphic illustration of the pose graph. Each node represents states (position, orientation, velocity and so on) at one moment. Each edge represents a factor, which is derived by one measurement. Edges constrain one state, two states or multiple states.

instead of the whole pose graph. Afterward, an accelerated solver was proposed in [21], which improved efficiency by reconstructing dense structure into sparse blocks. Methods, that keep a fixed sized of pose graph, are called slidingwindow approaches. Impressive optimization-based VIO approaches, such as [8,10,12], optimized variables over a bounded-size sliding window. The previous states were marginalized into a prior factor without loss of information in [8, 12]. In this paper, we adopt a sliding-window optimization-based framework for state estimation.

III. System Overview

The structure of proposed framework is shown in Fig. 1. Multiple kinds of sensors can be freely combined. The mesurement of each sensor is treated as a general factor. Factors and their related states form the pose graph. An illustration of pose graph is shown in Fig. 2. Each node represents states (position, orientation, velocity and so on) at one moment. Each edge represents a factor, which is derived by one measurement. Factors constrain one state, two states or multiple states. For IMU factor, it constrains two consecutive states by continuous motion restriction. For a visual landmark, its factor constrains multiple states since it is observed on

multiple frames. Once the graph is built, optimizing it equals to finding the configuration of nodes that match all edges as much as possible.

In this paper, we specifically demonstrate the system with visual and inertial sensors. Visual and inertial sensors can form three combinations for 6-DoF state estimation, which are stereo cameras, a monocular camera with an IMU, and stereo cameras with an IMU. A graphic illustration of the proposed framework with visual and inertial sensors is shown in Fig. 3. Several camera poses, IMU measurements and visual measurements exist in the pose graph. The IMU and one of cameras are optional.

IV. Methodology

A. Problem Definition

1. States: Main states that we need to estimate includes 3D position and orientation of robot's center. In addition, we

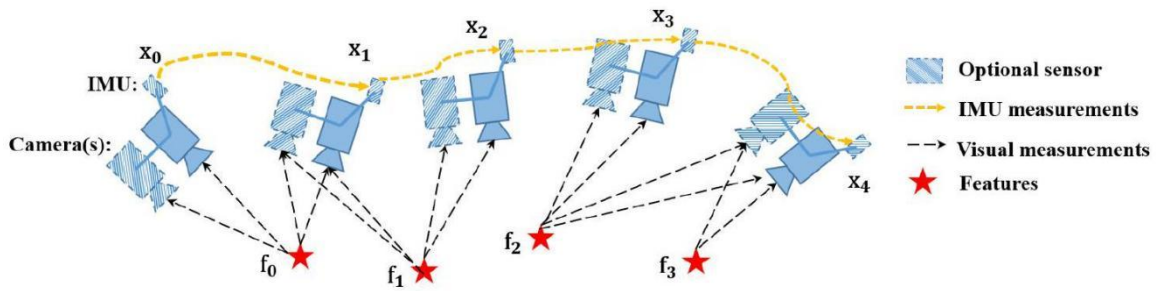

Fig. 3. A graphic illustration of the proposed framework with visual and inertial sensors. The IMU and one of cameras are optional. Therefore, it forms three types (stereo cameras, a monocular camera with an IMU, and stereo cameras with an IMU). Several camera poses, IMU measurements and visual measurements exist in the pose graph.

have other optional states, which are related to sensors. For cameras, depths or 3D locations of visual landmarks need to be estimated. For IMU, it produces another motion variable, velocity, Also, time-variant acceleration bias and gyroscope bias of the IMU are needed to be estimated. Hence, for visual and inertial sensors, whole states we need to estimate are defined as follows:

$$\mathcal{X} = [\mathbf{p}_0, \mathbf{R}_0, \mathbf{p}_1, \mathbf{R}_1, \dots, \mathbf{p}_n, \mathbf{R}_n, \mathbf{x}_{cam}, \mathbf{x}_{imu}]$$
$$\mathbf{x}_{cam} = [\lambda_0, \lambda_1, \dots, \lambda_l] \quad (1)$$
$$\mathbf{x}_{imu} = [\mathbf{v}_0, \mathbf{b}_{a_0}, \mathbf{b}_{g_0}, \mathbf{v}_1, \mathbf{b}_{a_1}, \mathbf{b}_{g_1}, \dots, \mathbf{v}_n, \mathbf{b}_{a_n}, \mathbf{b}_{g_n}], \quad (1)$$

where $\mathbf{p}$ and $\mathbf{R}$ are basic system states, which correspond to position and orientation of body expressed in world frame. $\mathbf{x}_{\text{cam}}$ is camera-ralated state, which includes depth λ of each feature observed in the first frame. $\mathbf{x}_{imu}$ is IMU-ralated variable, which is composed of velocity $\mathbf{v}$, acceleration bias $\mathbf{b}_a$ and gyroscope bias $\mathbf{b}_g$. $\mathbf{x}_{imu}$ can be omitted if we only use stereo camera without an IMU. The translation from sensors' center to body's center are assumed to be known, which are calibrated offline. In order to simplify the notation, we denote the IMU as body's center (If the IMU is not used, we denote left camera as body's center).

2) Cost Function: The nature of state estimation is an MLE (Maximum Likelihood Estimation) problem. The MLE consists of the joint probability distribution of robot poses over a period of time. Under the assumption that all measurements are independent, the problem is typically derived as,

$$\mathcal{X}^* = \arg\max_{\mathcal{X}} \prod_{t=0}^{n} \prod_{k\in\mathbf{S}} p\left(\mathbf{z}_t^k \mid \mathcal{X}\right), \tag{2}$$

where $\mathbf{S}$ is the set of measurements, which come from cameras, IMU and other sensors. We assume the uncertainty of measurements is Gaussian distributed, $p\left(\mathbf{z}_t^k \mid \mathcal{X}\right) \sim \mathcal{N}\left(\overline{\mathbf{z}}_t^k, \Omega_t^k\right)$. Therefore, the negative log-likelihood of abovementioned equation is written as,

$$\begin{aligned}\mathcal{X}^* &= \arg\max_{\mathcal{X}} \prod_{t=0}^{n} \prod_{k\in\mathbf{S}} \exp\left(-\frac{1}{2}\left\|\mathbf{z}_t^k - h_t^k(\mathcal{X})\right\|_{\mathbf{\Omega}_t^k}^2\right) \\ &= \arg\min_{\mathcal{X}} \sum_{t=0}^{n} \sum_{k\in\mathbf{S}} \left\|\mathbf{z}_t^k - h_t^k(\mathcal{X})\right\|_{\mathbf{\Omega}_t^k}^2. \end{aligned} \tag{3}$$

The Mahalanobis norm is defined as $\|\mathbf{r}\|_{\mathbf{\Omega}}^2 = \mathbf{r}^T\mathbf{\Omega}^{-1}\mathbf{r}$. $h(\cdot)$ is the sensor model, which is detailed in the following section. Then the state estimation is converted to a nonlinear least squares problem, which is also known as Bundle Adjustment (BA).

B. Sensor Factors

1. Camera Factor: The framework supports both monocular and stereo cameras. The intrinsic parameters of every camera and the extrinsic transformation between cameras are supposed to be known, which can be easily calibrated offline. For each camera

frame, corner features [22] are detected. These features are tracked in previous frame by KLT tracker [23]. For the stereo setting, the tracker also matches features between the left image and right image. According to the feature associations, we construct the camera factor with per feature in each frame. The camera factor is the reprojection process, which projects a feature from its first observation into following frames.

Considering the feature l that is first observed in the image i, the residual for the observation in the following image t is defined as:

$$\begin{aligned}\mathbf{z}_t^l - h_t^l(\mathcal{X}) &= \mathbf{z}_t^l - h_t^l(\mathbf{R}_i, \mathbf{p}_i, \mathbf{R}_t, \mathbf{p}_t, \lambda_l) \\ &= \begin{bmatrix} u_t^l \\ v_t^l \end{bmatrix} - \pi_c \left(\mathbf{T}_c^{b-1} \mathbf{T}_t^{-1} \mathbf{T}_i \mathbf{T}_c^b \pi_c^{-1} \left(\lambda_l, \begin{bmatrix} u_i^l \\ v_i^l \end{bmatrix} \right) \right), \quad (4)\end{aligned}$$

where $[u_i^l, v_i^l]$ is the first observation of the l feature that appears in the i image. [u_t^l, v_t^l] is the observation of the same feature in the t image. π_c and π_c^{-1} are the projection and back-projection functions which depend on camera model (pinhole, omnidirectional or other models). $\mathbf{T}$ is the 4×4 homogeneous transformation, which is $\begin{bmatrix} \mathbf{R} & \mathbf{p} \\ \mathbf{0} & 1 \end{bmatrix}$. We omit some homogeneous terms for concise expression. $\mathbf{T}_b^c$ is the extrinsic transformation from body center to camera center, which is calibrated offline. The covariance matrix $\mathbf{\Omega}_t^l$ of reprojection error is a constant value in pixel coordinate, which comes from the camera's intrinsic calibration results.

This factor is universal for both left camera and right camera. We can project a feature from the left image to the left image in temporal space, also we can project a feature from the left image to the right image in spatial space.

For different cameras, a different extrinsic transformation $\mathbf{T}_b^c$ should be used.

2) IMU Factor: We use the well-known IMU preintegration algorithm [11, 12] to construct the IMU factor. We assume that the additive noise in acceleration and gyroscope measurements are Gaussian white noise. The time-varying acceleration and gyroscope bias are modeled as a random walk process, whose derivative is Gaussian white noise. Since the IMU acquires data at a higher frequency than other sensors, there are usually multiple IMU measurements existing be-

tween two frames. Therefore, we pre-integrate IMU measurements on the manifold with covariance propagation. The detailed preintegration can be found at [12]. Within two time instants, $t-1$ and t, the preintegration produces relative position $\boldsymbol{\alpha}_t^{t-1}$, velocity $\boldsymbol{\beta}_t^{t-1}$ and rotation $\boldsymbol{\gamma}_t^{t-1}$. Also, the preintegration propagates the covariance of relative position, velocity, and rotation, as well as the covariance of bias. The IMU residual can be defined as:

$$\mathbf{z}_t^{imu} - h_t^{imu}(\mathcal{X}) = \begin{bmatrix} \boldsymbol{\alpha}_t^{t-1} \\ \boldsymbol{\beta}_t^{t-1} \\ \boldsymbol{\gamma}_t^{t-1} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} \ominus \begin{bmatrix} \mathbf{R}_{t-1}{}^{-1}\left(\mathbf{p}_t - \mathbf{p}_{t-1} + \frac{1}{2}\mathbf{g}dt^2 - \mathbf{v}_{t-1}dt\right) \\ \mathbf{R}_{t-1}{}^{-1}(\mathbf{v}_t - \mathbf{v}_{t-1} + \mathbf{g}dt) \\ \mathbf{R}_{t-1}{}^{-1}\mathbf{R}_t \\ \mathbf{b}_{a_t} - \mathbf{b}_{a_{t-1}} \\ \mathbf{b}_{g_t} - \mathbf{b}_{g_{t-1}} \end{bmatrix} \quad (5)$$

where $\ominus$ is the minus operation on manifold, which is specially used for non-linear rotation. dt is the time interval between two time instants. g is the known gravity vector, whose norm is around 9.81. Every two adjacent frames construct one IMU factor in the cost function.

3) Other Factors: Though we only specify camera and IMU factors, our system is not limited to these two sensors. Other sensors, such as wheel speedometer, LiDAR and Radar, can be added into our system without much effort. The key is to model these measurements as general residual factors and add these residual factors into cost function.

C. Optimization

In traditional, the nonlinear least square problem of eq. 3 is solved by Newton-Gaussian or Levenberg-Marquardt approaches. The cost function is linearized with respect to an initial guess of states, $\hat{\mathcal{X}}$. Then, the cost function is equals to:

$$\arg\min_{\delta\mathcal{X}} \sum_{t=0}^{n} \sum_{k\in\mathbf{S}} \left\| \mathbf{e}_t^k + \mathbf{J}_t^k \delta\mathcal{X} \right\|_{\boldsymbol{\Omega}_t^k}^2, \quad (6)$$

where $\mathbf{J}$ is the Jacobian matrix of each factor with respect to current states $\hat{\mathcal{X}}$. After linearization approximation, this cost function has closed-form solution of $\delta\mathcal{X}$. We take NewtonGaussian as example, the solution is derived as follows,

$$\underbrace{\sum \sum \mathbf{J}_t^{k^T} \mathbf{\Omega}_t^{k^{-1}} \mathbf{J}_t^k}_{\mathbf{H}} \delta\mathcal{X} = \underbrace{-\sum \sum \mathbf{J}_t^{k^T} \mathbf{\Omega}_t^{k^{-1}} \mathbf{e}_t^k}_{\mathbf{b}}. \tag{7}$$

Finally, current state $\hat{\mathcal{X}}$ is updated with $\hat{\mathcal{X}} \oplus \delta\mathcal{X}$, where $\oplus$ is the plus operation on manifold for rotation. This procedure iterates several times until convergence. We adopt Ceres solver [24] to solve this problem, which utilizes advanced mathematical tools to get stable and optimal results efficiently.

D. Marginalization

Since the number of states increases along with time, the computational complexity will increase quadratically accordingly. In order to bound the computational complexity, marginalization is incorporated without loss of useful information. Marginalization procedure converts previous measurements into a prior term, which reserves past information. The set of states to be marginalized out is denoted as $\mathcal{X}_m$, and the set of remaining states is denoted as $\mathcal{X}_r$. By summing all marginalized factors (eq.7), we get a new $\mathbf{H}$ and $\mathbf{b}$. After rearrange states' order, we get the following relationship:

$$\begin{bmatrix} \mathbf{H}_{mm} & \mathbf{H}_{mr} \\ \mathbf{H}_{rm} & \mathbf{H}_{rr} \end{bmatrix} \begin{bmatrix} \delta\mathcal{X}_m \\ \delta\mathcal{X}_r \end{bmatrix} = \begin{bmatrix} \mathbf{b}_m \\ \mathbf{b}_r \end{bmatrix} \tag{8}$$

The marginalization is carried out using the Schur complement [25] as follows:

$$\underbrace{(\mathbf{H}_{rr} - \mathbf{H}_{rm}\mathbf{H}_{mm}{}^{-1}\mathbf{H}_{mr})}_{\mathbf{H}_p} \delta\mathcal{X}_r = \underbrace{\mathbf{b}_r - \mathbf{H}_{rm}\mathbf{H}_{mm}{}^{-1}\mathbf{b}_m}_{\mathbf{b}_p} \tag{9}$$

We get a new prior $\mathbf{H}_p, \mathbf{b}_p$ for the remaining states. The information about marginalized states is converted into prior term without any loss. To be specific, we keep ten spacial camera frames in our system. When a new keyframe comes, we marginalize out the visual and inertial factors, which are related with states of the first frame.

After we get the prior information about current states, with Bayes' rule, we could calculate the posterior as a product of likelihood and prior: $p(\mathcal{X} \mid \mathbf{z}) \propto p(\mathbf{z} \mid \mathcal{X})p(\mathcal{X})$. The state estimation then becomes a MAP (Maximum A Posteriori) problem. Denote that we keep states from instant m to instant n in the sliding window. The states before m are marginalized out and converted to a prior term. Therefore, the MAP problem is written as:

$$\mathcal{X}^*_{m:n} = \arg\max_{\mathcal{X}_{m:n}} \prod_{t=m}^{n} \prod_{k\in\mathbf{S}} p\left(\mathbf{z}_t^k \mid \mathcal{X}_{m:n}\right) p(\mathcal{X}_{m:n})$$

$$= \arg\min_{\mathcal{X}_{m:n}} \sum_{t=m}^{n} \sum_{k\in\mathbf{S}} \left\| \mathbf{z}_t^k - h_t^k(\mathcal{X}_{m:n}) \right\|^2_{\Omega_t^k} \quad (10)$$

$$+\left(\mathbf{H}_p \delta\mathcal{X}_{m:n} - \mathbf{b}_p\right) \quad (10)$$

Compared with eq.3, the above-mentioned equation only adds a prior term. It is solved as same as eq. 3 by Ceres solver [24].

E. Discussion

The proposed system is a general framework. Various sensors can be easily added into our system, as long as it can be derived as a general residual factor. Since our system is not specially designed for a certain sensor, it is capable to handle sensor failure case. When sensor failure occurs, we just remove factors of the inactive sensor and add new factors from other alternative sensors.

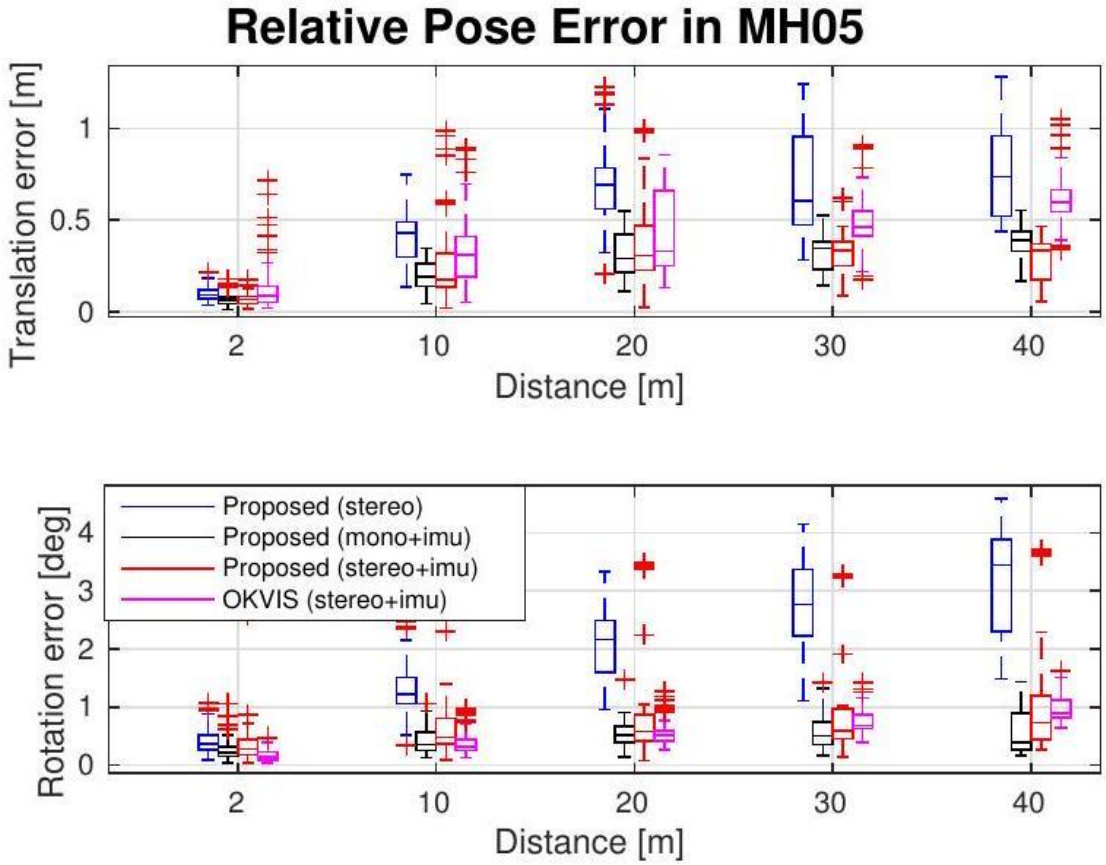

Fig. 4. Relative pose error [26] in MH_05_difficult. Two plots are relative errors in translation and rotation respectively.

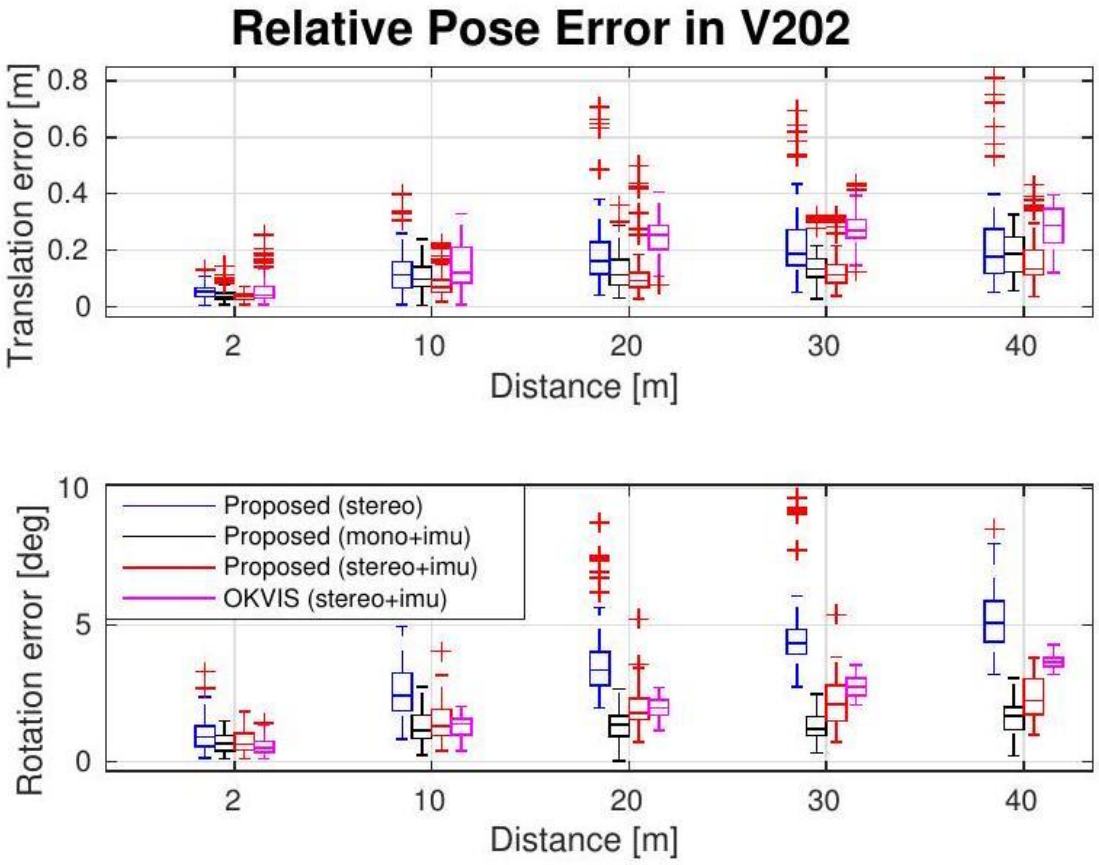

Fig. 5. Relative pose error [26] in V2_02_medium. Two plots are relative errors in translation and rotation respectively.

V. Experimental Results

We evaluate the proposed system with visual and inertial sensors both on datasets and with real-world experiments. In the first experiment, we compare the proposed algorithm with another state-of-the-art algorithm on public datasets. We then test our system in the large-scale outdoor environment. The numerical analysis is generated to show the accuracy of our system in detail.

A. Datasets

We evaluate our proposed system using the EuRoC MAV Visual-Inertial Datasets [27]. This datasets are collected onboard a micro aerial vehicle, which contain stereo images (Aptina MT9V034 global shutter, 752×480 monochrome, 20

TABLE I

RMSE[m] in EuRoC dataset.

Sequence	Length	Proposed RMSE			OKVIS RMSE
		stereo	mono+imu	stereo+imu	
MH_01	79.84	0.54	0.18	0.24	0.16
MH_02	72.75	0.46	0.09	0.18	0.22
MH_03	130.58	0.33	0.17	0.23	0.24
MH_04	91.55	0.78	0.21	0.39	0.34
MH_05	97.32	0.50	0.25	0.19	0.47
V1_01	58.51	0.55	0.06	0.10	0.09
V1_02	75.72	0.23	0.09	0.10	0.20
V1_03	78.77	x	0.18	0.11	0.24
V2_01	36.34	0.23	0.06	0.12	0.13
V2_02	83.01	0.20	0.11	0.10	0.16
V2_03	85.23	x	0.26	0.27	0.29

FPS), synchronized IMU measurements (ADIS16448, 200 Hz), Also, the ground truth states are provided by VICON and Leica MS50. We run datasets with three different combinations of sensors, which are stereo cameras, a monocular camera with an IMU, stereo cameras with an IMU separately.

In this experiment, we compare our results with OKVIS [8], a state-of-the-art VIO that works with stereo cameras and an IMU. OKVIS is another optimization-based sliding-window algorithm. OKVIS is specially designed for visual-inertial sensors, while our system is a more general framework, which supports multiple sensors combinations. We tested the proposed framework and OKVIS with all sequences in EuRoC datasets. We evaluated accuracy by RPE (Relative Pose Errors) and ATE (Absolute Trajectory

Errors). The RPE is calculated by tools proposed in [26]. The RPE (Relative Pose Errors) plot of two sequences, MH_05_difficult and V2_02_medium, are shown in Fig. 4 and Fig. 5 respectively.

The RMSE (Root Mean Square Errors) of ATE for all sequences in EuRoC datasets is shown in Table. I. Estimated trajectories are aligned with the ground truth by Horn's method [28]. The stereo-only case fails in V1_03_difficult and V2_03_difficult sequences, where the movement is too aggressive for visual tracking to survive. Methods which involves the IMU work successfully in all sequences. It is a good case to show that the IMU can dramatically improve motion tracking performance by bridging the gap when visual tracks fail due to illumination change, textureless area, or motion blur.

From the relative pose error and absolute trajectory error, we can see that the stereo-only method performed worst in most sequences. Position and rotation drift obviously grown along with distance in stereo-only case. In other words, the IMU significantly benefited vision in states estimation. Since the IMU measures gravity vector, it can effectively suppress drifts in roll and pitch angles. Stereo cameras with an IMU didn't always perform best, because it requires more accurate calibration than the case of a monocular camera with an IMU. Inaccurate intrinsic and extrinsic calibration will introduce more noise into the system. In general, multiple sensor fusion increase the robustness of the system. Our results outperforms OKVIS in most sequences.

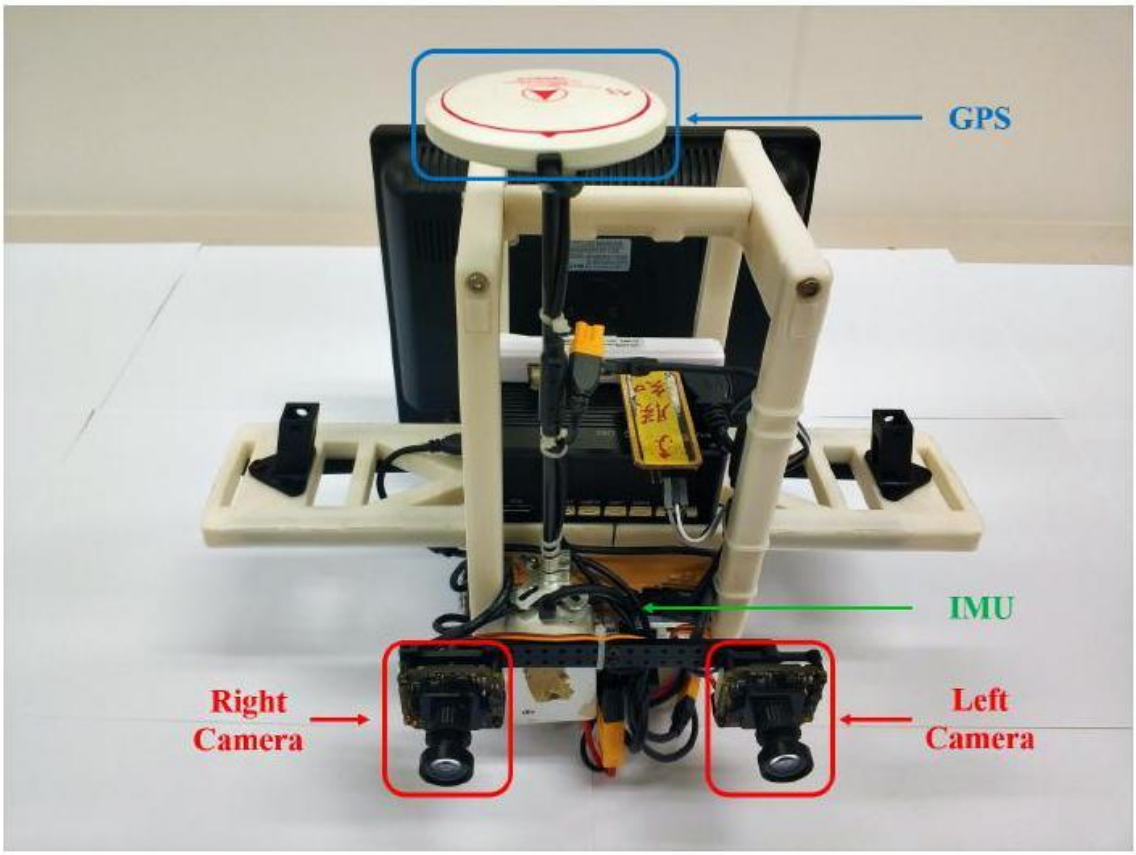

Fig. 6. The self-developed sensor suite used in the outdoor environment. It contains stereo cameras (mvBlueFOX-MLC200w, 20 Hz) and DJI A3 controller, which include inbuilt IMU (200 Hz) and GPS receiver.

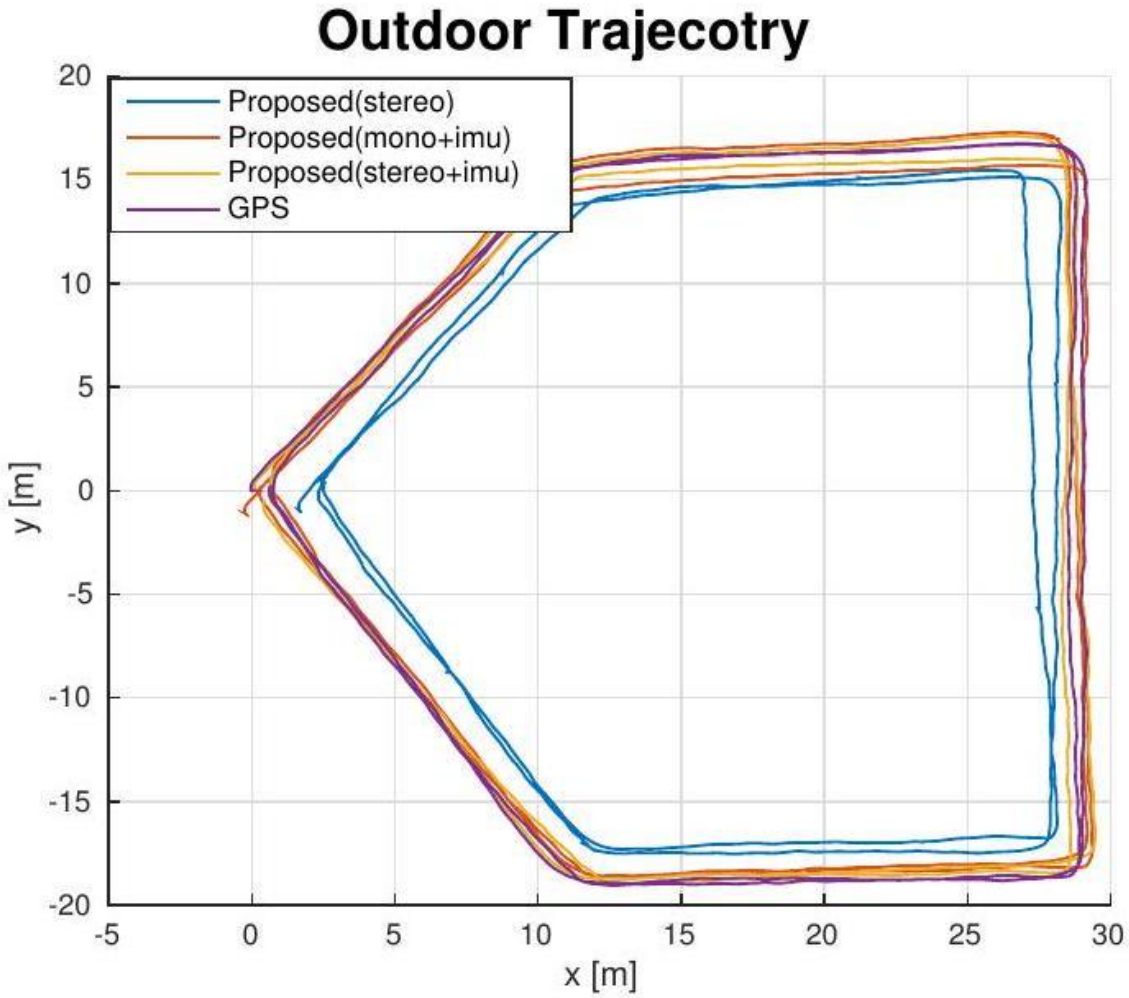

Fig. 7. Estimated trajectories in outdoor experiment.

B. Real-world experiment

In this experiment, we used a self-developed sensor suite to demonstrate our framework. The sensor suite is shown in Fig. 6. It contains stereo cameras (mvBlueFOX-MLC200w, 20 Hz) and DJI A3 controller [2], which inculdes inbuilt IMU (200 Hz) and GPS receiver. The GPS position is treated as ground truth. We hold the sensor suite by hand and walk

[2] http://www.dji.com/a3

Relative Pose Error in Outdoor Dataset

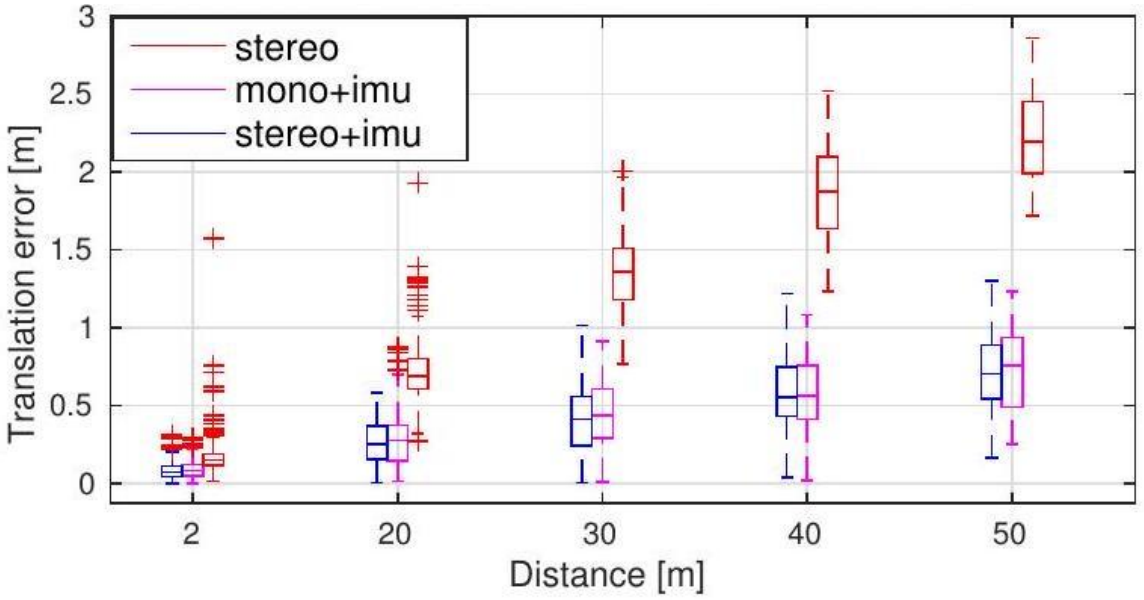

Fig. 8. Relative pose error [26] in outdoor experiment.

around on the outdoor ground. We run states estimation with three different combinations, which are stereo cameras, a monocular camera with an IMU, and stereo cameras with an IMU.

For accuracy comparison, we walked two circles on the ground and compared our estimation with GPS. The trajectory is shown in Fig. 7, and the RPE (Relative Pose Error) is shown in Fig. 8. As same as dataset experiment, noticeable position drifts occurred in the stereo-only scenario. With the assistance of the IMU, the accuracy improves a lot. The RMSE of more outdoor experiments is shown in Table. II. The method which involves the IMU always performs better than the stereo-only case.

VI. Conclusion

In this paper we have presented a general optimizationbased framework for local pose estimation. The proposed framework can support multiple sensor combinations, which is desirable in aspect of robustness and practicability. We further demonstrate it with visual and inertial sensors, which form three sensor suites (stereo cameras, a monocular camera with an IMU, and stereo cameras with an IMU). Note that although we only show the factor formulations for the camera and IMU, our framework can be generalized to other sensors as well. We validate the performance of our system with multiple sensors on both public datasets and real-world experiments. The numerical result indicates that our framework is able to fuse sensor data with different settings.

In future work, we will extend our framework with global sensors (e.g. GPS) to achieve locally accurate and globally aware pose estimation.

TABLE II
RMSE[M] IN OUTDOOR EXPERIMENT.

Sequence	Length	Proposed RMSE		
		stereo	mono+imu	stereo+imu
outdoor1	223.70	1.85	0.71	**0.52**
outdoor2	229.91	2.35	0.56	**0.43**
outdoor3	232.13	2.59	**0.65**	0.75

References

[1] G. Klein and D. Murray, "Parallel tracking and mapping for small ar workspaces," in Mixed and Augmented Reality, 2007. IEEE and ACM International Symposium on, 2007, pp. 225-234.

[2] C. Forster, M. Pizzoli, and D. Scaramuzza, "SVO: Fast semi-direct monocular visual odometry," in Proc. of the IEEE Int. Conf. on Robot. and Autom., Hong Kong, China, May 2014.

[3] J. Engel, T. Schöps, and D. Cremers, "Lsd-slam: Large-scale direct monocular slam," in European Conference on Computer Vision. Springer International Publishing, 2014, pp. 834-849.

[4] R. Mur-Artal, J. Montiel, and J. D. Tardos, "Orb-slam: a versatile and accurate monocular slam system," IEEE Trans. Robot., vol. 31, no. 5, pp. 1147-1163, 2015.

[5] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017.

[6] A. I. Mourikis and S. I. Roumeliotis, "A multi-state constraint Kalman filter for vision-aided inertial navigation," in Proc. of the IEEE Int. Conf. on Robot. and Autom., Roma, Italy, Apr. 2007, pp. 3565-3572.

[7] M. Li and A. Mourikis, "High-precision, consistent EKF-based visualinertial odometry," Int. J. Robot. Research, vol. 32, no. 6, pp. 690-711, May 2013.

[8] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale, "Keyframe-based visual-inertial odometry using nonlinear optimization," Int. J. Robot. Research, vol. 34, no. 3, pp. 314-334, Mar. 2014.

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

[10] R. Mur-Artal and J. D. Tardós, "Visual-inertial monocular slam with map reuse," IEEE Robotics and Automation Letters, vol. 2, no. 2, pp. 796-803, 2017.

[11] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, "On-manifold preintegration for real-time visual-inertial odometry," IEEE Trans. Robot., vol. 33, no. 1, pp. 1-21, 2017.

[12] 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, 2018.

[13] J. Zhang and S. Singh, "Loam: Lidar odometry and mapping in realtime." in Robotics: Science and Systems, vol. 2, 2014, p. 9.

[14] C. Kerl, J. Sturm, and D. Cremers, "Dense visual slam for rgb-d cameras," in Proc. of the IEEE/RSJ Int. Conf. on Intell. Robots and Syst.

[15] H. Rebecq, T. Horstschaefer, G. Gallego, and D. Scaramuzza, "Evo: A geometric approach to event-based 6-dof parallel tracking and mapping in real time," IEEE Robotics and Automation Letters, vol. 2, no. 2, pp. 593-600, 2017.

[16] S. Lynen, M. W. Achtelik, S. Weiss, M. Chli, and R. Siegwart, "A robust and modular multi-sensor fusion approach applied to mav navigation," in Proc. of the IEEE/RSJ Int. Conf. on Intell. Robots and Syst. IEEE, 2013, pp. 3923-3929.

[17] G. P. Huang, A. I. Mourikis, and S. I. Roumeliotis, "Observabilitybased rules for designing consistent ekf slam estimators," Int. J. Robot. Research, vol. 29, no. 5, pp. 502-528, 2010.

[18] S. Shen, Y. Mulgaonkar, N. Michael, and V. Kumar, "Multi-sensor fusion for robust autonomous flight in indoor and outdoor environments with a rotorcraft MAV," in Proc. of the IEEE Int. Conf. on Robot. and Autom., Hong Kong, China, May 2014, pp. 4974-4981.

[19] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "g2o: A general framework for graph optimization," in Proc. of the IEEE Int. Conf. on Robot. and Autom. IEEE, 2011, pp. 3607-3613.

[20] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. J. Leonard, and F. Dellaert, "isam2: Incremental smoothing and mapping using the bayes tree," Int. J. Robot. Research, vol. 31, no. 2, pp. 216-235, 2012.

[21] H. Liu, M. Chen, G. Zhang, H. Bao, and Y. Bao, "Ice-ba: Incremental, consistent and efficient bundle adjustment for visual-inertial slam," in Proc. of the IEEE Int. Conf. on Pattern Recognition, 2018, pp. 19741982.

[22] J. Shi and C. Tomasi, "Good features to track," in Computer Vision and Pattern Recognition, 1994. IEEE Computer Society Conference on, 1994, pp. 593-600.

[23] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in Proc. of the Intl. Joint Conf. on Artificial Intelligence, Vancouver, Canada, Aug. 1981, pp. 24-28.

[24] S. Agarwal, K. Mierle, and Others, "Ceres solver," http://ceres-solver. org.

[25] G. Sibley, L. Matthies, and G. Sukhatme, "Sliding window filter with application to planetary landing," J. Field Robot., vol. 27, no. 5, pp. 587-608, Sep. 2010.

[26] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in Proc. of the IEEE Int. Conf. on Pattern Recognition, 2012, pp. 3354-3361.

[27] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. W. Achtelik, and R. Siegwart, "The euroc micro aerial vehicle datasets," Int. J. Robot. Research, 2016.

[28] B. K. Horn, "Closed-form solution of absolute orientation using unit quaternions," JOSA A, vol. 4, no. 4, pp. 629-642, 1987.

All authors are with the Department of Electronic and Computer Engineering, Hong Kong University of Science and Technology, Hong Kong, China. {tong.qin, jie.pan, shaozu.cao}@connect.ust.hk, eeshaojie@ust.hk.

[1] https://github.com/HKUST-Aerial-Robotics/VINS-Fusion

〈文献翻译二：译文〉

基于广义优化的多传感器局部里程估计框架

Tong Qin 、 Jie Pan 、 Shaozu Cao 和
Shaojie Shen

摘要

当前，越来越多的传感器被集成到机器人系统中，以增强其鲁棒性和自主能力。我们观察到不同平台搭载了多种传感器组合：地面车辆配备立体相机，手机搭载单目相机与惯性测量单元（IMU），而空中机器人则采用立体相机与 IMU 组合。尽管过去提出了多种状态估计算法，但它们通常仅适用于单一传感器或特定传感器组合，鲜有能兼容多传感器选择的方案。本文提出了一种基于通用优化的里程计估计框架，支持多传感器组合。在该框架中，每个传感器都被视为通用因子，共享状态变量的因子会被合并到优化问题中。我们通过视觉传感器与惯性传感器的组合（形成三种传感器组合：立体相机、单目相机与 IMU、立体相机与 IMU）验证了该框架的通用性。系统性能已在公开数据集和多传感器真实场景实验中得到验证，并与现有最先进算法进行对比。我们强调该框架具有通用性，可轻松将不同传感器融合至姿态图优化中。所有实现方案均采用开源代码。[1]

一、引言

实时六自由度（6-DoF）状态估计是机器人技术的核心基础。精准的状态估计在机器人探索、自动驾驶、虚拟现实（VR）和增强现实（AR）等智能应用中发挥着关键作用。这类应用中最常用的传感器当属摄像头。过去几十年间，基于视觉的位姿估计算法层出不穷，例如文献[1]-[5]所述。除了摄像头，惯性测量单元（IMU）也是状态估计的热门选择。IMU 能高频测量加速度和角速度，这正是实时应用中实现低延迟位姿反馈的关键。因此，将视觉与 IMU 结合的研究成果层出不穷，如文献[6]-[12]所示。激光雷达（LiDAR）同样是状态估计的常用传感器。基于 LiDAR 的方法[13]在受限空间内能实现

精准的位姿估计。尽管已有大量算法问世，但这些方法通常仅适用于单一传感器或特定传感器组合。

近期，我们观察到配备多种传感器的平台，例如地面车辆上的立体相机、手机上的单目相机与惯性测量单元（IMU）组合、立体

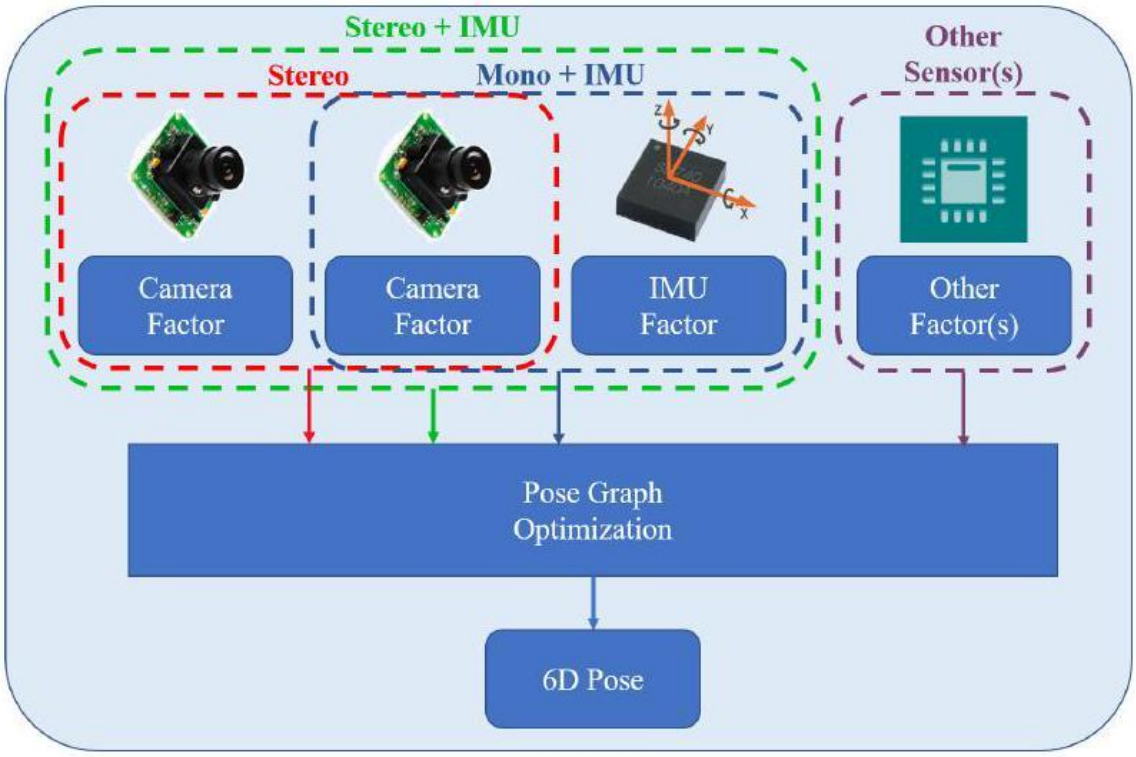

图 1 展示了所提出的状态估计框架示意图，该框架支持多种传感器选择，例如立体相机、配备惯性测量单元（IMU）的单目相机，以及配备 IMU 的立体相机。每个传感器均被视为一个通用因子。具有共同状态变量的因子会被合并，以构建优化问题。
在空中机器人上搭载带有惯性测量单元（IMU）的摄像头。然而，由于传统算法大多针对单一传感器或特定传感器组设计，无法移植到不同平台。即便在同一平台上，不同场景也需要选择不同的传感器组合。因此，需要一种能兼容多种传感器套件的通用算法。另一个实际需求是：当传感器发生故障时，应能快速移除失效传感器并添加替代传感器。由此可见，开发兼容多传感器的通用算法势在必行。

本文提出一种通用的基于优化的位姿估计框架，支持多种传感器组合。我们以视觉传感器与惯性传感器为例，构建了三种传感器组合方案（立体相机、单目相机+惯性测量单元（IMU）、立体相机+IMU），并展示了其灵活性。该框架支持不同传感器组合间的无缝切换。本文的核心贡献体现在以下方面：

- 一种基于全局优化的状态估计框架，支持多传感器协同工作。
- 通过视觉传感器与惯性传感器（形成不同传感器组合：立体相机、单目相机+惯性测量单元（IMU）、立体相机+惯性测量单元（IMU））进行状态估计的详细演示。
- 在公开数据集和真实实验中对所提系统的评估。
- 为社区提供的开源代码。

二、相关工作

状态估计是近几十年来备受关注的研究领域。众多算法致力于实现高精度的六自由度（6-DoF）姿态估计。我们已见证多种基于单一传感器的创新方法，例如视觉方法[1]-[5]、激光雷达（LiDAR）方法[13]、RGB-D 方法[14]以及事件驱动方法[15]。由于单目相机无法获取绝对尺度信息，其姿态估计难以实现六自由度精度。为提升可观测性和鲁棒性，研究者们将具有互补特性的多传感器数据进行融合处理。

多传感器融合方法主要分为两大方向：基于滤波器的方法和基于优化的方法。其中，基于滤波器的方法通常采用扩展卡尔曼滤波器（EKF）实现。在 6 自由度状态估计中，视觉与惯性测量数据通常会共同进行滤波处理。具体而言，高采样率惯性传感器负责状态传播，而视觉测量数据则用于状态更新[9,16]。 MSCKF [6,7]曾是基于 EKF 的视觉惯性里程计（VIO）的典型代表，该方法通过维护多个相机姿态并利用多视角信息来构建多约束更新。基于滤波器的方法往往在状态线性化阶段就存在缺陷，容易因线性化点的不准确性导致误差累积。为解决线性化误差引发的不一致性问题，研究者提出了可观察性约束 EKF [17]以提升精度和一致性。文献[18]则提出了 UKF（无香料卡尔曼滤波器）算法，实现了视觉、激光雷达（Li DAR）和 GPS 测量数据的融合。 UKF 是 EKF 的扩展形式，无需解析雅可比矩阵。值得注意的是，基于滤波器的方法对时间同步性要求极高——任何延迟传入的测量数据都会引发问题，因为状态信息无法在滤波过程中回溯传播。因此需要特殊的排序机制来确保多传感器数据的时序一致性。

基于优化的方法通过同时维护大量测量数据并优化多个变量，这种技术也被称为束调整（BA）。相较于基于滤波器的方法，优化方法在时间同步方面具有显著优势。由于大束数据本身具有天然缓冲特性，即使多个传感器的测量数据存在时间错位，该方法也能轻松处理。在计算复杂度增加的情况下，优化算法在精度方面也优于滤波器方法。早期

的优化求解器如 G2O[19]，主要采用高斯-牛顿法和莱文贝格-马夸特法求解问题。尽管这些算法采用了稀疏结构，但其计算复杂度仍会随着状态数和测量数据量的增加呈二次方增长。为实现实时性能，部分算法探索了增量求解器，而其他算法则通过限制姿态图的规模来优化。iSAM2[20]作为高效的增量求解器，通过复用先前优化结果来减少新测量数据带来的计算负担。其优化迭代过程仅更新少量状态参数。

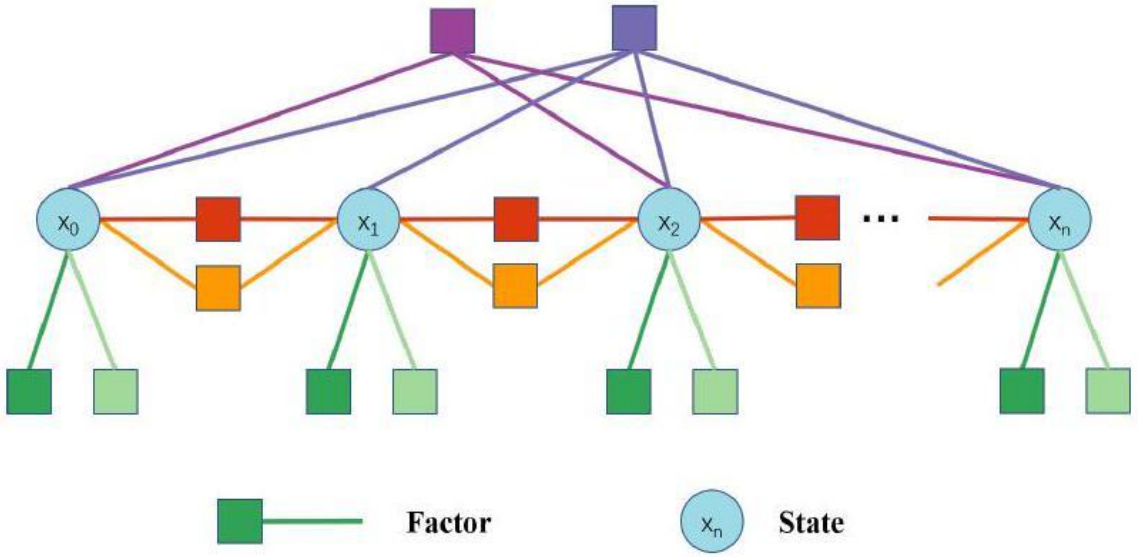

图 2. 姿势图的图形化示意图。每个节点代表某一时刻的状态（位置、方向、速度等）。每条边代表一个因子，该因子由一次测量得出。边约束一个状态、两个状态或多个状态。

相较于完整姿态图的处理方式，文献[21]提出了一种加速求解器，通过将密集结构重构为稀疏块来提升计算效率。保持姿态图固定尺寸的方法被称为滑动窗口法。基于优化的 VIO 方法（如文献[8,12]所述）通过在限定尺寸的滑动窗口内优化变量，将先前状态信息整合为先验因子且不丢失信息。本文[8,10,12]采用基于滑动窗口优化的框架进行状态估计。

三、系统概述

该框架的结构如图 1 所示。多种传感器可自由组合使用，每个传感器的测量数据均作为通用因子处理。这些因子及其对应状态共同构成姿态图，其示意图见图 2。每个节点代表某一时刻的状态（包括位置、方向、速度等），每条边则表示由单次测量数据推导出的因子。因子可约束单个状态、两个状态或多个状态：惯性测量单元（IMU）因子通过连续运动约束实现对两个连续状态的约束；视觉地标因子因需在多帧图像中观测，其约

束作用可覆盖多个状态。完成图构建后，优化过程实质上就是寻找能使所有边匹配度最高的节点配置方案。

本文重点展示了配备视觉与惯性传感器的系统。该系统可采用三种组合方案实现六自由度状态估计：立体相机、单目相机搭配惯性测量单元（IMU），以及立体相机与 IMU 的组合。图 3 展示了该视觉-惯性传感器框架的示意图。姿态图中包含多种相机姿态、IMU 测量数据及视觉测量数据，其中 IMU 和其中一个相机为可选配置。

四、方法

A. 问题定义

1. 状态：我们需要估算的主要状态包括机器人的三维位置和中心方位。此外，

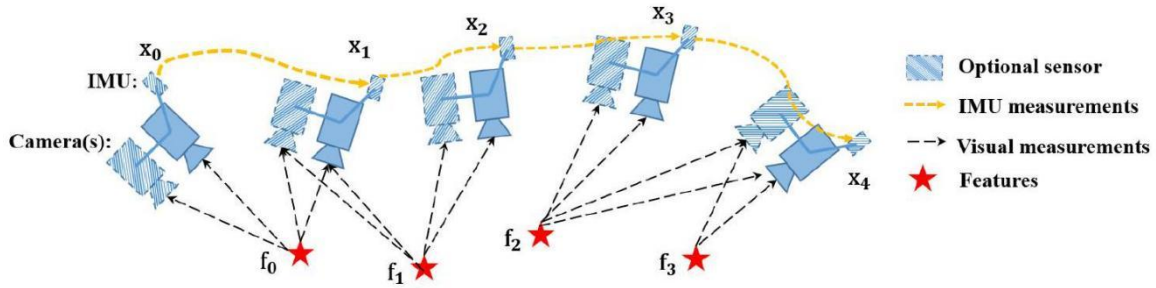

图 3. 所提框架的示意图，包含视觉传感器和惯性传感器。其中 IMU 和单个摄像头为可选配置，因此可形成三种类型：立体摄像头、配备 IMU 的单目摄像头以及配备 IMU 的立体摄像头。姿态图中包含多种摄像头姿态、IMU 测量值及视觉测量值。

存在其他与传感器相关的可选状态。对于摄像头而言，需估算视觉标志物的深度或三维位置；对于惯性测量单元（IMU），其会产生另一个运动变量——速度。此外，还需估算 IMU 的时间变化加速度偏差和陀螺仪偏差。因此，视觉传感器与惯性传感器的完整状态估计定义如下：

$$
\begin{aligned}
X &= [\mathbf{p}_0, \mathbf{R}_0, \mathbf{p}_1, \mathbf{R}_1, \ldots, \mathbf{p}_n, \mathbf{R}_n, \mathbf{x}_{cam}, \mathbf{x}_{imu}] \\
\mathbf{x}_{cam} &= [\lambda_0, \lambda_1, \ldots, \lambda_l] \quad (1) \\
\mathbf{x}_{imu} &= \left[\mathbf{v}_0, \mathbf{b}_{a_0}, \mathbf{b}_{g_0}, \mathbf{v}_1, \mathbf{b}_{a_1}, \mathbf{b}_{g_1}, \ldots, \mathbf{v}_n, \mathbf{b}_{a_n}, \mathbf{b}_{g_n}\right], (1)
\end{aligned}
$$

其中和表示$\mathbf{pRx}_{\text{cam}}\,\lambda\mathbf{x}_{imu}\mathbf{vb}_a\mathbf{b}_g.\mathbf{x}_{imu}$基本系统状态，对应于以世界坐标系表达的物体位置和姿态。是相机相关状态，包含第一帧中观测到的每个特征的深度信息。是惯性测量单元相关变量，由速度、加速度偏差和陀螺仪偏差组成（若仅使用立体相机而不配备惯性

测量单元，则可省略该变量）。传感器中心到物体中心的平移量已知且经过离线校准。为简化符号表示，我们将惯性测量单元定义为物体中心（若未使用惯性测量单元，则将左相机定义为物体中心）。

2)成本函数：状态估计本质上是一个 MLE（最大似然估计）问题。该 MLE 描述了机器人姿态在一段时间内的联合概率分布。在所有测量值相互独立的假设下，该问题通常可表示为：

$$X^{*}=\arg\max_{X}\prod_{t=0}^{n}\prod_{k\in\mathbf{S}}p\left(\mathbf{z}_{t}^{k}|X\right), \tag{2}$$

其中，测量$\mathbf{S}p\left(\mathbf{z}_{t}^{k}|X\right)\sim N\left(\overline{\mathbf{z}}_{t}^{k},\Omega_{t}^{k}\right)$值集合来源于摄像头、惯性测量单元（IMU）及其他传感器。我们假设测量值的不确定性服从高斯分布。因此，上述方程的负对数似然可表示为：

$$\begin{aligned}X^{*}&=\arg\max_{X}\prod_{t=0}^{n}\prod_{k\in\mathbf{S}}\exp\left(-\frac{1}{2}\left\|\mathbf{z}_{t}^{k}-\mathrm{h}_{t}^{k}(X)\right\|_{\mathbf{\Omega}_{t}^{k}}^{2}\right)\\&=\arg\min_{X}\sum_{t=0}^{n}\sum_{k\in\mathbf{S}}\left\|\mathbf{z}_{t}^{k}-\mathrm{h}_{t}^{k}(X)\right\|_{\mathbf{\Omega}_{t}^{k}}^{2}.\end{aligned} \tag{3}$$

马氏距离准则定义为：其中为传感器$\|\mathbf{r}\|_{\mathbf{\Omega}}^{2}=\mathbf{r}^{T}\mathbf{\Omega}^{-1}\mathbf{r}$h(·)模型，其具体细节将在下文详述。随后状态估计被转化为非线性最小二乘问题，该问题亦称为束调整（Bundle Adjustment，BA）。

B. 传感器因素

1. 相机因子：该框架同时支持单目相机和立体相机。每个相机的固有参数及相机间的外在变换关系需预先确定，这些参数可通过离线校准轻松获取。针对每个相机帧，系统会检测角点特征[22]，并通过 KLT 跟踪器[23]在前一帧中实现特征追踪。在立体场景下，跟踪器还会匹配左右图像间的特征。基于特征关联关系，我们构建了每帧中每个特征对应的相机因子。该相机因子实质上是一个重投影过程，能够将特征从首次观测点投射到后续帧中。

基于图像中首次观测到的*lit*特征，后续图像观测的残差定义为：

$$\mathbf{z}_t^l - \mathrm{h}_t^l(X) = \mathbf{z}_t^l - \mathrm{h}_t^l\left(\mathbf{R}_i, \mathbf{p}_i, \mathbf{R}_t, \mathbf{p}_t, \lambda_l\right) = \begin{bmatrix} u_t^l \\ v_t^l \end{bmatrix} - \pi_c\left(\mathbf{T}_c^{b-1}\mathbf{T}_t^{-1}\mathbf{T}_i\mathbf{T}_c^b\pi_c^{-1}\left(\lambda_l, \begin{bmatrix} u_i^l \\ v_i^l \end{bmatrix}\right)\right), (4)$$

其中，是$\left[u_i^l, v_i^l\right] liu_t^l, v_t^l t\pi_c\pi_c^{-1}\mathbf{T}4\times4\begin{bmatrix}\mathbf{R} & \mathbf{p} \\ \mathbf{0} & 1\end{bmatrix}\mathbf{T}_b^c\boldsymbol{\Omega}_t^l$图像中首次出现特征的观测点。[]表示同一特征在图像中的观测值。和是投影函数与反投影函数，其具体形式取决于相机型号（针孔相机、全向相机或其他类型）。是齐次变换，即。为简洁起见，我们省略了部分齐次项。是从物体中心到相机中心的外在变换，该变换需进行离线校准。重投影误差的协方差矩阵在像素坐标系中为常数值，该数值源自相机的内在校准结果。

该因子对左右相机具有普遍性，可以将左图像的特征在时间空间上投影到左图像，也可以将左图像的特征在空间上投影到右图像。

对于不同相机，应采用不同的外在变换。$\mathbf{T}_b^c$

2) IMU 因子：我们采用广为人知的 IMU 预积分算法[11,12]来构建 IMU 因子。假设加速度和陀螺仪测量中的加性噪声均为高斯白噪声。时变加速度和陀螺仪偏差被建模为随机游走过程，其导数为高斯白噪声。由于 IMU 的采样频率高于其他传感器，通常在两个帧之间存在多个 IMU 测量值。因此，我们通过协方差传播在流形上对 IMU 测量值进行预积分。详细的预积分方法可参考文献[12]。在两个时间点 t_1和 t_2之间，预积分可生成相对位置、速度和旋转量。同时，预积分还会传播相对位置、速度和旋转的协方差，以及偏差的协方差。IMU 残差可定义为：$t-1t\boldsymbol{\alpha}_t^{t-1}\boldsymbol{\beta}_t^{t-1}\boldsymbol{\gamma}_t^{t-1}$

$$\mathbf{z}_t^{imu} - \mathrm{h}_t^{imu}(X) = \begin{bmatrix} \boldsymbol{\alpha}_t^{t-1} \\ \boldsymbol{\beta}_t^{t-1} \\ \boldsymbol{\gamma}_t^{t-1} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} \ominus \begin{bmatrix} \mathbf{R}_{t-1}{}^{-1}\left(\mathbf{p}_t - \mathbf{p}_{t-1} + \frac{1}{2}\mathbf{g}dt^2 - \mathbf{v}_{t-1}dt\right) \\ \mathbf{R}_{t-1}{}^{-1}(\mathbf{v}_t - \mathbf{v}_{t-1} + \mathbf{g}dt) \\ \mathbf{R}_{t-1}{}^{-1}\mathbf{R}_t \\ \mathbf{b}_{a_t} - \mathbf{b}_{a_{t-1}} \\ \mathbf{b}_{g_t} - \mathbf{b}_{g_{t-1}} \end{bmatrix} (5)$$

其中，是$\ominus dt$流形上的减运算，专门用于非线性转动，是两个时间点之间的时间间隔，g 是已知的重力矢量，其范数约为 9.81，每两个相邻的帧构成一个 IMU 因子，在代价函数中。

3) 其他因素：虽然我们仅针对相机和惯性测量单元（IMU）进行建模，但该系统并不局限于这两种传感器。诸如车轮转速表、激光雷达（LiDAR）和雷达等其他传感器，均可轻松集成至系统中。关键在于将这些测量数据建模为通用残差因子，并将其纳入成本函数。

C. 优化

传统上，方程 3 的非线性最小二乘问题通过牛顿-高斯法或 Levenberg-Marquardt 法求解。成本函数相对于初始状态估计值进行线性化处理，此时成本函数可表示为：$\hat{X}$

$$\arg\min_{\delta X} \sum_{t=0}^{n} \sum_{k \in \mathbf{S}} \left\| \mathbf{e}_t^k + \mathbf{J}_t^k \delta X \right\|_{\mathbf{\Omega}_t^k}^2, \tag{6}$$

其中为各$\mathbf{J}\hat{X}\delta X$因子相对于当前状态的雅可比矩阵。经过线性化近似后，该成本函数具有闭式解。以 NewtonGaussian 为例，其解推导如下：

$$\underbrace{\sum \sum \mathbf{J}_t^{k^T} \mathbf{\Omega}_t^{k^{-1}} \mathbf{J}_t^k}_{\mathbf{H}} \delta X = \underbrace{-\sum \sum \mathbf{J}_t^{k^T} \mathbf{\Omega}_t^{k^{-1}} \mathbf{e}_t^k}_{\mathbf{b}}. \tag{7}$$

最终，当前状态通过加法$\hat{X}\hat{X}\oplus\delta X\oplus$运算更新，其中为流形上的旋转加法操作。该过程将迭代多次直至收敛。我们采用 Ceres 算法。

为解决该问题，采用求解器[24]，该工具运用先进的数学工具高效获取稳定且最优的结果。

D. 被边缘化

随着状态数量随时间推移不断增加，计算复杂度将呈平方级增长。为有效控制计算复杂度，我们引入了边际化处理方法，确保有效信息不丢失。该方法将先前测量值转化为先验项，保留历史信息。被边缘化的状态集合记为，剩余状态集合记为。通过累加所有边缘化因子（公式 7），我们得到新的和。重新排列状态顺序后，可得出以下关系式：$X_m X_r \mathbf{Hb}$

$$\begin{bmatrix} \mathbf{H}_{mm} & \mathbf{H}_{mr} \\ \mathbf{H}_{rm} & \mathbf{H}_{rr} \end{bmatrix} \begin{bmatrix} \delta X_m \\ \delta X_r \end{bmatrix} = \begin{bmatrix} \mathbf{b}_m \\ \mathbf{b}_r \end{bmatrix} \tag{8}$$

边缘化处理采用 Schur 补集[25]进行，具体如下：

$$\underbrace{\left(\mathbf{H}_{rr}-\mathbf{H}_{rm}\mathbf{H}_{mm}^{-1}\mathbf{H}_{mr}\right)}_{\mathbf{H}_p}\delta X_r=\underbrace{\mathbf{b}_r-\mathbf{H}_{rm}\mathbf{H}_{mm}^{-1}\mathbf{b}_m}_{\mathbf{b}_p} \tag{9}$$

我们为剩余状态获得$\mathbf{H}_p,\mathbf{b}_p$新的先验。关于边缘化状态的信息被转换为先验项且无任何损失。具体而言，系统中保留十个空间相机帧。当新关键帧出现时，我们对与首帧状态相关的视觉和惯性因素进行边缘化处理。

在获取当前状态的先验信息后，根据贝叶斯定理，我们可以将后验概率计算为似然函数与先验概率的乘积：。此时状态估计问题转化$p(X|\mathbf{z})\propto p(\mathbf{z}|X)p(X)mnm$为最大后验概率（MAP）问题。假设我们通过滑动窗口保存从某一时刻到下一时刻的状态。先前状态被边缘化并转化为先验项。因此，MAP 问题可表示为：

$$X_{m:n}^{*}=\arg\max_{X_{m:n}}\prod_{t=m}^{n}\prod_{k\in\mathbf{S}}p\left(\mathbf{z}_t^k|X_{m:n}\right)p(X_{m:n})$$

$$=\arg\min_{X_{m:n}}\sum_{t=m}^{n}\sum_{k\in\mathbf{S}}\left\|\mathbf{z}_t^k-\mathrm{h}_t^k(X_{m:n})\right\|_{\Omega_t^k}^2 \tag{10}$$

$$+\left(\mathbf{H}_p\delta X_{m:n}-\mathbf{b}_p\right) \tag{10}$$

与公式 3 相比，上述方程仅增加了一个先验项。其求解方法与公式 3 相同，均采用 Ceres 求解器[24]进行计算。

E. 讨论

该系统是一个通用框架，只要能推导出通用残差因子，各种传感器都能轻松加入。由于系统并非专为特定传感器设计，因此能够处理传感器失效情况。当传感器失效时，我们只需移除失效传感器的因子，并从其他备用传感器中添加新因子。

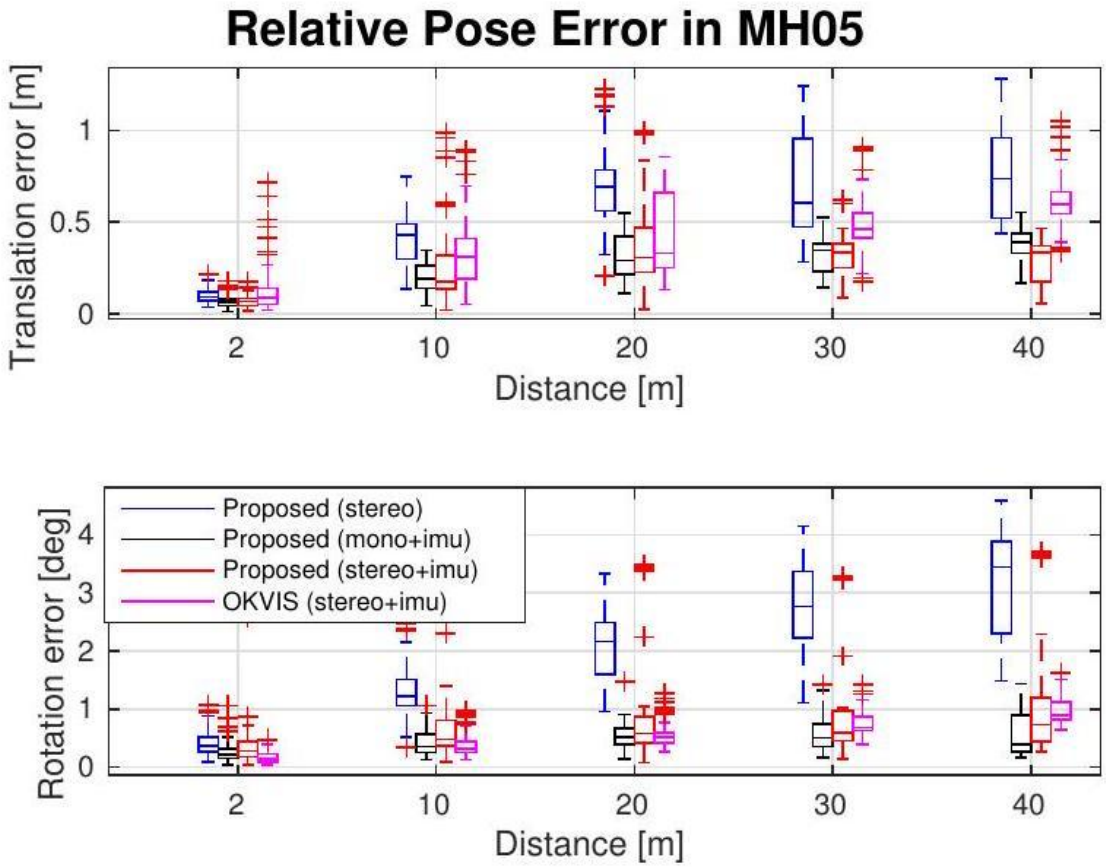

图 4. MH_05_difficult 中的相对姿态误差[26]。两个图分别展示了平移和旋转的相对误差。

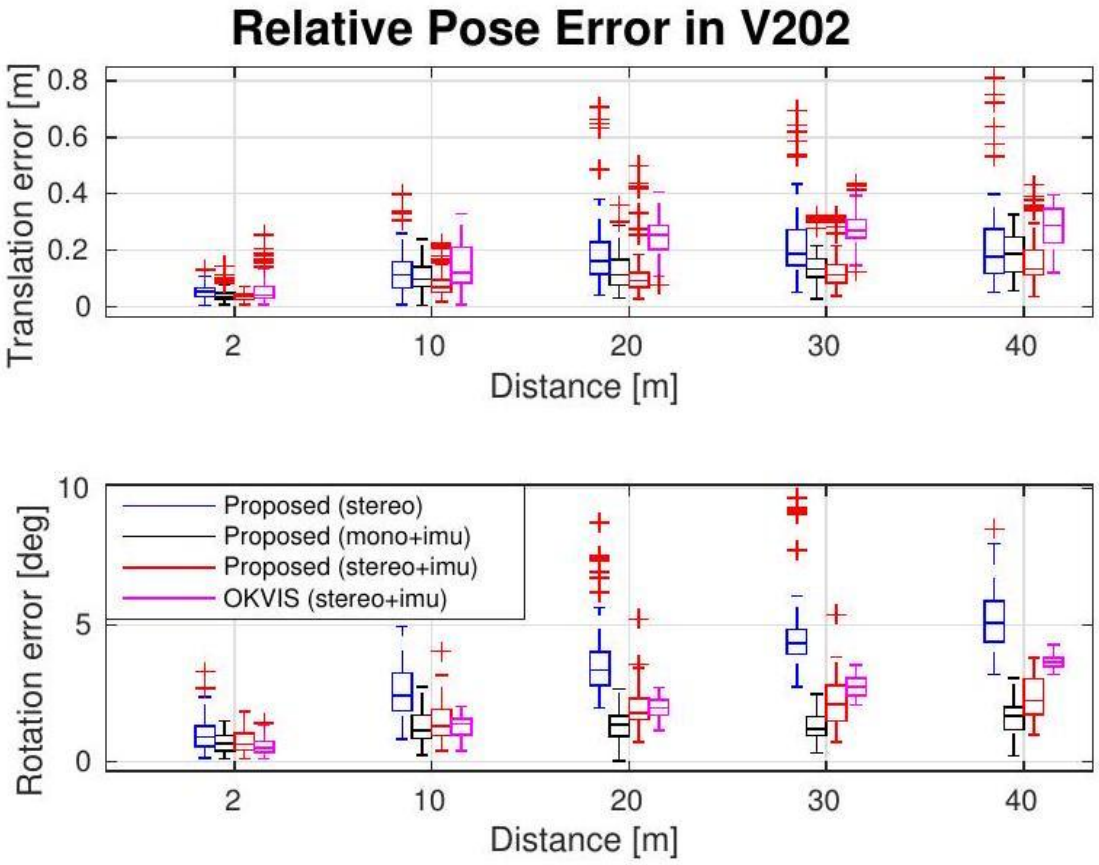

图 5. V2_02_medium 中的相对姿态误差[26]。两个图分别展示了平移和旋转的相对误差。

五、实验结果

我们通过视觉传感器和惯性传感器对所提出的系统进行了数据集评估和真实世界实验验证。在首次实验中，我们使用公开数据集将所提算法与另一项前沿算法进行对比。随后在大规模户外环境中测试了该系统。数值分析结果详细展示了系统的精确度。

A. 数据集

我们采用欧洲航天局 MAV 视觉惯性数据集[27]来评估所提出的系统。该数据集由微型飞行器采集，包含立体图像（Aptina MT9V034 全局快门，单色，20752×480

表 I

EuRoC 数据集中的 RMSE [m]。

序列	长度	建议的 RMSE			OKVIS RMSE
		立体声	单极免疫	立体声+惯性测量	

MH_01	79.84	0.54	0.18	0.24	0.16
MH_02	72.75	0.46	0.09	0.18	0.22
MH_03	130.58	0.33	0.17	0.23	0.24
MH_04	91.55	0.78	0.21	0.39	0.34
MH_05	97.32	0.50	0.25	0.19	0.47
V1_01	58.51	0.55	0.06	0.10	0.09
V1_02	75.72	0.23	0.09	0.10	0.20
V1_03	78.77	x	0.18	0.11	0.24
V2_01	36.34	0.23	0.06	0.12	0.13
V2_02	83.01	0.20	0.11	0.10	0.16
V2_03	85.23	x	0.26	0.27	0.29

（FPS），同步 IMU 测量（ADIS16448,200Hz），此外，真实状态由 VICON 和 LeicaMS50 提供。我们运行了三种不同传感器组合的数据集，包括立体相机、单目相机与 IMU 组合、以及单独的立体相机与 IMU 组合。

在本实验中，我们将研究结果与 OKVIS [8]进行对比。该研究采用最先进的视觉惯性光学（VIO）技术，通过立体相机和惯性测量单元（IMU）实现视觉-惯性同步。 OKVIS 是另一种基于优化的滑动窗口算法，专为视觉惯性传感器设计，而我们的系统则是一个更通用的框架，支持多种传感器组合。我们使用 EuRoC 数据集中的所有序列对所提出的框架和 OKVIS 进行了测试，并通过相对姿态误差（RPE）和绝对轨迹误差（ATE）进行评估。RPE 的计算采用文献[26]中提出的工具。图 4 和图 5 分别展示了 MH_05_difficult 和 V2_02_medium 两个序列的 RPE（相对姿态误差）曲线。

表 I 展示了 EuRoC 数据集中所有序列的平均绝对误差（RMSE）RMSE 。通过 Horn 方法[28]将估计轨迹与真实轨迹进行对齐。仅使用立体视觉的方案在 V1_03_difficult 和 V2_03_difficult 序列中失效，因为这些序列的运动过于剧烈，导致视觉追踪无法有效追

踪。而采用惯性测量单元（IMU）的方法在所有序列中均表现良好。这充分证明，当视觉追踪因光照变化、无纹理区域或运动模糊而失效时，IMU 能通过填补数据空白显著提升运动追踪性能。

从相对姿态误差和绝对轨迹误差来看，纯立体视觉方法在多数场景中表现最差。在纯立体视觉情况下，位置和旋转漂移会随着距离增加而明显加剧。换句话说，惯性测量单元（IMU）在状态估计中显著提升了视觉系统的性能。由于 IMU 能测量重力矢量，因此能有效抑制滚转角和俯仰角的漂移。配备 IMU 的立体相机并非总能表现最佳，因为相比单目相机，其需要更精确的校准。若内校准和外校准不准确，系统会引入更多噪声。总体而言，多传感器融合能显著提升系统的鲁棒性。我们的实验结果在多数场景中均优于 OKVIS 。

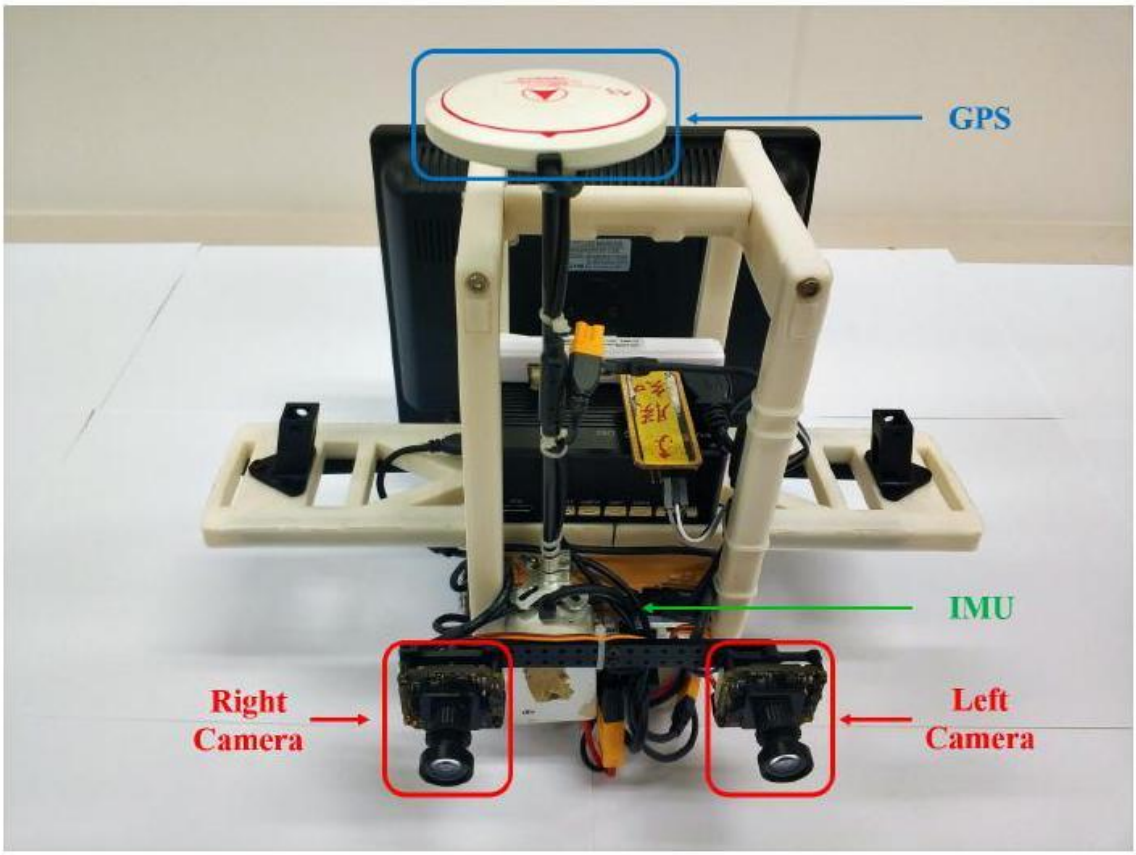

图 6. 用于户外环境的自主研发传感器套件。该套件包含立体相机（mvBlueFOX-MLC200w，20 Hz）和 DJI A3 控制器，后者内置惯性测量单元（IMU）及 GPS 接收器。(200 Hz)

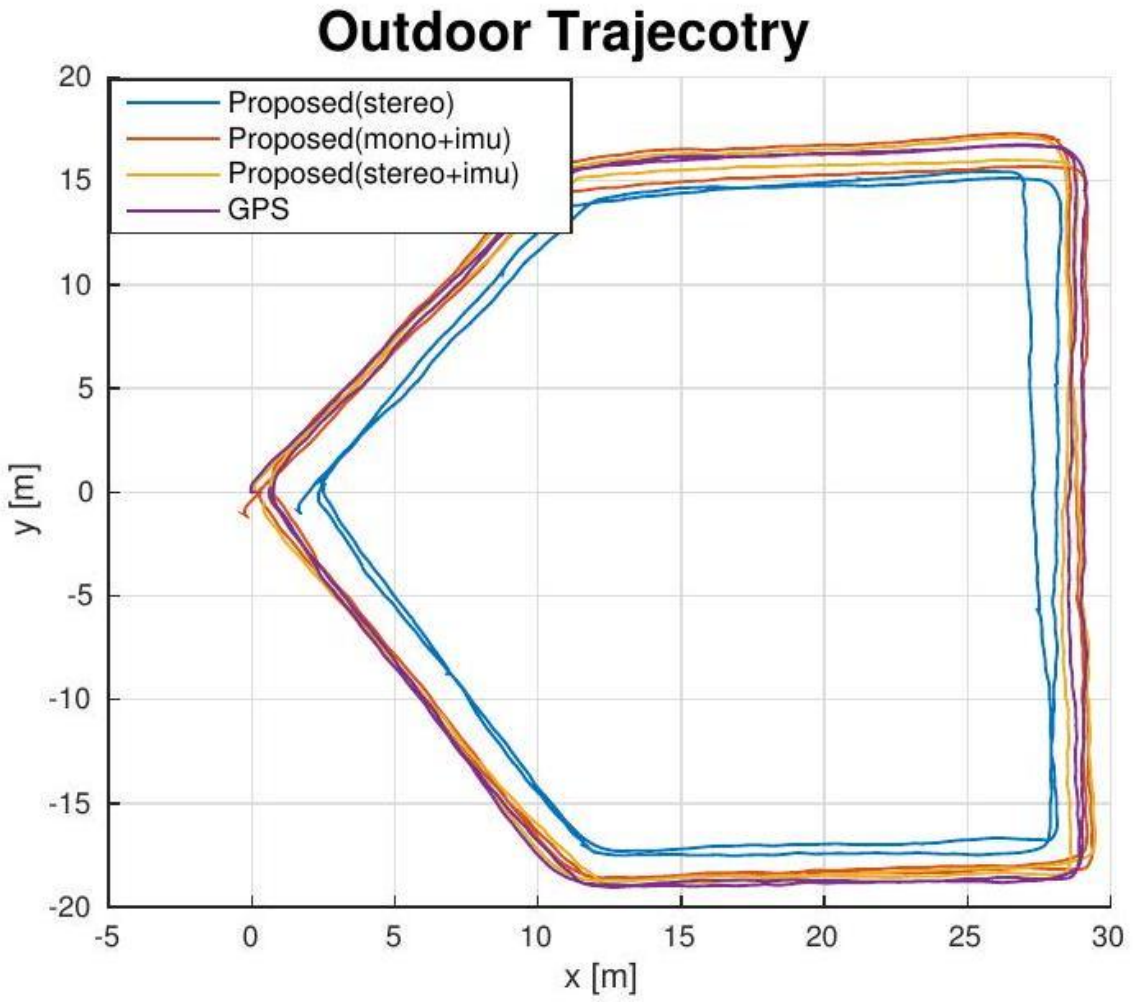

图 7. 户外实验中的估算轨迹。

B. 现实世界实验

本实验采用自主研发的传感器套件验证框架性能。如图 6 所示，该套件包含立体相机（型号 mvBlueFOX-MLC200w，20Hz）和 DJI A3 控制器（内置惯性测量单元 IMU 及 GPS 接收器），其中 [2](200 Hz)GPS 定位数据作为基准参考。实验中我们手持该传感器套件进行移动测试。

[2]http://www.dji.com/a3

户外数据集的相对位姿误差

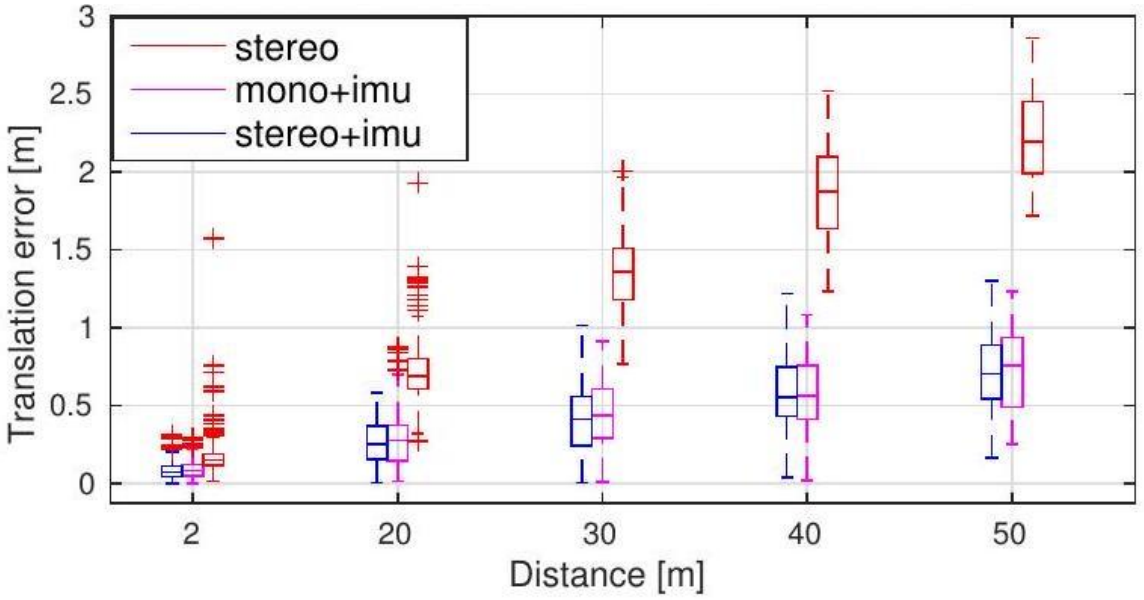

图 8. 户外实验中的相对姿态误差[26]。

在户外场地进行状态估计，采用三种不同组合：立体相机、单目相机与惯性测量单元（IMU）组合，以及立体相机与 IMU 组合。

为验证精度，我们在地面行走两个圆周后，将估算结果与 GPS 数据进行对比。轨迹如图 7 所示，相对姿态误差（RPE）如图 8 所示。与数据集实验结果一致，仅使用立体视觉时会出现明显的位置漂移。借助惯性测量单元（IMU）辅助后，精度显著提升。更多户外实验的 RMSE 见表 II。采用 IMU 的方法始终优于仅使用立体视觉的情况。

六、结语

本文提出了一种基于优化的通用局部姿态估计框架。该框架支持多种传感器组合，这在鲁棒性和实用性方面具有显著优势。我们通过视觉传感器与惯性传感器的组合进行了验证，形成了三种传感器组：双目相机、单目相机搭配惯性测量单元（IMU），以及双目相机搭配 IMU。需要说明的是，虽然本文仅展示了相机与 IMU 的因子公式，但该框架可推广至其他传感器组合。我们在公共数据集和真实场景实验中对多传感器系统进行了性能验证。数值结果表明，该框架能够有效融合不同配置的传感器数据。

在后续工作中，我们将通过引入全局传感器（如 GPS）扩展现有框架，以实现局部精确且全局感知的姿态估计。

表 II

户外实验中的均方根误差。

序列	长度	建议的 RMSE		
		立体声	单极免疫	立体声+惯性测量
户外 1	223.70	1.85	0.71	**0.52**
户外 2	229.91	2.35	0.56	**0.43**
户外 3	232.13	2.59	**0.65**	0.75

参考文献

[1]G.Klein 与 D.Murray 合著的《小型 Ar 工作空间的并行跟踪与映射》，收录于《混合与 Augmented Reality 》2007 年 IEEE 与 ACM 国际研讨会论文集，第 225-234 页。

[2]C.Forster、M.Pizzoli 和 D.Scaramuzza 合著的《 SVO ：快速半直接单目视觉里程计》，收录于 2014 年 5 月中国香港 IEEE 国际机器人与自动化会议论文集。

[3] J. Engel、T. Schops 和 D. Cremers，《Lsd-slam：大规模单目直接 SLAM》，载于《欧洲计算机视觉会议论文集》，Springer International Publishing，2014 年，第 834-849 页。

[4] R. Mur-Artal、J. Montiel 和 J. D. Tardos，《Orb-slam：一种多功能且精确的单目 SLAM 系统》，IEEE 机器人学汇刊，第 31 卷第 5 期，第 1147-1163 页，2015 年。

[5] J. Engel、V. Koltun 和 D. Cremers，《直接稀疏里程计》，《IEEE 模式分析与机器智能汇刊》，2017 年。

[6] A. I. Mourikis 与 S. I. Roumeliotis 合著的《视觉辅助惯性导航的多状态约束卡尔曼滤波器》，收录于《IEEE 国际机器人与自动化会议论文集》（2007 年 4 月，意大利罗马），第 3565-3572 页。

[7]李明和 A.莫里基斯，《基于视觉惯性 EKF 的高精度视觉惯性里程计》，《国际机器人研究杂志》，第 32 卷第 6 期，第 690-711 页，2013 年 5 月。

[8] S. Leutenegger、S. Lynen、M. Bosse、R. Siegwart 和 P. Furgale，《基于关键帧的视觉惯性里程计非线性优化方法》，《国际机器人研究杂志》，第 34 卷第 3 期，第 314-334 页，2014 年 3 月。

[9] M. Bloesch、S. Omari、M. Hutter 和 R. Siegwart 合著的《基于直接扩展卡尔曼滤波的鲁棒视觉惯性里程计》一文，发表于 2015 年 IEEE/RSJ 国际智能机器人与系统会议论文集，第 298-304 页。

[10] R. Mur-Artal 与 J. D. Tardos 合著的《基于地图重用的视觉-惯性单目 SLAM》发表于《IEEE 机器人与自动化快报》2017 年第 2 卷第 2 期，第 796-803 页。

[11] C. Forster、L. Carlone、F. Dellaert 和 D. Scaramuzza，《实时视觉惯性里程计的流形预积分方法》，IEEE 机器人学汇刊，第 33 卷第 1 期，第 1-21 页，2017 年。

[12] T. Qin、P. Li 和 S. Shen，《Vins-mono：一种稳健且多功能的单目视觉-惯性状态估计器》，IEEE 机器人学汇刊，第 34 卷第 4 期，第 1004-1020 页，2018 年。

[13] 张杰与辛格 S.，《Loam：实时激光雷达里程计与测绘》，载于《机器人学：科学与系统》第 2 卷，2014 年，第 9 页。

[14] C. Kerl、J. Sturm 和 D. Cremers，《RGB-D 相机的密集视觉碰撞检测》，载于《IEEE/RSJ 国际智能机器人与系统会议论文集》

[15] H. Rebecq、T. Horstschaefer、G. Gallego 和 D. Scaramuzza 合著的《Evo：基于几何的实时事件驱动六自由度并行跟踪与映射方法》，发表于《IEEE 机器人与自动化快报》2017 年第 2 卷第 2 期，第 593-600 页。

[16] S. Lynen、M. W. Achtelik、S. Weiss、M. Chli 与 R. Siegwart 合著的《一种稳健且模块化的多传感器融合方法在微型飞行器导航中的应用》，载于《IEEE/RSJ 国际智能机器人与系统会议论文集》（2013 年），第 3923-3929 页。

[17] 黄 G.P.、穆里基斯 A.I.与鲁梅利奥蒂斯 S.I.，《基于可观测性的规则设计一致的 ekf slam 估计器》，《国际机器人研究杂志》第 29 卷第 5 期，第 502-528 页，2010 年。

[18]S.Shen、Y.Mulgaonkar、N.Michael 和 V.Kumar 合著的《多传感器融合技术在旋翼机 MAV 室内外环境中的鲁棒自主飞行》，收录于 2014 年 5 月中国香港 IEEE 国际机器人与自动化会议论文集，第 4974-4981 页。

[19] R. Kummerle、G. Grisetti、H. Strasdat、K. Konolige 和 W. Burgard 合著的《g2o：图优化通用框架》，收录于《IEEE 国际机器人与自动化会议论文集》（2011 年），第 3607-3613 页。

[20] M. Kaess、H. Johannsson、R. Roberts、V. Ila、J. J. Leonard 和 F. Dellaert，《isam2：基于贝叶斯树的增量平滑与映射》，《国际机器人研究杂志》，第 31 卷第 2 期，第 216-235 页，2012 年。

[21] 刘浩、陈明、张刚、鲍浩和鲍勇，《Ice-ba：视觉惯性 SLAM 的增量、一致且高效的束调整方法》，载于《IEEE 国际模式识别会议论文集》2018 年，第 1974-1982 页。

[22]J.Shi 与 C.Tomasi 合著的《可追踪的优良特征》，收录于 1994 年 IEEE 计算机学会会议论文集《 Computer Vision 与 Pattern Recognition 》，第 593-600 页。

[23] B. D. Lucas 与 T. Kanade 合著的《一种迭代图像配准技术及其在立体视觉中的应用》，收录于《国际人工智能联合会议论文集》，加拿大温哥华，1981 年 8 月，第 24-28 页。

[24] S. Agarwal、K. Mierle 等人，《Ceres 求解器》，http://ceres-solver.org。

[25] G. Sibley、L. Matthies 和 G. Sukhatme，《滑动窗口滤波器在行星着陆中的应用》，《野外机器人学杂志》，第 27 卷第 5 期，第 587-608 页，2010 年 9 月。

[26] A. Geiger、P. Lenz 和 R. Urtasun，《我们准备好迎接自动驾驶了吗？——Kitti 视觉基准测试套件》，载于《IEEE 国际模式识别会议论文集》，2012 年，第 3354-3361 页。

[27] M. Burri、J. Nikolic、P. Gohl、T. Schneider、J. Rehder、S. Omari、M. W. Achtelik 和 R. Siegwart，《欧洲微型飞行器数据集》，《国际机器人研究杂志》，2016 年。

[28]B.K.Horn，《单位四元数求绝对方位的闭式解》， JOSA A，第 4 卷第 4 期，第 629-642 页，1987 年。