[hptb]

Selection and implementation of technical approaches for path planning using ROS2 Humble

Miguel Obrebski

1 Task description

1. Skim through the papers and other online resources that you collected and select two promising approaches for your topic.
2. Explain your reasons for selecting these approaches.

   For this you have to imagine the scenario in which the robot is moving through the university campus. Think about how the data would be in this scenario (e.g. images of people, speech, language, lighting, background noise, motion blur, etc.). Would this affect your choice of the approaches?
3. Identify the hardware and software requirements for the selected approaches.
4. Add the answers for all the above items in a document and upload in PDF format.
5. Start implementing the selected approaches.
6. Set up a GitHub/GitLab repository for your code and mention the link in the PDF document in Item 4.
7. During the class on 17.11., present the progress in technical implementation:
   (a) Programming language, libraries, dependencies
   (b) Technical issues you are facing
   (c) Development plan until 15.12.2023
8. If you cannot attend the class on 17.11., include your answer for Item 7 in the PDF document.

Note: The sections are not ordered to fit the tasks' order.

2 Selecting possible approaches

2.1 Considerations

2.1.1 General

- Number of nodes?
- Number of dimensions? 2 or 3?
  2 to make it easier.
  Different floors? put them next to each other and connect the elevators

2.1.2 General Ideas

- Dynamic Programming or Cache?

2.1.3 Algorithm-specific

- Runtime
- Memory/Storage usage

Algorithm	Runtime			Memory			Notes	Acceptable Solution?
	min	med	max	min	med	max		
Dijkstra	→	→	$O(nlog(n) + m)$	→	→	$O(n + m)$	with fibonacci heap	optimal
A*			$O(n^2)$ or $O(b^d)$	$O(n)$		$O(b^d)$	heavily based on heuristics; memory should be the bottleneck (?)	optimal
IDA*			$O(b^d)$	$O(n)$		$O(d)$	takes years (?)	optimal, but too long
Best-First-Search			$O(b^d)$	$O(n)$				not optimal
Floyd-Warshall	→	→	$O(n^3)$	→	→	$O(n^2)$	not efficient for big graphs; this is from all to all	optimal, very interesting
RRT	→	→	$O(K \log K)$	→	→	$O(K)$	near optimal	near optimal, very interesting
RRT* (or similar)	→	→	$O(K \log K)$	→	→	$O(K)$		near optimal, very interesting
Jump Point Search			$O(nlogn)$	→	→	$O(n)$		optimal for grid-maps
Trace			$O(V + E)$	$O(n)$		$O(V + E)$	not optimal, weird movement	no

The blue are selected

3 Reasons for selection of approaches

3.1 Floyd-Warshall algorithm

Seems like a good solution to preplan all paths instead of calculating manually.
Should be resource-efficient.
FWA was always interesting for me.

3.2 RRT* algorithm(s)

Seems to be running edge in motion planning.
Interesting approach.
It's something else.

3.3 Scenarios

Sorry, for now I just wanna come from A to B.

4 Identify Hardware and Software Requirements

For now see above

5 Start implementing

6 Setup GitHub/GitLab repo

`https://github.com/pr1mee/PSR_Robile`

7 Present progress in technical solution

7.1 Programming language, libraries, dependencies

7.2 Technical issues you are facing

1. Extreme problems getting ROS and Gazebo to work, maybe consider making a clean aio tutorial?

7.3 Development plan until 15.12.2023

8 Conclusion