sensors

MDPI

Article

MAV Localization in Large-Scale Environments: A Decoupled Optimization/Filtering Approach

Abanob Soliman, Hicham Hadj-Abdelkader, Fabien Bonardi, Samia Bouchafa and Désiré Sidibé *

IBISC Laboratory, Université d'Evry-Paris Saclay, 91020 Evry-Courcouronnes, France
* Correspondence: drodesire.sidibie@univ-evry.fr

Abstract: Developing new sensor fusion algorithms has become indispensable to tackle the daunting problem of GPS-aided micro aerial vehicle (MAV) localization in large-scale landscapes. Sensor fusion should guarantee high-accuracy estimation with the least amount of system delay. Towards this goal, we propose a linear optimal state estimation approach for the MAV to avoid complicated and high-latency calculations and an immediate metric-scale recovery paradigm that uses low-rate noisy GPS measurements when available. Our proposed strategy shows how the vision sensor can quickly bootstrap a pose that has been arbitrarily scaled and recovered from various drifts that affect vision-based algorithms. We can consider the camera as a "black-box" pose estimator thanks to our proposed optimization/filtering-based methodology. This maintains the sensor fusion algorithm's computational complexity and makes it suitable for MAV's long-term operations in expansive areas. Due to the limited global tracking and localization data from the GPS sensors, our proposal on MAV's localization solution considers the sensor measurement uncertainty constraints under such circumstances. Extensive quantitative and qualitative analyses utilizing real-world and large-scale MAV sequences demonstrate the higher performance of our technique in comparison to most recent state-of-the-art algorithms in terms of trajectory estimation accuracy and system latency.

Keywords: MAV; multimodal sensing; localization; odometry; visual drifts; sensor fusion; Kalman filter; calibration; optimization

Citation: Soliman, A.; Hadj-Abdelkader, H.; Bonardi, F.; Bouchafa, S.; Sidibé, D. MAV Localization in Large-Scale Environments: A Decoupled Optimization/Filtering Approach. *Sensors* **2023**, *23*, 516. https://doi.org/10.3390/s23010516

Academic Editor: Gregor Klancar

Received: 31 October 2022
Revised: 28 December 2022
Accepted: 29 December 2022
Published: 3 January 2023

1. Introduction

Robust localization of micro aerial vehicles (MAVs) in uncharted large-scale areas can rely on complementary data gathered by many sensor modalities. The study of simultaneous localization and mapping (SLAM), primarily used for MAV navigation in expansive and dynamic settings, may be enriched and expanded by using multi-modal datasets [1]. These settings have certain traits, such as the dynamic range of the scene's object intensities. For instance, mapping a small interior space with adequate illumination might be of more outstanding quality than mapping a rural area at night with heavy rain, wind, and fog (outdoors dynamic environment). The benefits of multimodal approaches become apparent when systems rely on sensors with a high dynamic range and strong sensing capabilities, such as event cameras, LiDARs, or radars, or typical inexpensive cameras fused with other sensor modalities such as the inertial measurement units (IMUs) and GPS sensors. These multimodal approaches can indeed fill some lack of data during scene mapping and MAV localization.

Towards this aim, we develop a trustworthy (quick and precise) localization solution that utilizes information from three sensor modalities: camera frame data, IMU measurements, and GPS readings. Nevertheless, the GPS sensor readings are consistently slower and noisier than those from the IMU or camera modules, and they frequently experience signal loss in GPS-restricted locations. Therefore, a localization system that depends on GPS data must perform effectively when GPS readings are lost.

Visual-inertial odometry (VIO) is one of the most mature and well-established approaches in the localization field [2–4]. Efficient visual odometry can be achieved using a high-quality perception of the surroundings. Sensors performing this perception task can differ in their nature of data collection. On the one hand, the most common visual odometry sensors are cameras such as RGB cameras [5], event cameras [6], and RGB-D cameras [7]. On the other hand, using the LiDAR sensor [8] can provide point clouds and a GPS sensor [9,10] can locate the MAV using satellite signals triangulation, as represented in Figure 1.

Figure 1. An example for the on-map GPS readings of the large-scale environment of the Fast Flight dataset [11] sequences: gps175, gps15, gps10, and gps5. The sequence number denotes the maximum flight velocity of each sequence: 17.5, 15, 10, and 5 (m/s), respectively. The color bar (bottom) denotes the map scale in (km) on the x axis and the altitude of each sequence in (m) on the y axis. In the blue dotted box: Comparing the maximum MAV's altitude at instance before the descent stage to the height of an aircraft hangar. The estimated airport asset height is 54.72 (m), corresponding to the maximum MAV altitude. Images are courtesy of Google Earth.

The accuracy of the state estimation process relies on an error-state extended Kalman filter (ES-EKF) and the bootstrapping quality of its states. A well-established IMU-based state estimator initialization technique was discussed in [5]. In this bootstrapping method, the global metric scale of the trajectory and the IMU-camera gravity alignment is optimized using a specific amount of IMU readings preintegration combined with an initial up-to-scale trajectory estimated using the camera only. This bootstrapping process is prone to failure due to insufficient IMU excitation, especially when the MAV navigates in a planar terrain.

The MAV should contain a localization system that continually calculates the pose with high accuracy and low latency during search and rescue missions, for instance. The MAV is equipped with restricted resources regarding the data processing unit and the limited power source capacity for long-term navigation operations in large-scale situations. In light of this, the state estimate approach should consistently have low computational complexity and resist sensor readings that deviate from the norm.

Our work's main contribution to tackle the aforementioned challenges is three-fold:

- In the case of state estimator initialization failure, we propose a unique instant bootstrapping technique based on continuous-time manifold optimization via pose graph optimization (PGO) and range factors, which depends on low-rate GPS signals.
- A closed-form estimation method without nonlinear optimization during IMU/CAM fusion produces a reduced system latency with constant CPU computing complexity. The mathematical modeling of a linear ES-EKF with a precise and quick gyroscope integration strategy accounts for the simplicity of our proposed localization solution.
- The EuRoC benchmark [12], for MAV localization assessment in indoor environments, and the Fast Flight dataset [11], for large-scale outdoor environments, are two real-world publicly available benchmarks on which our IMU/GPS-CAM fusion

system has been thoroughly tested. With thorough ablation investigations into the role of each sensor modality in the overall accuracy of the state estimation process, the assessment is conducted using the most recent state-of-the-art visual-inertial odometry methodologies.

2. Related Work

2.1. Sensor Fusion

Figure 2 presents a global overview of the current state-of-the-art approaches for localization. The ability to continually estimate the robot's ego-motion (position and orientation) over time is a significant difficulty in autonomous navigation, path planning, object tracking, and collision avoidance platforms [13]. The Global Positioning System (GPS) is a well-known localization method applied to several autonomous system domains. One kind of global navigation satellite system (GNSS) is GPS [10]. GPS is used as a self-localization source, such as for MAVs security applications, and gives any user with a GPS receiver positional information with meter-level precision. The satellite signal blockage, high noise levels, multipath effects, and other issues with GPS, on the other hand, make it a less trustworthy alternative sensor for self-localization modules. However, real-time kinematic (RTK) and precise point positioning (PPP) [9], two GPS technologies that are rapidly developing, can provide locations with decimeter- or centimeter-level precision.

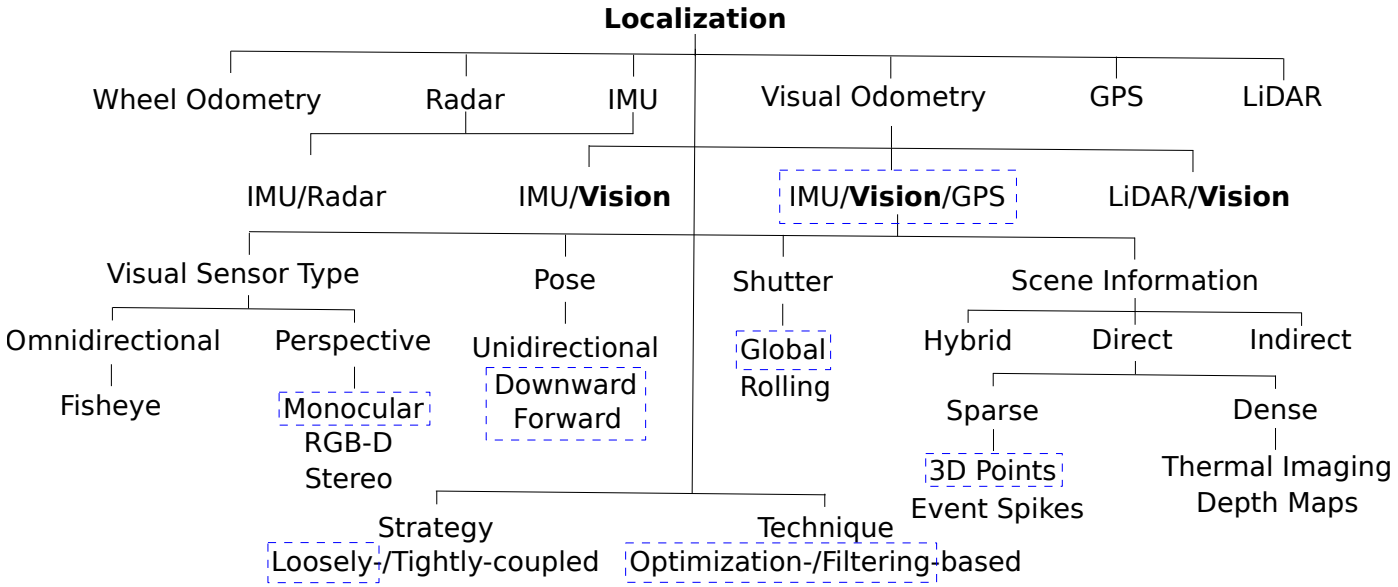

Figure 2. Visual odometry is generally categorized together with self-contained and global localization methods.

The effectiveness of GPS satellite signals heavily depends on the surrounding environment; it works best in locations with clear skies and is ineffective for inside navigation since walls and other obstacles impede it [14]. This makes the GPS module an unsuitable primary sensor for reliable autonomous vehicle localization under adverse weather and environmental conditions. Hence, the fusion of GPS signals with other inertial and/or visual sensors is indispensable for a reliable localization solution, especially in such environments. The state-of-the-art sensor fusion systems are differentiated into two prominent families: loosely- [15], and tightly-coupled [16] fusion strategies. In loosely-coupled fusion, the camera frames for pose estimation are processed as a black-box. A filter or an optimization model is developed to fuse the arbitrary-scaled poses from the visual sensor with the noisy metric-scaled re-integrated IMU readings [17].

On the contrary, in the tightly coupled approach, scene information from the visual sensor is fused with the IMU measurements (linear accelerations and angular velocities) using a fusion filter or an optimization model that estimates the metric-scaled pose, visual

odometry scale factor, IMU biases, and visual drift between the IMU-camera inertial frames. One of the prominent advantages of a tightly-coupled fusion scheme is that it can estimate accurate scene information to reconstruct a precise scene map, along with providing the SLAM system with high confidence in loop closure during re-localization situations.

2.2. Fusion Strategies

The two sensor fusion strategies (loosely and tightly coupled) have two main execution techniques: filter-based and optimization-based execution. Some filter-based state-of-the-art approaches are deterministic, such as MSCKF [18], S-MSCKF [11], S-UKF-LG/S-IEKF [19], and ROVIO [20]. At the same time, alternative strategies can be based on nondeterministic filters such as particle filters [21].

Optimization-based methods such as VINS-Mono [22], OKVIS [23], ORB-SLAM [24], and BASALT [25], can be deterministic or non-deterministic based on the optimization strategy and the convergence constraints. The estimation and robustness of visual localization frameworks have advanced significantly in recent decades, and this development may be furthered by tightly integrating visual and inertial data. Most methods integrate data utilizing optimization methods or filtering-based procedures.

Filtering approaches are ideally suited to real-time applications [26,27], which is the main emphasis of this study. In contrast, optimization-based methods are more precise but often have a more extensive processing complexity. The observability-constrained technique addresses the consistency issue, a shortcoming of traditional VIO filter-based algorithms [28]. The EKF/MSCKF and its cutting-edge variations are among the most widely used solutions because they effectively balance accuracy and computational complexity.

A recent study [29] shows that if the air mass's random character is considered, the EKF system states of an MAV are observable. The drag and lift forces on the MAV will directly impact the projected pose and velocity due to the nature of air mass randomization. To make an online update for the uncertainties brought on by these random effects on the precise position of the sensors' reference frames, we contribute with a visual drift augmentation technique to our EKF measurement model. The EKF's ability to tolerate significant disturbances in the MAV's velocity state variable and still converge to the undisturbed estimates is what we target.

2.3. Visual Odometry

The main objective of a visual odometry solution is to perform an accurate and precise localization of the robot (ground or aerial vehicle) to estimate its pose during the navigation task. Estimated poses can be on either discrete- or continuous-time manifolds. Cioffi et al. [30] studied the reliability of the estimated poses on both manifolds using IMU/Visual/GPS sensors. They came to an important conclusion: similar results are produced by the two representations when the camera and IMU are time-synchronized.

In [13], the sliding window pose-graph optimization of the most recent robot states uses global position data with poses predicted by a VIO method. Like [15], pose-graph optimization employs an independent VIO technique to generate pose estimations fused with GPS data. In contrast to [13], the pose-graph in [15] includes an extra node representing the local coordinate frame's origin to confine the absolute orientation. However, these methods are loosely connected, meaning that a separate VIO algorithm generates the relative pose estimations. Inspired by [13,15], we present a loosely coupled strategy that considers the correlations between all measures by including them in a hybrid optimization and filtering problem.

It is demonstrated in [23] that considering all measurement correlations is essential for high-precision estimations in the visual-inertial situation. A tightly coupled sliding window optimization for visual and inertial data with loosely connected GPS refinement is presented in [14]. The GPS readings are given the same timestamp as the temporally nearest image to be included in the sliding window because it is believed that they would only be accessible at low rates. As opposed to [14], we efficiently compute the global

positional factors by closely coupling the global position measurements using the Runge–Kutta 4th-order gyroscope preintegration scheme [31]. This enables the sliding window to incorporate numerous global parameters and each keyframe with barely any additional processing load.

2.4. Methodology Background

We highlight the methodology that inspires our study in blue-dashed rectangles in Figure 2. Where the loosely coupled fusion strategy [32] is adopted to keep constant computational complexity for real-time performance, along with adding a reset mode for the framework, as discussed in [33] as well as an online IMU-camera extrinsic calibration paradigm [4]. Integrating the IMU/GPS readings with the global shutter visual sensor monocular frames raises our localization solution's accuracy level, leveraging the MAV's inertial and global localization information.

Pushing the limits of the extended Kalman filter to raise the robustness of our localization solution towards a resilient system, we leverage the high accuracy of the optimization to initialize the filter pose states using a novel instant approach utilizing the low-rate noisy GPS readings when available. Sensor fusion on continuous-time (CT) manifolds, such as B-splines [34], suffers from a high execution complexity, especially with the time derivatives of high-order manifolds for integrating the IMU measurements in the estimation process. Hence, in our novel method, we avoid this dilemma with a simple spline-fitting approach for the GPS readings during the data pre-processing stage.

3. System Architecture

Our core sensor setup consists of an inertial navigation sensor (IMU), a global positioning sensor (GPS), and a monocular camera, as illustrated in Figure 3. The pipeline starts with the data acquisition and pre-processing for the initialization process, as discussed in Section 3.1. The initialization is an optimization-based phase (see Algorithm 1) with a considerably low complexity and processing time whose output is an instant metric-scaled pose estimated from the camera, GPS, and gyroscope readings. Then, an ES-EKF (see Algorithm 2) whose dynamic model is given in Section 3.2, is applied to estimate all the system states, including the MAV's trajectory, velocity, and a scale factor to recover the initially estimated trajectory in the case of GPS readings loss. Finally, we present the measurement model in Section 3.3 with a novel false pose augmentation paradigm to ensure the observability of all the filter states, as analyzed in Appendix A.

Algorithm 1 Bootstrapping: Pose Graph Optimization and Range Factors

Input: RGB frames (c), camera matrix ($\mathcal{K}_c$), GPS readings (DT-GPS), IMU readings ($\mathcal{I}$)
Output: Metric-scaled trajectory ($\mathcal{T}_{vc}[p_v^c,\ q_v^c] \in SE(3)$)

1: $\mathcal{T}_{vc}^0 \Leftarrow$ KLT-VO (c, $\mathcal{K}_c$) ▷ Arbitrary-scaled pose
2: $p(u) \Leftarrow$ spline_fit (DT-GPS) ▷ CT-GPS by Equation (4)
3: $[\phi, \theta, \psi] \Leftarrow$ RK4 ($\mathcal{I}_{gyro}(\omega)$) ▷ Initial orientations
4: **while** *not converged* **do** ▷ Initial trajectory optimization
5: $\quad \mathcal{T}_{vc} \Leftarrow$ optimize ($\mathcal{T}_{vc}^0$, $p(u)$, $[\phi, \theta, \psi]$) ▷ Equation (6)
6: **end while**

The state representation is a 31-element state vector $\mathcal{X}$:

$$\mathcal{X} = \begin{bmatrix} {p_w^i}^\top & {v_w^i}^\top & {q_w^i}^\top & {b_\omega}^\top & {b_a}^\top & \lambda & {p_i^c}^\top & {q_i^c}^\top & {p_v^w}^\top & {q_v^w}^\top \end{bmatrix}^\top, \tag{1}$$

where p_w^i is the position of the IMU in the world frame (world frame is a gravity-aligned frame.) (w), its velocity v_w^i, and its attitude rotation quaternion q_w^i describing a rotation from the IMU frame (i) into the world frame (w). b_ω and b_a are the gyro and acceleration biases along with the visual odometry scale factor λ. $R_{(q)}$ is the quaternion q rotational

matrix, g is the gravity vector aligned with the world frame (w), and $\Omega(\omega)$ is the quaternion-multiplication matrix of ω.

The IMU/camera calibration states are the rotation from the camera frame into the IMU frame q_i^c, and the position of the camera center with regard to the IMU frame p_i^c.

Finally, the visual attitude drifts between the black-boxed visual frame (vision frame is the frame to which the camera pose is estimated in the black-box vision framework) (v) and the world inertial frame (w) are reflected in q_v^w and the translational ones in p_v^w. We assume that all the visual drifts are spatial without any temporal drifts, i.e., the IMU and the camera have synchronized timestamps.

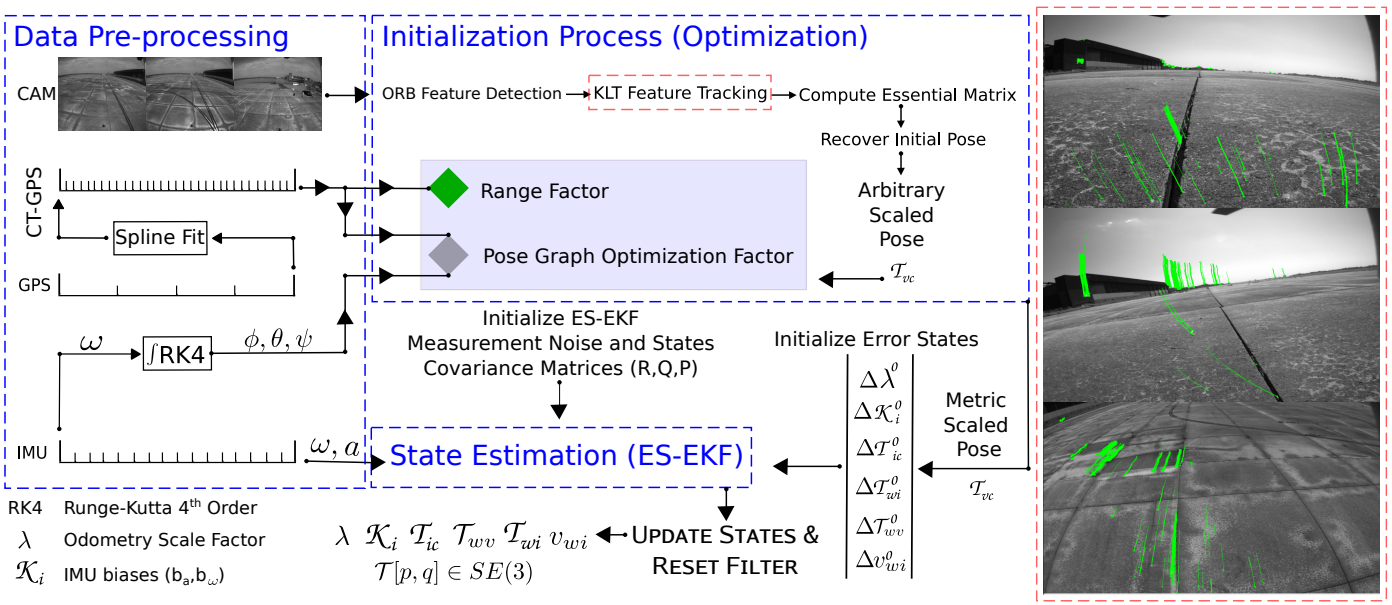

Figure 3. Overview of our proposed entire system architecture.

Algorithm 2 End-to-End State Estimation Scheme

Input: IMU readings, initial optimized trajectory $\mathcal{T}_{vc}$
Output: FilterStates $\mathcal{X} = \{\lambda, \mathcal{K}_i[b_a,\ b_\omega], \mathcal{T}_{ic}, \mathcal{T}_{wv}, \mathcal{T}_{wi}, v_{wi}\},\ \forall T[p,q] \in SE(3)$

$P, Q_c, \mathcal{R}$_initialization, FilterStates_initialization
ErrorStates_initialization=0
while imuRead **do**
    Read LastStep (k) P, FilterStates, ErrorStates
    Read LastStep (k) IMU (Accel, Gyro) values
    Read current (k+1) IMU (Accel, Gyro) values
    Step 1: Propagate IMU states ▷ Equation (12)
    Step 2: Calculate F_d and Q_d ▷ Equations (14) and (16)
    Step 3: Compute P state covariance matrix ▷ Equation (18)
    if camRead **then**
        Read current (k+1) CAM $\mathcal{T}_{vc}$ values ▷ Metric-scaled pose
        Step 4: Estimate false pose ▷ Equation (19)
        Step 5: Calculate $\tilde{z}$, H ▷ Equation (20)
        Step 6: Calculate S, K, ErrorStates $\hat{\tilde{x}}$, P ▷ Equations (26) and (27)
        Step 7: Update: FilterStates += ErrorStates
        Step 8: RESET $\hat{\tilde{x}} = 0$, P ▷ Equation (29)
    end if
end while

The corresponding 28-elements error state vector is defined by:

$$\tilde{x} = \left[\Delta p_w^{i\top}\ \Delta v_w^{i\top}\ \delta\theta_w^{i\top}\ \Delta b_\omega^\top\ \Delta b_a^\top\ \Delta\lambda\ \Delta p_i^{c\top}\ \delta\theta_i^{c\top}\ \Delta p_v^{w\top}\ \delta\theta_v^{w\top}\right]^\top, \tag{2}$$

as the difference of an estimate $\hat{x}$ to its quantity x, i.e., $\tilde{x} = x - \hat{x}$. We apply this to all state variables except the error quaternions, which are defined by:

$$\delta q_y^x = q_y^x \otimes \hat{q_y^x} \approx \left[\ \tfrac{1}{2}\delta\theta_y^x \quad 1\ \right]^\top. \tag{3}$$

This error quaternion representation increases the numerical stability of the estimation process and handles the quaternion in its minimal representation [35].

3.1. State Estimator Initialization

An incremental structure from motion (SfM) algorithm [36] is applied to the acquired image frames, whose goal is to retrieve the camera poses and the 3D structure of the scene, based on the five-point algorithm proposed in [37]. ORB features are detected, and the highest quality points are tracked between 10 consecutive frames using the KLT method [38].

To solve the arbitrary-scale problem of the camera trajectory only, we apply an on-manifold cumulative B-spline (https://github.com/AbanobSoliman/B-splines (accessed on 1 October 2022)) interpolation [34] to synthesize a very smooth continuous-time (CT) trajectory in $\mathbb{R}^3$ from the low-rate noisy GPS readings.

The matrix form for the cumulative B-spline manifold of order $k = n + 1$, where n is the spline degree, is modeled at $t \in [t_i, t_{i+k-1}]$ as:

$$p(u) = p_i + \sum_{j=1}^{k-1} \widetilde{B}_j^{(k)} . \bar{u}_j^{(k)} . d_j^i, \tag{4}$$

where $p(u) \in \mathbb{R}^3$ is the continuous-time B-spline increment that interpolates k GPS measurements on the normalized unit of time $u(t) := (t - t_i)/\Delta t_s - P_n$ with $1/\Delta t_s$ denoting the spline generation frequency and P_n being the pose number that contributes to the current spline segment $P_n \in [0, \cdots, k-1]$. p_i is the initial discrete-time (DT) GPS location measurement at time t_i. The term $d_j^i = p_{i+j} - p_{i+j-1}$ is the difference vector between two consecutive DT-GPS readings. The matrix $\widetilde{B}_j^{(k)}$ is the cumulative basis blending and $\bar{u}_j^{(k)}$ is the normalized time vector, both of which are defined as:

$$\begin{aligned}
\widetilde{B}_j^{(k)} &= \widetilde{b}_{j,n}^{(k)} = \textstyle\sum_{s=j}^{k-1} b_{s,n}^{(k)}, \\
b_{s,n}^{(k)} &= \tfrac{C_{k-1}^n}{(k-1)!} \textstyle\sum_{l=s}^{k-1} (-1)^{l-s} C_k^{l-s} (k-1-l)^{k-1-n}, \\
\bar{u}_j^{(k)} &= [u^0, \cdots, u^{k-1}, u^k]^\top,\ u \in [0, \cdots, 1].
\end{aligned} \tag{5}$$

Our GPS-IMU aided initialization system comprises two optimization factors: the first is a pose graph optimization (PGO) factor r^p that optimizes the 6-DoF of every pose, whereas the second is a range factor r^s that constraints the translation limits between every two KLT-VO poses. Hence, the metric scale of the visual odometry pose is recovered using the gyroscope and GPS readings, leveraging the high accuracy of the optimization process. An illustrative scheme for the initialization process factor graph is shown in Figure 4.

Level 1's objective function $L^{p,s}$ is modeled as:

$$L^{p,s} = \underset{\mathcal{T}_{wi}}{\arg\min} \left[\sum_{(i,j)}^{N} \left(||r^p(i,j)||^2_{\Sigma^p_{i,j}} + ||r^s(i,j)||^2_{\Sigma^s_{i,j}} \right) \right]. \tag{6}$$

$\Sigma^{p}_{i,j}, \Sigma^{s}_{i,j}$ are the information matrices associated with the GPS readings covariance, reflecting the PGO and Range factors noises on the global metric scale estimation process between two RGB-D aligned frames.

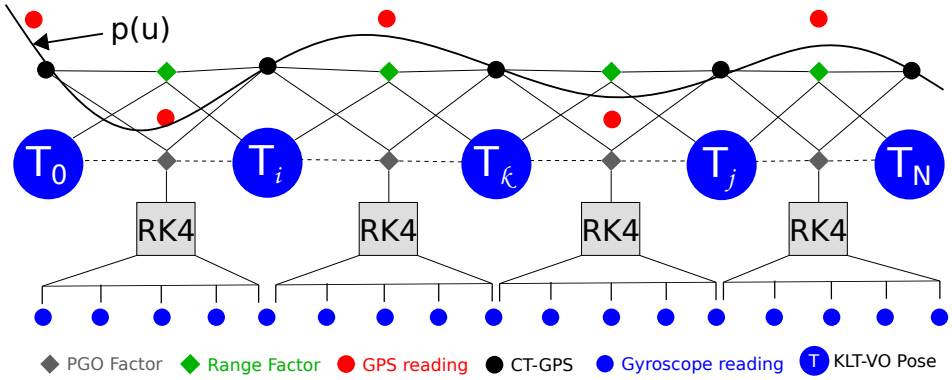

Figure 4. Initialization factor graph. $p(u)$ is the CT-GPS trajectory generated at high frequency. RK4 is the Runge–Kutta 4th order gyroscope integration scheme. Dotted lines denote the error term $(\hat{\boldsymbol{T}}_i^{-1}\hat{\boldsymbol{T}}_j)$ in Equation (7) between any two KLT-VO poses.

Pose Graph Optimization (PGO) factor. The PGO is a 6-DoF factor that controls the relative pose error between two consecutive edges i, j and is formulated as:

$$r^{p} = \left|\left|\left(\hat{\boldsymbol{T}}_i^{-1}\hat{\boldsymbol{T}}_j\right) \ominus \boldsymbol{\Delta T}_{ij}^{\omega,GPS}\right|\right|_2 \tag{7}$$

where $||.||_2$ is the L2-norm, $\hat{\boldsymbol{T}}_{i,j} \in SE(3)$ is the $\mathcal{T}_{wi}^{0}$ estimated from the front-end pipeline at frames i, j. The operator $\ominus$ is the $SE(3)$ logarithmic map as defined in [39]. The error transformation $\Delta T_{ij}^{\omega,GPS}[\delta R_{ij}^{\omega}, \delta p_{ij}^{GPS}] \in \mathfrak{se}(3)$, where $\delta p_{ij}^{GPS} = p_j - p_i$ is the CT-GPS measurement increment and $\delta R_{ij}^{\omega} = [\delta\phi, \delta\theta, \delta\psi]^{\top} \in \mathfrak{so}(3)$ is the gyroscope integrated increment $\delta R_{ij}^{\omega} = \int_{k=i}^{j}(\omega_k).dk$ using the Runge–Kutta 4th order (RK4) integration method [31] between the keyframes i and j.

Range factor. The range factor limits the front-end visual drift and keeps the global metric scale under control within a sensible range defined by the GPS signal and is formulated as:

$$r^{s} = \left|\left|\,||\hat{\boldsymbol{t}}_j - \hat{\boldsymbol{t}}_i||_2 - ||p_j^{GPS} - p_i^{GPS}||_2\right|\right|_2 \tag{8}$$

where $\hat{\boldsymbol{t}}_{i,j}, p_{i,j}^{GPS} \in \mathbb{R}^3$ are the translation vectors of two consecutive front-end (KLT-VO) poses and CT-GPS signals, respectively.

3.2. Dynamic Model

The core state estimation is performed by fusing the RGB camera frames and the IMU reading using an error states extended Kalman filter (ES-EKF). Figure 5 illustrates the inter-sensor extrinsic relation between the IMU/GPS sensors and a monocular camera.

To use the linear states estimator, we assume that the IMU measurements contain a particular bias $b_a \in \mathcal{N}(0, \sigma_{ba})$, $b_\omega \in \mathcal{N}(0, \sigma_{b\omega})$ and a white Gaussian noise $n_a \in \mathcal{N}(0, \sigma_a)$, $n_\omega \in \mathcal{N}(0, \sigma_\omega)$.

Thus, the real angular velocities ω and accelerations a in the IMU body frame (i) can be written as:

$$\omega = \omega_m - b_\omega - n_\omega \quad and \quad a = a_m - b_a - n_a, \tag{9}$$

where the subscript m denotes the measured value. The dynamics of the non-static biases are modeled as a random process:

$$\dot{b_\omega} = n_{b_\omega}, \quad \dot{b_a} = n_{b_a}. \tag{10}$$

The standard deviation $\sigma_{b_\omega}, \sigma_{b_a}, \sigma_w, \sigma_a$ values are generally given by the IMU manufacturer's data in Allan deviation plots. For discrete time steps, it will be applied in the filter. We need to convert these values according to their units:

$$d\sigma^2_{\omega,a} = \frac{\sigma^2_{\omega,a}}{\nabla t}, \quad d\sigma^2_{b_{\omega,a}} = \sigma^2_{b_{\omega,a}} * \nabla t. \tag{11}$$

The following differential equations govern IMU state propagation:

$$\begin{aligned}
\dot{p^i_w} &= v^i_w, \\
\dot{v^i_w} &= R^\top_{(q^i_w)}(a_m - b_a - n_a) - g, \\
\dot{q^i_w} &= \tfrac{1}{2}\Omega(\omega_m - b_\omega - n_\omega)q^i_w, \\
\dot{b_\omega} &= n_{b_\omega},\ \dot{b_a} = n_{b_a},\ \dot{\lambda} = 0, \\
\dot{p^c_i} &= 0,\ \dot{q^c_i} = 0,\ \dot{p^w_v} = 0,\ \dot{q^w_v} = 0,
\end{aligned} \tag{12}$$

For the quaternion integration inside the ES-EKF, we use the first-order integrator defined in [35] as:

$$\begin{aligned}
\bar{w} &= \tfrac{\omega_{k+1}+\omega_k}{2}, \quad \kappa = \tfrac{1}{2}.\Omega(\bar{\omega}).\Delta t, \\
\hat{q^i_{wk+1}} &= [e^\kappa + \tfrac{\Delta t^2}{48}(\Omega(\omega_{k+1}).\Omega(\omega_k) - \Omega(\omega_k).\Omega(\omega_{k+1}))].\hat{q^i_{wk}}.
\end{aligned} \tag{13}$$

where the hat term ˆ means the estimated value. The exponential term e^κ is expanded by the Maclaurin series.

The states transition matrix F_d is modeled as:

$$F_d = \begin{bmatrix}
I_{d_3} & \Delta t & A & B & -R^\top_{(\hat{q^i_w})}\frac{\Delta t^2}{2} & 0_{3\times 13} \\
0_3 & I_{d_3} & C & D & -R^\top_{(\hat{q^i_w})}\Delta t & 0_{3\times 13} \\
0_3 & 0_3 & E & F & 0_3 & 0_{3\times 13} \\
0_3 & 0_3 & 0_3 & I_{d_3} & 0_3 & 0_{3\times 13} \\
0_3 & 0_3 & 0_3 & 0_3 & I_{d_3} & 0_{3\times 13} \\
0_{13\times 3} & 0_{13\times 3} & 0_{13\times 3} & 0_{13\times 3} & 0_{13\times 3} & I_{d_{13}}
\end{bmatrix}. \tag{14}$$

Then, we apply the small-angle approximation for which $|\omega| \to 0$ apply the de l'Hopital rule and obtain a compact solution for the six matrix blocks A, B, C, D, E, F [35]:

$$\begin{aligned}
A &= -R^\top_{(\hat{q^i_w})}\lfloor \hat{a} \rfloor_\times (\tfrac{\Delta t^2}{2!} - \tfrac{\Delta t^3}{3!}\lfloor \hat{\omega} \rfloor_\times + \tfrac{\Delta t^4}{4!}\lfloor \hat{\omega} \rfloor^2_\times), \\
B &= -R^\top_{(\hat{q^i_w})}\lfloor \hat{a} \rfloor_\times (\tfrac{-\Delta t^3}{3!} + \tfrac{\Delta t^4}{4!}\lfloor \hat{\omega} \rfloor_\times - \tfrac{\Delta t^5}{5!}\lfloor \hat{\omega} \rfloor^2_\times), \\
C &= -R^\top_{(\hat{q^i_w})}\lfloor \hat{a} \rfloor_\times (\Delta t - \tfrac{\Delta t^2}{2!}\lfloor \hat{\omega} \rfloor_\times + \tfrac{\Delta t^3}{3!}\lfloor \hat{\omega} \rfloor^2_\times), \\
D &= -A, \\
E &= I_{d_3} - \Delta t\lfloor \hat{\omega} \rfloor_\times + \tfrac{\Delta t^2}{2!}\lfloor \hat{\omega} \rfloor^2_\times, \\
F &= -\Delta t + \tfrac{\Delta t^2}{2!}\lfloor \hat{\omega} \rfloor_\times - \tfrac{\Delta t^3}{3!}\lfloor \hat{\omega} \rfloor^2_\times,
\end{aligned} \tag{15}$$

with $\hat{\omega} = \omega_m - \hat{b_\omega}$, $\hat{a} = a_m - \hat{b_a}$ and $\lfloor \hat{\omega} \rfloor_\times$, $\lfloor \hat{a} \rfloor_\times$ the skew-symmetric matrices for IMU readings.

We can now derive the discrete-time input noise covariance matrix Q_d as:

$$Q_d = \int_{\Delta t} F_d(\tau) G_c Q_c G_c^\top F_d(\tau)^\top d\tau, \tag{16}$$

where Q_c is the CT process noise covariance, and G_c is calculated in the form:

$$G_c = \begin{bmatrix} 0_3 & 0_3 & 0_3 & 0_3 \\ -R^\top_{(\hat{q}^i_w)} & 0_3 & 0_3 & 0_3 \\ 0_3 & 0_3 & I_{d_3} & 0_3 \\ 0_3 & 0_3 & 0_3 & I_{d_3} \\ 0_3 & -I_{d_3} & 0_3 & 0_3 \\ 0_{13\times 3} & 0_{13\times 3} & 0_{13\times 3} & 0_{13\times 3} \end{bmatrix}. \tag{17}$$

The closed-form solution of the complete derivation of the Q_d covariance matrix is given in detail in Appendix B.

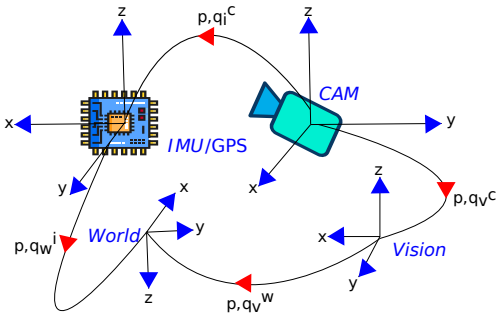

Figure 5. The frames of reference annotations.

Finally, the propagated state covariance matrix computation is defined as:

$$P_{k+1|k} = F_d P_{k|k} F_d^\top + Q_d. \tag{18}$$

3.3. Measurement Model

The main contribution of our measurement model for an observable ES-EKF is the false relative pose augmentation methodology of the visual drift quaternion state at the previous time step (k) updated with the current camera measurement at a time (k+1) and modeled as:

$$q^w_v(k) = \hat{q}^i_w(k)^{-1} \otimes \hat{q}^c_i(k)^{-1} \otimes q^c_v(k+1). \tag{19}$$

The camera position measurement model yields the position of the camera with respect to the vision frame p^c_v. The error in measurement modeled as $\tilde{z}_p$ and linearized as $\tilde{z}_{pL}$:

$$\tilde{z}_p = z_p - \hat{z}_p = p^c_v - R^\top_{(\hat{q}^w_v)}(\hat{p}^i_w + R^\top_{(\hat{q}^i_w)}\hat{p}^c_i)\hat{\lambda} \doteq \tilde{z}_{pL} = H_p \tilde{x}, \tag{20}$$

with

$$H_p^\top = \begin{bmatrix} R^\top_{(\hat{q}^w_v)}\hat{\lambda} \\ 0_{3\times 3} \\ -R^\top_{(\hat{q}^w_v)}R^\top_{(\hat{q}^i_w)}\lfloor \hat{p}^c_i \rfloor_\times \hat{\lambda} \\ 0_{6x3} \\ R^\top_{(\hat{q}^w_v)}R^\top_{(\hat{q}^i_w)}\hat{p}^c_i + R^\top_{(\hat{q}^w_v)}\hat{p}^i_w \\ R^\top_{(\hat{q}^w_v)}R^\top_{(\hat{q}^i_w)}\hat{\lambda} \\ 0_{6x3} \\ -R^\top_{(\hat{q}^w_v)}\left\lfloor (\hat{p}^i_w + R^\top_{(\hat{q}^i_w)}\hat{p}^c_i)\hat{\lambda} \right\rfloor_\times \end{bmatrix}, \tag{21}$$

using the definition of the error-quaternion

$$\begin{aligned} q^i_w &= \delta q^i_w \otimes \hat{q}^i_w, \\ R_{(q^i_w)} &\approx (I_{d_3} - \lfloor \delta\theta^i_w \rfloor_\times).R_{(\hat{q}^i_w)}. \end{aligned} \tag{22}$$

The vision algorithm yields the rotation from the camera frame into the vision frame q_v^c. We can model the error measurement as

$$\tilde{z}_q = z_q - \hat{z}_q = q_i^c \otimes q_w^i \otimes q_v^w \otimes (q_i^c \otimes \hat{q}_w^i \otimes \hat{q}_v^w)^{-1}. \tag{23}$$

Finally, the measurements Jacobian H in $\tilde{z} = H.\tilde{x}$ is calculated based on the method in [33] and can be stacked together in the form

$$\begin{bmatrix} \tilde{z}_p \\ \tilde{z}_q \end{bmatrix} = \begin{bmatrix} & & & H_p & & & \\ 0_{3x6} & \tilde{H}_q^{wi} & 0_{3x10} & \tilde{H}_q^{ic} & 0_{3\times3} & \tilde{H}_q^{vw} & \end{bmatrix} \tilde{x}. \tag{24}$$

with the Jacobian matrices $\tilde{H}_q^{xy}$, known as the right Jacobian of $SO(3)$, and are defined as:

$$\begin{aligned} \tilde{H}_q^{xy} &= J_r(\theta_x^y) = \lim_{\delta\theta\to 0} \frac{Log(Exp(\theta)^* \otimes Exp(\theta+\delta\theta))}{\delta\theta}, \\ J_r(\theta_x^y) &= I_{d_3} - \left(\frac{1-cos\|\delta\theta\|}{\|\delta\theta\|^2}\right).\left\lfloor \delta\theta_x^y \right\rfloor_\times + \left(\frac{\|\delta\theta\|-sin\|\delta\theta\|}{\|\delta\theta\|^3}\right).\left\lfloor \delta\theta_x^y \right\rfloor_\times^2. \end{aligned} \tag{25}$$

3.4. States Update

To update the framework for the current time step (k+1), we compute the innovation term S, Kalman gain K, and the states correction vector $\hat{\tilde{x}}$ defined as:

$$S = HPH^\top + \mathcal{R}, \quad K = PH^\top S^{-1}, \quad \hat{\tilde{x}} = K\tilde{z} \ . \tag{26}$$

The error state covariance is updated as follows:

$$P_{k+1|k+1} = (I_{d_{28}} - KH)P_{k+1|k}(I_{d_{28}} - KH)^\top + K\mathcal{R}K^\top, \tag{27}$$

where $\mathcal{R}_{[6x6]} = diag(\mathcal{R}_{position}, \mathcal{R}_{orientation})$ is the measurement noise covariance matrix.

The error quaternion is calculated by (3) to ensure its unit length, and then update the states vector: $\mathcal{X}_{k+1} = \mathcal{X}_k + \hat{\tilde{x}}$.

For the quaternions state update:

$$\hat{q}_{k+1} = \frac{\begin{bmatrix} 1 & \frac{1}{2}\delta\theta_{k+1}^1 & \frac{1}{2}\delta\theta_{k+1}^2 & \frac{1}{2}\delta\theta_{k+1}^3 \end{bmatrix} \otimes \hat{q}_k}{\left\| \begin{bmatrix} 1 & \frac{1}{2}\delta\theta_{k+1}^1 & \frac{1}{2}\delta\theta_{k+1}^2 & \frac{1}{2}\delta\theta_{k+1}^3 \end{bmatrix} \otimes \hat{q}_k \right\|}, \tag{28}$$

where $\delta\theta_{k+1}^i$ is the ith error state of this quaternion.

3.5. Reset Mode

The ES-EKF reset mode is performed by setting $\hat{\tilde{x}} \leftarrow 0$ and $P \leftarrow G.P.G^\top$, where G is the Jacobian matrix defined by

$$\begin{aligned} G &= diag(I_{d_6}, J_{r_{wi}}, I_{d_{10}}, J_{r_{ic}}, I_{d_3}, J_{r_{vw}}), \\ J_{r_{xy}} &= \frac{\partial\delta\theta_x^{y^+}}{\partial\delta\theta_x^y} = I_{d_3} - \frac{1}{2}\left\lfloor \delta\hat{\theta}_x^y \right\rfloor_\times. \end{aligned} \tag{29}$$

4. Experiments

4.1. Setup

An extensive quantitative and qualitative evaluation is carried out to validate all the state estimation process aspects. This thorough performance analysis is run on the EuRoC benchmark [12] for an indoor system global positioning evaluation in low-speed flights and on the Fast Flight dataset [11] for outdoor experimentation at relatively high-speed flights. For a fair comparison, all the pipeline processing stages in both Algorithms 1 and 2 are performed on a 16 GB RAM laptop computer running 64-bit Ubuntu 20.04.3 LTS with AMD(R) Ryzen 7 4800 h × 16 cores 2.9 GHz processor and a Radeon RTX NV166 Renoir

graphics card. In Table 1, we represent the quantitative insights of our experiment settings regarding the benchmarks statistical data and the sensors parameters in-detail.

Table 1. Insights into our experiments' statistical information and sensor settings.

	Parameter	EuRoC Benchmark [12]				Fast Flight Dataset [11]			
Stats	Total processed sequences	6 (Vicon room)				4 (airport runway)			
	Total sequences duration	11.6111 min				8.8867 min			
	Total sequences length	411.5425 m				2539.0599 [1] m			
	Maximum speed	2.3 (m/s)				17.5 (m/s)			
Camera	Total processed frames	13,736				21,312			
	Frame resolution	752 × 480 pixels				960 × 800 pixels			
	Intrinsics (f_x, f_y, c_x, c_y)	458.65	457.30	367.22	248.38	606.58	606.73	474.93	402.28
	Distortion (k_1, k_2, p_1, p_2)	−0.2834	0.0739	0.0001	0.000018	−0.0147	−0.0058	0.0072	−0.0046
	Camera-IMU p_i^c(x,y,z,1) (m)	−0.0216	−0.0647	0.0098	1.0000	0.1058	−0.0177	−0.0089	1.0000
	Camera-IMU q_i^c(x,y,z,w) [-]	−0.0077	0.0105	0.7018	0.7123	−1.0000	0.0042	−0.0039	0.0015
	Frame rate	20 (Hz)				40 (Hz)			
IMU	Gyroscope noise density (σ_{n_ω})	1.6968×10^{-4} [rad/s/$\sqrt{\text{Hz}}$]				6.1087×10^{-5} [rad/s/$\sqrt{\text{Hz}}$]			
	Gyroscope random walk $(\sigma_{n_{b_\omega}})$	1.9393×10^{-5} [rad/s^2/$\sqrt{\text{Hz}}$]				9.1548×10^{-5} [rad/s^2/$\sqrt{\text{Hz}}$]			
	Accelerometer noise density (σ_{n_a})	2.0000×10^{-3} [m/s^2/$\sqrt{\text{Hz}}$]				1.3734×10^{-3} [m/s^2/$\sqrt{\text{Hz}}$]			
	Accelerometer random walk $(\sigma_{n_{b_a}})$	3.0000×10^{-3} [m/s^3/$\sqrt{\text{Hz}}$]				2.7468×10^{-3} [m/s^3/$\sqrt{\text{Hz}}$]			
	Data rate $(1/\Delta t)$	200 (Hz)				200 (Hz)			
GPS	Type/operation	Indoors/Vicon system				Outdoors/satellite Triangulation			
	Readings	X (m), Y (m), Z (m)				Long. (deg), Lat. (deg), Alt. (m)			
	Data rate	1 (Hz) (down-sampled)				5 (Hz)			

[1] Denotes the exact value of the total trajectories lengths for all of the sequences of Fast Flight dataset shown on the x axis of Figure 1 (≈2.5 (km)).

The front-end of the pipeline, including both the data acquisition and pre-processing steps, is developed as a Python API that sends the optimization variables to the factor graph implemented in C++ using the Ceres solver [40] to achieve the lowest possible system latency before the state estimation process. The Sparse Normal Cholesky linear solver by the Ceres solver is employed to solve the least-squares convex optimization problem formulated in Equation (6) along with the Levenberg–Marquardt trust region strategy with the automatic differentiation tool for Jacobian calculations. The sparse Schur linear method is applied to utilize the Schur complement for a more robust and fast optimization process. The pipeline's back-end for the state estimation process is developed entirely in MATLAB (https://github.com/AbanobSoliman/VIO_RGB_IMU (accessed on 30 October 2022)) and all the initialization parameters are given explicitly in Table 2.

Table 2. The ES-EKF initialization parameters for both the EuRoC and Fast Flight sequences.

Parameter Initialization	EuRoC Benchmark [12]	Fast Flight Dataset [11]
28-element error state vector $(\hat{x})$	$0_{28\times1}$	$0_{28\times1}$
31-element state vector [1] $(\mathcal{X})$	$\begin{pmatrix} 0_{3\times1} & 0_{3\times1} & \bar{q}^\top & 0_{3\times1} & 0_{3\times1} & 1 & p_i^{c\top} & q_i^{c\top} & 0_{3\times1} & \bar{q}^\top \end{pmatrix}^\top$	
States propagation covariance (P)	$10^{-7} \times I_{d_{28}}$	$10^{-12} \times I_{d_{28}}$
CT process noise covariance [2] (Q_c)	$diag(d\sigma^2_{n_a}.I_{d_3}, d\sigma^2_{n_{b_a}}.I_{d_3}, d\sigma^2_{n_\omega}.I_{d_3}, d\sigma^2_{n_{b_\omega}}.I_{d_3})$	
Measurement noise covariance $(\mathcal{R})$	$diag(0.01, 0.01, 0.03, 10^{-4}, 10^{-4}, 10^{-4})$	

[1] $\bar{q}$ denotes the unity quaternion [0,0,0,1]. [2] IMU noise density values for each dataset are from Table 1 and discretized using Equation (11).

The performance analysis is performed using the two trajectory evaluation metrics: root mean square error (RMSE) for the Fast Flight dataset compared to the GPS trajectory p_{gps}, and the RMS absolute trajectory error (ATE) for the EuRoC benchmark compared to the ground truth trajectory T_{gt} provided with Vicon room sequences. The positional RMSE

metric for the Fast Flight sequences is chosen because the ground truth GPS trajectories exist with unknown ground truth orientations. However, for EuRoC sequences, we select the RMS ATE metric for two reasons: 1. the Vicon system provides ground truth poses (positions and orientations); and 2. to ensure a fair comparison with the latest state-of-the-art methods based on the same error metric. The two trajectory evaluation metrics are formulated as follows:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left|\left|\hat{p}(i) - p_{gps}(i)\right|\right|^2}, \text{ ATE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left|\left|p(T_{gt}^{-1}(i).T_{rel}.\hat{T}(i))\right|\right|^2}\,[m], \quad (30)$$

where $\hat{p}$ is the estimated translation vector of the $\hat{T} \in SE(3)$ trajectory; $p(.)$ is the translation vector of the $T \in SE(3)$ pose; and T_{rel} is rigid-body transformation corresponding to the least-squares solution that maps the $\hat{T}$ trajectory onto the T_{gt} trajectory calculated by optimization. We set it constant for all sequences that belong to the same benchmark.

4.2. The EuRoC MAV Benchmark

The two main characteristics of the EuRoC MAV sequences are the complex combined 6-DoF motions and the relatively low speeds compared to the Fast Flight sequences. These prominent characteristics allow an accurate evaluation of the ES-EKF marginally stable states, such as the velocity and the visual drift. In Table 3, we report the ATE values as an evaluation parameter for the trajectory estimation accuracy compared to the ground truth. Moreover, Table 3 shows an ablation study that investigates the contribution of the GPS sensor to the overall estimation accuracy, especially for the monocular vision-based optimization methods: ours (PGO) and the recent work of Cioffi et al. [30]. The selection of the six Vicon room sequences from the EuRoC benchmark is because a comparison with an alternative method such as [30] incorporating GPS signals simulated from the Vicon system readings better emphasizes the findings of this ablation study.

Table 3. Ablation study on the contribution of the GPS sensor on the system accuracy. The latest state-of-the-art (monocular/stereo) VI-SLAM systems are compared to our proposed trajectory initialization (PGO factors) and ES-EKF state estimation methods. **Bold** denotes the most accurate.

	Method	EuRoC Benchmark [12] (RMS ATE [m])						Avg.
		V1-01	V1-02	V1-03	V2-01	V2-02	V2-03	
Mono-VI	OKVIS [23]	0.090	0.200	0.240	0.130	0.160	0.290	0.185
	ROVIO [20]	0.100	0.100	0.140	0.120	0.140	0.140	0.123
	VINS-Mono [22]	0.047	0.066	0.180	0.056	0.090	0.244	0.114
	OpenVINS [41]	0.056	0.072	0.069	0.098	0.061	0.286	0.107
	CodeVIO [1] [42]	0.054	0.071	0.068	0.097	0.061	0.275	0.104
	Cioffi et al. [2] [16]	0.034	0.035	0.042	0.026	0.033	0.057	0.038
Stereo-VI	VINS-Fusion [13]	0.076	0.069	0.114	0.066	0.091	0.096	0.085
	BASALT [25]	0.040	0.020	0.030	0.030	0.020	0.050	0.032
	Kimera [43]	0.050	0.110	0.120	0.070	0.100	0.190	0.107
	ORB-SLAM3 [24]	0.038	0.014	0.024	0.032	0.014	0.024	0.024
Mono-(V/I/G) [3]	CT (V+I+G) [30]	0.024	0.014	**0.011**	0.012	0.010	**0.010**	0.014
	CT (V+G) [30]	0.011	0.013	0.012	0.009	**0.008**	0.012	**0.011**
	CT (I+G) [30]	0.062	0.102	0.117	0.112	0.164	0.363	0.153
	DT (V+I+G) [30]	0.016	0.024	0.018	0.009	0.018	0.033	0.020
	DT (V+G) [30]	0.010	0.025	0.024	0.010	0.012	0.029	0.018
	DT (I+G) [30]	0.139	0.137	0.138	0.138	0.138	0.139	0.138
	Ours (PGO)	**0.008**	0.017 [4]	0.023 [4]	**0.008**	0.022	0.025 [4]	0.017
	Ours (ES-EKF)	0.009	**0.012**	**0.011**	0.010	0.011	**0.010**	**0.011**

[1] Denotes the only learning-based baseline in the table and incorporates point clouds using LiDAR. [2] Denotes values from the original work with four GPS readings connected to each optimization state. [3] V,I,G: Vision, IMU, and GPS (generated from the Vicon system readings). [4] Denotes KLT-VO tracks features in 5 consecutive frames instead of 10 due to the rapid movement of the MAV.

A prominent finding of this ablation study is that vision is the most significant type of sensor. In most sequences, the lowest ATE is obtained by fusing the camera trajectory from the vision KLT-based SfM algorithm to a gravity-aligned frame using the noisy simulated GPS data, and adding inertial measurements does not provide a measurable benefit in this case. However, adding the gyroscope measurements to the visual-GPS fusion has led to the least ATE achieved by our PGO model compared to all other discrete-time (DT) methods. Figures 6 and 7 show our trajectory and velocity estimations after incorporating the accelerometer readings in the ES-EKF model, resulting in the lowest achievable errors that can compete with the continuous-time optimization model in [30].

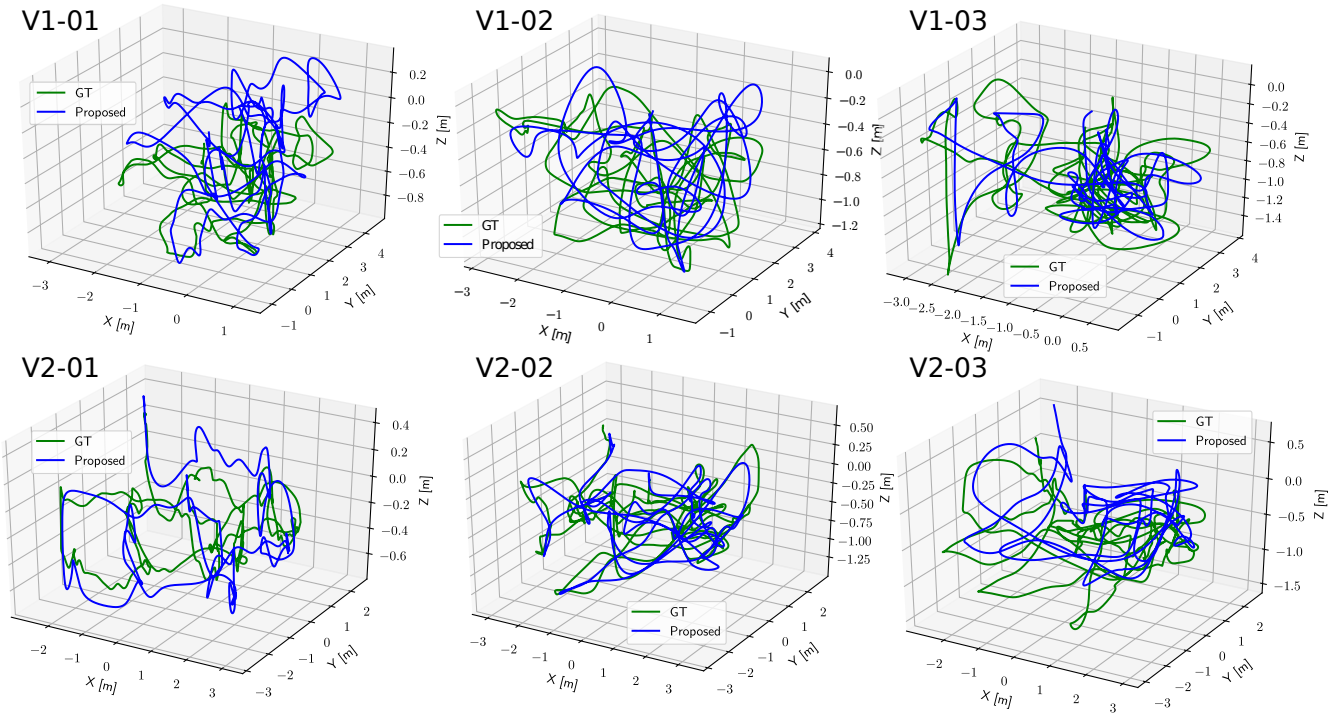

Figure 6. EuRoC 3D trajectory estimation compared to the ground truth.

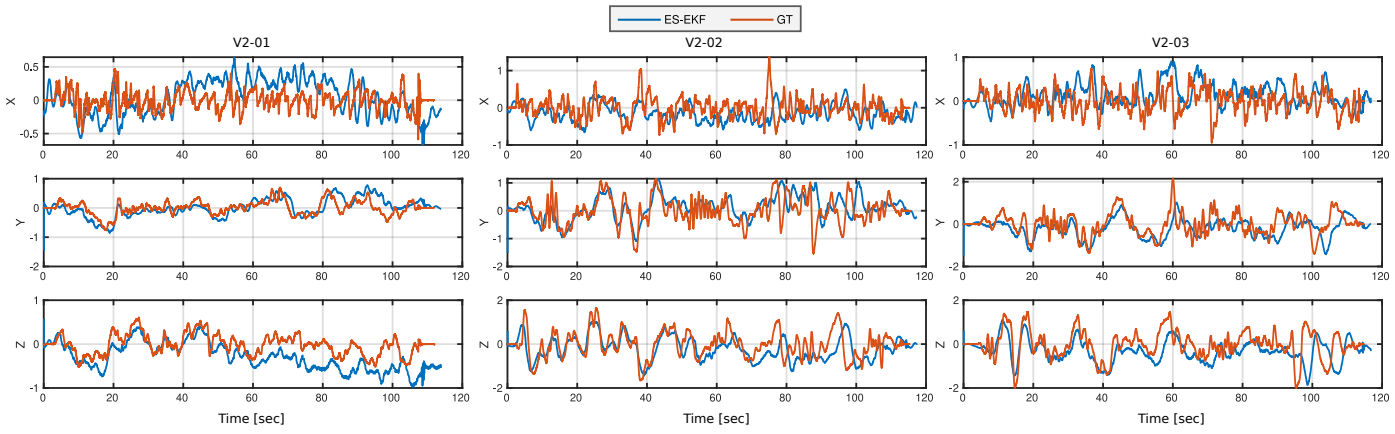

Figure 7. Estimated velocity profile validation with the ground truth. Comparison of sample sequences from EuRoC benchmark.

4.3. The Fast Flight Dataset

The main observation, which is validated upon both the EuRoC and Fast Flight sequences (see Table 4 and Figure 8), is that for velocities less than 5 (m/s), the monocular loosely coupled ES-EKF can achieve considerably lower estimation errors concerning the other filter- or optimization-based methods. For velocities more than 5 (m/s), our proposed

optimization-based initialization scores the lowest RMSE compared to all other methods in comparison in Table 4. On the contrary, the monocular ES-EKF scores the lowest RMSE, especially for velocities more than 10 (m/s), compared to the best-performing Kalman filter stereo model of the S-MSCKF.

Table 4. Ablation study on the effect of the high MAV speed on the accuracy of the filtering approaches compared to optimization approaches. The first sub-section compares monocular (VINS-Mono and Ours) to stereo (OKVIS) optimization-based VI systems. The second sub-section compares stereo filtering-based approaches to our proposed method. **Bold** denotes the most accurate in each sub-section.

Method	Fast Flight [11] (RMSE (m))				Avg.
	gps5	gps10	gps15	gps175	
OKVIS [23]	3.224	4.987	3.985	4.535	4.183
VINS-Mono [22]	5.542	8.753	2.875	3.452	5.156
Ours (PGO)	**0.417**	**0.759**	**0.180**	**0.927**	**0.571**
S-MSCKF [11]	4.985	2.751	**4.752**	**7.852**	**5.085**
S-UKF-LG [19]	4.875	2.589	5.128	7.865	5.114
S-IEKF [19]	4.986	**2.544**	5.124	8.152	5.201
Ours (ES-EKF)	**4.751**	7.924	7.221	9.488	7.346

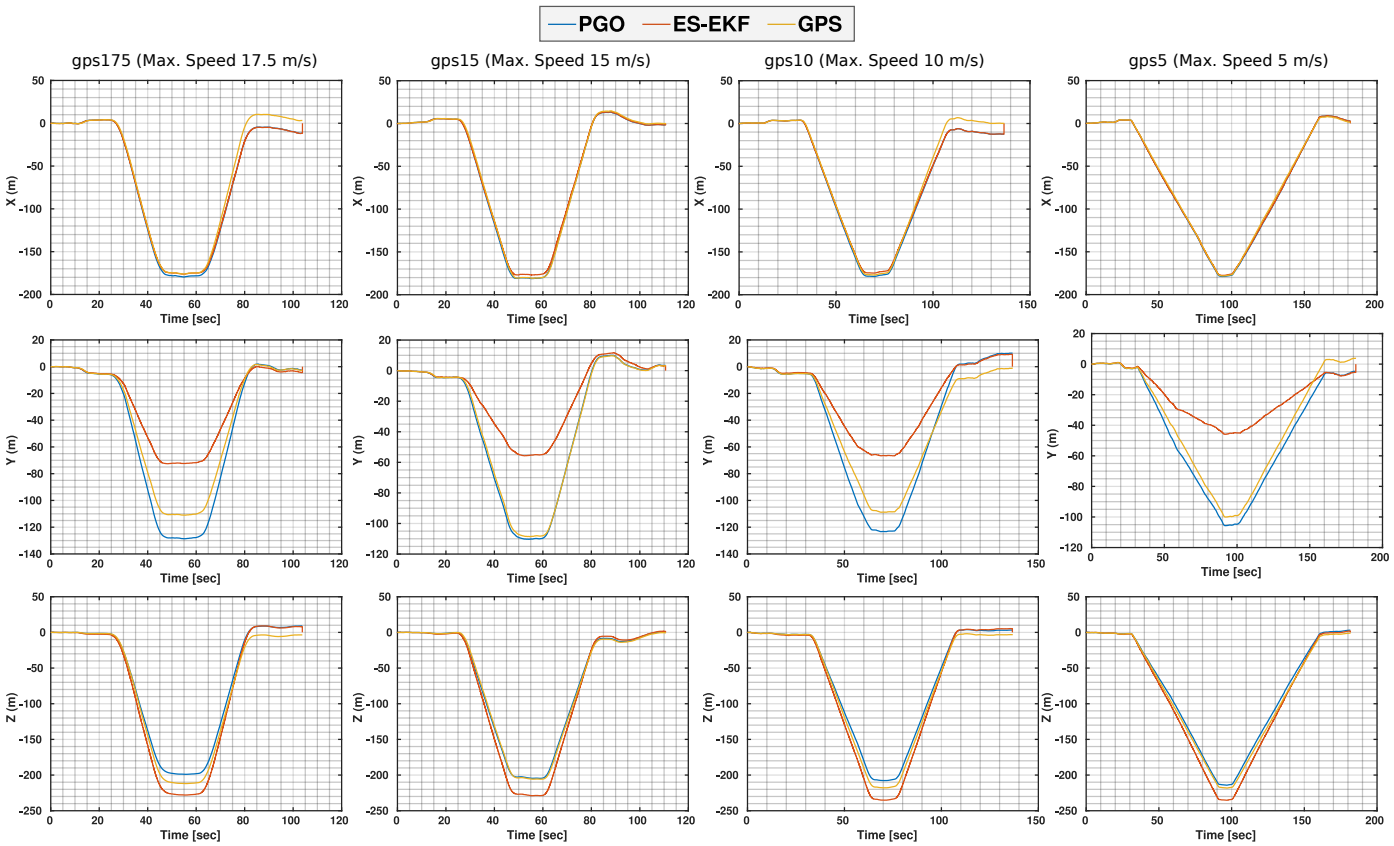

Figure 8. Fast Flight (X (**top**)−Y (**middle**)−Z (**bottom**)) trajectory estimation compared to the GPS readings.

Since the maximum achieved velocity of the EuRoC MAV is nearly 2.3 (m/s), the quantitative results in Table 3 further support this conclusion, where our ES-EKF scores the best performance compared to the other state-of-the-art methods. In-depth reasoning for this degraded performance at high speeds (more than 5 (m/s)) can be clarified based on the hardware characteristics of the MAV sensors' properties, such as the data rate, latency,

and noise effects at high speeds. Our optimization-based (PGO) initialization outperforms all other optimization- or filtering-based methods with high-rate visual-inertial sensors.

An insightful overview of the velocity profiles estimated by our ES-EKF is represented in Figure 9. The main conclusion is that the estimated velocity profile during the planar motion of the MAV in the X–Z plane optimally fits the upper and lower bounds of the top speed for each sequence. Towards an in-depth investigation to understand the high perturbations in the estimated velocity when approaching the maximum limit, we plot the velocity error states in the ES-EKF showing a high error at the instances when approaching top speeds due to the strong vibrations in the MAV structure affecting the IMU readings.

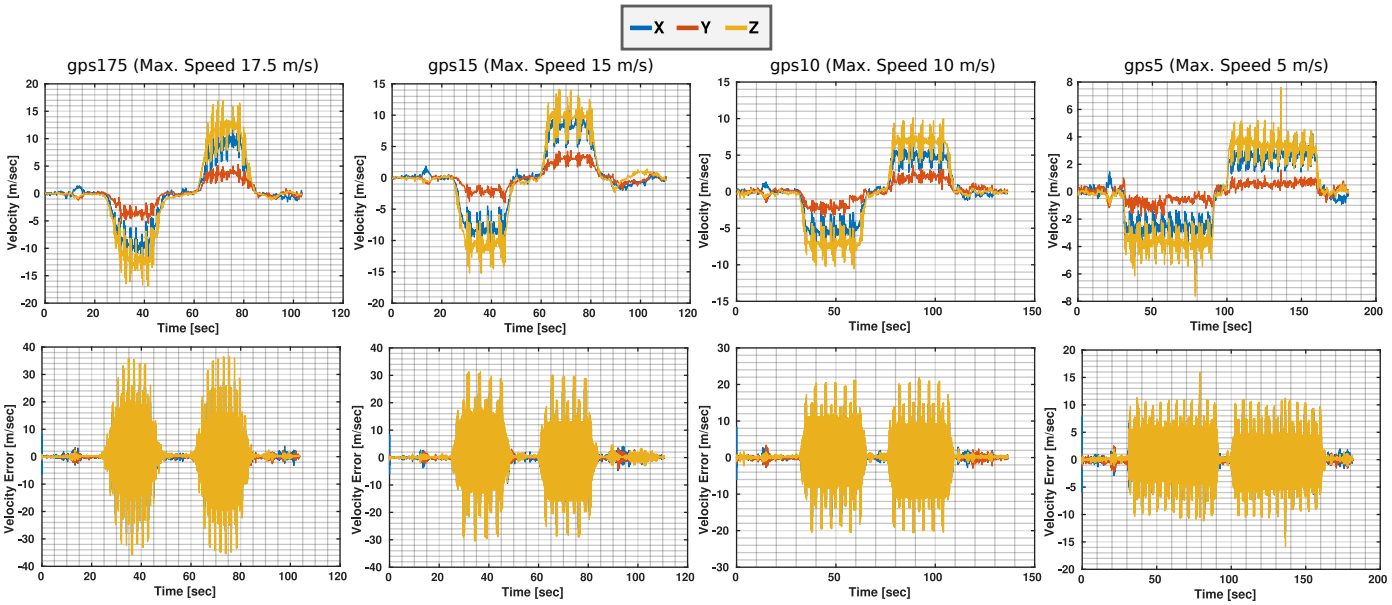

Figure 9. (**Top**): Fast Flight velocity profile validation with the top speed of each sequence. (**Bottom**): velocity error states in the ES−EKF.

The high estimation accuracy of our ES-EKF model compared to GPS readings and the PGO optimization-based initialization process is further verified by the Y axis trajectory estimation in Figure 8. The maximum estimated altitude for all sequences by the ES-EKF is nearly 60 (m), whereas both the GPS readings and the initialization optimizer estimate a maximum altitude of nearly 100 (m). To physically validate which is a more accurate altitude estimation, we took snippets of the scene at a time instance in the exact halfway of all trajectories as shown in Figure 1. We can observe that the MAV is nearly on the same level as the roof of a commercial aircraft hangar, which is in the range of 30 (m) to 66 (m). This observation validates the high estimation accuracy of the altitude using our ES-EKF.

4.4. Real-Time Performance Analysis

The filter-based approaches are more advantageous for real-time onboard applications because they use the CPU more efficiently than the monocular and stereo optimization-based methods. Due to its computationally intensive front-end pipeline for both temporal and stereo matching, OKVIS uses more CPU than VINS-Mono. Additionally, OKVIS's back-end operates at a speed that is much faster than the set 10 (Hz) rate of VINS-Mono. Approximately 90% of the work in our back-end, ES-EKF, is brought on by the front-end, which includes ORB feature detection, KLT-based tracking, and matching. At 200 (Hz), the filter uses approximately 10% of a core. Our suggested technique offers the maximum estimation frequency, which provides the optimal balance between the precision and computing cost.

Figure 10 contrasts how much CPU time various VIO solutions used on the EuRoC benchmark and the Fast Flight dataset. Since V2-03 has considerable scale drift with S-IEKF

and S-UKF-LG techniques and hence has significantly worse accuracy when compared to other methods, the CPU consumption of V2-03 is excluded from the comparison. According to the testing, the ES-EKF achieves the lowest CPU consumption while retaining a similar level of accuracy in comparison with other methods. We notice that the proposed method puts more computing work into the image processing front-end than the tests using the EuRoC dataset. Higher imaging frequency and resolution are one explanation, while Fast Flight results in a shorter feature lifetime, necessitating frequent new feature identification, is another reason.

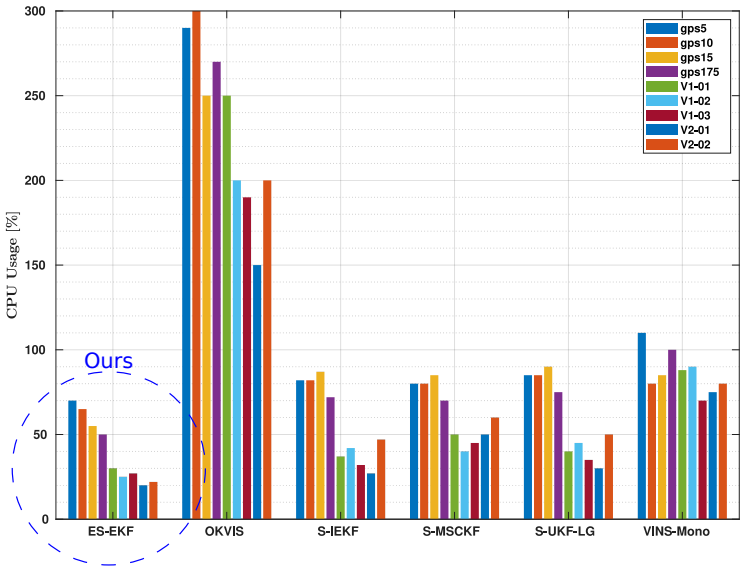

Figure 10. CPU usage as a real-time performance analysis indicator.

5. Conclusions

Our work aimed to provide an accurate and computationally inexpensive localization solution during MAVs' long-term navigation in large-scale environments. We represented a loosely coupled IMU/GPS camera fusion framework with pose failure detection methodology toward this goal. Moreover, we proposed a novel decoupled optimization- and filtering-based sensor fusion technique that achieves a high estimation accuracy and minimum system complexity compared to the other methods in the literature. We used real-world indoor and outdoor settings for the MAV localization studies to validate and test the findings of our proposed method.

The vision-based black-box pose estimation accuracy is first examined in a controlled laboratory Vicon room of the EuRoC benchmark. The outcomes confirmed the system's reliance on monocular vision. The experiments on EuRoC and Fast Flight sequences have shown remarkable accuracy in the trajectory estimation studies. We also evaluated the proposed scheme in terms of computational complexity, measured by CPU usage, where our monocular-vision optimization/filtering solution outperformed all the competing techniques.

This conclusion enforces our work's contributions to a reliable (fast and accurate) sensor fusion solution for challenging and large-scale environments. From a future perspective, it will be necessary to comprise situations where GPS sensor constraints, such as the multipath effects on the optimizer. Finally, further generalizing the optimization problem will be necessary to extend the algorithm's pose estimation capability to include multiple vision sensors (stereo RGB, for instance).

Author Contributions: Conceptualization, A.S.; methodology, A.S.; software, A.S.; validation, A.S., H.H.-A., F.B., D.S. and S.B.; formal analysis, A.S., H.H.-A. and F.B.; investigation, A.S.; resources, S.B.; data curation, A.S.; writing—original draft preparation, A.S.; writing—review and editing, S.B., D.S., H.H.-A., F.B. and A.S.; supervision, S.B., D.S., H.H.-A. and F.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Direction Générale de l'Armement (DGA) and the French National Research Agency as a part of the LOCA3D project in the framework of the MALIN challenge (https://challenge-malin.fr).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The EuRoC MAV benchmark is publicly available online at (https://projects.asl.ethz.ch/datasets/doku.php?id=kmavvisualinertialdatasets (accessed on 1 October 2022)). The Fast Flight dataset is publicly available online at (https://github.com/KumarRobotics/msckf_vio/wiki/Dataset (accessed on 3 October 2022)).

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

Abbreviations

The following abbreviations are ordered according to the mentioned precedence in this article:

MAV	Micro Aerial Vehicle
VO	Visual Odometry
VIO	Visual Inertial Odometry
SLAM	Simultaneous Localization Furthermore, Mapping
CT/DT	Continuous Time/Discrete Time
KLT	Kanade–Lucas Tracking
PGO	Pose Graph Optimization
ES-EKF	Error States Extended Kalman Filter
S-MSCKF	Stereo Multi-State Constraint Kalman Filter
S-UKF-LG	Stereo Unscented Kalman Filter on Lie Groups
S-IEKF	Stereo (Invariant-)Extended Kalman Filter
ATE	Absolute Trajectory Error
RMS	Root Mean Square
RMSE	Root Mean Square Error

Appendix A. Observability Analysis

The EKF-based VIO for 6-DOF motion estimate contains four unobservable states corresponding to the global position and rotation around the gravity axis, or yaw angle, as demonstrated in [44]. A simple EKF VIO implementation will gather false information about yaw. The different processes and measurements' linearizing point causes this unobservability. To ensure that the uncertainty of the current camera states in the state vector is not impacted by the uncertainty of the current IMU state during the propagation step, in our implementation, the camera poses in the state vector can be represented with respect to its inertial frame (v) instead of the latest IMU frame. Besides the efficient gyroscope RK4 integration scheme during the initialization process, our ES-EKF implementation minimizes the effect of the unobservable modes of the basic EKF. Figure A1 shows the IMU intrinsics, IMU-CAM extrinsic parameters, and odometry scale ES-EKF states plotted for sample EuRoC and Fast Flight sequences.

The main observation from Figure A1 is that when the motion of the MAV is smooth with no abrupt rotations and translations, our optimization-based initialization estimates an optimal metric-scaled trajectory with $\lambda = 1$. Moreover, we also observe that when the IMU-camera setup is not accurately calibrated, the ES-EKF can optimally align the sensor setup in a robust online calibration process. Furthermore, the estimated IMU biases using our ES-EKF model are accurate and in a sensible range. One crucial observation is the estimated attitude visual drift of the visual sensor and the detection of consistent drift patterns based on the MAV speed (Fast Flight sequences) and abrupt motions (EuRoC

sequences). These observations validate the contribution of the ES-EKF to the sustainability of the proposed method to achieve a resilient system that observes all the state vector parameters in addition to all the 6 DoF of the MAV trajectory. Finally, after the initial trajectory optimization, the filtering process is indispensable to estimate the false camera poses during long-term navigation caused by the visual attitude drifts.

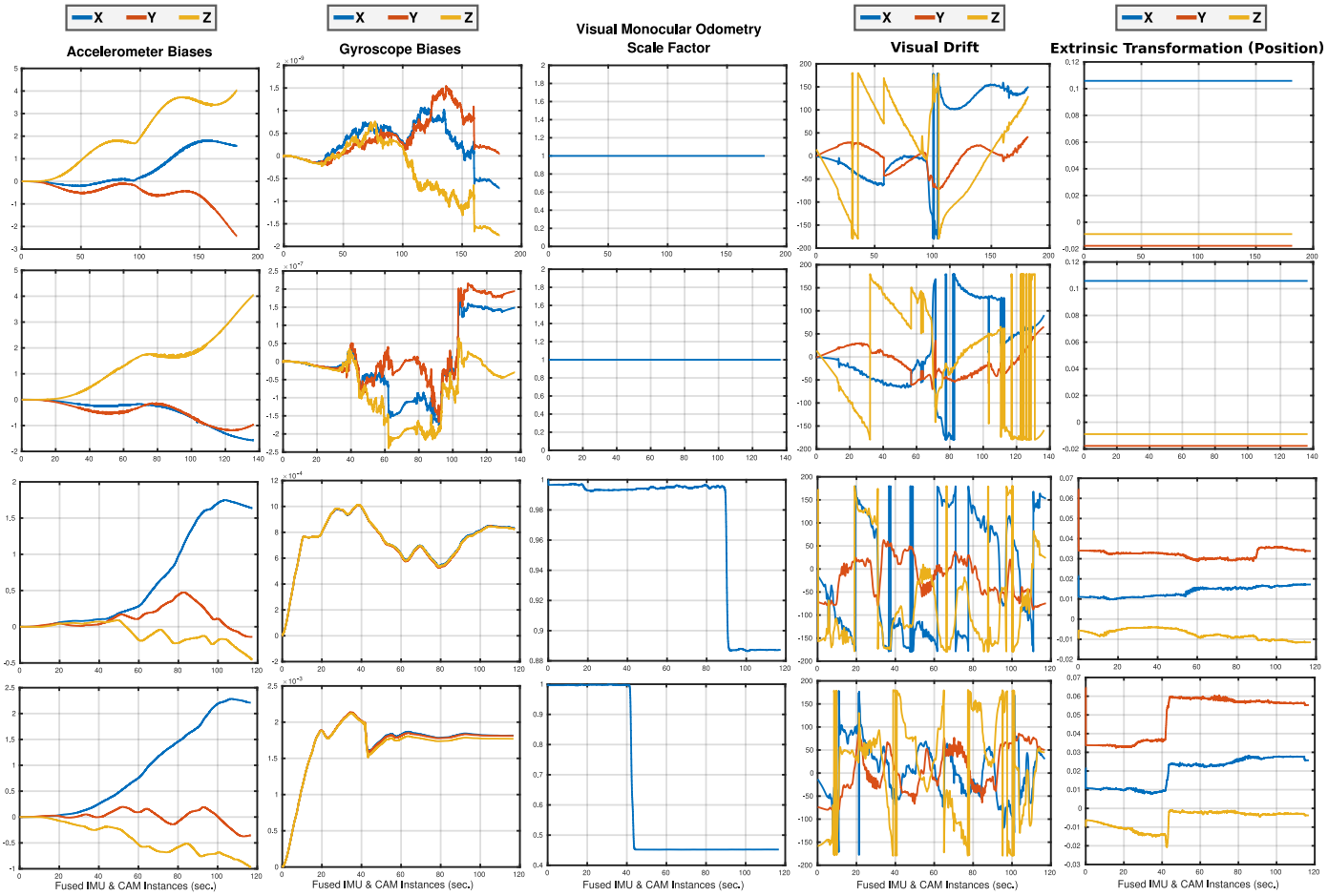

Figure A1. Our ES−EKF estimated states. Columns from left to right: IMU (accelerometer/gyroscope) biases b_a, b_ω, odometry scale factor λ, visual drift orientations q_v^w, and IMU-camera translation online calibration p_i^c. Rows 1,2 for sample Fast Flight sequences (gps5, gps10) and rows 3, 4 for sample EuRoC sequences (V2-02, V2-03), respectively.

Appendix B. Q_d Derivation Equations

The Q_d in Equation (16) can be obtained after the consecutive matrix multiplications are performed using the following formulas. For simplicity, let $t = \Delta t$, $\sigma = d\sigma$, $\beta = -R_{(\hat{q}_w^i)}$:

$$Q_{d_{11}} = \int_{\Delta t} (\sigma_{n_a}^2 . t^2 . \beta . \beta^\top + \sigma_{n_\omega}^2 . A . A^\top + \sigma_{n_{b_\omega}}^2 . B . B^\top + \sigma_{n_{b_a}}^2 . \frac{t^4}{4} . \beta . \beta^\top),$$

$$\begin{aligned} Q_{d_{11}} = {} & \sigma_{n_a}^2 . \frac{t^3}{3} . \beta . \beta^\top + \sigma_{n_\omega}^2 [(\beta \lfloor \hat{a} \rfloor_\times)(I_{d_3} . \frac{t^5}{20} + \frac{t^7}{504} \lfloor \hat{\omega} \rfloor_\times^2)(\beta \lfloor \hat{a} \rfloor_\times)^\top] + \dots \\ & + \sigma_{n_{b_\omega}}^2 [(\beta \lfloor \hat{a} \rfloor_\times)(I_{d_3} . \frac{t^7}{252} + \frac{t^9}{8640} \lfloor \hat{\omega} \rfloor_\times^2)(\beta \lfloor \hat{a} \rfloor_\times)^\top] + \sigma_{n_{b_a}}^2 . \frac{t^5}{20} . \beta . \beta^\top, \end{aligned}$$

$$Q_{d_{12}} = \int_{\Delta t} (\sigma_{n_a}^2 . t . \beta . \beta^\top + \sigma_{n_\omega}^2 . A . C^\top + \sigma_{n_{b_\omega}}^2 . B . D^\top + \sigma_{n_{b_a}}^2 . \frac{t^3}{2} . \beta . \beta^\top),$$

$$Q_{d_{12}} = \sigma_{n_a}^2.\frac{t^2}{2}.\beta.\beta^\top + \sigma_{n_\omega}^2[(\beta\lfloor\hat{a}\rfloor_\times)(I_{d_3}.\frac{t^4}{8} + \frac{t^5}{60}\lfloor\hat{\omega}\rfloor_\times + \frac{t^6}{144}\lfloor\hat{\omega}\rfloor_\times^2)(\beta\lfloor\hat{a}\rfloor_\times)^\top] + \dots$$
$$+\sigma_{n_{b_\omega}}^2[(\beta\lfloor\hat{a}\rfloor_\times)(I_{d_3}.\frac{t^6}{72} + \frac{t^7}{1008}\lfloor\hat{\omega}\rfloor_\times + \frac{t^8}{1920}\lfloor\hat{\omega}\rfloor_\times^2)(\beta\lfloor\hat{a}\rfloor_\times)^\top] + \sigma_{n_{b_a}}^2.\frac{t^4}{8}.\beta.\beta^\top,$$

$$Q_{d_{13}} = \int_{\Delta t}(\sigma_{n_\omega}^2.A.E^\top + \sigma_{n_{b_\omega}}^2.B.F^\top),$$

$$Q_{d_{13}} = \sigma_{n_\omega}^2[(\beta\lfloor\hat{a}\rfloor_\times)(I_{d_3}.\frac{t^3}{6} + \frac{t^4}{12}\lfloor\hat{\omega}\rfloor_\times + \frac{t^5}{40}\lfloor\hat{\omega}\rfloor_\times^2)] + \dots$$
$$+\sigma_{n_{b_\omega}}^2[(\beta\lfloor\hat{a}\rfloor_\times)(I_{d_3}.\frac{t^5}{30} + \frac{t^6}{144}\lfloor\hat{\omega}\rfloor_\times + \frac{11.t^7}{5040}\lfloor\hat{\omega}\rfloor_\times^2)],$$

$$Q_{d_{14}} = \int_{\Delta t}(\sigma_{n_{b_\omega}}^2.B) = \sigma_{n_{b_\omega}}^2[(\beta\lfloor\hat{a}\rfloor_\times)(-I_{d_3}.\frac{t^4}{24} + \frac{t^5}{120}\lfloor\hat{\omega}\rfloor_\times - \frac{t^6}{720}\lfloor\hat{\omega}\rfloor_\times^2)],$$

$$Q_{d_{15}} = \int_{\Delta t}(\sigma_{n_{b_a}}^2.\frac{t^2}{2}.\beta) = \sigma_{n_{b_a}}^2.\frac{t^3}{6}.\beta,$$

$$Q_{d_{16}} = \int_{\Delta t}(0_{3\times 13}) = 0_{3\times 13},$$

$$Q_{d_{21}} = \int_{\Delta t}(\sigma_{n_a}^2.t.\beta.\beta^\top + \sigma_{n_\omega}^2.C.A^\top + \sigma_{n_{b_\omega}}^2.D.B^\top + \sigma_{n_{b_a}}^2.\frac{t^3}{2}.\beta.\beta^\top),$$

$$Q_{d_{21}} = \sigma_{n_a}^2.\frac{t^2}{2}.\beta.\beta^\top + \sigma_{n_\omega}^2[(\beta\lfloor\hat{a}\rfloor_\times)(I_{d_3}.\frac{t^4}{8} - \frac{t^5}{60}\lfloor\hat{\omega}\rfloor_\times + \frac{t^6}{144}\lfloor\hat{\omega}\rfloor_\times^2)(\beta\lfloor\hat{a}\rfloor_\times)^\top] + \dots$$
$$+\sigma_{n_{b_\omega}}^2[(\beta\lfloor\hat{a}\rfloor_\times)(I_{d_3}.\frac{t^6}{72} - \frac{t^7}{1008}\lfloor\hat{\omega}\rfloor_\times + \frac{t^8}{1920}\lfloor\hat{\omega}\rfloor_\times^2)(\beta\lfloor\hat{a}\rfloor_\times)^\top] + \sigma_{n_{b_a}}^2.\frac{t^4}{8}.\beta.\beta^\top,$$

$$Q_{d_{22}} = \int_{\Delta t}(\sigma_{n_a}^2.\beta.\beta^\top + \sigma_{n_\omega}^2.C.C^\top + \sigma_{n_{b_\omega}}^2.D.D^\top + \sigma_{n_{b_a}}^2.t^2.\beta.\beta^\top),$$

$$Q_{d_{22}} = \sigma_{n_a}^2.t.\beta.\beta^\top + \sigma_{n_\omega}^2[(\beta\lfloor\hat{a}\rfloor_\times)(I_{d_3}.\frac{t^3}{3} + \frac{t^5}{60}\lfloor\hat{\omega}\rfloor_\times^2)(\beta\lfloor\hat{a}\rfloor_\times)^\top] + \dots$$
$$+\sigma_{n_{b_\omega}}^2[(\beta\lfloor\hat{a}\rfloor_\times)(I_{d_3}.\frac{t^5}{20} + \frac{t^7}{504}\lfloor\hat{\omega}\rfloor_\times^2)(\beta\lfloor\hat{a}\rfloor_\times)^\top] + \sigma_{n_{b_a}}^2.\frac{t^3}{3}.\beta.\beta^\top,$$

$$Q_{d_{23}} = \int_{\Delta t}(\sigma_{n_\omega}^2.C.E^\top + \sigma_{n_{b_\omega}}^2.D.F^\top),$$

$$Q_{d_{23}} = \sigma_{n_\omega}^2[(\beta\lfloor\hat{a}\rfloor_\times)(I_{d_3}.\frac{t^2}{2} + \frac{t^3}{6}\lfloor\hat{\omega}\rfloor_\times + \frac{t^4}{24}\lfloor\hat{\omega}\rfloor_\times^2)] + \dots$$
$$+\sigma_{n_{b_\omega}}^2[(\beta\lfloor\hat{a}\rfloor_\times)(I_{d_3}.\frac{t^4}{8} + \frac{t^5}{60}\lfloor\hat{\omega}\rfloor_\times + \frac{t^6}{144}\lfloor\hat{\omega}\rfloor_\times^2)],$$

$$Q_{d_{24}} = \int_{\Delta t}(\sigma_{n_{b_\omega}}^2.D) = \sigma_{n_{b_\omega}}^2[-(\beta\lfloor\hat{a}\rfloor_\times)(I_{d_3}.\frac{t^3}{6} - \frac{t^4}{24}\lfloor\hat{\omega}\rfloor_\times + \frac{t^5}{120}\lfloor\hat{\omega}\rfloor_\times^2)],$$

$$Q_{d_{25}} = \int_{\Delta t}(\sigma_{n_{b_a}}^2.t.\beta) = \sigma_{n_{b_a}}^2.\frac{t^2}{2}.\beta,$$

$$Q_{d_{26}} = \int_{\Delta t}(0_{3\times 13}) = 0_{3\times 13},$$

$$Q_{d_{31}} = \int_{\Delta t}(\sigma_{n_\omega}^2.E.A^\top + \sigma_{n_{b_\omega}}^2.F.B^\top),$$

$$Q_{d_{31}} = \sigma^2_{n_\omega}[(I_{d_3}.\frac{t^3}{6} - \frac{t^4}{12}\lfloor\hat{\omega}\rfloor_\times + \frac{t^5}{40}\lfloor\hat{\omega}\rfloor^2_\times)(\beta\lfloor\hat{a}\rfloor_\times)^\top] + \dots$$
$$+\sigma^2_{n_{b_\omega}}[(I_{d_3}.\frac{t^5}{30} - \frac{t^6}{144}\lfloor\hat{\omega}\rfloor_\times + \frac{11.t^7}{5040}\lfloor\hat{\omega}\rfloor^2_\times)(\beta\lfloor\hat{a}\rfloor_\times)^\top],$$

$$Q_{d_{32}} = \int_{\Delta t}(\sigma^2_{n_\omega}.E.C^\top + \sigma^2_{n_{b_\omega}}.F.D^\top),$$

$$Q_{d_{32}} = \sigma^2_{n_\omega}[(I_{d_3}.\frac{t^2}{2} - \frac{t^3}{6}\lfloor\hat{\omega}\rfloor_\times + \frac{t^4}{24}\lfloor\hat{\omega}\rfloor^2_\times)(\beta\lfloor\hat{a}\rfloor_\times)^\top] + \dots$$
$$+\sigma^2_{n_{b_\omega}}[(I_{d_3}.\frac{t^4}{8} - \frac{t^5}{60}\lfloor\hat{\omega}\rfloor_\times + \frac{t^6}{144}\lfloor\hat{\omega}\rfloor^2_\times)(\beta\lfloor\hat{a}\rfloor_\times)^\top],$$

$$Q_{d_{33}} = \int_{\Delta t}(\sigma^2_{n_\omega}.E.E^\top + \sigma^2_{n_{b_\omega}}.F.F^\top),$$

$$Q_{d_{33}} = \sigma^2_{n_\omega}[(I_{d_3}.t)] + \sigma^2_{n_{b_\omega}}[(I_{d_3}.\frac{t^3}{3} + \frac{t^5}{60}\lfloor\hat{\omega}\rfloor^2_\times)],$$

$$Q_{d_{34}} = \int_{\Delta t}(\sigma^2_{n_{b_\omega}}.F) = \sigma^2_{n_{b_\omega}}[(I_{d_3}.\frac{-t^2}{2} + \frac{t^3}{6}\lfloor\hat{\omega}\rfloor_\times - \frac{t^4}{24}\lfloor\hat{\omega}\rfloor^2_\times)],$$

$$Q_{d_{35}} = \int_{\Delta t}(0_{3\times3}) = 0_{3\times3},$$

$$Q_{d_{36}} = \int_{\Delta t}(0_{3\times13}) = 0_{3\times13},$$

$$Q_{d_{41}} = \int_{\Delta t}(\sigma^2_{n_{b_\omega}}.B^\top) = \sigma^2_{n_{b_\omega}}[(\beta\lfloor\hat{a}\rfloor_\times)(-I_{d_3}.\frac{t^4}{24} + \frac{t^5}{120}\lfloor\hat{\omega}\rfloor_\times - \frac{t^6}{720}\lfloor\hat{\omega}\rfloor^2_\times)]^\top,$$

$$Q_{d_{42}} = \int_{\Delta t}(\sigma^2_{n_{b_\omega}}.D^\top) = \sigma^2_{n_{b_\omega}}[-(\beta\lfloor\hat{a}\rfloor_\times)(I_{d_3}.\frac{t^3}{6} - \frac{t^4}{24}\lfloor\hat{\omega}\rfloor_\times + \frac{t^5}{120}\lfloor\hat{\omega}\rfloor^2_\times)]^\top,$$

$$Q_{d_{43}} = \int_{\Delta t}(\sigma^2_{n_{b_\omega}}.F^\top) = \sigma^2_{n_{b_\omega}}[(I_{d_3}.\frac{-t^2}{2} + \frac{t^3}{6}\lfloor\hat{\omega}\rfloor_\times - \frac{t^4}{24}\lfloor\hat{\omega}\rfloor^2_\times)]^\top,$$

$$Q_{d_{44}} = \int_{\Delta t}(\sigma^2_{n_{b_\omega}}) = \sigma^2_{n_{b_\omega}}.t,$$

$$Q_{d_{45}} = \int_{\Delta t}(0_{3\times3}) = 0_{3\times3},$$

$$Q_{d_{46}} = \int_{\Delta t}(0_{3\times13}) = 0_{3\times13},$$

$$Q_{d_{51}} = \int_{\Delta t}(\sigma^2_{n_{b_a}}.\beta^\top.\frac{t^2}{2}) = \sigma^2_{n_{b_\omega}}.\beta^\top.\frac{t^3}{6},$$

$$Q_{d_{52}} = \int_{\Delta t}(\sigma^2_{n_{b_a}}.\beta^\top.t) = \sigma^2_{n_{b_\omega}}.\beta^\top.\frac{t^2}{2},$$

$$Q_{d_{53}} = \int_{\Delta t}(0_{3\times3}) = 0_{3\times3},$$

$$Q_{d_{54}} = \int_{\Delta t}(0_{3\times3}) = 0_{3\times3},$$

$$Q_{d_{55}} = \int_{\Delta t}(\sigma^2_{n_{b_a}}) = \sigma^2_{n_{b_a}}.t,$$

$$Q_{d_{56}} = \int_{\Delta t}(0_{3\times13}) = 0_{3\times13},$$

$$Q_{d_{61\to65}} = \int_{\Delta t}(0_{13\times3}) = 0_{13\times3}, \quad Q_{d_{66}} = \int_{\Delta t}(0_{13\times13}) = 0_{13\times13}.$$

References

1. Soliman, A.; Bonardi, F.; Sidibé, D.; Bouchafa, S. IBISCape: A Simulated Benchmark for multi-modal SLAM Systems Evaluation in Large-scale Dynamic Environments. *J. Intell. Robot. Syst.* **2022**, *106*, 53. [CrossRef]
2. Dong, B.; Zhang, K. A Tightly Coupled Visual-Inertial GNSS State Estimator Based on Point-Line Feature. *Sensors* **2022**, *22*, 3391. [CrossRef] [PubMed]
3. Gu, N.; Xing, F.; You, Z. GNSS Spoofing Detection Based on Coupled Visual/Inertial/GNSS Navigation System. *Sensors* **2021**, *21*, 6769. [CrossRef] [PubMed]
4. Huang, W.; Wan, W.; Liu, H. Optimization-Based Online Initialization and Calibration of Monocular Visual-Inertial Odometry Considering Spatial-Temporal Constraints. *Sensors* **2021**, *21*, 2673. [CrossRef] [PubMed]
5. Ma, S.; Bai, X.; Wang, Y.; Fang, R. Robust Stereo Visual-Inertial Odometry Using Nonlinear Optimization. *Sensors* **2019**, *19*, 3747. [CrossRef] [PubMed]
6. Zhang, S.; Wang, W.; Li, H.; Zhang, S. EVtracker: An Event-Driven Spatiotemporal Method for Dynamic Object Tracking. *Sensors* **2022**, *22*, 6090. [CrossRef]
7. Ren, G.; Yu, Y.; Liu, H.; Stathaki, T. Dynamic Knowledge Distillation with Noise Elimination for RGB-D Salient Object Detection. *Sensors* **2022**, *22*, 6188. [CrossRef]
8. Alliez, P.; Bonardi, F.; Bouchafa, S.; Didier, J.Y.; Hadj-Abdelkader, H.; Muñoz, F.I.I.; Kachurka, V.; Rault, B.; Robin, M.; Roussel, D. Real-Time Multi-SLAM System for Agent Localization and 3D Mapping in Dynamic Scenarios. In Proceedings of the International Confererence on Intelligent Robots and Systems (IROS 2020), Las Vegas, NV, USA, 25–29 October 2020.
9. Alonge, F.; Cusumano, P.; D'Ippolito, F.; Garraffa, G.; Livreri, P.; Sferlazza, A. Localization in Structured Environments with UWB Devices without Acceleration Measurements, and Velocity Estimation Using a Kalman-Bucy Filter. *Sensors* **2022**, *22*, 6308. [CrossRef]
10. Cao, S.; Gao, H.; You, J. In-Flight Alignment of Integrated SINS/GPS/Polarization/Geomagnetic Navigation System Based on Federal UKF. *Sensors* **2022**, *22*, 5985. [CrossRef]
11. 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]
12. Burri, M.; Nikolic, J.; Gohl, P.; Schneider, T.; Rehder, J.; Omari, S.; Achtelik, M.W.; Siegwart, R. The EuRoC micro aerial vehicle datasets. *Int. J. Robot. Res.* **2016**, *35*, 1157–1163.
13. Qin, T.; Pan, J.; Cao, S.; Shen, S. A general optimization-based framework for local odometry estimation with multiple sensors. *arXiv* **2019**, arXiv:1901.03638.
14. Yu, Y.; Gao, W.; Liu, C.; Shen, S.; Liu, M. A GPS-aided Omnidirectional Visual-Inertial State Estimator in Ubiquitous Environments. In Proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, China, 3–8 November 2019; pp. 7750–7755. [CrossRef]
15. Mascaro, R.; Teixeira, L.; Hinzmann, T.; Siegwart, R.; Chli, M. GOMSF: Graph-Optimization Based Multi-Sensor Fusion for robust UAV Pose estimation. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, QLD, Australia, 21–25 May 2018; pp. 1421–1428. [CrossRef]
16. Cioffi, G.; Scaramuzza, D. Tightly-coupled Fusion of Global Positional Measurements in Optimization-based Visual-Inertial Odometry. In Proceedings of the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, 24 October 2020–24 January 2021; pp. 5089–5095. [CrossRef]
17. Dai, J.; Liu, S.; Hao, X.; Ren, Z.; Yang, X. UAV Localization Algorithm Based on Factor Graph Optimization in Complex Scenes. *Sensors* **2022**, *22*, 5862. [CrossRef]
18. Mourikis, A.I.; Roumeliotis, S.I. A Multi-State Constraint Kalman Filter for Vision-aided Inertial Navigation. In Proceedings of the 2007 IEEE International Conference on Robotics and Automation, Rome, Italy, 10–14 April 2007; pp. 3565–3572. [CrossRef]
19. Brossard, M.; Bonnabel, S.; Barrau, A. Unscented Kalman Filter on Lie Groups for Visual Inertial Odometry. In Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1–5 October 2018; pp. 649–655. [CrossRef]
20. Bloesch, M.; Burri, M.; Omari, S.; Hutter, M.; Siegwart, R. Iterated extended Kalman filter based visual-inertial odometry using direct photometric feedback. *Int. J. Robot. Res.* **2017**, *36*, 1053–1072.
21. Brunello, A.; Urgolo, A.; Pittino, F.; Montvay, A.; Montanari, A. Virtual Sensing and Sensors Selection for Efficient Temperature Monitoring in Indoor Environments. *Sensors* **2021**, *21*, 2728. [CrossRef]
22. 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]
23. Leutenegger, S.; Lynen, S.; Bosse, M.; Siegwart, R.; Furgale, P. Keyframe-based visual—Inertial odometry using nonlinear optimization. *Int. J. Robot. Res.* **2015**, *34*, 314–334.
24. Campos, C.; Elvira, R.; Rodríguez, J.J.G.; Montiel, J.M.; Tardós, J.D. OrbSLAM3: An accurate open-source library for visual, visual–inertial, and multimap slam. *IEEE Trans. Robot.* **2021**, *37*, 1874–1890. [CrossRef]
25. Usenko, V.; Demmel, N.; Schubert, D.; Stückler, J.; Cremers, D. Visual-inertial mapping with non-linear factor recovery. *IEEE Robot. Autom. Lett.* **2019**, *5*, 422–429. [CrossRef]
26. Schimmack, M.; Haus, B.; Mercorelli, P. An Extended Kalman Filter as an Observer in a Control Structure for Health Monitoring of a Metal–Polymer Hybrid Soft Actuator. *IEEE/ASME Trans. Mechatron.* **2018**, *23*, 1477–1487. TMECH.2018.2792321. [CrossRef]

27. Mercorelli, P. A switching Kalman Filter for sensorless control of a hybrid hydraulic piezo actuator using MPC for camless internal combustion engines. In Proceedings of the 2012 IEEE International Conference on Control Applications, Dubrovnik, Croatia, 3–5 October 2012; pp. 980–985. [CrossRef]
28. Huang, G.; Kaess, M.; Leonard, J.J. Towards consistent visual-inertial navigation. In Proceedings of the 2014 IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China, 31 May–7 June 2014; pp. 4926–4933. [CrossRef]
29. Huang, P.; Meyr, H.; Dörpinghaus, M.; Fettweis, G. Observability Analysis of Flight State Estimation for UAVs and Experimental Validation. In Proceedings of the 2020 IEEE International Conference on Robotics and Automation (ICRA), Paris, France, 31 May–31 August 2020; pp. 4659–4665. [CrossRef]
30. Cioffi, G.; Cieslewski, T.; Scaramuzza, D. Continuous-Time Vs. Discrete-Time Vision-Based SLAM: A Comparative Study. *IEEE Robot. Autom. Lett.* **2022**, *7*, 2399–2406. [CrossRef]
31. Nurhakim, A.; Ismail, N.; Saputra, H.M.; Uyun, S. Modified Fourth-Order Runge-Kutta Method Based on Trapezoid Approach. In Proceedings of the 2018 4th International Conference on Wireless and Telematics (ICWT), Nusa Dua, Bali, Indonesia, 12–13 July 2018; pp. 1–5. [CrossRef]
32. Lv, M.; Wei, H.; Fu, X.; Wang, W.; Zhou, D. A Loosely Coupled Extended Kalman Filter Algorithm for Agricultural Scene-Based Multi-Sensor Fusion. *Front. Plant Sci.* **2022**, *13*, 9260. [CrossRef] [PubMed]
33. Sola, J. Quaternion kinematics for the error-state Kalman filter. *arXiv* **2017**, arXiv:1711.02508.
34. Sommer, C.; Usenko, V.; Schubert, D.; Demmel, N.; Cremers, D. Efficient Derivative Computation for Cumulative B-Splines on Lie Groups. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, 13–19 June 2020; pp. 11145–11153. [CrossRef]
35. Trawny, N.; Roumeliotis, S.I. Indirect Kalman filter for 3D attitude estimation. *Eng. Tech. Rep.* **2005**, *2*, 2005.
36. Moulon, P.; Monasse, P.; Marlet, R. Adaptive Structure from Motion with a Contrario Model Estimation. In Proceedings of the Computer Vision—ACCV 2012, Daejeon, Republic of Korea, 5–9 November 2012; Lee, K.M., Matsushita, Y., Rehg, J.M., Hu, Z., Eds.; Springer: Berlin/Heidelberg, Germany, 2013; pp. 257–270.
37. Nister, D. An efficient solution to the five-point relative pose problem. *IEEE Trans. Pattern Anal. Mach. Intell.* **2004**, *26*, 756–770. [CrossRef]
38. Tomasi, C.; Kanade, T. Detection and tracking of point. *Int. J. Comput. Vis.* **1991**, *9*, 137–154. [CrossRef]
39. Wang, Y.; Chirikjian, G.S. Nonparametric second-order theory of error propagation on motion groups. *Int. J. Robot. Res.* **2008**, *27*, 1258–1273. [CrossRef]
40. Agarwal, S.; Mierle, K. Ceres Solver. Available online: https://github.com/ceres-solver/ceres-solver (accessed on 10 October 2022).
41. Geneva, P.; Eckenhoff, K.; Lee, W.; Yang, Y.; Huang, G. OpenVINS: A Research Platform for Visual-Inertial Estimation. In Proceedings of the 2020 IEEE International Conference on Robotics and Automation (ICRA), Paris, France, 31 May–31 August 2020; pp. 4666–4672. [CrossRef]
42. Zuo, X.; Merrill, N.; Li, W.; Liu, Y.; Pollefeys, M.; Huang, G.P. CodeVIO: Visual-Inertial Odometry with Learned Optimizable Dense Depth. In Proceedings of the 2021 IEEE International Conference on Robotics and Automation (ICRA), Xi'an, China, 30 May–5 June 2021; pp. 14382–14388.
43. Rosinol, A.; Abate, M.; Chang, Y.; Carlone, L. Kimera: An open-source library for real-time metric-semantic localization and mapping. In Proceedings of the 2020 IEEE International Conference on Robotics and Automation (ICRA), Paris, France, 31 May–31 August 2020; pp. 1689–1696.
44. Li, M.; Mourikis, A.I. High-precision, consistent EKF-based visual-inertial odometry. *Int. J. Robot. Res.* **2013**, *32*, 690–711. [CrossRef]