

# 제한된 적응형 해상도를 가진 격자맵 기반의 경로 계획 연구

김선휘<sup>1</sup>, 이강빈<sup>2</sup>, 김태현<sup>3</sup>, 강창근<sup>3</sup>, 전정환<sup>1,2\*</sup>  
<sup>1</sup>울산과학기술원 전기전자공학과, <sup>2</sup>울산과학기술원 인공지능대학원, <sup>3</sup>한화시스템

## A Path Planning Study Based on Limited Adaptive Resolution Grid Map

Sunhwi Kim<sup>1</sup>, Kangbeen Lee<sup>2</sup>, Tae-Hyun Kim<sup>3</sup>, Chang-Keun Kang<sup>3</sup>, Jeong hwan Jeon<sup>1,2\*</sup>

<sup>1</sup>Department of Electrical Engineering, Ulsan National Institute of Science and Technology  
<sup>2</sup>Graduate School of Artificial Intelligence, Ulsan National Institute of Science and Technology  
<sup>3</sup>Hanhwa Systems Co., Ltd.

**요약** 이 논문에서는 일정한 해상도를 가진 격자맵과 적응형 해상도를 가진 격자맵이 가지는 단점을 보완하기 위해 두 격자맵을 결합한 제한된 적응형 해상도를 가진 격자맵을 제안한다. 제안된 방식은 쿼드 트리를 이용하여 일정한 저해상도를 가진 격자맵의 각 격자를 가르는 방식으로 적응형 해상도를 추가하며 그 후, 감감수, 이차원 가우시안 함수를 이용하여, 환경을 반영한 비용을 각 격자에 할당한다. 경로 계획은 주어진 환경 하에서의 최소 비용 경로를 생성하는 문제이다. 특히, 경로 계획 알고리즘은 환경 모델링의 영향을 크게 받기 때문에 생성되는 경로의 질과 안정성을 잘 고려하기 위해서는 환경을 적절히 모델링할 필요가 있다. 제안된 방식을 통해 제한된 적응형 해상도를 가진 격자맵으로 환경을 모델링할 경우 일정한 고해상도를 가진 격자맵보다 적은 메모리로 환경을 효율적으로 표현할 수 있으며 적응형 해상도를 가지는 격자맵에 비해 위험지역 근처에서 좀 더 세밀한 환경을 모델링 할 수 있다. 또한, 제안된 환경 모델링 하에서 A\* 알고리즘으로 경로를 생성할 경우 좀 더 높은 질과 안정성을 기대할 수 있다.

**Abstract** This paper proposes a limited adaptive resolution grid map, which combines the regular grid map with the adaptive grid map to supplement the disadvantages of two kinds of grid map. The proposed method adds the adaptive resolution on the regular grid map by dividing each grid on a low-resolution regular grid map based on the quadtree, then assigns a cost based on the given environment for each grid, using the winding number and 2D Gaussian function. Path planning is the problem of generating a minimum-cost path in a given environment. In particular, modeling the environment appropriately is important for considering the quality and stability of the generated path because path planning algorithms are highly influenced by environmental modeling. According to the proposed method, when modeling an environment with a limited adaptive resolution grid map, the environment can be represented efficiently with less memory than a grid map with a fixed high resolution, and finer environments can be modeled near dangerous areas than an adaptive grid map. Furthermore, higher quality and stability can be expected when generating paths with the A\* algorithm under the proposed environmental modeling.

**Keywords** : Path Planning, Environment Modeling, Quad Tree, Winding Number, Gaussian Function

본 논문은 2021년도 정부(방위사업청)의 재원으로 국방기술진흥연구소의 지원을 받아 수행된 연구임. (No. KRIT-CT-21-009, 전장정보 기반 실시간 자동입무실행/수정기술 개발)

\*Corresponding Author : Jeong hwan Jeon(Ulsan National Institute of Science and Technology)  
email: jhjeon@unist.ac.kr

Received December 5, 2022

Revised January 3, 2023

Accepted February 3, 2023

Published February 28, 2023

### 1. 서론

경로 계획은 주어진 환경을 기반으로 하여, 최소의 비용을 가지는 경로를 만드는 문제로, 승차 공유부터 자율주행까지 다양한 분야에서 활용된다[1,2]. 경로 계획을 위한 알고리즘으로, A\*[3], D\*[4], 다익스트라[5]와 같이 그래프를 기반으로 한 알고리즘과, RRT[6], RRT\*[7]와 같은 샘플링 기반의 알고리즘이 쓰이며, 현재는 머신러닝을 접목시킨 Neural A\*[8], Neural RRT\*[9]와 같은 알고리즘도 등장한다.

위 알고리즘들의 공통점은, 주어진 지도에서 경로를 찾기 때문에 환경 모델링의 영향을 받는다는 것이다. 환경 모델링이란, 주어진 환경을 경로 계획에 고려하고자 하는 요소를 기반으로 각 영역에 비용을 부여하는 것을 말한다. 간단한 방식으로는 장애물로 인해 로봇이 갈 수 없는 공간에 높은 비용을, 갈 수 있는 공간에 낮은 비용을 적용하는 방법을 사용한다. 또한, 특정 지역을 모델링한다고 했을 때, 각 영역의 특성에 따라서 로봇이 통과하기 어려운 정도에 비례하게 비용을 적용한다. 예를 들어, Fig. 1 (b)는, Fig. 1 (a)에서 보여주는 문경의 한 지역을 도로, 산지, 논, 강 등의 특성에 따라, 10m\*10m 해상도의 격자에 비용을 적용한다.

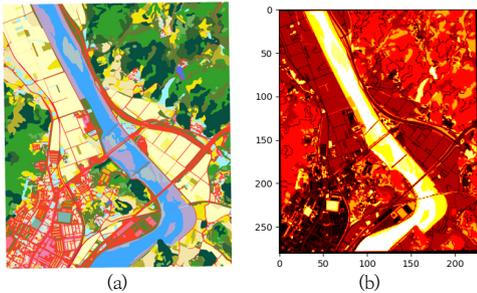


Fig. 1. Planning environment based on Mungyeong  
(a) Given Mungyeong environment (b) Mungyeong environment modelled on 10m\*10m regular grid map.

이 때, 샘플링 기반 알고리즘보다 그래프 기반 알고리즘은 환경 모델링의 영향을 더 크게 받는다[10]. 예를 들어, 격자맵으로 환경을 모델링 한다면, 격자의 형태와 정렬 상태, 그리고 해상도 등에 따라서 장애물이 다르게 표현된다. 특히 격자의 해상도 같은 경우, 장애물의 경계선 표현에 크게 영향을 준다. 예를 들어, Fig. 2는 문경의 같은 지역을 모델링한 두 격자맵으로, 경사를 고려한 주행 가능성에 기반하여 각 격자에 비용을 할당한다. Fig. 2

(a)는 10m\*10m 해상도를 이용하며 Fig. 2 (b)는 25m\*25m 해상도를 이용한다. 더 높은 해상도를 가지는 Fig. 2 (a)에서 비용이 달라지는 구간의 경계선이 더 선명하게 표현된다. 이는 장애물 근처를 지나는 경로 경유점의 정확도에 영향을 준다.

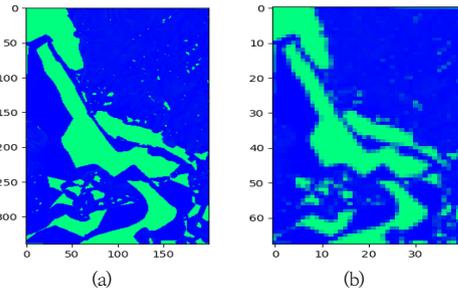


Fig. 2. Two Mungyeong traversability maps based on slope in each grid.  
(a) 10m\*10m resolution grid (b) 25m\*25m resolution grid

더해서, 장애물이 보충된 정도를 다르게 설정할 수 있다. 장애물이 보충된 정도는 환경 모델링이 아닌, 알고리즘 자체에서 반영될 수 있으나[11,12], 포텐셜 필드맵[13]과 같이, 장애물 근처에 높은 비용을 주는 방식으로 환경에 표현될 수 있다. 실제 포텐셜 필드맵에서는 장애물의 위치를 기반으로 척력의 포텐셜을 이용하는 방식이 주로 쓰이나, 단순히 장애물 주변으로 가우시안 함수를 따르는 비용을 부여하여, 장애물을 보충할 수 있다. Fig. 3는 랜덤하게 생성된 원형 장애물들을 표현한 두 격자맵으로 장애물 주변으로 가우시안 함수를 따르는 비용을 할당한다. Fig. 3 (b)가 더 큰 표준편차를 이용해, 장애물 주변의 더 넓은 영역을 보충하며, 이는 경로 계획이 더 장애물을 둘러 가는 경로를 생성하게 한다.

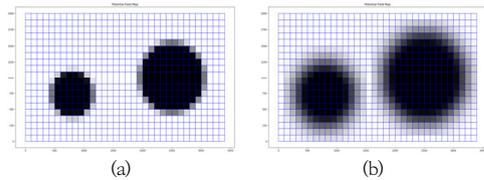


Fig. 3. Two potential field maps with 100m\*100m resolution. Costs are assigned to grid cells around obstacles, using the Gaussian function based on the distance from the boundary of the closest obstacle.  
(a) uses 0.5 and (b) uses 2 as the standard deviation.

위의 예시들은, 전쟁 또는 재해 이후 구조 상황과 같이 위험지역 근처를 위주로 목표점이 생길 경우에 경로의 비용 함수와 함께, 환경 모델링이 경로의 정확성과 안정성을 높임을 시사한다. 이 논문에서는 여러 환경 모델링 방식 중, 가우시안 함수를 이용한 포텐셜 필드맵을 기반으로 일정한 해상도를 가지는 격자맵과 적응형 해상도를 가지는 격자맵을 비교한다. 그 후, 두 방식을 적절하게 결합하여, 그래프 기반 경로 계획 알고리즘에서 정확성과 안정성을 높이는 것을 목적으로 한다.

이 논문의 구성은 다음과 같다. 2장에서 두 환경 모델링을 소개한 후, 장단점을 비교하며, 3장에서 두 방식을 결합한 환경 모델링을 소개한다. 4장에서 이 논문에서 제시한 환경 모델링과 기존의 두 방식을 비교하여 제시된 환경 모델링의 효과를 실험적으로 보이며, 5장에서 결론을 제시한다.

## 2. 환경 모델링 방식

기존 그래프 기반 경로 계획 문제에서 환경을 모델링하는 방식은 크게 두 방식이 있다. 하나는 환경을 일정한 해상도를 가진 격자맵으로 표현하는 방식이고, 다른 하나는 환경을 그래프로 표현하는 방식이다. 그래프로 환경을 표현하는 방식에는 Voronoi diagram[14]과, 적응형 해상도를 가진 격자맵 등이 존재한다.

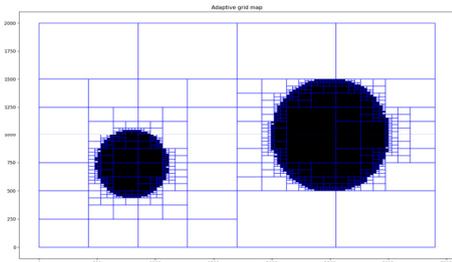


Fig. 4. An adaptive grid map based on the quad tree

Fig. 4는 적응형 해상도를 가진 격자맵의 예시를 보여준다. 대표적인 특징으로, 장애물이 있는 부분은 높은 해상도를 이용하여 더 자세하게 표현되고, 그렇지 않은 부분은 낮은 해상도로 표현된다. 이차원 환경 모델링에서, 적응형 해상도를 표현하기 위해 주로 쿼드 트리 방식을 이용한다. 쿼드 트리는 공간을 사분할하며 만들어지는 트리를 말하며, 두 가지 방식으로 만들어질 수 있다. 첫 번째 방식은 최고 해상도까지 환경을 표현한 뒤, 특정 기

준을 통해서 환경을 합하며 만드는 방식이고[15], 두 번째 방식은, 장애물이 있는 공간을 지정된 최고 깊이까지 가리며 만드는 방식이다[16]. 가르는 기준은 장애물의 경계선 좌표가 될 수 있고, 장애물과의 교집합 유무가 될 수 있다. 이 논문에서는 지도 생성 시 사용되는 메모리를 줄이기 위해서 두 번째 방식을 사용하였다. 경로 계획은 쿼드 트리 구조에서 마지막 잎 노드에 해당하는 격자들을 이웃끼리 연결한 그래프 위에서 이루어진다.

소개된 두 격자맵은 각각 장단점이 존재한다. 일정한 해상도를 가진 격자맵은 만들고 수정하는 것이 쉬우며, 한 격자가 항상 같은 수의 행동만을 취하기 때문에, 경로 계획 단계에서의 계산도 간단하다. 하지만 장애물의 표현이 해상도의 영향을 크게 받는다. 해상도가 낮을 때는 장애물이 부정확하게 표현되고, 높을 때는 장애물을 세밀하게 표현되지만 많은 메모리를 요구한다. 적응형 해상도를 가진 격자맵은 적은 메모리로 장애물 근처에서 높은 해상도를 가질 수 있는 장점이 있으나, 만들어지는 경로의 질이 떨어진다는 단점이 있다[10].

예를 들어, 적응형 해상도를 가진 격자맵을 기반으로 경로가 계획되었을 때, 경로가 들쭉날쭉하게 (jagged movement) 만들어졌으며, 1m\*1m 해상도를 가지는 격자맵에서 구한 최적 경로와 비교할 때, 평균적으로 20%가 긴 경로가 생성된다. 추가적인 후처리 과정을 거친 후에도 10%가 긴 경로가 생성된다[17].

이 논문에서는 쿼드 트리를 사용했을 시의 단점을 보완하기 위해서 적응형 해상도를 가진 격자맵의 기본 해상도를 제한하는, 제한된 적응형 해상도를 가진 격자맵을 제안한다.

제한된 격자맵을 통해, 기본 해상도를 가진 격자맵과 적응형 해상도를 가진 격자맵에 비해서는 높지만, 높은 해상도의 격자맵에 비해서는 낮은 메모리를 사용할 것이다. 또한, 장애물 근처에서 자세히 환경을 표현함과 동시에 경로의 질을 높이는 것을 이루고자 한다.

## 3. 방법론

제한된 적응형 해상도를 가진 격자맵은 Fig. 5를 따라 만들어진다. 우선 전체 환경을 의미하는 트리의 루트 노드 아래에 기본 해상도를 가진 격자를 자식 노드로 두며, 전체 맵이 초기화된다(Initialize a map). 각 장애물의 외곽 좌표는 보간되고(Interpolate coordinates), 각 자식 노드들은 보간된 외곽 좌표를 기준으로 최대 깊이까

지 갈라지며 트리를 확장한다(Partition the map). 동시에, 외곽 좌표가 들어있는 노드는 높은 비용을 부여한다. 이 과정에서 장애물이 높은 해상도로 표현되어, 장애물 근처 경유점의 위치에 정확도가 높아진다. 후에, 감김수를 사용하여 각 노드들이 장애물의 안쪽에 위치하는지 판단 후, 안쪽에 위치한다면 해당 노드에 높은 비용을 할당한다(Assign cost). 후에 장애물의 외곽 좌표에 해당하는 노드들의 근처 노드에 거리에 따른 가우시안 함수를 사용하여 비용을 부여한다(Spread cost). 이 과정에서 장애물이 보충되어 경로의 안정성이 높아진다.

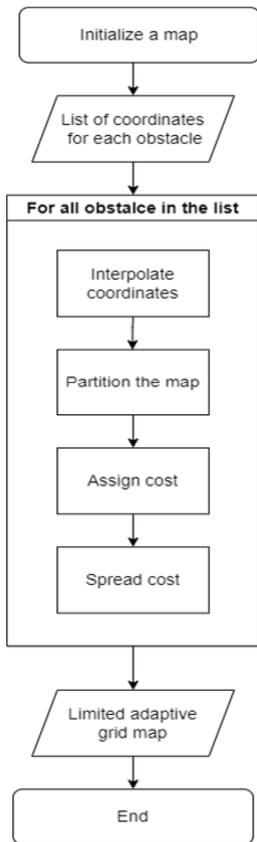


Fig. 5. The algorithm flow to make a limited adaptive grid map

### 3.1 장애물의 외곽 좌표

적응형 해상도의 장점을 최대한 살리기 위해서는 장애물의 변이 가장 높은 해상도로 표현되어야 한다. 이를 위해, 장애물의 외곽 좌표 사이의 간격이 최대 깊이를 가진 노드로 구성되도록 충분히 작아야 한다. 해당 조건에서, 장애물의 두 외곽 좌표 사이의 간격이 가장 클 수 있는

경우는 장애물의 외곽 좌표가 기본 해상도의 짧은 쪽 변을 따라 존재할 때도 장애물의 변을 따라 최대 깊이를 가진 노드를 가지게 되는 경우이다. 이 경우를 만족하기 위해서는, Eq. (1)을 만족해야 한다.

$$l \leq \min(s_{n-1,1}, s_{n-1,2}) \quad (1)$$

$l$ 은 최대 깊이를 의미하며,  $s_{a,b}$ 는  $a$ 번째 깊이에 해당하는  $b$ 번째 변의 길이에 해당한다.  $l$ 은 두 외곽 좌표의 거리에 해당한다. 이 논문에서 기본 해상도는 100m\*100m로 설정되고,  $n$ 은 3으로 설정된다. 이것은 두 외곽 좌표의 거리가 25m보다 크면, 장애물의 변에 깊이가 3이 아닌 노드가 생기게 된다는 것을 의미한다. 이 논문은  $l$ 은 경계에서 버퍼를 두기 위해 20m로 설정하고, 장애물의 외곽 좌표를 보간한다.

### 3.2 감김수

감김수[18]는 다각형의 변이 특정 방향으로 돌아간다고 가정하고, 점에서 반직선을 그어 점이 다각형의 안쪽에 존재하는지를 판단할 수 있는 수를 말한다. 이 논문에서는 다각형의 변을 시계방향으로 회전시켜 반직선을 긋는다. 아래로 내려가는 변과 만날 시 감김수를 1 증가시키고, 반대일 경우 1 감소시킨다. 최종 감김수가 0이면 장애물의 바깥에, 그 외는 장애물의 안쪽에 위치하는 점으로 판단한다.

각 노드에서 감김수를 계산하기 전, 연산을 줄이기 위해 해당 장애물의 최대와 최소의  $x$ 와  $y$ 좌표를 이용하여, 범위를 좁힌다. 전체 환경이 이미 장애물의 외곽 좌표를 기준으로 나누어져 있는 상태이기 때문에, 각 노드의 중심에서만 감김수를 판단한다. 이 논문에서는 장애물이 스스로 교점을 만들지(self-intersecting) 않는다고 가정하기 때문에, 감김수를 직접 구하지 않고, 처음에 노드가 장애물 바깥에 존재한다고 판단한 후, 반직선이 변과 만났을 때마다 판단을 뒤집는 방식을 선택한다.

### 3.3 이차원 가우시안 함수

모든 옆 노드에 높은 비용을 할당한 후, 장애물의 외곽 좌표를 가지고 있는 근처 노드들을 중심으로 이차원 가우시안 함수를 이용하여 비용을 퍼뜨린다. 이 때, 근처 노드들의 비용은 Eq. (2)가 된다.

$$\text{cost}(cell') = \text{cost}(cell) * \exp\left[-\frac{\{dist(cell, cell')/r\}^2}{2\sigma^2}\right] \quad (2)$$

$cost()$ 는 해당 격자에 해당하는 비용을 의미하며,  $cost(cell)$ 은 장애물에 해당하는 높은 비용이 된다.  $dist()$ 는 두 격자 중심 사이의 유클리디안 거리를 의미한다.  $r$ 은 파라미터로 너무 작은 값이 지수 함수에 적용되어 값의 대부분이 0에 가까워지는 것을 방지한다. 이 논문에서는  $r$ 을 기본 해상도의 한 변의 길이로 설정한다.  $\sigma$ 는 가우시안 함수의 표준편차에 해당하는 파라미터로, 높을수록 비용이 넓게 퍼지게 된다.

이차원 가우시안 함수를 사용하기 전에, 비용을 퍼뜨릴 노드를 거리 기준으로 제한한다. 1 look을 한 기본 해상도 격자를 보는 것으로 정의하며, 높은 look값은 비용을 더 넓게 퍼뜨린다. 이 논문의 경우 기본 해상도가  $100m*100m$ 이기 때문에, 1 look은 중심 노드를 자식 노드로 가지는 기본 해상도 격자 주변의  $300m*300m$ 의 범위를 본다는 것을 의미한다. 이때,  $\sigma$ 가 높아져도 look이 1이라면, 비용은 더 넓게 퍼지지 않고  $300m*300m$  범위에 해당하는 격자들이 조금 더 높은 비용을 가지게 된다. look을 1로 두더라도,  $\sigma$ 를 1로 설정할 시, 비용이 장애물 주변으로 충분히 퍼지기에, 이 논문에서는 look을 1로 제한한다.

## 4. 실험 결과

이 장에서는 일정한 해상도를 가진 격자맵, 적응형 해상도를 가진 격자맵 그리고, 제안된 격자맵을 사용하여 당시의 환경 모델링 변화에 따른 경로의 질을 비교한다. 더해서, 경로 생성 시간과 격자맵이 요구하는 메모리를 비교한다. 마지막으로 제안된 격자맵과 적응형 해상도를 가진 격자맵에서 파라미터를 변화시켰을 때 환경변화를 비교하며, 제안된 격자맵이 경로 생성에서 갖는 이점을 보여준다. 실험에 CPU는 Intel(R) Core(TM) i9-9900K CPU @ 3.60GH이 사용되며, RAM은 16GB를 가지는 Samsung DDR4가 사용된다.

### 4.1 경로 계획 문제 설정

경로 계획 시, 한 격자에서 바로 이웃한, 즉 격자의 변 또는 점이 접하는, 격자의 중심으로 이동하는 행동만을 고려한다. 경로 계획 방식은 대표적인 그래프 경로 계획 방식 중 하나인 A\*를 사용하며 이때, 비용은 Eq. (3)[17]을 사용한다.

$$\begin{aligned} cost(c1, c2) = \\ dist(c1, c2) * danger(c2) \end{aligned} \quad (3)$$

$cost(a,b)$ 는 격자 a에서 격자 b까지 가는 비용을 나타낸다.  $dist()$ 는 두 격자 중심 사이의 유클리디안 거리를 나타내며,  $danger()$ 는 해당 격자가 가지는 위험도를 의미한다.  $danger()$ 가 0 인 경우,  $cost(a,b)$ 는  $dist(a,b)$ 와 같은 값이 할당된다. A\*를 사용하기 위한 휴리스틱은 현재 격자 중심과 목표 격자 중심 사이의 유클리디안 거리를 사용하고자 하며, 이 휴리스틱 함수는 A\*가 최적 경로를 찾기 위한, admissibility와 consistency를 만족한다.

### 4.2 격자맵 생성

제안된 격자맵은 3장에서 소개된 방식으로 생성된다. 그 후, 트리의 각 잎 노드에 대해서 이웃들을 구하고, 경로 계획에 사용될 그래프를 생성한다.

#### 4.2.1 일정한 해상도를 가진 격자맵

제안된 격자맵과 일정한 해상도를 가진 격자맵 사이의 환경 모델링을 비교하기 위해서, 일정한 해상도를 가진 격자맵은 제안된 격자맵의 트리에서 기본 해상도에 해당하는 격자를 가져온다. 이때, 기본 해상도를 가진 격자의 비용은 그 노드의 모든 자식 노드들의 평균으로 계산된다. 일정한 해상도를 가진 격자맵의 경우, 이웃 노드를 구하는 규칙이 뚜렷하기 때문에 경로 계획을 위한 그래프를 따로 생성하지 않는다.

#### 4.2.2 적응형 해상도를 가진 격자맵

적응형 해상도를 가진 격자맵은 2장에서 소개된 것처럼, 전체 환경에서 시작하여, 장애물의 외곽 좌표를 기준으로 퀴드 트리를 생성한다. 비용을 할당하는 방식은 3장과 같은 방식을 사용한다. 하지만, 일정한 기본 해상도가 없기에, Spread cost 시 이웃한 격자만을 고려하였고,  $\sigma$ 도 너무 큰 격자가 높은 값을 갖는 것을 방지하기 위해, 값을 낮추었다. 트리의 최대 깊이(d)는 장애물 근처에서 환경 모델링을 제안된 격자맵과 유사하게 만들기 위해서 Eq. (4)를 만족하는 d값을 사용한다.

$$\min |res - \max(w, h)/2^d| \quad (4)$$

res는 제안된 격자맵의 최고 해상도에 해당하는 값이며, w와 h는 각각 전체 환경의 넓이와 높이를 의미한다. 이 논문은 res는 12.5m, w는 3400m 그리고 h는 2000m이기 때문에, d는 8로 설정된다. 이 기준에서 최대 해상도는, 제안된 격자맵에서  $12.5m*12.5m$ , 적응형 해상도를 가진 격자맵에서  $13.28m*7.81m$ 로, 제안된 격자맵이 1.507배 큰 영역으로 더 낮은 최대 해상도를 표

현한다. 이 부분은 4.3.4에서 고려한다.

### 4.3 결과 비교

Fig. 6, 7과 8을 통해 각 격자맵에서 두 가지 경우의 경로 계획의 결과를 보여주고, Table 1에서 각 경우에서 나온 경로의 길이를 보여준다. 첫 번째 경우는 (0m,0m)에서 (2495m,1495m)까지 경로를 생성하고, 두 번째 경우는 (0m,0m)에서 (3275m,1495m)까지 경로를 생성한다. 일정한 해상도를 가진 격자맵에서 A\*를 실행할 시, 정해진 해상도 상에서 최적 경로를 획득하기 때문에, 일정한 해상도를 가진 격자맵을 기준으로, 나머지 두 격자맵에서 나온 경로를 비교할 것이다. Fig. 7 (a) 같은 경우 경로의 끝부분이 경로의 길이를 기준으로 한다면 최적이지 아니지만, 한 격자가 가지는 위험도를 함께 고려한 비용에서 최적이 된다. 그러므로 Fig. 7에서 두 경우는 최적 경로를 얻는다.

Table 1. Path distance on different grid maps for each case in meter

map	case 1	case 2
Adaptive grid map	3877.6	3735.1
Regular grid map	3421.3	3779.9
Limited adaptive grid map	3421.3	3809.0

#### 4.3.1 환경 모델링과 생성된 경로

적응형 해상도를 가진 격자맵에서는 두 경우에서 성능 차이가 크다. 첫 번째 경우에 대해서는 최적 경로와 비교했을 때, 들쭉날쭉한 경로를 만들지만, 두 번째 경우에는 Fig. 7 (b)에서 구불구불한 경로가 직선 경로로 대체되면서 더 짧은 경로를 만든다. 적응형 해상도를 가진 격자맵에서 항상 최적 경로와 비슷한 경로를 만들기 위해서는, 후처리 과정이 필요하다.

이런 불완전한 성능은 공간마다 크게 변화하는 해상도에서 기인한다. 일정한 해상도를 가진 격자맵에서는 장애물과 같은 거리에 있는 격자는 같은 위험도를 갖는다고 가정되나, 적응형 해상도를 가진 격자맵에서는 그렇지 않다. Fig. 5 (a)에서 경로의 뒤에서부터 네 번째 경우 점, (2125, 1250), 이 존재하는 큰 격자는 위험도가 0에 가깝다. 하지만 해당 격자와 장애물에서 같은 거리에 존재하는 영역을 포함한 격자, (2231.25, 1562.5), 같은 경우 위험도가 1.932의 값을 갖는다. 이는 제한되지 않고 공간마다 크게 변하는 해상도가 공간마다 일정한 환

경 모델링을 방해하는 것을 보여준다. 이런 환경 모델링 아래에서는, Fig. 5 (b)처럼 대체할 수 있는 격자가 주변에 없는 경우에는 최적 경로를 만들지만, Fig. 5 (a)같은 경우, 뒤에서 세 번째 경우점을 가진 격자가 충분히 작은 비용을 가지면서, 돌아가는 경로를 만든다.

하지만 제안된 격자맵은 환경을 기본 해상도로 제한하기 때문에, 장애물과 같은 거리에 있는 격자가 서로 같은 위험도를 갖는다고 가정된다. 그로 인해, Fig. 8 (a)와 같이, 목표점이 기본 해상도를 가지는 격자에 존재하고, 경로가 장애물 근처를 지나는 것이 필연적이지 않은 경우, 일정한 해상도를 가지는 격자맵에서 만든 최적 경로를 동일하게 만든다. 하지만 두 번째 경우처럼, 목표점이 기본 해상도를 가진 격자에 존재하지 않는 경우, 더 긴 경로를 만든는데, Fig. 8 (b)가 보이듯, 장애물 근처에서 더 자세한 경로를 생성하여, 목표점에 더 가까운 경우점들이 경로에 포함된 것에서 기인한다. 이는 제안된 격자맵이 장애물 근처에서 일정한 해상도를 가진 격자맵 보다 자세한 환경 모델링이 가능하다는 것을 의미한다.

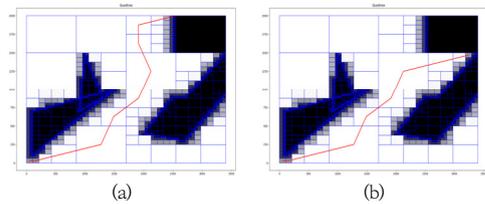


Fig. 6. Two paths on the adaptive grid map  
(a) Path from [0,0] to [2495,1995] (b) path from [0,0] to [3275,1495]

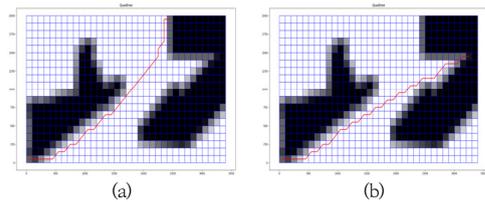


Fig. 7. Two paths on the regular grid map.  
(a) Path from [0,0] to [2495,1995] (b) Path from [0,0] to [3275,1495]

#### 4.3.2 메모리 사용량

각 격자는 모두 float 64 타입의 값을 가지기 때문에, 각 격자맵이 가지는 격자 수는 메모리 사용량을 근사시킨다. Fig. 9가 보이듯, 예상과 같이 일정한 기본 해상도를 가진 격자맵이 가장 적은 수의 격자를 가진다. 하지만, 해당 격자맵은 가장 낮은 해상도를 가진다. 제안된

격자맵의 가장 높은 해상도를 전체적으로 가진다면, 격자의 수는 64배로 증가한다. 이 경우에 일정한 해상도를 가진 격자맵은 가장 높은 격자 수를 가진다.

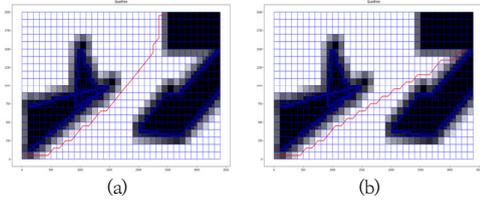


Fig. 8. Two paths on the limited adaptive grid map. (a) Path from [0,0] to [2495,1995] (b) Path from [0,0] to [3275, 1495]

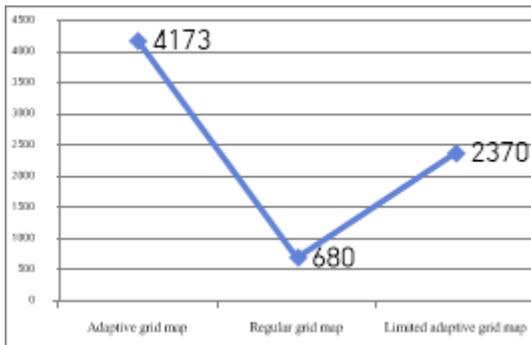


Fig. 9. The number of grids for each grid map

적응형 해상도를 가진 격자맵은 일정한 높은 해상도를 가진 격자맵보다 10분의 1에 가까운 격자 수를 가지며, 이는 적응형 해상도의 장점을 다시 보여준다. 하지만, 비슷한 최대 해상도를 표현하는 제안된 격자맵과 비교했을 때, 더 많은 격자 수를 갖는다. 즉, 메모리 사용량 측면에서 제안된 격자맵이 이점을 갖는다.

하지만 이는 장애물의 배치에 영향을 받을 수 있다. 특히 기본 해상도의 격자와 변을 공유하는 장애물에 대해, 제안된 격자맵은 해상도를 높이지 않기 때문에, 더 작은 격자 수를 갖는다. 이 부분은 4.3.4에서 고려한다.

### 4.3.3 경로 생성 시간

일정한 해상도를 가진 격자맵은 두 경우에 대해 가장 낮은 경로 생성 시간을 보인다. 알고리즘이 탐색하는 노드의 수는 각각 전체 노드 수의 77.5%, 94.6%로 대부분의 노드를 탐색하지만, 전체 노드의 수 자체가 작기에, 계산상의 이점을 갖는다.

적응형 해상도를 가진 격자맵은 두 경우에서 성능의 큰 차이를 보인다. 해당 격자맵은 특정 경우에 일정한 해

상도를 가진 격자맵에서의 경로 계획 보다 더 짧은 경로를 얻으나, 그 외에는 돌아가는 경로를 얻는 단점을 갖는다.

첫 번째 경우는 경로 계획 시, 더 짧은 경로가 높은 비용을 가지면서 돌아가는 길을 탐색한다. 이 때, 많은 수의 노드를 탐색하고 탐색된 노드 수는 전체 노드의 50.9%가 된다. 일정한 해상도를 가진 격자맵에 비하면 비율은 작으나, 수 자체가 많기에 더 많은 시간이 걸린다. 반대로, 두 번째 경우에는 돌아가는 경로를 탐색하지 않기에, 훨씬 작은 수의 노드를 탐색한다. 그 비율을 전체 노드의 12.1%로, 수 자체가 작기에, 일정한 해상도를 가진 격자맵보다 시간이 짧게 걸린다. 적응형 해상도를 가진 격자맵에서 경로 계획을 할 시, 장애물이 없는 평탄한 지역은 한 번에 넘어가기 때문에, 경로의 끝이 장애물 근처에서 많은 경로점을 가져도 탐색 노드 수 자체는 작다. 하지만 후처리가 필요한 돌아가는 경로에서 더 많은 경로 생성 시간이 요구되기에 개선이 필요하다.

제안된 격자맵에서 경로 생성 시간은 두 경우에서 탐색된 노드 수가 일정한 해상도를 가진 격자맵보다 증가한다. 이는 자세한 환경 모델링을 위해 장애물 근처에서 증가한 노드 수와 관련된다. 그러므로, 장애물 근처 탐색이 더 많이 필요할수록, 경로 생성 시간이 증가한다.

첫 번째 경우에 기본 해상도를 가진 격자맵에서 대부분 탐색이 이뤄지고 장애물을 추가로 고려한다. 전체 노드 수는 장애물 근처에서 대부분 증가하기에, 탐색된 노드 수가 전체 노드 수의 증가보다 훨씬 적게 증가하며, 전체 노드 수의 28.9%만을 차지한다. 두 격자맵에서 비슷한 형태의 탐색이 이뤄지며, 결과로 나온 경로의 길이도 같으며, 시간의 증가량도 크지 않다. 하지만, 두 번째 경우는 탐색된 노드가 96.5%로 일정한 해상도를 가진 격자맵에 비해 1.9% 증가한다. 탐색 노드 수의 증가로, 경로 생성 시간도 크게 증가한다. 이는 경로 끝에서 장애물 근처 탐색과 관련된다.

적응형 해상도를 가진 격자맵도 경로 생성 시간에서는 제안된 격자맵보다는 이점을 갖는다. 이는 A\* 알고리즘과 관련된다. A\* 알고리즘은 휴리스틱을 통해 최적 경로가 아니라고 판단되는 경우점은 고려하지 않는다. 이 때, 적응형 해상도를 가진 격자맵은 장애물이 존재하지 않는 영역에서, 제안된 격자맵보다 더 큰 영역을 거를 수 있고, 알고리즘이 진행되면서 고려되는 영역을 크게 좁힐 수 있다. 이로 인해, 탐색되는 노드 수를 줄여 경로 생성 시간은 줄어들 수 있으나, 공간상의 균일하지 않은 환경 모델링에서 큰 영역을 버리는 것은 특정 경우에서 크게 돌아가는 경로를 생성할 수 있다.

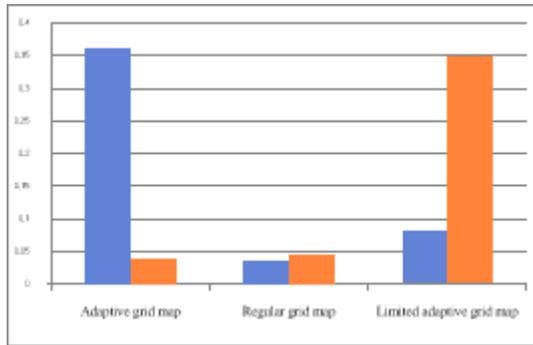


Fig. 10. Planning time for each case on each grid map in second. Blue bars and orange bars represent case 1 and case 2, respectively.

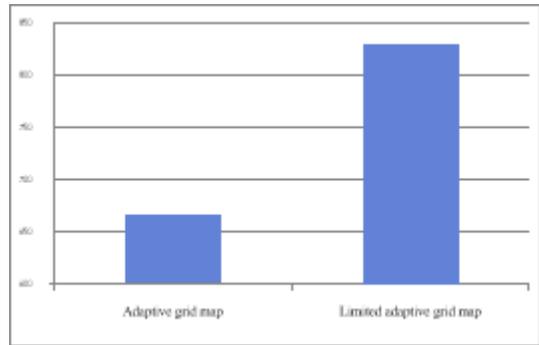


Fig. 12. Mean searched cell for each case on adaptive grid map and limited adaptive grid map.

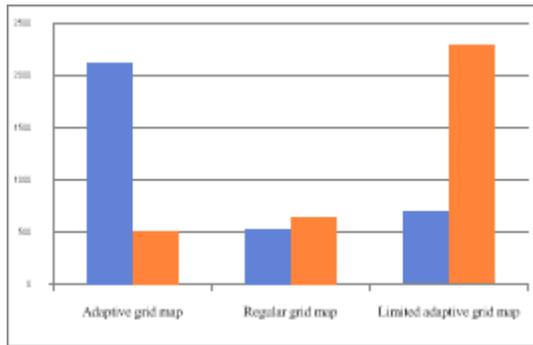


Fig. 11. The number of cells searched by A\* for each case on each grid map in second. Blue bars and orange bars represent case 1 and case 2, respectively.

Table 2와 Fig. 12는 10개의 다른 목표지점을 가진 경로 생성에 관한 결과이며, 앞의 두 경우와 비슷한 경향성을 보인다. 제한된 격자맵에서 탐색되는 노드가 더 많고, 경로 생성 시간은 증가하지만, 일정한 해상도를 가진 격자맵에서의 경로를 기준으로 적응형 해상도를 가진 격자맵과 비교했을 때, 경로 생성의 질이 더 높고 안정적이다.

Table 2. Mean planning time on each grid map for 10 cases in second.

map	Mean
Adaptive grid map	0.084
Regular grid map	0.024
Limited adaptive grid map	0.101

Table 3. Mean and standard deviation of distance on an adaptive grid map and a limited adaptive grid map for 10 cases in meter. Each distance is subtracted by the distance obtained from a regular grid map for the corresponding case.

map	Mean	Std
Adaptive grid map	180.337	233.814
Limited adaptive grid map	-3.377	14.304

#### 4.3.4 비교 공정성

현재 비교조건은 제안된 격자맵에 유리한 부분이 있다. 공정한 비교를 위해서 기본 해상도 격자와 변을 공유하지 않는 장애물들만 사용하고 적응형 해상도의 최대 깊이를 7로 줄인다. 해당 최대 깊이에서 제안된 해상도에 비해 적응형 해상도가 2.65배 넓은 영역으로 더 낮은 해상도를 표현한다. 장애물은 Fig. 13와 같이 설정한다.

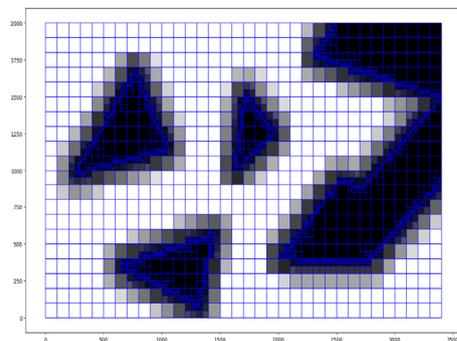


Fig. 13. Limited adaptive grid map with the new obstacle setting.

이전 절에 소개된 알고리즘상 특징으로 인해, Table 4가 보이듯, 제안된 격자맵이 가장 큰 경로 생성 시간을 갖는다. 하지만 Table 5가 보이듯, 제안된 격자맵은 적응형 해상도를 가진 격자맵에 비해, 경로의 질과 안정성에서 큰 이점을 가진다. 더해서, Fig. 14는 적응형 해상도를 가진 격자맵이 더 낮은 최대 해상도를 표현함에도, 더 적은 메모리를 요구함을 보인다. 이는 다양한 상황에서도 제안된 격자맵이 적응형 해상도를 가진 격자맵에 비해 이점을 가짐을 의미한다.

Table 4. Mean planning time on each grid map for 10 cases in second with the new setting

map	Mean
Adaptive grid map	0.098
Regular grid map	0.024
Limited adaptive grid map	0.154

Table 5. Mean and standard deviation of distance on an adaptive grid map and a limited adaptive grid map for 10 cases in meter with the new setting. Each distance is subtracted by the distance obtained from a regular grid map for the corresponding case.

map	Mean	Std
Adaptive grid map	614.790	1095.907
Limited adaptive grid map	4.735	21.708

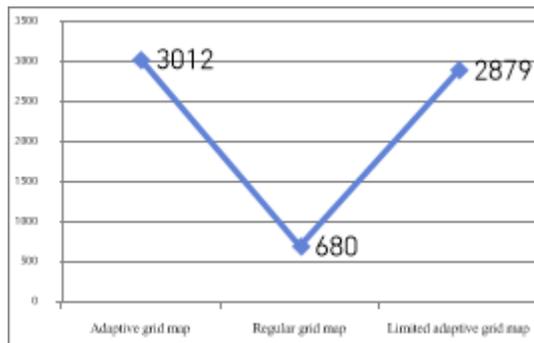


Fig. 14. The number of grids for each grid map with new condition.

#### 4.3.5 파라미터 변화

더 자세한 환경 모델링이 요구되는 상황에서, 일정한 높은 해상도를 가진 격자맵은 상당한 메모리를 사용한

다. 해당 상황에서 격자맵을 적응형 해상도와 제안된 방식으로 표현할 수 있다. 이 절에서는 목표로 하는 최대 해상도를 6.25m\*6.25m와 1.25m\*1.25m로 두고 두 방식을 비교할 것이다.

적응형 해상도를 가진 격자맵에서 최대 해상도를 높이는 방법은 최대 깊이를 높이는 것이다. 각 경우에서 최대 깊이는 Eq. (4)기반으로 구하며, 메모리를 줄이기 위해 결과에 1을 더했다. 하지만 최대 깊이를 높이는 것은, Fig. 15에서 보이듯, 기존 적응형 해상도의 단점을 그대로 가지며 메모리를 높인다.

제안된 방식은 두 가지 방식으로 최대 해상도를 높일 수 있는데, 기본 해상도를 높이는 방식과 최대 깊이를 높이는 방식이 있다. 각 방식은 장단점을 가진다. 기본 해상도를 높이는 방식은 Fig. 16에서 보이듯, 장애물 주변뿐 아니라 전체 환경을 더 자세하게 표현하지만, 메모리의 여유가 필요하다. 반면에 최대 깊이를 높이는 방식은 Table 6에서 보이듯, 메모리의 사용이 가장 낮다. 또한, Fig. 17에서 보이듯, 제한된 해상도에 의해 균일한 환경 모델링은 여전히 가능하지만, 장애물 근처 지역과 다른 부분 사이의 해상도가 커지게 된다. 즉, 제안된 방식은 사용자의 상황에 가장 맞는 환경 모델링을 파라미터 조정으로 얻을 수 있다. 이는 제안된 방식이 환경 모델링에서 가지는 추가적인 이점이다.

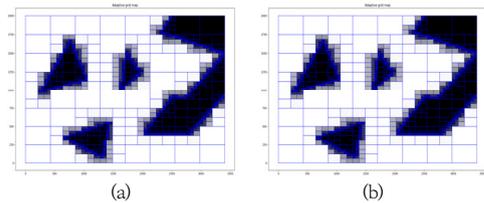


Fig. 15. Two adaptive grid maps with different maximum depth. (a) The maximum depth is set to 8. (b) The maximum depth is set to 10.

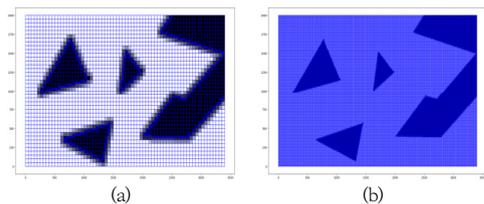


Fig. 16. Two limited adaptive grid maps with different base resolution and same maximum depth, 3. (a) The base resolution is set to 50m\*50m. (b) The base resolution is set to 10m\*10m

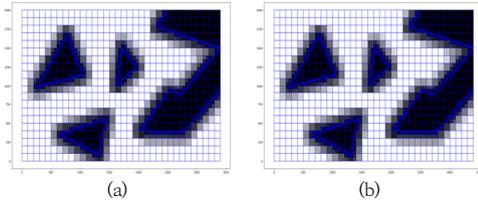


Fig. 17. Two limited adaptive grid maps with different maximum depth and same base resolution 100m\*100m..

(a) The maximum depth is set to 4. (b) The maximum depth is set to 5.

Table 6. Number of grids for each target highest resolution on adaptive grid map and limited adaptive grid map.

map	6.25m	1.25m
Adaptive grid map	4573	7720
Limited adaptive grid map1	5787	71616
Limited adaptive grid map2	4124	5372

## 5. 결론

이 논문에서는 제한된 적응형 해상도를 가진 격자맵을 제안하며, 해당 격자맵에서의 환경 모델링을 적응형 해상도를 가진 격자맵과 일정한 해상도를 가진 격자맵과 비교한다. 제안된 격자맵은 전체적으로 일정한 높은 해상도를 가진 격자맵과 비교했을 때, 더 적은 메모리를 요구한다. 또한, 장애물 근처에서 더 세밀한 환경 모델링을 얻는다. 적응형 해상도를 가진 격자맵과 비교했을 때는, 공간상에서 고른 환경 모델링을 일정한 해상도를 가진 격자맵과 유사한 정도로 만든다.

이 논문에서 제안된 방식의 이점은 다음과 같다. 제안된 방식을 통한 환경 모델링은 공간상에서 고르며, 장애물 근처에서 더 세밀한 환경 모델링을 얻는다. 두 번째로, 경로 생성 시간이 증가하지만, 경로의 질의 향상과 안정성을 이끈다. 마지막으로, 파라미터 조정을 통해 사용자의 상황에 맞춘 환경 모델링을 만든다.

## References

[1] J. A. Escalona, B. Manalo, W. J. R. Limjoco and C. C. D. Electrical, "A Ride Sharing System based on An

Expansive Search-Based Algorithm", *2020 IEEE REGION 10 CONFERENCE*, pp. 870-874, 2020.

DOI: <https://doi.org/10.1109/TENCON50793.2020.9293943>

[2] Y. Kuwata, J. Teo, G. Fiore, S. Karaman, E. Frazzoli and J. P. How, "Real-Time Motion Planning With Applications to Autonomous Urban Driving", *IEEE Transactions on Control Systems Technology*, vol. 17, no. 5, pp. 1105-1118, Sept. 2009.

DOI: <https://doi.org/10.1109/TCST.2008.2012116>

[3] P. E. Hart, N. J. Nilsson and B. Raphael, "A Formal Basis for the Heuristic Determination of Minimum Cost Paths", *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100-107, Jul. 1968.

DOI: <https://doi.org/10.1109/TSSC.1968.300136>

[4] A. Stentz, "Optimal and Efficient Path Planning for Partially-Known Environments", *1994 IEEE International Conference on Robotics and Automation*, San Diego, CA, USA, pp. 3310-3317, May 1994.

DOI: <https://doi.org/10.1109/ROBOT.1994.351061>

[5] E. W. Dijkstra, "A note on two problems in connexion with graphs", *Num. Math.*, vol. 1, pp. 269-271, 1959.

DOI: <https://doi.org/10.1007/BF01386390>

[6] S. M. LaValle, "Rapidly-exploring random trees: A new tool for path planning", TR 98-11, Computer Science Dept., Iowa State University, 1998.

[7] S. Karaman, M. R. Walter, A. Perez, E. Frazzoli and S. Teller, "Anytime motion planning using the RRT\*", *2011 IEEE International Conference on Robotics and Automation*, Shanghai, China, pp. 1478-1483, 2011.

DOI: <https://doi.org/10.1109/ICRA.2011.5980479>

[8] R. Yonetani, T. Taniyai, M. Barekatin, M. Nishimura and A. Kanazaki, "Path planning using neural a\* search", *International conference on machine learning*, pp. 12029-12039, 2021.

[9] J. Wang, W. Chi, C. Li, C. Wang, M. Q.-H. Meng, "Neural RRT\*: Learning-based optimal path planning", *IEEE Transactions on Automation Science and Engineering*, vol. 17, issue 4, pp. 1748-1758, 2020.

DOI: <https://doi.org/10.1109/TASE.2020.2976560>

[10] O. Souissi, R. Benatitallah, D. Duvivier, A. Artiba, N. Belanger and P. Feyzeau, "Path planning: A 2013 survey", *2013 International Conference on Industrial Engineering and Systems Management (IESM)*, Agdal, Morocco, pp. 1-8, 2013.

[11] J. Park and K. Iagnemma, "Sampling-based planning for maximum margin input space obstacle avoidance", *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Hamburg, Germany, pp. 2064-2071, 2015.

DOI: <https://doi.org/10.1109/IROS.2015.7353651>

[12] A. R. Willms and S. X. Yang, "Real-time robot path planning via a distance-propagating dynamic system with obstacle clearance", *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 38, no. 3, pp. 884-893, June 2008.

DOI: <https://doi.org/10.1109/TSMCB.2008.921002>

- [13] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots", *Proceedings. 1985 IEEE International Conference on Robotics and Automation*, St. Louis, MO, USA, pp. 500-505, 1985. DOI: <https://doi.org/10.1109/ROBOT.1985.1087247>
- [14] P. Bhattacharya and M. L. Gavrilova, "Voronoi diagram in optimal path planning", *4th International Symposium on Voronoi Diagrams in Science and Engineering (ISVD 2007)*, Glamorgan, UK, pp. 38-47, 2007. DOI: <https://doi.org/10.1109/ISVD.2007.43>
- [15] L. Jin, J. Rückin, S. H. Kiss, T. Vidal-Calleja and M. Popović, "Adaptive-resolution field mapping using Gaussian process fusion with integral kernels", *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 7471-7478, July 2022. DOI: <https://doi.org/10.1109/LRA.2022.3183797>
- [16] Y. Chen, W. Shuai and X. Chen, "A probabilistic, variable-resolution and effective quadtree representation for mapping of large environments", *2015 International Conference on Advanced Robotics (ICAR)*, Istanbul, Turkey, pp. 605-610, 2015. DOI: <https://doi.org/10.1109/ICAR.2015.7251518>
- [17] F. M. Garcia, M. Kapadia and N. I. Badler, "GPU-based dynamic search on adaptive resolution grids", *2014 IEEE International Conference on Robotics and Automation*. Hong Kong, China, pp. 1631-1638, 2014. DOI: <https://doi.org/10.1109/ICRA.2014.6907070>
- [18] D. Alciatore and R. Miranda, "A winding number and point-in-polygon algorithm", Report, Dep. of Mech. Engineering, Colorado State University, 1995.

김 선 휘(Sunhwi Kim)

[준회원]



- 2022년 2월 : 울산과학기술원 전기전자컴퓨터공학부 졸업
- 2022년 3월 ~ 현재 : 울산과학기술원 전기전자공학과 대학원 석박 통합과정

<관심분야>

알고리즘, 경로계획

이 강 빈(Kangbeen Lee)

[준회원]



- 2022년 2월 : 울산과학기술원 기계공학과 졸업
- 2022년 3월 ~ 현재 : 울산과학기술원 인공지능대학원 석사과정

<관심분야>

알고리즘, 경로계획

김 태 현(Tae-Hyun Kim)

[정회원]



- 2020년 2월 : 금오공과대학교 컴퓨터소프트웨어공학과 졸업 (학사)
- 2019년 7월 : 나비스오토모티브시스템즈(주) 입사
- 2022년 1월 : 나비스오토모티브시스템즈(주) 대리
- 2022년 6월 ~ 현재 : 한화시스템 연구원

<관심분야>

알고리즘, 임베디드 SW

강 창 근(Chang-Keun Kang)

[정회원]



- 2009년 2월 : 동국대학교 전자공학과 (공학사)
- 2009년 1월 ~ 현재 : 한화시스템 전문연구원

<관심분야>

무인로봇, 임무통제, 경로계획

전 정 환(Jeong hwan Jeon)

[정회원]



- 2007년 2월 : 서울대학교 기계항공공학부 (공학사)
- 2009년 9월 : 메사추세츠 공과대학 항공우주공학과 (석사)
- 2015년 9월 : 메사추세츠 공과대학 항공우주공학과 (박사)
- 2015년 9월 ~ 2019년 8월 : nuTonomy (an Aptiv company since 2017)
- 2019년 8월 ~ 현재 : 울산과학기술원 전기전자공학과/인공지능대학원 조교수

<관심분야>

자율로봇, 모빌리티, 최적 계획/제어