

Part IV. Mapping for Autoware (2 hours)

숭실대학교 ICHTHUS 팀
김강희 교수(khkim@ssu.ac.kr)
이효은, 최규진, 백한나 연구원

Outline

- ❖ Normal Distributions Transform
- ❖ point map
- ❖ vector map

Normal Distributions Transform

Overview

- ❖ Autoware는 mapping과 localizing을 위해서 NDT 알고리즘을 사용함
 - ndt_mapping : 3차원 point map 생성
 - ndt_matching : point map 안에서 localizing 수행
- ❖ NDT 알고리즘은 다음 논문들에서 다루고 있음
 - “The Normal Distributions Transform : A New Approach to Laser Scan Matching” 논문에서 처음 제안됨
 - ❖ Peter Biber, University of Tübingen, 2003
 - “Factors to Evaluate Capability of Map for Vehicle Localization” (IEEE Access) 논문에서 point map에 대한 다양한 평가 기준이 제시됨
 - ❖ Ehsan Javanmardi et al. University of Tokyo, 2018

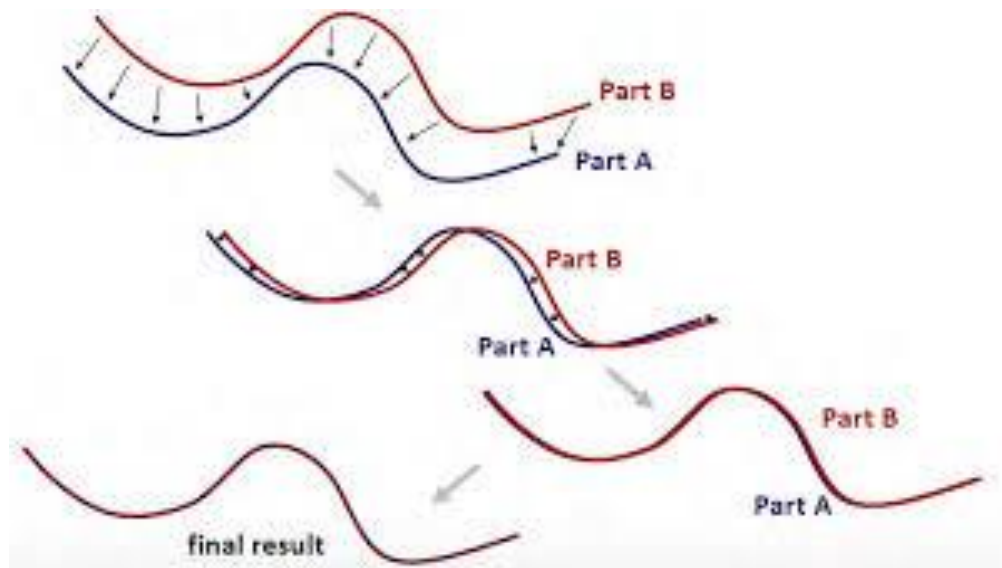
ICP (Iterative Closest Point)

❖ Goal

- 두 PCD (point cloud) 사이의 차이를 최소화하는 행렬 변환 찾기

❖ Algorithm steps

1. source points 에 속하는 각 point 에 대하여 reference points 중에 가장 가까운 point 를 찾아 match 시킴
2. point 와 point 간의 거리를 최소화하는 translation & rotation matrix 를 계산
3. 해당 matrix transform 을 적용한 후 전체 알고리즘을 반복함



NDT (Normal Distributions Transform)

❖ Overview

- PCD 를 단순히 point 집합으로 이해하지 않고, point 들이 위치할 확률 공간으로 이해함
- reference PCD 를 grid (or voxel)로 분할하고 각 cell 마다 NDT 변환을 적용하여 해당 cell 에 point 가 위치할 확률을 확률 함수로 표현함
- source PCD 의 각 point 에 대해서 특정 cell 내부의 특정 위치에 있을 확률을 계산하고 모든 source point에 대한 확률들의 합을 최대화하는 matrix transform 을 얻어냄
- middle-size indoor 환경에서 높은 정확도의 point map 을 생성함 → vast-size outdoor 는 어떻게 하나?
- ICP 알고리즘보다 정확도가 높고 계산량이 작다고 알려져 있음

NDT 적용 과정

❖ 2차원 reference PCD 를 대상으로

① laser scan 을 grid cell 단위로 분할 : $1m \times 1m$ 크기의 grid cell

- 각 grid cell 안에 있는 포인트들의 집합

$$x_i = \{x_1, x_2, \dots, x_n\}$$

② grid cell 마다 포인트들(의 좌표)의 평균을 계산함

- $q = \frac{1}{n} \sum_i x_i$

③ grid cell 마다 계산된 평균과 각 포인트의 차이를 통해 분산 계산함

- $\Sigma = \frac{1}{n-1} \sum_i (x_i - q)(x_i - q)^t$

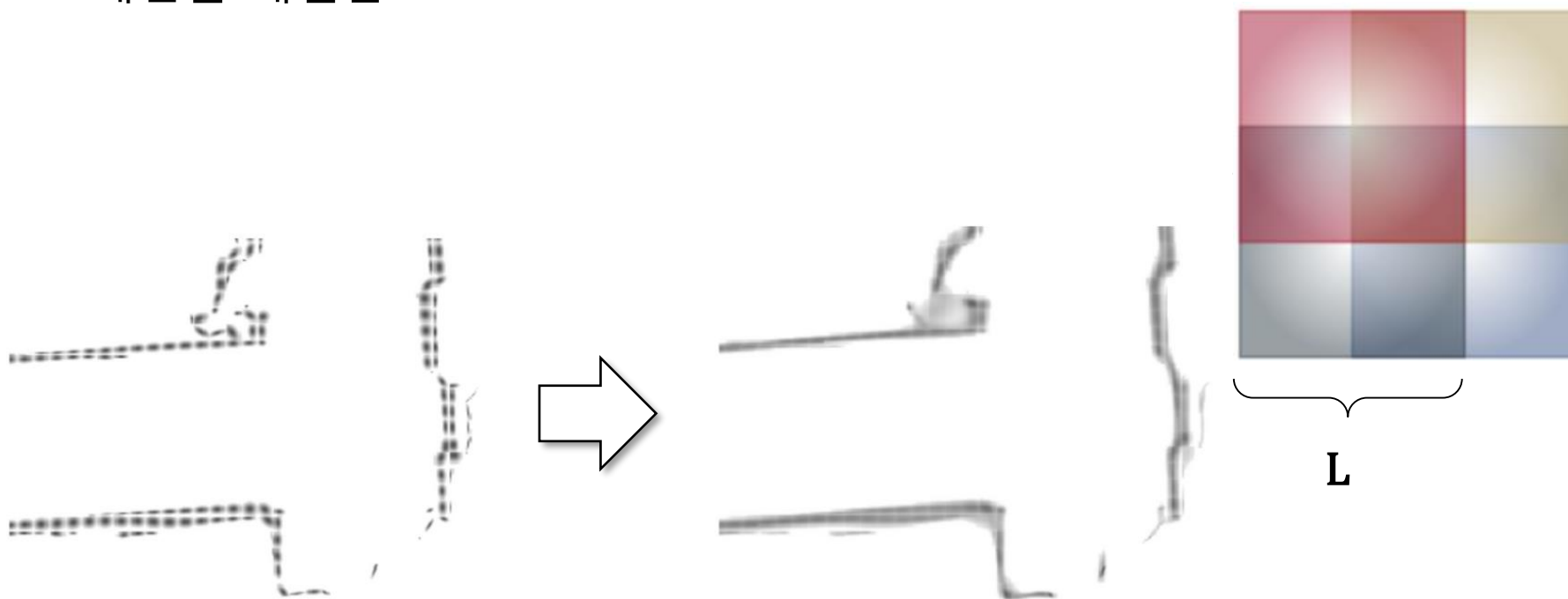
④ 평균과 분산을 통해 grid cell 내에서 sample x 가 존재할 확률 분포를 결정함

- $p(x) = C \exp\left(-\frac{(x-q)^t \Sigma^{-1}(x-q)}{2}\right)$

NDT 적용시 고려사항들

❖ Discretization effects

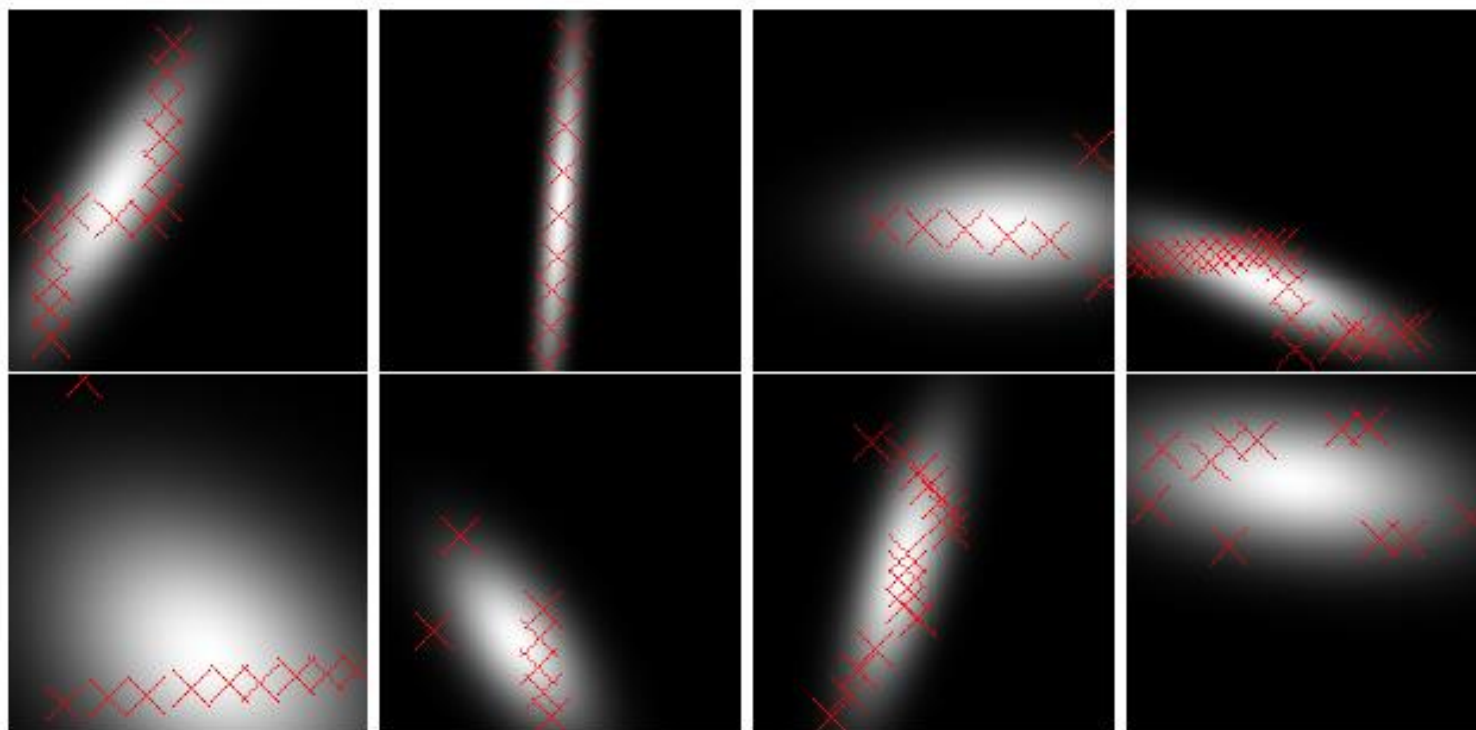
- grid cell 단위로 PCD 를 분할하고 NDT 를 적용하면 좌측 그림과 같이 cell 경계에서 point span 이 끊어지는 문제점이 발생함
- PCD 의 모든 영역에 대해서 서로 겹치는 4개의 grid cell 을 만들어 문제점을 해결함



NDT 적용시 고려사항들

❖ Noise free measured world lines

- singular covariance matrix \rightarrow NDT 확률을 계산할 수 없게 됨
- smaller eigenvalue 가 larger eigenvalue $\times 0.001$ 보다 작으면 이 값으로 강제 설정함



Spatial Mapping T

- ❖ 2개의 서로 다른 좌표계의 상대적인 관계를 나타낸 것으로 수평 이동과 회전 이동을 가지고 있음

$(x', y') : \text{Robot}_1 \text{ Frame in Robot}_2 \text{ Frame}, (x, y) : \text{Robot}_1 \text{ Frame}$

$$T : \begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix}$$
- ❖ first scan : reference points
- ❖ second scan : reference points 로 match 될 source points
- ❖ $\mathbf{p} = (p_i)_{i=1..3}^t = (t_x, t_y, \phi)^t$: reference points 에 대한 source points 의 spatial mapping T 의 추정 parameters
- ❖ \mathbf{x}_i : source points 중 i 번째 point
- ❖ \mathbf{x}'_i : reference points 좌표계로 변환한 source points 중 i 번째 point
 - $\mathbf{x}'_i = T(\mathbf{x}_i, \mathbf{p}) = \begin{pmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{pmatrix} \begin{pmatrix} x_i \\ y_i \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix}$
- ❖ Σ_i, q_i : \mathbf{x}'_i 가 속하는 grid Cell 이 가지고 있는 covariance matrix 와 mean

NDT Scan Matching

1. reference points 에 대한 NDT 계산
2. spatial mapping T 에 대한 parameters \mathbf{p} 초기화
 - $(t_x, t_y, \emptyset) = (0, 0, 0)$: source points 와 reference points 가 동일 좌표계
 - (t_x, t_y, \emptyset) : calculation by using odometry
3. \mathbf{p} 를 통해 계산한 spatial mapping T 를 적용하여 source points 를 reference points 의 좌표계로 변환 : $\mathbf{x}'_i = T(\mathbf{x}_i, \mathbf{p})$
4. 각 source point 가 속하는 grid cell 을 찾고 해당 grid cell 의 mean 과 covariance matrix 를 결정 : Σ_i , q_i
5. parameters 평가
 - $score(\mathbf{p}) = \sum_i \exp(-\frac{1}{2}(\mathbf{x}'_i - q_i)^t \Sigma_i^{-1}(\mathbf{x}'_i - q_i))$
 - 각각의 \mathbf{x}'_i 가 해당 grid cell 에 속할 확률을 모두 더한 것으로 score 값이 높을수록 parameters 즉, 변환 행렬이 정확하다는 것을 의미
6. new estimate parameters 를 계산
 - Newton's Algorithm 을 통해 $score(\mathbf{p})$ 를 최적화하는 parameters 계산
7. new estimate parameters 를 가지고 수렴 기준을 만족할 때까지 3~6 과 정 반복

Map Evaluation Criteria 1 (Javanmardi et al)

❖ Feature sufficiency

● feature

❖ map 이 가지고 있는 정보

- point cloud map : points
- normal distribution map : ND cells (with mean, variance)
- occupancy grid map : occupied cells

● accurate localization

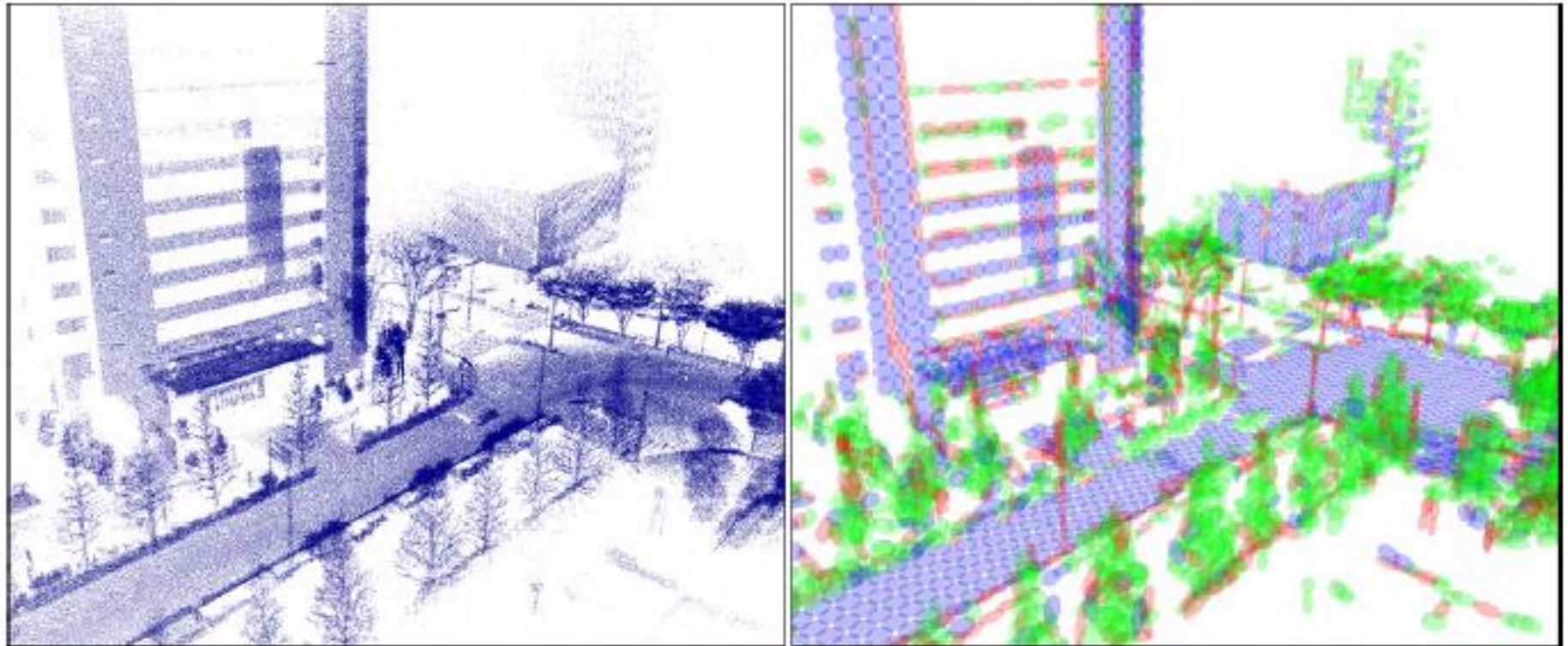
❖ plenty and high-quality feature 필요함

- map 이 가지고 있는 feature 개수가 많아야 함 → feature_count
- localization 을 하기 쉬운 특성을 가진 feature 가 많아야 함 → dimension_ratio

- 예: 평지보다는 건물, 나무, 신호등 등 다양한 Feature 가 있는 환경

Map Evaluation Criteria 1 (Javanmardi et al)

- ❖ Feature sufficiency
 - dimension_ratio



1D feature



2D feature

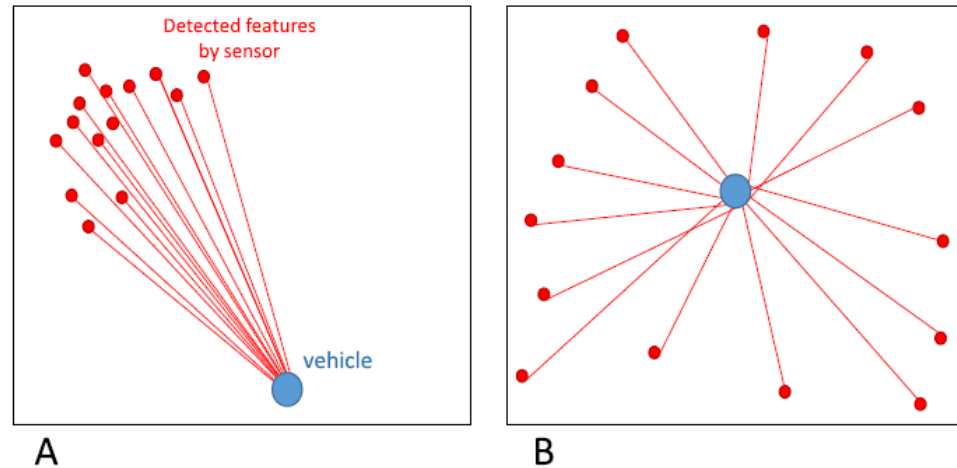


3D feature

Map Evaluation Criteria 2 (Javanmardi et al)

❖ Layout

- Feature 들의 배치

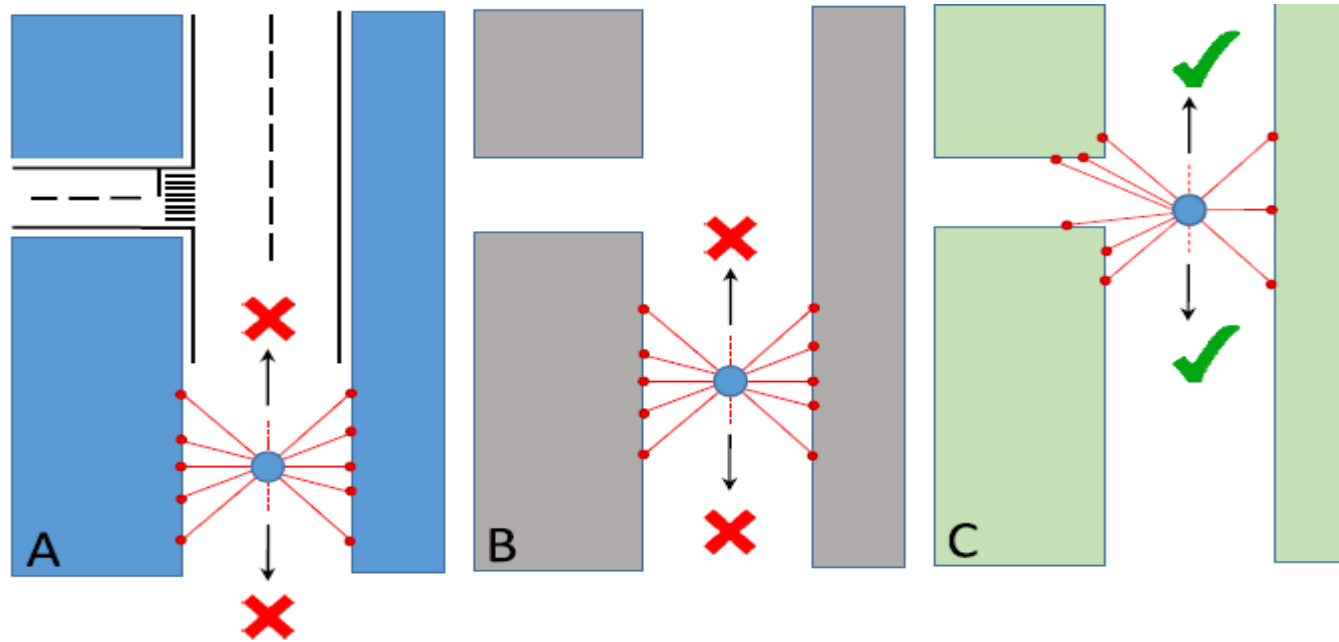


- feature sufficiency (plenty and quality) 를 만족하더라도 feature 들이 한 곳에 모여있으면 localization accuracy 를 낮춤 → FDOP (feature dilution of precision)

Map Evaluation Criteria 2 (Javanmardi et al)

❖ Layout

- 터널 및 고속도로와 같은 환경

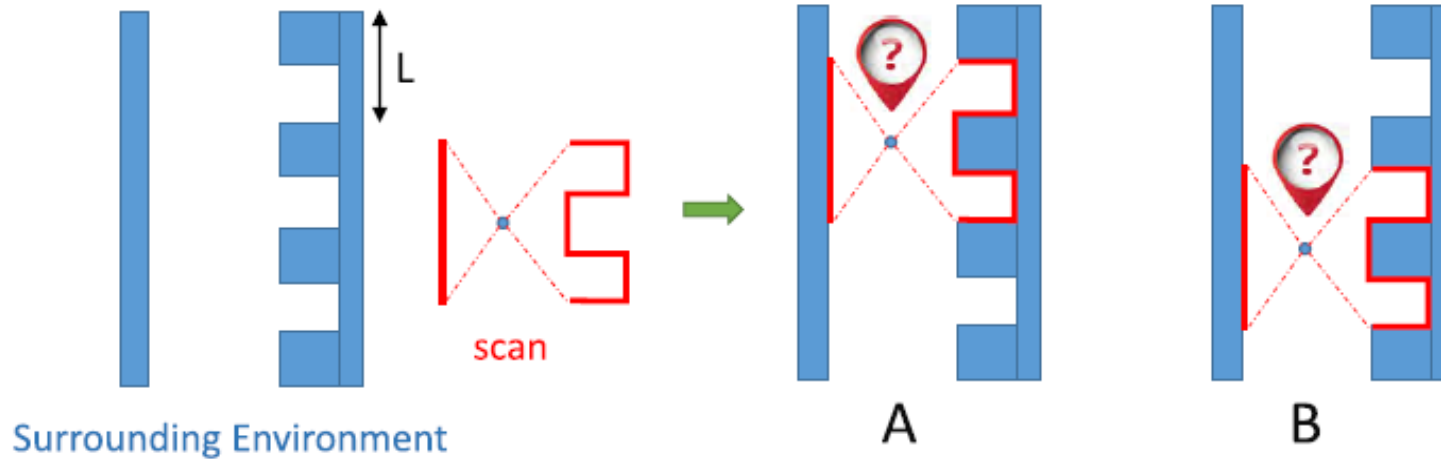


- feature 가 고르게 분포되어 있지만 longitudinal 혹은 lateral 힌트가 부족하여 localization 하기 어려움 → normal_entropy

Map Evaluation Criteria 3 (Javanmardi et al)

❖ Local similarity

- feature 형태의 유사성



- feature 의 quality 및 layout 을 만족했음에도 불구하고 feature 간의 유사성으로 인해 localization 을 하기 어려움 → score_entropy

Map Evaluation Criteria 4 (Javanmardi et al)

❖ Representation quality

- map 이 실제 환경과 얼마나 유사한가를 나타냄

❖ map format

- point cloud : points → high quality
- occupancy grid : occupied cells → relatively low quality

❖ resolution

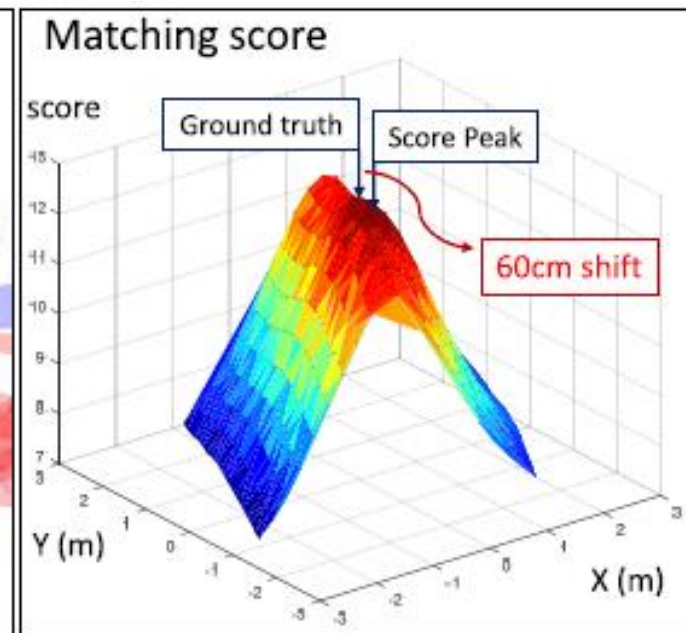
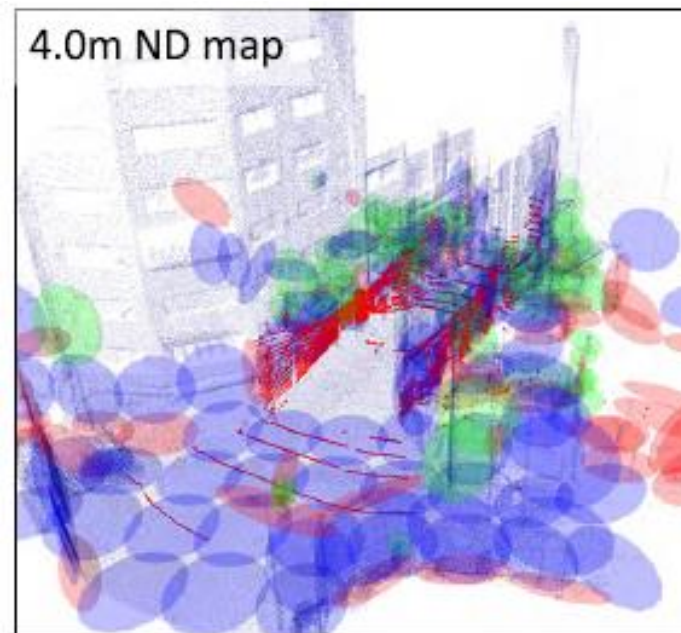
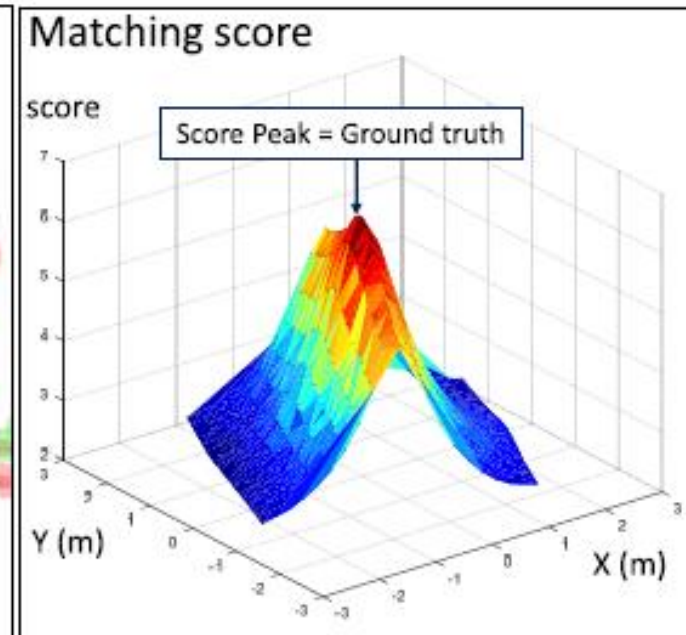
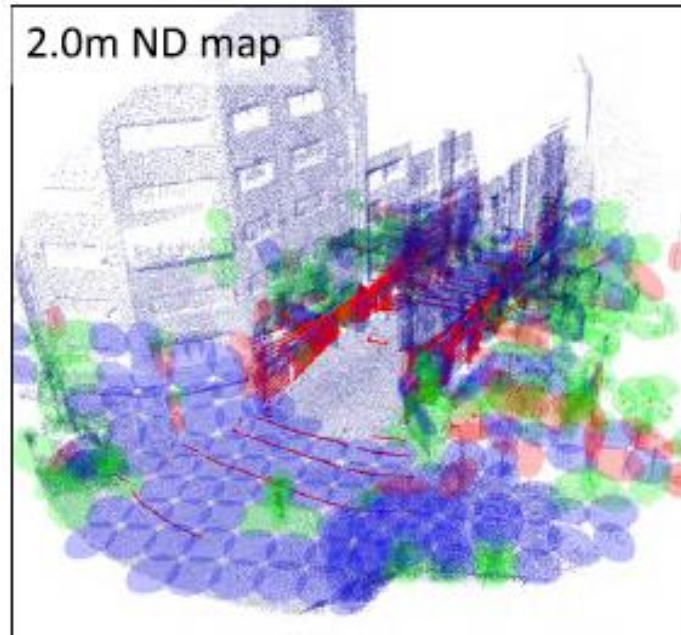
- ND map 0.2m : high quality
- ND map 1.0m : relatively low quality

❖ LiDAR channel, parameters

- 64 channels vs. 16 channels
- frequency, range, number of sensors, position of sensor

Map Evaluation Criteria 4 (Javanmardi et al)

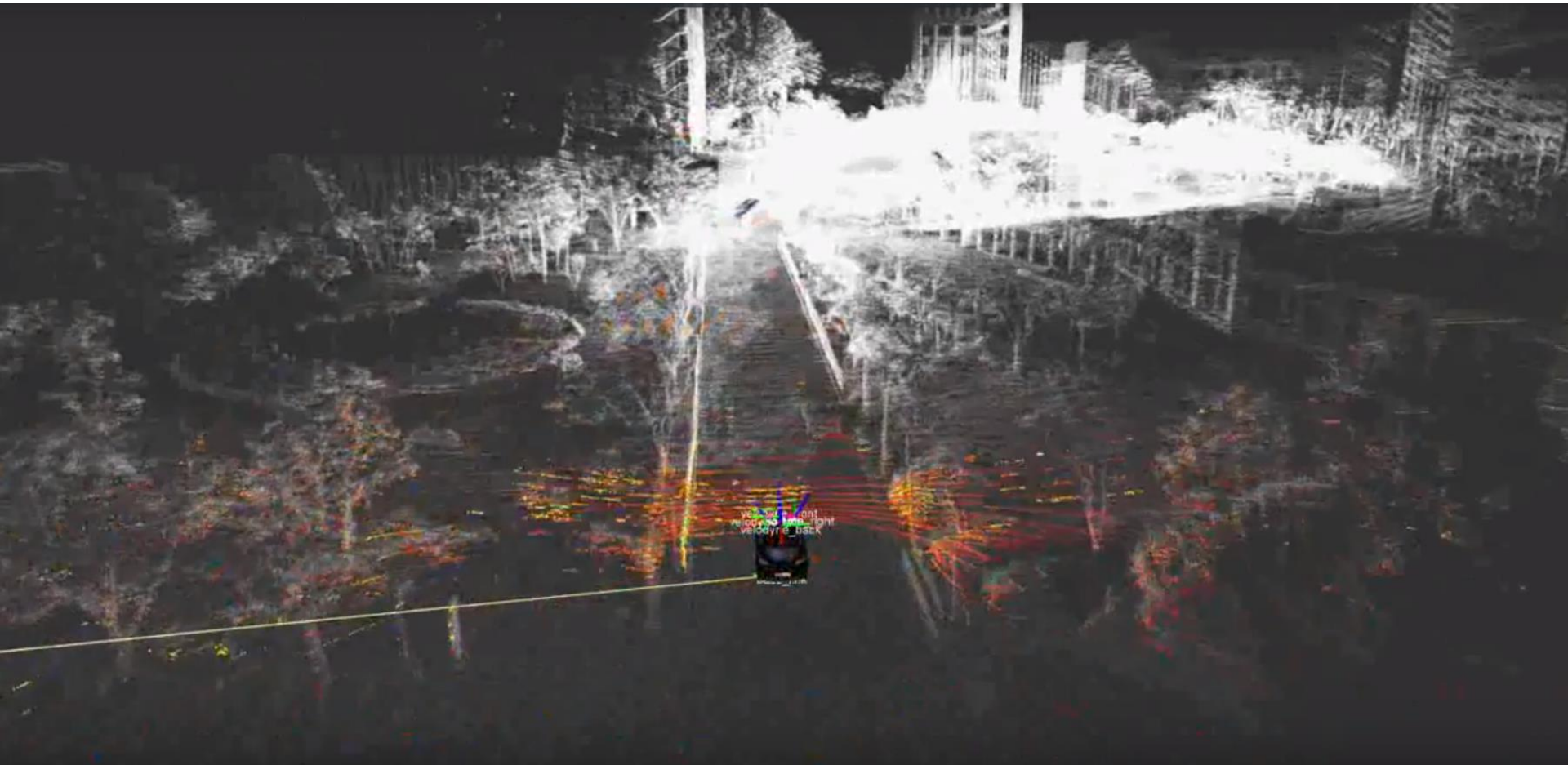
- ❖ Representation quality



Point Map

Point Map

- ❖ 3차원 공간 상에 존재하는 모든 사물들을 점들로 표현한 지도
- ❖ 자차 위치 추정 알고리즘의 입력으로 사용됨
- ❖ <https://www.youtube.com/watch?v=3SHEGn33dsQ>



Point map 생성

- ❖ 1단계: pcap 분할 및 병합
 - 장시간 record된 pcap 파일이 있으면 여러 개 파일로 분할함
 - ❖ `$ tcpdump -r input.pcap -w output.pcap -C 100 #100MB단위 분할`
 - 분할된 파일이 A, B, C, D, E 라고 하면 AB, BC, CD, DE 로 다시 병합함
 - ❖ `$ mergecap -w merge.cap output.pcap1 output.pcap2`

Point map 생성

❖ 2단계: 부분 PCD 생성

- 이 슬라이드는 pcap-based 로 설명함 (rosbag-based 도 가능)
- AB.cap, BC.cap, CD.cap, DE.cap 파일 각각에 대해서 ndt mapping 수행 하여 AB.pcd, BC.pcd, CD.pcd, DE.pcd 파일들을 생성함
- ndt mapping 실행 절차
 - ❖ \$ roslaunch runtime_manager runtime_manager.launch
 - ❖ \$ roslaunch nodelet_lidar_driver nodelet_lidar_driver.launch
 - ❖ runtime_manager > computing > lidar_localizer > ndt_mapping
체크
 - ❖ rviz > frame : velodyne > points_raw 를 선택하여 pcap play 가 끝나면 nodelet_lidar_driver 를 종료함 (미 종료시 ndt_mapping 에러 발생)
 - ❖ runtime_manager 와 함께 실행된 gnome-terminal 에서 ndt_mapping 출력이 종료했는지 확인
 - ❖ runtime_manager > computing > lidar_localizer > ndt_mapping > app 선택 후 "PCDOUT" 체크하여 *.pcd 파일 생성

Point map 생성

❖ 3단계: 부분 PCD 들을 병합하여 하나의 PCD 생성 → Cloudcompare

동영상

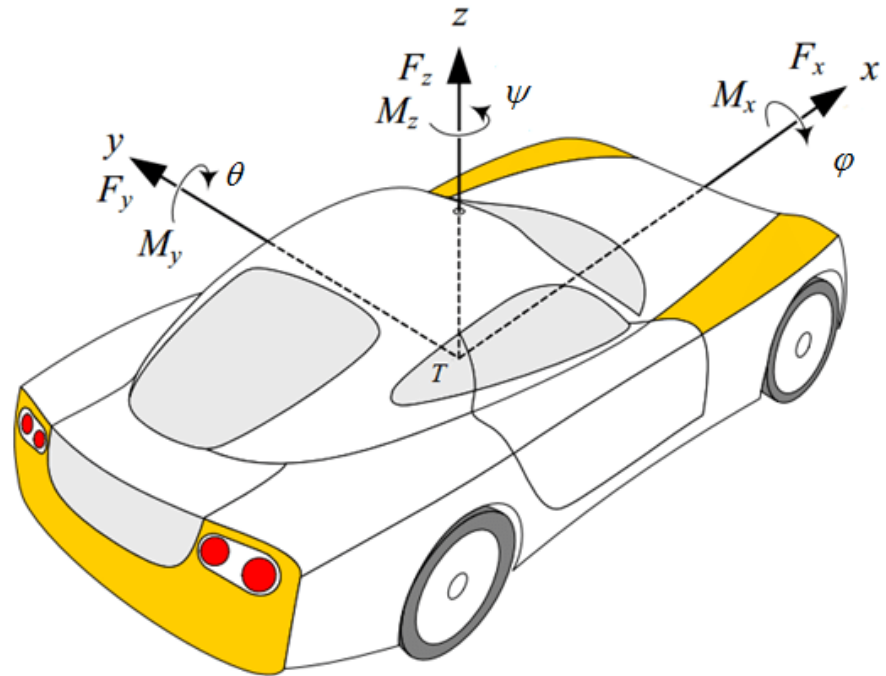
Point map 생성

- ❖ 3단계: 부분 PCD 들을 병합하여 하나의 PCD 생성
 - Cloudcompare 에서 생성한 포인트 맵을 .ply 확장자로 저장함
 - ❖ .pcd 확장자로 저장 시 (x, y, z, intensity) 순서가 autoware가 요구하는 순서와 다르게 저장되어 autoware에서 사용 불가
 - vi 편집기를 이용하여 .ply 파일 안에 scalar_intensity 필드명을 → intensity 로 명칭 변경
 - ❖ \$ vi *.ply
 - pcl library 를 이용하여 .ply -> .pcd 형식으로 변환함
 - ❖ \$./pcl/build/bin/pcl_ply2pcd input.ply output.pcd
 - 최종 결과물 output.pcd 를 autoware 에서 적재해서 사용할 수 있음

Local coordinate system

❖ local coordinate system

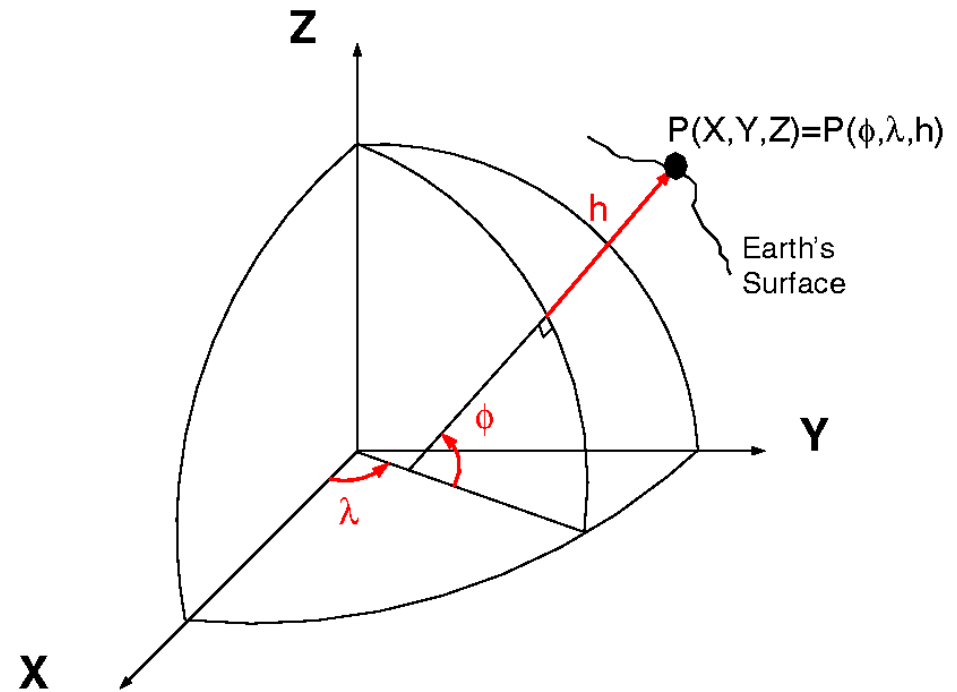
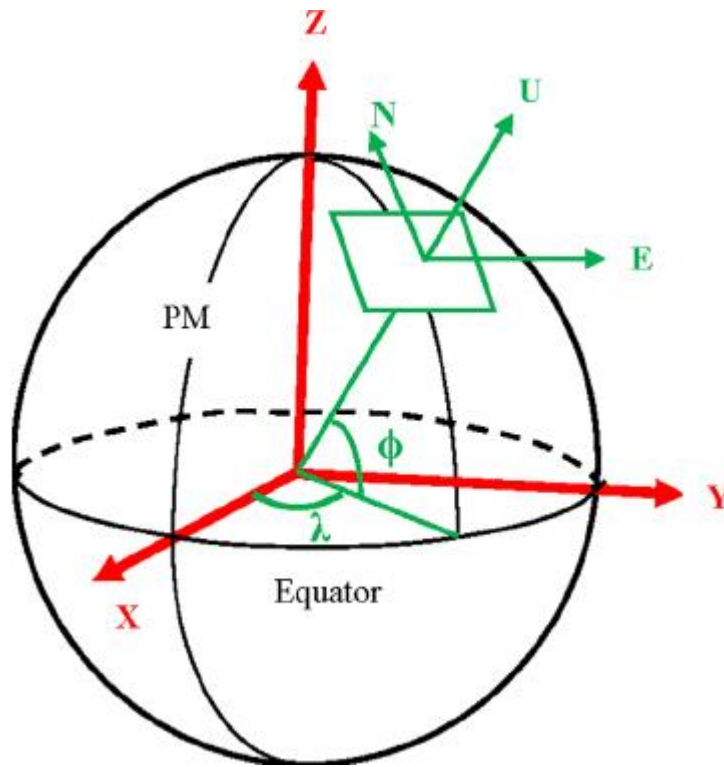
- 원점은 pcap 파일 수집이 시작된 지점으로 정의됨
- point map 의 좌표 축들은 초기 차량의 포즈에 의해서 결정됨
 - ❖ x 축 : 차량 전방
 - ❖ y 축 : 차량 좌측방
 - ❖ z 축 : 차량 지붕 방향



WGS84 (World Geodetic System 84)

❖ WGS 84

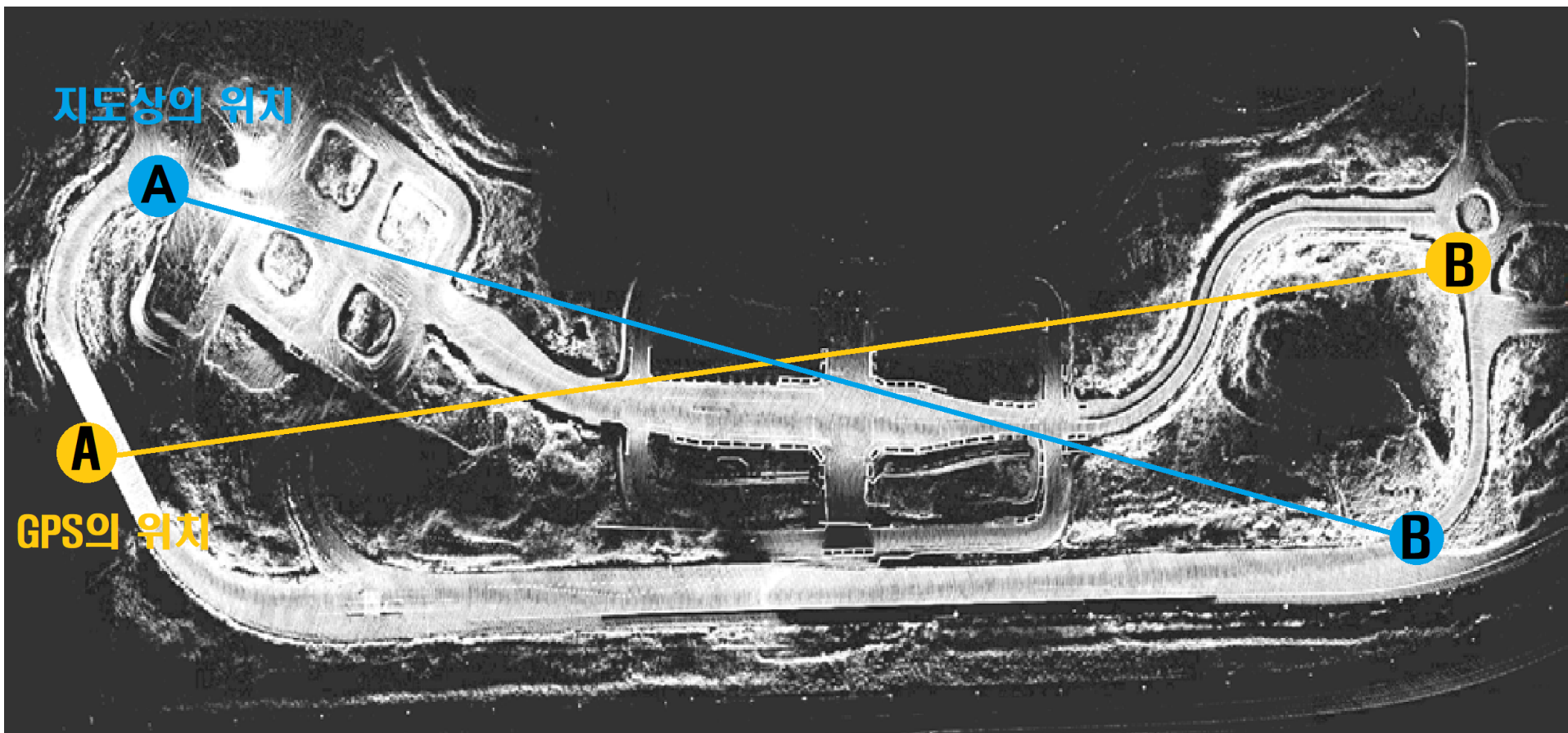
- GPS 에서 사용되는 표준 좌표계
- 원점은 지구 질량 중심으로 정의함
- 지구 표면 상의 한 점 P는 (위도, 경도, 고도) = (lat, lon, alt) 로 정의함



LCS 와 WGS84 의 동기화

❖ LCS 와 WGS84 사이의 변환 행렬 유도

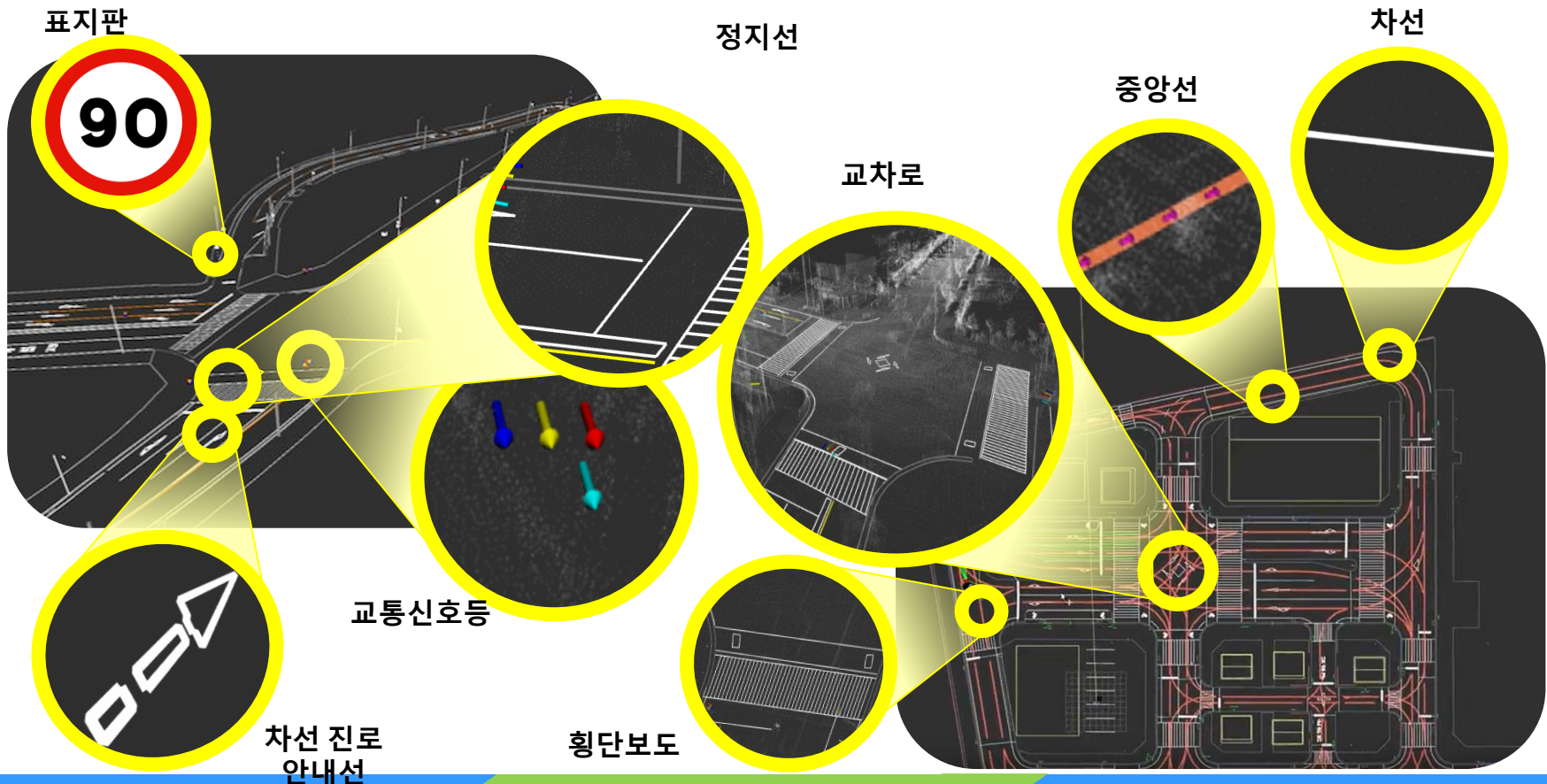
- point map 상의 여러 점들에 대해서 ndt matching 좌표 (LCS)와 GPS 좌표 (WGS84) 쌍들을 얻어냄
- 이 좌표 쌍들을 이용해서 변환 행렬을 유도함



Vector Map

Vector Map

- ❖ 점, 선, 면 등을 벡터 형식으로 표현한 논리적 교통 정보 지도 (3D point map과 함께 사용)
- ❖ 차선 정보, 차선 중앙, 정지선, 횡단보도, 신호등, 표지판 등의 도로정보
- ❖ 경로 계획 알고리즘의 입력으로 사용됨



NGII vs. Autoware

❖ NGII's vector map

- .shp 형식의 벡터 맵
- 주행 유도선은 경로점들의 나열 형태가 아니며, 교차로 구간에 대해서는 주행 유도선 자체를 제공하지 않음
- NGII는 특정 자율주행 SW 플랫폼에 의존적인 방식으로 벡터 맵을 제작하지 않음

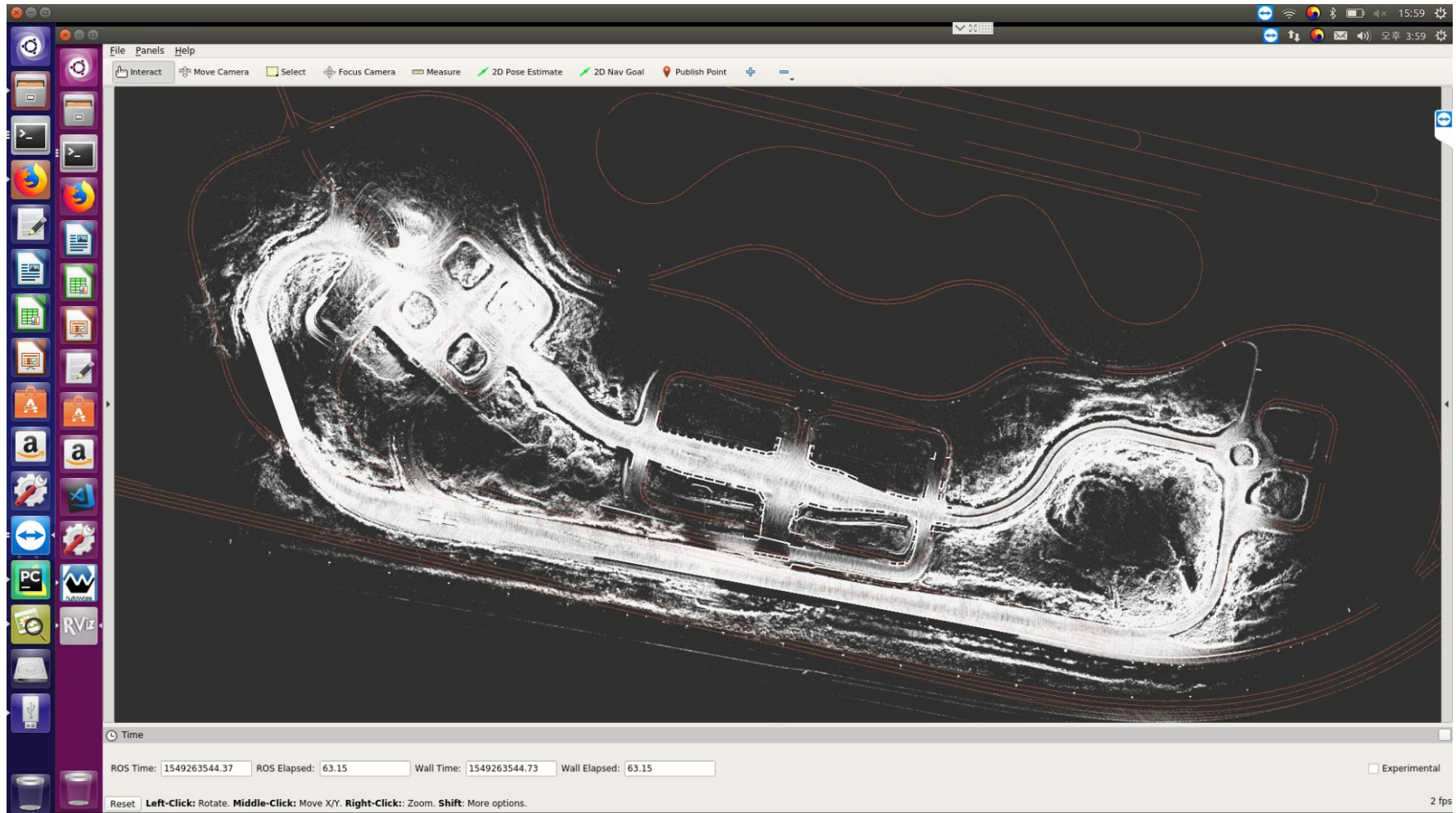
	NGII	Autoware
정보형태	연속(.shp)	이산(.csv)
좌표계	WGS84	LCS
방향성, 곡률	없음	있음
교차로	끊어짐	연결됨

❖ Autoware's vector map

- .csv 형식의 벡터 맵
- 주행 유도선(dtlane), 차선(line), 횡단보도(cross), 정지선(stopline) 등 포함
- 주행 유도선은 Autoware의 기준 경로와 주행 궤적 생성에 필수적
 - ❖ 주행 유도선은 경로점들의 나열로 표현함
 - ❖ 모든 경로점들은 등간격(약 1 미터)으로 배치되어야 함
 - ❖ 차량이 교차로를 건너가기 위해서는 하나의 경로점이 2개 이상의 다른 경로점들로 분기하도록 미리 연결 링크가 만들어져 함
 - ❖ 곡선로 상에 위치하는 경로점들은 차량의 조향휠이 급격하게 회전하는 일이 없도록 완만하게 변화하는 곡률을 가져야 함

NGII's vector map vs. Our point map

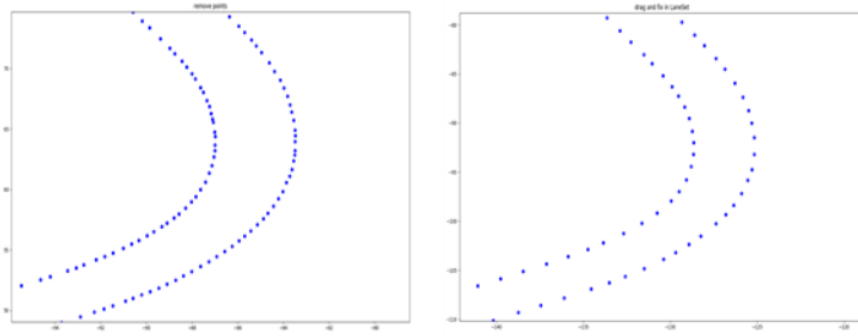
- ❖ 문제점: NGII's vector map과 우리의 point map 은 동기화되지 않음
 - 해결방법: 우리의 point map 안에서 vector map을 추출해야 함



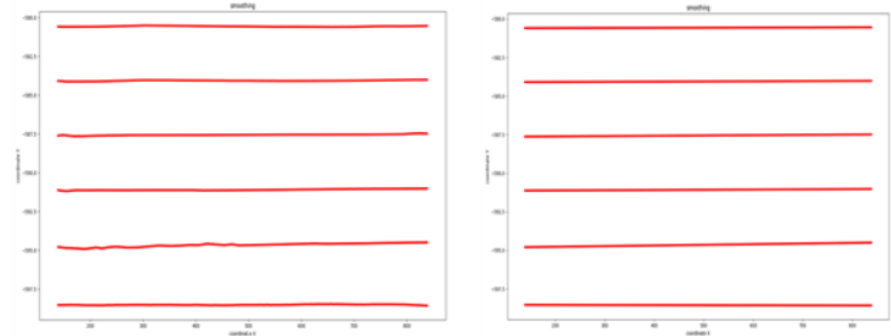
dtlane 생성 방법

- ❖ 단계 0: **rosvag 파일 생성**
 - autoware 실행 중 /ndt_pose 의 메시지들을 rosvag 파일로 저장함
- ❖ 단계 1: rosvag 파일을 읽어 **경로점들을 약 1미터의 등간격으로 조정함**
 - 곡선로 상의 경로점들이 간격이 짧을 때 방향 벡터의 순간 변화율이 커짐으로 인해 발생하는 조향휠의 급회전을 방지하기 위함
 - Autoware는 또한 최단 경로를 탐색할 때 경로점들의 개수로 경로의 길이를 판단함
- ❖ 단계 2: **직선로 상의 경로점들의 방향 벡터를 평탄화함**
 - 이 또한 조향휠의 급회전을 방지하기 위함
- ❖ 단계 3: **교차로에서 미연결상태인 주행유도선들을 수작업으로 상호 연결함**
 - 이 때 교차로 상에서 좌회전 또는 우회전을 통해서 서로 연결되는 주행유도선들에 대해서는 곡선 유도선을 끼워넣어서 연결함
 - 곡선 유도선은 라인 샘플링(line sampling) 기법 등을 사용하여 완만한 곡률을 갖는 곡선 유도선으로 생성함
- ❖ 단계 4: **포인트 맵을 참조하여 경로점들을 수작업으로 미세 조정함**
 - 차로 중앙에 경로점들이 위치하도록 주행 유도선을 미세 조정함

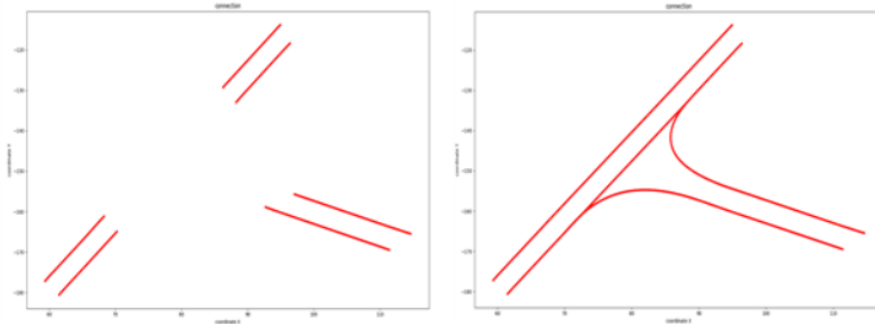
dtlane 생성 방법



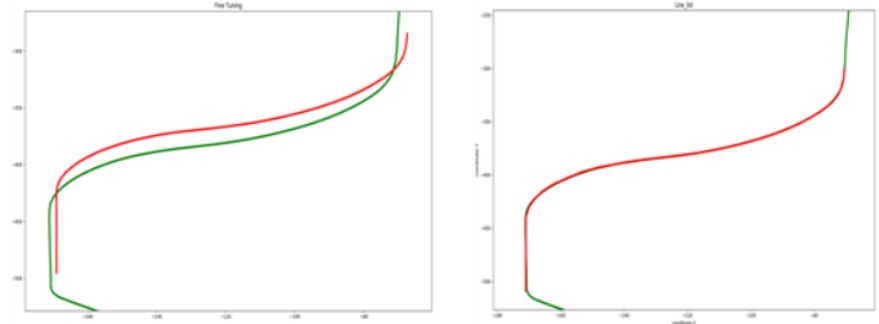
(a) 단계 1: 경로점들을 등간격으로 조정하기



(b) 단계 2: 방향 벡터 평탄화(smoothing)하기

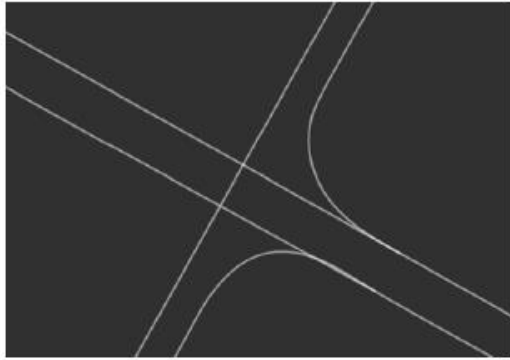


(c) 단계 3: 교차로에서 미연결 상태인 주행
유도선들을 상호 연결하기

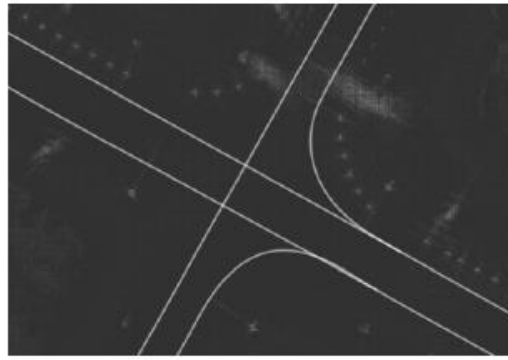


(d) 단계 4: 차로 중앙선(적색)과 실제 주행
경로(녹색)를 동기화하기

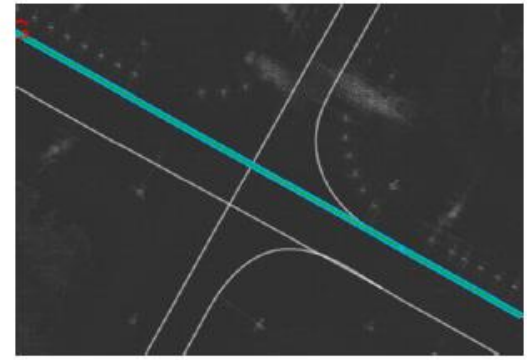
dtlane 생성 방법



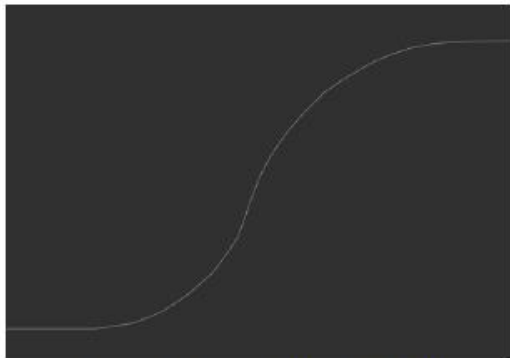
(a) 교차로 구간 벡터 맵



(b) 포인트 맵과 동기화한
교차로 구간 벡터 맵



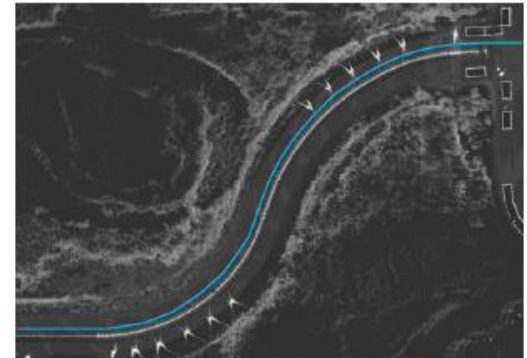
(c) 교차로 구간의 경로 계획
결과 화면



(d) 곡선로 구간 벡터 맵



(e) 포인트 맵과 동기화한
곡선로 구간 벡터 맵



(f) 곡선로 구간의 경로 계획
결과 화면


Vector map 생성

- ❖ 차량 주행 궤적을 저장한 rosbag 파일 생성
 - pcap or rosbag 파일을 기반으로 ndt matching 을 실행시켜 /ndt_pose 토픽을 발행함
 - /ndt_pose 메시지들을 rosbag 파일로 저장함
 - ❖ \$ rosbag record /ndt_pose → 20190807_ndtpose.bag 생성
- ❖ vectormapper.py 실행 (**dtlane 생성 방법의 1단계만 구현함!!**)
 - \$ cd ~/shared_dir/mapping_demo/vectormapper
 - \$ vi vectormapper.py # 다음 변수값들을 설정함
 - bagPath =
'/home/autoware/shared_dir/mapping_demo/vectormapper/rosbag/'
 - bagName = '20190807_ndtpose.bag'
 - \$ python vectormapper.py
 - \$ cd result
 - \$ ls → dtlane.csv, idx.csv, lane.csv, node.csv, point.csv 파일들 확인됨
- ❖ runtime_manager 에서 op_global_planner 실행하여 위 csv 파일들로 global planning 이 가능한지 검증함

Vector map 생성

❖ point.csv

(x, y, z)



```
PID,B(Lat),L(Long),H,Bx,Ly,ReF,MCODE1,MCODE2,MCODE3
10001,0,0,2.755,-222.116,263.56,0,0,0,0
10002,0,0,2.747,-221.99,263.34,0,0,0,0
10003,0,0,2.728,-221.526,262.494,0,0,0,0
10004,0,0,2.726,-221.06,261.556,0,0,0,0
10005,0,0,2.714,-220.557,260.689,0,0,0,0
10006,0,0,2.706,-220.059,259.732,0,0,0,0
10007,0,0,2.7,-219.557,258.865,0,0,0,0
10008,0,0,2.677,-219.021,257.947,0,0,0,0
10009,0,0,2.67,-218.508,256.965,0,0,0,0
10010,0,0,2.66,-217.886,256.044,0,0,0,0
10011,0,0,2.653,-217.474,255.113,0,0,0,0
10012,0,0,2.634,-216.925,254.183,0,0,0,0
10013,0,0,2.624,-216.41,253.25,0,0,0,0
10014,0,0,2.605,-215.864,252.258,0,0,0,0
10015,0,0,2.595,-215.33,251.354,0,0,0,0
10016,0,0,2.591,-214.838,250.491,0,0,0,0
10017,0,0,2.58,-214.204,249.493,0,0,0,0
10018,0,0,2.573,-213.644,248.512,0,0,0,0
10019,0,0,2.555,-213.11,247.563,0,0,0,0
```

Vector map 생성

❖ dtlane.csv

direction curvature = 1 / (direction_k - direction_{k-1})

```

DID,Dist,PID,Dir,Apara,r,slope,cant,LW,RW
10001,0,10001,2.639769234710163,0,90000000000,0,0,2,2
10002,1,10002,2.6376132093554716,0,-463.81643788372116,0,0,2,2
10003,2,10003,2.623493467365048,0,-70.82282386450235,0,0,2,2
10004,3,10004,2.6190860214699203,0,-226.88877499449444,0,0,2,2
10005,4,10005,2.6193552815625396,0,3713.8812152675555,0,0,2,2
10006,5,10006,2.6169194991207694,0,-410.54569687810056,0,0,2,2
10007,6,10007,2.6168200537585777,0,-10055.773119644313,0,0,2,2
10008,7,10008,2.611935181969462,0,-204.7136635659855,0,0,2,2
10009,8,10009,2.6113036291765064,0,-1583.3989037083952,0,0,2,2
10010,9,10010,2.607849747192196,0,-289.529290387627,0,0,2,2
10011,10,10011,2.6089893455271644,0,877.5021595900062,0,0,2,2
10012,11,10012,2.606108828401946,0,-347.15988710678954,0,0,2,2
10013,12,10013,2.601547180585499,0,-219.21902791235038,0,0,2,2
10014,13,10014,2.59935461532176,0,-456.0867658254959,0,0,2,2
10015,14,10015,2.597287716731973,0,-483.8166734164649,0,0,2,2
10016,15,10016,2.597419247358445,0,7602.792040311393,0,0,2,2
10017,16,10017,2.5999973546014803,0,387.881459431744,0,0,2,2
10018,17,10018,2.603056334167815,0,326.90640075058644,0,0,2,2
10019,18,10019,2.604540412696646,0,673.8187909690137,0,0,2,2
  
```