

Article

A Resilient Method for Visual–Inertial Fusion Based on Covariance Tuning

Kailin Li, Jiansheng Li *, Ancheng Wang, Haolong Luo, Xueqiang Li and Zidi Yang

Institute of Geospatial Information, Information Engineering University, Zhengzhou 450001, China

* Correspondence: ljszhx@163.com

Abstract: To improve localization and pose precision of visual–inertial simultaneous localization and mapping (viSLAM) in complex scenarios, it is necessary to tune the weights of the visual and inertial inputs during sensor fusion. To this end, we propose a resilient viSLAM algorithm based on covariance tuning. During back-end optimization of the viSLAM process, the unit-weight root-mean-square error (RMSE) of the visual reprojection and IMU preintegration in each optimization is computed to construct a covariance tuning function, producing a new covariance matrix. This is used to perform another round of nonlinear optimization, effectively improving pose and localization precision without closed-loop detection. In the validation experiment, our algorithm outperformed the OKVIS, R-VIO, and VINS-Mono open-source viSLAM frameworks in pose and localization precision on the EuRoc dataset, at all difficulty levels.

Keywords: resilient sensor fusion; simultaneous localization and mapping; visual–inertial fusion; nonlinear optimization; covariance tuning



Citation: Li, K.; Li, J.; Wang, A.; Luo, H.; Li, X.; Yang, Z. A Resilient Method for Visual–Inertial Fusion Based on Covariance Tuning. *Sensors* **2022**, *22*, 9836. <https://doi.org/10.3390/s22249836>

Academic Editors: Shuanggen Jin and Rong Wang

Received: 4 November 2022

Accepted: 12 December 2022

Published: 14 December 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Due to the maturation of positioning navigation and time (PNT) systems [1] and elastic PNT frameworks [2], multisource PNT data fusion techniques are becoming increasingly intelligent and adaptive. Visual–inertial navigation systems (VINS) are commonly used to combine multisource PNT data, and they are commonly used in mobile equipment, mobile robots, and small flying devices for simultaneous localization and mapping (SLAM). The development of a robust visual–inertial fusion algorithm for pose and location determination would represent a significant step forward for the use of intelligent vehicles in complex or dynamic environments.

Due to the camera's characteristic of small size, low cost, low power consumption and easy assembly, the visual simultaneous localization and mapping system (VSLAM) has attracted wide attention [3]. As an important branch of the VSLAM system, visual odometry (VO) is widely studied, such as library for visual odometry (LIBVISO) [4], semi-direct monocular visual odometry (SVO) [5], and direct sparse odometry (DSO) [6] which are the three most representative visual odometry algorithms. Mainstream VSLAM systems can be divided into two categories. The first one is filtering-based methods, such as MonoSLAM [7]. The second one is the optimization methods of bundle adjustment (BA), such as parallel tracking and mapping (PTAM) [8]. Since Strasdat [9] pointed out that optimization algorithms are more cost-effective than filtering algorithms, BA-based VSLAM algorithms have developed rapidly, such as ORB-SLAM [10] with sparse features and LSD-SLAM [11] with dense features. These algorithms achieve better relative accuracy in common indoor scenes. However, the VSLAM algorithm inevitably has the disadvantages of difficult scale estimation (monocular camera), strong scene texture dependence, great influence by illumination, and is extremely unstable under high dynamics. The high-precision angular velocity and acceleration measurement of inertial sensors in a short time can make up for the shortcomings of the VSLAM system and improve the accuracy and stability of the

system [12]. Recently, visual-inertial SLAM has become an important area of study in SLAM research. In viSLAM, the visual sensor front end extracts features directly from sensor-captured images or through visual features. The high short-term precision of inertial measurement units (IMUs) is then used to constrain the visual data, which results in precise localization. The development of filter-based and factor graph optimization (FGO)-based sensor fusion techniques led to the emergence of several highly performant open-source visual-inertial frameworks. One example of a filter-based open-source framework is the multi-state constraint Kalman filter (MSCKF) proposed by Mourikis in 2007 [13]. MSCKF treats all poses in the active window as filter state variables, and the state vector does not include 3D feature positions, which reduces computational complexity to some extent. However, the local state estimates of the MSCKF algorithm tend to be imprecise, and the problem of inconsistent filtering estimation exist, which makes the unobservable state produce false observability, such as heading angle. Subsequently, the visual-inertial integrated method based on MSCKF extended SR-ISWF [14,15] and unscented Kalman filter (UKF) algorithm [16] were proposed. Other open-source frameworks, such as the extended Kalman filter (EKF)-based ROVIO (2015) fusion framework [17], MSCKF-based R-VIO (2018) framework [18], and OpenVINS (2020) [19] were subsequently proposed. In ROVIO, the sparse direct method is used as the visual front end, and a photometric residual is employed in the EKF update; although these innovations improved localization precision, the ROVIO framework is computationally complex. R-VIO is a robocentric viSLAM framework that reformulates VINS with respect to a moving local frame, whereas OpenVINS is an MSCKF-based framework that improves on MSCKF in terms of landmark estimation; both are highly efficient and lightweight visual-inertial frameworks. Filter-based viSLAM frameworks tend to produce insufficiently precise landmark coordinates, and they are, in theory, highly susceptible to linear errors, which results in suboptimal robustness and precision in complex settings. FGO-based viSLAM frameworks, such as iSAM2 [20], VIM-SLAM [21], OKVIS [22], VINS-Mono [23], and ORB-SLAM3 [24], have seen rapid development since the emergence of the IMU preintegration technique [25] and the refinement and validation of preintegration theory by Forster et al. [26]. The iSAM2 uses Bayes tree to achieve incremental smoothing and mapping, which takes full advantage of the sparsity of the Hessian matrix. VIM-SLAM is an extension algorithm of ORB-SLAM that incorporates inertial sensors. OKVIS processes landmarks using a keyframe-based sliding window, and it incorporates weighted IMU preintegration and reprojection error terms into a factor graph for nonlinear optimization with marginalization. VINS-Mono is a monocular visual-inertial state estimator that utilizes a loop detection module for re-localization and also performs four-degree-of-freedom pose graph optimization to enforce global consistency. This approach improves trajectory fitting and estimator initialization compared to its contemporaries while remaining computationally efficient. ORB-SLAM3 uses maximum-a-posteriori (MAP) estimation from the initialization phase, which allows for rapid initialization, and also provides multi-map data association to facilitate loop detection and BA; these features allow ORB-SLAM3 to operate robustly for long periods with poor visual information.

Although the aforementioned visual front end, loop detection, and initialization methods have greatly improved the robustness of viSLAM, they all share the same flaw: although filter-based visual-inertial odometry is able to propagate and update covariance matrices, the residual introduced to each visual measurement is always a fixed a priori value. For example, R-VIO always assumes that all landmarks have a tracking precision of 1 pixel. Although filtering-based algorithms such as ROVIO and MSCKF update the covariance matrix, the covariance introduced by the newly added sensor's measurement of filter are fixed. Apparently, the initial values of these covariance are independent of the propagation of the covariance. In FGO-based viSLAM, it is necessary to determine the information matrix of the sensor outputs, i.e., the inverse of the covariance matrix, prior to nonlinear optimization. For IMUs, the information matrix can be derived from the propagation of the IMU preintegration, which is initialized using the random walk error

and noise of the IMU. For visual sensors, the magnitude of the covariance matrix elements often depends on the image registration precision of the landmarks; as this parameter roughly describes the reprojection error, it is a measure of the visual sensors' precision. Many open-source viSLAM frameworks employ quantitatively descriptive methods to determine the information matrix of the visual sensors' outputs. For instance, VINS-Mono assumes that landmarks have a registration precision of 1.5 pixels. In ORB-SLAM3, the information matrix of the reprojection error is hierarchically defined by a pyramidal structure, albeit with each layer having fixed values; the first layer has a reprojection error of 1 pixel, and the reprojection errors of the subsequent layers are set according to some ratio. A fixed covariance matrix based on empirical a priori values is a simple and feasible strategy. However, this implies that the visual weights are fixed, which may result in a loss of pose and positioning precision in complex or dynamic scenarios in which the actual sensor outputs may not be compatible with the pre-determined ratio of visual and inertial weights. In the course of our experiments on VINS-Mono test, we found that in some complex scenes, by changing the registration precision of 1.5 pixels to 0.5 pixels, the accuracy increased by about 30%, which led to the study in this paper. Weight ratio tuning problem is common in the field of surveying and mapping. Yang et al. [27] proposed an adaptive filtering algorithm that estimates and amends model and noise characteristics, with robust estimation [28] and variance component estimation used to solve the weight adjustment problem [29]. However, the iterative calculations required for variance component estimation are computationally complex and intractable for real-time applications like SLAM. As a result, this approach is not commonly used in SLAM.

To address the aforementioned problems, we propose a method for visual-inertial fusion based on covariance tuning. During this study, we analyzed the error patterns of visual sensors and IMUs, and studied strategies to adjust the covariance of the visual sensors' outputs. On this basis, we chose an approach in which the unit-weight root-mean-square errors (RMSEs) are computed from a posteriori visual reprojection errors and then used to construct a covariance tuning function for some interval. We also introduced a re-optimization process to the cycle in which the new visual covariance are used for nonlinear optimization to produce new pose solutions. In validation tests conducted using the time-synchronous EuRoC dataset [30], our method outperformed VINS-Mono by 2.39–35.65% in pose precision without closed-loop detection.

2. Methods and Principles

Figure 1 illustrates the architecture of the proposed covariance tuning-based method for resilient sensor fusion in visual-inertial odometry (VIO). This architecture was inspired by VINS-Mono. The architecture of the proposed algorithm consists of five modules: data acquisition and processing, data management, VIO initialization, back end, and re-optimization. The data acquisition and processing module receives and preprocesses the measurement data of the camera and IMU; the data management module manages IMU preintegration and features information in images; VIO initialization estimates gravity vector, velocity, gyroscope bias, and metric scale to ensure the normal operation of the system; the back end performs nonlinear optimization on the sliding window composed of multiple image frames; the re-optimization module uses the covariance tuning model to re-optimize the factor graph constructed at the back end to obtain more accurate navigation results. The architecture of the proposed method retains the same visual front end as VINS-Mono. The raw data inputs of the system include camera-captured images, gyroscope-measured angular velocities, and acceleration values from the accelerometer. The visual front end uses the Good Features To Track (GFTT) proposed by Shi-Tomasi in combination with the Lucas-Kanade optical flow method [31,32], as this strategy provides excellent real-time performance and sensitivity to movement. An IMU preintegration model is used to process the IMU measurements. VIO initialization is performed using the fast VIO initialization method described by Qin and Shen [33]. First, visual measurements are used to perform structure from motion (SfM) for a short period of time. The core work

is to determine the three-dimensional coordinates in the camera coordinate system by triangulating the features detected in the images over a period of time, and use the efficient perspective-n-point (EPnP) [34] method to solve the camera poses. IMU preintegration is then performed to roughly align the visual and IMU measurements to estimate the initial state of the system (i.e., gravity vector, velocity, gyroscope bias, and metric scale). This ensures that the VIO system will operate with high-quality initial states. After the system has been initialized, nonlinear optimization is performed using a keyframe-based sliding window to construct a factor graph model, which consists of factor nodes such as the sliding window's state nodes and visual reprojection factors, IMU preintegration factors, and marginalization factors. Each factor represents the residual of the error equation established by the relative sensor. The marginalization factor is a constraint brought by fixing some old states when the sliding window is updated. It is obtained by the Schur complement operation of the previous sensor factor. The specific process is clearly described in the reference [35]. Nonlinear optimization in the back end is an iterative solution of the error equation in the factor graph. After the first round of nonlinear optimization, the unit-weight RMSEs of the reprojection residuals will be used as inputs for the covariance tuning model, which adjusts the visual covariance. In the subsequent re-optimization process, the new visual covariance matrix is used to reconstruct the factor graph model for a second round of nonlinear optimization, which ultimately solves for the pose and localization of the system. As the proposed method is meant to ensure pose and localization precision in real-time navigation, we only evaluated the forward pose estimation algorithm and did not consider global pose recalibration by loop detection.

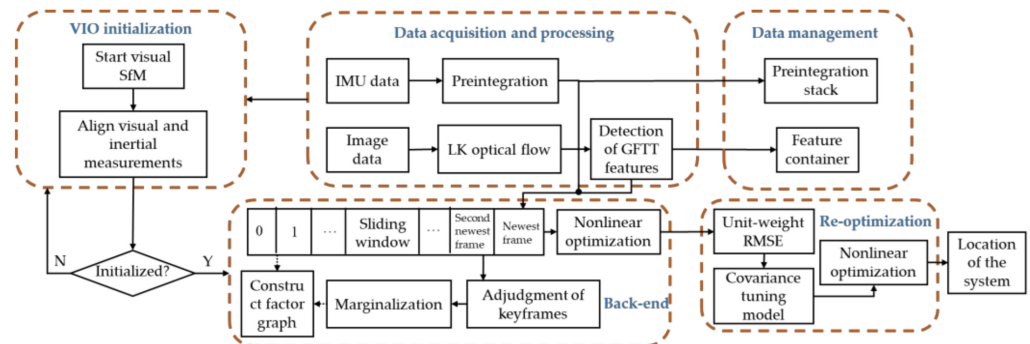


Figure 1. Architecture of the resilient covariance tuning-based visual-inertial fusion algorithm.

Here we briefly describe the variables and frames used in this work. $(\cdot)^b$ represents the IMU frame, which is also referred to as the carrier frame. $(\cdot)^w$ represents the world frame, which is obtained by leveling the initial carrier frame. $(\cdot)^c$ is the camera frame, whose direction is defined by its super- and subscripts. For instance, $(\cdot)_b^w$ represents the transformation of the carrier frame to the world frame, and the superscript $\hat{\cdot}$ is the measured value. Rotation is represented by the rotational matrix \mathbf{R} and rotation quaternion \mathbf{q} , and the parameters being optimized by the nonlinear optimization algorithm are quaternions.

2.1. Construction of the Factor Graph

In the proposed algorithm, the length of the sliding window is marginalized to avoid redundant optimization and improve computational efficiency. The image frames in the sliding window consist of the current image frame and previously found keyframes, and the to-be-optimized state vectors are ascertained based on the keyframes and the current image frame. All of the state vectors that await optimization in the sliding window are defined as follows:

$$\begin{aligned} \mathbf{X} &= [\mathbf{x}_1^T, \mathbf{x}_2^T, \dots, \mathbf{x}_n^T, \mathbf{p}_c^b, \mathbf{q}_c^b, \rho_1, \rho_2, \dots, \rho_m]^T \\ \mathbf{x}_k &= [\mathbf{p}_{b_k}^w, \mathbf{v}_{b_k}^w, \mathbf{q}_{b_k}^w, \mathbf{b}_a, \mathbf{b}_\omega]^T, k \in [0, 1, \dots, n] \end{aligned} \quad (1)$$

In this equation, n and m are the number of to-be-optimized state vectors in the sliding window and the number of observed features, respectively. \mathbf{p}_c^b and \mathbf{q}_c^b are the translation and rotation quaternions between the camera and carrier frames, respectively. ρ is the inverse depth of the feature, which represents its metric scale. $\mathbf{p}_{b_k}^w$ and $\mathbf{q}_{b_k}^w$ are the translation and rotation quaternions, respectively, of the k th-state vector between the camera and carrier frames, and $\mathbf{v}_{b_k}^w$ is the carrier velocity in the world frame that corresponds to the k th-state vector. \mathbf{b}_a and \mathbf{b}_ω are the biases of the IMU's accelerometer and gyroscope, respectively.

The observation model of the algorithm is shown in Figure 2. Within the sliding window, the movement of the carrier is represented by the camera's sampling frequency. As there will be some degree of overlap between adjacent frames, the same feature may be observed in multiple images. As the carrier continues to move, a continuous stream of new frames is selected as keyframe x_k . If the second-newest frame is selected as a keyframe, the oldest keyframe in the sliding window x_1 is marginalized. It should be noted that IMU measurements of some duration will also be taken between adjacent image frames. Figure 3 illustrates a factor graph model that corresponds to a window length of 11 ($n = 11$). The preintegration factor b_k is constructed from IMU measurements that occur between state variables in the sliding window, and the visual reprojection factor c_j is formed by image frames that observe the same feature. As the sliding window moves, a new image frame enters the window as an old state is marginalized (i.e., its state vector is fixed). Nonetheless, the state vectors in the new sliding window are constrained by the state vector of this old keyframe. These constraints include IMU preintegration constraints between the marginalized image frame and the current oldest image frame in the new sliding window, and reprojection error constraints with the marginalized image frame being the first observation image frame and the constraints from the previous marginalization. All of these constraints are added to the nonlinear constraints as the sliding window's marginalization factor.

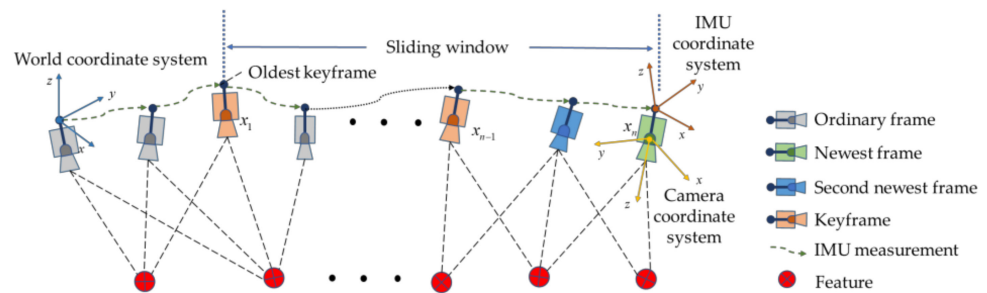


Figure 2. Observation model.

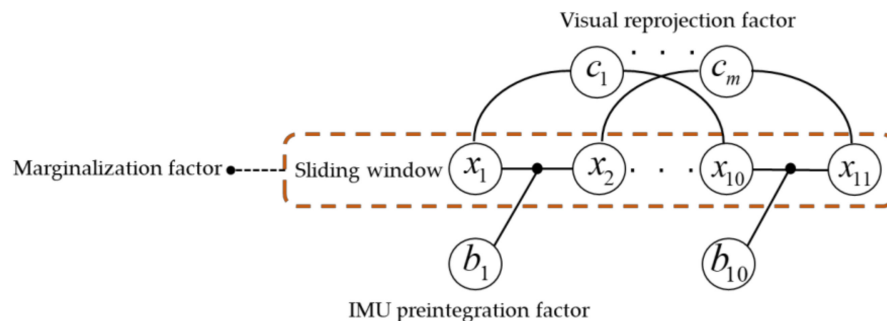


Figure 3. Factor graph model of the algorithm.

The overall cost function of the nonlinear optimization may be constructed from the factor graph model of the sliding window:

$$f(\mathbf{X}) = \min_{\mathbf{X}} \left\{ \|r_p - \mathbf{H}_p \mathbf{X}\|^2 + \sum_{k \in B} \|r_B(\hat{\mathbf{z}}_{b_{k+1}}^{b_k}, \mathbf{X})\|_{\mathbf{P}_{b_{k+1}}^{b_k}}^2 + \sum_{(l,j) \in C} \|r_C(\hat{\mathbf{z}}_l^{c_j}, \mathbf{X})\|_{\mathbf{P}_l^{c_j}}^2 \right\} \quad (2)$$

In this equation, r_p and \mathbf{H}_p are the a priori marginalization information, i.e., the marginalization residual and the Hessian matrix of the marginalization, respectively; $r_B(\hat{\mathbf{z}}_{b_{k+1}}^{b_k}, \mathbf{X})$ is the IMU preintegration factor; $r_C(\hat{\mathbf{z}}_l^{c_j}, \mathbf{X})$ is the visual reprojection factor; $\hat{\mathbf{z}}$ is a measured value; B is a set formed by the series of IMU preintegrations; C is a set formed by the image frame series and all features; $\mathbf{P}_{b_{k+1}}^{b_k}$ is the covariance of the preintegration noise term formed by the k th and $k + 1$ th carrier state variables, and $\mathbf{P}_l^{c_j}$ is the covariance of the noise in visual observations.

When performing iterative optimization using the Newton–Raphson method, the incremental equation that corresponds to the overall cost function (Equation (2)) is:

$$\begin{aligned} & \left(\mathbf{H}_p + \sum_{k \in B} \mathbf{J}_{b_{k+1}}^{b_k T} \mathbf{P}_{b_{k+1}}^{b_k -1} \mathbf{J}_{b_{k+1}}^{b_k} + \sum_{(i,j) \in C} \mathbf{J}_l^{c_j T} \mathbf{P}_l^{c_j -1} \mathbf{J}_l^{c_j} \right) \Delta \mathbf{X} \\ & = \mathbf{b}_p + \sum_{k \in B} \mathbf{J}_{b_{k+1}}^{b_k T} \mathbf{P}_{b_{k+1}}^{b_k -1} r_B + \sum_{(i,j) \in C} \mathbf{J}_l^{c_j T} \mathbf{P}_l^{c_j -1} r_C \end{aligned} \quad (3)$$

In this equation, \mathbf{J} is the Jacobian, \mathbf{b}_p is the marginalization constant, and $\Delta \mathbf{X}$ is the increment of the variable to be optimized.

It may be observed from the incremental equation that the iterative optimization is affected by the covariance matrix. If the visual reprojection covariance $\mathbf{P}_l^{c_j}$ is large, information matrix term $\mathbf{P}_l^{c_j -1}$ will be small, which reduces its effect on the incremental equation, and increases the influence of the observed IMU preintegration term. The opposite applies if $\mathbf{P}_l^{c_j}$ is small. Therefore, the relative magnitudes of the visual and IMU covariance will affect the final result of the optimization.

2.2. Visual Reprojection Factor

Based on the pinhole camera model, we used the definition proposed by Tong et al. [23] for visual residuals to define the reprojection error of the visual front end in terms of a unit sphere. The advantage of this approach is that wide-angle and fish-eye lenses may be modeled using unit rays connected to a unit sphere. Suppose that feature l is first observed in image series i . The reprojection error that occurs when image series j revisits feature l can then be expressed as follows:

$$\begin{aligned} \mathbf{r}_{c_i}^{c_j} &= \left(\hat{\mathbf{p}}_l^{c_j} - \frac{\mathbf{p}_l^{c_j}}{\|\mathbf{p}_l^{c_j}\|} \right) \\ \hat{\mathbf{p}}_l^{c_j} &= \pi_c^{-1} \left(\begin{bmatrix} \hat{u}_l^{c_j} \\ \hat{v}_l^{c_j} \end{bmatrix} \right) \\ \mathbf{p}_l^{c_j} &= \mathbf{R}_b^c \left(\mathbf{R}_w^{b_j} \left(\mathbf{R}_{b_i}^w \left(\mathbf{R}_c^b \frac{1}{\rho_l} \pi_c^{-1} \left(\begin{bmatrix} \hat{u}_l^{c_i} \\ \hat{v}_l^{c_i} \end{bmatrix} \right) + \mathbf{p}_c^b \right) + \mathbf{p}_{b_i}^w - \mathbf{p}_{b_j}^w \right) - \mathbf{p}_c^b \right) \end{aligned} \quad (4)$$

In this equation, $[\hat{u}_l^{c_i}, \hat{v}_l^{c_i}]$ are the image-plane coordinates of l (in pixels) in image series i ; $[\hat{u}_l^{c_j}, \hat{v}_l^{c_j}]$ are the image-plane coordinates of l in image series j ; \mathbf{R}_b^c and \mathbf{P}_c^b are the rotation matrix and translation that relate the camera and carrier frames, respectively, with the direction of rotation/translation indicated by their super- and subscripts. \mathbf{R}_w^b and \mathbf{P}_b^w are the rotation matrix and translation that relate the carrier and world frames, respectively, with the direction of rotation/translation indicated by their super- and subscripts. $\pi_c^{-1}(\cdot)$ is the pixel plane-to-unit vector transformation function, and ρ_l is the inverse depth of the

feature, which represents the scalar relationship between unit sphere vectors and real-world spatial coordinates.

In a pinhole camera, the scaling of the normalization plane to the unit sphere may be embedded in the estimation of λ_l , the inverse depth of feature l . Therefore, $\pi_c^{-1}(\cdot)$ may be expressed as the internal parameter transform from pixel coordinates to normalization plane coordinates:

$$\pi_c^{-1}\left(\begin{bmatrix} u \\ v \end{bmatrix}\right) = \mathbf{K}^{-1} \begin{bmatrix} u \\ v \end{bmatrix} \quad (5)$$

$$\mathbf{K} = \begin{bmatrix} f_x & 0 & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

In this equation, $[u, v]$ are the pixel coordinates of the point, $[u_0, v_0]$ are the coordinates of the principal point, and f_x and f_y are the focal lengths of the image frame in the x and y directions, respectively.

The visual reprojection error factor can be expressed as follows:

$$r_C(\hat{\mathbf{z}}_l^{c_j}, \mathbf{X}) = [\mathbf{e}_1, \mathbf{e}_2]^T \cdot \mathbf{r}_{c_i}^{c_j} \quad (6)$$

In this equation, \mathbf{e}_1 and \mathbf{e}_2 are orthogonal bases that span the unit sphere's tangent plane.

2.3. IMU Preintegration Factor

As the carrier frame is identical to the IMU frame, the IMU's gyroscope and accelerometer models may be expressed as follows:

$$\begin{aligned} \hat{\boldsymbol{\omega}}^b &= \boldsymbol{\omega}^b + \mathbf{b}_\omega + \mathbf{n}_\omega \\ \hat{\mathbf{a}}^b &= \mathbf{a}^b + \mathbf{R}_w^b \mathbf{g}^w + \mathbf{b}_a + \mathbf{n}_a \end{aligned} \quad (7)$$

In these equations, $\hat{\boldsymbol{\omega}}$ and $\hat{\mathbf{a}}$ are the values measured by the gyroscope and accelerometer, respectively, and $\boldsymbol{\omega}$ and \mathbf{a} are the true angular velocity and acceleration, respectively. \mathbf{R}_w^b is the rotation matrix that relates the world frame to the carrier frame, and \mathbf{g}^w is the gravitational acceleration of the world frame. \mathbf{n}_ω and \mathbf{n}_a are the additive noise of the gyroscope and accelerometer, which are modeled as Gaussian white noise, such that $\mathbf{n}_a \sim N(0, \sigma_a^2)$ and $\mathbf{n}_\omega \sim N(0, \sigma_\omega^2)$, respectively. \mathbf{b}_ω and \mathbf{b}_a are the gyroscope and accelerometer biases, respectively, which are modeled as Gaussian random walks $\dot{\mathbf{b}}_a \sim N(0, \sigma_{b_a}^2)$ and $\dot{\mathbf{b}}_\omega \sim N(0, \sigma_{b_\omega}^2)$, respectively.

As the IMU preintegrations are continuous in time, let us suppose that there is some reference time i . The position, velocity, and pose at j may then be expressed as:

$$\begin{aligned} \mathbf{p}_{b_j}^w &= \mathbf{p}_{b_i}^w + \mathbf{v}_{b_i}^w \Delta t - \frac{1}{2} \mathbf{g}^w \Delta t^2 + \mathbf{R}_{b_i}^w \iint_{t \in (i,j)} \left(\mathbf{R}_{b_t}^{b_i} \mathbf{a}^{b_t} \right) \delta t^2 \\ \mathbf{v}_{b_j}^w &= \mathbf{v}_{b_i}^w - \mathbf{g}^w \Delta t + \mathbf{R}_{b_i}^w \int_{t \in (i,j)} \left(\mathbf{R}_{b_t}^{b_i} \mathbf{a}^{b_t} \right) \delta t \\ \mathbf{q}_{b_j}^w &= \mathbf{q}_{b_i}^w \otimes \int_{t \in (i,j)} \mathbf{q}_{b_t}^{b_i} \otimes \begin{bmatrix} 0 \\ \frac{1}{2} \boldsymbol{\omega}^{b_t} \end{bmatrix} \delta t \end{aligned} \quad (8)$$

In these equations, \otimes indicates a quaternion multiplication.

The position, velocity, and pose preintegrations are defined as follows:

$$\begin{aligned}\alpha_{ij} &= \iint_{t \in (i,j)} \left(\mathbf{R}_{b_t}^{b_i} \mathbf{a}^{b_t} \right) \delta t^2 \\ \beta_{ij} &= \int_{t \in (i,j)} \left(\mathbf{R}_{b_t}^{b_i} \mathbf{a}^{b_t} \right) \delta t \\ \gamma_{ij} &= \int_{t \in (i,j)} \mathbf{q}_{b_t}^{b_i} \otimes \left[\begin{array}{c} 0 \\ \frac{1}{2} \boldsymbol{\omega}^{b_t} \end{array} \right] \delta t\end{aligned}\quad (9)$$

In the algorithm, the IMU preintegrations are discretized by the median and then propagated. Based on the two equations above, the IMU preintegration factor and IMU preintegration error may be defined as follows:

$$r_B \left(\hat{\mathbf{z}}_{b_{k+1}}^{b_k}, \mathbf{X} \right) = \begin{bmatrix} \mathbf{r}_p \\ \mathbf{r}_q \\ \mathbf{r}_v \\ \mathbf{r}_{b_a} \\ \mathbf{r}_{b_g} \end{bmatrix}_{15 \times 1} = \begin{bmatrix} \mathbf{q}_{b_i}^{b_j} \left(\mathbf{p}_{b_j}^w - \mathbf{p}_{b_i}^w - \mathbf{v}_{b_i}^w \Delta t + \frac{1}{2} \mathbf{g}^w \Delta t^2 \right) - \alpha_{ij} \\ 2 \left[\gamma_{ij} \otimes \left(\mathbf{q}_{b_i}^w \otimes \mathbf{q}_{b_j}^w \right) \right]_{xyz} \\ \mathbf{q}_{b_i}^{b_j} \left(\mathbf{v}_{b_j}^w - \mathbf{v}_{b_i}^w + \mathbf{g}^w \Delta t \right) - \beta_{ij} \\ \mathbf{b}_{aj} - \mathbf{b}_{ai} \\ \mathbf{b}_{gj} - \mathbf{b}_{gi} \end{bmatrix} \quad (10)$$

where the r terms are residuals. $[\cdot]_{xyz}$ indicates that only the three-dimensional vector from the quaternion's imaginary part will be taken.

2.4. Covariance Tuning Based on Unit-Weight RMSE

In many open-source viSLAM frameworks, the visual covariance is some fixed value that does not change after it is incorporated into the nonlinear optimization. For instance, in VINS-Mono, the covariance factor σ in the covariance matrix $\mathbf{P}_l^{c_j}$ is a fixed value, $1.5/f$, where f is the virtual focal length. Therefore, the reprojection error is assumed to be approximately 1.5 pixels in the image plane, which causes the visual weights to be fixed in any scenario. Although 1.5 pixels is a stable empirical threshold, $\sigma = 1.5/f$ is a poor assumption if any significant change occurs in the sensor devices (e.g., if a different sensor model is used) or in the external environment. Therefore, a resilient and tunable scheme should be used to tune the visual and inertial weights to improve pose and localization precision.

The idea of using covariance tuning function to deal with SLAM problem was inspired by reference [29]. Yang's application of variance component estimation has great implications for the covariance tuning model proposed in this paper. In reference [29], the unit-weight RMSE was used to estimate the variance component. Simulation results show that the variance component estimation method based on unit-weight RMSE can greatly improve the positioning accuracy of multi-sensor fusion. In order to ensure the real-time performance, with more complex establishment of sensor measurement in the field of SLAM, the variance component estimation with multiple iterative optimization will bring computational challenges. Although this method cannot be used directly, the idea of changing the measurement covariance of sensors by the posterior unit-weight RMSE is applicable. As the factor graph problem is a nonlinear least squares problem, if there are m sets of visual measurements, the visual residual after the first round of visual-inertial fusion and optimization may be expressed as follows:

$$\mathbf{V}_C = [r_{u_1}, r_{v_1}, \dots, r_{u_m}, r_{v_m}]_{1 \times 2m}^T \quad (11)$$

The unit-weight RMSE of each visual reprojection residual is given by the following:

$$\begin{aligned}\sigma &= \sqrt{\frac{\mathbf{V}_C^T \mathbf{P}^{-1} \mathbf{V}_C}{n_d - t}} \\ \mathbf{P}^{-1} &= \begin{bmatrix} 1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 1 \end{bmatrix}_{2m \times 2m} \\ n_d &= 2m \\ t &= m + 6\end{aligned}\quad (12)$$

where \mathbf{P}^{-1} is the weight matrix, in unit weights; n_d is the total number of dimensions of the visual reprojection factor, with each observation point providing two constraint equations, and t is the number of variables to be optimized, which includes m inverse depths and pose in six degrees of freedom.

The unit-weight RMSE of the visual reprojections is a reflection of visual quality. A large unit-weight RMSE indicates that the environment or carrier's movements severely affect localization precision and that the visual measurements are of poor quality. On this basis, we propose a covariance-tuning function based on the unit-weight RMSE:

$$\begin{aligned}\mathbf{P}_l^{cj} &= \begin{bmatrix} \sigma'^2 & 0 \\ 0 & \sigma'^2 \end{bmatrix}, j \in (1, 2, \dots, m) \\ \sigma' &= Q(\sigma^2) = \frac{k\sqrt{\sigma^2}}{\lg\left(\frac{1}{\sigma^2}\right)}\end{aligned}\quad (13)$$

where σ' is the new covariance factor, σ is the unit-weight RMSE computed from the a posteriori residuals, k is the confidence factor, and $\lg(\cdot)$ is the log function with base 10.

The inclusion of the log function is meant to increase the function's sensitivity to change, and $k = 2$ corresponds to a confidence level of 95.44%. The function is plotted in Figure 4, which shows that the plot with the log function has larger gradients than the original plot, which increases the sensitivity of the re-optimization process to unit-weight RMSE.

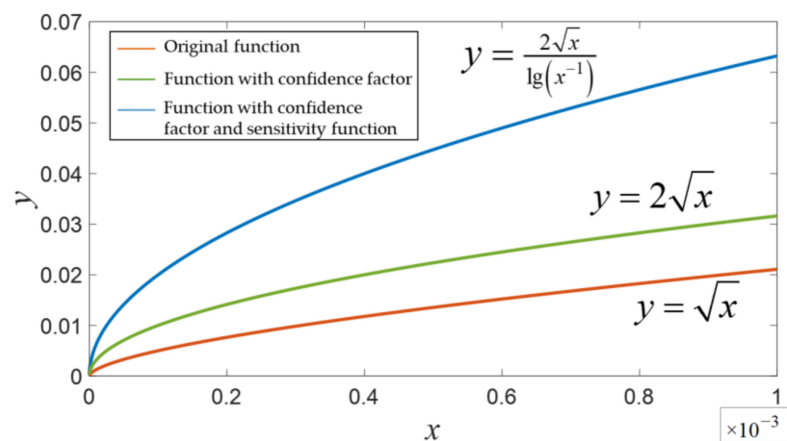


Figure 4. The covariance tuning function.

The inputs of the covariance tuning function are the a posteriori unit-weight RMSEs of the visual reprojection residuals, which reflect the quality of the visual measurements. Using this parameter to adjust the covariance matrix, it becomes possible to accurately assign weights to the visual and inertial measurements and make the best use of the precision of each sensor.

2.5. Re-Optimization

Passing the unit-weight RMSEs to the covariance-tuning function produces a new visual covariance matrix that reflects the level of visual measurement noise after the first round of nonlinear optimization. During re-optimization, the new information matrix is determined according to the level of visual noise, which is then re-substituted into the factor graph for a second round of nonlinear optimization. Creating a new information matrix permits optimization of the ratio of weights between the visual and inertial measurements, which changes the reliance of the system on these measurements.

It should be noted that the weight of the marginalization factor will not be actively altered during re-optimization. As shown in Figure 5, the algorithm will use the new visual covariance matrix for marginalization. As the marginalization factor of the current sliding-window optimization was originally propagated from the marginalization process that occurred after the previous round of sliding-window optimization, the reprojection-residual covariance matrix would have been altered by the algorithm during the processing of the previous image frame. Therefore, when the marginalization factor is propagated to the nonlinear optimization of the current image frame, it is unnecessary to reweigh the visual reprojection factor in the marginalization process.

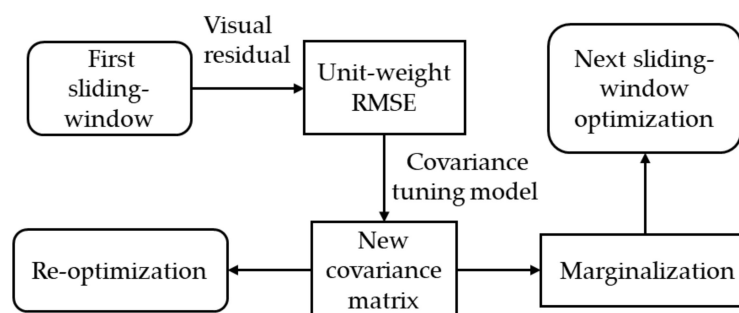


Figure 5. Role of the covariance matrix.

3. Experiments and Analysis

The experimental dataset was the EuRoc dataset, which consists of hardware-synchronized stereo camera data and IMU data from a hexacopter unmanned aerial vehicle (UAV). The ground-true pose values were acquired by a Vicon motion capture system and Leica laser tracker, which have millimeter-level precision. The EuRoc dataset includes two settings: a machine hall and an ordinary room (“Vicon Room”) in ETH Zurich. This dataset also includes 11 image series that were officially classified as “easy,” “medium,” and “difficult,” as shown in Table 1. We use “E” to represent “easy,” “M” to represent “medium,” “D” to represent “difficult,” and “/” is used to separate different data segments in the same series.

Table 1. Sub-datasets of the EuRoc dataset.

Sub-Dataset	Quantity of Segments	Difficulty	Traveled Distance/m
MH01–MH05	5	E/E/M/D/D	80.6/73.5/130.9/91.7/97.6
V101–V103	3	E/M/D	58.6/75.9/79.0
V201–V203	3	E/M/D	36.5/83.2/86.1

3.1. Effectiveness of Covariance Tuning

First, the covariance factor σ for the to-be-optimized visual factor was set to $1.5/f$, without further re-optimization. The V102 dataset from the Vicon Room was analyzed with this setting. The algorithm processes the image by frame, so the abscissa “Frame number” represents the image frame sequence of the data segment V102, and the “Frame number” in the subsequent figures also represents the same meaning. The results are shown in Figure 6. Most of the unit-weight RMSEs of the visual reprojection are within 10^{-4} , and the visual reprojection

factor is rather unstable. This is caused by changes in environmental illumination and the movements of the UAV, which induce large changes in the visual measurements. In this scenario, a fixed visual covariance matrix will result in large losses of precision.

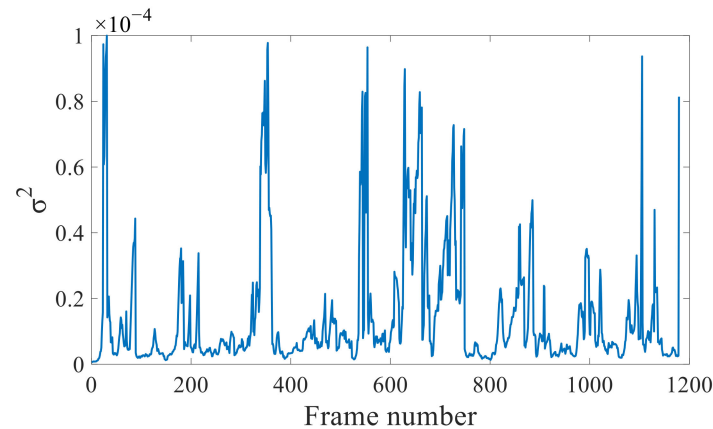


Figure 6. Unit-weight RMSEs of the visual reprojection without re-optimization.

Next, re-optimization was performed using the new visual covariance obtained from the covariance tuning function. The increases in the unit-weight RMSEs of the IMU measurements and visual reprojections after re-optimization are shown in Figure 7a,b. According to Equation (12) and the aforementioned figures, the IMU residuals become larger after re-optimization; although the visual residuals fluctuate somewhat, they become significantly smaller in some frames. Therefore, the confidence level of the visual measurements was increased by the re-optimization of these image frames. Figure 7c,d show that after re-optimization, the weight of the visual reprojection factor σ^{-1} varies from 100 to 3000. This range corresponds to a reprojection error range of 0.2–3.5 pixels in the image plane. Hence, our resilient covariance tuning-based method for visual-inertial fusion is capable of flexible visual weight adjustments.

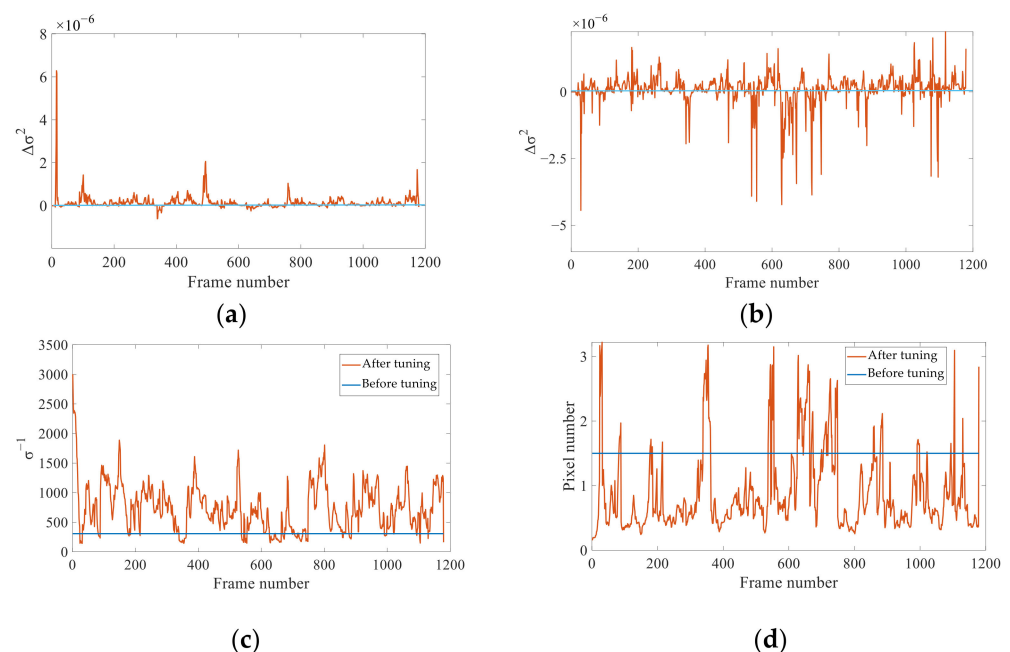


Figure 7. Changes in the relevant parameters after re-optimization. (a) Increase in unit-weight RMSE of the IMU measurements. (b) Increase in unit-weight RMSE of the visual reprojections. (c) Changes in the visual reprojection weights. (d) Manifestation of the visual reprojection weights on the image plane.

Figure 8 compares the numerically downscaled σ^{-1} values shown in Figure 7c to the corresponding IMU unit-weight RMSEs obtained without re-optimization. The tuning of the visual weight factor is correlated with the IMU unit-weight RMSEs, in terms of peak values, and there is also some resemblance in their fluctuations. Therefore, our method can react to changes in IMU precision and apply the appropriate visual weights to make the best use of the sensors' real-time precision.

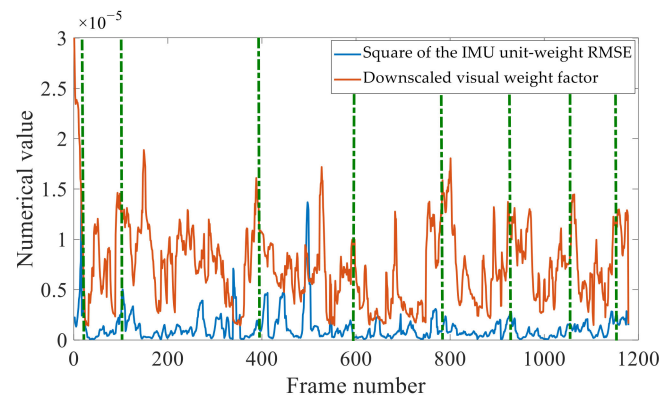


Figure 8. Real-time relationship between visual weight factor adjustments and IMU precision.

3.2. Experimental Analysis of the Resilient Covariance Tuning-Based Visual–Inertial Fusion Algorithm

To test the feasibility of the proposed algorithm, we conducted a validation experiment using the entirety of the EuRoc dataset. We also compared the results to those obtained with VINS-Mono without closed-loop detection. The comparison data with the open-source frameworks R-VIO and OKVIS are presented in Table 2 below. Since the accuracy of VINS-Mono is superior to the other two algorithms in most scenarios, in order to facilitate the observation of trajectory characteristics, this paper only selects VINS-Mono for trajectory comparison. The trajectory comparison between VINS-Mono, R-VIO, and OKVIS is described in detail in reference [18]. The validation and algorithm comparison experiments were performed using the Robot Operating System (ROS) suite with the Ubuntu 18.04.6 LTS operating system. The CPU used was an Intel(R) Xeon(R) Silver 4214 running at 2.20 GHz with 48 threads. The results are shown in Figure 9. Our algorithm produced trajectories that were more similar to the ground-truth trajectories than those produced by VINS-Mono. The trajectories produced by our algorithm were much more stable at bends and straights, and their termination points were also much closer to the real termination points.

Table 2. Comparison between the proposed algorithm and several open-source frameworks in terms of APE. Bold indicates the excellent value of the results in the same row.

Sub-Dataset	Difficulty	Our Algorithm	VINS-Mono	R-VIO	OKVIS	Improvement in Precision Compared to VINS-Mono
MH01	Easy	0.101233	0.157314	0.328240	0.331345	35.65%
MH02	Easy	0.131429	0.178440	0.639892	0.387684	26.35%
MH03	Medium	0.174250	0.195266	0.233700	0.268468	10.76%
MH04	Hard	0.315295	0.439647	1.297599	0.287485	28.28%
MH05	Hard	0.221533	0.303964	0.521598	0.393153	27.12%
V101	Easy	0.079860	0.088830	0.098709	0.095340	10.10%
V102	Medium	0.106666	0.111855	0.134505	0.148746	4.64%
V103	Hard	0.159576	0.187750	0.151586	0.211350	15.01%
V201	Easy	0.074389	0.094752	0.123188	0.099128	21.49%
V202	Medium	0.137455	0.168498	0.169666	0.176457	18.42%
V203	Hard	0.280010	0.286872	0.837517	0.237462	2.39%

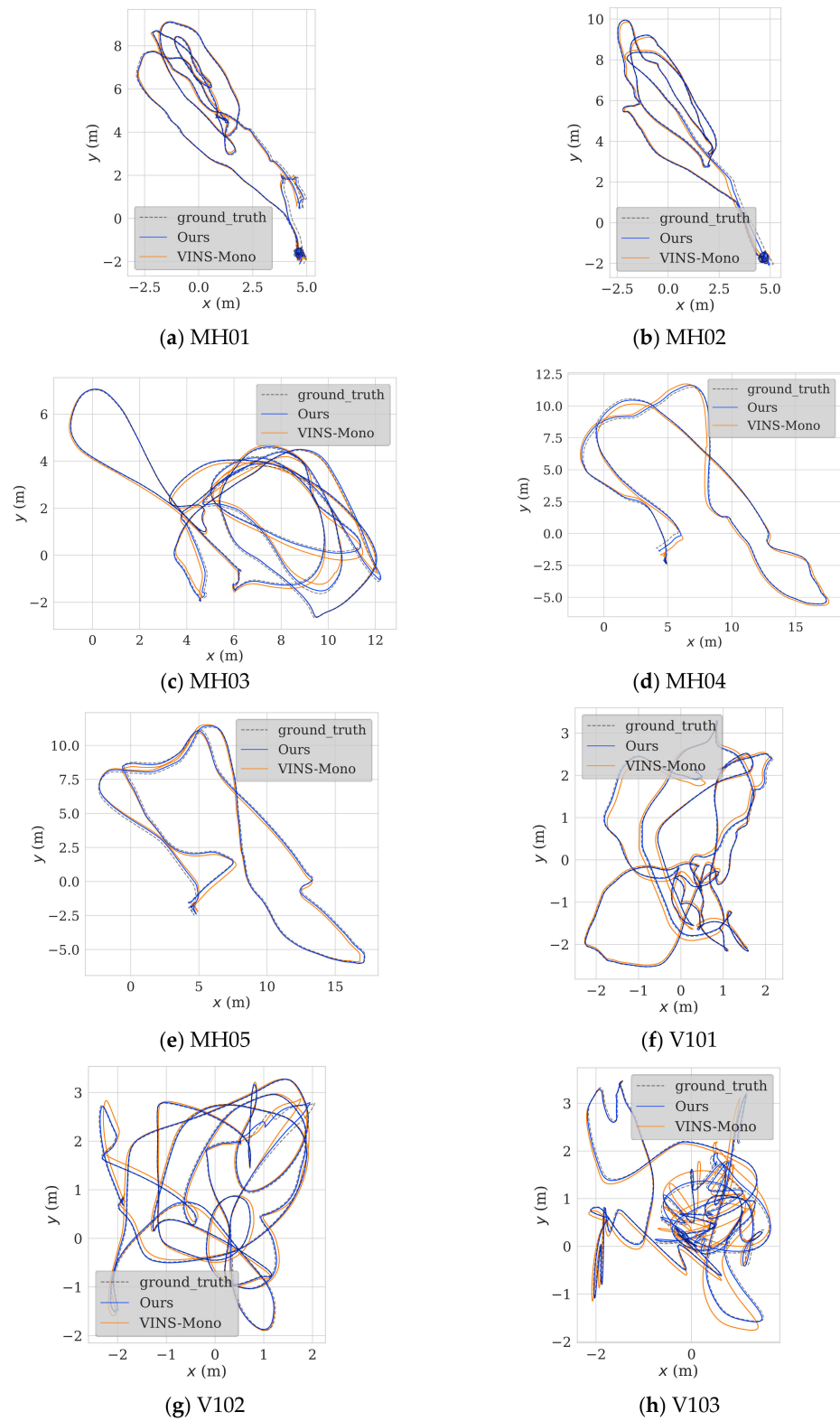


Figure 9. Cont.

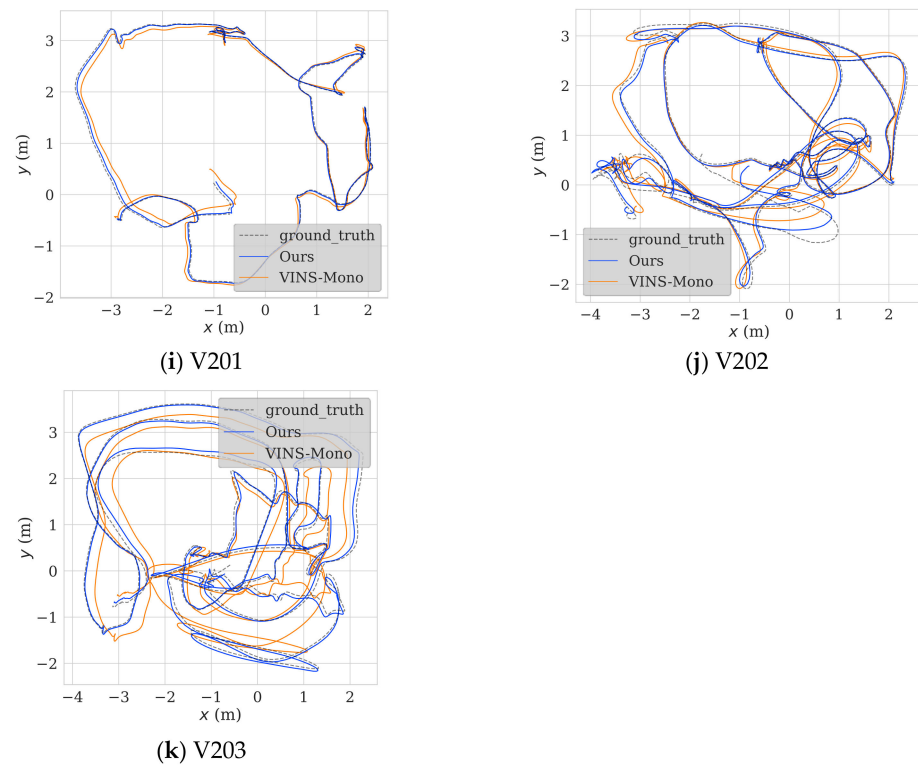


Figure 9. Trajectories computed by the proposed algorithm and VINS-Mono.

The absolute pose precision of the proposed algorithm was evaluated using its root-mean-square absolute pose error (RMS-APE) for the EuRoC dataset, as shown in Figure 10. Although large fluctuations occurred in some spots, the RMS-APE only ranged from 0.074 to 0.31 over all of the EuRoC sub-datasets. Furthermore, the localization precision of the algorithm was better than 30 cm across all of the sub-datasets.

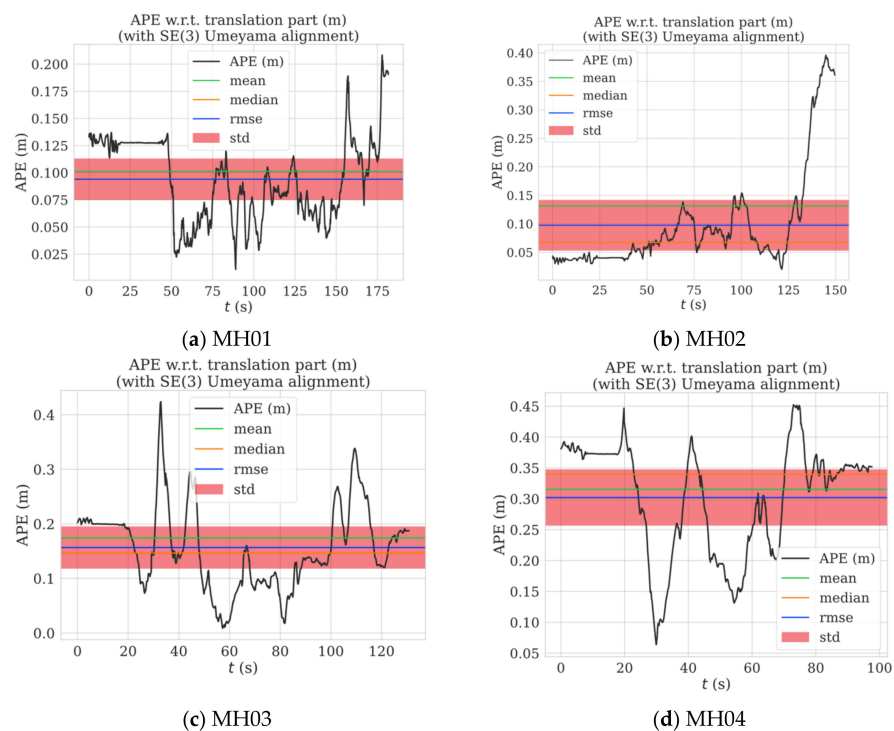


Figure 10. Cont.

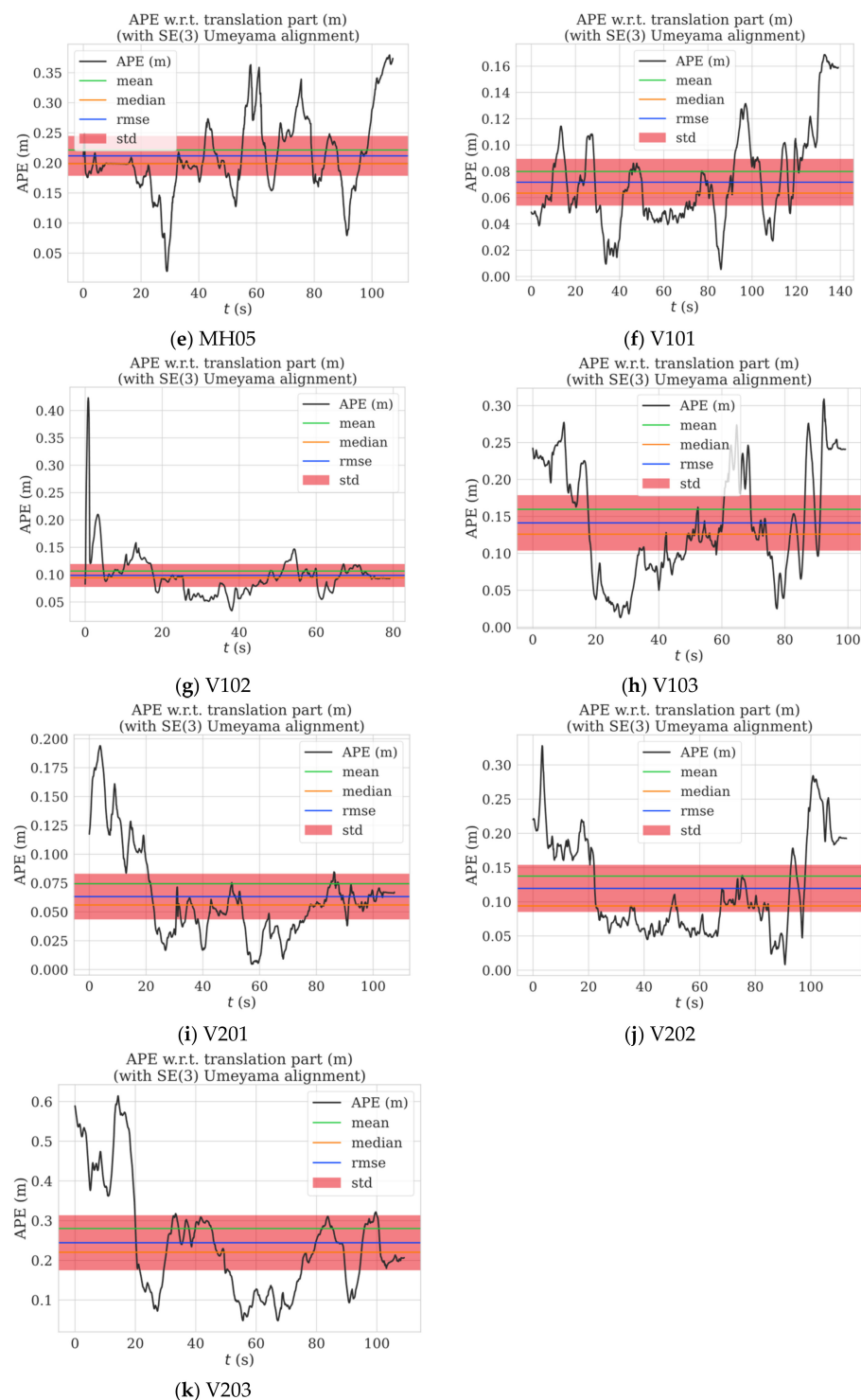


Figure 10. Pose error across the EuRoC dataset.

The precision of the proposed algorithm was then compared to three other open-source viSLAM frameworks (VINS-Mono, R-VIO, and OKVIS), as shown in Table 2. As the R-VIO algorithm has high computational demands during the initialization stage, it required some time for static initialization. Therefore, the R-VIO data in the table are the best results obtained after multiple trials.

Our algorithm significantly outperformed the VINS-Mono, R-VIO, and OKVIS open-source viSLAM frameworks in terms of precision. Furthermore, our algorithm was the most precise of the algorithms compared over most of the EuRoC sub-datasets, and its

precision was better than that of VINS-Mono by 2.39 to 35.65% over the whole EuRoc dataset (18.20% on average).

4. Conclusions

In this study, we constructed a covariance-tuning function based on a posteriori unit-weight RMSEs and proposed a method for visual–inertial fusion based on covariance tuning that improves the localization and pose precision of VIO systems. First, nonlinear optimization was performed based on the factor graph model, and the optimized residuals are used to compute the a posteriori unit-weight RMSEs of the visual reprojection. Next, the a posteriori unit-weight RMSEs were passed to the visual covariance tuning function to create a new visual covariance matrix that is used for re-optimization and marginalization. The proposed algorithm was validated over the entire EuRoc dataset and compared to the VINS-Mono, R-VIO, and OKVIS open-source viSLAM frameworks. The results show that our algorithm maintains a high level of pose and localization precision in settings of varying difficulty. The disadvantage of the proposed algorithm is that the introduction of re-optimization may have an impact on the real-time performance of pose calculation, and the influence of gross error factors is not considered when calculating the unit-weight RMSE.

Author Contributions: Conceptualization, K.L., J.L. and A.W.; methodology, K.L.; software, K.L.; validation, K.L., H.L., X.L. and Z.Y.; formal analysis, K.L. and J.L.; writing—original draft preparation, K.L.; writing—review and editing, J.L. and A.W. All authors have read and agreed to the published version of the manuscript.

Funding: The research described above was supported by the foundation of National Key Laboratory of Electromagnetic Environment (Grant No.6142403210201).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Publicly available datasets, EuRoc, were analyzed in this study. This data can be found here: <https://projects.asl.ethz.ch/datasets/doku.php?id=knavvisualinertialdatasets> (accessed on 17 July 2022).

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

Abbreviations

The following acronyms are used in this manuscript:

viSLAM	Visual–inertial simultaneous localization and mapping
RMSE	Root-mean-square error
PNT	Positioning navigation and time
VINS	Visual–inertial navigation systems
SLAM	Simultaneous localization and mapping
VSLAM	Visual simultaneous localization and mapping system
VO	Visual odometry
LIBVISO	Library for visual odometry
SVO	Semi-direct monocular visual odometry
DSO	Direct sparse odometry
BA	Bundle adjustment
PTAM	Parallel tracking and mapping
IMUs	Inertial measurement units
FGO	Factor graph optimization
MSCKF	Multi-state constraint Kalman filter
UKF	Unscented Kalman filter
EKF	Extended Kalman filter
MAP	Maximum-a-posteriori

RMSEs	Root-mean-square errors
GFTT	Good Features To Track
SfM	Structure from motion
EPnP	efficient perspective-n-point
UAV	Unmanned aerial vehicle
ROS	Robot Operating System
RMS-APE	Root-mean-square absolute pose error

References

1. Yang, Y. Concepts of comprehensive PNT and related key technologies. *Acta Geod. Cartogr. Sin.* **2016**, *45*, 505.
2. Yang, Y. Resilient PNT concept fame. *Acta Geod. Cartogr. Sin.* **2018**, *47*, 893–898.
3. Huang, G. Visual-inertial Navigation: A Concise Review. In Proceedings of the International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019; pp. 9572–9582.
4. 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.
5. Forster, C.; Pizzoli, M.; Scaramuzza, D. SVO: Fast Semi-direct Monocular Visual Odometry. In Proceedings of the International Conference on Robotics and Automation (ICRA), Hong Kong, China, 31 May–7 June 2014; pp. 15–22.
6. Engel, J.; Koltun, V.; Cremers, D. Direct sparse odometry. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *40*, 611–625. [[CrossRef](#)] [[PubMed](#)]
7. 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](#)]
8. Klein, G.; Murray, D. Parallel Tracking and Mapping for Small AR Workspaces. In Proceedings of the 6th IEEE and ACM International Symposium on Mixed and Augmented Reality, Nara, Japan, 13–16 November 2007; pp. 225–234.
9. Strasdat, H.; Montiel, J.M.; Davison, A.J. Real-time monocular SLAM: Why filter? In Proceedings of the International Conference on Robotics and Automation (ICRA), Anchorage, AK, USA, 3–7 May 2010; pp. 2657–2664.
10. 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](#)]
11. Engel, J.; Schöps, T.; Cremers, D. LSD-SLAM: Large-scale direct monocular slam. *Eur. Conf. Comput. Vis.* **2014**, *8690*, 834–849.
12. Shen, S.; Michael, N.; Kumar, V. Tightly-coupled monocular visual-inertial fusion for autonomous flight of rotorcraft MAVs. In Proceedings of the IEEE International Conference on Robotics and Automation, Seattle, WA, USA, 26–30 May 2015; pp. 5303–5310.
13. Mourikis, A.I.; Roulletiotis, 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, Rome, Italy, 10–14 April 2007; pp. 3565–3572.
14. Wu, K.J.; Ahmed, A.M.; Georgiou, G.A.; Roulletiotis, S.I. A Square Root Inverse Filter for Efficient Vision-aided Inertial Navigation on Mobile Devices. In Proceedings of the Robotics: Science and Systems, Rome, Italy, 13–17 July 2015.
15. Paul, M.K.; Wu, K.; Hesch, J.A.; Nerurkar, E.D.; Roulletiotis, S.I. A Comparative Analysis of Tightly-coupled Monocular, Binocular, and Stereo VINS. In Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 29 May–3 June 2017; pp. 165–172.
16. Hu, J.S.; Chen, M.Y. A Sliding-window Visual-IMU Odometer Based on Tri-focal Tensor Geometry. In Proceedings of the International Conference on Robotics and Automation (ICRA), Hong Kong, China, 31 May–7 June 2014; pp. 3963–3968.
17. Bloesch, M.; Omari, S.; Hutter, M.; Siegwart, R. Robust Visual Inertial Odometry Using a Direct EKF-Based Approach. In Proceedings of the IEEE Publications/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, Germany, 28 September–2 October 2015; pp. 298–304.
18. Huai, Z.; Huang, G. Robocentric Visual-Inertial Odometry. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1–5 October 2018; pp. 6319–6326.
19. Geneva, P.; Eckenhoff, K.; Lee, W.; Yang, Y.; Huang, G. OpenVINS: A Research Platform for Visual-Inertial Estimation. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Paris, France, 31 May–31 August 2020; pp. 4666–4672.
20. Kaess, M.; Johannsson, H.; Roberts, R.; Ila, V.; Leonard, J.J.; Dellaert, F. iSAM2: Incremental smoothing and mapping using the Bayes tree. *Int. J. Robot. Res.* **2011**, *31*, 216–235. [[CrossRef](#)]
21. Mur-Artal, R.; Tardos, J.D. Visual-Inertial Monocular SLAM With Map Reuse. *IEEE Robot. Autom. Lett.* **2017**, *2*, 796–803. [[CrossRef](#)]
22. 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](#)]
23. 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](#)]
24. Campos, C.; Elvira, R.; Rodriguez, J.J.G.; Montiel, J.M.M.; Tardos, J.D. ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual-Inertial, and Multimap SLAM. *IEEE Trans. Robot.* **2021**, *37*, 1874–1890. [[CrossRef](#)]
25. Lupton, T.; Sukkarieh, S. Visual-Inertial-Aided Navigation for High-Dynamic Motion in Built Environments Without Initial Conditions. *IEEE Trans. Robot.* **2011**, *28*, 61–76. [[CrossRef](#)]

26. Forster, C.; Carlone, L.; Dellaert, F.; Scaramuzza, D. On-Manifold Preintegration for Real-Time Visual–Inertial Odometry. *IEEE Trans. Robot.* **2016**, *33*, 1–21. [[CrossRef](#)]
27. Yang, Y.; He, H.; Xu, G. Adaptively robust filtering for kinematic geodetic positioning. *J. Geod.* **2001**, *75*, 109–116. [[CrossRef](#)]
28. Yang, Y.X.; Gao, W.G. Integrated navigation based on robust estimation outputs of multi-sensor measurements and adaptive weights of dynamic model information. *Geom. Inf. Sci. Wuhan Univ.* **2004**, *29*, 885–888.
29. Yang, Y.X.; Gao, W.G. Integrated navigation by using variance component estimates of multi-sensor measurements and adaptive weights of dynamic model information. *Acta Geod. Cartogr. Sin.* **2004**, *33*, 22–26.
30. Burri, M.; Nikolic, J.; Gohl, P.; Schneider, T.; Rehder, J.; Omari, S.; Achtelik, M.W.; Siegwart, R. The EuRoC micro aerial vehicle datasets. *Int. J. Robot. Res.* **2016**, *35*, 1157–1163. [[CrossRef](#)]
31. Shi, J.B.; Tomasi, C. Good Features to Track. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 21–23 June 1994; pp. 593–600.
32. Lucas, B.D.; Kanade, T. An Iterative Image Registration Technique with an Application to Stereo Vision. In Proceedings of the 7th International Joint Conferences on Artificial Intelligence, Vancouver, BC, Canada, 24–28 August 1981; pp. 674–679.
33. Qin, T.; Shen, S. Robust Initialization of Monocular Visual–Inertial Estimation on Aerial Robots. In Proceedings of the IEEE/RISJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada, 24–28 September 2017; pp. 4225–4232.
34. Lepetit, V.; Moreno-Noguer, F.; Fua, P. EPnP: An Accurate $O(n)$ Solution to the PnP Problem. *Int. J. Comput. Vis.* **2008**, *81*, 155–166. [[CrossRef](#)]
35. Sibley, G.; Matthies, L.; Sukhatme, G. Sliding window filter with application to planetary landing. *J. Field Robot.* **2010**, *27*, 587–608. [[CrossRef](#)]