

심층 강화학습을 이용한 이동 장애물 회피 및 최적 경로 탐색 기법

송연지^{1,◦}, 유영재², 이충연², 장병탁²

홍콩과학기술대학교¹, 서울대학교²

yjsongab@connect.ust.hk, {yjyoo, cylee, btzhang}@bi.snu.ac.kr

Deep RL-based Optimal Path Planning and Obstacle Avoidance for Mobile Robots

YeonJi Song^{1,◦}, Youngjae Yoo², Chung-Yeon Lee², Byoung-Tak Zhang²

Hong Kong University of Science and Technology¹, Seoul National University²

요약

본 논문에서는 실내 환경에서 로봇이 스스로 학습하고 움직이는 장애물을 피하는 심층 강화학습(Deep Reinforcement Learning) 기반의 알고리즘을 제안한다. 학습 모델은 전방 카메라로부터 받아오는 이미지와 라이다 센서로부터 받아오는 전방 물체와의 거리 값, 그리고 현재 위치에서 목적지까지의 방향각을 입력으로 받아와서 시뮬레이션 환경에서 학습을 진행한다. 로봇이 목적지를 탐색하는 과정에서 매 위치에 어떤 행동을 취하였는지의 정보를 저장하는 알고리즘을 생성하여 목적지에 효율적으로 도착했음을 볼 수 있다.

1 서론

최근 서비스 로봇의 산업적 수요가 늘어남에 따라 동적인 실내 환경에서의 장애물 회피와 최적경로 계획 등 로봇의 자율주행 연구에 대한 관심이 높아지고 있다. 이와 관련하여 고려되는 주요 방법 중 하나는 심층 강화학습(Deep reinforcement learning) [1]이다. 대규모 테스트를 통해 자체 데이터를 생성하고, 최상의 결과에 도달하도록 돋는 행동 패턴들을 스스로 학습하는 심층 강화학습은 이미 알파고를 통해 그 잠재력이 증명되었고, 협동로봇(Collaborative robot)이나 무인자동차의 자율주행을 포함한 다양한 분야에서 활발히 연구가 진행되고 있다 [2]. 본 논문에서는 1) 로봇이 RGB 카메라와 LiDAR 센서를 통해 실시간으로 주변 공간 데이터를 획득하고, 이 데이터를 통해 학습된 심층 강화학습 모델을 이용하여 목적지까지의 최적 경로를 탐색하는 자율주행 알고리즘을 제안한다. 또한, 2) 대규모 실험을 위해 실내 공간을 모사하여 구축한 시뮬레이션 환경과 3) 로봇이 자율주행 도중에 움직이는 물체들을 인식하고, 이를 회피하면서 목표 지점까지 도달하는 실험 결과를 제시한다.

2 관련 연구

최적경로 탐색 알고리즘. 기존의 최단경로 탐색 모델들은 주로 경로의 단일 속성만을 고려한다 [3]. 즉, 일반적인 목적지의 중요도, 장애물, 방문 빈도 등을 고려하지 않은 최단경로 탐색을 수행한다. 그러나 사람은 일상생활에서 길의 특성 혹은 통행 시간과 같은 다양한 속성을 종합적으로 고려하여 최적의 경로를 선택한다 [4]. 실생활에서 로봇의 효율적인 자율주행을 위해서는 이와 같이 기존 경로탐색 알고리즘에 다양한 속성을 반영하여 최적경로를 탐색할 필요가 있다.

심층 강화학습(Deep Reinforcement Learning). 심층 강화학습이란 자율적 에이전트가 강화학습의 시행착오 알고리즘과 누적 보상

함수를 이용해 신경망 디자인을 가속화하는 방식을 일컫는다 [1]. 알려지지 않은 환경에서 임의의 행동(Action)을 수행하는 경험을 반복하면서 이에 따른 보상(Reward)을 통해 학습해나간다 [5].

심층 강화학습 기반의 알고리즘인 Deep Q-Networks (DQN)은 학습 정책(Policy)과 행동 정책이 달라도 학습이 가능한 Off-Policy 알고리즘이며, 정책이 업데이트되더라도 이전의 경험들을 학습에 사용할 수 있다는 것이 심층 강화학습 알고리즘의 특징이다.

3 연구 내용

3.1 심층 강화학습 알고리즘

본 논문에서 제안하는 자율주행 방법은 그림 1과 같다. 제안하는 방법에서는 로봇의 전방 카메라와 라이다 센서를 통해 받아오는 정보를 입력으로 한다. 입력된 이미지는 합성곱 신경망(CNN)으로 들어가게 된다. 이때 이미지와 현재 위치에서 목적지까지의 방향

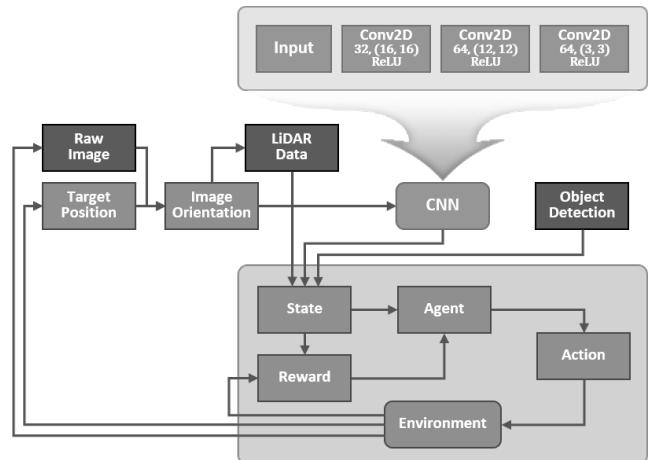


그림 1: 카메라 센서, 라이다 센서 데이터와 방향각을 입력으로 하는 심층 강화학습 알고리즘과 물체 인식 알고리즘 모델

값인 타겟 값을 입력으로 사용한다 [6]. CNN에서 연산량을 줄이기 위해 이미지를 자르거나 전처리를 거칠 수 있지만, 시뮬레이션 환경이 단조롭기 때문에 이미지를 입력으로 그대로 사용하였다. 라이다 센서 데이터는 로봇의 좌우로 90° 씩 총 180° 를 스캔하며 센서의 위치는 각 10° 씩 벌어져 있어서 총 11개가 장착되어 있다.

입력 이미지 x_t 의 특징을 추출한 CNN의 결과와 라이다 센서 데이터는 강화학습의 상태로 들어가게 된다. 에이전트는 상태와 보상을 받아서 그에 알맞은 행동 a_t 을 취하면서 학습한다.

방향 전환은 전방에 장애물을 인식할 시 라이다 센서를 통해 좌우를 살피 후 적당한 방향으로 90° 회전하도록 하였다. 시뮬레이션의 제한된 리소스로 인해 회전은 90° 씩 하도록 하였으며 이에 적합하도록 테스트 환경을 생성하였다.

보상 r_t 은 환경과 충돌할 경우 -5를 주며, 한 에피소드가 끝나고 다음 에피소드를 시작한다. 무제한적인 탐색을 방지하고, 실제 환경에서 로봇의 자원을 고려하여 목적지에 도달하지 못하거나 충돌하지 않아도 보상 r_t 이 -20에 도달하면 에피소드가 끝나게 된다.

로봇이 목적지 $0.5m$ 이내에 위치하면 도착으로 간주하고 +14의 보상을 주며 에피소드가 끝난다. 목적지까지 이동하는 동안에 매 스텝에 대한 보상은 “(이전 위치에서 목적지까지의 거리 + 현재 위치에서 목적지까지의 거리) - 3”으로 계산이 된다. 이 보상 계산식을 통해 목적지 방향으로 이동하면 양수 보상을 받고, 잘못된 방향으로 이동하면 음수 보상을 받게 하였다.

학습을 진행하는 총 스텝 수는 7.5M, 학습률은 0.0007이다. 환경과 입력을 아래와 같이 적용하여 실험을 진행하였고, 사용된 코드는 실험 재현을 위해 공개하였다¹⁾.

Algorithm 1 심층 강화학습 알고리즘

```

Initialize replay memory  $\mathcal{D}$ 
Initialize action-value function  $Q$  with random weights
for episode = 1, 25000 do
    Initialize seq.  $s_1 = \{x_1\}$  and preprocessed seq.  $\phi = \phi(s_1)$ 
    for t = 1, 300 do
        With probability  $\epsilon$  select a random action  $a_t$ 
        otherwise select  $a_t = \max_a Q^*(\phi(s_t), a; \theta)$ 
        Execute action  $a_t$  and observe reward  $r_t$  and image  $x_{t+1}$ 
        Memorize action  $a_t$ , reward  $r_t$ , position  $(x, y)$ 
        Store transition  $(\phi_t, a_t, r_t, \phi(t+1))$  in  $\mathcal{D}$ )
    end for
end for

```

3.2 움직이는 물체 인식 알고리즘

시뮬레이션 환경에 실생활을 최대한 반영하기 위해 움직이는 물체를 생성하였다. 심층 강화학습 알고리즘 만으로는 움직이는 물체를 인식하고 이에 알맞은 행동을 취하는 데 한계가 있는데 장애물

을 피하는 알고리즘이 필요하다. 장애물을 노란색 공 한 가지로 통일하였으며 그림 2에서 하늘색 선은 공의 움직이는 방향을 나타낸다. 사전 훈련된 물체인식 모델(YOLO v3 [7])을 사용하여 로봇이 환경 속에서 목적지를 향해 주행하는 중 움직이는 물체를 발견하면 이미지를 입력받는다. 그리드로 이미지를 나눈 후 각 그리드에서 예측하여 이를 종합한 바운딩 박스를 생성한다.

더욱 효율적이고 빠른 학습을 위하여 시작 지점을 (0,0)으로 지정하여 로봇이 각 위치에서 어떤 행동을 취하였는지 저장하는 알고리즘을 생성하였다. 로봇이 해당 위치에서 움직이는 물체를 인식하였다는 정보 혹은 해당 위치에서는 어떤 방향으로 움직였다는 정보를 저장하여 다음 에피소드에 같은 위치에 올 때 물체와의 충돌을 방지하고 효율적인 경로를 선택하도록 하였다.

3.3 실험 환경

제안하는 로봇 주행 시스템은 일반적으로 로봇 제어에 사용되는 Robot Operating System (ROS)를 기반으로 하며, 본 논문에서는 Coppelia Robotics의 V-REP 시뮬레이터를 이용해 현실과 비슷한 환경을 시뮬레이션에서 모델링 하였다 [8].

실험 환경은 무작위로 배치된 벽과 주어진 좌표 안에서 움직이는 물체로 구성되어 있고 환경에 대한 정보는 그림 2와 같다. 전체 크기는 가로 15m, 세로 15m이다. 학습 에이전트는 시뮬레이터에 내장된 모바일 로봇 플랫폼인 티틀봇(TurtleBot)이며 에이전트의 시작점은 그림 2내 로봇의 현재 위치이고, 목적지는 별(★) 이미지로 표시가 되어있다.

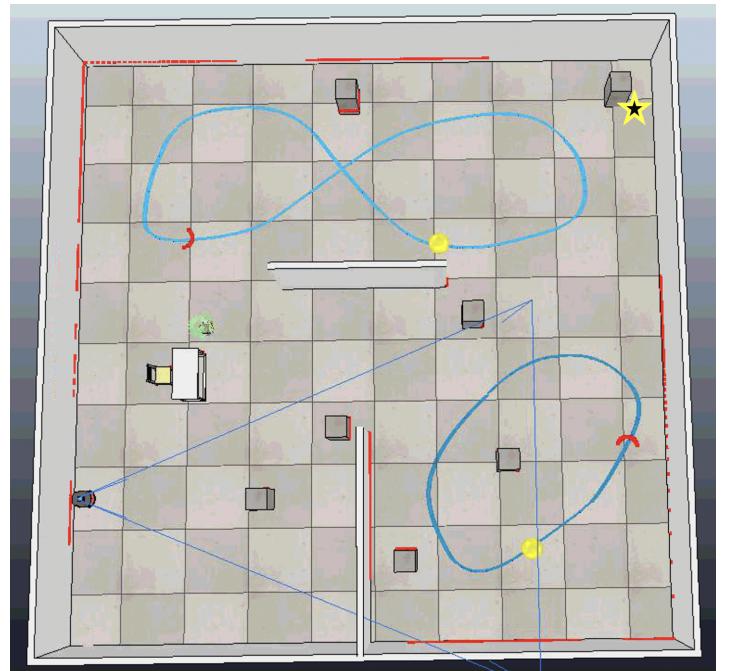


그림 2: 강화학습을 적용한 시뮬레이션 환경. 왼쪽 아래에 위치한 로봇에서 오른쪽을 향해 길게 이어진 두 파란선은 카메라 센서가 인식하는 환경 공간을 의미한다.

1) Code: https://github.com/yeonjisong/deep_rl_pathplanning_avoidance.git

4 실험 결과

알고리즘을 학습한 결과는 표1과 같다. 각 에피소드는 300개의 스텝으로 구성되었으며 총 25000번의 에피소드를 학습한 결과 성공 횟수는 3261번이다. 성공한 에피소드의 평균 스텝은 234, 평균 보상은 10.58로 높은 성공 횟수를 보인다.

알고리즘	성공 횟수	평균 스텝	평균 보상
심층 강화학습	3261	234	10.58

표 1: 알고리즘 학습 결과

그림 3을 살펴보면 0부터 13000번째 에피소드까지는 보상 값이 매우 낮다. 즉, 시작 지점에서 멀리 벗어나지 못한채 장애물과 충돌을 하였거나 300번째 스텝이 되도록 목적지에 도달하지 못한 것이다. 이후 많은 탐색을 통해 얻은 환경에 대한 정보를 이용해, 대략 14000번째 에피소드부터 점차 평균 보상 값이 오르면서 최대 보상 값에 가까워져서 최적의 정책을 찾는다고 볼 수 있다.

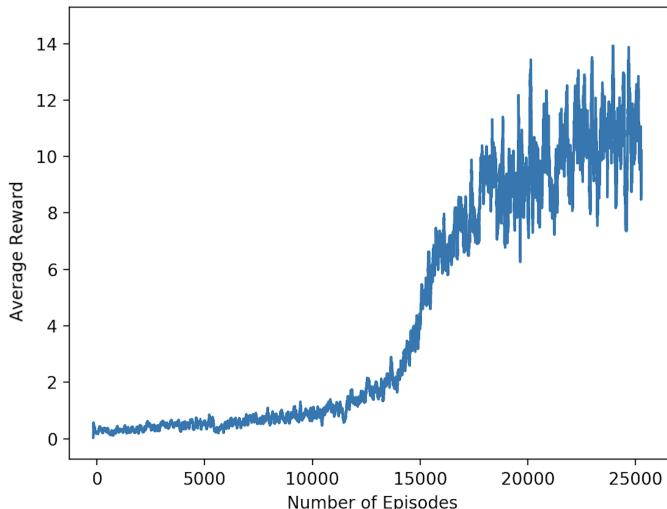


그림 3: 제안하는 알고리즘의 학습결과 에피소드에 따른 평균 보상

그림 4는 로봇이 시뮬레이션 환경에서 목적지를 찾아가는 과정을 나타내는 그림이다. 에피소드를 거듭할수록 별(★) 기호로 표시된 목적지에 가까워지는 것을 나타낸다.

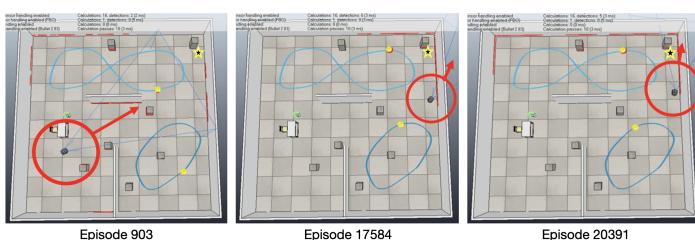


그림 4: 시뮬레이션 환경에서 목적지로 향하는 로봇 주행 예시

5 결론 및 향후 연구

본 논문에서는 환경에 대한 정보가 주어지지 않은 상황에서 이미지와 라이다 센서 데이터, 방향 값을 사용하여 목적지까지 최적의 경로로 탐색하는 자율주행 로봇을 제안하였다. 동일한 환경에서 다양한 종류의 장애물이 추가되거나 다른 목적지를 설정하여 주행하더라도 높은 성공률을 보일 것으로 기대된다.

향후 계획으로는 해당 알고리즘을 실제 환경에서 테스트해보며 불확실하고 다양한 제약이 있는 실제 환경은 시뮬레이션과 어떠한 차이점이 있고 어떻게 해결할 수 있는지 학습해볼 계획이다.

감사의 글

본 연구는 과학기술정보통신부의 재원으로 정보통신기획평가원 (2015-0-00310-SW.StarLab, 2017-0-01772-VTT, 2018-0-00622-RMI, 2019-0-01367-BabyMind)의 지원을 받아 수행되었습니다.

참고 문헌

- [1] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, “Playing Atari with deep reinforcement learning,” *NIPS Deep Learning Workshop*, 2013.
- [2] F. Bounini, D. Gingras, V. Lapointe, and D. Gruyer, “Poster: Real-time simulator of collaborative autonomous vehicles,” in *International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pp. 1991–1994, 2014.
- [3] Y. Wang, I. P. W. Sillitoe, and D. J. Mulvaney, “Mobile robot path planning in dynamic environments,” in *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 71–76, 2007.
- [4] L. Cheng, C. Liu, and B. Yan, “Improved hierarchical a-star algorithm for optimal parking path planning of the large parking lot,” in *IEEE International Conference on Information and Automation (ICIA)*, pp. 695–698, 2014.
- [5] A. C. Egea, S. Howell, M. Knutins, and C. Connaughton, “Assessment of reward functions for reinforcement learning traffic signal control under real-world limitations,” *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2020.
- [6] V. François-Lavet, P. Henderson, R. Islam, M. G. Bellemare, and J. Pineau, “An Introduction to Deep Reinforcement Learning,” *Foundations and Trends® in Machine Learning*, vol. 11, no. 3-4, pp. 219–354, 2018.
- [7] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 779–788, 2016.
- [8] E. Rohmer, S. P. N. Singh, and M. Freese, “V-REP: A versatile and scalable robot simulation framework,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1321–1326, 2013.