Individual Lab Report - 09

Team D: CuBi

Nithin Subbiah Meganathan

Teammates:

Bobby

Laavanye Bahl

Paulo Camasmie

Jorge Anton Garcia

October 24th, 2019

Carnegie Mellon University

Individual Progress

The obstacle detection pipeline was conceptualized and designed in the past two weeks. Currently, the state machine does not have obstacle detection, exploration, and planning components in it. As they are currently being independently developed, how all the new subsystems fit into the current stack is depicted in figure 1.

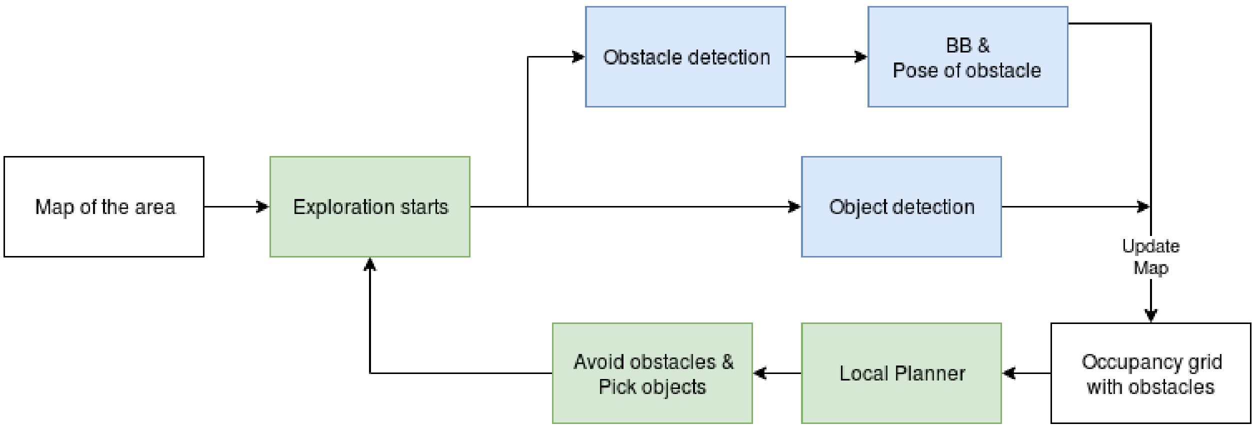

Figure 1. System Logic with Obstacle Detection, Mapping, Planning, and Exploration subsystems

The logic or action sequence that CuBi follows when put in a room is as follows:

1. Map area using LiDAR.
2. Start Exploration

    2.1 While Exploring:

    - Bounding box, obstacles pose ← Obstacle detection
    - Updated map ← Occupancy grid with obstacles
    - Toys location ← Object detection

    Local Planning

    - Avoid obstacles
    - Pick-up toys

3. Termination condition

I designed and finalized the above logical sequences laying the groundwork for integration and implementation of obstacle detection.

For obstacle detection pipeline, two approaches are being evaluated. First is the classical approach of using point cloud data. This is a very straight forward approach in which point cloud inputs were received using a ROS subscriber. The received point clouds were then pre-processed. First the planes were filtered out using RANSAC algorithm. Voxel grid filtering was then performed, and threshold was set to obtain the filtered point cloud data. Voxel filtering reduces the density of the data while still retaining identifiable key points of objects.

Next step was to filter the point clouds according to the size of the obstacles. Was not able to test and get the threshold value. So, a basic point cloud filtering script for obstacle detection was developed.

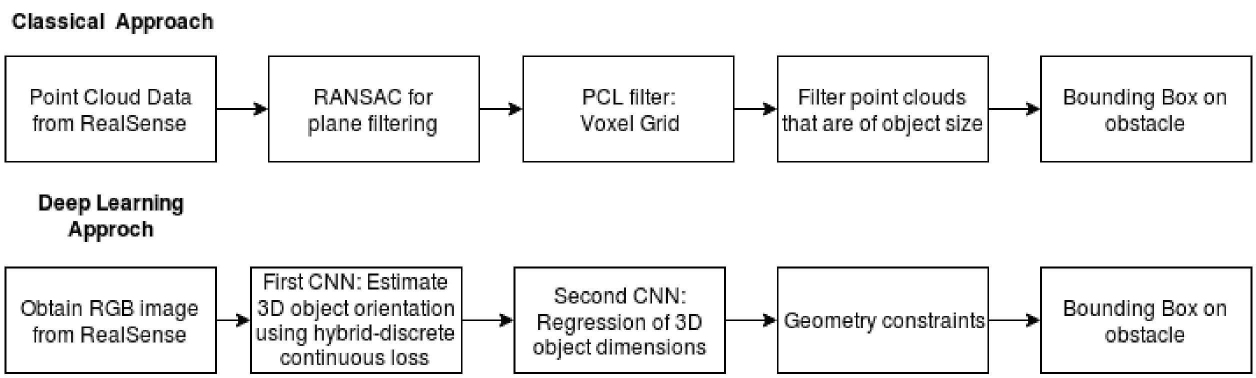

Figure 2. Proposed Obstacle Detection Flow

For a Deep Learning approach on obstacle detection, a paper titled "3D Bounding Box Estimation Using Deep Learning and Geometry" by Mousavian et al, was studied. This paper achieved high results on the KITTI dataset and is also unique in the sense that it estimates a 3D bounding box with only RGB images and uses the notion of objects geometry.

The architecture uses method to estimate an object's 3D pose and dimensions using the constraints provided by projective geometry and estimates of the object's orientation and size regressed using a deep CNN. It also introduced a novel MultiBin regression to estimate an object's orientation. While 2D object detectors such as SSD, YOLO can be much faster, our project needs the depth information of the obstacle. But we also cannot leverage the 3D detectors as it is computationally very expensive. So the approach of estimating 3D pose using just RGB image fits our needs.

The team site was also set up along with teammates Bobby and Jorge at NSH B512. Having an easily accessible test site is very helpful as the problem is not open-ended. Although the testing might become unavailable soon and an alternative should be found in order to continue the progress.

Challenges

For the obstacle detection using point cloud data, the threshold value that needs to be set can be very hard to set. Because this makes use of the camera calibration and the transforms, even a slight offset in the camera can make the detection go wrong. As for the deep learning technique, inference on the Jetson TX2 can be a problem. I have had some experience deploying Deep Learning models on Jetson Nano and it proved to be a hard problem as the inference may not be as high as expected. While TX2 should be a lot better than Nano, still deploying the model can be a bottleneck. One possible solution is quantization of models. Quantization of models can speed up inference on the edge device. These are the reasons why multiple ways of implementing obstacle detection are being attempted.

Teamwork

Jorge, Laavanye, and I own the obstacle detection pipeline. We discussed about design decisions for the obstacle detection workflow. Bobby mapped the testing site using LiDAR with the help of Hector SLAM package. Jorge and I worked along with him for this process. Jorge worked on the cell decomposition algorithm and we discussed certain implementation details such as using RANSAC to obtain the boundaries of the room and how to fit a convex hull to cover the entire span of the room. Paulo worked on improving the gains of the controller, and as I had developed the controller, I had a discussion with him to understand how I could have achieved better results at the beginning.

Plans

The next step for me would be to finalize on an architecture and implement the pipeline. A significant thing that we took for granted is the availability of labelled data. So that needs to be on priority for us. Early integration into the state machine is also needed for creating a robust system. As a team, the plans are as following. Now that localization is done using particle filter (ROS navigation stack) and it seems good, fusion of sensors for localization might not be needed. Object detection must become more robust. Exploration and planning are the other crucial subsystems where most of our focus lies on in the upcoming weeks.