

복분자 재배 환경에서 수확물 운반 도우미 로봇을 위한 YOLOv5와 LiDAR 활용 자율주행 시스템

김기오*¹, 온병원*², 이석훈*³, 이연식*⁴, 정현준**¹, 정동원**²

YOLOv5 and LiDAR-based Autonomous Driving System for Agricultural Produce Transportation in the Cultivation Environment of Rubus Coreanus

Gio Gim*¹, Byung-Won On*², Sukhoon Lee*³, YonSik Lee*⁴, Hyunjun Jung**¹, and Dongwon Jeong**²

본 결과물은 행정안전부와 전라북도농업기술원의 재원으로 “행안부 주민주도형 지역균형 뉴딜 우수사업”의 지원을 받아 연구되었음. 또한 이 성과는 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(No. 2021R1F1A1047768)

요약

농가와 농가인구의 지속적인 감소 및 고령화 문제는 농업의 생산성에 큰 영향을 주고 있다. 이런 문제점들을 해결하기 위한 방안으로 농업용 로봇 개발의 필요성이 강조되고 있다. 복분자 재배의 특성 상 가파른 언덕과 좁은 사로와 같은 독특한 환경 조건을 고려한 농업용 로봇의 개발이 요구된다. 이에 본 논문에서는 LiDAR 센서와 YOLOv5를 기반으로 한 실시간 객체 인식 기술을 활용하여 복분자 재배 환경에 특화된 자율주행 시스템을 제안한다. 제안 시스템은 사용자의 키 입력으로 로봇을 목적지로 이동시키며, 이동 경로 상에 있는 설정된 장애물과의 안전 거리를 유지하면서 효과적으로 목적지까지 이동한다. 실험 결과 거리유지 실험에서는 평균 약 0.038m의 오차를 보였고, 위치정확도에 대한 실험에서는 평균 약 5.7252%의 오차를 보였다. 이러한 결과는 복분자 재배 환경같은 좁은 사로에서도 원활한 자율주행을 보여준다.

Abstract

The continuous decrease in the number of farming households and the aging population have a significant impact on agricultural productivity. In response to these challenges, the need for the development of agricultural robots is being emphasized. Considering the unique environmental conditions of rubus coreanus farming, such as steep hills and narrow paths, the development of specialized agricultural robots is required. Therefore, this paper proposes an autonomous driving system optimized for the rubus coreanus farming environment, utilizing LiDAR sensors and real-time object detection technology based on YOLOv5. The proposed system allows the robot to be moved to its destination through user key inputs, effectively maintaining a safe distance from predetermined obstacles along the path. The experimental results showed an average error of about 0.038 meters in the distance maintenance test and an average error of about 5.7252% in the location accuracy test. These results demonstrate smooth autonomous driving in narrow paths, similar to those found in rubus coreanus farming.

Keywords

agricultural robots, autonomous driving, obstacle avoidance, YOLOv5, LiDAR

* 군산대학교 소프트웨어학부
- ORCID¹: <https://orcid.org/0009-0007-5615-6361>
- ORCID²: <https://orcid.org/0000-0001-6929-3188>
- ORCID³: <https://orcid.org/0000-0002-3390-5602>
- ORCID⁴: <https://orcid.org/0000-0002-4868-1058>
** 군산대학교 소프트웨어학부(교신저자)
- ORCID¹: <https://orcid.org/0000-0002-6717-1395>

- ORCID²: <https://orcid.org/0000-0001-9881-5336>

· Received: Nov. 10, 2023, Revised: Dec. 05, 2023, Accepted: Dec. 08, 2023

· Corresponding Author: Hyunjun Jung and Dongwon Jeong
School of Software, Kunsan National University, 558, Deahak-ro, Gunsan,
Jeollabuk-do, Korea

Tel.: +82-63-469-8911, Email: {junghj85, djeong}@kunsan.ac.kr

1. 서 론

2022년 통계청의 농림어업총조사 결과에 따르면 농가와 농가인구의 지속적인 감소와 고령 인구의 비율 증가가 두드러지게 나타난다. 2022년 기준 농가는 103만 3000가구, 농가인구 216만 6000명으로 전년 대비 2000가구, 4만 9000명 각각 감소했다. 또한, 65세 이상의 고령 인구가 농가인구 중 49.8%로 전년 대비 3.0% 증가했다. 이와 같은 변화는 농업 분야의 노동 인력 감소와 그에 따른 생산성 저하를 일으킨다[1].

농업 분야에서의 생산성 저하를 방지하고 노동 인력 부족 문제를 해결하기 위해 다양한 농업용 로봇 개발이 국내외로 활발히 진행되고 있으며, 고된 농작업을 대신할 수 있는 농업용 로봇의 필요성이 점점 증가하고 있다[2][3].

[2]에서는 기술을 통합한 자율주행 트랙터의 작동 원리와 응용을 기술한다. 트랙터는 GPS, 자이로스코프, LiDAR 센서의 결합을 통해 정밀한 위치 정보와 방향성을 파악한다. 또한, 자동화된 시스템을 통해 농작물의 상태와 환경 변화를 실시간으로 감지하며, 이를 바탕으로 최적의 작업 경로를 결정한다. 하지만 정형화된 넓은 환경에서 활용할 수 있다.

[3]에서는 토마토 자동 수확 로봇의 고도화된 기술을 다룬다. 로봇은 다양한 센서, 특히 GPS와 LiDAR, 카메라 기술의 조합을 활용하여 수확물의 위치를 정밀하게 탐지한다. 또한, 로봇이 작물을 수확할 뿐 아니라 운반하며, 상태를 지속해서 모니터링을 수행한다. 하지만 실내 환경에서 활용되며, 레일을 설치해야 하기 때문에 비용이 많이 드는 단점이 있다.

농업용 로봇의 개발과 활용은 기술적 성능 향상을 넘어서, 재배 환경과 농작물의 특성을 고려한 다양한 연구가 진행되고 있다[4][5].

그림 1과 같은 복분자 재배 환경은 가파른 언덕에서 재배되는 농작물이며, 사람들이 직접적으로 농작물을 수확할 때 언덕을 몇 차례 오르내리며 운반해야 하는 고된 작업이다. 따라서, 사람이 수확한 복분자를 하적하기까지 운반할 수 있도록 도와주는 수확물 운반 도우미 로봇이 필요하다. 그러나 좁은

사로로 이루어져 있어, 로봇이 회피 기동하기 힘들고, 복분자 가지 같은 방해 요소가 많아 장애물을 인식하고 대응하는 방법이 필요하다.



그림 1. 복분자 재배 환경

Fig. 1. Rubus coreanus cultivation environment

수확물 운반 도우미 로봇 개발하기 위해 다양한 요인을 고려해야 한다[6]. 하드웨어 측면에서는 4륜형 또는 궤도형의 구동 플랫폼, 통합 제어 유닛, 인터페이스 모듈 등 주요 구성 요소들의 개발이 필요하다. 반면, 소프트웨어 측면에서는 로봇의 이동 및 안전 관련 동작을 위한 자율주행 시스템이 필요하다.

따라서 본 논문에서는 소프트웨어 측면을 중점으로, LiDAR 센서와 실시간 객체 인식 모델인 YOLOv5 모델을 사용하여 복분자 재배 환경에 최적화된 자율주행시스템을 제안한다.

본 논문의 구성은 다음과 같다. 제2장에서는 관련 연구를 소개하고 분석한다. 제3장에서는 본 논문에서 제안하는 시스템 구조와 하드웨어 구성, 자율주행 시스템을 설명한다. 제4장에서는 실험 환경, 실험 방법 및 결과를 기술한다. 마지막으로, 제5장에서는 결론 및 향후 연구를 기술한다.

II. 관련 연구

[7]에서는 농업 환경의 자율 주행 로봇이 농지 손상을 방지하기 위한 정확한 경로 생성을 강조한다. 이를 위해 Livox Mid70 LiDAR를 활용한 릿지 감지 기반의 실시간 경로 계획 알고리즘을 제안하였다. 알고리즘은 RANSAC을 사용하여 LiDAR 데이터를 선형화하며, 실제 환경 테스트에서 로봇의 안정적인 주행을 확인하였다. 하지만 장애물 회피에 대해 고려하지 않은 문제점이 있다.

[8]에서는 3D LiDAR센서를 활용하여, 거리 데이터를 수집하였다. 얻은 데이터를 Open 3D 데이터로 변환되었으며, 특정 영역의 데이터를 활용하기 위해 Pass through filter를 적용하여 ROI(지정영역)를 선정하였다. 이후, 데이터 밀도를 조절하기 위해 Voxel Down Sampling을 진행하였고, 비정상적인 노이즈 제거를 위해 Radius based outlier removal(거리 기반 이상치 제거)을 적용하여 객체와의 거리 및 장애물 인식을 최적화하였다. 하지만 장애물을 감지할 경우, 자동으로 회피 경로를 생성하고 목적지까지 이동하여 좁은 사로에서 사용하기에 문제점이 있다.

[9]에서는 디지털 트윈(Digital twin) 환경에서 농업용 로봇의 자율주행 및 맵핑 시스템을 제안하였다. 지맵핑(Gmapping)과 카토그래퍼(Cartographer) 2가지 방법으로 지도 작성과 로봇의 위치를 추정하여 스마트팜에서의 결과를 예측 및 비교하였으며, 카토그래퍼로 맵핑한 결과가 오차가 더 적은 결과를 얻었다. 로봇의 주행을 위해 AMCL 기법을 사용하고 5개의 경로 지점을 순차적으로 통과하여, 목적지까지 순조롭게 경로를 따라가는 결과를 보였다. 하지만 LiDAR 센서만으로 장애물을 감지하고 회피하는 것은 좁은 사로에서 사용하기에 문제점이 있다.

[10]에서는 과수원의 특정 환경에서 로봇의 자율주행을 위한 방법론을 제시하였다. 이 연구에서는 LiDAR 스캔 데이터를 활용하여 나무 간의 위치와 거리를 정확하게 감지하는 알고리즘을 개발하였다. 특히, 이 알고리즘은 새롭게 제안된 레이저 빔 모델과 함께 확률적 입자 필터(PF) 방식을 통해 높은 정확도와 안정성을 보장한다. 하지만 장애물 회피에 대해 고려하지 않은 문제점이 있다.

[11]에서는 농업 로봇의 자율주행에 필요한 기술들을 설명한다. SLAM(Simultaneous Localization and Mapping)은 농업 로봇의 핵심 기술 중 하나로, 2D Lidar 기반의 SLAM과 3D Lidar 기반의 SLAM 두 가지 방식을 비교한다. 특히, Cartographer의 2D Lidar SLAM을 활용한 주행 실험이 진행되었으며, Dijkstra 알고리즘과 DWA(Dynamic Window Approach)를 이용하여 경로 계획 및 주행 속도 제어

를 수행한다. 온실 환경에서 실험했으며, 장애물을 회피하여 좁은 사로에서 사용하기에 문제점이 있다.

위의 연구들은 주로 온실과 같이 제어 가능한 환경이나 장애물을 회피하여 목적지까지 이동하는 기능에 초점을 맞추고 있어, 좁은 사로가 특징인 복분자 재배 환경에서 적용하기에 어려움이 있다.

III. 제안 시스템

이 절에서는 제안하는 시스템의 구조를 기술한다. 먼저, 전체적인 작동 흐름과 하드웨어 구조를 설명한다. 이후, 로봇의 이동 방법과 이동 중 장애물을 감지하고 대응하는 기능을 기술한다.

3.1 제안 시스템 구조

그림 2는 제안 시스템 구조를 보여준다. 각 계층에서 동작하는 인식(Perception), 계획(Planning), 제어(Control) 기능은 다음과 같다.

인식 계층은 로봇이 외부 환경을 파악하기 위한 핵심 부분이다. 카메라와 LiDAR 같은 센서들을 사용하여 주변 환경 데이터를 수집하고, 이 데이터는 맵 서버와 오도메트리(Odometry) 소스 및 AMCL(Adaptive Monte Carlo Localization) 알고리즘을 통해 로봇의 위치와 주변 환경에 대한 정보를 제공한다[12].

계획 계층은 로봇이 목표 지점까지의 경로를 계획한다. 입력받은 Move Toward를 기반으로, 전역 및 지역 계획 알고리즘을 사용하여 최적의 경로를 결정한다. 전역 계획(Global Planner)는 전체 경로를 설정하는 반면, 지역 계획(Local Planner)는 로봇이 인식 계층에서 받은 정보를 바탕으로 실시간으로 경로를 결정한다.

제어 계층은 계획된 경로에 따라 로봇을 실제로 움직이는 부분이다. 장애물 검출 여부를 계산하고, 속도와 방향을 조정한다. 이 계층은 로봇의 기본 컨트롤러를 통해 모터와 기타 구동 장치를 조작하여 경로를 따라 이동 명령을 전달한다.

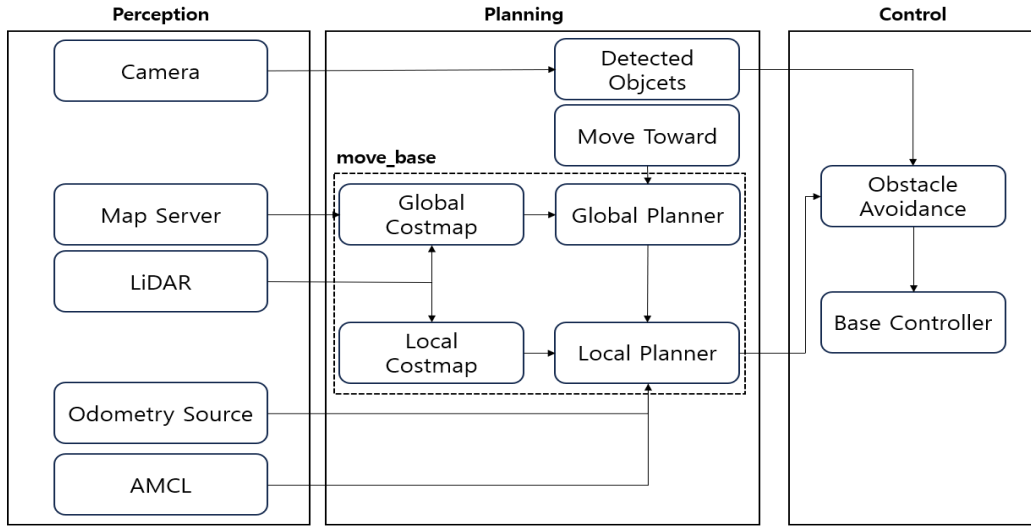


그림 2. 제안 시스템의 구조
Fig. 2. Architecture of proposed system

그림 3은 제안 시스템의 전체적인 작동 흐름을 보여준다. 사용자로부터 입력을 받은 후, 내비게이션 스택을 사용하여 경로를 계획하고 이동한다. 이동 중 장애물을 감지하면 정지하고, 장애물이 감지되지 않으면 다시 이동하여 목적지에 도달한다.

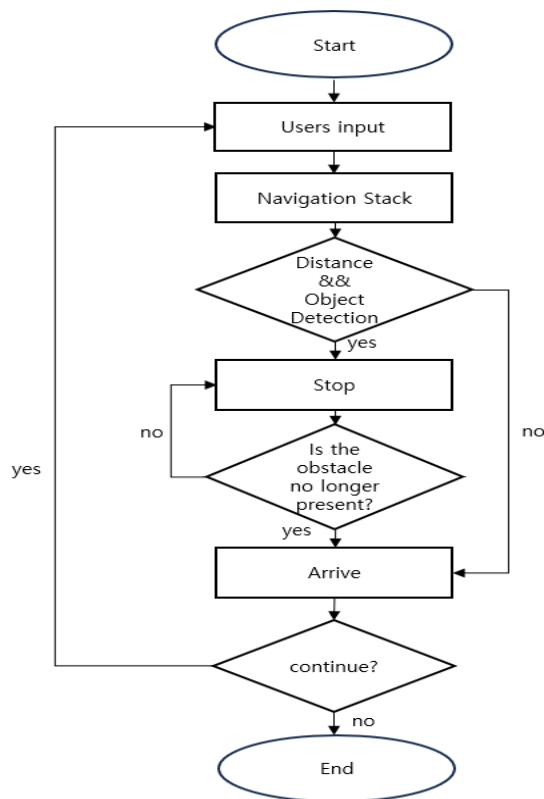


그림 3. 시스템 흐름도
Fig. 3. System flowchart

3.2 하드웨어 구조

본 논문에서는 로봇 응용 프로그램 개발을 위한 통합 플랫폼으로 ROS(Robot Operating System)을 사용한다. ROS는 다양한 로봇 하드웨어에 대한 지원과 함께 노드 기반의 통신 구조를 제공하여 각각의 기능을 독립적인 모듈로 개발하는 것이 가능하다. 또한, 다양한 하드웨어 및 소프트웨어 플랫폼 간의 통합을 용이하게 한다.

그림 4는 리눅스 기반의 Remote PC에서 실행되는 ROS 시스템의 핵심 구성 요소들과 그들 사이의 연결 방식을 보여준다. 마스터(Master) 노드는 중앙 집중식 통신 관리자로서 다수의 노드(Node)간 메시지 교환을 조정한다. 각 노드는 분산된 로봇 시스템 내에서 특정 작업을 수행한다.

무선 통신망(Wi-Fi)를 통해 Remote PC는 라즈베리파이(Raspberry Pi)와 OpenCR 등의 주변 장치들과 데이터를 교환한다. 라즈베리파이는 실시간 객체 인식, 거리 센싱 역할을 수행하는 카메라 및 LiDAR(LDS-02) 센서에 연결되어 있다. 센서로부터 수집된 정보는 실시간으로 처리할 수 있으며, OpenCR은 TTL(Time to Live) 통신을 통해 액추에이터로 구성된 로봇의 이동부를 제어하여 움직임을 가능하게 한다.

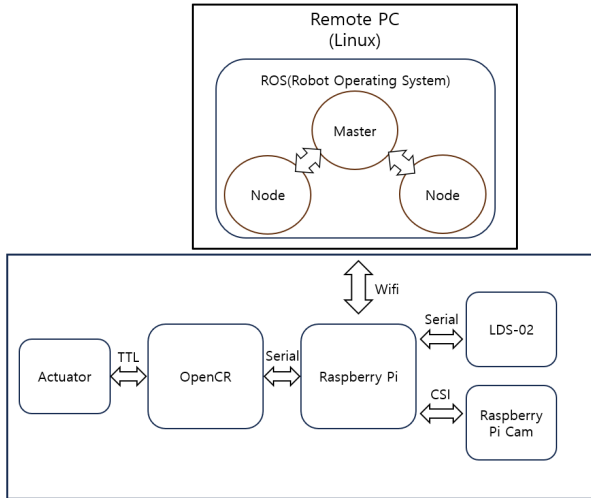


그림 4. ROS 기반 하드웨어 구성도
Fig. 4. Hardware architecture based on ROS

3.3 인터페이스 기반 목표 지점 이동

사용자 인터페이스를 기반으로 한 목표 지점 이동 기능을 통해, 로봇은 사용자가 지정한 목표 위치로 자동으로 이동하게 된다. 로봇이 목표 지점까지 안정적으로 이동하기 위해서는 로봇이 자신의 위치를 파악하고, 동시에 주변 환경의 지도를 생성하는 SLAM(Simultaneous Localization and Mapping) 기술이 필요하다[13]. SLAM을 통해 작성된 지도를 바탕으로 로봇에게 경로 지점을 줬을 때 Navigation stack을 사용하여, 최적의 경로를 생성하고 목적지까지 이동한다[14]. 또한, 생성된 지도상에서 /amcl_pose 토픽을 통해 로봇의 위치 및 방향 정보를 알 수 있다. 본 논문에서는 로봇의 위치와 방향 값을 미리 설정하여 저장한다.

본 논문에서 사용하는 주요 토픽은 표 1과 같다.

표 1. 주요 토픽
Table 1. Key topics

Topic name	Function
/move_base_simple/goal	designating the robot's target location and direction
/amcl_pose	information on the robot's current location and direction
/detected_objects_count	status of real-time object detection
/scan	distance information
/nav_vel, /cmd_vel	robot linear and angular velocity

그림 5는 사용자의 키보드 입력을 기반으로 로봇이 목표 지점까지 이동하는 과정을 나타낸다. 사용자의 입력은 /move_toward 노드로 전달되며, 이 노드는 대응하는 위치 및 방향 값을 /move_base_simple/goal 메시지로 변환하여 /move_base 노드에 토픽을 발행한다. /move_base 노드는 이 토픽을 받아 로봇의 현재 위치에서 목표 지점까지의 최적 경로를 계산하고, 해당 경로를 따라 로봇이 목적지에 도달할 수 있도록 지시를 내린다. 이러한 방식으로 로봇은 사용자가 지정한 위치까지 안정적으로 이동할 수 있다.

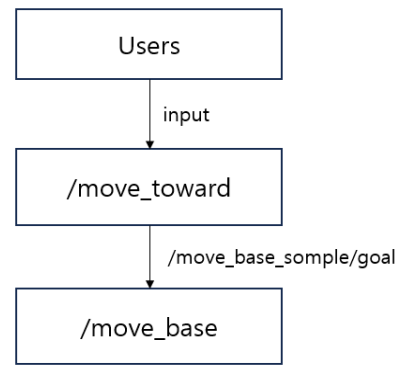


그림 5. 목표 지점 이동 동작 과정
Fig. 5. Target point movement process

3.4 장애물 대응

로봇이 자율주행 환경에서 안정적으로 작동하기 위해서는 실시간 객체 검출의 정확도와 속도가 핵심이다. YOLO(You Only Look Once)는 요구사항을 만족시키는 객체 검출 방식 중 하나로, 단일 합성곱 신경망(CNN) 구조를 사용하여 빠르면서도 정확한 검출이 가능하다. 특히, YOLOv5는 이 중에서도 높은 성능을 자랑하며, 모델의 크기와 경량화 측면에서도 이점이 있어 본 논문에서 선택하여 사용하였다[15].

그림 6은 로봇이 장애물을 감지하고 대응하는 동작 과정을 보여준다. 이동 명령은 /move_base 노드에 의해 처리되며, 해당 명령은 /nav_vel 토픽을 통해 /obstacle_avoidance 노드로 전송된다. /detected_objects 노드는 설정된 장애물에 대한 실시간 객체 인식 결과를 /detected_objects_count 토픽으로 발행한다.

또한, /turtlebot3_lds 노드는 로봇의 정면 20도 범위 내의 장애물과 거리를 측정한 데이터를 /scan 토픽을 통해 /obstacle_avoidance 노드로 전달한다. /obstacle_avoidance 노드는 받은 데이터를 분석하여, 설정된 장애물이 검출되고 동시에 해당 장애물이 로봇의 정면 20도 범위 내에서 설정된 특정 거리 안에 위치할 경우, 이동 명령을 수정하여 장애물과의 충돌 없이 안전한 거리를 유지하며 멈춘다. 이와 반대로, 설정된 장애물이 검출되지 않거나, 장애물이 안전거리 바깥에 있을 경우, /obstacle_avoidance 노드는 아무런 변경 없이 /move_base 노드에서 발행된 원래의 이동 명령을 /turtlebot3_core 노드로 전달하여 이동한다.

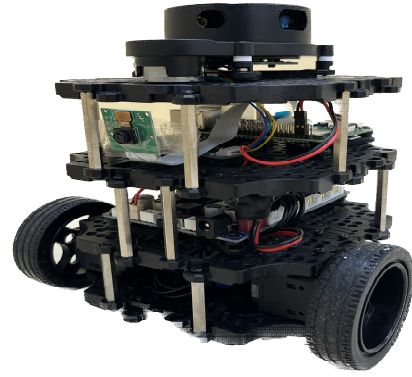


그림 7. 터틀봇3 버거
Fig. 7. Turtlebot3 burger

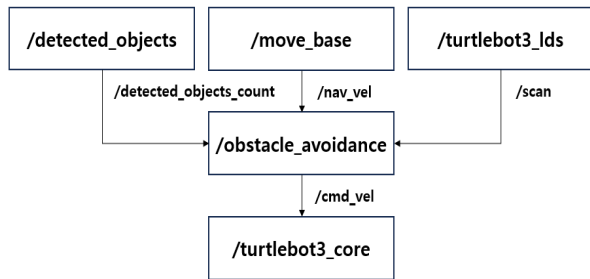


그림 6. 장애물 대응 동작 과정
Fig. 6. Obstacle detection and response processes

IV. 실험 및 평가

4.1 실험 환경

본 논문에서는 그림 7과 같은 ROBOTIS사의 Turtlebot3 Burger 모델을 활용하였으며, 구현 환경은 표 2와 같다. 중앙 처리 장치로는 ROS Noetic 버전을 탑재한 리눅스 우분투 20.04 OS가 설치된 리모트 PC를 사용하였다. 이 PC는 Intel i5-10210U CPU와 NVIDIA GeForce GTX 1650 GPU를 갖추고 있어 높은 처리 능력을 제공한다. 또한, 로봇의 주된 컨트롤러로는 Raspberry Pi 4 Model B를 사용하였다. 로봇의 움직임과 주변 환경 인식 및 실시간 객체 인식을 위해, 360도 레이저 거리 센서 LDS-02와 Raspberry Pi Camera Module V2를 장착하였고, 움직임은 XL430-W250 액추에이터로 제어된다. 또한, 위치와 자세 인식을 위한 IMU는 자이로스코프와 가속도계를 포함한다.

표 2. 구현 환경

Table 2. Implementation environment

Device	Feature	Specification
RemotePc (UX534F)	OS	Linux ubuntu 20.04
	CPU	Intel i5-10210U
	GPU	GeForce GTX 1650
Turtlebot3 Burger	Memory	8GB
	Size(L x W x H)	13.8cm x 17.8cm x 19.2cm
	SBC	Raspberry Pi 4 Model B (4GB)
	MCU	32-bit ARM Cortex®-M7 with FPU (216 MHz, 462 DMIPS)
	LDS	360 Laser Distance Sensor LDS-02
	Camera	Raspberry Pi Camera Module V2
	Actuator	XL430-W250
	IMU	Gyroscope 3 Axis, Accelerometer 3 Axis

4.2 실험 방법 및 결과

본 논문에서 자율주행 로봇 주행 테스트를 수행하기 위해 그림 8과 같은 복분자 재배 환경을 구축하였다. 로봇의 크기를 고려하여 환경은 길이 113cm, 폭 54cm, 높이 27cm의 크기로 설계되었다. 또한, 해당 테스트 환경 내에서 SLAM 기술을 사용하여 로봇이 생성한 지도를 보여주며, a, b, c 지점은 사용자가 지정한 목표 도착 지점을 보여준다.



그림 8. 주행 테스트 환경 및 SLAM 경로 지점
Fig. 8. Navigation test environment and SLAM path points

표 3은 SLAM을 통해 맵핑된 환경 내에서 정의된 각 목표 지점 a, b, c에 대한 2차원 좌표 위치 (x,y)와 3차원 공간에서 해당 지점의 회전을 나타내는 방향(z,w) 정보 값을 나타낸다.

표 3. 위치 및 방향 좌표
Table 3. Position and orientation coordinates

Position & orientation	Target point		
	a	b	c
x	0.1057861	1.9142286	0.9023067
y	-0.0205037	-0.3593802	-0.5280922
z	-0.0308275	-0.9907958	0.9997763
w	0.9995247	0.1353643	0.0211472

로봇에게 임의로 선택된 지점 a, b, c로 이동 명령을 내렸다. 이동 경로 상에는 객체 인식이 설정된 장애물과 설정되지 않은 장애물을 배치하여 실험하였으며, 설정된 장애물과의 안전 거리는 1m로 설정하였다. 로봇이 목표 지점으로 이동 중 설정된 장애물을 감지하였을 때는 해당 장애물과 안전 거리를 유지하며 멈추고, 장애물이 제거될 때까지 대기한 후 경로를 계속 추종하여 목표 지점으로 이동하였다. 설정되지 않은 장애물의 경우, 로봇은 동적으로 경로를 재 계획하여 목표 지점까지 이동하였다.

표 4는 로봇이 10회의 자율주행을 수행할 때 이동 명령에 따른 도착 지점의 오차 및 거리유지 성능을 나타낸다.

거리유지(Keep Distance)에 대한 실험에서는 안전 거리를 1m로 설정하고, 이 거리 내에서의 오차를 측정하였다. 실험 결과, 설정된 거리에 대비하여 실제로 측정된 거리는 평균 약 0.038m의 오차를 보였다. 위치정확도(Location Accuracy)에 대한 실험에서는 a,b,c 세 지점을 이동하며, 각 지점에 도착할 때

마다 /amcl_pose 토픽을 통해 실제 로봇의 위치 및 방향 좌표값을 확인하였다. 실시간 위치 및 방향 좌표값과 미리 설정한 좌표값을 비교하여 오차를 계산하였다. 계산된 오차값들의 평균은 약 5.7252%로 나타났다.

표 4. 자율 주행 실험 결과
Table 4. Autonomous driving experiment results

Test	Keep distance error (m) (Set distance = 1m)	Location accuracy error (%)
a->b	0.037	6.1048
b->a	0.029	5.3356
a->c	0.037	6.4017
c->b	0.044	5.5126
b->c	0.043	5.7001
c->a	0.036	6.2112
a->b	0.044	6.1949
b->a	0.031	5.5128
a->c	0.040	5.3652
c->a	0.037	4.9131
average	0.038	5.7252

V. 결론 및 향후 연구

본 논문에서는 복분자 재배 환경 특성을 고려하여 LiDAR 센서와 YOLOv5 모델을 기반으로 한 실시간 객체 검출을 사용해 자율주행 시스템을 구현하였다. SLAM을 활용하여 맵을 생성하였으며, 사용자로부터의 키보드 입력을 통해 로봇은 지정된 경로로 이동하였다. 이동 중에는 LiDAR와 YOLOv5 기반의 실시간 객체 검출을 활용해 설정한 장애물을 실시간으로 인식하고, 거리 값을 수집하였다.

진행된 10회의 자율주행 테스트 결과 거리 유지 실험에서 로봇은 설정된 1m의 거리를 유지하기 위해 약 0.038m의 평균 오차로 움직이는 결과를 보였다. 또한, 위치정확도 실험에서 로봇은 세 지점 a, b, c로 이동하며 해당 도착 지점에서의 위치 및 방향 오차를 계산하여, 평균 5.7252%의 오차로 움직였다는 결과를 나타냈다. 이러한 결과는 단순히 목적지 까지 장애물 인식 및 회피 기능뿐만 아니라, 좁은 복분자 재배 사로에서 장애물과의 안전한 거리를 유지하면서도 사용자가 원하는 위치에 도달할 수 있었다.

향후 연구에서는 실제 복분자 재배 환경에 적합한 수확물 운반 도우미 로봇에 본 논문에서 개발한 자율주행 시스템을 적용하여 실험을 진행할 계획이다. 이 과정에서 발생할 수 있는 다양한 문제점을 파악하고, 그에 따른 해결책과 개선 방안을 연구할 계획이다.

References

- [1] S. S. Lee "Agricultural, Forestry, and Fisheries Survey Results", KOSTAT, Apr. 2022.
- [2] D. H. Lee and G. H. Kim, "Current and Future Robot Technologies in Open-field Agriculture", The Journal of The Korean Institute of Communication Sciences, Vol. 39, No. 10, pp. 13-18, Sep. 2022.
- [3] K. B. Lee, et al., "Implementation of 3D Location Detection Embedded System for Tomato Harvesting Robots", The Journal of Korean Institute of Communications and Information Sciences, Vol. 47, No. 11, pp. 2007-2019, Nov. 2022. <https://doi.org/10.7840/kics.2022.47.11.2007>.
- [4] S. Fountas, et al., "Agricultural Robotics for Field Operations", Sensors, Vol. 20, No. 9, May 2020. <https://doi.org/10.3390/s20092672>.
- [5] J. Yucheng, L. Jizhan, X. Zhujie, Y. Shouqi, L. Pingping, and W. Jizhang, "Development status and trend of agricultural robot technology", International Journal of Agricultural and Biological Engineering, Vol. 14, No. 4, Jul. 2021. <https://doi.org/10.25165/j.ijabe.20211404.6821>.
- [6] S. J. Park, "A Study on Transport Robot for Autonomous Driving to a Destination Based on QR Code in an Indoor Environment", Journal of platform technology, Vol. 11, No. 2, pp. 26-38, Apr. 2023.
- [7] S. B. Baek, B. S. Kim, J. W. Lee, and H. C. Moon, "Real-time Path Planning Algorithm based on Ridge Detection for Solid-State LiDAR-based Autonomous Agricultural Robot", Journal of Institute of Control, Robotics and Systems, Vol. 28, No. 2, pp. 146-152, Feb. 2022. <https://doi.org/10.5302/J.ICROS.2022.21.0207>.
- [8] J. S. Kim, Y. T. Ju, and E. K. KIM E, "Object Recognition Technology using LiDAR Sensor for Obstacle Detection of Agricultural Autonomous Robot", The Journal of The Korea Institute of Electronic Communication Sciences, Vol. 16, No. 3, pp. 565-570, Jan. 2021.
- [9] H. J. Hwang, M. H. Park, D. Y. Kim, and Y. T. Kim, "A Study on Implementation of Autonomous Navigation Robot System in Unity-Based Digital Smart Farm", Journal of KIIS, Vol. 32, No. 6, pp. 507-514, Dec. 2022. <https://doi.org/10.5391/JKIIS.2022.32.6.507>.
- [10] P. M. Blok, H. K. Suh, K. V. Boheemen, H. J. Kim, and G. H. Kim, "Autonomous In-Row Navigation of an Orchard Robot with a 2D LIDAR Scanner and Particle Filter with a Laser-Beam Model", Journal of Institute of Control, Robotics and Systems, Vol. 24, No. 8, pp. 726-735, Aug. 2018. <https://doi.org/10.5302/J.ICROS.2018.0078>.
- [11] J. Saike, W. Shilin, Y. Zhongyi, Z. Meina, and L. Xiaolan, "Autonomous Navigation System of Greenhouse Mobile Robot Based on 3D Lidar and 2D Lidar SLAM", Frontiers in Plant Science, Vol. 13, Mar. 2022. <https://doi.org/10.3389/fpls.2022.815218>.
- [12] S. H. Byeon and J. B. Park, "A Resampling Method to Improve the Performance of AMCL Under a Few Features Environment", in Proceedings of The Korean Institute of Electrical Engineers Workshop, pp. 162-165, Apr. 2021.
- [13] Y. S. Jeong, "Trends in SLAM Technology for Autonomous Smart Robots", The Journal of The Korean Institute of Communication Sciences, Vol. 40, No. 5, pp. 40-46, Apr. 2023.
- [14] L. Zhi and M. Xuesong, "Navigation and control system of mobile robot based on ROS", In 2018 IEEE 3rd Advanced Information Technology, Electronic and Automation Control Conference

(IAEAC), Chongqing, China, pp. 368-372 Oct. 2018. <https://doi.org/10.1109/IAEAC.2018.8577901>.

[15] J. M. Kang, "A Study on a Wildlife Detection and Classification Model using YOLOv5", The Journal of Korean Institute of Information Technology, Vol. 20, No. 10, pp. 21-27, Oct. 2022. <https://doi.org/10.14801/jkiit.2022.20.10.21>.

저자소개

김 기 오 (Gio Gim)



2017년 3월 ~ 현재 : 군산대학교
소프트웨어학부 학사과정
관심분야 : 자율주행, 사물인터넷,
엣지 컴퓨팅

온 병 원 (Byung-Won On)



2007년 : 펜실베이니아주립대학교
컴퓨터공학과 박사
2008년 : 브리티시컬럼비아대학교
박사후연구원
2010년 : 일리노이대학교
ADSC센터 선임연구원
2011년 : 서울대학교

차세대융합기술연구원 연구교수
2014년 ~ 현재 : 군산대학교 소프트웨어학부 교수
관심분야 : 자연어처리, 인공지능, 강화학습

이 석 훈 (Sukhoon Lee)



2009년 2월 : 고려대학교
전자및정보공학부(학사)
2011년 2월 : 고려대학교
컴퓨터·전파통신공학과(공학석사)
2016년 2월 : 고려대학교
컴퓨터·전파통신공학과(공학박사)
2016년 3월 ~ 2017년 3월 :

아주대학교 의료정보학과 연구강사
2017년 4월 ~ 현재 : 군산대학교 소프트웨어학부 교수
관심 분야 : 사물인터넷, 엣지 컴퓨팅, 데이터 품질,
시맨틱웹

이 연 식 (YonSik Lee)



1982년 2월 : 전남대학교
전자계산학과(이학사)
1984년 2월 : 전남대학교
전자계산학과(이학석사)
1994년 2월 : 전북대학교
전산응용공학전공(공학박사)
1986년 3월 ~ 현재 : 군산대학교

소프트웨어학부 교수
관심분야 : 클라우드 컴퓨팅, 능동규칙 시스템, 에이전트
시스템, S/W미들웨어

정 현 준 (Hyeonjun Jung)



2008년 : 삼육대학교
컴퓨터과학과(학사)
2010년 : 숭실대학교
컴퓨터학과(공학석사)
2017년 : 고려대학교
컴퓨터·전파통신공학과(공학박사)
2017년 8월 ~ 2020년 8월 :

광주과학기술원 블록체인인터넷경제연구센터 연구원
2021년 ~ 현재 : 군산대학교 소프트웨어학부 교수
관심분야 : 블록체인, 데이터 사이언스, 센서 네트워크,
사물인터넷, 머신러닝

정 동 원 (Dongwon Jeong)



1997년 2월 : 군산대학교
컴퓨터과학과(학사)
1999년 2월 : 충북대학교
전자계산학과(석사)
2004년 2월 : 고려대학교
컴퓨터학과(박사)
2005년 4월 ~ 현재 : 군산대학교

소프트웨어학부 교수
관심분야 : 데이터베이스, 시맨틱 서비스, 빅데이터,
사물인터넷, 엣지컴퓨팅, 지능형 융합 서비스