I. Fantoni, G. Sanahuja
(Université de Technologie
de Compiegne)

E-mail: isabelle.fantoni@hds.utc.fr

DOI : 10.12762/2014.AL08-03

Optic Flow-Based Control and Navigation of Mini Aerial Vehicles

This paper concerns recent work on the application of optic flow for control and navigation of small unmanned aerial vehicles. Bio-inspired strategies, such as the use of optic flow, have always motivated researchers in the control community. Recent methodologies for active and passive navigation of aerial vehicles using optic flow, such as obstacle detection or terrain following, are presented. Applications to path following that have been achieved at Heudiasyc Laboratory (CNRS-UTC) in Compiègne are described, in order to illustrate the strength of the concept.

Introduction

Nowadays, we see an increased interest in mini and micro-UAVs. This growing activity in regard to small flying machines is motivated by recent advances in miniaturized electronics, but also by the increasing demand for military and civilian applications. Autonomous functions become more and more essential for UAVs and vision-based approaches offer undeniable assets for the autonomous navigation of such systems. In this paper, we will show the interest and the contribution of vision in the navigation of UAVs and especially the use of optic flow for control and navigation.

Ideally, in order to improve its autonomy, a vehicle has to perceive, model and interpret its environment to adapt its actions. In the same way as a man relies on his senses, an autonomous vehicle uses data provided by sensors. Among the sensory resources, we find proprioceptive sensors, which perform their actions in relation to what they locally perceive of the robot movement, or exteroceptive sensors, which are based on measurements taken in relation to their overall environment. The eye is indeed a perfect illustration of an exteroceptive sensor. Physiologically, vision is the ability that seems to give to living beings the most information about their environment. Motivated by work on insects, principles inspired on their behavior were developed in [5], [16] and have allowed natural ideas to solve localization issues, navigation, obstacle detection, etc., to be brought to robotics. Insects move and act according to exterior movements caused by their own movement. This visual characterization, called optic flow, is increasingly used for guidance assistance in robotics. We are particularly interested in the contribution of optic flow for UAVs navigation, which is the purpose of this article.

Optic flow

Definition

The optic flow is defined as the apparent motion of the image intensities (or brightness patterns) caused by the 2D projection onto a retina of the relative 3D motion of scene points. Ideally, it corresponds to the approximation of the velocity field in the image plane (or visual displacement field of image points) obtained by the 2D projection of the speed of moving objects in 3D space (geometric concept). This movement helps to explain variations in displacement of a moving picture. The optic flow is formulated as a vector field over two dimensions, where the domain is either the two-dimensional visual field or a focal plane and the range is a vector describing how fast and in which direction the texture is moving. The optic flow is created by the translational and the rotational movements of a point $P\ (X, Y, Z)$ in the camera reference frame. Considering the projection of this point in the image plane $p(x, y)$ (figure 1), we can express it according to point P and the focal length f.

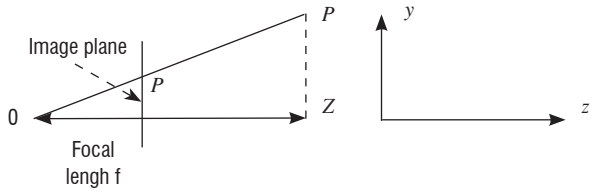


Figure 1 - Scheme of projective geometry for optic flow computing

$$\begin{bmatrix} x \\ y \end{bmatrix} = \frac{f}{Z}\begin{bmatrix} X \\ Y \end{bmatrix} \quad (1)$$

The velocity of p is found through differentiation of (1). Thus, the optic flow computed at an image point (x, y) can be expressed as a summation of translational and rotational parts, as follows:

$$\begin{bmatrix} OF_x \\ OF_y \end{bmatrix} = T_{OF} + R_{OF} \tag{2}$$

with the translational part

$$T_{OF} = \frac{1}{Z}\begin{bmatrix} -f & 0 & x \\ 0 & -f & y \end{bmatrix}\begin{bmatrix} V_x \\ V_y \\ V_z \end{bmatrix} \tag{3}$$

and the rotational part

$$R_{OF} = \begin{bmatrix} \frac{xy}{f} & -(f+\frac{x^2}{f}) & y \\ (f+\frac{y^2}{f}) & -\frac{xy}{f} & -x \end{bmatrix}\begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \tag{4}$$

where OF_j is the optic flow component in the coordinate j of the point p, V_k and ω_k are the translation velocities and rotation rates, respectively, of the point P at the coordinate k. This expression represents the optic flow defined on the image plane. Note that the optic flow can be defined on projection surfaces other than image planes, such as spheres, which are often used for their passivity properties [18]. Furthermore, relationships exist between different expressions of optic flow, so there is no information loss when the projection surface is changed.

Computation methods

Standard techniques for calculating optic flow can be classified into four groups: differential or gradient methods based on intensity [22], [24], [26], [35], correlation or block matching methods [1], [32], methods based on energy [19] and those based on phase [15], [37]. Block matching techniques present very good accuracy and performance against aperture problems (figure 2) and large displacements, but they are computationally expensive, less accurate in the presence of deformation and displacements of less than one pixel are not detectable. Energy-based and phase-based methods are also very expensive in terms of processing time and also very complex to implement. On the other hand, differential methods suffer from sensitivity issues due to changing lighting and noise due to the calculation of derivatives. They are nevertheless precise, with less computing time necessary compared with all cited techniques. Differential or gradient-based methods are thus well known, are the most implemented techniques because of their properties and are widely used in the literature for optic flow computations.

Despite their differences, many of the gradient-based techniques can be conceptually viewed in terms of a set of three processing stages:
1) pre-filtering or smoothing;
2) computation of spatiotemporal derivatives or local correlation surfaces
3) integration of these measurements to produce a two-dimensional flow field.

Differential methods are based on calculations of the derived spatiotemporal image intensity over a region of the image. The conservation of intensity is reflected by the following equation (see [4])

$$I(x,y,t) = I(x+dx, y+dy, t+dt) \tag{5}$$

where $I(x, y, t)$ is the intensity of the image, t is the time and (dx, dy) is the image displacement. From equation (5) it is possible to obtain the intensity conservation constraint, also called optic flow or gradient constraint equation [4], expressed by

$$\nabla I(x,y)(v_x, v_y) + I_t = 0 \tag{6}$$

where $\nabla I(x, y)$ is the spatial gradient, (v_x, v_y) is the image velocity and I_t is the temporal derivative of the image intensity. Giving this constraint, B. Lucas and T. Kanade [24] have constructed a technique for estimating the optic flow, based on a weighted least squares minimization of the intensity conservation constraint in each small spatial neighborhood S.

$$\min \sum_{(x,y)\in S} W^2(x,y)[\nabla I(x,y,t)\vec{v} + I_t(x,y,t)]^2 \tag{7}$$

where W is a window that gives more importance to the constraints near the center of the chosen neighborhood. The solution of equation (7) is given by

$$\vec{v} = [A^T W^2 A]^{-1} A^T W^2 b \tag{8}$$

where

$$A = [\nabla I(x_1, y_1), \ldots, \nabla I(x_n, y_n)]^T \tag{9}$$

$$W = diag[W(x_1, y_1), \ldots, W(x_n, y_n)] \tag{10}$$

$$b = -[I_t(x_1, y_1), \ldots, I_t(x_n, y_n)] \tag{11}$$

One advantage of the Lucas-Kanade algorithm is that it provides a measure of the estimation error, given that the matrix $[A^T W^2 A]^{-1}$ is consistent with a covariance matrix. Unreliable estimates can be identified using the inverse eigenvalues of this matrix. However, this method is not suitable for displacements larger than one pixel per frame, which causes estimation to fail. Nevertheless, an extension of this method has been proposed, resulting in a solution that computes optic flow via a hierarchical coarse-to-fine process, called pyramidal representation. This coarse-to-fine method is based on the Lucas-Kanade algorithm, which is complemented by a pseudo-iterative scheme that allows optic flow to be computed by propagating the flow in lower resolutions to the larger resolutions [9]. Despite all of the advantages of the Lucas-Kanade algorithm, it is important to remember that optic flow estimated by this method represents the apparent movement of objects in the scene, which may not correspond to real object movement. Indeed, a number of assumptions are made, in order to be able to compute optic flow. These hypotheses can be summarized as follows:

- Lighting must remain constant over time. This hypothesis is the basis of optic flow computation. Without it, equation (5) could not be written. If the lighting in the scene changes from one instant to another, objects may seem to move when in reality they are static over time. Indeed, shadows are different depending on the position of the light source and while the object is stationary the camera can detect shadow movement;
- Rich textured features are another important condition for accurate estimation of optic flow. For an effective differential approach, such as the Lucas-Kanade algorithm, there must be contrasts in the scene and thus contrasting textures must be chosen. Furthermore, in order to increase image contrasts, low pass filtering can be applied. There are two ways of achieving this filtering; one is to apply

a Gaussian filter to create a sort of blurring of the images. On the other hand, a simple way to apply such a filter is to de-focus the lens (the image becomes blurred). Nevertheless, this is prejudicial to the computation of the optic flow, since it requires a good texture but must not be too discontinuous;

• Finally, the well-known aperture problem must be taken into account. In general, a differential method alone does not solve what is called the aperture problem (figure 2). However, this entails increased computing time. One solution to this problem is to eliminate inconsistent points by means of the covariance matrix previously defined.

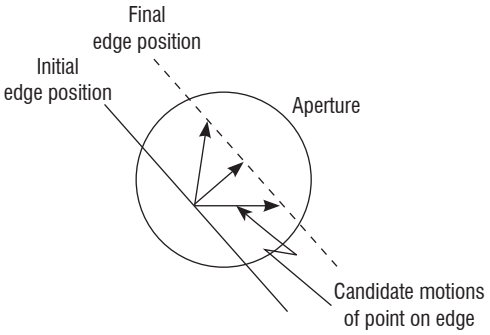


Figure 2 - Aperture problem

Navigation using optic flow

Active navigation

In order to ensure UAV autonomy with respect to unexpected changes in its environment, several approaches using optic flow can be found in the literature. Indeed, the main techniques developed have sought to provide the necessary information for the vehicle, enabling it to react against obstacles. Among reactive navigation modes using optic flow algorithms, we can cite: the effect of centering (Urban Canyon), wall monitoring, frontal obstacle detection and altitude centering for navigation within buildings.

Centering in a corridor

L. Muratet et al. [25] developed a centering strategy by equaling pixel motion calculated on the right and left of a helicopter. Optic flow rotational components were compensated using inertial unit (IMU) data. Indeed, at each point having an estimated optic flow, the rotational components were subtracted. These components were predicted using the camera projective model and the rotational speeds were calculated by the inertial measurement unit.

$$\overrightarrow{OF_i}^{rot} = f\begin{bmatrix} x_i y_i \omega_y - (1+x_i^2)\omega_z + y_i \omega_x \\ (1+y_i^2)\omega_y - x_i y_i \omega_z - x_i \omega_x \end{bmatrix} \quad (12)$$

where $\overrightarrow{OF_i}^{rot}$ is the optic flow rotational component, ω_k are rotational velocities around the axes and x_i and y_i are coordinates in the image plane. The optic flow resulting from the difference between the estimation and the prediction of the rotational component was considered as the optic flow due only to translational movement. The difference of the translational optic flows on either side was used to construct a proportional controller that provides the centering effect. Note that these methods that compensate for the rotational components require accurate calibration between the camera and the inertial measurement unit.

A. Argyros and F. Bergholm [2] achieved a centering effect in real time, for a mobile robot using a trinocular vision system: a camera pointing toward the front of the robot and two lateral cameras. The central camera was used to remove the optic flow rotational components calculated by peripheral cameras. The difference of both remaining optic flows was fed back, so as to produce a centering effect. Moreover, in [3], Argyros et al. used a panoramic camera to develop a control strategy for centering a mobile robot in a hallway. By using a coordinate transformation (polar to Cartesian) to find a set of cylindrical images, a large number of perspective images were approximated, while each column of pixels defined a steering angle. The central region was considered to be the one representing the angle $\theta = 0$, the optic flow was used to compensate offset components caused by the rotation of the robot. By taking symmetrical windows with respect to the central region, a centering effect was produced by performing a proportional control strategy using the optic flow calculated in each window. The embedded implementation was performed using two Pentium III (800 MHz) and Lucas-Kanade algorithm [24] for estimating optic flow.

S. Serres *et al.* [31] studied insects, in order to develop a stabilized binocular system comprising optic flow sensors instead of a standard camera system and an estimation algorithm. Each sensor had only two pixels and they were always aligned to the direction of displacement with gyro-compass servoing. Lateral flows measured by these sensors were then used in an optic flow regulator that ensured the stabilization of a hovercraft in the middle of a corridor. This strategy was only tested in simulation.

In Conroy *et al.* [14], a bio-inspired optic flow navigation system was implemented on a quad-rotorcraft and demonstrated in an indoor textured corridor. The inner-loop pitch and roll stabilization was accomplished using rate gyros and accelerometers. Altitude control was achieved via the fusion of sonar and accelerometer measurements. A ventrally located optic flow sensor from Centeye was used to increase the longitudinal and lateral damping, thus improving vehicle stability. An omnidirectional visual sensor, based on a Surveyor camera board and a parabolic mirror, was used for optic flow estimation and outer-loop control. The navigation strategy consisted in decomposing translational optic flow patterns (magnitude, phase, and asymmetry) with weighting functions, in order to extract signals that encoded relative proximity and speed, with respect to obstacles in the environment, which were used directly for outer-loop navigation feedback. The flight tests were performed in a textured corridor about 1.5m wide and 9m long. The quadrotor successfully avoided corridor walls and finished its course without collisions.

Obstacle detection

T. Camus used [10] a method for estimating optic flow in real time, in order to calculate the time-to-contact or time-to-collision of an obstacle. Time-to-contact is mathematically defined as the time until an object crosses an infinite plane defined by the image plane. T. Camus showed that taking a series of circles with different radii and center as the focus of image expansion, the time-to-contact may be calculated as the ratio between the distance between each point and the circle center, and the divergence from this point.

$$TTC = \frac{\sqrt{(x-x_{FOE})^2 + (y-y_{FOE})^2}}{\sqrt{V_x^2 + V_y^2}} \quad (13)$$

where (x_{FOE}, y_{FOE}) are the coordinates of the expansion focus (the point in the center of the horizon from which, when we are in motion, all points in the perspective image seem to emanate), V_x et V_y are the components of the velocity vector at point (x, y). The time-to-contact was calculated in each region defined by means of a circle of a given radius. The global result, in the entire image, was taken as the estimate of Least Squares of means for each region. The expansion focus position was estimated at any time from the average displacement of pixels. Indeed, T. Camus only considered full frontal obstacles, whose initial FOE estimate was the image center. The average of each coordinate of the optic flow vector was considered as an offset of the initial estimate.

In [25], L. Muratet et al. implemented, in a helicopter, a simplified method of this technique. They considered a fixed focus of expansion in the image center. They developed a control law that operated the opposite of this time-to-contact, commonly known as the relative depth.

J.-C. Zufferey and D. Floreano [38] implemented an obstacle avoidance technique for a 30g autonomous aerial vehicle. In this study, they were inspired by insects and optic flow sensors were used instead of the usual vision system. Optic flow sensors were composed of a 102 pixel vector and a microcontroller containing an estimation algorithm proposed by Srinivasan [33]. In order to compensate for the optic flow rotational components, sensors were first calibrated using a gyroscope. Since the output signal of these sensors was of the same order as that of the gyroscope, the measurement of the latter was directly used to subtract rotational components of the measured optic flow. The proposed obstacle avoidance method was only based on the optic flow divergence. When approaching an obstacle, the optic flow at the sensor extremes had a maximum amplitude and opposite directions. The difference of these two maximum speeds gave an estimate of the optic flow divergence, which is proportional to the inverse of the time-to-contact. This approach was justified by a theoretical study of areas of interest generated by calculating the optic flow, using a camera as it approached an obstacle. Experiments were conducted on a platform on the ground and on the aircraft for indoor flight.

A. Beyeler *et al.* [7] conducted a study on altitude control, considering it as a case of detection and obstacle avoidance. By applying the method developed previously by [38], they built a pseudo-centering method for a vehicle in indoor flight. The floor and roof were therefore taken as a corridor. The authors showed simulation results, but the algorithm was not experimentally tested.
A. Beyeler, J.-C. Zufferey and D. Floreano [8] described another optic-flow-based autopilot (optiPilot) for an outdoor fixed-wing MAV. The optiPilot was based on seven optic mouse sensors, MEMS rate gyroscopes and a pressure-based airspeed sensor. It was validated in simulations and demonstrated in real flights to avoid a group of tall trees (lateral avoidance) and small trees.

Passive navigation: terrain following and automatic landing

The passive navigation is the UAV ability to determine its own motion parameters with respect to the navigation reference. The estimation of these parameters is commonly called egomotion problem. However, there are new approaches used to estimate other important parameters for navigation, such as: altitude, direction of travel, pitch and roll angle, in order to perform terrain following, or altitude control and automatic landing.

J.-C. Zufferey *et al.* [38] used an altitude control using optic flow. They showed that an optic flow sensor pointing downward from the vehicle measured optic flow generated by the translational movement. A regulator allowed a constant optic flow to be maintained, which was equivalent to keeping the vehicle at a constant height. In [6], A. Beyeler *et al.* developed a similar strategy, but directly using raw (unprocessed) data from the optic flow sensor. By applying an optimization technique to the sensor raw data, the pitch angle and the altitude were estimated in real time, without needing to estimate any velocity vector.

The optic floww obtained from a camera pointing downwards may be used to perceive depth (ground). In fact, the optic flow model that appears for a 1D translational movement can be expressed as [29]

$$OF[^\circ] = \frac{V_x}{Z}\sin\alpha - \Omega_y \tag{14}$$

where Ω_y is the pitch angular velocity and α is the elevation angle. Since the optic flow rotational component Ω_y does not contain distance information, it is generally assumed to be equal to zero, or it is compensated for using inertial data. Most techniques based on optic flow for UAVs altitude control refer to equation (14).

The idea of terrain following is to fly at a fixed altitude above the ground by maintaining the optic floww at a constant value and then following the ground profile by regulating the optic flow [29], [12]. Maintaining a constant optic flow in (14) leads to an automatic reduction of the horizontal speed V_x when the height Z decreases. Thus, if the descent speed V_z is controlled proportionally to the forward speed, the descent angle remains constant.

Bees use this simple strategy (two rules) to ensure a smooth landing without explicit measurement or knowledge of their flight speed, or height above the ground [34]. Therefore, the height, the forward and descent speeds will exponentially decrease with time and become null at landing. This landing strategy inspired by insects has been demonstrated by many researchers [29], [13], [17].

Wagter and Mulder [36] constructed an algorithm for performing the 3D terrain reconstruction using optic flow. However, their system considered that the real rotation and translation speeds were known. Since the optic flow component created by the vehicle motion is inversely proportional to the depth and directly proportional to the translation speed, if the translation speed is known, the depth can be deduced. By using a camera pointing downward from an autonomous vehicle, the altitude can be estimated in real time so a reconstruction of the terrain can thus be achieved. The algorithm estimated the optic flow generated by the vehicle rotations from the gyro data, and this estimation was subtracted from the measured optic flow.

In Kendoul, Fantoni, and Nonami [23], the Structure From Motion (SFM) method (which deals with camera ego-motion estimation and the reconstruction of the 3D structure of the scene) was successfully applied for the state estimation and flight control of a small quadrotor system. The proposed implementation was based on three nested Kalman filters (3NKF) and involved real-time computation of the optic flow, fusion of visual and inertial data, and recovery of translational

motion parameters. The scale factor ambiguity was solved by using height measurements from a pressure sensor and assuming flat ground. The system was also augmented by a visual odometer that estimated the horizontal position by integrating image displacements. The developed system was first validated on a ground vehicle and then on a quadrotor system through vision-based autonomous flights, including automatic takeoff and landing, outdoor and indoor hovering, and velocity command tracking.

An optic flow-based terrain-following algorithm for mini quad-rotorcraft was proposed in Hérissé et al. [20]. The developed system computed optic flows at multiple observation points, obtained from two onboard cameras, using the *LK* algorithm and combined this information with forward speed measurements to estimate the height above the ground. A backstepping controller was used to regulate the height to some desired value. Indoor closed-loop flights were performed over a textured terrain using the CEA quadrotor vehicle flying at a forward speed of 0.3-0.4 m/s. The system was able to maintain a desired height of 1.5 m above a ramp and 2D corner textured terrain of about 4 m in length. In an extended work [21], the authors presented a nonlinear controller for a *VTOL UAV* that exploited a measured optic flow to enable hover and landing control on a moving platform. Their first objective concerned the stabilization of the vehicle relative to the moving platform that maintained a constant offset from a moving reference. The second concerned the regulation of automatic vertical landing onto a moving platform. Experimental results were provided for a quadrotor *UAV* to demonstrate the performance of the proposed control strategy.

In Sabiron *et al.* [30], a 6-pixel low-speed Visual Motion Sensor (VMS), inspired by insect visual systems, performed local 1-D angular speed measurements ranging from 1.5°/s to 25°/s. The sensor was tested under free-flying outdoor conditions over various fields, onboard an 80kg unmanned helicopter. The results showed that the optic flow measured closely matched the approximate ground-truth optic flow, despite the complex disturbances encountered. The sensor was also able to accurately sense low visual angular speeds, giving quite frequently refreshed measurements even at great heights over an unknown complex outdoor environment.

Application to path following

The strength of optic flow-based strategies for aerial vehicles has been extensively demonstrated through the work cited above. In the Heudiasyc laboratory, we have addressed the problem of hover flight and velocity regulation of a quad-rotorcraft along a followed path (line). This objective can be viewed as a first step for future work relating to road following, for traffic monitoring, or for power or railway line supervision.

Line following with velocity regulation

In [27] and [28], the problem of stabilizing the position and translational velocity of a quad-rotorcraft during autonomous flight along a road was considered. The proposed solution was based on a vision algorithm for line detection and optic flow computation. The algorithm used images provided by a monocular camera system embedded onboard the *UAV*. Such research involves two fundamental characteristics for any autonomous navigation system: we seek to accurately measure translational displacements and to eliminate the position drift when hovering. If the translational drift is correctly compensated, the hover flight can be used as an intermediary task between different flying behaviors, each suited to different conditions of the environment. Furthermore, velocity regulation was implemented to establish the different flying modes, such as lateral displacement and forward displacement actions. In order to make use of the optic flow in a very appropriate manner, a vision-based altitude controller was also developed. The combination of these three vision-based controllers (hover flight, velocity regulation and altitude stabilization) allowed the vehicle to navigate autonomously over a road model in a real-time application. Two different kinds of missions were tested: position hold over a road segment and road following at constant velocity. In the work [28], the road model was known (especially the width of the road); this allowed the altitude of the *UAV* above the road to be deduced. A diagram of the quad-rotorcraft above the road model can be seen in figure 3. The performance of the proposed methodologies was validated in a simulation environment. Real-time experimental applications with a quad-rotorcraft aerial vehicle, consisting of autonomous hover and forward flight at constant velocity were successfully achieved. Note that the experiments were achieved through a supervisory PC ground station, while the vision system was embedded on the *UAV*. The communication between the quad-rotorcraft and the ground station was provided by a wireless data link. The image processing computations were done on the ground station and sent to the *UAV*, where the low-level controllers were executed onboard the vehicle.

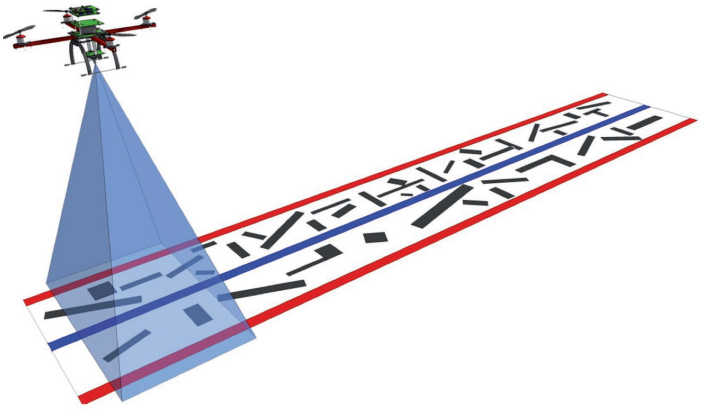


Figure 3 - Quad-Rotorcraft navigation: a diagram of the proposed setup presented in [28]

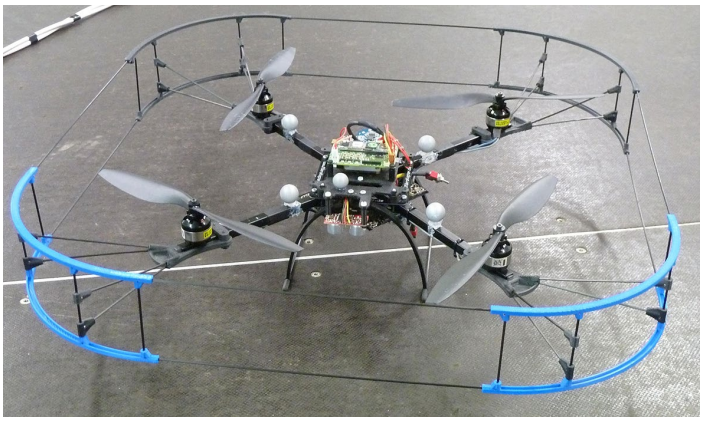


Figure 4: Experimental quadrotor designed and constructed at the Heudiasyc laboratory

Full autonomous line following

Heudiasyc quad-rotorcraft

The Heudiasyc laboratory at the University of Technology of Compiègne, in France, has a long experience with control and navigation of *UAVs*, especially using quadrotors. The latest version of our experimental quadrotor, shown in figure 4, is a *UAV* based on a Mikrokopter frame (with modifications). It has four brushless motors driven by *BLCTRLV2* controllers. The total weight is 1.2 kg, using an 11.1 V *LiPO* battery of 6000 mAh, giving about 15 minutes of flight time. The main electronic board is based on an *IGEP* module, equipped with a System On Chip (*SOC*) DM3730 from Texas Instruments. This *SOC* has one *ARM Cortex A8* core running at 1 GHz and one *DSP C64x+* core running at 800 MHz, which allows embedded image processing. The *ARM* processor allows Linux and its real-time extension Xenomai to be run. Thus, the control law runs in real time at 100 Hz. The *UAV* is also equipped with a Microstrain *3DMGX3-25 IMU* giving Euler angles and rotation speed measurements at 100Hz, a SRF10 ultrasonic range finder that provides the vehicle altitude at 50 Hz in a range between 0 and 2 meters and a PS3Eye camera capable of providing up to 120 images per second, at a resolution of 320x240 pixels. The camera points downwards, which allows the scene below the vehicle to be observed. Note that planarity of the scene is assumed in this work by the Heudiasyc laboratory. The images provided by the camera are processed by computer vision algorithms, in order to estimate the helicopter translational velocity in the x-y plane and the heading angle, as well as the lateral position w.r.t. the road direction. The translational velocity in the x-y plane is obtained from an optic flow algorithm, which is based on the pyramidal Lucas-Kanade method. For this purpose, the algorithm uses two pyramid levels, seeking up to 64 points of interest in the image. A Harris affine region detector was implemented to perform the characteristic feature detection. The DSP allows embedded calculation of the optic flow at about 100 Hz. Finally, the *UAV* is connected to the ground station via a WiFi connection.

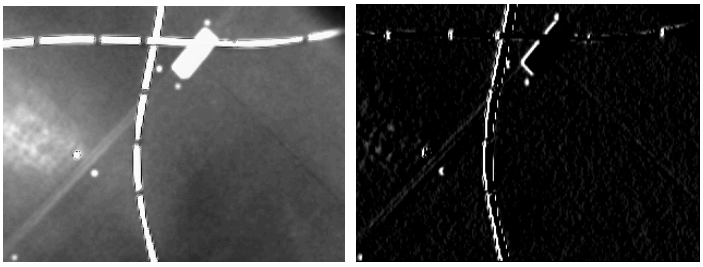
Line following

The strategy described here, is based on the work developed in [11]. A line is placed on the ground and the *UAV* follows it autonomously. This is done thanks to the downward looking embedded camera, which computes a Hough transform to detect lines. Thus, the *UAV* stabilizes itself over the line and can move forward using the optic flow to regulate its speed. Again, the embedded *DSP* is used for all image processing, running at 90 frames per second, to provide high speed measurements. The main contribution was to permit the quadrotor to navigate over a line in a completely autonomous manner with full embedded control and image processing. No external sensors or system (such as a motion capture system) are needed here, despite the presence of markers onboard the quadrotor (figure 4 of the quadrotor).

Indeed, we have used markers for other experiments where the knowledge of the vehicle position was needed. The present methodology has resulted in several improvements to the embedded control and image processing and a summary of real-time experiments that are currently being shown in the Heudiasyc Laboratory is presented here.

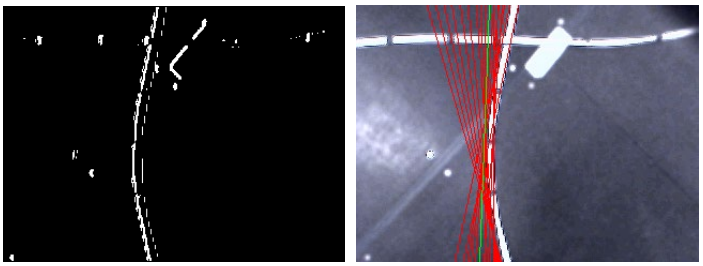
The quadrotor is able to follow a line without knowing its size, its color or its shape. We perform a demonstration with a closed 8-shape, where the quadrotor can continuously navigate over the path as shown in figure 5. A raw gray scale image obtained from the onboard camera while the vehicle is flying over the road is shown in Figure 6(a). The image obtained by edge detection using Sobel is shown in figure 6(b) and a binary image (black and white image) deduced by thresholding is presented in figure 6(c). Specifically, the grayscale image is used by the optic flow algorithm, while the black and white image is used by the line detection algorithm. A Hough transform computation allows vertical straight lines to be found in the image with an appropriate selection of the followed line. Finally, the post-processed image where the road has been detected is highlighted in green, as shown in Figure 6(d). Note that the Sobel detection is only performed in the x-axis direction to speed up image processing. Therefore, the heading angle of the vehicle is aligned

Figure 5 - Line following mission: the quadrotor flies over the line autonomously



a) Raw grayscale image

b) Sobel detection



c) Binary image

d) Processed image after Hough transform algorithm

Figure 6 - Image processing

with the direction of the line. Using imaging sensors, the lateral distance of the vehicle is then stabilized through the yaw angle, in order to navigate exactly over the road. Optic flow computation also allows the vehicle to navigate at a constant velocity over the path. Figure 7 shows optic flow vectors at points of interest and projected on one image. Figures 8(a) to 8(d) describe the consecutive images along the path. Since the overall objective of such an application would be

to supervise road traffic, power lines or railway infrastructure, the idea was also to consider cases when the line goes out the field of view of the downward pointing camera. Therefore, a switching control strategy was designed in order to consider both operational modes: the nominal case presented above, i.e., the camera allows the lateral position and the yaw angle to be stabilized; and the degraded mode, which is implemented in case the vehicle loses the line. In the nominal case, the forward speed is constant and regulated by optic flow. The degraded mode involves an algorithm integrating the lateral optic flow, in order to estimate the position and return to the line.

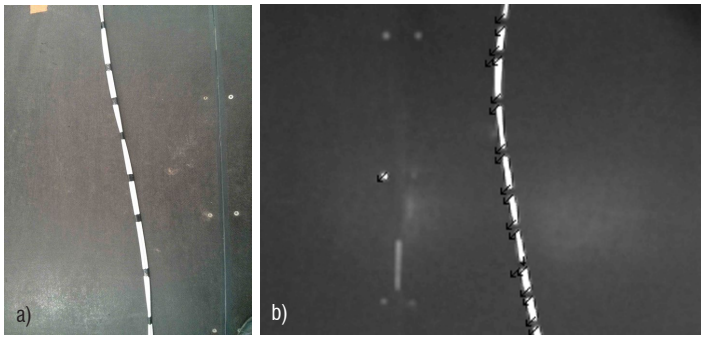


Figure 7 - Optic flow vectors from points of interest projected on one image along the line
a) Example of an original image with pieces of black tapes to improve Harris point detection, in order to avoid the aperture problem
b) Optic flow vectors projected on one image

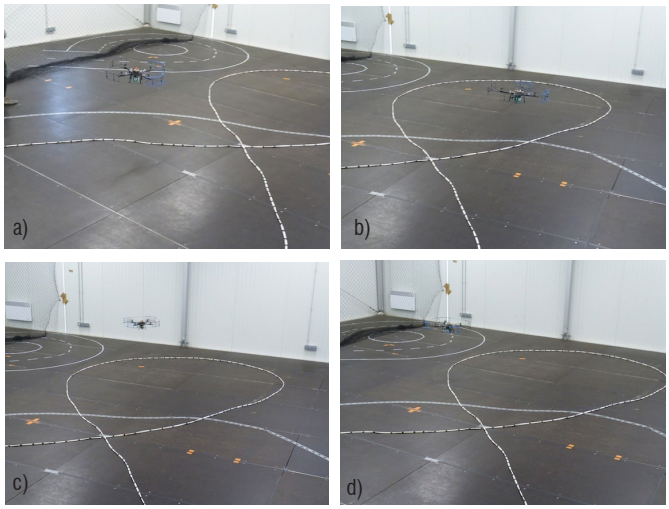


Figure 8 - Consecutive images along the 8-path

In this mode, the yaw angle is pointed using the IMU with the preceding value before the loss of the road detection. Figures 9(a)-9(c) demonstrate the degraded mode when the quadrotor is perturbed. A video of the demonstration can be viewed on the Heudiasyc *UAV* team web site: www.hds.utc.fr/uav-horus/platform/videos/.

Figure 9 - The quadrotor is moved from its initial path by a manual perturbation. It can retrieve the followed line

Conclusion

Significant progress has been made in the development of autonomous *UAVs*. Although there is often a gap between theoretical work and experimental developments, numerous reported experiments are appearing with validation in real time. Vision-based navigation, including bio-inspired strategies such as the use of optic flow, is a very active research topic. Numerous developments are arising, especially indoors. New technologies and extensive engineering work are also needed for outdoor implementations. We believe that bio-inspired visual control and navigation approaches for *UAVs* will increase in the future■

Acronims

VTOL	(Vertical Take-Off and Landing)
UAV	(Unmanned Aerial Vehicule)
IMU	(Inertial Measurement Unit)
FOE	(Focus Of Expansion)
MAV	(Micro Aerial Vehicule)
SFM	(Structure From Motion)
3NKF	(Three Nested Kalman Filter)
LK	(Lucas Kanade(algorithm))
VMS	(Visual Motion Sensor)
SOC	(System On Chip)

Acknowledgements

This work was carried out within the framework of the Labex MS2T (Reference ANR-11-IDEX-0004-02) and the Equipex ROBOTEX (Reference ANR-10-EQPX-44-01), which were funded by the French Government, through the program “Investments for the future” managed by the National Agency for Research. The second author is financed by the European Regional Development Fund.
The European Union is investing in your future.

References

[1] P. ANANDAN - *A Computational Framework and an Algorithm for the Measurement of Visual Motion*. International Journal of Computer Vision, 2(3):283–310, 1989.

[2] A. ARGYROS, F. BERGHOLM - *Combining Central and Peripheral Vision for Reactive Robot Navigation*. Computer Vision and Pattern Recognition Conference CVPR'99, pages 2646–2651, Fort Collins, Colorado, USA, June 23-25, 1999.

[3] A. ARGYROS, D.-P. TSAKIRIS, C. GROYER - *Biomimetic Centering Behavior for Mobile Robots with Panoramic Sensors*. IEEE Robotics and Automation Magazine, 11(4):21–68, 2004.

[4] J.-L. BARRON, D.-J. FLEET, S.-S. BEAUCHEMIN - *Performance of Optical Flow Techniques*. International Journal of Computer Vision, 12(1):43–77, 1994.

[5] G.-L. BARROWS, J.-S. CHAHL, M.-V. SRINIVASAN - *Biomimetic Visual Sensing and Flight Control*. Bristol UAVs Conference, Bristol, UK, 2002.

[6] A. BEYELER, C. MATTIUSSI, J.-C. ZUFFEREY, D. FLOREANO - *Vision-Based Altitude and Pitch Estimation for Ultra-Light Indoor Microflyers*. IEEE International Conference on Robotics and Automation, pages 2836–2841, Orlando, Florida, 2006.

[7] A. BEYELER, J.-C. ZUFFEREY, D. FLOREANO - *3D Vision-Based Navigation for Indoor Microflyers.* IEEE international conference on Robotics and Automation, pages 1336–1341, Roma, Italy, 2007.

[8] A. BEYELER, J.-C. ZUFFEREY, D. FLOREANO - *Vision-Based Control of Near-Obstacle Flight*. Autonomous Robots, 27:201–219, 2009.

[9] J.-Y. BOUGUET - *Pyramidal Implementation of the Lucas Kanade Feature Tracker.* Technical report, Intel Corporation, Microprocessor Research Labs, Technical report, 1999.

[10] T. CAMUS - *Calculating Time-to-Contact Using Real-Time Quantized Optical Flow*, 1995.

[11] L.R. GARCÍA CARRILLO, G. FLORES, G. SANAHUJA, R. LOZANO - *Quad Rotorcraft Switching Control: An Application for the Task of Path Following*. IEEE Transactions on Control Systems Technology, 22(4): 1255-1267, July 2014.

[12] J. CHAHL, A. MIZUTANI - *An Algorithm for Terrain Avoidance Using Optical Flow*. IEEE American Control Conference, pages 2742–2747, Minnesota, USA, 2006.

[13] J.-S. CHAHL, M.-V. SRINIVASAN, S.-W. ZHANG - *Landing Strategies in Honeybees and Applications to Uninhabited Airborne Vehicles*. International Journal of Robotics Research, 23(2):101–110, February 2004.

[14] J. CONROY, G. GREMILLION, B. RANGANATHAN, J.-S. HUMBERT - *Implementation of Wide-Field Integration of Optic Flow for Autonomous Quadrotor Navigation*. Autonomous Robots, 27:189–198, 2009.

[15] D.-J. FLEET, A.-D. JEPSON - *Computation of Component Image Velocity from Local Phase Information.* International Journal of Computer Vision, 5:77–104, 1990.

[16] N. FRANCESCHINI, J. PICHON, C. BLANES - *From insect Vision to Robot Vision*. Philosophical Transactions of the Royal Society of London B, 4(4):283–294, 1992.

[17] W.-E. GREEN, P.-Y. OH, G.-L. BARROWS - *Flying Insect Inspired Vision for Autonomous Aerial Robot Maneuvers in Near-Earth Environments.* IEEE International Conference on Robotics and Automation ICRA 2004, volume 3, pages 2347–2352, New Orleans, USA, April 2004.

[18] T. HAMEL, R. MAHONY - *Visual Servoing of an Under-Actuated Dynamic Rigid-Body System: An Imagebased Approach.* IEEE Transactions on Robotics and Automation, 18(2):187–198, 2002.

[19] D.-J. HEEGER - *Optical Flow Using Spatiotemporal Filters*. International Journal of Computer Vision, 1(4), 1988.

[20] B. HÉRISSÉ, T. HAMEL, R. MAHONY, F.-X. RUSSOTTO - *A Terrain-Following Control Approach for a VTOL Unmanned Aerial Vehicle Using Average Optical Flow.* Autonomous Robots, 29(3-4), 2010.

[21] B. HÉRISSÉ T. HAMEL, R. MAHONY, F.-X. RUSSOTTO - *Landing a VTOL Unmanned Aerial Vehicle on a Moving Platform Using Optical Flow.* IEEE Transactions on Robotics, 28(1), 2012.

[22] B.-K.-P. HORN, B.-G. SCHUNCK - *Determining optical Flow*. Artificial Intelligence, 17:185–204, 1981.

[23] F. KENDOUL, I. FANTONI, K. NONAMI - *Optic Flow-Based Vision System for Autonomous 3D Localization and Control of Small Aerial Vehicles.* Robotics and Autonomous Systems, 57:591–602, 2009.

[24] B. LUCAS, T. KANADE - *An Iterative Image Registration Technique with an Application to Stereo Vision*. Proceedings DARPA IU Workshop, pages 121–130, 1981.

[25] L. MURATET, S. DONCIEUX, Y. BRIRE, J.-A. MEYER - *A Contribution to Vision-Based Autonomous Helicopter Flight in Urban Environments*. Robotics and Autonomous Systems, 50(4):195–209, 2005.

[26] H.-H. NAGEL - *Displacement Vectors Derived from Second-Order Intensity Variations in Image Sequences*. Computer Vision, Graphics, and Image Processing, 21(1):85–117, 1983.

[27] E. RONDÓN, I. FANTONI-COICHOT, A. SANCHEZ, G. SANAHUJA - *Optical-Flow Based Controller for Reactive and Relative Navigation Dedicated to a Four Rotorcraft.* International Conference on Intelligent Robots and Systems (IROS), pages 684–689, St. Louis, USA, October 10-15 2009.

[28] E. RONDÓN, L.-R. GARCIA-CARRILLO, I. FANTONI - *Vision-Based Altitude, Position and Speed Regulation of a Quadrotor Rotorcraft*. International Conference on Intelligent Robots and Systems (IROS), pages 628–633, Taipei, Taiwan, October 2010.

[29] F. RUFFIER, N. FRANCESCHINI - *Optic Flow Regulation: the Key to Aircraft Automatic Guidance.* Elsevier, Robotics and Autonomous Systems, 50:177–194, 2005.

[30] G. SABIRON, P. CHAVENT, T. RAHARIJAONA, P. FABIANI, F. RUFFIER - *Low-Speed Optic-Flow Sensor Onboard an Unmanned Helicopter Flying Outside Over Fields.* Proc. IEEE International Conference on Robotics and Automation (ICRA), pages 1742–1749, Karlsruhe, Germany, 2013.

[31] J. SERRES, F. RUFFIER, S. VIOLLET, N. FRANCESCHINI - *Toward Optic Flow Regulation for Wall-Following and Centring Behaviours*. International Journal of Advanced Robotic Systems, 3(2):147–154, 2006.

[32] A. SINGH - *An Estimation-Theoretic Framework for Image-Flow Computation.* Proceedings of the third International Conference on Computer Vision ICCV, pages 168–177, Osaka, Japan, 1990.

[33] M.-V. SRINIVASAN - *An Image-Interpolation Technique for the Computation of Optic Flow and Egomotion*. Biological Cybernetics, 71:401–416, 1994.

[34] M.-V. SRINIVASAN, S.-W. ZHANG, J.-S. CHAHL - *Landing Strategies in Honeybees, and Possible Applications to Autonomous Airborne Vehicles.* Biological Bulletin, 200:216–221, April 2001.

[35] S. URAS, F. GIROSI, A. VERRI, V. TORRE - *A Computational Approach to Motion Perception.* Biological Cybernetics, 60(2):79–97, 1988.

[36] C. De Wagter, J.A. Mulder - *Towards Vision-Based UAV Situation Awareness.* AIAA Guidance, Navigation, and Control Conference, pages 1–16, San Francisco, California, 2005.

[37] A.-M. WAXMAN, S. ULLMAN - *Surface Structure and Three Dimensional Motion from Image Flow Kinematics*. International Journal of Robotics Research, 4(3):72–94, 1985.

[38] J.-C. ZUFFEREY, D. FLOREANO - *Toward 30-Gram Autonomous Indoor Aircraft: Vision-Based Obstacle Avoid-Ance and Altitude Control.* IEEE International Conference on Robotics and Automation, Barcelona, Spain, 2005.

AUTHORS

Isabelle Fantoni received the PhD degree, with the European label, in Non-linear Control for Underactuated Mechanical Systems, in 2000 from the “Université de Technologie de Compiègne”, in France. Since October 2001, she is a permanent Researcher at Heudiasyc laboratory, UTC, in Compiègne, France, employed by the French National Foundation for Scientific Research (CNRS) and CNRS Research Director since October 2013. Her research interests include non-linear control, modelling and control for Unmanned Aerial Vehicles (UAVs), fault-tolerant control for UAVs, vision for navigation of aerial vehicles, cooperation of UAVs, heterogeneous robotic systems in cooperation.

Guillaume Sanahuja has done his PhD thesis at Heudiasyc laboratory, Université de Technologie de Compiègne, in 2010 on the topic of vision for UAVs. He is now working in the same laboratory as a research engineer. His work is related to the French project Equipex (equipment of excellence) ROBOTEX, which aim is to equip Heudiasyc and its partners with robotic platforms. His function is to design and develop embedded architectures at both software and hardware levels. His topics of interests are embedded systems, embedded vision, non-linear control.