

LiDAR를 이용한 End to End 딥러닝 기반의 실내용 자율주행

백승우, 이재복, 김선우
한양대학교 융합전자공학과
{swbaek, ok7393, remero}@hanyang.ac.kr

Indoor Autonomous Driving based on End to End Deep Learning using LiDAR

Seungwoo Baek, Jaebok Lee, and Sunwoo Kim
Department of Electronic Engineering, Hanyang University

요약

본 논문에서는 머신러닝을 이용한 실내용 자율주행 기법을 제안하고 성능을 검증하였다. 제안하는 자율주행은 지도학습 중 실제 주행 데이터로부터 특징을 추출하여 학습하는 End to End 딥러닝을 적용하였다. 머신러닝에 필요한 학습데이터는 LiDAR(light detection and ranging)를 통하여 측정된 거리 정보로 구성하였으며, IMU(inertial measurement unit) 센서를 통해 측정된 각속도를 레이블로 설정하였다. 제안하는 자율주행은 실시간으로 측정된 거리 데이터를 입력받아 각속도를 출력하며, 차량은 추정된 각속도에 따라 주행 방향과 속도를 결정한다. 시뮬레이션을 통해 제안하는 End to End 딥러닝 모델의 추정 성능을 확인하였으며, 주행 실험을 통해 주행 안정성과 충돌 방지 기능을 검증하였다.

I. 서론

자율주행 기술은 비단 자동차에 국한되지 않고, 로봇, 드론, 트랙터 그리고 무인전차 등 다방면으로 확대되어 적용되고 있다. 특히, 사람의 통제로부터 자유로운 완전자율주행의 구현을 위해 복합적인 환경에 유연하게 대처하는 기술에 대한 요구가 높아지고 있다[1]. 기존의 자율주행 기술은 경로설정, 주변인식, 주행계획, 주행 및 피드백의 분절된 네 단계가 개별적이고 순차적으로 이루어져 유연성이 떨어지고 계산 복잡도가 높다.

실제 주행 환경을 반영한 안정적인 자율주행을 위하여 LfD(learning from demonstration) 방법이 제안되었다[2]. 대표적인 LfD 연구로는 Nvidia社에서 제안한 CNN(convolutional neural network) 기반의 End to End Learning[3]과 OpenAI社에서 제안한 역강화학습 기반의 GAIL(generative adversarial imitation learning)이 있다[4]. 본 논문에서는 LiDAR(light detection and ranging)를 이용하여 실내 환경에서의 충돌 방지를 위한 End to End Learning 기법을 제안하였다.

II. 시스템 구성

LfD 기술은 보상함수의 존재에 따라 행동복제(behavior cloning)와 역강화학습(inverse reinforcement learning)으로 나뉜다[5]. 본 논문에서 적용된 End to End 딥러닝 기술은 보상함수가 없는 행동복제 기술에 포함된다. 행동복제 기술은 학습 데이터가 서로 IID(independent and identically distributed) 관계, 즉 서로 독립적이고 동일한 확률분포를 가짐을 가정하는 특징이 있다.

기존의 연구에서는 그림 1-(a)와 같이 충돌방지 기능을 구현하기 위하여 매 시간마다 센서의 FoV(field of view)내의 모든 방향에서 거리를 측정하고, 측정된 거리를 기반으로 방향에 따라 손실함수를 구한다. 방향에 따른 손실함수 집합을 코스트맵(costmap)이라고 하고, 코스트맵 기반의 자율주행은 손실함수를 최소화하는 방향으로 차량을 이동하는 방법을 사용한다. 이전의 자율주행 방법은 매 시간마다 센서의 측위오차 또는 주변 환경의 특성에 따라 손실함수가 최대인 방향이 달라지기 때문에 주행의 안정성이 부족하다[6]. 이와 달리 제안하는 LfD 기반 End to End 딥러닝의 구조는 그림 1-(b)와 같이 실제 운전자의 주행 패턴을 모방하여 학습하였기에 센서 측정값의 노이즈와 주행 경로 상의 돌발 변수에 대하여서 더 강건한 특성을 가질 수 있다[7].

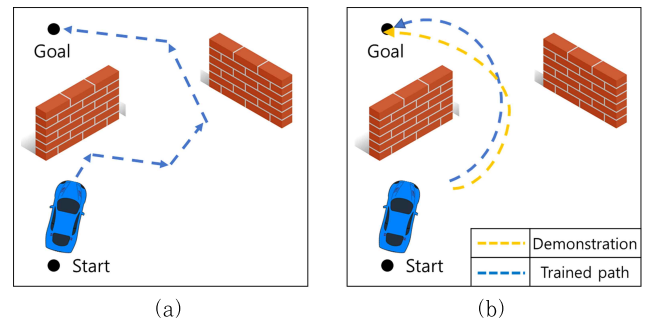


그림 1. (a) 코스트맵 기반 자율주행, (b) 제안하는 LfD 기반 자율주행

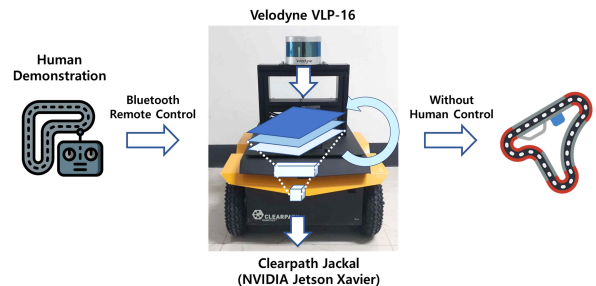


그림 2. 제안하는 자율주행의 구현을 위한 시스템 구성도

본 논문에서는 End to End 딥러닝의 구현을 위하여 그림 2와 같이 직접 블루투스 원격 조종기로 차량을 주행하여 학습 데이터를 형성하였다. 주행 중 LiDAR로 측정된 1차원 거리 정보를 학습 데이터로 설정하였으며, IMU(inertial measurement unit) 센서로 측정된 차량의 각속도를 레이블로 설정하였다. 수집된 LiDAR 거리 데이터셋 중 차량 주행에 직접 영향을 주는 전방의 0~180°의 거리 데이터만을 학습에 이용하였다. 거리를 측정하는 과정에서 장애물의 거리, 재질이나 지형의 구조에 의하여 송신한 레이저가 주어진 시간 내에 수신되지 않을 경우 무한대(inf)로 저장되는 데이터 상의 오류가 발생한다. 따라서 데이터 전처리 과정에서 실험결과에 영향을 주지 않을 정도로 충분히 먼 거리인 20m로 치환하여 다시 저장하였으며, 주행 실험을 진행할 때도 실시간으로 측정된 inf 데이터를 시뮬레이션의 데이터 전처리 과정과 동일한 방식으로 보정해 주었다.

표 1. 시뮬레이션 환경

시뮬레이션 변수		값		
학습 데이터		$1 \times 189 \times 524$		
CPU		Intel I5-7500 (3.4GHz)		
RAM		16 GB		
Name	Input	Output	Stride	No. Kernels
Conv1D	$1 \times 189 \times 524$	$1 \times 189 \times 262$	2	16
ReLU	$1 \times 189 \times 262$	$1 \times 189 \times 262$		
Conv1D	$1 \times 189 \times 262$	$1 \times 189 \times 131$	2	16
ReLU	$1 \times 189 \times 131$	$1 \times 189 \times 131$		
Conv1D	$1 \times 189 \times 131$	$1 \times 189 \times 131$	1	16
ReLU	$1 \times 189 \times 131$	$1 \times 189 \times 131$		
Fully Connected	512	1	1	

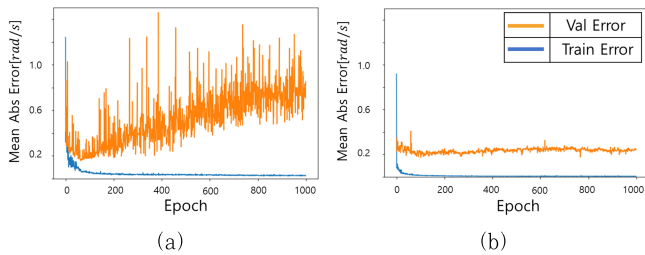


그림 3. (a) DNN의 학습 결과 그래프, (b) CNN의 학습 결과 그래프

III. 시뮬레이션 및 실험 결과

본 논문에서는 차량 주행 학습 시뮬레이션을 위해 표 1과 같은 환경에서 기계학습을 진행하였다. 학습에 필요한 데이터는 0.42m/s의 일정한 선속도로 주행하는 차량에서 수집하였으며, 실제 주행 테스트 시에도 차량의 선속도는 동일하게 유지한다. 제안하는 자율주행에 사용되는 머신러닝 알고리즘을 결정하기 위하여 DNN(dense neural network)과 CNN의 시뮬레이션을 동일한 네트워크 레이어 수, 데이터셋, 시뮬레이션 횟수로 실행하여, 각속도 추정 성능을 그림 3과 같이 비교하였다. LiDAR의 거리 측정 데이터를 입력받아 각속도를 추정하는 DNN 시뮬레이션의 결과 추정 각속도의 MAE(mean absolute error)가 1.14deg/s로 수렴 하였으며, CNN의 학습결과 0.0068deg/s로 수렴하였다. CNN을 사용 하였을 때, DNN에 비하여 더 낮은 각속도의 MAE를 달성하는 우수한 추정 성능을 나타내었으며, 검증 데이터셋이 발산한 DNN과 달리 CNN에서는 검증 데이터셋과 훈련 데이터셋의 학습 곡선이 모두 수렴하여 적은 수의 데이터셋에서도 수렴하는 성질을 나타내었다. CNN에서 검증 데이터셋이 수렴한 이유는 CNN을 이용한 방법이 데이터를 전부 활용하는 것이 아닌 컨볼루션 연산을 이용해 데이터셋의 공통된 특징을 추출하는 방법을 사용하기 때문에 DNN에 비하여 강건한 특성을 나타낸 것으로 분석된다. 차량 주행 실험은 한양대학교 IT/BT관 3층 복도에서 진행되었다. 주어진 환경에서 안정적으로 주변 환경과 충돌 없이 주행하였으며, 주행한 장소의 지도와 실시간으로 측정된 좌측, 전방, 우측 벽으로부터의 거리는 그림 4에 표시하였다. 제안하는 자율주행은 표 2와 같이 기존의 자율주행 시스템에 비하여 더 적은 데이터셋을 이용함과 동시에 우수한 주행 안정도를 달성하였다. 실내 환경에서는 차량 주행 시 양쪽 벽이 항상 존재하기에 카메라의 RGB 데이터나 LiDAR의 포인트클라우드를 이용한 방법에 비하여 제안된 방법처럼 벽과의 거리를 이용하는 방식이 더욱 효과적으로 자율주행을 구현한 것으로 분석된다. 주행 안정도는 [8]에서 제시된 바에 따라 직선 주행 시 차량의 오실레이션을 이용하였으며, 벽으로부터 거리의 표준편차를 계산함으로써 도출할 수 있다.

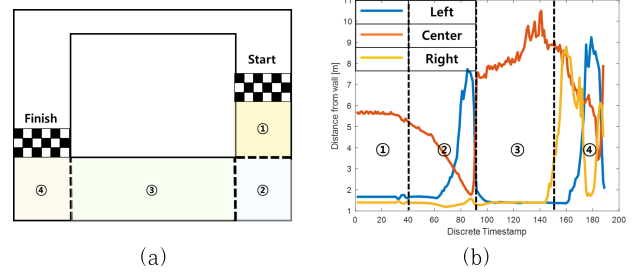


그림 4. (a) 주행 테스트 맵, (b) 주행 테스트 결과

표 2. 주행 성능 비교

	학습 데이터셋 수	최대 주행 안정도[m]
N. Patel et al. [8]	14456	0.0575
Proposed	189	0.0599

IV. 결론

본 논문에서는 실내 자율주행 차량 또는 로봇에 적용될 수 있는 End to End Learning 기법을 제안하였으며, 약 0.06m의 낮은 오실레이션으로 우수한 주행 안정도를 달성함과 더불어 학습에 필요한 데이터셋의 양을 기존에 비하여 98.6% 낮출 수 있음을 실제 주행 실험을 통하여 검증하였다. 본 연구는 실내용 차량 또는 로봇뿐 아니라 제한된 경로 내에서, 전문 운전사의 주행 패턴을 학습할 수 있는 대중교통의 자율주행에 활용될 수 있다.

ACKNOWLEDGMENT

본 연구는 과학기술정보통신부 및 정보통신기획평가원의 대학ICT연구센터육성지원사업의 연구결과로 수행되었음(IITP-2021-0-01637)

참고 문헌

- [1] B. Paden, M. Čáp et al., "A survey of motion planning and control techniques for self-driving urban vehicles," *IEEE Trans. Intell. Veh.*, vol. 1, no. 1, pp. 33-55, Mar. 2016.
- [2] B. D. Argall, S. Chernova et al., "A survey of robot learning from demonstration," *Rob. Auton. Syst.*, vol. 57, no. 5, pp. 469-483, May. 2009.
- [3] M. Bojarski, M. Monfort et al., "End to end learning for self-driving cars," *arXiv preprint arXiv: 1604.07316*, Apr. 2016.
- [4] J. Ho and S. Ermon, "Generative adversarial imitation learning," *Adv. Neural Inf. Process. Syst.*, Dec. 2016, pp. 4565-4573.
- [5] S. Choi, K. Lee and S. Oh, "Robust learning from demonstrations with mixed qualities using leveraged gaussian processes," *IEEE Trans. Robot.*, vol. 35, no. 3, pp. 564-576, Jun. 2019.
- [6] A. Ng and S. Russell, "Algorithms for inverse reinforcement learning", *Proceeding of 17th International Conference on Machine Learning*, Jun. 2000, pp. 663-670.
- [7] Silver, D., Bagnell, J.A., Stentz et al., "A learning from demonstration for autonomous navigation in complex unstructured terrain," *Int. J. Robot. Res.* vol. 29, no. 5, pp. 1565-1592, Jun. 2010.
- [8] N. Patel, A. Choromanska et al., "Sensor modality fusion with cnns for ugv autonomous driving in indoor environments," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sep. 2017, pp. 1531-1536.