

---

# Capstone Design

## 프로젝트 최종 발표

20180402  
최주형

# INDEX



01 과제 개요



02 시스템 요청 사항



03 구현할 시스템 Image



04 시스템 주요 기능



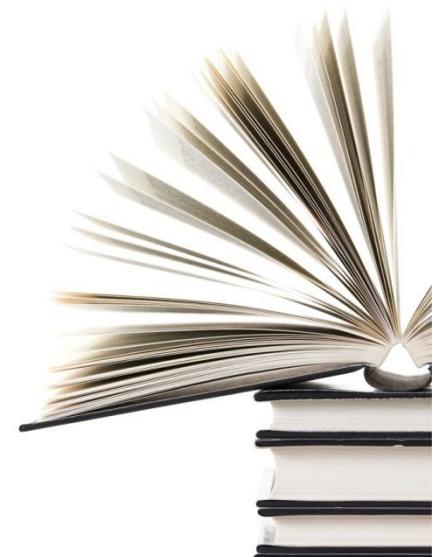
05 향후 추진 일정



06 고려 사항



07 기대 효과



01

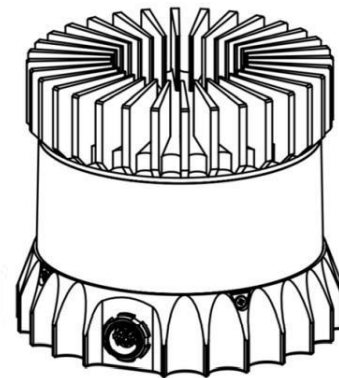
과제 개요



# 1. 과제 개요



## 배경 설명



LiDAR(Light Detection And Ranging, LiDAR)



레이저 펄스를 물체에 쏜 후 이를 다시 수신하는 데까지 걸리는 시간을 측정해 '물체의 형상'을 추측



반사체의 위치좌표 측정/ 거리를 비롯해 폭과 높낮이 정보까지 측정 → 3차원으로 인식



자율주행에서의 LiDAR 활용



mapping, localizing, detecting



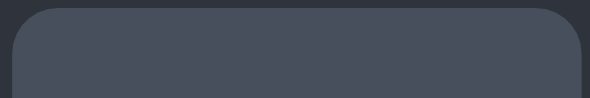
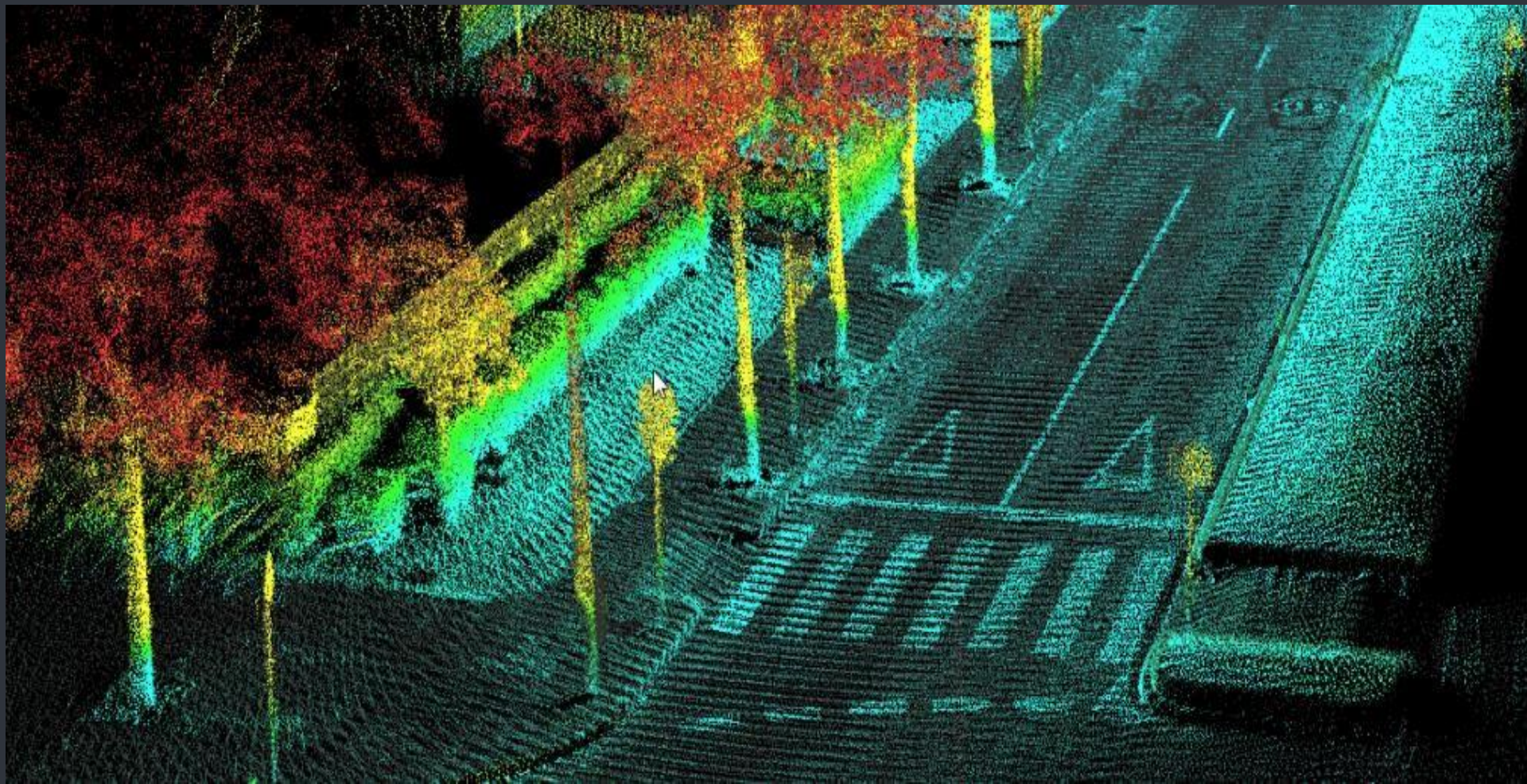
실시간으로 물체, 장애물, 사람, 자동차 등의 객체 인식 및 추적, 이동체의 속도와 방향 측정 등을 높은 정밀도로 수행



자율주행 자동차에서 공간 좌표와 거리를 기반으로 도로 구조 mapping과 전방 물체에 대한 3차원 정보를 실시간으로 획득하는 센서로 활용



## LiDAR scan을 통해 보여지는 모습





# 1. 과제 개요



## 문제 정의



LiDAR의 정보를 통해 얻은 point cloud



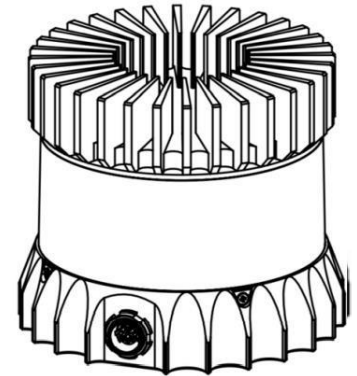
하나의 point cloud 입장에서 수집된 LiDAR 빔의 형상



이상적인 결과 : 시작 포인트와 끝 포인트가 만나는 하나의 원의 형상



실제 point cloud : 시작 포인트와 끝 포인트가 만나지 않음



## 문제 원인



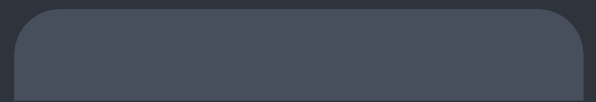
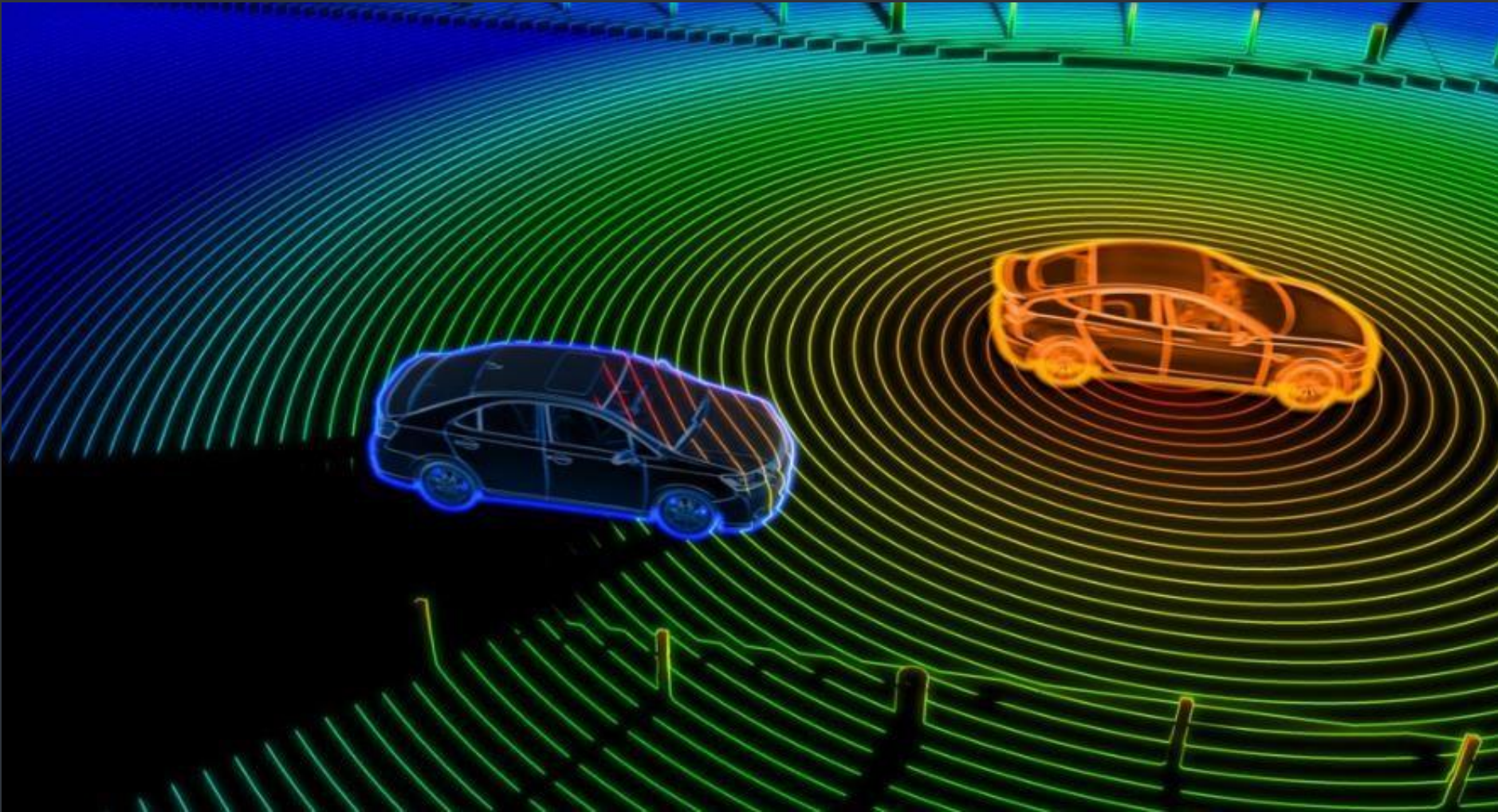
이동하는 vehicle → LiDAR 빔을 쏘는 지점이 **한 지점**이 아니라 **한 구간**이 됨

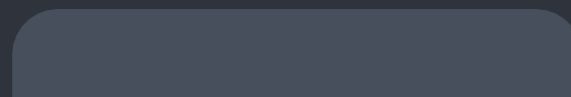
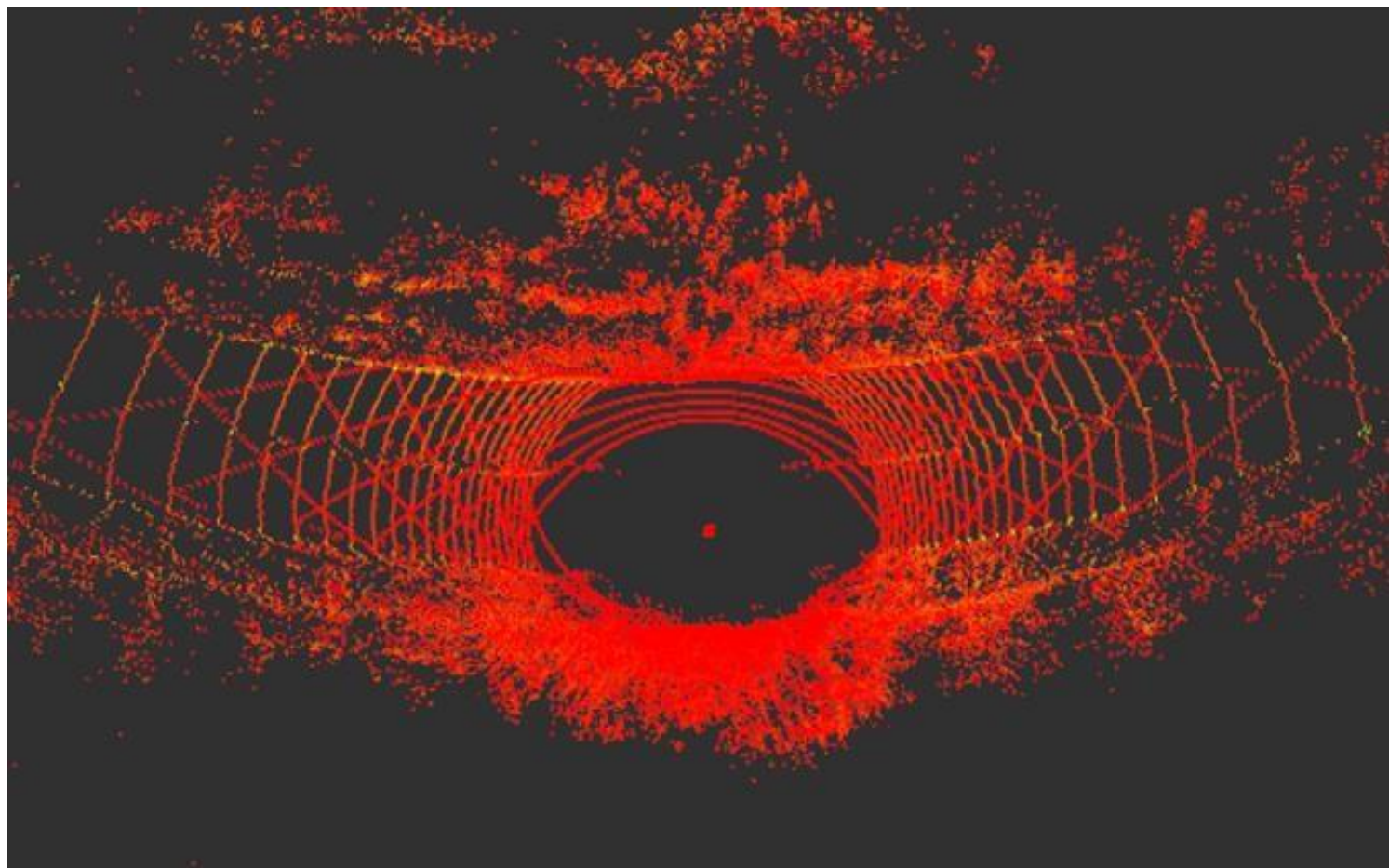


시간  $T(s) \sim T+0.1(s)$  동안 vehicle이 전방으로 이동한 거리



LiDAR scan은 반지름이 다른 수십 개의 동심원으로 구성됨

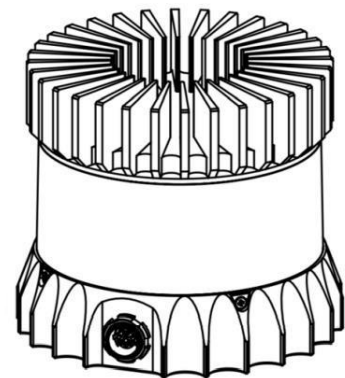








# 1. 과제 개요



## 목적



과제 추진 목적 및 과제 수행을 통해 얻을 수 있는 기대 효과



실제 LiDAR가 수집한 point cloud는 서로 다른 시간에 발사한 빔이 검출한 points



LiDAR scan을 보정하여 시간  $T+0.1(s)$ 에 vehicle이 정지해 있다고 가정할 때, 검출될 수 있는 point cloud의 모습으로 변형



Ground truth(실제 물리 환경, 차량이 정지해 있을 때 얻을 수 있는 point cloud)와 보다 유사한 point cloud를 얻을 수 있음



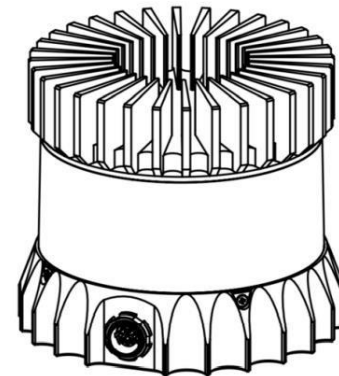
# 1. 과제 개요



## 목적



## 특장점



LiDAR를 활용하는 분야가 많은 만큼 다양한 분야에서의 개선이 기대됨



예) 우주 탐사 로봇에서의 활용, 3차원 영상 복원을 위한 스캐너, 미래 자율주행자동차 등



자율주행에서의 성능 개선을 목표로 하고 있음



mapping



matching



Object detecting



clustering

02

## 시스템 요청 사항





## 2. 시스템 요청 사항



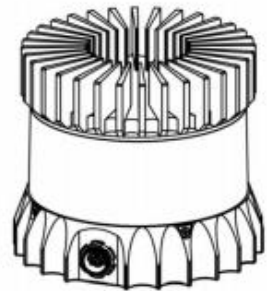
### LiDAR 정보



Point cloud를 생성할 LiDAR packet 수신



구현할 LiDAR 특징 → velodyne, ouster 등



### odometer



차에 장착하여 바퀴가 돌아가는 대로 주행 거리를 측정



Linear.x 와 angular.z



delta\_x, delta\_y, delta\_th, time stamp 추출





차량의 odometer정보를 통해  
lidar 빔의 기준이 되는 **구간의** 정보를 얻는다.



단위 시간 마다 각각의 point를 보정하여  
point cloud를 생성한다.





## 2. 시스템 요청 사항



### 통합 LiDAR driver



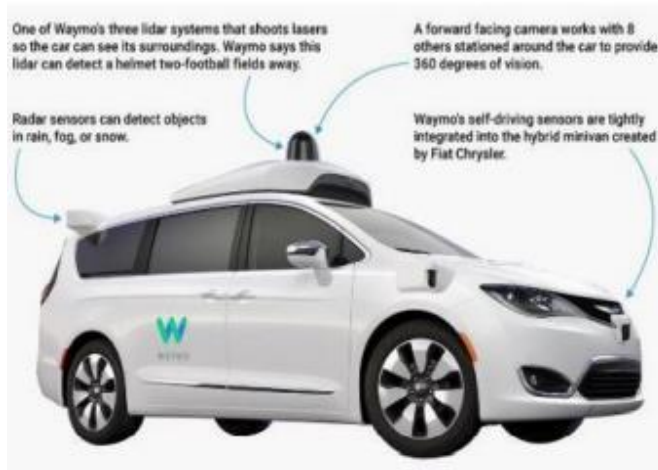
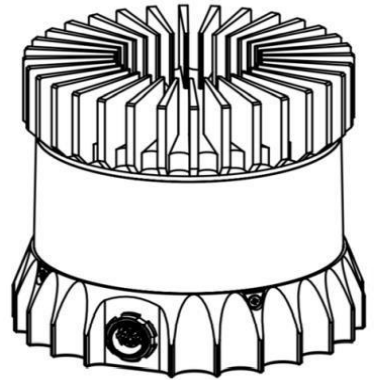
필요성



LiDAR 구입시, 해당 LiDAR를 구동하는 드라이버를 제공함



LiDAR 의 종류는 매우 다양 → packet 구조가 모두 다름 → 구동하는 드라이버가 다름





03

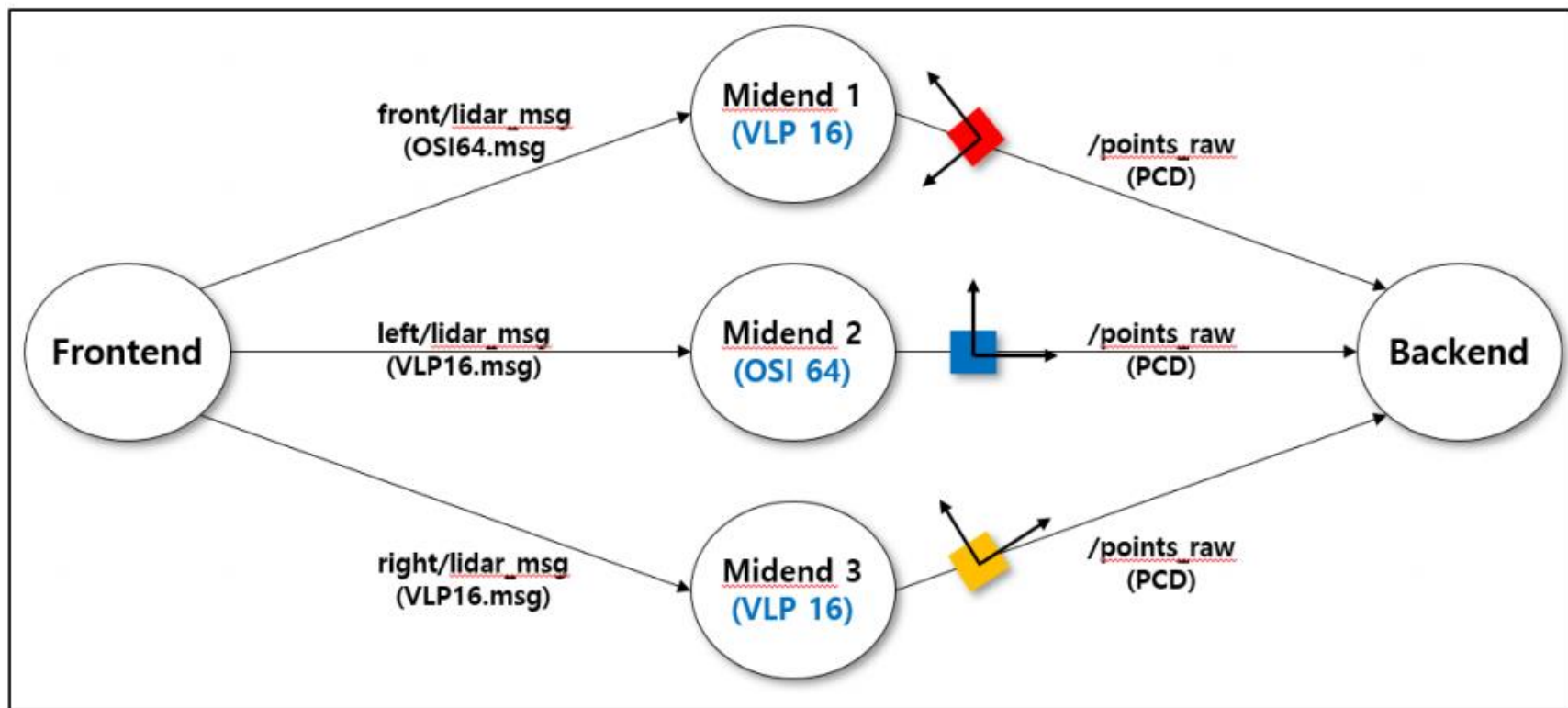
구현할 시스템 Image



### 3. 구현할 시스템 Image

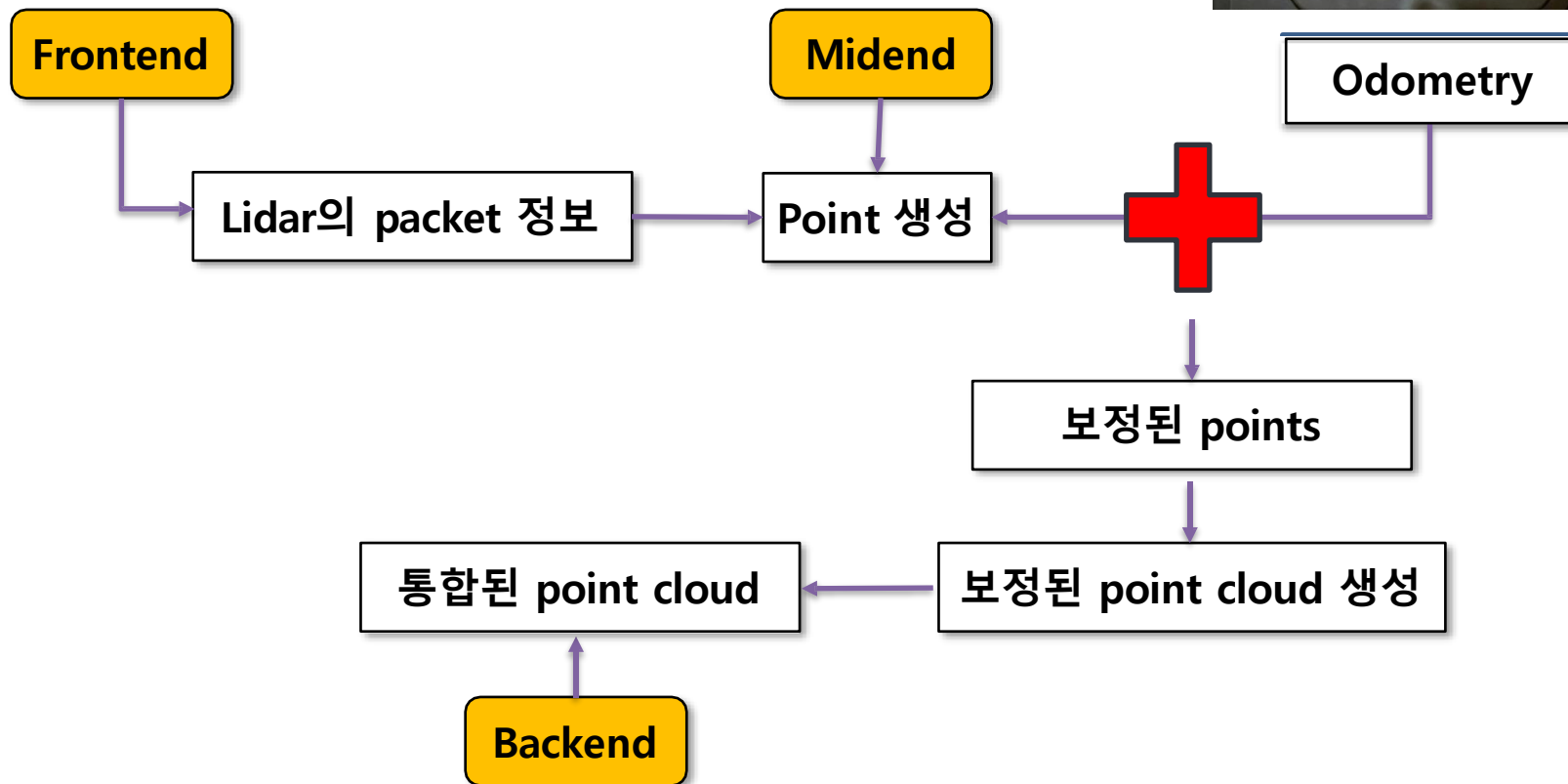
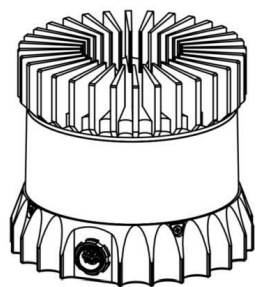


통합 LiDAR driver





### 3. 구현할 시스템 Image





04

## 시스템 주요 기능



## 4. 시스템 주요 기능



멀티 라이다 시스템을 구동하기 위한 드라이버 구조



ROS(Robot Operating System)환경



Nodelet 구조



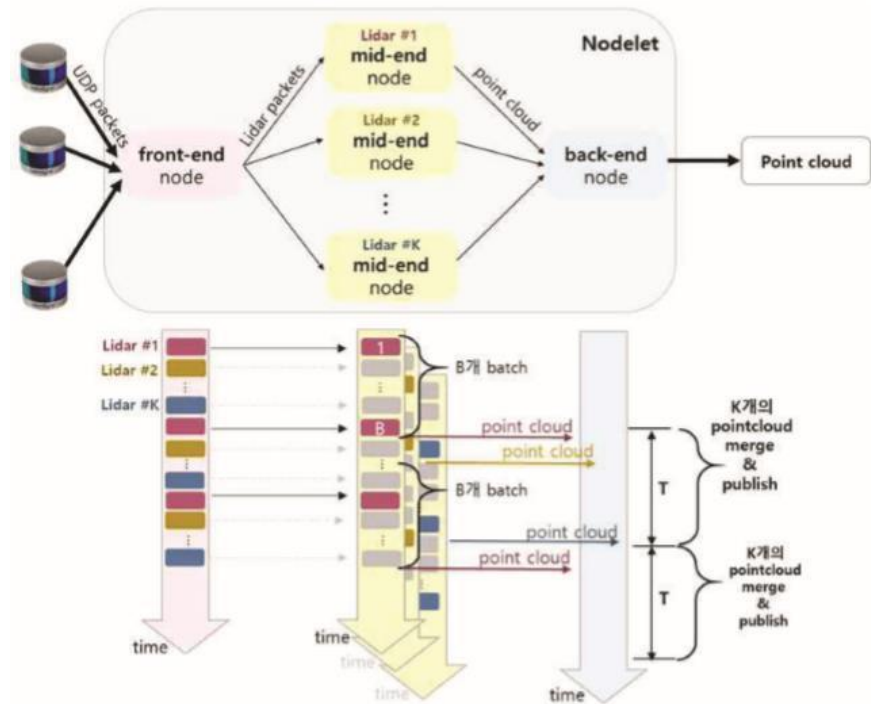
한 개의 프론트 엔드 (front-end) 노드



K 개의 미드엔드 (mid-end) 노드



한 개의 백엔드 (back-end) 노드





## 4. 시스템 주요 기능



멀티 라이다 시스템을 구동하기 위한 드라이버 구조



프론트 엔드 (front-end) 노드



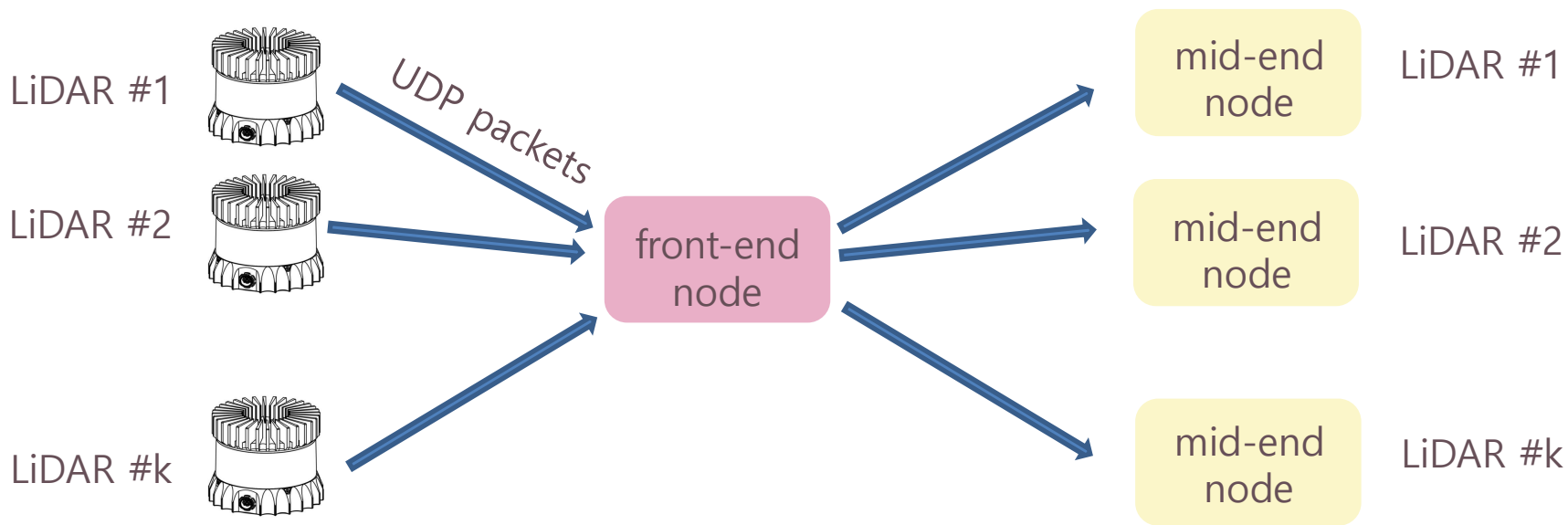
K 개의 LiDAR로부터 UDP 패킷들을 수신



LiDAR packet 추출



LiDAR x로부터 수신한 packet을 미드엔드 노드 x 로 전달





## 4. 시스템 주요 기능



멀티 라이다 시스템을 구동하기 위한 드라이버 구조



미드 엔드 (mid-end) 노드



연속한 B개의 packet들이 동일한 시간에 생성된 것으로 취급



단위 시간 동안 odometry 정보를 수신



Odometry 정보를 활용하여 각각의 point를 보정



이들을 모두 묶어서 한꺼번에 포인트 클라우드 (point cloud) 형태로 변환



미드엔드 노드 (mid-end node) x는 LiDAR x로부터 수신한 packet 안에 존재하는 데이터 points들을 포인트 클라우드 (point cloud) 형태로 변환



백엔드 노드 (back-end node)에 전달





## 4. 시스템 주요 기능



멀티 라이다 시스템을 구동하기 위한 드라이버 구조



백 엔드 (back-end) 노드



K개의 미드 엔드 노드(mid-end node)들이 전달한 포인트 클라우드 (point cloud)



차량 중심을 원점으로 삼는 하나의 3차원 좌표계로 변환, 통합



최종 output : 하나의 3차원 좌표계로 통합된 포인트 클라우드 (point cloud)



활용



3차원 지도 매칭 알고리즘 : 자차 위치 추정



포인트 클러스터링 알고리즘 : 사물인식



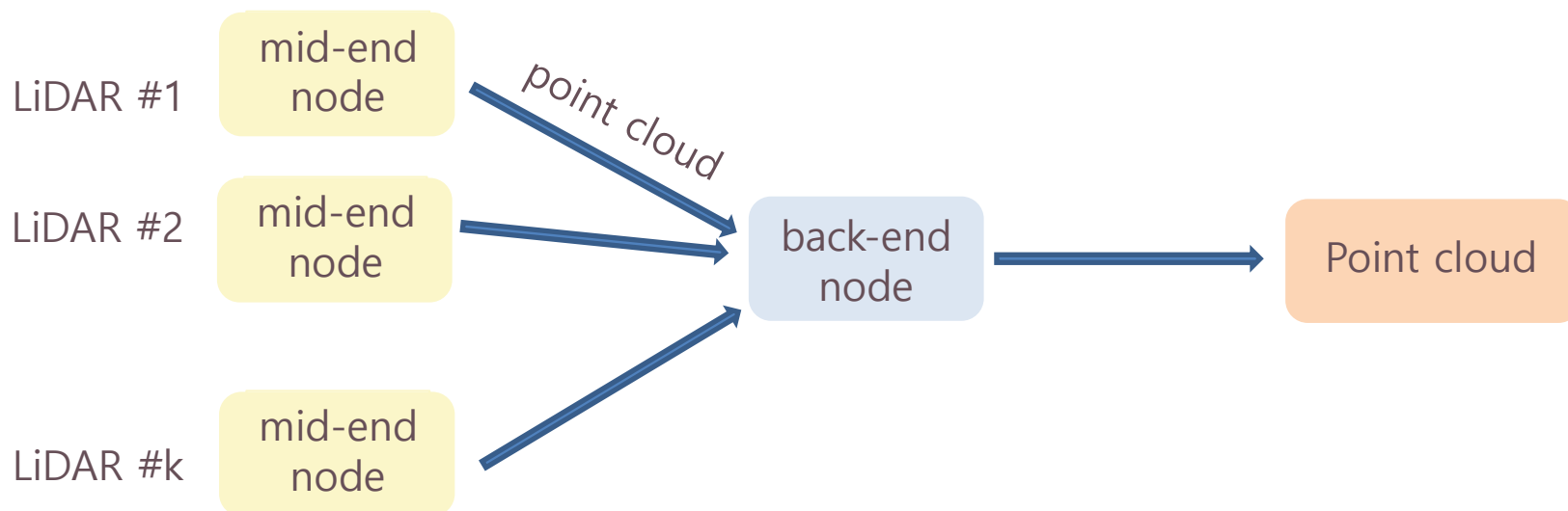
## 4. 시스템 주요 기능



멀티 라이다 시스템을 구동하기 위한 드라이버 구조



미드 엔드 노드(mid-end node)와 백 엔드 (back-end node)



05

향후 추진 일정



## 5. 향후 추진 일정

구분	Activity	일정							
		2021							
		3	4	5	6	7	8	9	10
설계	SW 아키텍처 설계								
	시스템 모델링								
	simulation 환경 세팅								
개발	샘플 수집 (pcap, rosbag)								
	lidar A 시스템 구현								
	lidar B 시스템 구현								
	시스템 통합								
평가	평가 도구 SW 아키텍처 설계								
	알고리즘 적용								
	구현								
	평가 도구 활용한 평가								
	실시간 주행 테스트								
유지	시스템 디버깅								
	품질 분석								
	사용자 요구사항 재분석								



06

고려 사항



## 6. 고려 사항



### Nodelet 구조

- ✿ 드라이버를 3단계의 노드(node) 구조로 분할 → 파이프라인 처리 효과를 위해
- ✿ 모든 노드(node)들이 독립된 실행 개체, 즉 스레드(thread)들로서 병렬로 실행할 수 있도록 설계
- ✿ 스레드(thread)들이 하나의 주소 공간 안에서 실행되도록 설계 → synchronization
  - 🍃 Lidar packet data가 짧은 시간에 많이 들어오기 때문
  - 🍃 (Ex. 16-channel velodyne LiDAR → 초당 753.6개의 packet 생성)
  - 🍃 노드간 메시지 전달 시 포인터 전달
  - 🍃 불필요한 메시지(msg) 복사 오버헤드 제거



### 멀티 라이더 퓨전 시스템

- ✿ LiDAR 종류에 따라 packet 구조가 다름 → 센서 데이터 처리를 위한 driver 코드 다양
- ✿ 여러 종류의 LiDAR를 함께 부착해도 모두 적용 가능 → 확장 가능하도록 설계

07

기대 효과



## 7. 기대 효과



### 자율주행에서의 성능 개선

- vehicle이 빠르게 이동할수록, lidar driver를 통해 얻을 수 있는 point cloud의 오차 (ground truth, 실제 물리환경과의 차이)가 커지는 문제를 해결
  - 기존의 방식으로 생성한 point cloud는 vehicle이 빠르게 이동할수록 블러링이 심함
  - 오차가 심할수록 실제 건물, obstacle 등이 넓게 퍼져 보이게 됨
- LiDAR scan을 보정하여 시간  $T+0.1(s)$ 에 vehicle이 정지해 있다고 가정할 때, 검출될 수 있는 point cloud의 모습으로 변형
  - Ground truth (실제 물리 환경, 차량이 정지해 있을 때 얻을 수 있는 point cloud)와 보다 유사한 point cloud를 얻을 수 있음
- 점군 지도 제작, 자차 측위, object detecting, clustering



### LiDAR를 활용하는 많은 분야에 적용 가능

- 예) 우주 탐사 로봇에서의 활용 등

THANK YOU

