

PointCloud에서 2D Object Detection 모델을 활용한 3D Object Detection 수행 방법 제안

임수환, 김효은, 임유정, 오준영, 박용주*

*한국전자기술연구원

ish0723@keti.re.kr, hekim@keti.re.kr, is4655@keti.re.kr, ohjy708@keti.re.kr, *suede8247@keti.re.kr

A Proposal of 3D Object Detection method using 2D Object Detection model in PointCloud

Lim Su Hwan, Kim Hyo Eun, Lim Yu Jeong, Oh Jun Yeong, Park Yong Ju*

*Korea Electronics Technology Institute

요 약

본 논문은 PointCloud 기반의 3D 객체 탐지에서 긴 개발 주기를 요구하는 기존 인공지능 접근 방식의 한계를 지적하고, 기개발된 2D Object Detection 모델과 IMU 센서를 활용하여 학습 과정 없이 3D 객체를 탐지하는 방법을 제안한다. 제안한 방법은 라이더-카메라 간 거리, 시야각, IMU 회전행렬, 2D 객체 탐지 결과를 기반으로 객체가 존재할 수 있는 공간 영역을 정의하고, 법선 벡터를 활용한 점 필터링 및 DBSCAN 알고리즘을 통한 군집화를 통해 객체 점을 추출한다. 실험 결과를 통해 제안 방식의 정확한 동작을 확인하였으며, 본 연구는 3D 객체 탐지의 빠른 프로토타이핑과 학습 데이터 생성 도구로서의 가능성을 보여준다.

I. 서 론

로봇, 자율주행 등 산업분야에서 라이더 센서는 빛의 ToF를 이용하여 거리를 측정하는 센서로, 카메라와 함께 주변 탐색에 유용하게 사용되고 있다. 최근방식은 다중으로 결합하여 라이더 센서 주변을 PointCloud 형태로 정보를 수집하는 것이 일반적이다. PointCloud는 BEV(Bird's Eye View)로 변환하여 2D 이미지를 생성하고 2D CNN을 이용하여 Object를 찾거나 PV-RCNN과 3D CNN 모델을 개발하여 적용하는 방식으로 활용된다.[1][2][3][4] 최근에는 라이더센서만을 이용한 알고리즘에서 적은 정보량으로 인하여 다양한 센서와 인공지능을 융합하고자하는 시도가 이어지고 있다.[5][6][7][8] 하지만 이러한 연구들은 인공지능을 활용하는 과정에서 새로운 인공지능 모델 개발, 학습용 데이터셋 생성, 학습, 모델 검토로 긴 개발 주기를 가지게 된다. 반면, 2D Object Detection은 이미 연구가 활발히 진행되어 좋은 성능의 다양한 객체를 학습한 모델들이 계속해서 만들어지고 있다. 이 부분에 착안하여 본 논문에서는 기개발된 2D Object Detection 모델을 활용하여 라이더, 카메라, IMU센서와 간단한 알고리즘을 결합하여 3D Object Detection을 수행할 수 있는 방법을 제안한다.

II. 본론

본 논문에서는 크게 객체 영역 설정, 점 탐색, 객체 탐색 이하 세 단계에 걸쳐 3D Object Detection을 수행한다.

II-1. 객체 영역 설정

본 과정에서는 1) 라이더와 카메라 사이의 거리, 2) 카메라의 시야각, 3) IMU 사이의 회전행렬, 4) 객체 탐지정보를 결합하여 객체가 존재할 수 있는 영역을 찾는다. 그림 1의 (a)에서 식 1과 및 식 2의 계산을 통해 두 IMU센서를 통해 얻어지는 쿼터니언을 활용하여 회전행렬을 얻어, 이동행

렬과 결합하면 변환행렬을 얻을 수 있다. 아래의 식에서 q_{ds} 는 카메라에서 라이더로의 쿼터니언, q_d 는 라이더의 쿼터니언, q_s 는 카메라의 쿼터니언이고, C 는 변환행렬, R 은 회전행렬, T 는 전이행렬을 의미한다.

$$q_{ds} = q_d q_s^{-1} \dots (\text{식1}), \quad C = \begin{pmatrix} R & T \\ 0 & 1 \end{pmatrix} \dots (\text{식2})$$

(b)에서는 카메라의 시야각과 위치 및 식 3을 이용하여 객체의 각 꼭지점으로 향하는 4개의 벡터를 구할 수 있다. 이 때, \vec{p} 는 카메라 초점에서 객체 사각형의 각 점으로 향하는 벡터들, \vec{d} 는 카메라가 보는 방향의 벡터, \vec{v} 는 사각형의 수직벡터, \vec{h} 는 사각형의 수평벡터를 의미한다.

$$\vec{p}_0 = \vec{d} + \vec{v}_{up} + \vec{h}, \vec{p}_1 = \vec{d} + \vec{v}_{down} + \vec{h}_l, \vec{p}_2 = \vec{d} + \vec{v}_{down} + \vec{h}_r, \vec{p}_3 = \vec{d} + \vec{v}_{up} + \vec{h}_r \dots (\text{식3})$$

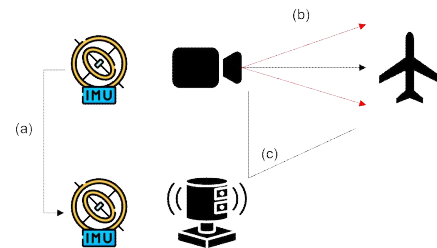


그림 1. 객체 영역 설정

식 3에서 구한 벡터에 변환행렬을 곱하면 라이더 중심의 좌표계에서 객체가 존재할 수 있는 영역을 사각뿔의 형태로 얻을 수 있다.

II-2. 점 탐색

카메라의 특성을 생각해볼 때, 바라보는 방향의 가장 가까운 점만 볼 수 있고, 그 뒤에 존재하는 점은 볼 수 없다. 그러므로 카메라에서 객체를 탐지할 때, 그 근거가 되는 객체는 카메라 기준 가장 앞의 점을 포함한 군집을 보고 객체를 인식했다고 판단할 수 있다. 본 과정에서는 그 점을 찾는다. 먼저 객체 영역 설정 과정에서 얻은 사각뿔 내부의 점을 찾는다. 그림

2와 같이 각 면에 대해 법선벡터를 구하고, 꼭지점과 목표로 하는 점을 잇는 벡터를 구한뒤, 점의 벡터를 4가지 법선벡터와 각각 내적한다. 이 때, 모두 음수일 경우, 법선의 반대 방향에 점이 존재함을 의미하므로, 사각뿔의 내부에 있음을 알 수 있다. 구현한 프로그램에서는 이렇게 찾은 점들을 모아 새로운 배열을 생성한다.

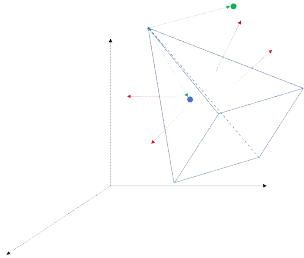


그림 2. 내부 점 판별

다음 과정으로 찾은 점들과 꼭지점 사이의 거리를 측정하여 가장 가까운 점을 찾으면 해당 점이 객체에 포함된 점으로 id를 부여한다. 구현한 프로그램에서는 속도의 문제로, 새로운 배열을 생성하는 과정에서 최단거리인 점의 인덱스를 기록해두었다.

II-3. 객체 탐색

마지막 과정에서는 점 탐색 과정에서 찾은 점을 이용하여 군집을 찾는다. 본 논문에서는 DBSCAN 알고리즘의 군집을 찾는 알고리즘을 활용하였다. 단, DBSCAN 알고리즘은 모든 점이 노이즈로 판별될 경우, 일반적으로 최고 N^2 시간복잡도를 가진다. 하지만 본 논문에서는 1개의 군집만 확정적으로 찾으므로 빠른 시간 내에 객체에 해당하는 점들을 찾아낼 수 있다.

III. 결론

본 논문에서는 기계발되어 있는 2D Object Detection 모델과 라이다 및 IMU 센서를 활용하여 3D Object Detection 방법에 대해 제안했으며, 그림 3은 그 결과를 보여준다.

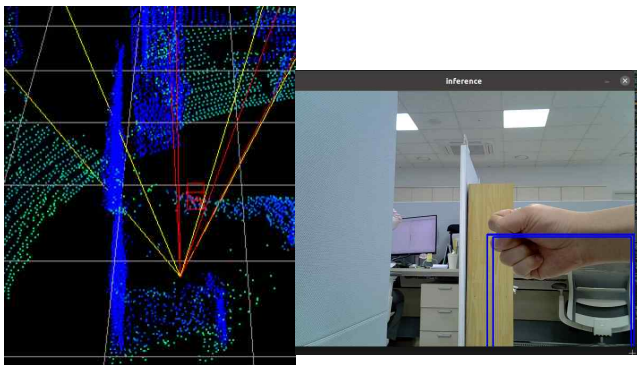


그림 3. 구현 결과

사용한 모델로는 yolov5n을 사용하였으며, yolov5 공식 github에 있는 모델로 coco dataset으로 학습되어 45.7 mAP@50의 준수한 성능을 보여주는 모델이다.[9] 그림 3과 같이 탐지된 객체를 3D화면에서 빨간 점으로 표시하고, 그 객체의 영역을 빨간 사각형으로 표시했을 때, 기대와 같이 동작하는 모습이다.

본 논문의 제안방식과 그 결과는 기계발된 모델들을 활용하여 별도의 인공지능 모델링이나 학습없이 PointCloud에서 3D 객체를 찾는 방법으로 긴 개발주기를 요구하지 않아 산업의 빠른 발전에 기여할 수 있음을 의미한다. 또한, 더욱 높은 정확도를 위해 3D 방식의 객체 탐지 모델을 개발할 때 데이터셋을 만들어내는 데 사용할 수 있어, 관련 분야 연구의 발전에도

기여할 수 있다.

추후 연구 방향으로는 기존의 3D Object Detection과 비교하여 속도 및 정확도를 비교하고 더 높은 성능을 내기 위한 연구를 지속할 계획이다.

ACKNOWLEDGMENT

본 연구는 과학기술정보통신부/정보통신기획평가원의 연구비를 지원으로 수행됨 (RS-2024-00400712,, 도심항공교통의 비정상 상황 인지 및 대응을 위한 온보드 기반 지능형 항전SW 플랫폼 기술 개발 및 실증)

참 고 문 헌

- [1] Shaoshuai Shi, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li, "PV-RCNN: Point-Voxel Feature Set Abstraction for 3D Object Detection," Proc. of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 10529 - 10538, 2020.
- [2] Shaoshuai Shi, Chaoxu Guo, Li Jiang, Zhe Wang, and Hongsheng Li, "PV-RCNN++: Point-Voxel Feature Set Abstraction With Local Vector Representation for 3D Object Detection," arXiv preprint arXiv:2102.00463, 2021.
- [3] Tianwei Yin, Xingyi Zhou, and Philipp Krähenbühl, "Center-based 3D Object Detection and Tracking," Proc. of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 11784 - 11793, 2021.
- [4] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "PointPillars: Fast Encoders for Object Detection from Point Clouds," Proc. of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 12697 - 12705, 2019.
- [5] T. T. Nguyen, M. A. Kabir, H. T. Nguyen, et al., "An Advanced Approach to Object Detection and Tracking in Robotics and Autonomous Vehicles Using YOLOv8 and LiDAR Data Fusion," Electronics, vol. 13, no. 12, pp. 1 - 18, 2024.
- [6] C. Nguyen, M. A. Kabir, and H. Nguyen, "Enhanced Object Detection in Autonomous Vehicles through LiDAR - Camera Sensor Fusion," Sensors, vol. 24, no. 3, pp. 1 - 17, 2024.
- [7] L. Zhu, Z. Wang, and H. Ye, "LIC-Fusion: LiDAR-Inertial-Camera Odometry," arXiv preprint, arXiv:1909.04102, 2019.
- [8] A. K. Panigrahi, S. K. Nayak, and B. B. Pati, "Real-Time Object Detection Using LiDAR and Camera Fusion for Autonomous Driving," Proceedings of 2021 International Conference on Intelligent Technologies (CONIT), pp. 1 - 6, 2021.
- [9] Ultralytics, "YOLOv5: Real-Time Object Detection," GitHub Repository, <https://github.com/ultralytics/yolov5>, Accessed Apr. 15, 2025.