Stereo visual-inertial aided navigation for UAVs
Duarte Bustorff de Dornellas Martins da Silva
Instituto Superior Técnico, Lisboa, Portugal
May 2019

Abstract
This work, developed in the scope of the ELEVAR project, tested the applicability of a navigation system based on cameras and inertial sensors for a quadrotor UAV (Unmanned Aerial Vehicle), motivated by the desire to expand its operational envelope to regions with low GPS signal reception. This project aims at the development of solutions for the inspection of Civil Engineering structures such as dams or bridges using UAVs.

A system using a stereo camera pair and an IMU (Inertial Measurement Unit) was developed, as well as intrinsically and extrinsically calibrated. A hardware synchronization circuit was designed, and an SoC (System on a Chip) was used for onboard processing using OKVIS, a recent open-source VIO (Visual-Inertial Odometry) algorithm. In order to quantitatively assess its performance, an indoors dataset was recorded in a treated environment, with precise ground-truth from a motion capture system.

The system was integrated with the UAV developed by TEKEVER, one of the project partners, functioning as an additional GPS sensor from the perspective of the onboard autopilot. For this end, the VIO trajectory data was georeferenced using information from the onboard GPS receiver, as well as from the orientation estimator embedded in the IMU.

Having validated the system with handheld testing, flight tests were performed, having shown that the inclusion of a system of this nature effectively yields improved trajectory estimates under low GPS signal reception, as is the case at the vicinity of structures such as those which are to be monitored within the ELEVAR project.

Keywords: Autonomous systems, aerial robotics, computer vision, SLAM, visual-inertial odometry

1. Introduction
The main goal of this work was the development of a visual-inertial navigation solution for an unmanned aerial vehicle, based on a stereo camera pair and an IMU. This system is to be used for the inspection of vertical structures of difficult access such as dams, and was developed in the context of the ELEVAR project, [5, 6, 7]. Its ultimate goal was the inspection and detection of fissures in inaccessible structures, by generating high resolution mosaics captured from a camera onboard the UAV.

The need for a high precision, robust navigation system arose from the desire to perform autonomous flight. This would allow for the a priori definition of frame superposition, as well as the registration of camera position at each frame, invaluable characteristics for an image stitching pipeline.

The choice of a stereo visual-inertial platform departed from the shortcomings of typical industry-standard solutions based on GPS, and was supported on the recent success of systems with this configuration. Amongst its most favourable characteristics would be the complementary nature of cameras (exteroceptive sensors) and inertial measurement units (proprioceptive sensors), along with the low weight and small dimensions that make it a good candidate for onboarding the restrictive platform that is an unmanned aerial vehicle.

2. Visual-Inertial Navigation
The problem of visual-inertial odometry aims at the estimation of the sequence of poses, i.e. position and orientation, endured by a set of visual and inertial sensors, given a sequence of image frames and accelerometer/gyroscope measurements.

2.1. Visual Navigation
There are two main divides in the approaches to the problem of computing the trajectory of a set of cameras from its measurements: the decision on which image information to use, and the formulation of the estimation algorithm. The first is split between sparse approaches, in which feature points are triangulated, and dense methods, in which image patches are directly aligned; The second is mostly divided between incremental approaches, using some sort of filter, and window-based batch optimization approaches, using either pose-graph optimization or full bundle adjustment.
A comparison of the performance of feature-based and dense approaches was made for the monocular case in [13], which concluded that using current processors both approaches show similar accuracy, feature-based methods yielding more accurate results with limited computational resources. In [15], a benchmark evaluation concluded that keyframe-based algorithms consistently outperform its filtering-based counterparts, resulting in higher accuracies per unit of computing time for both the monocular and stereo cases.

Feature-based methods rely on visual features that represent an abstraction built on top of the raw image data, so that they can also be classified as indirect, whereas algorithms that use raw image data can be classified as direct. It should be noted that direct methods are not necessarily dense, as indirect methods are also not necessarily sparse.

The projective nature of cameras renders each point in the image plane equivalent to a ray in the 3D environment. For a monocular system, this has the direct consequence that the absolute (metric) scale of the trajectory is rendered unobservable.

For a calibrated stereo system, matching points in both camera frames would ideally allow for the unambiguous triangulation of each point, such that the consecutive positions of the detected point cloud could be used for the estimation of the incremental displacement and rotation of the camera system, i.e. visual odometry. This approach was first explored in [12], its author having effectively coined this term.

Stereo triangulation is, however, not without limitations. For circumstances in which the camera baseline is significantly smaller than the environment being captured, the stereo pair of images would have virtually no disparity, and each pair of corresponding rays being parallel would render triangulation impossible. In this scenario, no additional information would be gained from the second camera, and the stereo configuration is said to degenerate to the monocular case. This fact, its smaller footprint, and its omnipresence in current smartphones have been significant drivers behind the study of monocular, or single camera platforms.

SLAM methods have a natural proximity to pure odometry algorithms, the greatest difference being that odometry is mainly concerned with the incremental transformation between instants, while typical SLAM methods also estimate a global map of the environment and perform tasks such place recognition and trajectory correction through loop closures.

2.2. Visual-Inertial Navigation
The introduction of inertial sensors as a complement to camera systems came as a way of compensating for the shortcomings of the projective nature of these systems. By directly providing metric information about the motion of the sensor assembly, it can help resolve scale ambiguity problems in monocular systems, as well as increase the robustness of multi-camera systems for operation in scenarios with low baseline-to-depth ratios. It also translates in increased independence on external infrastructure to determine scale, such as the use of objects of known dimensions for system initialization, or domain knowledge in any other form.

The fusion of sensor data can either be loosely-coupled - in which pose is typically estimated with image information and fused in a later stage with inertial measurements - or tightly coupled - in which the estimation algorithm jointly takes into account both visual and inertial measurements. The latter approach takes into account data correlations that are central to a high performing trajectory estimation algorithm, and has thus been widely adopted.

3. State Of The Art
One of the early successful visual-inertial navigation algorithms was proposed in [11], an EKF-based algorithm which by not including features in the state vector kept the algorithmic complexity only linear in the number of observed points. Its latest update, in [10], improved its consistency based on an observability analysis of the linearized and original systems. Despite being one of the earliest examples of visual-inertial odometry systems, its ideas have been proved especially relevant for low computing power applications, a context in which it still yields very competitive results.

An example of a semi-dense monocular visual-inertial SLAM system, also based on an EKF can be found in [1], which by using an inverse-depth parametrization does not require any initialization procedure, a common issue with monocular systems.

Following the idea introduced in the pure-odometry algorithm [8], the application of keyframe-based methods within visual-inertial odometry was adopted in [9], in which the optimization problem was formulated directly on the $SE(3)$ manifold.

Recently, the work in [4] formulated the VIO problem using factor graphs. The fact that inertial sensors and cameras have significantly different working frequencies was explicitly addressed by grouping the high frequency IMU measurements into motion constraints on the factor graph of lower frequency visual keyframes, an idea on which the work in [14] based its approach.

Finally, a benchmark evaluation of VIO methods in which the previously cited algorithms are included was put forward in [3].
exp (bitrarily spaced keyframes, and a sliding window of
process is schematically illustrated in Figure 1.
The original space and consequent application. This
its local algebra, followed by its transformation to
its inverse, the state update can be computed at
state space transformation, and the logarithm map
in its associated algebra.

cording to a gaussian distribution around the origin
ally, this can be seen as a state parametrization ac-
tive perturbations in euclidean space. Conceptu-
vector and the other terms are traditional addi-

\[ \delta \chi_k \]

with (4) corresponds to online extrinsic calibra-
tion, and (5) to computing a single pose of an a
priori calibrated system, the adopted approach.

The inertial cost function error terms are com-
puted as the difference between a prediction based
on the previous internal state and the IMU mea-
surements, \( \hat{x}_R \), and the actual internal state, \( x_R \).
This is computed through the classical Runge-
Kutta numerical integration of the IMU kinematic
model, keeping the internal state symbolic:

\[
\begin{align*}
W_{\mathbf{r}} &= \left[ W_{\mathbf{r}_B}, W_{\mathbf{q}_B}, B \mathbf{v}, b_a, b_v \right]^T \\
W_{\hat{x}_R} &= \left[ \hat{x}_{\mathbf{r}_B}, \hat{x}_{\mathbf{q}_B}, \hat{x}_B, \hat{x}_B, \hat{x}_B \right]^T \\
&\in \mathbb{R}^{15}
\end{align*}
\]

where the \( B \) and \( W \) indexes refer to the body and
world reference frames, respectively.

The orientation quaternion not being in Eu-
clidean space motivated a dual representation, in
which an internal error-state, \( \delta \chi_{x_R} \), was defined at
the local algebra of each state, \( x_R \):

\[
\delta \chi_{x_R} = \left[ \delta \mathbf{r}_B, \delta \mathbf{q}_B, B \delta \mathbf{v}, \delta \mathbf{b}_a, \delta \mathbf{b}_v \right]^T \\
&\in \mathbb{R}^{15}
\]

where \( \delta \mathbf{q}_B \) is an infinitesimal axis/angle rotation
vector and the other terms are traditional additive
perturbations in euclidean space. Conceptually,
this can be seen as a state parametrization ac-

gording to a gaussian distribution around the origin
in its associated algebra.

With the exponential map as the error-state to
state space transformation, and the logarithm map
its inverse, the state update can be computed at
its local algebra, followed by its transformation to
the original space and consequent application. This
process is schematically illustrated in Figure 1.

The cost function is computed as the sum of vi-
ual and inertial terms, respectively \( f_v \) and \( f_i \), com-
puted over two fixed-size windows: a window of ar-
bitrarily spaced keyframes, and a sliding window of
the latest images.

\[
F(x) = \sum_{i \in \mathcal{I}} \left[ f_i^T \cdot W^i \cdot f_i \right] \text{ inertial terms} \\
+ \sum_{j \in \mathcal{J}} \sum_{k \in \mathcal{K}(i,j)} f^{ijk}_v \cdot W^{ijk}_v \cdot f^{ijk}_v \text{ visual terms}
\]

where \( \mathcal{I} \) the set of time indexes corresponding to
frames in the optimization window; \( \mathcal{J} \) the set of cameras;
\( \mathcal{K}(i,j) \) the set of points visible by the \( j \)-th
camera at time \( i \); \( W^{ijk}_v \) the information matrix of
the \( k \)-th point observed by the \( j \)-th camera at time
\( i \), and \( W^i \) the IMU measurements information
matrix at time \( i \).

The visual cost function terms follow the tradi-
tional reprojection error formulation. With \( \mathbf{T}_c \)
the \( i \)-th camera extrinsics, and expressing land-
marks using homogeneous coordinates this can be
formulated as:

\[
\begin{align*}
\hat{p}^{ijk}_v &= p^{ijk}_v - \mathbf{C}_i \mathbf{T}_W \cdot p^{ijk}_v \\
&= p^{ijk}_v - \mathbf{C}_i \mathbf{T}_c_0 \cdot \mathbf{C}_i \mathbf{T}_W \cdot p^{ijk}_v
\end{align*}
\]

5. Experimental Setup
Prior to addressing their integration, the experi-
mental setup can be thought of as divided between
the quadrotor UAV, developed by TEKEVER, and the visual-inertial sensor setup, developed during this work.

5.1. UAV

The quadrotor UAV, shown in Figure 2, has a diagonal inter-rotor distance of 86cm and is approximately 46cm high. It has an endurance of around 18 minutes carrying 1Kg of payload, powered by a 20000mAh 6s LiPo battery.

Among the sensors onboard the UAV can be listed a GPS receiver and compass, a Range-Finder and an IMU embedded in the PixHawk autopilot module. The sensor to autopilot communications are based on the MAVLink 2.0 protocol, and the ground control station is based on the Mission Planner software and Ardupilot.

Additionally, a commercial camera is mounted on a gimbal on the bottom of the UAV. Not meant for navigation purposes, this is used for capturing high-definition images of the structures to be inspected.

5.2. Visual-Inertial Sensor Setup

For inertial sensing, an XSens MTx-28A33G35 IMU was used. This unit encompasses 3D accelerometers, gyroscopes and magnetometers. It supports an external trigger signal and is also capable of onboard processing, allowing for the output of filtered orientation estimates. Its maximum sampling frequency is 512Hz for unprocessed data and 256Hz for the filtered orientation data, and its minimum sampling frequency is 100Hz. Despite this generation of products not having Linux support, a C communication API is provided which was used for sensor configuration and data transmission.

For visual sensing, a pair of global shutter FLIR/Point Grey BFLY-U3-05S2M-CS cameras were selected, with a resolution of 808 by 608 pixels, or around 0.5MP. Their maximum frame rate is 50 fps and, like the selected IMU, they support external triggering. FLIR/Point Grey offers a C++ API based on the GenICam3 standard for machine vision cameras, which has been used for configuration and data capture.

A pair of Fujinon YF2.8A-2 lenses are used with the cameras, having a manually adjustable focal length in the range of 2-2.8mm.

An Arduino Nano microcontroller is used to handle the interface with external switches, as well as to generate the trigger signals, in order to guarantee the synchronized operation of the cameras and the IMU.

An NVIDIA Jetson TX2 was chosen as the embedded computing platform onboard this sensor setup, to which the sensors are connected via a 4-way USB 3.0 hub.

6. Implementation

6.1. Sensor Settings

Despite the maximum frame rate of 50 fps, the cameras were chosen to operate at 20 fps. This decision was mainly driven by the fact that using an external trigger imposes restrictions on this parameter, since the sensor readout and the exposure cannot happen simultaneously on the standard triggering mode.

An overlapped exposure/readout mode is provided, wherein the image is first stored in an internal buffer and the data transfer only occurs isochronously with the next trigger activation, or the start of exposure of the next image. However, should the trigger arrive during data transfer, exposure would be delayed until the latter is complete, a consequence of the limited capacity of the internal camera buffer, able to hold a single frame. This would have the undesired consequence of loss of synchronism between camera and IMU measurements, making the overlapped exposure mode unsuitable.

The standard triggering mode was used instead, albeit at the cost of frame rate reduction.

The cameras have an internal exposure controller, which stops the sensor exposure when a predefined average brightness level has been reached. It should be noted that for very low luminosity scenarios this may have the undesired consequence of exposure time exceeding the interval between triggers, which may result in dropped frames and loss of synchronism. Conversely, circumstances where luminosity is very high may result in overexposed images, should the shutter not be capable of a fast enough response.

Consequently, lens aperture is a factor which should also be taken into account, as it plays a direct part in the amount of light reaching the CCD sensor. For the reasons described in the previous paragraph, too small apertures may result in too
long exposure times and in dropped frames, while too large values may yield overexposed images. As a compromise allowing for indoors and outdoors operation, the lenses were set to an aperture close to the middle of the operating range, a configuration which was tested successfully in both scenarios.

Since the cameras were chosen to operate at 20fps and the minimum IMU sampling frequency is 100Hz, it is not possible to use a single signal to trigger both sensors. As such, two independent trigger signals were required to respect the operating frequencies of both sensors. In order to guarantee measurement isochronism, these should ideally have no phase offset and have frequencies related by an integer multiple, \( k \). This is illustrated schematically in Figure 3, with the camera and IMU triggering instants represented respectively in red and blue, and with \( t_0 \) an arbitrary instant of simultaneous camera and IMU triggering.

![Camera/IMU trigger sequence](image)

**Figure 3:** Camera/IMU trigger sequence

In order to be within the frequency range of both calibrated and filtered IMU data (512Hz and 256Hz, respectively), this factor was chosen to be \( k = 10 \), meaning the IMU was triggered at a frequency of 10 times that of the cameras: \( f_{IMU} = 200Hz \).

As detailed in [16], the internal IMU orientation estimator is based on a Kalman Filter and fuses information from the accelerometers, gyroscopes and magnetometers to compute an attitude estimate at each instant. By assuming an average null acceleration over predefined time intervals, it computes the direction of the local gravitational field from the accelerometer measurements. Two out of the three degrees of freedom associated with orientation can be computed from this, corresponding to the roll and pitch angles. The third degree of freedom, corresponding to a rotation along a gravitationally-aligned axis (or yaw) is computed with respect to the direction of the local magnetic field, which is initialized by the magnetometers and updated by the gyroscopes. The drift, which would be introduced by dead reckoning orientation using only the gyroscope measurements, is corrected by also using the magnetometers.

### 6.2. UAV Integration

As an essential part of the integration of the two systems, a bidirectional communication link between the NVIDIA Jetson and the PixHawk had to be implemented. This was done using the MAVLink protocol, already in use for the onboard sensors and autopilot communications.

The communications on the PixHawk \( \rightarrow \) Jetson direction were used for the logging of all the sensor and estimator messages, which were republished as ROS topics both for the georeferencing pipeline and for completeness of the generated datasets.

The Jetson \( \rightarrow \) PixHawk communication channel was used for publishing the georeferenced odometry estimates, as well as for the software triggering of the HD camera.

The functional diagram of both systems as well as their interconnection is provided in the diagram in Figure 4.

In order to make use of all the sensors onboard the UAV in conjunction with the visual-inertial system described in the previous sections, the integration was performed using the estimator onboard the PixHawk autopilot. This estimator originally fused information from the sensors onboard the UAV, but can also fuse data from additional sensors. The in-
integration was done by providing the odometry estimates as an additional source of information in the form of a simulated GPS sensor. To do so, the georeferencing of the odometry results was performed, as put forward in the next subsection.

6.2.1 VIO Georeferencing

A fundamental characteristic of odometry algorithms is the fact that trajectories are computed relatively to a starting point: there is no a priori requirement as to the reference frame in which they are to be expressed. The process of georeferencing odometry data is performed in two steps: first, the odometry results are converted into the local NED (North-East-Down) reference frame, after which they are converted into LLA (Longitude-Latitude-Altitude) coordinates. To do so, an estimate of the relative orientation between the sensor setup and the NED reference frame at startup is required, for which the filtered IMU data is used. Afterwards, the absolute position at system startup is required to convert between NED and LLA coordinates, being obtained from the GPS receiver onboard the UAV. These reference frames are schematically represented in Figure 5.

7. Results

For the preliminary validation of the standalone algorithm, the dataset in [2] was used. After this, an indoors dataset was recorded at the ISR offices, using an Optitrack motion capture system as a source of accurate ground truth. The recording room was treated so as to have constant illumination and as much visual texture as possible, that is, to provide the algorithm with near-ideal conditions in order to evaluate it at its best performing scenario.

Afterwards, several handheld outdoors trajectories were recorded at the TEKEVER offices in Caldas da Rainha and Parque das Nações, followed by in-flight tests at the site of the final demonstration of the ELEVAR project, the IC2 viaduct near the Trancão river. These datasets include GPS position estimates, although naturally not always at a sanity level that might make them suitable to be used as precise ground truth. The handheld trajectories were, however, recorded for several cyclical runs by running roughly through the same path multiple times while stopping at a control point at the starting position. Although naturally not ideal, this was done in an attempt to provide a way of assessing estimation drift at these positions. For the in-flight trajectories this was not possible, so that no form of ground-truth is provided and the evaluation of results is left at a qualitative level.

7.1. Error Metrics

The evaluation of trajectories with precise ground-truth typically requires two preprocessing steps: time-alignment of the ground truth and odometry data, followed by their transformation into a common reference frame. The temporal alignment was performed by resampling the ground truth data at the closest instant to each odometry estimate, where the reception timestamps were used directly. Afterwards, the ground-truth data was converted to the local reference frame, the odometry data naturally left unchanged. To do so, both orientation and position had to be corrected for their respective initial values, \( r_0 \) and \( \hat{q}_0 \), after which they were rotated by \( B R_{GT} \), the rotation matrix between the motion capture system and the local odometry algorithm reference frames:

\[
\begin{align*}
\mathbf{r} &= B R_{GT} \cdot (r_{GT} - r_0) \\
\mathbf{q} &= Q^T B R_{GT} \times \hat{q}_{GT} \times q_0^{-1}
\end{align*}
\]

with \( B R_{GT} \) obtained by inspection of the reference frame specifications.

The formulation of the error terms followed the approach in [17], the error-pose \( \Delta \mathbf{r} \) and \( \Delta \mathbf{q} \) being:

\[
\begin{align*}
\Delta \mathbf{r} &= \mathbf{r} - \hat{\mathbf{r}} \\
\Delta \mathbf{q} &= \mathbf{q} - \hat{\mathbf{q}}^{-1}
\end{align*}
\]

with \( \mathbf{r} \) and \( \mathbf{q} \) corresponding to the ground truth terms and estimated values indicated by a hat.

While for ground truth-logged trajectories these can be computed for each pose estimate, for cyclical trajectories they can only be computed at the control points, where position is assumed to be exactly that of the starting point. Naturally, should there be no other form of trajectory ground-truth, these figures cannot be computed.

The trajectory position error, \( TPE \), is defined as the root mean square of the position error \( \Delta \mathbf{r} \):

\[
TPE = \sqrt{\frac{1}{N} \sum_{i=1}^{N} ||\Delta \mathbf{r}_i||^2},
\]
while the trajectory orientation error, $TOE$, is defined as the root mean square of the angle error obtained by converting the error quaternion to an axis/angle vector.

$$TOE = \sqrt{\frac{1}{N} \sum_{i=1}^{N} \angle(\Delta q_i)^2}$$  \hspace{1cm} (12)

These figures are useful for their immediacy, but do not take into account the evolution of the error terms with time, so that their presentation will be complemented by plots showing this dependency.

7.2. Indoors Results

Throughout the indoors recording sessions at ISR, the movement was broadly restricted to regions within a $2 \times 2m$ square. This was used as a visual guide while recording, so as to be able to perform trajectories which were planned a priori, either restricting or including certain types of motions.

In this section, two figures are shown for each trajectory. A 2D top view of the trajectory is put forward (with the ground-truth data in red and the estimated positions in blue) followed by plots of the position and orientation errors as functions of time. Finally, the $TPE$ and $TOE$ error metrics are put forward for each sequence.

7.2.1 Experiment 1

The trajectory in Experiment 1, roughly an 8-figure, is shown in Figure 6 and was mostly translational, with the sensor orientation kept constant and approximately that of startup.

The orientation error was found to be below $5^\circ$, the position error having its maximum below $5 cm$.

The trajectory error metrics were computed as:

$$\begin{cases} TPE = 0.021 m \\ TOE = 2.183^\circ \end{cases}$$  \hspace{1cm} (13)

7.2.2 Experiment 2

The trajectory in Experiment 2, shown in Figure 8, was meant as an extension of that in Experiment 1 where rotation was introduced at each corner. Now, this was done so that the motion was always in the local forward direction. The challenge of rotating the complete setup is clear from the fact that the middle lines no longer intersect. This is naturally not a problem from the standpoint of this analysis, so that despite being visually less pleasing it was still informative of the algorithm behaviour for different motions.

The position error is consistently below $14 cm$, while the orientation error has a peak near $19^\circ$. This relatively isolated value was due to a partial occlusion of the reflective markers, wherein the setup was not observed by the motion capture system and the corresponding pose estimates were kept constant.

The position and orientation error metrics, $TPE$...
Figure 10: Experiment 3 - Map view

and $TOE$, where computed as:

\[
\begin{aligned}
TPE &= 0.050 \text{m} \\
TOE &= 2.719 \degree
\end{aligned}
\] (14)

7.3. Outdoors datasets

Two outdoors trajectories are now put forward - one recorded at the TEKEVER offices in Caldas da Rainha, in which the UAV was carried by hand and which were performed as part of the integration tests, and the in-flight trajectory of the final demonstration of the ELEVAR project.

For each of these, the results obtained by using the GPS receiver or the georeferenced visual-inertial odometry algorithm exclusively, as well as the result of their fusion within the autopilot are put forward. They are respectively represented in blue, green and red, and are shown in the following section both superimposed with a map and in latitude and longitude plots.

7.3.1 Experiment 3

This trajectory, shown in Figure 10, was recorded so as to be able to assess the outdoor performance of the VIO algorithm. As stated before, several cycles were performed while passing at a control point at the trajectory starting position. The trajectory was resemblant of a slim rectangle wherein the setup endured mostly translational movement along the sides and mostly rotational movement at the corners, so that the cameras were kept at a locally forward-facing orientation.

As expected, the plots in Figure 11 correspond approximately to triangle waves of different amplitudes, a consequence of having approximately constant speed as well as of the rough East-West trajectory alignment.

It should be noted that a significant bias in the odometry estimates is introduced as time evolves, this can be seen by inspection of Figure 10, where similarly to the GPS curve in blue, approximately closed loops should be represented. Again, looking at the plots in Figure 11, this is especially clear in the VIO longitude curve in green, wherein the average value seems to follow a linearly increasing trend, resulting in minima/maxima at increasingly larger values, contrary to what should happen.

7.3.2 Experiment 4

The trajectory of the final project demonstration corresponded to the inspection of part of a viaduct, as shown in Figure 12. As such, it consisted on taking off, flying towards this structure and sweeping part of a pillar and the side of the deck at low speeds, followed by landing at a different position from the start.

Contrary to the previous trajectories, this was not performed so as to mimic simple geometric shapes, so that its analysis in terms of latitude and longitude plots is not as informative. Despite this fact, for consistency these plots are put forward in Figure 13.

Despite the difficulty in evaluating this trajectory, for which no ground truth or geometric constraints were available, there is something to be said about the estimated trajectories close to the bridge, where the GNSS signal reception was significantly degraded. The line in blue, corresponding to the GPS trajectory estimates, indicates that the UAV travelled as far as half the viaduct deck width, which was undoubtedly not the case, as observed on-site and made impossible by the 4m minimum distance enforced by the collision avoidance system onboard the UAV.
As such, for these regions, the VIO estimates were deemed more rigorous than those computed by GPS triangulation, and effectively contributed to an overall improvement of the estimated trajectory, shown in red and significantly more plausible than that in blue.

8. Conclusions
Having described in detail most of the aspects of this work, the final remarks are now put forward.

8.1. Achievements
A significant challenge within this work was the conception and development of the hardware setup. The implementation of a synchronization system, as well as its intrinsic and extrinsic calibration were successfully performed, effectively forming the base on top of which the system was to be built.

A visual-inertial odometry algorithm with state-of-the-art performance was first integrated with the sensor platform, for which the code based on the manufacturer’s APIs had to be adapted so as to use the ROS middleware. Its performance was quantitatively evaluated on a custom dataset with ground truth from a motion capture system, followed by the integration with the onboard autopilot and sensors.

This was performed through the implementation of a MAVLink communication link, used for bidirectional data and command transmission. Onboard sensor data was transmitted to the visual-inertial payload, and georeferenced VIO trajectory estimates were sent to the autopilot, as well as software trigger messages for the HD camera onboard the UAV.

The odometry estimates were georeferenced using the onboard GPS receiver to initialize the setup position, as well as the internal IMU attitude estimator to initialize its orientation.

Having concluded the system integration, the qualitative assessment of its performance was carried out through handheld testing. The in-flight performance was then validated and showcased at the final demonstration of the ELEVAR project.

Finally, the performance of the VIO algorithm was shown to be linked to the existence of visual texture at different depths. Despite this fact, the inclusion of a system of this type was shown to effectively increase the robustness of the onboard navigation system for the tests that were carried out, yielding significantly more rigorous pose estimates in circumstances for which GNSS signals were only available at a low sanity level.

8.2. Future Work
Seen as the ultimate goal of the ELEVAR project was to perform the inspection of structures through the stitching of photographs, a tighter integration with the navigation results would also be an interesting path to follow. Specifically, these could be used to trigger the camera so as to ensure specific frame superposition, avoiding the generation of redundant data as well as recognizing under-represented sections of the structure.

Regarding the navigation, a limitation of the approach in this work is the fact that no means of recovery from algorithm divergence were implemented. Along with more sophisticated ways of assessing algorithm performance in real-time, this would be an interesting way of augmenting the robustness of this system.

Another aspect where further work could be pursued is connected to the setup integration with the onboard autopilot. Instead of using GPS measurements and the IMU attitude estimator only at the initialization stage, a more sophisticated data fusion approach could be pursued. In fact, either using a dedicated filter to fuse all sources of information or including GPS triangulation terms into the overall cost function would be interesting ways of both correcting bias and allowing for the reset of the VIO algorithm in case of divergence.

Finally, the fact that no accurate ground truth was available for outdoors trajectories also posed an additional difficulty in the evaluation of results. Performing outdoors tests with ground truth from an additional system such as differential GPS would be of great value to the extension of this work, as it would enable the rigorous quantification of the impact the environment could have on algorithm performance.

Acknowledgements
I would like to thank all those who, directly or indirectly, may have contributed to the development of this work.

To my family, for their unwavering support and for actively encouraging me to work on things I am passionate about. To my friends, for all the thought-provoking discussions and fun times we have had.

To CMU Portugal, Alonzo Kelly, Michael George and Tekin Meriçli, for all their support during the
incredibly stimulating time I had at NREC. It was a time of intense learning, which definitely helped to establish my interest in this field.

To TEKEVER, for their help with the integration and willingness to generously share their resources. Both welcoming me to work at their offices and having Filipe Rosa working at ISR were without a doubt essential to the success of this work.

To Professor Ricardo Ribeiro, for his patience and having taken the time to teach me, and whose invaluable input and ideas were indispensable for the conclusion of this work.

To Professor Alexandre Bernardino, for steadily helping me along the way. For all the opportunities he gave me and the trust and expectations I feel he put in me. They helped me grow academically and as a person and for that I am truly grateful.

References


