EE245 Project Final Report

A Study on Reinforcement Learning for Parking: Vision/Radar/LiDAR Sensing

Kunyi Yu
Student ID: 862548836
University of California, Riverside
Riverside, CA 92521
kyu135@ucr.edu

Abstract

5% The abstract content ...

1 Introduction 15%

Traffic accidents caused by human judgment errors remain a leading cause of fatalities worldwide. However, the rapid development of autonomous driving technologies holds promise for significantly reducing such incidents. In December 2024, the author visited the Bay Area and observed a growing presence of autonomous vehicles (AVs) on the road, either undergoing testing or already in commercial operation. Waymo, one of the industry pioneers, has been operating a fleet of AVs in San Francisco since August 2021. On the other hand, unlike Waymo's radar-based approach, Tesla's Full Self-Driving (FSD) system relies primarily on vision-based sensing. Tesla's commercial success demonstrates that a camera-only solution can be viable for autonomous driving.

Sensing modality, like vision, Radar or LiDAR, are widely researched and adopted in the industry, each with its own advantages and limitations. A 2024 news report [1] highlighted an incident where Waymo's AVs were excessively honking in a San Francisco parking lot, disturbing nearby residents multiple times at night. The issue was reportedly caused by interference from other vehicles, leading to a deadlock scenario—a common challenge in multi-agent systems. This also underscores the complexity and importance of autonomous parking as a research topic.

Thus, the modality of sensing is crucial for autonomous driving, especially in complex environments like parking lots where dense traffic and pedestrians could lead to problems such as deadlocks and security concerns. Reasonable solutions could be combined with different sensing modalities or dynamically switching between them to enhance the robustness of the system. Therefore, this project aims to explore the effectiveness of different sensing modalities in various parking scenarios by horizontally comparing the performance of several reinforcement learning (RL) algorithms.

In the rest of this report, Section 2 reviews related work in the the application of RL algorithms in autonomous parking and some common sensing modalities. Section 3 describes the methodology, including task definition, environment setup, RL algorithms, observation modalities, reward function and training settings. Section 4 presents the experimental results, including the performance comparison and findings. Finally, Section 5 includes summary, contributions, and future work.

2 Related Work 15%

This section reviews related work in the application of reinforcement learning (RL) algorithms in autonomous parking and common sensing modalities in autonomous driving. Most papers in this section are based on the `highway-env` environment [2], which is the platform used in this project and also widely adopted in the autonomous driving research community.

2.1 Reinforcement Learning in Autonomous Parking

The `highway-env` environment [2] is a widely used platform (2.9k star on GitHub) for simulating autonomous driving scenarios, including parking tasks. The well written documentation and abided by OpenAI Gym API make it easy to use and extend. The environment provides multiple sensing modalities (Kinematics, GrayScale Image, LiDAR, etc.), continuous/discrete action spaces, and predefined scenarios with reward functions.

There are several papers study the autonomous parking problem directly in the `highway-env` environment. Kapoor et al. [3] introduces a model-based RL approach that integrates neural dynamics prediction with Signal Temporal Logic (STL) guided model predictive control, applied to robotics and autonomous driving. This approach is demonstratesd on toy robotics tasks including a parking-lot scenario. The key strenth is its use of formal task specifications to helps avoid reward-shaping issues. A weakness is that it relies on a learned model and uses computationally expensive planning.

Moreira [4]'s master's thesis proposes a deep reinforcement learning method with SAC, DDPG, and TD3 algorithms to teach a wheeled vehicle to park in confined spaces. The agents are trained in a simulated environment to follow a predefined parking trajectory. The study finds that TD3 converges fastest and achieves the most reliable policy, while SAC also learns satisfactorily but DDPG is less stable and efficient. Strengths of this work include a thorough comparison and careful designed reward function. However, the method needs to predefined the parking path manually and leaves out perception or sensing issues.

Lazzaroni et al. [5] presents a DRL-based agent trained in different environments (a Unity-based, highway-env, and CARLA) for low-speed parking maneuvers, achieving a 97% success rate. The paper also uses Stable-Baselines3 [6] toolkits for training the agents, which provides a set of reliable RL algorithm implementations. The ego-vehicle is equipped with a 360-degree LiDAR sensor, and the agent observation also includes relative target position and self velocity. Training uses PPO algorithm with MLP. Although the high success rate is impressive, the long training time (over 60M timesteps) and needs for hyperparameter tuning are drawbacks.

Note that Stable-Baselines3 [6] is library of reliable implementations of standard deep reinforcement learning algorithms. It is not specifically designed for autonomous driving, but provides high-quality code for many algorithms (PPO, DDPG, SAC, TD3, A2C, etc.) with consistent API. There also exist some YouTube videos [7, 8] visulizing the training process of reverse parking and parallel parking. The former video uses a version of Rainbow DQN on raw camera images, showing that even a complex 3D task like reverse parking can be learned but missing performance metrics and robustness analysis due to the pattern of videos. The latter one utilizes a PPO-based policy using Unity3D environment. However, details on sensors (maybe also vision-based), reward function, or evaluation are not provided.

A survey paper [9] provides a comprehensive overview of RL applications in autonomous driving, not limited to only parking scenarios. With in this survey, some instances parking tasks are discussed, including a RL-guided Monte Carlo Tree Search (MCTS) algorithm research. Strengths is its comprehensive coverage of perception, planing, and control. As it is a high-level review, however, sensor modalities for parking are not included.

2.2 Sensing Modalities in Autonomous Driving

The operational logic of autonomous system, from a simple housekeeping robot to a complex self-driving car, can be summaryized into a foundamental loop: See, Think, Act. In this paradigm, the "See" component-perception-serves as the foundation layer upon which all subsequent capabilities are built. To meet the stringent safety and reliability demands of autonomous driving, the academic and industrial have invested significant sensing strategies: cameras (Vision), Radio Detection and

Ranging (Radar), and Light Detection and Ranging (LiDAR). As no signle sensor can provide a complete and infallible representation of the environment, each sensing modality provide unique advantages and inherent limitations.

Yurtsever et al. [10] provides a comprehensive survey on common practices and emerging trends, including serveral sensoring technologies: vision-based sensors (monocular cameras, omnidirectional cameras, and event cameras), radar, and LiDAR (Point clouds). Firstly, vision-based sensors have advantages in color sensing, passive sensing, and low cost due to established technology. However, their illumination sensitivity and difficulty in depth perception are significant drawbacks. Radar sensors, on the other hand, have better long range detection, robustness to bad weather, and also low cost. However, they have lower resolution and their field of view is limited. Lastly, LiDAR have pros in high accuracy, accuracy in depth perception, and robustness to illumination changes. However, they are expensive, heavy, and require large-scale data processing. If combing these sensors, the advantages of each sensor can be utilized.

Some studies also explore the integration of multiple sensing modalities. Pederiva et al. [11] alleviates the computationally demands of tranditional detecting objects by integrating Camera and LiDAR data. Their method utilizes cutting-edge Deep Learning techniques into the feature extraction process achieving a 2.39% accuracy improvement and 10% parameter reduction within 10 ms inference time. The lightweight and powerful model is suitable for real-time applications.

3 Methodology 30%

This section describes the methodology of this project, including task definition, environment setup, RL algorithms, observation modalities, reward function and training settings.

3.1 Task Definition

4 Experiments 30%

5 Conclusion 5%

Acknowledgments

The final project was independently conducted by the author Kunyi Yu. Meanwhile, the author would like to thank Professor Jiachen Li and TAs for their patient guidance and help.

References

[1] Tim Johns. Waymo cars honk at each other throughout the night, disturbing sf neighbors, 2024. URL https://abc7ne.ws/3Ai47Ci.

[2] Edouard Leurent. An environment for autonomous driving decision-making. https://github.com/eleurent/highway-env, 2018.

[3] Parv Kapoor, Anand Balakrishnan, and Jyotirmoy V Deshmukh. Model-based reinforcement learning from signal temporal logic specifications. *arXiv preprint arXiv:2011.04950*, 2020.

[4] Dinis Moreira. Deep reinforcement learning for automated parking. Master's thesis, Universidade do Porto (Portugal), 2021.

[5] Luca Lazzaroni, Alessandro Pighetti, Francesco Bellotti, Alessio Capello, Marianna Cossu, and Riccardo Berta. Automated parking in carla: A deep reinforcement learning-based approach. In *International Conference on Applications in Electronics Pervading Industry, Environment and Society*, pages 352–357. Springer, 2023.

[6] Antonin Raffin, Ashley Hill, Adam Gleave, Anssi Kanervisto, Maximilian Ernestus, and Noah Dormann. Stable-baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research*, 22(268):1–8, 2021. URL http://jmlr.org/papers/v22/20-1364.html.

[7] Samuel Arzt. Ai learns to park - deep reinforcement learning, 2019. URL https://www.youtube.com/watch?v=VMp6pq6_QjI.

[8] Matthew Euliano. Parallel parking using reinforcement learning - unity3d and ml-agents, 2022. URL https://www.youtube.com/watch?v=QDVwSAgY6cA.

[9] Badr Ben Elallid, Nabil Benamar, Abdelhakim Senhaji Hafid, Tajjeeddine Rachidi, and Nabil Mrani. A comprehensive survey on the application of deep and reinforcement learning approaches in autonomous driving. *Journal of King Saud University-Computer and Information Sciences*, 34(9):7366–7390, 2022.

[10] Ekim Yurtsever, Jacob Lambert, Alexander Carballo, and Kazuya Takeda. A survey of autonomous driving: Common practices and emerging technologies. *IEEE access*, 8:58443–58469, 2020.

[11] Marcelo Eduardo Pederiva, José Mario De Martino, and Alessandro Zimmer. A light perspective for 3d object detection. In *Seventeenth International Conference on Machine Vision (ICMV 2024)*, volume 13517, pages 136–143. SPIE, 2025.

Appendix

How to Find the Code?

The GitHub public repository for this project is at https://github.com/Rock3Yu/EE245_Advanced_Robotics. Please feel free to explore the code and data files. The structure of the repository is as follows:

```
.
|-- final
|   |-- main.pdf
|   |-- main.tex
|   |-- ...
```

```
  |-- proposal
  |   |-- main.pdf
  |   |-- main.tex
  |   |-- ...
  |-- src
      |-- HighwayEnv
      |-- make_gif.py
      |-- models
      |-- parking_config.py
      |-- smallCNN.py
      |-- tensorboard_logs
      |-- train_0_kinematics.py
      |-- train_1_vision.py
      |-- train_2_lidar.py
      |-- ...
# The purpose of each dictory/file is as their name suggests.
```