DART Project: A High Precision UAV Prototype Exploiting On-board Visual Sensing

Michele Basso, Luca Bigazzi, Giacomo Innocenti

Dipartimento di Ingegneria dell'Informazione
Università degli Studi di Firenze
via S. Marta 3, 50139, Firenze, Italy.
Email: {michele.basso,giacomo.innocenti}@unifi.it

***Abstract*—This work explores a way to achieve high precision in the positioning of a 250-class aerial drone by means of only on-board sensors. The proposed technology is still in development, and its basic idea is to compensate the errors of the sensors by fusing together strongly correlated data streams. The main players are a 6 Degrees of Freedom (DoF) Inertial Measurement Unit (IMU) and a computer vision system, arranged to work together as a "virtual sensor" providing the pose of the drone relative to one or more markers acting as reference points of known position and orientation. The proposed advance sensing exploits complementary filters to merge inertial and visual data. Such a refined positioning is then used to feed a custom control strategy acting as auto-pilot for implementing autonomous navigation. Preliminary results on the developed technologies are reported.**

Keywords–Advance sensing; Drone; Computer Vision; Sensor Fusion.

I. INTRODUCTION

Thanks to their versatility, small drones like multirotor Unmanned Aerial Vehicles (UAVs) have received more and more interest over the last few years, both in the academic and industrial communities. In the scientific literature the use of drones is getting very popular and applications are increasing in pace with the technological development of these systems. Moreover, recent advances in microelectronics have made single-board microcontrollers and embedded systems economically affordable, making their use widespread to add advanced features in many small and medium-sized systems such as drones. In the context of UAV systems, for example, the availability of these advanced single-board computers is important in applications where extreme positioning precision is required. In this regard, interesting applications that have been proposed recently concern precision agriculture, where drones equipped with Real-Time Kinematic Global Navigation Satellite Systems (GNSSs-RTK) [1] are used to minimize human intervention [2] [3]. Other recent usage scenarios see drones equipped with specific scientific equipment, for example for the reconstruction of 3D environments through LIDAR (Laser Remote Sensing) techniques [4], or for monitoring road traffic [5]. These systems often use Vision-Based Navigation (VBN) algorithms to improve positioning accuracy [6]–[8]. However, currently in the scientific literature there are no direct references to the precision achieved by these multirotor systems, since in almost all of today's applications high precision specifications are not required, while for applications that require it, alternative solutions are sought. In particular, at the current technological level there are no drones capable of following trajectories with sub-centimetric precision and this precludes the use of these drones in many potential novel applications.

The main purpose of this work consists in the development of a high-precision positioning system for UAV multirotors, which makes these devices usable in a wide class of applications where high accuracy is the main requirement. To increase the level of accuracy of the currently available solutions, the proposed approach integrates and extends the performance of computer vision and inertial sensor navigation techniques, where position, velocity and attitude data extracted simultaneously from video streams and inertial sensors are merged together by filter fusion algorithms for error compensation. The above task involves the development of software and hardware dedicated to video processing. In particular, by exploiting a single camera mounted on-board, the main task is based on the identification of specific markers in the environment, from which it is possible to extract with high precision the information of attitude and position relative to the drone. Through appropriate algorithms, it is then possible to determine in real-time the 3D coordinates of the drone with respect to the marker, which can be used as a position virtual sensor for controlling the drone trajectory. This visual sensor has been designed to be employed as an on-board autonomous navigation system add-on for commercial drones. Indeed, the developed prototype has been implemented on a low-cost hardware board (Raspberry PI), which is able to process video and inertial sensor data to autonomously pilot a 250-class multirotor for tracking specific trajectories.

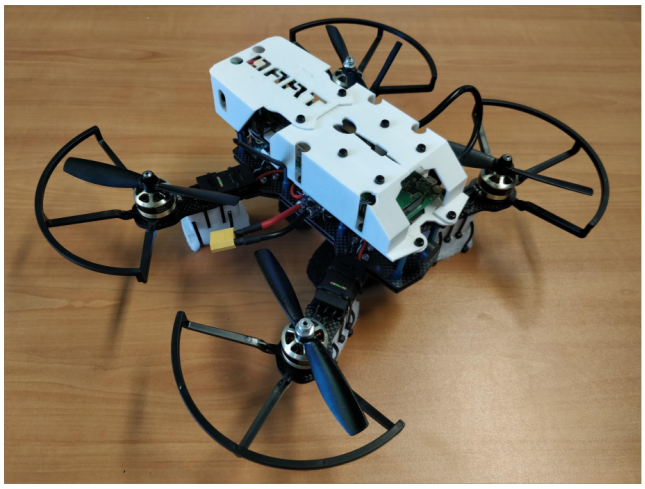
The rest of the paper is structured as follows. In Section II, the hardware and software architectures are illustrated. In Section III, the computer vision technique developed for the advanced sensing of the drone's position is presented. We conclude the paper in Section IV reporting some preliminary results.

II. PROTOTYPE ARCHITECTURE

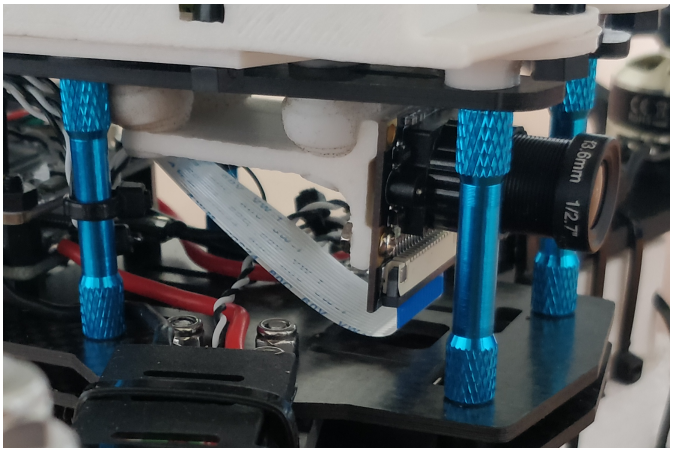
A. Hardware

The DART prototype is a 250-class aerial drone developed at the Systems & Control Lab of the Department of Information Engineering of the University of Florence (see Figure 1). The drone features a flight controller CC3D Revo running the open-source LibrePilot software as low-level interface to the hardware. This very basic architecture has been extended by

 ISBN: 978-1-61208-712-2



(a)



(b)

Figure 1. (a) DART prototype. (b) Zoom of the camera suspension support.

two additional board: A Raspberry PI 3 B+ and a Arduino nano. The first board is equipped with an Pololu 2739 IMU and a Raspicam camera module, and it implements the autonomous driving commands based on image processing and data from the onboard IMU. The second board, instead, implements a Pulse Position Modulation mixer (PPM-mixer) between the commands coming from the 2.4 GHz receiver, and those generated by the autonomous driving module. The mixer allows also for hybrid driving modes, and it is responsible of the safety during transitions from manual to autonomous flight modes and vice-versa. Figure 2 illustrates the related functional scheme. The Raspberry module communicates with the PPM-mixer via a bidirectional Universal Asynchronous Receiver Transmitter (UART) protocol, while the commands from the receiver arrive to the Arduino nano via its native PPM protocol. PPM is also the kind of signal expected by the LibrePilot interface, and so this has to be the nature of the mixer output, thus explaining its name. Regarding the commands from the receiver, then, the mixer operates just as a pass-through, while the commands from the autonomous driving module have to be transformed into proper PPM signals. In the Raspberry board, API V4L2 are exploited to handle the signals from the connector of the camera, whereas data from the IMU arrive via I2C bus.

B. Software

All the necessary software runs on the three boards described in the previous subsection according to the following scheme. The CC3D Revo board executes the LibrePilot software [9], which provides access to the drone IMU sensors and motors, thus acting as low-level interface to each single device on the drone. The Raspberry module, instead, executes the software for image processing, and it is also responsible for generating the commands of the autonomous driving. In particular, the developed computer vision software is based on OpenCV [10] and Visp [11] libraries. In the proposed version, the computer vision software is designed to recognize special markers. Figure 3 depicts the result of the image processing that allows the module to detect the marker, and to compute its orientation. The autonomous driver implements an in-house control algorithm, described in more details later. The Arduino nano board, finally, runs UART and PPM protocols specifically designed for it objective function, i.e. the hardware switch.

C. Control algorithm

The autonomous driving module computes the commands in the same form of those coming from the receiver, i.e., as proper reference values for roll, pitch, yaw and thrust. At this stage of the project, they are simply conceived to have the drone maintaining a desired position $\overline{Q}$ with a zero yaw angle $\overline{\psi} = 0$, such that the drone is facing the marker. The image processing module provides both relative orientation and relative position with respect to the marker. This information is further integrated by means of a fusion algorithm with data coming from the onboard IMU to improve the estimate of the image processing. It is worth stressing that with the implemented algorithm and hardware platform the image processing module works at about 30-40 Hz, while the IMU can provide data at higher frequency. Therefore, the sensor fusion algorithm also synchronizes the two different information streams in order to use the right samples from each sensor. The final result is a refined estimate of the drone position Q with respect to the marker, whose details will be described in the next section. The extension to multiple markers is still under development, but early results are promising.

The information on the drone pose errors $(\overline{Q} - Q)$ and $(0 - \psi)$ are used to feed four distinct controllers, which generate the driving commands as references for roll and pitch angles, yaw angular velocity, and thrust. Such a preliminary solution is not expected to provide the best performance, since it does not consider the mutual connections between the drone pose components, but it allows one to design the autonomous driver by composition of simpler modules. The resulting control architecture is illustrated in Figure 4.

III. Advanced Sensing

A. Virtual sensor

The vision algorithm takes care of detecting one or more known markers present in the environment in order to obtain their poses (positions and orientations) with respect to the camera reference frame (body frame). To achieve this task, the algorithm computes the gradient for each pixel of the image such that pixels with similar gradient directions are grouped into sets. The latter sets identify edges in the image from which

 ISBN: 978-1-61208-712-2

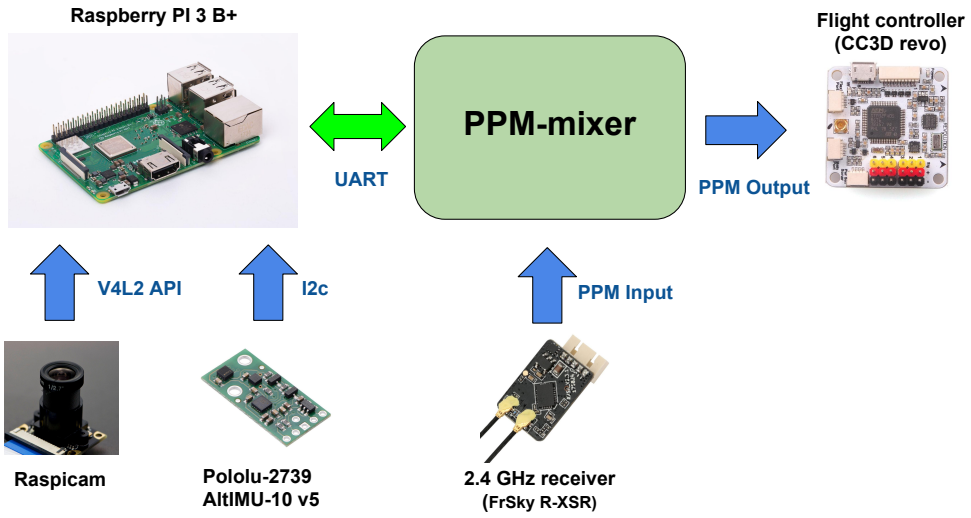


Figure 2. Hardware configuration and dependencies.

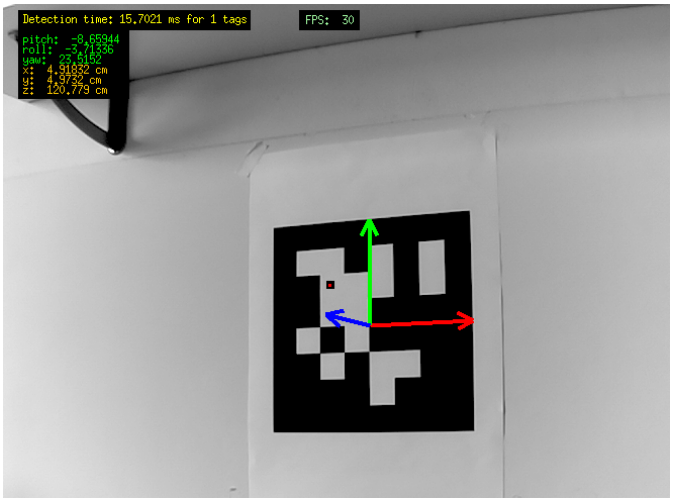


Figure 3. Marker as seen from the image processing module.

the algorithm searches for the correct sequence that recognizes the two-dimensional position (u, v) of each marker in pixel coordinates.

The marker coordinates (x_m, y_m, z_m) in body frame can be computed through the relations for barrel distortion

$$y_m/z_m = \frac{(v - v_0)(1 + k_{ud}R^2)}{p_y} \tag{1}$$

where

$$R^2 = \frac{(u - u_0)^2}{p_x^2} + \frac{(v - v_0)^2}{p_y^2}, \tag{2}$$

(u_0, v_0) represents pixel coordinates of the image center, whereas parameters p_x e p_y are the focal length to pixel dimension ratios. Finally, k_{ud} and k_{du} are parameters needed to correct lens distortions. Therefore, k_{ud}, k_{du}, p_x and p_y are intrinsic camera parameters which can be obtained through an iterative evaluation process that involves the acquisition of frames of a known image in different poses.

If the geometrical properties of the marker are known, it is possible to retrieve additional information such as the distance z_m and orientation of the marker itself which, in turn, provide the full pose of the marker frame with respect to the body or camera frame. Indeed, the following change of coordinates provide the drone 3D-position Q with respect to the marker frame

$$Q = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = R_{XYZ}(\Phi) \begin{bmatrix} x_m \\ y_m \\ z_m \end{bmatrix} \tag{3}$$

where

$$\Phi = \begin{bmatrix} \varphi \\ \theta \\ \psi \end{bmatrix} \tag{4}$$

is the vector of the roll, pitch and yaw angles between the frames (computed in the experiments by using the Homography method), and

$$R_{XYZ}(\Phi) = \begin{bmatrix} c_\varphi c_\psi & c_\varphi s_\psi s_\theta + s_\varphi c_\theta & -c_\varphi s_\psi c_\theta + s_\varphi s_\theta \\ -s_\varphi c_\psi & -s_\varphi s_\psi s_\theta + c_\varphi c_\theta & s_\varphi s_\psi c_\theta + c_\varphi s_\theta \\ s_\psi & -c_\psi s_\theta & c_\psi c_\theta \end{bmatrix}$$

is the corresponding rotation matrix.

B. *Sensor fusion*

The use of an addition 6-DoF IMU makes it possible to merge the information of the drone attitude with the estimate provided by the vision, considerably increasing performance, especially at high frequencies.

At first, the algorithm running on the software platform prefilters both the attitude and position data coming from the vision and the angular velocities measured by the IMU gyroscope. Then, to improve the estimation accuracy, at each iteration k the attitude vector Φ_k computed by the vision system is fused with the angular velocities Ω_k measured by the gyroscope, through the use of the following first order complementary filter

$$\hat{\Phi}_{k+1} = \rho(\hat{\Phi}_k + T_k\Omega_k) + (1 - \rho)\Phi_k \tag{5}$$

 ISBN: 978-1-61208-712-2

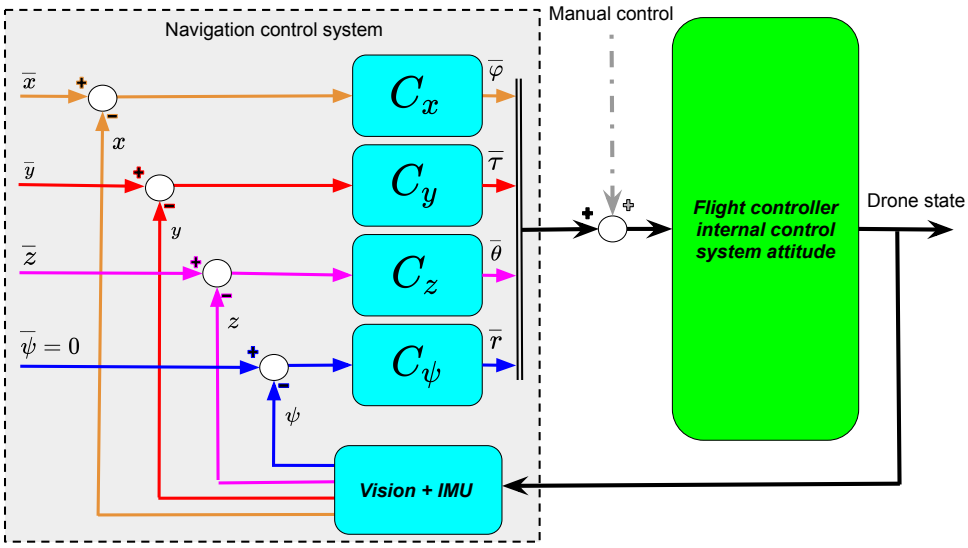


Figure 4. Architecture of the autonomous driving module. Blue blocks stands for custom made functions, while the green block represents the CC3D Revo board programmed with the LibrePilot software.

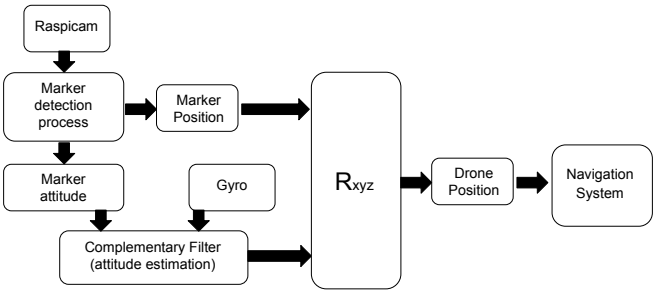


Figure 5. Schematic of the estimation process for the drone position and orientation.

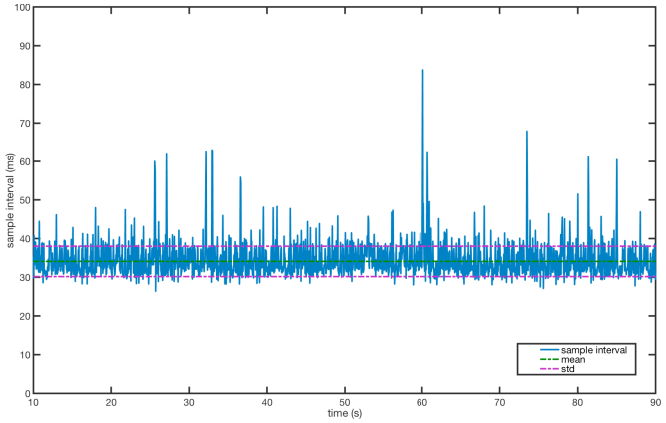


Figure 6. Measured sample interval of the navigation system.

where T_k is the actual sample interval and

$$\Omega_k = \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{6}$$

is the vector of the angular velocities around the three main drone axes. The new orientation estimate $\hat{\Phi}$ of equation (5) can now be employed in the coordinates transformation (3) providing a refined position estimate Q. The schematic of the complete filtering and estimation process is depicted in Figure 5.

IV. Conclusion and Future Developments

In this section, some preliminary results will be reported and commented. A depiction of the next development of the project will be illustrated, as well.

Figure 6 depicts how the sampling time varies during a test flight. It is worth observing that its average is around the already mentioned 34ms (i.e., about 30Hz) with a sufficiently narrow standard deviation less than 2.5ms. Nevertheless, since the control board in not meant for real-time computing, the varying computational burden and the variable power supply to the processor generate relatively small oscillation of the sampling time. In Figure 7, the trend of positions x, y, and z are shown as they evolve during a test flight in auto-piloting mode. The experiment is meant to provide a glimpse of the output coming from the virtual sensor based on computer vision techniques. The comparison with the real position of the drone would require a special environment (such as a set of ground cameras), which will be developed later in the project to assess the final result quality. Still, a simple visual inspection of the diagrams suggests that the virtual sensor may be able to catch sufficiently rapid variations of the position without introducing relevant noise. In this respect, Figure 8 reports the position estimate while the drone stands still on a test bench. Since the position is constant, the plot shows the trend of the estimation errors and it provides a good graphical representation of the system intrinsic noise (appraisable in only few centimeters for a 2.72m distant marker) and its related power. Looking at Figure 8, the result turns out particularly encouraging once the absence of a gimball for the camera is stressed, since such a solution would strongly reduce the

 ISBN: 978-1-61208-712-2

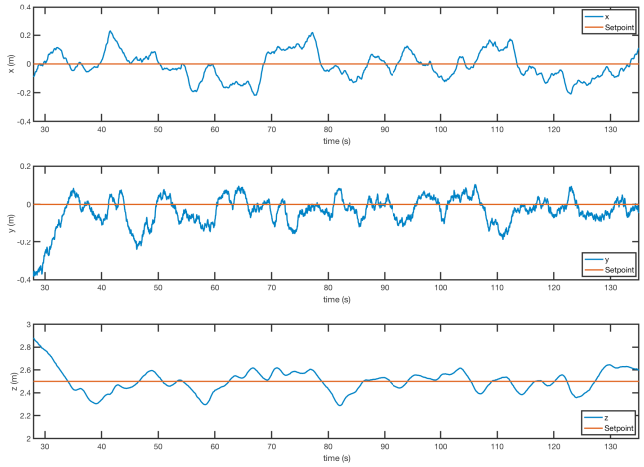


Figure 7. Drone detected position during an autonomous flight.

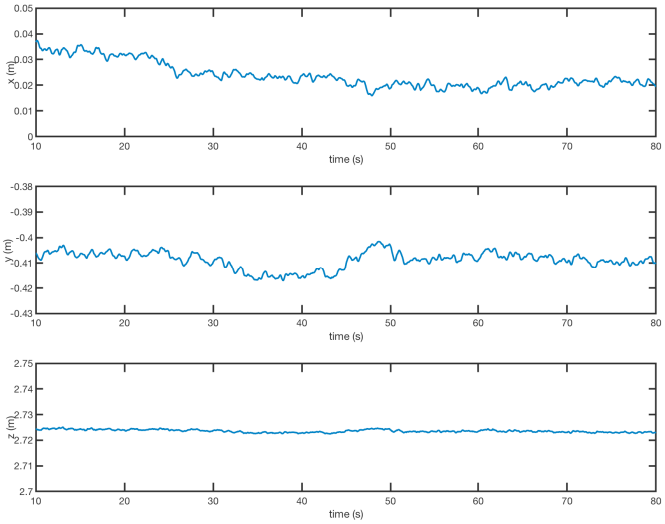


Figure 8. Drone detected position in static conditions.

compensation needed to refine the pose estimate. The addition of a gimball is planned as one of the next improvements.

Other tests have already been planned to check the reliability of the proposed technology up to this stage of the project. Experiments aimed at testing the auto piloting functionality, instead, are scheduled later on, because they are more complicated to perform due to the many precautions needed to ensure the drone safety during an autonomous flight. Most likely, better performance could be achieved by using model based controllers. Therefore, an accurate physical model of the drone will be top priority for the continuation of the project.

REFERENCES

[1] P. Henkel, U. Mittmann, and M. Iafrancesco, "Real-time kinematic positioning with GPS and GLONASS," in 2016 24th European Signal Processing Conference (EUSIPCO), Aug 2016, pp. 1063–1067.

[2] P. Tokekar, J. V. Hook, D. Mulla, and V. Isler, "Sensor planning for a symbiotic UAV and UGV system for precision agriculture," IEEE Transactions on Robotics, vol. 32, no. 6, Dec 2016, pp. 1498–1511.

[3] B. Reshma and S. S. Kumar, "Precision aquaculture drone algorithm for delivery in sea cages," in 2016 IEEE International Conference on Engineering and Technology (ICETECH), March 2016, pp. 1264–1270.

[4] M. Sanfourche, B. Le Saux, A. Plyer, and G. Le Besnerais, "Environment mapping & interpretation by drone," in 2015 Joint Urban Remote Sensing Event (JURSE), March 2015, pp. 1–4.

[5] H. Zhou, H. Kong, L. Wei, D. Creighton, and S. Nahavandi, "Efficient road detection and tracking for unmanned aerial vehicle," IEEE Transactions on Intelligent Transportation Systems, vol. 16, no. 1, Feb 2015, pp. 297–309.

[6] F. Samadzadegan and G. Abdi, "Autonomous navigation of unmanned aerial vehicles based on multi-sensor data fusion," in 20th Iranian Conference on Electrical Engineering (ICEE2012), May 2012, pp. 868–873.

[7] A. G. Kendall, N. N. Salvapantula, and K. A. Stol, "On-board object tracking control of a quadcopter with monocular vision," in 2014 International Conference on Unmanned Aircraft Systems (ICUAS), May 2014, pp. 404–411.

[8] P. Rudol, M. Wzorek, and P. Doherty, "Vision-based pose estimation for autonomous indoor navigation of micro-scale unmanned aircraft systems," in 2010 IEEE International Conference on Robotics and Automation, May 2010, pp. 1913–1920.

[9] "LibrePilot open source project," 2019, URL: https://www.librepilot.org [accessed: 2019-04-02].

[10] "Open Source Computer Vision Library," 2019, URL: https://opencv.org/ [accessed: 2019-04-02].

[11] "Open source visual servoing platform library," 2019, URL: https://visp.inria.fr/ [accessed: 2019-04-02].

 ISBN: 978-1-61208-712-2