

Research Review: Historical Developments in AI Planning Domain

For this research review, I tried to include three major algorithms from the autonomous driving domain. There are multiple challenges for finding the path for autonomous vehicles. The environment it interacts with is dynamic and constantly changing. More specifically, the state space of an autonomous vehicles is 3D kinematic. This makes it a complex continuous-variable optimization path-finding problem. All the three planning algorithms presented in this report were used in standard or improved form in the autonomous vehicle entries for the 2007 DARPA Grand Challenge, where robotic vehicles had to autonomously navigate parking lots [1].

Rapidly-exploring Random Tree (RRT): RRT is good at efficiently searching non-convex high-dimensional spaces particularly suited for path planning problems that involve obstacles and differential constraints (non-holonomic and kinodynamic) [2] [3] [4]. RRT builds a rooted tree from the starting node and expands by randomly sampling nodes in the search space till reaching the goal node. Finally, a continuous path from the starting node to the goal node is found. RRTs tend to be biased towards large unsearched areas of the search space [5]. RRTs remain fairly simple computationally and hence are widely used in the autonomous robotic motion planning domain. Many versions and extensions of the standard RRT has been developed enabling it to be used for real-time robotic vehicles, notably, in the 2007 MIT DARPA Urban Challenge vehicle – Talos (4th Place) [6]. Sample input for improving computational efficiency, lazy checks, risk-penalty in tree to incorporate uncertain environments etc., were some of the changes incorporated in Talos which demonstrated that RRT based planner was a good general-purpose motion planner capable of handling uncertain and very dynamic driving scenarios.

Anytime Dynamic A*: An ‘Anytime algorithm’ is an algorithm which returns a valid solution even upon interruption before its completion and keeps producing better results with time [7]. Anytime Dynamic A* (sometimes referred to as anytime D*), is a graph-based planning and replanning algorithm which produces sub-optimal solution in the initial state. As new information regarding the underlying graph is obtained, the algorithm repairs its previous solution and keeps bettering the solution within the given time [8]. This is a heuristic-based algorithm which combines dynamic (requiring replanning) and complex (requiring anytime approaches) aspects together. This algorithm was specifically designed for efficient path planning for land-based mobile robots. Anytime Dynamic A* was later used in Boss, the autonomous vehicle entry for DARPA 2007 Grand Challenge by the Carnegie Mellon Tartan Racing Group (1st Place) [9].

Hybrid State A* Algorithm: In 2008, Dolgov, D., Thrun, S., *et. al.* published their work on ‘Hybrid-State A*’, a modified version of the popular A* algorithm for **smooth** path finding of autonomous vehicles which most of the previous algorithms lacked. This research was motivated by their work on 2007 DARPA Urban Challenge [10]. Junior (Stanford Racing Team’s Robot in Urban Challenger, DARPA 2007 - 2nd Place), demonstrated flawless performance in complex general path-planning tasks using this path-planning algorithm. Hybrid-State A* is a modified version of A* with a modified state-update rule that captures continuous-state data in the discrete search nodes of A* [11]. Hybrid_State A* is not guaranteed to find a minimal-cost path but is practically drivable as it associates each grid cell with a continuous 3D state of the vehicle [11]. There were two main heuristics used in this research paper; first - “non-holonomic without-obstacles”—ignores obstacles but takes into account the non-holonomic nature of the car, second – ignores non-

holonomic nature of car and uses obstacle map to compute the shortest distance to the goal. The first heuristic does not depend on real-time sensors and hence can be pre-computed to help the search algorithm prune branches heading for the goal in possibly wrong ways. The second heuristic does dynamic programming in 2D to discover U-shaped obstacles and dead-ends and helps navigate the expensive 3-D search away from these areas [10]. The Hybrid A* generated paths are further processed and smoothed to remove unnecessary swerves and manoeuvres.

References

- [1] Wikipedia: DARPA 2007 Grand Challenge. [Online]. [https://en.wikipedia.org/wiki/DARPA_Grand_Challenge_\(2007\)](https://en.wikipedia.org/wiki/DARPA_Grand_Challenge_(2007))
- [2] Steve LaValle. Rapidly-exploring Random Trees. [Online]. <http://msl.cs.uiuc.edu/rrt/index.html>
- [3] Wikipedia: RRTs. [Online]. https://en.wikipedia.org/wiki/Rapidly-exploring_random_tree
- [4] James J Kuffner, Jr. Steven M LaValle, "Randomized Kinodynamic Planning," *The International Journal of Robotics Research (IJRR)*, 2001.
- [5] Steven M. Lavalle, "Rapidly-Exploring Random Trees: A New Tool for Path Planning," Computer Science Dept., Iowa State University, 1998.
- [6] Gaston A. Fiore, Justin Teo, Emilio Frazzoli, and Jonathan P. How Yoshiaki Kuwata, "Motion Planning for Urban Driving using RRT," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2008.
- [7] Wikipedia: Anytime Algorithm. [Online]. https://en.wikipedia.org/wiki/Anytime_algorithm
- [8] Dave Ferguson, Geoff Gordon, Anthony Stentz, Sebastian Thrun Maxim Likhachev, "Anytime Dynamic A*: An Anytime, Replanning Algorithm," in *ICAPS'05 Proceedings of the Fifteenth International Conference on International Conference on Automated Planning and Scheduling*, 2005.
- [9] Joshua Anhalt, et. al. Chris Urmson, "DARPA Urban Challenge Final Report," Carnegie Mellon University, 2007.
- [10] Sebastian Thrun, Michael Montemerlo, and James Diebel Dmitri Dolgov, "Practical Search Techniques in Path Planning for Autonomous Driving.," *The International Journal of Robotics Research*, 2010.
- [11] Sebastian Thrun, Robot Path Planning (Lecture Video: https://www.youtube.com/watch?time_continue=61&v=thepzbTYuJ8).
- [12]