

포인트 클라우드를 이용한 충돌회피 알고리즘

진영훈¹, 이승화², 김혜경^{3*}

¹백석대학교 첨단IT학부 교수, ²백석대학교 컴퓨터공학부 교수, ³백석대학교 사회복지학부 교수

Collision Avoidance Algorithm Using Point Cloud

Young-Hoon Jin¹, Seunghwa Lee², Hyei-Kyung Kim^{3*}

¹Professor, Division of Advanced IT, Baekseok University

²Professor, Division of Computer Engineering, Baekseok University

³Professor, Division of Social Welfare, Baekseok University

요약 본 논문에서는 유니티 가상공간에서 포인트 클라우드를 활용한 충돌회피 알고리즘을 제안한다. 충돌회피를 위한 알고리즘은 먼저, 카메라의 전방을 기준으로 랜덤하게 레이를 발사하여 포인트 클라우드를 획득하고, 이를 바탕으로 바닥과 그 이외의 영역을 추출한다. 추출된 바닥에서 설정된 목표점까지의 방향 벡터를 생성하고, 이 벡터를 기준으로 가장 먼 포인트까지 갈 수 있는 경로를 A* 알고리즘을 사용하여 구성한다. 이후, 경로상의 각 웨이포인트 waypoint마다 주변 장애물들과의 거리를 계산하여 일정 거리 이하의 웨이포인트를 제거하는 필터링 과정을 거친다. 필터링된 웨이포인트를 따라 에이전트는 이동하며, 이동 중 좌우로 설정된 각도에서 레이를 방출하여 충돌 가능성이 있는 물체를 감지한다. 충돌 물체가 감지되면, 감지된 방향과 반대 방향으로 회전하여 웨이포인트를 따라 계속 이동함으로써 장애물을 회피하며 목표 지점까지 도달할 수 있다. 이 알고리즘은 실시간 환경에서 장애물 회피와 경로 최적화를 결합한 효율적인 경로 계획 방법을 제시하며, 자율주행 및 로봇 내비게이션 시스템에 적용 가능할 것이다.

주제어 : 포인트 클라우드, A* 알고리즘, 웨이포인트, 길찾기, 시뮬레이션, 봇, 사물인터넷

Abstract This paper proposes a collision avoidance algorithm using point clouds in a virtual environment created in Unity. First, random rays are fired from the front of a camera to acquire a point cloud, which is then used to extract the floor and other regions. A direction vector towards the target point is generated from the extracted floor, and based on this vector, a path to the farthest point is constructed using the A* algorithm. Next, each waypoint along the path is filtered by calculating the distance to nearby obstacles, removing waypoints that are too close. The agent follows the filtered waypoints while emitting rays at set angles to the left and right to detect potential obstacles. If an obstacle is detected, the agent rotates in the opposite direction and continues along the waypoint path to avoid the obstacle and reach the target. This algorithm presents an efficient method for combining obstacle avoidance and path optimization in real-time environments, with strong applicability in autonomous driving and robotic navigation systems.

Key Words : PointCloud; A* Algorithm; WayPoint; Pathfinding; Simulation; Bot; IoT;

1. 서론

자율주행 및 로봇 공학 분야에서 실시간 경로 계획과 충돌회피는 필수적인 기술 과제로 대두되고 있다. 이러한 시스템은 주행 중 발생할 수 있는 다양한 장애물들을 효율적으로 인식하고 회피하는 기능을 필요로 한다. 특히, 복잡한 3차원 환경에서의 장애물 회피는 높은 정확도와 빠른 연산 성능을 요구하며, 이를 위해 포인트 클라우드(Point Cloud) 데이터가 널리 사용되고 있다[1-5]. 포인트 클라우드는 3차원 공간에서 객체의 형상과 위치 정보를 효율적으로 표현할 수 있는 데이터 형식으로, 각종 센서에서 수집된 데이터를 기반으로 환경을 정밀하게 분석할 수 있다. 이를 통해 장애물과의 거리를 계산하거나, 경로를 최적화하여 충돌을 피할 수 있는 방법론이 개발되고 있다[6-11]. 본 연구는 이러한 포인트 클라우드 데이터를 활용하여 실시간 경로 탐색과 장애물 회피를 동시에 처리할 수 있는 알고리즘을 제안한다. 유니티(Unity) 가상공간에서 구현된 환경을 사용하였으며, 카메라 전방에 무작위 레이를 발사하여 생성된 포인트 클라우드로부터 바닥 영역 및 장애물 영역을 분리한 뒤, 추출된 바닥 영역에서 설정된 목표점까지의 방향 벡터를 생성하고, A* 알고리즘(Astar)을 활용하여 가장 먼 포인트까지의 최적 경로를 구성한다. 이 경로는 장애물에 가깝지 않도록 장애물들과의 거리를 계산하여 일정 거리 이하인 웨이포인트(Waypoint)를 제거하는 필터링 과정을 거친다. 또한 에이전트가 이동하는 동안 좌우로 레이를 방출하여 충돌 가능성이 있는 물체를 감지하고, 감지된 물체와 반대 방향으로 회전해 충돌을 피하도록 설계하였다. 이를 통해 에이전트는 장애물을 회피하며 목표 지점까지 안정적으로 이동할 수 있다. 제안된 알고리즘은 경로 최적화와 실시간 충돌회피를 동시에 제공하는 데 의의가 있으며, 자율주행 차량 및 로봇 내비게이션 시스템에 적용 가능할 것으로 기대된다.

2. 관련연구

포인트 클라우드를 이용한 경로 계획과 충돌회피 알고리즘은 자율주행 및 로봇 내비게이션 분야에서 오랜 연구 주제 중 하나이다[1-4]. 특히, 실시간으로 동작하는 시스템에서 정확하고 효율적인 경로 탐색과 장애물 회피는 중요한 문제로 여겨지며, 다양한 접근법들이 제안되어왔다.

2.1 포인트 클라우드 기반의 장애물 회피

포인트 클라우드는 레이저 스캐너, 라이다(LiDAR), 스테레오 카메라 등에서 수집된 데이터를 3차원으로 표현하며, 복잡한 환경에서 객체의 모양과 위치를 인식하는 데 유용하다. 최근 연구에서는 이러한 포인트 클라우드를 활용하여 자율주행 차량이 주변 환경을 이해하고, 장애물 회피를 위해 경로를 계획하는 방법들이 발전해왔다. 기존 연구들에서는 포인트 클라우드 데이터를 기반으로 한 경로 탐색 알고리즘에 A*나 D* Lite와 같은 전통적인 경로 탐색 알고리즘을 결합하여 보다 정교한 경로 최적화와 장애물 회피를 시도하고 있다[13-16].

2.2 A* 알고리즘을 이용한 경로 탐색

A* 알고리즘은 그래프 기반 경로 탐색에서 많이 사용되는 휴리스틱 기법으로, 시작점에서 목표 지점까지 최적의 경로를 찾는 데 널리 사용된다. 자율주행 및 로봇 시스템에서도 A* 알고리즘은 실시간 경로 탐색에 매우 적합하다. 최근 연구들은 포인트 클라우드 데이터를 A* 알고리즘에 적용하여, 3차원 공간에서 장애물을 효과적으로 피하면서 경로를 최적화하는 방법을 제안하고 있다[13-15]. 예를 들어, Liang Chu의 연구는 복잡한 도시 환경에서 실시간으로 경로를 계획하고 장애물을 회피할 수 있는 A* 알고리즘의 변형을 제안하였다[13].

2.3 충돌회피와 경로 필터링

경로 계획에서 충돌회피는 필수적인 요소이다. 포인트 클라우드를 사용한 경로 탐색에서는 경로상에 위치한 장애물과의 거리를 계산하여 이를 회피하는 방법이 자주 사용된다. 예를 들어 인간의 움직임이나 환경의 변화를 실시간으로 감지하고, 이를 바탕으로 경로를 적응적으로 조정하기 위해 다양한 필터를 사용한다[17-18]. 이를 통해 경로가 안전한지 평가하고 불필요한 경로를 배제함으로써 보다 최적화된 경로를 생성할 수 있다.

2.4 레이캐스팅 기반 충돌 탐지

레이캐스팅(Raycasting)은 3차원 공간에서 객체와의 충돌을 감지하기 위해 광범위하게 사용된다. 특히, 유니티와 같은 가상 시뮬레이션 환경에서는 에이전트가 이동하는 동안 레이를 발사하여 장애물을 탐지하고, 감지된 방향에 따라 에이전트의 경로를 조정하는 방식이 적용된다. 이러한 방식은 실시간으로 동작할 수 있어 자율주행 차량이나 로봇 내비게이션 시스템에서 충돌을 회피하는

데 매우 유용하다[19-21].

2.5 포인트 클라우드 기반 로봇 내비게이션

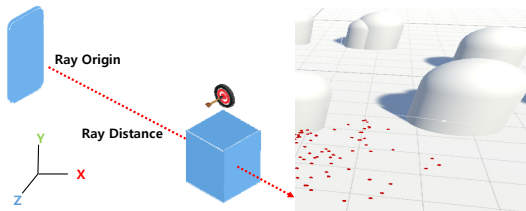
포인트 클라우드는 3차원 공간의 객체 형상과 위치 정보를 정밀하게 표현할 수 있는 데이터 형식으로, 자율주행 및 로봇 내비게이션에 널리 활용되고 있다[10-12]. 특히, 자율주행 차량에서 포인트 클라우드는 복잡한 도심 환경에서도 장애물의 위치를 정확히 감지하고, 경로를 실시간으로 최적화하는 데 사용된다. 이와 함께 무인 항공기(UAV)나 로봇 청소기 등 다양한 로봇 시스템에서도 포인트 클라우드 기반 충돌회피가 적용되어 있으며, 복잡한 환경에서도 안전한 이동이 가능하도록 돕는다. 이러한 기술은 자율주행 및 로봇 공학 분야에서 지속적으로 연구되고 있으며, 본 논문에서는 기존 연구를 바탕으로 실시간 환경에서 효율적인 충돌회피와 경로 계획을 결합한 새로운 알고리즘을 제안한다.

3. 본론

본 논문에서는 유니터 가상공간에서 획득한 포인트 클라우드를 기반으로 한 충돌회피 알고리즘 PCAA(Point Cloud-based Collision Avoidance Algorithm)를 제안한다. 알고리즘의 전반적인 흐름은 다음과 같다.

3.1 포인트 클라우드 생성

PCAA 알고리즘은 먼저 가상공간에서 카메라를 중심으로 레이캐스팅을 통해 포인트 클라우드를 생성한다. 카메라의 전방에 무작위로 레이를 발사하여 장애물 및 지형의 포인트 데이터를 [Fig. 1]과 같이 수집한다. 이렇게 생성된 포인트 클라우드는 장애물 회피를 위한 기반 데이터를 제공한다.



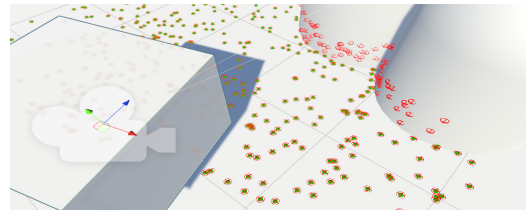
[Fig. 1] Pointcloud using Raycasting(left : operation of raycasting, right : pointcloud generation using raycasting)

3.2 바닥과 장애물 구분

이후 PCAA는 획득된 포인트 클라우드는 바닥과 그 외 영역으로 분류된다. 이를 위해 RANSAC(Random Sample Consensus) 알고리즘을 적용하여 평면에 해당하는 바닥 부분을 추출하고, 나머지 부분은 장애물로 간주한다. 이 과정은 지형을 이해하고 경로 탐색에서 사용될 수 있는 유효한 바닥 영역을 확인하는 데 필수적이다.

$$Dist(P_i) = \frac{|ax_i + by_i + cz_i + d|}{\sqrt{a^2 + b^2 + c^2}} \quad (1)$$

유효한 바닥 영역은 식 (1)을 통해 평면으로부터 각 점의 거리를 계산하여 일정 범위 이내의 점들을 유효한 바닥으로 확정하는 방식이다. 식 (1)을 반복적으로 수행하여 가장 많은 점이 속하는 평면을 최적 평면으로 간주한다.



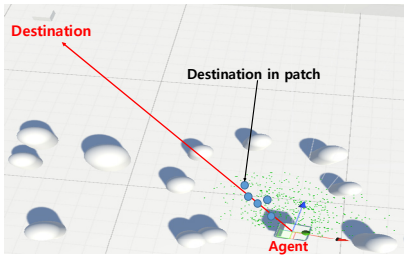
[Fig. 2] segmentation of plane and non-plane points (red wireframe : original points, green mesh : plane points)

3.3 경로 탐색

PCAA 알고리즘은 바닥 영역에서 목표 지점까지의 최적 경로를 탐색하기 위해 A* 알고리즘을 사용한다. A* 알고리즘은 식 (2)와 같이 휴리스틱 함수와 실제 비용을 결합하여 최적의 경로를 찾는 방법으로, 목표를 향하는 방향 벡터 v 를 기준으로 가장 먼 포인트까지 이동 가능한 경로를 A* 알고리즘을 통해 탐색한다. 이 과정은 [Fig. 3]과 같다. 이때 식 (3)을 통해 목표 방향을 기준으로 시작점에서 가장 먼 포인트를 계산하여 포인트 클라우드에서 목표점을 설정한다.

$$f(n) = g(n) + h(n) \quad (2)$$

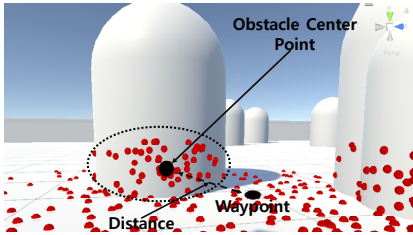
$$u \cdot v = |u| |v| \cos\theta \quad (3)$$



[Fig. 3] Pathfinding towards a destination within a point cloud representing the floor

3.4 웨이포인트 필터링

A* 알고리즘을 통해 생성된 경로는 일련의 웨이포인트로 구성되며, 각 웨이포인트는 장애물과의 거리를 기준으로 필터링된다. 다시 말해, 각 웨이포인트마다 주변의 장애물들과의 거리를 계산하여 설정된 최소 거리 이하의 웨이포인트는 제거된다. [Fig. 4]는 이러한 과정을 묘사하는데 장애물과의 거리 계산은 장애물 포인트들의 중심 좌표를 이용한다.

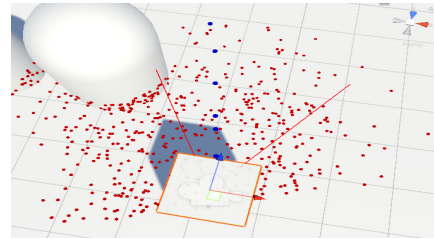


[Fig. 4] Waypoint filtering process using the distance between obstacles and waypoints

[Fig. 4]의 과정을 통해, 경로상에서 안전한 지점만을 유효한 경로로 구성하여 경로의 안전성을 보장한다.

3.5 충돌회피

에이전트는 필터링된 웨이포인트를 따라 이동하며, 이동 중 일정 각도($\pm\theta$)로 레이를 발사하여 실시간으로 장애물을 감지한다. 충돌이 예측되면, 에이전트는 감지된 물체의 방향과 반대 방향으로 회전하여 경로를 따라 계속 이동한다. [Fig. 5]는 충돌 회피방식을 묘사하는데 에이전트의 중심을 기준으로 좌우 30°로 레이를 발사하여 충돌 여부를 확인한다. [Fig. 5]에서는 좌측에 장애물을 감지하였으며 이때 우측으로 30°를 회전하여 진행하면 경로 우측으로 비껴나며 장애물을 안전하게 회피동할 수 있다.

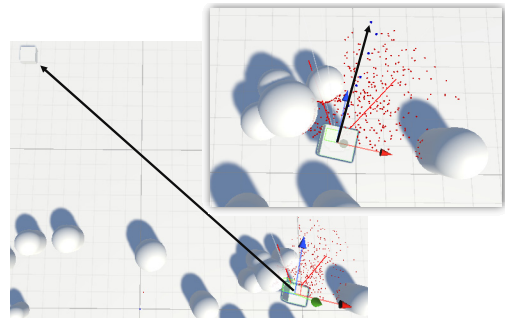


[Fig. 5] Stable obstacle avoidance maneuver Using left and right raycasting in a dynamic environment

이러한 충돌회피 방식은 실시간 환경에서 동적인 장애물을 처리할 수 있도록 설계되었다.

3.6 최종 경로 계획

PCAA 알고리즘은 장애물 회피와 경로 탐색 과정을 반복하여, 에이전트는 최종적으로 목표 지점까지 도달한다. 이 알고리즘은 경로의 최적화뿐만 아니라, 실시간으로 발생하는 장애물을 효율적으로 회피할 수 있는 유연성을 제공한다. 이때 목표로 이동할 경로를 일정 횟수 동안 확보하지 못한다면 에이전트의 앞 방향을 중심으로 최적경로를 다시 계산한다. [Fig. 6]은 이러한 과정을 묘사한다.



[Fig. 6] Optimal Pathfinding Through Target Direction Adjustment

[Fig. 6]은 목표 방향을 중심으로 이동 경로를 탐색할 때 20회 이상 실패할 경우 에이전트의 앞 방향을 중심으로 이동 경로를 재탐색하여 최적경로를 확인하는 과정을 나타낸다. 이 알고리즘은 자율주행 및 로봇 내비게이션 시스템에서 실시간 장애물 회피와 경로 최적화 문제를 해결하거나, 복잡한 환경에서도 안정적인 경로 계획을 제공할 수 있다.

4. 실험

4.1 실험 환경

본 실험은 유니티 2022.3.22f1 버전과 C#을 사용하여 진행하였으며, PC 환경은 Microsoft Windows 11, Intel i7-12700H 프로세서, 16GB RAM, NVIDIA RTX 3060 Laptop 그래픽 카드 사양을 기반으로 한다. 에이전트 크기는 가로·세로 1m로 설정하였으며, 장애물 지대는 랜덤 장애물과 고정 장애물 환경으로 구분하였다. 랜덤 장애물은 에피소드마다 30개의 물체를 무작위로 배치하였고, 고정 장애물은 간격을 최소 1.5m에서 최대 3m까지 유지하였다. 목표 지점과 에이전트의 시작 위치는 무작위로 배치되었다. 알고리즘은 시도 횟수가 2회를 초과하면 경로를 자동 재설정하며, 재설정 횟수가 50회를 넘거나 충돌이 발생하면 에피소드를 종료한다.

4.2 알고리즘 구성

본 논문에서 제안하는 PCAA 알고리즘은 <Table 1>의 의사코드를 따른다.

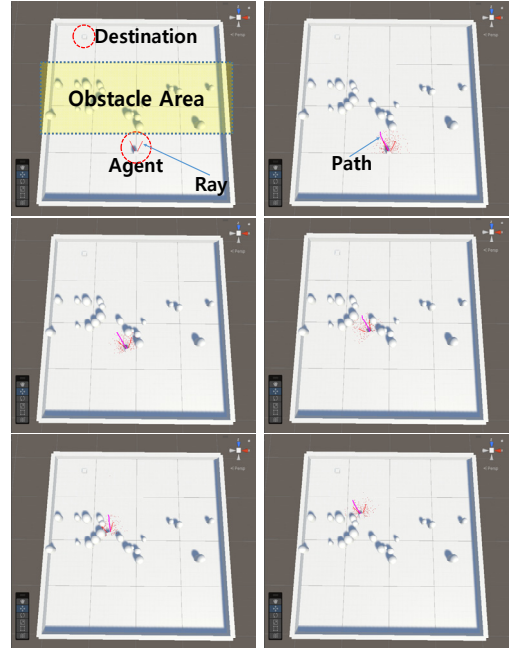
<Table 1> Pseudocode of the proposed algorithm

Seq.	Algorithm
1	$pcd = \{p_i \mid p_i \in R^3\}$
2	$\pi : ax + by + cz + d = 0$ $O : \pi LSUP \neg \forall$
3	$v = G - S$ if tryCount then $v = fv_{agent}$
4	$f(n) = g(n) + h(n)$ $W = \{w_1, w_2 \dots w_k\}$
5	$O' = A VG(O_i)$ $d_i = \ w_i - o_i\ $
6	$W' = \{w_i \in W \mid > d_{th} \circ LSUP \forall_j \in O'\}$
7	$r_i = q_i \times fv_{agent}$ $r_i \cdot o_j = \ r_i\ \ o_j\ \cos \theta$
8	$\theta = \cos^{-1} \left(\frac{fv_{agent} \cdot o_j}{\ fv_{agent}\ \ o_j\ } \right)$

먼저, (1)에서 에이전트 전방에 무작위로 레이를 발사하여 포인트 데이터를 획득한다. 이렇게 획득된 포인트 클라우드는 (2)에서 바닥 평면과 장애물을 분리하기 위해 RANSAC 기반 평면 추출을 수행한다. 바닥 영역이 결정되면, (3)에서 설정된 목표 방향을 기준으로 (4)에서 A* 알고리즘을 적용하여 웨이포인트를 생성한다. 이때 목표 방향은 일정 횟수 이상 경로를 찾지 못할 경우 에이전트

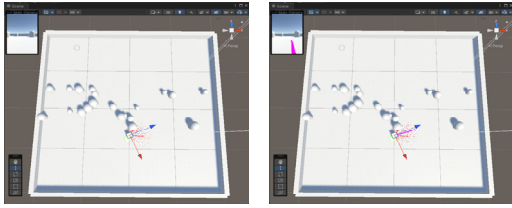
의 정면 방향으로 재설정된다. 생성된 웨이포인트 각각은 (5)에서 장애물들과의 거리를 계산하여, 최소 안전 거리를 만족하지 못하면 (6)에서 선별된다. 이후 (7)에서 에이전트가 이동하는 동안 일정 범위의 레이캐스팅으로 충돌체를 감지하고, (8)에서 충돌 물체가 확인되면 반대 방향으로 회전하여 진행함으로써 안전하게 회피 동작을 수행한다.

4.3 실험 결과



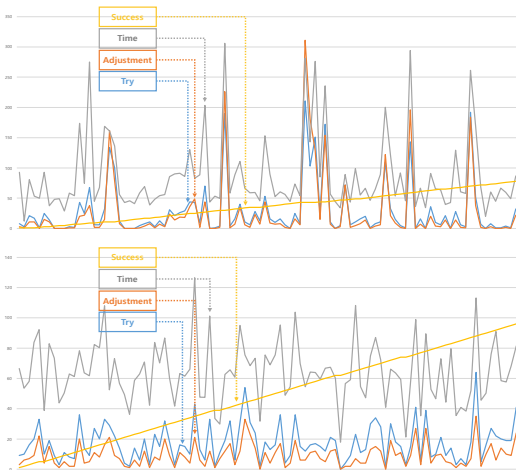
[Fig. 7] Implementation and Execution of the Proposed Algorithm(Proceed along the optimal path from the top-left to the right in sequence)

[Fig. 7]은 <Table 1>을 수행한 결과이다. 여기서 가장 왼쪽 그림은 실험을 위한 가상공간으로, 상단부·중단부·하단부로 구성된다. 상단부에는 목표 지점이 위치하며, 중단부는 장애물 영역을 배치하였다. 하단부는 에이전트의 시작 위치가 된다. 에이전트가 포인트 데이터를 수집하면 두 번째 그림처럼 경로 탐색이 완료되어 진행 방향이 설정된다. 이 과정을 반복함으로써 목표 지점까지 회피기능이 가능하게 된다. 이때 최적경로를 확보하지 못한다면 <Table 1>의 순서(3)에 해당하는 시도 횟수별 목표 방향 설정 [Fig. 8]같이 조정하여 이동 경로를 재설정한다.



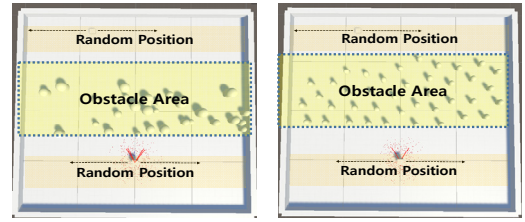
[Fig. 8] Setting the target direction based on the number of attempts

본 논문에서 제안하는 PCAA 알고리즘은 가상 환경에서 대다수의 장애물을 회피하며 최적 경로를 탐색할 수 있다. 그러나 전체 맵을 사전에 완전히 파악하지 못하기 때문에, 모든 환경에서 목표 지점까지의 이동을 항상 보장할 수 있는 것은 아니다. 이러한 한계를 확인하기 위해 [Fig. 9]와 같이 100회 반복 실험을 수행하여 알고리즘의 성공률을 평가하였다.



[Fig. 9] Success rate for 100 attempts(above : random obstacle configuration(success 78%), below : fixed obstacle configuration(success 96%))

[Fig. 9]의 상단 그림은 랜덤한 장애물 환경에서의 실험 결과를 나타낸다. 랜덤 장애물 환경에서 성공률은 78%로, 고정 장애물 환경(하단)의 성공률(96%)에 비해 낮게 나타났다. 또한, 랜덤 장애물 환경에서는 수행 시간, 경로 재설정 횟수, 시도 횟수가 고정 장애물 환경에 비해 상대적으로 높았다. 이러한 결과는 랜덤 장애물 환경에서 목표 지점까지의 경로가 존재하지 않거나, 장애물이 지나치게 복잡하게 배치된 상황에서 발생한 것으로 판단된다. 이를 보완하여 [Fig. 10]은 랜덤 장애물과 고정 장애물의 구체적인 형태를 나타낸다. 이는 장애물 환경의 차이가 알고리즘 성능에 미치는 영향을 분석하는 데 기여한다.



[Fig. 10] The shape of the obstacles used in the experiment(left : random obstacles, right : fixed obstacles)

4.4 개선 사항

본 논문에서 제안한 PCAA 알고리즘은 가상 환경에 국한되었기 때문에, 실제 환경에서의 적용에는 다음과 같은 한계가 존재할 가능성이 있다. 첫째, 실제 환경에서는 센서 데이터의 품질이 가상 환경보다 낮아질 수 있다. LiDAR나 카메라로 획득한 포인트 클라우드는 조명, 날씨, 표면 반사율 등 외부 요인으로 인해 잡음(noise)이나 데이터 누락이 발생할 가능성이 있다. 이는 바닥 및 장애물 분할 과정의 정확도를 저하해 알고리즘 성능에 영향을 줄 수 있다. 둘째, 복잡한 장애물 환경에서는 경로 탐색 실패율이 증가할 가능성이 있다. 랜덤 장애물 환경의 성공률이 78%로 낮고 경로 재설정 횟수가 많은 것은, 경로가 존재하지 않거나 장애물이 지나치게 복잡한 상황에서 발생한 결과로 해석된다. 실제 환경에서도 비정형 장애물 또는 동적 장애물로 인해 유사한 문제가 발생할 수 있다. 이러한 문제를 극복하기 위해 센서 데이터의 품질을 보정하는 노이즈 제거 알고리즘과 데이터 보간법을 적용할 필요가 있다. 또한, 복잡한 장애물 환경에서 경로 탐색 성능을 개선하기 위해 A* 알고리즘에 동적 장애물 회피 기능을 추가하거나, 강화학습과 같은 머신 러닝 기반 접근법을 결합하는 방법을 고려할 수 있다. 이와 같은 개선 방향은 실제 환경에서 발생할 수 있는 잠재적 문제를 해결하기 위한 초기 제안으로, 향후 연구를 통해 구체화하고 실험적으로 검증할 계획이다.

5. 결론

본 연구에서는 유니티 가상공간에서 포인트 클라우드를 활용한 충돌회피 알고리즘을 제안하였다. 제안된 알고리즘은 다음과 같은 주요 단계로 구성된다. 첫째, 카메라의 전방을 기준으로 랜덤하게 발사한 레이를 통해 포인트 클라우드를 획득하고, 이를 기반으로 바닥과 그 이

의 영역을 구분한다. 둘째, 바닥에서 설정된 목표까지의 방향 벡터를 생성하고, A* 알고리즘을 사용하여 목표까지의 경로를 구성한다. 셋째, 경로상의 웨이포인트를 필터링하여 장애물과의 거리가 기준값 이하인 웨이포인트를 제거한다. 마지막으로, 에이전트는 필터링된 웨이포인트를 따라 이동하며, 이동 중 레이를 방출하여 충돌 가능성을 탐지하고, 충돌 물체가 감지될 경우 해당 방향과 반대 방향으로 회전하여 장애물을 회피한다. 이 알고리즘은 실시간 환경에서의 장애물 회피와 경로 최적화를 효과적으로 결합한 방법을 제시하며, 자율주행 시스템 및 로봇 내비게이션에 유용한 기술적 기초를 제공한다. 실험 결과, 제안된 방법은 복잡한 환경에서도 안정적으로 장애물을 회피하며 목표 지점까지 도달할 수 있는 성능을 보였다. 향후 연구에서는 다양한 환경 조건과 장애물 유형에 대한 추가 실험을 통해 알고리즘의 보편성과 안정성을 더욱 강화할 것이다. 또한, 이 알고리즘을 실시간 응용 프로그램에 통합하여 실세계 문제 해결에 기여할 수 있는 방향으로 발전시킬 수 있을 것이다.

REFERENCES

- [1] Y. Peng, D. Qu, Y. Zhong, S. Xie, J. Luo, and J. Gu, "The obstacle detection and obstacle avoidance algorithm based on 2-d lidar," in Proc. 2015 IEEE Int. Conf. Information and Automation, Lijiang, China, Aug. pp.1648-1653, 2015.
- [2] A. Shitsukane, W. Cheriuyot, C. Otieno, and M. Mgala, "A survey on obstacles avoidance mobile robot in static unknown environment," Int. J. Comput., Vol.28, No.1, pp.160-173, 2018.
- [3] Y. Cao and N. M. Nor, "An improved dynamic window approach algorithm for dynamic obstacle avoidance in mobile robot formation," Decision Analytics Journal, Vol.11, p.100471, 2024.
- [4] E. J. Molinos, A. Llamazares, and M. Ocana, "Dynamic window based approaches for avoiding obstacles in moving," Robotics and Autonomous Systems, Vol.118, pp.112-130, 2019.
- [5] K. Shibata, N. Shibata, K. Nonaka, and K. Sekiguchi, "Model predictive obstacle avoidance control for vehicles with automatic velocity suppression using artificial potential field," IFAC-PapersOnLine, Vol.51, No.20, pp.313-318, 2018.
- [6] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," The International Journal of Robotics Research, Vol.5, No.1, pp.90-98, 1986.
- [7] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp.652-660, 2017.
- [8] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," in Advances in Neural Information Processing Systems, Vol.30, 2017.
- [9] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph CNN for learning on point clouds," ACM Trans. Graphics (TOG), Vol.38, No.5, pp.1-12, 2019.
- [10] T. Gomes, D. Matias, A. Campos, L. Cunha, and R. Roriz, "A survey on ground segmentation methods for automotive LiDAR sensors," Sensors, Vol.23, No.2, p.601, 2023.
- [11] B. Wang, J. Lan, and J. Gao, "LiDAR filtering in 3D object detection based on improved RANSAC," Remote Sensing, Vol.14, No.9, p.2110, 2022.
- [12] C. Chen, J. Guo, Y. Li, and L. Xu, "Segmentation-based hierarchical interpolation filter using both geometric and radiometric features for LiDAR point clouds over complex scenarios," Measurement, Vol.211, p.112668, 2023.
- [13] L. Chu, Y. Wang, S. Li, Z. Guo, W. Du, J. Li, and Z. Jiang, "Intelligent vehicle path planning based on optimized A* algorithm," Sensors, Vol.24, No.10, p.3149, 2024.
- [14] R. Yonetani, T. Taniai, M. Berekatain, M. Nishimura, and A. Kanezaki, "Path planning using neural A* search," in Proc. Int. Conf. Machine Learning, PMLR, pp.12029-12039, 2021.
- [15] M. Yermo, F. F. Rivera, J. C. Cabaleiro, D. L. Vilarino, and T. F. Pena, "A fast and optimal pathfinder using airborne LiDAR data," ISPRS Journal of Photogrammetry and Remote Sensing, Vol.183, pp.482-495, 2022.
- [16] K. Al-Mutib, M. AlSulaiman, M. Emaduddin, H. Ramdane, and E. Mattar, "D* lite based real-time multi-agent path planning in dynamic environments," in 2011 Third International Conference on Computational Intelligence, Modelling & Simulation, IEEE, pp.170-174, 2011.
- [17] M. Kamezaki, A. Kobayashi, R. Kono, M. Hirayama, and S. Sugano, "Dynamic waypoint navigation: model-based adaptive trajectory planner for human-symbiotic mobile robots," IEEE Access, Vol.10, pp.81546-81555, 2022.
- [18] S. J. Al-Kamil and R. Szabolcsi, "Enhancing mobile robot navigation: Optimization of trajectories through machine learning techniques for improved path planning efficiency," Mathematics, Vol.12, No.12, p.1787, 2024.
- [19] C. Ericson, Real-time Collision Detection, CRC Press, 2004.

- [20] D. Meister, S. Ogaki, C. Benthin, M. J. Doyle, M. Guthe, and J. Bittner, "A survey on bounding volume hierarchies for ray tracing," Computer Graphics Forum, Vol.40, No.2, pp.683-712, 2021.
- [21] H. A. Sulaiman and A. Bade, "Bounding volume hierarchies for collision detection," Computer Graphics, Vol.10, pp.5772-2386, 2012.

진 영 훈(Young-Hoon Jin) [중신회원]



- 2019년 8월 : 중앙대학교 (영상학박사)
- 2021년 3월 ~ 현재 : 백석대학교 첨단IT 조교수

<관심분야>

XR, RL, IoT, Network, Reverse Engineering

이 승 화(Seunghwa Lee) [정회원]



- 2005 2월 : 성균관대학교 컴퓨터공학과 (공학석사)
- 2008 8월 : 성균관대학교 컴퓨터공학과 (공학박사)
- 2008년 9월 ~ 2012년 2월 : 성균관대학교 정보통신공학부 조교수
- 2012년 2월 ~ 현재 : 백석대학교 컴퓨터공학부 교수

<관심분야>

지능형SW, 소프트웨어공학

김 혜 경(Hyei-Kyung Kim) [정회원]



- 1982년 2월 : 한양대학교 가족학과 (이학사)
- 1987년 2월 : Kent State University, Ohio, USA (가족사회학 석사, 노년학 전공)
- 1999년 3월 ~ 2024년 8월 : 백석대학교 사회복지학부 교수

- 2024년 8월 ~ 현재 : 백석대학교 교수, 지역혁신추진단 단장

<관심분야>

노인복지, 공동체