

# 고속 자율주행 레이싱을 위한 Camera-LiDAR 센서 퓨전 인지 알고리즘

봉은정·최연호·안광림·기석철\*

충북대학교 지능로봇공학과

## Camera-LiDAR Sensor Fusion Cognitive Algorithm for High-Speed Autonomous Racing

Eun-Jung Bong · Youn-ho Choi · Gwangrim Ahn · Seok-Cheol Kee\*

Department of Intelligent Systems and Robotics, Chungbuk National University, Chungcheongbuk-do 28644, Korea  
 (Received 9 December 2024 / Revised 3 January 2025 / Accepted 4 January 2025)

**Abstract** : Autonomous driving technology has made significant advancements in recent years, and various autonomous racing competitions have influenced the development of autonomous driving technologies in universities and research institutions. However, in the extreme environment of high-speed racing, implementing a reliable autonomous driving system remains a challenging task. This paper proposes a Camera-LiDAR sensor fusion algorithm to improve object detection performance in high-speed and bank-road environments, and achieve processing speeds of 10 Hz by participating in the 2024 College Student Autonomous Driving Competition. The proposed algorithm improves the calibration accuracy between the camera and LiDAR through internal, external, and temporal calibration. The 2D object detection algorithm utilizes YOLOv5, while the 3D object detection algorithm utilizes PointPillars. The 2D and 3D object detection results are fused, based on late fusion. The proposed algorithm was tested in the KIAPI proving ground environment, and it performed reasonably.

**Key words** : Autonomous driving(자율주행), Sensor fusion(센서 퓨전), 2D object detection(2D 객체 검출), 3D object detection(3D 객체 검출), Sensor calibration(센서 캘리브레이션), Racing competition(레이싱 대회)

### Nomenclature

BEV : bird eye view

RMSE : root mean squared error

$f_x$  : focal length x

$f_y$  : focal length y

$c_x$  : principal point x

$c_y$  : principal point y

$k_1$  : radial distortion coefficient 1

$k_2$  : radial distortion coefficient 2

$k_3$  : radial distortion coefficient 3

$p_1$  : tangential distortion coefficient 1

$p_2$  : tangential distortion coefficient 2

$R$  : rotation matrix

$T$  : translation matrix

$A$  : 3D coordinates of 2D object detection result

$B$  : 3D coordinates of 3D object detection result

$d$  : distance between two objects

$P$  : 2-dimension feature point

$\tilde{P}$  : re-projected 3-dimension feature point

### 1. 서론

자율주행 레이싱은 최근 몇 년간 자율주행 분야에서 새롭게 주목받아 왔다. 특히 DARPA,<sup>1)</sup> Roborace,<sup>2)</sup> Indy Autonomous Challenge<sup>3)</sup> 등 다양한 자율주행 레이싱 대회는 여러 대학팀을 비롯하여 기업, 연구 기관의 자율주행 기술 발전에 큰 영향을 주었지만, 고속 레이싱이라는 극

\*A part of this paper was presented at the KSAE 2024 Fall Conference and Exhibition

\*Corresponding author, E-mail: sckee@chungbuk.ac.kr

<sup>†</sup>This is an Open-Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License(<http://creativecommons.org/licenses/by-nc/3.0>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium provided the original work is properly cited.

한의 환경에서 신뢰성 있는 자율주행 시스템을 구현하는 것은 여전히 어려운 과제로 남아있다. 100 km/h 이상의 고속 주행 환경에서는 주변 객체를 실시간으로 정확하게 인지해야 하며, 이는 판단 및 제어를 효과적으로 수행하여 자율주행 시스템의 신뢰성을 확보하는 데 필수적이다.

딥러닝의 발전과 함께 카메라, 라이다 센서 데이터를 기반으로 한 2D 및 3D 객체 검출 연구는 괄목할 만한 성장을 이루었다. 2D 객체 검출에는 Convolutional Neural Network(CNN) 백본을 활용한 One-stage 및 Two-stage detector 접근 방식과, Transformer 백본을 활용한 접근 방식이 존재한다. One-stage detector는 2D 바운딩박스의 좌표를 예측하는 Localization과 클래스를 예측하는 Classification을 동시에 수행하는 구조를 가진다. 이러한 구조는 정확도가 제한적이지만, 실시간 객체 검출에 적합하여 주로 자율주행 분야에서 많이 활용된다. 대표적인 모델로는 YOLO<sup>4)</sup>와 SSD<sup>5)</sup>가 있다. Two-stage detector는 Coarse-fine 프로세스를 통해 Localization과 Classification을 별도로 수행하는 구조를 가지며, 상대적으로 높은 정확도를 제공하지만, 객체 검출 속도가 낮다. 대표적인 모델로는 Fast Region-based Convolutional Neural Network (R-CNN),<sup>6)</sup> Faster R-CNN<sup>7)</sup> 등이 있다. CNN 기반 모델은 국소적인 특징을 추출하는 데 특화된 반면, Transformer 기반 모델은 Attention<sup>8)</sup> 메커니즘을 활용하여 이미지의 전역적인 정보를 효율적으로 처리할 수 있다. 대표적인 모델로는 DETR,<sup>9)</sup> SWIN<sup>10)</sup> 등이 있다. 3D 객체 검출에는 포인트 기반 접근 방식과 그리드 기반 접근 방식이 존재한다. 포인트 기반 접근 방식은 포인트클라우드 데이터에 점진적인 샘플링 과정을 반복하며 특징을 추출하고 객체를 검출하는 방식으로, 3DSSD<sup>11)</sup>는 특징 공간 샘플링 기법을 제안하였으며, Point-GNN<sup>12)</sup>는 Voxel 기반 샘플링을 제안하였다. 그리드 기반 접근 방식은 포인트클라우드를 3D voxel, pillar, BEV 등의 그리드 형태로 분할한 후 2D 또는 3D 신경망을 적용하는 방식으로, VoxelNet<sup>13)</sup>은 포인트클라우드를 3D voxel로 분할한 후 3D CNN을 적용하였으며, PointPillars<sup>14)</sup>는 포인트클라우드를 여러 개의 Pillar로 변환한 후 2D CNN을 적용하였다.

이처럼 카메라 또는 라이다 데이터를 기반으로 한 다양한 2D 및 3D 객체 검출 연구가 존재하지만, 단일 센서로 구성된 인지 알고리즘에는 여전히 한계가 존재한다. 반면, 다중 센서를 활용할 경우 각 센서의 장점을 극대화하고, 리던던시를 통해 단일 센서의 한계를 보완할 수 있어 다양한 센서 퓨전 연구가 활발히 수행되고 있다.

센서 퓨전은 데이터를 융합하는 시점에 따라 Early fusion, Deep fusion, Late fusion으로 나눌 수 있다. Early

fusion은 각 센서의 미가공 데이터를 융합하는 접근 방식으로, 서로 다른 양식의 정보를 적극적으로 활용할 수 있다. VirConv<sup>15)</sup>는 Depth completion을 기반으로 생성한 가상 포인트클라우드에 StVD와 NRConv를 적용하여 노이즈를 제거한 후, 이를 이미지 데이터와 융합하였다. PointPainting<sup>16)</sup>은 포인트클라우드를 2D semantic segmentation 이미지에 투영한 후, 클래스 정보를 가진 포인트클라우드를 재구성하여 이를 3D 객체 검출 모델에 입력하였다. Deep fusion은 서로 다른 미가공 데이터에서 추출한 특징을 융합하는 접근 방식으로, 다양한 형태의 특징을 효과적으로 융합하는 방법에 초점을 맞춘 연구가 수행되고 있다. PointAugmenting<sup>17)</sup>은 포인트클라우드를 이미지의 특징맵으로 투영한 후, 각 포인트에 대한 이미지 특징과 포인트클라우드 특징을 별도의 Convolution 블록에 입력하여 서로 다른 양식 간의 차이를 효과적으로 연결하였다. BEVFusion<sup>18)</sup>은 이미지의 2D 특징과 포인트클라우드의 3D 특징을 BEV 공간으로 변환하여, 카메라의 Semantic 정보와 라이다의 기하학적 구조를 보존하면서 특징을 효과적으로 융합하였다. Late fusion은 서로 다른 센서 데이터에서 추출된 객체 검출 결과를 융합하는 접근 방식으로, 하나의 센서가 오작동하는 경우에도 정상 작동하는 센서의 객체 검출 결과를 활용할 수 있다. 이러한 방식은 다른 센서 퓨전 기법에 비해 안정적인 인지 성능을 제공하며, 자율주행과 같이 실시간 고성능 객체 인지가 요구되는 분야에 적합하다. CLOCs<sup>19)</sup>는 각 객체 검출 모델의 Non Maximum Suppression(NMS) 전 단계 데이터를 입력으로 활용하여, 2D 객체 후보와 3D 객체 후보 간의 의미론적, 기하학적 일관성을 기반으로 최종 객체 검출 결과를 산출하는 네트워크를 제안하였다.

본 논문은 2024 대학생 자율주행 경진대회에 참가하여 고속 주행 및 뱅크 도로 환경에서 객체 인지 성능을 향상시키고, 10 Hz의 처리 속도를 달성하기 위한 카메라-라이다 센서 퓨전 알고리즘을 제안한다. 본 논문의 기여는 다음과 같다.

1. 카메라-라이다 캘리브레이션을 수행하여 서로 다른 양식의 데이터를 융합하고, 높은 정합 정밀도를 달성하였다.
2. 10 Hz 이상의 처리 속도를 구현한 Late fusion 기반 객체 인지 알고리즘을 활용하여 100 km/h 이상의 고속 주행 상황에서 실시간 객체 검출을 구현하였다.
3. 유클리드 거리 기반 Late fusion을 적용함으로써 단일 센서 기반 객체 검출 결과에 비해 합리적인 성능을 확보하였다.

본 논문의 구성은 다음과 같다. 2장에서는 센서 퓨전

을 위한 센서 캘리브레이션 기법을 다룬다. 3장에서는 2D 및 3D 객체 검출 알고리즘과 이를 기반으로 한 Late fusion을 설명한다. 4장에서는 실험 환경을 소개하고, 5장에서는 성능 평가 결과를 분석한다. 6장에서는 연구의 결론을 도출하고, 향후 연구 방향을 제시한다.

## 2. 센서 캘리브레이션

다중 센서 퓨전 기반 인지 알고리즘은 단일 센서의 한계를 극복하고, 각 센서의 장점을 극대화하여 자율주행 인지 성능 향상을 위한 핵심 접근법으로 채택되어 왔다. 카메라는 풍부한 색상 및 텍스처 정보를 제공하여 작은 객체의 특징을 효과적으로 추출할 수 있지만, 깊이 정보에는 제한이 있다. 반면 라이다는 텍스처 정보가 부족하지만, 정확한 거리 및 깊이 정보를 기반으로 3D 환경에서 객체를 정밀하게 모델링할 수 있다. 본 논문에서는 단거리 및 장거리 객체를 검출하고, 3차원 공간에서의 정확한 위치 좌표를 계산하기 위해 카메라와 라이다로 구성된 다중 센서 시스템을 채택하였다.

다중 센서 퓨전을 효과적으로 수행하기 위해서는 다양한 센서 캘리브레이션이 선행되어야 한다. 센서 캘리브레이션에는 센서 내부 요인으로 인한 왜곡 오차를 보정하는 내부 캘리브레이션, 서로 다른 위치에 장착된 센서 간 좌표 관계를 계산하는 외부 캘리브레이션, 그리고 각 데이터를 시간적으로 동기화하는 Temporal 캘리브레이션이 포함된다. 본 논문에서는 센서 퓨전 성능을 향상시키기 위해 내부, 외부, Temporal 캘리브레이션을 수행하였으며, 이를 통해 카메라 이미지와 라이다 포인트클라우드 간의 정합 정밀도를 개선하였다.

### 2.1 Intrinsic 및 Extrinsic 캘리브레이션

라이다 제조사는 직접 내부 캘리브레이션을 수행한 후 제품을 제공하므로, 대부분의 포인트클라우드 데이터는 높은 신뢰성을 갖는다. 반면 카메라는 기구적 요소, 렌즈 품질, 정렬 과정 등 여러 요인에 의해 영향을 받으므로 최적의 성능을 보장하기 위해서는 내부 캘리브레이션을 반드시 수행해야 한다. 본 논문에서는 Perspective<sup>20)</sup> 모델을 활용하여 자율주행 차량 전후방에 장착된 협각 카메라의 내부 파라미터와 왜곡 계수를 계산하였다. 식 (1)은 렌즈의 초점 및 주점으로 구성된 카메라 행렬을 나타내며 식 (2)는 렌즈의 방사 및 접선 왜곡 계수 행렬로, 이미지 왜곡을 보정하는 데 활용된다. 식 (3)은 카메라 행렬 및 회전-이동 행렬을 결합한 투영 행렬 기반 좌표계 변환을 나타내며, Fig. 1과 같이 좌표계 변환을 통해 왜곡이 보정된 이미지에 포인트클라우드를 투영하여 시각화

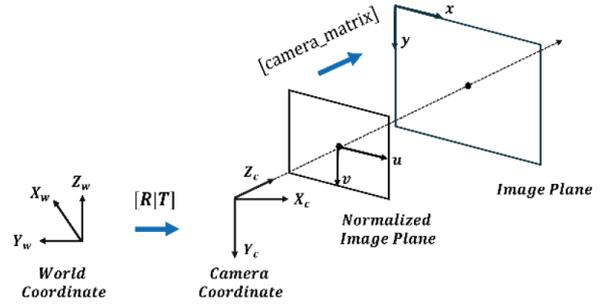


Fig. 1 Coordinate system transformation and projection process

할 수 있다.

$$camera\_matrix = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \quad (1)$$

$$distortion\_coefficient = [k_1 \ k_2 \ p_1 \ p_2 \ k_3]^T \quad (2)$$

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = [camera\_matrix][R|T] \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \quad (3)$$

다중 센서로부터 검출된 객체를 정확하게 추적하고, 회피하기 위해서는 서로 다른 좌표계에서 검출된 객체를 기준 좌표계로 변환하는 과정이 필요하다. 부정확한 외부 캘리브레이션은 인지 알고리즘의 신뢰성을 저하시킬 수 있으므로, 자율주행 시스템의 안전성을 확보하기 위해서는 정밀한 외부 캘리브레이션이 필수적이다.

본 논문에서는 체커보드를 활용하여 타겟 기반 카메라-라이다 외부 캘리브레이션<sup>21)</sup>을 수행하였다. 이 과정에서 PnP<sup>22)</sup>를 기반으로 두 좌표계 사이의 회전 행렬과 이동 행렬을 추정하였으며, PnP를 수행하기 위해 이미지와 포인트클라우드로부터 체커보드의 2D 및 3D 특징점 쌍을 추출하였다. 이미지는 풍부한 텍스처를 바탕으로 원거리의 2D 특징점을 정확하게 추출할 수 있지만, 포인트클라우드는 텍스처 정보가 부족하여 거리가 멀어질수록 정확한 3D 특징점 추출이 어렵다. 본 논문에서는 Fig. 2(a)와 같이 텍스처 정보를 강화하기 위해 포인트클라우드를 누적하여 체커보드 맵을 생성하였으며, 4~20 m 범위의 특징점 쌍을 추출하고, 매칭하여 외부 캘리브레이션의 정밀도를 향상시켰다. Fig. 2(b)와 (c)는 정지 상태에서 취득한 3D 및 2D 특징점을 나타낸다.

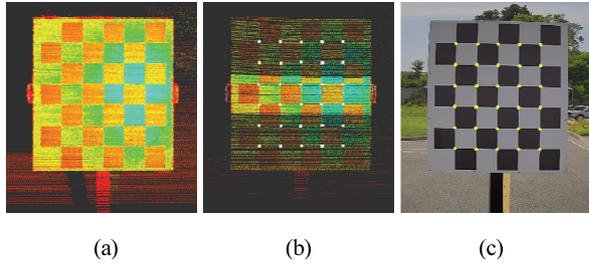


Fig. 2 Checkerboard map and 2D & 3D features

### 2.2 Temporal 캘리브레이션

각 센서의 데이터 취득 주기는 종류에 따라 다르다. 실제 주행 환경에서 데이터를 융합하기 위해서는 센서 데이터 취득 주기를 일치시켜야 하며, 동일한 시점에 취득된 데이터를 선별해야 한다. 특히 서로 다른 하드웨어에서 센서 데이터를 취득하는 경우에는 하드웨어 간 시간 동기화 작업이 필수적으로 수행되어야 한다. 본 논문에서는 동일한 시점의 이미지와 포인트클라우드를 선별하기 위해 센서의 주기를 고려하여 데이터의 타임스탬프를 조절하였으며, ROS의 Message filter<sup>23)</sup>를 활용하여 타임스탬프 일치 여부를 확인하였다. 또한, Chrony<sup>24)</sup>를 활용하여 하드웨어 간 시간 동기화를 수행하였으며 그 결과 200 ns ~ 20 μs 사이의 시간 오프셋이 발생함을 확인하였다. ns 및 μs 수준의 오프셋은 제안하는 인지 알고리즘에 큰 영향을 미치지 않으므로, 동기화된 하드웨어 시간 데이터를 신뢰하여 활용하였다. Fig. 3(a)와 (b)는 포인트클라우드를 이미지로 투영한 결과이며, 각각 Temporal 캘리브레이션 수행 전, 후를 나타낸다.

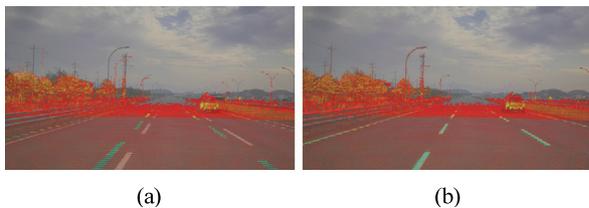


Fig. 3 Point cloud projection results before & after temporal calibration

## 3. 객체 인지 알고리즘

본 논문에서 제안하는 인지 알고리즘의 전체 구조는 Fig. 4에서 확인할 수 있으며, 2D Object Detection, 3D Object Detection, Late Fusion의 세 가지 주요 프로세스로 구성된다. 해당 구조는 동일한 객체가 중복으로 검출되는 특성으로 인해 하나의 프로세스에서 오류가 발생하더라도 시스템의 안정성을 유지할 수 있다. 또한 2D 객

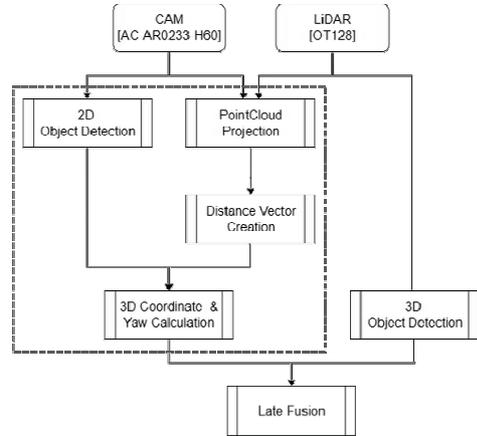


Fig. 4 Diagram of sensor fusion based object detection algorithm

체 검출 알고리즘은 이미지의 텍스처 정보를 활용하여 소형 객체 검출에 우수한 성능을 보이므로, 이를 3D 객체 검출 결과와 결합함으로써 전체적인 객체 검출 성능을 향상시킬 수 있다.

### 3.1 2D Object Detection

본 논문에서는 YOLOv5<sup>25)</sup>를 기반으로 설계된 2D 객체 검출 알고리즘을 활용하였다. Fig. 5는 YOLOv5의 구조를 나타내며, Backbone, neck, head의 3가지 주요 구성요소로 이루어진다. Backbone은 이미지에서 특징을 추출하는 부분으로, YOLOv5는 CSPNet<sup>26)</sup> 기반의 CSP Darknet을 채택하였다. CSP Darknet은 특징을 효율적으로 추출하는 CBS 모듈, 다양한 스케일의 특징을 통합하는 SPPF 모듈, 그리고 특징맵 일부에만 연산을 수행하는 C3 모듈로 구성되어 계산 효율성을 높이고 실시간 성능을 극대화할 수 있다. Neck은 특징을 고도화하는 부분으로, PAN 구조를 활용하여 하향식 및 상향식 경로를 결합함으로써 다중 스케일을 융합하고, 객체 검출 성능을 향상시킨다. Head는 객체의 정보를 예측하는 부분으로, 다양한 스케일에서 바운딩박스, 클래스 확률, 존재 확률을 계산하여 객체의 정확한 위치와 크기를 예측한다.

YOLOv5는 Backbone의 너비와 깊이에 따라 YOLOv5n, YOLOv5s, YOLOv5m, YOLOv5l, YOLOv5x로 나뉘며, 규모가 커질수록 객체 검출 성능은 향상되지만, 검출 속도는 감소한다. 본 논문에서는 실시간 객체 검출을 위해 YOLOv5n 모델을 채택하였다. 해당 모델은 객체 검출 정확도가 다소 낮지만, 전후방 이미지를 각각 30 Hz 이상의 응답 속도로 처리하는 데 적합하다. 최종적으로는 BDD100K<sup>27)</sup> 데이터셋과 지능형자동차부품진흥원 대구주행 시험장(KIAPI)에서 취득한 자율주행 차량 데이터

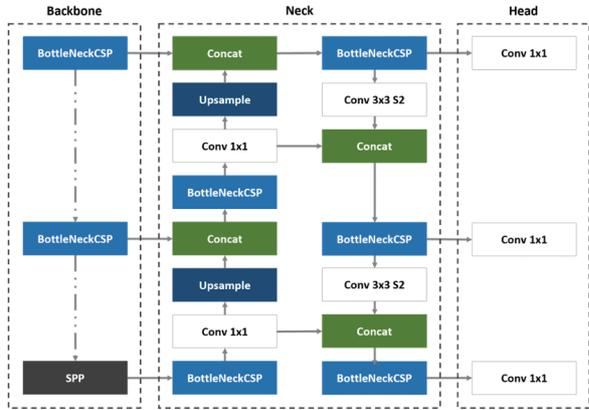


Fig. 5 Diagram of 2D object detection algorithm

를 활용하여 모델 학습을 수행하였으며, 100 km/h 이상의 고속 주행 환경에서 전후방 150 m 범위 내의 객체를 30 Hz로 실시간 검출할 수 있는 성능을 구현하였다.

### 3.2 3D Object Detection

본 논문에서는 PointPillars를 기반으로 설계된 3D 객체 검출 알고리즘을 활용하였다. 기존 PointPillars 모델은 포인트클라우드를 수직으로 인코딩하여 Pseudo Image 형식으로 변환한 후 객체를 검출한다. 이러한 방식은 포인트 클라우드의 밀도와 분포를 기반으로 다양한 특징을 추출하여 정확한 객체 검출이 가능하지만, 처리 속도가 낮아 실시간 객체 검출에 적합하지 않다. 따라서 본 논문에서는 실시간으로 객체를 검출하기 위해 객체의 존재 여부를 바이너리 맵으로 표현하는 Bool VFE 방식을 채택하였다. Bool VFE는 포인트클라우드를 볼륨 기반 바이너리맵으로 변환한 후, 각 포인트를 바이너리 블록으로 매핑하여 물체의 존재 여부를 0과 1로 간단히 표시한다. 이를 통해 복잡한 딥러닝 기반 특징 추출 과정을 생략하여 시스템 복잡성을 줄였으며, CUDA GPU 프로그래밍, TensorRT<sup>28)</sup> 가속화 처리를 통해 객체 검출 속도를 향상시켰다.

Fig. 6은 PointPillars의 구조를 나타내며, 특징을 추출하는 Pre-process, TensorRT를 사용하여 바운딩박스, 클래스 확률, 존재 확률을 계산하는 TRT, 그리고 객체의 바운딩박스 정보를 최종 예측하는 Post-process의 3가지 주요 구성 요소로 이루어진다. 본 논문에서는 전방 객체와의 충돌 회피를 위해 객체 인지 후 최소 3.6초의 시간이 필요하다고 판단하였고, 이를 만족하기 위해서는 100 km/h로 주행 시 최소 100 m 이상의 탐지 범위가 요구됨을 확인하였다. 따라서 3D 객체 검출 모델 설계 시, 후방에 비해 전방의 탐지 범위를 더 길게 설정하였다. 최종적으로는 Nusences<sup>29)</sup> 데이터셋을 활용하여 모델 학습을 수행하

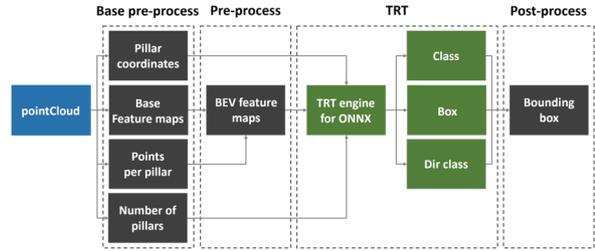


Fig. 6 Diagram of 3D object detection algorithm

였으며, 전방 128 m, 후방 25 m 범위 내의 객체를 10 Hz로 실시간 검출할 수 있는 성능을 구현하였다.

### 3.3 Late Fusion

본 논문에서는 2D 및 3D 객체 검출 결과를 융합하기 위해 Late fusion을 적용하였다. 이때 2D 객체 검출 모델에서 검출된 객체의 3D 좌표를 계산하기 위해 Fig. 4에 제시된 2D Object Detection, PointCloud Projection, Distance Vector Creation, 3D Coordinate Calculation의 네 가지 프로세스를 활용하였다. Fig. 7(a)는 YOLOv5를 기반으로 자율주행 차량을 검출하고 바운딩박스로 시각화한 결과를 나타내며, Fig. 7(b)는 포인트클라우드를 이미지로 투영한 결과를 나타낸다. 이 과정에서 이미지 픽셀에 투영된 각 포인트클라우드의 X, Y, Z값을 벡터 형태로 변환하여 거리 정보를 보존하였다. 이후 Fig. 7(c)와 같이 포인트클라우드 거리 정보를 보유한 픽셀 중 객체의 바운딩박스 중심 좌표와 가장 근접한 픽셀을 선별하였으며, 해당 픽셀의 거리 정보를 기반으로 Fig. 7(d)와 같이 객체의 3차원 좌표를 복원하였다. Fig. 8(b)와 (c)는

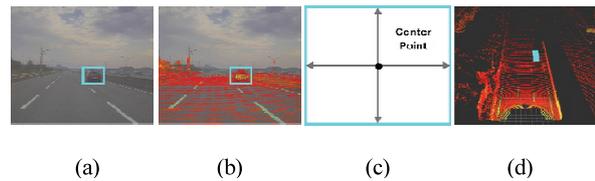


Fig. 7 3D visualization process of 2D object detection result

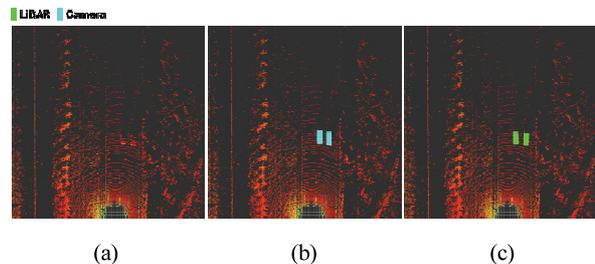


Fig. 8 3D visualization of 2D and 3D object detection result

(a)의 객체에 대해 2D 및 3D 객체 검출 결과를 3차원 공간에서 시각화한 결과를 나타낸다.

2D 및 3D 객체 검출 결과를 3차원 공간에서 융합하고, 최종 객체를 선별하기 위해 유클리드 거리를 기반으로 객체를 구분하였다. 서로 다른 객체들의 중심 좌표 간 유클리드 거리가 2 m 이내일 경우 중복으로 검출된 객체로 간주하였으며, 2 m 이상인 경우 각 센서에서 검출된 개별 객체로 처리하였다. 3D 객체 검출 결과는 2D 객체 검출 결과에 비해 더 정밀한 yaw 값을 제공하므로, 중복으로 검출된 객체에 대해서는 3D 객체 검출 결과를 우선적으로 채택하였다. 식 (4)는 유클리드 거리 계산식을 나타내며, A와 B는 각각 2D 및 3D 객체 검출 모델에서 검출된 객체의 3D 좌표를 나타낸다.

$$d(A, B) = \sqrt{(A_x - B_x)^2 + (A_y - B_y)^2 + (A_z - B_z)^2} \quad (4)$$

카메라 데이터는 40 Hz, 라이다 데이터는 10 Hz의 주기로 입력되었으며, 2D 객체 검출 알고리즘은 30 Hz, 3D 객체 검출 및 Late fusion 알고리즘은 라이다 데이터 주기에 맞춰 10 Hz로 구현하였다. 본 논문에서는 데이터 처리 및 알고리즘 주기를 적절하게 조절하여 뱅크 도로와 직선 도로로 구성된 고속 주회로 환경에서 10대의 자율주행 차량을 실시간으로 인지할 수 있음을 실험을 통해 확인하였다.

#### 4. 실험 환경

본 논문에서는 Fig. 9에 제시된 것과 같이 2023년형 AVANTE CN7 차량에 카메라-라이다 다중 센서 시스템을 구축하였다. 이때 후방 카메라는 전방 카메라를 기준으로 반대 방향을 향하므로 좌표축이 역방향으로 설정되었다. 차량의 전후방에는 장거리 객체 검출을 위해 60° 화각과 FHD 해상도를 제공하는 AR-0233 카메라를 장착하였으며, 상단에는 객체 검출 성능을 향상시키기 위해 주변 환경을 고해상도로 스캔하여 3D로 재구성하는 OT128 라이다를 장착하였다.

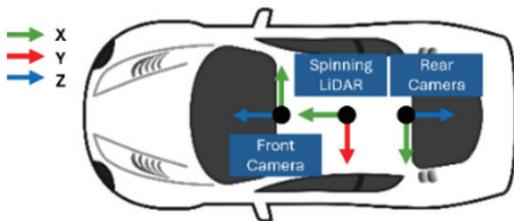


Fig. 9 Sensor configuration

Table 1 Sensor and PC information

	Type	Model
Sensors	Camera	AR-0233
	Spinning LiDAR	OT128
Systems	Sensor processing module	NRU-230V
	Main processing module	NUVO-10108GC-NX

인지 시스템의 처리 효율을 높이기 위해 카메라 데이터를 처리하는 PC와, 인지 알고리즘 및 라이다 데이터를 처리하는 PC를 구분하여 활용하였다. Table 1은 실험에 활용된 인지 센서 및 PC의 상세 구성을 나타낸다. 카메라 데이터는 NRU-230V에서 처리하였으며, 인지 알고리즘 및 라이다 데이터는 NUVO-10108GC-NX에서 처리하였다.

### 5. 성능 평가

#### 5.1 카메라-라이다 외부 캘리브레이션 성능 평가

카메라-라이다 외부 캘리브레이션 성능을 평가하기 위해 식 (3)을 사용하여 3D 특징점을 960×540 해상도 이미지로 투영한 후, 실제 2D 특징점과의 픽셀 오차를 계산하였다. 식 (5)는 재투영 RMSE 계산식을 나타내며,  $P_i$ 는 2D 특징점을,  $\hat{P}_i$ 는 이미지로 투영된 3D 특징점을 나타낸다. Table 2는 포인트클라우드와 전방 및 후방 이미지 간의 정합 정밀도를 재투영 RMSE 평가지표를 통해 나타내며, 다양한 거리에서 1픽셀 이내의 오차를 보이는 것을 확인하였다.

$$Reprojection_{RMSE} = \frac{\sqrt{\sum_{i=1}^N (P_i - \hat{P}_i)^2}}{N} \quad (5)$$

Table 2 Camera-LiDAR extrinsic calibration reprojection RMSE

Distance (m)	Reprojection RMSE (pixel)	
	OT128 & Front	OT128 & Rear
4 - 6	0.8784	0.9212
6 - 8	0.7882	0.8467
8 - 10	0.8377	0.7545
10 - 12	0.7716	0.7260
12 - 14	0.7118	0.6871
14 - 16	0.6405	0.7005
16 - 18	0.7289	0.6706
18 - 20	0.8119	-

5.2 성능 평가용 데이터셋 구성

객체 인지 알고리즘의 성능을 평가하기 위해 Fig. 10에 제시된 지능형자동차부품진흥원 대구 주행 시험장에서 데이터를 취득하였다. Table 3은 전방 및 후방 데이터셋 구성을 나타내며, 이는 벙크 도로와 직선 도로에서 전후방 10 ~ 150 m 범위 내에 위치한 자율주행 차량 데이터를 포함한다. 데이터 취득 차량과 타겟 자율주행 차량은 각각 80 ~ 100 km/h의 속도로 주행하였으며, 모든 데이터의 상대 속도는 0 ~ 20 km/h 범위 내에 해당한다. 또한, 실제 주행 환경에서 발생할 수 있는 다양한 상황을 반영하기 위해 카메라의 가시 범위 내에 객체가 포함된 경우와 포함되지 않은 경우를 모두 고려하여 데이터를 취득하였다.



Fig. 10 KIAPI proving ground map

Table 3 Configuration of front and rear dataset

	Front	Rear
Frame	1446	1050
Instances	1930	1454
Object range (m)	10 ~ 150	10 ~ 150
Target velocity (km/h)	80 ~ 100	80 ~ 100
Relative velocity (km/h)	0 ~ 20	0 ~ 20

5.3 성능 평가 결과

본 논문에서 활용한 3D 객체 검출 알고리즘의 전방 및 후방 객체 탐지 범위가 상이하므로, 성능 평가는 전방과 후방으로 구분하여 수행하였다. 식 (6)과 식 (7)을 사용하여 2D, 3D, late fusion 기반 객체 검출 결과에 대한 precision과 recall을 계산하였으며, Table 4와 Table 5는 각각 전방 및 후방 객체 인지 성능 평가 결과를 나타낸다.

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \quad (6)$$

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} \quad (7)$$

Table 4 Evaluation of front object detection performance

	Precision	Recall
Camera (2D)	0.9798	0.9544
LiDAR (3D)	0.9714	0.6690
Late fusion	0.9619	0.9933

Table 5 Evaluation of rear object detection performance

	Precision	Recall
Camera (2D)	0.9468	0.9415
LiDAR (3D)	0.8684	0.2043
Late fusion	0.9149	0.9759

2D 객체 검출 결과는 전방 및 후방 영역 모두에서 높은 Precision과 Recall을 기록한 반면, 3D 객체 검출 결과는 상대적으로 낮은 성능을 기록하였다. 이는 최대 150 m 거리의 객체를 검출하는 조건에서 실험이 수행되었으

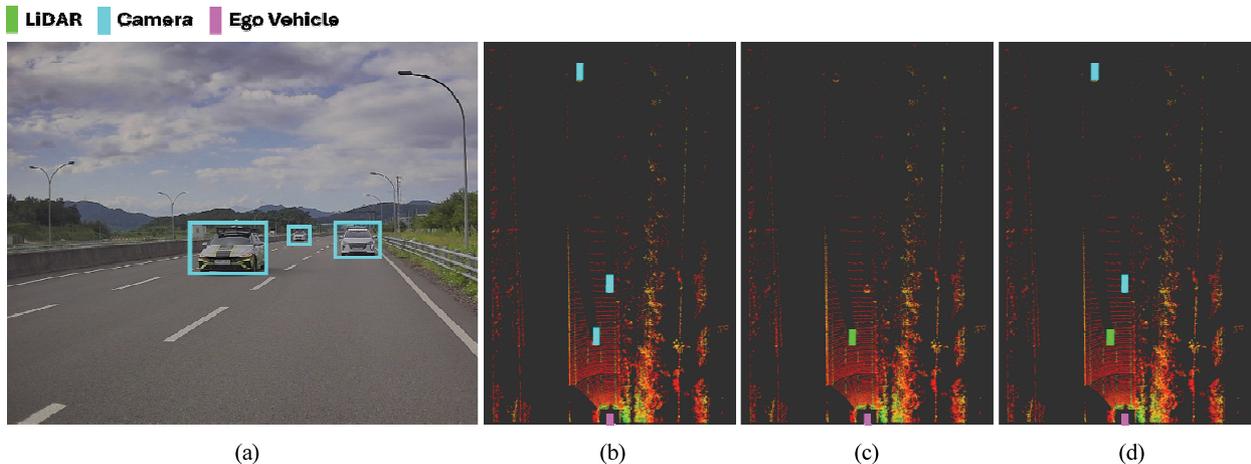


Fig. 11 Object detection results based on 2D, 3D, and late fusion

나, 해당 조건은 3D 객체 검출 알고리즘의 탐지 범위를 초과하여 객체 검출에 실패하였기 때문이다. 또한, 2D 및 3D 객체 검출 결과와 Late fusion 기반 객체 검출 결과를 비교한 결과, Precision이 감소한 것을 확인하였다. 이는 Late fusion 과정에서 각 알고리즘의 False positive가 합산되어, 전체 False positive가 증가하였기 때문이다. 반면 Recall은 증가하였는데, 이는 카메라의 가시 범위에 포함되지 않은 객체가 2D 객체 검출에서 실패할 경우 3D 객체 검출 결과로 보완되고, 3D 객체 검출에서 실패한 객체는 2D 객체 검출 결과로 보완되어 True positive가 증가하였기 때문이다. 해당 결과는 대부분의 객체를 성공적으로 검출했음을 나타내며, 100 km/h 이상의 고속 주행 환경에서 안정적으로 장거리 객체를 검출하여 충돌 회피가 가능함을 의미한다. 또한, 차량 측면에 카메라를 장착하여 가시 범위를 확장할 경우, 2D 객체 검출 및 Late fusion 기반 객체 검출의 성능이 전반적으로 향상될 수 있음을 나타낸다. Fig. 11은 후방 카메라와 라이다 센서의 Late fusion 과정을 나타낸다. Fig. 11(a)와 (b)는 각각 2D 객체 검출 결과를 2차원, 3차원 공간에서 시각화한 결과를 나타내며, Fig. 11(c)는 3D 객체 검출 결과를 3차원 공간에서 시각화한 결과를 나타낸다. 마지막으로 Fig. 11(d)는 Late fusion을 적용하여 중복으로 검출된 객체를 제거하고, 단일 센서로 검출된 객체들을 융합한 최종 객체 검출 결과를 나타낸다.

## 6. 결론

본 논문에서는 고속 레이싱 환경에서의 Camera-LiDAR 센서 퓨전 인지 알고리즘을 제안한다. 센서 데이터의 정합 정밀도를 향상시키기 위해 내부, 외부 및 Temporal 캘리브레이션을 수행하였다. 2D 객체 검출 알고리즘은 YOLOv5를 활용하여 전방 및 후방 150 m 범위 내의 객체를 30 Hz로 검출하였으며, 3D 객체 검출 알고리즘은 PointPillars를 활용하여 전방 128 m, 후방 25 m 범위 내의 객체를 10 Hz로 검출하였다. 최종 Late fusion 기반 객체 인지 알고리즘은 라이다 데이터 주기에 맞춰 10 Hz로 구현하였다. 객체의 중복 검출 여부는 유클리드 거리를 기반으로 판단하였으며, 3D 객체 검출 결과를 우선적으로 채택하였다.

제안하는 인지 알고리즘의 성능을 평가하기 위해 전방 및 후방 10 ~ 150 m 범위 내에서 상대 속도 0 ~ 20 km/h로 주행하는 자율주행 차량 데이터를 취득하였으며, 3D 객체 검출 알고리즘의 탐지 범위를 고려하여 전방과 후방으로 구분하여 성능 평가를 수행하였다. 성능 평가 결과 Late fusion을 적용하여 2D 및 3D 객체 검출 결과를 융합하였을 때 Precision은 감소하고 Recall이 증가하는 경

향을 보였으며, Recall이 증가함을 통해 본 논문에서 제안하는 인지 알고리즘의 합리성을 확인할 수 있다. 추후 Early fusion 또는 Deep fusion 방식을 활용한 객체 검출 연구를 수행할 예정이며, 도심 등 복잡한 환경에서의 실시간 객체 인지를 위해 인지 주기 개선 방안을 연구할 계획이다.

## 후 기

이 논문은 2024년도 정부(산업통상자원부)의 재원으로 한국산업기술기술평가원의 지원을 받아 수행된 연구임(RS-2024-00408818).

## References

- 1) S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, G. Hoffmann, K. Lau, C. Oakley, M. Palatucci, V. Pratt, P. Stang, S. Strohband, C. Dupont, L. E. Jendrossek, C. Koelen, C. Markey, C. Rummel, J. van Niekerk, E. Jensen, P. Alessandrini, G. Bradski, B. Davies, S. Ettinger, A. Kaehler, A. Nefian and P. M. Stanley, "The Robot That Won the DARPA Grand Challenge," *Journal of Field Robotics*, Vol.23, No.9, pp.661-692, 2006.
- 2) J. Betz, A. Wischnewski, A. Heilmeier, F. Nobis, T. Stahl and L. Hermansdorfer, "A Software Architecture for an Autonomous Racecar," 2019 IEEE 89th Vehicular Technology Conference, pp.1-7, 2019.
- 3) C. C. Chung, A. Finazzi, H. Seong, D. Lee, S. Lee, B. Kim, G. Gang and D. H. Shim, "Autonomous System for Head-to-Head Race: Design, Implementation and Analysis; Team KAIST at the Indy Autonomous Challenge," *IEEE Transactions on Field Robotics*, 2024.
- 4) J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," *IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, pp.779-788, 2016.
- 5) W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu and A. C. Berg, "SSD: Single Shot Multibox Detector," *European Conference on Computer Vision(ECCV)*, pp.21-37, 2016.
- 6) R. Girshick, "Fast R-CNN," *IEEE International Conference on Computer Vision(ICCV)*, pp.1440-1448, 2015.
- 7) S. Ren, K. He, R. Girshick and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Transactions on*

- Pattern Analysis and Machine Intelligence, Vol.39, No.6, pp.1137-1149, 2016.
- 8) A. Vaswani, "Attention Is All You Need," *Advances in Neural Information Processing Systems(NeurIPS)*, 2017.
  - 9) N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov and S. Zagoruyko, "End-to-End Object Detection with Transformers," *European Conference on Computer Vision(ECCV)*, pp.213-229, 2020.
  - 10) Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin and B. Guo, "Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows," *IEEE/CVF International Conference on Computer Vision(ICCV)*, pp.10012-10022, 2021.
  - 11) Z. Yang, Y. Sun, S. Liu and J. Jia, "3DSSD: Point-Based 3D Single Stage Object Detector," *IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR)*, pp.11040-11048, 2020.
  - 12) W. Shi and R. Rajkumar, "Point-GNN: Graph Neural Network for 3D Object Detection in a Point Cloud," *IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR)*, pp.1711-1719, 2020.
  - 13) Y. Zhou and O. Tuzel, "VoxelNet: End-to-End Learning for Point Cloud-Based 3D Object Detection," *IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, pp.4490-4499, 2018.
  - 14) A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang and O. Beijbom, "PointPillars: Fast Encoders for Object Detection from Point Clouds," *IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR)*, pp.12697-12705, 2019.
  - 15) H. Wu, C. Wen, S. Shi, X. Li and C. Wang, "Virtual Sparse Convolution for Multimodal 3D Object Detection," *IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR)*, pp.21653-21662, 2023.
  - 16) S. Vora, A. H. Lang, B. Helou and O. Beijbom, "PointPainting: Sequential Fusion for 3D Object Detection," *IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR)*, pp.4604-4612, 2020.
  - 17) C. Wang, C. Ma, M. Zhu and X. Yang, "PointAugmenting: Cross-Modal Augmentation for 3D Object Detection," *IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR)*, pp.11794-11803, 2021.
  - 18) Z. Liu, H. Tang, A. Amini, X. Yang, H. Mao, D. L. Rus and S. Han, "BEVFusion: Multi-Task Multi-Sensor Fusion with Unified Bird's-Eye View Representation," *IEEE International Conference on Robotics and Automation(ICRA)*, pp.2774-2781, 2023.
  - 19) S. Pang, D. Morris and H. Radha, "CLOCs: Camera-LiDAR Object Candidates Fusion for 3D Object Detection," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp.10386-10393, 2020.
  - 20) A. E. Conrady, "Decentred Lens-Systems," *Monthly Notices of the Royal Astronomical Society*, Vol.79, No.5, pp.384-390, 1919.
  - 21) D. Lee and S. C. Kee, "Efficient Camera-LiDAR Calibration Using Accumulated LiDAR Frames," *IEEE Access*, Vol.10, pp.132349-132362, 2022.
  - 22) Y. Wu and Z. Hu, "PnP Problem Revisited," *Journal of Mathematical Imaging and Vision*, Vol.24, pp.131-141, 2006.
  - 23) ROS, "message\_filters/ApproximateTime," Retrieved from [https://wiki.ros.org/message\\_filters/ApproximateTime](https://wiki.ros.org/message_filters/ApproximateTime), 2024.
  - 24) Chrony Project, "Chrony," Retrieved from <https://chrony-project.org/>, 2024.
  - 25) Ultralytics, "YOLOv5," Retrieved from <https://github.com/ultralytics/yolov5>, 2024.
  - 26) C. Y. Wang, H. Y. M. Liao, Y. H. Wu, P. Y. Chen, J. W. Hsieh and I. H. Yeh, "CSPNet: A New Backbone That Can Enhance Learning Capability of CNN," *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp.390-391, 2020.
  - 27) F. Yu, H. Chen, X. Wang, W. Xian, Y. Chen, F. Liu, V. Madhavan and T. Darrell, "BDD100K: A Diverse Driving Dataset for Heterogeneous Multitask Learning," *IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR)*, pp.2633-2642, 2020.
  - 28) NVIDIA, "TensorRT," Retrieved from <https://developer.nvidia.com/tensorrt>, 2024.
  - 29) H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan and O. Beijbom, "NuScenes: A Multimodal Dataset for Autonomous Driving," *IEEE/CVF Conference on Computer Vision and Pattern Recognition(CVPR)*, pp.11621-11631, 2020.
  - 30) G. Ahn, Y. H. Choi and S. C. Kee, "Research on Optimization and Improving High-Speed YOLO-Based Vehicle Detection Performance," *KSAE Spring Conference Proceedings*, pp.548-552, 2024.
  - 31) E. J. Bong, D. Lee and S. C. Kee, "Wireless

Charging Pad Detection for Autonomous Charging of Electric Vehicles Based on AVM Image,” Transactions of KSAE, Vol.31, No.9, pp.727-736, 2023.