

# 거리 기반 적응형 임계값을 활용한 강건한 3차원 물체 탐지

## Robust 3D Object Detection through Distance based Adaptive Thresholding

이 은 호<sup>1</sup> · 정 민 우<sup>2</sup> · 김 중 호<sup>2</sup> · 이 경 수<sup>3</sup> · 김 아 영<sup>†</sup>

Eunho Lee<sup>1</sup>, Minwoo Jung<sup>2</sup>, Jongho Kim<sup>2</sup>, Kyongsu Yi<sup>3</sup>, Ayoung Kim<sup>†</sup>

**Abstract:** Ensuring robust 3D object detection is a core challenge for autonomous driving systems operating in urban environments. To tackle this issue, various 3D representation, including point cloud, voxels, and pillars, have been widely adopted, making use of LiDAR, Camera, and Radar sensors. These representations improved 3D object detection performance, but real-world urban scenarios with unexpected situations can still lead to numerous false positives, posing a challenge for robust 3D models. This paper presents a post-processing algorithm that dynamically adjusts object detection thresholds based on the distance from the ego-vehicle. While conventional perception algorithms typically employ a single threshold in post-processing, 3D models perform well in detecting nearby objects but may exhibit suboptimal performance for distant ones. The proposed algorithm tackles this issue by employing adaptive thresholds based on the distance from the ego-vehicle, minimizing false negatives and reducing false positives in the 3D model. The results show performance enhancements in the 3D model across a range of scenarios, encompassing not only typical urban road conditions but also scenarios involving adverse weather conditions.

**Keywords:** 3D Object Detection, Adaptive Thresholding, Autonomous Driving

### 1. 서 론

자율주행 기술은 운전자의 개입 없이 차량이 주변 환경을 인식하고 주행할 수 있도록 하여, 현대 교통 시스템의 패러다임을 혁신적으로 변화시키는 중요한 기술이다. 이를 가능하게 하는 핵심 기술 중 하나는 자율주행 인식 모듈이다. 이는 자율주행 시스템의 데이터 처리가 진행되는 첫 번째 모듈이면서,

라이다(LiDAR, Light Detection And Ranging), 카메라나 레이더(Radar, Radio Detection and Ranging) 센서 데이터를 사용해 주변 환경을 3차원 상에서 감지 및 해석할 수 있게 하고 차량이 도로를 안전하게 운행할 수 있게 하여 중요성이 더욱 강조되고 있다. 자율주행 인식 모듈에서는 주로 3차원 물체 탐지를 통해 주변 환경을 인지한다. 그러나, 실제 도심 도로에서는 교차로 또는 차량이 아주 많은 도로 등의 주행하기 어려운 상황이 발생하거나 다양한 기상 환경으로 인한 센서 데이터의 노이즈 발생 및 예상하지 못한 물체들이 탐지되고, 이로 인해 자율주행차량 시스템에서 급정거 등의 문제가 발생하여 안정적인 주행이 어렵다. 이러한 문제를 해결하기 위해 자율주행 시스템의 핵심 인지 기술인 3차원 물체 탐지 강건성에 대한 고도화가 필요하다.

최근 자율주행 인지 기술에서는 이를 위해 라이다에서 얻을 수 있는 3차원 포인트 클라우드를 주로 [Fig. 1]과 같이 voxel이나 pillar로 표현하여 활용한다. Voxel은 포인트 클라우드를 3차원 공간 내에서 격자(grid)로 나누고, 각 격자 셀 내에 포인트 클라우드 수 또는 정보들을 모으는 데이터 표현 방

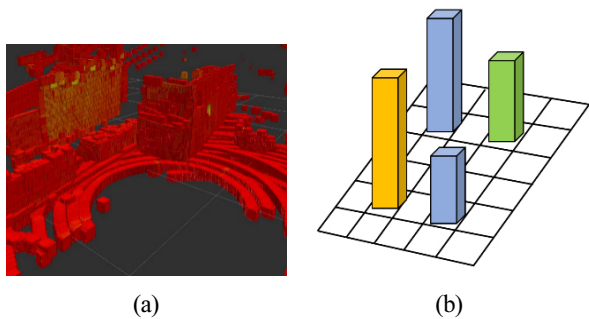
Received : Oct. 30. 2023; Revised : Dec. 1. 2023; Accepted : Dec. 5. 2023

※ This work was supported by the Technology Innovation Program (or Industrial Strategic Technology Development Program - Mobility and Connectivity Platform for Digital Transformation Acceleration in Unmanned Delivery) (20024355, Development of autonomous driving connectivity technology based on sensor-infrastructure cooperation) funded By the Ministry of Trade, Industry & Energy (MOTIE, Korea)

1. Master Student, Interdisciplinary Program in Artificial Intelligence, Seoul National University, Seoul, Korea (eunho1124@snu.ac.kr)
2. Ph.D. Student, Mechanical Engineering, Seoul National University, Seoul, Korea (moonshot, kimjhmj@snu.ac.kr)
3. Professor, Interdisciplinary Program in Artificial Intelligence, Seoul National University, Seoul, Korea (kyi@snu.ac.kr)

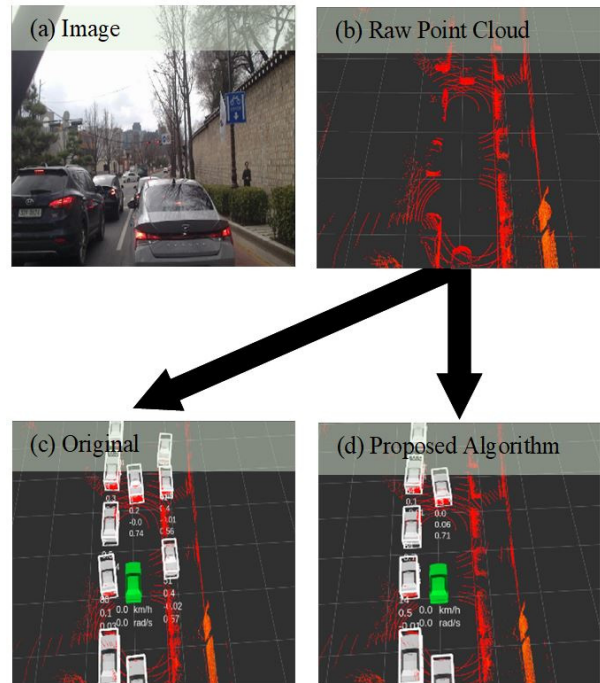
† Associate Professor, Corresponding author: Interdisciplinary Program in Artificial Intelligence, Seoul National University, Seoul, Korea (ayoungk@snu.ac.kr)

법이다. Pillar는 3차원 공간을 2차원 격자 열(column) 형태로 구조화하는 데이터 표현 방법이다. 각 열은 수직으로 정렬된 형태이며, 열 내 데이터들은 포인트 클라우드들의 높이 정보를 포함한 좌표 정보들을 나타낸다. 카메라는 2차원으로 표현되는 이미지를 사용하여 주변을 인지하기 때문에, 각 픽셀들에 대한 깊이 추정을 통해 3차원 포인트 클라우드들로 변환하여 물체 탐지를 수행한다. 레이더는 파장이 긴 전파를 이용하여 주변을 인지하기 때문에 상대적으로 탐지 거리가 멀며 다양한 기상 환경에 강건하지만 3차원 공간 내 데이터가 높이 정보를 포함하고 있지 않아, 주로 라이다나 카메라와의 센서 퓨전을 활용하여 3차원 물체 탐지를 수행한다. 라이다, 카메라, 레이더 센서를 활용한 3차원 물체 탐지 모델들은 각 센서 데이터를 활용하여 물체의 클래스, 위치 및 신뢰도 점수 등을 예측하고, 예측 결과를 신뢰도 점수로 후처리하여 최종 결과를 생성한다. 기존의 3차원 물체 탐지 모델들은 탐지 후 후처리에서 단일 임계값을 하이퍼파라미터(hyperparameter)로 정의하여 신뢰도 점수 후처리에 사용하는 것이 일반적이었다. 그러나 센서 해상도, 범위 등의 센서 데이터 특성과 훈련 데이터에 기반한 3차원 물체 탐지 모델들은 거리에 따라 물체 탐지의 정확도에 상이한 특성을 보인다. 상대적으로 자차로부터 가까운 거리의 물체들을 잘 탐지하여 비교적 높은 Recall과 신뢰도 점수를 결과로 나타내고, 먼 거리의 물체들에 대해서는 비교적 낮은 Recall과 신뢰도 점수를 결과로 나타낸다. 물체 인식에서 Recall의 향상은 미인지의 감소를 의미하며, Precision의 향상은 오인지의 감소를 의미한다. 또한, 일반적으로 Recall과 Precision은 반비례 관계이다. 이에 기반하여 자율주행차량으로 도로 위를 주행할 때, 가까운 거리에 대해서는 상대적으로 Recall보다 Precision을 우선시하고, 먼 거리에 대해서는 상대적으로 Precision보다 Recall을 우선시하는 것이 더 적합하다.



[Fig. 1] (a) and (b) show the representation for point cloud. (a) is the voxel, which subdivides point clouds into a 3D grid structure, collecting point cloud counts or information within each grid cell. (b) is pillar, which structurally organizes 3D space into a 2D grid column structure and represents coordinate information, including the height of point clouds

이러한 상황에 3차원 물체 탐지 모델 후처리에 단일 임계값을 사용하는 것은 탐지 정확도가 감소하며 오인지가 발생하고 자율주행에 적합하지 않다는 문제점이 존재한다. 본 논문에서는 이러한 특성을 보이는 3차원 물체 탐지 모델의 성능 개선을 위해 단일 임계값이 아닌 자차로부터의 거리 기반한 Adaptive thresholding 알고리즘으로 임계값을 사용하는 방법을 제안한다. 해당 방법론은 기존의 3차원 물체 탐지 알고리즘에 추가적인 학습이나 복잡한 알고리즘의 추가 없이 후처리 단계를 통합하여 거리에 따라 균형 잡힌 Recall/Precision을 갖도록 하고 정량적 성능을 향상시켜 자율주행에 더 적합한 모델로 개선하는 강점을 가진다. 이는, 자율주행차량에서 가까운 거리의 물체들에 대해서는 오인지를 최소화하고, 먼 거리에서 미인지는 물체의 수를 감소시켜 안정적이며 효율적인 주행을 가능하게 한다. 예를 들어, [Fig. 2]와 같이 단일 임계값을 적용했을 때 발생하던 실제 도심 도로 위의 차량과 가까운 거리에 존재하는 수풀과 같이 예상하지 못한 물체들이 차량으로 오인되어 급정거가 발생하는 문제를 가까운 거리의 물체에 대해 더 높은 임계값을 적용하여 차량으로 인지하지 않도록 개선하여 안정적인 주행을 가능하게 하였다. 이와 같은 물체 탐지의 성능 및 강건성 개선은 자율주행 기술의 핵심 기술 중 하나인 인지도도화에 큰 역할을 수행할 수 있다.



[Fig. 2] (a) Image of the urban road near the Cheongwadae. (b) Point clouds of the urban road near the Cheongwadae. (c) 3D Object Detection results that mistakenly identify bushes as vehicles using a single threshold. (d) The proposed algorithm improves the misidentification situation as depicted in (c)

본 논문에서 제시하고자 하는 논점은 다음과 같다.

- 3차원 물체 탐지 모델 후처리에 단일 임계값이 아닌 자차로부터의 거리에 기반한 Adaptive thresholding을 적용하여 거리에 따라 오인지 및 미인지 결과를 줄여 모델의 강건성을 높였고, 이를 통해 보다 안정적인 주행이 가능하게 하는 자율주행에 적합한 모델로 개선하였다.
- 간단하지만 효과적인 알고리즘으로 여러가지 3차원 물체 탐지 모델에 쉽게 적용시킬 수 있음을 검증하였다.
- Adaptive thresholding 알고리즘이 모델의 오인지 및 성능을 개선하였음을 Recall/Precision, Trade-off 및 mAP를 통해 검증하였다.
- 다양한 실제 도심 도로 및 기상 환경에서 모델의 오인지 문제를 개선하였음을 실차 주행 데이터에 기반해 정성적으로 검증하였다.

## 2. 선행 연구 조사

### 2.1 Adaptive thresholding

기준에 진행된 Adaptive thresholding에 관한 연구들은 [1,2]와 같이 2차원 이미지 상 움직이는 객체 탐지를 위해 진행되었다. [1]은 이미지 상에서 객체 탐지를 진행한 후, 이전 프레임과의 픽셀 간 비교를 통해 객체의 상태와 거리를 추정해 거리에 따른 Adaptive thresholding을 사용하여 탐지 정확도를 향상시켰다. [2]는 이미지를 3차원 표현으로 변환하기 위한 과정에서 깊이 추정을 수행할 때, 해당 이미지 상의 특징들(features)을 한 쌍으로 고려하여 움직이는 물체와 정지한 물체를 구분하기 위한 임계값을 정하는데 Adaptive thresholding을 사용한다. 이 임계값은 특징들 간의 거리에 따라 동적으로 조절되며, 이를 통해 객체의 움직임 여부를 판단한다. 이러한 Adaptive thresholding에 관한 선행 연구들은 2차원 이미지에 초점을 맞추고 있어, 다양한 상황 및 물체 인식을 위한 3차원 공간 내 표현이 필요한 자율주행의 3차원 물체 탐지 모델에는 적합하지 않다.

### 2.2 3D Object Detection

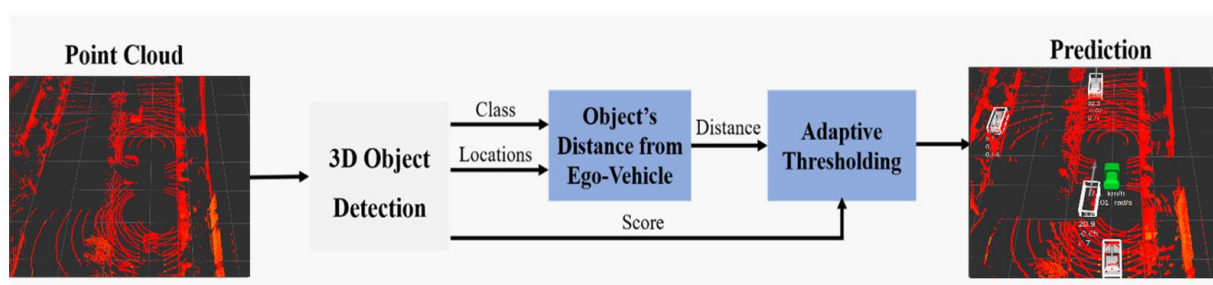
#### 2.2.1 LiDAR based 3D Object Detection

라이다 기반 3차원 물체 탐지 관련 연구는 voxel, pillar, 그리고 raw point cloud 방법 모델로 분류할 수 있다. VoxelNet<sup>[3]</sup>은 3차원 포인트 클라우드들을 voxel 단위로 집단화(grouping)하고 sampling하여 특징을 표현하고 RPN<sup>[4]</sup>으로 3차원 컨볼루션 연산을 하여 물체 탐지를 수행한다. SECOND<sup>[5]</sup>는 3차원 공간에서 희소화(sparse) 특성을 가지는 voxel을 효율적으로 계산

하는 방법을 제안한 모델이다. PointPillars<sup>[6]</sup>는 pillar 표현을 사용하여 2차원 컨볼루션 연산을 가능하게 해 계산 및 메모리 사용을 효율적으로 좋은 성능을 보이는 모델이다. PointRCNN<sup>[7]</sup>은 raw point cloud를 사용하여 객체와 배경을 분류한 후, 이를 활용하여 물체 탐지를 수행하여 높은 정확도를 달성하였다. PC-RCNN<sup>[8]</sup>은 raw point cloud와 voxel을 모두 활용하여 물체 탐지를 수행하는 모델이고, 계산량이 많지만 포인트 클라우드의 특성을 거의 보존하여 좋은 성능을 보인다. 이러한 연구들은 anchor에 기반하여 물체 탐지를 진행하는 모델들이다. 이후에, Yin et al.은 이를 anchor-free 방식으로 대체한 CenterNet<sup>[9]</sup>을 제안하였다. [9]는 object size, dimension, orientation, pose 등과 같은 특징들을 center location으로부터 직접적으로 regression하여 3차원 물체 탐지를 수행하여, 빠른 속도로 탐지하며 좋은 성능을 보인다. 3차원 물체 탐지 모델들은 대부분 일반적인 도로 상황에 대한 특정 데이터셋을 기반으로 학습되어 평가된다. 그러나, 이러한 모델들은 자율주행차량이 실제 도심 도로 위에서 직면하는 더욱 다양하고 어려운 상황들에 대해 대처하지 못한다. 강한 비, 눈, 안개와 같은 다양한 기상 환경으로 인한 센서 데이터의 노이즈 발생으로 인하여 오인지 발생이 증가할 수 있다. 또한, 수풀, 나무 또는 도로 표지판 등이 차량 또는 장애물로 인지되는 예상치 못한 오인지 상황도 발생할 수 있다. 자율주행차량이 실제 도심 도로 위에서 안정적인 주행을 하기 위해 이러한 오인지 상황들에 유연하게 대응할 수 있는 추가적인 방법이 필요하다.

#### 2.2.2 Camera based 3D Object Detection

라이다 센서의 높은 비용으로, 카메라를 활용한 3차원 물체 탐지 연구가 활발히 진행되었다. 이에 기반이 되는 2차원 물체 탐지에 관한 다양한 연구<sup>[4,10-12]</sup>가 진행되었다. [13]은 복잡한 환경에서 2차원 물체 탐지 모델인 yolo와 mask 기반 물체 탐지와 클러스터링 알고리즘을 사용하여 물체의 3차원 pose를 효율적으로 감지하며, 겹치는 bounding box에 대한 성능 저하를 개선한 알고리즘을 제안하였다. FCOS3D<sup>[14]</sup>는 기존의 2차원 물체 탐지 모델인 FCOS<sup>[15]</sup>에서 regression target을 3D target으로 재정의하고, 추가적인 disentangled head를 통해 얻은 구성 요소를 활용하여 3D bounding box regression을 수행한다. 또한, 2D detector 구조로 3차원 물체 탐지를 진행하기 위해 기존의 center-ness를 가우시안 분포를 추가한 3D-center로 변경하여 사용하여 이미지 깊이 추정 모델링 방법을 개선하였다. 이후에 트랜스포머(Transformer)를 사용한 모델들이 연구되었는데, 2차원 물체 탐지에 트랜스포머를 적용한 DETR<sup>[16]</sup>을 기반으로 DETR3D<sup>[17]</sup>, PETR<sup>[18]</sup> 및 GraphDETR3D<sup>[19]</sup> 모델들이 연구되었다. 이 모델들은 이미지 상에서 직접적으로 3D bounding box를 예측하거나 깊이 추정을 하는 기존의 방법들을 사



[Fig. 3] Overall architecture illustrating pipeline of proposed algorithm

용하는 것이 아닌, 학습가능한 객체 쿼리를 사용한 [17] 기반 detection head를 3차원 공간에서 곧바로 물체 탐지를 수행하여 성능이 개선되었음을 입증하였다.

### 2.2.3 Fusion based 3D Object Detection

최근, 센서 퓨전은 3차원 물체 탐지 분야에서 더 활발하게 연구되고 있다. 선행 연구들은 proposal-level과 point-level fusion 방법으로 분류할 수 있다. Proposal-level fusion 방법은 물체 중심이어서, 3차원 물체 탐지 외에 다른 연구들에 일반화하여 적용할 수 없다. MV3D<sup>[20]</sup>는 3차원 공간에서 포인트 클라우드를 활용하여 object proposal을 생성하고, 이를 이미지에 투영하여 RoI (Region of Interest)를 추출한다. F-PointNet<sup>[21]</sup>, F-ConvNet<sup>[22]</sup>, 그리고 CenterFusion<sup>[23]</sup>은 이미지 proposal을 3차원 표현으로 변환한다. Point-level fusion 방법은 일반적으로 이미지의 semantic 특징을 포인트 클라우드와 퓨전하여 물체 탐지를 수행한다. FUTR3D<sup>[24]</sup>와 TransFusion<sup>[25]</sup>은 각 센서들의 데이터들을 통합된 3차원 공간 내에서 물체 쿼리(object queries)로 정의하고, 트랜스포머(Transformer)<sup>[26]</sup>를 사용해 물체 탐지를 수행한다. BEVFusion<sup>[27]</sup>는 카메라와 라이다 센서 데이터의 특징들을 Bird Eye's View로 투영한 후 퓨전하여 물체 탐지를 수행한다. 또한, 센서 퓨전을 활용하여 물체만이 아니라 주변 환경을 탐지하는 모델들<sup>[28-31]</sup>도 활발히 연구가 진행 중이다.

## 3. 연구 방법

### 3.1 System Overview

전체적인 시스템에 대한 diagram은 [Fig. 3]와 같다. 라이다 포인트 클라우드를 전달받아 3차원 물체 탐지 모델이 주변 물체들을 탐지하고, 탐지된 물체들에 대해 Car, Pedestrian, Cyclist 등의 물체 클래스와 3차원 공간 내 위치 및 신뢰도 점수를 결과로 내보낸다. 해당 결과에서 물체들의 위치를 3차원 좌표들로 전달받아 자차로부터 각 물체들의 거리를 계산하고, 계산

된 거리와 3차원 물체 탐지의 신뢰도 점수 결과를 전달받아 3.4절에서 자세하게 설명할 Adaptive Thresholding 알고리즘을 적용한다. 해당 알고리즘을 통해 모델의 탐지 결과를 후처리하여, 최종적으로 물체를 Bounding Box 형태로 예측한다.

### 3.2 3D Object Detection model

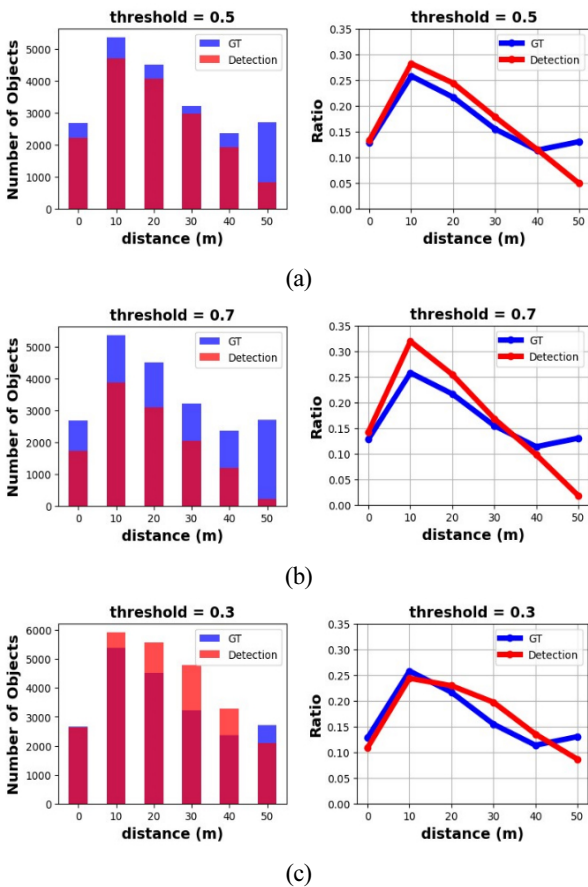
연구에는 PointPillars<sup>[6]</sup>를 3차원 물체 탐지 모델로 사용하여 알고리즘 연구를 진행하였는데, 이 모델은 총 3개의 단계로 구성된다. 첫 번째 단계인 Pillar Feature Net은 3차원 물체 탐지를 2차원 컨볼루션 연산이 가능한 형태로 변환한다. 먼저,  $(x, y, z, r)$  좌표로 구성되는 포인트 클라우드를 [Fig. 1(b)]와 같이  $x$ - $y$  평면을 grid로 세분화하여 각각 좌표 및 반사율인  $(x, y, z, r)$  뿐만 아니라  $x, y, z$ 의 각 산술평균  $(x_c, y_c, z_c)$ 와 pillar 중심점 좌표  $(x_p, y_p)$ 를 포함하는 9차원 pillar 형태로 변환한다. 변환한 pillar들에 특징 인코딩(feature encoding)하는 과정을 거친 후, 특징들을 원래의 pillar 위치로 다시 이동시켜 2차원 컨볼루션 연산이 가능한 pseudo-image를 생성한다. 두 번째 단계인 Backbone에서는 pseudo-image를 전달받아 2차원 컨볼루션 연산을 진행해 점점 작은 resolution feature를 생성한 후, upsampling 및 concatenate하여 feature들을 모은다. 마지막 단계인 Detection Head에서는 SSD<sup>[32]</sup>을 사용하여 높이 정보도 포함된 물체의 3차원 좌표  $(x, y, z)$ , 클래스와 신뢰도 점수를 결과로 내보낸다. 제안된 알고리즘은 탐지된 물체들의 각 3차원 좌표, 클래스 및 신뢰도 점수를 결과로 내보내는 모델들에 모두 적용할 수 있다<sup>6)</sup>. 모델 이외에 다른 여러가지 3차원 물체 탐지 모델에도 해당 알고리즘을 적용시킨 결과는 4.2절에서 자세하게 설명할 것이다.

### 3.3 Analysis Dataset and 3D Object Detection Result

연구에 사용한 데이터셋의 Ground Truth와 0.3, 0.5, 0.7의 단일 임계값을 적용한 3차원 물체 탐지 모델의 결과를 [Fig. 4]에 나타내었다. [Fig. 4]에서 Ground Truth와 탐지된 물체 수 비



교로 거리에 따른 Recall의 변화 추이를 분석할 수 있다. 또한, 다양한 임계값에 대하여 거리에 따른 물체 탐지 수의 비율 비교로 각 임계값이 거리 별로 물체 탐지 성능에 미치는 영향을 파악할 수 있다. 일반적으로 많이 사용되는 임계값인 0.5에 대한 [Fig. 4(a)]의 결과에서 알 수 있듯이, 3차원 물체 탐지 결과들은 상대적으로 차차로부터 가까운 거리( $d < 30\text{ m}$ )에서는 잘 탐지하여, Ground Truth보다 더 높은 비율로 물체들을 탐지함을 알 수 있다. 상대적으로 먼 거리( $d > 30\text{ m}$ )에서는 Ground Truth보다 낮은 비율로 물체들을 탐지하는 결과에서 탐지 성능이 떨어짐을 알 수 있다. 즉, 상대적으로 가까운 거리에서는 Recall이 높지만, 먼 거리에서는 Recall이 낮음을 알 수 있다. 또한, Recall과 Precision은 반비례 관계이므로 상대적으로 가까운 거리에서는 Precision이 낮고, 먼 거리에서는 Precision이 높음을 알 수 있다. 따라서, 가까운 거리의 물체들에 대해 Precision을 높이기 위해 더 높은 0.7의 단일 임계값을 적용하

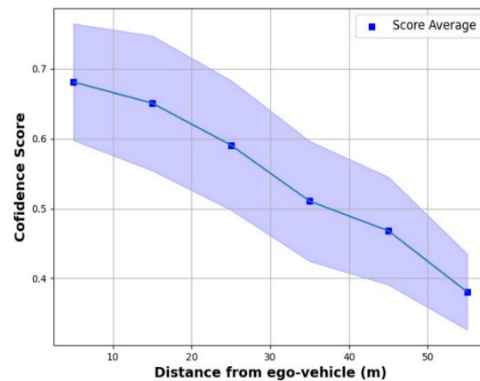


[Fig. 4] PointPillars<sup>[6]</sup> Detection Result Analysis based on distance about number of objects and ratio with different score threshold values. Blue color is Ground Truth result and Red color is Detection result. (a) is 0.5 score threshold, (b) is detection result with high score threshold (=0.7), (c) is detection result with low score threshold (=0.3). Single score threshold is not effective in managing both false negatives and false positives

여 결과를 분석하였다. [Fig. 4(a)]와 [Fig. 4(b)]를 비교한 결과, 높은 임계값 적용으로 전반적으로 탐지된 물체 수가 감소하였고, 특히 40 m 이상의 물체들의 탐지 비율이 많이 감소하여 Recall이 낮아지므로 미인지가 더 많이 발생함을 확인할 수 있었다. 반대로, 먼 거리의 물체들에 대해 Recall을 향상시키기 위해 더 낮은 0.3의 단일 임계값을 적용하여 결과를 분석하였다. [Fig. 4(a)]와 [Fig. 4(c)] 비교 결과, 낮은 임계값 적용으로 탐지된 40 m 이상의 물체 수 및 비율이 증가하여 Recall이 향상되었지만, 10 m 이상 40 m 이하 거리에서 탐지된 물체 수가 Ground Truth보다 더 많고, 탐지된 비율도 증가하여 전반적으로 Precision이 더 감소함을 확인할 수 있었다. 그래서, 3.4절에서 자세하게 설명할 차차로부터의 거리 기반 Adaptive Thresholding 알고리즘을 모델에 적용하기 위해, 단일 임계값을 적용하였을 때 3차원 물체 탐지 결과를 분석하였다. 모델의 결과를 10 m 단위의 거리로 나눈 후에, 각 거리마다 신뢰도 점수 경향성을 평균값과, 표준편차로 분석하였고, 이를 [Fig. 5]에 나타냈다. 이 때, 거리에 기반한 신뢰도 점수를 나타내기 위해 [Fig. 4]에 기반하여 40 m 이하의 거리에서는 0.5의 단일 임계값을 적용한 결과를 사용하였고, 40 m 이상에서는 0.3의 단일 임계값을 적용한 결과를 사용하였다. 분석한 결과를 알고리즘에 적용하는 것은 3.4절에서 자세하게 설명할 것이다.

### 3.4 Distance based Adaptive Thresholding

3.3에서 분석한 내용대로, 차차로부터 가까운 곳에 데이터셋과 탐지 결과가 상대적으로 많고 신뢰도 점수도 높으며 먼 곳은 데이터셋과 탐지 결과가 상대적으로 적고 신뢰도 점수가 낮다. 따라서, 일반적으로 모델 후처리에 사용되는 동일함 임계값을 적용하는 것이 아닌, 이러한 결과에 맞는 Distance based Adaptive Thresholding을 적용할 것이다. [Fig. 5]에 기반한 자



[Fig.5] Confidence score with a single threshold (=0.4). The blue dots represent the score mean values at each distance (10 m), and the blue-shaded area indicates the standard deviation

차로부터의 거리 기반 Adaptive thresholding 식은 아래와 같다.

$$\text{Score Threshold} = \begin{cases} \alpha d^2 + \beta d + \gamma & (0 < d \leq \delta) \\ k & (d > \delta) \end{cases} \quad (1)$$

[Fig. 5]에서 볼 수 있듯이, 10 m 단위의 각 거리마다 신뢰도 점수 평균값들에 대한 추세를 추정했을 때, 이차 함수 형태의 곡선이 가장 적합하다는 결과를 얻었다. 이 때, 단일 임계값을 0.5가 아닌 0.4로 사용하여 차차로부터 먼 거리에 있는 물체들에 대한 정보들이 더 잘 포함되게 하였다.  $d$ 는 차차로부터 물체까지의 거리를 뜻하고  $m$  단위이며,  $\alpha$ ,  $\beta$ 와  $\gamma$ 는 Adaptive Thresholding에 사용되는 이차곡선의 형태를 결정하는 파라미터이다. 또한,  $\delta$ 는 제시된 알고리즘을 적용할 최대 거리와 관련한 파라미터이고,  $k$ 는 거리가  $\delta$ 일 때의 이차곡선의 함수 값인 상수함수를 나타내는 파라미터이다.  $\delta$ 는 라이다 최대 인지 거리 성능 및 3차원 물체 탐지 모델의 성능에 따라 달라질 수 있으며, 이차곡선 형태에서 거리가 멀어질수록 신뢰도 임계값이 계속해서 감소하므로,  $k$ 는  $\delta$ 보다 먼 거리에서 임계값이 너무 작은 값이 되어 오인되지는 물체가 많이 증가하는 상황을 발생시키지 않기 위해 파라미터로 설정하였다. 이러한 파라미터들을 통하여, PointPillars<sup>[6]</sup> 모델이 아닌 여러가지 3차원 물체 탐지 모델에도 해당 알고리즘을 적용할 수 있다.

Adaptive Thresholding 알고리즘은 자율주행 시스템에 적합하게 오인지를 줄이며 미인지를 크게 증가시키지 않는 것이 목표이다. 다시 말해, 제시된 알고리즘은 가까운 거리의 물체들의 Recall은 줄이고 Precision은 높이며, 먼 거리의 물체들의 Recall은 높이고 Precision을 낮춘 모델의 균형 잡힌 예측을 달성하는 알고리즘이다. 이에 따라 Adaptive Thresholding 알고리즘의 식 (1)의 파라미터들을 조정하기 위한 기준이 필요하다. 이에 대한 기준을 식 (2)와 식 (3)으로 정의된 거리 별 신뢰도 점수 평균값과 그에 대한 표준편차를 사용하여 정하였다.

$$\text{Score Mean}_d = \frac{\sum_{i=d}^{d+1} \text{Score}_i}{N_d} \quad (d=0, 1, 2, 3, 4, 5) \quad (2)$$

$$\text{Score Std}_d = \sqrt{\frac{1}{N_d} \sum_{i=d}^{N_d} (\text{Score}_i - \text{Score Mean}_d)^2} \quad (d=0, 1, 2, 3, 4, 5) \quad (3)$$

식 (2)는 0 ~ 60 m까지 10 m 간격으로 구간을 나누어 각각  $d$ 번째 구간으로 정의하였고, 각 구간 별로 탐지된 물체들의 신뢰도 점수 합을 해당 구간에서의 탐지된 물체 수로 나누어 신뢰도 점수 평균 값을 구한다. 마찬가지로, 식 (3)도 10 m 간격으로 구분된  $d$ 번째 구간에서 탐지된 물체의 신뢰도 점수와 신

뢰도 평균값으로 각각의 표준편차를 구한다. 이와 같이 식 (2)에서 얻은 거리 별 신뢰도 점수 평균값 6개를 식 (3)의 표준편차 값 내에서 조정하며 2차 곡선 형태의 추세선을 얻어 식 (1)의  $\alpha$ ,  $\beta$ ,  $\gamma$  및  $k$  파라미터들의 값들을 정한다.

## 4. 실험 및 결과

### 4.1 평가 데이터셋 및 방법

평가에 활용한 데이터는 3차원 물체 탐지 모델의 정성적 성능 평가를 위한 오픈 데이터셋인 Kitti 3D Object Detection Dataset<sup>[33]</sup>와 정량적 성능 평가를 위한 자체 도식 도로 데이터셋을 사용하였다.

Kitti 3D Object Detection Dataset<sup>[33]</sup>은 두 대의 고해상도 컬러 및 흑백 카메라, Velodyne HDL-64E S2 모델 라이다와 GPS를 장착한 차량으로 취득한 오픈 데이터셋이다. Velodyne HDL-64E S2 모델 라이다는 32개의 레이저 빔을 수직 방향 0.5°로 나눈 64채널 라이다로 최대 감지 거리는 120 m이고, 수직 시야각(Vertical Field of View)은 31.5°이다. 센서는 5-20 Hz로 회전하며 주변 360°를 스캔하고, 초당 130만개의 포인트 클라우드들을 수집한다. 이 데이터셋은 독일 바덴-뷔르템베르크 주에 위치한 Karlsruhe의 중소도시, 시골 지역 및 고속도로를 주행한 데이터를 포함하고 있으며 총 7,481개의 training image/point cloud와 7,518개의 test image/point cloud를 포함한다. 물체에 대한 ground truth를 이미지 및 포인트 클라우드를 사용하여 제공하고 Car, Pedestrian, 그리고 Cyclist의 세 가지 클래스로 구성되어 있다. Car 클래스가 전체 데이터셋의 93%를 이루고 있으며, 나머지는 Pedestrian과 Cyclist가 할당되어 있다.

자체 도식 도로 데이터셋은 Velodyne 32CH LiDAR 6개가 장착된 일렉시티 차량으로 청와대와 경복궁 주위를 주행하며 좋은 기상 환경과 우천 환경에서의 데이터를 취득하였다. 또한, Velodyne 32CH LiDAR 4개가 장착된 카니발 차량으로 안개가 많고 보슬비가 내리는 강릉 도로 주위를 주행하며 데이터를 취득하였다.

본 논문에서 제안한 Adaptive thresholding 알고리즘의 성능을 평가하기 위해 단일 임계값을 적용한 3차원 물체 탐지 모델과 제안한 알고리즘을 적용한 3차원 물체 탐지 모델의 결과에 대해 정량적, 정성적으로 비교하였다. 정량적, 정성적 평가에 대한 실험 방법은 각각 4.2, 4.3절에 자세히 나타냈다.

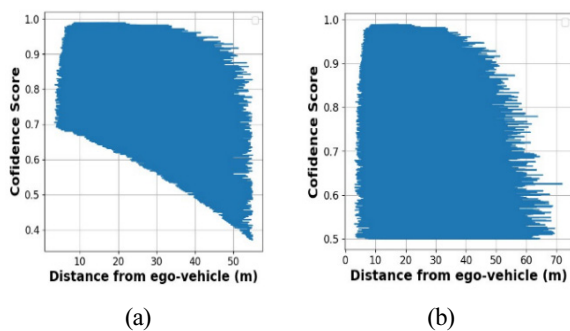
### 4.2 정량적 평가

본 논문에서 제안하는 알고리즘은 3차원 물체 탐지 모델이 자율주행 시스템에 적합한 모델인가에 대한 정량적 성능 평가

를 수행하기 위해 오픈 데이터셋인 Kitti 3D Object Detection Dataset<sup>[33]</sup>를 사용하였고, 가장 많은 비율을 차지하는 Car class를 사용하였다. 물체 인식에서 Recall과 Precision의 향상은 각각 미인지와 오인지의 감소를 뜻한다. 또한, mAP는 Recall과 Precision 값에 기반하여 모델의 강건성과 정확도를 종합적으로 평가하는 지표이다. 이를 통해, 안전하고 효율적인 주행이 가능하도록 하는 자율주행 시스템에 적합한 모델인가에 대한 평가를 정량적으로 정의하면 3차원 물체 탐지 모델이 mAP가 감소하지 않으면서, 균형 잡힌 Recall과 Precision 값을 가지는 것이다. 이는 모델이 물체를 올바르게 감지하고 인지하는 것 뿐만 아니라, 오인지를 최소화하여 안전한 주행을 할 수 있음을 의미한다. 이에 따라, 알고리즘의 성능을 3차원 물체 탐지 모델에 적용하기 전과 후의 Recall, Precision, Recall과 Precision 간의 Trade-off 및 mAP를 이용하여 측정하였다.

PointPillars<sup>[6]</sup>에 0.5의 단일 임계값을 적용했을 때와 제시된 알고리즘을 적용하였을 때의 탐지된 물체들에 대한 거리별 신뢰도 점수를 [Fig. 6]에 나타냈다. 기존에 사용되던 단일 임계값을 적용하였을 때는 [Fig. 6(a)]와 같이 탐지 결과가 거리에 상관없이 같은 임계값으로 후처리되지만, [Fig. 6(b)]에서 볼 수 있듯이 제시된 알고리즘을 적용하면 거리에 따라 이차곡선 형태로 임계값이 사용되어 거리 별 데이터 특성에 맞는 탐지 결과를 예측함을 확인할 수 있다. 다시 말해, 자차로부터 가까운 거리의 물체들에 대해서는 더 높은 임계값을 적용하고 먼 거리의 물체들에 대해서는 상대적으로 낮은 임계값을 적용한다.

PointPillars<sup>[6]</sup>에 여러가지 단일 임계값을 적용하였을 때와 제시된 알고리즘을 적용하였을 때의 결과를 비교한 것이 [Table 1]이다. 3.4절의 내용에 기반하여 식 (2)에서 얻은 거리 별 신뢰도 점수 평균값 6개를 식 (3)의 표준편차 값 내에서 거리 별



[Fig. 6] (a) The post-processed confidence score results of detected objects using the conventional single threshold (=0.5) (b) In contrast to (a), the post-processed confidence score results of detected objects applying our algorithm. Through score thresholding with a quadratic function shape based on the distance from the ego-vehicle, we predict detection results that align with the characteristics of the data based on the distance

특성을 고려해 조정하며 반복적으로 실험하여 최적의 파라미터 값을 정하였다. 실험 결과에 따라 알고리즘에 사용한 식 (1)의 파라미터는  $\alpha = -0.00002, \beta = -0.0061, \gamma = 0.6828, k = 0.3$ 이다. 알고리즘을 적용하였을 때, 0.3, 0.5와 0.7의 단일 임계값을 사용하였을 때보다 Trade-off가 감소하였고, mAP는 유사한 성능을 보인다. 이를 통해, 본 연구에서 제안한 알고리즘을 통해, 가까운 거리의 물체들은 Recall이 감소하고 Precision이 증가하였지만 먼 거리의 물체들은 Recall이 증가하고 Precision이 감소하였음을 알 수 있다.

또한, 다른 3차원 물체 탐지 모델들에 대해서도 해당 알고리즘의 유효성을 검증하기 위해 평가를 진행하였다. [6]와 동일하게 포인트 클라우드를 전달받아 물체의 클래스, 위치 및 신뢰도 점수를 Bounding Box 형태로 결과로 내보내는 SECOND<sup>[5]</sup>, PointRCNN<sup>[7]</sup>, PV-RCNN<sup>[8]</sup> 모델들을 사용하였고, [6]를 활용하여 진행한 파라미터들로 성능 평가를 진행한다. 각 모델들에 대해 0.5값의 단일 임계값을 적용하였을 때의 결과는 [Table 2]이고 제시된 알고리즘을 적용하였을 때의 결과는 [Table 3]이다. [Table 3]에서 볼 수 있듯이, PointPillars<sup>[6]</sup>, SECOND<sup>[5]</sup>, PointRCNN<sup>[7]</sup>, PV-RCNN<sup>[8]</sup> 모두 단일 임계값을 적용했을 때와 비교해 Trade-off가 감소하는 효과를 확인하였으며, 이는

[Table 1] Recall, Precision and mAP comparison for PointPillars<sup>[6]</sup> applying various single threshold and adaptive threshold

Threshold	Recall	Precision	Trade-off	mAP
0.3	0.895	0.646	0.249	77.28
0.5	0.807	0.847	0.040	77.29
0.7	0.655	0.943	0.288	77.49
Ours	0.786	0.813	0.025	77.29

[Table 2] Recall, Precision and mAP comparison for various 3D Object Detection models applying single threshold 0.5

Model	Recall	Precision	Trade-off	mAP
PointPillars <sup>[6]</sup>	0.807	0.847	0.040	77.28
SECOND <sup>[5]</sup>	0.808	0.856	0.048	78.62
PointRCNN <sup>[7]</sup>	0.899	0.848	0.051	78.74
PV-RCNN <sup>[8]</sup>	0.969	0.731	0.238	79.25

[Table 3] Recall, Precision and mAP comparison for various 3D Object Detection models applying adaptive threshold

Model	Recall	Precision	Trade-off	mAP
PointPillars <sup>[6]</sup>	0.786	0.813	0.025 (-0.015)	77.28
SECOND <sup>[5]</sup>	0.792	0.823	0.031 (-0.016)	78.62
PointRCNN <sup>[7]</sup>	0.849	0.815	0.034 (-0.017)	78.73
PV-RCNN <sup>[8]</sup>	0.893	0.792	0.101 (-0.137)	79.49

본 연구에서 제안한 알고리즘을 적용했을 때 여러가지 3차원 물체 탐지 모델들이 자율주행에 더 적합한 결과로 예측함을 나타낸다. 또한, PointPillars<sup>[6]</sup>로 진행한 연구의 결과 파라미터들이 다른 모델들에도 효과적임을 확인하였다.

#### 4.3 정성적 평가

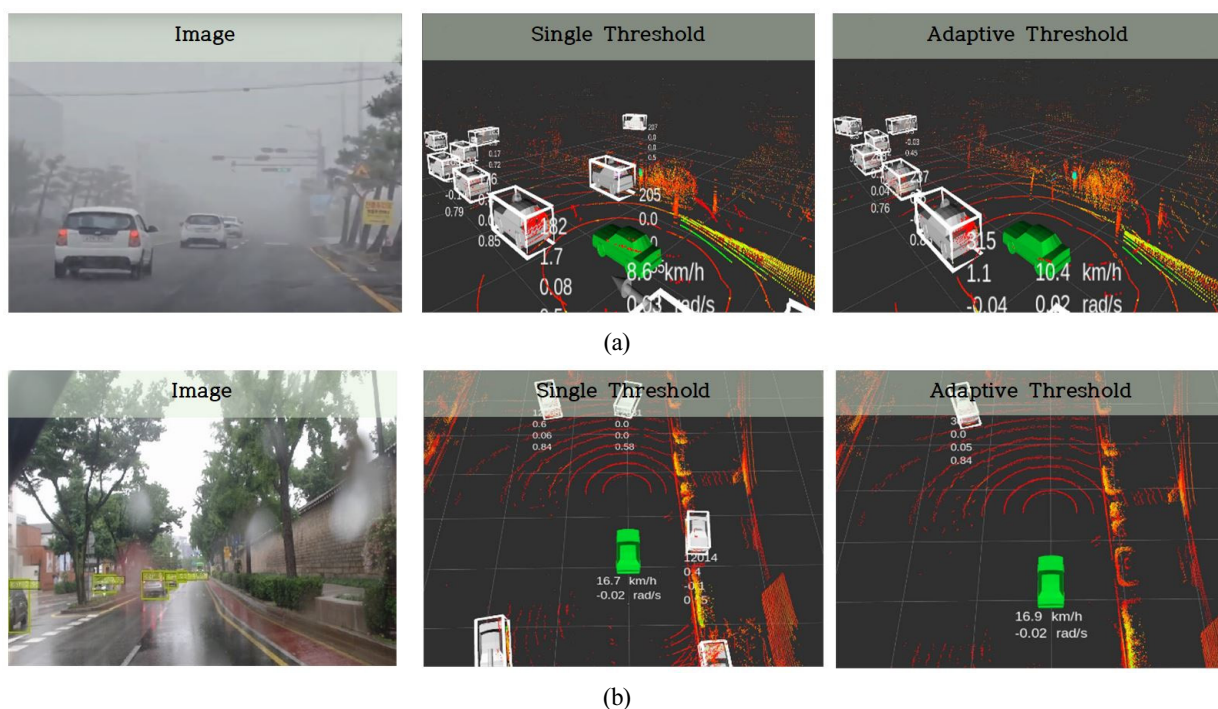
자체 도심 도로 데이터셋을 사용해 다양한 실제 도심 도로 및 기상 환경에서 모델의 오인지 문제에 대하여 정성적으로 평가할 것이다. 정성적 평가에서는 PointPillars<sup>[6]</sup>를 활용하여 진행하였다.

청와대와 경복궁 주위에서 좋은 기상환경 내 취득한 도심 도로 데이터가 [Fig. 2]이다. [Fig. 2(a)]에서 주행 중인 도로 오른쪽에 수풀이 수평 방향으로 계속해서 존재하는 것을 확인할 수 있다. 수풀에서 많은 포인트 클라우드가 나오며 형상이 차량과 비슷하여 단일 임계값을 적용하였을 때 모델이 수풀을 차량으로 오인지하는 것을 [Fig. 2(c)]에서 확인할 수 있다. 실제 자율주행 시, 수풀이 차량으로 오인지되고, 이것이 순간적으로 오른쪽 차선에서 주행 중인 차선으로 차선 변경하는 것으

로 잘못 인지되어 차량이 급정거하는 문제가 발생하였다. 단일 임계값이 아닌 본 논문에서 제시한 Adaptive Thresholding 알고리즘을 모델에 적용하였을 때, 차량으로 오인지 됐던 수풀을 인지하지 않으며, 왼쪽 차선과 전방에 존재하는 차량은 여전히 정확하게 인지되는 결과를 [Fig. 2(d)]에서 확인할 수 있다.

강릉 주위에서 안개가 많고 보슬비가 내리는 환경에서 취득한 도심 도로 데이터가 [Fig. 7(a)]이다. [Fig. 7(a)] 왼쪽 이미지에서 왼쪽 차선에 차량들이 존재하고, 안개가 많은 상황임을 알 수 있다. 안개 데이터의 특징은 라이더와 같은 높이에서 연기 같은 포인트 노이즈 형태로 나타나는 것이다. 이로 인해, 단일 임계값을 적용하였을 때, [Fig. 7(a)] 중앙 이미지처럼 안개가 많이 모여있는 곳에 공중에 떠 있는 차량으로 오인지가 발생함을 확인할 수 있다. 이 상황에 해당 알고리즘을 적용하였을 때, 오인지 됐던 안개를 인지하지 않으며, 왼쪽 차선에 존재하는 차량들은 여전히 정확하게 인지되는 결과를 [Fig. 7(a)] 오른쪽 이미지에서 확인할 수 있다.

청와대와 경복궁 주위에서 강수량이 많은 우천 상황 내 취득한 도심 도로데이터가 [Fig. 7(b)]이다. [Fig. 7(b)] 왼쪽 이미지에서 강수량이 많은 상황임을 알 수 있고, 왼쪽 차선에는 차



[Fig. 7] Qualitative results of Our algorithm. (a) Left: Images of urban roads near Gangneung in foggy conditions. Center: 3D Object Detection's misidentification situation using the single threshold where a significant amount of fog data appears as noise at similar height to LiDAR, leading to the misclassification of vehicles as suspended in the air. Right: The proposed algorithm improves the misidentification situation as depicted in (a)'s middle image. (b) Left: Images of urban roads near Cheongwadae in rain conditions. Center: 3D Object Detection's misidentification situation using a single threshold where a significant amount of rain data lead to the misclassification of vehicles and mistakenly identify bushes in the adjacent lane as vehicles. Right: The proposed algorithm improves the misidentification situation as depicted in (b)'s middle image



량들이 존재하지만 주행 중인 전방 차선 및 오른쪽 차선에는 차량이 존재하지 않는 것을 확인할 수 있다. [Fig. 7(b)] 중앙 이미지에서 주행 중인 전방 차선 내에 많은 양의 비에 반사된 포인트 클라우드들로 인해 차량으로 오인지가 발생하고, [Fig. 2(c)]와 동일하게 오른쪽에 존재하는 수풀들로 인한 오인지가 발생함을 확인할 수 있다. 이에 알고리즘을 적용하여 우천과 수풀에 의한 오인지를 개선하고 왼쪽 차선에 존재하는 차량은 동일하게 인지하는 결과를 [Fig. 7(c)]에서 확인할 수 있었다.

## 5. 결 론

본 논문에서 라이다 기반의 3차원 물체 탐지 결과를 센서 특성에 맞게 후처리하는 Adaptive thresholding 알고리즘을 도입하였다. 이 알고리즘은 3차원 물체 탐지 모델에 차차로부터의 거리에 기반하여 동적으로 임계값을 조정하는 후처리를 통해 오인지 결과를 감소시켜 모델의 강건성을 향상시켰음을 오픈 데이터셋에서의 기존 단일 임계값을 적용하였을 때와 비교한 실험을 통해 입증하였다. 또한, 본 연구에서 사용된 3차원 물체 탐지 모델에 국한되지 않고, 다양한 3차원 물체 탐지 모델에도 쉽게 적용 가능하며 효과적인 결과를 나타냄도 실험을 통해 확인하였다. 데이터셋에서의 실험만이 아니라, 실제 안개, 우천 등의 다양한 기상 환경과 장애물들의 변화가 많은 도심 도로 환경에서도 해당 알고리즘이 효과적임을 자체 데이터셋을 통해 확인하였다. 특히, 실제 도로 주행 시 발생하는 옆 차선의 수풀을 차량으로 잘못 인지하거나, 센서 특성으로 많은 양의 안개와 비로부터 포인트 클라우드들이 인식되어 차량으로 오인지는 문제를 개선하였음을 정성적으로 보여주었다. 이와 같은 결과를 통해, 본 알고리즘은 자율주행차량의 물체 탐지 강건성 향상을 달성하여 인지 고도화에 기여함을 보였고, 핵심 인지 기술인 3차원 물체 탐지 모델을 차량이 보다 안정적이고 효율적으로 주행할 수 있도록 하는 자율주행에 적합한 모델로 개선하였음을 보였다. 향후, 추가적인 성능 향상을 위해 연구를 진행한 모델로 얻은 파라미터들을 다른 모델들에 적용하는 것이 아닌 각 3차원 물체 탐지 모델의 결과를 활용한 파라미터를 각각 적용하여 성능을 향상시킬 수 있을 것이다. 또한, 라이다를 사용한 3차원 물체 탐지 모델에 대해서만이 아닌 카메라, 레이더 및 센서 퓨전을 사용한 3차원 물체 탐지 모델 및 다른 데이터셋을 사용하여 학습한 모델에 적용하기 위해 각 센서 특성, 모델을 학습한 데이터셋과 평가 결과에 대한 연구를 필요로 한다.

## References

- [1] K. Zhang, S. Tong, H. Shi, G. Yue, and J. Zhao, "Moving object detection of assembly components based on improved background subtraction algorithm," *IOP Conference Series: Materials Science and Engineering*, vol. 1009, 2021, DOI: 10.1088/1757-899X/1009/1/012063.
- [2] S. F. Lin and S. H. Huang, "Moving object detection from a moving stereo camera via depth information and visual odometry," *2018 IEEE International Conference on Applied System Invention (ICASI)*, Chiba, Japan, pp. 437-440, 2018, DOI: 10.1109/ICASI.2018.8394278.
- [3] 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.
- [4] G. Ross, "Fast r-cnn," *IEEE international conference on computer vision (ICCV)*, Santiago, Chile, pp. 1440-1448, 2015, DOI: 10.48550/arXiv.1504.08083.
- [5] Y. Yan, Y. Mao, and B. Li, "Second: Sparsely embedded convolutional detection," *Sensors*, vol. 18, no. 10, Oct., 2018, DOI: 10.3390/s18103337.
- [6] 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.
- [7] S. Shi, X. Wang, and H. Li, "PointRCNN: 3D Object Proposal Generation and Detection From Point Cloud," *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA, pp. 770-779, 2019, DOI: 10.1109/CVPR.2019.00086.
- [8] S. Shi, C. Guo, J. Yang, and H. Li, "PV-RCNN: Point-Voxel Feature Set Abstraction for 3D Object Detection," *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, USA, pp. 10526-10535, 2020, DOI: 10.1109/CVPR.42600.2020.01054.
- [9] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, and Q. Tian, "CenterNet: Keypoint Triplets for Object Detection," *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, Seoul, Korea (South), pp. 6568-6577, 2019, DOI: 10.1109/ICCV.2019.00667.
- [10] H. Kim and S. Park, "Monocular Camera based Real-Time Object Detection and Distance Estimation Using Deep Learning," *The Journal of Korea Robotics Society*, vol. 14, no. 4, pp. 357-362, Nov., 2019, DOI: 10.7746/jkros.2019.14.4.357.
- [11] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, pp. 779-788, 2016, DOI: 10.1109/CVPR.2016.91.
- [12] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich Feature

- Hierarchies for Accurate Object Detection and Semantic Segmentation,” *2014 IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, OH, USA, pp. 580-587, 2014, DOI: 10.1109/CVPR.2014.81.
- [13] 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.
- [14] T. Wang, X. Zhu, J. Pang, and D. Lin, “FCOS3D: Fully Convolutional One-Stage Monocular 3D Object Detection,” *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, Montreal, BC, Canada, pp. 913-922, 2021, DOI: 10.1109/ICCVW54120.2021.00107.
- [15] Z. Tian, C. Shen, H. Chen, and T. He, “FCOS: Fully Convolutional One-Stage Object Detection,” *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, Seoul, Korea (South), pp. 9626-9635, 2019, DOI: 10.1109/ICCV.2019.00972.
- [16] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, “End-to-end object detection with transformers,” *European conference on computer vision (ECCV)*, Glasgow, UK, pp. 213-229, 2020, DOI: 10.48550/arXiv.2005.12872.
- [17] Y. Wang, V. Guizilini, T. Zhang, Y. Wang, H. Zhao, and J. Solomon, “DETR3D: 3D Object Detection from Multi-view Images via 3D-to-2D Queries,” *2021 Conference on Robot Learning (CoRL)*, London, UK, DOI: 10.48550/arXiv.2110.06922.
- [18] Y. Liu, T. Wang, X. Zhang, and J. Sun, “Petr: Position embedding transformation for multi-view 3D object detection,” *European Conference on Computer Vision*, Tel Aviv, Israel, pp. 531-548, 2022, DOI: 10.48550/arXiv.2203.05625.
- [19] Z. Chen, Z. Li, S. Zhang, L. Fang, Q. Jiang, and F. Zhao, “Graph-DETR3D: rethinking overlapping regions for multi-view 3D object detection,” *The 30th ACM International Conference on Multimedia*, Lisboa, Portugal, pp. 5999-6008, 2022, DOI: 10.1145/3503161.3547859.
- [20] 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.
- [21] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas, “Frustum PointNets for 3D Object Detection from RGB-D Data,” *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, pp. 918-927, 2018, DOI: 10.1109/CVPR.2018.00102.
- [22] Z. Wang and K. Jia, “Frustum ConvNet: Sliding Frustums to Aggregate Local Point-Wise Features for Amodal 3D Object Detection,” *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Macau, China, pp. 1742-1749, 2019, DOI: 10.1109/IROS40897.2019.8968513.
- [23] R. Nabati and H. Qi, “CenterFusion: Center-based Radar and Camera Fusion for 3D Object Detection,” *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*, Waikoloa, HI, USA, pp. 1526-1535, 2021, DOI: 10.1109/WACV48630.2021.00157.
- [24] X. Chen, T. Zhang, Y. Wang, Y. Wang, and H. Zhao, “FUTR3D: A Unified Sensor Fusion Framework for 3D Detection,” *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Vancouver, BC, Canada, pp. 172-181, 2023, DOI: 10.1109/CVPRW59228.2023.00022.
- [25] X. Bai, Z. Hu, X. Zhu, Q. Huang, Y. Chen, H. Fu, and C. Tai, “TransFusion: Robust LiDAR-Camera Fusion for 3D Object Detection with Transformers,” *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, New Orleans, LA, USA, pp. 1080-1089, 2022, DOI: 10.1109/CVPR52688.2022.00116.
- [26] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” *arXiv:1706.03762*, 2017, DOI: 10.48550/arXiv.1706.03762.
- [27] Z. Liu, H. Tang, A. Amini, X. Yang, H. Mao, D. Rus, and S. Han, “BEVFusion: Multi-Task Multi-Sensor Fusion with Unified Bird’s-Eye View Representation,” *2023 IEEE International Conference on Robotics and Automation (ICRA)*, London, United Kingdom, pp. 2774-2781, 2023, DOI: 10.1109/ICRA48891.2023.10160968.
- [28] Y. Li, A. W. Yu, T. Meng, B. Caine, J. Ngiam, D. Peng, J. Shen, B. Wu, Y. Lu, D. Zhou, Q. V. Le, A. Yuille, and M. Tan, “DeepFusion: Lidar-Camera Deep Fusion for Multi-Modal 3D Object Detection,” *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, New Orleans, LA, USA, pp. 17161-17170, 2022, DOI: 10.1109/CVPR52688.2022.01667.
- [29] Y. Chen, Y. Li, X. Zhang, J. Sun, and J. Jia, “Focal Sparse Convolutional Networks for 3D Object Detection,” *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, New Orleans, LA, USA, pp. 5418-5427, 2022, DOI: 10.1109/CVPR52688.2022.00535.
- [30] S. Xu, D. Zhou, J. Fang, J. Yin, Z. Bin, and L. Zhang, “Fusion Painting: Multimodal Fusion with Adaptive Attention for 3D Object Detection,” *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, Indianapolis, IN, USA, pp. 3047-3054, 2021, DOI: 10.1109/ITSC48978.2021.9564951.
- [31] M. Liang, B. Yang, S. Wang, and R. Urtasun, “Deep continuous fusion for multi-sensor 3D object detection,” *The European conference on computer vision (ECCV)*, Munich, Germany, pp. 641-656, 2018, DOI: 10.48550/arXiv.2012.10992.
- [32] 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, The Netherlands, pp. 21-37, 2016, DOI: 10.1007/978-3-319-46448-0\_2.
- [33] 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.



### 이 은 호

2023 한양대학교 기계공학부(학사)  
2023~현재 서울대학교 협동과정 인공지능  
전공 석사과정

관심분야: 3D Object Detection, Sensor Fusion



### 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



### 정 민 우

2021 한국과학기술원 건설 및 환경공학과 학사  
2023 서울대학교 기계공학부 석사  
2023~현재 서울대학교 기계공학부 멀티  
스케일기계설계전공 박사과정

관심분야: Lidar SLAM, Multiple LiDAR, Place recognition



### 김 아 영

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

관심분야: 영상기반 SLAM



### 김 종 호

2019 홍익대학교 기계시스템디자인공학과  
(공학사)  
2019~현재 서울대학교 기계공학부 박사과정

관심분야: Lidar-based object perception, urban autonomous  
driving