Chalmers University of Technology

TME290 Autonomous robots

Home problem 6: Implement path planning and following

Thanisorn Sriudomporn
thasri@student.chalmers.se

May 22, 2020

1 Overall

In this assignment, the main objective is to plan a path from start point, goal point, and the map provided in .txt file. The procedure how to create map and planing path will be describe in this report. Since the programming part have done in Matlab, there are the conversion from .txt file to map explained also. The path planing is done using Dijkstra's algorithm and will be well explained in Path planing section. In the end, there will be the result of this algorithm and the procedure to run the code giving only .txt file, start position and end position.

2 Map from text file

In the first step, one read the .txt file and store it in table variables. Then, one convert it into array and save it as $x0$, $y0$, $x1$ and $x2$ which are start and end points of walls. While reading text file, the positions are in form of x0,y0,x1,y1; as in figure 1. Matlab need some conversion from this form of data to integer array. Next, one calculate and create the map array from the maximum x and y divided by the grid size. After that, one go through all walls read in .txt file and fill it into map array. The created map is shown in figure 2.

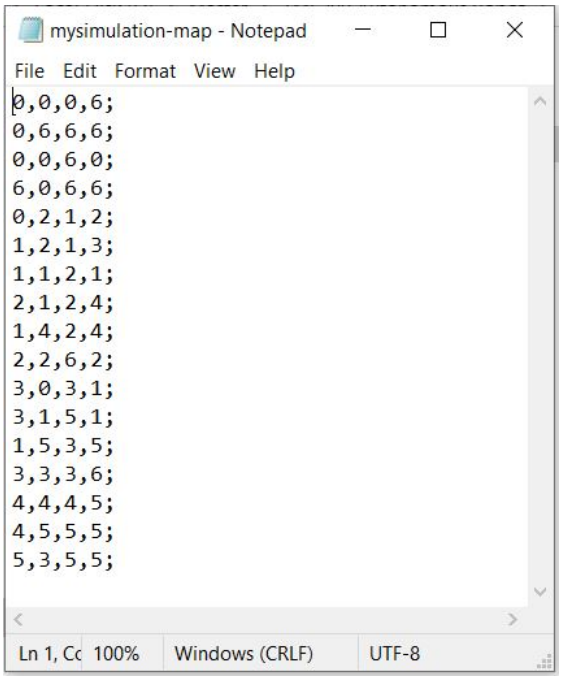

Figure 1: The example of .txt file using to create map.

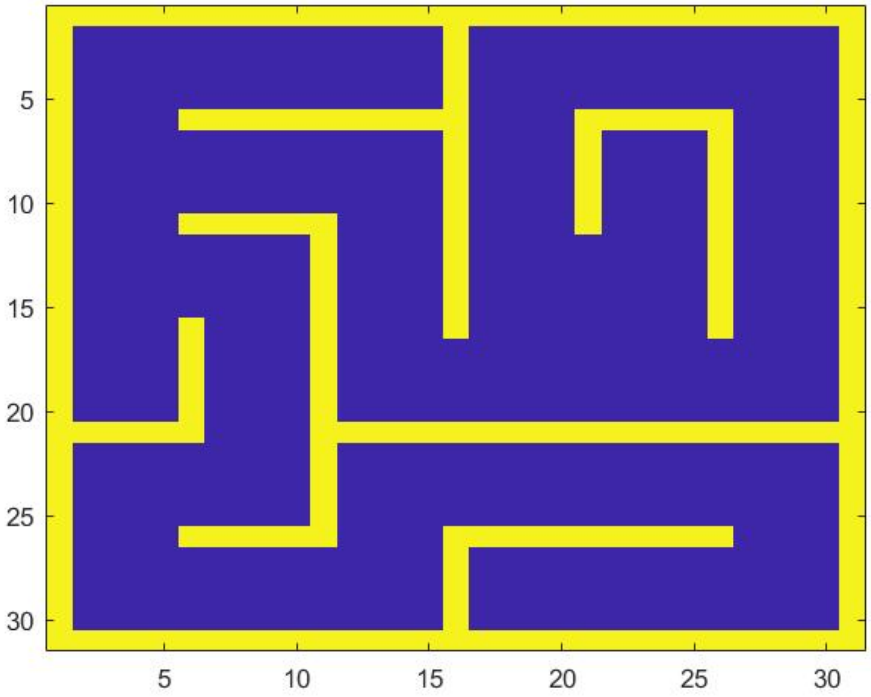

Figure 2: The map created from .txt file.

3 Path planing

Now, one will start the path finding algorithm. In this case, one use the Dijkstra's algorithm to find the path. One start with create the array in the same size as map to store node's name as 1 2 3 and so on. Calling a node by it's name is easier than calling it by index i,j in order to understand the procedure while coding. Then, one copy the node name array to unexplored nodes array and create the node array, which contains infinity value, to store distance to this node.

Next, one convert start position and goal position into grid index, i.e. array index of nodes. The start node and goal node are checked that in the wall or not. If they are not in the wall node, one is good to go.

In this step, the algorithm start. One set the distance for start node to 0 and start while loop. This while loop will keep running until there is no node left in unexplored array or the current node is at the goal node. One first select the current node by finding the lowest distance node and still in the unexplored array. Now, one have current node and the current node's distance. One check that current node is the goal node or not.

Next, one get the neighbor node of the current node. There are 4 neighbors for each node. One select only if the neighbor is not in the wall and not outside the map.

Now, one go through each neighbor in the previous step and calculate the new distance. The new distance is calculated from current node's distance plus distance to that neighbors which is 1 node. Then, one check that this new distance is shorter than current distance of neighbor. If it is shorter one set the patent of neighbor node to current node and the new distance to neighbor node's distance. One keep doing this while loop until reach the destination node.

In the last step, one now have the parental array which contain the node before reach that node. One start from the goal node and step back until reach the start node to present the path.

4 Result

In this section, one marked walls as in yellow. The explored nodes and unexplored nodes are filled with purple and light blue respectively. The green-blue and orange states start position and destination position. The path is in green.

One test the algorithm with different start and goal position on custom map one made on one own. The results are in figure 3, figure 4 and figure 5.

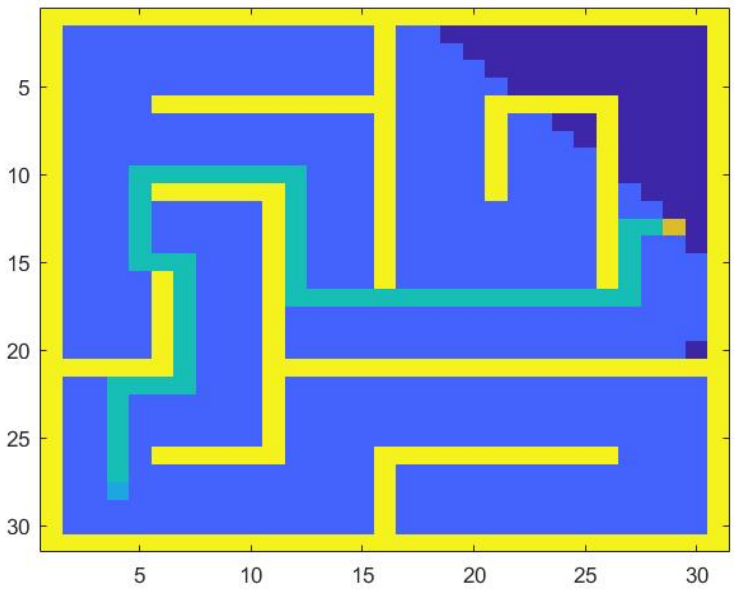

Figure 3: Result of the algorithm.

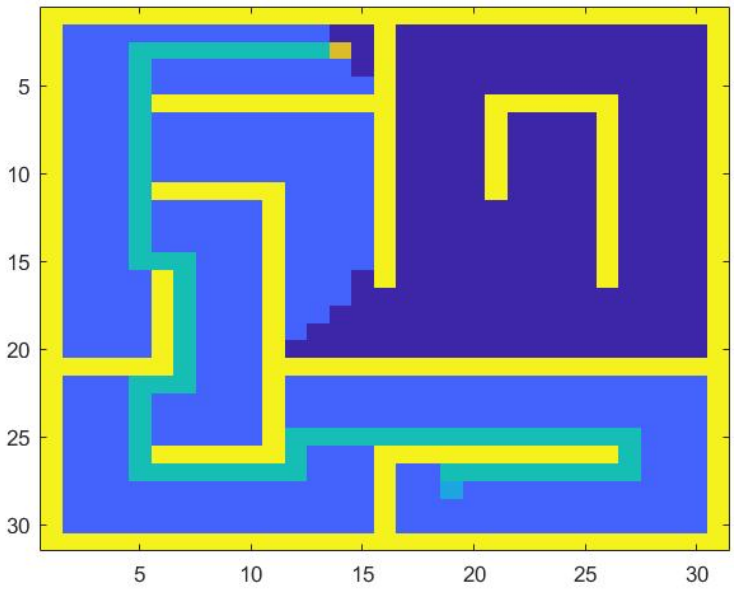

Figure 4: Result of the algorithm.

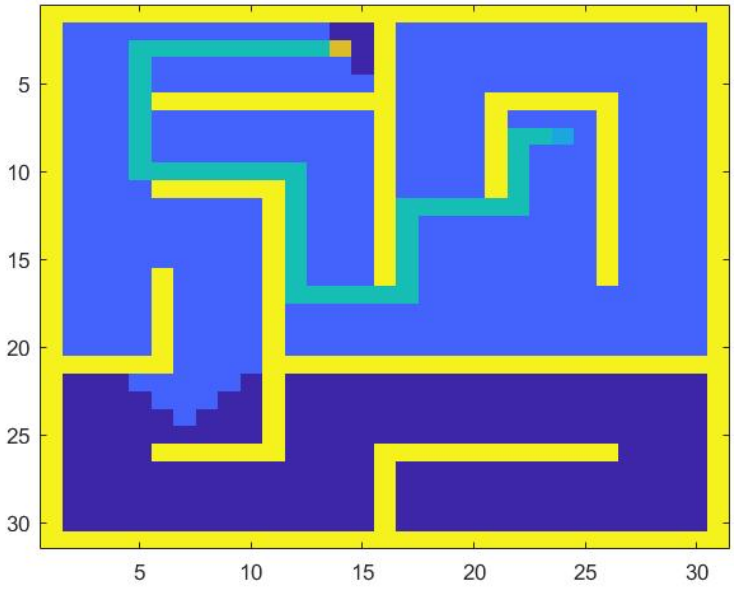

Figure 5: Result of the algorithm.

5 Procedure to run code step by step

There are only few steps needed in order to run the code as following step along with figure 6.

1. Put the simulation map .txt file in the same folder with the script file as in balloon 1.
2. Change the start position and destination as one wish pointed in balloon 2. Remember to keep it in the map size and not in the wall position.
3. Change the simulation map variable to .txt name match with the file in step 1.
4. Press run and wait for the result in balloon 4.

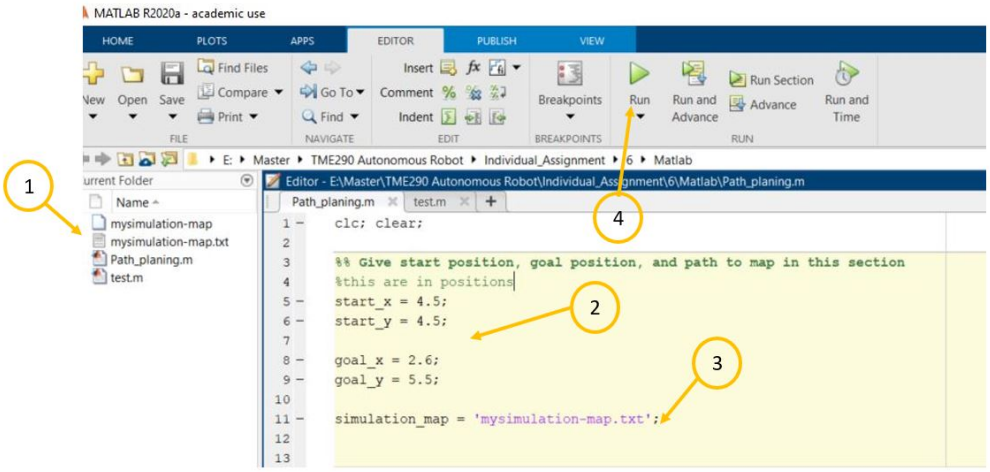

Figure 6: The procedure to run the code step by step.