

Motion Planning for Autonomous Driving

Philippe Weingertner, Minnie Ho

November 5, 2019

1 Introduction

2 Related Work

There is a rich literature related to Motion Planning and a very detailed survey of traditional methods is provided in [13]. Among the first 4 successful participants of DARPA Urban Challenge in 2007, the approaches vary. The winner, CMU Boss vehicle used variational techniques for local trajectory generation in a structured environment. This was done in a 2 steps path-velocity decomposition. A first step of path planning, using variational techniques, is performed and for every candidate path, a combination of different velocity profiles (constant, linear, linear ramp, trapezoidal) is applied : the combination of a path and velocity profile defines a trajectory. In unstructured environments (parking lots) or in error recovery situations a lattice graph in 4-dimensional configuration space (position, orientation and velocity) is searched with Anytime D* algorithm to find a collision-free path. More details are provided in [5, 1, 11]. The vehicle from Stanford used a search strategy coined Hybrid A* that constructs a tree of motion primitives by recursively applying a finite set of maneuvers. The search was guided by a carefully designed heuristic. The vehicle arriving 3rd, Victor Tango from Virginia Tech, constructs a graph discretization of possible maneuvers and searches the graph with the A* algorithm. The vehicle arriving 4th, developed by MIT used a variant of RRT algorithm with biased sampling. While all these techniques differ, they fundamentally rely on a graph search where nodes correspond to a configuration state and edges correspond to elementary

motion primitives. Although they provide solutions, the runtime and state space can grow exponentially large. In this context, the use of heuristic to guide the search is important.

More recently, Reinforcement Learning and Deep RL have been investigated in the context of Autonomous Driving for Decision Making either at the Behavioural Planning or Motion Planning level. In some research papers from Volvo [7] and BMW [6], an RL agent is trained in simulation to take decision at a higher tactical level: the decisions relate to a maneuver selection, like lane change, rather than a low level acceleration command. DQN is used to train an agent. But the problem with Reinforcement Learning is that the utility is optimized in expectation. So even if the reward is designed to avoid collisions, this will be optimized in expectation: ultimately it is as if safety would be enforced with soft constraints rather than hard constraints. Which is of course not acceptable for a real vehicle. To solve this problem in [6] an additional safety check layer is added after the DQN agent to eventually override the DQN agent decision if it is considered unsafe. Checking a decision wrt to a specific criteria is simpler than designing a decision making system that jointly optimizes efficiency, comfort and safety objectives. With RL applied to AD we have to account for additional safety checks. In [2] Deep RL is applied at the local planner level: the action space is a set of longitudinal accelerations $\{-4m/s^2, -2m/s^2, 0m/s^2, 2m/s^2\}$ applied along a given path at a T-intersection. Safety is handled in a different way here compared to previous BMW approach: the agent is constrained to choose among a restricted set of safe actions per state. So the safety is enforced before Deep RL. Ultimately car man-

ufacturers may want to combine both types of safety checks: constraining the action set per state before enabling an RL agent to make its own decision, and checking again the final sequence of decisions proposed by the RL agent.

Now the interesting topic is how to best combine traditional Motion Planning with RL. What are the limitations of these techniques in isolation and how to use the strengths of both approaches and circumvent their weaknesses.

3 Approach

4 Experimental Setup and Status

The source code is available here: CS221 Project

- Baseline: the decision is fast, immediate, but we find a collision free velocity profile only in 35% of the cases.
- Oracle: the search space is big. With a time step of 250 ms, looking for a collision free velocity profile over the next 100 meters we find in 100% of the cases a collision free solution; but in 47.2 seconds with UCS and in 190.7 seconds with DP running on an iCore9. We explored the graph with a depth of 24. UCS has a complexity of $\mathcal{O}(n \log n)$ with $n = |\text{states}|$. It potentially explores fewer states than DP.
- Planning with MCTS tree search
- Q-learning
- Combining Planning and Learning (with an efficient learned heuristic)

References

- [1] David Bissell et al. “Autonomous automobiles: The social impacts of driverless vehicles”. In: *Current Sociology* (Dec. 2018), p. 001139211881674. DOI: 10.1177/0011392118816743.
- [2] Maxime Bouton et al. “Reinforcement Learning with Probabilistic Guarantees for Autonomous Driving”. In: *CoRR* abs/1904.07189 (2019). arXiv: 1904.07189. URL: <http://arxiv.org/abs/1904.07189>.
- [3] Panpan Cai et al. “LeTS-Drive: Driving in a Crowd by Learning from Tree Search”. In: *CoRR* abs/1905.12197 (2019). arXiv: 1905.12197. URL: <http://arxiv.org/abs/1905.12197>.
- [4] Michael Everett, Yu Fan Chen, and Jonathan P. How. “Motion Planning Among Dynamic, Decision-Making Agents with Deep Reinforcement Learning”. In: *CoRR* abs/1805.01956 (2018). arXiv: 1805.01956. URL: <http://arxiv.org/abs/1805.01956>.
- [5] Dave Ferguson, Thomas M. Howard, and Maxim Likhachev. “Motion Planning in Urban Environments”. In: *The DARPA Urban Challenge*. 2009.
- [6] Andreas Folkers, Matthias Rick, and Christof BÄEskens. “Controlling an Autonomous Vehicle with Deep Reinforcement Learning”. In: June 2019. DOI: 10.1109/IVS.2019.8814124.
- [7] Carl-Johan Hoel, Krister Wolff, and Leo Laine. “Automated Speed and Lane Change Decision Making using Deep Reinforcement Learning”. In: *CoRR* abs/1803.10056 (2018). arXiv: 1803.10056. URL: <http://arxiv.org/abs/1803.10056>.
- [8] Carl-Johan Hoel et al. “Combining Planning and Deep Reinforcement Learning in Tactical Decision Making for Autonomous Driving”. In: *CoRR* abs/1905.02680 (2019). arXiv: 1905.02680. URL: <http://arxiv.org/abs/1905.02680>.

- [9] Mykel J. Kochenderfer. *Decision Making Under Uncertainty: Theory and Application*. MIT Press, 2015.
- [10] Edouard Leurent. “A Survey of State-Action Representations for Autonomous Driving”. working paper or preprint. Oct. 2018. URL: <https://hal.archives-ouvertes.fr/hal-01908175>.
- [11] M. McNaughton et al. “Motion planning for autonomous driving with a conformal spatiotemporal lattice”. In: *2011 IEEE International Conference on Robotics and Automation*. May 2011, pp. 4889–4895. DOI: 10.1109/ICRA.2011.5980223.
- [12] Subramanya Nagesh Rao, H. Eric Tseng, and Dimitar P. Filev. “Autonomous Highway Driving using Deep Reinforcement Learning”. In: *CoRR* abs/1904.00035 (2019). arXiv: 1904.00035. URL: <http://arxiv.org/abs/1904.00035>.
- [13] B. Paden et al. “A Survey of Motion Planning and Control Techniques for Self-Driving Urban Vehicles”. In: *IEEE Transactions on Intelligent Vehicles* 1.1 (Mar. 2016), pp. 33–55. DOI: 10.1109/TIV.2016.2578706.
- [14] F. Pusse and M. Klusch. “Hybrid Online POMDP Planning and Deep Reinforcement Learning for Safer Self-Driving Cars”. In: *2019 IEEE Intelligent Vehicles Symposium (IV)*. June 2019, pp. 1013–1020. DOI: 10.1109/IVS.2019.8814125.
- [15] Markus Schratter et al. “Pedestrian Collision Avoidance System for Scenarios with Occlusions”. In: *CoRR* abs/1904.11566 (2019). arXiv: 1904.11566. URL: <http://arxiv.org/abs/1904.11566>.
- [16] David Silver et al. “Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm”. In: *CoRR* abs/1712.01815 (2017). arXiv: 1712.01815. URL: <http://arxiv.org/abs/1712.01815>.
- [17] David Silver et al. “Mastering the game of Go without human knowledge”. In: *Nature* 550 (2017), pp. 354–359.