

# 효율적인 로봇 주행을 위한 통합 비전과 포인트 클라우드 필터링

심명보, 조세운, 이혜민

smb0112@keti.re.kr, swcho@keti.re.kr, lee10849@keti.re.kr

## Integrated Vision and Point Cloud Filtering for Efficient Robot Navigation

Sim Myong Bo, Cho Se Woon, Lee Hea Min  
Korea Electronics Technology Institute

### 요약

이 논문에서는 자율 주행 로봇이 과수원에서 다양한 장애물을 효과적으로 인식하는 새로운 접근 방법을 제시한다. 이 시스템은 ROS2를 사용해 ZED 카메라로부터 얻은 2D 이미지와 포인트 클라우드 데이터를 결합하여 처리한다. 그 과정에서 YOLOv8을 활용한 객체 분할을 통해 장애물을 분할하고 불필요한 포인트 클라우드 데이터를 제거함으로써 로봇의 경로 탐색 능력을 개선하고자 한다. 제안한 방법을 통해 로봇이 과수원과 같은 복잡한 환경에서 더욱 효율적인 주행을 할 수 있을 것으로 기대한다.

### I. 서론

자율 주행 로봇은 농업 분야에서 생산성 향상과 인력 문제 해결을 위한 중요한 기술로 주목받고 있다.[1] 특히 과수원과 같은 농업 환경에서의 로봇 활용은 수확, 운반, 모니터링 등 다양한 작업에 적용될 수 있어 그 가치가 높다. 그러나 과수원은 일반 도로와 달리 복잡하고 비정형적인 장애물이 많아 로봇의 주행에 어려움이 따른다.[2][3]

특히 감귤류 과수원의 경우 나무의 가지가 길게 뻗어 있고 불규칙한 모양을 띄고 있어 나무 일부가 로봇의 경로를 침범하는 경우가 많으며 로봇은 이를 장애물로 인식하는 경우가 많다. 사람은 이러한 문제에 대해 유연하게 대처하여 일부 가지는 무시하고 지나갈 수 있지만, 로봇은 이러한 장애물을 모두 회피하려 하기 때문에 주행 경로가 비효율적으로 변경되거나 심지어 일부는 막힌 것으로 판단하여 주행을 포기하는 문제가 발생한다.

따라서 로봇이 과수원 환경에서 효과적으로 주행하기 위해서는 불필요한 장애물을 감지하지 않거나 판별하여 무시하고 주요 장애물만 인식하여 경로를 설정하는 것이 중요하다. 이를 위해 본 연구는 비전 데이터를 활용한 포인트 클라우드 필터링을 통한 효율적인 장애물 인식 방법을 제안하고자 한다.

### II. 시스템 구현

#### 2.1 시스템 구성 및 흐름

본 연구에서 제안하는 시스템은 NVIDIA Jetson 보드와 ZED 스테레오 카메라를 기반으로 구축된다. ROS2 프레임워크 위에서 실행되는 기본 시스템을 위해 본 시스템은 여러 노드로 구성되어 있으며 각각의 노드들은 토픽의 수신과 처리 같은 기본적인 기능과, yolov8-seg 모델을 활용한 객체 분할 적용, 객체 분할 마스크를 바탕으로 포인트 클라우드 필터링 등의 작업을 분산 수행한다.

최종적으로는 zed-ros2-wrapper로부터 발행되는 토픽을 구독하여 객체

검출을 적용한 후 필터링된 포인트 클라우드를 다시 ROS2 토픽으로 발행하게 된다.

#### 2.2 데이터 동기화

ZED 스테레오 카메라로부터 수신되는 2D 이미지와 로봇의 주행을 위해 사용되는 포인트 클라우드 데이터는 시간적으로 일치해야 하므로 동기화는 필수적인 과정이다. 데이터에 대한 타임스탬프를 기반으로 카메라 2D 이미지와 포인트 클라우드 간 허용 가능한 시차  $\Delta t$  이내의 데이터만 선별하여 쌍으로 묶어 처리한다.

$$|t_{image} - t_{pointcloud}| \leq \Delta t$$

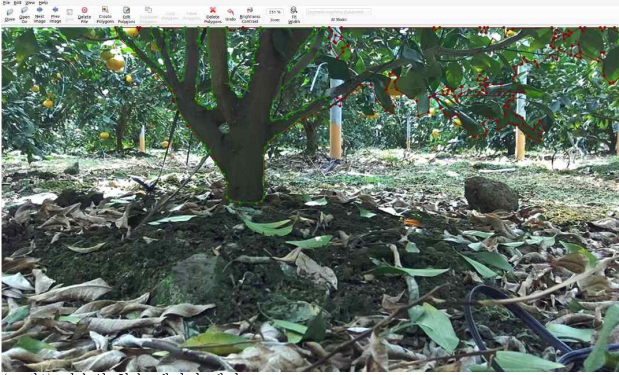
이미지와 포인트 클라우드 데이터는 별도의 큐에 저장되어 수식에 따라 허용 범위 내의 데이터 쌍이 선택되어 다음 처리 단계로 이동되며, 그 외의 시간차가 큰 데이터는 버려지게 된다. 이를 통해 시간적으로 정합성이 있는 데이터를 확보하여 로봇 주행에 차질이 발생하지 않도록 한다.

#### 2.3 객체 분할 인공지능 모델

로봇 주행 과정에서 회피가 불필요한 객체를 인지하기 위한 수단으로 yolov8-seg 모델을 사용한다. 사용 환경과 실시간성을 고려하여 모델 크기는 nano를 사용하며 TensorRT 모델로 제작되었다.[4] 모델은 과수원 로봇 주행 데이터를 바탕으로 학습되었다.

#### 2.4 포인트 클라우드 투영 및 필터링

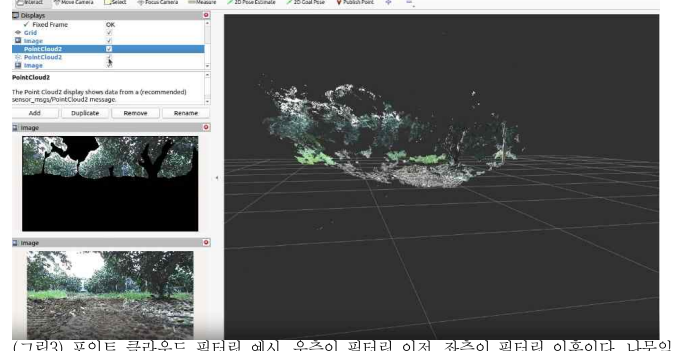
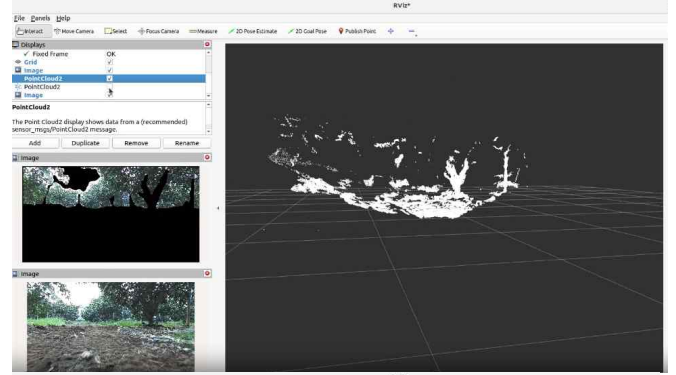
yolov8-seg 모델은 ZED 카메라를 기반으로 2D 이미지에 대한 객체 분할 작업을 수행하므로 이를 3D 포인트 클라우드 데이터에 접목해야 한다.



(그림1) 과수원 학습 데이터 예시



(그림2) yolov8-seg 모델 예측



(그림3) 포인트 클라우드 필터링 예시. 우측이 필터링 이전, 좌측이 필터링 이후이다. 나뭇잎 부분의 포인트 클라우드가 제거된 것을 확인할 수 있다.

분할된 2D 이미지 정보를 3D 포인트 클라우드에 투영하여 객체의 포인트 클라우드를 추출하는 과정은 알려진 바와 같이 다음 수식을 통해 설명할 수 있다.

$$u = \frac{f_y \times x}{z} + C_x$$

$$v = \frac{f_y \times y}{z} + C_y$$

그러나 ZED 카메라는 ROS2에서 사용하는 좌표계가 서로 다르므로 이를 보정하여 사용해야 한다.[5]

$$u = \frac{f_x \times (-y)}{x} + C_x$$

$$v = \frac{f_y \times (-z)}{x} + C_y$$

매핑된 포인트 클라우드의 정보를 바탕으로 객체 분할 마스크 내부에 있는 포인트인지 판별한 후 사용자가 지정한 조건에 따라 필터링 여부를 적용하여 최종적으로 완성된 포인트 클라우드를 다시 ROS2 토픽으로 발행하게 된다.

### III. 포인트 클라우드 필터링 결과

제안된 방법을 검증하기 위해 실제 과수원 환경에 대해 실험을 진행하였다. 그 결과 그림 5를 보면 알 수 있듯 제안된 방법이 복잡한 과수원 환경 속에서 안정적으로 작동하며 불필요한 장애물의 포인트 클라우드를 효과적으로 필터링할 수 있음을 확인할 수 있다.

### IV. 결론

본 논문에서는 과수원 환경에서 로봇의 주행을 위한 포인트 클라우드 필터링 기법을 제안하였다. 관심 영역이 아닌 객체의 포인트를 제거함으로써 로봇 주행에 방해가 되는 불필요한 요소를 효과적으로 필터링할 수 있다.

제안된 방법은 ROS2 프레임워크를 통해 임베디드 시스템에서도 안정적으로 작동하며 포인트 클라우드의 필터링 효과를 시각적으로 확인하였다.

이에 따라 본 연구는 로봇 주행에 대한 센서 정보의 선별적 활용 방안을 제시하고 실시간성을 해치지 않아 다양한 환경에서 안정성과 효율성을 확보하는 하나의 방법론으로서 농업뿐 아니라 다양한 복잡한 환경 속 자율주행이 요구되는 분야에 적용할 가능성을 보았다는 점에서 의미가 있다. 추후 필터링과 객체 분할의 성능과 속도를 더욱 개선하기 위한 연구가 필요할 것으로 보인다.

## ACKNOWLEDGMENT

본 결과물은 농림축산식품부의 재원으로 농림식품기술기획평가원의 노지 분야 스마트농업기술 단기 고도화사업의 지원을 받아 연구되었음 (122032-03)

## 참고 문헌

- [1] MDPI Agriculture, "Special Issue: Robots and Autonomous Machines for Agriculture," MDPI, 2023
- [2] Biehler, D., and Paudel, A., "Using Autonomous Pathing Orchard Robots," Center for Sustaining Agriculture and Natural Resources, Washington State University, Aug. 2023
- [3] Xiamen University & University of Essex, "Novel Laser-Based Obstacle Detection for Autonomous Robots on Unstructured Terrain," Sensors 2020, vol. 20, no. 18, Sept. 2020
- [4] Shin, D., Kim, J. "A Deep Learning Framework Performance Evaluation to Use YOLO in Nvidia Jetson Platform," Applied Sciences, vol. 12, no. 8, April 2022.
- [5] Stereolabs (<https://www.stereolabs.com/docs>)