Reinforcement Learning to Drive in Unstructured Environments*

Minsoo Kim¹ and Jaeheung Park¹

Abstract—In this paper, we present a method to drive in unstructured environments using reinforcement learning. We propose construction of state and reward in reinforcement learning for safe driving at unstructured environments. Experimental results show the performance of the proposed method in a simulation.

I. INTRODUCTION

One of the important issues for fully automated driving is to drive in an unstructured environment. Especially, in a parking lot, various static obstacles exist and dynamic obstacles move without any rule. Therefore, it is difficult to drive in the mentioned environments by only using the rule-based approach.

Many studies use the reinforcement learning based method to make a driving policy. In particular, End-to-End approach [1] is generally used to determine the steering angle of the vehicle by using a raw front image of the camera as an input. However, the raw front image is high-dimensional data, so that it is difficult to train the driving policy. Also, they only consider structured and simple environments.

In this paper, we propose a method to determine steering angle in unstructured environments using reinforcement learning. The proposed method refines the raw front image as low-dimensional information. In addition, in order to prevent collision with obstacles, the risk is predicted for the front area of the vehicle as well as the collision information, and is configured as reward. We apply Deep Deterministic Policy Gradient method [2], which is one of the reinforcement learning algorithms, having an advantage in continuous control.

II. PROBLEM DEFINITION

The purpose of our method is to drive in unstructured environments with avoiding obstacles using reinforcement learning. We assume that a reference path does not exist, and only static obstacles are considered. Also, the occupancy grid map can be obtained. It represents the existence of obstacles at the grid cell. Its size is 40 x 40 (Occ Map) with 0.3[m] resolution. The origin of Occ Map is on a center of the vehicle in Fig. 1(a), such that information of obstacles about both sides and front of the vehicle is received. The velocity of the vehicle is constant, 21[m/s].

III. PROPOSED SOLVING APPROACH

We introduce a configuration of reinforcement learning for unstructured environments and Deep Deterministic Policy Gradient (DDPG) algorithm.

A. Configuration of Reinforcement Learning

• State: 40 x 40 occupancy grid map and previous steering angle are received from the environments. Each column of the occupancy grid map is converted to a scalar which represents how close obstacles are (red dotted box in Fig. 1(a)). Therefore, the 40 x 40 occupancy grid map is compacted as a 1-D array which has 40 dimensions. To control steering angle smoothly, the previous steering angle is considered in the state as well. The previous steering angle is appended to the compacted array.

• Reward: Reward function is defined as,

\[ R_t = \omega_c C_t + \omega_f \frac{n_{free}}{n_{total}} + \omega_h |H_t - H_{t-1}|. \]

Three aspects are considered in the reward function; i) Collision (\( C_t \in \{0, 1\} \)). ‘\( C_t = 0 \)’ means no collision and ‘\( C_t = 1 \)’ means collision. ii) A ratio of the total number of free grid cells (\( n_{free} \)) in Region Of Interest (ROI) around the vehicle (blue rectangle in Fig. 1(b)). In the ROI, the total number of grid cells (\( n_{total} \)) is 180. iii) Difference between previous and determined (\( H_t \)) steering angle. \( \omega_c, \omega_f \) and \( \omega_h \) are weight-parameters with -1, +0.002 and -0.004.

• Action: In Fig. 1(c), the action is a value having a range of -1.0 to +1.0. The value with multiplied by 540 is used as a steering input of a vehicle.

*This work was not supported by any organization

¹are with the DYROS (Dynamic Robotics Systems) Lab, Graduate School of Convergence Science and Technology, Seoul National University, Seoul, Republic of Korea. Jaeheung Park is the corresponding author. minsoo@korea.ac.kr, park73@snu.ac.kr

Fig. 1: The configuration of the proposed method, (a) process of refining occupancy grid map as State, (b) three cases considered for Reward function, (c) Action in a range of -1.0 to +1.0
B. Deep Deterministic Policy Gradient (DDPG)

We use Deep Deterministic Policy Gradient (DDPG) algorithm, which has an advantage in continuous control. There are three characteristics in the DDPG: i) It uses Actor-Critic model [3] with using a deterministic policy [4]. The actor network aims at improving the policy, and the critic network evaluates the policy. The Actor network which uses a policy gradient method enables to address the continuous output. Using the Critic network separately, it reduces the variance, such that the learning is more stable than pure policy gradient methods. ii) A target network ($\theta^t$) is applied. The target network ($\theta^t$) is slightly updated every times, which makes the network stably updates the action. The target network ($\theta^t$) is updated as follows.

$$\theta^t \leftarrow \tau \theta^t + (1 - \tau) \theta^t,$$

(2)

where $\theta$ represents weights of the networks and $\tau$ is a soft-update term to determine speed of tracking the weights of the current network ($\theta^t$). iii) Exploration issue of the deterministic policy is addressed by using Ornstein-Uhlenbeck (OU) Process [5] as exploration noise. The action ($a_t$) is chosen as follows.

$$a_t \leftarrow \pi(s_t | \theta^t) + N(\omega, \mu, \sigma),$$

(3)

where $\pi(s_t | \theta^t)$ is an action using a deterministic policy ($\pi$) given state ($s_t$). $N$ represents the exploration noise. The exploration noise ($N$) is determined by Gaussian distribution with mean ($\mu$) and standard deviation ($\sigma$). Variable reversion term ($\omega$) represents convergence speed of the noise moving to the mean ($\mu$).

We set total episodes as 20000 iterations with batch size 32. Discount factor is 0.9 and learning rates are 0.0001 for Actor network and 0.001 for Critic network. Both networks consist of two hidden layers with 300 and 600 units. The soft-update term, $\tau$ in (1) is 0.001, and the parameters, $\omega$, $\mu$ and $\sigma$ in (2) are set to be 0.6, 0.0 and 0.3, respectively. The hyperbolic tangent is used as an activation function, which makes the output in a range of -1.0 to 1.0.

IV. EXPERIMENT RESULT

The proposed method is trained and tested in two unstructured environments in CARLA simulator. A trajectory of the trained vehicle is indicated in the simple map and the parking lot in Fig. 2 (green line). In the simple map shown in Fig. 2(a), the vehicle drove with minimum change of steering angle in areas with few obstacles (yellow box), and could pass through the sharp-curve area (red box) without any collision. In the parking lot shown in Fig. 2(b), the vehicle drives with avoiding various types of obstacles. Especially, at narrow areas #1 and #2 (white dotted boxes in Fig. 2(b)), it is difficult to drive at high velocity (21m/s) for human. However, our tested model could drive without any collision. Online videos show the performance of the trained vehicle at the simple map (https://youtu.be/cVAur8zzuLA) and the parking lot (https://youtu.be/zAwX7OTMrZ4).

V. CONCLUSION

This paper presents the method to drive in unstructured environments using Deep Deterministic Policy Gradient reinforcement learning algorithm. The proposed method was trained and tested in a simulation using the occupancy grid map input. The results show that the vehicle can drive in unstructured environments without collision. We plan to consider, i) dynamic obstacles, ii) velocity control and iii) testing the model in different environments as future work.

REFERENCES


