Inertially Assisted Semi-Direct Visual Odometry for Fixed Wing Autonomous Unmanned Air Vehicles

Eduardo Gallo*†‡ and Antonio Barrientos §¶/uni2016

April 2022

Abstract

This article proposes a method to diminish the pose (position plus attitude) drift experienced by an SVO (Semi-Direct Visual Odometry) based 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 a GNSS (Global Navigation Satellite System) Denied inertial navigation system. The method is inspired in a PI (Proportional Integral) control system, in which the attitude, altitude, and rate of climb inertial outputs act as targets to ensure that the visual estimations do not deviate far 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 fixed wing low SWaP (Size, Weight, and Power) UAVs. Additionally, the IA-VNS can be considered as a virtual incremental position (ground velocity) sensor capable of providing observations to the inertial filter. 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 as well as to the quality of the onboard sensors on which the priors are based. The author releases the C++ implementation of both the navigation algorithms and the high fidelity simulation as open-source software [1].

Keywords: GNSS-Denied, GPS-Denied, visual navigation, autonomous navigation, VO, VIO, UAV, SVO, priors, non linear optimization

*Eduardo Gallo holds a MSc in Aerospace Engineering by the Polytechnic University of Madrid and has twenty-three years of experience working in aircraft trajectory prediction, modeling, and flight simulation. He is currently a Senior Trajectory Prediction and Aircraft Performance Engineer at Boeing Research & Technology Europe (BR&TE), although he is publishing this article in his individual capacity and time as part of his PhD thesis titled “Autonomous Unmanned Air Vehicle GNSS-Denied Navigation”, advised by Dr. Antonio Barrientos within the Centre for Automation and Robotics of the Polytechnic University of Madrid.

†Contact: edugallo@yahoo.com, e.gallo@alumnos.upm.es, https://orcid.org/0000-0002-7397-0425

‡Affiliation: Centro de Automática y Robótica, Universidad Politécnica de Madrid - Consejo Superior de Investigaciones Científicas. Centre for Automation and Robotics, Polytechnic University of Madrid.

§Antonio Barrientos received his MSc and PhD degrees from the Polytechnic University of Madrid in 1982 and 1986, respectively. In 2002 he obtained the MSc Degree in Biomedical Engineering by the National University of Distance Education. Since 1988 he is a professor at the Polytechnic University of Madrid, where he is presently a full professor teaching robotics and automatic control. He has worked in robotics for more than 30 years, developing industrial and service robots for different areas. He is a permanent staff member of the Center for Automation and Robotics (UPM-CSIC) where he is the head of the Applied Robotics Unit. He is a Senior Member of IEEE and former member of the steering committee of CEA (Spanish Automatics Committee - IFAC Spanish chapter).

¶Contact: antonio.barrientos@upm.es, https://orcid.org/0000-0003-1691-3907

/uni2016

Affiliation: Centro de Automática y Robótica, Universidad Politécnica de Madrid - Consejo Superior de Investigaciones Científicas. Centre for Automation and Robotics, Polytechnic University of Madrid.

1 Contact: antonio.barrientos@upm.es, https://orcid.org/0000-0003-1691-3907

Affiliation: Centro de Automática y Robótica, Universidad Politécnica de Madrid - Consejo Superior de Investigaciones Científicas. Centre for Automation and Robotics, Polytechnic University of Madrid.
Mathematical Notation

Any variable with a hat accent $\hat{\cdot}$ refers to its (inertial) estimated value, with a circular accent $\hat{\cdot}$ to its (visual) estimated value, and with a tilde $\tilde{\cdot}$ to its measured value. In the case of vectors, which are displayed in bold (e.g., $\mathbf{x}$), other employed symbols include the wide hat $\hat{\cdot}$, which refers to the skew-symmetric form, the bar $\overline{\cdot}$, which represents the vector homogeneous coordinates, and the double vertical bars $|\cdot|$ or $\|\cdot\|$ to refer to the norm. In the case of scalars, the vertical bars $|\cdot|$ refer to the absolute value. When employing quaternions and rigid body poses, which are also displayed in bold (e.g., $\mathbf{q}$ and $\mathbf{\zeta}$), 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. It is assumed that the reader is familiar with the Lie algebra of the special orthogonal group of $\mathbb{R}^3$, known as $\text{SO}(3)$, and that of the special euclidean group of $\mathbb{R}^3$, represented by $\text{SE}(3)$, 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 $\mathbf{\omega}$ for rotations, the transform vector $\mathbf{T}$ and twist $\mathbf{\xi}$ for motions). [2, 3, 4] are recommended as references.

Five different reference frames are used in this article: the ECEF frame ($F_E$), the NED frame ($F_N$), the ground frame ($F_X$), the body frame ($F_B$), and the camera frame ($F_C$). In addition, the two dimensional image frame ($F_{\text{IMG}}$) is also employed as introduced in appendix A. Superindexes are employed over vectors to specify the reference frame in which they are viewed (e.g., $\mathbf{v}^E$ refers to ground velocity viewed in $F_E$, while $\mathbf{v}^N$ is the same vector but viewed in $F_N$). Subindexes may be employed to clarify the meaning of the variable or vector, such as in $\mathbf{v}_{\text{TAS}}$. 

| Symbol | Description |
|--------|-------------|
| $\gamma_{\text{TAS}}$ | Aerodynamic path angle |
| $\delta$ | Error threshold |
| $\delta_{\text{CTR}}$ | Position of throttle and control surfaces |
| $\Delta$ | Estimation error, increment |
| $\Delta p$ | Atmospheric pressure offset |
| $\Delta T$ | Atmospheric temperature offset |
| $\theta$ | Body pitch angle |
| $\zeta$ | Pose, unit dual quaternion |
| $\mu$ | Mean or expected value |
| $\xi$ | Body bank angle |
| $\xi$ | Motion (SE(3)) velocity or twist |
| $\Pi$ | Camera projection |
| $\varphi_{\text{TUK}}$ | Tukey error function |
| $\sigma$ | Standard deviation |
| $\tau$ | Pose, transform vector |
| $\phi$ | Attitude, Euler angles |
| $\chi$ | Bearing |
| $\psi$ | Heading or body yaw angle |
| $\omega$ | Angular (SO(3)) velocity |
| $E_h$ | Altitude adjustment error |
| $E_{\text{PO}}$ | Pose optimization error |
| $E_{\text{q}}$ | Attitude adjustment error |
| $E_{\text{RP}}$ | Reprojection error |
| $E$ | Full sensor error |
| $f$ | Focal length |

| Symbol | Description |
|--------|-------------|
| $g$ | Lie group action |
| $h$ | Geometric altitude |
| $H_{\text{HEIGHT}}$ | Height, depth over terrain |
| $H_p$ | Pressure altitude |
| $I$ | Camera image |
| $J$ | Jacobian |
| $M$ | SE(3) Lie group element |
| $p$ | Point, feature |
| $q$ | Attitude, unit quaternion |
| $r$ | Attitude, rotation vector |
| $R$ | Attitude, rotation matrix |
| $S$ | Sensor dimension |
| $t$ | Time |
| $T_{\text{E,GDT}}$ | Geodetic coordinates |
| $v$ | Speed |
| $w_{\text{TUK}}$ | Tukey weight function |
| $x$ | Position |
| $x$ | Horizontal distance |
| $x_{\text{REF}}$ | Reference objectives |
| $x_{\text{EST}}$ | Inertial estimated trajectory |
| $x_{\text{IMG}}$ | Visual estimated trajectory |
| $x_{\text{SENS}}$ | Sensed trajectory |
| $x_{\text{TRUTH}}$ | Real trajectory |
for air velocity instead of the ground velocity $v$, 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. $v^2_N$ refers to the second component of $v^N$. 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, $\omega^B_N$ refers to the angular velocity from the $F_N$ frame to the $F_B$ frame viewed in $F_B$.

In addition, there exist various indexes that appear as subindexes: $j$ identifies an specific run within each Monte Carlo simulation, $n$ identifies a discrete time instant ($t_n$) for the inertial estimations, $s$ ($t_s$) refers to the sensor outputs, $i$ identifies an image or frame ($t_i$), 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.

**Acronyms**

| ACC  | ACCelerometers  | MAV | Micro Air Vehicle |
|------|-----------------|-----|-------------------|
| AOA  | Angle Of Attack | MEMS| Micro Machined Electromechanical System |
| AOS  | Angle Of Sideslip | MSCKF | Multi State Constraint Kalman Filter |
| BRIEF | Binary Robust Independent Elementary Features | MSF | Multi Sensor Fusion |
| DS   | DeSert          | MSL | Mean Sea Level    |
| DSO  | Direct Sparse Odometry | MX | MiX |
| ECEF | Earth Centered Earth Fixed | NED | North East Down |
| EKF  | Extended Kalman Filter | NSE | Navigation System Error |
| FAST | Features from Accelerated Segment Test | OAT | Outside Air Temperature |
| FM   | FarM            | OKVIS | Open Keyframe Visual Inertial SLAM |
| FR   | FoRest          | ORB | Oriented FAST and rotated BRIEF |
| FTE  | Flight Technical Error | OSP | Outside Static Pressure |
| GNSS | Global Navigation Satellite System | PI | Proportional Integral |
| GPS  | Global Positioning System | PR | PRaire |
| GYR  | GYRoscopes      | RANSAC | RANdom SAample Consensus |
| IA-VNS | Inertially Assisted Visual Navigation System | ROC | Rate Of Climb |
| IA-VNSE | Inertially Assisted Visual Navigation System Error | ROVIO | Robust Visual Inertial Odometry |
| ILS  | Instrument Landing System | SAR | Synthetic Aperture Radar |
| IMU  | Inertial Measurement Unit | SENS | SENSors |
| INS  | Inertial Navigation System | SLAM | Simultaneous Localization And Mapping |
| INSE | Inertial Navigation System Error | SLERP | Spherical linear interpolation |
| iSAM | Incremental Smoothing and Mapping | SoOP | Signals of OPPportunity |
| ISO  | International Organization for Standardization | SWaP | Size, Weight, and Power |
| LIDAR | Light Detection And Ranging | TAS | True Air Speed |
| LSD  | Large scale Semi Direct | TSE | Total System Error |
| MAG  | MAGnetometers | UAV | Unmanned Aerial Vehicle |
|     |                 | UR  | URban |
1 Introduction and Outline

The extreme dependency of an autonomous UAV (Unmanned Air Vehicle) navigation to the presence of GNSS (Global Navigation Satellite System) signals, without which it incurs in a slow but unavoidable position drift that may ultimately lead to the loss of the platform, is discussed in section 2. This dependency does not only constitute a significant obstacle for the widespread usage of these platforms in civil airspace, but also represents a threat to military missions. Section 2 also reviews the current approaches to GNSS-Denied navigation, with emphasis on those based on the images taken by onboard cameras. Figure 1 shows a schematic view of a generic navigation system, which relies on the sensor outputs \( \tilde{x}(t_s) \) and initial estimations \( \hat{x}_0 \) and provides its pose estimations \( \hat{x} = x_{EST} \) to the guidance and control system. Please refer to section 4 and table 1 for further details on the terms employed in the various flow diagrams below.

\[
\begin{align*}
\tilde{x}(t_s) &= x_{SENSED}(t_s) \\
\hat{x}_0 &= \hat{x}(t_n) = x_{EST}(t_n)
\end{align*}
\]

Figure 1: Navigation system flow diagram

This article focuses on the need to develop navigation systems capable of diminishing the position drift inherent to the flight in GNSS-Denied conditions of an autonomous fixed wing low SWaP (Size, Weight, and Power) aircraft so it has a higher probability of reaching the vicinity of a recovery point, from where it can be landed by remote control. A previous article by the same authors, [5], showed that it is possible to take advantage of sensors already present onboard fixed wing aircraft\(^1\), 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 (Inertial Navigation System) that results in bounded (no drift) estimations for attitude\(^2\), altitude\(^3\), and ground velocity\(^4\), as well as an unavoidable drift in horizontal position caused by integrating the ground velocity. Figure 2 graphically depicts that the INS inputs include all sensor measurements with the exception of the camera images 1.

\[
\begin{align*}
\tilde{x}(t_n) \setminus I(t_i) &= x_{SENSED}(t_n) \setminus I(t_i) \\
\hat{x}_0 &= \hat{x}(t_n) = x_{EST}(t_n)
\end{align*}
\]

Figure 2: INS flow diagram

This article focuses on visual navigation, this is, that based on the images periodically taken by a down facing camera rigidly attached to the aircraft structure. The objectives, novelty, and applications of the proposed

---

\(^1\)These include accelerometers, gyroscopes, magnetometers, Pitot tube, air vanes, thermometer, and barometer.

\(^2\)A bounded attitude estimation ensures that the aircraft can remain aloft in GNSS-Denied conditions for as long as there is fuel available.

\(^3\)The altitude estimation error depends on the change in atmospheric pressure offset \( \Delta p \) [6] from its value at the time the GNSS signals are lost, which is bounded by atmospheric physics.

\(^4\)The ground velocity 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.
approach are discussed in section 3. Section 4 introduces the stochastic high fidelity simulation employed to evaluate the navigation results by means of Monte Carlo executions of two scenarios\(^5\) representative of the challenges of GNSS-Denied navigation, which are described in detail in [7]. Its reading is highly recommended in order to understand the rest of this article, as is that of [8], which describes the different error sources modeled for each of the onboard sensors.

A VNS (Visual Navigation System) based on an advanced publicly available VO (Visual Odometry) algorithm known as SVO (Semi-Direct Visual Odometry) is introduced in section 5 and employed as a baseline. Visual navigation does not rely on absolute references and hence slowly accumulates error (drifts) not only in horizontal position, but also in attitude and altitude. Figure 3 shows that the VNS relies exclusively on the images \(I(t_i)\) as well as the initial visual estimations \(x_0\), as explained in section 5.

\[ I(t_i) \subset x_{\text{SENSED}}(t_i) \]
\[ x_0 \]

**Figure 3: VNS flow diagram**

Section 6 describes the proposed IA-VNS (Inertially Assisted VNS), which employs the INS attitude and altitude estimations to introduce priors into the pose estimation non linear optimizations present in SVO, resulting not only in significant improvements in the visual attitude and altitude estimations, but specially in a major reduction in horizontal position drift, which is a full order of magnitude lower than the drifts obtained by either the INS or the VNS by themselves. The IA-VNS is graphically depicted in figure 4.

\[ \hat{x}(t_i) = x_{\text{IMG}}(t_i) \]
\[ x_0 \]

**Figure 4: IA-VNS flow diagram**

The IA-VNS horizontal position estimations can be considered as the output of a virtual incremental horizontal displacement sensor\(^6\), which are fed back as periodic observations into the EKF, resulting in a VA-INS (Visually Assisted INS) with not only a major reduction in horizontal position drift compared to the stand alone INS, but also a significant improvement in attitude estimation accuracy.

The combined VINS (Visual Inertial Navigation System) is hence capable of combining the inertial and visual sensors and algorithms, employing the inertial attitude and altitude outputs to reduce the visual drifts, as well as the visual horizontal position output to improve the filter consistency and accuracy, resulting in an accurate and reliable navigation system capable of consistently leading the low SWaP aircraft to the vicinity of a recovery point even when faced with an extended flight duration in GNSS-Denied conditions coupled with weather changes and high turbulence levels. This article however focuses on the IA-VNS algorithms and its results, not on the fused VINS.

\(^5\)These are the same scenarios employed to evaluate the INS in [5].

\(^6\)This is synonymous with a ground velocity sensor.
The results of both the SVO based VNS as well as the proposed IA-VNS when applied to Monte Carlo simulations of the two scenarios representative of the challenges of GNSS-Denied conditions are described in section 7. The analysis of the results continues in section 8, which discusses the sensitivity of the IA-VNS 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. Similarly to the analysis performed in [5], section 9 discusses the sensitivity of the results to the quality or grade of the onboard sensors, on which the priors are based. By ensuring that the results obtained when employing sensors of inferior quality are qualitatively the same and quantitatively slightly inferior to those presented in section 7, the benefits of the proposed IA-VNS are safeguarded from possible errors introduced when modeling the performances of the various sensors. Last, the results are summarized for convenience in section 10, while section 11 provides a short conclusion.

2 GNSS-Denied Navigation

The number, variety, and applications of UAVs have grown exponentially in the last few years, and the trend is expected to continue in the future [9, 10]. This is particularly true in the case of low SWaP vehicles because their reduced cost makes them suitable for a wide range of applications, both civil and military. A significant percentage of these vehicles are capable of operating autonomously, this is, employing onboard computers to execute the previously uploaded mission objectives. With small variations, these platforms rely on a suite of sensors that provide data about the airframe state, a navigation algorithm to estimate the aircraft pose, and a control system that, based on the navigation outputs, adjusts the aircraft control mechanisms to successfully execute the preloaded mission.

This article focuses on fixed wing autonomous platforms, which are generally equipped with a GNSS receiver, accelerometers, gyroscopes, magnetometers, a Pitot tube, air vanes, and one or more cameras. The combination of triads of accelerometers and gyroscopes is known as the Inertial Measurement Unit (IMU). The errors introduced by all sensors grow significantly as their SWaP decreases, in particular in the case of the IMU. The recent introduction of solid state accelerometers and gyroscopes has dramatically improved the performance of low SWaP IMUs, with new models showing significant improvements when compared to those manufactured only a few years ago. The problem of noisy measurement readings is compounded in the case of low SWaP vehicles by the low mass of the platforms, which results in less inertia and hence more high frequency accelerations and rotations caused by the atmospheric turbulence. [11] presents a comprehensive review of low SWaP UAV navigation systems and the problems they face, including the degradation or absence of GNSS signals.

Aircraft navigation 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 [12, 13, 14].
On the negative side, low SWaP 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 velocity and position drift, with the aircraft slowly but steadily deviating from its intended route [15].

The availability of GNSS signals cannot be guaranteed by any means. In addition to the (unlikely) event of one or various GNSS satellites ceasing to broadcast (voluntarily or not), the GNSS signals can be accidentally blocked by nearby objects (mountains, buildings, trees), corrupted by their own reflections in those same objects, or maliciously interfered with by broadcasting a different signal in the same frequency (jamming or spoofing). A throughout analysis of GNSS threats and reasons for signal degradation is presented in [16]. Any of the above results in what is known as GNSS-Denied navigation. In that event, 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). Although there exist various GNSS threat monitoring and reporting systems [17], autonomous aircraft need to continuously evaluate the quality of the available signals to decide if they can rely on them or not. [18] proposes a continuous evaluation of GNSS accuracy based on its integration within the INS.

Note that although the term GNSS-Denied is more generic and implies the absence of valid signals from any navigation satellite system, GPS-Denied is more commonly employed in the literature, particularly in non scientific circles. Both terms are analogous in the context of navigation, as GPS-Denied does not imply the availability of navigation signals from non GPS satellite navigation systems; the widespread use of the term GPS-Denied can be attributed to GPS being the first widely available global navigation satellite system and, at least until recently, the system on which most GNSS receivers relied. This article however employs the term GNSS-Denied, which is considered more adequate and technically correct, representing the absence of navigation signals from any GNSS, GPS included.

The extreme dependency on GNSS availability is not only one of the main impediments for the introduction of small 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 IMUs (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.

Possible Approaches to GNSS-Denied Navigation

Inertial navigation employs the periodic readings provided by the IMU 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 [19]. There exist various approaches to mitigate this problem, with detailed reviews provided by [5, 20].

A promising approach for completely eliminating the position drift is to triangulate using existing signals originally

7 The most widely used radio aids for aircraft navigation are VHF Omnidirectional Radio Range (VOR), Non Directional Beacon (NDB), and Instrument Landing System (ILS).
intended for other purposes, such as those of television and cellular networks \cite{21, 22, 23}. Receivers for these signals, known as \textit{Signals of Opportunity} (SoOP), have a sufficiently low \textit{SWaP} to be mounted on most autonomous aircraft, although the quality and quantity of the available signals varies enormously depending on the location where the flight takes place.

\textit{Georegistration} matches landmarks or terrain features as scanned or imaged by vehicles to preloaded data \cite{19, 24}, and can work based on Synthetic Aperture Radar (SAR) \cite{25, 26, 27}, Light Detection and Ranging (LIDAR) \cite{28, 29, 30}, or camera visual systems, in what is known as \textit{image registration} \cite{31, 32, 33, 34}. While SAR suffers from being memory and computationally expensive and LIDAR is restricted for aviation purposes by its limited range, image registration is a potentially valid solution for completely removing the position drift in GNSS-Denied scenarios. Its main challenge is obtaining a high percentage of positive matches when supplied with onboard generated images that display the terrain not only at different altitudes and attitudes than those of the database images, but which are also obtained in different seasons, illumination, and weather conditions.

\textbf{Visual Odometry}

\textit{Visual Odometry} (VO) consists on employing the ground images generated by one or more onboard cameras (either visual or infrared) 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 \cite{35, 36, 37}. 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\footnote{This article relies on monocular vision exclusively.} (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 multirotors 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 \textit{Simultaneous Localization and Mapping} (SLAM) \cite{38, 39}, 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 \cite{35}. The result is that SLAM is potentially more accurate (only if a certain patch of terrain is visited more than once), but also slower, computationally more expensive, and less robust.

Modern stand-alone VO algorithms such as Semi Direct Visual Odometry (SVO) \cite{40, 41} and Direct Sparse Odometry (DSO) \cite{42} are robust and exhibit a limited drift, while Large scale Semi Dense SLAM (LSD-SLAM) \cite{43} and large scale feature based SLAM (ORB-SLAM)\footnote{ORB stands for Oriented \textit{FAST} and rotated \textit{BRIEF}, a type of blob feature.} \cite{44, 45, 46} may be more appropriate if the vehicle revisits an already imaged area.

A typical VO algorithm includes steps to obtain the images, detect and extract its features, either match or track those features, estimate the relative motion between consecutive frames, concatenate them to obtain the full camera pose trajectory, and finally perform some local optimization (bundle adjustment) \cite{35}. VO algorithms can be generally divided into two broad categories:

- \textit{Feature-based methods}, also known as \textit{matching methods}, rely on detecting features (image coordinates plus descriptor) in every frame and employing the descriptors to match features among consecutive frames. The camera pose and terrain structure are then recovered by epipolar geometry\footnote{Epipolar geometry is that existing between two views of the same object, and only depends on the camera parameters (which are the same for VO) and their relative pose \cite{47, 48}. When imaging a fixed terrain 3D point, its two images in both camera frames, the two optical centers, and the 3D point itself are located on the same plane, named the \textit{epipolar plane}.} and refined through minimization of the \textit{reprojection error}. This enables handling large inter-frame motions, but at the expense of slow detection algorithms, reliance on detection and matching thresholds, and the use of \textit{RANSAC} \cite{49} to identify and remove outliers. The estimated structure and motion relies exclusively on the average of the
pixel distance between the positions of each feature in different frames, so many features are required to average the errors and reduce drift [37].

- **Direct or tracking methods** avoid the expensive feature extraction and matching process and instead extract the scene structure and camera motion directly from the luminosity values of selected pixels in the different frames, based on the local gradient magnitude and direction. Direct methods in turn can be dense (if the minimization algorithm is applied to every pixel in the image), sparse (if features are extracted and then tracked instead of matched based on the photometric error at their location), or semi-dense if somewhere in between (as for example tracking all pixels in the vicinity of each feature). Minimizing the photometric or luminosity difference between consecutive images requires small inter-frame motions, and although more precise, the computation of the photometric error is more expensive than the reprojection error employed in feature-based methods. A final refinement of the structure and motion based on the reprojection error (bundle adjustment) is possible, although expensive [37].

### Visual Inertial Odometry

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) [50, 51], 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, multirotors, 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 [50, 51, 52, 53, 54].

There are currently several open source VIO packages, such as the Multi State Constraint Kalman Filter (MSCKF) [55], the Open Keyframe Visual Inertial SLAM (OKVIS) [56, 57], the Robust Visual Inertial Odometry (ROVIO) [58], the monocular Visual Inertial Navigation System (VINS-Mono) [59], SVO combined with Multi Sensor Fusion (MSF) [40, 41, 60, 61], and SVO combined with Incremental Smoothing and Mapping (iSAM) [40, 41, 62, 63]. All these open source pipelines are compared in [50], and their results when applied to the EuRoC MAV datasets [64] are discussed in [65]. There also exist various other published VIO pipelines with implementations that are not publicly available [66, 67, 68, 69, 70, 71, 72], and there are also others that remain fully proprietary.

The existing VIO schemes to fuse the visual and inertial measurements 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 [50, 51]. 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 [50, 51, 73], 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 nonlinear 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*)\(^{11}\), the separation of tracking and mapping into different threads, and the development of incremental smoothing techniques based on factor graphs [63]. 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

---

\(^{11}\) Common criteria to identify keyframes involve thresholds on the time, position, or pose increments from the previous keyframe.
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 [50, 51]. Additional challenges applicable to all pipelines include the different working frequencies of IMUs and cameras, as well the initialization requirements to bootstrap the algorithms.

Additional VIO pipelines are also being developed to take advantage of event cameras [74, 75], which measure per-pixel luminosity changes asynchronously instead of capturing images at a fixed rate. Event cameras hold very advantageous properties for visual odometry, including low latency, very high frequency, and excellent dynamic range, which make them well suited to deal with high speed motion and high dynamic range, which are problematic for traditional VO and VIO pipelines. On the negative side, new algorithms are required to cope with the sequence of asynchronous events they generate [50].

3 Objective, Novelty, and Application

Objective

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: an INS specifically designed for the flight without GNSS signals of an autonomous fixed wing low SWaP aircraft [5] and a VNS that relies on an advanced visual odometry pipeline such as SVO [40, 41]. Note that the INS makes use of all onboard sensors except the camera, while the VNS relies exclusively on the images.

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, 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. The resulting vastly improved horizontal position estimation outputs can be considered as the measurements of a virtual ground velocity or incremental displacement sensor, which can in turn be fed back as observations to the INS.

Novelty

The VIO solutions listed in section 2 are quite generic with respect to the platforms on which they are mounted, with most applications focused on ground vehicles, indoor robots, and multirotors, as well as with respect to the employed sensors, which are usually restricted to the IMU (gyroscopes and accelerometers) and one or more cameras. The existing solutions can also be classified into loosely or tightly coupled based on how they fuse the incoming data, and into smoothers or filters according to the number of images considered in each estimation.

This article focuses on an specific case (long distance GNSS-Denied turbulent flight of fixed wing low SWaP aircraft), and as such is simultaneously more restrictive but also takes advantage of the sensors already 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 [5], these are key to the obtainment of the bounded attitude and altitude INS outputs on which the proposed IA-VNS relies.
The proposed IA-VNS 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 system, in which the attitude, altitude, and rate of climb inertial outputs act as targets to ensure that the visual estimations do not deviate far 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.

When the IA-VNS is treated as a virtual ground velocity or incremental horizontal displacement sensor and its outputs (measurements) fed back into the inertial filter, the resulting VINS combines the high accuracy representative of tightly coupled smoothers with the lack of complexity and reduced computing requirements of filters and loosely coupled approaches. This difficulty in assigning the proposed fusion to either category is a testament to the novelty of the approach:

- The existence of two independent solutions (inertial and visual) hints to a loosely coupled approach, but in fact the proposed solution shares most traits with tightly coupled pipelines. First, no information is discarded because the two solutions are not independent, as each simultaneously feeds and is fed by the other. The visual estimations (IA-VNS) depend on the inertial attitude and altitude outputs, while the inertial filter (VA-INS) uses the visual incremental horizontal position estimations as observations. Unlike loosely coupled solutions, the visual estimations for attitude and altitude are not allowed to deviate from the inertial ones above a certain threshold, so they do not drift. Additionally, the two estimations are never fused together, as the final VINS output coincides with that of the VA-INS, as shown in figure 5.
- The proposed solution contains both a filter within the VA-INS (similar to the EKF described in [5]) and a keyframe based sliding window smoother within the SVO based VA-INS, obtaining the best properties of both categories. As proven in [5], a filter is sufficient to eliminate the drift in attitude and altitude; these two outputs can then be provided as priors to the SVO based smoother (section 6), whose nonlinear optimizations take into account all keyframes that share at least some terrain 3D points with the last frame viewed by the camera in order to reduce the horizontal position drift. The integrated solution nevertheless runs several times faster than real time because SVO relies on tracking features instead of matching, while employing different threads for tracking and mapping.

Application

The main application of this work is the development of a virtual sensor capable of providing the VINS of autonomous aircraft flying in GNSS-Denied conditions with periodic ground velocity or horizontal incremental displacement observations, which results in major reductions to the horizontal position drift in the absence of GNSS signals. 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 position and ground velocity observations provided by GNSS receivers. The proposed system is not a full replacement of GNSS as it provides incremental instead of absolute position observations, but as shown in the following sections, these can significantly increase the possibilities of the aircraft safely reaching the vicinity of the intended recovery location, where it can be landed by remote control.

4 High Fidelity Stochastic Simulation and Scenarios

To evaluate the performance of the proposed visual navigation algorithms, this article relies on Monte Carlo simulations consisting of one hundred runs each of two different scenarios based on the high fidelity stochastic flight simulation graphically depicted in figure 6. The simulation models the flight in varying weather and turbulent conditions of a low SwaP fixed wing piston engine autonomous UAV, and has been specifically developed to analyze the performances of different navigation systems in GNSS-Denied conditions. Described in detail in [7], the simulation models the influence on the resulting aircraft trajectory of many different factors, such as the guidance objectives that make up the mission, the atmospheric conditions, the wind field, the air turbulence, the local
perturbations to the Earth gravity and magnetic fields, the aircraft aerodynamic and propulsive performances, its onboard sensors and their error sources, the control system that moves the throttle and the aerodynamic controls so the trajectory conforms to the guidance objectives, and the navigation system that processes the data obtained by the sensors and feeds the control system. The simulation also considers the asymmetric flight conditions caused by the single engine torque and cross winds [76, 77], the influence of the fuel consumption on the aircraft mass and inertial properties, as well as the relative position and orientation of the gyroscopes, accelerometers, and camera with respect to the body frame and its origin, the aircraft center of mass. The open source C++ implementation of the high fidelity simulation is available in [1].

The simulation consists on two distinct processes. The first, represented by the yellow blocks on the right of figure 6, focuses on the physics of flight and the interaction between the aircraft and its surroundings that results in the real aircraft trajectory $x = x_{\text{TRUTH}}$; 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 $\tilde{x} = x_{\text{SENSED}}$, the navigation system in charge of filtering it to obtain the estimated trajectory $\hat{x} = x_{\text{EST}}$, the guidance system that converts the reference objectives $x_{\text{REF}}$ into the control targets $\delta_{\text{TGT}}$, and the control system that adjusts the position of the throttle and aerodynamic control surfaces $\delta_{\text{CNTR}}$ so the estimated trajectory $\hat{x}$ is as close as possible to the reference objectives $x_{\text{REF}}$. As shown in the figure, the two parts of the simulation are not independent. The total system error (TSE) or difference between $x$ and $x_{\text{REF}}$ is the combination of the navigation system error (NSE) or difference between $x$ and $\hat{x}$ and the flight technical error (FTE) or difference between $\hat{x}$ and $x_{\text{REF}}$. In the simulation, the real trajectory is integrated at 500 Hz, the navigation system and sensors operate at 100 Hz (with the exception of the camera, which works at 10 Hz), while the guidance and control systems work at 50 Hz [7] (table 1). Note that the inertially estimated trajectory $\hat{x} = x_{\text{EST}}$ shown in figures 1, 2, 4, 5, and 6 has a frequency 10 times faster than the visual trajectory $\circ x = x_{\text{IMG}}$ shown in figures 3, 4, and 5.

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

Table 1: Working frequencies of the different systems and trajectory representations

All components of the simulation have been modeled with as few simplifications as possible to increase the realism of the results, as explained in [7]. With the exception of the aircraft performances and its control system, which are deterministic, all other simulation components are treated as stochastic and hence vary from one execution to the next, enhancing the significance of the Monte Carlo simulation results. The representation of the different errors introduced by the onboard sensors is of particular interest for inertial navigation. A detailed description of the expressions that provide the measurement errors introduced by the various sensors is included in [8], as well as the sensor performances values employed in the simulation.
Camera

This section summarizes the camera model employed on the simulation, but the reader should refer to [8] for further details. The camera is positioned facing down and rigidly attached to the aircraft structure. Similarly to the case of the inertial sensors, the navigation system needs to cope with certain uncertainties in the exact position and attitude of the camera with respect to the aircraft structure.

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 [48], 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 [48], 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.

The camera model differs from all other onboard sensor models in that it does not return a sensed variable $\tilde{x}$ consisting of its real value $x$ plus a sensor error $E$, but instead generates a digital image simulating what a real camera would record based on the aircraft position and attitude as given by the actual or real trajectory $x = x_{\text{TRUTH}}$. When provided with the camera pose with respect to the Earth at equally time spaced intervals, the simulation is capable 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 [78] (which in turn relies on OpenSceneGraph [79]) 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 [8].

Scenarios and Terrain Types

Most VIO packages discussed in section 2 include in their release articles an evaluation when applied to the EuRoC Micro Air Vehicle (MAV) datasets [64], and so do independent articles such as [65]. These datasets contain perfectly synchronized stereo images, IMU measurements, and ground truth readings obtained with laser, for eleven 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 [7]. Most parameters comprising the scenario are defined stochastically, resulting in different values for every execution. Note that all results shown in sections 7, 8, and 9 are based on Monte Carlo simulations comprising one hundred 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 pre-defined 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 \((v_{\text{TAS}, \text{INI}}, H_{\text{P}, \text{INI}}, \chi_{\text{INI}}})\), their final values \((v_{\text{TAS, END}}, H_{\text{P, END}}, \chi_{\text{END}})\), and the time at which each of the three maneuvers is initiated\(^\text{12}\). The scenario lasts for \(t_{\text{END}} = 3800\, \text{s}\), while the GNSS signals are lost at \(t_{\text{GNSS}} = 100\, \text{s}\), which does not involve any loss of generality as the accuracy of the aircraft pose (attitude and position) estimation does not degrade with time when GNSS signals are available.

The wind field is also defined stochastically, as its two parameters (speed and bearing) are constant both at the beginning \((v_{\text{WIND, INI}}, \chi_{\text{WIND, INI}})\) and conclusion \((v_{\text{WIND, END}}, \chi_{\text{WIND, END}})\) 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 [7], 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 [6], as they are constant both at the start \((\Delta T_{\text{INI}}, \Delta p_{\text{INI}})\) and end \((\Delta T_{\text{END}}, \Delta p_{\text{END}})\) 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 \(t_{\text{GNSS}} = 100\, \text{s}\), but the scenario duration is shorter \((t_{\text{END}} = 500\, \text{s})\). The initial airspeed and pressure altitude \((v_{\text{TAS, INI}}, H_{\text{P, INI}})\) 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 \((v_{\text{WIND, INI}}, \chi_{\text{WIND, INI}}, \Delta T_{\text{INI}}, \Delta p_{\text{INI}})\) remain constant throughout scenario #2.

The type of terrain overflown by the aircraft has a significant influence in the performance of the visual navigation algorithms. In particular, the terrain texture (or lack of) and its elevation relief are the two most important characteristics in this regard. For this reason, each of the scenario #1 one hundred Monte Carlo simulation runs can be combined with four different initial geodetic coordinates, so the resulting flights occur over four different zones or types of terrain. The zones described below are intended to represent a wide array of terrain types\(^\text{13}\). Note that changing the area of the world were the flight takes place also modifies the gravitational and magnetic fields, but as all considered zones are far from the Earth magnetic poles, this does not imply any significant change in the performance of the inertial algorithms [5]. Images representative of each zone as viewed by the onboard camera are included below.

- The “desert” (DS) zone 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\(^\circ\). 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 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 nonexistent in the farmlands, where extracting features is often impossible.

\(^{12}\) Turns are executed with a bank angle of \(\gamma_{\text{TURN}} = \pm 10\, \text{°}\), altitude changes employ an aerodynamic path angle of \(\gamma_{\text{TAS, CLIMB}} = \pm 2\, \text{°}\), and airspeed modifications are automatically executed by the control system as set-point changes.

\(^{13}\) Alpine terrain with elevated vertical relief is not included as piston engines aircraft can not fly high enough for the previously described Earth Viewer tool to generate realistic images in this type of terrain.
• The “forest” (FR) zone 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 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 presents 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.

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 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 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.

The MX terrain zone is considered the most generic and hence employed to evaluate the visual algorithms in sections 7 and 9, as well as the inertial ones in [5]; the remaining terrain zones are only used to test the sensitivity
of the proposed algorithms to the terrain type, as described in section 8. 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 on 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.

5 Semi-Direct Visual Odometry

As introduced in section 2, Semi-Direct Visual Odometry (SVO) [40, 41] is a publicly available advanced combination of feature-based and direct VO techniques primarily intended towards the navigation of land robots, road vehicles, and multirotors, 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 [40, 41] for a more detailed description; the pose optimization phase is however described in depth, as it is the focus of the proposed modifications described in section 6.

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 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.

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 nonlinear minimization of the reprojection error.

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 nonlinear optimization processes listed above makes use of RANSAC, and pose optimization is the only one that employs a robust M-estimator [80, 81] 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.

**Pose Optimization**

As the focus of the proposed improvements within section 6, the pose optimization phase is the only one described in detail in this article. Graphically depicted in figure 14, pose optimization is executed for every new frame \( i \) and makes use of the different features and associated 3D points \( j \) present in the map to generate the visual estimation of the pose between the ECEF (\( F_e \)) and camera (\( F_c \)) frames \( \xi_{ECI} \).

The previous feature alignment process improved the projection of the feature points into the current image\(^{14}\) \( p_{ij}^e \) by minimizing the photometric error while ignoring their epipolar constraints, which enables fine tuning the camera pose \( \xi^*_{ECI} \) provided by the sparse image alignment process by minimizing the reprojection error of the different terrain 3D points \( p_j^e \) present in the map, which originate at the structure optimization process of the previous frame. This process is known as pose refinement or motion-only bundle adjustment \(^{[40]}\).

The reprojection error \( E_{RP} \) is defined in (1) as the sum for each 3D point of the norm of the difference between the camera projection \( \Pi \) of the ECEF coordinates \( p_j^e \) transformed into the camera frame\(^{15}\) and the image coordinates \( p_{ij}^\text{IMG} \).

\[
E_{RP} \left( \xi_{ECI} \right) = \sum_j \left\| \Pi \left( g_{\xi_{ECI}}^{-1} \left( p_j^e \right) \right) - p_{ij}^\text{IMG} \right\| \tag{1}
\]

This problem can be solved by means of an iterative Gauss-Newton gradient descent process \(^{[82]}\). Given an initial pose estimation \( \xi_{ECI0} \) taken from the sparse image alignment result \( \xi^*_{ECI} \), each iteration step \( l \) minimizes (2) and

\(^{14}\)The image frame \( F_{IMG} \) is a 2D reference frame centered at the sensor corner, as depicted in figure 32.

\(^{15}\)\( g_{\xi_{AB}} () \) represents the SE(3) transformation of a point from frame B to frame A, as described in \([2]\).
advances the estimated solution by means of (3) until the step diminution of the reprojection error falls below a
given threshold $|E_{RP,i+1} - E_{RP,i+1} < \delta_{RP}|$:

$$E_{RP,i+1} (\Delta \hat{\tau}_{ecil}^C) = \sum_j \left\| \Pi(g_{\hat{\tau}_{ecil}^C}^{-1} \Delta \hat{\tau}_{ecil}^C (p_j^i)) - p_{ij}^{IMG} \right\| = \sum_j \left\| E_{RP,i+1,j} (\Delta \hat{\tau}_{ecil}^C) \right\|$$

(2)

$$\hat{\tau}_{ecil,i+1} \leftarrow \hat{\tau}_{ecil,i} \circ \text{Exp}(\Delta \hat{\tau}_{ecil}^C) = \hat{\tau}_{ecil}^C + \Delta \hat{\tau}_{ecil}^C$$

(3)

Each $\Delta \hat{\tau}_{ecil}^C$ represents the update to the camera pose given by the transform vector [2, 4] viewed in the local
camera frame $F_{cil}$, which is obtained by following the process described in [82], and results in (4), where $J_{RP,ij}$
represents the optical flow obtained in appendix A:

$$\Delta \hat{\tau}_{ecil}^C = - \left( \sum_j J_{RP,ij}^T J_{RP,ij} \right)^{-1} \sum_j J_{RP,ij}^T \left[ \Pi(g_{\hat{\tau}_{ecil}^C}^{-1} (p_j^i)) - p_{ij}^{IMG} \right] \in \mathbb{R}^6$$

(4)

$$J_{RP,ij} = J_{\oplus M} \in \mathbb{R}^{2x6}$$

(5)

In order to protect the resulting pose from the possible presence of outliers in either the 3D points $p_j^i$ or their
image projection $p_{ij}^{IMG}$, 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 [80, 81]. The error to be minimized in each iteration step is then
given by (6), where the Tukey error function $\rho_{TUK}(x) = \rho_{TUK}(\sqrt{x})$ can be found in [81].

$$E_{RP,i+1} (\Delta \hat{\tau}_{ecil}^C) = \sum_j \rho_{TUK} \left( \left[ \Pi(g_{\hat{\tau}_{ecil}^C}^{-1} + \Delta \hat{\tau}_{ecil}^C (p_j^i)) - p_{ij}^{IMG} \right]^T \left[ \Pi(g_{\hat{\tau}_{ecil}^C}^{-1} + \Delta \hat{\tau}_{ecil}^C (p_j^i)) - p_{ij}^{IMG} \right] \right)$$

(6)

A similar process to that employed above leads to the solution (7), where the Tukey weight function $w_{TUK}(x)$ is
provided by [81]:

$$\Delta \hat{\tau}_{ecil}^C = - \left( \sum_j w_{TUK} \left( E_{RP,ij}^T E_{RP,ij} \right) J_{RP,ij}^T J_{RP,ij} \right)^{-1}$$

$$\left[ \sum_j w_{TUK} \left( E_{RP,ij}^T E_{RP,ij} \right) J_{RP,ij}^T E_{RP,ij} \right] \in \mathbb{R}^6$$

(7)

$$E_{RP,ij} = \Pi(g_{\hat{\tau}_{ecil}^C}^{-1} (p_j^i)) - p_{ij}^{IMG} \in \mathbb{R}^2$$

(8)

Initialization

The SVO based VNS depicted in figure 3 requires an estimation of the visual state $\hat{x}_0$ for initialization, this is,
the $\hat{\tau}_{ec0}$ pose between the ECEF and camera $F_{cil}$ frames. As in the case of the initialization of the GNSS-Denied
INS described in [5], it is possible to rely exclusively on the output of the GNSS-Based INS before the loss of the
GNSS signals. The only difference between both cases is that the GNSS-Denied INS filters are initialized at the exact
moment that the GNSS signals are lost ($t_{\text{GNSS}}$), while the IA-VNS shall be periodically reinitialized during
GNSS-Based conditions to minimize the inevitable accumulation of errors with time. Note however that during
the simulation scenarios described in section 4, the VNS is only initialized once at $t = 0$ s.

The initial pose $\hat{\tau}_{ec0}$ can be obtained from the geodetic coordinates provided by the position
filter [5] ($T_{0}^{E,\text{GDT}} = T_{\text{GNSS-based, last}}$), the body attitude estimated by the attitude filter [5]
($\hat{q}_{\text{BNN0}} = \hat{q}_{\text{GNSS-based, last}}$), and the pose between the body ($F_B$) and camera ($F_{cil}$) frames $\hat{\tau}_{ec0}$.

In addition to $\hat{\tau}_{ec0}$, monocular SVO also requires the initial distance (depth) of the camera optical center with
respect to the terrain ($H_{\text{ Height 0}}$) to provide the scale of the initial homography. The SVO initialization applies
feature detection and matching to two images, considers that all terrain 3D points viewed in both images are
located on a flat surface (which does not need to be horizontal), and then solves the corresponding homography
employing RANSAC to obtain the direction normal to the terrain plane, the rotation between the two camera frames, and the (up to scale) translation between the two camera frames.

These two images are not consecutive, as a sufficiently high optical flow between them is required to improve the conditioning of the homography solution and hence the consistency of the results. The homography hence is only solved once the aircraft pose has accumulated enough change, and as long as the scale is provided as an input, it provides the \( p_j^E \) ECEF terrain coordinates of the features, their \( p_{ij0}^{IMG} \) and \( p_{ij1}^{IMG} \) image coordinates in both frames (which become the first two keyframes of the SVO motion thread), and the relative pose \( \xi_{c0c1} \) between both keyframes. This relative pose is then combined with the input \( \xi_{EC0} \) absolute pose of the first keyframe to obtain the absolute pose of the second keyframe, \( \xi_{EC1} \). The initialization is then complete, and from that point on SVO proceeds without performing any additional feature matching.

A key point of the homography solution is that while the normal to the terrain plane and the relative rotation between the two keyframes are independent of the input depth that provides the scale, the translation between the two keyframes is proportional to it [83] (the homography solution provides the direction of that translation, which needs to be multiplied by the scale).

The homography scale, depth, height, or distance from the camera optical center of the first keyframe to the Earth surface (\( \hat{H}_{\text{HEIGHT}} \)) can hence be obtained by linear interpolation between any two arbitrary input heights to ensure that the homography estimation of the distance traveled by the camera between the two keyframes (\( \| T_{c0c1} \| \)), which is part of \( \xi_{c0c1} \), coincides with that estimated by the GNSS-Based inertial system (\( \| \hat{T}_{c0c1} \| \)), which is easily obtained from their respective estimated geodetic coordinates.

6 Proposed Inertially Assisted Semi-Direct Visual Odometry

Lacking any absolute references, the SVO based VNS described in section 5 gradually accumulates errors in each of the six dimensions of the vehicle pose \( \xi_{EN} \), as shown in section 7. If accurate estimations of attitude and altitude can be provided by an INS such as that described in [5], these can be employed to ensure that the visual estimations for body attitude and vertical position (\( \hat{q}_{NS} \) and \( \hat{h} \)) do not deviate in excess from their inertial counterparts \( q_{NS} \) and \( h \), so they too can be considered as bounded up to a certain degree. The modified SVO is referred to as IA-SVO (Inertially Assisted SVO).

Note that the inertial estimations should not replace the visual ones 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 too far 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 so it can take advantage of the INS outputs.

![Figure 15: Prior based pose optimization flow diagram](Image)
Rationale for the Introduction of Priors

The prior based pose optimization process starts by executing exactly the same pose optimization described in section 5, which seeks to obtain the camera pose $\hat{\zeta}_{\text{rci}}$ that minimizes the reprojection error $E_{\text{RIP}}$ (1). The iterative optimization results in a series of transform vector updates $\Delta\hat{\zeta}_{\text{rci}}$ (7), where $l$ 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.

The resulting camera pose, $\hat{\zeta}_{\text{rci}}$, is marked with the superindex ** to indicate that it is the reprojection only solution, resulting in $\hat{\zeta}_{\text{rci}}^{**}$. Its concatenation with the body to camera pose $\hat{\zeta}_{\text{rc}}$ obtained in calibration results in the reprojected body pose $\hat{\zeta}_{\text{rb}}^{**}$:

$$\hat{\zeta}_{\text{rb}}^{**} = \hat{\zeta}_{\text{rci}}^{**} \otimes \hat{\zeta}_{\text{rc}}$$

The reprojected ECEF body attitude $\hat{q}_{\text{rb}}^{**}$ and cartesian coordinates $\hat{T}_{\text{rb}}^{**}$ can then be readily obtained, which leads on one hand to the reprojected body attitude $\hat{\phi}_{\text{rb}}^{**}$, equivalent to the Euler angles $\hat{\psi}_{\text{rb}}^{**} = \left[ \hat{\psi}_{\text{rb}}^{**}, \hat{\rho}_{\text{rb}}^{**}, \hat{\chi}_{\text{rb}}^{**} \right]^T$, and on the other to the geodetic coordinates $\hat{T}_{\text{r}, \text{GDT}}^{**}$ and ECEF to NED rotation $\hat{q}_{\text{ne}}^{**}$. Considering a cartesian reference frame $F_X$ with axes parallel to those of $F_{NI}$ but with origin located on the WGS84 ellipsoid surface ($h = 0$) at the longitude and latitude given by $\hat{T}_{\text{r}, \text{GDT}}^{**}$, the displacement $\hat{T}_{\text{sni}}^{**} = \left[ 0, 0, -\hat{h}_{\text{ni}}^{**} \right]^T$ is easily derived from $\hat{T}_{\text{r}, \text{GDT}}^{**}$. It is also possible to derive the relative pose between $F_X$ and the ECEF frame, noted as $\hat{\zeta}_{\text{ne}}$. Two different situations (compatible with each other) can be envisioned:

- Let’s consider that the estimated body attitude provided by the INS ($\hat{\phi}_{\text{nbi}}$) [5] enables the IA-VNS to conclude that it would be preferred if the optimized body attitude were closer to a certain target attitude identified by the superindex oo, $\hat{\phi}_{\text{nbi}}^{oo}$, equivalent to the target Euler angles $\hat{\phi}_{\text{nbi}}^{oo} = \left[ \hat{\phi}_{\text{nbi}}^{oo}, \hat{\rho}_{\text{nbi}}^{oo}, \hat{\chi}_{\text{nbi}}^{oo} \right]^T$. Various alternatives to obtain the target attitude from the inertially estimated attitude $\hat{\phi}_{\text{nbi}}^{**}$ are provided below. The target body attitude is converted into a target camera attitude by means of $\hat{q}_{\text{nc}} \equiv \hat{\phi}_{\text{nc}}$ and the original reprojected ECEF to NED rotation $\hat{q}_{\text{rb}}^{**}$, incurring in a negligible error by not considering the attitude change of the NED frame as the iteration progresses.

$$\hat{q}_{\text{nc}} = \hat{q}_{\text{rb}}^{**} \otimes \hat{q}_{\text{nbi}}^{oo} \otimes \hat{q}_{\text{nc}}$$

Note that the objective is not for the resulting body attitude $\hat{\phi}_{\text{nbi}}$ to equal the target $\hat{\phi}_{\text{nbi}}^{oo}$, but to balance both objectives (minimization of the reprojection error of the different terrain 3D points and minimization of the incremental rotation between the resulting camera attitude $\hat{\zeta}_{\text{rci}}$ and the target $\hat{\zeta}_{\text{rci}}^{oo}$) without imposing any hard constraints on the pose (position plus attitude) of the aircraft.

- Let’s also consider that the estimated geometric altitude provided by the INS ($\hat{h}_{\text{a}}$) [5] enables the IA-VNS to conclude that it would be preferred if the optimized aircraft altitude were closer to a certain target altitude $\hat{h}_{\text{ti}}^{oo}$, equivalent to a target displacement $\hat{T}_{\text{sni}}^{oo} = \left[ 0, 0, -\hat{h}_{\text{ti}}^{oo} \right]^T$. The attainment of the target altitude from the inertially estimated altitude $\hat{h}_{\text{a}}$ is also discussed below. A negligible error occurs by not considering the motion of the F_X origin over the ellipsoid as the iteration progresses.

In this case the objective is not to set the resulting geodetic coordinates $\hat{T}_{\text{r}, \text{GDT}}^{i}$ to equal the target ones $\hat{T}_{\text{r}, \text{GDT}}^{oo}$, but to balance that objective with the minimization of the reprojection error of the different 3D points, without imposing any hard constraints on the aircraft pose.

Prior Based Pose Optimization

The following bullets describe how to optimize for the above two objectives independently or concurrently:

\footnote{Note that a single asterisk superindex \(<^*\) applied to a quaternion refers to its conjugate or inverse.}
• **Attitude Adjustment Optimization.** The attitude adjustment error \( E_q \) is defined in (11) as the norm of the Euclidean difference between rotation vectors corresponding to the camera attitude \( \hat{q}_{\text{eci}} \) and the target camera attitude \( \hat{q}_{\text{eci}}^{\infty} \) [2, 4]:

\[
E_q(\log(\hat{q}_{\text{eci}})) = E_q(\hat{r}_{\text{eci}}) = \left\| \log(\hat{q}_{\text{eci}}) - \log(\hat{q}_{\text{eci}}^{\infty}) \right\| = \left\| \hat{r}_{\text{eci}} - \hat{r}_{\text{eci}}^{\infty} \right\|
\]

Its minimization can be solved by means of an iterative Gauss-Newton gradient descent process [82]. Given an initial attitude estimation \( \hat{r}_{\text{eci},0} = \log(\hat{q}_{\text{eci},0}) \) taken from \( \xi_{\text{eci},0}^{*} = \xi_{\text{eci}}^{*} \), 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 \( E_{q,i,l} - E_{q,i,l+1} < \delta_q \):

\[
E_{q,i,l+1}(\Delta \hat{r}_{\text{eci}}^C) = \left\| \log(\hat{q}_{\text{eci}}) + \Delta \hat{r}_{\text{eci}}^C - \log(\hat{q}_{\text{eci}}^{\infty}) \right\|
\]

\[
\hat{r}_{\text{eci},l+1} \leftarrow \hat{r}_{\text{eci}} + \exp(\Delta \hat{r}_{\text{eci}}^C) = \hat{r}_{\text{eci}} + \Delta \hat{r}_{\text{eci}}^C
\]

Each \( \Delta \hat{r}_{\text{eci}}^C \) represents the update to the camera attitude given by the rotation vector viewed in the local camera frame \( F_{ci} \), which is obtained by following the process described in [82]\(^1\), and results in (14), where \( J_R \) is the \( SO(3) \) right Jacobian \( J_R(\hat{r}) \) provided by [2, 4]\(^2\).

\[
\Delta \hat{r}_{\text{eci}}^C = - \left[ J_{Ril}^T J_{Ril}^{-1} \right] J_{Ril} \left[ \hat{r}_{\text{eci}} - \hat{r}_{\text{eci}}^{\infty} \right]
\]

\[
J_{Ril}^T J_{Ril}^{-1} J_{Ril} \hat{r}_{\text{eci}} - \hat{r}_{\text{eci}}^{\infty} \in \mathbb{R}^3
\]

\[
J_{Ril} = J_R(\hat{r}_{\text{eci}}) = J_R(\log(\hat{q}_{\text{eci}})) \in \mathbb{R}^{3 \times 3}
\]

• **Altitude Adjustment Optimization.** The altitude adjustment error \( E_h \) is defined in (16) as the difference between the aircraft of mass altitude \( \hat{h} \) and the target altitude \( \hat{h}_{\text{f}}^{\infty} \). Note that its attainment relies on a double transformation of the \( F_c \) to \( F_{hi} \) position vector \( T_{ci}^{\text{f}} \):

\[
E_h(\hat{\xi}_{\text{eci}}) = T_{X_{hi},3}^X - T_{X_{hi},3}^{X_{ci}} = -\hat{h} + \hat{h}_{\text{f}}^{\infty} = \left[ g_{\xi} T_{X_{hi},3} \right] + \hat{h}_{\text{f}}^{\infty}
\]

Its minimization is also solved by means of an iterative Gauss-Newton gradient descent process [82]. Given an initial pose estimation \( \xi_{\text{eci},0}^{*} = \xi_{\text{eci}}^{*} \), each iteration step \( l \) minimizes (17) and advances the estimated solution by means of (18) until the step diminution of the altitude adjustment error falls below a given threshold \( E_{h,i,l} - E_{h,i,l+1} < \delta_h \). Note that the subindex \( 3 \) stands for the corresponding matrix \( 3 \) row.

\[
E_{h,i,l+1}(\Delta \hat{\xi}_{\text{eci}}^C) = \left[ g_{\xi} \right] \left[ \xi_{\text{eci},l}^{*} \right] + \hat{h}_{\text{f}}^{\infty}
\]

\[
\hat{\xi}_{\text{eci},l+1} \leftarrow \hat{\xi}_{\text{eci},l} + \exp(\Delta \hat{\xi}_{\text{eci}}^C) = \hat{\xi}_{\text{eci},l} + \Delta \hat{\xi}_{\text{eci}}^C
\]

Each \( \Delta \hat{\xi}_{\text{eci}}^C \) represents the update to the camera pose given by the transform vector viewed in the local camera frame \( F_{ci} \) [82], and results in (19), where \( J_{h,il} \) is the matrix product between the \( 3 \) row of the \( R_{xhi} \) rotation matrix and the \( J^+ \left[ E_{\text{eci}}(T)(P) \right] \) jacobian [2, 4], which represents the derivative of a transformed

\(^1\)Note that in this process the jacobian coincides with the identity matrix because the map \( f(\hat{r}_{\text{eci}}) = \hat{r}_{\text{eci}} \) coincides with the rotation vector itself.

\(^2\)Also include an expression for the right jacobian inverse \( J_{Ril}^{-1} = J_{Ril}^{-1}(\hat{r}) \).
point with respect to perturbations in the Euclidean tangent space (not on the curved manifold) that generates the motion.

\[
\Delta \hat{\theta}^C_{\text{ecil}} = - \left[ J_{h,il}^T J_{h,il} \right]^{-1} \left[ J_{h,il}^T \hat{T}^{x_{\text{Xni}},3} + \hat{R}_{x_{\text{Xni}},3}^n \right] \left[ g^C_{\text{ecil}} (\hat{T}^C_{\text{cmi}}) + \hat{h}^o_{il} \right] \\
= - \left[ J_{h,il}^T J_{h,il} \right]^{-1} \left[ J_{h,il}^T E_{h,il} \right] \in \mathbb{R}^6 \\
J_{h,il} = \hat{R}_{x_{\text{Xni}},3}^n + J^T \hat{\gamma}_{\text{ecil}} (\hat{T}^C_{\text{cmi}}) \in \mathbb{R}^{1 \times 6}
\]  

(19)

\[ J_{\text{RRU}}^{-1} = \left[ O_{3 \times 3} \quad J_{\text{RRU}}^{-1} \right] \in \mathbb{R}^{3 \times 6} \]  

(22)

The application of the iterative process described in [84] results in the following solution, which combines the contributions from the three different optimization targets:

\[
H_{\text{PO,il}} = \sum_j w_{TUK} \left( E_{\text{RP,ij}}^T E_{\text{RP,ij}} \right) J_{\text{RP,ij}}^T J_{\text{RP,ij}} \\
+ \hat{f}^2_q \left[ J_{\text{RRU}}^{-1} J_{\text{RRU}}^T \right] + \hat{f}^2_h \left[ J_{h,il}^T J_{h,il} \right] \in \mathbb{R}^{6 \times 6}
\]

\[
\Delta \hat{\theta}^C_{\text{ecil}} = - H_{\text{PO,il}} \left[ \sum_j w_{TUK} \left( E_{\text{RP,ij}}^T E_{\text{RP,ij}} \right) J_{\text{RP,ij}}^T E_{\text{RP,ij}} \right] \\
+ \hat{f} \cdot J_{\text{RRU}}^T E_{\text{R,il}} + \hat{f}_h \cdot J_{h,il}^T E_{h,il}
\]

\[
\hat{\psi}_{\text{ecil},i+1} \leftarrow \hat{\psi}_{\text{ecil}} \circ \text{Exp} (\Delta \hat{\theta}^C_{\text{ecil}}) = \hat{\psi}_{\text{ecil}} \oplus \Delta \hat{\theta}^C_{\text{ecil}}
\]

(23)

(24)

(25)

**PI Control Inspired Pose Adjustment Activation**

The previous paragraphs describe the possible attitude and altitude adjustments and their fusion with the default reprojection error minimization pose optimization algorithm, but they do not specify the conditions under which the adjustments are activated, the \( \hat{\phi}_{\text{Xni}} \) and \( \hat{h}^o_{il} \) targets are determined, or the obtainment of their \( f_q \) and \( f_h \) relative weights when applying the joint optimization. These parameters are determined below in four different cases: an attitude adjustment in which only pitch is controlled, an attitude adjustment in which both pitch and bank angles are controlled, a complete attitude adjustment, and an altitude adjustment.

- **Pitch Adjustment Activation.** The attitude adjustment described in (11) through (15) can be converted into a pitch only adjustment by forcing the yaw and bank angle targets to coincide in each optimization step \( i \) with the output of the reprojection only optimization:

\[
\hat{\psi}_{i}^{\text{oo}} = \hat{\psi}_{i}^{**} \\
\hat{\theta}_{i}^{\text{oo}} = \hat{\theta}_{i}^{**} + \Delta \hat{\theta}_{i}^{\text{oo}} \\
\hat{\xi}_{i}^{\text{oo}} = \hat{\xi}_{i}^{**}
\]

(26)

(27)

(28)

When activated as explained below, the new body pose target \( \hat{\psi}_{\text{Xni}}^{oo} \) only differs in one out of six dimensions (the pitch) from the reprojection only \( \text{ERP} \) optimum pose \( \hat{\psi}_{\text{Xni}}^{**} \), 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 (21) through (25) freely optimizes within $\mathbb{SE}(3)$ with six degrees of freedom to minimize the joint cost function $E_{PO}$ that not only considers the reprojection error, but also the resulting pitch target.

The pitch adjustment aims for the visual estimations for altitude $\hat{h}_i$ and pitch $\hat{\theta}_i$ (in this order) not to deviate in excess from their inertially estimated counterparts $\hat{h}_n$ and $\hat{\theta}_n$. It is inspired in a proportional integral (PI) control scheme [85, 86, 87, 88] as the geometric altitude adjustment error $\Delta h = \hat{h}_i - \hat{h}_n$ can be considered as the integral of the pitch adjustment error $\Delta \theta = \hat{\theta}_i - \hat{\theta}_n$ 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 addition, the adjustment also depends on the rate of climb (ROC) adjustment error $\Delta \text{ROC} = \text{ROC}_i - \text{ROC}_n$, 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 $\Delta h_{\text{LOW}}, \Delta \theta_{\text{LOW}},$ and $\Delta \text{ROC}_{\text{LOW}}$ below which the adjustments are not activated. These thresholds are arbitrary but have been set taking into account the INS accuracy and its sources of error, as described in [5]. If the absolute value of a certain adjustment error (difference between the visual and estimated estates) is above its threshold, the IA-VNS 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 state) may have the opposite sign, in which case the adjustment would be counterproductive.

| Variable                      | Value | Unit | Variable | Value | Unit |
|-------------------------------|-------|------|----------|-------|------|
| $\Delta h_{\text{LOW, GNSS-BASED}}$ | 8.0   | m    | $\Delta \theta_{\text{LOW}}$ | 0.2   | $^\circ$ |
| $\Delta h_{\text{LOW, GNSS-DENIED}}$ | 25.0  | m    | $\Delta \text{ROC}_{\text{LOW}}$ | 0.01  | m/s  |
| $\Delta t_{\text{GNSS}}$ | 1500.0 | s   | $\Delta \theta_1^{\text{MAX}}$ | 0.0005 | $^\circ$ |
| $t_{\text{GNSS}}$ | 100.0 | s    | $\Delta \theta_2^{\text{MAX}}$ | 0.0003 | $^\circ$ |

Table 2: Pitch adjustment settings

As an example, let’s consider a case in which the visual altitude $\hat{h}_i$ is significantly higher than the observed one $\hat{h}_n$, resulting in $|\Delta h| > \Delta h_{\text{LOW}}$; in this case the IA-VNS concludes that the aircraft is “high” and applies a negative pitch adjustment to slowly decrease the body pitch visual estimation $\hat{\theta}$ over many images, with these accumulating over time into a lower altitude $\hat{h}$ 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 ($|\Delta h| < \Delta h_{\text{LOW}}$), 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 $h_t$, not the the observed one $\hat{h}_n$), and a negative pitch adjustment would only exacerbate the situation. A similar reasoning applies for the adjustment pitch error, in which the IA-VNS reacts or not to correct perceived “nose-up” or “nose-down” visual estimations.

The applied thresholds are displayed in table 2, in which the altitude threshold $\Delta h_{\text{LOW}}$ varies per (29) taking into account that the INS vertical position accuracy varies depending on whether GNSS signals are available or not, and if they are not the uncertainty increases with time although it is bounded by atmospheric physics [5].

$$\Delta h_{\text{LOW}} = \Delta h_{\text{LOW, GNSS-BASED}} + \frac{t - t_{\text{GNSS}}}{\Delta t_{\text{GNSS}}} \Delta h_{\text{LOW, GNSS-DENIED}}$$  (29)

The $\hat{\theta}_1^{\text{MAX}}$ pitch target to be applied for each image is given by (27), where the obtainment of the pitch

---

19In this context, adjustment error is understood as the difference between the visual and inertial estimations.

20To avoid noise, this is smoothed over the last 100 images or 10 s.
The pitch adjustment due to altitude, $\Delta \hat{\theta}_h^o$, linearly varies between zero when the adjustment error is below the threshold $\Delta h_{LOW}$ to $\Delta \hat{\theta}_h^{1,MAX}$ when the error is twice the threshold, as shown in (31). The adjustment is bounded at this value to avoid destabilizing $\text{SVO}$ with pose adjustments that differ too much from their reprojection only optimum $\hat{\xi}_{\text{INS}}^*$ (9).

$$\Delta \hat{\theta}_h^o = \begin{cases} 0 & \text{when } |\Delta h| < \Delta h_{LOW} \\ - \text{sign}(\Delta h) \Delta \hat{\theta}_h^{1,MAX} \left( |\Delta h| - \Delta h_{LOW} \right) / \Delta h_{LOW} & \Delta h_{LOW} \leq |\Delta h| \leq 2 \cdot \Delta h_{LOW} \\ - \text{sign}(\Delta h) \Delta \hat{\theta}_h^{1,MAX} & \text{when } |\Delta h| > 2 \cdot \Delta h_{LOW} \end{cases}$$  

The pitch adjustment due to rate of climb, $\Delta \hat{\theta}_\theta^o$, works similarly but employing $\Delta \theta$ instead of $\Delta h$ and $\Delta \theta_{LOW}$ instead of $\Delta h_{LOW}$, while also relying on the same limit $\Delta \hat{\theta}_h^{1,MAX}$. In addition, $\Delta \hat{\theta}_\theta^o$ is set to zero if its sign differs from that of $\Delta \hat{\theta}_h^o$, and reduced so the combined effect of both targets does not exceed the limit $(|\Delta \hat{\theta}_h^o + \Delta \hat{\theta}_\theta^o| \leq \Delta \hat{\theta}_h^{1,MAX})$.

$$\Delta \hat{\theta}_\theta^o = \Delta \hat{\theta}_\theta^{*} + \Delta \hat{\theta}_\theta^{**}$$  

The pitch adjustment due to rate of climb, $\Delta \hat{\theta}_\theta^o$, works similarly but employing $\Delta \theta$ instead of $\Delta h$ and $\Delta \theta_{LOW}$ instead of $\Delta h_{LOW}$, and $\Delta \hat{\theta}_h^{1,MAX}$ instead of $\Delta \hat{\theta}_h^{1,MAX}$. Additionally, it is multiplied by the ratio between $\Delta \hat{\theta}_h^o$ and $\Delta \hat{\theta}_h^{1,MAX}$ to limit its effects when the altitude estimated error $\Delta h$ is small. This adjustment can act in both directions, imposing bigger pitch adjustments if the attitude error is increasing or lower one if it is already diminishing.

If activated, the weight value $f_q$ required for the (21) joint optimization is determined by imposing that the weighted attitude error $f_q \cdot E_q(\hat{\xi}_{\text{reci}0})$ coincides with the reprojection error $E_{\text{RP}}(\hat{\xi}_{\text{reci}})$ when evaluated before the first iteration, this is, it assigns the same weight to the two active components of the joint $E_{\text{PO}}(\hat{\xi}_{\text{reci}})$ cost function (21).

**Pitch and Bank Adjustment Activation.** The previous scheme can be modified to also make use of the observed body bank angle $\hat{\psi}_b$, within the framework established by the (11) through (15) attitude adjustment optimization:

$$\hat{\psi}_b^o = \hat{\psi}_b^{*}$$  

$$\hat{\theta}_\theta^o = \hat{\theta}_\theta^{*} + \Delta \hat{\theta}_\theta^o$$  

$$\hat{\xi}_i^o = \hat{\xi}_i^{*} + \Delta \hat{\xi}_i^o$$  

Although the new body pose target $\hat{\xi}_i^{o}$ only differs in two out of six dimensions (pitch and bank) from the optimum pose $\hat{\xi}_i^{*}$ obtained by minimizing the reprojection error exclusively, all six degrees of freedom are allowed to vary when minimizing the joint cost function.

| Variable   | Value | Unit |
|------------|-------|------|
| $\Delta \xi_{LOW}$ | 0.2 | $^\circ$ |
| $\Delta \xi_{1,MAX}$ | 0.0003 | $^\circ$ |

Table 3: Additional bank adjustment settings

The determination of the pitch adjustment $\Delta \hat{\psi}_b^o$ does not vary, and that of the bank adjustment $\Delta \hat{\xi}_i^o$ relies on a linear adjustment between two values similar to any of the three components of (30) explained in the previous section, but relying on the bank angle adjustment error $\Delta \xi = \hat{\xi}_i - \hat{\xi}_n$, as well as a $\Delta \xi_{LOW}$ threshold and $\Delta \hat{\xi}_i^{1,MAX}$ maximum adjustment whose values are provided in table 3. Note that the value of the $\Delta \xi_{LOW}$ threshold coincides with that of $\Delta \theta_{LOW}$ as the INS accuracy for both pitch and roll is similar according to [5].

It is important to remark that the combined pitch and bank adjustment activation is the one employed to show the results when applied to the IA-INS Monte Carlo simulations in sections 7 through 9.
• **Attitude Adjustment Activation.** The use of the observed yaw angle $\hat{\psi}$ is not recommended as the visual estimation $\hat{\psi}$ provided by the VNS (without any inertial inputs) is in general more accurate than its inertial counterpart $\psi$, as shown in section 7 (table 5). 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 the VNS, and on the other to the INS 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 [5].

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 observed attitude $q_{n^Bn}$ provided by the INS and the initial estimation $\hat{q}_{n^Bn}$ provided by the reprojection only pose optimization process. Its difference is given by $\Delta \hat{q}_{n^Bn} = q_{n^Bn} \odot \hat{q}_{n^Bn}$, where the superindex $B1$ indicates that it is viewed in the pose optimized body frame. This perturbation can be decoupled into a rotating direction and an angular displacement [2, 4], resulting in $\Delta \hat{\theta}_{n^Bn} = \hat{n}_{n^Bn} \odot \Delta \hat{\phi}_{n^Bn}$.

Let’s now consider that the IA-VNS decides to set an attitude target that differs by $\Delta \hat{\phi}_{\theta_{\theta\theta}}$ from its reprojection only solution $\hat{q}_{n^Bn}$, but rotating about the axis that leads to its inertial estimation $q_{n^Bn}$. The target attitude $\hat{q}_{n^Bn}$ can then be obtained by $SO(3)$ Spherical Linear Interpolation (SLERP) [2, 3], where $t = \Delta \hat{\phi}_{\theta_{\theta\theta}} / \Delta \hat{\phi}_{\theta_{\theta\theta}}$ is the ratio between the target rotation and the attitude error or estimated angular displacement:

$$
\hat{q}_{n^Bn} = \hat{q}_{n^Bn} \odot (\hat{q}_{n^Bn} \odot q_{n^Bn})^t
$$

• **Altitude Adjustment Activation.** The activation and target determination of the (16) through (20) altitude adjustment is similar to the attitude adjustments described above. When activated as explained below, the new body pose target $\hat{\xi}_{\hat{\theta}_{\theta\theta}}$ only differs in one out of six dimensions (the geometric altitude) from the optimum pose $\hat{\xi}_{\hat{\theta}_{\theta\theta}}$, obtained by minimizing the reprojection error exclusively, 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 freely optimizes within $SE(3)$ with six degrees of freedom to minimize the joint cost function that not only considers the reprojection error, but also the resulting altitude target.

$$
\hat{h}_{\hat{\theta}_{\theta\theta}} = \hat{h}_{\hat{\theta}_{\theta\theta}} + \Delta \hat{h}_{\hat{\theta}_{\theta\theta}}
$$

The objective is not for the visual altitude estimations $\hat{h}_i$ to closely track the inertial ones $\hat{h}_i$, but only to avoid excessive deviations, so there exists a lower threshold $\Delta h_{LOW}$, which is obtained by means of (29), below which the adjustment is not activated.

| Variable               | Value | Unit | Variable | Value | Unit |
|------------------------|-------|------|----------|-------|------|
| $\Delta h_{LOW,GNSS-\text{BASED}}$ | 8.0   | m    | $\Delta t_{\text{GNSS}}$ | 1500.0 | s    |
| $\Delta h_{LOW,GNSS-\text{DENIED}}$ | 25.0  | m    | $t_{\text{GNSS}}$      | 100.0  | s    |
| $\Delta h_{\hat{\theta}_{\theta\theta}}^{\text{MAX}}$ | 0.01  | m    |          |       |      |

Table 4: Altitude adjustment settings

The altitude adjustment $\Delta \hat{h}_{\hat{\theta}_{\theta\theta}}$ linearly varies between zero when the adjustment error is below the threshold $\Delta h_{\text{LOW}}$ to $\Delta h_{\hat{\theta}_{\theta\theta}}^{\text{MAX}}$ when the error is twice the threshold, as shown in (37). The adjustment is bounded at this value to avoid distabilizing $SVO$ with pose adjustments that differ too much from their reprojection only optimum $\hat{\xi}_{\hat{\theta}_{\theta\theta}}$ (9).

$$
\Delta \hat{h}_{\hat{\theta}_{\theta\theta}} = \begin{cases} 
0 & \text{when } |\Delta h| < \Delta h_{\text{LOW}} \\
-\text{sign}(\Delta h) \Delta \hat{h}_{\hat{\theta}_{\theta\theta}}^{\text{MAX}} (|\Delta h| - \Delta h_{\text{LOW}}) / \Delta h_{\text{LOW}} & \Delta h_{\text{LOW}} \leq |\Delta h| \leq 2 \cdot \Delta h_{\text{LOW}} \\
-\text{sign}(\Delta h) \Delta \hat{h}_{\hat{\theta}_{\theta\theta}}^{\text{MAX}} & \text{when } |\Delta h| > 2 \cdot \Delta h_{\text{LOW}}
\end{cases}
$$

(37)
If activated, the weight value $f_h$ required for the (21) joint optimization is determined by imposing that the weighted altitude error $f_h \cdot E_h(\hat{\xi}_{ECI})$ coincides with the reprojection error $E_{RP}(\hat{\xi}_{ECI})$ when evaluated before the first iteration, this is, it assigns the same weight to the two active components of the joint $E_{PO}(\hat{\xi}_{ECI})$ cost function (21).

Note that the altitude adjustment has been implemented and successfully tested, but extensive simulations have shown that the attitude adjustment (either the pitch only or the pitch and bank adjustment) is capable by itself of controlling the visually estimated altitude $\hat{h}$ by the accumulation over many images of changes in the aircraft path angle with less negative effects on the remaining components of the aircraft pose. The results described in section 7 hence do not rely on the altitude adjustment activation, but only on the previously described pitch and bank adjustment activation.

Additional Modifications to SVO

In addition to the PI inspired introduction of priors into the pose optimization phase of SVO, the proposed IA-VNS also includes various other modifications to the original SVO pipeline described in section 5:

- **Addition of Current Features to the Structure Optimization Phase.** The objective of the structure optimization phase (figure 13) is to adjust the ECEF coordinates of the terrain 3D points $p^E_j$ by minimization of the reprojection error into all those keyframes from which each 3D point is visible. For each 3D point $p_j^E$, the original algorithm relies on the pose $\xi_{ECLI}$ of every keyframe from where the 3D point is visible, plus the keyframes feature coordinates $p_{j,k}^{IMG}$ corresponding to that 3D point, but the current frame pose $\xi_{ECI}$ and associated feature coordinates $p_{j,k}^{IAG}$ are not employed in the optimization.

Such an algorithm hence results in Earth surface 3D points $p_j^E$ that do not take into consideration the previously obtained camera pose $\xi_{ECI}$. In the next cycle, corresponding to image $I_{i+1}$, these 3D points constitute a key input to the feature alignment and pose optimization processes, and as they are based on the keyframe poses and not that of $\xi_{ECI}$, so is the new pose $\xi_{ECI+1}$, reverting any adjustments gained by the adoption of attitude and altitude priors during pose optimization.

To ensure that the slight pose adjustments induced by the priors with each new image accumulate over time, the structure optimization algorithm needs to be modified. For each visible 3D point $p_j^E$, the algorithm shall rely not only on the pose and image feature coordinates of all keyframes from which the 3D point is visible, but also on those of the current frame (the one whose pose has just been optimized with the priors). In this way, the pose adjustments stick and are not reverted in the next cycle.

- **Inertial Pose Estimation.** The sparse image alignment phase (figure 13) constitutes the most delicate and error prone phase of the whole SVO pipeline. As a non linear optimization based on the minimization of the luminosity error in small patches located around the tracked features, whose objective is to obtain a camera pose approximation $\dot{\xi}_{ECI}$ that is later fine tuned by the pose optimization phase, its convergence depends on the proximity of the initial estimation $\dot{\xi}_{ECI_0}$ to the solution. The stand alone SVO algorithm relies on the previous image pose for the initial estimation of $\dot{\xi}_{ECI_0} = \dot{\xi}_{ECI-1}$.

This initial estimation works very well for a big majority of the images being processed, but suffers in those frames in which the optical flow induced by the camera motion is highest, which for the considered scenarios corresponds to the roll-in and roll-out maneuvers executed when initiating or concluding a turn. Short periods of intense turbulence can also generate initial pose estimations that differ too much from their true values and hence compromise the convergence of the non linear minimization. As the sparse image alignment output $\dot{\xi}_{ECI}$ is employed in the feature alignment and pose optimization phases (figure 13), where it is fine tuned to obtain the final pose estimation $\xi_{ECI}$, these two phases can be compromised by incorrect sparse image alignment estimations.

\[21\]The authors attempted to replace this initial estimation with an extrapolation to the current frame based on the relative visually estimated pose between the two previous frames. The results were mixed and hence the option discarded, as the accumulation of turbulence plus maneuvers may induce significant errors.
A more accurate and robust pose estimation can however be obtained by discarding the sparse image alignment algorithm and replacing it with a pose estimation based on the output of the previous frame pose optimization $\hat{\chi}_{ECi-1}$ concatenated with the various inertial incremental pose estimations generated by the INS since that last frame\textsuperscript{22}. To get an even more accurate estimation $\hat{\chi}_{ECi}$ in GNSS-Denied conditions, it is possible to employ a strategy such as that described in [5], in which only the attitude and air velocity inertial changes are utilized, but the wind velocity is frozen to its value at the time the GNSS signals were lost.

- **Automated Homography Scale Determination.** The initial height $\hat{H}_{HEIGHT0}$ estimation described in section 5 suffers from the fact that, although not contiguous, the two keyframes on which it relies are taken sufficiently close\textsuperscript{23} that the estimated distance between the two keyframes is significantly influenced by the GNSS receiver white noise. To fine tune the initial height estimation, it is possible to allow the SVO pipeline to continue running for an amount of time approximately an order of magnitude longer\textsuperscript{24} while storing all images for later use, until reaching a certain image, noted with # in the expressions below, which is sufficiently separated from the initial one so as to dilute the influence of the GNSS receiver white noise. After obtaining the visual ($\|T_{c0c#}\|$) and inertial ($\|\hat{T}_{c0c#}\|$) distances up to this frame as indicated in section 5, the initial height can be linearly approximated as follows:

$$\hat{H}_{HEIGHT} \approx \hat{H}_{HEIGHT} \frac{\|T_{c0c#}\|}{\|\hat{T}_{c0c#}\|}$$  \hspace{1cm} (38)

This equivalence can also be used to estimate a correction to the initial body pitch angle based on the differences in geometric altitude between the visual and inertial estimations:

$$\Delta \hat{\theta} = - \arctan \frac{\hat{h}_# - \hat{h}_#}{\|T_{c0c#}\|}$$  \hspace{1cm} (39)

At this point, all the previously stored images are processed again, but this time using $\hat{H}_{HEIGHT}$ as the depth for the homography initialization, and correcting the initial body attitude with $\Delta \hat{\theta}$. Once the execution catches up with the images being generated in real time, it is not necessary to store any more images, and each image is processed as soon as it is generated\textsuperscript{25}.

### 7 Visual Navigation System Error in GNSS-Denied Conditions

This section presents the results obtained with the proposed IA-VNS (section 6) when executing Monte Carlo simulations of the two GNSS-Denied scenarios over the MX terrain type, each consisting of one hundred executions. They are compared with the results obtained with the stand alone VNS that relies on the original SVO algorithm (section 5), and with those of the INS described in [5]. Detailed descriptions of the different modules of the high fidelity simulation together with the definition of the two scenarios are provided in [7] and summarized in section 4. [8] explains the expressions employed to model the measurement errors introduced by the different sensors, as well as their “baseline” values employed in the simulation.

The navigation system error (NSE) incurred by the proposed navigation systems (and accordingly denoted as INSE, VNSE, and IA-VNSE) when estimating the value of the different variables in GNSS-Denied conditions is evaluated in the tables below according to the aggregated final state metrics defined in [7]. They include the mean, standard deviation, and maximum values taking as inputs the NSE of the last or final estimation within each simulation.

\textsuperscript{22}In the simulation, as the inertial system operates at 100 Hz and the visual one at 10 Hz, the ten last inertial incremental poses shall be used.

\textsuperscript{23}The first two keyframes are usually separated by one or two seconds.

\textsuperscript{24}Twenty seconds are employed in the simulations.

\textsuperscript{25}Note that the IA-VNS is initialized when GNSS signals are present, so its outputs are not yet required for guidance and control purposes.
run. When depicting the NSE, the figures below show the variation with time of the time aggregated metrics (mean and standard deviation) of the one hundred different executions of each scenario; these are also formally defined in [7].

Before analyzing the simulation results, it is necessary to remark the following:

- The results obtained with the INS under the same two GNSS-Denied scenarios are described in detail in [5]. The attitude INSE does not drift and is bounded by the quality of the onboard sensors, ensuring the aircraft can remain aloft for as long as there is fuel available. The vertical position and ground velocity INSEs are also bounded by atmospheric physics and do not drift; their estimation errors depend on the atmospheric pressure offset and wind field changes that occur since the GNSS signals are lost. On the other hand, the horizontal position INSE drifts as a consequence of integrating the ground velocity without absolute references. Of the six SE(3) degrees of freedom (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.

- 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 as described below, their attitude and altitude estimations shall only be understood as a means to provide an accurate horizontal position estimation, which represents their only real objective.

- As described below, the SVO based VNS drifts in all six degrees of freedom. 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 dimensions. The resulting horizontal position IA-VNSE is just a fraction of the INSE, and hence can be fed back as additional observations into the inertial filter, converting it into a fully integrated VINS; this feed back loop however is not described in this article. 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.

### Body Attitude Estimation

Table 5 shows the aggregated final state metrics for the three Euler angles representing the body attitude (yaw $\psi$, pitch $\theta$, roll $\zeta$), together with the norm of the rotation vector between the real body attitude $q_{NB}$ and its estimations, $\hat{q}_{NB}$ by the INS and $\hat{q}_{NB}$ by the IA-VNS, when applied to both scenarios over the MX terrain type. The errors hence can be formally written as $\|\Delta \hat{r}^a\| = \|q_{NB} \odot \hat{q}_{NB}\|$ or $\|\Delta \hat{r}^b\| = \|q_{NB} \odot q_{NB}\|$ [2].

| Scenario | MX | INSE | VNSE | IA-VNSE |
|----------|----|------|------|---------|
|          | [°] (t\textsubscript{END}) | $\Delta \hat{\psi}$ | $\Delta \hat{\theta}$ | $\Delta \hat{\zeta}$ | $\|\Delta \hat{r}^a\|$ | $\Delta \hat{\psi}$ | $\Delta \hat{\theta}$ | $\Delta \hat{\zeta}$ | $\|\Delta \hat{r}^b\|$ | $\Delta \hat{\psi}$ | $\Delta \hat{\theta}$ | $\Delta \hat{\zeta}$ | $\|\Delta \hat{r}^b\|$ |
| #1       | 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 |
| #2       | 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 |

Table 5: Aggregated final body attitude INSE, VNSE, and IA-VNSE (100 runs)

The variation with of the mean and standard deviation of the attitude error for the two MX scenarios is depicted in figures 16 and 17, respectively. In the case of scenario #1, figure 16 also depicts the worst IA-VNSE seed (red) among the one hundred executions; this is not shown for scenario #2 in figure 17 because the significant oscillations hinder the visualization of the remaining lines. In addition, figure 18 shows the estimation of each...
individual Euler angle for the MX scenario #1 exclusively. Note that the INSE (blue lines), which shows no drift with time in either scenario, coincides with that depicted in [5].

With respect to the VNSE (pink lines), most of the scenario #1 error is incurred during the turn maneuver at the beginning of the scenario (refer to $t_{\text{TURN}}$ within [7]), 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 and even slightly decreases 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 [7]) 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 that 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 initial roll-in and final roll-out maneuvers, where the optical flow is highest and hence more difficult to track by SVO.

The IA-VNSE results (orange lines) show that the introduction of priors in section 6 works as intended and there exists a clear benefit for the use of an IA-VNS when compared to the stand alone VNS. In spite of IA-VNSE values at the beginning of both scenarios that are nearly double those of the VNSE, caused by the initial pitch adjustment (39) required to improve the fit between the homography output and the initial inertial estimations, the balance between both errors quickly flips as soon as the aircraft starts maneuvering, resulting in IA-VNSE.

---

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).
values significantly lower than those of the VNSE for the rest 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.

Qualitatively, the biggest difference between the three estimations resides in the nature of the errors. While the attitude INSE is bounded by the quality of the sensors and the navigation algorithms [5], drift is present in the VNS and IA-VNS estimations. The drift resulting from the Monte Carlo simulations may be small, and so is the attitude estimation error \( ||\Delta \hat{\theta}|| \), but more challenging conditions with more drastic maneuvers and a less idealized image generation process than that described in section 4 may generate additional drift.

Focusing now on the quantitative results shown in table 5, 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 \( ||\Delta \hat{\theta}|| \) and \( ||\Delta \hat{\phi}|| \), 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 IA-VNS succeeds when employing the observed pitch and bank angles \((\hat{\theta}, \hat{\xi})\), whose errors are bounded, to limit the drift of their visual counterparts \((\tilde{\theta}, \tilde{\xi})\), as \( \sigma_{\text{ENDB}} \) and \( \sigma_{\text{ENDB}}^{28} \) are significantly lower for the IA-VNS than for the VNS. Remarkably, this is achieved with no degradation in the body yaw angle, as \( \sigma_{\text{END}} \) 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 \( \hat{\psi} \). The following sections show how the benefits of an improved fit between the displayed terrain and the adjusted pose also improve the horizontal position estimation.

It is interesting to remark that although qualitatively the IA-VNS estimations of the three Euler angles as well as that of the total attitude error drift and are hence worse than the bounded estimations obtained by the INS, quantitatively the IA-VNS estimations are only worse than the INS ones in the case of the pitch, bank, and total attitude, while the visual yaw angle estimation is consistently superior to the inertial one (refer to the top of figure 18).

In the case of a real life scenario based on a more realistic image generation process than that described in section 4 and [7], 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 6 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 tables 2 and 3 would be required. The IA-VNS is hence more resilient against high drift values than the VNS.

**Vertical Position Estimation**

Table 6 shows the aggregated final state metrics for vertical position or altitude \((\Delta \hat{h} = \hat{h} - h, \Delta \hat{\text{h}} = \hat{\text{h}} - h)\) when applied to both scenarios over the MX terrain type. The results can be considered unbiased or zero mean in all six cases (two scenarios and three estimation methods), as the mean of the final altitude error \( \mu_{\text{ENDBh}} \) is always significantly lower than both the standard deviation \( \sigma_{\text{ENDBh}} \) or the maximum value \( \zeta_{\text{ENDB|h|}} \). With respect to the

\[28\] 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.
Figure 18: Body Euler angles INSE, VNSE, and IA-VNSE for scenario #1 MX (100 runs)

The VNSE estimation of the geometric altitude (pink 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 the scenario #1 because of its much longer duration.

A small percentage of this drift can be attributed to the slow accumulation of error inherent to the SVO motion.

Table 6: Aggregated final vertical position INSE, VNSE, and IA-VNSE (100 runs)
thread algorithms introduced in section 5, 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 above). 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 20. As explained in the attitude analysis 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.

The benefits of introducing priors to limit the differences between the visual and inertial altitude estimations are reflected in the IA-VNSE (orange 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 \( \sigma_{END} \) 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 19 (red 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 \( t_{END} \). As soon as the differences between the visual pitch, bank, or altitude estimations (\( \hat{\theta}, \hat{\xi}, \hat{h} \)) and their inertial counterparts (\( \hat{\theta}, \hat{\xi}, \hat{h} \)) exceed certain limits (section 6), the attitude adjustment comes into play and slowly adjusts the aircraft pitch to prevent the visual altitude from deviating too far 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.

Significantly better altitude estimation errors (closer to the inertial ones) could be obtained if more aggressive settings were employed for \( \Delta \hat{\theta}_1^{MAX} \) and \( \Delta \hat{\theta}_2^{MAX} \) 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 setting 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 6. Higher $\Delta \tilde{\theta}_1^{\text{MAX}}$ and $\Delta \tilde{\theta}_2^{\text{MAX}}$ 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 $\text{IA-VNSE}$. This is indeed the real objective for the introduction of the priors, as then the visual estimation can be more accurate than the inertial one and hence be employed as a virtual sensor to provide ground velocity or incremental horizontal displacement observations back to the inertial filter.

The $\text{IA-VNS}$ altitude estimation improvements over those of the $\text{VNS}$ are not only quantitative. Figure 19 shows no increment in $\sigma_{\Delta h}^\text{n}$ (orange lines) in the second half of scenario #1 (once on average the deviation has activated the attitude adjustment feature). The altitude estimation by the $\text{IA-VNS}$ can hence also be described as bounded and driftless, which represents a qualitative and not only quantitative improvement over that of the $\text{VNS}$. The bounds are obviously bigger for the $\text{IA-VNS}$ than for the $\text{INS}$. In the case of scenario #2, figure 20 shows a slow but steady $\sigma_{\Delta h}^\text{n}$ 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. Note however that it is only on aggregated terms where drift is eliminated and the visual error bounded to the $\text{INSE}$, as a limited drift may exist for an individual trajectory, although on the long term the visual estimation tends to return to values not too far from the inertial ones.

**Horizontal Position Estimation**

The horizontal position estimation capabilities of the $\text{INS}$, $\text{VNS}$, and $\text{IA-VNS}$ share the fact that all of them exhibit and unrestrained drift or growth with time, as shown in figures 21 and 22. The errors obtained at the end of both scenarios are shown in table 7, following the same scheme as in previous sections. The approximately linear $\text{INS}$ drift appears when integrating the bounded ground velocity errors; the visual estimation drifts (both $\text{VNS}$ and $\text{IA-VNS}$) 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 supplying the aircraft height over the terrain during the initial homography.
Table 7: Aggregated final horizontal position INSE, VNSE, and IA-VNSE (100 runs)

| Scenario | INSE | VNSE | IA-VNSE |
|----------|------|------|---------|
|          | Distance | $\Delta_{\text{HOR END}}^x$ | $\Delta_{\text{HOR END}}^x$ | $\Delta_{\text{HOR END}}^x$ |
| (#1)     | [m]    | [m]  | [%]     | [m]    | [%]  | [m]    | [%]  |
| Mean     | 107873 | 7276 | 7.10    | 4179   | 3.82 | 488    | 0.46 |
| Std      | 19756  | 4880 | 5.69    | 3308   | 2.73 | 350    | 0.31 |
| Max      | 172842 | 25288| 32.38   | 21924  | 14.22| 1957   | 1.48 |
| (#2)     | [m]    | [m]  | [%]     | [m]    | [%]  | [m]    | [%]  |
| Mean     | 14198  | 216  | 1.52    | 251    | 1.77 | 33     | 0.23 |
| Std      | 1176   | 119  | 0.86    | 210    | 1.48 | 26     | 0.18 |
| Max      | 18253  | 586  | 4.38    | 954    | 7.08 | 130    | 0.98 |

In the case of the VNS (pink 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 4 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 22 (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 $t_{\text{GNSS}}$ [5].

The IA-VNS (orange lines) results in major horizontal position estimation improvements over the VNS. The final horizontal position error mean $\mu_{\text{END HORIZ}}$ 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 $\sigma_{\text{END HORIZ}}$ 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.

It is interesting to remark how the prior based pose optimization described in section 6, 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.

The major improvement in the horizontal position estimation capabilities enabled by the introduction of priors within SVO imply that the the IA-VNS can be considered as a virtual ground velocity or incremental horizontal displacement sensor, whose outputs can be periodically fed back to the inertial filter to create a fully integrated VINS that successfully fuses the inertial and visual algorithms.

8 Influence of Terrain Type

This section analyzes the influence of the terrain type overflown by the aircraft on the capability of the IA-VNS to estimate the aircraft pose when executing Monte Carlo simulations of both scenarios, each composed of 100 runs. 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. Note however that each terrain type has an associated initial location and applicable Earth gravitational and magnetic fields, which slightly impact the inertial estimations as described in [5].

The SVO algorithms described in section 5 can not operate unless the feature detector can periodically locate features in the various keyframes, and also requires the depth filter to correctly estimate the 3D terrain coordinates of each feature. The texture and elevation relief of the overflown terrain hence play a key role in the ability of the VNS and IA-VNS to estimate the aircraft pose. The use of different types of terrain is intended to provide a more complete validation of the proposed visual algorithms by evaluating their performance when flying over regions that differ in both their texture and vertical relief. Refer to section 4 for a detailed description of the terrain types employed in each of the two scenarios.

Body Attitude Estimation

Tables 8 and 9 show the total attitude error aggregated final state metrics for both scenarios and all terrain types. The corresponding time aggregated metrics are depicted in figures 23 and 24.

Note that the MX columns of tables 8 and 9 coincide with the results shown in table 5.

Note that the blue lines of figure 23 (top and bottom plots) are the same as the blue and orange lines of figure 16, respectively.

Note that the blue lines of figure 24 (top and bottom plots) are the same as the blue and orange lines of figure 17, respectively.
Table 8: Influence of terrain type on final body attitude INSE and IA-VNSE for scenario #1 (100 runs)

These results indicate that although the terrain type impacts the attitude estimation accuracy, its influence is minor when compared with other factors, such as the specific turbulence and sensor errors employed in each execution. This assessment is true for the inertially assisted visual estimations, for which scenario #1 runs 35, 79, and 92 are particularly negative independently of the terrain type, as well as runs 16 and 91 for scenario #2, but also for the inertial estimations, for which #1 runs 23 and 43 as well as #2 executions 16 and 73 are generally the ones with worst results.

Figure 23: Influence of terrain type on body attitude INSE and IA-VNSE for scenario #1 (100 runs)

The influence of the terrain type on the attitude INSE is indirect and caused by the Earth magnetic field intensity and inclination. Each terrain type is associated with a certain Earth location and hence a different magnetic field, which impacts the tracking of the aircraft attitude by the inertial filter, specially its bearing. With respect to the IA-VNSE, the influence of the terrain type is very low given the significant variations in texture and vertical relief among the various zones evaluated. Certain trends can however be identified:

Table 9: Influence of terrain type on final body attitude INSE and IA-VNSE for scenario #2 (100 runs)
• The MX zone provides the best IA-VNSE results for scenario #1 and third best for #2, although the differences are slim. This is attributed to its benign characteristics, with abundant texture and continuous but slight terrain altitude variations.

• The FR zone results in significantly worse IA-VNS attitude estimations than all other terrain types at the beginning of both scenarios, which is attributed to a less accurate initial homography and scale determination caused by an initial location that differs the most from the flat terrain assumed by the homography. The FR initial disadvantage slowly dissipates (note that it is still present at $t_{\text{END}} = 500$ s, when the alternate scenario concludes) because of its favorable conditions (abundant features and significant but never extreme vertical relief).

• The FM zone provides the worst IA-VNSE results for scenario #1, as well as second worst for #2, which is expected as it contains by far the least favorable terrain characteristics from a visual odometry point of view, with very scarce features and virtually no vertical relief to assist in the estimation of the aircraft pose.

• The DS zone a priori looks like an excellent location for visual navigation, with lots of features and continuous terrain altitude variations, but it is penalized by the presence of a few very steep mountain ranges, which challenge the depth filter when overflown, resulting in error growth not present in the other terrain zones.

• The UR and PR zones are only available for scenario #2, resulting in the lowest IA-VNS errors because of their abundant texture. The lack of altitude relief would likely create some drift on the long term, but it is not a problem with the continuous turns and short duration of this scenario.

**Vertical Position Estimation**

Tables 10 and 11 show the vertical position error aggregated final state metrics for both scenarios and all terrain types. The corresponding time aggregated metrics are depicted in figures 25 and 26.

---

32 Note that the MX columns of tables 10 and 11 coincide with the results shown in table 6.

33 Note that the blue lines of figure 25 (top and bottom plots) are the same as the blue and orange lines of figure 19, respectively.

34 Note that the blue lines of figure 26 (top and bottom plots) are the same as the blue and orange lines of figure 20, respectively.
Table 10: Influence of terrain type on final vertical position INSE and IA-VNSE for scenario #1 (100 runs)

The INSE results are virtually the same for all terrain types because the inertial vertical position error only depends on the pressure offset change from the time the GNSS signals are lost and on the ionospheric effects, but not on the terrain type and associated Earth location [5].

The altitude IA-VNSE obtained with the various terrain types reinforces the conclusions obtained when discussing the attitude error. They are more evident for scenario #1 than for #2 because of its longer duration and the fact that the aircraft does not fly over previously visited areas. The lowest errors, not only for the standard deviation $\sigma_{\Delta h}$ but also for the mean $\mu_{\Delta h}$, are obtained with the FR and DS terrain types, which can be attributed not only to their abundant texture that results in an elevated number of features at all times, but specially to the continuous variation in the terrain altitude (refer to their descriptions in section 4), which in most cases can be correctly estimated by the depth filter\textsuperscript{35} but that results in beneficial differences in the distance to the camera of the different terrain 3D points visible in a given image.

\textsuperscript{35}The exception are some small but very steep areas in the DS terrain type.
The FM zone lies on the opposite side, with significantly worse vertical position estimations for both scenarios, due to the scarcity of texture in the prevalent rectangular plots of farmland, and the near absence of any vertical relief. The SVO pipeline hence not only operates with a lower number of features, which increases the negative effects of any outliers, but also employs features that are very similar to each other from a luminosity point of view and whose terrain 3D altitude is virtually the same, increasing the chances of inaccuracies in the optimizations.

The MX terrain used as the baseline lies between the two extremes in both texture and vertical relief, and its results hence are also intermediate. In the case of the UR and PR types, their abundance of features but lack of vertical relief provides average alternate scenario results, but they would likely be worse for longer flights if it were possible to locate big enough locations with these types of terrain.

Horizontal Position Estimation

Tables 12 and 13 show the horizontal position error aggregated final state metrics for both scenarios and all terrain types. The corresponding time aggregated metrics are depicted in figures 27 and 28.

| Scenario #1 Zone | MX | FR | FM | DS |
|------------------|----|----|----|----|
| **INSE Δx_HOR [m, °] (t_{END})** | mean | 7276 | 7.10 | 7269 | 7.09 | 7276 | 7.10 | 7278 | 7.10 |
| | std | 4880 | 5.69 | 4865 | 5.67 | 4873 | 5.67 | 4892 | 5.71 |
| | max | 25288 | 32.38 | 25281 | 32.19 | 25268 | 32.28 | 25331 | 32.65 |

| **IA-VNSE Δx_HOR [m, °] (t_{END})** | 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 |

Table 12: Influence of terrain type on final horizontal position INSE and IA-VNSE for scenario #1 (100 runs)

The horizontal position INSE results are very similar for all terrain types as the error originates when integrating the horizontal velocity, whose errors are caused by the wind field variations from the time the GNSS signals are received.

---

36Note that the MX columns of tables 12 and 13 coincide with the results shown in table 7.
37Note that the blue lines of figure 27 (top and bottom plots) are the same as the blue and orange lines of figure 21, respectively.
38Note that the blue lines of figure 28 (top and bottom plots) are the same as the blue and orange lines of figure 22, respectively.
The influence of the terrain type by means of its associated Earth location and magnetic field is by means of the body attitude estimation and hence very weak.

The influence of the terrain type on the horizontal position \( \text{IA-VNSE} \) is very small, with slim differences among the various evaluated terrains. The only terrain type that clearly deviates from the others is the 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 estimations.

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, hampering its accuracy, and hence results in less precise initial estimations, including that of the scale. The estimated scale is significantly more important for the horizontal position estimation than for either the attitude or the vertical position, and the IA-VNS has no means to compensate the initial errors, which remain approximately equal (percentage wise) for the full duration of both scenarios.

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 Influence of Sensor Quality

The IA-VNS relies on the inertial estimations provided by the INS to generate the priors needed to ensure that the visual attitude and altitude estimations do not deviate in excess from their inertial counterparts, as well as to quickly obtain a rough pose estimation that replaces the expensive and less accurate sparse image alignment process (section 6). [5] includes a detailed analysis of the influence of the quality or grade of each individual sensor on the INSE. This analysis instead varies the quality of all onboard sensors simultaneously and does not only analyze the influence of these changes on the INSE, but also on the final IA-VNSE. The analysis consists on repeating the Monte Carlo simulation with the only change of replacing the performance parameters shown in [8], referred to as the “baseline” configuration, with others representing higher or lower quality sensors, and then comparing the results. As in the case of [5], such an analysis validates the proposed IA-VNS algorithms against optimistic assumptions when modeling the errors introduced by the various sensors. By ensuring that the IA-VNS also works as intended when relying on inertial estimations significantly worse than those obtained in the baseline configuration of previous sections, the confidence on the results in significantly enhanced. The analysis relies exclusively on scenario #1; similar conclusions would be obtained if employing scenario #2.

|            | SENS Baseline | SENS Better | SENS Worse |
|------------|---------------|-------------|------------|
| GYR        | Baseline      | Better      | Worse      |
| ACC        | Baseline      | Better      | Worse      |
| MAG        | Baseline      | Better      | Worse      |
| TAS-AOA-AOS| Baseline      | Better      | Worse      |
| OSP-OAT    | Baseline      | Baseline    | Worse      |

Table 14: Composition of SENS configurations

Table 14 proposes two different sensor grade combinations in addition to the “SENS Baseline” configuration employed to obtain the results shown in sections 7 and 8. The specific values of the different sensor parameters coincide with those employed in [5], and they are not repeated here. They consist of the “SENS Better” configuration, which employs sensors of better quality or grade than those of the baseline, which is generally associated to higher SwaP and price, and the “SENS Worse” configuration, composed by sensors with a significantly worse

With the exception of the barometer and thermometer, which are the same ones employed in the baseline.
In contrast with the sensitivity analyses of section 8 and [5], in which exactly the same algorithms and parameters are employed with the exception of the subject factor, the sensitivity of the prior based pose optimization adjustments in case of the “SENS Worse” configuration has been reduced while simultaneously increasing its strength, to adjust for the expected lower accuracy of the inertial estimations on which it relies. The modified parameters are displayed in table 15. In addition, it has been necessary to slightly adjust the configuration parameters of the “SENS Worse” inertial filter in a few runs to increase its stability.

Table 15: Pitch and bank adjustment setting for SENS configurations

| Parameter      | SENS Baseline | SENS Better | SENS Worse |
|----------------|---------------|-------------|------------|
| $\Delta \theta_{LOW}$ | 0.2°          | 0.2°        | 0.3°       |
| $\Delta \theta_{1,MAX}$ | 0.0005°       | 0.0005°     | 0.0006°    |
| $\Delta \zeta_{LOW}$ | 0.2°          | 0.2°        | 0.3°       |
| $\Delta \zeta_{1,MAX}$ | 0.0003°       | 0.0003°     | 0.0004°    |

Body Attitude Estimation

Table 16\(^40\) shows the total attitude error aggregated final state metrics for the scenario #1 over the MX terrain type with all three SENS configurations. The corresponding time aggregated metrics are depicted in figure 29\(^41\).

The attitude INSE results show a clear relationship with the sensors quality or grade, with better sensors consistently resulting in more accurate inertial attitude estimations. Note however that as explained in [5], this dependency can be traced to the gyroscopes, magnetometers, and air velocity (Pitot tube and air vanes) sensors, and not to the accelerometers and air data sensors, which do not impact the attitude estimation accuracy.

---

\(^{40}\)Note that the baseline columns of table 16 coincide with the results shown in table 5.

\(^{41}\)Note that the blue lines of figure 29 (top and bottom plots) are the same as the blue and orange lines of figure 16, respectively.
Table 16: Influence of sensors quality on final body attitude INSE and IA-VNSE for scenario #1 MX (100 runs)

A similar dependency can be observed for the attitude IA-VNSE, although the difference between the “SENS Better” and baseline configurations is very slim. Although not tested, it is likely that the “SENS Better” results could be improved if employing different prior based pose optimization parameters to take advantage of the more accurate inertial estimations, similarly to the changes introduced in table 15 for the “SENS Worse” configuration.

Note also that most of the IA-VNSE is already present at the beginning of the scenario, as the IA-VNS employs the inertial estimations for its initialization (section 5).

**Vertical Position Estimation**

Table 17 shows the vertical position error aggregated final state metrics for the scenario #1 over the MX terrain type and all three SENS configurations. The corresponding time aggregated metrics are depicted in figure 30.

As indicated in [5], the vertical position INSE is independent of the quality of all sensors, and depends exclusively on ionospheric effects and the atmospheric pressure offset change since the time the GNSS signals are lost. The vertical position IA-VNSE however shows a slight dependency with the quality or grade of the onboard sensors as a consequence of the activation of the prior based pose optimization process (section 6). Note however that of the three differences between the visual and inertial estimations that control the pitch and bank adjustment mechanism (pitch, bank, and altitude differences), only the first two are responsible for the impact of the sensors grade on the vertical position inertially assisted visual estimation.

---

42 Note that the baseline columns of table 17 coincide with the results shown in table 6.
43 Note that the blue lines of figure 30 (top and bottom plots) are the same as the blue and orange lines of figure 19, respectively.
### Horizontal Position Estimation

Table 18 shows the horizontal position error aggregated final state metrics for the MX scenario #1 and all three SENS configurations. The corresponding time aggregated metrics are depicted in figure 31.

| Scenario #1 MX  | INSE | IA-VNSE |
|-----------------|------|---------|
| SENS Config.    | Baseline | Better | Worse | Baseline | Better | Worse |
| [m] (t_{END})   | Δ\(\hat{h}\) | Δ\(\hat{h}\) | Δ\(\hat{h}\) | Δ\(\hat{h}\) | Δ\(\hat{h}\) | Δ\(\hat{h}\) |
| mean            | -4.18 | -4.18 | -4.15 | +22.86 | +20.74 | +28.74 |
| std             | 25.78 | 25.78 | 25.80 | 49.17  | 48.36  | 59.40  |
| max             | -70.49 | -70.50 | -69.74 | +175.76 | +138.61 | +279.11 |

Table 17: Influence of sensors quality on final vertical position INSE and IA-VNSE for scenario #1 MX (100 runs)

Figure 31: Influence of sensors quality on horizontal position INSE and IA-VNSE for scenario #1 MX (100 runs)

The horizontal position INSE results confirm the conclusions obtained in [5]. The error originates when integrating the ground velocity errors, which can be attributed to the wind velocity change from the time the GNSS signals are lost, with smaller contributions from the air velocity and body attitude inertial estimation errors. The last two factors are responsible for the minor variations in horizontal position INSE caused by changes in the quality of the onboard sensors.

| Scenario #1 MX  | INSE | IA-VNSE |
|-----------------|------|---------|
| SENS Config.    | Baseline | Better | Worse | Baseline | Better | Worse |
| [m, %] (t_{END})| Δ\(\hat{x}_{HOR}\) | Δ\(\hat{x}_{HOR}\) | Δ\(\hat{x}_{HOR}\) | Δ\(\hat{x}_{HOR}\) | Δ\(\hat{x}_{HOR}\) | Δ\(\hat{x}_{HOR}\) |
| mean            | 7276  | 7.10 | 7149 | 6.97 | 7499 | 7.33 | 488 | 0.46 | 476 | 0.45 | 578 | 0.54 |
| std             | 4880  | 5.69 | 4806 | 5.61 | 4958 | 5.81 | 350 | 0.31 | 363 | 0.33 | 364 | 0.32 |
| max             | 25288 | 32.38 | 25116 | 31.49 | 25731 | 33.24 | 1957 | 1.48 | 1934 | 1.46 | 2094 | 1.58 |

Table 18: Influence of sensors quality on final horizontal position INSE and IA-VNSE for scenario #1 (100 runs)

With respect to the horizontal position IA-VNSE, its variation with the quality of the sensors also relies on the
differences between the visual and inertial pitch and bank estimations that determine the activation of the prior based adjustments within the pose optimization process. As in the case of the attitude and altitude visual estimations, there are two reasons why the simulation results indicate a significant degradation in accuracy when employing the “SENS Worse” configuration instead of the baseline, but only a slight improvement when using the “SENS Better” sensors. The first reason is that, as explained in [5], there exists significantly more room for deterioration in the body attitude INSE (including the pitch and bank estimations related to the prior based pose adjustment) than for improvement, which is caused by the characteristics of GNSS-Denied navigation, and in particular by the multiple inaccuracies present in the attitude filter that prevent the tracking of the full accelerometer error $E_{acc}$. The second reason lies in the use of the same pitch and bank adjustment settings for the “SENS Better” and baseline configurations, but customized ones for “SENS Worse” (table 15). Although not tested, slightly more sensitive (lower) $\Delta \theta_{LOW}$ and $\Delta \zeta_{LOW}$ parameters coupled with slightly weaker (lower) $\Delta \dot{\theta}_{1,MAX}$ and $\Delta \dot{\zeta}_{1,MAX}$ settings would likely result in some improvements for the “SENS Better” configuration.

10 Summary of Results

This article proposes an SVO based IA-VNS installed onboard a fixed wing autonomous UAV that takes advantage of the GNSS-Denied estimations provided by an inertial filter (INS) to assist the visual pose optimization algorithms. The method is inspired in a PI control system, in which the attitude, altitude, and rate of climb inertial outputs act as targets to ensure that the visual estimations do not deviate far 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 stand-alone 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 far from their INS counterparts, while the body yaw angle visual estimation is significantly better than that obtained by the INS.

- The vertical position estimation shows major improvements over that of a stand-alone 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 stand-alone VNS or the INS, although drift is still present.

The conclusions of sensitivity analyses on the influence of the estimation results with respect to the overflown terrain type as well as the quality or grade of the onboard sensors on which the pose adjustments rely are the following:

- The terrain texture (or lack of) and its elevation relief are key factors for the IA-VNS visual odometry algorithms. Their influence on the aircraft pose estimation results are however slim with the only exception of regular plots of farmland characterized by a scarcity of features and zero vertical relief, over which the obtained results suffer some degradation.

- The quality of the onboard sensors influences the accuracy of the inertial estimations on which the visual targets rely, and although the IA-VNS quantitative pose estimation results vary based on the sensor grade, the qualitative conclusions listed above are independent from them.

11 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 stand-alone VNS, the most important of which is a major reduction in its horizontal position drift. The advantages occur independently of the quality of the onboard sensors, and when flying over a variety of terrain types. The IA-VNS can hence be considered as a virtual incremental horizontal displacement or ground velocity sensor, whose measurements can be fed back to the inertial filter, resulting in a fully integrated VINS. 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.

A Optical Flow

Let’s consider a pinhole camera [47] (one that adopts an ideal perspective projection) such as that depicted in figure 32 that is moving with respect to the Earth while maintaining within its field of view a given point \( p \) fixed to the Earth surface. The composition of positions and its time derivation, considering the ECEF as \( F_E \), the camera frame as \( F_C \), and a frame \( F_P \) with its origin in the terrain point \( p \) that does not move with respect to \( F_E \), results in the following expression when viewed in \( F_C \):

\[
\begin{align*}
T_{EP}^E &= T_{CP}^E + T_{EC}^E = R_{EC} T_{CP}^C + T_{EC}^E \\
\dot{T}_{EP}^E &= \dot{R}_{EC} T_{CP}^C + R_{EC} \dot{T}_{CP}^C + \dot{T}_{EC}^E = R_{EC} \dot{\omega}_{EC}^C T_{CP}^C + R_{EC} \dot{T}_{CP}^C + \dot{T}_{EC}^E \\
v_{EP}^E &= R_{EC} v_{CP}^C + v_{EC}^E + R_{EC} \dot{\omega}_{EC}^C T_{CP}^C = v_{EC}^E + \dot{\omega}_{EC}^C T_{CP}^E \\
v_{CP}^C &= R_{CR} v_{RP}^R = v_{EC}^C + v_{CP}^C + \dot{\omega}_{EC}^C T_{CP}^C = 0
\end{align*}
\]

(40)  

(41)  

(42)  

(43)

\[
\begin{align*}
v_{CP}^C &= -v_{EC}^C - \dot{\omega}_{EC}^C T_{CP}^C = -v_{EC}^C - \dot{\omega}_{EC}^C p^C \\
(44)
\end{align*}
\]

The homogeneous camera coordinates \( \bar{p}^C \) are defined as the ratio between the camera coordinates \( p^C \) and its third coordinate or depth \( p_3^C \), and represent an alternative view to \( p^{IMG} = \Pi(p^C) \) of how the point is projected in the image. Its time derivative is hence:

\[
\dot{\bar{p}}^C = \frac{\bar{p}^C}{p_3^C} \rightarrow \dot{p}^C = \frac{p_2^C \dot{p}^C - p_3^C}{p_3^C} \frac{\dot{p}^C}{p_2^C}
\]

(45)
Substituting both \( p^C \) and \( p^C_3 \) within (44) into (45), rearranging terms, and considering the relationship between the image and homogeneous camera coordinates, leads to the following expression for the optical flow \([89]\) or variation of the point image coordinates:

\[
\dot{p}^{\text{IMG}} = J^+ \Pi(\mathcal{M}^{-1}_{\text{EC}}(p^E)) \xi^C_{\text{EC}} = f \begin{bmatrix}
-\frac{1}{p^E_3} & 0 & \frac{\dot{p}^C_1}{p^E_3} & \frac{\dot{p}^C_2}{p^E_3} & -1 - \frac{p^C_1}{p^E_3} & 2 \frac{\dot{p}^C_2}{p^E_3} - \frac{\dot{p}^C_2}{p^E_3} & -\frac{p^C_1}{p^E_3} \\
0 & -\frac{1}{p^E_3} & \frac{\dot{p}^C_1}{p^E_3} & \frac{\dot{p}^C_2}{p^E_3} & 1 + \frac{p^C_2}{p^E_3}^2 & -\frac{p^C_1}{p^E_3} & -\frac{\dot{p}^C_1}{p^E_3}
\end{bmatrix} \begin{bmatrix}
\nu^C_{\text{EC}} \\
\omega^C_{\text{EC}}
\end{bmatrix}
\] (46)

Considering that the twist \( \xi \) is the time derivative of the transform vector \( \tau \) \([4]\), the optical flow jacobian is defined as the derivative of the local frame ideal perspective projection of a point fixed to the spatial frame with respect to the \( \text{SE}(3) \) element \( \mathcal{M} \) caused by a perturbation \( \Delta \tau \) in its local tangent space:

\[
J^+ \Pi(g_{\mathcal{M}}(p)) = \lim_{\Delta \tau \to 0} \frac{\Pi(g_{\mathcal{M} \oplus \Delta \tau}(p)) - \Pi(g_{\mathcal{M}}(p))}{\Delta \tau} \in \mathbb{R}^{2x6}
\] (47)

\[
\Pi(g_{\mathcal{M} \oplus \Delta \tau}(p)) \approx \Pi(g_{\mathcal{M}}(p)) + \left[ J^+ \Pi(g_{\mathcal{M}}(p)) \right] \Delta \tau \in \mathbb{R}^2
\] (48)

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 \( p^C_3 \), the effect on the image of a bigger linear velocity cannot be distinguished from that of a smaller depth.

References

[1] E. Gallo, “High Fidelity Flight Simulation for an Autonomous Low SWaP Fixed Wing UAV in GNSS-Denied Conditions.” https://github.com/edugallogithub/gnssdenied_flight_simulation, 2020. C++ open source code.

[2] E. Gallo, “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,” 2022. arXiv:2205.12572v1 [cs.RO], https://doi.org/10.48550/arXiv.2205.12572.

[3] J. Sola, “Quaternion Kinematics for the Error-State Kalman Filter,” 2017. arXiv:1711.02508v1 [cs.RO], https://doi.org/10.48550/arXiv.1711.02508.

[4] J. Sola, J. Deray, and D. Atchuthan, “A Micro Lie Theory for State Estimation in Robotics,” 2018. arXiv:1812.01537v9 [cs.RO], https://doi.org/10.48550/arXiv.1812.01537.

[5] E. Gallo and A. Barrientos, “Reduction of GNSS-Denied Inertial Navigation Errors for Fixed Wing Autonomous Unmanned Air Vehicles,” Aerospace Science and Technology, 2021. https://doi.org/10.1016/j.ast.2021.107237.

[6] E. Gallo, “Quasi Static Atmospheric Model for Aircraft Trajectory Prediction and Flight Simulation,” 2021. arXiv:2101.10744v1 [eess.SY], https://doi.org/10.48550/arXiv.2101.10744.

[7] E. Gallo, “Stochastic High Fidelity Simulation and Scenarios for Testing of Fixed Wing Autonomous GNSS-Denied Navigation Algorithms,” 2021. arXiv:2102.00883v3 [cs.RO], https://doi.org/10.48550/arXiv.2102.00883.

[8] E. Gallo, “Customizable Stochastic High Fidelity Model of the Sensors and Camera onboard a Low SWaP Fixed Wing Autonomous Aircraft,” 2021. arXiv:2102.06492v3 [cs.RO], https://doi.org/10.48550/arXiv.2102.06492.

[9] M. Hassanalian and A. Abdelkefi, “Classifications, Applications, and Design Challenges of Drones: A Review,” Progress in Aerospace Sciences, 2017. https://doi.org/10.1016/j.paerosci.2017.04.003.
[10] H. Shakhatreh, A. H. Sawalmeh, A. Al-Fuqaha, Z. Dou, E. Almaita, I. Khalil, N. S. Othman, A. Khreishah, and M. Guizani, “Unmanned Aerial Vehicles (UAVs): A Survey on Civil Applications and Key Research Challenges,” *IEEE Access*, 2019. https://doi.org/10.1109/ACCESS.2019.2909530.

[11] S. Bijjahalli, R. Sabatini, and A. Gardi, “Advances in Intelligent and Autonomous Navigation Systems for Small UAS,” *Progress in Aerospace Sciences*, 2020. https://doi.org/10.1016/j.paerosci.2020.100617.

[12] J. A. Farrell, *Aided Navigation, GPS with High Rate Sensors*. McGraw-Hill, Electronic Engineering Series, 2008.

[13] P. D. Groves, *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems*. Artech House, GNSS Technology and Application Series, 2008.

[14] A. B. Chatfield, *Fundamentals of High Accuracy Inertial Navigation*. American Institute of Aeronautics and Astronautics, Progress in Astronautics and Aeronautics, vol 174, 1997.

[15] M. Elbanhawi, A. Mohamed, R. Clothier, J. Palmer, M. Simic, and S. Watkins, “Enabling Technologies for Autonomous MAV Operations,” *Progress in Aerospace Sciences*, 2017. https://doi.org/10.1016/j.paerosci.2017.03.002.

[16] R. Sabatini, T. Moore, and S. Ramasamy, “Global Navigation Satellite Systems Performance Analysis and Augmentation Strategies in Aviation,” *Progress in Aerospace Sciences*, 2017. https://doi.org/10.1016/j.paerosci.2017.10.002.

[17] S. Thombre, M. Z. H. Bhuiyan, P. Eliardsson, B. Gabrielson, M. Pattison, M. Dunville, D. Fryganiotis, S. Hill, V. Manikundalam, M. Poloskey, S. Lee, L. Ruotsalainen, S. Soderholm, and H. Kuusniemi, “GNSS Threat Monitoring and Reporting: Past, Present, and a Proposed Future,” *The Journal of Navigation*, 2018. https://doi.org/10.1017/S0373463317000911.

[18] G. Zhang and L. Hsu, “Intelligent GNSS/INS Integrated Navigation System for a Commercial UAV Flight Control System,” *Aerospace Science and Technology*, 2018. https://doi.org/10.1016/j.ast.2018.07.026.

[19] C. Tippitt, A. Schultz, and W. Procino, “Vehicle Navigation: Autonomy Through GPS-Enabled and GPS-Denied Environments,” state of the art report DSIAC-2020-1328, Defense Systems Information Analysis Center, December 2020.

[20] N. Gyagenda, J. V. Hatilima, H. Roth, and V. Zhmud, “A Review of GNSS Independent UAV Navigation Techniques,” *Robotics and Autonomous Systems*, 2022. https://doi.org/10.1016/j.robot.2022.104069.

[21] R. Kapoor, S. Ramasamy, A. Gardi, and R. Sabatini, “UAV Navigation using Signals of Opportunity in Urban Environments: A Review,” *Energy Procedia*, 2017. https://doi.org/10.1016/j.egyproc.2017.03.156.

[22] A. Coluccia, F. Ricciato, and G. Ricci, “Positioning Based on Signals of Opportunity,” *IEEE Communications Letters*, 2014. https://doi.org/10.1109/LCOMM.2013.123013.132297.

[23] S. T. Goh, O. Abdelkhalik, and S. A. Zekavat, “A Weighted Measurement Fusion Kalman Filter Implementation for UAV Navigation,” *Aerospace Science and Technology*, 2013. https://doi.org/10.1016/j.ast.2012.11.012.

[24] A. Couturier and M. A. Akhloufi, “A Review on Absolute Visual Localization for UAV,” *Robotics and Autonomous Systems*, 2020. https://doi.org/10.1016/j.robot.2020.103666.

[25] T. White, J. Wheeler, C. Lindstrom, R. Christensen, and K. R. Moon, “GPS-Denied Navigation Using SAR Images and Neural Networks,” 2020. arXiv:2010.12108v1 [cs.CV], https://doi.org/10.48550/arXiv.2010.12108.

[26] Z. Sjanic and F. Gustafsson, “Simultaneous Navigation and Synthetic Aperture Radar Focusing,” *IEEE Transactions on Aerospace and Electronic Systems*, 2015. https://doi.org/10.1109/TAES.2015.120820.
[27] D. O. Nitti, F. Bovenga, M. T. Chiaradia, M. Greco, and G. Pinelli, “Feasibility of Using Synthetic Aperture Radar to Aid UAV Navigation,” Sensors, 2015. https://doi.org/10.3390/s150818334.

[28] G. Csaba, L. Somlyai, and Z. Vamossy, “Mobil Robot Navigation using 2D LIDAR,” IEEE 16th World Symposium on Applied Machine Intelligence and Informatics (SAMI), 2018. https://doi.org/10.1109/SAMI.2018.8324002.

[29] G. Hemann, S. Singh, and M. Kaess, “Long Range GPS-Denied Aerial Inertial Navigation with LIDAR Localization,” IEEE/RSJ International Conference on Intelligent Robots and Systems (iROS), 2016. https://doi.org/10.1109/IROS.2016.7759267.

[30] M. M. Miller, A. Soloviev, M. U. de Haag, M. Veth, J. Raquet, T. J. Klausutis, and J. E. Touma, “Navigation in GPS Denied Environments: Feature-Aided Inertial Systems,” Tech. Rep. RTO-EN-SET-116, NATO, 2010.

[31] H. Goforth and S. Lucey, “GPS-Denied UAV Localization using Pre Existing Satellite Imagery,” in IEEE International Conference on Robotics and Automation, IEEE, 2019. https://doi.org/10.1109/ICRA.2019.8793558.

[32] N. Ziaei, “Geolocation of an Aircraft using Image Registration Coupling Modes for Autonomous Navigation,” 2019. arXiv:1909.02875v1 [eess.IV], https://doi.org/10.48550/arXiv.1909.02875.

[33] T. Wang, “Augmented UAS Navigation in GPS Denied Terrain Environments using Synthetic Vision,” Iowa State University Graduate Theses and Dissertations, 15835, 2018. https://doi.org/10.31274/etd-180810-5462.

[34] M. D. Pritt and K. J. LaTourette, “Aircraft Navigation by means of Image Registration,” IEEE Applied Imagery Pattern Recognition Workshop (AIPR), 2013. https://doi.org/10.1109/AIPR.2013.6749335.

[35] D. Scaramuzza and F. Fraundorfer, “Visual Odometry Part 1: The First 30 Years and Fundamentals,” IEEE Robotics & Automation Magazine, 2011. https://doi.org/10.1109/MRA.2011.943233.

[36] F. Fraundorfer and D. Scaramuzza, “Visual Odometry Part 2: Matching, Robustness, Optimization, and Applications,” IEEE Robotics & Automation Magazine, 2012. https://doi.org/10.1109/MRA.2012.2182810.

[37] D. Scaramuzza, “Tutorial on Visual Odometry.” Robotics & Perception Group, University of Zurich, 2012.

[38] D. Scaramuzza, “Visual Odometry and SLAM: Past, Present, and the Robust Perception Age.” Robotics & Perception Group, University of Zurich, 2017.

[39] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, “Past, Present, and Future of Simultaneous Localization and Mapping: Towards the Robust Perception Age,” IEEE Transactions on Robotics, 2016. https://doi.org/10.1109/TRO.2016.2624754.

[40] C. Forster, M. Pizzoli, and D. Scaramuzza, “SVO: Fast Semi-Direct Monocular Visual Odometry,” in IEEE International Conference on Robotics and Automation, 2014. https://doi.org/10.1109/ICRA.2014.6906584.

[41] C. Forster, Z. Zhang, M. Gassner, M. Werlberger, and D. Scaramuzza, “SVO: Semidirect Visual Odometry for Monocular and Multicamera Systems,” IEEE Transactions on Robotics, 2016. https://doi.org/10.1109/TRO.2016.2623335.

[42] J. Engel, V. Koltun, and D. Cremers, “Direct Sparse Odometry,” IEEE Transactions on Pattern Analysis and Machine Intelligence, 2018. https://doi.org/10.1109/TPAMI.2017.2658577.

[43] J. Engel, T. Schops, and D. Cremers, “LSD-SLAM: Large Scale Direct Monocular SLAM,” European Conference on Computer Vision, 2014. https://doi.org/10.1007/978-3-319-10605-2_54.

[44] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, “ORB-SLAM: a Versatile and Accurate Monocular SLAM System,” IEEE Transactions on Robotics, 2015. https://doi.org/10.1109/TR0.2015.2463671.
[45] R. Mur-Artal and J. D. Tardos, “ORB-SLAM2: an Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras,” IEEE Transactions on Robotics, 2017. https://doi.org/10.1109/TRO.2017.2705103.

[46] R. Mur-Artal, Real-Time Accurate Visual SLAM with Place Recognition. PhD thesis, University of Zaragoza, 2017. https://doi.org/10.1109/TRO.2017.2705103.

[47] R. Hartley and A. Zisserman, Multiple View Geometry in Computer Vision. Cambridge University Press, 2nd ed., 2003.

[48] Y. Ma, S. Soatto, J. Kosecka, and S. S. Sastry, An Invitation to 3-D Vision, From Images to Geometric Models. Imaging, Vision, and Graphics, Springer, 2001.

[49] M. A. Fischler and R. C. Bolles, “RANSAC Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography,” Communications of the ACM, 1981. https://doi.org/10.1145/358669.358692.

[50] D. Scaramuzza and Z. Zhang, “Visual-Inertial Odometry of Aerial Robots,” 2019. arXiv:1906.03289v2 [cs.RO], https://doi.org/10.48550/arXiv.1906.03289.

[51] G. Huang, “Visual-Inertial Navigation: A Concise Review,” 2019. arXiv:1906.02650v1 [cs.RO], https://doi.org/10.48550/arXiv.1906.02650.

[52] L. von Stumberg, V. Usenko, and D. Cremers, “Chapter 7 - A Review and Quantitative Evaluation of Direct Visual Inertial Odometry,” in Multimodal Scene Understanding (M. Y. Yang, B. Rosenhahn, and V. Murino, eds.), Academic Press, 2019.

[53] X. Feng, Y. Jiang, X. Yang, M. Du, and X. Li, “Computer Vision Algorithms and Hardware Implementations: A Survey,” Integration, the VLSI Journal, 2019. https://doi.org/10.1016/j.vlsi.2019.07.005.

[54] A. Al-Kaff, D. Martin, F. Garcia, A. de la Escalera, and J. Maria, “Survey of Computer Vision Algorithms and Applications for Unmanned Aerial Vehicles,” Expert Systems With Applications, 2017. https://doi.org/10.1016/j.eswa.2017.09.033.

[55] A. I. Mourikis and S. I. Roumeliotis, “A Multi-State Constraint Kalman Filter for Vision-aided Inertial Navigation,” Proceedings 2007 IEEE International Conference on Robotics and Automation, 2007. https://doi.org/10.1109/ROBOT.2007.364024.

[56] S. Leutenegger, P. Furgale, V. Rabaud, M. Chli, K. Konolige, and R. Siegwart, “Keyframe Based Visual Inertial SLAM Using Nonlinear Optimization,” Robotics: Science and Systems, 2013. https://doi.org/10.3929/ethz-b-000236658.

[57] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale, “Keyframe Based Visual Inertial SLAM Using Nonlinear Optimization,” The International Journal of Robotics Research, 2015. https://doi.org/10.1177/0278364914554813.

[58] M. Bloesch, S. Omari, M. Hutter, and R. Siegwart, “Robust Visual Inertial Odometry Using a Direct EKF Based Approach,” International Conference of Intelligent Robot Systems, 2015. https://doi.org/10.3929/ethz-a-010566547.

[59] T. Qin, P. Li, and S. Shen, “VINS-Mono: A Robust and Versatile Monocular Visual Inertial State Estimator,” IEEE Transactions on Robotics, 2018. https://doi.org/10.1109/TR0.2018.2853729.

[60] S. Lynen, M. W. Achtelik, S. Weiss, M. Chli, and R. Siegwart, “A Robust and Modular Multi Sensor Fusion Approach Applied to MAV Navigation,” International Conference of Intelligent Robot Systems, 2013. https://doi.org/10.1109/IR0S.2013.6696917.

[61] M. Faessler, F. Fontana, C. Forster, E. Mueggler, M. Pizzoli, and D. Scaramuzza, “Autonomous, Vision Based Flight and Live Dense 3D Mapping with a Quadrotor Micro Aerial Vehicle,” Journal of Field Robotics, 2015. https://doi.org/10.1002/rob.21581.
[62] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, “On Manifold Pre Integration for Real Time Visual Inertial Odometry,” IEEE Transactions on Robotics, 2017. https://doi.org/10.1109/TRO.2016.2597321.

[63] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. Leonard, and F. Dellaert, “iSAM2: Incremental Smoothing and Mapping Using the Bayes Tree,” The International Journal of Robotics Research, 2012. https://doi.org/10.1177/0278364911430419.

[64] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. W. Achtelik, and R. Siegwart, “The EuRoC MAV Datasets,” IEEE International Journal of Robotics Research, 2016. https://doi.org/10.1177/0278364915620033.

[65] J. Delmerico and D. Scaramuzza, “A Benchmark Comparison of Monocular Visual-Inertial Odometry Algorithms for Flying Robots,” IEEE International Conference on Robotics and Automation, 2018. https://doi.org/10.1109/ICRA.2018.8460664.

[66] R. Mur-Artal and J. M. M. Montiel, “Visual Inertial Monocular SLAM with Map Reuse,” IEEE Robotics and Automation Letters, 2017. https://doi.org/10.1109/LRA.2017.2653359.

[67] R. Clark, S. Wang, H. Wen, A. Markham, and N. Trigoni, “VINet: Visual-Inertial Odometry as a Sequence-to-Sequence Learning Problem,” Proceedings of the AAAI Conference on Artificial Intelligence, 2017. https://ojs.aaai.org/index.php/AAAI/article/view/11215.

[68] M. K. Paul, K. Wu, J. A. Hesch, E. D. Nerurkar, and S. I. Roumeliotis, “A Comparative Analysis of Tightly Coupled Monocular, Binocular, and Stereo VINS,” IEEE International Conference on Robotics and Automation, 2017. https://doi.org/10.1109/ICRA.2017.7989022.

[69] Y. Song, S. Nuske, and S. Scherer, “A Multi Sensor Fusion MAV State Estimation from Long Range Stereo, IMU, GPS, and Barometric Sensors,” Sensors, 2017. https://doi.org/10.3390/s17010011.

[70] A. Solin, S. Cortes, E. Rahtu, and J. Kannala, “PIVO: Probabilistic Inertial Visual Odometry for Occlusion Robust Navigation,” IEEE Winter Conference on Applications of Computer Vision, 2018. https://doi.org/10.1109/WACV.2018.00073.

[71] S. Houben, J. Quenzel, N. Krombach, and S. Behnke, “Efficient Multi Camera Visual Inertial SLAM for Micro Aerial Vehicles,” IEEE/RSJ International Conference on Intelligent Robots and Systems, 2016. https://doi.org/10.1109/IROS.2016.7759261.

[72] K. Eckenhoff, P. Geneva, and G. Huang, “Direct Visual Inertial Navigation with Analytical Preintegration,” IEEE International Conference on Robotics and Automation, 2017. https://doi.org/10.1109/ICRA.2017.7989171.

[73] H. Strasdat, J. M. M. Montiel, and A. J. Davison, “Real Time Monocular SLAM: Why Filter?,” IEEE International Conference on Robotics and Automation, 2010. https://doi.org/10.1109/ROBOT.2010.5509636.

[74] G. Gallego, T. Delbruck, G. Orchard, C. Bartolozzi, B. Taba, A. Censi, S. Leutenegger, A. Davison, J. Conradt, K. Daniilidis, and D. Scaramuzza, “Event Based Cameras: A Survey,” IEEE Transactions on Pattern Analysis and Machine Intelligence, 2022. https://doi.org/10.1109/TPAMI.2020.3008413.

[75] E. Mueggler, G. Gallego, H. Rebecq, and D. Scaramuzza, “Continuous Time Visual Inertial Odometry for Event Cameras,” IEEE Transactions on Robotics, 2018. https://doi.org/10.1109/TRO.2018.2858287.

[76] P. Stojakovic and B. Rasuo, “Single Propeller Airplane Minimal Flight Speed based upon the Lateral Maneuver Condition,” Aerospace Science and Technology, 2016. https://doi.org/10.1016/j.ast.2015.12.012.

[77] P. Stojakovic, K. Velimirovic, and B. Rasuo, “Power Optimization of a Single Propeller Airplane Take-Off Run on the Basis of Lateral Maneuver Limitations,” Aerospace Science and Technology, 2018. https://doi.org/10.1016/j.ast.2017.10.015.

[78] “osgEarth.” http://osgearth.org.
[79] “Open Scene Graph.” http://openscenegraph.org.

[80] P. J. Huber, *Robust Statistics*. John Wiley & Sons, 1981.

[81] J. Fox and S. Weisberg, “Robust regression,” 2013.

[82] S. Baker and I. Matthews, “Lucas-Kanade 20 Years On: A Unifying Framework,” *International Journal of Computer Vision*, 2004. https://doi.org/10.1023/B:VISI.0000011205.11775.fd.

[83] O. Faugeras and F. Lustman, “Motion and Structure from Motion in a Piecewise Planar Environment,” *International Journal of Pattern Recognition and Artificial Intelligence*, 1998. https://doi.org/10.1142/S0218001498000285.

[84] S. Baker, R. Gross, and I. Matthews, “Lucas-kanade 20 years on: A unifying framework: Part 4,” Tech. Rep. CMU-RI-TR-04-14, Carnegie Mellon University, 2004.

[85] K. Ogata, *Modern Control Engineering*. Prentice Hall, 4th ed., 2002.

[86] S. Skogestad and I. Postlethwaite, *Multivariable Feedback Control: Analysis and Design*. John Wiley & Sons, 2nd ed., 2005.

[87] B. L. Stevens and F. L. Lewis, *Aircraft Control and Simulation*. John Wiley & Sons, 2nd ed., 2003.

[88] G. F. Franklin, J. D. Powell, and M. Workman, *Digital Control of Dynamic Systems*. Ellis-Kagle Press, 3rd ed., 1998.

[89] David J. Heeger, “Notes on Motion Estimation,” 1998.