Nonlinear Ego-Motion Estimation from Optical Flow for Online Control of a Quadrotor UAV

Volker Grabe, Heinrich H. Bülthoff, Davide Scaramuzza, and Paolo Robuffo Giordano

***Abstract*—For the control of Unmanned Aerial Vehicles (UAVs) in GPS-denied environments, cameras have been widely exploited as main sensory modality for addressing the UAV state estimation problem. However, the use of visual information for ego-motion estimation presents several theoretical and practical difficulties, such as data association, occlusions, and lack of direct metric information when exploiting monocular cameras. In this paper, we address these issues by considering a quadrotor UAV equipped with an onboard monocular camera and an Inertial Measurement Unit (IMU). First, we propose a robust ego-motion estimation algorithm for recovering the UAV scaled linear velocity and angular velocity from optical flow by exploiting the so-called *continuous* homography constraint in presence of planar scenes. Then, we address the problem of retrieving the (unknown) metric scale by fusing the visual information with measurements from the onboard IMU. To this end, two different estimation strategies are proposed and critically compared: a first one exploiting the classical Extended Kalman Filter (EKF) formulation, and a second one based on a novel nonlinear estimation framework. The main advantage of the latter scheme lies in the possibility of imposing a *desired* transient response to the estimation error when the camera moves with a constant acceleration norm w.r.t. the observed plane. We indeed show that, when compared against the EKF on the same trajectory and sensory data, the nonlinear scheme yields considerably superior performance in terms of convergence rate and predictability of the estimation. The paper is then concluded by an extensive experimental validation, including an onboard closed-loop control of a real quadrotor UAV meant to demonstrate the robustness of our approach in real-world conditions.**

I. INTRODUCTION

In recent years, inspection and search and rescue tasks have become one of the most important envisaged applications for Unmanned Aerial Vehicles (UAVs). For instance, small-size UAVs, such as quadrotors, have been used to investigate disaster sites after earthquakes, such as Christchurch (New Zealand) and Emilia Romagna (Italy), and most prominently the Fukushima Daiichi power plant in Japan. Along similar lines, some large-scale research projects have been recently funded on these and related topics, see, e.g., [1]–[3]. Indeed, thanks to their high agility, pervasiveness and customizability, small UAVs represent an ideal robotic platform for navigating in harsh and cluttered environments, when operating in either full autonomy or under the (partial) remote control of skilled human operators.

In all cases, a widely-recognized key component for the successful deployment of such systems is a reliable state estimation module able to deal with highly unstructured and/or GPS-denied indoor environments. This typically imposes a major requirement on the system: since the target environment for the considered applications cannot be prepared before the deployment of the vehicle, the UAV is constrained to only rely on onboard sensing and processing capabilities. These constraints have motivated, over the last years, the extensive use of onboard cameras as main sensory modality for state estimation purposes, see [4], [5] for a recent overview. Vision indeed provides a rich sensory feedback which could, in principle, yield a full understanding of the surrounding environment. However, as well-known, an effective use of the visual information also presents many theoretical and practical difficulties. For instance, robust data extraction and association across multiple frames is often a major issue in real-world scenarios, especially when comparing images from distant points of view. Also, when relying on monocular cameras (as in most typical settings), any position/velocity information can only be retrieved up to an arbitrary scale factor, which must then be disambiguated by fusing the visual information with independent (metric) measurements from other onboard sensors. This has motivated many recent works on data fusion exploiting the concurrent (metric) measurements from onboard accelerometers embedded in the Inertia Measurement Units (IMUs) present on most flying robotic systems [6]–[8].

Existing work on *metric* camera-based state estimation mostly relies on Simultaneous Localization and Mapping (SLAM) techniques to build and maintain a map of visual features, with the position of the vehicle in the map being estimated in parallel. Acceleration measurements are then used to reconstruct the metric scale of the underlying map [6]. This has been achieved in [9], [10] using visual-SLAM algorithms, such as PTAM [11] and SVO [12] respectively, and, even more remarkably, through the observation of just a single feature over time by providing a closed-form solution for the computation of the metric scale factor [6]. However, all these approaches depend on the possibility to continuously track features for an extended period of time. Therefore, the level of robustness required for the UAV control can be hard to guarantee as the visual system can be affected by, e.g., unexpected occlusions or the need of heavy computations for a reliable feature matching. This is not the case, however, when relying on motion estimation from *optical flow*, as, in this case, data extraction and association is performed on consecutive—and, thus, spatially very close-by—acquired images. Motivated

V. Grabe is with the Max Planck Institute for Biological Cybernetics, Spemannstraße 38, 72076 Tübingen, Germany and the University of Zurich, Andreasstrasse 15, 8050 Zurich, Switzerland. E-mail: volker.grabe@gmail.com.

H. H. Bülthoff is with the Max Planck Institute for Biological Cybernetics, Spemannstraße 38, 72076 Tübingen, Germany, and with the Department of Brain and Cognitive Engineering, Korea University, Seoul, 136-713 Korea. E-mail: hhb@tuebingen.mpg.de.

D. Scaramuzza is with the Robotics and Perception Group of the University of Zurich, Andreasstrasse 15, 8050 Zurich, Switzerland. E-mail: davide.scaramuzza@ieee.org.

P. Robuffo Giordano is with CNRS at Irisa and Inria Rennes Bretagne Atlantique, Campus de Beaulieu, 35042 Rennes Cedex, France prg@irisa.fr.

by these considerations, a first contribution of this paper is then the development of an approach based on optical flow decomposition for providing a reliable and robust ego-motion estimation module for safe UAV operation in unstructured environments.

Exploiting purely optical flow, a system capable of hovering and landing on a moving platform was presented in [13]. However, this work did not consider the issue of determining the unknown scene scale factor since the proposed control approach was implemented by only relying on the non-metric linear velocity directly obtained from a monocular camera. An approach combining optical flow decomposition and sensor fusion for metric scale estimation was instead presented in [14] by developing a small sensor for velocity estimation onboard UAVs. However, the proposed sensor relied on a ground facing sonar for metric scale estimation, thus limiting the vehicle to near ground operations within the range of the sonar. Finally, by exploiting an Extended Kalman Filter (EKF) to fuse optical flow measurements with IMU readings, in [15] the authors demonstrated the possibility of estimating the metric scale, sensor biases, and the IMU/camera relative pose. This system was later extended in [16]. However, the system was mainly designed for serving as an initialization of the PTAM framework in near hovering mode rather than for closed-loop UAV control over an extended period of time.

It can be noted that most of the presented camera-based state estimation methods rely on the classical EKF framework to fuse together the available sensor readings (e.g., from vision and IMU) for then extracting the metric scale. However, despite being widespread, the use of a EKF-based scheme presents two (often overlooked) drawbacks: (i) it necessarily involves a linearization of the (nonlinear) system dynamics (such as when dealing with visual-inertial estimation problems), and (ii) as a consequence, it does not allow for an explicit characterization of the estimation error behavior (e.g., its convergence rate). Therefore, as an additional contribution, in this paper we propose a novel visual-inertial estimation scheme exploiting optical flow and IMU measurements based on a recently-developed nonlinear observation framework for *active* structure from motion [17], [18]. Compared to a classical EKF, the use of this nonlinear filter yields an estimation error dynamics with a *fully characterized convergence behavior* when the camera is moving with a constant acceleration norm w.r.t. the observed scene. It is then possible, for instance, to *actively impose* a desired error transient response by suitably acting on the estimation gains and on the UAV motion, as well as to *predict* the convergence time of the estimation error in terms of percentages of the initial error. Finally, the reported results will also extensively show that the use of this nonlinear filter yields a substantial faster (and more controlled) error convergence compared to a classical and 'fully-informed' EKF, thus making it a viable and robust alternative to other consolidated schemes.

A. Summary of Contributions

Part of the work presented in this paper is based on [19]–[21]. In this paper, we nevertheless provide a more comprehensive overview with additional details and several extensions. In particular, we introduce the following novel contributions: (i) a full characterization of the estimation error transient reponse for the proposed nonlinear estimation scheme used to retrieve the metric scale from optical flow and IMU measurements. As explained, this analysis also determines the conditions on estimation gains and UAV motion which then allow to impose a *desired* convergence behavior to the scale estimation error; (ii) an extensive set of new simulative and experimental results investigating various aspects of the proposed ego-motion estimation scheme, including robustness against noise, trade-off between noise sensitivity and amount of 'control effort' (UAV linear acceleration), as well as an experimental validation of closed-loop velocity control with all computations run onboard and in real-time on a quadrotor UAV.

The rest of the paper is then structured as follows: in Sec. II we first review the proposed ego-motion estimation algorithm from optical flow which provides an estimation of the UAV (non-metric) linear and angular velocity. Subsequently, Sec. III introduces two estimation schemes meant to recover the unknown scale factor by fusing the visual information with the IMU readings: a filter based on the standard EKF machinery, and a novel deterministic nonlinear observer. The two filters are then compared by highlighting, for the latter, the possibility to characterize and actively shape its error transient response. Afterwards, Sec. IV reports and discusses the results of several simulations and experiments aimed at validating and comparing the two ego-motion estimation approaches, and, finally, some experiments of closed-loop control on a quadrotor UAV are presented. Section V then concludes the paper and discusses some open points and future directions.

II. EGO-MOTION ESTIMATION FROM OPTICAL FLOW

The approach adopted in this work for ego-motion estimation from the perceived optical flow is based on the decomposition of the so-called *continuous homography matrix* [22], complemented with the typical measurements obtainable from an onboard IMU. A distinguishing feature of our method w.r.t. most of the previous literature is the use of a *continuous approach* for motion recovery. In fact, the typical incremental ego-motion estimation algorithms (e.g., the so-called visual odometry [23]) assume presence of small but *finite* camera displacements over frames, and are thus based on reconstruction methods involving the *discrete* epipolar/homography constraints. However, since most cameras acquire images at high rates, we judged more appropriate to adopt a *continuous* point-of-view in order to recover, at each step, the camera instantaneous linear/angular velocity rather than a finite displacement over time.

In the following we summarize, for the reader's convenience, the main features of the employed ego-motion estimation method whose full details can be found in [19], [20].

A. Review of the Continuous Homography Constraint

Seen from a moving camera, the apparent velocity of a point $\boldsymbol{X} \in \mathbb{R}^3$ still in space as a result of the camera motion is

$$\dot{\boldsymbol{X}} = [\boldsymbol{\omega}]_\times \boldsymbol{X} + \boldsymbol{v} \tag{1}$$

where $\boldsymbol{v} \in \mathbb{R}^3$, $\boldsymbol{\omega} \in \mathbb{R}^3$ are the camera linear/angular velocity (both expressed in the camera frame), and $[\boldsymbol{\omega}]_\times \in so(3)$ is the skew-symmetric matrix associated to vector $\boldsymbol{\omega} \in \mathbb{R}^3$.

Consider a set of point features located on a common plane of equation $\boldsymbol{n}^T\boldsymbol{X} = d$ where $\boldsymbol{n} \in \mathbb{S}^2$ is the unit normal vector to the plane, and $d \in \mathbb{R}$ the orthogonal distance of the plane to the camera frame. By rearranging the plane constraint as $\frac{1}{d}\boldsymbol{n}^T\boldsymbol{X} = 1$, eq. (1) becomes

$$\dot{\boldsymbol{X}} = [\boldsymbol{\omega}]_\times \boldsymbol{X} + \boldsymbol{v}\frac{1}{d}\boldsymbol{n}^T\boldsymbol{X} = \left([\boldsymbol{\omega}]_\times + \frac{1}{d}\boldsymbol{v}\boldsymbol{n}^T\right)\boldsymbol{X} = \boldsymbol{H}\boldsymbol{X}. \tag{2}$$

Matrix $\boldsymbol{H} \in \mathbb{R}^{3\times3}$ is commonly referred to as the *continuous homography matrix*: it encodes the camera linear/angular velocity $(\boldsymbol{v}, \boldsymbol{\omega})$, as well as the scene structure $(\boldsymbol{n}, d)$.

Defining $\lambda\boldsymbol{x} = \boldsymbol{X}$ for a scalar depth factor λ as the image of a point $\boldsymbol{X}$, and exploiting the fact that $\dot{\boldsymbol{X}} = \dot{\lambda}\boldsymbol{x} + \lambda\boldsymbol{u}$ and $\dot{\boldsymbol{x}} = \boldsymbol{u}$, where $\boldsymbol{u}$ is the *observed* velocity of the point $\boldsymbol{x}$ on the image plane, one obtains

$$\boldsymbol{u} = \boldsymbol{H}\boldsymbol{x} - \frac{\dot{\lambda}}{\lambda}\boldsymbol{x}. \tag{3}$$

The depth factor λ in (3) can be removed by pre-multiplication of $[\boldsymbol{x}]_\times$. This results in the so-called *continuous homography constraint* [22]

$$[\boldsymbol{x}]_\times \boldsymbol{H}\boldsymbol{x} = [\boldsymbol{x}]_\times \boldsymbol{u} \tag{4}$$

which involves the measured $(\boldsymbol{x}, \boldsymbol{u})$ and the (unknown) continuous homography matrix $\boldsymbol{H}$.

B. The 4-Point Algorithm (Algorithm V1)

Matrix $\boldsymbol{H}$ in (4) can be recovered from a set of N measured pairs $(\boldsymbol{x}_i, \boldsymbol{u}_i)$ of detected features $\boldsymbol{x}_i \in \mathbb{R}^3$ on the image plane and associated feature velocities $\boldsymbol{u}_i \in \mathbb{R}^3$. This can be easily obtained as follows: by stacking the elements of $\boldsymbol{H}$ into the vector $\boldsymbol{H}^S = [H_{11}, H_{21}, \cdots, H_{33}] \in \mathbb{R}^9$, one can rewrite (4) as

$$\boldsymbol{a}_i^T \boldsymbol{H}^S = [\boldsymbol{x}_i]_\times \boldsymbol{u}_i \tag{5}$$

where $\boldsymbol{a}_i = \boldsymbol{x}_i \otimes [\boldsymbol{x}_i]_\times \in \mathbb{R}^{9\times3}$, and $\otimes$ stands for the Kronecker product. By then defining $\boldsymbol{A} = [\boldsymbol{a}_1, \cdots, \boldsymbol{a}_N]^T \in \mathbb{R}^{3N\times9}$ and $\boldsymbol{B} = [[\boldsymbol{x}_1]_\times \boldsymbol{u}_1, \cdots, [\boldsymbol{x}_N]_\times \boldsymbol{u}_N]^T \in \mathbb{R}^{3N}$, one obtains the linear system

$$\boldsymbol{A}\boldsymbol{H}^S = \boldsymbol{B}. \tag{6}$$

Assuming presence of at least $N \geq 4$ measured pairs $(\boldsymbol{x}_i, \boldsymbol{u}_i)$, system (6) can be solved in a least-square sense as $\boldsymbol{H}^S = \boldsymbol{A}^\dagger\boldsymbol{B}$, with $\boldsymbol{A}^\dagger$ denoting the pseudo-inverse of matrix $\boldsymbol{A}$.

After having recovered $\boldsymbol{H}$, using standard techniques [22] it is further possible to algebraically decompose it into the scaled linear velocity $\boldsymbol{v}/d$, the angular velocity $\boldsymbol{\omega}$, and the plane normal $\boldsymbol{n}$. However, it can be shown that, in general, two physically-equivalent solutions are compatible with a given homography matrix $\boldsymbol{H}$.

Thanks to this decomposition, one can then obtain the quantities $(\boldsymbol{v}/d, \boldsymbol{\omega}, \boldsymbol{n})$ from two consecutive visual observations and without the use of additional sensor readings or previously acquired data structures such as, e.g., a map. This first version of the ego-motion algorithm will be referred to as *V1* in all the following developments.

C. Exploiting the known Angular Velocity (Algorithm V2)

Since any typical onboard IMU can directly measure the angular velocity $\boldsymbol{\omega}_{IMU}$, we can consider $\boldsymbol{\omega} = \boldsymbol{\omega}_{IMU}$ as *known* from external (i.e., not vision-based) sources. Knowledge of $\boldsymbol{\omega}$ can then be used to derotate the perceived optical flow field as

$$\begin{bmatrix} u'_x \\ u'_y \end{bmatrix} = \begin{bmatrix} u_x \\ u_y \end{bmatrix} - \begin{bmatrix} -x_x x_y & 1+x_x^2 & -x_y \\ -(1+x_y)^2 & x_x x_y & x_x \end{bmatrix} \boldsymbol{\omega}, \tag{7}$$

where the interaction matrix relating $\boldsymbol{u}$ to $(\boldsymbol{v}, \boldsymbol{\omega})$ was exploited [24]. This derotation step then reduces matrix $\boldsymbol{H}$ to

$$\boldsymbol{H} = \frac{1}{d}\boldsymbol{v}\boldsymbol{n}^T. \tag{8}$$

Since $\boldsymbol{n}$ spans $\boldsymbol{H}^T$ and $\|\boldsymbol{n}\| = 1$, we can obtain $\boldsymbol{n}$ from the singular value decomposition $\boldsymbol{H} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$ as the first column of matrix $\boldsymbol{V}$. The inherent sign ambiguity can be resolved by enforcing $\boldsymbol{n}_z > 0$. Having retrieved $\boldsymbol{n}$, we then obtain $\boldsymbol{v}/d = \boldsymbol{H}\boldsymbol{n}$.

This algorithm, which will be referred to as *V2* in the following, requires the observation of at least three feature pairs $(\boldsymbol{x}_i, \boldsymbol{u}_i)$ and yields a *unique* solution for $\boldsymbol{v}/d$ and $\boldsymbol{n}$.

D. Exploiting the known Angular Velocity and Plane Orientation (Algorithm V3)

In most indoor environments and when considering UAVs with downlooking cameras, the dominant plane can be safely taken as horizontal with, thus, its normal vector $\boldsymbol{n}$ parallel to the gravity vector. Therefore, one can exploit the ability of onboard IMUs to estimate (via internal filtering) the local gravity vector, thus allowing to consider $\boldsymbol{n} \approx \boldsymbol{n}_{IMU}$ as measured independently from the visual input.

Plugging (8) in (4) yields

$$[\boldsymbol{x}]_\times \frac{1}{d}\boldsymbol{v} = \frac{[\boldsymbol{x}]_\times \boldsymbol{u}}{\boldsymbol{n}^T\boldsymbol{x}}. \tag{9}$$

where now $\boldsymbol{n}^T\boldsymbol{x}$ is a known quantity. Letting $\boldsymbol{\delta} = ([\boldsymbol{x}]_\times \boldsymbol{u})/(\boldsymbol{n}^T\boldsymbol{x}) \in \mathbb{R}^3$, one then obtains the following equation linear in $\boldsymbol{v}/d$ (the only unknown left)

$$[\boldsymbol{x}_i]_\times \frac{\boldsymbol{v}}{d} = \boldsymbol{\delta}_i, \qquad i = 1 \ldots N. \tag{10}$$

A least-square approximation of $\boldsymbol{v}/d$ over all N tracked features can be obtained by stacking all $[\boldsymbol{x}_i]_\times$ into the matrix $\boldsymbol{\Gamma} = [[\boldsymbol{x}_1]_\times, \cdots, [\boldsymbol{x}_N]_\times]^T \in \mathbb{R}^{3N\times3}$ and all $\boldsymbol{\delta}_i$ into the vector $\boldsymbol{\Delta} = [\boldsymbol{\delta}_1, \cdots, \boldsymbol{\delta}_N]^T \in \mathbb{R}^{3N}$ resulting in the linear system

$$\boldsymbol{\Gamma}\frac{\boldsymbol{v}}{d} = \boldsymbol{\Delta} \tag{11}$$

which can be solved as $\boldsymbol{v}/d = \boldsymbol{\Gamma}^{\dagger}\boldsymbol{\Delta}$.

Note that any two distinct feature point vectors $\boldsymbol{x}_i, \boldsymbol{x}_j$ will never be parallel due to the perspective projection of the camera. Thus, in principle, only two flow vectors are required to obtain a solution for $\boldsymbol{v}/d$. However, a more robust estimation is of course obtained by incorporating all observed flow vectors.

This third algorithm will be referred to as *V3* in the following.

E. Segmentation of Features

In order to further improve the robustness of the velocity estimation system, we developed a thresholding method to test whether a set of features belongs to a common plane, and whether a new feature is a member of an existing plane [20].

This method is based on the observation that the 'reprojection' vector

$$\boldsymbol{E} = \boldsymbol{B} - \boldsymbol{A}\boldsymbol{H}^S = \boldsymbol{B} - \boldsymbol{A}\boldsymbol{A}^{\dagger}\boldsymbol{B} \in \mathbb{R}^{3N}$$

must vanish when all observed features belong to a common plane, since (6) and its variants admit an exact solution only in this case. Therefore, the quantity $\|\boldsymbol{E}\|/N$ (where N is the number of tracked features) can be taken as a 'planarity measure' to decide whether the observed features belong to a common plane. This approach can then be generalized to test newly tracked features against a previously known plane.

Use of this thresholding method can clearly improve the robustness of the velocity estimation algorithms presented in the previous section, since outliers (of the planarity assumption) can be more easily rejected from the maintained set of visual features. Therefore, we proposed in [20] a RANSAC-based approach meant to find an initial set of features belonging to a predominant plane which is then extended as the UAV explores the scene. In parallel, the system constantly monitors the set of rejected features to detect new or more consistent planes in the environment. To maintain the quality of the feature set, the agreement of each feature with the main segmented plane is also constantly monitored, with outliers being removed as the camera traverses the environment.

Using this filtering technique, we showed in [20] that the estimation of the scaled linear velocity $\boldsymbol{v}/d$ could be improved by a factor of 25% in presence of a dominant ground plane partially covered by additional objects compared to an unfiltered version of the algorithm. This then validated the use of the proposed ego-motion estimation algorithms (which are based on the continuous *homography* constraint) also in non-perfectly planar scenes as those one can typically find in indoor environments.

F. Final Discussion

We conclude with some final considerations about the choice of basing the proposed ego-motion estimation algorithm upon the *homography* constraint instead of the other well-known *epipolar* constraint often exploited in computer and robotic vision. In general, the applicability of the homography vs. epipolar constraint depends on the structure of the environment: methods relying on the *epipolar constraint* are more appropriate for highly unstructured scenes where a collection of points in *general position* can always be detected[1]. On the other hand, solutions based on the *homography constraint* should be favored when dealing with approximately planar scenes. Indeed, in the ideal case of features extracted from a perfect planar scene, the epipolar constraint 'loses rank' and one cannot obtain any longer a unique solution (up to a scalar factor) for the ego-motion recovery [22]. The homography constraint, on the other hand, allows to correctly address the case of planar scenes (which is the case considered in this paper).

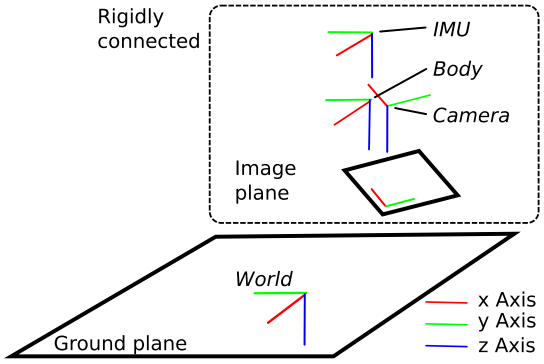


Fig. 1: Locations of the IMU ($\mathcal{I}$), camera ($\mathcal{C}$), body ($\mathcal{B}$) and world frames ($\mathcal{W}$) relative to each other. Frames $\mathcal{I}, \mathcal{B}$ and $\mathcal{C}$ are assumed to be rigidly linked to each other. The world frame $\mathcal{W}$ is oriented horizontally with its z-axis pointing down, following the NED convention.

Furthermore, compared to epipolar-based algorithms, the homography constraint remains better conditioned also in case of stationary flight, e.g., with a small amount of translational motion (indeed, the epipolar constraint vanishes for a zero translation/linear velocity). Therefore, while the epipolar constraint can be more suited for persistently-translating maneuvers or vehicles (e.g., fixed-wing), the homography constraint ought to be more robust for typical indoor (close to stationary) flight regimes of vertical take-off and landing vehicles such as quadrotor UAVs.

These considerations then motivated our choice of the *homography* constraint for dealing with the ego-motion recovery. Indeed, as explained, in many scenarios, such as indoor hallways or areal coverage, one can safely assume presence of a dominant ground plane spanning most of the observed features. Coupled with the RANSAC-based outlier rejection described in the previous section, our proposed solution then proved to yield accurate results as it will be discussed in the next Sect. IV-B.

III. ONLINE SCALE ESTIMATION

The ego-motion algorithm (in its three variants) presented in the previous section allows to obtain a reliable estimation of the scaled camera linear velocity $\boldsymbol{v}/d$. In this section, we discuss two estimation schemes meant to recover the plane distance d by fusing the optical flow decomposition with the onboard accelerometer measurements.

[1]Specifically, the points must not lie on some particular quadratic surfaces which, however, include planes as special case, see [22].

A. Equations of Motion

We start deriving the equations of motion relevant to the case under consideration. In the following, we will denote with $\mathcal{B}, \mathcal{C}, \mathcal{I}$ and $\mathcal{W}$ the body, camera, IMU and inertial world frames, respectively. The origin of frame $\mathcal{B}$ is assumed to be located at the quadrotor barycenter, while frames $\mathcal{C}$ and $\mathcal{I}$ are supposed to be rigidly attached to $\mathcal{B}$, see Fig. 1. Throughout the text, left superscripts will be exploited to indicate the frames where quantities are expressed in. The symbol ${}^{\mathcal{X}}\boldsymbol{R}_{\mathcal{Y}} \in SO(3)$ will be used to denote the rotation matrix from frame $\mathcal{X}$ to frame $\mathcal{Y}$, and ${}^{\mathcal{Z}}\boldsymbol{p}_{\mathcal{X}\mathcal{Y}} \in \mathbb{R}^3$ to represent the vector from the origin of frame $\mathcal{X}$ to the origin of frame $\mathcal{Y}$, expressed in frame $\mathcal{Z}$. We also introduce the following quantities instrumental for the next developments: $\boldsymbol{g} \in \mathbb{R}^3$ as the gravity vector, and ${}^{\mathcal{I}}\boldsymbol{f} \in \mathbb{R}^3$, ${}^{\mathcal{I}}\boldsymbol{\omega} \in \mathbb{R}^3$ as the specific acceleration and angular velocity at the origin of $\mathcal{I}$.

Define ${}^{\mathcal{C}}\boldsymbol{v} = {}^{\mathcal{C}}\boldsymbol{R}_{\mathcal{W}}\, {}^{\mathcal{W}}\dot{\boldsymbol{p}}_{\mathcal{W}\mathcal{C}}$ as the camera linear velocity in camera frame, and ${}^{\mathcal{B}}\boldsymbol{v} = {}^{\mathcal{B}}\boldsymbol{R}_{\mathcal{W}}\, {}^{\mathcal{W}}\dot{\boldsymbol{p}}_{\mathcal{W}\mathcal{B}}$ as the body linear velocity in body frame. Since ${}^{\mathcal{B}}\boldsymbol{R}_{\mathcal{C}}$, ${}^{\mathcal{B}}\boldsymbol{R}_{\mathcal{I}}$, ${}^{\mathcal{B}}\boldsymbol{p}_{\mathcal{B}\mathcal{C}}$ and ${}^{\mathcal{B}}\boldsymbol{p}_{\mathcal{B}\mathcal{I}}$ are assumed constant, from standard kinematics the following relationships hold

$$
\begin{aligned}
{}^{\mathcal{B}}\boldsymbol{v} &= {}^{\mathcal{B}}\boldsymbol{R}_{\mathcal{C}}({}^{\mathcal{C}}\boldsymbol{v} + [{}^{\mathcal{C}}\boldsymbol{\omega}]_{\times}\, {}^{\mathcal{C}}\boldsymbol{p}_{\mathcal{C}\mathcal{B}}) = {}^{\mathcal{B}}\boldsymbol{R}_{\mathcal{C}}\, {}^{\mathcal{C}}\boldsymbol{v} + [{}^{\mathcal{B}}\boldsymbol{\omega}]_{\times}\, {}^{\mathcal{B}}\boldsymbol{p}_{\mathcal{C}\mathcal{B}}, \quad (12)\\
{}^{\mathcal{C}}\dot{\boldsymbol{v}} &= {}^{\mathcal{C}}\boldsymbol{R}_{\mathcal{I}}({}^{\mathcal{I}}\boldsymbol{a} + [{}^{\mathcal{I}}\dot{\boldsymbol{\omega}}]_{\times}\, {}^{\mathcal{I}}\boldsymbol{p}_{\mathcal{I}\mathcal{C}} + [{}^{\mathcal{I}}\boldsymbol{\omega}]_{\times}^2\, {}^{\mathcal{I}}\boldsymbol{p}_{\mathcal{I}\mathcal{C}}) - [{}^{\mathcal{C}}\boldsymbol{\omega}]_{\times}\, {}^{\mathcal{C}}\boldsymbol{v}\\
&= {}^{\mathcal{C}}\boldsymbol{R}_{\mathcal{I}}\, {}^{\mathcal{I}}\boldsymbol{a} + [{}^{\mathcal{C}}\dot{\boldsymbol{\omega}}]_{\times}\, {}^{\mathcal{C}}\boldsymbol{p}_{\mathcal{I}\mathcal{C}} + [{}^{\mathcal{C}}\boldsymbol{\omega}]_{\times}^2\, {}^{\mathcal{C}}\boldsymbol{p}_{\mathcal{I}\mathcal{C}} - [{}^{\mathcal{C}}\boldsymbol{\omega}]_{\times}\, {}^{\mathcal{C}}\boldsymbol{v}\\
&= {}^{\mathcal{C}}\boldsymbol{\kappa} - [{}^{\mathcal{C}}\boldsymbol{\omega}]_{\times}\, {}^{\mathcal{C}}\boldsymbol{v} \quad (13)\\
{}^{\mathcal{C}}\boldsymbol{\omega} &= {}^{\mathcal{C}}\boldsymbol{R}_{\mathcal{I}}\, {}^{\mathcal{I}}\boldsymbol{\omega} \quad (14)\\
{}^{\mathcal{C}}\dot{\boldsymbol{\omega}} &= {}^{\mathcal{C}}\boldsymbol{R}_{\mathcal{I}}\, {}^{\mathcal{I}}\dot{\boldsymbol{\omega}} \quad (15)
\end{aligned}
$$

where ${}^{\mathcal{I}}\boldsymbol{a} = {}^{\mathcal{I}}\boldsymbol{R}_{\mathcal{W}}\, {}^{\mathcal{W}}\ddot{\boldsymbol{p}}_{\mathcal{W}\mathcal{I}}$ is the linear acceleration experienced by the IMU and

$$
{}^{\mathcal{C}}\boldsymbol{\kappa} = {}^{\mathcal{C}}\boldsymbol{R}_{\mathcal{I}}\, {}^{\mathcal{I}}\boldsymbol{a} + [{}^{\mathcal{C}}\dot{\boldsymbol{\omega}}]_{\times}\, {}^{\mathcal{C}}\boldsymbol{p}_{\mathcal{I}\mathcal{C}} + [{}^{\mathcal{C}}\boldsymbol{\omega}]_{\times}^2\, {}^{\mathcal{C}}\boldsymbol{p}_{\mathcal{I}\mathcal{C}} \quad (16)
$$

is the camera linear acceleration w.r.t. $\mathcal{W}$. We note that ${}^{\mathcal{I}}\boldsymbol{a} = {}^{\mathcal{I}}\boldsymbol{f} + {}^{\mathcal{I}}\boldsymbol{g}$ and ${}^{\mathcal{I}}\boldsymbol{g} = {}^{\mathcal{I}}\boldsymbol{R}_{\mathcal{W}}[0, 0, g]^T$ in case of a horizontal orientation of the world frame, see Fig. 1.

In presence of a planar scene ${}^{\mathcal{C}}\boldsymbol{n}^T\, {}^{\mathcal{C}}\boldsymbol{P} + d = 0$ one also has (see, e.g., [25])

$$
\begin{aligned}
{}^{\mathcal{C}}\dot{\boldsymbol{n}} &= -[{}^{\mathcal{C}}\boldsymbol{\omega}]_{\times}\, {}^{\mathcal{C}}\boldsymbol{n} \quad (17)\\
\dot{d} &= {}^{\mathcal{C}}\boldsymbol{v}^T\, {}^{\mathcal{C}}\boldsymbol{n}. \quad (18)
\end{aligned}
$$

Finally, according to this notation, the decomposition of the optical flow summarized in the previous Section allows to directly measure the scaled linear velocity ${}^{\mathcal{C}}\tilde{\boldsymbol{v}} = {}^{\mathcal{C}}\boldsymbol{v}/d$. The estimation schemes presented in the following are then meant to recover the (unmeasurable) value of the plane distance d and the metric linear velocity vector ${}^{\mathcal{C}}\boldsymbol{v}$.

B. Scale Ambiguity

As in any monocular Structure from Motion (SfM) problem, the inherent *scale ambiguity* of the observed scene can be resolved only if the camera motion is sufficiently exciting w.r.t. the estimation task. For instance, the depth of a point feature can be recovered only if the camera travels with a non-zero linear velocity not aligned with the projection ray of the observed point, and other constraints exist for different geometrical primitives [18].

In the case under consideration, a camera traveling with zero linear acceleration (${}^{\mathcal{C}}\boldsymbol{\kappa} = \boldsymbol{0}$) does not allow to infer the plane distance d and the linear velocity ${}^{\mathcal{C}}\boldsymbol{v}$ from the measured ${}^{\mathcal{C}}\boldsymbol{v}/d$: an accelerated motion w.r.t. the observed plane is necessary in order to recover the correct pair $({}^{\mathcal{C}}\boldsymbol{v},\, d)$. This (known) requirement can be briefly justified as follows: by setting ${}^{\mathcal{C}}\boldsymbol{\kappa} = 0$, the dynamics (13)–(18) reduce to

$$
\begin{cases}
{}^{\mathcal{C}}\dot{\boldsymbol{v}} &= -[{}^{\mathcal{C}}\boldsymbol{\omega}]_{\times}\, {}^{\mathcal{C}}\boldsymbol{v}\\
\dot{d} &= {}^{\mathcal{C}}\boldsymbol{v}^T\, {}^{\mathcal{C}}\boldsymbol{n}
\end{cases}, \quad (19)
$$

that is, an expression homogeneous in the pair $({}^{\mathcal{C}}\boldsymbol{v},\, d)$. Therefore, given a solution $({}^{\mathcal{C}}\boldsymbol{v}^*(t),\, d^*(t))$ of system (19), any other pair $(\lambda\, {}^{\mathcal{C}}\boldsymbol{v}^*(t),\, \lambda d^*(t))$ is also a valid solution, with $\lambda \in \mathbb{R}$ being an arbitrary scale factor. The measured output ${}^{\mathcal{C}}\boldsymbol{v}/d$, however, does *not* depend on the scale factor λ: hence, whatever the adopted filter, a non-accelerating camera cannot disambiguate the correct pair $({}^{\mathcal{C}}\boldsymbol{v},\, d)$ from the perceived optical flow as any other pair $(\lambda\, {}^{\mathcal{C}}\boldsymbol{v}(t),\, \lambda d(t))$ would be equally consistent with the dynamics (19) and the measured ${}^{\mathcal{C}}\boldsymbol{v}/d$. On the other hand, presence of a non-zero (and known) linear acceleration ${}^{\mathcal{C}}\boldsymbol{\kappa} \neq \boldsymbol{0}$ breaks the homogeneity of (19) w.r.t. $({}^{\mathcal{C}}\boldsymbol{v},\, d)$ and makes it possible to recover the correct scale factor λ during motion. Section III-D will revisit these considerations in the context of Persistency of Excitation of the camera motion during the estimation task.

C. Scale Estimation based on the Extended Kalman Filter

As a first approach to estimate the distance to the planar scene d, we develop a classical EKF upon the equations of motion of the system. In particular, we adopt the discrete version of the EKF, and let index $k \in \mathbb{N}$ denote the k-th iteration step. For clarity, we will append a right subscript m to identify all those quantities that are directly available through one of the onboard sensors, e.g., specific force ${}^{\mathcal{I}}\boldsymbol{f}_m$ and angular velocity ${}^{\mathcal{I}}\boldsymbol{\omega}_m$ from the IMU, and the scaled linear velocity ${}^{\mathcal{C}}\tilde{\boldsymbol{v}}_m = ({}^{\mathcal{C}}\boldsymbol{v}/d)_m$ from the camera.

We define the EKF state vector $\boldsymbol{x}$ to consist of the metric camera linear velocity in camera frame ${}^{\mathcal{C}}\boldsymbol{v}$ and the *inverse* $\rho = 1/d$ of the camera distance to the planar scene:

$$
\boldsymbol{x} = \begin{bmatrix} {}^{\mathcal{C}}\boldsymbol{v} \\ \rho \end{bmatrix}, \qquad {}^{\mathcal{C}}\boldsymbol{v} \in \mathbb{R}^3, \rho \in \mathbb{R}. \quad (20)
$$

The choice of taking an *inverse parameterization* $\rho = 1/d$ (in place of just d) for the scene scale is indeed known to yield better results in the context of structure estimation from motion, see, e.g., [26], [27]. This inverse parameterization will also be instrumental in the design of the nonlinear observer of the next Sec. III-D.

Since the onboard IMU directly outputs the specific acceleration ${}^{\mathcal{I}}\boldsymbol{f}_m$ and and angular velocity ${}^{\mathcal{I}}\boldsymbol{\omega}_m$ in its own frame[2],

[2]We assume the IMU is able to self-calibrate by an initial estimation of the measurement biases (indeed, this is often the case of many commercial IMUs, including the one employed in the experiments reported in Sect. IV). The interested reader is referred to, e.g., [4], [6], [15] for other estimation approaches addressing the concurrent IMU bias and structure estimation during flight.

we rewrite (13) as:

$$\begin{aligned}{}^{\mathcal{C}}\dot{\boldsymbol{v}} &= {}^{\mathcal{C}}\boldsymbol{R}_{\mathcal{I}}({}^{\mathcal{I}}\boldsymbol{a} + [{}^{\mathcal{I}}\dot{\boldsymbol{\omega}}]_{\times}\,{}^{\mathcal{I}}\boldsymbol{p}_{\mathcal{IC}} + [{}^{\mathcal{I}}\boldsymbol{\omega}_m]_{\times}^2\,{}^{\mathcal{I}}\boldsymbol{p}_{\mathcal{IC}}) - [{}^{\mathcal{C}}\boldsymbol{\omega}_m]_{\times}\,{}^{\mathcal{C}}\boldsymbol{v} \\ &\approx {}^{\mathcal{C}}\boldsymbol{R}_{\mathcal{I}}({}^{\mathcal{I}}\boldsymbol{f}_m + {}^{\mathcal{I}}\boldsymbol{g} + [{}^{\mathcal{I}}\boldsymbol{\omega}_m]_{\times}^2\,{}^{\mathcal{I}}\boldsymbol{p}_{\mathcal{IC}}) - [{}^{\mathcal{C}}\boldsymbol{\omega}_m]_{\times}\,{}^{\mathcal{C}}\boldsymbol{v}. \quad (21)\end{aligned}$$

As no direct measurement of ${}^{\mathcal{I}}\dot{\boldsymbol{\omega}}$ is possible on most quadrotor setups, and being ${}^{\mathcal{I}}\boldsymbol{\omega}$ usually a noisy signal, we approximate ${}^{\mathcal{I}}\dot{\boldsymbol{\omega}} \approx \boldsymbol{0}$ in (21) rather than attempting to recover ${}^{\mathcal{I}}\dot{\boldsymbol{\omega}}$ via a numerical differentiation. Exploiting (18), the dynamics of ρ is instead given by

$$\dot{\rho} = -\frac{\dot{d}}{d^2} = -\rho^2\,{}^{\mathcal{C}}\boldsymbol{v}^T\,{}^{\mathcal{C}}\boldsymbol{n}. \quad (22)$$

Consequently, the following equations govern the predicted state $\bar{\boldsymbol{x}}[k]$ in terms of the previous estimated state $\hat{\boldsymbol{x}}[k-1]$:

$${}^{\mathcal{C}}\bar{\boldsymbol{v}}[k] = {}^{\mathcal{C}}\hat{\boldsymbol{v}}[k-1] + T\,{}^{\mathcal{C}}\dot{\boldsymbol{v}}[k-1] \quad (23)$$

$$\bar{\rho}[k] = \hat{\rho}[k-1] - T\hat{\rho}^2[k-1]\,{}^{\mathcal{C}}\hat{\boldsymbol{v}}[k-1]^T\,{}^{\mathcal{C}}\boldsymbol{n}[k-1] \quad (24)$$

where T denotes the sampling time of the filter.

Although most quantities derived in the following steps are time varying, from now on, for the sake of exposition clarity, we will omit the time dependency $[k]$ wherever possible.

To compute the predicted covariance matrix of the system uncertainty $\bar{\boldsymbol{\Sigma}}[k] \in \mathbb{R}^{4\times4}$, we first derive the Jacobian matrix $\boldsymbol{G}[k] \in \mathbb{R}^{4\times4}$

$$\begin{aligned}\boldsymbol{G} &= \begin{bmatrix} \dfrac{\partial\,{}^{\mathcal{C}}\bar{\boldsymbol{v}}[k]}{\partial\,{}^{\mathcal{C}}\hat{\boldsymbol{v}}[k-1]} & \dfrac{\partial\,{}^{\mathcal{C}}\bar{\boldsymbol{v}}[k]}{\partial\hat{\rho}[k-1]} \\ \dfrac{\partial\bar{\rho}[k]}{\partial\,{}^{\mathcal{C}}\hat{\boldsymbol{v}}[k-1]} & \dfrac{\partial\bar{\rho}[k]}{\partial\hat{\rho}[k-1]} \end{bmatrix} \\ &= \begin{bmatrix} \boldsymbol{I}_3 - T[{}^{\mathcal{I}}\boldsymbol{\omega}_m]_{\times} & \boldsymbol{0}_{3\times1} \\ -T\hat{\rho}^2\,{}^{\mathcal{C}}\boldsymbol{n}^T & 1 - 2T\hat{\rho}\,{}^{\mathcal{C}}\hat{\boldsymbol{v}}^T\,{}^{\mathcal{C}}\boldsymbol{n} \end{bmatrix}. \quad (25)\end{aligned}$$

Matrix $\hat{\boldsymbol{\Sigma}}[k-1]$ from the previous step is then propagated as:

$$\bar{\boldsymbol{\Sigma}}[k] = \boldsymbol{G}\hat{\boldsymbol{\Sigma}}[k-1]\boldsymbol{G}^T + \boldsymbol{R}. \quad (26)$$

Here, matrix $\boldsymbol{R} \in \mathbb{R}^{4\times4}$ is obtained from

$$\boldsymbol{R} = \boldsymbol{V}\begin{bmatrix} \mathrm{cov}({}^{\mathcal{I}}\boldsymbol{f}_{\mathrm{m}}) & \boldsymbol{0}_{3\times3} \\ \boldsymbol{0}_{3\times3} & \mathrm{cov}({}^{\mathcal{I}}\boldsymbol{\omega}_{\mathrm{m}}) \end{bmatrix}\boldsymbol{V}^T \quad (27)$$

where

$$\begin{aligned}\boldsymbol{V} &= \begin{bmatrix} \dfrac{\partial\,{}^{\mathcal{C}}\bar{\boldsymbol{v}}[k]}{\partial\,{}^{\mathcal{I}}\boldsymbol{f}_m} & \dfrac{\partial\,{}^{\mathcal{C}}\bar{\boldsymbol{v}}[k]}{\partial\,{}^{\mathcal{I}}\boldsymbol{\omega}_m} \\ \dfrac{\partial\bar{\rho}[k]}{\partial\,{}^{\mathcal{I}}\boldsymbol{f}_m} & \dfrac{\partial\bar{\rho}[k]}{\partial\,{}^{\mathcal{I}}\boldsymbol{\omega}_m} \end{bmatrix} \\ &= \begin{bmatrix} T\,{}^{\mathcal{C}}\boldsymbol{R}_{\mathcal{I}} & T({}^{\mathcal{C}}\boldsymbol{R}_{\mathcal{I}}\boldsymbol{M} + [{}^{\mathcal{C}}\hat{\boldsymbol{v}}]_{\times}\,{}^{\mathcal{C}}\boldsymbol{R}_{\mathcal{I}}) \\ \boldsymbol{0}_{1\times3} & \boldsymbol{0}_{1\times3} \end{bmatrix} \in \mathbb{R}^{4\times6} \quad (28)\end{aligned}$$

$$\boldsymbol{M} = ({}^{\mathcal{I}}\boldsymbol{\omega}_m^T\,{}^{\mathcal{I}}\boldsymbol{p}_{\mathcal{IC}})\boldsymbol{I}_3 + {}^{\mathcal{I}}\boldsymbol{\omega}_m\,{}^{\mathcal{I}}\boldsymbol{p}_{\mathcal{IC}}^T - 2\,{}^{\mathcal{I}}\boldsymbol{p}_{\mathcal{IC}}\,{}^{\mathcal{I}}\boldsymbol{\omega}_m^T, \quad (29)$$

and $\mathrm{cov}({}^{\mathcal{I}}\boldsymbol{f}_{\mathrm{m}}) \in \mathbb{R}^{3\times3}$, $\mathrm{cov}({}^{\mathcal{I}}\boldsymbol{\omega}_{\mathrm{m}}) \in \mathbb{R}^{3\times3}$ are the covariance matrixes of the accelerometers/gyroscopes sensors in the IMU, which can be experimentally determined.

The predicted state $\bar{\boldsymbol{x}}$ is then updated whenever a new scaled visual velocity estimate $\boldsymbol{z}_m = {}^{\mathcal{C}}\tilde{\boldsymbol{v}}_m = \left({}^{\mathcal{C}}\boldsymbol{v}/d\right)_m$ becomes available from the optical flow decomposition. Let $\bar{\boldsymbol{z}}$ be the predicted scaled visual velocity estimation based on the predicted state $\bar{\boldsymbol{x}}$

$$\bar{\boldsymbol{z}} = {}^{\mathcal{C}}\bar{\boldsymbol{v}}\bar{\rho}. \quad (30)$$

The Kalman gain $\boldsymbol{K} \in \mathbb{R}^{4\times3}$ is obtained as

$$\boldsymbol{K} = \bar{\boldsymbol{\Sigma}}\boldsymbol{J}^T(\boldsymbol{J}\bar{\boldsymbol{\Sigma}}\boldsymbol{J}^T + \mathrm{cov}(\boldsymbol{z}_{\mathrm{m}}))^{-1}, \quad (31)$$

where $\mathrm{cov}(\boldsymbol{z}_{\mathrm{m}}) \in \mathbb{R}^{3\times3}$ is the covariance matrix of the scaled visual velocity measurement, and the Jacobian $\boldsymbol{J} \in \mathbb{R}^{3\times4}$ is given by

$$\begin{aligned}\boldsymbol{J} &= \begin{bmatrix} \dfrac{\partial\bar{\boldsymbol{z}}[k]}{\partial\,{}^{\mathcal{C}}\bar{\boldsymbol{v}}[k]} & \dfrac{\partial\bar{\boldsymbol{z}}[k]}{\partial\bar{\rho}[k]} \end{bmatrix} \\ &= \begin{bmatrix} \boldsymbol{I}_3\bar{\rho} & {}^{\mathcal{C}}\bar{\boldsymbol{v}} \end{bmatrix}. \quad (32)\end{aligned}$$

Finally, the predicted state $\bar{\boldsymbol{x}}[k]$ is updated to the estimated state $\hat{\boldsymbol{x}}[k]$, together with uncertainty matrix $\hat{\boldsymbol{\Sigma}}[k]$, as

$$\hat{\boldsymbol{x}} = \begin{bmatrix} {}^{\mathcal{C}}\hat{\boldsymbol{v}} \\ \hat{\rho} \end{bmatrix} = \begin{bmatrix} {}^{\mathcal{C}}\bar{\boldsymbol{v}} \\ \bar{\rho} \end{bmatrix} + \boldsymbol{K}\left(\boldsymbol{z}_m - \bar{\boldsymbol{z}}\right) \quad (33)$$

$$\hat{\boldsymbol{\Sigma}} = (\boldsymbol{I}_4 - \boldsymbol{K}\boldsymbol{J})\bar{\boldsymbol{\Sigma}}. \quad (34)$$

1) Discussion: Several quantities are needed for the implementation of the proposed EKF. Apart from the estimated state $\hat{\boldsymbol{x}}[k]$, one needs knowledge of:

1. the constant IMU/camera rotation matrix ${}^{\mathcal{I}}\boldsymbol{R}_{\mathcal{C}}$ and displacement vector ${}^{\mathcal{I}}\boldsymbol{p}_{\mathcal{IC}}$;
2. the IMU angular velocity ${}^{\mathcal{I}}\boldsymbol{\omega}_m$;
3. the scaled camera linear velocity ${}^{\mathcal{C}}\tilde{\boldsymbol{v}}_m = ({}^{\mathcal{C}}\boldsymbol{v}/d)_m$;
4. the IMU linear acceleration ${}^{\mathcal{I}}\boldsymbol{a} = {}^{\mathcal{I}}\boldsymbol{f}_m + {}^{\mathcal{I}}\boldsymbol{g}$;
5. the plane normal ${}^{\mathcal{C}}\boldsymbol{n}$.

The quantities in item 1 are assumed to be known from a preliminary IMU/camera calibration phase, see, e.g., [28], while vector ${}^{\mathcal{I}}\boldsymbol{\omega}$ in item 2 is available directly from the IMU gyroscope readings. Similarly, vector ${}^{\mathcal{C}}\tilde{\boldsymbol{v}}_m$ in item 3 is retrieved from the optical flow decomposition described in Sec. II.

Measurement of the linear acceleration ${}^{\mathcal{I}}\boldsymbol{a}$ in item 3 requires the specific acceleration ${}^{\mathcal{I}}\boldsymbol{f}_m$ (directly available through the IMU accelerometer readings) and knowledge of the gravity vector ${}^{\mathcal{I}}\boldsymbol{g}$ in IMU frame. An estimation of this latter quantity is also provided by standard IMUs *in near-hovering conditions*, a fact largely exploited when recovering the UAV attitude from onboard sensing, see, e.g., [29]. In our case, we found the employed IMU able to provide a reliable estimation of ${}^{\mathcal{I}}\boldsymbol{g}$ even when undergoing moderate accelerations thanks to its internal filtering scheme. An experimental validation of these observations is provided in Sec. IV-D.

Finally, ${}^{\mathcal{C}}\boldsymbol{n}$ can be directly recovered from the optical flow decomposition as discussed in Sects. II-B–II-C (this step can also be complemented by the use of filtering techniques such as those discussed in [30]), or, in case of a horizontal planar scene as often found in indoor environments, by identifying its direction with that of ${}^{\mathcal{I}}\boldsymbol{g}$. This latter possibility was exploited in all the experiments reported in Sec. IV, which indeed involved a plane normal ${}^{\mathcal{C}}\boldsymbol{n}$ parallel to $\boldsymbol{g}$ (horizontal ground plane).

D. Scale Estimation based on a Nonlinear Estimation Scheme

In this Section, we summarize the derivations of the alternative *nonlinear observer* for retrieving the plane distance d based on a framework originally proposed in [25], [26] for dealing with Structure from Motion (SfM) problems, and recently revisited in [17], [18] in the context of *nonlinear active estimation*. Compared to the previously discussed EKF, this estimation scheme does not require any linearization/approximation step of the system dynamics. This results in an overall cleaner design which, following the analysis reported in [17], also allows for a *full characterization* of the estimation error transient response in case of a camera traveling with a constant linear acceleration norm ($\| {}^{\mathcal{C}}\boldsymbol{\kappa}(t)\| = const$). In particular, we will discuss how one can impose to the estimation error a transient response equivalent to that of *reference linear second-order system with desired poles* by suitably acting on the estimation gains and on the UAV acceleration. This possibility enables the designer, for instance, to choose the needed combination of estimation gains/UAV motion yielding a desired estimation performance, as well as to predict in advance the convergence time of the filter in terms of percentages of the initial error[3]. However, we also note that, as opposed to the EKF, the filter design assumes deterministic dynamics. Therefore, it does not explicitly account for the noise present in the system concerning state prediction and sensor measurements. This point will be thoroughly addressed in the next Sec. IV.

For the sake of exposition, the following developments are here formulated in continuous time. We first recall a classical result of the adaptive control literature, also known as the Persistency of Excitation Lemma [31], and upon which the next developments are based.

Lemma 1 (Persistency of Excitation): Consider the system

$$\left\{ \begin{array}{rcl} \dot{\boldsymbol{\xi}} & = & -\boldsymbol{D}\boldsymbol{\xi} + \boldsymbol{\Omega}^T(t)\boldsymbol{z}, \\ \dot{\boldsymbol{z}} & = & -\boldsymbol{\Lambda}\boldsymbol{\Omega}(t)\boldsymbol{P}\boldsymbol{\xi}, \end{array} \right. \tag{35}$$

where $\boldsymbol{\xi} \in \mathbb{R}^m$, $\boldsymbol{z} \in \mathbb{R}^p$, $\boldsymbol{D} > 0$, $\boldsymbol{P} = \boldsymbol{P}^T > 0$ such that

$$\boldsymbol{D}^T\boldsymbol{P} + \boldsymbol{P}\boldsymbol{D} = \boldsymbol{Q}, \qquad \text{with } \boldsymbol{Q} > 0,$$

and $\boldsymbol{\Lambda} = \boldsymbol{\Lambda}^T > 0$. If $\|\boldsymbol{\Omega}(t)\|$, $\|\dot{\boldsymbol{\Omega}}(t)\|$ are uniformly bounded and the *persistency of excitation* condition is satisfied, i.e., there exist a $T > 0$ and $\gamma > 0$ such that

$$\int_t^{t+T} \boldsymbol{\Omega}(\tau)\boldsymbol{\Omega}^T(\tau)\mathrm{d}\tau \geq \gamma \boldsymbol{I} > 0, \qquad \forall t \geq t_0, \tag{36}$$

then $(\boldsymbol{\xi}, \boldsymbol{z}) = (\boldsymbol{0}, \boldsymbol{0})$ is a globally exponentially stable equilibrium point.

This result can be exploited as follows: assume a vector $\boldsymbol{x} = [\boldsymbol{x}_1^T \ \boldsymbol{x}_2^T]^T \in \mathbb{R}^{m+p}$ can be split into a *measurable* component $\boldsymbol{x}_1$ and an *unmeasurable* component $\boldsymbol{x}_2$. Defining an estimation vector $\hat{\boldsymbol{x}} = [\hat{\boldsymbol{x}}_1^T \ \hat{\boldsymbol{x}}_2^T]^T \in \mathbb{R}^{m+p}$, and the corresponding estimation error

$$\boldsymbol{e} = \left[\begin{array}{c} \boldsymbol{\xi} \\ \boldsymbol{z} \end{array} \right] = \left[\begin{array}{c} \boldsymbol{x}_1 - \hat{\boldsymbol{x}}_1 \\ \boldsymbol{x}_2 - \hat{\boldsymbol{x}}_2 \end{array} \right],$$

[3]Therefore, given an upper bounded initial error $d(t_0) - \hat{d}(t_0)$, one can, for instance, plan in advance the duration of the UAV motion so as to yield a guaranteed accuracy in the estimated plane distance.

the goal is to design an update rule for $\hat{\boldsymbol{x}}$ such that the closed-loop error dynamics matches formulation (35). When this manipulation is possible, Lemma 1 ensures global exponential convergence of the estimation error $\boldsymbol{e} = [\boldsymbol{\xi}^T \ \boldsymbol{z}^T]^T$ to $\boldsymbol{0}$, thus allowing to infer the unmeasurable value of $\boldsymbol{x}_2$ from knowledge of $\boldsymbol{x}_1$. The PE condition (36) plays the role of an observability constraint: estimation of $\boldsymbol{x}_2$ is possible iff matrix $\boldsymbol{\Omega}(t) \in \mathbb{R}^{p\times m}$ is *sufficiently exciting* over time in the sense of (36). We finally note that, being $\boldsymbol{\Omega}(t)$ a generic (but known) *time-varying* quantity, formulation (35) is not restricted to only span the class of linear systems, but it can easily accommodate nonlinear terms as long as they are embedded in matrix $\boldsymbol{\Omega}(t)$.

We now detail how to tailor (35) to the case under consideration. We start by defining (similarly to what done in Sec. III-C)

$$x_2 = \rho = \frac{1}{d} \tag{37}$$

and

$$\boldsymbol{x}_1 = {}^{\mathcal{C}}\tilde{\boldsymbol{v}} = \frac{{}^{\mathcal{C}}\boldsymbol{v}}{d} = {}^{\mathcal{C}}\boldsymbol{v}x_2 \tag{38}$$

with, therefore, $m = 3$ and $p = 1$. Exploiting (22), the dynamics of x_2 is given by

$$\dot{x}_2 = -x_2^2 \, {}^{\mathcal{C}}\boldsymbol{v}^T \, {}^{\mathcal{C}}\boldsymbol{n} = -x_2 \boldsymbol{x}_1^T \, {}^{\mathcal{C}}\boldsymbol{n}. \tag{39}$$

As for the dynamics of $\boldsymbol{x}_1$, using (13) we have

$$\begin{aligned} \dot{\boldsymbol{x}}_1 &= {}^{\mathcal{C}}\dot{\boldsymbol{v}}x_2 + {}^{\mathcal{C}}\boldsymbol{v}\dot{x}_2 = {}^{\mathcal{C}}\dot{\boldsymbol{v}}x_2 - {}^{\mathcal{C}}\boldsymbol{v}x_2\boldsymbol{x}_1^T \, {}^{\mathcal{C}}\boldsymbol{n} \\ &= {}^{\mathcal{C}}\dot{\boldsymbol{v}}x_2 - \boldsymbol{x}_1\boldsymbol{x}_1^T \, {}^{\mathcal{C}}\boldsymbol{n} = {}^{\mathcal{C}}\boldsymbol{\kappa}x_2 - [{}^{\mathcal{C}}\boldsymbol{\omega}]_\times \boldsymbol{x}_1 - \boldsymbol{x}_1\boldsymbol{x}_1^T \, {}^{\mathcal{C}}\boldsymbol{n}. \end{aligned} \tag{40}$$

We then define

$$\begin{aligned} \boldsymbol{\Omega}^T(t) = {}^{\mathcal{C}}\boldsymbol{\kappa} &= {}^{\mathcal{C}}\boldsymbol{R}_{\mathcal{I}} \, {}^{\mathcal{I}}\boldsymbol{a} + [{}^{\mathcal{C}}\boldsymbol{\omega}]_\times^2 \, {}^{\mathcal{C}}\boldsymbol{p}_{\mathcal{IC}} + [{}^{\mathcal{C}}\dot{\boldsymbol{\omega}}]_\times \, {}^{\mathcal{C}}\boldsymbol{p}_{\mathcal{IC}} \\ &\approx {}^{\mathcal{C}}\boldsymbol{R}_{\mathcal{I}} \, {}^{\mathcal{I}}\boldsymbol{a} + [{}^{\mathcal{C}}\boldsymbol{\omega}]_\times^2 \, {}^{\mathcal{C}}\boldsymbol{p}_{\mathcal{IC}} \end{aligned} \tag{41}$$

where, analogously to the EKF case, ${}^{\mathcal{I}}\dot{\boldsymbol{\omega}}$ is neglected since no direct measurement of the UAV angular acceleration is available onboard. With these settings, the update rule for the estimated state $\hat{\boldsymbol{x}}$ can be designed as

$$\left\{ \begin{array}{rcl} \dot{\hat{\boldsymbol{x}}}_1 & = & \boldsymbol{\Omega}^T(t)\hat{\boldsymbol{x}}_2 - [{}^{\mathcal{C}}\boldsymbol{\omega}]_\times \boldsymbol{x}_1 - \boldsymbol{x}_1\boldsymbol{x}_1^T \, {}^{\mathcal{C}}\boldsymbol{n} + \boldsymbol{D}\boldsymbol{\xi} \\ \dot{\hat{x}}_2 & = & -\hat{x}_2\boldsymbol{x}_1^T \, {}^{\mathcal{C}}\boldsymbol{n} + \boldsymbol{\Lambda}\boldsymbol{\Omega}(t)\boldsymbol{\xi} \end{array} \right. \tag{42}$$

with $\boldsymbol{D} > 0$ and $\boldsymbol{\Lambda} > 0$ being symmetric and positive definite gain matrixes. Note that (42) involves only measured and estimated quantities, including a feedback action on $\boldsymbol{\xi} = \boldsymbol{x}_1 - \hat{\boldsymbol{x}}_1$, i.e., the measurable component of the error vector. With this choice, the dynamics of the estimation error $\boldsymbol{e} = [\boldsymbol{\xi}^T \ \boldsymbol{z}^T]^T$ then becomes

$$\left\{ \begin{array}{rcl} \dot{\boldsymbol{\xi}} & = & -\boldsymbol{D}\boldsymbol{\xi} + \boldsymbol{\Omega}^T(t)z \\ \dot{z} & = & -\boldsymbol{\Lambda}\boldsymbol{\Omega}(t)\boldsymbol{\xi} - z\boldsymbol{x}_1^T \, {}^{\mathcal{C}}\boldsymbol{n}. \end{array} \right. \tag{43}$$

It is easy to verify that, by letting $\boldsymbol{P} = \boldsymbol{I}_3$, formulation (35) is almost fully recovered apart from the spurious scalar term

$$g(\boldsymbol{e}, t) = -z\boldsymbol{x}_1^T \, {}^{\mathcal{C}}\boldsymbol{n} \tag{44}$$

in (43). Nevertheless, exponential convergence of the estimation error $\boldsymbol{e}(t)$ to $\boldsymbol{0}$ can still be proven by resorting to Lyapunov theory and by noting that the spurious term $g(\boldsymbol{e}, t)$ is a *vanishing perturbation* of an otherwise globally

exponentially stable nominal system, i.e., $g(\mathbf{0},\, t) = \mathbf{0},\, \forall t$. We refer the reader to [17], [25], [26] for additional discussion and proofs of these facts.

We note that the design of observer (42) did not require any linearization step as for the previous EKF thanks to the more general class of (nonlinear) systems spanned by formulation (35). It is also worth analyzing, in our specific case, the meaning of the PE condition (36) necessary for obtaining a converging estimation. Being $\boldsymbol{\Omega}^T(t) \in \mathbb{R}^3$ a vector, condition (36) requires that $\|\boldsymbol{\Omega}(t)\|$ (i.e., at least one component) does not ultimately vanish over time. On the other hand, vector $\boldsymbol{\Omega}^T(t)$ represents the camera linear acceleration through space w.r.t. the inertial world frame $\mathcal{W}$, see (41). Therefore, we recover the requirement of Sect. III-B stating that estimation of d is possible if and only if the camera undergoes a physical acceleration w.r.t. the observed plane, and, consequently, moving at constant velocity w.r.t. $\mathcal{W}$ cannot allow the estimation to converge.

Finally, as done in the previous Sec. III-C1, we list the quantities necessary for implementing the proposed observer (42). In addition to the estimated state $\hat{\boldsymbol{x}}$, these are:

1 the constant IMU/camera rotation matrix ${}^{\mathcal{I}}\boldsymbol{R}_{\mathcal{C}}$ and displacement vector ${}^{\mathcal{I}}\boldsymbol{p}_{\mathcal{IC}}$;
2 the IMU angular velocity ${}^{\mathcal{I}}\boldsymbol{\omega}_m$;
3 the scaled camera linear velocity $\boldsymbol{x}_1 = {}^{\mathcal{C}}\tilde{\boldsymbol{v}}_m$;
4 the IMU linear acceleration ${}^{\mathcal{I}}\boldsymbol{a} = {}^{\mathcal{I}}\boldsymbol{f}_m + {}^{\mathcal{I}}\boldsymbol{g}$;
5 the plane normal ${}^{\mathcal{C}}\boldsymbol{n}$.

Thus, the same quantities discussed in Sec. III-C1 are also required for the nonlinear estimation scheme (42).

1) Shaping the Estimation Transient Response: We now apply the theoretical analysis developed in [17] to the case at hand, and aimed at characterizing the transient response of the error system (43) in the unperturbed case, i.e., with $g(\boldsymbol{e},\, t) = \mathbf{0}$.

Let $\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T = \boldsymbol{\Omega}$ be the singular value decomposition of matrix $\boldsymbol{\Omega}$ where $\boldsymbol{\Sigma} = [\boldsymbol{S}\ \ \mathbf{0}] \in \mathbb{R}^{p\times m}$, $\boldsymbol{S} = \mathrm{diag}(\sigma_i) \in \mathbb{R}^{p\times p}$, and $0 \leq \sigma_1 \leq \ldots \leq \sigma_p$ are the singular values of $\boldsymbol{\Omega}$. In the case under consideration ($p = 1$, $m = 3$), it is $\boldsymbol{U} = 1$ and $\sigma_1 = \|\boldsymbol{\Omega}\|$. Let also $\boldsymbol{\Lambda} = \alpha\boldsymbol{I}$, $\alpha > 0$ (scalar gain $\boldsymbol{\Lambda}$). By then designing the gain $\boldsymbol{D} \in \mathbb{R}^{3\times 3}$ as

$$\boldsymbol{D} = \boldsymbol{V} \begin{bmatrix} D_1 & \mathbf{0} \\ \mathbf{0} & \boldsymbol{D}_2 \end{bmatrix} \boldsymbol{V}^T \tag{45}$$

with $D_1 \in \mathbb{R} > 0$ and $\boldsymbol{D}_2 \in \mathbb{R}^{2\times 2} > 0$, it is possible to show that, under the change of coordinates

$$\eta = \frac{1}{\sqrt{\alpha}\sigma_1} z \tag{46}$$

and in the approximation $\sigma_1(t) = \|\boldsymbol{\Omega}(t)\| \approx const$, the estimation error $z(t)$ (when expressed in the coordinates $\eta(t)$) obeys the following second-order linear dynamics

$$\ddot{\eta} = -D_1\dot{\eta} - \alpha\sigma_1^2\eta, \tag{47}$$

that is, a (unit-)mass-spring-damper system with stiffness $\alpha\sigma_1^2$ and damping D_1.

The transient response of the estimation error $z(t) = x_2(t) - \hat{x}_2(t) = 1/d(t) - 1/\hat{d}(t)$ can then be imposed by properly 'placing the poles' of system (47). This can be achieved by:

1) taking $D_1 = 2\sqrt{\alpha}\sigma_1$ so as to obtain a critically-damped state evolution for (47) (real and coincident poles). Note that this choice univocally determines D_1 as a function of the system state (the current value of $\sigma_1 = \|\boldsymbol{\Omega}\|$);
2) *actively* enforcing $\alpha\sigma_1^2(t) = \alpha\|\boldsymbol{\Omega}(t)\|^2 = \sigma_d^2 = const$ over time for some desired $\sigma_d^2 \neq 0$, that is, by flying with a given non-zero and constant norm of the camera linear acceleration $\boldsymbol{\Omega}$ scaled by gain α (a free parameter whose role will be better detailed in the following).

From standard linear system theory, these choices then result in the following behavior for the estimation error $z(t) = 1/d(t) - 1/\hat{d}(t)$

$$z(t) = (1 + \sigma_d(t - t_0))e^{-\sigma_d(t-t_0)}z(t_0) \quad \forall t \geq t_0. \tag{48}$$

We conclude noting that, in general, equation (48) represents a *reference evolution* for $z(t)$ since (i) in real-word conditions it may be hard to maintain *exactly* a $\|\boldsymbol{\Omega}(t)\| = const$ over time (condition also needed to render exactly valid the change of coordinates (46)), and (ii) this analysis omits the effect of the (vanishing) disturbance $g(\boldsymbol{e},\, t)$ in (44). Nevertheless, Sec. IV will show an excellent match between the reported results and the reference behavior (48).

Finally, we comment about the role of gain α (the only free parameter of the nonlinear observer): being $\sigma_d = \sqrt{\alpha}\|\boldsymbol{\Omega}\|$, the same convergence rate for $z(t)$ (dictated by the value of σ_d, see (48)) can be equivalently obtained by either accelerating faster or by increasing gain α. While increasing gain α may always appear more convenient in terms of reduced control effort, practical issues such as noise, discretization or quantization errors, may impose an upper limit on the possible value of α, thus necessarily requiring a larger $\|\boldsymbol{\Omega}\|$ (larger acceleration norm) for obtaining the desired convergence speed. More details about this point will be presented in the next Sec. IV.

IV. EXPERIMENTAL EVALUATION

This section reports the results of several experiments meant to illustrate and validate the proposed framework for ego-motion estimation on simulated and real scenarios. After the description of the experimental setup in Sec. IV-A, we replicate the structure of the previous sections by first comparing in Sec. IV-B the three different variants discussed in Sec. II for retrieving the scaled linear and angular velocity from the optical flow decomposition. Then, we proceed to present in Sec. IV-C a series of experiments involving the two scale estimation techniques illustrated in Sec. III. An assessment of the ability of the employed IMU in estimating the gravity direction while the quadrotor undergoes the accelerations needed to recover the unknown scale is also presented in Sec. IV-D. Further considerations about the possibility of predicting (and imposing) an estimation transient behavior for the nonlinear observer of Sec. III-D, and about its robustness against noise, are then discussed in Sec. IV-E and IV-F, respectively. The experimental evaluation is then finally concluded in Sec. IV-G by reporting the results of a closed-loop velocity control on a real quadrotor UAV with all the computations performed onboard.

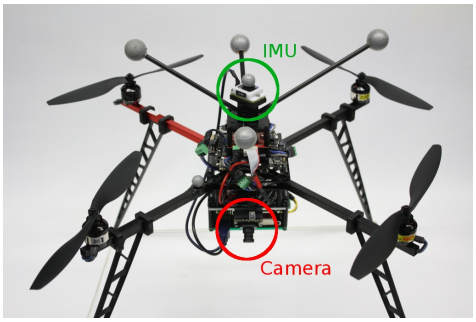


Fig. 2: Experimental setup with the highlighted location of IMU and camera. The x-axis of the body frame is oriented along the red metal beam.

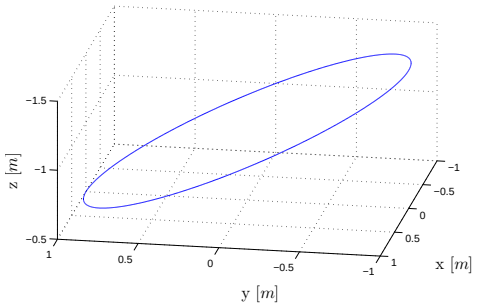


Fig. 3: Circular trajectory used in the experiments for comparing the three algorithms V1, V2, V3 of Sec. II. The trajectory has a diameter of $2\,m$ and lies on a plane at 26.5 deg w.r.t the ground. The UAV completes one revolution every $10\,s$. The height varies between $0.5\,m$ and $1.5\,m$. A similar trajectory, but lying on a horizontal plane, has also been used for the experiments of Sec. IV-C

A. Experimental Setup

For our experiments, we made use of a quadrotor UAV from MikroKopter as experimental platform, and of the free *TeleKyb* software framework [32] as middleware and control software. An external motion tracking system (Vicon[4]) was employed for obtaining the ground truth data and for allowing closed-loop control of the UAV in all the following experiments apart from the results reported in Sec. IV-G where autonomous flight was instead achieved without any external aid.

Figures 1–2 depict the location of all relevant sensors and frames. The quadrotor was equipped with an additional 3DM-GX3-25 IMU from MicroStrain (frame $\mathcal{I}$) to provide the measurements of the specific acceleration ${}^{\mathcal{I}}\boldsymbol{f}_m$, of the angular velocity ${}^{\mathcal{I}}\boldsymbol{\omega}_m$, and of the gravity vector ${}^{\mathcal{I}}\boldsymbol{g}_m$ at 200 Hz. The gravitational vector was estimated internally by the IMU via fusion of the measurements from the accelerometer, rate gyroscopes, temperature sensor and a 3D magnetometer. The visual input was provided by a precalibrated MatrixVision mvBlueFox camera (frame $\mathcal{C}$) mounted downfacing on the UAV, and delivering images of size 752×480 pixel. The onboard processing was carried out on a small PC with an Intel Atom processing unit, while velocity command inputs were transmitted to the vehicle using a wireless serial interface. When processing recorded data offline on a desktop computer, the optical flow could be decomposed at 50 Hz. When, instead, relying on the onboard processing unit (as in the experiments of Sec. IV-G), the acquired images were processed online at a frame rate of 17 Hz. In this case, we also found that, in average, the images were delayed by $100\,ms$ w.r.t. the IMU measurements. In order to compensate for this effect, we delayed the processing of the IMU readings accordingly.

[4]www.vicon.com.

Employed algorithm:	*V1* (II-B)	*V2* (II-C)	*V3* (II-D)
Linear velocity m/s			
Mean error	0.134	0.117	0.113
Standard deviation	0.094	0.093	0.088
Angular velocity rad/s			
Mean error	0.151	IMU	IMU
Standard deviation	0.110	IMU	IMU

TABLE I: Comparison of the errors in retrieving the scaled linear velocity and angular velocity for the three algorithms of Sec. II-B–II-D: the pure visual estimation (*V1*), the inclusion of the readings from the rate gyros (*V2*), and the additional integration of the measured normal vector taken coincident with the gravity direction (*V3*). In the case of algorithms *V2* and *V3*, the angular velocity is measured from the IMU.

In order to allow for an efficient feature tracking, the point features were initially detected using a FAST tracker [33]. An efficient pyramidal implementation of the Lucas-Kanade algorithm [34] was then used to track the features over all consecutive images. For further guaranteeing a minimum number of observed features while avoiding unnecessary and costly feature re-sampling, the point features were only sampled once the set of currently tracked features did drop below a given threshold. New FAST features were then detected and added to the set until a desired maximum set size was reached.

B. Comparison of Algorithms V1–V3 for Ego-Motion Estimation from Optical Flow

We start presenting a comparison of the three possibilities discussed in Sec. II-B–II-D for recovering $\boldsymbol{v}/d$ and $\boldsymbol{\omega}$ during flight. For the reader's convenience, we recall that algorithm *V1* (Sec. II-B) is only based on the perceived optical flow for obtaining $(\boldsymbol{v}/d,\, \boldsymbol{\omega})$, algorithm *V2* (Sec. II-C) exploits the additional (independent) measurement of $\boldsymbol{\omega}$ from the onboard gyros, and algorithm *V3* (Sec. II-D) further exploits the independent knowledge of the scene normal $\boldsymbol{n}$ taken coincident with the gravity vector.

To allow for a controlled and direct comparison of the three possibilities *V1–V3*, we recorded data while flying along a circular trajectory of $2\,m$ in diameter as shown in Fig. 3. The trajectory was chosen in order to have the UAV accelerating sinusoidally along the three Cartesian directions with a period of $10\,s$, and a maximum speed and acceleration of $0.6\,m/s$ and $0.296\,m/s^2$, respectively. The height varied from $0.5\,m$ to $1.5\,m$ along the trajectory. Additionally, the vehicle was periodically rotated around its body z-axis in the range of $[-70 \ldots 70]\,deg$. The quadrotor relied on an external motion tracking system to track this precomputed trajectory. Onboard hardware was used to record vision and IMU data during flight. Afterwards, all three algorithms of Sec. II-B–II-D were run on this common dataset to allow for a direct comparison.

Figure 4 summarizes the results of this first experiment. The plots in Figs. 4a–4c show a superimposition of the three estimations of $\boldsymbol{v}$ against the ground truth provided by

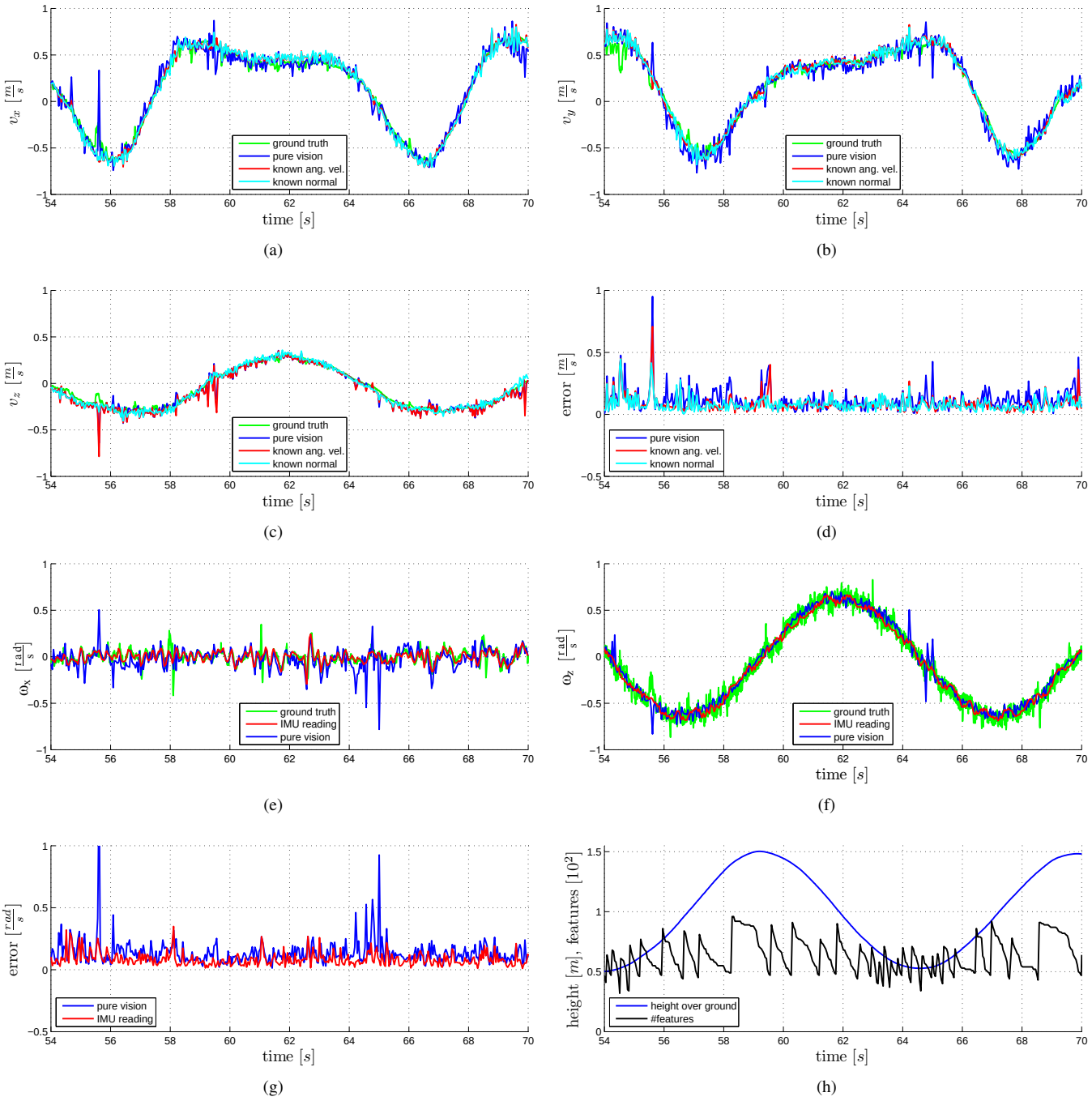


Fig. 4: Experimental results — Comparison against a ground truth of the algorithms *V1* ('pure vision'), *V2* ('known $\boldsymbol{\omega}$'), and *V3* ('known $\boldsymbol{\omega}$ and $\boldsymbol{n}$') for retrieving the scaled linear velocity and angular velocity (Sec. II). For the purpose of visualization only, the estimated non-metric linear velocity has been scaled using the real distance to the plane d obtained from an external optical tracking system.
(a–c) Estimated linear velocity along the x, y and z axes. (d) Norm of the error between ground truth and the estimated linear velocity for the three cases. (e–f) Estimated angular velocity along the x and z axes. (g) Norm of the error between the ground truth and the estimated angular velocities. (h) Altitude of the vehicle and number of tracked features.

the external tracking system. In all three cases, the linear velocity $\boldsymbol{v}$ was recovered from the estimated $\boldsymbol{v}/d$ scaled by the ground truth measurement of d. Figure 4d then reports the corresponding error norms.

From these results we can conclude that all the three algorithms *V1–V3* yield a satisfactory performance. However, one can also note how the integration of the IMU measurements (i.e., algorithms *V2–V3*) reduces the mean error of the linear velocity estimation compared to algorithm *V1*, see Table I. A comparison of the angular velocity $\boldsymbol{\omega}$ as retrieved from the visual system (algorithm *V1*) against the IMU readings and the ground truth of the external tracking system is given in Figs. 4e–4f, with Fig. 4g reporting the corresponding error norm w.r.t. the ground truth. We can note how the IMU provides a less noisy estimate of $\boldsymbol{\omega}$ almost free of spikes, while the camera-based estimation is less accurate during low altitude flights.

Figure 4h indicates the origin of the irregularities in the recovered $\boldsymbol{\omega}$. Since the vehicle is moving with a constant speed in world frame, $\boldsymbol{v}/d$ increases with lower altitudes. Therefore, in this case the point features tend to vanish faster from the field of view and can only be tracked over a shorter amount of time. In particular, at $0.5\,m$ altitude and under the given experimental conditions, motion blur starts to affect the tracking and feature re-sampling considerably. Therefore, the system can only maintain a set of 30 to 65 features instead of the, in this case, intended 50 to 100 with a negative effect on the retrieved $\boldsymbol{\omega}$. On the other hand, the estimated $\boldsymbol{v}/d$ results almost insensitive (in the algorithms V2 and V3) w.r.t. the number of tracked features as can be seen in Figs. 4a–4c: here, the estimated $\boldsymbol{v}/d$ in the V2 and V3 cases tracks very well the ground truth despite the various drops in the number of tracked features (Fig. 4h). Table I finally provides a quantitative summary of the reported results. One can again notice the improvements obtained when fusing the decomposition of the perceived optical flow with measurements from the IMU with, in particular, algorithm *V3* performing slightly better than *V2*.

In view of these considerations, all the following results were obtained by relying on algorithm *V2* for retrieving $\boldsymbol{v}/d$ from the decomposition of the optical flow, with $\boldsymbol{\omega}$ instead provided by the onboard IMU. This choice is motivated by the fact that algorithm *V2* grants a performance comparable with algorithm *V3* but under milder assumptions on the surrounding environment (no need of a horizontal dominant plane and of an accurate sensing of the gravity direction).

C. Comparison of the two Scale Estimation Schemes

In the following, the two scale estimation schemes of Sec. III-C and Sec. III-D are first evaluated in simulation to compare their performance in ideal conditions. Subsequently, the results obtained on recoded sensor data from real flights are presented. We also compare our solution against a recent related work by qualitatively reproducing the experiments presented in [15].

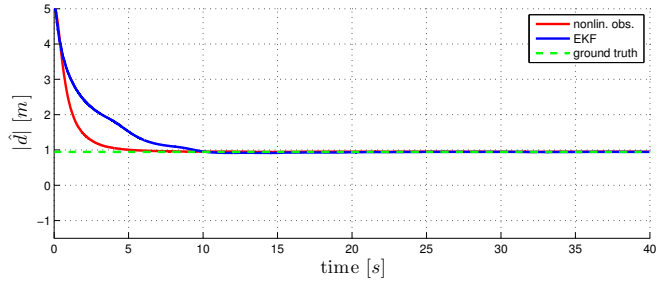
Before starting the discussion, we note that each filter allows for two distinct possibilities for retrieving the metric linear velocity $\boldsymbol{v}$. Indeed, a first possibility is to just set $\hat{\boldsymbol{v}} = (\boldsymbol{v}/d)_m \hat{d}$ exploiting the estimated $\hat{d}$ and the measured $(\boldsymbol{v}/d)_m$ from the optical flow decomposition. However, a second possibility is to exploit the *internal* estimation of $\boldsymbol{v}$ maintained by both filters, that is $\hat{\boldsymbol{v}}$ from (33) in the EKF case, and $\hat{\boldsymbol{v}} = \hat{\boldsymbol{x}}_1/\hat{x}_2$ for the nonlinear observer case (see Sect. III-D). Both possibilities have their pros/cons: in general the second possibility may result in a less noisy but also potentially more delayed estimation of $\boldsymbol{v}$ (because of the 'filtering' action of both estimation schemes). Nevertheless, for the sake of obtaining a less noisy estimate, all the following simulation and experiment results rely on this second possibility.

1) Simulated Data: in order to illustrate the convergence of the proposed estimation schemes, we generated a synthetic acceleration profile together with the corresponding (simulated) sensor readings. This resulted in a camera motion similar to the horizontal circular trajectory used for the experimental results of the next Sec. IV-C2, with an associated constant acceleration norm $\|\boldsymbol{\Omega}\| \approx 0.296\,m/s^2$. All generated sensor readings were perturbed with an additive zero-mean Gaussian noise with covariance matrices taken from the real sensor characteristics: $0.00003\,\boldsymbol{I}_3\,m/s^2$, $0.00002\,\boldsymbol{I}_3\,rad/s$ and $0.00005\,\boldsymbol{I}_3\,1/s$ for ${}^{\mathcal{I}}\boldsymbol{f}_m$, ${}^{\mathcal{I}}\boldsymbol{\omega}_m$, and for the non-metric linear velocity from optical flow $({}^{\mathcal{C}}\boldsymbol{v}/d)_m$, respectively. The same covariance matrices were employed in the steps (27)–(31) of the EKF. As for the nonlinear observer, its only free parameter (gain α) was taken as $\alpha = 12$: this value was chosen so as to obtain, for both filters, the same noise level (that is, variance of the estimation error) in the estimated $\hat{d}(t)$ after convergence. This was meant to ensure a 'fair' comparison among the EKF and the nonlinear observer in terms of estimation accuracy. The choice of $\alpha = 12$ then resulted in a value of $\sigma_d = \sqrt{\alpha}\|\boldsymbol{\Omega}\| \approx 1.025$ for the ideal estimation error convergence in (48).

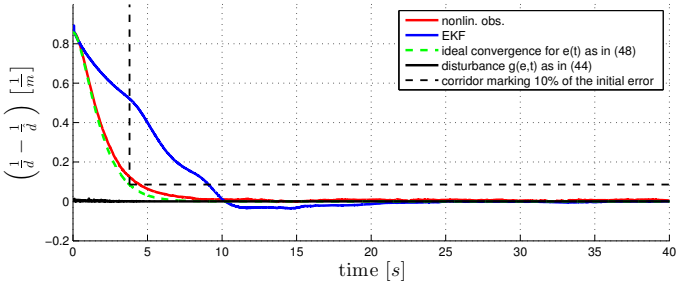
In order to test the robustness against an unknown scene depth, the initial estimation of the plane distance was chosen as $\hat{d}(t_0) = 5\,m$ for both filters against a real distance of $d(t_0) = 1\,m$. Furthermore, in the EKF case the initial ${}^{\mathcal{C}}\hat{\boldsymbol{v}}(t_0)$ was taken as ${}^{\mathcal{C}}\hat{\boldsymbol{v}}(t_0) = \tilde{\boldsymbol{v}}_m(t_0)\hat{d}(t_0)$ (i.e., exploiting the first measurement of $\tilde{\boldsymbol{v}}_m = ({}^{\mathcal{C}}\boldsymbol{v}/d)_m$ scaled by the initial estimation $\hat{d}(t_0)$). In the nonlinear observer case, we instead directly set $\hat{\boldsymbol{x}}_1(t_0) = \tilde{\boldsymbol{v}}_m(t_0)$ (see (38)). Finally, the EKF was initialized with $\hat{\boldsymbol{\Sigma}}(t_0) = \boldsymbol{I}_4$ for representing the initial level of uncertainty, although we will later show that the EKF performance is basically insensitive to the initial choice of $\hat{\boldsymbol{\Sigma}}(t_0)$.

Figure 5a reports a comparison of the estimation performance of both the EKF and the nonlinear observer run in parallel on the same UAV trajectory and sensor readings, together with the ground truth for the plane distance d. One can then appreciate the faster convergence of the estimation error for the nonlinear observer case w.r.t. the 'fully-informed' EKF (in the sense of being aware of the exact noise characteristics). This better performance can be ascribed to the lack of linearization of the system dynamics in the nonlinear observer case.

Furthermore, Fig. 5b shows the behavior of the estimation error $z(t) = 1/d(t) - 1/\hat{d}(t)$ for both filters, with superimposed the 'ideal' transient response (48). We can then note the very good match of this latter ideal response with the



(a)



(b)

Fig. 5: Simulated results — Estimated plane distance d using a simulated dataset. (a) Estimated distance $d(t)$ and (b) estimation error $z(t) = (1/d(t) - 1/\hat{d}(t))$ for the nonlinear observer and the EKF filter. In (b) the ideal convergence of the nonlinear observer as per (48) is superimposed together with the influence of the neglected disturbace $g(\boldsymbol{e}, t)$ given in (44). The vertical dashed line in (b) denotes the predicted time at which the estimation error of the nonlinear observer should have dropped below the threshold of 10% of the initial error (highlighted with a horizontal dashed line).

behavior of the estimation error $z(t)$ for the nonlinear observer case, thus confirming the theoretical analysis of Sec. III-D. The plot also shows a horizontal band at $10\,\%$ of the initial error: according to the ideal reponse (48), the estimation error of the nonlinear observer should have dropped below $10\,\%$ of the initial error $z(t_0)$ after $3.79\,s$. The result in Fig. 5b is slightly off this prediction because of the presence of the perturbation term $g(z, t)$ in (44) (also shown in the plot). Indeed, presence of this term (neglected in the analysis) initially slows down the convergence rate of $\hat{d}(t)$ w.r.t. its predicted behavior. Note that, ideally, one should have had $g(z, t) = 0$ in this case, since $\boldsymbol{v}$ was always perpendicular to the plane normal $\boldsymbol{n}$ (see again (44)). However, the simulated noisy measurements resulted in the presence of a small $\boldsymbol{v}_z$ which gave rise to the initial non-null value of $g(z, t)$. Additional details about the possibility of predicting the convergence time of the nonlinear observer are also given in Sec. IV-E.

The linear velocities estimated by both observers are shown in comparison in Fig. 6. As with the estimation of the scale factor d, the estimation for the metric velocity $\boldsymbol{v}$ converges faster in the nonlinear observer case. However, both scale estimation schemes provide very reliable velocity measurements after $20\,\mathrm{s}$ as shown in Fig. 7 where the norm of the velocity estimation errors are reported. Finally, Fig. 8 shows the behavior of the distance $\hat{d}(t)$ estimated by the EKF when initializing the state covariance $\hat{\boldsymbol{\Sigma}}(t_0)$ with values between $\hat{\boldsymbol{\Sigma}}(t_0) = 10^{-6}\boldsymbol{I}_4$ and $\hat{\boldsymbol{\Sigma}}(t_0) = 10^{9}\boldsymbol{I}_4$ against the estimation

	EKF	nonlin. obs.
Simulated data		
RMS error after convergence (scale)	$0.0075\,\mathrm{m}$	$0.0078\,\mathrm{m}$
RMS error after convergence (velocity)	$0.0071\,\mathrm{m/s}$	$0.0111\,\mathrm{m/s}$
Recorded data		
RMS error after convergence (scale)	$0.0923\,\mathrm{m}$	$0.0988\,\mathrm{m}$
RMS error after convergence (velocity)	$0.1160\,\mathrm{m/s}$	$0.0945\,\mathrm{m/s}$
Comparison againist [15]		
RMS error x-axis (velocity)	$0.0101\,\mathrm{m/s}$	$0.0074\,\mathrm{m/s}$
RMS error y-axis (velocity)	$0.0141\,\mathrm{m/s}$	$0.0095\,\mathrm{m/s}$
RMS error z-axis (velocity)	$0.0107\,\mathrm{m/s}$	$0.0114\,\mathrm{m/s}$
Improvement over [15] (avg)	$250\,\%$	$323\,\%$

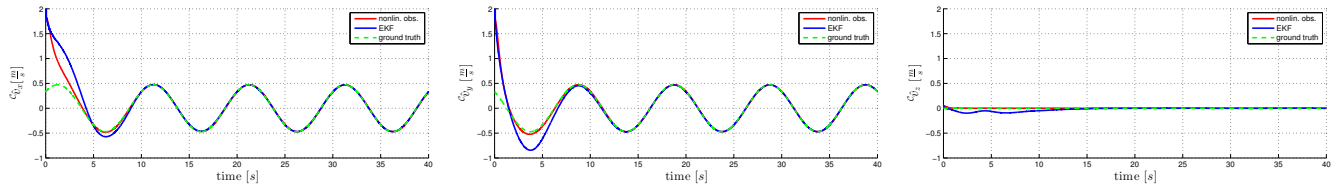
TABLE II: Comparison of the two scale estimation approaches presented in this work. The results are discussed in Sec. IV-C.

provided by the nonlinear observer. One can then verify how the EKF convergence is always slower than in the nonlinear observer case, and how any 'reasonable' initial $\hat{\boldsymbol{\Sigma}}(t_0)$ has only a small effect on the EKF performance.

Table II finally summarizes the quantitative results for the comparison of both scale estimation approaches. The reported Root Mean Square (RMS) error for both systems was computed after 20 sec. of flight, i.e., after both estimates have converged.

2) Recorded Data: the two scale estimation approaches were also compared by processing a dataset collected during a real quadrotor flight. The chosen trajectory was similar to the circular one used in Sec. IV-B but rotated on a horizontal plane in order to yield a constant height during motion. The chosen trajectory was again characterized by an acceleration of $\|\boldsymbol{\Omega}\| \approx 0.296\,m/s^2$, thus resulting in a $\sigma_d = \sqrt{\alpha}\|\boldsymbol{\Omega}\| \approx 1.025\,m/s^2$ for the ideal system (48). However, on the real quadrotor the actual accelerations were found slightly higher than those expected from the ideal circular trajectory due to additional constant regulations necessary to overcome small real-world disturbances. All sensor offsets were calibrated before the recording of the dataset, and the covariance matrices of ${}^{\mathcal{I}}\boldsymbol{f}_m$, ${}^{\mathcal{I}}\boldsymbol{\omega}_m$ and $({}^{\mathcal{C}}\boldsymbol{v}/d)_m$ were estimated over a period of $60\,s$. Both filters were initialized as described in the previous Sec. IV-C1, thus with an initial $\hat{d}(t_0) = 5\,m$ against a real height of $d(t_0) \approx 1\,m$. The gain for the nonlinear observer was again selected as $\alpha = 12$ to ensure a similar noise level for the scale estimation from the EKF and the nonlinear observer after convergence.

Figure 9a shows the behavior of the estimated plane distance $\hat{d}(t)$ for both filters, with superimposed the ground truth $d(t)$. The behavior of the estimation error $z(t) = 1/d(t) - 1/\hat{d}(t)$ is shown in Fig. 9b, and the behavior of $d(t) - \hat{d}(t)$ is also reported in Fig. 9c for a better appreciatoin of the convergence behavior. Again, from these plots one can note the good match among the estimation error of the nonlinear observer against the ideal response (48). On this dataset from real sensor measurements, the mismatch between the predicted and actual convergence rate of the nonlinear estimator was smaller than in the previous case of simulated data. This can be explained by the fact that the real acceleration of the vehicle was slightly higher than the acceleration expected from the commanded circular trajectory, and this in turn compensated for the initial



(a) (b) (c)

Fig. 6: Simulated results — Estimated linear velocity from simulated data in the (a) x, (b) y and (c) z components. The estimation provided by the nonlinear observer converges faster, but both state estimators provide reliable results after convergence is reached.

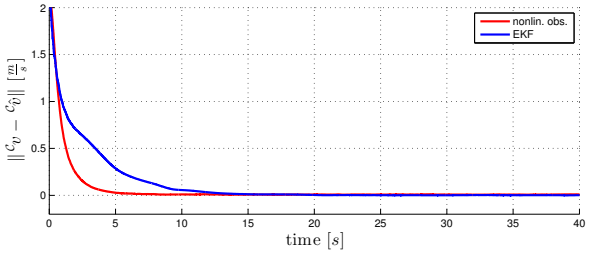


Fig. 7: Simulated results — Norm of the error for the linear velocity estimation for the nonlinear observer and EKF w.r.t. the ground truth.

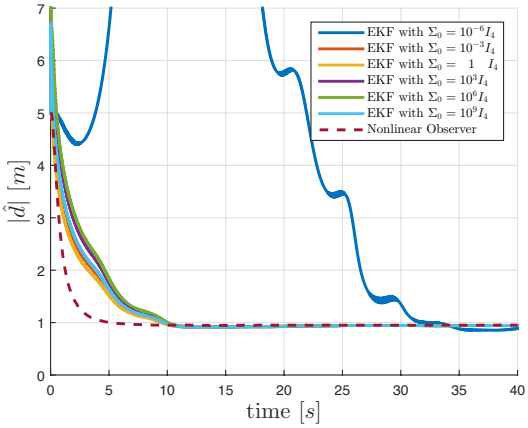


Fig. 8: Simulated results — Influence of the initial choice of $\hat{\mathbf{\Sigma}}(t_0)$ on the convergence rate of the plane distance $d(t)$ estimated by the EKF. Note how the EKF convergence is always slower than in the nonlinear observer case (dashed line) despite the wide range of initial $\hat{\mathbf{\Sigma}}(t_0)$. Furthermore, the EKF convergence results are almost insensitive to any 'reasonable' initial choice of $\hat{\mathbf{\Sigma}}(t_0)$ (this plot should also be compared against Fig. 5a).

'disturbing' effects of the perturbation g (thanks to the higher level of excitation of the actual trajectory).

The estimated metric linear velocities are shown in Fig. 10. The error against the ground truth (depicted in Fig. 11) shows a fast convergence rate for the nonlinear observer. Nevertheless, the EKF yielded a reliable output after approximately 20 s as well. All quantitative results can be compared in Tab. II.

As an additional validation of our approach, we also tested both algorithms on an inclined circular trajectory similar to the one depicted in Fig. 3 in order to deal with the case of a *time-varying* plane distance $d(t)$. The plots shown in Figs. 12–13 again fully confirm that both the scene depth and the vertical component of the linear velocity can be reliably estimated despite the more challenging UAV motion (time-varying $d(t)$).

Motivated by these good results in comparison with the EKF, all the following experiments only involve the nonlinear observer scheme.

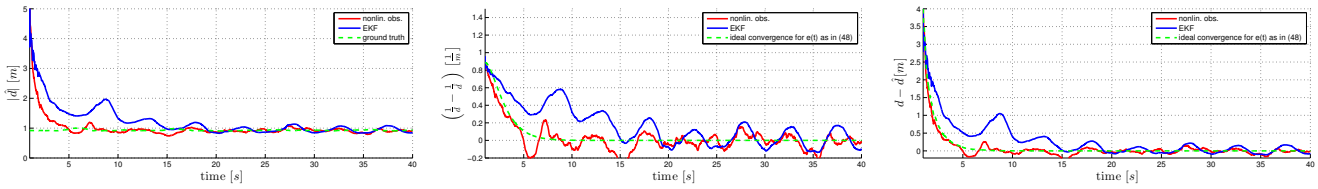
3) Comparison to previous work: for a direct comparison with a state-of-the-art scale estimation approach, we applied the nonlinear observer scheme under experimental conditions similar to those used in [15]. Therefore, the camera-IMU system was moved along a trajectory consisting of small sinusoidal hand-held motions. This resulted in a vehicle trajectory with an amplitude of about $0.1\,m/s$ and a frequency of approximately $1\,\mathrm{Hz}$ at a height of $0.5\,m$. All three directions were tested individually while motion along the other axes was avoided.

Figure 14 shows the estimated metric velocity against ground truth for the nonlinear observer. On this trajectory, we found an RMS value of $[0.0074, 0.0095, 0.0114]\,m/s$ for the three Cartesian directions, respectively. This allows us to compare our results to the figures of $[0.028, 0.035, 0.025]\,m/s$ reported in [15]. Therefore, using the nonlinear observer, we could obtain an average improvement of $320\,\%$ compared to the results reported in [15]. All results are summarized in Tab. II. Although the experimental conditions are obviously different since the original dataset is not available, we believe these results do indicate the effectiveness of the proposed nonlinear observer in dealing with visual-inertial scale and ego-motion estimation vs. other consolidated approaches.

D. Estimation of the Gravity Vector

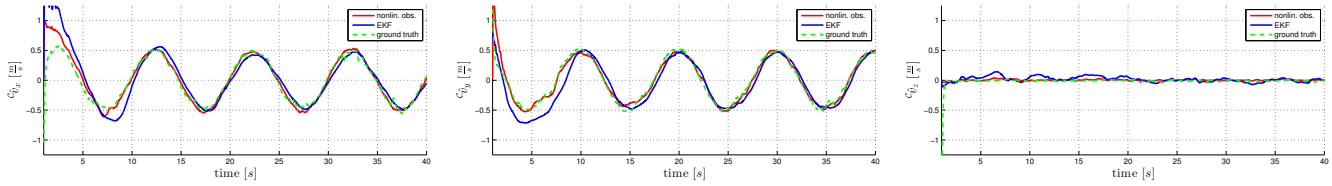
Both scale estimation schemes require knowledge of the direction of the gravity vector $\boldsymbol{g}$, which, as explained in Sec. III-C1, is assumed provided by the IMU itself. In order to verify the accuracy of this gravity estimation, we compared the IMU estimation of $\boldsymbol{g}$ against the ground truth obtained from the motion tracking system during the circular flights conducted for the experiments of Sec. IV-C2.

The comparison of the x and y components of the normalized gravity vector $\boldsymbol{g}/\|\boldsymbol{g}\|$ are shown in Fig. 15. In average, the error of the gravity direction estimated by the IMU vs. the ground truth is smaller than 1 deg. This error is smaller than the accuracy with which we were able to define the object frame of the quadrotor in the motion tracking system. Therefore, we can consider the estimated $\boldsymbol{g}$ from the onboard IMU suitable for our needs.



(a) (b) (c)

Fig. 9: Experimental results — (a) estimation of $d(t)$, (b) estimation error $1/d(t) - 1/\hat{d}(t)$ and (c) estimation error $d(t) - \hat{d}(t)$ against the ground truth exploiting a recoded dataset. The initial $\hat{d}(t_0)$ is set to 5 m against a real distance of approximately $d(t_0) = 1$ m. The ideal transient response of the nonlinear observer as per (48) is also shown in (b–c) with a dashed line. Note, again, the very good match between the predicted and actual behavior of the estimation error for the nonlinear observer case, as well as the faster overall convergence w.r.t. the EKF scheme



(a) (b) (c)

Fig. 10: Experimental results — Estimated linear velocity from a recorded dataset in the (a) x, (b) y, and (c) z components. The ground truth was obtained by numerically differentiating the position information from the motion capture system

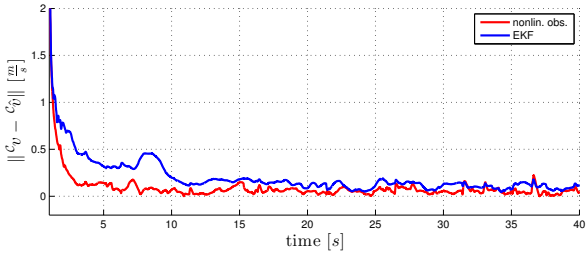


Fig. 11: Experimental results — Norm of the velocity estimation error against the ground truth for the nonlinear filter and EKF scheme. Note, again, the faster convergence rate of the nonlinear observer w.r.t. the EKF

E. Predicting the Error Convergence Time with the Nonlinear Observer

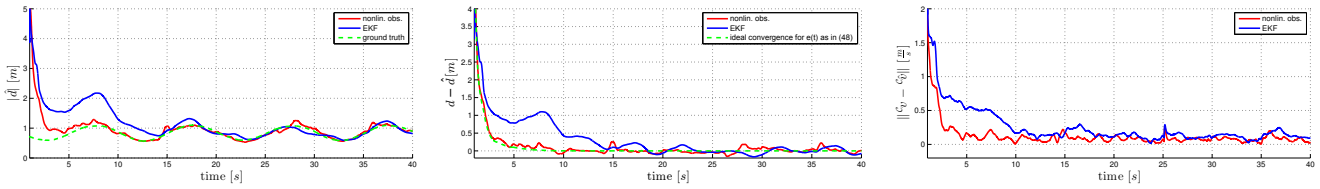
As shown in Figs. 5b–9c and discussed in Sec. IV-B–IV-C, keeping a constant acceleration norm $\|\boldsymbol{\Omega}(t)\|$ allows to impose to the estimation error $z(t) = 1/d(t) - 1/\hat{d}(t)$ of the nonlinear observer a behavior equivalent to the ideal response (48). This possibility allows for a more general observation. Given a desired time for reaching some percentage of the initial error $z(t_0)$, and choosing a gain α, one can exploit (48) to determine the needed *acceleration norm* to achieve this goal. Similarly, for a given trajectory with a known acceleration norm, and a given gain α, one can determine the time needed to reach a desired percentage of the initial estimation error. Such a relation between acceleration norm and convergence time is plotted in Fig. 16 for the three cases of reaching $10\,\%$, $1\,\%$, and $0.1\,\%$ of the initial error under the (arbitrary) choice $\alpha = 12$. Thus, for a bounded initial estimation error, i.e., with the maximum distance to the planar scene known before launch, one can univocally predict how long to fly (with constant acceleration norm) for having a guaranteed accuracy in the estimated distance. We note that this analysis can, of course, be done for any choice of gain α.

F. Noise Robustness of the Nonlinear Observer

As opposed to the fully informed EKF, the nonlinear observer handles presence of noise in the sensor readings in an indirect way. In particular, the single gain α regulates the 'aggressiveness' of the filter, influencing both the convergence speed and its sensitivity to noise. As explained, in all the previous experiments we tuned α so as to obtain the same noise level (variance of the estimation error after convergence) in the nonlinear observer and EKF cases.

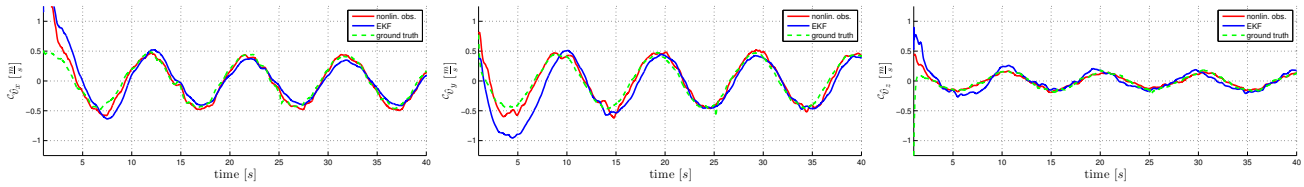
To test the robustness of the nonlinear observer to increased levels of sensor noise for the same choice of α, we increased the simulated noise by a factor of 50 for both the IMU readings and the scaled velocity $\boldsymbol{v}/d$ from the optical flow decomposition. Figure 17a shows the influence of different sensor noise levels on the estimation error $d(t) - \hat{d}(t)$. Despite the high noise level, the filter demonstrates a good level of robustness in generating a consistent state estimation.

Furthermore, from the theoretical analysis of Sec. III-D, the convergence rate of the nonlinear observer in (48) is determined by the quantity $\sigma_d = \|\boldsymbol{\Omega}\|\sqrt{\alpha}$. Thus, one can always trade smaller accelerations with a higher gain α for obtaining the same (ideal) convergence rate. In this sense, we compared a circular flight with an acceleration norm of $\|\boldsymbol{\Omega}\| \approx 0.296\, m/s^2$ and a choice of $\alpha = 12$ to two flights with acceleration norms $0.148\, m/s^2$ and $0.037\, m/s^2$, and the gain α chosen so as to yield the same σ_d (i.e., same error convergence rate). The results are reported in Fig. 17b where we can note, as expected, that by keeping the same σ_d one



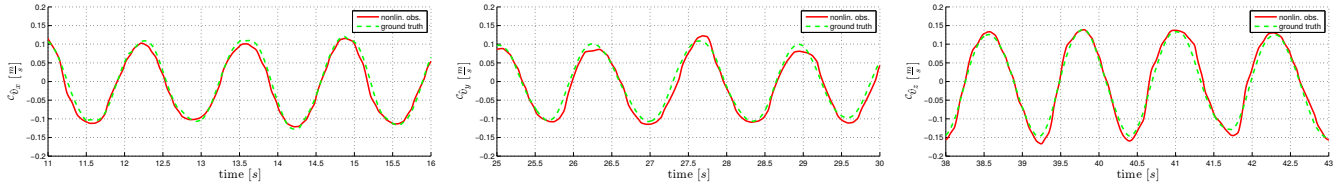
(a) (b) (c)

Fig. 12: Experimental results — Results from flying along an inclined circular trajectory, thus involving a time-varying $d(t)$. (a) Estimated $\hat{d}(t)$ for the nonlinear observer and EKF, (b) Estimation error $d(t) - \hat{d}(t)$ and ideal convergence behavior (48) for the nonlinear observer, and (c) norm of the velocity estimation error against the ground truth. Note, again, the very good performance of the nonlinear observer in recovering $d(t)$ and $\boldsymbol{v}(t)$ despite the more challenging UAV motion (as well as, again, the almost perfect match between the actual and ideal transient response of the estimation error)



(a) (b) (c)

Fig. 13: Experimental results — Results from flying along an inclined circular trajectory, thus involving a time-varying $d(t)$. Estimated linear velocities in the (a) x, (b) y, and (c) z components.



(a) (b) (c)

Fig. 14: Experimental results — Estimated linear velocity applying the nonlinear observer on the experimental conditions similar to those used in [15]. The results for the (a) x, (b) y and (c) z directions are plotted against the ground truth obtained from the tracking system.

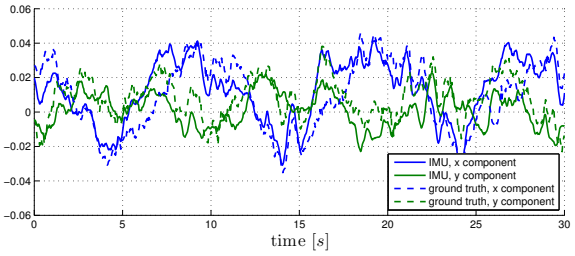


Fig. 15: Experimental results — x and y components of the normalized gravity vector $\boldsymbol{g}/\|\boldsymbol{g}\|$ estimated by the internal sensor fusion algorithm of the employed IMU against the ground truth obtained from the motion capture system. In average, the error between the IMU estimation and the ground truth is in the range of 1 deg despite the accelerated motion undergone by the UAV.

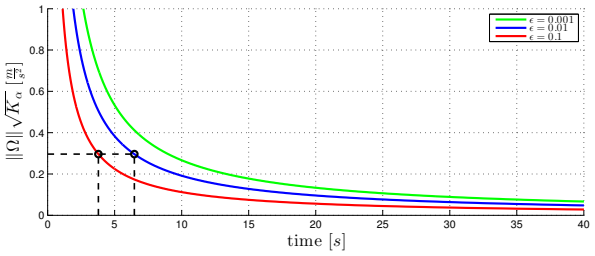
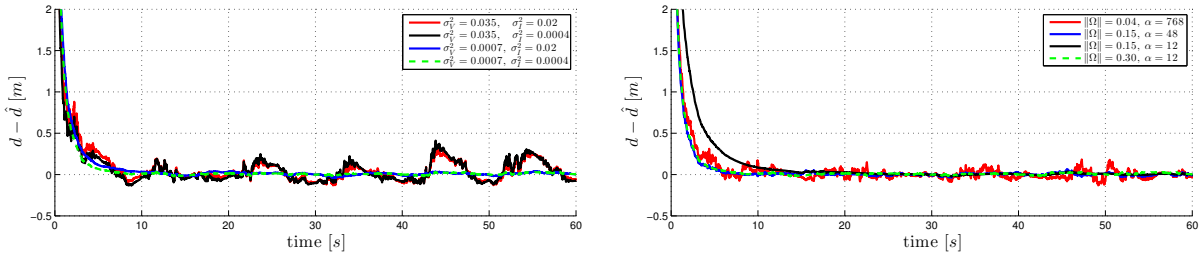


Fig. 16: Relation between the required 'excitement' $\|\boldsymbol{\Omega}\|\sqrt{\alpha}$ and convergence time t needed to reach the same estimation accuracy. The plot was generated for a gain $\alpha = 12$. The three curves show the relation between the time needed to reach a percentage $\epsilon = [10\%, 1\%, 0.1\%]$ of the initial error vs $\|\boldsymbol{\Omega}\|\sqrt{\alpha}$. The plot can be read as follows: on a trajectory with an acceleration norm of $0.296\, m/s^2$, the error will drop below 10% of its initial value within 3.79 s and below 1% within 6.47 s.

obtains the same transient behavior for the estimation error $z(t)$, although a higher noise level is clearly induced by the larger employed α.

G. Closed-Loop Control of a Real Quadrotor UAV

As a final validation of the overall framework, we made use of the estimated metric velocity to 'close the loop' and control the UAV motion in real time. For this experiment, we combined algorithm *V2* for the optical flow decomposition



(a) (b)

Fig. 17: Experimental results — Influence of sensor noise on the nonlinear observer scheme. (a) Robustness of the nonlinear observer to different noise levels. The variance values of $\sigma_V^2 = 0.0007$ and $\sigma_I^2 = 0.0004$ for the visual and inertial sensor data, respectively, correspond to the actual noise found on the real sensors. The noise was then increased 50 times for a comparison. (b) plot showing the trade-off between 'control effort' (norm of the acceleration $\|\boldsymbol{\Omega}\|$) and gain α. In case of lower accelerations, the same convergence rate can be re-established using a higher gain α but at the expense of an increased noise level as clear from the plot. Gain α can then be used to tune the 'aggressiveness' of the filter vs. a given amount of control effort $\|\boldsymbol{\Omega}\|$.

with the scale estimation obtained from the nonlinear observer. The vehicle was commanded using a gamepad to send velocity commands via a wireless link.

Figure 18 shows a section of a longer flight relying purely on onboard sensors. To test the robustness of the system, we moved an object through the field of view of the down-facing camera at time $t = 125\,s$, causing some disturbances in the estimated velocity. Starting from time $t = 134\,s$, the vehicle was additionally commanded to move in the vertical direction, causing the height, and therefore the distance to the plane, to change. The plots show some temporary over- and under-estimations of the linear velocity due to the abrupt commanded motions, but otherwise the proposed approach is always able to adapt to the changing height and recover a reliable estimation.

A video from one of these experiments (Extension 1) is also attached to the paper. Both external views of the vehicle and the image stream from the down-facing camera are shown while the vehicle navigates purely based on onboard sensors and processing capabilities.

We then believe these experiments contribute to demonstrate that the presented velocity-based state estimation pipeline can be used for the closed-loop control of a real quadrotor UAV using solely onboard sensors.

V. CONCLUSIONS AND FUTURE WORK

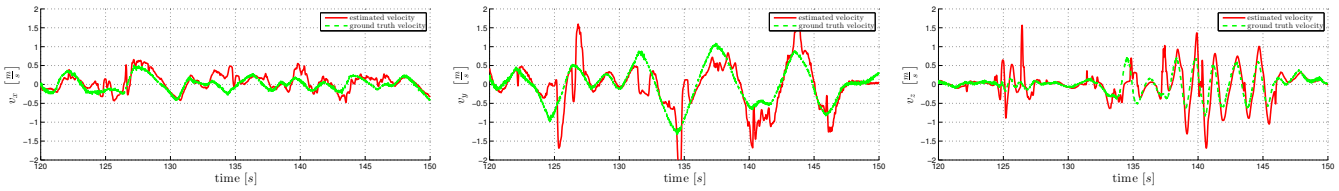
In this paper, we addressed the need of a reliable onboard ego-motion estimation for quadrotor UAVs to overcome the boundaries of controlled indoor environments (such as when relying on external motion tracking systems for recovering the UAV state). To this end, we first discussed three variants of an algorithm based on the continuous homography constraint to obtain an estimation of the UAV scaled linear velocity and angular velocity from the decomposition of the perceived optical flow. This step, indeed, allows to retrieve ego-motion information independently of maps, known landmarks, and the need for tracking features over an extended period of time. Subsequently, we extensively discussed the issue of estimating the (unknown) metric scale factor by fusing the scaled velocity retrieved from optical flow with the high frequency accelerometer readings of an onboard IMU. Scale estimation was achieved by proposing two estimation schemes: a first one based on a classical EKF and a second one on a novel nonlinear observation framework. Simulated and real experimental data were presented to assess and compare the performance of both filters in ideal and real conditions. When compared to the EKF, the nonlinear observer demonstrated a consistent better performance in terms of convergence rate of the scale estimation error. Furthermore, the proposed theoretical analysis showed the possibility to actively impose (and thus predict) the error transient response of the nonlinear observer by suitably acting on the estimation gains and UAV motion (the norm of its acceleration). This analysis was, again, fully confirmed by the reported simulative/experimental results in several conditions, also involving different levels of sensor noise for testing the robustness of the approach.

With the advantage of a fast and predictable convergence for recovering the metric UAV linear velocity, the nonlinear observer proved to be a suitable choice for then implementing a fully onboard control of flying vehicles. In particular, we successfully demonstrated the reliability of the proposed framework in closed-loop experiments on a quadrotor UAV.

A. Future Work

Despite the convincing results, we are considering to extend our work in several ways. A first possibility is to further exploit the awareness of the error convergence behavior when using the nonlinear filter. Inspired by what presented in [35] in the context of *offline* path planning, one could devise an *online* strategy meant to adjust the input UAV trajectory (e.g., the velocity commands) in order to maintain a desired level of *excitement*, in our case represented by the quantity $\sigma_d = \sqrt{\alpha}\|\boldsymbol{\Omega}\|$. This could result in, e.g., execution of small circular trajectories when the vehicle is commanded to hover in place with, instead, a more precise tracking of the desired/commanded velocities when undergoing more aggressive manoeuvres.

On a similar note, we are currently investigating the possibility to dynamically adapt gain α for the nonlinear observer. Indeed, in the presence of noisy sensor readings, it might be desirable to start with a high value of α for imposing an initial



(a) (b) (c)

Fig. 18: Experimental results — Estimated and ground truth linear velocity during a closed loop flight under velocity control for the (a) x, (b) y, and (c) z axes. An object was moved through the field of view at $t = 125\,s$ inducing a transient disturbance on the estimated velocity.

quick error convergence, and to then reduce it when a sufficient convergence level is reached so as to obtain a smoother state estimation (see also Fig. 17b). Finally, it would also be worth to investigate the possibility to extend the structure of the nonlinear observer in order to account for possible *biases* in the employed measurements: this is, for instance, the case of the squared term $\boldsymbol{x}_1\boldsymbol{x}_1^T$ in (42) which, being $\boldsymbol{x}_1 = \boldsymbol{v}/d$ a noisy measurement, will introduce a non-compensated bias in the observer dynamics. While presence of this bias had, in practice, a negligible effect in the reported results, an extension able to compensate for it online would clearly provide an interesting theoretical and practical improvement (especially in situations involving higher level of noise in the measurements).

Appendix A: Index to Multimedia Extensions

The multimedia extensions to this article are at: http://www.ijrr.org.

Extension	Type	Description
1	Video	Experiments of UAV closed-loop control with only onboard sensing and processing power

VI. Acknowledgments

The authors would like to thank Dr. Antonio Franchi and Martin Riedel for their valuable suggestions and contribution on the development of the underlying software framework. Part of Heinrich Bülthoff's research was supported by the Brain Korea 21 PLUS Program through the National Research Foundation of Korea funded by the Ministry of Education. Correspondence should be directed to Heinrich H. Bülthoff.

References

[1] EU Collaborative Project ICT-248669, "AIRobots," www.airobots.eu.

[2] EU Collaborative Project ICT-287617, "ARCAS," www.arcas-project.eu.

[3] EU Collaborative Project ICT-600958, "Sherpa," www.sherpa-project.eu/sherpa.

[4] S. Weiss, M. W. Achtelik, S. Lynen, M. C. Achtelik, L. Kneip, M. Chli, and R. Siegwart, "Monocular Vision for Long-term Micro Aerial Vehicle State Estimation: A Compendium," *Journal of Field Robotics*, vol. 30, no. 5, pp. 803–831, 2013.

[5] D. Scaramuzza, M. C. Achtelik, L. Doitsidis, F. Fraundorfer, E. B. Kosmatopoulos, A. Martinelli, M. W. Achtelik, M. Chli, S. A. Chatzichristofis, L. Kneip, D. Gurdan, L. Heng, G. H. Lee, S. Lynen, L. Meier, M. Pollefeys, R. Siegwart, J. C. Stumpf, P. Tanskanen, C. Troiani, and S. Weiss, "Vision-Controlled Micro Flying Robots: from System Design to Autonomous Navigation and Mapping in GPS-denied Environments," *IEEE Robotics and Automation Magazine*, vol. 21, no. 3, pp. 26–40, 2014.

[6] A. Martinelli, "Vision and IMU Data Fusion: Closed-Form Solutions for Attitude, Speed, Absolute Scale, and Bias Determination," *Transactions on Robotics*, vol. 28, no. 1, pp. 44–60, 2012.

[7] M. Li and A. I. Mourikis, "High-precision, consistent EKF-based visual-inertial odometry," *International Journal of Robotics Research*, vol. 32, no. 6, pp. 690–711, 2013.

[8] S. Omari and G. Ducard, "Metric Visual-Inertial Navigation System Using Single Optical Flow Feature," in *Proceedings of the European Control Conference*, Zurich, Switzerland, 2013, pp. 1310–1316.

[9] G. Nützi, S. Weiss, D. Scaramuzza, and R. Siegwart, "Fusion of IMU and Vision for Absolute Scale Estimation in Monocular SLAM," *Journal of Intelligent & Robotic Systems*, vol. 61, no. 1-4, pp. 287–299, 2011.

[10] M. Faessler, F. Fontana, C. Forster, and D. Scaramuzza, "Automatic Re-Initialization and Failure Recovery for Aggressive Flight with a Monocular Vision-Based Quadrotor," in *Proceedings of the International Conference on Robotics and Automation*, Seattle, WA, USA, 2015.

[11] G. Klein and D. Murray, "Parallel Tracking and Mapping for Small AR Workspaces," in *Proceedings of the International Symposium on Mixed and Augmented Reality*, Nara, Japan, 2007, pp. 225–234.

[12] C. Forster, M. Pizzoli, and D. Scaramuzza, "SVO: Fast Semi-Direct Monocular Visual Odometry," in *Proceedings of the International Conference on Robotics and Automation*, Hong Kong, China, 2014.

[13] B. Herissé, T. Hamel, R. Mahony, and F.-X. Russotto, "Landing a VTOL Unmanned Aerial Vehicle on a Moving Platform Using Optical Flow," *Transactions on Robotics*, vol. 28, no. 1, pp. 77–89, 2012.

[14] D. Honegger, L. Meier, P. Tanskanen, and M. Pollefeys, "An Open Source and Open Hardware Embedded Metric Optical Flow CMOS Camera for Indoor and Outdoor Applications," in *Proceedings of the International Conference on Robotics and Automation*, Karlsruhe, Germany, 2013, pp. 1736–1741.

[15] S. Weiss, M. W. Achtelik, S. Lynen, M. Chli, and R. Siegwart, "Real-time Onboard Visual-Inertial State Estimation and Self-Calibration of MAVs in Unknown Environments," in *Proceedings of the International Conference on Robotics and Automation*, Saint Paul, MN, USA, 2012, pp. 957–964.

[16] S. Weiss, R. Brockers, and L. Matthies, "4DoF Drift Free Navigation Using Inertial Cues and Optical Flow," in *Proceedings of the International Conference on Intelligent Robots and Systems*, Tokyo, Japan, 2013, pp. 4180–4186.

[17] R. Spica and P. Robuffo Giordano, "A Framework for Active Estimation: Application to Structure from Motion," in *Proceedings of the Conference on Decision and Control*, 2013, pp. 7647–7653.

[18] R. Spica, P. Robuffo Giordano, and F. Chaumette, "Active Structure from Motion: Application to Point, Sphere, and Cylinder," *IEEE Trans. on Robotics*, vol. 30, no. 6, pp. 1499–1513, 2014.

[19] V. Grabe, H. H. Bülthoff, and P. Robuffo Giordano, "On-board Velocity Estimation and Closed-loop Control of a Quadrotor UAV based on Optical Flow," in *Proceedings of the International Conference on Robotics and Automation*, St. Paul, MN, USA, 2012, pp. 491–497.

[20] ——, "Robust Optical-Flow Based Self-Motion Estimation for a Quadrotor UAV," in *Proceedings of the International Conference on Intelligent Robots and Systems*, Vilamoura, Portugal, 2012, pp. 2153–2159.

[21] ——, "A Comparison of Scale Estimation Schemes for a Quadrotor UAV based on Optical Flow and IMU Measurements," in *Proceedings of the International Conference on Intelligent Robots and Systems*, Tokyo, Japan, 2013, pp. 5193–5200.

[22] Y. Ma, S. Soatto, J. Kosecka, and S. S. Sastry, *An Invitation to 3-D Vision*. Springer, 2004.

[23] D. Scaramuzza and F. Fraundorfer, "Visual Odometry: Part I - The First 30 Years and Fundamentals," *IEEE Robotics and Automation Magazine*, vol. 18, no. 4, pp. 80–92, 2011.

[24] F. Chaumette and S. Hutchinson, "Visual Servo Control, Part I. Basic Approaches," *Robotics & Automation Magazine*, vol. 13, no. 4, pp. 82–90, 2006.

[25] P. Robuffo Giordano, A. De Luca, and G. Oriolo, "3D Structure Identification from Image Moments," in *Proceedings of the International Conference on Robotics and Automation*, 2008, pp. 93–100.

[26] A. De Luca, G. Oriolo, and P. Robuffo Giordano, "Feature Depth Observation for Image-based Visual Servoing: Theory and Experiments," *International Journal of Robotics Research*, vol. 27, no. 10, pp. 1093–1116, 2008.

[27] J. Civera, A. J. Davison, and J. Montiel, "Inverse Depth Parametrization for Monocular SLAM," *Transactions on Robotics*, vol. 24, no. 5, pp. 932–945, 2008.

[28] J. Lobo and J. Dias, "Relative Pose Calibration Between Visual and Inertial Sensors," *The International Journal of Robotics Research*, vol. 26, no. 6, pp. 561–575, 2007.

[29] R. Mahony, T. Hamel, and J.-M. Pflimlin, "Nonlinear Complementary Filters on the Special Orthogonal Group," *Transactions on Automatic Control*, vol. 53, no. 5, pp. 1203–1218, 2008.

[30] A. Eudes, P. Morin, R. Mahony, and T. Hamel, "Visuo-inertial fusion for homography-based filtering and estimation," in *Proceedings of the International Conference on Intelligent Robots and Systems*, Tokyo, Japan, 2013, pp. 5186–5192.

[31] R. Marino and P. Tomei, *Nonlinear Control Design: Geometric, Adaptive and Robust*. Prentice Hall, 1995.

[32] V. Grabe, M. Riedel, H. H. Bülthoff, P. Robuffo Giordano, and A. Franchi, "The TeleKyb Framework for a Modular and Extendible ROS-based Quadrotor Control," in *Proceedings of the European Conference on Mobile Robots*, Barcelona, Spain, 2013, pp. 19–25.

[33] E. Rosten and T. Drummond, "Fusing points and lines for high performance tracking," in *Proceedings of the International Conference on Computer Vision*, Beijing, China, 2005, pp. 1508–1515.

[34] B. D. Lucas and T. Kanade, "An Iterative Image Registration Technique with an Application to Stereo Vision," in *Proceedings of the International Joint Conference on Artificial Intelligence*, Vancouver, BC, Canada, 1981, pp. 674–679.

[35] M. W. Achtelik, S. Weiss, M. Chli, and R. Siegwart, "Path Planning for Motion Dependent State Estimation on Micro Aerial Vehicles," in *Proceedings of the International Conference on Robotics and Automation*, 2013, pp. 3926–3932.