

논문 2023-18-31

SLAM 기술을 활용한 저가형 자율주행 배달 로봇 시스템 개발

(Development of Low Cost Autonomous-Driving Delivery Robot System Using SLAM Technology)

이 동 훈, 박 제 현, 정 경 훈*

(Donghoon Lee, Jehyun Park, Kyunghoon Jung)

Abstract : This paper discusses the increasing need for autonomous delivery robots due to the current growth in the delivery market, rising delivery fees, high costs of hiring delivery personnel, and the need for contactless services. Additionally, the cost of hardware and complex software systems required to build and operate autonomous delivery robots is high. To provide a low-cost alternative to this, this paper proposes a autonomous delivery robot platform using a low-cost sensor combination of 2D LIDAR, depth camera and tracking camera to replace the existing expensive 3D LIDAR. The proposed robot was developed using the RTAB-Map SLAM open source package for 2D mapping and overcomes the limitations of low-cost sensors by using the convex hull algorithm. The paper details the hardware and software configuration of the robot and presents the results of driving experiments. The proposed platform has significant potential for various industries, including the delivery and other industries.

Keywords : Delivery Robot, Visual SLAM, Autonomous Driving Robot, Mapping

1. 서 론

최근 배달 시장의 호황 및 배달 인력 부족으로 인해 상승한 배달비, 고물가 및 비대면 서비스의 필요성으로 인해 자율주행 배달 로봇의 필요성이 대두되고 있다 [1, 2]. 배달 로봇 시장은 2021년 2백만 달러에서 2026년까지 연평균 35.1% 성장할 것으로 전망되고 있다 [3]. 자율주행 배달 로봇은 사람 없이 주행하고 상품을 효율적으로 목적지까지 전달할 수 있는 능력으로 인해 국내 및 해외 배달업에서 주목받고 있다.

실제로 2019년 10월 서울 송파구 우아한 형제들 본사에서 진행된 시범 운영에서 배달원들은 배달 로봇 도입으로 인해 배달 시간을 약 5~16분가량 절약할 수 있었다고 한다 [4]. 이는 자율주행 배달 로봇을 통해 한정된 시간 내에서 더 많은 주문량을 배송할 수 있는 셈이다. 이러한 동향 속에서 푸드테크 배달 로봇 산업이 더욱 활발하게 진행될 것으로 보인다.

그러나 자율주행 배달 로봇은 고가의 하드웨어와 복잡한 소프트웨어 시스템을 필요로 하여, 제작 및 운영 비용이 매우 높다는 문제점이 있다. 특히, 고가의 3D LIDAR를 필요로 하는 배달 로봇의 특성상 해당 가격은 평균 4,000~5,000만 원을 필요로 한다 [5]. 실제로 미국 Serve Robotics의 차세대 배달 로봇은 8000달러의 Ouster OS1 3D LiDAR를 장착할 계획이라고 밝혔다 [6, 7]. 또한, 중국 알리바바의 배달로봇 Xiaomanlv의 경우 3350유로에 판매 중인 고가의

RS-Helios-5515 3D LIDAR를 탑재하고 있다 [8, 9].

이에 대한 대안으로, 본 논문은 Open Source 소프트웨어와 저가의 하드웨어를 활용한 자율주행 배달 로봇 플랫폼 설계 방법을 제안한다. 본 논문에서 제시하는 자율주행 배달 로봇은 제한된 구역 내에서 최적 경로를 찾고, 동적 장애물을 회피하기 위해 RTAB-Map (Real-Time Appearance-Based Mapping) SLAM (Simultaneous Localization And Mapping) [10] Open Source Package를 활용하여 구현되었다. 2013년 Mathieu Labbe에 의해 개발되어 출시된 RTAB-Map은 loop-closure detection 기반 Visual SLAM 방식을 채택한다. 이는 graph-based 접근 방식을 사용하여 실시간 주변 환경 mapping 및 loop-closure detection을 수행하여, odometry의 미끄러짐을 보완하고, 실제 환경에서 현재 위치와 이전에 방문했던 위치를 비교하고 판단하여 로봇의 표류 문제를 방지할 수 있다. 이러한 이점으로 인해 RTAB-Map은 mapping 및 localization이 필요한 배달로봇 및 드론 등을 제작하는 데 강력한 도구로 활용될 수 있다.

또한, 기존 자율주행 배달 로봇 구성에서 가장 큰 비용을 차지하는 3D LIDAR를 대체하기 위해 2D LIDAR, Depth Camera, Tracking Camera의 센서 조합을 활용한 자율주행 배달 로봇 플랫폼 설계법을 제안한다.

본 논문은 2장 하드웨어 구성에서 로봇의 전체적인 구성과 탑재되는 각 센서에 관해 설명하고, 3장 소프트웨어 구성에서 전체 흐름도 및 각 알고리즘에 관해 설명한다. 또한, 4장에서 주행 실험을 통한 검증 결과를, 5장에서 결론 및 추후 개선 방향을 다루고 마무리한다.

*Corresponding Author (khjung@kookmin.ac.kr)

Received: Apr. 7, 2023, Revised: May 11, 2023, Accepted: Jul. 11, 2023.
D. H. Lee, J. H. Park: Kookmin University (B.S. Student)
K. H. Jung: Kookmin University (Prof.)

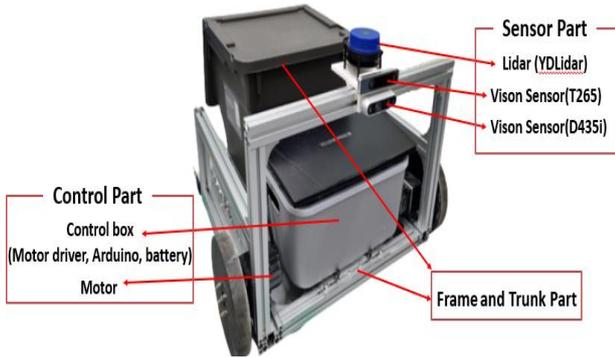


그림 1. 배달 로봇 플랫폼
Fig. 1. Delivery robot platform

II. 하드웨어 구성

하드웨어 설계 과정에서 실내 및 실외 구동을 위한 최소한의 부품을 사용하도록 구성하였고 기존 배달 로봇에 탑재되는 부품중 가장 높은 가격의 센서인 3D LIDAR의 역할을 2D LIDAR와 Depth Camera로 대체하였다. 전체 부품 가격은 2,099,200원으로, 3D LIDAR를 탑재하는 기존 배달로봇에 비해 약 90~95% 이상의 비용 절감을 의미한다.

하드웨어는 그림 1과 같이 Frame and Trunk part, Sensor part, Control part 3부분으로 나눌 수 있다.

1. Frame and trunk part

본 논문에서 제시하는 배달 로봇의 외관은 기본적으로 알루미늄 프로파일을 가공하여 frame을 제작하였으며, 3D printer를 활용하여 센서 거치 지지대를 추가 설계 후 조립하였다. 알루미늄 프로파일로 제작된 Frame은 로봇의 하중을 지탱하며 로봇의 외관 형성을 담당한다. 배달 물품 적재를 위한 리빙박스 및 keypad 등의 기타 부품들은 Frame에 용접하여 고정된 아크릴판 위에 장착되었다. 또한, trunk part (수납부)는 설계의 단순화와 비용 절감을 위해 두 개의 리빙박스를 활용하였다. 하나는 시스템 구동에 필요한 배터리, 모터 드라이버, PC 등을 수납하는 공간이며, 하나는 배달 물품을 수납할 수 있는 공간이다.

2. Sensor part

본 논문에서 제안한 자율주행 배달 로봇 시스템에서 활용한 sensor part는 배달 로봇의 전방 가장 높은 곳에 설치되어 로봇의 형상에 간섭받지 않도록 설계하였으며, mapping과 주행을 위한 point cloud 데이터 수집 및 odometry 추적 역할을 담당한다. Sensor는 YDLIDAR X4와 Depth Camera D435i, Tracking Camera T265로 구성되어 기존 배달 로봇의 3D LIDAR 역할을 대체한다.

그림 2의 YDLIDAR X4 모델은 2차원 LIDAR로, 로봇 탐색 및 장애물 회피를 위한 2D laser Scan 기능과 빠른 통신을 지원한다. 수평 방향으로 최대 10m 거리에 있는 장애물

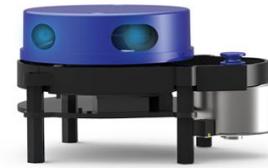


그림 2. 2차원 라이다 YDLIDAR X4
Fig. 2. 2D LIDAR YDLIDAR X4



그림3. 깊이 카메라 D435i
Fig. 3. Depth Camera D435i



그림 4. 추적 카메라 T265
Fig. 4. Tracking Camera T265

을 5,000Hz의 속도로 인식할 수 있다 [11]. 본 논문에서 제시하는 배달 로봇 주변의 2D laser scan을 투사하여 정적, 동적 장애물 간의 거릿값을 획득하는 역할을 담당한다.

그림 3의 Intel RealSense Depth Camera D435i는 Intel사의 depth 카메라로, 수직 방향 86° 및 수평 방향 57°의 화각 (FOV)으로 정적 IR 패턴을 전방에 투사한 뒤 패턴의 왜곡으로부터 depth를 추정하여 최대 3m까지 인지할 수 있다 [12]. 주행 시 얻어지는 RGB-D 이미지는 point cloud로 변환되어 mapping 과정에서 활용된다.

그림 4의 Intel RealSense Tracking Camera T265는 Intel사의 odometry 추적 카메라로, IMU를 내장해 관성을 측정하여 Visual SLAM에서 높은 정확도를 보인다 [13]. 양측에 달린 스테레오 카메라를 통해 특징점 변화를 추적하고, 카메라의 순간 속도를 추정하고 누적하여 배달 로봇의 odometry 값을 얻을 수 있다.

3. Control part

배달 로봇은 표 1과 같이 Intel NUC 10i5FNHN PC 및 PNT50 모터 드라이버, Arduino Mega 2560에 의해 제어된다. 또한, 사용자로부터 목적지 정보를 입력받기 위해 keypad를 사용하였다. 먼저, 사용자가 keypad로 목적지에 대한 호실 혹은 건물 번호를 입력하면, 코드 내부 조건문을 거쳐 로봇에 미리 저장된 Map 내부의 global 좌표로 치환된 후, RTAB-Map의 Navigation System에게 변환된 좌표를 전달하는 방식을 활용하였다.

하지만 Keypad로 목적지를 입력하는 방식은 사용자가 목적지에 대한 정보를 미리 알고 있지 않다면 혼동할 수 있다

표1. 제어부 구성

Table 1. Control part configuration

Components	Model	Size (mm)
Main Controller	Intel NUC 10i5FNHN	117x112x51
Motor Driver	PNT50	98x82x32
Destination Input	Number keypad	131x89x21
LCD	LCD 16x2 I2C	80x36x18
Input Controller	Arduino 2560	115x53x10

는 한계점이 존재한다. 따라서 이러한 한계점을 개선할 방안으로 지도를 터치하여 목적지를 입력할 수 있는 GUI가 포함된 터치패드를 장착하여 개선할 예정이다.

본 논문에서 제시하는 배달 로봇 시스템은 Ubuntu 18.04 운영체제 기반 ROS Melodic을 내장한 Intel 사의 컴퓨팅 장치인 Intel NUC 10i5FNHN로, 최대 4.2GHz의 속도와 4코어, 8스레드의 프로세서를 탑재하여 RTAB-Map 구동에 적절한 성능을 가졌다 [14]. Intel NUC에 내장된 navigation system은 Arduino로부터 전달받은 global 좌표와 RTAB-Map으로부터 전달받은 2D map을 활용하여 목적지까지의 최적 경로를 계산한다. 이후 LIDAR로부터 검출된 2D laser scan 데이터를 활용해 현재 위치를 인식하고, 동적 및 정적 장애물을 회피할 수 있는 RPM 값을 계산하여, PNT50 모터 드라이버에게 전송한다.

PNT50은 MDROBOT사에서 제작한 모터 드라이버로 작은 전류와 전압을 제어하는 PC에서 계산한 RPM을 전송받아 K6LH40N2-06 모터 [15]에 전달하는 통신부 역할을

담당한다. 또한, PNT50에서 지원하는 저/과전압 및 과열 관리 기능으로 모터를 보호하고, RS485 통신을 통해 계산된 RPM 값을 모터에 전달하였다 [16].

III. 소프트웨어 구성

1. 프레임워크

배달 로봇의 소프트웨어 프레임워크는 그림 5와 같이 Visual SLAM (RTAB-Map) package, navigation system, motor control 세 단계로 구성되어 있다.

각 프로세스 간 통신을 위해서 ROS (Robot Operating System) 및 RS485, Serial 통신을 활용했다. 먼저, T265를 통해 odometry 데이터를 얻어냈다. T265는 내부적으로 visual odometry 정보와 IMU odometry 데이터를 조합하고, PC로 반환하여 주기 때문에 데이터를 가공하지 않고 그대로 사용할 수 있다. 이후 D435i를 통해 얻은 depth 및 RGB 정보가 담겨있는 point cloud 데이터, YDLIDAR X4를 통해 얻어지는 2D laser scan 데이터를 RTAB-Map에 전송하고 건물 내부 map을 제작했다.

Navigation system은 ROS의 move_base package [16]를 활용해 로봇의 위치와 목표 지점 간의 최적 경로를 계산하고, 로봇이 해당 경로로 주행하도록 설계하였다.

이때 로봇에 목적지를 입력할 수 있도록 숫자 keypad와 Arduino를 이용해 목적지 (숫자)를 입력하면 이를 좌표로 변환한 뒤 ROS를 통해 navigation system 내부 move_base package에 전달하여 로봇이 목적지까지 이동하도록 설계했

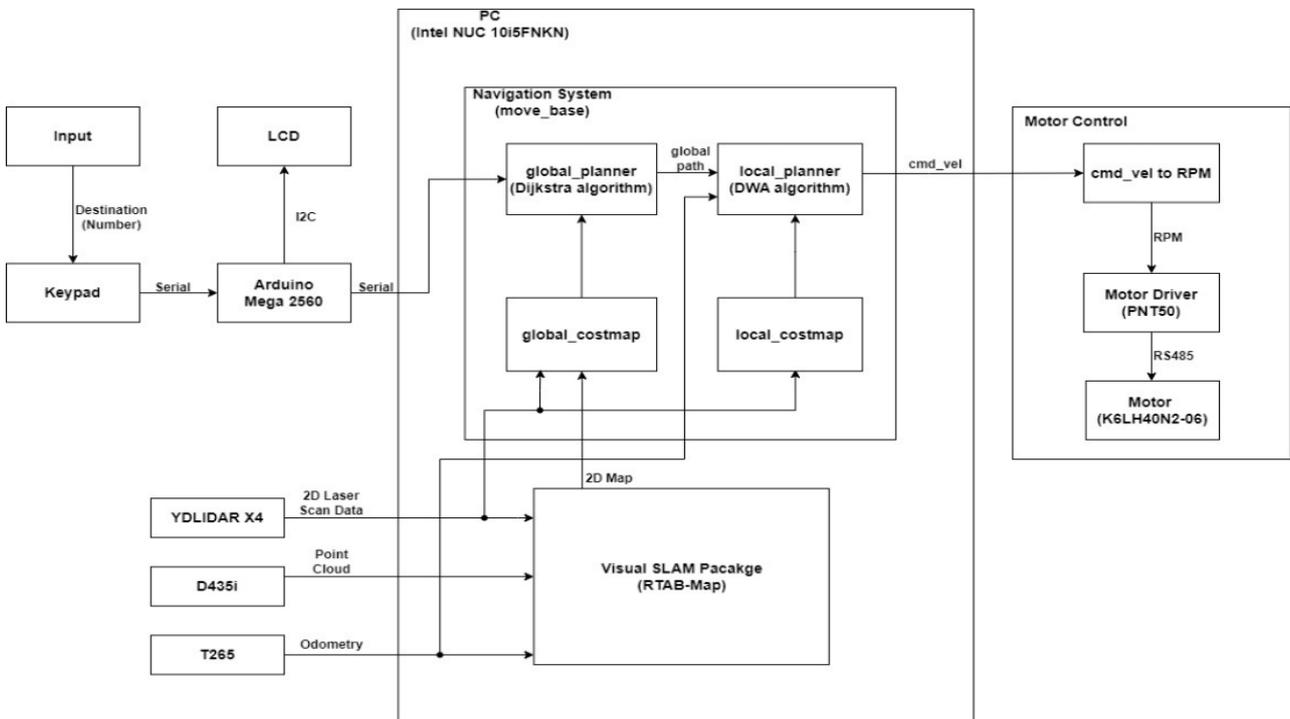


그림 5. 배달 로봇 설계 구성 및 제어 흐름도

Fig. 5. Delivery Robot design composition and Control Flowchart

다. 이후 move_base package를 통해 얻은 데이터를 RPM 값으로 변환하여 모터 드라이브에 전달하여 모터를 제어하는 방식을 채택했다.

2. Visual SLAM(RTAB-Map)

본 논문에서 제시하는 배달 로봇은 Visual SLAM을 위해 open source package 중 하나인 RTAB-Map package를 사용하였다.

2.1 Mapping

배달 로봇이 T265를 통해 로봇의 위치를 지속해서 추정해 가고, D435i로 획득한 point cloud 데이터를 로봇의 위치가 변할 때마다 기록하여 그림 6과 같이 3D mapping 후 2D map으로 변환을 진행하였다. T265와 D435i를 조합하여 3D map을 만들 때, 완전한 3D map을 만들기 위해서는 센서가 mapping하고자 하는 구역을 빈틈없이 탐지해야 한다.

하지만 건물 외부 벽면과 같은 구조는 특징점이 적기 때문에 mapping이 제대로 진행되지 않아 특정 구역은 빈 구역으로 탐지되거나 noise가 발생하는데, 이렇게 제작된 map을 통해 navigation을 진행하면 그림 7의 (a)와 같이 해당 구역이 벽면임에도 불구하고 벽면을 관통하는 경로를 생성

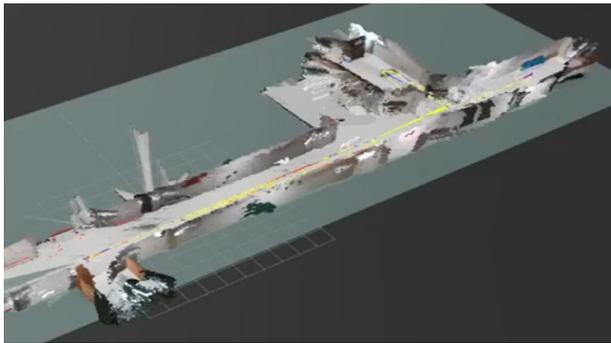


그림 6. 깊이 카메라로 작성한 제한된 구역의 3차원 지도 Fig. 6. 3D mapping of restricted area by depth camera

할 수도 있다. 따라서 이를 보정하기 위한 YDLIDAR X4의 2D laser Scan 기능이 필요하다. 시야각에 제한이 있는 D435i와는 달리 YDLIDAR X4는 360°로 주변 정보를 탐지할 수 있으므로 수월한 주행을 위해 탑재하여 그림 7의 (b)와 같은 경로를 생성할 수 있다.

2.2 2D map 개선

주행 및 mapping과정에서 YDLIDAR X4를 통해 얻은 point cloud data와 2D map을 매칭 하는 과정에서 그림 8과 같이 뚫려있는 난간 혹은 계단은 2D LIDAR를 이용하여 감지하기 어렵다. 따라서 이런 구역에서 mapping을 진행하면 그림 9의 (a)와 같이 불완전한 map이 생성된다.

이러한 구역에서 주행하기 위해서 PGM 파일 형식으로 저장되는 2D map을 그림 9의 (b)와 같이 Convex hull 알고리즘 [17]을 적용하여 완성된 2D map을 제작하였다. Convex hull 알고리즘은 2차원 공간에서 주어진 점 집합을 둘러싸는 가장 작은 볼록 다각형을 구하는 알고리즘이다. 그림 9의 (a)에서 감지된 모든 point cloud들을 둘러싸는 다각형을 만들어 (b)와 같이 수정된 2D map은 map_server ROS package [18]를 이용하여 navigation system에 전송할 수 있다.

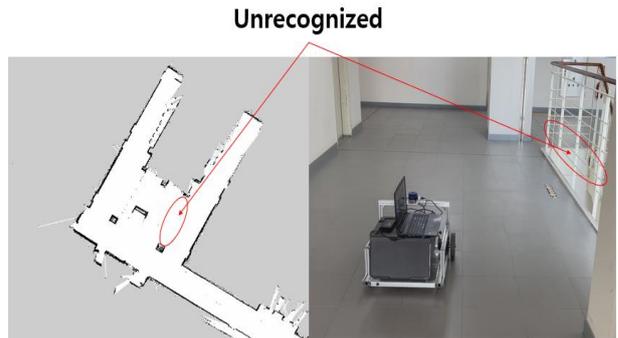


그림 8. 난간에 의한 이차원 지도 왜곡 Fig. 8. 2D map distortion cause of handrail



그림 7. 부정확한 지도 작성으로 발생한 경로 오류 (a) 시작점 (S)로부터 도착점 (E)까지의 잘못된 경로 (b) 시작점 (S)로부터 도착점 (E)까지의 올바른 경로

Fig. 7. Path error caused by incorrect mapping. (a) Incorrect path from starting point (S) to destination (E), (b) Correct path from starting point (S) to destination (E).

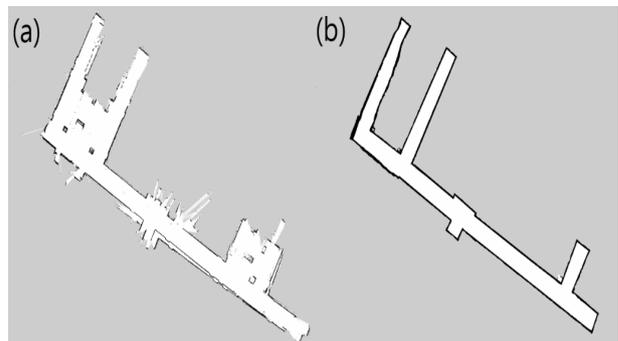


그림 9. (a) 라이다 레이저 스캔 데이터로 제작된 2차원 지도 (b) 미인식 영역의 컨벡스 헐 알고리즘에 보정된 이차원 지도 Fig. 9. (a) 2D map using laser scan data of LIDAR, (b) Calibrated 2D map with Convex hull for unrecognized areas.

3. Navigation system

Navigation system에서는 cost map을 통해 경로를 생성하게 된다. Global cost map을 통해 global 경로를 생성하고, 로봇은 이를 따라 주행하되 local cost map을 통해 장애물을 감지하면 해당 장애물을 우회하는 새로운 경로를 생성하며 움직이게 된다. Cost map으로부터 경로 계획을 수행하기 위해 본 논문에서는 global planner로 Dijkstra 알고리즘 [19]을, local planner로 DWA 알고리즘 [20]을 활용했다.

3.1 Loop-closure detection

배달 로봇이 T265를 통해 대략적인 자기 위치를 추정할 때, odometry 누적을 통해 위치를 추정하기 때문에 장시간 위치를 추정하면 오차가 발생할 수밖에 없다. Visual SLAM에서는 이러한 오차를 해결하기 위해 loop-closure detection을 사용한다 [21]. 사진에 만들어진 map을 활용하면 LIDAR 및 depth camera에서 odometry 누적으로 추정된 로봇의 현재 위치에 어떤 point cloud 데이터가 들어와야 하는지 예측할 수 있다. 만약 실제로 받는 데이터와 예측된 데이터의 오차가 크다면 로봇의 위치가 올바르게 추정되고 있지 않다는 뜻이고, 올바른 로봇의 위치가 어디인지 다시 탐색하게 된다. 이때 map에서 현재 수신 중인 데이터를 얻을 수 있는 지점이 어디인지 탐색 및 예측하게 되며, 예측된 지점에서 저장된 point cloud 정보와 현재 센서가 받는 데이터가 일치하지 이상 일치한다면 해당 위치로 로봇의 위치를 바로잡는다.

예시로, loop-closure detection에서는 그림 10과 같이 센서를 통해 (a)의 주행 경로를 예측하였다 하더라도, 좌측 (Pose 807 및 Pose 1098)과 같은 센서 데이터를 얻었다면 이를 통해 예측된 주행 경로 (a)를 (b)와 같이 수정하는 방식을 채택한다. 본 논문에서는 D435i depth 카메라를 loop-closure detection을 위한 장비로 활용했으며, depth 카메라로부터 받은 이미지의 특징점과 depth 정보를 조합하여, 생성한 map 상에서 현재 위치 및 주행 경로를 추정했다.

3.2 Global cost map

Navigation system에서 경로 계획을 위해 ROS의 navigation package 중 하나인 move_base 명령을 활용하였다. 로봇이 목적지까지 안전하게 도달하려면, 벽면과 같은 정적 장애물을 회피할 수 있도록 경로를 생성해야 한다. 이때, Navigation system에서 경로를 결정하는 과정에서 cost map을 사용한다. 여기서 global cost map은 map 전체에 대한 cost를 계산한 것이다. 장애물에 대해 cost가 생성될 때 어느 범위까지 생성되도록 하는지는 navigation system의 내부 hyper parameter로 설정할 수 있으며 본 논문에서는 배달 로봇 크기의 2배 범위까지 점진적으로 감소하면서 생성되도록 설계하였다.

3.3 Local cost map

Global cost map이 전체 정적 map에 대한 cost map이라면, local cost map은 주변 지역에 대한 실시간 cost map을 생성한 것이다. 이때 global map을 참고하지 않고 LIDAR 센서 데이터에만 의존하여 cost map을 생성하게 되며, map에 표시되지 않은 장애물에 대해서도 센서 데이터에 의해

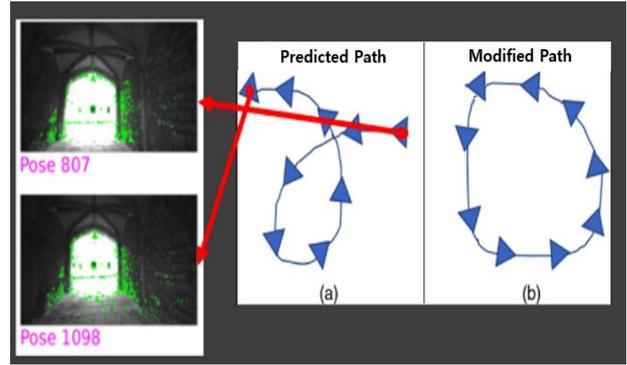


그림 10. 루프 폐쇄 검출의 예시
Fig. 10. Example of loop-closure detection

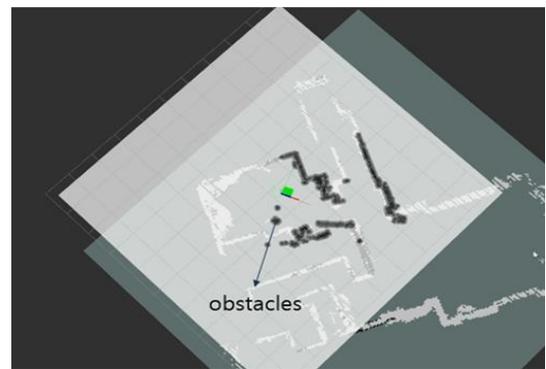


그림 11. 이차원 라이다를 이용한 국소 비용 지도
Fig. 11. Local cost map using 2D LIDAR

local cost map을 생성할 수 있다. 이때 local cost map을 생성하기 위한 센서로 2D LIDAR를 사용하게 된다.

그림 11에서 흰 영역은 local cost map의 범위이며, 이 범위에서 LIDAR가 감지한 장애물들은 검은색으로 표시된다. 배달 로봇은 흰 영역 내에서 검은색으로 표시된 장애물들을 회피할 수 있는 실시간 최적 경로를 따라 주행하기 위해 local cost map을 local planner로 전송한다. 이후 그림 11과 같이 local cost map과 global cost map을 교합시켜 전체 cost map을 구하고 이를 통해 local planner에서 최적 경로를 생성한다.

3.4 Global Planner

Global Planner는 그림 5와 같이 아두이노 serial 통신으로부터 전달받은 목적지와 global cost map을 토대로 global path를 생성하여 local planner에게 전달한다.

Navigation System의 global planner는 목적지까지의 최적 경로를 계산하기 위하여 Dijkstra 알고리즘을 활용하여 global path를 계산한다.

Dijkstra 알고리즘은 그래프에서 출발 지점부터 목적지까지의 최단 경로를 찾기 위해 사용되는 알고리즘이다 [19]. Global planner에서는 occupancy grid map 형태의 global cost map을 그래프로 나타낸 후 각 grid를 그래프의 노드로, 각 grid 간의 이동을 그래프의 간선으로 표현한다. Global planner에서는 이를 Dijkstra 알고리즘에 적용하여

목적지까지의 최단 경로를 계산하고, 출력된 global path를 local planner에게 전달한다.

3.5 Local Planner

Local Planner는 그림 5와 같이 odometry, local cost map을 토대로 global path를 따라 주행하되, 실시간 동적 장애물을 회피하는 local path를 생성하여 cmd_vel 형태로 Motor Control part에 전달한다.

본 논문에서는 Local Planner로 DWA (Dynamic Window Approach) 알고리즘을 활용하였다.

DWA 알고리즘은 로봇의 현재 상태와 주변 환경 정보를 바탕으로 로봇이 이동 가능한 윈도우를 계산하고, 이를 기반으로 최적 경로를 선택한다. 이때, 윈도우란 일정 주기 동안 변화 가능한 로봇의 속도 영역을 의미한다 [20].

Navigation system에서 DWA 알고리즘은 다음과 같은 순서로 동작한다 :

1. 로봇의 운동 상태 추정: 로봇의 현재 위치와 속도, 회전 등의 운동 상태를 추정한다. 이를 위해 t265의 odometry 정보를 활용한다.
 2. 이동 가능한 윈도우 계산: 로봇은 허용 가능한 속도와 회전 범위 내에서 주행해야 한다. 현재 로봇 상태와 local cost map을 기반으로 윈도우 영역을 계산한다. 이 윈도우는 로봇이 주행 가능한 속도와 회전 속도의 범위를 결정한다.
 3. 윈도우 점수 계산: 가능한 윈도우들에 대해 목적지와와의 거리, 경로의 정확성, 장애물 회피 등을 고려하여 각 윈도우에 점수를 부여한다. 점수는 로봇이 안전하게 목적지에 도달할 가능성을 나타낸다.
 4. 최적 경로 선택: 점수가 가장 높은 윈도우를 선택하여 로봇의 최적 경로로 결정한다.
- 이를 통해 최적의 local path를 생성하고, 이를 cmd_vel 형태로 Motor Control part에 전달하여 목적지까지 주행하는 동안 탐지되는 장애물을 회피할 수 있게 설계하였다.

4. Motor control

마지막으로 Motor control에서는 RTAB-Map 및 navigation system에서 계산한 경로로 주행하기 위한 모터 제어 역할을 담당한다.

PC에서 계산한 로봇의 속도 및 방향 정보를 담고 있는 cmd_vel message로 모터를 구동하기 위해 그림 12와 같이 선속도 및 각속도 값을 활용해 각 모터의 RPM 값을 구하고, Serial 통신을 통해 모터 드라이버에 RPM 제어 명령을 전송해야 한다.

$$RPM = GR \left(v_x + \frac{L}{2} w_z \right) \frac{60}{\pi D}. \quad (1)$$

식 (1)은 SLAM navigation으로부터 받은 cmd_vel을 모터 제어를 위한 RPM값으로 변환하는 수식이다. cmd_vel은 linear_x와 angular_z로 구성되므로 이를 모터에 적용하기 위해서는 식 (1)과 같이 RPM변환 연산을 수행해야 한다.

수식 (1)에서 GR은 기어비, v_x 는 직선 방향 속도, w_z 는 z

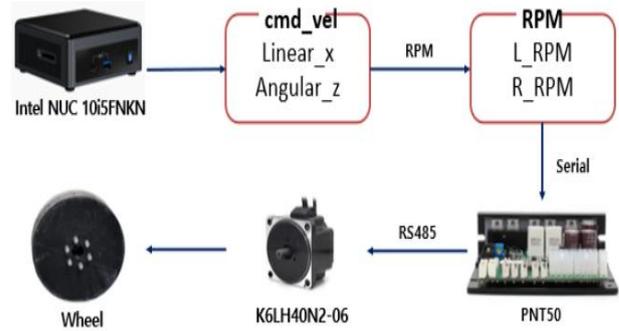


그림 12. 속도 입력으로부터 바퀴까지의 제어 흐름도
Fig. 12. Control flow diagram from the input of velocity to the wheels.

축을 기준으로 한 반시계 방향 각속도, L은 모터 사이의 거리, D는 바퀴의 반지름에 해당한다. 수식은 오른쪽 모터를 기준으로 한 것인데, 왼쪽 모터는 오른쪽과 반대방향으로 장착되므로 이 수식에 음수 부호 (-)를 취해 값을 구하게 된다. 이렇게 구한 각 모터의 RPM값을 모터 드라이버에서 바퀴가 장착된 모터에 전달해 주변 RTAB-Map 및 navigation part에서 계산한 움직임으로 정적, 동적 장애물들을 회피하며 목적지까지 주행할 수 있다.

IV. 실험 및 검증

앞서 설명한 하드웨어 및 소프트웨어를 활용한 배달 로봇의 자율주행 알고리즘을 검증하기 위해 mapping 및 주행 실험을 진행하였다.

실험 과정은 국민대학교 미래관 6층 실내 및 공학관에서 도서관까지의 캠퍼스 내에서 진행되었다.

실외 구역의 경우, 2D LIDAR를 활용했기 때문에 벽면이 존재하지 않는 구역의 경우 mapping 과정에서 준수한 성능을 도출하기 어려워, Depth Camera로 작성한 지도에 Convex hull 알고리즘을 적용한 map을 활용하여 주행하였다. 이러한 hand-crafted 기법을 적용하면 미리 mapping된 실외 공간에서도 주행 가능함을 확인할 수 있었다.

본 논문에서 제시한 자율주행 배달 로봇의 mapping 성능을 테스트하기 위해, 그림 13의 (a)에서 국민대학교 미래관 6층 내부 평면도와 로봇이 제작한 2D map (b)를 겹쳐 (c)와 같이 나타내었다. 이는 로봇이 제작한 2D map과 실측을 비교하였을 때 로봇이 자율주행하기 적절한 실내 mapping 성능을 도출함을 나타낸다. 또한, 그림 14에서 국민대학교 공학관 및 도서관 내부 실내 지도를 작성한 뒤 실외 지도 (초록색 영역)를 작성하여 교차한 것을 나타낸 것이다. 이를 통해 실외 환경에서도 배달 로봇이 스스로 자율주행하기 적절한 mapping 성능을 도출함을 확인할 수 있다. 실외의 경우, 하드웨어 특성상 평지만을 주행 가능하며 실내보다 noise가 크기 때문에 정적 장애물을 경계로 mapping 하는 것이 높은 주행 성능을 도출할 수 있다.

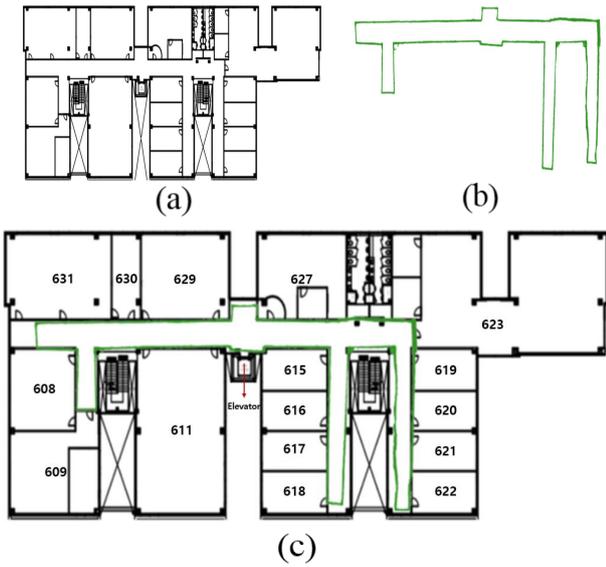


그림 13. 배달 로봇의 실내 지도작성 성능
 (a) 미래관 6층 평면도
 (b) 자율 주행 배달 로봇이 생성한 지도
 (c) (a)와 (b)를 교차한 지도.

Fig. 13. Indoor mapping performance of self-driving delivery robots. (a) The floor plan of the 6-th floor of mirae hall. (b) The indoor map produced by autonomous delivery robot. (c) The overlapping map of (a) and (b)

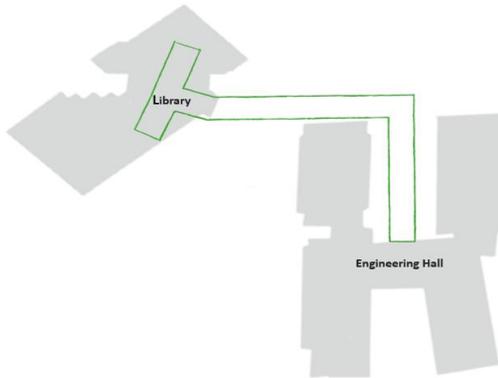


그림 14. 배달 로봇의 실외지도 작성 성능
 Fig. 14. Outdoor mapping performance of delivery robots.

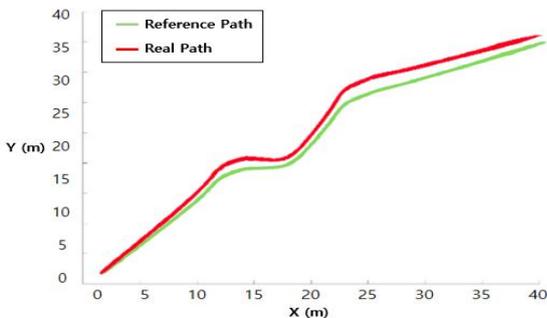


그림 15. 기준 경로와 실제 경로와의 비교
 Fig. 15. Comparison of the reference path and real path

표 2. 배달 로봇의 실내주행 실험 결과

Table 2. Indoor driving experiment result of delivery robot

	Starting point	Destination	Result	Distance error
1	Elevator	Room 620	Success	0.251 m
2	Room 615	Room 620	Success	0.328 m
3	Room 621	Room 617	Success	0.153 m
4	Room 608	Elevator	Success	0.277 m
5	Elevator	Room 609	Failure	1.532 m
6	Elevator	Room 609	Success	0.418 m
7	Room 609	Room 619	Success	0.341 m
8	Room 619	Room 617	Success	0.445 m
9	Room 617	Room 620	Success	0.241 m
10	Room 620	Room 615	Success	0.421 m

표 3. 배달 로봇의 실외주행 실험 결과

Table 3. Outdoor driving experiment result of delivery robot

	Starting point	Destination	Result	Distance error
1	Library	Engineering hall	Failure	0.843 m
2	Library	Engineering hall	Success	0.351 m
3	Library	Engineering hall	Success	0.417 m
4	Library	Engineering hall	Success	0.229 m
5	Library	Engineering hall	Success	0.384 m
6	Engineering hall	Library	Success	0.461 m
7	Engineering hall	Library	Failure	2.384 m
8	Engineering hall	Library	Success	0.292 m
9	Engineering hall	Library	Success	0.236 m
10	Engineering hall	Library	Success	0.485 m

그림 15에서 608호를 출발점으로 엘리베이터를 목적지로 입력했을 때 RTAB-Map에서 생성한 reference 경로와 로봇이 실제 주행한 경로를 비교하였다. 엘리베이터의 좌표와 실제 로봇이 도착한 좌표를 비교하였을 때, x축 및 y축을 기준으로 약 0.5m 내로 오차를 기록했다. 이는 실제 배달 로봇이 목적지까지 도달하여 물품을 전달하기에 적절한 주행 성능을 나타냄을 확인할 수 있다.

표 2의 실내주행 테스트에서, 출발지에서 사용자가 입력한 목적지까지 충돌 없이 주행하여 목적지 좌표까지 0.5m 내로 도달하였을 때를 성공 여부의 기준으로 하였다. 표 2는 실내주행 실험 결과를 표로 정리한 것이며, 10번의 주행 실험 횟수 중 9번을 목적지까지 주행 성공하여 90%의 성공률을 달성하였다. 5번 실험의 경우 매우 빠르게 움직이는 보행자는 회피하지 못하여 충돌한 경우로, 이는 실패로 기록했다. 실험마다 경로와 출발지 및 목적지를 달리하였지만, 낮은 오차 안의 범위에 도달하는 것을 확인할 수 있었다.

표 3에서, 표2과 마찬가지로의 기준으로 실외주행 실험을 진행하였다. 국민대학교 공학관과 도서관까지, 10번의 주행 중 8번의 주행 성공으로 80%의 주행 성공률을 확인할 수

있었다. 1번과 7번 실험의 경우 거리 오차가 0.5m를 초과하여 실패로 기록했다.

실내 실패 case는 여러 센서 데이터 및 큰 연산량에 의한 FPS (Frame Per Second) 부족으로 인해 발생하였다. 따라서 이를 개선하고 정확도 향상을 위해 코드 최적화 및 하드웨어 보완 등을 통해 개선할 예정이다.

실외주행의 경우, 실내와는 다르게 주행 경로가 평평하지 않아 odometry 오차가 상대적으로 크게 누적된다. 이를 보완하기 위해 로봇의 바퀴를 진동 및 충격에 강인한 바퀴로 대체하여 distance error를 향상시킬 수 있다.

V. 결론

본 논문에서는 최소한의 센서와 모듈로 제작할 수 있는 저가형 자율주행 배달 로봇 플랫폼의 설계 방법에 대해 제시하고, 실제 환경에서 원하는 목적지까지 주행하며 준수한 성능을 검증하였다.

기존의 고가 3D LIDAR를 대체하기 위해 2D LIDAR, Depth 카메라, Tracking 카메라의 저가 센서 조합과 Visual SLAM 기반 RTAB-Map에 활용한 저가형 자율주행 배달 로봇 플랫폼 설계 방법을 제안한다.

제안한 자율주행 배달 로봇의 성능을 검증하기 위해 대학 캠퍼스를 대상으로 실험을 수행한 결과, 복잡한 캠퍼스 내부를 적절하게 mapping 할 수 있었으며 정적 장애물뿐만 아니라 보행자와 같은 동적 장애물을 회피하면서 목적지까지 상품을 성공적으로 배달하는 것을 확인할 수 있었다.

또한, 외부 주행을 위해서 나무, 벽면과 같은 정적 장애물을 경계선 삼아 mapping 한 후, Convex hull 알고리즘을 적용하여 2D LIDAR 및 Depth 카메라의 한계점을 극복하였다.

그러나, 본 논문에서 제시한 자율주행 배달 로봇은 미리 mapping된 제한된 구역 내에서만 작동 가능하다는 한계점이 있으며, 평지가 아닌 계단과 같은 장애물이 있는 경로로는 주행할 수 없다는 한계점이 있다. 추후 이러한 한계를 보완하여 배달 로봇의 주행 가능 경로를 확장해갈 예정이다.

기존 배달 로봇은 높은 제작비용으로 인해 소상공인들에게 높은 진입 장벽이 되고 있다. 본 논문에서 제시한 저가형 자율주행 배달 로봇의 설계 방법을 활용하여 기존보다 저가의 구성으로 우수한 성능의 배달 로봇을 도입할 수 있기를 기대한다.

References

- [1] S. Srinivas, S. Ramachandiran, S. Rajendran, "Autonomous Robot-driven Deliveries: A Review of Recent Developments and Future Directions," *Transportation Research Part E: Logistics and Transportation Review*, Vol. 165, 2022.
- [2] S. I. Choi, D. H. Kim, J. Y. Lee, S. H. Park, B. S. Seo, B. J. Park, B. Y. Song, J. B. Kim, W. P. Yu, J. I. Cho, "Trends and Implications of Logistics/delivery Robots in the era of the 4th Industrial Revolution," *ETRI Electronics and Telecommunications Trends*, Vol. 34, No. 04, pp. 98-107, 2019 (in Korean).
- [3] J. E. Ryu, N. J. Baek, J. W. Jun, G. S. Cho, "A Study on the Development of Safety Requirements Using a Support Model by Raising the Supporting Muscles of a Delivery Robot," *Journal of the Korea Academia-Industrial Cooperation Society*, Vol. 22, No. 09, pp. 10-18, 2021 (in Korean).
- [4] Available on : <https://www.hani.co.kr/arti/economy/consumer/1003202.html>
- [5] Available on : <https://www.news1.kr/articles/?4464961>
- [6] Available on : <https://www.serverobotics.com/ouster-and-serve>
- [7] Available on : <https://ouster.com/blog/os1-32-high-resolution-low-cost-lidar-sensor/>
- [8] Available on : <https://www.robosense.ai/en/news-show-1561>
- [9] Available on : https://www.roscomponents.com/en/31_robotosense
- [10] M. Labbé, F. Michaud "RTAB-map as an Open-source Lidar and Visual SLAM Library for Large-scale and Long-term Online Operation," *J Field Robot*, Vol. 36, No. 2, pp. 416 - 446, 2019.
- [11] D. Hutabarat, M. Rivai, D. Purwanto, H. Hutomo, "Lidar-based Obstacle Avoidance for the Autonomous Mobile Robot," 2019 12th International Conference on Information & Communication Technology and System (ICTS), pp. 197-202, 2019.
- [12] E. Tsykunov, V. Ilin, S. Perminov, A. Fedoseev, E. Zainulina, "Coupling of Localization and Depth Data for Mapping Using Intel RealSense T265 and D435i Cameras," *arXiv*, Vol. 2004.00269, 2020.
- [13] P. Hausamann, C. B. Sinnott, M. Daumer, "Evaluation of the Intel RealSense T265 for Tracking Natural Human Head Motion," *Sci Rep* Vol. 11, No. 1, 2021.
- [14] Available on : <https://www.intel.co.kr/content/www/kr/ko/products/sku/214591/intel-nuc-10-performance-kit-nuc10i5fnhn/specifications.html>
- [15] Available on : <https://www.mdrobot.co.kr/BLDCMotorDCstore/?idx=264>
- [16] Available on : <https://www.mdrobot.co.kr/BLDCMotordriverDCstore/?idx=156>
- [17] M. M. Rahman, K. Ishii, N. Noguchi, "Optimum Harvesting Area of Convex and Concave Polygon Field for Path Planning of Robot Combine Harvester," *Intel Serv Robotics*, Vol. 12, pp 167-179, 2019.
- [18] S. J. Baek, A. H. Kim, J. W. Kim, "Implementation of Wheelchair Robot Applying SLAM and Global Path Planning Methods Suitable for Indoor Autonomous Driving," *IEMEK J. Embed. Sys. Appl.*, Vol. 16, No. 6, pp. 293-297, 2021.
- [19] E. W. Dijkstra, "A Note on Two Problems in Connection with Graphs," *Numerische Mathematic*, June, 1959.
- [20] D. Fox, W. Burgard, S. Thrun, "The Dynamic Window Approach to Collision Avoidance," *IEEE Robotics and Automation Magazine*, Vol. 4, No. 1, pp. 23-33, 1997.

[21] J. Y. Heo, H. G. Cho, N.J. Doh, "Precise Visual Loop Closure Detection for LiDAR-based SLAM," Summer Annual Conference of Institute of Electronics and

Information Engineers, Jeju, Korea, pp. 938-939. 2019 (in Korean).

DongHoon Lee (이 동 훈)



2023 Department of Automobile and IT Convergence in Kookmin University (B.S.)

2023~Robotics Program in Korea Advanced Institute of Science and Technology (M.S.)

Field of Interests: Robotics, Autonomous Driving, SLAM
Email: dh_lee99@naver.com

JeHyung Park (박 제 현)



2018~School of Mechanical Engineering, in Kookmin University (B.S.)

Field of Interests: Robotics, Autonomous Driving, SLAM, Humanoid
Email: super3714@kookmin.ac.kr

KyeongHoon Jung (정 경 훈)



1987 Seoul National University Department of Electronic Engineering (B.S.)

1989 Seoul National University, Department of Electronic Engineering (M.S.)

1996 Seoul National University Department of Electronic Engineering (Ph.D.)

Career:

1997~2004 Professor of Computer Science and Electronic Engineering, Handong University

2005~Professor, School of Electronic Engineering, Kookmin University

Field of Interests: Image processing, computer vision, 3D broadcasting

Email: khjung@kookmin.ac.kr