

Lidar를 사용한 Multiple Object Tracking

성균관대학교
기계공학부
차량시스템연구실

이주현 2018314852
엄동희 2018310465
김성준 2018310755

Table of Contents

I . Introduction

II. Method

- Detection

1. Data Processing

2. Bounding Box

- Tracking

1. Data Association

2. Point Tracking

3. Box Tracking

III. Result

IV. Discussion

V . References

1. Introduction 연구 주제 선정

기존 정부과제

목표도심 주행 버스의 자율주행 기반기술 개발

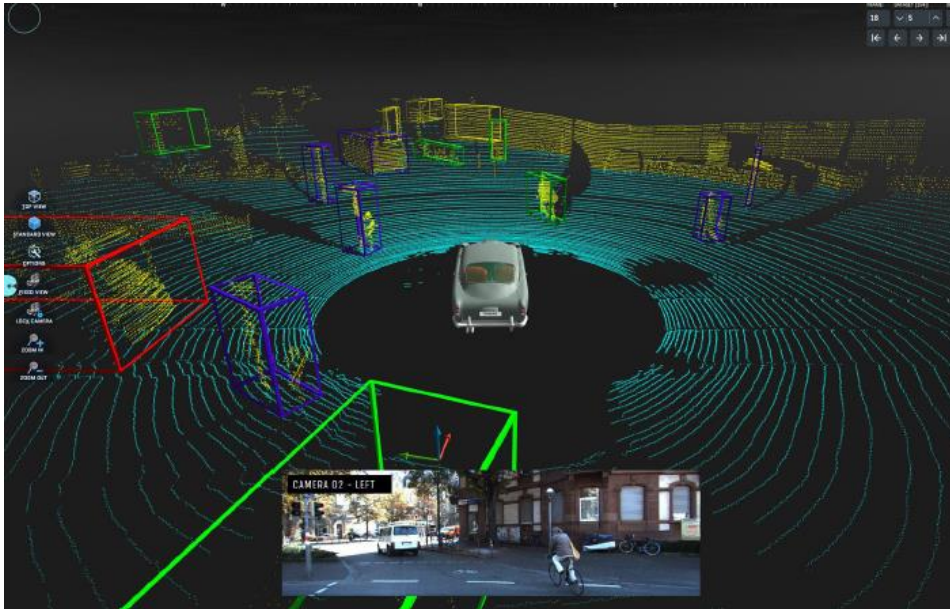


주행가능영역 탐지를 위해
차량 전방의 물체 인식 및 추적 필요

1. Introduction 연구 주제 소개

- MOT (Multi-Object Tracking)

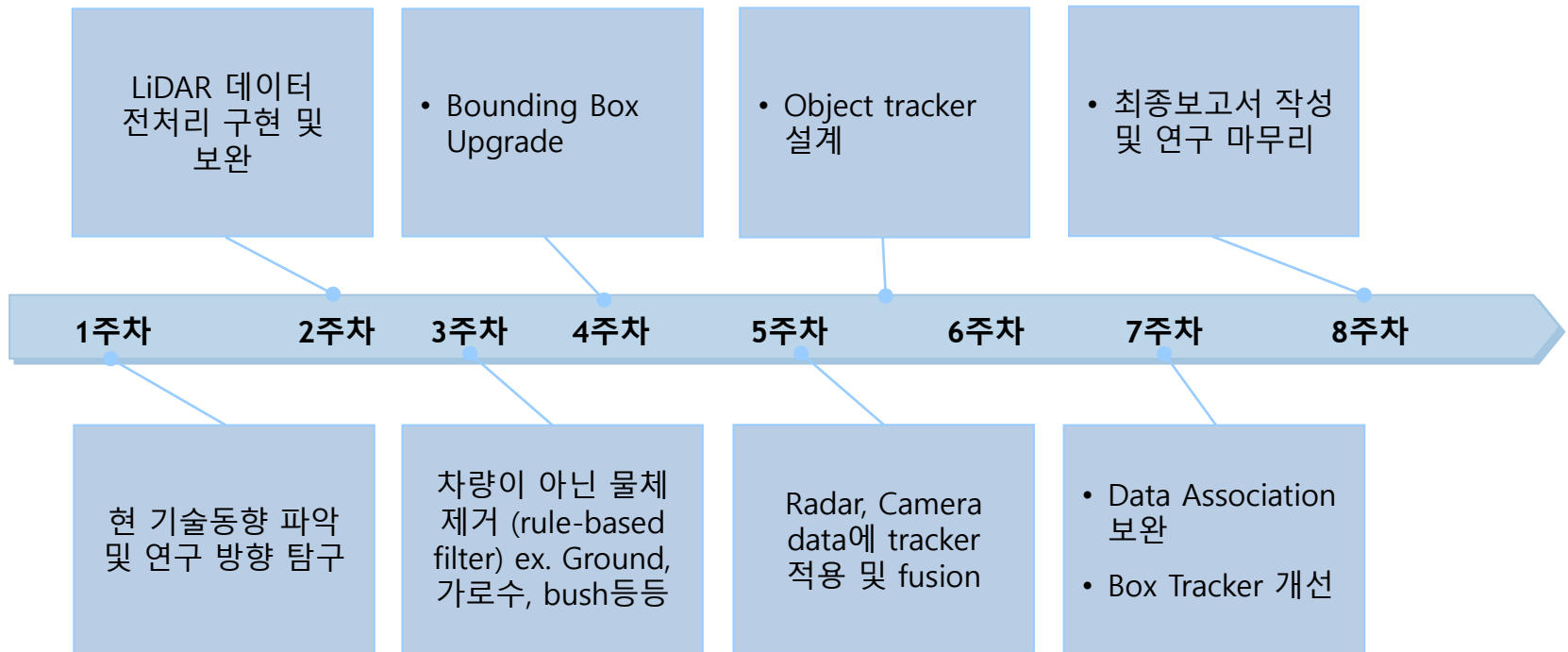
: 차량 전후방의 움직이는 object에 대하여 탐지 및 추적을 수행



$$track = \begin{bmatrix} trackId \\ x \\ y \\ v_x \\ v_y \\ a_x \\ a_y \\ \psi \\ box\ data \end{bmatrix}$$

실제 주행 환경에서는 주변 사물을 인지하고 고려하여 주행해야 하므로 주행 차량 전후방의 object들의 상대적인 정보들을 실시간으로 정확하게 수집하는 것이 중요

1. Introduction 주차별 실습 진행 계획



1. Introduction 각 팀원 별 역할 분담

팀원	역할
이주현	데이터 개발 환경 설정, Kalman filter, Bounding Box, 전체적인 코드 구현 및 디버깅, 속도벡터 plotting
엄동희	Sensor Fusion - Camera, Data association, Box Tracking, Point Tracking
김성준	Kalman filter, minimum rectangle algorithm 적용

Detection

- Data Processing
- Bounding Box

Tracking

- Data Association
- Point Tracking
- Box Tracking

1. Introduction

자율 차량 센서

자율주행 센서 카메라·레이더·라이다 비교

	기능	장점	단점
카메라	-렌즈를 통해 시각적으로 주변 사물, 상황 인식	-질감, 색상, 대비 정도 포착 가능 -저렴한 가격	-날씨 등 외부 환경에 취약 -장거리 측정 취약
레이더	-전자기파 송수신을 통한 거리, 속도 측정	-날씨 등 외부환경 영향 거의 없음 -사물 투과 측정가능	-표지판 인식 불가 -직선 거리만 측정 가능
라이다	-빛(레이저)으로 사물 원근감, 형태, 거리, 속도 인식	-센서 중 가장 높은 해상도·정확도 -3D 입체 지도 구현 가능	-눈, 비 등 날씨에 민감 -비싼 가격

●자료 현대자동차, 현대모비스, 벨로다인, LG이노텍, 테슬라

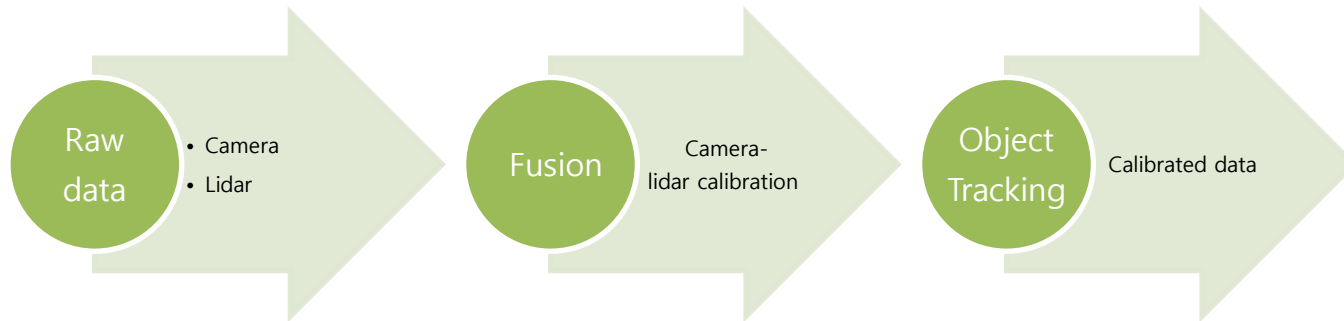
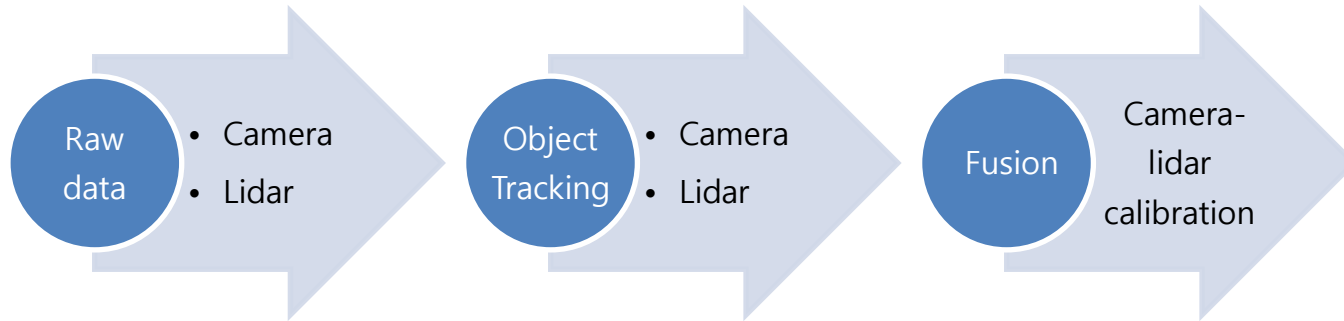
- 카메라 : 객체 인식 알고리즘(YOLO)를 통한 객체 판별에 용이함
- 라이다 : 높은 해상도와 정확도 및 각 점으로의 거리 정보

→ 다양한 환경에서의 다중 객체 인식(MOT)을 위해서 센서 퓨전 (Sensor fusion)을 활용

1. Introduction

센서 퓨전(Sensor Fusion)의 대표적인 방법

Late Fusion

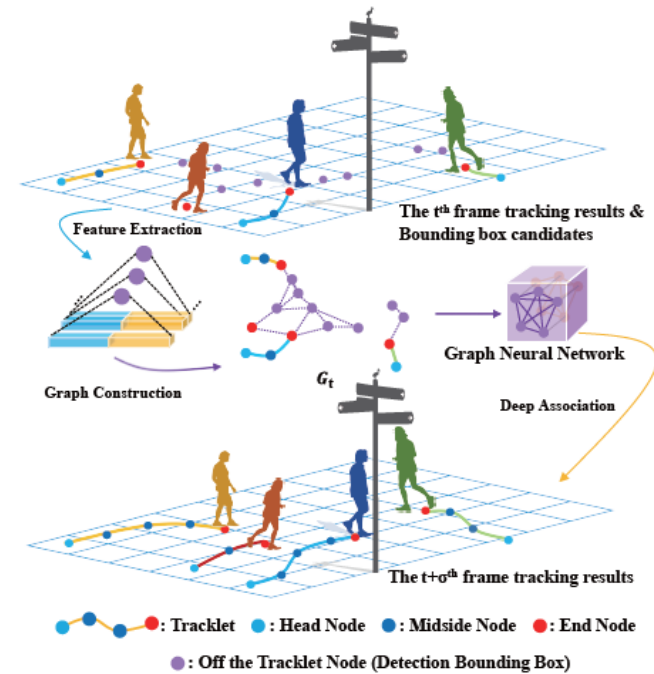


Early Fusion

1. Introduction

Camera MOT

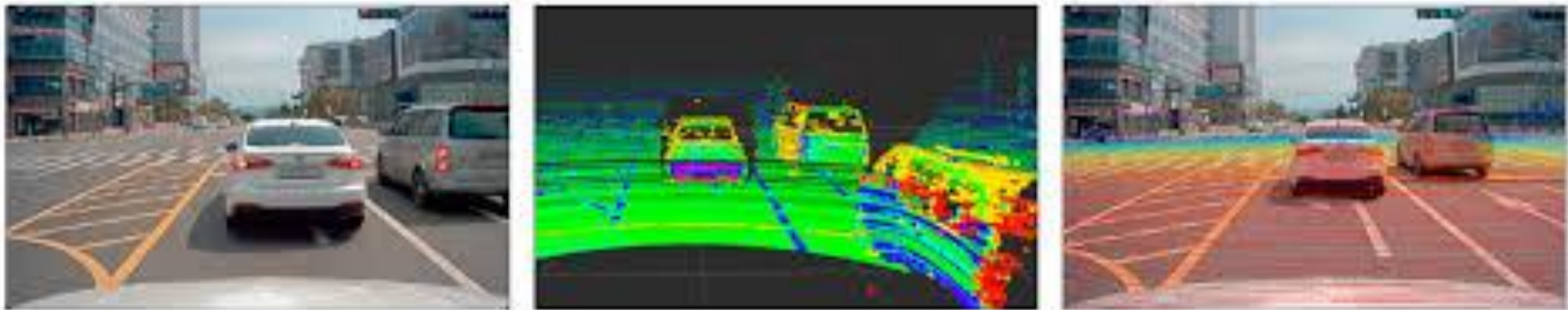
YOLO (실시간 객체 탐지 알고리즘)



- 가장 흔하게 사용되는 tracking 방식
- 딥러닝을 활용하여 object를 인지하고 deep sort를 활용해
- 다양한 모델에 따른 정확도와 연산 속도를 목적에 맞추어 적용 가능하다.
- 이미지 전체를 학습하기 때문에 Background에 의한 Error가 적다.

1. Introduction

Camera – Lidar calibration



- Lidar의 Point Cloud(3D) 데이터를 Camera(2D) 데이터에 일치시키는 것
 - Camera 데이터의 렌즈에 의한 왜곡 보정
 - Camera와 Lidar의 좌표계 일치
- 인식된 개체에 대한 정보를 얻어 이에 따른 알고리즘 설계 용이

1. Introduction



- 3D 데이터를 처리하는 오픈소스 라이브러리
- C++, Python 개발환경 지원
- LiDAR Pointcloud 데이터 처리에 활용



2. Method

1. Data Set

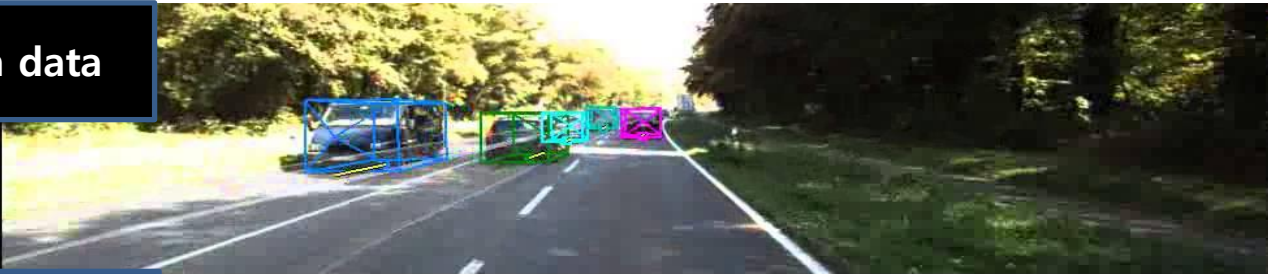
- KITTI Data Set

→ 수집 환경 : 독일

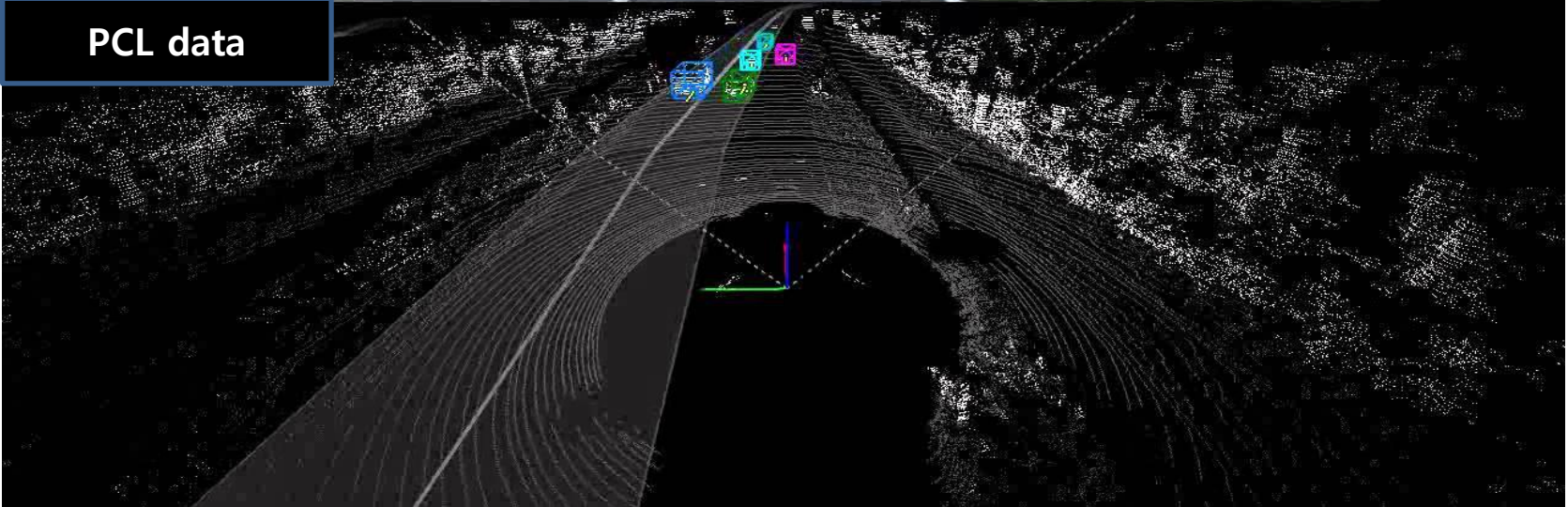
→ 센서 종류 : Lidar , RGB camera 2개, Gray camera 2개, GPS

→ 시간이 동기화된 데이터이다.

Camera data



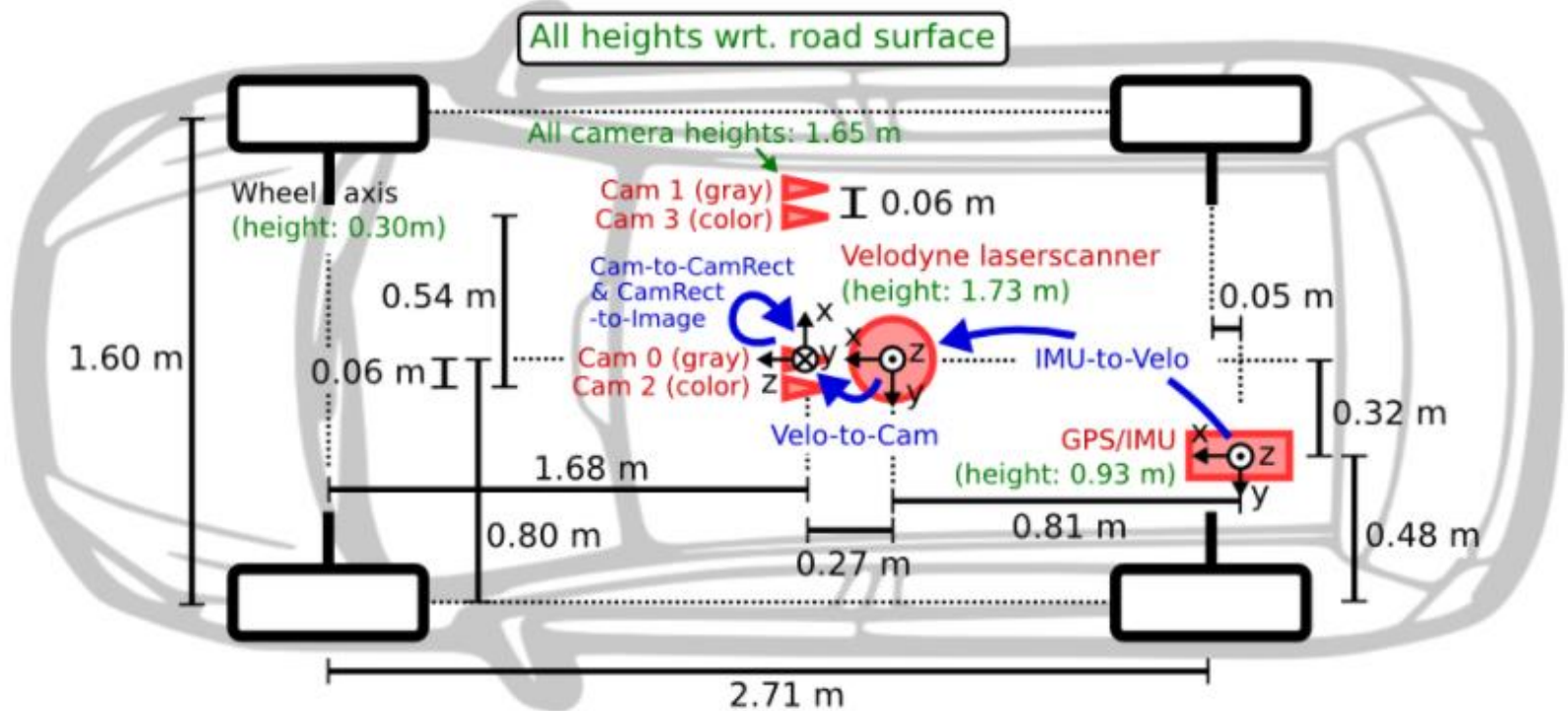
PCL data



2. Method

1. Data Set
 - KITTI Data Set

KITTI 데이터 수집 환경



2. Method - Detection

1. Data Processing

- Ground Removal

⇒ RANSAC 알고리즘을 활용하여 바닥 제거

⇒ Open3d의 segment_plane 함수 활용

```
⇒ cur_plane_model, cur_inliers = xyz_o3d.segment_plane(distance_threshold=0.3,  
                                                         ransac_n=3,  
                                                         num_iterations=50)
```

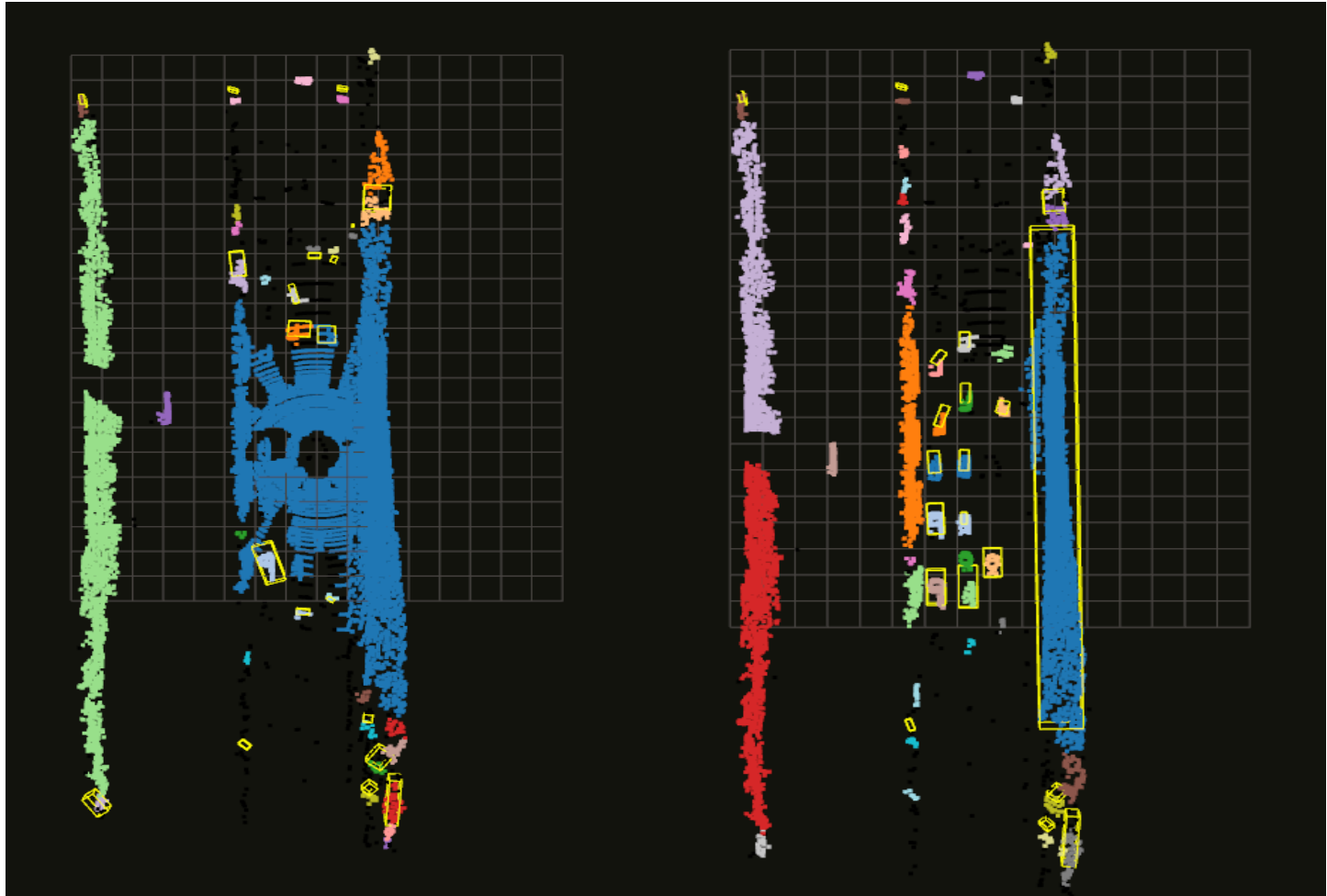
Parameter	용도
distance_threshold	inlier로 간주되기 위한 최소 거리
ransac_n	plane을 추정하기 위해 사용되는 point의 수
num_iterations	RANSAC알고리즘 반복 횟수

⇒ 결과물로 plane model에 해당하는 equation을 $ax+by+cz+d = 0$ 의 형태로 얻을 수 있으며, 해당 plane의 inlier에 해당하는 점들을 구할 수 있다.

2. Method - Detection

1. Data Processing

- Ground Removal

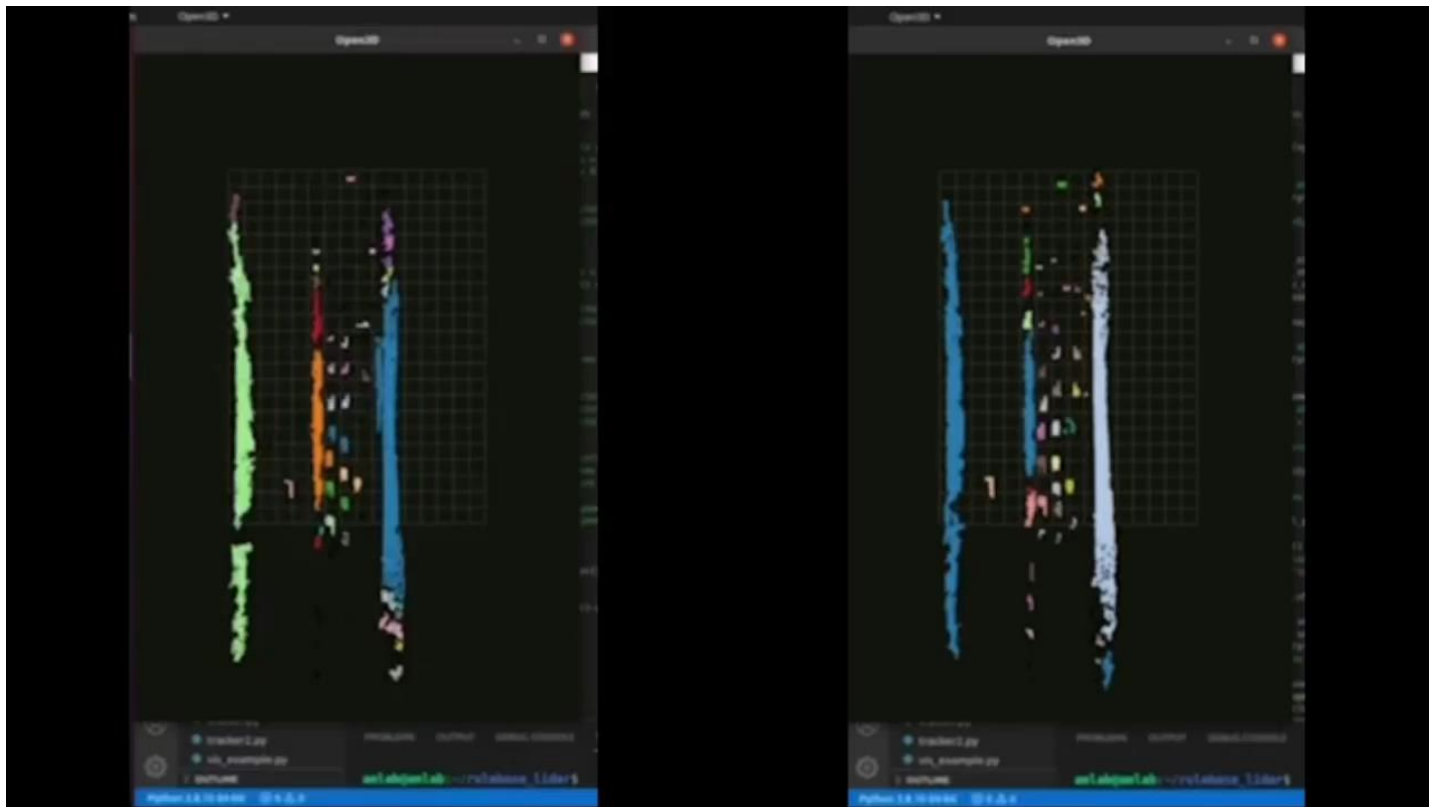


2. Method - Detection

1. Data Processing

- Voxel down sampling

- ⇒ 일반적으로 많이 쓰이는 pointcloud 데이터 전처리 과정중 하나
- ⇒ Voxel당 이를 대표하는 하나의 점으로 대체된다.
- ⇒ Voxel의 크기를 키울 수록 연산량이 줄어 효율적이지만, 점이 해당 voxel을 대표하지 못하여 데이터가 변질될 우려가 있다.



2. Method - Detection

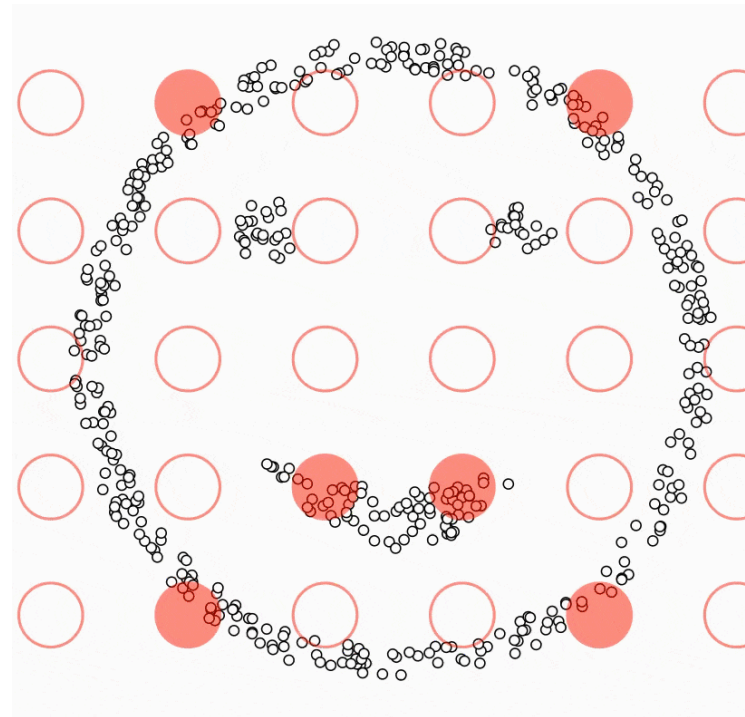
1. Data Processing

[Pointcloud Data Clustering]

- DBSCAN(Density-based spatial clustering of applications with noise)

→ 밀도 기반 군집화

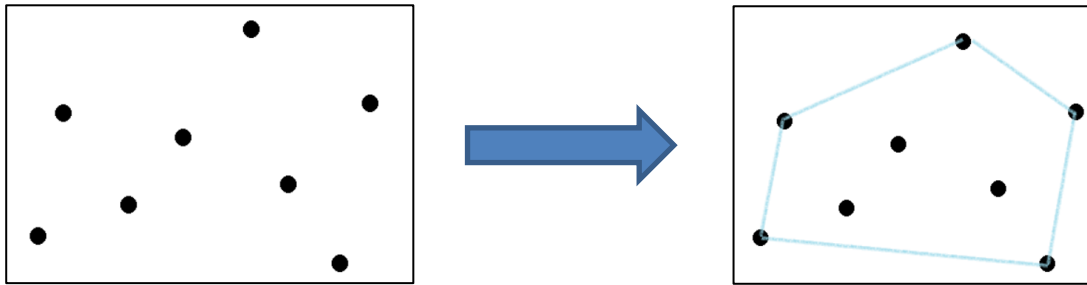
- 거리 $\text{eps}(\text{epsilon}) = 1.3$
- 점의 갯수 $m(\text{min Points}) = 8$



2. Method - Detection

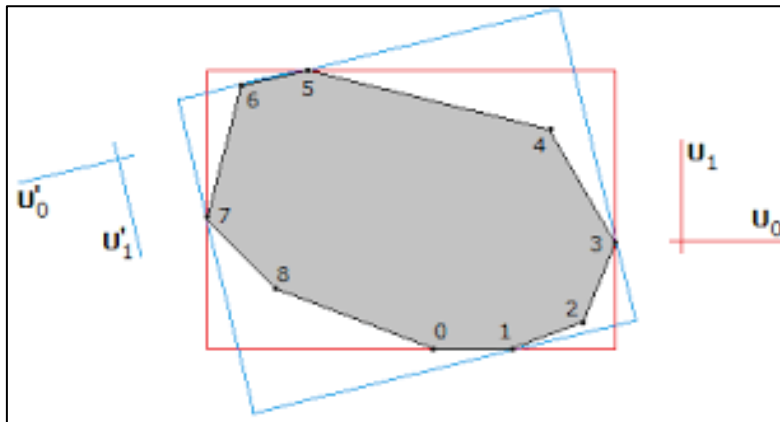
2. Bounding Box

- Convex Hull



2차원 평면상 여러개의 점의 일부를 이용한 볼록 다각형을 만들되 내부에 모든 점을 포함시키도록 하는 방법.

- Minimum rectangle algorithm



다각형의 한 변을 포함하는 직사각형 중 가장 작은 넓이를 가진 사각형을 찾는 알고리즘

2. Method - Detection

2. Bounding Box

- Threshold

- Bounding box로 인식한 객체를 filtering 하기 위한 기준

No.	Threshold variable	Description
1	$T_{height_}(min max)$	min. and max. height of object
2	$T_{width_}(min max)$	min. and max. width of object
3	$T_{length_}(min max)$	min. and max. length of object
4	$T_{area_}(min max)$	min. and max. top-view area of object
5	$T_{ratio_}(min max)$	min. and max. ratio between length and width
6	$T_{ratiocheck_l_}(min)$	min. length of object for ratio check
7	$T_{pt_per_m3_}(min)$	min. point count per bounding box volume.

⇒ T_{ratio} 와 $T_{area(volume)}$, $T_{pt_per_m3_}$ 를 활용하였다.

⇒ T_{ratio} : 도로 양 끝의 나무와 풀들이 좁고 긴 형태의 cluster로 labeling된 것 제외

⇒ T_{volume} : maximum volume 을 설정하여 그 이상의 volume을 가진 cluster 제외

⇒ $T_{pt_per_m3_}$: bounding box안에 point들의 점유율이 일정 비율 이하일 경우 제외

2. Method - Tracking

3. Data Association

- Cost에 따른 동일 객체인식

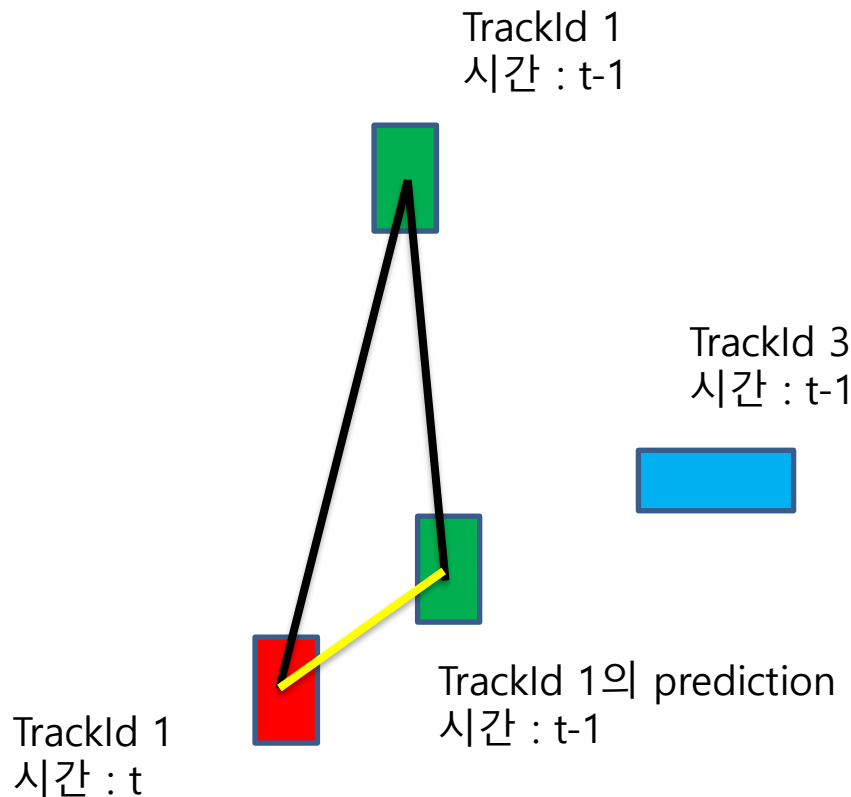
$$cost = \alpha * (\Delta l * \cos(\Delta \theta)) + \beta * \Delta volume + \gamma * \Delta distance$$

- 동일 객체는 size, volume, distance 정보를 통해 구별할 수 있음
- Detection 된 물체의 개수 M과 tracking 하는 물체의 개수 N을 활용하여 N*M의 행렬을 생성
- 행렬에 linear_sum_assignment를 사용하여 tracking에 대해 낮은 cost를 가지는 물체를 동일 물체로 인식

2. Method - Tracking

3. Data Association

- Cost에 따른 동일 객체인식



[필요 정보]

- t-1의 detection
- t-1의 prediction
- 시간 t의 일정 거리 내의 bounding box의 정보



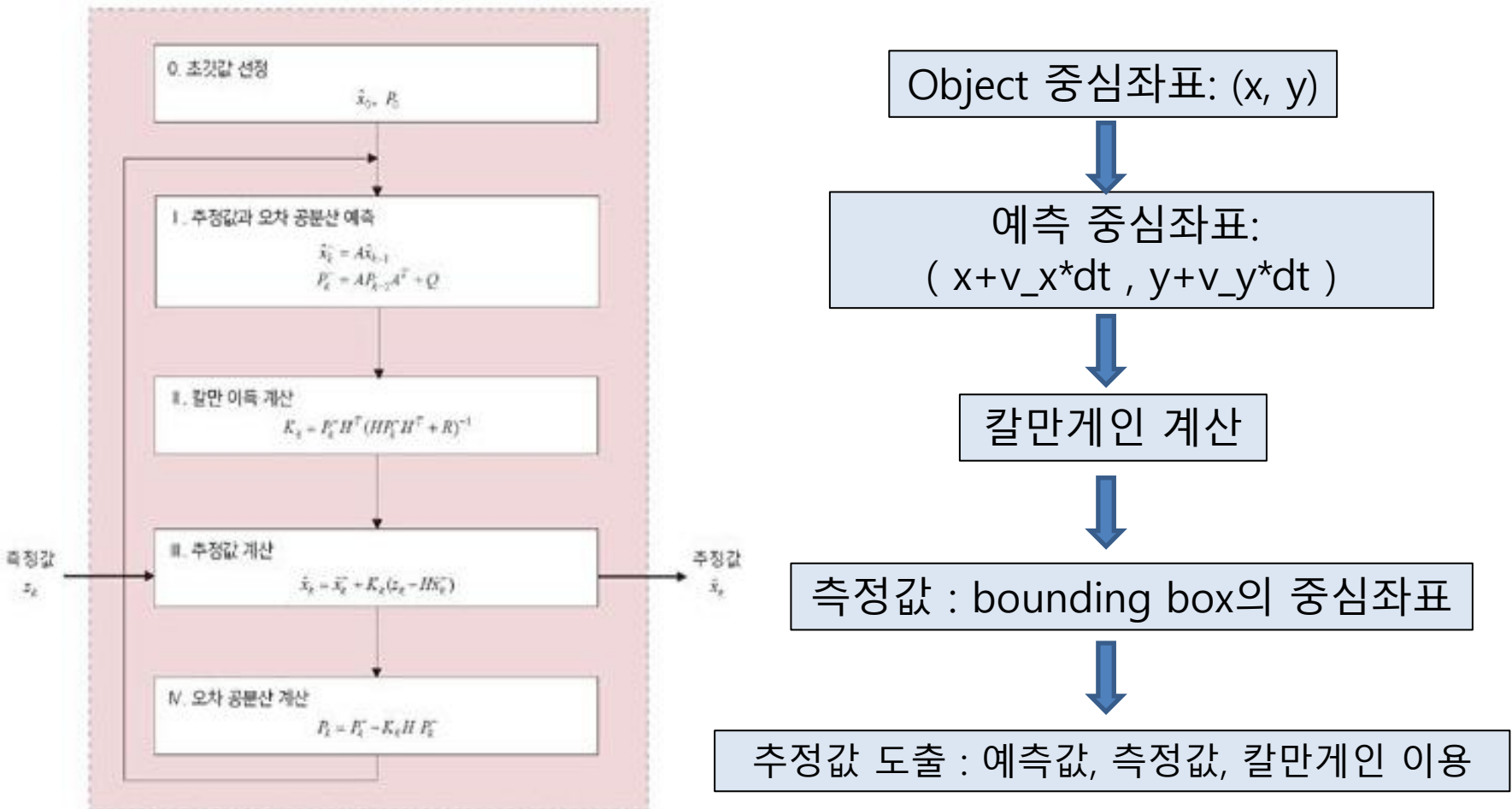
- Size difference
- Angle difference
- Length difference

2. Method - Tracking

4. Point Tracking

- Kalman Filter (Position Tracking)

→ 과거의 값, 현재의 값을 재귀적 연산을 통하여 최적 값을 추정하는 필터

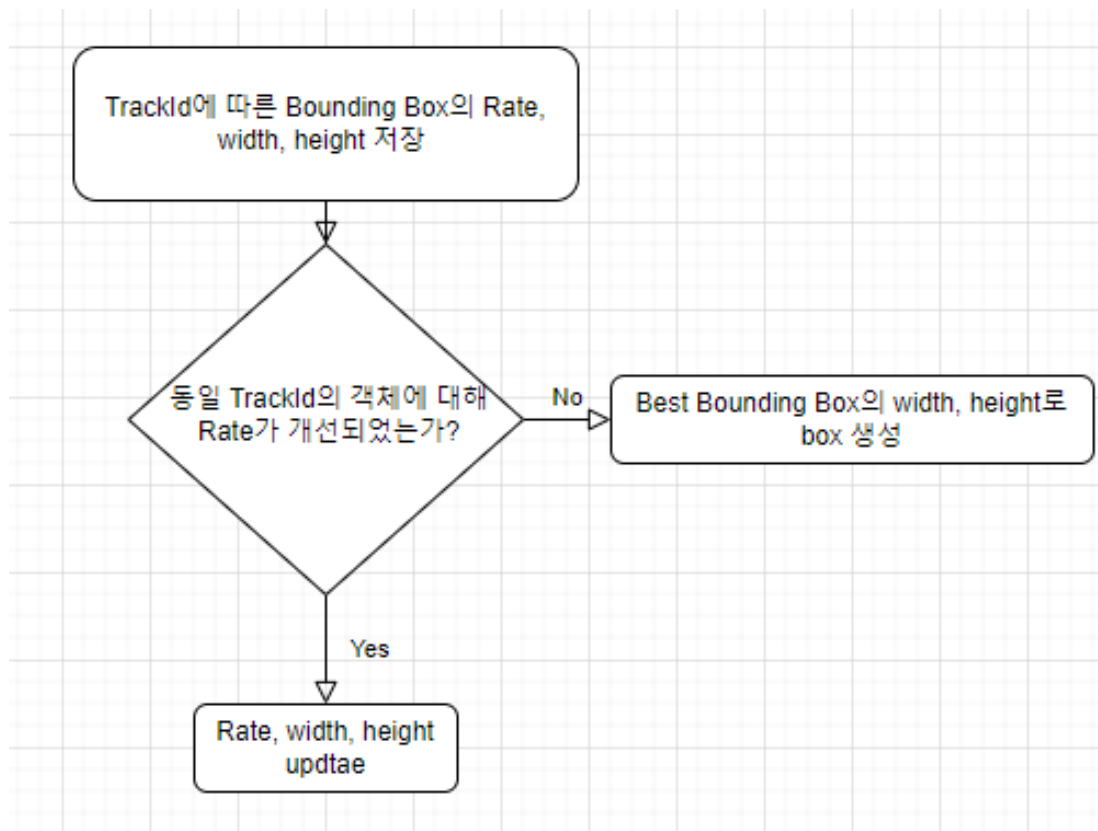


2. Method - Tracking

5. Box Tracking

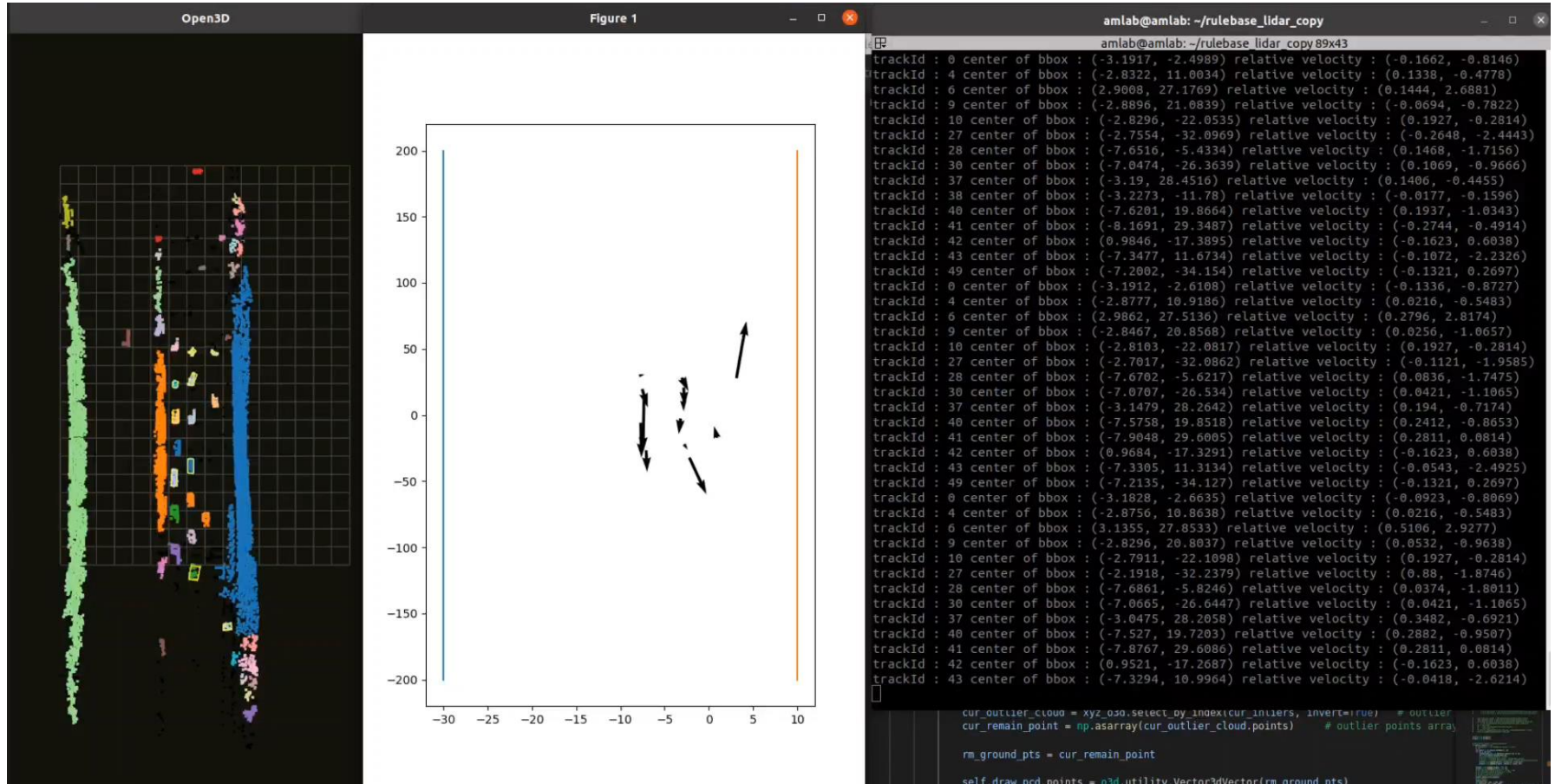
- Method

→ Tracking하는 객체의 TrackID(고유 번호)에 따라 Best Bounding Box Rate와 width, height 저장.



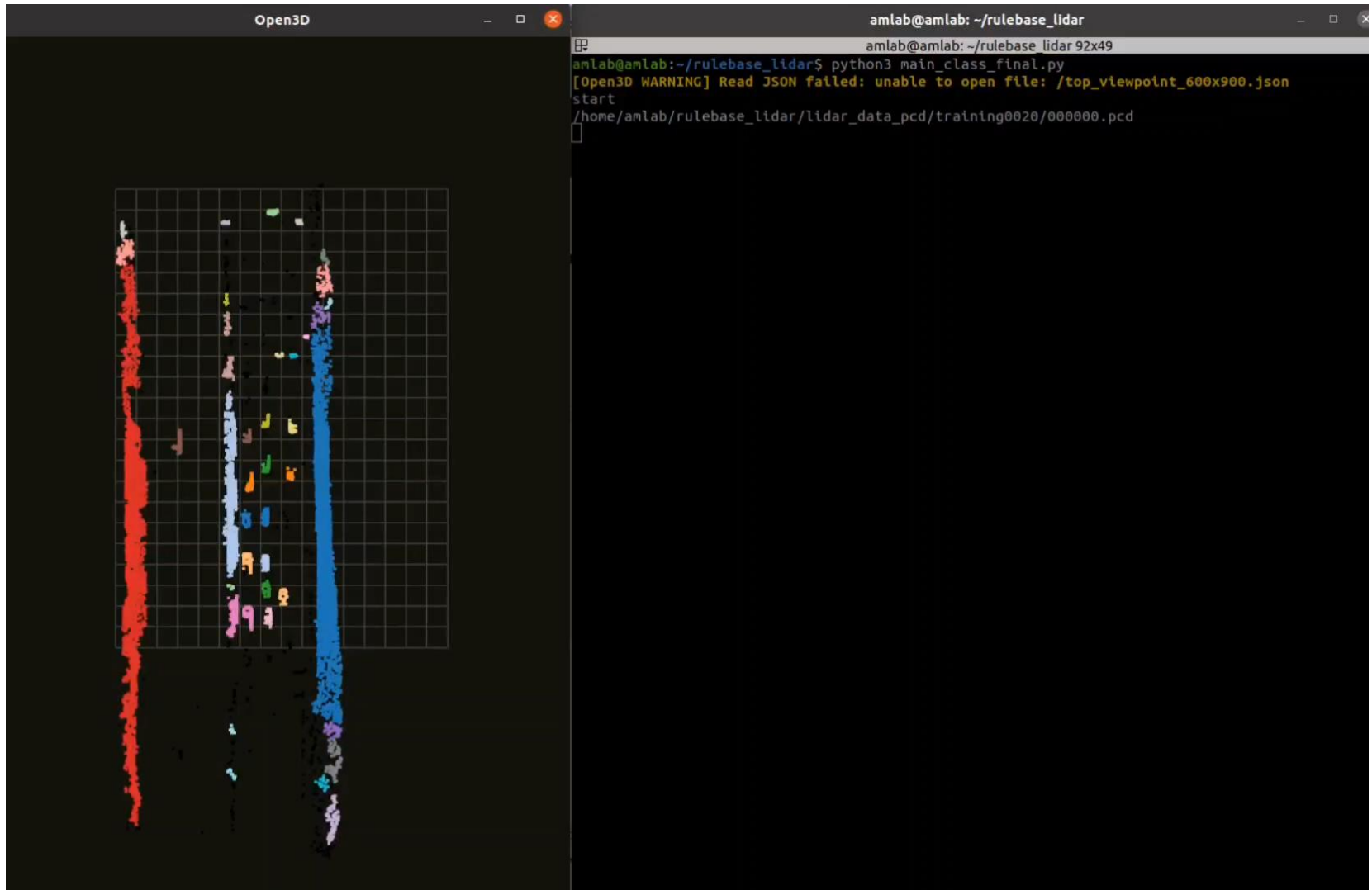
3. Result

결과 영상



3. Result

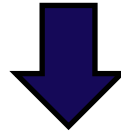
결과 영상





AMLAB 기존의 연구

- LiDAR 데이터를 활용한 차량 전방의 Object Tracking
- 기본적인 Kalman Filter를 적용한 tracking과 Euclidean distance를 활용한 data association

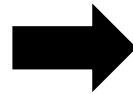
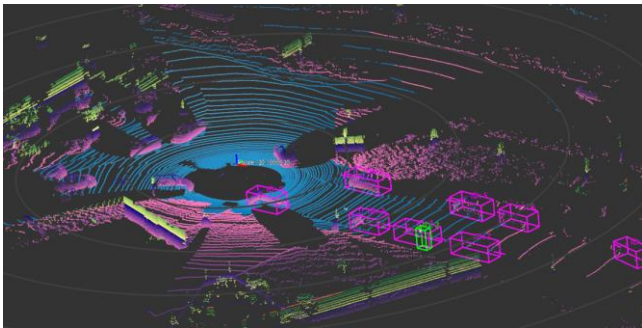


- 최근 연구동향을 반영하여 새롭게 제시된 다음의 아이디어들을 코드로 직접 구현
 - ⇒ Threshold를 활용하여 차량, 보행자 등 움직이는 물체에 대해서만 tracking을 수행하도록 filtering
 - ⇒ Bounding box upgrade를 통해 차량의 heading값 고려
 - ⇒ Data association 보완을 통해 tracking 성능 개선
 - ⇒ 인식 개체의 최적의 인식상태 활용

4. Discussion

연구 결과

- Lidar의 Point Cloud 단일 데이터 사용



다양한 센서 퓨전 적용

- Lidar – Camera sensor fusion



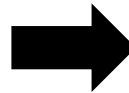
➤ Sensor fusion을 통한 **Data association** 개선

4. Discussion

연구결과

- Constant Velocity case만 고려

$$F_{CV_k} = \begin{bmatrix} x_{posk} + v_k T \sin(\psi_k) \\ y_{posk} + v_k T \cos(\psi_k) \\ 0 \\ v_k \\ \dot{\psi}_k \end{bmatrix}$$



다양한 Motion Model을 적용

- Constant Turn-Rate Velocity

$$F_{CTRV_k} = \begin{bmatrix} x_{posk} + \frac{v_k}{\dot{\psi}_k} (-\sin(\psi_k) + \sin(T\dot{\psi}_k + \psi_k)) \\ y_{posk} + \frac{v_k}{\dot{\psi}_k} (\cos(\psi_k) - \cos(T\dot{\psi}_k + \psi_k)) \\ T\dot{\psi}_k + \psi_k \\ v_k \\ \dot{\psi}_k \end{bmatrix}$$

- Random Motion

$$F_{RM_k} = \begin{bmatrix} x_{posk} \\ y_{posk} \\ \psi_k \\ v_k \\ \dot{\psi}_k \end{bmatrix}$$

Reference

1. 김성준, 양동원, 정영현, 김수진, 윤주홍, 2014, [이동물체 탐지를 위한 레이더 데이터 거리 - 도플러 클러스터링 기법], 국방과학연구소 국방무인기술센터
2. 김수진, 정영현, 강재웅, 윤주홍 , 2016, [표적의 형상정보를 활용한 다중표적 추적 기법]
3. Himani S. Parekh, 2014, [A Survey on Object Detection and Tracking Methods], IJIRCCE
4. A.S. Abdul Rachman, [3D-Lidar Multi Object Tracking for Autonomous Driving], TuDelft