SCIENCE CHINA
Information Sciences

August 2024, Vol. 67, Iss. 8, 180201:1–180201:19
https://doi.org/10.1007/s11432-023-4087-4

• RESEARCH PAPER •
Special Topic: UAV Swarm Autonomous Control

A path planning algorithm for a crop monitoring fixed-wing unmanned aerial system

Longhao QIAN[1], Yi Lok LO[2] & Hugh Hong-tao LIU[1*]

[1]*Institute of Aerospace Studies, University of Toronto, Toronto* ON M3H 5T6, *Canada;*
[2]*Department of Mechanical Engineering, The University of Hong Kong, Hong Kong* 999077, *China*

Received 24 September 2023/Revised 19 January 2024/Accepted 18 March 2024/Published online 25 July 2024

Abstract With the growing demand for automation in agriculture, industries increasingly rely on drones to perform crop monitoring and surveillance. In this regard, fixed-wing unmanned aerial systems (UASs) are viable platforms for scanning a large crop field, given their payload capacity and range. To achieve maximum coverage without landing for battery replacement, an algorithm for producing a minimal required energy survey path is essential. Hence, an energy-aware coverage path planning algorithm is proposed herein. The constraints for a fixed-wing UAS to fly at low altitudes while achieving full coverage of the crop field are first analyzed. Then, the full path is decomposed into straight-line and U-turn primitives. Finally, an algorithm to calculate a combination of straight-line segments and U-turns is proposed to obtain the path with minimum required energy consumption. The genetic algorithm is used to efficiently determine the order of the straight-line paths to traverse. Case studies show that the proposed algorithm can produce planning results for a convex-polygon-shaped crop field.

Keywords path planning, genetic algorithm, energy minimization, fixed-wing UAS, aerial survey

1 Introduction

1.1 Background

Crop monitoring is essential in the agricultural industry as maximum crop output can be ensured by assessing the health of crops and detecting crop infestations. This process is often conducted manually by farmers through their observations and experiences. With the development of satellites and unmanned aerial systems (UASs), remote-sensing-based crop monitoring has become a promising method of deriving crop information while reducing labor and operational costs [1]. Hence, the goal of a crop monitoring mission is to gather data about crops by taking aerial photos of the entire crop fields, as shown in Figure 1. To collect remotely sensed images with high resolution, a fixed-wing UAS is proposed in this paper for low-altitude crop monitoring. Fixed-wing drones are known for their high endurance and ability to carry heavy payloads compared with multirotor UAS, making them suitable for such long-distance flight missions. To best utilize UAS resources, the amount of time for battery replacement must be reduced because landing and battery swapping for a fixed-wing UAS is time-consuming. Reducing the battery swapping times means reducing the overall operational cost. To minimize the number of battery replacements during a mission, the flight path has to be designed such that the energy requirement is minimized while covering the entire crop field with sufficient overlap. Thus, this paper provides a scheme to solve the minimal required energy path problem by developing a coverage path planning (CPP) algorithm.

1.2 Related work

Several CPP algorithms have been proposed in the past [2–4]. However, directly determining the minimal energy path based on a given crop field geometry is notably numerically intractable. One engineering practice in this regard is assembling the entire path using a set of path primitives, such as back-and-forth

* Corresponding author (email: hugh.liu@utoronto.ca)

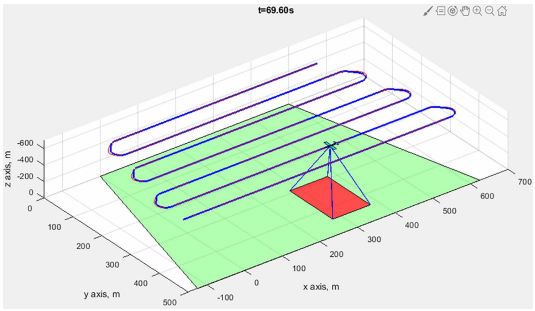

Figure 1 (Color online) Crop monitoring mission.

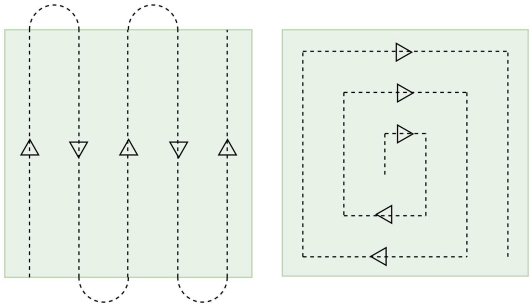

Figure 2 (Color online) Back-and-forth and spiral patterns.

and spiral patterns [3, 4], as shown in Figure 2. For the spiral pattern, pictures are taken during the constant turning maneuvers. Meanwhile, straight-line paths are set along the scanning area for the back-and-forth pattern, and the scanning area is covered using a back-and-forth motion in rows perpendicular to the sweep direction. The back-and-forth pattern is advantageous for crop monitoring in the sense that area coverage can be easily ensured, and the integration of photos can be done more easily. The back-and-forth pattern also makes the optimal path computationally tractable. However, this pattern requires the UAS to perform steep turns to travel from the current straight path to the next segment [5]. Other than the spiral and straight-line patterns, there are other methods implementing higher-order curve fitting at some control points for the optimal path [6].

For arbitrary paths, algorithms such as $A^\star$ are often used to find a path on a two-dimensional discrete map [7]. This method is also used on quadrotor UAS. However, unlike a quadrotor, a fixed-wing UAS has a minimum turning radius. To simplify the analysis of a fixed-wing UAS with a proper flight controller, a Dubins car [8, 9] is often used to model such a system. There are various optimal path algorithms proposed for the Dubins car. For instance, Askari et al. [10] proposed a Bezier-Dubins curvature path planner. The resultant paths were suboptimal Dubins-based paths, and close to the minimum length. Meanwhile, Babel [11] proposed a method to solve the Dubins traveling salesman problem (TSP), which finds the optimal flight path by minimizing the distance to traverse between all waypoints. Minimizing the turning radius is also taken into account. This is done by optimizing the heading angle of each waypoint. Tripicchio et al. [12] proposed an algorithm that generates waypoints inside a polygon and then proposed an optimal flight path by solving the TSP with smooth curves to minimize the overall memory footprint. Coombes et al. [13] proposed a method to improve the path planning algorithm by accounting for wind in the sweep direction optimization and using cell decomposition and recombination to decompose polygons into convex cells.

Hawary and Chipperfield [14] proposed a method that iterates through a route planner, path planner, and coverage planner until the optimal solution is found. This route algorithm can solve the TSP. If the solution is not collision-free, the path planner alters the path so that it avoids the obstacle. For a large nonconvex crop field, a common approach for efficient planning is to decompose the field into several convex subfields. Li et al. [15] proposed a coverage path planning method using convex decomposition. Their method decomposes shapes into convex-shaped cells minimizing the sum of the widths of the convex subcells using the greedy recursive method. To perform a crop field survey using a group of drones, Kapanoglu et al. [16] separated straight path sequences into N sections, where N is the number of UAVs used. They altered the fitness function such that it calculated the total energy used for the N sections. Finally, each robot was responsible for its own surveying area.

1.3 Contributions and paper structure

In this paper, we propose an algorithm to determine the optimal flight path for crop monitoring using a fixed-wing UAS. An improved back-and-forth pattern is presented for crop fields with the shape of a convex polygon. First, the kinematics of a fixed-wing UAS are analyzed to produce the physical constraints of the system. Additional path constraints are presented according to the geometry of the camera and the crop field. Then, the entire path is decomposed into straight-line segments and steady U-turns. The optimal number of straight-line paths is determined by finding the minimum height of the polygon-shaped crop field. After straight-line paths are obtained, the problem of finding steady level

U-turns is transformed into a TSP problem. The TSP problem is then solved using a genetic algorithm (GA) [17,18]. The contributions of this paper are listed as follows.

(1) An algorithm for finding the optimal sweep direction is proposed. By defining the height of a polygon-shaped crop field, the optimal sweep direction can be determined resulting in the minimal amount of required turns.

(2) After the optimal sweep direction is obtained, the order of path switching using U-turns becomes a TSP. A GA is used to obtain the final complete path consisting of straight-lines and U-turns.

(3) Compared with the classic brute-force approach to solve the TSP, the proposed method using GA has significantly higher computational speed, making it more scalable to large crop fields.

The remainder of this paper is structured as follows. Section 2 models the fixed-wing UAS, the geometry of the onboard camera, and the parameters of a crop field. Section 3 analyzes the path constraints and proposes path primitives to formulate the planning problem as an optimization problem. Section 4 proposes a method to optimize the flight trajectory for a convex polygonal crop field to minimize the energy used throughout the mission. Section 5 demonstrates the aircraft completing the whole mission in a simulated environment using MATLAB. Finally, Section 6 presents the conclusion and some future work that could be done.

2 System modeling

2.1 Mathematical preliminaries

Some mathematical conventions are used in this work. Scalars are defined with regular letters, such as $a \in \mathbb{R}$. Let bold lowercase letters denote vectors, such as $\boldsymbol{v} \in \mathbb{R}^{N\times 1}$. Let bold upper case letters denote matrices, such as $\boldsymbol{M} \in \mathbb{R}^{N\times M}$. For a matrix $\boldsymbol{M}$, $\boldsymbol{M}_{i,j}$ denotes the element at the ith row and the jth column. $\mathrm{col}(\boldsymbol{M})_i$ and $\mathrm{row}(\boldsymbol{M})_i$ denote the ith column and row, respectively. $||\boldsymbol{v}||$ is the 2-norm of a vector.

2.2 Dynamics of a fixed-wing UAS

Because the back-and-forth pattern is used to generate the flight path, the physical constraints of a fixed-wing UAS must be found as prerequisites of the optimization problem. For a UAS, s is the reference area, and C_D and C_L are drag and lift coefficients, respectively. v is the airspeed, ρ is the air density, and g is the gravitational acceleration. W and m are the weight and mass of the drone, respectively. Let L and D be the lift and drag of the aircraft, respectively:

$$L = \frac{1}{2}\rho v^2 s \cdot C_L,\ D = \frac{1}{2}\rho v^2 s \cdot C_D. \tag{1}$$

The drag coefficient C_D can be expressed by the summation of the zero-lift drag C_{D_0} and the induced drag κC_L^2:

$$C_D = C_{D_0} + \kappa C_L^2 = C_{D_0} + \frac{C_L^2}{\pi e \zeta}, \tag{2}$$

where ζ is the aspect ratio of the main wing, e is the Oswald efficiency coefficient, and κ is the induced drag coefficient. As this research focuses on the trajectory generation for a crop monitoring mission, the UAS performs steady straight-line and turning flights. Therefore, we simplify the analysis of the UAS and ignore the acceleration and deceleration as stated in Assumption 1.

Assumption 1. We assume that the UAS is equipped with a complete and functional autopilot system so that we can explicitly control the UAS to perform desired steady-state level flights. The whole flight path is assumed to be two-dimensional at a constant altitude. Both turning and straight-line flights are steady level, and we ignore the take-off and landing mission segments. Moreover, the energy consumption for acceleration required for speed changes in the flight direction is ignored.

The steady-state equations of motion of a UAS in straight-line flight are summarized as follows:

$$\begin{cases} D = T_s, \\ L = W, \end{cases} \tag{3}$$

where T_s is the steady-level flight thrust. Therefore, according to (1), we have the following:

$$\frac{T_s}{W}=\frac{D}{L}=\frac{\frac{1}{2}\rho v^2 s\cdot C_D}{\frac{1}{2}\rho v^2 s\cdot C_L}=\frac{C_D}{C_L},\quad T_s=\frac{W}{C_L/C_D}=\frac{W}{C_L/(C_{D_0}+\kappa C_L^2)}. \tag{4}$$

The governing equations for a steady-level turn with a radius r are shown below:

$$\begin{cases} D=T_t, \\ L=\sqrt{W^2+\left(\frac{mV^2}{r}\right)^2}, \end{cases} \tag{5}$$

where T_t is the thrust at the steady-level turn. Therefore, according to (1), we have the following:

$$\begin{aligned} L&=\frac{1}{2}\rho v^2 s\cdot C_L=\sqrt{W^2+\left(\frac{mv^2}{r}\right)^2}, \\ C_L&=\frac{2\sqrt{\frac{W^2}{v^4}+\frac{m^2}{r^2}}}{\rho s}. \end{aligned} \tag{6}$$

Combining (1), (5), and (6), we have the following:

$$T_t(v,r)=\frac{1}{2}\rho v^2 s\cdot(C_{D_0}+\kappa C_L^2)=\frac{1}{2}\rho v^2 s\cdot\left(C_{D_0}+\kappa\frac{\frac{W^2}{v^4}+\frac{m^2}{r^2}}{\frac{1}{4}\rho^2 s^2}\right). \tag{7}$$

For straight-line flight, the airspeed must be above the minimum airspeed to prevent the aircraft from stalling:

$$\sqrt{\frac{W}{\frac{1}{2}\rho s\cdot C_{L_{\max}}}}\leqslant v, \tag{8}$$

where $C_{L_{\max}}$ is the maximum lift coefficient before stalling. During a steady-level turn, the required lift coefficient is limited to prevent stalling. The load factor n is limited by the structural limit of the drone. For a given maximum load factor $n_{\max}$, the constraints on n and C_L are shown in (9) and (10):

$$n(v,r)=\sqrt{\left(\frac{v^2}{r\cdot g}\right)^2+1}\leqslant n_{\max}, \tag{9}$$

$$C_L(v,r)=\frac{2W}{\rho s\cdot v^2}\sqrt{\left(\frac{v^2}{r\cdot g}\right)^2+1}\leqslant C_{L_{\max}}. \tag{10}$$

2.3 Geometry of the crop field

The shape of the crop field is shown in Figure 3. Let M be the number of edges in the crop field. The convex polygonal crop field is defined by a list of coordinates of its vertices in counterclockwise order. The edges e_i are also defined by the line starting at (x_i, y_i) and ending at (x_{i+1}, y_{i+1}) with the exception of the final edge e_M, where the line starts at (x_M, y_M) and ends at (x_1, y_1). Let $\boldsymbol{V}\in\mathbb{R}^{2\times(M+1)}$ denote the matrix containing the column-wise stacked polygon vertices:

$$\boldsymbol{V}=\begin{bmatrix} x_1 & x_2 & \cdots & x_i & \cdots & x_M & x_1 \\ y_1 & y_2 & \cdots & y_i & \cdots & y_M & y_1 \end{bmatrix}. \tag{11}$$

Note that we duplicate the coordinates of the first vertex into the last column for subsequent calculation. For the crop field used in this work, we have the following convex assumption.

Assumption 2. The crop field is assumed to be convex and polygonal. It is composed of straight edges and vertices, where all the internal angles are less than 180°. We also assume that there is no wind in the crop field and that the crop field is level.

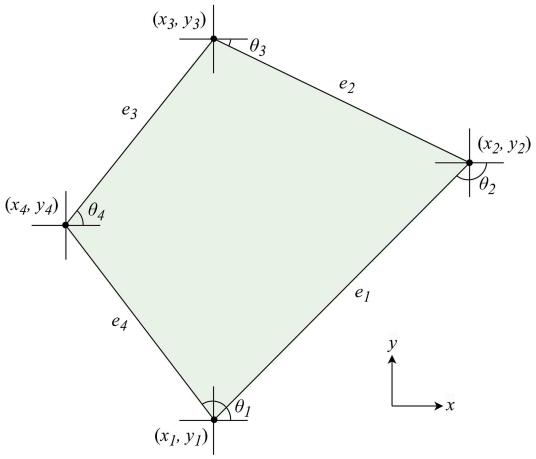

Figure 3 (Color online) Geometry of the crop field.

2.4 Geometry of aerial photography

As shown in Figure 4(a), the camera parameters are as follows: f is the camera focal length. ξ_x and ξ_y are the sensor height and width, respectively. I_x and I_y are the corresponding image height and width measured in pixels. The pixel length p is the physical size of a camera pixel:

$$p = \frac{\xi_x}{I_x} = \frac{\xi_y}{I_y}. \tag{12}$$

To describe the quality of aerial photography, the ground sampling distance (GSD) and overlap percentages (o_x, o_y) are used. Let L_x and L_y be the physical distance measured by a single frame taken by the camera. The GSD represents the physical distance measured on the ground by a single camera pixel:

$$\text{GSD} = \frac{L_x}{I_x} = \frac{L_y}{I_y}. \tag{13}$$

To ensure the integrity of the photo survey, we require an overlap between consecutive photos. Let η_x and η_y be the overlap distance between two ground areas measured by two consecutive camera frames. The overlap percentages (o_x, o_y) are defined as the amount of overlap between two spatially consecutive pictures in both x and y directions as shown in Figure 4(b):

$$o_x = \eta_x / L_x, \ o_y = \eta_y / L_y. \tag{14}$$

The percentages have to be sufficiently high for adjacent photos to successfully align with each other, as shown in Figure 4(b). Additional assumptions are made for the camera model.

Assumption 3. To simplify the analysis of the camera geometry, a small-angled gimbal is used to compensate for small rotational errors during flight, such that the onboard camera is parallel to the ground at all times.

3 Problem formulation

3.1 Height constraints

The UAS is assumed to fly at the same altitude at all times. To achieve the minimum required GSD_m, the maximum altitude is defined as $h_{\max}$ [19]. According to Figure 4(a), applying the property of side proportionality for similar triangles, $h_{\max}$ can be obtained as follows:

$$\frac{h}{f} = \frac{L_x}{\xi_x} = \text{GSD} \cdot \frac{I_x}{\xi_x} = \frac{\text{GSD}}{p} \rightarrow h_{\max} = \text{GSD}_m \cdot \frac{f}{p}. \tag{15}$$

Note that in the remainder of the analysis, the UAS is assumed to fly at $h_{\max}$. Hence, the optimal path is horizontal at $h_{\max}$.

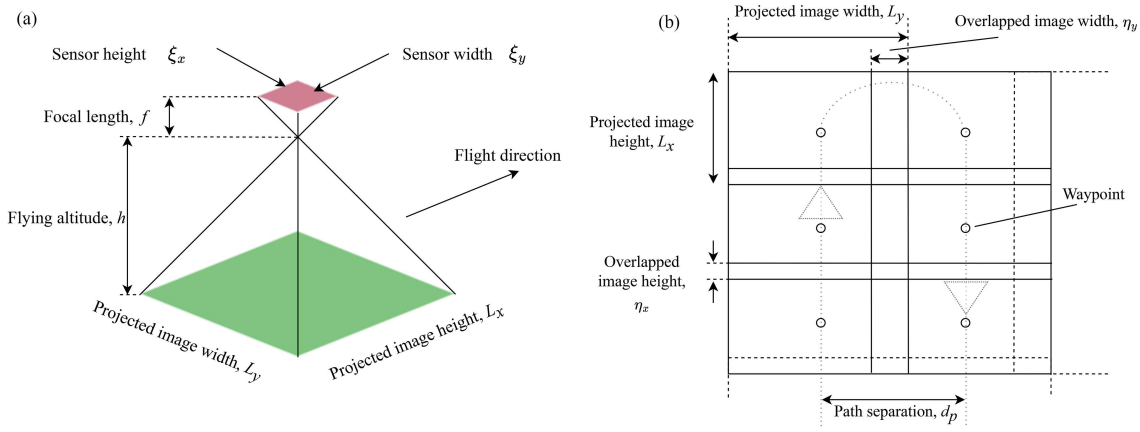

Figure 4 (Color online) (a) Geometry of aerial photography; (b) definition of image overlap.

3.2 Straight-line path constraints

On the basis of the minimum required image overlap $o_{x,\min}$, the airspeed must not exceed an upper limit $v_{\max}$, as shown in Figure 4(b). Let t_s be the sampling time of the camera. The airspeed must be slow enough such that the distance traveled between two consecutive images satisfies $o_{x,\min}$:

$$v \cdot t_s \leqslant L_x - l_x. \tag{16}$$

Therefore, the maximum allowable flight speed $v_{\max}$ is

$$v_{\max} = \frac{L_x}{t_s}(1 - o_{x,\min}). \tag{17}$$

On the basis of the minimum required $o_{y,\min}$, the minimum separation between two straight flight paths d_p is

$$d_p = L_y(1 - o_{y,\min}). \tag{18}$$

According to the dynamics constraint in (8), the straight-line path constraints are summarized as follows:

$$\begin{cases} \sqrt{\frac{2W}{\rho s \cdot C_{L\max}}} - v \leqslant 0, \\ v - \frac{L_x}{t_s}(1 - o_{x,\min}) \leqslant 0, \\ h - \mathrm{GSD}_{\min} \cdot \frac{f}{P} \leqslant 0, \\ L_y(1 - o_{y,\min}) - d_p \leqslant 0. \end{cases} \tag{19}$$

3.3 Turning constraints

The UAS does not take any images during turning. Hence, the constraints are purely based on kinematics. According to the dynamics in (9) and (10), the turning path constraints are

$$\begin{cases} v^4 - r^2 g^2 (n_{\max}^2 - 1) \leqslant 0, \\ (4W^2 - \rho^2 s^2 C_{L\max}^2 r^2 g^2) v^4 + 4W^2 r^2 g^2 \leqslant 0. \end{cases} \tag{20}$$

3.4 Path primitives

Directly solving for an optimum path satisfying constraints in (19) and (20) is an infinite-dimensional and intractable problem. Hence, to make the problem feasible, the entire path is decomposed into N_s straight-line primitives and N_t turning primitives.

Straight-line path primitive (SPP). The SPP is used as the main part of the survey path. The ith path segment is parameterized by its starting point $\boldsymbol{x}_{s,i}$ and its endpoint $\boldsymbol{x}_{e,i}$. Hence, let $\mathrm{SPP}_i = \{\boldsymbol{x}_{s,i}, \boldsymbol{x}_{e,i}\}$ denote an SPP. $E_{ts,i}$ is defined as the total required energy at the tth SPP:

$$E_{ts,i} = T_{s,i} \cdot l_i, \tag{21}$$

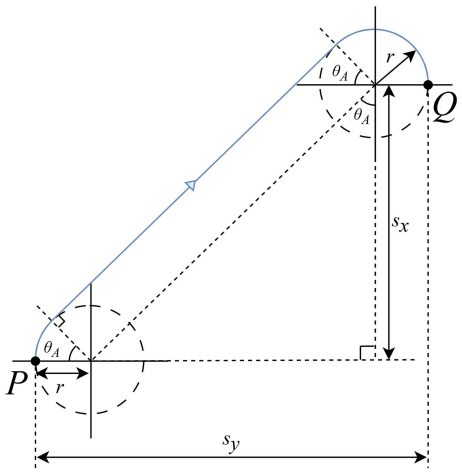

Figure 5 (Color online) Type 1 U-turn ($r \leqslant s_y/2$).

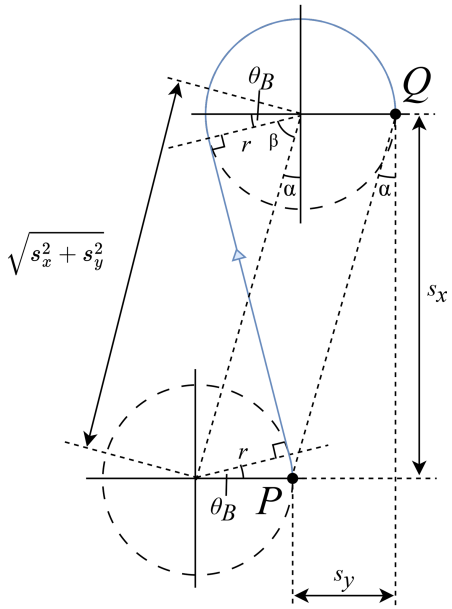

Figure 6 (Color online) Type 2 U-turn ($s_y/2 < r \leqslant s/2$).

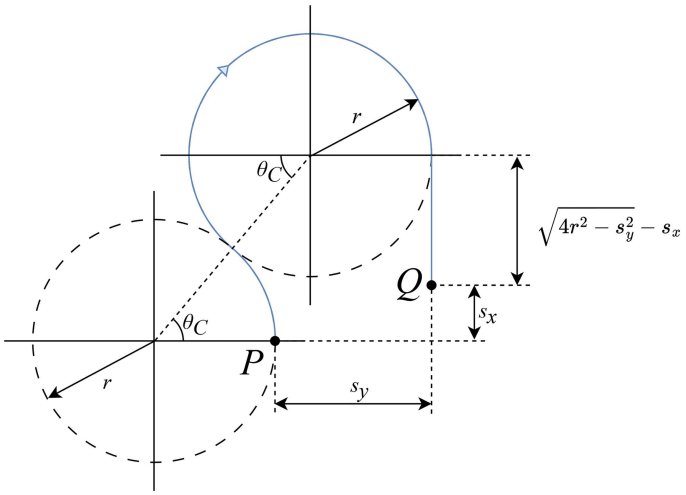

Figure 7 (Color online) Type 3 U-turn ($r > s/2$).

Table 1 Parameters of an SUP

Symbol	Definition	Symbol	Definition
r	Radius of the circle arc	α	$\text{atan}(s_y/s_x)$
s_x	Distance between P and Q in the x-directio	β	$\text{acos}(2r/s)$
s_y	Distance between P and Q in the y-direction	θ_B	$\pi/2 - \alpha - \beta$
s	Absolute distance between points P and Q	θ_C	$\text{acos}(2r/s_y)$
θ_A	$\text{atan}((s_y - 2r)/s_x)$		

where $T_{s,i}$ is the required thrust on the ith straight-line primitive. $l_i = ||\boldsymbol{x}_{s,i} - \boldsymbol{x}_{e,i}||$.

Steady-level U-turn primitive (SUP). At the end of each SPP, the plane performs steady-level U-turns to switch to another straight-line path. To minimize the required energy in these turns, Dubins paths are adopted [20]. Three types of Dubins paths, i.e., Types 1, 2, and 3, are used, each suited for a certain range of turning radius [10]. The geometries of Types 1, 2, and 3 SUPs are shown in Figures 5, 6, and 7, respectively, and the shape of each candidate is defined using the variables in Table 1. These SUPs are composed of straight lines and arcs with radius r. The shape of each candidate is defined using the variables in Table 1. The optimal required energy of an SUP is a function of s_x and s_y defined as

$E^{\star}_{tt,i}(s_x, x_y)$. Subsection 4.2 provides the formulas to calculate $E^{\star}_{tt,i}(s_x, x_y)$ in (33).

3.5 Optimization problem

As shown in Figure 1, the photos are taken when the UAS is on steady-level straight paths. The UAS switches from one path to another using steady-level U-turns. The two maneuvers are independent, and we, therefore, define the energy required in the following manner (22):

$$E_t = E_{ts} + E_{tt} = \sum_{i=1}^{N_s} E_{ts,i} + \sum_{i=1}^{N_t} E_{tt,i}, \tag{22}$$

where E_{ts} is the total energy of the SPPs and E_{tt} is the total energy of SUPs. The optimization problem is formulated as finding a combination of SPPs and SUPs for the full coverage of a polygon field, that is, $\mathrm{SPP} = \{\mathrm{SPP}_1, \mathrm{SPP}_2, \ldots, \mathrm{SPP}_{N_s}\}$ and $\mathrm{SUP} = \{\mathrm{SUP}_1, \mathrm{SUP}_2, \ldots, \mathrm{SUP}_{N_t}\}$, which satisfies the constraints in (19) and (20), such that E_t is minimized:

$$\begin{aligned} \{\mathrm{SPP}^{\star}_1, \ldots, \mathrm{SPP}^{\star}_{N_s}\},\ \{\mathrm{SUP}^{\star}_1, \ldots, \mathrm{SUP}^{\star}_{N_t}\} = \operatorname{argmin}\ & E_t \text{ in (22)} \\ \text{s.t.}\quad & \text{Eqs. (19) and (20) are satisfied.} \end{aligned} \tag{23}$$

4 Proposed algorithm

The proposed algorithm encompasses three main steps. This first step is finding the minimum required energy on SPP and SUP. Note that SPP and SUP are independent of each other; thus, the optimal required energy is achieved when the required energy on the individual primitives is minimized. Then, the optimal sweep direction is determined to obtain $\mathrm{SPP}^{\star}$. The optimal sweep direction ensures a minimum number of SPPs, which results in a minimum number of SUPs. Finally, the optimal sequence of straight-line path execution to determine $\mathrm{SUP}^{\star}$ using the GA is proposed.

4.1 Minimum required energy of SPPs

On an SPP, the energy consumption is modeled in (24):

$$E_{ts,i} = T_s \cdot l_i. \tag{24}$$

Because l_i and the required thrust T_s are independent of each other, $E_{ts,i}$ is minimized when T_s is minimized. According to (4), we could minimize T by maximizing the lift-to-drag ratio (LDR). The maximum LDR is calculated as follows:

$$\mathrm{d}(C_L/C_D)/\mathrm{d}C_L = 0 \to C_L = \sqrt{C_{D_0}/\kappa}. \tag{25}$$

Hence we have the following:

$$\left(\frac{C_L}{C_D}\right)_{\max} = \frac{1}{2\sqrt{\kappa C_{D_0}}}. \tag{26}$$

Substituting the above into (4) and the drag equation, we can find the minimized thrust $T^{\star}_s$ and the optimized flight speed v^*_s for SPP.

$$T^{\star}_s = \frac{W}{(\frac{C_L}{C_D})_{\max}} = 2W\sqrt{\kappa C_{D_0}}, \tag{27}$$

$$T^{\star}_s = \frac{1}{2}\rho {v^*_s}^2 s\left(C_{D_0} + \kappa\sqrt{\frac{C_{D_0}}{\kappa}}^2\right) \to v^*_s = \sqrt{\frac{2W}{\rho s\sqrt{C_{D_0}/\kappa}}}. \tag{28}$$

Hence, for an SPP with starting and ending coordinates $\boldsymbol{x}_{s,i}$ and $\boldsymbol{x}_{e,i}$, the minimal energy required $E^{\star}_{ts,i}(\boldsymbol{x}_{s,i}, \boldsymbol{x}_{e,i})$ is

$$E^{\star}_{ts,i}(\boldsymbol{x}_{s,i}, \boldsymbol{x}_{e,i}) = T^{\star}_s ||\boldsymbol{x}_{s,i} - \boldsymbol{x}_{e,i}||. \tag{29}$$

4.2 Minimum required energy of SUPs

The thrust required for turning $T_t(v,r)$ is defined in (7). According to the nonlinear constraints shown in (9) and (10), the minimum required energy of Types 1, 2, and 3 SUPs defined in Subsection 3.4 is obtained as follows:

(1) When $r \leqslant s_y/2$, Type 1 turns are used. The minimum required energy in Type 1 turns is obtained by solving the following optimization problem:

$$E_1^\star = \min_{v,r} \left[T_s^\star \cdot \sqrt{s_x^2 + (s_y - 2r)^2} + T_t(v,r) \cdot r \cdot \pi \right]$$
$$\text{s.t.} \begin{cases} 0 < r \leqslant s_y/2, \\ v > 0, \\ \text{Eq. (20) is satisfied.} \end{cases} \tag{30}$$

(2) When $s_y/2 < r \leqslant s/2$, Type 2 turns are used. The minimum required energy in Type 2 turns is obtained by solving the following optimization problem:

$$E_2^\star = \min_{v,r} \left[T_s^\star \cdot \sqrt{s^2 - 4r^2} + T_t(v,r) \cdot r \cdot (\pi + 2\theta_B) \right]$$
$$\text{s.t.} \begin{cases} s_y/2 < r \leqslant s/2, \\ v > 0, \\ \text{Eq. (20) is satisfied.} \end{cases} \tag{31}$$

Note that θ_B can be obtained using the formulas in Table 1.

(3) When $r > s/2$, Type 3 turns are used. The minimum required energy in Type 3 turns is obtained by solving the following optimization problem:

$$E_3^\star = \min_{v,r} \left[T_s^\star \cdot \left(\sqrt{4r^2 - s_y^2} - s_x \right) + T_t(v,r) \cdot r \cdot (\pi + 2\theta_C) \right]$$
$$\text{s.t.} \begin{cases} r > s/2, \\ v > 0, \\ \text{Eq. (20) is satisfied.} \end{cases} \tag{32}$$

Note that θ_C can be obtained using the formulas in Table 1.

For each turn, the algorithm calculates the minimum required energy for all three SUP candidates and picks the one with the lowest required energy. The required energy of the ith SUP is defined as $E_{tt,i}(j) \in \{E_1^\star, E_2^\star, E_3^\star\}$. $(v_{t,i}^\star, r_i^\star)(j) \in \{(v_{t,1}^\star, r_2^\star), (v_{t,2}^\star, r_1^\star), (v_{t,3}^\star, r_3^\star)\}$ is the corresponding velocity and radius pair, where $j = 1, 2, 3$. Hence, $E_{tt,i}^\star(s_x, x_y)$ can be obtained as follows:

$$E_{tt,i}^\star(s_x, s_y) = \min_{j \in \{1,2,3\}} E_{tt,i}(j). \tag{33}$$

The corresponding SUP type to achieve $E_{tt,i}^\star(s_x, x_y)$ is $j^\star$. The optimal velocity and radius of the SUP are $(v_{t,i}^\star, r_i^\star)(j^\star)$.

4.3 Determining the optimal sweep direction

To cover a polygon-shaped crop field, we must determine the orientation of SPPs given the spacing d_p defined in (18). As shown in Figure 8, because a back-and-forth pattern is used in this paper, more turns are needed in some sweep directions. Hence, an optimal sweep direction must be determined to minimize the number of turns to save energy and time.

We define a mission coordinate frame as shown in Figure 9. The x-axis of the mission frame is parallel to these straight paths, whereas the y-axis of the mission frame is parallel to the sweep direction. Note that rotating the mission frame is equivalent to rotating the crop field w.r.t. the mission frame. For subsequent analysis, we fix the mission frame and rotate the crop field around its geometric center. Let d denote the height of the polygon crop field, as shown in Figure 9. The height is defined as the maximum

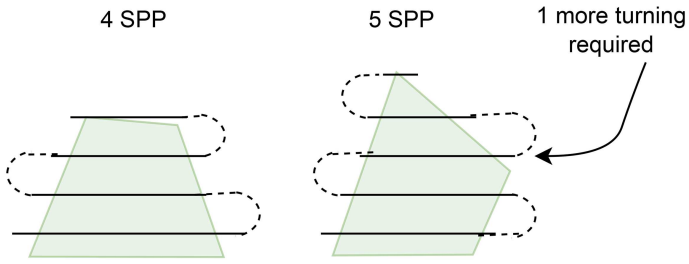

Figure 8 (Color online) Sweeping direction determining the number of turns.

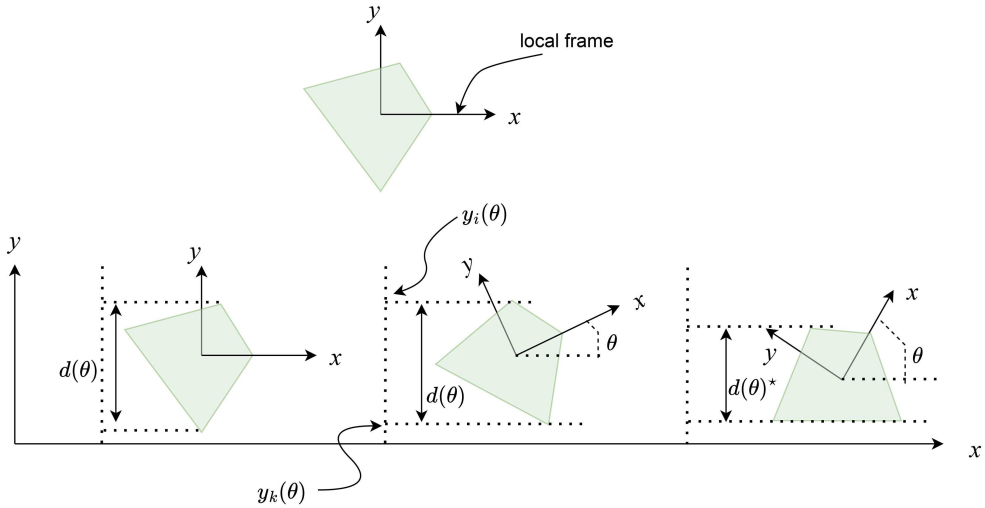

Figure 9 (Color online) Illustration of finding d^* by rolling the polygon.

difference between two vertices in y coordinates. If the crop field rotates by θ, the height d changes. Hence, d is a function of θ defined in the following formula:

$$d(\theta) = \max_{1 \leqslant i,k \leqslant M} (y_i(\theta) - y_k(\theta)). \tag{34}$$

Because d is a function of θ, we must find an optimal $\theta^\star$ such that d is minimized, as stated in the following optimization problem:

$$\theta^\star = \arg\min_{\theta} \Big[\max_{1 \leqslant i,k \leqslant M} (y_i(\theta) - y_k(\theta)) \Big]. \tag{35}$$

From a previous study [21], we know that the local minimum of $d(\theta)$ is located at the orientation where an edge is perpendicular to the y axis. Another work [21] also suggested that this problem could be solved by calculating $d(\theta)$ for every possible θ where y is perpendicular to an edge. On this basis, a pseudo-code of the above process is provided in Algorithm 1 in finding d^* and $\theta^\star$. The coordinates of the polygon field after rotating $\theta^\star$ is $\boldsymbol{V}^\star \in \mathbb{R}^{2\times(M+1)}$. Note that we shifted the order of vertex coordinates in $\boldsymbol{V}^\star$ such that the vertex defining the bottom of the polygon is in the first two columns.

After $\theta^\star$ is obtained, we set the sweep direction parallel to the y axis of the mission frame. The number of SPPs required, i.e., N_s, could be found by removing the overlapped distance η_y from the optimized height $d(\theta^\star)$, and then dividing the result by the path separation d_p and rounding up to the nearest integer, as shown in (36):

$$N_s = \left\lceil \frac{d(\theta^*) - \eta_y}{d_p} \right\rceil. \tag{36}$$

The path separation d_p could then be readjusted so that the photos do not cover areas outside the region of interest, as shown in Figure 10.

$$d_p^* = \frac{d(\theta^\star) - L_y}{N_s - 1}. \tag{37}$$

Algorithm 1 Sweep direction optimization

Input: Map vertex matrix $\boldsymbol{V}$ from (11).
Output: d^*, θ^*, $\boldsymbol{V}_f$.

```
for i = 1 to M do
    θ ← atan2((V_{2,i+1} − V_{2,i}), (V_{1,i+1} − V_{1,i}));
    Ṽ ← [ cosθ  sinθ ; −sinθ  cosθ ] × V;
    d ← max(row(Ṽ)_2) − Ṽ_{2,i};
    if i == 1 or d < d* then
        d* ← d;
        θ* ← θ;
        i* ← i;
    end if
end for
if i* == M then
    V_s ← [col(V)_M col(V)_1 ··· col(V)_M];
else
    V_s ← [col(V)_{i*} col(V)_{i*+1} ··· col(V)_M col(V)_1 ··· col(V)_{i*−1} col(V)_{i*}];
end if
V* ← [ cosθ*  sinθ* ; −sinθ*  cosθ* ] × V_s.
```

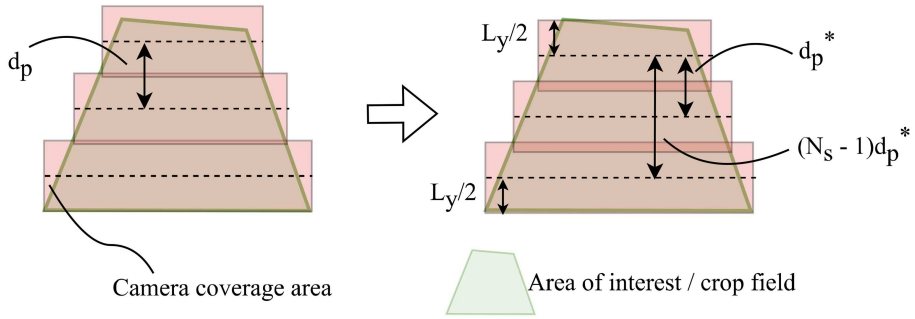

Figure 10 (Color online) Optimizing path separation d_p while ensuring full aerial coverage.

4.4 Determining the coordinates of SPPs

Let $\boldsymbol{W}_n$, $\boldsymbol{W}_f \in \mathbb{R}^{2\times N_s}$ denote the starting and ending waypoints for each SPP. These waypoints are also used to determine the SUPs connecting SPPs. $\boldsymbol{W}_n$ contains the coordinates of the SPP closer to the UAS starting position and vice versa for $\boldsymbol{W}_f$. Starting with the leftmost path, the y-coordinates are set according to $\boldsymbol{V}^\star$, L_y, and d_p^*. Then, the x-coordinates are set subsequently depending on the intersection points between the straight flight path and the perimeter of the crop field. Figure 11 shows the waypoints set given a crop field and photogrammetry constraints. In this case, the UAS starts from the left side of the crop field. The starting SPP index is determined by the optimization results in Subsection 4.5. Algorithm 2 is provided to illustrate the process.

Given $\boldsymbol{W}_n$, $\boldsymbol{W}_f$, we can then calculate s_x and s_y between the ith and jth straight path at the near side according to (38) as follows:

$$s_x = |\boldsymbol{W}_{n,1,i} - \boldsymbol{W}_{n,1,j}|,\ s_y = |\boldsymbol{W}_{n,2,i} - \boldsymbol{W}_{n,2,j}|. \tag{38}$$

Similarly, for the far side, we use (39) to obtain s_x and s_y:

$$s_x = |\boldsymbol{W}_{f,1,i} - \boldsymbol{W}_{f,1,j}|,\ s_y = |\boldsymbol{W}_{f,2,i} - \boldsymbol{W}_{f,2,j}|. \tag{39}$$

4.5 Optimizing the sequence of straight path execution

Once $\boldsymbol{W}_n$ and $\boldsymbol{W}_f$ are obtained, the SPP portion of the path is fixed. The parameters for an SUP connecting the ith and the jth SPPs can be determined using (38) and (39). We must determine the order of turnings such that the total required energy is minimized. The required energy for using a single SUP to switch from one SPP to another SPP is introduced in (33). Hence, the remaining problem becomes a TSP, that is, determining the order in which the UAS should traverse the SPPs. To formulate

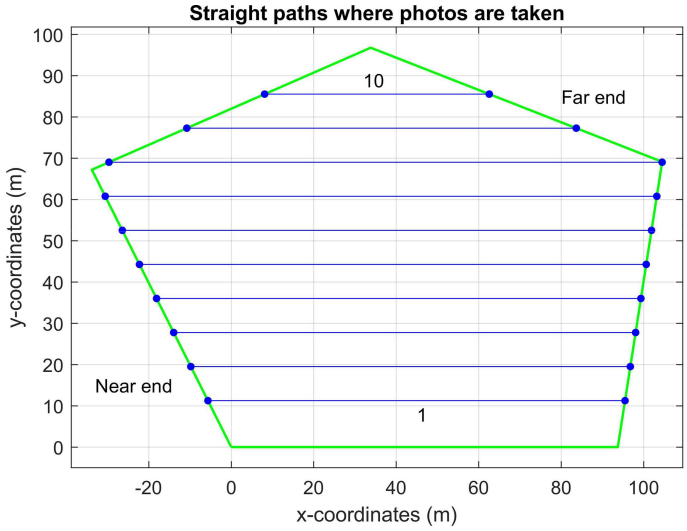

Figure 11 (Color online) Waypoint setting given a crop field.

Algorithm 2 Waypoint setting

Input: L_y, d_p^*, $\boldsymbol{V}^\star$, N_s, M;
Output: $\boldsymbol{W}_n$, $\boldsymbol{W}_f$;
$y_{n,1} \leftarrow L_y/2$;
$y_{f,1} \leftarrow L_y/2$;
for j $= 2$ to N_s **do**
  $y_{n,j} \leftarrow y_{n,j-1} + d_p^*$;
  $y_{f,j} \leftarrow y_{f,j-1} + d_p^*$;
end for
for $i = 1$ to N_s **do**
  for $j = M$ to 2 **do**
    if $\boldsymbol{V}^\star_{2,j} \geqslant y_{n,i}$ **then**
      $k \leftarrow \frac{\boldsymbol{V}^\star_{1,j} - \boldsymbol{V}^\star_{1,j+1}}{\boldsymbol{V}^\star_{2,j} - \boldsymbol{V}^\star_{2,j+1}}$;
      $x_{n,i} \leftarrow k \times (y_{n,i} - \boldsymbol{V}^\star_{2,j}) + \boldsymbol{V}^\star_{1,j}$;
      break;
    end if
  end for
end for
for $i = 1$ to N_s **do**
  for $j = 2$ to M **do**
    if $\boldsymbol{V}^\star_{2,j} \geqslant y_{f,i}$ **then**
      $k \leftarrow \frac{\boldsymbol{V}^\star_{1,j} - \boldsymbol{V}^\star_{1,j-1}}{\boldsymbol{V}^\star_{2,j} - \boldsymbol{V}^\star_{2,j-1}}$;
      $x_{f,i} \leftarrow k \times (y_{f,i} - \boldsymbol{V}^\star_{2,j}) + \boldsymbol{V}^\star_{1,j}$;
      break;
    end if
  end for
end for
$\boldsymbol{W}_n \leftarrow \begin{bmatrix} x_{n,1} & x_{n,2} & \cdots & x_{n,N_s} \\ y_{n,1} & y_{n,2} & \cdots & y_{n,N_s} \end{bmatrix}$;
$\boldsymbol{W}_f \leftarrow \begin{bmatrix} x_{f,1} & x_{f,2} & \cdots & x_{f,N_s} \\ y_{f,1} & y_{f,2} & \cdots & y_{f,N_s} \end{bmatrix}$.

our problem to find the sequence of SUPs, two energy matrices, $\boldsymbol{E}_n \in \mathbb{R}^{N_s \times N_s}$ and $\boldsymbol{E}_f \in \mathbb{R}^{N_s \times N_s}$, are built for the path-switching maneuver that occurs at the near end and the far end, respectively, to store the required energy for SUPs,

$$\boldsymbol{E}_n = \begin{bmatrix} 0 & \cdots & E^\star_{tt,1,N_s} \\ \vdots & \ddots & \vdots \\ E^\star_{tt,N_s,1} & \cdots & 0 \end{bmatrix}, \quad \boldsymbol{E}_f = \begin{bmatrix} 0 & \cdots & E^\star_{tt,1,N_s} \\ \vdots & \ddots & \vdots \\ E^\star_{tt,N_s,1} & \cdots & 0 \end{bmatrix}. \tag{40}$$

In the energy matrix, the element in the ith column and jth row corresponds to the energy required

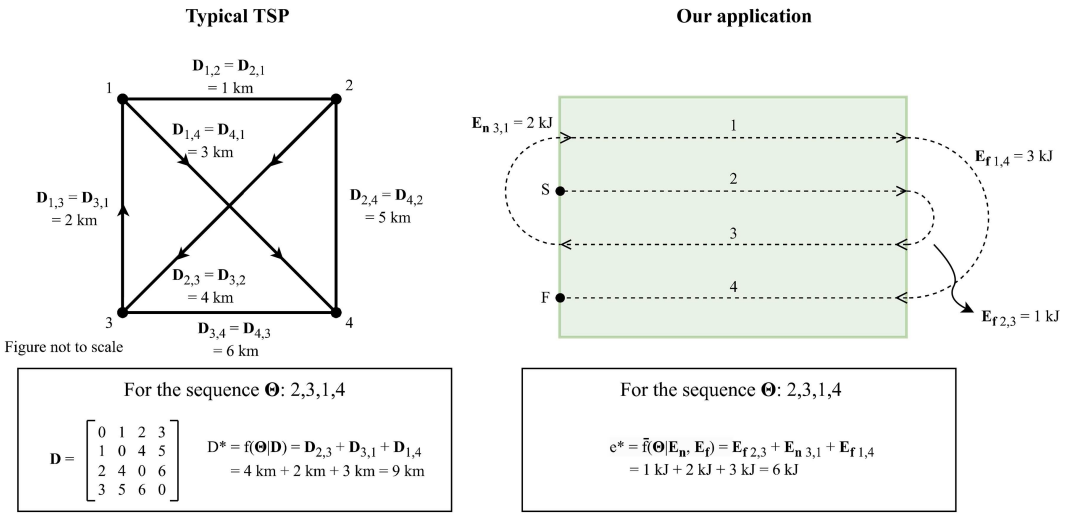

Figure 12 (Color online) Illustration of the application of GA and TSP into solving the optimal sequence of traversing SPPs.

to travel from the ith SPP to the jth SPP according to (33). As shown in Figure 11, the 1st straight path corresponds to the one with the lowest y coordinate, whereas the Nth corresponds to the one with the largest y coordinate. The diagonal of the matrices does not contain useful information and is set to 0. These matrices contain all the information needed to determine the energy required by SUPs from one waypoint to another at one of the ends of the path. Therefore, we can compute $\sum_{i=1}^{N_t} E_{tt,i}$ given a certain sequence.

The GA can be utilized to solve such a TSP. GA is a method inspired by the biological evolution process, where a solution is considered a gene. Each gene could form other genes through mutations and crossovers (in our case, changing the sequence of straight path execution), and only the fittest genes would survive in each generation (in our case, the ones with the minimal energy consumption). An optimized solution can, therefore, be found after a certain number of generations. GA has a strong global search ability, especially in complex optimization problems like the TSP [17,18]. For standard GA implementation, a fitness function $\hat{f}$ is defined as follows:

$$D^{\star} = \hat{f}(\boldsymbol{\Theta}|\boldsymbol{D}) = \sum_{i=1}^{i=N-1} \boldsymbol{D}_{\boldsymbol{\Theta}_i,\boldsymbol{\Theta}_{i+1}}, \tag{41}$$

where $\boldsymbol{D} \in \mathbb{R}^{N\times N}$ is the distance matrix containing the distance between each city. $\boldsymbol{\Theta} \in \mathbb{R}^{1\times N}$ is a vector containing a specific order of cities. $\hat{f}$ is a function parameterized by $\boldsymbol{D}$ and takes $\boldsymbol{\Theta}$ as an input. The GA algorithm $g(\hat{f})$ takes the fitness function $\hat{f}$ and outputs the optimal sequence $\boldsymbol{\Theta}^{\star}$ such that the total distance traveled is minimized.

For our problem, we must modify the fitness function such that it is parameterized by $\boldsymbol{E}_n$ and $\boldsymbol{E}_f$. Note that minimizing the total distance is the same as minimizing the total energy. The modified fitness function $\bar{f}(\boldsymbol{\Theta}|\boldsymbol{E}_f, \boldsymbol{E}_n)$ is defined as follows:

$$\begin{aligned} E &= \bar{f}(\boldsymbol{\Theta}|\boldsymbol{E}_f, \boldsymbol{E}_n) = \sum_{i=1}^{i=N_s-1} E_i, \\ E_i &= \begin{cases} \boldsymbol{E}_{f,\boldsymbol{\Theta}_i,\boldsymbol{\Theta}_{i+1}}, \text{ if } i \text{ is odd}, \\ \boldsymbol{E}_{n,\boldsymbol{\Theta}_i,\boldsymbol{\Theta}_{i+1}}, \text{ if } i \text{ is even}. \end{cases} \end{aligned} \tag{42}$$

Hence, after obtaining $\bar{f}$, we use it as the input to the GA algorithm to obtain the minimum energy for SUPs and the optimal sequence of traversing SPPs:

$$E_{tt}^{\star}, \boldsymbol{\Theta}^{\star} = g(\bar{f}). \tag{43}$$

In this work, we modify the GA so that it can take $\boldsymbol{E}_n$ and $\boldsymbol{E}_f$ as inputs and outputs, respectively, for the optimal SPP execution order. The process of (42) is also visually illustrated in Figure 12. The total required energy is the summation of the energy of SPPs and energy of SUPs as stated in (22).

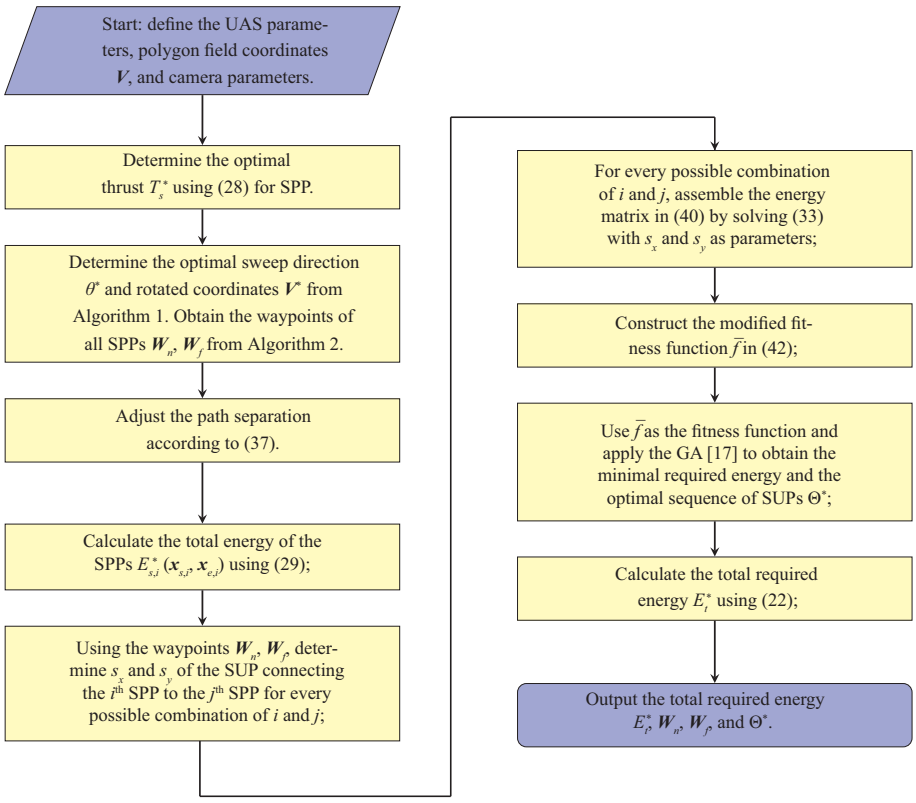

Figure 13 (Color online) Flowchart of the complete algorithm.

The entire process of the proposed algorithm is summarized in Figure 13. The algorithm begins with obtaining the parameters of the Crop field, Photogrammetry, and Flight Dynamics, as listed in Table 2. Then, following Subsection 4.3, the optimal sweep direction of SPPs is determined. Using three types of Dubins paths in Subsection 4.2, solving (33) using the GA in Subsection 4.5 provides the optimal execution sequence and the SUP part of the path.

5 Simulation results

In this section, we provide several simulation results to show the capability of the proposed algorithm. The computer used to run the algorithms is equipped with an i7-1260P 2.10-GHz core, 16 GB RAM, and a 64-bit operating system. In Subsection 5.1, we analyze the algorithm for obtaining the optimal SUP parameter for a given starting point, i.e., solving (33). In Subsection 5.2, we test our algorithm summarized in Figure 13 for four different crop fields. The optimal survey paths are provided in both two-dimensional and three-dimensional forms. To show the performance of the proposed algorithm when compared with the classical brute-force method, Subsection 5.3 compares the computation time and the results when there are different overlapping requirements. The code to generate all the results is available in our Github repository[1)]. We use an open-source solver to implement the GA. The link to the GA solver is also provided[2)].

5.1 Minimum required energy of SUPs

The optimal velocity and radius for SUPs given starting and ending points are obtained using the fmincon function in MATLAB. The point chosen by the program is indicated with a cross on the contour plot. The parameters of the drone are listed in Table 2. Meanwhile, the results of the three test cases are presented in Table 3. As shown in Figures 14 and 15, the energy as a function of velocity and radius is

1) Github link. https://github.com/FSC-Lab/A-Path-Planning-Algorithm-for-A-Crop-Monitoring-Fixed-wing-Unmanned-Aerial-System.

2) GA solver. https://www.mathworks.com/matlabcentral/fileexchange/31818-travelling-salesman-problem-with-genetic-algorithm.

Table 2 Parameters for path-planning simulation

Category	Parameter	Value	Unit
Crop field	g	9.81	$\mathrm{m/s^2}$
	ρ	1.293	$\mathrm{kg/m^3}$
Photogrammetry	f	0.152	m
	ξ_x	0.024	m
	ξ_y	0.036	m
	p	4.8×10^{-6}	m/Pixel
	GSD	0.003	m/Pixel
	o_x	0.85	%
	o_y	0.6	%
Flight dynamics	W	30	N
	s	0.36	$\mathrm{m^2}$
	ζ	4	–
	e	0.775	–
	C_{D_0}	0.03	–
	$C_{L_{\max}}$	1	–
	$n_{\max}$	1.5557	–

Table 3 Results of the optimal turning radius and speed

Start point	End point	Type	v_t^*	r^*	Figure
(0, 0)	(50, 60)	1	13.01	20.29	14(a)
(0, 0)	(10, 60)	2	13.75	17.95	14(b)
(0, 0)	(10, 10)	3	13.99	17.46	15

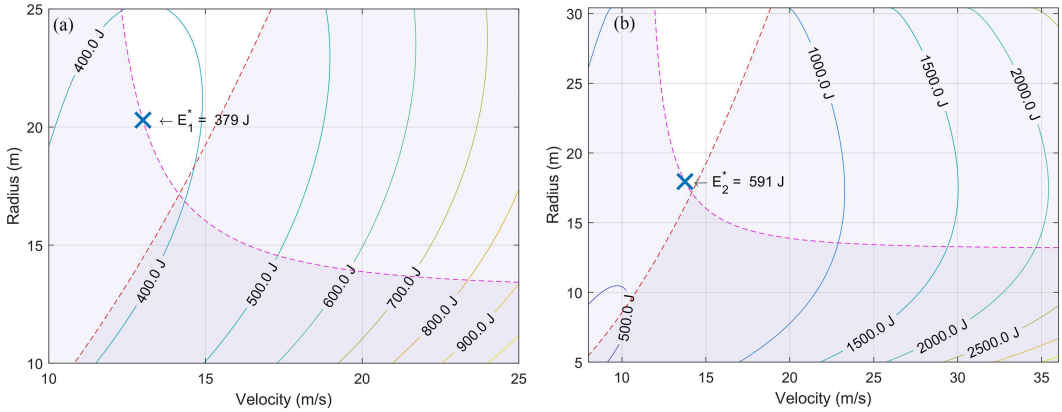

Figure 14 (Color online) (a) Contour plot of energy consumption for a Type 1 U-turn starting from (0, 0) to (50, 60); (b) contour plot of energy consumption for a Type 2 U-turn starting from (0, 0) to (10, 60).

plotted as contour lines. The red contour line shows the location of $n = n_{\max}$, and the magenta contour line shows the location of $C_L = C_{L_{\max}}$. The shaded region is where the combination of v and r exceeds one or both of the constraints.

5.2 Complete optimal paths

The planning algorithm was tested with four different farm field shapes of increasing complexity: a square, a rectangle, a triangle, and a random polygon. All the parameters used in the simulation for path generation are listed in Table 2. The paths generated for the four shapes are shown in Figures 16–19. The perimeter of the crop field is colored green, whereas the flight path generated by the proposed path planner is plotted with a blue line. The flight path starts at the point labeled with an "S" and ends at the point labeled with an "F." The geometry of the crop field and the calculation results of the tests are shown in Table 4.

Moreover, the v_t^*, r^*, and type for each turn were determined by the proposed path planner. Together with the determined v_s^* for the straight paths and $h_{\max}$, the whole trajectory of the crop monitoring

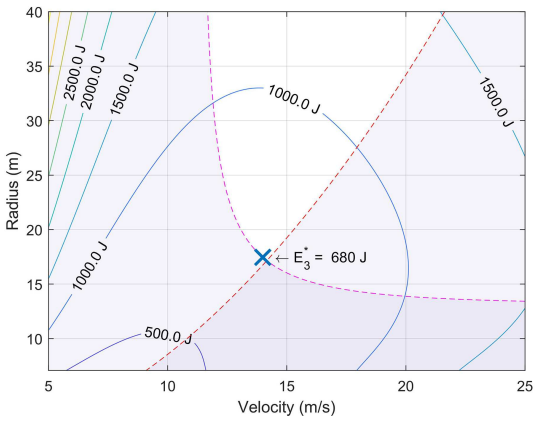

Figure 15 (Color online) Contour plot of energy consumption for a Type 3 U-turn starting from (0, 0) to (10, 10).

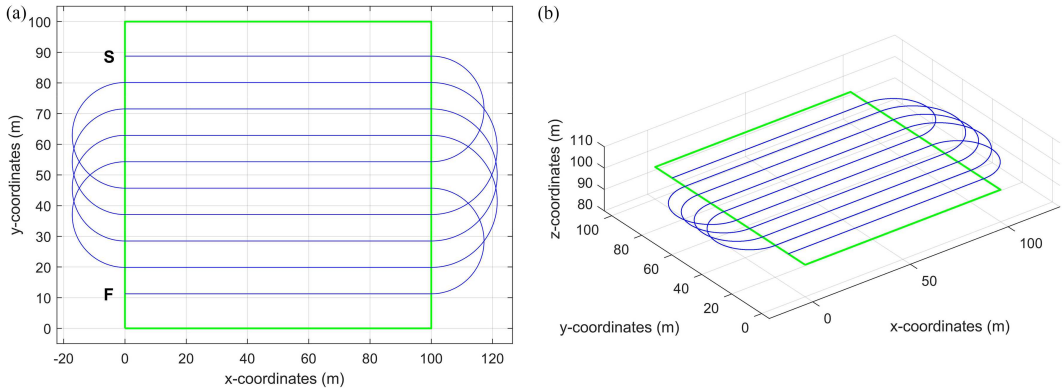

Figure 16 (Color online) Optimal flight path for a square-shaped crop field. (a) In top view; (b) in isometric view.

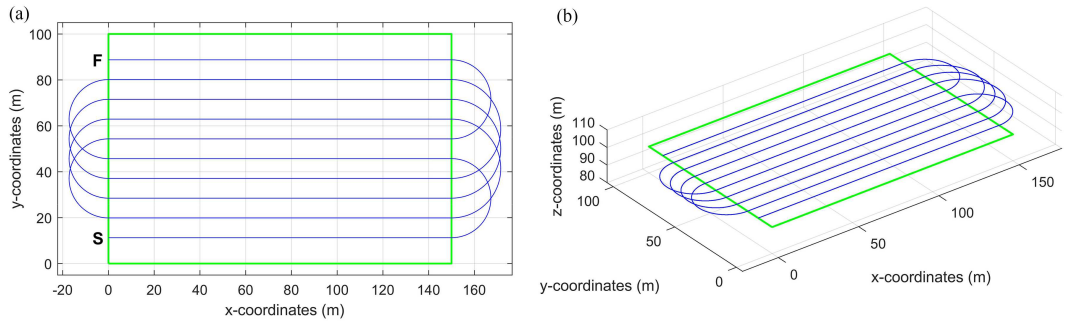

Figure 17 (Color online) Optimal flight path for a rectangular-shaped crop field. (a) In top view; (b) in isometric view.

mission was defined. Table 5 shows the complete optimal flight path specifications for the polygonal crop field in Figure 19.

5.3 Analysis of the algorithm

In this subsection, various experiments were performed on the basis of the crop field shown in Figure 19.

First, to evaluate the superiority of the GA, the computation time for finding the optimal straight-path execution sequence using GA was compared against that of the brute-force method. For the brute-force method, the total turning energy for every permutation of the straight path sequence was calculated, and the minimum was found among all calculated values. For illustration, the crop field in Figure 19 was scaled both up and down such that the scaled crop fields would yield optimal flight paths with an N_s of 2 to 20. The run time for brute force with 12 straight paths was estimated to take more than 10 h. For practical reasons, only $N_s = 2$ to 11 were computed for the brute force method.

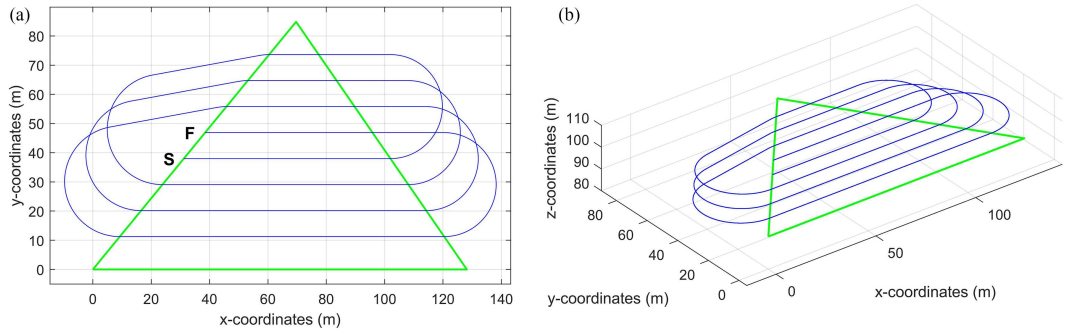

Figure 18 (Color online) Optimal flight path for a triangular crop field. (a) In top view; (b) in isometric view.

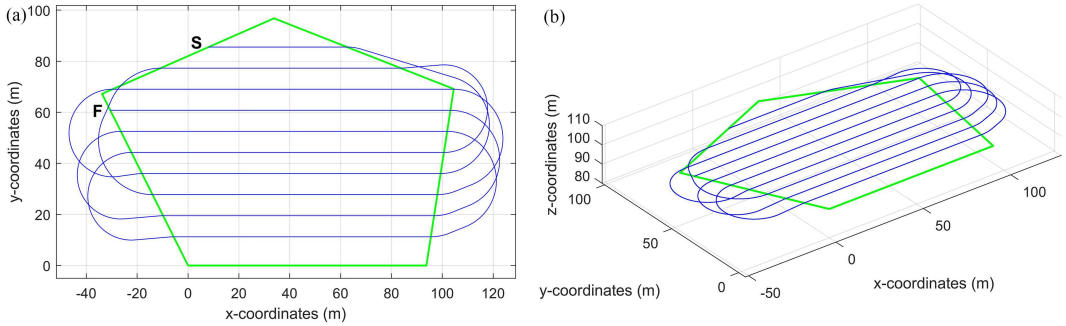

Figure 19 (Color online) Optimal path for a polygonal crop field. (a) In top view; (b) in isometric view.

Table 4 Simulation results

Coordinates of shape vertices	Coordinates after Subsection 4.3	$E_t^\star$ (J)	Total distance (m)
$\begin{bmatrix}0\\0\end{bmatrix}, \begin{bmatrix}100\\0\end{bmatrix}, \begin{bmatrix}100\\100\end{bmatrix}, \begin{bmatrix}0\\100\end{bmatrix}$	$\begin{bmatrix}0\\0\end{bmatrix}, \begin{bmatrix}100\\0\end{bmatrix}, \begin{bmatrix}100\\100\end{bmatrix}, \begin{bmatrix}0\\100\end{bmatrix}$	6348	1528
$\begin{bmatrix}0\\0\end{bmatrix}, \begin{bmatrix}100\\0\end{bmatrix}, \begin{bmatrix}100\\150\end{bmatrix}, \begin{bmatrix}0\\150\end{bmatrix}$	$\begin{bmatrix}0\\0\end{bmatrix}, \begin{bmatrix}150\\0\end{bmatrix}, \begin{bmatrix}100\\150\end{bmatrix}, \begin{bmatrix}0\\100\end{bmatrix}$	8014	2028
$\begin{bmatrix}0\\0\end{bmatrix}, \begin{bmatrix}100\\25\end{bmatrix}, \begin{bmatrix}45\\120\end{bmatrix}$	$\begin{bmatrix}0\\0\end{bmatrix}, \begin{bmatrix}128.2\\0\end{bmatrix}, \begin{bmatrix}69.6\\84.9\end{bmatrix}$	4712	1125
$\begin{bmatrix}0\\0\end{bmatrix}, \begin{bmatrix}72\\24\end{bmatrix}, \begin{bmatrix}105\\90\end{bmatrix}, \begin{bmatrix}36\\120\end{bmatrix}, \begin{bmatrix}-36\\60\end{bmatrix}$	$\begin{bmatrix}0\\0\end{bmatrix}, \begin{bmatrix}93.7\\0\end{bmatrix}, \begin{bmatrix}104.5\\69.1\end{bmatrix}, \begin{bmatrix}33.8\\96.8\end{bmatrix}, \begin{bmatrix}-33.8\\67.2\end{bmatrix}$	7138	1735

Table 5 Flight path specifications for the polygonal crop field

Turn	v_{turn}^* (m/s)	r^* (m)	Type
1	13.31	19.16	1
2	14.16	17.15	2
3	13.25	19.35	1
4	12.81	21.25	1
5	14.16	17.15	2
6	14.16	17.15	2
7	13.25	19.35	1
8	14.16	17.15	2
9	13.25	19.35	1
$v_s^* = 15.44$ m/s			
$h_{\max} = 95.0$ m			

As shown in Figure 20(a), as N_s increased, the computation time using brute force increased exponentially, whereas that of GA only showed a slight increase. Moreover, the optimal total turn energies provided by both algorithms were exactly the same for N_s of 2 to 11 as shown in Figure 20(b), which

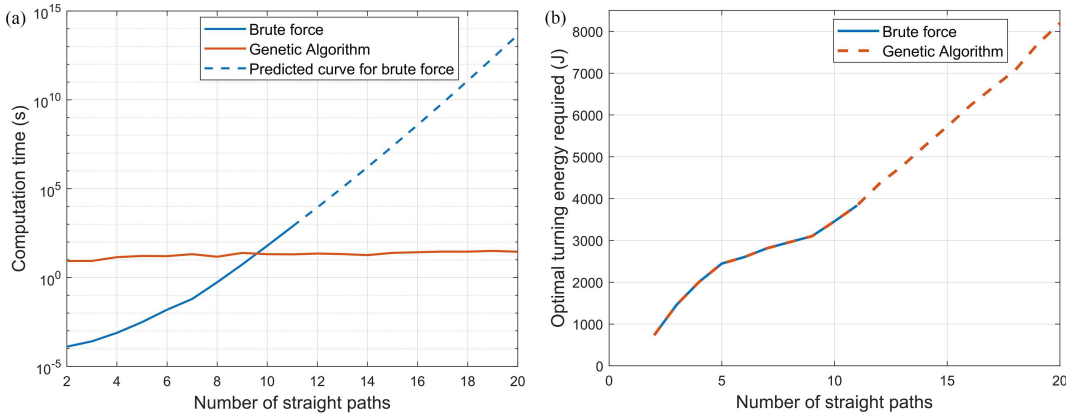

Figure 20 (Color online) (a) Computation time against N_s using different algorithms; (b) optimal turning energy derived by different algorithms.

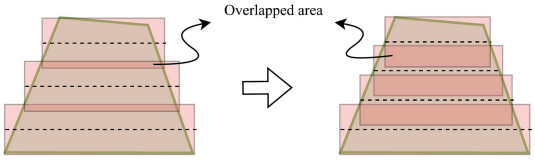

Figure 21 (Color online) Overlap percentage determining the number of straight paths.

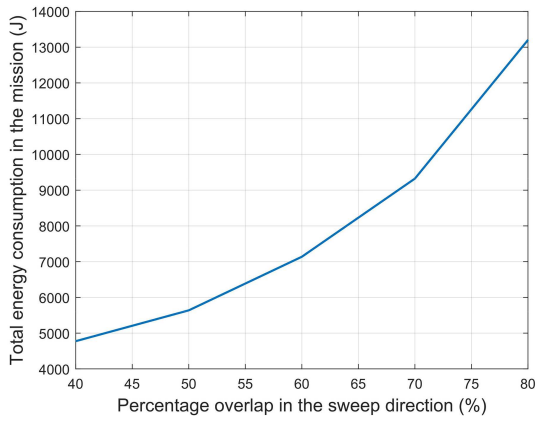

Figure 22 (Color online) Optimal total energy consumed in the mission against the percentage overlap.

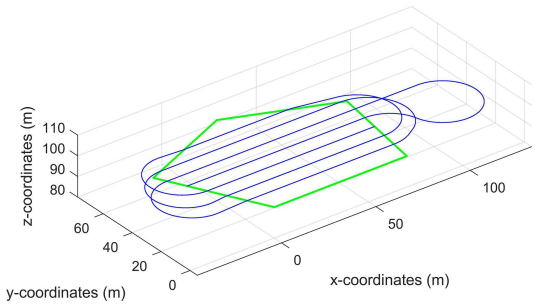

Figure 23 (Color online) Optimal path for the scaled-down polygonal field in Figure 19.

means the GA could find the path sequence with the minimum turning energy.

Second, the relationship between the overlap percentage of the aerial photos and the total energy consumption of the crop monitoring mission was evaluated. Our algorithm produced an optimal flight path with complete aerial coverage. However, varying the percentage overlap affected the quality of the integrated aerial photos. A higher percentage overlap increased the quality, but a higher total energy consumption was induced as N_s was increased, as shown in Figure 21. The comparison was done on the crop field shown in Figure 19. The result is shown in Figure 22.

Lastly, the crop field in Figure 19 was scaled down to generate optimal flight paths with Type 3 turns. Type 3 turns seldom appeared in the optimal flight path. They were generally less efficient as they required the UAS to travel extra distances during a U-turn. However, Type 3 turns was used when $d(\theta^\star)$ was sufficiently small. Figure 23 shows the optimal flight path for the polygonal field in Figure 19, which was scaled down by a factor of 0.75. The specifications of the optimal flight path, such as v^*_{turn}, r^*, and type of the original and the scaled-down versions, are shown in Tables 5 and 6, respectively.

Table 6 Flight path specifications for the scaled-down polygon crop field

Turn	v^*_{turn} (m/s)	r^* (m)	Type
1	13.95	17.53	3
2	14.16	17.15	2
3	13.13	19.80	1
4	14.16	17.15	2
5	13.37	18.96	1
6	14.16	17.15	2
$E^\star_t = 4593$ J			
Total distance = 1060 m			
$v^*_s = 15.44$ m/s			
$h_{\max} = 95.0$ m			

6 Conclusion

In this paper, a path planner was presented to propose an energy-optimal trajectory for a fixed-wing UAS performing a crop monitor mission of a convex polygon crop field. Full coverage was ensured, and energy was optimized by optimizing sweep direction, flight trajectory for straight paths and U-turns, and the sequence of straight path execution. The propulsion system of the fixed-wing aircraft was also modeled to calculate energy consumption with higher accuracy. In future work, we aim to design a path planner that deals with concave crop field shapes. We also plan to extend the current algorithm to a group of UAVs to perform a cooperative survey.

Acknowledgements This work was supported by Natural Sciences and Engineering Research Council of Canada (NSERC) Discovery Grant (Grant No. RGPIN-2023-05148) and 5G-Enabled Trustworthy Common Operational Picture with Edge Server Data Engine (5G-TCOP) (Grant No. 10037356 MN3-026).

References

1 Ponti M, Chaves A A, Jorge F R, et al. Precision agriculture: using low-cost systems to acquire low-altitude images. IEEE Comput Grap Appl, 2016, 36: 14–20
2 Cabreira T, Brisolara L, Ferreira J P R. Survey on coverage path planning with unmanned aerial vehicles. Drones, 2019, 3: 4
3 Franco C D, Buttazzo G. Coverage path planning for UAVs photogrammetry with energy and resolution constraints. J Intell Robot Syst, 2016, 83: 445–462
4 Cabreira T M, Franco C D, Ferreira P R, et al. Energy-aware spiral coverage path planning for UAV photogrammetric applications. IEEE Robot Autom Lett, 2018, 3: 3662–3668
5 Fevgas G, Lagkas T, Argyriou V, et al. Coverage path planning methods focusing on energy efficient and cooperative strategies for unmanned aerial vehicles. Sensors, 2022, 22: 1235
6 Nam L H, Huang L, Li X J, et al. An approach for coverage path planning for UAVs. In: Proceedings of IEEE 14th International Workshop on Advanced Motion Control (AMC), 2016. 411–416
7 Song X, Hu S. 2D path planning with dubins-path-based A algorithm for a fixed-wing UAV. In: Proceedings of the 3rd IEEE International Conference on Control Science and Systems Engineering (ICCSSE), 2017. 69–73
8 Owen M, Beard R W, McLain T. Implementing dubins airplane paths on fixed-wing UAVs. In: Handbook of Unmanned Aerial Vehicles. Dordrecht: Springer, 2015. 1677–1701
9 Wang Z, Liu L, Long T, et al. Enhanced sparse A* search for UAV path planning using dubins path estimation. In: Proceedings of the 33rd Chinese Control Conference, 2014. 738–742
10 Askari A, Mortazavi M, Talebi H A, et al. A new approach in UAV path planning using Bezier-Dubins continuous curvature path. Proc Institution Mech Engineers Part G-J Aerospace Eng, 2016, 230: 1103–1113
11 Babel L. New heuristic algorithms for the Dubins traveling salesman problem. J Heuristics, 2020, 26: 503–530
12 Tripicchio P, Unetti M, D'Avella S, et al. Smooth coverage path planning for UAVs with model predictive control trajectory tracking. Electronics, 2023, 12: 2310
13 Coombes M, Fletcher T, Chen W H, et al. Optimal polygon decomposition for UAV survey coverage path planning in wind. Sensors, 2018, 18: 2132
14 Hawary A F, Chipperfield A J. Routeing strategy for coverage path planning in agricultural monitoring activity using UAV. In: Proceedings of International Congress on Recent Development in Engineering and Technology, 2016
15 Li Y, Chen H, Joo E M, et al. Coverage path planning for UAVs based on enhanced exact cellular decomposition method. Mechatronics, 2011, 21: 876–885
16 Kapanoglu M, Alikalfa M, Ozkan M, et al. A pattern-based genetic algorithm for multi-robot coverage path planning minimizing completion time. J Intell Manuf, 2012, 23: 1035–1045
17 Sonmez A, Kocyigit E, Kugu E. Optimal path planning for UAVs using genetic algorithm. In: Proceedings of International Conference on Unmanned Aircraft Systems (ICUAS), 2015. 50–55
18 Yuan J, Liu Z, Lian Y, et al. Global optimization of UAV area coverage path planning based on good point set and genetic algorithm. Aerospace, 2022, 9: 86
19 Avellar G, Pereira G, Pimenta L, et al. Multi-UAV routing for area coverage and remote sensing with minimum time. Sensors, 2015, 15: 27783–27803
20 Lugo-Cárdenas I, Flores G, Salazar S, et al. Dubins path generation for a fixed wing UAV. In: Proceedings of International conference on unmanned aircraft systems (ICUAS), 2014. 339–346
21 Huang W H. Optimal line-sweep-based decompositions for coverage algorithms. In: Proceedings of IEEE International Conference on Robotics and Automation, 2001. 27–32