A System for Autonomous Landing of a UAV on a Moving Vehicle

Sebastiano Battiato[1], Luciano Cantelli[2], Fabio D'Urso[1], Giovanni Maria Farinella[1], Luca Guarnera[1], Dario Guastella[2], Carmelo Donato Melita[2], Giovanni Muscato[2], Alessandro Ortis[1](✉), Francesco Ragusa[1], and Corrado Santoro[1]

[1] Dipartimento di Matematica e Informatica, University of Catania, Viale A. Doria 6, 95125 Catania, Italy
ortis@dmi.unict.it

[2] Dipartimento di Ingegneria Elettrica, Elettronica e Informatica, University of Catania, Viale A. Doria 6, 95125 Catania, Italy

Abstract. This paper describes the approach employed to implement the autonomous landing of an Unmanned Aerial Vehicle (UAV) upon a moving ground vehicle. We consider an application scenario in which a target, made of a visual pattern, is mounted on the top of a ground vehicle which roams in an arena using a certain path and velocity; the UAV is asked to find the ground vehicle, by detecting the visual pattern, and then to track it in order to perform the approach and finalize the landing. To this aim, Computer Vision is adopted to perform both detection and tracking of the visual target; the algorithm used is based on the TLD (Tracking-Learning-Detection) approach, suitably integrated with an Hough Transform able to improve the precision of the identification of the 3D coordinates of the pattern. The output of the Computer Vision algorithm is then exploited by a Kalman filter which performs the estimation of the trajectory of the ground vehicle in order to let the UAV track, follow and approach it. The paper describes the software and hardware architecture of the overall application running on the UAV. The application described has been practically used with success in the context of the "Mohamed Bin Zayed" International Robotic Challenge (MBZIRC) which took place in March 2017 in Abu Dhabi.

1 Introduction

Autonomous landing on a moving vehicle is an important problem that has been investigated by different research groups worldwide [2–4]. Cooperation between UAVs and Unmanned Ground Vehicles (UGVs) to help humanitarian demining operations [5–7] and for aerial monitoring [8,9] are some of the main applications in this context. In this paper we describe the system we have designed and employed in the MBZIRC Challenge. The Mohamed Bin Zayed International Robotics Challenge 2017 (MBZIRC) is a robotic competition held in Abu Dhabi in March 2017. The team of the University of Catania has been selected to

S. Battiato et al. (Eds.): ICIAP 2017, Part I, LNCS 10484, pp. 129–139, 2017.
https://doi.org/10.1007/978-3-319-68560-1_12

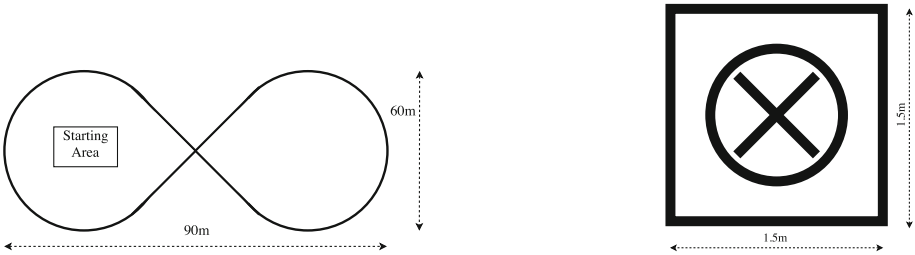

Fig. 1. Playing Area (left) and Visual Target (right)

participate to the Challenge 1 and this paper reports an overview and some details on the developed system. The Challenge 1 consists in the autonomous landing of an UAV on a moving platform [1]. According to MBZIRC rules, Challenge 1 requires a UAV to locate, track and land on a moving ground vehicle. The competition was performed in an open arena where a ground vehicle moves following an eight shaped trajectory, as shown in Fig. 1. On top of the vehicle, the landing area is a square of dimensions $1.5\,\mathrm{m} \times 1.5\,\mathrm{m}$ indicated by a given target (see the Visual Target depicted in Fig. 1). The UAV takes off from a given position and autonomously lands on the target placed above the moving vehicle, in the shortest time possible.

In the following sections an overview of the developed system and its components will be presented. Focus will be given to the main modules related to the dynamical estimator and the vision system. In particular, a tracking module able to detect and track a known pattern is employed to select a region of interest within the whole image. Then, a circle Hough transform is used to detect the center of the target with high precision. This system resulted the best solution defined taking into account several constrains related to the considered task. Indeed, the addressed Challenge requires the definition of an hardware and software system able to detect the target and its coordinates with very high precision, and combine this information with other data coming from different sensors (e.g., UAV position, speed, altitude) in order to define the best trajectory for the UAV. Due to the nature of the Challenge, this software pipeline is performed in real-time, taking into account further limits caused by the need of a lightweight hardware. Therefore, we discarded approaches to visual object tracking existing in the literature that have been formulated making specific assumptions on the application domain. For the considered task, a method to update the target representation during the tracking is required, due to the pose and scale changes the target is subject to. Several solutions based on the state of the art in object tracking have been considered [12–15]; the final choice is an hardware and software vision system consisting of an Ocam camera (chosen due its wide Field of View), which rectified images are processed with a pre-trained TLD based detection and tracking of the target and the circle Hough transform. Results of simulations and of the on-field trials will be presented and commented.

2 System Architecture

Challenge 1 requires carefully taking into account of the control aspects, Computer Vision algorithms and the development and integration of suitable hardware needed to perform the autonomous task. The basic overall approach we followed consists in reaching the center of the path by using a precise RTK-DGPS at an altitude suitable for a global view of the environment, for a preliminary detection and localization of the target by means of a wide range camera. Then, a visual detection and tracking procedure is able to estimate the position of the target and generate a suitable trajectory for the UAV. A dynamic estimator merges the measurements of the vision algorithms with the inertial and positioning measurements of the UAV and the estimated trajectories of the UGV. Then, based on the UAV dynamic, the estimator generates the optimal trajectory to reach the target in real time. When the UAV is in proximity of the target, Computer Vision techniques are adopted for the accurate estimation of the 3D coordinates of the target center to be used for safe landing. Once landed, all motors are switched off. The emphasis has been put on the use of lightweight hardware platforms. To this aim, the Computer Vision and control algorithms are optimized to run effectively on a lightweight high performance embedded system.

2.1 Hardware Architecture

The multirotor frame chosen for the competition is the "Spreading Wings S900" by DJI, characterized by high payload and stability. The PixHawk is used as autopilot, it is a high-performance system able to deal with both the stabilization and the navigation of the UAV. This simple but powerful system can be connected to an on-board companion computer that, by running the high-level navigation algorithms, can easily drive the UAV. The "eyes" of the multirotor are represented by an Ocam camera, a fish-eye camera which allows the exploitation of a wide Field of View. The image processing algorithm is executed by a Jetson TX1, an embedded system developed by NVIDIA for visual computing which provides a high performance GPU computing. The computed target position is used by the high level control algorithms to give the proper commands to the Pixhawk autopilot by means of the Mavlink protocol. The accuracy in the localization of the multirotor is ensured by an on-board RTK-DGPS system, receiving the corrections from a base station. In Fig. 2 the whole hardware platforms selected are shown.

2.2 Software Architecture

The control software runs on the Jetson TX1. The software architecture is designed as a multi-thread C/C++ application and it is executed on a Linux environment. Furthermore, for simulation purposes, the software is able to run inside a SITL (Software In The Loop) environment, using Gazebo as physics engine.

Fig. 2. Hardware platforms used.

The multi-thread process is composed by four threads, as shown in Fig. 3.

MAVLINK, PLANNER and COMPUTER VISION are the threads that provide support to the STRATEGY one:

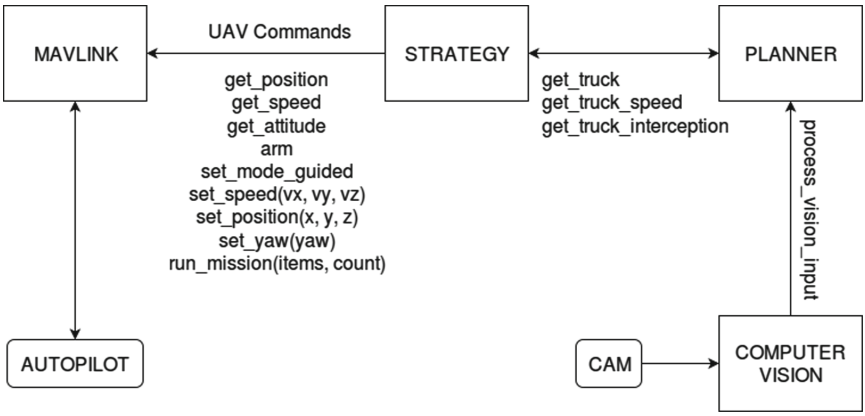

Fig. 3. Software architecture.

- The MAVLINK thread is used as an interface between the process and the autopilot. It allows translating messages from and to the autopilot through the Mavlink protocol.
- The COMPUTER VISION thread acquires and analyses images from the camera and provides the coordinates of the target to the PLANNER thread.
- The PLANNER thread is the interface between the main thread (STRATEGY) and a Finite State Machine (FSM). It receives coordinates from COMPUTER VISION in order to update the FSM and gives the position of the target over time as output to the main thread.
- The STRATEGY is the main thread and represents the decision-making module of the overall system. It has a continuous acknowledge of the state variables of both the system and the target. Its aim is to choose, in each condition, the best strategy to optimally achieve the result.

3 Dynamic Target Position Estimation

The output of the PLANNER thread consists on the estimated target position (in terms of latitude and longitude pairs) over time. This information is inferred by combining the data coming from the Computer Vision algorithms and the target trajectory estimation, which takes into account the known information about the path and the vehicle speed.

This thread is composed of the following basic software modules:

- *Target Detector* is the module handling the visual identification and tracking of the target;
- *Trajectory Predictor* is the estimator of the trajectory of the target that takes into account the (known) path and speed, and suitably adjust the position of the ground vehicle on the basis of the information given by the Target Detector.

The first module is described in depth in the following sections since it is the main objective of this paper, while the latter is briefly described here. The *Target Predictor* is a Kalman estimator that tries to determine the position of the target at each time instant. It basically implements the equation of the motion of the ground vehicle using a virtual point that drives on the path at the speed of 15 km/h. The output of the predictor is the expected Earth coordinates (latitude and longitude) of the target, information that is then used by the High-level Control to proper drive the UAV. These coordinates are continuously adjusted using data coming from the Target Detector: this module returns the center of the target, in local coordinates; a local-to-global transformation is then applied and the error between the detected and estimated coordinates is used to update the estimate. The Target Detector and the Target Predictor thus work in a tight cooperation according to the schema reported in Fig. 4.

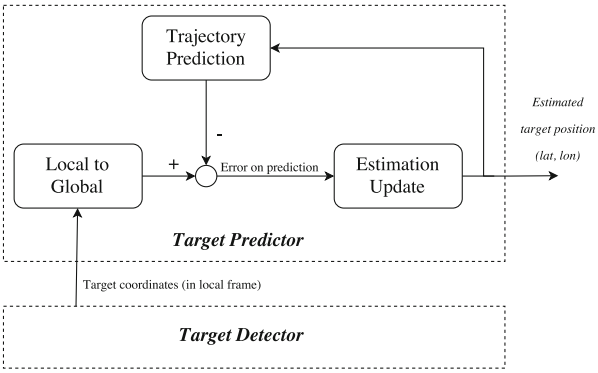

Fig. 4. Working schema of the target predictor

4 Target Detection and Tracking

The Target Detector module is aimed to detect and track the target from a live video stream. For each frame, this module provides to the system the coordinates of the target, according to the coordinate system of the camera (i.e., the target coordinates from the UAV point of view). These local coordinates are then transformed in global coordinates, referred to the global coordinate system.

4.1 Acquisition Hardware

The images processed by the Target Detector module are acquired by an Ocam camera (see Fig. 5). We selected this device due its large Field of View given by the fish-eye lens. The fish-eye lens produces a strong visual distortion in the acquired frames. Therefore, the first step of the vision module is a camera calibration aimed to perform a proper image rectification. Figure 6 an image frame acquired by the Ocam camera, in Fig. 7 the results of image rectification is shown.

Fig. 5. Exploited acquisition hardware consisting on an Ocam camera.

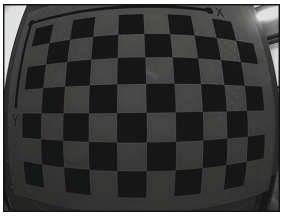

Fig. 6. Camera calibration: the chessboard pattern, with known squares dimensions, is exploited to perform the camera calibration (i.e., find the camera calibration parameters.

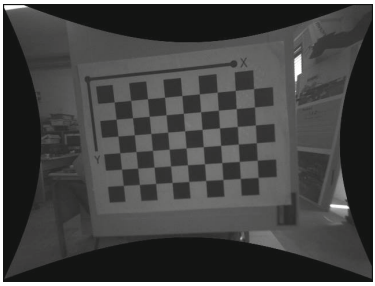

Fig. 7. Camera calibration: this figure shows the result obtained after the image rectification.

4.2 Video Analysis

The employed video analysis algorithm implements a combination of two different well known Computer Vision techniques for the detection and tracking of a known pattern. The aim of an object tracking algorithm is to estimate the trajectory of an object as it moves over time by identifying the object positions in different frames of an input video. Tracking objects can be complex depending on the application domain that can involve specific constrains. One of the main issue related to object tracking is to address with the appearance change of the target object. Generative tracking algorithms represents the target object in a specific feature space, and then perform a research of the best match within the image [17,18,21]. Discriminative tracking algorithms define a binary classification problem aimed to distinguish the target from the background [11,16,19,20]. In particular, the vision system exploits the Tracking Learning Detection (TLD) [11] algorithm to detect and continuously track the position of the target over time, considering both the vehicle and UAV movements. This algorithm implements a real-time detection and tracking of a given image pattern specified at the starting frame. In our system, the object of interest is provided by the initial detection of the target. It was possible because the TLD algorithm has been previously trained to detect the considered target. The TLD has been trained off-line, considering several target positions and distances. Furthermore, the TLD algorithm simultaneously tracks the object and learns the object appearances. As a result, the detection and tracking performance improve over time during the execution of the algorithm, allowing the system to learn from a large amount of target examples taken with huge acquisition variability. The TLD algorithm performs a fusion step, which combines the bounding box given by the tracker and the bounding box of the detector into a single output bounding box. When at least one of the two algorithms provide a bounding box, the fusion step outputs the maximally confident one, otherwise, if neither

the tracker nor the detector provides a candidate bounding box, the object is declared as not visible by the system. The whole TLD pipeline is shown in Fig. 8.

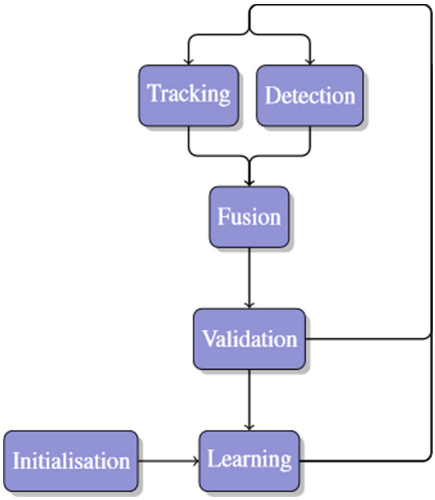

Fig. 8. Scheme of the TLD algorithm.

Once the TLD algorithm provides a bounding box containing the target, the system employs a circle Hough Transform [10] to detect a circular shaped pattern in the provided bounding box. This technique allows to find an image patch which contains an object with the shape of a circle taking into account imperfect shapes, low quality images and changes in the target pose. The aim of this step is to find the center of the target, corresponding to the center of the detected circle. The previous target detection and tracking provided by TLD gives robustness to the circle detection provided by the Hough transform. This, combined with the wide Field of View of the camera, allows to find the target and its position at almost any distance from the UAV with very high precision.

5 Basic Landing Strategy

The STRATEGY software module implements the high-level code to control the overall behavior of the UAV. The first state, that is achieved when the challenge is started, is *TAKE-OFF* and implies to drive the UAV to take-off and reach a certain starting altitude; immediately after this phase, the UAV is driven towards the center of the path[1] (i.e., the center of the eight shaped path) at an altitude of 10 m from the ground; then, the UAV waits for the passage of the ground vehicle. When the target is detected, the Target Predictor is initialized and the *intercept*

[1] The GPS coordinates of the area are known a priori.

position is computed (i.e., the position in which the vehicle can be intercepted again). When this event occurs, the UAV starts following the vehicle by tracking the target, also approaching the landing area by means of a descending path. During the approaching phase, the UAV trajectory is continuously modulated considering the output of the Target Predictor. When the landing gear touch sensors detect the successful touch-down event, it causes the turning-off of the propellers. For what concerns the Computer Vision module, when the target enter the visible area of the camera, the Detection Module performs the target detection exploiting the Hough Transform and provides the first target example to the tracker module, as well as the position of the target.

Starting from this first information, the tracker updates the position of the target over time. When the vehicle is in detected in the rectilinear part of the path, the UAV starts the landing phase. When the UAV touches the landing area, its motors are turned off.

6 Results

6.1 Simulations

Several simulations have been executed to test both the software architecture and the sub-blocks. PLANNER block has been extensively simulated in MATLAB/Simulink environment. The mission strategy has been improved by further simulations in both Gazebo (Fig. 9) and MATLAB (Fig. 10) environments by introducing the dynamical estimation of the target, to generate in real time the optimal trajectory to reach the target. The whole Software architecture has been initially simulated in Gazebo environment (Fig. 11).

6.2 On Field Trials

Several on field tests have been performed to acquire real images and data; moreover target tracking and landing on the mobile platform have been executed. Initially the videos have been acquired by using a Phantom 3 DJI UAV, and then the camera was mounted on an ASCTEC Firefly. The software architecture has been preliminary tested on a Raspberry PI board communicating to the Pixhawk autopilot and installed on two smaller UAVs (DJI F450 and DJI F550). Finally, the involved hardware and software solutions has been installed and tested on the selected DJI S900 platform. Several different trials have been also performed on the field arena concerning autonomous take-off, navigation and landing. The experiments highlighted the importance of the vision system during the target detection, tracking and the approaching of the landing area. The video of autonomous UVA in action during the MBZIRC competition is available at the following link: http://iplab.dmi.unict.it/MBZIRC/video.mp4.

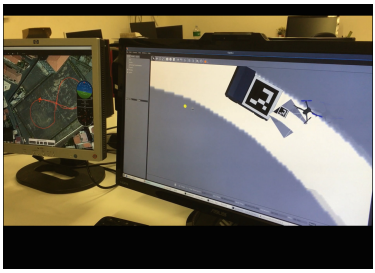

Fig. 9. GAZEBO simulations.

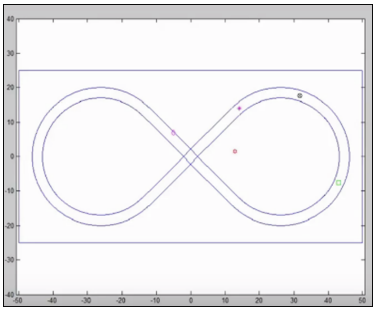

Fig. 10. MATLAB/SIMULINK simulations.

Fig. 11. S900 platform during the field trials.

7 Conclusions

The system described in this paper has been used during the above mentioned International Challenge in March 2017. The developed system reached the goal to land on the moving vehicle in 120" and the achieved result has been placed at the fourth position in the ranking of the International Challenge. The support of the visual module to the whole UAV driving system resulted a crucial factor for the achieved result during the attended competition. Indeed, all the teams that didn't exploit a vision system were unable to detect the target even when it was very close to the UAV, and often to land at all.

References

1. MBZIRC. www.mbzirc.com
2. Serra, P., Cunha, R., Hamel, T., Cabecinhas, D., Silvestre, C.: Landing of a quadrotor on a moving target using dynamic image-based visual servo control. IEEE Trans. Robot. **32**(6), 1524–1535 (2016)
3. Jin, S., Zhang, J., Shen, L., Li, T.: On-board vision autonomous landing techniques for quadrotor: a survey. In: IEEE 35th Chinese Control Conference (CCC), pp. 10284–10289, July 2016

4. Amidi, O., Kanade, T., Miller, R.: Vision-based autonomous helicopter research at Carnegie Mellon robotics institute 1991–1997. American Helicopter Society (1998)
5. Cantelli, L., Laudani, P., Melita, C.D., Muscato, G.: UAV/UGV cooperation to improve navigation capabilities of a mobile robot in unstructured environments. In: Proceedings of CLAWAR 2016, London, September 2016
6. Cantelli, L., Mangiameli, M., Melita, C.D., Muscato, G.: UAV/UGV cooperation for surveying operations in humanitarian demining. In: 11th IEEE International Symposium on Safety Security and Rescue Robotics, 21–26 October, Linkoping, Sweden (2013)
7. Cantelli, L., Lo Presti, M., Mangiameli, M., Melita, C.D., Muscato, G.: Autonomous cooperation between UAV and UGV to improve navigation and environmental monitoring in rough environments. In: 10th International Symposium on Humanitarian Demining coupled with the 11th IARP WS HUDEM2013, 23 April 2013, ibenik, Croatia (2013)
8. De Benedetti, M., D'Urso, F., Messina, F., Pappalardo, G., Santoro, C.: UAV-based aerial monitoring: a performance evaluation of a self-organising flocking algorithm. In: Proceedings of 2015 IEEE International Conference on P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC) (2015)
9. De Benedetti, M., D'Urso, F., Messina, F., Pappalardo, G., Santoro, C.: Self-Organising UAVs for wide area fault-tolerant aerial monitoring. In: WOA2015 CEUR Workshop Proceedings, Napoli, pp. 142–145, 17–19 Giugno 2015. ISSN 1613–0073
10. Hough, P.V.C.: Method and means for recognizing complex patterns. US Patent 3,069,654 (1962)
11. Kalal, Z., Mikolajczyk, K., Matas, J.: Tracking-learning-detection. IEEE Trans. Pattern Anal. Mach. Intell. 1409–1422
12. Battiato, S., et al.: An integrated system for vehicle tracking and classification. Expert Syst. Appl. **42**(21), 7263–7275 (2015)
13. Smeulders, A.W.M., et al.: Visual tracking: an experimental survey. IEEE Trans. Pattern Anal. Mach. Intell. **36**(7), 1442–1468 (2014)
14. Maggio, E., Cavallaro, A.: Video Tracking: Theory and Practice. Wiley, Hoboken (2011)
15. Yilmaz, A., Javed, O., Shah, M.: Object tracking: a survey. ACM Comput. Surv. (CSUR) **38**(4), 13 (2006)
16. Babenko, B., Yang, M.-H., Belongie, S.: Robust object tracking with online multiple instance learning. IEEE Trans. Pattern Anal. Mach. Intell. **33**(8), 1619–1632 (2011)
17. Bao, C., et al.: Real time robust l1 tracker using accelerated proximal gradient approach. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE (2012)
18. Black, M.J., Jepson, A.D.: Eigentracking: robust matching and tracking of articulated objects using a view-based representation. Int. J. Comput. Vis. **26**(1), 63–84 (1998)
19. Grabner, H., Leistner, C., Bischof, H.: Semi-supervised on-line boosting for robust tracking. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008. LNCS, vol. 5302, pp. 234–247. Springer, Heidelberg (2008). doi:10.1007/978-3-540-88682-2_19
20. Hare, S., et al.: Struck: structured output tracking with kernels. IEEE Trans. Pattern Anal. Mach. Intell. **38**(10), 2096–2109 (2016)
21. Kwon, J., Lee, K.M.: Visual tracking decomposition. In: 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE (2010)