

실내 환경에서의 주행가능성을 고려한 라이다 기반 이동 로봇 탐사 기법

LiDAR-based Mobile Robot Exploration Considering Navigability in Indoor Environments

유혜정[†] · 최진우¹ · 김태현²

Hyejeong Ryu[†], Jinwoo Choi¹, Taehyeon Kim²

Abstract: This paper presents a method for autonomous exploration of indoor environments using a 2-dimensional Light Detection And Ranging (LiDAR) scanner. The proposed frontier-based exploration method considers navigability from the current robot position to extracted frontier targets. An approach to constructing the point cloud grid map that accurately reflects the occupancy probability of glass obstacles is proposed, enabling identification of safe frontier grids on the safety grid map calculated from the point cloud grid map. Navigability, indicating whether the robot can successfully navigate to each frontier target, is calculated by applying the skeletonization-informed rapidly exploring random tree algorithm to the safety grid map. While conventional exploration approaches have focused on frontier detection and target position/direction decision, the proposed method discusses a safe navigation approach for the overall exploration process until the completion of mapping. Real-world experiments have been conducted to verify that the proposed method leads the robot to avoid glass obstacles and safely navigate the entire environment, constructing the point cloud map and calculating the navigability with low computing time deviation.

Keywords: Mobile Robot Exploration, Path-Planning, Rapidly-Exploring Random Tree, Grid Map Skeletonization, Indoor Environment With Glass Obstacles

1. 서 론

이동 로봇 탐사 기술은 사전(prior) 정보 없는 환경에 대한 지도 작성을 위해 다음 탐사 목표 지점을 자율적으로 결정하여 주행하는 기술이다. 즉, 환경 전체에 대한 지도 작성이 완료

될 때까지, 동시 위치 인식 및 지도 작성(SLAM, Simultaneous Localization and Mapping)을 수행하면서 최적 센서 데이터 획득을 위한 목표 지점을 설정하고 설정된 목표 지점까지 경로를 계획하여 주행하는 과정을 반복한다^[1]. 자율 주행을 활용하는 기술을 위해서는 주행 환경에 대한 지도가 필수적이며, 이동 로봇 탐사 기술은 환경 지도 작성 자동화에 기여한다. 본 논문에서는 투명한 유리 문이 있는 실내 환경에서, 탐사 목표 지점까지의 주행가능성(navigability)을 고려한 라이다(LiDAR: Light Detection And Ranging) 기반²차원 탐사 기술을 제안한다.

이동 로봇 탐사 기술은 다음 세 가지 알고리즘으로 구성된다: 1) 동시 위치 인식 및 지도 작성 2) 다음 탐사 목표 영역 설정 3) 설정된 탐사 목표 영역까지의 경로 계획 및 주행. 이동 로봇 탐사 기술의 가장 중요한 목적은 전체 환경에 대한 지도 작성을 효율적으로 완료하는 것이기 때문에, 기존의 탐사 연구는 위 세 알고리즘들 중 두 번째 다음 탐사 영역 설정 알고리즘

Received : Aug. 5. 2023; Revised : Aug. 25. 2023; Accepted : Aug. 30. 2023

※ This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. NRF-2022R1C1C1010931) and was supported by a grant from Endowment Projects of "Development of core technology for cooperative navigation of multiple marine robots and underwater wireless cognitive network" funded by Korea Research Institute of Ships and Ocean engineering under Grant PES4810

1. Principal Researcher, Korea Research Institute of Ships and Ocean Engineering, Daejeon, Korea (jwchoi@kriso.re.kr)

2. BS Student, Mechatronics Engineering, Kangwon National University, Chuncheon, Korea (th991023@kangwon.ac.kr)

† Associate Professor, Corresponding author: Mechatronics Engineering, Kangwon National University, Chuncheon, Korea (hjryu@kangwon.ac.kr)

개발 위주로 진행되었다. 이를 위해, 지도의 프론티어(frontier) 영역이 활용된다. 프론티어는 지도 작성이 완료된 비점유(unoccupied) 영역과 지도 작성이 완료되지 않은 미지의(unknown) 영역 사이의 경계 지역으로 정의된다²⁾. 프론티어에서는 새로운 영역에 대한 센서 데이터 획득이 보장되어 지도 작성 커버리지(coverage)를 확장시킬 수 있기 때문에, 많은 탐사 알고리즘들은 작성 중인 지도에서 프론티어를 추출하는 과정을 기본적으로 포함한다. 이때, 격자 지도(grid map)에서는 프론티어 영역이 직관적으로 구분되기 때문에, 탐사의 SLAM 프레임워크가 격자 지도를 사용하지 않는 경우에도 SLAM 상태(state) 변수가 추정될 때마다 프론티어 추출을 위해서 추가로 격자 지도를 작성하고 갱신한다^{3,4)}.

기존의 프론티어 기반 탐사 연구는 격자 지도에서 다수의 프론티어 영역을 추출하고 각 프론티어 영역의 유틸리티(utility) 혹은 정보 이득(information gain)을 계산하여, 유틸리티가 높은 방향으로 탐사 목표 영역이나 주행 방향을 설정한다^{5,6)}. 탐사의 SLAM 프레임워크로 Extended-Kalman filter (EKF) 혹은 파티클 필터(particle filter)를 사용하는 경우에는 상태 추정의 불확실성(uncertainty)이 loop-closing 실행에 큰 영향을 받기 때문에, 탐사(exploration) 유틸리티와 활용(exploitation) 유틸리티를 예측하여 다음 목표 지점을 선택하는 연구들이 진행되었다^{7,8)}. 탐사 유틸리티는 새로운 센서 데이터 양측, 확장될 수 있는 새로운 지도 범위가 넓은 곳을 선호한다. 활용 유틸리티는 이전에 관측된 환경 정보를 다시 관측하여 loop-closing이 발생할 수 있는 곳으로 로봇을 이동시키는데, 자율적으로 loop-closing을 유도하기 때문에 active-SLAM 혹은 active-mapping이라는 용어가 함께 사용되기도 한다.

최근에는 이동 로봇 탐사의 SLAM 프레임워크로 graph-SLAM이 주로 사용되고 있다⁹⁻¹¹⁾. Graph-SLAM은 급속히 발전된 컴퓨팅 하드웨어 성능에 따라 높은 수준의 최적화 및 상태 변수 추정 성능을 보여준다. 특히 실시간으로 그래프(graph) 최적화 및 loop-closing이 가능하기 때문에, graph-SLAM을 사용하는 탐사 기술은 탐사 유틸리티 향상에 집중한다. 이를 위해 빠른 프론티어 영역 추출 방법에 대한 연구가 수행되었다^{12,13)}. 로봇의 현재 위치에서 탐색을 시작하는 웨이브프론트(wavefront) 알고리즘을 적용하여, 로봇으로부터 가까운 위치의 프론티어 영역을 순차적으로 빠르게 추출할 수 있다¹²⁾. Graph-SLAM이 다수의 submap을 작성하는 경우, 전역 지도에서 프론티어를 추출하는 것이 아니라 각 submap에서 프론티어를 추출하고 데이터베이스를 구축하여 탐사 유틸리티를 향상시킬 수 있다. 이때, 데이터베이스에 등록된 프론티어의 탐사 완료 여부를 Breadth-First Search(BFS)를 이용하여 갱신하는 연구가 진행되었다^{10,11)}. 이 방법은 전역 격자 지도(global grid map)에서 프론티어를 추출하는 경우 발생하는 전역 격자 지도 크기 증가로 인한

프론티어 추출 속도 저하 문제를 해결하기 위해서 제안되었다.

최근 탐사 알고리즘들이 graph-SLAM의 향상된 추정 성능을 이용하여 빠른 프론티어 영역 추출에 집중하고 있지만, 추출된 프론티어 영역까지의 경로 계획 및 주행에 대한 연구는 비교적 덜 이루어졌다. 또한, 실제 실내 환경에서는 투명한 유리 문과 같이 라이다 센서에 적합하지 않은 재질의 장애물이나 좁은 통로가 존재하는 등 주행가능한 프론티어 영역 추출이 어려울 수 있다. 유리 재질을 고려한 라이다 기반 SLAM 알고리즘들이 개발되기도 했지만, 이들은 강인한 정보 교환(data association)을 위한 특징점(feature) 관점에서 주로 수행되었다. 탐사를 위한 주행 알고리즘에 대한 논의가 덜 이루어진 이유는 경로 계획 자체가 비교적 잘 정의된 독립적인 연구분야로 많은 알고리즘들이 개발되었으며, 특히 Rapidly exploring Random Tree (RRT), Probabilistic Road Map (PRM) 등 샘플링 기반 플래너(sampling-based planner)들이 좋은 성능을 보여주고 있기 때문이다. 하지만, 기존의 샘플링 기반 플래너는 전처리(preprocessing)된 지도 상에서 고정된 출발/목표 지점에 대해 최적 경로를 계산하는 것에 적합하다. 따라서, 지속적으로 갱신되는 지도 상에서 주행가능 영역이 변하는 상황이 고려된 탐사 관점의 경로 계획 알고리즘이 필요하다.

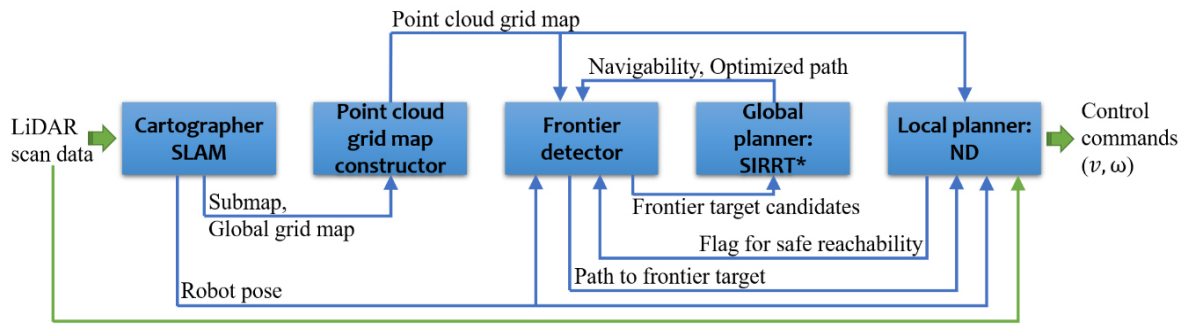
본 논문에서는 2차원 격자 지도에서 추출된 프론티어 영역의 주행가능성을 고려한 라이다 기반 탐사 기술을 제안한다. SLAM 프레임워크로 Cartographer SLAM을 사용한다. Cartographer SLAM이 작성하는 submap을 이용해서 우리가 존재하는 환경에서 안전한 프론티어 영역 추출을 위한 포인트 클라우드(point cloud) 격자 지도를 작성한다. 포인트 클라우드 격자 지도에 Skeletonization-Informed RRT* (SIRRT*)를 적용하여 각 프론티어 영역까지의 주행가능성을 빠르게 확인하는 방법을 제안한다. SIRRT*¹⁴⁾는 격자 지도의 골격화(skeletonization)을 이용하기 때문에, 무작위 샘플링(random sampling)에만 의존하는 다른 RRT 계열 알고리즘에 비해 계산 시간의 편차가 작은 것을 확인할 수 있다.

본 논문은 다음과 같이 구성되어 있다. 2장에서는 Cartographer SLAM의 submap을 이용한 포인트 클라우드 격자 지도 작성 방법과 SIRRT*를 이용한 주행가능성 확인 방법을 소개한다. 3장에서는 제안된 알고리즘을 이용한 실내 환경에서의 탐사 실험 결과를 제시하고 분석한다. 4장에서는 결론을 기술한다.

2. 주행가능성을 고려한 라이다 기반 탐사

2.1 전체 시스템 구성

주행가능성을 고려한 라이다 기반 탐사의 전체 시스템 구성과 각 알고리즘의 입출력 요소는 [Fig. 1]과 같다. Carto-



[Fig. 1] Overall flow of the proposed LiDAR-based mobile robot exploration considering navigability of frontiers

grapher SLAM은 일종의 지역 격자 지도(local grid map)인 submap 데이터베이스를 구축한다¹⁵⁾. 라이다 스캔(scan) 데이터가 획득될 때마다 submap-to-scan matching을 통하여 submap 데이터베이스를 최적화하고 로봇 위치를 추정한다. 또한, submap들을 병합하여 작성한 전역 격자 지도(이하 “Cartographer 전역 지도”)를 일정 주기 마다 갱신한다.

포인트 클라우드 격자 지도는 submap 데이터베이스와 Cartographer 전역 지도를 이용하여 작성되고, 프론티어 디텍터(frontier detector)와 장애물 회피를 위한 로컬 플래너(local planner)로 전달된다.

프론티어 디텍터는 포인트 클라우드 격자 지도 상의 안전한 프론티어 영역들을 추출하고, 각 프론티어 영역의 대표 지점, 즉 탐사 후보지(frontier target candidate)를 선택한다. 현재 로봇 위치에서 각 탐사 후보지까지의 주행가능성은 글로벌 플래너(global planner)인 SIRRT*의 골격화를 이용한 초기 경로 계산 알고리즘에 의해 확인된다. 주행가능성이 확보된 탐사 후보지 중 가장 가까운 탐사 후보지가 다음 탐사 목적지(frontier target)로 선택된다.

선택된 탐사 목적지까지 주행을 위한 전역 경로는 앞서 주행가능성 확인을 위해 생성된 초기 경로를 최적화하여 계산된다. 이때, SIRRT*의 informed sampling 최적화 단계가 적용된다. 최적화된 경로는 로봇의 현재 위치와 탐사 목적지를 연결하는 경유지들로 구성되어 있으며, 로봇은 순차적으로 경유지까지 주행하는 과정을 통해 최종 목적지에 도달할 수 있다.

로봇을 각 경유지까지 이동시키는 제어 입력 생성을 위한 로컬 플래너(local planner)로는 Nearness Diagram (ND)¹⁶⁾가 사용된다. 리액티브(reactive) 기반 로컬 플래너인 ND는 현재 스캔 데이터를 이용해서, 안전 거리가 확보된 주행 가능한 방향을 찾고 선속도(v)와 각속도(ω)를 계산한다. 일반적으로 리액티브 기반 플래너는 안전한 주행 방향을 선택하기 위해 다수의 매개 변수에 의존하며 이들은 경험적으로 결정된다. 반면, ND는 서로 근접한 장애물들의 분포 상황에 따라 안전한 속도를 계산하기 때문에 의존하는 매개 변수의 수가 적어 탐

사를 위한 로컬 플래너로 적합하다. 또한, 로컬 플래너는 포인트 클라우드 격자 지도 상에서 앞으로 추종해야 할 경유지들의 안전성 즉, 장애물에 해당되는 점유 격자로부터의 안전 거리 확보 여부를 확인한다. 경유지들이 안전한 경우 현재 탐사 목적지까지 주행을 계속하고, 안전하지 않은 경우 프론티어 디텍터에게 새로운 탐사 목적지 검출을 요청한다. 위 과정은 전체 환경에 대한 지도 작성이 완료될 때 즉, 포인트 클라우드 격자 지도에서 안전한 탐사 목적지가 추출되지 않을 때까지 반복된다.

2.2 포인트 클라우드 격자 지도 작성

유리가 존재하는 환경에서 안전한 프론티어 영역을 추출하고 탐사 후보지까지의 주행가능성을 확인하기 위하여 포인트 클라우드 격자 지도 작성 방법을 제안한다. 또한, 포인트 클라우드 격자 지도는 최적화된 경로를 따라 실시간으로 주행하면서 탐사 목적지까지 안전하게 도달할 수 있는지(reachability)를 확인하기 위해서도 사용된다. Cartographer SLAM의 각 submap에는 환경 일부분에 대한 센서 정보가 반영되어 있기 때문에 경계 지점에서 점유 정보가 완전하지 않을 수 있다. 또한, submap들을 병합하여 Cartographer 전역 지도의 점유 확률을 갱신하는 과정에서 submap의 점유 격자가 Cartographer 전역 지도에서는 비점유 격자로 갱신될 수 있다. 이런 현상은 주행하고 있는 로봇이 투명한 유리 구조물에 대한 라이다 스캔 데이터를 획득했을 때 주로 나타나는데, 유리 구조물을 어느 방향에서 스캔했는지에 따라서 라이다의 반사도 및 구조물 감지 여부가 달라지기 때문이다. 따라서 불투명한 구조물에 대해서는 반복적으로 획득되는 측정값을 이용해서 격자 지도를 갱신할 때 점유 확률이 점차 증가하는 경향을 보이지만, 투명한 유리 구조물은 다양한 각도에서 획득된 라이다 스캔 데이터가 격자 지도 갱신에 사용되는 경우 비점유 격자로 계산될 수 있다.

본 논문에서는 장애물과 충돌 없는 안전한 탐사를 최우선

목표로 설정한다. 이를 위한 유리 구조물에 대한 점유 확률이 보존되는 포인트 클라우드 격자 지도($gridmap^{PCL}$) 작성 방법은 [Fig. 2]의 Algorithm 1과 같다. $submapDB^{carto}$ 와 $gridmap^{carto}$ 는 Cartographer SLAM으로부터 획득되는 submap 데이터베이스와 Cartographer 전역 지도이다. $gridmap^{PCL}$ 을 $gridmap^{carto}$ 와 같은 크기, 해상도로 초기화한 다음, 모든 격자를 점유 확률이 0인 빈(empty) 격자로 설정한다(line 1-4). 다음 과정에서는 모든 submap에서 유리를 비롯한 장애물에 해당되는 격자를 찾아 $gridmap^{PCL}$ 의 점유 격자로 등록한다(line 5-17). $submapDB^{carto}$ 에 저장된 각 submap($submap^{carto}$)의 격자($s_{p,q}$) 중 점유 확률이 높은 격자들에 대해서(line 5-7), 이들의 $gridmap^{PCL}$ 상의 위치(x^m, y^m)를 찾는다(line 8-9). 해당되는 위치의 $gridmap^{carto}$ 격자($grid_{x^m, y^m}^{carto}$)가 미지의(unknown) 격자가 아닌 경우(line 10) 즉, 라이다 스캔 데이터를 이용하여 갱신된 적이 있는 격자인 경우 확실한 점유 격자로 갱신한다(line 12). 이때, 라이다 스캐

Algorithm 1 Point cloud(PCL) grid map construction

Require:

$submapDB^{carto}$,
 $gridmap^{carto} = \{grid_{m',n'}^{carto} | 0 \leq m' \leq M, 0 \leq n' \leq N\}$
 $\triangleright grid_{m',n'}^{carto}$: each grid in $gridmap^{carto}$
 $\triangleright m'$: column index of $grid_{m',n'}^{carto}$
 $\triangleright n'$: row index of $grid_{m',n'}^{carto}$

Ensure:

$gridmap^{PCL} = \{grid_{m,n}^{PCL} | 0 \leq m \leq M, 0 \leq n \leq N\}$
 $\triangleright grid_{m,n}^{PCL}$: each grid in $gridmap^{PCL}$
 $\triangleright m$: column index of $grid_{m,n}^{PCL}$
 $\triangleright n$: row index of $grid_{m,n}^{PCL}$

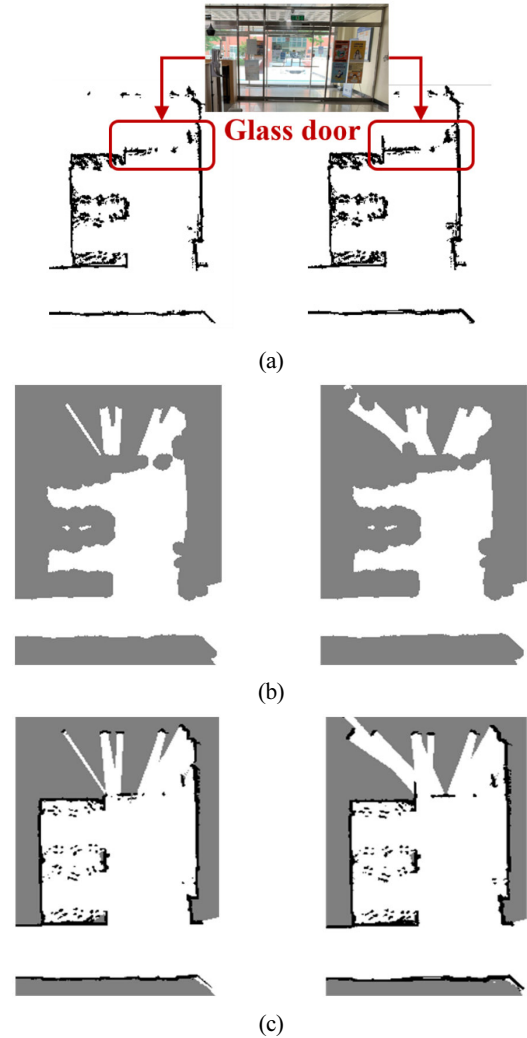
- 1: Initialize $gridmap^{PCL} \leftarrow gridmap^{carto}$;
- 2: for all $grid_{m,n}^{PCL} \in gridmap^{PCL}$ do
- 3: $grid_{m,n}^{PCL} \leftarrow 0$;
- 4: end for
- $\triangleright submap^{carto}$: each submap in $submapDB^{carto}$
- $\triangleright s^{p,q}$: each grid in $submap^{carto}$
- $\triangleright p$ and q : column and row index of $s^{p,q}$
- 5: for all $submap^{carto} \in submapDB^{carto}$ do
- 6: for all $s_{p,q} \in submap^{carto}$ do
- 7: if $s_{p,q} > th_{occ}$ then
- 8: Calculate the metric position (x_p, y_q) of (p, q) ;
- 9: Find the $gridmap^{PCL}$ index (x_m, y_n) of (x_p, y_q) ;
- 10: if $grid_{x_m, y_n}^{carto} \neq th_{unknown}$ then
- 11: for $x_m - 1 \leq i \leq x_m + 1, y_n - 1 \leq j \leq y_n + 1$ do
- 12: $grid_{i,j}^{PCL} \leftarrow 1$;
- 13: end for
- 14: end if
- 15: end if
- 16: end for
- 17: end for

[Fig. 2] Algorithm to construct a point cloud grid map using submaps and a global grid map obtained from Cartographer SLAM

너의 측정 오차를 고려하여 해당 격자뿐만 아니라 인접한 주변 여덟 개의 격자들도 점유 격자로 갱신한다(line 11). 본 연구에서는 th_{occ} 와 $th_{unknown}$ 모두 0.5로 설정하였다.

[Fig. 3(a)]는 포인트 클라우드 격자 지도이며, 검정색 픽셀(pixel)이 점유 확률이 1로 설정된 점유 격자이다. [Fig. 3(b)]는 안전한 프론티어 추출과 주행가능성 확인을 위해 작성된 안전 격자 지도(safety grid map)이며 흰색 픽셀이 안전한 격자이다. 안전 격자 지도는 포인트 클라우드 격자 지도에서 점유 격자를 로봇의 반지름에 해당되는 픽셀만큼 팽창(dilation)시킨 다음, Cartographer 전역 지도와 비교해서 식 (1)에 따라 계산된다.

$$grid_{m,n}^{safety} = \begin{cases} safe, & grid_{m,n}^{carto} = 0 \text{ and } grid_{m,n}^{PCL,dilated} \neq 1 \\ \text{not safe}, & \text{otherwise} \end{cases} \quad (1)$$



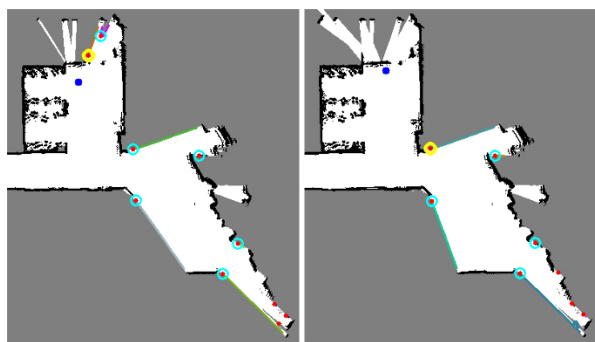
[Fig. 3] Point cloud grid map (a), safety grid map (b), and grid map by Cartographer SLAM (c): the results by 22nd frontier query (left) and by 25th frontier query (right)

팽창된 포인트 클라우드 격자 지도를 사용하기 때문에 로봇은 안전한 것으로 판명된 격자의 위치에서 장애물과 충돌 없이 존재할 수 있다. [Fig. 3(c)]는 Cartographer 전역 지도이며, [Fig. 3]의 왼쪽 그림들은 22번째 프론티어 추출 알고리즘 실행 결과이며, 오른쪽은 25번째 프론티어 추출 알고리즘 실행 결과이다. 포인트 클라우드 격자 지도에서는 닫혀 있는 투명 유리 문 영역에 해당되는 점유 격자가 증가한 것을 확인할 수 있고, Cartographer 전역 지도에서는 오히려 개방된 영역으로 계산된 것을 확인할 수 있다.

2.3 격자 지도의 골격화를 이용한 주행가능성 확인

안전 격자 지도에서 비점유 격자이며 미지의 격자([Fig. 3(b)에서 회색 픽셀]와 인접한 격자들이 안전한 프론티어 격자로 추출된다. 이들은 Connected-component labeling 알고리즘을 이용해서 분할된다¹⁷⁾. 분할된 각 프론티어 세그먼트(segment)의 격자들 중에서 현재 로봇 위치에서 가장 가까운 격자가 대표 격자 즉, 탐사 후보지로 지정된다. [Fig. 4]에서 파란색 원이 로봇의 위치이며, Connected-component labeling 알고리즘에 의해 분할된 프론티어 격자들이 서로 다른 색의 점으로 표시되어 있다. 빨간색 원은 분할된 각 프론티어 영역의 대표 격자인 탐사 후보지를 나타낸다.

탐사 후보지의 주행가능성은 두 가지 기준에 의해 판명된다. 첫 번째는 프론티어 세그먼트가 차지하는 면적이 로봇 단면보다 넓어서 로봇이 프론티어 세그먼트에 위치할 수 있는지 여부이다. 두 번째는 로봇이 현재 위치에서 탐사 후보지까지 장애물과 충돌하지 않고 주행하여 도달할 수 있는지 여부이다. 이는 글로벌 플래너가 로봇의 현재 위치에서 탐사 후보지까지 연결되는 경로를 생성할 수 있는지에 의해 확인될 수 있다. 본 연구에서는 이를 위해 SIRRT*의 격자 지도 골격화를 이용한 초기 경로 생성 단계를 이용한다. [Fig. 4]에서 시안(cyan)색 원은 SIRRT*에 의해 주행가능성이 확인된 탐사 후보



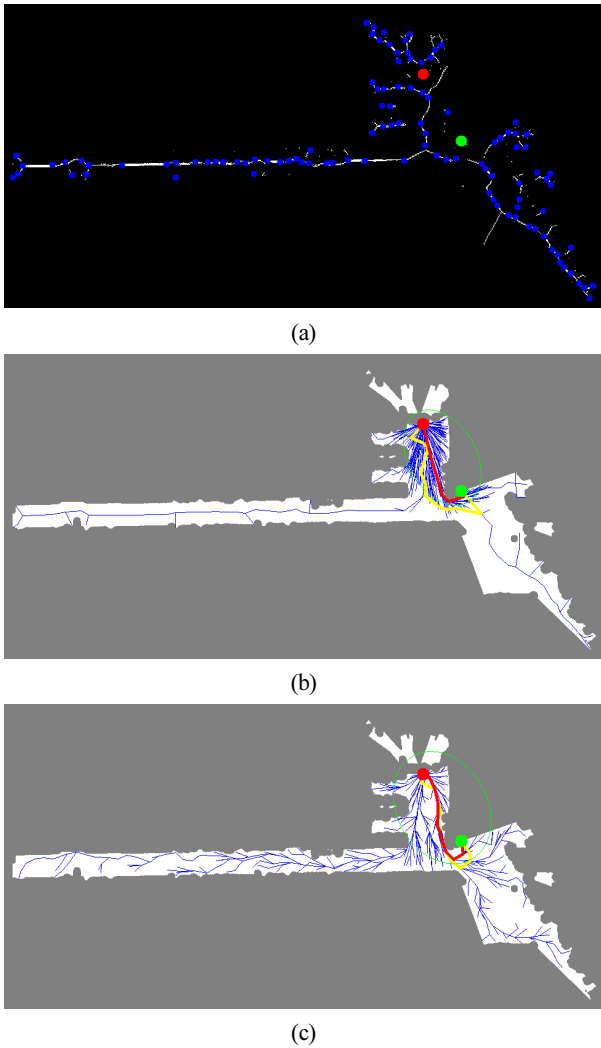
[Fig. 4] Frontier detection results: the results by 22nd frontier query (left) and by 25th frontier query (right)

지를 나타낸다.

SIRRT*는 경로 최적화를 수행하는 RRT* 계열 알고리즘 중 하나인 Informed-RRT* (IRRT*)¹⁸⁾의 초기 경로 계산 성능을 향상시킨 알고리즘이다. SIRRT* 알고리즘은 격자 지도의 골격화를 이용한 초기 경로 생성 단계와 Informed sampling을 이용한 경로 최적화 단계로 구성된다. RRT 기반 알고리즘들은 기본적으로 출발 지점에서부터 임의로 노드(node)를 생성-샘플링(sampling)하여 목표 지점에 닿을 때까지 트리(tree)를 확장한다. 출발 지점부터 목표 지점까지 연결되는 트리가 확장된 이후에는, 경로 최적화를 위해 노드 샘플링을 반복한다. IRRT*는 RRT*를 이용해서 출발 지점과 목표 지점을 연결하는 초기 경로를 생성하고, 생성된 초기 경로 최적화를 위해 샘플링 영역이 타원으로 한정된 Informed sampling을 적용하여 최적화 시간을 줄인다. 그러나 IRRT*는 초기 경로 생성을 위해 임의(random) 샘플링에 의존하는 RRT*를 사용하기 때문에, RRT 기반 알고리즘들이 보이는 초기 경로 생성 지연 문제를 가지고 있다. 즉, 동일한 출발/목표 지점에 대해서도 초기 경로 생성을 위한 샘플링 횟수가 알고리즘이 실행될 때마다 큰 차이를 보이고, 샘플링 횟수를 제한하는 경우 최종 경로 비용 역시 편차가 크다.

SIRRT*는 초기 경로 생성을 위해 환경 지도의 구조적 특성에 의해 결정되는 격자 지도의 골격화를 이용한다. 초기 경로 생성이 임의성에 의존하지 않기 때문에 실행에 따른 편차가 매우 작은 안정적인 성능을 보인다. 이러한 특성은 특히 실시간으로 지도가 확장되며 전체 지도 작성이 완료될 때까지 경로 생성을 반복해야 하는 탐사에 유리하다. 본 논문에서는 현재 로봇 위치에서 각 탐사 후보지까지 주행가능성 확인을 위해 SIRRT*의 골격화를 이용한 초기 경로 생성 단계를 적용하였고, 주행가능성이 확보된 탐사 후보자들 중 선택된 다음 탐사 목적지까지 주행하기 위한 경로 최적화를 위해 Informed sampling을 적용하였다.

[Fig. 5]는 안전 격자 지도 상의 초기 경로 생성 및 최적 경로 생성 결과를 보여준다. 빨간색 원은 로봇의 현재 위치-출발 지점을 나타내고, 녹색 원은 다음 탐사 목적지-목표 지점을 나타낸다. [Fig. 5(a)]는 안전 격자 지도의 골격화 영상이며, 파란색 원은 골격화 영상으로부터 추출된 Harris corner 노드이다. SIRRT*는 이 노드들에 프림 알고리즘(Prim's algorithm)을 적용하여 트리와 초기 경로를 생성한다([Fig. 5(b)]의 노란색 실선). [Fig. 5(b)]의 녹색 타원은 초기 경로 최적화를 위한 Informed sampling의 한정된 샘플링 영역을 보여주며, 빨간색 실선은 1000번의 샘플링을 통해 최적화된 경로이다. 주행할 경로와 가까운 영역에서 트리가 주로 확장된 것을 확인할 수 있다. [Fig. 5(c)]는 IRRT*를 이용한 초기 경로(노란색 실선)와 최적화된 경로(빨간색 실선)이다. IRRT*의 초기 경로 생성은

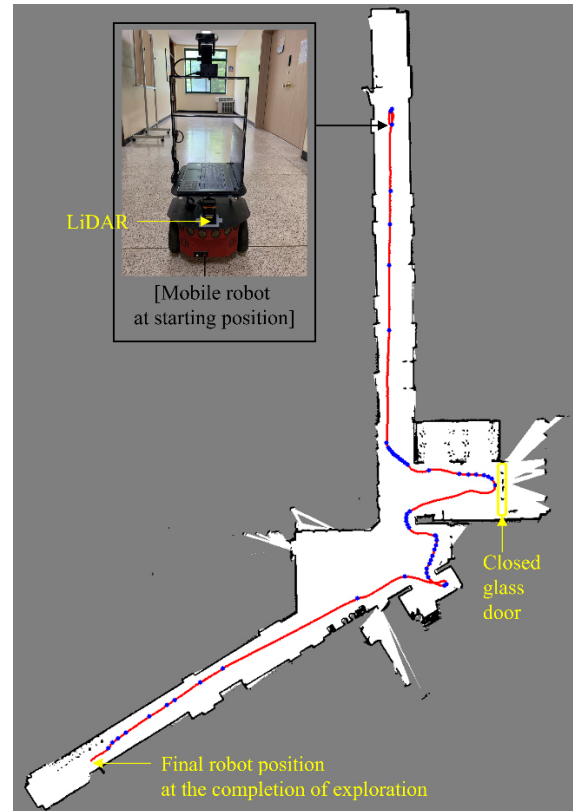


[Fig. 5] Path planning from the robot position to a frontier target position: (a) skeletonization of the safety grid map and extracted nodes by Harris corner detector (b) path planning by Skeletonization-Informed RRT* (SIRRT*) (c) path planning by Informed RRT* (IRRT*)

임의 샘플링에 의존하기 때문에 트리 확장이 주행할 경로와 멀리 떨어진 영역에서도 진행되었으며, 이는 초기 경로 생성을 지연시키는 요인 중 하나이다. [Fig. 5(c)]의 경우 출발 지점부터 목표 지점까지 연결되는 초기 경로 생성까지 1472번의 샘플링이 필요했으며, 최적화를 위한 Informed sampling까지 더하여 모두 2000번의 샘플링을 수행하였다.

3. 실험 결과

투명한 유리 장애물이 있는 실내 환경에서 주행가능성을 고려한 라이다 기반 탐사 기법의 성능 검증을 위한 실험을 수행하였다. 실험에 사용된 로봇은 차동 구동 방식(differential



[Fig. 6] Grid map generated by Cartographer SLAM: red dots and blue dots are the estimated trajectory of a robot, and the robot queried the frontier detection algorithm at the blue dot positions

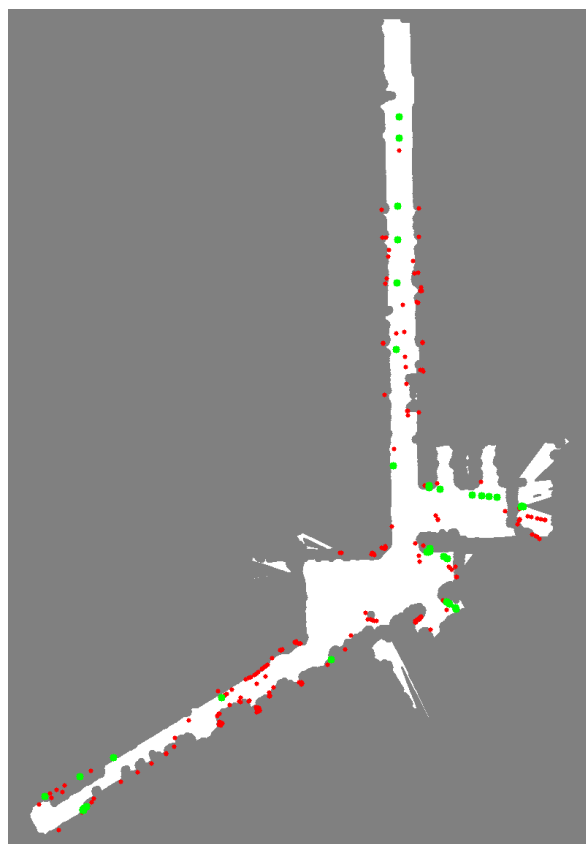
drive)의 Pioneer3-DX이며 바닥에서부터 약 30 cm 높이에 전방 270° 범위를 30 m까지 스캔할 수 있는 2차원 LiDAR (Hokuyo UTM-30LX)를 장착하고 있다. 탐사 알고리즘 실행을 위해 사용된 노트북은 Intel i7-10870H (5.0 GHz) CPU와 32 GB 메모리를 탑재하고 있다. 격자의 크기는 0.05 m이며, 실험 환경의 전체 크기는 약 가로 40 m, 세로 60 m이다. 로봇의 반지름은 0.25 m, 격자 다섯 개에 해당되는 길이로 설정하였다.

유리 장애물에 대한 점유 확률이 보존되는 포인트 클라우드 격자 지도를 이용해서 장애물과 충돌 없이 전 영역에 대한 지도 작성을 완료할 수 있는지 확인하였다.

[Fig. 6]은 탐사가 완료된 시점 즉, 주행가능성이 확보된 탐사 목적지가 더 이상 검출되지 않을 때까지 작성된 Cartographer 전역 지도와 탐사 시작부터 완료까지 로봇의 주행 경로(빨간색 실선)를 보여준다. 주행 경로 위의 파란색 원은 다음 탐사 목적지 선택을 위해 프론티어 추출 알고리즘이 실행된 로봇의 위치를 나타낸다. 파란색 원들이 가깝게 분포되어 있는 부분은 프론티어 추출 알고리즘이 자주 실행되었음을 의미한다. 현재 포인트 클라우드 격자 지도 상에서는 주행가능성이 확보되어 탐사 목적지로 설정되었지만, 이후 갱신된

포인트 클라우드 격자 지도를 이용하는 로컬 플래너에 의해 탐사 목적지까지 도달할 수 없다고 확인된 경우 발생한다. 탐사 과정 중 프론티어 탐사 목적지 선택에 사용되는 환경 정보는 부분적이다. 따라서, 주행 중 획득되는 센서 데이터를 통해 장애물에 대한 점유 확률을 정확히 갱신하여 로컬 플래너에 전달하는 것이 중요하다. 노란색 사각형으로 표시된 영역은 닫힌 유리 문이다. 유리 문 건너 편에 탐사 목적지가 설정되었지만([Fig. 4]의 왼쪽 그림) 최적화된 경로를 따라 주행하는 과정에서 포인트 클라우드 격자 지도가 갱신되었으며, 로컬 플래너는 기존의 탐사 목적지까지 안전하게 주행할 수 없다고 판단하여 새로운 프론티어 검출을 수행하였다([Fig. 4]의 오른쪽 그림).

[Fig. 7]은 탐사가 완료된 시점의 포인트 클라우드 격자 지도에 의한 안전 격자 지도를 보여준다. 탐사 완료 시점까지 모두 58번 프론티어 검출 알고리즘이 실행되었다. 빨간색 원은 탐사 과정 중 검출된 탐사 후보지들이며, 녹색 원은 탐사 후보지들 중 선택되었던 탐사 목적지들이다. 앞서 언급했던 것처럼 탐사 목적지는 부분적인 정보를 이용해서 추출되고, 포인트 클라우드 격자 지도 작성에 사용되는 submap들의 정보는 Cartographer SLAM에 의해 지속적으로 최적화 및 수정되었

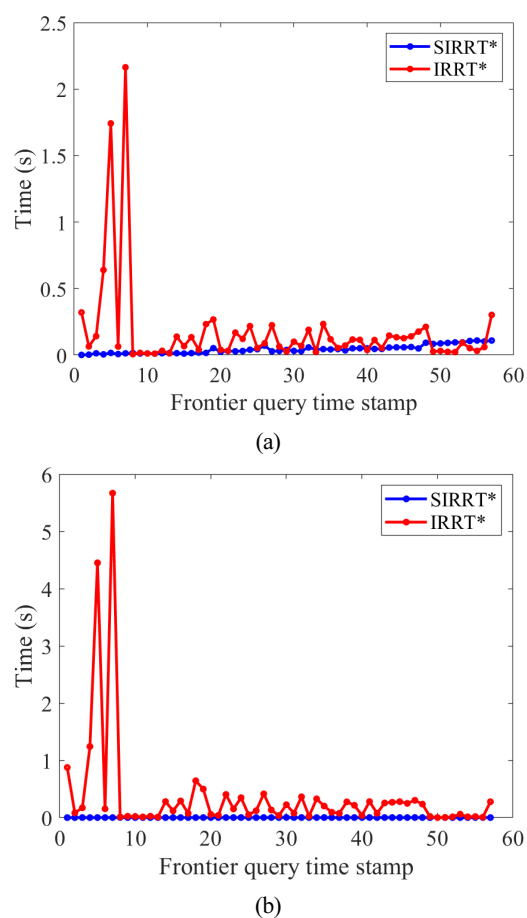


[Fig. 7] Safety grid map and frontiers: red dots are frontier candidates and green dots are selected frontier targets

기 때문에, 탐사 완료 시점의 안전 격자 지도 상에서 탐사 목적지들은 안전하지 않은 영역(회색 픽셀)에 위치할 수 있다.

실험 결과를 통하여, 제안된 포인트 클라우드 격자 지도가 투명 유리 장애물에 대한 점유 확률을 정확히 반영할 수 있으며, 주행가능성을 고려한 탐사 기법을 이용하여 유리 문과 충돌하지 않고 탐사를 완료할 수 있음을 확인하였다.

SIRRT*가 기존의 임의 샘플링에 의존하는 RRT 계열 알고리즘에 비해 안정적으로 주행가능성을 판단할 수 있음을 확인하였다. 탐사 과정 중 작성된 안전 격자 지도, 추정된 로봇의 위치, 그리고 추출된 탐사 후보지 데이터를 이용해서 SIRRT*와 IRRT*의 초기 경로 생성 성능을 비교하였다. SIRRT*는 안전 격자 지도의 골격화를 이용하여 초기 경로를 생성하고, IRRT*는 RRT*를 이용해서 초기 경로를 생성하였다. IRRT* 알고리즘 시작 시, 샘플링 횟수를 1000번으로 제한하고 1000번 샘플링 이내에서 초기 경로 생성에 실패하였을 경우 전체 샘플링 횟수를 1000번씩 증가시켰다. 성능 비교를 위해서 Intel i9-13980HX (5.6 GHz) CPU와 32 GB 메모리를 사용하였다.



[Fig. 8] The mean (a) and standard deviation (b) of computing time to calculate an initial path from the robot position to each frontier target position

[Fig. 8]은 탐사가 완료될 때까지의 각 탐사 목적지 선택 과정에서 소요된 초기 경로 생성 시간의 평균과 표준 편차를 보여준다. 각 탐사 목적지 선택 과정에서 다수의 주행가능성이 확보된 탐사 후보지들이 추출되었으며, 현재 로봇 위치에서 이들까지의 주행가능성 확인을 위해 소요된 시간의 평균과 표준 편차를 계산하였다. SIRRT*는 안전 격자 지도의 골격화 즉, 환경의 구조적 특성을 이용하기 때문에 빠르고 편차가 적은 결과를 보여준다. IRRT*는 소요 시간의 편차가 매우 크다. 각 탐사 목적지 선택 과정 내에서는 동일한 안전 격자 지도가 사용되더라도 IRRT*를 이용하여 주행가능성을 확인하면 목표 지점의 위치에 따른 성능 편차가 큰 것을 의미한다.

SIRRT*는 주행 불가능한 탐사 후보지들에 대해서도 빠르게 주행가능성을 확인할 수 있다. 하지만, IRRT*를 사용하면, 초기 경로 생성이 불가능한 경우 샘플링 횟수가 부족해서 초기 경로가 생성되지 않는지 아니면 주행 불가능한 영역에 목표 지점이 있기 때문에 초기 경로가 생성되지 않는지를 자율적으로 판단할 수 없다. 따라서, 한정된 시간 내에서 빠르게 현재 위치에서 탐사 후보지까지 주행가능성을 확인하기 위해서는 SIRRT*의 골격화를 이용한 초기 경로 생성이 적합하다고 볼 수 있다.

4. 결 론

본 논문에서는 유리 장애물이 있는 환경에서 주행가능성을 고려한 라이다 기반 탐사 기법을 제안하였다. 주행가능성 판단을 위해 포인트 클라우드 격자 지도와 안전 격자 지도를 작성하고, SIRRT*의 초기 경로 작성 과정을 이용해서 주행가능성 확인 방법을 제안하였다. 기존의 탐사 기법들이 격자 지도상의 프론티어 추출 방법 및 다음 목표 지점 혹은 주행 방향 선택에 집중해왔던 반면, 본 논문에서는 로봇이 실제 환경에서 장애물에 부딪히지 않고 안전하게 탐사를 완료하기 위한 주행 관점의 탐사 기법을 논의하였다.

실내 환경 실험을 통해서, 유리 장애물의 점유 확률이 감소되지 않고 반영된 포인트 클라우드 격자 지도 작성 및 안전한 프론티어 영역 추출이 가능하다는 것을 확인하였다. 또한, SIRRT*의 초기 경로 생성 단계를 이용해 빠르고 안정적인 주행가능성 계산이 가능함을 확인하였다.

References

- [1] J. A. Placed, J. Strader, H. Carrillo, N. Atanasov, V. Indelman, L. Carlone, and J. A. Castellanos, "A Survey on Active Simultaneous Localization and Mapping: State of the Art and New Frontiers," *IEEE Transactions on Robotics*, vol. 39, no. 3, pp. 1686-1705, Jun., 2023, DOI: 10.1109/TRO.2023.3248510.
- [2] B. Yamauchi, "A frontier-based approach for autonomous exploration," *IEEE International Symposium on Computational Intelligence in Robotics and Automation*, Monterey, USA, pp. 146-151, 1997, DOI: 10.1109/CIRA.1997.613851.
- [3] J. Vallve and J. Andrade-Cetto, "Potential information fields for mobile-robot exploration," *Robotics and Autonomous Systems*, vol. 69, pp. 68-79, Jul., 2015, DOI: 10.1016/j.robot.2014.08.009.
- [4] C. Stachniss, G. Grisetti, and W. Burgard, "Information gain-based exploration using Rao-Blackwellized particle filters," *Robotics: Science and systems*, Cambridge, USA, pp. 65-72, 2005, DOI: 10.15607/RSS.2005.I.009.
- [5] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, "Past present and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1309-1332, Dec., 2016, DOI: 10.1109/TRO.2016.2624754.
- [6] A. A. Makarenko, S. B. Williams, F. Bourgault, and H. F. Durrant-Whyte, "An experiment in integrated exploration," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Lausanne, Switzerland, pp. 534-539, 2002, DOI: 10.1109/IRDS.2002.1041445.
- [7] H. Lehner, M. J. Schuster, T. Bodenmuller, and S. Kriegel, "Exploration with active loop closing: A trade-off between exploration efficiency and map quality," *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Vancouver, Canada, pp. 6191-6198, 2017, DOI: 10.1109/IROS.2017.8206521.
- [8] S. Oßwald, M. Bennewitz, W. Burgard, and C. Stachniss, "Speeding-up robot exploration by exploiting background information," *IEEE Robotics and Automation Letters*, vol. 1, no. 2, pp. 716-723, Jul., 2016, DOI: 10.1109/LRA.2016.2520560.
- [9] G. Francis, L. Ott, R. Marchant, and F. Ramos, "Occupancy map building through Bayesian exploration," *The International Journal of Robotics Research*, vol. 38, no. 7, pp. 769-792, May, 2019, DOI: 10.1177/0278364919846549.
- [10] J. Oršulic, D. Miklic, and Z. Kovacic, "Efficient dense frontier detection for 2-d graph slam based on occupancy grid submaps," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3569-3576, Oct., 2019, DOI: 10.1109/LRA.2019.2928203.
- [11] Z. Sun, B. Wu, C. Xu, S. E. Sarma, J. Yang, and H. Kong, "Frontier detection and reachability analysis for efficient 2D graph-SLAM based active exploration," *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Las Vegas, USA, pp. 2051-2058, 2020, DOI: 10.1109/IROS45743.2020.9341735.
- [12] M. Keidar and G. A. Kaminka, "Efficient frontier detection for robot exploration," *The International Journal of Robotics Research*, vol. 33, no. 2, pp. 215-236, Feb., 2014, DOI: 10.1177/0278364913494911.
- [13] Z. Sun, B. Wu, C. Xu, and H. Kong, "Ada-detector: Adaptive frontier detector for rapid exploration," *IEEE International Con-*

ference on Robotics and Automation (ICRA), Philadelphia, USA, pp. 3706-3712, 2022, DOI: 10.1109/ICRA46639.2022.9811614.

[14] H. Ryu and Y. Park, "Improved informed RRT* using gridmap skeletonization for mobile robot path planning," *International Journal of Precision Engineering and Manufacturing*, vol. 20, no. 11, pp. 2033-2039, Sept., 2019, DOI: 10.1007/s12541-019-00224-8.

[15] W. Hess, D. Kohler, H. Rapp, and D. Andor, "Real-time loop closure in 2D LIDAR SLAM," *IEEE International Conference on Robotics and Automation (ICRA)*, Stockholm, Sweden, pp. 1271-1278, 2016, DOI: 10.1109/ICRA.2016.7487258.

[16] J. Minguez and L. Montano, "Nearness diagram (ND) navigation: Collision avoidance in troublesome scenarios," *IEEE Transactions on Robotics and Automation*, vol. 20, no. 1, pp. 45-59, Feb., 2004, DOI: 10.1109/TRA.2003.820849.

[17] L. He, X. Ren, Q. Gao, X. Zhao, B. Yao, and Y. Chao, "The connected component labeling problem: A review of state-of-the-art algorithms," *Pattern Recognition*, vol. 70, pp. 25-43, Oct., 2017, DOI: 10.1016/j.patcog.2017.04.018.

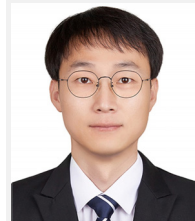
[18] J. D. Gammell, S. S. Srinivasa, and T. D. Barfoot, "Informed rrt*: Optimal sampling-based path planning focused via direct sampling of an admissible ellipsoidal heuristic," *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Chicago, USA, pp. 2997-3004, 2014, DOI: 10.1109/IROS.2014.6942976.



유혜정

2006 포항공과대학교 기계공학과(공학사)
 2014 포항공과대학교 기계공학과(공학박사)
 2015 호쿠리쿠 첨단과학기술대학원대학 (JAIST) 연구조교수
 2016~현재 강원대학교 기계의용·메카트로닉스공학부 부교수

관심분야: 이동 로봇 탐사, 경로 계획, 주행, Simultaneous localization and mapping



최진우

2003 포항공과대학교 기계공학과(공학사)
 2005 포항공과대학교 기계공학과(공학석사)
 2011 포항공과대학교 기계공학과(공학박사)
 2013~현재 선박해양플랜트연구소 책임연구원

관심분야: 자율운항선박, 수중 로봇, 자율 항법



김태현

2018~현재 강원대학교 기계의용·메카트로닉스공학부 학사과정

관심분야: 이동 로봇 주행, 센서 융합