



J. Korean Soc. Aeronaut. Space Sci. 47(12), 874-880(2019)

DOI:https://doi.org/10.5139/JKSAS.2019.47.12.874

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

무인기의 복잡한 지형 회피를 위한 Convex Hull 기반의 계층형 Visibility Graph

임대희¹, 박지훈², 민찬오³, 장환철⁴, 이대우⁵

Layered Visibility Graph With Convex Hull to Avoid the Complex Terrain for UAV

Daehee Lim¹, Jihoon Park², Chanoh Min³, Hwanchol Jang⁴ and Daewoo Lee⁵
Pusan National University^{1,2,5}, Korea Aerospace Industries³, Agency for Defense Development⁴

ABSTRACT

This paper introduces a method which can be effectively used for the path planning of UAV in a realistic map which has mountainous terrains, air defense networks and radars based on the Visibility Graph. Existing studies of Visibility Graph have been studied mainly for simple shape obstacles in 2-dimensional environment such as self-driving cars which avoid buildings. However, for UAV, Visibility Graph must be used in 3-dimensional environment for the variance of altitude. This occurs significant elapsed time increase because of the increase of the amount of the visibility of node sets. To solve this problem, this paper decrease the number of nodes which consists the complex terrain environments using convex hull based on Layered Visibility Graph. With convex hull method, this paper confirmed that the elapsed time is decreased about 99.5% compared to the case which has no decrease of the number of nodes.

초 록

본 논문에서는 무인기의 경로 계획을 위한 맵 분할 방법론 중 하나인 Visibility Graph를 산악 지형, 방공망, 그리고 레이더 등의 장애물이 존재하는 실제적인 3차원 환경에서 효율적으로 사용하기 위한 방안에 대해 서술한다. 기존의 가시성 그래프는 빌딩 사이를 주행하는 자율주행 자동차와 같이 주로 2차원 환경에서 간단한 형상의 장애물에 대해 연구되어왔다. 무인기 분야에서 사용하기 위해서는 고도 변화를 위해 3차원 가시성 그래프가 적용되어야 하는데, 3차원 가시성 그래프의 경우 2차원 환경에 비해 가시성을 판단해야 하는 노드 쌍이 매우 많아진다. 이에 더해 복잡한 다각형으로 이루어진 산악 지형은 가시성 그래프의 계산 시간을 더욱 상승시키는 요인으로 작용한다. 이러한 문제를 해결하기 위해 본 논문에서는 맵을 일정 고도로 분할하는 계층형 가시성 그래프 방식을 기반으로 복잡한 산악 지형을 Convex Hull 개념을 활용하여 노드 수를 감소시켜 계산 시간을 줄이는 방법에 대해 서술하며, 노드 수를 감소시키지 않은 상태와의 계산 시간을 비교한 결과 계산 시간이 약 99.5% 감소하였음을 확인하였다.

Key Words : Fixed-wing UAV(고정익 무인기), Layered Visibility Graph(가시성 그래프), 3-D Path Planning (경로 계획), 3D Visibility Graph(3차원 가시성 그래프)

† Received : November 5, 2019 Revised : November 27, 2019 Accepted : November 27, 2019

¹ Graduate student, ² Ph.D candidate, ^{3,4} Senior Researcher, ⁵ Professor

⁵ Corresponding author, E-mail : baenggi@pusan.ac.kr, ORCID 0000-0002-9546-0610

© 2019 The Korean Society for Aeronautical and Space Sciences

I. 서 론

무인 비행체(Unmanned Aerial Vehicle, UAV)는 최초 군에서 정찰 등의 임무를 위해 개발되었으나, 최근에는 민간, 산업용으로 그 용도가 넓어졌으며, 무인기의 수요가 늘어남에 따라 그와 관련된 연구들이 많이 진행되고 있다. 경로 계획은 무인기의 운용에 있어 필수적인 요소 중 하나로, 경로 계획은 최적성, 즉 목표지점까지 도달할 수 있는 최단 경로를 도출시키는 것과 그 경로를 도출해내는데 소요되는 시간을 최소화시키는 것이 중요하다. 따라서 모든 경로 계획 알고리즘들은 최적 경로를 최단시간 내에 도출해내는 것을 목적으로 개발되었다. 그러나 군용 정찰기와 같이 넓은 지역에서 운용되는 무인기의 경우 경로 계획을 위해 구성되는 맵의 크기가 매우 커져 경로 계획 도출에 필요한 시간이 기하급수적으로 증가하는 문제가 발생하였다. 이러한 문제를 해결하기 위해 맵을 단순하게 표현하는 방법에 대한 연구가 많이 진행되었으며, 이에 대한 한 가지 방법으로 Quadtree와 같은 Cell Decomposition, Voronoi와 Probabilistic RoadMap(PRM), 가시성 그래프와 같은 그래프 기반 맵 분할 방법 등이 대두되었다[1-3].

그 중 가시성 그래프는 경로 계획 단계 이전에 맵을 분할하는 방법 중 하나이며, 장애물의 경계선의 꼭짓점들의 연결 관계를 이용하여 충돌 회피를 포함하는 맵 분할 방법의 일종이다. 그리고 맵 분할을 통해 획득한 맵 정보를 바탕으로 A*, Dijkstra 알고리즘 [4,5] 등의 경로 계획 알고리즘들을 사용하여 무인기의 경로를 획득한다. 가시성 그래프는 맵 상에서 장애물이 아닌 지점, 즉 free space를 효율적으로 다룰 수 있다는 장점이 있다. 이 장점은 격자 기반의 맵 분할 방식에 비해 맵을 효율적으로 분할하여 계산 시간을 보다 단축시킬 수 있으며, Rapidly Exploring Random Tree(RRT)와 같은 랜덤 샘플링 방식 알고리즘[6]에 비해서도 불필요한 공간에 샘플링하지 않는다는 부분에 있어 효율적이라고 할 수 있다. 이러한 장점들 덕분에 가시성 그래프는 많은 연구 분야에서 응용되었으며, 초기의 가시성 그래프는 무인기보다는 주로 로봇의 이동 경로 또는 자율주행 자동차의 경로 계획에 사용되었다.

가시성 그래프가 2차원에서 주로 사용됐던 원인으로 당시의 컴퓨터가 3차원 가시성 그래프의 계산량을 처리하기에 그 성능이 좋지 않았던 점도 있었지만 지상의 장애물들은 빌딩과 같이 그 형상이 삼각형, 또는 사각형 등 단순한 형태인 점 역시 큰 원인 중 하나였다. 그러나 무인기의 운용에도 가시성 그래프가 쓰이기 시작하자 3차원 환경의 가시성 그래프를 사용하는 것이 불가피해졌고, 지속적인 컴퓨터의 성능이 향상됨에 따라 3차원 가시성 그래프의 계산량을 처리할 수 있게 됨에 따라 3차원 가시성

그래프에 대한 연구들이 진행되었다.

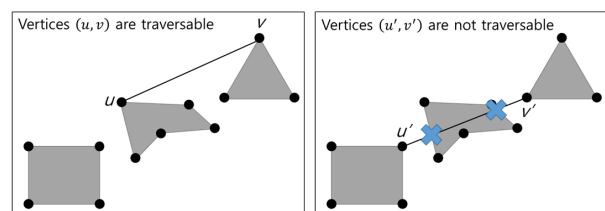
본 논문은 3차원 가시성 그래프를 활용하여 넓은 산악 지형에서 임무를 수행하는 무인기의 최적 경로 생성을 도출하기 위해, 실제와 유사한 3차원 시뮬레이션 환경을 구축하는 것이다. 가시성 그래프는 장애물을 다각형으로 모사하여 꼭짓점을 정점으로 사용하는데, 산악 지형은 매우 복잡하고 많은 다각형으로 이루어져있어 가시성을 파악해야 하는 정점의 수가 매우 많아진다. 이러한 문제점을 해결하기 위해 본 논문에서는 맵을 일정 고도마다 층으로 나누는 계층형 가시성 그래프(Layered Visibility Graph, 이하 LVG)를 기반으로 Convex Hull 기법을 통해 기존의 3차원 가시성 그래프에 비해 노드 수를 획기적으로 감소시켜 계산 시간을 감소시키고자한다.

II. Literature Reviews

2.1 Visibility Graph

Visibility Graph는 맵 상의 장애물의 꼭짓점에서 추출된 정점들에 대해 가시적인 관계를 갖는 연결쌍들의 정보가 포함된 인접 행렬과 각 연결쌍들의 거리 정보들의 집합이다. 여기서 가시적인 관계란 장애물의 내부에 포함되지 않는 두 정점 u, v 가 있을 때 두 정점 사이에 장애물이 없다면, 즉 Fig. 1(a)와 같이 정점 u 의 시야에서 정점 v 가 보이고, 동시에 정점 v 에서 정점 u 가 보일 때 두 정점 u, v 는 서로 가시적인 관계에 있다고 정의한다. 또한 Fig. 1(b)와 같이 두 정점 사이에 장애물이 존재하여 두 정점을 잇는 직선이 장애물을 통과하는 경우 두 정점은 가시적인 관계에 있지 않다고 정의한다.

가시적 관계 파악은 한 개의 정점에 대해 자신을 제외한 맵 상의 모든 정점에 대해 실시하며, 임의의 두 정점이 가시적인 관계임이 판단되면 두 정점 사이의 거리를 추정하여 거리 정보를 저장한다. 이 때 거리 추정은 주로 유클리드 거리를 사용한다. 맵 상의 모든 정점에 대해 가시적 관계 파악이 끝나면 그 결과로 모든 정점들의 가시적 관계 정보가 담긴 인접 행렬과 그에 대한 거리 정보가 도출되며, 이를 기반으로 하여 A*, Dijkstra 알고리즘 등의 경로 계획 알고리즘을 통해 Fig. 2와 같이 최단 경로를 도출한다.



(a) Traversable vertices (b) Not traversable vertices

Fig. 1. Conditions whether vertices are traversable or not

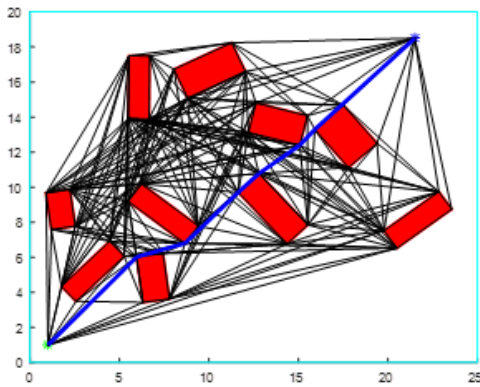


Fig. 2. Path planning with Dijkstra based on Visibility Graph

2.2 3차원 환경에서의 가시성 그래프

가시성 그래프는 주로 2차원 환경에서 많이 사용되었다. 가시성 그래프가 3차원 환경에서 사용되지 않았던 이유는 서론에서 서술한 바와 같이 3차원 환경으로 확장할 시에 계산량이 많아지기 때문이다.

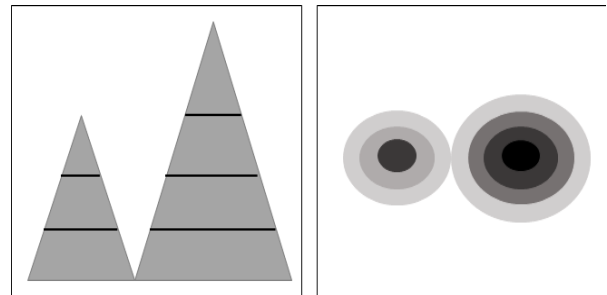
가시성 그래프의 3차원 적용에 있어 가장 중요한 요소는 서로 다른 고도에 위치한 정점들의 가시적 관계 판단이다. 이를 위해서는 우선 3차원 장애물을 다수의 다각형으로 묘사하는 작업이 필요하다. 그러나 복잡한 형상의 장애물에 대해 다수의 다각형으로 묘사하는 것은 그 양이 많다. 게임과 같이 지형을 따라 움직이는 물체에 대해서는 해당 방법이 필연적이기에 계산량의 증가를 감수해야하지만 무인기의 경우 장애물의 회피가 목적이기 때문에 장애물을 더욱 단순하게 묘사하여도 무인기의 운용에 관계가 없다 [7]. 이에 따라 맵을 단순하게 표현하는 방법에 대해 많은 연구들이 진행되었다[8-11]. 본 논문에서는 3차원 맵의 고도를 일정한 간격으로 등분하여 층으로 나누는 LVG에 대해 서술하고자 한다.

2.3 Layered Visibility Graph (LVG)

LVG는 앞 절에서 서술한 바와 같이 등고선을 만드는 것과 같이 맵의 고도를 일정한 간격으로 나누고, 각 층에서 정점을 추출하여 가시성을 판단하는 방법이다. 이를 도식화하면 Fig. 3과 같다.

Figure 3과 같이 맵을 나눈 뒤에는 2차원 가시성 그래프에서 실시한 것과 동일한 방법으로 각 정점간 가시성 관계를 파악한다. 그러나 3차원 가시성 그래프에서의 가시성 파악은 2차원에서의 방법을 그대로 사용할 수 없는 문제가 있으며, 이를 Fig. 4에 나타내었다.

Figure 4는 3층으로 나눈 계층형 가시성 그래프에서 Dijkstra 알고리즘을 활용해 경로를 도출하였으나 경로가 장애물을 통과하는 잘못된 결과의 한 예이며, Fig. 4에서 적색, 녹색, 청색 선은 3층으로 나누어진 맵에서 각 층을 구분하기 쉽도록 1층, 2층, 3층에 놓



(a) Front view (b) Top view

Fig. 3. Front view and Top view of k-layer Visibility Graph

Map : Layered Visibility Graph / Path Planning : Dijkstra

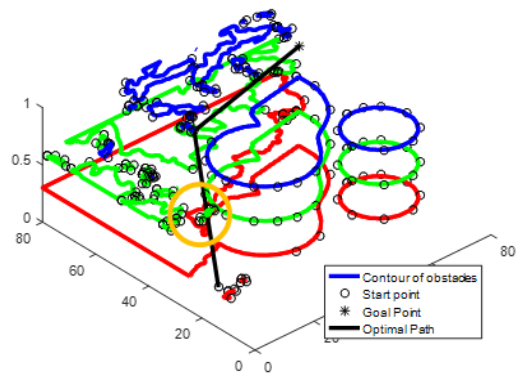


Fig. 4. A situation that there are no obstacles between two vertices but there is a collision with obstacle actually

여있는 노드들을 이은 것이다. Fig. 4를 참조하면 1층과 3층을 가로지르는 경로를 보면 경로에 직접 간섭하는 장애물이 없음을 알 수 있으며, 이 때문에 두 노드 사이에는 장애물이 없다고 생각할 수 있다. 그러나 해당 경로는 실제로는 1층과 2층에 걸쳐 존재하는 산악 지형을 통과하는, 즉 장애물을 통과하는 잘못된 경로이며, Fig. 4를 Fig. 5와 같이 평면도로

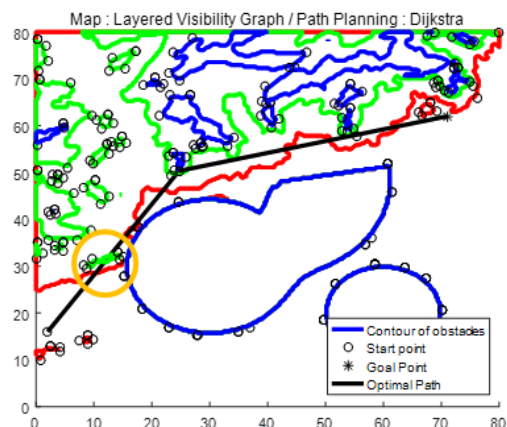


Fig. 5. Planform of Fig. 4

보면 이에 대해 파악할 수 있다. 이러한 문제를 해결하기 위해서는 고도가 다른 두 정점 간의 가시성을 파악할 때 두 고도 사이에 있는 모든 고도들에 존재하는 장애물들에 대해 충돌 확인이 필수적으로 고려되어야 한다.

III. 3차원 임무 환경 모델링

3.1 3차원 임무 환경 개요

본 논문의 목표는 LVG를 이용하여 산악 지역 임무 수행 환경에서 무인기의 최적 경로를 생성하는 방법을 제시하고 시뮬레이션을 통해 검증하는 것이다. III장은 실제 무인기의 임무 상황에 가까운 환경 구축을 위한 내용이 서술되어 있다.

기존에 수행된 LVG에 대한 연구에서는 단순한 형태의 장애물이 포함된 3차원 환경을 사용한 것에 비해, 본 논문에서는 DTED(Digital Terrain Elevation Data)를 이용하여 실제 산악지형을 모델링하였으며 레이더 및 방공망을 모델링하고 적용함으로써 더욱 실제 임무 상황에 가까운 환경을 구현하였다. 그러나 이러한 복잡한 환경에서는 경로 생성에 필요한 노드의 수가 기하급수적으로 증가하기 때문에 연산 시간 또한 이에 비례하여 증가한다. 이를 해결하기 위해 Convex Hull을 적용한 노드 수 감소법을 제시하였다.

3.2 DTED를 이용한 지형 모델링

본 논문에서는 실제 산악지형을 시뮬레이션 환경에 구현하기 위해 DTED를 적용하였다. DTED란 Fig. 6과 같이 지형고도정보를 행렬로 표현한 표준 디지털 정보로, 위도와 경도를 레벨에 따라 일정 간격으로 양자화하여 그 크기와 동일한 행렬로 표현되며 행렬의 원소는 그 점의 고도를 의미한다. DTED의 레벨이란 위도와 경도에 대한 분해능을 의미하며, 본 논문에서는 레벨 0(약 900m 단위)을 적용하였다. 이 때 위도와 경도의 단위는 arc sec로 표현되므로 m단위로 환산했을 시 위도에 따라 경도의 값이 달라지는 특징이 있다.

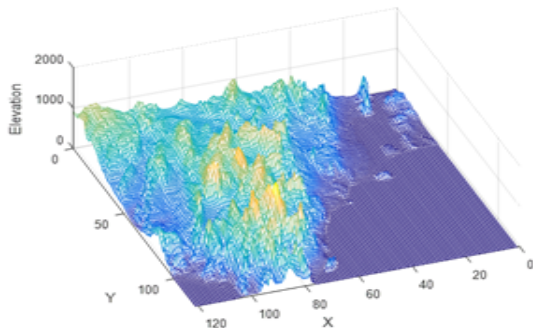


Fig. 6. Figure of DTED

3.3 3차원 장애물 모델링

실제 무인기의 임무 환경에서의 경로 계획을 위해서는 지형 이외에도 외부 위협요소 회피 또한 고려되어야 한다. 본 연구에서는 외부 위협요소로 악천후(적란운 등), 적 레이더, 지대공 미사일을 고려하였으며, 이를 장애물로 간주하였다. 이는 Figs. 7-9와 같이 3차원 가우시안 확률분포로 모델링하였다. Fig. 10에서는 DTED를 이용한 지형 정보와 모델링된 장애물 정보를 합산하여 3차원 Layered Map을 생성하는 과정을 나타내었다. 본 논문에서 대공망을 모델링하여 장애물을 표현하는 경우 대공망의 고도가 높지 않아 경로가 장애물을 회피하는 모양이 아닌 매우 단순한 형태로 도출되어 해당 문제를 해결하기 위해 대공망을 적란운(원통형 장애물)으로 대체하여 시뮬레이션을 진행하였다.

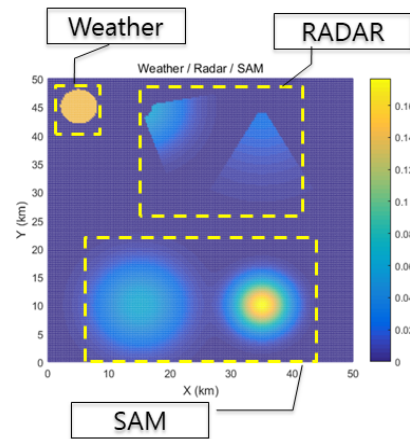


Fig. 7. 3-D Obstacle modeling

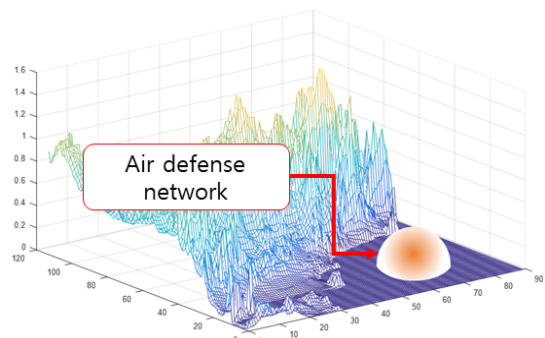


Fig. 8. Model of Air defense network in 3D

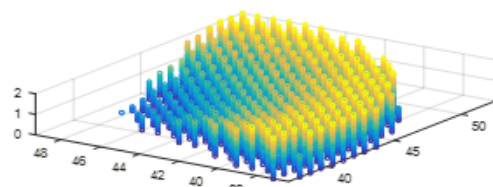


Fig. 9. Model of Radar in 3D

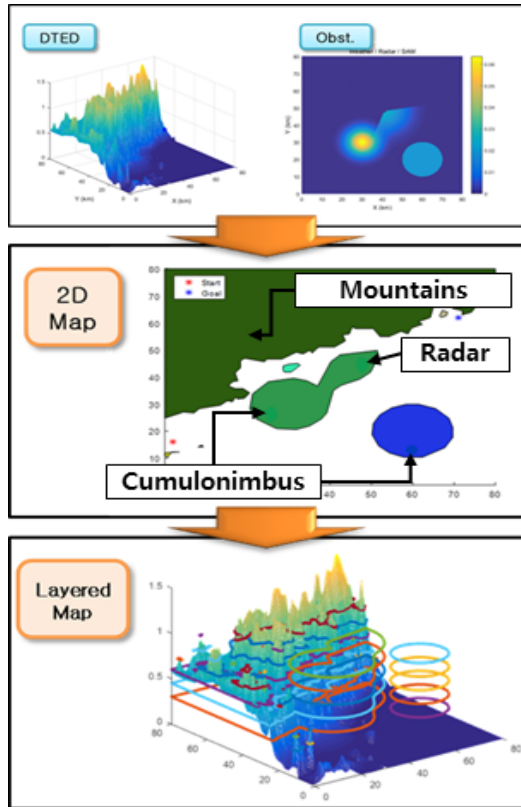


Fig. 10. 3-D layered map with obstacles

3.4 Convex hull을 적용한 노드 수 감소

LVG에서는 3차원 지형의 정보에 대하여 특정 고도의 등고선 정보를 획득하여 장애물로 인식한다. 그러나 DTED 기반의 실제 지형의 등고선 정보는 불규칙적이고 복잡한 형상을 지니므로 Fig. 11의 좌측과 같이 표현에 필요한 노드와 모서리의 수가 매우 많다. 그러므로 기존의 LVG에서 사용하는 방법을 적용 시 맵의 크기에 따라 필요한 연산량은 기하급수적으로 증가한다.

이를 극복하기 위해, 본 논문에서는 Convex Hull을 활용하여 장애물을 구성하는 폴리곤의 형상을 단순화하고 노드 수를 감소시켰다. Convex Hull이란 2차원 평면상에 여러 개의 점이 존재할 때, 가장 외곽에 위치한 점들을 연결하여 형성된 다각형 내부에 모든 점이 위치하게 하는 알고리즘이다. 이를 Fig.

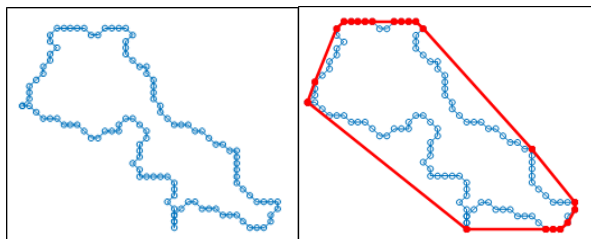


Fig. 11. Node reduction using convex hull

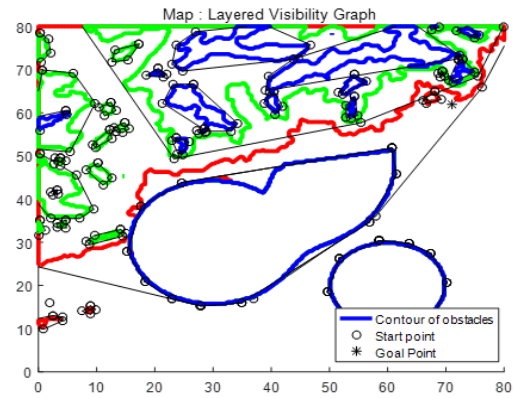


Fig. 12. Layered map using node reduction

11과 같이 Layered Map 상의 장애물에 적용하여 노드의 수를 크게 감소시켰다. 또한 특정 모서리 부분에서 불필요한 노드의 수를 줄이기 위해 일정 Tolerance 이내의 점을 제거한다. Fig. 12는 Fig. 11의 과정을 통해 생성된 Layered Map에 Convex Hull을 적용하여 노드 수를 감소시킨 결과를 나타낸다.

IV. 시뮬레이션

4.1 시뮬레이션 시나리오

본 논문의 3.4절에서 제안한 방법의 유효성을 검증하기 위해 시뮬레이션을 통한 성능 비교를 실시하였다. Convex Hull을 통한 노드 수 감소 기법을 적용하지 않은 case를 비교군으로 사용하였다. 시뮬레이션의 과정은 Fig. 13과 같다.

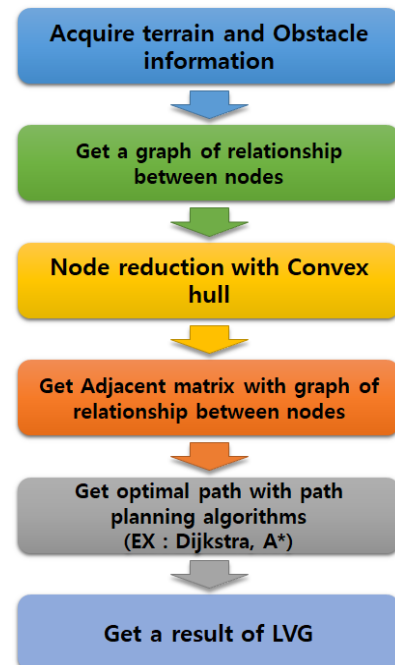


Fig. 13. Sequence of simulation

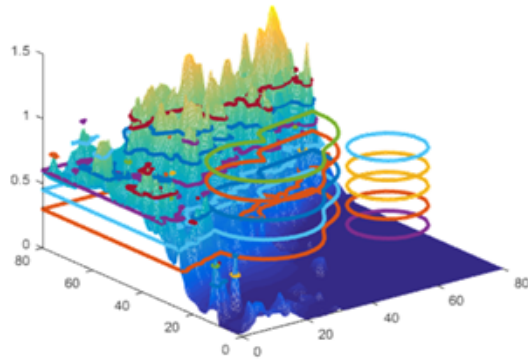


Fig. 14. 3-D layered map for simulation

시뮬레이션 환경은 MATLAB으로 구현하였다. DTED 지형정보를 이용하여 80 x 80 x 1.5km 범위의 산악지형을 구현하였고 적란운(악천후), 지대공미사일, 레이더를 모델링하여 장애물이 포함된 3차원 맵을 생성하였다. 이를 5개의 고도로 구분하여 Layered Map을 생성하여 Fig. 14와 같이 나타내었다. 경로 탐색 알고리즘은 Dijkstra를 사용하였다.

4.2 시뮬레이션 결과

4.1절에서 제시한 환경에서 두 가지 시뮬레이션을 실시하였으며 노드 수 감소 기법 적용 유무에 따라 연산 시간 향상 정도를 측정하였다. LVG와 Dijkstra를 적용한 시뮬레이션 결과는 Table 1, Table 2와 같으며, Fig. 15와 Fig. 16을 통해 Dijkstra를 활용한 경로가 장애물을 통과하지 않는 경로를 도출해냈음을 확인할 수 있다.

Table 1. Result of simulation

case 1	before node reduction	after node reduction
total boundary node	7,954 EA	415 EA
map generation time	about 28,230s	about 112s
path cost	115.7861 km	115.0112 km
path generation time	0.041s	0.037s

Table 2. Result of simulation

case 2	before node reduction	after node reduction
total boundary node	7,233 EA	372 EA
map generation time	about 25,400s	about 102s
path cost	99.4256 km	99.7474 km
path generation time	0.031s	0.021s

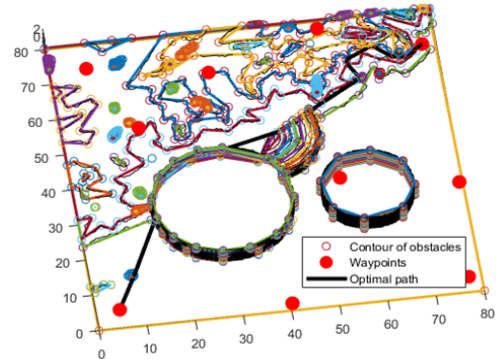


Fig. 15. Result of simulation case 1

Table 1을 참조하면 노드 수 감소 기법 적용 전에는 약 7900개의 노드가 생성되었고 인접행렬을 포함한 3차원 맵을 생성하는데 약 28,230s가 소요되었다. 이는 약 7.8시간에 달하는 수치이다. 노드 수 감소 기법을 적용한 후에는 530개의 노드가 생성되었으며 이는 적용 전 case에 대하여 약 93%의 감소율을 가진다. 인접행렬을 포함한 3차원 맵 생성에는 약 112s가 소요되었고 적용 전에 비해 연산 시간이 99.5% 감소된 것을 확인하였다. 생성된 경로의 cost (길이)는 두 case 모두 약 115km로 최적 경로 도출 성능에는 영향이 없는 것을 확인하였다.

Figure 16과 Table 2는 복잡한 산악지형에서 LVG가 경로 비용에 미치는 영향을 파악하기 위해 실시한 것이다. 산악지형에서의 경로 생성에 있어 적란운과 레이더 등은 경로 생성에 영향을 미치지 않아 Fig. 15에 있던 적란운과 레이더를 모델링한 부분은 삭제하였다. 복잡한 산악지형을 Convex Hull 기법을 통해 단순화하는 경우 장애물을 보다 크게 인식하여 이동이 가능한 경로가 줄어들어 지형 회피를 위한 최적 경로 대상이 줄어들었다고 생각할 수 있다. 그러나 Table 2에서 노드 감소 전후의 경로 비용을 비교하면 그 차이가 0.86%로 거의 차이가 나지 않음을 알 수 있다. 따라서 Convex Hull 기법을 통한 노드 수 감소는 경로 비용에 영향을 거의 미치지 않는다고 판단할 수 있다.

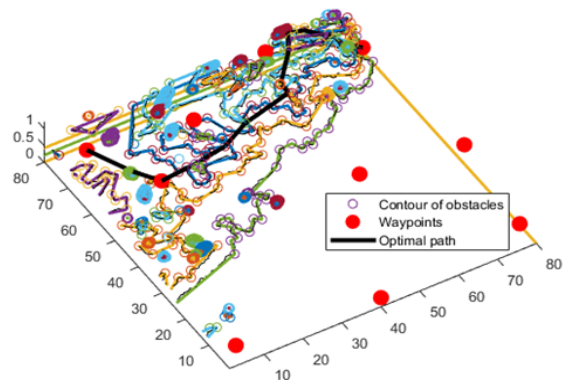


Fig. 16. Result of simulation case 2

V. 결 론

본 논문은 LVG를 활용하여 넓은 산악 지형에서 임무를 수행하는 무인기의 최적 경로 생성을 도출하기 위해, 실제와 유사한 3차원 시뮬레이션 환경을 구축하는 것이다. 또한 연산 속도 확보를 위해 LVG에 Convex Hull을 이용한 노드 수 감소 기법을 적용하는 방법을 제안하였다.

실제와 유사한 시뮬레이션 환경을 구축하기 위해 DTED 기반 3차원 지형을 적용하였으며, 해당 지형 맵에 악천후, 적 레이더 시설, 적 대공 시설을 가우시안 확률분포로 모델링하여 배치하였다. 3차원 경로 생성을 위한 노드와 인접행렬을 생성하는 방법으로 LVG를 선정하였으며, 이를 위해 지형 맵의 특정 고도에 대한 등고선 정보를 활용해 계층형 맵을 생성하였다.

그러나 실제 지형 환경은 불규칙적이고 복잡하기에 노드 수가 맵의 크기에 따라 기하급수적으로 증가하게 된다. 일반적인 LVG를 그대로 적용하면 연산 시간 또한 상승하기 때문에 실제 무인기 임무 환경에 적용하기는 불가능하다. 따라서 본 논문은 생성된 80 x 80 x 1.5km 크기의 두 case의 Layered Map에 Convex Hull을 적용하여 노드 수를 약 93%~94% 감소시켰으며, 연산 속도는 약 99.5% 감소시키는데 성공했다. 이 때 도출된 최적 경로의 cost는 두 case 모두 1% 미만의 경로 비용 차이를 가지며, 따라서 본 논문이 제안하는 방법이 경로 최적화의 성능에는 영향을 끼치지 않는 것을 확인하였다.

본 논문에서는 산악지형을 비롯하여 장애물들이 대체적으로 불록한 형태를 갖고 있어 Convex Hull 기법을 적용하기 수월하다. 그러나 오목한 형태의 장애물과 그에 인접한 장애물이 존재하는 경우 Convex Hull 기법을 적용하면 두 장애물이 하나로 합쳐져 경로를 도출했을 때 실제 이동 가능한 최단 거리와의 차이가 발생할 수 있다. 그러나 두 장애물이 하나로 합쳐지는 경우는 두 장애물 사이의 거리가 매우 가까운 경우에 발생하며, 인접한 두 장애물 사이를 비행할 경우에는 골바람 등이 기체의 안전에 영향을 미칠 수 있다. 이 때문에 본 논문에서는 특수한 상황에 대해서는 고려하지 않았으며, 특수한 지형에 Convex Hull 기법을 적용하고 최적 경로를 도출하는 것에 대해서는 추가적인 연구가 필요할 것으로 보인다.

본 논문에서 제시한 Convex Hull을 이용한 노드 수 감소 기법을 적용하여 연산 시간을 확보함으로써 실제 환경에서의 LVG 적용 가능성을 확인하였다.

후 기

본 연구는 방위사업청과 국방과학연구소의 지원을 받아 수행된 “동적 임무 환경에서 운용 가능한 무인

기용 실시간 임무 재계획 알고리즘 연구(No. UD170 098JD)” 과제 수행결과의 일부이다.

References

- 1) Sullivan, G. J., and Baker, R. L., “Efficient quadtree coding of images and video,” *IEEE Transactions on image processing*, Vol. 3, No. 3, 1994, pp. 327~331.
- 2) Amato, N. M., Bayazit, O. B., and Dale, L. K., “OBPRM: An obstacle-based PRM for 3D workspaces,” *Proceeding of International Workshop on Algorithmic Foundations of Robotics (WAFR)*, 1998.
- 3) Bhattacharya, P., and Gavrilova, M. L., “Roadmap-based path planning-using the voronoi diagram for a clearance-based shortest path,” *IEEE Robotics & Automation Magazine*, Vol. 15, No. 2, 2008, pp. 58~66.
- 4) 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.
- 5) Dijkstra, E. W., “A note on two problems in connexion with graphs,” *Numerische mathematik*, Vol. 1, No. 1, 1959, pp. 269~271.
- 6) LaValle, S. M., “Rapidly-exploring random trees: A new tool for path planning,” 1998.
- 7) Kim, H. I., Jung, D. M., Um, K. H., Cho, H. J., and Kim, J. T., “Efficient Path Finding in 3D Games by Using Visibility Tests,” *Journal of Korea Multimedia Society*, Vol. 9, No. 11, 2006, pp. 1483~1495.
- 8) Lu, Y., Gou, Z., Ye, Y., and Sheng, Q., “Three-dimensional visibility graph analysis and its application,” *Environment and Planning B: Urban Analytics and City Science*, Vol. 46, No. 5, 2019, pp. 948~962.
- 9) Bygi, M. N., and Ghodsi, M., “3D visibility graph,” *Computational Science and its Applications*, Kuala Lumpur, 2007.
- 10) Štola, J., “3D visibility representations of complete graphs,” *International Symposium on Graph Drawing*. Springer, Berlin, Heidelberg, 2003.
- 11) Bose, P., Everett, H., Fekete, S. P., Houle, M. E., Lubiw, A., Meijer, H., and Zelle, C., “A visibility representation for graphs in three dimensions,” *Graph Algorithms And Applications I*, 2002, pp. 103~118.