sensors

MDPI

Article

Latency Compensated Visual-Inertial Odometry for Agile Autonomous Flight

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: 28 February 2020; Accepted: 12 April 2020; Published: 14 April 2020

Abstract: In visual-inertial odometry (VIO), inertial measurement unit (IMU) dead reckoning acts as the dynamic model for flight vehicles while camera vision extracts information about the surrounding environment and determines features or points of interest. With these sensors, the most widely used algorithm for estimating vehicle and feature states for VIO is an extended Kalman filter (EKF). The design of the standard EKF does not inherently allow for time offsets between the timestamps of the IMU and vision data. In fact, sensor-related delays that arise in various realistic conditions are at least partially unknown parameters. A lack of compensation for unknown parameters often leads to a serious impact on the accuracy of VIO systems and systems like them. To compensate for the uncertainties of the unknown time delays, this study incorporates parameter estimation into feature initialization and state estimation. Moreover, computing cross-covariance and estimating delays in online temporal calibration correct residual, Jacobian, and covariance. Results from flight dataset testing validate the improved accuracy of VIO employing latency compensated filtering frameworks. The insights and methods proposed here are ultimately useful in any estimation problem (e.g., multi-sensor fusion scenarios) where compensation for partially unknown time delays can enhance performance.

Keywords: VIO; UAV; EKF; IMU; camera vision; time delay; latency compensation; online temporal calibration; 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 (e.g., extended Kalman filter (EKF) [3,4] and unscented Kalman filter (UKF) [5]). The design of the standard Kalman filter does not inherently allow for significant sensor-related delays in computation. Figure 2 shows that the delay is the time difference between an instant when a measurement is taken by a sensor and another instant when the measurement is available in the filter. As an example of key delay sources, some complex sensors such as vision processors for navigation often require extensive computations to obtain higher-level information from raw sensor data. Furthermore, a closed-loop system including control logic may be an overall computational burden to a single processor. Delays resulting from heavy computation may distort the quality of state estimation since a current measurement is compared to past states of a system model. In other words, unless compensating delays in Kalman filtering, large estimation errors may accumulate over time, or even cause the filter to diverge.

The delay value is typically at least partially unknown and at least partially variable in many real applications. As an example of delay uncertainty contributors, even though a local clock is initially forced to synchronize with the centralized clock, deviations between clocks would occur because of clock drift, skew, or bias. In sensor fusion systems, when the timestamps of each sensor are typically recorded by triggered signals, non-deterministic, or non-quantized transmission delays lead to unknown time offsets on sensor streams. Moreover, if low-cost sensors such as rolling shutter cameras or software triggered devices are mounted on a vehicle, the variance of the uncertainty of timestamps might be larger. In particular, in visual-inertial odometry (VIO), we do not know the exact time instant when a camera opens and captures images for any particular pixel location. Often, exposure time depends on surrounding illumination conditions. The timestamp of the latest image by some cameras corresponds to some event such as when the shutter was triggered to start or when the entire image was available in memory. In practice, these uncertainties may be small compared to traditional sensors used for feedback in aerospace applications—but can be a major contributor to errors in emerging estimation problems such as VIO. Indeed, when estimating faster motions such as a highly agile unmanned aerial vehicle (UAV) or using progressive scan cameras, the unknown time delays may be a major driver of navigation quality and achievable controller bandwidth. We have experimentally observed the necessity of the time delay compensation to be more accurate than typically demanded, specifically for a UAV flying closed-loop on a vision-based navigation solution. With poor time delay compensation, we have observed oscillations and even divergence of estimates and closed-loop tracking error as expected, but even when we use fixed/known time delay compensation, we find time delay is still a limiting factor in accuracy and achievable control bandwidth. To illustrate, consider a UAV with a body-fixed camera that maneuvers with a more rapid rotation than reference design. Any time error produces a larger potential position estimation error with this faster rotation. Thus, even after the very best job possible has eliminated as much deterministic time delay error as is practical, we find that adapting to the non-deterministic error can enhance performance. In particular, we find it can be beneficial to deal with unknown time delays in VIO systems used in closed flight control and other systems like these.

1.1. Related Work

1.1.1. Visual-Inertial Odometry

In recent years, an increasing demand for the research of UAVs has prompted substantial interest in VIO systems [6–9]. Delmerico and Scaramuzza [10] provide a benchmark comparison of monocular VIO algorithms for flying robots. Similar to their comparison, Table 1 illustrates state-of-the-art VIO techniques even including stereo VIO. Let us explain some relevant terms for clarity. The tightly-coupled VIO jointly optimizes over all sensor measurements (i.e., visual and inertial cost terms in VIO) within a single process which results in higher accuracy. The opposite is referred to as the loosely-coupled. Indeed, the loosely-coupled VIO does not handle the correlation of visual and inertial motion constraints, resulting in the loss of information. Moreover, at the back-end of VIO, the optimization-based VIO solves a nonlinear least-squares problem (e.g., pose-graph optimization or bundle adjustment [11]) to update a window of states, which allows for reducing errors by re-linearization [12] but with a high computational cost and possibly stuck in the local minima. In contrast, the filtering-based VIO updates only the most recent state by the Kalman filter or EKF framework, resulting in computationally faster and more efficient, but one-time linearization possibly leads to linearization errors into the estimator. For more details of the terminology, see reference [13,14].

Table 1. State-of-the-art Visual-Inertial Odometry.

Name	**ROVIO** [15]	**VINS-MONO** [16]	**SVO +MSF** [17]	**Alternating Stereo VINS** [18]	**S-MSCKF** [19]	**OKVIS** [20]
Monocular	×	×	×			
Stereo				×	×	×
Indirect		×		×	×	×
Semi-direct			×			
Direct	×					
Loosely-Coupled			×			
Tightly-Coupled	×	×		×	×	×
Optimization-based		×				×
Filtering-based	×		×	×	×	
Open-source	×	×	×		×	×

VINS-MONO [16,21] is optimization-based visual simultaneous localization and mapping (SLAM) including loop closure. Some processes in this approach are not efficient due to the following reasons. VINS-MONO duplicates integration with the same IMU data at different timestamps for prediction and optimization purposes. That is, for publishing odometry at IMU rate, it integrates whenever IMU data arrives, whereas IMU data are also accumulated in a buffer for batch processing of integration at the time of image measurement update steps. Mourikis first introduced a multi-state constraint Kalman filter (MSCKF) [22,23], and Sun et al. [19] recently provided its stereo version. Although the real-time high-frequency VIO outputs might be crucial for UAV attitude control, MSCKF does not publish the odometry at the IMU rate but at the image rate. Furthermore, batch processing for IMU data integration in MSCKF may add redundant time delays to the filter when vision measurements are available. VINS-MONO and MSCKF are applicable to IMU and vision fusion. If we fuse other sensors such as the global positioning system (GPS) and altimeters in navigation systems, those approaches may not be operable since measurements from other sensors are available to update between images. Another limitation is that assumptions for IMU pre-integration between keyframes and backward propagation with loop closure in their approaches do not always hold. Hence, the EKF-based VIO frameworks cover a greater scope of sensor fusion problems.

Faessler et al. [17] combined semi-direct visual odometry (SVO) [24,25] with modular multi-sensor fusion (MSF) [26]. Even though this approach uses IMU data for fusing, since it is loosely-coupled, its results are sub-optimal. Paul et al. [18,27] recently proposed alternating stereo VINS that requires computation comparable to monocular VIO, yet provides scale information from the visual observations. However, this method may not be sufficient for tracking fast motion in low-latency demanding applications. Since the implementation is not open-source, this is not used for comparison in this paper. Leutenegger et al. [20] introduced a consistent keyframe-based stereo SLAM algorithm that performs nonlinear optimization over both visual and inertial cost terms. To maintain the sparsity of the system, their approach employs an approximation rendering it sub-optimal. Since it requires considerable computation resources or specific levels of sensors such as industrial grade IMUs, operating OKVIS in real-time is more challenging. Among the six algorithms in Table 1, only S-MSCKF and SVO+MSF handle an unknown time delay, so we will use their estimation results for comparison in this study.

1.1.2. State Estimation Using Time-Delayed Measurements

In a number of applications, a vital problem for combining data from various sensors is the fusion of delayed observations, and if the computational delay is crucial, fusing the data in a Kalman filter is challenging. During the last 20 years, the sensor time-delay problem has been addressed by a number of methods, most of which modify the Kalman filter so that it handles delay in the sensor update step. Alexander [28] derived a method of calculating a correction term and then added it to filter estimates when lagged measurements arrive. However, because the uncertainty of measurements is often an unknown quantity until the data are processed, applying the method in time-varying systems is not addressed. To overcome the shortcoming, Larsen et al. [29] extrapolated a measurement to a current time using the past and present estimates of the Kalman filter and calculated an optimal gain for this extrapolated measurement. However, Larsen's approach is exact for linear systems only, but if the system dynamics and measurement equations are significantly nonlinear, it can be highly inaccurate. For optimally fusing lagged sensor data in a general nonlinear system, Van Der Merwe et al. [30,31] introduced a new technique called "sample-state augmentation," based on the Schmidt–Kalman filter [32] or stochastic cloning [33]. Appendix C provides detailed background information about the new technique. Lastly, Gopalakrishnan et al. [34] provided a survey of all previously noted methods.

All of the above methods assume that the amount of time delay is known. As an illustration, those methods only work with synchronized sensors. However, the hardware synchronization of most low-cost or customized sensors is not always available. Moreover, situations in which a current, accurate time delay might not be known can arise in real applications. To deal with the unknown time delays, Julier and Uhlmann [35] introduced the covariance union algorithm, and Sinopoli et al. [36] modeled the arrival of intermittent observations as a random variable with a probability. In addition, Choi et al. [37] and Yoon et al. [38] augmented a state vector with as many past states as the maximum number of delayed steps. The size of this augmented state vector is extremely large, and calculations with the large-size vector might require additional extensive computational effort. Recently, for the uncertainty of time delays in state estimation, Lee and Johnson [39] also suggested an approach combined with multiple-model adaptive estimation. However because of imperfect information on a certain range of the delay value, this method might not be suitable if uncertainty of time delay is high.

Instead, we directly estimate the time delay as an additional state since augmentation is a straightforward means of handling the unknown delay. Nilsson et al. [40] investigated this idea using the Taylor series expansion for small delays. However, delay values are typically larger than a time step, and the linearization in their approach does not hold for large delays. Li and Mourikis [41] also examined the state augmentation for estimating an unknown time offset between the timestamps of two sensors. However, their approach is not optimal since it performs the measurement update of delayed sensor data without the covariance correction that uses the cross-covariance term computed during the delay period. Furthermore, in the recent optimization-based method proposed by Qin and Shen [42], if cameras move at non-constant speed during the short time period like progressive scan cameras, then their assumption does not hold. Despite the short time period, the camera coordinate frame is still changing and moving. Their assumption of a constant time offset is also not general since the unknown delay may be varying. To overcome all previously noted limitations, this paper proposes a novel approach, "latency compensated filtering" based on the combined parameter-state estimator [43,44].

1.2. Summary of Contributions

To fuse visual measurements with unknown time delays in VIO systems and systems like them (e.g., multi-sensor fusion), the approach in this paper incorporates three correction techniques into state estimation. First, we directly estimate the unknown part of actual delays in online fashion by augmenting vehicle-feature states. With the estimated unknown part and the approximately known part of the delays, we find the most precise measurement times based on the definition of total delays introduced in this paper. Next, at the calibrated measurement time, we evaluate the Jacobian and the residual for the EKF using interpolated states. At the measurement update of the EKF, the third correction is to formulate a modified Kalman gain by the cross-covariance term computed during the delay period. The testing results of this study on flight datasets show that the proposed latency compensated VIO is a more reliable and accurate navigation solution than the existing VIO systems.

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 unknown time delays and states of VIO, Sections 3 and 4 present theory and implementation for a novel combination of the parameter estimation technique with the modified EKF that compensates delayed measurements, respectively. Section 5 shows the testing results of this study on the benchmark flight dataset. The last section concludes and plans future work.

2. Preliminaries

2.1. Sequential Measurement Update

When multiple measurements are observed at one discrete-time, sequential Kalman filtering, shown in Figure 1, is useful [45]. In fact, we obtain N measurements, $y_1, y_2, \cdots, y_N$, at time k; that is, we first measure y_1, then y_2, $\cdots$, and finally y_N.

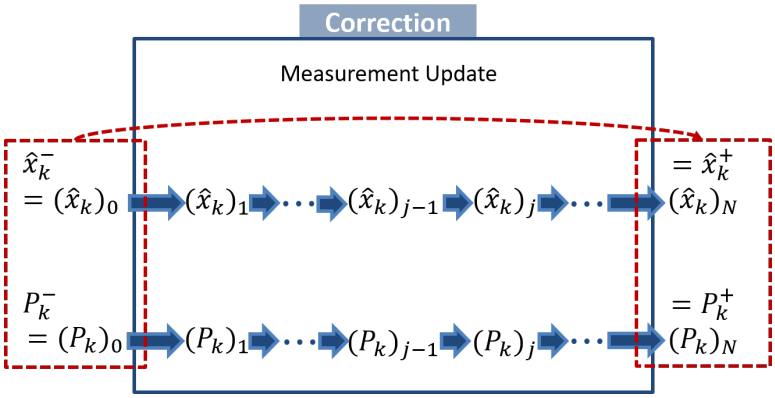


Figure 1. A schematic of the sequential measurement update.

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 $i = 1, \cdots, N$, 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 $P_k^+ \leftarrow (P_k)_N$. To clarify, hat "^" denotes an estimate, and superscript $-$ and $+$ a priori and a posteriori estimates, respectively. Based on Simon [45]'s proof that the sequential Kalman filtering is an equivalent formulation of the standard EKF, the order of updates does not affect the overall performance of estimation.

2.2. Vehicle Model

The nonlinear dynamics of a vehicle is driven by IMU sensor data including specific force and angular velocity. The estimated vehicle state is given by

$$\hat{x}_V = \begin{bmatrix} {}^i\hat{p}_{b/i}^{\mathrm{T}} & {}^i\hat{v}_{b/i}^{\mathrm{T}} & \delta\hat{\theta}^{\mathrm{T}} & \hat{b}_a^{\mathrm{T}} & \hat{b}_\omega^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}, \tag{1}$$

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 references [46–48]. b_a, 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 IMU sensor measurements a_{raw} and ω_{raw} are corrupted by noise and bias as follows:

$$a_{\mathrm{raw}} = a_{\mathrm{true}} - \mathcal{T}_{b/i}\,{}^i g + b_a + \eta_a, \qquad \dot{b}_a = \eta_{b_a} \tag{2}$$

$$\omega_{\mathrm{raw}} = \omega_{\mathrm{true}} + b_\omega + \eta_\omega, \qquad \dot{b}_\omega = \eta_{b_\omega}, \tag{3}$$

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

The vehicle dynamics is given by

$${}^i\dot{\hat{p}}_{b/i} = {}^i\hat{v}_{b/i} \tag{4}$$

$${}^i\dot{\hat{v}}_{b/i} = \hat{\mathcal{T}}_{i/b}\,(\,a_{\mathrm{raw}} - \hat{b}_a\,) + {}^i g \tag{5}$$

$$\dot{\hat{q}}_{i/b'} = \frac{1}{2}\,\mathcal{Q}(\,\omega_{\mathrm{raw}} - \hat{b}_\omega\,)\,\hat{q}_{i/b'} \tag{6}$$

$$\delta\dot{\hat{\theta}} = -\left\lceil\,(\,\omega_{\mathrm{raw}} - \hat{b}_\omega\,) \times \right\rfloor \delta\hat{\theta} \tag{7}$$

$$\dot{\hat{b}}_a = 0 \tag{8}$$

$$\dot{\hat{b}}_\omega = 0, \tag{9}$$

where $\lceil \alpha \times \rfloor$ 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 [44]. 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 $\delta q_{b/b'}$ to express the incremental difference between tracked reference body frame b' and actual body frame b for the vehicle.

$$q_{i/b} = \hat{q}_{i/b'} \otimes \delta q_{b'/b} \tag{10}$$

$$\delta q_{b'/b} = \hat{q}_{i/b'}^{-1} \otimes q_{i/b} \simeq \begin{bmatrix} 1 \\ \frac{1}{2}\delta\theta \end{bmatrix}, \tag{11}$$

where $\otimes$ is quaternion product defined in reference [47]. Resulting rotation matrices with error quaternion and with respect to the nominal reference body frame are

$$\mathcal{T}(q_{i/b}) = \hat{\mathcal{T}}_{b/i} = \hat{\mathcal{T}}_{b/b'}\,\hat{\mathcal{T}}_{b'/i} \tag{12}$$

$$\hat{\mathcal{T}}_{i/b'} = \hat{\mathcal{T}}_{b'/i}^{\mathrm{T}} = \mathcal{T}(\hat{q}_{i/b'})^{\mathrm{T}} \tag{13}$$

$$\hat{\mathcal{T}}_{b'/b} = \hat{\mathcal{T}}_{b/b'}^{\mathrm{T}} \simeq \left(\,I + [\delta\hat{\theta}\times]\,\right)^{\mathrm{T}}. \tag{14}$$

Jacobian matrix $A = \frac{\partial \dot{x}}{\partial x}|_{\hat{x}}$ and $B = \frac{\partial \dot{x}}{\partial \eta}$, where $\eta = [\,\eta_a^{\mathrm{T}}, \eta_\omega^{\mathrm{T}}, \eta_{b_a}^{\mathrm{T}}, \eta_{b_\omega}^{\mathrm{T}}\,]^{\mathrm{T}}$, are computed in Appendix A.

2.3. Camera Model

An intrinsically calibrated pinhole camera model [49,50] is given by

$$\begin{bmatrix} u_j \\ v_j \end{bmatrix} = y_j = h_j(x) + \zeta_j = \begin{bmatrix} f_u \frac{{}^cX_j}{{}^cZ_j} + \zeta_{u_j} \\ f_v \frac{{}^cY_j}{{}^cZ_j} + \zeta_{v_j} \end{bmatrix} \tag{15}$$

$$\begin{aligned} \begin{bmatrix} {}^cX_j, & {}^cY_j, & {}^cZ_j \end{bmatrix}^{\mathrm{T}} = {}^cp_{f_j/c} &= \mathcal{T}_{c/i}\left({}^ip_{f_j/i} - {}^ip_{c/i}\right) \\ &= \mathcal{T}_{c/b}\,\mathcal{T}(q_{i/b})\left({}^ip_{f_j/i} - {}^ip_{b/i}\right) - \mathcal{T}_{c/b}\,{}^bp_{c/b} \end{aligned} \tag{16}$$

where x is the state vector including the vehicle and feature state, and measurement y_j is the j-th feature $2D$ location on the image plane. f_u, f_v are the horizontal and vertical focal lengths, respectively, and ζ_u, 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 ${}^bp_{c/b}$ are known and constant, and rotation matrix $\hat{\mathcal{T}}_{c/i} = \mathcal{T}_{c/b}\,\hat{\mathcal{T}}_{b/b'}\,\hat{\mathcal{T}}_{b'/i}$. Jacobian matrix $C_j = \frac{\partial y_j}{\partial x}|_{\hat{x}}$ is computed in Appendix A.

2.4. Feature Initialization

From Equation (16), if j-th measurement y_j on an image is a new feature, then ${}^ip_{f_j/i}$ is unknown so it needs to be initialized. In the first step of the measurement update, we employ Gauss–Newton least-squares minimization [22,51] 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 [52] that is numerically more stable than the Cartesian parameterization. In other words, by the derivation explained in Appendix B, we obtain j-th feature $3D$ position ${}^{c_1}\hat{p}_{f_j/c_1}$ with respect to c_1 left camera frame of a stereo camera.

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

$$\begin{aligned} {}^i\hat{p}_{f_j/i} &= \hat{\mathcal{T}}_{i/c_1}\,{}^{c_1}\hat{p}_{f_j/c_1} + {}^i\hat{p}_{c_1/i} \\ &= \hat{\mathcal{T}}_{i/b}\,\mathcal{T}_{b/c_1}\,{}^{c_1}\hat{p}_{f_j/c_1} + \left({}^i\hat{p}_{b/i} + \hat{\mathcal{T}}_{i/b}\,{}^bp_{c_1/b}\right) \\ &= \hat{\mathcal{T}}_{i/b'}\hat{\mathcal{T}}_{b'/b}\left(\mathcal{T}_{b/c_1}\,{}^{c_1}\hat{p}_{f_j/c_1} + {}^bp_{c_1/b}\right) + {}^i\hat{p}_{b/i}. \end{aligned} \tag{17}$$

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

$$\hat{x} = \begin{bmatrix} \hat{x}_V^{\mathrm{T}} & {}^i\hat{p}_{f_j/i}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}, \tag{18}$$

where $\hat{x}_V$ is the estimated vehicle state vector defined in Equation (1). 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:

$$\begin{bmatrix} P & * \\ * & * \end{bmatrix} = \begin{bmatrix} I \\ J \end{bmatrix} P \begin{bmatrix} I & J^{\mathrm{T}} \end{bmatrix} = \begin{bmatrix} P & PJ^{\mathrm{T}} \\ JP & JPJ^{\mathrm{T}} + P_{f_{\mathrm{new}}} \end{bmatrix}, \tag{19}$$

where Jacobian $J = \frac{\partial p_{f/i}}{\partial x}\big|_{\hat{x}}$ is computed as follows:

$$J = \begin{bmatrix} I_{3\times3} & 0_{3\times3} & -\hat{\mathcal{T}}_{i/b'} \left[\left(\mathcal{T}_{b/c_1}{}^{c_1}\hat{p}_{f_j/c_1} + {}^{b}p_{c_1/b} \right) \times \right] & 0_{3\times6} & 0_{3\times3N_f} \end{bmatrix}. \tag{20}$$

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. In fact, since Montiel et al. [52] validate the initial uncertainty as a Gaussian distribution, the EKF including the feature initialization still holds optimality. Equations (18)–(20) arise based on "consider covariance analysis [53,54]".

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

$$\begin{bmatrix} \Phi & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} P_{VV} & P_{Vf} \\ P_{fV} & P_{ff} \end{bmatrix} \begin{bmatrix} \Phi^{\mathrm{T}} & 0 \\ 0 & I \end{bmatrix} + \begin{bmatrix} Q_V & 0 \\ 0 & Q_f \end{bmatrix} = \begin{bmatrix} \Phi\, P_{VV}\, \Phi^{\mathrm{T}} + Q_V & \Phi\, P_{Vf} \\ P_{fV}\, \Phi^{\mathrm{T}} & P_{ff} + Q_f \end{bmatrix}, \tag{21}$$

where state transition matrix $\Phi \approx I + A\,\Delta t$. $P_{VV} := \mathbb{E}[\,(x_V - \hat{x}_V)(x_V - \hat{x}_V)^{\mathrm{T}}\,]$, P_{ff} is the error covariance of all features, and $P_{Vf} = P_{fV}^{\mathrm{T}}$ represents vehicle–feature correlations. In addition, we assume that the surroundings are 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.

3. Theory

3.1. Definition of Time Delays

Based on dead reckoning, the EKF propagates state x and its error covariance P at time t when IMU sensor data a_{raw} and ω_{raw} are measured. Since an IMU is a discrete-time sensor, the time update of the EKF is processed in discrete time step $k = (\text{integer})\,(t\,/\,\Delta t_{\text{IMU}})$, where (integer) defines the conversion of all data types to integers, continuous time $t \in [0, t_{\text{final}}]$, and Δt_{IMU} is the sampling rate of the IMU. Δt_{IMU} is generally almost constant since a micro controller such as Arduino and Pixhawk calculates precise timestamps in milliseconds for each IMU measurement. Next, whenever new vision data from an image are arrived at the filter, the EKF performs the measurement update for correcting the state estimate and its error covariance. As introduced in Section 1, various reasons such as image processing produce time delays that the time stamps of vision data contain. For clarity, this section defines the time delay in detail.

Latency is the time difference between when an image was grabbed and when vision data from the image are updated in the filter, shown in Figure 2.

That is, true delays Δt_d are written as

$$t = t_{\text{img}} + \Delta t_d, \tag{22}$$

where t is current IMU time and t_{img} is the time when the current image was captured. In essence, we treat IMU time as our common time reference, and we do not necessarily know the exact time when images are grabbed. The timestamp of each image is encoded by indirect ways such as triggers. In other words, true image time t_{img} constitutes readable timestamps $t_{\text{img, raw}}$ and unknown δt_d. Let us define time differences $\Delta \bar{t}_d$ between the time readouts of sensors as follows:

$$t_{\text{img, raw}} = t_{\text{img}} + \delta t_d \tag{23}$$

$$\Delta \bar{t}_d := t - t_{\text{img, raw}} = \Delta t_d - \delta t_d \tag{24}$$

$$\Delta t_d = \Delta \bar{t}_d + \delta t_d, \tag{25}$$

where $\Delta \bar{t}_d$ and δt_d are the approximately known and the unknown parts of true delays t_d, respectively.

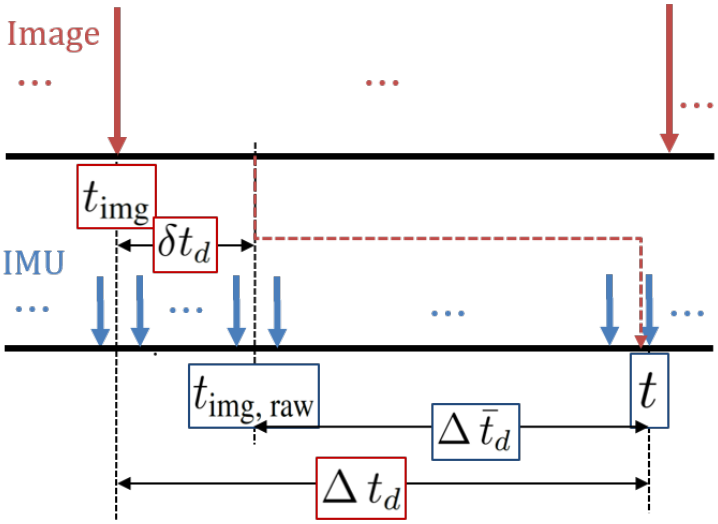


Figure 2. Data streams of the IMU and the delayed vision data.

3.2. *Approximately Known Part of Time Delays*

$\Delta\, \bar{t}_d$ is either a fixed value determined by offline beforehand tuning or readable differences between the time stamps of image and the time stamps of IMU data. Indeed, regardless of a constant value or readable varying delays, approximate delay $\Delta\, \bar{t}_d$ is a known value. Let the discrete steps of the approximately known part be $d = (\text{integer})\, (\Delta\, \bar{t}_d \,/\, \Delta\, t_{\text{IMU}})$, where (integer) means type conversion to integer from other types; that is, d is the quotient of division $\frac{\Delta\, \bar{t}_d}{\Delta\, t_{\text{IMU}}}$.

3.2.1. Jacobian and Residual—"Baseline Correction"

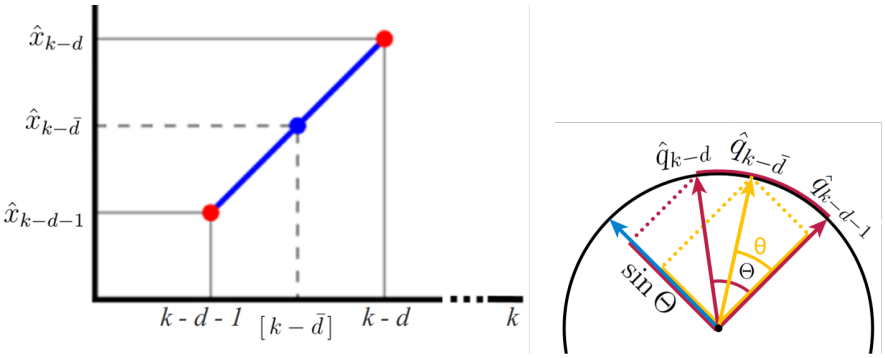
Since δt_d is unknown, we first consider only the $\Delta\, \bar{t}_d$ term as our delay of the system. From the system models given in Sections 2.2 and 2.3, only measurements from the camera model depend on the time delays. To correct the Jacobian and residual with approximately known delays, interpolation and quaternion slerp are required. Since $k - d \neq (\text{integer}) \left(\frac{t - \Delta\, \bar{t}_d}{\Delta t_{\text{IMU}}} \right)$, we define new time notation $[\, k - \bar{d}\,]$ as

$$[\, k - \bar{d}\,] := \frac{t_{\text{img, raw}}}{\Delta t_{\text{IMU}}} = \frac{t - \Delta\, \bar{t}_d}{\Delta t_{\text{IMU}}}.$$

When time $[\, k - \bar{d}\,]$ is expressed at subscript (e.g., $x_{[\, k - \bar{d}\,]}$, $P_{[\, k - \bar{d}\,]}$), we will use the shorthand notation without $[\ ]$ (e.g., $x_{k - \bar{d}}$, $P_{k - \bar{d}}$).

Although delay d in discrete-time systems is the number of delayed samples, time $[\, k - \bar{d}\,]$ is not required to be an integer by reading timestamps of each sensor. Since $[\, k - \bar{d}\,]$ is not an integer, we cannot directly access the values of either $\hat{x}_{k - \bar{d}}$ or its corresponding error covariance $P_{k - \bar{d}}$, so relevant interpolation is required instead.

Mathematically, linear interpolation constructs a new data point within the range of two known adjacent data points by the same slope of two lines [55]. Let us take the nearest integer time step $k - d$, which is greater than or equal to $[\, k - \bar{d}\,]$, shown in Figure 3a.



(**a**) (**b**)

Figure 3. Examples of interpolation and slerp. (**a**) Linear interpolation, (**b**) Quaternion slerp.

With two data points, either $(k-d-1,\, \hat{x}_{k-d-1})$ and $(k-d,\, \hat{x}_{k-d})$ or $(k-d-1,\, P_{k-d-1})$ and $(k-d,\, P_{k-d})$, the interpolants at time $[k-\bar{d}]$ are given by

$$\hat{x}_{k-d} - \hat{x}_{k-d-1} = \frac{\hat{x}_{k-d} - \hat{x}_{k-\bar{d}}}{k-d-[k-\bar{d}]}$$

$$\hat{x}_{k-d} - \hat{x}_{k-\bar{d}} \simeq \left(\frac{t}{\Delta t_{\mathrm{IMU}}} - d - \frac{t - \Delta \bar{t}_d}{\Delta t_{\mathrm{IMU}}} \right) (\hat{x}_{k-d} - \hat{x}_{k-d-1}) \tag{26}$$

$$\hat{x}_{k-\bar{d}} = \left(1 - \frac{\Delta \bar{t}_d}{\Delta t_{\mathrm{IMU}}} + d \right) \hat{x}_{k-d} + \left(\frac{\Delta \bar{t}_d}{\Delta t_{\mathrm{IMU}}} - d \right) \hat{x}_{k-d-1}, \tag{27}$$

where $k = (\text{integer}) \left(\frac{t}{\Delta t_{\mathrm{IMU}}} \right) \approx \frac{t}{\Delta t_{\mathrm{IMU}}}$ in Equation (26). Likewise,

$$P_{k-\bar{d}} = \left(1 - \frac{\Delta \bar{t}_d}{\Delta t_{\mathrm{IMU}}} + d \right) P_{k-d} + \left(\frac{\Delta \bar{t}_d}{\Delta t_{\mathrm{IMU}}} - d \right) P_{k-d-1}.$$

Although we compute the interpolants at time $[k-\bar{d}]$ using linear interpolation because of the constraint and specialty of quaternion, another interpolation is required. Slerp is shorthand for spherical linear interpolation, introduced by Ken Shoemake [56] in the context of quaternion interpolation for the purpose of animating $3D$ rotation. Interpolants refer to constant-speed motion along a unit-radius circle arc, shown in Figure 3b. Based on the fact that any point on the curve is linear combination of the given ends, the geometric formula [56,57] is

$$\Theta = \cos^{-1} \left(q_{k-d} \cdot q_{k-d+1} \right) \tag{28}$$

$$\hat{q}_{k-\bar{d}} = \frac{\sin \left[\left(1 - \frac{\Delta \bar{t}_d}{\Delta t_{\mathrm{IMU}}} + d \right) \Theta \right]}{\sin \Theta} \hat{q}_{k-d} + \frac{\sin \left[\left(\frac{\Delta \bar{t}_d}{\Delta t_{\mathrm{IMU}}} - d \right) \Theta \right]}{\sin \Theta} \hat{q}_{k-d-1}, \tag{29}$$

where since only unit quaternions are valid rotations, normalization of each quaternion before applying Slerp is a prerequisite.

Θ is a smaller angle between two end quaternions, so we ensure that $-90\,\mathrm{deg} \leq \Theta \leq 90\,\mathrm{deg}$. If the dot product in Equation (28) is negative, Slerp does not represent the shortest path. To prevent long paths, we negate one of end quaternions since q and $-q$ are equivalent when the negation is applied to all four components. If two quaternions input q_{k-d}, q_{k-d+1} are too close, then interpolants by linear interpolation are acceptable. Otherwise, $\cos^{-1}(\cdot)$ in Equation (28) is safe computation because the dot product is in the range of the threshold.

With suitable interpolants at time $[k-\bar{d}]$, a baseline approach modifies the feature initialization in Appendix B and the measurement update. At time k, the vision data of an image grabbed at time $(t-\Delta t_d)$ arrive at the filter for either the feature initializations or the sequential measurement updates. If j-th measurement y_j on the last image is a new feature, then, from Equations (18) and (19), state $\hat{x}$ and covariance P at current time k are augmented as follows:

$$\hat{x}_k \overset{\text{aug}}{\Longrightarrow} \begin{bmatrix} \hat{x}_k^{\mathrm{T}} & {}^{i}\hat{p}_{f_j/i}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}} \tag{30}$$

$$P_k \overset{\text{aug}}{\Longrightarrow} \begin{bmatrix} P_k & P_k\,(J_j)_{k-\bar{d}}^{\mathrm{T}} \\ (J_j)_{k-\bar{d}}\,P_k & (J_j)_{k-\bar{d}}\,P_k\,(J_j)_{k-\bar{d}}^{\mathrm{T}} + P_{f_{j\text{new}}} \end{bmatrix} \tag{31}$$

where ${}^{i}\hat{p}_{f_j/i} = \mathcal{T}_{i/b}|_{k-\bar{d}}\,{}^{b}\hat{p}_{f_j/b} + ({}^{i}p_{b/i})_{k-\bar{d}}$ and $(J_j)_{k-\bar{d}} = \frac{\partial p_{f_j/i}}{\partial x}\big|_{\hat{x}_{k-\bar{d}}}$. ${}^{b}\hat{p}_{f_j/b}$ is initialized by Gaussian-Newton least-squares minimization derived in Appendix B. Although we assume static features, since the feature initialization is related to estimated camera pose at the time when the delays begin, corrected Jacobian J_j is required in the initialization steps.

If j-th measurement y_j on the image is a tracked feature, then we correct only residual r and Jacobian C in the following measurement update:

$$K_j = (P_k)_{j-1}\,(C_j)_{k-\bar{d}}^{\mathrm{T}}\,\Big(\,(C_j)_{k-\bar{d}}\,(P_k)_{j-1}\,(C_j)_{k-\bar{d}}^{\mathrm{T}} + R\,\Big)^{-1} \tag{32}$$

$$(\hat{x}_k)_j = (\hat{x}_k)_{j-1} + K_j\,\Big(\,y_j|_{t-\Delta t_d} - h_j(\hat{x}_{k-\bar{d}})\,\Big) \tag{33}$$

$$(P_k)_j = (P_k)_{j-1} - K_j\,(C_j)_{k-\bar{d}}\,(P_k)_{j-1} \tag{34}$$

where corrected residual $(r_j)_{k-\bar{d}} = y_j|_{t-\Delta t_d} - h_j(\hat{x}_{k-\bar{d}})$ and Jacobian $(C_j)_{k-\bar{d}} = \frac{\partial\, h_j(x)}{\partial\, x}\Big|_{\hat{x}_{k-\bar{d}}}$. R is the measurement noise covariance of $y_j|_{t-\Delta t_d}$, and K_j is sub-optimal Kalman gain computed by current covariance. As sequential Kalman Filtering introduced in Section 2.1, if j is the first feature on the current image (i.e., $j=0$), then assign $(\hat{x}_k)_0 \leftarrow \hat{x}_k^-$, $(P_k)_0 \leftarrow P_k^-$, and if j is the last feature on the current image (i.e., $j=N_k$), then assign $\hat{x}_k^+ \leftarrow (\hat{x}_k)_N$, $P_k^+ \leftarrow (P_k)_N$. Before measurement updates (32)–(34), a chi-squared gating test rejects outliers of each measurement. For only this test purpose in the case of baseline correction, we add uncertainty due to time delay. Procedures in Equations (30)–(34) are referred to as the "baseline correction."

3.2.2. Cross-Covariance—"Covariance Correction"

During the delay period, even though an image was already captured in the past, since vision data from the image have not yet arrived at the filter, the EKF is not able to perform the measurement update. Indeed, the filter processes only time update. When a vision data packet from the image finally arrives and is ready to update in the filter, we simply execute the Jacobian and residual correction in Equations (32)–(34) using the delayed measurements. However, unlike the baseline correction, if the filter updates as if the measurements arrive immediately without delays (like red lines in Figure 4), then filter can achieve a more accurate estimate. In fact, covariance correction presented in this section (like blue lines in Figure 4) is as if the filter accomplished the general measurement update at the time instant when the image was captured. In other words, red lines in Figure 4 are ideal but unrealistic, and blue lines in the figure are practical. The red lines process the measurement update first and then time update; however, the order of the processes of the blue lines is the opposite. Only the order of the processes has changed.

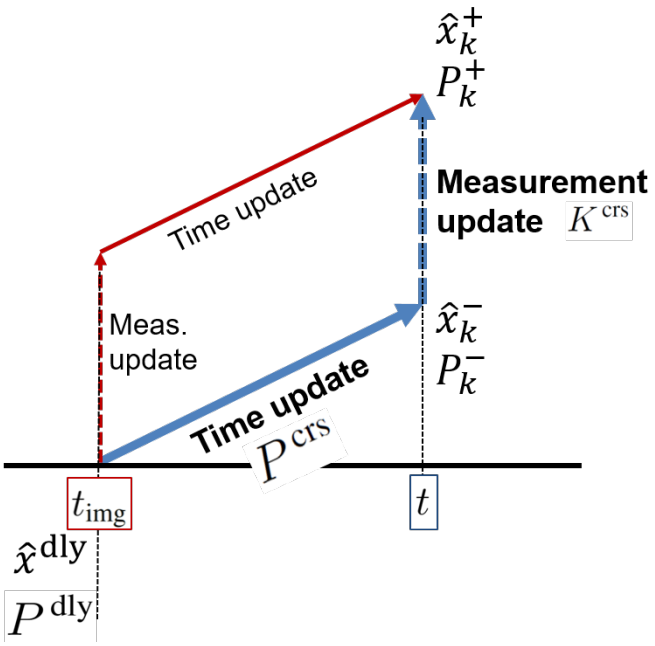


Figure 4. A schematic of a modified measurement update using covariance correction.

Among a variety of fusing techniques for time-delayed observations discussed in Section 1.1.2, the stochastic cloning [33]-based method (i.e., the Schmidt EKF [30,31]) is applicable to varying delays and nonlinear functions such as the vehicle and camera models described in Sections 2.2 and 2.3, respectively. Thus, this study modifies the method for finding the optimal navigation solution of vision-aided inertial navigation systems.

Let us introduce new notation P^{dly}. P^{dly} is P covariance matrix at the time when the true delays begin. In the scope of this section, $P^{\text{dly}} \simeq P_{k-\bar{d}}$. In addition, when this section uses corrected residual $(r_j)_{k-\bar{d}}$ and Jacobians $(J_j)_{k-\bar{d}}$, $(C_j)_{k-\bar{d}}$, we will use their shorthand notations as r_j and J_j, C_j, respectively. That is, each residual and Jacobian is corrected based on Section 3.2.1. In addition to the baseline correction, we correct error covariance in both the feature initialization and the measurement update when delayed vision data are available in the filter.

If j-th feature measurement y_j on the recent image is a new feature, the augmentation of P^{dly} in the feature initialization is similar to Equation (31). On the other hand, since Jacobian J_j is computed at the time when the delays begin, the augmentation of covariance matrix P_k at the current time is as follows in a different way:

$$P_k \overset{\text{aug}}{\Longrightarrow} \begin{bmatrix} P_k & 0 \\ 0 & d\,Q_f \end{bmatrix} \tag{35}$$

$$P^{\text{dly}} \overset{\text{aug}}{\Longrightarrow} \begin{bmatrix} P^{\text{dly}} & P^{\text{dly}} J_j^{\text{T}} \\ J_j P^{\text{dly}} & J_j P^{\text{dly}} J_j^{\text{T}} + P_{f_{j_{\text{new}}}} \end{bmatrix}, \tag{36}$$

where $d = (\text{integer}) \frac{\Delta \bar{t}_d}{\Delta t_{\text{IMU}}}$ and $J_j = \frac{\partial p_{f_j/i}}{\partial x}\big|_{\hat{x}_{k-\bar{d}}}$. State estimate $\hat{x}_k$ is augmented by Equation (30).

When j-th delayed vision data y_j is ready to update at time k, we modify the measurement update steps of the sequential Kalman filtering as follows:

$$S_j = C_j \left(P^{\text{dly}}\right)_{j-1} C_j^{\text{T}} + R$$

$$K_j^{\text{crs}} = \left(P^{\text{crs}}\right)_{j-1} C_j^{\text{T}} S_j^{-1} \tag{37}$$

$$\left(\hat{x}_k\right)_j = \left(\hat{x}_k\right)_{j-1} + K_j^{\text{crs}} r_j \tag{38}$$

$$\left(P_k\right)_j = \left(P_k\right)_{j-1} - K_j^{\text{crs}} C_j \left(P^{\text{crs}}\right)_{j-1}^{\text{T}}, \tag{39}$$

where $r_j = y_j\big|_{t-\Delta t_d} - h_j(\hat{x}_{k-\bar{d}})$ and $C_j = \left.\frac{\partial h_j(x)}{\partial x}\right|_{\hat{x}_{k-\bar{d}}}$. P^{crs} is the relevant cross-covariance term during the delay period. This term, which fuses a current prediction of the state with an observation related to the lagged state of the system, is used for formulating modified Kalman gain matrix K^{crs}. Equation (39) still holds Joseph's form [58] that preserves the symmetry of the updated covariance and ensures its the positive definiteness. By sequential update provisions, the state estimate and covariance at time $[k-\bar{d}]$ are also updated as follows:

$$K_j^{\text{dly}} = \left(P^{\text{dly}}\right)_{j-1} C_j^{\text{T}} S_j^{-1} \tag{40}$$

$$\left(\hat{x}_{k-\bar{d}}\right)_j = \left(\hat{x}_{k-\bar{d}}\right)_{j-1} + K_j^{\text{dly}} r_j \tag{41}$$

$$\left(P^{\text{dly}}\right)_j = \left(P^{\text{dly}}\right)_{j-1} - K_j^{\text{dly}} C_j \left(P^{\text{dly}}\right)_{j-1}. \tag{42}$$

At time t_{img}, when cameras open for capturing the image, the cross-covariance matrix is initialized with covariance at that time; that is, $P^{\text{crs}} \leftarrow P^{\text{dly}} \approx P_{k-\bar{d}}$. During the delay period, from time $[k-\bar{d}]$ to current time k, if no other measurements are fused into the filter, the cross-covariance is only propagated by the following computation based on the Schmidt–Kalman filter [30,32,33]:

$$\Phi^{\text{crs}} = \prod_{i=k-1}^{k-d} \begin{bmatrix} \Phi_i & 0 \\ 0 & I \end{bmatrix} = \begin{bmatrix} \left(\prod_{i=k-1}^{k-d} \Phi_i\right) & 0 \\ 0 & I \end{bmatrix} \tag{43}$$

$$P^{\text{crs}} = \Phi^{\text{crs}} P^{\text{dly}} \tag{44}$$

$$= \begin{bmatrix} \left(\prod_{i=k-1}^{k-d} \Phi_i\right) P_{vv}^{\text{dly}} & \left(\prod_{i=k-1}^{k-d} \Phi_i\right) P_{vf}^{\text{dly}} \\ P_{fv}^{\text{dly}} & P_{ff}^{\text{dly}} \end{bmatrix} \tag{45}$$

where Φ is the state transition matrix. In the sequential measurement update, based on updated $\left(P^{\text{dly}}\right)_j$ in Equation (42), updating $\left(P^{\text{crs}}\right)_{j-1}$ is straightforward as follows:

$$\left(P^{\text{crs}}\right)_j = \Phi^{\text{crs}} \left(P^{\text{dly}}\right)_j \tag{46}$$

If other measurements from other sensors such as an altimeter and GPS are fused during the delay period, then P^{dly} and cross-covariance P^{crs} are also recursively updated using the Kalman gain of the other measurements. For this case, Equations (43)–(46) do not hold any longer. For more details, see Appendix C. All modification in this section is referred to as "covariance correction." Furthermore, the optimality of this covariance correction is guaranteed based on the fact that the standard Kalman filter is an optimal filter since Appendix C proves that the covariance correction is identical to the standard EKF. Hence, the proposed correction still holds its optimality. Section 4 will describe its practical implementation.

3.3. Unknown Part of Time Delays—"Online Calibration"

Although residual, Jacobians, and covariance are corrected for measurements with time delays, if $\Delta\,\bar{t}_d$ is uncertain readouts or δt_d is the larger portion of true delays, we cannot guarantee the reliability of the correction algorithm (Figure 5). For the robustness of vision-aided navigation systems, we need to additionally investigate the unknown part of true delays.

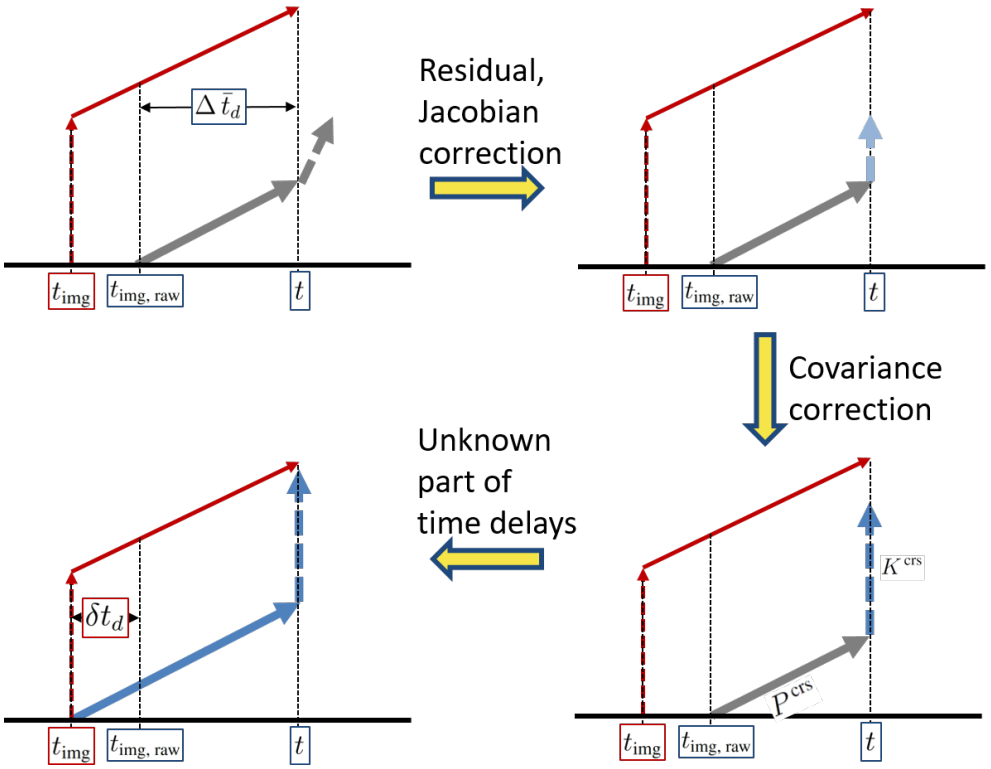


Figure 5. Three corrections in the latency compensated VIO.

Figure 5 shows three corrections in the latency compensated VIO presented in this study. From the standard Kalman filter, if one does not account for time delay, propagation, and measurement update look like grey lines in Figure 5. For the last correction, we estimate the unknown part of time delays to obtain more precise time instant when the delays begin. As discussed in Section 1, unknown phenomena such as clock bias, drift, skews, and asynchronization cause δt_d, so δt_d may be a positive or negative value.

State estimation theory can be used to estimate not only the states but also the unknown parameters of the system [59]. Numerous researchers [60–62] have proved that state augmentation functions are easy to use with state observers, so we enable design a state observer by state augmentation to estimate the unknown part of the time delays. To estimate unknown delay value δt_d, we first augment state estimates $\hat{x}_V$ and covariance P_{vv} of the vehicle as follows:

$$x_V \overset{\text{aug}}{\Longrightarrow} \begin{bmatrix} x_V^{\mathrm{T}}, & \delta t_d \end{bmatrix}^{\mathrm{T}}, \quad P_{vv} \overset{\text{aug}}{\Longrightarrow} \begin{bmatrix} P_{vv} & P_{v\,\delta t_d} \\ P_{\delta t_d\, v} & P_{\delta t_d} \end{bmatrix}. \tag{47}$$

Like the modeling of the IMU biases in Equations (2) and (3), we model the dynamics of δt_d using a small artificial noise term

$$\dot{\delta t}_d = \eta_d, \quad \dot{\hat{\delta}}t_d = 0, \tag{48}$$

where η_p is a random walk rate that allows the EKF to change its estimate of δt_d; that is, the power spectral density of η_p represents the variability of δt_d. In fact, this is a conventional random walk

model for an unknown parameter that may be varying—commonly seen for things like gyro bias, as done here. If additional modeling information about the way time delays is expected to vary is known, then it could be captured here with a more complex model.

Let us rewrite the definition of time delays:

$$\begin{aligned} t - t_{\text{img}} &= \Delta t_d \\ \Delta t_d &= \Delta \bar{t}_d + \delta t_d = \left(t - t_{\text{img, raw}}\right) + \delta t_d. \end{aligned}$$

For clarity, we define new time notation $[k - \hat{d}]$ as

$$\begin{aligned} [k - \hat{d}] &:= \frac{t_{\text{img, raw}} - \hat{\delta} t_d}{\Delta t_{\text{IMU}}} \approx \frac{t_{\text{img}}}{\Delta t_{\text{IMU}}} \\ &= \frac{t - (\Delta \bar{t}_d + \hat{\delta} t_d)}{\Delta t_{\text{IMU}}}, \end{aligned}$$

where now time $[k - \hat{d}]$ is the most precise time instant when the image was captured. To apply the relevant interpolation techniques to the state estimates and covariance at time $[k - \hat{d}]$, we access their values at the nearest integer time step $k - s$, where $s = (\text{integer}) \left(\Delta \bar{t}_d + \hat{\delta} t_d\right) / \Delta t_{\text{IMU}}$. In other words, s, discrete delayed samples including estimated latency, is greater than or equal to $[k - \hat{d}]$, as shown in Figure 2.

To operate the augmented system, we match its dimension by augmenting other matrices. In the time update, since $\dot{\hat{\delta}} t_d = 0$, the state transition matrix and the process noise covariance matrix are augmented

$$\Phi \overset{\text{aug}}{\Longrightarrow} \begin{bmatrix} \Phi & 0 \\ 0 & I \end{bmatrix}, \quad Q_v \overset{\text{aug}}{\Longrightarrow} \begin{bmatrix} Q_v & 0 \\ 0 & Q_d \end{bmatrix} \tag{49}$$

where I is due to $\dot{\hat{\delta}} t_d = 0$ and the Gaussian white noise $\eta_d \sim \mathcal{N}(0, Q_d)$. Under the assumption of static features, since estimated latency δt_d pertains to only vision measurements, we compute augmented elements of Jacobian matrices J and C [41,43]. In fact, from Equation (20), Jacobian J_j in the feature initialization is augmented as follows:

$$(J_j)_{k-\hat{d}} \overset{\text{aug}}{\Longrightarrow} \begin{bmatrix} I_{3\times 3} & 0_{3\times 3} & \left.\frac{\partial \hat{p}_{f_j/i}}{\partial \delta \hat{\theta}}\right|_{[k-\hat{d}]} & 0_{3\times 6} & J_{j_{\delta t_d}} & |0 \cdots \end{bmatrix},$$

where

$$\begin{aligned} J_{j_{\delta t_d}} &= \left.\frac{\partial \hat{p}_{f_j/i}}{\partial \hat{\delta} t_d}\right|_{[k-\hat{d}]} \simeq \left.\frac{\partial \hat{p}_{f_j/i}}{\partial x}\right|_{\hat{x}_{k-\hat{d}}} \cdot \left.\frac{\partial x}{\partial t}\right|_{[k-\hat{d}]} \cdot \left.\frac{\partial t}{\partial \hat{\delta} t_d}\right|_{[k-\hat{d}]} \\ &= (J_j)_{k-\hat{d}}\, \dot{\hat{x}}_{k-\hat{d}} \\ &= \begin{bmatrix} I_{3\times 3} & 0_{3\times 3} & \left.\frac{\partial \hat{p}_{f_j/i}}{\partial \delta \hat{\theta}}\right|_{[k-\hat{d}]} & 0_{3\times 3} & 0_{3\times 3} & |0 \cdots \end{bmatrix} \begin{bmatrix} {}^i\dot{\hat{p}}_{b/i} \\ {}^i\dot{\hat{v}}_{b/i} \\ \delta\dot{\hat{\theta}} \\ 0 \\ 0 \\ \hline (0 \cdots 0)^{\mathrm{T}} \end{bmatrix}_{[k-\hat{d}]} \\ &= \left({}^i\hat{v}_{b/i}\right)_{k-\hat{d}} + \left(\left.\frac{\partial \hat{p}_{f_j/i}}{\partial \delta \hat{\theta}}\right|_{[k-\hat{d}]}\right) \delta\dot{\hat{\theta}}_{k-\hat{d}} \end{aligned} \tag{50}$$

Furthermore, from Equation (A2), augmented Jacobian C_j in the measurement update is

$$(C_j)_{k-\hat{d}} \overset{\text{aug}}{\Longrightarrow} \left[\frac{\partial y_j}{\partial {}^i\hat{p}_{b/i}} \quad 0 \quad \frac{\partial y_j}{\partial \delta\hat{\theta}} \quad 0 \quad 0 \quad \frac{\partial y_j}{\partial \delta\hat{t}_d} \quad \Big| 0 \cdots \quad \frac{\partial y_j}{\partial {}^i\hat{p}_{f_j/i}} \quad \cdots 0 \right]_{[k-\hat{d}]},$$

where

$$\begin{aligned} \left.\frac{\partial y_j}{\partial \delta\hat{t}_d}\right|_{[k-\hat{d}]} &\simeq \left.\frac{\partial y_j}{\partial x}\right|_{\hat{x}_{k-\hat{d}}} \cdot \left.\frac{\partial x}{\partial t}\right|_{[k-\hat{d}]} \cdot \left.\frac{\partial t}{\partial \delta\hat{t}_d}\right|_{[k-\hat{d}]} = (C_j)_{k-\hat{d}}\, \dot{\hat{x}}_{k-\hat{d}} \\ &= \left(\left.\frac{\partial y_j}{\partial {}^i\hat{p}_{b/i}}\right|_{[k-\hat{d}]} \right) ({}^i\hat{v}_{b/i})_{k-\hat{d}} + \left(\left.\frac{\partial y_j}{\partial \delta\hat{\theta}}\right|_{[k-\hat{d}]} \right) \delta\dot{\hat{\theta}}_{k-\hat{d}}. \end{aligned} \tag{51}$$

Here, let us call the combination of the estimation of the unknown latency in this section with the baseline correction "online calibration." Therefore, to reliably estimate the state variable and effectively compensate the total delays, we incorporate all three corrections, called "latency-adaptive filtering."

4. Implementation

This section summarizes and describes an implementation of the proposed method. Figure 6 illustrates a flow chart of the overall process.

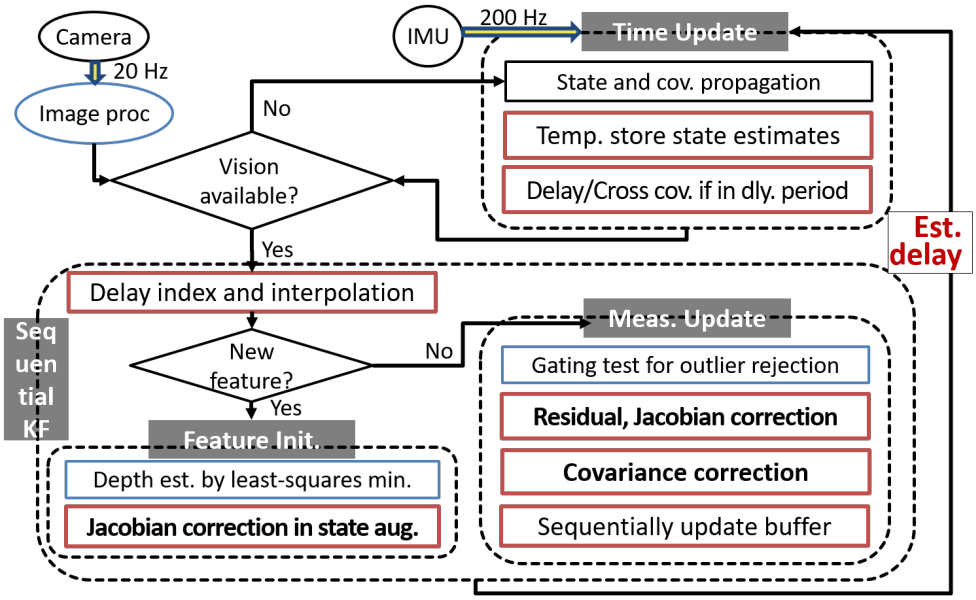


Figure 6. A flow chart of the overall process of the latency compensated VIO.

4.1. Forward Computation of Cross-Covariance

Even though delays begins Δt_d time prior, estimated delay value $\hat{t}_d$ is only accessible when delay finished. That is, during the delay period from t_{img} to t, $\Delta\hat{t}_d$ is unknown yet. $\Delta\hat{t}_d$ is estimated at current time k. Since estimated delay value $\Delta\hat{t}_d$ is unknown up to time k, we are not sure when the covariance correction begins computing cross-covariance P^{crs}. Theoretically, when $\Delta\hat{t}_d$ is estimated at time k, we compute P^{dly} and P^{crs} by backward from time k to time $[k-\hat{d}]$ with saved Jacobians and covariance during the delay period. This is an ideal computation, but not realistic. Backward computation that is used in [43] is impossible for real-time operations since storing large matrices such as sequences of Jacobian and covariance matrices allocates huge memory uses. Furthermore, backward computing is not efficient because it iterates backward at time k like batch processing.

Instead, for a real-time framework, an approximated way of forward computation of cross-covariance is introduced. Since $\delta\hat{t}_d = 0$, we assume that the time delay does not change

in state propagation during the delay period, so a posteriori estimate of time delay when the last measurement update is assumed to be a priori estimate of the delay at the current time. Next, under this assumption, we predict when the time delay of the next image begins. At the predicted time instant, we store the covariance matrix once for P^{dly} and recursively calculate Φ^{crs} for P^{crs}.

4.2. Summarized Algorithm

When the size of the state after augmentation in the feature initialization steps exceeds a maximum threshold, we prune the number of features in the database. The system in this study finds an index for the best place to insert a new point in the database. The one with the least number of observations or frequent outliers is marginalized. Unlike Lee at al. [43], this study does not estimate the total parts of time delays, so the latency compensated filter does not entail specific constraints. That is, this study estimates only unknown part δt_d that is a possible positive or negative value. To save computation, constrained Kalman filtering is not necessary. Instead, interpolation and quaternion Slerp explained in Section 3.2.1 are tractable.

From the definition of time delays presented in Section 3.1, the total time delay is not estimated as negative. For example, if the estimated delay is negative (i.e., an exceeded index), estimation is impossible since this case is forecasting states or obtaining measurements from the future, so the total delay has to be bounded by zero. Moreover, in the sequential measurement update, if estimated time delay δt_d is larger than the sampling time of the IMU, Δt_{IMU}, then we indicate another slot in the delay buffer. Algorithm 1 is a summarized algorithm of the overall processes of the latency compensated VIO.

Algorithm 1 The Latency Compensated VIO

Require: $\hat{x}_0^+ (= \hat{x}_{V_0}^+), P_0^+, Q, R,\ P^{\text{dly}} (= P_0^+), \Phi^{\text{crs}} (= I),\ \chi^2$

for $k = 1 : T$ **do**
  if new IMU packet arrival **then**
Time Update:
    $\dot{\hat{x}}_V = f\left(\hat{x}_{V_{k-1}}^+,\ a_{\text{raw}}, \omega_{\text{raw}}\right)$ ▷ static features
    Numerically integrate with $\Delta t_{\text{IMU}} (= t_k - t_{k-1})$
    $\hat{x}_k^- = \hat{x}_{k-1}^+ + \int_{t_{k-1}}^{t_k} \dot{\hat{x}}(\tau)\, d\tau$
    $\Phi_{k-1} = \exp\left(\int_{t_{k-1}}^{t_k} A(\tau)\, d\tau\right)$ ▷ $A = \frac{\partial f}{\partial x}\big|_{\hat{x}_V}$
    $(P_{vv})_k^- = \Phi_{k-1} (P_{vv})_{k-1}^+ \Phi_{k-1}^{\text{T}} + B_{k-1} Q_v B_{k-1}^{\text{T}}$ ▷ $B = \frac{\partial f}{\partial \eta}\big|_{\hat{x}_V}$
    $P_k^- = \begin{bmatrix} (P_{vv})_k^- & \Phi_{k-1} (P_{vf})_{k-1}^+ \\ (P_{fv})_{k-1}^+ \Phi_{k-1}^{\text{T}} & (P_{ff})_{k-1}^+ + Q_f \end{bmatrix}$
    Store the state estimates into the delay buffer
    if during delay period **then**
      $\Phi^{\text{crs}} \leftarrow \Phi_{k-1} \Phi^{\text{crs}}$ ▷ recursive
    else
      $P^{\text{dly}} \leftarrow P_k^-$
      $\Phi^{\text{crs}} \leftarrow I$
    end if
  end if

if new vision data packet arrival **then**
  Compute index $\hat{d}^{-}$ of delay
  Interpolate using the state estimates from the buffer
  for $j = 1 : \sharp$ of observed features **do**
    if new feature **then**
Feature Initialization:
      $\hat{p}_{f_j/i} = g_j\left(\hat{x}_{k-\hat{d}},\, y_j\right)$ ▷ least-squares minimization
      Augment state if feature is valid ▷ if positive depth
      $\hat{x}_k \overset{\text{aug}}{\Longrightarrow} \begin{bmatrix} \hat{x}_k^{\mathrm{T}} & \hat{p}_{f_j/i}^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}$
      $P^{\text{dly}} \overset{\text{aug}}{\Longrightarrow} \begin{bmatrix} P^{\text{dly}} & P^{\text{dly}} J_j^{\mathrm{T}} \\ J_j P^{\text{dly}} & J_j P^{\text{dly}} J_j^{\mathrm{T}} + P_{f_{j\text{new}}} \end{bmatrix}$ ▷ $J_j = \frac{\partial p_{f_j/i}}{\partial x}\big|_{\hat{x}_{k-\hat{d}}}$
      $P_k \overset{\text{aug}}{\Longrightarrow} \begin{bmatrix} P_k & 0 \\ 0 & \hat{d}\, Q_f \end{bmatrix}$
      Prune state vector if exceed maximum
    else ▷ tracked feature
Measurement Update:
      Update if gating test is passed ▷ $r_j^{\mathrm{T}} S_j^{-1} r_j \overset{?}{<} \chi_j^2$
      $r_j = y_j - h_j(\hat{x}_{k-\hat{d}})$
      $S_j = (C_j)_{k-\hat{d}}\, P^{\text{dly}}\, (C_j)_{k-\hat{d}}^{\mathrm{T}} + R$ ▷ $(C_j)_{k-\hat{d}} = \frac{\partial h_j}{\partial x}\big|_{\hat{x}_{k-\hat{d}}}$
      $P^{\text{crs}} \leftarrow \begin{bmatrix} \Phi^{\text{crs}} P_{vv}^{\text{dly}} & \Phi^{\text{crs}} P_{vf}^{\text{dly}} \\ P_{fv}^{\text{dly}} & P_{ff}^{\text{dly}} \end{bmatrix}$
      $K_j^{\text{crs}} = P^{\text{crs}}\, (C_j)_{k-\hat{d}}^{\mathrm{T}}\, S_j^{-1}$
      $\Delta\, \hat{x}_k = +K_j^{\text{crs}}\, r_j$ ▷ $\Delta \hat{t}_d \overset{?}{<} \Delta t_{\text{IMU}}$
      $\Delta\, P_k = -K_j^{\text{crs}}\, (C_j)_{k-\hat{d}}\, (P^{\text{crs}})^{\mathrm{T}}$
      Sequentially update the buffer
      $K_j^{\text{dly}} = P^{\text{dly}}\, (C_j)_{k-\hat{d}}^{\mathrm{T}}\, S_j^{-1}$
      $\Delta\, \hat{x}_{k-\hat{d}} = +K_j^{\text{dly}}\, r_j$
      $\Delta\, P^{\text{dly}} = -K^{\text{dly}}\, (C_j)_{k-\hat{d}}\, P^{\text{dly}}$
    end if
  end for
  Store index $\hat{d}^{+}$ of the posterior estimated delay
  $P^{\text{dly}} \leftarrow P_k^{+}$, $\Phi^{crs} \leftarrow I$
  Erase used slots in the delay buffer
end if
end for

5. Results

This section provides results from Monte Carlo trials and flight datasets testing. First, in a simulation environment, since we can set true time-delay values at measurements, we test if the proposed framework estimates the actual time delays accurately. The next subsection presents the performance of the proposed approach by testing with benchmark flight datasets for validating whether it solves real-world problems.

5.1. Monte Carlo Simulations of a Simple Example

To show actual-time delays being estimated accurately, this section simulates a simple example problem by 100 Monte Carlo trials. The vehicle and measurement models of this simulation are direct from Lee and Johnson's previous work [43]. The models are a second-order dynamic system with a non-delayed speed measurement and two delayed bearing angles measured from each location of two stations. From Equation (49), variance Q_d value of this simulation is $0.25\,s^2$.The actual time delay of the delayed measurements in this simulation is $0.9\,s$, and this value is identical to 18 delayed samples since the propagation rate of the simulation is $0.05\,s$. Monte Carlo simulations estimate the values of time delays, shown in Figure 7.

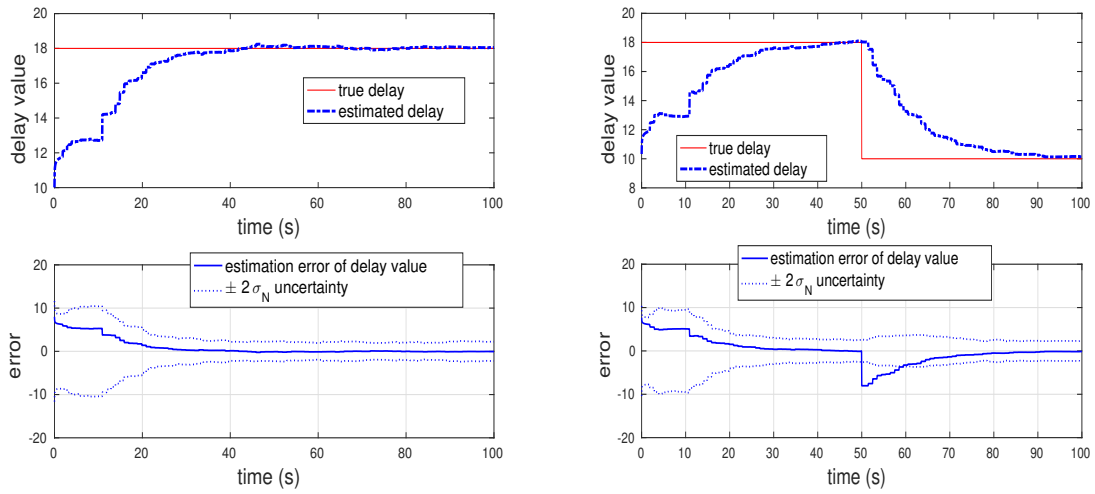


Figure 7. Estimation of total delays in simulation. (**a**) A static delay, (**b**) Varying delays.

Figure 7a shows that the estimated delay rapidly converges to the true delay value. That is, the estimation error of the delayed samples gradually decreases toward zero. Moreover, we may wonder whether the latency compensated filtering algorithm works when the delay is not static. See Figure 7b, which illustrates a response to a change in time delay. Although the values of unknown delays vary over time, estimation resulting from the online calibration method converges to true delay values.

5.2. Flight Datasets' Test Results

To validate the reliability of the proposed approach for estimating states and unknown delay values, we test one of the benchmark datasets, so-called "EuRoC MAV datasets [63]." The visual-inertial sequences of the datasets were recorded onboard a micro aerial vehicle while a pilot manually flew around indoor Vicon environments. For more details, see reference [63]. Although the datasets include noise model parameters from the IMU at rest, we need to tune each variance of process noise covariance Q for the best performance. Likewise, to estimate the unknown part of time delays, we set the standard

deviation of random walk η_d in Equation (48) as 1.0×10^{-5} since the order of this value is set to the same order of the smallest value among the provided noise parameters.

Given that datasets provide various levels of challenging sequences such as faster motion and poor illumination in each environment, to articulate the significance of time delays defined in Section 3.1, we select two datasets of slow motion, called "EuRoC V1 Easy," and fast motion, called "EuRoC V1 Medium." Since the vehicle in the medium dataset maneuvers twice as fast, we hypothesize that the time delays have greater impact on the navigation solution of the medium dataset. Algorithms of image processing and filtering are developed under the robot operating system (ROS) [64], given that IMU data and images from the stereo camera are also subscribed under the ROS, shown in Figure 8.

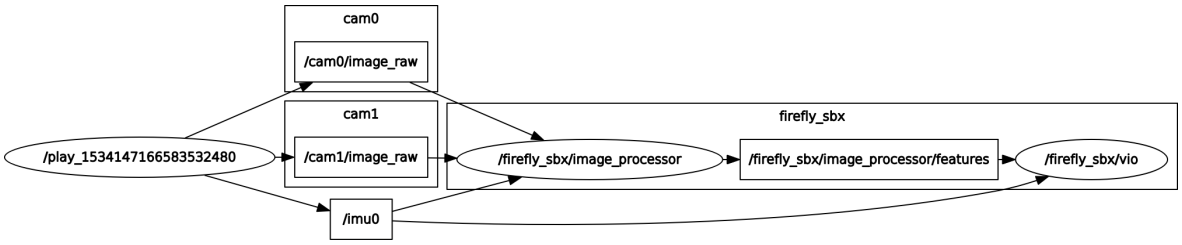


Figure 8. ROS rqt graph of the latency compensated VIO linked to the EuRoC dataset.

The simplest solution to the estimation problem of the given datasets is to run the baseline in Section 3.2.1 that corrects only Jacobians and residual. However, the novel latency compensated filter described in Algorithm 1 compensates for delayed measurements at the time when the vision data are fused at the filter and estimate the refined state and the delay values. This compensated filtering follows the processes of all three corrections, shown in Figure 6.

The EKF estimates relative location from a starting point. Since we do not know the exact absolute location of origin of given datasets, to compare with ground truth data given in the datasets, certain evaluation error metrics such as so-called "absolute trajectory error [65]" are required.

After applying the absolute trajectory error, Figure 9 illustrates the top-down view of the estimated flight trajectory of the medium dataset. Figure 10 exhibit estimated x, y, z position and their estimation errors.

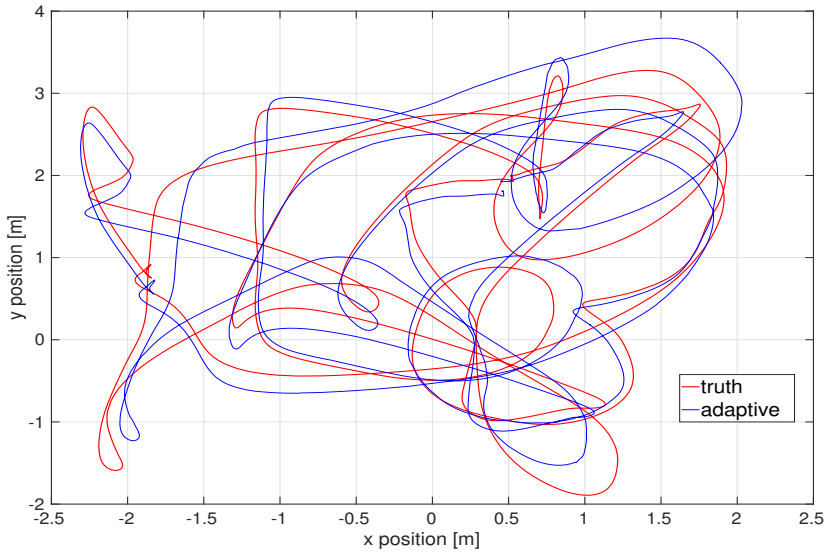


Figure 9. Top-down view of flight trajectory of the EuRoC V1 medium dataset by the latency compensated VIO.

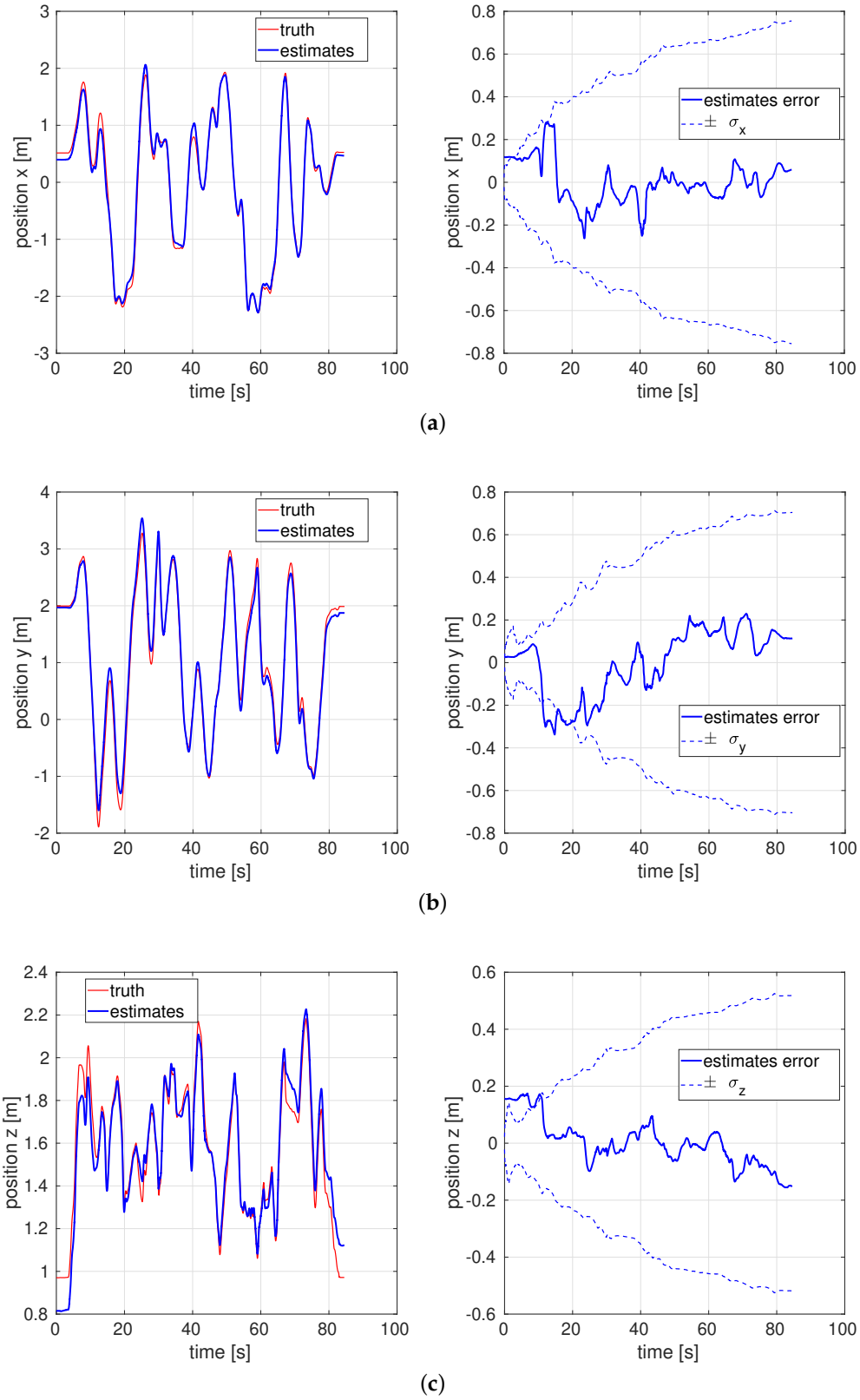


Figure 10. Position and estimation error of the EuRoC V1 medium dataset by the latency compensated VIO. (**a**) Position x, (**b**) Position y, (**c**) Position z.

All estimation errors are bounded within each standard deviation σ bounds. We should expect significant time correlation in error plots and a generally growing error covariance for vision-aided inertial navigation problems like this one. Conceptually, position error gets "locked in" and to the extent new features are being mapped the position error will tend to grow with the length of the trajectory. Starting from the noise model parameters reported for the datasets, the compensated filter is a well-tuned estimator—the performance of doing runs with $\times 3$ or $\times 10$ ($/3$ or $/10$) multiplier on the R term used in the filter is worse for all of those, shown in Table 2. In other words, the fact that using those multipliers shows larger root mean square (RMS) estimation errors indicates that our approach is a well-tuned filter.

Table 2. Indication that the latency compensated VIO is well-tuned for EuRoC V1 medium dataset.

Multiplier on R	**/10**	**/3**	**1**	**$\times$3**	**$\times$10**
RMS error [m]	1.5096	0.1969	0.1619	0.2636	0.2850

Figure 11 shows the advantages of each correction in the latency compensated VIO by comparing it with the baseline and the covariance correction. The baseline discards cross-covariance and unknown part of the delays, and, although the latency compensated filter might increase the computational effort of the entire system, it significantly improves the accuracy of estimation.

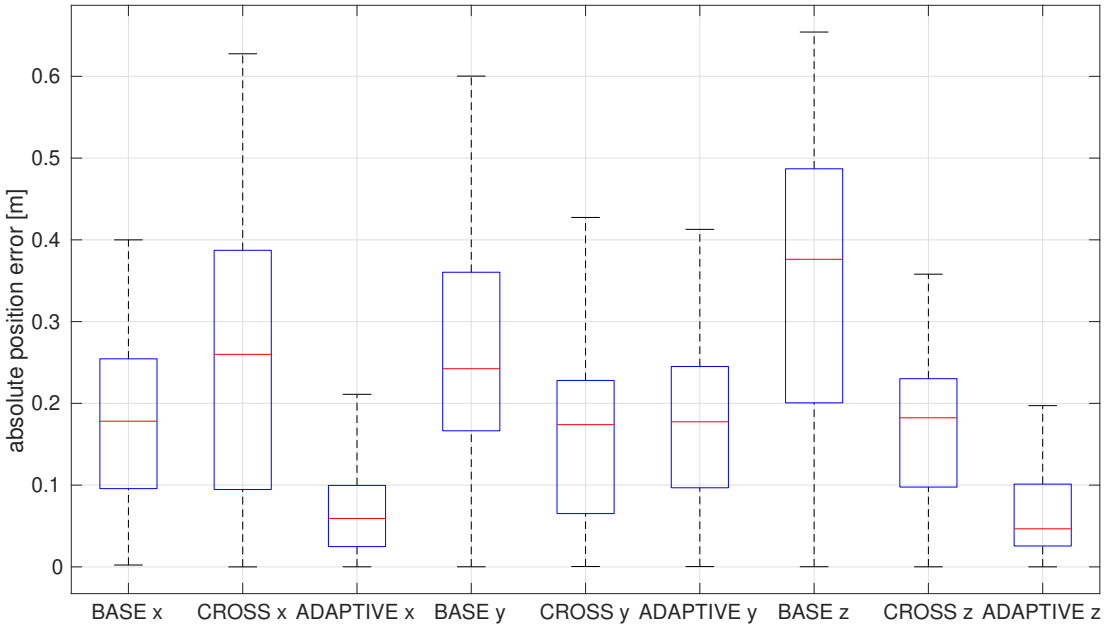


Figure 11. Box plot of absolute estimation error of position of the EuRoC V1 medium dataset by the latency compensated VIO.

Unlike either the baseline or the covariance correction, the latency compensated filter calibrates the unknown part of time delays. Figure 12 shows that estimation resulting from the compensated filter converges to a certain, final delay value, and its variance rapidly decreases although initial uncertainty is high.

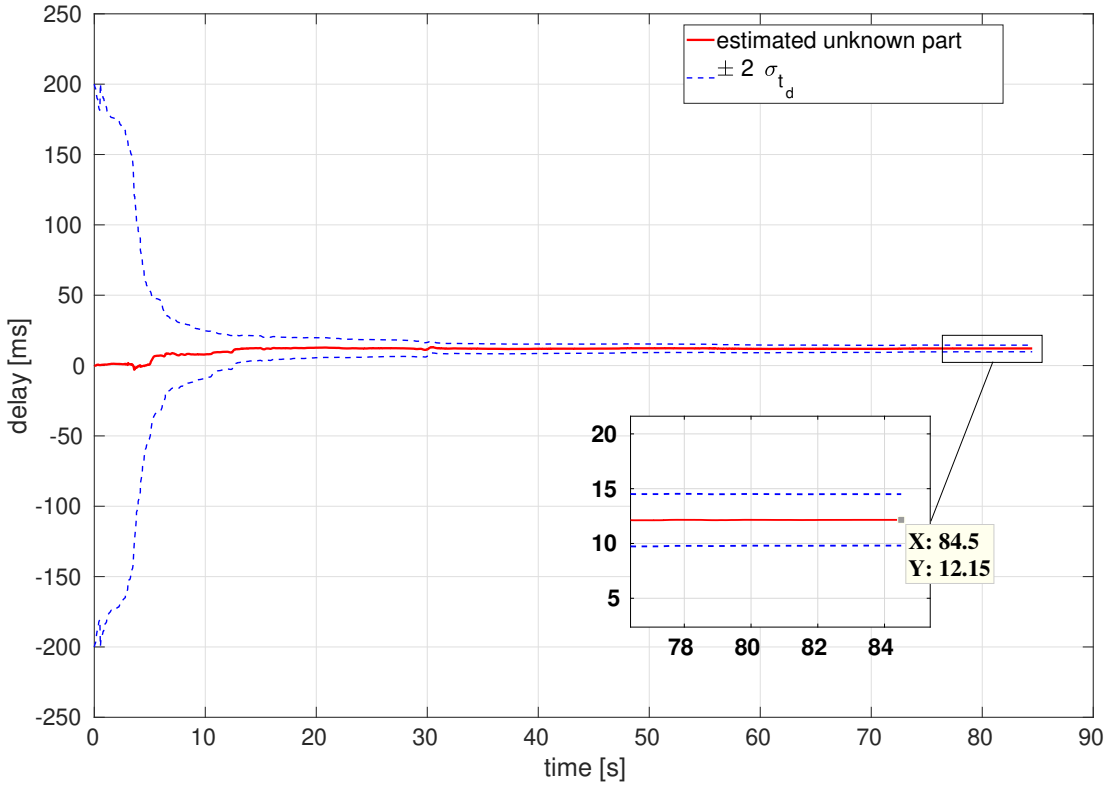


Figure 12. Estimation of unknown part of time delays of the EuRoC V1 medium dataset.

As shown in Figure 13, the average of estimated total delays is around 45 ms that could generate about 4 cm drift and offset during the delay period when the vehicle fly at 0.91 m/s average speed.

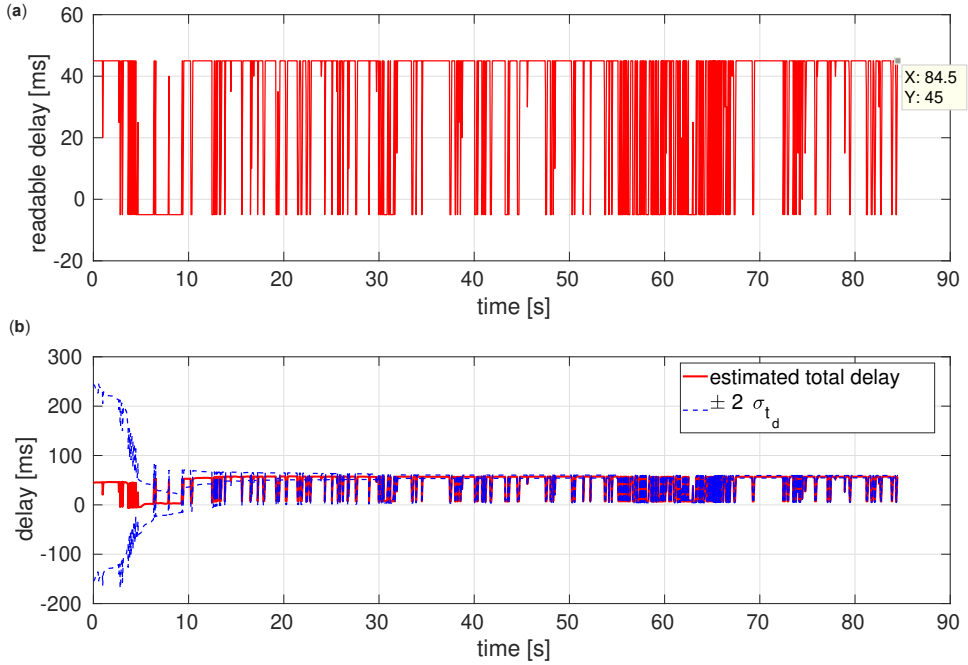


Figure 13. Estimation of time delays of the EuRoC V1 medium dataset. (**a**) Readable delays, (**b**) Estimated total delays.

When readable delay values are negative, the timestamps of images might indicate wrong pairs or packets.

Table 3 lists the RMS position errors of cases for sensitivity analysis. Approximately known part of time delays introduced in Section 3.2 are either fixed $\bar{t}_d$ by tuning or readouts $\bar{t}_{d_{\text{raw}}}$, which is the difference of readable timestamps of current IMU and image. In addition, we can directly estimate entire parts of time delays without information on the approximately known part. For another case, using the final value of the estimated unknown part of time delays, we add a fixed $\bar{\delta t}_d$ to the total delays at every time. However, this case might not work when the delay is varying, and we can know the final value only after running the proposed filter. In other words, before applying the compensation filtering, fixed $\bar{\delta t}_d$ is still unknown. The estimation results from the latency compensated VIO approach depict the influence of the delays and the effectiveness of the corrections in the sensor fusion of the lagged measurements.

Table 3. Sensitivity analysis in RMS position error [m] of latency compensated VIO.

Method \ Dataset		EuRoC V1 Easy *Slow Motion* 0.41 m/s, 16.0 deg/s		EuRoC V1 Medium *Fast Motion* 0.91 m/s, 32.1 deg/s	
		Cross-Cov OFF	**Cross-Cov ON**	**Cross-Cov OFF**	**Cross-Cov ON**
Fixed $\bar{t}_{d_{\text{const}}}$		0.3376	0.2677	0.4644	0.3135
Entirely Estimated $\hat{t}_d$		0.2282	0.2406	0.4734	0.3538
Readouts $\bar{t}_d$	+ N/A	0.2558	0.2032	0.4163	0.3121
	+ Fixed $\bar{\delta t}_d$	0.2869	0.2285	0.3281	0.2218
	+ Estimated $\hat{\delta t}_d$	0.2019	**0.1461**	0.3353	**0.1619**

Fast motion datasets are more sensitive to time delays since the improvement is larger when applied to those datasets.

Although numerous researchers have explored the VIO of the EuRoC datasets, few of them thoroughly considered measurements with unknown time delays. Table 4 reveals that the proposed estimator, the latency compensated VIO, outperforms the existing state-of-the-art methods, called "S-MSCKF" and "SVO+MSF" in which stereo is available.

Table 4. Comparison with other methods in RMS position error [m] of latency compensated VIO

Method \ Dataset	EuRoC V1 Easy	EuRoC V1 Medium
	Slow Motion 0.41 m/s, 0.28 rad/s	*Fast Motion* 0.91 m/s, 0.56 rad/s
Latency Compensated VIO	0.1461	0.1619
S-MSCKF (stereo-filter)	0.34	0.20
SVO+MSF (loosely-coupled)	0.40	0.63

6. Discussion

This study develops a practical extended Kalman filter (EKF)-based visual-inertial odometry (VIO) accounting for vehicle-feature correlations; that is, we develop tightly-coupled VIO for autonomous flight of unmanned aerial vehicles (UAVs). In particular, this paper has presented the development of a reliable and accurate filtering scheme for measurements with unknown time delays. We define time delays of vision data measurements in VIO. For compensating delayed measurements and estimating unknown delay values, this paper presents latency compensated filtering that includes state augmentation, interpolation, and residual, Jacobian, covariance corrections. The optimality of the three corrections and the observability of the state augmentation are validated; in other words, the appendix shows that the proposed latency compensated filter results in optimal estimates as if there were no delays in the data.

We test the performance of VIO employing the latency compensated filtering algorithms in the benchmark flight datasets for comparison to other state-of-the-art VIO algorithms. Results from flight

datasets testing show that the novel navigation approach in this paper improves the accuracy and reliability of state estimation with unknown time delays in VIO. With the latency compensated filtering, the root mean square (RMS) errors of estimation are decreased. In particular, we show improved accuracy of our method over previous approaches for state estimation in the fast motion datasets.

The overall approach in this document can be easily employed in other filter-based, sensor-aided inertial navigation frameworks and is suitable to monocular VIO although this study uses a stereo camera to showcase the methods. Although the reliability and robustness of this study are validated by testing benchmark flight datasets, validating with other datasets is of interest.

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 and 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 conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

VIO	Visual-Inertial Odometry
IMU	Inertial Measurement Unit
EKF	Extended Kalman Filter
UAV	Unmanned Aerial Vehicle
ROS	Robot Operating System

Nomenclature

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

Appendix A. Jacobians of Models

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

$$\frac{\partial\, {}^i\dot{\hat{p}}_{b/i}}{\partial\, {}^i\hat{v}_{b/i}} = I_{3\times3}, \quad \frac{\partial\, {}^i\dot{\hat{v}}_{b/i}}{\partial\, \delta\hat{\theta}} = -\hat{\mathcal{T}}_{i/b'} \left\lfloor\, (\, a_{\text{raw}} - \hat{b}_a\,) \times \right\rfloor, \quad \frac{\partial\, {}^i\dot{\hat{v}}_{b/i}}{\partial\, \hat{b}_a} = -\hat{\mathcal{T}}_{i/b},$$

$$\frac{\partial\, \delta\dot{\hat{\theta}}}{\partial\, \delta\hat{\theta}} = -\left\lfloor\, (\, \omega_{\text{raw}} - \hat{b}_\omega\,) \times \right\rfloor, \quad \frac{\partial\, \delta\dot{\hat{\theta}}}{\partial\, \hat{b}_\omega} = -I_{3\times3},$$

$$\frac{\partial\, {}^i\dot{\hat{v}}_{b/i}}{\partial\, \eta_a} = -\hat{\mathcal{T}}_{i/b}, \quad \frac{\partial\, \delta\dot{\hat{\theta}}}{\partial\, \eta_\omega} = -I_{3\times3}, \quad \frac{\partial\, \dot{b}_a}{\partial\, \eta_{b_a}} = I_{3\times3}, \quad \frac{\partial\, \dot{b}_\omega}{\partial\, \eta_{b_\omega}} = I_{3\times3},$$

$$C_j = \begin{bmatrix} \frac{\partial y_j}{\partial {}^i\hat{p}_{b/i}} & 0 & \frac{\partial y_j}{\partial \delta\hat{\theta}} & 0 & 0 & \big| 0 \cdots & \frac{\partial y_j}{\partial {}^i\hat{p}_{f_j/i}} & \cdots 0 \end{bmatrix} \tag{A2}$$

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

$$\frac{\partial y_j}{\partial \delta\hat{\theta}} = \left(\frac{\partial y_j}{\partial {}^c\hat{p}_{f_j/c}}\right)\mathcal{T}_{c/b}\left\lceil {}^{b'}\hat{p}_{f_j/b}\times\right\rfloor$$

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

where, for more detailed derivations, see reference [66].

Appendix B. Feature Initialization

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 ${}^c[X, Y, Z]^{\mathrm{T}}$ with respect to both cameras are

$${}^{c_2}p_{f_j/c_2} = \mathcal{T}_{c_2/c_1}\,{}^{c_1}p_{f_j/c_1} + {}^{c_2}p_{c_1/c_2} \tag{A5}$$

$$\begin{bmatrix} {}^{c_2}X_j & {}^{c_2}Y_j & {}^{c_2}Z_j \end{bmatrix}^{\mathrm{T}} = \mathcal{T}_{c_2/c_1}\begin{bmatrix} {}^{c_1}X_j & {}^{c_1}Y_j & {}^{c_1}Z_j \end{bmatrix}^{\mathrm{T}} + {}^{c_2}p_{c_1/c_2} \tag{A6}$$

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

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

Simplifying Equation (A8),

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

where m_x, m_y, and m_z are scalar functions of given j-th measurement and constant extrinsic rotation matrix.

Based on Equation (15), since 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:

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

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

where

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

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

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

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),

$${}^{c_1}\hat{p}_{f_j/c_1} = \begin{bmatrix} (\hat{u}_{j,1} \,/\, f_{u_1})\, {}^{c_1}\hat{Z} & ((\hat{v}_{j,1} \,/\, f_{v_1})\, {}^{c_1}\hat{Z} & {}^{c_1}\hat{Z} \end{bmatrix}^{\mathrm{T}}, \tag{A13}$$

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.

Appendix C. Stochastic Cloning (or the Schmidt–Kalman filter)

For shorthand expressions in this Appendix, we denote state x and covariance P without the augmented state by the feature initialization.

First, we prove Equations (43)–(46). During the delay period, cross-covariance term P^{crs} is propagated from time $k - \hat{d}$ to time $k - s$

$$\begin{aligned} &\begin{bmatrix} P_{k-\hat{d}} & P^{\mathrm{crs}}_{(k-s)\,|\,(k-\hat{d})}{}^{\mathrm{T}} \\ P^{\mathrm{crs}}_{(k-s)\,|\,(k-\hat{d})} & P_{k-s} \end{bmatrix} \\ &= \begin{bmatrix} I & 0 \\ 0 & \Phi_{k-\hat{d}} \end{bmatrix} \begin{bmatrix} P_{k-\hat{d}} & P_{k-\hat{d}} \\ P_{k-\hat{d}} & P_{k-\hat{d}} \end{bmatrix} \begin{bmatrix} I & 0 \\ 0 & \Phi_{k-\hat{d}} \end{bmatrix}^{\mathrm{T}} + \begin{bmatrix} 0 & 0 \\ 0 & Q\,\Delta t_{\mathrm{IMU}} \end{bmatrix} \\ &= \begin{bmatrix} P_{k-\hat{d}} & P_{k-\hat{d}} \Phi^{\mathrm{T}}_{k-\hat{d}} \\ \Phi_{k-\hat{d}} P_{k-\hat{d}} & \Phi_{k-\hat{d}} P_{k-\hat{d}} \Phi^{\mathrm{T}}_{k-\hat{d}} + Q\,\Delta t_{\mathrm{IMU}} \end{bmatrix}, \end{aligned}$$

where $P_{k-\hat{d}} \approx P^{\mathrm{dly}}$ and Φ denotes the state transition matrix.

After s time steps, at time k, the final cross-covariance term computed during the delay period is

$$\begin{bmatrix} P_{k-\hat{d}} & P^{\mathrm{crs}}_{k\,|\,(k-\hat{d})}{}^{\mathrm{T}} \\ P^{\mathrm{crs}}_{k\,|\,(k-\hat{d})} & P^{-}_{k} \end{bmatrix} = \begin{bmatrix} P_{k-\hat{d}} & P_{k-\hat{d}} \left(\prod_{i=k-\hat{d}}^{k-1} \Phi_i^{\mathrm{T}} \right) \\ \left(\prod_{i=k-1}^{k-\hat{d}} \Phi_i \right) P_{k-\hat{d}} & P^{-}_{k} \end{bmatrix}. \tag{A14}$$

Next, we prove Equations (37)–(42). The modified Kalman gain is computed as follows:

$$\begin{aligned}\begin{bmatrix} K^s \\ K^{\mathrm{crs}} \end{bmatrix} &= \begin{bmatrix} P_{k-\hat{d}} & P^{\mathrm{crs}}_{k\,|\,(k-\hat{d})}{}^{\mathrm{T}} \\ P^{\mathrm{crs}}_{k\,|\,(k-\hat{d})} & P^{-}_{k} \end{bmatrix} \begin{bmatrix} C_{k-\hat{d}} & 0 \end{bmatrix}^{\mathrm{T}} \left(C_{k-\hat{d}} P_{k-\hat{d}} C^{\mathrm{T}}_{k-\hat{d}} + R \right)^{-1} \\ &= \begin{bmatrix} * \\ P^{\mathrm{crs}}_{k\,|\,(k-\hat{d})} C^{\mathrm{T}}_{k-\hat{d}} \left(C_{k-\hat{d}} P_{k-\hat{d}} C^{\mathrm{T}}_{k-\hat{d}} + R \right)^{-1} \end{bmatrix}, \end{aligned} \tag{A15}$$

where K^s denotes the stationary Kalman gain and

$$C_{k-\hat{d}} P_{k-\hat{d}} C^{\mathrm{T}}_{k-\hat{d}} = \begin{bmatrix} C_{k-\hat{d}} & 0 \end{bmatrix} \begin{bmatrix} P_{k-\hat{d}} & P^{\mathrm{crs}}_{k\,|\,(k-\hat{d})}{}^{\mathrm{T}} \\ P^{\mathrm{crs}}_{k\,|\,(k-\hat{d})} & P^{-}_{k} \end{bmatrix} \begin{bmatrix} C_{k-\hat{d}} & 0 \end{bmatrix}^{\mathrm{T}}.$$

The update of covariance matrix using the cross-covariance term is

$$\begin{aligned}\begin{bmatrix} * & * \\ * & P^{+}_{k} \end{bmatrix} &= \begin{bmatrix} P_{k-\hat{d}} & P^{\mathrm{crs}}_{k\,|\,(k-\hat{d})}{}^{\mathrm{T}} \\ P^{\mathrm{crs}}_{k\,|\,(k-\hat{d})} & P^{-}_{k} \end{bmatrix} - \begin{bmatrix} K^s \\ K^{\mathrm{crs}} \end{bmatrix} \begin{bmatrix} C_{k-\hat{d}} & 0 \end{bmatrix} \begin{bmatrix} P_{k-\hat{d}} & P^{\mathrm{crs}}_{k\,|\,(k-\hat{d})}{}^{\mathrm{T}} \\ P^{\mathrm{crs}}_{k\,|\,(k-\hat{d})} & P^{-}_{k} \end{bmatrix} \\ &= \begin{bmatrix} * & * \\ * & P^{-}_{k} - K^{\mathrm{crs}} C_{k-\hat{d}} P^{\mathrm{crs}}_{k\,|\,(k-\hat{d})}{}^{\mathrm{T}} \end{bmatrix}. \end{aligned} \tag{A16}$$

We finally prove the optimality of the latency compensated filtering in this paper. Since the standard EKF is an optimal estimator, if we prove that the latency compensated filter is identical to the standard EKF, then the latency compensated filtering approach becomes also optimal estimation.

Let us recall Equations (37)–(39):

$$\begin{aligned} K^{\mathrm{crs}} &= P^{\mathrm{crs}}_{k\,|\,(k-\hat{d})} C^{\mathrm{T}}_{k-\hat{d}} \left(C_{k-\hat{d}} P_{k-\hat{d}} C^{\mathrm{T}}_{k-\hat{d}} + R \right)^{-1} \\ &= \Phi_{k-1}\Phi_{k-2}\cdots\Phi_{k-\hat{d}} P_{k-\hat{d}} C^{\mathrm{T}}_{k-\hat{d}} \left(C_{k-\hat{d}} P_{k-\hat{d}} C^{\mathrm{T}}_{k-\hat{d}} + R \right)^{-1} && \text{(A17)} \\ \hat{x}^{+}_{k} &= \hat{x}^{-}_{k} + K^{\mathrm{crs}} r_{k-\hat{d}} && \text{(A18)} \\ P^{+}_{k} &= P^{-}_{k} - K^{\mathrm{crs}} C_{k-\hat{d}} P^{\mathrm{crs}}_{k\,|\,(k-\hat{d})}{}^{\mathrm{T}} \\ &= P^{-}_{k} - K^{\mathrm{crs}} C_{k-\hat{d}} P_{k-\hat{d}} \Phi^{\mathrm{T}}_{k-\hat{d}} \Phi^{\mathrm{T}}_{k-s} \cdots \Phi^{\mathrm{T}}_{k-1}, && \text{(A19)} \end{aligned}$$

Next, we assume that delayed measurement y is available immediately without delays. In other words, for this assumed case, measurement update is first performed and then propagation steps are processed. At time $[k-\hat{d}]$, given $\hat{x}^{-}_{k-\hat{d}}$ and $P^{-}_{k-\hat{d}}$, the standard measurement update performs as follows:

$$\begin{aligned} K_{k-\hat{d}} &= P^{-}_{k-\hat{d}} C^{\mathrm{T}}_{k-\hat{d}} \left(C_{k-\hat{d}} P^{-}_{k-\hat{d}} C^{\mathrm{T}}_{k-\hat{d}} + R \right)^{-1} \\ \hat{x}'^{+}_{k-\hat{d}} &= \hat{x}^{-}_{k-\hat{d}} + K_{k-\hat{d}} r_{k-\hat{d}} \\ P'^{+}_{k-\hat{d}} &= P^{-}_{k-\hat{d}} - K_{k-\hat{d}} C_{k-\hat{d}} P^{-}_{k-\hat{d}}, \end{aligned}$$

where $\hat{x}'^{+}_{k-\hat{d}} \neq \hat{x}_{k-\hat{d}}$ and $P'^{+}_{k-\hat{d}} \neq P_{k-\hat{d}}$ since $\hat{x}'^{+}_{k-\hat{d}}$ and $P'^{+}_{k-\hat{d}}$ are values after the corrections by the measurement update, shown in Figure A1. That is, red lines illustrate the original processes of the latency compensated filtering and blue lines present the processes of the assumed case.

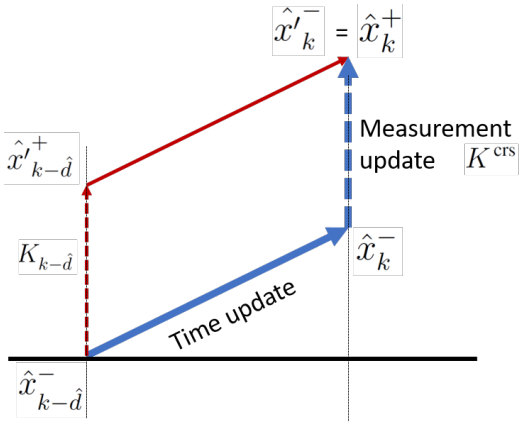


Figure A1. Optimality of the latency compensated VIO.

From time $(k-s)$ to time k during the delay period, the assumed case propagates state estimates and covariance recursively. At time $(k-s)$,

$$\begin{aligned}
\hat{x'}_{k-s}^- &\approx \Phi_{k-\hat{d}}\,\hat{x'}_{k-\hat{d}}^+ \\
&= \Phi_{k-\hat{d}}\,\hat{x}_{k-\hat{d}}^- + \Phi_{k-\hat{d}}\,K_{k-\hat{d}}\,r_{k-\hat{d}} \\
P'^{-}_{k-\hat{d}+1} &= \Phi_{k-\hat{d}}\,P'^{+}_{k-\hat{d}}\,\Phi_{k-\hat{d}}^{\mathrm{T}} + Q\,\Delta t_{\mathrm{IMU}} \\
&= \Phi_{k-\hat{d}}\,P_{k-\hat{d}}^-\,\Phi_{k-\hat{d}}^{\mathrm{T}} + Q\,\Delta t_{\mathrm{IMU}} - \Phi_{k-\hat{d}}\,K_{k-\hat{d}}C_{k-\hat{d}}P_{k-\hat{d}}^-\,\Phi_{k-\hat{d}}^{\mathrm{T}}.
\end{aligned}$$

Likely, at time k, only time update is processed since the measurement was already used to update:

$$\begin{aligned}
\hat{x}_k^+ = \hat{x'}_k^- &\approx \Phi_{k-1}\hat{x'}_{k-1}^+ \\
&= \Phi_{k-1}\Phi_{k-2}\cdots\Phi_{k-\hat{d}}\,\hat{x}_{k-\hat{d}}^- + \Phi_{k-1}\Phi_{k-2}\cdots\Phi_{k-\hat{d}}\,K_{k-\hat{d}}\,r_{k-\hat{d}} \\
&= \hat{x}_k^- + \Phi_{k-1}\Phi_{k-2}\cdots\Phi_{k-\hat{d}}\,K_{k-\hat{d}}\,r_{k-\hat{d}} \\
&= \hat{x}_k^- + K^{\mathrm{crs}}\,r_{k-\hat{d}}
\end{aligned} \tag{A20}$$

$$\begin{aligned}
P_k^+ = P'^{-}_k &= \Phi_{k-1}\Phi_{k-2}\cdots\Phi_{k-\hat{d}}\,P_{k-\hat{d}}^-\,\Phi_{k-\hat{d}}^{\mathrm{T}}\Phi_{k-s}^{\mathrm{T}}\cdots\Phi_{k-1}^{\mathrm{T}} \\
&\quad + Q\,\Delta t_{\mathrm{IMU}} + \Phi_{k-1}Q\,\Delta t_{\mathrm{IMU}}\,\Phi_{k-1}^{\mathrm{T}} + \cdots \\
&\quad - \Phi_{k-1}\Phi_{k-2}\cdots\Phi_{k-\hat{d}}\,K_{k-\hat{d}}\,C_{k-\hat{d}}P_{k-\hat{d}}^-\,\Phi_{k-\hat{d}}^{\mathrm{T}}\Phi_{k-s}^{\mathrm{T}}\cdots\Phi_{k-1}^{\mathrm{T}} \\
&= P_k^- - \Phi_{k-1}\Phi_{k-2}\cdots\Phi_{k-\hat{d}}\,K_{k-\hat{d}}\,C_{k-\hat{d}}P_{k-\hat{d}}^-\,\Phi_{k-\hat{d}}^{\mathrm{T}}\Phi_{k-s}^{\mathrm{T}}\cdots\Phi_{k-1}^{\mathrm{T}} \\
&= P_k^- - K^{\mathrm{crs}}\,C_{k-\hat{d}}P_{k-\hat{d}}^-\,\Phi_{k-\hat{d}}^{\mathrm{T}}\Phi_{k-s}^{\mathrm{T}}\cdots\Phi_{k-1}^{\mathrm{T}}.
\end{aligned} \tag{A21}$$

Since Equations (A18) and (A19) are identical to Equations (A20) and (A21), respectively, the hypothesis was completely proved. In other words, the latency compensated filter for VIO acts as if the delayed vision data from an image are available at the right time when the image was captured.

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; pp. 193–222.

4. Bar-Shalom, Y.; Li, X.R.; Kirubarajan, T. *Estimation with Applications to Tracking and Navigation: Theory Algorithms and Software*; John Wiley & Sons: River Street Hoboken, NJ, USA, 2004.
5. Julier, S.J.; Uhlmann, J.K. A New Extension of the Kalman Filter to Nonlinear Systems. In Proceedings of the International Symposium on Aerospace/Defense, Sensing, Simulation and Controls, Orlando, FL, USA, 20–25 April 1997; Volume 3, pp. 182–193. [CrossRef]
6. 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. Inform. Syst.* **2013**, *10*, 172–186. [CrossRef]
7. Chowdhary, G.; Johnson, E.N.; Magree, D.; Wu, A.; Shein, A. GPS-Denied Indoor and Outdoor Monocular Vision Aided Navigation and Control of Unmanned Aircraft. *J. Field Robot.* **2013**, *30*, 415–438. [CrossRef]
8. 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]
9. Kong, W.; Hu, T.; Zhang, D.; Shen, L.; Zhang, J. Localization Framework for Real-Time UAV Autonomous Landing: An On-Ground Deployed Visual Approach. *Sensors* **2017**, *17*, 1437. [CrossRef]
10. 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. [CrossRef]
11. Triggs, B.; McLauchlan, P.F.; Hartley, R.I.; Fitzgibbon, A.W. Bundle Adjustment—A Modern Synthesis. In Proceedings of the International Workshop on Vision Algorithms, Corfu, Greece, 21–22 September 1999; pp. 298–372. [CrossRef]
12. Dellaert, F.; Kaess, M. Square Root SAM: Simultaneous Localization and Mapping via Square Root Information Smoothing. *Int. J. Rob. Res.* **2006**, *25*, 1181–1203. [CrossRef]
13. Huang, G. Visual-Inertial Navigation: A Concise Review. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019. [CrossRef]
14. Cadena, C.; Carlone, L.; Carrillo, H.; Latif, Y.; Scaramuzza, D.; Neira, J.; Reid, I.; Leonard, J.J. Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age. *IEEE Trans. Robot.* **2016**, *32*, 1309–1332. [CrossRef]
15. Bloesch, M.; Omari, S.; Hutter, M.; Siegwart, R. Robust Visual Inertial Odometry Using a Direct EKF-Based Approach. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, Germany, 28 September–2 October 2015; pp. 298–304. [CrossRef]
16. 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]
17. 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]
18. Paul, M.K.; Roumeliotis, S.I. Alternating-Stereo VINS: Observability Analysis and Performance Evaluation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 4729–4737. [CrossRef]
19. Sun, K.; Mohta, K.; Pfrommer, B.; Watterson, M.; Liu, S.; Mulgaonkar, Y.; Taylor, C.J.; Kumar, V. Robust Stereo Visual Inertial Odometry for Fast Autonomous Flight. *IEEE Robot. Autom. Lett.* **2018**, *3*, 965–972. [CrossRef]
20. Leutenegger, S.; Lynen, S.; Bosse, M.; Siegwart, R.; Furgale, P. Keyframe-Based Visual–Inertial Odometry Using Nonlinear Optimization. *Int. J. Rob. Res.* **2015**, *34*, 314–334. [CrossRef]
21. Lin, Y.; Gao, F.; Qin, T.; Gao, W.; Liu, T.; Wu, W.; Yang, Z.; Shen, S. Autonomous Aerial Navigation Using Monocular Visual-Inertial Fusion. *J. Field Robot.* **2018**, *35*, 23–51. [CrossRef]
22. 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]
23. Li, M.; Mourikis, A.I. High-Precision, Consistent EKF-Based Visual-Inertial Odometry. *Int. J. Rob. Res.* **2013**, *32*, 690–711. [CrossRef]
24. Forster, C.; Pizzoli, M.; Scaramuzza, D. SVO: Fast Semi-Direct Monocular Visual Odometry. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China, 31 May–7 June 2014; pp. 15–22. [CrossRef]

25. 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]
26. Lynen, S.; Achtelik, M.W.; Weiss, S.; Chli, M.; Siegwart, R. A Robust and Modular Multi-Sensor Fusion Approach Applied to MAV Navigation. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Tokyo, Japan, 3–7 November 2013; pp. 3923–3929. [CrossRef]
27. 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 International Conference on Robotics and Automation (ICRA), Singapore, 29 May–3 June 2017; pp. 165–172.
28. Alexander, H.L. State Estimation for Distributed Systems with Sensing Delay. In Proceedings of the Data Structures and Target Classification, Orlando, FL, USA, 1–2 April 1991; Volume 1470, pp. 103–111.
29. Larsen, T.D.; Andersen, N.A.; Ravn, O.; Poulsen, N.K. Incorporation of Time Delayed Measurements in a Discrete-Time Kalman Filter. In Proceedings of the 37th IEEE Conference on Decision and Control (CDC), Tampa, FL, USA, 16–18 December 1998; Volume 4, pp. 3972–3977. [CrossRef]
30. Van Der Merwe, R.; Wan, E.A.; Julier, S. Sigma-Point Kalman Filters for Nonlinear Estimation and Sensor-Fusion: Applications to Integrated Navigation. In Proceedings of the AIAA Guidance, Navigation, and Control Conference, Providence, RI, USA, 16–19 August 2004; pp. 16–19. [CrossRef]
31. Van Der Merwe, R. Sigma-Point Kalman Filters for Probabilistic Inference in Dynamic State-Space Models. Ph.D. Thesis, Oregon Health & Science University, Portland, OR, USA, 2004.
32. Schmidt, S.F. Applications of State Space Methods to Navigation Problems. *Adv. Control Syst.* **1966**, *3*, 293–340. [CrossRef]
33. Roumeliotis, S.I.; Burdick, J.W. Stochastic Cloning: A Generalized Framework for Processing Relative State Measurements. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Washington, DC, USA, 11–15 May 2002; Volume 2, pp. 1788–1795. [CrossRef]
34. Gopalakrishnan, A.; Kaisare, N.S.; Narasimhan, S. Incorporating Delayed and Infrequent Measurements in Extended Kalman Filter Based Nonlinear State Estimation. *J. Process Contr.* **2011**, *21*, 119–129. [CrossRef]
35. Julier, S.J.; Uhlmann, J.K. Fusion of Time Delayed Measurements with Uncertain Time Delays. In Proceedings of the American Control Conference (ACC), Portland, OR, USA, 8–10 June 2005; pp. 4028–4033. [CrossRef]
36. Sinopoli, B.; Schenato, L.; Franceschetti, M.; Poolla, K.; Jordan, M.I.; Sastry, S.S. Kalman Filtering with Intermittent Observations. *IEEE Trans. Automat. Contr.* **2004**, *49*, 1453–1464. [CrossRef]
37. Choi, M.; Choi, J.; Park, J.; Chung, W.K. State Estimation with Delayed Measurements Considering Uncertainty of Time Delay. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Kobe, Japan, 12–17 May 2009; pp. 3987–3992. [CrossRef]
38. Yoon, H.; Sternberg, D.C.; Cahoy, K. Interpolation Method for Update with out-of-Sequence Measurements: The Augmented Fixed-Lag Smoother. *J. Guid. Control Dyn.* **2016**, *39*, 2546–2553. [CrossRef]
39. Lee, K.; Johnson, E.N. Multiple-Model Adaptive Estimation for Measurements with Unknown Time Delay. In Proceedings of the AIAA Guidance, Navigation, and Control Conference, Grapevine, TX, USA, 9–13 January 2017. [CrossRef]
40. Nilsson, J.O.; Skog, I.; Händel, P. Joint State and Measurement Time-Delay Estimation of Nonlinear State Space Systems. In Proceedings of the 10th IEEE International Conference on Information Sciences, Signal Processing and their Applications (ISSPA), Kuala Lumpur, Malaysia, 10–13 May 2010; pp. 324–328. [CrossRef]
41. Li, M.; Mourikis, A.I. Online Temporal Calibration for Camera–IMU systems: Theory and Algorithms. *Int. J. Rob. Res.* **2014**, *33*, 947–964. [CrossRef]
42. Qin, T.; Shen, S. Online Temporal Calibration for Monocular Visual-Inertial Systems. In Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1–5 October 2018; pp. 3662–3669. [CrossRef]
43. Lee, K.; Johnson, E.N. State and Parameter Estimation Using Measurements with Unknown Time Delay. In Proceedings of the IEEE Conference on Control Technology and Applications (CCTA), Mauna Lani, HI, USA, 27–30 August 2017; pp. 1402–1407. [CrossRef]
44. Lee, K. Adaptive Filtering for Vision-Aided Inertial Navigation. Ph.D. Thesis, Georgia Institute of Technology, Atlanta, GA, USA, 2018.

45. Simon, D. *Optimal State Estimation: Kalman, H Infinity, and Nonlinear Approaches*; John Wiley & Sons: Hoboken, NJ, USA, 2006.
46. 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.
47. Sola, J. Quaternion Kinematics for the Error-State Kalman Filter. *arXiv* **2017**, arXiv:1711.02508.
48. Markley, F.L. Attitude Error Representations for Kalman Filtering. *J. Guid. Control Dyn.* **2003**, *26*, 311–317. [CrossRef]
49. Forsyth, D.A.; Ponce, J. *Computer Vision: A Modern Approach;* Prentice Hall: Upper Saddle River, NJ, USA, 2002; pp. 96–147.
50. Hartley, R.; Zisserman, A. *Multiple View Geometry in Computer Vision*; Cambridge University Press: Cambridge, UK, 2003; pp. 22–236.
51. Bjorck, A. *Numerical Methods for Least Squares Problems*; Society for Industrial and Applied Mathematics (SIAM): Philadelphia, PA, USA, 1996; Volume 51, Chapters 2–9.
52. Montiel, J.M.; Civera, J.; Davison, A.J. Unified Inverse Depth Parametrization for Monocular SLAM. In Proceedings of the Robotics: Science and Systems (RSS), Philadelphia, PA, UAS, 16–19 August 2006. [CrossRef]
53. Schutz, B.; Tapley, B.; Born, G.H. *Statistical Orbit Determination*; Elsevier Academic Press: Burlington, MA, USA, 2004; pp. 387–438.
54. Hinks, J.; Psiaki, M. A Multipurpose Consider Covariance Analysis for Square-Root Information Filters. In Proceedings of the AIAA Guidance, Navigation, and Control Conference, Minneapolis, MN, USA, 13–16 August 2012; p. 4599. [CrossRef]
55. Meijering, E. A Chronology of Interpolation: From Ancient Astronomy to Modern Signal and Image Processing. *Proc. IEEE* **2002**, *90*, 319–342. [CrossRef]
56. Shoemake, K. Animating Rotation with Quaternion Curves. In Proceedings of the ACM SIGGRAPH Computer Graphics, San Francisco, CA, USA, 20–28 July 1985; Volume 19, pp. 245–254. [CrossRef]
57. Dam, E.B.; Koch, M.; Lillholm, M. *Quaternions, Interpolation and Animation*; Citeseer: Copenhagen, Denmark, 1998; Volume 2, pp. 34–76.
58. Gelb, A. *Applied Optimal Estimation*; MIT Press: Cambridge, MA, USA, 1974; pp. 102–155, 180–228.
59. Kopp, R.E.; Orford, R.J. Linear Regression Applied to System Identification for Adaptive Control Systems. *AIAA J.* **1963**, *1*, 2300–2306. [CrossRef]
60. Viegas, D.; Batista, P.; Oliveira, P.; Silvestre, C. Nonlinear Observability and Observer Design through State Augmentation. In Proceedings of the IEEE 53rd Annual Conference on Decision and Control (CDC), Los Angeles, CA, USA, 15–17 December 2014; pp. 133–138. [CrossRef]
61. Del Vecchio, D.; Murray, R. Observability and Local Observer Construction for Unknown Parameters in Linearly and Nonlinearly Parameterized Systems. In Proceedings of the American Control Conference (ACC), Denver, CO, USA, 4–6 June 2003; Volume 6, pp. 4748–4753. [CrossRef]
62. Carrassi, A.; Vannitsem, S. State and Parameter Estimation with the Extended Kalman Filter: An Alternative Formulation of the Model Error Dynamics. *Q. J. R. Meteorol. Soc.* **2011**, *137*, 435–451. [CrossRef]
63. 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. Rob. Res.* **2016**, *35*, 1157–1163. [CrossRef]
64. Quigley, M.; Conley, K.; Gerkey, B.; Faust, J.; Foote, T.; Leibs, J.; Wheeler, R.; Ng, A.Y. ROS: An Open-Source Robot Operating System. In Proceedings of the ICRA Workshop on Open Source Software, Kobe, Japan, 12–17 May 2009; Volume 3, p. 5.
65. 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]
66. 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.