Towards Autonomous Navigation of Multiple Pocket-Drones in Real-World Environments

Mcguire, Kimberly; Coppola, Mario; de Wagter, Christophe; de Croon, Guido

DOI
10.1109/IROS.2017.8202164

Publication date
2017

Document Version
Submitted manuscript

Published in
IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)

Towards Autonomous Navigation of Multiple Pocket-Drones in Real-World Environments

Kimberly McGuire[1], Mario Coppola[1], Christophe de Wagter[1], and Guido de Croon[1]

***Abstract*— Pocket-drones are inherently safe for flight near humans, and their small size allows maneuvering through narrow indoor environments. However, achieving autonomous flight of pocket-drones is challenging because of strict on-board hardware limitations. Further challenges arise when multiple pocket-drones operate as a team and need to coordinate their movements. This paper presents a set-up that can achieve autonomous flight in an indoor environment with avoidance of both static obstacles and other pocket-drones. The pocket-drones use only on-board sensing and processing implemented on a STM32F4 microprocessor ($168MHz$). Experiments were conducted with two $40g$ pocket-drones flying autonomously in a real-world office while avoiding walls, obstacles, and each-other.**

I. INTRODUCTION

Pocket-drones are Micro Aerial Vehicles (MAVs) characterized by their low mass and small size. These attributes make them safe for flight near humans and allow maneuvering through narrow indoor areas like corridors, windows, or rooms (as shown in Fig. 1). Pocket-drones are thus ideal for indoor exploration and surveillance tasks, such as green-house observations or search-and-rescue operations [1]. However, real-world applications of pocket-drones are limited by the short flight duration and range of a single platform. By using multiple pocket-drones together, exploratory tasks would be performed more efficiently and transcend the individual limitations [2].

In order for teams of pocket-drones to perform tasks in indoor spaces, they must be able to avoid collisions with static obstacles *and* with each-other. While solutions with a centralized computer or external sensors are possible (e.g. using a motion tracking system [3] or fixed ultra wide-band beacons [4]), they are not applicable to exploration scenarios, where the link to a possible base station can easily be interrupted. Furthermore, the need to set up an external sensor suite would inherently defeat the purpose of the exploration task. It follows that the drones must operate fully autonomously using on-board sensors.

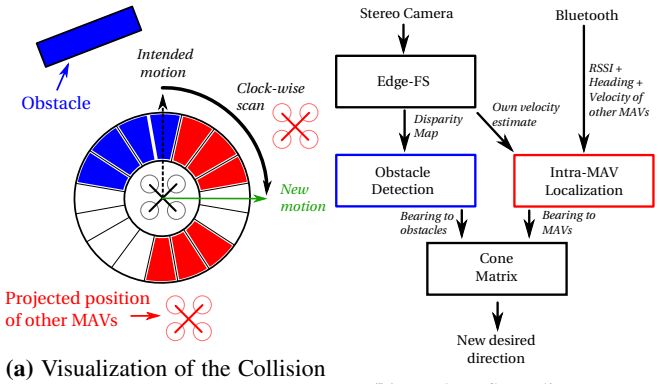
The challenge tackled in this work is to achieve this on real-world pocket-drones. This challenge may be broken down into three sub-challenges. The pocket-drones must: 1) fly autonomously indoors, 2) detect and avoid obstacles in the environment, and 3) localize and avoid each-other. The hard-ware to achieve this must be small, light-weight, and energy-efficient.

Fig. 1: Two pocket-drones (40 g each) flying autonomously in a real-world indoor environment using only on-board sensing and processing.

The first and second sub-challenges require an efficient method for own-state estimation and environment detection. Computer vision is often used for such purposes, as it can turn a simple camera into a versatile sensor, capable of measuring multiple variables. Examples are the detection of ego-motion with optical flow, successfully demonstrated by [5] and creating 3D maps by stereo-vision, as in [6]. Many of the latest computer vision techniques do not scale well for small micro-processors and low resolution cameras. Nevertheless, some efficient methods exists, which are able to on-board miniature MAVs of 40 grams or less ([8] [9]). Recent work by [10] demonstrated successful real-world flight by a pocket-drone in a room using a light-weight stereo camera. However, the pocket-drone was unable to control its height own and, in case of side-wards drift, could collide into obstacles that were not seen by the camera due to its limited Field-Of-View (FOV). The third sub-challenge requires an efficient method for relative localization. In literature, typical methods rely on: high-resolution cameras [11] [12], infra-red sensors [13], or mounted microphone arrays [14]. Recently, Coppola et al. [15] have shown that it is possible to use communication between MAVs to achieve relative localization to a sufficient accuracy for collision avoidance. However, the tests were only performed in a controlled environment. Furthermore, their collision avoidance strategy (a variant of Velocity Obstacle (VO) [16][17]) did not account for heading change by a drone.

The main contribution in this article is a system for fully autonomous flight by a small group of pocket-drones, with active avoidance of static obstacles and other drones.

[1]All authors are with Faculty of Aerospace Engineering, Delft University of Technology, 2628 HS Delft, The Netherlands
k.n.mcguire@tudelft.nl
m.coppola@tudelft.nl
c.dewagter@tudelft.nl
g.c.h.e.decroon@tudelft.nl



(a) Visualization of the Collision Disk.

(b) Update flow diagram.

Fig. 2: Schematic explanation of Collision Disk.

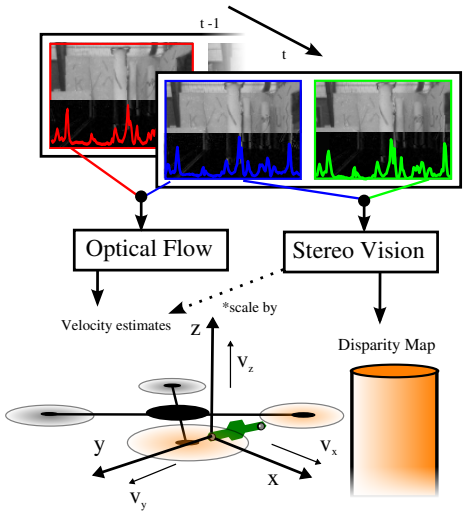


Fig. 3: Schematic explanation of Edge-FS.

The system is demonstrated in a real-world office with two pocket-drones. The work from [10] and [15], described above, were used as starting points. In this work, we add a binary collision avoidance structure to efficiently store the bearing of static obstacles (as sensed by the camera) and the other drones (as sensed via communication). Furthermore, to allow the drones to control their height and avoid drift, we also introduce a range sensor array. This new light-weight sensor provides accurate ranging data side-ways, down-wards, and upwards, which allows the drone to control its own height and detect & react on obstacles outside of the stereo camera's FOV.

This paper is structured as follows. The approach is explained in detail in section II. The drone's behavior was first tested in simulation, as discussed in section III. Finally, section IV shows the results of the real-world experiments. Section V provides concluding remarks.

II. METHOD

This section explains the separate sub-systems that were implemented on the pocket-drone, and the behavior that was used. To efficiently select a safe flight direction, we implemented a binary disk array where directions are marked as either safe or unsafe. This array will be referred to as Collision Disk. It is depicted in Fig.2a. The disk is continuously updated using information coming from both the stereo-camera and the intra-MAV localization, as in Fig.2b.

A. Velocity estimation and Static Obstacles Detection

Velocity estimation and static obstacle detection can be performed simultaneously with a stereo-camera running Edge-Flow Stereo (Edge-FS). Edge-FS stems from the work in [18]. It was followed up by [19] and applied for an autonomous flight of a pocket-drone in [10]. This computer vision algorithm is efficient thanks to the use of edge distributions. Its working principle is depicted in Fig. 3. First, the gradients of an image are computed using a Sobel filter. Then, these gradients are compressed to an edge distribution. This can be either compared to one of a previous time-step, to compute optical flow (Edge-Flow), or with a stereo camera, to compute depth (Edge-Stereo). By scaling Edge-Flow with Edge-Stereo we can estimate velocity along all axes of the drone's coordinate system in North-East-Down (NED). Using Edge-Stereo, a disparity map can be used to also detect obstacles such as walls or objects (within the FOV of the stereo-camera). If the distance to the obstacle is below a threshold, it is added to the Collision Disk with the angle relative to the current heading of the pocket-drone (see Fig.2a). This work currently only considers the closest obstacle.

B. Pocket-Drone Relative Localization

In this paper, we use the relative localization method from Coppola et al. [15], achieving inter-drone localization via communication between the pocket-drones. The height, and the estimated velocity in the horizontal plane (as from Edge-FS) are communicated between drones while the Received Signal Strength Indication (RSSI) is measured. Albeit coarse (Root Mean Squared Error (RMSE) was of $\approx 0.8rad$), the method efficiently provides data on un-safe flight directions to include in the Collision Disk. It was experimentally demonstrated that the accuracy is sufficient for collision avoidance even in small rooms.

A finding from [15] was that too high cautiousness in collision avoidance leads to a restriction in motion, which is ultimately detrimental. To avoid this, the collision cone only considers drones at an estimated distance below a threshold d_{drone}. To account for the relative motion of the drones, the Collision Disk does not directly use the estimated location, but a projected location of the other MAV a certain time into the future (Fig. 4). This is based on the angular velocity of the drone, such that

$$\Delta\beta \approx \kappa_t \cdot \frac{v_\perp}{d}, \tag{1}$$

where: d is the distance to another drone; $v_\perp$ is the perpendicular velocity of the moving drone about the observing drone; and κ_t is a factor equal to the amount of seconds in

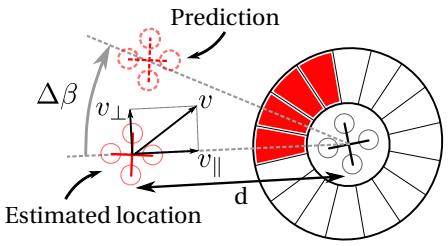


Fig. 4: MAV location prediction in the Collision Disk

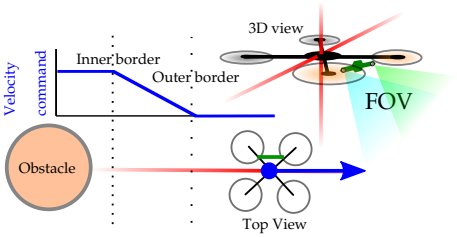


Fig. 5: Visualization about the forcefield generated by the range sensors where the red lines illustrate their direction of gaze.

the future that are estimated. In its current implementation, κ_t is set manually. In all the experiments in this paper $\kappa_t = 1$.

C. Height control and drift compensation

The pocket-drones will be equipped with four range-sensors, pointed toward the top, bottom, right and left (see Fig. 5). The top and bottom range-sensors will be used for basic height control, where the option exist to transverse along the ceiling of an indoor environment. For the sides it will act like a fail safe in case the pocket drone drifts towards an obstacle outside of the stereo-camera's FOV. Here a simple force-field principle is applied with an inner- and outer-border, as illustrated in Fig. 5. If the drones drift to a distance between the obstacle and the outer-border, it will get an extra velocity command to get out of this situation. From here on, the magnitude of this command is linearly dependable on the distance between pocket-drone and obstacle, which is bounded with a maximum velocity command from the inner-border on. For the experiments, the inner- and outer border are set on 0.8 and 1.2 meters respectively and the maximum bounded velocity is 0.3 m/s.

D. Behaviour

To navigate within the FOV of the camera, the MAVs should always be in forward flight so that they can visually detect obstacles (e.g. walls). The avoidance behavior then performs the following operations, in order: 1) velocity is reduced to zero, 2) the MAV rotates to face a new direction that it deems safe, 3) the MAV resumes forward flight in the new direction. If the MAV is flying towards a region marked as unsafe, it will stop and turn clock-wise until it is facing a direction marked *safe*. This is depicted in Fig. 2a.

III. SIMULATION

Prior to real-world tests, the behavior was tested in simulation using Robotics Operating System (ROS) [20] and the *hector-quadrotor* simulator within Gazebo [21] (Fig. 6a). The simulated MAVs fly at the same height in an arena. The simulated MAV diameter was $0.2m$. The RSSI noise and lobes were set to $5dB$, which is similar to the real-world Bluetooth performance. No simulated range-sensors and stereo-camera were used here, so the height and velocity were taken directly from the ground truth. The velocity and height estimation error were set at $0.2m/s$ and $0.2m$, respectively. The arena was $6m \times 6m$.

	No Avoidance	$d_{drone} = 2m$	$d_{drone} = 5m$
2 MAVs	$194s$ & 10 coll.	$266s$ & 8 coll.	$421s$ & 4 coll.
3 MAVs	$66s$ & 10 coll.	$103s$ & 9 coll.	$177s$ & 10 coll.

TABLE I: Simulation statistics (mean flight time and number of collisions). 10 simulated flights were run for each parameter pair.

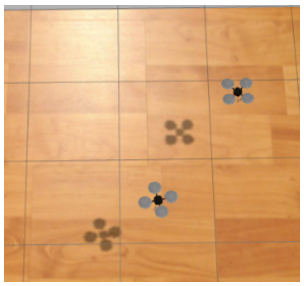
In the first test, one drone was held static while the other was let lose in the space. Repeated simulations of 500 s showed no collisions. The log of a simulation is shown in Fig. 6b. The moving MAV could successfully combine knowledge of the other drone and the walls to choose a safe path.

To test out the scalability of the method, trials were conducted with 2 and 3 MAVs in the same arena. The threshold distance from the drone to the wall was $1.2m$. The maximum trial duration was $500s$, but an inter-drone collision will end the trial prematurely. d_{drone} was set to $2m$ (Fig. 7a and 7c) and $5m$ (Fig. 7b and 7d). For each parameter pair, 10 trials were run. An overview of the results can be found in Tab. I. For the 2 and 3 MAVs scenarios, the average flight times (over 10 flights) increased with $d_{drone} = 5m$ instead of $2m$ and the number of collisions went down. However, the trajectories (see Fig.7) show that increasing d_{drone} restricts freedom of movement. In 7d), MAV 1 and MAV 3 were stuck in one corner and could not move out of their position. To favor unrestricted movement, a smaller d_{drone} would be preferred, although this inherently increases collision risk.

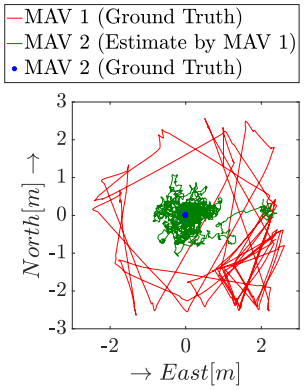
Overall, the results from this behavior do not improve the average flight-times if compared to the simulated results from [15]. One of the added complication is that now the MAV behaviour is highly non-holonomic, as it constantly stops to change its own heading. Instead, [15] simply changed direction of flight. Stopping to change heading decreases the accuracy of the relative localization, which relies on motion. Finally, adding the real wall detection to the Collision Disk adds further complexity. Nevertheless, the simulation results show that 2 pocket-drones can fly within a room for the almost full duration of their battery (approx. $5min$), which will be demonstrated in a real-world environment in the next section.

IV. REAL-WORLD EXPERIMENTS

The system was implemented in real pocket-drones. In this section, the hardware and software specifics are presented and the experiments are shown of the pocket-drones flying autonomously in a real-world environment.

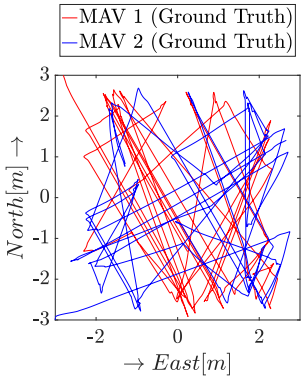


(a) Screen-shot of a simulation trial.

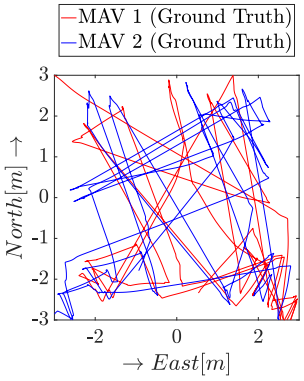


(b) Simulation log with a static drone over a $500s$ flight.

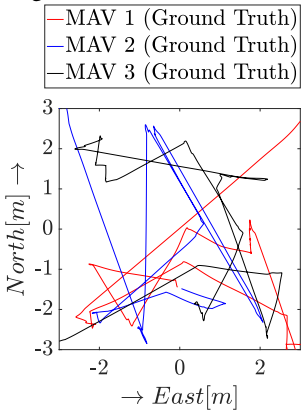
Fig. 6: Set-up of simulation environment.



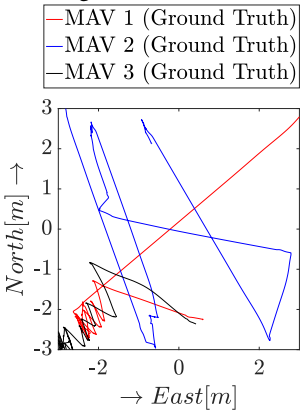
(a) 3 MAVs with $d_{drone} = 2$. Flight duration = $500s$.



(b) 2 MAVs with $d_{drone} = 5m$. Flight duration = $500s$.



(c) 3 MAVs with $d_{drone} = 2m$. Flight duration = $163s$.



(d) 3 MAVs with $d_{drone} = 5m$. Flight duration = $437s$.

Fig. 7: Simulation logs with 2 and 3 MAVs for $d_{drone} = 2m$ and $d_{drone} = 5m$

A. Hardware and Software Set-up

A single pocket-drone consists of a Lisa-MXs auto-pilot module (a smaller variant of the Lisa-MX [22]), similar to the one used in [10] (see Fig. 8). It carries an STM32F4 microprocessor with a speed of $168MHz$ and $1MB$ of flash memory. The 4 gram stereo-camera also features an embedded STM32F4 microprocessor with a speed of $168MHz$ and $196kB$ of memory in which the largest consecutive memory block spans 128 kB. The processed images are $128 \times 96px$ and the camera has a $57.4 \times 44.5deg$ FOV. With this hardware, the Edge-FS algorithm can run in parallel with the regular flight controllers of the Lisa-MXs. Everything is mounted on a Walkera QR LadyBug quad-copter frame [23].

The intra-drone communication and RSSI measurement is done by a Bled112 Bluetooth smart dongle [24] (as used in [15]). For testing and validation purposes, an ESP-09 WiFi module was used to broadcast high-speed telemetry to the ground computer. It was not used to send any prior information about the testing area as the pocket-drones interprets the environment (by Edge-FS) and perform the intra-MAV localization (RSSI) all on-board.

Finally, a $0.2mm$ thick, $7mm$ wide and $88mm$ long Polyimid Flex-PCB with four VL53L0X Time-of-Flight ranging sensor [25] was designed. The flexible board is bent into a ring and attached to the pocket drone resulting in ranging sensors pointing towards the sides and towards the bottom and the ceiling. By configuring the range sensors into *long range mode*, they can measure an absolute range up to $2m$ at $8Hz$. A local *ATmega328P-MLF28* microcontroller interfaces with all the sensors and sends the combined measurements to the Lisa-MXs over a single wire. The total weight of the board is $0.25g$.

With everything combined, the MAV's total mass is approximately 43 gram (including a 11 gram battery).

The auto-pilot program flashed on the Lisa-MXs is Paparazzi UAV [26]. All algorithms and controllers of the software runs entirely on the microprocessor. The basic low levels controllers regulate the attitude of the pocket-drone. On top of this, a PID guidance controller coordinates the MAV's velocity in the X (forward) and Y (sideways) direction. In this paper, the velocity estimated in the X-Y-Z is given by Edge-FS and is actively controlled in the horizontal plane. Since the range sensors provide an accurate position of the altitude of the pocket-drone, it can maintain a fixed height for the duration of the flight. The side range sensors will not be used for the main navigation, since individually they have a very narrow receptive angle, but will act as a velocity force field. If the pocket-drone comes too close to a wall or obstacle at the sides — which is beyond the FOV of the camera — it will give an opposite velocity commands that is added to the one from the main navigation and steers the MAV away from the lateral obstacle. In the experiments, only one pocket-drone (PD1) is equipped with the range sensors and the other one is not (PD2).

B. Experiment results

The experiments with two pocket-drones were conducted in a real-world environment: an office at the faculty of Aerospace Engineering from the Delft University of Technology (Fig. 9). This office's dimensions are $5.0 \times 4.0 \times 2.7m$ in length, width, and height, respectively. The office features varying types texture as commonly found in such areas. The glass cabinets were given a bit of additional coverage

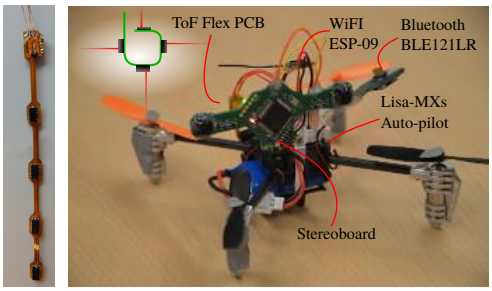


Fig. 8: Picture of the pocket-drone and its hardware.

Fig. 9: Panorama of testing site

as this is still a difficult scenario for Edge-FS as well for the proximity sensors. Four infrared OptiTrack cameras [27] were placed near the ceiling of the room to measure a (sparse) trajectory for determining their coverage, but is used for post flight analysis only.

The pocket drones started out with a manual take-off, from which they switch to autonomous control mid-air. The thresholded distance for both the range for the other MAV as the obstacles was set to $1.5m$. The pocket-drone (PD1) carrying the range sensors, would start out first as it is able to maintain its own attitude (which it maintains at $1.5m$, taking the ceiling as a reference). Once PD1 is flying autonomously for a few seconds, the second MAV (PD2) takes off and switched to guided mode. As the drone does not contain the range sensor PCB ring, its height has to be controlled manually. In the horizontal plane however, the same avoidance logic for the turning exists as with the first drone, with some extra velocity guidance of the remote control[1]. This means that only the preferred velocity is given by the remote control, from which it has to match with its own velocity estimate of Edge-FS, and not the exact angle set points as with the common attitude manual control. However, the difficulty in guiding PD2, is that it is controlling its own heading based on obstacles and the location of PD1. Since this and its velocity control (given by the remote control) are decoupled, the safety pilot had a hard time controlling PD2, resulting in a longer average flight duration for the full autonomous PD1.

Based on the full duration of PD1's flight, 4 tests were performed, with a duration of $119s$, $311s$, $321s$ and $103s$ respectively. PD1 crashed in the 1^{st} flight because of an undetected static obstacle. In the 2^{nd} and 3^{rd} flight, PD1 flew autonomously until the end of its battery life. In the 4^{th}

[1]Future work will include 2 full autonomous pocket-drones without any guidance of the remote control.

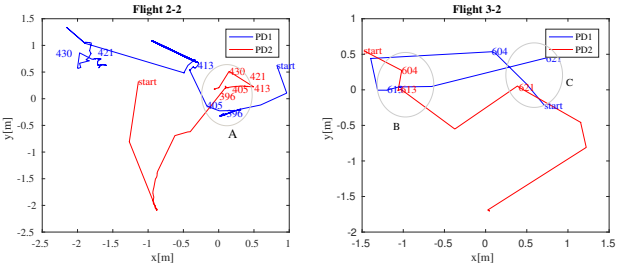


Fig. 10: Top view flight tracking of flight 2-2 and 3-2

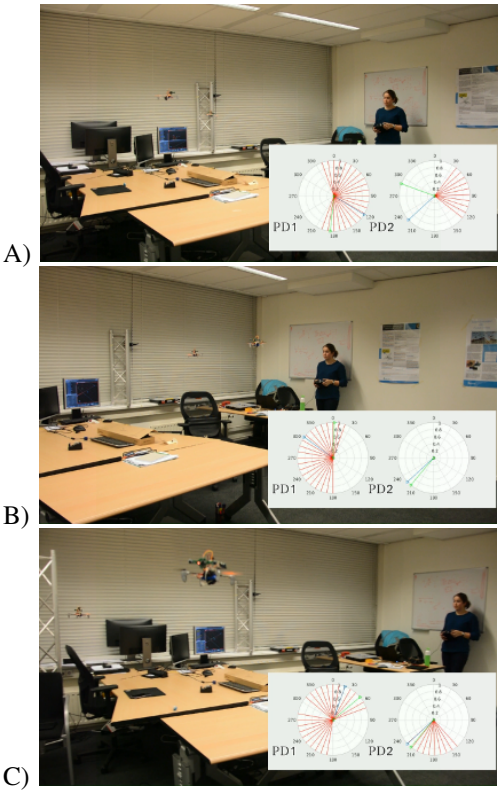


Fig. 11: Screen shots of flight 2-2 at A) 2:21 sec and flight 3-2 at B) 2:46 and C) 2:54 sec.

flight, PD1 was caught in between two pillars and facing a wall, and was unable to escape from this dead-lock. For now, this paper will focus on flight 2 and 3 specifically because of their length, as more intra-MAV collision avoidance situations can be analyzed. The flights contain multiple restarts of PD2, therefore can be split up in sub-flights (2-1 to 2-3 and 3-1 to 3-4). In Fig. 10, the trajectory by the OptiTrack cameras is shown for flight 2-2 and 3-2. The motion capture system was undersized for full room coverage, and was thus unable to track the drones the entire time due to occlusion and their small size. This resulted in some artifacts in the tracked position. However, some moments were identified where the pocket-drones came in proximity of each other, which are annotated in the plots of Fig. 10.

We shall discuss three representative scenarios, which have been recorded on video and their screen-shots can be found in Fig. 11. These images show the moments of close

proximity, as annotated in Fig. 10. The first screen-shot shows Scenario A, where both drones were able to see each-other and changed their heading accordingly. In the Scenario B, PD2 failed to locate PD1 correctly as it is not shown in the Collision Disk. However, PD1 successfully detected PD2 and changed its heading to an obstacle free direction. In the last screen-shot, Scenario C, PD2 did see PD1 (as indicated in the Collision Disk), and only had to adjust its heading slightly. PD1 did also detect PD2, but since it is not on its planned trajectory it did not go into an evasive maneuver at first. However, as it was heading towards an obstacle, it will plan a turn into an collision-free direction. All video recorded flights can be find in a dedicated YouTube play-list. [2]

V. Conclusion

To the best of our knowledge, this paper presents the first attempt to fly a fully autonomous team (duo) of pocket-drones in a real-world environment. We combined state-of-the-art methods for own-state estimation and inter-drone tracking for pocket-drones, and added additional range sensors to control height and side-ways drift. With this set-up, the pocket-drones can achieve stable flight. Using a binary structure called Collision Disk, they could efficiently select collision free paths (from static obstacles and other drones) while exploring their environment. The experiments showed that the pocket-drones made the right maneuvers at close proximity of each-other. By means of simulation, there are indications that this method can scale to teams of three or more drones. However, the accuracy of the relative localization and the avoidance behavior needs to be developed further in order to achieve this successfully. Nevertheless, this work takes a step closer towards achieving a team of pocket-drones, which is able navigate indoor without any external sensors or prior knowledge of the environment.

References

[1] D. Scaramuzza, M. C. Achtelik, L. Doitsidis, F. Friedrich, E. Kosmatopoulos, A. Martinelli, M. W. Achtelik, M. Chli, S. Chatzichristofis, L. Kneip, D. Gurdan, L. Heng, G. H. Lee, S. Lynen, M. Pollefeys, A. Renzaglia, R. Siegwart, J. C. Stumpf, P. Tanskanen, C. Troiani, S. Weiss, and L. Meier, "Vision-controlled micro flying robots: From system design to autonomous navigation and mapping in gps-denied environments," *IEEE Robotics Automation Magazine*, vol. 21, no. 3, pp. 26–40, Sept 2014.

[2] M. Brambilla, E. Ferrante, M. Birattari, and M. Dorigo, "Swarm robotics: a review from the swarm engineering perspective," *Swarm Intelligence*, vol. 7, no. 1, pp. 1–41, 2013.

[3] Y. Mulgaonkar, G. Cross, and V. Kumar, "Design of small, safe and robust quadrotor swarms," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, May 2015, pp. 2208–2215.

[4] A. Ledergerber, M. Hamer, and R. D'Andrea, "A robot self-localization system using one-way ultra-wideband communication," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sept 2015, pp. 3131–3137.

[5] D. Honegger, L. Meier, P. Tanskanen, and M. Pollefeys, "An open source and open hardware embedded metric optical flow cmos camera for indoor and outdoor applications," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. IEEE, 2013, pp. 1736–1741.

[6] A. Geiger, J. Ziegler, and C. Stiller, "Stereoscan: Dense 3d reconstruction in real-time," in *Intelligent Vehicles Symposium (IV), 2011 IEEE*. IEEE, 2011, pp. 963–968.

[7] O. Dunkley, J. Engel, J. Sturm, and D. Cremers, "Visual-inertial navigation for a camera-equipped 25g nano-quadrotor," in *IROS2014 aerial open source robotics workshop*, 2014, p. 2.

[8] R. J. D. Moore, K. Dantu, G. L. Barrows, and R. Nagpal, "Autonomous mav guidance with a lightweight omnidirectional vision sensor," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, May 2014, pp. 3856–3861.

[9] C. De Wagter, S. Tijmons, B. Remes, and G. de Croon, "Autonomous flight of a 20-gram flapping wing mav with a 4-gram onboard stereo vision system," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*. IEEE, 2014, pp. 4982–4987.

[10] K. McGuire, G. de Croon, C. D. Wagter, K. Tuyls, and H. Kappen, "Efficient optical flow and stereo vision for velocity estimation and obstacle avoidance on an autonomous pocket drone," *IEEE Robotics and Automation Letters*, vol. 2, no. 2, pp. 1070–1076, April 2017.

[11] S. Shen, N. Michael, and V. Kumar, "3d indoor exploration with a computationally constrained mav," in *Robotics: Science and Systems*, 2011.

[12] P. Conroy, D. Bareiss, M. Beall, and J. v. d. Berg, "3-d reciprocal collision avoidance on physical quadrotor helicopters with on-board sensing for relative positioning," *arXiv preprint arXiv:1411.3794*, 2014.

[13] J. F. Roberts, T. S. Stirling, J. C. Zufferey, and D. Floreano, "2.5d infrared range and bearing system for collective robotics," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct 2009, pp. 3659–3664.

[14] M. Basiri, F. Schill, P. Lima, and D. Floreano, "On-board relative bearing estimation for teams of drones using sound," *IEEE Robotics and Automation Letters*, vol. 1, no. 2, pp. 820–827, July 2016.

[15] M. Coppola, K. McGuire, K. Y. Scheper, and G. C. de Croon, "On-board bluetooth-based relative localization for collision avoidance in micro air vehicle swarms," *arXiv preprint arXiv:1609.08811*, 2016.

[16] D. Wilkie, J. van den Berg, and D. Manocha, "Generalized velocity obstacles," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct 2009, pp. 5573–5578.

[17] P. Fiorini and Z. Shiller, "Motion planning in dynamic environments using velocity obstacles," *The International Journal of Robotics Research*, vol. 17, no. 7, pp. 760–772, 1998. [Online]. Available: http://dx.doi.org/10.1177/027836499801700706

[18] D.-J. Lee, R. W. Beard, P. C. Merrell, and P. Zhan, "See and avoidance behaviors for autonomous navigation," in *Optics East*. International Society for Optics and Photonics, 2004, pp. 23–34.

[19] K. McGuire, G. de Croon, C. de Wagter, B. Remes, K. Tuyls, and H. Kappen, "Local histogram matching for efficient optical flow computation applied to velocity estimation on pocket drones," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, May 2016, pp. 3255–3260.

[20] "Robot Operating System (ROS) ," http://www.ros.org/.

[21] "Gazebo," http://gazebosim.org/.

[22] "Lisa MX paparazzi wikipedia manual," http://wiki.paparazziuav.org/wiki/Lisa/MX.

[23] "Walkera qr ladybug quad-copter," http://www.walkera.com/.

[24] "Scilabs bled112 bluetooth smart dongle," http://www.silabs.com/products/wireless/bluetooth/bluetooth-low-energy-modules/ble121lr-bluetooth-smart-long-range-module.

[25] "STMicroElectronics vl53l0x time-of-flight ranging sensor," http://www.st.com/content/st_com/en/products/imaging-and-photonics-solutions/proximity-sensors/vl53l0x.html.

[26] "PaparazziUAV wikipedia manual," http://wiki.paparazziuav.org/.

[27] "OptiTrack," http://www.optitrack.com/.

Acknowledgments

We would also like to thank Erik van der Horst from the MAVlab for setting up several testing and measurement equipment used in this paper.

[2] https://www.youtube.com/playlist?list=PL_KSX9GOn2P9CdtNz_p_tv1cpEs1E3KUx