An Agile Low-cost Testbed for Multi-Drone Target Tracking

J.M. Amador, J.R. Martínez-de Dios, J.L. Paneque and A. Ollero

***Abstract*— Most existing aerial robotics testbeds have significant set-up and maintenance costs that restrict their use. This paper presents a low-cost multi-drone testbed for agile and safe performing proof-of-concept experiments. It uses low-weight, low-size *Crazyflies* drones and low-cost lighthouse positioning system for drone pose estimations. Its flexible and modular ROS-based architecture can be used to easily test multi-drone perception, control and planning techniques and can easily integrate bigger drones. The presented testbed does not require a dedicated room and can be set-up in less than one hour. The paper presents the testbed and validates its operation with multi-drone target tracking experiments.**

***Index Terms*— UAS Testbeds; Micro- and Mini- UAS; Sensor Fusion**

I. Introduction

The increasing interest in aerial robotics technologies has motivated the need for tools that enable the quick development of proof-of-concept developments or tests. Several open software frameworks, libraries and tools are aligned with this tendency offering a hugely wide range of compatible software tools ready to use or to integrate in larger projects with very low or no cost and involving low development efforts. ROS [1] or OpenCV [2] are good examples of these tools. However, despite these efforts in the software components, there is still lack of low-cost agile hardware tools for aerial robots testing and experimentation.

Most aerial robotics laboratories around the world are equipped with drone testbeds, most of them are indoor testbeds. However, the implementation and maintenance of these testbeds is complex and expensive. They often need a dedicated room and a motion capture system capable of providing drone pose estimation with very high accuracy and very low delay. These are significant constraints for some small and medium companies, which are forced to rent drone testbeds, or to educational centers, which often have to refuse using them for teaching in undergraduate or Master degree courses.

This paper presents a low-cost multi-drone testbed for agile and safe performing proof-of-concept experiments. It uses low-cost, low-size, low-weight *Crazyflie* platforms, which are easy-to-use drones with simple and flexible hardware. The testbed uses a lighthouse positioning drone system, a low-cost system capable of providing local poses with accuracies in the range of cms with sufficiently low delay. The presented testbed uses a flexible and modular architecture based on ROS that can be used to easily test multi-drone perception, control and planning techniques. Besides, it can integrate bigger drones by simply changing the motor drivers and frames, without modifying any other module of the testbed modules. The testbed can be easily and quickly deployed and calibrated without requiring a dedicated room. Several papers have used *Crazyflies* to experiment the technique that they are presenting, see e.g. [3]. However, to the best of the author's knowledge, this is the first low-cost agile general-purpose multi-drone testbed that has been reported. We believe that the presented testbed is useful for quickly performing proof-of-concept tests. It is also a low-cost alternative to traditional indoor testbeds for small companies and can also be used for educational purposes in undergraduate and Master courses.

This paper is structured as follows. Section II summarizes the main types of UAS testbeds that have been developed. The architecture of the presented testbed is briefly presented in Sec-

This work has received funding from the EU H2020 Research and Innovation Programme under grant agreement No 731667 (MULTIDRONE).

The authors are with the GRVC Robotics Lab, University of Seville, Seville 41092, Spain `email: {jamador, jdedios, jlpaneque, aollero}@us.es`

tion III. The main hardware components are described in Section IV. Section V briefly describes the multi-drone target tracking technique implemented, which proof-of-concept experiment in the testbed is summarized in Section VI. Section VII presents the main conclusions and future work.

II. Related Work

Many testbeds for experimentation with UAS have been developed, see e.g. [4], [5]. They can be coarsely classified in indoor and outdoor testbeds. Most research-focused indoor testbeds employ expensive infrastructure, such as accurate positioning systems such as VICON or OptiTrack [6], [7]. These systems require a dedicated space and have high purchase and maintenance cost, often requiering personnel for its maintenance. There are also outdoor testbeds that are used for experimenting with large drones that need to fly in open areas [8]. Although outdoor versions of camera-based motion capture systems have been developed, their high cost for covering large scenarios recommend the use of localization systems, such as accurate RTK GPS systems. The cost, maintenance, risks and safety procedures when working with big drones make them unsuitable for performing quick proof-of-concept experiments.

The indoor testbed presented in this paper obtains accurate UAS pose estimations using the lighthouse positioning system. These systems have been recently used in Virtual Reality (VR) positioning, e.g. [9]. Similarly to camera-based systems such as VICON, it is necessary to set up the lighthouse base stations around the drone flying arena. However, the sensing principle is the opposite: while in camera systems the light from drones are received by the cameras, in lighthouse systems each base station generates sweeps of infrared light that is received by the light receivers onboard the drones. The good accuracy of lighthouse positioning systems has been analyzed in several works, see e.g. [10]. For instance, works such as [11] indicate some small modifications to further improve their pose estimation repeatability and accuracy. In fact, lighthouse systems have been used to provide ground truth pose estimations in VR systems.

In the presented multi-drone testbed we propose the use of lighthouse 1 positioning systems to provide the ground truth drone pose estimates. The big advantage of lighthouse positioning systems is their cost: which is hundredths lower than those of camera-based systems such as VICON. Another advantage is that the pose estimation is calculated on-board each drone, whereas in camera-based systems it is computed in a ground computer. Hence, each drone can perform flight stabilization and trajectory control with lower delay compared to external visual positioning systems.

III. Architecture

The testbed presented in this paper has been developed to address a threefold objective. Firstly, it is low-cost and save, therefore can be used for agile testing of techniques by users that do not necessarily have to be experts. From a scientific perspective, it can be used as a tool to test proof-of-concept developments. From an application point of view, it can be used as an initial step in the development of techniques before testing in the real application. The same experiment can be repeated many times with the same or different conditions and configurations to assess its performance.

The main design requirements can be summarized in the following: *safety* to allow users with low expertise and know-how levels; *low cost*; *openness and flexibility* to allow testing different perception, control and planning techniques; *ROS compatibility* to enable testing ROS-based techniques; *scalability* to enable using large number of drones in the same experiment.

Figure 1 shows a simplified scheme of the architecture of the presented testbed. Although only one drone is shown in Fig. 1, the architecture can include a high number of drones. Two main types of modules can be found. Those that are executed in the Central Computer and those that are executed on-board each of the drones.

The testbed uses the *Crazyflie 2* multirotor platforms developed by *Bitcrazy*, which have been successfully used in various projects and researches, see e.g. [12]. *Crazyflie* multirotors are open and flexible platforms that can be easily hardware and software customized. In our testbed we adopted this aerial platform and added some hardware and software modules to provide the testbed with higher capabilities for performing higher number of experiments.

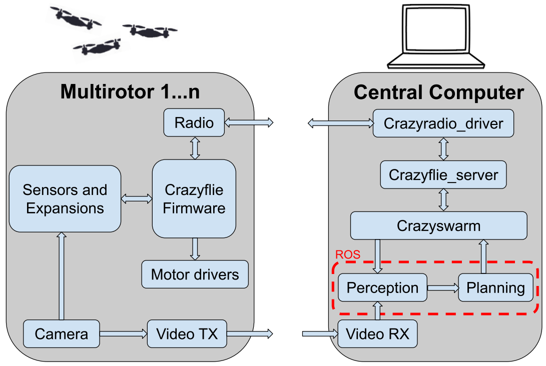

Fig. 1. Simplified scheme of the architecture of the presented testbed.

The modules running in the multirotors include the *Crazyflie firmware*, *radio*, *sensors and expansions*, *motor drivers* and optionally, a *camera* and the *video transmitter*. The *Crazyflie firmware* is the core of the drone: it executes the low level control and communicates with other electronics boards and sensors. The radio allows the multirotor to receive internal parameters and commands, and to transmits the drone state and sensor measurements. The motor driver controls directly the currents given to the motors. It can be noticed that the testbed enables using bigger drones by modifying the motor drivers and without modifying any other module. *Crazyflies* use direct-current micro motors. These motors can be substituted by more powerful three-phase UAS motors by adding also a variable frecuency drive to control these motors. Hence, bigger multirotor frames with higher payload can be used in the testbed without any firmware or software change, only changing the drone physical parameters.

The *sensors and expansion* module includes an IMU and a lighthouse deck for pose estimation and a pressure sensor. We added a small visual camera. However, the *Crazyflie* microprocessor does not have high processing capacity, and only can process low resolution images at low frame rate. To handle high resolution cameras it is necessary to add: 1) a electronic board to include onboard image processing capabilities or 2) a video transmitter to send the images to the *ROS Computer*, where the image processing algorithms are performed.

The testbed includes a Central Computer running ROS. It includes software modules that can be flexibly changed and modified for each experiment. Modules *Crazyradio driver* and *Crazyflie server* manage the USB drivers and communication protocols to share data and commands with the multirotors. *Crazyswarm* [12] provides ROS topics and services to share the data and commands with multiple multirotors using standard ROS nodes. It also includes user-defined modules *Perception* and *Planning* where an user can program the techniques to be experimented. These modules have access to the multirotor sensor measurements and states and can be used to command the drones, for instance to follow a trajectory or perform more complex actions. Finally, the Central Computer is connected to the *Emergency Stop* module, which is a physical switch that stops all the motors of every drone in the system.

IV. Hardware

Figure 2 shows a picture of the presented testbed deployed in an improvised 2x2 space, evidencing the testbed flexibility. The *HTC VIVE* base stations can be mounted on tripods, and Central Computer can be a laptop. The calibration of the lighthouse localization system is also very quick, enabling quick and easy set-up.

This section summarizes the main hardware components of the presented testbed.

A. *Crazyflie 2 multirotor*

Crazyflie 2 platforms are cost-efficient, safe (low weight, low size) and flexible platforms with open-source software tools. Figure 3 shows three of the drones used in the experiments. One basic *Crazyflie 2* platform is shown in Figure 3-left. Figure 3-center shows one platform equipped with the lighthouse deck used for pose estimation. The lighthouse deck has four IR receivers and its own FPGA to process the IR received signals and obtain the position estimate of each IR receiver on the board. One of the platforms used in the reported experiments is shown in Figure 3-right. It is equipped with the lighthouse deck and an ultra lightweight nano-camera. A detail picture is shown in Figure 4. Figure 3 also shows the *Crazyradio*, a $2.4GHz$ transceptor with an USB interface to teleoperate the *Crazyflie* drones. Each *Crazyradio* is able to control a group of *Crazyflies*, depending

Fig. 2. The presented tested installed in an improvised 2x2 space.

on the amount of data transmitted. Using only 3 *Crazyradio*, 49 *Crazyflies* are operated in [12].

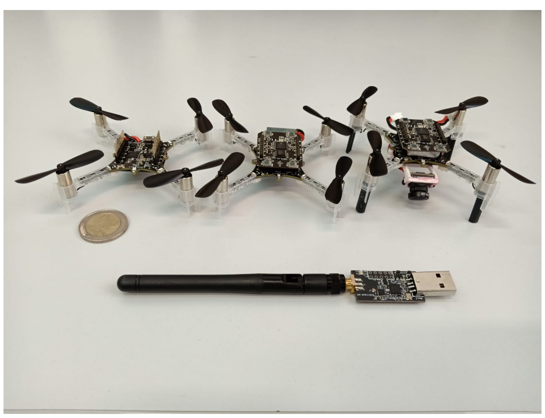

Fig. 3. *Crazyflie* platforms and *Crazyradio* used in the presented testbed.

The simple hardware design of *Crazyflies*, in which the electronic board is the main part of the multirotor frame, allows these nano-drones to fly carrying several lightweight onboard sensors. This fact enables these multirotors being a fairly powerful flying testing devices of only $35g$ (including the lighthouse deck) suitable for the presented testbed. A *Crazyflie* equipped with a lighthouse deck is able to fly for around 6-7 minutes. The use of a nano-camera increases the energy consumption and reduces the flight time to 4-5 minutes, which it is still enough to perform many proof-of-concept experiments.

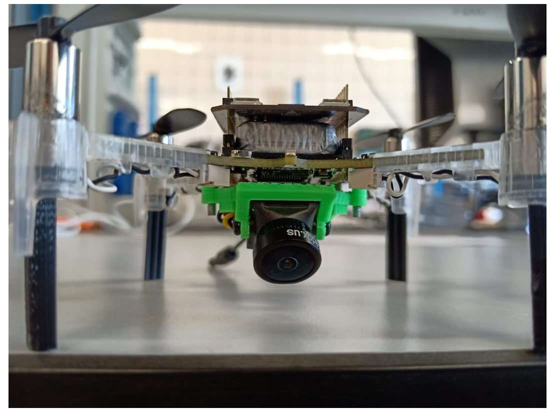

Fig. 4. A detail picture of one *Crazyflie* drone used in the experiment. It is equipped with the lighthouse deck and a nano-camera.

In the testbed we used as firmware the *Crazyflie* master branch modified to obtain drone orientation using the lighthouse system and not only its IMU. We updated the master branch with an SVD (Singular Value Descomposition) optimization algorithm based on [13] in order to obtain the drone orientation. SVD optimization was chosen due to the accuracy and stability of the algorithm as shown in [14].

B. Lighthouse tracking system

Currently, the only commercial lighthouse system is the *HTC VIVE* virtual reality system [9]. The *HTC VIVE* system is comprised of lighthouse decks, which emit synchronized light sweeps, and trackers that use IR receivers to measure light pulse timings for estimating the horizontal and vertical angles to the lighthouse decks. The lighthouse deck on-board the *Crazyflies* has four IR receivers. The *Crazyflies* uses the angles obtained by each IR receiver to obtain its pose employing the SVD optimization algorithm to obtain the rotation matrix of the drone pose.

Figure 5 shows the *HTC VIVE* base stations and the tracker. The tracker is necessary only for the calibration of the *HTC VIVE* system. *HTC VIVE* base stations need to synchronize their IR sweeps. In outdoor experiments wireless synchronization can fail: some works recommend the use of a synchronization cable for outdoor experiments. In our indoor experiments we did not notice such issues and wireless synchronization worked satisfactorily.

Fig. 5. *HTC VIVE Base Stations* (left and right) and *Tracker* (center).

It is recommended to mount the base stations maintaining $90°$ between them in order to obtain the best ground truth drone pose estimations. Typically, they are mounted at $45°$ with the floor plane and in two top faced edges of the room, in order to avoid occlusions and to get good pose estimations. The maximum distance between both base stations of the testbed is 5 m, limiting the maximum size of the testbed.

V. Multi-drone target tracking

The presented testbed allows performing fairly realistic proof-of-concept experiments. This section summarizes a multi-drone target tracking method that has been experimented to show its capabilities.

The requirements of the target tracking method can be summarized in the following: low computational burden, robustness to noise and data loss, distributed computation and low communications burden. We adopted a scheme based on Recursive Bayesian Filters (RBFs) to integrate the camera measurements from all the drones involved. We selected an Information Filter (IF) due to its computational efficiency and numerical stability. The target tracking method that is experimented in the testbed is a simplified extension of that described in [15].

Information Filters (IFs) are parametric RBFs that employ the canonical Gaussian representation, which is based on the information vector $\xi = \Omega^{-1}\mu$ and the information matrix $\Omega = \Omega^{-1}$, where μ and Ω are the mean and variance of the distribution. IFs are dual to Kalman Filters (KFs) and are more suitable for cases with simple prediction model and high number of measurements, as in our case.

We selected a simple state vector typical in target tracking, $\mathbf{x}_t = [x_t, y_t, z_t, vx_t, vy_t, vz_t]^T$, where (x_t, y_t, z_t) is the 3D target location at time t and (vx_t, vy_t, vz_t) is its local velocities. IFs require a prediction model and a measurement model, both assumed under White Gaussian noise with covariance matrices R_t and Q_t. For the prediction we used a simple rectilinear motion model to represent the target local motion.

The observations are the coordinates of the center of the target in the image plane of the camera. Let $z_{j,t}$ be the measurement gathered by camera j at time t:

$$z_{j,t} = \begin{bmatrix} x_{j,t} \\ y_{j,t} \end{bmatrix} = h_j(\mathbf{x}_t), \quad (1)$$

where $h_j(\mathbf{x}_t)$ is the observation model obtained using the camera pin-hole model. $h_j(\mathbf{x}_t)$ is a non-linear function. Thus, we need to use the Extended Information Filter (EIF), which linearizes $h_j(\mathbf{x}_t)$ by its Jacobian:

$$H_{j,t} = \frac{\partial h_j(\mathbf{x}_t)}{\partial \mathbf{x}_t} \quad (2)$$

One advantage of IFs over KFs is their capability to integrate measurements obtained in a distributed manner, e.g. by different drones. The Prediction stage of the IF must be executed in one central drone but the Update stage can be performed in a fully distributed way. We implemented a distributed version of the EIF in which each drone computes its contribution to the EIF and sends it the central drone, which integrates all the contributions and computes the Prediction stage. Although the detailed description of the implemented method is omitted for brevity, the following paragraphs summarize the performance of the implemented EIF.

The operation of the EIF for time t is as follows. The predicted state for time t, $\bar{\Omega}_t$, $\bar{\xi}_t$ and $\bar{\mu}_t$, is assumed available. It was computed in the iteration at $t-1$ by the central drone. At time t the central drone first broadcasts the predicted state $\bar{\mu}_t$. Each drone j that receives the prediction computes its Jacobian $H_{j,t}$, takes a measurement $z_{j,t}$ from its camera and computes its contribution to the EIF update stage ($\Omega_{j,t}$ and $\xi_{j,t}$) as follows:

$$\Omega_{j,t} = H_{j,t}^T Q_{j,t}^{-1} H_{j,t} \tag{3}$$

$$\xi_{j,t} = H_{j,t}^T Q_{j,t} [z_{j,t} - h_j(\bar{\mu} + H_{j,t}\bar{\mu})] \tag{4}$$

The central drone i is also equipped with a camera: it also computes its Jacobian $H_{i,t}$, takes a measurement $z_{i,t}$ and computes its contribution to the EIF Update. Next, each drone transmits to the central drone its contribution $\Omega_{j,t}$ and $\xi_{j,t}$. The central drone receives the packets, extracts the contributions and computes the updated state by adding the contributions from all drones (including its own contribution), as follows:

$$\Omega_t = \bar{\Omega}_t + \Omega_{i,t} + \sum_j \Omega_{j,t} \tag{5}$$

$$\xi_t = \bar{\xi}_t + \xi_{i,j} + \sum_j \xi_{j,t} \tag{6}$$

The measurement update stage in IFs is achieved by summing up the contributions from different cameras. Each drone broadcasts the predicted state. The rest of the drones compute its contribution to the EIF update and transmit it back to the source, which only has to add the contributions it receives in order to recover the updated state.

VI. Experiments

This section presents how the presented testbed can be used to perform multi-drone target tracking proof-of-concept experiments. The experiments were implemented under C++ and ROS following the architecture shown in Figure 1. Figure 6 shows a picture taken during the multi-drone target tracking experiments. The figure on the top shows the Central Computer, some *Crazyflies* (marked with red circles), the *HTC VIVE* lighthouse base stations mounted on the walls of the room (marker with green circles), and ground robots used as targets to be tracked in the experiments.

Fig. 6. Pictures of the testbed set-up for the multi-drone target tracking experiments. The *Crazyflies* are marked with red circles, and the lighthouse base stations, in green.

The images gathered by each of the drones were transmitted to the Central Computer, where a ROS-based user-defined *Perception* module executed the multi-drone target tracking EIF-based technique. The pose of each drone was estimated by the lighthouse system. The drone pose estimates are sent through the *CrazyRadio* at a rate of 50Hz and the camera images are sent through the video transmitter at a rate of 25Hz.

The *Perception* module in the Central Computer implements one ROS node for each drone. Each of them performs the operation of the EIF-based target tracking corresponding to each of the drones in a distributed manner. Each ROS nodes first processes the target tracking segmentation in the image plane using simple color and shape criteria. The result is the input to the EIF filter, which obtains the 3D target tracking as described in Section V. The EIF filter is executed at 25Hz.

The testbed was mounted in a $4x4m$ room, as shown in the Figure 6. The lighthouse system was used not only to estimate the drone poses, but also to obtain the positions of the targets that were used as ground truth in these target tracking experiments. The *SteamVR* calibration system was used to calibrate the system. The lighthouse system calibration is programmed also in the multirotor firmware, in order to be able to calculate its own position. Finally, each drone camera was calibrated too.

To validate our tracking algorithm and the testbed we performed a series of experiments where the motion of the targets is tracked by two drones. To obtain the target ground truth position, the target was equipped with a lighthouse deck similar to those on-board the drones.

Figure 7 shows the 3D target position estimation of one of the targets obtained by the EIF running on each drone in one experiment. The target tracking estimates obtained by Drone1 are in blue, and those obtained by Drone2, in red. The target estimations from both drones are very similar and only the estimations from Drone1 (in red color) are visible in the figure. The target ground truth positions along the experiment are in black color. The obtained tracking errors are reasonable.

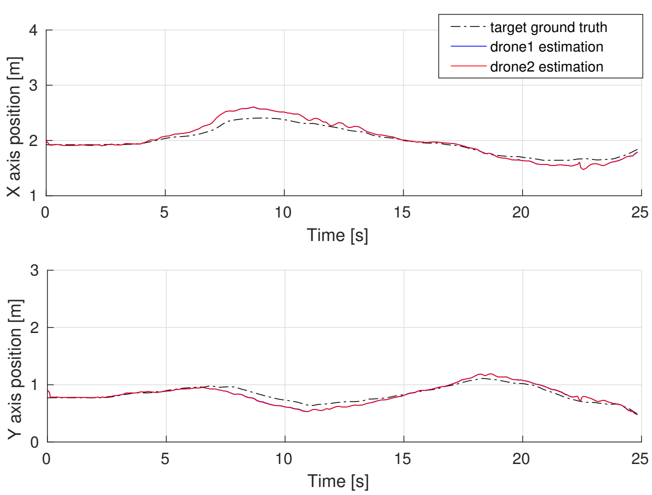

Fig. 7. 3D target position estimations obtained by the EIF running on each drone. The estimations from Drone1 and Drone2 are in blue and in red, respectively. The target ground truth position is in black color.

Figure 8 shows the number of contributions integrated in the EIF Update stage running in Drone1 and in Drone2 along the above experiment. The contributions of both drones are integrated in almost every time instants. Not all contributions were integrated for all times due to communication loss or image segmentation errors. This result evidences the robustness of the tracking method tested.

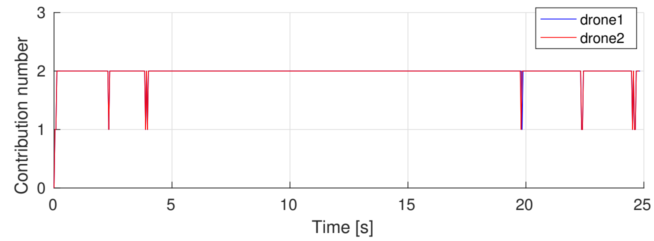

Fig. 8. Number of contributions integrated in the EIF Update stage for Drone1 and for Drone2 along the above experiment.

Table VI shows the mean tracking errors along the set of experiments performed. A high component of the target tracking error is originated by 2D tracking errors or losses. The simple image processing technique used to segment the target in the images fails occasionally originating target tracking inaccuracies. Despite these errors, the performance of the tracking system was good enough for a proof-of-concept test even considering that the testbed was set-up in less than one hour in an not-conditioned space, a regular PhD student office.

	Mean	Maximum	Minimun
Target tracking error [m]	0.047	0.176	0.0013
Target tracking std. deviation [m]	0.021	0.035	0.012
2D target tracking losses [times/min]	5.68	15.1	0.73
EIF contribution losses [times/min]	14.4	36.2	3.5

VII. Conclusions

The increasing interest in aerial robotics has motivated the need for tools that enable the quick and low-cost development of proof-of-concept tests. Significant effort has been devoted to make available ready-to-use aerial robotics software frameworks, libraries and tools. However, the high set-up and maintenance cost of existing aerial robotics testbeds is still a constraint for many potential drone developers.

This paper presents a low-cost multi-drone testbed for agile, flexible and safe performing proof-of-concept experiments. It uses low-size, hardware-efficient and simple *Crazyflie* drones and low-cost lighthouse positioning system. The presented testbed does not require a dedicated space and its flexible ROS-based architecture can be used to easily test multi-drone perception and planning techniques. This paper presented the testbed and validated its operation with multi-drone experiments that implemented EIF-based target tracking.

The results obtained in the presented experiments evidenced that despite the simplicity of the testbed, it is valid for proof-of-concept testing. The total time required for setting-up the experiment was lower than one week. Almost all the time was devoted to the drone hardware modifications to mount the nano-cameras, and the camera calibration for each drone. The full testbed using tripods for the lighthouse base stations was set up in a not conditioned space (a regular PhD student office) in less than one hour. This result shows the agility of the presented testbed to perform proof-of-concept experiments.

The presented experiments tested a version of the EIF that was implemented in the Central Computer due to the computational constraints of the hardware on-board the drones. Future developments include the change in the hardware on-board the *Crazyflies* to enable onboard execution of the full tracking system. Another constraint of the presented tracking system is the simplicity of the target segmentation in the images gathered by the drones. Adding more computational capabilities on-board the drones will enable more complex and accurate target 2D segmentation, which would significantly improve 3D tracking performance.

REFERENCES

[1] M. Quigley, B. Gerkey, K. Conley, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler, and A. Ng, "Ros: an open-source robot operating system," in *Proc. of the IEEE Intl. Conf. on Robotics and Automation (ICRA) Workshop on Open Source Robotics*, Kobe, Japan, May 2009.

[4] R. Clark, G. Punzo, G. Dobie, R. Summan, C. N. MacLeod, G. Pierce, and M. Macdonald, "Autonomous swarm testbed with multiple quadcopters," in *1st World Congress on Unmanned Systems Enginenering, 2014-WCUSEng*, 2014.

[2] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[3] W. Hnig, J. A. Preiss, T. K. S. Kumar, G. S. Sukhatme, and N. Ayanian, "Trajectory planning for quadrotor swarms," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 856–869, Aug 2018.

[5] S. Moss, C. Davis, S. van der Velden, G. Jung, J. Smithard, C. Rosalie, P. Norman, M. Knowles, S. Galea, P. Dang *et al.*, "A uas testbed for the flight demonstration of structural health monitoring systems," *Procedia Engineering*, vol. 188, pp. 456–462, 2017.

[6] R. Cotsakis, D. St-Onge, and G. Beltrame, "Decentralized collaborative transport of fabrics using micro-uavs," *arXiv preprint arXiv:1810.00522*, 2018.

[7] D. Gunasinghe, K. K. Lawson, E. Davis, R. Strydom, and M. Srinivasan, "Mid-air conflict avoidance and recovery: An acceleration-based approach for unmanned aircraft," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 2054–2061, 2019.

[8] B. Smith, B. Stark, T. Zhao, and Y. Chen, "An outdoor scientific data drone ground truthing test site," in *2015 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, 2015, pp. 436–443.

[9] P. Dempsey, "The teardown: Htc vive vr headset," *Engineering & Technology*, vol. 11, no. 7-8, pp. 80–81, 2016.

[10] D. C. Niehorster, L. Li, and M. Lappe, "The accuracy and precision of position and orientation tracking in the htc vive virtual reality system for scientific research," *i-Perception*, vol. 8, no. 3, 2017.

[11] M. Borges, A. Symington, B. Coltin, T. Smith, and R. Ventura, "Htc vive: Analysis and accuracy improvement," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 2610–2615.

[12] J. A. Preiss, W. Honig, G. S. Sukhatme, and N. Ayanian, "Crazyswarm: A large nano-quadcopter swarm," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 3299–3304.

[13] O. Sorkine, "Least-squares rigid motion using svd," *Technical notes*, vol. 120, no. 3, p. 52, 2009.

[14] A. Lorusso, D. W. Eggert, and R. B. Fisher, *A comparison of four algorithms for estimating 3-D rigid transformations*. University of Edinburgh, Department of Artificial Intelligence, 1995.

[15] A. de San Bernabe, J. R. Martinez-de Dios, and A. Ollero, "Efficient cluster-based tracking mechanisms for camera-based wireless sensor networks," *IEEE Transactions on Mobile Computing*, vol. 14, no. 9, pp. 1820–1832, 2014.