Journal of Korea Robotics Society (2024) 19(1):130-138
https://doi.org/10.7746/jkros.2024.19.1.130 ISSN: 1975-6291 / eISSN: 2287-3961

Moving Object Segmentation을 활용한 자동차 이동 방향 추정 성능 개선

Moving Object Segmentation-based Approach for Improving Car Heading Angle Estimation

노 치 윤[1] · 정 상 우[2] · 김 유 진[2] · 이 경 수[3] · 김 아 영†
Chiyun Noh[1], Sangwoo Jung[2], Yujin Kim[2], Kyongsu Yi[3], Ayoung Kim†

Abstract: High-precision 3D Object Detection is a crucial component within autonomous driving systems, with far-reaching implications for subsequent tasks like multi-object tracking and path planning. In this paper, we propose a novel approach designed to enhance the performance of 3D Object Detection, especially in heading angle estimation by employing a moving object segmentation technique. Our method starts with extracting point-wise moving labels via a process of moving object segmentation. Subsequently, these labels are integrated into the LiDAR Pointcloud data and integrated data is used as inputs for 3D Object Detection. We conducted an extensive evaluation of our approach using the KITTI-road dataset and achieved notably superior performance, particularly in terms of AOS, a pivotal metric for assessing the precision of 3D Object Detection. Our findings not only underscore the positive impact of our proposed method on the advancement of detection performance in lidar-based 3D Object Detection methods, but also suggest substantial potential in augmenting the overall perception task capabilities of autonomous driving systems.

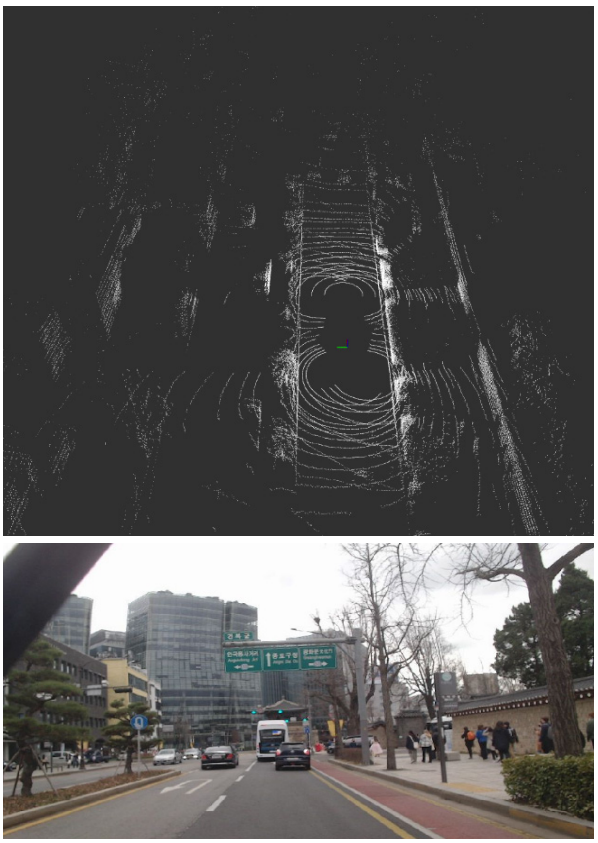
Keywords: 3D Object Detection, Moving Object Segmentation, Pointcloud, Autonomous Driving

1. 서 론

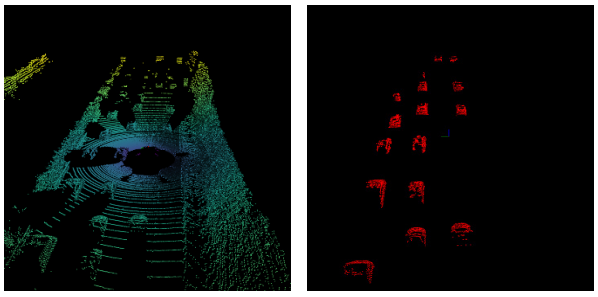
물체를 정확하게 탐지하는 능력은 자율주행 분야에서 핵심적인 역할을 수행한다. 이는 다중 물체 추적 및 경로 계획과 같은 다양한 하위 작업의 기반이 되며, 자율 주행 차량의 안전성과 효율성을 보장하기 위한 필수적인 구성 요소이다. 특히 물체 탐지를 통한 다른 차량의 heading angle 추정은 자율 주행 차량이 주변 교통 환경을 예측하고 상호작용하기 위한 핵심적인 작업이다. 다른 차량의 heading angle을 추정함으로써, 자신의 운전 환경의 변화에 대응하고, 도로 상황을 실시간으로 이해하며, 안전한 주행 경로를 선택하고 제어할 수 있다.

물체 탐지를 위해서 다양한 센서들이 사용되고 있으며 대표적인 센서로는 카메라와 라이다(LiDAR)가 있다[1]. 카메라는 물체의 표면에서 반사되는 빛을 감지함으로써 사진을 만들어내는 장비이다. 대게 가격이 저렴하며, 주변 환경을 고해상도로 인지할 수 있다는 장점이 존재한다. 하지만 2차원의 정보만 담고 있기 때문에 물체와의 거리 및 속도를 측정할 수 없다는 단점이 존재한다. 라이다는 광 펄스의 송신과 수신을 이용해 물체와의 거리를 추정하고, 이 정보를 기반해 포인트 클라우드를 생성하는 장비이다. 다른 센서들에 비해 인지 거리가 길고, 거리를 측정하여 3차원 좌표를 얻을 수 있으며, 분해능이 좋다는 장점이 존재해 라이다 센서의 측정 결과를 이용해 3차원에서의 물체 탐지를 하는 방법들이 많이 연구되어지고 있다[2,3]. 도심에서의 포인트 클라우드 샘플은 [Fig. 1] 과 같다. 사

Received : Oct. 30. 2023; Revised : Dec. 11. 2023; Accepted : Dec. 18. 2023
※ This research was conducted with the support of the 'National R&D Project for Smart Construction Technology (23SMIP-A158708-04)' funded by the Korea Agency for Infrastructure Technology Advancement under the Ministry of Land, Infrastructure and Transport, and managed by the Korea Expressway Corporation
1. Master Student, Mechanical Engineering, Seoul National University, Seoul, Korea (gch06208@snu.ac.kr)
2. Ph. D. Student, Mechanical Engineering, Seoul National University, Seoul, Korea (dan0130, zxc123xc@snu.ac.kr)
3. Professor, Mechanical Engineering, Seoul National University, Seoul, Korea (kyi@snu.ac.kr)
† Associate Professor, Corresponding author: Mechanical Engineering, Seoul National University, Seoul, Korea (ayoungk@snu.ac.kr)



[Fig. 1] Pointcloud (up) and image (down) in the urban environment



[Fig. 2] Raw Pointcloud (left) and moving objects' Pointclouds segmented using MOS (right)

진에서 볼 수 있듯, 가로수나 빌딩 등 도심 환경 장애물들이 많이 존재하고 이는 포인트 클라우드 상에서 노이즈로 작용하여 3차원 물체 탐지에서 오 탐지 등 여러 악영향을 미칠 수 있다.

Moving Object Segmentation (MOS)란 라이다 센서의 측정결과인 포인트 클라우드에서 동적인 점과 정적인 점을 분할하는 알고리즘을 말한다. 포인트 클라우드가 MOS를 거치게 되면 [Fig. 2]와 같이 정적인 점들과 동적인 점들로 분할되고, 이러한 정보는 도심 환경 장애물을 구분하고 도심 환경에서 3차원 물체를 탐지하는데 큰 도움을 준다.

Segmentation 정보를 이용해 물체 탐지 성능을 높인 알고리즘으로는 PointPainting[4]이 있다. PointPainting은 image segmentation 정보를 Pointcloud에 사영시키는 방법을 사용하였다. 하지만 image 내에서는 깊이 추정이 어렵고, occlusion으로 인해 segmentation이 되지 않는 영역이 발생할 수 있다. 또한 카메라와 라이다 사이 calibration 오차로 image segmentation 정보가 잘못된 Pointcloud 위치에 사영되었을 경우 오히려 성능을 저하시키는 원인이 될 수도 있다.

따라서 본 논문은 카메라와의 퓨전이 아닌 라이다 기반 MOS를 통해 포인트 클라우드를 재 가공함으로써 3차원 물체의 heading angle 추정 성능을 높이기 위한 방법을 제안한다.

알고리즘은 크게 MOS와 3차원 물체 탐지, 두 단계로 나누어서 진행된다. 먼저 라이다 센서의 측정결과인 포인트 클라우드를 MOS를 통해 각 점마다 'moving' 혹은 'static'의 라벨을 얻는다. 라벨 정보를 포인트 클라우드에 하나의 채널을 추가하고, 이를 입력 값으로 하여 3차원 물체 탐지를 진행한다.

본 논문의 알고리즘을 설명하기 위해 2장에서는 MOS와 3차원 물체 탐지에 관련한 기존 연구들을 소개하고, 3장에서는 알고리즘의 구성 및 방법을 제시한다. 4장에서는 학습 시 사용한 데이터셋과 학습 모듈에 대한 설명, 그리고 알고리즘 실험 결과 및 분석을 다루고, 5장에서는 논문의 결론과 추후 연구 방향에 대해 기술한다. 본 논문에서 제시하는 기여점은 다음과 같다.

- MOS 결과로 얻은 점 단위의 라벨을 포인트 클라우드에 추가하여 3D Object Detection 알고리즘의 heading angle 추정 성능을 향상시키는 새로운 방법론을 제안하였다.
- MOS 성능 향상에 따른 3D Object Detection 알고리즘의 heading angle 추정 성능 향상 가능성을 검증하였으며, 추후 MOS 알고리즘 성능 향상 연구의 필요성에 대해 시사하였다.
- KITTI-road 공개 데이터셋과 SemanticKITTI 공개 데이터셋을 이용하여 알고리즘의 heading angle 추정 성능을 측정하였고, 성능이 향상됨을 검증하였다.

2. 선행 연구 조사

2.1 Moving Object Segmentation (MOS)

포인트 클라우드에서 움직이는 물체를 식별하는 것은 SLAM (Simultaneous localization and mapping)이나 물체의 이동 경로 예측, 경로 계획 등 자율주행에서 라이다를 이용한 하위 작업들의 성능을 결정하는 중요한 요소이다. 따라서 강건한 자율주행을 위해서 정적인 물체와 동적인 물체를 구분하는 일은 매우 중

요하다. 이에 포인트 클라우드의 각 점을 정적인 점과 동적인 점으로 구분하는 MOS 관련 여러 연구들이 수행되어지고 있다. 기존 MOS 방법은 크게 사영 기반(projection-based) 방법론과 비사영 기반(non-projection-based) 방법론으로 나뉘어진다.

2.1.1 사영 기반 방법론

사영 기반 방법론은 3차원 포인트 클라우드를 2차원 이미지로 사영하고, 이를 이용해 MOS를 진행하는 방법을 말한다. 3차원 포인트 클라우드의 경우 다른 센서 데이터들에 비해 희박한 정보를 가지고 있고, convolution 연산을 진행 시 많은 연산량을 필요로 한다. 이에 많은 연구들은 포인트 클라우드를 2차원 이미지로 사영하여 2차원 convolution 연산을 통해 spatio-temporal feature를 추출하고, 이를 다시 3차원 포인트 클라우드로 역사영하는 방법으로 한계를 해결하고자 하였다. X. Chen et al.은 range image를 이용하여 MOS를 수행하는 LMNet[5]을 제안하였으며, P.Wu et al.는 Bird's Eye View 이미지를 이용하여 MOS 뿐만 아니라 미래 경로까지 예측하는 MotionNet[6]을 제안하였다.

하지만 사영 기반 방법론의 경우 사영 과정에서 생기는 정보 손실로 인해 성능이 떨어지고, 학습시키지 않은 새로운 환경 혹은 다른 센서 모델을 사용할 경우에는 성능이 떨어지는 여러 한계점들이 존재하였다.

2.1.2 비사영 기반 방법론

비사영 기반 방법론은 사영 시 발생하는 여러 문제점들을 해결하기 위해 포인트 클라우드 자체를 MOS의 입력 값으로 사용하는 방법론을 말한다. 포인트 클라우드는 희박한 데이터이기 때문에 sparse convolution을 사용하여 시간 및 메모리 측면에서 효율적으로 spatio-temporal feature를 추출하고, 이를 통해 포인트 별로 정적 혹은 동적인 점인지를 판단한다. 라이다 포인트 클라우드는 카메라 이미지와는 달리 intensity보다는 3차원 위치 정보에 치중하므로, 다른 환경 및 다른 센서 모델을 사용하여도 사영 기반 방법론보다 강건한 성능 결과를 보여주고 있다. B. Mersch et al.는 Binary Bayes Filter와 Receding horizon strategy를 이용하는 방법인 4DMOS[7]를 제안하였고, N. Wang et al.는 물체 탐지 결과를 MOS에 사용하는 InsMOS[8]를 제안하였다.

본 논문에서는 라이다만을 이용한 비사영 기반 MOS 중, 다른 환경에서도 강건한 성능을 유지하는 4DMOS 알고리즘을 차용하여 MOS를 진행하였다.

2.2 라이다 포인트 클라우드에서의 3D Object Detection

물체를 탐지하는 능력은 여러 하위 작업을 위한 핵심 역할을 수행한다. [9, 10]에서는 라이다와 카메라 등 여러 센서들을 이용해 물체 탐지를 기반으로 주차 공간을 검출하는 알고리즘에 대해 소개하였으며, [11]에서는 복잡한 환경에서 image와 k-nearest clustering 알고리즘을 이용하여 물체를 탐지하고 물체의 pose를 인식하여 알맞은 파지 방법을 찾는 알고리즘을 소개하였다. 이처럼 3차원에서 물체 탐지를 기반으로 다양한 하위 작업을 수행하는 여러 연구들이 진행 중이고, 그 중 많은 연구들이 라이다 포인트 클라우드를 물체 탐지를 위해 사용하고 있다.

라이다 포인트 클라우드는 x, y, z 좌표를 가지는 3차원 공간이기 때문에 물체 위치를 정확하게 파악하기 위해서는 3차원 정보를 처리해야 하고, 이에 3차원 물체 탐지를 위해서는 3차원 convolution network를 사용해야 한다. 하지만 이 경우 많은 연산량을 요구하기 때문에 실행 속도가 느리다. 이를 해결하고자 MOS의 사영 기반 방법론과 같이 Bird's Eye View 평면이나 이미지 평면으로 사영시키는 방법들이 연구되고 있다[12,13]. 이후 포인트 클라우드에서 특징을 추출하는 PointNet[14]의 등장으로 이를 이용하여 3차원 물체 탐지에 사용하는 VoxelNet[2]과 PointPillars[3]가 제안되었다. VoxelNet은 x, y, z축마다 일정 크기를 가지는 voxel로, PointPillars는 x, y축마다 일정 크기를 가지고 z축으로는 한계가 없는 pillar로 포인트 클라우드를 쪼개고, 각 voxel, pillar마다 CNN을 이용해 특징을 추출하여 3D Object Detection을 수행하는 알고리즘이다. 이후 VoxelNet의 실행 속도를 개선한 SECOND[15]가 등장하였으나 3D convolution을 사용하는 특성상 속도 개선 측면에서 한계가 존재하였다.

PointPillars는 VoxelNet과 달리 z축으로 한계가 없는 pillar를 사용하기 때문에 3D가 아닌 2D feature map을 얻어낼 수 있고, 이를 통해 2D convolution 연산을 진행할 수 있어 빠른 연산이 가능하다는 장점이 있다. 이에 본 논문에서는 PointPillars를 3D Object Detection 알고리즘으로 선정하여 사용하였다.

라이다 포인트 클라우드 만을 이용하지 않고 다른 센서에서 얻은 데이터와 함께 사용하여 성능을 높이려고 시도한 연구들 또한 존재한다. S. Vora et al.는 라이다와 다른 센서 정보를 함께 사용할 경우 성능이 감소되는 원인이 viewpoint misalignment라 주장하며, 새로운 센서 퓨전 방법인 PointPainting[4]을 제시하였다. PointPainting은 카메라로 찍은 이미지에 semantic segmentation을 진행하고, 그 결과를 라이다 포인트 클라우드에 사영시켜, 점마다 클래스 정보를 추가한다. 이후, 채널이 늘어난 포인트 클라우드를 기존 라이다 기반 3차원 물체 탐지 알고리즘의 입력 값으로 사용하여 성능을 높이는 알고리즘이다.

이 방법에 착안하여 본 논문에서는 PointPainting 알고리즘의 image segmentation 대신, 라이다 포인트 클라우드의 MOS 결과를 포인트 클라우드에 추가하여 3차원 물체 탐지 성능을 개선하는 방안에 대한 연구를 진행하였다.

3. 동적 물체 분할을 통한 3차원 물체 heading angle 추정 성능 개선 알고리즘

본 논문의 알고리즘 구성은 [Fig. 3]과 같으며 heading angle 추정 성능을 높이기 위해 Pointcloud를 segmentation을 통해 전처리를 하고 이를 3차원 물체 탐지 알고리즘의 input으로 사용하였다. Pointcloud segmentation은 자율주행 상황에서 알고리즘의 실행 속도를 고려하여 3D semantic segmentation이 아닌 MOS를 이용하였다. 또한, MOS를 진행한 Pointcloud의 경우 움직이는 물체들만 분류가 되고, 이를 잇달아 계속적으로 관찰한다면 물체의 움직임 정보를 취득할 수 있다. 이를 이유로 [Fig. 3]과 같이 heading angle 추정 성능을 높이기 위한 알고리즘을 구성하였고, 이는 크게 두 단계로 나뉘어진다.

첫 번째 단계는 MOS 단계이다. MOS단계에서는 B. Mersch et al.가 제안한 4DMOS 모델을 이용하였다. 4DMOS 모델은 비사영 기반 MOS 방법으로 특정 개수의 포인트 클라우드 시퀀스를 가지고 실시간으로 MOS를 진행하는 알고리즘이다. 두 번째 단계는 3차원 물체 탐지 단계이다. 물체 탐지 단계에서는 A.H. Lang et al.가 제안한 PointPillars 모델을 사용하였다. 3.1에서는 4DMOS 모델에 대해, 3.2에서는 PointPillars 모델에 대해 기술하였다.

3.1 4DMOS

4DMOS과정은 크게 4단계로 구분된다. 첫 번째 단계는 라이다의 측정 결과 얻어진 n개의 포인트 클라우드 시퀀스를 현재의 포인트 클라우드 좌표계로 좌표 변환하고, 이를 병합하는 단계이다. 이 때 각 포인트 클라우드 별 좌표계 변환 행렬의 경우 wheel encoder나 IMU 같은 센서로 얻을 수 있다. 그 다음 단계는 병합된 포인트 클라우드를 4D sparse covolution을 통해 spatio-temporal feature를 뽑아내는 단계이다.

세 번째 단계는 Receding horizon strategy와 Binary Bayes Filter를 이용하여, 점마다 부여된 동적 예측 점수를 재귀적으로 정제하는 단계이다. Receding horizon strategy란, 라이다에서 새롭게 포인트 클라우드를 얻을 경우 직전에 계산했던 포인트 클라우드 시퀀스에서 가장 오래된 포인트 클라우드를 삭제하고, 새로 들어온 포인트 클라우드를 추가함으로써 새로운 시퀀스를 만드는 것을 의미한다. Receding horizon strategy를 취하게 될 경우 한 개의 포인트 클라우드는 새롭게 시퀀스에 추가된 이후로 삭제되기 전까지 총 시퀀스의 길이만큼, 즉 n번의 MOS 추정 작업을 거치게 된다. 새로운 센서 입력 값이 들어올 경우 한 개의 포인트 클라우드는 지금까지 누적된 센서 입력 값과 새로운 센서 입력 값을 가지고 moving state를 추정하게 된다.

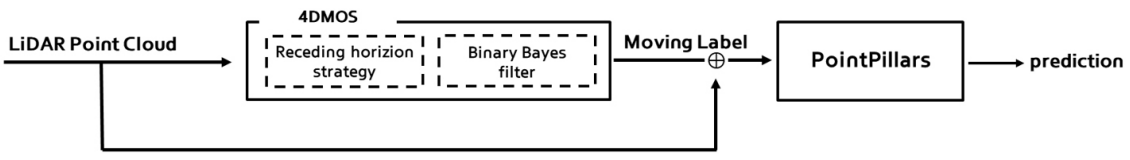
t초에서 들어온 포인트 클라우드를 S_0, 포인트 클라우드 시퀀스를 $z_t = \{S_0\, S_1 \cdots, S_{n-1}\}$, 추정하고자 하는 포인트 클라우드 S_j에 속한 점 p_i에서의 동적 상태를 $m_i^j \in \{0,1\}$이라고 할 때, 현재까지 누적된 시퀀스를 기반으로 점 p_i의 동적 상태에 대한 확률 식은 $p(m_i^j | z_{0:t})$와 같이 나타낼 수 있다. 이 확률 식에 Binary Bayes Filter를 적용시킬 경우 식 (1)과 같이 나타낼 수 있다.

$$l(m_i^j \mid z_{0:t}) = l(m_i^j \mid z_{0:t-1}) + l(m_i^j \mid z_t) - l(m_i^j) \qquad (1)$$
$$where\ l(x) = \log \frac{p(x)}{1-p(x)}$$

식 (1)과 같이 재귀적으로 새로운 포인트 클라우드가 들어올 때마다 누적된 시퀀스를 통해 동적 확률을 계산해 나가고, 이를 최종적으로 S_j가 시퀀스에서 삭제될 때까지 반복함으로써 최종 확률을 얻는다. 최종 확률 $p(m_i^j \mid z_{0:t})$이 0.5를 넘을 경우에만 동적 상태라고 판단하고 분할한다. 본 논문에서 시퀀스에 포함되는 포인트 클라우드 개수는 10개, 포인트 클라우드 간 시간 차이는 0.1초로 설정하였다.

3.2 PointPillars

희박한 3차원 포인트 클라우드에서 사영 없이 정보를 추출하기에는 많은 연산량과 시간이 필요하다. 이에 Y. Zhou et al.는 x, y, z축마다 일정 크기를 가지는 voxel을 만들어, 연산량 측면에서 생기는 문제를 해결한 VoxelNet을 제안하였다. 그럼에도 실시간 탐지를 하기에는 여전히 느린 속도를 보였고, 이에 voxel과는 달리 x, y축 방향으로만 일정 그리드로 나누고, z축으로 한계를 두지 않는 기둥(pillar)를 만들어 더욱 빠른 연산을 진행하도록 만든 알고리즘인 PointPillars가 제안되었다. PointPillars는 크게 feature encoder network, 2D convolution



[Fig. 3] Overview of our algorithm

backbone, detection head 총 3 단계로 나누어 진행된다.

Feature encoder network는 포인트 클라우드를 pseudo image로 바꾸는 단계이다. 포인트 클라우드를 pillar로 나누고, 각 포인트를 $(x,y,z,r,x_c,y_c,z_c,x_p,y_p)$의 9차원($D$) vector로 구성한다. 이 때 x_c,y_c,z_c는 pillar 내부 점들의 arithmetic mean을 의미하고, x_p,y_p는 pillar 중심점을 의미한다. 그 후 각 pillar마다 N개의 점을 선택하고, P개의 pillar들을 모아 (D,P,N) 사이즈의 tensor로 encoding 한다. encoding된 tensor는 PointNet을 거쳐 (C,P,N) 사이즈의 tensor로 변환되고, 채널 방향으로 max pooling을 함으로써 (C,P)의 tensor로 변환된다. 각 pillar별 feature를 원래의 pillar 위치로 되돌림으로써 (C,H,W) 사이즈의 pseudo image를 생성한다.

생성된 pseudo image는 2D convolution backbone의 input으로 사용된다. Pseudo image는 순서대로 3개의 2D convolution block을 거치고, 각각의 결과는 upsampling 되어 하나의 tensor로 합쳐지게 된다. 마지막으로 2D convolution backbone의 output은 Single Shot Detector (SSD)[16]를 거쳐 3D Object Detection을 수행하게 된다.

PointPillars의 loss function은 크게 localization loss (L_{loc}), direction loss (L_{dir}), object classification loss (L_{cls})로 나누어 계산한 후 total loss (L_{total})를 계산한다. localization loss는 bounding box를 나타내는 7개의 숫자 (x,y,z,w,l,h,θ)에 대해 식 (2)와 같이 residual을 계산한 후 식 (3)과 같이 SmoothL1 loss를 이용하여 계산한다.

$$\Delta x = \frac{x^{gt}-x^a}{d^a}, \Delta y = \frac{y^{gt}-y^a}{d^a}, \Delta z = \frac{z^{gt}-z^a}{h^a}, \quad (2)$$
$$\Delta w = \log\frac{w^{gt}}{w^a}, \Delta l = \log\frac{l^{gt}}{l^a}, \Delta h = \log\frac{h^{gt}}{h^a},$$
$$\Delta\theta = sin(\theta^{gt}-\theta^a),$$
$$\text{where } x^{gt}: \text{ground truth}, x^a: \text{anchor box},$$
$$d^a = \sqrt{(w^a)^2+(l^a)^2}$$

$$L_{loc} = \Sigma_{b\in(x,y,z,w,l,h,\theta)} SmoothL1(\Delta b) \quad (3)$$

식 (2)에서 $\Delta\theta$를 구하는 방식으로 계산을 할 경우 θ^{gt}와 θ^a의 차이가 180도일 경우를 구분하지 못하므로, softmax classification loss를 사용하여 L_{dir}를 계산하고, Object classification loss의 경우 focal loss[17]를 사용하여 L_{cls}을 계산한다. 계산된 3개의 loss들은 식 (4)와 같이 합쳐져 total loss를 계산하게 된다.

$$L_{total} = \frac{1}{N_{pos}}(\beta_{loc}L_{loc}+\beta_{cls}L_{cls}+\beta_{dir}L_{dir})$$
$$\text{where } N_{pos}: \text{number of positive anchor box}. \quad (4)$$

[Table 1] PointPillars setup

Parameter	Value
xy resolution (m)	0.16
max number of pillars	16000
max number of points per pillar	32

[Table 2] Anchor box setup for region proposal network

Parameter	Value
Pointcloud range $(x_{min},y_{min},z_{min},x_{max},y_{max},z_{max})$	[0, -39.68, -3, 69.12, 39.68, 1]
Anchor size (x,y,z)	[3.9, 1.6, 1.56]
Anchor rotations (rad)	0~1.57
Matched threshold	0.6

본 논문에서 사용한 PointPillars의 setup은 [Table 1]과 같다. 또한 region proposal network를 위한 자동차 클래스 anchor box setup은 [Table 2]와 같이 설정하였다.

4. 실험 및 결과

4.1 Dataset Description

본 논문의 알고리즘 학습 및 평가를 위해서 라이다 포인트 클라우드의 각 점 별 동적, 정적 라벨링 및 물체의 클래스에 대한 라벨링이 달린 3차원 bounding box 정보, odometry 정보를 가진 데이터셋 이 필요하다. 이에 많은 MOS 연구에 사용되고 있는 KITTI-road dataset[18]과 SemanticKITTI dataset[19]을 선정하여 학습과 평가에 사용하였다.

4.1.1 KITTI-road Dataset

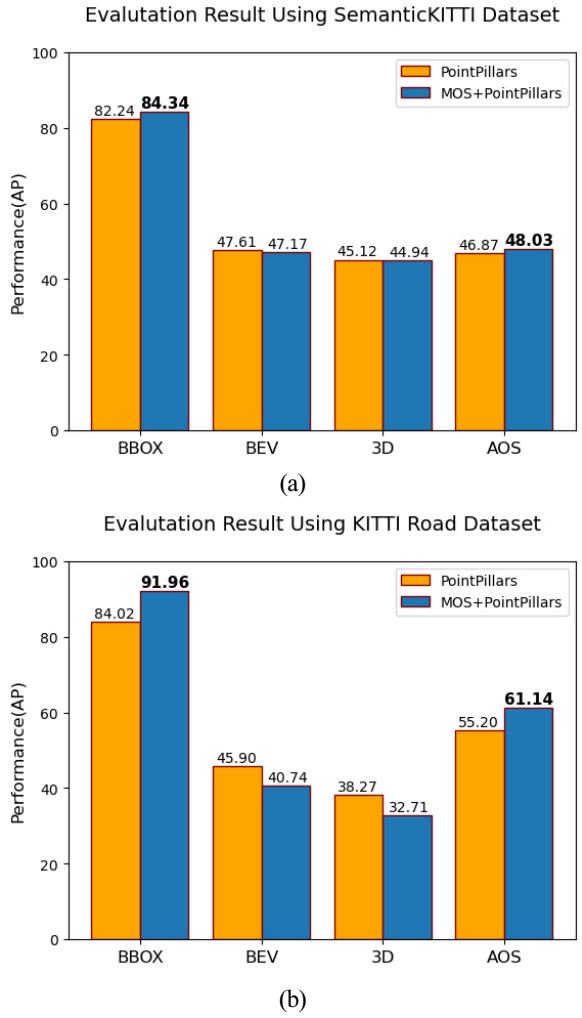
KITTI-road dataset에서는 라이다를 통해 얻은 포인트 클라우드와 odometry 정보를 12개의 scene (30~41)에 걸쳐 제공한다. 본 논문에서는 MotionSeg3D[20]에서 Auto-MOS[21]를 이용해 제작하여 배포한 KITTI-road 데이터셋의 MOS ground truth 라벨과, InsMOS에서 Euclidean 클러스터링 방법을 이용해 만들어진 클러스터를 에워싸는 방법으로 배포한 3D bounding box ground truth를 이용하여 학습과 평가에 사용하였다. 알고리즘의 학습에는 총 4958개의 포인트 클라우드로 구성된 scene 30~40을 사용하였고, 평가에는 838개의 포인트 클라우드로 구성된 scene 41를 이용하여 평가를 진행하였다.

4.1.2 SemanticKITTI Dataset

SemanticKITTI dataset에서는 MOS ground truth 라벨을 제

공하지만 3D bounding box는 제공하지 않는다. 이에 마찬가지로 InsMOS에서 배포한 3D bounding box ground truth를 이용하여 학습과 평가에 사용하였다. scene 0~7, 9~10을 알고리즘의 학습에 사용하였고, scene 8은 알고리즘의 평가를 위해 사용하였다.

알고리즘은 KITTI 평가 지표[22]를 사용하여 3차원 물체 탐지 성능을 평가하였다. 추가적으로 물체의 이동 경로 예측, path planning 등 물체 탐지 결과를 이용하는 하위 작업의 성능을 높이는데 물체의 heading angle을 예측하는 것이 중요하다 생각하여, AOS (Average Orientation Similarity)를 가장 중요한 지표로 설정하여 알고리즘을 평가하였다. 데이터셋 내 자동차 클래스가 KITTI-road dataset에서는 93%, SemanticKITTI dataset에서는 81%를 차지할 정도로 한 개의 클래스에 대해 치중되어 있어 평가 및 학습은 오로지 자동차 클래스만을 이용하여 진행하였다.



[Fig. 4] (a) Performance of the proposed algorithm using KITTI-road dataset and (b) using SemanticKITTI dataset

4.2 알고리즘 성능 평가

4DMOS의 경우 pretrained 된 모델을 사용하였다. MOS 결과 각 포인트 별 'static', 'moving' 2개의 클래스로 나누어진 라벨을 얻을 수 있고, 이를 각각 순서대로 1, 2로 라벨링한다. 4채널로 이루어진 포인트 클라우드의 뒤에 MOS class를 추가하여, $[x,y,z,reflectivity,class]$로 이루어진 5채널 포인트 클라우드를 생성하고, PointPillars의 입력 값으로 사용하여 학습을 진행하였다. 알고리즘 학습은 KITTI-road 데이터셋의 경우 80 epoch까지 진행하였고, SementicKITTI 데이터셋의 경우 120 epoch까지 진행하였다. KITTI 평가 지표를 사용하여 평가한 결과는 [Fig. 4]와 같고 각각 지표에 대한 설명은 [Table 3]와 같다[22].

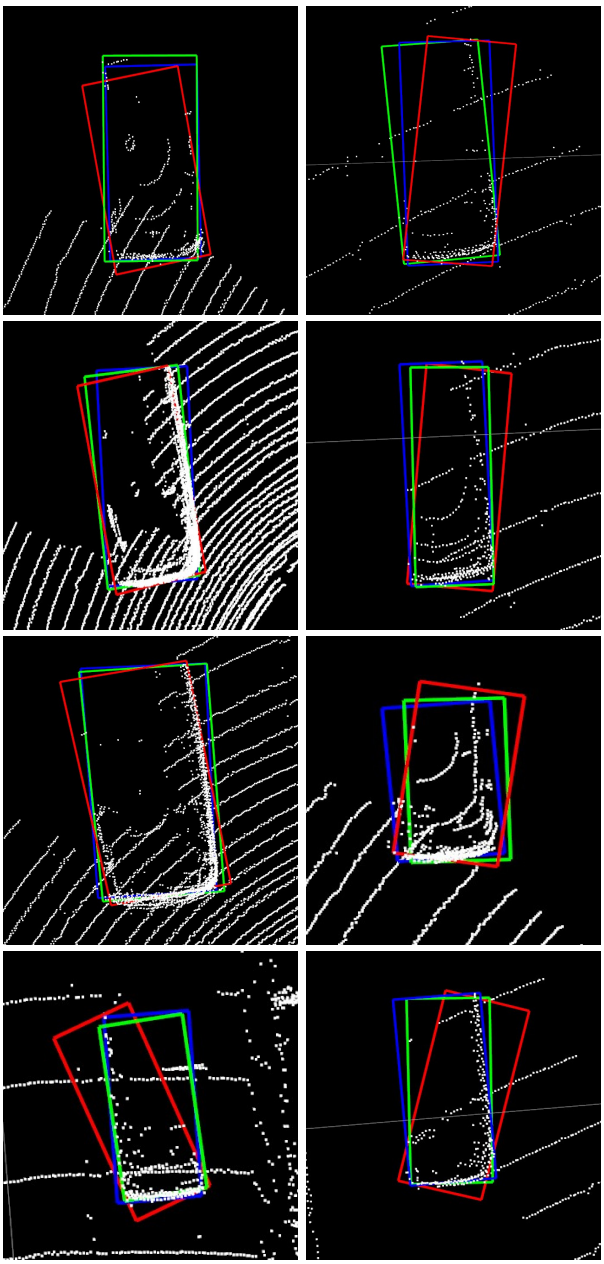
본 논문에서는 각 지표 별로 bounding box를 TP로 판단하는 IoU 임계값을 BBOX는 0.7, BEV와 3D는 0.5로 설정하여 average precision을 계산하였다. 평가 결과 BBOX와 AOS 지표 측면에서는 본 논문에서 제안한 알고리즘의 성능이 PointPillars에 비해 더 높은 수치로 나타났지만, BEV와 3D지표 측면에서는 더 낮은 수치로 나타났다.

3D bounding box의 heading angle에 따라 이미지 평면으로 사영되는 2D bounding box의 크기가 달라지므로, BBOX 지표는 heading angle에 높은 dependency가 존재한다. 따라서 이미지 평면으로 사영시켰을 때의 지표인 BBOX와 AOS가 기존 PointPillars에 비해 더 높은 수치를 나타낸다는 것은 MOS 정보가 3차원 물체의 heading angle 추정 성능을 향상시켜준다는 것을 의미한다. [Fig. 5]에서도 MOS 라벨을 추가하여 학습한 알고리즘이 heading angle 측면에서 포인트 클라우드 만을 이용해 학습한 결과보다 더 잘 추정하는 것을 확인할 수 있다.

BBOX와 AOS 성능은 향상하였지만 BEV, 3D 성능은 하락했는데, 이는 본 논문에서 제시한 알고리즘이 움직이는 점들

[Table 3] Description about KITTI evaluation detection metrics

Evaluation metrics	Description
BBOX	An evaluation metric that assesses the Intersection over Union (IoU) between the projected 3D bounding box onto the image plane and the ground truth bounding box of the actual object.
BEV	An evaluation metric that assesses the IoU between the projected 3D bounding box onto the Bird's Eye View plane and the ground truth bounding box of the actual object.
3D	An evaluation metric that assesses the IoU between the 3D bounding box and the ground truth bounding box.
AOS	An evaluation metric that calculates the cosine similarity of the heading angles between the predicted bounding box and the ground truth box.



[Fig. 5] Qualitative results of our algorithm. Green box means result of MOS + PointPillars, blue box means result of ground truth and red box means result of PointPillars

과 그렇지 않은 점들, 즉 자동차와 주변 환경 사이 interaction을 학습하지 못했기 때문으로 판단된다. 이로 인해 bounding box의 size 추정 성능은 하락하였고, BEV, 3D 지표 성능이 낮게 측정되었다. 이는 알고리즘을 MOS, Detection 단계로 나누지 않고 end-to-end로 학습을 진행할 시 개선될 사항이라 생각되며, 추후 연구를 진행할 예정이다.

4.3 Ablation study

4.3.1 MOS 정보가 Detection 알고리즘의 heading angle 추정에 미치는 영향 분석

Detection 알고리즘의 input으로 MOS 정보를 추가로 부여했을 때 heading angle 추정 성능에 미치는 영향을 분석하기 위해 PointPillars 대신 SECOND를 사용하여 성능 평가를 진행하였다.

평가 결과는 [Table 4]와 같다. SECOND 알고리즘 또한 4채널로 이루어진 포인트 클라우드의 뒤에 MOS class를 추가하여 input으로 사용한 결과 BBOX지표와 AOS 지표 모두 향상한 것을 확인할 수 있다. 이는 MOS 정보가 Detection 알고리즘의 heading angle 추정에 도움을 준다는 것을 시사한다.

4.3.2 MOS 성능에 따른 Detection 알고리즘 성능 분석

MOS 성능이 Detection 알고리즘 성능에 미치는 영향을 분석하기 위해 ground truth MOS label을 이용하여 성능을 평가하였다. 이전과 마찬가지로 ground truth MOS label을 추가한 5채널 포인트 클라우드를 PointPillars의 입력 값으로 사용하여 학습을 진행하였다. 학습 시 모든 파라미터는 이전 학습과 동일한 값을 사용하였으며 실험 결과는 [Table 5]와 같다.

[Table 4] Performance of the algorithm that changed the detection part to SECOND

Dataset	Method	BBOX (AP)	AOS
KITTI-road	SECOND	94.88	61.89
	MOS+SECOND	95.11	62.16

[Table 5] Dependency on MOS quality. Δ means the difference from result of PointPillars

Dataset	Method	BBOX (AP)	AOS	Δ	
				BBOX	AOS
KITTI-road	PointPillars	84.02	55.20	0	0
	MOS+PointPillars	91.96	61.14	+7.94	+5.94
	GT+PointPillars	93.97	62.36	+9.95	+7.16
Semantic KITTI	PointPillars	82.24	46.87	0	0
	MOS+PointPillars	84.34	48.03	+2.1	+1.16
	GT+PointPillars	89.42	49.96	+7.18	+3.09

Ground truth를 이용하여 학습한 결과, KITTI-road 데이터셋을 사용하였을 경우에는 MOS 라벨을 사용하지 않고 포인트 클라우드만을 이용하여 학습한 PointPillars보다 BBOX 지표에서는 +9.95, AOS 지표에서는 +7.16만큼 성능 향상이 있었다. 또한 SementicKITTI 데이터셋을 사용하였을 경우에는 BBOX 지표에서는 +7.18, AOS 지표에서는 +3.09만큼 성능 향상을 보였다. 이는 MOS의 성능이 Object Detector의 heading angle 추정 성능에 영향을 미친다는 것을 의미한다. 그럼에도 4DMOS를 사용한 본 논문에서 제시한 알고리즘 또한 기존 PointPillars에 비해 성능 향상을 보임을 확인할 수 있고, 추후 MOS 알고리즘의 성능 향상에 따른 heading angle 추정 성능 향상 가능성 또한 존재한다.

5. 결 론

본 논문은 가로수나 빌딩과 같은 도심 환경 장애물이 노이즈로 작용하여 3차원 물체 오 탐지 등의 성능 저하 현상을 저감하기 위하여, 포인트 클라우드 전처리를 통한 3차원 물체의 heading angle 추정 성능 향상을 위한 알고리즘을 제안했다. 기존 포인트 클라우드를 MOS를 통해 움직이는 물체와 움직이지 않는 물체로 구분하고, 이를 포인트 클라우드에 새로운 정보로 추가하여 3차원 물체 탐지를 수행하였다. 실험 결과, AOS지표 측면에서 1~5%가량의 성능 향상을 확인하였고, 포인트 클라우드의 MOS 정보가 3차원 물체의 heading angle 추정에 도움을 줌을 확인하였다. 또한, GT MOS label을 사용하여 알고리즘을 수행한 결과 Object Detector의 heading angle 성능 향상을 가져왔고, 이로 MOS의 성능이 heading angle 성능에 영향을 미친다는 것을 검증하였다.

카메라로 촬영한 이미지에 semantic segmentation을 진행하여 3차원 물체 탐지 알고리즘의 추가 정보로 부여한 선행 연구와는 달리, 오직 라이다 센서의 결과만 가지고 물체 탐지 알고리즘의 성능 향상을 이루어 냈다는 점에서 의의가 있다. 또한, AOS 지표 측면에서 성능 향상은 물체 탐지의 하위 작업인 다중 물체 추적 및 경로 계획의 성능도 향상될 가능성이 존재함을 시사한다. 이에 본 연구에서 나아가 MOS를 통한 포인트 클라우드 가공을 이용해 다중 물체 추적 알고리즘 고도화 작업을 진행할 예정이다.

References

[1] H. A. Ignatious, H. El-Sayed, and M. Khan, “An overview of sensors in Autonomous Vehicles,” *Procedia Computer Science*, vol. 198, pp. 736-741, 2022, DOI: 10.1016/j.procs.2021.12.315.

[2] Y. Zhou and O. Tuzel, “VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection,” *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, pp. 4490-4499, 2018, DOI: 10.1109/CVPR.2018.00472.

[3] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, “PointPillars: Fast Encoders for Object Detection From Point Clouds,” *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA, pp. 12689-12697, 2019, DOI: 10.1109/CVPR.2019.01298.

[4] S. Vora, A. H. Lang, B. Helou, and O. Beijbom, “PointPainting: Sequential Fusion for 3D Object Detection,” *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, USA, pp. 4603-4611, 2020, DOI: 10.1109/CVPR42600.2020.00466.

[5] X. Chen, S. Li, B. Mersch, L. Wiesmann, J. Gall, J. Behley, and C. Stachniss, “Moving Object Segmentation in 3D LiDAR Data: A Learning-Based Approach Exploiting Sequential Data,” *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 6529-6536, Oct., 2021, DOI: 10.1109/LRA.2021.3093567.

[6] P. Wu, S. Chen, and D. N. Metaxas, “MotionNet: Joint Perception and Motion Prediction for Autonomous Driving Based on Bird’s Eye View Maps,” *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, USA, pp. 11382-11392, 2020, DOI: 10.1109/CVPR42600.2020.01140.

[7] B. Mersch, X. Chen, I. Vizzo, L. Nunes, J. Behley, and C. Stachniss, “Receding Moving Object Segmentation in 3D LiDAR Data Using Sparse 4D Convolutions,” *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 7503-7510, Jul., 2022, DOI: 10.1109/LRA.2022.3183245.

[8] N. Wang, C. Shi, R. Guo, H. Lu, Z.Zheng, and X. Chen, “InsMOS: Instance-Aware Moving Object Segmentation in LiDAR Data,” *arXiv:2303.03909*, Mar., 2023, DOI: 10.48550/arXiv.2303.03909.

[9] K. Park, G. Im, M. Kim, and J. Park, “Parking Space Detection based on Camera and LIDAR Sensor Fusion,” *Journal of Korea Robotics Society*, vol. 14, no. 3, pp. 170-178, Aug., 2019, DOI: 10.7746/jkros.2019.14.3.170.

[10] Y. Cho, H. C. Roh, and M. Chung, “Accurate Parked Vehicle Detection using GMM-based 3D Vehicle Model in Complex Urban Environments,” *The Journal of Korea Robotics Society*, vol. 10, no. 1, pp. 33-41, 2015, DOI: 10.7746/jkros.2015.10.1.033.

[11] D. Song, J.-B. Yi, and S.-J. Yi, “Development of an Efficient 3D Object Recognition Algorithm for Robotic Grasping in Cluttered Environments,” *The Journal of Korea Robotics Society*, vol. 17, no. 3, pp. 255-263, Aug., 2022, DOI: 10.7746/jkros.2022.17.3.255.

[12] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, “Multi-view 3D Object Detection Network for Autonomous Driving,” *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, pp. 6526-6534, 2017, DOI: 10.1109/CVPR.2017.691.

[13] J. Ku, M. Mozifian, J. Lee, A. Harakeh, and S. L. Waslander, “Joint 3D Proposal Generation and Object Detection from View Aggregation,” *2018 IEEE/RSJ International Conference on*

Intelligent Robots and Systems (IROS), Madrid, Spain, pp. 1-8, 2018, DOI: 10.1109/IROS.2018.8594049.

[14] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas, "PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, pp. 77-85, 2017, DOI: 10.1109/CVPR.2017.16.

[15] Y. Yan, Y. Mao, and B. Li, "SECOND: Sparsely embedded convolutional detection," *Sensors*, vol. 18, no. 10, Oct., 2018, DOI: 10.3390/s18103337.

[16] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Fu, and A. C. Berg, "SSD: Single Shot MultiBox Detector," *Computer Vision–ECCV 2016: 14th European Conference*, Amsterdam, pp. 21-37, 2016, DOI: 10.1007/978-3-319-46448-0_2.

[17] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal Loss for Dense Object Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 2, pp. 318-327, Feb., 2020, DOI: 10.1109/TPAMI.2018.2858826.

[18] J. Fritsch, T. Kühnl, and A. Geiger, "A new performance measure and evaluation benchmark for road detection algorithms," *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, The Hague, Netherlands, pp. 1693-1700, 2013, DOI: 10.1109/ITSC.2013.6728473.

[19] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall, "SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences," *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, Seoul, Korea (South), pp. 9296-9306, 2019, DOI: 10.1109/ICCV.2019.00939.

[20] J. Sun, Y. Dai, X. Zhang, J. Xu, R. Ai, W. Gu, and X. Chen, "Efficient Spatial-Temporal Information Fusion for LiDAR-Based 3D Moving Object Segmentation," *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Kyoto, Japan, pp. 11456-11463, 2022, DOI: 10.1109/IROS47612.2022.9981210.

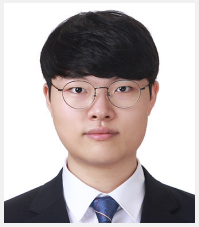
[21] X. Chen, B. Mersch, L. Nunes, R. Marcuzzi, I. Vizzo, J. Behley, and C. Stachniss, "Automatic Labeling to Generate Training Data for Online LiDAR-Based Moving Object Segmentation," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 6107-6114, Jul., 2022, DOI: 10.1109/LRA.2022.3166544.

[22] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," *2012 IEEE Conference on Computer Vision and Pattern Recognition*, Providence, RI, USA, pp. 3354-3361, 2012, DOI: 10.1109/CVPR.2012.6248074.

노 치 윤

2023 서울대학교 기계공학부(공학사)
2023~현재 서울대학교 기계공학부 석사과정

관심분야: Multi Object Tracking, Sensor Fusion



정 상 우

2021 KAIST 기계공학과(공학사)
2021 KAIST 전산학부(공학사)
2023 서울대학교 기계공학부(공학석사)
2023~현재 서울대학교 기계공학부 박사과정

관심분야: Legged Robot SLAM, Radar SLAM

김 유 진

2018 경희대학교 신소재공학과(공학사)
2020 서울대학교 기계공학부(공학석사)
2020~현재 서울대학교 기계공학부 박사과정

관심분야: 경로 예측, 물체 추적, 컴퓨터 비전

이 경 수

1985 서울대학교 기계공학부(공학사)
1987 서울대학교 기계공학부(공학석사)
1992 UC Berkeley 기계공학 전공(공학박사)
1993~2006 한양대학교 기계공학부 조교수, 부교수, 교수
2006~2023 서울대학교 기계공학부 교수

관심분야: Control Systems, Automated Driving of Ground Vehicles

김 아 영

2005 서울대학교 기계항공공학과(공학사)
2007 서울대학교 기계항공공학전공(공학석사)
2012 미시간대학교 기계공학전공(공학박사)
2014~2021 한국과학기술원 건설 및 환경공학과 부교수
2021~현재 서울대학교 공과대학 기계공학부 부교수

관심분야: 영상기반 SLAM