

Article

Robust Outlier-Adaptive Filtering for Vision-Aided Inertial Navigation

Kyuman Lee ^{1,*},[†]  and Eric N. Johnson ²

¹ School of Aerospace Engineering, Georgia Institute of Technology, 270 Ferst Drive, Atlanta, GA 30313, USA

² Faculty of Aerospace Engineering, The Pennsylvania State University, 229 Hammond Building, University Park, PA 16802, USA; eric.johnson@psu.edu

* Correspondence: kyumanlee84@gmail.com; Tel.: +1-404-422-3697

† Current address: 55 River Oaks Pl, San Jose, CA 95134, USA.

Received: 7 March 2020; Accepted: 1 April 2020; Published: 4 April 2020



Abstract: With the advent of unmanned aerial vehicles (UAVs), a major area of interest in the research field of UAVs has been vision-aided inertial navigation systems (V-INS). In the front-end of V-INS, image processing extracts information about the surrounding environment and determines features or points of interest. With the extracted vision data and inertial measurement unit (IMU) dead reckoning, the most widely used algorithm for estimating vehicle and feature states in the back-end of V-INS is an extended Kalman filter (EKF). An important assumption of the EKF is Gaussian white noise. In fact, measurement outliers that arise in various realistic conditions are often non-Gaussian. A lack of compensation for unknown noise parameters often leads to a serious impact on the reliability and robustness of these navigation systems. To compensate for uncertainties of the outliers, we require modified versions of the estimator or the incorporation of other techniques into the filter. The main purpose of this paper is to develop accurate and robust V-INS for UAVs, in particular, those for situations pertaining to such unknown outliers. Feature correspondence in image processing front-end rejects vision outliers, and then a statistic test in filtering back-end detects the remaining outliers of the vision data. For frequent outliers occurrence, variational approximation for Bayesian inference derives a way to compute the optimal noise precision matrices of the measurement outliers. The overall process of outlier removal and adaptation is referred to here as “outlier-adaptive filtering”. Even though almost all approaches of V-INS remove outliers by some method, few researchers have treated outlier adaptation in V-INS in much detail. Here, results from flight datasets validate the improved accuracy of V-INS employing the proposed outlier-adaptive filtering framework.

Keywords: V-INS; UAV; EKF; IMU; camera vision; computer vision; image processing; outlier rejection; adaptive filtering; sensor fusion; navigation

1. Introduction

The most widely used algorithms for estimating the states of a dynamic system are a Kalman Filter [1,2] and its nonlinear versions such as an extended Kalman filter (EKF) [3,4]. After the NASA Ames Research Center implemented the Kalman filter into navigation computers to estimate the trajectory of the Apollo program, engineers have developed a myriad of applications of the Kalman filter in navigation system research areas [5]. For example, Magree and Johnson [6] developed a simultaneous localization and mapping (SLAM) architecture with improved numerical stability based on the UD factored EKF, and Song et al. [7] proposed a new EKF system that loosely fuses both absolute state measurements and relative state measurements. Furthermore, Mostafa et al. [8] integrated radar odometry and visual odometry via EKF to help overcome their limitations in navigation. Despite the

development of numerous applications of the Kalman filter in various fields, it suffers from inaccurate estimation when required assumptions fail.

Estimation using a Kalman filter is optimal when process and measurement noise are Gaussian. However, sensor measurements are often corrupted by unmodeled non-Gaussian or heavy-tailed noise. An abnormal value relative to an overall pattern of the nominal Gaussian noise distribution is called an outlier. In other words, in statistics, an outlier is an observation that deviates so much from other observations as to arouse suspicion that it is generated by a different mechanism [9]. Such outliers have many anomalous causes. They arise due to unanticipated changes in system behavior (e.g., temporary sensor failure or transient environmental disturbance) or unmodeled factors (e.g., human errors or unknown characteristics of intrinsic noise). As an example of measurement outliers in many navigation systems, either computer vision data contaminated by outliers or sonar data corrupted by phase noise lead to erroneous measurements. Process outliers also occur by chance. Inertial measurement unit (IMU) dead reckoning and wheel odometry as a proxy often generate inaccurate dynamic models in visual-inertial odometry (VIO) and SLAM algorithms, respectively. Without accounting for outliers, the accuracy of the estimator significantly degrades, and control systems that rely on high-quality estimation can diverge.

1.1. Related Work

1.1.1. State Estimation for Measurements with Outliers

As the performance of the Kalman filter degrades at the presence of measurement outliers, many researchers have investigated other approaches to mitigate the impact of the outliers. Mehra [10] created adaptive filtering with the identification of noise covariance matrices and showed the asymptotic convergence of the estimates towards their true values. Maybeck [11] and Stengel [12] found other noise-adaptive filtering such as covariance matching. However, all of these filters performed only offline and required filter tuning. To estimate parameter values in unknown covariances without the need for manual parameter tuning, Ting et al. [13] used a variational expectation–maximization (EM) framework. That is, they introduced a scalar weight for each observed data sample and modeled the weights to be Gamma distributed random variables. However, it assumed that noise characteristics are homogeneous across all measurements even though sensors have distinct properties. Särkkä and Nummenmaa [14] provided the online learning of the parameters of the measurement noise variance, but to simultaneously track the system states and the levels of sensor noise, they additionally defined a heuristic transition model for the noise parameters. Piché et al. [15] developed Gaussian assumed density filtering and smoothing framework for nonlinear systems using the multivariate Student t -distribution, and Roth et al. [16] included an approximation step for heavy-tailed process noise, but these filters are not applicable in high dimensions. Next, Solin and Särkkä [17] found that the added flexibility of Student- t processes over Gaussian processes robustifies inference in outlier-contaminated noisy data, but they treated only analytic solutions enabled by the noise entanglement.

Recently, Agamennoni et al. developed the outlier-robust Kalman filter (ORKF) [18,19] to obtain the optimal precision matrices of measurement outliers by variational approximation for Bayesian inference [20]. However, this method requires iterations at every time, even when observed data contain no outliers. Graham et al. also established the ℓ_1 -norm filter [21] for both types of sparse outliers. However, the filter might not work for nonlinear systems since they derived the constraint of ℓ_1 -norm optimization based on only linear system equations. Similar to the ORKF, the ℓ_1 -norm filter needs the constrained optimization at all times, even when no additional noise present as outliers. Therefore, these two approaches demand some extensive computational complexity for either iterations or optimization. As outliers do not always arise (i.e., are rare), we reduce such computation cost if a test detects the time when outliers occur. All of the above methods were not validated for complicated systems such as unmanned aerial vehicles or vision-aided inertial navigation [6,22,23] or with sequential measurement updates [24,25].

1.1.2. Outlier Rejection Techniques

One of the primary problems in vision-aided inertial navigation systems (V-INS) is incorrect vision data correspondence or association. Matched features between two different camera views are corrupted by outliers because of image noise, occlusions, and illumination changes that are not modeled by the feature matching techniques. To provide cleaned measurement data to the filter, outlier removal in image processing front-end is essential. One of standard outlier rejection techniques is RANdom SAmple Consensus (RANSAC) [26]. RANSAC is an iterative approach to estimate the parameters of a mathematical model from a set of observed data contaminated by outliers. An underlying assumption is that the data consist of inliers whose distribution is described by some set of the parameters of the model and outliers that do not fit the model. The generated parameters are then verified on the remaining subset of the data, and the model with the highest consensus is a selected solution. In particular, 2-point RANSAC [27,28] is an extended RANSAC-based method for two consecutive views of a camera rigidly mounted on a vehicle platform. Given gyroscopic data from IMU measurements, randomly selected two-feature correspondences hypothesize an ego-motion of the vehicle. This motion constraint discards wrong data associations in the feature matching processes.

For detecting remaining outliers that are not rejected in the image processing front-end, outlier detection tests are required in filtering back-end. Most of the statistical tests [29] that require access to the entire set of data samples for detecting outliers might not be a viable option in real-time applications. For example, the typicality and eccentricity data analysis [30,31] used in [32] is an inadequate measure in V-INS, as computing the means and the variances of each residual of sequential measurements is challenging. In V-INS, the tracking of some measured features is possibly lost due to out of sight, and new feature measurements are coming for initialization.

For the real-time outlier detection of sequential measurements in V-INS, the Mahalanobis gating test [33] is a useful measure based on the analysis of residual and covariance signals at each feature measurement. The approach builds upon each Mahalanobis distance [34] of residuals and compares each value against a threshold given by the quantile of the chi-squared distribution with a certain degree of freedom. The confidence level of the threshold is designated prior to examining the data. Most commonly, the 95% confidence level is used. This hypothesis testing, called goodness of fit, is a commonly used outlier detection method in practice. Because of such suitability of the Mahalanobis gating test to real-time detection in V-INS, this paper combines the test with the ORKF [18,32] to detect and handle measurement outliers in vision-aided estimation problems. Similar to the derivation of update steps for handling measurement outliers in the ORKF, for computing the optimal precision matrices of unmodeled outliers in V-INS, Section 4 will derive feasible update procedures by variational inference. In other words, whenever unexpected outliers appear, the outlier-adaptive filtering in this paper updates and marginalizes measurement outliers to improve the robustness of the navigation systems.

1.2. Summary of Contributions

This paper presents improving the use of outlier removal techniques in image processing front-end and the development of a robust and adaptive state estimation framework for V-INS when frequent outliers occur. For outlier removal in the image processing front-end of V-INS, feature correspondence constitutes the following three steps: tracking, stereo matching, and 2-point RANSAC. To estimate the states of V-INS in which vision measurements still contain remaining outliers, we propose a novel approach that combines a real-time outlier detection technique with an extended version of the ORKF in the filtering back-end of V-INS. Therefore, our approach does not restrict noise at either a constant or Gaussian level in filtering. The testing results of benchmark flight datasets show that our approach leads to greater improvement in accuracy and robustness under severe illumination environments.

Starting from the architecture of the existing vision-aided inertial navigation system, this paper more focuses on contributing to the development of red boxes in Figure 3.

1.3. A Guide to This Document

The remainder of this document contains the following sections. Section 2 introduces background for all of this study. To estimate the states of V-INS in which frequent outliers arise, Section 3 examines outlier rejection techniques in image processing front-end, and Sections 4 and 5 formulate a novel implementation of robust outlier-adaptive filtering. Section 6 shows testing results of this study on benchmark flight datasets. The last section concludes and plans future work.

2. Preliminaries

2.1. The Extended Kalman Filter

The system equations with continuous-time dynamics and a discrete-time sensor are as follows,

$$\dot{x}(t) = f(x(t), \eta(t)) \quad (1)$$

$$y(t_k) = h(x(t_k)) + \zeta(t_k), \quad (2)$$

where $x \in \mathbb{R}^n$ is the state and $y \in \mathbb{R}^m$ a measurement. $f(\cdot)$ and $h(\cdot)$ are the nonlinear dynamic and measurement functions, respectively. Let us assume that these functions are known based on each equation of motion and modeling. To clarify, t denotes continuous time, subscript k represents the k -th time step, and initial condition $x(0) = x_0$ is given. Moreover, let us assume that both propagation and measurements are corrupted by additive zero-mean white Gaussian noise; that is, $\eta(t) \sim \mathcal{N}(0, Q)$ and $\zeta(t_k) \sim \mathcal{N}(0, R)$.

2.1.1. Time Update

To estimate the state variables of the system, we design a hybrid EKF in the following steps. In the propagation step, state estimate $\hat{x} := \mathbb{E}[x]$ and its error-covariance $P := \mathbb{E}[(x - \hat{x})(x - \hat{x})^T]$ are integrated from time $(k-1)^+$ to time k^- with respect to variable τ

$$\hat{x}_k^- = \hat{x}_{k-1}^+ + \int_{t_{k-1}}^{t_k} f(\hat{x}(\tau)) d\tau,$$

where let $\hat{x}_k = \hat{x}(t_k)$. Hat “^” denotes an estimate, and superscripts $-$ and $+$ represent a priori and a posteriori estimates, respectively. Here, for one numerical solution of the ordinary differential equation, we use the Heun’s method [35] that refers to the improved Euler’s method or a similar two-stage Runge–Kutta method. Jacobian A , B , and state transition matrix Φ are defined by

$$\begin{aligned} A_{k-1} &= \left. \frac{\partial f(x)}{\partial x} \right|_{\hat{x}_{k-1}^+}, \quad B_{k-1} = \left. \frac{\partial f(x)}{\partial \eta} \right|_{\hat{x}_{k-1}^+} \\ \Phi_{k-1} &= \exp(A_{k-1} \Delta t_{k-1}) \approx I + A_{k-1} \Delta t_{k-1}, \end{aligned} \quad (3)$$

where $\Delta t_{k-1} = t_k - t_{k-1}$. Letting $P_k = P(t_k)$, the time update of error covariance is

$$P_k^- = \Phi_{k-1} P_{k-1}^+ \Phi_{k-1}^T + B_{k-1} Q B_{k-1}^T \Delta t_{k-1}. \quad (4)$$

2.1.2. Measurement Update

Using actual sensor measurements, the measurement update step of the EKF corrects state estimates and its corresponding error covariance after propagation. Letting $y_k = y(t_k)$, at every time k when y_k is measured,

$$K_k = P_k^- C_k^T (C_k P_k^- C_k^T + R)^{-1} \tag{5}$$

$$\hat{x}_k^+ = \hat{x}_k^- + K_k (y_k - h(\hat{x}_k^-)) \tag{6}$$

$$P_k^+ = P_k^- - K_k C_k P_k^-, \tag{7}$$

where K is called the Kalman gain and Jacobian C is defined as

$$C_k = \left. \frac{\partial h(x)}{\partial x} \right|_{\hat{x}_k^-}.$$

Equation (7) is the Joseph's form [36] of the covariance measurement update, so this form preserves its symmetry and positive definite. For more details such as optimality and derivation, see References [24,37].

2.1.3. Sequential Measurement Update

When myriad measurements are observed at one time, sequential Kalman filtering is useful. In fact, we obtain N measurements, y_1, y_2, \dots, y_N , at time k ; that is, we first measure y_1 , then y_2, \dots , and finally y_N , shown in Figure 1.

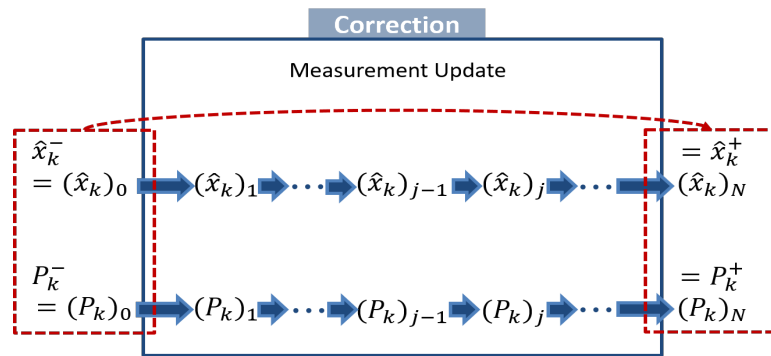


Figure 1. A schematic of the sequential measurement update.

We first initialize a posteriori estimate and covariance after zero measurement is processed; that is, they are equal to the a priori estimate and covariance. For $i = 1, \dots, N$, perform the general measurement update using the i -th measurement. We lastly assign the a posteriori estimate and covariance as $\hat{x}_k^+ \leftarrow (\hat{x}_k)_N$ and $P_k^+ \leftarrow (P_k)_N$. Based on Simon [24]'s proof that the sequential Kalman filtering is equivalent formulation of the standard EKF, the order of updates does not affect overall performance of estimation.

2.2. Models of Vision-Aided Inertial Navigation

2.2.1. Vehicle Model

The nonlinear dynamics of a vehicle is driven by raw micro-electromechanical system (MEMS) IMU sensor data including specific force and angular velocity inputs. The estimated vehicle state is given by

$$\hat{x}_V = \left[{}^i \hat{p}_{b/i}^T \quad {}^i \hat{v}_{b/i}^T \quad \delta \hat{\theta}^T \quad \hat{b}_a^T \quad \hat{b}_\omega^T \right]^T, \tag{8}$$

where $p_{b/i}$, $v_{b/i}$ are the position and velocity of the vehicle with respect to the inertial frame, respectively. $\delta \theta$ is the error quaternion of the attitude of the vehicle, and its more details are explained in [38–40]. b_a and b_ω are the acceleration and gyroscope biases of the IMU, respectively. Left superscript i denotes a vector expressed in the inertial frame. The EKF propagates the vehicle state vector by dead reckoning with data from the IMU. Raw MEMS IMU sensor measurements a_{raw} and ω_{raw} are corrupted by noise and bias as follows,

$$\begin{aligned}
 a_{\text{raw}} &= a_{\text{true}} - \mathcal{T}_{b/i} {}^i g + b_a + \eta_a \\
 \omega_{\text{raw}} &= \omega_{\text{true}} + b_\omega + \eta_\omega \\
 \dot{b}_a &= \eta_{b_a} \tag{9} \\
 \dot{b}_\omega &= \eta_{b_\omega}, \tag{10}
 \end{aligned}$$

where a_{true} and ω_{true} are the true acceleration and angular rate, respectively, and g is the gravitational acceleration in the inertial frame. η_a and η_ω are zero-mean, white, Gaussian noise of the accelerometer and gyroscope measurement, and $\mathcal{T}_{b/i} = \mathcal{T}_{i/b}^T$ denotes the rotation matrix from the inertial frame to the body frame.

η_{b_a} and η_{b_ω} in Equations (9) and (10) are the random walk rate of the acceleration and gyroscope biases. From the works in [40–42], the MEMS accelerometer and gyroscope are subject to flicker noise in the electronics and other components susceptible to random flickering. The flicker noise causes their biases to wander over time. Such bias fluctuations are usually modeled as a random walk. In other words, slow variations in the bias of the IMU sensor are modeled with a “Brownian motion” process, also termed a random walk in discrete-time. In practice, the biases are constrained to be within some range, and thus the random walk model is only a good approximation to the true process for short periods of time.

The vehicle dynamics is given by

$${}^i \dot{\hat{p}}_{b/i} = {}^i \hat{v}_{b/i} \tag{11}$$

$${}^i \hat{v}_{b/i} = \hat{\mathcal{T}}_{i/b} (a_{\text{raw}} - \hat{b}_a) + {}^i g \tag{12}$$

$$\hat{q}_{i/b'} = \frac{1}{2} \mathcal{Q}(\omega_{\text{raw}} - \hat{b}_\omega) \hat{q}_{i/b'} \tag{13}$$

$$\delta \dot{\hat{\theta}} = - \left[(\omega_{\text{raw}} - \hat{b}_\omega) \times \right] \delta \hat{\theta} \tag{14}$$

$$\dot{\hat{b}}_a = 0 \tag{15}$$

$$\dot{\hat{b}}_\omega = 0, \tag{16}$$

where $[a \times]$ is a skew symmetric matrix, and function $\mathcal{Q}(\cdot)$ maps a 3 by 1 vector of the angular velocity into a 4 by 4 matrix [25]. The use of the 4 by 1 quaternion representation in state estimation causes the covariance matrix to become singular, so it requires considerable accounting for the quaternion constraints. To avoid these difficulties, engineers developed the error-state Kalman filter in which 3 by 1 infinitesimal error quaternion $\delta\theta$ is used instead of 4 by 1 quaternion q in the state vector. In other words, we use attitude error quaternion to express the incremental difference between tracked reference body frame b' and actual body frame b for the vehicle. Jacobian matrix $A = \frac{\partial \hat{x}}{\partial x}|_{\hat{x}}$ and $B = \frac{\partial \hat{x}}{\partial \eta}$, where $\eta = [\eta_a^T, \eta_\omega^T, \eta_{b_a}^T, \eta_{b_\omega}^T]^T$, are computed in Appendix A.

2.2.2. Camera Model

An intrinsically calibrated pinhole camera model [27,43] is given by

$$\begin{bmatrix} u_j \\ v_j \end{bmatrix} = y_j = h_j(x) + \zeta_j = \begin{bmatrix} f_u \frac{{}^c X_j}{{}^c Z_j} + \zeta_{u_j} \\ f_v \frac{{}^c Y_j}{{}^c Z_j} + \zeta_{v_j} \end{bmatrix} \tag{17}$$

$$\begin{aligned}
 \begin{bmatrix} {}^c X_j & {}^c Y_j & {}^c Z_j \end{bmatrix}^T &= {}^c p_{f_j/c} = \mathcal{T}_{c/i} \left({}^i p_{f_j/i} - {}^i p_{c/i} \right) \\
 &= \mathcal{T}_{c/b} \mathcal{T}(q_{i/b}) \left({}^i p_{f_j/i} - {}^i p_{b/i} \right) - \mathcal{T}_{c/b} {}^b p_{c/b}, \tag{18}
 \end{aligned}$$

where x is the state vector including the vehicle state and the feature state, and measurement y_j is the j -th feature 2D location on the image plane. f_u and f_v are the horizontal and vertical focal lengths,

respectively, and ζ_u and ζ_v are additive, zero-mean, white, Gaussian noise of the measurement. Vectors $p_{f_j/c}$, $p_{f_j/i}$ are the j -th feature 3D position with respect to the camera frame and the inertial frame, respectively. Extrinsic parameter $\mathcal{T}_{c/b}$ and ${}^b p_{c/b}$ are known and constant. Jacobian matrix $C_j = \frac{\partial y_j}{\partial x} |_{\hat{x}}$ is computed in Appendix A. In addition, from Equation (18), if j -th measurement y_j on an image is a new feature, then ${}^i p_{f_j/i}$ is unknown so need to be initialized. Details of feature initialization are explained in Appendix B.

3. Outlier Rejection in Image Processing Front-End

3.1. Feature Correspondence

In this study, a feature detector using the Features from Accelerated Segment Test (FAST) algorithm [44,45] maintains a minimum number of features in each image. For each new image, a feature extractor using the Kanade–Lucas–Tomasi (KLT) sparse optical flow algorithm [46] tracks the existing features. Even though Paul et al. [47] proved that descriptor-based methods for temporal feature tracking are more accurate than KLT-based methods, as Sun et al. [48] found that descriptor-based methods require much more computing resource with a small gain in accuracy, we employ the KLT optical flow algorithm in the image processing front-end of this study. Next, our stereo matching using a fixed baseline stereo configuration also applies to the KLT optical flow algorithm for saving computational loads compared to other stereo matching approaches. With the matched features, a 2-point RANSAC [26] is applied to remove remaining outliers by utilizing the RANSAC step in the fundamental matrix test [27]. In the scope of this study, we implement the 2-point RANSAC algorithm by simply running one of open source codes.

Similar to [48,49], our outlier rejection is composed of three steps, shown in Figure 2. We assume that features from previous c_1 and c_2 images are outlier-rejected points, where c_1 and c_2 are left and right camera frames of a stereo camera, respectively. The three steps form a closed loop of previous and current frames of left and right cameras. The first step is the stereo matching of tracked features on current c_1 image to c_2 image. The next steps are applying 2-point RANSAC between previous and current images of the left camera and another 2-point RANSAC between previous and current images of the right camera. For steps 2 and 3, stereo-matched features are directly used in each RANSAC.

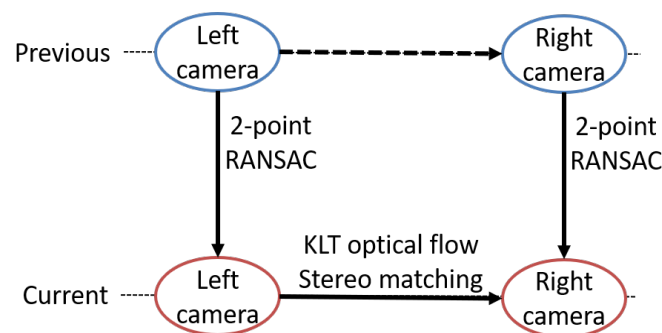


Figure 2. Close loop steps of outlier rejection in image processing front-end.

3.2. Algorithm of Feature Correspondence

Algorithm 1 summarizes the feature correspondence for outlier rejection. For the scope of this paper, the OpenCV library [50] and open source codes of RANSAC are extremely useful and directly applied.

In Algorithm 1, Pyramid is a type of multi-scale signal representation in which an image is subject to repeated smoothing and sub-sampling.

Algorithm 1 Feature Correspondence for Outlier Rejection

Require: Pyramids and outlier-rejected points of previous c_1, c_2 images

```

1: Feature Tracking:
2: function BUILDOPTICALFLOWPYRAMID(current  $c_1$  or  $c_2$  image) ▷ OpenCV [50]
3:   return pyramid of current  $c_1$  or  $c_2$ 
4: end function
5: function PREDICTFEATURES(outlier-rejected points of previous  $c_1, \hat{T}_{\text{curr} \leftarrow \text{prev}}$  of  $c_1$ , Intrinsic  $c_1$ )
6:   return predicted features of current  $c_1$ 
7: end function
8: function CALCOPTICALFLOWPYRLK(pyramids of previous and current  $c_1$ , outlier-rejected points
   of previous  $c_1$ , predicted features of current  $c_1$ ) ▷ OpenCV [50]
9:   return tracked points of previous  $c_1$  and  $c_2$ , tracked features of current  $c_1$ 
10: end function
11: Stereo Matching:
12: function STEREOMATCHING(tracked points of previous  $c_1$  and  $c_2$ , tracked features of current  $c_1$ )
13:   Initialize  $c_2$  points by projecting the tracked features of current  $c_1$  to  $c_2$  using the rotation from
   stereo extrinsic
14:   function CALCOPTICALFLOWPYRLK(pyramid of current  $c_1$  and  $c_2$ , tracked features of current
    $c_1$ , initialized  $c_2$  points) ▷ OpenCV [50]
15:   end function
16:   Further remove outliers based on the essential matrix
17:   return matched points of previous  $c_1$  and  $c_2$ , matched features of current  $c_1$  and  $c_2$ 
18: end function
19: 2-Point RANSAC:
20: function TWOPINTRANSAC(matched points of previous  $c_1$  or  $c_2$ , matched features of current  $c_1$ 
   or  $c_2, \hat{T}_{\text{curr} \leftarrow \text{prev}}$  of  $c_1$  or  $c_2$ , Intrinsic of  $c_1$  or  $c_2$ )
21:   return outlier-rejected points of current  $c_1$  or  $c_2$ 
22: end function
23: Addition of Newly Detected Features:
24: Create a mask to avoid re-detecting existing features
25: function FASTFEATUREDETECTOR(current  $c_1$  image, mask)
26:   return new features on current  $c_1$ 
27: end function
28: function STEREOMATCHING(new features on current  $c_1$ )
29:   return matched new features on current  $c_2$ 
30: end function
31: Group all of outlier-rejected features

```

4. Outlier Adaptation in Filtering Back-End

Even though image processing front-end removes outliers by tracking, stereo matching, and 2-point RANSAC, some outlier features still survive and enter the filter as inputs. This section explains the outlier rejection procedure in filtering back-end.

4.1. Outlier Removal in Feature Initialization

If a measurement is a new feature, our system initializes its 3D position with respect to the inertial frame. In feature initialization, Gaussian–Newton least-squares minimization first estimates the depth of left c_1 camera. If either the estimated depth of the left or right camera is negative, then the solution of the minimization is invalid since features are always in front of both camera frames observing

them. This process of removing features that has the invalid depth is referred to as outlier removal in feature initialization.

4.2. Outlier Detection by Chi-Squared Statistical Test

Before operating navigation systems, we initialize the chi-squared test table with a 95% confidence level. While the systems estimate the state variable, if j -th measurement y_j at time k is the existing feature, then its residual r_j and Jacobian C_j are computed. Next, we proceed a Mahalanobis gating test [33] for residual r_j to detect remaining outliers. In fact, Mahalanobis distance [34] γ_j is a measure of the distance between residual r_j and covariance matrix $S_j = C_j (P_k)_{j-1} C_j^T + R$

$$\gamma_j = r_j^T \left(C_j (P_k)_{j-1} C_j^T + R \right)^{-1} r_j. \quad (19)$$

In the statistic test, we compare γ_j value against a threshold given by the 95-th percentile of the χ^2 distribution with ν_j degree of freedom. Here, ν_j is the number of observations of the j -th feature minus one. If the feature passes the test, the EKF uses residual r_j to process the measurement update.

4.3. Outlier-Adaptive Filtering

Unlike the extended ORKF (EORKF) [32], for a practical estimation approach in V-INS, this study investigates only measurement outliers due to the following reasons. As the measurement update is not the process performed at every time step, the outlier detection by each residual value cannot directly detect the outliers of IMU measurements. Furthermore, in the sequential measurement update, multiple residuals are computed to update at one IMU time stamp. In other words, as only rare observations among feature measurements from one image are corrupted by the remaining outliers, hypothesizing that the outliers come from the IMU may be faulty. Therefore, in the scope of this paper, we handle only measurement outliers.

4.3.1. Student's t -Distribution

Despite the true system with outliers, the classical EKF assumes that each model in the filter is corrupted with additive white Gaussian noise. The levels of the noise are assumed to be constant and encoded by sensor covariance matrices Q and R (i.e., $\eta_k \sim \mathcal{N}(0, Q)$, $(\zeta_j)_k \sim \mathcal{N}(0, R)$). However, as outliers arise in the realistic system, now we do not restrict noise at either a constant or Gaussian level. Instead, their levels vary over time, or noise have heavier tails than the normal distribution as follows,

$$\zeta_j|_k \sim \text{ST}(0, \tilde{R}_j, \nu_j), \quad \text{where} \quad \tilde{R}_j \sim \mathcal{W}^{-1}(\nu_j \Lambda_j, \nu_j), \quad (20)$$

where $\text{ST}(\cdot)$ denotes a Student's t -distribution, and $\nu_k > m - 1$ is degrees of freedom. Covariance matrix \tilde{R}_j follows the inverse-Wishart distribution, denoted as $\mathcal{W}^{-1}(\cdot)$. $\Lambda_j \succ 0$ is $m \times m$ precision matrix.

In Bayesian statistics, the inverse-Wishart distribution is used as the conjugate prior for the covariance matrix of a multivariate normal distribution [20]. The probability density function (pdf) of the inverse-Wishart is

$$p(\tilde{R}_j | \nu_j, \Lambda_j) \propto |\tilde{R}_j|^{-\frac{\nu_j+m+1}{2}} \exp \left[-\frac{\nu_j}{2} \text{tr}(\Lambda_j \tilde{R}_j^{-1}) \right], \quad (21)$$

where $\text{tr}(\cdot)$ denotes the trace of a square matrix in linear algebra. Moreover, in probability and statistics, a Student's t -distribution is any member of a family of continuous probability distributions that arises when estimating the mean of a normally distributed population in situations where the standard deviation of the population is unknown [51]. Whereas a normal distribution describes a full population, a t -distribution describes samples drawn from a full population; thus, the larger the sample, the more the distribution resembles a normal distribution. Indeed, as the degree of freedom goes to infinity, the t -distribution approaches the standard normal distribution. In other words, when the variance of a

normally distributed random variable is unknown and a conjugate prior placed over it that follows an inverse-Wishart distribution, the resulting marginal distribution of the variable follows a Student's t -distribution [52]. Then, the Student- t , a sub-exponential distribution with much heavier tails than the Gaussian, is more prone to producing outlying values that fall far from its mean.

4.3.2. Variational Inference

The purpose of filtering is generally to find the approximations of posterior distributions $p(x_k | y_{1:k})$, where $y_{1:k} = [y_1, y_2, \dots, y_k]$ is the histories of sensor measurements obtained up to time k . For systems with the heavy-tailed noise, we also wish to produce another inference about covariance matrices whose priors follow the inverse-Wishart distribution. Hence, our goal in this section is to find both approximations for posterior distribution $p(x_{1:k}, \tilde{R}_{1:k} | y_{1:k})$ and model evidence $p(y_{1:k})$. Compared to sampling methods, the variational Bayesian method performs approximate posterior inference at low computational cost for a wide range of models [20,52]. In the method, we decompose log marginal probability

$$\ln p(y_{1:k}) = \text{KL}[q \| p] + \mathcal{L}[q], \quad (22)$$

where

$$\text{KL}[q \| p] = \int q(x_{1:k}, \tilde{R}_{1:k}) \ln \frac{q(x_{1:k}, \tilde{R}_{1:k})}{p(x_{1:k}, \tilde{R}_{1:k} | y_{1:k})} \quad (23)$$

$$\mathcal{L}[q] = \int q(x_{1:k}, \tilde{R}_{1:k}) \ln \frac{p(x_{1:k}, \tilde{R}_{1:k}, y_{1:k})}{q(x_{1:k}, \tilde{R}_{1:k})}. \quad (24)$$

p is the true distribution that is intractable for non-Gaussian noise models, and q is a tractable approximate distribution.

In probability theory, a measure of the difference between two probability distributions p and q is the Kullback–Leibler divergence, denoted as $\text{KL}[\cdot]$. If we allow any possible choice for q such as the Gaussian distribution, then lower bound $\mathcal{L}[q]$ is maximum when the KL divergence vanishes; that is, $q(x_{1:k}, \tilde{R}_{1:k}) = p(x_{1:k}, \tilde{R}_{1:k} | y_{1:k})$. To minimize the KL divergence, we seek the member of a restricted family of $q(x_{1:k}, \tilde{R}_{1:k})$. Indeed, maximizing $\mathcal{L}[q]$ is equivalent to minimizing another new KL divergence [52], and thus the minimum occurs when factorized distributions $q(x_{1:k}, \tilde{R}_{1:k}) = q(x_{1:k}) q(\tilde{R}_{1:k})$ and the following Equations (25) and (26) hold,

$$\ln q(x_{1:k}) = \ln p(x_1) + \sum_{t=2}^k \mathbb{E}_{q(\tilde{R}_{1:t})}[\ln p(x_t | x_{t-1})] + \sum_{t=1}^k \mathbb{E}_{q(\tilde{R}_{1:t})}[\ln p(y_t | x_t, \tilde{R}_t)] + \dots \quad (25)$$

$$\ln q(\tilde{R}_k) = \mathbb{E}_{q(x_{1:k})}[\ln p(y_k | x_k, \tilde{R}_k)] + \ln p(\tilde{R}_k) + \dots, \quad (26)$$

where $\mathbb{E}_{q(\cdot)}$ represents the expectation with respect to $q(\cdot)$. With assuming that initial state x_1 is Gaussian, the measurement update with varying noise covariance $\mathbb{E}[\tilde{R}_t^{-1}] = \Lambda_t^{-1}$, which closely resemble the EKF updates, solve Equation (25). Algorithm 2 describes the details of the updates.

Now let us assume that the true priors are IID noise models as the case in this study; that is, $p(\tilde{R}_k)$ follows $\mathcal{W}^{-1}(\nu R, \nu)$ distribution. Then, second term $\ln p(\tilde{R}_k)$ in the right-hand side of Equation (26) is computed using the pdf of the inverse-Wishart distribution in Equation (21) with its prior noise model.

$$\ln p(\tilde{R}_k) = -\frac{\nu + m + 1}{2} \ln |\tilde{R}_k| - \frac{\nu}{2} \text{tr}(R \tilde{R}_k^{-1}). \quad (27)$$

As the term is conjugate prior for Equation (20), the approximations of $q(\tilde{R}_k)$ have same mathematical forms as priors; that is, $q(\tilde{R}_k)$ also follows $\mathcal{W}^{-1}(\tilde{\nu}_k \Lambda_k, \tilde{\nu}_k)$ distribution.

$$\ln q(\tilde{R}_k) = -\frac{\tilde{\nu}_k + m + 1}{2} \ln |\tilde{R}_k| - \frac{\tilde{\nu}_k}{2} \text{tr}(\Lambda_k \tilde{R}_k^{-1}). \quad (28)$$

As $y_t | \{x_t, \tilde{R}_t\} \sim \mathcal{N}(h(x_t), \tilde{R}_t)$,

$$\mathbb{E} [\ln p(y_k | x_k, \tilde{R}_k)] = -\frac{1}{2} \ln |\tilde{R}_k| - \frac{1}{2} \text{tr} \left(\mathbb{E}[\zeta_k \zeta_k^T] \tilde{R}_k^{-1} \right). \quad (29)$$

From Equations (26)–(29), to handle measurement outliers, similar to Agamennoni et al. [18,19]’s derivation, we derive how to compute precision matrix Λ_k of approximate distribution $q(\tilde{R}_k)$ of \tilde{R}_k as follows,

$$\begin{aligned} \tilde{\nu}_k &= 1 + \nu, \quad \tilde{\nu}_k \Lambda_k = \mathbb{E}[\zeta_k \zeta_k^T] + \nu R \\ \Rightarrow \Lambda_k &= \frac{\nu R + \mathbb{E}[\zeta_k \zeta_k^T]}{\nu + 1}, \end{aligned} \quad (30)$$

where each feature from one image is independent and

$$\begin{aligned} (\zeta_j)_k &= (y_j)_k - h(x(t_{\text{img}})) \\ &= (y_j)_k - h((\hat{x}_k)_j + e_j) \\ &\approx (y_j)_k - h((\hat{x}_k)_j) - C_j e_j. \end{aligned}$$

Next, in Equation (30),

$$\begin{aligned} \mathbb{E} [(\zeta_j)_k (\zeta_j)_k^T] &= [(y_j)_k - h((\hat{x}_k)_j)] [(y_j)_k - h((\hat{x}_k)_j)]^T \dots \\ &\quad + C_j \mathbb{E}[e_j e_j^T] C_j^T \\ &= r_j r_j^T + C_j (P_k)_j C_j^T, \end{aligned} \quad (31)$$

where estimation error $e_j = x_k - (\hat{x}_k)_j$ and the Jacobian $C_j = \frac{\partial h_j}{\partial x} |_{(\hat{x}_k)_j}$. In the sequential measurement update, $(\hat{x}_k)_j$ and $(P_k)_j$ are corrected by Kalman gain K_j that is a function of $(\Lambda_j)_k$, so these update steps are coupled. Hence no a closed-form solution exists, and we can only solve iteratively. The purpose of the iteration seems to be similar to that of the online learning of unknown variances of each noise [10]. In addition, similar to Agamennoni et al.’s interpretation [19], the convergence and optimality of the derived update steps for outliers are guaranteed since the variational lower bound is convex with respect to $(\hat{x}_k)_j$, $(P_k)_j$, and $(\Lambda_j)_k$. In particular, as the j -th feature is observed countless times (i.e., $\nu_j \rightarrow \infty$), Λ_j converges to R in the limit of an infinitely precise noise distribution, so the iterative update steps reduce to the standard sequential measurement update of the EKF.

If true state $x(t_{\text{img}})$ is significantly different from its estimate $(\hat{x}_k)_j$, then statistics $\mathbb{E} [(\zeta_j)_k (\zeta_j)_k^T]$ dominates, and $(\Lambda_j)_k$ becomes much larger than R . This ζ_j is regarded as a measurement outlier at time k . As Kalman gain K_j is a function of the inverse of precision matrix $(\Lambda_j)_k$, the larger $(\Lambda_j)_k$ values, the smaller the Kalman gain. Therefore, to deal with situations where measurement outliers occur, the iteration for Equations (30) and (31) corrects the state estimates and its covariance with low weights.

5. Implementation

5.1. Marginalization of Feature States

If measurement outliers often occur, a few numbers of sequential updates in the EKF are proceeded to correct the state estimates. Without a sufficient number of measurement updates, the EKF is not robust and even diverges. Hence, outlier-adaptive filtering introduced in Section 4.3 performs the modified measurement update even when a residual is detected as an outlier. Indeed, to save computation resources, this study operates the outlier-adaptive filtering for only features detected frequently outliers. For implementation, we count how many numbers of features augmented in state variables are detected as outliers. Once updating feature outliers by the outlier-adaptive filtering

approach, we prune the used feature states from the state vector (Figure 3). In addition, similar to that mentioned in Appendix B, to maintain a certain size of the state vector, after the feature initialization, we marginalize the features with the least number of observations among tracked features.

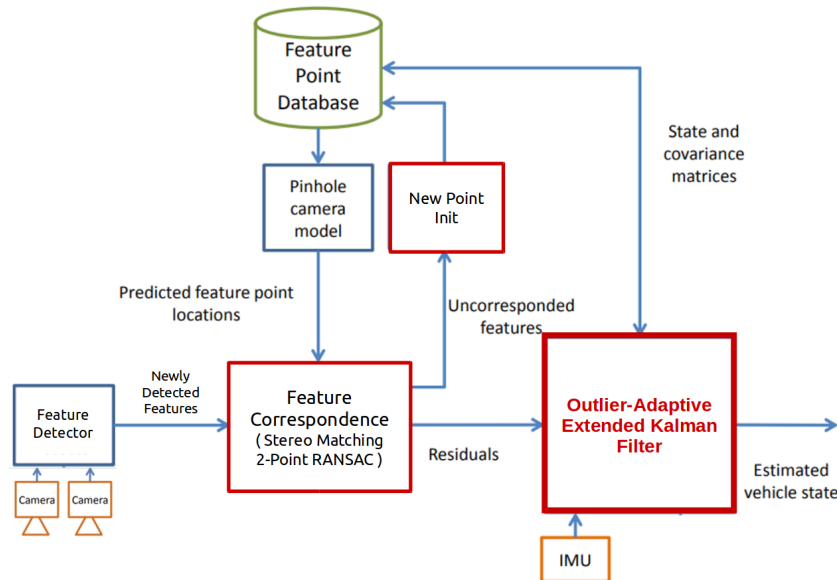


Figure 3. A block diagram of the vision-aided inertial navigation system employing the outlier-adaptive filtering.

5.2. Summarized Algorithm

This section summarizes and describes an implementation of the proposed method. Figure 4 and Algorithm 2 illustrate a flow chart and the pseudocode of the overall process of the outlier-adaptive filtering approach for V-INS, respectively. For the robust outlier-adaptive filter presented in this paper, the blue boxes in Figure 4 are extended from the figure in [25].

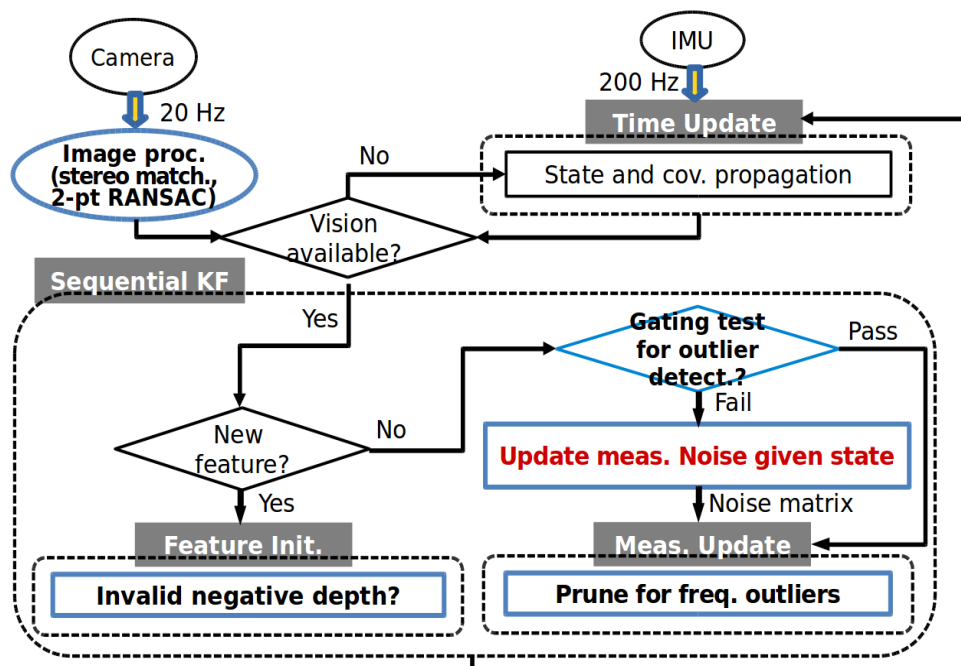


Figure 4. A flow chart of the overall process of the outlier-adaptive filtering.

Algorithm 2 The Outlier-Adaptive Filtering**Require:** \hat{x}_0^+ ($= \hat{x}_{V_0}^+$), P_0^+ , Q , R , χ^2

```

1: for  $k = 1 : T$  do
2:                                     ▷ Image processing front-end in different thread
3:   if new image capture then
4:     Image Processing:                                                         ▷ Algorithm 1
5:       Stereo matching between current images of left camera  $c_1$  and right camera  $c_2$ 
6:       RANSAC between previous and current images of camera  $c_1$ 
7:       RANSAC between previous and current images of camera  $c_2$ 
8:   end if
9:                                     ▷ Filtering back-end in different thread
10:  if new IMU packet arrival then
11:    Time Update:
12:      State prediction
13:    end if
14:  if new vision data packet arrival then
15:    for  $j = 1 : \#$  of observed features  $N$  do
16:      if new feature then
17:        Feature Initialization:
18:          If any depth of  $c_1$  or  $c_2$  is negative,  $j$ -th feature is outlier
19:        else                                                                     ▷ tracked feature
20:          Outlier Gating Test:
21:             $r_j = y_j - h_j(\hat{x}_k)_{j-1}$ 
22:             $S_j = (C_j)_k (P_k)_{j-1} (C_j)_k^T + R$                                      ▷  $(C_j)_k = \frac{\partial h_j}{\partial x} |_{(\hat{x}_k)_{j-1}}$ 
23:             $\gamma = r_j^T S_j^{-1} r_j < \chi_j^2$ 
24:          Measurement Updates:
25:          if outlier detected then
26:             $\tilde{x}_j \leftarrow (\hat{x}_k)_{j-1}$ ,  $\tilde{P}_j \leftarrow (P_k)_{j-1}$                                ▷  $(\hat{x}_k)_0 = \hat{x}_k^-$ ,  $(P_k)_0 = P_k^-$ 
27:            while until converged do
28:              Update measurement noise given the state
29:               $\tilde{r}_j = y_j - h_j(\tilde{x}_j)$ 
30:               $\tilde{C}_j = \frac{\partial h_j}{\partial x} |_{\tilde{x}_j}$ 
31:               $W_j = \tilde{r}_j \tilde{r}_j^T + \tilde{C}_j \tilde{P}_j \tilde{C}_j^T$ 
32:               $\Lambda_j \leftarrow \frac{v_j R + W_j}{v_j + 1}$ 
33:              Update the posteriori state given noise
34:               $\tilde{S}_j = (C_j)_k (P_k)_{j-1} (C_j)_k^T + \Lambda_j$ 
35:               $\tilde{K}_j = (P_k)_{j-1} (C_j)_k^T \tilde{S}_j^{-1}$ 
36:               $\tilde{x}_j \leftarrow (\hat{x}_k)_{j-1} + \tilde{K}_j r_j$ 
37:               $\tilde{P}_j \leftarrow (P_k)_{j-1} - \tilde{K}_j (C_j)_k (P_k)_{j-1}$ 
38:            end while
39:             $(\hat{x}_k)_j = \tilde{x}_j$ ,  $(P_k)_j = \tilde{P}_j$ 
40:          else                                                                     ▷ standard sequential EKF in Section 2
41:             $K_j = (P_k)_{j-1} (C_j)_k^T S_j^{-1}$ 
42:             $(\hat{x}_k)_j = (\hat{x}_k)_{j-1} + K_j r_j$ 
43:             $(P_k)_j = (P_k)_{j-1} - K_j (C_j)_k (P_k)_{j-1}$ 
44:          end if
45:        end if
46:      end for
47:    end if
48:  end for

```

6. Flight Datasets Test Results

To examine the influence of outliers in V-INS and validate the reliability and robustness of the proposed outlier-adaptive approach for navigation systems with outliers, we test one of benchmark flight datasets, the so-called “EuRoC MAV datasets” [53]. The visual-inertial sequences of the datasets were recorded onboard a micro aerial vehicle (MAV) while a pilot manually flew around the indoor motion capture environment. For more details, see Appendix C. To articulate the significance of outliers,

we select two datasets of the bright scene, called “EuRoC V1 Easy”, and motion blur, called “EuRoC V1 Difficult.” As the images in the difficult dataset are dark scene or motion blur, we hypothesize that outliers occur more frequently in the difficult dataset.

The EKF estimates the relative location and orientation from a starting point. As we do not know the exact absolute location of the origin of given datasets, to compare with ground-truth data given in the datasets, we require certain evaluation error metrics such as so-call “absolute trajectory error [54]”. For more details, see Appendix D. The absolute trajectory error as an evaluation error metric yields the following various comparison plots. Figure 5 illustrates the top-down view of the estimated flight trajectory of the difficult dataset.

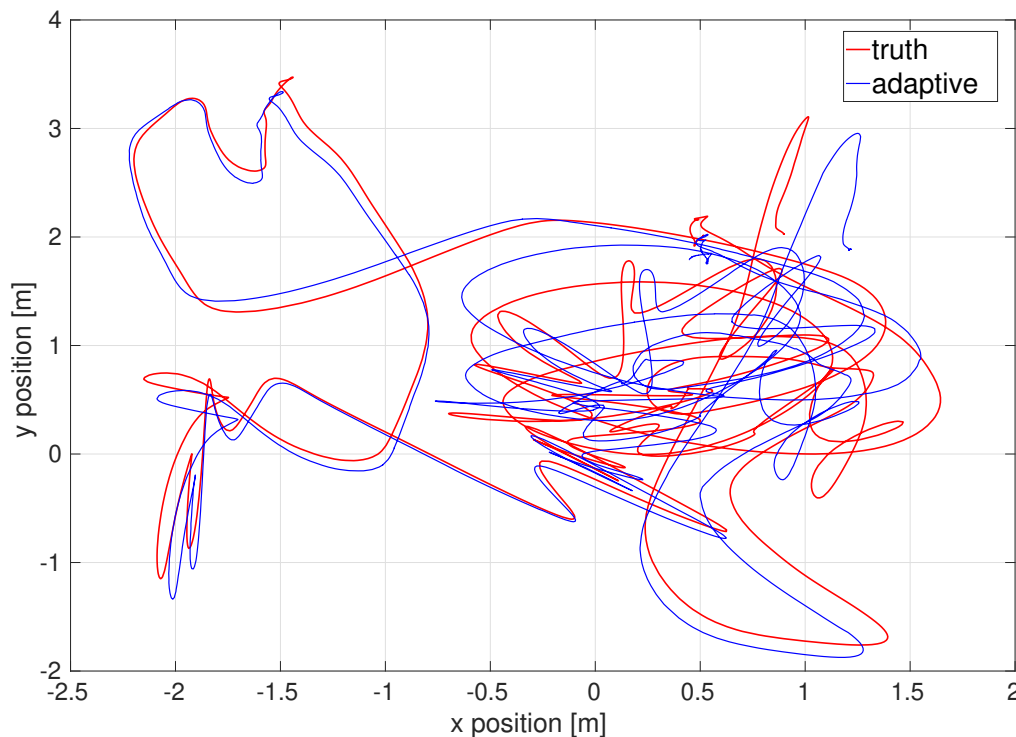


Figure 5. Top-down view of flight trajectory of the EuRoC V1 difficult dataset by the outlier-adaptive filter.

Figures 6 depict the estimated x, y, z position and their respective estimation errors. All the estimation errors are bounded within each standard deviation σ envelope, so the proposed approach is reliable vision-aided inertial navigation under even poor illumination environments. Conceptually, the position error gets locked in, and it tends to increase with the length of the trajectory until new features are being mapped.

Similar to the analysis presented in [25], the adaptive filter is a well-tuned estimator since the performance of doing runs with $\times 3$ or $\times 10$ ($/3$ or $/10$) multiplier on the R matrix used in the filter is worse for all of those, shown in Table 1. That is, the fact that using the multipliers reveals larger root mean square (RMS) estimation errors indicates that our filter is well-tuned.

Table 1. Indication that the outlier-adaptive filter is well-tuned for the EuRoC V1 difficult dataset.

Multiplier on R	$/10$	$/3$	1	$\times 3$	$\times 10$
RMS error [m]	0.9240	0.3801	0.1700	0.5153	0.5610

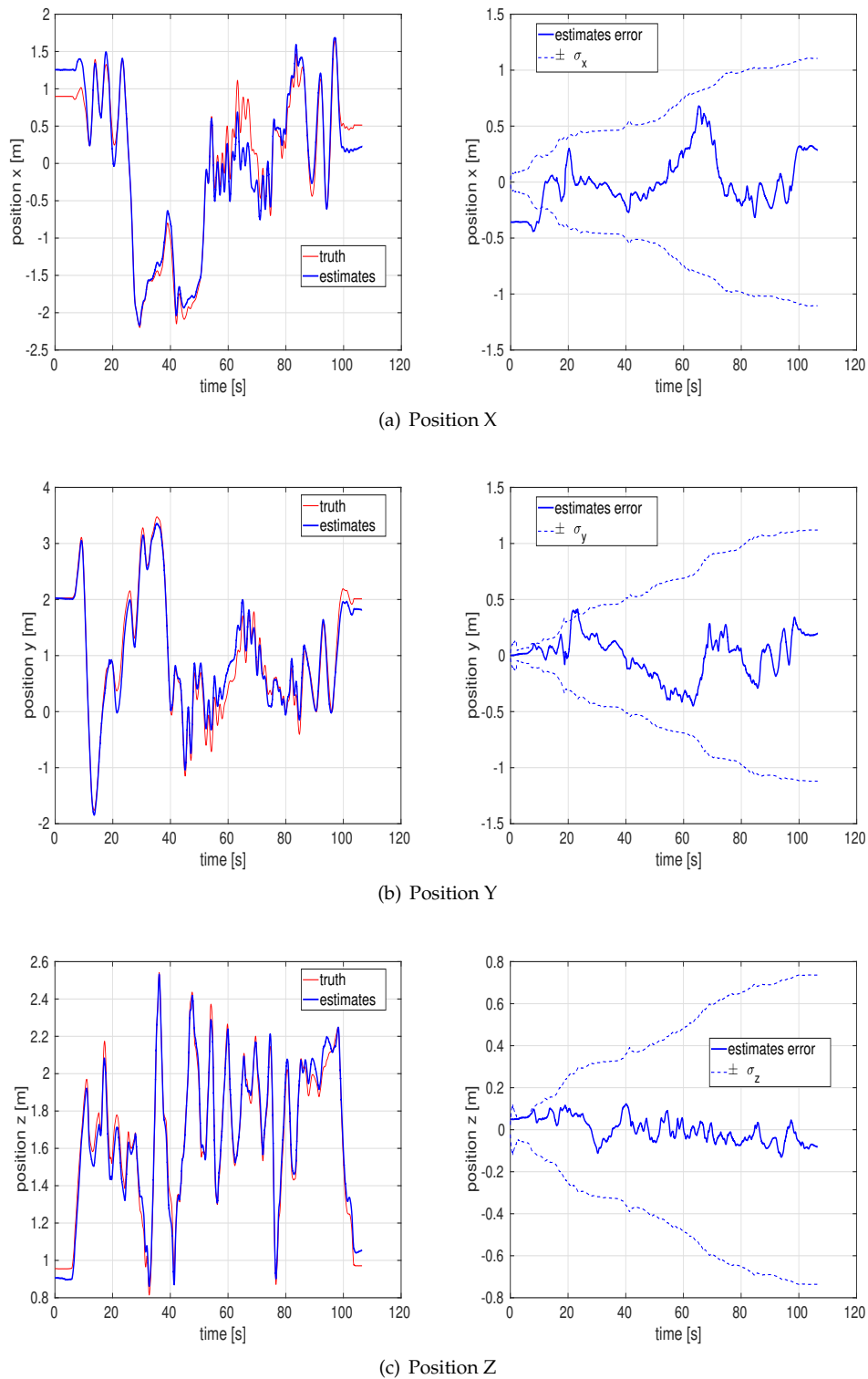


Figure 6. Position and estimation error of the EuRoC V1 difficult dataset by the outlier-adaptive filter.

Figure 7 shows the advantages of the addition of the outlier adaptation proposed in this paper by comparing it with a baseline, the standard EKF in V-INS.

As Lee [25] and other researchers showed that the standard EKF is a basic filter in V-INS, we choose the method as a baseline here. The baseline only rejects outliers whenever the chi-squared test fails, whereas the outlier-adaptive filtering follows all proposed adaptive approaches in Algorithm 2.

Although the iteration in the outlier-adaptive filtering might increase computational resources, it significantly improves the accuracy of estimation. Fortunately, the “while” loop iteration in Algorithm 2 rapidly converges to the optimal noise covariance by twice or three times iterations. For sensitivity analysis, RMS position errors resulting from the baseline and the outlier-adaptive filtering are compiled in Table 2.

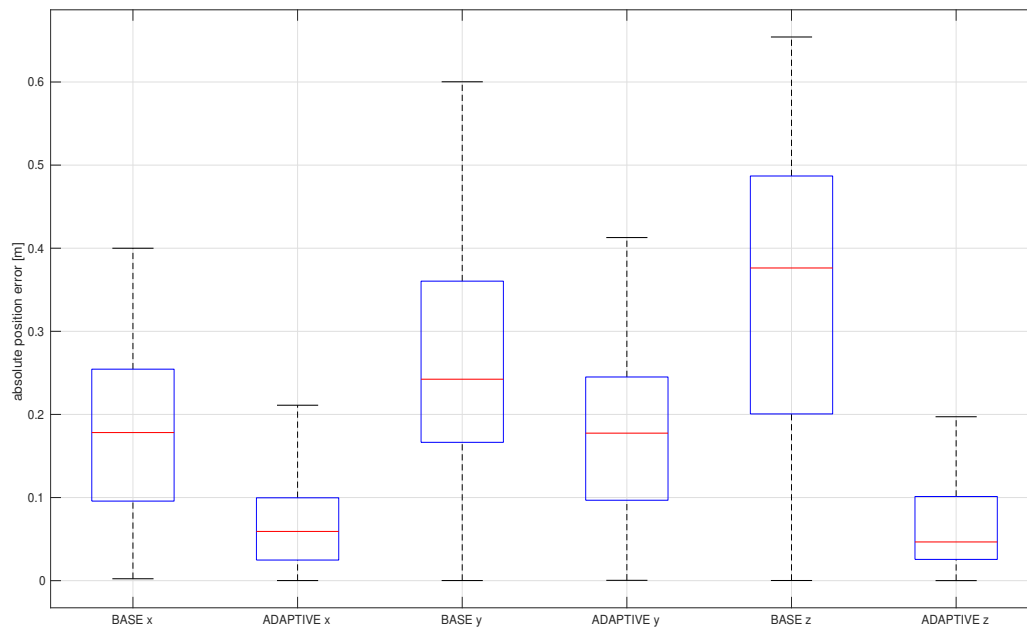


Figure 7. Box plot of absolute estimation error of the position of the EuRoC V1 difficult dataset by the outlier-adaptive filter.

Table 2. Sensitivity analysis in RMS position error [m] of the outlier-adaptive filtering.

Dataset	EuRoC V1 Easy Slow Motion 0.41 m/s, 16.0 deg/s <i>Bright Scene</i>	EuRoC V1 Difficult Fast Motion 0.75 m/s, 35.5 deg/s <i>Motion Blur</i>
Baseline	0.2558	0.3656
Outlier-Adaptive	0.2237	0.2264

Motion blur datasets are more sensitive to outliers as the improvement is larger when applied to those datasets. Although the outlier-adaptive filtering is the best choice for motion blur datasets, we can select an adequate mode depending on computation margin and cost.

Although a number of researchers have investigated V-INS of the EuRoC datasets [55], only a few of them thoroughly has focused on vision measurements with outliers. Table 3 reveals that the proposed estimator, the outlier-adaptive filter, outperforms other state-of-the-art V-INS techniques, called “SVO+MSF [56,57]” and “S-MSCKF [48,58]” in which stereo is available. As SVO+MSF is loosely coupled, its algorithm actually gets diverged.

Table 3. Comparison with other methods in RMS position error [m] of the outlier-adaptive filtering.

Dataset Method	EuRoC V1 Easy <i>Bright Scene</i>	EuRoC V1 Difficult <i>Motion Blur</i>
Outlier-Adaptive Filter	0.2237	0.2264
SVO+MSF [56,57] (loosely coupled)	0.40	×
S-MSCKF [48,58] (stereo-filter)	0.34	0.67

7. Discussion

This paper has presented practical outlier-adaptive filtering for a vision-aided inertial navigation systems (V-INS) and evaluated its performance with flight datasets testing. In other words, this study develops a robust and adaptive state estimation framework for V-INS under frequent outliers occurrence. In the image processing front-end of the framework, we propose the improved utilization of outlier removal techniques. In filtering back-end, for estimating the states of V-INS with measurement outliers, we implement a novel approach of the outlier-robust extended Kalman filter (EKF) to V-INS, for which we derive iterative update steps for computing the precision noise matrices of vision outliers when the Mahalanobis gating test detects remaining outliers.

To validate the accuracy of the proposed approach and compare it with other state-of-the-art V-INS algorithms, we test the performance of V-INS employing the outlier-adaptive filtering algorithm in the realistic benchmark flight datasets. In particular, to show more improvements of our method over the others' approaches, we use the fast motion and motion blur flight datasets. Results from the flight datasets testing show that the novel navigation approach in this study improves the accuracy and reliability of state estimation in V-INS with frequent outliers. Using the outlier-adaptive filtering reduces the root mean square (RMS) error of the estimates and accelerates the robustness of the estimates, especially for the motion blur datasets.

The primary goals of future work are listed as follows. Since an inertial measurement unit (IMU) is also a sensor, it could generate outliers in V-INS. With accounting for the process outliers, the accuracy and robustness of the estimator would be improved. If we distinguish process outliers from IMU sensors with measurement outliers from vision data, the extended outlier-robust EKF [32] may be an impressive and innovate approach for this case. Furthermore, the investigation of color noise in V-INS is another possible future work. One of the required assumptions of the Kalman filter is the whiteness of measurement noise. As an illustration, during sampling and transmission in image processing, color noise that may be originated from a multiplicity of sources could degrade the quality of images [59]. The vibrational effects of camera sensors might also produce colored measurement noise [60]. That is, if the residuals of vision data are correlated with themselves at different timestamp, then colored measurement noise occurs in V-INS. Therefore, the images with color noise would be filtered for ensuring the accuracy of locating landmarks. As modeling noise without additional prior knowledge of the noise statistics is typically difficult, the machine learning techniques-based state estimator for colored noise [61,62] may handle the unknown correlations in V-INS.

This study showcases the approaches using stereo cameras but is also suitable for monocular V-INS and employable to other filter-based V-INS frameworks. We test benchmark flight datasets to validate the reliability and robustness of this study, but additional validating with other flight datasets or real-time flight tests would help to prove its more robustness. In addition, we can operate unmanned aerial vehicles (UAVs) stacked the navigation algorithms in this study and a controller-in-the-loop. The use of the controller-in-the-loop could be a more important validation criteria due to the potential for navigation-controller coupling. The research objectives and contributions presented here will remarkably advance the state-of-the-art techniques of vision-aided inertial navigation for UAVs.

Author Contributions: Conceptualization, K.L. and E.N.J.; Data curation, Formal analysis, Investigation, Methodology, Resources, and Software, K.L.; Supervision, E.N.J.; Validation, Visualization, and Writing-original draft, K.L.; Writing-review & editing, K.L. and E.N.J. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

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

Abbreviations

The following abbreviations are used in this manuscript:

V-INS	Vision-aided Inertial Navigation Systems
VIO	Visual-Inertial Odometry
IMU	Inertial Measurement Unit

MEMS	Micro-ElectroMechanical System
EKF	Extended Kalman Filter
UAV	Unmanned Aerial Vehicle
RANSAC	RANdom SAmple Consensus
FAST	Features from Accelerated Segment Test
KLT	Kanade–Lucas–Tomasi
ST	Student's <i>t</i> -distribution
ROS	Robot Operating System

Nomenclature

x	state	y	measurement
k	discrete time	j	index of measurements
Q	process noise covariance	R	measurement noise covariance
P	error state covariance	r	residual

Appendix A. Jacobians of Models

From Sections 2.2.1 and 2.2.2, Jacobian A, B, and C are

$$A = \begin{bmatrix} 0 & \frac{\partial^i \hat{p}_{b/i}}{\partial^i \hat{b}_{b/i}} & 0 & 0 & 0 \\ 0 & 0 & \frac{\partial^i \hat{b}_{b/i}}{\partial \delta \hat{\theta}} & \frac{\partial^i \hat{b}_{b/i}}{\partial \hat{b}_a} & 0 \\ 0 & 0 & \frac{\partial \delta \hat{\theta}}{\partial \delta \hat{\theta}} & 0 & \frac{\partial \delta \hat{\theta}}{\partial \hat{b}_\omega} \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, \quad B = \begin{bmatrix} 0 & 0 & 0 & 0 \\ \frac{\partial^i \hat{b}_{b/i}}{\partial \eta_a} & 0 & 0 & 0 \\ 0 & \frac{\partial \delta \hat{\theta}}{\partial \eta_\omega} & 0 & 0 \\ 0 & 0 & \frac{\partial \hat{b}_a}{\partial \eta_{b_a}} & 0 \\ 0 & 0 & 0 & \frac{\partial \hat{b}_\omega}{\partial \eta_{b_\omega}} \end{bmatrix} \quad (A1)$$

$$\begin{aligned} \frac{\partial^i \hat{p}_{b/i}}{\partial^i \hat{b}_{b/i}} &= I_{3 \times 3}, & \frac{\partial^i \hat{b}_{b/i}}{\partial \delta \hat{\theta}} &= -\hat{\mathcal{T}}_{i/b'} \left[(a_{\text{raw}} - \hat{b}_a) \times \right], & \frac{\partial^i \hat{b}_{b/i}}{\partial \hat{b}_a} &= -\hat{\mathcal{T}}_{i/b}, \\ \frac{\partial \delta \hat{\theta}}{\partial \delta \hat{\theta}} &= - \left[(\omega_{\text{raw}} - \hat{b}_\omega) \times \right], & \frac{\partial \delta \hat{\theta}}{\partial \hat{b}_\omega} &= -I_{3 \times 3}, \\ \frac{\partial^i \hat{b}_{b/i}}{\partial \eta_a} &= -\hat{\mathcal{T}}_{i/b}, & \frac{\partial \delta \hat{\theta}}{\partial \eta_\omega} &= -I_{3 \times 3}, & \frac{\partial \hat{b}_a}{\partial \eta_{b_a}} &= I_{3 \times 3}, & \frac{\partial \hat{b}_\omega}{\partial \eta_{b_\omega}} &= I_{3 \times 3}, \end{aligned}$$

$$C_j = \left[\frac{\partial y_j}{\partial^i \hat{p}_{b/i}} \quad 0 \quad \frac{\partial y_j}{\partial \delta \hat{\theta}} \quad 0 \quad 0 \quad | \quad 0 \dots \quad \frac{\partial y_j}{\partial^i \hat{p}_{f_j/i}} \quad \dots \quad 0 \right] \quad (A2)$$

$$\frac{\partial y_j}{\partial^i \hat{p}_{b/i}} = \left(\frac{\partial y_j}{\partial^c \hat{p}_{f_j/c}} \right) (-\hat{\mathcal{T}}_{c/i}), \quad \frac{\partial y_j}{\partial^i \hat{p}_{f_j/i}} = -\frac{\partial y_j}{\partial^i \hat{p}_{b/i}}, \quad (A3)$$

$$\frac{\partial y_j}{\partial \delta \hat{\theta}} = \left(\frac{\partial y_j}{\partial^c \hat{p}_{f_j/c}} \right) \mathcal{T}_{c/b} \left[{}^b \hat{p}_{f_j/b} \times \right]$$

$$\left(\frac{\partial y_j}{\partial^c \hat{p}_{f_j/c}} \right) = \frac{1}{{}^c \hat{Z}_j} \begin{bmatrix} f_u & 0 & -f_u \frac{{}^c \hat{X}_j}{{}^c \hat{Z}_j} \\ 0 & f_v & -f_v \frac{{}^c \hat{Y}_j}{{}^c \hat{Z}_j} \end{bmatrix}, \quad (A4)$$

where for more detailed derivations, see reference [25,63].

Appendix B. Feature Initialization

In the first step of the measurement update, we employ Gauss–Newton least-squares minimization [58,64] to estimate feature 3D position ${}^i \hat{p}_{f_j/i}$. To avoid local minima, we apply the inverse depth parameterization of the feature position [65] that is numerically more stable than the Cartesian parameterization.

We assume that the intrinsic and extrinsic parameters of a stereo camera are known and constant values. c_1, c_2 frames are the left and right camera frame of the stereo, respectively. Since the baseline of the stereo is fixed, rotation \mathcal{T}_{c_2/c_1} and translation ${}^{c_2}p_{c_1/c_2}$ between both cameras are constant and known values. Feature coordinates ${}^c[X, Y, Z]^T$ with respect to both cameras are

$${}^{c_2}p_{f_j/c_2} = \mathcal{T}_{c_2/c_1} {}^{c_1}p_{f_j/c_1} + {}^{c_2}p_{c_1/c_2} \quad (\text{A5})$$

$$\begin{bmatrix} {}^{c_2}X_j & {}^{c_2}Y_j & {}^{c_2}Z_j \end{bmatrix}^T = \mathcal{T}_{c_2/c_1} \begin{bmatrix} {}^{c_1}X_j & {}^{c_1}Y_j & {}^{c_1}Z_j \end{bmatrix}^T + {}^{c_2}p_{c_1/c_2} \quad (\text{A6})$$

$$= {}^{c_1}Z_j \left(\mathcal{T}_{c_2/c_1} \begin{bmatrix} \frac{{}^{c_1}X_j}{{}^{c_1}Z_j} \\ \frac{{}^{c_1}Y_j}{{}^{c_1}Z_j} \\ 1 \end{bmatrix} + \frac{1}{{}^{c_1}Z_j} {}^{c_2}p_{c_1/c_2} \right) \quad (\text{A7})$$

$$= {}^{c_1}\hat{Z}_j \left(\mathcal{T}_{c_2/c_1} \begin{bmatrix} \hat{u}_{j,1}/f_{u_1} \\ \hat{v}_{j,1}/f_{v_1} \\ 1 \end{bmatrix} + \frac{1}{{}^{c_1}\hat{Z}_j} {}^{c_2}p_{c_1/c_2} \right). \quad (\text{A8})$$

Simplifying Equation (A8),

$$\begin{bmatrix} {}^{c_2}X_j \\ {}^{c_2}Y_j \\ {}^{c_2}Z_j \end{bmatrix} = {}^{c_1}\hat{Z}_j \left(\begin{bmatrix} m_x \\ m_y \\ m_z \end{bmatrix} + \frac{1}{{}^{c_1}\hat{Z}_j} \begin{bmatrix} tr_x \\ tr_y \\ tr_z \end{bmatrix} \right), \quad (\text{A9})$$

where $m_x, m_y,$ and m_z are scalar functions of given j -th measurement and constant extrinsic rotation matrix. Based on Equation (17), as the measurement equations from the stereo camera are

$$y_j = \begin{bmatrix} u_{j,1} \\ v_{j,1} \\ u_{j,2} \\ v_{j,2} \end{bmatrix} = \begin{bmatrix} f_{u_1} \frac{{}^{c_1}X_j}{{}^{c_1}Z_j} \\ f_{v_1} \frac{{}^{c_1}Y_j}{{}^{c_1}Z_j} \\ f_{u_2} \frac{{}^{c_2}X_j}{{}^{c_2}Z_j} \\ f_{v_2} \frac{{}^{c_2}Y_j}{{}^{c_2}Z_j} \end{bmatrix} + \zeta_j,$$

right c_2 camera measurements are expressed in $Ax = b$ form.

$$\begin{bmatrix} \hat{u}_{j,2}/f_{u_2} \\ \hat{v}_{j,2}/f_{v_2} \end{bmatrix} = \begin{bmatrix} \frac{m_x + (tr_x / {}^{c_1}\hat{Z}_j)}{m_z + (tr_z / {}^{c_1}\hat{Z}_j)} \\ \frac{m_y + (tr_y / {}^{c_1}\hat{Z}_j)}{m_z + (tr_z / {}^{c_1}\hat{Z}_j)} \end{bmatrix} \quad (\text{A10})$$

$$\begin{bmatrix} m_x - (\hat{u}_{j,2}/f_{u_2}) m_z \\ m_y - (\hat{v}_{j,2}/f_{v_2}) m_z \end{bmatrix} {}^{c_1}\hat{Z}_j = \begin{bmatrix} (\hat{u}_{j,2}/f_{u_2}) tr_z - tr_x \\ (\hat{v}_{j,2}/f_{v_2}) tr_z - tr_y \end{bmatrix}, \quad (\text{A11})$$

where let

$$x = {}^{c_1}\hat{Z}_j, \quad A = \begin{bmatrix} m_x - (\hat{u}_{j,2}/f_{u_2}) m_z \\ m_y - (\hat{v}_{j,2}/f_{v_2}) m_z \end{bmatrix}, \quad b = \begin{bmatrix} (\hat{u}_{j,2}/f_{u_2}) tr_z - tr_x \\ (\hat{v}_{j,2}/f_{v_2}) tr_z - tr_y \end{bmatrix}. \quad (\text{A12})$$

Therefore, the Gauss–Newton least-squares minimization estimates depth ${}^{c_1}Z$ of left c_1 camera using the pseudo-inverse of A is

$$Ax = b \Rightarrow (A^T A)x = A^T b \Rightarrow \hat{x} = (A^T A)^{-1} A^T b.$$

If either estimated depth ${}^{c_1}\hat{Z}$ or ${}^{c_2}\hat{Z}$ is negative, the solution of the minimization is invalid since the feature is always in front of both camera frames observing it. By substituting estimated ${}^{c_1}\hat{Z}$ into Equation (A8),

$${}^{c_1}\hat{p}_{f_j/c_1} = \left[(\hat{u}_{j,1}/f_{u_1}) {}^{c_1}\hat{Z} \quad ((\hat{v}_{j,1}/f_{v_1}) {}^{c_1}\hat{Z} \quad {}^{c_1}\hat{Z})^T, \quad (\text{A13})$$

where $\hat{p}_{f_j/c}$ is not related to the pose of the vehicle. Likewise, if a monocular camera is used instead, c_1 is the camera frame in which the feature was observed at the first time, and c_2 is the camera frame at a different time instance.

The j -th feature 3D position with respect to the inertial frame is

$$\begin{aligned} {}^i\hat{p}_{f_j/i} &= \hat{\mathcal{T}}_{i/c_1} {}^{c_1}\hat{p}_{f_j/c_1} + {}^i\hat{p}_{c_1/i} \\ &= \hat{\mathcal{T}}_{i/b} \mathcal{T}_{b/c_1} {}^{c_1}\hat{p}_{f_j/c_1} + \left({}^i\hat{p}_{b/i} + \hat{\mathcal{T}}_{i/b} {}^b p_{c_1/b} \right) \\ &= \hat{\mathcal{T}}_{i/b'} \hat{\mathcal{T}}_{b'/b} \left(\mathcal{T}_{b/c_1} {}^{c_1}\hat{p}_{f_j/c_1} + {}^b p_{c_1/b} \right) + {}^i\hat{p}_{b/i}. \end{aligned} \quad (\text{A14})$$

The new feature is initialized using only one image in which the feature is first observed. Although the new feature is initialized, as it still entails uncertainty, the EKF recursively estimates and updates its 3D position by augmenting into the state vector.

$$\hat{x} = \left[\hat{x}_V^T \quad {}^i\hat{p}_{f_j/i}^T \right]^T, \quad (\text{A15})$$

where \hat{x}_V is the estimated vehicle state vector defined in Equation (8). The overall initialization includes the initial value of the feature state and its error covariance assignment. The error covariance of the new feature are initialized using state augmentation with Jacobian J .

$$\begin{bmatrix} P & * \\ * & * \end{bmatrix} = \begin{bmatrix} I \\ J \end{bmatrix} P \begin{bmatrix} I & J^T \end{bmatrix} = \begin{bmatrix} P & P J^T \\ J P & J P J^T + P_{f_{\text{new}}} \end{bmatrix}, \quad (\text{A16})$$

where Jacobian $J = \left. \frac{\partial p_{f_j/i}}{\partial x} \right|_{\hat{x}}$ is computed as follows,

$$J = \begin{bmatrix} I_{3 \times 3} & 0_{3 \times 3} & -\hat{\mathcal{T}}_{i/b'} \left[\left(\mathcal{T}_{b/c_1} {}^{c_1}\hat{p}_{f_j/c_1} + {}^b p_{c_1/b} \right) \times \right] & 0_{3 \times 6} & 0_{3 \times 3N_f} \end{bmatrix}. \quad (\text{A17})$$

N_f is the number of all features and $P_{f_{\text{new}}}$ is the initial uncertainty of the initialized new feature. The error pertains to measurement noise and the error of the least-squares minimization and so on. In fact, since Montiel et al. [65] validate that the initial uncertainty is coded as Gaussian, the EKF including the feature initialization still holds optimality.

Once initialized, the EKF processes the feature state in the prediction-update loop. In the time update of the EKF, we propagate P by Equation (4).

$$\begin{bmatrix} \Phi & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} P_{vv} & P_{vf} \\ P_{fv} & P_{ff} \end{bmatrix} \begin{bmatrix} \Phi^T & 0 \\ 0 & I \end{bmatrix} + \begin{bmatrix} Q_v & 0 \\ 0 & Q_f \end{bmatrix} = \begin{bmatrix} \Phi P_{vv} \Phi^T + Q_v & \Phi P_{vf} \\ P_{fv} \Phi^T & P_{ff} + Q_f \end{bmatrix}, \quad (\text{A18})$$

where state transition matrix $\Phi \approx I + A \Delta t$. $P_{VV} := \mathbb{E}[(x_V - \hat{x}_V)(x_V - \hat{x}_V)^T]$, P_{ff} is the error covariance of all features, and $P_{Vf} = P_{fV}^T$ represents vehicle-feature correlations. In addition, we assume that surrounding is static, so the dynamics of features $\dot{\hat{p}}_{f_j/i} = 0$. In the measurement update of the EKF, only tracked features are used for the update. For the efficient management of the map database, if the size of the state vector exceeds than the maximum limit, then the feature with the least number of observations is pruned and marginalized.

Appendix C. Experimental Equipment and Environments

Burri et al. [53] provide the benchmark datasets of UAV flying, and Table A1 illustrates the specifications of the sensors used in the datasets.

Table A1. Sensors of EuRoC Datasets

Sensor	Rate	Characteristics
Stereo Images (Aptina MT9V034)	2 × 20 FPS	Global Shutter, WVGA Monochrome
MEMS IMU (ADIS16448)	200 Hz	Instrumentally Calibrated

They obtain the noise model parameters from the IMU at rest and provide them; that is, σ_a , σ_ω , σ_{b_a} , and σ_{b_ω} are known. The intrinsic and extrinsic parameters of both cameras are also given; that is, f_u , f_v , $\mathcal{T}_{c/b}$, and ${}^b p_{c/b}$ are known. The visual-inertial sensor unit is calibrated with *Kalibr* [66] prior to dataset collection. Furthermore, IMU and cameras are hardware time-synchronized such that the middle of the exposure aligned with the IMU measurements.

Visual and inertial data is logged and timestamped onboard the MAV while ground truth is logged in the Vicon 6D motion capture system on the base station. The accuracy of the synchronization between the ground truth and the sensor data is limited by the fact that both sources are recorded on different machines and that the timestamps of the devices are unavailable for the ground-truth system. A maximum likelihood (ML) estimator [11] aligns the data temporally and calibrates the position of the ground-truth coordinate with respect to the body sensor unit. In fact, the ML estimator synchronizes the time-varying temporal offset between the ground truth and the body sensor system. Moreover, it determines the unknown transform between the ground-truth reference frame and the body frame. To obtain the full ML solution, they employ a batch estimator in an offline procedure. Finally, as ground truth, they provide the ML solutions instead of raw data.

Appendix D. Evaluation Error Metric

Sturm et al. [54] provide a set of tools used to preprocess the datasets and evaluate the tracking results. To validate estimation results, we need to evaluate the errors in the estimated trajectory by comparing it with the ground truth. Among various error metrics, two prominent methods are the absolute trajectory error (ATE) and the relative pose error (RPE). In this paper, to evaluate the overall performance of V-INS employing the outlier-adaptive filtering, the ATE measure is selected.

Absolute Trajectory Error (ATE)

The absolute trajectory error directly measures the difference between points of the true and the estimated trajectory. As a preprocessing step, we associate the estimated poses with ground-truth poses using the recorded timestamps. Based on this association, we align the true and the estimated trajectory using the Horn et al. [67]'s closed-form method based on singular value decomposition (SVD). Finally, we compute the differences between each pair of poses and output the mean, median, and standard deviation of these differences.

References

1. Kalman, R.E. A New Approach to Linear Filtering and Prediction Problems. *J. Basic Eng.* **1960**, *82*, 35–45. [\[CrossRef\]](#)
2. Kalman, R.E.; Bucy, R.S. New Results in Linear Filtering and Prediction Theory. *J. Basic Eng.* **1961**, *83*, 95–108. [\[CrossRef\]](#)
3. Anderson, B.D.; Moore, J.B. *Optimal Filtering*; Prentice Hall: Englewood Cliffs, NJ, USA, 1979; Chapters 2–4.
4. Bar-Shalom, Y.; Li, X.R.; Kirubarajan, T. *Estimation with Applications to Tracking and Navigation: Theory Algorithms and Software*; John Wiley & Sons: Hoboken, NJ, USA, 2004; Chapters 2–9.
5. Schmidt, S.F. The Kalman Filter: Its Recognition and Development for Aerospace Applications. *J. Guid. Control Dyn.* **1981**, *4*, 4–7. [\[CrossRef\]](#)
6. Magree, D.; Johnson, E.N. Factored Extended Kalman Filter for Monocular Vision-Aided Inertial Navigation. *J. Aerosp. Inf. Syst.* **2016**, *13*, 475–490. [\[CrossRef\]](#)

7. Song, Y.; Nuske, S.; Scherer, S. A Multi-Sensor Fusion MAV State Estimation from Long-Range Stereo, IMU, GPS and Barometric Sensors. *Sensors* **2017**, *17*, 11. [[CrossRef](#)]
8. Mostafa, M.; Zahran, S.; Moussa, A.; El-Sheimy, N.; Sesay, A. Radar and Visual Odometry Integrated System Aided Navigation for UAVS in GNSS Denied Environment. *Sensors* **2018**, *18*, 2776. [[CrossRef](#)]
9. Hawkins, D.M. *Identification of Outliers*; Springer: Berlin/Heidelberg, Germany, 1980; Chapters 2–5.
10. Mehra, R.K. On the Identification of Variances and Adaptive Kalman Filtering. *IEEE Trans. Autom. Control* **1970**, *15*, 175–184. [[CrossRef](#)]
11. Maybeck, P.S. *Stochastic Models, Estimation, and Control*; Academic Press: Cambridge, MA, USA, 1982; Chapter 10, pp. 68–158.
12. Stengel, R.F. *Optimal Control and Estimation*; Courier Corporation: Chelmsford, MA, USA, 2012; Chapter 4.
13. Ting, J.A.; Theodorou, E.; Schaal, S. A Kalman Filter for Robust Outlier Detection. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), San Diego, CA, USA, 29 October–2 November 2007; pp. 1514–1519. [[CrossRef](#)]
14. Särkkä, S.; Nummenmaa, A. Recursive Noise Adaptive Kalman Filtering by Variational Bayesian Approximations. *IEEE Trans. Autom. Control* **2009**, *54*, 596–600. [[CrossRef](#)]
15. Piché, R.; Särkkä, S.; Hartikainen, J. Recursive Outlier-Robust Filtering and Smoothing for Nonlinear Systems Using the Multivariate Student-*t* Distribution. In Proceedings of the IEEE International Workshop on Machine Learning for Signal Processing (MLSP), Santander, Spain, 23–26 September 2012; pp. 1–6. [[CrossRef](#)]
16. Roth, M.; Özkan, E.; Gustafsson, F. A Student's *t* Filter for Heavy Tailed Process and Measurement Noise. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Vancouver, BC, Canada, 26–31 May 2013; pp. 5770–5774. [[CrossRef](#)]
17. Solin, A.; Särkkä, S. State Space Methods for Efficient Inference in Student-*t* Process Regression. In Proceedings of the International Conference on Artificial Intelligence and Statistics, San Diego, CA, USA, 9–12 May 2015; pp. 885–893.
18. Agamennoni, G.; Nieto, J.I.; Nebot, E.M. An Outlier-Robust Kalman Filter. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Shanghai, China, 9–13 May 2011; pp. 1551–1558. [[CrossRef](#)]
19. Agamennoni, G.; Nieto, J.I.; Nebot, E.M. Approximate Inference in State-Space Models with Heavy-Tailed Noise. *IEEE Trans. Signal Process.* **2012**, *60*, 5024–5037. [[CrossRef](#)]
20. Beal, M.J. Variational Algorithms for Approximate Bayesian Inference. Ph.D. Thesis, University of London, London, UK, 2003.
21. Graham, M.C.; How, J.P.; Gustafson, D.E. Robust State Estimation with Sparse Outliers. *J. Guid. Control Dyn.* **2015**, *38*, 1229–1240. [[CrossRef](#)]
22. Wu, A.D.; Johnson, E.N.; Kaess, M.; Dellaert, F.; Chowdhary, G. Autonomous Flight in GPS-Denied Environments Using Monocular Vision and Inertial Sensors. *J. Aerosp. Inf. Syst.* **2013**, *10*, 172–186. [[CrossRef](#)]
23. Nakamura, T.; Haveland, S.T.; Bershady, D.; Johnson, E.N. Vision Sensor Fusion for Autonomous Landing. In Proceedings of the AIAA Information Systems-AIAA Infotech@ Aerospace, Grapevine, TX, USA, 9–13 January 2017; p. 0674. [[CrossRef](#)]
24. Simon, D. *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches*; John Wiley & Sons: Hoboken, NJ, USA, 2006; Chapters 2–13.
25. Lee, K. Adaptive Filtering for Vision-Aided Inertial Navigation. Ph.D. Thesis, Georgia Institute of Technology, Atlanta, GA, USA, 2018.
26. Fischler, M.A.; Bolles, R.C. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Commun. ACM* **1981**, *24*, 381–395. [[CrossRef](#)]
27. Hartley, R.; Zisserman, A. *Multiple View Geometry in Computer Vision*; Cambridge University Press: Cambridge, UK, 2003; pp. 22–236.
28. Troiani, C.; Martinelli, A.; Laugier, C.; Scaramuzza, D. 2-Point-Based Outlier Rejection for Camera-IMU Systems with Applications to Micro Aerial Vehicles. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China, 31 May–7 June 2014; pp. 5530–5536. [[CrossRef](#)]
29. Kriegel, H.P.; Kröger, P.; Zimek, A. Outlier Detection Techniques. *Tutor. KDD* **2010**, *10*, 1–76.

30. Angelov, P. Anomaly Detection Based on Eccentricity Analysis. In Proceedings of the IEEE Symposium on Evolving and Autonomous Learning Systems (EALS), Orlando, FL, USA, 9–12 December 2014; pp. 1–8. [[CrossRef](#)]
31. Costa, B.S.J.; Bezerra, C.G.; Guedes, L.A.; Angelov, P.P. Online Fault Detection Based on Typicality and Eccentricity Data Analytics. In Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN), Killarney, Ireland, 12–17 July 2015; pp. 1–6. [[CrossRef](#)]
32. Lee, K.; Johnson, E.N. Robust State Estimation and Online Outlier Detection Using Eccentricity Analysis. In Proceedings of the IEEE Conference on Control Technology and Applications (CCTA), Mauna Lani, HI, USA, 27–30 August 2017; pp. 1350–1355. [[CrossRef](#)]
33. Chang, G. Robust Kalman Filtering Based on Mahalanobis Distance as Outlier Judging Criterion. *J. Geod.* **2014**, *88*, 391–401. [[CrossRef](#)]
34. Mahalanobis, P.C. *On the Generalized Distance in Statistics*; National Institute of Science of India: Brahmapur, India, 1936.
35. Atkinson, K.E. *An Introduction to Numerical Analysis*; John Wiley & Sons: Hoboken, NJ, USA, 2008; Chapter 5–6.
36. Gelb, A. *Applied Optimal Estimation*; MIT Press: Cambridge, MA, USA, 1974; Chapters 4, 6, pp. 102–155, 180–228.
37. Crassidis, J.L.; Junkins, J.L. *Optimal Estimation of Dynamic Systems*; CRC Press: Boca Raton, FL, USA, 2011; Chapters 3–4.
38. Stevens, B.L.; Lewis, F.L.; Johnson, E.N. *Aircraft Control and Simulation: Dynamics, Controls Design, and Autonomous Systems*; John Wiley & Sons: Hoboken, NJ, USA, 2015; Chapters 1–2.
39. Markley, F.L. Attitude Error Representations for Kalman Filtering. *J. Guid. Control Dyn.* **2003**, *26*, 311–317. [[CrossRef](#)]
40. Sola, J. Quaternion Kinematics for the Error-State Kalman Filter. *arXiv* **2017**, arXiv:1711.02508.
41. Woodman, O.J. *An Introduction to Inertial Navigation*; Technical Report; University of Cambridge, Computer Laboratory: Cambridge, UK, 2007.
42. Titterton, D.; Weston, J.L.; Weston, J. *Strapdown Inertial Navigation Technology*; The Institution of Engineering and Technology (IET): London, UK, 2004; Volume 17, Chapters 4–7.
43. Forsyth, D.A.; Ponce, J. *Computer Vision: A Modern Approach*; Prentice Hall: Upper Saddle River, NJ, USA, 2002; Chapters 4–5, pp. 96–147.
44. Trajković, M.; Hedley, M. Fast Corner Detection. *Image Vis. Comput.* **1998**, *16*, 75–87. [[CrossRef](#)]
45. Rosten, E.; Drummond, T. Machine Learning for High-Speed Corner Detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*; Springer: Berlin/Heidelberg, Germany, 2006; pp. 430–443. [[CrossRef](#)]
46. Lucas, B.D.; Kanade, T. An Iterative Image Registration Technique with an Application to Stereo Vision. In Proceedings of the 7th International Joint Conference on Artificial Intelligence, Vancouver, BC, Canada, 24–28 August 1981; Morgan Kaufmann Publishers Inc.: Bullington, MA, USA, 1981; Volume 2, pp. 674–679.
47. Paul, M.K.; Wu, K.; Hesch, J.A.; Nerurkar, E.D.; Roumeliotis, S.I. A Comparative Analysis of Tightly-Coupled Monocular, Binocular, and Stereo Vins. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Singapore, 29 May–3 June 2017; pp. 165–172. [[CrossRef](#)]
48. Sun, K.; Mohta, K.; Pfommer, 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](#)]
49. Kitt, B.; Geiger, A.; Lategahn, H. Visual Odometry Based on Stereo Image Sequences with Ransac-Based Outlier Rejection Scheme. In Proceedings of the IEEE Intelligent Vehicles Symposium (IV), San Diego, CA, USA, 21–24 June 2010; pp. 486–492. [[CrossRef](#)]
50. Open CV. 2019. Available online: <https://opencv.org/> (accessed on 8 November 2019).
51. Devore, J.L. *Probability and Statistics for Engineering and the Sciences*; Cengage Learning: Boston, MA, USA, 2015; Chapter 7, pp. 267–299.
52. Bishop, C.M. *Pattern Recognition and Machine Learning*; Springer: Berlin/Heidelberg, Germany, 2006; Chapter 10, pp. 461–522.
53. 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](#)]
54. Sturm, J.; Engelhard, N.; Endres, F.; Burgard, W.; Cremers, D. A Benchmark for the Evaluation of RGB-D SLAM Systems. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vilamoura, Portugal, 7–12 October 2012; pp. 573–580. [[CrossRef](#)]

55. Delmerico, J.; Scaramuzza, D. A Benchmark Comparison of Monocular Visual-Inertial Odometry Algorithms for Flying Robots. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Brisbane, Australia, 21–25 May 2018; number CONF. [\[CrossRef\]](#)
56. Faessler, M.; Fontana, F.; Forster, C.; Mueggler, E.; Pizzoli, M.; Scaramuzza, D. Autonomous, Vision-Based Flight and Live Dense 3D Mapping with a Quadrotor Micro Aerial Vehicle. *J. Field Robot.* **2016**, *33*, 431–450. [\[CrossRef\]](#)
57. Forster, C.; Zhang, Z.; Gassner, M.; Werlberger, M.; Scaramuzza, D. SVO: Semidirect Visual Odometry for Monocular and Multicamera Systems. *IEEE Trans. Robot.* **2017**, *33*, 249–265. [\[CrossRef\]](#)
58. Mourikis, A.I.; Roumeliotis, S.I. A Multi-State Constraint Kalman Filter for Vision-Aided Inertial Navigation. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Roma, Italy, 10–14 April 2007; pp. 3565–3572. [\[CrossRef\]](#)
59. Xiong, S.; Zhou, Z.; Zhong, L.; Xu, C.; Zhang, W. Adaptive Filtering of Color Noise Using the Kalman Filter Algorithm. In Proceedings of the 17th IEEE Instrumentation and Measurement Technology Conference (IMTC), Baltimore, MD, USA, 1–4 May 2000; Volume 2, pp. 1009–1012. [\[CrossRef\]](#)
60. Kumar, A.; Crassidis, J.L. Colored-Noise Kalman Filter for Vibration Mitigation of Position/Attitude Estimation Systems. In Proceedings of the AIAA Guidance, Navigation and Control Conference and Exhibit, Hilton Head, SC, USA, 20–23 August 2007; p. 6516. [\[CrossRef\]](#)
61. Lee, K.; Johnson, E.N. State Estimation Using Gaussian Process Regression for Colored Noise Systems. In Proceedings of the IEEE Aerospace Conference, Big Sky, MT, USA, 4–11 March 2017; pp. 1–8. [\[CrossRef\]](#)
62. Lee, K.; Choi, Y.; Johnson, E.N. Kernel Embedding-Based State Estimation for Colored Noise Systems. In Proceedings of the IEEE/AIAA 36th Digital Avionics Systems Conference (DASC), St. Petersburg, FL, USA, 17–21 September 2017; pp. 1–8. [\[CrossRef\]](#)
63. Wu, A.D. Vision-Based Navigation and Mapping for Flight in GPS-Denied Environments. Ph.D. Thesis, Georgia Institute of Technology, Atlanta, GA, USA, 2010.
64. Bjorck, A. *Numerical Methods for Least Squares Problems*; Society for Industrial and Applied Mathematics (SIAM): Philadelphia, PA, USA, 1996; Volume 51, Chapters 2–9.
65. Montiel, J.M.; Civera, J.; Davison, A.J. Unified Inverse Depth Parametrization for Monocular SLAM. In Proceedings of the Robotics: Science and Systems, Philadelphia, PA, USA, 16–19 August 2006. [\[CrossRef\]](#)
66. Furgale, P.; Rehder, J.; Siegwart, R. Unified Temporal and Spatial Calibration for Multi-Sensor Systems. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Tokyo, Japan, 3–7 November 2013; pp. 1280–1286. [\[CrossRef\]](#)
67. Horn, B.K.; Hilden, H.M.; Negahdaripour, S. Closed-Form Solution of Absolute Orientation Using Orthonormal Matrices. *J. Opt. Soc. Am. A* **1988**, *5*, 1127–1135. [\[CrossRef\]](#)



© 2020 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 (<http://creativecommons.org/licenses/by/4.0/>).