aerospace

Article

# GNSS-Denied Semi-Direct Visual Navigation for Autonomous UAVs Aided by PI-Inspired Inertial Priors

**Eduardo Gallo** * and **Antonio Barrientos**

Centro de Automática y Robótica, Universidad Politécnica de Madrid—Consejo Superior de Investigaciones Científicas, 28006 Madrid, Spain; antonio.barrientos@upm.es

* Correspondence: e.gallo@alumnos.upm.es or edugallo@yahoo.com; Tel.: +34-600643104

**Abstract:** This article proposes a method to diminish the horizontal position drift in the absence of GNSS (Global Navigation Satellite System) signals experienced by the VNS (Visual Navigation System) installed onboard a UAV (Unmanned Air Vehicle) by supplementing its pose estimation non-linear optimizations with priors based on the outputs of the INS (Inertial Navigation System). The method is inspired by a PI (Proportional Integral) control loop, in which the attitude and altitude inertial outputs act as targets to ensure that the visual estimations do not deviate past certain thresholds from their inertial counterparts. The resulting IA-VNS (Inertially Assisted Visual Navigation System) achieves major reductions in the horizontal position drift inherent to the GNSS-Denied navigation of autonomous UAVs. Stochastic high-fidelity Monte Carlo simulations of two representative scenarios involving the loss of GNSS signals are employed to evaluate the results and to analyze their sensitivity to the terrain type overflown by the aircraft. The authors release the C++ implementation of both the navigation algorithms and the high-fidelity simulation as open-source software.

**Keywords:** GNSS-Denied; visual inertial navigation; autonomous navigation; autonomy; UAV; optimization

check for updates

**Citation:** Gallo, E.; Barrientos, A. GNSS-Denied Semi-Direct Visual Navigation for Autonomous UAVs Aided by PI-Inspired Inertial Priors. *Aerospace* **2023**, *10*, 220. https://doi.org/10.3390/aerospace10030220

Academic Editors: Gokhan Inalhan and Yan Xu

Received: 12 January 2023
Revised: 21 February 2023
Accepted: 22 February 2023
Published: 25 February 2023

## 1. Mathematical Notation

Any variable with a hat accent $< \hat{\cdot} >$ refers to its (inertial) estimated value, and with a circular accent $< \mathring{\cdot} >$ to its (visual) estimated value. In the case of vectors, which are displayed in bold (e.g., $\mathbf{x}$), other employed symbols include the wide hat $< \widehat{\cdot} >$, which refers to the skew-symmetric form, the bar $< \bar{\cdot} >$, which represents the vector homogeneous coordinates, and the double vertical bars $< \| \cdot \| >$, which refer to the norm. In the case of scalars, the vertical bars $< | \cdot | >$ refer to the absolute value. When employing attitudes and rigid body poses (e.g., $\mathbf{q}$ and $\zeta$), the asterisk superindex $< \cdot^* >$ refers to the conjugate, their concatenation and multiplication are represented by $\circ$ and $\otimes$, respectively, and $\oplus$ and $\ominus$ refer to the plus and minus operators.

This article includes various non-linear optimizations solved in the spaces of both rigid body rotations and full motions, instead of Euclidean spaces. Hence, it relies on the Lie algebra of the special orthogonal group of $\mathbb{R}^3$, known as $\mathbb{SO}(3)$, and that of the special Euclidean group of $\mathbb{R}^3$, represented by $\mathbb{SE}(3)$, in particular what refers to the groups actions, concatenations, perturbations, and Jacobians, as well as with their tangent spaces (the rotation vector $\mathbf{r}$ and angular velocity $\boldsymbol{\omega}$ for rotations, the transform vector $\boldsymbol{\tau}$ and twist $\boldsymbol{\xi}$ for motions). Refs. [1–3] are recommended as references.

Five different reference frames are employed in this article: the ECEF frame $\mathrm{F_E}$ (centered at the Earth center of mass $\mathrm{O_E}$, with $\mathbf{i}_3^{\mathrm{E}}$ pointing towards the geodetic North along the Earth rotation axis, $\mathbf{i}_1^{\mathrm{E}}$ contained in both the Equator and zero longitude planes, and $\mathbf{i}_2^{\mathrm{E}}$ orthogonal to $\mathbf{i}_1^{\mathrm{E}}$ and $\mathbf{i}_3^{\mathrm{E}}$ forming a right handed system), the NED frame $\mathrm{F_N}$ (centered at the aircraft center of mass $\mathrm{O_N}$, with axes aligned with the geodetic North, East, and Down directions), the body frame $\mathrm{F_B}$ (centered at the aircraft center of mass $\mathrm{O_B} = \mathrm{O_N}$, with $\mathbf{i}_1^{\mathrm{B}}$ contained in the plane of symmetry of the aircraft pointing forward along a fixed direction, $\mathbf{i}_3^{\mathrm{B}}$ contained in the plane of symmetry of the aircraft, normal to $\mathbf{i}_1^{\mathrm{B}}$ and pointing downward, and $\mathbf{i}_2^{\mathrm{B}}$ orthogonal to both in such a way that they form a right hand system), the camera frame $\mathrm{F_C}$ (centered at the optical center $\mathrm{O_C}$, defined in Appendix A, with $\mathbf{i}_3^{\mathrm{C}}$ located in the camera principal axis pointing forward, and $\mathbf{i}_1^{\mathrm{C}}$, $\mathbf{i}_2^{\mathrm{C}}$ parallel to the focal plane), and the image frame $\mathrm{F_{IMG}}$ (two-dimensional frame centered at the sensor corner with axes parallel to the sensor borders). The first three frames are graphically depicted in Figure 1, while $\mathrm{F_C}$ and $\mathrm{F_{IMG}}$ can be visualized in Appendix A.

Superindexes are employed over vectors to specify the reference frame in which they are viewed (e.g., $\mathbf{v}^{\mathrm{N}}$ refers to ground velocity viewed in $\mathrm{F_N}$, while $\mathbf{v}^{\mathrm{B}}$ is the same vector but viewed in $\mathrm{F_B}$). Subindexes may be employed to clarify the meaning of the variable or vector, such as in $\mathbf{v}_{\mathrm{TAS}}$ for air velocity instead of the ground velocity $\mathbf{v}$, in which case the subindex is either an acronym or its meaning is clearly explained when first introduced. Subindexes may also refer to a given component of a vector, e.g., $\mathrm{v}_2^{\mathrm{N}}$ refers to the second component of $\mathbf{v}^{\mathrm{N}}$. In addition, where two reference frames appear as subindexes to a vector, it means that the vector goes from the first frame to the second. For example, $\boldsymbol{\omega}_{\mathrm{NB}}^{\mathrm{B}}$ refers to the angular velocity from the $\mathrm{F_N}$ frame to the $\mathrm{F_B}$ frame viewed in $\mathrm{F_B}$. Table 1 summarizes the notation employed in this article.

**Table 1.** Mathematical notation.

| | | | |
|---|---|---|---|
| $\gamma_{\mathrm{TAS}}$ | Aerodynamic path angle | $\mathbf{g}$ | Lie group action (transformation) |
| $\delta$ | Error threshold | $\mathrm{h}$ | Geometric altitude |
| $\delta_{\mathrm{CNTR}}$ | Throttle and control surfaces position | $\mathrm{H_P}$ | Pressure altitude |
| $\delta_{\mathrm{TARGET}}$ | Control targets | $\mathbf{I}$ | Camera image |
| $\varDelta$ | Estimation error, increment | $\mathbf{J}$ | Jacobian |
| $\varDelta\mathrm{p}$ | Atmospheric pressure offset | $\mathcal{M}$ | $\mathbb{SE}(3)$ Lie group element |
| $\varDelta\mathrm{T}$ | Atmospheric temperature offset | $\mathbf{p}$ | Point, feature |
| $\theta$ | Body pitch angle | $\mathbf{q}$ | Attitude, unit quaternion |
| $\lambda$ | Longitude | $\mathbf{r}$ | Attitude, rotation vector |
| $\zeta$ | Pose, unit dual quaternion | $\mathbf{R}$ | Attitude, rotation matrix |
| $\mu$ | Mean or expected value | $\mathcal{R}$ | $\mathbb{SO}(3)$ Lie group element |
| $\xi$ | Body bank angle | $\mathrm{s_{PX}}$ | Pixel size |
| $\xi$ | Motion ($\mathbb{SE}(3)$) velocity or twist | $\mathrm{S}$ | Sensor dimension |
| $\varPi$ | Camera projection | $\mathrm{t}$ | Time |
| $\varrho_{\mathrm{TUK}}$ | Tukey error function | $\mathbf{T}$ | Displacement |
| $\sigma$ | Standard deviation | $\mathbf{T}^{\mathrm{E,GDT}}$ | Geodetic coordinates |
| $\tau$ | Pose, transform vector | $\mathrm{v}$ | Speed |
| $\varphi$ | Latitude | $\mathbf{v}$ | Velocity |
| $\phi$ | Attitude, Euler angles | $\mathrm{w_{TUK}}$ | Tukey weight function |
| $\emptyset$ | Bearing | $\mathrm{x}$ | Horizontal distance |
| $\psi$ | Heading or body yaw angle | $\mathbf{x}$ | Position |
| $\boldsymbol{\omega}$ | Angular ($\mathbb{SO}(3)$) velocity | $\hat{\mathbf{x}} = \mathbf{x}_{\mathrm{EST}}$ | Inertial estimated trajectory |
| $\mathrm{E_{PO}}$ | Pose optimization error | $\mathring{\mathbf{x}} = \mathbf{x}_{\mathrm{IMG}}$ | Visual estimated trajectory |
| $\mathrm{E_q}$ | Attitude adjustment error | $\mathbf{x}_{\mathrm{REF}}$ | Reference objectives |
| $\mathrm{E_{RP}}$ | Reprojection error | $\widetilde{\mathbf{x}} = \mathbf{x}_{\mathrm{SENSED}}$ | Sensed trajectory |
| $\mathrm{f}$ | Focal length | $\mathbf{x} = \mathbf{x}_{\mathrm{TRUTH}}$ | Real trajectory |

In addition, there exist various indexes that appear as subindexes: $n$ identifies a discrete time instant ($\mathrm{t_n}$) for the inertial estimations, $s$ ($\mathrm{t_s}$) refers to the sensor outputs, $i$ identifies an image or frame ($\mathrm{t_i}$), and $k$ is employed for the keyframes used to generate the map or terrain structure. Other employed subindexes are $l$ for the steps of the various iteration processes that take place, and $j$ for the features and associated 3D points. With respect to superindexes, two stars $< \cdot^{\star\star} >$ represent the reprojection only solution, while two circles $< \cdot^{\circ\circ} >$ identify a target.

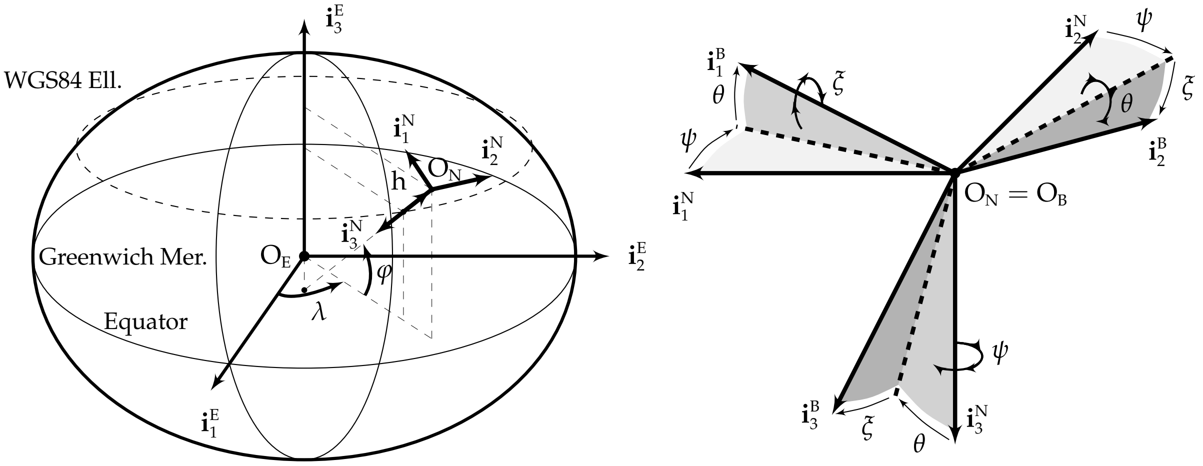

**Figure 1.** ECEF ($F_E$), NED ($F_N$), and body ($F_B$) reference frames.

## 2. Introduction and Outline

This article focuses on the need to develop navigation systems capable of diminishing the position drift inherent to the flight in GNSS (Global Navigation Satellite System)-Denied conditions of an autonomous fixed wing aircraft so it has a higher probability of reaching the vicinity of a recovery point, from where it can be landed by remote control.

The article proposes a method that employs the inertial navigation outputs to improve the accuracy of VO (Visual Odometry) algorithms, which rely on the images of the Earth surface provided by a down looking camera rigidly attached to the aircraft structure, resulting in major improvements in horizontal position estimation accuracy over what can be achieved by standalone inertial or visual navigation systems. In contrast with most visual inertial methods found in the literature, which focus on short term GNSS-Denied navigation of ground vehicles, robots, and multi-rotors, the proposed algorithms are primarily intended for the long distance GNSS-Denied navigation of autonomous fixed wing aircraft.

Section 3 describes the article objectives, novelty, and main applications. When processing a new image, VO pipelines include a distinct phase known as *pose optimization*, *pose refinement*, or *motion-only bundle adjustment*, which estimates the camera pose (position plus attitude) based on previously estimated positions for the identified terrain features, both as ECEF 3D coordinates, as well as 2D coordinates of their projected location in the current image. Section 4 reviews the pose optimization algorithm when part of a standalone visual navigation system that can only rely on periodically generated images, while Section 5 proposes improvements to take advantage of the availability of aircraft pose estimations provided by an inertial navigation system.

Section 6 introduces the stochastic high-fidelity simulation employed to evaluate the navigation results by means of Monte Carlo executions of two scenarios representative of the challenges of GNSS-Denied navigation. The results obtained when applying the proposed algorithms to these two GNSS-Denied scenarios are described in Section 7, comparing them with those achieved by standalone inertial and visual systems. Section 8 discusses the sensitivity of the estimations to the type of terrain overflown by the aircraft, as the terrain texture (or lack of) and its elevation relief are key factors on the ability of the visual algorithms to detect and track terrain features. Last, the results are summarized for convenience in Section 9, while Section 10 provides a short conclusion.

Following a list of acronyms, the article concludes with three appendices. Appendix A provides a detailed description of the concept of optical flow, which is indispensable for the pose optimization algorithms of Sections 4 and 5. Appendix B contains an introduction to GNSS-Denied navigation and its challenges, together with reviews of the state-of-the-art in two of the most promising routes to diminish its negative effects, such as visual odometry (VO) and visual inertial odometry (VIO). Last, Appendix C describes the different algorithms within Semi-Direct Visual Odometry (SVO) [4,5], a publicly available VO pipeline employed in this article, both by itself in Section 4 when relying exclusively on the images, and in the proposed improvements of Section 5 taking advantage of the inertial estimations.

## 3. Objective, Novelty, and Application

The main objective of this article is to improve the GNSS-Denied navigation capabilities of autonomous aircraft, so in case GNSS signals become unavailable, they can continue their mission or safely fly to a predetermined recovery location. To do so, the proposed approach combines two different navigation algorithms, employing the outputs of an INS (Inertial Navigation System) specifically designed for the flight without GNSS signals of an autonomous fixed wing low SWaP (Size, Weight, and Power) aircraft [6] to diminish the horizontal position drift generated by a VNS (Visual Navigation System) that relies on an advanced visual odometry pipeline, such as SVO [4,5]. Note that the INS makes use of all onboard sensors except the camera, while the VNS relies exclusively on the images provided by the camera.

As shown in Section 7, each of the two systems by itself incurs in unrestricted and excessive horizontal position drift that renders them inappropriate for long term GNSS-Denied navigation, but for different reasons: while in the INS the drift is the result of integrating the bounded ground velocity estimations without absolute position observations, that of the VNS originates on the slow but continuous accumulation of estimation errors between consecutive frames. The two systems however differ in their estimations of the aircraft attitude and altitude, as they are bounded for the INS but also drift in the case of the VNS. The proposed approach modifies the VNS so in addition to the images it can also accept as inputs the INS bounded attitude and altitude outputs, converting it into an Inertially Assisted VNS or IA-VNS with vastly improved horizontal position estimation capabilities.

The VIO solutions listed in Appendix B are quite generic with respect to the platforms on which they are mounted, with most applications focused on ground vehicles, indoor robots, and multi-rotors, as well as with respect to the employed sensors, which are usually restricted to the gyroscopes and accelerometers, together with one or more cameras. This article focuses on an specific case (long distance GNSS-Denied turbulent flight of fixed wing aircraft), and, as such, is simultaneously more restrictive but also takes advantage of the sensors already present onboard these platforms, such as magnetometers, Pitot tube, and air vanes. In addition, and unlike the existing VIO packages, the proposed solution assumes that GNSS signals are present at the beginning of the flight. As described in detail in [6], these are key to the obtainment of the bounded attitude and altitude INS outputs on which the proposed IA-VNS relies.

The proposed method represents a novel approach to diminish the pose drift of a VO pipeline by supplementing its pose estimation non-linear optimizations with priors based on the bounded attitude and altitude outputs of a GNSS-Denied inertial filter. The method is inspired in a PI (Proportional Integral) control loop, in which the inertial attitude and altitude outputs act as targets to ensure that the visual estimations do not deviate in excess from their inertial counterparts, resulting in major reductions to not only the visual attitude and altitude estimation errors, but also to the drift in horizontal position.

This article proves that inertial and visual navigation systems can be combined in such a way that the resulting long term GNSS-Denied horizontal position drift is significantly smaller than what can be obtained by either system individually. In the case that GNSS signals become unavailable in mid flight, GNSS-Denied navigation is required for the platform to complete its mission or return to base without the absolute position and ground velocity observations provided by GNSS receivers. As shown in the following sections, the proposed system can significantly increase the possibilities of the aircraft safely reaching the vicinity of the intended recovery location, from where it can be landed by remote control.

## 4. Pose Optimization within Visual Odometry

Visual navigation, also known as visual odometry or VO, relies on images of the Earth's surface generated by an onboard camera to incrementally estimate the aircraft pose (position plus attitude) based on the changes that its motion induces on the images, without the assistance of image databases or the observations of any other onboard sensors. As it does not rely on GNSS signals, it is considered an alternative to GNSS-Denied inertial navigation, although it also incurs in an unrestricted horizontal position drift. Appendix B.2 provides an overview of various VO pipelines within the broader context of the problems associated to GNSS-Denied navigation and the research paths most likely to diminish them (Appendix B).

This article employs SVO (Semi-Direct Visual Odometry) [4,5], a state-of-the-art publicly available VO pipeline, as a baseline on which to apply the proposed improvements based on the availability of inertial estimations of the aircraft pose. Although Appendix C describes the various threads and processes within SVO, the focus of the proposed improvements within Section 5 lies in the *pose optimization* phase, which is the only one described in detail in this article. Note that other VO pipelines also make use of similar pose optimization algorithms.

Graphically depicted in Figure 2, pose optimization is executed for every new frame $i$ and estimates the pose between the ECEF ($\mathrm{F_E}$) and camera ($\mathrm{F_C}$) frames ($\mathring{\zeta}_{\mathrm{ECi}}$). It requires the following inputs:

- The ECEF terrain 3D coordinates of all features $j$ visible in the image ($\mathbf{p}_{\mathrm{j}}^{\mathrm{E}}$) obtained by the structure optimization phase (Appendix C) corresponding to the previous image. These terrain 3D coordinates are known as the terrain *map*, and constitute a side product generated by VO pipelines.
- The 2D position of the same features $j$ within the current image $i$ ($\mathbf{p}_{\mathrm{ij}}^{\mathrm{IMG}}$) supplied by the previous feature alignment phase (Appendix C).
- The rough estimation of the ECEF to camera pose $\mathring{\zeta}_{\mathrm{ECi}}^{\star}$ for the current frame $i$ provided by the sparse image alignment phase (Appendix C), which acts as the initial value for the camera pose ($\mathring{\zeta}_{\mathrm{ECi0}}$) to be refined by iteration.

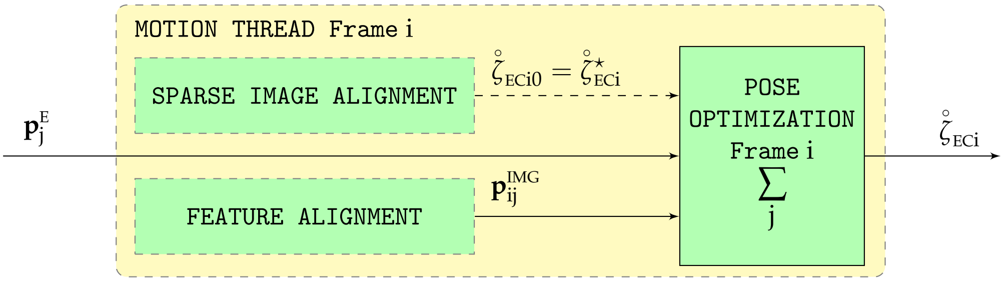
MOTION THREAD Frame i
SPARSE IMAGE ALIGNMENT
$\mathring{\zeta}_{\mathrm{ECi0}} = \mathring{\zeta}_{\mathrm{ECi}}^{\star}$
$\mathbf{p}_{\mathrm{j}}^{\mathrm{E}}$
FEATURE ALIGNMENT
$\mathbf{p}_{\mathrm{ij}}^{\mathrm{IMG}}$
POSE OPTIMIZATION Frame i
$\sum_{\mathrm{j}}$
$\mathring{\zeta}_{\mathrm{ECi}}$

**Figure 2.** Pose optimization flow diagram.

The pose optimization algorithm, also known as pose refinement or motion-only bundle adjustment, estimates the camera pose by minimizing the reprojection error of the different features. Pose optimization relies exclusively on the information obtained from the images generated by the onboard camera, and is described in detail to act as a baseline on which to apply in Section 5 the proposed improvements enabled by the availability of additional pose estimations generated by an inertial navigation system or INS.

The *reprojection error* $\mathrm{E_{RPi}}$, a function of the estimated ECEF to camera pose for image $i$ ($\mathring{\zeta}_{\mathrm{ECi}}$), is defined in (1) as the sum for each feature terrain 3D point $j$ of the norm of the difference between the camera projection $\mathit{\Pi}$ of the ECEF coordinates $\mathbf{p}_{\mathrm{j}}^{\mathrm{E}}$ transformed into the camera frame and the image coordinates $\mathbf{p}_{\mathrm{j}}^{\mathrm{IMG}}$. Note that $\mathbf{g}_{\zeta_{\mathrm{AB}}}()$ represents the $\mathbb{SE}(3)$

transformation of a point from frame B to frame A, as described in [1], and the camera projection $\Pi$ is defined in Appendix A.

$$\mathrm{E_{RPi}}\left(\mathring{\zeta}_{\mathrm{ECi}}\right) = \sum_{\mathrm{j}} \left\| \Pi\left(\mathbf{g}^{-1}_{\mathring{\zeta}_{\mathrm{ECi}}}\left(\mathbf{p}^{\mathrm{E}}_{\mathrm{j}}\right)\right) - \mathbf{p}^{\mathrm{IMG}}_{\mathrm{ij}} \right\| \tag{1}$$

This problem can be solved by means of an iterative Gauss-Newton gradient descent process [1,7]. Given an initial camera pose estimation $\mathring{\zeta}_{\mathrm{ECi0}}$ taken from the sparse image alignment result ($\mathring{\zeta}^{\star}_{\mathrm{ECi}}$, Figure 2), each iteration step $l$ minimizes (2) and advances the estimated solution by means of (3) until the step diminution of the reprojection error falls below a given threshold $\delta_{\mathrm{RP}}$ ($\mathrm{E_{RP,i,l}} - \mathrm{E_{RP,i,l+1}} < \delta_{\mathrm{RP}}$). Note that $\Delta\mathring{\tau}^{\mathrm{C}}_{\mathrm{ECil}}$ represents the estimated tangent space incremental ECEF to camera pose (transform vector) viewed in the $\mathrm{F_C}$ camera frame for image $i$ and iteration $l$, $\oplus$ and $\circ$ represent the $\mathbb{SE}(3)$ plus and concatenation operators, and $\mathrm{Exp}()$ refers to the $\mathbb{SE}(3)$ capitalized exponential function [1,3]. Additionally, note that, while $\mathrm{E_{RPi}}$ and $\mathrm{E_{RPi,l+1}}$ present in (1) and (2) are both positive scalars, the feature $j$ reprojection error $\mathbf{E}_{\mathrm{RPi,l+1,j}}$ that appears in (2) is an $\mathbb{R}^3$ vector.

$$\begin{aligned}
\mathrm{E_{RPi,l+1}}\left(\Delta\mathring{\tau}^{\mathrm{C}}_{\mathrm{ECil}}\right) &= \sum_{\mathrm{j}} \left\| \Pi\left(\mathbf{g}^{-1}_{\mathring{\zeta}_{\mathrm{ECil}} \oplus \Delta\mathring{\tau}^{\mathrm{C}}_{\mathrm{ECil}}}\left(\mathbf{p}^{\mathrm{E}}_{\mathrm{j}}\right)\right) - \mathbf{p}^{\mathrm{IMG}}_{\mathrm{ij}} \right\| = \sum_{\mathrm{j}} \left\| \mathbf{E}_{\mathrm{RPi,l+1,j}}\left(\Delta\mathring{\tau}^{\mathrm{C}}_{\mathrm{ECil}}\right) \right\| && (2) \\
\mathring{\zeta}_{\mathrm{ECi,l+1}} &\longleftarrow \mathring{\zeta}_{\mathrm{ECil}} \circ \mathrm{Exp}\left(\Delta\mathring{\tau}^{\mathrm{C}}_{\mathrm{ECil}}\right) = \mathring{\zeta}_{\mathrm{ECil}} \oplus \Delta\mathring{\tau}^{\mathrm{C}}_{\mathrm{ECil}} && (3)
\end{aligned}$$

Each $\Delta\mathring{\tau}^{\mathrm{C}}_{\mathrm{ECil}}$ represents the update to the camera pose $\mathring{\zeta}_{\mathrm{ECil}}$ viewed in the local camera frame $\mathrm{F_{Cil}}$, which is obtained by following the process described in [1,7], and results in (4), where $\mathbf{J}_{\mathrm{OF,ilj}}$ (5) is the optical flow for image $i$, iteration step $l$, and feature j obtained in Appendix A:

$$\begin{aligned}
\Delta\mathring{\tau}^{\mathrm{C}}_{\mathrm{ECil}} &= -\left[\sum_{\mathrm{j}} \mathbf{J}_{\mathrm{OF,ilj}}{}^{\mathrm{T}}\,\mathbf{J}_{\mathrm{OF,ilj}}\right]^{-1} \sum_{\mathrm{j}} \mathbf{J}_{\mathrm{OF,ilj}}{}^{\mathrm{T}} \left[\Pi\left(\mathbf{g}^{-1}_{\mathring{\zeta}_{\mathrm{ECil}}}\left(\mathbf{p}^{\mathrm{E}}_{\mathrm{j}}\right)\right) - \mathbf{p}^{\mathrm{IMG}}_{\mathrm{ij}}\right] \in \mathbb{R}^6 && (4) \\
\mathbf{J}_{\mathrm{OF,ilj}} &= \mathbf{J}_{\mathrm{OF}}\left(\Pi\left(\mathbf{g}^{-1}_{\mathring{\zeta}_{\mathrm{ECil}}}\left(\mathbf{p}^{\mathrm{E}}_{\mathrm{j}}\right)\right)\right) \in \mathbb{R}^{2\mathrm{x}6} && (5)
\end{aligned}$$

In order to protect the resulting pose from the possible presence of outliers in either the feature terrain 3D points $\mathbf{p}^{\mathrm{E}}_{\mathrm{j}}$ or their image projections $\mathbf{p}^{\mathrm{IMG}}_{\mathrm{ij}}$, it is better to replace the above squared error or mean estimator by a more robust M-estimator, such as the bisquare or Tukey estimator [8,9]. The error to be minimized in each iteration step is then given by (6), where the Tukey error function $\varrho_{\mathrm{TUK}}(\mathrm{x})$ can be found in [9].

$$\begin{aligned}
\mathrm{E_{RPi,l+1}}\left(\Delta\mathring{\tau}^{\mathrm{C}}_{\mathrm{ECil}}\right) &= \sum_{\mathrm{j}} \varrho_{\mathrm{TUK}}\left(\left[\Pi\left(\mathbf{g}^{-1}_{\mathring{\zeta}_{\mathrm{ECil}} \oplus \Delta\mathring{\tau}^{\mathrm{C}}_{\mathrm{ECil}}}\left(\mathbf{p}^{\mathrm{E}}_{\mathrm{j}}\right)\right) - \mathbf{p}^{\mathrm{IMG}}_{\mathrm{ij}}\right]^{\mathrm{T}} \left[\Pi\left(\mathbf{g}^{-1}_{\mathring{\zeta}_{\mathrm{ECil}} \oplus \Delta\mathring{\tau}^{\mathrm{C}}_{\mathrm{ECil}}}\left(\mathbf{p}^{\mathrm{E}}_{\mathrm{j}}\right)\right) - \mathbf{p}^{\mathrm{IMG}}_{\mathrm{ij}}\right]\right) \\
&= \sum_{\mathrm{j}} \varrho_{\mathrm{TUK}}\left(\mathbf{E}_{\mathrm{RPi,l+1,j}}{}^{\mathrm{T}}\,\mathbf{E}_{\mathrm{RPi,l+1,j}}\right) && (6)
\end{aligned}$$

A similar process to that employed above leads to the solution (7), where the Tukey weight function $\mathrm{w_{TUK}}(\mathrm{x})$ is also provided by [9]:

$$\begin{aligned}
\Delta\mathring{\tau}^{\mathrm{C}}_{\mathrm{ECil}} &= -\left[\sum_{\mathrm{j}} \mathrm{w_{TUK}}\left(\mathbf{E}_{\mathrm{RP,ilj}}{}^{\mathrm{T}}\mathbf{E}_{\mathrm{RP,ilj}}\right)\mathbf{J}_{\mathrm{OF,ilj}}{}^{\mathrm{T}}\mathbf{J}_{\mathrm{OF,ilj}}\right]^{-1} \\
&\quad \left[\sum_{\mathrm{j}} \mathrm{w_{TUK}}\left(\mathbf{E}_{\mathrm{RP,ilj}}{}^{\mathrm{T}}\mathbf{E}_{\mathrm{RP,ilj}}\right)\mathbf{J}_{\mathrm{OF,ilj}}{}^{\mathrm{T}}\mathbf{E}_{\mathrm{RP,ilj}}\right] \in \mathbb{R}^6 && (7) \\
\mathbf{E}_{\mathrm{RP,ilj}} &= \Pi\left(\mathbf{g}^{-1}_{\mathring{\zeta}_{\mathrm{ECil}}}\left(\mathbf{p}^{\mathrm{E}}_{\mathrm{j}}\right)\right) - \mathbf{p}^{\mathrm{IMG}}_{\mathrm{ij}} \in \mathbb{R}^2 && (8)
\end{aligned}$$

## 5. Proposed Pose Optimization within Visual Inertial Odometry

Lacking any absolute references, all visual odometry (VO) pipelines gradually accumulate errors in each of the six dimensions of the estimated ECEF to vehicle body pose $\mathring{\zeta}_{\mathrm{EB}}$. The resulting estimation error drift is described in Section 7 for the specific case of SVO, which is introduced in Appendix C, and whose pose optimization phase is described in Section 4.

This article proposes a method to improve the pose estimation capabilities of visual odometry pipelines by supplementing them with the outputs provided by an inertial navigation system. Taking the pose optimization algorithm of SVO (Section 4) as a baseline, this section describes the proposed improvements, while Section 7 explains the results obtained when applying the algorithms to two scenarios representative of GNSS-Denied navigation (Section 6).

If accurate estimations of attitude and altitude can be provided by an inertial navigation system (INS) such as that described in [6], these can be employed to ensure that the visual estimations for body attitude and vertical position ($\mathring{\mathbf{q}}_{\mathrm{NB}}$ and $\mathring{\mathrm{h}}$, part of the body pose $\mathring{\zeta}_{\mathrm{EB}}$) do not deviate in excess from their inertial counterparts $\hat{\mathbf{q}}_{\mathrm{NB}}$ and $\hat{\mathrm{h}}$, improving their accuracy. This process is depicted in Figure 3.

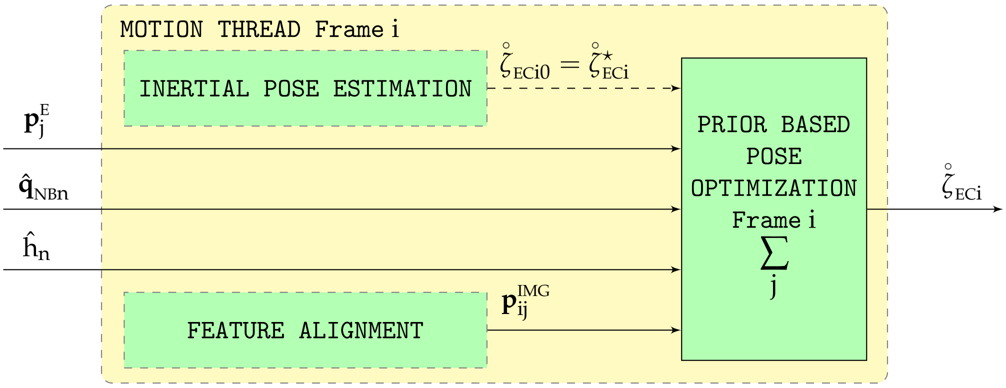
MOTION THREAD Frame i
INERTIAL POSE ESTIMATION
$\mathring{\zeta}_{\mathrm{ECi0}} = \mathring{\zeta}^{\star}_{\mathrm{ECi}}$
$\mathbf{p}_{\mathrm{j}}^{\mathrm{E}}$
$\hat{\mathbf{q}}_{\mathrm{NBn}}$
$\hat{\mathrm{h}}_{\mathrm{n}}$
FEATURE ALIGNMENT
$\mathbf{p}_{\mathrm{ij}}^{\mathrm{IMG}}$
PRIOR BASED POSE OPTIMIZATION Frame i
$\sum_{\mathrm{j}}$
$\mathring{\zeta}_{\mathrm{ECi}}$

**Figure 3.** Prior-based pose optimization flow diagram.

The inertial estimations ($\hat{\mathbf{q}}_{\mathrm{NB}}$, $\hat{\mathrm{h}}$) should not replace the visual ones ($\mathring{\mathbf{q}}_{\mathrm{NB}}$, $\mathring{\mathrm{h}}$) within SVO, as this would destabilize the visual pipeline preventing its convergence, but just act as anchors so the visual estimations oscillate freely as a result of the multiple SVO optimizations but without drifting from the vicinity of the anchors. This section shows how to modify the cost function within the iterative Gauss-Newton gradient descent pose optimization phase (Section 4) so it can take advantage of the inertial outputs. It is necessary to remark that, as indicated in Section 6, the inertial estimations (denoted by the subindex $n$) operate at a much higher rate than the visual ones (denoted by the subindex $i$).

### 5.1. Rationale for the Introduction of Priors

The prior based pose optimization process starts by executing exactly the same pose optimization described in Section 4, which seeks to obtain the ECEF to camera pose $\mathring{\zeta}_{\mathrm{ECi}}$ that minimizes the reprojection error $\mathrm{E}_{\mathrm{RPi}}$ (1). The iterative optimization results in a series of $\mathbb{SE}(3)$ tangent space updates $\Delta\mathring{\tau}^{\mathrm{C}}_{\mathrm{ECil}}$ (7), where $i$ identifies the image and $l$ indicates the iteration step. The camera pose is then advanced per (3) until the step diminution of the reprojection error falls below a certain threshold $\delta_{\mathrm{RP}}$.

The resulting ECEF to camera pose, $\mathring{\zeta}_{\mathrm{ECi}}$, is marked with the superindex $\star\star$ to indicate that it is the reprojection only solution, resulting in $\mathring{\zeta}^{\star\star}_{\mathrm{ECi}}$. Its concatenation with the constant body to camera pose $\zeta_{\mathrm{BC}}$ results in the reprojected ECEF to body pose $\mathring{\zeta}^{\star\star}_{\mathrm{EBi}}$ (note that a single asterisk superindex $< \cdot^{*} >$ applied to a pose refers to its conjugate or inverse, and that the concatenation $\circ$ and multiplication $\otimes$ operators are equivalent for $\mathbb{SE}(3)$ rigid body poses):

$$\mathring{\zeta}^{\star\star}_{\mathrm{EBi}} = \mathring{\zeta}^{\star\star}_{\mathrm{ECi}} \circ \zeta^{*}_{\mathrm{BC}} = \mathring{\zeta}^{\star\star}_{\mathrm{ECi}} \otimes \zeta^{*}_{\mathrm{BC}} = \mathring{\zeta}^{\star\star}_{\mathrm{ECi}} \otimes \zeta_{\mathrm{CB}} \tag{9}$$

The reprojected ECEF to body attitude $\mathring{\mathbf{q}}^{\star\star}_{\mathrm{EBi}}$ and Cartesian coordinates $\mathring{\mathbf{T}}^{\mathrm{B}\star\star}_{\mathrm{EBi}}$ can then be readily obtained from $\mathring{\zeta}^{\star\star}_{\mathrm{EBi}}$, which leads on one hand to the reprojected NED to body attitude $\mathring{\mathbf{q}}^{\star\star}_{\mathrm{NBi}}$, equivalent to the Euler angles $\mathring{\phi}^{\star\star}_{\mathrm{NBi}} = \left[\mathring{\psi}^{\star\star}_{\mathrm{i}}, \mathring{\theta}^{\star\star}_{\mathrm{i}}, \mathring{\xi}^{\star\star}_{\mathrm{i}}\right]^{\mathrm{T}}$ (yaw, pitch, and bank angles, respectively), and on the other to the geodetic coordinates $\mathring{\mathbf{T}}^{\mathrm{E,GDT}\star\star}_{\mathrm{i}} = \left[\mathring{\lambda}^{\star\star}_{\mathrm{i}}, \mathring{\varphi}^{\star\star}_{\mathrm{i}}, \mathring{\mathrm{h}}^{\star\star}_{\mathrm{i}}\right]^{\mathrm{T}}$ (longitude, latitude, and altitude) and ECEF to NED rotation $\mathring{\mathbf{q}}^{\star\star}_{\mathrm{ENi}}$.

Let us assume for the time being that the inertially estimated body attitude ($\hat{\mathbf{q}}_{\mathrm{NBn}}$) or altitude ($\hat{\mathrm{h}}_{\mathrm{n}}$) [6] enable the navigation system to conclude that it would be preferred if the visually optimized body attitude were closer to a certain target attitude identified by the superindex $\circ\circ$, $\mathring{\mathbf{q}}^{\circ\circ}_{\mathrm{NBi}}$, equivalent to the target Euler angles $\mathring{\phi}^{\circ\circ}_{\mathrm{NBi}} = \left[\mathring{\psi}^{\circ\circ}_{\mathrm{i}}, \mathring{\theta}^{\circ\circ}_{\mathrm{i}}, \mathring{\xi}^{\circ\circ}_{\mathrm{i}}\right]^{\mathrm{T}}$. Section 5.3 specifies when this assumption can be considered valid, as well as various alternatives to obtain the target attitude from $\hat{\mathbf{q}}_{\mathrm{NBn}}$ and $\hat{\mathrm{h}}_{\mathrm{n}}$. The target NED to body attitude $\mathring{\mathbf{q}}^{\circ\circ}_{\mathrm{NBi}}$ is converted into a target ECEF to camera attitude $\mathring{\mathbf{q}}^{\circ\circ}_{\mathrm{ECi}}$ by means of the constant body to camera rotation $\mathbf{q}_{\mathrm{BC}}$ and the original reprojected ECEF to NED rotation $\mathring{\mathbf{q}}^{\star\star}_{\mathrm{ENi}}$, incurring in a negligible error by not considering the attitude change of the NED frame as the iteration progresses. The concatenation $\circ$ and multiplication $\otimes$ operators are equivalent for $\mathbb{SO}(3)$ rigid body rotations:

$$\mathring{\mathbf{q}}^{\circ\circ}_{\mathrm{ECi}} = \mathring{\mathbf{q}}^{\star\star}_{\mathrm{ENi}} \circ \mathring{\mathbf{q}}^{\circ\circ}_{\mathrm{NBi}} \circ \mathbf{q}_{\mathrm{BC}} = \mathring{\mathbf{q}}^{\star\star}_{\mathrm{ENi}} \otimes \mathring{\mathbf{q}}^{\circ\circ}_{\mathrm{NBi}} \otimes \mathbf{q}_{\mathrm{BC}} \tag{10}$$

Note that the objective is not for the resulting body attitude $\mathring{\mathbf{q}}_{\mathrm{NBi}}$ to equal the target $\mathring{\mathbf{q}}^{\circ\circ}_{\mathrm{NBi}}$, but to balance both objectives (minimization of the reprojection error of the various terrain 3D points and minimization of the attitude differences with the targets) without imposing any hard constraints on the pose (position plus attitude) of the aircraft.

### 5.2. Prior-Based Pose Optimization

The *attitude adjustment error* $\mathrm{E}_{\mathrm{q,i}}$, a function of the estimated ECEF to camera attitude for image *i* ($\mathring{\mathbf{q}}_{\mathrm{ECi}}$), is defined in (11) as the norm of the Euclidean difference between rotation vectors corresponding to the estimated and target ECEF to camera attitudes ($\mathring{\mathbf{q}}_{\mathrm{ECi}}$, $\mathring{\mathbf{q}}^{\circ\circ}_{\mathrm{ECi}}$) [1,3]. Note that Log() refers to the $\mathbb{SO}(3)$ capitalized logarithmic function [1,3].

$$\mathrm{E}_{\mathrm{q,i}}\big(\mathrm{Log}(\mathring{\mathbf{q}}_{\mathrm{ECi}})\big) = \mathrm{E}_{\mathrm{q,i}}(\mathring{\mathbf{r}}_{\mathrm{ECi}}) = \left\|\mathrm{Log}(\mathring{\mathbf{q}}_{\mathrm{ECi}}) - \mathrm{Log}(\mathring{\mathbf{q}}^{\circ\circ}_{\mathrm{ECi}})\right\| = \left\|\mathring{\mathbf{r}}_{\mathrm{ECi}} - \mathring{\mathbf{r}}^{\circ\circ}_{\mathrm{ECi}}\right\| \tag{11}$$

Its minimization can be solved by means of an iterative Gauss-Newton gradient descent process [1,7]. Given an initial rotation vector (attitude) estimation $\mathring{\mathbf{r}}_{\mathrm{ECi},0} = \mathrm{Log}(\mathring{\mathbf{q}}_{\mathrm{ECi},0})$ taken from the initial pose $\mathring{\zeta}_{\mathrm{ECi0}} = \mathring{\zeta}^{\star}_{\mathrm{ECi}}$, each iteration step *l* minimizes (12) and advances the estimated solution by means of (13) until the step diminution of the attitude adjustment error falls below a given threshold $\delta_{\mathrm{q}}$ $\left(\mathrm{E}_{\mathrm{q,i,l}} - \mathrm{E}_{\mathrm{q,i,l+1}} < \delta_{\mathrm{q}}\right)$. Note that $\Delta\mathring{\mathbf{r}}^{\mathrm{C}}_{\mathrm{ECil}}$ represents the estimated tangent space incremental ECEF to camera attitude (rotation vector) viewed in the $\mathrm{F}_{\mathrm{C}}$ camera frame for image *i* and iteration *l*, $\oplus$ and $\circ$ represent the $\mathbb{SO}(3)$ plus and concatenation operators, and Exp() and Log() refer to the $\mathbb{SO}(3)$ capitalized exponential and logarithmic functions, respectively [1,3].

$$\begin{aligned} \mathrm{E}_{\mathrm{q,i,l+1}}\big(\Delta\mathring{\mathbf{r}}^{\mathrm{C}}_{\mathrm{ECil}}\big) \quad &= \quad \left\|\mathrm{Log}\big(\mathring{\mathbf{q}}_{\mathrm{ECil}} \oplus \Delta\mathring{\mathbf{r}}^{\mathrm{C}}_{\mathrm{ECil}}\big) - \mathrm{Log}\big(\mathring{\mathbf{q}}^{\circ\circ}_{\mathrm{ECi}}\big)\right\| \\ &= \quad \left\|\mathrm{Log}\big(\mathrm{Exp}(\mathring{\mathbf{r}}_{\mathrm{ECil}}) \oplus \Delta\mathring{\mathbf{r}}^{\mathrm{C}}_{\mathrm{ECil}}\big) - \mathring{\mathbf{r}}^{\circ\circ}_{\mathrm{ECi}}\right\| \qquad (12) \\ \mathring{\mathbf{q}}_{\mathrm{ECi,l+1}} \quad &\longleftarrow \quad \mathring{\mathbf{q}}_{\mathrm{ECil}} \circ \mathrm{Exp}\big(\Delta\mathring{\mathbf{r}}^{\mathrm{C}}_{\mathrm{ECil}}\big) = \mathring{\mathbf{q}}_{\mathrm{ECil}} \oplus \Delta\mathring{\mathbf{r}}^{\mathrm{C}}_{\mathrm{ECil}} \qquad (13) \end{aligned}$$

Each $\Delta\mathring{\mathbf{r}}_{\text{ECil}}^{\text{C}}$ represents the update to the camera attitude $\mathring{\mathbf{q}}_{\text{ECil}}$ given by the rotation vector viewed in the local camera frame $\text{F}_{\text{Cl}}$, which is obtained by following the process described in [1,7] (in this process the Jacobian coincides with the identity matrix because the map $\mathbf{f}(\mathring{\mathbf{r}}_{\text{ECi}}) = \mathring{\mathbf{r}}_{\text{ECi}}$ coincides with the rotation vector itself), and results in (14), where $\mathbf{J}_{\text{Ril}}$ (15) is the $\mathbb{SO}(3)$ right Jacobian $\mathbf{J}_{\text{R}}(\mathbf{r})$ for image $i$ and iteration step $l$ provided by [1,3]. These references also provide an expression for the right Jacobian inverse $\mathbf{J}_{\text{Ril}}^{-1}$. Note that while $\text{E}_{\text{q,i}}$ and $\text{E}_{\text{q,i,l+1}}$ present in (11) and (12) are both positive scalars, the adjustment error $\mathbf{E}_{\text{q,i,l}}$ that appears in (14) is an $\mathbb{R}^3$ vector.

$$\begin{aligned}
\Delta\mathring{\mathbf{r}}_{\text{ECil}}^{\text{C}} &= -\left[\mathbf{J}_{\text{Ril}}^{-\text{T}}\,\mathbf{J}_{\text{Ril}}^{-1}\right]^{-1}\mathbf{J}_{\text{Ril}}^{-\text{T}}\,\left[\mathring{\mathbf{r}}_{\text{ECil}} - \mathring{\mathbf{r}}_{\text{ECi}}^{\circ\circ}\right] \\
&= -\left[\mathbf{J}_{\text{Ril}}^{-\text{T}}\,\mathbf{J}_{\text{Ril}}^{-1}\right]^{-1}\mathbf{J}_{\text{Ril}}^{-\text{T}}\,\left[\text{Log}(\mathring{\mathbf{q}}_{\text{ECil}}) - \text{Log}(\mathring{\mathbf{q}}_{\text{ECi}}^{\circ\circ})\right] \\
&= -\left[\mathbf{J}_{\text{Ril}}^{-\text{T}}\,\mathbf{J}_{\text{Ril}}^{-1}\right]^{-1}\mathbf{J}_{\text{Ril}}^{-\text{T}}\,\mathbf{E}_{\text{q,il}} \in \mathbb{R}^3 && (14) \\
\mathbf{J}_{\text{Ril}} &= \mathbf{J}_{\text{R}}(\mathring{\mathbf{r}}_{\text{ECil}}) = \mathbf{J}_{\text{R}}\left(\text{Log}(\mathring{\mathbf{q}}_{\text{ECil}})\right) \quad \in \mathbb{R}^{3\times3} && (15)
\end{aligned}$$

The prior-based pose adjustment algorithm attempts to obtain the `ECEF` to camera pose $\mathring{\zeta}_{\text{ECi}}$ that minimizes the reprojection error $\text{E}_{\text{RPi}}$ discussed in Appendix C combined with the weighted attitude adjustment error $\text{E}_{\text{q,i}}$. The specific weight $\text{f}_{\text{q}}$ is discussed in Section 5.3. Inspired in [10], the main goal of the optimization algorithm is to minimize the reprojection error of the different terrain 3D points while simultaneously trying to be close to the attitude and altitude targets derived from the inertial filter.

$$\text{E}_{\text{POi}}\left(\mathring{\zeta}_{\text{ECi}}\right) = \text{E}_{\text{RPi}}\left(\mathring{\zeta}_{\text{ECi}}\right) + \text{f}_{\text{q}} \cdot \text{E}_{\text{q,i}}(\mathring{\mathbf{r}}_{\text{ECi}}) \tag{16}$$

Although the rotation vector $\mathring{\mathbf{r}}_{\text{ECi}} = \text{Log}(\mathring{\mathbf{q}}_{\text{ECi}})$ can be directly obtained from the pose $\mathring{\zeta}_{\text{ECi}}$ [1,3], merging the two algorithms requires a dimension change in the (15) Jacobian, as indicated by (17).

$$\mathbf{J}_{\text{RRil}}^{-1} = \left[\mathbf{O}_{3\times3}\ \mathbf{J}_{\text{Ril}}^{-1}\right] \quad \in \mathbb{R}^{3\times6} \tag{17}$$

The application of the iterative process described in [10] results in the following solution, which combines the contributions from the two different optimization targets:

$$\begin{aligned}
\mathbf{H}_{\text{PO,il}} &= \left[\sum_{\text{j}} \text{w}_{\text{TUK}}\left(\mathbf{E}_{\text{RP,ilj}}{}^{\text{T}}\mathbf{E}_{\text{RP,ilj}}\right)\mathbf{J}_{\text{OF,ilj}}{}^{\text{T}}\mathbf{J}_{\text{OF,ilj}}\right] + \text{f}_{\text{q}}^2 \cdot \left[\mathbf{J}_{\text{RRil}}^{-\text{T}}\,\mathbf{J}_{\text{RRil}}^{-1}\right] \quad \in \mathbb{R}^{6\times6} && (18) \\
\Delta\mathring{\tau}_{\text{ECil}}^{\text{C}} &= -\mathbf{H}_{\text{PO,il}}^{-1}\left[\left[\sum_{\text{j}} \text{w}_{\text{TUK}}\left(\mathbf{E}_{\text{RP,ilj}}{}^{\text{T}}\mathbf{E}_{\text{RP,ilj}}\right)\mathbf{J}_{\text{OF,ilj}}{}^{\text{T}}\mathbf{E}_{\text{RP,ilj}}\right] + \text{f}_{\text{q}} \cdot \mathbf{J}_{\text{RRil}}^{-\text{T}}\,\mathbf{E}_{\text{q,il}}\right] && (19) \\
\mathring{\zeta}_{\text{ECi,l+1}} &\longleftarrow \mathring{\zeta}_{\text{ECil}} \circ \text{Exp}(\Delta\mathring{\tau}_{\text{ECil}}^{\text{C}}) = \mathring{\zeta}_{\text{ECil}} \oplus \Delta\mathring{\tau}_{\text{ECil}}^{\text{C}} && (20)
\end{aligned}$$

### *5.3. PI Control-Inspired Pose Adjustment Activation*

Sections 5.1 and 5.2 describe the attitude adjustment and its fusion with the default reprojection error minimization pose optimization algorithm, but they do not specify the conditions under which the adjustment is activated, how the $\mathring{\mathbf{q}}_{\text{NBi}}^{\circ\circ} \equiv \mathring{\phi}_{\text{NBi}}^{\circ\circ}$ target is determined, or the obtainment of its $\text{f}_{\text{q}}$ relative weight when applying the (16) joint optimization. These parameters are determined below in three different cases: an adjustment in which only pitch is controlled, an adjustment in which both pitch and bank angles are controlled, and a complete attitude adjustment.

#### 5.3.1. Pitch Adjustment Activation

The attitude adjustment described in (11) through (15) can be converted into a pitch only ($\theta$) adjustment by forcing the yaw ($\psi$) and bank ($\xi$) angle targets to coincide in each optimization *i* with the outputs of the reprojection only optimization. The target geodetic coordinates ($\mathbf{T}_{\mathrm{i}}^{\mathrm{E,GDT}}$) also coincide with the ones resulting from the reprojection only optimization.

$$\overset{\circ}{\psi}_{\mathrm{i}}^{\circ\circ} = \overset{\circ}{\psi}_{\mathrm{i}}^{\star\star} \tag{21}$$

$$\overset{\circ}{\theta}_{\mathrm{i}}^{\circ\circ} = \overset{\circ}{\theta}_{\mathrm{i}}^{\star\star} + \Delta\overset{\circ}{\theta}_{\mathrm{i}}^{\circ\circ} \tag{22}$$

$$\overset{\circ}{\xi}_{\mathrm{i}}^{\circ\circ} = \overset{\circ}{\xi}_{\mathrm{i}}^{\star\star} \tag{23}$$

$$\overset{\circ}{\mathbf{T}}_{\mathrm{i}}^{\mathrm{E,GDT}\circ\circ} = \overset{\circ}{\mathbf{T}}_{\mathrm{i}}^{\mathrm{E,GDT}\star\star} = \left[\overset{\circ}{\lambda}_{\mathrm{i}}^{\star\star}, \overset{\circ}{\varphi}_{\mathrm{i}}^{\star\star}, \overset{\circ}{\mathrm{h}}_{\mathrm{i}}^{\star\star}\right]^{\mathrm{T}} \tag{24}$$

When activated as explained below, the new ECEF to body pose target $\overset{\circ}{\zeta}_{\mathrm{EBi}}^{\circ\circ}$ only differs in one out of six dimensions (the pitch) from the reprojection only $\mathrm{E}_{\mathrm{RPi}}$ optimum pose $\overset{\circ}{\zeta}_{\mathrm{EBi}}^{\star\star}$, and the difference is very small as its effects are intended to accumulate over many successive images. This does not mean however that the other five components do not vary, as the joint optimization process described in (16) through (20) freely optimizes within $\mathbb{SE}(3)$ with six degrees of freedom to minimize the joint cost function $\mathrm{E}_{\mathrm{POi}}$ that not only considers the reprojection error, but also the resulting pitch target.

The pitch adjustment aims for the visual estimations for altitude $\overset{\circ}{\mathrm{h}}_{\mathrm{i}}$ and pitch $\overset{\circ}{\theta}_{\mathrm{i}}$ (in this order) not to deviate in excess from their inertially estimated counterparts $\hat{\mathrm{h}}_{\mathrm{n}}$ and $\hat{\theta}_{\mathrm{n}}$. It is inspired in a *proportional integral* (PI) control scheme [11–14] in which the geometric altitude *adjustment error* $\Delta\mathrm{h} = \overset{\circ}{\mathrm{h}}_{\mathrm{i}} - \hat{\mathrm{h}}_{\mathrm{n}}$ can be considered as the integral of the pitch adjustment error $\Delta\theta = \overset{\circ}{\theta}_{\mathrm{i}} - \hat{\theta}_{\mathrm{n}}$ in the sense that any difference between adjusted pitch angles (the P control) slowly accumulate over time generating differences in adjusted altitude (the I control). In this context, *adjustment error* is understood as the difference between the visual and inertial estimations. In addition, the adjustment also depends on the rate of climb (ROC) adjustment error (to avoid noise, this is smoothed over the last 100 images or 10 s) $\Delta\mathrm{ROC} = \overset{\circ}{\mathrm{ROC}}_{\mathrm{i}} - \hat{\mathrm{ROC}}_{\mathrm{n}}$, which can be considered a second P control as ROC is the time derivative of the pressure altitude.

Note that the objective is not for the visual estimations to closely track the inertial ones, but only to avoid excessive deviations, so there exist lower thresholds $\Delta\mathrm{h}_{\mathrm{LOW}}$, $\Delta\theta_{\mathrm{LOW}}$, and $\Delta\mathrm{ROC}_{\mathrm{LOW}}$ below which the adjustments are not activated. These thresholds are arbitrary but have been set taking into account the inertial navigation system (INS) accuracy and its sources of error, as described in [6]. If the absolute value of a certain adjustment error (difference between the visual and estimated states) is above its threshold, the visual inertial system can conclude with a high degree of confidence that the adjustment procedure can be applied; if below the threshold, the adjustment should not be employed as there is a significant risk that the true visual error (difference between the visual and actual states) may have the opposite sign, in which case the adjustment would be counterproductive.

As an example, let us consider a case in which the visual altitude $\overset{\circ}{\mathrm{h}}_{\mathrm{i}}$ is significantly higher than the inertial one $\hat{\mathrm{h}}_{\mathrm{n}}$, resulting in $|\Delta\mathrm{h}| > \Delta\mathrm{h}_{\mathrm{LOW}}$; in this case the system concludes that the aircraft is “high” and applies a negative pitch adjustment to slowly decrease the body pitch visual estimation $\overset{\circ}{\theta}$ over many images, with these accumulating over time into a lower altitude $\overset{\circ}{\mathrm{h}}$ that what would be the case if no adjustment were applied. On the other hand, if the absolute value of the adjustment error is below the threshold ($|\Delta\mathrm{h}| < \Delta\mathrm{h}_{\mathrm{LOW}}$), the adjustment should not be applied as there exists a significant risk that the aircraft is in fact “low” instead of “high” (when compared with the true altitude $\mathrm{h}_{\mathrm{t}}$, not the the inertial one $\hat{\mathrm{h}}_{\mathrm{n}}$), and a negative pitch adjustment would only exacerbate the situation. A similar reasoning applies for the adjustment pitch error, in which the visual inertial system reacts or not to correct perceived “nose-up” or “nose-down” visual estimations. The applied thresholds are displayed in Table 2.

**Table 2.** Pitch and bank adjustment settings.

| Variable | Value | Unit | Variable | Value | Unit |
|---|---|---|---|---|---|
| $\Delta \mathrm{h}_{\mathrm{LOW}}$ | 25.0 | m | $\Delta \mathring{\theta}_{1,\mathrm{MAX}}^{\circ\circ}$ | 0.0005 | ° |
| $\Delta \theta_{\mathrm{LOW}}$ | 0.2 | ° | $\Delta \mathring{\theta}_{2,\mathrm{MAX}}^{\circ\circ}$ | 0.0003 | ° |
| $\Delta \mathrm{ROC}_{\mathrm{LOW}}$ | 0.01 | m/s | $\Delta \mathring{\xi}_{1,\mathrm{MAX}}^{\circ\circ}$ | 0.0003 | ° |
| $\Delta \xi_{\mathrm{LOW}}$ | 0.2 | ° | | | |

The $\mathring{\theta}_{\mathrm{i}}^{\circ\circ}$ pitch target to be applied for each image is given by (22), where the obtainment of the pitch adjustment $\Delta\mathring{\theta}_{\mathrm{i}}^{\circ\circ}$ is explained below based on its three components (25):

$$\Delta\mathring{\theta}_{\mathrm{i}}^{\circ\circ} = \Delta\mathring{\theta}_{\mathrm{h}}^{\circ\circ} + \Delta\mathring{\theta}_{\theta}^{\circ\circ} + \Delta\mathring{\theta}_{\mathrm{ROC}}^{\circ\circ} \tag{25}$$

- The pitch adjustment due to altitude, $\Delta\mathring{\theta}_{\mathrm{h}}^{\circ\circ}$, linearly varies between zero when the adjustment error is below the threshold $\Delta \mathrm{h}_{\mathrm{LOW}}$ to $\Delta\mathring{\theta}_{1,\mathrm{MAX}}^{\circ\circ}$ when the error is twice the threshold, as shown in (26). The adjustment is bounded at this value to avoid destabilizing SVO with pose adjustments that differ too much from their reprojection only optimum $\mathring{\zeta}_{\mathrm{EBi}}^{\star\star}$ (9).

$$\Delta\mathring{\theta}_{\mathrm{h}}^{\circ\circ} = \begin{cases} 0 & \text{when } |\Delta \mathrm{h}| < \Delta \mathrm{h}_{\mathrm{LOW}} \\ -\,\mathrm{sign}(\Delta \mathrm{h})\, \Delta\mathring{\theta}_{1,\mathrm{MAX}}^{\circ\circ}\, \big(|\Delta \mathrm{h}| - \Delta \mathrm{h}_{\mathrm{LOW}}\big)/\Delta \mathrm{h}_{\mathrm{LOW}} & \Delta \mathrm{h}_{\mathrm{LOW}} \leq |\Delta \mathrm{h}| \leq 2 \cdot \Delta \mathrm{h}_{\mathrm{LOW}} \\ -\,\mathrm{sign}(\Delta \mathrm{h})\, \Delta\mathring{\theta}_{1,\mathrm{MAX}}^{\circ\circ} & \text{when } |\Delta \mathrm{h}| > 2 \cdot \Delta \mathrm{h}_{\mathrm{LOW}} \end{cases} \tag{26}$$

- The pitch adjustment due to pitch, $\Delta\mathring{\theta}_{\theta}^{\circ\circ}$, works similarly but employing $\Delta\theta$ instead of $\Delta \mathrm{h}$ and $\Delta\theta_{\mathrm{LOW}}$ instead of $\Delta \mathrm{h}_{\mathrm{LOW}}$, while also relying on the same limit $\Delta\mathring{\theta}_{1,\mathrm{MAX}}^{\circ\circ}$. In addition, $\Delta\mathring{\theta}_{\theta}^{\circ\circ}$ is set to zero if its sign differs from that of $\Delta\mathring{\theta}_{\mathrm{h}}^{\circ\circ}$, and reduced so the combined effect of both targets does not exceed the limit ($|\Delta\mathring{\theta}_{\mathrm{h}}^{\circ\circ} + \Delta\mathring{\theta}_{\theta}^{\circ\circ}| \leq \Delta\mathring{\theta}_{1,\mathrm{MAX}}^{\circ\circ}$).
- The pitch adjustment due to rate of climb, $\Delta\mathring{\theta}_{\mathrm{ROC}}^{\circ\circ}$, also follows a similar scheme but employing $\Delta\mathrm{ROC}$ instead of $\Delta \mathrm{h}$, $\Delta\mathrm{ROC}_{\mathrm{LOW}}$ instead of $\Delta \mathrm{h}_{\mathrm{LOW}}$, and $\Delta\mathring{\theta}_{2,\mathrm{MAX}}^{\circ\circ}$ instead of $\Delta\mathring{\theta}_{1,\mathrm{MAX}}^{\circ\circ}$. Additionally, it is multiplied by the ratio between $\Delta\mathring{\theta}_{\mathrm{h}}^{\circ\circ}$ and $\Delta\mathring{\theta}_{1,\mathrm{MAX}}^{\circ\circ}$ to limit its effects when the altitude estimated error $\Delta \mathrm{h}$ is small. This adjustment can act in both directions, imposing bigger pitch adjustments if the altitude error is increasing or lower one if it is already diminishing.

If activated, the weight value $\mathrm{f_q}$ required for the (16) joint optimization is determined by imposing that the weighted attitude error $\mathrm{f_q} \cdot \mathrm{E_{q,i}}\big(\mathring{\mathbf{r}}_{\mathrm{ECi0}}\big)$ coincides with the reprojection error $\mathrm{E_{RPi}}\big(\mathring{\zeta}_{\mathrm{ECi0}}\big)$ when evaluated before the first iteration, this is, it assigns the same weight to the two active components of the joint $\mathrm{E_{POi}}\big(\mathring{\zeta}_{\mathrm{ECi}}\big)$ cost function (16).

### 5.3.2. Pitch and Bank Adjustment Activation

The previous scheme can be modified to also make use of the inertially estimated body bank angle $\hat{\xi}_{\mathrm{n}}$ within the framework established by the (11) through (15) attitude adjustment optimization:

$$\mathring{\psi}_{\mathrm{i}}^{\circ\circ} = \mathring{\psi}_{\mathrm{i}}^{\star\star} \tag{27}$$

$$\mathring{\theta}_{\mathrm{i}}^{\circ\circ} = \mathring{\theta}_{\mathrm{i}}^{\star\star} + \Delta\mathring{\theta}_{\mathrm{i}}^{\circ\circ} \tag{28}$$

$$\mathring{\xi}_{\mathrm{i}}^{\circ\circ} = \mathring{\xi}_{\mathrm{i}}^{\star\star} + \Delta\mathring{\xi}_{\mathrm{i}}^{\circ\circ} \tag{29}$$

$$\mathring{\mathbf{T}}_{\mathrm{i}}^{\mathrm{E,GDT}\circ\circ} = \mathring{\mathbf{T}}_{\mathrm{i}}^{\mathrm{E,GDT}\star\star} = \left[\mathring{\lambda}_{\mathrm{i}}^{\star\star}, \mathring{\varphi}_{\mathrm{i}}^{\star\star}, \mathring{\mathrm{h}}_{\mathrm{i}}^{\star\star}\right]^{\mathrm{T}} \tag{30}$$

Although the new body pose target $\mathring{\zeta}_{\rm EBi}^{\circ\circ}$ only differs in two out of six dimensions (pitch and bank) from the optimum pose $\mathring{\zeta}_{\rm EBi}^{\star\star}$ obtained by minimizing the reprojection error exclusively, all six degrees of freedom are allowed to vary when minimizing the joint cost function.

The determination of the pitch adjustment $\Delta\mathring{\theta}_{\rm i}^{\circ\circ}$ does not vary with respect to (25), and that of the bank adjustment $\Delta\mathring{\xi}_{\rm i}^{\circ\circ}$ relies on a linear adjustment between two values similar to any of the three components of (25), but relying on the bank angle adjustment error $\Delta\xi = \mathring{\xi}_{\rm i} - \hat{\xi}_{\rm n}$, as well as a $\Delta\xi_{\rm LOW}$ threshold and $\Delta\mathring{\xi}_{\rm 1,MAX}^{\circ\circ}$ maximum adjustment whose values are provided in Table 2. Note that the value of the $\Delta\xi_{\rm LOW}$ threshold coincides with that of $\Delta\theta_{\rm LOW}$ as the INS accuracy for both pitch and roll is similar according to [6].

It is important to remark that the combined pitch and bank adjustment activation is the one employed to generate the results described in Sections 7 and 8.

#### 5.3.3. Attitude Adjustment Activation

The use of the inertially estimated yaw angle $\hat{\psi}$ is not recommended as the visual estimation $\mathring{\psi}$ (without any inertial inputs) is, in general, more accurate than its inertial counterpart $\hat{\psi}$, as discussed in Section 7. This can be traced on one side to the bigger influence that a yaw change has on the resulting optical flow when compared with those caused by pitch and bank changes, which makes the body yaw angle easier to track by visual systems when compared to the pitch and bank angles, and on the other to the inertial system relying on the gravity pointing down to control pitch and bank adjustments versus the less robust dependence on the Earth magnetic field and associated magnetometer readings used to estimate the aircraft heading [6].

For this reason, the attitude adjustment process described next has not been implemented, although it is included here as a suggestion for other applications in which the objective may be to adjust the vehicle attitude as a whole. The process relies on the inertially estimated attitude $\hat{\mathbf{q}}_{\rm NBn}$ and the initial estimation $\mathring{\mathbf{q}}_{\rm NBi}^{\star\star}$ provided by the reprojection only pose optimization process. Its difference is given by $\Delta\mathring{\mathbf{r}}^{\rm Bi,\star\star} = \hat{\mathbf{q}}_{\rm NBn} \ominus \mathring{\mathbf{q}}_{\rm NBi}^{\star\star}$, where $\ominus$ represents the $\mathbb{SO}(3)$ minus operator and the superindex "`Bi`" indicates that it is viewed in the pose optimized body frame. This perturbation can be decoupled into a rotating direction and an angular displacement [1,3], resulting in $\Delta\mathring{\mathbf{r}}^{\rm Bi,\star\star} = \mathring{\mathbf{n}}^{\rm Bi,\star\star}\,\Delta\mathring{\phi}^{\star\star}$.

Let us now consider that the visual inertial system decides to set an attitude target that differs by $\Delta\mathring{\phi}^{\circ\circ}$ from its reprojection only solution $\mathring{\mathbf{q}}_{\rm NBi}^{\star\star}$, but rotating about the axis that leads towards its inertial estimation $\hat{\mathbf{q}}_{\rm NBn}$. The target attitude $\mathring{\mathbf{q}}_{\rm NBi}^{\circ\circ}$ can then be obtained by $\mathbb{SO}(3)$ Spherical Linear Interpolation (SLERP) [1,2], where $\rm t = \Delta\mathring{\phi}^{\circ\circ} \;/\; \Delta\mathring{\phi}^{\star\star}$ is the ratio between the target rotation and the attitude error or estimated angular displacement:

$$\mathring{\mathbf{q}}_{\rm NBi}^{\circ\circ} = \mathring{\mathbf{q}}_{\rm NBi}^{\star\star} \otimes \left( {\mathring{\mathbf{q}}_{\rm NBi}^{\star\star}}^{*} \otimes \hat{\mathbf{q}}_{\rm NBn} \right)^{\rm t} \tag{31}$$

### 5.4. Additional Modifications to SVO

In addition to the PI-inspired introduction of priors into the pose optimization phase, the availability of inertial estimations enable other minor modifications to the original SVO pipeline described in Appendix C. These include the addition of the current features to the structure optimization phase (so the pose adjustments introduced by the prior based pose optimization are not reverted), the replacement of the sparse image alignment phase by an inertial estimation of the $\mathring{\zeta}_{\rm ECi0} = \mathring{\zeta}_{\rm ECi}^{\star}$ input to the pose optimization process, and the use of the GNSS-based inertial distance estimations to obtain more accurate height and path angle values for the SVO initialization.

## 6. Testing: High-Fidelity Simulation and Scenarios

To evaluate the performance of the proposed visual navigation algorithms, this article relies on Monte Carlo simulations consisting of 100 runs each of two different scenarios based on the high fidelity stochastic flight simulator graphically depicted in Figure 4. Described in detail in [15] and with its open source C++ implementation available in [16], the simulator models the flight in varying weather and turbulent conditions of a fixed wing piston engine autonomous UAV.

The simulator consists of two distinct processes. The first, represented by the yellow blocks on the right of Figure 4, models the physics of flight and the interaction between the aircraft and its surroundings that results in the real aircraft trajectory $\mathbf{x} = \mathbf{x}_{\text{TRUTH}}$; the second, represented by the green blocks on the left, contains the aircraft systems in charge of ensuring that the resulting trajectory adheres as much as possible to the mission objectives. It includes the different sensors whose output comprise the sensed trajectory $\widetilde{\mathbf{x}} = \mathbf{x}_{\text{SENSED}}$, the navigation system in charge of filtering it to obtain the estimated trajectory $\hat{\mathbf{x}} = \mathbf{x}_{\text{EST}}$, the guidance system that converts the reference objectives $\mathbf{x}_{\text{REF}}$ into the control targets $\delta_{\text{TARGET}}$, and the control system that adjusts the position of the throttle and aerodynamic control surfaces $\delta_{\text{CNTR}}$ so the estimated trajectory $\hat{\mathbf{x}} = \mathbf{x}_{\text{EST}}$ is as close as possible to the reference objectives $\mathbf{x}_{\text{REF}}$. Table 3 provides the working frequencies employed for the different trajectories shown in Figures 4–7.

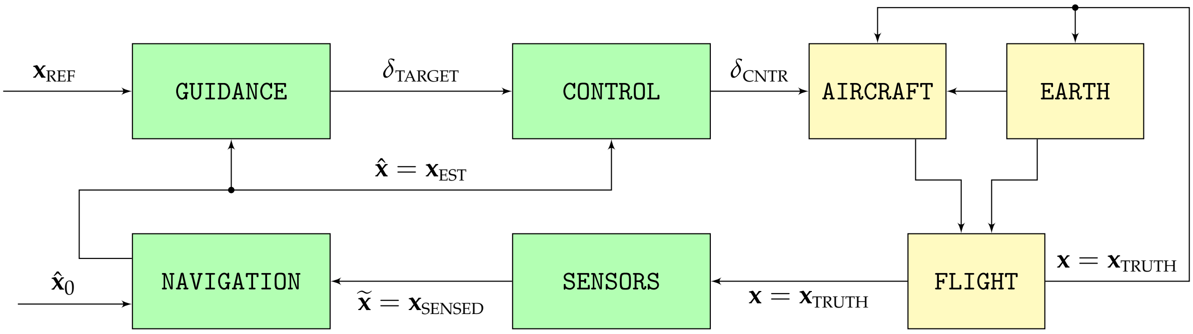
$\mathbf{x}_{\text{REF}}$
GUIDANCE
$\delta_{\text{TARGET}}$
CONTROL
$\delta_{\text{CNTR}}$
AIRCRAFT
EARTH
$\hat{\mathbf{x}} = \mathbf{x}_{\text{EST}}$
$\hat{\mathbf{x}}_0$
NAVIGATION
$\widetilde{\mathbf{x}} = \mathbf{x}_{\text{SENSED}}$
SENSORS
$\mathbf{x} = \mathbf{x}_{\text{TRUTH}}$
FLIGHT
$\mathbf{x} = \mathbf{x}_{\text{TRUTH}}$

**Figure 4.** Components of the high-fidelity simulation.

All components of the flight simulator have been modeled with as few simplifications as possible to increase the realism of the results, as explained in [15,17]. With the exception of the aircraft performances and its control system, which are deterministic, all other simulator components are treated as stochastic and hence vary from one execution to the next, enhancing the significance of the Monte Carlo simulation results.

**Table 3.** Working frequencies of the different systems and trajectory representations.

| Discrete Time | Frequency | Period | Variables | Systems |
|---|---|---|---|---|
| $t_t = t \cdot \Delta t_{\text{TRUTH}}$ | 500 Hz | 0.002 s | $\mathbf{x} = \mathbf{x}_{\text{TRUTH}}$ | Flight physics |
| $t_s = s \cdot \Delta t_{\text{SENSED}}$ | 100 Hz | 0.01 s | $\widetilde{\mathbf{x}} = \mathbf{x}_{\text{SENSED}}$ | Sensors |
| $t_n = n \cdot \Delta t_{\text{EST}}$ | 100 Hz | 0.01 s | $\hat{\mathbf{x}} = \mathbf{x}_{\text{EST}}$ | Inertial navigation |
| $t_c = c \cdot \Delta t_{\text{CNTR}}$ | 50 Hz | 0.02 s | $\delta_{\text{TARGET}}$, $\delta_{\text{CNTR}}$ | Guidance and control |
| $t_i = i \cdot \Delta t_{\text{IMG}}$ | 10 Hz | 0.1 s | $\mathring{\mathbf{x}} = \mathbf{x}_{\text{IMG}}$ | Visual navigation and camera |

### *6.1. Camera*

The flight simulator has the capability, when provided with the camera pose (the camera is positioned facing down and rigidly attached to the aircraft structure) with respect to the Earth at equally spaced time intervals, of generating images that resemble the view of the Earth surface that the camera would record if located at that particular pose. To do so, it relies on the `Earth Viewer` library, a modification to `osgEarth` [18] (which, in turn, relies on `OpenSceneGraph` [19]) capable of generating realistic Earth images as long as the camera height over the terrain is significantly higher than the vertical relief present in the image. A more detailed explanation of the image generation process is provided in [17].

It is assumed that the shutter speed is sufficiently high that all images are equally sharp, and that the image generation process is instantaneous. In addition, the camera ISO setting remains constant during the flight, and all generated images are noise free. The simulation also assumes that the visible spectrum radiation reaching all patches of the Earth surface remains constant, and the terrain is considered Lambertian [20], so its appearance at any given time does not vary with the viewing direction. The combined use of these assumptions implies that a given terrain object is represented with the same luminosity in all images, even as its relative pose (position and attitude) with respect to the camera varies. Geometrically, the simulation adopts a perspective projection or pinhole camera model [20], which, in addition, is perfectly calibrated and hence shows no distortion. The camera has a focal length of 19 mm and a sensor with 768 by 1024 pixels.

### *6.2. Scenarios*

Most visual inertial odometry (VIO) packages discussed in Appendix B include in their release articles an evaluation when applied to the EuRoC Micro Air Vehicle (MAV) datasets [21], and so do independent articles, such as [22]. These datasets contain perfectly synchronized stereo images, Inertial Measurement Unit (IMU) measurements, and ground truth readings obtained with a laser, for 11 different indoor trajectories flown with a MAV, each with a duration in the order of two minutes and a total distance in the order of 100 m. This fact by itself indicates that the target application of exiting VIO implementations differs significantly from the main focus of this article, which is the long term flight of a fixed wing UAV in GNSS-Denied conditions, as there may exist accumulating errors that are completely non discernible after such short periods of time, but that grow non-linearly and have the capability of inducing significant pose errors when the aircraft remains aloft for long periods of time.

The algorithms introduced in this article are hence tested through simulation under two different scenarios designed to analyze the consequences of losing the GNSS signals for long periods of time. Although a short summary is included below, detailed descriptions of the mission, weather, and wind field employed in each scenario can be found in [15]. Most parameters comprising the scenario are defined stochastically, resulting in different values for every execution. Note that all results shown in Sections 7 and 8 are based on Monte Carlo simulations comprising 100 runs of each scenario, testing the sensitivity of the proposed navigation algorithms to a wide variety of values in the parameters.

- Scenario #1 has been defined with the objective of adequately representing the challenges faced by an autonomous fixed wing UAV that suddenly cannot rely on GNSS and hence changes course to reach a predefined recovery location situated at approximately one hour of flight time. In the process, in addition to executing an altitude and airspeed adjustment, the autonomous aircraft faces significant weather and wind field changes that make its GNSS-Denied navigation even more challenging.
  With respect to the mission, the stochastic parameters include the initial airspeed, pressure altitude, and bearing ($v_{TAS,INI}$, $H_{P,INI}$, $\varnothing_{INI}$), their final values ($v_{TAS,END}$, $H_{P,END}$, $\varnothing_{END}$), and the time at which each of the three maneuvers is initiated (turns are executed with a bank angle of $\xi_{TURN} = \pm 10^{\circ}$, altitude changes employ an aerodynamic path angle of $\gamma_{TAS,CLIMB} = \pm 2^{\circ}$, and airspeed modifications are automatically executed by the control system as set-point changes). The scenario lasts for $t_{END} = 3800$ s, while the GNSS signals are lost at $t_{GNSS} = 100$ s.
  The wind field is also defined stochastically, as its two parameters (speed and bearing) are constant both at the beginning ($v_{WIND,INI}$, $\varnothing_{WIND,INI}$) and conclusion ($v_{WIND,END}$, $\varnothing_{WIND,END}$) of the scenario, with a linear transition in between. The specific times at which the wind change starts and concludes also vary stochastically among the different simulation runs. As described in [15], the turbulence remains strong throughout the whole scenario, but its specific values also vary stochastically from one execution to the next.
  A similar linear transition occurs with the temperature and pressure offsets that define the atmospheric properties [23], as they are constant both at the start ($\Delta T_{INI}$, $\Delta p_{INI}$) and end ($\Delta T_{END}$, $\Delta p_{END}$) of the flight. In contrast with the wind field, the specific times at

which the two transitions start and conclude are not only stochastic but also different from each other.

- Scenario #2 represents the challenges involved in continuing with the original mission upon the loss of the GNSS signals, executing a series of continuous turn maneuvers over a relatively short period of time with no atmospheric or wind variations. As in scenario #1, the GNSS signals are lost at $t_{GNSS} = 100\,s$, but the scenario duration is shorter ($t_{END} = 500\,s$). The initial airspeed and pressure altitude ($v_{TAS,INI}, H_{P,INI}$) are defined stochastically and do not change throughout the whole scenario; the bearing however changes a total of eight times between its initial and final values, with all intermediate bearing values, as well as the time for each turn varying stochastically from one execution to the next. Although the same turbulence is employed as in scenario #1, the wind and atmospheric parameters ($v_{WIND,INI}, \varnothing_{WIND,INI}, \Delta T_{INI}, \Delta p_{INI}$) remain constant throughout scenario #2.

## 7. Results: Navigation System Error in GNSS-Denied Conditions

This section presents the results obtained with the proposed Inertially Assisted Visual Navigation System or IA-VNS (comprised by SVO, as described in Appendix C and Section 4, together with the proposed modifications described in Section 5) when executing Monte Carlo simulations of the two GNSS-Denied scenarios over the MX terrain type (Section 8 defines various terrain types, and then analyzes their influence on the simulation results), each consisting of 100 executions. They are compared with the results obtained with the standalone Visual Navigation System or VNS that relies on the baseline SVO pipeline (Appendix C and Section 4), and with those of the Inertial Navigation System or INS described in [6].

Tables 4–6 contain the *navigation system error* or NSE (difference between the real or true states $\mathbf{x}$ and their inertial $\hat{\mathbf{x}}$ or visual $\mathring{\mathbf{x}}$ estimations) incurred by the various navigation systems (and accordingly denoted as INSE, VNSE, and IA-VNSE) at the conclusion of the two GNSS-Denied scenarios, represented by the mean, standard deviation, and maximum value of the estimation errors. In addition, the figures shown in this section depict the variation with time of the NSE mean (solid line) and standard deviation (dashed lines) for the 100 executions. The following remarks are necessary:

- The results obtained with the INS under the same two GNSS-Denied scenarios are described in detail in [6], a previous article by the same authors. It proves that it is possible to take advantage of sensors already present onboard fixed wing aircraft (accelerometers, gyroscopes, magnetometers, Pitot tube, air vanes, thermometer, and barometer), the particularities of fixed wing flight, and the atmospheric and wind estimations that can be obtained before the GNSS signals are lost, to develop an EKF (Extended Kalman Filter)-based INS that results in bounded (no drift) estimations for attitude (ensuring that the aircraft can remain aloft in GNSS-Denied conditions for as long as there is fuel available), altitude (the estimation error depends on the change in atmospheric pressure offset $\Delta p$ [23] from its value at the time the GNSS signals are lost, which is bounded by atmospheric physics), and ground velocity (the estimation error depends on the change in wind velocity from its value at the time the GNSS signals are lost, which is bounded by atmospheric physics), as well as an unavoidable drift in horizontal position caused by integrating the ground velocity without absolute observations. Note that of the six $\mathbb{SE}(3)$ degrees of freedom or the aircraft pose (three for attitude, two for horizontal position, one for altitude), the INS is hence capable of successfully estimating four of them in GNSS-Denied conditions. Figure 5 graphically depicts that the INS inputs include all sensor measurements $\widetilde{\mathbf{x}} = \mathbf{x}_{SENSED}$ with the exception of the camera images $\mathbf{I}$.

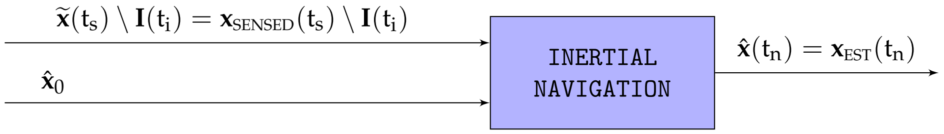
$\widetilde{\mathbf{x}}(t_s) \setminus \mathbf{I}(t_i) = \mathbf{x}_{SENSED}(t_s) \setminus \mathbf{I}(t_i)$
$\hat{\mathbf{x}}_0$
INERTIAL NAVIGATION
$\hat{\mathbf{x}}(t_n) = \mathbf{x}_{EST}(t_n)$

**Figure 5.** INS flow diagram.

- Visual navigation systems (either VNS or IA-VNS) are only necessary to reduce the estimation error in the two remaining degrees of freedom (the horizontal position). Although both of them estimate the complete six dimensional aircraft pose, their attitude and altitude estimations shall only be understood as a means to provide an accurate horizontal position estimation, which represents their sole objective. Figure 6 shows that the VNS relies exclusively on the images **I** without the use of any other sensors; on the other hand, the IA-VNS represented in Figure 7 complements the images with the $\hat{\mathbf{x}} = \mathbf{x}_{EST}$ outputs of the INS.

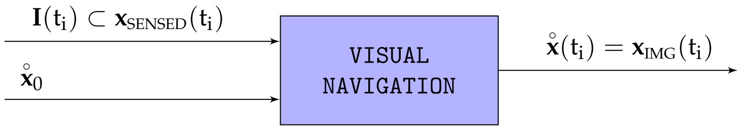
$\mathbf{I}(t_i) \subset \mathbf{x}_{SENSED}(t_i)$
$\mathring{\mathbf{x}}_0$
VISUAL NAVIGATION
$\mathring{\mathbf{x}}(t_i) = \mathbf{x}_{IMG}(t_i)$

**Figure 6.** VNS flow diagram.

- As it does not rely on absolute references, visual navigation slowly accumulates error (drifts) not only in horizontal position, but also in attitude and altitude. The main focus of this article is on how the addition of INS based priors enables the IA-VNS to reduce the drift in all six dimensions, with the resulting horizontal position IA-VNSE being just a fraction of the INSE. The attitude and altitude IA-VNSEs, although improved when compared to the VNSEs, are qualitatively inferior to the driftless INSEs, but note that their purpose is just to enable better horizontal position IA-VNS estimations, not to replace the attitude and altitude INS outputs.

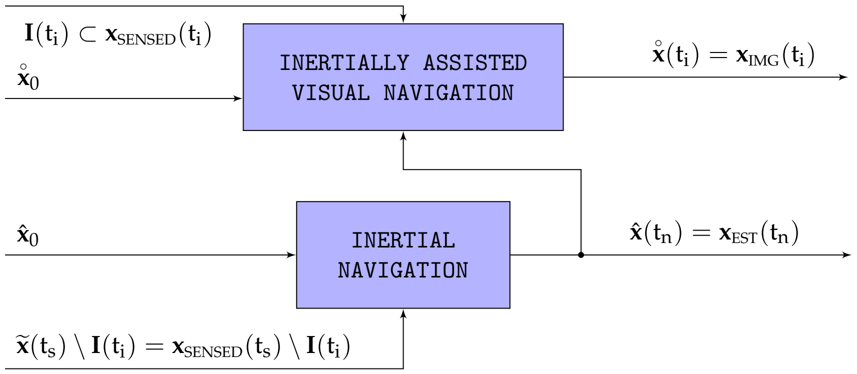
$\mathbf{I}(t_i) \subset \mathbf{x}_{SENSED}(t_i)$
$\mathring{\mathbf{x}}_0$
INERTIALLY ASSISTED VISUAL NAVIGATION
$\mathring{\mathbf{x}}(t_i) = \mathbf{x}_{IMG}(t_i)$
$\hat{\mathbf{x}}_0$
INERTIAL NAVIGATION
$\hat{\mathbf{x}}(t_n) = \mathbf{x}_{EST}(t_n)$
$\widetilde{\mathbf{x}}(t_s) \setminus \mathbf{I}(t_i) = \mathbf{x}_{SENSED}(t_s) \setminus \mathbf{I}(t_i)$

**Figure 7.** IA-VNS flow diagram.

### *7.1. Body Attitude Estimation*

Table 4 shows the NSE at the conclusion of both scenarios for the three Euler angles representing the body attitude (yaw $\psi$, pitch $\theta$, roll $\xi$), as well as the norm of the rotation vector between the real body attitude $\mathbf{q}_{NB}$ and its estimations, $\hat{\mathbf{q}}_{NB}$ by the INS and $\mathring{\mathbf{q}}_{NB}$ by the VNS or IA-VNS. The yaw angle estimation errors respond to $\Delta\hat{\psi} = \hat{\psi} - \psi$ and $\Delta\mathring{\psi} = \mathring{\psi} - \psi$, respectively; those for the body pitch and roll angles are defined accordingly. In the case of the rotation vector, the errors can be formally written as $\|\Delta\hat{\mathbf{r}}^{B}_{NB}\| = \|\hat{\mathbf{q}}_{NB} \ominus \mathbf{q}_{NB}\|$ or $\|\Delta\mathring{\mathbf{r}}^{B}_{NB}\| = \|\mathring{\mathbf{q}}_{NB} \ominus \mathbf{q}_{NB}\|$ [1], where $\ominus$ represents the $\mathbb{SO}(3)$ minus operator. In addition, Figures 8 and 9 depict the variation with time of the body attitude NSE for both scenarios, while Figure 10 shows those of each individual Euler angle for scenario #1 exclusively.

- After a short transition period following the introduction of GNSS-Denied conditions at $t_{GNSS} = 100$ s, the body attitude inertial navigation system error or INSE (blue lines) does not experience any drift with time in either scenario, and is bounded by the quality of the onboard sensors and the inertial navigation algorithms [6].

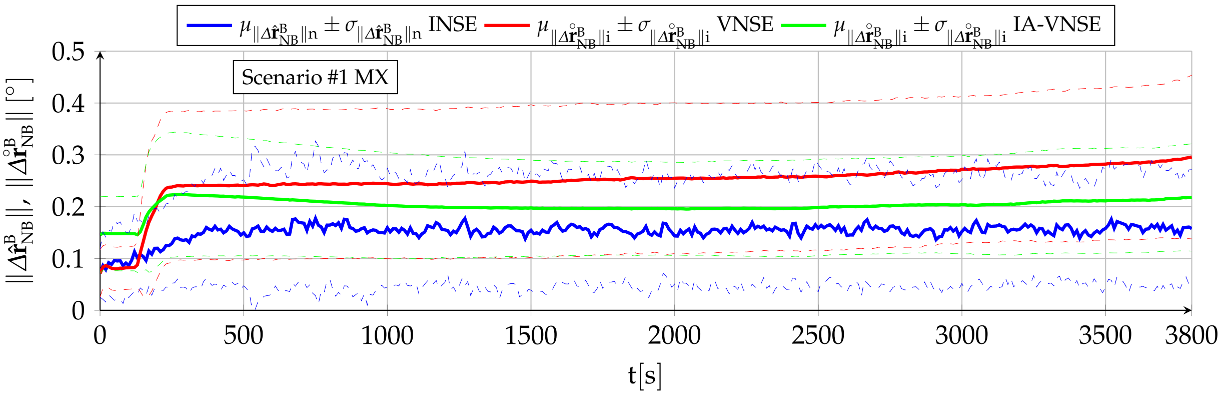

**Figure 8.** Body attitude INSE, VNSE, and IA-VNSE for scenario #1 MX (100 runs).

- With respect to the visual navigation system error or VNSE (red lines), most of the scenario #1 error is incurred during the turn maneuver at the beginning of the scenario (refer to $t_{TURN}$ within [15]), with only a slow accumulation during the rest of the trajectory, composed by a long straight flight with punctual changes in altitude and speed. Additional error growth would certainly accumulate if more turns were to occur, although this is not tested in the simulation. This statement seems to contradict the results obtained with scenario #2, in which the error grows with the initial turns but then stabilizes during the rest of the scenario, even though the aircraft is executing continuous turn maneuvers. This lack of error growth occurs because the scenario #2 trajectories are so twisted (refer to [15]) that terrain zones previously mapped reappear in the camera field of view during the consecutive turns, and are hence employed by the pose optimization phase as absolute references, resulting in a much better attitude estimation than what would occur under more spaced turns. A more detailed analysis (not shown in the figures) shows that the estimation error does not occur during the whole duration of the turns, but only during the roll-in and final roll-out maneuvers, where the optical flow is highest and hence more difficult to track by SVO (for the two evaluated scenarios, the optical flow during the roll-in and roll-out maneuvers is significantly higher than that induced by straight flight, pull-up, and push-down maneuvers, and even the turning maneuvers themselves once the bank angle is no longer changing).

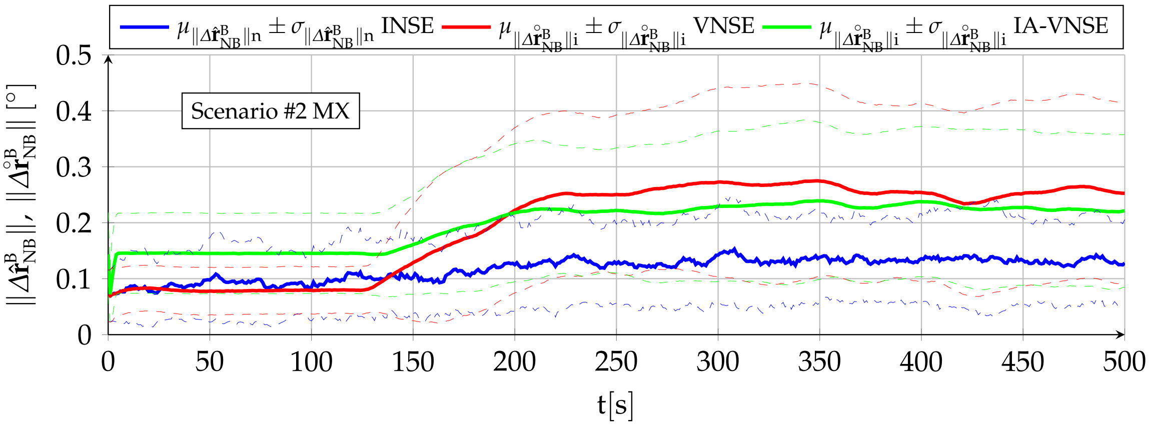

**Figure 9.** Body attitude INSE, VNSE, and IA-VNSE for scenario #2 MX (100 runs).

- The inertially assisted VNSE or IA-VNSE results (green lines) show that the introduction of priors in Section 5 works as intended and there exists a clear benefit for the use of an IA-VNS when compared to the standalone VNS described in Appendix C. In spite of IA-VNSE values at the beginning of both scenarios that are nearly double those of the VNSE (refer to Figures 8 and 9), caused by the initial pitch adjustment required to improve the fit between the homography output and the inertial estimations (Section 5.4), the balance between both errors quickly flips as soon as the aircraft starts maneuvering, resulting in body attitude IA-VNSE values significantly lower than those of the VNSE for the remaining part of both scenarios. This improvement is more significant in the case of scenario #1, as the prior based pose optimization is by design a slow adjustment that requires significant time to slowly correct attitude and altitude deviations between the visual and inertial estimations.

**Table 4.** Aggregated MX final body attitude INSE, VNSE, and IA-VNSE (100 runs). The most important metrics appear in bold.

|  | **NSE** |  |  |  | **VNSE** |  |  |  | **IA-VNSE** |  |  |  |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| [°] | $\Delta\hat{\psi}$ | $\Delta\hat{\theta}$ | $\Delta\hat{\xi}$ | $\|\|\Delta\hat{\mathbf{r}}^{\mathbf{B}}_{\mathbf{NB}}\|\|$ | $\Delta\mathring{\psi}$ | $\Delta\mathring{\theta}$ | $\Delta\mathring{\xi}$ | $\|\|\Delta\mathring{\mathbf{r}}^{\mathbf{B}}_{\mathbf{NB}}\|\|$ | $\Delta\mathring{\psi}$ | $\Delta\mathring{\theta}$ | $\Delta\mathring{\xi}$ | $\|\|\Delta\mathring{\mathbf{r}}^{\mathbf{B}}_{\mathbf{NB}}\|\|$ |
| Scenario #1 MX ($\mathrm{t_{END}}$) |  |  |  |  |  |  |  |  |  |  |  |  |
| mean | +0.03 | −0.03 | −0.00 | **0.158** | +0.03 | +0.08 | +0.00 | **0.296** | +0.03 | −0.01 | −0.03 | **0.218** |
| std | 0.18 | 0.05 | 0.06 | 0.114 | 0.13 | 0.23 | 0.21 | 0.158 | 0.11 | 0.16 | 0.14 | 0.103 |
| max | −0.61 | −0.27 | −0.23 | 0.611 | +0.63 | +0.74 | +0.78 | 0.791 | +0.55 | −0.37 | −0.51 | 0.606 |
| Scenario #2 MX ($\mathrm{t_{END}}$) |  |  |  |  |  |  |  |  |  |  |  |  |
| mean | −0.02 | +0.01 | +0.00 | **0.128** | +0.02 | −0.02 | +0.00 | **0.253** | +0.02 | −0.00 | +0.01 | **0.221** |
| std | 0.13 | 0.05 | 0.05 | 0.078 | 0.08 | 0.21 | 0.20 | 0.161 | 0.08 | 0.16 | 0.19 | 0.137 |
| max | +0.33 | −0.15 | +0.15 | 0.369 | +0.22 | −0.65 | −0.73 | 0.730 | +0.24 | +0.62 | +0.74 | 0.788 |

Qualitatively, the biggest difference between the three estimations resides in the nature of the errors. While the attitude INSE is bounded, drift is present in both the VNS and IA-VNS estimations. The drift resulting from the Monte Carlo simulations may be small, and so is the attitude estimation error $\|\Delta\mathring{\mathbf{r}}^{\mathrm{B}}_{\mathrm{NB}}\|$, but more challenging conditions with more drastic maneuvers and a less idealized image generation process than that described in Section 6 may generate additional drift.

Focusing now on the quantitative results shown in Table 4, aggregated errors for each individual Euler angle are always unbiased and zero mean for each of the three estimations (INS, VNS, IA-VNS), as the means tend to zero as the number of runs grows, and are much smaller than both the standard deviations and the maximum values. With respect to the attitude error $\|\Delta\hat{\mathbf{r}}^{\mathrm{B}}_{\mathrm{NB}}\|$ and $\|\Delta\mathring{\mathbf{r}}^{\mathrm{B}}_{\mathrm{NB}}\|$, their aggregated means are not zero (they are norms), but are nevertheless quite repetitive in all three cases, as the mean is always significantly higher than the standard deviation, while the maximum values only represent a small multiple of the means. It is interesting to point out that while in the case of the INSE the contribution of the yaw error is significantly higher than that of the pitch and roll errors, the opposite occurs for both the VNSE and the IA-VNSE. This makes sense as the the gravity direction is employed by the INS as a reference from where the estimated pitch and roll angles can not deviate, but slow changes in yaw generate larger optical flow variations than those caused by pitch and roll variations.

These results prove that the algorithms proposed in Section 5 succeed when employing the inertial pitch and bank angles ($\hat{\theta}$, $\hat{\xi}$), whose errors are bounded, to limit the drift of their visual counterparts ($\mathring{\theta}$, $\mathring{\xi}$), as $\sigma_{\mathrm{END}\mathring{\theta}}$ and $\sigma_{\mathrm{END}\mathring{\xi}}$ are significantly lower for the IA-VNS than for the VNS (as the individual Euler angle metrics are unbiased or zero mean, the benefits of the proposed approach are reflected in the variation of the remaining metrics, this is, the standard deviation and the maximum value). Remarkably, this is achieved with no degradation in the body yaw angle, as $\sigma_{\mathrm{END}\mathring{\psi}}$ remains stable. Note that adjusting the output of certain variables in a minimization algorithm (such as pose optimization) usually results in a degradation in the accuracy of the remaining variables as the solution moves away from the true optimum. In this case, however, the improved fit between the adjusted aircraft pose and the terrain displayed in the images, results in the SVO pipeline also slightly improving its body yaw estimation $\overset{\circ}{\psi}$. Section 7.3 shows how the benefits of an improved fit between the displayed terrain and the adjusted pose also improve the horizontal position estimation.

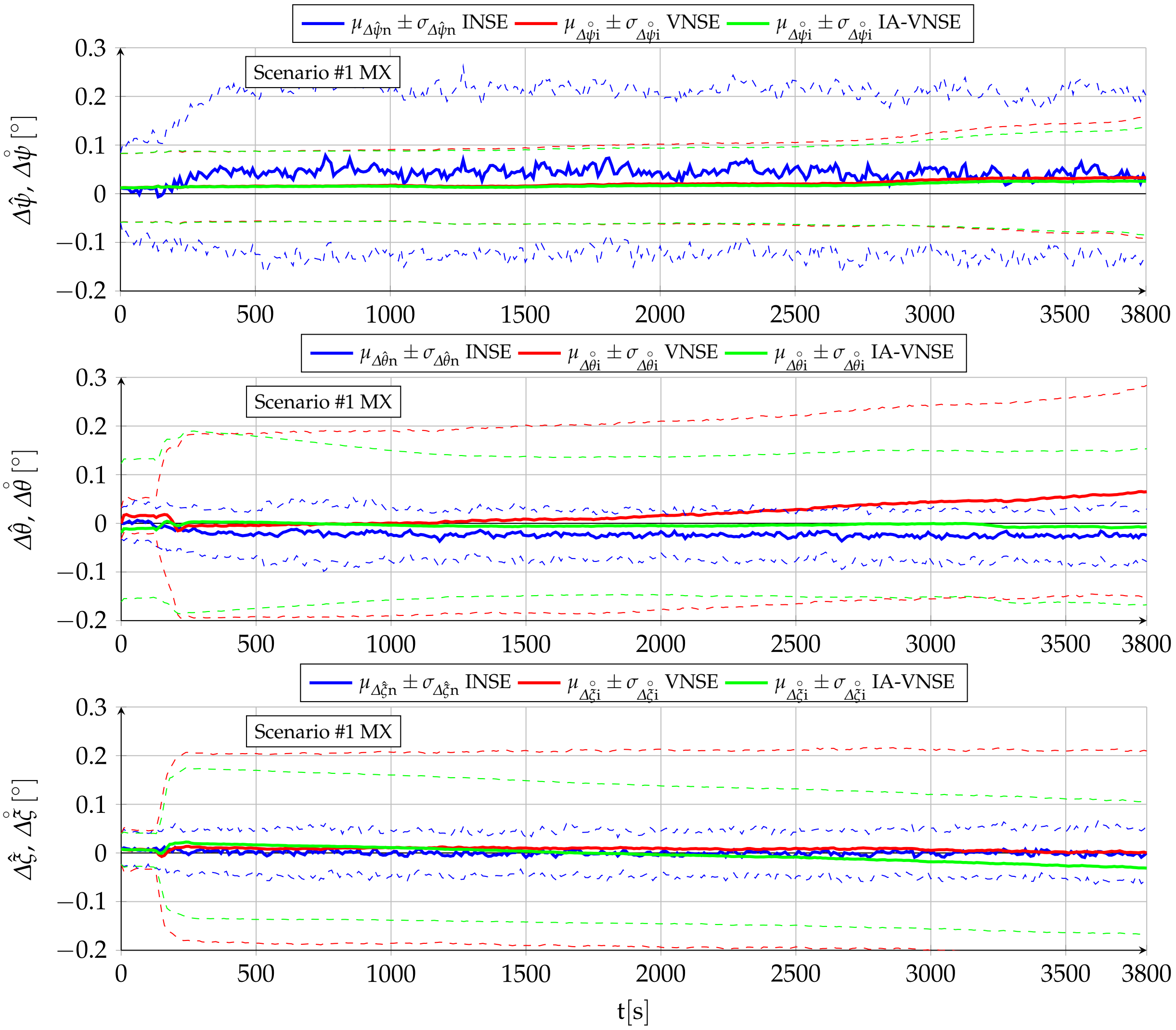

**Figure 10.** Body Euler angles INSE, VNSE, and IA-VNSE for scenario #1 MX (100 runs).

In the case of a real life scenario based on a more realistic image generation process than that described in Section 6, the VNS would likely incur in additional body attitude drift than in the simulations. If this were to occur, the IA-VNS pose adjustment algorithms described in Section 5 would react more aggressively to counteract the higher pitch and bank deviations, eliminating most of the extra drift, although it is possible that higher pose adjustment parameters than those listed in Table 2 would be required. The IA-VNS is hence more resilient against high drift values than the VNS.

### *7.2. Vertical Position Estimation*

Table 5 contains the vertical position NSE ($\Delta\hat{h} = \hat{h} - h$, $\Delta\overset{\circ}{h} = \overset{\circ}{h} - h$) at the conclusion of both scenarios, which can be considered unbiased or zero mean in all six cases (two scenarios and three estimation methods) as the mean $\mu_{ENDh}$ is always significantly lower than both the standard deviation $\sigma_{ENDh}$ or the maximum value $\zeta_{END|h|}$. The NSE evolution with time is depicted in Figures 11 and 12, which also include (magenta lines) those Monte Carlo executions that result in the highest IA-VNSEs.

- The geometric altitude INSE (blue lines) is bounded by the change in atmospheric pressure offset since the time the GNSS signals are lost. Refer to [6] for additional information.
- The VNS estimation of the geometric altitude (red lines) is worse than that by the INS both qualitatively and quantitatively, even with the results being optimistic because of the ideal image generation process employed in the simulation. A continuous drift or error growth with time is present, and results in final errors much higher than those obtained with the GNSS-Denied inertial filter. These errors are logically bigger for scenario #1 because of its much longer duration.

**Table 5.** Aggregated MX final vertical position INSE, VNSE, and IA-VNSE (100 runs). The most important metrics appear in bold.

| Scenario MX ($t_{END}$) | [m] | INSE $\Delta\hat{h}$ | VNSE $\Delta\mathring{h}$ | IA-VNSE $\Delta\mathring{h}$ |
|---|---|---|---|---|
| #1 | mean | −4.18 | +82.91 | +22.86 |
| | std | **25.78** | **287.58** | **49.17** |
| | max | −70.49 | +838.32 | +175.76 |
| #2 | mean | +0.76 | +3.45 | +3.59 |
| | std | **7.55** | **20.56** | **13.01** |
| | max | −19.86 | +72.69 | +71.64 |

A small percentage of this drift can be attributed to the slow accumulation of error inherent to the SVO motion thread algorithms introduced in Appendix C, but most of it results from adding the estimated relative pose between two consecutive images to a pose (that of the previous image) with an attitude that already possesses a small pitch error (refer to the attitude estimation analysis in Section 7.1). Note that even a fraction of a degree deviation in pitch can result in hundreds of meters in vertical error when applied to the total distance flown in scenario #1, as SVO can be very precise when estimating pose changes between consecutive images, but lacks any absolute reference to avoid slowly accumulating these errors over time. This fact is precisely the reason why the vertical position VNSE grows more slowly in the second half of scenario #2, as shown in Figure 12. As explained in Section 7.1 above, continuous turn maneuvers cause previously mapped terrain points to reappear in the camera field of view, stopping the growth in the attitude error (pitch included), which indirectly has the effect of slowing the growth in altitude estimation error.

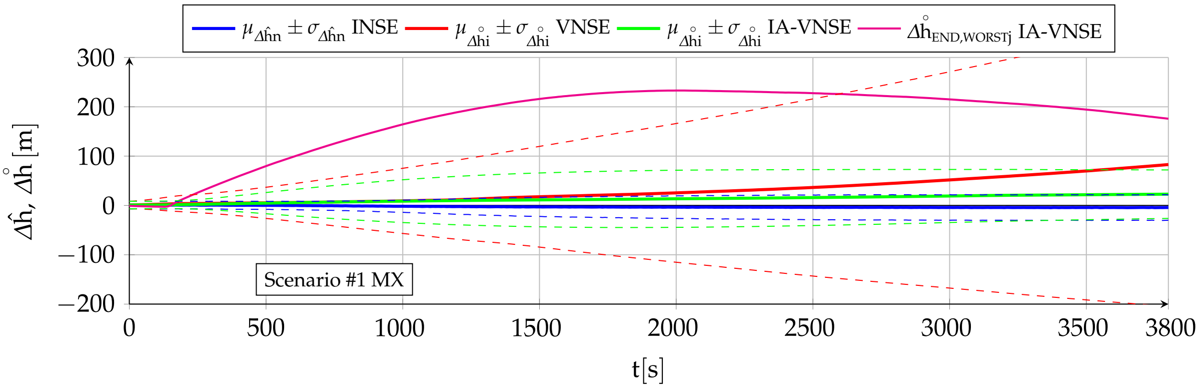

**Figure 11.** Vertical position INSE, VNSE, and IA-VNSE for scenario #1 MX (100 runs).

- The benefits of introducing priors to limit the differences between the visual and inertial altitude estimations are reflected in the IA-VNSE (green lines). The error reduction is drastic in the case of the scenario #1, where its extended duration allows the pose optimization small pitch adjustments to accumulate into significant altitude corrections over time, and less pronounced but nevertheless significant for scenario #2, where the VNSE (an hence also the IA-VNSE) results already benefit from previously mapped terrain points reappearing in the aircraft field of view as a result of the continuous maneuvers. It is necessary to remark the amount of the improvement, as the final standard deviation $\sigma_{\text{ENDh}}$ diminishes from 287.58 to 49.17 m for scenario #1, and from 20.56 to 13.01 m for scenario #2.

The benefits of the prior based pose optimization algorithm can be clearly observed in the case of the scenario #1 execution with the worst final altitude estimation error, whose error variation with time is depicted in Figure 11 (magenta line). After a rapid growth in the first third of the scenario following a particularly negative estimation during the initial turn, the altitude error reaches a maximum of +233.05 m at 2007.5 s. Attitude adjustment has become active long before, lowering the estimated pitch angle to first diminish the growth of the altitude error and then being able to reduce the error itself, reaching a final value of +175.76 m at $\text{t}_{\text{END}}$. As soon as the differences between the visual pitch, bank, or altitude estimations ($\mathring{\theta}$, $\mathring{\xi}$, $\mathring{\text{h}}$) and their inertial counterparts ($\hat{\theta}$, $\hat{\xi}$, $\hat{\text{h}}$) exceed certain limits (Section 5), the attitude adjustment comes into play and slowly adjusts the aircraft pitch to prevent the visual altitude from deviating in excess from the inertial one. This behavior not only improves the IA-VNS altitude estimation accuracy when compared to that of the VNS, but also its resilience, as the system actively opposes elevated altitude errors.

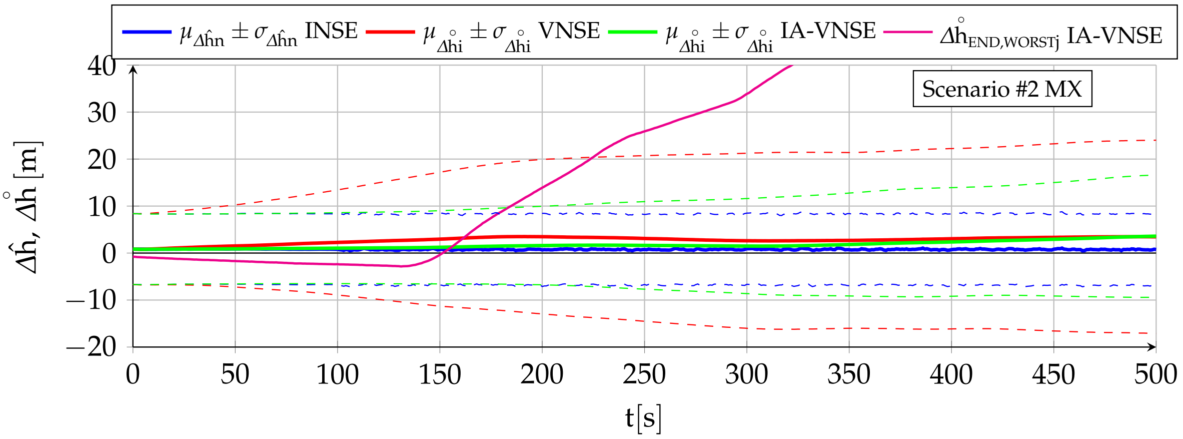

**Figure 12.** Vertical position INSE, VNSE, and IA-VNSE for scenario #2 MX (100 runs).

Significantly better altitude estimation errors (closer to the inertial ones) could be obtained if more aggressive settings were employed for $\Delta\mathring{\theta}^{\circ\circ}_{1,\text{MAX}}$ and $\Delta\mathring{\theta}^{\circ\circ}_{2,\text{MAX}}$ within Table 2, as the selected values are far from the level at which the pose optimization convergence is compromised. This would result in more aggressive adjustments and important accuracy improvements for those cases in which altitude error growth is highest. The settings employed in this article are modest, as the final objective is not to obtain the smallest possible attitude or vertical position IA-VNSE (as they are always bigger than their INSE counterparts), but to limit them to acceptable levels so SVO can build a more accurate terrain map, improving the fit between the multiple terrain 3D points displayed in the images and the estimated aircraft pose. To do so it is mandatory to balance the pitch and bank angle adjustments with the need to stick to solutions close to those that minimize the reprojection error, as explained in Section 5. Higher $\Delta\mathring{\theta}^{\circ\circ}_{1,\text{MAX}}$ and $\Delta\mathring{\theta}^{\circ\circ}_{2,\text{MAX}}$ accelerate the adjustments but may decrease the quality of the map. It is expected that a better rendition of the real 3D position of the features detected in the keyframes as they are tracked along successive images will lower the incremental horizontal displacement errors, and, hence, result in a lower horizontal position IA-VNSE, which is the real objective for the introduction of the priors.

The IA-VNS altitude estimation improvements over those of the VNS are not only quantitative. Figure 11 shows no increment in $\sigma_{\overset{\circ}{\mathrm{h}}\mathrm{n}}$ (green lines) in the second half of scenario #1 (once on average the deviation has activated the attitude adjustment feature). The altitude estimation by the IA-VNS can hence also be described as bounded and driftless, which represents a qualitative and not only quantitative improvement over that of the VNS. The bounds are obviously bigger for the IA-VNS than for the INS. In the case of scenario #2, Figure 12 shows a slow but steady $\sigma_{\overset{\circ}{\mathrm{h}}\mathrm{n}}$ growth with time, but this is only because the error amount on average is not yet significant enough to activate the attitude adjustment feature within pose optimization.

### *7.3. Horizontal Position Estimation*

The horizontal position estimation capabilities of the INS, VNS, and IA-VNS share the fact that all of them exhibit an unrestrained drift or growth with time, as shown in Figures 13 and 14. The errors obtained at the end of both scenarios are shown in Table 6, following the same scheme as in previous sections. While the approximately linear INS drift appears when integrating the bounded ground velocity errors [6], the visual drifts (both VNS and IA-VNS) originate in the slow accumulation of errors caused by the concatenation of the relative poses between consecutive images without absolute references, but also show a direct relationship with the scale error committed when estimating the aircraft height over the terrain during the initial homography (Appendix C).

**Table 6.** Aggregated MX final horizontal position INSE, VNSE, and IA-VNSE (100 runs). The most important metrics appear in bold.

| **Scenario MX ($\mathbf{t}_{\mathrm{END}}$)** | | | **INSE** | | **VNSE** | | **IA-VNSE** | |
|---|---|---|---|---|---|---|---|---|
| | | **Distance** | $\Delta\hat{\mathrm{x}}_{\mathrm{HOR}}$ | | $\Delta\mathring{\mathrm{x}}_{\mathrm{HOR}}$ | | $\Delta\mathring{\mathrm{x}}_{\mathrm{HOR}}$ | |
| | | [m] | [m] | [%] | [m] | [%] | [m] | [%] |
| #1 | mean | 107,873 | 7276 | **7.10** | 4179 | **3.82** | 488 | **0.46** |
| | std | 19,756 | 4880 | 5.69 | 3308 | 2.73 | 350 | 0.31 |
| | max | 172,842 | 25,288 | 32.38 | 21,924 | 14.22 | 1957 | 1.48 |
| #2 | mean | 14,198 | 216 | **1.52** | 251 | **1.77** | 33 | **0.23** |
| | std | 1176 | 119 | 0.86 | 210 | 1.48 | 26 | 0.18 |
| | max | 18,253 | 586 | 4.38 | 954 | 7.08 | 130 | 0.98 |

In the case of the VNS (red lines), its scenario #1 horizontal position estimations appear to be significantly more accurate than those of the INS (blue lines). Note, however, that the ideal image generation process discussed in Section 6 implies that the simulation results should be treated as a best case only, and that the results obtained in real world conditions would likely imply a higher horizontal position drift. The drift experienced by the VNS in Figure 14 (scenario #2) also shows the same diminution in its slope in the second half of the scenario discussed in previous sections, which is attributed to previously mapped terrain points reappearing in the camera field of view as a consequence of the continuous turns present in scenario #2. Additionally, notice how the VNSE starts growing at the beginning of the scenario, while the INSE only starts doing so after the GNSS signals are lost at $\mathrm{t}_{\mathrm{GNSS}} = 100\,\mathrm{s}$ [6].

The IA-VNS (green lines) results in major horizontal position estimation improvements over the VNS. The final horizontal position error mean $\mu_{\mathrm{END}\Delta\mathring{\mathrm{x}}_{\mathrm{HOR}}}$ diminishes from 3.82 to 0.46% for scenario #1, and from 1.77 to 0.23% for scenario #2. The repeatability of the results also improves, as the final standard deviation $\sigma_{\mathrm{END}\Delta\mathring{\mathrm{x}}_{\mathrm{HOR}}}$ falls from 2.73 to 0.31% and from 1.48 to 0.18% for both scenarios. Note that although these results may be slightly optimistic due to the optimized image generation process, they are much more accurate than those obtained with the INS, for which the error mean and standard deviation amount to 7.10 and 5.69% for scenario #1, and 1.52 and 0.86% in case of scenario #2.

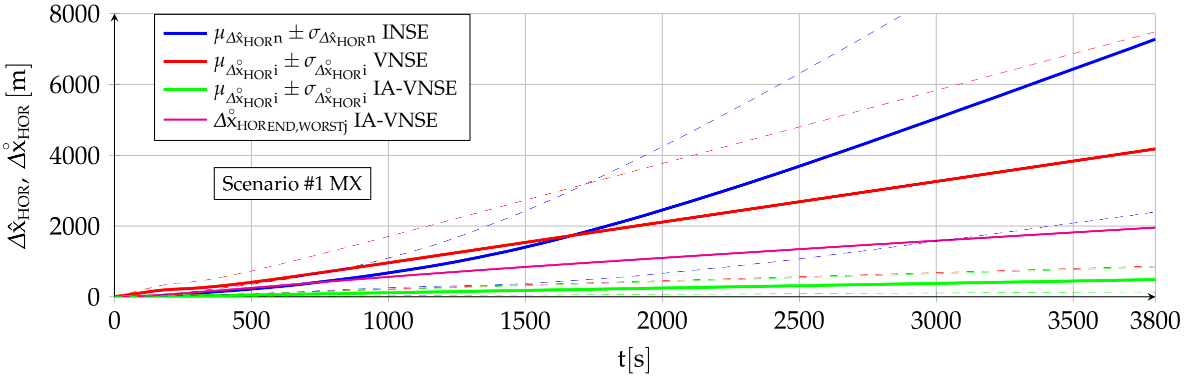

**Figure 13.** Horizontal position INSE, VNSE, and IA-VNSE for scenario #1 MX (100 runs).

It is interesting to remark how the prior based pose optimization described in Section 5, an algorithm that adjusts the aircraft pitch and bank angles based on deviations between the visually estimated pitch angle, bank angle, and geometric altitude, and their inertially estimated counterparts, is capable of not only improving the visual estimations of those three variables, but doing so with a minor improvement in the body yaw estimation and an extreme reduction in the horizontal position error. When the cost function within an optimization algorithm is modified to adjust certain target components, the expected result is that this can be achieved only at the expense of the accuracy in the remaining target components, not in addition to it. The reason why in this case all target components improve lies in that the adjustment creates a better fit between the ground terrain and associated 3D points depicted in the images on one side, and the estimated aircraft pose indicating the position and attitude from where the images are taken on the other.

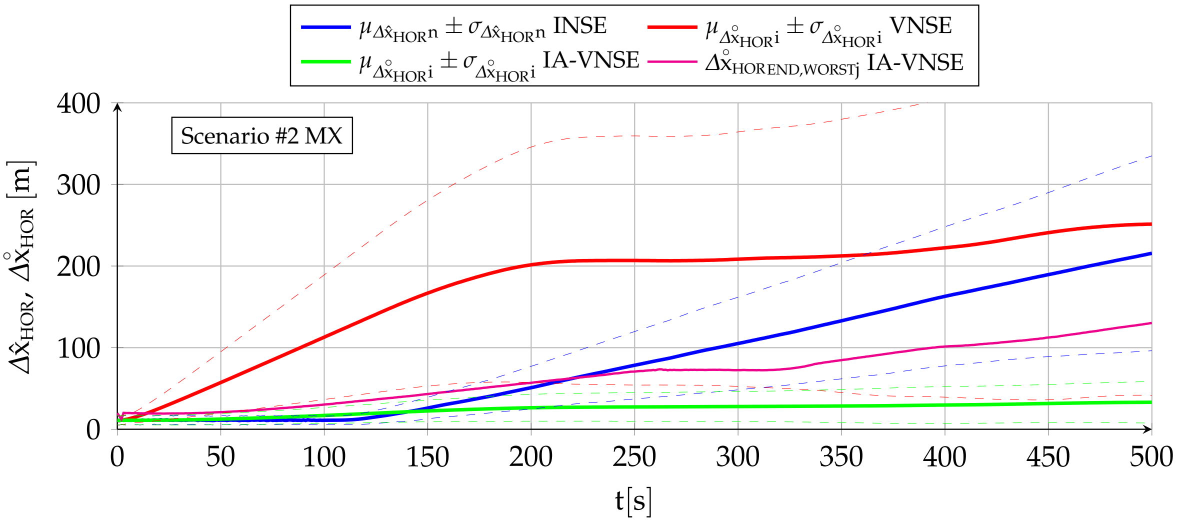

**Figure 14.** Horizontal position INSE, VNSE, and IA-VNSE for scenario #2 MX (100 runs).

## 8. Influence of Terrain Type

The type of terrain overflown by the aircraft has a significant influence on the performance of the visual navigation algorithms, which can not operate unless the feature detector is capable of periodically locating features in the various keyframes, and which also requires the depth filter to correctly estimate the 3D terrain coordinates of each feature (Appendix C). The terrain texture (or lack of) and its elevation relief are, hence, the two most important characteristics in this regard. To evaluate its influence, each of the scenario #1 100 Monte Carlo runs are executed flying above four different zones or types of terrain, intended to represent a wide array of conditions; images representative of each zone as viewed by the onboard camera are included below. The use of terrains that differ in both their texture and vertical relief is intended to provide a more complete validation of the proposed algorithms. Note that the only variation among the different simulations is the terrain type, as all other parameters defining each scenario (mission, aircraft, sensors, weather, wind, turbulence, geophysics, initial estimations) are exactly the same for all simulation runs.

- The "desert" (DS) zone (left image within Figure 15) is located in the Sonoran desert of southern Arizona (USA) and northern Mexico. It is characterized by a combination of bajadas (broad slopes of debris) and isolated very steep mountain ranges. There is virtually no human infrastructure or flat terrain, as the bajadas have sustained slopes of up to 7°. The altitude of the bajadas ranges from 300 to 800 m above MSL, and the mountains reach up to 800 m above the surrounding terrain. Texture is abundant because of the cacti and the vegetation along the dry creeks.
- The "farm" (FM) zone (right image within Figure 15) is located in the fertile farmland of southeastern Illinois and southwestern Indiana (USA). A significant percentage of the terrain is made of regular plots of farmland, but there also exists some woodland, farm houses, rivers, lots of little towns, and roads. It is mostly flat with an altitude above MSL between 100 and 200 m, and altitude changes are mostly restricted to the few forested areas. Texture is non-existent in the farmlands, where extracting features is often impossible.

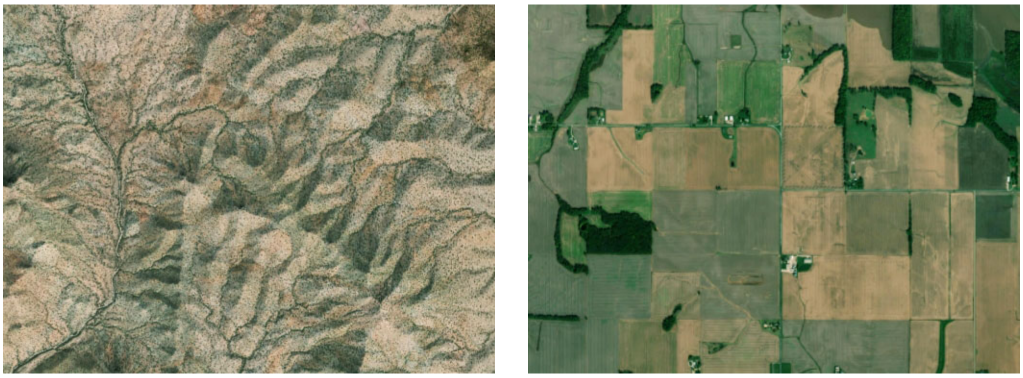

**Figure 15.** Typical "desert" (DS) and "farm" (FM) terrain views.

- The "forest" (FR) zone (left image within Figure 16) is located in the deciduous forestlands of Vermont and New Hampshire (USA). The terrain is made up of forests and woodland, with some clearcuts, small towns, and roads. There are virtually no flat areas, as the land is made up by hills and small to medium size mountains that are never very steep. The valleys range from 100 to 300 m above MSL, while the tops of the mountains reach 500 to 900 m. Features are plentiful in the woodlands.
- The "mix" (MX) zone (right image within Figure 16) is located in northern Mississippi and extreme southwestern Tennessee (USA). Approximately half of the land consists of woodland in the hills, and the other half is made up by farmland in the valleys, with a few small towns and roads. Altitude changes are always present and the terrain is never flat, but they are smaller than in the DS and FR zones, with the altitude oscillating between 100 and 200 m above MSL.

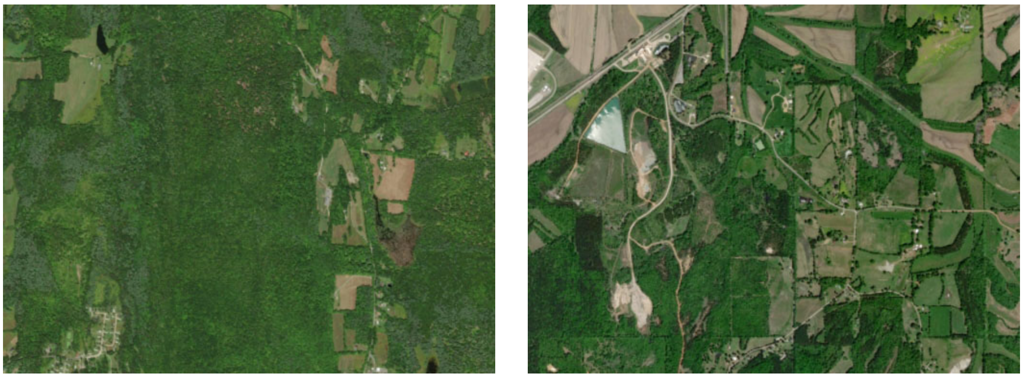

**Figure 16.** Typical "forest" (FR) and "mix" (MX) terrain views.

The short duration and continuous maneuvering of scenario #2 enables the use of two additional terrain types. These two zones are not employed in scenario #1 because the authors could not locate wide enough areas with a prevalence of this type of terrain (note that scenario #1 trajectories can conclude up to 125 km in any direction from its initial coordinates, but only 12 km for scenario #2).

- The "prairie" (PR) zone (left image within Figure 17) is located in the Everglades floodlands of southern Florida (USA). It consists of flat grasslands, swamps, and tree islands located a few meters above MSL, with the only human infrastructure being a few dirt roads and landing strips, but no settlements. Features may be difficult to obtain in some areas due to the lack of texture.
- The "urban" (UR) zone (right image within Figure 17) is located in the Los Angeles metropolitan area (California, USA). It is composed by a combination of single family houses and commercial buildings separated by freeways and streets. There is some vegetation but no natural landscapes, and the terrain is flat and close to MSL.

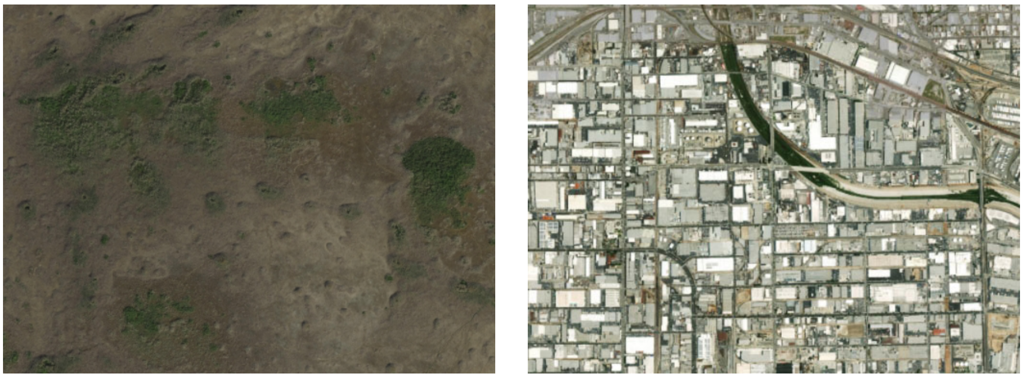

**Figure 17.** Typical "prairie" (PR) and "urban" (UR) terrain views.

The MX terrain zone is considered the most generic and hence employed to evaluate the visual algorithms in Section 7. Although scenario #2 also makes use of the four terrain types listed for scenario #1 (DS, FM, FR, and MX), it is worth noting that the variability of the terrain is significantly higher for scenario #1 because of the bigger land extension covered. The altitude relief, abundance or scarcity of features, land use diversity, and presence of rivers and mountains is, hence, more varied when executing a given run of scenario #1 over a certain type of terrain, than when executing the same run for scenario #2. From the point of view of the influence of the terrain on the visual navigation algorithms, scenario #1 should theoretically be more challenging than #2.

Table 7 and Figure 18 show the horizontal position IA-NVSE for scenario #1 and all terrain types. Table 8 and Figure 19 do the same for scenario #2.

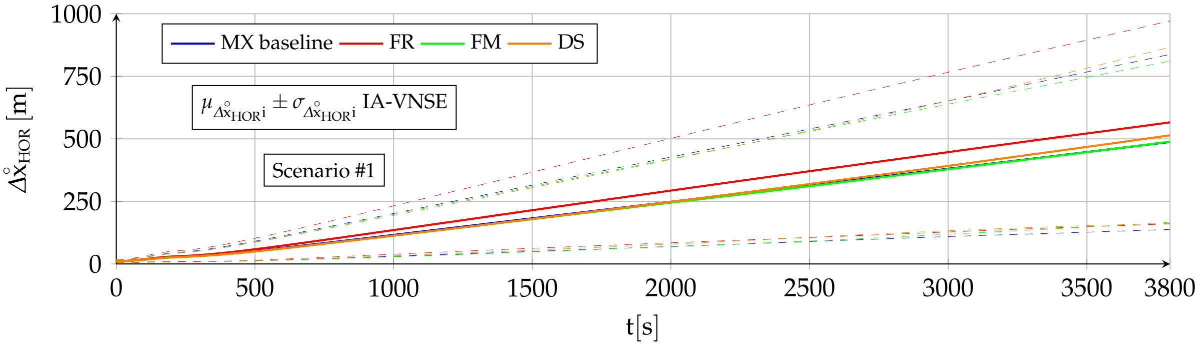

**Figure 18.** Influence of terrain type on horizontal position IA-VNSE for scenario #1 (100 runs).

**Table 7.** Influence of terrain type on final horizontal position IA-VNSE for scenario #1 (100 runs). The most important metrics appear in bold.

| **Scenario #1 Zone** | | **MX** | | **FR** | | **FM** | | **DS** | |
|---|---|---|---|---|---|---|---|---|---|
| $\Delta\mathring{x}_{HOR}(t_{END})$ | | **[m]** | [%] | **[m]** | [%] | **[m]** | [%] | **[m]** | [%] |
| IA-VNSE | mean | 488 | **0.46** | 566 | **0.53** | 489 | **0.45** | 514 | **0.48** |
| | std | 350 | 0.31 | 406 | 0.38 | 322 | 0.28 | 352 | 0.31 |
| | max | 1957 | 1.48 | 2058 | 1.71 | 1783 | 1.34 | 1667 | 1.37 |

The influence of the terrain type on the horizontal position IA-VNSE is very small, with slim differences among the various evaluated terrains. The only terrain type that clearly deviates from the others is FR, with slight but consistently worse horizontal position estimations for both scenarios. This behavior stands out as the abundant texture and continuous smooth vertical relief of the FR terrain is a priori beneficial for the visual algorithms.

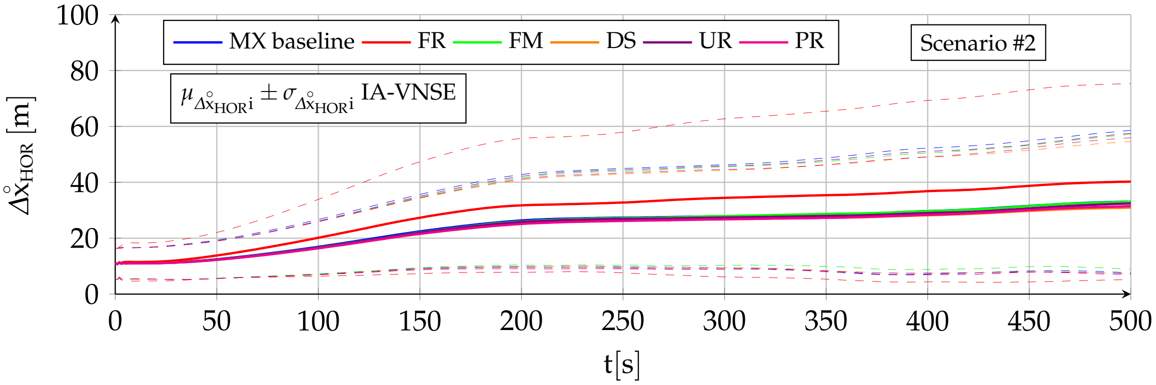

**Figure 19.** Influence of terrain type on horizontal position IA-VNSE for scenario #2 (100 runs).

Although beneficial for the SVO pipeline, the more pronounced vertical relief of the FR terrain type breaches the flat terrain assumption of the initial homography (Appendix C), hampering its accuracy, and, hence, results in less precise initial estimations, including that of the scale. The IA-VNS has no means to compensate the initial scale errors, which remain approximately equal (percentage wise) for the full duration of both scenarios.

**Table 8.** Influence of terrain type on final horizontal position IA-VNSE for scenario #2 (100 runs). The most important metrics appear in bold.

| **Scenario #2** | **Zone** | **MX** | | **FR** | | **FM** | | **DS** | | **UR** | | **PR** | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $\Delta\mathring{\mathbf{x}}_{\mathbf{HOR}}(\mathbf{t}_{\mathbf{END}})$ | | [**m**] | [%] | [**m**] | [%] | [**m**] | [%] | [**m**] | [%] | [**m**] | [%] | [**m**] | [%] |
| IA-VNSE | mean | 33 | **0.23** | 40 | **0.28** | 33 | **0.23** | 31 | **0.22** | 32 | **0.23** | 31 | **0.22** |
| | std | 26 | 0.18 | 35 | 0.24 | 24 | 0.17 | 24 | 0.17 | 25 | 0.18 | 25 | 0.17 |
| | max | 130 | 0.98 | 188 | 1.29 | 117 | 0.85 | 114 | 0.86 | 128 | 0.96 | 119 | 0.90 |

A similar but opposite reasoning is applicable to the FM type and in a lesser degree to the UR and PR types. Although a flat terrain in which all terrain features are located at a similar altitude is detrimental to the overall accuracy of SVO, and results in slightly worse body attitude and vertical position estimations, it is beneficial for the homography initialization and the scale determination, resulting in consistently more accurate horizontal position estimations.

## 9. Summary of Results

This article proposes a Semi-Direct Visual Odometry (SVO)-based Inertially Assisted Visual Navigation System (IA-VNS) installed onboard a fixed wing autonomous UAV that takes advantage of the GNSS-Denied estimations provided by an Inertial Navigation System (INS) to assist the visual pose optimization algorithms. The method is inspired in a Proportional Integral (PI) control loop, in which the inertial attitude and altitude outputs act as targets to ensure that the visual estimations do not deviate in excess from their inertial counterparts, resulting in major improvements when estimating the aircraft horizontal position without the use of GNSS signals. The results obtained when applying the proposed algorithms to high fidelity Monte Carlo simulations of two scenarios representative of the challenges of GNSS-Denied navigation indicate the following:

- The **body attitude** estimation shows significant quantitative improvements over a standalone Visual Navigation System (VNS) in both pitch and bank angle estimations, with no negative influence on the yaw angle estimations. A small amount of drift with time is present, and can not be fully eliminated. Body pitch and bank angle estimations do not deviate in excess from their INS counterparts, while the body yaw angle visual estimation is significantly more accurate than that obtained by the INS.
- The **vertical position** estimation shows major improvements over that of a standalone VNS, not only quantitatively but also qualitatively, as drift is fully eliminated. The visual estimation does not deviate in excess from the inertial one, which is bounded by atmospheric physics.
- The **horizontal position** estimation, whose improvement is the main objective of the proposed algorithm, shows major gains when compared to either the standalone VNS or the INS, although drift is still present.

In addition, although the **terrain** texture (or lack of) and its elevation relief are key factors for the visual odometry algorithms, their influence on the aircraft pose estimation results are slim, and the accuracy of the IA-VNS does not vary significantly among the various evaluated terrain types.

## 10. Conclusions

The proposed inertially assisted VNS (IA-VNS), which in addition to the images taken by an onboard camera also relies on the outputs of an INS specifically designed for the challenges faced by autonomous fixed wing aircraft that encounter GNSS-Denied conditions, possesses significant advantages in both accuracy and resilience when compared with a standalone VNS, the most important of which is a major reduction in its horizontal position drift independently of the terrain type overflown by the aircraft. The proposed IA-VNS can significantly increase the possibilities of the aircraft safely reaching the vicinity of the intended recovery location upon the loss of GNSS signals, from where it can be landed by remote control.

**Author Contributions:** Conceptualization, E.G.; methodology, E.G.; software, E.G.; validation, E.G.; formal analysis, E.G.; investigation, E.G.; resources, E.G.; data curation, E.G.; writing—original draft preparation, E.G.; writing—review and editing, A.B.; visualization, E.G.; supervision, A.B.; project administration, A.B.; funding acquisition, A.B. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work has received funding from RoboCity2030-DIH-CM, Madrid Robotics Digital Innovation Hub, S2018/NMT-4331, funded by R&D Activity Programs in the Madrid Community and co-financed by the EU Structural Funds.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** An open source C++ implementation of the described algorithms can be found at [16].

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

## Abbreviations

The following abbreviations are used in this manuscript:

| | |
|---|---|
| BRIEF | Binary Robust Independent Elementary Features |
| DS | DeSert terrain type |
| DSO | Direct Sparse Odometry |
| ECEF | Earth Centered Earth Fixed |
| EKF | Extended Kalman Filter |
| FAST | Features from Accelerated Segment Test |
| FM | FarM terrain type |
| FR | FoRest terrain type |
| GNSS | Global Navigation Satellite System |
| IA-VNS | Inertially Assisted VNS |
| IA-VNSE | Inertially Assisted Visual Navigation System Error |
| IMU | Inertial Measurement Unit |
| INS | Inertial Navigation System |
| INSE | Inertial Navigation System Error |
| iSAM | Incremental Smoothing And Mapping |
| ISO | International Organization for Standardization |
| LSD | Large Scale Direct |
| MAV | Micro Air Vehicle |
| MSCKF | Multi State Constraint Kalman Filter |
| MSF | Multi-Sensor Fusion |
| MSL | Mean Sea Level |
| MX | MiX terrain type |
| NED | North East Down |
| NSE | Navigation System Error |
| OKVIS | Open Keyframe Visual Inertial SLAM |
| ORB | Oriented FAST and Rotated BRIEF |
| PI | Proportional Integral |
| PR | Praire terrain type |
| RANSAC | Random SAmple Consensus |
| ROC | Rate Of Climb |
| ROVIO | Robust Visual Inertial Odometry |

| | |
|---|---|
| SLAM | Simultaneous Localization And Mapping |
| SLERP | Spherical linear interpolation |
| SVO | Semi direct Visual Odometry |
| SWaP | Size, Weight, and Power |
| TAS | True Air Speed |
| UAV | Unmanned Aerial Vehicle |
| UR | Urban terrain type |
| USA | United States of America |
| VINS | Visual Inertial Navigation System |
| VIO | Visual Inertial Odometry |
| VNS | Visual Navigation System |
| VNSE | Visual Navigation System Error |
| VO | Visual Odometry |
| WGS84 | World Geodetic System 1984 |

## Appendix A. Optical Flow

Consider a pinhole camera [24] (one that adopts an ideal perspective projection) such as that depicted in Figure A1. The *image frame* $F_{IMG}$ is a two-dimensional Cartesian reference frame $F_{IMG} = \{O_{IMG}, \mathbf{i}_1^{IMG}, \mathbf{i}_2^{IMG}\}$ whose axes are parallel to those of the $F_C$ camera frame ($\mathbf{i}_1^{IMG} \parallel \mathbf{i}_1^C$, $\mathbf{i}_2^{IMG} \parallel \mathbf{i}_2^C$), and whose origin $O_{IMG}$ is located on the focal plane displaced a distance $\mathbf{c}^{IMG}$ from the principal point so the $F_{IMG}$ coordinates $p_1^{IMG}$ and $p_2^{IMG}$ of any point in the image domain $\Omega$ are always positive. The perspective projection map $\mathbf{p}^{IMG} = \Pi(\mathbf{p}^C)$ that converts points viewed in $F_C$ into $F_{IMG}$ is hence the following:

$$p_1^{IMG} = \frac{f}{s_{PX}} \frac{p_1^C}{p_3^C} + c_1^{IMG} \tag{A1}$$

$$p_2^{IMG} = \frac{f}{s_{PX}} \frac{p_2^C}{p_3^C} + c_2^{IMG} \tag{A2}$$

Consider also that the camera is moving with respect to the Earth while maintaining within its field of view a given point $\mathbf{p}$ fixed to the Earth surface. The composition of positions and its time derivation, considering ECEF as $F_E$, the camera frame as $F_C$, and a frame $F_P$ with its origin in the terrain point $\mathbf{p}$ that does not move with respect to $F_E$, results in the following expression when viewed in $F_C$:

$$\mathbf{T}_{EP}^E = \mathbf{T}_{CP}^E + \mathbf{T}_{EC}^E = \mathbf{R}_{EC}\,\mathbf{T}_{CP}^C + \mathbf{T}_{EC}^E \tag{A3}$$

$$\dot{\mathbf{T}}_{EP}^E = \dot{\mathbf{R}}_{EC}\,\mathbf{T}_{CP}^C + \mathbf{R}_{EC}\,\dot{\mathbf{T}}_{CP}^C + \dot{\mathbf{T}}_{EC}^E = \mathbf{R}_{EC}\,\widehat{\boldsymbol{\omega}}_{EC}^C\,\mathbf{T}_{CP}^C + \mathbf{R}_{EC}\,\dot{\mathbf{T}}_{CP}^C + \dot{\mathbf{T}}_{EC}^E \tag{A4}$$

$$\mathbf{v}_{EP}^E = \mathbf{R}_{EC}\,\mathbf{v}_{CP}^C + \mathbf{v}_{EC}^E + \mathbf{R}_{EC}\,\widehat{\boldsymbol{\omega}}_{EC}^C\,\mathbf{T}_{CP}^C = \mathbf{v}_{CP}^E + \mathbf{v}_{EC}^E + \widehat{\boldsymbol{\omega}}_{EC}^E\,\mathbf{T}_{CP}^E \tag{A5}$$

$$\mathbf{v}_{EP}^C = \mathbf{R}_{CE}\,\mathbf{v}_{EP}^E = \mathbf{v}_{EC}^C + \mathbf{v}_{CP}^C + \widehat{\boldsymbol{\omega}}_{EC}^C\,\mathbf{T}_{CP}^C = \mathbf{0} \tag{A6}$$

Note that (A6) connects the point coordinates as viewed from the camera $\mathbf{T}_{CP}^C = \mathbf{p}^C$ and their time derivative $\mathbf{v}_{CP}^C = \dot{\mathbf{p}}^C$ with the twist $\xi_{EC}^C$ of the motion of the camera with respect to the Earth viewed in the $F_C$ or local frame, which is composed by its linear and angular velocities $\mathbf{v}_{EC}^C$ and $\boldsymbol{\omega}_{EC}^C$ [3].

$$\mathbf{v}_{CP}^C = \dot{\mathbf{p}}^C = -\mathbf{v}_{EC}^C - \widehat{\boldsymbol{\omega}}_{EC}^C\,\mathbf{T}_{CP}^C = -\mathbf{v}_{EC}^C - \widehat{\boldsymbol{\omega}}_{EC}^C\,\mathbf{p}^C \tag{A7}$$

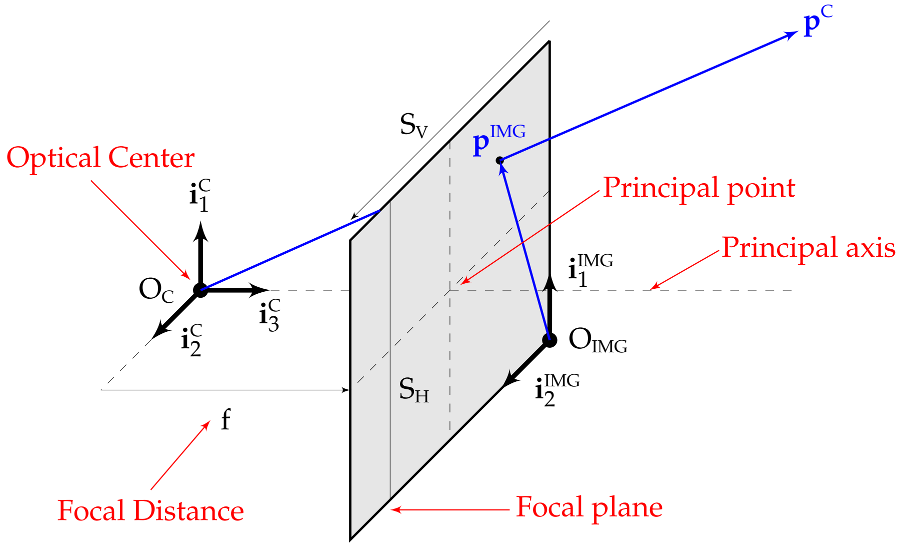
$\mathbf{p}^{\mathrm{C}}$
$\mathrm{S_V}$
$\mathbf{p}^{\mathrm{IMG}}$
Optical Center
$\mathbf{i}_1^{\mathrm{C}}$
Principal point
Principal axis
$\mathrm{O_C}$
$\mathbf{i}_1^{\mathrm{IMG}}$
$\mathbf{i}_3^{\mathrm{C}}$
$\mathbf{i}_2^{\mathrm{C}}$
$\mathrm{O_{IMG}}$
$\mathrm{S_H}$
$\mathbf{i}_2^{\mathrm{IMG}}$
f
Focal Distance
Focal plane

**Figure A1.** Frontal pinhole camera model.

The homogeneous camera coordinates $\bar{\mathbf{p}}^{\mathrm{C}}$ are defined as the ratio between the camera coordinates $\mathbf{p}^{\mathrm{C}}$ and its third coordinate or depth $\mathrm{p}_3^{\mathrm{C}}$, and represent an alternative view to $\mathbf{p}^{\mathrm{IMG}} = \Pi(\mathbf{p}^{\mathrm{C}})$ of how the point is projected in the image. Its time derivative is hence:

$$\bar{\mathbf{p}}^{\mathrm{C}} = \frac{\mathbf{p}^{\mathrm{C}}}{\mathrm{p}_3^{\mathrm{C}}} \longrightarrow \dot{\bar{\mathbf{p}}}^{\mathrm{C}} = \left[\dot{\bar{\mathrm{p}}}_1^{\mathrm{C}},\, \dot{\bar{\mathrm{p}}}_2^{\mathrm{C}},\, 1\right]^{\mathrm{T}} = \frac{\mathrm{p}_3^{\mathrm{C}}\,\dot{\mathbf{p}}^{\mathrm{C}} - \dot{\mathrm{p}}_3^{\mathrm{C}}\,\mathbf{p}^{\mathrm{C}}}{\mathrm{p}_3^{\mathrm{C}\,2}} \tag{A8}$$

Substituting both $\mathbf{p}^{\mathrm{C}}$ and $\dot{\mathrm{p}}_3^{\mathrm{C}}$ within (A7) into (A8), rearranging terms, and considering the (A1, A2) relationship between the image and the homogeneous camera coordinates, leads to the following expression for the *optical flow* [25] or variation of the point image coordinates:

$$\dot{\mathbf{p}}^{\mathrm{IMG}} = \mathbf{J}_{\mathrm{OF}}\left(\Pi\left(\mathbf{p}^{\mathrm{C}}\right)\right)\,\xi_{\mathrm{EC}}^{\mathrm{C}} = \mathrm{f}\begin{bmatrix} -\dfrac{1}{\mathrm{p}_3^{\mathrm{C}}} & 0 & \dfrac{\bar{\mathrm{p}}_1^{\mathrm{C}}}{\mathrm{p}_3^{\mathrm{C}}} & \bar{\mathrm{p}}_1^{\mathrm{C}}\,\bar{\mathrm{p}}_2^{\mathrm{C}} & -1-\bar{\mathrm{p}}_1^{\mathrm{C}\,2} & \bar{\mathrm{p}}_2^{\mathrm{C}} \\ 0 & -\dfrac{1}{\mathrm{p}_3^{\mathrm{C}}} & \dfrac{\bar{\mathrm{p}}_2^{\mathrm{C}}}{\mathrm{p}_3^{\mathrm{C}}} & 1+\bar{\mathrm{p}}_2^{\mathrm{C}\,2} & -\bar{\mathrm{p}}_1^{\mathrm{C}}\,\bar{\mathrm{p}}_2^{\mathrm{C}} & -\bar{\mathrm{p}}_1^{\mathrm{C}} \end{bmatrix}\begin{bmatrix}\mathbf{v}_{\mathrm{EC}}^{\mathrm{C}} \\ \boldsymbol{\omega}_{\mathrm{EC}}^{\mathrm{C}}\end{bmatrix} \tag{A9}$$

Considering that the twist $\xi$ is the time derivative of the transform vector $\tau$ [3], the *optical flow* $\mathbf{J}_{\mathrm{OF}}$ is defined as the derivative of the local frame ideal perspective projection of a point fixed to the spatial frame with respect to the $\mathbb{SE}(3)$ element $\mathcal{M}$ caused by a perturbation $\Delta\tau$ in its local tangent space:

$$\mathbf{J}_{\mathrm{OF}}\left(\Pi(\mathbf{g}_{\mathcal{M}}(\mathbf{p}))\right) = \lim_{\Delta\tau\to\mathbf{0}} \frac{\Pi(\mathbf{g}_{\mathcal{M}\oplus\Delta\tau}(\mathbf{p})) - \Pi(\mathbf{g}_{\mathcal{M}}(\mathbf{p}))}{\Delta\tau} \in \mathbb{R}^{2\mathrm{x}6} \tag{A10}$$

$$\Pi(\mathbf{g}_{\mathcal{M}\oplus\Delta\tau}(\mathbf{p})) \approx \Pi(\mathbf{g}_{\mathcal{M}}(\mathbf{p})) + \left[\mathbf{J}_{\mathrm{OF}}\left(\Pi(\mathbf{g}_{\mathcal{M}}(\mathbf{p}))\right)\Delta\tau\right] \in \mathbb{R}^2 \tag{A11}$$

Less formally, the optical flow Jacobian represents how the projection of a fixed point moves within the image as the camera pose varies. Note that the Jacobian only depends on the point camera (local) coordinates and the camera focal length, and that as all terms multiplying the linear twist component are divided by the image depth $\mathrm{p}_3^{\mathrm{C}}$, the effect on the image of a bigger linear velocity can not be distinguished from that of a smaller depth.

## Appendix B. Introduction to GNSS-Denied Navigation

The number, variety, and applications of UAVs (Unmanned Air Vehicles) have grown exponentially in the last few years, and the trend is expected to continue in the future [26,27].

This is particularly true in the case of low SWaP (Size, Weight, and Power) vehicles because their reduced cost makes them suitable for a wide range of applications, both civil and military. Ref [28] presents a comprehensive review of low SWaP UAV navigation systems and the problems they face, including the degradation or absence of GNSS (Global Navigation Satellite System) signals.

Aircraft navigation has traditionally relied on the measurements provided by accelerometers, gyroscopes, and magnetometers, incurring in an slow but unbounded position drift that could only be stopped by triangulation with the use of external navigation (radio) aids. More recently, the introduction of satellite navigation (GNSS) has completely removed the position drift and enabled autonomous inertial navigation in low SWaP platforms [29–31]. On the negative side, inertial navigation exhibits an extreme dependency on the availability of GNSS signals. If the signals are not present or can not be employed, inertial systems rely on dead reckoning, which results in position drift, with the aircraft slowly but steadily deviating from its intended route [32]. The availability of GNSS signals cannot be guaranteed; a throughout analysis of GNSS threats and reasons for signal degradation is presented in [33]. In GNSS-Denied conditions, the vehicle is unable to fly its intended route or even return to a safe recovery location, which leads to the uncontrolled loss of the airframe if the GNSS signals are not recovered before the aircraft runs out of fuel (or battery in case of electric vehicles).

The extreme dependency on GNSS availability is not only one of the main impediments for the introduction of autonomous UAVs in civil airspace, where it is not acceptable to have uncontrolled vehicles causing personal or material damage, but it also presents a significant drawback for military applications, as a single hull loss may compromise the onboard technology. At this time there are no comprehensive solutions to the operation of low SWaP autonomous UAVs in GNSS-Denied scenarios, although the use of onboard cameras seems to be one of the most promising routes. Bigger and more expensive UAVs, this is, with less stringent SWaP requirements, can rely to some degree on more accurate accelerometers and gyroscopes (at the expense of SWaP) and additional communications equipment to overcome this problem, but for most autonomous UAVs, the permanent loss of the GNSS signals is equivalent to losing the airframe in an uncontrolled way.

### *Appendix B.1. Possible Approaches to GNSS-Denied Navigation*

*Inertial navigation* employs the periodic readings provided by the Inertial Measurement Unit or IMU (accelerometers and gyroscopes) to estimate the pose of a moving object by means of dead reckoning or integration. On aircraft, inertial sensors are complemented by magnetometers and a barometer to add robustness to the inertial solution. Fixed wing aircraft are also equipped with a Pitot tube and air vanes required by their control system, although their measurements are usually not employed for navigation. Absolute references, such as those provided by navigation radio aids or GNSS receivers, are required to remove the position drift inherent to inertial navigation.

Low SWaP autonomous aircraft are too small to incorporate *navigation aid receivers*, which in any case are not available over vast regions of the Earth, exhibiting an extreme dependency on the availability of GNSS signals. A summary of the challenges of GNSS-Denied navigation and the research efforts intended to improve its performance is provided by [34]. There exist various approaches to mitigate this problem, with detailed reviews provided by [6,35]. Two promising techniques for completely eliminating the position drift are the use of *signals of opportunity* (existing signals originally intended for other purposes, such as those of television and cellular networks, can be employed to triangulate the aircraft position) [36–38], and *georegistration* (the position drift can be eliminated by matching landmarks or terrain features as viewed from the aircraft to preloaded data) [39–42], also known as *image registration*.

### *Appendix B.2. Visual Navigation*

*Visual Odometry* (VO) consists of employing the ground images generated by one or more onboard cameras without the use of prerecorded image databases or any other sensors, incrementally estimating the vehicle pose based on the changes that its motion induces on the images [43–45]. It requires sufficient illumination, dominance of static scene, enough texture, and scene overlap between consecutive images or frames. It can rely on a single camera (monocular vision), in which case the motion can only be recovered up to a scale factor, or on various cameras (stereo vision), where the differences among the simultaneous images taken with the different cameras are employed to determine the scale. It has been employed for navigation of ground robots, road vehicles, and multi-rotors flying both indoors and outdoors.

The incremental concatenation of relative poses results in a slow but unbounded pose drift, which can only be eliminated if aided by *Simultaneous Localization and Mapping* (SLAM) [46,47], a particular case of VO in which the map of the already viewed terrain is stored and employed for loop closure in case it is revisited by the vehicle during its motion. In this sense, VO only uses the map to improve the local consistency of the solution, while SLAM is more concerned with its global consistency [43]. The result is that SLAM is potentially more accurate, but also slower, computationally more expensive, and less robust.

Modern standalone algorithms, such as Semi Direct Visual Odometry (SVO) [4,5], Direct Sparse Odometry (DSO) [48], Large Scale Direct SLAM (LSD-SLAM) [49], and large scale feature based SLAM (ORB-SLAM) [50–52], are robust and exhibit a limited drift.

A typical VO algorithm includes steps to obtain the images, detect and extract its features, either match or track those features (VO algorithms can be divided into feature-based or matching methods and direct or tracking methods [45]), estimate the relative motion between consecutive frames, concatenate them to obtain the full camera pose trajectory, and finally perform some local optimization (bundle adjustment) [43].

### *Appendix B.3. Visual Inertial Navigation*

Estimating the aircraft pose based on both IMUs and cameras represents the most promising solution to GNSS-Denied navigation, in what is known as *Visual Inertial Odometry* (VIO) [53,54], which can also be combined with image registration to fully eliminate the remaining pose drift. Current VIO implementations are also primarily intended for ground robots, multi-rotors, and road vehicles, and, hence, rely exclusively on the vehicle IMU readings and the images taken by the onboard cameras, but do not use other sensors commonly found onboard fixed wing aircraft. VIO has matured significantly in the last few years, with detailed reviews available in [53–57].

VIO currently appears to represent the state of the art in GNSS-Denied navigation for low SWaP UAVs [28]. There exist several open source VIO packages, such as the Multi State Constraint Kalman Filter (MSCKF) [58], the Open Keyframe Visual Inertial SLAM (OKVIS) [59,60], the Robust Visual Inertial Odometry (ROVIO) [61], the monocular Visual Inertial Navigation System (VINS-Mono) [62], SVO combined with Multi-Sensor Fusion (MSF) [4,5,63,64], and SVO combined with Incremental Smoothing and Mapping (iSAM) [4,5,65,66]. All these open source pipelines are compared in [53], and their results when applied to the EuRoC MAV datasets [21] are discussed in [22]. There also exist various other published VIO pipelines with implementations that are not publicly available [67–73], and there are also others that remain fully proprietary.

The existing VIO schemes can be broadly grouped into two paradigms: *loosely coupled* pipelines process the measurements separately, resulting in independent visual and inertial pose estimations, which are then fused to get the final estimate; on the other hand, *tightly coupled* methods compute the final pose estimation directly from the tracked image features and the IMU outputs [53,54]. Tightly coupled approaches usually result in higher accuracy, as they use all the information available and take advantage of the IMU integration to predict the feature locations in the next frame. Loosely coupled methods, although less complex and more computationally efficient, lose information by decoupling the visual and inertial constraints, and are incapable of correcting the drift present in the visual estimator.

A different classification involves the number of images involved in each estimation [53,54,74], which is directly related with the resulting accuracy and computing demands. *Batch algorithms*, also known as *smoothers*, estimate multiple states simultaneously by solving a large non-linear optimization problem or bundle adjustment, resulting in the highest possible accuracy. Valid techniques to limit the required computing resources include the reliance on a subset of the available frames (known as *keyframes*), the separation of tracking and mapping into different threads, and the development of incremental smoothing techniques based on factor graphs [66]. Although employing all available states (*full smoothing*) is sometimes feasible for very short trajectories, most pipelines rely on *sliding window* or *fixed lag smoothing*, in which the optimization relies exclusively on the measurements associated to the last few keyframes, discarding both the old keyframes, as well as all other frames that have not been cataloged as keyframes. On the other hand, *filtering algorithms* restrict the estimation process to the latest state; they require less resources but suffer from permanently dropping all previous information and a much harder identification and removal of outliers, both of which lead to error accumulation or drift.

The success of any VIO approach relies on an accurate calibration of the pose and time offsets between the IMU and the camera [53,54]. Additional challenges applicable to all pipelines include the different working frequencies of IMUs and cameras, as well as the initialization requirements to bootstrap the algorithms.

## Appendix C. Semi-Direct Visual Odometry

*Semi-Direct Visual Odometry* (SVO) [4,5] is a publicly available advanced combination of feature-based and direct VO techniques primarily intended towards the navigation of land robots, road vehicles, and multi-rotors, holding various advantages in terms of accuracy and speed over traditional VO algorithms. By combining the best characteristics of both approaches while avoiding their weaknesses, it obtains high accuracy and robustness with a limited computational budget. This section provides a short summary of the SVO pipeline, although the interested reader should refer to [4,5] for a more detailed description; the pose optimization phase is however described in depth (Section 4), as it is the focus of the proposed modifications described in Section 5.

SVO initializes like a feature-based monocular method, requiring the height over the terrain to provide the scale (initialization), and using feature matching and RANSAC [75] based triangulation (initial homography) to obtain a first estimation of the terrain 3D position of the identified features. After initialization, the SVO pipeline for each new image can be divided into two different threads: the *mapping thread*, which generates terrain 3D points, and the *motion thread*, which estimates the camera motion (Figure A2).

Once initialized, the expensive feature detection process (mapping thread) that obtains the features does not occur in every frame but only once a sufficiently large motion has occurred since the last feature extraction. When processing each new frame, SVO initially behaves like a direct method, discarding the feature descriptors and skipping the matching process, and employing the luminosity values of small patches centered around every feature to (i) obtain a rough estimation of the camera pose (sparse image alignment, motion thread), followed by (ii) a relaxation of the epipolar restrictions to achieve a better estimation of the different features sub-pixel location in the new frame (feature alignment, motion thread), which introduces a reprojection residual that is exploited in the next steps. At this point, SVO once again behaves like a feature-based method, refining (iii) the camera pose (pose optimization, motion thread) and (iv) the terrain coordinates of the 3D points associated to each feature (structure optimization, motion thread) based on non-linear minimization of the reprojection error.

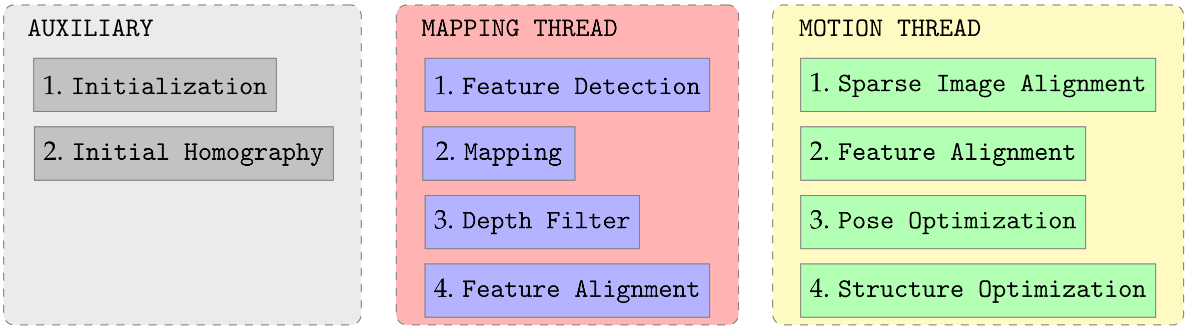

**Figure A2.** SVO threads and processes.

In this way, SVO is capable of obtaining the accuracy of direct methods at a very high computational speed, due to only extracting features in selected frames, avoiding (for the most part) robust algorithms when tracking features, and only reconstructing the structure sparsely. The accuracy of SVO improves if the pixel displacement between consecutive frames is reduced (high frame rate), which is generally possible as the computational expenses associated to each frame are low.

None of the motion thread four non-linear optimization processes listed above makes use of RANSAC, and pose optimization is the only one that employs a robust M-estimator [8,9] instead of the traditional mean or squared error estimator. This has profound benefits in terms of computational speed but leaves the whole process vulnerable to the presence of outliers in either the features terrain or image positions. To prevent this, once a feature is detected in a given frame (note that the extraction process obtains pixel coordinates, not terrain 3D ones), it is immediately assigned with a depth filter (mapping thread) initialized with a large enough uncertainty around the average depth in the scene; in each subsequent frame, the feature 3D position is estimated by reprojection and the depth filter uncertainty reduced. Once the feature depth filter has converged, the detected feature and its associated 3D point become a map candidate, which it is not yet employed in the motion thread optimizations required to estimate the camera pose. The feature alignment process is however applied in the background to the map candidates, and it is only after several successful reprojections that a candidate is upgraded to a map 3D point and, hence, allowed to influence the motion result. This two step verification process that requires depth filter convergence and various successful reprojections before a 3D point is employed in the (mostly) non-robust optimizations is key to prevent outliers from contaminating the solution and reducing its accuracy.

## References

1. Gallo, E. The SO(3) and SE(3) Lie Algebras of Rigid Body Rotations and Motions and their Application to Discrete Integration, Gradient Descent Optimization, and State Estimation. *arXiv* **2022**, arXiv:2205.12572v1.
2. Sola, J. Quaternion Kinematics for the Error-State Kalman Filter. *arXiv* **2017**, arXiv:1711.02508v1.
3. Sola, J.; Deray, J.; Atchuthan, D. A Micro Lie Theory for State Estimation in Robotics. *arXiv* **2018**, arXiv:1812.01537v9.
4. Forster, C.; Pizzoli, M.; Scaramuzza, D. SVO: Fast Semi-Direct Monocular Visual Odometry. In Proceedings of the IEEE International Conference on Robotics and Automation, Seattle, WA, USA, 26–30 May 2014. [CrossRef]
5. Forster, C.; Zhang, Z.; Gassner, M.; Werlberger, M.; Scaramuzza, D. SVO: Semidirect Visual Odometry for Monocular and Multicamera Systems. *IEEE Trans. Robot.* **2016**, 33, 249-265. [CrossRef]
6. Gallo, E.; Barrientos, A. Reduction of GNSS-Denied Inertial Navigation Errors for Fixed Wing Autonomous Unmanned Air Vehicles. *Aerosp. Sci. Technol.* **2022**, 120. [CrossRef]
7. Baker, S.; Matthews, I. Lucas-Kanade 20 Years On: A Unifying Framework. *Int. J. Comput. Vis.* **2004**, *56*, 221–255. [CrossRef]
8. Huber, P.J. *Robust Statistics*; John Wiley & Sons: New York, NY, USA, 1981.
9. Fox, J.; Weisberg, S. Robust Regression, 2013. Available online: http://users.stat.umn.edu/~sandy/courses/8053/handouts/robust.pdf (accessed on 10 January 2023).
10. Baker, S.; Gross, R.; Matthews, I. *Lucas-Kanade 20 Years On: A Unifying Framework: Part 4*; Technical Report CMU-RI-TR-04-14; Carnegie Mellon University: Cambridge, MA, USA, 2004.
11. Ogata, K. *Modern Control Engineering*, 4th ed.; Prentice Hall: 2002. Available online: https://scirp.org/reference/referencespapers.aspx?referenceid=123554 (accessed on 10 January 2023).

12. Skogestad, S.; Postlethwaite, I. *Multivariable Feedback Control: Analysis and Design*, 2nd ed.; John Wiley & Sons: New York, NY, USA, 2005.
13. Stevens, B.L.; Lewis, F.L. *Aircraft Control and Simulation*, 2nd ed.; John Wiley & Sons: New York, NY, USA, 2003.
14. Franklin, G.F.; Powell, J.D.; Workman, M. *Digital Control of Dynamic Systems*, 3rd ed.; Ellis-Kagle Press: Sunnyvale, CA, USA, 1998.
15. Gallo, E. Stochastic High Fidelity Simulation and Scenarios for Testing of Fixed Wing Autonomous GNSS-Denied Navigation Algorithms. *arXiv* **2021**, arXiv:2102.00883v3.
16. Gallo, E. High Fidelity Flight Simulation for an Autonomous Low SWaP Fixed Wing UAV in GNSS-Denied Conditions. C++ Open Source Code. 2020. Available online: https://github.com/edugallogithub/gnssdenied_flight_simulation (accessed on 10 January 2023).
17. Gallo, E.; Barrientos, A. Customizable Stochastic High Fidelity Model of the Sensors and Camera onboard a Fixed Wing Autonomous Aircraft. *Sensors* **2022**, *22*, 5518. [CrossRef]
18. osgEarth. Available online: http://osgearth.org (accessed on 10 January 2023).
19. Open Scene Graph. Available online: http://openscenegraph.org (accessed on 10 January 2023).
20. Ma, Y.; Soatto, S.; Kosecka, J.; Sastry, S.S. *An Invitation to 3-D Vision, From Images to Geometric Models*; Imaging, Vision, and Graphics; Springer: Berlin, Germany, 2001.
21. Burri, M.; Nikolic, J.; Gohl, P.; Schneider, T.; Rehder, J.; Omari, S.; Achtelik, M.W.; Siegwart, R. The EuRoC MAV Datasets. *IEEE Int. J. Robot. Res.* **2016**. [CrossRef]
22. Delmerico, J.; Scaramuzza, D. A Benchmark Comparison of Monocular Visual-Inertial Odometry Algorithms for Flying Robots. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, QLD, Australia, 21–25 May 2018; pp. 2502–2509. [CrossRef]
23. Gallo, E. Quasi Static Atmospheric Model for Aircraft Trajectory Prediction and Flight Simulation. *arXiv* **2021**, arXiv:2101.10744v1.
24. Hartley, R.; Zisserman, A. *Multiple View Geometry in Computer Vision*, 2nd ed.; Cambridge University Press: Cambridge, UK, 2003.
25. Heeger, D.J. Notes on Motion Estimation, 1998. Available online: https://www.cns.nyu.edu/csh/csh04/Articles/carandinifix.pdf (accessed on 10 January 2023).
26. Hassanalian, M.; Abdelkefi, A. Classifications, Applications, and Design Challenges of Drones: A Review. *Prog. Aerosp. Sci.* **2017**, *91*, 99–131. [CrossRef]
27. Shakhatreh, H.; Sawalmeh, A.H.; Al-Fuqaha, A.; Dou, Z.; Almaita, E.; Khalil, I.; Othman, N.S.; Khreishah, A.; Guizani, M. Unmanned Aerial Vehicles (UAVs): A Survey on Civil Applications and Key Research Challenges. *IEEE Access* **2019**, *7*, 48572–48634. [CrossRef]
28. Bijjahalli, S.; Sabatini, R.; Gardi, A. Advances in Intelligent and Autonomous Navigation Systems for Small UAS. *Prog. Aerosp. Sci.* **2020**, *115*, 100617. [CrossRef]
29. Farrell, J.A. *Aided Navigation, GPS with High Rate Sensors*; Electronic Engineering Series; McGraw-Hill: New York, NY, USA, 2008.
30. Groves, P.D. *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems*; GNSS Technology and Application Series; Artech House: Norwood, MA, USA, 2008.
31. Chatfield, A.B. *Fundamentals of High Accuracy Inertial Navigation*; American Institute of Aeronautics and Astronautics, Progress in Astronautics and Aeronautics: Reston, VA, USA, 1997. Volume 174,
32. Elbanhawi, M.; Mohamed, A.; Clothier, R.; Palmer, J.; Simic, M.; Watkins, S. Enabling Technologies for Autonomous MAV Operations. *Prog. Aerosp. Sci.* **2017**, *91*, 27–52. [CrossRef]
33. Sabatini, R.; Moore, T.; Ramasamy, S. Global Navigation Satellite Systems Performance Analysis and Augmentation Strategies in Aviation. *Prog. Aerosp. Sci.* **2017**, *95*, 45–98. [CrossRef]
34. Tippitt, C.; Schultz, A.; Procino, W. *Vehicle Navigation: Autonomy Through GPS-Enabled and GPS-Denied Environments*; State of the Art Report DSIAC-2020-1328; Defense Systems Information Analysis Center: Belcamp, MD, USA, 2020.
35. Gyagenda, N.; Hatilima, J.V.; Roth, H.; Zhmud, V. A Review of GNSS Independent UAV Navigation Techniques. *Robot. Auton. Syst.* **2022**, *152*, 104069. [CrossRef]
36. Kapoor, R.; Ramasamy, S.; Gardi, A.; Sabatini, R. UAV Navigation using Signals of Opportunity in Urban Environments: A Review. *Energy Procedia* **2017**, *110*, 377–383. [CrossRef]
37. Coluccia, A.; Ricciato, F.; Ricci, G. Positioning Based on Signals of Opportunity. *IEEE Commun. Lett.* **2014**, *18*, 356–359. [CrossRef]
38. Goh, S.T.; Abdelkhalik, O.; Zekavat, S.A. A Weighted Measurement Fusion Kalman Filter Implementation for UAV Navigation. *Aerosp. Sci. Technol.* **2013**, *28*, 315–323. [CrossRef]
39. Couturier, A.; Akhloufi, M.A. A Review on Absolute Visual Localization for UAV. *Robot. Auton. Syst.* **2020**, *135*, 103666. [CrossRef]
40. Goforth, H.; Lucey, S. GPS-Denied UAV Localization using Pre Existing Satellite Imagery. In Proceedings of the IEEE International Conference on Robotics and Automation, Montreal, QC, Canada, 20–24 May 2019. [CrossRef]
41. Ziaei, N. Geolocation of an Aircraft using Image Registration Coupling Modes for Autonomous Navigation. *arXiv* **2019**, arXiv:1909.02875v1.
42. Wang, T. Augmented UAS Navigation in GPS Denied Terrain Environments using Synthetic Vision. Ph.D. Thesis, Iowa State University, Ames, IA, USA, 2018. [CrossRef]
43. Scaramuzza, D.; Fraundorfer, F. Visual Odometry Part 1: The First 30 Years and Fundamentals. *IEEE Robot. Autom. Mag.* **2011**, *18*, 80–92. [CrossRef]

44. Fraundorfer, F.; Scaramuzza, D. Visual Odometry Part 2: Matching, Robustness, Optimization, and Applications. *IEEE Robot. Autom. Mag.* **2012**, *19*, 78–90. [CrossRef]
45. Scaramuzza, D. *Tutorial on Visual Odometry*; Robotics & Perception Group, University of Zurich: Zurich, Switzerland, 2012.
46. Scaramuzza, D. *Visual Odometry and SLAM: Past, Present, and the Robust Perception Age*; Robotics & Perception Group, University of Zurich: Zurich, Switzerland, 2017.
47. 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: Towards the Robust Perception Age. *IEEE Trans. Robot.* **2016**, *32*, 1309–1332. [CrossRef]
48. Engel, J.; Koltun, V.; Cremers, D. Direct Sparse Odometry. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *40*, 611–625. [CrossRef] [PubMed]
49. Engel, J.; Schops, T.; Cremers, D. LSD-SLAM: Large Scale Direct Monocular SLAM. In Proceedings of the Computer Vision—ECCV 2014: 13th European Conference, Zurich, Switzerland, 6–12 September 2014; pp. 834–849. [CrossRef]
50. Mur-Artal, R.; Montiel, J.M.M.; Tardos, J.D. ORB-SLAM: A Versatile and Accurate Monocular SLAM System. *IEEE Trans. Robot.* **2015**, *31*, 1147–1163. [CrossRef]
51. Mur-Artal, R.; Tardos, J.D. ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras. *IEEE Trans. Robot.* **2017**, *33*, 1255–1262. [CrossRef]
52. Mur-Artal, R. Real-Time Accurate Visual SLAM with Place Recognition. Ph.D. Thesis, University of Zaragoza, Zaragoza, Spain, 2017.
53. Scaramuzza, D.; Zhang, Z. Visual-Inertial Odometry of Aerial Robots. *arXiv.* **2019**, arXiv:1906.03289v2.
54. Huang, G. Visual-Inertial Navigation: A Concise Review. *arXiv.* **2019**, arXiv:1906.02650v1.
55. von Stumberg, L.; Usenko, V.; Cremers, D. Chapter 7—A Review and Quantitative Evaluation of Direct Visual Inertial Odometry. In *Multimodal Scene Understanding*; Yang, M.Y., Rosenhahn, B., Murino, V., Eds.; Academic Press: New York, NY, USA, 2019. [CrossRef]
56. Feng, X.; Jiang, Y.; Yang, X.; Du, M.; Li, X. Computer Vision Algorithms and Hardware Implementations: A Survey. *Integr. VLSI J.* **2019**, *69*, 309–320. [CrossRef]
57. Al-Kaff, A.; Martin, D.; Garcia, F.; de la Escalera, A.; Maria, J. Survey of Computer Vision Algorithms and Applications for Unmanned Aerial Vehicles. *Expert Syst. Appl.* **2017**, *92*, 447–463. [CrossRef]
58. Mourikis, A.I.; Roumeliotis, S.I. A Multi-State Constraint Kalman Filter for Vision-aided Inertial Navigation. In Proceedings of the IEEE International Conference on Robotics and Automation, Rome, Italy, 10–14 April; 2007; pp. 3565–3572. [CrossRef]
59. Leutenegger, S.; Furgale, P.; Rabaud, V.; Chli, M.; Konolige, K.; Siegwart, R. Keyframe Based Visual Inertial SLAM Using Nonlinear Optimization. In Proceedings of the International Conference on Robotics: Robotics: Science and Systems IX, Berlin, Germany, 24–28 June 2013. [CrossRef]
60. Leutenegger, S.; Lynen, S.; Bosse, M.; Siegwart, R.; Furgale, P. Keyframe Based Visual Inertial SLAM Using Nonlinear Optimization. *Int. J. Robot. Res.* **2015**, *34*, 314–334. [CrossRef]
61. 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]
62. 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]
63. 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, Tokyo, Japan, 3–7 November; 2013; pp. 3923–3929. [CrossRef]
64. 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.* **2015**, *33*, 431–450. [CrossRef]
65. Forster, C.; Carlone, L.; Dellaert, F.; Scaramuzza, D. On Manifold Pre Integration for Real Time Visual Inertial Odometry. *IEEE Trans. Robot.* **2017**, *33*, 1–21. [CrossRef]
66. Kaess, M.; Johannsson, H.; Roberts, R.; Ila, V.; Leonard, J.; Dellaert, F. iSAM2: Incremental Smoothing and Mapping Using the Bayes Tree. *Int. J. Robot. Res.* **2012**, *31*, 216–235. [CrossRef]
67. Mur-Artal, R.; Montiel, J.M.M. Visual Inertial Monocular SLAM with Map Reuse. *IEEE Robot. Autom. Lett.* **2017**, *2*, 796-803. [CrossRef]
68. Clark, R.; Wang, S.; Wen, H.; Markham, A.; Trigoni, N. VINet: Visual-Inertial Odometry as a Sequence-to-Sequence Learning Problem. *Proc. AAAI Conf. Artif. Intell.* **2017**. Available online: https://ojs.aaai.org/index.php/AAAI/article/view/11215 (accessed on 10 January 2023). [CrossRef]
69. 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 EEE International Conference on Robotics and Automation (ICRA), Singapore, 29 May–3 June; 2017; pp. 165–172. [CrossRef]
70. 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]

71. Solin, A.; Cortes, S.; Rahtu, E.; Kannala, J. PIVO: Probabilistic Inertial Visual Odometry for Occlusion Robust Navigation. In Proceedings of the 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), Lake Tahoe, NV, USA, 12–15 March 2018; pp. 616–625. [CrossRef]
72. Houben, S.; Quenzel, J.; Krombach, N.; Behnke, S. Efficient Multi Camera Visual Inertial SLAM for Micro Aerial Vehicles. In Proceedings of the 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Daejeon, Republic of Korea, 9–14 October; 2016; pp. 1616–1622. [CrossRef]
73. Eckenhoff, K.; Geneva, P.; Huang, G. Direct Visual Inertial Navigation with Analytical Preintegration. In Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 29 May 2017–3 June 2017; pp. 1429–1435. [CrossRef]
74. Strasdat, H.; Montiel, J.M.M.; Davison, A.J. Real Time Monocular SLAM: Why Filter? In Proceedings of the 2010 IEEE International Conference on Robotics and Automation, Anchorage, AK, USA, 3–7 May 2010; pp. 2657–2664. [CrossRef]
75. Fischler, M.A.; Bolles, R.C. RANSAC Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Commun. ACM* **1981**, *24*, 381–395. [CrossRef]

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.