Drone Search and Rescue!

Rohan Ondkar
College of Computer, Mathematical, and Natural Sciences
University of Maryland, College Park
Maryland, United States
rondkar@terpmail.umd.edu

Naveen Harish
College of Computer, Mathematical, and Natural Sciences
University of Maryland, College Park
Maryland, United States
nharish@terpmail.umd.edu

Rahul Shah
A. James Clark School of Engineering
University of Maryland, College Park
Maryland, United States
rshah135@terpmail.umd.edu

Judy Song
College of Computer, Mathematical, and Natural Sciences
University of Maryland, College Park
Maryland, United States
jsong902@terpmail.umd.edu

Paulius Vilimas
College of Computer, Mathematical, and Natural Sciences
University of Maryland, College Park
Maryland, United States
pvilimas@terpmail.umd.edu

***Abstract*—This project presents an approach to navigate a drone though obstacles and reach the end target.**

I. Problem Statement

In this project, we aim to have our simulation drone travel to an end target and avoid any and all obstacles in the way. The algorithm is tested on a simulation which has trees as obstacles, and the drone travels through a forest to reach the target.

II. The Approach

In order to detect obstacles, we checked the top left corner, (coordinate (1,1)) to the top right, (coordinate (1,240)) to determine the depth value accordingly. The higher the depth value, the closer the obstacle is to the camera, thus visually darker. To enable the drone to physically avoid the obstacles, it moves left if the obstacle is to the right and right if the obstacle is to the left. The algorithm can be described below:

if $depth(1,1) < 0.1$ **then**
  $VelX = 0.5$
else if $depth(1,240) < 0.1$ **then**
  $VelX = -0.5$
else
  Substract cylinder location from the camera location
end if

As seen, the x-component of the velocity is set to a positive or negative amount based on whether the object is detected to the right or left of the drone.

III. Failure Cases

There were issues with starting off this project. We were not able to get OpenEXR installed on most devices, but after numerous different methods, we were able to get OpenEXR on Naveen's personal computer. Aside from this, it was difficult to determine where to begin with the code algorithmically, as we were all unfamiliar with the work space. Another major error we were unable to resolve in the code was the drone moving all the way to the left and hitting the wall, as it would keep bouncing off the wall on and continue straight.

IV. Sample Images

Fig. 1. Obstacle avoidance

V. Work Distribution

Because Naveen was able to get OpenEXR installed, he worked primarily on the code itself. Rohan and Paul were also a part of this, as the three of them are roommates and were able to have constant collaboration. Thus, Judy and Rahul worked primarily on the LaTeX portion. The entire team however, bounced ideas off each other to figure out the necessary algorithms to complete this project. We also collaborated with Charles Parrot and his group for the algorithm to move the drone left or right, depending on where the detected obstacle was.

References

[1] JokerMartiniJokerMartini4, anilsathyan7anilsathyan7 66688 silver badges1919 bronze badges, amp; SpencerSpencer (1966, September 1). Convert EXR to JPEG using imageio and python. Stack Overflow. Retrieved October 15, 2021, from https://stackoverflow.com/questions/50748084/convert-exr-to-jpeg-using-imageio-and-python.

[2] Python: Thresholding techniques using opencv: Set-1 (simple thresholding). GeeksforGeeks. (2021, October 12). Retrieved October 15, 2021, from https://www.geeksforgeeks.org/python-thresholding-techniques-using-opencv-set-1-simple-thresholding/.