

## ROI 기반 카메라-라이다 융합 위치정보 추출 알고리즘

윤 영 진\*, 한 동 석<sup>o</sup>ROI-based Calibration Algorithm of Camera-Lidar Sensor Fusion  
That is Strong in Object Recognition and Distance Information  
ExtractionYoung Jin Yoon\*, Dong Seog Han<sup>o</sup>

## 요 약

자율주행 자동차에서 도로상 객체와의 정확한 위치인지는 주행 안전에 매우 중요한 요소이다. 본 논문에서는 객체의 정확한 위치 추출을 위한 카메라와 라이다를 융합한 ROI(region of interest) 기반의 위치정보 추출 알고리즘을 제안한다. 일반적으로 카메라와 라이다를 융합하여 객체의 위치 정합은 체스보드 등의 타겟보드를 사용하여 센서 간 위치 차이를 보정한다. 자율주행 자동차의 객체 인식을 위한 관심 영역은 넓은 도로 환경으로 인하여 객체의 위치에 따라 카메라와 라이다의 위치 차이가 다르게 발생하게 된다. 제안하는 알고리즘은 카메라 영상을 다수의 ROI로 나누어 영역별 독립적으로 카메라와 라이다의 위치 정합을 수행한다. 제안하는 ROI 기반 위치 정합 알고리즘의 정확도 성능을 가상 및 실제 도로 환경에서 검증한다.

**키워드** : 자율주행, 객체 인식, 센서융합

**Key Words** : Autonomous driving, Object detection, Calibration, Sensor Fusion

## ABSTRACT

In an autonomous vehicle, accurate location recognition with an object on the road is a very important factor for driving safety. In this paper, we propose an region of interest (ROI)-based location information extraction algorithm that converges a camera and a lidar for accurate location extraction of an object. In general, when the camera and lidar are fused to match the position of an object, a target board such as a chess board is used to calibrate the position difference between the sensors. In the area of interest for object recognition of an autonomous vehicle, a difference in the positions of the camera and lidar occurs depending on the position of the object due to the wide road environment. The proposed algorithm divides the camera image into multiple ROIs and performs the position matching of the camera and lidar independently for each area. We verify the accuracy of the proposed ROI-based positioning algorithm in virtual and real road environment.

※ “본 연구는 산업통상자원부와 한국산업기술진흥원이 지원하는 5G기반 자율주행 융합기술 실증 플랫폼 과제(과제고유번호 : 1415169669)의 지원을 받아 수행하였습니다.”

• First Author : Kyungpook National University, School of Electronics Engineering, skag2603@knu.ac.kr, 학생회원

◦ Corresponding Author : Kyungpook National University, School of Electronics Engineering, dshan@knu.ac.kr, 종신회원

논문번호 : 202108-195-C-RU, Received July 30, 2021; Revised September 3, 2021; Accepted September 7, 2021

## I. 서론

첨단운전자 보조 시스템(ADAS, advanced driver assistance system)은 다양한 센서와 GPS, 통신 등을 이용하여 주행 상황을 인지 및 판단한다<sup>1)</sup>. 그리고 자동차를 상황에 대응하여 제어하거나 운전자가 미리 위험 요소를 감지할 수 있도록 알려준다. 카메라와 라이다는 객체 검출과 위치정보 추출을 위한 ADAS의 핵심 센서이다. 카메라와 라이다의 설치 위치 차이에 의한 객체의 위치정보가 다르게 나타나며 이것을 정합시킬 필요가 있다. 이를 위하여 카메라와 라이다를 융합한 두 센서 간의 객체 위치의 정합을 위한 좌표 변환이 필요하다<sup>2,3)</sup>. 즉, 객체 위치의 정합은 서로 다른 센서에서 구한 위치정보를 센서 간 보정 계수를 적용하여 일치시키는 과정이다. 이것은 여러 센서가 동일 객체의 위치를 동일하게 인식하기 위해 필수적인 작업이다.

본 논문에서는 ROI 기반의 카메라와 라이다를 이용한 객체의 위치정합 알고리즘을 제안한다. 일반적으로 카메라와 라이다의 위치 정합을 위하여 체스보드<sup>4)</sup>, 다이아몬드와 같은 다각형 보드 등의 다양한 타겟 보드<sup>5)</sup>를 이용 위치정합을 수행한다. 그러나 타겟 보드를 사용한 카메라와 라이다의 위치 정합 알고리즘은 라이다의 해상도 및 환경에 따라 타겟보드의 모서리 검출이 정확하지 않다. 따라서 두 센서 간 좌표 변환에 오차가 발생한다<sup>6)</sup>. 그리고 영상의 모든 영역에서 카메라 포인트와 라이다 포인트의 정합이 정확하지 않다.

본 논문에서는 영상을 일정한 비율로 영역을 나누어서 영역별 센서 간 위치 오차를 보정하는 ROI(region of interest) 기반의 위치 정합 알고리즘을 제안한다.

본 논문의 구성은 다음과 같다. 2장에서는 카메라와 라이다 간 위치 정합에 관한 선행 연구를 소개한다. 3장에서는 자율주행에서 센서 간 동일 객체의 정확한 위치 정합을 할 수 있는 ROI 기반 위치 정합 알고리즘을 제안한다. 또한, ROI 기반 위치 정합과 YOLO v4를 융합함으로써 객체 인식 및 정확한 위치 정보 추출 시스템을 제안한다. 4장에서는 제안 알고리즘의 성능을 가상 및 실 도로환경의 실험을 통하여 검증하고 5장에서 결론을 맺는다.

## II. 선행 연구

카메라와 라이다의 위치 보정 알고리즘에는 여러

가지가 있다. 대표적 방법으로는 사각형의 체스보드를 이용한 위치 정합<sup>4)</sup>, 다각형 보드를 이용한 위치 정합<sup>5)</sup>, 삼면체를 이용한 위치 정합<sup>7)</sup> 등과 같이 타겟보드를 사용한 캘리브레이션 방법과 표적이 없는 위치 정합<sup>8)</sup> 알고리즘, 마지막으로 딥러닝을 이용한 위치보정 알고리즘<sup>9)</sup>이 있다. 체스보드를 이용한 위치 정합 알고리즘은 카메라와 라이다를 차량에 고정된 후 그림 1과 같이 차량 전면에 체스보드를 설치하여 수행한다. 카메라 센서를 통해 들어온 프레임으로부터 체스보드의 패턴 및 가장자리를 인식한다<sup>9)</sup>. 그리고 라이다 센서를 이용하여 체스보드를 인식하고 카메라 픽셀과 라이다 포인트를 매칭 시킨다. 그리고 각 센서의 매칭 포인트를 통해 카메라의 외부행렬을 계산한다. 외부행렬은 카메라 좌표계와 월드 좌표계 사이의 변환 관계로 회전변환 행렬과 이동변환 행렬이 있다. 이와 같은 위치 정합 알고리즘에는 다음과 같은 단점이 있다. 채널의 개수가 적은 라이다를 사용할 때, 체스보드의 가장자리를 정확하게 탐지하기가 어려워 센서 간 좌표 변환을 위한 외부행렬을 정확하게 구할 수 없다<sup>6)</sup>. 우수한 자율주행 성능을 확보하기 위하여 채널의 개수와 상관없이 정확한 외부행렬을 구하는 것이 매우 중요하다.

또 다른 대표적 방법으로 다각형을 이용한 위치 정합은 다이아몬드 모양의 보드나 삼각형의 평평한 보드를 이용하여 진행하는 것이다<sup>5)</sup>. 이러한 방법은 한 이미지에서 최소 3개의 타겟보드를 사용한다. 이는 여러 개의 매칭 포인트가 존재하므로 많은 계산량이 요구된다.

삼면체 모양을 사용한 보정 알고리즘은 삼면체 보정 보드<sup>7)</sup>를 사용하여 각 센서 간의 보정을 한다. 삼면체를 이용하여 매칭 포인트를 구할 때 최소 2개의 영



그림 1. 체스보드를 이용한 위치 정합 실험  
Fig. 1. Calibration experiment using chess board.

상 프레임이 필요하므로 처리할 데이터가 체스보드를 사용할 때보다 현저하게 많다. 그리고 삼면체 모양의 타겟물체를 제작하기가 매우 어렵다. 또한, 삼면체 내 매칭 포인트의 수동분리 및 삼면체의 평면을 수동으로 해야 하므로 전처리과정이 복잡하다.

표적이 없는 보정 알고리즘<sup>8)</sup>은 보정 타겟이 없는 상태로 위치 정합을 하므로 정합 정확도가 떨어진다<sup>16)</sup>. 따라서 높은 위치 정확도가 필요한 자율주행 환경에는 적합하지 않다.

딥러닝을 이용한 위치보정 알고리즘<sup>11)</sup>은 카메라 이미지와 라이다 포인트들을 이미지로 투영한 깊이 맵(depth map)의 차이를 학습함으로써 센서 간의 변환 계수를 구한다. 이러한 방법은 학습의 차이에 따라 결과가 달라지므로 자율주행 환경에 적합하지 않다.

### III. 제안하는 ROI 기반 위치 정합 알고리즘

#### 3.1 센서 위치 정합의 정의

모든 센서는 각자의 장착 위치와 센서가 바라보는 방향에 따라 서로 다른 좌표계를 가진다. 그러므로 하나의 좌표계로 통일하여 활용하기 위해서는 서로 다른 좌표계 상의 데이터들을 하나의 기준 좌표계로 변환해주는 센서 간 위치 정합 과정이 필요하다<sup>11)</sup>. 즉, 카메라 센서와 라이다 센서가 그림 2와 같이 설치되어 있을 때, 한 센서를 기준으로 나머지 센서의 평행 이동과 회전의 정도를 나타내는 외부행렬을 구하는 것이다. 카메라 센서와 라이다 센서의 위치 정합이 정확하게 되어야만 두 센서의 좌표 매칭이 가능하다.

체스보드를 이용한 위치 정합 알고리즘의 관계식은 다음과 같다<sup>12)</sup>.

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & x_0 \\ 0 & f_y & y_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (1)$$

여기서  $X, Y, Z$ 는 라이다 센서로부터 도출된 포인트 클라우드 좌표이며  $x, y$ 는 라이다 포인트들을 투영(projection)한 카메라 좌표이다. 포인트 클라우드 좌표를 회전 행렬인  $r_{11} - r_{33}$  과 평행이동 행렬  $t_1 - t_3$  간의 행렬 곱을 통하여  $3 \times 1$ 의 행렬식으로 도출된다. 그 이후에는 내부행렬과 곱을 취한다<sup>13)</sup>. 내부행렬이란 카메라 내부 파라미터값으로서 카메라 내부의 초점거리, 주점, 비대칭 계수를 의미한다.  $f_x, f_y$ 는 카메라의 초점거리(focal length)이다. 초점거리란 렌즈 중

심과 이미지 센서와의 거리를 의미한다.  $x_0$ 와  $y_0$ 는 카메라 센서로부터 들어온 프레임의 중심점이다. 이러한 수식으로 인해서 라이다 포인트는 이미지 포인트로 투영된다<sup>14)</sup>. 내부행렬과 외부행렬을 하나의 변수로 정의하여 다음과 같이 표현할 수 있다.

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (2)$$

여기서  $m$ 은 센서 간 변환 계수로 카메라 내부행렬과 외부행렬의 곱이다. 하나의 변수로 정의한 이유는 라이다의 포인트 클라우드와 카메라 센서의 픽셀 데이터의 관계를 나타내기 위한 것이다. 식 (2)를 통해 각 라이다 좌표에 대응하는 카메라 좌표 한 쌍에 대해 다음과 같은 두 개의 방정식을 얻을 수 있다. 식 (3)을 통하여 변환 계수를 구하기 위해서는 다양한 매칭 포인트들이 필요하다.

$$\begin{aligned} x &= \frac{m_{11}X + m_{12}Y + m_{13}Z + m_{14}}{m_{31}X + m_{32}Y + m_{33}Z + m_{34}} \\ y &= \frac{m_{21}X + m_{22}Y + m_{23}Z + m_{24}}{m_{31}X + m_{32}Y + m_{33}Z + m_{34}} \end{aligned} \quad (3)$$

그림 3은 체스보드를 이용한 카메라와 라이다 센서 위치 정합의 결과이다. 이것은 위치 정합을 통해 추출한 외부행렬을 구하여 빨간색 점으로 표시된 라이다 포인트를 이미지 포인트 즉 픽셀에 위에 표시한 것이다. 그림 3에서 이미지 왼쪽 위의 건물 가장자리 부분이 라이다와 카메라 사이의 위치 정합이 되지 않음을 알 수 있다. 이것은 이미지의 중앙에서 멀어질수록 외부행렬이 정확하지 않음을 의미한다. 이러한 현상은 자율주행 환경에서 객체의 위치를 정확히 검출하지 못하여 매우 위험한 상황을 초래할 수 있다. 본 논문에서는 이 문제를 해결할 수 있는 프레임 전 영역에

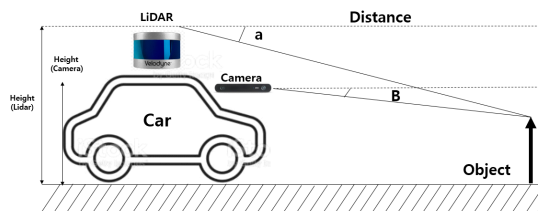


그림 2. 센서 간의 상대적인 위치 수치  
Fig. 2. Relative position figures between sensor.



그림 3. 체스보드를 이용한 위치 정합 알고리즘 결과  
Fig. 3. Calibration result using chess board.

정확도가 높은 새로운 위치 정합 방법을 제안한다.

### 3.2 ROI 위치 보정 알고리즘

ROI 위치 정합 알고리즘에서는 카메라 영상을 적절한 비율의 ROI로 나누어 라이다 센서와 위치 정합을 진행한다. ROI 기반 위치보정 알고리즘은 기존의 방법과 같이 체스보드를 이용하여 외부행렬을 구하여 위치 정합을 진행한다. 기존의 방법은 한 이미지에 대한 외부행렬을 구하지만, 제안하는 ROI 위치 정합 알고리즘은 각각 나누어진 영역별 외부행렬을 구한다. 기존의 방법은 그림 3과같이 정확한 위치 정합이 되지 않지만 제안하는 ROI 위치보정 알고리즘은 모든 영역에서 정확한 위치 정합을 진행할 수 있다. 또한, 영역을 나눔으로써 장애물이 어느 영역에 위치하는지를 정확하게 파악할 수 있다. 이러한 환경은 자율주행 차량을 사용하는 ROS(robot operating system)에도 매우 적절하다. 왜냐하면, ROS 환경에서는 각 노드에 장애물의 위치를 신속하게 정보를 보낼 수 있기 때문이다<sup>15)</sup>. 각 영역에서 장애물이 발견된다면 장애물의 위치가 필요한 차량 내부 노드에 정보를 바로 보낼 수 있다.

그림 4는 카메라 센서로부터 들어오는 이미지를 2×3의 총 6개 ROI로 나눈 이미지이다. 이미지의 영역 1과 4는 왼쪽, 영역 2와 5는 중앙, 영역 3과 6은 오른쪽 영역이다. 각 영역의 너비는 동일하다. 영역 1, 2, 3과 4, 5, 6을 나눈 이유는 각각의 영역에서 인식해야 하는 객체가 다르기 때문이다. 영역 4, 5, 6은 주로 차량 주변으로 장애물 즉 사람, 자동차, 오토바이 등 차량 환경에서 접촉이 가능한 장애물로 판단되지만, 영역 1, 2, 3에서는 주로 신호등 또는 드론 등 비교적 지면에서 떨어져 있는 물체들을 분류한다<sup>16)</sup>. 그리고 언덕과 내리막길과 같은 비정형 도로에서도 적용할 수 있다. 이렇게 나누어진 ROI의 영역에서 각각 체스보드를 이용하여 위치 정합을 진행한다.

두 센서 간의 위치 정합 과정을 그림 5에 나타내었

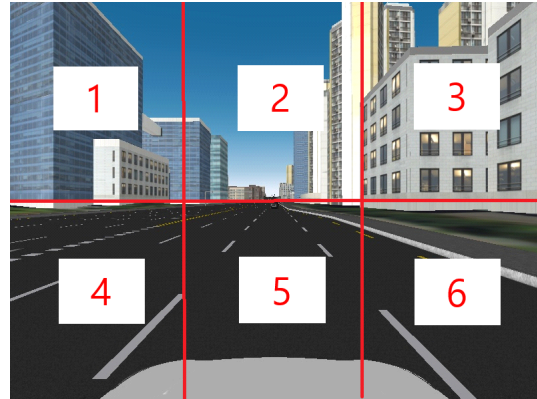


그림 4. 위치 정합 계수를 위한 ROI 분할  
Fig. 4. ROI segmentation for calibration coefficients.

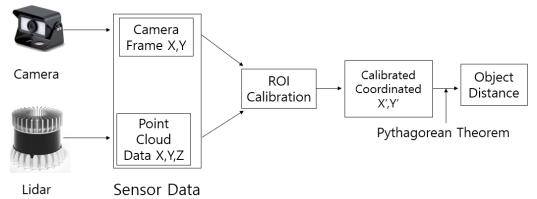


그림 5. 카메라-라이다를 이용한 ROI 기반 위치정합 블록도  
Fig. 5. ROI-based positioning block diagram using camera-lidar.

다. 먼저 카메라와 라이다로부터 데이터가 추출된다. 카메라 센서로부터는 이미지에 대한 데이터는 픽셀 데이터가 추출되며 라이다 센서로부터는 포인트 클라우드 데이터가 추출된다. 데이터를 추출할 때는 센서 간의 주기를 동일하게 하여야 한다. 따라서 카메라 센서로부터 추출되는 프레임의 주기와 라이다 센서로부터 추출되는 데이터의 주기는 달라서 전처리 작업으로 주기를 동기화시킬 필요가 있다.

각 센서로부터 추출된 데이터를 이용하여 ROI 위치 정합 알고리즘을 진행한다. 기존의 방법은 한 이미지에 대하여 여러 개의 매칭 포인트들을 구하지만, ROI 위치 정합 알고리즘은 각 ROI 영역에서 매칭 포인트를 구한다. 각각 나누어진 영역에서는 1000개의 매칭 포인트 세트를 이용하여 영역마다 외부행렬을 추출한다. 구하는 과정은 그림 1과 같이 실제 차량 환경에서 체스보드를 이용하여 영역마다 매칭 포인트를 구하여 위치 정합을 진행한다. 기존의 체스보드를 이용한 위치 정합 알고리즘은 한 프레임에 한 개의 쌍의 매칭 포인트를 구한다. 그러나 제안 알고리즘은 영역을 나눈 상태에서 여러 개의 매칭 포인트를 구하기 때문에 기존 방법보다 정확도가 높다. 그리고 영역마다 얻어진 변환 계수를 이용하여 라이다 포인트를 이미



지 포인트로 투영한다. 이후에는 객체 인식알고리즘 YOLO v4를 이용하여 차량 전방의 객체를 인식한다. 인식한 객체의 중심좌표는 이미지 픽셀로 2D 좌표이다. 이러한 좌표를 사전에 구한 변환 계수를 이용하여 역변환한 좌표를 피타고라스의 정리를 적용하여 객체의 거리를 추출한다.

### 3.3 YOLO v4

YOLO는 입력값으로 이미지가 주어지면 배경과 사물을 구분하고 어떤 사물인지 분류한다<sup>17)</sup>. YOLO가 등장하기 이전에도 딥러닝 모델을 이용하여 객체 탐지를 수행하는 방법은 있었다. 대표적으로 R-CNN<sup>18)</sup>이 존재하는데 YOLO는 기존 모델들보다 더 높은 정확도를 추구하는 것이 아닌, 근접한 정확도를 가지면서 더 많은 양의 이미지를 처리할 수 있는 실시간 객체 탐지를 수행한다. 자율주행 환경에서는 차량 주변을 실시간으로 객체 탐지를 해야 하므로 YOLO를 사용한다. YOLO 알고리즘 v1 버전은 실시간 객체 탐지를 위하여 만들어졌다<sup>17)</sup>. 그러나 버전이 향상되면서 정확도 또한 많이 향상되었다. 본 논문에서는 YOLO의 최신 알고리즘인 YOLO v4를 사용하였다<sup>19)</sup>. YOLO v4의 기본적인 목표는 시스템 내에서 병렬계산을 위한 최적화와 신경망의 빠른 작동속도를 가지게 하는 것이다. YOLO v4의 아키텍처는 다음과 같다. YOLO v4의 백본(backbone)은 CSP-Darknet 53을 사용하였고 Neck은 SPP(Spatial Pyramid Pooling), PAN(Path Aggregation Network)를 사용하였고 Head는 YOLO v3의 구조를 사용하였다<sup>20)</sup>.

그림 6과 같이 YOLO v4의 구조는 다음과 같다. 백본은 입력 이미지를 피쳐맵(feature map)으로 변형시켜주는 부분이며 헤드는 백본에서 추출한 피쳐맵의 위치 작업을 수행하는 부분이다. 이러한 아키텍처를 이용하여 Imagenet 데이터 세트를 분류 테스트<sup>21)</sup> 및 MS COCO 데이터 세트를 객체 탐지 성능 테스트를 진행하였을 때 현존하는 알고리즘보다 성능이 뛰어나

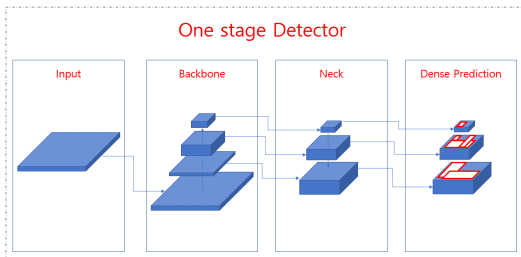


그림 6. YOLO v4의 아키텍처  
Fig. 6. Architecture of YOLO v4.

다<sup>22)</sup>.

그림 7은 가상환경에서 YOLO v4 알고리즘을 적용하여 실험한 결과이다. 그림과 같이 YOLO v4는 차량 전방에 있는 물체들을 차량으로 인식하고 객체 인식정확도가 97%, 100%인 것을 알 수 있다. 앞에서 서술한 ROI 보정 알고리즘을 적용해볼 때 바운딩 박스의 중심점은 5, 6번 영역 안에 있는 것을 알 수 있다. 바운딩 박스란 YOLO v4를 적용하여 3D 객체의 형태를 모두 포함할 수 있는 최소 크기의 박스를 말한다. 인식된 전방 객체의 좌표는 YOLO v4 결괏값인 바운딩 박스 좌표의 왼쪽 상단과 오른쪽 하단 좌표를 제공한다. 2개의 좌표를 각각 더해서 이등분하면 바운딩 박스의 중심점이 도출된다. 바운딩 박스의 중심점은 ROI 보정 알고리즘에 적용할 때 5, 6번 영역에 존재하므로 5, 6번 영역의 내부 및 외부행렬을 역변환하여 포인트 클라우드 데이터로 변환한다.

변환한 포인트 클라우드 데이터는  $x, y, z$ 의 형태로 존재한다. 변환한 데이터는 자율주행 차량 전방 장애물의 거리를 추출할 수 있다. 거리는 좌표  $x, y, z$ 를 각각 제곱하여 제곱근을 취하면 자율주행 차량과 전방 장애물과의 거리가 추출된다.

그림 8은 실제 환경에서 YOLO v4를 적용한 결과이다. 본 실험에서 카메라는 ZED 카메라를 사용하였다. 그림 8에서 ROI 기반 위치 정합 알고리즘을 적용하면 왼쪽 차량은 영역 4, 사람은 영역 6에 포함된다.

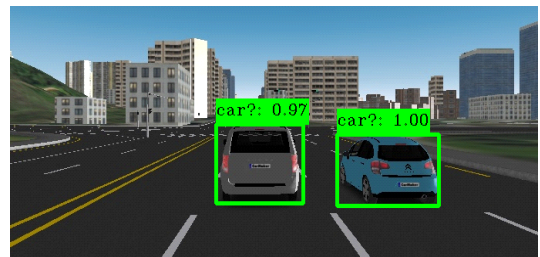


그림 7. 가상환경에서 YOLO v4를 이용한 객체 인식  
Fig. 7. Object recognition using YOLO v4 in virtual environment.



그림 8. 실제 환경에서 적용한 YOLO V4의 결과  
Fig. 8. Results of YOLO v4 applied in real environment.

각 영역에서의 회전행렬과 변환행렬을 이용하여 각각 포인트 클라우드 데이터의 형태로 변환한다. 변환한 포인트 클라우드 데이터의 형태는 가상환경과 동일하게  $x, y, z$  좌표의 형태로 존재하며 이를 이용하여 전방의 객체 거리를 추출할 수 있다.

#### IV. 실험 환경 및 결과

##### 4.1 실험 환경

제안하는 알고리즘에 사용되는 센서의 종류는 카메라 센서와 라이다 센서이다. 카메라 센서는 객체의 2D 정보를 알 수 있으므로 차량 주변 사물을 인지하기 위해서 사용된다. 반면에 라이다 센서는 카메라 센서와는 달리 주변 객체의 3D 정보를 알 수 있으므로 차량 주변 객체의 거리를 추출하는 데 사용된다.

카메라 센서는 STEREO LABS 사의 ZED 스테레오 카메라를 사용하였다. ZED 카메라는 120°의 FOV(field of view)를 가지고 있으며 1920×1080 크기의 해상도를 가지고 있다. ZED 카메라는 최대 100FPS(frame per second)의 속도를 가지고 있다. 카메라 센서는 그림 9와 같이 부착하였다. 전방의 카메라를 그림 9의 빨간색 사각형 부분에 고정한 이유는 카메라가 차량 보닛을 최소한으로 보이게 하고 전방의 장애물을 더욱더 넓게 보기 위함이다.

라이다 센서는 벨로다인 사의 VLP-16 모델을 사용하였다. VLP-16은 100m의 넓은 측정 범위와 최대 3cm의 오차를 가지고 있으며 시야는 수직으로는 위아래 15° 수평으로는 360°의 범위를 가지고 있다. 또한, 회전속도는 5Hz~20Hz를 가지는 센서이다. 16채널 라이다를 선택한 이유는 32 ch 64 ch보다 데이터 처리를 빠르게 할 수 있기 때문이다.

라이다 센서는 포인트 클라우드의 형태를 가진 점들로 이루어져 있다. 라이다 센서는 레이저를 목표물에 비추므로써 사물의 거리, 방향, 속도, 온도, 물질 분



그림 9. 차량 내부의 카메라 센서의 위치  
Fig. 9. Position of the camera sensor inside the vehicle.

포 및 농도 특성 등을 감지할 수 있는 기술이다. 그중에서 본 논문에서는 사물의 거리, 속도를 이용한다.

라이다 센서는 그림 10의 빨간색 사각형 라인 부분과 같이 차량의 루프에 부착하였다. 이렇게 부착한 이유는 라이다는 수평으로 360°의 범위를 인지할 수 있기 때문이다. 그러나 수직으로는 15°의 범위를 인지할 수 있으므로 최대한 라이다 센서의 위치를 차량과 가까이 설치한다.

가상환경은 구현된 알고리즘을 검증하기 위해 IPG Automotive 사의 Carmaker 프로그램을 사용하였다. Carmaker를 이용하여 도로 주행환경에서 제안 알고리즘의 성능을 평가할 수 있다<sup>23)</sup>. 대표적으로 도로, 자동차, 사람, 센서 등이 실제 환경과 유사하게 구현



그림 10. 차량 위 라이다 센서의 위치  
Fig. 10. Position of the lidar sensor on the vehicle.

표 1. 라이다 센서를 이용한 포인트 클라우드 데이터 좌표  
Table 1. Point cloud data coordinates using lidar sensor.

Position	$x$	$y$	$z$
Point 53552	3.9391	-0.1934	-0.0822
Intensity	4074.64		

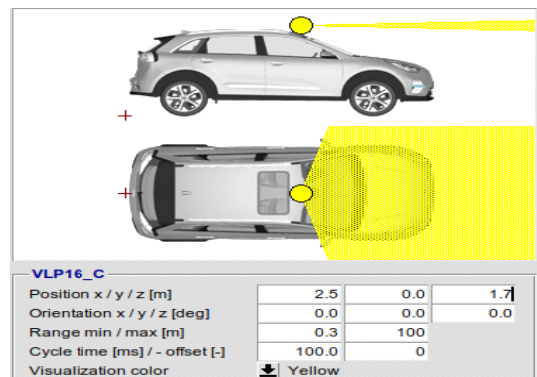


그림 11. 가상환경에서의 라이다 센서의 위치  
Fig. 11. Position of the lidar sensor in virtual.

되었으며 그림 11과 같이 센서 위치 및 데이터 취득 주기를 실제 시스템과 동일하게 구현하였다.

### 4.2 결과

실험은 가상환경 및 실제 환경에서 검증한다. 가상 환경에서는 실제 환경과 똑같이 환경을 구성하였다. 가상환경을 이용하는 이유는 제안하는 알고리즘의 성능을 사전에 검증하기 위함이다. 그림 12는 자율주행 차량 전방에 장애물이 있는 상황이다. 장애물의 실제 거리를 구하기 위해서는 라이다 포인트의 좌표를 알아야 한다. 라이다 포인트의 정확한 좌표를 얻기 위해서는 3D 시각화 도구인 RVIZ 소프트웨어를 이용한다. 그림 12에서 빨간색 줄은 라이다 포인트를 연속적으로 표현하였기 때문에 선처럼 보이지만 실제로는 점으로 이루어져 있다. 표 1은 그림 12의 자율주행 차량 전방에 장애물의 위치 좌표이다. 포인트 53552는 전방 차량의 라이다 포인트 한 개를 의미한다. 라이다 포인트는  $x, y, z$ 는 각각 3.9391, -0.19348, -0.082291이다. 포인트 좌표를 이용하여 차량과 장애물 사이의 거리를 피타고라스의 정리를 이용하여 구할 수 있다. 정리를 이용하여 추출한 거리는 3.944m이다. 이 거리는 실제 자율주행 차량으로부터 장애물까지의 거리이다.

그림 12와 같은 환경을 구성한 제안 ROI 기반 위치 정합 알고리즘의 결과 값은 다음 그림 13과 같다. 전방 장애물의 거리는 ROI 캘리브레이션을 통해 4.1815m가 추출됐다. 실제 거리와 측정된 거리의 오차는 약 6% 나는 것을 확인할 수 있다. 표 2는 실험을 여러 번 진행하여 각 영역의 오차이다.

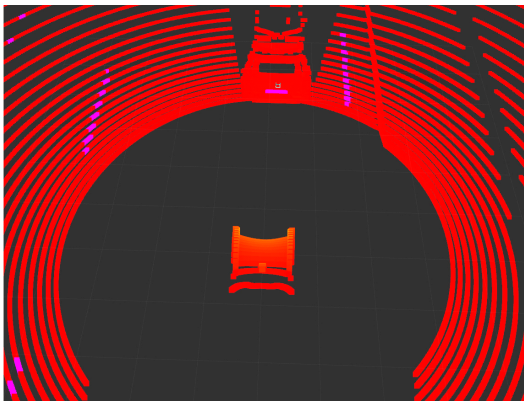


그림 12. 가상환경에서 실험한 자율주행 환경  
Fig. 12. Autonomous driving environment tested in virtual environment.

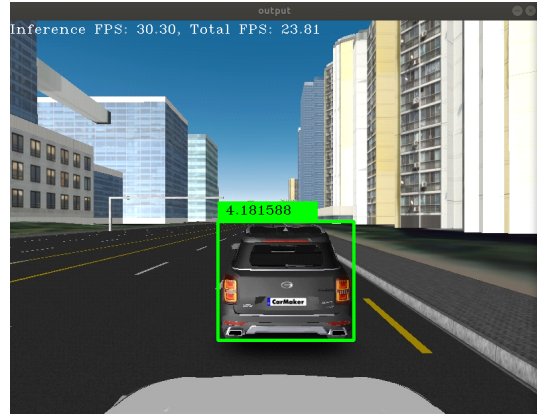


그림 13. ROI 캘리브레이션을 이용한 장애물의 거리 결과  
Fig. 13. Obstacle Distance Results Using ROI Calibration.

표 2. 가상환경에서 측정된 전방 차량 거리와 제안하는 알고리즘을 이용한 거리 측정 비교  
Table 2. Comparison of distance measurement using the proposed algorithm with the distance of the vehicle in front measured in a virtual environment.

Area	4	5	6
Lidar Distance (m)	15.3m	3.94m	19.9
Calibration Distance (m)	14.9m	4.18m	20.8
Error (%)	2.6%	6%	4.5%

표 2에서는 제안하는 방법으로 추출된 전방 차량의 거리와 실제 차량의 거리를 비교한 표이다. Lidar Distance는 실제 라이다 센서를 이용하여 측정된 차량 거리이며 Calibration Distance는 제안하는 방법으로 측정된 전방 차량의 거리이다. 표 2에서 영역 4, 5, 6 번의 오차를 측정하였을 때 평균 5% 미만인 것을 확인하였다. 실제 환경에서도 그림 14와 같이 ROI 기반 위치 정합 알고리즘을 이용하여 차량 전방 객체의 거리를 추출하였다. 전방 객체의 거리는 각각 8.9m, 9.5m이다. 전방 객체는 각각 4, 5영역에 있으므로 각 영역의 회전, 이동 행렬을 이용하여 객체 거리를 추출하였다.

실제 환경에서도 그림 14와 같이 ROI 기반 위치 정합 알고리즘을 이용하여 차량 전방 객체의 거리를 추출하였다. 전방 객체의 거리는 각각 8.9m, 9.5m이다. 전방 객체는 각각 4, 5영역에 있으므로 각 영역의 회전, 이동 행렬을 이용하여 객체 거리를 추출하였다.

표 3은 표 2와 같이 제안하는 방법으로 추출한 객체의 거리와 실제 거리와의 비교표이다. 4번 영역에서 제안하는 알고리즘의 결과 값은 8.93m이며 실제 측정



거리는 9.12m이다. 4번 영역에서의 오차는 1.9% 5번 영역에서의 오차는 2.5% 6번 영역에서는 1.8%로 정확한 거리가 추출되었다는 것을 알 수 있다.

표 4는 표 2와 같이 제안하는 방법으로 추출한 거리와 기존의 체스 보드를 사용한 위치 정합 알고리즘 성능 비교표이다. 영역 5는 제안하는 알고리즘의 성능과 기존의 성능의 차이가 0.16m로 근접한 수치를 보여준다. 하지만 기존의 방법과 제안하는 방법의 차이는 영역 4, 6에서 차이가 크게 발생한다. 제안하는 알고리즘은 기존의 알고리즘보다 이미지 가장자리에서 성능이 우수하다는 것을 알 수 있다.



그림 14. ROI기반 위치보정을 실제 환경에서 적용한 결과  
Fig. 14. Result of using ROI-based positioning in real environment.

표 3. 실제 환경에서 전방 차량 거리와 제안하는 알고리즘을 이용한 거리 측정 비교  
Table 3. Comparison of distance measurement using the proposed algorithm with the distance of the vehicle ahead in the real environment.

Area	4	5	6
Lidar Distance (m)	9.12m	9.81m	7.05m
Calibration Distance (m)	8.94m	9.56m	6.92m
Error (%)	1.9%	2.5%	1.8%

표 4. 기존 위치 정합 알고리즘과 제안하는 알고리즘 성능 비교  
Table 4. Comparison of performance between the existing positioning algorithm and the proposed algorithm .

Area	4	5	6
Lidar Distance (m)	9.12m	9.81m	7.05m
Suggested Calibration Distance (m)	8.94m	9.56m	6.92m
Existing Calibration Distance (m)	8.32m	9.4m	5.8m
Difference Value	0.52m	0.16m	1.12m

## V. 결 론

본 논문에서는 ROI 기반 위치 정합 알고리즘을 이용하여 차량 전방의 객체를 인지하고 이것의 라이더 좌표를 카메라 좌표로 변환하여 위치를 추출하는 시스템을 제안하였다. ROI 기반 위치 정합 알고리즘은 기존 방법과 달리 카메라 센서로부터 들어오는 이미지를 2×3영역으로 나누어서 각각 라이더 센서와 위치 정합을 진행한다. 라이더 센서와 카메라 센서의 위치 정합은 센서 융합에 필수적이며 이를 통하여 객체의 위치와 거리를 알 수 있다. 객체의 위치와 거리를 구하기 위해서는 ROI 기반 위치 정합 알고리즘을 이용하였고, YOLO v4의 결과 값인 중심좌표를 이용하여 객체의 거리를 추출하였다. 실험은 차량 시뮬레이션 및 실제 환경에서 이루어졌다. 실험 결과는 실제 거리와 비교했을 때 정확했으며 실제 환경에 적용하여도 오차가 5% 미만인 것을 확인했다. 앞으로는 여러 대의 카메라를 이용하여 전방, 후방, 측방의 차량, 사람, 장애물을 인식하여 효율적인 자율주행이 될 것이다. 또한, 제안된 ROI 기반 위치 정합 알고리즘을 제안함으로써 더 좋은 자율주행이 될 것이라고 예상된다.

## References

- [1] R. Okuda, K. Yuki, and T. Kazuaki, "A survey of technical trend of ADAS and autonomous driving," in *Technical Papers of 2014 Int. Symp. VLSI Design, Automat. and Test*, pp. 1-4, 2014.
- [2] K. Jelena, J. Nenad, and D. Vujo, "Sensors and sensor fusion in autonomous vehicles," in *2018 26th TELFOR IEEE*, pp. 420-425, 2018.
- [3] Z. Feihu, C. Daniel, and K. Alois, "Vehicle detection based on LiDAR and camera fusion," in *17th Int. IEEE Conf. ITSC*, pp. 1620-1625, 2014.
- [3] Z. Feihu, C. Daniel, and K. Alois, "Vehicle detection based on LiDAR and camera fusion," in *17th Int. IEEE Conf. ITSC*, pp. 1620-1625, 2014.
- [4] E. S. Kim and S. Y. Park, "Extrinsic calibration of a camera-LIDAR multi sensor system using a planar chessboard," in *2019 Eleventh ICUFN IEEE*, pp. 89-91, 2019.
- [5] Y. Park, et al., "Calibration between color



- camera and 3D LIDAR instruments with a polygonal planar board,” *Sensors*, vol. 14, no. 3, pp. 5333-5353, 2014.
- [6] P. Zoltan and H. Levente, “Accurate calibration of LiDAR-camera systems using ordinary boxes,” in *Proc. IEEE Int. Conf. Comput. Vision Wkshps.*, pp. 394-402, 2017.
- [7] X. Gong, Y. Lin, and J. Liu, “3D LIDAR-camera extrinsic calibration using an arbitrary trihedron,” *Sensors*, vol. 13, no. 2, pp. 1902-1918, 2013.
- [8] G. Pandey, J. R. McBride, S. Savarese, and R. M. Eustice, “Automatic targetless extrinsic calibration of a 3D lidar camera by maximizing mutual information,” in *AAAI’12*, pp. 2053-2059, Jul. 2012.
- [9] Y. Liu, S. Liu, Y. Cao, and Z. Wang, “Automatic chessboard corner detection method,” *IET Image Process.*, vol. 10, no. 1, pp. 16-23, 2015.
- [10] S. Nick, et al., “RegNet: Multimodal sensor registration using deep neural networks,” in *2017 IEEE Intell. Veh. Symp.*, pp. 1803-1810, 2017.
- [11] V. Fremont and P. Bonnifait, “Extrinsic calibration between a multi-layer lidar and a camera,” in *2008 IEEE Int. Conf. Multisensor Fusion and Integration for Intell. Syst.*, pp. 214-219, 2008.
- [12] C. Huaiyu, et al., “A novel calibration board and experiments for 3D LiDAR and camera calibration,” *Sensors*, vol. 20, no. 4, 2020.
- [13] Z. Pusztai, I. Eichhardt, and H. Levente, “Accurate calibration of multi-lidar-multi-camera systems,” *Sensors*, vol. 18, no. 7, 2018.
- [14] V. Martin, et al., “Calibration of rgb camera with velodyne lidar,” *Comput. Sci.*, 2014.
- [15] M. Quigley, et al., “ROS: An open-source robot operating system,” in *ICRA*, p. 5, 2009.
- [16] L. Frank, K. Ulrich, and K. Stephan, “Robust recognition of traffic signals,” in *IEEE Intell. Veh. Symp.*, pp. 49-53, 2004.
- [17] R. Joseph, et al., “You only look once: Unified, real-time object detection,” in *Proc. IEEE Conf. Comput. Vision and Pattern Recognition*, pp. 779-788, 2016.
- [18] G. Ross, et al., “Rich feature hierarchies for accurate object detection and semantic segmentation,” in *Proc. IEEE Conf. Comput. Vision and Pattern Recognition*, pp. 580-587, 2014.
- [19] B. Alexey, W. Chien-Yao, and L. Hong-Yuan Mark, “Yolov4: Optimal speed and accuracy of object detection,” *arXiv preprint arXiv:2004.10934*, 2020.
- [20] R. Joseph and F. Ali, “Yolov3: An incremental improvement,” *arXiv preprint arXiv:1804.02767*, 2018.
- [21] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” *NIPS 2012*, vol. 25, pp. 1097-1105, 2012.
- [22] L. Tsung-Yi, et al., “Microsoft coco: Common objects in context,” *Eur. Conf. Comput. Vision*, Springer, Cham, pp. 740-755, 2014.
- [23] S. Ziegler and R. Höppler, “Extending the IPG CarMaker by FMI compliant units,” in *Proc. 8th Int. Modelica Conf.*, pp. 779-783, Dresden, Germany, Mar. 2011.

윤 영 진 (Young Jin Yoon)



2020년 2월 : 경북대학교 전자공학부 공학사  
 2020년 3월~현재 : 경북대학교 전자 전기공학부 석사과정  
 <관심분야> 인공지능, 자율주행, 센서 융합  
 [ORCID:0000-0003-4440-4069]

한 동 석 (Dong Seog Han)



1987년 : 경북대학교 전자공학과 공학사  
 1989년 : 한국과학기술원 전기 및 전자공학과 공학석사  
 1993년 : 한국과학기술원 전기 및 전자공학과 공학박사  
 1996년~현재 : 경북대학교 전자공학부 교수  
 <관심분야> 통신 신호처리, 자율주행자동차, 인공지능  
 [ORCID:0000-0002-7769-0236]