Optic-flow-based steering and altitude control for ultra-light indoor aircraft

Jean-Christophe Zufferey and Dario Floreano
Autonomous Systems Lab (http://asl.epfl.ch)
Swiss Federal Institute of Technology in Lausanne (EPFL)
name.surname@epfl.ch

***Abstract*— Our goal is to demonstrate the ability of bio-inspired techniques to solve the problem of piloting an autonomous 30-gram aircraft within textured indoor environments. Because of severe weight and energy constraints, inspiration has been taken from the fly and only visual and vestibular-like sensors are employed. This paper describes the models and algorithms that will be used for altitude control and obstacle avoidance, mainly relying on optic flow. For experimental convenience, both mechanisms have first been implemented and tested on a small wheeled robot featuring the same hardware as the targeted aircraft.**

I. Introduction

Our goal is autonomous navigation of a 30-gram indoor plane [11] in a square arena of about 16 by 16 meters at the speed of 2m/s. It should fly straight when no wall is in front of it, engage in a preprogrammed turn whenever a frontal obstacle is detected, and maintain altitude above ground. Because of severe weight and energy constraints, distance sensors (such as active infrared, sonar, and laser range finder) cannot be used on such an aircraft. Visual sensors represent a good alternative because they can be light-weight and low-power. But how can we retrieve distance information from the image stream? To answer this question, we take inspiration from flying insects, which are known to rely on optic flow (OF) for navigation. OF is the projection of 3D point velocities onto the imaging surface. Flies move through the environment by means of a series of straight trajectories separated by rapid turns, known as saccades. There is evidence that frontal image expansion (rate of OF divergence) is the signal triggering each saccade [16]. It has also been shown that honeybees use ventral image velocity to control landing [15]. In a stationary environment, OF on the retina is produced by translation and rotation of the agent. Only OF due to translation (TransOF) contains information on distance from objects in the scene. From the point of view of distance estimation, rotational optic flow (RotOF) is a spurious signal, which contaminates the motion field. This is probably why flies move on straight trajectories between saccades and also use a gaze stabilization mechanism based on halteres (gyroscopic sensors sensitive to angular velocities about the roll, pitch and yaw axes) [8]. In this paper, we highlight the interference between RotOF and TransOF and suggest a method to achieve efficient distance estimation.

The indoor aircraft is equipped with two 1D cameras (one-dimensional array of pixels) and miniature gyroscopes for RotOF cancellation. One camera looks forward horizontally (for frontal obstacle avoidance) and the other one downward with its array oriented in the direction of motion (for altitude control). OF is computed in real-time by an on-board 8-bit microcontroller. Distances to walls and ground are estimated using OF amplitude and field divergence, as suggested by the biological mechanisms mentioned above. For experimental convenience, we have first tested our approach on a small Khepera robot fitted with the same electronic hardware used on the indoor aircraft (microcontroller, camera, gyroscope). The robotic platforms (Figure 1) as well as their equipping hardware are described in [17].

Several similar investigations inspired by the vision-based navigational behaviour of flying insects, have been carried out. Terrestrial robots for obstacle avoidance [3], [6] or corridor following (for a review, see [14]) have been developed, but most of them either use wheel encoders for compensation of interference due to RotOF or implement gaze stabilization. Since flying robots have no contact with ground, our approach cannot rely on wheel encoders. The tight weight budget (about 10g available for the sensors and microcontroller) also precludes active camera mechanisms for gaze stabilization. Furthermore, all the above mentioned robots, except the machine based on analog-electronics by Franceschini and colleagues [3], feature off-board image processing and control. A couple of other projects on aerial robots also involve OF for navigation. Specific studies on altitude control have been conducted in simulation [7], in tethered helicopters [9], [12], and in outdoor unmanned air vehicles [2]. Another work in simulation [10] demonstrated full 3D navigation of a minimalist model of an insect. Some preliminary trials of obstacle avoidance and automatic landing have been made with an indoor plane [4], but without taking care of RotOF interference.

II. Ideal optic-flow properties and control strategies

In this section, we first consider ideal patterns of OF as if they were produced by ideal 2D cameras. In a second step, we focus on the limited field of view (FOV) corresponding to the 1D cameras equipping the robots in order to devise the control policies that will be tested in Section IV.

The 2D OF field is the projection of the relative 3D velocities of the points in the environment onto the imaging surface. For the sake of simplicity, we assume that the photoreceptors

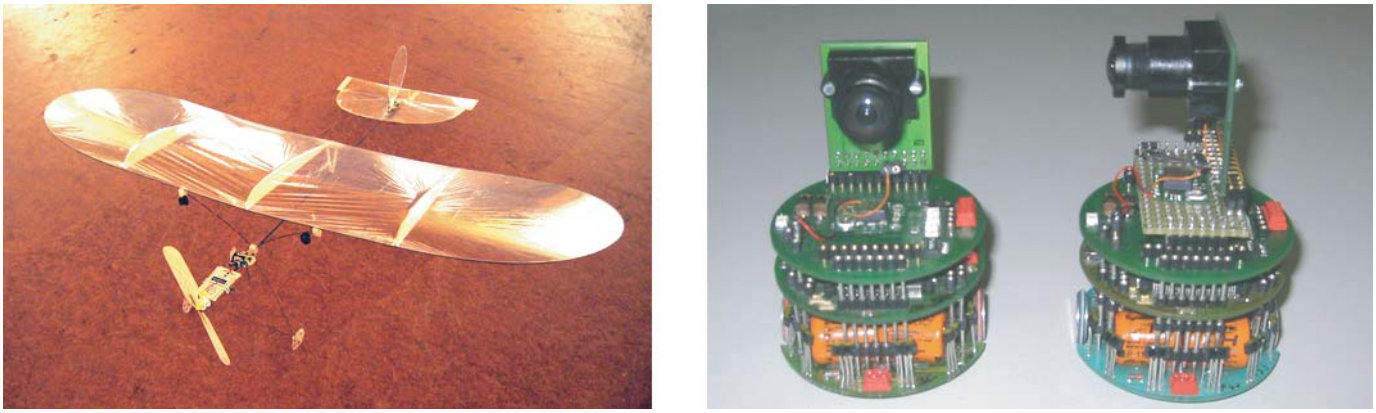

Fig. 1. *Left.* Our indoor slow-flyer able to fly at 2m/s with 1.3m turning radius. *Right.* Two Khepera robots, one equipped for frontal obstacle avoidance and the other for lateral wall following. MEMS gyroscopes for yaw rotation detection are visible on the top turret, below the cameras.

of the vision sensor are arranged on a sphere with unit radius (Figure 2, left). Each of them defines a viewing direction indicated by the 3D unit vector $\mathbf{d}(\psi,\theta)$, which is a function of the azimuth ψ and elevation θ. The 3D motion of a rigid body can always be characterized unambiguously by a 3D translation vector $\mathbf{t}$ and a 3D rotation vector $\mathbf{r}$ (describing the axis of rotation and its amplitude). When the vision sensor is moving in a stationary environment with translation $\mathbf{t}$ and rotation $\mathbf{r}$, the OF field $\mathbf{p}(\psi,\theta)$ can be described by [5]:

$$\begin{aligned} \mathbf{p}(\psi,\theta) &= \underbrace{-\frac{\mathbf{t}-(\mathbf{t}\cdot\mathbf{d}(\psi,\theta))\,\mathbf{d}(\psi,\theta)}{D(\psi,\theta)}}_{\text{TransOF}} \underbrace{-\mathbf{r}\times\mathbf{d}(\psi,\theta)}_{\text{+RotOF}} \\ \text{OF} &= \end{aligned} \tag{1}$$

where $D(\psi,\theta)$ is the distance between the sensor and the object seen in direction $\mathbf{d}(\psi,\theta)$. Although $\mathbf{p}(\psi,\theta)$ is a 3D vector field, it is by construction tangential to the sensor surface. OF field is thus generally represented by unfolding the spherical surface into a Mercator map (see for instance Figure 2, right-most panel).

It is important to notice that OF is a sum of two distinct components (see for instance Figure 2, right graphs). The first one, TransOF, is due to translation and contains information about distances from obstacles, while the second one, RotOF, is produced by rotation and does not depend on distances. From Eq.1 we see that TransOF amplitude is inversely proportional to distance $D(\psi,\theta)$. Therefore, if the translation is known and the rotation is null, it is possible to estimate that distance. However, it is quite common in free-maneuvering agents that RotOF overwhelms TransOF, thus making extraction of distance information difficult.

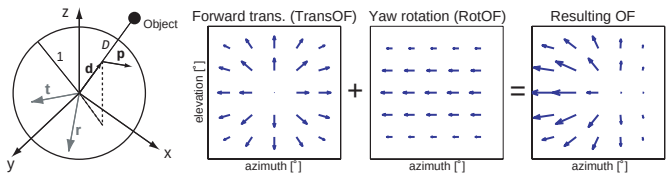

Fig. 2. *Left.* Ideal sphere model of the visual sensor. *Right.* OF maps from superposition of TransOF and RotOF.

Our aerial robot is intended to fly along straight trajectories interspersed with predefined turning actions. However, it is not possible to ensure absolutely null rotation during straight motion. For instance, whenever an attitude (pitch and roll) correction has to be performed for maintaining altitude, the plane will rotate about its pitch axis, generating some RotOF. We therefore continuously deduce RotOF for each camera using information coming from a MEMS gyroscope, whose axis is oriented perpendicularly to both the camera viewing direction and the line defined by the photoreceptor array. Gyroscopes provide a functionality similar to that given by halteres in insects.

Having discounted RotOF, we can now concentrate on typical TransOF patterns, as viewed from the plane. Figure 3 shows theoretical TransOF fields as computed from Eq.1 for ideal 2D cameras with 120° FOV. Also indicated in dashed line on the center column graphs are the corresponding FOV of the actual 1D vision sensors. Those also have a FOV of 120° but only in one dimension. The FOV is divided into four regions for average OF estimation, as explained in Section III.

When the plane is moving toward a textured wall, the field is divergent with an amplitude inversely proportional to the distance, as shown in the right column of Figure 3. This information can be used to trigger a turning action in order to avoid the wall. However, TransOF is relatively small around the focus of expansion, in the center of the image. Therefore we suggest to take the difference between right-most and left-most regions as the criterion for triggering the turning action. This difference, which is a rough estimate of the OF divergence, is independent of the angle at which the plane approaches the wall, as can be seen by comparing the curves labelled OFDiv in the first and second row of Figure 3. Therefore, it can be used as a robust measure to trigger an avoidance behavior. The third row of Figure 3 illustrates the situation for a ventral camera. In this case, all vectors are oriented in the same direction and maximum amplitude is located in the center of the camera where the photoreceptors have the minimal distance from the ground. Since TransOF amplitude is inversely proportional to altitude,

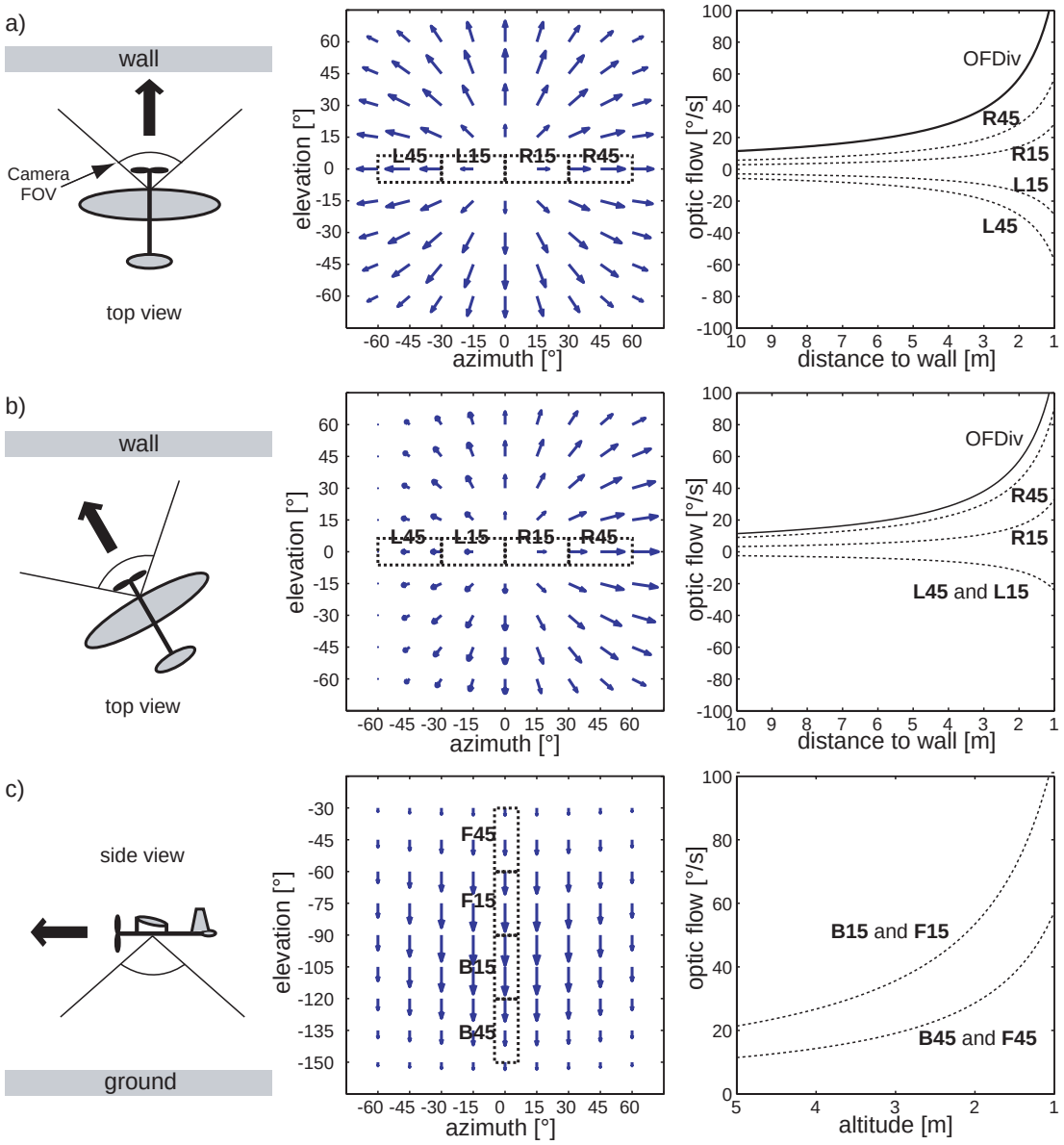

Fig. 3. Ideal patterns of OF as viewed from fictitious 2Dcameras with 120° FOV. a) Frontal approach toward a wall as viewed by a frontal camera. b) Same as previous but with an oblique angle of approach of 30°. c) Ventral camera looking downward and OF generated at different altitude. The first column depicts the camera position and orientation as well as the airplane trajectory. In these examples, the aircraft is flying at 2 m/s in pure forward translation. The second column gives snapshots of the typical OF patterns occurring in each situations. Indicated with dashed rectangles are the four regions corresponding to the 1D vision sensors used on the robots. The third column shows the OF amplitude as a function of the distance from the wall and from the ground at the center of each of the four sections (-45°, -15°, 15°, and 45°). The curves labelled OFDiv (for OF divergence) are simply the difference between right and left-most regions.

it is possible to control altitude of the plane by maintaining a given TransOF amplitude for the ventral camera. However, the locus of maximum TransOF corresponding the perpendicular distance from the ground depends on the pitch angle of the aircraft (not shown in Figure 3). Therefore, we suggest to take into account only the region where the TransOF is maximum, so that when the airplane is pitching (when going up and down) the minimal distance direction is automatically selected.

III. Minimalist optic-flow detector

Many methods for computing OF have been proposed [1], ranging from feature tracking to gradient- or energy-based schemes, but most of them are computationally demanding and therefore not suitable for a low-power microcontroller. Srinivasan proposed an image interpolation technique [13] that does not involve tracking of features, nor does it require measuring image velocity at a number of different locations. Instead, the parameters of global motion in a given region of the image can be estimated by a single-stage, non-iterative process, which interpolates the position of the moving image in relation to a set of reference images.

If $I(n)$ denotes the grey level of the n^{th} pixel in the 1D image array, the algorithm computes the amplitude of the translation s between an image captured at time t, $I_t(n)$, and

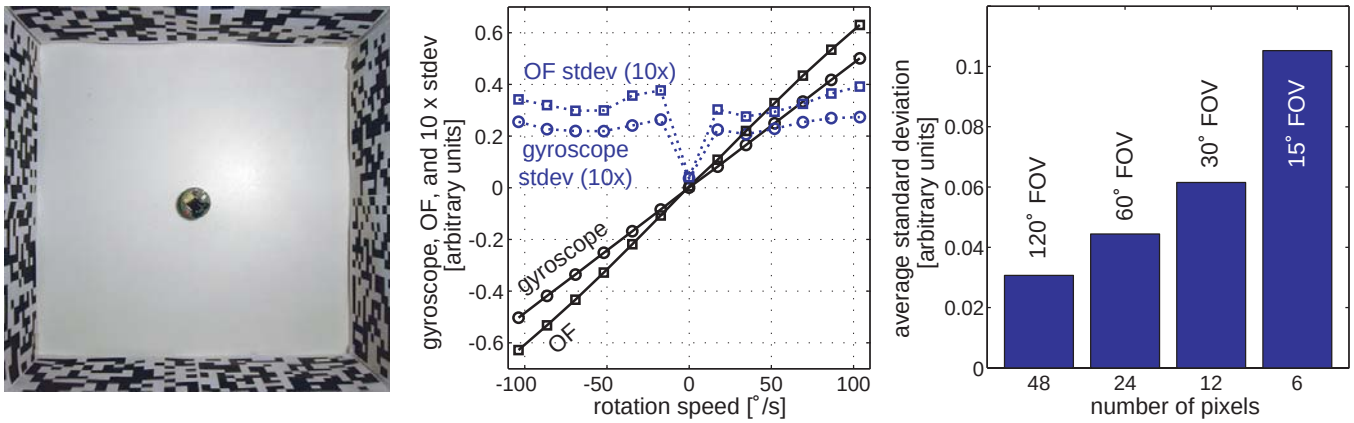

Fig. 4. *Left.* Test setup with the Khepera spinning in the middle of a 60 by 60cm arena with random black and white patterns. *Center.* Values read from the gyroscope (solid line with circles), related standard deviation of 1000 measurements for each rotation speed (dashed line with circles), and OF values estimated using 48 pixels (solid line with squares), related standard deviation (dashed line with squares). *Right.* Average standard deviation of OF for different size of FOV.

a later image captured at time $t+1$, $I_{t+1}(n)$. It assumes that, for small displacements of the image, $I_{t+1}(n)$ can be approximated by $\hat{I}_{t+1}(n)$, which is a weighted linear combination of the image $I_t(n)$ and two shifted versions $I_t(n \pm k)$ of the same image:

$$\hat{I}_{t+1}(n) = I_t(n) + s\frac{I_t(n-k) - I_t(n+k)}{2k}, \qquad (2)$$

where k is a small reference shift in pixels. The image velocity s is then estimated by minimizing the mean square error E between the estimated image $\hat{I}_{t+1}(n)$ and the actual one $I_{t+1}(n)$, with respect to s:

$$E = \sum_n \left[I_{t+1}(n) - \hat{I}_{t+1}(n)\right]^2, \text{ and } \frac{dE}{ds} = 0 \Leftrightarrow \qquad (3)$$

$$s = 2k\frac{\sum_n \left[I_{t+1}(n) - I_t(n)\right]\left[I_t(n-k) - I_t(n+k)\right]}{\sum_n \left[I_t(n-k) - I_t(n+k)\right]^2}. \qquad (4)$$

In our case, k has been set to 1 such that the reference images $I_t(n \pm k)$ are simply produced by shifting the acquired image by one pixel to the left and to the right, and s is taken as the estimation of the OF. Eq.4 has been implemented in an 8-bit microcontroller with no floating-point capabilities. At each sensory-motor cycle of the robot (lasting 45ms), two images corresponding to $I_t(n)$ and $I_{t+1}(n)$ are grabbed with an interval of 9.5ms. The image processing itself lasts 1.6ms.

In order to assess this algorithm, we put the Khepera robot equipped with a frontal 1D camera in the center of the arena (Figure 4). By letting it rotate on the spot, the resulting image motion is a pure translation. The obtained OF compares very well with the gyroscope value, which gives an information about the same movement. The center graph illustrates the perfect linearity of the OF values with respect to the rotation speed. More striking is the similarity of the standard deviations between the gyroscope measurements and OF estimates. This indicates that most of the noise, which is indeed very small, can be explained by mechanical vibrations of the Khepera (this is also why the standard deviation is almost null at 0°/s), and that OF is almost as good as the gyroscope at estimating rotational velocities. These measures support our earlier suggestion of cancelling RotOF by simply subtracting a scaled version of the gyroscope value from the global OF.

In order to estimate the OF in different subregions of the FOV (as required by the strategies described in Section II), Eq.4 must be applied several times to different groups of pixels. To assess the dependence of OF estimate accuracy on the region size, the same experiment as above was reproduced while varying the FOV and related number of pixels (see Figure 4, right). Based on those results, we chose a division into 4 regions (with 30° FOV and 12 pixels each, as outlined by dashed rectangles in Figure 3, center column) as a good trade-off between resolution and accuracy of OF estimates.

IV. Experimental results and conclusion

In this paper, we assume that the plane is able to fly almost straight and at constant speed. This can be achieved by passive roll stabilization with wing dihedral and active yaw control with a yaw gyroscope. Although it is not possible to exclude small corrections generating RotOF during straight sequences, the cancellation mechanism described above will remove spurious signals. For now, we also assume that it is feasible to pre-program a series of commands to let the aircraft engage a blind turn of about 90° without significantly altering its altitude and resume normal straight flight afterwards. These mechanisms are not tackled in this paper and we only concentrate on frontal obstacle detection and altitude control during straight sequences. Restricting the problem that way allows for testing the control mechanisms proposed in Section II on the wheeled robot (A) by letting it go straight forward at constant velocity and initiating pre-programmed turns whenever an obstacle is detected in front of it and (B) by implementing a wall following behavior, which is similar to altitude control (when looked at from above), where the heading direction of the Khepera stands for the pitch angle of the airplane. This is an approximation of what could happen in the air and it allows to assess the behavioral relevance of

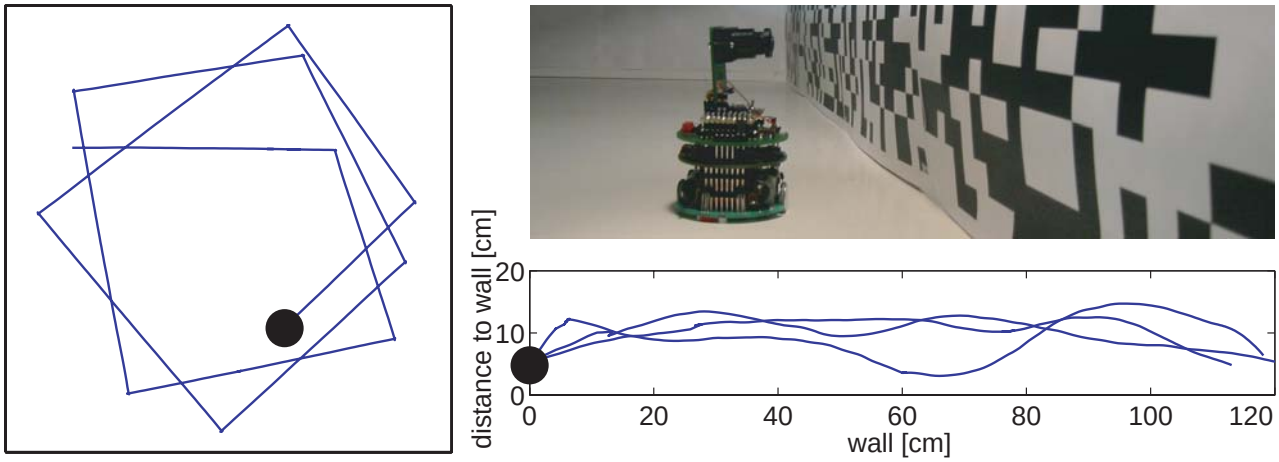

Fig. 5. *Left.* OF-based obstacle avoidance in the 60x60cm arena shown in Figure 4 (Khepera equipped with frontal camera). *Top right.* 120cm long setup for wall following experiments (Khepera with lateral camera). *Bottom right.* Wall following results (3 trials). The black circle indicates the robot initial position. Trajectories are reconstructed from odometry data.

the proposed control mechanisms and their feasibility in terms of hardware implementation and real-time requirements.

Figure 5 illustrates the results of these two experiments. In order to ensure a good similarity between winged and wheeled robots, we adapted the velocity and environment size of the Khepera. Since the speed and characteristic size of the environment are in a ratio of 8:1 for the airplane, we chose a ratio of 6:1 for the Khepera because of its simpler dynamics. These considerations led to an arena size of 60 by 60cm and a standard speed of 10cm/s for the Khepera. In experiment (A), the rotation direction of turning actions were determined by the asymmetry between left-most and right-most OF, i.e., the Khepera turned away from the largest OF value (see Figure 3, second row). This strategy is also supported by biological evidence [16]. The robot never crashed into a wall during 60'000 sensory-motor cycles (corresponding to about 45 minutes of continuous operation) where 84% of time was engaged in straight motion and the remaining 16% in turning actions. Furthermore, the algorithm is robust to change of illumination and of the patterns on the walls. In experiment (B), the wall following control (Figure 5, bottom right) has also proved to be robust, even though the robot does not always maintain the same distance from the wall. Nevertheless, these experiments show that a very simple controller can produce secure behavior in the sense of total collision avoidance.

In this paper, we showed vision-based obstacle avoidance and altitude control using very limited resources. The electronics used in these experiments, which are composed of a low-power 8-bit microcontroller, two sets of miniature 1D cameras and MEMS gyroscopes, weigh only 7g, which is within the payload of the 30-gram indoor slow-flyer.

Acknowledgements: The authors wish to thank C. Mattiussi for useful discussions and comments on the manuscript. Financial support was provided by the Swiss National Science Foundation.

REFERENCES

[1] J.L. Barron, D.J. Fleet, S.S. Beauchemin, and T.A. Burkitt. Performance of optical flow techniques. *International Journal of Computer Vision*, 92(1):236–242, 1994.

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

[3] N. Franceschini, J.M. Pichon, and C. Blanes. From insect vision to robot vision. *Philosophical Transactions of the Royal Society B*, 337:283–294, 1992.

[4] W.E. Green, P.Y. Oh, and G.L. Barrows. Flying insect inspired vision for autonomous aerial robot maneuvers in near-earth environments. In *Proceeding of the IEEE International Conference on Robotics and Automation*, 2004.

[5] J.J. Koenderink and A.J. van Doorn. Facts on optic flow. *Biological Cybernetics*, 56:247–254, 1987.

[6] M.A. Lewis. Visual navigation in a robot using zig-zag behavior. In *Neural Information Processing Systems 10*. MIT Press, 1998.

[7] F. Mura and N. Franceschini. Visual control of altitude and speed in a flying agent. In *From Animals to Animats III*, pages 91–99. MIT Press, 1994.

[8] G. Nalbach and R. Hengstenberg. The halteres of the blowfly calliphora - three-dimensional organization of compensatory reactions to real and simulated rotations. *Journal of Comparative Physiology A*, 175:695–708, 1994.

[9] T. Netter and N. Franceschini. A robotic aircraft that follows terrain using a neuromorphic eye. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2002.

[10] T.R. Neumann and H.H. Bülthoff. Behavior-oriented vision for biomimetic flight control. In *Proceedings of the EPSRC/BBSRC International Workshop on Biologically Inspired Robotics*, pages 196–203, 2002.

[11] J.D. Nicoud and J.C. Zufferey. Toward indoor flying robots. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 787–792, 2002.

[12] F. Ruffier and N. Franceschini. Visually guided micro-aerial vehicle : automatic take off, terrain following, landing and wind reaction. In *Proceeding of the IEEE International Conference on Robotics and Automation*, 2004.

[13] M.V. Srinivasan. An image-interpolation technique for the computation of optic flow and egomotion. *Biological Cybernetics*, 71:401–416, 1994.

[14] M.V. Srinivasan, J.S. Chahl, K. Weber, S. Venkatesh, and H. Zhang. Robot navigation inspired by principles of insect vision. In A. Zelinsky, editor, *Field and Service Robotics*, pages 12–16. Springer-Verlag, 1998.

[15] M.V. Srinivasan, S.W. Zhang, and J.S. Chahl. Landing strategies in honeybees, and possible applications to autonomous airborne vehicles. *The Biological Bulletin*, 200:216–221, 2001.

[16] L.F. Tammero and M.H. Dickinson. The influence of visual landscape on the free flight behavior of the fruit fly drosophila melanogaster. *The Journal of Experimental Biology*, 205:327–343, 2002.

[17] J.C. Zufferey, A. Beyeler, and D. Floreano. Vision-based navigation from wheels to wings. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2003.