An Autonomous Quadcopter Detecting and Tracking an Unmanned Ground Vehicle

Victor Massague Respall
Institute of Robotics, Innopolis University
Innopolis, 420500, Russia
Email: v.massague.respall@innopolis.university

Sami Sellami
Institute of Robotics, Innopolis University
Innopolis, 420500, Russia
Email: s.sellami@innopolis.university

***Abstract*—This paper presents the design and implementation of a system where an autonomous quadcopter has to detect, track and land on a moving Unmanned Ground Vehicle (UGV). The UAV needs to identify and track the target using camera based perception. A color-based detection algorithm is used to detect the position of the ground vehicle and Kernelized Correlation Filters (KCF) [1] is used for tracking it. This position is then used in the PID feedback control loop that enables the actual tracking of the moving platform. The whole project is first tested on Gazebo simulator, and after obtaining successful results, Parrot Ar Drone 2.0 (quadcopter) and Plato robot (ground vehicle) are used for testing the developed algorithm in real environment. The drone was able to detect and track the vehicle in real time and to land in the vicinity of the UGV.**

I. Introduction

Nowadays, considerable research has been carried out in the field of unmanned aerial vehicles (UAV). Its first usage was during the First World War and during the last years it became a new entertainment form and also a research area to explore. The most known form of UAV is quadcopters, they are an excellent option for different applications like terrain recognition, accessing difficult or dangerous areas, delivering objects, collecting data or simply for recreation. Clearly, one useful application would be the ability to track and land on a moving object, this could be used in many areas like terrain's exploration with a ground and aerial vehicles, rescue and military missions or simply commercial purposes like in goods delivery.

For all these tasks one needs a system for detection, tracking and landing in different situations and environments where the target or platform may be in motion. Landing on a moving target is a challenging task to deal with because of the a priori unknown motion and the nonlinear copter's dynamics. Another main problem is the autonomy of the copter, since typically, it is about 20 minutes. However this problem is out of consideration in this project.

The present project deals with the problem of designing a detection, tracking and landing system along with a controller for a low cost quadcopter which needs to follow a moving platform. Basically it implies two parts:

- Designing the feedback controller for the UAV
- Implementing the detection, tracking and landing system using on-board camera

The paper is organized as follows: Section II describes related works to this research. The methodology is explained in section III. Section IV contains the simulation and experiments results. Finally section V the conclusions.

II. Related Works

The development of an object tracking controller for a quadcopter using on-board vision system was presented in [**?**], using low-cost components. A low-frequency monocular computer vision algorithm was applied in closed-loop control to track an object of known color. Parallel PID controllers for aircraft bearing, relative height and range were implemented with feedback from object offset and size in the image frame. The noise generated in the image measurement is mitigated with Kalman filter. Similar project applied to a ship platform using an helicopter was performed. The problem was presented as a nonlinear tracking control problem for an underactuated system with a two time-scale controller. Through simulations, the proposed approach was demonstrated to be effective for the problem [2]. The work elaborated in [3] presents a copter system capable of autonomously land on a moving platform using state-of-the-art computer vision algorithms and multi-sensor fusion.

Another work [4], related to marine environment, used the quadcopter chosen for this research, Ar Drone, and presents an algorithm that takes up the challenge of tough landing conditions prevalent in the oceans due to winds and currents. Landing pad sensing was achieved through image processing techniques using MATLAB. Testing has been carried out both indoors, and outdoors over open water, with a success rate of over 75% .

In this paper [5], was presented the design and implementation of a real-time, vision-based landing algorithm for an autonomous helicopter. The landing algorithm was integrated with algorithms for visual acquisition of the target (a helipad) and navigation to the target, from an arbitrary initial position and orientation. They used vision for precise target detection and recognition, and a combination of vision and Global Positioning System for navigation. They presented significant results from flight trials in the field which demonstrated that their detection, recognition, and control algorithms were accurate, robust, and repeatable.

A low-cost flight control system for a small outdoor helicopter where all processing is performed on-board was described by [6] and [7] shows a vision-based road-following algorithm for an autonomous fixed-wing aircraft with a wingspan.

Experiments with an external motion capture system were conducted by [8]. They achieved a maximum deviation of 10 cm from the desired position. Also [9] were using a field monitored by stationary cameras. They presented an assistant and training system for controlling RC helicopters during flight.

The idea of using LEDs as landmarks is not completely new. In [10] is described how a miniature helicopter can be visually tracked in six degrees of freedom with only three on-board LEDs and a single on-ground camera. Also the Wii remote camera can be found in former projects. [11] presented an optical tracking system using the camera data of two Wii remotes. Similar approach using Wii remotes was used in [12] and the system provides small errors and allows for safe, autonomous indoor flights.

III. Methodology

Here we briefly describe each step of the algorithm used to solve the task. Since we are working with images, and a good precision is required to obtain accurate positions of the ground vehicle with respect to the quadcopter, calibration of the on-board's bottom camera is done for suppressing possible distortions. This procedure is repeated every time the drone is powered up. In the next step, the drone is ready to take off and start the process of detecting the ground vehicle in the current frame.

The algorithm is performed in two phases, the first one is detection and tracking and the second deals with the motion control of the drone. The detection process is executed for each frame until the UGV is detected. If the UGV is successfully localized in the image, a tracking algorithm is then used for acquiring the position of the ground vehicle in the image coordinate frame.

The second phase, is in charge of receiving the aforementioned position and calculate the required command inputs to apply in order to follow the ground vehicle. Kalman filter is used to help reducing the noise generated by the measurements. The process is repeated in an infinite loop until a predefined landing signal is sent to the drone. A simple scheme of this process can be seen in Figure 1

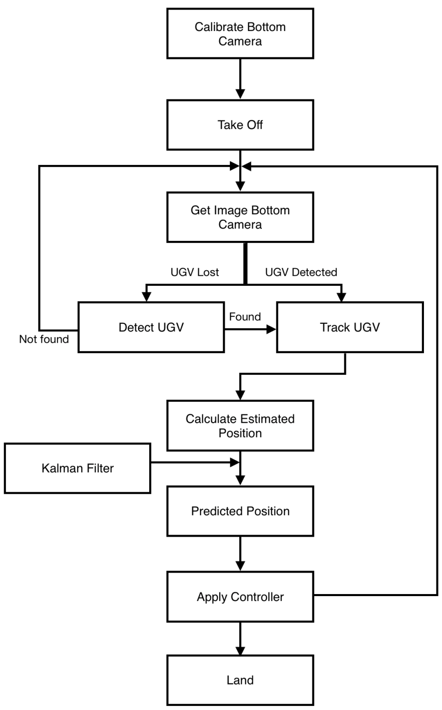

Fig. 1. Diagram showing the methodology developed

A. Detection and Tracking

To obtain the current position of the UGV considering it is within the on-board camera field of view, one approach would be to detect the UGV in every frame. However this technique is not computationally efficient since it tries to search for the object in the whole image frame, without using previous information about its last position.

Instead, we would like to apply object detection only once and then use the object tracker to handle every subsequent frame, leading to a faster, more efficient object tracking pipeline. This is exactly what many trackers do. Trackers are algorithms that try to optimize such process using different information acquired over time. A good tracking algorithm will use all information it has about the object up to that point while a detection algorithm always starts from scratch. Some examples of trackers are Boosting which is based on AdaBoost [13], Multiple Instance Learning (MIL) [14], Tracking, Learning and Detection (TLD) [15].

For our project, a fast and precise tracker is required which can work in online applications in order to keep the ground vehicle position in the current frame. it should also be robust to tracking failures, which means that it is able to recover from a lost position while tracking. All these characteristics are included in Kernelized Correlation Filters tracker (KCF). The basic idea of the correlation filter tracking is estimating an optimal image filter such that the filtration with the input image produces a desired response. The desired response is typically of a Gaussian shape centered at the target location, so the score decreases with the distance.

The filter is trained from translated (shifted) instances of the target patch. When testing, the response of the filter is evaluated and the maximum gives the new position of the target. The filter is trained on-line and updated successively with every frame in order for the tracker to adapts to moderate target changes. Major advantage of the correlation filter tracker is the computation efficiency. it can be performed efficiently in the Fourier domain. Thus the tracker runs super-real time, in several hundreds FPS.

The UGV and UAV used for this task are depicted in Figure

2.

Fig. 2. From left to right: Differential drive robot called Plato used as a target for tracking and platform for landing on it.

For the detection part, Gaussian filtering is applied for smoothing and removing possible noise in the image, due to the low quality of the constantly moving camera. Then we take advantage of the color of the UGV and try to detect the platform by color, in this case yellow. We used HSV color space to isolate the yellow color while taking into account the possible illumination changes. This step will allow us to know if the robot is in the frame of the copter. After the detection phase we construct a bounding box around the UGV that will be used by the tracker in future frames. An extra feature, which is the ORB feature detector, will further be used to double check that the bounding box found using color detection actually contains the UGV (suppressing the false positive cases).

B. PID Controller

PID controller is a well defined classical controller and has been in practice for many years due to it's design and performance. As the name suggests the PID consist solely of three elements Proportional, derivative and integral. The mathematical form of PID controller is given below,

$$u(t) = K_p e(t) + K_i \int e(\tau) d(\tau) + K_d \frac{de}{dt}$$

wheres K_p, K_i, K_d are the proportional, integral and derivative gains and $e(t)$ is the error signal i.e., the difference between desired and measured value.

The basic operation principle of PID control is that it takes measured value from output sensor, compares it with the desired value and than accordingly sends commands to a plant. Plant can be thought as the system on which the desired actions are being performed to optimize it's behaviour. The block diagram is shown in Figure 3.

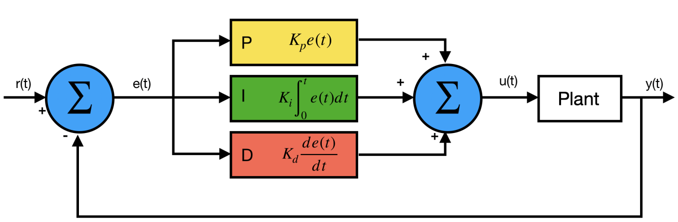

Fig. 3. PID Controller Schematic Diagram

The three terms of PID controller i.e., proportional integral and derivative are responsible for different actions. Proportional gain just multiplies a constant value with the error. It aggressively tries to match output and input. The higher value of proportional gain results in improved rise time . But increasing the proportional gain results in more oscillations around steady state value.

Next is the derivative term, increasing this term results in less oscillations in a system which can be a very good feature in designing stable systems. But it has a limit also, increasing the derivative term make our system over-damped i.e., it takes a lot of time to reach it's steady state value. Also it reduces the rise time of the system. The last term is the integral. It just integrates the error of the input and output over time. In this manner it tries to improve steady state error of the system [16].

In order to control the quadcopter x-y trajectory (Figure 4), we used a PD controller with a component proportional to the error between the desired and the actual trajectory, and a component proportional to the derivative of the error. For the altitude control, we used a PID controller with another term proportional to the error accumulation over time, This choice is primarily motivated by the continuous change of the trajectory in the xy plane, whereas the altitude should remain constant over time, the command scheme of the quadcopter allows us to control the linear velocities in each direction, thus the input signal would be proportional to the aforementioned velocities [17].

$$u_z(t) = K_{pz} e_z(t) + k_{iz} \int_0^t e_z(t) dt + K_{dz} \frac{d}{dt} e_z(t)$$

Where: $e_z(t) = z^* - z$ is the difference between the actual and the desired altitude (which is given by the sensors measurements)

$$u_x(t) = K_p e_x(t) + K_d \frac{d}{dt} e_x(t)$$

$$u_y(t) = K_p e_y(t) + K_d \frac{d}{dt} e_y(t)$$

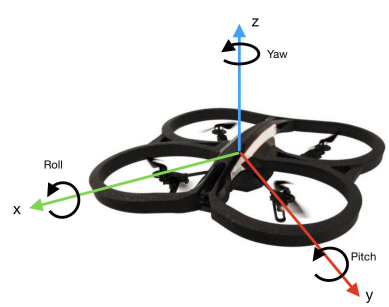

Fig. 4. Local frame representation of the UAV

We use different PID coefficient for the altitude and x-y positions control, the altitude being a sensor measurement it is quite different in scale from the desired x-y position which is obtained from the computer vision based algorithm used to track the UGV position. These coefficients are tuned in real time while controlling the UAV.

IV. Simulation and Experiments

A. Simulation

First phase of experiments was performed using with Gazebo simulator, this phase allowed studying the behaviour of the quadcopter while it was detecting and tracking the UGV platform. Figure 5 shows screenshots of the simulation performed. Here the task was just to obtain a robust and accurate version of the algorithm, especially for the detection and tracking system. The ground vehicle used is a modified version of the turtle bot. After obtaining a stable behavior for the system in the simulation, we moved to the real hardware experiment.

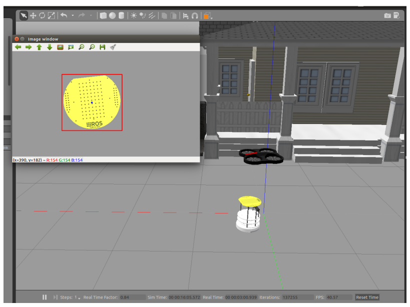

Fig. 5. Gazebo simulation of the task using a platform UGV and ArDrone 2.0. The figure depicts the initial the copter status after taking off and the live streaming of the bottom camera.

B. Results

We hereby discuss the results of these experiments, which are shown in Figure 6 and the video available at https://youtu.be/9nc1TGTTwhY. Performing the experiments, we encountered some problems due to unpredictable and uncontrollable factors such as noise in the camera images, illumination changes and hardware failures. Also, the drone having a self stabilization (auto-hover) feature which uses the bottom camera for auto-stabilization, we were unable to fully control the drone, this is why we disabled this feature.

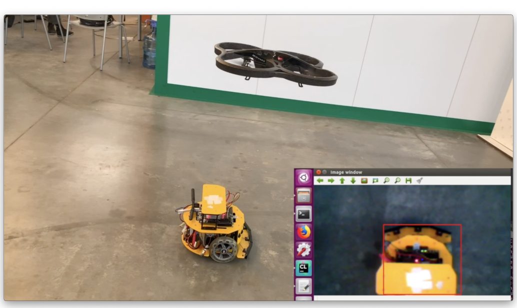

Fig. 6. Ar Drone 2.0 following Plato robot, and live video streaming of the copter in the bottom-right part of the figure with the bounding box around the object of interest

Testing the feedback controller results in quite different behavior from the simulation, indeed, the UAV was not able to accurately following the desired trajectories and altitude.

To resolve this we tuned the PID parameters for the different controls schemes (altitude and translation control) , and we finally obtained a good behavior suitable regarding our specifications

Finally, a simple landing system is developed in order to be able to land while the UGV is moving. This means that we are decreasing the altitude of the drone while tracking the platform. The altitude sensor is used to determine the current altitude of the drone and make it completely autonomous when landing. A slight problem is that the drone at certain altitude were not able to detect the UGV anymore due to the small altitude and the shrinking of the field of view.

V. Conclusion

This project presents a research focused on the design and implementation of a system composed of an UAV able to detect, track and land on an UGV. First stage of the system, was to implement a detection and tracking system for the ground vehicle using color-based algorithm and KCF tracker respectively. Second stage was to design a control scheme to perform the tracking. For this a PID controller was implemented using the position returned from the first stage. The process is repeated all over again until a predefined command for landing is sent to the drone. Ar Drone was performing quite well while tracking, however due to the structure of the ground vehicle and the quadcopter, and for safety reasons, it was not possible to properly execute the landing procedure exactly on top of Plato robot.

More research on this topic could be carried out, testing different detection algorithms and controllers to improve the proposed approach.

References

[1] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, no. 3, pp. 583–596, 2015.

[2] S.-R. Oh, K. Pathak, S. K. Agrawal, H. R. Pota, and M. Garrett, "Autonomous helicopter landing on a moving platform using a tether," in *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*. IEEE, 2005, pp. 3960–3965.

[3] D. Falanga, A. Zanchettin, A. Simovic, J. Delmerico, and D. Scaramuzza, "Vision-based autonomous quadrotor landing on a moving platform," in *2017 IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR)*, Oct 2017, pp. 200–207.

[4] T. Zhao and H. Jiang, "Landing system for ar. drone 2.0 using onboard camera and ros," in *Guidance, Navigation and Control Conference (CGNCC), 2016 IEEE Chinese*. IEEE, 2016, pp. 1098–1102.

[5] S. Saripalli, J. F. Montgomery, and G. S. Sukhatme, "Visually guided landing of an unmanned aerial vehicle," *IEEE transactions on robotics and automation*, vol. 19, no. 3, pp. 371–380, 2003.

[6] J. M. Roberts, P. I. Corke, and G. Buskey, "Low-cost flight control system for a small autonomous helicopter," in *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*, vol. 1. IEEE, 2003, pp. 546–551.

[7] E. Frew, T. McGee, Z. Kim, X. Xiao, S. Jackson, M. Morimoto, S. Rathinam, J. Padial, and R. Sengupta, "Vision-based road-following using a small autonomous aircraft," in *Aerospace Conference, 2004. Proceedings. 2004 IEEE*, vol. 5. IEEE, 2004, pp. 3006–3015.

[8] D. Gurdan, J. Stumpf, M. Achtelik, K.-M. Doth, G. Hirzinger, and D. Rus, "Energy-efficient autonomous four-rotor flying robot controlled at 1 khz," in *Robotics and Automation, 2007 IEEE International Conference on*. IEEE, 2007, pp. 361–366.
[9] K. Watanabe, Y. Iwatani, K. Nonaka, and K. Hashimoto, "A visual-servo-based assistant system for unmanned helicopter control," in *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*. IEEE, 2008, pp. 822–827.
[10] L. C. Mak and T. Furukawa, "A 6 dof visual tracking system for a miniature helicopter," in *2nd International Conference on Sensing Technology (ICST)*. Citeseer, 2007, pp. 32–37.
[11] S. Hay, J. Newman, and R. Harle, "Optical tracking using commodity hardware," in *Proceedings of the 7th IEEE/ACM international Symposium on Mixed and Augmented Reality*. IEEE Computer Society, 2008, pp. 159–160.
[12] K. E. Wenzel, A. Masselli, and A. Zell, "Automatic take off, tracking and landing of a miniature uav on a moving carrier vehicle," *Journal of intelligent & robotic systems*, vol. 61, no. 1-4, pp. 221–238, 2011.
[13] H. Grabner, M. Grabner, and H. Bischof, "Real-time tracking via on-line boosting." in *Bmvc*, vol. 1, no. 5, 2006, p. 6.
[14] B. Babenko, M.-H. Yang, and S. Belongie, "Visual tracking with online multiple instance learning," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2009, pp. 983–990.
[15] Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-learning-detection," *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 7, pp. 1409–1422, 2012.
[16] K. J. Åström and T. Hägglund, *PID controllers: theory, design, and tuning*. Instrument society of America Research Triangle Park, NC, 1995, vol. 2.
[17] R. C. G. Szafranski, *Different Approaches of PID Control UAV Type Quadrotor*. Silesian University of Technology, Akademicka St 16, Gliwice, Poland, 2011.