Robust Visual-Inertial Odometry Based on a Kalman Filter and Factor Graph

Zhiwei Wang, Bao Pang, Yong Song, *Member, IEEE*, Xianfeng Yuan,
Qingyang Xu, and Yibin Li

***Abstract*—We present a real-time, high-accuracy, robust, tightly coupled visual-inertial odometry (VIO) algorithm, including monocular-inertial odometry and stereo-inertial odometry, and uses inertial measurement unit (IMU) pre-integration that is based on fourth-order Runge–Kutta (PK4) and IMU initialization based on maximum a posteriori (MAP) estimation. In particular, we used the multi-state constraint Kalman filter (MSCKF) to fuse vision and IMU measurement data for state estimation. In the optimization stage, we simultaneously considered and optimized all of the historical constraints, and performed multiple iterations to reduce the linearity errors. For further reducing the cumulative error and improving the relocation accuracy, we used a bag-of-words model for global optimization. To lower the computational cost and increase the real-time performance, we set keyframe insertion mechanism and introduced sliding window, and used a new form of Kalman gain that converts the Kalman gain in multi-state constraint Kalman filtering into the inverse of the state dimension. We validated the proposed method by using the EuRoC MAV dataset and KITTI dataset. We performed physics experiments in an outdoor environment with unstable light, to further validate the accuracy and robustness of our method.**

***Index Terms*—Factor graph optimization, IMU initialization, Kalman filter, state estimation, visual-inertial odometry.**

I. Introduction

SIMULTANEOUS Localization and Mapping (SLAM) technology is the first key technical problem to be solved by intelligent mobile robots. Its core is state estimation; thus, accurate state estimation is the premise for stable operation of intelligent mobile robots. In the past 20 y, implementation of SLAM technology has mainly used a single sensor (such as a camera or LiDAR), with substantial success. Compared with LiDAR systems, cameras are widely used due to their smaller size, lower cost, better performance in rich texture environments, and extraction of semantic information [1], [2], [3], [4]. However, vision systems exhibit large drifts when under aggressive motion and are unable to recover scales, which limits their practical applications in intelligent mobile robots. Inertial measurement units (IMUs)—which are low cost, have high accuracy under fast motion, and have fusion vision sensors—can observe scales information and improve the motion-tracking performance markedly. Therefore, integration of vision systems and IMUs is increasingly common.

Visual-inertial systems (VIO) can be divided into two broad categories: filter-and optimization-based methods. Current popular filter-based methods include MSCKF [5] and ROVIO [6]; optimization-based methods include OKVIS [7] and VINS_Mono [8]. Filter-based methods assume Markov propertes, and the current frame state is only related to the previous frame state and regardless of previous history frame state. Therefore, the filtering-based method has less computational load and running time, but in the case of long running distance, it will produce a large cumulative error. The factor graph optimization method is proposed on the basis of Bayesian network, which considers the relationship between the current state and all previous historical states. Therefore, this method is beneficial to reduce the cumulative error, but it will lead to poor real-time performance of the system. To enhance the coupling degree of vision and IMU measurement data, as well as improve the accuracy and robustness of the VIO system while maintaining high real-time performance, we propose a tightly coupled VIO framework that consists of a filter-based odometer module and a factor graph-based optimization module.To solve the problem of insufficient correlation between current and historical information as well as large linearity error in filter-based methods, we set keyframe insertion mechanism, and used multi-state constraint Kalman filter (MSCKF) for state estimation at the normal frame level between two keyframes. We then used the sliding-window-based factor graph optimization method for local map optimization at the keyframe level; and used a bag-of-words model, loop-closure detection method for global optimization. The main contributions of our work are as follows:

- We built a tightly coupled VIO framework consisting of a MSCKF-based odometry module and a factor graph-based optimization module. We set up a new keyframe insertion method and proposed a new methodology for correlating all of the historical information and reducing the linearity errors—by using filtering and optimization. We used MSCKF for vision and IMU measurements fusion and poses propagation and updating at the normal frames level between two keyframes; and set a

Manuscript received 1 May 2022; revised 17 October 2022, 13 January 2023, and 7 March 2023; accepted 13 March 2023. This work was supported in part by the Science and Technology Innovation 2030-Major Project under Grant 2020AAA0108903 and in part by the National Natural Science Foundation of China under Grant 61973184 and Grant 61803227. The Associate Editor for this article was J. Li. *(Corresponding author: Yong Song.)*

Zhiwei Wang, Bao Pang, Yong Song, Xianfeng Yuan, and Qingyang Xu are with the School of Mechatronics and Information Engineering, Shandong University, Weihai 264209, China (e-mail: 202017456@mail.sdu.edu.cn; pangbao@sdu.edu.cn; songyong@sdu.edu.cn; yuanxianfeng@sdu.edu.cn; qingyangxu@sdu.edu.cn).

Yibin Li is with the School of Control Science and Engineering, Shandong University, Jinan 250012, China (e-mail: liyb@sdu.edu.cn).

Digital Object Identifier 10.1109/TITS.2023.3258526

TABLE I

THE MAIN TECHNOLOGY OF THE MOST REPRESENTATIVE VISUAL AND VISUAL-INERTIAL SYSTEMS

System name	Sensor	Feature extraction	Feature matching	State estimation	State estimation
MonoSLAM [10]	Mono	Shi Tomasi	Correlation	EKF	-
PTAM [11]	Mono	FAST	Pyramid	BA	-
SVO [1]	Mono,Stereo,Fisheye	FAST	Direct	Local BA	-
LSD-SLAM [2]	Mono,Stereo	Edgelets	Direct	PG	PG,FABMAP
ORB-SLAM [3]	Mono	ORB	Descriptor	Local BA	PG+BA,DBoW2
ORB-SLAM2 [4]	Mono,Stereo,RGB-D	ORB	Descriptor	Local BA	PG+BA,DBoW2
DSO [14]	Mono,Stereo,Fisheye	High grad.	Direct	Local BA	-
DSM [15]	Mono	High grad.	Direct	Local BA	-
MSCKF [5]	Mono,Mono+IMU,Stereo+IMU	Shi Tomasi	Cross Correlation	EKF	-
ROVIO [6]	Fisheye,Mono+IMU,Stereo+IMU	Shi Tomasi	Direct	EKF	-
OKVIS [7]	Fisheye,Mono+IMU,Stereo+IMU	BRISK	Descriptor	Local BA	-
VINS-Mono [8]	Mono,Fisheye,Mono+IMU	Shi Tomasi	KLT	Local BA	PG,DBoW2
VINS-Fusion [19]	Stereo,Fisheye,Mono+IMU,Stereo+IMU	Shi Tomasi	KLT	Local BA	PG,DBoW2
ORB-SLAM3 [23]	Mono,Stereo,Fisheye,Mono+IMU,Stereo+IMU	ORB	Descriptor	Local BA	PG+BA,DBoW2
our	Mono+IMU,Stereo+IMU	ORBShi Tomasi	KLT	EKF+Local BA	PG,DBoW2

sliding window; constructed IMU constraint, prior, visual observation, and loop-closure factors for local as well as global optimization of keyframe pose; and performed factor graph optimization to reduce cumulative error and linearity error while ensuring high real-time performance.

- We evaluate the performance of our algorithm on representative indoor and outdoor public benchmark datasets, and set up a real environment experiment to verify the effectiveness of our method.

This work improves the accuracy and robustness of VIO system state estimation. Our work will serve as a baseline for VIO system research and advance the state-of-the-art in visual-inertial odometry techniques.

II. RELATED WORK

There is extensive academic work related to visual SLAM. In this section, we review current work: pure vision state estimation and mapping, visual-inertial fusion SLAM methods. Table I summarizes representative visual and visual-inertial systems, indicating the main techniques used for state estimation, fusion methods, and pose optimization.

A. Pure Vision State Estimation and Mapping

A large number of high-precision pure visual state estimation algorithms have been proposed in the past few decades. Reference [10] applied the Structure From Motion (SFM) method to SLAM for the first time, using the probabilistic model framework to create a sparse but stable map online, which laid the foundation for the development of visual SLAM. A SLAM algorithm that separated tracking and mapping as two threads were first proposed in [11], applying FAST to extract feature points, and introducing a key frame mechanism as well as a non-linear optimization scheme. LSD_SLAM [1] uses the feature extraction method based on the direct method, and improves the speed of feature tracking. The back-end optimization uses pose-graph (PG) optimization, and directly tracks as well as maintains the depth map, but the accuracy is lower than in PTAM [11]. A pure visual odometry (VO) method, SVO [2], uses direct methods to track image features efficiently; but short-term data association limits its accuracy. However, ORB_SLAM [3], [4] systems based on the Oriented FAST and Rotated BRIEF (ORB) feature extraction method provide short- and medium-term data association, local optimization using Bundle Adjustment (BA), as well as a bag-of-words model (DBoW2) [12], [13] for loop closure detection and relocation in a manner that achieves long-term data association. The three types of data association are the key to stable orientation in large-scale indoor and outdoor environments. ORB-SLAM renders the SLAM system more modular and is an important milestone in the development of visual SLAM. In recent years, a relatively novel direct sparse mileage calculation method [14], [15], which uses the method of uniformly sampling key points on the entire image instead of the direct method to add an a priori method, has been proposed, that substantially improves the real-time functionality of the SLAM system. However, the system lacks long-term data correlation, resulting in lower accuracy.

B. Visual-Inertial State Estimation and Mapping

Inertial odometry has high accuracy at high frequency, which can increase the accuracy of visual odometry at high speed. Fusing vision and IMU can still be robust in complex environments where a single sensor fails, such as a lack of texture features or aggressive motion. Loose and tight coupling are the two main fusion forms of vision and IMU. There is not much work on visual-inertial fusion based on loose coupling, and the fusion effect is lower. Consequently, the tightly coupled visual-inertial fusion technique (based on filtering and optimization methods) is the mainstream of research.

The filtering-based method can be traced back to MSCKF (which integrates vision and IMU information on the basis of the EKF framework) and adds camera poses at different times to the state vector (which solves the EKF_SLAM state vector dimension explosion problem). A new, real-time VIO algorithm that is capable of consistent estimation was proposed in [16], which refined the MSCKF and was extended to the stereo visual-inertial SLAM system in [17]. ROVIO is a VIO system implemented by EKF based on sparse image patches, which directly uses the pixel intensity error of image patches

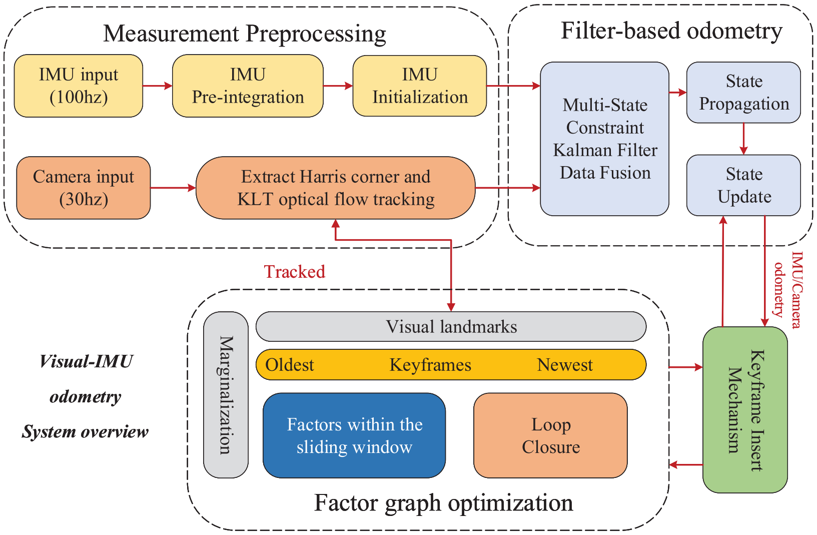

Measurement Preprocessing
IMU input (100hz)
IMU Pre-integration
IMU Initialization
Camera input (30hz)
Extract Harris corner and KLT optical flow tracking
Filter-based odometry
Multi-State Constraint Kalman Filter Data Fusion
State Propagation
State Update
IMU/Camera odometry
Tracked
Visual-IMU odometry System overview
Marginalization
Visual landmarks
Oldest
Keyframes
Newest
Factors within the sliding window
Loop Closure
Keyframe Insert Mechanism
Factor graph optimization

Fig. 1. Overview of proposed method. The entire system has three main parts: measurement preprocessing, filter-based odometry and factor graph optimization. The interconnected relationship of each part is represented by a red line with an arrow.

as a visual measure for updating the EKF; wheras [18] directly provides an EKF with photometric errors with high accuracy and robustness. Although filter-based VIO systems run faster, their accuracy is limited due to the lack of long-term data correlation.

Compared with filter-based methods, optimization-based methods can optimize multiple variables simultaneously and exhibit advantageous time synchronization. OKVIS [7] is a representative optimization-based VIO system that introduces a keyframe-based optimization method to correlate past poses with recent inertial states and inertial measurements for nonlinear optimization as well as precision: 6-degrees-of-freedom (6DOF) state estimation. To ensure that the system has better real-time performance yet ensure high accuracy, VINS_Mono, which uses the Lucas-Kanade (LK) optical flow method for feature tracking, uses graph optimization and optimizes variables on a window of limited size. Its loop closure detection applies DBoW2 and 4-DOF pose graph optimization, Thus, the system has high accuracy and robustness. The VINS_Mono system is simplified in VINS_Fusion [19] and has been extended to stereo and stereo-inertial odometry systems. In particular, HybVIO [33] extended PIVO [39] to stereo, and improved the IMU deviation model, outlier detection, stationarity detection and feature track selection, and to improve the long-term consistency of the system, based on the improved VIO system, optionally loosely couple with the ORB-SLAM2 system by running in parallel, and the VIO fixed-delay pose is used as the input of the ORB-SLAM2 system, and use the conversion relationship between the VIO system and the SLAM system to output the SLAM-processed pose on the input frame. This paper realized the combination of multi-state constraint Kalman filter method and factor graph optimization method from one system, according to the set keyframe mechanism, performs multi-state constraint Kalman filter and factor factor separately on different levels of frames.

A fast, accurate IMU initialization method is the premise of building a high-precision and robust VIO system, and is the core part of the visual-inertial system. Although [20] reused maps with short-, medium-, and long-term data associations, the slow IMU initialization technique substantially affects the accuracy and robustness of the system. For improving the rate of IMU initialization, VINS_Mono uses the initialization method proposed in [21], assuming that the external parameters of the IMU and the camera are known—ignoring the influence of acceleration bias; and estimating the velocity, gravity and scale separately to improve the efficiency. The IMU initialization method proposed in [9] (used in this paper) establishes the estimation of inertial parameters as an optimal estimation problem, considers IMU noise, simultaneously estimates all of the inertial parameters, avoids the problem of data inconsistency, does not ignore any bias, and uses it as the MAP probability a priori known information for IMU initialization.

III. System Overview

Fig. 1 shows the frame structure of our proposed visual-inertial state estimator system. The system consisted of four module, measurement preprocessing, filter-based odometry factor graph optimization and loop closure detection.

The system started with preprocessing of the measurement data, extraction and tracking image features, pre-integration of IMU measurements between two consecutive keyframes, and performing IMU initialization. See Section IV for details.

Then, the measurement data of the camera and IMU were sent to the odometer module of MSCKF for pose estimation. To further improve the visual measurements, we optimized the filtered pose with a sliding-window-based factor graph. See Section V for details.

IV. Preprocessing of IMU Measurements

This section introduces two parts: IMU pre-integration and initialization. First, we used the PK4 for IMU pre-integration of IMU measurements, and calculated the noise covariance matrix to propagate the measurements. Then, we built a MAP estimation model, the IMU measurement residual, and an a priori residual model to optimize the IMU parameters.

A. IMU Pre-Integration

To provide accurate initial values for initialization and IMU constraints for back-end optimization, we used the PK4 pre-integration method to perform IMU pre-integration; and propagated the bias noise.

1) Accelerometer and gyroscope measurements: IMUs can measure the body acceleration and gyroscope at every moment; the measurement model was as follows:

$$\hat{a}_t = a_t + b_{a_t} + R_w g^w + n_a, \hat{\omega}_t = \omega_t + b_{\omega_t} + n_\omega, \tag{1}$$

where $\hat{a}$ and a are the estimation of acceleration measurements and the acceleration true value, respectively; $\hat{\omega}$ and ω represent the estimation of gyroscope measurements and the gyroscope true value respectively. Estimation of the acceleration and gyroscope measurements is affected by the acceleration bias $b_a \in R^3$, gyroscope bias $b_\omega \in R^3$ and Gaussian white noise for the accelerometer and gyroscope: n_a and n_ω, respectively. The acceleration bias and gyroscope bias followed the following random walk model:

$$b_{a_{t+1}} = b_{a_t} + n_a, b_{\omega_{t+1}} = b_{\omega_t} + n_\omega. \tag{2}$$

We obtained the acceleration and gyroscope bias at each moment by adding Gaussian white noise to the deviation of the previous sampling moment.

2) Pre-integration: The position, velocity, and rotation state of two consecutive keyframes can be propagated by the measurements between them:

$$\begin{aligned} p^w_{b_j} &= p^w_{b_i} + v^w_{b_i}\Delta t_i - \frac{1}{2}g^w \Delta t_i^2 + \iint_{t\in[i,j]} \left(R^w_t a_t\right)dt^2, \\ v^w_{b_j} &= v^w_{b_i} - g^w \Delta t_i + \int_{t\in[i,j]} \left(R^w_t a_t\right)dt, \\ q^w_{b_i} &= q^w_{b_i} \otimes \int_{t\in[i,j]} \frac{1}{2}\Omega\left(\omega_t\right) q^{b_i}_t dt, \end{aligned} \tag{3}$$

where $\Omega(\omega) = \begin{bmatrix} -\lfloor\omega\rfloor_\times & \omega \\ -\omega^T & 0 \end{bmatrix}$, $\lfloor\omega\rfloor_\times = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}$, and Δt is the time interval between continuous time i and j.

From Eq. (3), the values of p, v, and q under the b_i frame and the b_j frame in the world coordinate system are evident. When performing state propagation from frame b_i to frame b_j, the pre-integration model must be used to convert the state in the world coordinate system to the IMU coordinate, as follows:

$$\begin{aligned} R^{b_i}_w p^w_{b_j} &= R^{b_i}_w (p^w_{b_i} + v^w_{b_i}\Delta t_i - \frac{1}{2}g^w \Delta t^2) + \alpha^{b_i}_{b_j}, \\ R^{b_i}_w v^w_{b_j} &= R^{b_i}_w (v^w_{b_i} - g^w \Delta t) + \beta^{b_i}_{b_j}, \\ q^{b_i}_w \otimes q^w_{b_j} &= \gamma^{b_i}_{b_j}, \end{aligned} \tag{4}$$

where $R^{b_i}_w$ and $q^{b_i}_w$ are the rotation matrix and rotation quaternion, respectively, from the world coordinate system to the IMU coordinate system. $\alpha^{b_i}_{b_j}$, $\beta^{b_i}_{b_j}$,and $\gamma^{b_i}_{b_j}$ are the pre-integration values corresponding to p, v and q, respectively. For the process of IMU pre-integration using PK4 numerical integration, see [38, Section 3.2].

3) Noise propagation: Next, we assumed that the IMU was synchronized with the camera, and calculated the solution as well as propagation of the pre-integration noise term between consecutive two keyframes at times $k = i$ and $k = j$. First, According to the pre-integration term between two consecutive keyframes, we define the relative motion increment between time i and j, as follows:

$$\begin{aligned} \Delta R\left(\gamma_{ij}\right) &= \prod_{k=i}^{j-1} Exp\left[\left(\hat{\omega}_{ik} - b_{\omega_k} - n_\omega\right)\Delta t_{ik}\right], \\ \Delta\beta_{ij} &= \sum_{k=i}^{j-1} \Delta R\left(\gamma_{ik}\right)\left(\hat{a}_{ik} - b_{a_k} - n_a\right)\Delta t_{ik}, \\ \Delta\alpha_{ij} &= \sum_{i}^{j-1} [\Delta\beta_{ik}\Delta t_{ik} + \frac{1}{2}R\left(\gamma_k\right)\left(\hat{a}_{ik} - b_{a_k} - n_a\right)\Delta t^2_{ik}], \end{aligned} \tag{5}$$

where $\hat{\omega}_{ik} = \widetilde{\omega}_{ik} + b_{\omega_k} + n_\omega$ and $\hat{a}_{ik} = \widetilde{a}_{ik} + b_{a_k} + n_a$ are the estimated value ω_{ik} and a_{ik}, respectively, after approximation by the PK4 numerical integration.

Using the estimated value (equal to the true value), we added the error term, from Eq. (5) to obtain noise terms $\delta\alpha_{ij}$,$\delta\beta_{ij}$ and $\delta r\left(\gamma_{ij}\right)$; corresponding to α_{ij}, β_{ij}, and $R\left(\gamma_{ij}\right)$, respectively, as follows:

$$\begin{aligned} \Delta\hat{R}\left(\gamma_{ij}\right) &= \Delta R\left(\gamma_{ij}\right) Exp\left(\delta r\left(\gamma_{ij}\right)\right), \\ \Delta\hat{\beta}_{ij} &= \Delta\beta_{ij} + \delta\beta_{ij}, \\ \Delta\hat{\alpha}_{ij} &= \Delta\alpha_{ij} + \delta\alpha_{ij}, \end{aligned} \tag{6}$$

For detailed calculations of the noise terms $\delta\alpha_{ij}$, $\delta\beta_{ij}$, and $\delta r\left(\gamma_{ij}\right)$, see Appendix A.

We decompose the noise terms $\delta\alpha_{ij}$, $\delta\beta_{ij}$, and $\delta r\left(\gamma_{ij}\right)$ in Appendix A. In accordance with the results of the decomposition, we can derive the continuous-time propagation equation for the noise terms, as follows:

$$\delta z_{ij} = F_{j-1}\delta z_{ij-1} + G_{j-1}n_{j-1}, \tag{7}$$

where $\delta z_{ij} = \left[\delta r\left(\gamma_{ij}\right)\ \delta\beta_{ij}\ \delta\alpha_{ij}\right]^T$; $n_{j-1} = \left[n_\omega\ n_a\right]^T$; and F_{j-1} and G_{j-1} are the Jacobian matrices of δz_{ij} and n_{j-1}, respectively. One can obtain detailed expressions in accordance with Eqs. (31), (32) and (33) in Appendix A.

Based on the continuous-time propagation Eq. (7) and the continuous-time noise covariance matrix $\sum_{n_{j-1}} \in R^{6\times 6}$, the covariance $\sum_{ij}$ propagates from the initial convariance $\sum_{ij} = 0$, as follows:

$$\sum_{ij} = F_{j-1}\sum_{ij}F^T_{j-1} + G_{j-1}\sum_{n_{j-1}}G^T_{j-1}, \tag{8}$$

where $\sum_{n_{j-1}} = diag\left(\sigma^2_\omega, \sigma^2_a\right)$.

In particular, the purpose of obtaining $\sum_{ij}$ is that when one obtains a new IMU measurement, only Eq. (7) and (8) are updated, rather than recomputation from scratch, which substantially reduces the computational complexity.

B. IMU Initialization

The IMU initialization builds an optimization model factor, optimizing the IMU pre-integration terms and parameters with factor graph, and uses the optimized pre-integration terms and parameters to repropagate the IMU pre-integration in a manner that updates the pose of the IMU. We adopt the IMU initialization model building method proposed in [9], build a MAP estimation model, and use all of the IMU noises as known prior information to perform the optimal estimation of the IMU pre-integration terms and parameters. The IMU parameters to be optimized are as follows:

$$Y_k = \{s, g^w_{dir}, b_a, b_\omega, \hat{v}_{0:k}\}, \tag{9}$$

where $s \in R^+$, $g^w_{dir} \in SO(3)$ and $\hat{v}_{0:k} \in R^3$ represent the scale factor in pure vision, gravity direction, and the unscaled speed of each frame in the body coordinate system, respectively. The true speed is expressed as $v_i = s\hat{v}_i$, but we used an unscaled speed $\hat{v}_i$ for simplifying the initialization.

We denoted the pre-integration of measurements between the ith and jth keyframes by M_{ij}, and we set pre-integration between consecutive keyframes in the IMU initialization window by $M_{0:k} = M_{0:1} \ldots M_{k-1:k}$.

In accordance with the defined IMU state and measurements, we formulated a MAP problem, but given that the measurements are independent of each other, maximizing the posterior distribution is equivalent to minimizing its negative logarithm. Therefore, the MAP estimate can be written:

$$\begin{aligned}(Y_k)^*_{\max} = \arg\min_{X_k}&(-\log(p(Y_k))\\ &-\sum_{i=1}^{k}\log(p(M_{i-1,i}\,|s\ , g^w_{dir}, b_a, b_\omega, v_{i-1}, v_i))).\end{aligned} \tag{10}$$

By given the measurements $M_{0:k}$ and measurement model in Eq. (5), and assuming that the noise follows a Gaussian distribution, For the calculation process of the measurements residual, see [9, Sec II-B].

With the known inertial parameters from Eq. (7), we must calculate its residual as the prior residual:

$$r_b = \left[\delta s, \delta g^w_{dir}, \delta b_a, \delta b_\omega, \delta\hat{\beta}_{i-1}, \delta\hat{\beta}_i\right], \tag{11}$$

where δb_a and δb_ω are immediately derived [22]; $\delta\hat{\beta}_{i-1}$ and $\delta\hat{\beta}_i$ are found in Appendix A; $\delta g^w_{dir} = \left(\delta\alpha_g, \delta\beta_g\right)$; and δs is derived from [9].

Assuming that IMU pre-integration follows the Gaussian distribution, the MAP problem can be equivalent to the following optimization problem:

$$(Y_k)^*_{\max} = \arg\min_{X_k}(\sum_{i=1}^{k}\|r_{M_{i-1,i}}\|^2_{\sum_{M_{i-1,i}}} + \|r_b\|^2_{\sum_b}), \tag{12}$$

where $r_{M_{i-1,i}}$ and r_b are the IMU measurement residuals and prior residuals between consecutive keyframes, and $\sum_{M_{i-1,i}}$ and $\sum_b$ are covariance matrices of the corresponding residuals. See [22] for the detailed definition of the covariance $\sum_{M_{i-1,i}}$.

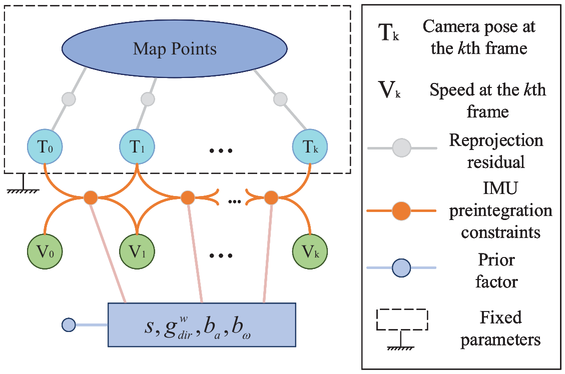

Map Points
T₀
T₁
...
Tₖ
V₀
V₁
...
Vₖ
$s, g^w_{dir}, b_a, b_\omega$
Tₖ Camera pose at the kth frame
Vₖ Speed at the kth frame
Reprojection residual
IMU preintegration constraints
Prior factor
Fixed parameters

Fig. 2. Diagram of the inertial optimization.

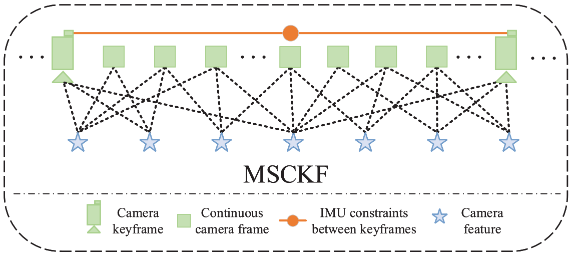


Fig. 3. Filter-based odometer and factor graph optimization process: We used MSCKF to fuse IMU and visual measurements between two consecutive frames. Simultaneously, we inserted keyframes, and utilized IMU constraints and visual reprojection errors between keyframes within a sliding window for factor graph optimization.

During optimization, the direction and scale factor of gravity are defined in [9, Sec II-B]. Fig. 2 shows the factor diagram of the inertial optimization. Only IMU residuals are included in the Fig. 2; the visual re-projection error is regarded as a constant; and the scale factor, gravity direction, acceleration bias, and angular velocity bias are regarded as the probability priors of the IMU residuals. Thus, only inertial optimization is completed; then we update the bias and repeat IMU pre-integration to propagate the IMU state.

V. Filter-Based Odometer And Factor Graph Optimization

In this section we describe a filter-based odometry and factor graph optimization method (Fig. 3). First, we insert keyframes based on the original frame through the set keyframe selection mechanism. Then, we fused each frame of the vision and IMU measurements between two keyframes with MSCKF and updated the pose of each frame. To reduce the computation, we replaced the EKF state vector of the camera pose with feature point information, used the QR orthogonal decomposition to decompose the H^c matrix in the state update stage, and converted the Kalman gain form, transforming the inversion of the measurements into the inversion of the states, equivalently. Since the state of the EKF at each moment is only related to the previous moment and lacks the constraint relationship with the historical state, we performed factor graph optimization at the keyframes level; constructed the visual reprojection error, IMU residuals and IMU prior factor for all of the information; added the loop-closure factor to the factor graph; and used the factor graph method to optimize them simultaneously within the sliding window. When the information of a new keyframe arrived, keyframes were optimized again within sliding window, and the linear errors were reduced after multiple iterations. This not only ensured tight coupling between the IMU and vision measurement, but also ensured the real-time performance and high precision of the system.

A. Insert Keyframes

We set up the keyframe mechanism as follows: we set up two situations to judge whether it is a keyframe. Considering the real-time and accuracy of our method, we set the current fame as a keyframe when it exceeded 40 image frames from the previous keyframe or the map point tracked by the current frame contained <40% of the map points of the previous keyframe. This keyframe setting method reduces the computational load of the system when optimizing keyframes using factor graphs. By inserting keyframes, we use factor graph optimization to optimize each keyframe pose at the keyframe level, and use MSCKF to fuse vision and IMU data and propagate and update each frame state at common frames level between two keyframes.

B. Filter-Based Odometer

1) IMU status: We updated and propagated the IMU status after IMU initialization. IMU status can be defined as follows:

$$X_{IMU} = \left[\,{}^I_G p^T \ {}^G v_I{}^T \ {}^I_G R^T \ b_a^T \ b_\omega^T\,\right]^T, \tag{13}$$

where ${}^I_G p^T$ and ${}^I_G R^T$ are translation and rotation, respectively, from the IMU coordinate system to the world coordinate system; ${}^G v_I{}^T$ is the IMU speed in the world coordinate system; and b_a^T and b_ω^T are the accelerometer and gyroscope bias, respectively. In accordance with Eq. (13), the error state of the IMU is defined as follows:

$$\tilde{X}_{IMU} = \left[\,{}^I_G \tilde{p}^T \ {}^G \tilde{v}_I^T \ {}^I_G \delta\theta^T \ \tilde{b}_a^T \ \tilde{b}_\omega^T\,\right]^T, \tag{14}$$

where ${}^I_G\delta\theta = Log\left({}^I_G\bar{R}^T \ {}^I_G R\right)$ is the pose error, and the rest is the standard additive error (i.e., the error is defined as $\tilde{x} = x - x$).

Here, the advantage of the pose error represented by $\delta\theta$ is that it enables us to represent the uncertainty of the pose with the covariance matrix, and the pose has only three degrees of freedom. Thus, we used the minimal representation of $\delta\theta$.

2) EKF state vector: With in a sliding window, to reduce the computational complexity, we only considered the camera states without considering the feature points information. Assuming that there are M camera states $\left[{}^C_G q^T \ {}^C_G p^T\right]$ in state space at time k, then the state vector is defined as follows:

$$X_k = \left[\, X^T_{IMU_k} \ {}^{C_1}_G q^T \ {}^{C_1}_G p^T \cdots {}^{C_M}_G q^T \ {}^{C_M}_G p^T \,\right]^T, \tag{15}$$

where X_{IMU_k} is the IMU status; ${}^{C_i}_G q^T$ and ${}^{C_i}_G p^T$, $i = 1 \ldots N$ are the estimation of the ith camera rotation and translation, respectively, to the world coordinate system. Its error state vector is defined as follows:

$$\tilde{X}_k = \left[\, \tilde{X}^T_{IMU_k} \ {}^{C_1}_G \tilde{q}^T \ {}^{C_1}_G \tilde{q}^T \cdots {}^{C_M}_G \tilde{q}^T \ {}^{C_M}_G \tilde{p}^T \,\right]^T. \tag{16}$$

3) IMU state propagation: In accordance with $\tilde{x} = \hat{x} - x$, the propagation of the IMU state is the propagation of the IMU error state. The discrete IMU error state propagation process is as follows:

$$\begin{aligned} \tilde{X}_{IMU_{i+1}} &= F_x \tilde{X}_{IMU_i} + F_n n_{IMU}, \\ n_{IMU} &= \left[n_\omega, n_a, n_{b_\omega}, n_{b_a}\right], \end{aligned} \tag{17}$$

where $\dot{b}_\omega = n_{b_\omega}$, $\dot{b}_a = n_{b_a}$, and n_{IMU} is the Gaussian white noise of the system. F_x and F_n are the discrete time IMU error- and noise-state transition matrices (the Jacobian of continuous-time IMU model in [5] with respect to the IMU state and noise), respectively.

4) Covariance matrix propagation: We denote the covariance matrices of the IMU error state $\tilde{X}_{IMU}$ and noise n_{IMU} as $P_{I_{i|i}}$ and Q, respectively; the covariance $P_{I_{i|i}}$ propagation process of the error state is as follows:

$$P_{I_{i+1|i}} = \Phi_{i+1,i} P_{I_{i|i}} \Phi^T_{i+1,i} + F_n Q F_n^T, \tag{18}$$

where
$Q = diag\left[\,\delta t\sigma_\omega^2 I_3 \ \delta t\sigma_a^2 I_3 \ \delta t\sigma_{b_\omega}^2 I_3 \ \delta t\sigma_{b_a}^2 I_3\,\right]$.
where $i \in \left[t_k, t_{k+1}\right]$ and $\Phi_{i+1,i} = exp(F_x\delta t) \simeq I + F_x\delta t$. When the state continues to propagate to a new scan t_{k+1}, we define the EKF error-state covariance matrix at this time as P_{k+1}; the propagated covariance matrix at time-step t_{k+1} is given by:

$$P_{k+1|k} = \begin{bmatrix} P_{II_{k+1|k}} & \Phi_{k+1,k} P_{IC_k} \\ P^T_{IC_k}\Phi^T_{k+1,k} & P_{II_{k|k}} \end{bmatrix}, \tag{19}$$

where $P_{II_{k+1|k}}$ can be recursively computed with Eq. (18).

5) Camera pose state augmentation: In accordance with the external parameters between the calibrated camera and IMU as well as the IMU pose, we calculated the camera pose, as follows:

$$\begin{aligned} {}^C_G q &= {}^C_I q \otimes {}^I_G q, \\ {}^C_G p &= {}^I_G p + C\left({}^I_G q\right)^T {}^C_I p, \end{aligned} \tag{20}$$

where ${}^C_I q$ and ${}^C_I p$ are the quaternion and position between the IMU and camera, respectively. $C(\cdot)$ denotes a rotational matrix. We appended this camera pose estimation to the state vector and augmented the covariance matrix accordingly:

$$P_{k|k} \leftarrow \begin{bmatrix} I_{6N+15} \\ J_{\pi_N} \end{bmatrix} P_{k|k} \begin{bmatrix} I_{6N+15} \\ J_{\pi_N} \end{bmatrix}^T, \tag{21}$$

where Appendix B shows the calculation process for J_{π_N}.

6) Camera observation model: We used the KLT optical flow method to track FAST corner features ${}^C p_i \in \Gamma_k$ extracted from the kth frame image; i is the corner index, and its corresponding landmark point in three-dimensional space is ${}^G P_i$. To estimate the state of the current frame, we constructed

this measurement model by using a reprojection error between the tracked feature points and visual landmarks:

$$r_c\left(X_k, {}^Cp_i, {}^GP_i\right) = {}^Cp_i - \left[f_x \cdot \frac{{}^CP_{i_x}}{{}^CP_{i_z}} + c_x \cdot f_y \cdot \frac{{}^CP_{i_y}}{{}^CP_{i_z}} + c_y\right]^T, \tag{22}$$

where f_x and f_y are focal lengths, and c_x and c_y are offsets of the main points on the image plane, ${}^CP_i = \left(C\left({}^I_Gq_k\right) \cdot C\left({}^C_Iq_k\right)\right)^T {}^GP_i - \left(C\left({}^C_Iq_k\right)\right)^T {}^I_Gp_k - {}^C_Ip_k$.

Considering that measurements ${}^Cp_i$ and ${}^GP_i$ are both affected by noise, we assumed that their true values are ${}^Cp_i^{tv}$ and ${}^GP_i^{tv}$, respectively, as follows:

$${}^Cp_i = {}^Cp_i^{tv} + n_{p_i}, {}^GP_i = {}^GP_i^{tv} + n_{P_i}, \tag{23}$$

where both n_{p_i} and n_{P_i} follow a standard normal distribution; $n_{p_i} \sim N\left(0, \sum_{n_{p_i}}\right)$ and $n_{P_i} \sim N\left(0, \sum_{n_{P_i}}\right)$, respectively. In accordance with the ground-truth corner features, we obtained a true zero-residual model:

$$\begin{aligned} 0 &= r_c\left(\hat{X}_k, {}^Cp_s^{tv}, {}^Gp_s^{tv}\right) \\ &\approx r_c\left(X_k, {}^Cp_i, {}^GP_i\right) + H_i^c\tilde{X}_k + N_i, \end{aligned} \tag{24}$$

where the reader can refer to Appendix C for H_i^c and N_i.

7) Status updates: To update the state, we must solve the error state $\Delta X = Kr_n$; where K is Kalman gain, and r_n is the residual that ignores only the noise $Q_2{}^T r_c$. The calculation process is as follows:

$$r_n = Q_1{}^T r_c = H\tilde{X} + n, \tag{25}$$

where $n = Q_1{}^T N$, H is the upper triangular matrix of $H^c = \left[H_1^c, \ldots, H_m^c\right]$. To reduce the amount of calculations, we performed QR decomposition on H^C; the decomposition process is as follows:

$$H^c = \left[\,Q_1 \;\; Q_2\,\right]\begin{bmatrix} H \\ 0 \end{bmatrix},$$

where Q_1 and Q_2 are orthogonal matrices.

Considering that the raw Kalman gain $K = PH^T\left(HPH^T + R\right)^{-1}$ must invert the matrix in the dimension of the measurements, to reduce the amount of calculations, we convert the Kalman gain equivalently:

$$K = \left(H^TR^{-1}H + P^{-1}\right)^{-1}H^TR^{-1}, \tag{26}$$

where $R = diag\left(\sum_{N_1}, \cdots, \sum_{N_m}\right)$, this Kalman gain indicates that inversion of the matrix in the dimension of state substantially reduces the computation complexity. Then, we updated the state estimate and covariance matrix as follows:

$$\begin{aligned} X_k &= X_k + KH\tilde{X} + Kn, \\ P_{k+1} &= (I - KH)\,P_k(I - KH)^T + KRK^T. \end{aligned} \tag{27}$$

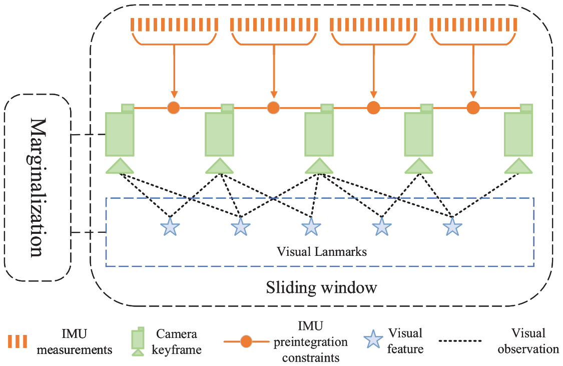


Fig. 4. Illustration of factor graph optimizing keyframe poses within a sliding window. There are several camera poses, visual features, IMU measurements, and IMU pre-integration constraints in the sliding window. IMU constraints and visual reprojection errors are optimization factors in factor graphs for optimizing keyframe poses.

C. Sliding Window-Based Factor Graph Optimization

We used the MSCKF to fuse the visual and IMU measurements, as well as update the poses, which still had a large cumulative error and linearization error. To further improve the accuracy of each frame pose, we inserted keyframes on the basis of the original frames, and performed MSCKF for each frame between two keyframes and sliding-window-based factor graph optimization for keyframes; which considered the IMU prior, visual constraints and IMU constraints between all of the keyframes in the sliding window in a manner that optimized the poses of the keyframes [26]. Considering all of the historical information for comprehensive optimization can reduce the cumulative error, and every time new data arrives, all of the historical data in the sliding window should be optimized once. Multiple iterative optimization not only reduces the linearization error but also can better approach the optimal solution. Fig. 4 shows the optimization process.

Our factor graph optimization method fixed the visual landmarks and optimized only the keyframe pose, and used a sliding window method to update keyframes and visual landmarks. Upon arrival of a new image frame, the old frame must be marginalized. If the penultimate frame is a keyframe, we moved the last frame pose out of the sliding window, and simultaneously marginalized both the visual and inertial data. If the penultimate frame is not the keyframe, to ensure the coherence of IMU pre-integration, we marginalized only the visual observation of the last frame. We comprehensively considered the size of the sliding window by real-time and running accuracy of the system. We set the size of the sliding window to 20 keyframes. According to the setting method of keyframe and sliding window, we ensured high real-time performance when performing factor graph optimization.

D. Loop-Closure Detection

To further improve the positioning accuracy and avoid excessive drift when the robot reached the same position again, we used the DBoW2 bag-of-words model for loop-closure detection [3], [4], [27], as Fig. 5. We used corner features and the BRIEF descriptors to describe the entire image, and

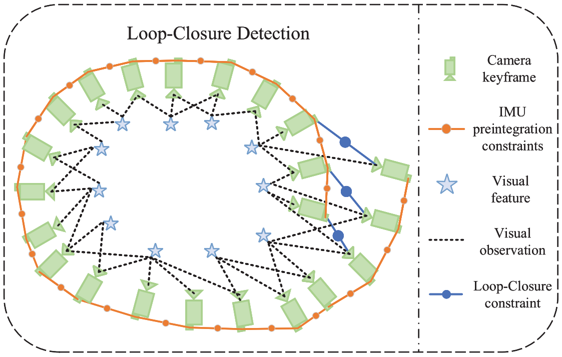


Fig. 5. Loop-closure constraint. The blue line and circle are loop-closure constraints, connecting the two keyframes that match the current moment and historical moment.

treated BRIEF descriptors as visual words for querying visual databases. After temporal and spatial consistency checks, one returns to the loop-closure detection candidate frame.

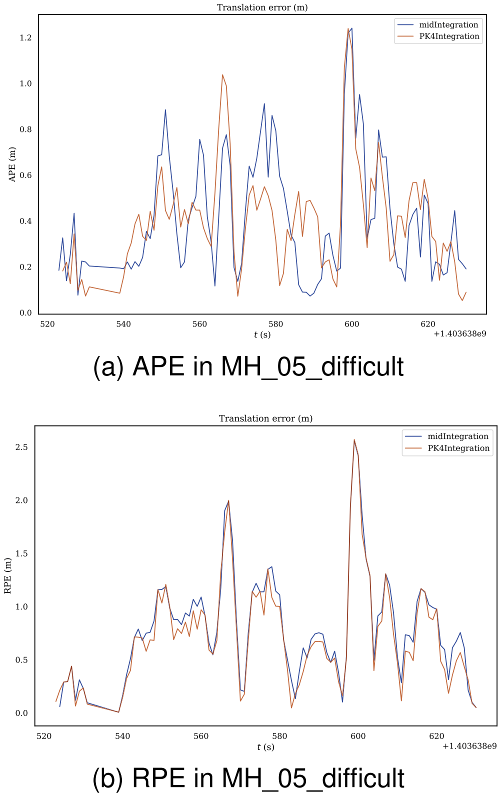
VI. Experiments

To qualitatively and quantitatively analyze our proposed method, we conducted a series of experiments. In the experiments, we used two datasets. One is the EuRoC MAV dataset [28], which has 11 subsequences. This dataset was collected using UAV in an indoor environment; each sequence has different light dark changes and motion intensity, which can sufficiently verify the robustness of our system. The other dataset is the KITTI dataset, which contains visual and IMU information. This dataset was collected using autonomous vehicles in an outdoors environment, which can be used to evaluate the performance of our algorithm outdoors. Finally, we put the proposed method on a physical robot for physical verification in light-stable indoor and light-unstable outdoor environments. We implemented our method based on C++ and executed it with an Intel i5-6500 CPU desktop computer by using the Ubuntu 16.04 system.

A. EuRoc MAV Dataset

We use the EuRoC MAV dataset to verify the robustness of our system in indoor environments. The EuRoC MAV dataset is collected by the Asctec Firefly hexagonal rotorcraft, which contains a stereo camera (Aptina MT9V034 global shutter, WVGA monochrome, 2CfB-20 FPS) and an IMU (ADIS16448, angular rate and acceleration, 200 Hz).

The IMU pre-integration is crucial to the VIO system, is the initial value of IMU initialization, and directly affects the accuracy of the pose solution. We use the PK4 numerical integration to approximate the IMU pre-integration and calculate the absolute pose error (APE) and relative pose error (RPE) to evaluate the performance of PK4 integration and mid-integration. Since the rotation error had little effect on the system, we only compare the root mean square error (RMSE) [29] of translation. Fig. 6 shows the translation component of the APE and the RPE comparison between the PK4 integration and mid-integration in the representative MH_05_difficult subsequence. The PK4 numerical integration slightly outperformed the mid-integration (Fig. 6). In addition, the mean value of the RMSE of PK4 numerical integration in APE and RPE in all of the EuRoC MAV dataset sequences was superior to the mid-integration by 0.015 m and 0.021 m. Thus, we used the PK4 numerical integration to conduct IMU pre-integration.



(a) APE in MH_05_difficult

(b) RPE in MH_05_difficult

Fig. 6. Comparison of error between mid integration and PK4 integration. (a) Error trajectory of the translation part of the APE; (b) Error trajectory of the translation part of the RPE.

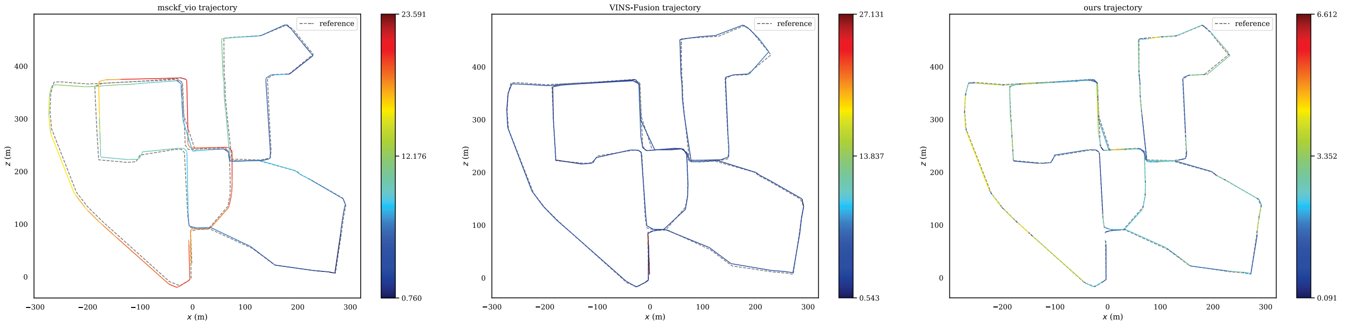
Our proposed VIO system that combined extended Kalman filtering and factor graphs. On the basis of using MSCKF to fuse visual and IMU measurements, as well as propagate and update the pose, we used the factor graph method to solve the problem that MSCKF is unable to associate the historical pose, and optimize the keyframe pose. In principle, the cumulative error of the system can be reduced by increasing the constraint relationship with the historical frame. We used the RMSE to evaluate the accuracy of the system. Table II summarizes the RMSE of all of the sequences in the EuRoC MAV datasets.

First, we used the VINS initialization method to initialize IMU and vision. The experimental results are expressed by our_mono (vins) and our_stereo (vins) in Table II. Then, we changed the IMU initialization model (see Section IV-B); the experimental results are shown as by our_mono and our_stereo in Table II. By comparing two IMU initialization methods, we chose the IMU initialization method with higher precision as the IMU initialization method of our system.

Next, we compared our system with state-of-the-art, filter-based VIO systems such as ROVIO [6], msckf_vio [30], and optimization-based VINS_Mono and VINS_Fusion by using the same computer configuration, running environment and

TABLE II
RMSE(M) IN EUROC MAV

	Method	MH01	MH02	MH03	MH04	MH05	V101	V102	V103	V201	V202	V203	Mean
Mono Inertial	ROVIO	0.308	0.440	0.411	0.892	1.270	0.227	**0.196**	0.191	0.292	0.583	0.200	0.772
	VINS-Mono	0.187	0.179	0.402	0.412	**0.403**	**0.140**	0.310	0.314	**0.121**	**0.295**	0.312	0.280
	our_mono(vins)	**0.180**	**0.172**	0.308	0.372	0.442	0.150	0.205	0.172	0.178	0.335	0.201	0.247
	our_mono	0.182	0.176	**0.262**	**0.340**	0.425	0.142	0.206	**0.159**	0.166	0.319	**0.171**	**0.232**
Stereo Inertial	msckf_vio	0.312	0.356	0.402	0.612	0.724	0.246	**0.187**	0.172	0.198	0.652	-	0.386
	VINS-Fusion	0.236	0.232	0.420	0.399	0.382	**0.146**	0.303	0.276	**0.126**	**0.279**	0.288	0.281
	our_stereo(vins)	**0.210**	0.221	0.301	0.368	0.362	0.157	0.200	0.192	0.150	0.330	0.198	0.244
	our_stereo	0.225	**0.204**	**0.286**	**0.341**	**0.356**	0.149	0.214	**0.149**	0.160	0.312	**0.146**	**0.231**



(a) msckf vio (b) VINS-Fusion (c) ours

Fig. 7. Comparison of trajectories under KITTI 2011_10_03_drive_0027 raw dataset. (a), (b) and (c) represent the comparison between the running trajectory of msckf_vio, VINS_Fusion and ours and the groundtruth trajectory (reference), respectively. The three numbers on the right side of each figure represent the maximum, median and minimum of the APE from top to bottom.

evaluation algorithm. Compared with the filter-based system, in monocular-inertial and stereo-inertial configuration, our method was improved by 2.3CfB- and 0.7CfB-, respectively. The advantage of our method is consideration of all of the historical information and prior information, conduction of multiple iterations in the sliding windows, and introducing global optimization (which contributes to reducing the linearization and cumulative errors). Compared with a factor graph-based VIO system, in monocular-inertial and stereo-inertial configuration, our method was improved by 15.4% and 17.4%, respectively. Thus, our method is superior to the factor graph-based method in the terms of data fusion, and can better use IMU measurements to reduce the cumulative drift of the system.

B. KITTI Dataset

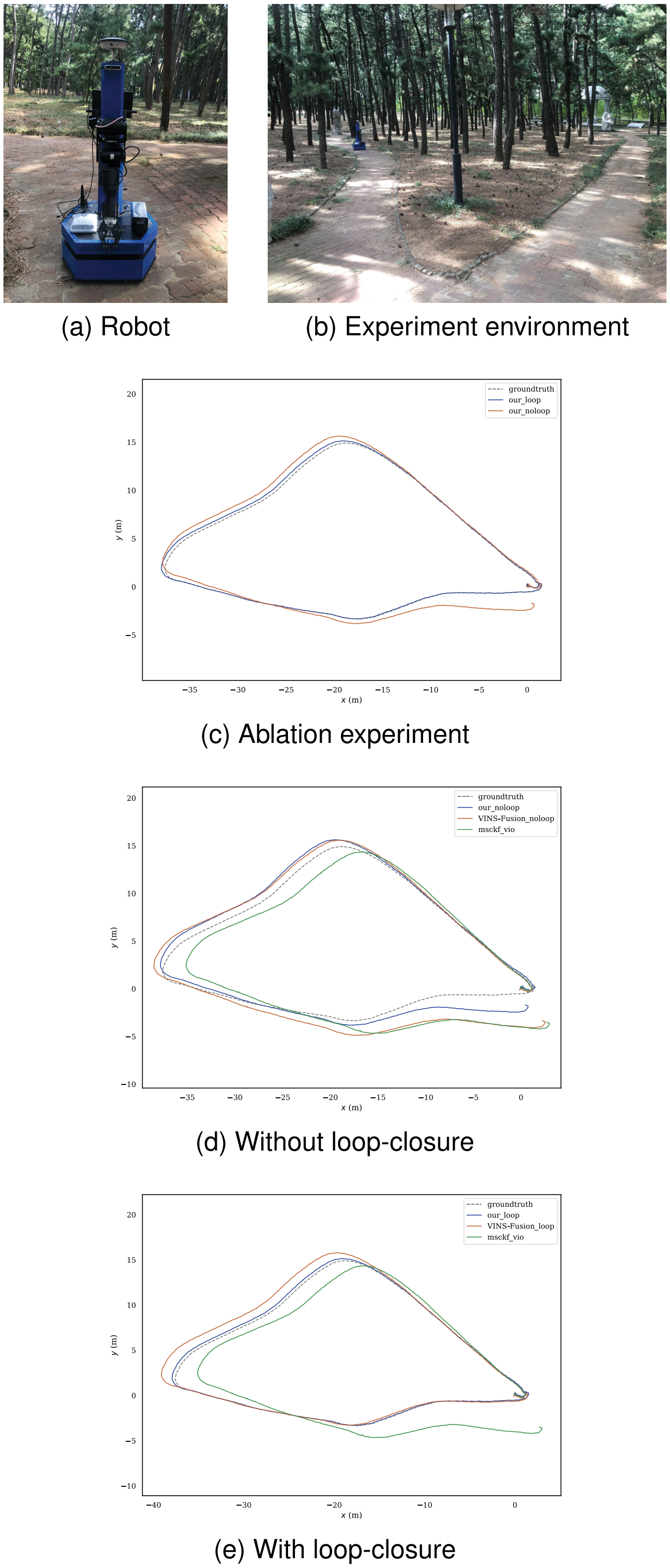
To further verify the performance of our system, we used the KITTI 2011_10_03_drive_0027 raw dataset (which involves a 10-Hz camera and 100-Hz IMU measurements, and was captured around the city of Karlsruhe, Germany). This dataset has a vehicle running in loops in outdoor real environments. We has a vehicle running in loops in outdoor real environments. We used this dataset to evaluate the performance of the system over long distances outdoors.

Fig. 7 and Table III are comparisons of the running results of the stereo-inertial system. Fig. 7 (a), (b) and (c) show running trajectories of msckf_vio, VINS_Fusion and ours compare with groundtruth and the distribution of errors, respectively. The running trajectory of our method is closer to the groundtruth trajectory. Table III shows the RMSE of all APEs and RPEs. In the table, Ate and Are represent the RMSE of translation and rotation parts of APE, respectively, whereas we calculate the RPE at intervals of 1m, Rte and Rre are the RMSE of translation and rotation parts of RPE, respectively. From Fig. 7 and Table III, our method outperformed state-of-the-art VIO systems msckf_vio and VINS_Fusion.

TABLE III
RMSE APE AND RPE METRICS

Method	Ate(m)	Are(deg)	Rte(m)	Rre(deg)	Time(ms)
msckf_vio	8.1259	0.1645	5.6501	0.1091	**15.0626**
VINS-Fusion	3.2517	0.1453	2.9473	0.0797	20.2278
our	**2.4021**	**0.9001**	**2.0104**	**0.0568**	16.6657

In addition, we also calculate the time consumed by pose processing and optimization. A total of 4544 frames of camera poses were processed and optimized during trajectory running, and the average processing time of each frame of camera poses is shown in Table III. The average processing time of each frame pose optimization of our algorithm is 16.6657ms, which is significantly superior to VINS_Fusion algorithm. This is because our method is much sparser than the VINS system in the selection of keyframes. At the same time, the real-time performance of the system is greatly improved and the computational complexity is reduced by introducing sliding windows, performing QR decomposition and converting the form of Kalman gain. Although our method takes slightly more time to calculate the pose optimization than the msckf_vio algorithm, the accuracy of our method is much higher than that of the



(a) Robot (b) Experiment environment

(c) Ablation experiment

(d) Without loop-closure

(e) With loop-closure

Fig. 8. Outdoor real experiment is performed using the Qiming mobile robot equipped with Realsense D435i and GNSS-850 in (a): (b) shows the outdoor experimental environment. In (c), (d) and (e), the gray dotted line represents the trajectory provided by GNSS-850, which is ground truth. (c) is ablation analysis, the blue solid line corresponds to our method with loop-closure, the red solid line corresponds to our method without loop-closure; (d) and (e) are comparative experiment without loop-closure and comparative experiment with loop-closure, respectively, the blue solid line corresponds to our method, the red solid line corresponds to VINS-Fusion, and the green solid line corresponds to msckf_vio.

TABLE IV
APE Translation Part Metric

	our_lp	our_nolp	VINS_lp	VINS_nolp	msckf_vio
RMSE(m)	**0.230**	0.801	0.572	1.128	1.319

msckf_vio algorithm. A comprehensive comparison of our method outperforms msckf_vio and VINS-Fusion.

C. Real-World Experiments

In the physical experiments, we migrated our proposed algorithm to the Qiming ROS mobile robot of the six workshops, which installed Intel's RealsenseD435i camera and GNSS-850, as shown in Fig. 8 (a). We use the trajectory obtained from GNSS-850 as the ground truth. In the outdoor environment shown in Fig. 8 (b), we set up ablation experiment to verify the necessity of increasing loop-closure detection, and set up comparison experiments to compare with VINS Fusion and msckf, as shown in Fig. 8 (c), (d) and (e), the total running length is 106.90 m, and the average running speed is 2.2 m/s. Table IV shows the trajectory accuracy, where our_lp, our_nolp, VINS_lp and VINS_nolp represents our method with loop-closure, our method without loop-closure, VINS-Fusion with loop-closure and VINS-Fusion without loop-closure, respectively.

From Fig. 8 (c), with the addition of loop-closure, the running trajectory of our method is obviously improved, and the accuracy is improved by 71.29% in Table IV. Therefore, it is necessary to increase the loop-closure. From Fig. 8 (d), Fig. 8 (e) and Table IV, compare our method with state-of-the-art VIO systems VINS-Fusion and msckf_vio. In case of loop-closure, compared with VINS-Fusion and msckf_vio, the RMSE of our method is reduced by 59.79% and 82.56%, respectively. But when loop-closure detection is not added, compared with VINS-Fusion and msckf_vio, the RMSE of our method is decreased by 28.99% and 39.27%, respectively. Thus, through these experiments, we verify the effectiveness of our method and the necessity of adding loop-closure detection.

VII. Conclusion and Future Work

We proposed a robust, general Kalman filter-based and factor graph visual-inertial systems. We fused vision and inertial measurements by using MSCKF, to achieve a high degree of coupling of sensor data. For the IMU measurements, we used the PK4 numerical integration method to complete the IMU pre-integration. To avoid calculating the IMU noise from scratch, we built an IMU noise propagation model (for updating the IMU pre-integration, when new IMU measurements arrive). In the IMU initialization phase, we used the MAP estimation model to optimize the IMU parameters, considering all of IMU noise as known prior information. After optimization we performed re-propagation of IMU pre-integration. We fused vision and IMU measurements using MSCKF, and propagated and updated each frame pose. In addition, to solve the problem of the large cumulative error in MSCKF, we used a factor graph to establish the constraint relationship between the current keyframe and historical keyframe, and optimized the pose of each keyframe, then used MSCKF to update each frame pose. To reduce the computational cost, we inserted keyframes in the factor graph optimization stage and constructed a sliding window. To further improve the accuracy of the system, we utilized a bag-of-words method for loop closure detection to reduce the cumulative drift. Our system achieved higher accuracy than existing public methods, as indicated by testing on datasets with aggressive motion and substantial lighting changes, as well as in indoor and outdoor environments.

We will continue, to verify the excellent scalability of our method. In future work, we will extend our method to fusion of some common sensors for autonomous vehicles, such as vision, IMU, LiDAR, and global sensors (e.g. GPS), to achieve robust large-scale outdoor pose estimation for autonomous vehicles. In follow-up work, we will also focus on removing dynamic and semi-static objects in the field.

Appendix

A. Calculate $\delta r(\gamma_{ij})$, $\delta\beta_{ij}$ and $\delta\alpha_{ij}$

In accordance with Eq. (5), we used mathematical derivations for decomposition, such that the left-hand side of the equation is evident in the form of an estimated value plus an error term. The decomposition process of $\Delta\mathrm{R}\left(\gamma_{ij}\right)$, $\Delta\beta_{ij}$ and $\Delta\alpha_{ij}$ is as follows:

$$\begin{aligned}\Delta R\left(\gamma_{ij}\right) &= \prod_{k=i}^{j-1} Exp\left[\left(\hat{\omega}_{ik} - b_{\omega_k} - n_\omega\right)\Delta t_{ik}\right] \\ &\simeq \prod_{i}^{j-1}\left[Exp\left(\left(\hat{\omega}_{ik} - b_{\omega_i}\right)\Delta t_{ik}\right) Exp\left(-J_r\left(\gamma_k\right) n_\omega \Delta t_{ik}\right)\right] \\ &= \Delta\hat{R}\left(\gamma_{ij}\right)\prod_{k=i}^{j-1} Exp\left(-\Delta\hat{R}\left(\gamma_{k+1j}\right) J_r\left(\gamma_k\right) n_\omega \Delta t_{ik}\right) \\ &= \Delta\hat{R}\left(\gamma_{ij}\right) Exp\left(-\delta r\left(\gamma_{ij}\right)\right),\end{aligned} \tag{28}$$

where $\delta r\left(\gamma_{ij}\right)$ is the error term.

We used the $\delta r\left(\gamma_{ij}\right)$ obtained with Eq. (28) into $\Delta\beta_{ij}$, and used the first-order approximation to remove the higher-order terms. The decomposition of $\Delta\beta_{ij}$ is as follows:

$$\begin{aligned}\Delta\beta_{ij} &= \sum_{k=i}^{j-1}\Delta R\left(\gamma_{ik}\right)\left(\hat{a}_{ik} - b_{a_k} - n_a\right)\Delta t_{ik} \\ &\simeq \sum_{k=i}^{j-1}\Delta\hat{R}\left(\gamma_{ik}\right)\left(I - \delta r(\gamma_{ik})^\wedge\right)\left(\hat{a}_{ik} - b_{a_i}\right)\Delta t_{ik} \\ &\quad - \Delta\hat{R}\left(\gamma_{ik}\right) n_a \Delta t_{ik} \\ &= \Delta\hat{\beta}_{ij} + \sum_{k=i}^{j-1}\left[\Delta\hat{R}\left(\gamma_{ik}\right)\left(\hat{a}_{ik} - b_{a_i}\right)^\wedge \delta r\left(\gamma_{ik}\right)\Delta t_{ik}\right. \\ &\quad \left. -\Delta\hat{R}\left(\gamma_{ik}\right) n_a \Delta t_{ik}\right] \\ &= \Delta\hat{\beta}_{ij} - \delta\beta_{ij},\end{aligned} \tag{29}$$

where $\delta\beta_{ij}$ is the error term.

Similarly to Eq. (29), we decomposed $\Delta\alpha_{ij}$:

$$\begin{aligned}\Delta\alpha_{ij} &= \sum_{i}^{j-1}[\Delta\beta_{ik}\Delta t_{ik} + \frac{1}{2}R\left(\gamma_k\right)\left(\hat{a}_{ik} - b_{a_k} - n_a\right)\Delta t_{ik}^2] \\ &\simeq \sum_{k=i}^{j-1}\left[\frac{1}{2}\hat{R}\left(\gamma_{ik}\right)\left(I - \delta r(\gamma_{ik})^\wedge\right)\left(\hat{a}_{ik} - b_{a_i}\right)\Delta t_{ik}^2\right. \\ &\quad \left. + \left(\Delta\beta_{ik} - \delta\beta_{ik}\right)\Delta t_{ik} - \frac{1}{2}\hat{R}\left(\gamma_{ik}\right) n_a \Delta t_{ik}^2\right] \\ &= \Delta\hat{\alpha}_{ij} + \sum_{k=i}^{j-1}\left[\frac{1}{2}\hat{R}\left(\gamma_{ik}\right)\left(\hat{a}_{ik} - b_{a_i}\right)^\wedge \delta r\left(\gamma_{ik}\right)\Delta t_{ik}^2\right. \\ &\quad \left. -\Delta\beta_{ik}\Delta t_{ik} - \frac{1}{2}\hat{R}\left(\gamma_{ik}\right) n_a \Delta t_{ik}^2\right] \\ &= \Delta\hat{\alpha}_{ij} - \delta\alpha_{ij},\end{aligned} \tag{30}$$

where, $\delta\alpha_{ij}$ is the error term.

To obtain the F_{j-1} and G_{j-1} matrices, we continued to iteratively decompose $\delta r\left(\gamma_{ij}\right)$, $\delta\beta_{ij}$, and $\delta\alpha_{ij}$. The decomposition process of $\delta r\left(\gamma_{ij}\right)$ is as follows:

$$\begin{aligned}\delta r\left(\gamma_{ij}\right) &= \sum_{k=i}^{j-1}\Delta\hat{R}^T\left(\gamma_{k+1j}\right) J_r\left(\gamma_k\right) n_\omega \Delta t_{ik} \\ &= \sum_{k=i}^{j-2}\Delta\hat{R}^T\left(\gamma_{k+1j}\right) J_r\left(\gamma_k\right) n_\omega \Delta t_{ik} \\ &\quad + \Delta\hat{R}^T\left(\gamma_{jj}\right) J_r\left(\gamma_{j-1}\right) n_\omega \Delta t_{ij-1} \\ &= \Delta\hat{R}^T\left(\gamma_{j-1j}\right)\sum_{k=i}^{j-2}\Delta\hat{R}^T\left(\gamma_{k+1j-1}\right) J_r\left(\gamma_k\right) n_\omega \Delta t_{ik} \\ &\quad + J_r\left(\gamma_{j-1}\right) n_\omega \Delta t_{ij-1} \\ &= \Delta\hat{R}^T\left(\gamma_{j-1j}\right)\delta r\left(\gamma_{ij-1}\right) + J_r\left(\gamma_{j-1}\right) n_\omega \Delta t_{ij-1},\end{aligned} \tag{31}$$

where $J_r\left(\gamma_k\right) = I - \frac{1-\cos(\|\gamma_k\|)}{\|\gamma_k\|^2}\gamma_k^\wedge + \frac{\|\gamma_k\|-\sin(\|\gamma_k\|)}{\|\gamma_k^3\|}\left(\gamma_k^\wedge\right)^2$.

We repeat the same process to decompose $\delta\beta_{ij}$.

$$\begin{aligned}\delta\beta_{ij} &= \sum_{k=i}^{j-1}\left[-\Delta\hat{R}\left(\gamma_{ik}\right)\left(\hat{a}_{ik} - b_{a_i}\right)^\wedge \delta r\left(\gamma_{ik}\right)\Delta t_{ik}\right. \\ &\quad \left. \Delta\hat{R}\left(\gamma_{ik}\right) n_a \Delta t_{ik}\right] \\ &= \sum_{k=i}^{j-2}\left[-\Delta\hat{R}\left(\gamma_{ik}\right)\left(\hat{a}_{ik} - b_{a_i}\right)^\wedge \delta r\left(\gamma_{ik}\right)\Delta t_{ik}\right. \\ &\quad \left. + \Delta\hat{R}\left(\gamma_{ik}\right) n_a \Delta t_{ik}\right] \\ &\quad - \Delta\hat{R}\left(\gamma_{ij-1}\right)\left(\hat{a}_{ij-1} - b_{a_i}\right)^\wedge \delta r\left(\gamma_{ij-1}\right)\Delta t_{ij-1} \\ &\quad + \Delta\hat{R}\left(\gamma_{ij-1}\right) n_a \Delta t_{ij-1} \\ &= \delta\beta_{ij-1} - \Delta\hat{R}\left(\gamma_{ij-1}\right)\left(\hat{a}_{ij-1} - b_{a_i}\right)^\wedge \delta r\left(\gamma_{ij-1}\right)\Delta t_{ij-1} \\ &\quad + \Delta\hat{R}\left(\gamma_{ij-1}\right) n_a \Delta t_{ij-1},\end{aligned} \tag{32}$$

Next, we calculated $\delta\alpha_{ij}$ by the same process.

$$\begin{aligned}
\delta\alpha_{ij} &= \sum_{k=i}^{j-1}\Bigg[-\frac{1}{2}\Delta\hat{R}\left(\gamma_{ik}\right)\left(\hat{a}_{ik}-b_{a_i}\right)^{\wedge}\delta r\left(\gamma_{ik}\right)\Delta t_{ik}^{2} \\
&\quad +\frac{1}{2}\Delta\hat{R}\left(\gamma_{ik}\right)n_a\Delta t_{ik}^{2}+\delta\beta_{ik}\Delta t_{ik}\Bigg] \\
&= \sum_{k=i}^{j-2}\Bigg[-\frac{1}{2}\Delta\hat{R}\left(\gamma_{ik}\right)\left(\hat{a}_{ik}-b_{a_i}\right)^{\wedge}\delta r\left(\gamma_{ik}\right)\Delta t_{ik}^{2} \\
&\quad +\frac{1}{2}\Delta\hat{R}\left(\gamma_{ik}\right)n_a\Delta t_{ik}^{2}+\delta\beta_{ik}\Delta t_{ik}\Bigg] \\
&\quad -\frac{1}{2}\Delta\hat{R}\left(\gamma_{ij-1}\right)\left(\hat{a}_{ij-1}-b_{a_i}\right)^{\wedge}\delta r\left(\gamma_{ij-1}\right)\Delta t_{ij-1}^{2} \\
&\quad +\frac{1}{2}\Delta\hat{R}\left(\gamma_{ij-1}\right)n_a\Delta t_{ij-1}^{2}+\delta\beta_{ij-1}\Delta t_{ij-1} \\
&= \delta\alpha_{ij-1}+\delta\beta_{ij-1}\Delta t_{ij-1} \\
&\quad -\frac{1}{2}\Delta\hat{R}\left(\gamma_{ij-1}\right)\left(\hat{a}_{ij-1}-b_{a_i}\right)^{\wedge}\delta r\left(\gamma_{ij-1}\right)\Delta t_{ij-1}^{2} \\
&\quad +\frac{1}{2}\Delta\hat{R}\left(\gamma_{ij-1}\right)n_a\Delta t_{ij-1}^{2}, \qquad (33)
\end{aligned}$$

By decomposing each term of $\delta r\left(\gamma_{ij}\right)$, $\delta\beta_{ij}$ and $\delta\alpha_{ij}$, we obtained F_{j-1} and G_{j-1} matrices for error propagation.

B. Calculate J_{π_N}

J_{π_N} in Eq. (21) is the derivation of the visual pose $\pi_N = \left[{}_{G}^{C}q\ {}_{G}^{C}p\right]^T$ to the error state vector $\widetilde{X}_{IMU}$:

$$\begin{aligned}
J_{\pi_N} &= \frac{\partial\tilde{\pi}_N}{\partial\tilde{X}} \\
&= \begin{bmatrix} \frac{\partial\delta\theta_{C_N}}{\partial\delta\theta_I}0_{3\times 9} & \frac{\partial\delta\theta_{C_N}}{\partial{}_{G}^{I}\tilde{p}}0_{3\times 6N} \\ \frac{\partial{}_{G}^{C_N}\tilde{p}}{\partial\delta\theta_I}0_{3\times 9} & \frac{\partial{}_{G}^{C_N}\tilde{p}}{\partial{}_{G}^{I}\tilde{p}}0_{3\times 6N} \end{bmatrix} \\
&= \begin{bmatrix} C\left({}_{I}^{C}q\right)0_{3\times 9}0_{3\times 3} & 0_{3\times 6N} \\ \left(C\left({}_{G}^{I}q\right)^T\cdot{}_{I}^{C}p\right) & 0_{3\times 9}I0_{3\times 6N} \end{bmatrix}. \qquad (34)
\end{aligned}$$

C. Calculate H_i^c and N_i

H_i^c is the covariance matrisx of the camera state. N_i is the covariance matrix of the noise, and follows the Gaussian distribution $N_i \sim \left(0, \sum_{N_i}\right)$. Take $\psi\left({}^{C}P_i\right) = \left[f_x\cdot\frac{{}^{C}P_{i_x}}{{}^{C}P_{i_z}}+c_x\cdot f_y\cdot\frac{{}^{C}P_{i_y}}{{}^{C}P_{i_z}}+c_y\right]^T$:

$$\begin{aligned}
H_i^c &= -\frac{\partial\psi\left({}^{C}P_i\right)}{\partial{}^{C}P_i}\cdot\frac{\partial{}^{C}P\left(\hat{X}_k,{}^{G}P_i\right)}{\partial\tilde{X}_k}, \\
\sum_{N_i} &= \sum_{n_{p_i}}+H_{P_i}\sum_{P_i}{H_{P_i}}^T, \\
H_{P_i} &= -\frac{\partial\psi\left({}^{C}P_i\right)}{\partial{}^{C}P_i}\cdot\frac{\partial{}^{C}P\left(X_k,{}^{G}P_i\right)}{\partial{}^{G}P_i}. \qquad (35)
\end{aligned}$$

Appendix E in [24] shows the detailed calculation processes for H_i^c and H_{P_i}.

References

[1] C. Forster, M. Pizzoli, and D. Scaramuzza, "SVO: Fast semi-direct monocular visual odometry," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2014, pp. 15–22.

[2] J. Engel, T. Schöps, and D. Cremers, "LSD-SLAM: Large-scale direct monocular SLAM," in *Proc. 13th Eur. Conf. Comput. Vis. (ECCV)*, Zurich, Switzerland. Springer, Sep. 2014, pp. 834–849.

[3] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM: A versatile and accurate monocular SLAM system," *IEEE Trans. Robot.*, vol. 31, no. 5, pp. 1147–1163, Oct. 2015.

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

[5] A. I. Mourikis and S. I. Roumeliotis, "A multi-state constraint Kalman filter for vision-aided inertial navigation," in *Proc. ICRA*, vol. 2, Apr. 2007, pp. 3565–3572.

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

[7] S. Leutenegger, P. Furgale, V. Rabaud, M. Chli, K. Konolige, and R. Siegwart, "Keyframe-based visual-inertial SLAM using nonlinear optimization," in *Proc. Robot., Sci. Syst. IX*, Jun. 2013, pp. 1–9.

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

[9] C. Campos, J. M. M. Montiel, and J. D. Tardós, "Inertial-only optimization for visual-inertial initialization," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2020, pp. 51–57.

[10] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, "MonoSLAM: Real-time single camera SLAM," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 6, pp. 1052–1067, Jun. 2007.

[11] G. Klein and D. Murray, "Parallel tracking and mapping for small AR workspaces," in *Proc. 6th IEEE ACM Int. Symp. Mixed Augmented Reality*, Nov. 2007, pp. 225–234.

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

[13] X. Gao, R. Wang, N. Demmel, and D. Cremers, "LDSO: Direct sparse odometry with loop closure," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2018, pp. 2198–2204.

[14] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 3, pp. 611–625, Mar. 2018.

[15] J. Zubizarreta, I. Aguinaga, and J. M. M. Montiel, "Direct sparse mapping," *IEEE Trans. Robot.*, vol. 36, no. 4, pp. 1363–1370, Aug. 2020.

[16] M. Li and A. I. Mourikis, "High-precision, consistent EKF-based visual-inertial odometry," *Int. J. Robot. Res.*, vol. 32, no. 6, pp. 690–711, 2013.

[17] 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. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 165–172.

[18] M. Bloesch, M. Burri, S. Omari, M. Hutter, and R. Siegwart, "Iterated extended Kalman filter based visual-inertial odometry using direct photometric feedback," *Int. J. Robot. Res.*, vol. 36, no. 10, pp. 1053–1072, 2017.

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

[20] R. Mur-Artal and J. D. Tardós, "Visual-inertial monocular SLAM with map reuse," *IEEE Robot. Autom. Lett.*, vol. 2, no. 2, pp. 796–803, Apr. 2017.

[21] T. Qin and S. Shen, "Robust initialization of monocular visual-inertial estimation on aerial robots," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2017, pp. 4225–4232.

[22] 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, Feb. 2017.

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

[24] J. Lin, C. Zheng, W. Xu, and F. Zhang, "R^{2}LIVE: A robust, real-time, LiDAR-inertial-visual tightly-coupled state estimator and mapping," *IEEE Robot. Autom. Lett.*, vol. 6, no. 4, pp. 7469–7476, Oct. 2021.

[25] W. Xu and F. Zhang, "FAST-LIO: A fast, robust LiDAR-inertial odometry package by tightly-coupled iterated Kalman filter," *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 3317–3324, Apr. 2021.

[26] T. Shan, B. Englot, D. Meyers, W. Wang, C. Ratti, and D. Rus, "LIO-SAM: Tightly-coupled LiDAR inertial odometry via smoothing and mapping," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2020, pp. 5135–5142.

[27] Y. Latif, C. Cadena, and J. Neira, "Robust loop closing over time for pose graph SLAM," *Int. J. Robot. Res.*, vol. 32, no. 14, pp. 1611–1626, Oct. 2013.

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

[29] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of RGB-D SLAM systems," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2012, pp. 573–580.

[30] K. Sun et al., "Robust stereo visual inertial odometry for fast autonomous flight," *IEEE Robot. Autom. Lett.*, vol. 3, no. 2, pp. 965–972, Apr. 2018.

[31] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 3354–3361.

[32] S. Heo, J. Cha, and C. G. Park, "EKF-based visual inertial navigation using sliding window nonlinear optimization," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 7, pp. 2470–2479, Jul. 2019.

[33] O. Seiskari, P. Rantalankila, J. Kannala, J. Ylilammi, E. Rahtu, and A. Solin, "HybVIO: Pushing the limits of real-time visual-inertial odometry," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2022, pp. 701–710.

[34] Y. Jiao, Y. Wang, X. Ding, M. Wang, and R. Xiong, "Deterministic optimality for robust vehicle localization using visual measurements," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 6, pp. 5397–5410, Jun. 2022.

[35] H. Yin, Y. Wang, X. Ding, L. Tang, S. Huang, and R. Xiong, "3D LiDAR-based global localization using Siamese neural network," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 4, pp. 1380–1392, Apr. 2020.

[36] Z. Huai and G. Huang, "Robocentric visual-inertial odometry," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Madrid, Spain, Oct. 2018, pp. 6319–6326.

[37] T. Sellers, T. Lei, C. Luo, G. E. Jan, and M. Junfeng, "A node selection algorithm to graph-based multi-waypoint optimization navigation and mapping," *Intell. Robot.*, vol. 2, no. 4, pp. 333–354, 2022.

[38] J. Cui, F. Zhang, D. Feng, C. Li, F. Li, and Q. Tian, "An improved SLAM based on RK-VIF: Vision and inertial information fusion via Runge–Kutta method," *Defence Technol.*, vol. 21, pp. 133–146, Oct. 2021.

[39] A. Solin, S. Cortes, E. Rahtu, and J. Kannala, "PIVO: Probabilistic inertial-visual odometry for occlusion-robust navigation," in *Proc. IEEE Winter Conf. Appl. Comput. Vis.*, Los Alamitos, CA, USA, Mar. 2018, pp. 616–625.

Zhiwei Wang is currently pursuing the master's degree with the School of Mechanical, Electrical and Information Engineering, Shandong University. His research interests include mobile robot navigation, machine learning, and neural networks.

Bao Pang received the M.S. degree in operations research and cybernetics from the University of Science and Technology Liaoning in 2014 and the Ph.D. degree in control theory and control engineering from Shandong University, Weihai, in 2020. He is currently a Lecturer with the School of Mechanical, Electrical and Information Engineering, Shandong University. His current research interests include robot control, robot path planning, deep reinforcement learning, and swarm intelligence.

Yong Song (Member, IEEE) received the B.S. degree in control science from Shandong University, Weihai, in 2001, and the M.S. and Ph.D. degrees in pattern recognition and intelligent system from Shandong University, in 2008 and 2012, respectively. He is currently a Professor with the School of Mechanical, Electrical and Information Engineering, Shandong University. His current research interests include robot control, machine learning, and swarm intelligence robotics.

Xianfeng Yuan received the Ph.D. degree in control theory and control engineering from Shandong University, China, in 2017. From 2016 to 2017, he was a Visiting Ph.D. student with Oklahoma State University, USA. He is currently an Associate Professor with the School of Mechanical, Electrical and Information Engineering, Shandong University, Weihai, China. His research interests include machine learning, intelligent fault diagnosis, and robotics.

Qingyang Xu received the M.S. and Ph.D. degrees in control theory and engineering from Dalian Maritime University, Dalian, China, in 2007 and 2010, respectively. From August 2010 to July 2012, he was a Post-Doctoral Researcher with the Dalian Institute of Chemical Physics, Chinese Academy of Sciences. He is currently an Associate Professor with the School of Mechanical, Electrical and Information Engineering, Shandong University, Weihai, China. His research interests include artificial intelligence, deep learning, and their applications on robot.

Yibin Li received the Ph.D. degree in pattern recognition and intelligent system from Tianjin University in 2008. He has published more than 100 contributions on robotics and control. His research interests include bionic robots, control of intelligent robots, machine learning, and intelligence architecture.