On-board and Ground Visual Pose Estimation Techniques for UAV Control

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

Abstract In this paper, two techniques to control UAVs (Unmanned Aerial Vehicles), based on visual information are presented. The first one is based on the detection and tracking of planar structures from an on-board camera, while the second one is based on the detection and 3D reconstruction of the position of the UAV based on an external camera system. Both strategies are tested with a VTOL (Vertical take-off and landing) UAV, and results show good behavior of the visual systems (precision in the estimation and frame rate) when estimating the helicopter's position and using the extracted information to control the UAV.

Keywords Computer vision · Unmanned aerial vehicle · Homography estimation · 3D reconstruction · Visual servoing

1 Introduction

Computer vision techniques applied to the UAV field have been considered a difficult but interesting research area for the academic and industrial fields in the last years. Vision-based solutions depend on specific time-consuming steps that in the majority of situations must be accomplished together (feature detection and tracking, 3D reconstruction, or pose estimation, among others), making the operation of systems in real-time a difficult task to achieve. On the other hand, UAV systems combine abrupt changes in the image sequence (i.e. vibrations), outdoors operations (non-structured environments), and 3D information changes, that allow them to be considered a challenging testbed for computer vision techniques.

C. Martínez (✉) · I. F. Mondragón · M. A. Olivares-Méndez · P. Campoy
Computer Vision Group, Centro de Automática y Robótica (CAR),
Universidad Politécnica de Madrid, C/ José Gutiérrez Abascal 2, 28006 Madrid, Spain
e-mail: carolviviana.martinez@upm.es

Nonetheless, successful works have been achieved by integrating vision systems and UAV systems to test solutions for: object detection and object tracking [1], pose estimation [2], 3D mapping [3] , obstacle detection [4], and obstacle avoidance [5, 6], among others.

Additionally, in other successful works, it has been demonstrated that the visual information can be used in tasks such as servoing and guiding of robot manipulators and mobile robots [7, 8] combining the image processing and control techniques in such a way that the visual information is used within the control loop. This area, known as Visual Servoing [9], is used in this paper to take advantage of the variety of information that can be recovered by a vision system and use it directly in the UAV control loop.

Visual Servoing solutions can be divided into Image Based (IBVS) and Position Based Control (PBVS) Techniques, depending on the information provided by the vision system, that determines the kind of references to be sent to the control structure. They can also be divided according to the physical disposition of the visual system, into eye-in-hand systems or eye-to-hand systems [9–11]. In this paper, the latter division (eye-in-hand and eye-to-hand) is translated to the case of UAVs as on-board visual systems and ground visual systems.

Therefore, in this paper, we present solutions for the two different approaches using the UAV as a testbed to develop visual servo-control tasks. Our main objective is to test how well the visual information can be included in the UAV control loop and how this additional information can improve the capabilities of the UAV. Taking this into account, the paper is divided into the following sections: Section 2 presents the pose estimation algorithms for the on-board approach and the external vision system. In Section 3, the visual servo-control architectures where the visual information is included in the UAV control loop are described. Section 4 shows the results of the position estimation and the control task; and finally, in Section 5, conclusions and future work are presented.

2 Pose Estimation Based on Visual Information

In this section, we present two vision algorithms that are implemented by using images from an on-board camera and an external camera system. The on-board vision system is based on the detection and tracking of a planar pattern and the UAV's pose estimation is derived from a projective transformation between the image plane and the planar pattern. On the other hand, the external vision system is based on the detection of landmarks on-board the UAV. Their respective 3D reconstruction is used to recover the UAV pose. Both algorithms run at real time frame rates (≈15 fps) and their respective structures are explained in the following subsections.

2.1 On-board Vision System

The on-board system estimates the position of a plane with respect to the camera center using frame-to-frame homographies ($\mathbf{H}_{i-1}^{i}$) and the projective transformation ($\mathbf{H}_{w}^{0}$) in the first frame, to obtain for each new image the camera rotation matrix $\mathbf{R}$,

and the translation vector **t**. This method is based on the method proposed by Simon et al. [12, 13].

– **Feature extraction and tracking**

Features in the landing pattern are detected using the algorithm of good features to track presented in [14]. To track these features, appearance-based methods are used. These methods work under the premises of intensity constancy, minimum changes in position of the features between two consecutive frames, and spatial coherence of the features. Because of these constraints, traditional appearance-based methods [15] can fail when they are tested on-board the UAV as a consequence of abrupt changes in the image information due to the UAV's vibrations and displacements.

For this reason, the Pyramidal Lucas–Kanade algorithm [16] is used to solve the problems that arise when there are large and non-coherent motions between consecutive frames. This is done by first tracking features in a low scale image, obtaining an initial motion estimation, and then refining this estimation in the different pyramid levels until arriving to the original scale of the image.

With the previously mentioned techniques, the landing pattern is robustly detected and tracked, allowing in some situations partial occlusions of the pattern, as presented in Fig. 1.

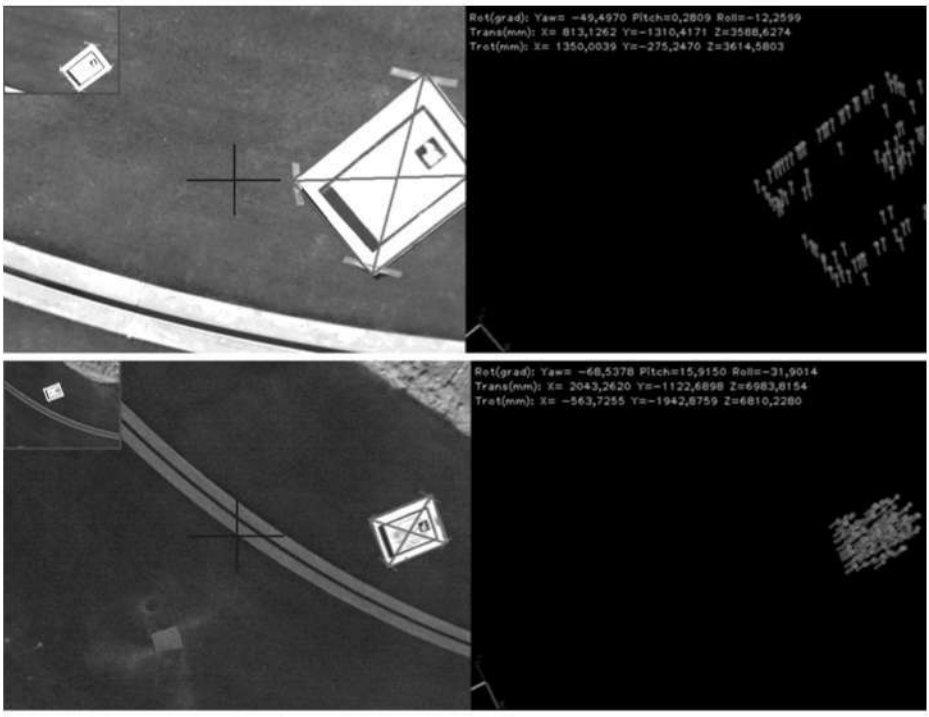

Fig. 1 Detection and tracking of a planar pattern using the pyramidal approach of the Lucas–Kanade algorithm

– **Pose Estimation**

In order to align the landing pattern (located in the world coordinate system) with the camera coordinate system, we consider the general pinhole camera model. In this model, the mapping of a point $\mathbf{x}_w$, defined in $\mathbb{P}^3$ to a point $\mathbf{x}^i$ in $\mathbb{P}^2$, can be defined as follows:

$$s\mathbf{x}^i = \mathbf{P}^i\mathbf{x}_w = \mathbf{K}\left[\mathbf{R}^i|\mathbf{t}^i\right]\mathbf{x}_w = \mathbf{K}\left[\mathbf{r}_1^i\ \mathbf{r}_2^i\ \mathbf{r}_3^i\ \mathbf{t}^i\right]\mathbf{x}_w \tag{1}$$

Where the matrix $\mathbf{K}$ is the camera calibration matrix, $\mathbf{R}^i$ and $\mathbf{t}^i$ are the rotation matrix and translation vector that relate the world coordinate system and camera coordinate system, s is an arbitrary scale factor, and the index i represents the image that is being analyzed. Figure 2 shows the relation between a world reference plane, and two images taken by a moving camera, showing the homography induced by a plane between these two frames.

If the point $\mathbf{x}_w$ is restricted to lie on a plane Π, with a coordinate system selected in such a way that the plane equation of Π is $Z = 0$, the camera projection matrix can be written as Eq. 2.

$$s\mathbf{x}^i = \mathbf{P}^i\mathbf{x}_\Pi = \mathbf{P}^i\begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix} = \langle\mathbf{P}^i\rangle\begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} \tag{2}$$

where $\langle\mathbf{P}^i\rangle = \mathbf{K}\left[\mathbf{r}_1^i\ \mathbf{r}_2^i\ \mathbf{t}^i\right]$. The deprived camera projection matrix (deprived of its third column) is a 3×3 projection matrix that transforms points from the world plane

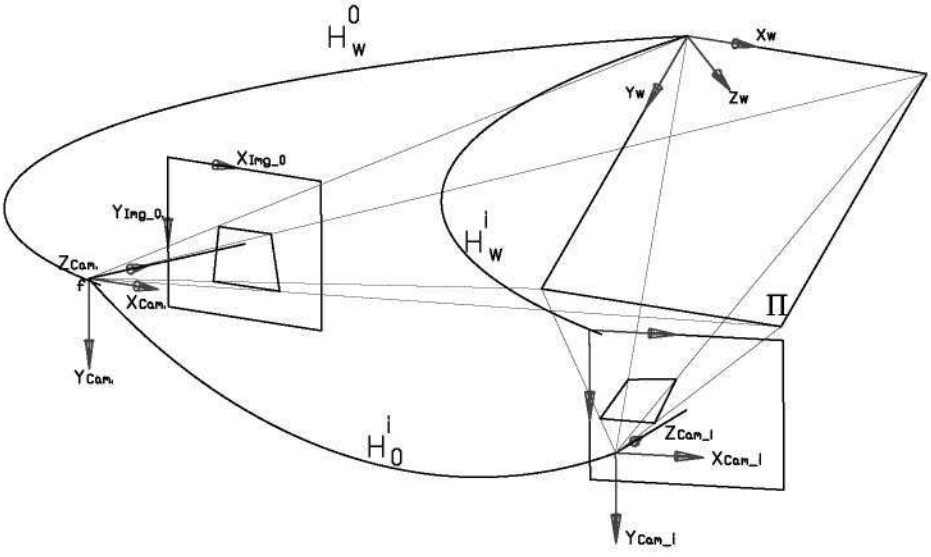

Fig. 2 Projection model on a moving camera, and the frame-to-frame homography induced by a plane

(now in $\mathbb{P}^2$) to the ith image plane. This transformation is a planar homography $\mathbf{H}_w^i$, defined up to scale factor, as presented in Eq. 3.

$$\mathbf{H}_w^i = \mathbf{K}\left[\mathbf{r}_1^i \ \mathbf{r}_2^i \ \mathbf{t}^i\right] = \langle \mathbf{P}^i \rangle \tag{3}$$

On the other hand, the world plane coordinate system is not known for the ith image. For this reason, $\mathbf{H}_w^i$ can not be directly evaluated. However, if the position of the world plane for a reference image is known, a homography $\mathbf{H}_w^0$ can be defined. Then, the ith image can be related with the reference image to obtain the homography $\mathbf{H}_0^i$. This mapping is obtained using sequential frame-to-frame homographies $\mathbf{H}_{i-1}^i$, calculated for any pair of frames (i-1,i), and used to relate the ith frame to the first image using $\mathbf{H}_0^i = \mathbf{H}_{i-1}^i \mathbf{H}_{i-2}^{i-1} \cdots \mathbf{H}_0^1$.

This mapping, and the aligning between the initial frame and the world reference plane is used to obtain the projection between the world plane and the ith image $\mathbf{H}_w^i = \mathbf{H}_0^i \mathbf{H}_w^0$.

In order to relate the world plane and the ith image, we must know the homography $\mathbf{H}_w^0$. A simple method to obtain it requires that the user selects four points on the image that correspond to corners of a rectangle in the scene, forming the matched points $(0,0) \leftrightarrow (x_1, y_1)$, $(0, \Pi_{\text{Width}}) \leftrightarrow (x_2, y_2)$, $(\Pi_{\text{Length}}, 0) \leftrightarrow (x_3, y_3)$, and $(\Pi_{\text{Length}}, \Pi_{\text{Width}}) \leftrightarrow (x_4, y_4)$. This manual selection generates a world plane defined in a coordinate frame in which the plane equation of Π is $Z = 0$. With these four correspondences between the world plane and the image plane, the minimal solution for homography $\mathbf{H}_w^0 = \left[\mathbf{h}_{\mathbf{1}w}^0 \ \mathbf{h}_{\mathbf{2}w}^0 \ \mathbf{h}_{\mathbf{3}w}^0\right]$ is obtained.

The rotation matrix and the translation vector are computed from the plane to image homography using the method described in [17].

From Eq. 3 and defining the scale factor as $\lambda = 1/s$, we have that

$$\begin{aligned} &\left[\mathbf{r}_1 \ \mathbf{r}_2 \ \mathbf{t}\right] = \lambda \mathbf{K}^{-1} \mathbf{H}_w^i = \lambda \mathbf{K}^{-1}\left[\mathbf{h}_1 \ \mathbf{h}_2 \ \mathbf{h}_3\right] \\ &\text{where} \\ &\mathbf{r}_1 = \lambda \mathbf{K}^{-1}\mathbf{h}_1, \qquad \mathbf{r}_2 = \lambda \mathbf{K}^{-1}\mathbf{h}_2, \\ &\mathbf{t} = \lambda \mathbf{K}^{-1}\mathbf{h}_3, \ \ \lambda = \frac{1}{\|\mathbf{K}^{-1}\mathbf{h}_1\|} = \frac{1}{\|\mathbf{K}^{-1}\mathbf{h}_2\|} \end{aligned} \tag{4}$$

Because the columns of the rotation matrix must be orthonormal, the third vector of the rotation matrix $\mathbf{r}_3$ can be determined by the cross product of $\mathbf{r}_1 \times \mathbf{r}_2$. However, the noise in the homography estimation causes the resulting matrix $\mathbf{R} = \left[\mathbf{r}_1 \ \mathbf{r}_2 \ \mathbf{r}_3\right]$ to not satisfy the orthonormality condition, and so we must find a new rotation matrix $\mathbf{R}'$ that best approximates to the given matrix $\mathbf{R}$ according to smallest Frobenius norm for matrices (the root of the sum of squared matrix coefficients) [17, 18]. As demonstrated by [17], this problem can be solved by forming the rotation matrix $\mathbf{R} = \left[\mathbf{r}_1 \ \mathbf{r}_2 \ (\mathbf{r}_1 \times \mathbf{r}_2)\right]$ and using the singular value decomposition (SVD) to form the new optimal rotation matrix $\mathbf{R}'$, as Eq. 5 shows:

$$\begin{aligned} &\mathbf{R} = \left[\mathbf{r}_1 \ \mathbf{r}_2 \ (\mathbf{r}_1 \times \mathbf{r}_2)\right] = \mathbf{U}\mathbf{S}\mathbf{V}^T \\ &\mathbf{S} = diag(\sigma_1, \sigma_2, \sigma_3) \\ &\mathbf{R}' = \mathbf{U}\mathbf{V}^T \end{aligned} \tag{5}$$

Thus, the solution for the camera pose problem is defined by Eq. 6.

$$\mathbf{x}^i = \mathbf{P}^i\mathbf{X} = \mathbf{K}[\mathbf{R}^i|\mathbf{t}]\mathbf{X} \tag{6}$$

Where $\mathbf{t} = [X_{\text{c.h\_cam}}, Y_{\text{c.h\_cam}}, Z_{\text{c.h\_cam}}]^t$ represents the position of the helipad with respect to the camera coordinate system, and $\mathbf{R}^i$ is the rotation matrix between the helipad coordinate system and the camera coordinate system. In order to obtain the translation refereed to the UAV coordinate system, a rotation is applied between the camera coordinate system and the UAV reference frame $\mathbf{R}_{\text{c\_UAV}}$ with the purpose of aligning the axes.

2.2 External Vision System

The external vision system is a trinocular system that estimates the position and orientation of the UAV based on the detection and tracking of on-board landmarks and their 3D reconstruction. This pose estimation algorithm was presented in [19] to estimate the UAV's position. In this paper, this algorithm is used to control the UAV based on the 3D image information recovered. The following paragraphs give an idea of the estimation algorithm.

- **Feature Extraction**

The backprojection algorithm proposed in [20] is used to extract the different landmarks on-board the UAV (color landmarks). This algorithm finds a *Ratio* histogram Rh_i^k for each landmark i in the kth camera, as defined in Eq. 7.

$$Rh_i^k(j) = \min\left[\frac{Mh_i(j)}{Ih^k(j)}, 1\right] \tag{7}$$

This ratio $Rh_i^k(j)$ represents the relationship between the bin j of a model histogram Mh_i, that defines the color we are looking for, and the bin j of the histogram of image Ih^k, which is the image of the kth camera that is being analyzed. Once Rh_i^k is found, it is then backprojected onto the image. The resulting image is a gray-scaled image, whose pixels' values represent the probability that each pixel belongs to the color we are looking for.

The location of the landmarks in the different frames are found by using the previously mentioned algorithm and the *Continuously Adaptive Mean Shift* (*CamShift*) algorithm [21]. The *CamShift* takes the probability image for each landmark i in each camera k, and moves a search window (previously initialized) iteratively in order to find the densest region (the peak), which will correspond to the object of interest (colored-landmark i). The centroid of each landmark $(\bar{x}_i^k, \bar{y}_i^k)$ is determined using the information contained inside the search window in order to calculate the zeroth $(m_{i_{00}}^k)$ and the first order moments $(m_{i_{10}}^k, m_{i_{01}}^k)$, as shown in Eq. 8.

$$\bar{x}_i^k = \frac{m_{i_{10}}^k}{m_{i_{00}}^k}; \quad \bar{y}_i^k = \frac{m_{i_{01}}^k}{m_{i_{00}}^k} \tag{8}$$

When working with overlapping FOVs (Field Of Views) in a 3D reconstruction process, it is necessary to find the relation of the information between the different cameras. This is a critical process, which requires the differentiation of features in the same image and also the definition of a metric, which tells us if the feature i in image $\mathbf{I}^1$ is the same feature i in image $\mathbf{I}^2$ (image $\mathbf{I}$ of camera k). In our case, this

Fig. 3 Feature extraction in the trinocular system. Color-based features have been considered as key-points to detect the UAV

feature-matching process has been done taking into account the color information of the different landmarks. Therefore, the features are matched by grouping only the characteristics with the same color (the central moments of each landmark) found in the different cameras, that will correspond to the cameras that are seeing the same landmarks.

These matched centroids found in the different images (as presented in Fig. 3) are then used as features for the 3D reconstruction stage.

- **3D Reconstruction and pose estimation**

Assuming that the intrinsic parameters ($\mathbf{K^k}$) and the extrinsic parameters ($\mathbf{R^k}$ and $\mathbf{t^k}$) of each camera are known (calculated through a calibration process [17]), the 3D position of the matched landmarks can be recovered by intersecting, in the 3D space, the backprojection of the rays from the different cameras that represent the same landmark.

Thus, considering a "pinhole" camera model and reorganizing the equations for each landmark i seen in each camera k, it is possible to obtain the following system of equations:

$$\begin{aligned} x_{u_i}^k &= f^k \frac{r_{11}^k x_{w_i} + r_{12}^k y_{w_i} + r_{13}^k z_{w_i} + t_x^k}{r_{31}^k x_{w_i} + r_{32}^k y_{w_i} + r_{33}^k z_{w_i} + t_z^k} \\ y_{u_i}^k &= f^k \frac{r_{21}^k x_{w_i} + r_{22}^k y_{w_i} + r_{23}^k z_{w_i} + t_y^k}{r_{31}^k x_{w_i} + r_{32}^k y_{w_i} + r_{33}^k z_{w_i} + t_z^k} \end{aligned} \tag{9}$$

Where $x_{u_i}^k$ and $y_{u_i}^k$ represent the coordinates of landmark i expressed in the *Central Camera Coordinate System* of the kth camera, r^k and t^k are the components of the rotation matrix $\mathbf{R^k}$ and the translation vector $\mathbf{t^k}$ that represent the extrinsic parameters, f^k is the focal length of each camera, and $x_{w_i}, y_{w_i}, z_{w_i}$ are the 3D coordinates of landmark i.

In Eq. 9 we have a system of two equations and three unknowns. If we consider that there are at least two cameras seeing the same landmark, it is possible to form an over-determined system of the form $\mathbf{Ac} = \mathbf{b}$, that can be solved using the least squares method, whose solution $\mathbf{c}$ will represent the 3D position $(x_{w_i}, y_{w_i}, z_{w_i})$ of the ith landmark with respect to a reference system (*World Coordinate System*), that in this case, it is located in the central camera of the trinocular system (camera 2).

Once the 3D coordinates of the landmarks on-board the UAV have been calculated, the UAV's position ($\mathbf{x}_{\mathbf{w}_{\text{uav}}}$) and its orientation with respect to the *World*

Coordinate System can be estimated using the 3D position found, and the landmark's distribution around the *Helicopter Coordinate System*.

The helicopter's orientation is defined only with respect to the Z_h axis (Yaw angle θ). We assume that the angles, with respect to the other axes, are considered to be ≈ 0 (helicopter on hover state or flying at low velocities < 4 m/s).

$$\begin{bmatrix} x_{w_i} \\ y_{w_i} \\ z_{w_i} \\ 1 \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 & x_{w_{\text{uav}}} \\ \sin(\theta) & \cos(\theta) & 0 & y_{w_{\text{uav}}} \\ 0 & 0 & 1 & z_{w_{\text{uav}}} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{h_i} \\ y_{h_i} \\ z_{h_i} \\ 1 \end{bmatrix} \tag{10}$$

Therefore, formulating 10 for each reconstructed landmark ($\mathbf{x_{w_i}}$), and taking into account the position of those landmarks with respect to the helicopter coordinate system ($\mathbf{x_{h_i}}$), it is possible to create a system of equations with five unknowns: $\cos(\theta)$, $\sin(\theta)$, $x_{w_{\text{uav}}}$, $y_{w_{\text{uav}}}$, $z_{w_{\text{uav}}}$. If at least the 3D position of two landmarks is known, this system of equations can be solved, and the solution is a 4×1 vector whose components define the orientation (*yaw* angle) and the position of the helicopter, both expressed with respect to a *World Coordinate System*.

3 Position-Based Control

The pose estimation techniques presented in Section 2 are used to develop positioning tasks of the UAV by integrating the visual information into the UAV control loop using *Position Based* control for a *Dynamic Look and Move System* [9–11].

Depending on the camera configuration in the control system, we will have an *eye-in-hand* or an *eye-to-hand* configuration. In the case of onboard control, it is considered to be an *eye-in-hand* one, as shown in Fig. 4, while in the case of ground control it is an *eye-to-hand* configuration (see Fig. 5).

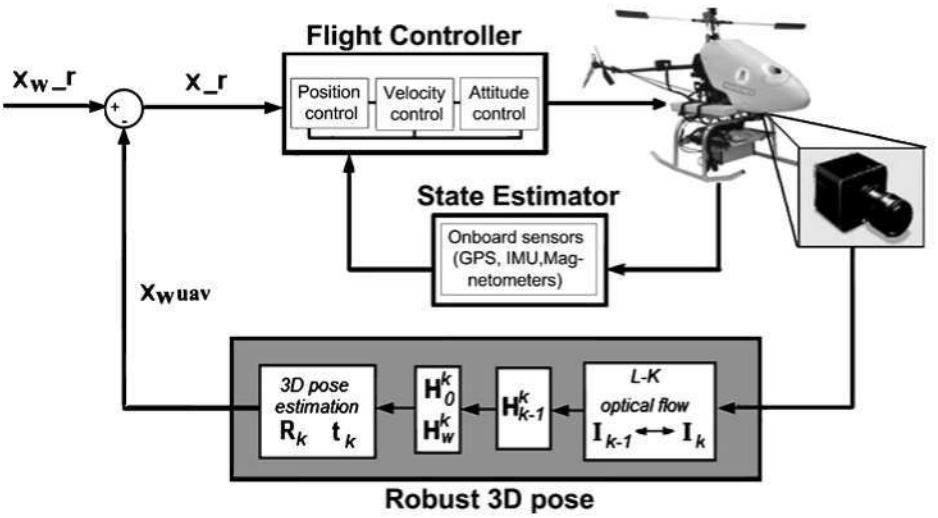

Fig. 4 *Eye-in-hand* configuration (onboard control). *Dynamic look-and-move* system architecture

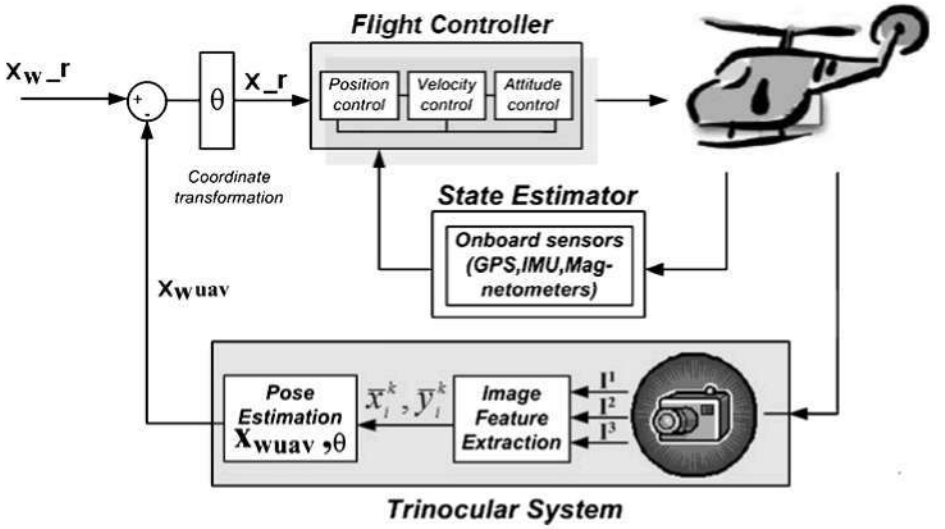

Fig. 5 *Eye-to-hand* configuration (ground control). *Dynamic look-and-move* system architecture

The *Dynamic Look and Move System*, as shown in Figs. 4 and 5, has a hierarchical structure. The vision systems (external loop) use the visual information to provide set-points as input to the low level controller (internal loop) which is in charge of the helicopter's stability. This configuration allows to control a complex system using the low rate information made available by the vision system [22].

The internal loop (Flight control system) is based on two components: a state estimator that fuses the information from different sensors (GPS, Magnetometers, IMU) to determine the position and orientation of the vehicle, and a flight controller that allows the helicopter to move to a desired position. The flight controller is based on PID controllers, arranged in a cascade formation so that each controller generates references to the next one. The attitude control reads roll, pitch, and yaw values needed to stabilize the helicopter, the velocity control generates references of roll, pitch, and collective of the principal rotor to achieve lateral and longitudinal displacements (it also allows external references), and the position controller is at the highest level of the system and is designed to receive GPS coordinates or visual-based references.

The configuration of the Flight control system, as shown in Fig. 6, allows different modes of operation. It can be configured to receive velocity commands from external references or visual features (as presented in [1]) or, on the other hand, it can be configured to receive either position commands directly from external references (GPS-based positions) or visual references derived by algorithms as the ones we have presented in this paper.

Therefore, taking into account the configuration of the flight control system presented in Fig. 6, and the information that is recovered from the vision systems, position commands are sent to the flight controller.

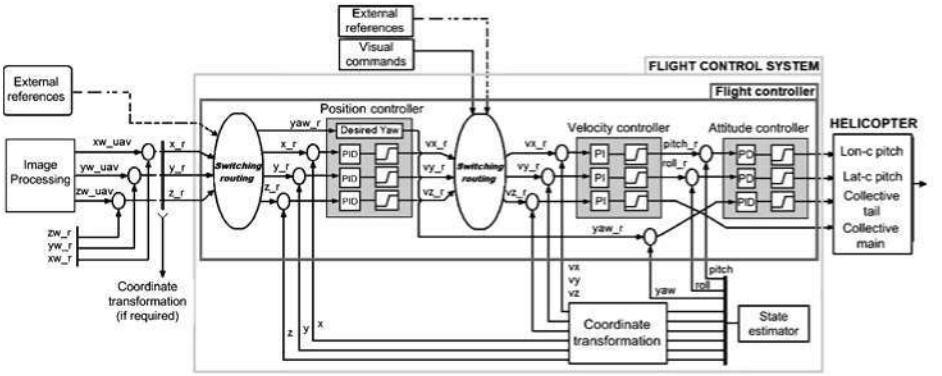

Fig. 6 Schematic flight control system. The flight control system is based on PID controllers in a cascade configuration. The position control loop can be switched to receive visual based references or GPS-based position references. The velocity references can be based on external references or visual references. Attitude control reads roll, pitch, and yaw values to stabilize the vehicle

When the onboard control is used, the vision system determines the position of the UAV with respect to the landing pattern $\mathbf{x}_{\mathbf{w}_{\mathrm{uav}}}$ (See Fig. 7). Then, this information is compared with the desired position $\mathbf{x}_{\mathbf{w\_r}}$ (which corresponds with the center of the pattern) to generate the position commands $\mathbf{x}_{\mathbf{\_r}}$ that are sent to the UAV.

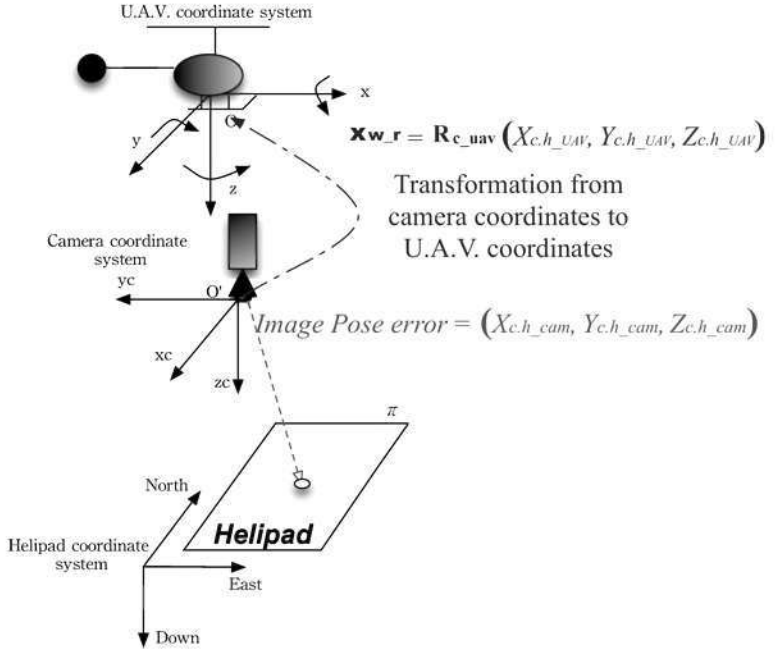

Fig. 7 Onboard vision-based control task. The vision system determines the position of the UAV with respect to the helipad. This estimation is sent as reference to the position controller in order to move the helicopter to the desired position (in our case, *the center* of the helipad at a specific height)

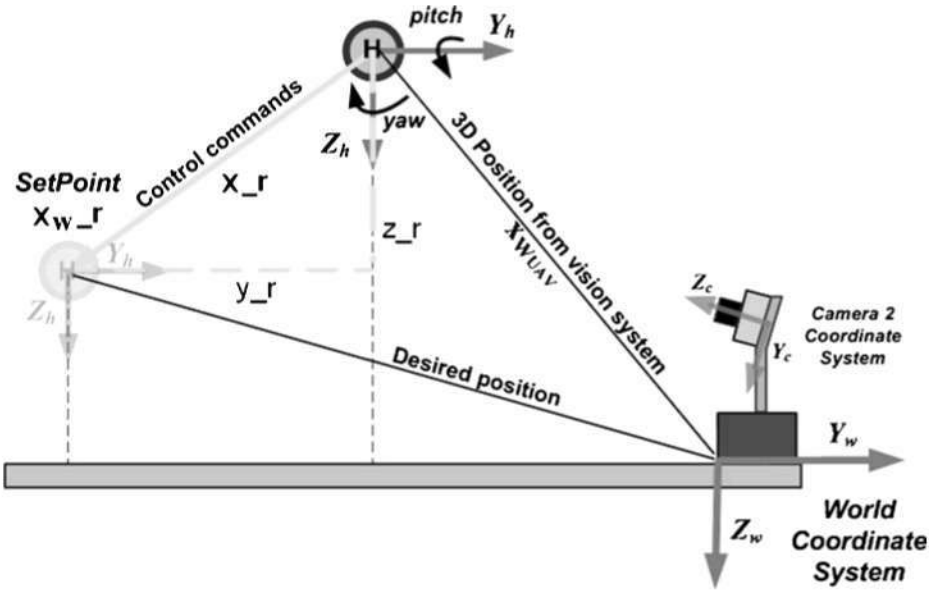

Fig. 8 External vision-based control task. A desired position expressed in the *World Coordinate System* is compared with the position information of the helicopter that is extracted by the trinocular system. This comparison generates references to the position controller in order to make the helicopter move towards the desired position

On the other hand, when the ground control is used, the vision system determines the position of the UAV in the *World Coordinate System* (located in the central camera of the trinocular system). Then, the desired position $\mathbf{x}_{w\_r}$, and the position information given by the trinocular system $\mathbf{x}_{w_{uav}}$, both defined in the *World Coordinate System*, are compared to generate references to the position controller, as shown in Fig. 8. These references are first transformed into commands to the helicopter $\mathbf{x}_r$ by taking into account the helicopter's orientation, and then these references are sent to the position controller in order to move the helicopter to the desired position (Fig. 5).

4 Experiments and Results

4.1 System Description

– **UAV system**

The experiments were carried out with the Colibri III system, presented in Fig. 9, which is a Rotomotion SR20 electric helicopter. This system belongs to the COLIBRI Project [23], whose purpose is to develop algorithms for vision-based control tasks [24]. An on-board computer running Linux OS (Operative System) is in charge of the on-board image processing. It supports FireWire cameras and

Fig. 9 Helicopter tesbed Colibri III, during a flight test

uses an 802.11 g wireless Ethernet protocol for sending and receiving information to/from the ground station. The communication of the flight system with the ground station is based on TCP/UDP messages, and uses a client-server architecture. The vision computers (the on-board computer and the computer of the external visual system) are integrated in the architecture, and through a high level layer defined by a communication API (Application Programming Interface) the communication between the different processes is achieved.

– **External camera system (Trinocular system)**

As shown in Fig. 10, the external vision system is composed of three cameras, located on an aluminium platform with overlapping FOVs. The cameras are connected to a laptop running Linux as its OS. This redundant system will allow to obtain a robust 3D position estimation by using trinocular or binocular estimation.

Fig. 10 Trinocular system and Helicopter testbed (Colibri III) during a vision-based landing task

4.2 Pose Estimation Tests

The pose estimation algorithms presented in Section 2 are tested during different flight tests, and the results are discussed comparing the visual estimation with the estimation obtained by the on-board sensors and the images taken during the flight (more tests and videos in [23]).

– **On-board Vision System**

The results of the 3D pose estimation based on a reference helipad are shown in Fig. 11. The estimated 3D pose is compared with the helicopter's position, estimated by the Kalman Filter of the flight controller in a local plane that takes as reference the takeoff point (Center of the Helipad). Because the local tangent plane to the helicopter is defined in such a way that the X axis is pointing to the North direction, the Y axis is pointing to the East direction, and the Z axis is pointing Down (negative), the measured X and Y values must be rotated according to the helicopter's heading or yaw angle, for them to be comparable with the estimated values obtained from the homographies.

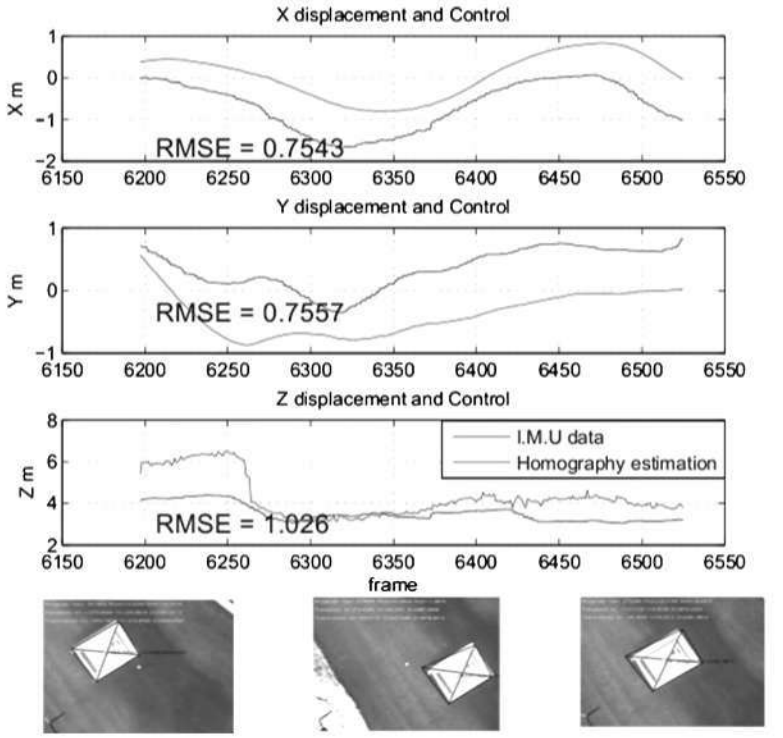

Fig. 11 Comparison between the homography estimation and IMU data

The results show that the estimated positions have the same behavior as the IMU-GPS estimated data's (on-board sensors). The RMSE (Root Mean Squared Error) between the IMU-GPS data and the values estimated by the vision system are less than 0.75 m for the X and Y axes, and around 1 m for the Z axis. This comparison only indicates that the estimated values are coherent with the one estimated by the on-board sensors. The estimation made by the on-board sensors is not considered as a ground truth because its precision is dependent on the GPS signal's quality, whose precision (in the best conditions) is approximately 0.5 m for X and Y axes, and more than 1 m for the Z axis.

– **External Vision System**

The trinocular system was used to estimate the position and orientation of the helicopter during a landing task in manual mode. In Fig. 12a–c and d, it is possible to see the results of the UAV's position estimation. In these figures, the vision-based position and orientation estimation (red lines) are compared with the estimation obtained by the on-board sensors of the UAV (green lines).

From these tests, we have found that the reconstructed values are consistent with the real movements experienced by the helicopter (analyzing the position of the UAV in the images), and also that these values have a behavior that is similar to the one estimated by the on-board sensors.

In a previous work [19], we analyzed the precision of the external visual system by comparing the visual estimation with 3D known positions. The precision that was achieved (±10 cm) in the three axes allow us to conclude that the estimation obtained

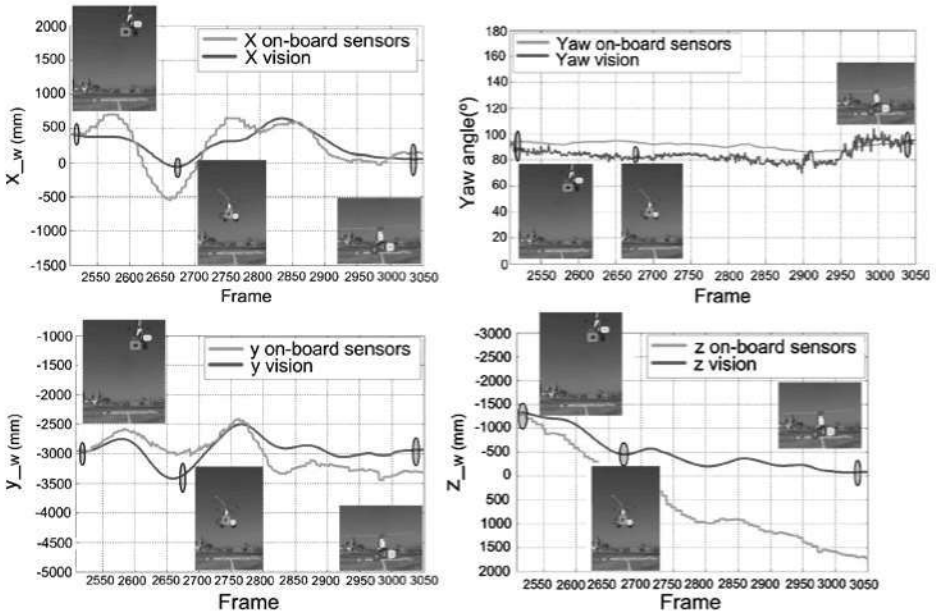

Fig. 12 Vision-based estimation vs. helicopter state estimation. The state values given by the helicopter state estimator after a *Kalman filter* (*green lines*) are compared with the trinocular estimation of the helicopter's pose (*red lines*)

by the external visual system is more accurate that the one estimated by the on-board sensors, taking into account that the errors in the position estimation using the on-board sensors, which is based on GPS information, are around ± 0.5 m for the X and Y axes, and ± 1 m for the Z axis (height estimation).

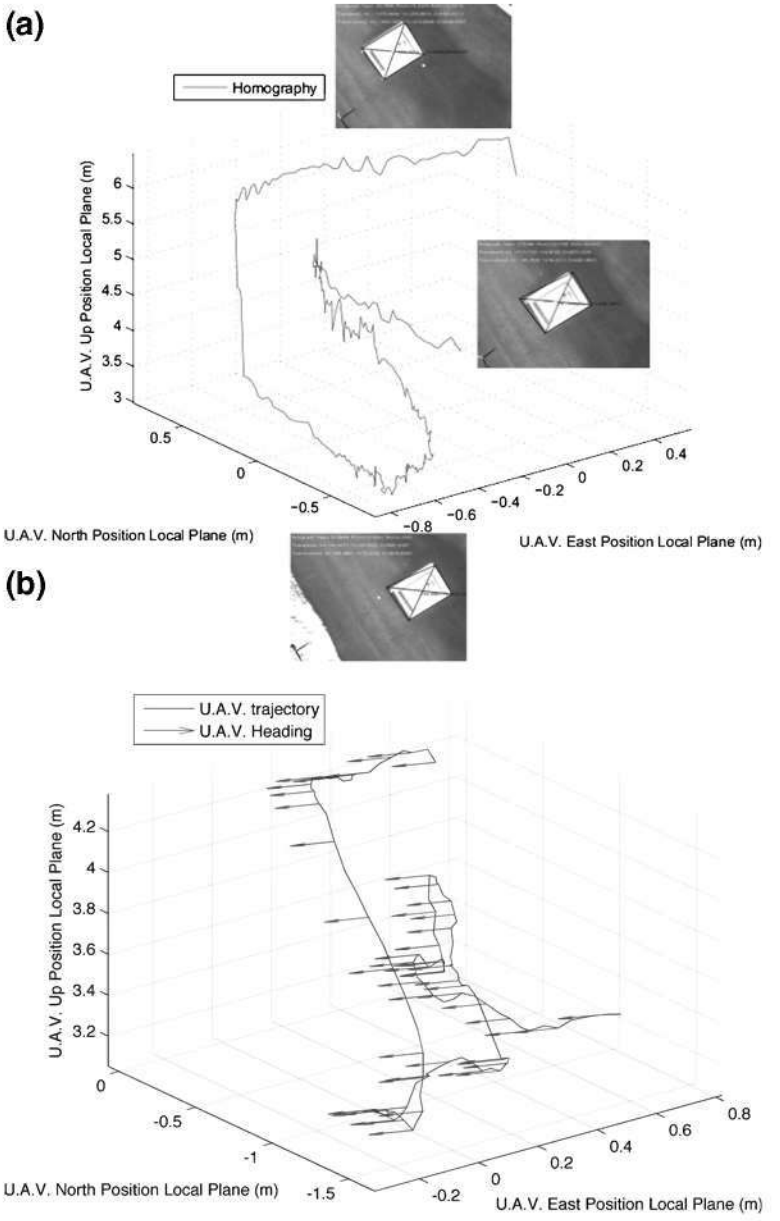

Fig. 13 3D reconstruction of the flight test using the IMU-GPS values (**a**) and the vision-based estimation (**b**). The *blue line* is the reconstructed trajectory of the UAV, and the *red arrows* correspond to the heading of the UAV

This analysis led us to use the position estimation from the on-board sensors only to compare the behavior of those signals with the one estimated by the visual system, instead of comparing the absolute values, and also led us to use the images as an approximate reference of the real position of the UAV.

On the other hand, for the yaw angle estimation, the absolute values of the orientation are compared with the values estimated by the visual system. Additionally, the images are also used as an additional reference to evaluate the results.

Taking into account the previous considerations, the results show a similar behavior between the visual estimation and the on-board sensors. On the other hand, the images show that the estimation is coherent with the real UAV position with respect to the trinocular system (e.g. helicopter moving to the left and right from the center of the image of camera 2). Analyzing the Z axis estimation, it is possible to see that the signals behave similarly (the helicopter is descending); however, when the helicopter has landed, the GPS-based estimation does not reflect that the helicopter is on the ground, whereas the visual estimation does reflect that it has landed. From the image sequence, it can be seen that the visual estimation notably improves the height estimation, which is essential for different control tasks, especially the landing task.

Regarding the *yaw* (θ) angle estimation, the results show a good correlation (RMSE of *prox* 8.3°) of the visual estimation with the IMU (Inertial Measurement Unit) estimation, whose value is taken as reference.

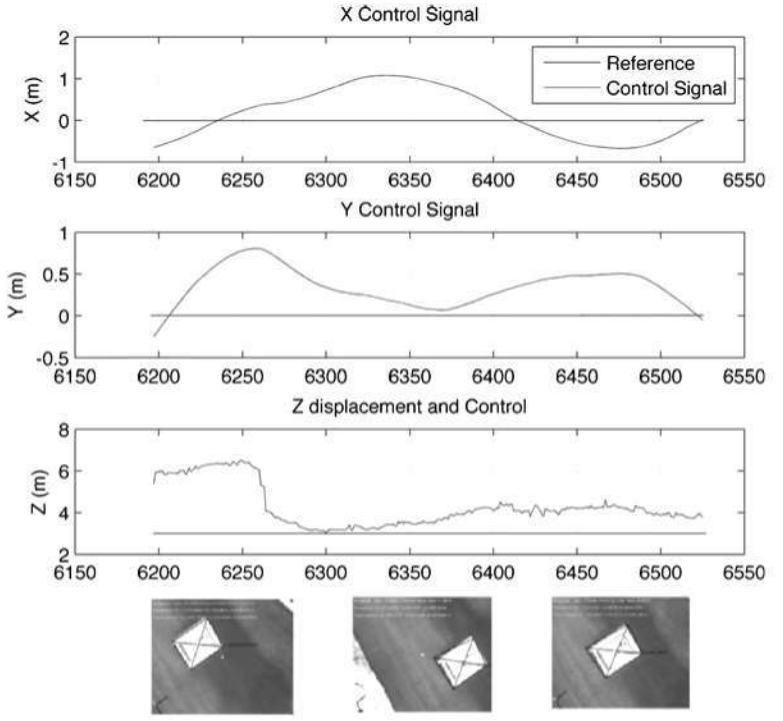

Fig. 14 On-board UAV control. Position commands are sent to the flight controller to achieve the desired position [0, 0 and 2 m]

4.3 Control Test

In these tests, the vision-based position and orientation estimations have been used to send position-based commands to the flight controller in order to develop a vision-based landing task using the control architectures presented in Figs. 4 and 5.

– **On-board Vision System**

The on-board system has been used to control the UAV position, using the helipad as a reference. For this test, a defined position above the Helipad (0, 0, 2 m) has been used in order to test the control algorithm. The test begins with the helicopter hovering over the helipad with an altitude of 6 m. The on-board visual system is used to estimate the relative position of the helipad with respect to the camera system; the obtained translation vector is used to send the reference commands to the UAV's controller with an average frame rate of 10 fps. The algorithm first centers the helicopter on the X and Y axes, and when the position error in these axes is inferior to a defined threshold (0.4 m), it begins to send references to the Z axis. In Fig. 13, the 3D reconstruction of the flight is presented.

Figure 14 shows the results of the positioning task. The red lines represent the position estimated by the visual system, which is used to generate relative position commands to the UAV controller, and the blue line represents the desired position. The test shows that the helicopter centers on X and Y axes, taking into account the resolution of the estimated pose obtained by the IMU. However, the altitude error is a little higher because the Z axis' precision is above 1 meter. Therefore, sometimes the IMU pose precision makes small references not to be executed because they are smaller than the UAV pose resolution.

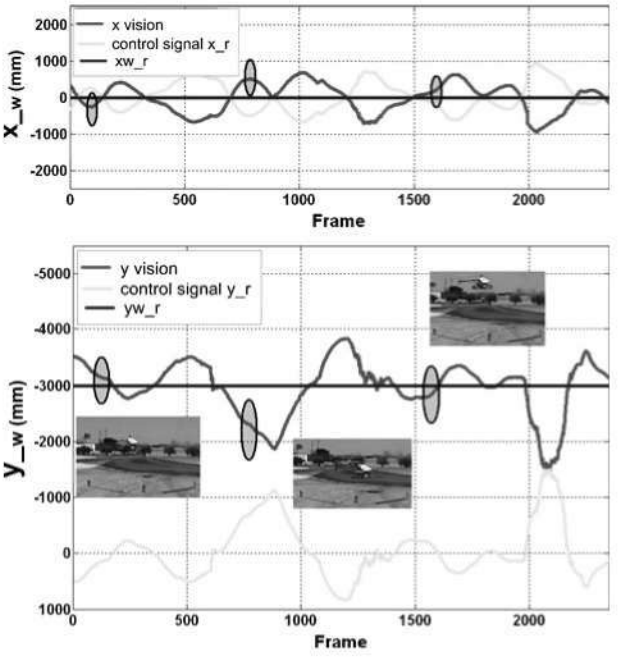

Fig. 15 UAV control using the trinocular system (X and Y axes). The vision-based position estimation (*red lines*) is used to send position commands (*yellow lines*) to the UAV flight controller in order to move the helicopter to the desired position (*blue lines*)

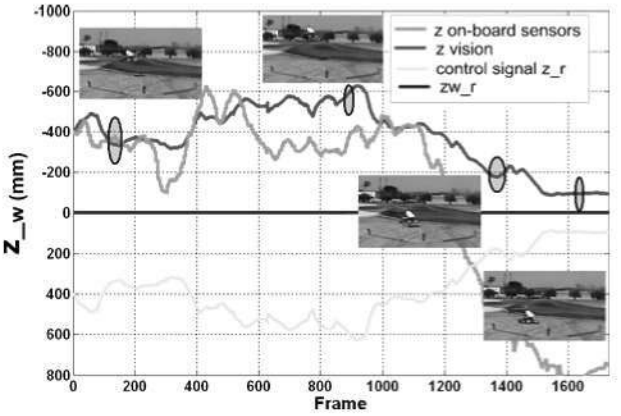

Fig. 16 UAV control using the trinocular system (Z axis). Vision-based position commands (*yellow line*) are sent to the flight controller to accomplish a vision-based landing task. The vision-based estimation (*red line*) is compared with the position estimation of the on-board sensors (*green line*) during the landing task

– **External Vision System**

The control task tested consisted in positioning the helicopter on the desired position: $xw_r = 0$ m, $yw_r = -3$ m and $zw_r = 0$ m. In the first test (Fig. 15), position-based commands in the X and Y axes (yellow lines) were generated using the vision-based position estimation (red lines). As can be seen in the figures, the commands that were sent allowed to place the helicopter around the reference position.

The second test that was carried out consisted in sending position-based commands to the Z axis in order to develop a vision-based landing task. In Fig. 16, the visual estimation (red line), the position commands that were generated (yellow line), and the height estimation obtained by the on-board sensors (green line), are presented. In this test, it was possible to accomplish a successful stable and smooth landing task using the visual information extracted by the trinocular system.

5 Conclusion

In this paper, we have presented and validated real-time algorithms that estimate the UAV's pose based on visual information extracted from either an onboard camera, or an external camera system. The paper also shows how the visual information can be used directly in the UAV control loop to develop vision-based tasks.

In Section 2.1, different computer vision techniques were presented to detect and track features with two approaches (onboard and external vision systems). Their quality has been tested in real flight tests, allowing us to detect and track the different features in a robust way in spite of the difficulties posed by the environment (outdoors, changes in light conditions, or high vibrations of the UAV, among others). Important additional results are the real-time frame rates obtained by using the proposed algorithms, that allow the use of this information to develop visual servoing tasks.

Tests have been done at different altitudes, and the estimated values have been compared with the GPS-IMU values in order to analyze the behavior of the signals. The results show coherence in the behavior of the signals. However, in the magnitude of the position estimation, it was possible to see, by analyzing the image sequences,

that the visual system is able to perceive adequately small variations in position, improving the position estimation, especially the helicopter's height estimation, whose accuracy based on GPS can reach ± 2 m.

All these results and improvements in the pose estimation of the UAV make the proposed systems suitable for maneuvers at low heights (lower that 5 m), and for situations where the GPS signal is inaccurate or unavailable. Additionally, the proposed systems improve the vehicle capabilities to accomplish tasks, such as autonomous landing or visual inspection, by including the visual information in the UAV control loop. The results in the position based control using the proposed strategy allowed to obtain a soft and stable positioning task for the height control. Future work will be oriented in exploring other control methodologies such as Fuzzy logic to obtain a stable positioning task in all axes. Additionally, our current work is focused on testing other feature extraction and tracking techniques, such as the Inverse Compositional Image Alignment Algorithm (ICIA), and on fusing the visual information with the GPS and IMU information in order to produce a unified UAV state estimation.

Acknowledgements The work reported in this paper is the product of several research stages at the Computer Vision Group of the Universidad Politécnica de Madrid. The authors would like to thank Jorge León for supporting the flight trials. We would also like to thank the Universidad Politécnica de Madrid, the Consejería de Educación de la Comunidad de Madrid and the Fondo Social Europeo (FSE) for some of the Authors' PhD Scholarships. This work has been sponsored by the Spanish Science and Technology Ministry under grant CICYT DPI 2007-66156.

References

1. Mejias, L., Saripalli, S., Campoy, P., Sukhatme, G.: Visual servoing of an autonomous helicopter in urban areas using feature tracking. Journal of Field Robotics **23**(3–4), 185–199 (2006)
2. Amidi, O., Kanade, T., Fujita, K.: A visual odometer for autonomous helicopter flight. In: Proceedings of the Fifth International Conference on Intelligent Autonomous Systems (IAS-5) (1998)
3. Artieda, J., Sebastian, J.M., Campoy, P., Correa, J.F., Mondragón, I.F., Martínez, C., Olivares, M.: Visual 3-d slam from uavs. J. Intell. Robot. Syst. **55**(4–5), 299–321 (2009)
4. McGee, T.G., Sengupta, R., Hedrick, K.: Obstacle detection for small autonomous aircraft using sky segmentation. In: Proceedings of the 2005 IEEE International Conference on Robotics and Automation, 2005. ICRA 2005. pp. 4679–4684 (2005)
5. He, Z., Iyer, R.V., Chandler, P.R.: Vision-based uav flight control and obstacle avoidance. In: American Control Conference, 2006, 5 pp. (2006)
6. Carnie, R., Walker, R., Corke, P.: Image processing algorithms for uav "sense and avoid". In: Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006, pp. 2848–2853 (2006)
7. Conticelli, F., Allotta, B., Khosla, P.K.: Image-based visual servoing of nonholonomic mobile robots. In: Proceedings of the 38th IEEE Conference on Decision and Control, 1999. vol. 4, pp. 3496–3501 (1999)
8. Mariottini, G.L., Oriolo, G., Prattichizzo, D.: Image-based visual servoing for nonholonomic mobile robots using epipolar geometry. IEEE Trans. Robot. **23**(1), 87–100 (2007)
9. Siciliano, B., Khatib, O., (eds.): Springer Handbook of Robotics. Springer, Berlin (2008)
10. Hutchinson, S., Hager, G.D., Corke, P.: A tutorial on visual servo control. IEEE Trans. Robot. Autom. **12**(5), 651–670 (1996)
11. Chaumette, F., Hutchinson, S.: Visual servo control. I. basic approaches. IEEE Robot. Autom. Mag. **13**(4), 82–90 (2006)
12. Simon, G., Fitzgibbon, A.W., Zisserman, A.: Markerless tracking using planar structures in the scene. In: Proceedings. IEEE and ACM International Symposium on Augmented Reality, 2000. (ISAR 2000), pp. 120–128 (2000)

13. Simon, G., Berger, M.-O.: Pose estimation for planar structures. IEEE Comput. Graph. Appl. **22**(6), 46–53 (2002)
14. Shi, J., Tomasi, C.: Good features to track. In: 1994 IEEE Conference on Computer Vision and Pattern Recognition (CVPR '94), pp. 593–600 (1994)
15. Baker, S., Matthews, I.: Lucas–kanade 20 years on: a unifying framework: part 1. Technical Report CMU-RI-TR-02-16, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA (2002)
16. Bouguet, J.-Y.: Pyramidal implementation of the lucas–kanade feature tracker. Technical report, Intel Corporation. Microprocessor Research Labs, Santa Clara, CA 95052 (1999)
17. Zhang, Z.: A flexible new technique for camera calibration. IEEE Trans. Pattern Anal. Mach. Intell. **22**(11), 1330–1334 (2000)
18. Sturm, P.: Algorithms for plane-based pose estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1010–1017. Hilton Head Island, South Carolina (2000)
19. Martínez, C., Campoy, P., Mondragon, I., Olivares, M.: Trinocular Ground System to Control UAVs. In: Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems. IROS, pp. 3361–3367 (2009)
20. Swain, M.J., Ballard, D.H.: Color indexing. Int. J. Comput. Vis. **7**(1), 11–32 (1991)
21. Bradski, G.R.: Computer vision face tracking for use in a perceptual user interface. Intel Technol. J. **2**(2), 12–21 (1998)
22. Kragic, S.D., Christensen, H.I.: Survey on visual servoing for manipulation. Tech. Rep ISRN KTH/NA/P–02/01–SE, Centre for Autonomous Systems, Numerical Analysis and Computer Science, Royal Institute of Technology, Stockholm, Sweden, Fiskartorpsv. 15 A 100 44 Stockholm. January 2002. Available at www.nada.kth.se/~danik/VSpapers/report.pdf
23. Computer Vision Group. Universidad Politécnica de Madrid. CVG 2010. http://www.vision4uav.com
24. Campoy, P., Correa, J.F., Mondragón, I., Martínez, C., Olivares, M., Mejías, L., Artieda, J.: Computer vision onboard UAVs for civilian tasks. J Intell. Robot Syst. **54**(1–3), 105–135 (2009)