Position control of a quad-rotor UAV using vision ⋆

Luis Rodolfo Garcia Carrillo * **Eduardo Rondon** *
Anand Sanchez ** **Alejandro Dzul** *** **Rogelio Lozano** *

* *University of Technology of Compiegne - CNRS UMR 6599 Heudiasyc, BP 20529, 60205 Compiegne Cedex, France (e-mail: name.surname@hds.utc.fr)*
** *Robotics and Advanced Manufacturing Division, Research Center for Advanced Studies-Cinvestav, Saltillo, MEXICO (e-mail: anand.sanchez@cinvestav.edu.mx).*
*** *Laguna Technologic Institute, Torreon, Coahuila, MEXICO (e-mail: dzul@faraday.itlalaguna.edu.mx)*

AbstractWe present a vision system that deals with the problem of local estimation of the position and velocity of a UAV quad-rotor indoor. Our approach is based in the estimation of homographies from a visual landmark. Stabilizing the UAV locally, real time experiments demonstrate the effectiveness of our method.

Keywords: Aerial Robotics, Localization, Computer Vision, Sensor Fusion, Visual Tracking.

1. INTRODUCTION

The development of unmanned aerial vehicles (UAV's) has recently gained attention from different research areas such as electronics, mechanics, aeronautics and automatic control. This great interest is due to the UAV's potential while performing missions where human intervention is impossible, risky or expensive. We can mention the tasks of hazardous material recovery, real-time forest fire monitoring, disaster relief support, surveillance of sensitive areas (borders, ports, oil pipelines), remote sensing and monitoring of traffic, search and rescue operations, etc. UAV's can fly autonomously or semi-autonomously and, in addition, they are expendable or recoverable (1).

The quad-rotor is a highly maneuverable and versatile UAV platform with attractive features: it can perform vertical take-off and landing tasks, hover and longitudinal or lateral flight. An autonomous helicopter must operate without human intervention, yet must meet the rigorous requirements associated with any airborne platform. Nowadays real-time embedded stabilization has been already proved (2), (3). A basic requirement for an UAV is robust autonomous navigation and positioning, which can be carried out by using vision based control techniques. Computer vision is gaining importance in the field of mobile robots. It is currently used in the feedback control loop, as a cheap, passive and information-abundant sensor, usually combined with an Inertial Measurement Unit (IMU) to provide robust relative pose information and allowing autonomous position and navigation. It is important to note that a vision system on board a UAV augments the sensor suit including a global positioning system (GPS) which provides position information relative to an inertial frame but fails indoor or in noisy environments.

Figure 1. The four-rotor rotorcraft experimental platform.

Onboard computer vision systems can be used to estimate a relative position information, which can be obtained, for example, from the detection of artificial landmarks. Once the UAV knows its position, a control strategy could be implemented, so that its position converges to a desired value. Several research works have been performed aiming at the control of UAV's using a camera as a vision sensor (4), (5). In (6), the authors use the projections of parallel lines for the purpose of estimating the location and orientation of a helicopter landing pad. Unfortunately, in this approach the vehicle can not estimate its velocity, which is important for controlling the UAV. A vision algorithm for visual navigation and autonomous landing is presented in (7). An approach based on optical flow techniques has been applied to the real time stabilization of an Eight-Rotor UAV in (8). Systems with more than one camera have also been studied. In (9) the authors present a two cameras

⋆ This work was partially supported by Mexico's National Council of Science and Technology (CONACYT).

system, with the drawback that both cameras are not embedded on the UAV. A stereo vision system combined with a multi-sensor suit is proposed in (10). Stereo systems onboard present the counterpart of the extra payload (two cameras) that the UAV must lift. Another kind of vision sensor is the omnidirectional camera, implemented in (11) over a video sequence captured from an UAV in order to compute its attitude.

In this paper, we present a computer vision system consisting of an algorithm for landmark detection and tracking, which estimates the UAV motion (position and linear velocity) relative to a landing pad on the ground. Real time experiments are performed using the quad-rotor platform shown in Figure 1. This prototype, capable of quasi-stationary flight, was built at the University of Technology of Compiegne. Experimental results prove that the information generated by the vision system ensures that the UAV can maintain the selected desired position relative to a landmark.

This paper is organized as follows. Section 2 presents the developed vision system. The mechanical configuration of the UAV and the embedded electronics are presented in Section 3. The fundamental equations of motion for a quad-rotor helicopter UAV are presented in Section 4. Section 5 is devoted to the proposed control strategies for hovering flight and position hold. Experimental results in Section 6. Finally, some concluding remarks are presented in Section 7.

2. VISION SYSTEM

The vehicle's location with respect to a well-known reference frame is a necessary information for the controller, in order to generate the control inputs that will stabilize the UAV over a desired (X-Y-Z) position. This section presents a vision system setup, which consists of an aerial vehicle, an onboard calibrated camera, a landing pad on ground, and a vision algorithm running on a ground station PC. This system is intended to estimate the quad-rotor position.

2.1 Visual System Setup

The system configuration showed in Figure 2 can be described as follows.

- The aerial vehicle with body fixed frame (X_h, Y_h, Z_h). Z_h represents the yaw axis, pointing upwards. X_h and Y_h are the pitch and roll axis respectively.
- A strapdown camera pointing downwards, having (X_c, Y_c, Z_c) as reference frame. When moving, the camera surveys the scene passing below. Notice that the $X_c - Y_c$ and $X_h - Y_h$ planes are parallel, therefore the information collected by the camera can be used to stabilize the vehicle.
- The landing pad, formed by four circles on high contrast background placed underneath the rotorcraft. The coordinates frame (X_{lp}, Y_{lp}, Z_{lp}) represents the inertial reference frame.

When the UAV is in hover flight over the landing pad, the planes formed by $(X_h - Y_h)$ and $(X_{lp} - Y_{lp})$ are considered

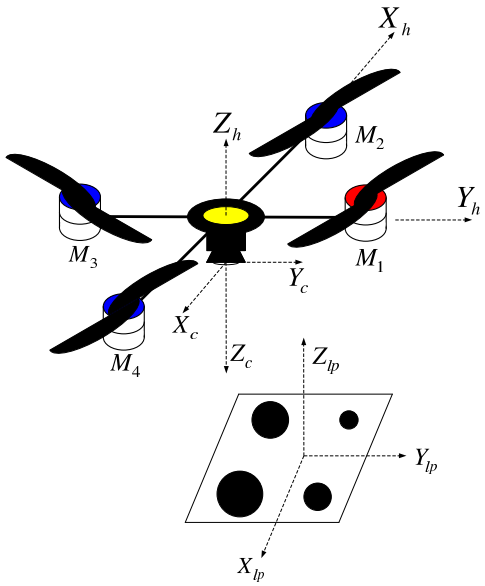

Figure 2. Visual system setup.

to be parallel. Computing the extrinsic parameters of the camera at every image frame, this information can be used to estimate the relative position of the UAV with respect to the landing pad.

2.2 Vision Algorithm for Camera Pose Estimation Using Planar Homographies

The vision algorithm used in our system creates an homography H for each view of the landing pad circles. Using the homogeneous coordinates we can express the action of the homography as (12):

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = sM[r_1\, r_2\, r_3\, \mathbf{t}] \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix} = sM[r_1\, r_2\, \mathbf{t}] \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} \quad (1)$$

where $[x, y, 1]$ represents the landing pad position in the camera image, s is an arbitrary scale factor, $M \in \mathbb{R}^{3\times3}$ represents the camera intrinsic matrix, $r_1, r_2, r_3 \in \mathbb{R}^{3\times1}$ are extrinsics rotation parameters, $\mathbf{t} \in \mathbb{R}^{3\times1}$ is the extrinsics translation parameters vector, $[X, Y, 0, 1]$ is the real landing pad position, and $H = sM[r_1\ r_2\ \mathbf{t}]$ is the homography matrix. The homography matrix H is divided in two parts: the physical transformation (which locates the observed object plane) and the projection (camera intrinsic matrix). For every instant, when the aerial vehicle is in hover flight over the landing pad, it is possible to compute this homography matrix using the a priori knowledge of the position of the four centroids of the circles (13). Using this estimated transformation matrix, and because the intrinsic camera matrix is constant and was previously identified by an off-line calibration, we are able to calculate the camera extrinsic parameters, and therefore we have the position of the quad-rotor.

For the purpose of eliminating any erroneous detection of the circles, a parallelism verification procedure is performed over the lines mapped from the landing pad. From

Figure 2, we see that the four circles of the landing pad are positioned forming the corners of a rectangle. Thus, the line between the two upper corners and the line joining the two lower corners must be are parallel. The same characteristic is valid for the line joining the two left corners and the line between the two right corners are parallel. The procedure from parallelism verification is based on the slope of a line equation:

$$m = \frac{y_f - y_i}{x_f - x_i} \tag{2}$$

where i and f stand for final and initial coordinates respectively. Thus, the slope m_{up} of the upper line must be almost equal to the slope m_{lo} of the lower line. At the same time the slope m_{le} of the left line must be almost equal to the slope m_{ri} of the right line:

$$|m_{up} - m_{lo}| < \epsilon \quad \text{and} \quad |m_{le} - m_{ri}| < \epsilon \tag{3}$$

here ϵ stands for a parameter helping to determine the lines parallelism. Once this verification has been validated, we know that the four circles of the landing pad have been detected successfully, thus, a good planar homography could be estimated, resulting in a good computation of the camera extrinsic parameters.

When the UAV is in hover flight, the image of the landing pad viewed by the camera presents some displacements due to the quad-rotor movements. To track the displacement, an optical flow computation process can be performed. From the different approaches available for computing optical flow (14), we have implemented the Lucas-Kanade pyramidal algorithm (15) in combination with a feature-detecting algorithm. This method provides an accurate estimation of the motion field since it does not take into account the non landing pad areas, where the motion field cannot be accurately determined. The tracking process of the four circles centroids is performed by defining a region of interest around each one of the circles using its radius magnitude. Then, the most representative features over each one of these regions are selected as features to track for. These features are usually the circles perimeter. Once this group of features has been defined, a tracking process is performed over the entire image, and the optical flow is estimated, based on the image displacements of the features tracked. Next step consists on using optical flow values in order to estimate the new position of the circles centroid.

Aiming to use this tracked landing pad position in our extrinsic camera parameters estimation algorithm, equations (2) and (3) must be also satisfied. Although the circles tracking process is performed over each new image, this data will not be used for computing the extrinsic parameters of the camera, unless the four circles centroids real position data is not available or does not satisfy the parallelism condition.

Considering that there are no moving objects or obstacles in the visual field of the camera, and assuming rigid body motion, optical flow can be expressed as

$$\begin{aligned} \bar{OF}_x^d &= \bar{V}_{OF_\mathrm{x}}^d + K_{xy}^x \omega_\mathrm{x} - K_{x^2}^x \omega_\mathrm{y} + K_y^x \omega_\mathrm{z} \\ \bar{OF}_y^d &= \bar{V}_{OF_\mathrm{y}}^d + K_{y^2}^y \omega_\mathrm{x} - K_{xy}^y \omega_\mathrm{y} - K_x^y \omega_\mathrm{z} \end{aligned} \tag{4}$$

with

$$\begin{aligned} \bar{V}_{OF_\mathrm{x}}^d &= -f\frac{V_\mathrm{x}}{Z} + K_x^x \frac{V_\mathrm{z}}{Z} \\ \bar{V}_{OF_\mathrm{y}}^d &= -f\frac{V_\mathrm{y}}{Z} + K_y^y \frac{V_\mathrm{z}}{Z} \end{aligned} \tag{5}$$

where $\bar{OF}_x^d$ and $\bar{OF}_y^d$ are the optical flows in the image coordinate system. $\bar{V}_{OF_\mathrm{x}}^d$, $\bar{V}_{OF_\mathrm{y}}^d$ are the relative velocities of the vehicle in the image coordinate system, K_j^i are known scale factors depending on intrinsic parameters of the camera, and $\omega_\mathrm{x}, \omega_\mathrm{y}$ and ω_z are the UAV angular velocities.

Due to the attitude control law, the optical flow induced by rotations of the engine will be considered negligible with respect to the optical flow produced by translational movement. Therefore the optical flow expression is simplified as

$$\begin{aligned} \bar{OF}_x^d &= \bar{V}_{OF_\mathrm{x}}^d + \upsilon_1 \\ \bar{OF}_y^d &= \bar{V}_{OF_\mathrm{y}}^d + \upsilon_2 \end{aligned} \tag{6}$$

where υ_1 and υ_2 are noise signals. We compute a relative velocity as an average value of the optical flow produced by the displacement of the four circles centroids. A noise signal is added into the process, which takes into account the small changes in the attitude due to the position correction:

$$\begin{aligned} \rho_x^{k+1} &= \rho_x^k + \Delta T \bar{V}_{OF_\mathrm{x}}^d + \nu_x \\ \rho_y^{k+1} &= \rho_y^k + \Delta T \bar{V}_{OF_\mathrm{y}}^d + \nu_y \end{aligned} \tag{7}$$

where ρ_x^k and ρ_y^k represent the circle's centroid position, ΔT is the sampling period and ν_x and ν_y are noise signals.

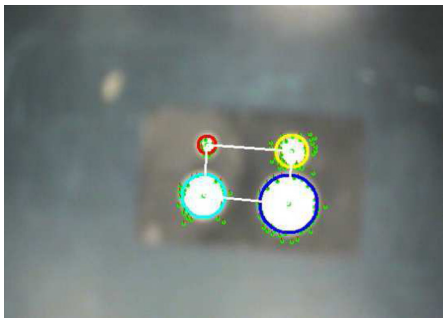

Figure 3. Detection and tracking of the four landing pad circles.

Figure 3 shows the landing pad image when using the proposed video processing technique. The four circles of the landing pad are detected and identified from smaller to greater. The parallelism condition is also verified. Points around the circles perimeter represent the characteristics selected to be the features to track for.

3. SYSTEM CONFIGURATION

3.1 Aerial Vehicle

The rotorcraft was developed in the University of Technology of Compiegne, France. The distance (center to center) between two same-axis rotors is 40cm, and its total weight is 800 grams. The aerial vehicle is powered by a Li-Po battery which drives four electric brushless motors, allowing a flight autonomy of 15 minutes approximately. Figure 1 shows the developed prototype.

The electronic embedded system has two interconnected boards: the first board is the control unit, and the second one deals with the motors drivers. The control unit card is showed in Figure 4. This electronic card performs the essential tasks of sensing, communicating and stabilizing the UAV attitude during fly. Properties of this board can be summarized as follows:

- Texas Instruments TMS320F2812 DSP module, used to process the data coming from different sensing devices and compute the control algorithm, before sending it in the form of four PWM signals to the motor drivers.
- The inertial sensors: a MIDG II INS/GPS from Microbotics Inc. is used to measure the angular position of the rotorcraft. We also use three additional gyros to measure angular velocity at a higher rate.
- Freescale MPX4115A atmospheric pressure sensor, used with an appropriate amplifier circuit, to measure the altitude of the engine on an appropriate sensing range.
- Battery voltage measurement circuit: this electronic circuit is intended to provide the actual tension level of the supply battery. This information is used for several goals: perform a safety landing and turn-off before an unwanted discharge of tension (avoiding accidents). Also, information regarding the supply voltage level is used in a preprocessing stage of the incoming measurements from the atmospheric pressure sensor.
- XBee ZB ZigBee PRO radio modem: the link between the base station and the aerial vehicle is made by a 2.45 GHz IEEE 802.15.4 wireless sensor networks. This communication link can be used to introduce external control inputs, send the sensors information to the ground station, etc.

The second board contains:

- Signal conditioning circuits:
  In this stage each control input of the four motors is decoupled from the rest of the electronic systems. The PWM signals are also filtered and conditioned.

3.2 Base Station

Consisting of a desktop PC, a flight simulator Cyborg-X joystick and a XBee ZB ZigBee PRO Radio Modem, the base station sends information triggered by the user, to the aerial vehicle. It also receives and saves all information needed to debug and analyze the flight experiments and results. From this base, different flying modes can be chosen: manual control, altitude stabilization using the

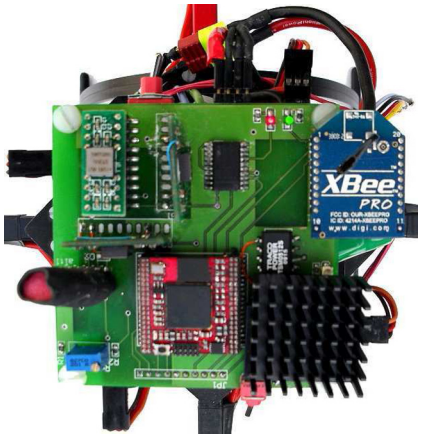

Figure 4. The electronics on board. Electronic card for the DSP, rate gyros, IMU connnections, atmospheric pressure sensor and wireless modem.

pressure sensor, vision-based position hold and reactive navigation. The platform also includes safety features, like an emergency stop switch and a condition that verifies that the thrust command on the ground station is at zero before starting the motors. Image in Figure 5 shows our complete system: UAV and ground station.

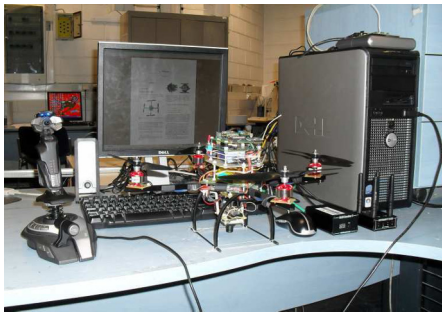

Figure 5. The experimental platform: UAV and ground station.

3.3 Vision System

Figure 6. The UAV vision system: (a) CTDM-5351 camera on board. (b) 4-Antenna Diversity System Receiver.

The UAV vision system, showed in Figure 6, consists of a high definition CTDM-5351 camera with a resolution of 640×480 pixels, placed pointing downwards. The camera is connected to a 200mW micro video and audio radio frequency transmitter. Images from the vision system are recovered by a 4-Antenna Diversity System Receiver. This receiver is connected to the ground station PC using a USB frame grabber. The frequency of the video transmission is performed at a rate of 30 frames per

second. This system is intended to perform diverse tasks such as altitude estimation using landmarks, navigation and obstacle avoidance. Vision algorithms run in the ground station PC, and are programmed in Visual C++ using OpenCV functions (16), which ensures a fast image capture and processing.

4. DYNAMICAL MODEL

The position of the quad-rotor's center of gravity, with respect to the inertial frame, is denoted by $\xi = [x\ y\ z]^T \in \mathbb{R}^3$. The three Euler angles (roll, pitch and yaw), which represent the orientation of the vehicle, are expressed as $\eta = [\phi\ \theta\ \psi]^T \in \mathbb{R}^3$. The simplified model of the rotorcraft dynamics is obtained from Euler-Lagrange equations (17):

$$m\ddot{x} = -u\sin\theta \tag{8}$$
$$m\ddot{y} = u\cos\theta\sin\phi \tag{9}$$
$$m\ddot{z} = u\cos\theta\cos\phi - mg \tag{10}$$
$$\ddot{\theta} = \tilde{\tau}_\theta \tag{11}$$
$$\ddot{\phi} = \tilde{\tau}_\phi \tag{12}$$
$$\ddot{\psi} = \tilde{\tau}_\psi \tag{13}$$

where u is the main thrust directed out of the top of the aircraft, x and y are coordinates in the horizontal plane, z is the vertical position, 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.

5. CONTROL STRATEGY

5.1 Attitude Control

Given equations (11), (12) and (13) we can identify three independent systems. In order to assure the stabilization of such system we propose the following control strategy, based on PD controllers.

$$\tilde{\tau}_\theta = -K_p^{\theta\phi}\theta - K_d^{\theta\phi}\dot{\theta} \tag{14}$$
$$\tilde{\tau}_\phi = -K_p^{\theta\phi}\phi - K_d^{\theta\phi}\dot{\phi} \tag{15}$$
$$\tilde{\tau}_\psi = -K_p^{\psi}\psi - K_d^{\psi}\dot{\psi} \tag{16}$$

5.2 Position Control

The position's control signals are sent as joystick inputs to the UAV. We propose a PD controller to stabilize the position on the $(X-Y)$ plane and a PI controller for altitude stabilization as follows

$$x_u = -K_p(T_x^k - T_x^0) - K_d(T_x^k - T_x^{k-1}) \tag{17}$$
$$y_u = -K_p(T_y^k - T_y^0) - K_d(T_y^k - T_y^{k-1}) \tag{18}$$
$$z_u = -K_i^z\sum(T_z^j - T_z^0) - K_p^z(T_z^k - T_z^0) \tag{19}$$

where (T_x^k, T_y^k, T_z^k) are the estimated position computed by the vision system at time k, (T_x^0, T_y^0, T_z^0) are the initial position references, K_p, K_d, K_i^z, K_p^z are constants and x_u, y_u, z_u are the position control inputs.

6. EXPERIMENTAL APPLICATION

The position estimation algorithm is programmed in C/C++ and is based on OpenCV functions. it uses an image resolution of 640×480 pixels and provides the position feedback information, while the embedded inertial electronic system provides the attitude data. To verify the performance of the algorithm and the proposed controllers, we conducted a set of two experiments which are described next.

A desired (X-Y-Z) vehicle position is selected when the UAV is flying over the landing pad, it will be used as a position reference in the rest of the experiment. Notice that the attitude stabilization control system is always running at high frequency to guarantee that the Euler angles are close to zero, which simplifies the image processing tasks. The set of Figures 7, 8, 9 shows the results (euler angles, position error and velocities respectively) for the experiment performed.

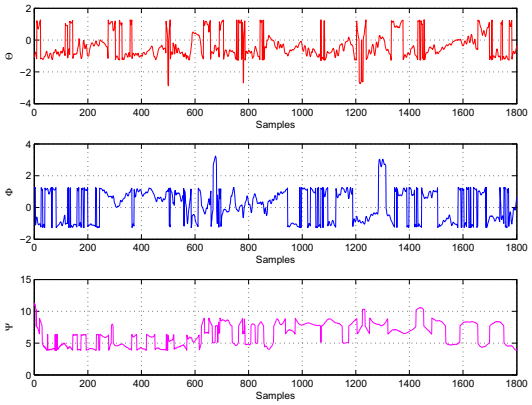

Figure 7. Euler angles, experimental results

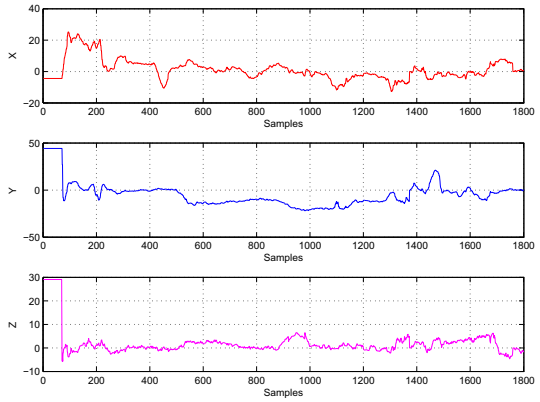

Figure 8. Position errors, experimental results

It is important to note that the pitch and roll angles always remain in the interval $(-1.5, 1.5)$ degrees. Therefore, it can be concluded that the position control adds only small changes in the attitude of the rotorcraft for bringing the actual position to the desired one. This is an important property because the position controller runs at a lower rate compared to the attitude controller, then, a smooth

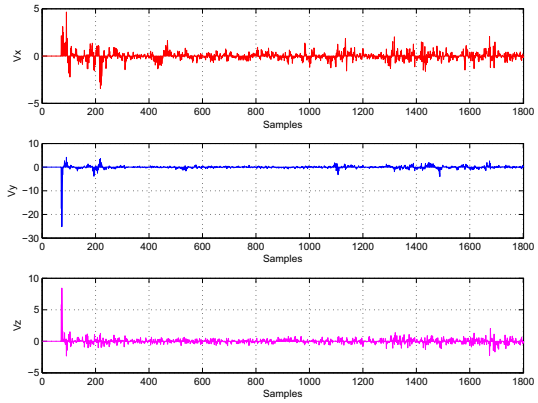

Figure 9. Velocities, experimental results

position control is necessary to ensure the global stability of the vehicle.

7. CONCLUDING REMARKS

A vision algorithm for detection and tracking of a landmark was proposed and tested in a real time application. The monocular vision system was used to estimate the position of a UAV with respect to a landing pad, using an homography estimation technique. Optical flow is also obtained from the vision system to estimate the translational UAV speed. A control algorithm was implemented to stabilize the UAV's attitude, running at a higher rate as compared to than the image processing rate. The control algorithm ensures that the Euler angles of the vehicle are very close to zero which considerably simplifies all the tasks concerning image processing (position estimation, optical flow).

Several experiments were successfully performed indoor showing that the quad-rotor was stabilized at a selected position above the landing pad. The attitude of the vehicle was not significantly perturbed by the control input used to correct the UAV position. The vehicles's velocity remained also very close to zero.

A video of the experiments can be seen at

`http://www.youtube.com/watch?v=SQlSXruTnj0`

Future work will focus on the improvement of the quad-rotor platform to increase its robustness in presence of wind gust, aiming at performing autonomous UAV takeoff and landing tasks indoors as well as outdoors.

REFERENCES

[1] J.R. Reinhardt, J. James, and E. Flannagan, *Future Employment of UAVS: Issues of Jointness*, Joint Force Quarterly, (Summer 1999), 39.

[2] F. Kendoul, D. Lara, I. Fantoni-Coichot and R. Lozano, *Real-time nonlinear embedded control for an autonomous quadrotor helicopter*, AIAA, Journal of Guidance, Control and Dynamics, vol. 30, issue 4, pp. 1049-1061, July-August 2007.

[3] S. Salazar, J. Escareno, D. Lara and R. Lozano, *Embedded control system for a four rotor UAV*, International Journal of Adaptive Control and Signal Processing, vol. 21, issue 2-3, pp. 189-204, March-April 2007.

[4] S. Saripalli, J. Montgomery and G. Sukhatme, *Vision-based Autonomous Landing of an Unmanned Aerial Vehicle*, IEEE International Conference on Robotics and Automation, pp. 2799-2804, 2002.

[5] S. Salazar, H. Romero, R. Lozano and P. Castillo, *Modeling and Real-Time Stabilization of an Aircraft Having Eight Rotors*, Journal of Intelligent and Robotic Systems, vol. 54, issue 1-3, pp. 455 - 470, March 2009.

[6] Z.F. Yang, and W.H. Tsai, *Using parallel line information for vision-based landmark location estimation and an application to automatic helicopter landing*, Robotics and Computer-Integrated Manufacturing, vol. 14, no. 4, pp. 297-306, August 1998.

[7] S. Saripalli, J. Montgomery and G. Sukhatme, *Visually-Guided Landing of an Unmanned Aerial Vehicle*, IEEE Transactions on Robotics and Automation, vol. 19, no. 3, pp. 371-381, Jun 2003.

[8] H. Romero, S. Salazar, R. Lozano, *Real-Time Stabilization of an Eight-Rotor UAV Using Optical Flow*, In IEEE Transactions on Robotics, vol. 25, issue 4, pp.809-817, August 2009.

[9] E. Rondon, S. Salazar, J. Escareno and R. Lozano, *Vision-Based Position Control of a Two-rotor VTOL miniUAV*, Journal of Intelligent and Robotic Systems, DOI 10.1007/s10846-009-9370-6, 2009.

[10] M. Achtelik, A. Bachrach, R. He, S. Prentice and N. Roy, *Autonomous navigation and exploration of a quadrotor helicopter in GPS-denied indoor environments*, in First Symposium on Indoor Flight Issues. Mayagüez, Puerto Rico. July 2009.

[11] C. Demonceaux, P. Vasseur and C. Pégard, *Omnidirectional vision on UAV for attitude computation*, In IEEE International Conference on Robotics and Automation, May 2006.

[12] G. Bradski and A. Kaehler, *Learning OpenCV, Computer Vision with the OpenCV Library*, O'Reilly Media, September 2008.

[13] Horaud R., Conio B., Leboulleux O. and Lacolle B., *An Analytic Solution for the Perspective 4-Point Problem*, Computer Vision, Graphics and Image Processing, Vol. 47, pp. 33–44, 1989.

[14] J.L. Barron D.J. Fleet S.S. Beauchemin, *Performance of optical flow techniques*, In International Journal of Computer Vision, vol. 12, issue 1, pp. 43-77, February 1994.

[15] J.Y. Bouguet, *Pyramidal Implementation of the Lucas Kanade Feature Tracker – Description of the algorithm*, Intel Corporation – Microprocessor Research Labs.

[16] *Open Computer Vision Lib*, http://sourceforge.net/projects/opencvlibrary/

[17] A. Sanchez, P. Garcia, P. Castillo and R. Lozano, *Simple Real-Time Stabilization of Vertical Takeoff and Landing Aircraft with Bounded Signals*, Journal of Guidance, Control, and Dynamics, vol. 31, issue 4, pp. 1166-1176, July-August 2008.