Visual Model Feature Tracking For UAV Control

Iván Fernando Mondragón*, Pascual Campoy*, Juan Fernando Correa*, Luis Mejias**
*Universidad Politécnica de Madrid, Spain
{imondragon,campoy,jfc}@etsii.upm.es
**ARCAA-Queensland University of Technology, Australia
luis.mejiasalvarez@qut.edu.au.

***Abstract* – This paper explores the possibilities to use robust object tracking algorithms based on visual model features as generator of visual references for UAV control. A Scale Invariant Feature Transform (SIFT) algorithm is used for detecting the salient points at every processed image, then a projective transformation for evaluating the visual references is obtained using a version of the RANSAC algorithm, in which a series of matched key-points pairs that fulfill the transformation equations are selected, rejecting otherwise the corrupted data. The system has been tested using diverse image sequences showing its capability to track objects significantly changed in scale, position, rotation, generating at the same time velocity references to the UAV flight controller. The robustness our approach has also been validated using images taken from real flights showing noise and lighting distortions. The results presented are promising in order to be used as reference generator for the control system.**

***Keywords* – Unmanned Aerial Vehicle, feature tracking, autonomous helicopter, SIFT, RANSAC.**

I. Introduction

Our work is focused on the integration of different visual feature detection and tracking algorithms in UAVs. The ultimate goal is to extend the UAVs capabilities through the use of visual sensors with the aim to be used in tasks like object recognition and tracking, visual inspection and visual navigation. The techniques proposed are intended to control in real-time the UAV displacement based on image velocity references. Using previous works developed by the authors as foundation. We extend these approaches based on appearance with techniques based in visual models. These techniques are evaluated in quality, efficiency and the capacity to be implemented in real time for control process.

We implement visual control techniques in UAVs using the first generation testbed developed at Universidad Politécnica de Madrid, COLIBRI I [1]. This platform has a control architecture that permits the integration of many different visual algorithms in the control process. The vision-based system acts as an overall controller sending navigation commands to a low level flight controller which is responsible for autonomous control of the helicopter.

The paper is organized as follows, in the next section we briefly discusses the related work. Section III describes the platform COLIBRI I, used as the main testbed platform. In section IV we show the approach used to control the helicopter based on visual references using a salient point tracker. Section V shows the experimental results. Finally conclusions and future work are drawn in section VI.

II. Related Work

Autonomous aerial vehicles have been an active area of research for several years. Autonomous helicopters have been used as testbeds to investigate problems ranging from control, navigation, path planning to object detection and tracking, visual navigation, etc. Several teams from MIT, Stanford, Berkeley and USC have had an ongoing AFV project for the past decade. The reader is referred to [2] for a good overview of the various types of vehicles and algorithms used for their control. Recent work has included autonomous landing [3], [4] and aggressive maneuvering [5]

Many techniques for detection or tracking of interests objects in the scene are based on model features or descriptors. In the literature there are many feature detectors based on salient point, shape, Differential Invariants, SIFT, etc. The suitability of a feature detector is closely related with the application or task intended to perform. In the work of Mikolajczyk and Schmid [6], they made a comparison of many different descriptors, based in a matching and recognition context and under a variety of viewing conditions, finding that better performance and robustness for affine transformations, scale changes, image rotation, blurring and illumination changes are present in the SIFT descriptors [7].

Some applications of matching using SIFT were proposed by Se and Lowe [8], and have been tested in ground robots with very good results for navigation, 3D reconstruction and SLAM. SIFT also has been used in UAVs to find landmarks based on infrared images. The aim of this SLAM works is to implement landmark

Fig. 1. UPM-COLIBRI I. HELICOPTER PLATFORM USED AS MAIN RESEARCH PLATFORM

recognition to be used for UAV navigation [9]. A similar work was done by Adrien [10] for M.A.V. in which a combination of Harris [11] corner detector and SIFT for 2D localization is used. In all these approaches the visual system is used for landmarks detection and map building, but it is not directly integrated as a reference for the flight control.

III. THE AUTONOMOUS HELICOPTER TESTBED, COLIBRI I

The COLIBRI I [12] testbed (figure 1), is based on a gas powered industrial twin helicopter with a two stroke engine 52 cc and 8 hp. The platform is fitted with a xscale-based flight computer augmented with sensors (GPS, IMU, Magnetometer, etc fused with a Kalman filter for state estimation). For vision processing it has a VIA mini-ITX 1.25 GHz onboard computer with 512 Mb Ram, wireless interface and a videre STH stereo head for acquiring the images. Both Computers run Linux OS. The ground station is a laptop used to send high-level control commands to the helicopter. It is also used for visualization of image data and communication with the onboard image processing algorithm. Communication with the ground station is via 802.11g wireless Ethernet protocol.

The system runs in an client-server architecture using TCP/UDP messages. This architecture allows embedded application to run onboard the autonomous helicopter while interact with external processess 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. This layer is based on a communication API where all the messages and data types are defined. The helicopter low-level controller is based on simple PID control loops and ensures the stability of the helicopter. This controller has been validated empirically. The higher level controller uses various sensing modalities such as GPS and/or vision to perform tasks such as navigation, landing, visual tracking, etc.

IV. SALIENT POINTS TRACKING

SIFT (Scale Invariant Feature Transform) developed by Lowe [7] is used to detect stable features in an object template. The template is initially selected by the user in the video sequence. The object is matched along the video sequence comparing the model template and the image SIFT descriptor using the nearest neighbor method. Given the high dimensionality of our descriptor (128), its matching performance is improved using the Kd-tree search algorithm with the Best Bin First search modification proposed by Lowe. Once the matching is performed, a perspective transformation is calculated using the matched Keypoints, then the RANSAC algorithm [13] is applied to obtain the best possible transformation taking into consideration bad correspondences. This transformation includes the parameters for translation, rotation and scaling of the interest object, and is defined in equations (1),(2).

$$\bar{X}_p = H\bar{X} \tag{1}$$

$$\begin{pmatrix} x_p \\ y_p \\ \lambda \end{pmatrix} = \begin{pmatrix} a & b & c \\ d & e & f \\ g & h & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \tag{2}$$

where:
$(x, y, 1)^T$: Homographic coordinates of the Keypoint $(x, y)^T$ at the model image.
$(x_p,\, y_p,\, \lambda)$: Homographic coordinates of the Keypoint $(x^+, y^+)^T$, in the current image, corresponding to the matched Keypoint of $(x, y)^T$ in the model image.
From this we can find that:

$$\begin{pmatrix} x^+ \\ y^+ \end{pmatrix} = \begin{pmatrix} \frac{x_p}{\lambda} \\ \frac{y_p}{\lambda} \end{pmatrix} \tag{3}$$

Solving equations (2) and (3):

$$x^+ = \frac{ax + by + c}{gx + hy + 1} \qquad y^+ = \frac{dx + ey + f}{gx + hy + 1} \tag{4}$$

According with equation (4), to obtain the Matrix H, we need to calculate eight parameters. Considering that every pair of matched keypoints give us two equations, we need a minimum of four pairs of correctly matched keypoints to solve the system. Equation (5) shows the equation systems to be calculated. The solution is obtained using Singular Value Decomposition.

$$\begin{pmatrix} x & y & 1 & 0 & 0 & 0 & -xx^+ & -yx^+ \\ 0 & 0 & 0 & x & y & 1 & -xy^+ & -yy^+ \\ & & & & & \cdots\cdots & & \\ & & & & & \cdots\cdots & & \end{pmatrix} \begin{pmatrix} a \\ b \\ c \\ d \\ e \\ f \\ g \\ h \end{pmatrix} = \begin{pmatrix} x^+ \\ y^+ \\ . \\ . \\ . \\ . \\ . \end{pmatrix} \tag{5}$$

As mentioned before, not all pair of matched keypoints corresponds correctly. For this reason a method to discard the corrupted data before solve equation (5) is used. The RANSAC algorithm is evaluated for this purpose. Its aim is to obtain the pairs of Keypoints that have the best projection (defined as inliers points). 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. If the total trials are reached, a good solution for (5) can not be obtained.

Once the detection is performed in the current frame and the transformation has been resolved, the velocity reference can be generated using the center of gravity of the tracked object. The center of gravity is used often when is desired to visually align the vehicle with the object. Following the integration scheme is described.

A. Integration of Image-Based References in the Flight Control

The output of the detection and tracking algorithm can be integrated in the flight controller using velocity references. The algorithm should be able to generate suitable image-based velocity references that will be integrated with the controller through a high level layer that switch and routes messages between processes. Different processes (e.g. flight control, vision algorithm, ground based commands, etc) can interact simultaneously using this layer and relying on protocols like TCP and UDP

Three velocity commands are currently available to control the displacement of an aerial platform, v_x, v_y, v_z for longitudinal, lateral and vertical displacements, respectively. A complete formal description of the velocity commands and camera configurations is made in Mejias et. al [14]. A comprehensive description of the vision-control integration using the high level layer is made in [15], for related work using this approach please refere to derived publications.

To derive suitable references from image measurement we assume a fixed kinematic relationship between the camera and the helicopter. In this way, and without loss of generality the camera velocity and orientation can be approximated to the helicopter velocity and orientation in bodyframe.

When the vision algorithm perform object tracking, the velocity of the object in the image plane can be obtained and is denoted by $(\dot{x}_p, \dot{y}_p)$. If we refer to classical image-based visual servoing (IBVS) techniques [16], the linear and angular velocities of the camera are related with the tracked object by:

$$\begin{bmatrix} \dot{x}_p \\ \dot{y}_p \end{bmatrix} = L \begin{bmatrix} V_c \\ \omega_c \end{bmatrix} = [L_v L_\omega] \begin{bmatrix} V_c \\ \omega_c \end{bmatrix} \tag{6}$$

where L is the interaction matrix which has two component for linear and angular velocity. The above model take into consideration the linear and angular velocities and is applicable in most cases where is desired to control 6 d.o.f. This model present non-linearities in the interaction matrix and depends on the unknown *feature depth*, that cannot be measured directly using monocular images. This represent a classical problem in IBVS the estimation or approximation of the Image Jacobian [17][18]. Therefore, in practice is useful to linearize this model and use an approximation of this matrix L^+. We have used previously an approximation of this matrix to control the lateral, vertical and longitudinal displacement of an autonomous helicopter using visual references [14]. Once the object translation and rotation have been resolved in equation 5, this result can be used to control the helicopter solving equation 6.

B. Implementation

This algorithm has been implemented in C language programming and combined with the Open Source Computer Vision libraries (OpenCV). Our approach is aided by the SIFT implementation developed by Hess [19]. Our algorithm is able to process online input sequences from either USB or firewire sources, or process offline images sequence from hard disk.

The process is initiated by selecting in the first image an interest area or zone around the object that is intended to track. This represent the template in which the SIFT Keypoints is performed obtaining the set of "keypoints". This set of points are stored for successive matching along the video sequence. Along with the first frame a second frame is acquired with a similar area but twice bigger. This second area is centered taking into consideration the projected center from the first frame. This area will be the local processing area in order to improve the speed of the algorithm.

For each new image, a new set of SIFT parameters are calculated and matched with the initial template. The matching process is followed by the RANSAC algorithm to fit the data to a perspective projection model. The RANSAC algorithm gives us two kind of answers:

- A Projective Matrix cannot be obtained: In this case, the search area is incremented, and a new image is processed. This is repeated until an object that corresponds with the original frame is found.

- If the matrix is found: It calculates the original frame contour projection in the current image and shows it. The search area is also centered to the current position of the projected center of the object and the algorithm close the loop.

Figure 2 shows a pseudocode of this algorithm.

V. ALGORITHM VALIDATION

In this section we present several experimental trials with the aim to validate our approach. First, the algorithm is tested with some sequences, in which movements to planar objects

```
-Define Image Template by selecting a window in the
image.
-Obtain SIFT Keypoints for Image Template

For (Every new acquire image)
    -Redefine Image Window processed based in last
    template position and size.
    -Extract SIFT Keypoint Features from Image Window.
    -Obtain a Match table of current image and template.
    Mt= maximum tolerance of reprojection error.
    Trials = Number of trials for RANSAC obtained from
    the Size of Match Table
    Concensus = Desired Number of inliers based on Size
    of Match Table

    While ( Consesus Get = FALSE, AND , Trial < Maximun
    Number of Trials)
      Randomly select four pairs from the Match Table.
      Calculate a Homography H with this keypoints
      For (every pair on the Match Table)
        d= distance based on the reprojection error using
        H
        if (d < Mt)
            inlier = inlier+1
        End if
      End for
      If ( number of inlier > Consensus)
        Consensus Get=True
      Else
        Consensus Get =False
      End if
        Trials= Trials +1
      End While

      If (consensus Get = True)
        -Calculated Final H using only inliers Keypoints
        -Obtain projected template and window coordinates
        -Define visual error based in image coordinates
        -Calculate velocity commands
        -Send commands to the flight controller
      Else
        -Double window area
      End if
End for
```

Fig. 2. ALGORITHM PSEUDOCODE

TABLE I.
TEST RESULTS

seq	window width	window height	average SIFT	average matched keypoints	correct projection (%)	average frame rate (s)
1	324	450	83.46	17.76	70	0.63
2	510	382	331.28	70.66	82.5	1.07
3	596	480	617.09	67	72.3	1.93
4	418	192	184.11	9.61	60.1	0.95
5	270	186	389.49	14.71	55.4	1.88
6	328	250	432.01	32.26	78.5	1.66

are applied including translation, scale and rotation in three axes with a constant illumination. The images acquired at 30 fps in full color have a resolution of 640x480 pixels and every sequence has 1000 frames.

Figure 3 shows the objects used in the tests sequences. The algorithm was tested with the sequences evaluating the robustness and efficiency in terms of the number of correctly matched Keypoints, projected frames and average time spent in the process. Table I shows the summarized results.

From table I is clear that the size of the search window has a big influence in the speed of the algorithm but it does not always

Fig. 3. TEMPLATES IMAGES AND OBJECTS USED DURING THE EXPERIMENTS

yield to a better result, because big areas have a lot of Keypoints and sometimes they cause that the matching process obtain a big number of bad matched keypoints (outliers). Also, the RANSAC algorithm spends more time to reach the projection or the maximum number of trials caused by more comparison between Keypoints and model under Consensus evaluation at every cycle. A good performance is obtained when the number of matched keypoints is low, showing that the RANSAC algorithm has more time variability to spend more part of the time employed by the algorithm to obtain the solution. The implemented algorithm works well when the object has large variations in form and intensity, but has some reduction in performance when the object does not have a differentiated structure or when it has a planar texture like the case of the voltmeter or the chessboard (Figure 3(4), 3(5)). The experimental trial performed shows that the algorithm can match and obtain an adjusted projection when the object has changes in scale by a factor of 2X and rotation up to 45 degrees in all axes. Figure 4 shows some examples of these conditions. In theses images the original frame is in the upper left of the image (without change the scale) and the matched Keypoints between original frame and the current image are connected by the different color lines. The obtained projection of the original frame is shown as the white box and the black box is the area processed in current scene (window processed).

A final sequence of images taken during a real flight test of the COLIBRI I UAV is used to test the tracking of a defined window in a building. These images are in gray scale at 640x480 and contain a large influence of vibrations, noise and motion

Fig. 4. EXPERIMENTS SHOWING CHANGES IN SCALE AND ROTATION OF OBJECTS

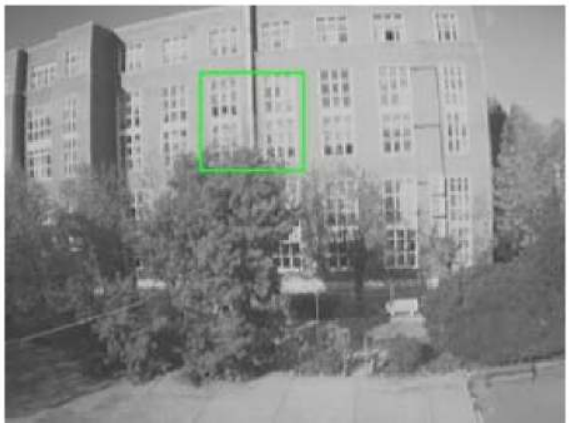

Fig. 5. IMAGE SEQUENCE FROM REAL FLIGHT TRIAL WITH A SELECTED WINDOW TO TRACK

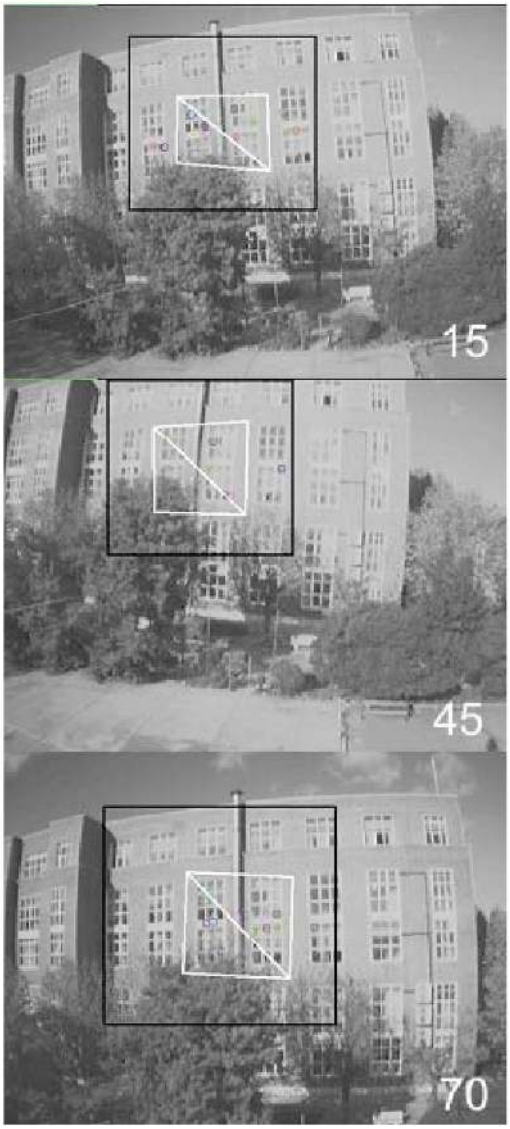

Fig. 6. MATCHED OBJECTS DURING EXPERIMENTS IN FRAMES 15, 45 AND 70

generated by the helicopter. Also, these images contain natural changes on illumination, and a significant quantity of rotational and translational movements. This sequence was used without a previous process or property enhancement as shows in figure 5.

Figure 6, shows the building windows tracked at frames 15, 45 and 70. The search window has 236x224 pixels. The mean SIFT keypoints detected by frame is 535.15 and the number of matched points is 11.90. The average time spend is 0.92 seconds with 59% of frames detected correctly. The noise and vibration in the sequence generated by the Helicopter and the changes in illumination influence the capability of the algorithm to find the object in some frames. In addition, the selected scene has a recurrent structure that is not a good to reach adequate matched keypoints and perspective transformation. However these preliminary results are promising, showing the capability of this algorithm to track objects in real flight image sequences.

Finally, a special attention needs to be taken on the computational time spent by the algorithm. This computational time is variable, and depends directly of the size of the window area processed, the number of Keypoints obtained, and the facility in which a transformation is found by the RANSAC Algorithm. In the worst case, the algorithm spends approximately two second to obtain the model or to reach the maximum number of trials. In these way improvements to this part of the algorithm has to be done before use it to real time detection.

VI. CONCLUSION AND FUTURE WORK

In this paper an implementation for object tracking based on model features has been presented. The tests using real images from an onboard UAV camera show that the algorithm works efficiently for tracking a selected object within long a video sequence. A model of the desired object to be tracked is obtained from a set of images and used to detect it using a comparison method based on salient Keypoints. The initial selection of the template to be tracked is essential to guarantee a good performance of the algorithm. The algorithm performs better when tracking objects presenting a large variation in texture and intensity than objects presenting homogenous and recurrent shapes, due to a more stable and descriptive feature calculation. The implemented algorithm can match an object corrupted with noise and vibration caused by the helicopter movement. Also the images can be used to track the object rotated up to 45 degrees, shifted and scaled up to 2X, and partially changed in illumination. Further efforts need to be done in reducing the dimensionality of the descriptors and improving the computational time spent comparing the descriptors, taking care that the new descriptor (with low dimension) continues representing correctly the Keypoints.

In this way new modifications of SIFT, like PCA-SIFT[20] and GLOTH[6], and similar descriptors as SURF[21] are currently under analysis. The algorithm also can be optimized by making changes in the Keypoitns comparison method, for another that reduces the probability of incorrect correspondences in the matching process or approaches that do not depend of the trial method. The RANSAC function proposed to fit the data to a specific model returns good results, eliminating the wrong matched points in the matrix computation. Since the function needs a variable number of trials and comparisons to reach the consensus, the time spent to get the transformation is too variable and it needs to be bounded for real time applications.

Additional improvements are being carried out by using a state estimator like Kalman Filter to center the search windows, reducing the area and therefore the number of descriptors that have to be compared. Also it will reduce the probability of incorrect matched points and therefore the number of trials to reach the consensus in RANSAC function.

ACKNOWLEDGMENT

This work is sponsored in part by the Spanish Science and Technology Ministry under a project grant CICYT DPI2004-06624, and by Universidad Politécnica de Madrid-CAM. The authors would like to thank to Jorge León for support with the flight trials, and the rest of the Computer Vision Group at Universidad Politécnica de Madrid.

REFERENCES

[1] COLIBRI, "Universidad Politécnica de Madrid. Computer Vision Group. COLIBRI Project," http://www.disam.upm.es/colibri, 2005.

[2] Luis Mejias, Srikanth Saripalli, Pascual Campoy, and Gaurav 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.

[3] Srikanth Saripalli, James F. Montgomery, and Gaurav 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.

[4] Torsten Merz, Simone Duranti, and Gianpaolo Conte, "Autonomous landing of an unmanned helicopter based on vision and inbertial sensing.," in *International Symposium on Experimental Robotics*, Singapore, June 2004.

[5] V. Gavrilets, *Autonomous Aerobatic Maneuvering of Miniature Helicopters: Modeling and Control*, PhD thesis, School of Aeronautics and Astronautics, June 2003.

[6] M. Mikolajczyk and C. Smid, "A performance evaluation of local descriptors," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 27, no. 10, pp. 1615–1630, 2005.

[7] David G. Lowe, "Distintive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[8] S. Se, D. Lowe, and J. Little, "Local and global localization of for mobile robots using visual landmarks," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Hawaii, October 2001, pp. 414–420.

[9] j. Nilsson, "Visual landmark Relection and Recognition for Autonomous Unmanned Aerial Vehicle Navigation," MSc thesis, School vehicle engineering. Royal Institute of Technology, 2005.

[10] A. Adrien, D. Filliat, S. Doncieux, and J. Meyer, "2d simultaneous localization and mapping for micro air vehicles," in *European Micro Air Vehicle Conference and Flight Competition*, 2006.

[11] C. G. Harris and M. Stephens, "A combined corner and edge detection," in *In Proceedings of the 4th Alvey Vision Conference*, 1988, pp. 147–151.

[12] Luis Mejías, Ivan Mondragón, Juan Fernando Correa, and Pascual Campoy, "Colibri: Vision-guided helicopter for surveillance and visual inspection," in *Video Proceedings of IEEE International Conference on Robotics and Automation*, Rome, Italy, April 2007.

[13] 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.

[14] Luis Mejias, Srikanth Saripalli, Pascual Campoy, and Gaurav Sukhatme, "Visual servoing of an autonomous helicopter in urban areas using feature tracking," *Journal Of Field Robotics*, vol. 23, no. 3-4, pp. 185–199, April 2006.

[15] Luis Mejias, *Control visual de un vehiculo aereo autonomo usando detección y seguimiento de características en espacios exteriores.*, PhD thesis, Escuela Técnica Superior de Ingenieros Industriales. Universidad Politécnica de Madrid, Spain, December 2006.

[16] F. Chaumette and S. Hutchinson, "Visual servo control, part ii: Advanced approaches," *IEEE Robotics and Automation Magazine*, vol. 14, no. 1, pp. 109–118, March 2007.

[17] A. C. Sanderson and L. E. Weiss, "Adaptative visual servo control of robots," *In Robot Vision (A. Pugh, ed)*, pp. 107–116, 1983.

[18] Lee Weiss, *Dynamic Visual Servo Control of Robots: An Adaptive Image-Based Approach*, PhD thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, April 1984.

[19] R. Hess, "Sift feature detector implementation in c.," http://www.web.engr.oregonstate.edu/~hess/index.html, Feb. 2007.

[20] Y. Ke and R. I. Sukthankar, "Pca-sift: A more distinctive representation for local image descriptors," in *Proceeedings of 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2004, vol. 2, pp. 506–513.

[21] Herbert Bay, Tinna Tuytelaars, and Luc Van Gool, "Surf: Speeded up robust features," in *Proceeedings of the ninth european conference on computer vision*, Graz, Austria, May 2006.