Vision-based control of near-obstacle flight

Antoine Beyeler · Jean-Christophe Zufferey · Dario Floreano

Abstract This paper presents a novel control strategy, which we call *optiPilot*, for autonomous flight in the vicinity of obstacles. Most existing autopilots rely on a complete 6-degree-of-freedom state estimation using a GPS and an Inertial Measurement Unit (IMU) and are unable to detect and avoid obstacles. This is a limitation for missions such as surveillance and environment monitoring that may require near-obstacle flight in urban areas or mountainous environments. *OptiPilot* instead uses optic flow to estimate proximity of obstacles and avoid them.

Our approach takes advantage of the fact that, for most platforms in translational flight (as opposed to near-hover flight), the translatory motion is essentially aligned with the aircraft main axis. This property allows us to directly interpret optic flow measurements as proximity indications. We take inspiration from neural and behavioural strategies of flying insects to propose a simple mapping of optic flow measurements into control signals that requires only a lightweight and power-efficient sensor suite and minimal processing power.

In this paper, we first describe results obtained in simulation before presenting the implementation of *optiPilot* on a real flying platform equipped only with lightweight and inexpensive optic computer mouse sensors, MEMS rate gyroscopes and a pressure-based airspeed sensor. We show that the proposed control strategy not only allows collision-free flight in the vicinity of obstacles, but is also able to stabilise both attitude and altitude over flat terrain. These results shed new light on flight control by suggesting that the complex sensors and processing required for 6 degree-of-freedom state estimation may not be necessary for autonomous flight and pave the way toward the integration of autonomy into current and upcoming gram-scale flying platforms.

Keywords vision-based control · optic-flow-based control · obstacle avoidance · near-obstacle flight · autonomous unmanned aerial vehicle (UAV) · micro-air vehicle (MAV)

1 Introduction

Unmanned Aerial Vehicles (UAVs) are increasingly used for environmental and security missions (Valavanis 2007) and only legal issues are currently limiting their potential use in many civilian applications. Current autopilots (Procerus Technologies® Kestrel™, MicroPilot® MP Series, Beard et al 2005, *e.g.*) rely on a complete estimation of the 6-degree-of-freedom state (translational and angular position) using a sensor suite that comprises a GPS receiver and an Inertial Measurement Unit (IMU) in order to maintain trajectory and stability of the UAV in obstacle-free space. This prevents the use of UAVs at low altitude in urban environments, mountain regions and forests, which require lightweight vehicles that are capable of continuously steering among obstacles without relying on GPS signal. This paper presents *optiPilot*, a novel control strategy for near-obstacle flight that uses optic flow to detect proximity of obstacles and does not require explicit estimation of translational nor angular position of the aircraft. It consists of directly mapping optic flow estimates into control signals for roll and pitch control using two weighted sums, similar to the neural matched filters of flying insects (Wehner 1987, Krapp et al 1998, Egelhaaf and Kern 2002, Karmeier et al 2006). We show, by means of experiments in simulation, that the

This work was supported by the Swiss National Science Foundation (grant 200020-116149). Part of this work has been submitted for patenting (application PCT/IB2008/051497).

Ecole Polytechnique Fédérale de Lausanne (EPFL)
Laboratory of Intelligent Systems (LIS)
Station 11
CH-1015 Lausanne, Switzerland
E-mail: antoine.beyeler@a3.epfl.ch

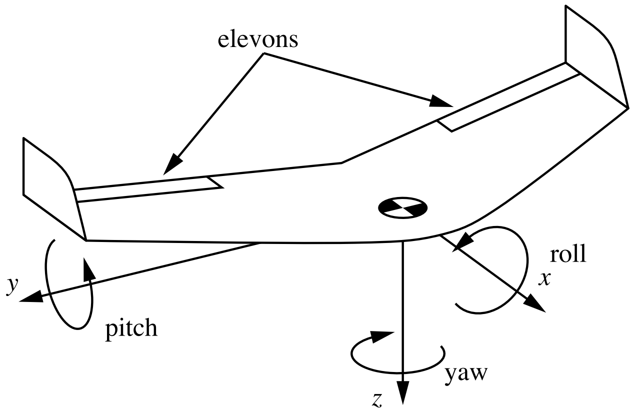

Fig. 1 Coordinate system of the aircraft reference frame, with the names of the three rotation directions. On typical aircraft, roll is controlled using the ailerons, and pitch using the elevator. On flying wings such as the one displayed here, roll and pitch rotations are controlled by the differential and, respectively, common mode of actuation of the two control surfaces called elevons. These two modes of actuation are functionally identical to the ailerons and elevator. Yaw is usually either passively stabilised using fixed vertical surfaces or controlled using a rudder. In normal flight, passive or active yaw regulation is used to produce so-called coordinated turns (Stevens and Lewis 2003). Only the ailerons and elevator, or the elevons, are used to actually steer the aircraft. Note that steering helicopters in translational flight is very similar.

method is capable of avoiding obstacles that may be encountered. We also demonstrate that *optiPilot*, while primarily designed for obstacle avoidance, is also capable of regulating both the attitude angles and the altitude of the aircraft, suggesting that the 6-degree-of-freedom state estimation usually performed for this task is not necessarily required. As an initial set of validation experiments on a real aircraft, we demonstrate flight stability with a small flying wing platform equipped only with lightweight and inexpensive optic computer mouse sensors, MEMS rate gyroscopes and a pressure-based airspeed sensor. Demonstration of obstacle avoidance by the real platform in natural environments is provided as a video in the electronic supplementary material.

On a moving system, optic flow can serve as a means to estimate proximity of surrounding obstacles (Gibson 1950, Whiteside and Samuel 1970, Koenderink and van Doorn 1987) and thus be used to avoid them, providing that the egomotion of the system is known. Egomotion can be divided into rotational and translational components. Rotation about the 3 axes (Fig. 1) can easily be measured using rate gyroscopes. The components of the translation vector can be much more difficult to measure or estimate on a free-flying system, due to the lack of appropriate sensors. However, assuming no wind, translation can be derived in most cases from the dynamics of the aircraft. Fixed-wing aircraft typically have negligible lateral or vertical displacements, flying essentially along their main axis (x axis in Fig. 1). Rotorcraft behaviour is similar to fixed-wing platforms when they fly in the translational regime (as opposed to near-hover mode where translation patterns can be more complex). Whenever the translation vector is known, it is possible to interpret optic flow measurements as proximity estimation, as long as the rotational component, which is not proportional to proximity (Koenderink and van Doorn 1987), is removed from the measured optic flow. This process can be achieved either by predicting the optic flow generated by the rotations measured by the rate gyroscopes, as we do in the experiments described in this paper, or by actively rotating the vision system to counter the rotation of the body, as flying insects do (van Hateren and Schilstra 1999). Also note that during translational flight the amplitude of the translation vector can easily be measured by means of an onboard airspeed sensor, such as a differential pressure sensor or an anemometer.

In translational flight, steering is commonly achieved by a combination of rolling in the direction of the desired turn and successive pitching. It is therefore sufficient to generate only two signals to steer an aircraft.

After a review of the related work in the next section, section 3 provides a description of the control strategy. Section 4 then presents the platform and experiment methods used to assess the performance. The results are described in section 5. Finally, section 6 discusses potential extensions and improvements.

2 State of the art

Recently, several attempts have been made to add obstacle avoidance capabilities to unmanned aerial vehicles. For example, a 3-kg laser range finder has been embedded on a 95-kg autonomous helicopter (Scherer et al 2008). However, active proximity sensors like laser, ultrasonic range finders or radars tend to be heavy and power-consuming, thus precluding their use on lightweight platforms that are agile and safe enough to operate at low altitude in crowded environments.

Optic flow contains information about the distance to the surroundings that can be used to avoid obstacles and requires only a passive and lightweight vision sensor. For example, optic flow sensors were used to perceive proximity of obstacles or measure altitude (Muratet et al 2005, Barber et al 2005, Griffiths et al 2007, Kendoul et al 2009). However those systems still required GPS and/or an IMU for the control. Other approaches have included optic flow in the control of flying platforms (Barrows et al 2001, Green et al 2003, Chahl et al 2004), but for regulating exclusively altitude or lateral steering and thus still requiring partial manual control. Optic flow has also been used for the control of indoor systems where GPS is not available and weight constraints are even stronger (Zufferey et al 2007, Ruffier

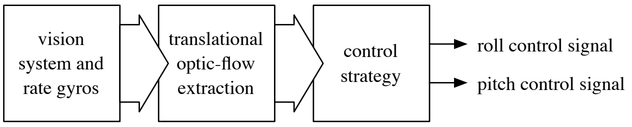
vision system and rate gyros
translational optic-flow extraction
control strategy
roll control signal
pitch control signal

Fig. 2 Overview of the steps required to map the data provided by the vision system and rate gyroscopes into control signals. Yaw is assumed to be passively regulated (see Fig. 1).

and Franceschini 2005; 2008), but external assistance was still required to achieve fully autonomous flight. A control-theoretic framework has been proposed to derive optic-flow-based control laws for autonomous aircraft (Hyslop and Humbert 2008, Humbert et al 2009). However, the demonstration made so far on real and simulated platforms also required external assistance to estimate one or more attitude angle. Recently, altitude estimation and drift control was implemented on a 80-kg helicopter (Garratt and Chahl 2008), but the resulting demonstration was only partially autonomous and the strategy relied on an IMU for attitude control. Finally, one study has proposed a complete autopilot based on visual cues (Neumann and Bülthoff 2002), but the system relied on a separate attitude stabilisation mechanism that would require an additional IMU to be implemented on a realistic flying platform.

In earlier work (Beyeler et al 2007, Zufferey 2008, Zufferey et al 2009), we presented control strategies for indoor flying robots that relied exclusively on visual and gyroscopic information for autonomous flight, with no requirement for explicit state estimation. In this manuscript, we present a generalisation of these preliminary studies that is not limited to indoor platforms and provide a validation of its performance in a simulated urban environment and in real outdoors settings.

3 Control strategy

The vision-based control strategy we propose is made of the three stages shown in Fig. 2. The data provided by a vision system and three orthogonal rate gyroscopes is mapped into signals that can be used to drive the aircraft's controls. The first step consists of extracting optic flow from the information provided by the embedded vision system. Section 3.1 discusses the properties of extracted optic flow that are relevant to proximity detection. Section 3.2 then describes how the visual field can be sampled and section 3.3 describes how optic flow measurements are combined into control signals for steering the aircraft. Finally, section 3.4 proposes a generalisation of the control strategy.

3.1 Proximity estimation using translational optic flow

The fundamental property of optic flow that enables proximity estimation is often referred to as *motion parallax* (Whiteside and Samuel 1970). Essentially, it states that the component of optic flow that is induced by translatory motion is proportional to the magnitude of this motion and inversely proportional to the distance to obstacles in the environment. It is also proportional to the sine of the angle between the translation vector and the viewing direction. This can be written

$$p_{\mathbf{T}}(\theta,\psi) = \frac{|\mathbf{T}|}{D(\theta,\psi)}\sin(\alpha), \tag{1}$$

where $p_{\mathbf{T}}(\theta,\psi)$ is the amplitude of translational optic flow measured in direction (θ,ψ) (see Fig. 3 for the polar coordinate system convention), $\mathbf{T}$ is the translation vector, $D(\theta,\psi)$ is the distance to the obstacle seen in direction (θ,ψ) and α is the angle between the translation vector $\mathbf{T}$ and the viewing direction (θ,ψ).

The optic flow perceived by a free-flying aircraft also contains the component induced by its rotations in addition to the translational optic flow described above. Consequently, it is necessary to exclude the optic flow component due to rotations to estimate the proximity of obstacles, a process known as *derotation* of optic flow (Argyros et al 2004, *e.g.*). In a UAV, this can be achieved by predicting the optic flow generated by rotation, as measured by the rate gyroscopes, and then subtracting the predicted optic flow from the measured optic flow.

In translational flight, the translation vector is essentially aligned with the aircraft's main axis at all times. If the vision system is positioned on the aircraft so that the optical axis is aligned with the translation direction, the angle α in (1) is equal to the polar angle θ of the coordinate system introduced in Fig. 3 (also known as *eccentricity*). Equ. (1) can then be rearranged to express the proximity of obstacle μ (*i.e.* the inverse of distance, also referred to as *nearness*):

$$\mu(\theta,\psi) = \frac{1}{D(\theta,\psi)} \propto \frac{p_{\mathbf{T}}(\theta,\psi)}{\sin(\theta)}. \tag{2}$$

This means that, assuming a constant translation speed, the magnitude of translational optic flow measurements can be directly interpreted as proximity signals, scaled with the sine of the eccentricity θ of the direction where the measurements are taken. In brief, optic flow can provide cues on the nearness of the surrounding obstacles. The next questions are where to look and how to map optic flow signals into control commands.

3.2 Selection of the viewing directions

Let us now consider the directions where optic flow should be measured, the number of measurements that should be

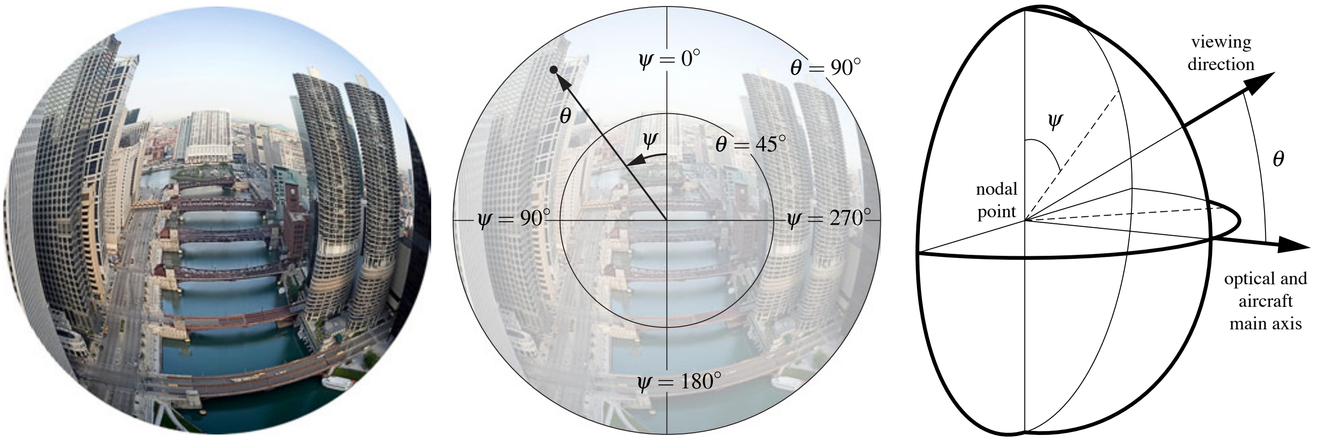

Fig. 3 A large field-of-view is desirable to detect potentially dangerous obstacles in front of the aircraft. *(left)* An example image taken with a fisheye lens covering most of the frontal field-of-view. *(centre)* The image-plane coordinate system used throughout this paper. $\psi \in [0; 2\pi]$ is the azimuth angle, with $\psi = 0$ corresponding to the dorsal part of the visual field and positive extending leftward. $\theta \in [0; \pi]$ is the polar angle. *(right)* Perspective sketch of the spherical vision system. Note that ψ and θ completely define a viewing direction with respect to the optical and the aircraft main axis.

taken, and how to combine them to generate control signals for the aircraft. In order to reduce the sensory and computational requirements, it is desirable to keep the number of measurements as low as possible. It also turns out that not all the viewing directions in the visual field have the same relevance for flight control. Directions pointing at $\theta > 90°$ correspond to obstacles that are behind the aircraft and thus do not require avoidance. For θ values close to 0 (*i.e.* close to the centre of the visual field), the magnitude of the optic flow measurements tends to zero because of the $\sin(\theta)$ factor. Since the resolution of the vision system limits the possibility of measuring small amounts of optic flow, proximity estimation is not reliable for small eccentricities. These two limits ($\theta < 90°$ and $\theta > 0°$) suggest that the area of interest lies around $\theta = 45°$ (Fig. 4) where optic flow measurements are relevant and reliable for controlling the course of an aircraft.

To sample this domain of interest, we propose to measure μ according to (2) in N viewing directions along the specific polar angle $\theta = \hat{\theta}$ and with an inter-azimuthal angle $\hat{\psi}$, as shown in Fig. 5. Formally, these viewing directions can be described by $\{(\theta_k; \psi_k) \mid \theta_k = \hat{\theta}, \psi_k = k \cdot \hat{\psi}, k = 0, 1, ..., N-1\}$.

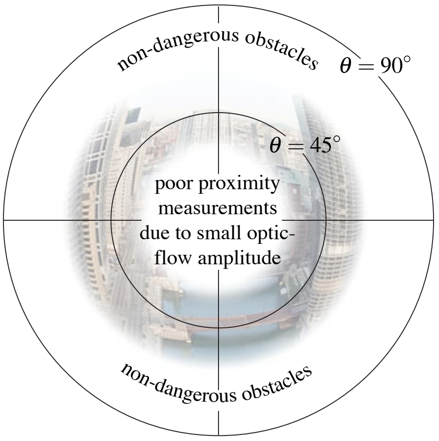

Fig. 4 Representation of the region where proximity estimates are both reliable and relevant for obstacle avoidance. The original fisheye image is faded to white outside this region.

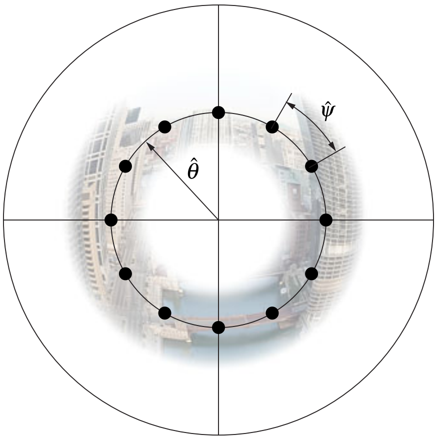

Fig. 5 Possible sampling of the visual field by an hypothetical vision system. N viewing directions are uniformly spaced on a circle at the specific polar angle $\hat{\theta}$. Each viewing direction is separated by an inter-azimuthal angle $\hat{\psi}$. On this illustration, $N = 12$, $\hat{\theta} = 45°$ and $\hat{\psi} = 30°$.

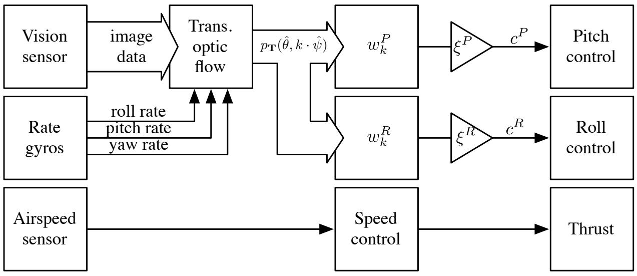

Fig. 6 The complete *optiPilot* control architecture. Data from the vision system and rate gyroscopes is used to extract translational optic flow (section 3.1). Optic flow measurements $p_{\mathbf{T}}$ are then linearly combined using two sets of weights w_k^P and w_k^R, corresponding to pitch and roll control (section 3.3). In parallel, the thrust is controlled by a simple regulator to maintain cruise speed, based on measurements from an airspeed sensor.

3.3 Mapping optic flow into control signals

In order to map optic flow estimations into control signals, we propose the use of a simple weighted sum, which can be written as:

$$\begin{cases} c^P = \dfrac{\xi^P}{N\cdot\sin(\hat{\theta})}\cdot\displaystyle\sum_{k=0}^{N-1} p_{\mathbf{T}}(\hat{\theta}, k\cdot\hat{\psi})\cdot w_k^P, \\ \\ c^R = \dfrac{\xi^R}{N\cdot\sin(\hat{\theta})}\cdot\displaystyle\sum_{k=0}^{N-1} p_{\mathbf{T}}(\hat{\theta}, k\cdot\hat{\psi})\cdot w_k^R, \end{cases} \tag{3}$$

where c^P and c^R are the control signals for the pitch and roll, respectively, w_k^P and w_k^R the associated sets of weights and ξ^P and ξ^R gaisn to adjust the amplitude of the corresponding control signal. This summation process is qualitatively similar to what is believed to occur in the tangential cells of flying insects (Wehner 1987, Krapp et al 1998, Egelhaaf and Kern 2002, Karmeier et al 2006); namely, a wide-field integration of a relatively large number of optic flow estimates into a reduced number of control-relevant signals.

In order to use this approach to steer an aircraft, two sets of weights $\{w_k^R\}$ and $\{w_k^P\}$, $k = 0, 1, ..., N-1$ must be devised, for the roll and the pitch control, respectively. Along with a thrust controller to regulate the flight speed, this control strategy forms a complete autopilot that is illustrated in Fig. 6.

Let us first consider the pitch control signal c^P (Fig. 7 top). Proximity signals from the ventral region (*i.e.* ψ near 180°) correspond to obstacles beneath the aircraft. The corresponding weights should thus be positive to generate a positive control signal that results in a pitch-up manoeuvre. Likewise, proximity signals from the dorsal region (*i.e.* ψ near 0°) correspond to obstacles above the aircraft and the corresponding weights should be negative in order to generate a pitch-down manoeuvre. Finally, proximity signals from the two lateral regions of the aircraft (*i.e.* ψ near 90° and 270°) should not influence the pitching behaviour and the corresponding weights should thus be set to zero. An example of such a weight distribution (Fig. 7 bottom) is given by

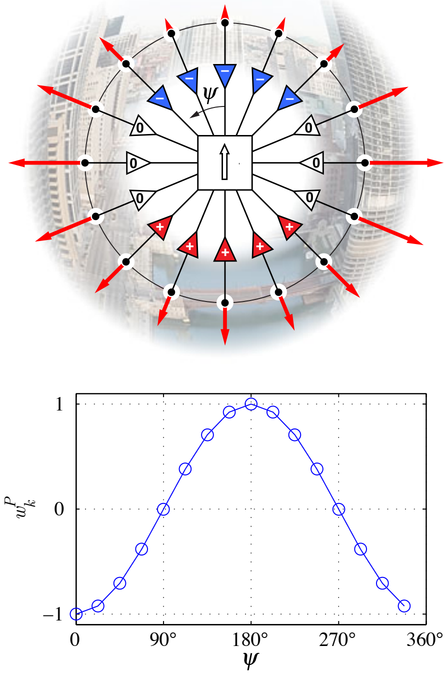

Fig. 7 *(top)* Possible distribution of w_k^P for the generation of the pitch control signal. The arrow in the centre indicates pitch direction for a positive pitch signal. *(bottom)* Example weight distribution according to (4).

$$w_k^P = -\cos(k\cdot\hat{\psi}). \tag{4}$$

Using a similar reasoning, one can derive the qualitative distribution needed for the weights related to the roll signal (Fig. 8). Weights corresponding to the left of the aircraft should be positive, in order to initiate a rightward turn in reaction to the detection of an obstacle on the left. Inversely, weights on the right should be negative. Since obstacles in the ventral region ($\psi = 180°$) are avoided by pitching only, the weights in this region should be set to zero. At first sight, the same reasoning should apply for weights from the dorsal region too. However, doing so would not help the situation when the aircraft is in an upside-down position (*i.e.* with the dorsal part facing the ground), which may result from strong air turbulence, for example. In such situations, it is desirable to steer the aircraft back to an upright and level attitude. This can be achieved by extending the non-zero weights of the lateral regions up to the dorsal field-of-view, as illustrated on Fig. 8 top. These weights, combined with the proximity of ground in the dorsal region, will generate a roll signal leading to the levelling of the aircraft. The following equation

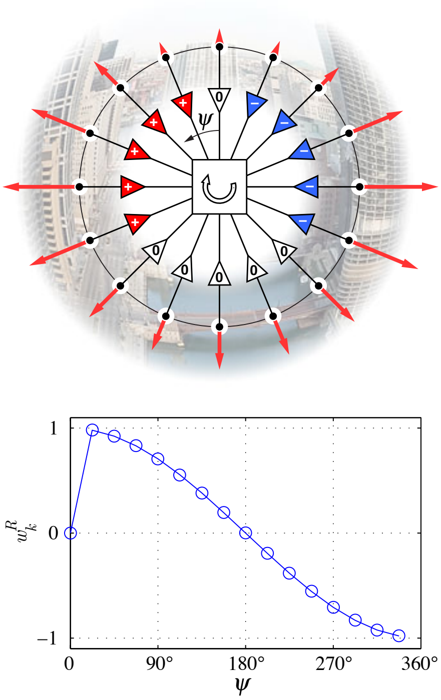

Fig. 8 *(top)* Possible distribution of w_k^R for the generation of the roll control signal. The arrow in the centre indicates roll direction for a positive roll signal. *(bottom)* Example weight distribution according to (5).

is one way to implement such a weight distribution (Fig. 8 bottom):

$$w_k^R = \begin{cases} 0 & k = 0 \\ \cos(k \cdot \frac{\hat{\psi}}{2}) & k > 0 \end{cases} \tag{5}$$

Note that, for the sake of symmetry, the weight at $\psi = 0$ (*i.e.* $k = 0$) is set to zero.

3.4 Extension to non-circular sets of viewing directions

In section 3.2, we described a set of viewing directions uniformly distributed on a single circle at $\theta = \hat{\theta}$. While this approach is intuitive because the $\sin(\theta)$ factor of (2) is constant and thus optic flow measurements can be directly compared, it might be useful to consider alternative distributions. This could be useful if the optic flow estimation algorithm has constraints on the arrangement of the viewing directions or is sensitive to the contrast distribution of the image, or if the environment displays an anisotropic object distribution. Equ. (3) can be generalised to take into account any arbitrary position of the viewing directions in the visual field. After placing the viewing directions on the visual field, the weight values can be computed using (4) and (5) by simply feeding in the azimuth angles ψ_k of the corresponding viewing directions. The control signals are then computed using the following variation of (3):

$$\begin{cases} c^P = \dfrac{\xi^P}{N} \displaystyle\sum_{k=0}^{N-1} \dfrac{p_{\mathbf{T},k}}{\sin(\theta_k)} \cdot w^P(\psi_k), \\[2ex] c^R = \dfrac{\xi^R}{N} \displaystyle\sum_{k=0}^{N-1} \dfrac{p_{\mathbf{T},k}}{\sin(\theta_k)} \cdot w^R(\psi_k), \end{cases} \tag{6}$$

where θ_k is the polar angle for the k^{th} viewing direction.

4 Materials and methods

To validate the *optiPilot* control strategy, we ran a series of experiments both in simulation and with a real flying wing platform. In this section, we describe the platform, the software used and the experimental method we used.

4.1 Flying platform

The platform used for experiments is a flying wing developed in our laboratory (Leven et al 2007; 2009, Fig. 9). This aircraft is neutrally stable in roll and pitch. This means that following a disturbance about either of these axis, the aircraft does not depart from its new orientation nor return to its previous one (Stevens and Lewis 2003). Active stabilisation of both pitch and roll is therefore required to maintain the aircraft airborne. The platform has a wingspan of 80 cm and a total weight of 407 g, including 50 g for the sensor payload required for our experiments. No particular efforts have been made at this stage to reduce the weight of the sensors. It is equipped with an electronic board including a Microchip dsPIC33FJ256GP506 microcontroller, on which our control strategy was implemented. This controller is interfaced to three Analog Devices ADXRS610 rate gyroscopes that we used for optic flow derotation. In order to measure airspeed, it is also equipped with a Freescale MPXV5004DP differential pressure sensor and a custom-built pitot tube. A simple proportional regulator is sufficient to regulate the thrust of this platform to maintain the cruise airspeed at 14 m/s, with a precision of ± 2 m/s[1].

In order to record the state of the aircraft during the experiments, it was further fitted with a Xsens MTi-G unit which provides a full 6-degree-of-freedom state estimation. At no time however this unit was used for the control of the

[1] Note that variations of velocity affect the estimation of proximity and therefore the resulting behaviour. An increase of velocity renders apparent distances shorter than they are in reality, resulting in the aircraft being steered further away from obstacles. This is desirable as high velocities incur a greater risk of damage in case of collision. Inversely, lower velocities result in the aircraft being steered closer to obstacles, which is acceptable due to the reduced kinetic energy and increased manoeuvrability.

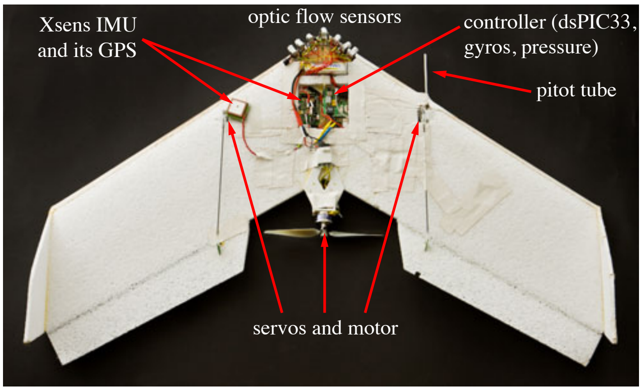

Fig. 9 Top view of the flying wing used for the experiments. It has a wing span of 80 cm and a total weight of 407 g including about 50 g of sensor payload.

aircraft. The states of the aircraft and its sensors were monitored and recorded in real time using a 2.4 GHz Digi XBee-PRO radio-link and the *Ishtar* monitoring software framework (Beyeler et al 2008).

During the experiments, a human pilot could take over the control of the aircraft using a regular RC controller. This capability was used to steer the platform into specific situations and to subsequently activate autonomous control to assess its behaviour in autonomous mode. The data presented in this paper include only fully autonomous flight sequences, unless mentioned otherwise.

4.2 Optic flow detection

There are a number of technologies available to estimate optic flow, including standard cameras and vision processing (Srinivasan 1994, Barron et al 1994 for a review), dedicated sensors such as aVLSI or mixed-mode custom vision chips (Barrows et al 2001, Mehta and Etienne-Cummings 2003, Moeckel and Liu 2007), custom motion detectors based on photodiodes (Pudas et al 2007) or optic mouse chips (Barber et al 2005, Griffiths et al 2007, Rodriguez et al 2007, Dahmen et al 2009). The latter have the advantages of being lightweight, available off-the-shelf, easy to interface to the electronics and not requiring further processing, as the optic flow extraction is done on-chip. Also, they provide a true image velocity measurement that is independent from the contrast frequency and the image intensity. Each sensor provides a single optic flow estimation, which requires the installation of as many chips as required viewing directions. This limitation, for a small number of viewing directions, is outweighed by the advantages listed above.

Fig. 10 illustrates the optic flow detectors we developed. They are based on the Avago ADNS5050 optic mouse sensors, the Philips CAX100 collimator lens ($f = 10$ mm) and a custom-designed lens mount that clips directly onto the chip casing. The optics were calibrated such as to maximise the measure of image quality provided by the sensor. Each optic flow detector weighs 0.8 g. The raw output of the optic flow sensor is derotated with the raw output of the rate gyroscopes and fed to the control strategy. In this initial implementation, no attempt has been made to reduce sensor noise with low-pass filtering.

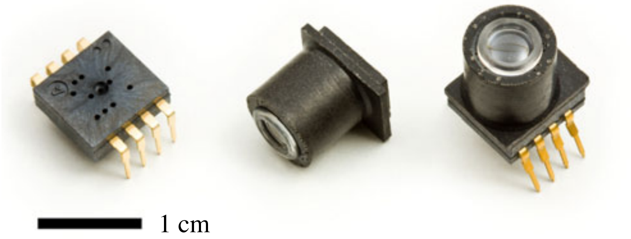

Fig. 10 From left to right: the Avago ADNS5050 optic mouse sensor, the custom-designed optics based on the Philips CAX100 collimator lens ($f = 10$ mm) and the assembled optic flow detector (weighing 0.8 g).

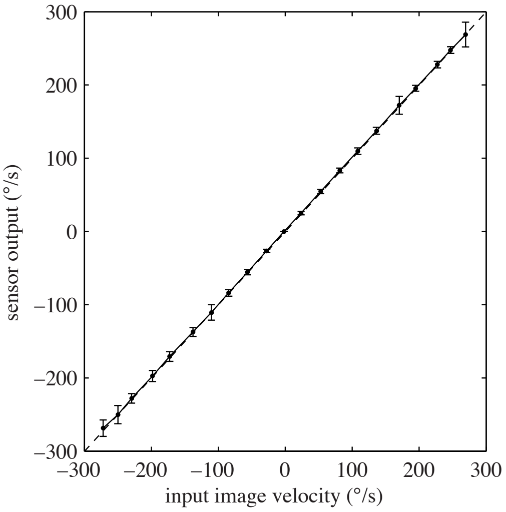

Fig. 11 Characterisation of the optic flow detector. The graph shows the sensor output for varying image velocities experienced when placed on a rotating platform in an outdoor environment surrounded with trees and buildings. The input image velocities where measured using an Analog Devices ADXRS610 rate gyroscope. The sensor has a linear output and usable standard error in the range of $\pm 280°/s$. Saturation of the sensor occurs beyond 300°/s.

In order to assess the performance of the optic flow detectors, we characterised them by comparing their output to a rate gyroscope when placed on a rotating platform in an outdoor environment. Fig. 11 shows the resulting data. The optic flow detectors have a linear output and a usable standard error in the range of $\pm 280°/s$.

4.3 Simulation setup

The experiments in simulation were performed with a custom simulation package relying on OpenGL[2] for visual rendition and on the Open Dynamics Engine[3] (ODE) to simulate the physics. The software includes an aerodynamic model of the flying wing described above. The aerodynamics model is implemented using the standard stability derivative method (Cooke et al 1992). Coefficients are applied to the various parameters of the state of the aircraft (such as sideslip, angle of attack and the translational and rotational components of the speed), in order to compute the resulting aerodynamical force and moment at the centre of gravity. These forces and moments are then passed on to ODE for the computation of the aircraft kinematics. The coefficients were identified from wind-tunnel experiments and empirically tuned so that the simulated and real platforms displayed the same behaviour when remotely controlled by an expert pilot.

To model the optic flow detectors presented above, we first computed a theoretical measure of translational optic flow derived from the motion of the aircraft and the distance to obstacles in each of the viewing directions. We then perturbed these values using a noise model that captures the noise behaviour of the optic flow sensors. We consider two sources of noise. The first source of noise is the consequence of the aperture problem inherent to optic flow (Fennema and Thompson 1979) and can also be caused by aliasing problems when viewing objects that are textured with repetitive patterns. In such cases, the optic flow estimation can be completely altered across the entire range. We modelled this type of noise with a uniform distribution. The second source of noise is given by the image-capture process, imprecision in optics geometry and other imperfections of the vision hardware. We modelled this type of noise with a Gaussian distribution centred on the theoretical optic flow value. We use as optic flow input in the simulation the noisy value generated by one of the two sources of error with a given probability (Thrun et al 2005):

$$x' = \begin{cases} \mathscr{U}(0, k_{\max} \cdot x) & P = 10\% \\ \mathscr{N}(x, \sigma) & P = 90\% \end{cases} \tag{7}$$

where x is the theoretical optic flow value and x' the noisy value used in the simulation. The distribution parameters were adjusted so that the resulting probability distribution matches measurements made with our optic flow detector. Their values are $k_{\max} = 1.2$ and $\sigma = 0.07$ rad. Note that we did not model the possibility of temporary lack of contrast in the portion of the scene where the viewing direction is pointing. As discussed in section 6, such occurrence happen rarely in natural environments but may arise in man-made settings.

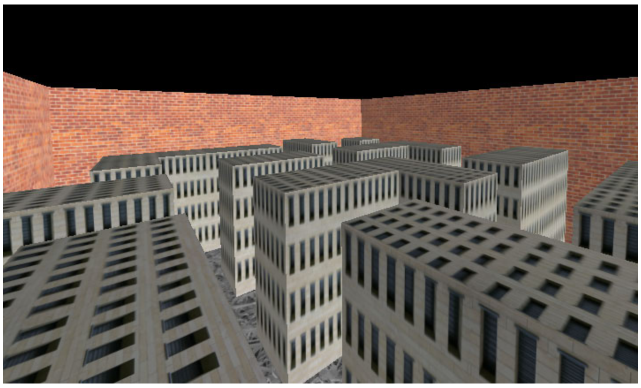

Fig. 12 Aerial view of the simulated urban environment. The environment spans 500 by 500 m and the alleys between buildings are 50 m wide.

4.4 Validation method

In order to validate the *optiPilot* control strategy, we designed four sets of experiments with both the simulated and the real platforms.

As many future applications and arguably the most challenging conditions for UAVs are related to flying at low altitude in constructed environments, we first explore the ability of *optiPilot* to avoid collisions in a simulated urban-like maze environment. The environment is composed of 150 m tall buildings of various shapes, separated by 50 m wide alleys and surrounded by high walls (Fig. 12; a precise map of the environment is overlaid in Fig. 14). To alleviate the potentially tedious process of tuning the real vision system manually, we take advantage of the simulation setup to systematically explore the effect of the eccentricity angle $\hat{\theta}$ and the inter-azimuthal angle $\hat{\psi}$ on the performances. For each combination of $\hat{\theta}$ and $\hat{\psi}$, the performance of the control strategy is measured as the average flight duration over 100 trials. At the beginning of each trial, the aircraft is positioned at a random point in the middle of an alley and at an altitude of 50 m. It is left free to fly for 5 minutes or until it collides with an obstacle in the environment. The maximum possible performance thus corresponds to an average flight duration of exactly 5 minutes, indicating that no collision occurred during the 100 trials.

With the second set of experiments, we aim to analyse the stability of the simulated aircraft while flying over a flat terrain. In such an obstacle-less situation, the aircraft should fly along straight trajectories and reject external perturbations. We show the disturbance rejection capabilities by systematically perturbing the aircraft around the pitch and roll axes. We also show how *optiPilot* is able to regulate altitude by studying the behaviour of the aircraft when launched from various altitudes with zero speed and a level attitude.

In order to demonstrate the stabilisation capability of *optiPilot* in reality, we ran a third set of experiments over flat

[2] `http://www.opengl.org/`

[3] `http://www.ode.org/`

Table 1 Parameter values used in the simulation experiments.

Parameter	Value
pitch gain ξ^P	6
roll gain ξ^R	12
pitch weights w_k^P	according to (4)
roll weights w_k^R	according to (5)
pitch bias	-15%
airspeed set-point	14 m/s

terrain, similar to the previous one but with the real platform (section 4.1). We show how *optiPilot* rejects disturbances when the aircraft is perturbed using predefined sequences of control signals and how it regulates altitude when activated while the platform is flying at various initial heights over ground. We finally consider situations where the aircraft is diving and must recover to level flight in order to avoid a collision with the ground.

As an initial validation of the obstacle avoidance capabilities of *optiPilot*, we ran a final set of experiments were the aircraft was manually steered towards different types of trees. Once the aircraft was aligned with the obstacle, *optiPilot* was switched on to asses its capability to avoid a collision.

Video excerpts of these experiments are available in the electronic supplementary material accompanying this paper. The experiments with the real platform were carried out during the winter over ploughed crop fields or meadows. Unreported experiments have also been run over water or the grass of a soccer field.

5 Results

5.1 Obstacle avoidance in simulation

Taking advantage of the simulation setup, we systematically explored in the urban-like environment (Fig. 12) the effect of the value of the eccentricity $\hat{\theta}$ and inter-azimuthal angle $\hat{\psi}$ on the performance. The other parameters required by the control strategy are shown in Table 1 and were maintained constant for all experiments in simulation. In order to keep the aircraft near to the ground, we added a bias of -15% on the elevator deflection. This means that, for a null signal generated by the control strategy, the aircraft has a slight tendency to pitch downward. This value, as well as those of the pitch and roll control gains ξ^P and ξ^R, were empirically set to produce a response profile that matches the flight dynamics of our flying platform.

Fig. 13 summarises the results of these experiments. It appears that with an inter-azimutal angle $\hat{\psi} = 90°$ (*i.e.* only $N = 4$ viewing directions homogeneously spread around the aircraft main axis), the performance is relatively poor, and naturally increases for $\hat{\psi} = 45°$ and $\hat{\psi} = 30°$ ($N = 8$ and $N = 12$, respectively). However, there is almost nothing to gain from further reducing the inter-azimuthal angle below $\hat{\psi} = 30°$, which seems to optimally combine parsimony and performance. For all values of $\hat{\psi}$, the performance is strongly influenced by the eccentricity $\hat{\theta}$, with an optimum lying near $\hat{\theta} = 45°$. This corresponds to results obtained in a similar situation using both a theoretical and an empirical method (Hrabar and Sukhatme 2006). Therefore, $\hat{\psi} = 30°$ and $\hat{\theta} = 45°$ are chosen as our reference values for the remaining experiments.

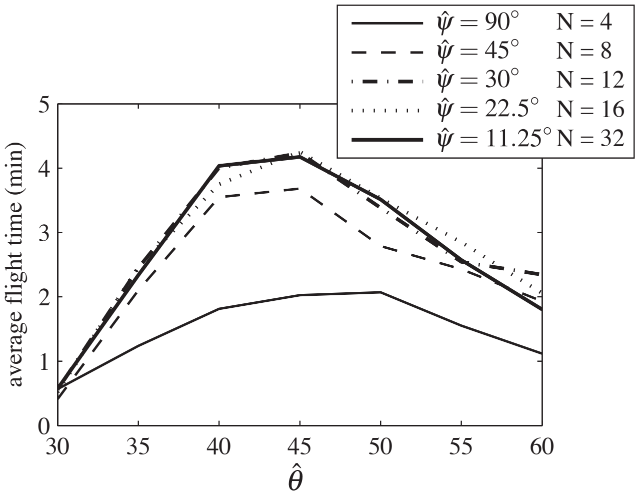

Fig. 13 Performance of the control strategy in the urban-like environment for each combination of eccentricity angle $\hat{\theta}$ and inter-azimuthal angle $\hat{\psi}$. Performance is indicated as the average flight time over 100 flights. All trials where limited to 5 min if no collision occurred. The optimal eccentricity angle is $\hat{\theta} = 45°$ and the performance does not increase with inter-azimuthal angles below $\hat{\psi} = 30°$.

Let us examine the behaviour of *optiPilot* in the urban-like environment with these parameters. Fig. 14 shows the occupancy density computed from the 100 flights with $\hat{\psi} = 30°$ and $\hat{\theta} = 45°$, indicating which areas of the test environment the aircraft visited most often. It shows that the aircraft flew by maintaining its trajectory in the middle of the alleys. This is reminiscent of the behaviour known as the *centring response* of flying insects, which are believed to balance optic flow perceived by each eye in order to automatically fly in the centre of the available space (Srinivasan and Zhang 2004). Fig. 14 also shows the location of the 16 collisions that happened during the 7 hours of test flights. In most cases, collisions occurred when an obstacle was presented symmetrically in front of the aircraft. In such situations, due to the symmetry of the two weight distributions, control signals have a low value, sometimes leading the aircraft to a collision. This is further discussed in section 6.

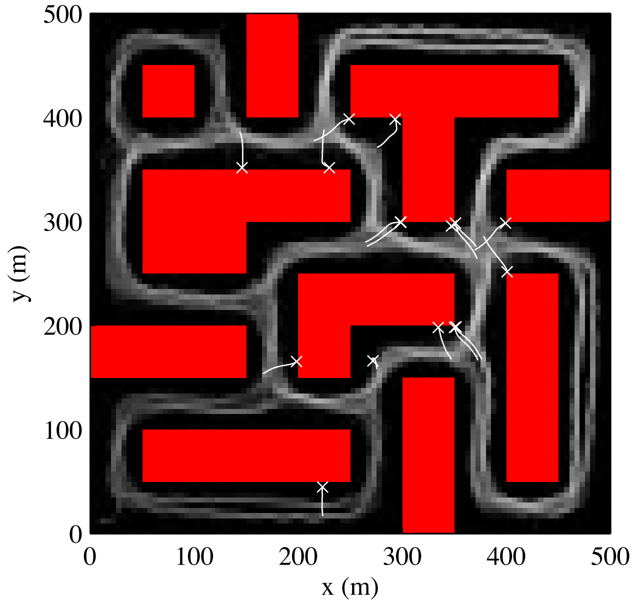

Fig. 14 Occupancy density of the aircraft in the urban-like environment for $\hat{\theta} = 45°$ and $N = 12$. The visiting frequency is indicated by the grey level of the corresponding cell; brighter areas correspond to locations that the aircraft visited more often (the resolution is 5 m). 100 flights are represented, spanning more than 7 hours in total. During this time, only 16 collisions were recorded and are represented by white crosses along with the trajectory during the 3 s before colliding.

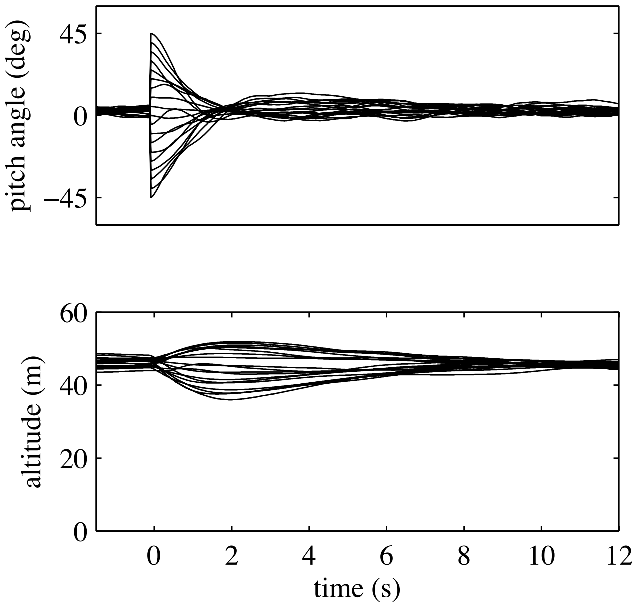

Fig. 15 Pitch angle (top) and altitude (bottom) profiles of the simulated platform during pitch angle perturbation experiments. The aircraft was initially in stable and level flight. At $t = 0$, the aircraft was rotated by an angle ranging from -45°to 45°(with 5°intervals) around its pitch axis (the 19 profiles are represented). The pitch angle was regulated within about 3.5 s back to the small positive value required to generate lift for level flight, with variations of altitude of approximately 10 m in the worst cases.

5.2 Flight stability in simulation

We also validated the capability of the proposed control strategy to regulate flight over a flat terrain, when no obstacle is present in the environment.

Fig. 15 shows the behaviour of the simulated aircraft when perturbed around the pitch axis over an infinitely flat ground. Rotations of various magnitudes, in the range of ±45°, were applied at time $t = 0$ in order to observe the reaction of the aircraft. In all cases, *optiPilot* steers the aircraft to the small positive pitch attitude required to generate lift for level flight within about 3.5 s. The variations of altitude remain within about 10 m in the worst cases (which corresponds to downward perturbations, where the effect of gravity adds up to the perturbation).

Fig. 16 shows that *optiPilot* rejects perturbations about the roll axis equally well. In this experiment, the aircraft was artificially rotated, at $t = 0$, with angles in the range of ±60° about the roll axis. In all cases, *optiPilot* steered back the aircraft to a level attitude (in about 1 s). The temporary reduction of lift due to the banked attitude explains the small variations of altitude.

Finally, Fig. 17 shows the behaviour of the aircraft when launched with zero speed and a level altitude at various heights over ground ranging from 10 to 60 m. In all cases, the aircraft initially drops while gaining the velocity needed to generate lift and, within about 5 s, reaches a cruise altitude

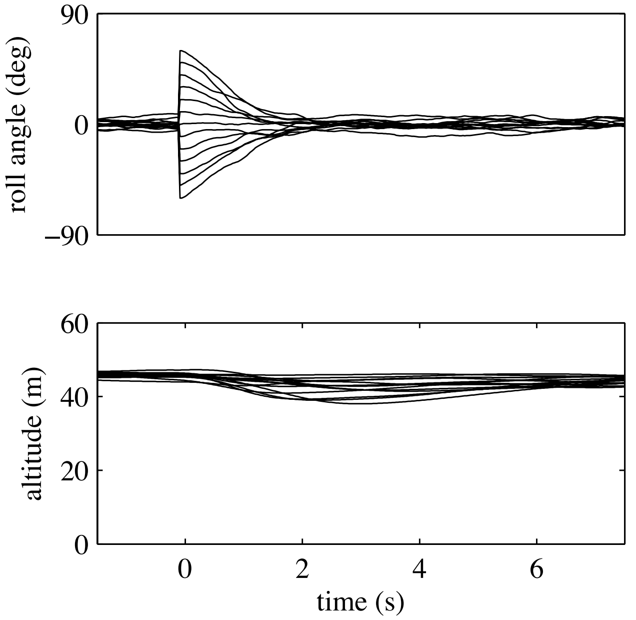

Fig. 16 Roll angle (top) and altitude (bottom) profiles of the simulated platform during roll angle perturbation experiments. The aircraft was initially in stable and level flight. At $t = 0$, the aircraft was rotated by an angle ranging from -60°to 60°(with 10°intervals) around its roll axis (the 13 profiles are represented). In all cases, the roll angle is regulated back to a level attitude in less than 1 s, with variations of altitude of approximately 8 m in the worst cases.

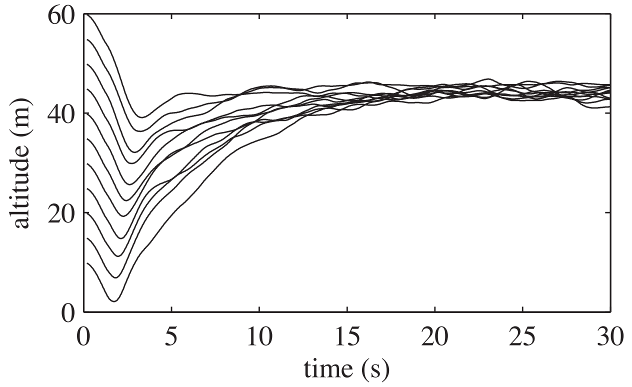

Fig. 17 Altitude profiles of the simulated platform. The platform was released at various altitudes with zero speed and level attitude. Eleven profiles are represented for flights starting at altitudes ranging from 10 to 60 m, with intervals of 5 m. After an initial altitude drop of approximately 5 s, occurring while gaining speed, the control strategy steers the aircraft to a stable altitude of approximately 40 m irrespective of the initial starting height.

of 40 m irrespective of the initial height. Note that this cruise altitude is not explicitly regulated. Rather, it results from the equilibrium between the nose-down trim and the tendency to avoid the ground. The resulting cruise altitude can be adjusted by tuning the pitch control gain and the pitch bias.

It is important to notice that the accurate regulation of both attitude and altitude implicitly derives from a control strategy originally designed for obstacle avoidance. Neither attitude angles nor altitude are explicitly estimated nor measured. Rather, flight stabilisation emerges from the interaction between the ground and the avoidance behaviour that strives to keep obstacles in the ventral region of the aircraft. This contrasts with the typical regulation strategies used by classical autopilots that require explicit estimation of the 6 degree-of-freedom state of the aircraft, at the cost of expensive sensing and processing systems. Contrary to classical autopilots, *optiPilot* regulates altitude with respects to the ground. On irregular terrain, the resulting behaviour corresponds to ground following, as illustrated by the videos included in the supplementary material.

5.3 Flight stability with the real platform

As an initial set of validation experiments with the real aircraft, we tested the ability of the control strategy to stabilise flight and reject disturbances when flying over flat terrain. Due to technical constraints (limitations of the I/O on the current embedded electronics), we could only implement 7 optic flow sensors. We chose to keep the eccentricity and inter-azimuthal angles to the value of $\hat{\theta} = 45°$ and $\hat{\psi} = 30°$, which lead to the best performance in the simulated urban-link environment (section 5.1), and implemented only the bottom half of the sampling circle. This means that the 7 optic flow detectors were pointing towards each side as well as below the aircraft, as shown in Fig. 18. Natural outdoor environments typically display a strong anisotropy as obstacles are mostly on the sides and below a flying agent such as our test platform. The lack of viewing direction pointing above the aircraft should therefore not impair its ability to stabilise flight. The other parameters used during the experiments with the real platform were manually tuned in-flight and are summarised in Table 2.

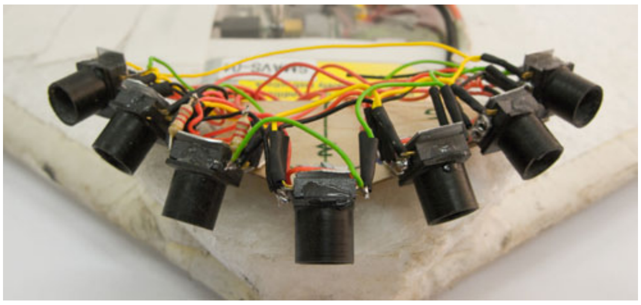

Fig. 18 Close-up view of the vision system made of 7 optic flow sensors (see Fig. 10). The viewing directions are pointing to each side as well as below the aircraft, with an eccentricity angle of $\hat{\theta} = 45°$ and azimuthal angles of $\psi_k = 90°$, 120°, 150°, 180°, 210°, 240° and 270° ($\hat{\psi} = 30°$).

Table 2 Parameter values used in the experiments with the real platform.

Parameter	Value
pitch gain ξ^P	8.1
roll gain ξ^R	8.1
pitch weights w_k^P	according to (4), $k = 3$ to 9
roll weights w_k^R	according to (5), $k = 3$ to 9
pitch bias	-25%
airspeed set-point	14 m/s

Fig. 19 illustrates how our control strategy rejects perturbations of the pitch angle during autonomous flight over flat terrain. It shows data from several flights that were perturbed, at time $t = 0$, by applying a predefined sequence of commands on the elevator (grey zone). In all cases, our control strategy managed to recover to a stable pitch angle within about 2 s, with variations of altitude below ±5 m. Fig. 19 also shows the average optic flow perceived during the experiments. In level flight, more optic flow is perceived below the aircraft than on the sides, which is expected when flying over a flat terrain. When perturbed upwards, the magnitude of optic flow slightly decreases as the aircraft pitches up and gains altitude. Inversely, when perturbed downward, the magnitude of optic flow strongly increases, resulting in a quick pitch-up reaction.

OptiPilot regulates the roll angle equally well. Fig. 20 shows data from several flights that were perturbed by applying, at time $t = 0$, full deflection of ailerons, leading to

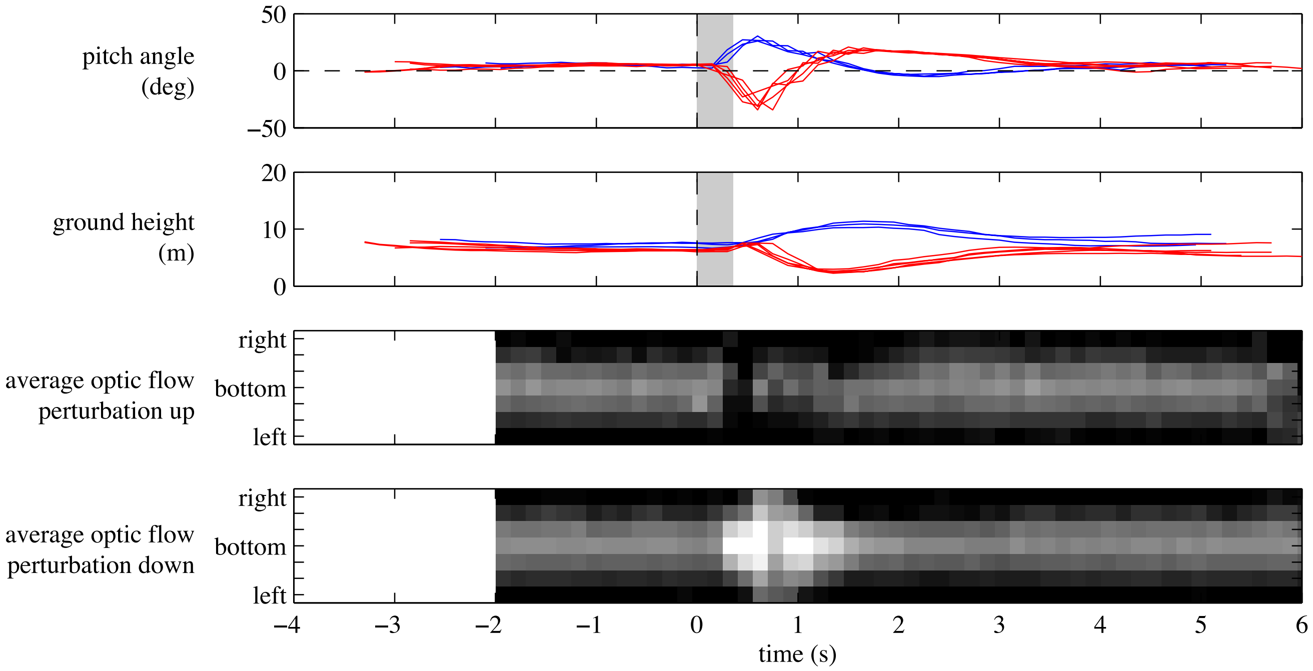

Fig. 19 Data recorded from the on-board sensors during the pitch perturbation experiments with the real platform. Eight flights are shown, during which the aircraft was perturbed by a predefined command sequence (grey background) on the pitch axis, either up- or downward. The pitch angle and altitude of each flight is plotted, as well as the average translation-induced optic flow perceived by the aircraft in each viewing direction.

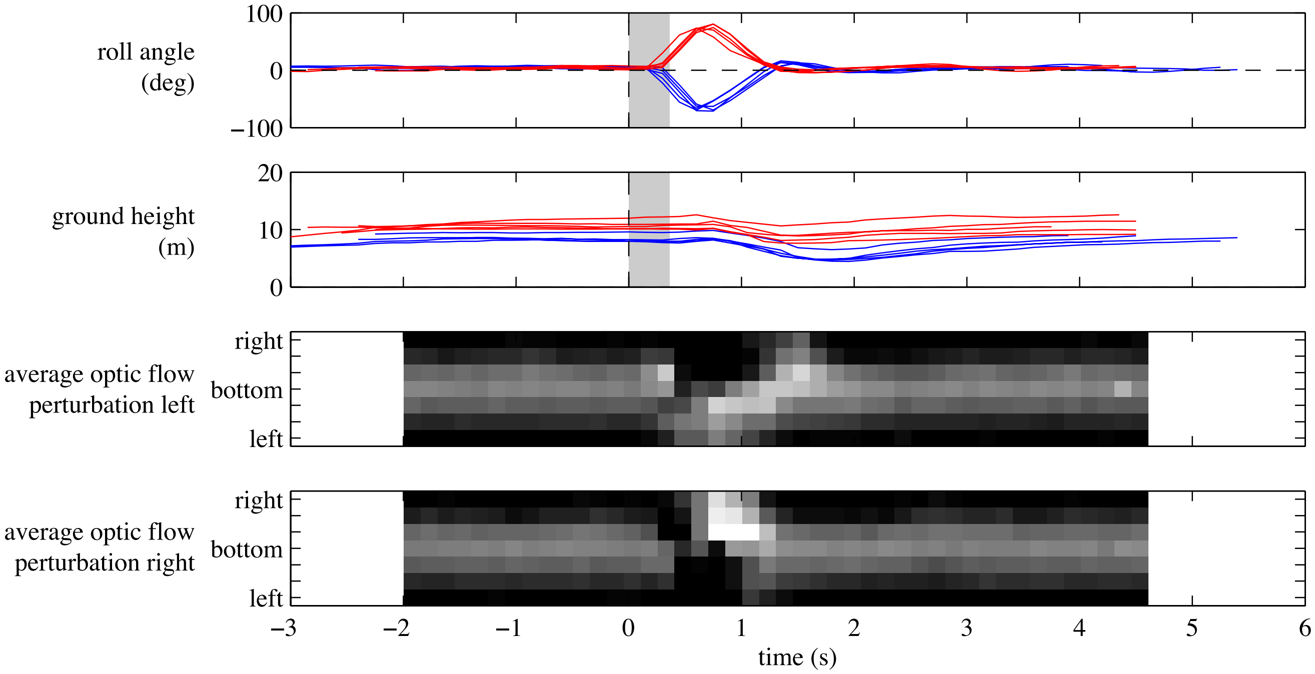

Fig. 20 Data recorded from the on-board sensors during the roll perturbation experiments with the real platform. Ten flights are shown, during which the aircraft was perturbed by a predefined command sequence (grey background) on the roll axis, either left- or right-ward. The roll angle and altitude of each flight is plotted, as well as the average translation-induced optic flow perceived by the aircraft in each viewing direction.

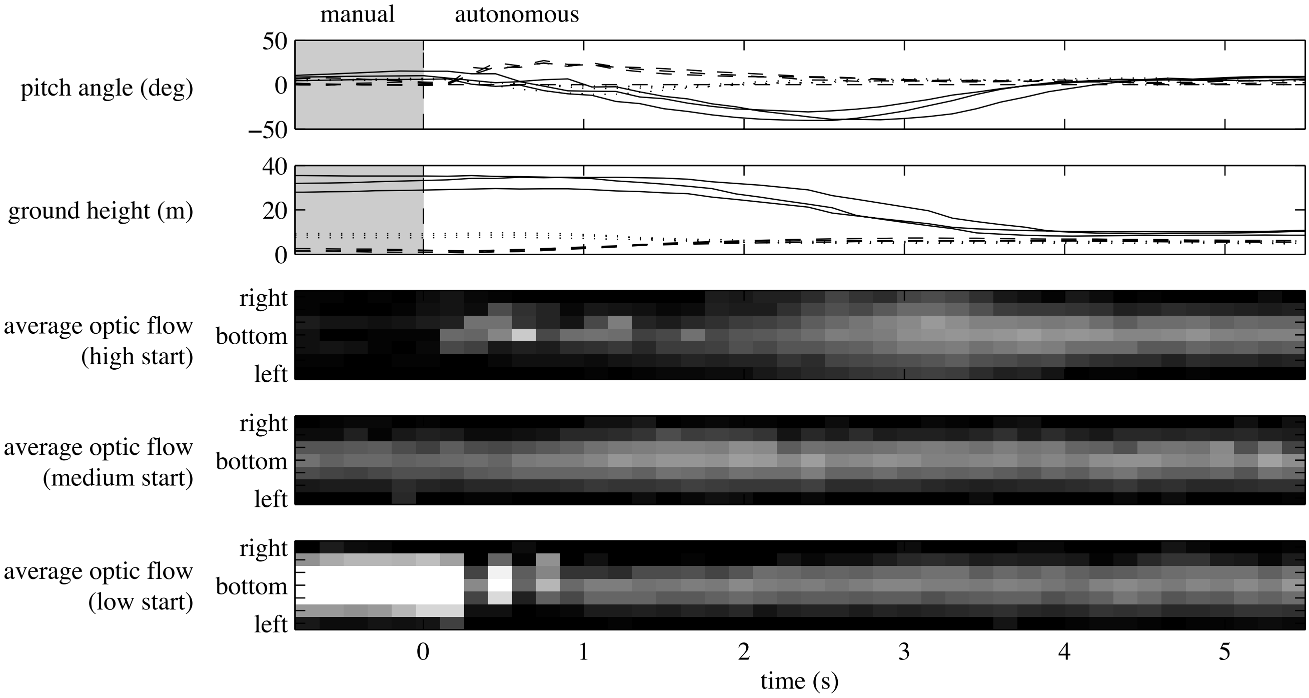

Fig. 21 Data recorded from the on-board sensors during the altitude regulation experiments with the real platform. Nine flights are shown, during which the aircraft was manually controlled (grey background) at an altitude of either about 2 m (dashed lines), 10 m (dotted lines) and 30 m (solid lines), before activating *optiPilot* at $t = 0$. The pitch angle and altitude of each flight is plotted, as well as the average translation-induced optic flow perceived in each viewing direction when started from each of the three altitudes.

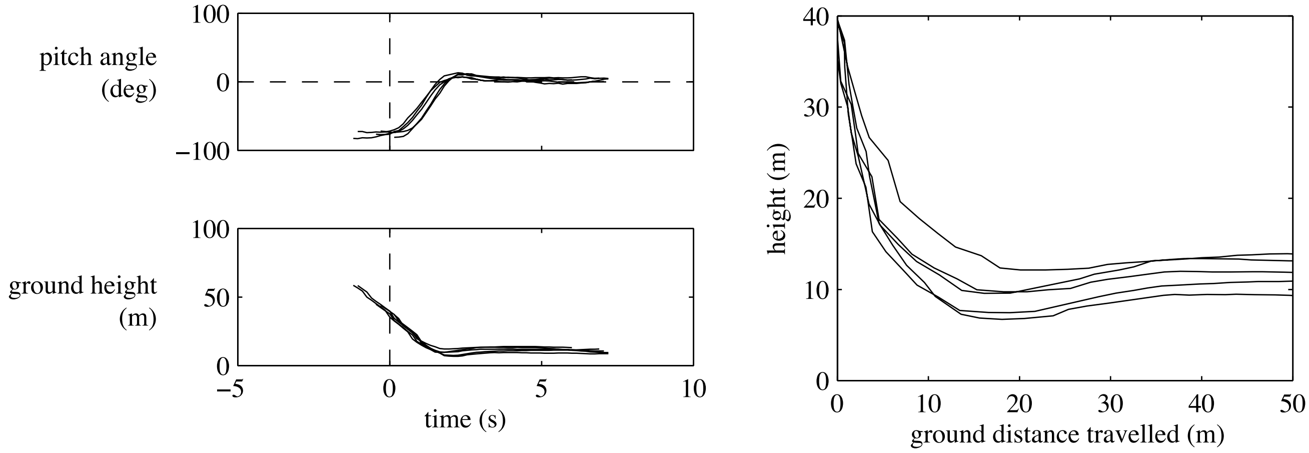

Fig. 22 Data recorded from the on-board sensors during the dive experiments with the real platform. Five flights are shown, during which the aircraft was manually steered into a dive from high altitude (more than 50 m) before activating *optiPilot*. Flight were aligned at $t = 0$ when crossing the 40 m altitude. On the left, the pitch angle, pitch rate and altitude of each flight are plotted. On the right, the vertical trajectories of the 5 flights are shown.

a roll angle of approximately ±80°. In all cases, the aircraft returned to level attitude in less than 1.5 s, with variations of altitude within ±3 m. The average optic flow perceived during the experiment shows that, when perturbed, the distribution strongly shifts toward the side of the perturbation, which leads to a roll reaction that brings the aircraft back to a level attitude. Note that the optic flow distributions during the perturbed portions of flight are not symmetrical. This is due to asymmetries within the vision system, whose individual optic flow detectors were aligned by hand. The data presented in this section shows that this asymmetry does not translate into a notably degraded or asymmetric behaviour.

Fig. 21 illustrates flights where, at time $t = 0$, autonomous control was switched on, while the aircraft was manually steered at various initial altitudes. In all cases, *optiPilot* steered the aircraft back to the same altitude of about 8 m.

The aircraft reaches this natural altitude in much less time when it starts below it than when it starts at higher altitudes. This is explained by the fact that the downward drive, when flying high, only comes from the nose-down bias on the elevator set to -25%. The upward reaction when flying low is instead generated by the strongly increasing ventral optic flow experienced when flying close to the ground. As the translation-induced optic flow is inversely proportional to the distance, reducing the altitude of a small amount generates a significant increase in the perceived optic flow. The bottom graphs of Fig. 21 illustrate how the average optic flow initially experienced by the aircraft is dependant on the altitude. In this experiment, the optic flow distribution is kept centred below the aircraft by the control strategy at all times except when the aircraft starts high, where the absence of consistent optic flow in the ventral region (due to the larger distance to the ground) may lead to some drift around the roll axis. Fig. 21 shows that this drift is immediately corrected as soon as the aircraft gets closer to the ground.

Fig. 22 illustrates the behaviour of the aircraft when manually steered into a vertical dive towards the ground before switching *optiPilot* on. At time $t = 0$, when the aircraft crosses 40 m above ground, the pitch angle is about -90°, *i.e.* completely nose-down. The control strategy progressively steers the aircraft towards a level attitude and completely stabilises the flight at an altitude of about 10 m.

Collectively, these experiments with the real platform show that a direct mapping of translation-induced optic flow measurements to control signals is sufficient to regulate both the attitude angles and the altitude of a free-flying aircraft, without requiring an explicit estimation of its 6-degree-of-freedom state.

5.4 Obstacle avoidance with the real platform

The last set of experiment aimed at assessing the capability of *optiPilot* to avoid obstacles. Flights were recorded where the aircraft was manually steered towards two different types of trees before switching *optiPilot* on.

Fig. 23 illustrates the behaviour of the aircraft when approaching large trees. Two different angles of approach are shown. When the aircraft approaches at an angle, a limited amount of rolling (up to about 60°) is necessary to curve the trajectory away from the obstacle. This behaviour has nearly no effect on the altitude. When the aircraft approaches the obstacle frontally, the required avoidance manoeuvre is sharper and the aircraft rolls to an angle of more than 90°. At such an attitude, where the wings are perpendicularly oriented with respect to the ground, no lift can be generated to counteract gravity. The altitude of the aircraft therefore drops temporarily until the obstacle is passed and *optiPilot* stabilises the aircraft back to a level attitude.

Fig. 24 shows encounters with small trees, where *optiPilot* steers the aircraft over the obstacle instead of the lateral manoeuvres observed with larger trees. In this case, the roll angle shows little variation but the altitude increases up to twice the cruise altitude in order to avoid a collision.

6 Discussion

Autonomous flight among obstacles in urban environment is one of the long-term goals of this research. So far, we obtained mixed results with man-made structures. The reason for this lies in the fundamental difference between images of natural and man-made scenes (Ruderman 1994). Contrast in natural scenes exhibits scale invariance, which means that the presence of contrast is not dependant on the distance between the viewer and the objects. This property is advantageous for optic flow extraction, as it depends on the presence of contrast. Unfortunately, man-made environments behave differently from natural scenes. As an example, contrast on a concrete wall can only be perceived from either a very close distance (where centimetre-scale irregularities become apparent) or from sufficiently far away (where building-scale edges are visible). At intermediate distances, any vision system will struggle to extract optic flow on such surfaces due to the lack of contrast. A number of measures can be taken to cope with this issue. For example, the number of viewing direction could be significantly increased to maximise the chance of looking at an edge. Assuming a vision-system made of a single, wide-field-of-view camera, an edge detection algorithm could be used to choose suitable viewing directions, before applying the generalised version of our control strategy (section 3.4). In any case, it is important to understand that this problem relates specifically to the process of estimating optic flow. Provided sufficiently accurate proximity estimates, the proposed control strategy will perform as well in man-made situations as natural environments, as demonstrated by the good results obtained in simulation (section 5.1).

A likely limitation of our control strategy that is independent from the optic flow estimation process concerns the ability to detect small obstacles that may arise in the centre of the field-of-view, without intersecting any of the viewing directions where optic flow is extracted. This limitation is inherent to the fundamental property of optic flow that limits the ability to estimate proximity of obstacles in the direction of flight (see (1)). Note however that if the aircraft is constantly manoeuvring, obstacles will rarely remain unseen by staying exactly in the centre of the field-of-view. One way to cope with this issue could be to complement the control strategy with a single, forward-pointing distance sensor, based on infrared triangulation or laser interferometry (Griffiths et al 2007, *e.g.*), for example. The output of this sensor could be directly linked to the elevator control signal

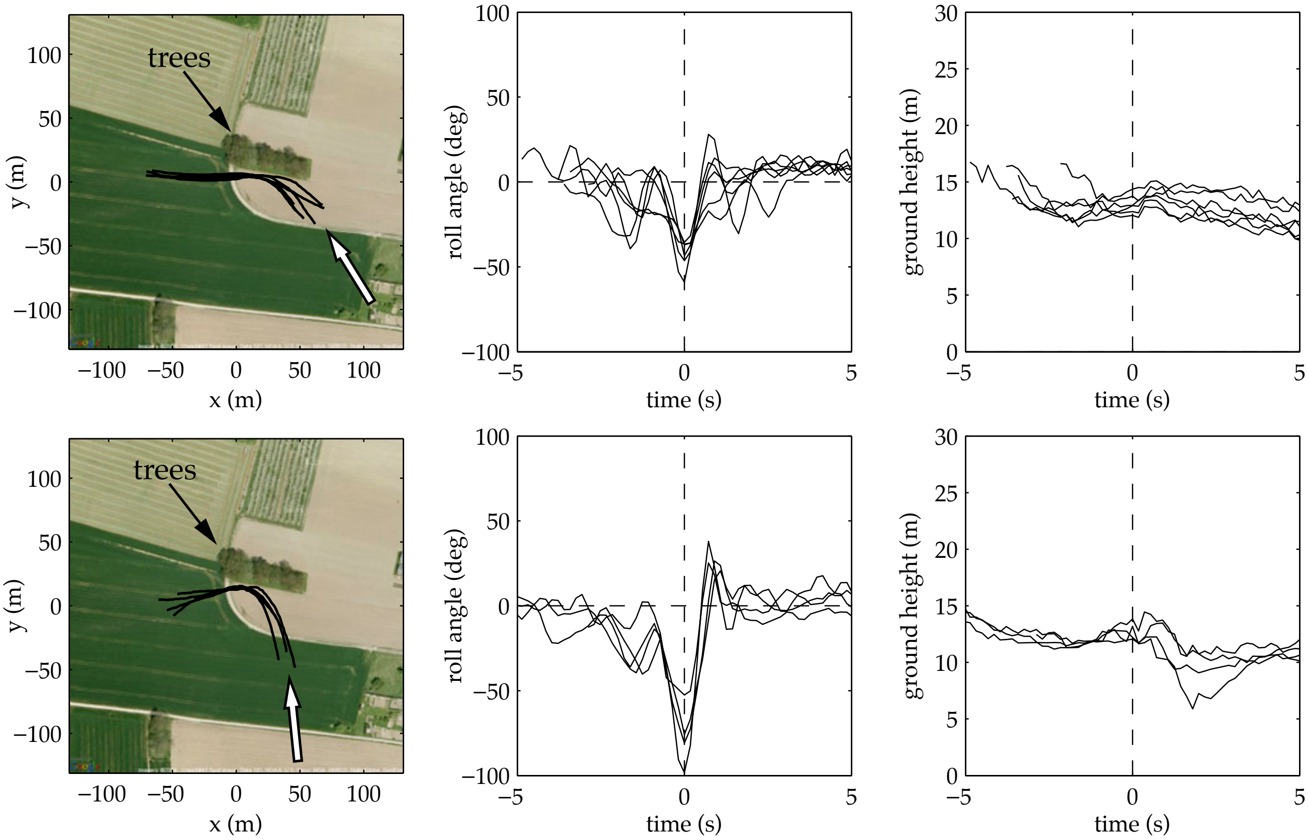

Fig. 23 Lateral avoidance of a group of tall trees (between 20 and 30 m) with two different angles of approach. Trajectories are shown on the left graphs, the middle graphs show the roll angle and the right graphs show the height over ground. Time $t = 0$ corresponds to the highest roll angle achieved during the manoeuvre. The top graphs show data from flights where the aircraft approached the obstacle at an angle. The roll angle reaches a maximum of about 60° towards the left to avoid the tree, while the height over ground remains mostly constant. The bottom graphs show data from flights where the aircraft approached nearly frontally the obstacle. The roll angle reaches more than 90° in order to achieve the sharp turn required to avoid the obstacle. In this case, the height over ground is temporarily perturbed due to the loss of lift incurred by highly banked attitudes.

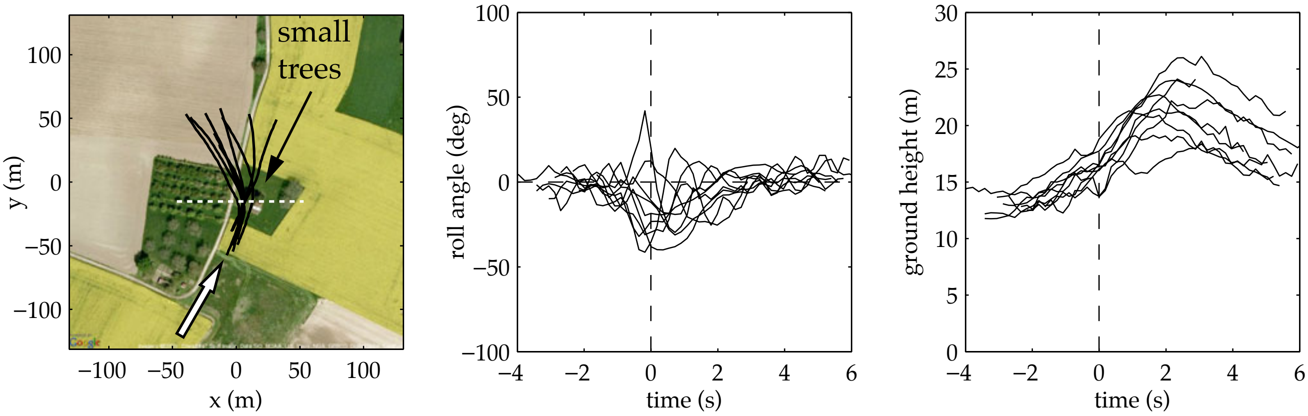

Fig. 24 Avoidance of small trees (between 10 and 15 m). The left graph shows trajectories of flights approaching frontally a group of two small trees. The middle graph shows the roll angle and the right graph shows the altitude followed by the aircraft. Time $t = 0$ corresponds to when the aircraft crossed the white dashed line on the left graph. The data shows that the aircraft rolled only of a small amount and followed a straight or slightly curved trajectory. The obstacle was avoided by flying over it, as shown be the increasing height over ground. The maximum height achieved (25 m over ground) is about twice the cruise altitude.

so that the presence of small obstacles in the centre of the field-of-view would trigger a pitch up manoeuvre, steering the aircraft over them.

A related problem arises when the aircraft flies perpendicularly toward a large, symmetrical obstacle, as noted before (Green and Oh 2008) and seen in section 5.1. In such situations, both pitching and rolling control signals remain small while the aircraft is approaching the surface because of the symmetry in the sets of weights associated to the control and in the perceived pattern of optic flow. This problem can easily be solved by adding the central distance sensor and elevator control suggested above. It would generate a pitch up manoeuvre when approaching the surface, creating an asymmetry that would then allow the normal obstacle-avoidance behaviour to take over. Alternatively, this situation can also be easily detected by monitoring the total amount of translational optic flow over all viewing directions. This value will reach unusual proportions as the aircraft approaches the obstacle. The control program could then include an open-loop sharp turn, *i.e.* a saccade, to be executed if the total optic flow signal exceeds a pre-defined threshold (Zufferey and Floreano 2006). This strategy, which does not require the presence of additional distance sensors, is similar to the behaviour observed in flies, which respond to an expanding optic flow field by generating saccadic turns (Tammero and Dickinson 2002).

The experiments described in this paper were performed with the sinusoidal weight distributions described in (4) and (5) (Fig. 7 and 8, respectively). According to our experience, as long as the weight distributions capture the features discussed in section 3.3, the choice of sinusoidal, piece-wise linear, or other shape does not significantly affect the performance. On the other hand, weight distributions that do not respect the conditions laid out in section 3.3 can produce completely different results, some of them being even desirable in specific conditions. For example, by setting to zero the dorsal weights associated with roll control, upside-down flight can be made possible, as well as upright level flight.

Throughout this paper, we assume that the translation vector of the aircraft is always aligned with its main axis. While this is essentially the case most of the time, the translation vector can occasionally deviate from this position. This can be due to an increase of the angle-of-attack needed to generate additional lift (during steep turns, *e.g.*), transient side-slip during fast manoeuvres, or lateral drift due to crosswind. In most case, the magnitude of the translation vector deviation remains within a few degrees (Stevens and Lewis 2003), which will not significantly affect the performance of *optiPilot*. However, strong crosswinds may on occasion lead to larger deviations. For example, crosswinds of the same magnitude as the cruise speed of the aircraft will lead to deviations of the translation vector of 45°, which will significantly perturb our control strategy. Future work includes studying the behaviour of *optiPilot* in such instances. Note that tail- or headwinds do not lead to significant deviations of the translation vector but rather to a scaling of the perceived optic flow, which is proportional to the speed of the aircraft relative to the ground. This will translate into changes in altitude of flight, as observed in previous studies (Ruffier and Franceschini 2005, Franceschini et al 2007).

As it stands, the behaviour implemented by *optiPilot* resembles a goal-less wandering directed only by how the ground and obstacles repel the aircraft. While the capability of flying near obstacles can be of critical use in real-world applications, it is often not sufficient to fulfil missions that require the aircraft to follow a path or to reach a specified goal. In future work, we will investigate the possibility to laterally steer the aircraft while retaining the basic flight regulation and obstacle avoidance behaviour. Initial work showed that this can be achieved using dynamic modifications of the weight distributions that lead to regulation of arbitrary attitudes. In particular, banked attitudes result in turns that could be triggered to steer the aircraft towards specific goal locations.

7 Conclusion

The *optiPilot* control strategy relies only on a few simple, lightweight and low-consumption sensors, such as optic mouse chips or other vision sensors, rate gyroscopes and an airspeed sensor. The proposed solution allows a UAV to fly and avoid obstacles using a simple sensor-to-actuator mapping by exploiting properties of translation-induced optic flow and the dynamics of flying platforms (which typically fly along their main axis). It is thus quite different from the majority of existing autopilots that rely on 6 degree-of-freedom state estimation (using GPS and IMU) and fly well above obstacles.

As demonstrated with the real platform, a control strategy based on optic flow is technologically competitive for UAVs in the sub-kilogram range because it does not rely on heavy sensors, such as laser range finders or other active sensors. However, this situation may change in the future with the advent of novel technologies, such as 3D imagers capable of recovering depth information (Niclass et al 2005, *e.g.*), or miniaturisation of existing ones, such as scanning laser range-finder (Scherer et al 2007, *e.g.*). In this context, optic-flow-based proximity estimation will still remain competitive because it relies on a passive and thus power efficient sensor, but may occasionally be discarded in favour of an alternative technology better suited for the task at hand, such as operation in the dark or in areas with heavy smoke and dust. Even in this case, the control strategy proposed in this article maintains its interest because it can be easily interfaced to any type of proximity estimation. This manner of directly linking proximity signals to actuators to steer away from obstacles is reminiscent of Braitenberg's imaginary vehicles (Braitenberg 1984) but extended to the third dimension. The simple wiring of his vehicles produced behaviours that an observer would attribute to complex control mechanisms. While being parsimonious in its implementation, *optiPilot* is capable of approximating flight performance that were so far only achieved by human pilots.

The results presented in this paper shed new light on the classic flight control problem, suggesting that the complex sensors and processing required for 6 degree-of-freedom state estimation are not required for altitude and attitude regulation; a problem that can instead be solved by a simple sensor-to-actuator wiring also capable of collision-free translational flight. The proposed control strategy may also be promising for our long-term goal of integrating autonomous control for current 10-gram (Zufferey et al 2007) and upcoming sub-gram flying platforms (Fearing et al 2002, Wood 2008).

Acknowledgements We would like to thank Severin Leven for the development of the test platform and André Guignard for the design and construction of the lens mounts. The authors are also indebted to Emily Baird for improving the manuscript. Finally, the authors would like to thank Prof. Mandyam V. Srinivasan for the fruitful collaboration in the domain of visual guidance of UAVs.

References

Argyros AA, Tsakiris DP, Groyer C (2004) Biomimetic centering behavior for mobile robots with panoramic sensors. IEEE Robotics and Automation Magazine 11(4):21–68

Barber D, Griffiths S, McLain T, Beard R (2005) Autonomous landing of miniature aerial vehicles. In: AIAA Infotech@Aerospace

Barron J, Fleet D, Beauchemin S (1994) Performance of optical flow techniques. International Journal of Computer Vision 12(1):43–77

Barrows G, Neely C, Miller K (2001) Optic flow sensors for MAV navigation. In: Mueller TJ (ed) Fixed and Flapping Wing Aerodynamics for Micro Air Vehicle Applications, Progress in Astronautics and Aeronautics, vol 195, AIAA, pp 557–574

Beard R, Kingston D, Quigley M, Snyder D, Christiansen R, Johnson W (2005) Autonomous vehicle technologies for small fixed-wing UAVs. Journal of Aerospace Computing, Information, and Communication 2(1):92–108

Beyeler A, Zufferey JC, Floreano D (2007) 3D vision-based navigation for indoor microflyers. In: Proc. IEEE Int. Conf. on Robotics and Automation, Roma, pp 1336–1341

Beyeler A, Magnenat S, Habersaat A (2008) Ishtar: a flexible and lightweight software for remote data access. In: Proceedings of the 2008 European Micro Air Vehicle Conference EMAV08

Braitenberg V (1984) Vehicles - Experiments In Synthetic Psychology. The MIT Press, Cambridge, MA

Chahl J, Srinivasan M, Zhang H (2004) Landing strategies in honeybees and applications to uninhabited airborne vehicles. The International Journal of Robotics Research 23(2):101–110

Cooke J, Zyda M, Pratt D, McGhee R (1992) NPSNET: Flight simulation dynamic modeling using quaternions. Presence: Teleoperators and Virtual Environments 1(4):404–420

Dahmen H, Millers A, Mallot H (2009) Insect inspired odometry by optic flow recorded with optical mouse chips. In: Floreano D, Zufferey J, Srinivasan M, Ellington C (eds) Flying Insects and Robots, Springer, in press

Egelhaaf M, Kern R (2002) Vision in flying insects. Current Opinion in Neurobiology 12(6):699–706

Fearing R, Avadhanula S, Campolo D, Sitti M, Jan J, Wood R (2002) A Micromechanical Flying Insect Thorax, MIT Press, pp 469–480

Fennema C, Thompson W (1979) Velocity determination in scenes containing several moving objects. Computer graphics and image processing 9:301–315

Franceschini N, Ruffier F, Serres J (2007) A bio-inpired flying robot sheds light on insect piloting abilities. Current Biology 17:1–7

Garratt MA, Chahl J (2008) Vision-based terrain following for an unmanned rotorcraft. Journal of Field Robotics 25(4):284–301

Gibson J (1950) The Perception of the Visual World. Houghton Mifflin, Boston

Green W, Oh P (2008) Optic-flow-based collision avoidance. IEEE Robotics & Automation Magazine 15(1):96–103

Green W, Oh P, Sevcik K, Barrows G (2003) Autonomous landing for indoor flying robots using optic flow. In: ASME International Mechanical Engineering Congress and Exposition, Washington, D.C., vol 2, pp 1347–1352

Griffiths S, Saunders J, Curtis A, McLain T, Beard R (2007) Obstacle and terrain avoidance for miniature aerial vehicles. In: Valavanis K (ed) Advances in Unmanned Aerial Vehicles: State of the Art and the Road to Autonomy, vol 33, Springer, chap I.7, pp 213–244

van Hateren J, Schilstra C (1999) Blowfly flight and optic flow. II. head movements during flight. Journal of Experimental Biology 202:1491–1500

Hrabar S, Sukhatme GS (2006) Optimum camera angle for optic flow-based centering response. In: Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, Beijing, China, pp 3922–3927

Humbert J, Conroy JK, Neely C, Barrows G (2009) Wide-field integration methods for visuomotor control. In: Floreano D, Zufferey J, Srinivasan M, Ellington C (eds) Flying Insects and Robots, Springer, chap 5, in press

Hyslop AM, Humbert J (2008) Wide-field integration methods for autonomous navigation of 3-D environments. In: Proceedings of the 2008 AIAA Guidance, Navigation and Control Conference and Exhibit

Karmeier K, van Hateren JH, Kern R, Egelhaaf M (2006) Encoding of naturalistic optic flow by a population of blowfly motion-sensitive neurons. Journal of Neurophysiology 96:1602–1614

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 57:591602

Koenderink J, van Doorn A (1987) Facts on optic flow. Biological Cybernetics 56:247–254

Krapp H, Hengstenberg B, Hengstenberg R (1998) Dendritic structure and receptive-field organization of optic flow processing interneurons in the fly. Journal of Neurophysiology 79:1902–1917

Leven S, Zufferey JC, Floreano D (2007) A simple and robust fixed-wing platform for outdoor flying robot experiments. In: International Symposium on Flying Insects and Robots, pp 69–70

Leven S, Zufferey JC, Floreano D (2009) A minimalist control strategy for small UAVs. In: Proceedings of the 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, St. Louis

Mehta S, Etienne-Cummings R (2003) Normal optical flow chip. In: Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS 2003), pp 784–787

Moeckel R, Liu SC (2007) Motion detection circuits for a time-to-travel algorithm. In: IEEE International Symposium on Circuits and Systems (ISCAS), New Orleans, pp 3079–3082

Muratet L, Doncieux S, Brière Y, Meyer J (2005) A contribution to vision-based autonomous helicopter flight in urban environments. Robotics and Autonomous Systems 50(4):195–209

Neumann T, Bülthoff H (2002) Behavior-oriented vision for biomimetic flight control. In: Proceedings of the EPSRC/BBSRC International Workshop on Biologically Inspired Robotics, pp 196–203

Niclass C, Rochas A, Besse PA, Charbon E (2005) Design and characterization of a CMOS 3-D image sensor based on single photon avalanche diodes. IEEE Journal of Solid-State Circuits 40(9):1847–1854

Pudas M, Viollet S, Ruffier F, Kruusing A, Amic S, Leppävuori S, Franceschini N (2007) A miniature bio-inspired optic flow sensor based on low temperature co-fired ceramics (ltcc) technology. Sensors and Actuators A 133:88–95

Rodriguez A, Andersen E, Bradley J, Taylor C (2007) Wind estimation using an optical flow sensor on a miniature air vehicle. In: AIAA Conference on Guidance, Navigation, and Control

Ruderman DL (1994) The statistics of natural images. Computation in Neural Systems 5:517–548

Ruffier F, Franceschini N (2005) Optic flow regulation: the key to aircraft automatic guidance. Robotics and Autonomous Systems 50(4):177–194

Ruffier F, Franceschini N (2008) Aerial robot piloted in steep relief by optic flow sensors. In: Proceedings of the 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2008) (ISCAS 2008), pp 1266–1273

Scherer S, Singh S, Chamberlain L, Saripalli S (2007) Flying fast and low among obstacles. In: Proceedings of the 2007 IEEE Conference on Robotics and Automation, pp 2023–2029

Scherer S, Singh S, Chamberlain L, Elgersma M (2008) Flying fast and low among obstacles: Methodology and experiments. The International Journal of Robotics Research 27(5):549–574

Srinivasan M (1994) An image-interpolation technique for the computation of optic flow and egomotion. Biological Cybernetics 71:401–416

Srinivasan M, Zhang S (2004) Visual motor computations in insects. Annual Reviews in Neuroscience 27:679–696

Stevens BL, Lewis FL (2003) Aircraft Control and Simulation, 2nd edn. Wiley

Tammero L, Dickinson M (2002) The influence of visual landscape on the free flight behavior of the fruit fly drosophila melanogaster. The Journal of Experimental Biology 205:327–343

Thrun S, Burgard W, Fox D (2005) Probabilistic Robotics. MIT Press

Valavanis KP (2007) Advances in Unmanned Aerial Vehicles. Springer

Wehner R (1987) Matched filters - neural models of the external world. Journal of Comparative Physiology A 161:511–531

Whiteside T, Samuel G (1970) Blur zone. Nature 225:94–95

Wood R (2008) Fly, robot, fly. IEEE Spectrum 45(3):25–29

Zufferey J, Beyeler A, Floreano D (2009) Optic-flow to steer and avoid collisions in 3D. In: Floreano D, Zufferey J, Srinivasan M, Ellington C (eds) Flying Insects and Robots, Springer, chap 6, in press

Zufferey JC (2008) Bio-inspired Flying Robots: Experimental Synthesis of Autonomous Indoor Flyers. EPFL/CRC Press

Zufferey JC, Floreano D (2006) Fly-inspired visual steering of an ultralight indoor aircraft. IEEE Transactions on Robotics 22:137–146

Zufferey JC, Klaptocz A, Beyeler A, Nicoud JD, Floreano D (2007) A 10-gram vision-based flying robot. Advanced Robotics, Journal of the Robotics Society of Japan 21(14):1671–1684