

경유지의 가시성을 고려한 2차원 라이다 센서 기반의 실용적인 경로 계획 프레임워크

유혜정^{1,+}

Practical Path-planning Framework Considering Waypoint Visibility for Indoor Autonomous Navigation using Two-dimensional LiDAR Sensors

Hyejeong Ryu^{1,+}

Abstract

Path-planning, a critical component of mobile robot navigation, comprises both local and global planning. Previous studies primarily focused on enhancing the individual performance of these planners, avoiding obstacles, and computing an optimal global path from a starting position to a target position. In this study, we introduce a practical path-planning framework that employs a target planner to bridge the local and global planners; this enables mobile robots to navigate seamlessly and efficiently toward a global target position. The proposed target planner assesses the visibility of waypoints along the global path, and it selects a reachable navigation target, which can then be used to generate efficient control commands for the local planners. A visibility-based target planner can handle situations, wherein the current, target waypoint is occupied by unknown obstacles. Real-world experiments demonstrated that the proposed path-planning framework with the visibility-based target planner allowed the robot to navigate to the final target position along a more efficient path than the framework without a target planner.

Keywords: Path planning, Obstacle avoidance, Mobile robot, Navigation, LiDAR

1. 서 론

이동 로봇의 주행 기술은 위치 인식, 지도 작성, 경로 계획의 세 가지 세부 기술로 구성된다. 경로 계획은 로봇이 주행하는 환경의 지도와 로봇의 현재 위치 정보가 주어진 경우, 현재/출발 위치부터 최종 목표 지점까지 장애물과 충돌없이 주행하는 것을 목표로 한다. 경로 계획은 글로벌 패스 플래너(global path planner)와 로컬 패스 플래너(local path planner)로 구분된다[1]. 글로벌 플래너는 주어진 환경 지도를 이용해서 출발 위치부터 최종 목표 지점까지 로봇이 따라가야 할 효율적인 전역 경로를 계산하는데, 전역 경로는 장애물과 충돌없이 순차적으로 연결되는 다수의 경유지로 구성된다. 로컬 플래너는 주어진 목표 지점까지 로봇을 이동시키기 위한 제어 입력(선속도와 각속도)을 계산하는 알고리즘으로, 로봇이 주행하는 순간 획득된 현재의 센

서 정보를 이용하기 때문에 지도에 등록된 장애물뿐만 아니라 등록되지 않은 동적 장애물 회피에도 유리하다.

글로벌 플래너를 위해 주어지는 지도의 형태가 주로 격자 지도이기 때문에, 과거에는 격자의 그래프 구조를 직접적으로 활용하는 A* 알고리즘이 주로 사용되었다. 하지만 A* 기반의 알고리즘들은 각 격자마다 주행 비용(cost)을 계산하기 때문에, 환경의 넓이 및 각 격자의 크기에 따라 계산량이 증가한다[2]. 따라서, 최근에는 Rapidly-exploring Random Tree (RRT)와 Probabilistic Roadmap (PRM) 등의 샘플링(sampling) 기반 플래너가 주로 사용되고 있다[3,4]. 특히 RRT 기반의 알고리즘들은 출발 지점부터 트리(tree)를 확장해서 목표 지점까지 연결되는 경로를 빠르게 생성할 수 있고, 트리의 노드를 생성하는 샘플링 과정에서 로봇의 다양한 동적 특성을 반영할 수 있기 때문에 널리 사용된다[5,6].

로컬 플래너는 모션 플래너(motion planner) 혹은 장애물 회피 알고리즘으로도 불리며, 즉각적으로 획득된 현재 센서 데이터를 이용하는 리액티브(reactive) 기반 플래너이다. Dynamic Window Approach (DWA)는 로봇이 순간적으로 원호(circular arc)를 따라 이동한다고 가정하고, 허용가능한 모든 선속도와 각속도들에 대해서 비용을 계산한 다음 최소 비용을 갖는 선속도와 각속도를 선택한다[7,8]. Nearness Diagram (ND)는 안전 거리가 확보된 주행 가능한 방향을 찾고, 근접한 장애물의 기하학

¹강원대학교 기계의용·메카트로닉스공학과 (Department of Mechanical & Biomedical-Mechatronics Engineering, Kangwon National University) Kangwondaehakgil, Chuncheon-si, Gangwon-do, 24341, Korea

⁺Corresponding author: hjryu@kangwon.ac.kr

(Received: May. 20, 2024, Revised: May. 31, 2024, Accepted: Jun. 3, 2024)

This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License(<https://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

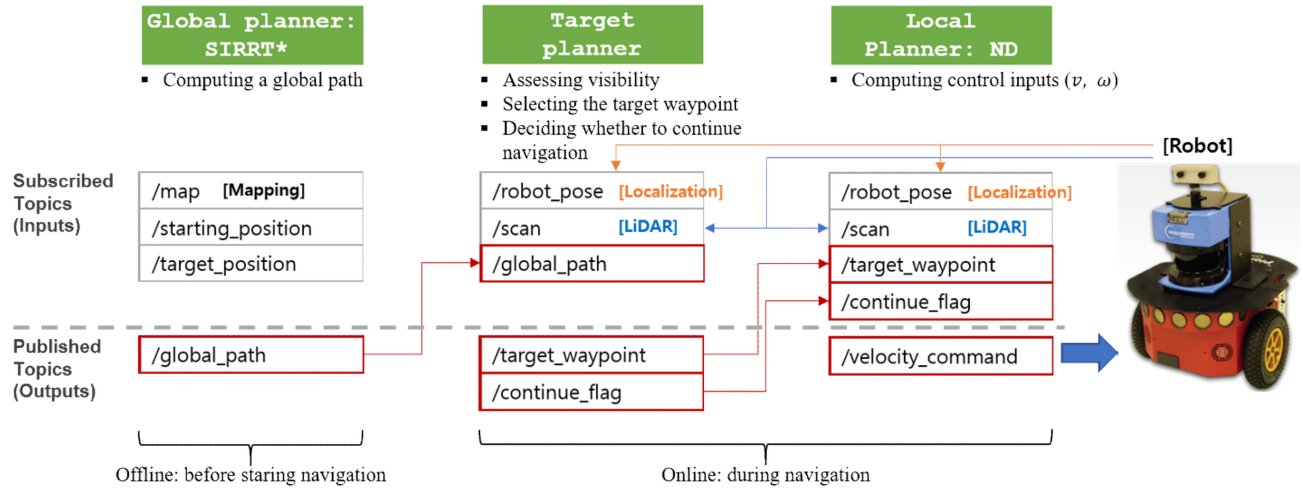


Fig. 1. Overall architecture of the path planning framework using the proposed visibility-based target planner

적 배치 상황에 따라 안전한 제어 입력을 정의하여 계산한다 [9,10]. DWA와 달리 ND는 가능한 제어 입력의 비용을 계산하지 않기 때문에 이를 위해 사용되는 매개 변수의 영향을 받지 않는 장점을 가지고 있다.

샘플링 기반의 글로벌 플래너를 이용해서 출발 지점부터 목표 지점까지 연결하는 경유지들과 이들의 주행 순서를 획득하고, 로컬 플래너를 이용해서 각 경유지들을 방문하는 과정을 반복해서 로봇은 최종 목표 지점까지 도달할 수 있다. 따라서, 효율적인 주행을 위해서는 로봇의 현재 위치를 고려하여 적절한 경유지를 선택하고 이를 로컬 플래너의 입력 값으로 사용하는 것이 중요하다. 따라서, 실시간 자율 주행을 위해서는 글로벌 플래너와 로컬 플래너가 모두 경로 계획 프레임워크에 포함되어야 한다. 기존의 경로 계획 연구들은 글로벌 플래너와 로컬 플래너 각각의 성능 향상에 집중하여 개별적으로 진행되었으며, 경로 계획 프레임워크를 위한 경유지 선택에 대한 연구는 상대적으로 덜 연구되었다. 또한, 구현의 관점에서 단순한 방법을 적용하였는데, 이는 전역 경로의 경유지를 순차적으로 선택하고 로봇이 현재 경유지의 일정 범위 내에 도달하는 경우 그 다음 경유지를 선택하는 방법이다[11].

전역 경로를 생성할 때 사용된 환경 지도로부터 예측되는 거리 값과 로봇의 현재 센서 데이터가 서로 다른 상황은 실제 주행 과정에서 빈번하게 발생된다. 또한, 로봇이 작동하는 환경은 보통 움직이는 장애물이 존재하는 동적(dynamic) 환경이기 때문에, 현재 목표 경유지가 지도에 등록되어 있지 않은 장애물로 가려져 있어서 로봇이 도달할 수 없는 상태이거나 동적 장애물 회피 과정에서 현재 목표 경유지로부터 지나치게 멀어진 상황이 발생할 수도 있다. 따라서, 최종 목표 지점까지 효율적으로 주행하기 위해서는 주행 가능한 경유지로 로컬 플래너의 현재 목표 지점을 변경해야 한다. 경유지 갱신 방법이 적용되지 않는다면, 로봇은 주행 불가능한 곳에 도달하기 위해서 장애물 회피

를 계속 수행하는 등 주행 효율성이 저하될 수 있다.

본 논문에서는 경유지의 가시성(visibility)을 고려하는 타겟 플래너(target planner)를 이용한 경로 계획 프레임워크를 제안한다. 라이다 스캔(scan) 데이터를 이용하여 현재 목표 경유지의 방향 가시성과 거리 가시성을 계산하고, 두 가시성에 따라 현재 목표 경유지를 변경한다. 제안된 타겟 플래너를 이용한 경로 계획 프레임워크의 주행 효율성을 검증하기 위하여 실내 환경에서 실험을 수행하였으며, 이를 타겟 플래너를 사용하지 않는 경우 주행 결과와 비교하였다.

2. 연구 방법

2.1 경로 계획 프레임워크의 전체 구조

본 논문에서 제안하는 경로 계획 프레임워크는 글로벌 플래너, 타겟 플래너, 그리고 로컬 플래너로 구성되며, 전체 구조는 Fig. 1과 같다. 주행을 위한 지도 작성과 위치 인식을 위해서는 별도의 알고리즘이 필요하며, 이들의 출력 요소인 격자 지도(/map)와 지도 상 로봇의 위치 추정 값(/robot_pose)은 경로 계획 프레임워크의 입력 요소로 활용된다. 글로벌 플래너는 로봇이 주행을 시작하기 전에 오프라인(offline)으로 실행되며, 격자 지도, 출발 지점(/starting_position), 그리고 최종 목표 지점(/target_position)을 입력받아 다수의 경유지로 구성된 전역 경로(/global_path)를 계산한다. 타겟 플래너와 로컬 플래너는 전역 경로가 계산된 이후, 로봇이 전역 경로를 따라서 최종 목표 지점에 도달할 때까지 실시간(online)으로 실행된다. 타겟 플래너는 실시간으로 획득되는 라이다 스캔 데이터와 로봇 위치 추정 값을 이용해서 현재 목표 경유지의 가시성을 확인하고, 그 결과에 따라 목표 경유지(/target_waypoint)를 변경한다. 또한, 로

Algorithm 1 Target waypoint selection using visibility

Require: $path_{global} = \{w^i | (w_x^i, w_y^i), 0 \leq i \leq N\}$, w^T , $p^R = (x, y, \theta)$, $scans = \{s_j | \theta_{min} \leq j \leq \theta_{max}\}$
 $\triangleright path_{global}$: a global path consisting of waypoints w^i
 $\triangleright w^0$: the starting position
 $\triangleright w^N$: the final goal/target position
 $\triangleright w^T$: the current target waypoint, $0 < T \leq N$
 $\triangleright p^R$: the pose of the robot
 $\triangleright \theta_{min}, \theta_{max}$: the minimum and maximum angular range of the LiDAR

Ensure: w^T , $flag_{continue}$

- 1: Initialize $flag_{continue} \leftarrow true$
- 2: Calculate the relative transformation $(\Delta\theta, \Delta d)$ between p^R and w^T ;
- 3: **if** $\Delta d > d_{th}$ **then**
- 4: **if** $T < N$ **then**
- 5: Find $d_{scan} = \min scans_{\theta}$;
 $\triangleright scans_{\theta} = \{s_k \in scans | \Delta\theta - \theta_e \leq k \leq \Delta\theta + \theta_e\}$
- 6: **if** $d_{scan} > \Delta d$ **then**
- 7: $v_d \leftarrow true$; $\triangleright v_d$: distance visibility
- 8: **end if**
- 9: **if** $\Delta\theta < \theta_{th}$ **then**
- 10: $v_h \leftarrow true$; $\triangleright v_h$: heading visibility
- 11: **end if**
- 12: **if** $v_h = true$ **and** $v_d = true$ **then**
- 13: **if** $\frac{w^T p^R \cdot w^T w^{T+1}}{w^T} > 0$ **then**
- 14: $w^T \leftarrow w^{T+1}$;
- 15: **end if**
- 16: **else if** $v_h = false$ **and** $v_d = true$ **then**
- 17: **for** $w^t \in path_{global}, T + 1 \leq t \leq N$ **do**
- 18: Calculate $(\Delta\theta, \Delta dist)$ between p^R and w^t ;
- 19: Assess the heading visibility v_h^t for w^t as
in line 5 - line 11;
- 20: **if** $v_h^t = true$ **then**
- 21: $w^T \leftarrow w^t$;
- 22: **break**;
- 23: **end if**
- 24: **end for**
- 25: **end if**
- 26: **end if**
- 27: **else**
- 28: **if** $T = N$ **then**
- 29: $flag_{continue} \leftarrow false$;
- 30: **else**
- 31: $w^T \leftarrow w^{T+1}$;
- 32: **end if**
- 33: **end if**

Fig. 2. Algorithm to select a target waypoint using visibility

봇이 최종 목표 지점에 도달했는지 확인하여, 주행 진행 여부 (“/continue_flag”)를 판별한다. 로컬 플래너는 타겟 플래너에 의해 선택된 목표 경유지를 향해 이동하기 위한 제어 입력 (“/velocity_command”)을 계산하는데, 이때 현재 스캔 데이터 즉, 근접한 장애물까지의 거리 정보에 따른 주행 방향이 활용된다. 본 논문에서는 글로벌 플래너로 Skeletonization-Informed RRT* (SIRRT*)가 사용되었다[12,13]. SIRRT*는 RRT 계열 알고리즘으로, 격자 지도의 골격화를 이용해서 초기 경로 계산 시간 및

편차를 감소시킬 수 있다. 일반적인 RRT 계열 알고리즘들은 초기 경로 계산 및 경로 최적화 과정에서 무작위(random) 샘플링을 사용하기 때문에, 동일한 출발/목표 지점에 대해서도 실행할 때마다 계산 시간의 편차가 큰 단점이 존재한다. 하지만, SIRRT*는 초기 경로 계산을 위해 환경의 구조에 따라 결정되는 골격화를 이용하기 때문에, 초기 경로 계산 시간이 짧고 편차가 매우 작은 안정적인 성능을 보여준다. 로컬 플래너를 위해서는 ND를 활용하였다. ND는 로봇 주변에 존재하는 장애물들 사이의 불연속성에 따라 결정되는 갭(gap)을 활용한다. 로봇이 직면할 수 있는 장애물 배치 상황과 주행 안전성을 현재 존재하는 갭의 특성에 따라 분류하고 로봇의 액션(action)을 계산한다. 그러나, Fig. 1의 경로 계획 프레임워크는 특정 글로벌 플래너와 로컬 플래너에 한정되지 않는 구조이기 때문에, 주행 상황에 맞는 다양한 경로 계획 알고리즘과 함께 활용 가능하다.

2.2 전역 경로 경유지의 가시성

경유지의 가시성은 방향 가시성(heading visibility)과 거리 가시성(distance visibility)으로 정의된다. 방향 가시성은 로봇이 진행하는 정면 방향 즉, 헤딩(heading) 방향을 중심으로 경유지가 얼마나 벗어났는지에 따라 결정된다. 거리 가시성은 경유지 방향으로 장애물이 존재하여 경유지가 가려졌는지 여부로 결정된다. 가시성 판별을 위해 먼저, 현재 로봇의 위치(x, y, θ)와 목표 경유지의 위치(w_x^T, w_y^T)사이의 상대 위치 $(\Delta\theta, \Delta dist)$ 를 계산한다.

$$\Delta\theta = \arctan\left(\frac{w_y^T - y}{w_x^T - x}\right) - \theta \tag{1}$$

$$\Delta dist = \sqrt{(w_x^T - x)^2 + (w_y^T - y)^2} \tag{2}$$

$\Delta\theta$ 가 정해진 역치 값(θ_{th}) 보다 큰 경우 방향 가시성이 확보되지 않았다고 판별한다. 거리 가시성 판별을 위해서는 실시간으로 획득된 라이다 스캔 데이터를 이용한다. 경유지 방향의 스캔 값(d_{scan})이 로봇으로부터 경유지까지의 거리(Δd) 보다 현저히 작은 경우 거리 가시성이 확보되지 않았다고 판별한다. 이때, 로봇의 위치 추정 오차 및 라이다의 측정 오차(θ_e)를 고려하여, $\Delta\theta \pm \theta_e$ 범위의 스캔 데이터 중 가장 작은 값(d_{scan})을 Δd 와 비교한다.

2.3 경유지의 가시성을 고려한 타겟 플래너

타겟 플래너는 현재 목표 경유지의 방향 가시성과 거리 가시성을 고려하여, 목표 경유지를 변경하거나 유지한다. Fig. 2는 가시성을 이용한 타겟 플래너의 경유지 선택 알고리즘을 보여준다. 전역 경로($path_{global}$)는 출발 지점(w^0)부터 최종 목표 지

점(w^N)까지 순차적으로 연결되는 경유지(w^i)로 구성되어 있다. 경유지 선택 알고리즘은, 로봇이 목표 경유지로 이동 중인 경우 목표 경유지의 가시성 판별 및 변경에 대한 과정(line 4-26)과 목표 경유지에 도달한 경우 목표 경유지 변경 과정(line 28-32)으로 구성된다. 로봇으로부터 현재 목표 경유지(w^T)까지의 상대 위치와 스캔 데이터를 이용하여 거리 가시성(v_d)과 방향 가시성(v_n)을 판별한다(line 5-11). 현재 목표 경유지까지의 거리 가시성과 방향 가시성이 모두 확보되었다라든가, 로봇이 지도에 등록되어 있지 않은 동적 장애물 회피를 계속했을 경우 현재 목표 경유지를 지나쳐서 전역 경로상의 다음 경유지(w^{T+1})와 현재 목표 경유지 사이에 로봇이 위치하는 상황이 발생할 수 있다(line 13). 이 경우, 로봇이 현재 목표 경유지로 되돌아가는 것보다 다음 경유지를 향해 주행하는 것이 효율적이다(line 12-15). 혹은 거리 가시성이 확보되었다라든가 방향 가시성이 확보되지 않은 경우(line 16-25), 아직 방문하지 않은 경유지 중(line 17) 방향 가시성이 확보되는 경유지로 목표 경유지를 변경하는 것이 효율적이다. 이를 위해, 앞으로 남은 경유지들의 가시성을 순차적으로 확인하여, 가시성이 확보된 경유지를 찾는다. 이 밖의 상황은, 로봇 주변에 장애물이 근접하여 이를 회피해야 하거나, 반대로 주행 방향에 목표 경유지가 장애물에 가려짐 없이 위치하는 상황으로 현재 목표 경유지를 유지하는 것이 효율적이다. 현재 목표 경유지가 최종 목표 지점이며(line 28), 로봇이 일정 범위 이내에 도달한 경우 주행을 종료한다(line 29).

3. 결과 및 고찰

경유지의 가시성을 고려한 타겟 플래너의 주행 성능 검증을 위해 실험을 수행하였다. 제안된 타겟 플래너 기반의 경로 계획 프레임워크를 이용한 주행 결과와 순차적인 경유지 변경만 적용된 기존의 경로 계획 프레임워크 즉, 타겟 플래너를 사용하지 않은 주행 결과를 비교하였다.

3.1 하드웨어 및 알고리즘 구성

실험에 사용된 로봇은 차동 구동(differential drive) 방식의 Pioneer3-DX이다. 로컬 플래너가 계산하는 제어 입력은 선속도와 각속도이며, 이는 로봇의 내부 제어 알고리즘에 의해 오른쪽과 왼쪽 바퀴 속도로 변환된다. 로봇의 바퀴가 닿는 바닥에서부터 0.3 m 높이에 전방 270° 범위를 30 m까지 스캔할 수 있는 2차원 LiDAR (Hokuyo UTM-30LX) 1대가 장착되어 있다. 로봇의 정면 방향을 중심으로 -135°~+135° 범위를 스캔하도록 설치되어 있는데, 이는 로봇의 움직임이 직진과 회전으로 구성되어 있어 후진하지 않는 특성을 반영한 것이다. 경로 계획 실험과 지도 작성 및 위치 인식을 위해 사용된 노트북은 동일하며, Intel i9-14900 (5.8 GHz) CPU와 64 GB 메모리가 탑재되어 있다.

주행 중 로봇의 실시간 위치 추정을 위해서는 Mobile Robot Programming Toolkit (MRPT) 라이브러리의 파티클 필터(Particle Filter, PF) 기반 위치 인식 알고리즘을 사용하였다[14,15]. 추정된 위치의 정확성에 따라 변하는 파티클 개수의 범위는 150~40000로 설정하였다. 위치 인식 알고리즘과 글로벌 플래너를 위해서는 환경의 격자 지도가 필요하다. Cartographer SLAM [16]을 이용해서 격자 지도를 작성하였고 이때 각 격자의 크기는 0.05 m이다. 전체 알고리즘에서 로봇의 반지름을 0.25 m 즉, 격자 다섯 개에 해당되는 길이로 설정하였다.

본 연구의 로컬 플래너로 채택된 ND 알고리즘을 위해서 사용자가 지정해야 하는 매개 변수는 로봇의 최대 선속도와 최대 각속도, 그리고 로봇이 장애물 회피 중 유지해야 할 장애물까지의 최소 거리인 안전 거리다. 최대 선속도와 최대 각속도를 각각 0.3 m/s와 0.5 radian/s로 제한하였다. 안전 거리는 0.3 m로 설정하였다.

3.2 실내 주행 실험 결과

주행 실험은 복도와 홀로 구성된 실내 환경에서 수행되었다. 실험 환경의 전체 크기는 가로 약 40 m, 세로 약 60 m이며, 복도의 너비는 약 1.9 m~2.4 m이다. 로봇이 전역 경로를 추종해서 주행하는 동안 목표 경유지가 동적 장애물에 의해 가려졌을 때, 제안된 타겟 플래너를 이용한 경로 계획 프레임워크가 효율적으로 새로운 목표 경유지를 설정하는지 확인하였다.

먼저, 글로벌 플래너는 로봇의 현재 위치(출발 지점)와 사용자로부터 입력받은 목표 지점사이의 전역 경로를 계산한다. 글로벌 플래너는 격자 지도 상 최단 거리 경로를 탐색하기 때문에, 격자 지도의 비점유(unoccupied) 영역을 모두 이용한다면, 벽에 인접한 경로 즉, 실제 주행할 때 벽과 충돌하는 경로가 생성될 수 있다. 이를 방지하기 위하여, 격자 지도의 빈 영역에 침식(erosion) 영상 처리를 적용하였다.

Fig. 3은 침식된 비점유 격자 지도로부터 생성된 전역 경로와 경유지를 보여준다. 본 연구에서 글로벌 플래너로 사용된 SIRRT*는 격자 지도의 골격화를 이용해서 초기(initial) 경로를 생성하고, 이 초기 경로를 Informed 샘플링을 이용해서 최적화한다. 노란색 실선은 초기 경로이며, 빨간색 실선은 최적화된 경로이다. 시안(cyan)색 원은 최적화된 경로를 구성하는 경유지를 나타낸다. SIRRT*의 초기 경로는 환경의 골격/구조에 의존하고, 최적화 과정에서 수행되는 샘플링은 초기 경로가 추출된 트리를 확장시킨다. 따라서, 일반적인 RRT 계열 알고리즘 결과와 달리, SIRRT*에 의해 생성된 경유지들 사이의 간격은 일정하지 않다.

제안된 프레임워크와 기존 프레임워크의 성능 비교를 위해, 로봇은 두 실험에서 동일한 전역 경로를 추종하여 주행하였다. 이때, 로봇이 각 경유지로부터 반경 0.2 m 이내에 위치하면 경유지에 도달한 것으로 판단하였다. 로봇이 주행하는 동안 현재 목표 경유지를 의도적으로 가로 막고 이동 경로를 확인하였다. Fig. 4는 주행 중 PF 위치 인식 알고리즘에 의해 추정된 로봇의

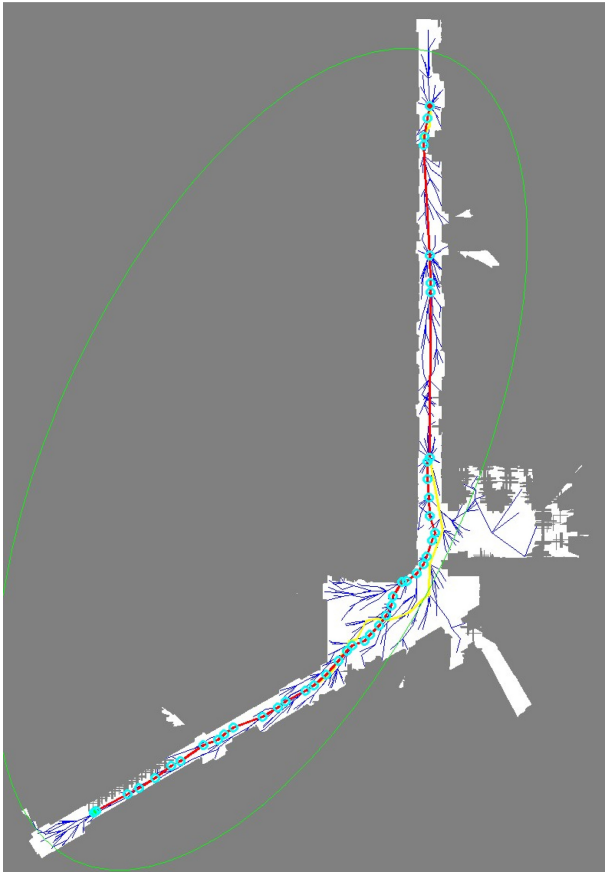


Fig. 3. A global path computed by SIRRT*

위치를 보여준다. 파란색 실선은 제안된 가시성 기반의 타겟 플래너 경로 계획 프레임워크를 이용한 주행 결과이며, 빨간색 실선은 타겟 플래너가 없는 기존 프레임워크를 이용한 주행 결과이다. 시안색 원은 전역 경로의 경유지들이다.

기존의 프레임워크를 사용했을 경우, 로봇은 현재 목표 경유지에 도달할 때까지 장애물 회피를 계속한다. 즉, 현재 목표 경유지가 장애물에 의해 완전히 가려져 있는 경우 장애물이 사라질 때까지 목표 경유지 주변에서 불필요한 이동을 계속한다. Fig. 4에서 녹색 점선 사각형 안의 빨간색 실선은 타겟 플래너를 사용하지 않은 로봇이 목표 경유지들을 향해 주행한 경로를 보여준다. 로봇이 이 영역을 주행하는 동안 경유지들이 장애물에 의해 점유되지 않은 상태를 유지하도록 하였고, 따라서 로봇은 경유지들을 순차적으로 추종하며 주행하였다. 반면, 노란색 점선 사각형 안의 빨간색 실선은 목표 경유지가 장애물에 의해 점유된 경우 비효율적인 로봇의 이동 경로를 보여준다. 타겟 플래너가 적용되지 않았기 때문에 현재 목표 경유지를 그대로 유지하며, 도달할 수 없는 장애물 점유 영역을 향해 회전 주행하였다. 장애물이 사라지고 나서 로봇은 현재 목표 경유지에 도달할 수 있었고, 다음 경유지를 향해 이동하였다.

제안된 가시성 기반 타겟 플래너를 적용한 프레임워크를 사용했을 경우, 로봇은 격자 지도에 등록되지 않은 동적 장애물을

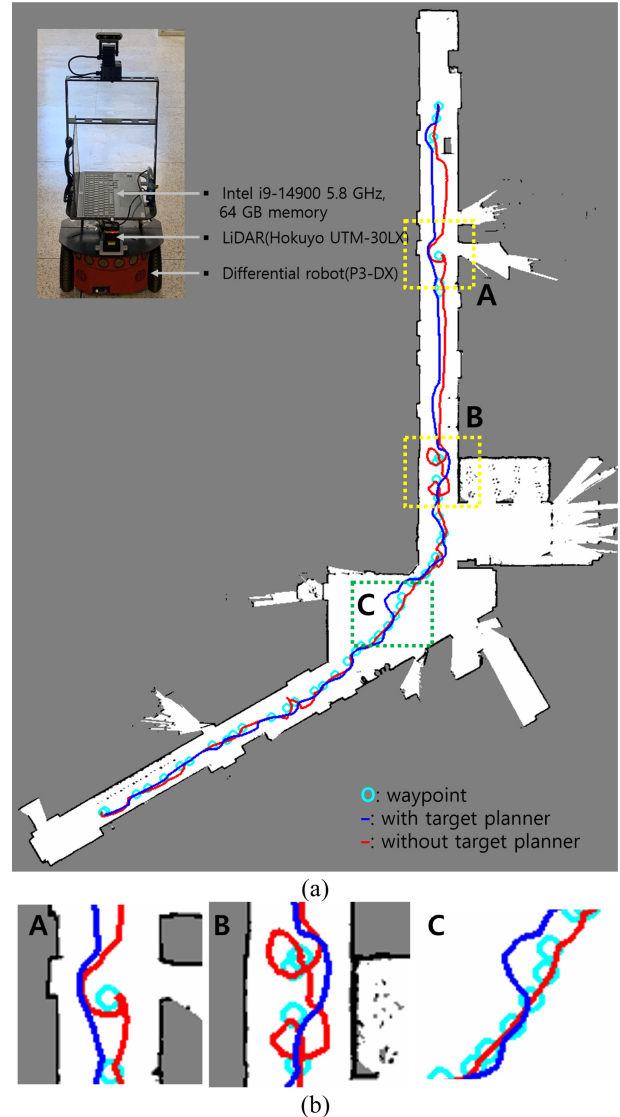


Fig. 4. The estimated position of the robot during navigation ; (a) the estimated position from the start to the final goal, and (b) enlarged views of regions A, B, and C.

회피할 뿐만 아니라 회피 과정에서 현재 목표 경유지의 가시성이 확보되지 않았을 때 가시성이 확보된 다음 경유지로 목표를 변경하여 주행했다. 또한, 현재 목표 경유지가 장애물에 의해 계속 점유된(occupied) 상황에서도 장애물 회피를 수행하며 목표 경유지를 변경하였다. 복도 환경(노란색 점선 사각형의 파란색 실선)과 홀(hall) 환경(녹색 점선 사각형의 파란색 실선)에서 현재 목표 경유지에 장애물을 배치하여 점유된 상태를 유지했다. 타겟 플래너는 현재 목표 경유지가 장애물에 의해 가려진 상황에서도 성공적으로 가시성이 확보된 다음 경유지를 선택하였다. 실제 주행 경로가 글로벌 플래너의 전역 경로에서 벗어난 경우는 목표 경유지 근처의 장애물 회피하여 주행한 결과이며, 전체 주행 경로의 방향은 최종 목표 지점을 향했다.

Fig. 5는 현재 목표 경유지가 점유된 경우 기존 프레임워크와

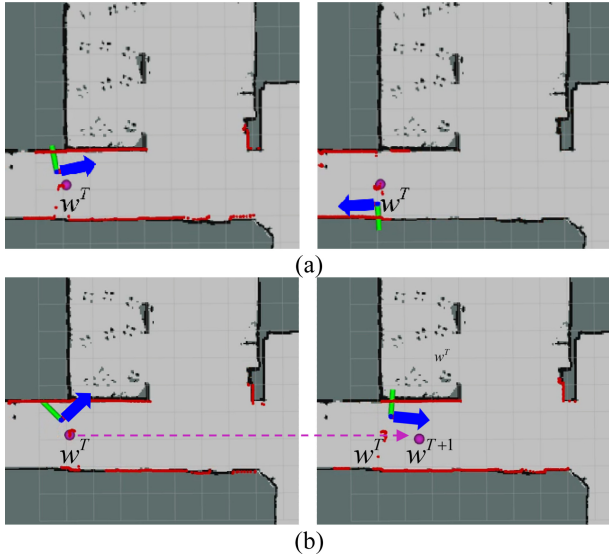


Fig. 5. Local target waypoint selection during navigation ; (a) path planning without a target planner, (b) path planning using the proposed waypoint visibility-based target planner.

제안된 프레임워크의 경로 계획 결과를 보여준다. 파란색 화살표는 로봇의 정면 방향을 나타낸다. 보라색 원은 현재 목표 경유지를 나타내며, 빨간 점들은 라이더 스캔 데이터 즉, 라이더 스캐너에 의해 탐지된 장애물이다. Fig. 5 (a)는 타겟 플래너가 적용되지 않은 프레임워크를 사용했을 때, 로봇이 현재 목표 경유지의 장애물을 회피한 다음(좌) 경유지에 도달하기 위해 주변을 우회하는 상황을 보여준다(우). Fig. 5 (b)는 제안된 타겟 플래너를 적용한 프레임워크에 의해 로봇이 현재 목표 경유지(w^T)의 장애물을 회피하고(좌), 가시성이 확보된 다음 경유지(w^{T+1})로 목표를 변경하여 주행하는 상황을 보여준다(우). 제안된 프레임워크를 사용한 로봇이 목표 경유지의 가시성 확보 여부에 따라 효율적으로 주행하는 것을 확인할 수 있다.

4. 결 론

본 논문에서는 격자 지도에 등록되지 않은 동적 장애물이 존재하는 환경에서 주행을 위한 실용적인 경로 계획 프레임워크를 제안했다. 이를 위해 전역 경로 경유지의 가시성을 이용한 타겟 플래너를 적용하였다. 제안된 타겟 플래너는 로봇이 주행하는 동안 경유지의 점유 여부 혹은 장애물 존재 여부에 따라 결정되는 가시성을 계산하여 성공적으로 현재 목표 경유지를 변경하였고, 이를 로컬 플래너의 목표 지점으로 설정하였다. 타겟 플래너가 적용된 경로 계획 프레임워크를 이용해서, 로봇은 경유지 주변에서 불필요한 이동없이 효율적인 주행 경로를 생성하였다. 따라서, 동적 장애물이 존재하는 환경에서도 최종 목표 지점까지 성공적으로 도달할 수 있었다.

감사의 글

이 성과는 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(No. 2022R1C1C1010931).

REFERENCES

- [1] R. Siegwart, I. R. Nourbakhsh, and D. Scaramuzza, *Introduction to Autonomous Mobile Robots*, The MIT Press, USA, pp. 369-422, 2011.
- [2] K. Fransen and J. van Eekelen, "Efficient path planning for automated guided vehicles using A* (Astar) algorithm incorporating turning costs in search heuristic", *Int. J. Prod. Res.*, Vol. 61, No. 3, pp. 707-725, 2021.
- [3] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning", *Int. J. Rob. Res.*, Vol. 30, No. 7, pp. 846-894, 2011.
- [4] S. Li and N. T. Dantam, "A sampling and learning framework to prove motion planning infeasibility", *Int. J. Rob. Res.*, Vol. 42, No. 10, pp. 938-956, 2023.
- [5] S. M. LaValle and J. J. Kuffner, "Randomized kinodynamic planning", *Int. J. Rob. Res.*, Vol. 20, No. 5, pp. 378-400, 2001.
- [6] 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", *Proc. of 2014 IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, pp. 2997-3004, Chicago, USA, 2014.
- [7] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance", *IEEE Robot. Autom. Mag.*, Vol. 4, No. 1, pp. 23-33, 1997.
- [8] L. Chang, L. Shan, C. Jiang, and Y. Dai, "Reinforcement based mobile robot path planning with improved dynamic window approach in unknown environment", *Auton. Robots*, Vol. 45, No. 1, pp. 51-76, 2021.
- [9] J. Minguez and L. Montano, "Nearness diagram (ND) navigation: Collision avoidance in troublesome scenarios", *IEEE Trans. Rob. Autom.*, Vol. 20, No. 1, pp. 45-59, 2004.
- [10] J. W. Durham and F. Bullo, "Smooth Nearness-Diagram Navigation", *Proc. of IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, pp. 690-695, Nice, France, 2008.
- [11] Y. Li, R. Jin, X. Xu, Y. Qian, H. Wang, S. Xu, and Z. Wang, "A mobile robot path planning algorithm based on improved A* algorithm and dynamic window approach", *IEEE Access*, Vol. 10, pp. 57736-57747, 2022.
- [12] H. Ryu and Y. Park, "Improved informed RRT* using grid-map skeletonization for mobile robot path planning", *Int. J. Precision Eng. Manuf.*, Vol. 20, No. 11, pp. 2033-2039, 2019.
- [13] H. Ryu, "Hierarchical path-planning for mobile robots using a skeletonization-informed rapidly exploring random tree", *Appl. Sci.*, Vol. 10, No. 21, pp. 7846(1)-7864(18), 2020.
- [14] J.-L. Blanco, J. González, and J.-A. Fernández-Madriral, "Optimal filtering for non-parametric observation models: Applications to localization and SLAM", *Int. J. Rob. Res.*,

Vol. 29, No. 14, pp. 1726-1742, Dec. 2010.

[15] <https://docs.mrpt.org/reference/latest/>(retrieved on May. 17, 2024).

[16] W. Hess, D. Kohler, H. Rapp, and D. Andor, "Real-time

loop closure in 2d lidar slam", *Proc. of 2016 IEEE Int. Conf. Robot. Autom. (ICRA)*, pp. 1271-1278, Stockholm, Sweden, 2016.