

군집 무인기들의 자가구성 통신중계 최적 배치
: 보로노이 다이어그램 기반 접근법장준희¹ · 김현우¹ · 박민수² · 최승환¹ · 송찬영¹ · 유혁² · 김덕수^{1,†}¹한양대학교 기계공학부²한국항공우주연구원Optimal Placement of UAVs for Self-Organizing Communication Relay:
Voronoi Diagram-Based MethodJunhee Jang¹, Hyunwoo Kim¹, Minsu Park², Seunghwan Choi¹, Chanyoung Song¹, Hyeok Yu² and
Deok-Soo Kim^{1,†}¹Department of Mechanical Engineering, Hanyang University²Korea Aerospace Research Institute

Abstract

The utilization of Unmanned Aerial Vehicles (UAVs) is expanding in various industries such as logistics, manufacturing, and transportation. However, to operate a large number of UAVs, it is imperative to first plan a secure and efficient self-configuring communication network for UAVs. In this study, we proposed a method for planning a secure and efficient UAV self-configuring communication network using Voronoi diagrams in the following three steps: 1) generating Voronoi diagrams using obstacles, 2) selecting obstacles to consider for path generation, and 3) planning the optimal path and outputting the path. The real-time feasibility of using the proposed method for planning optimal communication paths for a realistic number of UAVs was experimentally validated.

초 록

UAV (Unmanned Aerial Vehicle)는 물류, 산업, 운송 등 다양한 산업에서 그 활용처가 확대되고 있다. 하지만 다수의 UAV를 운용하기 위해서는 안전하고 효율적인 UAV 자가구성 통신망을 계획하는 것이 우선적으로 이루어져야 한다. 본 연구에서는 보로노이 다이어그램을 활용하여 안전하고 효율적인 UAV 자가구성 통신망을 계획하는 방법을 다음 3가지 단계로 제시한다: 1) 장애물들을 이용한 보로노이 다이어그램 생성, 2) 경로를 생성하기 위해 고려해야 할 장애물 선별, 3) 최적 경로 계획 및 경로 출력. 본 연구에서 제시한 방법은 실제 운용 시 UAV들의 최적 통신경로를 계획하는 것이 실시간으로 가능한 것을 실험을 통해 검증하였다.

Key Words : Voronoi diagram(보로노이 다이어그램), Unmanned Aerial Vehicle(무인기), Optimal placement (최적 배치), Communication relay(통신중계)

1. 서 론

4차 산업혁명이 진행된 이후, 산업에서 인터넷과 통

신은 중요한 역할을 하고 있다. 이에 따라, 새로운 지형을 탐사하거나 도시에서 네트워크 인프라를 구축할 때 지상국을 최적 위치에 설치하는 것이 중요한 과제로 부각되었다. 이러한 문제를 해결하기 위해 UAV(Unmanned Aerial Vehicle, 무인기)를 통한 자가구성 통신중계가 연구되고 있으며 본 논문에서는 Voronoi

Received: Nov. 24, 2023 Revised: May 12, 2024 Accepted: May 21, 2024

† Corresponding Author

Tel: ***-****-**** E-mail: dskim.hanyang@gmail.com

© The Society for Aerospace System Engineering

Diagram을 이용하여 UAV 통신 중계망을 구축하는 방법에 대해 다루고 있다.

UAV는 인간의 조종 없이 자동으로 또는 원격으로 조종되는 항공 장치를 의미한다. UAV는 환경 모니터링, 지형 조사, 재난 상황 대응 등 다양한 분야에서 유용하게 활용될 수 있고 물류, 산업, 운송 등 다양한 산업에서 통신 및 운송 수단 등으로 활용처가 확대되고 있는 것을 확인할 수 있다[1].

이와 같은 다양한 임무를 수행하기 위해서는 UAV의 자가구성 통신중계가 우선되어야 하며, 특히 복잡한 환경에서 중계 UAV를 활용한 자가구성 통신중계가 필요하다.

자가구성 통신중계란 지상국의 도움 없이 각 UAV끼리 상호작용하여 통신망을 구축하는 것으로 험지 탐사나 비상 상황에서 UAV의 지휘권을 유지하기 위해서 반드시 수행되어야 할 과제이며 이러한 문제를 해결하기 위한 방법으로 Voronoi Diagram이 많이 활용되고 있다. 논문[2~5]에서 Voronoi Diagram이 선행 연구로써 빈번히 활용되었으며, 인용횟수는 각각 330, 317, 584, 165회로 나타났다. 위와 같이 Voronoi Diagram은 공간을 효율적으로 분할하는 도구로써 자가구성 통신중계 문제를 빠른 시간 안에 정확히 풀 수 있는 수단으로 사용되고 있으며 향후 연구에서도 중요한 역할을 담당할 것이다.

2. 원천 기술

2.1 Voronoi Diagram

Voronoi Diagram은 공간 분할의 강력한 도구로서, 다양한 분야에서 활용되고 있는 혁신적인 개념이다. 이 도구는 기하학적인 원리를 바탕으로 특정 공간을 효율적으로 분할하며 이를 통해 다양한 기하학적 응용 분야에서 혁신적인 해결책을 제공한다.

Voronoi Diagram은 주어진 공간에 N 개의 객체가 있을 때 각 객체에서 가장 가까운 공간으로 분할하는 Tessellation이다[6,7]. $ci = (pi, ri)$ 가 중심이 pi 이고 반지름이 ri 인 원을 나타낸다고 가정할 때 $C = \{ci \mid i = 1, 2, \dots, n\}$ 은 원의 집합을 나타내며, $P = \{pi \mid i = 1, 2, \dots, n\}$ 은 원의 중심 집합을 나타낸다. 각 ci 에 대해, Voronoi Region

$R(C; ci) = \{p \in R^2 \mid d(p, pi) - ri < d(p, pj) - rj\}$ 로 정의된다. 여기서 $d(p, q)$ 는 두 점 p 와 q 사이의 유클리드 거리를 나타내며 공간은 N 개의 영역 $R(C; c1), R(C; c2), \dots, R(C; cn)$ 및 그 경계로 분할된다. 이 분할은 원 집합 C 에 대한 Voronoi Diagram이라고 불리며 $VD(C)$ 로 표시된다. 이때 $r1 = r2 = \dots = rn = 0$ 이라면 점 집합 P 에 대한 Voronoi Diagram이라 불리며 $VD(P)$ 라 부른다.

2.2 Yen's Algorithm

Yen's Algorithm은 Dijkstra Algorithm의 응용으로 최단 거리 알고리즘을 통해 얻은 경로 중 일부 Edge에 가중치를 변경하여 상위 k 개의 경로를 얻을 수 있도록 변형한 알고리즘이다. 위의 방법을 통해 사용자가 원하는 수의 경로들을 얻을 수 있다. 이후 출력된 경로들을 적은 장애물을 거치는 순으로 정렬하여 Pareto front의 형태로 저장한다. 이 방법은 기존에 1개의 해결책만을 제공하는 Dijkstra Algorithm에 비해 쉽게 다기준 문제를 적용할 수 있으며 다양한 비교가 가능하게 해준다[8].

3. 중계 무인기 최적 배치 알고리즘

본 논문에서는 지상국과 작업 영역 그리고 장애물에 대한 정보가 주어졌을 때 작업 영역 전체를 커버할 수 있는 중간 경유지(Intermediate Hub)를 찾고 지상국과 통신을 연결할 수 있는 최적 배치를 찾는 알고리즘을 제시한다. Fig. 1은 Mission Space에서 알고리즘에 사용된 각 명칭을 나타낸다.

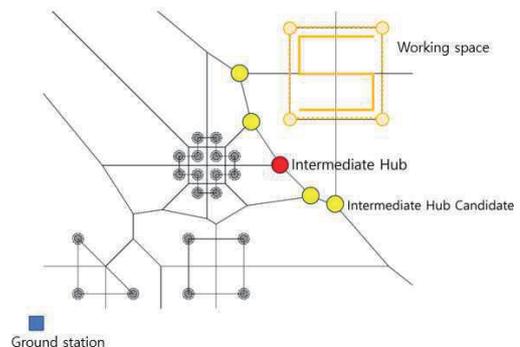


Fig. 1 Mission Space

본 알고리즘은 Fig. 2와 같이 크게 4가지로 이루어진다. 장애물과 지상국 등의 환경 정보가 주어지면 장애물들의 Voronoi Diagram을 계산하고 Voronoi Filter를 통해 탐색 영역을 축소시킴으로써 계산 시간을 획기적으로 줄일 수 있다. 이때 필터링된 영역 안에는 반드시 정답을 포함한다는 것을 수학적으로 보장한다 [9]. 이후 중계 UAV를 최적 배치하기 위한 Solution Pool Graph를 생성하고 출력된 경로에 중계 UAV를 배치하여 통신 중계망 계획을 완성한다.

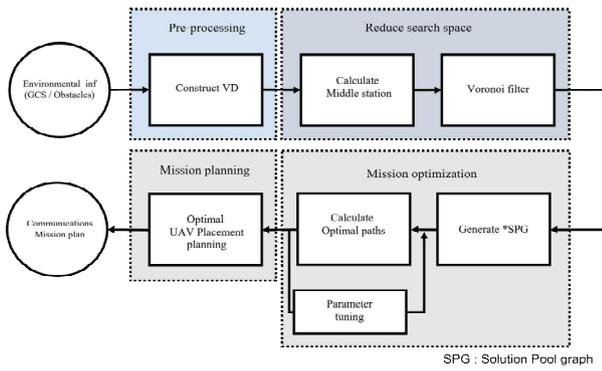


Fig. 2 Flow Chart

3.1. 전처리 : 장애물의 Voronoi Diagram 계산

본 논문에서는 Circle-set Voronoi Diagram을 이용하여 원의 형태를 가지는 장애물들의 Voronoi Diagram을 계산한다.

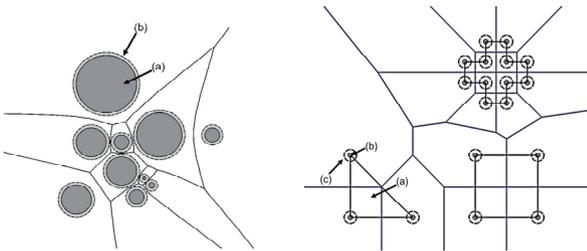


Fig. 3 Circle obstacles Fig. 4 Polygon obstacles

이때 장애물들은 각각 Circle Obstacles(Fig.3)와 Polygon Obstacles(Fig.4)로 구분된다. Circle Obstacles의 경우 모든 장애물(Fig.3-a)이 원으로 주어진 데이터를 의미하며 각 장애물을 Generator로 하는 Voronoi Diagram을 계산한다. 이때 장애물의 반지름은 장애물과 UAV의 안전거리(Fig.3-b)만큼 증가시켜 계산하

고 이후 계산된 모든 경로에 대해 안전거리를 보장하도록 한다. Polygon Obstacles의 경우 모든 장애물(Fig.4-a)이 다각형의 형태로 주어진 데이터를 의미하며 각 다각형의 꼭지점에 원(Fig.4-b)을 배치하여 각 원을 Generator로 한 Voronoi Diagram을 계산한다. 마찬가지로 이때 장애물의 반지름은 장애물과 UAV의 안전거리(Fig.4-c)만큼 증가시켜 계산한다.

3.2. 탐색 영역 축소

일반적으로 모든 장애물을 고려하여 Solution Pool Graph를 만들기 위해서는 $O(N^3)$ (N : 전체 장애물의 수)의 시간 복잡도가 요구된다. 따라서 고려해야 하는 장애물 수를 줄인다면 계산 시간을 상당 부분 줄일 수 있다. 본 논문에서 그 방법으로 Voronoi Filter를 제안하고 있으며 3단계로 나누어서 진행된다.

첫 번째로, Voronoi Diagram의 Vertex를 이용하여 장애물과 작업 영역 모두를 Generator로 가지는 Vertex를 탐색하고 그중 모든 작업 영역을 커버할 수 있는 Vertex를 탐색한다. 그리고 탐색된 후보들 중 지상국과 연결했을 때 짧은 경로를 제시하는 Vertex를 Intermediate Hub로 지정한다.

두 번째로, 지상국과 가장 가까운 Vertex를 찾고 이 Vertex와 Intermediate Hub를 각각 출발점과 도착점으로 지정하고 출발점과 도착점 사이의 Voronoi Edge를 타고 가는 최단 경로를 의미한다. 이 최단 경로(fig. 5)는 본 논문에서 V-chain이라고 하며 Voronoi Diagram의 Vertex로 이루어져 있다.

세 번째로, V-Chain을 이루는 각 Vertex를 생성하는데 관련된 Generator들을 전체 장애물에서 탐색한다. 이때 탐색된 Generator들을 V-Chain Defining Generators(fig. 6)라고 부르며 실제로 계산에 사용되는 탐색 영역이 된다. 이것을 계산하기 위한 시간 복잡도는 $O(N \log N)$ 이며 V-Chain Defining Generator들로 Solution Pool Graph를 생성하는 시간 복잡도는 $O(M^3)$ 이다. Voronoi Filter의 결과 얻어진 V-Chain Defining Generators의 수 M 은 전체 장애물 수 N 에 비해 현저히 작은 수이기 때문에 V-Chain Defining Generators로만 Solution Pool Graph를 생성할 경우 모든 장애물에 대해 계산하는 것 대비 계산 시간에서 상당 부분 이득을 얻을 수 있다.

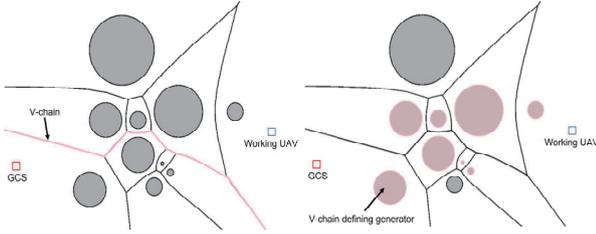


Fig. 5 V-chain Fig. 6 V-chain defining Generators

3.3. 통신 중계 미션 최적화

통신 임무를 최적화하기 위해서 전 단계에서 탐색된 Generators의 Solution Pool Graph를 생성한다. 최적 경로 생성을 위한 Solution Pool Graph를 만드는 과정은 다음과 같다. 먼저 Voronoi Filter를 통해 필터링된 Generators 중에 2개를 선정한다. 이때 2개의 원에서 만들 수 있는 접선은 Fig. 7의 접선 4가지와 같다.

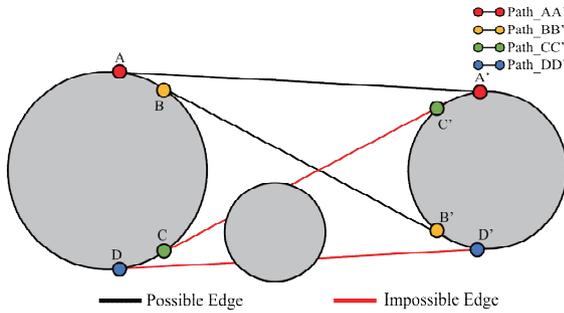


Fig. 7 Tangent Line between Two Obstacles

이 접선들은 Generators에서 Generators로 가는 최단 경로이며 실제로 UAV의 경로로 사용될 수 있는지 판단하기 위해서는 전체 Generator와 간섭이 있는지 검사해야 한다. 만약 전체 장애물에 대해서 간섭이 없다면 이 경로를 Solution Pool Graph에 포함시킨다. 이 과정을 모든 Generator 쌍에 대해 반복하면 Fig. 8과 같은 최종적인 Solution Pool Graph가 생성된다[10].

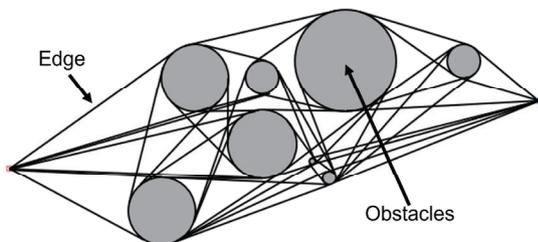


Fig. 8 Solution Pool Graph

본 논문에서는 단순히 지상국과 작업 영역을 연결하는 최단 경로가 아닌 중계 UAV의 수를 최소화해야 하므로 기존 최단 거리 알고리즘을 응용하여 다른 경로를 출력할 필요가 있다. 따라서 Edge의 가중치를 통신을 연결하기 위해 필요한 중계 UAV의 수로 단순화했으며 상위 k 개의 경로를 계산하여 비교할 수 있는 Yen's Algorithm을 채택했다.

3.4. 후처리

이전 단계의 결과로 출력된 경로는 직선과 호로 이루어져 있어 통신 가시선으로 사용하기에 적합하지 않다. 따라서 호로 이루어진 부분을 가공하여 직선화하고 최종적인 중계 UAV의 배치를 결정해야 한다. 이때 변환된 경로의 꼭지점에 중계 UAV가 배치되며 Edge의 길이가 통신 거리 이상인 경우 중간에 중계 UAV를 추가 배치한다.

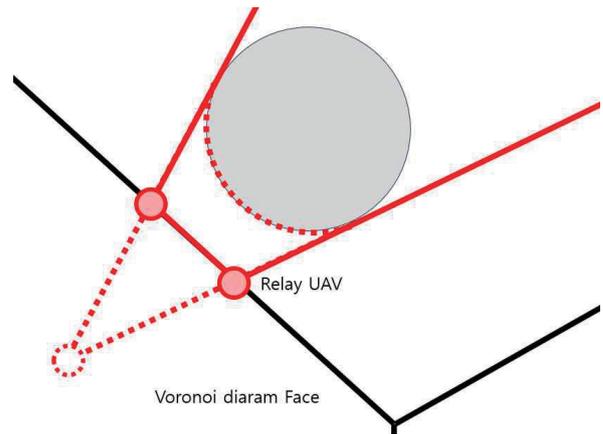


Fig. 9 Modified Path

또한 직선화 도중 접선의 교점이 장애물과 교차하거나 일정 거리 이상 길어지게 되어 Voronoi Diagram의 Face의 영역을 벗어나게 되면 fig. 9와 같이 Edge를 이용하여 경로를 수정하고 중계 UAV의 재배치한다. Voronoi Diagram의 특성상 Face 내부에는 다른 장애물이 존재하지 않는다는 것을 보장하기 때문에 수정된 배치 역시 안전한 경로임을 보장한다. 따라서 최종적으로 출력된 최적 배치는 모든 장애물에 대해 안전한 경로임을 보장하며 통신 가능한 경로이다.

4. 실험

4.1. 실험 조건 및 환경

본 논문에서 새롭게 제시한 알고리즘을 검증하기 위한 실험을 진행하였다. 일정한 공간 안에 장애물들의 수를 변화시켜 장애물의 수에 따른 계산 시간 비교를 진행하였다. 시작점과 도착점의 위치는 랜덤하게 지정하였으며 장애물의 위치 역시 랜덤하게 지정하였다. 이때 실험 장비의 사양은 CPU : AMD Ryzen 5 5600 X 6-Core / RAM : 16GB / GPU : RTX3070 / Windows 11이다.

4.2. 크기가 동일한 장애물

본 실험에서는 본 논문에서 제시한 방법이 장애물의 수에 따라 얼마나 계산 시간이 증가하는지 비교하는 실험을 진행하였고 실험 조건은 장애물 수를 100개부터 1000개까지 순차적으로 증가시켜 그에 따른 최적 배치 결과를 비교하였다.

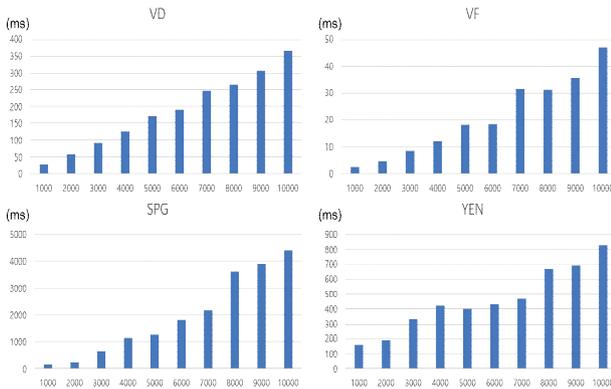


Fig. 10 Comparison of calculation time of #Obstacle

Fig.10은 각 단계별 계산 시간에 대한 그래프이며 장애물 1000개의 경우 Voronoi Diagram(VD)을 계산하는데 약 30ms, Voronoi Filter(VF)를 계산하는데 약 2ms, Solution Pool Graph(SPG)를 계산하는데 약 200ms, Yen's Algorithm(YEN)을 계산하는데 약 200ms의 시간이 소요되는 것을 확인할 수 있다.

Table 1 Comparison of calculation time of #Obstacle

#	VD (ms)	VF (ms)	SPG (ms)	YEN (ms)
100	2.6389	0.2227	0.7649	1.6893
200	5.3014	0.5058	4.0082	9.6745
300	8.4505	0.6478	12.98	22.6192
400	11.348	0.9718	14.3943	21.3532
500	13.4868	1.0812	27.5284	35.0595
1000	28.6983	2.4319	166.2073	160.5784
2000	56.9503	4.7682	263.2461	190.8752
3000	92.6034	8.5201	647.5396	329.3368
4000	125.7669	12.162	1144.221	423.8769
5000	170.5522	18.1709	1240.146	399.9002
6000	192.1613	18.4173	1817.425	431.6716
7000	246.5073	31.6734	2174.271	468.0636
8000	264.3781	31.2576	3616.811	665.606
9000	307.7969	35.6664	3889.568	691.7187
10000	367.1572	46.9974	4426.221	829.327

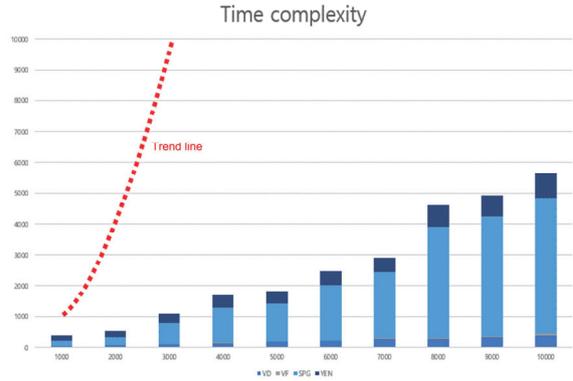


Fig. 11 Cumulative time complexity graph

Fig. 11를 보면 최종적인 계산 시간에 대한 누적 그래프를 볼 수 있으며 빨간 점선은 필터링을 하지 않았을 경우 걸리는 예상 시간이다. 장애물 1000개의 경우를 예를 들면 필터링을 하지 않았을 경우 전체 장애물에 대한 계산이 이루어져야 하므로 Solution Pool Graph를 생성하는데 걸리는 시간이 1000^3 이 된다. 하지만 필터링 후 추려진 장애물의 수는 평균적으로 10개 미만이기 때문에 $10^3(.avg)$ 이 되므로 위 그래프와 같은 차이를 보인다. 하지만 전체적인 누적 그래프를 보면 그림에도 Solution Pool Graph를 계산하는 시간이 많은 비중을 차지하는 것을 확인할 수 있으며 그 원인은 Solution Pool Graph를 생성할 때 모든 장애물에 대해 교차 검사가 이루어지고 있기 때문이다. 따라서 Convex hull filter 등을 이용하여 교차 검사를 할 때 사용되는 장애물의 수를 줄인다면 지금보다 더 빠른 시간 안에 계산이 가능할 것으로 보인다.

4.3. 크기가 랜덤한 장애물

위의 실험을 통해 크기가 같은 장애물에 대해 본 알고리즘이 효과적으로 작동하는 것을 확인할 수 있었으며 같은 조건으로 크기가 다른 장애물에 대해서도 효과적으로 사용할 수 있는지 확인하였다.

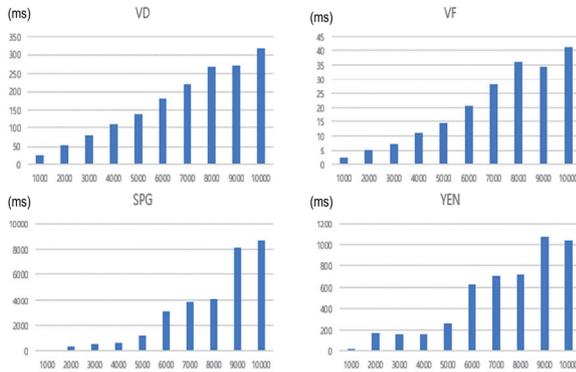


Fig. 12 Comparison of calculation time of #Obstacle (Random size)

Table 2 Comparison of calculation time of #Obstacle (Random size)

#	VD (ms)	VF (ms)	SPG (ms)	YEN (ms)
100	2.3056	0.2236	1.1142	2.7318
200	4.6231	0.4802	6.3804	10.4176
300	6.8434	0.6984	7.7924	10.1172
400	9.7011	0.8955	15.6667	19.808
500	11.774	1.2484	27.5919	26.5883
1000	24.5194	2.1946	66.6605	12.2328
2000	52.34	5.0782	352.3957	174.3392
3000	80.8663	7.1901	480.491	166.6503
4000	108.4364	10.958	639.1703	160.9541
5000	137.462	14.5103	1129.02	260.0568
6000	179.3178	20.7242	3129.698	635.4852
7000	246.5073	31.6734	2174.271	468.0636
8000	264.3781	31.2576	3616.811	665.606
9000	307.7969	35.6664	3889.568	691.7187
10000	367.1572	46.9974	4426.221	829.327

위의 실험을 통해 크기가 같은 장애물에 대해 본 알고리즘이 효과적으로 작동하는 것을 확인할 수 있었으며 같은 조건으로 크기가 다른 장애물에 대해서도 효과적으로 사용할 수 있는지 확인하였다. 크기가 랜덤한 장애물 역시 크기가 동일한 장애물과 유사한 결과가 나타났으며 다양한 환경에서 본 알고리즘이 효과적으로 작동한 다는 것을 확인하였다.

5. 결 론

초기 UAV는 단순 군사용 전략 무기였지만 현재는 우리의 일상에서 다양한 분야에서 사용되고 있다. 따라서 UAV가 다양한 변수에 대응할 필요가 있고 다수의 UAV를 사용하기 위한 통신망을 구성하는 것 역시 중요한 문제로 떠오르고 있다. 본 논문에서 제시한 새로운 알고리즘은 이러한 상황에서 최적의 대안을 제시할 수 있을 것이라 생각하며 다양한 분야에 활용도가 높을 것이라고 기대한다. 실제로 본 논문에서 전처리 과정에서 Voronoi Diagram을 계산함으로써 다양한 과정에서 Voronoi Diagram의 수학적 특성을 이용할 수 있었다. 그 결과 계산 시간이 획기적으로 줄었으며 반복적인 계산에서의 장점이 나타났다. 또한 차후 목적지가 움직이는 문제에 대해 Dynamic Voronoi Diagram을 접목시킨다면 실제 현장에서 UAV가 사용되는 것과 유사한 환경에서 다양하게 적용될 수 있는 알고리즘으로 발전할 가능성이 있다고 생각한다.

후 기

본 연구는 한양대학교의 분자기하학 및 분자기하운영체제 개발 연구단 연구비 지원(2017R1A3B1023591), 과학기술정보통신부의 재원으로 한국연구재단, 무인이동체원천기술개발사업단의 지원을 받아 무인이동체원천기술개발사업을 통해 수행되었음.(No. NRF-2020M3C1C1A02086425)

References

- [1] FAN Bangkui, LI Yun, ZHANG Ruiyu and FU Qiqi, "Review on the Technological Development and Application of UAV Systems." *Chinese Journal of Electronics*. Vol.29, No.2, p199-207, 2020
- [2] Pehlivanoglu, Y. Volkan. "A new vibrational genetic algorithm enhanced with a Voronoi Diagram for path planning of autonomous UAV." *Aerospace Science and Technology* 16.1, p47-55, 2012
- [3] Chandler, Phillip, Steven Rasmussen, and Meir

- Pachter. "UAV cooperative path planning." AIAA guidance, navigation, and control conference and exhibit, p14-17, August 2000.
- [4] Bortoff, Scott A. "Path planning for UAVs." Proceedings of the 2000 American Control Conference. ACC (IEEE Cat. No. 00CH36334). Vol. 1. No. 6, p28-30, August 2000.
- [5] Judd, Kevin, and Timothy McLain. "Spline based path planning for unmanned air vehicles." AIAA Guidance, Navigation, and Control Conference and Exhibit, p6-9 2001.
- [6] Deok-Soo Kim, Donguk Kim , Kokichi Sugihara. "Voronoi Diagram of a circle set from Voronoi Diagram of a point set: I." *Topology Computer Aided Geometric Design*,18-6 p541-62, 2001.
- [7] Deok-Soo Kim, Donguk Kim, Kokichi Sugihara. "Voronoi Diagram of a circle set from Voronoi Diagram of a point set:II." *Geometry. Computer Aided Geometric Design*.18-6, p:563-85, 2001
- [8] Ernesto Q.V. Martins, Marta M.B. Pascoal, "A new implementation of Yen's ranking loopless paths algorithm." *Quarterly Journal of the Belgian, French and Italian Operations Research Societies* volume. 1, p121-33, 2003
- [9] Deok-Soo Kim KY, Youngsong Cho, Donguk Kim & Chee Yap "Shortest Paths for Disc Obstacles." *Computational Science and Its Applications – ICCSA*, p62-70, 2004.
- [10] Luciano Blasi, Egidio D'Amato, Massimiliano Mattei and Immacolata Notaro. "Path Planning and Real-Time Collision Avoidance Based on the Essential Visibility Graph." *Applied Sciences* volume 10-16, 5613, 2020.