Jorge Anton Garcia

Team D – CuBi

Teammates: Laavanye Bahl, Paulo Camasmie, Changsheng Shen (Bobby), Nithin Subbiah Meganathan

ILR10

Nov. 6, 2019

Individual progress

For this progress review I worked on the global planner, obstacle avoidance given a map, documentation, and further abstracting our code.

CuBi Technical

For exploration, I created the pipeline that I included in figure 1. My goal was to create a pipeline which was as flexible as possible. I wanted it to be robust to different room shapes and sizes. The major assumption I made is that a room has four main sides that define it.

For the last progress review, the exploration module I had written read in the map and the meta data from the file and outputted the lines representing the edge of the room. This week, I built everything required to output the next waypoint for exploration. After finding the lines, I put them in projective space and used the cross product to calculate their intersection. The four points I found were used to define a convex hull which would represent what is inside the room. I labeled all the points in the occupancy that were outside of the convex hull as obstacles. Thanks to this, even if our room is not walled completely, it will not escape. This simulates testing in a room in a house where there are door openings. Cubi will stay in whatever room it is programmed to clean.

Once the room was defined, we now needed to decompose the cells. I also restricted Cubi's range of movement between cells to be forward, backward, right, and left. In this way, Cubi always enters the next cell perpendicular to the cell boundary and we can assure that it can scan the entire cell when going to its centroid. I decided the cell size should be 0.7m x 0.7m as Cubi can reliably see toys within a radius of 0.7m. This allows Cubi to scan one cell and parts of the one's in front and to the sides. This extra redundancy was created to ensure we pass through most sections twice.

To create the cells, I decided to use a conservative approach. Given the resolution of the cells in the occupancy grid (5cm) I grouped the cells, so they were the decomposed size we wanted: 0.7m x 0.7m. When decomposing, if any of the smaller 5cm cells were obstacles, we would label the entire cell as an obstacle. This conservative approach acted like a way to inflate obstacles.

Once I created, the decomposed cell grid, I used the centers of all the cells as the waypoint. To do so, I was able to map from the decomposed cell coordinates, to the image's original coordinates and then map these into a pose in the map frame. It worked really well!

Once we had the waypoints we needed to traverse, I had to create an exploration policy. As of now, I created a very simple one that right followed the walls only visiting unvisited cells. Remember Cubi can only move in four directions. If it was surrounded by obstacles or visited cells, it would go to the closest unvisited cell. The ending condition, was having visited all the waypoints. The exploration policy also takes into account the direction Cubi is facing, so that not only is Cubi sent to the right centroid, but is also pointing correctly towards the next place it needs to go.

I have also added functionality to show a simulation of how Cubi is exploring and ways to label obstacles on both the original and decomposed map. Overall, this process allows Cubi to avoid obstacles found on the map and ensure that it has covered the entire room. I also worked with Bobby to integrate it with the localization packages he installed.

The entire process can be seen in the figures 1-5 in the Appendix.

CuBi Project Management

Every week, I keep our task manager up to date. I am realizing we are having issues as everyone's tasks are not granular enough, so I am trying to help the team do so. We have improved in terms of accountability of tasks. These past weeks we have been meeting internal and external timelines. We are still a little behind, but we are close to finishing our FVD and should do so by the next PR.

Challenges

Technical

One of the biggest issues I faced this week was crashing into walls. The length from one end of Cubi to the other is 0.8m. Its turning radius is 0.55m which is very larger. Despite making the cells 0.7m x 0.7m and allowing for buffer Cubi would still crash into the walls. My method of inflating obstacles does not ensure that we will never crash into walls. Since Cubi needs to reach the center of each cell, the border to the next cell is 0.35m away, but its tray is 0.55m long. In the worst case scenario were the obstacle was actually at the border of the cell, we'd crash by 20cm. In most cases, the obstacle is not at the border, so we may just slightly touch the wall. If we can't reach the desired waypoint, we just move to the next one. As of now, these minor crashes have shown to be fine for Cubi, as it is moving very slowly and slows down when it is close to it's goal. The small crashes also don't affect localization. If I made the cells 1.1m in length to assure Cubi would never crash, then we'd not be able to assure coverage.

Another issue I was experiencing was that I needed to help in all integration tasks. This caused many people to get unnecessarily blocked or unable to do the final integration. To avoid this, I wrote a very extensive README explaining how to test individual modules of Cubi. I also wrote a couple more functions to make running one state of code individually easier. This has facilitated the rest of the teams work greatly.

Team Work

Nithin: I worked with Nithin on deciding how we would try to add the obstacles he would find with his module on the map. I wrote the interface to be called from our state machine.

Paulo: I wrote a global planner which performed the exploration and had an empty function in which given a map and two points, found a trajectory between the two points. He wrote that portion of the global planner algorithm.

Laavanye: I worked with him to integrate grasp validation into our code.

Bobby: Worked with him to integrate the global planner algorithm with the localization planner he added. We worked on the final integration and testing on the robot together.

Plans

There are 2 weeks left for us to finish the first version of the FVD. I will help out anyone who has issues integrating their part. I will also perform the final integration of the global planner module. It still only generates waypoints at the beginning and sends them all at once. The state machine should tell the global planner its state (going to a point or exploring) and the global planner should return a waypoint given the position. I will also work on making the trajectory smoother and potentially avoid the stopping in each cell centroid.

Appendix

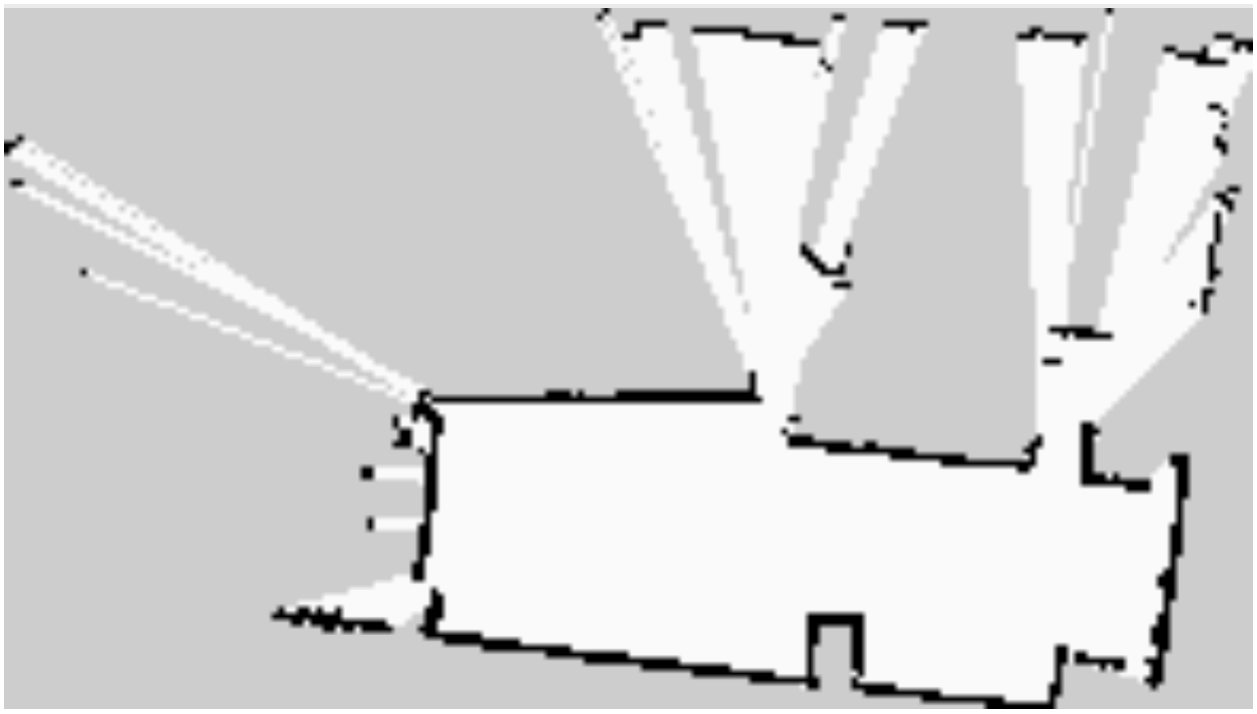

Figure 1: Original Room

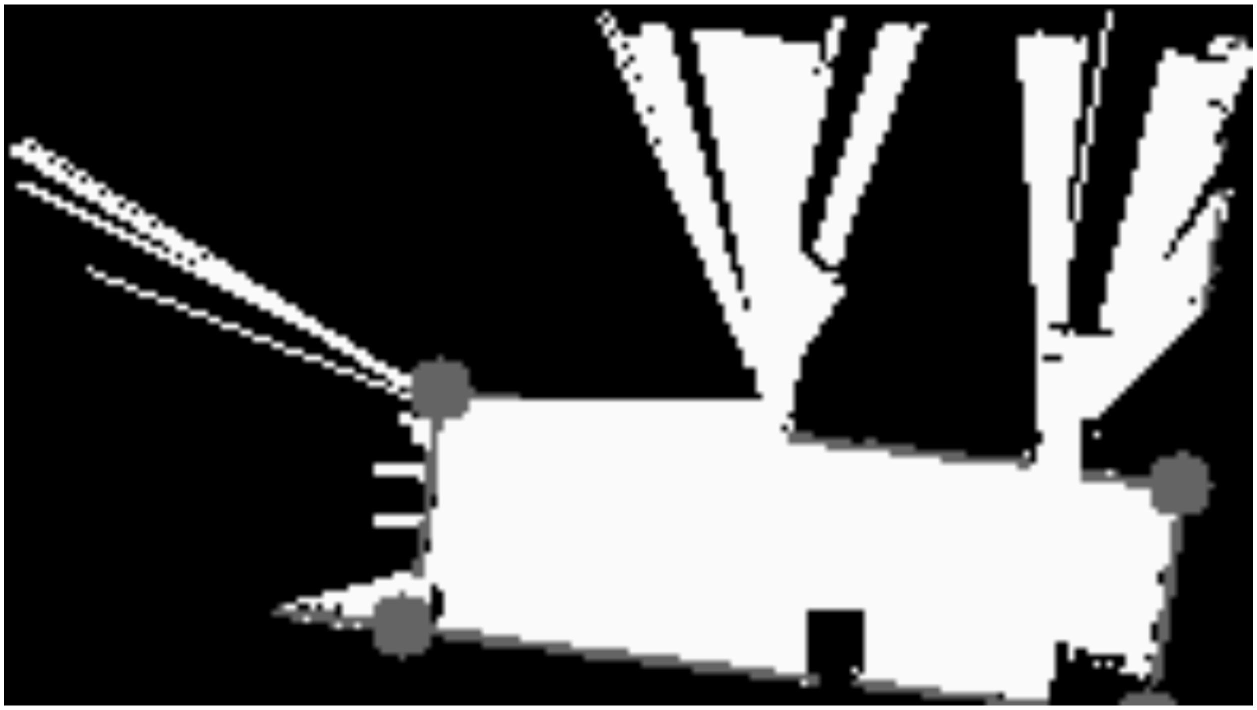
Figure 2: Detecting boundaries of room and creating convex hull

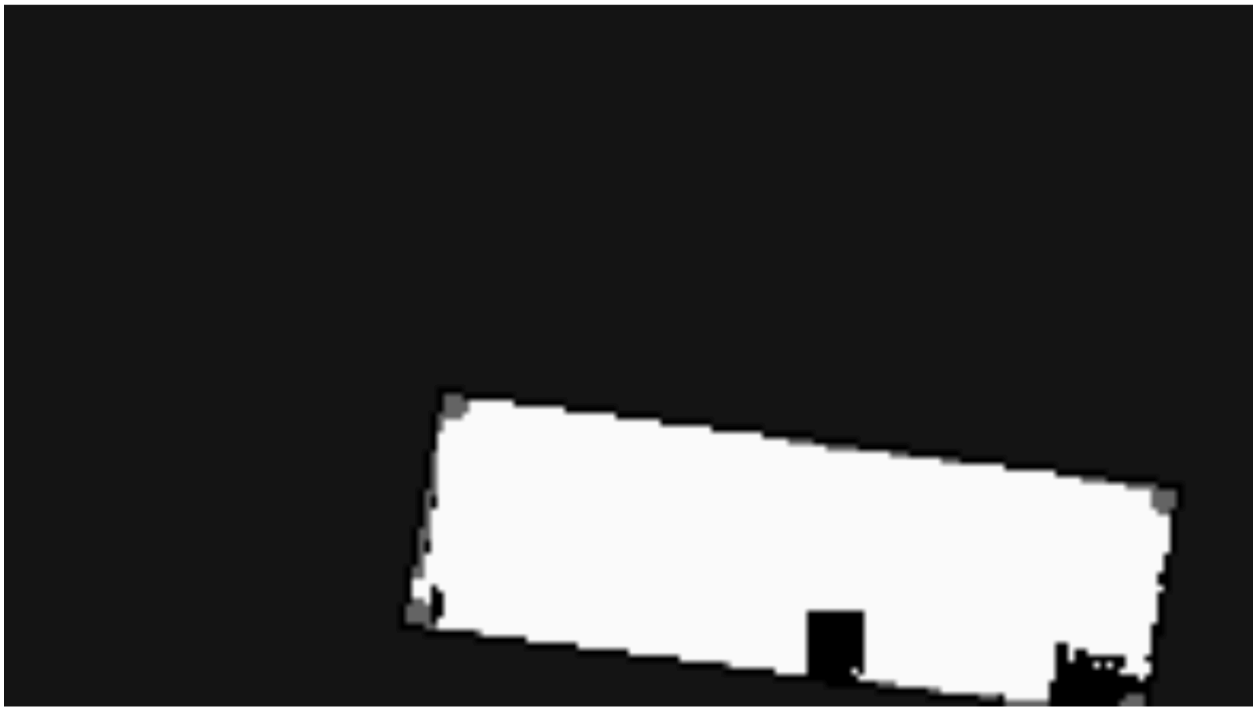
Figure 3: Labeling anything outside of the boundary as an obstacles

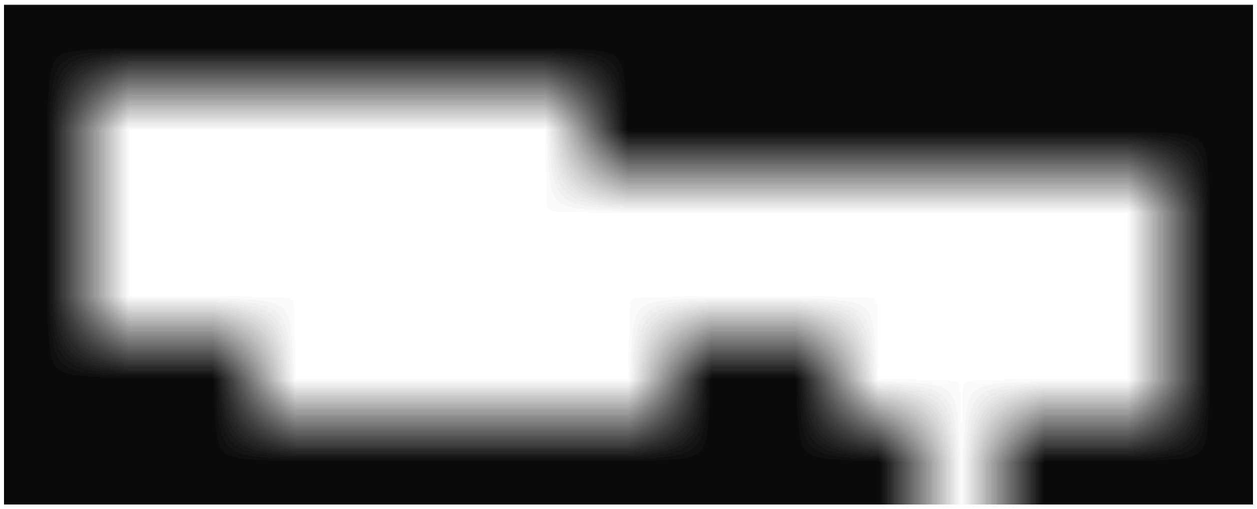
Figure 4: Decomposed cells into a 0.6m x 0.6m

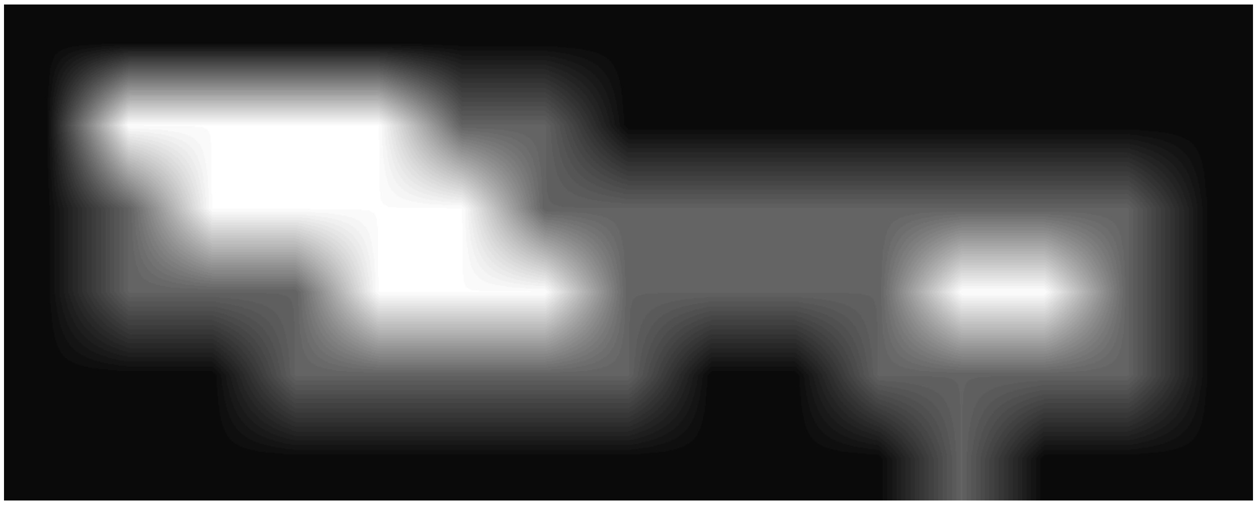
Figure 5: Decomposed cells with cells already visited in grey (can see the right following)