

A Pan-Tilt Camera Fuzzy Vision Controller on an Unmanned Aerial Vehicle

Miguel A. Olivares-Méndez, Pascual Campoy, Carol Martínez and Iván Mondragón
Computer Vision Group
Universidad Politécnica de Madrid
José Gutierrez Abascal 2, 28006, Madrid, Spain

Abstract—This paper presents an implementation of two Fuzzy Logic controllers working in parallel for a pan-tilt camera platform on an UAV. This implementation uses a basic Lucas-Kanade tracker algorithm, which sends information about the error between the center of the object to track and the center of the image, to the Fuzzy controller. This information is enough for the controller to follow the object by moving a two axis servo-platform, regardless the UAV vibrations and movements. The two Fuzzy controllers for each axis, work with a rules-base of 49 rules, two inputs and one output with a more significant sector defined to improve the behavior of those controllers. The controllers have shown very good performances in real flights for static objects, tested on the Colibri prototypes.

I. INTRODUCTION

Rotary UAV vibrations and long focal distances are big drawbacks for visual tracking and inspection from UAVs, which can be solve by using fuzzy controllers, as shown in this paper.

For the computer vision group at UPM the main research interest is to incorporate vision systems in UAVs in order to increase their navigation capabilities and applications. The image processing algorithms and tracking techniques are the main path of work of this researcher group, using them for different tasks, such as up-level control, state estimation, visual navigation and many others.

The increasing interest in UAVs, the significant developments on UAVs fields on control architectures and, also, the cheap and suitable alternative that it brings, is beginning a new kind of applications that is opening to the researching groups with similar robots and techniques. Currently, some applications have been developed, among which we can find Valavanis' works on traffic monitoring [1], and fire detection [2], the Rathinam uses of this platform for an autonomous searching and tracking of a river [3], the MacArthur unmanned ground vehicle state estimation [4] and the works in obstacles avoidance and 3D path planning for UAVs by [5].

Using a pan-tilt camera platform is an option to increase the possibilities to detect and track objects. It has a successful behavior regardless the vibrations of the helicopter and the unexpected movements of the objects. Related with this topic, there are some laboratory tests works made, like [6] based on the tracking on a biomimetic Eye, and real tested works used for a target tracking and motion estimation for moving targets by [7] or a vision assisted autonomous path following by [8].

Also, Fuzzy logic gives us a very suitable platform for creating a controller that has a rapid and good response for non linear movements. This technique give us easy, rapid and comprehensive way to create controllers by the definition of the linguistic value of its variables, base of rules and the easy way to tuning this kind of controllers, against the uses of the classical PID controllers, for what it is necessary to calculate the model of the helicopter and the visual platform.

Here, we present a pan and tilt video platform onboard a UAV, using a Lucas-Kanade tracker; the platform is controlled by two Fuzzy-controllers in order to follow static objects, in despite of the aircraft movements. This papers is divided in the following sections. Section II shows a briefly description of the different components of our UAVs. In Section III we describe the visual tracking used for this work, to continue, in Section IV, with the description of the Fuzzy software implementation. The next one, Section V, explains the fusion of the Fuzzy controller and the visual algorithm for the visual object tracking system. Experimental results divided in lab test and the UAV tests has been shown in the Section VI, to finalize with the Conclusions and the future works in Section VIII.

II. UAV SYSTEM DESCRIPTION

The Colibri project has three totally operative UAV platforms. One electric helicopter, and two gas powered (Fig. 1). The COLIBRI testbeds [9], are equipped with an xscale-based flight computer augmented with sensors (GPS, IMU, Magnetometer, fused with a Kalman filter for state estimation). Additionally it includes a two axis video-platform. In order to enable it to perform vision processing, it has a VIA mini-ITX 1.5 GHz onboard computer with 1 Gb RAM, a wireless interface, and support for many Firewire cameras including Mono (BW), RAW Bayer, color, and stereo head. It is possible to use IP and analog cameras as well.

The system runs in a client-server architecture using TCP/UDP messages. 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 the 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 integrated with the flight control through this layer using TCP/UDP messages. The layer is based on a communication



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

API where all messages and data types are defined. The helicopter's low-level controller is based on PID control loops to ensure its stability. The higher level controller uses various sensing mechanisms such as GPS and/or vision to perform tasks like navigation, landing, and visual tracking, among others.

III. VISUAL TRACKING

Image processing is used to find characteristics in the image that can be used to recognize an object or points of interest. This relevant information extracted from the image (called features) ranges from simple structures, such as points or edges, to more complex structures, like objects. These features will be used as reference for the visual flight control.

Most of the features used as reference are interest points, which are points in an image that have a well-defined position gradient maximum or minimum, so can be robustly detected. Some of these points are corners formed by the intersection of two edges, and some others are points in the image whose context has rich information based on the intensity of the pixels. the Harris corner detector [10] is used for this purpose is. It extracts a lot of corners very quickly based on the magnitude of the eigenvalues of the autocorrelation matrix. However, it is not enough to use this measure in order to guarantee the robustness of the corner, since the purpose of the features' extraction is to track them along an image sequence. This means that good features to track have to be selected in order to ensure the stability of the tracking process. The robustness of a corner extracted with the Harris detector can be measured by changing the size of the the detection window, which is increased to test the stability of the position of the extracted corners. The measure of this variation is then calculated based on a maximum difference criteria. Besides, the magnitude of the eigenvalues

is used only to keep features with eigenvalues higher than a minimum value. Combination of such criteria leads to the selection of the better features to track.

The problem of tracking features can be solved with different approaches. The most popular algorithm to track features like corner ones or interest points in consecutive images is the Lucas-Kanade algorithm [11]. It works under two premises: first, the intensity constancy in the vicinity of each pixel is considered as a feature; secondly, the change in the position of the features between two consecutive frames must be minimum, so that the features are close enough to each other. Given these conditions to ensure the performance of the algorithm, it can be expressed in the following form: if we have a feature position $p_i = (x, y)$ in the image I_k , the objective of the tracker is to find the position of the same feature in the image I_{k+1} that fits the expression $p'_i = (x, y) + t$, where $t = (t_x, t_y)$. The t vector is known as the optical flow, and it is defined as the visual velocity that minimizes the residual function $e(t)$ defined as:

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

where $w(W)$ is a function which assigns different weights to comparison window W . This equation can be solved for each tracked feature, but since it is expected that all features on physical objects move solidary, summation can be done over all features. The problem can be reformulated to make it possible to be solved in relation to all features in the form of a least squares' problem, having a closed form solution. Whenever features are tracked from one frame to another in the image, the measure of the position is affected by noise. This method is also desirable because it provides an estimation of the velocity of the pixel that is used as a reference to the velocity flight control of the UAV, like is shown in [9].

IV. FUZZY SOFTWARE IMPLEMENTATION (MOFS)

For this work is used the MOFS (Miguel Olivares Fuzzy Software), developed at first moment for an autonomous fuzzy navigation system applied in mobile robots [12] and [13]. This software was independently designed for the c++ platform. One class is defined for each part of the fuzzy-logic environment in order to facilitate the future updates and making easier work with it. There are different classes for variables, rules, membership functions and defuzzification modes. Depending on the system to create is possible to define the number of inputs and outputs or make different systems in serial mode, where the output of one system could be the input of another, also, it is possible to define two or more fuzzy controllers working in parallel. In the same way it is defined the different characteristics of the variables, the fuzzification inference type or the defuzzification mode.

The updates of the software can be implemented in each way of the fuzzy-logic parts, like introducing different membership functions, fuzzy inference types or introducing another kind of defuzzification mode.

One of the differences between this software and others Fuzzy softwares is that it lets to the user represent a more

important sector in one or more fuzzy variables, giving us the possibility to reduce the size of the rule-base, in order to get better responds at the same time that this can reduced the computational cost to obtain the output of the fuzzy controller. The MOFS has a direct communication with the server by TCP/UDP sockets, in another client-server structure, getting the information of the error between the center of the object and the center of the image and returning the movements in radians that the two servos of the platform must do. The idea of making a communication between server and the controller by sockets is to create a suitable base to use different controllers in the future, making a specific testbed for those.

V. VISION FUZZY TRACKING SYSTEM

The Fuzzy Logic give us a more versatile solution to control the platform and the helicopter, because it is easier to tune and to adapt to the real world, in order to represent non-linear problems, giving a better solution to face the vibrations of the helicopter and the other perturbation signals of the environment.

The fusion between the tracker and the controller of the platform was made using the explained MOFS-controller and the Lukas-Kanade algorithm. We put on the platform a fire-wire camera that takes images with 320x240 pixels resolution. We get the information of the error in pixels, getting a -160 to 160 pixels maximum yaw error and a -120 to 120 pixels error for the pitch axis. Based on it the problems are divided by two MOFS-Fuzzy controllers running in parallel, one for the pitch axis and the other for the yaw axis of the platform.

The fuzzification of the inputs and the outputs are defined by using a triangular membership function. The controller have two inputs, the error between the center of the object and the center of the image and the difference between the last and the actual error (fig. 2(a) and 2(b)). The output of the controller represents how many degrees the servo-motor must turn to gets the center of the object in the center of the image. All of it are guided by a 49 rules base. For the output we defined a more important sector in the section near to zero, as shown in fig. 2(c). This option gives the possibility to define a very sensible controller when the error is too little (when the object is close to the center of the image) and great movement response controller when the object is so far.

For the inference model (inside the defuzzification process) we used a product classic method, and for the defuzzification part itself, we used the Height Method, Eq. 2.

$$y = \frac{\sum_{l=1}^M \bar{y}^l \prod (\mu_{B'}(\bar{y}^l))}{\sum_{l=1}^M \prod (\mu_{B'}(\bar{y}^l))} \quad (2)$$

VI. EXPERIMENTAL RESULTS

In this section two kind of experiments will be show. At first we will present some tests in the laboratory in which is found the best rules-base for the Fuzzy controller that had the better response. In the second subsection we will present

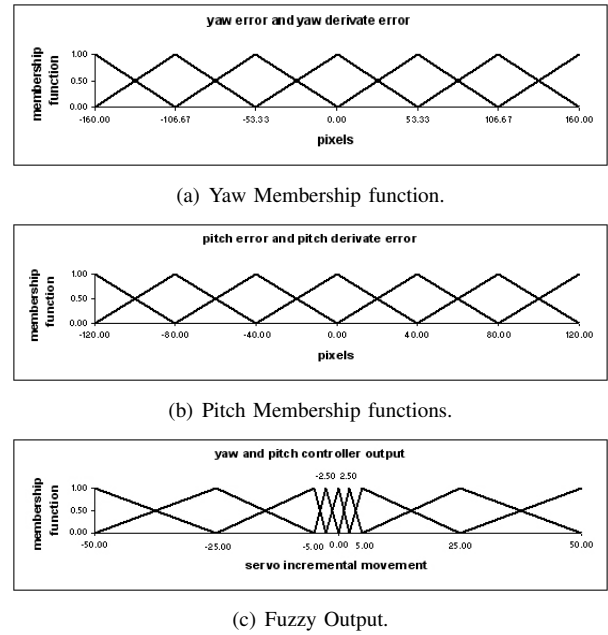


Fig. 2. Variables of the Fuzzy-MOFS controllers.

real tests on one of the UAV systems and the behaviors of the parallel fuzzy controllers during a flight.

A. Laboratory Tests

We made more than 20 tests in the laboratory, in order to find the rule base which had the best behavior following moving objects with the movements of the platform. For those tests we change the action range and the number of sets of the outputs. To measure the different behaviors of those systems, we compare results based on the stabilization time, the type, value and the number of the oscillations of the system's response in the same or very similar static object tests. For those, a platform is started at the same initial position, and the static object to follow, trying to made a real step signal. Some obtained results are shown in Fig. 3 and in the Table I.

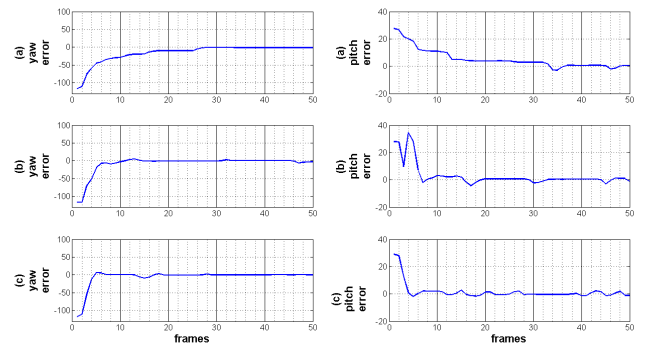


Fig. 3. Different rules-base and output definition fuzzy systems used in laboratory tests. a) Without more significant sector defined in the output. b) With a output minor range of action than the one show in fig. 2(c), sets in -10, -7.5, -5, -2.5, 0, 2.5, 5, 7.5 and 10. c) the selected one shown in fig. 2(c)

B. Tests on UAV

For the tests on the UAV, a static mark was tracked, making real flights with one of our helicopter. The tracking of the mark was made from the take off, to the landing of the UAV. This flight was made by sending set-points from the ground station making different movements in the helicopter attitude, as are shown in the next figures. In fig. 4 is shown a 3D reconstruction of the flight using the GPS and the IMU data. The X axis of the figures is measured by the number of frames that had been taken during the flight. Knowing that in those tests we are working with a frame-rate of 15 frames per seconds, those 2500 frames represent a full flight of almost 3 minutes.

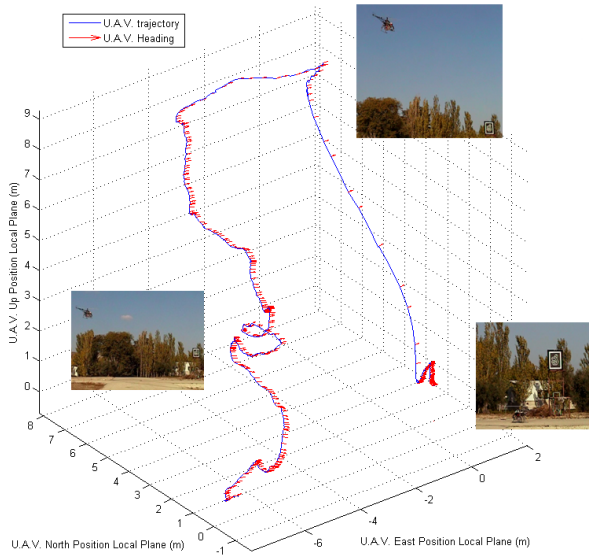


Fig. 4. 3D flight reconstruction from the GPS and the IMU data from the UAV. Where, the 'X' axis represents the NORTH axis of the surface of the tangent of the earth, the 'Y' axis represents the EAST axis of the earth, the 'Z' is the altitude of the helicopter and the red arrows show the pitch angle of the helicopter.

In the Fig. 4 it can be seen the take-off from the point (0,0,0), being (North, East, altitude), and positioned over the north axis facing to the east, where is the mark to follow. After that, the aircraft was risen with a quick movement up to 9 meters like is shown in fig. 5(a). Also, it made a movement of 6 meters in the north axis, that represent a sideward movement (fig. 5(c)) from 600 to 700 frames. Later the aerial vehicle was driven to a downward position,

TABLE I
MEASURES OF SOME DIFFERENT FUZZY SYSTEM.

Fuzzy System	axis	frames to stabilization
a)	Yaw	26
a)	Pitch	35
b)	Yaw	10
b)	Pitch	10
c)	Yaw	5
c)	Pitch	5

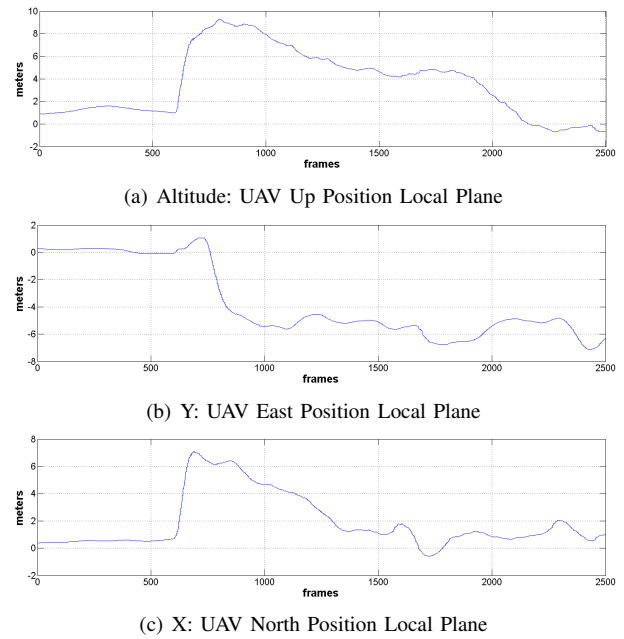


Fig. 5. Different axis in X, Y and Z movements of the UAV flight.

with a 6 meters movement, as we can see from 700 to 800 frames in the fig. 5(b). Continue with a different range of soft movements over the three axis in order to test the behavior of the fuzzy vision tracker, to finish with the landing of the UAV.

The information of the movements in the North, East and altitude axis and the one from the attitude movements of Yaw, Pitch and Roll give a better measure about all the problems to solve by the controller and the tracker for tracking and follow the mark. In fig. 6 is shown this division of the attitude of the UAV.

With those figures it is possible to realize and explain the following error of the fuzzy vision controller, shown in fig. 7. At first, we can notice that the error in yaw is greater than in the pitch axis of the platform, this is explained because this is the axis where more changes are made by the helicopter. At the beginning there is very big error in the yaw axis, $\pm 100\text{pixels}$, caused by the vibrations of the motor ignition of the helicopter. Putting this initial error apart, we see that the big error of the Pitch Fuzzy-controller is, just, the interval of 570-600 frames (sector number 1), which correspond to the great change of the pitch angle of the helicopter (fig. 6(a)). In Fig. 7 is possible to see that this error is solved by the controller with fast movements in, just, few frames. Talking about the yaw error, the most significant figure is fig. 6(b), because it represents the UAV's orientation changes. For a better explanation we divided the yaw error in four sectors clearly defined:

1. From 540 to 660: In this interval, there are three big changes in fig.6(b), of +8, -22 and +20 degrees. Also, there is little influence of the roll of the UAV (fig. 6(c)), in which there are changes of -5 and +10 degrees. The controller must compensate

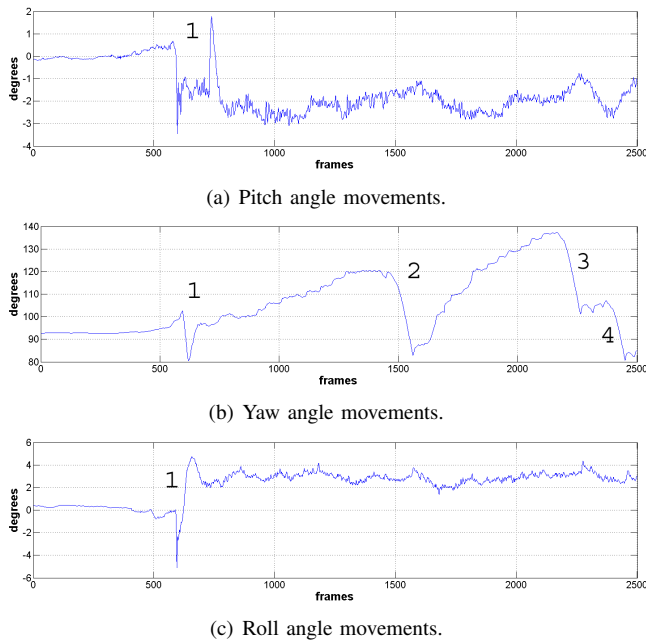


Fig. 6. Different pitch, yaw and roll movements of the UAV.

a maximum error of 100, 50 and -75 pixels, like it is possible to see in fig.7.

2. From 1460 to 1720: In this section, it is shown two big changes in the yaw of the helicopter of -40 and +28 degrees, giving a maximum errors of +52 and -42 pixels.
3. From 2176 to 2260: At this time, the helicopter made a change of -35 degrees in the yaw angle, causing a maximum error of 55 pixels.
4. From 2375 to 2450: Finally in this last part of the flight, 48 pixels maximum error has been provoked by an angle's variation of 27 degrees in the yaw angle too.

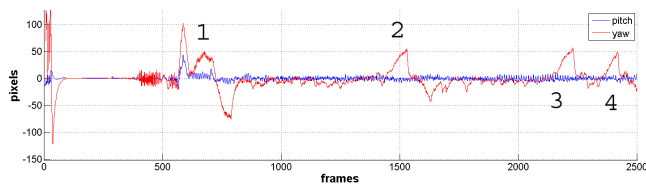


Fig. 7. Error between center of the image and center of the object to track.

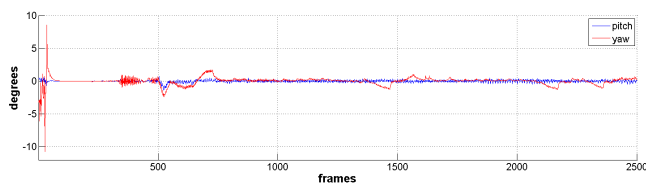


Fig. 8. Output from the Fuzzy Controller.

In Fig. 8 is shown the output of the two Fuzzy-MOFS controllers in order to compensate the error caused by the

changes of the different movements and angle's changes of the UAV flight, where we can see the different responses of the controllers, depending the sizes and the types of the perturbations. In order to put all together the data form all the figures, the Table III is created. This table had been divided by these more significant sections (1,2,3,4), and are measured the size of the perturbation in degrees and the size of the intervals that the perturbation made some action. Also, is measured the relation of the perturbation's degrees per seconds to become aware of how big is the perturbation, the size of the maximum error at the attitude axis of the helicopter and what axis of the platform had been affected. It is possible to see that the perturbations affect the control system depending on the size and the type of them. We had a maximum error in the flight (taking away the ignition of the motor) of +100 pixels during the initial elevation of the aircraft, the most critical section of the flight, and a maximum error of 55 pixels during the flight. Taking into account that we have a 320x240 resolution, the error represents a 62.2% in the initial elevation, where we have a fusion of all the possible movements and angles' variations in a UAV, with a bigger change of 15.38 degrees per sec. and a 34.375% of the yaw axis of the camera, with a maximum angle change of 6.06 degrees per sec. We can concluded that this maximum error is, just, for few frames and the control system has recovered in all the situations. In addition, looking in Fig. 7, it is possible to see that the controller has not oscillation, so we can say that the controller has a very good behavior solving those kind of movements.

Another kind of movements is when the helicopter's movements are not too hard, (movements between the intervals 1 to 2, 2 to 3 and 3 to 4, shown in Fig. 6). In that phases of the flight with soft movements, there is an error of ± 5 pixels in the Pitch axis of the platform and a +5, -15 pixels error in the yaw angle. Measuring it in degrees it is more or less a 0.6562 degrees for pitch and 0.6562, -1.96875 degrees for yaw, like we can see in the tab II. In these sections of soft movements, there are continued little yaw changes, as is shown in fig. 6(b) between section 1 and 2.

TABLE II
ERROR IN UAV SOFT FLIGHT MOVEMENTS SITUATIONS.

Platform Axis	error (pixels)	error (degrees)
Pitch	± 5	± 0.6265
Yaw	+5,-15	+0.6562,-1.96875

VII. CONCLUSIONS AND FUTURE WORKS

This paper presents an autonomous fuzzy control of pan and tilt camera platform on board on a UAV using a Lucas-Kanade tracker for static objects. The controller has been tested on real flights for tracking statics objects.

A. Conclusions

Based on the results presented on this papers is possible to conclude that the uses of this kind of platform onboard a

TABLE III
DATA FROM BIG ATTITUDE CHANGES SECTIONS OF THE FLIGHT.

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

UAV improve the behavior for track objects in comparison with other UAV system without it. The improvements are a faster response and a bigger freedom of movements for the UAV, making possible to track objects in different modes of operations, like autonomous flight following a pre-programmed series of way points, under manual commands and staying in hovering position in a safe place. Also the two Fuzzy controllers (one for each axis of the platform) have an excellent behavior tracking static objects.

In addition, the developed fuzzy learning algorithm explained in [12] and [13], can give us the possibility to adapt different rules-base for different object behaviors, improving the fuzzy visual tracker.

B. Future Works

In order to optimize the Fuzzy controllers system, we must make new tests with moving objects, like cars, giving to this work a more useful character, to continue with taking the up-level control of the helicopter with another Fuzzy-MOFS controller that sends to the UAV commands of yaw, roll and pitch movements to increase the possibility to follow moving objects for a long time.

One of the other updates that we are doing right now, is to create a testbed for different vision tracker algorithms with the Fuzzy-MOFS controller.

The developed visual tracking algorithms allows a lot of possible applications in situations, like staying near to electric structures, wind fields or dams inspection, and fire monitoring among many others.

VIII. ACKNOWLEDGMENTS

The work reported in this paper is the product of several research stages at the Computer Vision Group from the Universidad Politécnica de Madrid. The authors would like to thank Jorge León for supporting the flight trials, the I.A. Institute - CSIC for collaborating in the flights' consecution and Alba Olivares Nadal for her collaboration as proofreader. This work has been sponsored by the Spanish Science and Technology Ministry under grants CICYT DPI2004-06624, CICYT DPI2000-1561-C02-02 and MICYT DPI2007-66156..

REFERENCES

- [1] A. Puri, K.P. Valavanis, and M Kontitsis. Statistical profile generation for traffic monitoring using real-time uav based video data. In *Control and Automation, 2007.MED '07. Mediterranean Conference on*, MED, pages 1–6, 2007.
- [2] I. K. Nikolos, N. C. Tsourveloudis, and K. P. Valavanis. A uav vision system for airborne surveillance. In *Robotics and Automation, 2004. Proceedings. ICRA '04. 2004 IEEE International Conference on*, pages 77–83, New Orleans, LA, USA, May 2004.
- [3] S. Rathinam, P. Almeida, ZuWhan Kim, S. Jackson, A. Tinka, W. Grossman, and R. Sengupta. Autonomous searching and tracking of a river using an uav. *American Control Conference, 2007. ACC '07*, pages 359–364, July 2007.
- [4] D.K. MacArthur and C.D. Crane. Unmanned ground vehicle state estimation using an unmanned air vehicle. *Computational Intelligence in Robotics and Automation, 2007. CIRA 2007. International Symposium on*, pages 473–478, June 2007.
- [5] S. Hrabar. 3d path planning and stereo-based obstacle avoidance for rotorcraft uavs. *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pages 807–814, Sept. 2008.
- [6] Hairong Zou, Zhenbang Gong, Shaorong Xie, and Wei Ding. A pan-tilt camera control system of uav visual tracking based on biomimetic eye. *Robotics and Biomimetics, 2006. ROBIO '06. IEEE International Conference on*, pages 1477–1482, Dec. 2006.
- [7] V.N. Dobrokhodov, I.I. Kaminer, K.D. Jones, and R. Ghabcheloo. Vision-based tracking and motion estimation for moving targets using small uavs. *American Control Conference, 2006*, pages 6 pp.–, June 2006.
- [8] V.K. Chitrakaran, D.M. Dawson, H. Kannan, and M. Feemster. Vision assisted autonomous path following for unmanned aerial vehicles. *Decision and Control, 2006 45th IEEE Conference on*, pages 63–68, Dec. 2006.
- [9] Pascual Campoy, Juan Correa, Ivan Mondragon, Carol Martinez, Miguel Olivares, Luis Mejias, and Jorge Artieda. Computer vision onboard uavs for civilian tasks. *Journal of Intelligent and Robotic Systems*, 2008.
- [10] C. G. Harris and M. Stephens. A combined corner and edge detection. In *Proceedings of the 4th Alvey Vision Conference*, pages 147–151, 1988.
- [11] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proc. of the 7th IJCAI*, pages 674–679, Vancouver, Canada, 1981.
- [12] Miguel Olivares and J.A.F. Madrigal. Fuzzy logic user adaptive navigation control system for mobile robots in unknown environments. *Intelligent Signal Processing, 2007. WISP 2007. IEEE International Symposium on*, pages 1–6, Oct. 2007.
- [13] Miguel. Olivares, Pascual Campoy, , Juan Correa, Carol Martinez, and Ivan Mondragon. Fuzzy control system navigation using priority areas. In *Proceedings of the 8th International FLINS Conference*, pages 987–996, Madrid, Spain, September 2008.