

자율주행을 위한 라이다 기반 객체 인식 및 분류

변예림* · 박만복**

Lidar Based Object Recognition and Classification

Yerim Byeon*, Manbok Park**

Key Words: Lidar(라이다), Clustering(군집화), Classification(분류), Object(객체), Feature Point(특징점)

ABSTRACT

Recently, self-driving research has been actively studied in various institutions. Accurate recognition is important because information about surrounding objects is needed for safe autonomous driving. This study mainly deals with the signal processing of LiDAR among sensors for object recognition. LiDAR is a sensor that is widely used for high recognition accuracy. First, we clustered and tracked objects by predicting relative position and speed of objects. The characteristic points of all objects were extracted using point cloud data of each objects through proposed algorithm. The Classification between vehicle and pedestrians is estimated using number of characteristic points and distances among characteristic points. The algorithm for classifying cars and pedestrians was implemented and verified using test vehicle equipped with LiDAR sensors. The accuracy of proposed object classification algorithm was about 97%. The classification accuracy was improved by about 13.5% compared with deep learning based algorithm.

1. 서론

자율주행 자동차는 운전자의 개입 없이 목적지까지 스스로 주행하는 차량으로 현재 운전자 지원 시스템이라는 이름으로 양산되고 있다. 고속도로 지원 시스템은 SAE (미국자동차공학회) 기준으로 약 2단계 수준의 자율 주행 기술이 양산되고 있는 실정으로 많은 완성차, 부품 회사 뿐만 아니라 연구소 및 대학 등에서 활발히 연구되고 있다⁽¹⁾. 자율주행은 Radar, Lidar, GPS, 카메라와 같은 환경센서로 주위의 환경은 인식, 판단하고 차량을 제어하여 주행하는 인식-판단-제어 3단계로 이루어진다. 이 중 인식은 자율주행에 있어 매우 중요한 단계이며 현재 자율주행 차량에는 주로 라이다, 카메라, 레이더와 같은

환경센서를 장착하여 주변의 사람이나 자동차 등 필요한 정보를 취득하는데 활용하고 있다⁽²⁾. 환경센서 중 라이다는 빛의 반사를 이용하여 객체까지의 거리를 측정하는 장비로 인식 정확도가 높아 인식을 위한 센서로 주로 사용되고 있는 센서이다. 그러나 라이다 센서는 차량과 보행자 등의 필요한 객체 정보들을 검출함과 동시에 불필요한 정보들도 획득하게 되므로 그대로 사용할 수 없다. 따라서, 라이다의 raw data를 처리할 필요가 있다. 본 논문에서는 라이다의 raw data를 군집화하고, 상대 위치와 속도를 예측하여 추적하는 과정을 기술 하였다. 또한, 판단과 제어 단계에서 객체의 특징점과 객체의 종류가 필요한 상황이 있다⁽⁵⁾. 본 논문에서는 1차 군집화 결과를 이용하여 특징점을 구하는 알고리즘을 기술하였으며, 특징점 형상의 차이를 이용하여 차량과 보행자를 구분하는 알고리즘 개발하여 검증하였다.

최근 연구 중 객체를 분류하는 방법으로 Lidar 포인트 클라우드에 의해 제공되는 3D데이터를 이용하여 객체의

* 한국교통대학교 전자공학과, 석사

** 한국교통대학교 전자공학과, 교수

E-mail: anjfqha3@naver.com

형상을 Clustering 하는 클러스터링 접근법으로 모든 객체의 형상 집합을 만든 후 입력 데이터와 비교될 객체 모델을 만들고, 입력 데이터와 객체 모델을 비교하여 객체의 클래스를 결정하여 객체를 분류하는 방법이 있다. 이것은 딥러닝을 이용한 방법으로 차나 다른물체 2가지의 클래스로 나누어 약 92%의 정확도를 가진다⁽³⁾. 다른 연구에서는 사전에 학습된 Squeezeseg V2를 이용해 차량을 인식하고 분류 후 분류된 차량의 포인트 클라우드에 포함된 이상치에 강인하게 대응하고 더욱 정확한 차량 분류를 위해, Euclidean Clustering을 이용해 후처리 작업을 진행하여 차량을 인식,분류하였다. 이 방법 또한 딥러닝을 이용한 것으로 에러율은 약 24.4%이며, 정확도는 75.6%의 차량 분류 결과를 보여준다⁽⁴⁾.

본 논문에서는 딥러닝을 이용한 방법^(3,4)이 아닌 규칙 기반으로 라이다의 신호를 처리하여 객체를 인식하고, 분류하는 과정을 기술하였다. 객체 인식 및 분류 알고리즘의 결과를 이용하여 차선 변경이나 차량 제어에 활용할 수 있다.

Fig. 1은 본 연구에서 사용한 자율주행 플랫폼으로 차량에는 16채널 Lidar 3개와 32채널 Lidar 1개, 카메라 2대, 레이더 총 5대가 장착되어있다. 이 중 객체 인식 및 분류를 위해 Lidar 3개를 이용하였다.

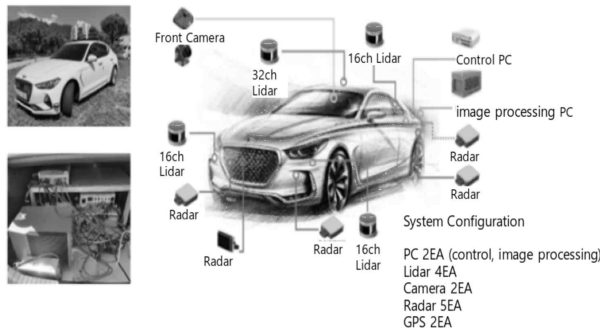


Fig. 1 System Configuration

2. 객체 인식

라이다 센서는 빛을 송출하고 반사되어 오는 빛의 정보를 이용하여 사물의 존재 여부에 대한 결과를 알려준다. 이 취득된 정보를 PCD(Point Cloud Data)라고 부르며, PCD는 좌표(X,Y,Z)로 표기할 수 있다. 2절에서는 라이다의 PCD를 이용하여 객체를 인식하는 과정을 나타내었다. 과정은 크게 군집화-특징점 추출-트래킹으로 총 3단계로 구성하였다.

2.1. 군집화

2.1.1. 1차 군집화

- 1) 라이다 센서는 초당 최대 600,000개의 방대한 데이터를 취득한다. 데이터를 효율적으로 처리하기 위해 같은 객체에 대한 PCD를 하나로 합치는 군집화 과정이 필요하다. 서로 다른 객체에 대한 PCD간의 거리 차이가 같은 객체에 대한 PCD간의 거리 차보다 크다는 점을 이용하여 같은 객체에 대한 PCD를 군집화 하였다. 1차 군집화는 같은 Layer의 PCD를 군집화 하는 과정이며 2차 군집화는 1차 군집화의 결과를 이용하여 서로 다른 Layer 간 군집화 과정이다.
- 2) 1차 군집화는 Fig. 2와 같이 같은 Layer의 인접한 데이터 간의 거리 차이가 일정 임계값보다 작을 경우 같은 객체로 군집화하고, 클 경우 다른 객체로 분류한다. 인접한 데이터 간의 거리 차이를 구하는 식은 식 (1)과 같다.

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (1)$$

1차 군집화의 임계값은 본 연구에서 사용한 Lidar 센서의 수평 resolution인 0.2°와 원점과 PCD간의 거리 distance에 따라 식 (2)와 같이 설정하였다.

$$d_{thr} = distance * \sin(2^\circ) + \alpha \quad (2)$$

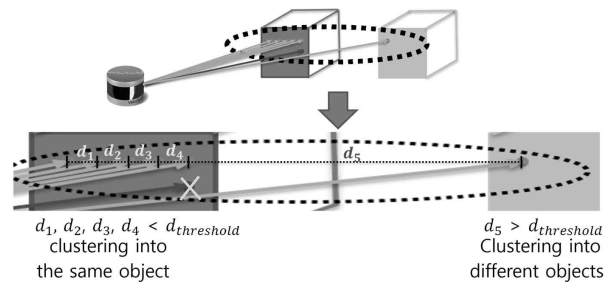


Fig. 2 Primary Clustering

2.1.2. 2차 군집화

- 1) 2차 군집화는 1차 군집화의 결과를 이용하여 객체를 분류한다. 2차 군집화는 Fig. 3과 같이 서로 다른 layer간의 데이터를 군집화하는 과정이다. 1차 군집화의 결과 중 하나의 좌표와 나머지 좌표와의 거리가 일정 임계 값보다 작은 경우 모두 같은 ID를 부여하고, ID가 부여되지 않은 1차 군집화 끼리 위와 같은

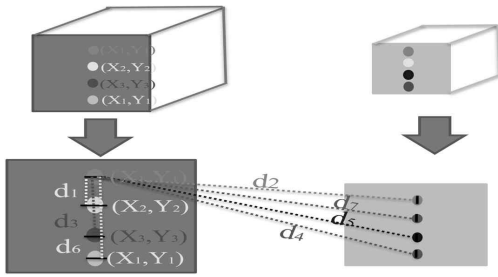


Fig. 3 Secondart Clustering

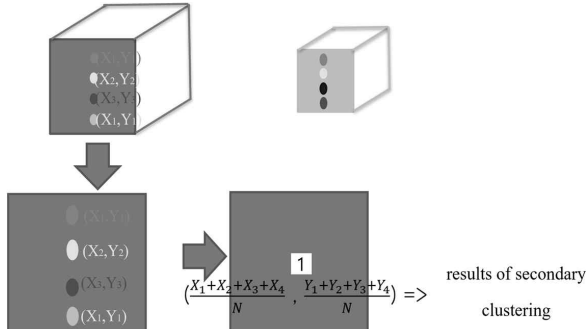


Fig. 4 Secondart Clustering Result

ID를 부여하는 과정을 반복한다.

- 2) Fig. 4는 2차 군집화 결과를 나타낸다. 모든 객체에 대해 그림과 같이 ID를 부여하고, 2.3의 트래킹 과정을 거친 후 최종적으로 객체에 대해 고유 ID를 부여한다.

2.2. 특징점 추출(Feature Point Extraction)

2.2.1. 개요

차선 변경과 같이 자율주행에 있어 객체에 대한 특징

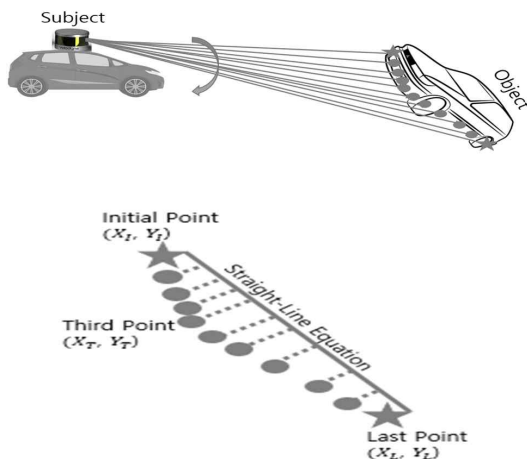


Fig. 5 Feature Point

점이 필요한 경우가 있다. 본 연구에서는 차선 변경 뿐만 아니라 해당 객체가 사람인지 차량인지 분류하는 과정에서 특징점을 이용하였다. 특징점을 추출하기 위해 라이다가 Fig. 5와 같이 시계 방향으로 객체를 스캔하는 특성을 이용하였다. Fig. 5에서의 Object에 해당하는 점 중 왼쪽 상단의 별은 시작 점을 나타내고, 오른쪽 하단의 별은 끝 점을 나타낸다.

2.2.2. 알고리즘

- 1) 1차 군집화 결과를 저장할 때 거치는 과정으로 하나의 객체에 대한 1차 군집화 결과 중 첫 번째 인덱스의 PCD와 마지막 인덱스의 PCD를 저장한 후 2개의 PCD를 이용하여 또 다른 특징점을 구한다.
- 2) Fig. 5와 같이 첫 번째 인덱스의 PCD와 마지막 인덱스의 PCD를 잇는 직선의 방정식을 구한 후, 하나의 객체에 대한 데이터 중 2개의 PCD를 제외한 모든 PCD와 구한 직선과 수직 거리를 계산하여 가장 먼 거리의 PCD(X_T, Y_T)를 저장한다.
- 3) 가장 먼 거리의 PCD를 저장할 때, 거리가 일정 거리보다 작을 경우 저장하지 않는다.

2.2.3. 결과

- 1) Fig. 6은 차량에 대한 특징점으로 첫 번째 인덱스의 점과 마지막 인덱스의 점, 알고리즘을 적용하여 구한 특징점 총 3개가 존재한다.
- 2) Fig. 7은 사람에 대한 특징점으로, 알고리즘을 적용한 결과 첫 번째 인덱스의 점, 마지막 인덱스의 점 총 2개의 특징점만 존재한다.

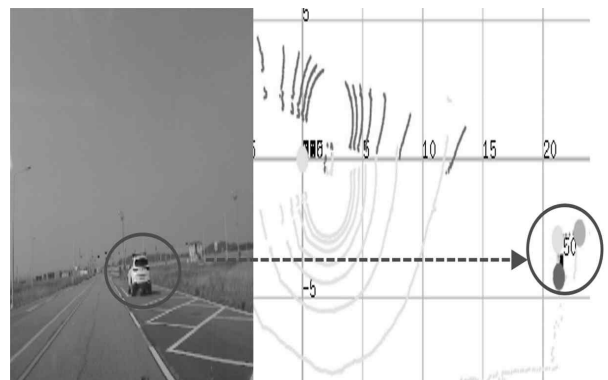


Fig. 6 Feature Point Result (Car)

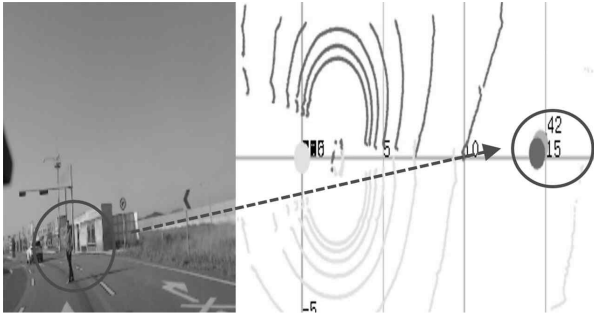


Fig. 7 Feature Point Result (People)

$$F = \begin{bmatrix} 10T0 \\ 010T \\ 0010 \\ 0001 \end{bmatrix} \quad (10)$$

$$H = \begin{bmatrix} 1000 \\ 0100 \end{bmatrix} \quad (11)$$

$$x = [x \ y \ \dot{x} \ \dot{y}] \quad (12)$$

$$z = [x \ y]^T \quad (13)$$

2.3. Prediction

라이다로 인식한 객체의 위치를 이용하여 다음 프레임에서의 위치, 속도를 예측하는 과정으로, 2.4절 object tracking 과정에서 객체의 좌표 (X,Y)에 칼만필터를 적용한다. 사용한 칼만필터 모델⁽⁵⁾은 Constant Velocity model이다.

2.3.1. 알고리즘

State Prediction 과정은 식 (3), 식 (4)와 같다.

$$\textcircled{1} \hat{x}(k+1|k) = F(k)\hat{x}(k|k) \quad (3)$$

$$\textcircled{2} \hat{P}(k+1|k) = F(k)\hat{P}(k|k)F^T + Q(k) \quad (4)$$

Measurement Prediction 과정은 식 (5), 식 (6)과 같다.

$$\textcircled{1} \hat{z}(k) = H(k)\hat{x}(k+1|k) \quad (5)$$

$$\textcircled{2} \hat{S}(k) = H(k)\hat{P}(k+1|k)H^T + R(k) \quad (6)$$

Measurement Update와 System Update는 식 (7), 식 (8), 식 (9)와 같이 나타낸다.

$$\textcircled{1} K(k) = \hat{P}(k+1|k)H^T\hat{S}(k)^{-1} \quad (7)$$

$$\textcircled{2} \hat{x}(k+1|k+1) = \hat{x}(k+1|k) + K(k)V(k) \quad (8)$$

$$\textcircled{3} \hat{P}(k+1|k+1) = (I - K(k)H(k))\hat{P}(k+1|k) \quad (9)$$

위의 식에서 F Matrix는 식 (10), H Matrix는 식 (11), State vector x는 식 (12), Measurement Vector는 식 (13)을 이용하였다.

2.4. Object Tracking

2.4.1. 알고리즘

- 1) 2차 근접화 결과를 이용하는 과정으로, 하나의 객체에 대해 고유ID를 부여했을 경우 다음 프레임에서도 ID를 유지하기 위한 과정이다. Fig. 8은 Tracking의 개념이다.

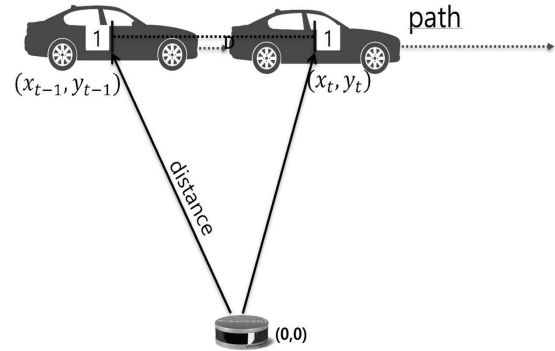


Fig. 8 Object Tracking

- 2) 트래킹 과정은 아래와 같이 이루어진다.
 - ① 객체와 라이다 원점 사이의 거리에 따라 threshold를 설정한다.
 - ② 식 (14)와 같이 과거의 좌표와 현재의 좌표 사이의 거리를 구한다.

$$d = \sqrt{(x_t - x_{t-1})^2 + (y_t - y_{t-1})^2} \quad (14)$$

- ③ threshold와 d를 비교하여 $d < \text{threshold}$ 인 경우 카운트를 증가시키고, $d > \text{threshold}$ 인 경우 카운트를 감소시킨다.
- ④ 카운트가 3보다 큰 경우 객체를 생성, 유지하고 0 이하가 될 경우 객체를 소멸시킨다.

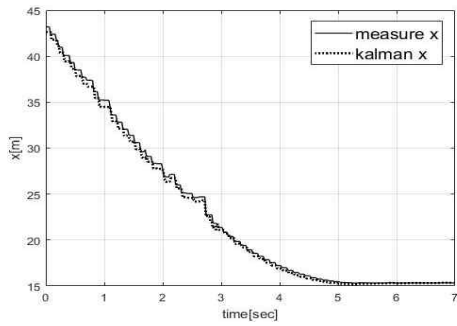


Fig. 9 Object Detection Result x [m]

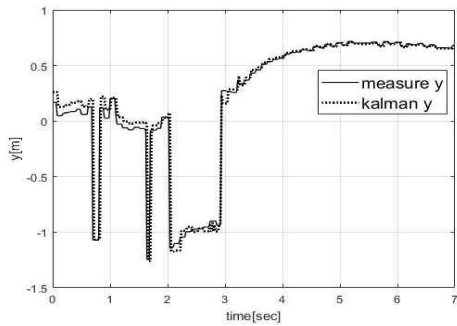


Fig. 10 Object Detection Result y [m]

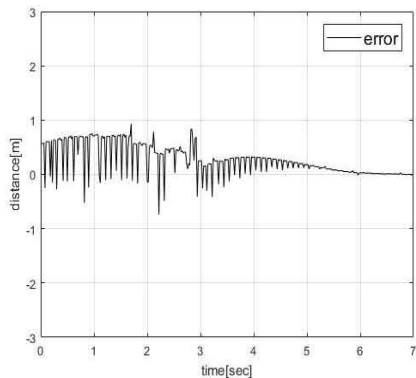


Fig. 11 Error [m]

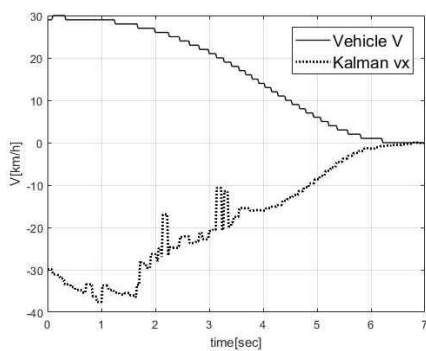


Fig. 12 Kalman Filter Result [km/h]

2.4.2. 결과

- 1) Fig. 9는 전방의 객체를 추적한 결과로 x축은 시간[sec] y축은 x[m]를 나타내며, Fig. 10은 x축은 시간[sec] y축은 y[m]를 나타낸 결과이다. Fig. 11은 측정 값과 예측 값의 오차에 대한 결과이다. 오차는 약 0.2412 m 이다.
- 2) Fig. 12는 전방의 객체에 다가갔을 경우에 대한 칼만 필터로 예측한 속도와 실험 차량의 속도를 나타낸 결과이다.

3. 객체 분류

인식한 해당 객체가 차량인지, 보행자인지 그 외의 것인지 분류하는 과정이 필요하다. 3장에서는 라이다의 raw data를 처리하여 얻은 객체가 자동차인지 사람인지 분류하는 과정을 기술하였다.

본 연구에서는 자동차에 대한 특징점과 보행자에 대한 특징점간의 차이를 이용하여 객체를 분류하였다.

3.1. 분류

3.1.1. 개념

자율주행차량 관점으로 봤을 때 자동차에 대한 특징점은 Fig. 13의 왼쪽 그림과 같이 2개나, 3개로 얻을 수 있다. 반면, 사람은 넓이가 좁아 특징점 2개가 좁게 형성된다. 사람과 자동차의 특징점 형성의 차이를 이용하여 두 객체를 분류하였다. Fig. 14는 자동차의 측면에 대한 특징점 결과로, 두 개의 특징점을 이용한 2.2.2 알고리즘을 적용하여 새로운 특징점을 저장한 결과이다. 반면, Fig. 15는 두 개의 특징점 간 거리차가 좁아 3번째 특징점이 형성되지 않았다.

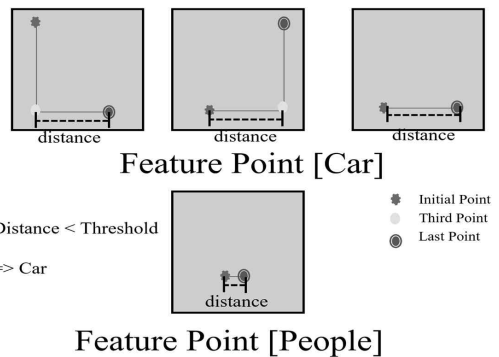


Fig. 13 Object Classification Method

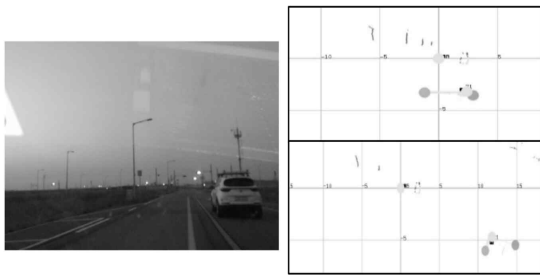


Fig. 14 features on the side of a car

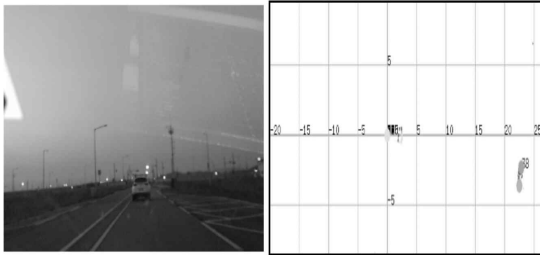


Fig. 15 features on the front of a car

을 적용한 결과이다. 해당 객체 (ID=8) 위에 숫자 1이 부여된 것을 확인할 수 있다.
 2) Fig. 17은 차량에 대한 특징점 3개에 3.1.2절의 알고리즘을 적용한 결과이다. 해당 객체 (ID=28) 위에 숫

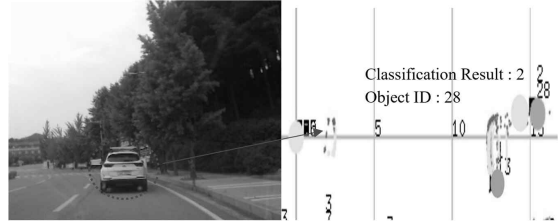


Fig. 17 Classification Result (Car)

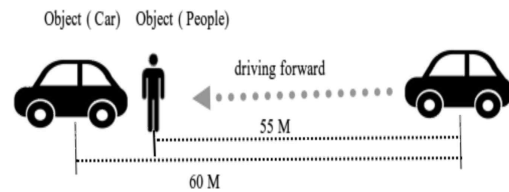


Fig. 18 Scenario

3.1.2. 알고리즘

- ① 객체에 대한 특징점의 개수가 2개이고, 2개의 좌표 간의 거리가 일정 값 보다 작을 경우 사람으로 판단하여 1을 부여하였다.
- ② 객체에 대한 특징점의 개수가 2개이고, 2개의 좌표 간의 거리가 일정 값보다 클 경우 차량으로 판단하여 2를 부여하였다.
- ③ 객체에 대한 특징점의 개수가 3개이고, 3개 중 자율주행 차량과 가까운 2 사이의 거리를 계산하고, 일정 값일 경우 차량으로 판단하여 2를 부여하였다.
- ④ 그 외의 것은 3을 부여하였다.

3.1.3. 시나리오 및 검증 결과

1) Fig. 16은 사람에 대한 특징점에 3.1.2절의 알고리즘

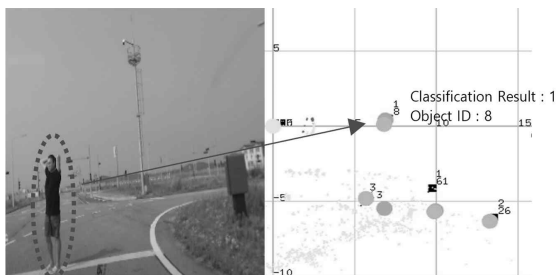
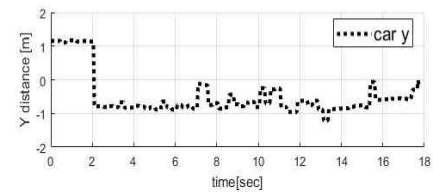
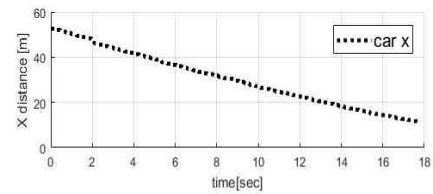
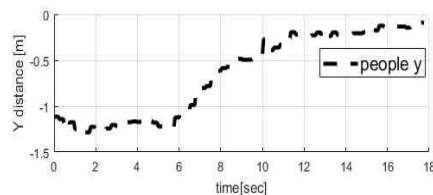
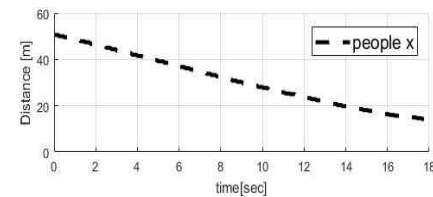


Fig. 16 Classification Result (People)



(a) Scenario - Position [Car]



(b) Scenario - Position [People]

Fig. 19

자 2이 부여된 것을 확인할 수 있다.

3) 알고리즘 검증을 위한 시나리오는 Fig. 18과 같이 전방의 객체(차량, 사람)에 다가가는 시나리오로 구성하였다.

Fig. 19는 Fig. 18 시나리오에서의 객체 위치를 그래프로 그린 것이다.

Fig. 20에서의 delta distance [car feature point] 점

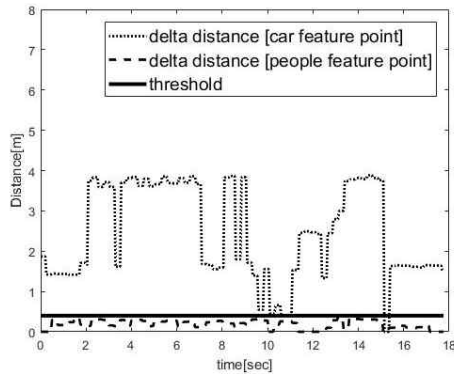


Fig. 20 Delta Distance - Feature Points

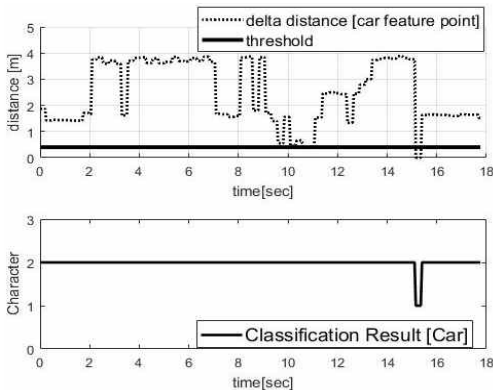


Fig. 21 Classification Result [Car]

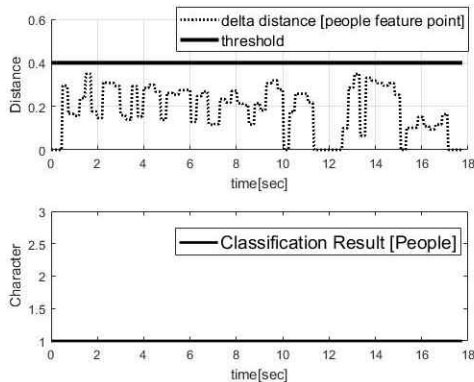


Fig. 22 Classification Result [People]

선은 자동차의 특징점 간의 거리 차이이며, delta distance [people feature point] 점선은 사람의 특징점간의 거리 차이이다.

사람에 대한 특징점 사이의 거리를 분석한 결과 약 0.4m 이하인 것을 확인하였다. 따라서, 자동차와 사람을 나누는 임계 값을 threshold 실선 0.4m로 설정하였다.

Fig. 21과 Fig. 22는 Fig. 20에 3.2절의 알고리즘을 적용한 결과이다. Fig. 21은 자동차에 대한 결과로, 97%의 정확도로 속성 2를 부여했다. Fig. 22는 사람을 인식한 결과로 100%의 정확도로 속성 1를 부여했다.

4. 결론

본 연구의 결과를 요약하면 다음과 같다.

- 1) 라이다의 Raw data를 군집화, 트래킹하여 객체를 추적하였다.
- 2) 군집화 과정에서 객체의 특징점 2개를 저장하였고, 3번째 특징점을 구하는 알고리즘을 개발하였다. 구한 특징점을 이용해 차량과 보행자를 구분하는 알고리즘을 개발 및 검증하였다.
- 3) 객체 분류 알고리즘을 적용했을 경우 약 97%의 정확도를 가진다.
- 4) 본 연구에서 차량이나 보행자를 구분하는 알고리즘을 개발, 검증하였다. 객체의 각도가 급격하게 변할 경우 정확도가 떨어지는 경향이 있다. 향후에는 가변 임계값을 적용한 알고리즘을 개발할 것이다.

후기

본 연구는 국토교통부 도심도로 자율협력주행 안전·인프라 연구 사업의 연구비지원(과제번호: 20PQOW-B152618-02)에 의해 수행되었습니다. 또한 이 논문은 2020년도 정부(경찰청)의 재원으로 도로교통공단의 지원을 받아 수행된 연구임(No.POLICE-L-00001-02-101, 자율주행차의 도로주행을 위한 운행체계 및 교통인프라 연구개발) 또한 이 연구는 2019년도 산업통상자원부 및 산업기술평가관리원(KEIT) 연구비 지원에 의한 연구임(과제번호: K_G012000307003, 과제명: NCAP 대응을 위한 후방 자동제동시스템Rear Automatic Braking System 개발)

참고문헌

- (1) D. Shin, B. Kim, K. Yi, A. Carvalho and F. Borrelli, 2019, "Human-Centered Risk Assessment of an Automated Vehicle Using Vehicular Wireless Communication", IEEE Transactions on Intelligent Transportation Systems, Vol. 20, No. 2, pp. 667~681.
- (2) 이호준, 채홍석, 서호태, 이경수, 2018, "자율주행을 위한 레이더 기반 인지 알고리즘의 정량적 분석," 자동차안전학회지, Vol. 10, No. 2, pp. 29~35.
- (3) Michael Kusenbach, Michael Himmelsbach and Hans-joachim Wuensche, 2016, "A New Geometric 3D LiDAR Feature for Model Creation and Classification of Moving Objects", 2016 IEEE Intelligent Vehicles Symposium (IV), pp. 272~278.
- (4) 김수연, 권순용, 김명준, 김정하, 2019, "LiDAR 기반의 딥러닝을 이용한 차량 분류 알고리즘", 한국자동차공학회 추계 학술대회, pp. 791~794.
- (5) Kai Arras, Cyrill Stachniss, Maren Bennewitz, Wolfram Burgard, 2012, "Robotics 2 Target Tracking", Uni Freiburg.