MTE 544 Lab 4 Report

Stephen Wang	Ajit Rakhra	Howard Dong
x2274wan	a3rakhra	h53dong
20821034	20832703	20847639

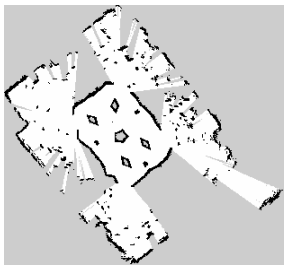
Station 3, Robot 14 on Fri Dec 1 at 8:30 AM

Path Planning

A* search algorithm is used to find a path on the occupancy grid, which can then be turned into a trajectory in the Cartesian birds-eye plane. The algorithm can be summarized in the following list. Each cell in the occupancy grid is represented as a custom `Node` datatype. A `Node` contains its parent and its location; `Node`s can subtract each other to calculate the heuristic distance; `Node`s can compare with each other to return `True` when their locations match.

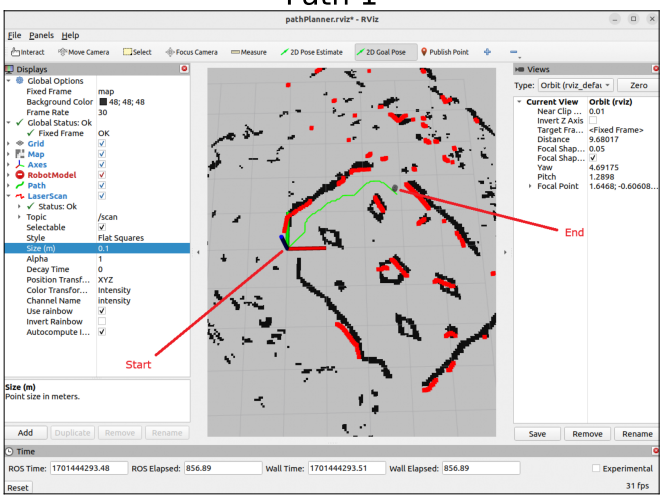
1. Initialize the list of unvisited pixels with a `Node` representing the start
2. For each unvisited pixel:
    A. Remove that pixel and add it to the list of visited pixels
    B. Create `Node`s for the 8 surrounding pixels
    C. For each neighbouring pixel:
        a. Skip if it's already been visited, otherwise
        b. Update the "real" cost to be 1 + its parent
        c. Update the heuristic cost either with Euclidean distance or Manhattan distance
        d. Update its total cost as the sum of the above
        e. If it's in the unvisited list and the heuristic cost is lower
            i. Add it to the list of unvisited nodes
3. Repeat until there is no more node in the unvisited list

Following this algorithm, every pixel will be visited but in contrast to Dijkstra's algorithm, the pattern of descending cost between the pixels/`Node`s will also be influenced by the heuristic cost.

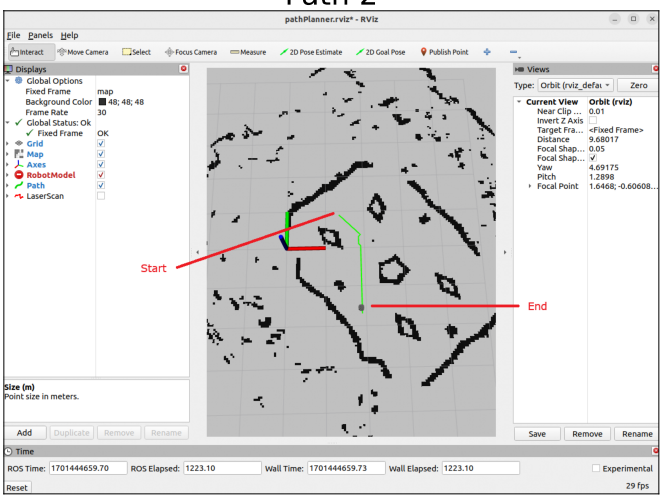


The following graphs show the robot's planed trajectory. Path 1 uses EKF while Path 2 uses the ROS sensors. These are two instances demonstraiting the A* algorithm.

Path 1



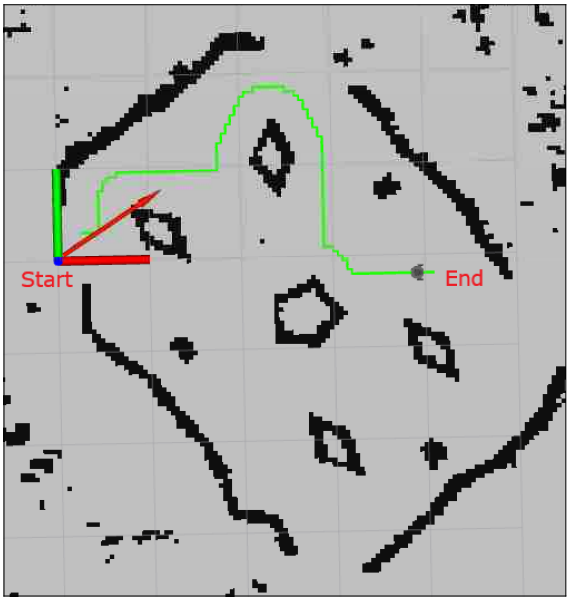
Path 2



The following plots show the paths found using the Manhatten and Euclidean distance heuristics. In the plots, it can be seen that the Manhatten distnace heuristic finds a path priotirizing movement parallel to the x and y axis, while the Euclidian distance heuristic finds a path priotiizing movement towards the direction of the goal point.

Furthermore, we noticed that during the map acquisition, the sensors picked up a small obstacle in the middle of the field which direclty impacts the path finding algorithm. The robot decides to traverse through the edge of the field instead of cutting thorugh the middle towards the goal point.

Path with Manhatten Heuristic



Path with Euclidian Heuristic