

Route Planner

PathPlanner class is defined to get a map, a start node and destination node and method is defined to compute the shortest path between those two while searching in the right direction. An open set and closed set are being kept. The closed set is to keep nodes that have already been evaluated and the open set is to keep nodes that have been discovered but not evaluated yet. The open set is also known to be the frontier of the search.

gScore and fScore are values associated to each node to keep track of the cost of a path. gScore is the cost of the already found path to a given node and it represents the real distance from the start to reach that node. $fScore = gScore + h$, where h is the heuristic or the estimated distance from the node to the goal, which is computed as a straight line between the two. So, the fScore of a node is the total already traveled distance to get there, plus the estimated distance to the goal.

The algorithm adds the start node in the openSet and explores from there. Includes the start node's neighbours in the openSet and calculates the gScore for each of them. Since we have latitude and longitude coordinates, the distance between two connected nodes is the straight line between them. The next node that is explored is the one that has the lowest fScore and this is to be done until the goal is found.

After having found the goal, to be able to reconstruct the path, the cameFrom information is used for each node. cameFrom is a dictionary constructed along with the search and it gives each node the most efficient previous node to be reached from.

Time Complexity:

It depends on the heuristic $h(x)$ value. $O(|E| + |V|)$ where E number of edges, V number of vertices. so we can say it is $O(n)$

Space Complexity:

$O(1)$ because we don't need extra space