

A method for ego-motion estimation in micro-hovering platforms flying in very cluttered environments

Adrien Briod¹ · Jean-Christophe Zufferey² · Dario Floreano¹

Received: 26 November 2014 / Accepted: 14 August 2015
© Springer Science+Business Media New York 2015

Abstract We aim at developing autonomous miniature hovering flying robots capable of navigating in unstructured GPS-denied environments. A major challenge is the miniaturization of the embedded sensors and processors that allow such platforms to fly by themselves. In this paper, we propose a novel ego-motion estimation algorithm for hovering robots equipped with inertial and optic-flow sensors that runs in real-time on a microcontroller and enables autonomous flight. Unlike many vision-based methods, this algorithm does not rely on feature tracking, structure estimation, additional distance sensors or assumptions about the environment. In this method, we introduce the *translational optic-flow direction constraint*, which uses the optic-flow direction but not its scale to correct for inertial sensor drift during changes of direction. This solution requires comparatively much simpler electronics and sensors and works in environments of any geometry. Here we describe the implementation and performance of the method on a hovering robot equipped with eight 0.65 g optic-flow sensors, and show that it can be used for closed-loop control of various motions.

Keywords Aerial robotics · Sensor fusion · Ego-motion estimation · Optic-flow

1 Introduction

The use of robots for the exploration of dangerous environments, such as collapsed buildings or nuclear plants after natural catastrophes, would avoid risking human lives. Compared to ground-based ones, flying robots have many advantages as they provide an elevated view-point and can navigate above rubble more efficiently. However, most flying platforms are unstable by design and need semi-autonomous or fully autonomous flight capabilities in order to be used by non-expert pilots in the field. Typically, they should be capable of performing autonomous tasks, such as stable hovering or following predetermined trajectories. In order to solve this challenge in GPS-denied environments, embedded sensors have to be used for position or velocity estimation, which is called ego-motion estimation. However, payload is certainly limited on-board small flying platforms, thus, the size and weight of embedded sensors used in such MAVs must be minimized. Tiny flying platforms weighing no more than a few grams exist (Kushleyev et al. 2013; Wood 2008), but no appropriate embedded solution for autonomous control for such robots has been achieved yet.

Because of their versatility and relatively low weight, monocular vision sensors are often used for ego-motion estimation. However, monocular vision sensors are affected by the scale ambiguity, which is the inability to distinguish the scale of a translation (Scaramuzza and Fraundorfer 2011). Visual information is thus usually converted to metric measurements with a 'scale factor', which is often identified by using additional sensors. One of the most popular solution is inertial SLAM (Simultaneous Localization and Mapping)

Electronic supplementary material The online version of this article (doi:10.1007/s10514-015-9494-4) contains supplementary material, which is available to authorized users.

✉ Adrien Briod
adrien.briod@epfl.ch

Dario Floreano
dario.floreano@epfl.ch

¹ The Laboratory of Intelligent Systems (LIS), Ecole Polytechnique Fédérale de Lausanne (EPFL), 1015 Lausanne, Switzerland

² SenseFly Ltd, 1033 Cheseaux-sur-Lausanne, Switzerland

as demonstrated by [Corke et al. \(2007\)](#), [Kelly and Sukhatme \(2010\)](#), [Jones and Soatto \(2011\)](#), [Scaramuzza et al. \(2013\)](#) and [Weiss et al. \(2012b\)](#). In this case, the scale factor is eventually obtained from accelerometers and is observable if linear accelerations are present ([Martinelli 2012](#)). Other SLAM approaches use manual scale initialization ([Blösch et al. 2010](#)) or stereo information ([Shen et al. 2013](#)). These methods allow for absolute positioning and even reconstruct the 3D structure of the environment in the process.

SLAM algorithms require relatively high processing power and memory, which may result in fairly bulky setups. A simpler approach consists of fusing inertial data directly with the epipolar constraint, one of the fundamental relations of multi-camera geometry ([Diel et al. 2005](#); [Taylor 2008](#); [Taylor et al. 2011](#); [Mourikis and Roumeliotis 2007](#); [Weiss et al. 2012a](#)). Applying directly this constraint by using the position of features in two different images bypasses the need for structure re-construction, and thus reduces significantly the amount of processing and memory required ([Taylor et al. 2011](#)). Also, it is possible to correlate measurements of features tracked multiple frames apart to minimize drift ([Diel et al. 2005](#); [Mourikis and Roumeliotis 2007](#)). However, the scale of the motion is not provided by epipolar constraint updates, and the relative scale is not conserved, which generally provokes large estimation errors as mentioned in [Taylor et al. \(2011\)](#). In [Taylor \(2008\)](#), the approach suffers from errors on the velocity estimation along the direction of motion, which pushed the authors to add an airspeed sensor. An interesting approach in [Weiss et al. \(2012a\)](#) suggests to estimate the average depth of the tracked features, so that the camera can be used as a metric sensor. The experiments present results where the same two features are tracked constantly, but it is unclear whether the method would adapt well to continuously changing features at varying depths.

All methods described above rely on feature tracking, which requires cameras with high enough resolution and sufficient computing power for the feature extraction and matching algorithms ([Fraundorfer and Scaramuzza 2012](#)). On the other hand, comparatively simpler methods and sensors exist for optic-flow extraction. Optic-flow is the apparent motion of the scene at discrete points in the field-of-view, and can typically be obtained from the variation of pixel intensity over time ([Lucas and Kanade 1981](#); [Srinivasan 1994](#)), or pattern displacement ([Kendoul et al. 2009](#)). Optic-flow sensors work at low resolution and thus exist in very small and cheap packages, such as a 1.9 g panoramic sensor using flat optics by [Floreano et al. \(2013\)](#), a 0.8 g sensor used for obstacle avoidance by [Beyeler et al. \(2009\)](#), a 3 g sensor rig comprising 8 sensors by [Barrows et al. \(2006\)](#), or even a 33 mg sensor used for altitude control by [Duhamel et al. \(2013\)](#). Also, optic-flow can be extracted from a scene that does not present recognizable features, such as blurry textures or repetitive patterns, or even in the dark ([Floreano](#)

[et al. 2013](#)). Optic-flow has been used on MAVs in the past for bio-inspired obstacle avoidance ([Zufferey and Floreano 2006](#); [Beyeler et al. 2009](#)), or speed regulation ([Ruffier and Franceschini 2005](#); [Barrows et al. 2006](#)).

Optic-flow has two main drawbacks compared to feature tracking when it comes to ego-motion estimation: (a) Optic-flow information is only related to the motion and is generated by unidentified visual cues, which prevents it from being used for absolute localization or in global algorithms like bundle adjustment ([Triggs et al. 2000](#)) or SLAM ([Davison 2003](#); [Klein and Murray 2007](#)). Ego-motion obtained from optic-flow is thus very likely to present position drift (error accumulation over time), because it is obtained by integrating velocity estimates. (b) The scale factor affecting optic-flow measurements changes at each step because the visual cues generating optic-flow are always different. On the other hand, feature tracking typically allows to retain a constant relative scale between all measurements if features are tracked for three frames or more ([Hartley and Zisserman 2004](#)). It is thus comparatively harder to convert optic-flow information into a metric value. It is possible in theory to obtain an absolute motion estimate from optic-flow sensors whose vantage points are separated with a certain distance ([Chen et al. 2001](#)). However, this only works in practice for large distances considering the resolution and noise level of typical optic-flow sensors ([Kim and Brambley 2007](#)). For these reasons, optic-flow alone is generally considered not suitable for visual odometry applications ([Scaramuzza and Fraundorfer 2011](#)).

It is however possible to estimate the direction of motion from optic-flow measurements ([Nelson and Aloimonos 1988](#); [Chen et al. 2001](#); [Schill and Mahony 2011](#)) and even a velocity that is scaled inversely proportionally to the average depth of the environment, which is used by [Barrows et al. \(2006\)](#) to control a MAV hovering at a fixed point. However it is unclear how this strategy can handle inhomogeneous distance distributions or maneuvers provoking the depth to change constantly. Also, when the scale of the velocity estimate is sensitive to the environment's size, the control parameters have to be tuned differently for each type of environment, which is not practical. Assuming a known average depth of the environment, it is possible to estimate the ego-motion in absolute scale ([Franz et al. 2004](#)), even though this is only applicable to specific situations. Inertial sensors are used by [Kendoul et al. \(2009\)](#) and [Herisse et al. \(2008\)](#) to obtain the scale of optic-flow measurements, but the methods only work over flat surfaces. A solution based on stereo vision to obtain depth is proposed by [Honegger et al. \(2012\)](#), but this requires multiple cameras with different vantage points and additional processing. A successful method that allows to solve the scale ambiguity problem is to couple optic-flow sensors with distance sensors, which continuously provide the absolute scale. For example in [Bristeau et al. \(2011\)](#) or

Honegger et al. (2013), an ultrasonic distance sensor is used together with a camera pointing downwards, which results in a very good velocity estimation and thus low position drift. However, such a solution adds bulkiness to the system and only works up to the limited range of the distance sensor.

It can be seen from the prior art that most solutions use vision as a metric sensor for ego-motion measurement, which requires determining the scale of the visual information. Feature tracking keeps the scale factor constant, which facilitates its estimation (Weiss et al. 2012b). On the other hand, optic-flow offers better potential for miniaturization, but is hard to convert into metric information, especially in unstructured environments where no simplifying assumptions can be made.

In this paper, we present a novel algorithm for ego-motion estimation based on inertial sensors and non-scaled optic-flow. This paper follows up on a workshop paper (Briod et al. 2013) which introduced the method and demonstrated qualitatively its use for closed-loop control of a 46 g quadrotor. Here, we characterized this approach quantitatively by means of a 278 g quadrotor equipped with eight sensors, infrared beacons and a motion capture system providing ground-truth values. The new sensor disposition also allows for the characterization of different viewing direction configurations. Section 2 describes the characteristics and implementation of the algorithm and Sect. 3 presents the experimental setup used for characterization. Section 4 describes the results of the experiments carried out for various motion types and in diverse environments.

2 Method for optic-flow and inertial sensor fusion

Assuming a moving robot equipped with inertial sensors (gyroscopes and accelerometers) and multiple optic-flow sensors, the goal is to estimate the robot's velocity by means of sensor fusion. We define the body frame as the frame moving with the robot, and assume the inertial sensors and vantage points of the optic-flow sensors to be positioned at its origin.

Herein, we complement the temporal integration of the linear acceleration provided by the inertial sensors with the optic flow information. Instead of using the scale of the translation-induced optic-flow, which depends on the generally unknown distance to the visual cues, we suggest to use only its direction, which is not environment-dependent and can thus be used with higher confidence. Similarly to the epipolar constraint, the translation-induced optic-flow direction defines a constraint on the velocity vector, which we call the translational optic-flow direction constraint (TOFDC). This is used to correct the inertial navigation velocity drift. Contrarily to the epipolar constraint, the TOFDC is only based on the instantaneous motion of the scene.

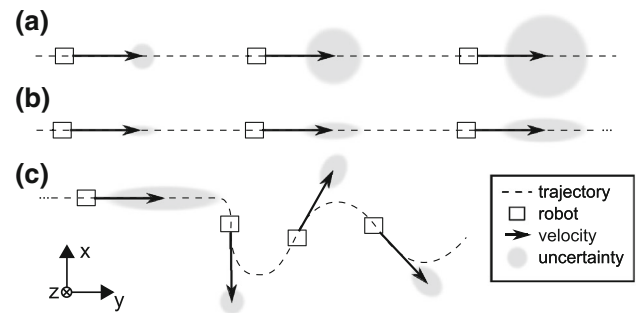


Fig. 1 Conceptual 2D drawing illustrating how the velocity of a moving robot can be observed by using a translational optic-flow direction constraint (TOFDC) to aid inertial navigation. An optic-flow sensor pointing towards the Z direction provides a 2D measurement vector (of unknown scale) defining a direction constraint on the velocity vector. The uncertainty on the velocity estimation is illustrated with a grey shading in 3 different situations. A large area represents a high uncertainty and a small area a good accuracy. In situation **a**, only inertial navigation is performed, in **b** a direction constraint is applied and in **c** changes of direction are executed, which allows to keep the uncertainty bounded. Note that the robot is always shown in the same orientation, but this principle applies regardless of the rotations experienced by the robot

The metric scale of the motion is solely provided by inertial navigation, which provides a fairly accurate velocity estimate in the short term. However this estimate drifts significantly over time if no correction is applied (Titterton 2004). The TOFDC constrains one degree of freedom of the velocity estimate to a half-plane. If more than one optic-flow sensors are used, up to two degrees of freedom of the velocity estimate can be constrained along a vector as mentioned by Chen et al. (2001) and Schill and Mahony (2011), thus keeping only one degree of freedom unobserved and subject to drift. Figure 1 illustrates in 2D how the velocity is observable in TOFDC-aided inertial navigation. First, situation (a) shows a growing uncertainty when only inertial navigation is performed. Then, a direction constraint is applied in situation (b) and (c). According to Dissanayake and Sukkariéh (2001), where inertial navigation is corrected by a vehicle model constraining the speed along a direction, such a system is observable during changes of direction only. The key is that the unobserved degree of freedom is along the direction of motion, and thus, all degrees of freedom can be observed if the sensors undergo sufficient changes in direction of motion. Note that this applies independently of the rotation experienced by the sensors. It is expected that the more changes of direction, the better the velocity estimate will be since the drift will be limited in time. This assumption is tested and characterized in practical experiments in Sect. 4. The goal of this paper is not to provide a formal mathematical proof of these statements, but rather to explain intuitively the basic principles of this method and then demonstrate and characterize it in practical experiments on a flying robot.

2.1 Algorithm implementation

In order to implement this idea, we suggest to use an Extended Kalman Filter (EKF) (Grewal and Andrews 2001) to aid an inertial navigation process with optic-flow sensors, each providing a TOFDC. The goal of the implementation is to validate the idea presented above on a hovering MAV, and the focus is thus on keeping the algorithm simple enough for real-time operation on a micro-controller, which leads to a few assumptions and simplifications. In this filter, the state vector \mathbf{x} comprises 6 elements, the velocity $\mathbf{v}^b = (v_x^b, v_y^b, v_z^b)$ and accelerometer biases $\mathbf{b}^b = (b_x^b, b_y^b, b_z^b)$, both expressed in body frame (superscript b):

$$\mathbf{x} = \begin{bmatrix} \mathbf{v}^b \\ \mathbf{b}^b \end{bmatrix} \quad (1)$$

Figure 2 describes each component of the ego-motion estimation algorithm. The prediction step of the EKF estimates a velocity vector by integrating linear accelerations, which are obtained from the accelerometers after compensation of the gravity and the centrifugal acceleration. The TOFDC is then applied in sequential Kalman updates for each optic-flow sensor, which allows to correct for velocity drift and to estimate accelerometer biases. Note that using the body frame as the computational frame requires to compensate for centrifugal accelerations and is not convenient for position estimation, but it reduces the number of necessary frame transformations since the accelerometer and optic-flow measurements are obtained in body coordinates. We use an independent separate module for attitude estimation (rather than integrating it to the Kalman filter) so as to limit the size of the state \mathbf{x} and reduce the amount of computations. This cascaded configuration is however sub-optimal since orientation errors are correlated to velocity errors and could thus be reduced by TOFDC updates, which would subsequently improve the velocity estimate. The estimation of \mathbf{b}^b allows to compensate for accelerometer biases due to calibration errors or temperature changes, as well as for other errors affecting the velocity integration indirectly, such as orientation estimation

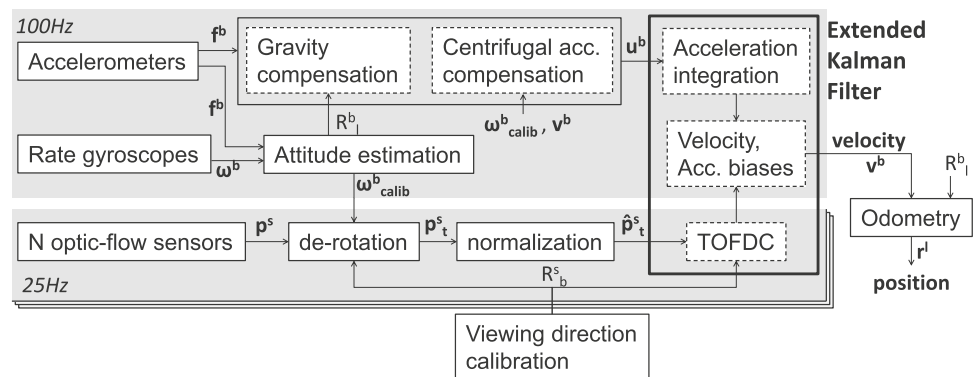
errors. Note that the biases are not used in the attitude estimation module in order to avoid generating a feedback loop with unpredictable effects. A position estimate is obtained by odometry at a later stage, using the output of the EKF. Note that the azimuth (or yaw angle) is unknown because no GPS or magnetic compass is used, and thus the azimuth of the local frame where odometry is performed is arbitrary and drifts over time.

2.1.1 Inertial navigation

Inertial navigation is the process of using only inertial sensors (generally 3 axis rate gyroscopes and 3 axis accelerometers) to estimate the velocity and position of the sensors (Titterton 2004). This method called dead reckoning relies on a correct initialization of the velocity and position, and the integration over time of the measured linear accelerations to obtain a velocity estimate, and integrating again to obtain a position estimate. An initialization is required, as inertial sensors only measure a rate of change, but not an absolute value. In addition, since noise or offsets accumulate over time, the velocity or position estimate will diverge from the actual value, with a rate that depends on the quality of the sensors. When cheap MEMS sensors are used, this drift is very substantial after several seconds already, ranging from tens to hundreds of meter of position drift after less than a minute, and are thus useless for navigation without corrections from some type of absolute sensors. For example, inertial navigation drift obtained with our calibrated MEMS sensors during flight is of 5 m/s and 200 m in average after 1 min.

In the prediction step of the EKF, we use inertial navigation to obtain an initial velocity estimate by means of linear accelerations integration. We assume that the local frame (superscript l) is an inertial frame, since the effects of the earth's rotation are below the noise level of our sensors for the types of motion envisioned here for hovering MAVs. Note that this assumption may not hold for other types of motion (e.g. at high speed) or with high-end sensors. Therefore, linear accelerations \mathbf{u}^b in body frame can be obtained by removing the gravity \mathbf{g}^l and the centrifugal accelerations

Fig. 2 Block diagram showing the proposed algorithm for fusion of inertial sensors and multiple optic-flow sensors. Superscript l stands for local frame (assumed to be an inertial frame). Superscript b stands for body frame (robot and inertial sensor frame). Superscript s stands for sensor frame (one frame per optic-flow sensor)



from the measured accelerations \mathbf{f}^b using the following equation (Titterton 2004):

$$\mathbf{u}^b = \mathbf{f}^b - \mathbf{R}_l^b \cdot \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix} - \boldsymbol{\omega}^b \times \mathbf{v}^b \quad (2)$$

where \mathbf{R}_l^b is the rotation matrix describing the attitude of the robot, $\boldsymbol{\omega}^b$ is the angular velocity and \mathbf{v}^b is the velocity estimate in the body frame. The attitude of the robot \mathbf{R}_l^b is estimated by means of a standard quaternion-based filter (Kim and Golnaraghi 2004) that also estimates gyroscope biases and thus provides a calibrated angular rate value $\boldsymbol{\omega}_{calib}^b$. The attitude estimation relies on the assumption that measured accelerations are pointing downwards in average ($\mathbf{f}^b \approx \mathbf{g}^b$), which is true for typical motions of MAVs (constant speed or frequent changes of direction). However, this assumption does not hold if constant accelerations are present for a significant amount of time, for example during a coordinated turn or while accelerating along a straight path for tens of seconds, in which case it would be very beneficial to implement a filter whose state comprises both the velocity and orientation. Note that the initialization of the orientation filter is achieved from a single accelerometer reading, which is assumed to indicate the down direction, while the yaw angle is arbitrarily set to 0° .

The EKF prediction (denoted with \sim) is thus the following:

$$\tilde{\mathbf{x}}_k = \mathbf{x}_{k-1} + \begin{bmatrix} (\mathbf{u}^b + \mathbf{b}^b) \cdot \Delta t \\ \mathbf{0}_{3 \times 1} \end{bmatrix} \quad (3)$$

$$\tilde{\mathbf{P}}_k = \Phi_k \mathbf{P}_{k-1} \Phi_k^T + \mathbf{Q}_k \quad (4)$$

where Δt is the integration period, $\tilde{\mathbf{P}}_k$ the covariance matrix, Φ_k is the state transition matrix :

$$\Phi_k = \begin{bmatrix} 1 & w_z^b \cdot \Delta t & -w_y^b \cdot \Delta t & \Delta t & 0 & 0 \\ -w_z^b \cdot \Delta t & 1 & w_x^b \cdot \Delta t & 0 & \Delta t & 0 \\ w_y^b \cdot \Delta t & -w_x^b \cdot \Delta t & 1 & 0 & 0 & \Delta t \\ \mathbf{0}_{3 \times 3} & & & \mathbf{I}_{3 \times 3} & & \end{bmatrix} \quad (5)$$

and \mathbf{Q}_k is the prediction noise matrix:

$$\mathbf{Q}_k = \begin{bmatrix} \text{diag}(\sigma_a^2) & \mathbf{0}_{3 \times 3} \\ \mathbf{0}_{3 \times 3} & \text{diag}(\sigma_b^2) \end{bmatrix} \quad (6)$$

where σ_a allows to tune how much weight is given to the prediction and σ_b allows to tune how fast the biases adapt.

2.1.2 Optic-flow correction

An optic-flow sensor is assumed to have a fixed viewing direction with respect to the body frame, and to provide one optic-flow measurement (note that a physical imager may

comprise multiple optic-flow sensors under this definition). Assuming a projection of the scene on a unit sphere centered at the vantage point, each optic-flow measurement can be expressed as a 3D vector tangent to the unit sphere and perpendicular to the viewing direction (Koenderink and Doorn 1987):

$$\mathbf{p} = \underbrace{-\boldsymbol{\omega} \times \mathbf{d}}_{\mathbf{p}_r} - \underbrace{\frac{\mathbf{v} - (\mathbf{v} \cdot \mathbf{d})\mathbf{d}}{D}}_{\mathbf{p}_t} \quad (7)$$

where \mathbf{d} is a unit vector describing the viewing direction, $\boldsymbol{\omega}$ the angular speed vector, \mathbf{v} the translational velocity vector and D the distance to the object seen by the sensor. The measured optic-flow \mathbf{p} can be expressed in two parts, namely the rotation-induced or 'rotational' optic-flow \mathbf{p}_r and translation-induced or 'translational' optic-flow \mathbf{p}_t .

Only the translational optic-flow \mathbf{p}_t is useful for velocity estimation, and the rotational optic-flow \mathbf{p}_r is considered here as a disturbance that is removed from the measurement \mathbf{p} using a process called de-rotation (Srinivasan et al. 2009). We use a method that we proposed in Briod et al. (2012) in order to automatically calibrate the viewing direction of optic-flow sensors in an initial step and then execute the de-rotation with rate gyroscopes. In theory, rotations should thus not affect the outcome of the algorithm, but in practice the de-rotation procedure may introduce noise, especially if the amplitude of rotational optic-flow is much larger than translational optic-flow.

To express the translational optic-flow direction constraint, we define a sensor frame (s superscript). By convention, the Z axis of the sensor frame (X_s, Y_s, Z_s) is aligned with the viewing direction \mathbf{d} and the X_s and Y_s axes define the image plane of the optic-flow sensor, as shown in Fig. 3. We can thus express the following vectors:

$$\mathbf{p}_t^s = \begin{bmatrix} p_{t,x}^s \\ p_{t,y}^s \\ 0 \end{bmatrix}, \mathbf{d}^s = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \text{ and } \mathbf{v}^s = \mathbf{R}_b^s \mathbf{v}^b = \begin{bmatrix} v_x^s \\ v_y^s \\ v_z^s \end{bmatrix}, \quad (8)$$

where \mathbf{v}^s is the velocity vector expressed in the sensor frame and \mathbf{R}_b^s is the rotation matrix that describes the orientation of the sensor with respect to the body frame.

The translational optic-flow can be rewritten as:

$$\begin{bmatrix} p_{t,x}^s \\ p_{t,y}^s \\ 0 \end{bmatrix} = -\frac{1}{D} \begin{bmatrix} v_x^s \\ v_y^s \\ 0 \end{bmatrix} \quad (9)$$

which is a relation between 2D vectors:

$$\mathbf{v}^I = -D \cdot \mathbf{p}_t^I \quad (10)$$

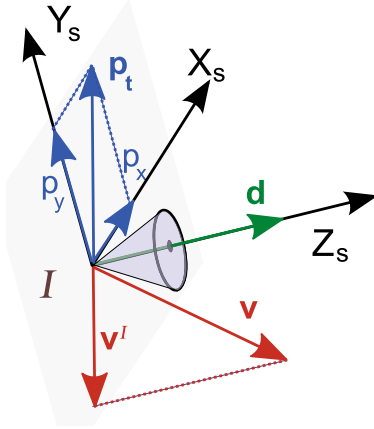


Fig. 3 The direction of the translational optic-flow vector \mathbf{p}_t depends on the velocity \mathbf{v} and the unit vector \mathbf{d} , pointing toward the viewing direction of the sensor. The translational optic-flow direction constraint (TOFDC) is expressed in the image plane I , and states that the projection \mathbf{v}^I of the velocity vector \mathbf{v} onto the image plane I has to be collinear to the translational optic-flow vector \mathbf{p}_t and of opposite direction

where $\mathbf{p}_t^I = [p_{t,x}^s, p_{t,y}^s]$ is the 2D translational optic-flow measurement, $D \in \mathbb{R}_{>0}$ is the distance to the visual cue, and

$$\mathbf{v}^I = \begin{bmatrix} v_x^s \\ v_y^s \end{bmatrix} = \begin{bmatrix} r_{11}^s & r_{12}^s & r_{13}^s \\ r_{21}^s & r_{22}^s & r_{23}^s \end{bmatrix} \mathbf{v}^b \quad (11)$$

is the projection of the velocity on the image plane. r_{ij}^s are the elements of the first two rows of the rotation matrix \mathbf{R}_b^s , which are calibrated in advance by means of an initial calibration process, for instance as described in Briod et al. 2012.

Equation (10) highlights the difficulty to convert optic-flow to a metric velocity measurement, because the distance D is apriori unknown and changes constantly in cluttered environments. However Eq. (10) states that, regardless of D , the projection \mathbf{v}^I of the velocity vector \mathbf{v} onto the image plane I has to be collinear to the translational optic-flow vector \mathbf{p}_t^I and of opposite direction. This constraint is what we call the translational optic-flow direction constraint (TOFDC), which can be expressed by using normalized vectors :

$$\hat{\mathbf{p}}_t^I = -\hat{\mathbf{v}}^I \quad (12)$$

where $\hat{\mathbf{p}}_t^I$ and $\hat{\mathbf{v}}^I$ are unit vectors:

$$\hat{\mathbf{p}}_t^I = \frac{\mathbf{p}_t^I}{\|\mathbf{p}_t^I\|} \text{ and } \hat{\mathbf{v}}^I = \frac{\mathbf{v}^I}{\|\mathbf{v}^I\|} \quad (13)$$

Considering the following Kalman update equations:

$$\mathbf{K}_k = \tilde{\mathbf{P}}_k \mathbf{H}_k^T (\mathbf{H}_k \tilde{\mathbf{P}}_k \mathbf{H}_k^T + \mathbf{R}_k)^{-1} \quad (14)$$

$$\mathbf{x}_k = \tilde{\mathbf{x}}_k + \mathbf{K}_k (\mathbf{z}_k - h[\tilde{\mathbf{x}}_k]) \quad (15)$$

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \tilde{\mathbf{P}}_k \quad (16)$$

we suggest to use the following measurement sequentially for each optic-flow sensor:

$$\mathbf{z}_k = \hat{\mathbf{p}}_t^I \quad (17)$$

and the following non-linear measurement model:

$$h[\tilde{\mathbf{x}}_k] = -\hat{\mathbf{v}}^I \quad (18)$$

The corresponding 2×6 observation matrix is obtained by taking the Jacobian $\mathbf{H}_k = \frac{\partial h[\mathbf{x}]}{\partial \mathbf{x}}$ and comprises the following elements:

$$\mathbf{H}_{k,ij} = \begin{cases} \frac{-r_{ij}^s}{\|\mathbf{v}^I\|} + \frac{v_i^s \cdot (v_1^s \cdot r_{1j}^s + v_2^s \cdot r_{2j}^s)}{\|\mathbf{v}^I\|^3} & \text{for } j = 1, 2, 3 \\ 0 & \text{for } j = 4, 5, 6 \end{cases} \quad (19)$$

The 2×2 measurement noise matrix is set to $\mathbf{R}_k = \text{diag}(\sigma_{of}^2)$ where σ_{of} describes the noise on the optic-flow measurement and can typically be varied in function of some known quality indicators. Typically, it was observed that the noise was higher for small optic-flow values or when a lot of rotational optic-flow is present. We suggest thus to vary σ_{of} according to the following equation:

$$\sigma_{of} = \frac{1}{k_1 + k_2 \cdot \|\mathbf{p}_t\|} \frac{\|\mathbf{p}_r\| + \|\mathbf{p}_t\|}{\|\mathbf{p}_t\|} \quad \text{with } k_1, k_2 \in \mathbb{R}_{>0} \quad (20)$$

The first fraction of Eq. (20) increases the standard deviation σ_{of} for smaller amplitudes of translational optic-flow (as tuned by k_2). The second fraction of Eq. (20) increases the standard deviation σ_{of} when the rotational optic-flow is large relatively to the translational optic-flow. The standard deviation scale can be changed by tuning k_1 and k_2 in parallel.

2.1.3 Odometry

As a last step, the velocity obtained in body frame \mathbf{v}^b needs to be converted in local frame and integrated over time to obtain a position estimate \mathbf{r}^l :

$$\mathbf{r}_k^l = \mathbf{r}_{k-1}^l + \mathbf{R}_b^l \cdot \mathbf{v}^b \cdot \Delta t \quad (21)$$

3 Experimental setup

Experiments are performed with a quadrotor equipped with optic-flow mouse sensors flying under different control schemes. These experiments do not aim at testing the performance under all possible conditions (visual scene, structure, sensor arrangement, motion types, etc.), but rather at assessing the viability of the approach presented in Sect. 2 for autonomous flight control of MAVs in realistic representative

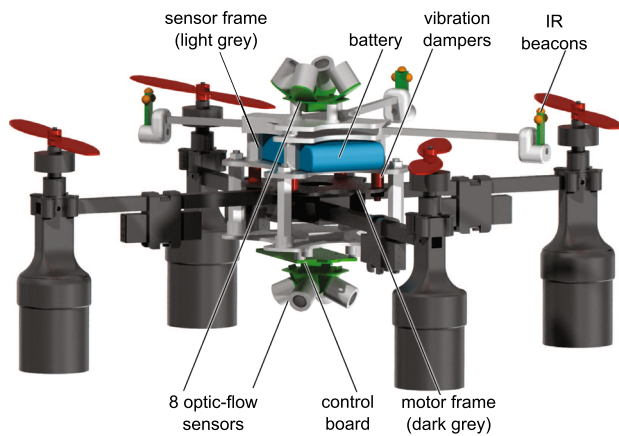


Fig. 4 278 g quadrotor equipped with 8 optic-flow sensors used for the experimental validation of the ego-motion estimation approach. It can be operated fully autonomously by the onboard microcontroller

situations. A motion capture system is used as ground-truth in order to provide insights into the method's performance in a particular environment, and a qualitative analysis is performed in other various indoor environments.

The 278 g quadrotor shown in Fig. 4 was developed specifically for these experiments. The platform is made of carbon and 3D-printed parts and comprises four brushless motors powered by a 1000 mAh 2-cell LiPo battery. A custom control board integrates a STM32 microcontroller with FPU (Floating Point Unit), a ST LIS3DH 3-axis accelerometer, a ST L3G4200 3-axis rate gyro and a Nordic NRF24 radio module. The radio module provides a two-ways link with a ground station and is used for debugging purposes. The biases and factors of the accelerometers are calibrated using the technique described by Pylvanainen (2008) and stored in memory. An 'OpenLog' SD card module is used to record onboard ego-motion estimation and sensor data at 100 Hz. An external receiver (Spektrum DSM2 satellite receiver) is used for sending commands from a remote control.

Two sensor boards, each comprising four 0.65 g optic-flow sensors are mounted on top and bottom of the platform. The sensor boards and battery are mounted on a frame which is decoupled from the motor frame by vibration dampers, so that the measurements are not affected by motor-induced vibrations. A misalignment of the vantage point of each optic-flow sensor with respect to the IMU generates additional translational optic-flow during rotations; in particular, the relative error due to misalignment is equal to the ratio between the offset and distance to obstacles. We make the assumption that the misalignments can be neglected because the offsets are generally not significant compared to the distance to obstacles and the rotations are limited in our applications. This assumption may not hold for other applications or larger platforms, for which the offsets may have to be calibrated.

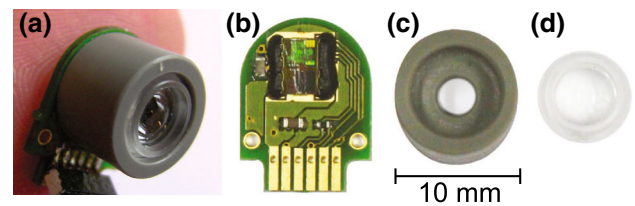


Fig. 5 Custom 0.65 g optic-flow sensor (a), comprising a bare ADNS-9500 chip which is bonded to a custom printed circuit board (b). A lens holder (c) is glued on top of the PCB and hosts a CAY046 lens with 4.6 mm focal length (d)

The optic-flow sensors are comprised of Avago ADNS-9500 optical mouse chips, whose bare dies were bonded to a custom printed circuit board in order to obtain a compact design (Fig. 5). These circuits are generally used in high performance computer mice and include the optic-flow extraction process on the chip. The image sensor is 30×30 pixel wide and is sampled at up to 11,750 fps. The sensors are fitted with a custom lens holder comprising a CAY046 aspheric lens with 4.6 mm focal length providing a 20° field of view. The chip communicates with the control board through SPI and does not provide any pixel value or image, but only the result of the optic-flow extraction performed on the chip, in the form of a 2D optic-flow vector \mathbf{p}_{raw}^I . It corresponds to the image displacement on the sensor in counts per time unit and can be converted to rad/s with the following formula:

$$\mathbf{p}^I = \frac{\mathbf{p}_{raw}^I}{K \cdot f \cdot \Delta t_{of} \cdot Res} \quad (22)$$

where K is a chip-specific constant (0.694 rad^{-1} in the case of the ADNS9500), f is the focal length (4.6 mm in our case), Δt_{of} is the sampling period (40 ms) and Res is the sensor's resolution (1.6×10^5 counts/m).

The 8 optic-flow sensors are arranged so that the viewing directions are spread evenly and symmetrically over the field of view, as shown in Fig. 6. This arrangement is cho-

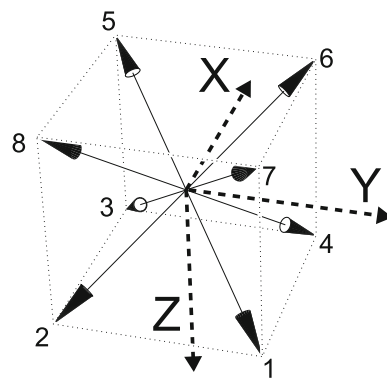


Fig. 6 The 8 optic-flow sensors are arranged so that the viewing directions point towards the corners of a cube centered at the origin and aligned with the body axes (X, Y, Z) of the quadrotor

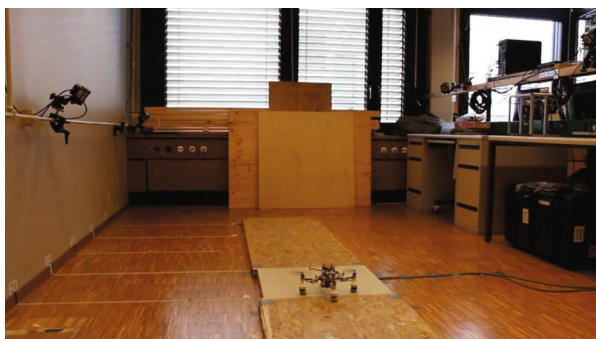


Fig. 7 Motion capture room used for the experiments

sen for the reason that a wide field-of-view is preferred for optic-flow-based ego-motion estimation because, even assuming de-rotated optic-flow, a wide field-of-view is necessary to distinguish clearly the direction of motion out of multiple optic-flow measurements (Nelson and Aloimonos 1988; Chen et al. 2001; Schill and Mahony 2011). Indeed, each optic-flow measurement constrains the velocity to a plane containing the viewing direction of the sensor and the optic-flow vector, and thus a narrow field-of-view is likely to provide planes with similar orientations whose intersection will have a high uncertainty. A wide field-of-view also provides more robustness and redundancy when the visual quality of the environment is irregular. Finally, the symmetry of the viewing direction arrangement is used for statistical analysis of different sensor configurations in Sect. 4.2. The exact orientation of the optic-flow sensors is calibrated as described by Briod et al. (2012), with a precision expected to be in the range of $\pm 2^\circ$. This calibration procedure also allows to determine a potential delay between sensor readings by cross-correlating rotational optic-flow and angular rates. A delay of 20 ms was found in this case, and the IMU readings were thus cached so as to be synchronized with the visual measurements.

Experiments were performed in a room equipped with an OptiTrack¹ motion capture system (Fig. 7), which allows to obtain ground truth measurements. The quadrotor is equipped with 3 infra-red (IR) beacons, each comprising five high intensity LEDs. The position and orientation of the rigid body defined by the three beacons are recorded at 240 Hz by the motion capture system. The position obtained from the motion capture system is smoothed with a Gaussian filter with standard deviation $\sigma = 20$ ms, and is then differentiated to obtain the velocity, which is then converted in the rigid body frame. Motion capture systems using reflective markers generally fill the environment with bright IR lights. In these experiments, we wanted to make sure to avoid any artificial

lighting and we therefore switched off the IR lights and used instead active beacons each comprised of 5 IR LEDs.

The relative orientation between the rigid body and the quadrotor's IMU is determined by least squares during a preliminary calibration procedure. The test room, whose visual appearance has not been altered in any particular way, comprises typical features found in offices: low-contrast walls, windows, clutter (desks, shelves, etc).

The optic-flow sensors are sampled at 25 Hz, the inertial sensors are sampled at 500 Hz and the STM32 runs both the attitude estimation filter and the velocity estimation EKF at 100 Hz. Three low-level PID controllers stabilize the roll, pitch and yaw angles of the robot at 500 Hz, while three high-level PIDs control the lateral position and altitude of the robot at 100 Hz using the velocity and position estimates obtained by the ego-motion algorithm. The high-level control can be switched off or overridden by remote control inputs at all time.

4 Results

The here-presented egomotion estimation algorithm is first characterized by carrying out various remote controlled maneuvers, providing insights on the performance of the algorithm as a function of the type of trajectory. Then, different types of closed-loop control flights are studied. In these experiments, the robot uses the output of the onboard velocity and position estimation as feedback to fly autonomously. Different optic-flow sensor configurations are tested by enabling or disabling some of the eight sensors. These experiments allow testing the influence of sensor number and viewing directions. Finally, autonomous flights are performed in various environments without ground-truth for qualitative analysis of the performance (Online video n°3).

An initial offline manual tuning procedure provided the following values for the 4 Kalman filter parameters: $\sigma_a = 0.02$, $\sigma_b = 0.005$, $k_1 = 1.25$ and $k_2 = 5$ (Eqs. 6, 20). The initial conditions (velocity and biases) are assumed unknown. Therefore the initialization of the Kalman filter is performed by setting the initial velocity and bias estimates to zero and by using a diagonal covariance matrix with arbitrarily high initial values ($P_0 = \text{diag}_{6 \times 6}(10)$ in our case). This ensures that the initial estimates quickly adapt to the measurements.

4.1 Remote-controlled flight

A 3 min remote-controlled flight comprising periods of slow, dynamic, and hover flight is performed in order to assess the performance of the algorithm. Figure 8 compares the onboard velocity estimate to the ground-truth (OptiTrack motion capture system) for each body axis. The graph shows that the estimate is close to the ground truth in most cases, but has

¹ www.naturalpoint.com/optitrack/.

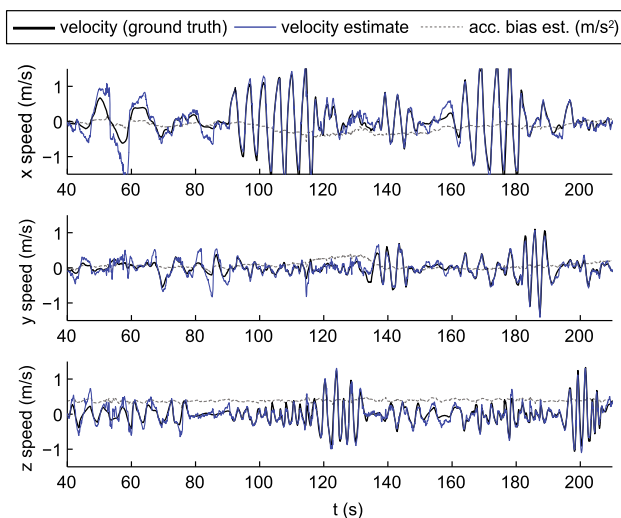


Fig. 8 Ground-truth (black thick line) and onboard estimate (blue thin line) of the quadrotor's velocity in body frame during a 3 min remote-controlled flight comprising periods of slow, hover and dynamic flight. The accelerometer biases estimated by the algorithm, which also account for other errors such as attitude estimation errors, are shown too (dashed line) (Color figure online)

relatively high errors in some other cases. The accelerometer bias estimates are also shown. The full flight can be watched in the online video n°1.

As stated in Sect. 2, changes of direction are crucial to the performance of the algorithm. In order to characterize the effect of changes of direction, we analyze the estimation error as a function of a simple criteria γ_k that describes the average curvature of the trajectory. The curvature of the trajectory κ_k is determined from the angle between consecutive ground-truth velocity vectors $\dot{\mathbf{r}}_{\text{gt}}$ in local frame at each timestep:

$$\kappa_k = \arcsin \|\dot{\mathbf{r}}_{\text{gt},k} \times \dot{\mathbf{r}}_{\text{gt},k-1}\| / \Delta t \quad [^\circ/\text{s}] \quad (23)$$

A high curvature describes an important change of direction while a zero curvature represents a straight trajectory. γ_k is defined as the average of the curvature pondered by a forgetting factor $\lambda < 1$, so that it reflects both current and past changes of direction with higher emphasis on recent values:

$$\gamma_k = \frac{\sum_{n=0}^N \kappa_{k-n} \lambda^n}{\sum_{n=0}^N \lambda^n}, \quad (24)$$

with N large enough so that $\lambda^N \ll 1$.

Figure 9 shows the median estimation error as a function of the change of direction criteria γ_k occurring during the remote-controlled flight. λ is selected by maximizing the absolute value of the correlation between γ_k and the estimation error, and is set to $\lambda = 0.993$ which corresponds to reducing by half the weight given to past curvature κ_k each second (considering 100 Hz sampling rate). While the aver-

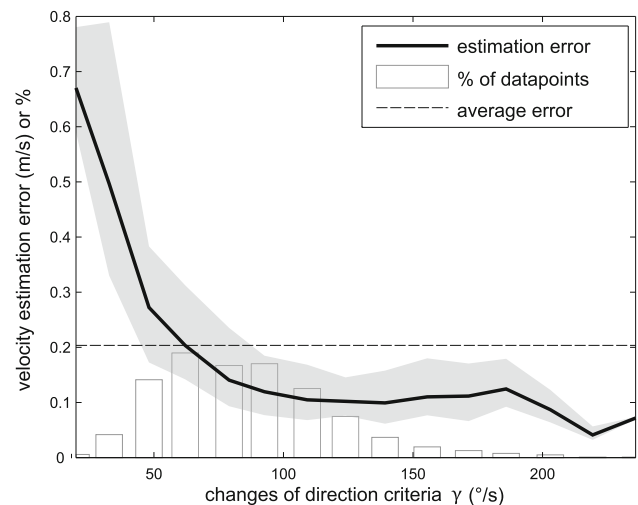


Fig. 9 Velocity estimation error for remote-controlled flight shown in Fig. 8 classified in function of the change of direction criteria γ_k (total number of datapoints: 21,000). The thick line corresponds to the median for each bin and the shaded area shows the lower and upper quartiles

age estimation error over the entire flight is about 0.2 m/s, it can be seen that the error is significantly below average for high γ_k , and significantly higher than the average for low γ_k . These results give us insights on the influence of changes of direction on the performance, as well as the frequency at which they should occur. Figure 9 can be used to determine the average curvature of a trajectory recommended for obtaining optimal results with our particular setup. Typically, it is recommended to achieve at least $\gamma_k > 100^\circ/\text{s}$, after which the estimation error does not improve and remains around 0.1 m/s in average.

4.2 Autonomous flights

The goal of this sub-section is to demonstrate autonomous flight based on the velocity and position estimation obtained from the ego-motion estimation algorithm. We then explore whether changes of direction can artificially be provoked to ensure a good performance of the algorithm (or even improve it), typically by avoiding long straight trajectories during navigation tasks. To answer this question, the performance is characterized in terms of velocity estimation error and position drift during autonomous flights where three motion types are tested. For the first flight, a fixed position setpoint is given as input to the controller without any artificially added motion. For a second and third flight, a trajectory along a circle in the vertical plane and up/down oscillations along a vertical line are programmed respectively. The oscillation frequencies are set to 0.5 Hz, which corresponds to a theoretical average curvature of $180^\circ/\text{s}$ for these two motion types.

The quadrotor successfully flew entirely autonomously during approximately 4 min straight for each motion type,

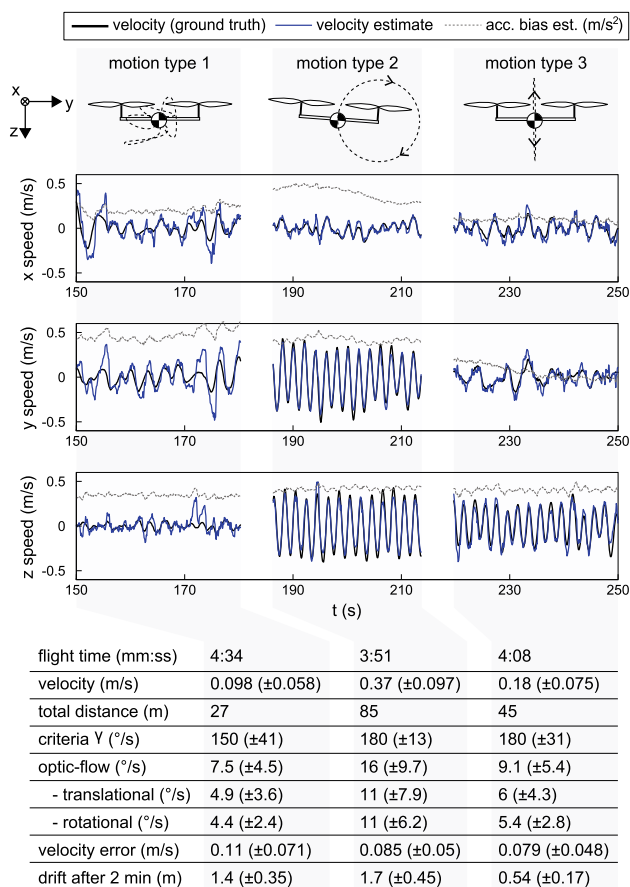


Fig. 10 Excerpts of the velocity estimate for three closed-loop controlled flights during which the robot uses the output of the onboard velocity and position estimation as feedback to fly autonomously according to different motion types. The first flight is stationary, and oscillations along a vertical circle and vertical line are automatically controlled during the 2nd and 3rd flight respectively. The table summarizes some average values obtained during flights and the average performance of the velocity and position estimation (standard deviation is given in parenthesis)

except in the case of motion type 2 during which the remote control was briefly used a few times to bring the robot back in the center of the room when it would drift too close to obstacles. The three autonomous flights can be watched in the online video n^o2. The results of the characterization show that the velocity estimation is reliable in all three cases, with average errors in the range of 0.1 m/s (Fig. 10). Interestingly, the stationary flight trajectory (motion type 1, Fig. 10) is comprised of enough changes of direction (150° /s of curvature in average) to keep the estimation error low (0.11 m/s in average). The average curvature of the trajectory for motion types 2 and 3 correspond to the theoretical prediction (180° /s). The velocity estimation error for motion types 2 and 3 (0.085 and 0.079 m/s in average respectively) is slightly reduced compared to the stationary motion, but the difference is not significant. There is approximately as much rotational and

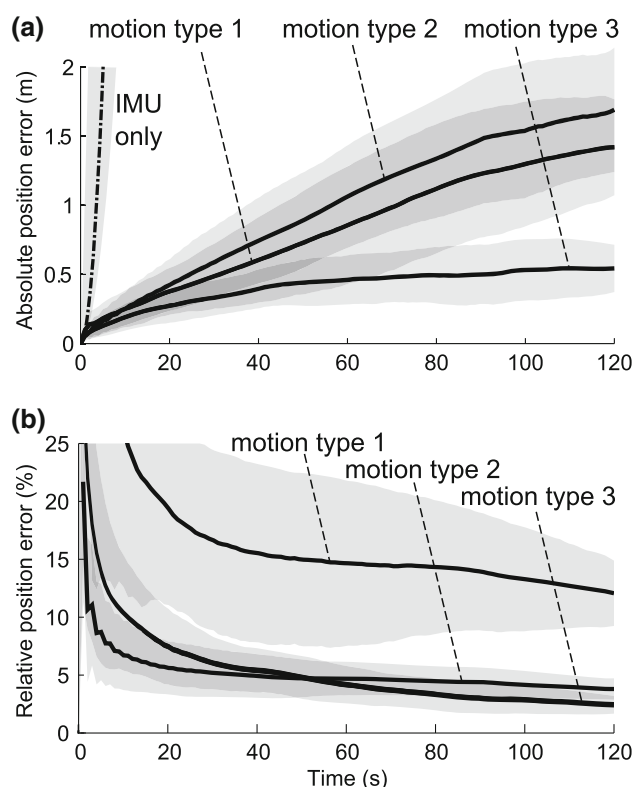


Fig. 11 Position drift obtained by comparing the position estimate to the ground-truth for 3 different flights. The absolute drift is shown in **a** and the drift relative to the distance travelled is shown in **b**. The quadrotor is controlled in closed-loop using the onboard ego-motion estimation and is programmed to perform different types of motion. All flights are about 4 min long, and the average drift was computed from multiple 2 min windows starting every second and comparing the onboard position estimate to the ground-truth position. Note that both lateral and altitude drifts are all included in this drift measurement. The grey shading represents the standard deviation

translational optic-flow measured during flight, which shows the importance of the de-rotation procedure.

The position drift for all motion types remain in average under 2 m after 2 min (Fig. 11a). The shape of the absolute drift curves fit approximately the error expected for random walk integration, with a higher increase at the beginning—typically, in case of gaussian random walk, the error is $\sim\sqrt{t}$ (Rudnick and Gaspari 2004). We note however that the velocity error is not exactly gaussian, which is why we include the position drift in the following analysis. Motion type 3 generates a significantly lower absolute drift after 2 min (0.5 m in average), but is comparable to motion type 2 in terms of error relative to distance traveled (about 3 % in average after 2 min, Fig. 11b). Because of the small distance traveled during motion type 1 (stationary flight), the relative error is very high (12 % after 2 min).

The position error after 2 min remains quite low and these results show the viability of the onboard ego-motion estimation approach for the autonomous control of MAVs. In

addition, these results show that it is possible to generate oscillations in order to ensure sufficient changes of direction. Typically motion types 2 and 3 meet the performance expected from Fig. 9 (about 0.1m/s average error for curvature around $180^\circ/s$). This means that oscillations can be added whenever a navigation task requires trajectories that would not meet the criteria of high average curvature (typically straight trajectories). Also, during stationary flight, the average curvature is $150^\circ/s$ in our case, but this number depends on the reactivity of the controller, and may be hard to reproduce among platforms. Adding oscillations, even around a stationary position, ensures less dependance on the control parameters that are otherwise responsible for the changes of direction. Finally, the results show that it may be possible to find an optimal motion type that would minimize drift, since typically motion type 3 presents significantly lower drift than the others, but this is beyond the scope of this work.

4.3 Optic-flow sensors configuration

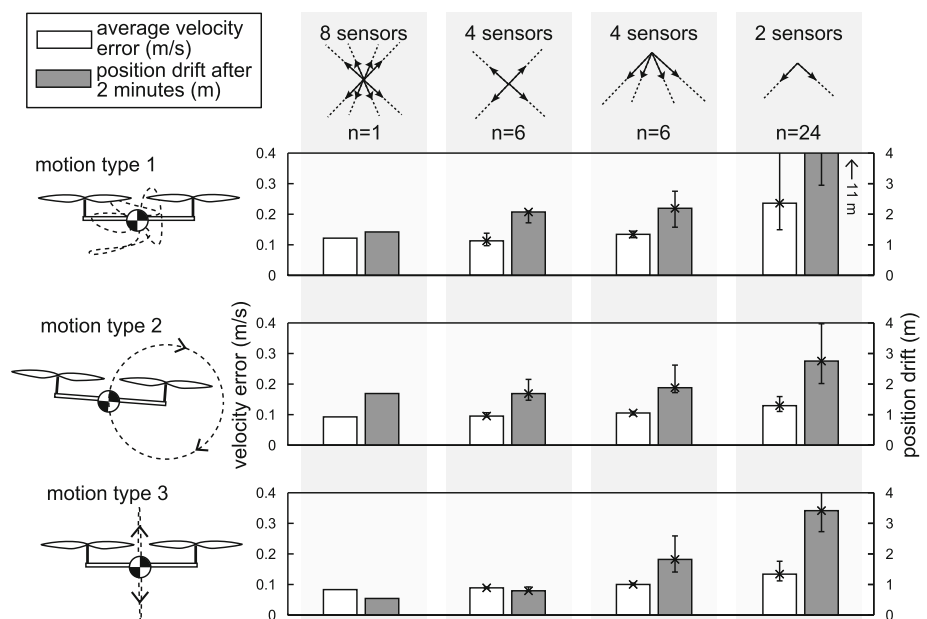
In theory, only two optic-flow sensors are sufficient to generate the direction constraint necessary for the algorithm to converge (Sect. 2). However, in practice, sensors may not always generate translational optic-flow, e.g. if a sensor is aligned with the direction of motion, or if it is pointing towards distant or non-textured objects. It is thus interesting to consider multiple sensors covering a large field in order to obtain redundancy. In order to assess the influence of the number of optic-flow sensors and viewing directions, we test here the performance of the ego-motion estimation in many sensor configurations. We run the ego-motion esti-

mation algorithm off-line multiple times on the 3 flights of Sect. 4.2, enabling or disabling some optic-flow sensors in each run. We compare 4-sensors and 2-sensors configurations to the original results obtained with the initial 8-sensors arrangement shown in Fig. 6.

Twelve 4-sensors configurations are tested, and classified in two types: 6 opposite-facing pairs, and 6 same-face configurations (where the 4 sensors define the corners of a same face in the cube of Fig. 6). The 6 opposite-facing pairs correspond to the largest field of view achievable with 4 sensors, while the other 6 configurations represent the smallest field of view. The comparison of these two 4-sensors configuration types aims at determining the influence of a wide field of view compared to a narrower one. Twenty-four 2-sensors configurations are tested, corresponding to all sensor permutations, except the configurations where two sensors are opposite-facing (or collinear). The configurations with collinear sensor pairs as well as the single sensor configurations are not tested, because they generate velocity estimates with unbounded errors (which is expected because they do not provide sufficient constraints to determine the direction of motion).

The results show that the velocity estimation error with only 4 sensors is not significantly different to the performance with 8 sensors (Fig. 12). However, the position drift is significantly higher for half of the 4-sensors configurations compared to the 8-sensors configuration (motion type 1: 2nd and 3rd column; motion type 3: 3rd column). Finally, the performance with only 2 sensors is significantly worse in all cases, both for velocity and position estimation. These results confirm that the redundancy provided by a number of sensors higher than two is profitable. They show that the con-

Fig. 12 Performance of velocity and position estimation for different sensor configurations. The original 8 sensors configuration is compared to 12 configurations using 4 sensors and then to 24 configurations using 2 sensors only. The bars represent the median values, while the error bars show the upper and lower quartiles



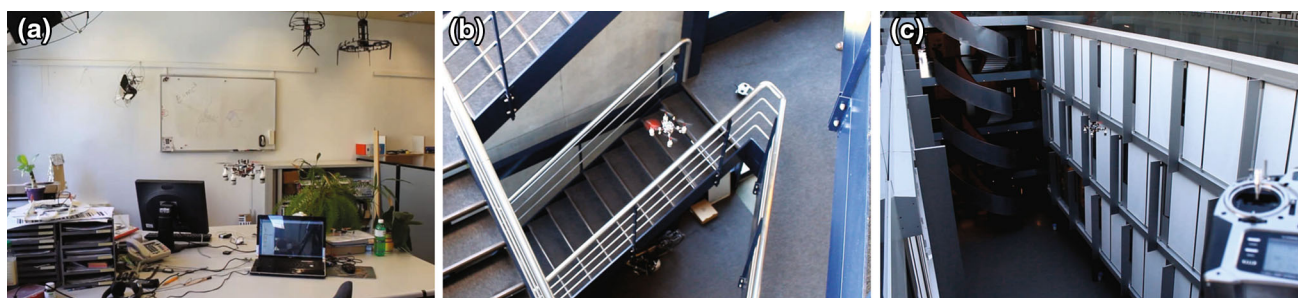


Fig. 13 Various environments in which the optic-flow-based autonomous control of the quadrotor is tested: **a** office, **b** stairwell, **c** large open volume. These flights can be watched in the third accompanying video

figuration with 4 sensors perform as well as with 8 sensors in some cases. Using only four sensors is thus a viable solution for closed-loop control of MAVs, as the drift after 2 min remains around 2 m in average. There is no significant difference between the two different 4-sensors configuration (2nd vs 3rd column). The 8-sensors configuration is still more reliable and provides significantly less position drift for motion types 1 and 3.

Note that the advantages of covering a wide field-of-view may stand out more in situations where the distance distribution to the environment is very inhomogeneous, typically if only a few obstacles are close-by while most other visual cues are distant. Indeed, translational optic-flow signals weaken as distance increases, and so does the accuracy of their direction. Therefore, the performance may be increased significantly for configurations comprised of sensors that see the few close-up objects, compared to the configurations with fewer sensors that only see the distant visual cues (because they have statistically less chances to see the few close-up objects). Ideally, a sensor would cover the full field of view, so that the direction of motion can be extracted with the highest precision from the regions of the field of view that generate the strongest translational optic-flow.

4.4 Flights in various environments

Finally, autonomous flights were performed without the motion capture system in the environments shown in Fig. 13 with the exact same position controller as for stationary flight in Sect. 4.2. The environments in Fig. 13a, b were specifically chosen for their inhomogeneous depth distribution and lack of structure, to illustrate the capability of the algorithm to cope with unstructured environments. Online video n°3 shows that the closed-loop control works well in all of these environments. It can be observed that the quadrotor exhibits a larger position drift in the large open volume shown in Fig. 13c. This is likely due to the larger depth that attenuates the translational optic-flow signal, and thus reduces its accuracy.

5 Conclusion

This paper describes a novel method for ego-motion estimation and its implementation on a miniature quadrotor. Because it uses only the direction of the translational optic-flow and not its scale, this method works in environments of any geometry without relying on depth estimation and without converting visual information into a metric measurement. The results confirm the hypothesis that frequent changes of direction improve the estimation, and thus we suggest to actively control oscillations in order to minimize the position estimation drift. The average position error after 2 min observed in our experiments varied from 50 cm to 1.7 m depending on the motion type, which remains better or as good as the drift to expect when using GPS outdoors during the same time frame. Note that the observability condition of the method proposed in this paper is more demanding than in the case of inertial SLAM, where linear accelerations (e.g. along a straight line) are sufficient to provide observability (Martinelli 2012) and where this condition does not have to be fulfilled continuously. The presented approach is particularly suited to hovering robots, which naturally change direction frequently and can even be actively steered in order to improve their ego-motion estimate, along the principles of embodied cognition. We note that methods advocating for active sensing are not new, and it is well known that visual information gets richer with motion (Aloimonos et al. 1988).

While the proposed method does not reach the accuracy of techniques based on feature tracking, it shows sufficient precision for closed-loop control of MAVs. Many applications may not require absolute positioning and simply require a local stabilization. Also, a slowly drifting position estimate still allows to obtain a short term map of the surroundings or generate trajectories to avoid obstacles. The advantage of our approach is that it can be implemented on simple microcontrollers and requires very light-weight sensors, and can thus be embedded on smaller flying platforms whose size, agility and robustness are beneficial when exploring tight spaces.

A parallel with biology, in particular insect flight, may be interesting to investigate because of the sensor suite similari-

ties between our robot and insects (Krapp and Hengstenberg 1996), and because our findings suggest that changes of direction are key to estimating the absolute scale of translatory motions in unstructured environments. Saccadic or stereotypical frequent changes of direction have been widely observed in insect flights, be it in drosophila (Tammero and Dickinson 2002), blowflies (Schilstra and van Hateren 1999) or bees (Boeddeker and Hemmi 2010), and their exact purpose is still an open question (Kern et al. 2012; van Breugel and Dickinson 2012). While it has been shown that insects use optic-flow amplitude to regulate their speed (Srinivasan et al. 1996; Baird et al. 2005), this work shows that optic-flow direction may be a source of information little explored to date, and that it can provide absolute velocity estimates when coupled with changes of direction. While our robot uses accelerometer, a simple internal model of stereotypical motions would play the same role. Typically, when an insect initiates a lateral saccade of known amplitude, it can infer its absolute speed by comparing the old and new translation direction: if the change of direction is relatively large, it means that its speed is relatively low, and vice-versa.

Future work will focus on optimizing the motion type of the flying robot to obtain just the required amount of changes of direction. Typically, the Kalman filter covariance matrix may be used to steer in real time the robot so as to constantly keep the estimation error low. Measurement of distance to obstacles will be studied too, which can be achieved by using the amplitude of the translational optic-flow. The implementation of RANSAC for outlier rejection will be explored in order to better handle moving objects or spurious measurements. The focus of this chapter was on demonstrating the viability of the new concept for ego-motion estimation and not on implementing an optimal algorithm, which leaves a lot of room for various improvements of the sensor fusion algorithm. For example: a rigorous modeling of the sensor errors, non-linear sensor fusion techniques or the implementation of a filter with both attitude and velocity estimation could be examined. Finally, future miniaturization of the sensor hardware may enable an implementation on even smaller flying robots.

Acknowledgments The authors thank the Parc Scientifique office of Logitech at EPFL for providing the bare mouse chips. The authors also thank Przemyslaw Kornatowski for helping designing and manufacturing the flying platform. We also thank Ramon Pericet-Camara, Felix Schill and Julien Lecoœur for their help. We thank Auke Ijspeert for giving us access to some motion capture equipment. Finally, we thank the anonymous reviewers for their contribution in improving the manuscript. The method described in this paper has been submitted for patenting (European patent filing number EP12191669.6). This research was supported by the Swiss National Science Foundation through the National Centre of Competence in Research (NCCR) Robotics.

References

- Aloimonos, J., Weiss, I., & Bandyopadhyay, A. (1988). Active vision. *International Journal of Computer Vision*, 1(4), 333–356.
- Baird, E., Srinivasan, M. V., Zhang, S., & Cowlings, A. (2005). Visual control of flight speed in honeybees. *The Journal of Experimental Biology*, 208(Pt 20), 3895–3905.
- Barrows, G. L., Humbert, S., Leonard, A., Neely, C. W., & Young, T. (2006). Vision Based Hover in Place. *Patent WO 2011/123758*.
- Beyeler, A., Zufferey, J.-C., & Floreano, D. (2009). Vision-based control of near-obstacle flight. *Autonomous Robots*, 27(3), 201–219.
- Blösch, M., Weiss, S., Scaramuzza, D., & Siegwart, R. (2010). Vision based mav navigation in unknown and unstructured environments. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)* (pp. 21–28), Anchorage, AK.
- Boeddeker, N., & Hemmi, J. M. (2010). Visual gaze control during peering flight manoeuvres in honeybees. *Proceedings of the Royal Society B*, 277(1685), 1209–1217.
- Briod, A., Zufferey, J.-C., & Floreano, D. (2012). Automatically calibrating the viewing direction of optic-flow sensors. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)* (pp. 3956–3961), Saint Paul, MN.
- Briod, A., Zufferey, J.-C., & Floreano, D. (2013). Optic-flow based control of a 46g quadrotor. In *Proceedings of the IEEE/RSJ IROS'13 International Workshop on Vision-based Closed-Loop Control and Navigation of Micro Helicopters in GPS-denied Environments*. Tokyo, Japan.
- Bristeau, P., Callou, F., Vissière, D., & Petit, N. (2011). The navigation and control technology inside the ar. drone micro uav. In *18th IFAC World Congress* (pp. 1477–1484).
- Chen, Y.-S., Liou, L.-G., Hung, Y.-P., & Fuh, C.-S. (2001). Three-dimensional ego-motion estimation from motion fields observed with multiple cameras. *Pattern Recognition*, 34(8), 1573–1583.
- Corke, P., Lobo, J., & Dias, J. (2007). An introduction to inertial and visual sensing. *International Journal of Robotics Research*, 26(6), 519–535.
- Davison, A. (2003). Real-time simultaneous localisation and mapping with a single camera. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 1403–1410).
- Diel, D. D., DeBitetto, P., & Teller, S. (2005). Epipolar constraints for vision-aided inertial navigation. In *IEEE Workshop on Application of Computer Vision*, (pp. 221–228), Breckenridge, CO.
- Dissanayake, G., & Sukkarieh, S. (2001). The aiding of a low-cost strap-down inertial measurement unit using vehicle model constraints for land vehicle applications. *IEEE Transactions on Robotics*, 17(5), 731–747.
- Duhamel, P.-E. J., Perez-Arancibia, C. O., Barrows, G. L., & Wood, R. J. (2013). Biologically Inspired optical-flow sensing for altitude control of flapping-wing microrobots. *IEEE/ASME Transactions on Mechatronics*, 18(2), 556–568.
- Floreano, D., Pericet-Camara, R., Viollet, S., Ruffier, F., Brückner, A., Leitl, R., et al. (2013). Miniature curved artificial compound eyes. *Proceedings of the National Academy of Sciences of the United States of America*, 110(23), 9267–9272.
- Franz, M., Chahl, J. S., & Krapp, H. G. (2004). Insect-inspired estimation of egomotion. *Neural Computation*, 16(11), 2245–2260.
- Fraundorfer, F., & Scaramuzza, D. (2012). Visual odometry: Part II: Matching, robustness, optimization, and applications. *IEEE Robotics and Automation Magazine*, 19(2), 78–90.
- Grewal, M. S., & Andrews, A. P. (2001). *Kalman filtering: Theory and practice using MATLAB* (2nd ed.). New York: Wiley-IEEE Press.
- Hartley, R. I., & Zisserman, A. (2004). *Multiple view geometry in computer vision* (2nd ed.). Cambridge: Cambridge University Press.
- Herisse, B., Russotto, F.-X., Hamel, T., & Mahony, R. (2008). Hovering flight and vertical landing control of a VTOL Unmanned Aerial

- Vehicle using optical flow. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 801–806).
- Honegger, D., Greisen, P., Meier, L., Tanskanen, P., & Pollefeys, M. (2012). Real-time velocity estimation based on optical flow and disparity matching. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 5177–5182), Vilamoura, Portugal.
- Honegger, D., Meier, L., Tanskanen, P., & Pollefeys, M. (2013). An open source and open hardware embedded metric optical flow CMOS camera for indoor and outdoor applications. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Karlsruhe, Germany.
- Jones, E. S., & Soatto, S. (2011). Visual-inertial navigation, mapping and localization: A scalable real-time causal approach. *The International Journal of Robotics Research*, 30(4), 407–430.
- Kelly, J., & Sukhatme, G. S. (2010). Visual-inertial sensor fusion: Localization, mapping and sensor-to-sensor self-calibration. *The International Journal of Robotics Research*, 30(1), 56–79.
- Kendoul, F., Fantoni, I., & Nonami, K. (2009). Optic flow-based vision system for autonomous 3D localization and control of small aerial vehicles. *Robotics and Autonomous Systems*, 57(6–7), 591–602.
- Kern, R., Boeddeker, N., Dittmar, L., & Egelhaaf, M. (2012). Blowfly flight characteristics are shaped by environmental features and controlled by optic flow information. *The Journal of Experimental Biology*, 215(14), 2501–2514.
- Kim, A., & Golnaraghi, M. (2004). A quaternion-based orientation estimation algorithm using an inertial measurement unit. In *Position Location and Navigation Symposium* (pp. 268–272).
- Kim, J., & Brambley, G. (2007). Dual optic-flow integrated navigation for small-scale flying robots. In *Proceedings of Australasian Conference on Robotics and Automation*, Brisbane, Australia.
- Klein, G., & Murray, D. (2007). Parallel tracking and mapping for small AR workspaces. In *Proceedings of the IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR)* (pp. 225–234). IEEE.
- Koenderink, J., & Doorn, A. (1987). Facts on optic flow. *Biological Cybernetics*, 56(4), 247–254.
- Krapp, H. G., & Hengstenberg, R. (1996). Estimation of self-motion by optic flow processing in single visual interneurons. *Nature*, 384(6608), 463–6.
- Kushleyev, A., Mellinger, D., Powers, C., & Kumar, V. (2013). Towards a swarm of agile micro quadrotors. *Autonomous Robots*, 35(4), 287–300.
- Lucas, B. D., & Kanade, T. (1981). An iterative image registration technique with an application to stereo vision. *Proceedings of the Seventh International Joint Conference on Artificial Intelligence*, vol. 130, (pp. 121–130).
- Martinelli, A. (2012). Vision and IMU data fusion: Closed-form solutions for attitude, speed, absolute scale, and bias determination. *IEEE Transactions on Robotics*, 28(1), 44–60.
- Mourikis, A., & Roumeliotis, S. (2007). A multi-state constraint Kalman filter for vision-aided inertial navigation. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)* (pp. 3565–3572), Roma, Italy.
- Nelson, R., & Aloimonos, J. (1988). Finding motion parameters from spherical motion fields (or the advantages of having eyes in the back of your head). *Biological Cybernetics*, 58(4), 261–273.
- Pylvanainen, T. (2008). Automatic and adaptive calibration of 3D field sensors. *Applied Mathematical Modelling*, 32(4), 575–587.
- Rudnick, J., & Gaspari, G. (2004). *Elements of the random walk*. Cambridge: Cambridge University Press.
- Ruffier, F., & Franceschini, N. (2005). Optic flow regulation: The key to aircraft automatic guidance. *Robotics and Autonomous Systems*, 50(4), 177–194.
- Scaramuzza, D., Achtelik, M. C., Doitsidis, L., Fraundorfer, F., Kosmatopoulos, E. B., Martinelli, A., Achtelik, M. W., Chli, M., Chatzichristofis, S. A., Kneip, L., Gurdan, D., Heng, L., Lee, G. H., Lynen, S., Meier, L., Pollefeys, M., Siegwart, R., Stumpf, J. C., Tanskanen, P., Troiani, C., & Weiss, S. (2013). Vision-controlled micro flying robots: From system design to autonomous navigation and mapping in GPS-denied environments. *IEEE Robotics and Automation Magazine*.
- Scaramuzza, D., & Fraundorfer, F. (2011). Visual odometry [Tutorial]. *IEEE Robotics and Automation Magazine*, 18(4), 80–92.
- Schill, F., & Mahony, R. (2011). Estimating ego-motion in panoramic image sequences with inertial measurements. *Robotics Research*, 70, 87–101.
- Schilstra, C., & van Hateren, J. H. (1999). Blowfly flight and optic flow, I. Thorax kinematics and flight dynamics. *Journal of Experimental Biology*, 202, 1481–1490.
- Shen, S., Mulgaonkar, Y., Michael, N., & Kumar, V. (2013). Vision-based state estimation and trajectory control towards high-speed flight with a quadrotor. In *Proceedings of Robotics: Science and Systems (RSS)*, Berlin, Germany.
- Srinivasan, M. V. (1994). An image-interpolation technique for the computation of optic flow and egomotion. *Biological Cybernetics*, 71(5), 401–415.
- Srinivasan, M. V., Thurrowgood, S., & Soccol, D. (2009). From visual guidance in flying insects to autonomous aerial vehicles. In D. Floreano, et al. (Eds.), *Flying insects and robots* (pp. 15–28). Berlin: Springer.
- Srinivasan, M. V., Zhang, S., Lehrer, M., & Collett, T. (1996). Honeybee navigation en route to the goal: Visual flight control and odometry. *The Journal of Experimental Biology*, 199, 237–244.
- Tammero, L. F., & Dickinson, M. H. (2002). The influence of visual landscape on the free flight behavior of the fruit fly *Drosophila melanogaster*. *The Journal of Experimental Biology*, 205(Pt 3), 327–343.
- Taylor, C. N. (2008). Enabling navigation of MAVs through inertial, vision, and air pressure sensor fusion. In *Proceedings of the IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, vol. 35 (pp. 475–480), Seoul.
- Taylor, C. N., Veth, M., Raquet, J., & Miller, M. (2011). Comparison of two image and inertial sensor fusion techniques for navigation in unmapped environments. *IEEE Transactions on Aerospace and Electronic Systems*, 47(2), 946–958.
- Titterton, D. H. (2004). *Strapdown inertial navigation technology* (2nd ed.). London: The Institution of Engineering and Technology.
- Triggs, B., McLauchlan, P., Hartley, R., & Fitzgibbon, A. (2000). Bundle adjustment: A modern synthesis. *Vision algorithms: Theory and practice. Lecture notes in computer science* (Vol. 34099, pp. 298–372). Berlin: Springer.
- van Breugel, F., & Dickinson, M. H. (2012). The visual control of landing and obstacle avoidance in the fruit fly *Drosophila melanogaster*. *The Journal of Experimental Biology*, 215(11), 1783–1798.
- Weiss, S., Achtelik, M., Lynen, S., Chli, M., & Siegwart, R. (2012a). Real-time onboard visual-inertial state estimation and self-calibration of MAVs in unknown environments. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Saint Paul, MN.
- Weiss, S., Achtelik, M. W., Chli, M., & Siegwart, R. (2012b). Versatile distributed pose estimation and sensor self-calibration for an autonomous MAV. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)* (pp. 31–38), Saint Paul, MN.
- Wood, R. J. (2008). The first takeoff of a biologically inspired at-scale robotic insect. *IEEE Transactions on Robotics*, 24(2), 341–347.
- Zufferey, J.-C., & Floreano, D. (2006). Fly-inspired visual steering of an ultralight indoor aircraft. *IEEE Transactions on Robotics*, 22(1), 137–146.



and CTO of Flyability, a spinoff company producing robust flying robots for inspection.

Adrien Briod obtained his MSc in robotics at the Ecole Polytechnique Fédérale de Lausanne (EPFL) and Harvard University in 2009. He then obtained in 2014 his PhD at EPFL (Laboratory of Intelligent Systems) during which he developed and patented novel approaches for the stabilization of flying robots in cluttered environments. He is the author or co-author of more than 10 peer-reviewed publications or patents on indoor flying robotics. Adrien is also founder



official intelligence, and robotics. He published three books on the topics of evolutionary robotics, bioinspired artificial intelligence, and biomimetic flying robots with MIT Press and Springer-Verlag.

Dario Floreano (SM'06) received the M.A. and Ph.D. degrees from the University of Trieste, Trieste, Italy, in 1988 and 1995, respectively, and the M.S. degree from the University of Stirling, Stirling, Scotland, in 1991. He is the Director of the Laboratory of Intelligent Systems, Ecole Polytechnique Fédérale de Lausanne, Lausanne, Switzerland, and the Director of the Swiss National Center of Competence in Robotics. His research interests include the convergence of biology, arti-



Jean-Christophe Zufferey obtained his Ph.D. in flying robotics from EPFL in 2005 for which he received the ABB best Ph.D. award. He is author or co-author of more than 60 peer-reviewed publications, among which a book on Bio-inspired Flying Robots (2008) by the EPFL/CRC Press. Jean-Christophe is also founder and CEO of senseFly, a spin-off company producing smart flying robots for mapping.