DEEP CONVOLUTION NEURAL NETWORK (CNN) USING A MONOCULAR CAMERA AND A 2D LIDAR FOR ROBUST QUADROTOR OBSTACLE AVOIDANCE

Yuthika Sagarage, J. K. Jayawardena, Malith Kuruppu
Communication Division
Arthur. C. Clarke Institute for Modern Technologies
Katubedda, Moratuwa, Sri Lanka e-mail:
kavindra@accmt.ac.lk

***Abstract*— Quadrotor platforms have become a vital tool in the industry as it is being used as the primary go to technology approach across many sectors including law-enforcement, survey, photography, videography, inspection and etc. However, most users in these industries are not trained or certified pilots, hence introducing safety cautions of crashing drones generated by piloting errors in urban environments causing damage to property and human lives. One of the ways to mitigate such crashes is by introducing robust obstacle avoidance capabilities to the drones in manual control mode. We propose a robust combination technique approach for obstacle avoidance on a quadrotor by a 2D LIDAR and a forward-facing monocular camera using them as a singular system to achieve a higher level of perception about the surrounding environment. 2D LIDAR is used to get an accurate dynamic two-dimensional map of the surrounding environment and the monocular forward-facing camera is used to implement a deep convolution neural network which acts as a classifier network to identify whether the path ahead is safe to continue or blocked. Training data is taken indoors and outdoors for a better generalization for the classifier. We demonstrate experiments showing the capability of this robust obstacle avoidance technique on dynamic obstacles and on static obstacles in both simulated environments (gazebo) and real-world scenarios.**

Keywords—Obstacle avoidance; Quadrotor; Deep Learning; Aerial Robotics;

I. INTRODUCTION

Tackling the problem of obstacle avoidance in robotics platforms goes back to the origins of the robotic navigation and perception. However, accomplishing this task to be used in a commercially viable case in an actual robotics platform such as a drone is still an ongoing research problem. As of today, drones are an emerging technology which encapsulates many industries to its grasps such as videography, photography, surveillance, inspection [1], survey, law-enforcement and etc. In many of these occasions, the drone/quadrotor platforms are being controlled manually by the pilot except for survey, delivery or any other autonomous application which is navigated using

GPS points. These manually piloted drones are frequently used in urban areas, cluttered spaces and beyond line of sight missions, which introduce a safety concern as these environments most probably have dynamic and static obstacles which cannot be observed or anticipated by the pilot.

Classical approaches that have been implemented to tackle the problem of obstacle avoidance are mostly based ultrasonic sensors [2][3], time of flight range sensors, stereo cameras [4] and recently, there is a trend to incorporate deep learning [5-7] aspects in order to address the problem.

However, most of the machine learning based solutions addresses the problem for autonomous mode obstacle avoidance which competes with vector polar histogram (VPH), vector field histogram (VFH), VFHF, and VPH+ [8-11] like classical algorithms for autonomous collision avoidance and trajectory generation where a goal is set in a known or an unknown map, the robot platform or the drone should reach the goal without crashing hence avoiding obstacles. However, in this scenario, control of the drone has to be given to the user just after making sure that the drone is safe after avoiding the obstacle, which introduces its own set concerns such as limiting the speed of the drone when a potential obstacle candidate is detected.

Most of the classical approaches which only use 1D or 2D distance measurements to the obstacles have their limitations according to the sensor data [2][3]. For an example, using the common 30-degree field of view (FOV), ultrasonic sensors are not a viable solution since it outputs a value of the nearest obstacle and it does not sufficiently cover around the whole quadrotor body. Same limitations and more apply for the stereo camera solutions, such as noisy measurements in low light condition, the heavy computational power to retrieve sensor data as a 3D point cloud and computation and the physical constraints to cover the whole quadrotor body.

In order to successfully avoid obstacles, the drone has to perceive the obstacles around it and respond accordingly to maintain a safe distance to the perceived obstacle/s. As the most of the manually controlled drones are usually piloted in the heading direction of the drone, we used a monocular camera to retrieve information of the front of the drone at an FOV of (75 degrees, 75 degrees), which then fed to a deep convolutional neural network to assess the collision probability. In order mitigate the limitations of not having the awareness around the propeller plane; we use a 2D LIDAR to get the 2D plane information as near as to the propeller plane to calculate the potential obstacles in the global map. In order to assess the direction controls in Z direction, we use a distance sensor which performs at around 40m height at a 500Hz to sense the ground distance.

II. METHODOLOGY

We propose a robust combination technique for obstacle avoidance where it acts as an assistive system for the drone operator which consists of two main nodes, i.e obstacle detection, and obstacle avoidance. Obstacle detection node consists of prediction of a collision by the front camera and taking 2D LIDAR data, as a 2D point cloud to get spatial awareness of the propeller plane of the drone and also a downward facing distance sensor to maintain the accuracy of the altitude measurement of the drone. As this requires a heavy computation to run an inference inside the drone itself [14], a Jetson TX2 CUDA compatible processor is used with Auvidea J120-rev8 carrier board. A HD 1080p un-branded USB camera and a RPLIDAR A2 are used to get the real-time image feed and point cloud data respectively. Our hardware setup, shown in Fig. 1, consists of a 680-size quadrotor with open source Pixhawk FMUv3 autopilot which runs on PX4 [15] 1.8.2 firmware, which ensures the highest middleware compatibility to pass control commands to the drone and also the simulation capabilities in software in the loop mode [14]. The downward facing distance sensor is a lidarlitev3 provides accurate distance measurement up-to 40 meters where the measurements are passed on through extended Kalman filter on the Pixhawk to run a sensor fusion [14] for higher precision.

Fig. 1: Hardware Setup of the Quadrotor

Robotic Operating System (ROS) is used as the middleware for publishing and subscribing real-time data with the quadrotor flight control system using the Mavlink protocol with a data rate around 18-20Hz. MAVROS package which is an open ROS package developed by PX4 community provides a higher-level API calls to subscribe and publish data to the pixhawk is used to issue Mavlink protocol control commands from the JetsonTX2, obstacle avoidance algorithm.

Vision/perception is realized through several ROS nodes. East North Up (ENU) body-frame coordinate system is used in vision computations and MAVROS translates it to the (North East Down) NED coordinate system which is used by the PX4 firmware. The classifier inference node which uses CUDA processing runs to detect objects in the path, to translate the quadcopters position in the Z body-frame axis. This CNN model has a ResNet [16] architecture to recognize the objects in the visual range of the camera and provide with a collision probability. 2D LIDAR, which percepts 360 degrees in the propeller plane is used to ensure that the system is robust and accounts for the dynamic obstacle to some extent.

A high-level system diagram is shown in the Fig. 2, explains the software architecture of the complete system. Based on the vision/perception outputs of the camera, 2D LIDAR, Range finder and the internal IMU the controlling is handled by a ROS node which is implemented through a base C++ class implementation issues command direction to the quadrotor for the next time-step. In order to assure that the quadrotor is able to move in tight spaced environments, (e.g. corridors, mines, etc.) the navigation algorithm predicts the path dynamically to optimize the future route of the quadrotor using the inputs of the 2D LIDAR and the camera.

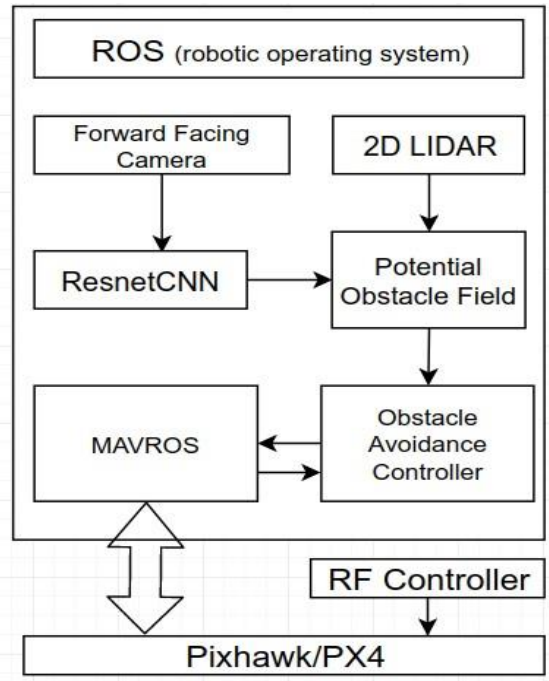

Fig. 2 :High level System Diagram

A. *ResNet implementation*

ResNet [16] architecture is one of the more promising network architectures which allow implementing deeper networks, where the core of its architecture lies in a so-called

identity connection which addresses vanishing gradient problem, however in shallow networks the 1x1 convolution shortcut heavily helps in optimization thus allowing for better generalization [18]. Residual blocks of the ResNet, by He et al. [16], are shown in the Fig. 3. This kind of architectures has been previously implemented for shallower networks successfully for autonomous navigation tasks [7]. We implement a similar shallow ResNet architecture with 12 layers, QuadNet-mini with total Trainable parameters of 90,000, Fig. 4. which is computationally efficient for it to be able to implement it on the companion computer with limited floatingpoint operations per second and memory bandwidth. We input a 224x224 image as the input of the layer to be processed by the QuadNet-mini.

The dataset Table1 was obtained by flying a go-pro camera in outdoors and indoors with a quadcopter for better generalization. We annotated the camera images as 1 and 0 by assigning 1 for the images which are near to the obstacles and 0 for the images which are far from the obstacles. The dataset was given among five annotators, so that annotations also become somewhat in-discriminated by annotator's intuition, thus will again help in better generalization. Also, additionally, we used some of the publicly available DroNet [7] dataset to train the model which comprised of data with human intervention and urban environments.

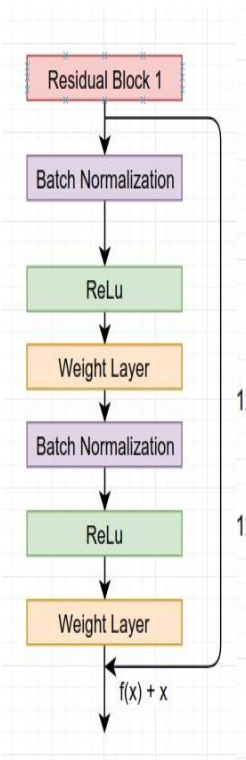

Fig. 4 : Residual Block

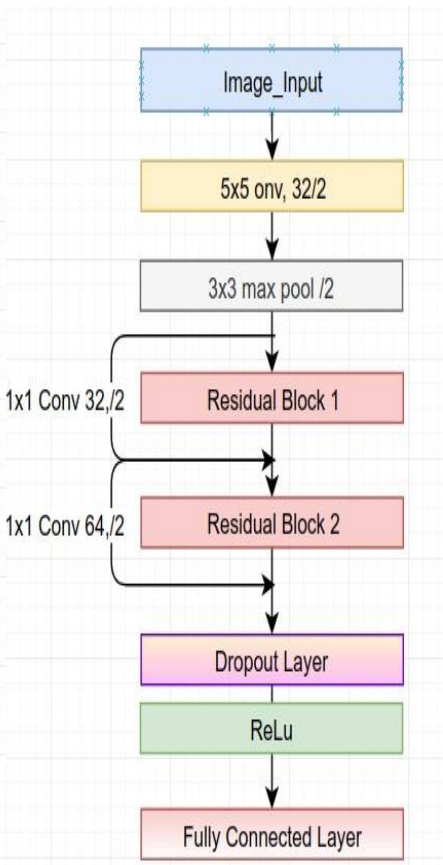

Fig. 3: Proposed QuadNetmini

B. *Potential Obstacle Field Implementation using 2D-LIDAR and QuadNet-mini*

Given the 2D-LIDAR (RPLidar A2) data a Potential Obstacle Field was calculated by using a simple equation of motion to reduce the maximum speed of the drone by changing a PX4 parameter by mavros-mavparam API call. This call is only done for three speed settings to reduce the maximum speed, in position hold mode. In this experiment, RPLidarA2 provides 12 m range, by considering all the range data a global map is built, this global map update which consists of a histogram is used to decide the maximum speed of the drone. For the 680-size quadcopter frame with tarot 4114 motors with a weight of 2.6 kg, we estimated around 2.3, 4.1, 5 ms^{-1} as the three speed settings. This heavily depends on the maximum acceleration/deceleration of the quadcopter and the LIDAR range, hence it should be calculated by the Eq. (1) – (3), given that the deceleration is known. This equation is derived by the basic equation of motion $vv^2 = uu^2 + 2aaaa$ assuming that LIDAR can only get data up to a 12m range.

Eq.1 $U_{\max}\sqrt{24a}$ s >= 12;

Eq.2 $U_{max}= 4a$ s >= 8;

Eq.3 $U_{max}\sqrt{8a}$ s >= 4;

Maximum acceleration values can be obtained by manually flying the quadcopter and logging the data. Further, there is a need to minimize these envelopes to account for the quadcopter size and a safety margin. In this test case, it is selected as 1m.

As distance and angle data is subscribed from the 2d-lidar, as a ros msg type. We divide the LIDAR data by π/4 segments, which results in 8 quadrants**.** Which in turn used to build the offboard autonomous repelling avoidance mode with fusing the collision probability output of the QuadNet-mini.

The pseudo code for the ros nodelet executing collision avoidance is shown in Table 2. We compute a thresholding value using a function which depends on the collision probability and the distance array, and also depending on the speed and acceleration of the drone at that moment, we compute a breaking distance value, which then feeds into a C++ base class which , which issues command directions to the quadrotor for the next time-step. In order to assure that the quadrotor is able to move in tight spaced environments, (e.g. corridors, mines, etc) the navigation algorithm predicts the path dynamically to optimize the future route of the quadrotor using the inputs of the 2D LIDAR and the camera. In manual mode, the implemented obstacle avoidance system gives the control back to the user by putting the drone to position/altitude hold mode; this will assure that the user is able to control the quadrotor

Table 1:Dataset structure

	Table1. The Number of training	
	Training	***Validation***
Dataset 1	~25000	~2500
Dronet Dataset	~10000	~1000

Table 2 : Pseudo code for fusing 2D Lidar and QuadNet-mini

Algorithm 1: Pseudo Code for fusing 2D Lidar and QuadNet-mini

```
breaking distance;
collision probability;
lidar_distance ;
lidar_quadrant [];
while in position hold mode do
    for in lidar_quadrant [];
        check lidar range;
        i = lidar_distance[min]
        check collision probabilty
        calculate breaking distance
        calculate threshold
    if collision probability > threshold && lidar distance > breaking distance then
        take control (offboard_mode)
        move in z direction;
    if collision probability > threshold && lidar distance < breaking distance then
        take control (offboard_mode)
        repel in lidar_quadrant[i]
    end
end
```

III. RESULTS AND DISCUSSION

The QuadNet-mini is trained around 100 epochs to get the best F-1 Score of 0.93 and avg accuracy of .967 and a precision of .991 using a NVIDIA Quadro 5000 at the 86th epoch, which is an adequate result for a smaller size network. This can be seen in the tensorboard while training the network, Fig. 5(a,b,c,d)

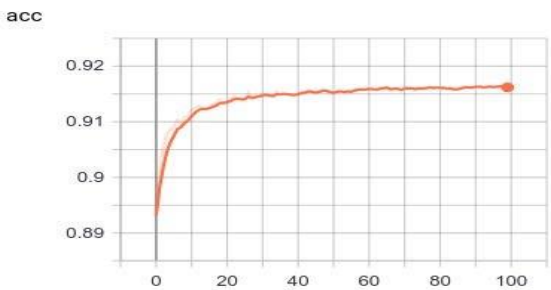

Fig. 5-a: Training Accuracy vs Epoch Number

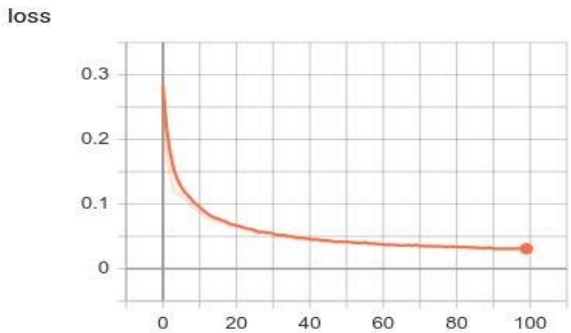

Fig. 5-b: Training Loss vs Epoch

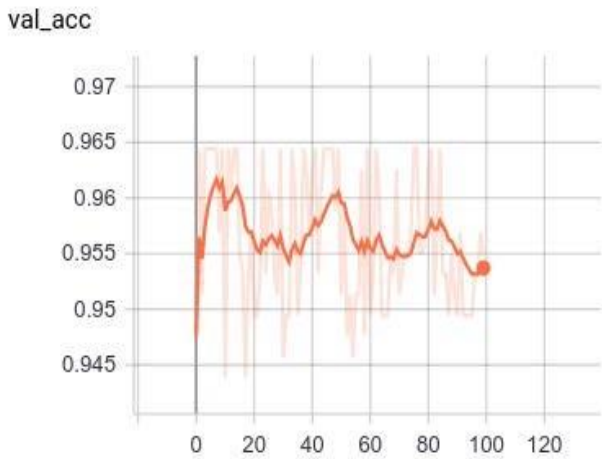

Fig. 5-c Validation Accuracy vs Epoch

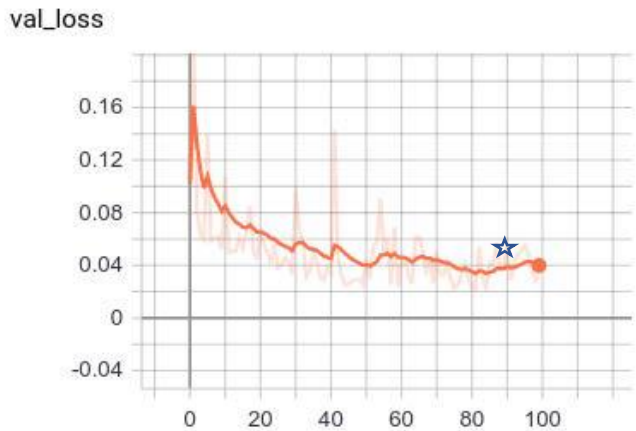

Fig. 5-d Validation Loss vs Epoch

After the training the model is then tested outdoors by walking predictions. In an Intel Xeon and Quadro-5000 GPU laptop the with a laptop webcam to get a qualitative view of the model's model performed at almost 28-30 fps which is to be expected. Then we deployed the model on the NVIDIA Jetson TX2 which also gave adequate results up to 15-20 fps.

The model is ported to a ROS package using ROS-Keras. Then an OpenCV cv_bridge was used to feed the images to the model for the prediction, where the model outputs a collision probability. Then the collision probability is packaged as a ROS_msg and published for the threshold calculation in C++ base class. The whole system is then checked with in the gazebo environment before deploying. Fig. 6**,**

We demonstrated experiments showing the capability of this robust fusion technique at dynamic and static obstacles around 10-15km/h scenarios. However, the need to evaluate the method in more environments such as rain, fog etc. are in need before deploying the method in a commercial/client application.

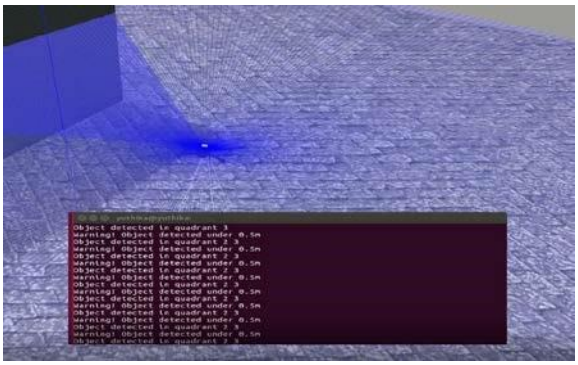

Fig. 6:Obstacle avoidance testing on gazebo environment

Nevertheless, it has adequate results to be deployed without major changes. Frame rates for running the inference model in Jetson TX2 can be significantly improved by wrapping the model in a TensorRT framework.

2D LIDAR might yet become an expensive option in the coming years as more features and architectures improve the whole deep learning [17] paradigm. Eliminating the need for a LIDAR is the go-to option if it's achievable, but however given the compute capabilities as of more computationally expensive to be used in a manual mode obstacle avoidance, since the scenario is bit different than the plan -map-explore kind of autonomy since it has to adapt the pilots controls up to a some point before getting the control of the drone.

IV. Conclusion

In this paper, we demonstrate a methodology which tackles the manual mode avoidance scenarios, using a fusion between traditional and deep neural network-based method using a 2D LIDAR and a forward-facing camera as the main input. This allows a user or operator to successfully operate the drone while the drone takes care of the obstacle avoidance. As the robotic perception improves, it naturally tends towards computer vision and machine learning approaches. Also, many of the researchers are trying to implement reinforcement learning approaches [17] and also methods with mapping [18] such as DSO [20]. In future work, an assistive approach using SLAM and CNN based algorithms may present more possibilities like teleoperation through buildings, debris or a forest, where the user might not be able to perceive the environment but wants to go through some obstacle course.

Acknowledgment

The project is funded as an internal research at Arthur C. Clarke Institute for Modern Technologies, through Science Research and Technology Ministry of Sri Lanka.

References

[1] N. Michael et al., "Collaborative mapping of an earthquake-damaged building via ground and aerial robots," J. Field Robot., vol. 29, no. 5, pp. 832–841, 2012.

[2] Prathamesh Salaskar, Saee Paranjpe, Jagdish Reddy, Arish Shah., "Quadcopter – Obstacle Detection and Collision Avoidance" IJETT., vol. 17, no 2, – 2014

[3] Ohya, I., Akio Kosaka, and Avi Kak. "Vision-based navigation by a mobile robot with obstacle avoidance using single-camera vision and ultrasonic sensing." IEEE Transactions on Robotics and Automation 14.6(1998):969-978

[4] Sadolikar, M. R., & Bhaskar, P. P. (2014). Obstacle Detection and Avoidance using Stereo Vision System with Region of Interest (ROI) on FPGA. International Journal of Engineering Research & Technology (IJERT).

[5] D. Gandhi, L. Pinto, and A. Gupta, "Learning to fly by crashing," in Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS), Sep. 2017.

[6] A. Giusti et al., "A machine learning approach to visual perception of forest trails for mobile robots," IEEE Robot. Autom. Lett., vol. 1, no. 2, pp. 661–667, Jul. 2016

[7] Antonio Loquercio, , Ana I. Maqueda, Carlos R. del-Blanco, Davide Scaramuzza "DroNet: Learning to Fly by Driving" ," IEEE Robot. Autom. Lett., vol. 3, no. 2, pp. 1088-1095, April 2018.

[8] A. Dong, W. Hong, "VPH: a new laser radar based obstacle avoidance method for intelligent mobile robots", WCICA 2004. Fifth World Congress on Intelligent Control and Automation, vol. 5, pp. 4681-4685, 2004.

[9] Borenstein, J. and Koren, Y., "The Vector Field Histogram - Fast Obstacle Avoidance for Mobile Robots", IEEE Journal of Robotics and Automation, June 1991, Vol. 7, No. 3, pp. 278-288

[10] Xiuzhi Li, Yue Gong, Songmin Jia, Huan Qiu, "VFHF: A combining obstacle avoidance method based on laser rangefinder", Information and Automation (ICIA) 2016 IEEE International Conference on, pp. 825-830, 2016.

[11] Jianwei Gong, Yulin Duan, Yiming Man, Guangming Xiong, "VPH+: An Enhanced Vector Polar Histogram Method for Mobile Robot Obstacle Avoidance", Mechatronics and Automation 2007. ICMA 2007.

[12] International Conference on, pp. 2784-2788, 2007Pham, Hung et al. "A survey on unmanned aerial vehicle collision avoidance systems." CoRR abs/1508.07723 (2015): n. pag.

[13] Ramasamy, Subramanian and Roberto Sabatini. "A unified approach to cooperative and non-cooperative Sense-and-Avoid." 2015 International Conference on Unmanned Aircraft Systems (ICUAS) (2015): 765-773.

[14] Smolyanskiy, Nikolai et al. "Toward low-flying autonomous MAV trail navigation using deep neural networks for environmental awareness." 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (2017): 4241-4247.

[15] L. Meier, P. Tanskanen, F. Fraundorfer, and M. Pollefeys. PIXHAWK: A system for autonomous flight using onboard computer vision. In ICRA, May 2011

[16] He, Kaiming et al. "Deep Residual Learning for Image Recognition." 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016): 770-778.

[17] Chakravarty, Punarjay et al. "CNN-based single image obstacle avoidance on a quadrotor." *2017 IEEE International Conference on Robotics and Automation (ICRA)*(2017): 6369-6374.

[18] Alvarez, H. et al. "Collision Avoidance for Quadrotors with a Monocular Camera." ISER (2014).

[19] Israelsen, Jason et al. "Automatic collision avoidance for manually teleoperated unmanned aerial vehicles." *2014 IEEE International Conference on Robotics and Automation (ICRA)* (2014): 6638-6643.

[20] Engel, Jakob et al. "Direct Sparse Odometry." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40 (2018): 611-625.