Laavanye Bahl

Team D: CuBi

Teammates:
Jorge Anton, Paulo Camasmie, Changshen Shen, Nithin Subbiah Meganathan

ILR11

November 19, 2019

Individual Progress

The main focus of this review was the integration of object detection pipeline and solving the various problems that occur in the integration along with exploration, localization and picking mechanism.

Reducing False Positives

The new environment created problems of different lighting conditions, walls, ground surface among many more. Hence the object detection pipeline had to be improved to accommodate all these differences.

I tackled this in the last PR by setting up the right thresholds for validating an object from a segmented point cloud and improving the hyper-parameters such as the pass-through filter (cropping) thresholds, voxel filtering size, clustering size, etc.

But after integrating the exploration and picking up modules, the pipeline failed. There were a lot of false positives near the walls and in areas of shadows and these were

I came up with a solution to make a convex hull/boundary polygon of the search region of the point cloud. The walls and edges were usually part of this convex hull. Hence, to completely ignore the walls and the false positives near them, I was shrunk this convex hull by some value. E.g. 80 percent of the previous size. This has been shown in Fig 1a and Fig 1b.

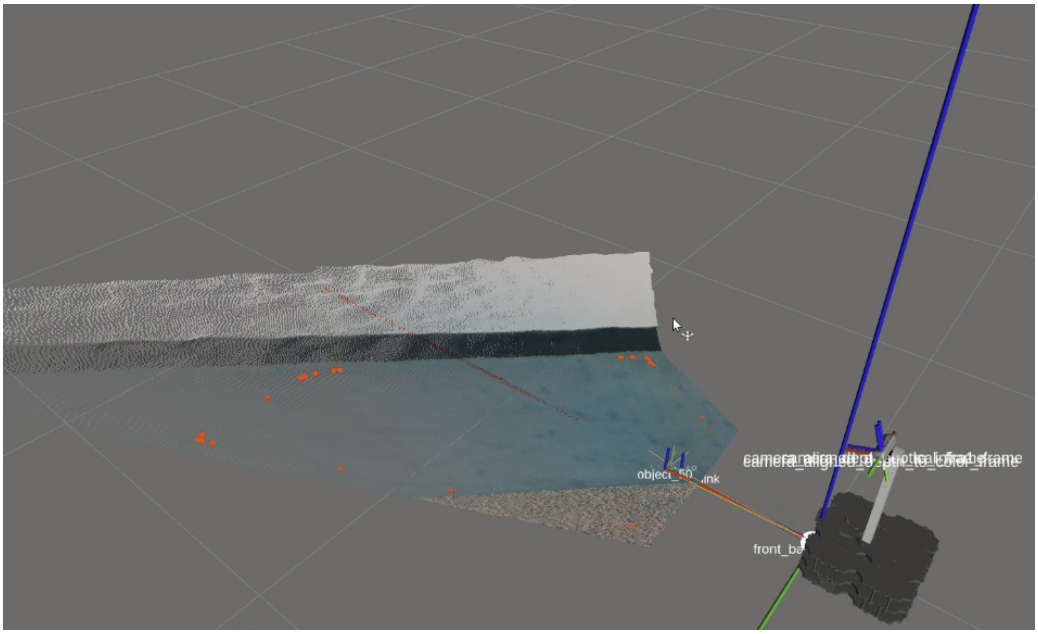

Fig 1a. Red lines show the boundary polygon / convex hull

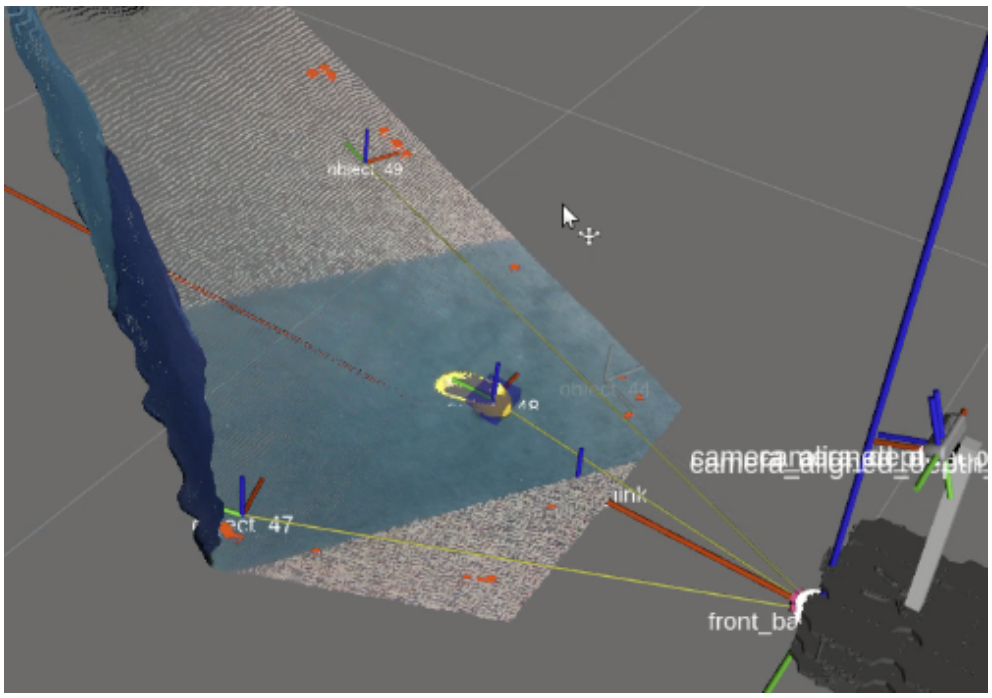

Fig 1b. Red lines show the boundary polygon / convex hull

Hence, with this, I could remove the walls and there were very few false positives for detections.

The results have been shown in Fig. 2a and Fig. 2b. The changes were tested at different positions in the new environment.

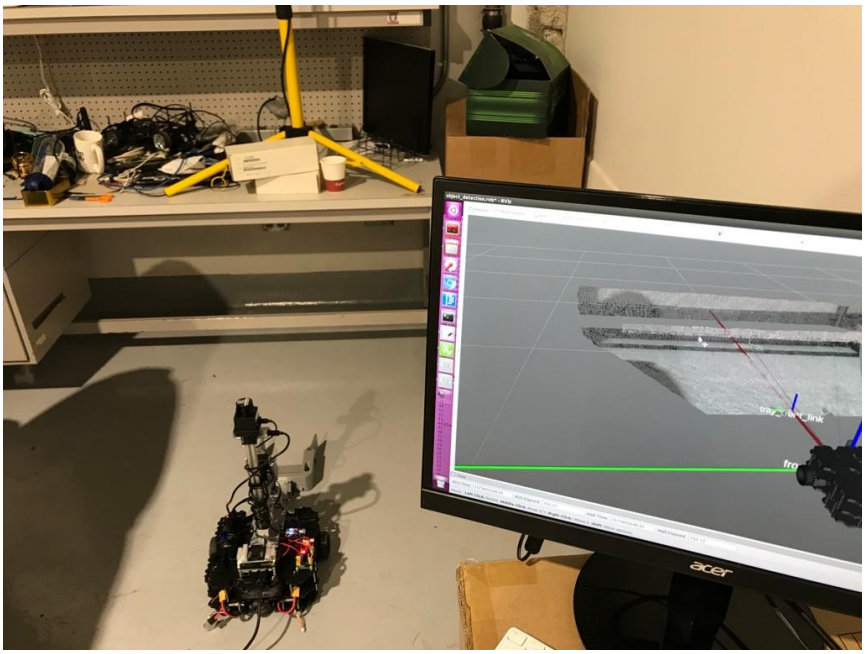

Fig 2a. No false positives near walls

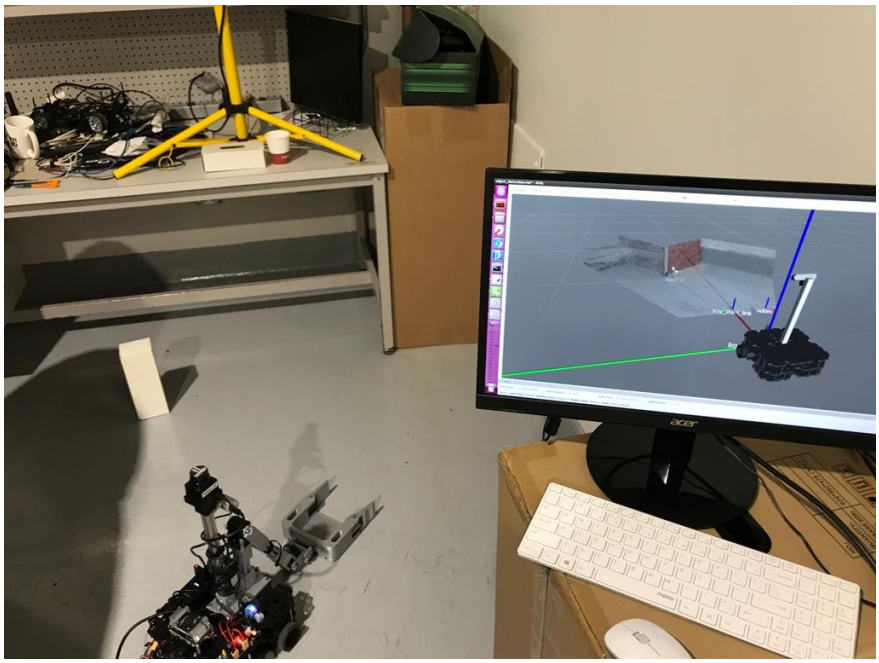

Fig 2b. No false positives near walls

Integrating object detection pipeline

With the newly coded modules for exploration and pickup, the current object detection pipeline had to be integrated. This was not so straight forward and ended up introducing a lot of interdependence (small changes in one module affects the other a lot).

I performed a lot of test cases to validate the integration and come up with the fallbacks. I then worked on those issues and solved them.

Improved locking of objects while picking up

CuBi explores and detects objects, but it needs to lock to an object to approach it and eventually pick it up.

As of now, this locking was done as soon as the detected object appears the CuBi's search area. This was bad when there were false positives. Ideally, the object had to be in the center of the camera image before locking.

As a solution to this, I coded a moving buffer that stores the object id to which we potentially lock. If the same object appears in the moving buffer for more than 4 consecutive frames, then only I give the command to lock to the object and approach it for a pick-up.

Challenges

Individual challenges:

I was getting a lot of false-positive detections in the areas with large lighting variations, mainly because of the shadows formed by the single source of light as opposed to the multiple ones in our previous testing location.

As a solution, I tried to put some extra light sources to compensate for the shadows. This, in turn, proved problematic as it created a lot of reflections on the smooth floor of our new testing location.

Hence, the pipeline was improved to work well in the shadows by tuning the hyper-parameters even more and putting extra checks for dimensions of the detected objects and improved tracking to reject the false positives.

The other challenge was of false positives near walls. The solution is explained in detail in the individual progress section.

Team challenges:

Breaking parts after rigorous testing

> There have been a lot of issues regarding the hardware we are using for CuBi as we try to tackle more challenging test cases. The top joint of CuBi broke after crashing with a wall. This is shown in Fig4. This hampered out testing at the last moment, which wasted a lot of time for the team. As a solution, we immediately printed more spare parts.

Unknown issues with the PCB

> The PCB got overpowered recently and some of the electrical components broke as a result. I think this was because of haphazardly plugging the charging adapter right after switching from the battery-powered mode. A lot of time was spent to get PCB back in working condition. We decided to be more careful with operating the switches and power adapters and set some guidelines that we need to follow whenever we switch CuBi on.

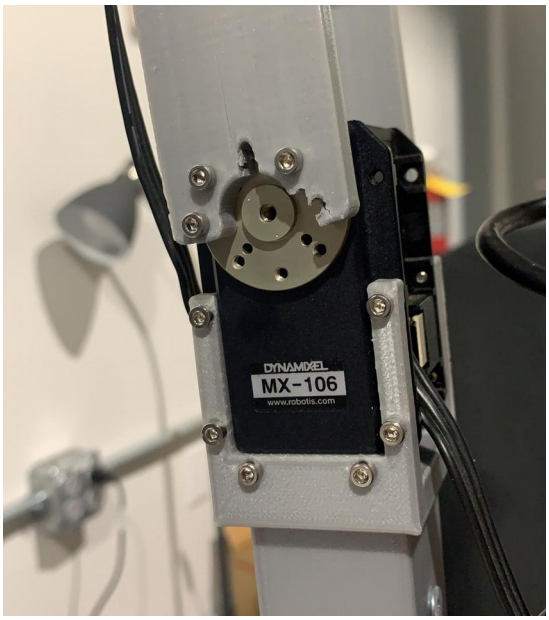

Fig 3. Broken Joint (Image Credits: Bobby)

Integration and updating state machine:

When there are a lot of changes made by different members of the team, it requires a lot of time and effort to add new functionality to the state machine.

An abstract state machine was developed by Jorge to make it easier to modify the state machine easily to run different individual experiments without hurting someone else's code.

Project management

Due to the end of the semester and different course schedules, it is difficult to find the same time to work on problems together and integration. Better project management and a shared free-time schedule were adopted to tackle this problem.

Time with the robot

Multiple people require access to the robot at the same time to do individual tasks.
It requires controlling the robot.
Bag files were efficiently used to do some work on personal laptops.

Teamwork

Following describes the work done by the team members and how I interacted with them:

Paulo:

Paulo worked on the way-point generation and improved robot turning by adding a cost function to every turn made. I interacted with him to order new toys for CuBi to pick up as well as arranging floodlights for the new environment.

Bobby:

Bobby worked with Jorge on exploration and localization. I interacted with him a lot for the integration of the object detection pipeline with the rest of the modules. There were a lot of problems since we were modifying the same files and changes made by either of us drastically affected the performance of the modules we were working on. So, we had to sit together and finalize the code.

Jorge:

Jorge worked along with Bobby for exploration and localization. I worked with him to integrate my code into the state machine and add a new locking mechanism for approaching objects. I performed a lot of testing in different scenarios to come up with the fallbacks and come up with a plan to solve them for this PR as well for FVD.

Nithin:

Nithin worked on the dropping mechanism for the robot. He also worked on solving some offset issues while approaching the objects. I interacted with him regarding if those issues could be solved with the vision pipeline.

Future Work

Individual plans:

We have the FVD after this PR. Hence, future work involves a lot of integration, testing, and improvements. This is where most of the time is going to be spent instead of adding new

functionality.

It would mostly be teamwork but there is still individual work of making object detection more robust.
I have to integrate the pickup validation along with the already integrated modules and perform many tests in different scenarios. This will give us a huge speedup in the whole demo as CuBi would not waste time going back to the dropping location when there is a pickup failure.

Also, the object detection is deeply coupled with other modules like exploration and pick up mechanism. So, it keeps breaking when changes are made in those modules. So, I have to continuously keep track of the new changes and perform some unit test cases regularly.

Team plans:

There is a lot of future work as a whole team as we approach the FVD. We will sit together to integrate everything to attain end to end automation. We will work on improving individual subsystem modules independently.

The goal is to Increase collaboration and interactions. We plan to have daily standups and allocate tasks daily.