

Article

Occlusion-Aware UAV Path Planning for Reconnaissance and Surveillance

Jian Zhang ¹  and Hailong Huang ^{2,*} 

¹ School of Electrical Engineering and Telecommunications, University of New South Wales, Sydney 2052, Australia; jian.zhang4@unsw.edu.au

² Department of Aeronautical and Aviation Engineering (AAE), Hong Kong Polytechnic University (PolyU), Hong Kong, China

* Correspondence: hailong.huang@polyu.edu.hk

Abstract: Unmanned Aerial Vehicles (UAVs) have become necessary tools for a wide range of activities including but not limited to real-time monitoring, surveillance, reconnaissance, border patrol, search and rescue, civilian, scientific and military missions, etc. Their advantage is unprecedented and irreplaceable, especially in environments dangerous to humans, for example, in radiation or pollution-exposed areas. Two path-planning algorithms for reconnaissance and surveillance are proposed in this paper, which ensures every point on the target ground area can be seen at least once in a complete surveillance circle. Moreover, the geometrically complex environments with occlusions are considered in our research. Compared with many existing methods, we decompose this problem into a waypoint-determination problem and an instance of the traveling-salesman problem.

Keywords: UAVs; aerial surveillance; coverage path planning; terrain coverage; path planning



Citation: Zhang, J.; Huang, H. Occlusion-Aware UAV Path Planning for Reconnaissance and Surveillance. *Drones* **2021**, *5*, 98. <https://doi.org/10.3390/drones5030098>

Academic Editor: Carlos Tavares Calafate

Received: 16 July 2021

Accepted: 6 September 2021

Published: 15 September 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The extensive use of Unmanned Aerial Vehicles (UAVs), also known as aerial drones, has recently jumped from military to hobby and professional applications [1]. Complete coverage has become a necessary function for activities including but not limited to border patrol [2,3], search and rescue [4,5], 3D reconstruction [6,7], infraction inspection [8], and surveillance and security [9–12], etc. In general, the coverage problem was first put forward over a 2D grid environment by [13]. We can classify this problem into two main categories based on vehicle movements. Static coverage focuses on the deployment of hovering UAVs to reconnoiter over certain terrains [14,15], while dynamic coverage addresses the reconnaissance and surveillance problem by moving UAVs [16,17].

A survey on computational-intelligence-based UAV path planning can be found in [18,19], which classifies UAV path-planning methods from the aspect of methodology, time domain, and space domain, respectively. In a common reconnaissance and surveillance scenario, a flying vehicle equipped with a downward-facing video camera with a certain visibility angle can monitor targets of interest on the ground, such as vehicles, humans, animals, etc. [20,21]. We can evaluate the quality of surveillance in terms of coverage and resolution [22]. In this case, the low altitude of the traveling path is preferred for a better resolution of the observed region of the terrain. One of the most significant technical challenges is to completely cover a given target area with the minimum number of drone waypoints, which requires every point on the target area to be seen at least once by the onboard camera during one complete surveillance circle. However, these two evaluation terms need to reach a suitable compromise to perform an ideal surveillance duty.

This paper presents two novel path-planning algorithms to address the aforementioned gaps. In contrast to the existing literature, our approach takes both UAV kinematics constraints and camera-sensing constraints into consideration. In the first algorithm, we consider the fixed-wing vehicle case, assuming the vehicle flies at a given altitude with

constant speed and limited turning radius. This Dubins aircraft model is similar to those in paper [11,23,24]. We present a two-phase strategy to solve this surveillance problem. First, an easily implementable estimation algorithm is developed, and the minimum number and locations of waypoints are determined to provide the complete coverage of the target area. The second phase addresses the allocation of the achieved locations over one or more UAVs and creates the shortest paths to reconnoiter the corresponding area of interest. The Dubins paths consist of straight lines and arcs of the circle of a constant radius. To achieve this, regular triangular patterns and the clustered spiral-alternating method are implemented, respectively. Our second algorithm concerns the surveillance problem over geometrically complex environments with varying altitudes and occlusions, such as mountainous terrains and urban regions. In the first stage, the challenge is to find a set of camera locations called the vantage waypoint set to provide full coverage of the area of interest, which can be viewed as a 3D Art Gallery Problem using drones as the observers. In the second stage, one or several UAVs are determined to conduct the full coverage reconnaissance and surveillance duty along individual routes respecting their kinematic constraints in the optimization criterion (the shortest time possible). This variant of the combinatorial traveling-salesman problem is solved by introducing unsupervised learning and Bézier curves.

The rest of the paper is organized as follows. Section 2 presents our approach for the scenario where UAVs fly at a given altitude. Section 3 then presents the proposed algorithm for the case where UAVs can fly at different altitudes. Each of these sections starts with the addressed problem with the necessary background, followed by details of the proposed surveillance solution. The performance of the algorithms is then evaluated using computer simulations. Finally, Section 4 concludes the paper.

2. Surveillance Algorithm at a Given Altitude

This section provides a two-stage approach, which is similar to [11,22,25] but more realistic and efficient. In the first stage, we construct a set of waypoints to be visited by UAVs. The fundamental idea is to create a triangulation of the terrain. This triangulation contains several congruent equilateral triangles, and the length of the sides of the triangles depends on the visibility of UAVs. The second stage aims at producing the shortest smooth trajectory so that all the waypoints can be visited by UAVs. As we consider fixed-wing UAVs, the popular Dubins car model is used. Then, the constructed trajectory has several straight-line segments and arcs. In the following subsections, we first formally state the problem of interest and then present our approach in detail.

2.1. Problem Statement

Consider a surveillance problem over one or multiple disjoint surface areas. We have one or a fleet of fixed-wing UAVs. For simplicity, all the UAVs are identical. They fly at a constant speed v and their angular velocity is u . In this section, we assume that the UAVs fly at a given altitude. We describe the UAV motion by the following well-known Dubins car model [26], which is often used for aircraft, UAVs and missiles [27–30]:

$$\begin{aligned}\dot{x}(t) &= v(t) \cos \theta(t) \\ \dot{y}(t) &= v(t) \sin \theta(t) \\ \dot{\theta}(t) &= u(t)\end{aligned}\tag{1}$$

where θ is the vehicle's heading.

Let (x_g, y_g) denote the Cartesian coordinates of a point on the ground. Let $z_g = F(x_g, y_g)$ be the perpendicular elevation of the point (x_g, y_g) on the terrain. Moreover, let \mathcal{G} denote a given Lebesgue measurable region [31] on the ground with zero elevation, i.e., $z_g = 0$. In addition, this region \mathcal{G} has piecewise boundary. Our goal is to survey the region \mathcal{G} by one or more UAVs. Let Z_{max} and Z_{min} be the given constants representing the maximum

and minimum altitudes of UAVs. Any vantage point (x, y, z) must make the following constraints hold:

$$(x, y) \in \mathcal{G}, z \in [Z_{min}, Z_{max}], 0 < Z_{min} < Z_{max}. \tag{2}$$

Each UAV carries a fixed EO/IR camera, which is downward-facing to the ground with a certain visibility angle $0 < \alpha < \pi$. The sensing process is based on pinhole perspective projection [32], and the quality of the resolution is influenced by both the distance to the object and the physical parameters. In particular, if we set the camera’s center as UAV’s position (x, y, z) , the UAV can only see points along the z axis with a limited angle of view, so that a cone-shaped field of view (FOV) with radius

$$R := z \cdot \tan\left(\frac{\alpha}{2}\right) \tag{3}$$

is constructed (see Figure 1). A point is invisible if it falls out of the FOV.

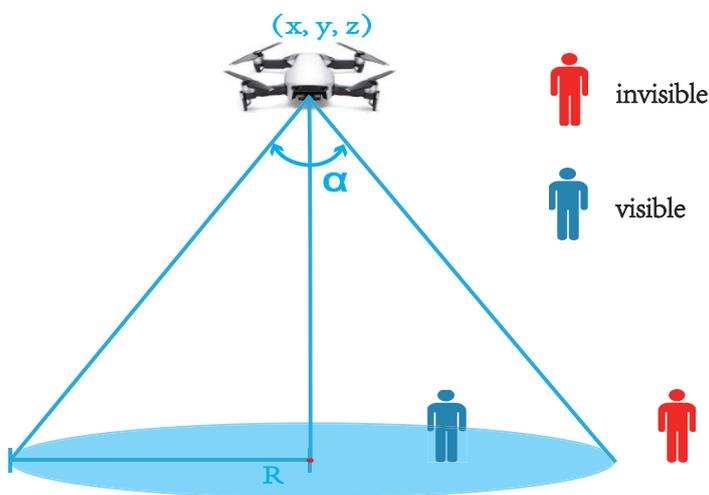


Figure 1. The visibility cone. Object within the FOV can be seen by the drone.

As aforementioned, a surveillance mission should deliver a coverage of the area of interest with a satisfactory resolution. Moreover, complete coverage means that every point of the region \mathcal{G} is seen at least once in a complete surveillance circle of UAVs. One of the most significant technical challenges is how to completely cover the target region by the minimum number of waypoints. The visibility cone enlarges when the UAV flies higher, but the resolution of the onboard recording is worse. We can choose a certain altitude that can achieve the required resolution. Here, we select the lowest possible altitude z , i.e., $Z_{min} < z < Z_{max}$ for the best resolution. At this altitude, the generated waypoint set, which provides complete coverage of the target area with a resolution that cannot be further better, is preferred to conduct an ideal surveillance mission.

Definition 1. The complete coverage means constraints (2) and (4) are satisfied and every point on the target area \mathcal{G} can be seen at least once in a complete surveillance circle.

In the herein-addressed surveillance problem, we consider one waypoint location $p_i \in \mathbb{R}^3$ from vantage waypoint set

Let $P = \{p_1, \dots, p_n\}$ denote the set of the vantage waypoints. Here, $p_i \in \mathbb{R}^3$ is a vantage waypoint, which should be visited by a UAV. We assume that when a UAV is within δ distance from point p_i , a record such as the photo/video can be taken. Formally, when a UAV reaches any point p_i^* satisfying $\|p_i^*, p_i\| \leq \delta$, where $\|\cdot, \cdot\|$ gives the standard Euclidean distance, we say the vantage point p_i is visited successfully.

We need some more symbols to complete the problem statement. Let q_{ij} denote the distance between waypoints i and j . Let q_i be the minimum distance between a waypoint i and the terrain. For collision avoidance, the following constraints must be met:

$$q_{ij} \geq c_1, \quad q_i \geq c_2 \quad (4)$$

where $c_1 > 2\delta > 0$ and $c_2 > \delta > 0$ are given safety margins.

With the above models, the problem of interest is to find the trajectory to visit the δ -neighborhood of all assigned vantage points. Such a problem involves the optimization of the sequence of visits, which can be regarded as an instance of the well-known Dubins vehicle Traveling Salesman Problem (DTSP). As discussed earlier, the final trajectory for a UAV is a sequence of straight-line segments and circular arcs with the minimum radius that traverses the δ -neighborhood of all the vantage points.

In DTSP, the shortest path between two configurations must be one of the six possible combinations: LSL, LSR, LRL, RSL, RSR, and RLR, where 'S' means going straight and 'L' and 'R' denote left and right turn with the minimum turning radius, respectively, see Figure 2.

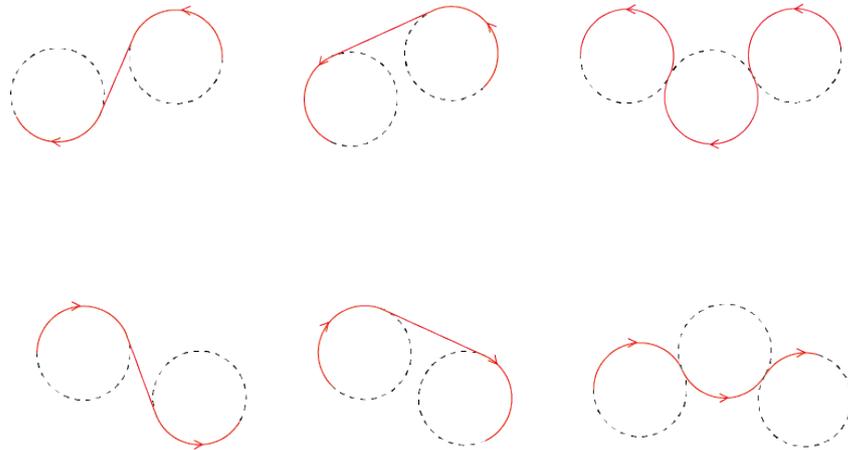


Figure 2. Six cases of Dubins curves.

2.2. Surveillance Algorithm

In this subsection, we present the proposed trajectory construction strategy. This strategy is based on a decomposition of the surveillance problem into a complete coverage problem and a DTSP. This strategy finds the waypoints to fully cover the area of interest first, and then plans trajectory along with these waypoints, such that the length of the path to visit all the waypoints is minimized.

2.2.1. Stage One: Waypoint Generation

In the first stage of the strategy, a method similar to [9] to estimate the minimal number of waypoint locations with coordinates on a given plane $z_p = z$ that is parallel to the ground is used. Then, the strategy finds the minimum altitude z to sense the achieved 2D coordinate.

The constructed triangulation consists of equilateral triangles, and the length of the side of all the triangles is $\sqrt{3}R$, where R is the radius of the vision-cone as defined in (3). We denote the direction of a side of a triangle with respect to a certain direction (such as the x -axis) as the angle of the triangle and we use $\lambda \in [0, \frac{\pi}{3})$ to represent this angle. Additionally, we denote (x_0, y_0) as the coordinate of one of the vertices (see Figure 3a). Then, a triangulation can be represented by $\Delta(\lambda, x_0, y_0)$. Moreover, in any triangle, the line segments connecting the center point, denoted by C , and the vertices divide the triangle into three congruent Voronoi cells. For instance, we use different colors to indicate these cells in Figure 3b. In an equilateral triangle with side length $\sqrt{3}R$, the distance between

any two points in the same Voronoi cell is within R the radius of the FOV. It is obvious that the drone located at the same Voronoi cell to any point of the area of interest can cover at least this point of the area.

Given similar environments in [9], the minimum number and the positions of waypoints are determined while decreasing the altitude z by 1 m each time from $z = Z_{min} = 120$ m, and λ, x_0, y_0 are randomly generated. Referring to the comparison results between [9,33], it is apparent that the method of [33] needs more waypoints than [9] to cover the same area at a similar altitude. So, the proposed method applies [9] for waypoint generation.

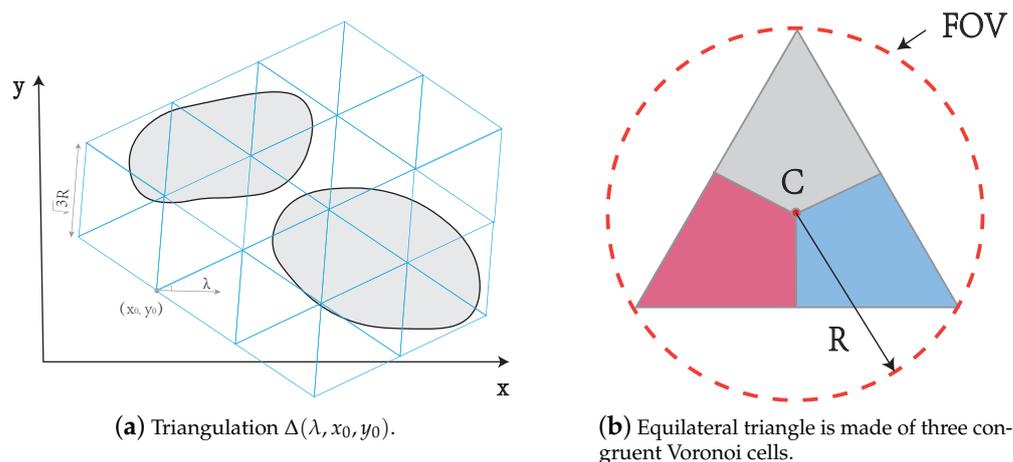


Figure 3. (a) Triangulation consists of equilateral triangles; (b) Congruent Voronoi cell.

2.2.2. Stage Two: UAV Path Planning for Surveillance

We consider that the UAVs fly along Dubins paths. DTSP, as an extension of the Traveling Salesman Problem (TSP), assumes the salesman travels along the Dubins path, where the travel cost is proportional to length of the path [34]. The algorithms to derive the Dubins path can determine the shortest path connecting vantage waypoints generated in the first phase.

We briefly discuss two algorithms to address the DTSP problem, which are the alternating algorithm [34] and the spiral algorithm [25]. The alternating algorithm is an approximation algorithm to solve DTSP with given lower and upper bounds on solution quality. This algorithm computes the sequence of visits first uses the optimal solver for Euclidean TSP (ETSP) and then connects two waypoints by either an alternating straight-line and Dubins arc segments. The spiral algorithm is another popular DTSP surveillance algorithm. It links the chain of the convex hull of the given sets of waypoints. The shape of the resulting path resembles a spiral respecting the turning radius constraints.

When the waypoints are sparse, i.e., when two waypoints are far enough with respect to the turning radius, the alternating algorithm computes the shortest path. By contrast, when the waypoints are dense, the spiral algorithm has been shown to be superior. In addition, the spiral algorithm is also more suitable for surveillance duty at low altitudes. The clustered spiral-alternating algorithm [35], which is actually a combination of the above two algorithms, is used in this paper to improve the effectiveness and efficiency of surveillance. To apply this algorithm we need to cluster the waypoints manually or by other methods. After clustering, waypoints in a cluster are dealt by the spiral algorithm. After this, the spiral paths for different clusters are combined, and the alternating line segments are replaced with Dubins segments to form the shortest path. From the observed result of total path length, the clustered spiral-alternating algorithm is always shorter than the original spiral algorithm in clusters. Compared to the alternating algorithm, the clustered spiral-alternating algorithm has fewer sharp turns.

The proposed scheme is also applicable when the target area consists of multiple disjointed areas (no overlaps). One of the most straightforward ways is to assign as many UAVs as the disjoint areas. The merits of the method include its simplicity, adaptability, and robustness against restricted communication among UAVs.

2.3. Simulation Results

In this section, computer simulation results are presented to demonstrate the effectiveness of the proposed approach.

2.3.1. Single-Area Surveillance

We first consider the vantage waypoints in a $700 \times 900 \text{ m}^2$ single area as shown in Figure 4a,b, which demonstrates the Voronoi diagram of the waypoints.

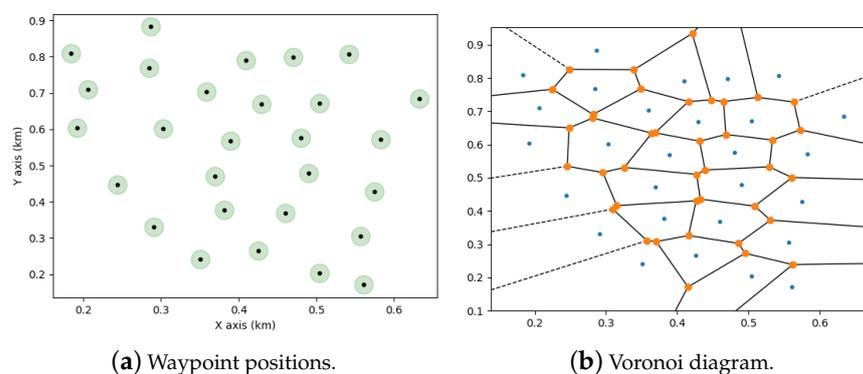


Figure 4. Single-area surveillance scenario.

Figure 5 shows the single UAV surveillance path generated by the spiral algorithm [34], the alternating algorithm [25], the genetic algorithm [17], Grey Wolf Optimizer (GWO) algorithm in [36], and the clustered spiral-alternating algorithm [35], respectively. We evaluated the selected clustered spiral-alternating algorithm with mentioned four algorithms in terms of path length and number of sharp turns. The path generated by clustered spiral-alternating algorithm Figure 5e is shorter than the spiral algorithm (SA) Figure 5a and Grey Wolf Optimizer (GWO) algorithm Figure 5c, and fewer sharp turns than the alternating algorithm (AA) Figure 5b and genetic algorithm (GA) Figure 5d. The straight-line segments and the Dubins segments are represented by yellow and red segments, respectively. The average execution times for the above methods are shown in Table 1. The average execution times for the above methods are shown in Table 1. The SA and AA are faster than the GWO, GA, and the proposed method. However, the proposed method has a far shorter path length than the SA method and fewer sharp turns than the AA method. The proposed method is available as an option when there is a tradeoff between the execution time and optimal path.

Table 1. Average execution times (seconds) for the different algorithms.

Algorithm	Time (s)
SA	3.21
AA	3.69
GWO	16.14
GA	10.36
Proposed	4.84

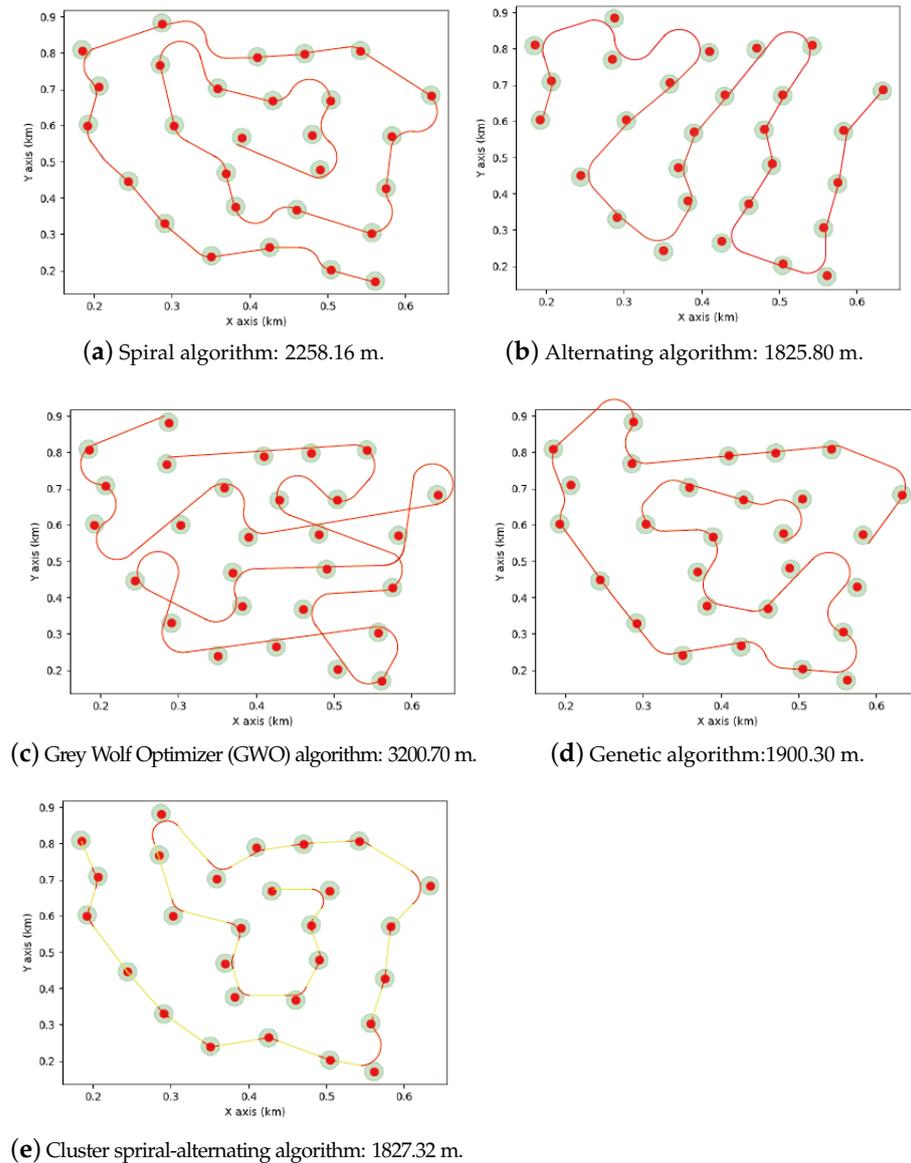


Figure 5. UAV surveillance trajectory—Single UAV.

2.3.2. Multiple Disjoint Areas Surveillance

The multiple disjoint surface area in a $600 \times 600 \text{ m}^2$ terrain is shown in Figure 6. In the first stage, the minimum number and position of the waypoints of our region of interest are generated and shown in Figure 6a, and the δ -neighborhood of each individual vantage waypoint location is in Figure 6b. Then, the path is achieved using clustered spiral-alternating algorithm and Dubins curves. For our results in Figure 6c, two clusters are assigned to two UAVs, respectively. In addition, the coverage of the area of interest is achieved when both finish their surveillance circle. It makes the quick parallel surveillance possible and the subsequent task allocations simple.

In the single UAV scenario, we can use only one drone to carry out the surveillance. With the consideration of computation efficiency, we decide to keep one obtained path and regenerate the other cluster's path. The final surveillance planning with Dubins curves by the proposed algorithm shows in Figure 6d. The simulation result demonstrated the validation of the algorithm.

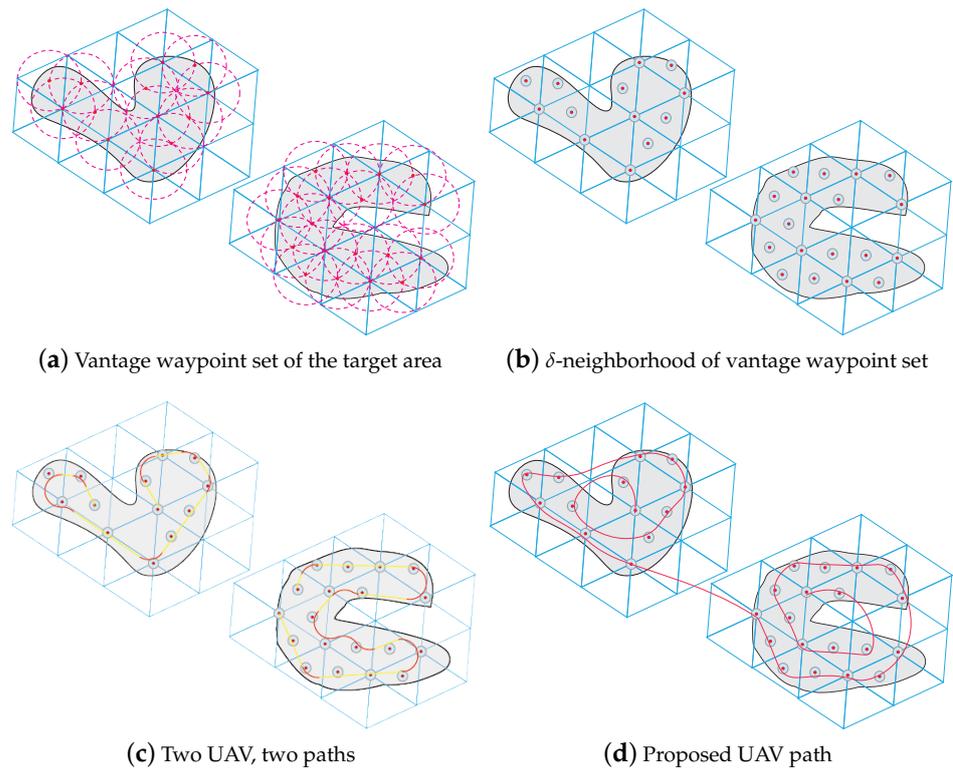


Figure 6. Proposed UAV surveillance in multiple disjoint surface area.

3. Surveillance Algorithm at Different Altitudes

This section considers a more challenging and realistic situation involving surveillance in geometrically complex environments, such as mountainous terrains and urban regions. Because the video camera can only see the points within its cone-shaped field of view (FOV), the FOV can be reduced when facing any kind of obstacle. Similar to our previous section, we address the mentioned problems separately, decoupling them into the waypoint-generation part and trajectory determination part.

3.1. Problem Statement

We consider the dynamic coverage problem in the target area, where UAVs keep flying through each pre-defined waypoint during surveillance and take video. The vehicle’s kinematic model at the point $p = (x, y, z)$ can be described as follows:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} = v \begin{bmatrix} \cos \theta \cos \psi \\ \sin \theta \cos \psi \\ \sin \psi \end{bmatrix} \quad (5)$$

where θ is the turning angle and ψ is the pitch (climb/dive) angle. The state of our drone is $s = (p, \theta, \psi)$.

Several assumptions described in the last section also apply to the model of the terrain. In the addressed reconnaissance and surveillance scenario, our UAV is equipped with a downward-facing video camera with a certain visibility angle. It can monitor the relatively small targets of interest on the ground with the required level of details within its FOV. Figure 7 is the onboard camera with a given visibility angle $0 < \alpha < \pi$. The drone at (x, y, z) can only see points (x_g, y_g, z_g) that are inside the cone-shaped field of view (FOV) of radius on the target area

$$\begin{aligned} R(z_g) &:= (z - z_g) \cdot \tan\left(\frac{\alpha}{2}\right), \\ z &> z_g. \end{aligned} \quad (6)$$

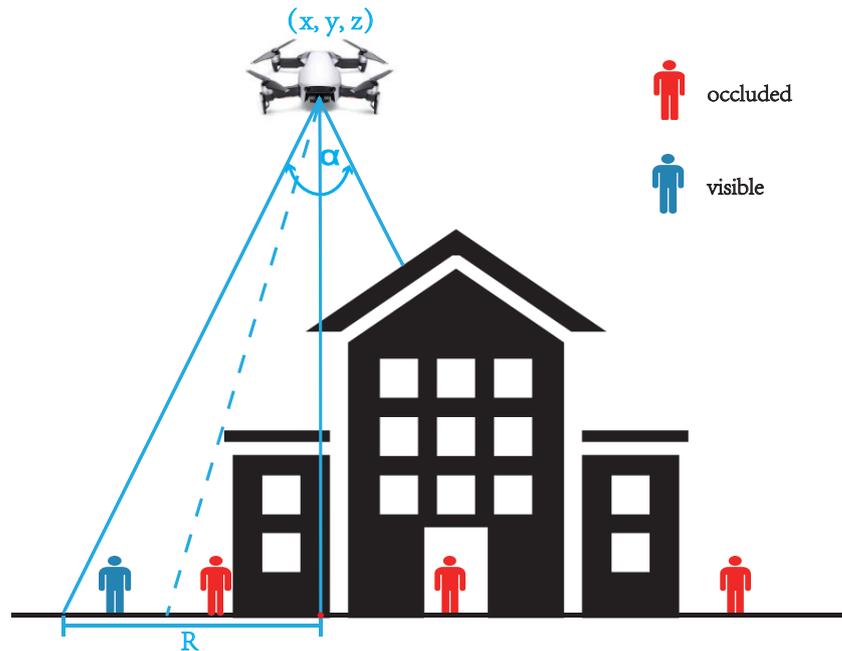


Figure 7. Occlusion effects on camera sensing.

The problem becomes to find the trajectory that goes through the δ -neighborhood of all assigned vantage points, which take the optimization of the sequence of visits into account, i.e., a variant of the combinatorial traveling-salesman problem. The final trajectory is connected by a sequence of piecewise cubic Bézier curves that traverse the δ -neighborhood of all assigned vantage points. A n degree Bézier curve can be parametrized as:

$$\Gamma(t) = \sum_{i=0}^n \mathbf{P}_i b_{i,n}(t) \tag{7}$$

where \mathbf{P}_i stands for the i -th control point, and $0 \leq t \leq 1$. $b_{i,n}(t)$ is named the Bernstein polynomial and defined as follows:

$$b_{i,n}(t) = \binom{n}{i} t^i (1-t)^{n-i} \tag{8}$$

and $0 \leq t \leq 1$. The binomial coefficient is given by

$$\binom{n}{i} = \frac{n!}{i!(n-i)!} \tag{9}$$

The used cubic Bézier curve is defined by four control points $(\mathbf{P}_0, \mathbf{P}_1, \mathbf{P}_2, \mathbf{P}_3)$, and can be expanded as

$$\Gamma(t) = \mathbf{P}_0(1-t)^3 + 3\mathbf{P}_1t(1-t)^2 + 3\mathbf{P}_2t^2(1-t) + \mathbf{P}_3t^3. \tag{10}$$

As the final trajectory Γ is closed and smooth curve, which consists of n Bézier curves, two consecutive curves Γ_i and Γ_j with control points $(\mathbf{P}_0^i, \mathbf{P}_1^i, \mathbf{P}_2^i, \mathbf{P}_3^i)$ and $(\mathbf{P}_0^j, \mathbf{P}_1^j, \mathbf{P}_2^j, \mathbf{P}_3^j)$ should be connected at the same endpoint:

$$\mathbf{P}_3^i = \mathbf{P}_0^j \tag{11}$$

And the tangents \mathbf{t}_a^i and \mathbf{t}_b^i with length $|t_a^i|$ and $|t_b^i|$ of Γ_i and Γ_j must point to the same direction to connect a sequence of Bézier curves into a smooth path:

$$l_a^j t_b^i = l_b^j t_a^i \quad (12)$$

where

$$\mathbf{t}_a^i = \mathbf{P}_1^i - \mathbf{P}_0^i, \quad \mathbf{t}_b^i = \mathbf{P}_3^i - \mathbf{P}_2^i, \quad l_a^i = |t_a^i|, \quad l_b^i = |t_b^i|. \quad (13)$$

Compared with the constant speed Dubins, the multi-motor model can slow down turning and accelerate on fairly straight paths. For the UAV model, the multi-rotor UAV is preferred to the Dubins model with limited curvature. Therefore, the goal is to find the fastest traveling route for UAV under the limit of maximum speed and acceleration, instead of finding the shortest path under the limit of Dubins speed. In this case, this section does not consider minimizing the length of the track, but considers the expected travel time of the path.

3.2. Surveillance Algorithm

In this section, first, we try to find the vantage waypoint set that can completely cover the area of interest, and then search for a smooth trajectory along with these locations at different altitudes as quickly as possible, so as to minimize the completion time of visiting all the locations. We decompose the surveillance problem into a drone version of the 3D Art Gallery Problem and an instance of the combinatorial traveling-salesman problem.

3.2.1. Stage One: Vantage Waypoint Set Generation

The problem of finding the set of vantage waypoints can be regarded as a drone version of the 3D Art Gallery Problem. It has been proved to be NP-hard [37], and the approximation approach is always employed. In this section, we use a method similar to [38] to estimate the minimal number of the waypoints by method in [39], and the vantage waypoint set $P = \{p_1, \dots, p_n\}$, $p_i \in \mathbb{R}^3$ in two major steps. In the first step, the 2D coordinates (x_i, y_i) of each vantage waypoints are determined. With the achieved 2D coordinates (x_i, y_i) , the optimal altitudes z_i are in the second step.

We assumed the terrain \mathcal{G} is a polygon with n vertices, and $\mathcal{G}_1, \dots, \mathcal{G}_i$ are obstacles inside \mathcal{G} with n_1, \dots, n_i vertices, respectively. Let $\mathcal{E}_1, \dots, \mathcal{E}_i$ be inside polygons of $\mathcal{G}_1, \dots, \mathcal{G}_i$, respectively. In addition, each \mathcal{E}_i has same number of vertices n_i as \mathcal{G}_i . We can consider \mathcal{E}_i as and “top” face, and \mathcal{G}_i as the corresponding “bottom” face of each obstacle polyhedron model. \mathcal{P} is obtained from \mathcal{G} without $\mathcal{G}_1, \dots, \mathcal{G}_i$, which is a non-convex polygon with i “holes”. Let $c > 0$ be a given constant, we assume that the relative even area’s altitude $|z_g| \leq c$, and $c + c_2 \leq Z_{min}$ holds. In addition, let $\hat{d}_e := \max\{d_e, \hat{d}_l\}$, d_e is the maximum distance between (x_i, y_i) to the corresponding vertices of \mathcal{E}_i , and b is the maximum altitude of the terrain points corresponding to the \mathcal{E}_i and its side quadrilaterals. Let d_l be the maximum length of the triangulation triangles sides whose vertex is one of the two end points. The developed Vantage Waypoint Set Generation algorithm can be found in Algorithm 1.

The number of waypoints in vantage waypoint set P is less than or equal to $\frac{n+2n_1+\dots+2n_k+2k}{3}$, because the color with the minimum number of vertices will be selected after painting all vertices of the triangulation $\hat{\mathcal{T}}$ by 3-coloring method (see from line 5–6 of Algorithm 1, vertices in $\hat{\mathcal{T}}$ is $n + 2n_1 + \dots + 2n_k + 2k$). For any waypoint $p_i = (x_i, y_i, z_i)$ generated by Algorithm 1, $(x_i, y_i) \in \mathcal{G}$ by line 2, and lines 9–12 guarantee $Z_{min} \leq z_i \leq Z_{max}$. As a result, (2) still hold. According to the construction (see line 1–4), every triangle of $\hat{\mathcal{T}}$ can find a waypoint located at one of its vertices. Moreover, any point outside the obstacles $\mathcal{G}_1, \dots, \mathcal{G}_i$ falls in one of the triangles of triangulation \mathcal{T} , the drone located at one of the three vertices of this particular triangle can see this point with the altitude determined by line 10–12. Furthermore, for the “side” face of each obstacle, there exists a waypoint located at one

of the four vertices with the altitude determined by line 10–11 can see any point of this “side” face. In addition, it is apparent that any point of a “top” face is visible to the drone located at the altitude determined by line 12. Consequently, the terrain \mathcal{G} can be completely covered by $\frac{n+2n_1+\dots+2n_k+2k}{3}$ waypoints or fewer.

Algorithm 1 Vantage Waypoint Set Generation

```

1: procedure STEP I
2:   Construct polygon  $\mathcal{P}$  without “holes” by  $i$  non-intersect diagonals with  $n + n_1 + \dots + n_i + 2i$  vertices
3:   Cut  $\mathcal{P}$  into triangulation  $\mathcal{T}$ , whose vertices are  $\mathcal{P}$ 's, and sides are either  $\mathcal{P}$ 's or its diagonals.
4:   Build the dual graph of  $\hat{\mathcal{T}}$  by enlarging  $\mathcal{T}$  by the vertices of  $\mathcal{E}_i$ 
5:   Paint the vertices of the triangulation  $\hat{\mathcal{T}}$  by 3-coloring method in [39]
6:   The minimum number of vertices subset of the three is selected as the 2D coordinates  $(x_i, y_i)$  of the vantage waypoint set
7: end procedure
8: procedure STEP II
9:   while  $z_i \leq Z_{max}$  do
10:    if  $(x_i, y_i)$  is not a vertex of any polygons  $\mathcal{E}_i, \mathcal{G}_i$  then
11:       $z_i := \max \left\{ Z_{min}, c + \frac{d_l}{\tan(\frac{\alpha}{2})} \right\}$ 
12:    else if  $(x_i, y_i)$  is a vertex of some polygons  $\mathcal{E}_i, \mathcal{G}_i$  then
13:       $z_i := \max \left\{ Z_{min}, b + c_2 + \frac{\hat{d}_i}{\tan(\frac{\alpha}{2})} \right\}$ 
14:    else
15:       $z_i := \max \left\{ Z_{min}, a + c_2 + \frac{\hat{d}_i}{\tan(\frac{\alpha}{2})} \right\}$ 
16:    end if
17:  end while
18: end procedure
19: return vantage waypoint set  $P = \{p_1, \dots, p_n\}, p_i = (x_i, y_i, z_i)$ 

```

3.2.2. Stage Two: UAV Path Planning for Surveillance

As mentioned in the previous section, the final trajectory includes the δ -neighborhood of the set of vantage waypoints to reduce the completion time. We do not directly tackle the coverage problems by visiting the neighborhoods. Regarding the kinematic constraints of the UAV, the variant of the combinatorial traveling-salesman problem is solved by introducing Self-Organizing Map (SOM) and Bézier curves.

As we assumed that the UAV will return to the initial location p_1 after the complete reconnaissance tour, $\delta = 0$ for the initial location. With the vantage waypoint set $P = \{p_1, \dots, p_n\}, p_i \in \mathbb{R}^3$ generated above, and the given initial location p_1 and δ , we can determine the final trajectory Γ as a sequence $\Sigma = (\sigma_1, \dots, \sigma_n)$ of Bézier curves $\Gamma_i, 1 \leq i \leq n$. The final trajectory $\Gamma = (\Gamma_{\sigma_1}, \dots, \Gamma_{\sigma_n}), 1 \leq \sigma_i \leq n$, we need to minimize the estimation of the travel time $E(\Gamma)$, which can be determined from (10). To simplify the model, we employ the Model Predictive Controller (MPC) for path following, and the vertical and horizontal movements of the drone are individually considered. We denote $a_{ver}, v_{ver}, a_{hor}, v_{hor}$ as maximal vertical and horizontal accelerations and speeds, respectively. We also assume that the initial and final velocity of the drone is zero, so that the drone will start from the initial location p_1 with zero velocity and return back to it in the end. The estimation of the travel time $E(\Gamma)$ and the profile of the velocity can be computed by the method in [40] by maximum possible tangent acceleration $a_{tan} = \sqrt{a_{hor}^2 - a_{rad}^2}$, a_{rad} is the radial acceleration.

The adaptation can be considered to be a movement of the waypoint locations towards the alternate location s_p and a new location of each adapted waypoint location v becomes v' and it follows the standard SOM learning [41].

$$v' = v + \mu f(\sigma, d)(s_p - v) \quad (14)$$

where μ is the learning rate, σ is the learning gain, d is the distance of v from the winner waypoint location v^* , and $f(\sigma, d)$ is the neighboring function.

$$f(\sigma, d) = \begin{cases} e^{-\frac{d^2}{\sigma^2}} & \text{for } d < 0.2M \\ 0 & \text{otherwise} \end{cases} \quad (15)$$

where M is the current number of neurons in the SOM.

The surveillance planning algorithm will stop the adaptation if $i \leq i_{max}$ or v^* are negligibly close to their respective s_p , or all winner waypoint locations are inside the δ -neighborhood of the respective initial waypoint location. Local Iterative Optimization (LIO) [42] is a procedure that optimizes the whole trajectory locally, e.g., it can consecutively optimize θ_i , ψ_i , l_a^i , and l_b^i in the loop with waypoint v_{i-1} , v_i , and v_{i+1} . The reason (5) can optimize variables θ_i , ψ_i , l_a^i independently is tangent vector t_a^i and t_b^{i-1} implicitly satisfy the smooth constraint (12). t_a^i and t_b^{i-1} are related to the same waypoint v_i .

Otherwise, go to Step 3. An intersection of the straight-line segment (s_p, p) with the sphere in \mathbb{R}^3 shaped δ -neighborhood of p is used to determine the alternate location s_p towards which the network is adapted instead of p to save the travel time.

We can easily extend the described mechanism into multiple UAV versions by two principal methods:

- 1 Once the surveillance path for single UAV is generated, we can distribute multiple UAVs travel along the same path as the single UAV, but with different initial position to avoid collisions. To avoid collisions, the initial deployment of UAVs must be coordinated with the drone's velocity and the length of the path, by, for instance, evenly spacing the appropriate number of UAVs along the determined trajectory. Thereafter, each UAV can perform its surveillance duty independently without further coordination. This method can markedly reduce the surveillance circle or duration, and significantly increase the frequency and intensity of surveillance.
- 2 The vantage waypoint set can be partitioned into several subsets, and dedicated UAV(s) can traverse through each subset independently. In the case of multiple UAVs, we may use the aforementioned method to perform collision-free monitoring tasks.

We also analyze the complexity of the algorithm. In each learning epoch, the computational complexity depends on the number of waypoints n and the number of neurons in the SOM network M . Notice the SOM is a two-layered neural network, whose input layer is the locations of vantage waypoints $P = \{p_1, \dots, p_n\}$, $p_i \in \mathbb{R}^3$, and output layer is an array of adapted waypoint locations $\mathcal{N} = \{v_1, \dots, v_M\}$. As the algorithm regenerates the loop N with winner waypoints between $n - 2n$ (see from line 8 of Algorithm 2), the complexity of the path-planning procedure is $O(n^2)$.

3.3. Simulation Results

To verify the effectiveness of the proposed algorithm, two simulation scenarios are given in this section. The target area is 20 m by 20 m terrain with $i = 3$ random shaped obstacles are shown in Figure 8a, and the obstacles have $n_1 = 3$, $n_2 = 4$, $n_3 = 4$ vertices, respectively. The average execution times for the above methods are shown in Table 2, and the average velocities and coverage times for the different algorithms are shown in Table 3. Furthermore, the simulation is conducted with the following parameters in Table 4.

Algorithm 2 UAV Path-Planning Algorithm

```

1: Create the loop  $\mathcal{N}$  with  $n$  waypoint locations around the initial location  $p_1$ 
2: Set the learning gain  $\gamma = 12.41n + 0.6$ , the learning rate  $\mu = 0.5$ , and the gain decreasing rate  $\eta = 0.1$ . Set the epoch counter  $i = 1, i_{max} = 100$ .
3: while the termination condition has not been reached do
4:   for  $p \in \Pi(P)$  do
5:     for each learning epoch do
6:       determine  $v^*$  and  $s_p$ 
7:       Adapt  $v^*$  and its neighbors towards  $s_p$  using (7)
8:       remove all non-winner waypoint locations and perform LIO-based optimization of the trajectory
9:       Update learning parameters:  $\gamma = (1 - \eta)\gamma, i = i + 1$ 
10:    end for
11:  end for
12: end while
13: return final trajectory  $\Gamma$ 

```

Table 2. Average execution times (seconds) for the different algorithms.

Algorithm	Single-Area Single UAV	Single-Area Multiple UAVs (Two UAVs, Two Paths)
TOS	3.48	4.53
3DAA	4.74	5.12
EA	5.30	5.91
Proposed	5.14	6.35

Table 3. Average velocity (meter/second) and coverage time (second) for the different algorithms.

Algorithm	Scenario	Average Velocity (m/s)	Minimum Coverage Time (s)
TOS	Single-Area Single UAV	6.5	22.2
	Single-Area Multiple UAVs (Three UAVs, three paths)	5.1	7.4
3DAA	Single-Area Single UAV	6.5	13.6
	Single-Area Multiple UAVs (Two UAVs, two paths)	6.5	8.7
EA	Single-Area Single UAV	6.6	12.5
	Single-Area Multiple UAVs (Two UAVs, two paths)	6.8	6.5
Proposed	Single-Area Single UAV	6.8	7.9
	Single-Area Multiple UAVs (Two UAVs, two paths)	6.8	4.2

Table 4. Simulation Parameters.

Camera						
θ	H_{min}	c_1	c_2	c		
$\frac{\pi}{2}$	4 m	1 m	0.5 m	0.2 m		
Mobility of UAV for proposed method				Mobility of UAV for time-optimal method		
a_{ver}	a_{hor}	v_{ver}	v_{hor}	a_z	v_z	v_f
0.5 m/s ²	1.2 m/s ²	0.5 m/s	1.2 m/s	0.5 m/s ²	0.5 m/s	1.2 m/s

3.3.1. Single-Area Single UAV

To confirm the performance of our monitoring strategy in a complex environment, we performed validation in the following scenario. First, we apply the proposed Vantage Waypoint Set Generation in step one to obtain the waypoint set that can fully cover the target area. Figure 8b shows the δ -neighborhood of all 10 waypoint locations at different heights from 6.4 m to 23.3 m. Secondly, we compare the proposed UAV path with the three mentioned methods' paths in terms of operation time and surveillance quality. In accordance with surveillance quality, we take the smoothness of the path, path length, and whether the uncovered area exits into consideration. Under the same velocity profiles as our proposed method, the time-optimal strategy (TOS) [43], evolutionary algorithm (EA) [44], and 3D Alternating algorithm (3DAA) then run under the same environment. The proposed algorithm (see Figure 8c) and evolutionary algorithm (see Figure 8e) are superior to the time-optimal surveillance path-planning algorithm (see Figure 8d) and 3D alternating algorithm (see Figure 8f) in terms of operation time. The time-optimal surveillance path-planning algorithm is the most straightforward method and smoothest path among them. Nonetheless, it inevitably neglects the problems caused by occlusion, so we may need to apply the geometric calculation to calculate the uncovered part and deploy other unmanned aerial vehicles to cover it completely. On the other hand, the evolutionary algorithm has a longer path and more sharp turns.

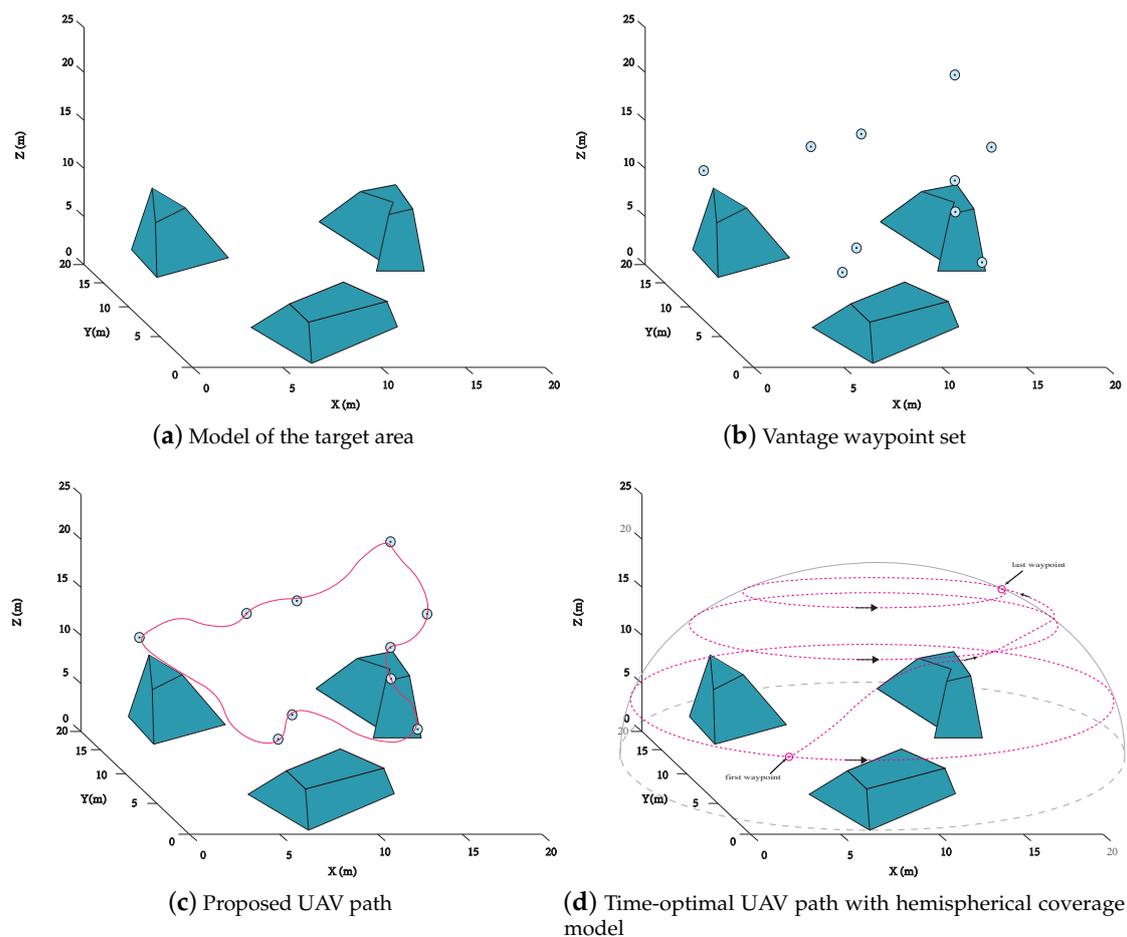


Figure 8. Cont.

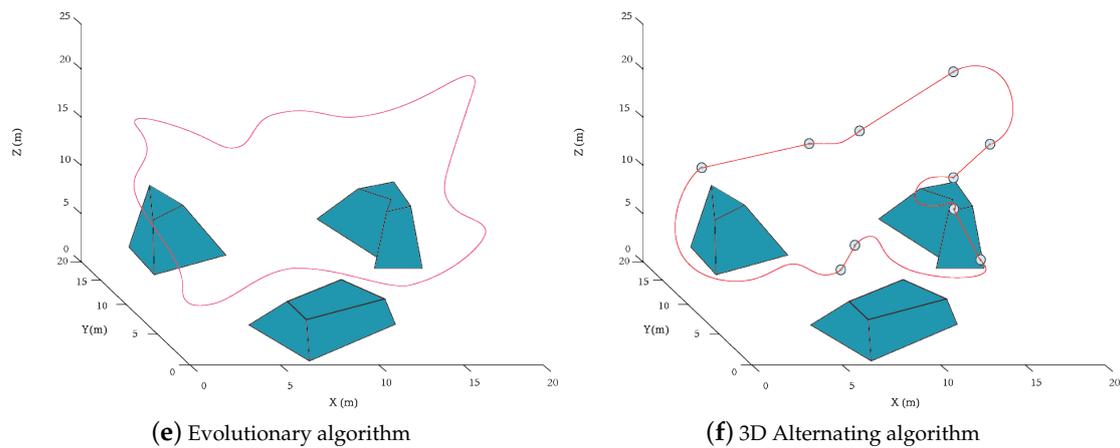


Figure 8. (a) the given environment, (b) the vantage waypoint set by the proposed method, UAV surveillance trajectory using (c) proposed strategy, (d) time-optimal strategy in [43], (e) evolutionary algorithm [44] and (f) 3D Alternating algorithm—Single UAV.

3.3.2. Single-Area Multiple UAVs

In this scenario, multiple UAVs are being used for full coverage reconnaissance and surveillance missions. The same waypoint positions are shown in Figure 8b. Figure 9a shows two drones in different initial positions share the same route. Figure 9b shows the situation where each UAV has a separate track to cover part of the terrain. Moreover, when both complete their monitoring circle, the coverage of the region of interest is realized. In both scenarios, the duration of covering the region of interest is obviously increased, so the average time of covering any point between two continuous times is reduced by about 47%. In other words, the points of interest are monitored more often. We also deploy three UAVs to cover each individual obstacle as a comparison. However, the uncovered part due to the overlapping of the flight surfaces is inevitable. Another two methods are applied for two UAVs to conduct the surveillance duty in Figure 9d,e. In this case, the evolutionary algorithm generates a smoother path than the proposed one. However, the proposed algorithm still outperforms the 3D alternating algorithm with a shorter length and fewer sharp turns. Table 3 shows the average velocity and minimum coverage time in both scenarios. The proposed approach shown in Figure 8c has the minimum coverage time in the single UAV scenario, and Figure 9a has the minimum coverage time in the multiple UAV scenario. Table 2 shows the computational time of the simulation above. The time-optimal strategy and 3D alternating algorithm take less time to cover the target area than the other two methods because of the lower computation load. Moreover, the evolutionary algorithm compromises the surveillance quality with computation time.

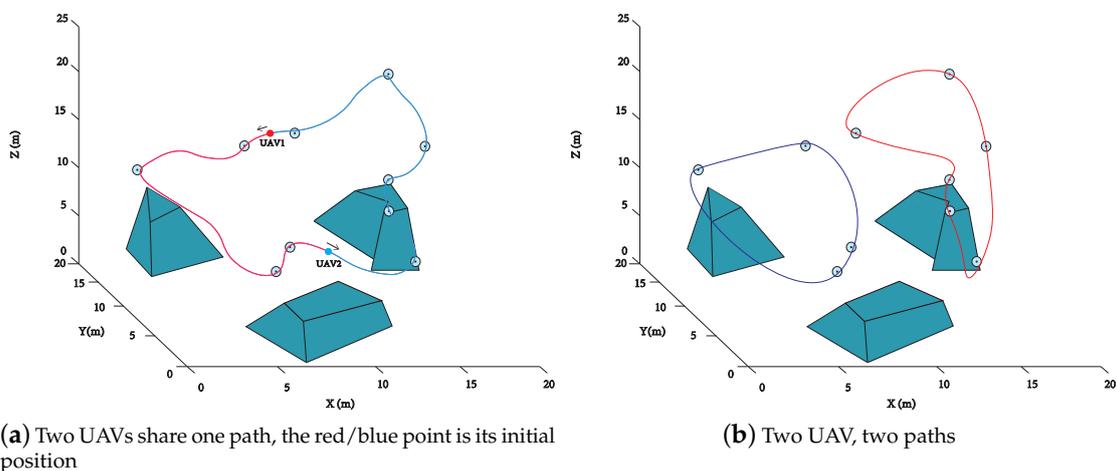


Figure 9. Cont.

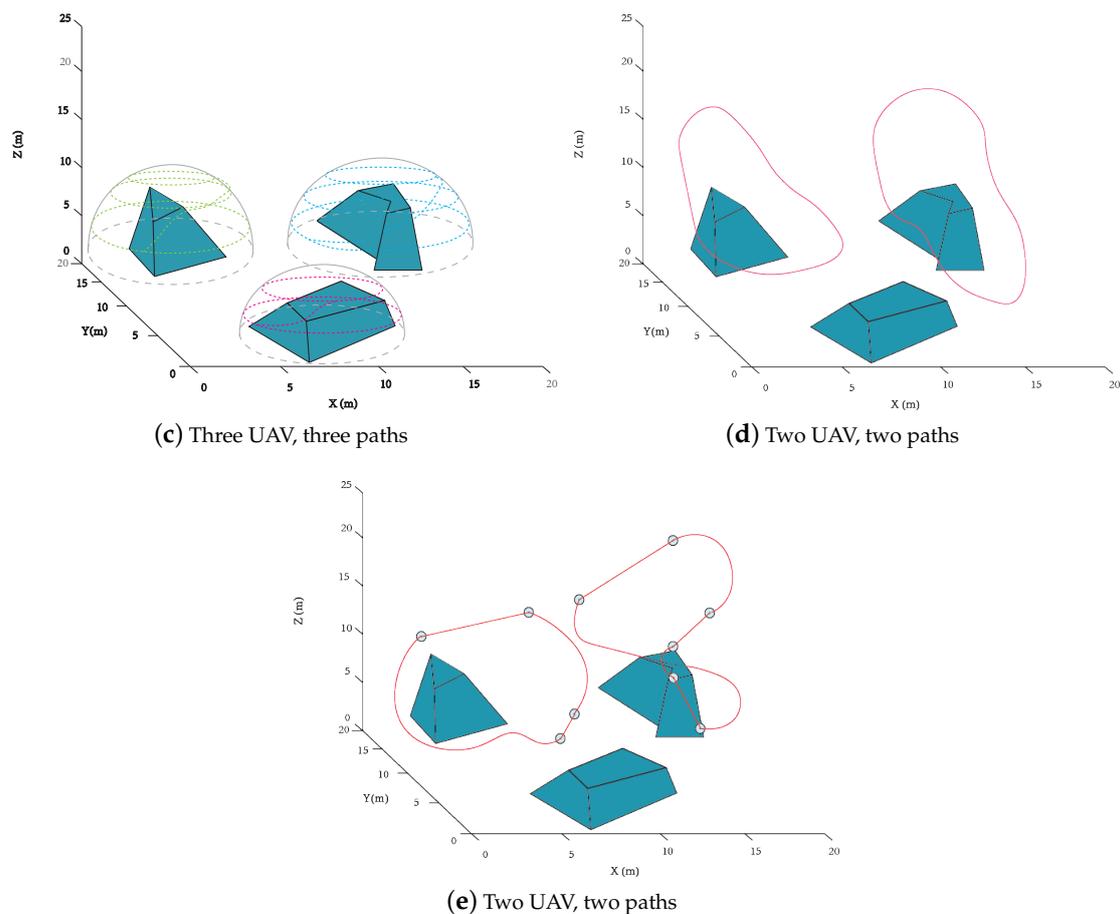


Figure 9. UAV surveillance trajectories using (a,b) proposed strategy and (c) time-optimal strategy in [43], (d) evolutionary algorithm [44], and (e) 3D alternating algorithm—Multiple UAVs

4. Conclusions and Future Work

We consider the dynamic coverage problem over the target area, where UAVs keep flying through each pre-defined waypoint during surveillance and taking video. In the first part, we propose a two-stage strategy to cover the target area completely at a given height. We tackle this thorny problem by decomposing it into a variant of the waypoint-determination problem and an instance of the Dubins Traveling Salesman Problem. An asymptotically optimal waypoint-generation algorithm is proposed, in the sense that the number of waypoints is closing to the minimum number of the waypoints as the target area tends to infinity. In addition, the proposed method considers solving the Dubins Vehicle Traveling-Salesman Problem (DTSP) by introducing the clustering spiral alternation algorithm, thus its performance is guaranteed to be better or the same as the benchmark methods. Furthermore, the trajectory takes into account the kinematics constraints of the UAV.

In the second part, we consider the reconnaissance and surveillance of UAV flying in complex geometric environments, such as mountainous and urban areas. We propose a two-stage strategy to completely cover the target area with different heights. The main contribution of this paper includes providing a method to determine the minimum number of waypoints that need to be visited and drawing a track at a given height to monitor the area, as well as an occlusion-aware UAV monitoring strategy based on UAV kinematic constraints and obstacle occlusion. The operation time and surveillance quality including smoothness, path length, and whether the uncovered area exists are considered to compare the performance of the described approach with the state of the art. The unsupervised learning method Self-Organizing Map (SOM) and Bézier curves are used to generate a

smooth and fast trajectory for the UAV. The proposed method achieves closer performance to the best performance among benchmarks. The performance and effectiveness of the two proposed algorithms have been confirmed by extensive computer simulations in different scenarios.

In the future research of occlusion-aware UAV surveillance, the change of the environment during the mission is an inevitable technical challenge to build an effective and efficient UAV surveillance system. For instance, drones should be able to change, adapt, modify and optimize paths to handle the occlusions by moving obstacles. It would be very interesting to modify the trajectories of the UAVs during the mission execution to avoid the occlusion effects by large moving objects.

Author Contributions: Conceptualization, H.H.; methodology, H.H., and J.Z.; software, J.Z.; validation, H.H. and J.Z.; formal analysis, J.Z.; investigation, H.H. and J.Z.; resources, J.Z.; data curation, J.Z.; writing—original draft preparation, J.Z.; writing—review and editing, J.Z. and H.H.; visualization, J.Z.; supervision, H.H.; project administration, J.Z.; funding acquisition, H.H. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Australian Research Council. Also, this work received funding from the Australian Government, via grant AUSMURIB000001 associated with ONR MURI grant N00014-19-1-2571.

Conflicts of Interest: The authors declare no conflict of interest.

References

- Castelli, T.; Sharghi, A.; Harper, D.; Treméau, A.; Shah, M. Autonomous navigation for low-altitude UAVs in urban areas. *arXiv* **2016**, arXiv:1602.08141.
- Girard, A.R.; Howell, A.S.; Hedrick, J.K. Border patrol and surveillance missions using multiple unmanned air vehicles. In Proceedings of the IEEE Conference on Decision and Control, Nassau, Bahamas, 14–17 December 2004; Volume 1, pp. 620–625. [\[CrossRef\]](#)
- Ozkan, O.; Kaya, M. UAV routing with genetic algorithm based matheuristic for border security missions. *Int. J. Optim. Control Theor. Appl.* **2021**, *11*, 128–138. [\[CrossRef\]](#)
- Tomic, T.; Schmid, K.; Lutz, P.; Domel, A.; Kassecker, M.; Mair, E.; Grixia, I.; Ruess, F.; Suppa, M.; Burschka, D. Toward a fully autonomous UAV: Research platform for indoor and outdoor urban search and rescue. *IEEE Robot. Autom. Mag.* **2012**, *19*, 46–56. [\[CrossRef\]](#)
- Silvagni, M.; Tonoli, A.; Zenerino, E.; Chiaberge, M. Multipurpose UAV for search and rescue operations in mountain avalanche events. *Geomat. Nat. Hazards Risk* **2017**, *8*, 18–33. [\[CrossRef\]](#)
- Calì, M.; Ambu, R. Advanced 3D photogrammetric surface reconstruction of extensive objects by UAV camera image acquisition. *Sensors* **2018**, *18*, 2815. [\[CrossRef\]](#)
- Mittal, M.; Mohan, R.; Burgard, W.; Valada, A. Vision-Based Autonomous UAV Navigation and Landing for Urban Search and Rescue. *arXiv* **2019**, arXiv:1906.01304.
- Metni, N.; Hamel, T. A UAV for bridge inspection: Visual servoing control law with orientation limits. *Autom. Constr.* **2007**, *17*, 3–10. [\[CrossRef\]](#)
- Savkin, A.V.; Huang, H. Asymptotically Optimal Deployment of Drones for Surveillance and Monitoring. *Sensors* **2019**, *19*, 2068. [\[CrossRef\]](#) [\[PubMed\]](#)
- Li, K.; Voicu, R.C.; Kanhere, S.S.; Ni, W.; Tovar, E. Energy efficient legitimate wireless surveillance of UAV communications. *IEEE Trans. Veh. Technol.* **2019**, *68*, 2283–2293. [\[CrossRef\]](#)
- Zhang, J. Occlusion-aware UAV Path Planning for Reconnaissance and Surveillance in Complex Environments. In Proceedings of the 2019 IEEE International Conference on Robotics and Biomimetics (ROBIO), Dali, China, 6–8 December 2019; pp. 1435–1440. [\[CrossRef\]](#)
- Huang, H.; Savkin, A.V. An algorithm of reactive collision free 3-d deployment of networked unmanned aerial vehicles for surveillance and monitoring. *IEEE Trans. Ind. Inform.* **2020**, *16*, 132–140. [\[CrossRef\]](#)
- Zelinsky, A.; Jarvis, R.; Byrne, J.; Yuta, S. Planning paths of complete coverage of an unstructured environment by a mobile robot. *Proc. Int. Conf. Adv. Robot.* **1993**, *13*, 533–538. [\[CrossRef\]](#)
- Koyuncu, E.; Shabanighazikelayeh, M.; Seferoglu, H. Deployment and trajectory optimization of UAVs: A quantization theory approach. *IEEE Trans. Wirel. Commun.* **2018**, *17*, 8531–8546. [\[CrossRef\]](#)
- Nemer, I.A.; Sheltami, T.R.; Mahmoud, A.S. A game theoretic approach of deployment a multiple UAVs for optimal coverage. *Transp. Res. Part A Policy Pract.* **2020**, *140*, 215–230. [\[CrossRef\]](#)
- Stodola, P.; Drozd, J.; Mazal, J.; Hodický, J.; Procházka, D. Cooperative Unmanned Aerial System Reconnaissance in a Complex Urban Environment and Uneven Terrain. *Sensors* **2019**, *19*, 3754. [\[CrossRef\]](#) [\[PubMed\]](#)

17. Roberge, V.; Tarbouchi, M.; Labonté, G. Fast genetic algorithm path planner for fixed-wing military UAV using GPU. *IEEE Trans. Aerosp. Electron. Syst.* **2018**, *54*, 2105–2117. [[CrossRef](#)]
18. Zhao, Y.; Zheng, Z.; Liu, Y. Survey on computational-intelligence-based UAV path planning. *Knowl.-Based Syst.* **2018**, *158*, 54–64. [[CrossRef](#)]
19. Li, X.; Savkin, A.V. Networked Unmanned Aerial Vehicles for Surveillance and Monitoring: A Survey. *Future Internet* **2021**, *13*, 174. [[CrossRef](#)]
20. Kellenberger, B.; Marcos, D.; Lobry, S.; Tuia, D. Half a percent of labels is enough: Efficient animal detection in UAV imagery using deep CNNs and active learning. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 9524–9533. [[CrossRef](#)]
21. Liu, Y.; Wang, Q.; Hu, H.; He, Y. A novel real-time moving target tracking and path planning system for a quadrotor UAV in unknown unstructured outdoor scenes. *IEEE Trans. Syst. Man Cybern. Syst.* **2018**, *49*, 2362–2372. [[CrossRef](#)]
22. Geng, L.; Zhang, Y.F.; Wang, J.J.; Fuh, J.Y.; Teo, S.H. Mission planning of autonomous UAVs for urban surveillance with evolutionary algorithms. In Proceedings of the 2013 10th IEEE International Conference on Control and Automation (ICCA), Hangzhou, China, 12–14 June 2013; pp. 828–833. [[CrossRef](#)]
23. Zhang, J.; Zhang, Y. A Method for UAV Reconnaissance and Surveillance in Complex Environments. In Proceedings of the 2020 6th International Conference on Control, Automation and Robotics (ICCAR), Singapore, 20–23 April 2020; pp. 482–485. [[CrossRef](#)]
24. Hoy, M.; Matveev, A.S.; Savkin, A.V. Algorithms for collision-free navigation of mobile robots in complex cluttered environments: A survey. *Robotica* **2015**, *33*, 463–497. [[CrossRef](#)]
25. Semsch, E.; Jakob, M.; Pavlíček, D.; Pěchouček, M. Autonomous UAV surveillance in complex urban environments. In Proceedings of the 2009 IEEE/WIC/ACM International Conference on Intelligent Agent Technology, IAT 2009, Milan, Italy, 15–18 September 2009; Volume 2, pp. 82–85. [[CrossRef](#)]
26. Dubins, L.E. On Curves of Minimal Length with a Constraint on Average Curvature, and with Prescribed Initial and Terminal Positions and Tangents. *Am. J. Math.* **1957**, *79*, 497. [[CrossRef](#)]
27. Ismail, A.; Tuyishimire, E.; Bagula, A. Generating dubins path for fixed wing uavs in search missions. In *International Symposium on Ubiquitous Networking*; Springer: Cham, Switzerland, 2018; pp. 347–358.
28. Kučerová, K.; Váň, P.; Faigl, J. On finding time-efficient trajectories for fixed-wing aircraft using dubins paths with multiple radii. In Proceedings of the 35th Annual ACM Symposium on Applied Computing, Brno, Czech Republic, 30 March–3 April 2020; pp. 829–831.
29. Manchester, I.R.; Savkin, A.V. Circular navigation missile guidance with incomplete information and uncertain autopilot model. *J. Guid. Control Dyn.* **2004**, *27*, 1078–1083. [[CrossRef](#)]
30. Fathi, Z.; Bidabad, B.; Najafpour, M. An exact penalty function method for optimal control of a dubins airplane in the presence of moving obstacles. *Optim. Lett.* **2021**, doi:10.1007/s11590-021-01773-6. [[CrossRef](#)]
31. Carothers, N.L. *Real Analysis*; Cambridge University Press: Cambridge, UK, 2000. [[CrossRef](#)]
32. Forsyth, D.; Ponce, J. *Computer Vision: A Modern Approach*; Prentice Hall: Englewood Cliffs, NJ, USA, 2011.
33. Savkin, A.V.; Huang, H. Deployment of unmanned aerial vehicle base stations for optimal quality of coverage. *IEEE Wirel. Commun. Lett.* **2019**, *8*, 321–324. [[CrossRef](#)]
34. Savla, K. Multi UAV Systems with Motion and Communication Constraints. Ph.D. Thesis, University of California, Santa Barbara, CA, USA, 2007.
35. Govindaraju, V.; Leng, G.; Qian, Z. Multi-UAV surveillance over forested regions. *Photogramm. Eng. Remote Sens.* **2014**, *80*, 1129–1137. [[CrossRef](#)]
36. Mirjalili, S.; Mirjalili, S.M.; Lewis, A. Grey wolf optimizer. *Adv. Eng. Softw.* **2014**, *69*, 46–61. [[CrossRef](#)]
37. Marengoni, M.; Draper, B.A.; Hanson, A.; Sitaraman, R. A System to place observers on a polyhedral terrain in polynomial time. *Image Vis. Comput.* **2000**, *18*, 773–780. [[CrossRef](#)]
38. Savkin, A.; Huang, H. Proactive Deployment of Aerial Drones for Coverage over Very Uneven Terrains: A Version of the 3D Art Gallery Problem. *Sensors* **2019**, *19*, 1438. [[CrossRef](#)]
39. Fisk, S. A short proof of Chvátal’s Watchman Theorem. *J. Comb. Theory Ser. B* **1978**, *24*, 374. [[CrossRef](#)]
40. Faigl, J.; Vana, P. Surveillance Planning with Bézier Curves. *IEEE Robot. Autom. Lett.* **2018**, *3*, 750–757. [[CrossRef](#)]
41. Kohonen, T.; Schroeder, M.R.; Huang, T.S. (Eds.) *Self-Organizing Maps*, 3rd ed.; Springer: Berlin/Heidelberg, Germany, 2001.
42. Vana, P.; Faigl, J. On the Dubins Traveling Salesman Problem with Neighborhoods. *IEEE Int. Conf. Intell. Robot. Syst.* **2015**, *2015*, 4029–4034. [[CrossRef](#)]
43. Cheng, P.; Keller, J.; Kumar, V. Time-optimal UAV trajectory planning for 3D urban structure coverage. In Proceedings of the 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, Nice, France, 22–26 September 2008; pp. 2750–2757. [[CrossRef](#)]
44. Geng, L.; Zhang, Y.F.; Wang, P.F.; Wang, J.J.; Fuh, J.Y.; Teo, S.H. UAV surveillance mission planning with gimbaled sensors. In Proceedings of the IEEE International Conference on Control and Automation (ICCA), Taichung, Taiwan, 18–20 June 2014; pp. 320–325. [[CrossRef](#)]