3D Object-Camera and 3D Face-Camera Pose Estimation for Quadcopter Control: Application to Remote Labs

Fawzi Khattar[1,3](✉), Fadi Dornaika[3,4], Benoit Larroque[2], and Franck Luthon[1]

[1] UNIV PAU & PAYS ADOUR/E2S UPPA, LIUPPA Lab, Anglet, France
{fawzi.khattar,franck.luthon}@univ-pau.fr

[2] UNIV PAU & PAYS ADOUR/E2S UPPA, SIAME Lab, Anglet, France
benoit.larroque@univ-pau.fr

[3] University of the Basque Country UPV/EHU, San Sebastian, Spain
fadi.dornaika@ehu.eus

[4] IKERBASQUE, Basque Foundation for Science, Bilbao, Spain

Abstract. We present the implementation of two visual pose estimation algorithms (object-camera and face-camera) with a control system for a low cost quadcopter for an application in a remote electronic laboratory. The objective is threefold: (i) to allow the drone to inspect instruments in the remote lab, (ii) to localize a teacher and center his face in the image for student-teacher remote communication, (iii) and to return back home and land on a platform for automatic recharge of the batteries. The object-camera 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 SLAM system by allowing close range inspection of a planar instrument and autonomous landing. The face-camera localization system is based on 3D modeling of the face, and a state of the art 2D facial point detector. Experiments conducted in a remote laboratory workspace are presented. They prove the robustness of the proposed object-camera visual pose system compared to the SLAM system, the performance of the face-camera visual servoing and pose estimation system in terms of real-time, robustness and accuracy.

1 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 [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

J. Blanc-Talon et al. (Eds.): ACIVS 2018, LNCS 11182, pp. 99–111, 2018.
https://doi.org/10.1007/978-3-030-01449-0_9

direct visual feedback of the results of an experiment to the student. Furthermore it can also search for a teacher in the lab and move towards him, centering his face in the image to allow remote student-teacher interaction. In this way the student can achieve his lab-work and can also ask questions to the teacher in case he needs to. To achieve this double objective 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 or face of the teacher in our case). Second, a control system that sends appropriate commands to the quadcopter in order to reach a reference 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 are 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. Stereo rig cameras [3] and RGBD cameras [4] have been investigated. They allow for absolute pose estimate however this comes with an additional weight and power consumption. In this work we use 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. The well known SLAM algorithm PTAM (Parallel tracking and mapping) [6] combined with inertial and ultrasound sensors readings in order to get the absolute 3D pose estimate is used in [7]. All the available data is processed in a Kalman filter allowing for information fusion and delay compensation. Here, starting from [7] we propose the object-camera pose system and use it for 3D pose estimation when the quadcopter is exploring the 3D world to search for an instrument or for a teacher. However, since the visual SLAM relies on corresponding points detected in the flow of images, it will 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. 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 (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 drone is in a short distance range from the object of interest or when it needs to land on the electrical recharge platform. To be able to control the drone to center the face of a teacher in the image, the position of the drone with respect to the face of the teacher must be estimated. For this purpose, we use a deformable 3D model of the face and fit it to the image. The paper is organised as follows: In the second section we present an overview of the remote electronic lab LaboREM. In Sect. 3 we explain how detection and 3D localization can be carried out for the purpose of remote instrument inspection. In Sect. 4 we present the face-camera pose estimation approach. In Sect. 5, the 3D pose visual servoing of the quadcopter is explained. In the final section, we present our experiments and results.

2 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. The remote lab application is developed using NI-LabVIEW software and the easy-to-use RFP protocol to pilot the remote devices. The hardware setup 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, inspecting electrical instruments and interacting with a teacher in order to increase student immersion and motivation.

3 Object-Camera Pose Estimation

3.1 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 object-camera pose estimation. The homography matrix is a matrix that relates 3D points lying on a plane to their 2D projections. Given this transform one can directly calculate the rotation and translation matrix as done in [9]. In order to estimate the homography 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. A survey about keypoint detectors and descriptors can be found in [10]. 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 NLM (non linear minimization) problems that can be solved using iterative algorithms like gradient descent or LM (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) [11]. In practice the first one uses a brightness model in order to cope with variation of additive and multiplicative change of illumination [12] 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 coarse 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 [13]. In the detection step, template matching on a pyramid of the image is used to search for the desired object. If the normalized correlation coefficient is greater than α the object is declared detected. Once the detection is done, a homography transformation is computed by using the bounding box of the detected object to determine the object-camera 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 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 distance between the camera of the drone and the instrument is less than a threshold λ, the SIFT descriptor [14] is used to allow more robustness to orientation changes. The object is declared detected if the number of matched keypoints is greater than N. 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 estimated traveled distance between two consecutive frames and comparing it to a threshold D, we can detect a loss of tracking. Another threshold β is also imposed on the difference of each angle of orientation (yaw, roll, pitch). If the tracking fails, 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 Fig. 1. 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 [7]. The homography estimation process is shown in Fig. 1 while the feedback control loop is shown in Fig. 2.

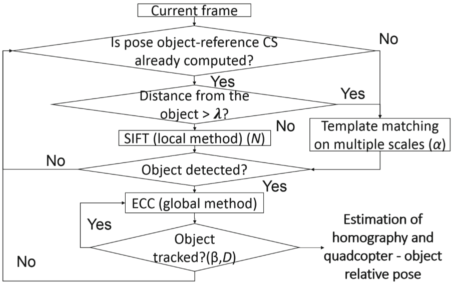

Current frame
Is pose object-reference CS already computed?
No
Yes
Distance from the object > λ?
Yes
No
SIFT (local method) (N)
Template matching on multiple scales (α)
No
Object detected?
Yes
ECC (global method)
Yes
Object tracked?(β,D)
No
Estimation of homography and quadcopter - object relative pose

Fig. 1. Homography estimation diagram.

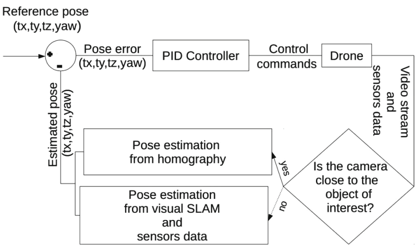

Reference pose (tx,ty,tz,yaw)
+
-
Pose error (tx,ty,tz,yaw)
PID Controller
Control commands
Drone
Video stream and sensors data
Estimated pose (tx,ty,tz,yaw)
Pose estimation from homography
Pose estimation from visual SLAM and sensors data
yes
no
Is the camera close to the object of interest?

Fig. 2. Feedback control loop.

4 Face-Camera Pose Estimation

Human-drone interaction is an interesting way of controlling drones. In [15] authors use face pose and hand gestures in order to allow human-drone interaction. Their face pose estimation process is based on the Viola and Jones face detector [16]. They compute a face score vector by applying frontal and side face detector on the flipped and the original image. Using this face score and a machine learning technique they estimate the *yaw* angle of the face pose. The distance from the face is estimated by the size of the face bounding box. Hand gestures are used to give order to the drone to move to an orientation while maintaining the distance from the face. In [17], authors also used hand gestures and face localization for drone-human interaction. Their approach is unique for the fact that it allows the drone to approach a human that is 20 meters away, by detecting periodic hand gestures. The drone then approach the target by tracking its appearance. Once at a short distance, the drone centers the face of the subject and detects hand gestures in order to take a picture. However, the orientation of the face is not estimated and the user has to be facing the camera in order to take a frontal photo. In this work, we adopt a 3D approach that models the human face in 3D and subsequently uses full perspective projection in order to recover the 3D face pose parameters. By using this modeling and matching it with image specific data related to the face, all the 6 pose parameters are inferred. The 3D modeling is based on the CANDIDE deformable 3D face model.

4.1 CANDIDE 3D Model

CANDIDE is a parameterized 3D face model specifically developed for model-based coding of human faces. CANDIDE is controlled by 3 sets of parameters: global, shape and animation parameters. The global parameters correspond to the pose of the face with respect to the camera. There exist 6 global parameters: 3 Euler angles for the rotation and 3 for the translation (t_x, t_y, t_z). The shape parameters adjust facial features position in order to fit to different subjects (eye width, distance between the eyes, face height etc.). The animation parameters adjust facial features position in order to display facial expressions and animations (smile, lowering of eyebrows). The 3D generic model is given by the 3D coordinates of its vertices $P_i, i = 1, n$. where n is the number of vertices. This way, the shape, up to a global scale, can be fully described by a 3n-vector g, the concatenation of the 3D coordinates of all vertices:

$$g = G + S\tau_s + A\tau_a \tag{1}$$

G is the standard shape of the model, the columns of S and A are the shape and animation units, $\tau_s \in \mathbb{R}^m$ and $\tau_a \in \mathbb{R}^k$, are the shape and animation control vectors, respectively.

4.2 Inferring Pose Parameters

In order to determine the pose from the 3D model, we have to fit this model to the face data available in the image. Fitting the model means determining its different parameters: pose, shape and animation parameters. In this work only pose and shape parameters are of interest for us, however recovering the animation parameters can be an interesting way to allow human-drone interaction based on facial expressions. Different approaches attempt to adapt the model in different ways. However, the majority of them follow a step by step approach, starting by estimating the shape parameters τ_s in order to adapt the 3D model to different face anatomy and then estimating the pose and animation parameters. From the face image, many face related data can be used to fit the 3D model. In [18], the authors use the gray scale appearance of the image to adapt the 3D model after estimating its shape parameter off-line. In our work, we make use of the advancement in facial landmark detection and use these landmarks to adapt the model and recover the 3D face pose from a set of 3D-to-2D correspondences. The shape parameters are estimated using a frontal picture of the subject following the method described in [19]. We use the facial point detector in [20], that can detect 68 2D landmarks on a face in one millisecond by a pre-trained ERT (Ensemble of Regression Trees), given that a face image patch is available. However, since the algorithm needs a region of interest that contains a face, the total time for its execution from the detection of the face to the detection of the landmarks is more than one millisecond due to the computationally expensive face detection step. One way to reduce this time and make the process working at more than 30 fps (frame per second) is to perform a search for the face around the last detected bounding box of the face instead of looking for the face in the whole image. We make use of only 46 points from the 68 points given by the landmark detector. The points were chosen to be semantic and mostly rigid thus eliminating points along face contour. Once the 2D landmarks are detected in the image we use state of the art pose estimation algorithms that are based on 3D-2D point correspondences to recover the pose. This problem is known in the literature as PnP (Perspective n Point). Many algorithms attempt to solve this problem. The P3P algorithm [21] (perspective 3 point) can estimate the pose using only 3 point correspondences. Other algorithms like EPNP [22] (Efficient Perspective N Point) can handle any number of points. Another approach is to use non-linear minimization techniques to recover the pose that best minimizes the distance between the projected 3D points and the 2D points. However, this method requires an initial guess of the pose parameters in order to converge to the global minimum. This initial guess can be made available using the estimated pose from the previous frame or using any closed-form solution like EPNP, P3P, etc. in case it is not available. We propose to use the Levenberg-Marquardt technique as it gives good results and fast execution time. The face-camera pose estimation process is shown in Fig. 3. The pose used to control the drone is computed by fusing the visual pose from the 3D model with inertial and ultrasound measurements in a Kalman filter as done in [7].

5 Visual Control of the Quadcopter

In order to control the 3D position and orientation of the quadcopter for the purpose of instrument inspection, a closed-loop control is used taking as a feedback the pose derived from the SLAM or the homography. The control loop is shown in Fig. 2. If the objective is to control the drone to maintain a relative position from the face, the algorithm explained in Sect. 4 is used as a feedback sensor for the control loop. The controlled degrees of freedom associated with the quadcopter are the 3D translation vector and the yaw angle. Each degree is controlled by a closed loop control system with a traditional PID controller.

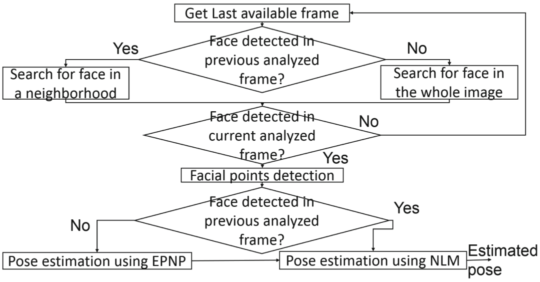

Get Last available frame
Face detected in previous analyzed frame?
Yes
No
Search for face in a neighborhood
Search for face in the whole image
Face detected in current analyzed frame?
No
Yes
Facial points detection
Face detected in previous analyzed frame?
No
Yes
Pose estimation using EPNP
Pose estimation using NLM
Estimated pose

Fig. 3. Face-camera pose estimation.

6 Experiments

Before presenting the different scenarios for evaluation, we begin with an evaluation of the performance of the Face-Camera pose estimation. We compare the accuracy of different techniques and their execution time on a database for pose estimation (UPNA head pose database) [23]. The UPNA database contains 120 videos corresponding to 10 different subjects, 12 videos each, in which the subject changes its head pose by following guided and free movement. The ground-truth relative 3D face motion is known for all frames in all videos. We conclude that all the techniques converge to the same optimal solution if followed by a non-linear minimization method. As shown in Table 1 the method that yields the best pose estimate and excecution time is the Levenberg- Marquardt method that tracks the 3D face from a frame to the next one based on an initial estimate. 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 drone-face visual servoing.

First Scenario: The first experiment aims 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 reference 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. Figure 4 shows the response of the system facing manual perturbations applied to the drone (5 perturbations to the x position, 2 for y, 2 for z and 3 for the angle yaw). The objective is to see the behavior of the system when the drone is pushed away from the reference pose causing the visual tracking to fail. Despite the loss of tracking (red curves in Fig. 4) due to the fast kicks applied to the drone, it is always able to return to a position that allows the tracking to restart. This is done by controlling the drone based on the pose estimation procured by the Kalman filter [7] that fuses the inertial and ultrasound measurements to have an estimate of the current position of the drone. Two videos of this scenario are available online [24,25].

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 (by following a vertical path in order to change the height and correctly estimate the scale of the SLAM map [7]), and initiate a search procedure for the required instrument. After object detection, the drone moves towards the instrument and the feedback loop uses the 3D pose based on the homography for control. In this way the quadcopter is able to fly to inspect an electrical instrument maintaining its position with respect to the instrument. After the mission is over the drone is sent back home by using the pose derived from the SLAM. The position of the landing platform is estimated by using the homography-based pose estimation applied on the video of the bottom camera of the AR Drone 2.0. The drone hover then at a certain altitude above the landing platform preparing for landing. A quantitative comparison between the approach proposed here for pose estimation when inspecting an object at close range with the SLAM algorithm used in [7] is shown in Fig. 5. It shows the superiority of our approach and the drift of the SLAM approach due to the reasons explained in the introduction. Figure 6 shows the first detection and the tracking of the object of interest and its robustness in spite of problems of tracking failure. The green cross represents the center of the image while the red cross represents the projection of the center of the object on the image. The objective is to control the position of the drone so that the two crosses are as close as possible. Videos of this scenario are available in [26,27].

Third Scenario: In this scenario we test the performance of the face-camera visual servoing system explained in Sect. 4. The drone has to fly, detect a face, align its line of sight with that of the subject face, and centering the face in the image while maintaining a fixed distance with it. In this experiment the user is in motion in order to induce perturbation to the control system. The drone has

Table 1. Average pose errors and computation time for different face pose estimation methods. t_x, t_y, t_z are in millimeters, *roll*, *yaw*, *pitch* in degrees, time in milliseconds.

Method	t_x	t_y	t_z	roll	yaw	pitch	time
EPNP	11,84	7,11	12,67	0,55	3,74	2,39	0.117
Ransac P3P	12,50	7,79	18,34	1,56	6,52	6,11	0.898
EPNP + NLM	11,51	7,23	13,38	0,56	2,28	1,45	0.363
P3P Ransac + NLM	11,51	7,23	13,38	0,56	2,28	1,45	1.138
NLM	**11,51**	**7,23**	**13,38**	**0,56**	**2,28**	**1,45**	**0.234**

to correct for the user displacement and the out-of-plane orientation of his face. Figure 7 shows the experiment seen from the camera of the drone and from an external camera. A video of the experiment is available on [28].

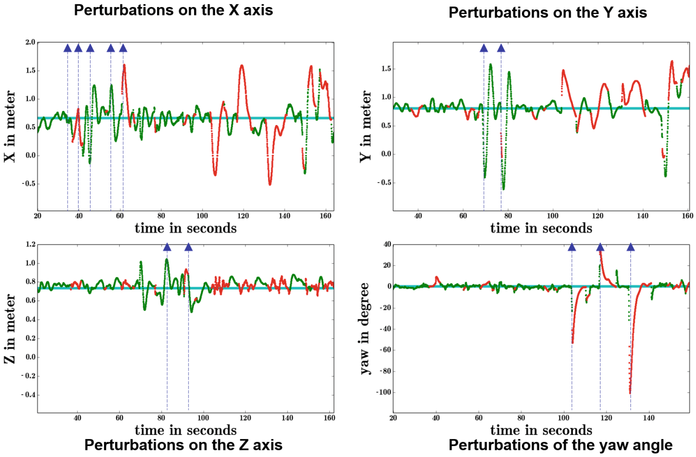
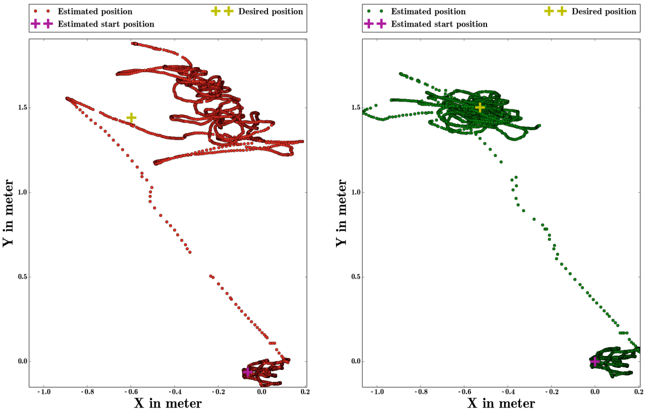


Fig. 4. Quadcopter control and estimated pose facing perturbations. For each control loop: In green the estimation from the homography when the tracking is good, in red estimation based solely on navigation data when the tracking fails and in light blue the reference 3D pose. Vertical arrows indicate the time each perturbation was applied. (Color figure online)



(a) (b)

Fig. 5. Estimated pose in the XY plane. (a) with [7], (b) with the proposed approach.

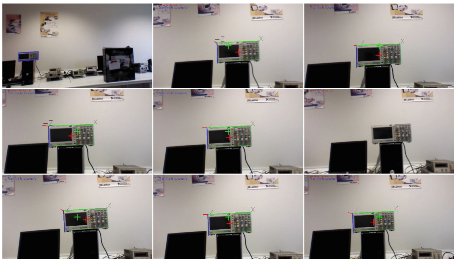


Fig. 6. Detection and tracking of the instrument. (Color figure online)

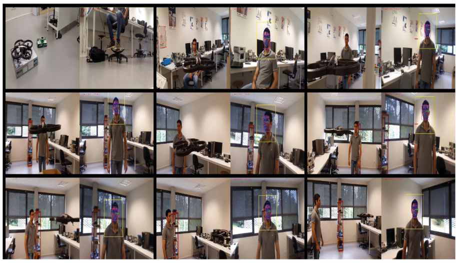


Fig. 7. Third scenario experiment: Face tracking.

7 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 objective is to allow remote instrument inspection and remote human-teacher communication. The proposed localization system for the first objective is proven to outperforms the SLAM system in [7] through qualitative and quantitative experiments, allowing the quadcopter to inspect an object and return to its base autonomously. The approach uses only the on-board sensors available on the low cost drone. The localization system for face-camera servoing is based on 3D modelling of the face and a state-of-the art 2D facial point detector. The approach controls all 4 degrees of freedom (3° for translation as well as the orientation of the face). It is shown robust, accurate and working at frame rate through qualitative and quantitative experiments.

References

1. Luthon, F., Larroque, B., Khattar, F., Dornaika, F.: Use of gaming and computer vision to drive student motivation in remote learning lab activities. In: 10th Annual International Conference of Education, Research and Innovation, ICERI 2017, pp. 2320–2329 (2017)
2. Eberli, D., Scaramuzza, D., Weiss, S., Siegwart, R.: Vision based position control for MAVs using one single circular landmark. J. Intell. Robot. Syst. **61**(1–4), 495–512 (2011)
3. Schauwecker, K., Zell, A.: On-board dual-stereo-vision for the navigation of an autonomous MAV. J. Intell. Robot. Syst. **74**(1–2), 1–16 (2014)
4. Flores, G., Zhou, S., Lozano, R., Castillo, P.: A vision and GPS-based real-time trajectory planning for a MAV in unknown and low-sunlight environments. J. Intell. Robot. Syst. **74**(1–2), 59–67 (2014)
5. https://jpchanson.github.io/ARdrone/ParrotDevGuide.pdf

6. Klein, G., Murray, D.: Parallel tracking and mapping for small AR workspaces. In: 6th IEEE and ACM International Symposium on Mixed and Augmented Reality, ISMAR 2007, pp. 225–234. IEEE (2007)
7. Engel, J., Sturm, J., Cremers, D.: Scale-aware navigation of a low-cost quadrocopter with a monocular camera. Robot. Auton. Syst. **62**(11), 1646–1656 (2014). Special Issue on Visual Control of Mobile Robots
8. Luthon, F., Larroque, B.: LaboREM a remote laboratory for game-like training in electronics. IEEE Trans. Learn. Technol. **8**(3), 311–321 (2015)
9. Medioni, G., Kang, S.B.: Emerging Topics in Computer Vision. Prentice Hall PTR, Upper Saddle River (2004)
10. Krig, S.: Interest Point Detector and Feature Descriptor Survey. In: Computer Vision Metrics. Apress, Berkeley (2014). https://doi.org/10.1007/978-1-4302-5930-5_6
11. Evangelidis, G.D., Psarakis., E.Z.: Parametric image alignment using enhanced correlation coefficient maximization. IEEE Trans. Pattern Anal. Mach. Intell. **30**(10), 1858–1865 (2008)
12. Dornaika, F.: Registering conventional images with low resolution panoramic images. In: The 5th International Conference on Computer Vision Systems (2007)
13. Fawzi, K., Fadi, D., Franck, L., Benoit, L.: Quadcopter control using onboard monocular camera for enriching remote laboratory facilities. In: 2018 IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR). IEEE (2018)
14. Lowe, D.G.: Object recognition from local scale-invariant features. In: The proceedings of the Seventh IEEE International Conference on Computer Vision, vol. 2, pp. 1150–1157. IEEE (1999)
15. Nagi, J., Giusti, A., Di Caro, G.A., Gambardella, L.M.: Human control of UAVS using face pose estimates and hand gestures. In: Proceedings of the 2014 ACM/IEEE International Conference on Human-Robot Interaction, pp. 252–253. ACM (2014)
16. Paul, V., Jones, M.J.: Robust real-time face detection. Int. J. Comput. Vis. **57**(2), 137–154 (2004)
17. Monajjemi, M., Mohaimenianpour, S., Vaughan, R.: UAV, come to me: end-to-end, multi-scale situated HRI with an uninstrumented human and a distant UAV. In: 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 4410–4417. IEEE (2016)
18. Dornaika, F., Davoine, F.: On appearance based face and facial action tracking. IEEE Trans. Circuits Syst. Video Technol. **16**(9), 1107–1124 (2006)
19. Unzueta, L., Pimenta, W., Goenetxea, J., Santos, L.P., Dornaika, F.: Efficient deformable 3d face model fitting to monocular images (2016)
20. Kazemi, V., Josephine, S.: One millisecond face alignment with an ensemble of regression trees. In: 27th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, United States, 23 June 2014 through 28 June 2014, pp. 1867–1874. IEEE Computer Society (2014)
21. Gao, X.-S., Hou, X.-R., Tang, J., Cheng, H.-F.: Complete solution classification for the perspective-three-point problem. IEEE Trans. Pattern Anal. Mach. Intell. **25**(8), 930–943 (2003)
22. Lepetit, V., Moreno-Noguer, F., Fua, P.: EPNP: an accurate O(n) solution to the PnP problem. Int. J. Comput. Vis. **81**(2), 155 (2009)
23. Ariz, M., Bengoechea, J.J., Villanueva, A., Cabeza, R.: A novel 2d/3d database with automatic face annotation for head tracking and pose estimation. Comput. Vis. Image Underst. **148**, 201–210 (2016)

24. https://youtu.be/42nZTCsfQjE
25. https://youtu.be/Kr6TnjoByZ0
26. https://youtu.be/kXZH9uz9Hkc
27. https://youtu.be/PTMVeJizjF8
28. https://youtu.be/Xytlz0UdaDk