Comparison of Grid-Based Search Path Planning - Dijkstra and A* Algorithm

Kelun Liu
Pei-Lun Hsu
Jan 9th 2020

Research Problem

Which algorithm (**Dijkstra and A***)

has better performance (including **distance of predicted path** and **execution time**)

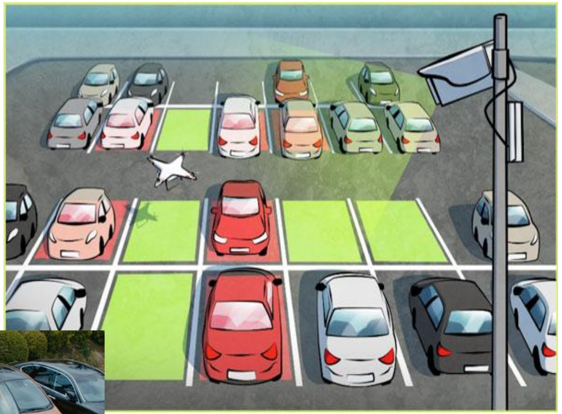
for **path planning of autonomous vehicles**?

Why is this Problem Important?

Efficient path planning is one of the main prerequisites for fully autonomous vehicles

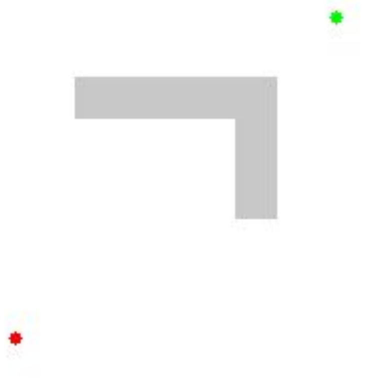
A vehicle navigates from a start pose to a goal pose and avoids all obstacles in an environment that does not offer any specific structure like lanes

Especially driving in complex environments containing obstacles and street barriers



Related Knowledge - Dijkstra and A* Algorithm

Dijkstra Algorithm



Start node: lower left

Goal node: upper right

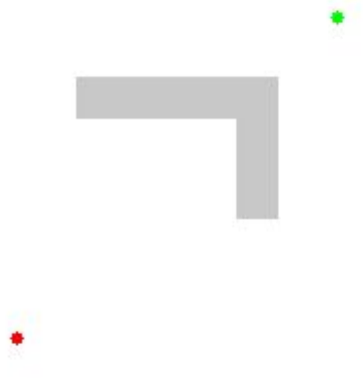
The greener, the closer

Open nodes: a set of unvisited nodes

Filled nodes: a set of visited nodes

When arrive a new node, it will explored all the different directions, like a wavefront (heuristic=0)

A* Algorithm



Modified from Dijkstra

But the greener, the farther

One can first see the A* moving in a straight line in the direction of the goal, then when hitting the obstacle, it explores alternative routes through the nodes from the open set.

Related Knowledge - Simulation

A gridding simulates the real world

Start point: Vehicle

End point: Parking Spot

Circle and rectangle: Barries

Underactuated problem

Motivation constraints of vehicles and robots

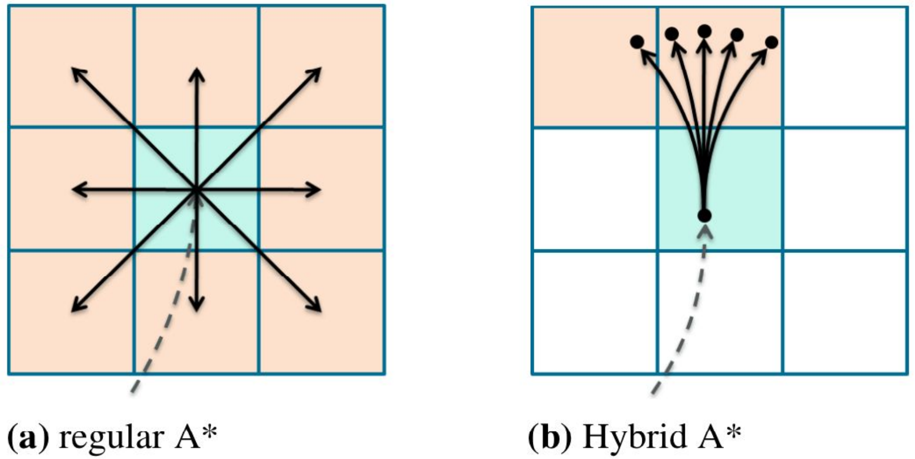
A point can get any position and orientation in math and software simulation plane

But in reality cannot move sideways

When a vehicle would like to turn right or left, there will be a turning radius θ (less than degrees of freedom)

Research Methods: Hybrid A* and Hybrid Dijkstra

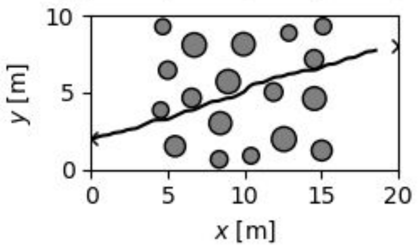
We adopt the hybrid version of A* and Dijkstra algorithms.
Hybrid algorithms consider the nonholonomic constraint of the real-world vehicle, and is widely used in the state-of-the-art researches and applications related to auto-driving.

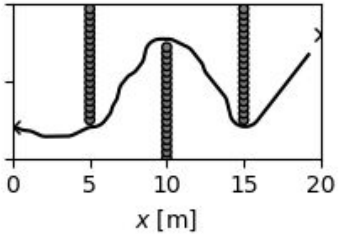


Research Methods

We test 2 path-finding algorithms on 10 different randomly generated maps, which simulates obstacles and barriers on streets.

Compare performance based on path lengths and execution times of algorithms.

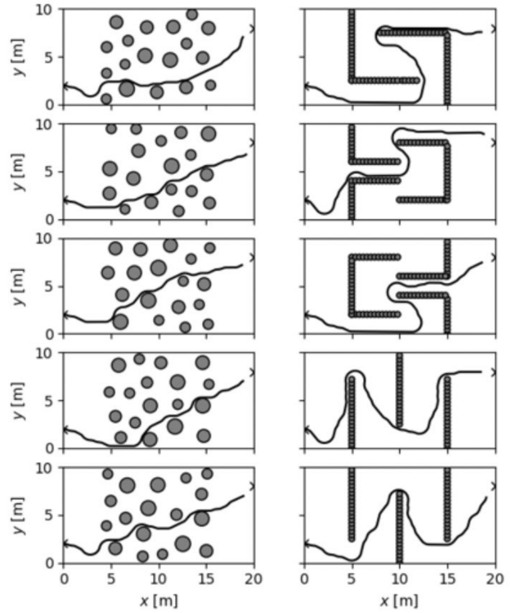




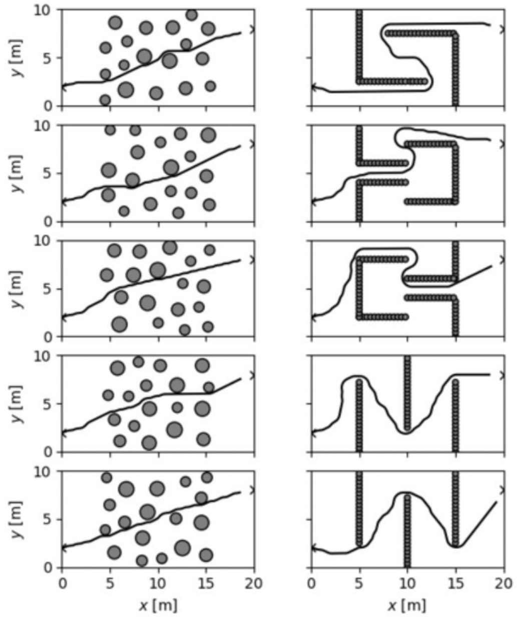
Results and Analysis

Paths generated by hybrid A* algorithm are slightly shorter than paths generated by hybrid Dijkstra.

Dijkstra

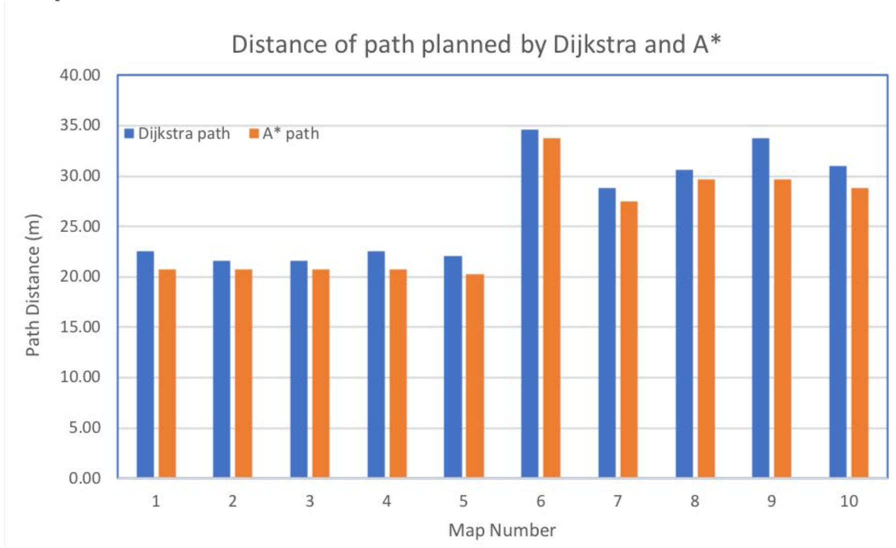


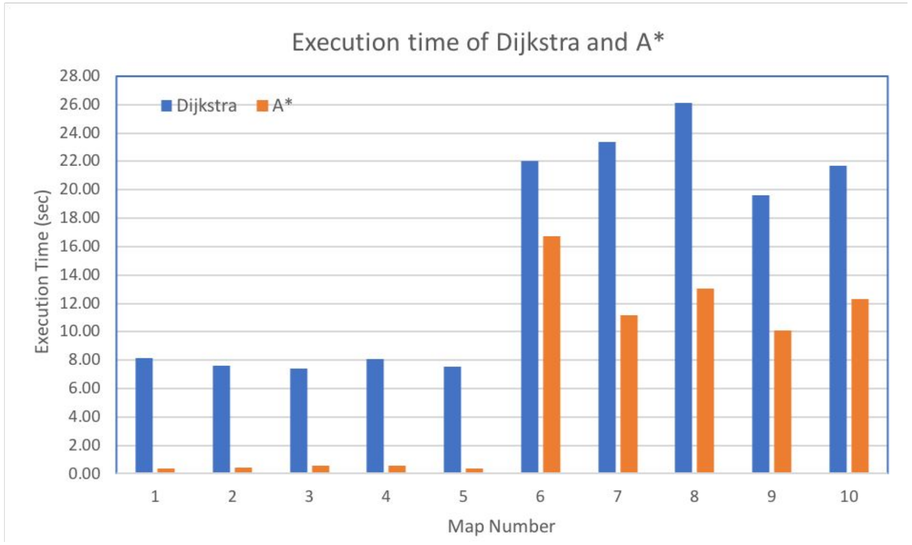
A*



Results and Analysis

Execution times of hybrid A* are shorter than hybrid Dijkstra because hybrid A* explores fewer unnecessary grid cells during the planning process.





Conclusions and Future Work

As hybrid Dijkstra performs worse not only in execution time but also in length of planned paths, one should avoid using Dijkstra in researches and applications related to auto-driving.

We could study more on how hybrid setting, i.e., discrete and continuous positions, affects the difference in path length between A* and Dijkstra algorithm in future.

Question?