sensors

Article

Robust Stereo Visual-Inertial Odometry Using Nonlinear Optimization

Shujun Ma *, Xinhui Bai, Yinglei Wang and Rui Fang

School of Mechanical Engineering and Automation, Northeastern University, Shenyang 110819, China
* Correspondence: mashujun@me.neu.edu.cn; Tel.: +86-24-8369-1002

Received: 20 July 2019; Accepted: 26 August 2019; Published: 29 August 2019

Abstract: The fusion of visual and inertial odometry has matured greatly due to the complementarity of the two sensors. However, the use of high-quality sensors and powerful processors in some applications is difficult due to size and cost limitations, and there are also many challenges in terms of robustness of the algorithm and computational efficiency. In this work, we present VIO-Stereo, a stereo visual-inertial odometry (VIO), which jointly combines the measurements of the stereo cameras and an inexpensive inertial measurement unit (IMU). We use nonlinear optimization to integrate visual measurements with IMU readings in VIO tightly. To decrease the cost of computation, we use the FAST feature detector to improve its efficiency and track features by the KLT sparse optical flow algorithm. We also incorporate accelerometer bias into the measurement model and optimize it together with other variables. Additionally, we perform circular matching between the previous and current stereo image pairs in order to remove outliers in the stereo matching and feature tracking steps, thus reducing the mismatch of feature points and improving the robustness and accuracy of the system. Finally, this work contributes to the experimental comparison of monocular visual-inertial odometry and stereo visual-inertial odometry by evaluating our method using the public EuRoC dataset. Experimental results demonstrate that our method exhibits competitive performance with the most advanced techniques.

Keywords: localization; visual-inertial SLAM; state estimation; simultaneous localization and mapping

1. Introduction

In recent years, with the advancement of sparse nonlinear optimization theory, camera technology, and computing performance, Visual Simultaneous Localization And Mapping [VSLAM] technology has achieved tremendous development [1,2]. Visual SLAM approaches have been widely used and researched because of their simple equipment and remarkable effects. Real-time and robust state estimation plays a significant role in robotics. Accurate state estimation is of crucial importance in a variety of intelligent applications, such as robot autonomy, Augmented Reality [AR] and Virtual Reality (VR).

The vision-based state estimation method is able to estimate the 6-Degrees-Of-Freedom (6-DOF) state of sensors simultaneously and reconstruct a three-dimensional (3D) map of the surrounding environment. Many works based on nonlinear optimization have been reported, including SVO [3], LSD-SLAM [4], DSO [5], ORB-SLAM [6,7]. The ORB-SLAM system supports monocular cameras, stereo cameras, and depth cameras. The system is also designed based on the idea of PTAM [8], which is divided into three threads: tracking, mapping and loop closing. However, due to the existence of observation noise, the pose estimation of the feature points in space contains uncertainty, leading to localization errors in Visual Odometry. The position change between the current frame and the previous one can be calculated and continuously accumulated by Visual Odometry in order to realize the estimation of the motion process of a robot. During this process, the error of each estimation is collected, thus limiting the accuracy of Visual Odometry over time.

An Inertial Measurement Unit (IMU) is composed of an accelerometer and a gyroscope. The pose of the robot can be calculated and the three-dimensional heading information can be estimated, respectively, by the accelerometer and gyroscope in the IMU. However, the navigation errors caused by the low-frequency noise of the IMU work process will accumulate over time due to the integral operation, and the accuracy will easily drift, resulting in inaccurate pose estimation. A general and effective solution to these problems is to fuse visual and IMU measurements using a filter-based or optimization-based system. During the fusing process, IMU and Vision are combined to form a Visual-Inertial Odometry, which not only takes advantage of the flexibility of the visual method and is adaptable to a wide range of scenes, but also utilizes the high-precision features of the IMU in the short term. Therefore, research into the SLAM algorithm based on visual and inertial sensors is of great significance and application value, allowing robots to perceive the surrounding environment in order to obtain localization information.

In Visual-Inertial Odometry (VIO), the easiest way to handle visual and inertial measurements is to loosely couple the sensor fusion [9], whereby the IMU is considered to be a separate module for assisting in visual pose estimation, and then is fused by an extended Kalman filter (EKF). In comparison, tightly coupled visual-inertial algorithms can mainly be classified into two types: EKF-based algorithms [10–12] and optimization-based algorithms [13–15]. For example, MSCKF [16] is a popular EKF-based VIO method. MSCKF keeps several previous camera poses in the state vector, and arranges them in chronological order, which is also known as a sliding window. If a feature point is observed in several poses of the sliding window, the constraint will be established between these poses, and KF updates will be performed. A disadvantage of using a filter-based approach is that it can lead to suboptimal results due to the state of early linearization estimates.

Meanwhile, with the deepening of research and the improvement of computer performance, optimization-based methods have commonly been used in visual-inertial SLAM systems to ensure higher accuracy. A full smoothing method [17] for estimating the entire state history is described by solving a substantial nonlinear optimization problem. Although the outlook is promising, its computational complexity is high, and its real-time performance will gradually decrease as the map grows. Recently, the work proposed in [14] applied a keyframe-based approach to fuse visual-inertial measurements. The use of a sliding window and marginal technology [14,15] ensures the real-time operation of the system, and it has achieved remarkable success. Additionally, the IMU pre-integration technique proposed in [18] can avoid the repeated calculation of the integral when the linearization point changes. Mur-Artal et al. [15] suggested a VI-SLAM system based on their original ORB-SLAM, which originated from the pre-integration ideas of Forster [18]. Qin et al. [19] demonstrated a new Visual-Inertial System (VINS) which is also a complete VI-SLAM system based on tight coupling and nonlinear optimization. The initialization process is robust to unknown states and has excellent application potentials in the field of UAV navigation control.

Very few VIO solutions are planned for stereo or multi-camera systems [14,20,21] compared to the large amount of work that has been carried out for monocular systems. This could partly be due to the costs associated with processing additional images and matching features. Leutenegger et al. [14] proposed a complete optimization framework for multi-camera VIO that can run in real-time. Usenko et al. [20] introduced direct methods into stereo VIO to further improve accuracy. The tested dataset in [20] was acquired using a stereo visual inertial camera that was processed offline. IMU pre-integration was further improved in the stereo VIO in [20]; however, this solution is not suitable for practical applications. Liu et al. [22] proposed a stereo VIO assembled with three separate Kalman filters, including an attitude filter, an orientation filter, and a position filter. Wang et al. [23] proposed a stereo direct sparse odometry, which maintains high accuracy while achieving real-time pose estimation. Xiong et al. [24] presented a dual-stage EKF-based algorithm for the robust stereo VIO. Zheng et al. [25] introduced a tightly coupled filtering-based stereo VIO system using both points and lines. Some

VIO algorithms have been explored in other areas, such as initialization [26], online calibration [27], optimization [28,29], and camera type [30]. These solutions are either based on optimization methods or filtering methods. For the optimization-based methods, they require a powerful CPU for real-time use and cannot run on low-cost devices such as embedded devices. For the filtering-based methods, the localization accuracy may not be enough.

In this paper, we summarize our contributions as follows:

- The FAST feature detector was employed for its efficiency, and features were tracked by the KLT sparse optical flow algorithm to decrease the cost of computation.
- Stereo matching was performed, which can remove outliers in the stereo matching and feature tracking steps leading to reduction of the mismatch of feature points and improvement of the robustness and accuracy of the system.
- A tightly coupled nonlinear optimization method was employed to combine pre-integrated IMU measurements with visual measurements of stereo cameras, and thus a highly accurate and robust visual-inertial odometry was achieved, which can run in real-time on devices such as drones.
- The proposed method was extensively evaluated and validated in comparison to state-of-the-art open source VIO methods (including OKVIS [14], VINS-MONO [19] and S-MSCKF [16]) by using the EuRoC dataset.

2. Visual-Inertial Odometry Overview

The schematic depiction of the Visual-Inertial optimization framework is shown in Figure 1. The input of the system is the image acquired by the stereo cameras and the acceleration and angular velocity measured by the IMU. The output is a 6-degree-of-freedom pose with real scale, i.e., the trajectory of the inertial camera (robot) motion. The system starts with Visual-Inertial Preliminaries (Section 3), where features are extracted and tracked, and pre-integrate IMU measurements between two consecutive frames. Then the initialization preprocessing is introduced (Section 4). The inertial pose and visual pose are combined by visual inertia joint initialization to obtain the initial estimated value of the system, including pose, velocity, gyroscope bias, gravity vector. These values will iteratively and sequentially be updated by the tightly coupled visual-inertial odometry (Section 5). Finally, the 6-DOF pose can be obtained.

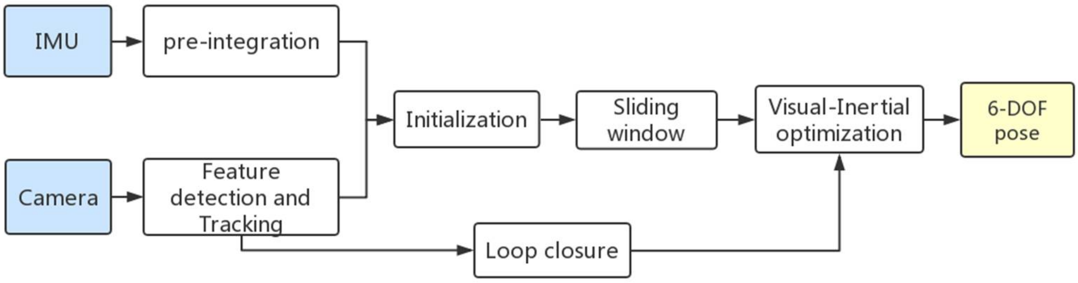

Figure 1. A block diagram illustrating our Visual-Inertial optimization framework.

In this work, we consider $(\cdot)_C$ to be the camera frame, $(\cdot)_B$ to be the body frame, and we regard the IMU frame as the body frame. The matrix R represents rotation. R_{WB} represents the rotation from the body frame to the world frame. P_{WB} is translation from the body frame to the world frame.

3. Visual-Inertial Preliminaries

This section gives the preliminary steps for stereo visual measurements and the inertial model. For visual measurements, features are tracked in real-time using the KLT optical flow algorithm. For IMU measurements, two consecutive frames are pre-integrated.

3.1. Visual Processing Frontend

In the Visual-Inertial odometry, the camera needs to track the pixel points of the captured image frame, and the pixel points need to be the feature points extracted from the previous and subsequent frames. However, there is only a small part of the overlap between successive image frames in many cases, and it would be inappropriate to match all the pixels, due to the massive demand for computing resources. Additionally, because the feature points are matched through the calculation process of the camera pose, it is unnecessary to extract many feature points. Therefore, in visual odometry, only the feature points from part of the image—i.e., the points with the most noticeable features in the image (corner points, points with sharp changes in brightness or at the edge of the contour)—are extracted and subsequently tracked.

In this paper, the feature points are extracted by the FAST [31] feature detector, and the feature tracking method is the KLT [32] optical flow algorithm. Through experiments, it is found that the KLT optical flow method takes up less CPU resources than the descriptor-based method, making the optical flow method more advantageous in robot applications. The KLT optical flow method was also used for stereo feature matching to save computing resources. Optical flow tracking is able to obtain the feature point coordinates of cam1 (the right camera of the stereo), which correspond to stereo cam0. Then, based on the principle of epipolar geometry, the following constraint can be defined as

$$X_2^T E X_1 = 0 \tag{1}$$

where X_1 and X_2 are the normalized plane coordinates of the feature points, $E = t^\wedge R$ is the essential matrix, and $t^\wedge$ is the antisymmetric matrix corresponding to the t vector. t and R are the translation vector and rotation matrix between the stereo cameras, respectively, which can be obtained from the initial calibration.

During the measurement, $X_2^T E X_1$ are not strictly equal to zero, due to the presence of noise. There is an error in this process, and if the error is greater than a certain threshold, it is marked as an outlier. X_1, X_2 are obtained by LK optical flow tracking according to the original image, and the accuracy of optical flow tracking is verified by the constraint (relative poses R and t) between stereo cameras. Therefore, the effect of removing outliers can be achieved. Meanwhile, several grids were added to the image, which was assigned with feature points. The number of feature points attached to each grid was checked to ensure that it did not exceed the maximum value of grid feature points.

In the visual processing frontend, outlier rejection was performed using two-point RANSAC similar to (16) in temporal tracking. Assuming that the normalized coordinates of the corresponding points of the previous and current frames are $X(x, y, 1)$, $X_2(x_2, y_2, 1)$, the following constraints can be established based on epipolar geometry:

$$X_2^T t^\wedge R X = 0 \tag{2}$$

where R is obtained from the average angular velocity of the IMU. When $X_1 = RX$, the coordinate system is unified into one coordinate system.

$$[x_2 \quad y_2 \quad 1] \cdot \begin{bmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} = 0 \tag{3}$$

Equation (3) can be expanded as

$$[y_1 - y_2 \quad -(x_1 - x_2) \quad x_1 y_2 - x_2 y_2] \cdot \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} = 0 \tag{4}$$

According to the RANSAC principle, two matching points of the previous frame and the current frame are arbitrarily selected, and the outliers with large errors are removed.

$$\begin{bmatrix} y_1 - y_2 & -(x_1 - x_2) & x_1y_2 - x_2y_2 \\ y_3 - y_4 & -(x_3 - x_4) & x_3y_4 - x_4y_3 \end{bmatrix} \cdot \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} = \begin{bmatrix} A_x \\ A_y \\ A_z \end{bmatrix}^T \cdot \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} \approx \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{5}$$

where $A_x = [y_1 - y_2\ y_3 - y_4]$. The following three formulas are available:

$$\begin{bmatrix} A_x \\ A_y \end{bmatrix}^T \cdot \begin{bmatrix} t_x \\ t_y \end{bmatrix} \approx A_z \cdot t_z \begin{bmatrix} A_x \\ A_z \end{bmatrix}^T \cdot \begin{bmatrix} t_x \\ t_z \end{bmatrix} \approx A_y \cdot t_y \begin{bmatrix} A_y \\ A_z \end{bmatrix}^T \cdot \begin{bmatrix} t_y \\ t_z \end{bmatrix} \approx A_x \cdot t_x \tag{6}$$

The values among A_x, A_y, A_z can be calculated, whereby if A_i is the minimum value, the corresponding $t_i = 1$ and the other two can be solved by matrix inversion. Then, the obtained t can be brought into Equation (4), and the value calculated by the coordinates of each matching point are regarded as the error. If the error satisfies certain conditions, the matching point is set as inliers. Finally, an inlier set will be obtained, and all inliers in the set will be substituted into Equation (5) to recalculate the new model by the least squares method. The model will be used to compute the final sum of error. Over multiple iterations, the inliers set with the smallest sum of errors is selected, thereby achieving the effect of removing the outliers. Additionally, performing circular matching [33] further removes the outlier matching step generated in the feature tracking and stereo matching between the previous and current stereo image pairs. Through the above steps, the mismatch of feature points is reduced, thereby improving the robustness and accuracy of the system.

3.2. IMU Measurement Model and Pre-Integration

As shown in Figure 2, the measurement rate of IMU is much faster than that of the visual camera. To simultaneously optimize the constraints of vision and IMU in a single framework, it is necessary to integrate the measurements of many IMUs between two adjacent visual keyframes into one constraint. Therefore, the theory of the pre-integration formula based on SO3 manifolds (18) was used in this paper.

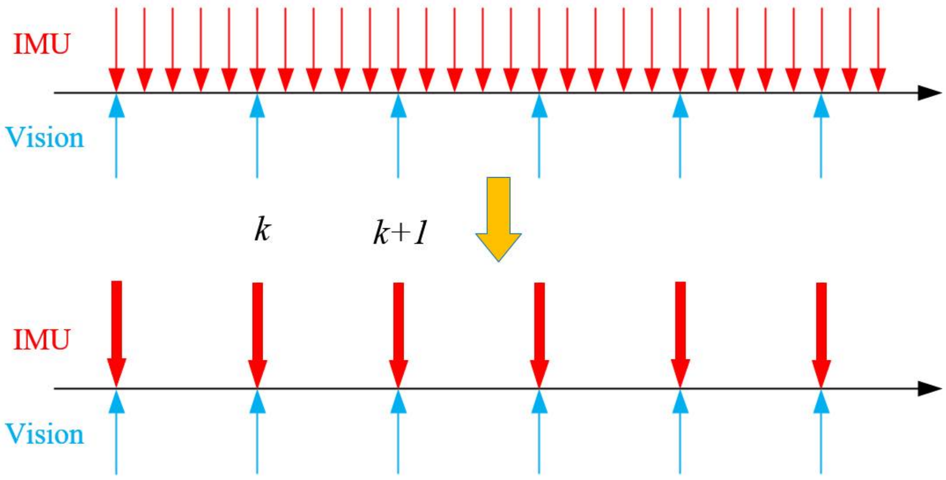

Figure 2. IMU pre-integration diagram.

The IMU measures acceleration and angular velocity in three directions within the inertial system by a three-axis accelerometer and a three-axis gyroscope. The IMU measurement is often affected by Gaussian white noise and zero offsets. The noise model of IMU can be expressed by the following formula:

$$\widetilde{\omega}_B(t) = \omega_B(t) + b_g(t) + \eta_g(t) \tag{7}$$

$$\widetilde{a}_B(t) = \mathrm{R}_{WB}^T(a_W(t) - g_W) + b_a(t) + \eta_a(t) \tag{8}$$

where $\widetilde{\omega}_B(t)$ is the angular velocity measured by the gyroscope; $\widetilde{a}_B(t)$ is the acceleration measured by the accelerometer; $\omega_B(t)$ and $a_B(t)$ are the real angular velocity and the real acceleration; b and η represent the corresponding zero-bias and white Gaussian noise, respectively.

To calculate the motion of the robot from the measured values of the IMU, the following kinematic models need to be introduced.

$$\dot{R}_{WB} = R_{WB}\omega_B^{\wedge}\dot{v}_W = a_W\dot{p}_W = v_W. \tag{9}$$

where $\dot{R}_{WB}$, $\dot{v}_W$ and $\dot{p}_W$ respectively represent the derivatives of the rotation matrix R_{WB}, the velocity vector v_W and the translation vector p_W with respect to time. Equation (9) describes the change in pose and velocity of the IMU in differential form. To obtain the value of the state of the IMU at each moment, the integration to Equation (9) needs to be determined. The pose and velocity of the IMU at time interval $[t, t + \Delta t]$ can be described as follows:

$$\begin{array}{c} R_{WB}(t+\Delta t) = R_{WB}(t)Exp(\omega_B(t)\Delta t) \\ v_W(t+\Delta t) = v_W(t) + a_W(t)\Delta t \\ p_W(t+\Delta t) = p_W(t) + v_W(t) + \frac{1}{2}a_W(t)\Delta t^2 \end{array} \tag{10}$$

where Δt is the time interval between two adjacent IMU measurements. According to Equations (7) and (8), $\omega_B(t)$ and $a_B(t)$ in Equation (10) can be rewritten as the following forms.

$$\begin{array}{c} R(t+\Delta t) = R(t)Exp\Big(\big(\widetilde{\omega}(t) - b_g(t) - \eta_g(t)\big)\Delta t\Big) \\ v(t+\Delta t) = v(t) + g\Delta t + R(t)(\widetilde{a}(t) - b_a(t) - \eta_a(t))\Delta t \\ p(t+\Delta t) = p(t) + v(t)\Delta t + \frac{1}{2}g\Delta t^2 + \frac{1}{2}R(t)(\widetilde{a}(t) - b_a(t) - \eta_a(t))\Delta t^2 \end{array} \tag{11}$$

It is worth mentioning that the subscript of the reference coordinate system in Equation (11) is hidden for the legibility of the formula.

Based on Equation (11), the positional relationship of the inertial component at the adjacent Δt can be obtained. The sampling frequency of the IMU can be optimized if a state is added that needs to be evaluated every time interval in the collection of IMU data, which, however, will result in a slow calculation and arduous process of the IMU data. The IMU pre-integration method combines the IMU measurements between two adjacent visual keyframes i and j into a composite term that constitutes the motion constraints of two adjacent visual keyframes. Assuming that the IMU is synchronized with the visual frame detection time and the measurement data is acquired at discrete time k, the pose relationship between the two keyframes $k = i$ and $k = j$ can be obtained by the IMU measurement. To avoid the situation where the estimation of the initial frame of the pose estimation leads to the recalculation of the integral, this paper uses the incremental expression, which is defined as

$$\begin{array}{c} \Delta R_{ij} = R_i^T R_j = \prod\limits_{k=i}^{j-1} Exp\Big(\big(\widetilde{\omega}_k - b_k^g - \eta_k^g\big)\Delta t\Big) \\ \Delta v_{ij} = R_i^T\big(v_j - v_i - g\Delta t_{ij}\big) = \sum\limits_{k=i}^{j-1} \Delta R_{ik}\big(\widetilde{a}_k - b_k^a - \eta_k^a\big)\Delta t \\ \Delta p_{ij} = R_i^T\Big(p_j - p_i - v_i\Delta t_{ij} - \frac{1}{2}g\Delta t_{ij}^2\Big) = \sum\limits_{k=i}^{j-1}\Big[\Delta v_{ik}\Delta t + \frac{1}{2}\Delta R_{ik}\big(\widetilde{a}_k - b_k^a - \eta_k^a\big)\Delta t^2\Big] \end{array} \tag{12}$$

It should be noted that the IMU biases are regarded as being constants in the time interval Δt. However, the estimated biases are changed with a small amount of δb during optimization. The Jacobians $J_{(\cdot)}^g$ and $J_{(\cdot)}^a$ are employed to account for a first-order approximation of the effect of changing the biases without explicitly recomputing the pre-integration. The pose and velocity can be described as

$$\begin{gathered} R_{WB}^{i+1} = R_{WB}^{i}\Delta R_{i,i+1}Exp\left(J_{\Delta R}^{g}b_{g}^{i}\right) \\ v_{B}^{i+1} = v_{B}^{i} + g_{W}\Delta t_{i,i+1} + R_{WB}^{i}\left(\Delta v_{i,i+1} + J_{\Delta v}^{g}b_{g}^{i} + J_{\Delta v}^{a}b_{a}^{i}\right) \\ R_{WB}^{i+1} = R_{WB}^{i}\Delta R_{i,i+1}Exp\left(J_{\Delta R}^{g}b_{g}^{i}\right) \\ v_{B}^{i+1} = v_{B}^{i} + g_{W}\Delta t_{i,i+1} + R_{WB}^{i}\left(\Delta v_{i,i+1} + J_{\Delta v}^{g}b_{g}^{i} + J_{\Delta v}^{a}b_{a}^{i}\right) \end{gathered} \tag{13}$$

4. Visual-Inertial Initialization

The primary purpose of Visual-Inertial SLAM system initialization is to obtain the parameters necessary for the system to optimize and the initial values of the state. Since the stereo inertial tightly coupled system is a highly nonlinear system, it becomes very sensitive to specific initial values. The initialization quality directly affects the robustness and accuracy of the localization of the entire tightly coupled system. Therefore, it is necessary to initialize the system properly to provide correct parameters and initial values.

In the process of initialization, the information that needs to be initialized or estimated can be divided into two categories. One is the parameters that almost remain unchanged during the system operation, such as absolute scale and gravity acceleration. Another is the initial values of the system's starting state quantities, including the pose and velocity information of the first few frames, the position of the 3D landmark points, and the zero offset of the IMU accelerometer and gyroscope. The initialization can be separated into two processes. Firstly, the stereo visual initialization can initialize the initial frame pose and the three-dimensional landmark position information based on the sliding window. Then, the visual inertia joint initialization can initialize the absolute scale, gravity acceleration, and camera speed information.

4.1. Stereo Vision Initialization Based on a Sliding Window

In the process of stereo visual initialization based on a sliding window, an up-to-scale purely visual structure of the sliding window is constructed to restore the information of the initial frames pose and the three-dimensional landmark point position.

Two frames are first selected that have sufficient feature disparity in the sliding window. Next, the eight-point method of the polar geometry recovery pose is used to recover the essential matrix E. The scale of the translation transformation is fixed, and E is used to retrieve the motion pose and triangulate the 3D map points. After initializing a batch of 3D points, the perspective-n-point (PnP) method is employed to solve the pose information of the remaining frames in the sliding window. A global full bundle adjustment is used to minimize the total reprojection error for all feature observations in all frames. At this point, the measurement can obtain the pose information and the three-dimensional information of the feature points. Since the external parameters $T_{CB} = (R_{CB}, p_{CB})$ between the camera and the IMU are known, all variables can be converted to representations in the IMU coordinate system:

$$R_{WB_k} = R_{CC_k}R_{CB}^{-1} \tag{14}$$

$$sp_{B_k}^{W} = sp_{C_k}^{W} + R_{WC}p_{B}^{C} \tag{15}$$

where s is an unknown scale factor that aligns the visual structure to the metric scale, and feature positions $(\cdot)^{W}$ are represented with respect to the world coordinate system.

4.2. Visual Inertial Joint Initialization

Through the visual inertia joint initialization, the absolute scale, gravity acceleration, camera state speed information, and IMU zero offset can be initialized.

4.2.1. Gyroscope Bias Estimation

The IMU bias is firstly initialized under the assumption that the IMU gyroscope zero offset b_g is constant in the current window. Considering the adjacent kth and k+1th frames in the window, in the stereo visual initialization of the previous step, we can obtain the rotations R_k and R_{k+1} in their relative world coordinate system. The angular velocity among the results of IMU pre-integration can be derived from Equation (12). The gyroscope bias can be predicted by minimizing the error of the term.

$$\min_{b_g}\sum_{k}^{N-1} \|Log\Big(\Big(\Delta R_{k,k+1}Exp\Big(J_{\Delta R}^{g}b_g\Big)\Big)^T R_{BW}^{k+1}R_{WB}^{k}\Big)\|^2 \tag{16}$$

By solving this least squares problem, we can estimate the gyroscope zero offsets b_g.

4.2.2. Accelerometer Bias, Gravity and Metric Scale Estimation

It is difficult to solve the accelerometer bias during the initialization process because of g and the accelerometer bias are measured simultaneously. However, the accelerometer bias b_a has little effect on system stability, a rough initial guess can thus be made, and the bias will be continuously optimized during subsequent optimizations. Therefore, in the initialization step, we set the accelerometer offset b_a to zero.

The remaining estimated parameters: the speed of the state, the gravity, and the scale factor are defined as the variables:

$$\chi_{v,s} = \Big[v_{B_0}, v_{B_1}, \cdots v_{B_n}, g, s\Big] \tag{17}$$

where v_{B_k} is velocity in the body frame while tracking the kth image, g is the gravity vector in the world frame, and s is the scale of Visual-Inertial SLAM system to metric units.

Considering two consecutive frames b_k and b_{k+1}, we can develop the following linear measurement model from Equations (12), (14) and (15):

$$\begin{aligned} \hat{z}_{b_{k,k+1}} &= \Big[\Delta p_{k,k+1} - {R_{WB}^{k}}^T\Big(R_{WC}^{k+1} - R_{WC}^{k}\Big)p_B^C\Big] \\ &= \Big[-R_{WB}^{k}\Delta t_{k,k+1} \;\; {R_{WB}^{k}}^T\Big(p_{C_{k+1}}^{W} - p_{C_k}^{W}\Big) \;\; -\tfrac{1}{2}{R_{WB}^{k}}^T\Delta t_{k,k+1}\Big]\chi_{v,s} \\ &= H_{k,k+1}\chi_{v,s} \end{aligned} \tag{18}$$

The initial value can be guessed by solving the following least squares problem:

$$\min_{v,s,g}\Big\|\hat{z}_{b_{k,k+1}} \quad - H_{k,k+1}\chi_{v,s}\Big\|^2 \tag{19}$$

The speed of each local frame, the gravity vector (including direction and intensity), and the scale factor can be obtained.

5. Tightly Coupled Stereo Visual-Inertial Odometry

After the state initialization, a highly accurate state is estimated using a sliding window estimator. A state estimator based on the combination of stereo vision and IMU is described in this section. The state estimator can be regarded as a backend based on nonlinear graph optimization. It optimizes the coordinates of the 3D landmark points and the pose, velocity and IMU bias of the inertial camera. In this section, the state vector of the system estimation is defined firstly, then the expression of the visual inertia optimization term is given. After that, the explicit calculation formulas of the visual error term and the inertia error term are provided, and finally, the system's marginalization method is presented.

5.1. System State Vector

First, the state vectors of the system are defined. The state vector to be estimated are the state variable χ of the inertial camera at the image time k, including position p_B, pose R_{WB}, velocity v_B, gyroscope bias b_g, and accelerometer bias b_a.

$$\chi = [\mathrm{x}_0, \mathrm{x}_1 \cdots x_n] \tag{20}$$

$$x_n = \left[R_{WB}, \mathrm{p}_B, \mathrm{v}_B, \mathrm{b}_g, b_a\right] \tag{21}$$

5.2. IMU and Visual Error Term

In the traditional visual SLAM or visual odometers, nonlinear optimization is used to obtain the best estimate of camera pose and 3D landmark points by minimizing the reprojection error of feature points in the camera frame. Once inertial measurements are introduced, constraints are applied to the camera's continuous motion poses, the IMU speed and the estimated IMU bias for continuous time, resulting in an increase of the system state variables. The IMU error term can be defined as:

$$\begin{gathered}
e_{IMU}(\hat{z}_{b_{k,k+1}}, \chi) = \rho\left(\left[e_R^T e_V^T e_p^T\right] \sum\nolimits_I \left[e_R^T e_V^T e_p^T\right]^T\right) + \rho\left(e_b^T \sum\nolimits_R e_b\right) \\
e_R = Log\left(\left(\Delta R_{k,k+1} Exp\left(J_{\Delta R}^g b_g^{k+1}\right)\right)^T R_{BW}^{k+1} R_{WB}^k\right) \\
e_v = R_{BW}^k\left(v_B^{k+1} - v_B^k - g\Delta t_{k,k+1}\right) - \left(\Delta v_{k,k+1} + J_{\Delta v}^g b_g^{k+1} + J_{\Delta v}^a b_a^{k+1}\right) \\
e_v = R_{BW}^k\left(v_B^{k+1} - v_B^k - g\Delta t_{k,k+1}\right) - \left(\Delta v_{k,k+1} + J_{\Delta v}^g b_g^{k+1} + J_{\Delta v}^a b_a^{k+1}\right) \\
e_p = R_{BW}^k\left(p_B^{k+1} - p_B^k - v_B^k \Delta t_{k,k+1} - \tfrac{1}{2} g \Delta t_{k,k+1}^2\right) - \left(\Delta p_{k,k+1} + J_{\Delta p}^g b_g^{k+1} + J_{\Delta p}^a b_a^{k+1}\right) \\
e_b = b^{k+1} - b^k
\end{gathered} \tag{22}$$

where ρ is the Huber robust cost function, and $\sum_I$ is the information matrix of the pre-integration and $\sum_R$ is the bias random walk.

Consider the feature in the j^{th} image, and the reprojection error of feature points e_{proj} for a given match k to be defined as follows:

$$\begin{gathered}
e_{proj}(k, j) = \rho\left(\left(x^k - \pi\left(X_c^k\right)\right)^T \sum\nolimits_k \left(x^k - \pi\left(X_c^k\right)\right)\right) \\
X_C^k = R_{CB} R_{BW}^j \left(X_W^k - p_{WB}^j\right) + p_{CB}
\end{gathered} \tag{23}$$

where x^k is the location of the observation keypoint in the image, X_c^k is the map point in world coordinates, and p_{WB}^j represents the translation of IMU relative to the world coordinate system at jth frames, and $\sum_k$ is the information matrix associated with the keypoint scale.

5.3. Marginalization

As time goes by, the feature points and camera poses will accumulate, and the amount of optimization calculation will increase accordingly. Because the dimension of the visual inertia system to be optimized is more than visual system, the number of variables to be optimized is large. If the estimated variables are not limited, the calculation load will surge with time. Therefore, the non-linearization is performed while the state of the estimated system is marginalized, i.e., the time-limited keyframes are removed without changing the consistency of the estimation.

The visual-inertial optimization term is a least squares problem, which can generally be solved by the Gauss-Newton iteration method. This can be defined as:

$$H\delta x = b \tag{24}$$

Suppose δx_a is the state variable to be marginalized, and δx_b is the state variable to be retained. Depending on the conditional independence, we can simplify the process of marginalization as follows:

$$\begin{bmatrix} H_a & H_b \\ H_b^T & H_d \end{bmatrix} \begin{bmatrix} \delta x_a \\ \delta x_b \end{bmatrix} = \begin{bmatrix} b_a \\ b_b \end{bmatrix} \tag{25}$$

δx_a in the above formula is the variable for marginalization, such as a camera pose. We cannot directly delete δx_b and its related landmark points, because this would reduce the constraint information. Therefore, the following Schur complement can be used to eliminate the element:

$$\left(H_c - H_c H_a^{-1} H_b\right)\delta x_b = b_b - H_c H_a^{-1} b_a \tag{26}$$

where $H_c - H_c H_a^{-1} H_b$ and $b_b - H_c H_a^{-1} b_a$ are marginalized H matrices and error quantities, and state variables δx_a are marginalized. By repeating the marginalization, as the new state variable is added, the number of state variables to be optimized remains constant, which dramatically decreases the amount of computation.

6. Experimental Results

To validate the effectiveness of the proposed reduced mismatch algorithm, the EuRoC dataset of images was used to test the visual front end. Two consecutive frames of images were randomly selected, a total of eight groups (among monocular image using the left camera image), keypoints were extracted for both the monocular image and stereo image, and the RANSAC algorithm was then used to reduce the mismatch. By calculating the comparison between the number of points removed and the original untreated points, it was found that the image matching accuracy after stereo processing was high, with the specific results shown in Figure 3.

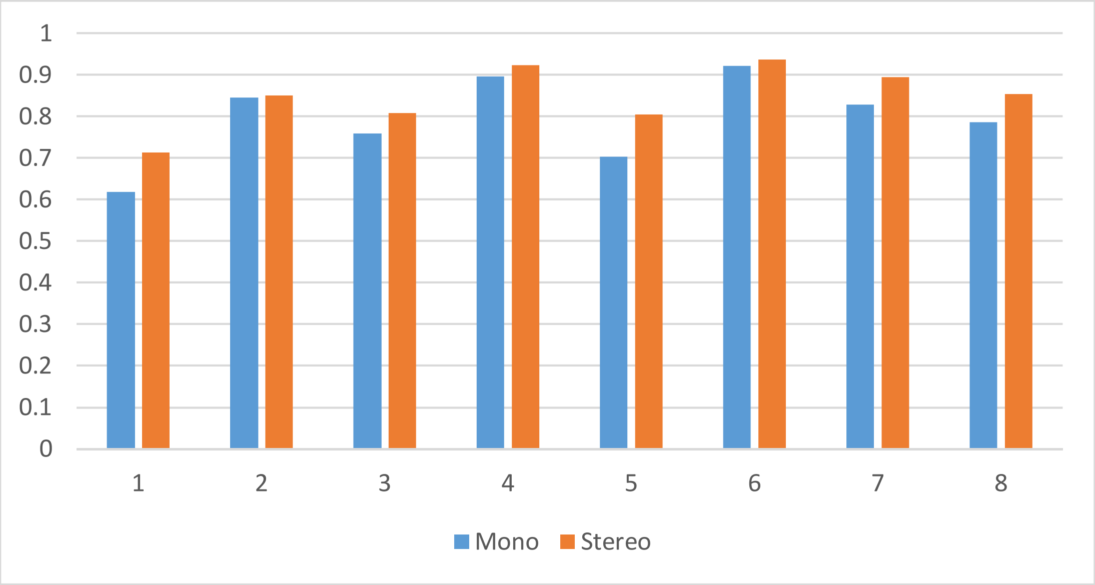

Figure 3. Mismatch accuracy of different types of cameras.

An experiment was performed to validate the proposed method. A comparison with the existing state-of-art visual-inertial odometry algorithms VINS-Mono and S-MSCKF through the EuRoC MAV Visual-Inertial datasets was also made. The EuRoC datasets were retrieved by a microdrone. The UAV was equipped with sensors such as stereo cameras (Aptina MT9V034, 20FPS) and IMU (ADIS16448, 200 Hz), and the real-time ground truth states were acquired by the motion capture device. The movement of the drone had a large rotation and a significant change in light intensity, which was challenging for our vision-based odometry. The tracking of the visual front end is shown in Figure 4.

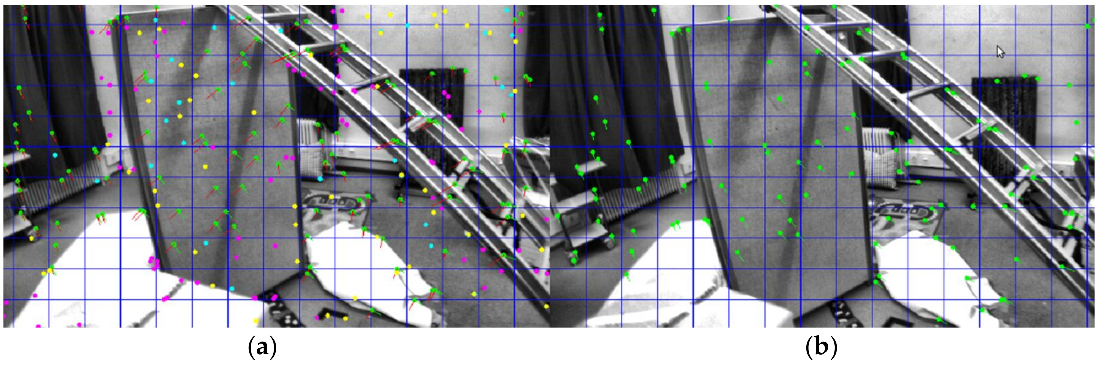

Figure 4. The tracking of the visual front end: (**a**) left image; (**b**) right image.

In these experiments, the proposed method was compared with VINS-Mono and S-MSCKF, which were tested on a desktop computer. The desktop CPU was equipped with quad-core i7-6700 3.4 Hz, the memory is 8 GB. The EuRoC datasets have eleven datasets, and four representative sequences, including MH_03_median, MH 05 difficult, V2_01_easy, and V2_02_median, were used in the dataset, to compare with VINS-Mono. Figure 5 shows the trajectory for the sequence MH_03_median, MH 05 difficult, V2_01_easy, and V2_02_median. Obviously, the estimated trajectory by our method is even closer to the real one.

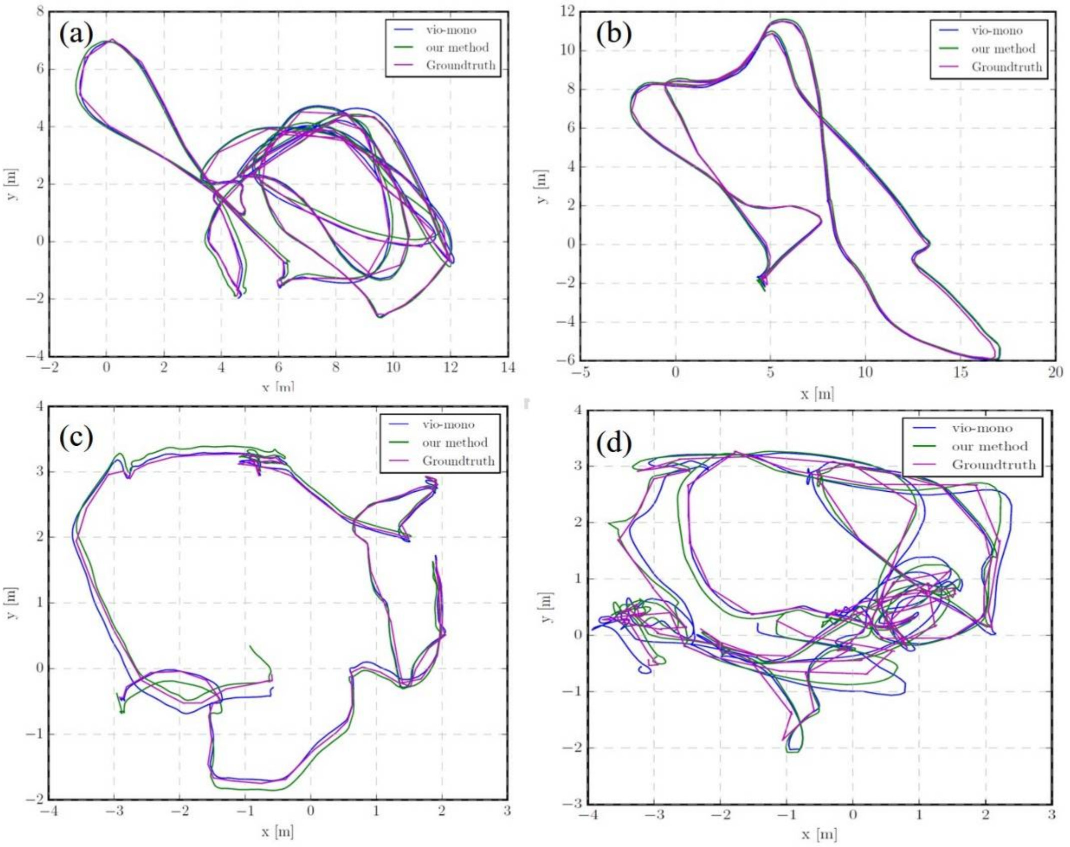

Figure 5. Trajectory compared with VINS in (**a**) MH_03_median, (**b**) MH 05 difficult, (**c**) V2_01_easy, and (**d**) V2_02_median.

The relationship between the translation error and the distance is shown in Figure 6. Due to space limitations, only the results of MH_03_median and MH_05_difficult are demonstrated. In the error graph, the proposed method produces the smallest translation error as the distance increases. Overall, the estimated error of this method is less than 5% on all sequences.

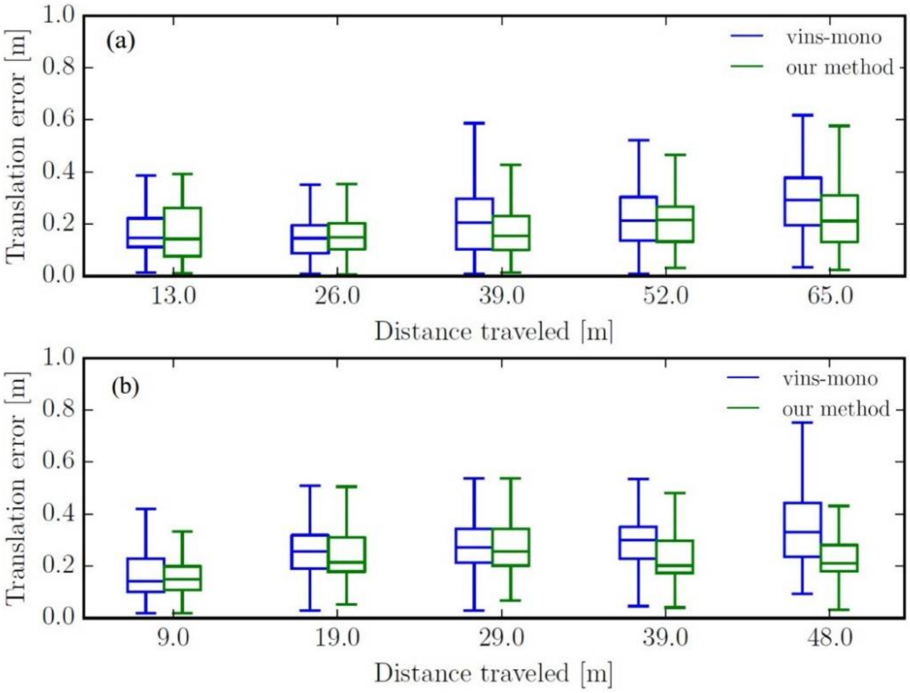

Figure 6. Translation error plot: (**a**) MH_03_median, (**b**) MH_05_median.

The relationship between the translation error and the distance is shown in Figure 6. Due to space limitations, only the results of MH_03_median and MH_05_difficult are demonstrated. In the error graph, the proposed method produces the smallest translation error as the distance increases. Overall, the estimated error of this method is less than 5% on all sequences.

In addition, we also compared our method with VINS-Mono in terms of CPU resource requirements, and the results are listed in Table 1. As can be seen, our method is comparable to the VINS-Mono on the CPU load, but the memory utilization statistics appears relatively better. Thus, from Figures 5 and 6 and Table 1, it can be concluded that our method obtains comparable accuracy at the start of the trajectory, but in a more efficient manner, and as the distance increases, the method begins to have much better performance than VINS-Mono.

Table 1. CPU load and memory utilization statistics of the algorithms (%).

Sequence	CPU Load		Memory Utilization	
	Ours	**VINS-Mono**	**Ours**	**VINS-Mono**
V1_01_easy	**40.25**	43.14	**26.18**	34.09
V1_02_medium	**34.05**	36.78	**25.98**	33.22
V1_03_difficult	36.72	**34.83**	**26.72**	34.75
V2_01_easy	42.32	**39.45**	**26.04**	32.02
V2_02_medium	45.88	**42.37**	**25.65**	34.48
V2_03_difficult	/	40.76	**/**	34.18
MH_01_easy	40.59	**38.87**	**26.24**	34.08
MH_02_easy	**37.06**	39.08	**26.43**	34.11
MH_03_medium	**38.57**	40.69	**26.06**	35.04
MH_04_difficult	44.02	**41.33**	**26.28**	34.77
MH_05_difficult	45.38	**40.9**	**26.2**	37.8
Average	40.48	**39.84**	**26.18**	34.41

The numbers in bold indicate the lower computational cost of the algorithms.

To evaluate the performance of this method, the tool evo (https://github.com/MichaelGrupp/evo) is used to determine the Root Mean Square Error (RMSE) of the experimental results, and the results are shown in Table 2. Notably, stereo feature matching failed because of the continuous inconsistency in brightness between stereo images in V2_03_difficult. Therefore, V2_03_difficult sequence was not included here. In the initial stage, the accuracy of our method is not greatly superior to VINS-Mono. As the flight distance increases, however, the error of our method becomes significantly lower than that of VINS, ensuring comparably accurate results over long paths. In the end, our method maintains the best performance, with an average RMSE of 0.075 m, as demonstrated in Table 2. It should be emphasized here that our method successfully recovers the metric scale, which, however, could not be achieved just by using the monocular camera. Compared to the optimization-based monocular SLAM, our method is more accurate than VINS in most scenarios except for individual scenes. For filter-based S-MSCKF, its accuracy is much lower than our method and VINS, which proves that optimization-based methods have greater potential than filter-based methods. It is worth mentioning that VINS-Mono, slightly inferior to our proposed algorithm, has outperformed both OKVIS and MSCKF in our results as shown in Table 2. These findings may not agree with the published results in [16] as VINS-Mono in their tests are running at a relatively low frequency without loop closure detection. When playing at full strength, VINS-Mono has experimentally been proved to be state-of-the-art and perform better than the widely used VIO systems, including OKVIS and MSCKF [33]. In summary, the optimization-based method has high localization accuracy and low memory utilization, while the filter-based method has advantages in computing resources. This is a remarkable result of our optimization-based stereo method, compared to the filter-based stereo VIO (16) and the optimization-based monocular VIO (19).

Table 2. RMSE of the algorithms (m).

Sequence	Ours	VINS-Mono	S-MSCKF	OKVIS
V1_01_easy	**0.039**	0.043	0.065	0.208
V1_02_medium	0.051	**0.047**	0.154	0.19
V1_03_difficult	0.105	**0.09**	0.281	0.195
V2_01_easy	**0.051**	0.062	0.069	0.172
V2_02_medium	**0.11**	0.121	0.147	0.181
V2_03_difficult	/	**0.113**	/	0.327
MH_01_easy	**0.053**	0.066	0.215	0.169
MH_02_easy	**0.078**	0.081	0.226	0.193
MH_03_medium	**0.041**	0.045	0.202	0.285
MH_04_difficult	0.134	**0.129**	0.323	0.386
MH_05_difficult	**0.089**	0.095	0.224	0.455
Average	**0.075**	0.081	0.19	0.251

The numbers in bold indicate the smallest values of RMSE of the algorithms.

7. Conclusions

In this paper, a robust stereo visual-inertial tightly coupled localization method was proposed. The method outputs the pose information of the drone in real-time through the onboard sensor, namely, the camera and the inertial sensing unit. The accuracy, real-time and robustness of the proposed method are validated by comparisons with the state of art open source algorithms in real-time operation in the EuRoC dataset. In most cases, this method shows better performance and lower computational cost, when running on low-computing devices. In future work, experiments will be conducted based on the drones, in combination with the proposed localization method as well as obstacle avoidance algorithms to achieve autonomous navigation.

Author Contributions: Conceptualization, S.M.; Data curation, X.B. and R.F.; Formal analysis, X.B.; Funding acquisition, S.M.; Investigation, S.M., Y.W. and R.F.; Methodology, S.M. and X.B.; Project administration, S.M.; Resources, S.M.; Software, Y.W. and X.B.; Supervision, S.M.; Validation, S.M. and X.B.; Visualization, Y.W.; Writing—original draft, S.M and X.B.; Writing—review & editing, S.M.

Funding: This research was funded by the National Natural Science Foundation of China (Grant Nos. 51505076), the Natural Science Foundation of Liaoning Province (No.2015020105), the Fundamental Research Funds for the Central Universities (N180304016, N140304010).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Davison, A.J.; Reid, I.D.; Molton, N.D.; Stasse, O. MonoSLAM: Real-Time Single Camera SLAM. *IEEE Trans. Pattern Anal. Mach. Intell.* **2007**, *29*, 1052–1067. [CrossRef] [PubMed]
2. Eade, E.; Drummond, T. Unified loop closing and recovery for real time monocular SLAM. In Proceedings of the British Machine Vision Conference 2008, Leeds, UK, 1–4 September 2008; p. 136.
3. Forster, C.; Pizzoli, M.; Scaramuzza, D. SVO: Fast semi-direct monocular visual odometry. In Proceedings of the 2014 IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China, 31 May–5 June 2014; pp. 15–22.
4. Engel, J.; Schöps, T.; Cremers, D. LSD-SLAM: Large-Scale Direct Monocular SLAM. *Inf. Secur. Appl.* **2014**, *8690*, 834–849.
5. Engel, J.; Koltun, V.; Cremers, D. Direct Sparse Odometry. *IEEE Trans. Pattern Anal.* **2018**, *40*, 611–625. [CrossRef] [PubMed]
6. Mur-Artal, R.; Tardos, J.D. ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras. *IEEE Trans. Robot.* **2017**, *33*, 1255–1262. [CrossRef]
7. Mur-Artal, R.; Montiel, J.M.M.; Tardos, J.D. ORB-SLAM: A Versatile and Accurate Monocular SLAM System. *IEEE Trans. Robot.* **2015**, *31*, 1147–1163. [CrossRef]
8. Klein, G.; Murray, D. Parallel tracking and mapping for small AR workspaces. In Proceedings of the 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality, Nara, Japan, 13–16 November 2007; pp. 1–10.
9. Weiss, S.; Achtelik, M.W.; Lynen, S.; Chli, M.; Siegwart, R. Real-time onboard visual-inertial state estimation and self-calibration of MAVs in unknown environments. In Proceedings of the 2012 IEEE International Conference on Robotics and Automation, Saint Paul, MN, USA, 14–18 May 2012; pp. 957–964.
10. Mourikis, A.I.; Roumeliotis, S.I. A Multi-State Constraint Kalman Filter for Vision-aided Inertial Navigation. In Proceedings of the 2007 IEEE International Conference on Robotics and Automation, Roma, Italy, 10–14 April 2007; pp. 3565–3572.
11. Li, M.; Mourikis, A.I. High-precision, consistent EKF-based visual-inertial odometry. *Int. J. Robot. Res.* **2013**, *32*, 690–711. [CrossRef]
12. Bloesch, M.; Omani, S.; Hutter, M.; Siegwart, R. Robust Visual Inertial Odometry Using a Direct EKF-Based Approach. In Proceedings of the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, Germany, 28 September–2 October 2015; pp. 298–304.
13. Shen, S.J.; Michael, N.; Kumar, V. Tightly-Coupled Monocular Visual-Inertial Fusion for Autonomous Flight of Rotorcraft MAVs. In Proceedings of the 2015 IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, USA, 26–30 May 2015; pp. 5303–5310.
14. Leutenegger, S.; Lynen, S.; Bosse, M.; Siegwart, R.; Furgale, P. Keyframe-based visual-inertial odometry using nonlinear optimization. *Int. J. Robot. Res.* **2015**, *34*, 314–334. [CrossRef]
15. Mur-Artal, R.; Tardos, J.D. Visual-Inertial Monocular SLAM with Map Reuse. *IEEE Robot. Autom. Lett.* **2017**, *2*, 796–803. [CrossRef]
16. Sun, K.; Mohta, K.; Pfrommer, B.; Watterson, M.; Liu, S.; Mulgaonkar, Y.; Taylor, C.J.; Kumar, V. Robust Stereo Visual Inertial Odometry for Fast Autonomous Flight. *IEEE Robot. Autom. Lett.* **2018**, *3*, 965–972. [CrossRef]
17. Bryson, M.; Johnson-Roberson, M.; Sukkarieh, S. Airborne Smoothing and Mapping using Vision and Inertial Sensors. In Proceedings of the 2009 IEEE International Conference on Robotics and Automation (ICRA), Kobe, Japan, 12–17 May 2009; pp. 3143–3148.
18. Forster, C.; Carlone, L.; Dellaert, F.; Scaramuzza, D. On-Manifold Preintegration for Real-Time Visual—Inertial Odometry. *IEEE Trans. Robot.* **2017**, *33*, 1–21. [CrossRef]
19. Qin, T.; Li, P.; Shen, S. VINS-Mono: A Robust and Versatile Monocular Visual-Inertial State Estimator. *IEEE Trans. Robot.* **2018**, *34*, 1004–1020. [CrossRef]

20. Usenko, V.; Engel, J.; Stuckler, J.; Cremers, D. Direct Visual-Inertial Odometry with Stereo Cameras. In Proceedings of the 2016 IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden, 16–21 May 2016; pp. 1885–1892.
21. Kitt, B.; Geiger, A.; Lategahn, H. Visual odometry based on stereo image sequences with ransac-based outlier rejection scheme. In Proceedings of the 2010 IEEE Intelligent Vehicles Symposium, La Jolla, CA, USA, 21–24 June 2010; pp. 486–492.
22. Liu, Y.; Xiong, R.; Wang, Y.; Huang, H.; Xie, X.J.; Liu, X.F.; Zhang, G.M. Stereo Visual-Inertial Odometry with Multiple Kalman Filters Ensemble. *IEEE Trans. Ind. Electron.* **2016**, *63*, 6205–6216. [CrossRef]
23. Wang, R.; Schworer, M.; Cremers, D. Stereo DSO: Large-scale direct sparse visual odometry with stereo cameras. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 3903–3911.
24. Xiong, X.; Chen, W.; Liu, Z.; Shen, Q. DS-VIO: Robust and Efficient Stereo Visual Inertial Odometry based on Dual Stage EKF. *arXiv* **2019**, arXiv:1905.00684.
25. Zheng, F.; Tsai, G.; Zhang, Z.; Liu, S.; Chu, C.-C.; Hu, H. Trifo-VIO: Robust and Efficient Stereo Visual Inertial Odometry using Points and Lines. In Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1–5 October 2018; pp. 3686–3693.
26. Yang, G.; Zhao, L.; Mao, J.; Liu, X. Optimization-Based, Simplified Stereo Visual-Inertial Odometry with High-Accuracy Initialization. *IEEE Access* **2019**, *7*, 39054–39068. [CrossRef]
27. Jung, J.H.; Heo, S.; Park, C.G. Stereo visual-inertial odometry with an online calibration and its field testing. *E3S Web Conf.* **2019**. [CrossRef]
28. Li, Y.; Zhong, X.; Tian, J.; Zou, C.; Peng, X. Stereo Visual Inertial Odometry Using Incremental Smoothing. In Proceedings of the 2018 37th Chinese Control Conference (CCC), Wuhan, China, 25–27 July 2018; pp. 5334–5339.
29. Von Stumberg, L.; Usenko, V.; Cremers, D. Direct sparse visual-inertial odometry using dynamic marginalization. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, Australia, 21–25 May 2018; pp. 2510–2517.
30. Mueggler, E.; Gallego, G.; Rebecq, H.; Scaramuzza, D. Continuous-time visual-inertial odometry for event cameras. *IEEE Trans. Robot.* **2018**, *34*, 1425–1440. [CrossRef]
31. Trajković, M.; Hedley, M. Fast corner detection. *Image Vis. Comput.* **1998**, *16*, 75–87. [CrossRef]
32. Hartley, R.; Zisserman, A. *Multiple View Geometry in Computer Vision*; Cambridge University Press: Cambridge, UK, 2003.
33. Delmerico, J.; Scaramuzza, D. A benchmark comparison of monocular visual-inertial odometry algorithms for flying robots. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, Australia, 21–25 May 2018; pp. 2502–2509.