Agent-based Obstacle Bypassing System

Yutong Zhou
Department of Information Engineering
The Chinese University of Hong Kong
Shatin, Hong Kong
zy319@ie.cuhk.edu.hk

Zhehao Jiang
Department of Information Engineering
The Chinese University of Hong Kong
Shatin, Hong Kong
jz018@ie.cuhk.edu.hk

Abstract

Self-driving cars are a popular topic in recent years. More and more researchers pay attention on the self-driving system, and some reinforcement learning methods are used in this topic. In this report, we simplify the self-driving system to an agent-based obstacle bypassing system. We build an environment from real world dataset and test the performance of Proximal Policy Optimization (PPO) with different actor-crisis networks.

1 Introduction

In 1977, the first self-driving car was developed by Japan's Tsukuba Mechanical Engineering Laboratory. However, the self-driving car cannot work without the support of an elevated rail. In the 1980s, researchers from Carnegie Mellon University created a self-driving car that can detect the lanes by the camera, and drivers only need to concentrate on stepping on the gas and hitting the brace (2). As entering the new century, the United States' Defense Advanced Research Projects Agency (DAPRA) organized the first match to pass the Mojave desert, but no one finished this game. However, the self-driving from Stanford University got the highest score in the match, and the sensors on the self-driving cars are cameras, radars, and 3D laser scanners, which are similar to modern self-driving cars. Meanwhile, machine learning had been used to analyze the collected data at that time and showed the way of the future self-driving system.

Recent years have seen the rapid development of deep learning because of the increment of computational capacity. Research groups from academia and industry pay more and more attention to combining deep learning and self-driving system. In 2009, Google started its research on the self-driving system. Four years later, automotive industries like Audi, Ford began to study the self-driving system. In 2013, Tesla proposed its first self-driving system, Autopilot. When the industrial companies spent more time on the self-driving system, more data can be collected. Dataset like KITTI (4), Apolloscape (7) helps researchers test their object detection based on the deep neural network and their control system based on deep reinforcement learning. The collaboration of academia and industry builds a conformable zone of the development of the self-driving system.

Most dataset of self-driving systems contains data from multiple sensors like camera, radar, and Lidar. In our project, we build a global map from dataset and simplify the self-driving problem to an obstacle bypassing system problem. We first build a global environment around the self-driving car and then use Proximal Policy Optimization (PPO) to control the self-driving car. In this project, we use different types of data as input and different network architectures to test the performance in the built environment.

2 Related Work

Over the years, many researchers have studied autonomous mobile obstacle avoidance through reinforcement learning. In 2002, a neural network (8) used for discrete input spatial coding was

proposed to make the structural credit problem, which is vital for the obstacle avoidance problem, a faster and more effective solution.Three years later, Huang (6) integrated Q-learning and neural networks to enhance the learning capabilities of autonomous robots and enable the robots to complete the given tasks in a complex and unpredictable environment. With a similar idea, (3) presented an obstacle avoidance solution based on double neural networks, and the double neural networks use Q-learning reinforcement technology to focus on inverse kinematics and multi-obstacle avoidance problems in complex unknown environments. In 2017, Zhao (9) provided an Adaptive and Random Exploration approach (ARE) method based on Q-learning to complete obstacle avoidance and navigation tasks for drones.

3 Methodology

The self-driving system is complex, and it contains object detection, semantic segmentation, and control system. In our project, we focus on how to bypass the obstacle based on reinforcement learning.

3.1 Dataset

In this project, we use the dataset from Waymo (1), a past Google Project, and Waymo became a stand-alone company in 2016.The dataset contains 1,950 segments in diverse geographies and conditions, and the duration of each segment is 20 seconds. These data are collected from two types of sensors (five Lidars and five Cameras), and all objects are divided into five categories, vehicles, pedestrians, cyclists, signs, and unknown. Meanwhile, 12.6M 3D bounding box labels with tracking IDs are labeled on Lidar data in 1,200 segments, as shown in Fig. 1, and 11.8M 2D bounding box labels with tracking IDs are labeled on Camera data, showing in Fig. 2. Considering how to construct the global environment of the system, we only use Lidar data in this project

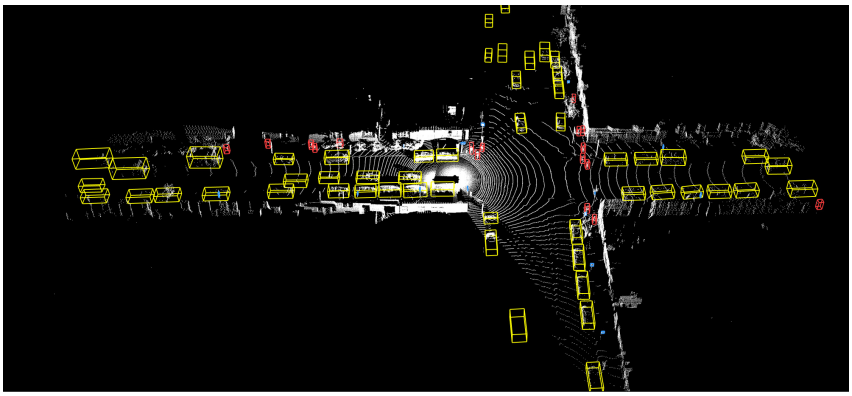

Figure 1: 3D bounding box labels for Lidar data

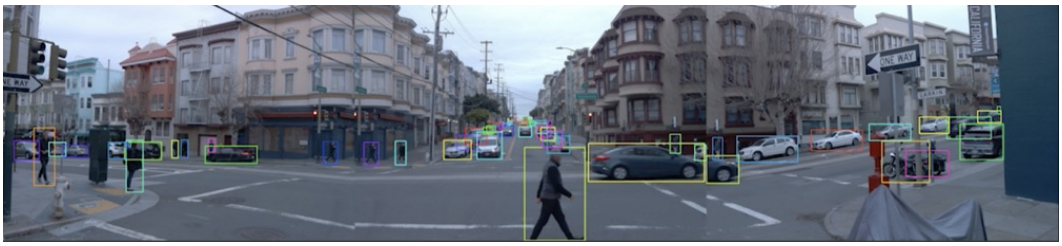

Figure 2: 2D bounding box labels for Camera data

3.2 Environment

First, we need to process the data and establish our environment. The Waymo dataset contains two types of data collected from Lidar and Camera. Usually, Lidar data provide 3D coordinates of

surrounding objects, while camera data provides the type of objects. As shown in 3, our goal is to build a global environment from an aerial view. However, we found that without an image from a bird's eye view, a global RGB map cannot be generated.Hence, we decided to generate a global map from the Lidar data, which can give us information about position, speed, and acceleration, as shown in Fig. 4. In order to simplify the map, we define the initial position of the self-driving car as the origin of the coordinates, and then divide the map into $50 \times 50 = 2,500$ grids (each grid is about 1.5 meters in length). When new Lidar data is input, we need to perform the coordinate conversion and pass the new global map to our self-driving car. Considering the global map as an image, each grid is a pixel with one channel or three channels. When there is only one channel, it indicates whether an obstacle occupies the pixel (grid). The two more channels represents the speed and acceleration of the obstacles.

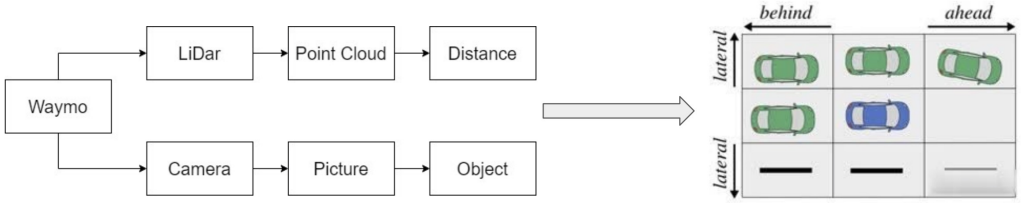

Figure 3: Establish environment from dataset

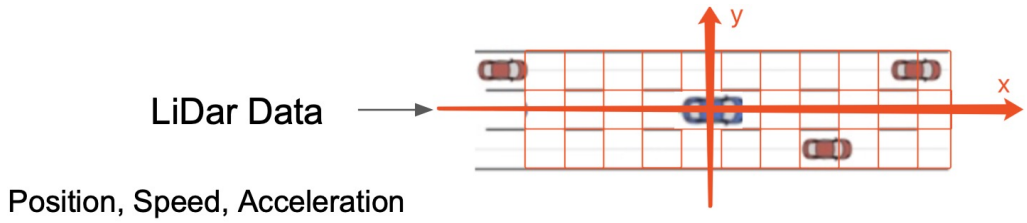

Figure 4: Global map generated from Lidar

3.3 Action

The real self-driving system's control system is complex due to the real environment and control panel in the real vehicle. To simplify the question, we define that the self-driving car can only move straight, turn right, or turn left within ten grids, as shown in Fig. 5. At current step, we assume that the change of action can be finished immediately.

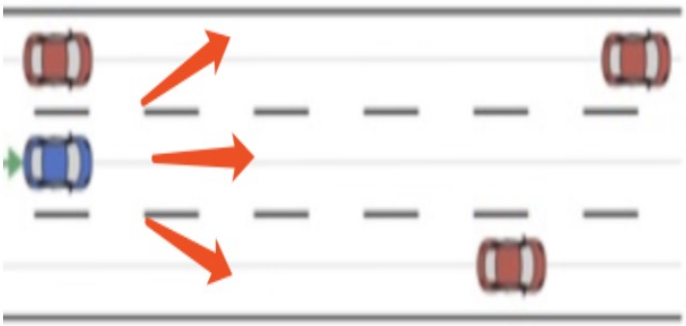

Figure 5: 2D bounding box labels for camera data

3.4 Rewards

Since it is a simple obstacle bypassing system, we define the reward function as

$$R(x) = \begin{cases} 1, & x = 1 \\ 0, & x = 0 \end{cases} \tag{1}$$

The symbol x defines whether the self-driving car hits the obstacle in current map/frame ("1" for not).

3.5 Agent

In our project, we choose the PPO algorithm as the agent. PPO collects a small batch of experiences interacting with the environment and uses that batch to update its decision-making policy. Once the policy is updated with this batch, the experiences are thrown away and a newer batch is collected with the newly updated policy. The key contribution of PPO is ensuring that a new update of the policy does not change it too much from the previous policy. This lead to less variance in training at the cost of some bias, but ensures smoother training and also makes sure the agents does not go down an unrecoverable path of taking senseless actions. The idea of PPO can be found in Fig.6

$$E_{(s_t,a_t)\sim\pi_{\theta'}}[\frac{p_\theta(a_t|s_t)}{p_{\theta'}(a_t|s_t)}\frac{p_\theta(s_t)}{p_{\theta'}(s_t)}A^{\theta'}(s_t,a_t)\nabla log p_\theta(a_t^n|s_t^n)]$$

Figure 6: PPO

s_t is the current state

a_t is the action

θ is the policy parameter

E denotes the empirical expectation over timesteps

π_θ is the policy with the policy parameter θ

$P_{\theta'}$ and P_θ is the ratio of the probability under the new and old policies, respectively

$A^{\theta'}$ is the estimated advantages at time t.

Meanwhile, we use multi-layer perceptron (MLP) and ResNet-18 (5) as the actor-critic networks. ResNet is a residual neural network designed for image recognition and it perform well in the image recognition. The shape of the agent input is $1 \times 50 \times 50$ or $3 \times 50 \times 50$ and it is similar to the input of neural network designed for image recognition ($1 \times M \times N$ or $3 \times M \times N$). Hence, we choose ResNet-18 as the actor-critic network.

4 Experiments

In the experiment, we test two different networks, MLP and ResNet-18, as actor-critic in PPO. The results is shown in Table 1, row 1. When the input channel is only one, which means only position information is used, the performance of ResNet-18 is twice of MLP. However, the duration of each segment is 20 seconds and each segment has around 200,000 frames, which means the total rewards should be around 200,000. In order to improve the performance of agent, we add speed and acceleration of surrounding obstacles and the result is shown in Table 1, row 2. It is obvious that the performance of MLP and ResNet-18 improves. However, the performance is still not good because the ResNet-18 is not designed for control system but for classification.

5 Conclusion

In this project, we established an environment for agent-based bypass system from Waymo dataset and test the PPO algorithm with MLP and ResNet-18. However, the performance of the agent is not satisfying. There are several reasons that the agent is not work well in this case. First, the environment

	MLP	ResNet-18
Position	3.569	7.865
Position + speed + accelerate	4.204	18.357

Table 1: Comparison of MLP and ResNet-18

is only generated from Lidar data and the agent lose information about lane, which is important for the self-driving to bypass the obstacles. Second, we use MLP and ResNet-18 as the actor-critic network. However, MLP is too simple to use while ResNet-18 is not designed for control system. In order to improve the performance of this agent, a network designed for control system should be created and it should be able to predict the trajectory of surrounding obstacles.

References

[1] D. Baltieri, R. Vezzani, and R. Cucchiara. 3dpes: 3d people dataset for surveillance and forensics. In *Proceedings of the 1st International ACM Workshop on Multimedia access to 3D Human Objects*, pages 59–64, Scottsdale, Arizona, USA, Nov. 2011.

[2] E. D. Dickmanns. Vision for ground vehicles: History and prospects. *International Journal of Vehicle Autonomous Systems*, 1(1):1–44, 2002.

[3] M. Duguleana, F. G. Barbuceanu, A. Teirelbar, and G. Mogan. Obstacle avoidance of redundant manipulators using neural networks based reinforcement learning. *Robotics and Computer-Integrated Manufacturing*, 28(2):132–146, 2012.

[4] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[5] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. arxiv 2015. *arXiv preprint arXiv:1512.03385*, 2015.

[6] B.-Q. Huang, G.-Y. Cao, and M. Guo. Reinforcement learning neural network to the problem of autonomous mobile robot obstacle avoidance. In *2005 International Conference on Machine Learning and Cybernetics*, volume 1, pages 85–89. IEEE, 2005.

[7] X. Huang, X. Cheng, Q. Geng, B. Cao, D. Zhou, P. Wang, Y. Lin, and R. Yang. The apolloscape dataset for autonomous driving. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 954–960, 2018.

[8] K. Macek, I. PetroviC, and N. Peric. A reinforcement learning approach to obstacle avoidance of mobile robots. In *7th International Workshop on Advanced Motion Control. Proceedings (Cat. No. 02TH8623)*, pages 462–466. IEEE, 2002.

[9] Z. Yijing, Z. Zheng, Z. Xiaoyi, and L. Yang. Q learning algorithm based uav path learning and obstacle avoidence approach. In *2017 36th Chinese Control Conference (CCC)*, pages 3397–3402. IEEE, 2017.