

# Unmanned aerial vehicles UAVs attitude, height, motion estimation and control using visual systems

Iván F. Mondragón · Miguel A. Olivares-Méndez · Pascual Campoy · Carol Martínez · Luís Mejías

Received: 8 February 2009 / Accepted: 18 March 2010 / Published online: 10 April 2010  
© Springer Science+Business Media, LLC 2010

**Abstract** This paper presents an implementation of an aircraft pose and motion estimator using visual systems as the principal sensor for controlling an Unmanned Aerial Vehicle (UAV) or as a redundant system for an Inertial Measure Unit (IMU) and gyros sensors. First, we explore the applications of the unified theory for central catadioptric cameras for attitude and heading estimation, explaining how the skyline is projected on the catadioptric image and how it is segmented and used to calculate the UAV's attitude. Then we use appearance images to obtain a visual compass, and we calculate the relative rotation and heading of the aerial vehicle. Additionally, we show the use of a stereo system to calculate the aircraft height and to measure the UAV's motion. Finally, we present a visual tracking system based on Fuzzy controllers working in both a UAV and a camera pan and tilt platform. Every part is tested using the UAV COLIBRI platform to validate the different approaches, which include comparison of the estimated data with the inertial values measured onboard the helicopter platform and the validation of the tracking schemes on real flights.

**Keywords** Omnidirectional images · Catadioptric systems · Stereo vision · Unmanned aerial vehicles (UAV) · Motion estimation · Fuzzy control

---

I.F. Mondragón (✉) · M.A. Olivares-Méndez · P. Campoy · C. Martínez  
Computer Vision Group U.P.M., C/José Gutiérrez Abascal 2,  
28006 Madrid, Spain  
e-mail: [imondragon@etsii.upm.es](mailto:imondragon@etsii.upm.es)  
url: <http://www.disam.upm.es/vision/>

L. Mejías  
Australian Research Centre for Aerospace Automation  
(ARCAA), School of Engineering Systems,  
Queensland University of Technology, GPO Box 2434, Brisbane,  
4000 Australia

## 1 Introduction

The main objective of this paper is to demonstrate that computer vision can be successfully used in several control loops onboard Unmanned Aerial Vehicles (UAVs) and not only as a sensing mechanism for providing visual information. For this purpose, fast and robust image processing algorithms are required in order to be able to close the control loops in near real time. A processing rate near 15 frames per second has been demonstrated to suffice, consistent with human pilots' reaction times. A general image processing algorithm that fully recognizes the environment of a UAV is currently very far from being realistic. In this paper, we demonstrate, however, that some classical image acquisition techniques can be integrated with ad-hoc image processing and with fuzzy controllers in order to achieve UAV navigation based on visual information.

Visual information in the control loop has the main advantage of providing information on UAV position/ attitude, relative to objects within the UAV visual field. This information can successfully complement and in some cases replace other classical sensors for UAV navigation such as GPS and IMU, which provide a very different type of information, i.e. absolute coordinates that are not related to the position of nearby objects. Visual information is crucial in some cases, such as detecting and avoiding static structures (Beyeler et al. 2009) and other flying objects (Carnie et al. 2006), or maneuvering close to fixed objects (e.g. visual inspection, spacial landings, flying close to outdoors structures) (Hrabar et al. 2005; Mejías et al. 2006) and in cases where the GPS is missing (e.g. indoors, on paths among buildings or other structures, and on space missions).

The idea of using visual information for UAV attitude estimation is not new. The first experiments attempted to obtain the skyline from images taken by a single perspective

camera looking forward on the aircraft, using this to estimate the roll angle with a horizontal reference (Ettinger et al. 2002; Todorovic et al. 2003; Cornall et al. 2006; Cornall and Egan 2004; Dusha et al. 2007). These works differs in the way that they segment the sky and ground and in how they estimate the horizon line. Todorovic et al. (2003) treat the horizon detection problem as image segmentation and object recognition applying statistical appearance models based on both color and texture clues. They make a Bayesian segmentation based on a statistical framework employing a Hidden Markov Tree on the appearance models. Cornall et al. (2006) use a technique focused on be implemented on a small microcontroller. The algorithm uses a simplistic method where the horizon is approximated as a threshold of the blue color plane, determining the optimal threshold by Otsu's method and Kmeans. This approximation gives good results under clear sky conditions. Dusha et al. apply morphological image processing and the Hough transform to detect the horizon line and derive the roll and pitch angles. In addition, they use optical flow to obtain the bodyframe rates.

Omnidirectional vision has also been used for UAV control and attitude estimation. Hrabar and Sukhatme (2003) use an omnidirectional system for sideways looking sensing on an autonomous helicopter by applying image unwrapping. Conroy et al. (2009) use spatial decompositions of the instantaneous optic flow to extract local proximity information from catadioptric images obtained onboard a micro-air-vehicle (MAV) for corridor-like environment navigation. Demoncaux et al. (2006), use a similar approach to the one presented in this paper, showing the advantages of using omnidirectional rather than perspective images for attitude estimation. They detect the horizon line on the catadioptric image using a Markov Random Fields segmentation and then project it on the equivalent sphere projection model for a Catadioptric system (Geyer and Daniilidis 2000) to obtain the attitude angles of the camera frame, which are related to the normal vector of the projected horizon line. Experimental results on video sequences demonstrated good performance; however, they do not make any comparison with ground truth data and also do not define the computational cost and the feasibility of a real-time implementation, both conditions that are necessary for use as a controller.

Motion estimation from imagery is often referred to as visual odometry (Matthies 1989). It determines the vehicle's position and orientation by detecting and tracking salient points using an onboard camera. This technique has been traditionally used in ground vehicles to improve robot localization. The use of visual odometry techniques has been documented in different situations such as, Mars rovers (Corke et al. 2004), and (Cheng et al. 2006). It has been tested on several ground autonomous robots (Nistér et al. 2006) and on a small shrimp robot (Milella and Siegwart

2006). Recently, a Visual Odometer have been implemented in a small rotorcraft UAV (Kendoul et al. 2009), by identifying and tracking visual features in the environment to obtain the optic flow and fusing it with the inertial measurements to determine the position and velocity of the aircraft. Catadioptric systems have also been used for robot odometry and relative orientation in outdoor vehicles (Labrosse 2006; Scaramuzza and Siegwart 2008), showing that it is possible to estimate relative orientation and the position of a mobile robot using panoramic images.

Object detection and tracking are common problems in vision systems in all of the robotics platforms. On the one hand, the use of UAVs enlarges the possible applications by increasing the workspace size, covering a bigger area for inspection, detection and tracking. On the other hand, they increase the problem of visual servoing, due to the difficulties in aircraft stabilization and control when we are trying to track an object. There are different visual algorithms tackling these problems on UAVs, some of which are presented in Campoy et al. (2008). Currently, some applications have been developed, including Valavanis' work on traffic monitoring (Puri et al. 2007), and fire detection (Nikolos et al. 2004) and work in obstacles avoidance and path planning for UAVs by Hrabar and Sukhatme (2009).

In this paper we propose an omnidirectional-based system for attitude estimation and a stereo- based system for height and motion estimation. We also provide visual information to a Fuzzy control system, which is specially trained for tracking moving objects and controlling the UAV heading. The position and attitude calculated by our vision systems are compared with those provided by the IMU/GPS system in Sects. 3 and 4, where there is shown an overall coherence that make them suitable for complementing those sensor information and even for replacing them, as it is demonstrated in following Sect. 5. The whole system has been tested in real flights in our COLIBRI-III prototype, showing the good results presented in Sects. 3.5, 4.3 and 5.3.

## 2 UAV system description

The Colibri project has three operational UAV platforms: one electric helicopter and two gasoline-powered helicopters (Fig. 1). The COLIBRI testbeds (Campoy et al. 2008), are equipped with an Xscale-based flight computer augmented with sensors (GPS, IMU, Magnetometer, fused with a Kalman filter for state estimation). Additionally they includes a pan and tilt servo-controlled platform for many different cameras and sensors. In order to enable it to perform vision processing, it also has a VIA mini-ITX 1.5 GHz onboard computer with 2 GB RAM, a wireless interface, and support for many Firewire cameras including Mono (BW), RAW Bayer, color, and stereo heads. It is possible to use IP and analog cameras as well.



**Fig. 1** Up: COLIBRI III Electric helicopter with a stereo camera system. Down: COLIBRI I Gas power helicopter

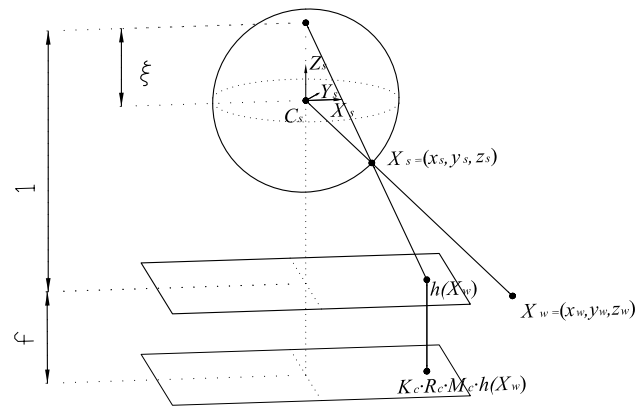
The system runs in a client-server architecture using TCP/UDP messages. The computers run Linux OS working in a multi-client wireless 802.11g ad-hoc network, allowing the integration of vision systems and visual tasks with flight control. This architecture allows embedded applications to run onboard the autonomous helicopter while it interacts with external processes through a high level switching layer. The visual control system and additional external processes are also integrated with the flight control through this layer using TCP/UDP messages. The layer is based on a communications API where all messages and data types are defined. The helicopter's low-level controller is based on PID control loops to ensure its stability. The higher level controller uses various sensing mechanisms such as GPS and/or vision to perform tasks such as navigation, landing, and visual tracking.

### 3 UAV attitude estimation using catadioptric systems

This section explains the use of omnidirectional images to obtain attitude information for a UAV, working as a RAW sensor data. First, we introduce the fundamentals of a catadioptric system and the unified model; second, we explain how the skyline is projected on the image and how it is segmented; then we described how it is used to estimate the attitude parameters, and finally we introduce the use of appearance images and its use as a visual compass.

#### 3.1 Central catadioptric cameras

Catadioptric cameras are devices that combine reflective elements (catoptric) and refractive systems (dioptric) to form



**Fig. 2** Catadioptric projection modeled by the unit sphere

a projection onto the image plane of the camera. They can be classified as central and non-central catadioptric cameras according to the single effective viewpoint criteria. Baker and Nayar (1997, 1999), define the configurations that satisfy the constraints of a single viewpoint, finding that a central catadioptric system can be built by combining a perspective camera with a hyperbolic, elliptical or planar mirror, or using an orthographic camera with a parabolic mirror.

Geyer and Daniilidis (2000, 2001) propose a unified model for the projective geometry induced by central catadioptric systems, showing that these projections are isomorphic to a projective mapping from a sphere (centered on the effective viewpoint) to a plane with the projection centered on the perpendicular axis to the plane.

A modified version of this unified model is presented by Barreto and Araujo (2001, 2002), where the mapping between points in the 3D world and points in the catadioptric image plane is split into three steps. First, a linear function maps the world into an oriented projective plane. Then a non-linear function transforms points between two oriented projective planes. Finally, there is a collineation function depending on the mirror parameters and the camera calibration matrix (intrinsic parameters). Figure 2 shows the general unit sphere projection for modeling catadioptric systems.

Consider a point in space (visible to the catadioptric system), with Cartesian coordinates  $X_w = (x_w, y_w, z_w)^T$  in the catadioptric reference (focus). This point is mapped into point  $X_s = (x_s, y_s, z_s)^T$  on the unitary sphere centered on the effective view point by (1):

$$X_s = \frac{X_w}{\sqrt{x_w^2 + y_w^2 + z_w^2}} = \frac{X_w}{\|X_w\|}. \quad (1)$$

To each projective point  $X_s$ , corresponds a projective point  $X_c = (x_c, y_c, z_c)^T$  in a coordinate system with origin at the camera projection center. This projection is a non-linear mapping between two projective planes and is defined

**Table 1** Parameters  $\xi$  and  $\psi$  for central catadioptric systems where  $d$  is distance between focus and  $4p$  is the Latus Rectum (the chord perpendicular to the principal axis and passing through a focus of an ellipse, parabola, or hyperbola)

	Parabolic	Hyperbolic	Elliptical	Planar
$\xi$	1	$\frac{d}{\sqrt{d^2+4p^2}}$	$\frac{d}{\sqrt{d^2+4p^2}}$	0
$\psi$	$1+2p$	$\frac{d+2p}{\sqrt{d^2+4p^2}}$	$\frac{d-2p}{\sqrt{d^2+4p^2}}$	1

by (2):

$$X_c = (x_c, y_c, z_c)^T = M_c \cdot \tilde{h}(X_w) \quad (2)$$

where

$$M_c = \begin{bmatrix} \psi - \xi & 0 & 0 \\ 0 & \xi - \psi & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

$$\tilde{h}(X_w) = \left( x_w, y_w, z_w + \xi \sqrt{x_w^2 + y_w^2 + z_w^2} \right)^T$$

where the matrix  $M_c$  depends on the mirror parameters  $\xi$  and  $\psi$ , defined for each one of the central catadioptric projections, as shown in Table 1.

Finally, the image in the catadioptric plane is obtained after a collineation between the image and the projective plane, depending on the camera's intrinsic parameters  $K_c$  (where  $m_x$  and  $m_y$  are the pixel per unit distance in image coordinates,  $f$  is the focal length, and  $(x_0, y_0)$  is the principal point), and the rotation of the camera is  $R_c$ . The projection of a world point on the catadioptric image is defined by (3):

$$\begin{aligned} H_c &= K_c \cdot R_c \cdot M_c, \\ X_i &= H_c \cdot \tilde{h}(X_w), \\ K_c &= \begin{bmatrix} fm_x & s & x_0 \\ 0 & fm_y & y_0 \\ 0 & 0 & 1 \end{bmatrix}. \end{aligned} \quad (3)$$

Function  $\tilde{h}(X_w)$  is a homogeneous positive injective with an inverse function  $\tilde{h}^{-1}(X_w)$ . This function maps points in a projective plane onto the unitary sphere. The non-linear inverse function is defined by

$$(x_s, y_s, z_s)^t = \tilde{h}^{-1}(H_c^{-1} X_i) = (\lambda_c x_c, \lambda_c y_c, \lambda_c z_c - \xi)^t \quad (4)$$

where

$$\lambda_c = \frac{z_c \xi + \sqrt{z_c^2 + (1 - \xi^2)(x_c^2 + y_c^2)}}{x_c^2 + y_c^2 + z_c^2}.$$

### 3.2 Skyline and catadioptric image

To be able to measure the body frame attitude based on the catadioptric image, it is necessary to know how the skyline is

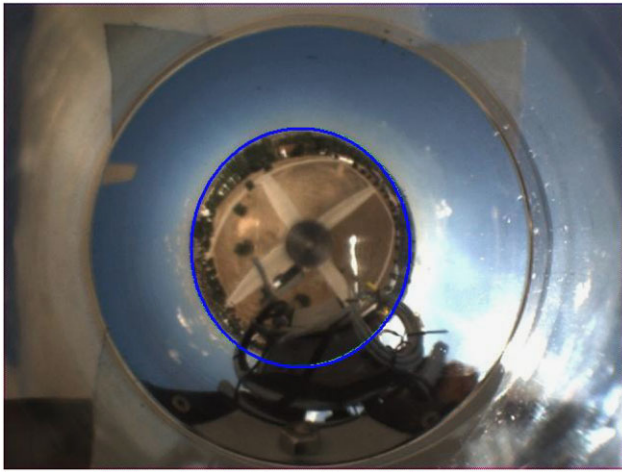


**Fig. 3** Skyline is the occluding contour of the earth sphere surface, whose projection on the equivalent unitary sphere model through a plane that intersects it forms the depicted red circle

projected onto the unitary sphere and onto the catadioptric plane. Ying and Hu (2004) demonstrate that the occluding contour of a sphere in space is projected onto a circle on the unit sphere or onto a conic in the catadioptric image plane. Considering the skyline the occluding contour on the earth sphere surface, finding it requires to look for a small circle on the unitary sphere model or for a conic or ellipse on the image plane, as proposed by Demonceaux et al. (2006) (see Fig. 3 for an illustration).

Because the original datum obtained is the image projection, the skyline detection focuses on isolating the sky from the ground in this image and then estimating the best adjusted ellipse to the skyline.

To isolate the sky from the ground we use an approach based on the method employed by Cornall et al. (2006) in which the *RGB* components of each pixel are weighted using the function  $f(RGB) = 3B^2/(R+G+B)$ . This function has shown very good results to sky-ground segmentation under different light and cloud conditions. For each resulting grayscale image from function  $f(RGB)$ , a Pyramid Segmentation (Antonisse 1982) followed by a Gaussian Adaptive Threshold function, is used to obtain a sky-ground binary image. The pyramid segmentation allows us to reduce the effects of sunlight or brightness in the image under variable lighting conditions. This threshold method is very fast and produces good results in real-time.



**Fig. 4** The best fitted ellipse (blue) to the skyline on an original catadioptric image obtained during a UAV flight

Once we have a sky-ground thresholded image, the ground contour on the image can be easily defined. This contour represents the skyline and is used by a fitting function to obtain the ellipse with the best approximation to the contour. Figure 4 shows an example of the best fitted ellipse to the skyline on an original catadioptric image obtained during a UAV flight test.

### 3.3 Pitch and roll estimation using the skyline backprojection on a sphere

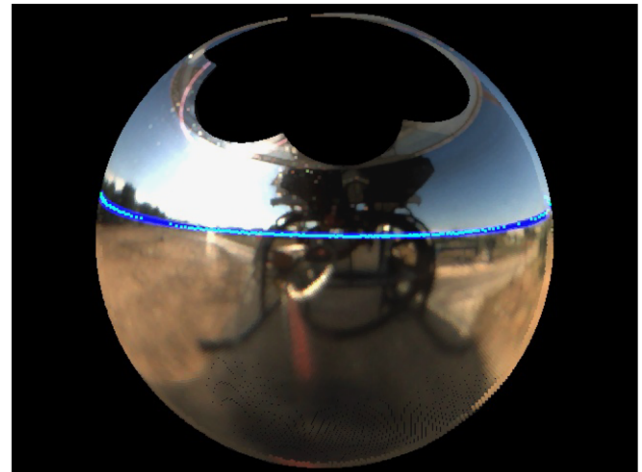
The segmented skyline is defined by the points contours that represent the ground border or the adjusted ellipse points  $SKY_{img} = (x_{SKY_{img}}, y_{SKY_{img}}, 1)$ . These points are backprojected onto the unitary sphere using (4) obtaining  $SKY_s = (x_{SKY_s}, y_{SKY_s}, z_{SKY_s})$  as shown in Fig. 5. The circle formed by the skyline points on the sphere forms a plane that intersects with the unitary sphere. To obtain the parameters of the skyline in the unitary sphere, it is sufficient to find the plane with normal equation  $N_x x_{SKY_s} + N_y y_{SKY_s} + N_z z_{SKY_s} + D = 0$  that best adjusts to the backprojected points of the skyline contour on the sphere surface.

For each point of the backprojected skyline, the normal equation of the plane is obtained by  $z_{SKY_s}^i = N_x x_{SKY_s}^i + N_y y_{SKY_s}^i + D$  with  $i = 1, \dots, n$  and an overdetermined linear system of the form  $(A \cdot x = b)$  is solved using the *pseudo-inverse method* to obtain the plane  $\pi_{sky} = (N_x, N_y, 1, D)$  (see (5)):

$$[N_x, N_y, D]^T = \arg \min_x \|A \cdot x - b\| \quad (5)$$

where

$$A = \begin{bmatrix} x_{SKY_s}^1 & y_{SKY_s}^1 & 1 \\ \vdots & \vdots & \vdots \\ x_{SKY_s}^n & y_{SKY_s}^n & 1 \end{bmatrix},$$



**Fig. 5** The best fitted ellipse (blue) to the skyline is backprojected on a unitary sphere model, forming a plane that intersects the sphere (which forms a *small circle*). The normal vector to this plane defines the attitude of the camera and the UAV

$$x = \begin{bmatrix} N_x \\ N_y \\ D \end{bmatrix},$$

$$b = \begin{bmatrix} z_{SKY_s}^1 \\ \vdots \\ z_{SKY_s}^n \end{bmatrix}.$$

The normal vector to the plane formed by the skyline and the unitary sphere is defined as  $N = [N_x, N_y, 1]^T$ . Assuming that the camera frame is aligned with the UAV frame, so that the  $x$  axis is the heading of the UAV and the  $y$  axis is aligned with the UAV wing, it is easy to obtain the desired *roll* ( $\phi$ ) and *pitch* ( $\theta$ ) angles, using (6):

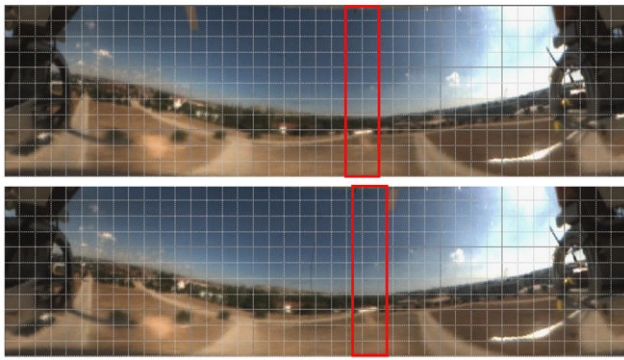
$$\theta = \arccos \left( \frac{N_x}{\sqrt{N_x^2 + N_y^2 + 1}} \right), \quad (6)$$

$$\phi = \arccos \left( \frac{N_y}{\sqrt{N_x^2 + N_y^2 + 1}} \right).$$

### 3.4 Yaw estimation using a visual compass

The relative heading of the UAV is calculated using the so called Appearance Images. This method was used by Labrosse (2006) and later by Scaramuzza and Siegwart (2008). It consists of a part of a panoramic image  $I(\alpha, R)$  obtained from a catadioptric image  $I_c$  (original captured image) using a polar to Cartesian coordinates change or unwrapping process employing (7):

$$I(\alpha, R) = I_c(R \cos(\alpha) + u_0, R \sin(\alpha) + v_0) \quad (7)$$



**Fig. 6** Two consecutive appearance images with a small rotation between them. A white grid is superimposed as reference and the red box shows clearly the column shift between images

where  $(u_0, v_0)$  is the catadioptric image center,  $\alpha$  is a linear function with maximum range  $[0, 2\pi]$  and  $R$  is a linear function that scans along the image radius. If the catadioptric image corresponds to a scene captured with an almost perfect vertical camera to the ground, then pure rotation will be appear on the appearance image as a pure pixel column-wise shift. The relative rotation between two consecutive images is obtained, by finding the best match based on the images' column shift using the Euclidean distance. Equation (8) shows the Euclidean distance between two panoramic images  $I_m$  and  $I_n$  with the same size and space color as a function of the column-wise shift on the image  $I_n$  by  $\alpha$  pixels (horizontal rotation). Figure 6 shows two consecutive appearance images obtained by an unwrapping process with a small rotation.

$$d(I_m, I_n, \alpha) = \sqrt{\sum_{i=1}^W \sum_{j=1}^H \sum_{k=1}^C (I_m(i, j, k) - I_n(i + \alpha, j, k))^2}. \quad (8)$$

The best shift  $\alpha_{\min}$  that minimize the distance function  $d(I_m, I_n, \alpha_{\min}) \leq d(I_m, I_n, \alpha) \forall \alpha \in \mathbb{R}$  is the best pixel rotation between this two images.

The rotation angle or yaw  $\psi$  between images is directly related to the obtained column shift between images, considering only the angular resolution of the appearance images defined by the images field of view FOV and the images' width, as shown in (9):

$$\psi_{(I_m, I_n)} = \alpha_{\min} \frac{FOV}{imgWidth}. \quad (9)$$

To obtain the final rotation relative to the first image, it is necessary to add the obtained value to a counter.

Equation (8) was developed under the assumption of a pure camera rotation between consecutive images on its vertical axis, which is perpendicular to the horizontal plane. In

the general case, the UAV has translational components and roll and pitch variations, causing the camera vertical axis to not always be perpendicular to the horizon. However, as shown by Labrosse (2006) and by Scaramuza and Siegwart (2008) the visual compass method based on appearance images is still valid under translation and attitude variations if the camera has small displacements or the distance to the objects is large compared with the displacement. Because images are captured continually, small variations of pitch and roll are present between consecutive images; therefore, the pure rotation assumption is still valid.

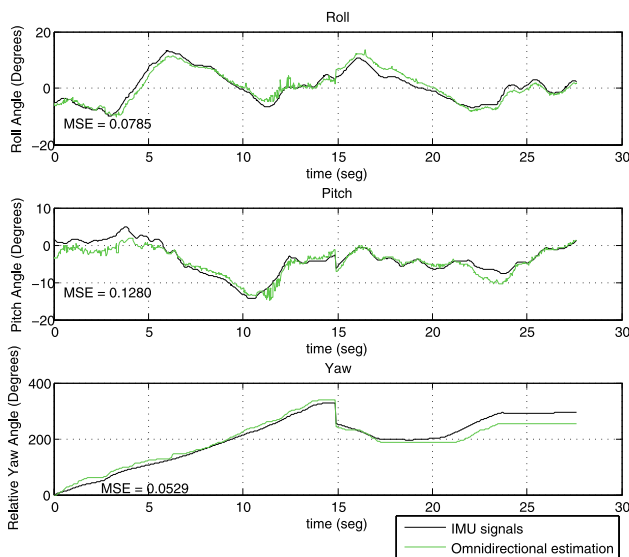
### 3.5 Attitude estimation tests

Several tests have been made using the Colibri testbeds (COLIBRI 2009). In this test, a series of flights were performed in both autonomous and manual modes. In autonomous mode, the helicopter takes a previously defined trajectory. In manual mode, a safety pilot takes a free flight with strong movements of the helicopter. The algorithm is tested during these flights and an image sequence is stored, associating to each of the processed images the UAV attitude information estimated by the omnidirectional system. Also, a flightlog is created with the GPS position, IMU data (heading, body frame angles and displacement velocities), and the helicopter attitude position as estimated by the Kalman filter of the controller on the local plane with reference to the takeoff point. These values are used for later comparisons with the estimated data using the catadioptric system.

For these tests, a mirror with a hyperbolic shape of the form  $y = 39.3\sqrt{1 + \frac{x^2}{19.646^2}} - 43.92$  and catadioptric parameters  $d = 100$  mm and  $4p = 19.646$  is combined with a CCD firewire camera with a  $640 \times 480$  pixels resolution and focal length of 12 mm. The camera is located on the Pan and Tilt platform of the helicopter in such a way that the vertical axes of the camera and helicopter are parallel (by adjusting the camera tilt). In this way, the hyperbolic mirror faces downward, and the camera looks up. This position ensures that all axes of the helicopter and the camera are coincident, so that the obtained roll and pitch angles for the camera are the same for the helicopter frame as shown in Fig. 3. The estimated values of roll, pitch and yaw from the test flight are compared with the corresponding stored IMU values. Figure 7 shows the results and the mean squared error (MSE) of the estimated values, compared with the IMU values as ground truth.

The estimated values for roll are very close to the IMU values and have a small MSE against the absolute values measured in the IMU.

Pitch values are also estimated accurately compared with the ground truth IMU. However when the helicopter has a high nose-up angle, a portion of the ground is occluded on the catadioptric image by the platform's structure and the



**Fig. 7** Omnidirectional values vs. IMU measurements. *Up*: Roll ( $-180 < \phi < 180$ ). *Center*: Pitch ( $-90 < \theta < 90$ ). *Down*: Relative Yaw ( $0 < \psi < 360$  in relative mode)

UAV's reflections, causing a small error in the adjustment of the skyline on the equivalent sphere projection and the pitch estimation. This causes the MSE to have the highest value of all the estimated parameters, although this value is still a high-quality measurement.

Yaw estimation uses the first image taken by the algorithm as a reference, calculating rotation with respect to this reference image. Absolute Yaw data measured by the IMU is rotated according to the first image angle and changed to a range between  $0 < \psi < 360$  for easy comparison with omnidirectional data. Results show that the rotation between frames, as well as the total rotation are both estimated correctly, taking into account the fact that the unwrapped panoramic image only has a resolution of 0.5 pixels per degree. As mentioned in Sect. 3.4, variations in roll and pitch angles and translational movements do not notably affect the estimated rotation between frames if the distances to surrounding objects are large and the displacements and angles between consecutive frames are small, as is the with UAV flight in an open area. This is reflected in the MSE which is the lowest value of the all estimated parameters, confirming that the approximation is valid under the helicopter's flight conditions.

Several tests have been done in situations of totally clear weather and also with cloudy skies. Results are good in both cases, showing the feasibility of using a catadioptric system as a UAV attitude and heading estimator or as a redundant visual system. The attitude estimation loop operates at an average rate of 12 frames per second (fps). The total video sequence for this manual flight and additional test in manual and autonomous mode can be seen on the Colibri Project Web Page (COLIBRI 2009).

## 4 Odometry estimation using stereo vision

This section presents a system for estimating the altitude and motion of an aerial vehicle using a stereo vision system. The system first detects and tracks interest points in the scene. Then the distance from the stereo system to the plane that contains the features is found by matching the features between the left and right images and using the disparity principle. The motion is then recovered by tracking pixels from one frame to the next, finding its visual displacement, and resolving camera rotation and translation by a least-squares method (Mejias et al. 2007).

### 4.1 Height estimation

Height estimation is performed on a stereo system first using a detection phase of salient features in the environment. This procedure is applied to each of the stereo images using the Harris corner detector because it offers the best performance in terms of speed, robustness, stability in outdoor environments and an acceptable invariance to rotation and translation and to small changes in scale. See Ashbrook (1992) for more details.

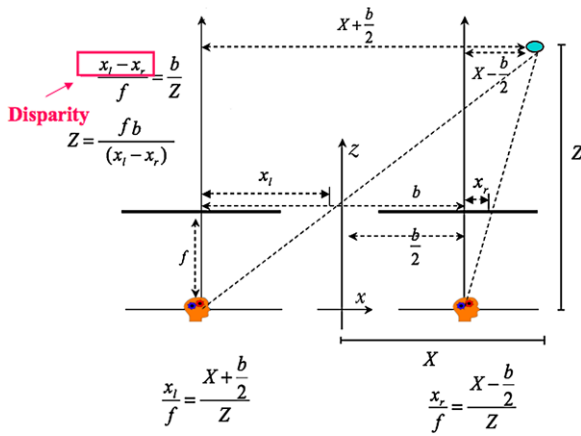
In the second phase, a correlation algorithm is applied in order to find the correspondences between two sets of features (features of the right and left images). A double check is performed by checking right against left and then, comparing left with right. The correlation is based on the ZNNC (Zero Mean Normalized Cross Correlation) technique, which offers good robustness against changes in light and other environmental conditions (Martin and Crowley 1995).

Once the correspondence problem has been solved, the stereo disparity principle is used to find the distance from the cameras to the plane that contains the features. This distance is found by taking into account an error tolerance (given that the correspondence is not perfect) and considering that all pixels belong to the same plane. The disparity is inversely proportional to the scene depth multiplied by the focal length ( $f$ ) and the baseline ( $b$ ), so that the depth can be computed using the expression shown in Fig. 8.

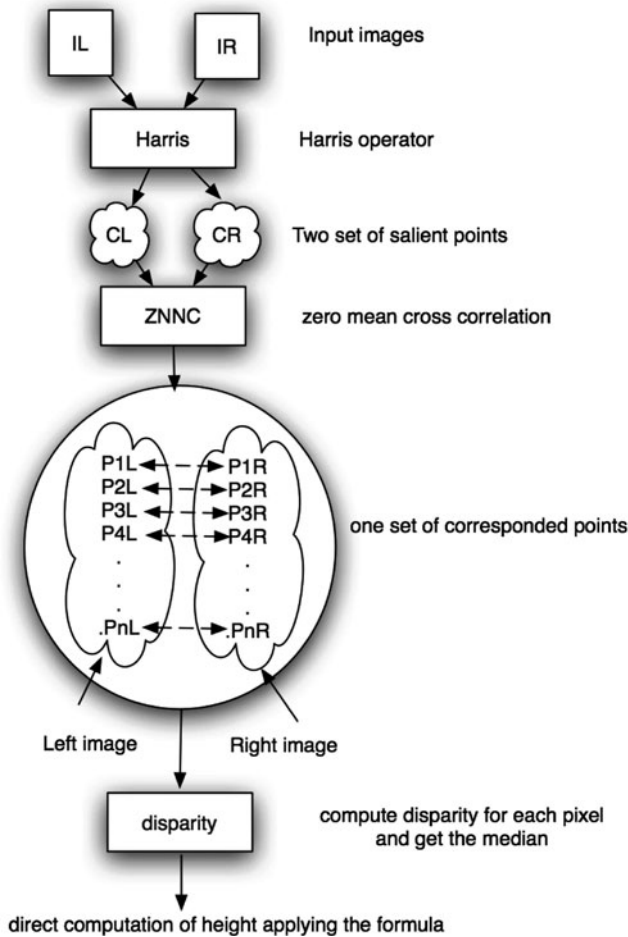
Figure 9 describes the algorithm used to estimate the distance from the stereo system to the plane. The stereo system has been located onboard the helicopter in two different positions. In the first configuration, the stereo system is looking down, perpendicular to the ground, so that the estimated distance corresponds to the UAV altitude. In the second configuration, the stereo system is looking forward, and the estimated distance corresponds to the distance between the UAV and an object or feature in the scene.

### 4.2 Motion estimation

The motion estimation is accomplished in two stages. In the first stage, features are detected and pre-matched using



**Fig. 8** Stereo disparity for aligned cameras with all pixels in the same plane. The stereo disparity principle is used to find the distance from the stereo system to the plane that contains the features



**Fig. 9** Height estimation using the Harris corner detector and ZNNC. Height is obtained using the stereo disparity principle

the Harris algorithm and the ZNNC technique; as a consequence, those points with a correlation coefficient higher than 0.85 are considered for the next stage.

The second stage, deals with the motion estimation problem. The motion estimation is solved using the ICP algorithm (Iterative Closest Point) and SVD technique (Singular Value Decomposition). The ICP algorithm is used to solve the correspondence problem (image registration): assuming there are two sets of points known as data and model,  $P = \{p_i\}_1^{N_p}$  and  $M = \{m_i\}_1^{N_m}$  respectively with  $N_p \neq N_m$ , the closest point is defined as the one that minimizes  $cp(p) = \arg \min_{m \in M} \|m - p\|$ . Then, SVD is used to find the best possible alignment of  $P$  and  $M$  by relating them through equation  $M = RP + t$ . Therefore, the Motion Estimation algorithm using ICP and SVD can be summarized in the following steps:

1. Compute the subset of closest points (CP):

$$y = \{m \in M \mid p \in P : m = cp(p)\}$$

2. Compute the least-squares estimate of motion bringing  $P$  onto  $y$  (the transformation is calculated using SVD):

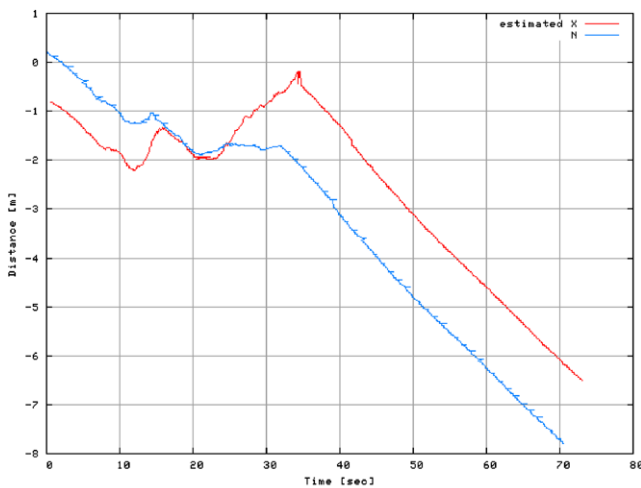
$$(R, t) = \arg \min_{R, t} \sum_{i=1}^{N_p} \|y_i - Rp_i - t\|^2$$

3. Apply motion to the data points,  $P \leftarrow RP + t$
4. If the stopping criterion is satisfied, exit; else goto 1.

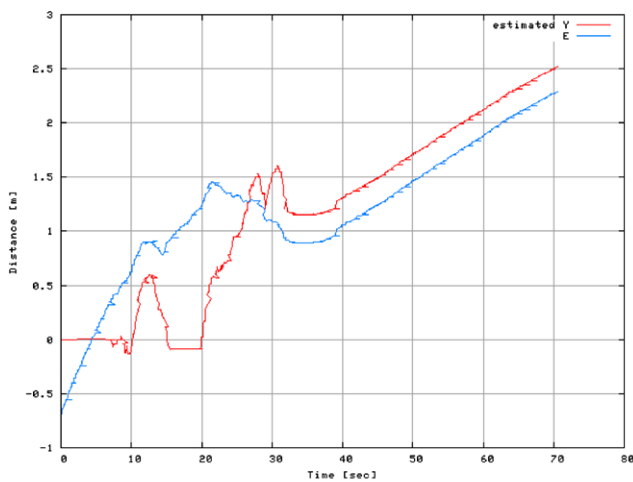
The calculation of the rotation matrix and translation vector in step (2) can be summarized as follows: first, the rotation matrix is calculated using the centroid of the set of points. The centroid is calculated as  $y_{c_i} = y_i - \bar{y}$  and  $p_{c_i} = p_i - \bar{p}$ , where  $\bar{y} = \frac{1}{N_p} \sum_{N_p} cp(p_i)$  and  $\bar{p} = \frac{1}{N_p} \sum_{N_p} p_i$ . Then, the rotation matrix is found minimizing  $\min_R \sum_{N_p} \|y_{c_i} - Rp_{c_i}\|^2$ . This equation is minimized when the trace (RK) is maximized with  $K = \sum_{N_p} y_{c_i} p_{c_i}^T$ . Matrix  $K$  is calculated using SVD as  $K = VDU^T$ . Thus, the optimal rotation matrix that maximizes the trace is  $R = VU^T$  and the optimal translation that aligns the centroids is  $t = \bar{y} - R\bar{p}$ .

#### 4.3 Height and motion estimation using a stereo system

Stereo tests were done using a Firewire stereo camera on-board the UAV, with 10 cm od distance between lenses (disparity). This camera captures images of  $240 \times 320$  size at 15 fps (frames per second). In these experiments, the helicopter is commanded to fly autonomously following a given trajectory while the onboard stereo vision algorithm is running at a processing frequency of 10 fps. The tests reveal a correlation between the stereo visual estimation and the on-board helicopter state given by its sensor suite. Figures 10 and 11 show the results of one flight trial in which the longitudinal displacement (X), lateral displacement (Y), altitude (H) and relative orientation (Yaw) are estimated. Altitude is computed to be negative since the helicopter's body



(a) Visually Estimated X and Northing (N).



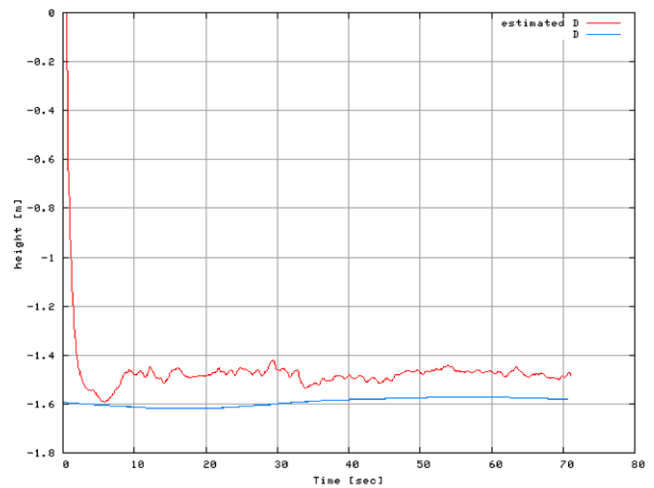
(b) Visually Estimated Y and Easting (E).

**Fig. 10** Results using a stereo system for Height and Motion Estimation I. The visual estimation (red lines) for longitudinal displacements (a) and lateral displacements (b) are compared with the helicopter state estimation (blue lines)

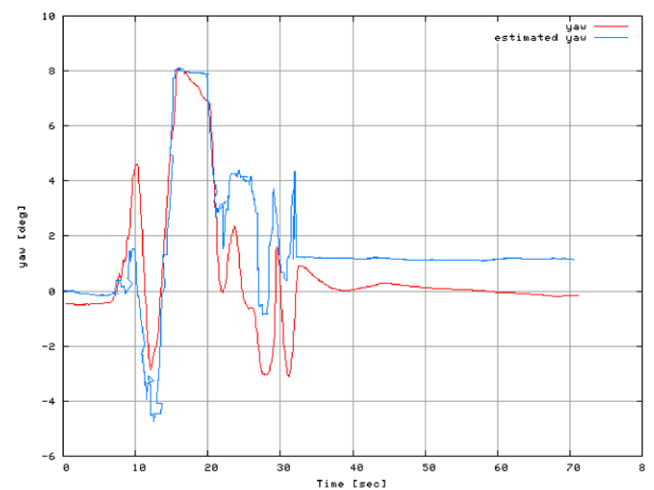
frame is used as a reference system. Each estimate is correlated with its similar value taken from the onboard helicopter state, which uses an EKF (Extended Kalman Filter) to fuse the onboard sensors. Table 2 shows the error analysis based on the mean squared error of the visual estimation and the helicopter's state. Four measures of the mean squared error have been calculated: the error vision-GPS Northing ( $MSE_N^V$ ), the error vision-GPS Easting ( $MSE_E^V$ ), the error vision-yaw ( $MSE_\psi^V$ ) and the error vision-altitude ( $MSE_H^V$ ).

## 5 Visual control

In this section, we describe a control system that can follow static and moving objects in real time. This new visual servoing control improves on previous works (Campoy et



(a) Visually Estimated H and helicopter altitude.



(b) Visually Estimated Yaw and helicopter Yaw.

**Fig. 11** Results using a stereo system for Height and Motion Estimation II. The visual estimation (red lines) for the altitude (a) and the relative orientation (b) are compared with the helicopter state estimation (blue lines)

**Table 2** Error analysis for the helicopter's experimental trials

Exp.	Test
$MSE_N^V$	1.0910
$MSE_E^V$	0.4712
$MSE_\psi^V$	1.7363
$MSE_H^V$	0.1729

al. 2008; Mejias et al. 2006) in the inclusion of control of a pan and tilt visual platform. This gives a quicker response than helicopter movements and also gives total freedom of movement to the UAV, making it possible to track objects not only during hovering flight, as in the previous work, but also during a path-planned flight or when flying under manual control.

In this section, we also describe the implemented visual control system, the tracking algorithm, and the fuzzy controllers used for improving visual servoing control.

### 5.1 Visual tracking system

The visual tracking system for helicopter and platform control on a UAV is based on the Lucas-Kanade algorithm (Lucas and Kanade 1981) which is a Gauss-Newton gradient descent non-linear optimization algorithm. An optical flow with a pyramidal implementation of this algorithm is used. It relies on two premises: first, intensity constancy in the vicinity of each pixel considered as a feature; second, minimal change in the position of the features between two consecutive frames. Given these restrictions, to ensure the performance of the algorithm, it can be expressed in the following form: if we have a feature position  $p_i = (x, y)$  in the image  $I_k$ , the objective of the tracker is to find the position of the same feature in the image  $I_{k+1}$  that fits the expression  $p'_i = (x, y) + t$ , where  $t = (t_x, t_y)$ . The  $t$  vector is known as the optical flow, and it is defined as the visual velocity that minimizes the residual function  $e(t)$  defined as:

$$e(t) = \sum^W (I_k(p_i) - I_{k+1}(p_i + t))^2 w(W) \quad (10)$$

where  $w(x)$  is a function that assigns different weights to the comparison window  $W$ . This equation can be solved for each tracked feature, but since it is expected that all features on physical objects move in solidarity, summation can be done over all features, obtaining the movement of the object on the image plane and using as the input to the Fuzzy controller explained in Sect. 5.3.

### 5.2 Control scheme

The flight control system is composed of three control loops arranged in a cascade formation, allowing it to perform tasks at different levels depending on the workspace of the task. The first control loop is in charge of the attitude of the helicopter. It is based on a decoupled PID control in which each degree of freedom is controlled separately based on the assumption that the helicopter dynamics are decoupled. The attitude control stabilizes the helicopter during hovering by maintaining the desired roll, pitch and heading. It is implemented as a proportional-plus-derivative (PD) control. The second control loop is the one that has the visual signal feedback. This control loop is composed of the visual pan-tilt platform and the UAV heading controllers. All of them are designed with Fuzzy Logic techniques, as explained in more detail in the next subsection. With this loop, the UAV is capable of receiving external references (from the Visual Process) to keep it aligned with a selected target, and it leaves the stability of the aircraft to the most internal

loop in charge of the attitude. The third controller (position based control) is at the higher level of the system, and is designed to receive GPS coordinates. The control scheme (Fig. 12) allows different modes of operation, one of which is to take the helicopter to a desired position (position control). Integration of the visual references uses the TCP/UDP and API architecture explained in Sect. 2. More detailed information of how those messages are integrated in the server process running onboard the UAV is given in Mejias (2007).

### 5.3 Visual servoing using fuzzy controllers

This section explains the fuzzy controllers used for controlling the heading and the camera platform and the various experiments to test them. The combined control of the video platform and the heading of the helicopter allows a suitable automatic control to be applied in different situations. In addition to overcoming environmental difficulties or adapting to the needs of the pilot, it provides the possibility of tracking static and moving objects in these situations:

1. With a preprogrammed series of way points.
2. When the pilot is controlling the helicopter or when the flight commands are sent from the ground station, making it easier to control the UAV by allowing it to control heading automatically.
3. Staying in hovering position in a safe place.

Also, the fuzzy logic provides a more versatile solution for controlling the platform and helicopter because it is easier to tune and to adapt to the real world, due to the fact that it can represent non-linear problems. This gives a better solution for overcoming the helicopter's own vibrations and other perturbation signals from the environment. In this chapter, we will first describe the control configuration of the UAV, continue with an explanation of our fuzzy software implementation, and finish by presenting the various fuzzy controllers that we implemented.

#### 5.3.1 Fuzzy controllers

For this work we use the software MOFS (*Miguel Olivares' Fuzzy Software*), developed in previous works (Olivares and Madrigal 2007; Olivares and Campoy 2008). This software was independently designed, defining one class for each part of the fuzzy-logic environment (variables, rules, membership functions, and defuzzification modes) in order to facilitate future updates and easy interaction with the system. There are different classes depending on the system we want to create; we can define the number of inputs and outputs that we prefer or make parts of the system work in serial or parallel mode.

The software can be updated in any of its fuzzy-logic parts, such as by introducing different membership functions, fuzzy inference types, or defuzzification modes.

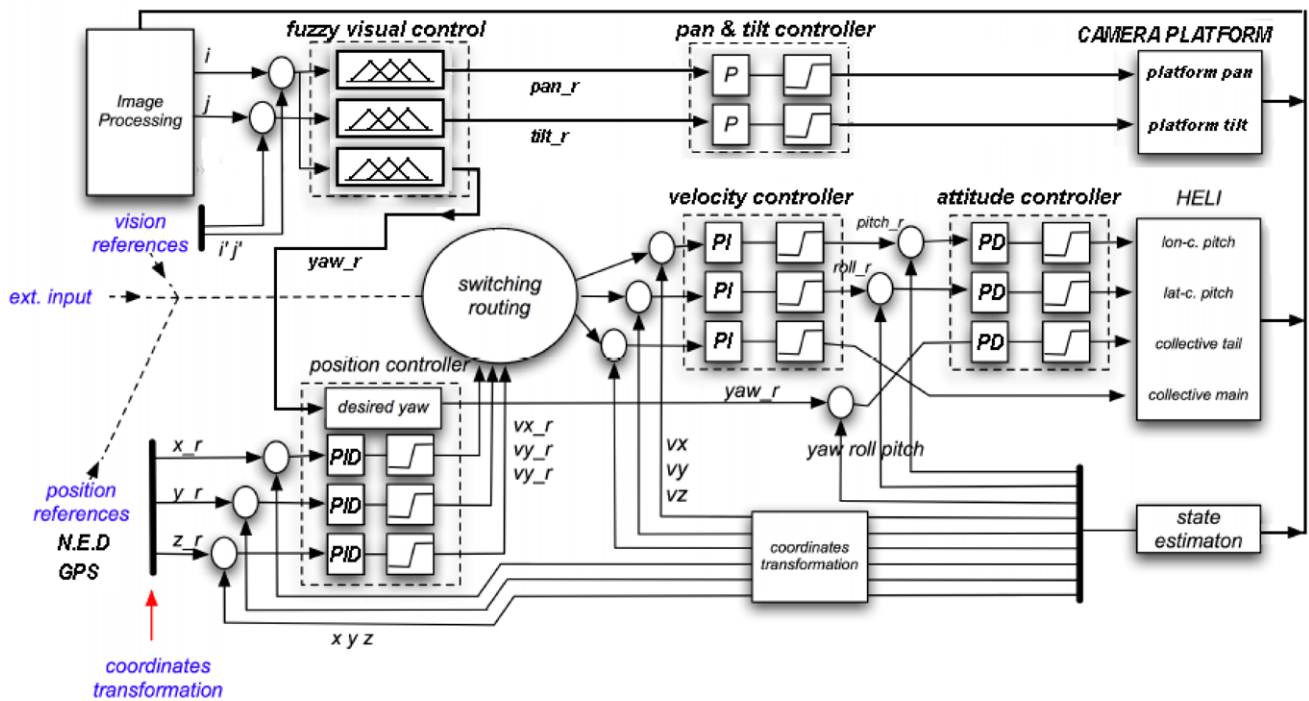


Fig. 12 Visual control scheme

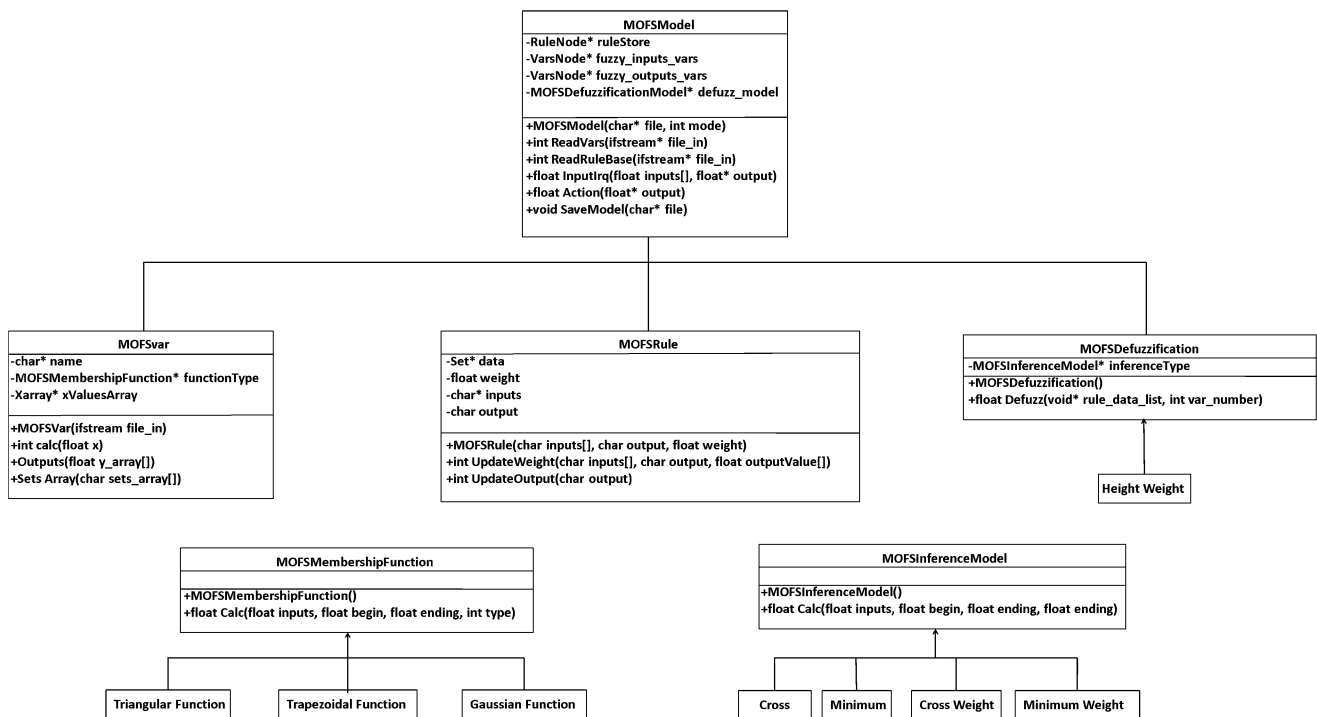


Fig. 13 Software definition

One of the differences between this software and other fuzzy software is that it allows us to represent a more important sector in each fuzzy variable, giving us the possibility of reducing the size of the rule-base, thereby improving

response time and reducing the computational cost. Other differences and more documentation can be found by consulting (Olivares and Madrigal 2007; Olivares and Campoy 2008; Olivares-Mendez et al. 2009a, 2009b). The MOFS is

defined like a controller class inside the server program running onboard the helicopter.

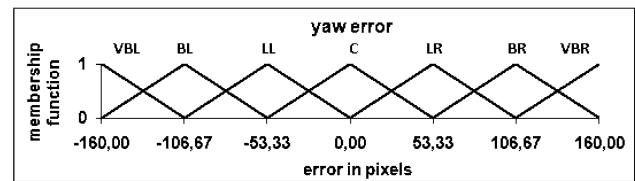
The fuzzification of the inputs and the outputs is defined using a triangular membership function, for the platform controllers, and trapezoidal membership functions, for the heading. The controllers have two inputs, the error between the center of the object and the center of the image (Figs. 14(a) and 14(b)) and the difference between the last and the actual error (Figs. 14(b) and 14(d)), derived from the position of the velocity of the object to track. The platform controller output represents the required movement of the servomotor in the two axes to minimize the error, on a scale from 0 to 255, (Fig. 15(a)). The heading controller takes the same inputs as the yaw controller (Figs. 14(a) and 14(b)), and the output represents how many radians the UAV must rotate to line up with the object (Fig. 15(b)).

The three controllers work in parallel, providing redundant operation on the yaw axis and reducing the error in the yaw-platform controller from the limitations of the visual algorithm and the movement of the servos. The third controller also serves to eliminate the turn limitations of the platform when the tracked object moves to the back part of the UAV. All of these are guided by a 49-rules base, defining for the output of the platform controllers a more important sector in the section near zero, as shown in Fig. 15(a). This option give us the possibility of defining a very sensible controller when the error is small and the object is very near to the center of the image and a very quick-response controller when the object is far away. For the heading controller, we defined a trapezoidal part in the middle of the output in order to help the platform controller when the object to be tracked is far from the center of the image. With these trapezoidal definitions, we obtain greater stability of helicopter motion in situations where the tracked object is near the center, obtaining a 0 value.

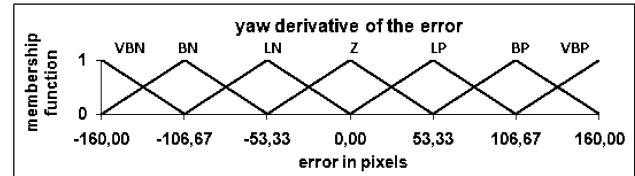
### 5.3.2 Heading control experiments

In order to test and fit the heading controller, we made some tests with the platform control, tracking a real object while remaining in real-time communication with the helicopter simulator. For this test, we used a static tracked object and moved the visual platform, trying to emulate the movements of the helicopter and the tracked object. The error is shown in Fig. 16 in pixels.

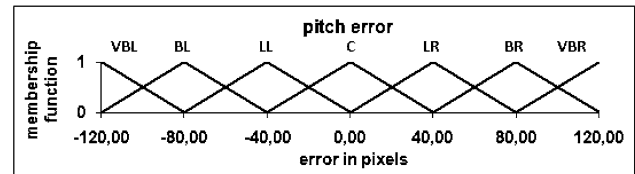
In Fig. 17 we can see the response of the Fuzzy controller of the visual platform pitch angle, responding very quickly and with good behavior. In addition, Fig. 18 shows the controller response of the other axis of the platform. We can see a big and rapid movement near 1600 frames, reaching an error of almost 100 pixels. For this change we can see that the response of the controller is very fast, only 10 frames.



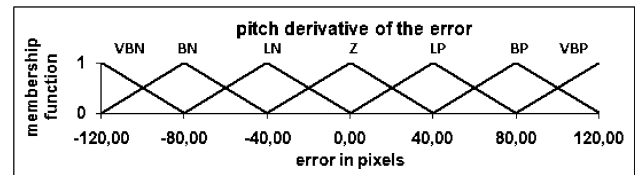
(a) Yaw Error.



(b) Derivative of the Yaw error.

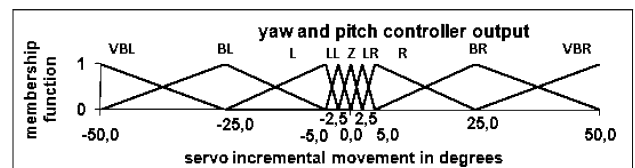


(c) Pitch Error.

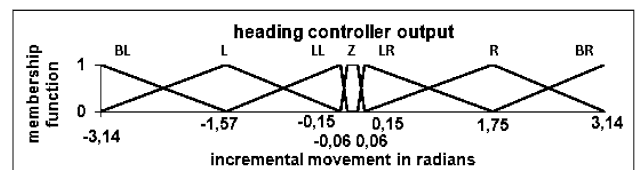


(d) Derivative of the Pitch error.

Fig. 14 Inputs variables of the controllers



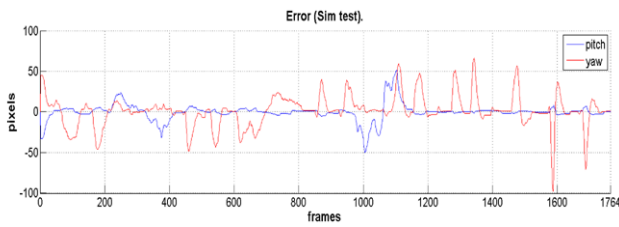
(a) Output of the Yaw and the Pitch Platform controller.



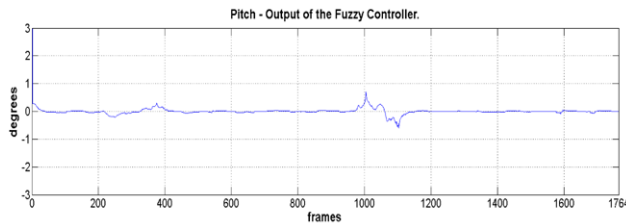
(b) Output of the Heading controller.

Fig. 15 Output variables of the controllers

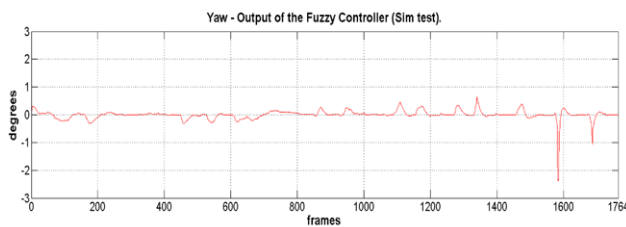
The response of the heading controller is shown in Fig. 19, where we can see that it only responds to big errors in the yaw angle of the image. Also, we can see, in Fig. 20, how these signals affect the helicopter's heading, changing the yaw angle in order to collaborate with the yaw controller of the visual platform.



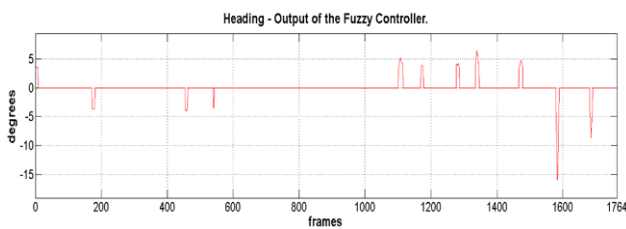
**Fig. 16** Error between the static object tracked and the center of the image, running with the UAV simulator



**Fig. 17** Response of the fuzzy control for the Pitch axis of the visual platform tracking a static object with the simulator of the UAV control



**Fig. 18** Response of the fuzzy control for the Yaw axis of the visual platform tracking a static object with the simulator of the UAV control

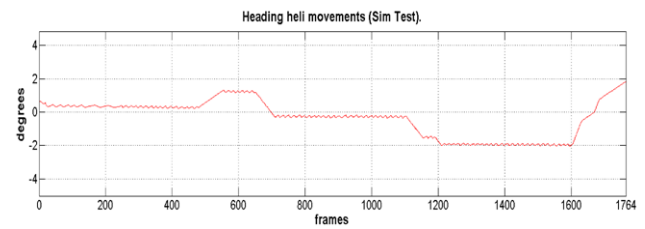


**Fig. 19** Response of the fuzzy control for the heading of the helicopter

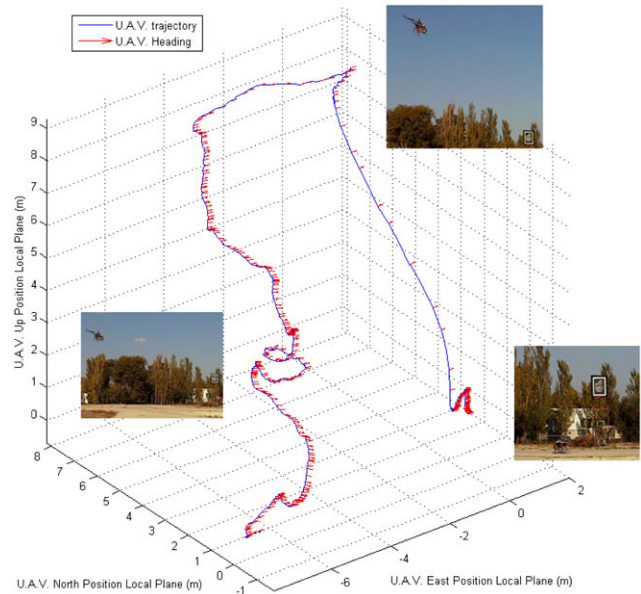
### 5.3.3 Tests on UAV

This subsection presents results from real tests onboard the UAV, tracking static and moving objects. For these tests, we use the controllers of the visual platform.

**Tracking static objects** In our tests, we tracked a static object during the full flight of the UAV, from takeoff to landing. This flight was made by sending set-points from the ground station. Figure 21 shows a 3D reconstruction of the flight using the GPS and IMU data on three axes: North (X),



**Fig. 20** Heading response



**Fig. 21** 3D flight reconstruction from the GPS and the IMU data from the UAV. Where, the 'X' axis represents the NORTH axis of the surface of the tangent of the earth, the 'Y' axis represents the EAST axis of the earth, the 'Z' is the altitude of the helicopter and the red arrows show the pitch angle of the helicopter

East (Y), and Altitude (Z), the first two of which are the axes forming the surface of the local tangent plane. The UAV is positioned over the north axis, looking to the east, where the mark to be tracked is located. The frame rate is 15 frames per second, so those 2500 frames represent a full flight of almost 3 minutes.

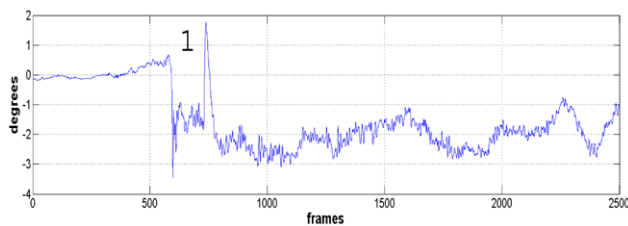
Figure 22 shows the UAV's yaw and pitch movements.

In Fig. 24, the output of the two Fuzzy-MOFS controllers in order to compensate the error caused by the changes of the different movements and angle changes of the UAV flight, where we can see the different responses of the controllers, depending the sizes and the types of the perturbations.

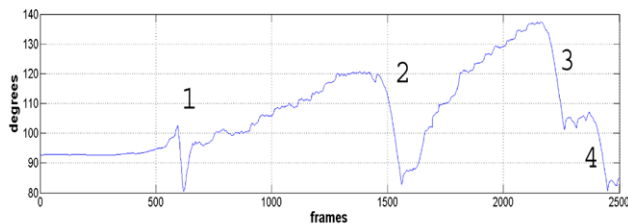
Looking the data shown in Table 3 and Figs. 22 and 23, we can realize that the perturbations affect the control system depending the size and the type of those, but never making it lost the mark to track. The maximum error in the flight, after the ignition of the motor, was +100 pixels during the initial elevation of the aircraft and 55 pixels during the flight. The error represents 62.2% of the initial elevation, where

**Table 3** Data from big attitude changes sections of the flight

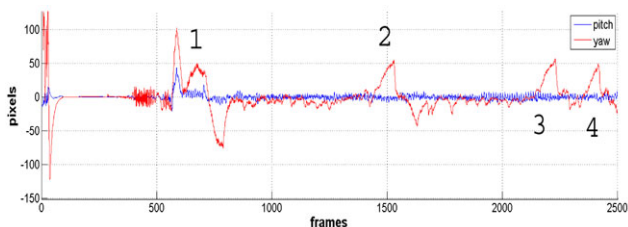
Section	Frames interval	Attitude angle	Degrees	Frames num.	Time	Degrees per sec.	Pixels error
1	540–595	Yaw	+8	55	3.6 s	+2.28/s	+100 (Yaw)
1	590–595	Roll	−5	5	0.33 s	−15/s	+100 (Yaw)
1	570–595	Pitch	−4	25	1.6 s	−2.5/s	+40 (Pitch)
1	595–620	Yaw	−22	25	1.6 s	−13.75/s	+50 (Yaw)
1	595–660	Roll	+10	65	4.3 s	+2.35/s	+50 (Yaw)
1	620–660	Yaw	+20	40	2.6 s	+15.38/s	−75 (Yaw)
2	1460–1560	Yaw	−40	100	6.6 s	−6.06/s	+52 (Yaw)
2	1560–1720	Yaw	+28	160	10.6 s	+2.64/s	48 (Yaw)
3	2170–2260	Yaw	−35	90	6 s	−5.8/s	55 (Yaw)
4	2375–2450	Yaw	−27	75	5 s	−5.4/s	48 (Yaw)



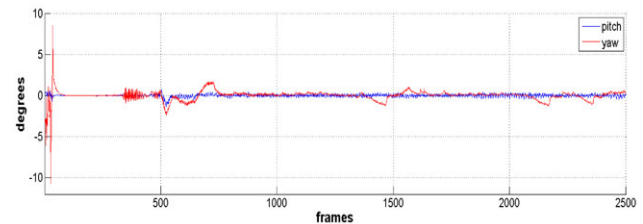
(a) Pitch angle movements.



(b) Yaw angle movements.

**Fig. 22** Different pitch and yaw movements of the UAV**Fig. 23** Error between center of the image and center of the object to track

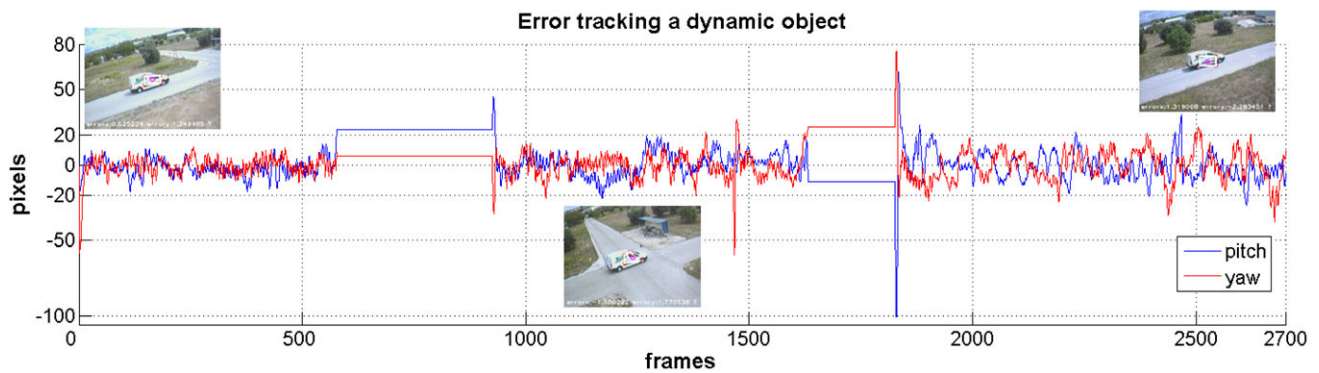
we have a fusion of all the possible movements and angles changes in a UAV, with a high rate of change of 15.38 degrees per sec. Also, we ranged over 34.375% of the yaw axis of the camera, with a maximum angle change of 6.06 degrees per second. Thus, we can say that the controller shows good behavior in solving these kinds of problems

**Fig. 24** Output from the fuzzy controller

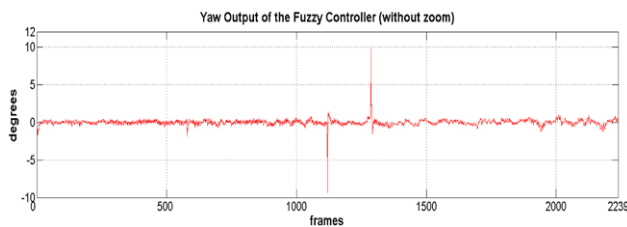
Another kind of problem occurs when the helicopter is not undergoing big changes in its attitude angles. In those phases of the flight with light movements, we have an error of  $\pm 5$  pixels in the pitch axis of the platform and a +5, −15 pixels error in the yaw angle. In degrees, this amounts to 0.6562 degrees of pitch and 0.6562, −1.96875 degrees of yaw. Notice that, in this section of light movements, we have continued small yaw changes, as shown in Fig. 22(b) between Sects. 1 and 2.

**Tracking moving objects** Here we present a tracking of a van with continuous movements of the helicopter increasing the difficulty of the test. In Fig. 25 we can see the error in pixels of the two axes of the image. Also, we can see the moments where we deselected the template and re-selected it, in order to increase the difficulty for the controller. These intervals show up as the error remains fixed in one value for a long time.

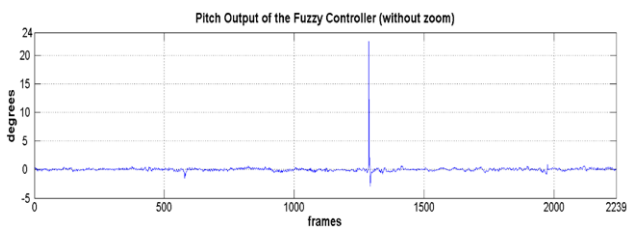
In Figs. 26 and 27 we can see the response of the to the controllers, showing the large movements sent by the controller to the servos when the mark is re-selected. Notice that in all of the figures that show the controller responses, there are no data registered when the mark selection is lost because no motion is tracked. Figure 25 shows the data from the flight log, the black box of the helicopter. We can see that the larger responses of the controllers are almost  $\pm 10$



**Fig. 25** Error between center of the image and center of the dynamic object (a van) to track



**Fig. 26** Response of the fuzzy control for the Yaw axis of the visual platform tracking a dynamic object (a van)



**Fig. 27** Response of the fuzzy control for the Pitch axis of the visual platform tracking a dynamic object (a van)

degrees for the yaw controller and almost 25 degrees for the pitch controller, corresponding to the control correction over a period of fewer than 10 frames.

It is possible to view these test videos and more on (COLIBRI 2009).

## 6 Conclusions

This paper has described the research, results, and discussion on the use of several computer vision techniques on-board a UAV. These computer vision techniques are not merely used to acquire environmental visual information that can be used afterward by offline processing. Rather, this paper has shown that computer vision can play an important role on-line during the flight itself, in order to acquire the adequate sequences necessary to actively track targets (fixed or moving) and to guide and control flight trajectories.

We have developed and tested a method for UAV attitude (roll and pitch) and heading estimation based totally on visual information taken by a catadioptric camera. This approach has been validated against inertial measures using a UAV testbed, showing that the estimated values are a very good approximation of the real state of the aircraft, demonstrating the feasibility of using this kind of system as a main sensor on micro UAVs with reduced sensor payloads or as a redundant system for IMU and gyroscopes in cases of failure or malfunction of these systems.

Additionally, we have been able to estimate the height and motion of a UAV by employing a stereo system. Results of an outdoors test using a UAV have been compared against GPS data, producing a very good estimate of the vertical and lateral displacements of the UAV using the ground as a reference.

Although the measured values do not have a high resolution, they have shown good response to large changes on the aircraft's flying state with a near real time computational cost. Considering these facts, the inertial data measured with the catadioptric system and the motion estimation with the stereo system have been shown to be suitable for a flight controller based on visual sensors, with additional features such as object tracking and servoing.

The outputs of the image processing algorithm using a Lucas-Kanade tracker for static and moving objects are the visual references used by an autonomous fuzzy system to control a pan and tilt camera platform and the UAV heading. The controllers show excellent behavior when tracking both static and moving objects, despite the perturbations in the environment and the helicopter's own vibrations. The uses of the pan and tilt visual platform give the helicopter freedom of movement, as well as faster response when moving objects are tracked, compared with other implementations of visual servoing on UAVs without a pan and tilt platform. The combined platform and heading control allows us to make smooth movements of the platform, increasing both the workspace of the visual tracker and the UAV. The developed visual tracking algorithms have many potential ap-

plications in specific situations, such as staying near electric structures, inspection of wind farms or dams, and fire monitoring.

Based on the results of our work, we conclude that the UAV field has reached an important stage of maturity, in which the possibility of using UAVs in civilian applications can now be imagined and in some cases attained. We have experimentally demonstrated several capabilities of an autonomous helicopter using visual information such as attitude estimation, navigation, trajectory planning, and visual servoing. The successful implementation of all these algorithms confirms the necessity of equipping UAVs with additional functionalities when tasks such as outdoor structures, inspection, and object tracking are required.

**Acknowledgements** The work reported in this paper is the conclusion of several research stages at the Computer Vision Group - Universidad Politécnica de Madrid. The authors would like to thank Jorge Leon for supporting the flight trials, the I.A. Institute - CSIC for collaborating in the flights, and the Universidad Politécnica de Madrid and Comunidad Autónoma de Madrid and Fondo Social Europeo (FSE) for the authors' Ph.D Scholarships. This work has been sponsored by the Spanish Science and Technology Ministry under the grant CICYT DPI 2007-66156.

## References

- Antonisse, H. J. (1982). Image segmentation in pyramids. *Computer Vision, Graphics, and Image Processing*, 19(4), 367–383.
- Ashbrook, A. P. (1992). *Evaluations of the Susan corner detection algorithm* (Tech. rep.). Electronic System group, Department of Electronic and Electrical Engineering, University of Sheffield, UK.
- Baker, S., & Nayar, S. K. (1999). A theory of single-viewpoint catadioptric image formation. *International Journal of Computer Vision*, 35(2), 1–22.
- Barreto, J., & Araujo, H. (2001). Issues on the geometry of central catadioptric image formation. *Computer Vision and Pattern Recognition*. doi: 10.1109/CVPR.2001.990992.
- Barreto, J., & Araujo, H. (2002). Geometric properties of central catadioptric line images. In *ECCV '02: proceedings of the 7th European conference on computer vision, part IV* (pp. 237–251). London: Springer.
- Beyeler, A., Zufferey, J. C., & Floreano, D. (2009). Vision-based control of near-obstacle flight. *Autonomous Robots*, 27(3), 201–219. doi: 10.1007/s10514-009-9139-6.
- Campoy, P., Correa, J., Mondragon, I., Martinez, C., Olivares, M., Mejias, L., & Artieda, J. (2008). Computer vision onboard UAVs for civilian tasks. *Journal of Intelligent and Robotic Systems*. doi: 10.1007/s10846-008-9256-z.
- Carnie, R., Walker, R., & Corke, P. (2006). Image processing algorithms for UAV “sense and avoid”. In *Robotics and Automation*. doi: 10.1109/ROBOT.2006.1642133.
- Cheng, Y., Maimone, M. W., & Matthies, L. (2006). Visual odometry on the Mars exploration rovers. *IEEE Robotics and Automation magazine*, 13(2), 54–62.
- COLIBRI. (2009). Universidad Politécnica de Madrid. Computer Vision Group. COLIBRI Project. <http://www.disam.upm.es/colibri>.
- Conroy, J., Gremillion, G., Ranganathan, B., & Humbert, J. S. (2009). Implementation of wide-field integration of optic flow for autonomous quadrotor navigation. *Autonomous Robots*, 27(3), 189–198. doi: 10.1007/s10514-009-9140-0.
- Corke, P., Strelow, D., & Singh, S. (2004). Omnidirectional visual odometry for a planetary rover. In *IEEE/RSJ international conference on intelligent robots and systems*, Japan.
- Cornall, T., & Egan, G. (2004). Measuring horizon angle from video on a small unmanned air vehicle. In *2nd international conference on autonomous robots and agents*.
- Cornall, T., Egan, G., & Price, A. (2006). Aircraft attitude estimation from horizon video. *Electronics Letters*, 42(13), 744–745. doi: 10.1049/el:20060547.
- Demonceaux, C., Vasseur, P., & Pgard, C. (2006). Omnidirectional vision on UAV for attitude computation. In *IEEE international conference on robotics and automation (ICRA'06)* (pp. 2842–2847). Orlando: IEEE.
- Dusha, D., Boles, W., & Walker, R. (2007). Fixed-wing attitude estimation using computer vision based horizon detection. In: *Proceedings 12th Australian international aerospace congress* (pp. 1–19), Melbourne, Australia.
- Ettinger, S. M., Nechyba, M. C., Ifju, P. G., Waszak, M. (2002). Vision-guided flight stability and control for micro air vehicles. In *IEEE international conference on intelligent robots and systems*. New York: IEEE.
- Geyer, C., & Daniilidis, K. (2000). A unifying theory for central panoramic systems and practical applications. In *ECCV* (Vol. 2, pp. 445–461).
- Geyer, C., & Daniilidis, K. (2001). Catadioptric projective geometry. *Journal of Computer Vision*, 43, 223–243.
- Hrabar, S., & Sukhatme, G. (2003). Omnidirectional vision for an autonomous helicopter. In *IEEE international conference on robotics and automation* (pp. 558–563).
- Hrabar, S., & Sukhatme, G. (2009). Vision-based navigation through urban canyons. *Journal of Field Robotics*, 26(5), 431–452. doi: 10.1002/rob.v26:5.
- Hrabar, S., Sukhatme, G., Corke, P., Usher, K., & Roberts, J. (2005). Combined optic-flow and stereo-based navigation of urban canyons for a UAV. *Intelligent Robots and Systems*. doi: 10.1109/IROS.2005.1544998.
- Kendoul, F., Nonami, K., Fantoni, I., & Lozano, R. (2009). An adaptive vision-based autopilot for mini flying machines guidance, navigation and control. *Autonomous Robots*, 27(3), 165–188. doi: 10.1007/s10514-009-9135-x.
- Labrosse, F. (2006). The visual compass: performance and limitations of an appearance-based method. *Journal of Field Robotics*, 23(10), 913–941.
- Lucas, B. D., & Kanade, T. (1981). An iterative image registration technique with an application to stereo vision. In *Proceedings of the 7th IJCAI* (pp. 674–679), Vancouver, Canada.
- Martin, J., & Crowley, J. (1995). *Experimental comparison of correlation techniques* (Tech. rep.). IMAG-LIFIA, 46 Av. Félix Viallet, 38031 Grenoble, France.
- Matthies, L. (1989). *Dynamic stereo vision*. Cmu-cs-89-195, Carnegie Mellon University. Computer Science Department.
- Mejias, L. (2006). *Control visual de un vehículo aéreo autónomo usando detección y seguimiento de características en espacios exteriores*. PhD thesis, Escuela Técnica Superior de Ingenieros Industriales, Universidad Politécnica de Madrid, Spain.
- Mejias, L., Saripalli, S., Campoy, P., & Sukhatme, G. (2006). Visual servoing of an autonomous helicopter in urban areas using feature tracking. *Journal of Field Robotics*, 23(3–4), 185–199.
- Mejias, L., Campoy, P., Mondragon, I., & Doherty, P. (2007). Stereo visual system for autonomous air vehicle navigation. In *6th IFAC symposium on intelligent autonomous vehicles (IAV 07)*, Toulouse, France.
- Milella, A., & Siegwart, R. (2006). Stereo-based ego-motion estimation using pixel tracking and iterative closest point. In *Proceedings of the fourth IEEE international conference on computer vision systems* (p. 21). Washington: IEEE Computer Society.

- Nayar, S., & Baker, S. (1997). *A theory of catadioptric image formation* (Technical report CUCS-015-97). Department of Computer Science, Columbia University.
- Nikolos, I. K., Tsourveloudis, N. C., & Valavanis, K. P. (2004). A uav vision system for airborne surveillance. In *Proceedings of the IEEE international conference on robotics and automation (ICRA '04)* (pp. 77–83), New Orleans, LA, USA.
- Nistér, D., Naroditsky, O., & Bergen, J. (2006). Visual odometry for ground vehicle applications. *Journal of Field Robotics*, 23(1), 3–20.
- Olivares, M., & Madrigal, J. (2007). Fuzzy logic user adaptive navigation control system for mobile robots in unknown environments. *Intelligent Signal Processing*. doi: [10.1109/WISP.2007.4447633](https://doi.org/10.1109/WISP.2007.4447633).
- Olivares, M., Campoy, P., Correa, J., Martinez, C., & Mondragon, I. (2008). Fuzzy control system navigation using priority areas. In *Proceedings of the 8th international FLINS conference* (pp. 987–996), Madrid, Spain.
- Olivares-Mendez, M. A., Campoy, P., Mondragon, I., & Martinez, C. (2009a). A pan-tilt camera fuzzy vision controller on an unmanned aerial vehicle. In *IEEE/RSJ international conference on intelligent robots and systems (IROS09)*.
- Olivares-Mendez, M. A., Campoy, P., Mondragon, I., & Martinez, C. (2009b). Visual servoing using fuzzy controllers on an unmanned aerial vehicle. In *Eurofuse workshop 09, preference modelling and decision analysis*.
- Puri, A., Valavanis, K., & Kontitsis, M. (2007). Statistical profile generation for traffic monitoring using real-time UAV based video data. In *Mediterranean conference on control and automation (MED'07)* (pp. 1–6).
- Scaramuzza, D., & Siegwart, R. (2008). Appearance guided monocular omnidirectional visual odometry for outdoor ground vehicles. *IEEE Transactions on Robotics*, 24(5), 1015–1026.
- Todorovic, S., Nechyba, M., & Ifju, P. (2003). Sky/ground modeling for autonomous MAV flight. *Robotics and Automation*. doi: [10.1109/ROBOT.2003.1241791](https://doi.org/10.1109/ROBOT.2003.1241791).
- Ying, X., & Hu, Z. (2004). Catadioptric camera calibration using geometric invariants. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(10), 1260–1271. doi: [10.1109/TPAMI.2004.79](https://doi.org/10.1109/TPAMI.2004.79).



**Iván F. Mondragón** studied electric engineering at Universidad Nacional de Colombia obtaining the degree of Electric Engineer (BSEE) in 2002. He joined the master program at Universidad de los Andes (Colombia) obtaining a MSc. in Electronics and Computers Engineering in 2005. From 2005 to 2006 he worked as Test Field Engineer at Siemens S.A. (Colombia). After it, he moved to DISAM (ETSII) Universidad Politécnica de Madrid (Spain) where he is currently working towards a PhD degree in Automatic and Robotics.

He is working on computer vision applied to Unmanned Aerial Vehicles in the fields of omnidirectional vision, visual tracking, visual control and servoing.



**Miguel A. Olivares-Méndez** studied computer engineering at Universidad de Malaga (Spain) obtaining his degree in 2006. After it, he moved to DISAM (ETSII) Universidad Politécnica de Madrid (Spain) where he is currently working towards a PhD degree in Automatic and Robotics. He is working on computer vision and artificial intelligence applied to Unmanned Aerial Vehicles in the fields of fuzzy logic and neural networks.



**Pascual Campoy** is Chair Professor on Automatics at the Universidad Politécnica Madrid (UPM), where he received his PhD in Engineer in 1988. He is presently teaches Control, Pattern Recognition and Neural Networks and Computer Vision at graduated and post-graduated level. He has been in charge of many research projects regarding the introduction of Computer Vision techniques in the Industry, focused in three main application areas: Automated Visual Inspection, 3D Vision and Visual In-

formation Management System. He has been member of the steering committees and task leader of 10 Projects funded by the European Commission. He is inventor of 6 patents in the field of Computer Vision, registered in 13 countries in Europe and South-America. His research activities have been gathered in over 100 international publications in technical journals and symposia contributions. He was founder of the company “I4: Innovation in Industrial Images Inspection”, which is a spin-off of the U.P.M. and it is aimed to transfer R&D results to the industry in the field of Machine Vision. He is presently leading a group for Vision on UAVs, that includes object tracking, visual control and guidance, visual SLAM, stereo and omnidirectional vision. He is presently working on new learning paradigms for bio-inspired Neural Networks that are aim to be used in image segmentation, image recognition, and image coding and retrieval.



**Carol Martínez** received the Degree in mechatronic engineering from the Universidad Autónoma de Bucaramanga, Colombia, in June 2004. She is currently working towards a PhD degree in Automatic and Robotics at the Universidad Politécnica de Madrid, Madrid, Spain. Currently she is a member of the Computer Vision Group at the UPM, whose work focuses in the use of vision systems to control UAVs. Her research areas are in the fields of visual tracking and image

processing algorithms, and 3D pose estimation and control of UAVs from external cameras.



**Luís Mejias** studied electronic engineering at UNEXPO (Venezuela) obtaining the degree of Electronic Engineer in November 1999. He joined the master program at ETSIT-Universidad Politécnica de Madrid obtaining a MSc in Network and telecommunication systems. After completing it he moved to DISAM (ETSII) Universidad Politécnica de Madrid where he obtained his doctorate. While completing his PhD he was leading the team in Vision-guided UAV for surveillance and visual inspection. He gained exten-

sive experience with Unmanned Aerial Vehicles and in particular Autonomous Helicopter platforms investigating in computer vision techniques for control and navigation of UAVs. Currently he is Lecturer in UAV at ARCAA, Queensland University of Technology.