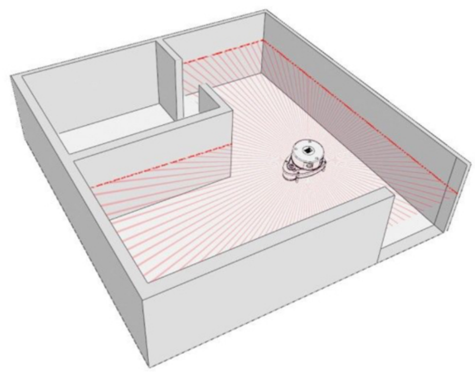
LiDAR-guided Obstacle Avoidance in a Quadcopter Drone

Adam Wathieu, Brian Cruz, Olu Roy, Kamel Saidi, Ph.D.

National Institute of Standards and Technology
U.S. Department of Commerce



Background: As drones become more and more popular, autonomous drone technology has become inevitable and very much needed. Autonomous systems in this realm would be idle for the use by first responders. Police officers, for example, could avoid blind spots with an autonomous drone while surveying the outside of a dangerous property. The fire department could use this technology to guide drones through dangerous and narrow subway tunnels to survey fire-damaged areas. In this work, we present a method by which a drone can stop autonomously if it is approaching an obstacle from any direction. Although this is not complete autonomy, we hope that it serves as motivation for continued research in the realm of autonomously flying drones.

Methods: For our research, we used the DJI Matrice 100 drone, and have attached an RP LiDAR A2 unit to the top of the drone via a 3D printed mounting plate. The LiDAR sensor works by firing laser pulses and measuring the delay until they are reflected back. The RPLiDAR C++ library provides an RPlidarDriver object that can be used to start and stop the unit and gather data. Data is returned as an array of objects containing the angle, distance, and quality of each measurement. The drone provides an Onboard SDK that developers can run on Windows or Linux. The library allows for communication with the Matrice 100's computer over a serial bus. The onboard computer can then give the drone commands, such as going to a certain position relative to its current location, taking off, or landing. We used a Raspberry Pi 3 and connected to it with SSH from a laptop to run our obstacle avoidance program. When the LiDAR registers objects within 1 meter of itself, the Raspberry Pi initiates a stopping function, and in turn the drone temporarily overrules the commands of the pilot to stop and hover in place in order to avoid hitting the object that was approaching.

Results: In initial testing in the computer simulator, we observed that some nodes with low quality data would appear in each cycle, which would register as 0 distance. The project worked as expected in real life, but raised some more issues that we didn't catch in the simulator. Lack of control can make the pilot feel anxious, and the drone can sometimes malfunction. For example, in our second trial, the drone drifted when it stopped. At this point, the pilot had no control, and the drone drifted into a wall and two of the propellers were damaged.

Conclusion: To circumvent the issue of registration of low quality data, we divided each cycle of the LiDAR into batches of 8 measurements and took the average of the high quality measurements in each batch. The second problem can be addressed in the future by either flying the drone outside, where the position control will improve due to connection with GPS, or changing how the program alerts the user so that they retain control the entire time. More can be done with this project. Implementation of a function for speed of the drone versus distance needed to stop will help for drones flying at greater speeds, in which case 1 meter to stop would not be enough. Also, implementation of a second LiDAR on the vertical axis would enable the drone to avoid obstacles above and below the drone, rather than simply on its horizontal axis.