

탐사 및 엘리베이터 연계를 이용한 완전 자율 다층 실내 지도 구축 시스템

Fully Autonomous Multi-Floor Indoor Map Building System with Exploration and Elevator Interaction

노주형¹·강규리¹·김연찬²·심현철[†]

Juhyeong Roh¹, Gyuree Kang¹, Yeonchan Kim², D. Hyunchul Shim[†]

Abstract: This paper presents a fully autonomous multi-floor map-building system that enables a mobile robot to explore indoor environments without human intervention. The proposed framework builds multi-floor maps by generating efficient frontiers based on 3D LiDAR-IMU-based SLAM and multi-sensor fusion costmap generation, and by enabling elevator interaction through 4-DoF manipulator system that performs button pressing using an RGB-D camera. The system performs exploration using MPPI as a local planner and graph-based global planner, allowing reliable navigation even when direct paths are blocked. To improve exploration efficiency, a temporal decay term is incorporated into the frontier scoring process, which reduces redundant backtracking and shortens exploration time by approximately 33% compared to a baseline method. The system was validated in the KAIST N1 building across five floors (5F-9F), where it successfully completed continuous exploration and elevator transitions in 27 minutes with an elevator interaction success rate of 95%. These results demonstrate that the proposed system can robustly construct multi-floor indoor maps in real-world environments and serve as a scalable foundation for indoor service and delivery applications; a video demonstrating the complete experimental procedure is available at <https://youtu.be/doCVS7Weils>

Keywords: Autonomous Robot, Sensor Fusion, Frontier-Based Exploration, Elevator Interaction, Multi-Floor Mapping

1. 서론

최근 실내 자율주행 및 라스트마일(last-mile) 배송 서비스에 대한 수요가 증가함에 따라, 복잡한 실내 환경에서 안정적이고 효율적으로 주행 가능한 배달 로봇 기술의 필요성이 커지고 있

다^{1,2}. 이러한 로봇이 실내에서 안정적으로 서비스를 수행하기 위해서는, 실시간 센서 데이터를 기반으로 한 정확한 자기 위치 추정(localization)과 신뢰도 높은 지도의 구축이 필수적이다. 이를 위해 일반적으로 3D LiDAR 기반 SLAM (Simultaneous Localization and Mapping)을 활용하여 사전 지도(pre-built map)을 생성한 뒤, 운영 단계에서 센서 데이터와의 정합을 통해 위치를 추정한다³⁻⁵. 그러나 사전 지도 구축 과정이 전적으로 사람의 수동 조작에 의존한다는 점은 다음과 같은 근본적인 한계를 가진다. 첫째, 확장성의 문제이다. 건물이 많아질수록 지도 구축에 필요한 인력·시간·비용이 크게 증가하며, 환경 변화에 대응하기 위한 지도 재구축 역시 사람이 다시 수행해야 한다. 둘째, 품질 편차 문제이다. 수동 주행 과정은 조작자의 숙련도, 주행 속도, 센서 방향 등에 영향을 받아 누락 구간이나 잡음이 포함된 비일관적 지도가 생성될 수 있다. 셋째, 탐사 경로의

Received : Nov. 20. 2025; Revised : Jan. 13. 2026; Accepted : Jan. 26. 2026

※ This work was supported by the Technology Innovation Program (Development of drone-robot cooperative multimodal delivery technology for cargo with a maximum weight of 40kg in urban areas) funded by the Ministry of Trade, Industry & Energy (MOTIE), South Korea, under Grant RS-2023-00256794.

1. Ph.D. Student, Electrical Engineering, KAIST, Daejeon, Korea (0298no, fngb20@kaist.ac.kr)

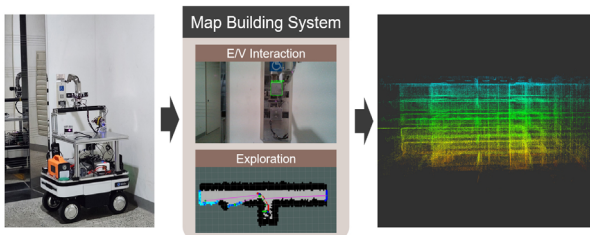
2. M.S. Student, Robotics Program, KAIST, Daejeon, Korea (kyc10206@kaist.ac.kr)

† Professor, Corresponding author: Electrical Engineering, KAIST, Daejeon, Korea (hcshim@kaist.ac.kr)

비효율성이다. 사람이 조종하는 경우 이미 충분히 스캔된 구역이나 막다른 지점으로 불필요하게 이동하여 지도 생성 관점에서 경로 효율이 저하될 수 있다.

이러한 한계를 극복하기 위한 시도로, 일부 연구에서는 엘리베이터를 활용한 다층 지도 구축 기법을 제안하였다⁶⁾. 그러나 해당 방식은 완전 자율 탐사 시스템이 아니며, 3개 층 규모의 제한된 환경에서 층 입구 주변만 부분적으로 스캔하는 데 그쳐, 실제 서비스 환경에 적용하기에는 여전히 제약이 존재한다. 또한, 엘리베이터와 연계하여 로봇의 자율적인 층간 이동을 시도한 연구들도 있다. 예를 들어, 한 연구⁷⁾에서는 엘리베이터 문이 열릴 경우 로봇이 탑승하여 버튼을 인식하고 누른 후 하차하는 과정만을 자율화하였으며, 엘리베이터 호출부터 탑승까지의 전 과정을 포함한 완전한 자동화는 이루어지지 않았다. 다른 연구⁸⁾에서는 전 과정을 자동화하였지만, 엘리베이터 버튼에 IoT(Internet of Things) 장치를 사전에 부착하여 외부 시스템과 연동하는 방식을 사용함으로써, 특정 엘리베이터 환경에 종속된다는 한계를 가진다. 이에 본 논문에서는 로봇이 스스로 환경을 탐색하고 로봇에 부착된 매니플레이터를 이용하여 엘리베이터 버튼을 물리적으로 눌러 층간 이동까지 수행하며, 처음부터 끝까지 완전 자율로 실내 지도를 구축할 수 있는 다층 지도 구축 시스템을 제안한다. 더 나아가, 기존의 탐사 알고리즘⁹⁾의 방식은 주로 이동 비용과 정보 이득을 중심으로 탐사 목표를 선정하고, 시간 정보는 오래된 프론티어(frontier)를 제거하는 용도로 제한적으로 활용해 왔던 것과 달리, 본 논문에서는 프론티어의 생성 시점을 탐사 목표 선택 과정에 명시적으로 반영한다. 이를 통해 특정 영역이 부분적으로 미탐사된 상태에서 과거 프론티어가 반복적으로 선택되어 로봇이 비효율적으로 동작하는 문제를 완화한다.

제안하는 알고리즘의 개요는 [Fig. 1]과 같고 전체 구조는 크게 세 단계로 구성된다. 첫째, 3D LiDAR-IMU SLAM 기반으로 생성된 포인트 클라우드 형태의 Submap과 실시간 2D LiDAR 데이터를 융합하여 강건한 ROI(Region of Interest) Costmap과 Global Costmap을 생성한다. 둘째, 생성된 Costmap을 기반으로 ROI 단위 프론티어 검출, 다단계 필터링, 프론티



[Fig. 1] Proposed multi-floor autonomous map building system and its resulting 3D multi-floor map

어의 생성 시점을 고려하는 시간 감쇠 항을 탐사 점수 함수에 명시적으로 결합한 최적 목표 선정, 그리고 MPPI(Model Predictive Path Integral)¹⁰⁾ 기반 회피 경로계획 및 그래프 기반 전역 경로 계획을 포함하는 프론티어 기반 자율 탐사 알고리즘이 수행된다. 셋째, YOLO (You Only Look Once) 기반 버튼 인식과 4-DoF (Degrees of Freedom) 매니플레이터 시스템을 이용한 물리적 버튼 조작을 통해 엘리베이터와 상호작용하며 층간 이동이 가능한 엘리베이터 연계 알고리즘이 적용되어, 다층 환경 전체에서 끊임 없는 연속 탐사가 가능해진다.

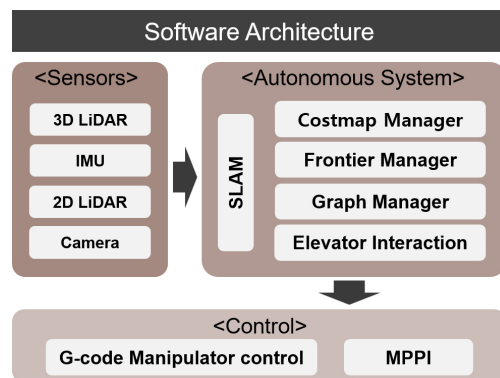
따라서 본 연구진의 주된 기여는 다음과 같다.

- 사람의 개입 없이 로봇 단독으로 건물 내 다층 공간 전체의 지도를 완전 자율로 구축하는 시스템을 제안하였다.
- 제안한 시스템을 실제 실내 건물 환경에서 구현 및 평가하여, 탐사·경로계획·엘리베이터 조작까지 모두 실증적으로 검증하였다.

또한, 다층 환경까지 확장 가능한 본 연구의 시스템은 실내 자율주행 기반의 라스트마일 배송 서비스의 대규모 적용성을 높이고, 환경 변화에 따른 지속적인 지도 갱신이 가능한 기반 기술로 활용될 수 있다.

2. 알고리즘

본 연구에서 제안하는 다층 환경용 자율 탐사 시스템은 단층에서 안정적으로 동작하는 탐사 알고리즘과, 층간 이동을 위한 엘리베이터 상호작용 알고리즘으로 구성된다. 전체 구조는 [Fig. 2]와 같다. 제안하는 알고리즘은 다음의 세 가지 핵심 구성 요소로 이루어진다. 다중 센서를 활용한 강인한 Costmap 생성 모듈(2.1절, [Fig. 2]의 Costmap Manager)과 생성된 Costmap 기반의 단층 자율 탐사 알고리즘(2.2절, [Fig. 2]의 Frontier Manager 및 Graph Manager), 그리고 엘리베이터와의 물리적 상호작용을 포함한 층간 이동 알고리즘(2.3절, [Fig. 2]의 Elevator



[Fig. 2] Overall software architecture of the proposed autonomous multi-floor indoor mapping system

Interaction)이 해당한다. 본 논문에서 사용하는 Costmap은 3차원 포인트 클라우드를 2차원 격자 형태로 사영하여 표현한 것이다. 각 셀은 장애물(occupied), 자유 공간(free), 미지 공간(unknown) 중 하나의 상태를 갖는다. 3차원 포인트 클라우드 데이터가 존재하는 셀은 장애물로 분류되며, 데이터가 존재하지 않는 셀은 자유 셀 또는 미지 셀로 구분되어 탐사 과정에 활용된다.

시스템의 전체 동작 흐름은 다음과 같다. 먼저, 3D LiDAR와 IMU를 이용한 LIO(Lidar Inertial Odometry) 기반 SLAM이 전역 3차원 포인트 클라우드 지도(global point cloud map)와 로봇 주변의 Submap을 생성한다. Submap은 로봇의 현재 위치를 기준으로 최근 N개의 키프레임에서 취득한 포인트 클라우드를 집합한 지역 지도이다. 키프레임은 로봇의 이동 거리 또는 회전 각도가 사전에 정의된 임계값을 초과할 때 선택되는 대표 프레임으로, 해당 시점에서 취득된 센서 데이터가 지도 갱신 및 Submap 구성을 위한 기준으로 사용된다. 전역 지도는 다층 실내 지도 구축의 결과물이 되며, Submap은 탐사를 위한 ROI Costmap 생성에 활용된다. Costmap Manager에서는 Submap과 실시간 2D LiDAR 데이터를 융합하여 ROI Costmap을 생성하고, 이를 누적하여 Global Costmap을 구축한다. Frontier Manager는 키프레임마다 새로 생성되는 ROI Costmap과 누적 업데이트되는 Global Costmap을 기반으로 프론티어 탐사 알고리즘을 수행한다. 이때, 자유 셀(free)과 미지 셀(unknown)의 경계에 위치한 셀을 프론티어 후보로 정의하며, 각 프론티어에 대해 도달 비용, 생성 시점, 탐사 시 기대되는 정보 이득을 기반으로 점수를 계산하여 가장 우수한 프론티어를 탐사 목표로 선택한다. Graph Manager는 SLAM으로부터 전달되는 각 키프레임 위치를 그래프의 노드로 설정하고, Global Costmap을 기반으로 노드 간 충돌 여부를 검사하여 충돌이 없는 노드들만을 엣지로 연결한다. 구축된 노드-엣지 그래프는 전역 경로계획에 사용되며, 로봇이 탐사 목표로 선정된 프론티어에 직접 접근할 수 없는 경우, Graph Manager는 해당 그래프 상에서 다익스트라(Dijkstra) 알고리즘을 사용해 최단 경로를 계산한다. 이후 로봇이 목적지에 도달할 수 있도록 중간 목표 지점을 순차적으로 제공한다. 로봇의 직접 접근 가능 여부는 실시간 2D LiDAR 포인트 클라우드 탐사 목표 방향으로 관측된 장애물 포인트까지의 최소 거리와 로봇에서 목표 지점까지의 유클리디안 거리를 비교하여 판단한다. 탐사 목표보다 가까운 위치에 장애물 포인트가 존재하면 차단된 경로로 판단해 전역 경로계획으로 전환한다. 결정된 탐사 목표 지점까지의 로봇 이동은 MPPI 기반 회피 경로 계획을 통해 수행된다. 단층 탐사가 완료되면, 로봇은 출발 지점으로 복귀한 뒤 장착된 소형 매니플레이터와 RGB-D 카메라를 이용해 엘리베이터 버튼을 인식하고 누름으로써 층간 이동을 수행한다.

2.1 다중 센서를 활용한 강인한 Costmap 생성 알고리즘

자율 탐사 알고리즘의 성능은 탐사의 기준이 되는 Costmap의 품질에 크게 의존한다. 본 연구에서는 3D LiDAR, IMU, 2D LiDAR를 융합하여 정확하고 강인한 Costmap을 생성하는 알고리즘을 개발하였다.

2.1.1 SLAM 기반 Submap 생성

3D LiDAR와 IMU를 융합한 LIO 계열 SLAM 알고리즘^[11]을 사용하여, 복도와 같이 특징점이 반복되는 실내 환경에서도 안정적인 위치 추정과 지도 생성을 수행한다. SLAM에서 새로운 LiDAR 키프레임 P_i^{sensor} 이 생성되면, 해당 포인트 클라우드는 식 (1)의 변환 행렬 $T \in SE(3)$ 을 통해 센서 좌표계에서 전역 좌표계(map frame)로 변환되어 P_i^{map} 로 표현된다. 이후 현재 로봇 위치를 기준으로 가장 가까운 K개의 키프레임을 선택하여 Submap M_{sb} 을 구성한다. 선택된 키프레임들의 포인트 클라우드를 합산하여 Submap을 형성하며, 로봇이 동일한 지역에 머무르는 경우에는 Submap 재구성을 생략해 연산 자원을 절약한다. 생성된 Submap은 추후 2D LiDAR 데이터와 융합되어 ROI Costmap 생성에 사용된다. 여기서 K_s 는 현재 로봇 위치를 기준으로 선택된 K개의 키프레임 인덱스 집합을 의미한다.

$$P_i^{map} = T_i P_i^{sensor}, M_{sb} = \bigcup_{i \in K_s} P_i^{map} \quad (1)$$

2.1.2 근거리 인식 보완을 위한 2D LiDAR 융합

3D LiDAR의 구조적 특성으로 인해 센서 근거리에서 반사광이 충분히 수신되지 않아 빈 영역(blind spot)이 발생할 수 있다. 이를 보완하기 위해 단일 레이저 기반의 2D LiDAR 데이터를 추가적으로 융합하였다. 2D LiDAR는 짧은 초점거리에서 벽면 및 장애물을 안정적으로 감지할 수 있어, 근거리 인식 성능을 효과적으로 향상시킨다.

2.1.3 다중 센서 기반의 ROI Costmap 및 Global Costmap 생성

Submap과 2D LiDAR로부터 획득된 포인트 클라우드 데이터는 타임 싱크(time synchronization)를 통해 동일한 시각 기준으로 정렬된 후, 레이크스팅 기반 주행 가능 영역 추출 알고리즘을 적용하여 ROI Costmap으로 생성된다. ROI Costmap 생성 과정은 다음과 같다. Submap 생성 시 기록된 타임스탬프를 이용해 동일 시각의 2D LiDAR의 스캔데이터를 불러오고, 당시의 로봇 자세를 활용해 해당 스캔데이터를 전역 좌표계로 변환한다. Submap은 로봇의 실제 이동 가능 영역만 사용하기 위해 로봇 높이 기반 필터링 및 ROR(Radius Outlier Removal) 필터링으로 노이즈를 제거한다. 그 다음, 필터링된 Submap과 전역

[Table 1] Parameters used for ROI Costmap generation

Parameter	Value
Costmap resolution	0.2 m
ROI size	20 m
Number of keyframes per Submap	7
ROR search radius	0.5 m
ROR minimum neighbors	2

좌표계로 변환된 LiDAR 스캔 데이터를 통합하여 하나의 3차원 포인트 클라우드를 구성한다. 통합된 포인트 클라우드는 2차원 Costmap으로 투영되며, 이후 레이캐스팅 기반 필터링을 통해 주행 가능 영역을 추출한다. 이렇게 생성된 Costmap은 ROI Costmap으로 저장되며, 시간에 따라 순차적으로 누적되어 전역 환경을 표현하는 Global Costmap을 형성한다. Submap 기반 ROI Costmap 생성에 사용된 파라미터는 [Table 1]에 요약되어 있다.

ROI Costmap은 투표 기반 누적 방식을 통해 Global Costmap으로 확장된다. 각 키프레임에서 얻은 ROI Costmap은 식(2)와 같이 Global Costmap 상태를 업데이트한다.

$$G_k(x, y) = A(G_{k-1}(x, y), R_k(x, y)) \quad (2)$$

여기서 $G_k(x, y)$ 는 k 번째 키프레임까지 누적된 Global Costmap을 의미하며, $R_k(x, y)$ 는 각 키프레임에서 생성된 ROI Costmap을 의미한다. 연산자 $A(\cdot)$ 는 새로운 관측을 기존 Global Costmap에 누적시키는 규칙을 정의한다. 각 셀은 자유공간 표(free vote) V_f 와 장애물 표(obstacle vote) V_o 를 독립적으로 누적하며, 서로 모순되는 관측이 발생하면 반대 표를 0으로 초기화하여 센서 노이즈로 인해 발생하는 오탐 관측이 Global Costmap에 누적되는 것을 억제한다. 상태 전이 규칙(state transition rule)은 식(3)과 같다.

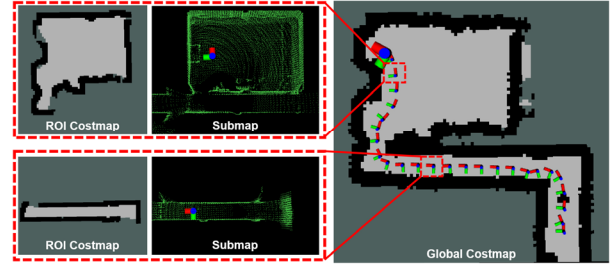
$$G_k(x, y) = \begin{cases} 100, & \text{if } V_o(x, y) \geq \theta_{fo}, \\ 0, & \text{if } V_f(x, y) \geq \theta_{of}, \\ G_{k-1}(x, y), & \text{otherwise.} \end{cases} \quad (3)$$

임계값 θ_{fo}, θ_{of} 는 각각 $Free \rightarrow Obstacle$ 및 $Obstacle \rightarrow Free$ 전이의 신뢰도 기준을 나타내며, 실험적으로 결정하였다. 이 과정은 일시적 관측 불확실성에 대한 강인성을 확보하는데 기여한다. 식 (3)의 상태 전이를 구현하기 위해 사용된 파라미터 값과 초기화 규칙은 [Table 2]에 정리되어 있다.

그 결과 생성된 ROI 및 Global Costmap의 예시는 [Fig. 3]에 나타나있다. [Fig. 3]의 오른쪽은 누적된 Global Costmap과 로봇의 이동 경로를 나타내며, 왼쪽은 특정 키프레임에서 취득된

[Table 2] Parameters for vote-based accumulation and state transition in the Global Costmap

Category	Variable	Value
Transition threshold	θ_{fo}	2
Transition threshold	θ_{of}	2
Vote initialization	free_vote_count, obstacle_vote_count	0
Vote reset rule	Conflict-based reset	Reset to 0
Vote reset rule	Post-transition reset	Reset to 0



[Fig. 3] Process of generating the accumulated Global Costmap (right) and the Submap with its ROI Costmap used at each location (left)

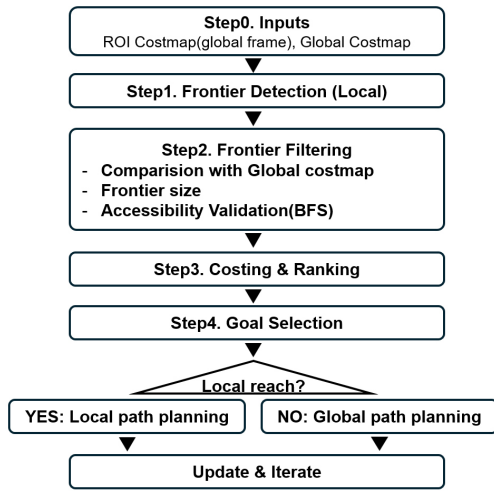
Submap과 해당 Submap으로부터 생성된 ROI Costmap을 보여준다.

2.2 Costmap 기반 탐사 알고리즘

2.2.1 탐사 알고리즘

본 연구에서 제안하는 탐사 알고리즘은 키프레임 단위로 생성된 ROI Costmap과 Global Costmap을 병행 활용하여 효율적이고 안정적인 프론티어 기반 탐사를 수행한다. 전체 과정은 [Fig. 4]와 같이 프론티어 검출, 필터링 및 비용 계산, 목표 선정 및 이동의 단계로 구성된다.

먼저, ROI Costmap에서 새로운 프론티어를 검출하고 해당 시점의 키프레임 위치 정보를 함께 저장한다. 프론티어는 기존 연구^[12]에서 제안된 바와 같이 미지의 셀과 자유 셀의 경계에 위치한 셀로 정의된다. 이후 탐사 안정성을 위해 3단계 필터링 과정을 수행한다. 첫 번째 단계는 셀 단위 유효성 필터링이다. Global Costmap과 비교하여 프론티어 주변 4방향 이웃 중 하나라도 자유 셀이 존재하는 경우에만 유효 프론티어로 유지한다. 모든 이웃이 미지의 셀 또는 장애물 셀인 경우 간헐 프론티어로 간주하여 제거한다. 두 번째 단계는 군집 크기 기반 필터링이다. 군집 크기가 일정 기준 미만인 소규모 프론티어는 노이즈로 판단하여 제거한다. 마지막 단계는 BFS(Breadth-First Search) 기반 접근성 필터링이다. 프론티어가 미지의 셀 또는 장애물 셀 사이에 존재하여 로봇이 실제로 접근 불가능한 경우 해당 프론티어를 제거함으로써 탐사 효율을 높인다.



[Fig. 4] Overall pipeline of the proposed frontier-based exploration algorithm, including frontier detection, filtering, cost evaluation, goal selection, and local/global path planning

필터링된 프론티어 f 에 대해 이동 항(travel term), 정보 항(information term), 시간 감쇠 항(temporal decay term)을 고려한 탐사 비용을 계산하고, 이를 정렬해 최적 탐사 목표를 선택한다. 이동 항 C_f 은 현재 로봇 위치에서 프론티어까지의 도달 시간을 예측하는 값으로, 단순 유클리드 거리뿐 아니라 방향 오차를 함께 고려하여 로봇이 안정적인 탐사 방향성을 유지하도록 한다. 시간 감쇠 항 R_f 은 지수함수 형태로 구성하여 최신 프론티어는 1에 가깝고, 오래된 프론티어는 0에 수렴하도록 하여 오래된 프론티어의 선택 확률을 자연스럽게 낮춘다. 이를 통해 로봇이 최신 프론티어를 우선적으로 탐사하도록 유도한다. 이를 통해 특정 영역이 부분적으로 미탐사된 상태에서 과거에 생성된 프론티어를 반복적으로 선택하여 장거리 이동을 수행하는 비효율적인 탐사 상황을 완화할 수 있다. 정보 항 I_f 은 프론티어 군집을 이루는 셀의 개수로 정의되며, 값이 클수록 더 넓은 미탐사 영역의 확장 가능성이 크다는 것을 의미한다.

탐사 점수식은 식(4)와 같다. 본 식은 이동 항 C_f 을 기반 비용으로 설정하고, 프론티어의 시간 감쇠 항 R_f 과 정보 항 I_f 을 비용 절감 효과를 부여하는 보상 계수로 적용하여 점수 S_f 를 산출한다. 이에 따라, 시간 감쇠 항과 정보 항이 높을수록 점수가 감소하며, 점수가 낮은 프론티어가 탐사 목표로 우선 선택된다.

$$S_f = \frac{C_f}{1 + \alpha R_f + \beta I_f} \quad (4)$$

각 비용 항목은 식(5)과 같이 정의된다. 식의 λ 는 시간 경과에 따른 시간 감쇠 속도를 조절하며, α 와 β 는 각 보상 항의 가중치를 조절하는 역할을 한다.

$$C_f = \frac{\sqrt{(x_f - x_r)^2 + (y_f - y_r)^2}}{\nu} + \gamma \frac{|\theta_f - \theta_r|}{\omega} \quad (5)$$

$$R_f = \exp[-\lambda(k_t - k_f)]$$

$$I_f = N_f$$

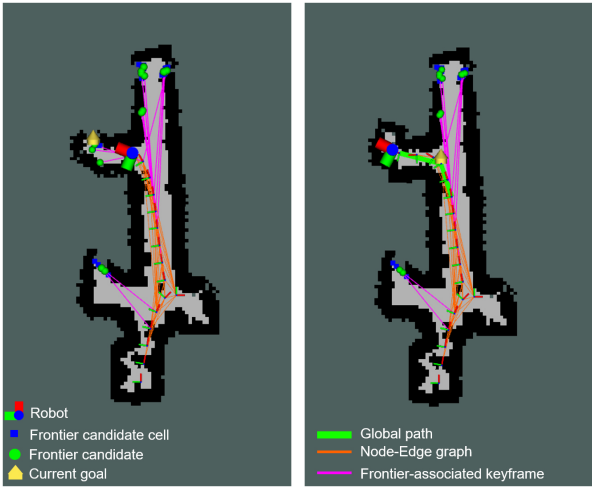
탐사 점수식에 의해 선정된 목표 지점이 로컬 영역 내에서 접근 가능한 경우 로봇 해당 목표 지점으로 직접 접근한다. 반면, 로봇과 목표 지점 사이의 경로가 장애물에 의해 차단된 경우에는 Graph Manager에 전역 경로 계획을 요청한다. 직접 접근 가능 여부는 현재 로봇 위치에서 획득된 2D LiDAR 데이터를 기반으로 판단하며, 로봇과 목표 지점을 연결하는 방향으로 기준으로 각도 범위 θ_{target} 이내에 포함되는 2D LiDAR 포인트를 선택하고, 이 중 장애물 포인트까지의 최소 거리와 목표 지점까지의 유클리드 거리를 비교한다. 최소 거리가 목표 지점까지의 거리보다 작은 경우, 로봇과 목표 지점 사이에 장애물이 존재하는 것으로 판단하여 전역 경로 계획을 수행한다. 로봇과 목표 지점 간의 거리가 사전에 정의된 임계 거리 d_{goal} 이내로 진입하면 목표 지점에 도달한 것으로 판단한다. 이후 프론티어 정보를 갱신하고, 동일한 절차를 반복함으로써 시스템은 미탐사 영역을 점진적으로 확장한다. 이러한 과정을 통해 모든 프론티어가 소진될 때까지 자율 탐사가 지속적으로 수행된다. 탐사 알고리즘에 사용된 파라미터 및 판단 조건은 [Table 3]에 정리되어 있다.

2.2.2 전역 경로계획을 위한 Graph Manager

탐사 과정에서는 막다른 길에서 후퇴가 필요하거나, 멀리 떨어진 탐사 목표로 이동해야 하는 경우가 발생한다. 이를 해결하기 위해 본 연구에서는 기존 연구^[13]에서 검증된 키프레임 기반 그래프 구성 방식을 기반으로 전역 경로계획을 수행한다. 구성된 그래프 위에서 다익스트라 알고리즘을 적용하여 최단 경로를 계산함으로써, 탐사 중 효율적이고 안정적인 장거리 이동을 가능하게 한다.

[Table 3] Parameters used for frontier scoring and exploration control

Parameter	Value
Linear speed ν	0.8 m/s
Angular speed ω	0.5 rad/s
Rotation weight γ	4.0
Temporal decay rate λ	0.05
Temporal decay weight α	30.0
Information gain weight β	0.05
Exploration termination condition	Number of frontiers = 0
Target direction threshold θ_{target}	5 deg
Goal proximity threshold d_{goal}	0.7 m



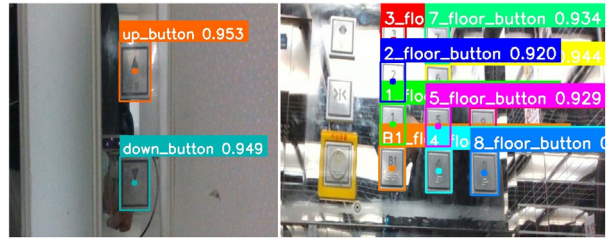
[Fig. 5] Exploration algorithm in operation (left) and global path planning result (right)

SLAM 알고리즘에서 새로운 키프레임이 생성되면 이를 노드로 등록하고, 각 노드 간 연결은 2.1절에서 생성된 Global Costmap을 통해 충돌 검사를 수행하여 결정한다. 새로운 키프레임이 생성될 때마다 그래프가 지속적으로 확장된다. 탐사 알고리즘(2.2.1절)으로부터 전역 경로계획 요청이 들어오면, Graph Manager는 현재 노드와 목표 노드를 입력받아 다익스트라 알고리즘을 수행해 최단 경로를 계산한다. 계산된 경로를 구성하는 노드들을 순차적으로 로봇의 중간 목적지로 제공하여 로봇이 장애물을 우회하면서 목적지에 도달하도록 한다.

[Fig. 5]는 탐사 과정에서 노드-엣지 그래프가 구성되는 방식과 전역 경로 계획 결과를 시각적으로 보여준다. 그림에서 주황색 선은 충돌이 없는 노드 간 연결을 의미하며, 초록색 점은 후보 프론티어, 파란색 셀은 해당 프론티어를 구성하는 셀이다. 또한 각 프론티어는 이를 최초로 관측한 키프레임과 연계되며, 이는 자주색 선으로 표현되어 있다. 노란색 화살표는 현재 선정된 탐사 목표를 나타낸다. 왼쪽의 경우 막다른 지형으로 인해 전역 경로 계획이 실행되며, 오른쪽의 굵은 형광 연두색 경로는 다익스트라 알고리즘을 통해 생성된 최단 경로를 나타낸다.

2.3 다층 확장을 위한 엘리베이터 연계 알고리즘

모바일 로봇이 다층 환경에서 자율적으로 지도를 확장하기 위해서는 층간 이동 수단인 엘리베이터와의 상호작용이 필수적이다. 엘리베이터 연계 방식은 크게 두 가지로 나눌 수 있다. 첫째는 엘리베이터 제어 API (Application Programming Interface)를 통해 직접 통신하는 방식이며, 둘째는 로봇이 카메라와 매니플레이터를 이용해 실제 버튼을 인식하고 눌러 연계하는 방식이다. 전자의 경우 건물마다 제어 규격이 상이하고 보안상 API 접근이 제한되는 경우가 많아 실제 서비스 환경 적용



[Fig. 6] Real-world detection results of the YOLO v11-based elevator button recognition model

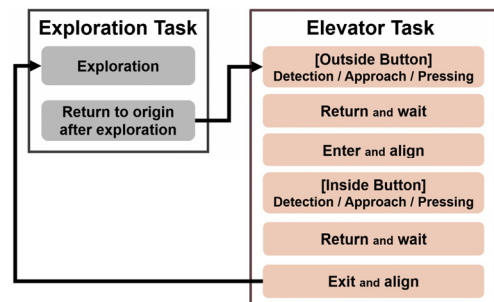
이 어렵다. 따라서 본 연구에서는 물리적 버튼 조작 기반의 엘리베이터 연계 방식을 채택하였다.

2.3.1 버튼 조작을 위한 하드웨어 구성 및 버튼 인식

엘리베이터 버튼 조작을 위해 로봇 상단에 4-DoF 매니플레이터 시스템과 RGB-D 카메라를 사용하였다. 매니플레이터 시스템은 G-code기반으로 제어되며, 엔드이펙터의 목표 위치는 G-code 명령으로 변환되어 각 축 모터로 전달된다. RGB-D 카메라는 엘리베이터 버튼 인식을 위한 주 센서로 사용되며 카메라의 RGB 영상과 깊이 정보를 융합하여 호출 버튼 및 층 선택 버튼의 위치를 추정한다. 다양한 엘리베이터 환경에서 촬영한 데이터셋을 구축해 YOLO v11 모델^[14]로 학습하였으며, [Fig. 6]과 같이 호출 버튼(상·하)과 층 선택 버튼을 안정적으로 탐지하는 성능을 확보하였다. 인식된 버튼의 3차원 위치는 매니플레이터 제어 모듈로 전달되어 엔드이펙터의 목표 위치로 변환되며, 이를 통해 다양한 구조의 엘리베이터에서도 일관된 인식-조작 파이프라인을 수행할 수 있다.

2.3.2 엘리베이터 상호작용 알고리즘

본 연구에서 제안하는 다층 지도 구축의 절차는 단층 탐사 (Exploration Task)와 엘리베이터 상호작용(Elevator Task)이 연속적으로 반복되는 구조로 이루어진다. 전체 동작 흐름은 [Fig. 7]에 요약되어 있으며, 한 층의 탐사가 완료되면 로봇은 엘리베이터 앞 원점으로 복귀하여 다음 층으로 이동하기 위한



[Fig. 7] Overall pipeline integrating exploration and elevator-interaction tasks

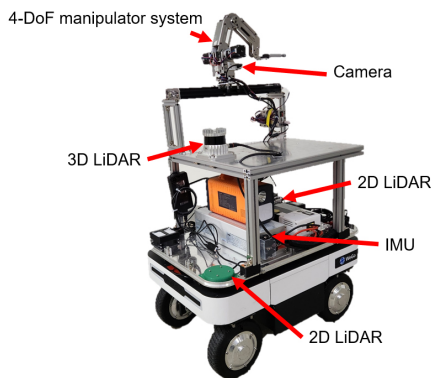
버튼 조작을 수행한 뒤, 새로운 층에서 다시 탐사 알고리즘을 실행하는 방식으로 운영된다.

탐사 단계에서는 프론티어 기반 알고리즘을 통해 해당 층의 미탐사 영역을 확장하고, 탐사 종료 조건을 만족하면 시작점으로 회귀한다. 이후 Elevator Task가 시작되며, 이 과정은 엘리베이터 외부 버튼 조작과 내부 버튼 조작으로 구성된다. 외부 버튼 조작 단계에서는 RGB-D 카메라 기반 YOLO v11 모델을 사용해 호출 버튼을 탐지한 후, 매니퓰레이터를 통해 접근 및 버튼 누름 동작을 수행한다. 호출 버튼이 눌리면 로봇은 엘리베이터 문 앞에서 정렬 후 대기하며, 2D LiDAR 거리 변화를 통해 문 개폐 여부를 실시간으로 판단한다. 문이 열리면 로봇은 엘리베이터 내부로 진입한다. 엘리베이터 내부에서는 주변 환경이 협소하고 금속 반사로 인해 LiDAR 기반 위치추정의 신뢰도가 급격히 저하되므로, 본 연구에서는 Wheel Encoder와 IMU를 조합한 EKF(Extended Kalman Filter) 기반 오도메트리 추정을 사용한다. 내부 버튼 조작은 외부와 동일한 절차(탐지-접근-누름)로 수행되며, 목표 층 도달 후 문이 열리면 로봇은 다시 정렬 후 엘리베이터에서 안전하게 이탈한다. 이후 로봇은 새로운 층의 기준 위치에서 SLAM을 초기화한 뒤, 탐사 알고리즘을 재시작하여 다음 층의 지도를 생성한다. 이와 같은 탐사-귀환-버튼 조작-탐승-층 이동-재탐사의 순환 구조를 통해 로봇은 외부 제어 시스템 없이도 실제 건물 환경에서 완전 자율로 다층 지도 구축을 수행할 수 있다.

3. 실험결과

3.1 실험 구성

실험에 사용된 로봇 플랫폼은 [Fig. 8]에 나타나 있다. 로봇 상단에 장착된 3D LiDAR와 중앙부에 부착된 IMU 센서를 기반으로 SLAM 알고리즘이 동작하여 각 층의 지도를 구축한다.



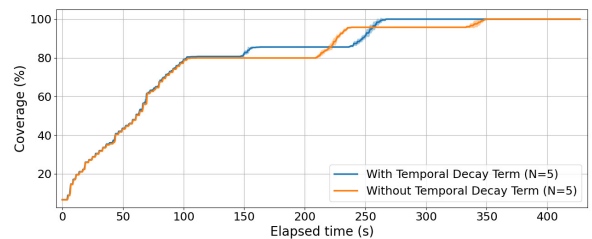
[Fig. 8] Hardware components of the robot, including sensors for SLAM and a robotic arm for elevator interaction

또한 로봇 하단부에 장착된 2D LiDAR는 지역 경로 계획 과정에서 장애물 회피를 위해 활용된다. 로봇 상단 측면에는 RGB-D 카메라와 함께 4-DoF 매니퓰레이터 시스템이 장착되어 있으며, 이를 통해 엘리베이터 버튼의 인식 및 조작을 수행한다. 해당 4-DoF 매니퓰레이터 시스템은 3-DoF 소형 매니퓰레이터와 1-DoF 직선 이동 레일로 구성되어, 엘리베이터 외부 및 내부 버튼의 다양한 위치와 높이에 대응할 수 있도록 설계되었다.

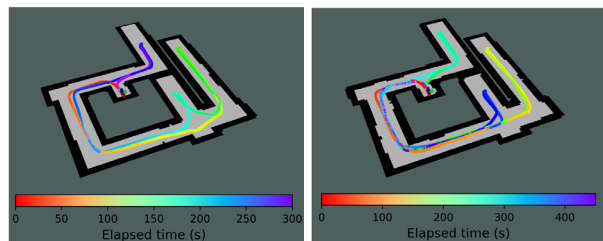
3.2 시간 감쇠 항의 효율성 실험

탐사 목표 선정 과정에서 도입한 시간 감쇠 항의 효과를 정량적으로 검증하기 위해 요소 제거 실험을 수행하였다. 이를 위해 동일한 시뮬레이션 환경과 동일한 파라미터 설정 하에서 시간 감쇠 항을 적용한 경우와 적용하지 않은 경우를 비교 실험하였다. 실험은 각 설정에 대해 5회 반복 수행하였으며, 결과는 평균값을 기준으로 분석하였다.

[Fig. 9(a)]는 시간 감쇠 항의 적용 여부에 따른 시간 대비 탐사 커버리지를 정량적으로 비교한 결과를 보여준다. 시간 감쇠 항을 적용한 경우, 로봇은 인접한 미탐사 영역을 우선적으로 탐사함으로써 더 빠른 속도로 커버리지를 증가시키는 경향을 보였다. 반면, 시간 감쇠 항을 적용하지 않은 경우에는 로봇이 이전에 이동한 경로를 반복적으로 다시 이동하는 경향이 나타나



(a)



(b)

(c)

[Fig. 9] Ablation study on the effect of the temporal decay term in frontier scoring. (a) Coverage over time averaged over multiple runs (N=5). (b) Robot trajectory with the temporal decay term, showing coherent local exploration with reduced backtracking. (c) Robot trajectory without the temporal decay term, exhibiting frequent back-and-forth motions.

탐사 효율이 저하되었다. 동일한 환경에서의 로봇 이동 경로를 시각화한 결과는 [Fig. 9(b)]와 [Fig. 9(c)]에 나타나 있다. 시간 감쇠 항을 적용한 경우([Fig. 9(b)]), 로봇은 이미 탐사된 영역 인근을 비교적 일관성 있게 탐사한 후 새로운 영역으로 확장하는 경향을 보였으며, 그 결과 중복 경로가 상대적으로 감소하였다. 반면, 시간 감쇠 항을 적용하지 않은 경우([Fig. 9(c)])에는 로봇이 이전에 이동한 경로를 반복적으로 다시 이동하는 현상이 빈번하게 관찰되었다.

정량적으로, 시간 감쇠 항을 적용한 경우 전체 탐사 시간은 약 300초로 측정되었으며, 이를 적용하지 않은 경우의 약 450초와 비교하여 탐사 시간이 약 33% 감소하였다. 이러한 결과는 시간 감쇠 항이 과거에 생성된 프론티어의 선택을 억제하고, 최신 프론티어를 우선적으로 탐사하도록 유도함으로써 탐사 경로의 중복을 줄이고 전체 탐사 효율을 향상시킴을 보여준다.

3.3 실제 환경에서 전체 시나리오 실험

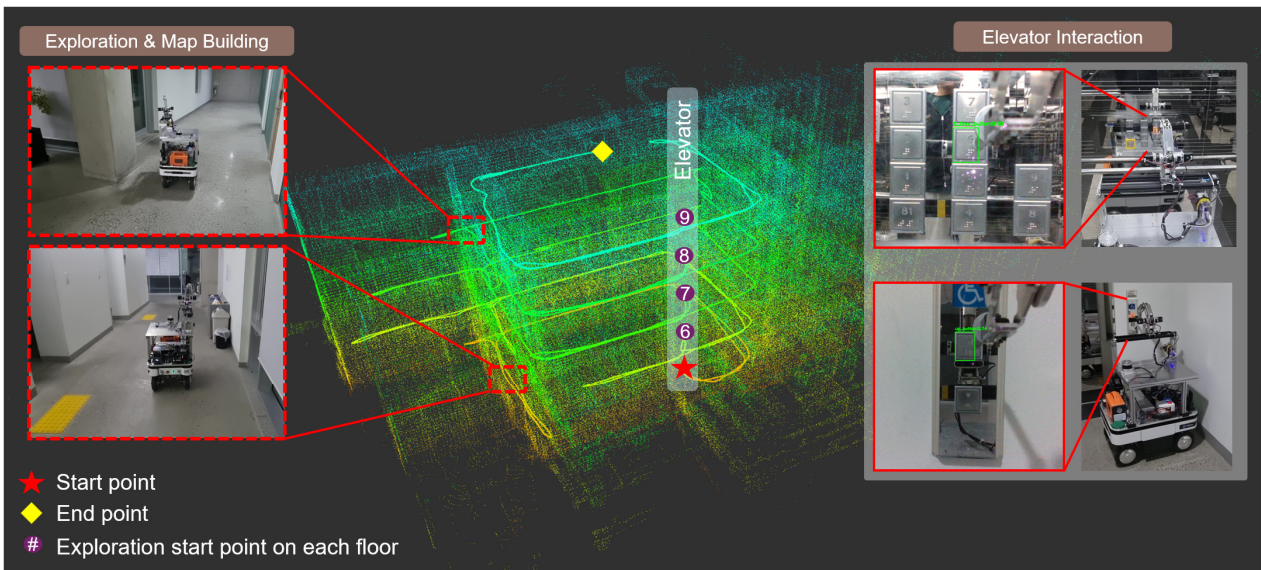
전체 시나리오 실험은 한국과학기술원(KAIST) IT 융합빌딩(N1)에서 진행하였다. 실험 환경의 규모와 구조적 특성을 고려하여 5층부터 9층까지 각 층의 절반 영역을 대상으로 평가를 수행하였다. 로봇은 5층 엘리베이터 앞에서 시작하여 탐사 알고리즘을 통해 해당 층의 지도를 자율적으로 구축한다. 탐사가 완료되었다고 판단되면 로봇은 시작 지점으로 복귀한 후, 후방에 위치한 엘리베이터의 호출 버튼을 인식하여 매니플레이터를 이용해 버튼을 누르고 탑승한다. 도착 층에서는 엘리베이터 문이 열리면 하차하고 새로운 층에 대한 탐사 알고리즘을 다시

실행한다. 이러한 과정은 9층까지 연속적으로 반복되며, 전 과정은 완전 자율 방식으로 수행되었으며, 실험 결과는 [Fig. 10]에 나타나 있다.

전체 실험 소요 시간은 27분 20초이며, KAIST N1 건물의 5층부터 9층까지 완전 자율 주행 기반의 다층 지도 구축을 성공적으로 수행하였다. [Table 4]에는 각 층별 탐사 시간, 이동 거리, 그리고 층간 이동 시간이 정리되어 있다. 여기서 탐사 시간과 이동 거리는 각 층에서의 자율 탐사에 소요된 시간과 주행 거리만을 의미하며, 다음 층으로 이동하기 위한 복귀 과정에서의 시간과 이동 거리는 포함하지 않는다. 층간 이동 시간은 엘리베이터를 이용한 층 이동 과정 전체를 포함하며, 구체적으로 엘리베이터 호출 버튼의 인식 및 조작, 엘리베이터 탑승, 엘리베이터 내부 버튼 인식 및 층 선택, 그리고 목표 층 도착 후 하차 과정을 모두 포함한다. 실험 결과, 각 층에서의 탐사 단계는 평균적으로 약 3분이 소요되었으며, 엘리베이터 호출, 탑승 및 하차를 포함한 층간 이동 과정에는 약 2분 내외의 시간이 소요되

[Table 4] Exploration time, travel distance, and inter-floor transition time for each floor

Floor	Exploration Time (min:sec)	Travel Distance (m)	Inter-floor Transition Time (min:sec)
5F	1:42	54.42	1:50
6F	3:08	93.33	3:00
7F	2:55	88.70	1:57
8F	2:46	79.06	1:55
9F	3:24	107.78	-



[Fig. 10] Result of the fully autonomous multi-floor map-building system. The robot explores Floors 5–9 sequentially by performing button recognition and pressing the corresponding elevator buttons with its robotic arm. The markers represent the start point (red star), end point (yellow diamond), and the exploration start point on each floor (purple icons)

[Table 5] Success rate of the elevator interaction task

	Successes	Trials	Success Rate(%)
Elevator Interaction Task	19	20	95%

었다. 단, 6층에서 7층으로 이동하는 구간의 경우 외부 이용자에 의해 엘리베이터가 다른 층으로 이동하는 지연이 발생하여 상대적으로 긴 시간이 소요되었다. [Table 5]에는 엘리베이터 연계 알고리즘의 성공률이 제시되어 있다. 실제 실내 환경에서 총 20회의 엘리베이터 상호작용 시도 중 19회가 성공적으로 수행되었으며, 이를 통해 95%의 성공률을 달성함을 확인하였다. 해당 결과는 제안된 엘리베이터 연계 알고리즘이 실제 환경에서도 높은 신뢰도로 동작함을 보여준다. 전체 실험 영상은 링크 (<https://youtu.be/doCVS7Weils>)에서 확인할 수 있다.

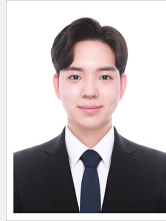
4. 결 론

본 논문에서는 3D LiDAR-IMU 기반 SLAM과 프론티어 기반 탐사 알고리즘, 그리고 엘리베이터 버튼 인식 및 조작을 결합하여, 로봇이 다층 건물을 완전 자율로 탐사하고 지도를 구축할 수 있는 다층 지도 구축 시스템을 제안하였다. 제안된 시스템은 다층 환경에서의 탐사에 국한되지 않고, 건물 내부에서 불가피하게 발생하는 층간 이동 과정까지 로봇 스스로 수행할 수 있도록 설계되었다. 제안된 시스템은 Submap과 2D LiDAR 정보를 결합하여 강인한 Costmap을 생성하고, 이를 기반으로 ROI 단위 프론티어 검출, 다단계 필터링, 이동 항-정보 항-시간 감쇠 항 기반의 최적 목표 선정 과정을 거쳐 탐사를 수행한다. 또한 MPPI 기반 지역 경로 계획과 그래프 기반 전역 경로 계획을 연계하여, 로봇이 장애물로 인해 직접 접근이 어려운 상황에서도 안정적으로 탐사를 지속할 수 있도록 하였다. 더불어, YOLO 기반 버튼 인식과 4-DoF 매니퓰레이터 시스템을 활용한 버튼 접근 및 누름 동작을 포함한 엘리베이터 상호작용 절차를 통해 새로운 층에 도달한 뒤 자동으로 탐사를 재개할 수 있는 구조를 구성하였다. 실험 결과, 프론티어 점수 계산 과정에 시간 감쇠 항을 포함함으로써 인접한 프론티어를 우선적으로 탐사하도록 유도할 수 있었으며, 이로 인해 탐사 경로의 중복이 감소하고 전체 탐사 효율이 향상됨을 확인하였다. 또한 제안된 시스템은 KAIST N1 건물 5층부터 9층까지의 실제 환경에서 전체 시나리오에 대해 검증되었다. 각 층에 대한 자율 탐사 수행 이후 엘리베이터 앞 복귀, 버튼 인식 및 탑승, 하차 후 탐사 재개로 이어지는 일련의 과정이 모두 안정적으로 동작함을 확인하였다. 아울러 버튼 인식 및 엘리베이터 조작을 포함한 엘리베이터 연계 작업에서 95%의 성공률을 기록함으로써, 제안하는 시스템이 실제 실내 환경에서도 적용 가능성을 입증하였다.

References

- [1] E. Alverhed, S. Hellgren, H. Isaksson, L. Olsson, H. Palmqvist, and J. Flodén, "Autonomous last-mile delivery robots: a literature review," *European Transport Research Review*, vol. 16, no. 4, pp. 1-13, Jan., 2024, DOI: 10.1186/s12544-023-00629-7.
- [2] N. Boysen, S. Fedtke, and S. Schwerdfeger, "Last-mile delivery concepts: a survey from an operational research perspective," *OR Spectrum*, vol. 43, no.1, pp. 1-58, Mar., 2021, DOI: <https://doi.org/10.1007/s00291-020-00607-8>.
- [3] D. Lee, G. Kang, B. Kim, and D. H. Shim, "Assistive Delivery Robot Application for Real-World Postal Services," *IEEE Access*, vol. 9, pp. 141981-141998, Oct., 2021, DOI: 10.1109/ACCESS.2021.3120618.
- [4] T. Kim, G. Kang, D. Lee, and D. H. Shim, "Development of an Indoor Delivery Mobile Robot for a Multi-Floor Environment," *IEEE Access*, vol. 12, pp. 45202-45215, Mar., 2024, DOI: 10.1109/ACCESS.2024.3381489.
- [5] H. Yin, X. Xu, S. Lu, X. Chen, R. Xiong, S. Shen, C. Stachniss, and Y. Wang, "A Survey on Global LiDAR Localization: Challenges, Advances and Open Problems," *International Journal of Computer Vision*, vol. 132, no.8, pp. 3139-3171, Mar., 2024, DOI: 10.1007/s11263-024-02019-5.
- [6] S. Leisiazar, M. Mahdavian, E. J. Park, and M. Chen, "Real-time Mapping of Multi-Floor Buildings Using Elevators," *2022 IEEE/ASME (AIM) International Conference on Advanced Intelligent Mechatronics*, Sapporo, Japan, pp. 314-321, 2022, DOI: 10.1109/AIM52237.2022.9863407.
- [7] S. Jiang, W. Yao, M.-S. Wong, M. Hang, Z. Hong, E.-J. Kim, S.H. Joo, and T.-Y. Kuc, "Automatic Elevator Button Localization Using a Combined Detecting and Tracking Framework for Multi-Story Navigation," *IEEE Access*, vol. 8, pp. 1118-1134, Dec., 2019, DOI: 10.1109/ACCESS.2019.2958092.
- [8] J. Palacín, R. Bitriá, E. Rubies, and E. Clotet, "A Procedure for Taking a Remotely Controlled Elevator with an Autonomous Mobile Robot Based on 2D LiDAR," *Sensors*, vol. 23, no. 13, p. 6089, Jul., 2023, DOI: 10.3390/s23136089.
- [9] B. P. L. Lau, B. J. Y. Ong, L. K. Y. Loh, R. Liu, C. Yuen, G. S. Soh, and U.-X. Tan, "Multi-AGV's Temporal Memory-based RRT Exploration in Unknown Environment," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 9256-9263, Jul., 2022, DOI: 10.1109/LRA.2022.3190628.
- [10] G. Williams, P. Drews, B. Goldfain, J. M. Rehg, and E. A. Theodorou, "Information-Theoretic Model Predictive Control: Theory and Applications to Autonomous Driving," *IEEE Transactions on Robotics*, vol. 34, no. 6, pp. 1603-1622, Dec., 2018, DOI: 10.1109/TRO.2018.2865891.
- [11] K. Chen, R. Nemirow, and B. T. Lopez, "Direct LiDAR-Inertial Odometry: Lightweight LIO with Continuous-time Motion Correction," *arXiv:2203.03749*, 2022, DOI: 10.48550/arXiv.2203.03749.

- [12] B. Yamauchi, "A frontier-based approach for autonomous exploration," *Proceedings 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation CIRA'97. 'Towards New Computational Principles for Robotics and Automation'*, Monterey, USA, pp. 146-151, 1997, DOI: 10.1109/CIRA.1997.613851.
- [13] B. Kim, H. Seong, and D. H. Shim, "Topological Exploration using Segmented Map with Keyframe Contribution in Subterranean Environments," *2024 IEEE International Conference on Robotics and Automation (ICRA)*, Yokohama, Japan, pp. 6199-6205, 2024, DOI: 10.1109/ICRA57147.2024.10610605.
- [14] R. Khanam and M. Hussain, "Yolov11: An Overview of the Key Architectural Enhancements," *arXiv:2410.17725*, 2024, DOI: 10.48550/arXiv.2410.17725.



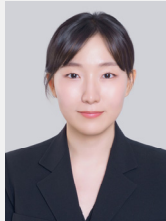
노 주 형

2022 광운대학교 로봇학부 정보제어전공 (학사)

2025 KAIST 로봇공학학제전공(석사)

2025~현재 KAIST 전기 및 전자공학부 박사 과정

관심분야: 로봇틱스, 항법, UGV, 자율탐사



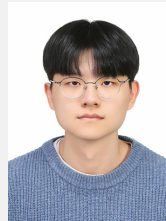
강 규 리

2020 성균관대학교 기계공학부(학사)

2022 KAIST 로봇공학학제전공(석사)

2022~현재 KAIST 전기 및 전자공학부 박사 과정

관심분야: 로봇틱스, 모바일 매니플레이터



김 연 찬

2024 한양대 ERICA 로봇공학과(학사)

2024~현재 KAIST 로봇공학학제전공 석사 과정

관심분야: 로봇틱스, 사회적 기반 자율주행



심 현 철

1991 서울대학교 기계공학(학사)

1993 서울대학교 기계공학(석사)

2000 University of California Berkeley Mechanical Engineering(박사)

관심분야: 로봇틱스, 드론, 자율로봇, 자율주행차