

Improving Freespace Planning (Parking)

#4936

 Closed



Answered by mkquda

mkquda asked this question in Ideas



mkquda on Jul 3 Collaborator

Background

The current implementation of freespace planner / parking is limited in what it can do, and recently some issues were reported when trying to utilize the parking functionality in field tests.

The aim here is to share experiences using free space planner, clarify the existing issues and limitations, and discuss ideas to improve quality and performance of free space planner.

Problem Statement

The problem can be split into two parts:

- Improving the path planning algorithm
- Improving the overall design of the parking feature

Path Planning Algorithm

Currently two freespace planning algorithms are available (*astar* and *rrtstar*), *astar* is primarily being used while *rrtstar* is not as mature and rarely used.

The path planning algorithm should be able to:

- Generate a feasible path satisfying the vehicle constraints
- Generate a smooth path so that the controller can track the path easily
- Generate a path that does not unnecessarily go back and forth repeatedly
- Generate a path within a reasonable period of time
- Generate a path with minimal deviation from desired goal position and orientation

Parking Feature Design

Currently the design specifications / requirements are not clearly stated, so it is difficult to define the situation which the system can / cannot handle. It is important to discuss and define the requirements and limitation, as well as share encountered issues to help improve the feature.

Current Work

Currently we (TIER IV) are working on some improvements to the path planning algorithm to address the points listed above. The improvements include:

Category



Ideas

Labels

component:planning

1 participant



Refactoring & Improving Hybrid A-star implementation:

- Refactor code for better readability and maintainability
- Use actual vehicle info (max_steering + base_length) to compute node transitions
- Fix node exploration logic to update opened states if explored again with lower cost
- Add cost for turning and cost for change in steering angle
- Add minimum distance threshold between direction changes to avoid frequent switching

Implement Obstacle EDT Map

- Implement edt map computation to store distance to nearest obstacle for each grid cell
- Use obstacle edt to limit collision check necessity
- Use obstacle edt to apply cost on proximity to obstacles

Implement Backward Search Option

- Modify algorithm to support backward search from goal to start
- Make it configurable
- Add option to constrain the direction of the path near goal for better arrival

Considerations

Due to limited resources, the current work being done by Tier IV is limited to improving the planning algorithm.

Making design / requirements modifications to the overall parking feature is out of the scope of the current work.

↑ 2



Answered by **mkquda** on Sep 1

Corresponding PR's:

- [autowarefoundation/autoware.universe#8068](https://github.com/autowarefoundation/autoware.universe/pull/8068)
- [autowarefoundation/autoware.universe#8089](https://github.com/autowarefoundation/autoware.universe/pull/8089)

[View full answer](#) ↓

1 comment

Oldest

Newest

Top



mkquda on Sep 1

Collaborator

Author

Corresponding PR's:

- [!\[\]\(7e19807c61da14f515588e95cd49886c_img.jpg\) refactor\(freespace_planning_algorithm\): refactor and improve astar search](#) [autoware.universe#8068](#)
- [!\[\]\(8ff9e60a4b0560d7ec99179ef4779d9e_img.jpg\) feat\(freespace_planning_algorithms\): use distance to nearest obstacle to improve path planning](#) [autoware.universe#8089](#)
- [!\[\]\(ab9b69bf5753a01c76b30af859454360_img.jpg\) feat\(freespace_planning_algorithms\): implement option for backward search from goal to start](#) [autoware.universe#8091](#)
- [!\[\]\(c5af66b13c724ca428497900cdbbc9b3_img.jpg\) feat\(freespace_planning_algorithms\): implement support for multiple goal candidates in A star planner](#) [autoware.universe#8092](#)
- [!\[\]\(1fde827780c8f912fd3ae9174d52d155_img.jpg\) feat\(scenario_selector, freespace_planner\): improve freespace planner edge case behavior](#) [autoware.universe#8348](#)
- [!\[\]\(49ab9fdb6ddb6816bcb8ccc012d5cebd_img.jpg\) feat\(freespace_planning_algorithms\): add new parameters for astar planning algorithm](#) [autoware_launch#1120](#)
- [!\[\]\(a10cf212d457430b842f8ac59c63db70_img.jpg\) perf\(freespace_planning_algorithms\): tune freespace planner parameters](#) [autoware_launch#1097](#)
- [!\[\]\(e8a826213cf8b53a8c13f5432344afc9_img.jpg\) feat\(freespace_planning_algorithm\): update freespace planner params](#) [autoware_launch#1080](#)

✓ Marked as answer

↑ 2

0 replies

Answer selected by **mkquda**