2010 IEEE International Conference on Robotics and Automation
Anchorage Convention District
May 3-8, 2010, Anchorage, Alaska, USA

3D pose estimation based on planar object tracking for UAVs control.

Iván F. Mondragón and Pascual Campoy and Carol Martínez and Miguel A. Olivares-Méndez
Computer Vision Group
Universidad Politécnica de Madrid
C. José Gutiérrez Abascal 2, 28006 Madrid, Spain
imondragon@etsii.upm.es

Abstract— **This article presents a real time Unmanned Aerial Vehicles UAVs 3D pose estimation method using planar object tracking, in order to be used on the control system of a UAV. The method explodes the rich information obtained by a projective transformation of planar objects on a calibrated camera. The algorithm obtains the metric and projective components of a reference object (landmark or helipad) with respect to the UAV camera coordinate system, using a robust real time object tracking based on homographies. The algorithm is validated on real flights that compare the estimated data against that obtained by the inertial measurement unit IMU, showing that the proposed method robustly estimates the helicopter's 3D position with respect to a reference landmark, with a high quality on the position and orientation estimation when the aircraft is flying at low altitudes, a situation in which the GPS information is often inaccurate. The obtained results indicate that the proposed algorithm is suitable for complex control tasks, such as autonomous landing, accurate low altitude positioning and dropping of payloads.**

I. INTRODUCTION

Autonomous aerial vehicles have been an active area of research for several years. They have been used as testbeds to investigate problems ranging from control, navigation and path planning to object detection and tracking, as well as visual navigation. Several teams from MIT, Stanford, Berkeley, ARCAA and USC among others, have had an ongoing UAV project for the past decade. The reader is referred to [1] for a good overview of the various types of vehicles and algorithms used for their control. Some of the recent work in this field, includes autonomous landing [2] [3], visual servoing [4], obstacle avoidance [5] [6].

Our research interest focuses on developing computer vision techniques to provide UAVs with an additional source of information to perform visually guided task - this includes tracking and visual servoing, inspection, autonomous landing and positioning, or ground-air cooperation. These situations needs reliable state information, that allows a onboard controller to generate accurate positioning. In general the pose information is estimated based on the the GPS and IMU sensor measurements. However, for low altitude tasks or in urban scenarios, the estimation often is inaccurate because it is affected by GPS dropouts, thus making flying in these constrained environments more vulnerable and more prone to problems. Computer vision as passive sensor not only offers a rich source of information for navigational purposes, but it can be also used as a main navigational sensor in place of GPS. With the increasing interest in UAVs, a visual system that can determine the robot 3D location in its operational environment is becoming a key sensor for civil applications.

Different works have been done where a vision system was used for low altitude position estimation and autonomous landing. In [7], the authors have evaluated the use of visual information at different stages of a UAV control system, including a visual controller and a pose estimation for autonomous landing using a checkboard pattern. Saripalli *et. al.* have proposed and experimental method for autonomous landing on a moving target, [2], [8], by tracking a known helipad and using it to complement the controller GPS-IMU state estimation. Hrabar *et. al.* [9] have used omnidirectional vision in order to generate control commands for a visual servoing using the centroid of known visual targets. In addition, 3D pose relative to a landing pad, estimated using a visual system, have also been used for an autonomous landing of a Multirotor, as is proposed in [10].

This paper presents a robust real time 3D pose and orientation estimation method based on the tracking of a piecewise planar object using robust homographies estimation for visual control, using our previous visual control architecture developed for UAVs [4]. Section II explains how the pose of a planar object relative to a moving camera coordinate center is obtained, using frame-to-frame homographies and the projective transformation of the reference object on the image plane. Section III explain the visual algorithm used in order to robustly track the reference landmark or helipad. The integration of the developed system for control a UAV electric helicopter is presented in section IV. Finally, section V show the test results of the proposed algorithm running onboard a UAV, by comparing the estimated 3D pose data with the one given by the inertial Measurement Unit IMU. This validates our approach for an autonomous landing control based in visual information.

II. 3D ESTIMATION BASED ON HOMOGRAPHIES

In this section, a 3D pose estimation method based on projection matrix and homographies is explained. The method estimates the position of a world plane relative to the camera projection center for every image sequence using previous frame-to-frame homographies and the projective transformation at first, obtaining for each new image, the camera rotation matrix **R** and a translational vector **t**. This method is based on the propose by Simon *et. al.* [11], [12].

978-1-4244-5040-4/10/$26.00 ©2010 IEEE

A. World plane projection onto the Image plane

In order to align the planar object on the world space and the camera axis system, we consider the general pinhole camera model and the homogeneous camera projection matrix, that maps a world point $\mathbf{x}_w$ in $\mathbb{P}^3$ to a point $\mathbf{x}^i$ on i^{th} image in $\mathbb{P}^2$, defined by equation 1:

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

where the matrix $\mathbf{K}$ is the camera calibration matrix, $\mathbf{R}^i$ and $\mathbf{t}^i$ are the rotation and translation that relates the world coordinate system and camera coordinate system, and s is an arbitrary scale factor. Figure 1 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.

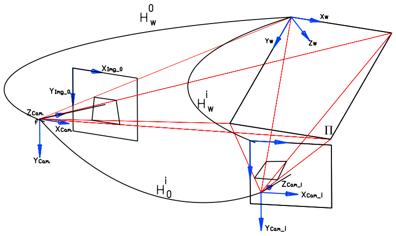


Fig. 1. Projection model on a moving camera and frame-to-frame homography induced by a plane.

If 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 equation 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 $\langle\mathbf{P}^i\rangle$ denotes that this matrix is deprived on its third column or $\langle\mathbf{P}^i\rangle = \mathbf{K}\begin{bmatrix}\mathbf{r}_1^i & \mathbf{r}_2^i & \mathbf{t}^i\end{bmatrix}$. The deprived camera projection matrix is a 3×3 projection matrix, which transforms points on the world plane (now in $\mathbb{P}^2$) to the i^{th} image plane (likewise in $\mathbb{P}^2$), that is none other that a planar homography $\mathbf{H}_w^i$ defined up to scale factor as equation 3 shows.

$$\mathbf{H}_w^i = \mathbf{K}\begin{bmatrix}\mathbf{r}_1^i & \mathbf{r}_2^i & \mathbf{t}^i\end{bmatrix} = \langle\mathbf{P}^i\rangle \tag{3}$$

Equation 3 defines the homography which transforms points on the world plate to the i^{th} image plane. Any point on the world plane $\mathbf{x}_\Pi = [x_\Pi, y_\Pi, 1]^T$ is projected on the image plane as $\mathbf{x} = [x, y, 1]^T$. Because the world plane coordinates system is not know for the i^{th} image, $\mathbf{H}_w^i$ can not be directly evaluated. However, if the position of the word plane for a reference image is known, a homography $\mathbf{H}_w^0$, can be defined. Then, the i^{th} 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 i^{th} frame to the first imagen $\mathbf{H}_0^i$ using equation 4:

$$\mathbf{H}_0^i = \mathbf{H}_{i-1}^i\mathbf{H}_{i-2}^{i-1}\cdots\mathbf{H}_0^1 \tag{4}$$

This mapping and the aligning between initial frame to world plane reference is used to obtain the projection between the world plane and the i^{th} image $\mathbf{H}_w^i = \mathbf{H}_0^i\mathbf{H}_w^0$. In order to relate the world plane and the i^{th} image, we must know the homography $\mathbf{H}_w^0$. A simple method to obtain it, requires that a user selects four points on the image that correspond to corners of rectangle in the scene, forming the matched points $(0,0) \leftrightarrow (x_1,y_1)$, $(0,\Pi_{Width}) \leftrightarrow (x_2,y_2)$, $(\Pi_{Lenght},0) \leftrightarrow (x_3,y_3)$ and $(\Pi_{Lenght},\Pi_{Width}) \leftrightarrow (x_4,y_4)$. 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 $\mathbf{H}_w^0 = \begin{bmatrix}\mathbf{h1}_w^0 & \mathbf{h2}_w^0 & \mathbf{h3}_w^0\end{bmatrix}$ is obtained using the method described on section III-B. The rotation matrix and the translation vector are computed from the plane to image homography using the method described in [13].

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

$$\begin{gathered}\begin{bmatrix}\mathbf{r}_1 & \mathbf{r}_2 & \mathbf{t}\end{bmatrix} = \lambda\mathbf{K}^{-1}\mathbf{H}_w^i = \lambda\mathbf{K}^{-1}\begin{bmatrix}\mathbf{h}_1 & \mathbf{h}_2 & \mathbf{h}_3\end{bmatrix}\\ \text{where}\\ \mathbf{r}_1 = \lambda\mathbf{K}^{-1}\mathbf{h}_1,\quad \mathbf{r}_2 = \lambda\mathbf{K}^{-1}\mathbf{h}_2,\quad \mathbf{t} = \lambda\mathbf{K}^{-1}\mathbf{h}_3\end{gathered} \tag{5}$$

The scale factor λ can be calculated using equation 6:

$$\lambda = \frac{1}{\|\mathbf{K}^{-1}\mathbf{h}_1\|} = \frac{1}{\|\mathbf{K}^{-1}\mathbf{h}_2\|} \tag{6}$$

Because the columns of the rotation matrix must be orthonormal, the third vector of the rotation matrix $\mathbf{r}_3$ could be determined by the cross product of $\mathbf{r}_1 \times \mathbf{r}_2$. However, the noise on the homography estimation causes that the resulting matrix $\mathbf{R} = \begin{bmatrix}\mathbf{r}_1 & \mathbf{r}_2 & \mathbf{r}_3\end{bmatrix}$ does not satisfy the orthonormality condition and 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) [14] [13]. As demonstrated by [13], this problem can be solved by forming the Rotation Matrix $\mathbf{R} = \begin{bmatrix}\mathbf{r}_1 & \mathbf{r}_2 & \mathbf{r}_2\end{bmatrix}$ and using singular value decomposition (SVD) to form the new optimal rotation matrix $\mathbf{R}'$ as equation 7 shows:

$$\begin{aligned}\mathbf{R} &= \begin{bmatrix}\mathbf{r}_1 & \mathbf{r}_2 & (\mathbf{r}_1\times\mathbf{r}_2)\end{bmatrix} = \mathbf{USV}^T\\ \mathbf{S} &= diag(\sigma_1,\sigma_2,\sigma_3)\\ \mathbf{R}' &= \mathbf{UV}^T\end{aligned} \tag{7}$$

The solution for the camera pose problem is defined by equation 8:

$$\mathbf{x}^i = \mathbf{P}^i\mathbf{X} = \mathbf{K}[\mathbf{R}'|\mathbf{t}]\mathbf{X} \tag{8}$$

The translational vector obtained is already scaled based on the dimensions defined for the reference plane during the alignment between the helipad and image I_0, so if the dimensions of the world rectangle are defined in *mm*, the resulting vector $\mathbf{t}_w^i$ is also in *mm*. The Rotation Matrix can be decomposed in order to obtain the Tait-Bryan or Cardan Angles, which is one of the preferred rotation sequences in flight and vehicle dynamics. Specifically, these angles are formed by the sequence: (1) ψ about z axis (yaw $\mathbf{R}_{z,\psi}$), (2) θ about y_a (pitch $\mathbf{R}_{y,\theta}$), and (3) ϕ about the final x_b axis (roll $\mathbf{R}_{x,\phi}$), where a and b denote the second and third stage in a three-stage sequence or axes. The final coordinate transformation matrix for Tait-Bryan angles is defined by the composition of the rotations $\mathbf{R}_{Tait-Bryan} = \mathbf{R}_{x,\phi}\mathbf{R}_{y,\theta}\mathbf{R}_{z,\psi}$. Defining $\mathrm{s}\theta = \sin\theta$, $\mathrm{c}\theta = \cos\theta$, $\mathrm{s}\psi = \sin\psi$, $\mathrm{c}\psi = \cos\psi$, $\mathrm{s}\phi = \sin\phi$ and $\mathrm{c}\phi = \cos\phi$, the Tait-Bryan Rotation matrix is expressed as equation 9:

$$\mathbf{R}_{Tait-Bryan} = \begin{bmatrix} \mathrm{c}\theta\mathrm{c}\psi & \mathrm{c}\theta\mathrm{s}\psi & -\mathrm{s}\theta \\ \mathrm{s}\phi\mathrm{s}\theta\mathrm{c}\psi - \mathrm{c}\phi\mathrm{s}\psi & \mathrm{s}\phi\mathrm{s}\theta\mathrm{s}\psi + \mathrm{c}\phi\mathrm{c}\psi & \mathrm{s}\phi\mathrm{c}\theta \\ \mathrm{c}\phi\mathrm{s}\theta\mathrm{c}\psi + \mathrm{s}\phi\mathrm{s}\psi & \mathrm{c}\phi\mathrm{s}\theta\mathrm{s}\psi - \mathrm{s}\phi\mathrm{c}\psi & \mathrm{c}\phi\mathrm{c}\theta \\ & & \end{bmatrix}$$

$$\mathbf{R}_{Tait-Bryan} = \mathbf{R}_w^i = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \tag{9}$$

The angles ψ, θ and ϕ can be obtained from the Rotation Matrix $\mathbf{R}_w^i$ (remember the rotation sequence order) using the equation 10.

$$\theta = -\arcsin(r_{13}), \psi = \arcsin(\tfrac{r_{12}}{\cos\theta}), \phi = \arcsin(\tfrac{r_{23}}{\cos\theta}) \tag{10}$$

III. Visual Processing

This section explains how the frame-to-frame homography is estimated using matched points and robust model fitting algorithms. For it, the pyramidal Lucas-Kanade optical flow [15] on corners detected using the method of Shi and Tomasi [16] is used to generate a set of corresponding points, then, a RANSAC [17] algorithm is used to robustly estimate projective transformation between the reference object and the image.

A. Pyramidal Lucas Kanade Optical Flow.

On images with high motion, good matched features can be obtained using the Pyramidal Lucas-Kanade algorithm modification [15]. It is used to solve the problem that arise when large and non-coherent motion are present between consecutive frames, by first tracking features over large spatial scales on the pyramid image, obtaining an initial motion estimation, and then refining it by down sampling the levels of the images pyramid until it arrives at the original scale.

The overall pyramidal tracking algorithm proceeds as follows: first, a pyramidal representation of a image I of size $widthpixels \times heightpixels$ is generated. The zeroth level is composed by the original image and defined as I^0, then pyramids levels are recursively computed by downsampling the last available level (compute I^1 form I^0, then I^2 from I^1 and so on until I^{L_m} form I^{L-1})). Typical maximum pyramids Levels L_m are 2,3 and 4. Then, the optical flow is computed at the deepest pyramid level L_m. The result of that computation is propagated to the upper level $L_m - 1$ in a form of an initial guess for the pixel displacement (at level $L_m - 1$). Given that initial guess, the refined optical flow is computed at level $L_m - 1$, and the result is propagated to level $L_m - 2$ and so on up to the level 0 (the original image).

B. Homography calculation

Here we will focus on estimating the 2D projective transformation that given a set of points $\bar{\mathbf{x}}_i$ in $\mathbb{P}^2$ and a corresponding set of points $\bar{\mathbf{x}}_i'$ in $\mathbb{P}^2$, compute the 3x3 matrix $\mathbf{H}$ that takes each $\bar{\mathbf{x}}_i$ to $\bar{\mathbf{x}}_i'$ or $\bar{\mathbf{x}}_i' = \mathbf{H}\bar{\mathbf{x}}_i$. Taking into account that the number of degrees of freedom of the projective transformation is eight (defined up to scale) and because each point to point correspondence $(x_i, y_i) \leftrightarrow (x_i', y_i')$ gives rise to two independent equations in the entries of $\mathbf{H}$. Four correspondences are enough to have a exact solution or minimal solution. If matrix $\mathbf{H}$ is written in the form of a the vector $\mathbf{h} = [h_{11}, h_{12}, h_{13}, h_{21}, h_{22}, h_{23}, h_{31}, h_{32}, h_{33}]^t$ the homogeneous equations $\bar{\mathbf{x}}' = \mathbf{H}\bar{\mathbf{x}}$ for n points could be formed as $\mathbf{A}\mathbf{h} = 0$, with $\mathbf{A}$ a $2n \times 9$ which in can be solved using the Inhomogeneous method [18]. In this method, one of the nine matrix elements is given a fixed unity value, forming an equation of the form $\mathbf{A}'\mathbf{h}' = \mathbf{b}$ as is shown on equation 11

$$\begin{bmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 & -x_1x_1' & -y_1x_1' \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -x_1y_1' & -y_1y_1' \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_n & y_n & 1 & 0 & 0 & 0 & -x_nx_n' & -y_nx_n' \\ 0 & 0 & 0 & x_n & y_n & 1 & -x_ny_n' & -y_ny_n' \end{bmatrix} \begin{bmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \end{bmatrix} = \begin{bmatrix} x_1' \\ y_1' \\ \vdots \\ x_n' \\ y_n' \end{bmatrix} \tag{11}$$

The resulting simultaneous equations for the 8 unknown elements are then solved using a Gaussian elimination in the case of a minimal solution or using a pseudo-inverse method in case of an over-determined system [19].

C. Homography robust estimation using RANSAC

Homography is calculated using a set of corresponding or matched points between two images ($(x_i, y_i) \leftrightarrow (x_i', y_i')$ for $i = 1 \ldots n$,), which often has two error sources. The first one is the measurement of the point position, which follows a Gaussian distribution. The second one is the *outliers* to the Gaussian error distribution, which are the mismatched points given by the selected algorithm. These outliers can severely disturb the estimated homography, and consequently alter any measurement based on homographies. In order to select a set of *inliers* from the total set of correspondences so that the homography can be estimated employing only the set of pairs considered as inliers, *robust estimation* using Random Sample Consensus (RANSAC) algorithm [17] is used. It achieves its goal by iteratively selecting a random subset of the original data points by testing it to obtain the model and evaluating the model consensus, which is the total number of original data points that best fit the model. This procedure is then repeated a fixed number of times, each time producing either a model which is rejected because too few points are classified as inliers, or a refined model. When total trials are reached, the algorithm return the Homography with the largest number of inliers. The Algorithm 1 shows the general steps to obtain a robust homography. Further description can be found in [19], [17].

Algorithm 1 Homography estimation using RANSAC

```
Require: Set of matched points $\mathbf{x}_i = (x_i, y_i) \leftrightarrow \mathbf{x}'_i = (x'_i, y'_i)$ for $i = 1 \dots n$
  Define $s =$ Minimun set of points to estimate the minimal solution ($s = 4$ for the Homography)
  Define $p =$ Probability that al least one of the random samples is free form outliers
  Define $t =$ distance threshold to consider a point as an inlier for some model.
  Define $\varepsilon =$ Initial probability that any selected point is an outlier.
  Define $Concesus =$ Desired number of minimum Inliers based on the total number of matched points
  Calculate the maximum number os samples $N = \log(1-p)/\log(1-(1-\varepsilon)^s)$
  while $N > Trials$ do
    Randomly select $s$ pairs of matched points
    Calculate the minimal solution for the model under test (Homography) using selected $s$ points
    $inliers = 0$
    for $i = 0$ to $n$ do
      Calculate the distance $d^2_{transfer} = d(\mathbf{x}'_i, \mathbf{H}\mathbf{x}_i)^2 + d(\mathbf{x}_i, \mathbf{H}^{-1}\mathbf{x}'_i)^2$
      if $d_{transfer} < t$ then
        $inliers = inliers + 1$
      end if
    end for
    if $inliers > Concensus$ then
      Calculate the Homography using all inliers points
      $Concensus = inliers$
    end if
    recalculate $\varepsilon = 1 - (inliers/n)$
    recalculate $N = \log(1-p)/\log(1-(1-\varepsilon)^s)$
    $Trials = Trials + 1$
  end while
```

IV. UAV System and visual control system description.

The Colibri project has three operational UAV platforms: one electric helicopter and two gasoline-powered helicopters [20] (figure (2)). The COLIBRI testbeds [4] are equipped with an Xscale-based flight computer augmented with sensors (GPS, IMU, Magnetometer, fused with a Kalman filter for state estimation). Additionally they include 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.

Fig. 2. COLIBRI III Electric helicopter UAV used for pose estimation tests.

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, forming a *dynamic look-and-move* [21] servoing architecture as figure 3 shows. The helicopter's low-level controller is based on PID control loops to ensure its stability. Because features are extracted in the image and then used to estimate the pose of the helipad or target with respect to the camera coordinate system (fixed on the UAV camera platform), our control scheme is considered to be *Position Base Visual Servoing* (PBVS) system [21], [22], [23]. In this kind of control, an error between the current and the desired pose of the camera-UAV is calculated and used by the low level onboard controller to generate the control references for positioning the UAV according with the measured error. Depending on the control task, a reference point in coordinates relative to the helipad will be defined (For landing the reference point will be (0,0,0)). Because, the estimated position of the helipad (relative to the camera coordinate system on the UAV) is know by the

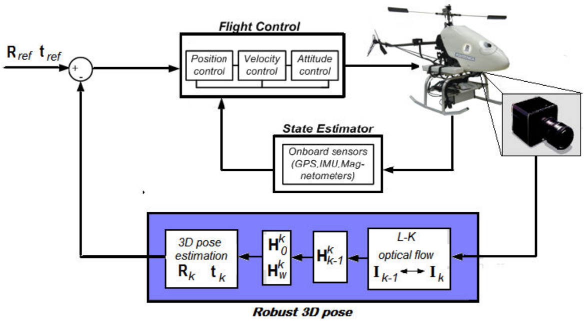


Fig. 3. UAV onboard visual control system following a *dynamic look-and-move* architecture

visual system, the reference point can be transformed to coordinates relative to the helicopter coordinate system and will be used to generate the references (X,Y,Z) and (*Heading*) commands, relative to the UAV coordinate system, that will be used by the low-level controller to position the helicopter (in the landing case the command will be the translation vector obtained by the visual system).

V. UAV 3D ESTIMATION TESTS AND RESULTS

This section shows the pose estimation tests using the Colibri 3 Electric UAV and visual control architecture explained in section IV. For these test a Monocromo CCD Firewire camera with a resolution of 640x480 pixels is used. The camera is calibrated before each test, so the intrinsic parameters are know. The camera is installed in such a way that it is looking downward with relation to the UAV. A know rectangular helipad is used as the reference object to which estimate the UAV 3D position. It is aligned in such a way that its axes are parallel to the local plane North East axes. This helipad was designed in such a way that it produces many distinctive corners for the visual tracking. Figure 4, shows the helipad used as reference and figure 5, shows the coordinate systems involved in the pose estimation. For these tests a series of flights in autonomous mode at different heights were done. The test begins when the UAV is hovering over the helipad. Then a user manually selects four point on the image that corresponds to four corners on the helipad, forming the matched points $(0,0) \leftrightarrow (x_1,y_1)$, $(910mm,0) \leftrightarrow (x_2,y_2)$, $(0,1190mm) \leftrightarrow (x_3,y_3)$ and $(910mm,1190mm) \leftrightarrow (x_4,y_4)$. This manual selection generates a world plane defined in a coordinates frame in which the plane equation of Π is $Z=0$ and also defining the scale for the 3D results. With these four correspondences between the world plane and the image plane, the minimal solution for homography $\mathbf{H}_w^0$ is obtained. Then, the UAV is moved, making changes in X,Y and Z axes, while the helipad is tracking by estimating the frame-to-frame homographies $\mathbf{H}_{i-1}^i$, which is used to obtaing the homographies $\mathbf{H}_0^i$, and $\mathbf{H}_w^i$ from which $\mathbf{R}_w^i$ and $\mathbf{t}_w^i$ is estimated. The process is successively repeated until either, the helipad is lost or the user finishes the process. The 3D poses estimation process is done with an average of 12 frame per second FPS, which is a enough for a hight level visual controller using the configuration explained on section IV.

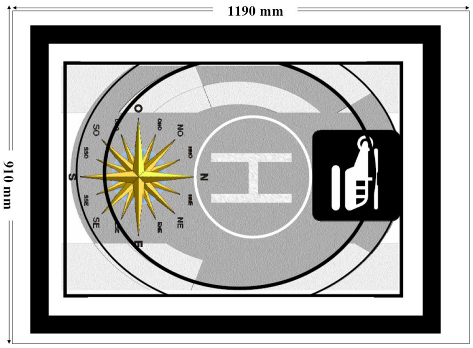


Fig. 4. Helipad used as a plane reference for UAV 3D pose estimation based on homographies.

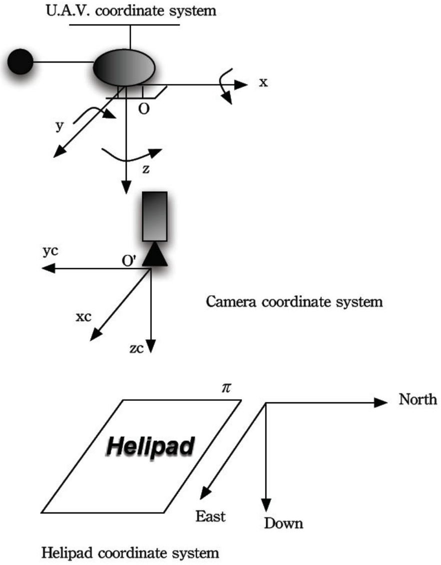
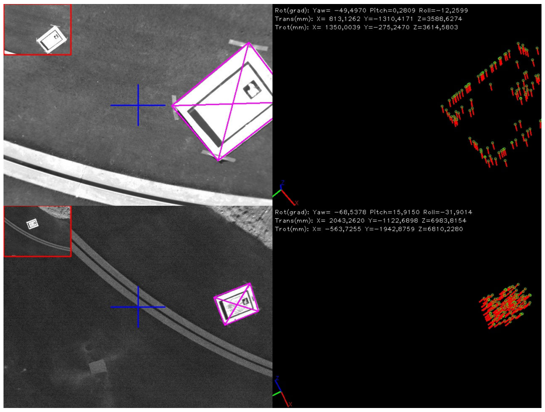


Fig. 5. Helipad, camera and U.A.V coordinate systems

Figure 6 shows two different 3D pose estimation tests, based on a reference helipad, in whose the helicopter is positioned at two different flight levels, the first one is a hovering beginning at *4.2m*, the second one, the test begins with a height of *10m*. This figure also shows the original reference image, the current frame, the optical flow between last and current frame, the helipad coordinates in the current frame camera coordinate system and the Tait-Bryan angles obtained from the rotation matrix. Figure 7 shows the reconstruction of the flight test 1, using the IMU data.



```
Rot(grad): Yaw= -49,4970 Pitch=0,2809 Roll=-12,2599
Trans(mm): X= 813,1262 Y=-1310,4171 Z=3588,6274
Trot(mm): X= 1350,0039 Y=-275,2470 Z=3614,5803
```

```
Rot(grad): Yaw= -68,5378 Pitch=15,9150 Roll=-31,9014
Trans(mm): X= 2043,2620 Y=-1122,6898 Z=6983,8154
Trot(mm): X= -563,7255 Y=-1942,8759 Z=6810,2280
```

Fig. 6. Two different test for 3D pose estimation based on a helipad tracking using Robust Homography estimation. Up: Flight test beginning at an altitude of *4.2m*. Down: Flight test beginning at an altitude of *10m*. In both images, the reference image I_0 is on the small rectangle on the upper left corner. Left it the current frame and Right the Optical Flow between the actual and last frame. Superimposed are the projection of the original rectangle, the translation vector and the Tait-Bryan angles.

The 3D pose estimated using the visual system is compared with helicopter position estimated by the Kalman Filter of the controller, with reference to the takeoff point (Center

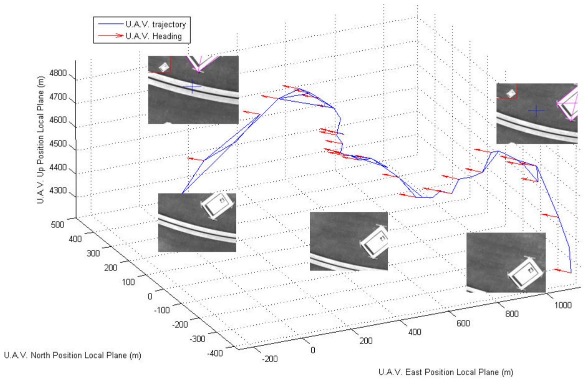


Fig. 7. 3D flight and heading reconstruction. Superimposed images show the helipad at different times of the test, from which the 3D position is estimated.

of the Helipad). Because the local tangent plane to the helicopter is defined in such a way that the X axis is the North position, the Y axis is the East position and Z axis is the Down Position (negative), the measured X and Y values must be rotated according with the helicopter heading or Yaw angle, in order to be comparable with the estimated values obtaining from the homographies. Figures 8, 9 and 10 shows the landmark position with respect to the UAV and figure 11, shows the estimated *yaw* angle.

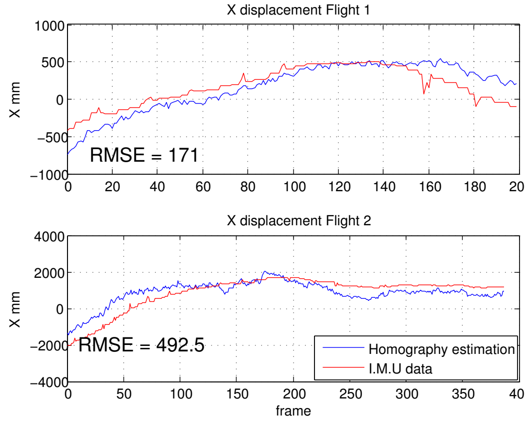


Fig. 8. Comparison between the X axis displacement for homography estimation and IMU data.

Results show a good performance of the visual estimated values compared with the IMU data. In general, estimated and IMU data have the same behavior for both test sequences. For X and Y there is an small error between the IMU and the estimated position, giving a maximum root mean squared error RMSE of *0.42m* in X axis and *0.16m* in Y axis. The estimated altitude position Z have a small error for flight 1 with a RMSE of *0.16m* and *0.85m* in test 2. Although the results are good for height estimation, is important to remember that the IMU altitude estimation have an accuracy of $\pm 0.5m$, causing that the reference altitude estimation used to validate our approach have a big uncertainty. Finally, the yaw angle is correctly estimated, presenting for the first flight and error of 2^o between the IMU and the estimated data, and 4^o for the second tests.

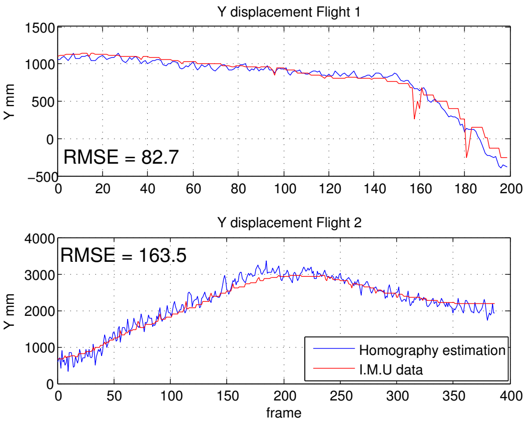


Fig. 9. Comparison between the Y axis displacement for homography estimation and IMU data.

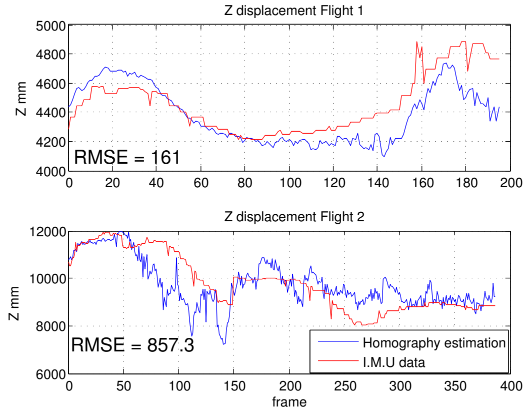


Fig. 10. Comparison between the Z axis displacement for homography estimation and IMU data.

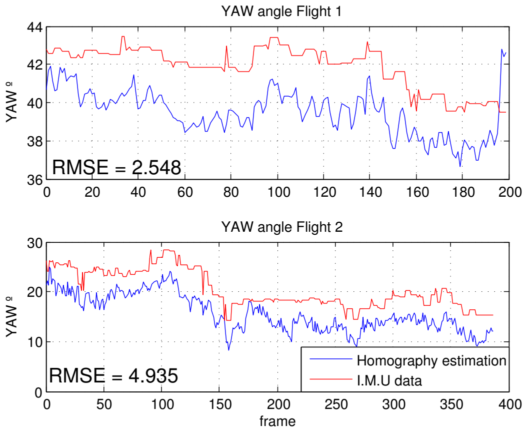


Fig. 11. Comparison between the *Yaw* angle measured using homography estimation and IMU data.

Results also have shown that the system correctly estimate the 3D position when a maximum of the 70 % of the landmark is partially occluded or out of the camera field of view. The accompanying video for this paper, shows the video sequences of the test explained in this section. The accompanying video (high quality) and additional test are also available at the Colibri project web page [20].

The proposed algorithm can be easily adapted for situations in which the ground is not totally flat or the onboard camera is not totally aligned with UAV frame. In this cases, an additional rotation matrix that aligns the camera coordinate system or include the ground rotation is necessary in order to generate the control signal based on the estimated data.

VI. CONCLUSIONS AND FUTURE WORK

This paper has presented a robust real time 3D pose estimation system for UAVs based on piecewise planar object tracking using homographies. The method was tested on real UAV flights, and results have shown that the estimated data is comparable in precision and quality with the one obtained by the IMU of the onboard controller. This indicates that the visual system can be implemented as part of a UAV flight controller for tasks such as autonomous landing or low altitude positioning, where the GPS signal is often inaccurate or unavailable, as well as for use in urban scenarios where piecewise reference marks are easily obtained.

Result have shown that the 3D pose estimated at a frame rate of 12 FPS by the visual system is consistent with the position calculated by the onboard controller. Test have been done at different altitudes, and the estimated values have been compared with the IMU values as ground truth data, producing a small RMSE error for all axes and for the yaw angle. This demonstrates the quality of our pose estimation and its viability as a high level controller in a *dynamic look-and-move* servoing architecture, as is proposed in this paper.

We also have tested the quality of the object tracking system by using a robust frame-to-frame homography estimator. The object can be correctly tracked and its 3D position obtained with high precision, when at least 30 % of the reference object is not occluded or out of the camera field of view as video sequences in the results shows.

Future work includes closing the high level control loop for an autonomous landing on the reference helipad. To achieve this purpose, the 3D pose will be used to generate the references for the low level controller. In addition, we are currently testing improved versions of the Lucas Kanade optical flow, like the Inverse compositional algorithm (ICA) as well as evaluating the use of a Kalman Filter for improved the 3D pose estimation.

VII. ACKNOWLEDGMENTS

The work reported in this paper is the consecution 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 consecution, 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's PhD Scholarships. This work has been sponsored by the Spanish Science and Technology Ministry under the grant CICYT DPI 2007-66156

REFERENCES

[1] L. Mejias, S. Saripalli, P. Campoy, and G. Sukhatme, "Visual servoing approach for tracking features in urban areas using an autonomous helicopter," in *Proceedings of IEEE International Conference on Robotics and Automation*, Orlando, Florida, May 2006, pp. 2503–2508.

[2] S. Saripalli and G. S. Sukhatme, "Landing a helicopter on a moving target," in *Proceedings of IEEE International Conference on Robotics and Automation*, Rome, Italy, April 2007, pp. 2030–2035.

[3] T. Merz, S. Duranti, and G. Conte, "Autonomous landing of an unmanned helicopter based on vision and inbertial sensing." in *International Symposium on Experimental Robotics*, Singapore, June 2004.

[4] P. Campoy, J. F. Correa, I. Mondragón, C. Martínez, M. Olivares, L. Mejías, and J. Artieda, "Computer vision onboard UAVs for civilian tasks," *Journal of Intelligent and Robotic Systems.*, vol. 54, no. 1-3, pp. 105–135, 2009.

[5] Z. He, L. R. V, and C. P. R, "Vision-based UAV flight control and obstacle avoidance," in *Proceedings of the American Control Conference*, June 2006, p. 5pp.

[6] R. Carnie, R. Walker, and P. Corke, "Image processing algorithms for UAV "sense and avoid"," in *Robotics and Automation, 2006. ICRA 2006. Proceedings 2006 IEEE International Conference on*, May 2006, pp. 2848–2853.

[7] C. De Wagter and J. Mulder, "Towards vision-based uav situation awareness," *AIAA Guidance, Navigation, and Control Conference and Exhibit*, August 2005.

[8] S. Saripalli, J. F. Montgomery, and G. S. Sukhatme, "Visually-guided landing of an unmanned aerial vehicle," *IEEE Transactions on Robotics and Automation*, vol. 19, no. 3, pp. 371–381, June 2003.

[9] S. Hrabar and G. Sukhatme, "Omnidirectional vision for an autonomous helicopter," in *In IEEE Internation Conference on Robotics and Automation*, 2003, pp. 558–563.

[10] S. Lange, N. Sünderhauf, and P. Protzel, "Autonomous landing for a multirotor UAV using vision," in *In Workshop Proceedings of SIMPAR Intl. Conf. on SIMULATION, MODELING and PROGRAMMING for AUTONOMOUS ROBOTS*, Venice, Italy, Nov 2008, pp. 482–491.

[11] G. Simon, A. Fitzgibbon, and A. Zisserman, "Markerless tracking using planar structures in the scene," in *Augmented Reality, 2000. (ISAR 2000). Proceedings. IEEE and ACM International Symposium on*, 2000, pp. 120–128.

[12] G. Simon and M.-O. Berger, "Pose estimation for planar structures," *Computer Graphics and Applications, IEEE*, vol. 22, no. 6, pp. 46–53, Nov/Dec 2002.

[13] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 22, no. 11, pp. 1330–1334, 2000.

[14] P. Sturm, "Algorithms for plane-based pose estimation," pp. 1010–1017, June 2000. [Online]. Available: http://perception.inrialpes.fr/Publications/2000/Stu00b

[15] Bouguet Jean Yves, "Pyramidal implementation of the lucas-kanade feature tracker," Intel Corporation. Microprocessor Research Labs, Santa Clara, CA 95052, Tech. Rep., 1999.

[16] J. Shi and C. Tomasi, "Good features to track," in *1994 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'94)*, 1994, pp. 593–600.

[17] M. A. Fischer and R. C. Bolles, "Random sample concensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[18] A. Criminisi, I. D. Reid, and A. Zisserman, "A plane measuring device," *Image Vision Comput.*, vol. 17, no. 8, pp. 625–634, 1999.

[19] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, ISBN: 0521540518, 2004.

[20] COLIBRI, "Universidad Politécnica de Madrid. Computer Vision Group. COLIBRI Project," http://www.disam.upm.es/colibri, 2009.

[21] S. Hutchinson, G. D. Hager, and P. Corke, "A tutorial on visual servo control," in *IEEE Transaction on Robotics and Automation*, vol. 12(5), October 1996, pp. 651–670.

[22] F. Chaumette and S. Hutchinson, "Visual servo control. i. basic approaches," *Robotics & Automation Magazine, IEEE*, vol. 13, no. 4, pp. 82–90, 2006. [Online]. Available: http://dx.doi.org/10.1109/MRA.2006.250573

[23] B. Siciliano and O. Khatib, Eds., *Springer Handbook of Robotics*. Berlin, Heidelberg: Springer, 2008.