



J. Korean Soc. Aeronaut. Space Sci. 50(3), 157-164(2022)

DOI:https://doi.org/10.5139/JKSAS.2022.50.3.157

ISSN 1225-1348(print), 2287-6871(online)

DTED 맵에서 무인기 경로 생성을 위한 Probabilistic RoadMap 병렬화

노기문¹, 박지훈², 민찬오³, 이대우⁴

Parallelization of Probabilistic RoadMap for Generating UAV Path on a DTED Map

Geemoon Noh¹, Jihoon Park², Chanoh Min³ and Daewoo Lee⁴

Pusan National University, Busan, Republic of Korea^{1,2,4}

Korea Aerospace Industries, Sacheon, Republic of Korea³

ABSTRACT

In this paper, we describe how to implement the mountainous terrain, radar, and air defense network for UAV path planning in a 3-D environment, and perform path planning and re-planning using the PRM algorithm, a sampling-based path planning algorithm. In the case of the original PRM algorithm, the calculation to check whether there is an obstacle between the nodes is performed 1:1 between nodes and is performed continuously, so the amount of calculation is greatly affected by the number of nodes or the linked distance between nodes. To improve this part, the proposed LineGridMask method simplifies the method of checking whether obstacles exist, and reduces the calculation time of the path planning through parallelization. Finally, comparing performance with existing PRM algorithms confirmed that computational time was reduced by up to 88% in path planning and up to 94% in re-planning.

초 록

본 논문에서는 무인기의 경로 계획을 위한 산악 지형, 레이더 그리고 방공망 등을 3차원 환경으로 구현하고, Sampling 기반의 경로 계획 알고리즘인 PRM 알고리즘을 사용하여 경로 계획 및 재계획을 수행하는 방안에 대해 서술한다. 기존의 PRM 알고리즘의 경우 생성된 노드 사이에 장애물 존재 여부를 확인하기 위한 계산이 노드 간 1:1로 이루어지고 연속적으로 수행되어 노드 수나 노드를 연결하는 거리에 계산량이 크게 영향을 받는다. 이러한 부분을 개선하기 위해 제안하는 LineGridMask 기법을 통해 장애물 존재 여부 확인 방식을 단순화하고, 병렬화를 통해 경로 계획의 계산 시간을 감소시킨다. 마지막으로 기존 PRM 알고리즘과의 성능을 비교한 결과, 경로 계획에서는 최대 88%, 재계획의 경우 최대 94%까지 계산 시간이 감소하였음을 확인하였다.

Key Words : UAV(무인기), Probabilistic Roadmap(PRM), 3-D Path Planning(3차원 경로 계획), Parallel computing(병렬 계산)

1. 서 론

최근 무인항공기는 감시, 정찰 등의 군사용 용도뿐만 아니라 통신, 운반 등의 목적으로 그 활용도가 증

가하고 있다. 이렇듯 폭넓은 사용을 위해서는 다양한 조건을 만족해야 하며 그 중 경로 계획은 효과적인 운용을 위해 필수적인 연구 분야이다[1-3]. 경로 계획은 장애물을 회피하며 비용을 최소화하는 것을 목표로

† Received : December 7, 2021 Revised : January 23, 2022 Accepted : February 8, 2022

^{1,2} Ph.D candidate, ³ Senior Researcher, ⁴ Professor

⁴ Corresponding author, E-mail : baenggi@pusan.ac.kr, ORCID 0000-0002-9546-0610

© 2022 The Korean Society for Aeronautical and Space Sciences

한다. 경로 계획은 다양한 방향으로 연구되었으며, 운용하고자 하는 환경 및 플랫폼에 따라 적절히 선택되어야 한다. Dijkstra[4], A*[5], D*[6] 등의 그리드 기반 알고리즘이나 Ant Colony Optimization[7], 유전 알고리즘(Genetic Algorithm)[8] 등의 Evolutionary 알고리즘, Rapidly-exploring Random Tree (RRT)[9], Probabilistic Roadmap(PRM)[10] 등의 샘플링 기반 알고리즘이 있다. 이 중 그리드 기반 알고리즘은 환경을 그리드 형태로 구현해야 하며 하나의 경로를 도출하기 위해 그리드 단위로 계산을 진행하기에 계산량이 많다는 단점이 있다. 또한 Evolutionary 알고리즘은 로컬미니마에 빠질 수 있다는 문제점이 있고 그리드 기반 알고리즘과 같이 계산량이 많다는 문제가 있다. 샘플링 기반 알고리즘은 구성 공간을 무작위로 샘플링하여 생성된 노드를 기반으로 탐색한다. RRT 알고리즘은 적은 계산량으로 경로를 빠르게 생성하는 장점이 있으나 기본적으로 단일 쿼리(Single query) 형태로 맵을 탐색하기에 시작점과 목표점이 바뀌면 매번 새롭게 탐색해야 하는 단점이 있다.

PRM 알고리즘의 경우 샘플링 방법을 통해 맵 전역을 연결하는 로드맵을 생성한다. 이 로드맵을 사용하여 시작점과 목표점을 새롭게 추가하여도 재활용할 수 있다는 장점이 있다. 그러나 로드맵을 생성하기 위해 샘플링되는 노드 수에 크게 영향을 받는다는 단점이 있다. 즉 샘플링되는 노드가 충분하지 않으면 완전한 로드맵을 생성하지 못해 경로를 찾지 못할 수도 있고, 반면에 노드 수가 많아지면 계산량이 증가하고 계산 시간이 증가한다.

본 논문은 PRM 알고리즘을 활용하여 3차원 지형 환경에서 임무를 수행하는 무인기의 경로를 생성하는 것을 목표로 한다. 여러 임무 지점에 대해 동시에 모든 경로를 생성하고 이후 임무 알고리즘에 그 데이터를 전달하는 것을 가정한다. 제안하는 알고리즘에서는 PRM 알고리즘의 계산 시간과 계산량을 단축하는 것을 목표로 노드 사이의 장애물 확인 과정에 병렬화를 적용하였다. 두 노드 사이의 장애물 정보를 미리 만들고 확인하고자 하는 노드에 투영하여 근처 노드들과의 장애물 관계를 병렬적으로 동시에 판단한다. 이 방법을 통해 순차적으로 진행되는 노드 사이 장애물 존재 여부 확인 과정을 축약시키고 결과적으로 전체적인 계산 시간과 계산량을 단축시킨다. 이후 장애물이 추가되는 맵을 업데이트하고 Roadmap을 수정하여 경로를 재계획하는 단계도 추가하였다.

II. 3차원 지형 환경 생성

2.1 DTED

DTED(Digital Terrain Elevation Data)는 지구의 각 지역에 대한 정보를 표현하기 위해 대표적인 데

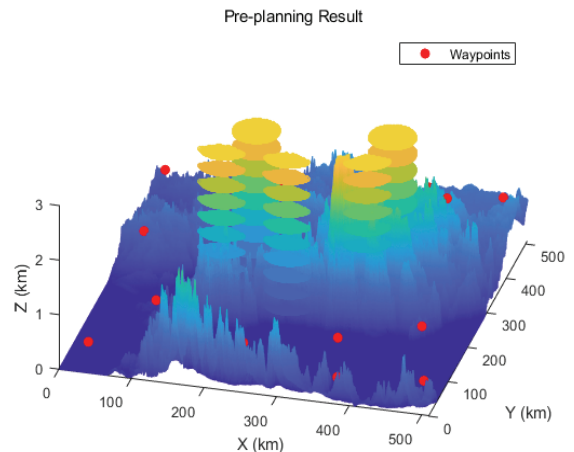


Fig. 1. 3D terrain environment using DTED

이터이다. 지도, 측량 지구 중력 등의 관계한 산출물들을 제작하는 미 영상지도국(National Image and Mapping Agency; NIMA)에서 만들어진 데이터로 군사적 응용 분야에 있어 지형 고도 및 경사 등의 정보를 필요로 하는 시스템을 위해 지형 고도에 대한 균일한 매트릭스 형태로 개발되었다. 고도 데이터는 지표면을 일정 간격으로 나누어 분포된 지점의 고도값을 기록한 자료구조로 되어있다. DTED는 해상도에 따라 Level 0부터 Level 5로 구분된다. 이 중 DTED Level 0은 해상도가 가장 낮으며 cell 크기가 약 $1^{\circ} \times 1^{\circ}$ 이다. 본 연구에서는 대부분이 공개되어 있는 Level 0 데이터를 사용하여 진행되었다.

2.2 3차원 맵 구성

무인기 경로 생성을 위한 시뮬레이션 환경을 구축하기 위해서 구성 공간을 3차원으로 재구성하는 단계를 거친다. DTED를 기반으로 사용하고자 하는 맵의 크기만큼 실제 거리 정보로 변환하게 된다. 이러한 방식으로 DTED 기반의 데이터를 $X \times Y \times Z$ 형태의 3차원 매트릭스로 구성한다. 본 연구에서 사용된 지형 환경은 Fig. 1과 같으며, 맵의 크기는 $510\text{km} \times 510\text{km} \times 3\text{km}$ 이고 각 그리드 간격은 0.3km 로 선정하였다.

III. Parallel PRM

3.1 PRM Algorithm

PRM 알고리즘은 Fig. 2와 같이 구성 공간에서 임의의 샘플링을 통해 노드를 생성하고 노드 간의 지역적 관계를 확인하고 연결하여 하나의 Roadmap 그래프를 생성하는 방식을 사용한다. 이후 시작점과 목표점을 추가하여 최종적으로 두 지점 간을 연결하는 경로를 탐색한다. PRM 알고리즘은 크게 구성 단계와 쿼리 단계로 이뤄진다.

구성 단계는 구성공간을 장애물 영역과 비장애물

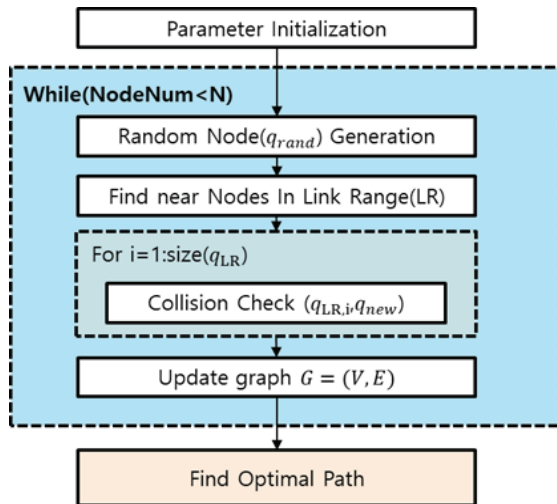


Fig. 2. PRM Algorithm

영역으로 구분한다. 다음으로 구성공간 내에서 정해진 수만큼 임의의 샘플링을 진행하고 샘플링된 노드 중 장애물 영역에 포함된 노드를 제거한다. 남은 노드들 각각을 이웃 노드 선택 조건(일정 거리 이내에 포함되는 노드들 혹은 k개의 가까운 노드들을 선택)에 따라 노드 간에 장애물이 없다면 연결한다. 이러한 방식을 통해 구성공간을 채우는 Roadmap을 생성한다. 쿼리 단계에서는 생성된 로드맵 그래프에 시작점과 목표점을 포함하여 Dijkstra 등을 알고리즘을 활용하여 최단 경로를 탐색한다.

PRM 알고리즘은 샘플링되는 노드의 개수가 무한대로 수렴할수록 경로를 찾지 못할 확률이 0으로 수렴한다. 그러나 알고리즘의 계산 시간 및 계산량의 한계로 인해 노드를 무한정으로 샘플링 할 수 없다. 샘플링되는 노드가 많아지면 더 나은 경로를 생성할 가능성이 증가하지만, 노드 간의 연결 관계를 확인을 위한 장애물을 확인하는 계산이 많아지고 연결되는 노드 수가 많아짐에 따라 저장 데이터가 커지는 단점이 있다.

이러한 특성으로 인해 사전에 샘플링 할 노드 개수와 이웃 노드 선택 조건, 장애물 충돌확인 방식 등을 적절히 조정하는 것이 필수적이다. 본 논문에서는 장애물 충돌방식의 병렬화를 통해 빠른 계산을 가능하게 한다. 구체적인 방식은 3.2장에서 설명한다.

3.2 Collision Check method

기존 PRM 알고리즘의 장애물 충돌확인 방식은 비장애물 영역에 생성된 샘플링된 노드들 사이를 연결한 선이 장애물의 위치로 판단한다. Fig. 3과 같이 장애물과 두 노드 사이의 연결이 중첩된다면 충돌했다고 판단하고 로드맵에 포함되지 않는다.

본 연구에서는 무인기 경로 생성을 위한 맵 환경이 그리드로 구성되기에 Fig. 4와 같은 방식으로 장애물 충돌을 구현한다. 두 노드를 연결한 선을 가장 근접한 그리드((2,3),(3,4),(4,5),(5,5),(5,6),(6,6)(7,7)(8,8))에

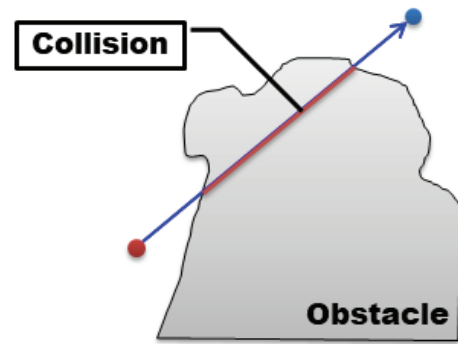


Fig. 3. Collision check method of original PRM

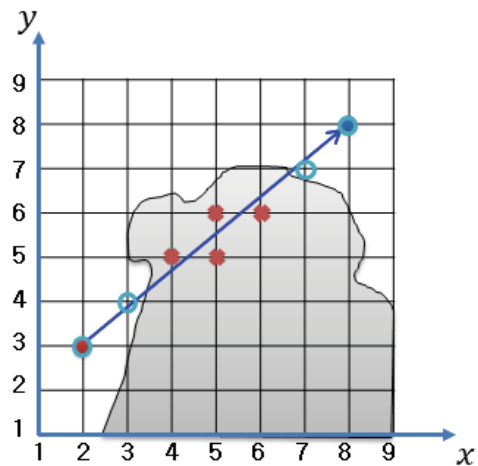


Fig. 4. Collision check method of Parallel PRM

대응시키고 대응된 지점 중 하나라도 장애물에 해당한다면 두 지점 사이는 충돌한다고 판단한다. Fig. 4에서는 ((4,5),(5,5),(5,6),(6,6))이 장애물에 해당하므로 충돌한 상황임을 알 수 있다.

3.3 LineGridMask

기존 PRM 알고리즘의 경우 새로운 노드가 샘플링되면 생성되어있는 나머지 노드 중 연결반경 이내에 있는 노드들과 1:1로 장애물 충돌확인을 수행한다. 이러한 방법으로 수행하는 경우 노드 개수가 많아질수록 계산 시간과 계산량이 증가한다. 이러한 계산을 GPU를 활용하여 병렬적으로 수행하면 전체적인 계산 시간을 단축할 수 있다.

우선 2D를 기준으로 Fig. 5와 같이 기준이 되는 빨간색의 노드를 선택하고 일정 거리에 존재하는 이웃 노드(파란색)를 선택한다. 이후 3.2장에서 설명한 방법을 사용하여 Fig. 6과 같이 두 지점 사이를 임의의 개수로 나누고 이때 나누어진 지점을 파란색으로 나타낸다. 다음으로 나누어진 각 점을 장애물의 그리드 크기에 맞추어 대응시키면 초록색 위치에 대응시킬 수 있다. 초록색 위치의 그리드 정보를 확인하여 해당 그리드가 장애물에 해당하는지 확인하고 모든 지점이 장애물이 아니라면 두 지점을 연결하게 된다.

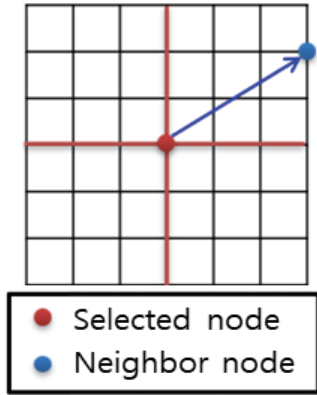


Fig. 5. 1st process of LineGridMask

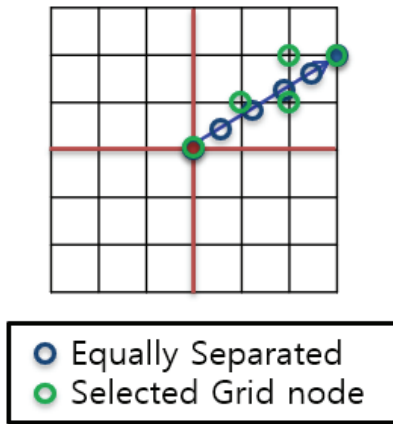


Fig. 6. 2nd process of LineGridMask

이때 이웃 노드가 존재할 수 있는 각 경우에 대해 대응되는 지점이 정해져 있음을 확인할 수 있다. 이때 Fig. 7과 같이 사전에 두 지점 사이를 n 등분하고 그리드 지점에 대응시키는 계산을 사전에 수행하여 두 지점을 연결하는 직선(Line)의 그리드(Grid) 배열을 저장한 마스크(Mask)를 생성한다. 이 마스크를 사용하여 반복적인 계산을 줄이고 이를 통해 단순해진 장애물 확인 단계를 단순화할 수 있다. 결과적으로 Fig. 7과 같이 2D 상황에서 $49(=7*7)$ 개의 경로를 미리 연산한 마스크를 생성하고 각 노드 n 개에 대응하여 최종적으로는 n 개의 노드에 대한 49개의 경로를 생성한다.

이러한 방식을 사용하여 본 논문에서 사용할 3차원 맵에도 같은 방식을 적용한다. Fig. 7을 기준으로 $7*7*7$ 의 3D 상황이라면 총 $343(=7*7*7)$ 개의 마스크를 미리 생성하여 n 개의 노드에 대해 대입하게 된다.

3.4 Parallel PRM with gpuArray

LineGridMask 기법을 사용하면 노드 간 장애물 여부 확인에 걸리는 단계를 크게 단축시킬 수 있다. 하지만 계산 과정 중 각 노드를 순차적으로 계산을 진행하면 생성된 경로를 추가하고 삭제하는 과정이 계산 과정 중 반복된다. 이때 계산 과정과 데이터 교

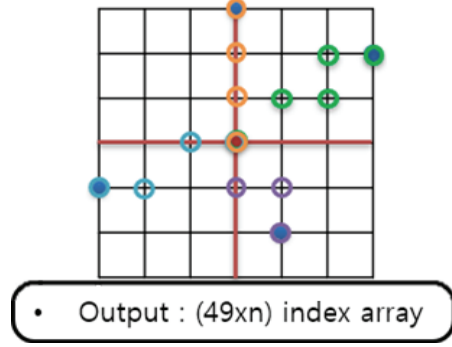


Fig. 7. 3rd process of LineGridMask

환 과정이 반복되게 되어 알고리즘의 병목현상이 발생하게 된다. 이 부분을 해결하기 위해 기존 PRM 알고리즘의 처리 순서를 Fig. 8과 같이 수정한다.

기존의 샘플링된 노드의 순서대로 이웃 노드 간의 장애물 충돌 여부를 확인하던 과정에서 장애물 존재 여부를 확인하는 과정을 알고리즘의 연결 관계 그래프가 완성된 이후 한 번에 확인하는 과정으로 변경한다. 기존의 방식은 하나의 노드가 추가될 때마다 조건에 해당하는 모든 노드와의 장애물 충돌확인을 1:1로 반복적으로 수행하게 되고 동일한 연산이 단일적으로 수행되기에 노드가 늘어날수록 연산량이 증가하게 된다. 변경된 방식은 모든 노드를 한 번에 생성하여 각 노드를 연결하고, 연결된 두 지점 사이에 대해서 장애물 충돌확인을 병렬적으로 수행함으로써 동시에 단순 연산을 수행할 수 있다. 이 방식을 통해 반복 수행되는 데이터 통신을 줄이고 단일 계산을 GPU로 계산하여 기존의 계산 시간을 단축하게 된다.

본 논문에서는 Matlab을 기반으로 시뮬레이션을 진행하였으며 해당 병렬화를 위해 Matlab에서 지원하는 GPU 병렬화 방식인 gpuArray를 사용하였다.

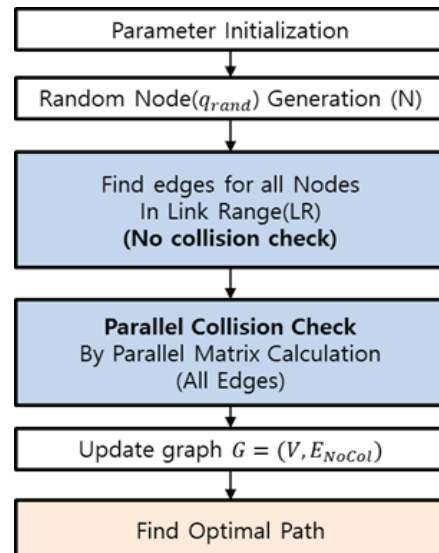


Fig. 8. Parallel PRM algorithm

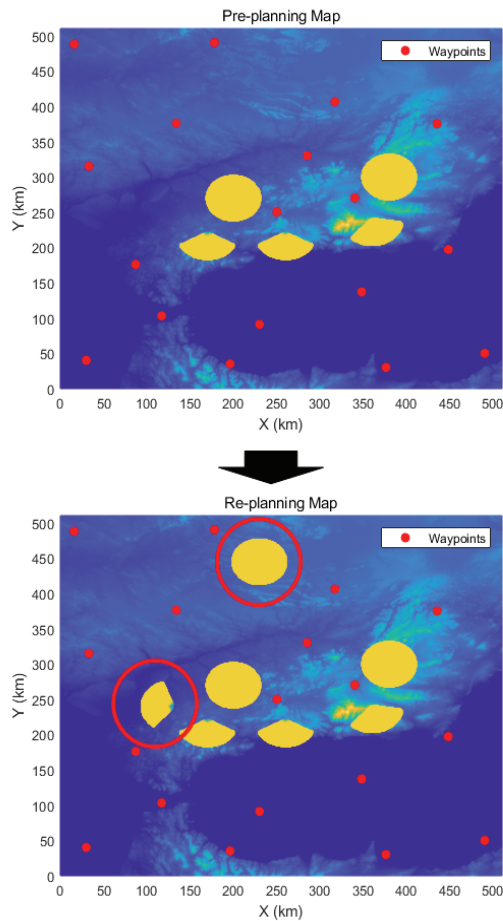


Fig. 9. Map with added obstacles

3.5 Path Replanning

본 장에서는 맵 환경이 변하는 경우를 가정하여 변화된 장애물 환경에 대해 경로 재계획을 수행하는 방식에 대해 설명한다. PRM 알고리즘을 통해 얻어진 결과는 노드 간의 연결 Roadmap이 $X \times Y \times Z$ 형태의 매트릭스로 구성된다.

Figure 9와 같이 장애물이 추가되는 구역(빨간원)이 발생하면 그 부분에 해당하는 부분에 대해 기존 연결들과 충돌확인을 수행한다. 이때 기존 연결과 추가된 장애물이 충돌하는 경우 그 연결들을 제거한다. 이 과정을 통해 수정된 Roadmap을 생성한다. 이후 수정된 Roadmap에서 경로 탐색을 재수행하여 새로운 경로를 탐색한다.

IV. 시뮬레이션

4.1 시뮬레이션 시나리오

본 논문에서 제안하는 Parallel PRM의 유효성을 검증하기 위해 시뮬레이션을 통한 성능 비교를 수행하였다. 동일한 시뮬레이션 환경에서 기존의 PRM과 Parallel PRM의 성능을 비교한다. 시뮬레이션의 순서

Table 1. Specification of Simulation PC

CPU	AMD Ryzen 7 5800X 3.80 GHz
RAM	32GB
GPU	NVIDIA GeForce RTX 3060

는 다음과 같다. 우선 주어진 맵에서 20개의 지점에 대해 190개의 경로를 생성한다. 이후 추가되는 장애물에 대해 경로 재계획을 수행한다. 이러한 시뮬레이션 과정을 통해 시간은 경로 계획 속도와 경로 재탐색 속도를 비교하고 경로는 전체 190개 경로의 합으로 비교한다. 본 시뮬레이션에서 사용된 PC의 스펙은 Table 1에 정리하였고 CUDA 코어 수는 3,584개이다. 시뮬레이션은 Matlab R2021a를 사용해서 진행하였다.

4.2 경로 계획 시뮬레이션 결과

제시한 시나리오에 따라 기존 PRM과 제안하는 Parallel PRM의 결과를 비교하였다. 20개의 지점 간의 연결 경로는 Fig. 10의 옥색으로 생성되었다.

우선은 동일한 조건에서 노드 수를 0개에서 2,000개로 100개씩 증가시키며 계산 시간과 190개의 전체 경로값을 비교하였다.

Figures 11, 12와 Tables 2, 3은 각각 계산 시간과 경로값을 비교한 결과이다. 노드 수가 0개에서 600개까지는 맵 전역을 연결하는 Roadmap을 완전히 생성하지 못하였다. 600개 이후의 결과를 확인하면 계산 시간의 경우 노드 수가 증가함에 따라 계산 시간이 증가함에도 제안하는 알고리즘의 계산 시간은 700개 일 때 PRM 대비 84% 감소한 수치를 보여주었으며, 2,000개까지 최대 88% 감소한 계산 시간을 보여주었다. 병렬화를 수행하지 않는 기존 PRM의 경우 노드 수에 따라 더 가파른 증가 추세를 보여주었으나 제안하는 Parallel PRM의 경우 원만한 증가 추세를 보여주었다. 전체 경로값의 경우 비슷한 경향으로 감소하였으며 경로 탐색을 개선하는 부분은 포함되어 있지 않기에 두 경로값은 거의 차이가 없다고 분석된다.

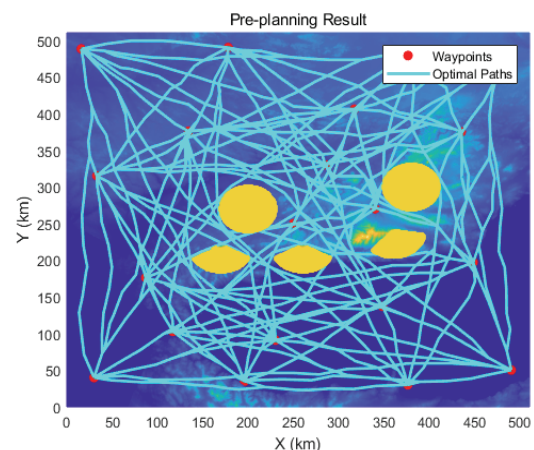


Fig. 10. Pre-planning result

다음은 동일한 조건에서 노드 간의 연결반경을 1km에서 35km로 1km씩 증가시키며 계산 시간과 190 개의 전체 경로값을 비교하였다. Tables 4, 5와 Figs. 13, 14는 각각 계산 시간과 경로값을 비교한 결과이다. 연결반경이 1km에서 20km까지는 맵 전역을 연결하는 Roadmap이 완전히 생성되지 못하였다. 연결 반경 20km 이후의 결과를 확인하면 계산 시간의 경우 연결반경이 증가함에 따라 계산 시간이 증가하는

경향을 보여주었다. 그러나 모든 경우에 대해 제안하는 알고리즘이 기존 PRM의 계산 시간 대비 최소 84%에서 최대 88%까지 감소된 계산 시간을 보여주었다. 특히 연결반경이 증가함으로써 계산량이 늘어나는 조건에서 계산 시간이 적은 비율로 증가하는 것을 확인할 수 있다. 경로값의 경우 제안하는 알고리즘과 기존 PRM의 경로값이 감소하는 경향이 유사하고 앞선 노드 수 변경 조건과 같이 두 경로값은 거의 차이가 없다고 분석된다. 이러한 결과를 통해 계산량에 영향을 미치는 노드 수와 연결반경을 변경하는 조건에서 병렬화를 통한 방식이 개선된 계산 시간을 보여주는 것을 확인하였다.

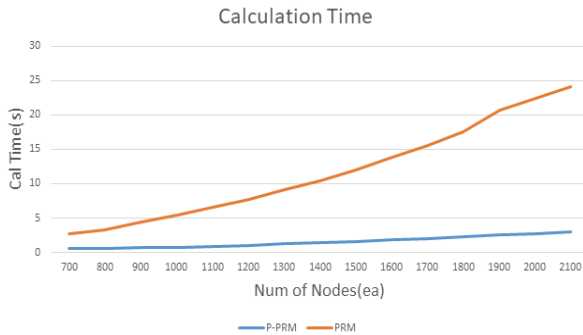


Fig. 11. Calculation time as the number of nodes increases

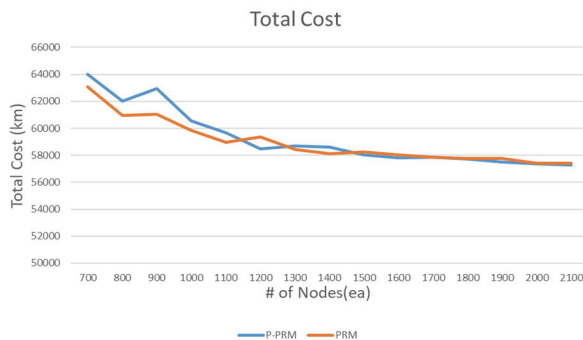


Fig. 12. Total path cost as the number of nodes increases

Table 2. Calculation time as the number of nodes increases

Cal Time(s)	PRM	Parallel PRM
Mean	11.72	1.55
Min	2.71	0.52
Max	24.03	2.99

Table 3. Total path cost as the number of nodes increases

Total Cost(km)	PRM	Parallel PRM
Mean	58,951	59,233
Min	57,410	57,276
Max	63,071	64,017

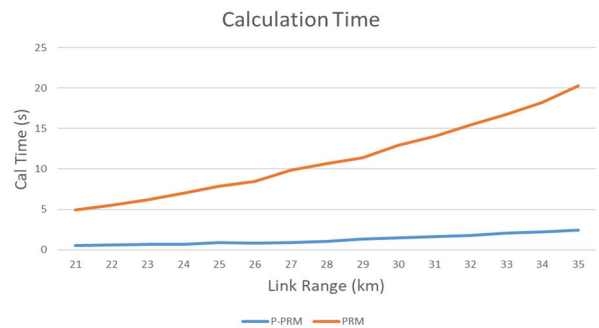


Fig. 13. Calculation time as the Link Range increases

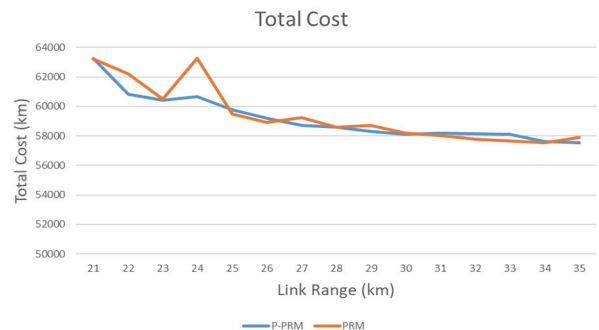


Fig. 14. Total path cost as the Link Range

Table 4. Calculation time as the Link Range increases

Cal Time(s)	PRM	Parallel PRM
Mean	11.3	1.28
Min	4.96	0.52
Max	20.28	2.48

Table 5. Total path cost as the Link Range increases

Total Cost(km)	PRM	Parallel PRM
Mean	59,414	59,163
Min	57,550	57,527
Max	63,274	63,247

4.3 경로 재계획 시뮬레이션 결과

다음은 맵의 변화에 따른 경로 재계획 결과 비교이다. 경로 재계획 결과는 Fig. 15와 같으며, 수정된 경로는 자홍색으로 표시되어 있다.

재계획 결과는 Parallel PRM의 성능 비교와 마찬가지로 노드 수와 연결반경을 변화시키며 계산 시간을 확인하였다.

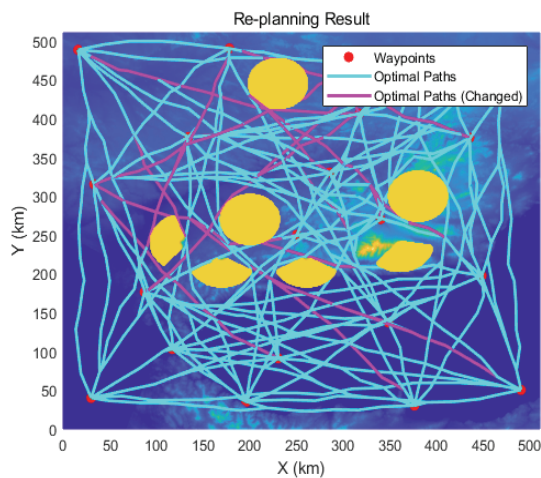


Fig. 15. Re-planning result

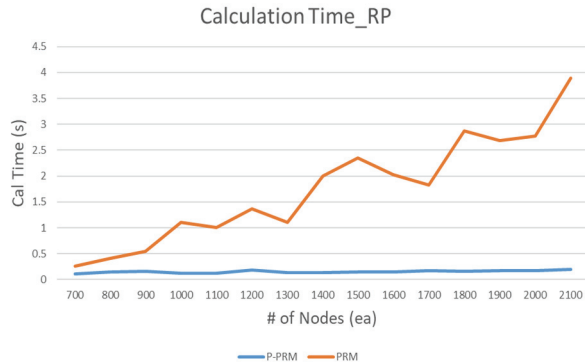


Fig. 16. Calculation time for Replanning as the number of nodes increases

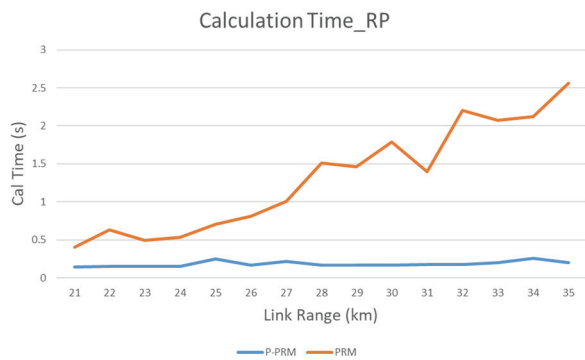


Fig. 17. Calculation time for Replanning as the Link Range increases

Figures 16과 17은 각각 노드 수와 연결반경 증가에 따른 계산 시간을 나타낸다. Fig. 16을 보면 PRM의 경우 노드 수가 700개일 때 약 0.26초에서 2,100개일 때 약 3.9초까지 증가하는 것을 볼 수 있다. 제안하는 알고리즘의 경우 노드 수 700개일 때 약 0.1초부터 2,100개일 때 약 0.2초로 증가하지만 기존 PRM의 계산 시간 증가 추세 대비 제안하는 알고리즘은 적은 폭으로 증가하는 것을 확인할 수 있다. Fig. 17에서도 PRM의 경우 0.4초에서 최대 2.6초까지 증가하였으나 제안하는 알고리즘은 0.15초에서 0.2초로 적은 폭으로 증가하는 것을 확인할 수 있다. 종합적으로 계산 시간이 최소 60%에서 최대 94%까지 감소하는 결과를 얻었다. 이를 통해 기존 PRM 알고리즘보다 제안하는 알고리즘의 경로 계획 및 재계획 성능이 개선된 계산 시간을 가지는 것을 알 수 있으며, 노드 수나 연결반경에 의해 계산이 필요한 노드가 늘어날수록 병렬화를 통한 계산 시간 단축 효과가 커지는 것을 확인할 수 있다.

V. 결론

본 논문은 넓은 지형의 임무 수행을 목적으로 한 무인기의 경로 생성을 위한 실제 지형과 비슷한 3차원 시뮬레이션 환경에서 PRM 알고리즘을 통해 경로 계획과 재계획을 목표로 한다. 이때 사용되는 PRM 알고리즘의 병렬화를 통한 계산 속도 개선을 위해 LineGridMask 방식을 제안하였다.

넓은 지형에서의 경로 생성 조건을 모사하기 위해 DTED 기반 3차원 지형을 생성하고, 추가적으로 레이다, 방공망, 약천후 등의 장애물을 모델링으로 추가하였다.

경로 생성을 위한 PRM 알고리즘의 경우 무작위 샘플링 알고리즘의 일종으로 탐색 공간에 Roadmap을 생성하여 다수 경로점에 대해 빠르게 경로를 탐색할 수 있다는 장점이 있다. 하지만 Roadmap을 생성하는 과정에서 노드와 노드 간의 장애물 여부 확인이 1:1로 진행되고 연속적인 계산 과정에 의해 경로 생성 중 많은 연산 시간을 차지하게 된다. 따라서 본 논문에서는 장애물 여부 확인을 LineGridMask 기법을 통해 단순화시키고 병렬화를 통해 연속적인 계산을 동시에 진행하게 된다. 이러한 방법으로 기존 PRM 알고리즘과의 성능 비교 결과 연산 속도가 경로 계획에서는 최대 88%, 재계획의 경우 최대 94%까지 계산 시간을 감소시키는 데 성공하였다. 이때 생성된 경로는 각 노드 간의 연결을 기반으로 하여 실제 비행에 바로 적용하기에는 제한적인 부분이 있으나 spline 등의 후처리를 적용한다면 더욱 안정적인 경로를 생성할 수 있을 것으로 기대된다.

후 기

본 논문은 한국산업기술평가관리원의 재원으로 항공우주부품기술개발사업-수출유망부품및핵심기술개발(과제번호: 20002712) 지원으로 작성되었습니다.

References

- 1) González-Arribas, D., Soler, M., Sanjurjo-Rivo, M., Kamgarpour, M. and Simarro, J., "Robust aircraft trajectory planning under uncertain convective environments with optimal control and rapidly developing thunderstorms," *Aerospace Science and Technology*, Vol. 89, 2019, pp. 445~459.
- 2) Han, H., Qiao, D., Chen, H. and Li, X., "Rapid planning for aerocapture trajectory via convex optimization," *Aerospace Science and Technology*, Vol. 84, 2019, pp. 763~775.
- 3) Noreen, I., Khan, A. and Habib, Z., "Optimal path planning using RRT* based approaches: a survey and future directions," *International Journal of Advanced Computer Science and Applications*, 2016, pp. 97~107.
- 4) Dijkstra, E. W., "A note on two problems in connexion with graphs," *Numerische Mathematik*, Vol. 1, No. 1, 1959, pp. 269~271.
- 5) Hart, P. E., Nilsson, N. J. and Raphael, B., "A formal basis for the heuristic determination of minimum cost paths," *IEEE Transactions on Systems Science and Cybernetics*, Vol. 4, No. 2, 1968, pp. 100~107.
- 6) Stentz, A., "Optimal and efficient path planning for unknown and dynamic environments," Carnegie Mellon University, Pittsburgh, The Robotics Institute, 1993.
- 7) Gerke, M., "Genetic path planning for mobile robots," *Proceedings of American Control Conference (ACC)*, Vol. 4, 1999, pp. 2424~2429.
- 8) Garcia, M. A. P., Montiel, O., Castillo, O., Sepúlveda, R. and Melin, P., "Path planning for autonomous mobile robot navigation with ant colony optimization and fuzzy cost function evaluation," *Applied Soft Computing*, Vol. 9, No. 3, 2009, pp. 1102~1110.
- 9) LaValle, S. M. and Kuffner Jr., J. J., "Randomized kinodynamic planning," *The International Journal of Robotics Research*, Vol. 20, No. 5, 2001, pp. 378~400.
- 10) Kavraki, L. E., Svestka, P., Latombe, J. C. and Overmars, M. H., "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Transactions on Robotics and Automation*, Vol. 12, No. 4, 1996, pp. 566~580.