A Novel Obstacle Detection and Avoidance Dataset for Drones

Dupeyroux, Julien; Dinaux, Raoul; Wessendorp, Nikhil; De Croon, Guido

DOI
10.1145/3522784.3522786

Publication date
2022

Document Version
Final published version

Published in
Proceedings of System Engineering for Constrained Embedded Systems - DroneSE

A Novel Obstacle Detection and Avoidance Dataset for Drones

Julien Dupeyroux
julien.dupeyroux@gmail.com
Micro Air Vehicle Lab, Faculty of Aerospace Engineering,
Delft University of Technology
Delft, The Netherlands

Raoul Dinaux
raouldinaux@gmail.com
Micro Air Vehicle Lab, Faculty of Aerospace Engineering,
Delft University of Technology
Delft, The Netherlands

Nikhil Wessendorp
n.wessendorp@gmail.com
Micro Air Vehicle Lab, Faculty of Aerospace Engineering,
Delft University of Technology
Delft, The Netherlands

Guido de Croon
g.c.h.e.decroon@tudelft.nl
Micro Air Vehicle Lab, Faculty of Aerospace Engineering,
Delft University of Technology
Delft, The Netherlands

ABSTRACT

In this paper, we introduce the Obstacle Detection & Avoidance (ODA) Dataset for Drones, aiming at providing raw data obtained in a real indoor environment with sensors adapted for aerial robotics in the context of obstacle detection and avoidance. Our micro air vehicle (MAV) is equipped with the following sensors: (i) an event-based camera, the performance of which makes it optimized for drone applications; (ii) a standard RGB camera; (iii) a 24-GHz radar sensor to enhance multi-sensory solutions; and (iv) a 6-axes IMU. The ground truth position and attitude are provided by an OptiTrack motion capture system. The resulting dataset consists of more than 1350 sequences obtained in four distinct conditions (one or two obstacles, full or dim light). It is intended for benchmarking algorithmic and neural solutions for obstacle detection and avoidance with UAVs, but also course estimation and in general autonomous navigation. The dataset is available at: https://github.com/tudelft/ODA_Dataset [6].

CCS CONCEPTS

• **Computer systems organization** → **Robotic autonomy**; **Robotic control**; **Sensors and actuators**.

KEYWORDS

Unmanned Aerial Vehicles (UAVs), Micro Air Vehicles (MAVs), Camera, Event-based Camera, Neuromorphic Vision, Radar, Robot Operating System (ROS)

ACM Reference Format:
Julien Dupeyroux, Raoul Dinaux, Nikhil Wessendorp, and Guido de Croon. 2022. A Novel Obstacle Detection and Avoidance Dataset for Drones. In *System Engineering for constrained embedded systems (DroneSE and RAPIDO '22), June 21, 2022, Budapest, Hungary.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3522784.3522786

DroneSE and RAPIDO '22, June 21, 2022, Budapest, Hungary

ACM ISBN 978-1-4503-9566-3/22/01.
https://doi.org/10.1145/3522784.3522786

1 INTRODUCTION

Nowadays, it is possible to make autonomous UAVs perform feats like autonomous drone racing (e.g. AlphaPilot 2019 [3] - Lockheed Martin AI Drone Racing Innovation Challenge), or crossing windows with aggressive maneuvers [17], or performing autonomous aerial acrobatic maneuvers [9]. Yet, it is quite disconcerting to note how difficult it is to autonomously achieve one of the most essential tasks for drones, namely, obstacle detection and avoidance. The complexity is essentially due to the difficulty to detect obstacles in the focus of expansion, where the optic flow is close to zero. This problem gets even harder in real environments where, for instance, the light intensity can change, sometimes abruptly, making it impossible to ensure the robustness of autonomous systems for obstacle detection and avoidance.

In recent years, a series of datasets have been collected for indoor navigation tasks. In 2015, [20] acquired a collection of more than 10k RGB-D images provided by multiple sensors such as the Kinect v1 and v2, as well as the Intel RealSense depth sensor. By doing so, they extended the NYU Depth v2 dataset which contains 1449 RGB-D images from the Kinect v1 [19]. Similarly, the ESAT dataset was released for the purpose of indoor visual-based obstacle detection and avoidance with a drone using a single RGB monocular camera [2]. It consists of 20k RGB-D images collected from a Kinect sensor by walking. The ESAT and the NYU Depth v2 datasets are mostly used in the context of scene understanding (e.g., including semantic segmentation, object detection and classification, context reasoning, etc.), making them less relevant for the context of obstacle detection and avoidance per se, particularly in case the obstacle is located in the course of the drone/robot. In the meantime, [22] proposed a dataset for obstacle avoidance onboard ground robots. The dataset consists of 10k RGB images used to estimate the depth and therefore allow safe navigation of a ground robot. More recently, [15] released a collision avoidance video dataset collected by means of a drone equipped with a frontward RGB camera. A distinctive feature of this dataset is that it includes 100 samples of actual collisions with static and moving obstacles.

Despite being very popular, the aforementioned datasets have limited applications to robotics in general, and for MAVs in particular: (i) obstacles are highly textured and outside of the course of the moving entity (robot/human), (ii) only a limited number of sensors (i.e., the RGB-D Kinect sensor) is used, and (iii) the light intensity is kept constant and optimal (dim light conditions not considered).

Obstacle detection and avoidance onboard drones is a very challenging task for standard technologies such as RGB cameras, which perform poorly in low-visibility conditions (e.g., dim light, smoke).

In recent years, the development of event-based cameras revolutionized the way robots and drones can sense the world [7]. Unlike conventional cameras for which the images are produced at a fix rate, typically 30-60 fps, event-based cameras like the Dynamic Vision Sensor (DVS, [11]) output a stream of asynchronous events at extremely fast rate. An event is the combination of four elements: the pixel coordinates x and y, the polarity p, and the timestamp t. Events are generated whenever a pixel of coordinates (x, y) perceives a change of its brightness. If the change is positive, the polarity is equal to $+1$, else it is equal to -1. The advantages of DVS cameras over traditional cameras are decisive for complex robotic tasks such as obstacle detection and avoidance:

(a) they feature a *high dynamic range* (often $> 120\ dB$) which allow applications in a wide range of lighting conditions (daylight and moonlight), as well as extreme conditions where the ambient light can suddenly change from high to low, and vice versa (the so-called tunnel effect);
(b) because pixels are updated independently, DVS cameras show a very *low latency*, often measured in the range of one millisecond;
(c) they benefit from a *high temporal resolution*, with a microsecond resolution for timestamps registration of events, thus making them suitable for applications to fast motion and insensitive to motion blur.

In light of these advantages, DVS cameras represent a powerful alternative to conventional cameras in the field of visual-based obstacle detection and avoidance, where motion perception is crucial to achieve robust and accurate performances. This is further supported by the growing number of datasets now available with applications for Visual Inertial Odometry (VIO, [4, 12]), depth-estimation [8], and image reconstruction [14, 18]. However, and to the best of our knowledge, none of the available datasets addresses the problem of obstacle detection when the obstacle is located in the drone's course, thus implying avoidance maneuvers.

Visual-based detection of obstacles in dim light represents a very challenging task, even for DVS cameras, particularly when exploration is performed in the total darkness. Alternatively, millimeter wave (MW) radars could help in obstacle detection regardless of the lighting conditions. Recent work showed the growing interest for radar applications to MAVs for autonomous exploration of unknown environments [10, 23]. MW radar sensors also have the advantage of being able to detect the Doppler signature of moving objects, from which the radial velocity can be extracted. Nowadays, the form factor of radars is small enough to be used onboard MAVs: typical MW radars weigh $10-30\ g$, for a total surface in the range of $25-30\ cm^2$. So far, MW radars remain quite unused in drones applications, where embedded vision remains the major type of sensing for robots. The side effect of this under-representation of radars in obstacle detection and avoidance onboard MAVs is a complete lack of datasets including MW radar sensors.

In this paper, we introduce the Obstacle Detection and Avoidance (ODA) dataset for applications in UAVs. This is a generic dataset specifically designed to address the problem of obstacle detection in varying challenging contexts: (i) obstacle(s) located in the drone's course, (ii) low-light conditions, and (iii) low-textured visual cues. In this respect, we provide a set of more than 1350 indoor navigation tests featuring data from the following sensors:

(a) the DVS240 event-based camera;
(b) a standard RGB Full HD camera;
(c) a 24-GHz radar sensor;
(d) a 6-axes inertial measurement unit (IMU).

For each run, the ground truth position and attitude of the drone is given by the OptiTrack motion capture system.

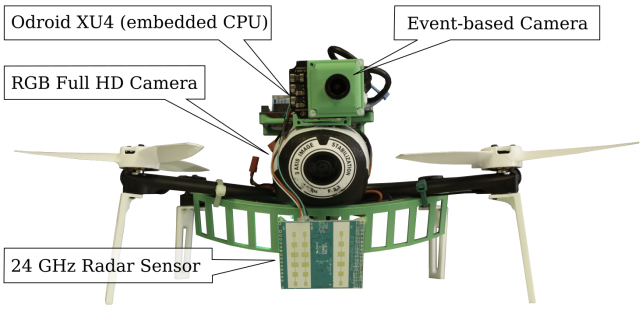

Figure 1: The MAV equipped with the sensors used in the ODA dataset, and the embedded CPU for data collection. The IMU is built in the DVS sensor.

2 MATERIAL AND METHODS

2.1 The hardware architecture

During the dataset collection, we used the Parrot Bebop-2 drone equipped with additional sensors and embedded CPU for data acquisition. More precisely, the drone embeds an Odroid XU4 board running the Linux Kernel 4.14 LTS. For the purpose of obstacle detection and avoidance in MAVs flying through challenging environments, the following sensors have been integrated on the drone: (i) the DVS240 event-based camera [1]; (ii) a 6-axes IMU, integrated with the DVS240 camera; and (iii) the Infineon XENSIV 24GHz Position2Go radar sensor. Integrated to the Bebop 2 drone, a full HD RGB camera is mounted at the front and provides wide-angle video recordings on flight. The hardware architecture is detailed in Table 1 and Fig 1.

The DVS240[1] is an event-based camera with a frame resolution of 240×180 pixels. It outputs a stream of events characterized by bandwidth of $12 \cdot 10^6$ events per second, a high dynamic range of 120 dB, and an extremely low latency (12 us). Its low power consumption (180 mA at 5 V DC) makes it suitable for embedded applications. However, the form factor of the sensor limits the use in MAVs. For that reason, the protecting case of the sensor was removed, and the lens was replaced by a miniature lens, mounted on a custom-made, 3d-printed mount designed for the camera, thus resulting in a drop of the total weight by 75%, at 35 g. In order to allow applications of obstacles detection in challenging environments, e.g., in the dark or in a room filled with smoke, we included the Infineon Position2Go 24GHz radar sensor. It features a fast-chirp

[1] https://inivation.com/wp-content/uploads/2020/04/DVS240.pdf

Table 1: General overview of the hardware required for the drone setup.

Property	Description	Details
Dimensions	382 mm x 328 mm x 140 mm	
Weight	600 g	*Battery included*
Flight performance	15 minutes	*11.1 V 3100 mAh LiPo battery*
Onboard processing (for autopilot)	- Parrot P7 dual-core CPU Cortex 9 - Quad-core GPU	
Embedded CPU (for dataset collection)	Odroid XU4 ft.: - Samsung Exynos5422 Cortex A15 2Ghz - Cortex A7 octa-core CPUs - Linux Kernel 4.14 LTS	
Embedded sensors	- DVS240 event-based camera (iniVation) - Full HD wide-angle RGB camera - 24 GHz radar sensor Position2Go (Infineon) - 6-axes IMU sensor	Resolution: 240×180 pixels Resolution: 1920×1080 pixels FOV: 76°×19° Integrated to DVS240 camera

frequency-modulated continuous wave (FMCW) radar transmitter and two receiver antennas to determine the range, bearing and radial velocity of obstacles and direction of motion. The radar sensor can detect obstacles at a maximum distance of 25 m, and has an FOV 76° (horizontal) × 19° (vertical). It is particularly adapted for MAVs applications, with a total weight of 10 g. In addition to the aforementioned sensors, we collected video recordings of the drone front view using the built-in RGB camera with full HD (1080p) settings. The output frame rate is equal to 30 fps.

2.2 The ROS-based acquisition framework

The dataset was collected by means of a dedicated ROS package running onboard the embedded CPU (Odroid XU4). This package manages communication with and data logging from the event-based camera, the 6-axes IMU, and the radar sensor. The embedded CPU clock allows to time-synchronize ROS bags. The ROS drivers for the DVS240 camera are provided by Robotics Perception Group[2] [13]. The events streamed by the camera consist of a set of pixel coordinates (x, y), a polarity $p \in \{-1, +1\}$, and the corresponding timestamp. Radar data are saved as raw ADC samples: the (I, Q) values of antennas 1 and 2 respectively, with 128 samples per chirp, 16 chirps per frame, and 300 microsecond per chirp as default settings. As a result, for each timestamp, the raw output consists of 4 vectors (2 antennas $\times(I, Q)$ vectors), each vector containing $16 \times 128 = 2048$ samples. The average acquisition rate of the radar is equal to 15.7 Hz.

2.3 The experimental setup

The obstacle avoidance dataset has been collected in the Cyber Zoo, the flying arena of the Delft University of Technology ($10 \times 10 \times 7$ meters). The arena is equipped with the OptiTrack motion capture system, featuring a total of 16 Prime X-13 cameras, allowing to real-time track drones (position and attitude) with millimeter precision. The MAV was equipped with IR LEDs and the OptiTrack cameras lights were turned off to avoid visual disturbances caused by the IR flickering from the cameras. For each trial, a pilot flies the MAV towards the center of the Cyber Zoo where one or two obstacles (i.e. poles) are placed (Fig. 2.) Depending on the position of the MAV, the pilot chooses to either turn left or right (avoidance), or continue straight-forward. An example is provided in Fig. 3. The samples (i.e., a collection of data gathered during one obstacle avoidance task) were recorded under the following lighting conditions: full light (100 Lux), and dim light (1–3 Lux). Also, the number of obstacles available in the flying arena varied between one and two poles.

[2]https://github.com/uzh-rpg/rpg_dvs_ros

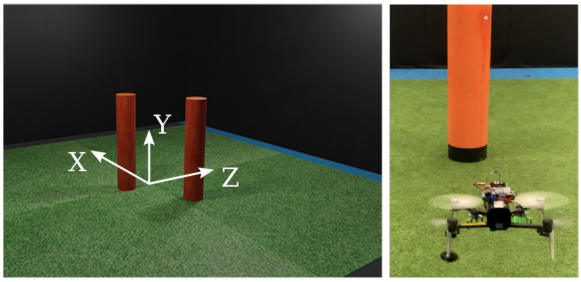

Figure 2: The experimental context. (Left) Rendering of the Cyber Zoo with two poles at the centre. (Right) View of the drone facing a pole before an avoidance maneuver.

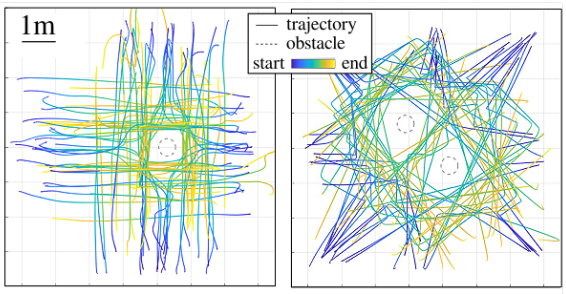

Figure 3: Examples of trajectories including one (left) and two (right) obstacles. These trajectories show the coordinates of the manually controlled MAV provided by OptiTrack.

Table 2: Multimedia extensions available in the ODA Dataset.

Extension	Media type	Folder/File name	Description
1		*./dataset/*	Folder containing the raw data collected.
2	Data	– *trial_overview.csv*	Overview of the dataset (lighting, obstacles, etc.).
3	Data	– *<ID>/<ID>.bag*	ROS bag for trial <ID> containing data.
4	Data	– *<ID>/dvs.csv*	Log files from the DVS240 event-based camera.
5	Data	– *<ID>/imu.csv*	Log files from the 6-axes IMU sensor.
6	Data	– *<ID>/radar.csv*	Log files from the 24-GHz radar sensor.
7	Data	– *<ID>/optitrack.csv*	Log files from OptiTrack (ground truth).
8	Video	– *<ID>/<ID>.avi*	Video output from RGB camera.
9	Calibration	./calibration/dvs/	Folder with calibration files for DVS camera.
10	Calibration	./calibration/camera/	Folder with calibration files for RGB camera.
11	Code	*./test_<sensor>_<format>.py*	Python scripts for data loading & visualization. Sensors: *'dvs'*, *'imu'*, *'radar'*, and *'optitrack'*. Formats: *'ros'*, or *'csv'*.
12	Code	*./sample_visualization_<format>.py*	Python scripts for data loading & visualization. All sensors rendered in one unique frame. Formats: *'ros'*, or *'csv'*.
13	Code	*./convert_rosbags_to_csv.py*	Python script for conversion of ROS bags to CSV.
14	Text	*./Readme.md*	Detailed description on how to use the dataset.

3 RESULTS

3.1 Dataset overview

The dataset contains a total of 1369 samples combining the data collected from the sensors embedded onboard the MAV, for a total of 92 GB after data compression. An overview of the files enclosed in the dataset is proposed in Table 2 (Extensions 1–14). All the data provided in the dataset are time-synchronized by means of the internal clock of the Odroid XU4 on which our ROS node for data acquisition is executed. First, the data for all sensors and all trials are provided in both ROS bag (Extension 3) and CSV (Extensions 4–7) formats, with the exception of the video recorded from the RGB camera, available under AVI format (Extension 8). A Python script is made available for conversion between ROS bags and CSV files (Extension 13). While ROS-based formatting is convenient for direct use in ROS, all data are also available in CSV format, making them easily used in standard frameworks (MATLAB, Python, C/C++, etc.). The ground truth position and orientation of the MAV w.r.t. time is provided by the OptiTrack motion capture system, and is available both in the ROS bags (Extension 3, under the topic */optitrack/pose*) and in the dedicated CSV files (Extension 7). The Cartesian coordinates of the MAV are given in meters, while the orientations are given in quarternions (a, b, c, d), with the following representation: $a + i \cdot b + j \cdot c + k \cdot d$, where (i, j, k) are the fundamental quaternion units. Lastly, an overview of the dataset is available in the *trial_overview.csv* file (Extension 2). This file contains information about the position of the obstacle in the flying arena for each sample, along with the lighting conditions (100 *Lux* vs. $1 - 3$ *Lux*). The first column provides the ID of the sample test, which can appear twice (in two consecutive rows) in case there are two obstacles in the arena. The three adjacent columns contain the (x, y, z) coordinates (expressed in m) of the obstacle given by the OptiTrack motion capture system. The next column features the lighting condition, followed by the initial rotation offset along the y axis in all OptiTrack measurements for the corresponding sample. The last column aims at telling the user whether the video from the RGB camera is available or not. Indeed, during the recordings, a total of 216 samples were saved without video recording because of hardware issues.

Content of the ROS bags – The ROS bags contain data from all sensors except the RGB camera, and including the OptiTrack ground truth data. The data are recorded within messages *'msg'* corresponding to the following *'topics'*:

(a) ***/optitrack/pose***: contains the timestamp (i.e., *msg.header.stamp*), the (x, y, z) positions (e.g., *msg.pose.position.x*, in meters), and the (a, b, c, d) quaternions (e.g., *msg.pose.orientation.x*);
(b) ***/dvs/events***: contains a list of events including the timestamp (i.e., *msg.events[i].ts*), the (x, y) pixel coordinates (e.g., *msg.events[i].x*), and the polarity (i.e., *msg.events[i].polarity*);
(c) ***/dvs/imu***: contains the timestamp (i.e., *msg.header.stamp*), the linear accelerations (e.g., *msg.linear_acceleration.x*, in m/s^2), and the angular velocities (e.g., *msg.angular_velocity.x*, in rad/s);
(d) ***/radar/data***: contains the timestamp (i.e., *msg.ts*), along with the (I, Q) vectors for both antennas (e.g., *msg.data_rx1_re*, *msg.data_rx1_im*). Each of these vectors consist of 1D arrays of 2048 float values.

Content of the CSV files – Alternatively to the ROS bags, we also provide a series of CSV files for each of the aforementioned ROS topics. For each CSV file, the first column provides the timestamp (expressed in nanoseconds). The next columns contain the sensor data:

(a) ***optitrack.csv***: first the x, y, and z positions (in meters), followed by the a, b, c, d quaternions;
(b) ***dvs.csv***: first the x and y pixel coordinates, then the polarity $p \in \{-1, +1\}$;

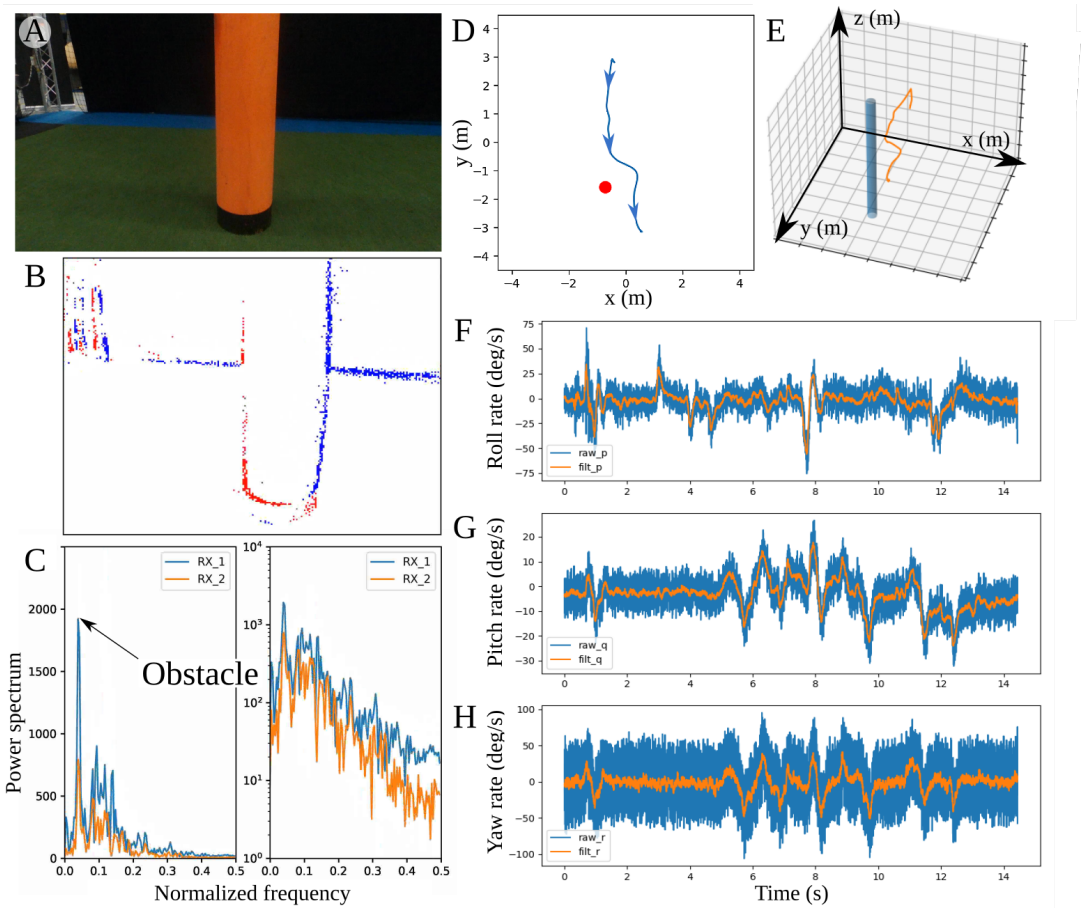

Figure 4: Visualization of a dataset sample. (A) RGB camera. (B) DVS camera. (C) Radar sensor (linear and logarithmic scales). (D-E) 2D and 3D representations of the ground truth trajectory of the MAV (OptiTrack). (F-H) IMU plots (angular rates).

(c) ***imu.csv***: first the linear accelerations (in m/s^2) and then followed by the angular velocities (in rad/s) along the x, y, and z axes;
(d) ***radar.csv***: the first 2048 columns contain *real* values from the first antenna, followed by the 2048 next columns containing the *imaginary* values for the same antenna, and so on with the second antenna.

An overview of the sensors' outputs for a given dataset sample is provided in Fig. 4 (sample no. 10). This trial has been performed under full light conditions and with one unique obstacle.

3.2 Parsing and indexing

In order to help users handling the dataset, we provide a series of Python scripts for data loading and visualization (Extensions 11, 12). In particular, for each type of sensor except for the RGB camera video, a Python script allows the user to load and plot the result of a specific sample for both ROS bags and CSV files formats. Similarly, another Python script automatically loads and displays all sensors data (including the RGB camera) in a convenient way, again for both ROS bags and CSV files.

3.3 Calibration of the visual sensors

A set of calibration files (Extensions 9, 10) are provided to the users for both cameras. The calibrations were performed using a checkerboard. In the case of the standard RGB camera, the checkerboard (8×5 squares, length: 2.4 cm) was printed and presented in front of the camera. We then use a Python script to apply the pinhole camera model and the calibration tools available in OpenCV to determine the calibration parameters. Due to the neuromorphic nature of the visual information, DVS cameras require the checkerboard to move over time. A checkerboard is automatically rendered on a computer screen using a MATLAB script (adapted from [16] and [13]). While the DVS is recording events, the checkerboard flickers at a predetermined rate. Several recordings are made from distinct positions and orientations of the sensor, while the checkerboard dimensions are kept constant. The recorded files are then processed to determine the undistorted maps for calibration.

3.4 Sensor coordinate systems

The coordinate frame conventions are shown in Fig. 5. Furthermore, a post-processing step that is applied to align the OptiTrack coordinate frame with the orientation of the MAV. This is required to

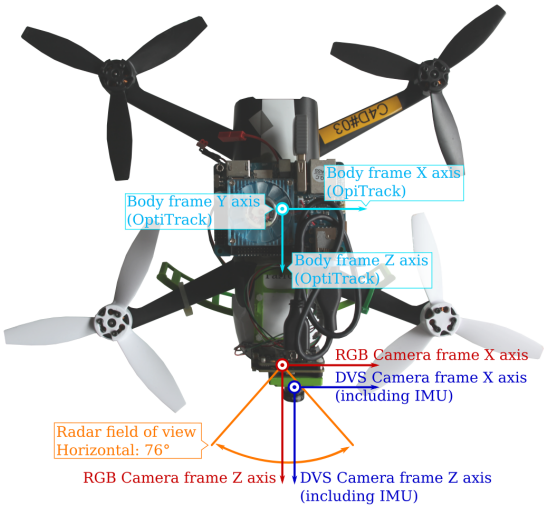

Figure 5: Coordinate systems of the drone and sensors.

compensate for the yaw offset (orientation w.r.t. the Y-axis, Fig. 2) of the OptiTrack during the initialization phase. The correction is performed every time the OptiTrack is reinitialized. The resulting correction values are provided in the dataset overview file (Extension 2) within the 6th column ('OptiTrack initial y rotation offset', values given in *rad*). To apply the correction, simply add the angle offset to the OptiTrack orientation (after quaternion-to-Euler conversion) to yield the true orientation of the MAV.

4 CONCLUSION

The Obstacle Detection and Avoidance (ODA) dataset for drones has been introduced [6]. Including raw data from a relevant range of sensors for obstacle detection, it aims at helping investigations on obstacle detection and avoidance in a challenging context (dim light condition, low-textured environment, obstacle within the drone's course), as demonstrated in recent work on radar-based obstacle avoidance [21]. In addition, the visual information available in the dataset makes it possible to use it for visual inertial odometry (VIO) and course estimation as shown in [5], thus opening a wide range of applications in autonomous navigation of drones in complex environments.

ACKNOWLEDGMENTS

This work is part of the Comp4Drones project and has received funding from the ECSEL Joint Undertaking (JU) under grant agreement No. 826610. The JU receives support from the European Union's Horizon 2020 research and innovation program and Spain, Austria, Belgium, Czech Republic, France, Italy, Latvia, Netherlands.

REFERENCES

[1] Christian Brandli, Raphael Berner, Minhao Yang, Shih-Chii Liu, and Tobi Delbruck. 2014. A 240× 180 130 db 3 μs latency global shutter spatio-temporal vision sensor. *IEEE Journal of Solid-State Circuits* 49, 10 (2014), 2333–2341. https://doi.org/10.1109/JSSC.2014.2342715

[2] Punarjay Chakravarty, Klaas Kelchtermans, Tom Roussel, Stijn Wellens, Tinne Tuytelaars, and Luc Van Eycken. 2017. CNN-based single image obstacle avoidance on a quadrotor. In *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 6369–6374. https://doi.org/10.1109/ICRA.2017.7989752

[3] Christophe De Wagter, Federico Paredes-Vallés, Nilay Sheth, and Guido De Croon. 2021. Learning fast in autonomous drone racing. *Nature Machine Intelligence* (2021), 1–1. https://doi.org/10.1038/s42256-021-00405-z

[4] Jeffrey Delmerico, Titus Cieslewski, Henri Rebecq, Matthias Faessler, and Davide Scaramuzza. 2019. Are We Ready for Autonomous Drone Racing? The UZH-FPV Drone Racing Dataset. In *IEEE International Conference on Robotics and Automation (ICRA)*. 6713–6719. https://doi.org/10.1109/ICRA.2019.8793887

[5] Raoul Dinaux, Nikhil Wessendorp, Julien Dupeyroux, and Guido De Croon. 2021. FAITH: Fast Iterative Half-Plane Focus of Expansion Estimation Using Optic Flow. *IEEE Robotics and Automation Letters* 6, 4 (2021), 7627–7634. https://doi.org/10.1109/LRA.2021.3100153

[6] Julien Dupeyroux, Raoul Dinaux, Nikhil Wessendorp, and Guido De Croon. 2021. The Obstacle Detection and Avoidance Dataset for Drones. https://doi.org/10.4121/14214236.v1

[7] Guillermo Gallego, Tobi Delbruck, Garrick Orchard, Chiara Bartolozzi, Brian Taba, Andrea Censi, Stefan Leutenegger, Andrew Davison, Jörg Conradt, Kostas Daniilidis, and Davide Scaramuzza. 2020. Event-based Vision: A Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2020), 1–26. https://doi.org/10.1109/TPAMI.2020.3008413

[8] Daniel Gehrig, Michelle Rüegg, Mathias Gehrig, Javier Hidalgo-Carrió, and Davide Scaramuzza. 2021. Combining Events and Frames Using Recurrent Asynchronous Multimodal Networks for Monocular Depth Prediction. *IEEE Robotics and Automation Letters* 6, 2 (2021), 2822–2829. https://doi.org/10.1109/LRA.2021.3060707

[9] Elia Kaufmann, Antonio Loquercio, René Ranftl, Matthias Müller, Vladlen Koltun, and Davide Scaramuzza. 2020. Deep Drone Acrobatics. In *Proceedings of Robotics: Science and Systems*. Corvalis, Oregon, USA, 1–10. https://doi.org/10.15607/RSS.2020.XVI.040

[10] Young K Kwag and Chul H Chung. 2007. UAV based collision avoidance radar sensor. In *2007 IEEE International Geoscience and Remote Sensing Symposium*. IEEE, 639–642. https://doi.org/10.1109/IGARSS.2007.4422877

[11] Patrick Lichtsteiner, Christoph Posch, and Tobi Delbruck. 2008. A 128x128 120 dB 15 us latency asynchronous temporal contrast vision sensor. *IEEE Journal of Solid-State Circuits* 43, 2 (2008), 566–576. https://doi.org/10.1109/JSSC.2007.914337

[12] Elias Mueggler, Guillermo Gallego, Henri Rebecq, and Davide Scaramuzza. 2018. Continuous-time visual-inertial odometry for event cameras. *IEEE Transactions on Robotics* 34, 6 (2018), 1425–1440. https://doi.org/10.1109/TRO.2018.2858287

[13] Elias Mueggler, Basil Huber, and Davide Scaramuzza. 2014. Event-based, 6-DOF pose tracking for high-speed maneuvers. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2761–2768. https://doi.org/10.1109/IROS.2014.6942940

[14] Federico Paredes-Vallés and Guido De Croon. 2021. Back to event basics: Self-supervised learning of image reconstruction for event cameras via photometric constancy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 3446–3455.

[15] Dário Pedro, André Mora, João Carvalho, Fábio Azevedo, and José Fonseca. 2020. ColANet: A UAV Collision Avoidance Dataset. In *Doctoral Conference on Computing, Electrical and Industrial Systems*. Springer, 53–62. https://doi.org/10.1007/978-3-030-45124-0_5

[16] Bas J Pijnacker Hordijk, Kirk YW Scheper, and Guido De Croon. 2018. Vertical landing for micro air vehicles using event-based optical flow. *Journal of Field Robotics* 35, 1 (2018), 69–90. https://doi.org/10.1002/rob.21764

[17] Valentin Riviere, Augustin Manecy, and Stéphane Viollet. 2018. Agile robotic fliers: A morphing-based approach. *Soft Robotics* 5, 5 (2018), 541–553. https://doi.org/10.1089/soro.2017.0120

[18] Cedric Scheerlinck, Henri Rebecq, Daniel Gehrig, Nick Barnes, Robert Mahony, and Davide Scaramuzza. 2020. Fast image reconstruction with an event camera. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 156–163.

[19] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. 2012. Indoor segmentation and support inference from rgbd images. In *European Conference on Computer Vision (ECCV)*. Springer, 746–760. https://doi.org/10.1007/978-3-642-33715-4_54

[20] Shuran Song, Samuel P Lichtenberg, and Jianxiong Xiao. 2015. Sun rgb-d: A rgb-d scene understanding benchmark suite. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 567–576.

[21] Nikhil Wessendorp, Raoul Dinaux, Julien Dupeyroux, and Guido De Croon. 2021. Obstacle Avoidance onboard MAVs using a FMCW RADAR. *arXiv preprint 2103.02050* (2021).

[22] Linhai Xie, Sen Wang, Andrew Markham, and Niki Trigoni. 2017. Towards monocular vision based obstacle avoidance through deep reinforcement learning. *arXiv preprint 1706.09829* (2017).

[23] Hang Yu, Fan Zhang, Panfeng Huang, Chen Wang, and Li Yuanhao. 2020. Autonomous Obstacle Avoidance for UAV based on Fusion of Radar and Monocular Camera. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 5954–5961. https://doi.org/10.1109/IROS45743.2020.9341432