Optic-Flow Based Control of a 46g Quadrotor

Conference Paper · November 2013

CITATIONS
42

READS
577

3 authors, including:

Adrien Briod
Flyability SA
15 PUBLICATIONS **555** CITATIONS

SEE PROFILE

Optic-Flow Based Control of a 46g Quadrotor

Adrien Briod[1], Jean-Christophe Zufferey[2] and Dario Floreano[1]

***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 allowing such platforms to fly autonomously. 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. Unlike many vision-based methods, this algorithm does not rely on feature tracking, structure estimation, additional distance sensors or assumptions about the environment. Key to this method is the introduction of the *translational optic-flow direction constraint* (TOFDC), which does not use the optic-flow scale, but only its direction 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 geometries. We demonstrate the implementation of this algorithm on a miniature 46g quadrotor for closed-loop position control.**

I. Introduction

Using robots instead of risking human lives for exploration missions in dangerous environments has great safety and efficiency benefits. Flying robots have many advantages as they provide an elevated view-point and can navigate above rubble more efficiently than ground-based robots. However, most flying platforms are unstable by design and need to be controlled in closed-loop in order to achieve useful tasks, such as remaining in a stable position in the air or following 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. Payload being limited on flying platforms, it is of great interest to miniaturize the hardware required for ego-motion estimation so as to reduce the size of autonomous Micro Aerial Vehicles (MAV).

Because of their versatility and relatively low weight, monocular vision sensors are often used for ego-motion estimation. Monocular vision sensors are affected by the scale ambiguity, in other words the inability to distinguish the scale of a translation [1]. Visual information is usually converted to metric measurements thanks to the so-called 'scale factor', which is often identified thanks to additional sensors. One of the most successful solution is inertial SLAM (Simultaneous Localization and Mapping) [2], [3], [4], [5], [6]. In this case, the scale factor is eventually obtained thanks to accelerometers and is observable if linear accelerations are present [7]. These methods allow for absolute positioning and even reconstruct the 3D structure of the environment in the process.

Authors are with: (1) the Laboratory of Intelligent Systems (LIS) http://lis.epfl.ch, Ecole Polytechnique Fédérale de Lausanne (EPFL), 1015 Lausanne, Switzerland, (2) SenseFly Ltd, 1024 Ecublens, Switzerland

This research was supported by the Swiss National Science Foundation through the National Centre of Competence in Research (NCCR) Robotics.

Contact e-mail: adrien.briod@epfl.ch.

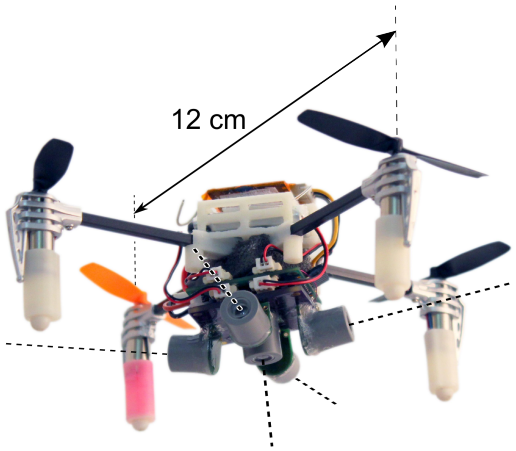

Fig. 1. A 46g quadrotor capable of autonomous flight thanks to inertial sensors and five 0.8g optic-flow sensors, each providing a 2D optic-flow measurement, and arranged in order to cover a wide field-of-view (dashed lines are the viewing directions).

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 [8], [9], [10], [11]. Applying directly this constraint bypasses the need for structure re-construction and thus reduces significantly the amount of processing and memory required [9], while it can still take advantage of error corrections from features tracked multiple frames apart [8], [10]. However, the scale of the motion is not provided by epipolar constraint updates, and the relative scale is not conserved, which provokes generally large estimation errors as mentioned in [9]. An interesting approach in [11] suggests to estimate the average depth of the tracked features, so that the camera can be used as a metric sensor.

However, methods based on feature tracking remain relatively complex because they require cameras with good enough resolution and sufficient computing power for the feature extraction and matching algorithms [12]. On the other hand, comparatively very simple 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 [13], [14]. Optic-flow sensors work at low resolution and thus exist in very small and cheap packages [15], [16]. 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 [15]. Optic-flow has been used on MAVs in the past for bio-inspired obstacle avoidance [17], [18], or speed regulation [19], [20].

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 motion and generated by unidentified visual cues, which prevents it to be used for absolute localization like feature tracking-based algorithms. 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. It is thus comparatively harder to convert optic-flow information into a metric value. For these reasons, optic-flow is generally considered not suitable for visual odometry applications [1].

It is however possible to estimate the direction of motion from optic-flow measurements [21] and even a velocity that is scaled inversely proportionally to the average depth of the environment, which is used in [20] to control a MAV hovering at a fixed point. However it is unclear how this strategy can handle maneuvers provoking the depth to change constantly. Inertial sensors are used in [22] and [23] to obtain the scale of optic-flow measurements, but the methods only work over flat surfaces. 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 [24] or [25], 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 below 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 to determine the scale of the visual information. Feature tracking keeps the scale factor constant, which facilitates its estimation because it has to be obtained once and for all [6] (possibly with small adjustments over time). 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 optic-flow and inertial sensors where it is not attempted to convert optic-flow information into a metric measurement. We then demonstrate its viability and miniaturization potential by using it for closed-loop control of a miniature hovering robot. Section II describes an EKF implementation of the algorithm, section III presents a 46g quadrotor equipped with 5 optic-flow sensors and able of running the algorithm in real time on a microcontroller. Finally, section IV describes briefly the results obtained during autonomous flights in different environments.

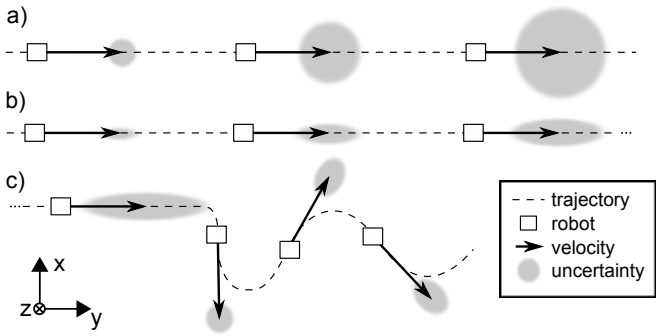

Fig. 2. Conceptual 2D drawing illustrating how the velocity of a moving robot can be observed by using a translational optic-flow direction constraint (TOFDC) to aid inertial navigation. An optic-flow sensor pointing towards the Z direction provides a 2D measurement vector (of unknown scale) defining 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.

II. Approach 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 thanks to 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.

We suggest to use the optic-flow sensors to aid an inertial navigation process, which is the temporal integration of the linear accelerations. 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 TOFDC (translational optic-flow direction constraint), and that we use to correct for inertial navigation velocity drift. Contrarily to the epipolar constraint, the TOFDC is not based on the features observed by a camera, but only on the instantaneous motion of the scene, and also never uses measurements made multiple frames apart.

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 [26]. 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 in [21], keeping only one degree of freedom unobserved and subject to drift. The key is that the unobserved degree of

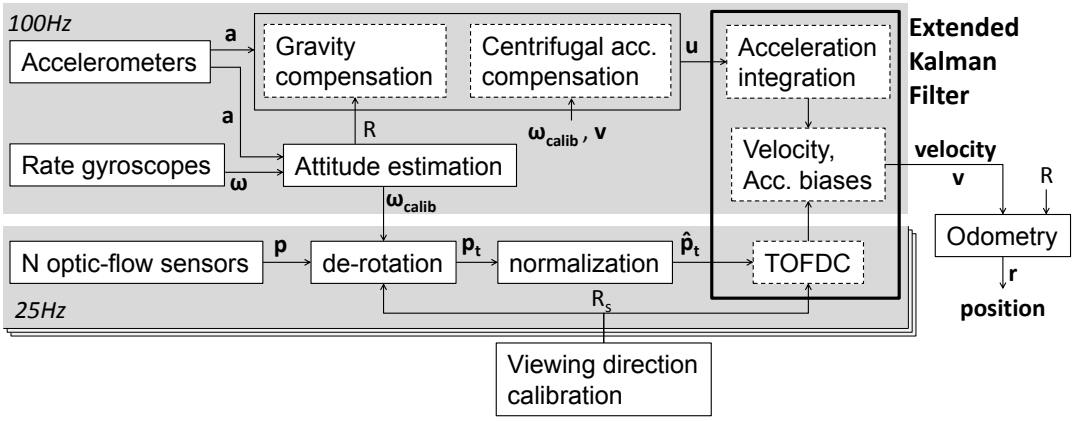

Fig. 3. Block diagram showing the proposed algorithm for fusion of inertial sensors and multiple optic-flow sensors.

freedom is along the direction of motion, and thus it changes if the direction of motion changes. Therefore the drift along all directions can be mitigated and all components of the velocity estimate 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.

Fig. 2 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, which is due to the drift in velocity estimation of any non-aided inertial navigation system [26]. Then, a direction constraint is applied in situation b), which reduces the uncertainty except along the direction of motion. Once changes of direction are executed in c), the drift accumulated along the previous direction can be corrected by the direction constraint. Finally, the uncertainty on the velocity estimate remains bounded as long as changes of direction are performed. This observability condition is more demanding than in the case of inertial SLAM, where linear accelerations (e.g. along a straight line) are sufficient to provide observability [7] and where this condition does not have to be fulfilled continuously.

This 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.

A. Algorithm implementation

In order to practically implement this idea, we suggest to use an Extended Kalman Filter (EKF) [27] to aid an inertial navigation process with optic-flow sensors, each providing a TOFDC. The state vector $\mathbf{x}$ comprises 6 elements, the velocity in body coordinates $\mathbf{v} = (\mathbf{v_x}, \mathbf{v_y}, \mathbf{v_z})$ and accelerometer biases $\mathbf{b} = (\mathbf{b_x}, \mathbf{b_y}, \mathbf{b_z})$:

$$\mathbf{x} = \begin{bmatrix} \mathbf{v} \\ \mathbf{b} \end{bmatrix} \tag{1}$$

The prediction step of the EKF estimates a velocity by integrating linear accelerations, which are obtained after gravity and the centrifugal acceleration are removed from the accelerometer measurements. 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. The estimation of $\mathbf{b}$ 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. A position estimate is obtained by odometry at a later stage, using the output of the EKF. Fig. 3 describes each component of the ego-motion estimation algorithm. Note that if not explicitly mentioned otherwise, vectors are expressed in body frame.

1) Inertial navigation: In the prediction step of the EKF, we use inertial navigation to obtain an initial velocity estimate by means of linear accelerations integration. Linear accelerations $\mathbf{u}$ in body frame can be obtained by removing the gravity $\mathbf{g}$ and the centrifugal accelerations from the measured accelerations $\mathbf{a}$ thanks to the following equation [26]:

$$\mathbf{u} = \mathbf{a} - \mathbf{R^{-1}} \cdot \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix} - \boldsymbol{\omega} \times \mathbf{v} \tag{2}$$

where $\mathbf{R}$ is the rotation matrix describing the attitude of the robot, $\boldsymbol{\omega}$ is the angular velocity and $\mathbf{v}$ is the velocity estimate in the body frame. The attitude of the robot $\mathbf{R}$ is estimated thanks to a standard quaternion-based filter that also estimates gyroscope biases and thus provides a calibrated angular rate value $\boldsymbol{\omega_{calib}}$.

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

$$\tilde{\mathbf{x}}_k = \mathbf{x}_{k-1} + \begin{bmatrix} (\mathbf{u} + \mathbf{b}) \cdot \boldsymbol{\Delta t} \\ \mathbf{0_{3x1}} \end{bmatrix} \tag{3}$$

$$\tilde{\mathbf{P}}_k = \Phi_k \mathbf{P}_{k-1} \Phi_k^{\mathrm{T}} + \mathbf{Q}_k \tag{4}$$

where Δt is the integration period, $\tilde{\mathbf{P}}_k$ the covariance matrix, Φ_k is the state transition matrix and $\mathbf{Q}_k$ is the prediction noise matrix that allows to tune how much weight is given to the prediction and how fast the biases adapt.

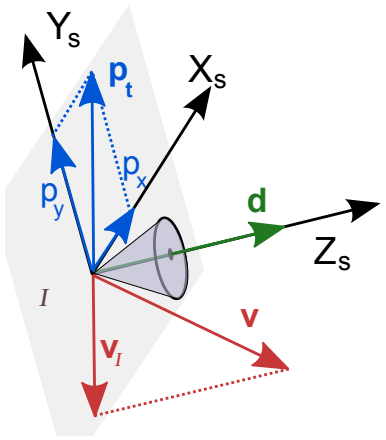

Fig. 4. The direction of the translational optic-flow vector $\mathbf{p_t}$ depends on the velocity $\mathbf{v}$ and the unit vector $\mathbf{d}$, pointing toward the viewing direction of the sensor. The translational optic-flow direction constraint (TOFDC) is expressed in the image plane I, and states 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.

2) Optic-flow correction: 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 [28]:

$$\mathbf{p} = \underbrace{-\boldsymbol{\omega} \times \mathbf{d}}_{\mathbf{p_r}} \underbrace{-\frac{\mathbf{v} - (\mathbf{v} \cdot \mathbf{d})\mathbf{d}}{D}}_{\mathbf{p_t}} \tag{5}$$

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 tranlsation-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 $\mathbf{p}$ thanks to a process called de-rotation [29]. We use a method that we proposed in [30] able to automatically calibrate the viewing direction of optic-flow sensors and execute the de-rotation thanks to 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 (X_s, Y_s, Z_s) whose Z_s axis is aligned with the viewing direction and whose X_s and Y_s axes define the image plane of the optic-flow sensor, as shown in Fig. 4. We can thus express the following vectors:

$$\mathbf{p_{t,s}} = \begin{bmatrix} p_{t,x} \\ p_{t,y} \\ 0 \end{bmatrix}, \mathbf{d_s} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \text{ and } \mathbf{v_s} = \mathbf{R_s v_r} = \begin{bmatrix} v_{s,x} \\ v_{s,y} \\ v_{s,z} \end{bmatrix} \tag{6}$$

Where $\mathbf{v_s}$ is the velocity vector expressed in the sensor frame and $\mathbf{R}_s$ 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:

$$\begin{bmatrix} p_{t,x} \\ p_{t,y} \\ 0 \end{bmatrix} = -\frac{1}{D} \begin{bmatrix} v_{s,x} \\ v_{s,y} \\ 0 \end{bmatrix} \tag{7}$$

which is a relation between 2D vectors:

$$\mathbf{v}_I = -D \cdot \mathbf{p_{t,I}} \tag{8}$$

where $\mathbf{p_{t,I}} = [p_{t,x}, p_{t,y}]$ is the 2D translational optic-flow measurement and

$$\mathbf{v}_I = \begin{bmatrix} v_{s,x} \\ v_{s,y} \end{bmatrix} = \begin{bmatrix} r_{s,11} & r_{s,12} & r_{s,13} \\ r_{s,21} & r_{s,22} & r_{s,23} \end{bmatrix} \mathbf{v} \tag{9}$$

is the projection of the velocity on the image plane. $r_{s,ij}$ are the elements of the first two rows of the rotation matrix $\mathbf{R}_s$, which are obtained thanks to an initial calibration process, for instance as described in [30].

Equation (8) 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 equation (8) 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 :

$$\hat{\mathbf{p}}_{\mathbf{t},I} = -\hat{\mathbf{v}}_I \tag{10}$$

where $\hat{\mathbf{p}}_{\mathbf{t},I}$ and $\hat{\mathbf{v}}_I$ are unit vectors:

$$\hat{\mathbf{p}}_{\mathbf{t},I} = \frac{\mathbf{p_{t,I}}}{||\mathbf{p_{t,I}}||} \quad \text{and} \quad \hat{\mathbf{v}}_I = \frac{\mathbf{v}_I}{||\mathbf{v}_I||} \tag{11}$$

Considering the following Kalman update equations:

$$\mathbf{K}_k = \tilde{\mathbf{P}}_k \mathbf{H}_k^{\mathrm{T}} (\mathbf{H}_k \tilde{\mathbf{P}}_k \mathbf{H}_k^{\mathrm{T}} + \mathbf{R}_k)^{-1} \tag{12}$$

$$\mathbf{x}_k = \tilde{\mathbf{x}}_k + \mathbf{K}_k (\mathbf{z}_k - h[\tilde{\mathbf{x}}_k]) \tag{13}$$

$$\mathbf{P}_k = (I - \mathbf{K}_k \mathbf{H}_k) \tilde{\mathbf{P}}_k \tag{14}$$

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

$$\mathbf{z}_k = \hat{\mathbf{p}}_{\mathbf{t},I} \tag{15}$$

and the following non-linear measurement model:

$$h[\tilde{\mathbf{x}}_k] = -\hat{\mathbf{v}}_I \tag{16}$$

The corresponding jacobian matrix is obtained from $\mathbf{H}_k = \frac{\partial h[x]}{\partial x}$. The 2×2 measurement noise matrix is set to $\mathbf{R}_k = diag(\sigma_{of}^2)$ where σ_{of} describes the noise on the optic-flow measurement and can typically be varied in function of some known quality indicators (such as amount of rotations, or optic-flow norm).

3) Odometry: As a last step, the velocity obtained in body frame $\mathbf{v}$ needs to be converted in earth frame and integrated over time to obtain a position estimate $\mathbf{r}$:

$$\mathbf{r_k} = \mathbf{r_{k-1}} + \mathbf{R^{-1}} \cdot \mathbf{v} \cdot \mathbf{\Delta t} \tag{17}$$

TABLE I

WEIGHT DISTRIBUTION FOR THE QUADROTOR SHOWN IN FIG. 1

	Weight
custom control board	6g
receiver	1.5 g
sensor board	7.7 g
motors and propellers	4x 3.7g
battery	7 g
mechanical parts	9 g
Total	**46 g**

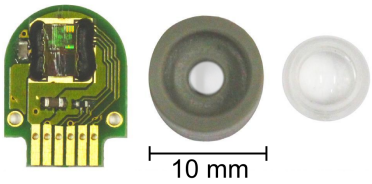

Fig. 5. Disassembled custom 0.8g optic-flow sensor, comprising a bare ADNS-9500 chip which is bonded to a custom printed circuit board (left). A lens holder (middle) is glued on top of the PCB and hosts a CAX100 lens with 10mm focal length (right).

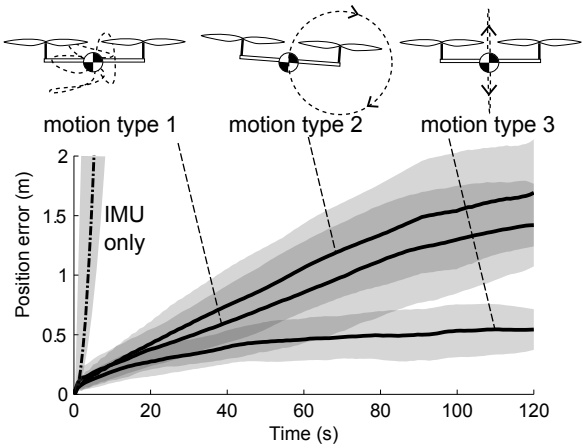

Fig. 6. Average position drift over time obtained by comparing the position estimate to the ground-truth for 3 different flights. The quadrotor is controlled in closed-loop using the onboard ego-motion estimation and is programmed to perform different types of motion. All flights are about 4 minutes long, and the average drift was computed from multiple 2 minutes windows starting every seconds (the grey shading represents the standard deviation). The average drift obtained during the same flights is shown for comparison when only the IMU without correction is used (drift reaches 2m in 5s in average). Note that the data is not obtained with the quadrotor shown in Fig. 1, but a larger one comprising 8 sensors.

III. QUADROTOR

The small quadrotor shown in Fig. 1 was developed in order to test the viability of the approach presented in section II for autonomous closed-loop control. The platform is made of carbon fiber and 3D-printed parts and comprises four DC brushed motors powered by a 260mAh 1-cell LiPo battery. A custom control board integrates a STM32 microcontroller, a ST LIS3DH 3-axis accelerometer and a ST L3G4200 3-axis rate gyro. The biases and factors of the accelerometers are calibrated thanks to the technique described in [31] and stored in memory. A sensor board comprising five 0.8g optic-flow sensors is connected to the control board. The viewing direction of the sensors is chosen so as to cover a wide field-of-view, which is generally preferred for optic-flow based ego-motion estimation [21] and increases robustness in environments with irregular visual conditions or dynamic objects. A piece of foam isolates the sensors from the vibrations exerted on the main frame by the motors. The weight distribution of the robot is presented in Table I.

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 30x30 pixel wide and is sampled at up to 11,750 fps. The sensors are fitted with a custom lens holder comprising a CAX100 aspheric lens with a 10mm focal length providing a 10° 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.

The optic-flow sensors are sampled at 25Hz, the inertial sensors are sampled at 500Hz and the STM32 runs both the attitude estimation filter and the velocity estimation EKF at 100Hz. Three low-level PID controllers stabilize the roll, pitch and yaw angles of the robot at 500Hz, while three high-level PIDs control the lateral position and altitude of the robot at 100Hz 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.

IV. RESULTS

A proof-of-concept experiment is performed where the onboard ego-motion estimation is used to control in closed-loop the miniature quadrotor presented in section III. In order to ensure a proper observability of the ego-motion (as described in section II), changes of direction are provoked by controlling a motion comprising vertical 0.5Hz oscillations around a fixed position. The accompanying video[1] shows two different flights carried out in unstructured environments in order to show the performance of the method even when the depths to the obstacles seen by the five optic-flow sensors vary constantly. The first flight is performed outdoors near vegetation (1 minute), and the other one indoors in a cluttered office (1.5 minutes). While the quadrotor is not keeping a constant position because of the artificially provoked oscillations, the overall position drift of the quadrotor remains relatively small, even for the altitude which is often controlled thanks to a height sensor in conventional approaches. Nonetheless, the position of the quadrotor had to be adjusted twice during the indoor experiment when the drift was bringing it close to obstacles.

[1] https://documents.epfl.ch/groups/l/li/lis-unit/www/Research/egomotion_46g_quadrotor.mp4

Further experiments will focus on characterizing the performance of the method for multiple motion types thanks to a ground truth. For this, a similar but bigger quadrotor equipped with motion capture beacons and 8 optic-flow sensors was built. Preliminary results show that the position drift is of 50cm in average after 2 minutes of flight when 0.5Hz vertical oscillations are controlled (figure 6). On the other hand, the position drift is higher for other motion types, such as stationary flight or 0.5Hz oscillations along a vertical circle.

V. Conclusion

This paper describes a novel method for ego-motion estimation and its implementation on a miniature quadrotor. By using the translational optic-flow direction only, this method works in environments of any geometry without relying on depth estimation and without converting visual information into a metric measurement.

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. 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.

Future work will focus on reactive obstacle avoidance, 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. Finally, future miniaturization may enable an implementation on even smaller flying robots, such as micromechanical flying insects, whose biological counterparts coincidentally also tend to never fly straight.

VI. Acknowledgment

The authors would like to thank the Parc Scientifique office of Logitech at EPFL for providing the bare mouse chips. The authors also want to thank Przemyslaw Kornatowski for helping designing and manufacturing the flying platform. Part of this work has been submitted for patenting (European patent filing number EP12191669.6).

References

[1] D. Scaramuzza and F. Fraundorfer, "Visual Odometry [Tutorial]," *IEEE Robot. Autom. Mag.*, vol. 18, no. 4, pp. 80–92, Dec. 2011.

[2] P. Corke, J. Lobo, and J. Dias, "An Introduction to Inertial and Visual Sensing," *Int. J. Rob. Res.*, vol. 26, no. 6, pp. 519–535, Jun. 2007.

[3] J. Kelly and G. S. Sukhatme, "Visual-Inertial Sensor Fusion: Localization, Mapping and Sensor-to-Sensor Self-calibration," *Int. J. Rob. Res.*, vol. 30, no. 1, pp. 56–79, Nov. 2010.

[4] E. S. Jones and S. Soatto, "Visual-inertial navigation, mapping and localization: A scalable real-time causal approach," *Int. J. Rob. Res.*, vol. 30, no. 4, pp. 407–430, Jan. 2011.

[5] D. Scaramuzza *et al.*, "Vision-Controlled Micro Flying Robots : from System Design to Autonomous Navigation and Mapping in GPS-denied Environments," *IEEE Robot. Autom. Mag.*, 2013.

[6] S. Weiss, M. W. Achtelik, M. Chli, and R. Siegwart, "Versatile Distributed Pose Estimation and Sensor Self-Calibration for an Autonomous MAV," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, Saint Paul, MN, 2012, pp. 31 – 38.

[7] A. Martinelli, "Vision and IMU Data Fusion: Closed-Form Solutions for Attitude, Speed, Absolute Scale, and Bias Determination," *IEEE Trans. Robot.*, vol. 28, no. 1, pp. 44–60, Feb. 2012.

[8] D. D. Diel, P. DeBitetto, and S. Teller, "Epipolar Constraints for Vision-Aided Inertial Navigation," in *IEEE Work. Appl. Comput. Vis.*, Breckenridge, CO, Jan. 2005, pp. 221–228.

[9] C. N. Taylor, M. Veth, J. Raquet, and M. Miller, "Comparison of Two Image and Inertial Sensor Fusion Techniques for Navigation in Unmapped Environments," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 47, no. 2, pp. 946–958, Apr. 2011.

[10] A. Mourikis and S. Roumeliotis, "A multi-state constraint Kalman filter for vision-aided inertial navigation," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, Roma, Italy, 2007, pp. 3565–3572.

[11] S. Weiss, M. Achtelik, S. Lynen, M. Chli, and R. Siegwart, "Real-time Onboard Visual-Inertial State Estimation and Self-Calibration of MAVs in Unknown Environments," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, Saint Paul, MN, 2012.

[12] F. Fraundorfer and D. Scaramuzza, "Visual Odometry : Part II: Matching, Robustness, Optimization, and Applications," *IEEE Robot. Autom. Mag.*, vol. 19, no. 2, pp. 78 – 90, Jun. 2012.

[13] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," *Proc. Seventh Int. Jt. Conf. Artif. Intell.*, vol. 130, pp. 121–130, 1981.

[14] M. V. Srinivasan, "An image-interpolation technique for the computation of optic flow and egomotion," *Biol. Cybern.*, vol. 71, no. 5, pp. 401–415, Sep. 1994.

[15] D. Floreano *et al.*, "Miniature curved artificial compound eyes," *Proc. Natl. Acad. Sci. U. S. A.*, vol. 110, no. 23, pp. 9267–72, Jun. 2013.

[16] P.-E. J. Duhamel, C. O. Perez-Arancibia, G. L. Barrows, and R. J. Wood, "Biologically Inspired Optical-Flow Sensing for Altitude Control of Flapping-Wing Microrobots," *IEEE/ASME Trans. Mechatronics*, vol. 18, no. 2, pp. 556–568, Apr. 2013.

[17] J.-C. Zufferey and D. Floreano, "Fly-inspired visual steering of an ultralight indoor aircraft," *IEEE Trans. Robot.*, vol. 22, no. 1, pp. 137–146, Feb. 2006.

[18] A. Beyeler, J.-C. Zufferey, and D. Floreano, "Vision-based control of near-obstacle flight," *Auton. Robots*, vol. 27, no. 3, pp. 201–219, 2009.

[19] M. V. Srinivasan, S. Zhang, M. Lehrer, and T. Collett, "Honeybee navigation en route to the goal: visual flight control and odometry," *J. Exp. Biol.*, vol. 199, pp. 237–244, Jan. 1996.

[20] G. L. Barrows, S. Humbert, A. Leonard, C. W. Neely, and T. Young, "Vision Based Hover in Place," *Pat. WO 2011/123758*, 2006.

[21] F. Schill and R. Mahony, "Estimating ego-motion in panoramic image sequences with inertial measurements," *Robot. Res.*, vol. 70, pp. 87–101, 2011.

[22] F. Kendoul, I. Fantoni, and K. Nonami, "Optic flow-based vision system for autonomous 3D localization and control of small aerial vehicles," *Rob. Auton. Syst.*, vol. 57, no. 6-7, pp. 591–602, 2009.

[23] B. Herisse, F.-X. Russotto, T. Hamel, and R. Mahony, "Hovering flight and vertical landing control of a VTOL Unmanned Aerial Vehicle using optical flow," in *Proc. IEEE Int. Conf. Intelligent Robots and Systems (IROS)*, Sep. 2008, pp. 801–806.

[24] P. Bristeau, F. Callou, D. Vissière, and N. Petit, "The navigation and control technology inside the ar. drone micro uav," in *18th IFAC World Congr.*, 2011, pp. 1477–1484.

[25] 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 *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, Karlsruhe, Germany, 2013.

[26] D. H. Titterton, *Strapdown inertial navigation technology. - 2nd ed.* The Institution of Engineering and Technology, 2004.

[27] M. S. Grewal and A. P. Andrews, *Kalman filtering: theory and practice using MATLAB*, 2nd ed. Wiley-IEEE Press, Jan. 2001.

[28] J. Koenderink and A. Doorn, "Facts on optic flow," *Biol. Cybern.*, vol. 56, no. 4, pp. 247–254, 1987.

[29] M. V. Srinivasan, S. Thurrowgood, and D. Soccol, "From Visual Guidance in Flying Insects to Autonomous Aerial Vehicles," in *Fly. Insects Robot.* Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 15–28.

[30] A. Briod, J.-c. Zufferey, and D. Floreano, "Automatically calibrating the viewing direction of optic-flow sensors," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, Saint Paul, MN, 2012.

[31] T. Pylvanainen, "Automatic and adaptive calibration of 3D field sensors," *Appl. Math. Model.*, vol. 32, no. 4, pp. 575–587, Apr. 2008.