REAL-TIME VISUAL SERVOING CONTROL OF A FOUR-ROTOR ROTORCRAFT

Romero Hugo [*,1] **Salazar Sergio** [**]
Lozano Rogelio [*,2] **Benosman Ryad** [***]

[*] *HEUDIASyC Centre de Recherches Royallieu, 60205 Compiegne France. hromero, rlozano@hds.utc.fr*
[**] *Instituto de Investigaciones Eléctricas Reforma 113, 62490 Cuernavaca , Mor. Mexico. ssalazar@iie.org.mx*
[***] *LISIF, 3 rue Galile Site "Saint Raphael", 94200 Ivry sur Seine, France. benosman@ccr.jussieu.fr*

Abstract:
In this paper we address the problem of stabilization and local positioning of a four-rotor rotorcraft using computer vision. Our approach combines the measurements from an Inertial Measurement Unit and a vision system composed of a simple camera used to estimate the orientation and position of the rotorcraft. The vision system provides the position and yaw angle while the IMU gives the pitch and roll angles at a higher rate. We present two different techniques to obtain the position from the image. We present real-time experiments of the stabilization of a four-rotor rotorcraft.

Keywords: Aircraft Control, Computer Vision, Robot Navigation.

1. INTRODUCTION

Unmanned Aerial Vehicles (UAV's) have recently attracted a lot of attention from the automatic control community because of their wide range of civil as well as military applications. Autonomous flying machines can be used in the surveillance of hostile sites or polluted environments (nuclear power stations, chemical plants, etc.,) or simply for monitoring forests, highway traffic. All these applications require UAV's capable of hovering and navigation which heavily depend on the applied control strategy and the on-board set of sensors which should provide information on the behavior of the UAV at an affordable price.

We have chosen the aerodynamical configuration of a four-rotor rotorcraft in view of its of versatility and maneuverability. This rotorcraft is able to carry out many different types of tasks including hovering as well as forward flight. We built an experimental platform based on the four-rotor rotorcraft to which we have added an Inertial Measurement Unit (IMU) and a webcam. It is clear that the performance of a control law heavily depends on the precision of the sensors that are used to estimate the orientation and position of the autonomous vehicle. The standard sensors for measuring orientation in UAV's are IMU, composed of gyros, accelerometers and magnetometers. The position outdoors is usually given by GPS (Global Positioning System) and laser radars and ultrasonic sensors are used to avoid obstacles. Nevertheless, these sensors are unfortunately heavy and expensive and they do not necessarily have the required accuracy. Depending on the application, we sometimes need a global position estimation to be able to follow a trajectory in long distance flights, while in other situations we require a high precision position measurement for hovering above a pre-defined geographical site. We are interested in the latter case where computer vision can be used to estimate the position of the aircraft with respect to a well known target in the environment. In our case the target is composed of 4 circles of different colors (red, blue, green and black) centered in the vertices of a square on the ceiling of the laboratory room.

Recent works as (Nordberg, 2002) (Saripalli, 2003) (A. Wu *et al.*, 2005) show that the computer vision has many advantages with respect to other sensors. A webcam is an unexpensive device that provides a large amount of information. Furthermore a webcam is light and has low energy consumption. Vision could be used in UAV's for navigating indoor and in urban environments.

This paper presents a visual servoing technique for a four-rotor rotorcraft. The approach is based on a camera calibration method which can be either the *two planes* approach or the *homogeneous transformation* approach. The position is estimated by using the *perspective of n-points* method or the *plane based pose* method respectively. The pitch and roll angles measurements are ob-

[1] Partially supported by SEP-PROMEP and UAEH, Mexico.
[2] Corresponding author

tained from the IMU while the $x-y$ position and the yaw angle are provided by the vision system. We illustrate the performance of the proposed vision-based control scheme in simulations and in a real-time experiment.

This paper is organized as follows: Section II is devoted to the camera calibration methods. The four-rotor rotorcraft model is described in Section III. The control strategy is given in Section IV. Section V presents the position estimation algorithms. The real-time experiments are shown in Section VI while concluding remarks are finally given in Section VII.

2. CAMERA CALIBRATION

Camera calibration is the process of determining the optical and internal camera geometric characteristics (intrinsic parameters) and the position and orientation of the camera with respect to a certain world coordinate system (extrinsic parameters) (Hartley *et al.*, 2004) (Corke, 1996) (K. Gremban *et al.*, 1988). Two different techniques of camera calibration are described below.

Tow-Planes Calibration Approach

This method initially described in (H. A. Martins, 1981) gives a solution for the back-projection problem. In (K. Gremban *et al.*, 1988) this result is extended given a solution of the projection problem and we can see an application in (J. Fabrizio *et al.*, 2002). In addition, this method is efficient since it requires only a matrix multiplication and some matrix inversions. This method is very straightforward and easy to implement. The planes for calibration are defined as P_1 and P_2.

Let $g_{ij} = \begin{pmatrix} x & y & z \end{pmatrix}^T$ be a point in real world coordinates and $q_{ij} = \begin{pmatrix} \rho & \gamma & 1 \end{pmatrix}^T$ a point in the image plane with homogeneous coordinates in pixels. The linear transformation between these two points is defined as

$$g_{ij} = H_i q_{ij} \tag{1}$$

where $H_i \in \mathbb{R}^{3\text{x}3}$. For n points in our calibration target, we have consequently a system of linear equations

$$G_i = H_i Q_i \tag{2}$$

where $g_{ij} \in G_i \in \mathbb{R}^{3\text{x}n}$, $q_{ij} \in Q_i \in \mathbb{R}^{3\text{x}n}$. The solution for this matrix equation is obtained applying the generalized matrix inverse, given by

$$H_i = [Q_i^T Q_i]^{-1} Q_i^T P_i = Q_i^+ G_i \tag{3}$$

where H_i is the "Homography" or matrix of transformation between the image plane and the real world. For each calibration plane and its projection in the image plane it is necessary to calculate one homography, defined by H_1 and H_2 for P_1 and P_2 respectively. In order to compute the focal point of the camera, we take all points q_j in the image plane and back-project them onto each plane P_1 and P_2 as the points $g_{ij} = H_i * q_j$ with $i = 1, 2$ and $j = 1, 2, \ldots, n$. Each couple (g_{1j}, g_{2j}) defines a vector $l_j = g_{2j} - g_{1j}$. Assuming no distortion in the lens of the camera each of those vectors converge to the focal point $F = \begin{pmatrix} x_f & y_f & z_f \end{pmatrix}^T$. However, some distortion may exist in the camera, and, in that case the location of the focal point of the camera is computed using a least square algorithm to find a **3D** point that minimizes the distance to all vectors.

Finally, we compute the camera's principal point (piercing point) (ρ_p, γ_p) and the orientation of the image plane following (K. Gremban *et al.*, 1988). Let us denote $\boldsymbol{V}_R = \begin{pmatrix} r_x & r_y & r_z \end{pmatrix}^T$ and $\boldsymbol{V}_C = \begin{pmatrix} c_x & c_y & c_z \end{pmatrix}^T$ as the vectors defining the image plane orientation and $V_i = \begin{pmatrix} v_{xi} & v_{yi} & v_{zi} \end{pmatrix}^T$ as the vector from the focal point F to any point $\overline{g_i}$ in the real world, *i.e.*

$$v_i = \frac{\overline{G_i} - F}{\|\overline{G_i} - F\|} \tag{4}$$

where $\overline{G_i}$ is the vector from the origin of the coordinate system to point $\overline{g_i}$.

Projecting vector V_i on the image plane defined by $\boldsymbol{V}_R$ and $\boldsymbol{V}_C$ we obtain the coordinates (ρ_i, γ_i) of point i in the image plane as follows

$$\begin{bmatrix} \rho_1 & \gamma_1 \\ \rho_2 & \gamma_2 \\ \vdots & \vdots \\ \rho_n & \gamma_n \end{bmatrix} = \begin{bmatrix} v_{x1} & v_{y1} & v_{z1} & 1 \\ v_{x2} & v_{y2} & v_{z2} & 1 \\ \vdots & \vdots & \vdots & \vdots \\ v_{xn} & v_{yn} & v_{zn} & 1 \end{bmatrix} \begin{bmatrix} r_x & c_x \\ r_y & c_y \\ r_z & c_z \\ \rho_p & \gamma_p \end{bmatrix} \tag{5}$$

which can be rewritten using obvious notation as

$$B = W \begin{bmatrix} \boldsymbol{V}_R & \boldsymbol{V}_C \\ \rho_p & \gamma_p \end{bmatrix} = WX \tag{6}$$

From the above equation we can compute the camera parameters $\boldsymbol{V}_R, \boldsymbol{V}_C, \rho_p$ and γ_p which together with the focal point F and the homographies H_1 and H_2 we have a complete characterization of the camera.

Homogeneous Transformation Approach

In this section we present an alternative approach for computing the parameters of a camera based on singular value decomposition. Consider n points in the real world defined as $\boldsymbol{d}_i = (x_i \; y_i \; z_i \; 1)^T$and their corresponding image point $\boldsymbol{e}_i = (\rho_i \; \gamma_i \; 1)^T$ for $i = 1, \ldots, n$, both in homogeneous coordinates. Then there exists a transformation $\boldsymbol{T} \in \mathbb{R}^{3\times4}$

$$\boldsymbol{T} = \begin{bmatrix} \boldsymbol{T}_1 & \boldsymbol{T}_2 & \boldsymbol{T}_3 \end{bmatrix}^T \tag{7}$$

such that

$$\boldsymbol{E} = \boldsymbol{T}\boldsymbol{D} \tag{8}$$

where $\boldsymbol{D} = [\boldsymbol{d}_1 \; \boldsymbol{d}_2 \; \cdots \; \boldsymbol{d}_n] \in \mathbb{R}^{4\times n}$ and $\boldsymbol{E} = [\boldsymbol{e}_1 \; \boldsymbol{e}_2 \; \cdots \; \boldsymbol{e}_n] \in \mathbb{R}^{3\times n}$. The matrix transformation $\boldsymbol{T}$ can be obtained by solving the matrix equation (9) by singular value decomposition (SVD) (see (Hartley *et al.*, 2004)(Zhang, 2002))

$$\begin{bmatrix} \mathbf{0}^* & -d_1^T & \gamma_i * d_1^T \\ d_1^T & \mathbf{0}^* & \rho_i * d_1^T \\ & \vdots & \\ \mathbf{0}^* & -d_n^T & \gamma_n * d_n^T \\ d_n^T & \mathbf{0}^* & \rho_n * d_n^T \end{bmatrix} \begin{bmatrix} \boldsymbol{T}_1^T \\ \boldsymbol{T}_2^T \\ \boldsymbol{T}_3^T \end{bmatrix} = \mathbf{0} \tag{9}$$

where $\mathbf{0}^*, \boldsymbol{T}_i \in \mathbb{R}^{1\text{x}4}$. The $\boldsymbol{T}$ matrix can be rewritten as

$$\boldsymbol{T} = [\, \boldsymbol{M} \,|\, -\boldsymbol{M}\tilde{C} \,] = \boldsymbol{K}[\, \tilde{\boldsymbol{R}} \,|\, -\tilde{\boldsymbol{R}}\tilde{C} \,] \tag{10}$$

where $\boldsymbol{M} \in \mathbb{R}^{3\text{x}3}$ and $\boldsymbol{M}\widetilde{C} \in \mathbb{R}^{3\text{x}1}$ are block matrices of $\boldsymbol{T}$. We can easily find $\boldsymbol{K}$ and $\widetilde{\boldsymbol{R}}$ matrices using

RQ decomposition over $\boldsymbol{M}$. $\boldsymbol{K} \in \mathbb{R}^{3x3}$ is an upper-triangular matrix representing the intrinsic parameters of the camera and $\widetilde{\boldsymbol{R}} \in \mathbb{R}^{3x3}$ is an orthogonal matrix with unitary norm per column, that gives the orientation of the camera. Finally the vector $\widetilde{C}$ gives the center of the camera. Further details of camera parameters can be found in (Hartley *et al.*, 2004).

3. FOUR-ROTOR ROTORCRAFT MODEL

In this work we use the commercial Draganfly IV four-rotor (Inc., n.d.) where the parameters are $m = 0.5kg$, $l = 0.24m$, like the one shown in figure 1. This kind of configuration has many advantages over conventional helicopters and other configurations. This configuration does not have a swashplate and it does not need any blade pitch control. Furthermore the gyroscopic effects and aerodynamic torques tend to cancel in trimmed flight, because the front and rear motors rotate counterclockwise while the right and left motors rotate clockwise.

The collective input (or throttle input) is the sum of the thrusts of each motor. Pitch movement is obtained by increasing (reducing) the speed of the rear motor while reducing (increasing) the speed of the front motor. The roll movement is obtained similarly using the lateral motors. The yaw movement is obtained by increasing (decreasing) the speed of the front and rear motors while decreasing (increasing) the speed of the lateral motors. This should be done while keeping the total thrust constant.

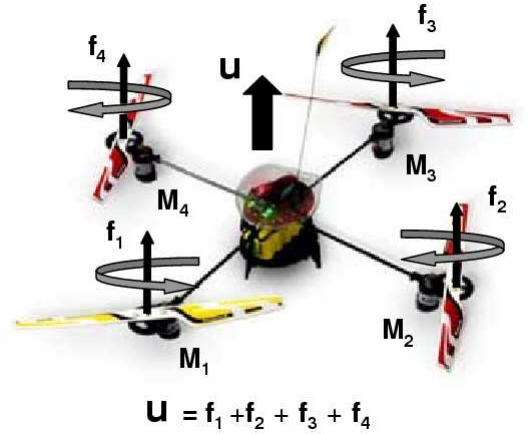

Fig. 1. Four-Rotor Rotorcraft Scheme

The dynamic model of the full rotorcraft used, as proposed in (Castillo *et al.*, 2005), is obtained from Euler-Lagrange equations with external generalized forces.

$$m\ddot{x} = -u\sin\theta, \tag{11}$$
$$m\ddot{y} = u\cos\theta\sin\phi, \tag{12}$$
$$m\ddot{z} = u\cos\theta\cos\phi - mg, \tag{13}$$
$$\ddot{\psi} = \tilde{\tau}_\psi, \tag{14}$$
$$\ddot{\theta} = \tilde{\tau}_\theta, \tag{15}$$
$$\ddot{\phi} = \tilde{\tau}_\phi, \tag{16}$$

where x and y are the position coordinates in the horizontal plane, z is the vertical position, ψ is the yaw angle around the z-axis, θ is the pitch angle around the modified y-axis, and ϕ is the roll angle around the modified x-axis, m denotes the mass of the quad-rotor, g is the acceleration due to gravity, and $\tilde{\tau}_\psi$, $\tilde{\tau}_\theta$, and $\tilde{\tau}_\phi$ are the yawing moment, pitching moment, and rolling moment, respectively, which are related to the generalized torques τ_ψ, τ_θ, τ_ϕ, see (Castillo *et al.*, 2005).

4. CONTROL STRATEGY

The control strategy used here follows the controller synthesis approach developed in (Castillo *et al.*, 2005). We have chosen this controller because the control algorithm is simple to implement and easy to tune. In addition, this control strategy uses a saturation functions in order to privilege the regulation of roll and pitch angles over the positions x and y respectively. The experimental setup is such that the four control inputs can independently operate in either manual or automatic modes.

Control of altitude and yaw

The vertical displacement z in (13) is controlled by forcing the altitude to satisfy the dynamics of a linear system. Thus, we set

$$u = (r_1 + mg)\frac{1}{\cos\theta\cos\phi}, \tag{17}$$

where r_1 is given by the PD controller

$$r_1 \triangleq -a_{z_1}\dot{z} - a_{z_2}(z - z_d), \tag{18}$$

where a_{z_1}, a_{z_2} are positive constants and z_d is a positive constant representing the desired altitude. To control yaw angle we set

$$\tilde{\tau}_\psi = -a_{\psi_1}\dot{\psi} - a_{\psi_2}(\psi - \psi_d). \tag{19}$$

Assuming $\cos\theta\cos\phi \neq 0$, that is, $\theta, \phi \in (-\pi/2, \pi/2)$ and from (13), (14), (17), (18) and (19) it follows that, if ψ_d and z_d are constants, then ψ and z converge. Therefore, $\dot{\psi}$ and $\ddot{\psi} \to 0$, which, using (14) and (19), implies that $\psi \to \psi_d$. Similarly, $z \to z_d$.

Control of lateral position and roll

We assume $\psi_d \equiv 0$ in (19). Therefore, from (14) and (19) it follows that $\psi \to 0$. Note that (13), (17), and (18) imply that $r_1 \to 0$.

The amplitudes of the saturation functions can be chosen in such a way that, after a finite time $\hat{t}$, the roll angle lies in the interval -1 rad $\leq \phi \leq 1$ rad. Therefore, for $t > \hat{t}$ $|\tan\phi - \phi| < 0.54$. Thus, after sufficient time, r_1 is small and the (y, ϕ) subsystem reduces to

$$\ddot{y} = g\phi, \tag{20}$$
$$\ddot{\phi} = \tilde{\tau}_\phi, \tag{21}$$

which represents four integrators in cascade.

For (20)-(21) the nested saturation controller has the form

$$\tilde{\tau}_\phi = -\sigma_{\phi_1}(\dot{\phi} + \sigma_{\phi_2}(\phi + \dot{\phi} + \sigma_{\phi_3}(2\phi + \dot{\phi} + \frac{\dot{y}}{g} + \sigma_{\phi_4}(\dot{\phi} + 3\phi + 3\frac{\dot{y}}{g} + \frac{y}{g})))), \tag{22}$$

where σ_a is a saturation function of the form

$$\sigma_a(s) = \begin{cases} -a & s < -a, \\ s & -a \leq s \leq a, \\ a & s > a. \end{cases}$$

The closed-loop is asymptotically stable, see (Castillo *et al.*, 2005), and therefore ϕ, $\dot{\phi}$, y and $\dot{y}$ converge to zero.

Control of forward position and pitch

Finally, we control the (x, θ) subsystem given by

$$\ddot{x} = -g \tan\theta,$$
$$\ddot{\theta} = \tilde{\tau}_\theta.$$

Using a procedure similar to the one proposed for the roll control, we obtain

$$\tilde{\tau}_\theta = -\sigma_{\theta_1}(\dot{\theta} + \sigma_{\theta_2}(\theta + \dot{\theta} + \sigma_{\theta_3}(2\theta + \dot{\theta} - \frac{\dot{x}}{g} + \sigma_{\theta_4}(\dot{\theta} + 3\theta - 3\frac{\dot{x}}{g} - \frac{x}{g})))), \quad (23)$$

and thus θ, $\dot{\theta}$, x and $\dot{x}$ also converge to zero.

5. POSITION AND ORIENTATION ESTIMATION APPROACHES

After applying the two calibration to obtain the intrinsic and extrinsic parameters of the camera, we can proceed to estimate the position and orientation of the camera with respect to a target composed of four circles. Each one of these circles has a different color and is posed in the vertices of a square whose geometry is well known. To compute the center of gravity of each circle in the scene we also apply a color calibration which provides robustness with respect to possible changes in the luminosity of the scene. In the following we will develop the position and orientation estimation algorithms for each one of the calibration approaches.

5.1 Perspective N-Points Method

In this section we present a technique to estimate position and orientation based on the two planes camera calibration approach presented previously. This approach consists of the determination of a distance between the camera and a set of well known points in an object coordinate space (Fabrizio *et al.*, 2004). Every point in the real world can be expressed in the general coordinate system as

$$g_i = F + \lambda_i v_i \quad (24)$$

where F is the focal point, v_i is a unitary vector defined in (4) and $\lambda_i > 0$ is a scale factor. In order to simplify we set the origin of our coordinate system at the focal point, *i.e.* $F = 0$, so that (24) becomes

$$g_i = \lambda_i v_i \quad (25)$$

Furthermore the vector that joints points g_j and g_i is given by

$$\boldsymbol{g}^{i,j} = g_j - g_i = \lambda_j v_j - \lambda_i v_i \quad (26)$$

Let us assign a number to each one of the vertices of the square target formed by the four circles. We assign number 1 to the vertex in the upper left corner and label the others clockwise. With this assignment we have $\boldsymbol{g}^{1,2} + \boldsymbol{g}^{3,4} = 0$, which can be expressed as a linear equation system

$$\begin{bmatrix} v_{x2} & -v_{x3} & v_{x4} \\ v_{y2} & -v_{y3} & v_{y4} \\ v_{z2} & v_{z3} & v_{z4} \end{bmatrix} \begin{pmatrix} \lambda_2 \\ \lambda_3 \\ \lambda_4 \end{pmatrix} = \lambda_1 \begin{pmatrix} v_{x1} \\ v_{y1} \\ v_{z1} \end{pmatrix} \quad (27)$$

Solving the above equation system by Cramer's rule we obtain

$$\lambda_2 = (\frac{\Delta_{134}}{\Delta_{234}})\lambda_1 \quad (28)$$

$$\lambda_3 = (\frac{\Delta_{124}}{\Delta_{234}})\lambda_1 \quad (29)$$

$$\lambda_4 = (\frac{\Delta_{123}}{\Delta_{234}})\lambda_1 \quad (30)$$

where Δ_{ijk} are the determinants associated with Cramer's rule solution. Note that λ_i depends on λ_1 with $i = 2, 3, 4$. Now, defining the distance (see 26)

$$L_{1,4} = \left\|\boldsymbol{g}^{1,4}\right\| = \|\lambda_4 v_4 - \lambda_1 v_1\|$$

Combining (30) and the above we have

$$\lambda_1 = L_{1,4} / \sqrt{1 - 2v_1 v_4 (\Delta_{123}/\Delta_{234}) + (\Delta_{123}^2/\Delta_{234}^2)} \quad (31)$$

Substituting the value of λ_1 above into (28)-(30) we obtain the values of λ_2, λ_3 and λ_4. Therefore, the location of the four circles constituting the target with respect to an inertial frame attached to the plane containing the four circles are given by $F + \lambda_1 v_1, F + \lambda_2 v_2, F + \lambda_3 v_3$ and $F + \lambda_4 v_4$. The location x, y and z of the focal point of the camera can be obtained as $g_i - \lambda_i v_i$ where g_i is the position of the *ith* circle for $i = 1, 2, 3, 4$. The orientation of the camera is obtained from the knowledge of the main vector of the camera going from the focal point to the piercing point and the vectors $\lambda_i v_i$ for $i = 1, 2, 3, 4$.

5.2 Plane-Based Pose Method

We now describe the coordinate transformations that lead from **2D** points on a plane P to the coordinates of their projections in the image plane (Sturm, 2000). Let $\boldsymbol{d}_k$ be the *kth* point on the P plane having coordinates (x_k, y_k, z_k). Let the position and orientation of the plane be given by a rotation matrix $\boldsymbol{S}$ and the translation vector $\bar{\boldsymbol{v}}$ with respect to a global **3D** world reference frame. In this reference frame $\boldsymbol{d}_k$ is expressed as

$$\boldsymbol{d}_k^w = \begin{bmatrix} \boldsymbol{S}_{3x3} & \bar{\boldsymbol{v}}_{3x1} \\ \boldsymbol{0}_{1x3} & 1 \end{bmatrix} \begin{bmatrix} x_k \\ y_k \\ 0 \\ 1 \end{bmatrix} \quad (32)$$

with $z_k = 0$. Let the position of the camera be given by the rotation matrix $\boldsymbol{R}$ and the translation vector t. Therefore the coordinates of $\boldsymbol{d}_k$ in the local camera frame are

$$\boldsymbol{d}_k^c = \begin{bmatrix} \boldsymbol{R}_{3x3} & \boldsymbol{t}_{3x1} \\ \boldsymbol{0}_{1x3} & 1 \end{bmatrix} \boldsymbol{d}_k^w \quad (33)$$

Combining (32), (33) and (10) the coordinates of the projected point are

$$\boldsymbol{e}_k \sim \begin{bmatrix} \boldsymbol{K} & \boldsymbol{0} \end{bmatrix} \boldsymbol{d}_k^c$$
$$\sim \boldsymbol{K} \begin{bmatrix} \boldsymbol{RS} & \boldsymbol{R\bar{v}} + \boldsymbol{t} \end{bmatrix} \begin{pmatrix} x_k & y_k & 0 & 1 \end{pmatrix}^T$$

The homography $\boldsymbol{T}$ is defined as

$$\boldsymbol{T} \sim \boldsymbol{K} \begin{bmatrix} (\boldsymbol{R\bar{S}})_{3x2} & (\boldsymbol{R\bar{v}} + \boldsymbol{t})_{3x1} \end{bmatrix}$$

where $\bar{\boldsymbol{S}}$ is the 3x2 submatrix of $\boldsymbol{S}$ consisting of its first two columns. Since the calibration is known we can compute

$$\boldsymbol{A} \sim \boldsymbol{K}^{-1} \boldsymbol{T} = \begin{bmatrix} \boldsymbol{R\bar{S}} & \boldsymbol{R\bar{v}} + \boldsymbol{t} \end{bmatrix} \quad (34)$$

Defining

$$\boldsymbol{N} = \boldsymbol{RS}$$
$$\boldsymbol{w} = \boldsymbol{S^T}\bar{\boldsymbol{v}} + \boldsymbol{S^T R^T t}$$

we can rewrite (34) as

$$\boldsymbol{A} \sim \boldsymbol{NW_1} = \boldsymbol{N}\begin{bmatrix} \boldsymbol{I}_{2x2} & \boldsymbol{w} \\ \boldsymbol{0} & \end{bmatrix}$$

In the general case, $\boldsymbol{A}$ is obtained from the following minimization problem

$$\min_{\boldsymbol{N},\boldsymbol{w},\boldsymbol{\mu}} \|\mu\boldsymbol{A} - \boldsymbol{NW}_1\|_F^2 \text{ , subject to } \boldsymbol{N^T N} = \boldsymbol{I}_3$$

As shown in (Sturm, 2000) the solution for $\boldsymbol{N}$ does not depend on μ and $\boldsymbol{w}$, and thus the optimal solution for $\boldsymbol{N}$ can be defined as

$$\min_{\bar{N}} \left\|\bar{\boldsymbol{A}} - \bar{\boldsymbol{N}}\right\|_F^2 \text{ , subject to } \bar{\boldsymbol{N}}^{\boldsymbol{T}}\bar{\boldsymbol{N}} = \boldsymbol{I}_2 \qquad (35)$$

To find the optimal solution of (35) we compute the SVD of $\bar{\boldsymbol{A}} = \boldsymbol{U}_{3x2}\boldsymbol{S}_{2x2}\boldsymbol{V}_{2x2}$, then $\bar{\boldsymbol{N}}$ is given by

$$\bar{\boldsymbol{N}} = \boldsymbol{UV}$$

The third column of rotation matrix $\boldsymbol{N}$ can be computed from the cross product of the columns of $\bar{\boldsymbol{N}}$. Once $\boldsymbol{N}$ is computed, the optimal scale factor μ and vector $\boldsymbol{w}$ are determined by

$$\mu = \frac{tr(\bar{\boldsymbol{N}}^{\,T}\bar{\boldsymbol{A}}^{\,T})}{tr(\bar{\boldsymbol{A}}^{\,T}\bar{\boldsymbol{A}}^{\,T})}$$
$$\boldsymbol{w} = \boldsymbol{N}^T\boldsymbol{A}\begin{pmatrix} 0 & 0 & \mu \end{pmatrix}^T$$

With all these relations we can set the position and orientation of the camera respect to a plane located in a well known coordinate system.

6. EXPERIMENTAL RESULTS

In this section, we present real-time experiments of the stabilization of the four-rotor rotorcraft using the methods to estimate the position and orientation proposed in the previous sections. We use the *two planes* approach described above and we obtained the following calibration matrices given in (3)

$$\left.\begin{aligned} \boldsymbol{H}_1 &= \begin{bmatrix} 0.0575 & -0.0005 & 0 \\ -0.0007 & -0.0578 & 0 \\ -9.7570 & 7.5165 & 1 \end{bmatrix} \\ \boldsymbol{H}_2 &= \begin{bmatrix} 0.0466 & -0.0003 & 0 \\ -0.0006 & -0.0468 & 0 \\ -7.6024 & 5.9127 & 6 \end{bmatrix} \end{aligned}\right\} \qquad (36)$$

for the planes P_1 and P_2. We computed the camera parameters given by the focal point F, the piercing point (ρ_p, γ_p) and the vectors V_R and V_C and we obtained the following values

$$\left.\begin{aligned} F &= \begin{bmatrix} 1.5928 & -0.7846 & 27.3204 \end{bmatrix}^T \\ (\rho_p, \gamma_p) &= \begin{bmatrix} 83.2578 & 113.7901 \end{bmatrix} \\ V_R &= \begin{bmatrix} 465.0340 & -4.4622 & -86.2788 \end{bmatrix}^T \\ V_C &= \begin{bmatrix} -6.2345 & -465.5784 & -59.6475 \end{bmatrix}^T \end{aligned}\right\} \qquad (37)$$

On the other hand, when using the homogeneous transformation approach for calibrating the camera we obtained the following intrinsic parameters matrix K as in (10)

$$K = \begin{bmatrix} 430 & 2 & 226 \\ 0 & 438 & 161 \\ 0 & 0 & 1 \end{bmatrix} \qquad (38)$$

6.1 Comparison between the two position estimation methods

We compared the two calibration methods described in the previous sections. We took a video sequence by manually deplacing the camera above the target composed of four circles of different colors. We tested both calibration methods on the video sequence using Matlab. The obtained position and orientation are shown in figure 2.

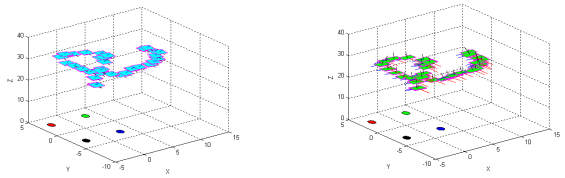

Fig. 2. Comparison between approaches. Left side, position estimation using the perspective n-points method. Right side, position estimation using the plane-based pose technique.

Figure 2 left, was obtained using the *two planes* approach for camera calibration and the *perspective of n points* technique for estimating the position and orientation. Figure 2 right, was obtained using the *homogeneous transformations* approach for camera calibration and the *plane-based pose* technique for estimating the position and orientation. Figure 3 presents a comparison between both methods which give similar results.

	METHOD 2						METHOD 2					
	x	y	z	φ	θ	ϕ	x	y	z	φ	θ	ϕ
1	0.5	-1.1	21.2	10.5	2.4	-6.9	0.6	-1.1	18.5	9.2	1.6	-8.6
2	2.6	-1.0	33.6	8.8	-0.2	-4.7	2.8	-1.1	29.1	7.9	-0.3	-3.3
3	0.7	-5.3	31.6	19.3	-0.3	-3.9	0.9	-5.6	27.9	18.5	-0.1	-4.3
4	8.3	-1.8	33.5	15.9	-6.3	7.5	9.0	-1.8	30.1	16.7	-5.0	6.0

Fig. 3. Comparison of location and attitude estimations obtained from each method

6.2 Real-Time Platform Architecture

For simplicity, for the real-time implementation, we have chosen the *two planes* approach for camera calibration and the *perspective of n points* technique for estimating the position and orientation. Our experimental platform is built using a Draganfly IV four-rotor rotorcraft equipped with an IMU and a webcam. The image captured by the on-board camera is sent to a PC on the ground through a USB connection. The frames of the image are treated over a computer devoted to vision. We work with 10 FPS and the image analysis program is carried out in C++. The images analysis allows the estimation of the position on the (x, y) plane and the yaw angle (ψ) with respect to a well known local

frame. This information is sent by RS232 to the real-time module which is composed of the host computer and the target computer running on Matlab Simulink XPc Target working in multitasking mode with a sample rate of 0.05 sec . The control law computation is done in the host computer. We use the commercial Futaba radio to transmit the control signals to UAV. The radio joystick potentiometers are connected through the data acquisitions cards Advantech PCL-818HG (16 channels A/D) and Advantech PCL-726 (6 channels D/A), to the PC. We have used finite differencing of position ($\dot{q} = \frac{q_t - q_{t-T}}{T}$) to estimate the velocities $(\dot{x}, \dot{y}, \dot{z})$.

We performed real-time stabilization of the four-rotor rotorcraft using computer vision and obtained the results presented in figures 5 through figure 10. The experiment starts at time $t = 40s$ and finishes at time $t = 140s$. The position in the (x, y) plane and the orientation yaw angle (ψ) obtained during the experiment show good performance of the proposed computer-vision based control technique.

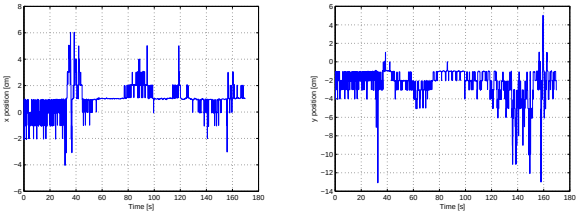

Fig. 4. Position estimation in $x - y$ plane.Left side, x-position of rotorcraft. Right side, y-position of rotorcraft.

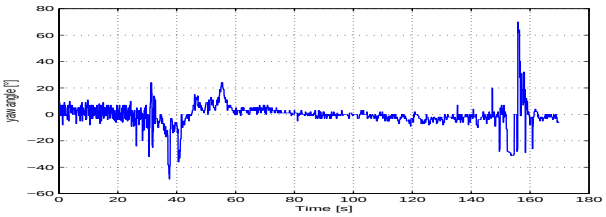

Fig. 5. Yaw angle of the rotorcraft.

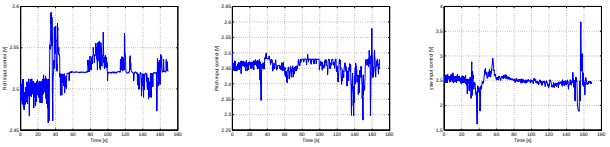

Fig. 6. Control inputs applied to rotorcraft. Left figure shows the roll control input. Center figure presents the yaw control input. Right figure shows the pitch control input.

7. CONCLUSIONS

In this paper we have presented a combination of a vision system with an IMU to compute the position and orientation in order to stabilize at hover a mini four-rotor rotorcraft.

Two different methodologies to estimate the position and orientation were presented and compared. We implemented in real-time the *two planes* approach for camera calibration and the *perspective of n points* technique for estimating the position and orientation. The vision system is composed of an on-board camera and a target given by four circles of different colors. The vision system was used to estimate the position in the $x - y$ plane and the yaw angle ψ. The measurement of the pitch and roll angles was obtained from an IMU at a higher rate as compared to the estimation provided by the vision system. The experimental tests showed that the proposed vision system based control technique performs satisfactorily for stabilizing a mini-aerial vehicle in hover.

REFERENCES

A. Wu, E. Johnson and A. Proctor (2005). Vision-aided inertial navigation for flight control. In: *Proc. of AIAA Guidance, Navigation, and Control Conference and Exhibit*.

Corke, P. (1996). *Visual Control of Robots, High-Performance visual servoing*. 1st. ed. John Wiley and Sons Inc.

J. Fabrizio and J. Devars (2004). The perspective-n-point problem for catadioptric sensors: An analytical approach. In: *Proc.of International Conference on Computer Vision and Graphics*.

H. A. Martins, J. Birk, R. B. Kelley (1981). Camera models based on data from two uncalibrated planes. *Computers Graphics and Image Processing* 17, 173–180.

Hartley, R. and A. Zisserman (2004). *Multiple View Geometry in Computer Vision*. 2nd edition ed.. Cambridge University Press.

Inc., Draganfly Innovations (n.d.). http://www.rctoys.com/.

J. Fabrizio, J. P. Tarel and R. Benosman (2002). Calibration of panoramic catadioptric sensor made easier. In: *Proc. of IEEE Workshop on Omnidirectional Vision Omnivis*. pp. 45–52.

K. Gremban, C. Thorpe and T. Kanade (1988). Geometric camera calibration using systems of linear equations. In: *Proc. of IEEE Conference on Robotics and Automation ICRA*. pp. 562 – 567.

Nordberg, Klas (2002). Vision for a uav helicopter. In: *Proc. of International Conference on Intelligent Robots and Systems*.

P. Castillo, R. Lozano and A. Dzul (2005). *Modelling and Control of Mini-Flying Machines*. 1st. ed.. Springer-Verlag in Advances in Industrial.

Saripalli, S. (2003). Visually-guided landing of an unmaned aerial vahicle. *IEEE Transaction on Robotics and Automation* 19(3), 371–381.

Sturm, P. (2000). Algorithms for plane-based pose estimation. In: *Proc. IEEE Conference on Computer Vision and Pattern Recognition*. pp. 706–711.

Zhang, Z. (2002). A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22, 1330–1334.