

## Energy-Efficient Drone Path Planning Method Considering Ground Surface Elevation Differences

Sooyoung Moon\*

\*Researcher, Convergence and Open Sharing System Immersive Media, Pai Chai University, Daejeon, Korea

### [Abstract]

Path planning methods play a major role (for example, collision avoidance and time/cost reduction) in unmanned aerial vehicles (UAV) based services. Path planning methods for ground robots cannot be directly applied to UAVs since they only consider 2D free space. A UAV should be able to fly over or fly around obstacles to reduce flight distance. The proposed path planning method is able to compute a 'fly-over' path with consideration of location, width and height of obstacles. The proposed path planning method is more efficient than existing Wavefront method when the obstacle height is less than half the width of the obstacle.

▶ **Key words:** Unmanned aerial vehicles (UAVs), Path planning, Collision avoidance, Wavefront algorithm, Software-in-the-loop simulation (SITL)

### [요 약]

드론 응용 서비스에서는 충돌 회피, 시간/비용 절감 등을 위한 경로 계획 방법이 핵심적인 역할을 담당한다. 기존 경로 계획 방법은 2차원 공간을 가정하며 지표면 높이 변화를 고려하지 않으므로 비행에 소요되는 에너지 관점에서 비효율적인 경로를 선택할 가능성이 존재한다. 제안 기법은 드론 비행 시 수직 기동에 소모되는 에너지량이 수평 운항에 비해 훨씬 높다는 점에 착안하여 지표면 높이 변화에 따른 에너지 모델을 고려한 경로 계획 방법을 제안한다. 제안된 경로 계획 방법은 지표면 높이 변화를 고려하여 단순한 거리 단축이 아닌 에너지 소모량 절감을 달성하는 경로 도출을 계산 목적. 제안 경로 계획 방법의 성능 평가를 위해 Python 기반 시뮬레이션을 수행하였으며 실험 결과 기존 경로 계획 방법인 Wavefront 알고리즘과 비교하여 제안 알고리즘은 드론 상승 거리 26.6%, 하강 거리 25.5%를 감소시켰으며, 평균 3.2%의 에너지 절감을 달성하였다. 본 연구 결과는 드론을 활용한 응용 분야에서 에너지 효율적 경로 계획에 적용될 수 있다.

▶ **주제어:** 무인기, 경로 계획, 충돌 회피, Wavefront 알고리즘, 시뮬레이션

• First Author: Sooyoung Moon, Corresponding Author: Sooyoung Moon  
\*Sooyoung Moon (symoon@pcu.ac.kr), Convergence and Open Sharing System Immersive Media, Pai Chai University  
• Received: 2025. 10. 02, Revised: 2025. 10. 13, Accepted: 2025. 10. 20.

## I. Introduction

드론은 항공 촬영, 측량, 시설물 점검, 택배 배송 등 다양한 분야에서 활용되고 있다[1-4]. 이러한 드론 기반 서비스의 성공적인 수행을 위해서는 효과적인 경로 계획 방법이 필수적이다. 드론 경로 계획의 주요 목표는 다음과 같다. 첫째, 가시권 밖 비행(Beyond Visual Line of Sight, BVLOS) 환경에서 조종자의 수동 제어 없이 자율 비행이 가능해야 한다. 둘째, 비행 중 지형지물 및 장애물과의 충돌을 회피해야 한다. 셋째, 비행 시간과 거리를 최소화하여 에너지 소비를 줄이고 임무 수행 범위를 확대함으로써 운영 비용을 절감해야 한다[5, 6].

지상 로봇을 위해 개발된 전통적인 경로 계획 방법, 예를 들어 Wavefront 알고리즘[8] 및 커버리지 경로 계획 방법[9, 10]은 2차원 공간을 가정한다. 이러한 알고리즘들은 너비 우선 탐색 메커니즘을 활용함으로써 최단 경로를 보장하는 장점이 있으나, 드론 운항에 적용할 때 근본적인 한계를 갖는다. 출발지와 목적지 사이에 장애물이 존재할 경우, 2차원 경로 계획 방법은 장애물의 측면으로 우회하는 경로만을 생성할 수 있다. 이는 드론이 가진 본질적인 3차원 이동 능력, 즉 장애물의 상공을 비행할 수 있는 능력을 활용하지 못하며, 결과적으로 상공 통과 경로에 비해 비행 거리가 증가하고 에너지 소비가 증가할 수 있다.

더욱 중요한 한계는 기존 방법들이 드론의 에너지 소비 특성을 고려하지 않는다는 점이다. 드론의 에너지 소비 모델에 관한 연구들[5-7]은 수평 운항 비행에 비해 수직 기동이 단위 거리당 에너지 소비가 훨씬 높음을 실증하였다. 특히 Rodrigues 등[7]은 188회의 배송 드론 비행 데이터를 분석하여 이를 검증하였으며, Zhang 등[4]은 다양한 배송 드론 에너지 소비 모델을 비교 평가하였다. 또한 수직 이동의 에너지 모델은 비대칭적 특성을 보이는데, 드론이 하강 비행을 할 때에 비해 상승 비행을 할 때 더 많은 에너지를 소비한다[5,6]. 그러나 거리만을 고려하는 기존 경로 계획 방법들은 이러한 에너지 소비 모델을 고려하지 않는다는 한계점이 있다.

실제 운용 환경에서 지표면 고도는 언덕, 계곡, 경사면과 같은 지형적 특성으로 인해 크게 변화한다. 각 위치에서 요구되는 최소 안전 비행고도는 지표면 높이에 따라 변화하게 된다. 따라서 고도가 높은 지형을 통과하는 경로는 더 큰 고도 변화를 필요로 하며, 이는 수직 기동에 따른 높은 에너지 비용을 발생시킨다. 이러한 지형-에너지 결합은 근본적인 상충 관계를 만든다. 고지대 지형을 우회하면 고도 변화는 줄어들지만 수평 이동 거리가 증가하고, 반대로

고지대 상공을 비행하면 수평 거리는 최소화되지만 에너지 집약적인 상승 기동이 필요하다. 기존 경로 계획 방법들은 이러한 상충 관계를 체계적으로 평가하고 최적화하는 메커니즘이 부족하다. 본 논문에서는 Wavefront 알고리즘을 확장하여 지형 고도와 에너지 소비를 동시에 고려하는 경로 계획 방법을 제안한다. 핵심 아이디어는 비용 척도를 단순한 셀 개수(거리)에서 실제 에너지 소비량으로 재정의하는 것이다. 제안 방법은 수평 비행, 상승, 하강에 대한 실증 기반 에너지 모델[5-7]을 적용하여 거리 최소화 대신 에너지 최소화를 직접적으로 달성한다. 구체적으로, 제안 알고리즘은 맵 데이터를 기반으로 경로 탐색 과정에서 인접 셀로 이동할 때의 비용을 수평 이동 거리와 고도 변화량을 모두 반영하여 계산된다.

본 논문의 주요 기여점은 기존 셀 개수 기반 비용 척도를 개선하여 비행 고도 변화에 따른 비용 모델을 제안한 것이다. 이를 통해 실제적 환경, 즉 지표면 높이 변화가 존재하는 환경에서 드론의 에너지 효율적 경로 계획을 가능케 한다는 점에서 의의가 있다. 제안 방법은 드론 배송, 정밀 농업, 시설물 점검 등 배터리 용량이 제한적인 실용적 드론 응용 분야에서 임무 수행 범위를 확대하고 운영 비용을 절감하는 데 기여할 수 있다.

본 논문의 구성은 다음과 같다. 2장에서는 기존 경로 계획 방법인 Wavefront 알고리즘과 드론 에너지 모델에 관한 관련 연구를 검토한다. 3장에서는 지형 인식 및 에너지 기반 경로 계획 알고리즘을 상세히 설명한다. 4장에서는 기존 경로 계획 방법과 제안 경로 계획 방법 간 성능을 비교 분석한다. 5장에서는 연구 결과를 요약하고 향후 연구 방향을 논의한다.

## II. Background

### 1. Wavefront algorithm

Wavefront 알고리즘 [8]은 A. Zelinsky 에 의해 제안되었으며 무인기가 임무를 수행하는 영역의 맵 데이터가 존재하고 모든 장애물의 위치가 고정이라는 가정 하에 출발지에서 목적지까지 최단 경로 또는 커버리지 경로 (coverage path) [9, 10] 계산이 가능하다. Wavefront 알고리즘은 다음과 같이 3 단계로 진행된다.

- 1) 전체 영역을 격자로 분할 및 출발지와 목적지 식별
- 2) 목적지에서 너비 우선 탐색 (BFS) 알고리즘 수행
- 3) 출발지 셀에서 목적지 셀까지 경로 계산 -

최단 경로 또는 커버리지 경로 계산

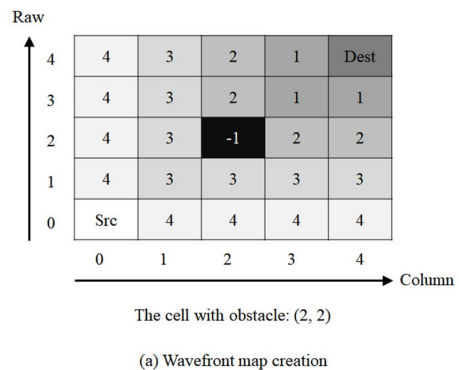
2)에서, Wavefront map 구성 영역을 구성하는 각 셀은 양의 정수인 레이블 (Label)을 할당 받으며 이는 해당 셀에서부터 목적지까지 이동하기 위해 거쳐야 하는 셀의 최소 개수를 의미한다. 그러므로 각 셀의 레이블은 탐색 트리 상에서 부모 셀의 레이블 값에 1을 더한 값으로 결정된다. Table 1은 Wavefront 경로 계획 알고리즘의 동작 과정을 나타낸다.

Table 1. Wavefront Path planning algorithm

<p>Inputs:</p> <ol style="list-style-type: none"> <li>1) <math>M</math>: Map data</li> <li>2) <math>S</math>: Source,</li> <li>3) <math>D</math>: Destination,</li> <li>4) <math>O</math>: Obstacle data</li> </ol>
<p>Output:</p> <p><math>P</math>: A path from source <math>S</math> to destination <math>D</math>  <math>P = \langle \text{Cell}_1(=S), \text{Cell}_2, \dots, \text{Cell}_N(=D) \rangle</math></p>
<p><math>WaveFront(M, S, D, O)</math>:</p> <ol style="list-style-type: none"> <li>1. Wavefront Map Creation             <ol style="list-style-type: none"> <li>① Divide <math>M</math> into a grid of cells of size <math>m \times n</math>, where each cell is denoted as <math>(i, j)</math></li> <li>② Determine the grid cells corresponding to <math>S</math> (source) and <math>D</math> (destination)</li> <li>③ Mark obstacle cells based on <math>O</math> (obstacle data) by assigning them a label of -1</li> <li>④ Assign the destination cell a label of 0</li> <li>⑤ Perform a Breadth-First Search (BFS) starting from the destination cell to propagate distance labels                     <ol style="list-style-type: none"> <li>i) Initialize an empty queue <math>Q</math></li> <li>ii) Mark all non-obstacle cells as 'undiscovered'</li> <li>iii) Set label[<math>D</math>] = 0, Enqueue <math>D</math> into <math>Q</math></li> <li>iv) While <math>Q</math> is not empty:                             <p>Dequeue the front cell <math>c</math> from <math>Q</math></p> <p>For each neighbor <math>n</math> of <math>c</math>:</p> <p>If <math>n</math> is 'undiscovered' and not an obstacle:</p> <p>Set label[<math>n</math>] = label[<math>c</math>] + 1</p> <p>Mark <math>n</math> as 'discovered'</p> <p>Enqueue <math>n</math> into <math>Q</math></p> </li> </ol> </li> </ol> </li> </ol> <li>2. Path Computation</li>

- ① Initialize path  $P$  with the source cell:  
 $P \leftarrow [S]$
  - ② Set current  $\leftarrow S$
  - ③ Repeat until current  $\neq D$ 
    - Among all neighbors (w/o obstacles) of current:
    - Select the neighbor  $n$  with the smallest positive label
    - Append  $n$  to path  $P$
    - Set current  $\leftarrow n$
3. Return the computed path  $P$

Table 1에서 Wavefront 알고리즘의 입력은 1) 맵 데이터 ( $M$ ), 2) 출발지 위치 ( $S$ ), 3) 목적지 위치 ( $D$ ), 그리고 맵 영역에 속한 장애물의 위치 정보 ( $O$ )이다. 또한 해당 알고리즘의 출력은 출발지  $S$ 에서 목적지  $D$ 까지의 이동 경로 ( $P$ )이다. Wavefront 알고리즘은 크게 1) Wavefront 맵 생성과 2) 경로 계산의 순서로 진행된다. Wavefront 맵 생성 단계에서는 먼저 맵 영역을 격자 형태로 분할하고 출발지 셀과 목적지 셀을 계산한다. 이후 장애물 데이터로부터 각 장애물의 위치에 해당하는 셀의 레이블을 -1으로 설정한다. 이는 해당 셀로 이동할 수 없음을 나타내기 위함이다. 또한 목적지 셀의 레이블을 0으로 할당한 후 너비 우선탐색을 통해 목적지 셀에서부터 모든 셀의 레이블을 계산한다. 맵 생성과 그에 따른 레이블 계산이 완료되면 출발지에서 목적지까지의 경로를 계산한다. 먼저 출발지 셀만 포함하는 임시 경로 변수를 생성한 후, 현재 셀의 이웃 셀들 중 레이블 값이 가장 작은 셀을 선택하고 이를 목적지 셀을 찾을 때까지 반복하면 출발지에서 목적지까지의 최단 경로가 도출된다. 그림 1은 Wavefront 알고리즘에 의한 최단 경로 계산 과정의 예를 나타낸다.



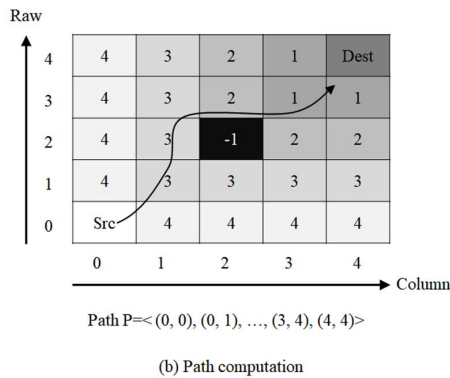


Fig. 1 Path Computation based on Wavefront algorithm

Fig. 1. (a) 에서 Wavefront map 계산을 위해 영역이 5-by-5 격자로 분할된 후 출발지 셀의 위치 (0, 0), 목적지 셀의 위치 (4,4) 가 각각 계산된다. 또한 목적지 셀에 레이블 값 (0) 을 할당하고 너비 우선 탐색을 통해 각 셀의 레이블 값을 계산한다. 장애물이 위치한 셀의 경우 -1 값이 할당된다.

Fig. 1. (b)에서는 변수 P에 출발지 셀 정보를 추가한 후 출발지 셀의 이웃 셀들 중 가장 레이블 값이 작은 셀 (1, 1) 을 변수 P에 추가한다. 변수 P에 목적지 셀 정보가 추가될 때까지 위 과정을 반복한다. 경로 계산이 완료되면 해당 경로 정보가 반환된다.

Wavefront 알고리즘은 너비 우선 탐색 알고리즘을 기반으로 하므로 출발지에서 목적지까지 최단 경로가 존재한다면 이를 반드시 찾을 수 있다는 장점이 있는 반면 영역의 크기가 증가함에 따라 시간 복잡도와 공간 복잡도가 지수적으로 증가한다는 단점을 갖는다. 또한 2차원 공간을 가정하므로 출발지에서 목적지 사이에 장애물이 존재할 경우 항상 해당 장애물의 측면으로 우회하는 경로만을 선택하게 되어 경우에 따라 해당 장애물의 상공을 통과하는 경로에 비해 이동 거리 면에서 상대적으로 비효율적인 경로를 선택하게 될 가능성이 존재한다.

2. Energy model

드론의 에너지 소비는 비행 단계(수평 비행, 상승, 하강, 호버링)에 따라 크게 다르다. 최근 실증 연구들은 각 비행 단계의 에너지 소비 특성을 정량적으로 규명하였다. 드론의 총 에너지 소비는 수평 이동과 수직 이동의 합으로 표현된다.

$$E_{total} = E_h + E_v \tag{1}$$

( $E_h$ : 수평 비행 에너지,  
 $E_v$ : 수직 이동 에너지)

수평 이동 에너지 모델

수평 이동 에너지는 드론이 지면과 평행하게 비행함에 따른 소모 에너지를 의미하며, Rodrigues 등[7]과 Zhang 등[4]의 실증 연구에 의하면 다음과 같이 표현된다.

$$E_h \approx k_h \times d \tag{2}$$

( $k_h$ : 수평 비행 에너지 계수 (약 150~200 J/m))

여기서  $k_h$ 는 드론 기체 특성에 따른 수평 비행 에너지 계수이며,  $d_h$ 는 수평 방향 이동 거리를 의미한다. Rodrigues 등[7]은 DJI Matrice 600 Pro 드론(총 중량 약 10kg)을 사용한 188회의 실제 배송 비행 데이터 분석을 통해  $k_h = 180$  J/m의 값을 도출하였다. Li 등[5]은 다양한 회전익 드론에 대한 실험을 통해  $k_h$ 가 드론의 질량과 공기역학적 특성에 따라 150-200 J/m 범위에 분포함을 확인하였다. 즉, 수평 이동 에너지는 수평 방향 이동 거리가 증가할수록 선형적으로 증가한다.

수직 이동 에너지 모델

수직 이동 에너지는 드론이 상승, 하강 등 지면을 기준으로 수직 운동을 할 때 소모되는 에너지를 의미한다. Abeywickrama 등[6]과 Li 등[5]의 연구에 의하면 드론의 수직 이동 에너지는 다음과 같이 표현된다.

$$E_v = \begin{cases} (k_c \cdot mg \cdot \Delta h) & \text{if } \Delta h > 0 (\text{상승}) \\ -k_d \cdot mg \cdot |\Delta h| & \text{if } \Delta h < 0 (\text{하강}) \end{cases} \tag{3}$$

( $m$ : 기체 질량,  
 $g$ : 중력 가속도,

$\Delta h$ : 이동 전/후 지면 높이 차 (>0: 상승, <0: 하강))

여기서  $m$ 은 기체 질량(kg),  $g$ 는 중력 가속도(9.81 m/s<sup>2</sup>)이다.  $\Delta h$ 는 이동 전후의 고도 차이(m)를 의미하며,  $\Delta h > 0$ 일 때는 상승 운동,  $\Delta h < 0$ 일 때는 하강 운동을 나타낸다.  $k_c$ 과  $k_d$ 는 각각 상승(climb)과 하강(descend) 시 에너지 효율을 나타내는 무차원 계수이다. Rodrigues 등[7]의 실증 연구에서 188회의 비행 데이터 분석 결과, 상승 시  $k_c=1.8$ , 하강 시  $k_d=0.5$ 의 값이 측정되었다. 이는 드론이 상승할 때 중력에 대항한 추력을 발생시켜야 하므로 이론적 에너지( $m \cdot g \cdot \Delta h$ ) 보다 약 80% 더 많은 에너지가 소모되며, 하강 시에는 중력을 이용할 수 있어 이론값의 약 50%만 소모됨을 의미한다. Li 등[5]은 다양한 회전익 드론에 대한 실험에서  $k_c$ 가 1.5-2.0 범위,  $k_d$ 가 0.4-0.6 범위에 분포함을 확인하였다.

에너지 소비 특성

기존 연구에서 드론 비행에 소모되는 에너지 소비 특성을 요약하면 다음과 같다.

1. 수평 vs 수직 비중 [11]: Rodrigues 등[7]은 일반적 배송 임무에서 수평 비행이 전체 에너지의 87.63%를 차지하며, 수직 기동(상승/하강)이 12.37%를 차지함을 실증함
2. 단위 거리당 에너지 집약도: Rodrigues 등[7]에 따르면 수직 이동이 수평 이동보다 단위 거리 당 2-3배 높음
3. 상승-하강 비대칭성: 상승 시 에너지 소비가 하강 시보다 약 3.6배 높음[6,7]. 즉  $|E_{climb}| \approx 3.6 \times |E_{descent}|$
4. 질량 의존성 Abeywickrama 등[6]과 Zhang 등[4]의 연구에 따르면, 페이로드 1kg 증가 시 전체 에너지 소비가 15-20% 증가하며 이는 식 (3)에서 수직 이동 에너지가 질량  $m$ 에 비례함을 통해 설명됨

### III. The Proposed Scheme

#### 1. Problem definition

##### 1.1 Definitions and Assumptions

본문에서 사용된 주요 용어들은 다음과 같다.

- 영역 (Area) : 드론이 임무를 수행하는 지리적 범위
- 셀 (Cell) : 영역을 격자(grid) 형태로 나눈 기본 단위
- 출발지 셀 (Start Cell) : 드론이 비행을 시작하는 셀
- 목적지 셀 (Destination Cell) : 드론이 이동하는 목적지 셀
- 경로 (Path) : 출발지에서 목적지까지 이동하기 위한 좌표들의 목록
- 최대 비행 고도 (Maximum Flight Altitude): 드론이 비행 가능한 최대 고도 (지표면 기준)
- 최소 비행 고도 (Minimum Safety Altitude): 드론이 특정 셀의 상공으로 통과할 경우, 충돌 방지를 위해 유지해야 하는 최소 고도
- 웨이포인트 (Waypoint) : 드론이 이동할 때 거치는 경로 내 각 좌표 (x, y, z)
- 비용 (Cost): 특정 셀을 거치는 경로의 비용

본 논문의 주요 가정사항은 다음과 같다.

1. 사전에 드론 임무 수행 영역의 맵 데이터가 제공된다
2. 맵 데이터에는 지표면 높이 정보가 포함된다
3. 드론 충돌 방지를 위한 최대/최소 비행고도가 존재한다

#### 2. Proposed method

##### 2.1 Limitation of Existing Methods

Wavefront 알고리즘은 2차원 공간을 가정하므로 드론이 임무를 수행하는 영역 내 지표면 높이 변화를 고려하지

않는다. 드론은 3차원 공간에서 이동하며 2.2 Energy Model에 기술된 바와 같이 수평, 상승, 하강 시 단위 거리당 에너지 소모량에 차이를 보인다. 그러나 Wavefront와 같이 2차원 공간을 가정하는 경로 계획 알고리즘들은 이러한 에너지 모델에 대한 고려가 없어 3차원 공간에서 에너지 효율적 이동 경로를 계산하는데 한계점을 갖는다. 제안 알고리즘은 드론의 수평, 상승, 하강 기동에 따른 에너지 모델을 고려한 에너지 효율적 경로 계산이 가능하다.

#### 2.2 Proposed Method

Table 2는 제안된 경로 계획 알고리즘을 나타낸다.

Table 2. Terrain-aware path planning

Inputs: 1) $M$ : Map data (including terrain elevation) 2) $S$ : A source cell, 3) $D$ : A destination cell,  Params: • $MFA$ : Maximum Flight Altitude • $MSA$ : Minimum Safe Altitude (above ground) • $TE_{Max}$ : Maximum Terrain Elevation • $TE_{Min}$ : Minimum Terrain Elevation
Output: $P$ : A path from source $S$ to destination $D$ $P = \langle \text{Waypoint}_1(\text{in } S), \text{Waypoint}_2, \dots, \text{Waypoint}(\text{in } D) \rangle$
Terrain-awarePathPlanning( $M, S, D$ )  1. Wavefront Map Creation  ① Divide $M$ into a grid of cells of size $m \times n$ , where each cell is denoted as $(i, j)$  ② Perform a Breadth-First Search (BFS) starting from the destination cell $D$ to propagate cost values  i) Initialize an empty queue $Q$ ii) Mark every cell in the area as 'undiscovered' iii) Set $\text{cost}[D] = 0$ , Enqueue $D$ into $Q$ iv) While $Q$ is not empty: Dequeue the front cell $c$ from $Q$ For each neighbor $n$ of $c$ : if $n$ is 'undiscovered' if $(\text{terrainHeight}[n] + MSA) \geq MFA$ mark $n$ as 'unreachable' else

```

hDistance =
    sqrt(c.x - n.x)^2 + (c.y - n.y)^2

vDistance = terrainHeight[n] - terrainHeight[c]

If (vDistance > 0): // climbing
    E_segment = E_model.h(hDistance) +
                E_model.c(vDistance)
else: // Descending
    E_segment = E_model.h(hDistance) +
                E_model.d(|vDistance|)
cost[n] = cost[c] + E_segment
flightAltitude[n] = terrainHeight[n] + MSA
Mark n as 'discovered'
Enqueue n into Q
    
```

2. Path computation

- ① Initialize path P with the source cell S with its flight altitude  
 $P \leftarrow [(S, flightAltitude[S])]$
- ② Set current  $\leftarrow S$
- ③ Repeat until current  $\neq D$   
 Among all neighbors (not 'unreachable') of current:  
 Select the neighbor n\* with the smallest cost  
 Append (n\*, flightAltitude[n\*]) to P  
 Set current  $\leftarrow n^*$
- ④ Return P

Energy Model Functions:

E\_model.h(d):  
 // Horizontal flight energy  
 Return  $k_h * d$

E\_model.c(h):  
 // Vertical flight energy (Ascending)  
 Return  $k_c * m * g * h$   
 (n: motor efficiency (ex. 0.65))

E\_model.d(h):  
 // Vertical flight energy (Descending)  
 Return  $k_d * m * g * h$

Table 2는 제안된 지형 및 에너지 인식 경로 계획 알고리즘 (Terrain-aware Energy Efficient Method; 이하 TEEM) 을 나타낸다. 제안된 알고리즘의 입력은 1) 맵 데이터 ( $M$ ), 2) 출발지 셀 ( $S$ ), 3) 목적지 셀 ( $D$ )이며 출력은 웨이포인트 (셀 번호, 비행 고도)들로 구성된 출발지-목적지 사이 경로 ( $P$ )이다.

주요 파라미터들은 최대 비행 고도 (Maximum Flight

Altitude; 이하 MFA), 지표면 위 최소 안전 고도 (Minimum Safe Altitude; 이하 MSA)를 포함한다. TEEM 알고리즘은 크게 1) Wavefront 맵 생성과 2) 경로 계산의 두 단계로 진행된다. Wavefront 맵 생성 단계에서는 먼저 맵을 이산적인 격자 (Grid)로 분할한다. 격자의 해상도(셀 크기)는 경로 계획의 정확도와 계산 복잡도 간의 트레이드-오프 관계를 결정한다. 다음으로 목적지 셀에서부터 너비 우선 탐색을 통해 각 셀까지의 최소 에너지 비용을 계산한다. 탐색 과정에서 각 셀의 상태는 초기 'Undiscovered' 상태에서 탐색이 진행됨에 따라 'Discovered' 또는 'Unreachable' 상태로 변화하게 된다. 각 셀이 가질 수 있는 상태와 그 의미는 다음과 같다.

- Undiscovered: 아직 탐색되지 않는 셀
- Discovered: 탐색과 비용 계산이 완료된 셀
- Unreachable: 최대 비행 고도 제한으로 인해 상공으로 통과가 불가능한 셀

각 셀의 도달 가능성 판단은 Table 3와 같이 이뤄진다.

Table 3. Reachability check for a cell

```

if(terrainHeight[n] + MSA) ≥ MFA :
    n을 'unreachable'로 표시
    
```

즉, 최대 비행 고도 제한으로 인해 해당 셀을 통과할 수 없다는 의미이다. 이는 안전 비행을 위해 유지해야 할 최소 비행고도가 존재하기 때문이다. 만약 최대 비행 고도가 50m이고 해당 셀의 지표면 높이가 45m이며 최소 비행 고도가 5m인 경우, 해당 셀은 최대 비행 고도 제한으로 접근 불가능한 셀로 간주된다.

만약 해당 셀이 도달 가능한 셀일 경우, 현재 셀에서 해당 셀 사이 구간 통과에 소모되는 에너지 비용을 계산하기 위해 Table 4와 같이 해당 구간의 수평·수직 거리를 계산한다.

Table 4. Computation of horizontal/vertical distance for a segment (from current cell to a neighbor)

```

else :
    hDistance = [수평거리 계산]
    vDistance = terrainHeight[n] - terrainHeight[c]
    
```

위 수식에서 hDistance, vDistance는 각각 현재 셀과 이웃 셀 간의 수평 및 수직 거리이다. 본 논문에서 드론은 지면으로부터 최소 비행 고도를 유지한다고 가정하므로 vDistance는 두 셀 간의 지표면 높이 차와 같다. 즉,

vDistance 가 양수인 경우 상승 운동, 반대로 음수인 경우 하강 운동에 해당한다. 수평 거리와 수직 거리로부터 드론의 이동에 필요한 에너지를 계산하는 과정은 Table 5와 같다.

Table 5. Computation of energy consumption for a segment (from current cell to a neighbor)

$E_{model.h}(d):$ Return $k_h * d$
$E_{model.c}(h):$ Return $k_c * m * g * h$
$E_{model.d}(h):$ Return $k_d * m * g * h$

즉 수평 이동의 경우 에너지 계수  $k_h$ 에 수평 이동 거리  $d$ 를 곱한 값 만큼 에너지가 소모되며, 상승 또는 하강의 경우 에너지 계수  $k_c$  (상승) 또는  $k_d$  (하강) 에 질량  $m$ , 중력 가속도  $g$ , 상승/하강 거리를 곱한 값 만큼 에너지가 소모된다. 각 셀의 비용 (cost)은 너비 우선 트리 상의 부모 셀이 가진 비용에 수평 이동 에너지와 상승/하강 에너지를 더한 값으로 결정된다( $cost[n] = cost[c] + E_{segment}$ ). 너비 우선 탐색이 종료되면 목적지 셀과 연결된 모든 셀들의 비용 (cost) 계산과 함께 접근 가능/불가 셀 식별이 완료된다.

경로 계산 단계에서는 출발지 셀에서부터 시작하여 목적지 셀에 도달할 때까지 현재 셀의 주변 셀들 중 가장 비용이 작은 셀을 선택하여 경로를 구성해 나간다. 목적지 셀이 경로에 추가되면 탐색을 중단하고 해당 경로를 반환한다.

## IV. Simulation result

### 4.1 Simulation Environment Variables

제안된 TEEM 알고리즘의 성능 검증을 위해 Python 기반 시뮬레이션을 수행하였다. 관련 내용은 Table 6와 같다.

Table 6. Simulation Parameters

Parameters	Value	Description
Area size	50m × 50m	Mission area
Cell Size	1m × 1m	Grid resolution
Start Cell	(0, 0)	-
Dest. Position	(49, 49)	-
MFA	100m	Maximum Flight Altitude
MSA	5m	Minimum Safe Altitude
kh	180 J/m	Horizontal energy coefficient
kc	1.8	Climbing energy coefficient

kd	0.5	Descending energy coefficient
Drone Mass	10 kg	Total mass including payload
Terrain Std Dev	1-10m	Gaussian distribution parameter
Iterations	10	Per standard deviation

드론 임무수행 영역의 전체 크기는 50m × 50m 이며, 각 셀의 크기는 1m × 1m로 설정하였다. 따라서 영역 내 전체 셀의 갯수는 50 \* 50 = 2500이다. 실제 드론 임무수행시 출발지와 목적지는 임의 위치로 지정될 수 있으나, 실험의 일관성을 위해 출발지 셀은 (0, 0), 목적지 셀은 (49, 49)로 지정하였다. 지형 높이는 가우시안 분포를 따르도록 생성하였으며, 표준편차를 1m부터 10m까지 변화시키며 총 100회(10개 표준편차 × 10회 반복)의 시뮬레이션을 수행하였다. 각 시뮬레이션에서 Wavefront 알고리즘과 TEEM 알고리즘을 동일한 지형 조건에 적용하여 공정한 비교를 수행하였다.

### 4.2 Comparison of Average Moving Distance

Table 7은 모든 시뮬레이션 결과에 대한 평균 이동 거리를 비교한 것이다.

Table 7. Average Distance Comparison

Metric	Wavefront	TEEM	Improvement (%)
Horizontal Distance (m)	69.30	69.24	0.1%
Climbing Distance (m)	7.53	5.53	26.6%
Descending Distance (m)	7.76	5.78	25.5%
Total Distance (m)	72.41	71.09	1.8%

분석 결과, TEEM 알고리즘은 수평 이동 거리에서는 Wavefront와 거의 동일한 성능을 보였으나(0.1% 개선), 수직 이동 거리에서 유의미한 차이를 보였다. 상승 거리는 26.6%, 하강 거리는 25.5% 감소하였으며, 이는 TEEM이 지형의 고도 변화를 고려하여 불필요한 수직 기동을 최소화한 결과이다. 총 이동 거리는 1.8% 감소에 그쳤으나, 이는 수평 이동 거리가 전체의 대부분을 차지하기 때문이다. 2.2절에서 언급한 바와 같이 Rodrigues 등[7]의 연구에서도 일반적 배송 임무에서 수평 비행이 전체 이동의 87.63%를 차지한다고 보고된 바 있으며, 본 시뮬레이션 결과도 이와 일치한다(수평 거리 69.30m / 총 거리 72.41m ≈ 95.7%).

수직 이동 거리의 감소는 에너지 소비 관점에서 더욱 중요한 의미를 갖는다. 2.2절에서 설명한 바와 같이 수직 이동의 단위 거리당 에너지 소비는 수평 이동보다 훨씬 높기 때문이다.

4.3 Analysis on Energy Saving

Fig. 2는 지형 높이의 표준편차 변화에 따른 두 알고리즘의 에너지 소비량을 비교한 것이다.

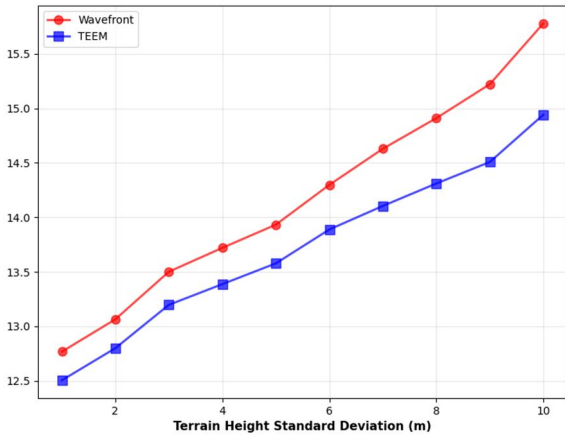


Fig. 2. Energy Consumption vs Terrain Height Variation

그래프에서 확인할 수 있듯이, 지형의 표준편차가 증가할수록 두 알고리즘 모두 에너지 소비가 증가하는 경향을 보였다. 이는 지형 기복이 커질수록 수직 이동이 증가하기 때문이다. 주목할 점은 TEEM 알고리즘이 모든 표준편차 구간에서 일관되게 낮은 에너지 소비를 보였다는 것이다. 특히 지형 변화가 큰 환경(표준편차 8-10m)에서 두 알고리즘 간 에너지 차이가 더욱 뚜렷하게 나타났다. 이는 TEEM이 지형 고도 변화를 명시적으로 고려하여 에너지 효율적인 경로를 선택하기 때문이다. Fig. 3는 지형 표준편차에 따른 TEEM의 에너지 개선율을 나타낸다.

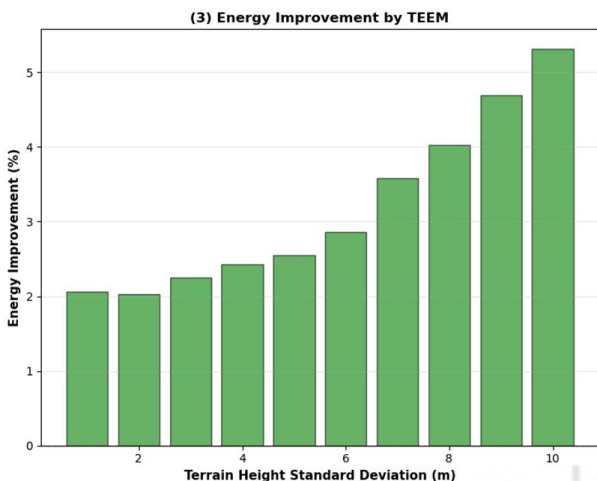


Fig. 3. Energy Improvement by TEEM

TEEM 알고리즘의 에너지 개선율은 지형 표준편차가 증가할수록 향상되는 경향을 보였다. 표준편차 1-2m의 완만한 지형에서는 약 2.0-2.1%의 개선율을 보였으나, 표준편차 10m의 기복이 심한 지형에서는 5.2%까지 향상되었다. 이러한 결과는 다음과 같이 해석할 수 있다:

1. 평탄 지형 환경: 지형 변화가 작을 때 ( $\sigma \leq 2m$ )  
Wavefront와 TEEM이 생성하는 경로가 유사하며, 수직 이동이 최소화되어 두 알고리즘 간 차이가 크지 않다.
2. 기복 지형 환경: 지형 변화가 클 때 ( $\sigma \geq 5m$ )  
TEEM은 에너지 비용을 고려하여 고지대 우회 또는 상공 통과를 선택적으로 결정하는 반면, Wavefront는 거리만 고려하여 비효율적인 경로를 선택할 가능성이 높아진다.
3. 선형적 증가: 표준편차 3-10m 구간에서 개선율이 거의 선형적으로 증가(2.4%에서 5.2%)하는 것은 지형 복잡도와 에너지 효율성 간 강한 상관관계를 보여준다.

위 시뮬레이션 결과를 종합한 결과는 1) 수직 이동 최적화, 2) 지형 적응성, 3) 에너지 효율성으로 요약된다. 수직 이동 최적화 관련하여 TEEM 알고리즘은 상승 거리 26.6%, 하강 거리 25.5%를 감소시켰다. 이는 지형 고도와 에너지 모델을 동시에 고려함으로써 불필요한 수직 이동을 효과적으로 회피한 결과이다. 지형 적응성 관련, 지형 복잡도(표준편차)가 증가할수록 TEEM의 성능 우위가 더욱 명확해졌고 이는 제안 알고리즘이 다양한 지형 환경에서 강건한 성능을 발휘함을 의미한다. 에너지 효율성 관련, 평균 3.2%의 에너지 절감은 배터리 용량이 제한적인 소형 드론 운용 환경에서 실질적인 가치를 갖는다. 예를 들어, 배터리 용량이 500 Wh인 드론의 경우 약 16 Wh의 에너지를 절감할 수 있으며, 이는 추가 비행 시간 또는 페이로드 증가로 활용 가능하다. 본 연구의 한계점은 시뮬레이션 과정에서 정적 환경(고정된 지형)을 가정하였으며, 동적 장애물, 바람, 기상 조건 등은 고려하지 않았다는 점이다. 향후 연구에서는 이러한 요인들을 에너지 모델에 통합하여 더욱 현실적인 경로 계획이 가능할 것으로 기대된다.

V. Conclusions

본 논문에서는 드론의 3차원 이동 특성과 비대칭적 에너지 소비 모델을 고려한 지형 인식 기반 에너지 효율적 경로

계획 알고리즘(TEEM)을 제안하였다. 기존 Wavefront 알고리즘을 포함한 2차원 경로 계획 방법들은 드론의 수직 이동 능력과 상승/하강 시 에너지 소비 차이를 고려하지 않아, 지형 고도 변화가 큰 환경에서 비효율적인 경로를 생성한다는 한계를 갖는다. 제안된 TEEM 알고리즘은 비용 함수를 거리에서 실제 에너지 소비량으로 재정의하고, 수평 이동, 상승, 하강 기동에 대한 실증 기반 에너지 모델을 적용하였다. 또한 Dijkstra 알고리즘을 활용하여 비균일한 에너지 비용 하에서도 최적 경로를 보장하며, 지형 높이 정보를 통합하여 각 셀의 도달 가능성과 비행 고도를 결정한다.

100회의 시뮬레이션(표준편차 1-10m × 10회 반복)을 통한 성능 검증 결과, TEEM 알고리즘은 Wavefront 대비 상승 거리 26.6%, 하강 거리 25.5%를 감소시켰으며, 평균 3.2%의 에너지 절감을 달성하였다. 특히 지형 복잡도가 증가할수록 에너지 개선율이 선형적으로 향상되어(표준편차 1m: 2.1% → 10m: 5.2%), 산악 지형이나 구릉지와 같은 복잡한 환경에서 제안 방법의 효과가 더욱 두드러짐을 확인하였다. 이러한 결과는 배터리 용량이 제한적인 드론 운용 환경에서 비행 시간 연장, 운용 범위 확대, 운영 비용 절감 등 실질적인 가치를 제공할 수 있음을 보여준다. 본 연구는 정적 환경을 가정하였으며, 동적 장애물, 기상 조건, 다중 드론 협업 등은 고려하지 않았다는 한계를 갖는다. 향후 연구에서는 실시간 기상 데이터 통합, 동적 경로 재계획, 다목적 최적화(에너지, 시간, 안전성), 실제 드론을 활용한 필드 테스트 등을 통해 알고리즘을 개선하고 Sim-to-Real 격차를 해소할 필요가 있다. 그럼에도 불구하고 제안된 TEEM 알고리즘은 드론 배송, 정밀 농업, 시설물 점검, 환경 모니터링 등 다양한 상용 응용 분야에서 에너지 효율적 경로 계획의 기반을 제공하며, 향후 더욱 복잡한 실제 운용 시나리오로 확장될 수 있는 가능성을 제시한다.

## REFERENCES

- [1] F. Nex and F. Remondino, "UAV for 3D mapping applications: A review," *Applied Geomatics*, vol. 6, no. 1, pp. 1-15, Mar. 2014.
- [2] J. T. K. Ping, A. E. Ling, T. J. Quan, and C. Y. Dat, "Generic unmanned aerial vehicle (UAV) for civilian application-A feasibility assessment and market survey on civilian application for aerial imaging," in *Proc. IEEE Conf. Sustainable Utilization and Development in Engineering and Technology (STUDENT)*, 2012, pp. 289-294.
- [3] A. Ollero and L. Merino, "Control and perception techniques for

aerial robotics," *Annual Reviews in Control*, vol. 28, no. 2, pp. 167-178, 2004.

- [4] S. H. Chung, B. Sah, and J. Lee, "Optimization for drone and drone-truck combined operations: A review of the state of the art and future directions," *Computers & Operations Research*, vol. 123, p. 105004, Nov. 2020.
- [5] Y. Li et al., "Energy-consumption model for rotary-wing drones," *J. Field Robot.*, vol. 41, no. 4, pp. 1139-1157, Jun. 2024. DOI: 10.1002/rob.22359
- [6] H. K. D. Abeywickrama et al., "Comprehensive energy consumption model for unmanned aerial vehicles, based on empirical studies of battery performance," *IEEE Access*, vol. 6, pp. 58383-58394, 2018. DOI: 10.1109/ACCESS.2018.2875040
- [7] T. A. S. Rodrigues et al., "Drone flight data reveal energy and greenhouse gas emissions savings for very small package delivery," *Patterns*, vol. 3, no. 8, p. 100569, Aug. 2022. DOI: 10.1016/j.patter.2022.100520[4]
- J. Zhang et al., "Energy consumption models for delivery drones: A comparison and assessment," *Transp. Res. D Transp. Environ.*, vol. 90, p. 102668, Jan. 2021. DOI: 10.1016/j.trd.2020.102668
- [8] A. Zelinsky, R. A. Jarvis, J. C. Byrne, and S. Yuta, "Planning paths of complete coverage of an unstructured environment by a mobile robot," in *Proc. International Conference on Advanced Robotics*, Nov. 1993, pp. 533-538.
- [9] V. Shivashankar, R. Jain, U. Kuter, and D. Nau, "Real-time planning for covering an initially-unknown spatial environment," in *Proc. FLAIRS Conference*, 2011, pp. 284-289.
- [10] E. Galceran and M. Carreras, "A survey on coverage path planning for robotics," *Robotics and Autonomous Systems*, vol. 61, no. 12, pp. 1258-1276, Dec. 2013.

## Authors



Sooyoung Moon received the B.S., M.S. and Ph.D. degrees in Computer Engineering from Sungkyunkwan University, Korea, in 2007, 2009 and 2015, respectively.