◎ 논 문

http://dx.doi.org/10.22680/kasa2021.13.4.123

# 최대우도법을 이용한 라이다 포인트군집의 박스특징 추정

김종호* · 이경수**,†

# Box Feature Estimation from LiDAR Point Cluster using Maximum Likelihood Method

Jongho Kim*, Kyongsu Yi**,†

*Key Words: Autonomous Driving(자율주행), LiDAR Point Cloud(라이다 포인트 클라우드), Box Feature(박스특징), Maximum Likelihood Method(최대우도법)*

ABSTRACT

This paper present box feature estimation from LiDAR point cluster using maximum likelihood Method. Previous LiDAR tracking method for autonomous driving shows high accuracy about velocity and heading of point cluster. However, Assuming the average position of a point cluster as the vehicle position has a lower accuracy than ground truth. Therefore, the box feature estimation algorithm to improve position accuracy of autonomous driving perception consists of two procedures. Firstly, proposed algorithm calculates vehicle candidate position based on relative position of point cluster. Secondly, to reflect the features of the point cluster in estimation, the likelihood of the particle scattered around the candidate position is used. The proposed estimation method has been implemented in robot operating system (ROS) environment, and investigated via simulation and actual vehicle test. The test result show that proposed cluster position estimation enhances perception and path planning performance in autonomous driving.

## 1. 서 론

자율주행 및 ADAS 기술은 현재 차량과 관련하여 가장 크게 연구되고 있는 분야이다. 초기연구에는 레이더(Radar) 및 초음파 센서를 이용한 장애물의 존재여부 파악과 같은 수준이었지만, 최근에 연구되는 자율주행 및 ADAS 기술은 유연한 상황판단과 능동적인 제어를 위해 주변 차량에 대한 높은 위치 정확도를 요구한다.

라이다(LiDAR) 센서는 레이저 광원을 이용하여 물체에서 반사되는 빛을 분석하는 인지장비이다. 레이더 및 초음파 센서에 비해 인지 거리가 길고 분해능이 높다는 장점이 있어, 양산차량의 ADAS를 위한 센서로 많이 적용되고 있다. 라이다 센서의 측정결과는 포인트 클라우드로 표현되기 때문에, 시간에 따른 포인트 클라우드 입력으로부터 이동물체를 탐지, 추적하는 알고리즘이 제안되고 있다. 이동물체의 탐지, 추적을 위한 접근방법은 크게 두가지로 나누어진다. 첫번째는 단일 프레임의 라이다 포인트 클라우드로부터 이동물체를 탐지하는 방법이다. 대표적으로 Najibi, M.가 제안한 방법이 있다.[1] 이는 인공신경망을 이용한 3D 단일프레임 탐지기(Single stage detector)이며, 차량으로부터 반사되는 포인트 클라우드의 수가 큰 128 채널의 라이다 센서를 이용한 Waymo Open 데이터셋으로 검증되었다. 하지만 대부분의 양산을 위한 라이다 센서는 16채널 이하의 레이어를 갖기 때문에, 차량에 반사되

* 서울대학교 기계공학부, 학생
** 서울대학교 기계공학부, 교수
†교신저자: kyi@snu.ac.kr
E-mail: kimjhmj@snu.ac.kr

자동차안전학회지: 제13권, 제4호, pp. 123~128, 2021
논문접수일: 2021.5.12, 논문수정일: 2021.9.9, 게재확정일: 2021.10.4

는 포인트의 수가 작다. 따라서 단일프레임에 대해 인공신경망 기반 방법을 사용하기에 많은 어려움이 있다. 두번째 접근방법으로는 별도로 차량을 탐지하지 않고, 포인트 클라우드를 군집화하여 모든 군집의 이동을 추적하는 방법이다. GMFA-PF로 제안된 H, Lee.의 방법은 물체의 기하학적 형상을 고려하지 않고 군집화하여 파티클필터에 기반해 군집의 상태를 추적하였다.[2]

상기한 것과 같이 라이다 포인트 군집를 추적하는 방법은 차량에 대한 탐지가 먼저 이루어지지 않기 때문에, 포인트 군집으로부터 차량의 실제 위치를 추정하는 방법이 요구된다. 한가지 예로 단일 프레임의 포인트 클라우드로부터 라이다 센서의 폐색된 영역을 이용해 바운딩박스를 추출하는 알고리즘이 제안되었다.[3] 이는 포인트 군집을 감쌀 수 있는 가장 작은 바운딩 박스를 추출하였지만 요각 추정 정확도가 떨어지는 결과를 보였다.

그러므로 이 논문에서는 추적중인 차량의 포인트 군집을 입력하여 최대우도법에 기반하여 박스형태로 가정한 차량의 중심을 추정하는 알고리즘을 제안한다. 알고리즘은 두가지 단계가 순차적으로 수행된다. 2.1장에서는 포인터 군집과 자차량의 상대위치를 고려하여 후보위치를 계산하는 방법을 소개한다. 2.2장에서는 후보위치로부터 LiDAR센서의 오차를 고려하여 가장 큰 우도를 갖는 차량 중심을 추정한다. 3장에서는 알고리즘의 구현과 시뮬레이션 실험환경을 설명하며, 4장에서는 실험결과를 기존의 방법들과 비교하여 분석한다(Fig. 1).

## 2. 라이다 포인트군집 박스특징 추정

이 장에서는 추적되는 라이다 포인트 군집으로부터 박스특징을 추정하는 방법을 소개한다. 포인트 군집의 상태 추정 알고리즘은 상기한 GMFA-PF방법을 이용한다. 이 알고리즘은 이동물체를 자 차량 좌표계에서의 위치, 요 각과 절대 좌표계에서의 속도, 각속도 상태를 추정하며, 자 차량 좌표계의 포인트 군집을 반환한다. 적은 레이어를 갖는 LiDAR센서에서도 높은 차량 재현율(Recall)을 보이며 파티클필터를 이용하기 때문에 높은 속도와 요 각 추정성능 보인다.

박스특징은 차량의 높이와 넓이가 정해진 회전가능한 강체로 가정하는 것으로, 요 각과 위치를 통해 표현될 수 있다. 이 논문에서는 일반차량과 버스의 두가지 높이와 넓이를 정의하여 사용한다.

### 2.1. 상대위치 기반 후보 박스특징 도출

LiDAR 센서는 레이저 광원을 이용하기 때문에, 차량에 대해 중심이 아닌 자차량과 가장 가까운 방향에서 반사된다. 그러므로 포인트 군집과 자차량의 상대적인 위치는 차량의 실제 위치를 포함하는 박스특징과 밀접한 관련이 있다.

추적하는 포인트 군집이 정해진 길이 h와 너비 w를 갖는 박스 형상을 측정했다면, 자차량으로부터 나온 가상의 대표 광원이 박스 형상과 가장 가까운 점에서 반사되었을 것이다. 이 때 박스형상의 요 각은 추적알고리즘에서 추정하는 요 각을 따른다. 따라서 Fig. 1과 같이 포인트 군집의 평균위치에서 반사되었다고 가정하여, 가상의 광원을 도시하면 차량의 박스특징에 대한 후보위치를 나타낼 수 있다.

Fig. 2에 나타낸 그림을 식으로 표현하면, 자 차량 좌표계에서 GMFA의 포인트 군집을 향하는 가상의 광원의 각도를 $\theta_{vray}$, GMFA의 이동 물체 좌표계에서 자 차량을 향하는 각도를 $\theta_{ego}$ 하고, 포인트 군집들의 평균 위치를 $p^{GMFA}$ 이라 하면 식 (1)과 같이 후보 위치가 도출되어진다.

$$
\begin{aligned}
\theta_{box} &= \tan^{-1}(w/h) \\
d_{box} &= \begin{cases} \left|(\frac{w}{2})/\sin(\theta_{ego})\right| & \text{if } (\pi\text{-}\theta_{box}) > |\theta_{ego}| > \theta_{box} \\ \left|(\frac{h}{2})/\cos(\theta_{ego})\right| & \text{else.} \end{cases} \\
p^{x,box} &= p^{x,GMFA} + d_{box}\cos(\theta_{vray}) \\
p^{y,box} &= p^{y,GMFA} + d_{box}\sin(\theta_{vray})
\end{aligned}
\tag{1}
$$

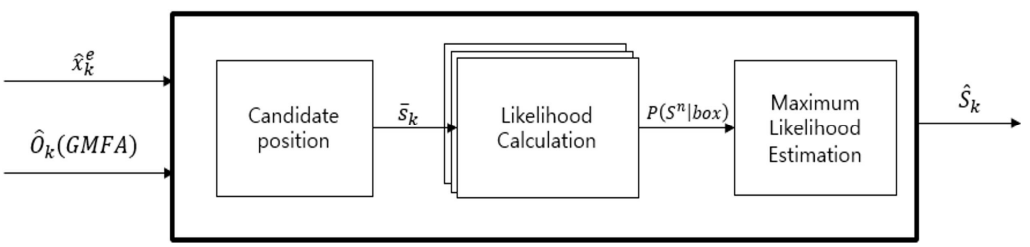



Fig. 1 Two stage procedures of proposed box feature estimation system

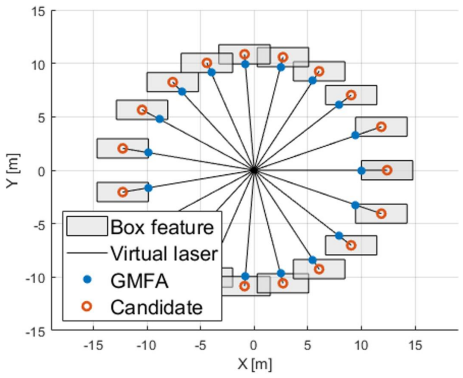



Fig. 2 Box feature and virtual laser by relative position of GMFA

## 2.2. 최대우도법을 이용한 최적 박스특징 탐색

2.1에서 계산한 후보위치는 본문에서 제안하는 적은 수의 점을 갖는 포인트 군집의 추정에 간단한 계산만으로 작동한다. 하지만 일정거리 이내로 가까워진 이동 물체의 포인트 군집은 포인트 수 가 충분히 많고 그 기하학적 분포도 다양하게 나타난다. Fig. 3은 동일한 Ground truth를 갖는 이동 물체에 대해 LiDAR 센서가 측정할 수 있는 다양한 포인트 군집의 형태 예를 보여준다.

그러므로 일정거리 이내 또는 일정 수 이상의 포인트 군집에 대해서 정밀한 박스특징 추정을 위해 기하학적 형태를 반영하는 과정이 필요하다. 라이다 센서는 광원으로부터 거리에 대해 측정오차를 갖는다. 또한 본문에서 제시한 직사각형 형태의 박스 가정은 실 차량의 표면에 대해 형상 오차를 갖는다. 그러므로 이를 일반화하여 포인트 군집의 한점이 가장 가까운 박스 표면을 측정하였다면, 포인트 군집의 한점과 표면의 거리는 정규분포하다고 가정할 수 있다.

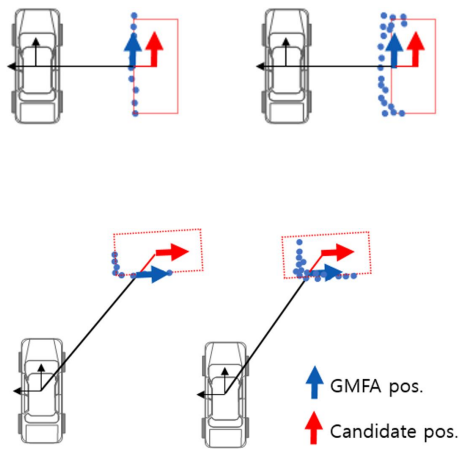



Fig. 3 Candidate box positions of point cluster shapes for the same situation

제시한 포인트 군집의 우도를 반영하기 위하여, 2.1에서 계산한 상대위치 기반 후보 박스특징으로부터 일정 범위 R 이내로 균등하게 분포하는 N개의 파티클을 생성한다. 이 때 범위 R은 경험에 의거하여 또다른 차량과 겹치지 않는 범위 내에서 설정하며, N은 수행시간과 비례하므로 목표 수행시간 이내가 되도록 설정한다. 생성된 파티클들은 각자 다른 박스특징의 위치를 나타내므로 식 (2)와 같이 각 파티클의 우도를 구할 수 있다.[4]

$$P(S^n \mid Box^m) = \prod_i^N p(x = d_i)$$
$$x \sim N(0, R) \qquad (2)$$
$N$ : points number of cluster n
$d_i$ : nearest distance to box about point i

따라서 최대 우도법에 의해 N개의 파티클 중 가장 큰 우도를 갖는 파티클의 위치를 해당 포인트 군집의 박스특징 위치로 정한다(Fig. 4).

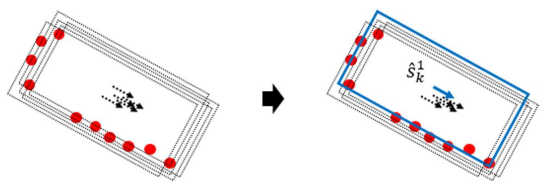



Fig. 4 Box feature selection based on maximum likelihood method

## 2.3. 박스특징 추정 후처리

상기한 과정을 통해 GMFA의 포인트 군집들로부터 각각의 박스특징을 추정하였다. 그러나 GMFA는 단순한 거리기반의 군집화 방법을 이용하기 때문에, Fig와 같이 한 차량에 대해서 2개 이상의 포인트 군집을 추적할 수 있다. 이는 서로 간의 박스형상이 겹치는 결과를 보일 수 있고, 자율주행을 위한 판단 및 제어 알고리즘의 비효율적 연산을 야기한다. 그러므로 이러한 상황을 제거하기 위한 방법으로 NMS(Non-maximum-suppression)을 사용한다.[5]

2.2과정에서 추정된 모든 포인트 군집들의 박스형상에 대하여 IOU(Intersection over union)가 일정 값 이상을

갖는 포인트 군집들을 가장 높은 우도를 갖는 포인트 군집으로 통합시키며, 이때 포인트 군집의 상태는 우도에 따라 가중 선형 조합하여 결정하였다.

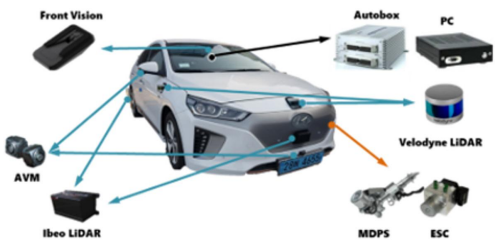
## 3. 차량 구성 및 실험 환경

### 3.1. 실험 차량 및 센서 구성

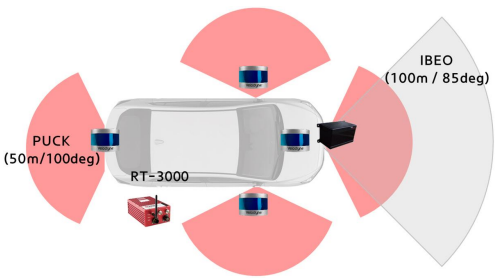
Fig. 5(a)는 박스특징 추정 알고리즘을 실험하기 위한 실험차량과 센서 구성을 보여준다. 자차량과 상대차량으로 현대 아이오닉 일렉트릭 2대가 사용되었다. 자차량은 Velodyne PUCK Hi-res 4개와 IBEO Lux 4L 1개가 Fig. 5(b)와 같이 장착 되어있다. GMFA알고리즘은 2D 포인트 클라우드 환경에서 동작하므로 Velodyne 센서의 포인트 클라우드를 지면에 투영하여 사용하였다. 자차량과 상대차량은 모두 고성능 GPS로 RT-3000이 장착되었으며, GPS동기화 장비인 RT-range가 장착되어 두 차량간 상대위치의 Ground truth를 측정할 수 있다.

### 3.2. 알고리즘 구현 및 실험 환경

제안한 박스특징 추정 알고리즘은 ROS(Robot Operating System)에서 Python 언어로 구현되었다.[6] ROS는 로봇




(a)




(b)

Fig. 5 (a) Test car sensor configuration (b) LiDAR field of view

구동을 위한 프로그램 인터페이싱 환경으로 각 알고리즘이 노드 단위로 병렬 작동하기 때문에, 본 논문에서 제안한 박스특징 추정과 GMFA 알고리즘을 병렬로 동작하게 하였다. 이는 LiDAR 측정주기 이내로 작동하여 자율주행차의 실시간 계산을 보장할 수 있게 한다.

실험은 서울대학교 미래모빌리티기술센터(SNU FMTC)에서 진행되었다. 실험 주행트랙은 도심환경 및 회전교차로 등이 설계되어 도심자율주행을 위한 알고리즘 실험에 적합하다.

## 4. 실험 결과

### 4.1. 정량적 분석

Fig. 6은 상대차량이 자차량을 추월하는 시나리오에서 GMFA, 상대위치기반, 최대우도법기반 방법 위치의 오차를 보여준다. 추월 시나리오는 자차량 위치를 기준으로 포인트 군집이 다양한 형상변화와 위치변화를 갖기 때문에, 제안한 알고리즘의 성능을 관찰 할 수 있다. 결과에서 포인트 군집을 추적하는 GMFA의 X방향 오차는 자 차량 후방에서 약 -2m 전방에서 약 2m로 확인되었는데, 이는 LiDAR센서가 상대차량의 가장 가까운 부분을 측정한 포인트 군집을 추적하기 때문이다. GMFA의 결과는 본문에서 제안한 알고리즘의 필요성을 보여준다고 할 수 있다. 반면에 2.1 상대위치(candidate)와 2.2의 최대우도법(proposed) 방법은 차량의 박스형상으로 가정하여 상대위치가 반영되어 X방향 오차는 0.5m 미만 유지하였다. 최대우도법에 기반해 추정하는 방법의 성능향상은 Y방향 오차에서 나타났다. 상대차량이 추월하며 자 차량 근처를 지날 때 (3~7s) 포인트 군집의 형상 변화가 크게 발생하므로, 최대우도법에 기반하여 추정된 위치가 오차를 감소하였음을 확인하였다.

Fig. 7은 자차량이 상대차량을 뒤따라가는 시나리오에서 GMFA, 상대위치기반, 최대우도법기반 방법 위치의 오차를 보여준다. 시나리오 내에서 전방의 상대차량이 회전하므로 포인트 군집으 형상변화가 크게 발생할 수 있다. 결과에서 X방향 오차는 앞선 결과와 같이 GMFA에 비해 상대위치(candidate), 최대우도법(proposed) 방법 모두 약 2미터 이상 오차가 감소한 것을 확인하였다. 상대차량이 우회전 하였을 때(35~45s), GMFA는 측정된 포인트 군집 자체를 추적하므로 Y방향오차가 크게 증가하였다. 하지만 상대위치를 반영한 결과에서 오차가 감소하였고, 최대우도법에 기반하여 포인트 군집의 형상을 반영한 결

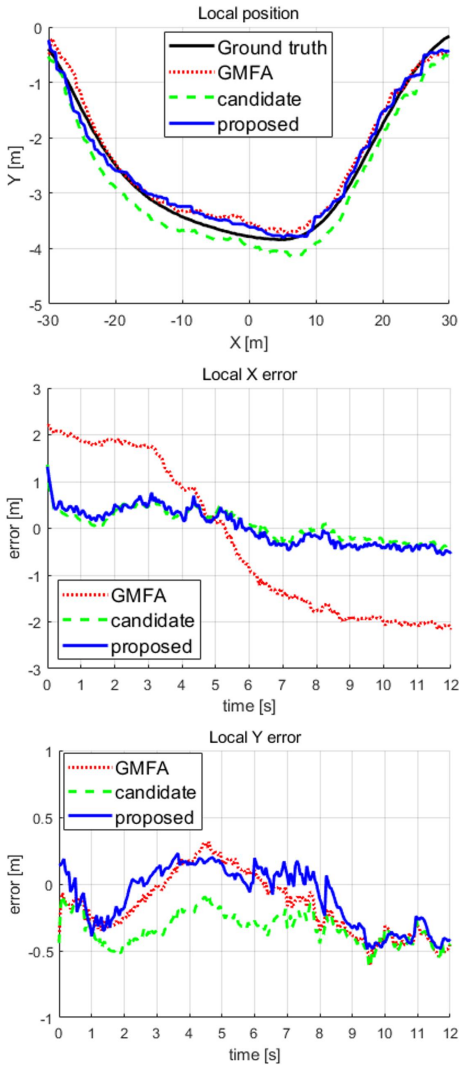



Fig. 6 Test results for scenarios overtaken by target vehicle

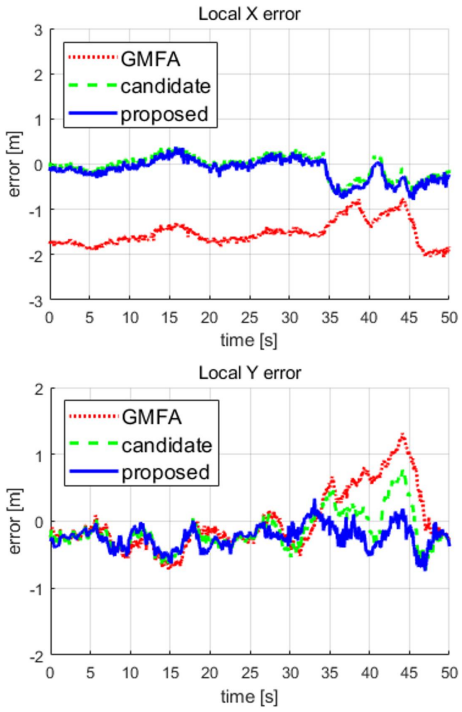
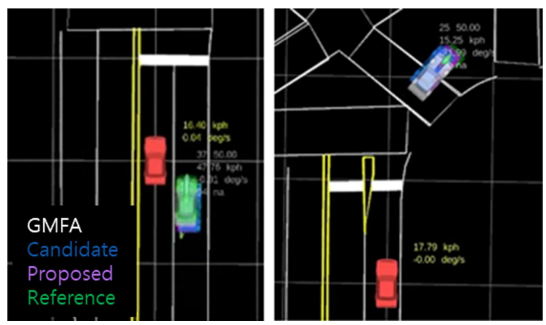



Fig. 7 Test results for scenarios in which the host vehicle follows the target vehicle




(a) (b)

Fig. 8 Visualization of test results. (a) overtake scenario (b) follow scenario (GMFA-white, Candidate-blue, Proposed-purple, Ground Truth-green)

과 Y방향 평균제곱오차는 약 26% 감소하는 것을 확인하였다.

### 4.2. 정성적 분석

Fig. 8은 4.1의 실험시나리오에서 제안한 알고리즘과 비교를 위한 알고리즘의 결과데이터를 지도에 시각화하여 나타낸 그림이다. (a)에서 상대차량이 자차량을 추월하기 위해 옆 차선을 지날 때 포인트 군집의 형상을 제안한 알고리즘(Proposed)이 가장 정확하게 반영한 것을 확인할 수 있다. 2.1에서 정의한 상대위치기반 가정이 근접한 차량에 대해선 정확하지 않다는 것을 보여주며, 여러 파티클의 우도를 통해 최적위치를 추정하는 과정이 효과적으로 작동했음을 보여준다. (b)는 자차량이 우회전하는 상대차량을 따라갈 때 포인트 군집의 형상변화를 제안한 추정 알고리즘이 가장 정확하게 반영한 것을 보여준다. GMFA 알고리즘의 특성 상 포인트 군집의 형상변화 반영

이 늦기 때문에, 단순하게 군집의 평균위치만을 반영하는 상대위치 기반 방법보다 최대우도를 갖는 추정위치가 더 좋은 결과를 나타낼 수 있었다.

## 5. 결 론

본 논문에서는 최대우도법을 이용한 포인트 군집의 박스특징 추정 알고리즘을 제안하였다. 자율주행차량에서 포인트 군집 추적 알고리즘에 의해 추적되고 있는 상대차량의 실제 위치를 정확하게 추정하기 위해 연구되었다. 제안된 알고리즘은 포인트 군집을 지정된 크기와 추적되고 있는 요 각을 가진 박스특징으로 가정하고 거리에 따른 두가지 단계로 구분하여 수행하였다. 첫번째로 박스특징 위치와 자차량 사이의 가상의 선과 박스형상이 교차하는 점을 포인트 군집의 평균위치로 가정하여 후보위치를 선정하였다. 두번째로 후보위치 주변에 균등 분포하는 박스특징 파티클을 생성하고 각 파티클에 따른 포인트 군집의 우도를 반영하여 가장 큰 우도를 갖는 파티클을 선택함으로써 포인트 군집의 형상과 박스 가정의 오차를 반영하였다.

자율주행 시험트랙에서 실제 차량을 이용하여 실험한 결과, 상대차량의 자 차량 추월 시나리오에서 X방향 평균위치오차는 약 68%, Y방향 평균위치오차는 약 21% 감소하였다. 상대차량을 추종하는 시나리오에서 X방향 평균위치오차는 약 55%, Y방향 평균위치오차는 26% 감소하였다. 따라서 차량을 박스특징으로 가정하고 포인트군집의 우도를 계산하여 추정하는 본 알고리즘이 특정상황에서 높은 효과를 보임을 확인하였다.

후속연구에서는 포인트 군집의 형상을 이용한 승용차량과 상용차량 및 VRU의 구분을 통해 각 분류에 따라 더욱 정확한 위치 및 동적상태추정이 필요할 것으로 생각된다.

## 후 기

본 논문은 산업통상자원부 산업기술혁신사업(10079730, 자동차전용도로/도심로 자율주행 시스템 개발 및 성능평가)의 지원을 받아 수행하였습니다.

## 참고문헌

(1) Najibi, M., Lai, G., Kundu, A., Lu, Z., Rathod, V., Funkhouser, T., Pantofaru, C., Ross, D., Davis, L. and Fathi, A., 2020, “Dops: Learning to detect 3d objects and predict An LSTM Approach to Temporal 3D Object Detection in Point Clouds”, CVPR.

(2) H. Lee, J. Yoon and K. Yi, 2020, “Moving Object Detection and Tracking Based on Interaction of Static Obstacle Map and Geometric Model-Free Approach for Urban Autonomous Driving”, IEEE Transactions on Intelligent Transportation Systems.

(3) 이호준, 채흥석, 서호태, 이경수, 2018, “라이다를 이용한 이동 차량 검출을 위한 바운딩 박스와 가상광선에 의한 주변 차량 형상 추출”, 한국자동차안전학회 춘계 학술대회 논문집.

(4) Roy L. Streit and Tod E. Luginbuhl, 1994, “Maximum likelihood method for probabilistic multihypothesis tracking”, Proc. SPIE 2235.

(5) Neubeck, Alexander and Luc Van Gool, 2006, “Efficient non-maximum suppression”, 18th International Conference on Pattern Recognition (ICPR'06), Vol. 3, IEEE.

(6) Quigley, Morgan, et al., 2009, “ROS: an open-source Robot Operating System”, ICRA workshop on open source software. Vol. 3, No. 3.2.