DRDO UAV-UGV

Team - 13

Overview

As described in the problem statement, our team had to map an area via UAV aerial imagery and aid a UGV in navigating a complex static environment. The task is considered completed when the UGV successfully navigates across the terrain to the end of the road with aid from the UAV.

Approaches Piloted:

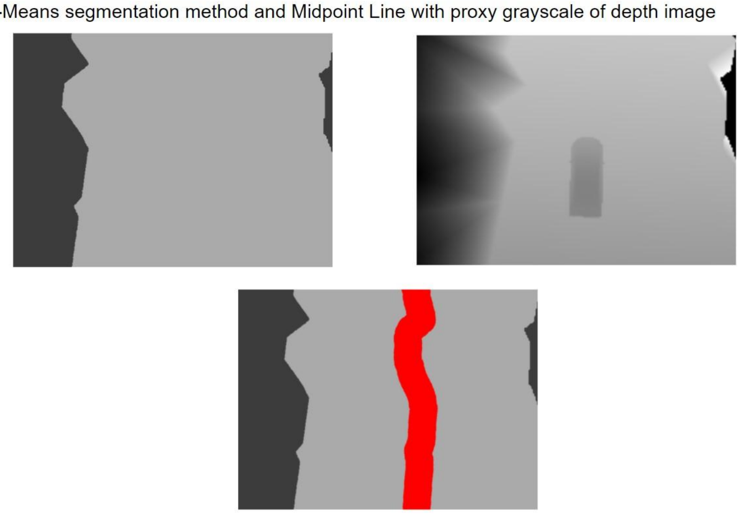
- **Initial Approach :**

    - Initially, we made the iris quadcopter take off to nearly 18m. Next, we use the images obtained from the depth camera to segment the hill slope, road, and steep valley into different classes.
    - Further at every class transition contours are drawn
    - Henceforth, we used the OpenCV library to find the road's midpoint from the contoured image and drew a line joining the nuclei of the contours.
    - The drone is now made to follow the line created in the previous step as a guide. However, this approach didn't perform well at some points where class transitions were not significant enough to distinguish, and shaky camera(In spite of the gimbal being present) added more to this approach's demise as contours were not detected in some places and setting default values proved to be a bit ambiguous.
    - The takeaway from this process is that it inspired us to work with the midpoint following method, which worked well for a few places in this algorithm.

- **Final Approach :**

    - We initialize the iris drone to take off to 15m. Similar to the previous algorithm, images are obtained from the depth camera, which is further converted into grayscale images for the OpenCV library to intake and work with.
    - The grayscale image obtained from the depth image conversion is then segmented into two different classes using K-means segmentation. The two classes are 1. Road and 2. Not-Road. The same midpoint approach is then used to form the line joining midpoints of the Road class vertically from the viewer's perspective.
    - The drone is then autonomously operated by an algorithm that makes it follow the midpoint line formed in the previous step, and it continues to do so. To determine which of the obtained classes is Road, we assume that the class with dominating occurrence will correspond to the road class as ultimately Drone starts just above the road and has to follow the same throughout.
    - A center section(*2:10* ratio ⇒ *section:Image*) of the depth image obtained in the first step is marked as ROI(region of interest), the maximum value of which is assumed to be the height of the drone above ground level at its position. The algorithm mentioned in the previous step is built to keep the drone between chosen upper and lower threshold values, thus maintaining the drone at a constant height range above the road level.

K-Means segmentation method and Midpoint Line with proxy grayscale of depth image

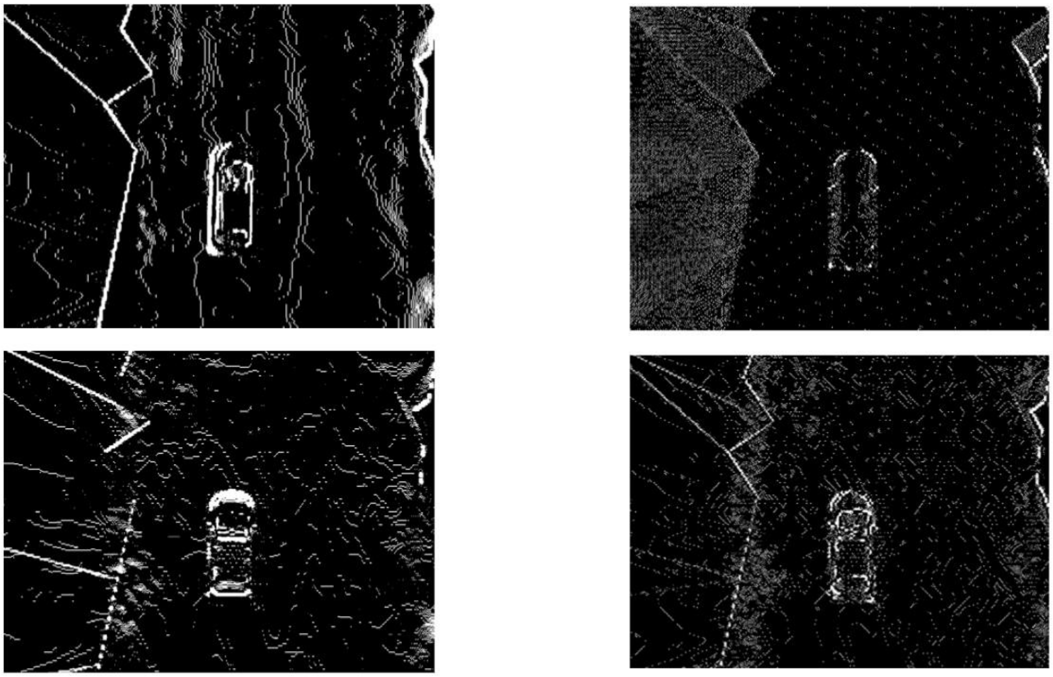


- And thus, using this approach, we traverse the entirety of the ROS world through the iris drone.

Image Processing Algorithms Tried:

- Canny Edge detection was applied in the initial trial where our team thought of using the camera mounted on the drone to achieve RGB images from it. Sobel images even provided good boundary detection, but the approach turned futile, keeping in mind the testing on a textureless white overlaid world.
- Further, we also tried Sobel Methods in the initial stages of ideation and testing as they were computationally less expensive than canny edge detection. Still, these were dropped for the same later reasons as the above method.
- Finally, we ideated using the depth image obtained from the depth camera mounted on the iris drone as a proxy grayscale image, which would give consistent images when the drone is maintained in an optimum height range. We applied Background subtraction with segmentation which failed due to less control over the camera and its shaky behavior.
- Adaptive thresholding methods were not implemented because those neither provided for good distinct classes nor good contours.
- Finally, we applied K-means segmentation on the proxy grayscale images obtained from the depth image normalization. It worked well in segmenting the image portions into two classes of Road and Not-Road. Thus, it was introduced in the final approach aforementioned.

Sobel Methods on Proxy RGB and RGB Image



Usage of ROSBAG for Communication:

- A rosbag or bag is a file format in ROS for storing ROS message data. These bags are often created by subscribing to one or more ROS topics, and storing the received message data in an efficient file structure. We can playback the bag file to publish all the topics which were recorded.
- We have recorded the `/mavros/setpoint_velocity/cmd_vel_unstamped` topic of the drone after the takeoff until it starts landing into a rosbag file.
- After the drone landed, we started playing the ROSBAG file and subscribed to the `/mavros/setpoint_velocity/cmd_vel_unstamped` topic which is being published from the ROSBAG file. Then we modified the received velocity msgs into msg format of /prius topic and published it to move the prius.
- Right now by subscribing to the /prius topic, we are using only the throttle and steer, we can also calculate the slope of the road by recording the vertical height of the drone from the road, so that we can use brakes depending on the slope and make it more generalized.

Packages Used:

- Ubuntu 18.04
- ROS Melodic
- Gazebo 9
- Python
- ArduPilot Firmware
- Dronekit 2.9.2
- OpenCV 4.5.5
- PIL