



# 자율주행을 위한 3차원 라이다 인지 기술

장 원 제\*

## Abstract

최근 자율주행 자동차에 대한 이슈가 끊이질 않고 있다. 자율주행을 위해서 가장 중요한 기술을 고르자면 주변 환경을 인식하는 인지 기술을 꼽을 것이다. 주변 환경 인식을 위해 다양한 센서가 사용된다. 그 중에서도 3차원 라이다는 자율주행 차량을 연구하는 다양한 기관에서 활발하게 연구를 하고 있을 만큼 중요하게 쓰이는 센서이다. 최근 인공지능 기술의 발달로 방대한 양의 포인트 클라우드 데이터 처리가 가능해지면서, 앞으로 라이다 센서의 중요성은 더욱 커질 것으로 생각된다.

본 원고는 자율주행 자동차의 인지 시스템에 대해 정리하고, 주요 센서인 3차원 라이다의 인지 기술에 대한 주요 방법과 예시를 정리하고, 나아가서 자율주행의 상용화를 위해 넘어야 할 한계와 방향을 고찰하고자 한다.

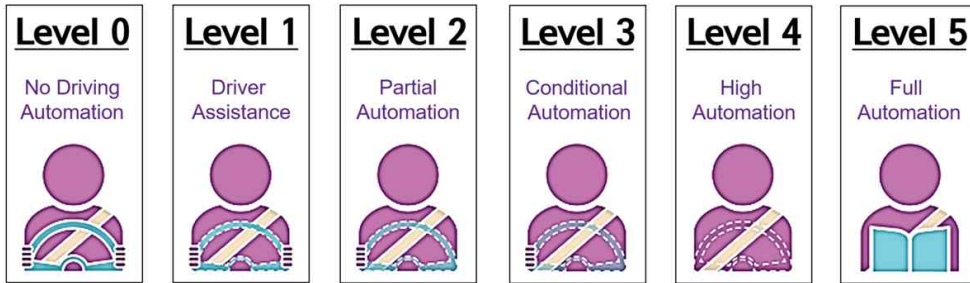
## I. 서론

자율주행 자동차라는 말은 예전부터 공상 과학에 빠짐없이 등장하던 소재이다. 공상 과학에서 생각만 하던 자율 주행 자동차가 최근 인공지능과 반도체의 발달에 힘입어 상용화 직전의 단계까지 와있다. 이미 해외에서는 자율 주행 택시가 운행을 시작했고, 자동차 회사들은 앞다투어 자율 주행 모드를 제공하는 서비스를 제공하고 있다. 이처럼 자율주행은 이미 일상생활에 스며들어 있지만, 자세한 내막까지 아는 이는 많지 않다.

자율주행이라 하는 기술은 다음 그림과 같이 총 6단계로 나뉘어서 단계별로 개발되고 있다.

\* 연세대학교 전기전자공학부 컴퓨터 지능 연구실 (Computational Intelligence Laboratory, CILAB), 자동차팀 팀장, 박사과정, jangwj1256@yonsei.ac.kr

그림 1 자율 주행 시스템의 6가지 단계.([1]의 Fig. 1.1)



Level 0은 어떠한 자동화 시스템도 없이 사람이 운전하는 수동 시스템을 말한다. Level 1 부터는 자동화 시스템의 개입이 발생한다. Level 1은 크루즈 컨트롤과 같은 기초적인 운전 보조 시스템을 말하고, 이 레벨에서는 운전자가 조향 및 제동거리 조절을 직접 해야 한다. Level 2는 발전된 운전 보조 시스템이다. Advanced Driver Assistance System(ADAS)이 Level 2에 속하는 기술이고, 이 단계부터는 시스템이 직접 조향 및 제동에 관여하기 시작한다. 하지만 운전자의 적극적인 개입이 필요한 단계이다. 현재 많은 차량 회사에서 제공하는 smart cruise control(SCC)과 lane keeping system(LKS)가 융합된 cruise system이 Level 2에 해당하는 기술이다. Level 3은 조건부 자동화 단계로, 이때부터는 판단의 주체가 운전자에서 시스템으로 넘어가기 시작한다. 자동차가 스스로 판단하여 전방의 장애물을 회피하거나 차로를 변경하는 간단한 판단을 내리고, 차량을 통제하는 단계가 시작된다. Level 4는 고도 자동화 단계로 대부분의 사람들이 자율주행의 최종 지점으로 생각하는 단계이다. Level 4에서는 운전자의 개입이 최소화되며, 대부분의 도로와 상황에서 가능하다. 하지만 운전자의 의지로 주행에 개입할 수 있기 때문에 완전한 의미의 자율주행은 아니다. Level 5는 완전 자동화 단계로 운전자의 개입이 완전히 불필요한 단계이다. 조향과 제동을 위한 장치가 무의미한 단계로 운전자 없이 탑승자만으로 목적지까지 주행이 가능한 단계이고, 시스템이 주행에 필요한 모든 판단을 수행하게 된다.

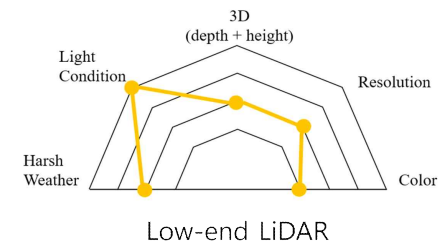
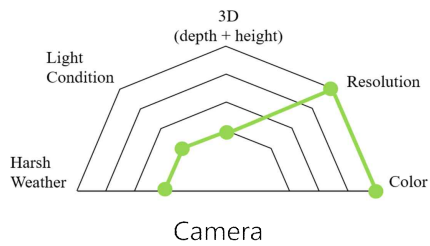
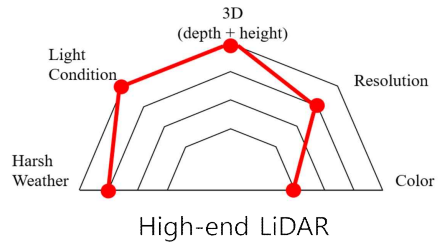
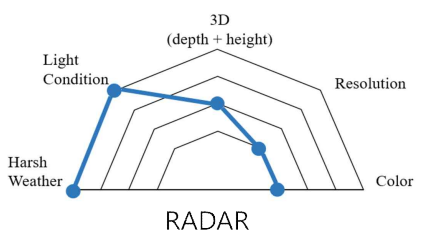
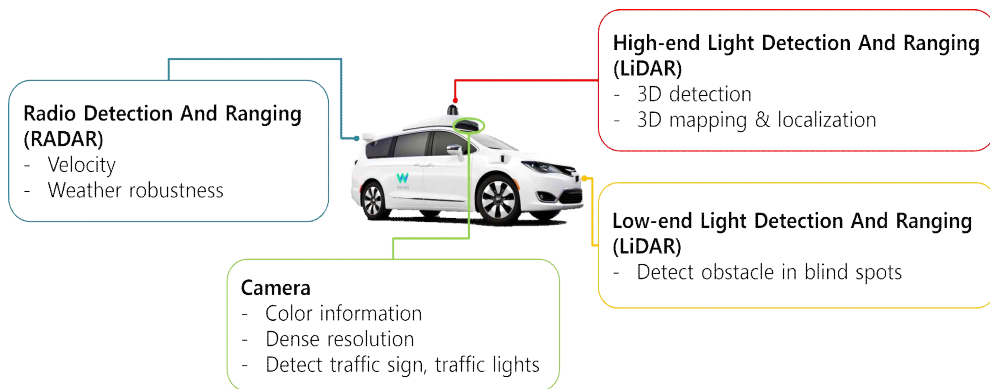
### 1. 자율 주행을 위한 센서 종류

자율주행 시스템에 있어서 가장 중요한 것은 센서이다. 시스템이 판단을 내리기 위해서는 주변 환경을 실시간으로 인식해서 올바른 정보를 전달할 수 있는 센서가 매우 중요하다. 자율주행 차량은 실외에서 밤낮이나 날씨에 상관 없이 주행이 가능해야 한다. 비가 오거나

밤에는 주행이 불가능한 자율차량이 있다면 그 차량은 시장에서 도태되고 말 것이다.

밤이나 흐린 날에는 상대적으로 빛이 없고, 사물을 판단하기 힘들다. 화창한 낮이라고 팬 찮은 것은 아니다. 실제로 2016년 발생한 테슬라의 사고 사례를 보면, 하얀색 대형 트레일러의 옆면에 반사된 햇빛 때문에 트레일러를 인지하지 못하고 충돌한 사고가 있었다. 이렇게 자율 차량을 위한 시스템은 굵은 날씨나 빛 상태에 구애받지 않고 사물을 인지해야 하기 때문에 센서의 타입이 매우 중요하다. 날씨나 조도만큼이나 시스템에 영향을 미치는 것이 센서의 가격이다. 아무리 성능이 뛰어나다고 하더라도, 가격이 비싸거나 충격에 약하면 자율차량용으로 쓰기는 어렵다.

그림 2 자율주행 차량에 쓰이는 다양한 센서 종류



자율 주행에 주로 사용되는 센서는 위 그림에 있듯이 카메라, 레이더(RADio Detection And Ranging), 라이다(Light Detection And Ranging) 센서 등이 있다. 그 외에도 초음파 센서 등이 있지만, 본문에서는 다루지 않았다. 카메라, 레이더, 라이다 이 세 가지 센서를 날씨, 조도, 거리와 높이를 포함한 3D 정보, 센서 분해각, 색깔 유무 이렇게 5가지 항목으로 비교를 해보았다. 레이더는 전파를 사용해서 반사되는 대상까지의 거리, 방향, 속도 등의 정보를 제공하는 센서이다. 레이더는 전파를 사용하기 때문에 날씨와 조도에 영향을 거의 받지 않고, 대상의 속도 정보를 알 수 있다는 장점이 있다. 이 때문에 많은 자동차 회사에서 차량 전면부에 레이더를 장착해서 ADAS 시스템을 위한 센서로 사용하고 있다. 하지만, 레이더는 센서의 분해각이 매우 작아서 물체의 유무 정도만 판단이 가능하고, 높이나 형상에 대한 정보는 전혀 알 수 없다는 단점이 존재한다.

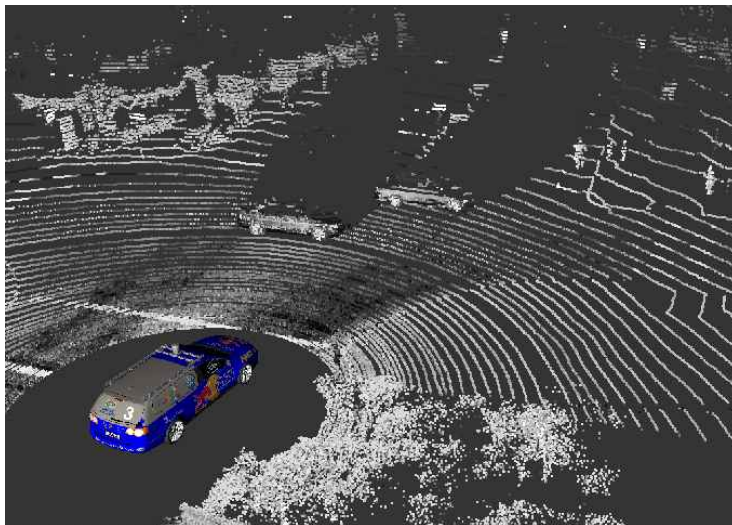
카메라는 센서 중 유일하게 색깔에 대한 정보를 제공할 수 있고, 위 그림에서 나열한 센서들 중에 가장 저렴하기 때문에 자율주행에 필수적인 센서이다. 특히, 신호등과 같이 색깔 정보가 꼭 필요한 상황이 존재하기 때문에 자율주행 차량과 카메라는 뗄 수 없는 관계이기도 하다. 테슬라의 경우는 카메라 센서만 이용해서 2단계를 넘어서 3단계 자율주행을 바라보고 있을 만큼 카메라는 자율주행과는 밀접한 센서이다. 하지만, 카메라는 날씨나 조도에 매우 민감하다. 갑자기 터널에 들어간다거나, 터널을 빠져나오는 경우와 같이 조도가 급격하게 변하는 경우나, 가로등이 존재하지 않는 어두운 시골길 같은 경우에는 카메라가 아무 역할을 하지 못한다. 또한 카메라만 단독으로 활용하기 어려운 큰 이유는 거리 정보를 제공하지 못한다는 것이다. 자율 차량이 대상을 관측하는 것에서 끝나는 것이 아니라 회피 주행을 수행하거나 제동을 하기 위해서는 대상까지의 거리가 필요하다. 카메라는 이런 거리정보를 제공하지 못한다. Dual camera를 통해서 이를 해결하고자 하는 노력은 있지만, 20m 밖에 존재하는 물체까지의 거리는 잘 파악하지 못한다. 이는 자율 차량 시스템에 있어서 큰 걸림돌이 된다.

라이다 센서는 레이더와 카메라의 중간 지점에 있는 센서이다. 조도에 영향을 받지 않고, 날씨에도 비교적 덜 민감하게 반응하면서도 물체의 형상이나 거리에 대한 정보를 완벽하게 제공할 수 있다. 라이다 센서의 가장 큰 단점으로 꼽혔던 것이 센서의 가격 문제와 미관 문제였는데, 이는 라이다 기술의 발전으로 점점 저렴해지고, 소형화 된 라이다 센서가 등장하면서 어느 정도 해결이 되고 있다. 특히, 고성능 라이다, 즉 3차원 라이다는 높은 센서 분해각과 완벽한 3차원 정보를 제공할 수 있기 때문에 미래 가치를 두고 활발하게 연구가 되고 있다.

본문에서는 3차원 라이다 센서의 데이터가 어떤 형태로 이루어져 있고, 인공지능이 라이다 센서에 어떻게 적용될 수 있는지를 살펴볼 예정이다. 또한 자율주행에서 라이다 센서를 사용한 분야 및 사례는 어떤 것들이 있는지 자세히 살펴보고자 한다.

## II. 라이다 센서와 인공지능

그림 3 3차원 라이다 센서의 데이터 예시([2]의 Fig. 1.)



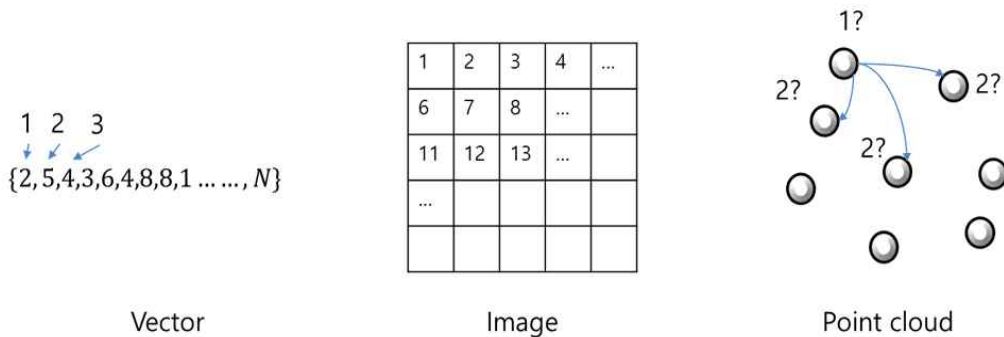
사람이 보는 것과 매우 비슷한 형태의 데이터를 출력하는 카메라와 달리 라이다 센서는 카메라에 비해서 센서 데이터가 직관적이지 않다. 또한, 방대한 크기의 데이터 때문에 인공지능 기술이 충분히 발전하기 전에는 라이다에서 인공지능을 사용한 연구를 찾기 어려웠다. 하지만 2017년도부터 라이다 센서의 데이터인 포인트 클라우드에서 인공지능을 사용하는 연구가 활발하게 진행되기 시작했고, 최근에는 주요 컴퓨터 비전 학회에서도 포인트 클라우드를 이용한 연구가 높은 비율을 차지할 만큼 많은 연구가 진행되고 있다. 다음 챕터부터 거리 정보가 포함된 데이터를 인공지능 알고리즘에 이용하기 위해서 어떠한 과정을 거쳐야 하는지, 이런 형태의 데이터를 이용해서 어떤 문제를 해결할 수 있고, 자율주행에 어떻게 사용될 수 있는지에 대해서 알아보려고 한다.

## 1. 라이다 센서 데이터: Point Cloud란

라이다는 빛을 쏘서 반사되는 시간을 측정하여 대상의 위치 좌표를 측정하는 거리 센서이다. 이렇게 라이다를 통해서 얻은 데이터의 예시가 위 그림에 나타나 있다. 이 데이터는 빛이 쏘아진 각도와 거리 값을 구면 좌표계로 제공하고, 이를 데카르트 좌표계(Cartesian coordinate)으로 투영해서 사용한다. 즉, 라이다 데이터는  $x,y,z$  좌표계를 갖는 점들의 집합 형태의 데이터를 제공하고, 이를 포인트 클라우드(Point Cloud) 혹은 점군이라고 부른다. 이 포인트 클라우드 데이터에서 인공지능을 활용하기 어렵게 만드는 몇 가지 특징이 있고, 그 중 가장 큰 특징은 데이터의 순서가 정해져있지 않다는 것이다.

데이터의 순서가 정해져 있지 않다는 것이 어떤 의미인지 쉽게 와닿지 않을 것이다. 이것은 아래 그림을 보면 쉽게 이해가 될 것이다.

**그림 4** 1차원 벡터와 2차원 이미지, 3차원 포인트 클라우드 데이터의 특징



1차원 벡터 데이터나 2차원 이미지 데이터는 데이터의 순서가 확실하다. 위 그림에서 벡터에서 3번째 값이 어떤 값인지를 묻는다면 대부분의 사람들이 “4”라고 대답할 것이다. 이미지에서도 마찬가지다. 픽셀 “7”의 주위에 있는 픽셀 값을 고르자고 한다면, 대부분의 사람들이 {2,6,8,12} 혹은 {1,2,3,6,8,11,12,13} 이라고 대답할 것이다. 이것은 데이터의 순서가 확실하게 정해져 있기 때문이다. 하지만 포인트 클라우드에서 첫 번째 점을 고르라고 하면 사람마다 다른 값을 고를 것이다. 누구는 가장 위에 있는 점을 첫 번째로 고를 수도 있고, 누구는 가장 왼쪽에, 혹은 가장 아래에 있는 점을 첫 번째 점으로 고를 수도 있다. 예를 들어 첫 번째 점을 가장 위에 있는 점으로 정의하더라도 그 다음이 문제가 된다. 두 번째 점을 가장 가까운 점으로 할 것인지, 아니면 첫 번째 점을 정한 것처럼 두 번째로 위에 있는

점이 되어야 할지 순서가 정해져 있지 않다. 이런 특징은 포인트 클라우드 데이터를 인공지능에 적용하기 어렵게 만들었다.

또 다른 특징으로는 포인트 클라우드는 매 순간 입력되는 센서의 입력 크기가 정해져 있지 않다. 카메라의 경우는 화소에 따라서 이미지의 크기가 정해지지만, 라이다의 경우는 매 순간 다른 개수의 점들이 센서로부터 제공된다.

이렇게 매 순간 다른 크기의 포인트 클라우드가 입력되고, 순서도 정해져 있지 않은 데이터를 처리하기 위해서 예전부터 다양한 방법으로 3차원 데이터를 처리해서 사용했었다. 다음 챕터부터 인공지능의 발달 이전부터 사용하던 방법과 인공지능을 사용한 방법에 대해 소개하도록 하겠다.

## 2. Point Cloud 처리 기법

포인트 클라우드를 처리한다는 것은 점들의 집합에서부터 유의미한 정보(feature)들을 뽑아주는 과정을 말한다. 어떤 형상의 포인트 클라우드가 주어질 때 그 사물의 종류를 맞추거나(classification), 운전 상황과 같은 넓은 상황에서 필요한 물체를 찾거나(detection), 혹은 의미적으로 연결되는 포인트끼리 같은 물체로 묶어주는(semantic segmentation) 등의 문제를 풀기 위해서는 입력된 포인트 클라우드로부터 알맞은 feature를 잘 추출하는 것이 중요하다. 인공지능이 활발하게 연구되기 이전부터도 포인트 클라우드를 처리해서 feature를 추출하고자 하는 연구들이 많이 있었고, 인공지능 즉, 딥러닝의 발달에 힘입어 딥러닝을 활용해서 포인트 클라우드로부터 feature를 추출하는 연구들이 지속적으로 등장하고 있다. 이번 챕터에서는 딥러닝 이전과 이후 포인트 클라우드를 처리하는 방법에 대해 자세히 알아보려고 한다.

### 2-1. 딥러닝 발전 이전 처리 방법들

딥러닝의 발전 이전에는 입력된 포인트 클라우드로부터 할 수 있는 것이 한정적이었고, 주로 자율주행 시 충돌할 가능성이 있는 물체를 찾는 연구를 진행했다. 여기서 충돌할 가능성이 있는 물체란 자동차, 보행자, 이륜차 등을 말한다. 포인트 클라우드가 주어졌을 때 가장 쉽게 생각할 수 있는 것은 그 포인트 클라우드 내에 존재하는 포인트의 개수를 사용하는 것이다 [3]. 하지만 포인트의 개수만으로는 정보가 매우 제한적이기 때문에 단독으로 사용하지는 못하고 추가적으로 다른 정보들을 사용해야 한다. 보편적으로 많이 쓰이는 것은 개수만큼이나 쉽게 생각할 수 있는 박스 정보를 활용하는 것이다 [2][4]. 포인트 클라우

드를 포함하고 있는 Contour box를 계산하고, 그 박스의 길이, 너비, 높이 정보를 사용한다. 이런 정보들은 보행자와 같이 작은 물체와 차량과 같이 비교적 큰 물체를 구분할 수 있게 해주는 지표가 되며, 건물의 외벽이나 가로수 같이 높이가 높은 장애물들과 사물을 구분할 수 있게 해주는 지표가 된다. 또한, 많이 쓰이는 특징으로는 포인트 클라우드를  $x,y$  평면 혹은  $x,z$  평면에 투영시켜서 만든 이미지를 이용해서 만든 histogram이 있다 [2],[5]. 혹은 포인트 클라우드의 주성분 분석(Principal Component Analysis(PCA)) 값을 사용하기도 한다 [3]. 이를 위해 covariance matrix를 구해서 원소값을 사용하거나, eigenvalue나 eigenvector 값을 활용하기도 하는 등 다양한 방법을 통해서 포인트 클라우드로부터 유의미한 정보들을 추출하고자 한다.

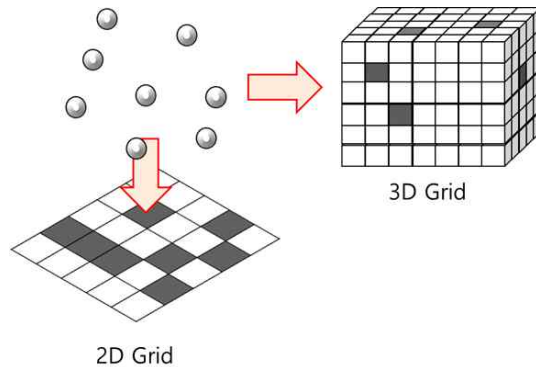
이런 방법들은 매우 빠르고 직관적으로 어떤 특징들을 추출했는지 이해가 되지만, 아주 많은 종류의 물체를 구분하거나 할 때는 도움이 되지 못한다.

## 2-2. 딥러닝 발전 이후 처리 방법들

인공지능이 활발하게 발달하면서 포인트 클라우드를 딥러닝으로 처리해보고자 하는 노력들이 있었다. 하지만 이들이 가장 크게 직면한 것이 포인트 클라우드의 순서가 없다는 특징과 강체 운동(Rigid Body Motion)에 상관없이 같은 값을 출력해야 한다는 것이었다. 이는 당연하게도 차량에 해당하는 포인트 클라우드는 어떤 방향을 바라보고 있던 상관 없이 차량이라는 클래스를 맞춰야 한다. 하지만, 순서가 없는 포인트 클라우드에서는 어떤 특정 기준을 잡아서 순서를 정해주는 순간 회전이나 변환할 때마다 다른 기준을 갖게 된다.

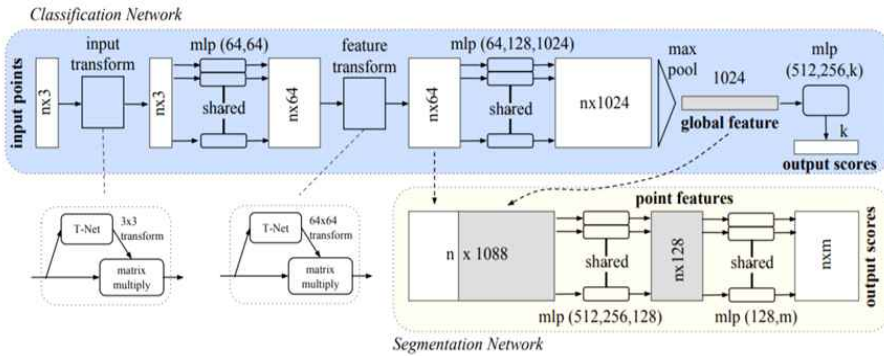
가장 쉽게 생각할 수 있는 방법은 포인트 클라우드를 2D 형태로 만들거나 순서가 있는 3차원으로 만들어버리는 것이다. 이렇게 포인트 클라우드의 포맷을 변환하면 이미지에서 사용하는 2차원 CNN이나 3차원 CNN 연산을 수행할 수 있게 된다. 하지만 이것은 각 좌표 평면에 투영하는 projection 과정도 필요하고, 진정한 의미로 포인트 클라우드를 처리한다고 보기 어렵다.

그림 5 포인트 클라우드를 2D grid나 3D grid 형태로 사용하는 방법의 예시



이때 포인트 클라우드에서 딥러닝의 위상을 바꿔놓는 연구가 등장했다. PointNet [6] 이라는 연구는 포인트 클라우드의 순서를 알 수 없으면 순서가 상관없이 같은 값이 나오는 연산을 이용하면 된다는 것이다. 포인트 클라우드에 존재하는 각 포인트 하나하나를  $1 \times 3$  벡터로 생각을 해주었고, 같은 학습 파라미터를 공유하는 MLP를 모든 포인트에 통과시켜 준다. 이후 나오는 값에 대해서는 Max 값을 사용하면, 포인트가 어떤 순서로 배열되어있든 상관 없이 같은 값이 나오게 된다. 또한, T-Net 이라는 구조를 제안했고, 이 구조를 통해서 입력 포인트 클라우드를 한 방향으로 정렬해준다. 이렇게 T-Net과 최댓값을 사용하는 방법을 통해서 PointNet은 포인트 클라우드에서 강제운동에 강인하면서도 순서에 상관 없는 특징을 추출해줄 수 있다. 아래 그림을 보면,  $N \times 3$  의 포인트 클라우드가 입력되고, 이를 input transform 과정을 거쳐서 강제 운동에 강인한 형태의 포인트 클라우드로 변환해 준 다음, 최종적으로 Max pooling 과정을 통해서 순서에 상관없는 global feature를 추출해 준다. 이 PointNet의 등장으로 인해서 포인트 클라우드에서 딥러닝을 활용하는 연구가 폭발적으로 증가하게 되었고, 이후 대부분의 연구가 이 PointNet의 아이디어를 기초로 삼아서 연구를 하고 있다.

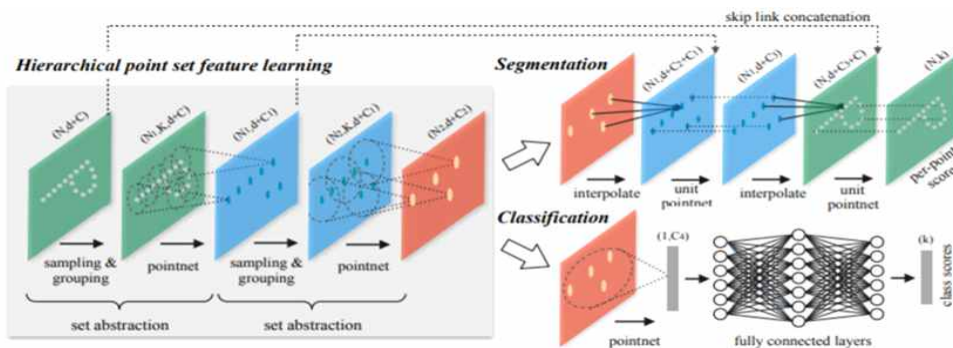
그림 6 딥러닝을 이용한 PointNet의 구조 그림.([6]의 Fig. 2.)



하지만, PointNet의 가장 큰 장점이자 단점은 최댓값을 취하는 것이다. 최댓값의 가장 큰 문제점은 실제 포인트의 위치 정보가 다 사라져 버린다는 것이다. 순서에 상관없는 연산을 했지만, 그렇기 때문에 최댓값으로 구해진 어떤 값이 어떤 위치에 있는 포인트에서 나온 값인지를 알 수 없어진 것이다.

이후 후속 연구인 PointNet++ [7]에서 이 문제를 해결하고자 했다. 전체 포인트 클라우드를 한 번에 max 값을 취해서 지역 정보들을 다 없애는 것이 아니라 포인트 클라우드를 여러 개의 작은 그룹으로 나눠서 그룹마다 max 값을 취하는 방식을 사용했다. 이러면 추출된 최댓값이 적어도 어떤 그룹에 있는 포인트에서 나온 값인지는 알 수 있게 되고, 어느 정도 세부적인 지역정보를 활용할 수 있게 된다. 아래 그림을 보면, Set abstraction 부분에서 작은 그룹으로 grouping을 하고, 각 그룹마다 PointNet 연산, 즉 최댓값 추출을 하는 것을 볼 수 있다.

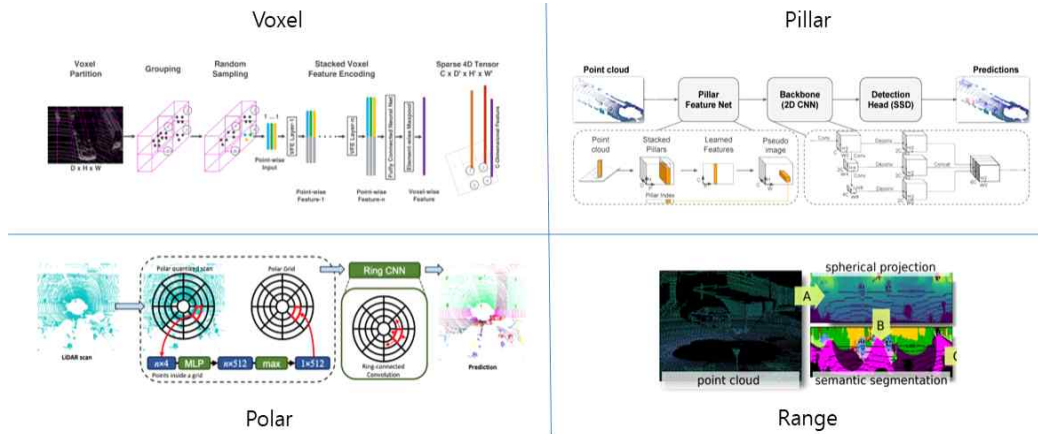
그림 7 PointNet을 개선한 PointNet++의 구조 그림.([7]의 Fig. 2.)



이후로도 포인트의 특징을 어떻게 잘 추출할 지, 각 포인트의 좌표값을 어떻게 잘 이용해 줄지를 고민하는 방향으로 딥러닝을 이용해서 포인트 클라우드를 처리하는 연구가 지속되고 있다. 특정 포인트에서 거리상 가까운 점들을 구한 뒤 좌표값과 그 차이를 이용해서 그래프를 만든 뒤 업데이트 하는 그래프 기반의 방법들[8]이나 특정 포인트를 중심으로 하는 3차원 커널을 만들어서 그 커널 안에 들어오는 포인트 정보를 업데이트하는 커널 기반의 방법들[9]이 계속해서 등장하고 있다. 하지만 이들의 공통적인 점은 PointNet에서 제안한 MLP + Max 구조를 기본으로 사용해서 포인트 클라우드의 특징을 추출하고 업데이트한다는 것이다. 그래프 기반의 방법들도 그래프로 연결된 포인트의 정보를 업데이트 할 때는 MLP를 사용하고 있고, 커널 기반의 방법들도 커널 연산을 수행할 때는 MLP를 통해 3차원 커널의 가중치를 정해준다.

이렇게 포인트 클라우드를 직접 가공하지 않고 처리해서 특징을 추출하는 방법이 있는가 하면, 위에서 한 번 언급했듯이 2차원 격자평면이나 3차원 복셀에 투영해서 처리하는 방법들도 꾸준히 연구가 되고 있다. 특히 자율주행을 위한 데이터는 넓은 영역의 데이터를 처리해야 하기 때문에 입력 포인트 클라우드를 가공 없이 처리하기 위해서는 너무 많은 연산이 들어가게 된다. 이를 방지하기 위해서 입력 포인트의 개수를 줄이는 방법 중 하나로 2차원 혹은 3차원 격자에 투영하는 방식을 활용하고 있다. 이런 연구들은 어떤 형태의 격자에 투영하는지에 따라서 Voxel 기반의 방법 [10], Pillar 기반의 방법 [11], Polar 기반의 방법 [12], Range 기반의 방법[13] 등으로 나뉜다. 이 방법들의 예시가 다음 그림에 있다.

**그림 8** 포인트 클라우드를 격자에 투영해서 처리하는 방법들.([10]의 Fig. 2., [11]의 Fig. 2., [12]의 Fig. 2., [13]의 Fig. 2.)



이처럼 딥러닝의 발달로 인해 포인트 클라우드를 처리해서 사용하는 기술도 활발하게 연구되고 있고, 포인트 클라우드 처리 기술의 발전에 힘입어 라이다 센서에 대한 관심도도 올라가고 있다.

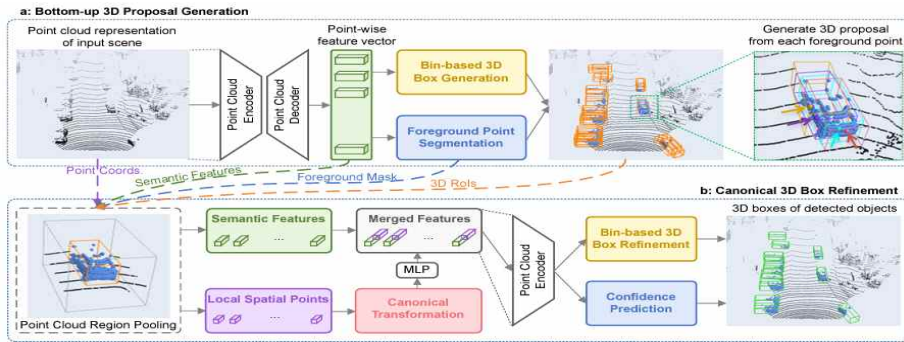
### 3. 자율주행을 위한 Point Cloud 처리 기술 응용

실제 차량에 설치된 라이다 센서를 이용해서 데이터를 취득하고, 해당 데이터에서 자율주행에 필요한 인지 시스템을 개발하는 연구가 많이 진행되고 있다. 이번 챕터에서는 실제 자율주행 시스템에 적용 가능한 포인트 클라우드 처리 방법의 응용화 단계에 대해 정리해보았다.

#### 3-1. Detection

환경 인지 시스템의 가장 기초적인 문제가 되는 검출 문제를 해결한 연구부터 알아보자. 라이다 센서로부터 제공된 데이터를 처리하기 위해서는 보통 2단계를 거친다. 먼저 원본 데이터를 처리해서 검출 대상일 가능성이 높은 후보군들을 골라주고(proposal generation), 이 후보군들의 클래스를 맞춰주는 식(refinement)으로 검출 문제를 해결한다 [14]. 입력 포인트 클라우드들에서 후보군에 해당하는 박스들을 구해주고, 위 챕터에서 설명한 포인트 클라우드 처리 기술을 사용해서 박스들 내부에 존재하는 포인트들의 특징들을 추출해준다. 이렇게 추출해준 특징들을 이용해서 미리 구해준 박스의 크기를 보정하거나, 후보군의 클래스가 보행자인지 차량인지 등을 맞춰준다. 최근에는 빠르면서도 90%에 가까운 average precision을 갖는 높은 성능을 내는 연구들이 많이 진행되고 있다.

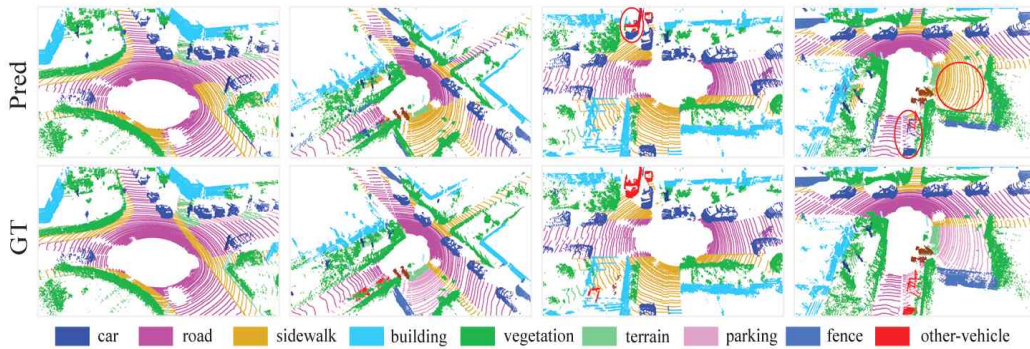
그림 9 포인트 클라우드 처리 기술을 이용한 검출 방법의 예시. proposal generation과 refinement 단계로 나뉘어져 있다.([14]의 Fig. 2.)



### 3-2. Semantic Segmentation

시멘틱 세그멘테이션 문제는 이미지에서부터 활발하게 진행되던 연구 분야이다. 같은 물체에서 나온 포인트끼리 하나로 묶어주는 문제를 해결하는 것으로, 검출 문제보다 훨씬 다양한 클래스의 물체가 존재하며 박스 단위가 아닌 포인트 단위로 물체를 찾아줘야 하는, 훨씬 난이도 있는 문제이다 [15]. 이 시멘틱 세그멘테이션 문제를 해결한 예시가 다음 그림에 나타나 있다.

그림 10 포인트 클라우드 시멘틱 세그멘테이션 기술의 결과 예시.([16]의 Fig. 6.)



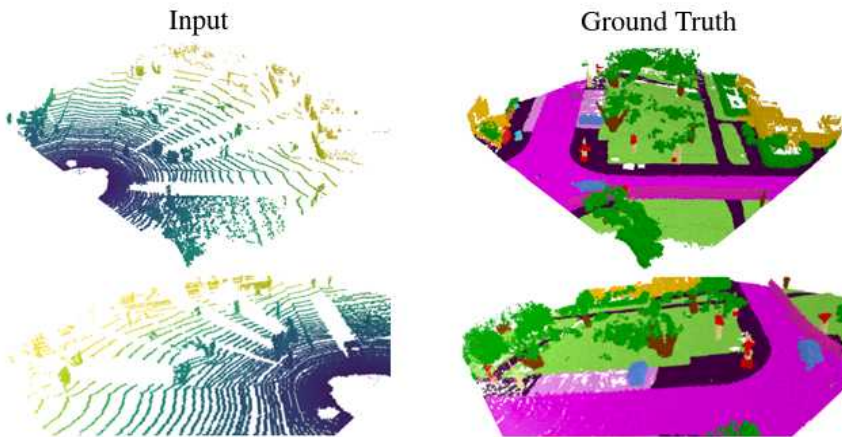
이 시멘틱 세그멘테이션 문제를 해결하기 위해서는 입력 포인트 클라우드의 세부적인 지역 정보를 얼마나 잘 추출할 수 있는지와 얼마나 빠르게 추출할 수 있는지가 중요한 관점이다. 이를 위해서 지역 정보를 따로 인코딩 해주거나 그래프를 이용해서 포인트 간의 연

관관계를 중점적으로 업데이트하는 식으로 빠르면서도 풍부한 정보를 추출하기 위한 연구가 계속되고 있다.

### 3-3. Scene Completion

포인트 클라우드를 이용한 3차원 데이터에서 최근에 활발하게 다뤄지고 있는 문제가 Scene completion 문제이다. 이 문제는 주어진 입력 데이터에서 비어있는 공간을 예측해서 채워 넣는 것으로, 입력 데이터의 클래스를 맞추는 시멘틱 세그멘테이션에서 조금 더 나아간 문제이다. 이 문제를 해결하기 위해서는 특정 포인트의 정보를 잘 추출하는 것을 넘어서 주변 정보를 잘 가져와서 빈 공간의 정보를 생성해내야 한다. 이제 막 시작 단계인 이 연구가 활발하게 진행되고 성능이 좋아진다면, 바로 앞 차량에 가려져서 보이지 않는 그 뒤도 예측할 수 있고, 너무 멀어서 라이다의 포인트가 희박한 곳까지도 인지 가능한 범위에 넣을 수 있게 될 것이다.

**그림 11** Scene Completion 기술의 결과 예시. 비어있는 공간의 클래스까지 예측해야 한다. ([17]의 Fig. 4.)



## III. 현재 자율주행 기술의 한계

포인트 클라우드를 처리하는 기술은 나날이 발전하고 있고, 다른 센서를 사용하는 연구도 마찬가지로 빠르게 발전하고 있다. 기술의 발전만으로 보면 4단계를 넘어서 5단계 자율주

행이 보편화될 날이 머지않아 보이기도 하지만, 실제 상용화 차량의 사례를 보면 3단계는 커녕 2단계 자율주행만 옵션화 돼서 나오고 있다. 이 간극은 왜 생기는 것일까. 이번 챕터에서는 완전 자율주행의 상용화까지 어떤 문제가 남아있고, 기술적인 한계는 어느 정도인지에 대해 알아보려고 한다.

## 1. 사고 영상 획득의 어려움

인공지능 알고리즘의 성능은 학습 데이터의 수에 절대적으로 비례할 수밖에 없다. 자율주행을 위한 데이터는 일반 차량으로 시내를 주행하거나 고속도로를 주행하면서 충분히 얻을 수 있지만, 특히 케이스에 대한 데이터는 얻기가 힘들다. 예를 들어 도로 한복판에 누워있는 트럭이라든지, 빙판길에서 주행한 데이터 같은 경우는 얻기 어려운 경우가 많다. 하지만, 자율주행 차량의 사고는 이런 상황에서 발생한다. 기존 데이터에서 학습하지 못한 데이터가 등장할 때 인공지능 시스템은 오류를 일으키고, 잘못된 판단을 내린다. 그리고 이런 잘못된 판단이 탑승자의 목숨과도 직결될 수 있을 만큼 큰 사고를 낼 수도 있다. 따라서 최대한 다양한 상황에서 데이터를 취득하고 학습을 하고 대비를 해야 하는데, 도로 상황이라는 것이 무한대에 가까운 경우의 수를 갖고 있기 때문에 모든 상황을 학습한다는 것은 불가능에 가깝고, 특히 모든 사고에 대한 데이터를 취득하고 학습한다는 것은 매우 어려운 일이 될 것이다.

## 2. 인공지능에 대한 신뢰도

현재 소비자들에게 자율주행차가 안전한지에 대한 설문은 한 결과를 보면, 대부분의 나라에서 절반 혹은 그 이상의 사람들이 자율주행 차의 안전을 믿지 못한다고 답한다. 이것은 비행 공포증이 있는 것과도 비슷한 맥락으로 이해할 수 있다. 직접 운전을 하는 상황에서는 자신이 상황을 통제하고 있지만, 비행기를 타고 있는 상황에서는 자신이 통제할 수 없는 상황에 놓여있기 때문에 그 상황을 공포스러운 상황으로 느끼는 것이다. 자율주행 차량도 마찬가지다. 자신이 통제하지 않는 차량을 믿지 못하는 것이다.

또한, 인공지능 자체의 한계도 명확하다. 엄청난 양의 데이터를 사용해서 학습을 해도 100%의 성능은 달성하지 못한다. 특히 학습되는 파라미터들의 의미를 알기 어려운 만큼 사람의 개입이 있어도 인공지능의 성능을 끌어올릴 수 있다고 장담할 수는 없다. 완벽에 가까운 시스템이지만, 완벽하지 않기 때문에 완전한 신뢰를 받지 못하고 있고, 이는 자율주행

시스템의 보급화에 큰 장애물이 되고 있다.

### 3. 사고에 대한 책임

가장 큰 문제는 사고가 난 이후에 발생한다. 자율주행 차량을 이용하다가 사고가 나게 되었다면, 그 사고의 책임을 누구에게 부여해야 하는지가 모호하다. 인공지능 시스템을 개발한 연구진인가, 인공지능 시스템을 장착한 차량을 판매한 자동차 회사인가, 인공지능 차량을 탑승한 탑승자인가. 사고가 나면 누군가는 책임을 져야 하지만, 그 책임을 지는 사람이 진짜 그 사고의 모든 책임이 있는지는 확답할 수 없을 것이다. 이런 사고와 관련된 이슈가 끊임없이 발생하고 있고, 해결책이 나오지 않았기 때문에 4단계 혹은 5단계 자율주행 차량의 상용화는 시기상조일 수밖에 없다.

## IV. 결 론

인공지능의 발달로 인해 자율주행을 위한 인지 능력은 특정 부분에 있어서는 사람보다도 나은 면을 보여주고 있다. 인간의 가시거리보다 훨씬 멀리 떨어진 물체도 검출이 가능하고, 인간의 판단이나 처리 속도보다 월등히 빠른 속도로 문제를 해결하는 모습도 볼 수 있다. 하지만 위에서도 언급했듯이 기술의 발전만으로는 한계가 명확하게 존재한다. 특히 사용자의 목숨 혹은 타인의 목숨과도 직결될 수 있는 사고가 발생할 가능성이 존재하는 한 자율주행 차량의 상용화는 요원한 것처럼 보이기도 한다. 사고가 절대 일어나지 않는 인공지능 기술을 개발할 수 있다면 좋겠지만, 그것은 불가능에 가깝기 때문에 사고가 일어났을 때 그 책임을 판단할 수 있는 법적인 제도가 생겨야 할 것이다.

가장 현실적인 가능성은 개인 차량이 아닌 버스나 택시 같은 대중교통부터 자율주행이 보급되는 것이다. 실제로도 상암 DMC 지역에서 운행하는 자율주행 버스나 구글의 자회사인 Waymo에서 서비스를 개시한 로보택시 같은 서비스는 상용화를 앞두고 있다. 이런 경우는 사고의 책임을 서비스를 제공하는 회사가 전적으로 부담하는 식으로 책임자의 부재 문제를 해결할 수 있다. 또한, 화물차와 같이 노동력이 매우 많이 필요한 경우도 자율주행 시스템이 보급되기 좋은 요건이다. 이렇게 주변에서 자율주행 차량을 접하는 경우가 많아지고, 대중교통부터 차츰 일상에 녹아들기 시작하면 언젠가는 너도나도 자율주행 차량을 타고 다니는 시대가 도래할 것이다.

자율주행 차량은 언젠가는 상용화가 될 것이다. 다만 그것이 많이 멀지 않은 가까운 미래 이길 바랄 뿐이다.

## 참고문헌

- [1] 안종현. "Novel moving vehicle detection and vehicle bounding box tracking using a low-end 3D laser scanner." 국내박사학위논문 Graduate School, Yonsei University, 2020. 서울
- [2] A. Teichman, J. Levinson and S. Thrun, "Towards 3D Object Recognition via Classification of Arbitrary Object Tracks," in Proc. of ICRA, 2011.
- [3] Z. Yan, T. Duckett and N. Bellotto, "Online Learning for Human Classification in 3D LiDAR-based Tracking," in Proc. of IROS, 2017.
- [4] J. Cheng, Z. Xiang, T. Cao and J. Liu, "Robust Vehicle Detection using 3D Lidar under Complex Urban Environment," in Proc. of ICRA, 2014.
- [5] H. Wang, B. Wang, B. Liu, X. Meng and G. Yang, "Pedestrian recognition and tracking using 3D LiDAR for autonomous vehicle," Robot. Auton. Syst., 2017.
- [6] C. Qi, H. Su, K. Mo and L. Guibas, "PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation," in Proc. of CVPR, 2017.
- [7] C. Qi, L. Yi, H. Su and L. Guibas, "PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space," in Proc. of NeurIPS, 2017.
- [8] Y. Wang, Y. Sun, Z. Liu, S. Sarma, M. Bronstein and J. Solomon, "Dynamic Graph CNN for Learning on Point Clouds," ACM Trans. Graph., vol. 38, no. 5, Oct, 2019.
- [9] H. Thomas, C. Qi, J. Deschaud, B. Marcotegui, F. Goulette and L. Guibas, "KPConv: Flexible and Deformable Convolution for Point Clouds," in Proc. of ICCV, 2019.
- [10] Y. Zhou and O. Tuzel, "VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection," in Proc. of CVPR, 2018.
- [11] A. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang and O. Beijbom, "PointPillars: Fast Encoders for Object Detection From Point Clouds," in Proc. of CVPR, 2019.
- [12] Y. Zhang, Z. Zhou, P. David, X. Yue, Z. Xi, B. Gong and H. Foroosh, "PolarNet: An Improved Grid Representation for Online LiDAR Point Clouds Semantic Segmentation," in Proc. of CVPR, 2020.
- [13] A. Milioto, I. Vizzo, J. Behley and C. Stachniss, "RangeNet++: Fast and Accurate LiDAR Semantic Segmentation," in Proc. of IROS, 2019.
- [14] S. Shi, X. Wang and H. Li, "PointRCNN: 3D Object Proposal Generation and Detection From Point Cloud," in Proc. of CVPR 2019.

- [15] 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," in Proc. of ICCV, 2019.
- [16] Q. Hu, B. Yang, L. Xie, S. Rosa, Y. Guo, Z. Wang, N. Trigoni and A. Markham, "RandLA-Net: Efficient Semantic Segmentation of LargeScale Point Clouds," in Proc. of CVPR, 2020.
- [17] L. Rodóo, R. Charette and A. V.-Blondet, "LMSCNet: Lightweight Multiscale 3D Semantic Completion," in Proc. of 3DV, 2020.