

카메라-라이다 센서 융합을 통한 VRU 분류 및 추적 알고리즘 개발

김유진* · 이호준** · 이경수***,†

Vision and Lidar Sensor Fusion for VRU Classification and Tracking in the Urban Environment

Yujin Kim*, Hojun Lee**, Kyongsu Yi***,†

Key Words: Autonomous driving(자율 주행), Sensor fusion(센서 융합), You Only Look Once:YOLO(올로), Vulnerable road user(교통 약자), Geometric Model-Free Tracking

ABSTRACT

This paper presents an vulnerable road user (VRU) classification and tracking algorithm using vision and LiDAR sensor fusion method for urban autonomous driving. The classification and tracking for vulnerable road users such as pedestrian, bicycle, and motorcycle are essential for autonomous driving in complex urban environments. In this paper, a real-time object image detection algorithm called Yolo and object tracking algorithm from LiDAR point cloud are fused in the high level. The proposed algorithm consists of four parts. First, the object bounding boxes on the pixel coordinate, which is obtained from YOLO, are transformed into the local coordinate of subject vehicle using the homography matrix. Second, a LiDAR point cloud is clustered based on Euclidean distance and the clusters are associated using GNN. In addition, the states of clusters including position, heading angle, velocity and acceleration information are estimated using geometric model free approach (GMFA) in real-time. Finally, the each LiDAR track is matched with a vision track using angle information of transformed vision track and assigned a classification id. The proposed fusion algorithm is evaluated via real vehicle test in the urban environment.

1. 서론

도심과 같이 타 차량뿐만 아니라 다양한 취약 도로 사용자(Vulnerable Road User, VRU)가 많고 복잡한 환경에 대응하는 자율 주행 시스템 구현을 위해 여러 방면의 연구가 활발히 진행되고 있다. 도심 자율주행 기술이 실제로

상용화될 때, 가장 예민한 사항이 VRU와의 사고 이슈이다. 현재까지 자율주행 중 VRU와의 사고 상황에서 많은 경우에 근본적인 원인이 인지 착오로 나타났다. 이에 하나의 예시가 되는 사건으로 지난 2018년 미국 애리조나주에서 발생한 Uber 사의 자율주행 시험 운행 차량의 보행자 사망사고가 있었다.⁽¹⁾ 밤 늦은 시각 어두운 도로에 자전거를 끌고 차도를 무단 횡단하고 있던 보행자를 들이받아 일어난 사고로 자율주행 차량이 해당 보행자를 인지하였으나, 보행자가 아닌 도로 위 무시 가능한 작은 물체로 판단하고 False Positive 상황으로 분류하여 정차하지 않은 것으로 추정되었다. 이와 같은 인지 관련 이슈는 현재 상

* 서울대학교 차량 동역학 및 제어 연구실, 박사과정
 ** 서울대학교 차량 동역학 및 제어 연구실, 박사
 *** 서울대학교 차량 동역학 및 제어 연구실, 교수
 †교신저자: kyi@snu.ac.kr
 E-mail: zxc123xc@snu.ac.kr

용화된 HDA시스템에서도 나타난 바가 있다. 지난 2020년 6월경에, 대만에서 HDA 모드로 주행하던 테슬라 모델 3 차량이 넘어진 흰색 화물차량을 들이 받은 사건이 있었다. 해당 사건에 대한 원인으로서는 햇빛이 강하게 내리쬐는 상황에서 차량에 부착된 카메라 시야에 상당 부분을 차지하는 대형 화물차량의 흰색 윗면이 빛을 반사하여 카메라 센서가 물체를 인식하고 분류하는데 오작동이 일어났을 가능성이 유추되고 있다. 이처럼 차 차량이 각 종 주행 상황에서 정확한 거동 계획을 위해서는 해당 물체의 위치, 속도, 거동 방향 등의 상태를 정밀한 추종해야 하며, 또한 물체의 종류를 정확하게 분류하여야 한다.

이를 위해 다양한 센서를 융합하여 성능을 확보하고자 하는 여러 연구가 진행되어 왔다. Koyel Banerjee⁽¹⁾는 라이다 포인트 클라우드 데이터를 RGB 이미지에 투영하고, 그것을 엡샘플링하여 특징을 도출해낸 융합된 데이터를 인풋으로 Faster R-CNN 방식을 적용했다. 또한, 이와 같은 하위 레벨 퓨전을 통하여 위치정보를 가진 분리된 물체를 자동차와 보행자 두 클래스로 분류하여 물체를 인식하였다. Hyunggi Cho⁽²⁾는 라이다와 레이더 센서를 융합하여 물체의 상태를 추정하고 도출된 물체의 중앙 위치 값을 카메라 이미지에 투영하여 카메라 센서를 통해 분류된 이미지와 융합하였다. Hongbo Gao⁽³⁾는 카메라 RGB이미지와 라이다 데이터를 통해 도출한 Depth 이미지 쌍을 인풋으로 CNN 러닝 기법을 통해 물체를 보행자, 자전거, 트럭 그리고 승용차로 분류하는 연구를 진행하였다. Chavez-Garcia⁽⁴⁾는 라이다와 레이더를 융합하여 물체의 상태를 인지하고, 추가적으로 라이다 센서를 이용해 그리드 방식으로 정지 물체와 동적 물체를 구별하였다. 이는 주변 상황에 대한 2차원 점운 맵을 만들어 해당 맵을 바탕으로 오 인지를 제거하는 데 사용하였다. 또한 라이다와 카메라를 각각에 분류 정확도 분포를 정의하여 두 센서의 분류 정보를 융합하였다.

자율주행을 위해서는 3차원 공간에서의 물체 위치 정보가 포함된 인지 결과가 필요하다. 위의 연구들은 이에 물체의 3차원 공간에서 분류하기 위해 라이다 포인트 클라우드 정보를 인풋으로 인지 알고리즘을 구성했다. 하지만 라이다 정보의 불규칙성과 3차원 공간에서의 포인트 밀도의 희박성으로 인해 높은 인지 성능을 확보하는 데 어려움이 있고, 로드도 굉장히 커 실시간성을 확보하기 어렵다. 반면, 이미지 기반의 물체 인지는 많은 개발이 이미 이루어졌으며, 높은 성능 및 속도를 보이는 YOLO와 같은 알고리즘들이 오픈 소스로 다방면의 용도로 널리 사용되고 있다. 따라서, 해당 논문에서는 3차원 좌표계에서

의 라이다 기반으로 추정된 물체 위치 및 헤딩, 속도, 가속도 트랙 정보에 이미지 기반 물체 클래스 정보를 활용할 수 있도록 융합하였다. 물체의 라이다 트랙은 GMFA(Geometric Model-Free Approach)를 적용하여 도출한다.⁽⁸⁾ 적용 절차로는 먼저 라이다 포인트 클라우드를 유클리디안 거리를 이용하여 클러스터링하고, GNN(Global Nearest Neighbor)을 이용하여 각 스텝의 클러스터링된 물체를 연관지어 인식한다. 또한 각 클러스터의 상태 정보들을 확장 칼만 필터(EKF)와 ICP(Iterative Closest Point)방식을 적용하여 추종한다. 이에 위치 및 속도 등의 상태를 추종하는 데 불확실성이 크지만, 물체를 분류에 유리한 카메라 트랙을 융합한다. 카메라 이미지를 통한 픽셀 좌표 상의 물체 분류는 실시간 성과 정확성이 입증된 오픈 소스 YOLO(You Only Look Once)⁽⁹⁾ 알고리즘을 적용하여 도출하며, 이는 호모그래피 행렬을 이용해 차량 로컬 좌표계로 변환되며, 다소 정확한 카메라 트랙의 각도 정보를 활용하여 라이다 트랙과 융합된다.

2. 차량 및 센서 구성

VRU 인지와 추적을 위해 사용된 자율 주행 차량은 Fig. 1과 같다. Fig. 1에서 보는 바와 같이 전방에 4 채널 이베

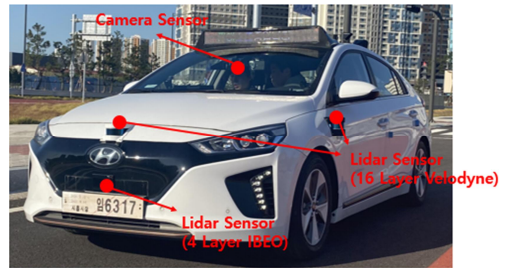


Fig. 1 Vehicle and sensors configuration

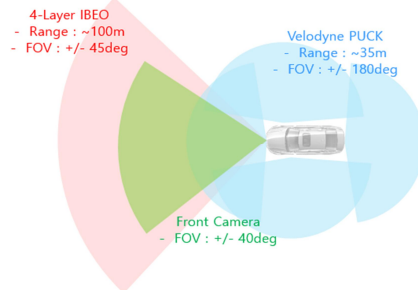


Fig. 2 A field of view (FOV) of sensing coverage

오 라이다 센서와 카메라 센서가 부착되어 있고, 차량의 4면에 각각 16 채널 벨로다인 라이다가 부착되어 있다. 해당 센서들의 FOV(Field of View)와 커버 범위는 Fig. 2와 같다.

3. 실시간 이미지 기반 VRU 분류: Yolo 알고리즘 적용

3.1. Yolo 알고리즘

Yolo 알고리즘은 Faster R-CNN에 비해 무려 6배 가량 빠른 속도를 가지고, 2배 가량 높은 mAP(mean average precision)을 보이는 혁신적인 물체 인식 알고리즘이다. Yolo의 작동 구조를 간단히 나타내면 Fig. 3과 같다. 이미지 입력을 그리드 영역으로 나누고, 물체가 존재할 가능성이 있는 영역에 물체 바운딩 박스를 예측한다. 추가로 각 바운딩 박스의 신뢰도가 계산된다. 이는 식 (1)과 같고, 여기서 IOU는 예측한 바운딩 박스와 참값인 실측 박스가 겹치는 비율을 나타낸다.

$$grid\ confidence = Pr(object) * IOU_{pred}^{truth} \quad (1)$$

따라서, 바운딩 박스는 중심의 픽셀 좌표 위치와 너비, 높이 그리고 신뢰도(x,y,w,h,c) 5차원 정보로 구성된다. 이와 동시에, 각 그리드마다 물체 분류 클래스별 확률도 계산된다. 이는 식 (2)와 같다.

$$class\ probability = Pr(class_i | object) \quad (2)$$

마지막으로, 해당 바운딩 박스가 특정 클래스일 확률 값은 식 (3)과 같이 계산된다.

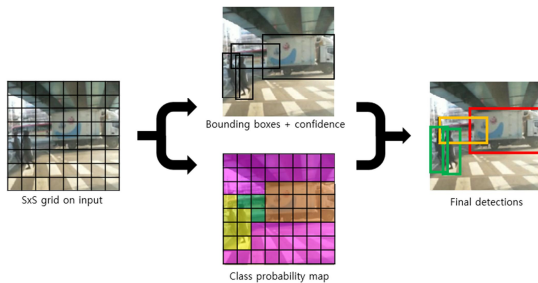


Fig. 3 A process of Yolo algorithm

class confidence

$$\begin{aligned} &= class\ probability * grid\ confidence \\ &= Pr(class_i | object) * Pr(object) * IOU_{pred}^{truth} \\ &= Pr(class_i) * IOU_{pred}^{truth} \end{aligned} \quad (3)$$

이러한 정보들을 이용한 네트워크 프로세스를 거쳐 입력 이미지로부터 바운딩 박스의 위치와 클래스를 추정한다. 해당 논문에서는 이렇게 도출된 Yolo 알고리즘의 분류 항목 가운데 관심 클래스인 보행자, 자전거, 승용차, 트럭 그리고 버스로 분류되는 물체 정보만 사용하였다.

3.2. 카메라 좌표 변환

픽셀 좌표 기준으로 도출된 물체의 바운딩 박스를 차량 좌표계 기준인 라이다 센서로 부터의 트랙 정보와 퓨전 하기 위해서는 좌표계 변환이 필요하다. 이 때, 하나의 평면을 다른 평면으로 투영하기 위해 사용되는 호모그래피(Homography) 행렬을 이용하였다. 호모그래피(Homography) 행렬은 식 (4)를 만족시키는 H 행렬이다.

$$\mathbf{x}_{i,vehicle} = H\mathbf{x}_{i,pixel} \quad (4)$$

이는 Fig. 4에 나타난 것과 같이 대응되는 각 좌표계에 서의 4쌍의 점에 대한 위치 정보만 있으면 도출 가능하다. 도출 과정은 아래 식 (5)~(9)와 같다.

$$\begin{bmatrix} x_{i,vehicle} \\ y_{i,vehicle} \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x_{i,pixel} \\ y_{i,pixel} \\ 1 \end{bmatrix} \quad (5)$$

$$\begin{aligned} x_{i,vehicle} &= \frac{h_{11}x_{i,pixel} + h_{12}y_{i,pixel} + h_{13}}{h_{31}x_{i,pixel} + h_{32}y_{i,pixel} + h_{33}} \\ y_{i,vehicle} &= \frac{h_{21}x_{i,pixel} + h_{22}y_{i,pixel} + h_{23}}{h_{31}x_{i,pixel} + h_{32}y_{i,pixel} + h_{33}} \end{aligned} \quad (6)$$

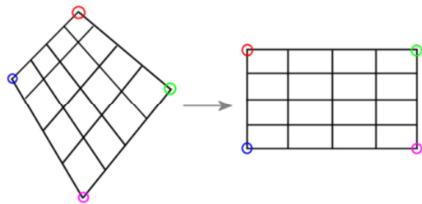


Fig. 4 A coordinate translation between two planes

$$\begin{aligned}
 & -(h_{11}x_{i,pixel} + h_{12}y_{i,pixel} + h_{13}) \\
 & + x_{i,vehicle} (h_{31}x_{i,pixel} + h_{32}y_{i,pixel} + h_{33}) = 0 \\
 & -(h_{21}x_{i,pixel} + h_{22}y_{i,pixel} + h_{23}) \\
 & + y_{i,vehicle} (h_{31}x_{i,pixel} + h_{32}y_{i,pixel} + h_{33}) = 0
 \end{aligned} \tag{7}$$

편의성을 위해 픽셀 좌표계는 x_i, y_i 그리고 변환된 차량 좌표계는 x'_i, y'_i 로 나타내었다.

$$\begin{bmatrix} -x_i & -y_i & -1 & 0 & 0 & 0 & x_i x'_i & x_i x'_i & x'_i \\ 0 & 0 & 0 & -x_i & -y_i & -1 & x_i y'_i & y_i y'_i & y'_i \end{bmatrix} \begin{bmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \\ h_{33} \end{bmatrix} = 0 \tag{8}$$

이와 같은 행렬식을 미리 알고있는 4쌍의 대응점에 대해서 적용한다면 다음과 같은 식이 도출된다.

$$\begin{bmatrix} -x_1 & -y_1 & -1 & 0 & 0 & 0 & x_1 x'_1 & x_1 x'_1 & x'_1 \\ 0 & 0 & 0 & -x_1 & -y_1 & -1 & x_1 y'_1 & y_1 y'_1 & y'_1 \\ -x_2 & -y_2 & -1 & 0 & 0 & 0 & x_2 x'_2 & x_2 x'_2 & x'_2 \\ 0 & 0 & 0 & -x_2 & -y_2 & -1 & x_2 y'_2 & y_2 y'_2 & y'_2 \\ -x_3 & -y_3 & -1 & 0 & 0 & 0 & x_3 x'_3 & x_3 x'_3 & x'_3 \\ 0 & 0 & 0 & -x_3 & -y_3 & -1 & x_3 y'_3 & y_3 y'_3 & y'_3 \\ -x_4 & -y_4 & -1 & 0 & 0 & 0 & x_4 x'_4 & x_4 x'_4 & x'_4 \\ 0 & 0 & 0 & -x_4 & -y_4 & -1 & x_4 y'_4 & y_4 y'_4 & y'_4 \end{bmatrix} \begin{bmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \\ h_{33} \end{bmatrix} = 0 \tag{9}$$

이를 통해 호모그래피 행렬의 각 원소의 값을 도출할 수 있다.

해당 연구에서는 이미지 상에서의 그라운드 평면을 차량 좌표계 상의 x-y평면으로 투영한다. 따라서 Yolo에서 도출된 물체의 바운딩 박스에서 그라운드와 접하는 사각형 하단의 두 점을 도출된 호모그래피 행렬을 이용해 차량 로컬 좌표계 상의 좌표로 변환한다. 이는 후에 도출된 라이다 트랙과 매칭하여 사용되게 된다.

4. 라이다 포인트 클라우드 기반 물체 인지

라이다 센서를 이용한 물체 인지 및 추적을 위해 물체의 기하학적 모델에 독립적으로 적용 가능한 GMFA 방식을 사용한다. 라이다 센서로부터 얻어지는 각각의 포인트 클라우드는 유클리디안 거리를 기반으로 군집화되며, 군집화된 클러스터는 다음 스텝에서 도출된 클러스터와 대응된다. 이는 클러스터 중심 간의 거리와 포인트들의 형태 유사성을 기반으로 한다. 다음으로 ICP를 이용하여 대응된 클러스터들을 매칭하고 EKF(Extended Kalman Filter)를 통해 물체 트랙의 상태를 업데이트한다. 여기서 라이다 트랙은 차량량의 뒷바퀴 축 기준 로컬 좌표계로 도출된다.

4.1. GMFT 기반 프로세스 보정

각 트랙은 먼저 이산 확장 칼만 필터의 타임 업데이트 과정을 거친다. 해당 모델은 질점 모델을 이용하여 식 (10)과 같다.

$$\begin{aligned}
 \dot{\mathbf{x}}_n &= \mathbf{a}(\mathbf{x}_n, \mathbf{u}) + \mathbf{q} \\
 &= [\mathbf{a}_1 \quad \mathbf{a}_2 \quad \mathbf{a}_3 \quad \mathbf{a}_4 \quad \mathbf{a}_5 \quad \mathbf{a}_6 \quad \mathbf{a}_7]^T + \mathbf{q} \\
 \mathbf{u} &= [v_x, \gamma] \\
 \mathbf{a}_1 &= v_{n,x} \cos \theta_n - v_x + p_{n,y} \cdot \gamma \\
 \mathbf{a}_2 &= v_{n,x} \sin \theta_n - p_{n,x} \cdot \gamma \\
 \mathbf{a}_3 &= \gamma_n - \gamma \quad \mathbf{a}_4 = a_{n,x} \\
 \mathbf{a}_5 &= \dot{\gamma}_n \quad \mathbf{a}_6 = -k_a \quad \mathbf{a}_7 = -k_\gamma \\
 \mathbf{q} &\sim (\mathbf{0}, \mathbf{Q})
 \end{aligned} \tag{10}$$

이를 통해 이전 스텝의 클러스터($\hat{Z}[\mathbf{k}-1]$)은 현재 스텝에 업데이트된 클러스터($\hat{Z}[\mathbf{k}]$)로 바뀐다. 이는 GNN을 이용해 현재 스텝에서 측정된 클러스터 중 대응되는 클러스터($Z[\mathbf{k}]$)를 찾기 위해 사용된다.

4.2. 트랙 매칭

클러스터로부터 물체의 추정 정보를 포함한 트랙은 $\mathbf{x}_n = [p_{n,x} \quad p_{n,y} \quad \theta_n \quad v_{n,x} \quad \gamma_n \quad a_{n,x} \quad \dot{\gamma}_n]^T$ 으로 클러스터의 중점 위치, 요각도, 속도, 요레이트, 가속도, 요레이트 미분 텀으로 구성된다.

GNN을 통해 찾아낸 대응 클러스터($Z[k]$)와 이전 클러스터($\hat{Z}[k-1]$)를 ICP 방법을 통해 매칭한다. 이를 통해 물체의 종방향 및 횡방향 위치, 요 각도가 도출되고 이를 EKF의 측정 업데이트를 위한 측정값으로 사용한다. 이 때, ICP 매칭으로 변화된 상태만 추정하고, 처음 클러스터 군집의 모양은 그대로 유지하여 클러스터 형상 변화에 따른 노이즈가 생기지 않도록 한다.

4.3. GMFT 기반 측정값 보정

트랙 매칭을 통해 도출된 클러스터의 3가지 측정값을 바탕으로 EKF의 측정값 보정 단계를 적용한다. 즉, 매칭된 $\hat{Z}[k-1]$ 를 기반으로 프로세스 업데이트된 $\bar{Z}[k]$ 를 보정하는 것이다. 사용된 측정 모델은 식 (11)과 같다.

$$\begin{aligned} \mathbf{z}_n[k] &= \mathbf{H}_n \mathbf{x}_n[k] + \mathbf{v}_n[k] \\ \mathbf{v}_n[k] &\sim N(\mathbf{0}, \mathbf{V}_n[k]) \end{aligned} \quad (11)$$

$$\mathbf{H}_n = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

5. 카메라 및 라이다 상위 레벨 센서 융합

최종적으로 라이다 트랙과 좌표 변환된 변환된 카메라 트랙을 융합해야 한다. 하지만, 카메라 트랙의 위치 정확성은 상당히 떨어지므로 물체 위치를 추정하기에 부적합하다. 반면, 자차 기준으로 카메라 트랙이 위치하는 각도 정보는 상당히 높은 정확도를 지닌다. 조사된 라이다 트랙 기준 카메라 트랙의 각도 오차는 Fig. 5와 같다. 이는 Fig. 1에서 언급한 차량으로 획득한 주행 데이터를 기반으로 하며, 약 3000개의 매칭된 트랙의 누적 데이터를 이용하였다. Fig. 5는 자차 기준의 종방향 및 횡방향 거리 각각에 따른 오차 절대값의 평균값을 나타낸 것이다. 각 점선은

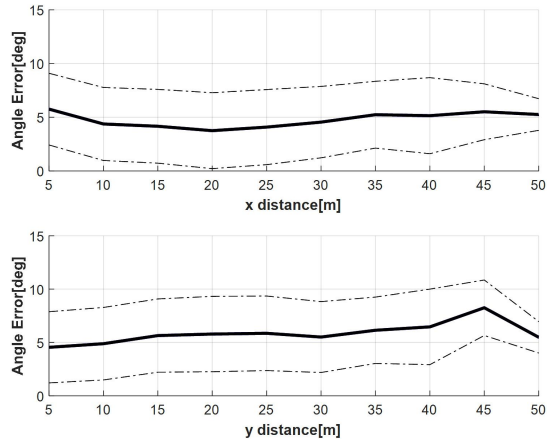


Fig. 5 Mean of angle error for longitudinal and lateral distance

1시그마를 나타낸다.

카메라 트랙의 각도 정보의 경우 중, 횡방향 거리에 독립적이며 평균적으로 약 5도, 그리고 약 68% 가량의 데이터가 10도 이내의 값을 보이는 것으로 확인되었다. 해당 오차는 센서 간의 타임 싱크 문제에서 기인된 것이라고 예측된다. 라이다 센서가 20Hz 그리고 카메라 센서가 15Hz로 정보가 도출되기 때문에 매칭되는 정보간의 시간 오차가 생긴다. 해당 시간 오차를 감안하여 일정 각도 마진을 두어 매칭을 진행하였다.

트랙 매칭은 카메라 트랙의 각도 정보를 이용하여 해당 각도에 대응하는 라이다 트랙 중 GNN방식을 통해 매칭을 진행했다. 또한, 매칭된 라이다 트랙에는 카메라 트랙의 분류 클래스 ID를 부여하였다.

6. 실험 결과

해당 알고리즘은 Linux 상의 Robot Operating System (ROS) 플랫폼으로 개발하였으며, Fig. 1과 같은 라이다와 카메라 센서가 구축된 실험 차량에 적용하여 결과를 도출하였다. 이 때, 관심 영역(ROI)은 종방향으로 50m, 횡방향으로 ± 8 m로 설정했으며, 특정 시점의 보행자 인지 결

Table 1 Quantitative evaluation of our fusion algorithm

Object Class	Total Object	Correctly Tracked Lidar Track	Correctly Tracked Vision Track	Correctly Tracked Both Track	Correctly Matched Track	Matching Performance (%)
Vehicle	2831	2812	2796	2782	2319	83.35
VRU	246	235	228	221	174	78.73

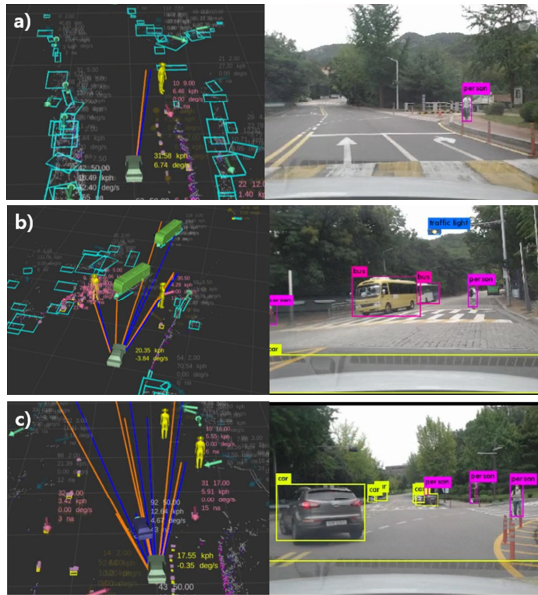


Fig. 6 Several situation applying lidar-camera track matching

과를 시각화하여 Fig. 6에 나타내었다. 파란색 박스는 클러스터링된 라이다 트랙이며, 자차로부터 직선으로 그려진 선은 카메라 트랙의 각도 정보를 나타낸 것이다. 최종 매칭된 트랙은 분류 클래스에 맞게 3D이미지로 나타내었다. 매칭 결과에 대한 정량적인 분석은 Table 1에 정리되었다.

7. 결 론

본 연구에서는 카메라 센서의 이미지 정보와 라이다 센서의 포인트 클라우드 정보를 후처리하여, 도로 위의 주요 객체들을 분류 및 추적하는 센서 융합 알고리즘을 제시하였다. 여기서 이미지 정보 처리는 실시간성과 정확성이 입증된 Yolo 알고리즘을 응용하였고, 포인트 클라우드 정보 처리는 GNN을 활용한 클러스터링 및 ICP와 EKF를 적용한 객체 상태 추적을 통해 트랙을 도출하였다. Yolo 알고리즘의 아웃풋인 바운딩 박스를 좌표 변환하여 라이다 트랙과 매칭함으로써 자율 주행 차의 움직임을 계획하는 데 필수적인 객체의 정보들을 도출해 낼 수 있었다.

해당 알고리즘은 실차 실험을 통한 데이터를 바탕으로 유효성을 검증하였고, 타 차량에 대해서는 약 81% 그리고 차량을 제외한 자전거 및 보행자에 대해서는 약 70%의 매칭률을 보였다. 이는 추후 카메라 트랙의 위치 오차에

대한 보정 및 라이다와 카메라 센서 사이의 타임 싱크를 고려하여 더욱 고도화할 수 있을 것으로 예측된다.

후 기

본 연구는 국토교통부 도심도로 자율협력주행 안전·인프라 연구 사업의 연구비지원(과제번호 19PQOW-B152473-01)에 의해 수행되었습니다.

참고문헌

- (1) Banerjee Koyel, et al., 2018, "Online camera lidar fusion and object detection on hybrid data for autonomous driving", IEEE Intelligent Vehicles Symposium (IV).
- (2) Cho Hyunggi, et al., 2014, "A multi-sensor fusion system for moving object detection and tracking in urban driving environments", IEEE International Conference on Robotics and Automation (ICRA).
- (3) Gao Hongbo, et al., 2018, "Object classification using CNN-based fusion of vision and LIDAR in autonomous vehicle environment", IEEE Transactions on Industrial Informatics 14.9: pp. 4224~4231.
- (4) Chavez-Garcia, Ricardo Omar, and Olivier Aycard, 2015, "Multiple sensor fusion and classification for moving object detection and tracking", IEEE Transactions on Intelligent Transportation Systems 17.2: pp.525~534.
- (5) Wang, Dominic Zeng, Ingmar Posner, and Paul Newman, "Model-free detection and tracking of dynamic objects with 2D lidar", The International Journal of Robotics Research 34.7: pp. 1039~1063.
- (6) Thuy, Michael, and Fernando Puente Leon, 2009, "Non-linear, shape independent object tracking based on 2d lidar data", IEEE Intelligent Vehicles Symposium.
- (7) Johnsen, Swantje, and Ashley Tews, 2009, "Real-time object tracking and classification using a static camera", Proceedings of IEEE International Conference on Robotics and Automation, workshop on People Detection and Tracking.
- (8) Lee Hojoon, et al., 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.
- (9) Redmon Joseph, et al., 2016, “You only look once: Unified, real-time object detection”, Proceedings of the IEEE conference on computer vision and pattern recognition.
- (10) Vincent, Etienne, and Robert Laganière, 2001, “Detecting planar homographies in an image pair”, ISPA. Proceedings of the 2nd International Symposium on Image and Signal Processing and Analysis. In conjunction with 23rd International Conference on Information Technology Interfaces.