Quadcopter Control Using Onboard Monocular Camera for Enriching Remote Laboratory Facilities

F. Khattar, F. Dornaika, Franck Luthon, Benoît Larroque

Quadcopter Control Using Onboard Monocular Camera for Enriching Remote Laboratory Facilities

Khattar Fawzi
UNIV PAU & PAYS ADOUR/ E2S UPPA
LABORATOIRE D'INFORMATIQUE DE L'UNIVERSITE DE PAU ET DES PAYS DE L'ADOUR, EA3000
64600, Anglet, France
fawzi.khattar@univ-pau.fr

Dornaika Fadi
Computer Science Department
University of the Basque Country
San Sebastian, Spain
fadi.dornaika@ehu.eus

Luthon Franck
UNIV PAU & PAYS ADOUR/ E2S UPPA
LABORATOIRE D'INFORMATIQUE DE L'UNIVERSITE DE PAU ET DES PAYS DE L'ADOUR, EA3000
64600, Anglet, France
franck.luthon@univ-pau.fr

Larroque Benoit
UNIV PAU & PAYS ADOUR/ E2S UPPA
LABORATOIRE DES SCIENCES DE L'INGÉNIEUR APPLIQUÉES À LA MÉCANIQUE ET AU GÉNIE ELECTRIQUE - MIRA , EA4581
64600, Anglet, France
benoit.larroque@univ-pau.fr

***Abstract*—We present the implementation of a visual localization and control system of a low cost quadcopter for an application in a remote electronic laboratory. The issues addressed are: environment exploration in remote laboratories, autonomous visual inspection of planar objects, and autonomous homing and landing. The localization system is composed of two complementary visual approaches: (i) a visual SLAM (Simultaneous Localization And Mapping) system, and (ii) a homography-based localization system. We extend the application scenarios of the first system by allowing close range inspection of a planar electrical instrument and autonomous landing. Experiments conducted in a remote laboratory workspace are presented. They prove the performance of the proposed system in terms of real-time and robustness.**

I. Introduction

Remote labs constitute an interesting and novel way of doing labs. Anywhere and at anytime the student can access the lab equipment and do his labwork. This new way of distance learning can be used to increase the motivation of nowadays students, especially if it is coupled with new technologies like quadcopters [1].

Quadcopters equipped with a camera can be used in these laboratories in order to mimic the student behavior in traditional lab and increase motivation for learning. It can be an interesting way to make the lab experience immersive and ludic. Specifically, in remote electronics laboratories it can fly and move in 3D space to inspect electrical instruments, consequently sending direct visual feedback of the results of an experiment in a new interesting way to the student. To achieve this, and to make the process autonomous two systems are needed. First, a localization system that can estimate the position and orientation of the quadcopter in 3D space with respect to an object of interest (front panel of an electrical instrument in our case). Second, a control system that sends appropriate commands to the quadcopter in order to reach a desired relative or absolute 3D position.

Many localization systems and sensors can be used in order to estimate quadcopter position and orientation in 3D space. For outdoor environments, GPS sensors constitute the best solution to localize a quadcopter. For indoor environments, artificial markers can be placed in the scene to facilitate the task of localization [2]. These markers can also be reflective and detected by an external localization system that gives accurate position estimate. However, the challenge in these applications is to use only available on-board sensors. Different sensors can be used to gather information about the environment and localize the quadcopter. Stereo rig cameras [3] and RGBD cameras [4] have been investigated. Using stereo or RGBD cameras allows for absolute pose estimate however this comes with an additional weight and power consumption. In this work we used the Parrot AR Drone 2.0 [5], a low cost quadcopter equipped with two monocular cameras facing forwards and downwards, in addition to pressure, ultrasound and inertial sensors. Using the monocular cameras available on-board constitutes a good trade-off between weight and information recovery from the environment (3D localization, environment recognition, etc.). However, a monocular camera alone cannot give absolute scale pose estimate due to the well known scale ambiguity rising from the perspective projection

of 3D world into 2D images. Despite this fact, combining the visual information with some prior knowledge of 3D world or other sensors, that can give partial but absolute pose estimate, can overcome this issue and allow for absolute 3D pose estimate. In [6] authors use the well known SLAM algorithm PTAM (Parallel tracking and mapping) [7] and combine it with inertial and ultrasound sensors readings in order to get the absolute 3D pose estimate. They fuse all the available information in a Kalman filter allowing for information fusion and delay compensation. In the present implementation, we build on their system and use it for 3D pose estimation when the quadcopter is exploring the 3D world. However, since the visual SLAM relies on corresponding points detected in the flow of images, it will drastically fail to give 3D pose estimate if the quadcopter is asked to inspect an object of interest, since these points will disappear when the object of interest occupies the majority of the image pixels. In this scenario, the quadcopter must move to a position near the object of interest which will cause the visual tracking of points to fail because there are not enough points available in the field of view of the camera. To overcome this limitation, this system is extended by using the object of interest as a landmark. In this way two localization modules are available: a visual SLAM module (needed for localization w.r.t. an arbitrary world coordinate system) and a localization module that relies on detecting and localizing a planar object with respect to the quadcopter and to the arbitrary world coordinate system (needed for controlling the remote lab activities and sending visual feedback to the student). The first module is suitable while exploring the environment whereas the latter is suitable when the quadrocopter is in a short distance range from the object of interest or when it needs to land on a landing zone. The paper is organised as follows: In the second section we present an overview of the remote electronic lab LaboREM. In section 3 we explain how detection and 3D localization can be carried out for a given planar object of interest. Based on the planar assumption the homography transform is estimated in order to estimate the quadcopter-object relative pose. It also presents the 3D pose visual servoing of the quadcopter. In the final section, we present some experiments and results.

II. Review of the remote electronics lab

LaboREM is a remote laboratory in electronics developed for first year undergraduate students in engineering. The learning objective of LaboREM is to enable students to wire and test remotely electronic circuits, make measurements and characterize each circuit by its time or frequency response. The electronic circuits consist of operational amplifiers, active filters and oscillators. Its design is based on a classic client-server architecture [8]. The student calls for a lab session by simple URL addressing. A first-in first-out strategy is adopted to give access to the remote lab to one client (student) at a time. A 5 minute connection is allotted to each student for one short manipulation. If he wants to repeat, he has to queue again. The remote lab application is developed using NI-LabVIEW software and the easy-to-use RFP protocol to pilot

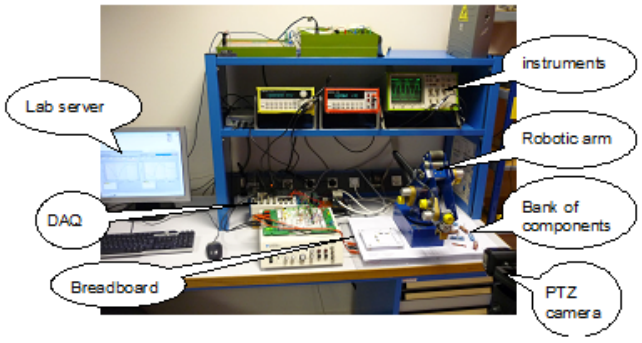

Fig. 1. Main components of the remote lab LaboREM.

the remote devices. The hardware setup is shown in Figure 1. It includes: (i) a robotic arm that mimics the student's hand for placing electronic components equipped with magnets on an electronic breadboard, (ii) measurement instruments and data acquisition system (DAQ), (iii) a webcam with zoom control that mimics the student's eye in order that the student doesn't feel so far away from what is actually happening in the lab, (iv) a quadcopter (AR quadcopter 2.0) with the role of flying in the lab for exploring the environment and inspecting electrical instruments in order to increase student immersion and motivation. The pedagogical scenario for the lab activity is based on a game-like approach: it includes a treasure hunt and a Top10 of the best measurements. Technically, the game scenario is built on four basic concepts like in e-games: time spent, score or mark obtained for an activity, level (beginner, medium, advanced) and number of lives or repetitions allowed.

III. Proposed implementation

A. Quadcopter-Object relative 3D pose : A real-time and marker free solution

Planar objects are a well defined type of objects that are widely available in human made environments. Incorporating the information that the object of interest is planar is of great benefit for camera object pose estimation. Suppose we want to determine the 3D rigid transformation from the planar object coordinate system to the camera coordinate system by using the projection of that object into the image. Suppose the coordinate system attached to the planar object is chosen in a way that the object lies in its XY plane. Any point $(X_i, Y_i, 0, 1)$ on the object will thus have zero Z coordinate. Using the pinhole camera model, a point on the object $(X_i, Y_i, 0, 1)$ will be projected to an image point p_i as follows:

$$p_i = s[u_i, v_i, 1]^T = KT \times \begin{pmatrix} X_i \\ Y_i \\ 0 \\ 1 \end{pmatrix}$$

$$= \begin{pmatrix} \alpha_u & 0 & u_c \\ 0 & \alpha_v & v_c \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{pmatrix} \begin{pmatrix} X_i \\ Y_i \\ 0 \\ 1 \end{pmatrix}$$

$$= \begin{pmatrix} \alpha_u & 0 & u_c \\ 0 & \alpha_v & v_c \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} r_{11} & r_{12} & t_x \\ r_{21} & r_{22} & t_y \\ r_{31} & r_{32} & t_z \end{pmatrix} \begin{pmatrix} X_i \\ Y_i \\ 1 \end{pmatrix}$$

$$= \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix} \begin{pmatrix} X_i \\ Y_i \\ 1 \end{pmatrix}$$

α_u, α_v are the focal length in image pixel coordinate system and u_c, v_c are the coordinate of the principal point in the same coordinate system. r_{ij} and t_i denotes the elements of the rotation transformation and the translation transformation respectively. s denotes the scale parameter. The 3 by 3 matrix H is called homography and it encapsulates the intrinsic and extrinsic parameters. Once this matrix is estimated, the aim is to extract the extrinsic parameters and use them for controlling the quadcopter using the paradigm of 3D pose-based servoing. However, due to scale ambiguity arising from perspective projection, there is not one single H matrix that can map the 3D points to the 2D points but a 1D vectorial space of 3 by 3 matrices $H_\lambda = \lambda\, K\, T$. The ambiguity can be eliminated by forcing the rotation matrix to be orthonormal and by having prior information about the dimensions of the object. From this homography matrix, one can directly calculate the rotation and translation matrix as follows [9]:

$$\begin{pmatrix} r_{11} \\ r_{21} \\ r_{31} \end{pmatrix} = \frac{1}{\lambda} K^{-1} \begin{pmatrix} h_{11} \\ h_{21} \\ h_{31} \end{pmatrix}$$

$$\begin{pmatrix} r_{12} \\ r_{22} \\ r_{32} \end{pmatrix} = \frac{1}{\lambda} K^{-1} \begin{pmatrix} h_{12} \\ h_{22} \\ h_{32} \end{pmatrix}$$

$$\begin{pmatrix} r_{13} \\ r_{23} \\ r_{33} \end{pmatrix} = \begin{pmatrix} 0 & -r_{31} & r_{21} \\ r_{31} & 0 & -r_{11} \\ -r_{21} & r_{11} & 0 \end{pmatrix} \begin{pmatrix} r_{12} \\ r_{22} \\ r_{32} \end{pmatrix}$$

$$\begin{pmatrix} t_x \\ t_y \\ t_z \end{pmatrix} = \frac{1}{\lambda} K^{-1} \begin{pmatrix} h_{13} \\ h_{23} \\ h_{33} \end{pmatrix}$$

$$\lambda = \left\| K^{-1} \begin{pmatrix} h_{11} \\ h_{21} \\ h_{31} \end{pmatrix} \right\|$$

B. Estimating the homography matrix

In order to estimate the homography matrix that maps any plane into another plane by means of perspective projections several methods can be used. These methods are usually classified into local (feature-based) and global (featureless) methods.

Given a template image of the planar object, local methods extract local keypoints and attribute a descriptor to each of them both in the template image and the current image. After this step, keypoints (at least four keypoints) in both images are matched according to a similarity metric performed on the descriptors. Given the point correspondences, the homography matrix is estimated using robust methods like RANSAC (Random Sample Consensus) in order to deal with the presence of outlier correspondences. Local methods can work well with no prior information on the homography parameters. However, in some cases the robust computation may be computationally expensive and do not work in real time. Different descriptors exist like SIFT [10], SURF [11], ORB [12], BRIEF [13] and many others. A survey about keypoint detectors and descriptors can be found in [14].

On the other hand, the global methods use all the information in the image and attempt to find the homography matrix that best aligns the template patch to the test image. This process however gives rise to non linear minimization problems that can be solved using iterative algorithms like gradient descent or Levenberg-Marquardt. Thus, a good initialization is necessary to guarantee the convergence of those algorithms. Different similarity functions exist to measure the degree to which two patches are aligned, the most used ones being the sum of squared distance (SSD) and the enhanced correlation coefficient (ECC) [15]. In practice the first one uses a brightness model in order to cope with variation of additive and multiplicative change of illumination [16] whereas the latter is by definition insensitive to those illumination changes. These methods have the advantage that they can run in real time and give good results if a rough estimate of the homography parameters is known. Thus the two families of methods are complementary: The first one is robust with no priors needed but computationally expensive, while the second is fast and works well if a prior is available.

Here, both approaches are used in order to estimate the 3D pose of the quadcopter with respect to the object of interest. The first approach is used for detecting the object of interest as well as for recovering from a tracking loss. The second is used in the tracking process. The approach is divided into two steps: detection and tracking. In the detection step, template matching on a pyramid of the image is used to search for the desired object. Once the detection is done, a homography transformation is computed by using the bounding box of the detected object to determine the quadcopter-object relative pose. A command is then sent to the drone in order to move it closer to the object. Template matching is used to allow successful detection of the object despite its distance from the camera and its size, as keypoints detector (SIFT)

fails to detect and put in correspondences keypoints if the object of interest doesn't occupy a certain amount of image pixels. However once the quadcopter's camera is close to the object we use the SIFT descriptor to allow more robustness to orientation changes. Once the object is detected, the tracking stage begins. As a rough estimation of the homography matrix is available from the detection stage, it is used as an initial solution for the next frame and the ECC algorithm is applied to estimate the homography in this frame. The homography estimation is propagated in this way from a frame to the next one, and used as a prior for the ECC algorithm. However, sometimes the ECC algorithm will fail to converge due to several reasons. For example, communication problems between the quadcopter and the computer makes the last estimated homography not close enough to the real solution of the current frame, which prevents algorithm convergence. Besides, the image quality can be degraded by motion blur or decoding/encoding problems. In this work, tracking loss is declared if the ECC algorithm is unable to converge or if it converges to a clearly unrealistic estimation. At each frame, we compute the 3D pose of the quadcopter with respect to the planar object. By monitoring the differences in the position of the quadcopter along time, we can detect a loss of tracking by setting a threshold on the difference of two consecutive pose estimates (position and orientation) which works well in practice. For example if the distance traveled in 3D space during two consecutive frames is larger than 30 cm, it is clear that the tracking is not correct and the pose is erroneous. In the latter case, we resort back to the local method (SIFT) if the quadcopter-object distance is relatively small, or to the template matching method in the other case, to reinitialize the ECC tracker as shown in Figure 2. This pose estimate is fused with inertial measurements sent by the drone in a Kalman filter framework in order to smooth this estimate, and to provide robustness when the visual tracker fails. The Kalman filter is used also to compensate for time delays as done in [6]. The homography estimation process is shown in Figure 2.

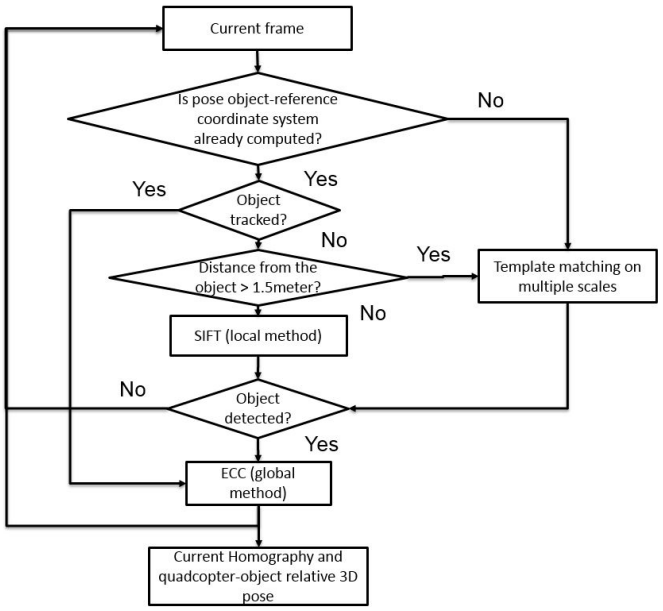

Fig. 2. Diagram showing the homography estimation process for each input video frame. The object is declared detected if the normalized correlation is greater than 0.8 in the case of template matching and if the Number of corresponding keypoints is greater than 8 in the case of SIFT. The object is declared tracked correctly if the estimated distance travelled between two consecutives frames is less than 30 cm or difference of yaw, roll or pitch angle is less than 15 degrees.

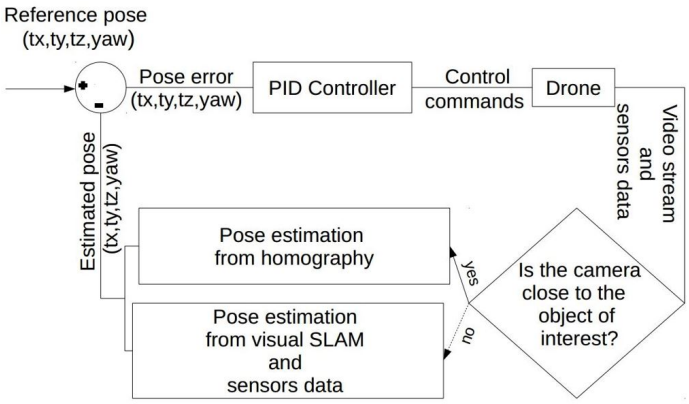

Fig. 3. Control loop using either the visual SLAM when exploring the environment or the homography estimation algorithm when inspecting a planar object at a short distance.

C. Visual control of the quadcopter

In order to control the 3D position and orientation of the quadcopter, a feedback control loop is used with two complementary sensors. One of the sensors is used for feedback at a time. The approach in [6] is used, taking as input video stream, ultrasound and inertial measurements when the quadcopter is exploring the environment far from the object of interest and when the quadcopter needs to return to the base station. The homography algorithm fused with inertial measurements is used when the quadcopter is inspecting the object of interest at a close distance. In this way, lost of localization is avoided when the quadcopter is approaching the instrument which allows for autonomous object inspection and environment exploration. The pose derived from either source is used in a closed servoing loop. The control loop is shown in Figure 3. The controlled degrees of freedom associated with the quadcopter are the 3D translation and the yaw angle. Each degree is controlled by a closed loop control system with a traditional PID controller. The other two rotational degrees of freedom (pitch and roll) cannot be used since they do not allow for position holding, they are used by the on-board embedded system on the AR Drone 2.0 to allow the quadcopter to move in different directions.

IV. Experiments

In order to evaluate the proposed implementation of 3D pose estimation and 3D pose-based servoing, we design three different scenarios. These scenarios are the following: behavior of the system in response to perturbations when asked to

inspect an object, autonomous visual inspection of planar object, and autonomous landing. In all these scenarios, the controlled degrees of freedom associated with the quadcopter are the 3D translation and the yaw angle.

a) First scenario: The first experiment is done to test the quality of the homography based visual feedback control system. To this end, we control the quadcopter in such a way that the desired 3D pose of its on-board camera is fronto-parallel to the planar object with a translation vector allowing a centered view. The pose used for the visual feedback control is the homography based pose. Since the servoing objective is to maintain a rigid link between the quadcopter and the object of interest, any motion induced to the object will force the quadcopter to compensate for it. We can induce such motion by a walking person that carries the object or by giving manual kicks to the quadcopter. The quadcopter then follows the object, centering it in the image as show in Figure 5. Two videos of this scenario are available online [17], [18].

b) Second scenario: In remote lab context, an interesting scenario is the following: the remote student will send a command to the quadcopter to go and inspect an electrical device. After receiving this command, the server tells the quadcopter to carry out the following tasks: it should first take off, initialize the SLAM algorithm, and initiate a search procedure for the required instrument. In a general case, to be able to search for the instrument a path planning and search algorithm must be used. However, here this is simplified by considering that the electrical instrument is already in the camera field of view and hence only detection and servoing are required. The instrument is searched for in the image by using template matching applied on the pyramid of the image to cope with scale change. Once the object is detected, the homography from the 3D world plane of the electrical instrument to the image plane is estimated. Based on the estimated homography, the 3D pose is estimated. We have now a 3D rigid transformation between coordinate systems of the instrument and the camera. Since the 3D pose of the quadcopter with respect to the visual SLAM coordinate system is known, the 3D pose of the planar object in that coordinate system can be calculated by cascading multiple rigid transforms between coordinate systems as shown in Figure 4. Thus, two localization sources are now available for visual feedback control. Based on the pose that is provided, it is possible to control the quadcopter through visual servoing in order to have a rigid link between the quadcopter and the instrument. To this end, at any time one needs the current localization information for the feedback control loops of the quadcopter. We emphasize the fact that the two sources of localization cannot be both available for all configurations and for all actual poses of the quadcopter. Indeed, the visual SLAM works well when there is enough keypoints in the image to detect and put in correspondence with 3D map points. However, as the quadcopter approaches the planar instrument, most of the keypoints will disappear and the visual SLAM algorithm might loose tracking. In the latter case, we use the 3D pose estimate given by the homography algorithm. In this way the quadcopter is able to fly and inspect an electrical instrument as shown in figure 6.

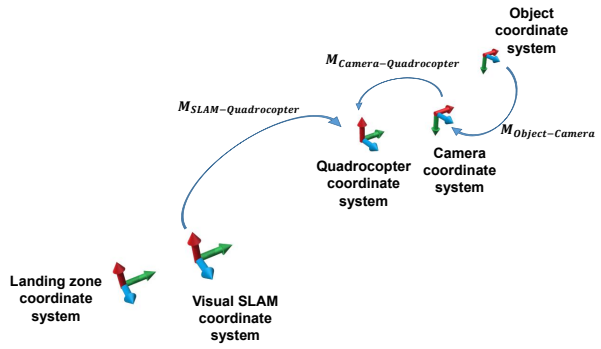

Fig. 4. Different coordinate systems and rigid transformations used in the application.

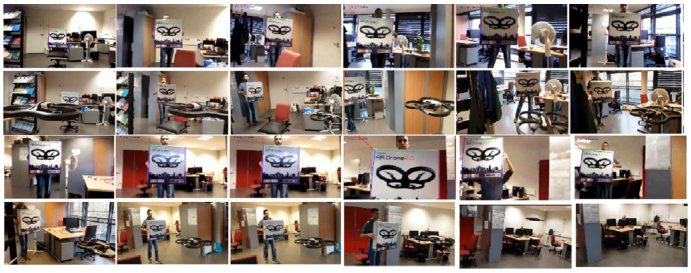

Fig. 5. Quadcopter and external views of the realization of the object tracking scenario.

c) Third scenario: Another scenario is getting the quadcopter home. The same principle is used, by relying on the homography algorithm to fly away from the electrical instrument and then switching to the visual SLAM to go back to the start point. Once being there, the bottom camera of the quadcopter and the homography algorithm are used to detect a landing zone. By doing that, the landing maneouvre can be acheived in an accurate manner. The ECC and the SIFT based homography estimation are both robust to partial occlusion as shown in figure 7. A video showing the hovering and landing maneouvre is available online [19].

The image-based localization and control work in real-time at an average of 30 FPS in different scenarios.

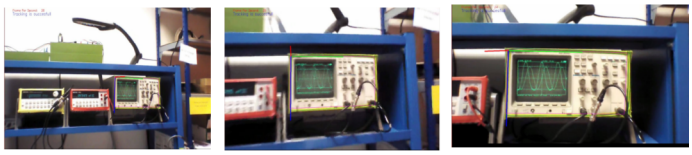

Fig. 6. Localization based on an electrical instrument (oscilloscope): From left to right, images taken from the quadcopter video stream that show how the system can control the quadcopter to obtain a front view of the instrument.

Fig. 7. Visual servoing for homing. On the left: the quadcopter hovering above the landing template and performing a controlled landing. On the right: the view seen from the bottom camera of the quadcopter that shows robustness to occlusion (when the landing object is partially out of the field of view).

V. Conclusion

This paper presents the implementation of a visual servoing system of a quadcopter in a remote lab environment to increase student immersion in the lab and hence his motivation. The localization system is based on two complementary algorithms that work in real-time at video rate in different scenarios. Qualitative results are presented demonstrating the ability to use a low cost quadcopter to inspect an object of interest and return to its base autonomously using only its on-board sensors. For future work, we intend to allow remote student-teacher interaction by using visual servoing for quadcopter control in order to follow and track the teacher's face.

References

[1] Franck Luthon, Benoît Larroque, Fawzi Khattar, and Fadi Dornaika. Use of gaming and computer vision to drive student motivation in remote learning lab activities. In *ICERI 2017: 10th annual International Conference of Education, Research and Innovation*, pages 2320–2329, 2017.

[2] Daniel Eberli, Davide Scaramuzza, Stephan Weiss, and Roland Siegwart. Vision based position control for MAVs using one single circular landmark. *Journal of Intelligent & Robotic Systems*, 61(1-4):495–512, 2011.

[3] Konstantin Schauwecker and Andreas Zell. On-board dual-stereo-vision for the navigation of an autonomous MAV. *Journal of Intelligent & Robotic Systems*, 74(1-2):1–16, 2014.

[4] Gerardo Flores, Shuting Zhou, Rogelio Lozano, and Pedro Castillo. A vision and GPS-based real-time trajectory planning for a MAV in unknown and low-sunlight environments. *Journal of Intelligent & Robotic Systems*, 74(1-2):59–67, 2014.

[5] https://jpchanson.github.io/ARdrone/ParrotDevGuide.pdf.

[6] Jakob Engel, Jurgen Sturm, and Daniel Cremers. Scale-aware navigation of a low-cost quadrocopter with a monocular camera. *Robotics and Autonomous Systems*, 62(11):1646 – 1656, 2014. Special Issue on Visual Control of Mobile Robots.

[7] Georg Klein and David Murray. Parallel tracking and mapping for small AR workspaces. In *Mixed and Augmented Reality, 2007. ISMAR 2007. 6th IEEE and ACM International Symposium on*, pages 225–234. IEEE, 2007.

[8] Franck Luthon and Benoit Larroque. LaboREM a remote laboratory for game-like training in electronics. *IEEE Transactions on Learning Technologies*, 8(3):311–321, 2015.

[9] Gerard Medioni and Sing Bing Kang. *Emerging topics in computer vision*. Prentice Hall PTR, 2004.

[10] David G Lowe. Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, volume 2, pages 1150–1157. IEEE, 1999.

[11] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (SURF). *Computer vision and image understanding*, 110(3):346–359, 2008.

[12] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. ORB: An efficient alternative to SIFT or SURF. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2564–2571. IEEE, 2011.

[13] Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua. Brief: Binary robust independent elementary features. *Computer Vision–ECCV 2010*, pages 778–792, 2010.

[14] Scott Krig. *Interest point detector and feature descriptor survey*. Springer, 2014.

[15] Georgios D Evangelidis and Emmanouil Z Psarakis. Parametric image alignment using enhanced correlation coefficient maximization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(10):1858–1865, 2008.

[16] Fadi Dornaika. Registering conventional images with low resolution panoramic images. *The 5th International Conference on Computer Vision Systems*, 2007.

[17] https://youtu.be/42nZTCsfQjE.

[18] https://youtu.be/Kr6TnjoByZ0.

[19] https://youtu.be/2gYpB9HaCcA.