energies

MDPI

Article

Energy-Efficient Autonomous Navigation of Solar-Powered UAVs for Surveillance of Mobile Ground Targets in Urban Environments

Hailong Huang and **Andrey V. Savkin** *

School of Electrical Engineering and Telecommunications, University of New South Wales,
Sydney 2052, NSW, Australia; hailong.huang@unsw.edu.au
* Correspondence: a.savkin@unsw.edu.au

Received: 8 September 2020; Accepted: 21 October 2020; Published: 23 October 2020

check for updates

Abstract: In this paper, we consider the navigation of a group of solar-powered unmanned aerial vehicles (UAVs) for periodical monitoring of a set of mobile ground targets in urban environments. We consider the scenario where the number of targets is larger than that of the UAVs, and the targets spread in the environment, so that the UAVs need to carry out a periodical surveillance. The existence of tall buildings in urban environments brings new challenges to the periodical surveillance mission. They may not only block the Line-of-Sight (LoS) between a UAV and a target, but also create some shadow region, so that the surveillance may become invalid, and the UAV may not be able to harvest energy from the sun. The periodical surveillance problem is formulated as an optimization problem to minimize the target revisit time while accounting for the impact of the urban environment. A nearest neighbour based navigation method is proposed to guide the movements of the UAVs. Moreover, we adopt a partitioning scheme to group targets for the purpose of narrowing UAVs' moving space, which further reduces the target revisit time. The effectiveness of the proposed method is verified via computer simulations.

Keywords: unmanned aerial vehicles (UAVs); autonomous systems; solar-powered UAVs; control of electric vehicles; applications of solar powered vehicles; energy-efficient UAV navigation; sensing coverage; dynamic coverage; surveillance and monitoring; rapidly-exploring random tree (RRT)

1. Introduction

Unmanned aerial vehicles (UAVs) have found numerous applications in both military and civilian domains. They are excellent tools for target surveillance and monitoring [1–3], thanks to their flexibility. Because using a single UAV is often inefficient to conduct a complex mission, employing a UAV team is the trend in order to complete missions quickly [4]. When multiple UAVs conduct some missions, they are often regarded as a multiagent system. In the past few decades, the coordination issue of multiagent systems has attracted great attention from different research communities [5–7].

This paper pays attention to the moving target surveillance by a group of UAVs. A practical application of the considered scenario is that, in wireless sensor networks, the sensor nodes collect data from the environment. UAVs function as data sinks to collect the sensory data from sensor nodes [8]. In general, the number of available UAVs is smaller than that of the sensor nodes. Thus, the UAVs carry out a periodical surveillance of the sensor nodes.

Because UAVs often have limited onboard battery capacity, their operation duration is constrained. Installing solar-panels enables the UAVs to harvest energy from the sun, which is promising for prolonging the lifetime of the UAVs in the sunny daytime [9]. We consider the surveillance of mobile targets by the solar-powered UAVs in urban environments. The tall buildings have some negative impact on the mission. Firstly, they may create some shadow region, where the Line-of-Sight (LoS) between the UAVs and the sun is blocked, so that the UAVs cannot harvest energy if they fly in the shadow region. Besides, the buildings may also block the LoS between the UAVs and considered targets. This may prevent a UAV from successfully surveying a target.

The problem of interest is formulated as an optimization problem to minimize the target revisit time by planning the UAVs' paths. To address the problem, we present a path planning method that is based on the Rapidly-exploring Random Tree (RRT). This method can quickly find a feasible path for the UAV to intercept the target in the scenario where the target moves along a known trajectory. We then consider the case with one UAV and multiple targets. We present a nearest neighbour (NN) based navigation method. The so-called NN involves both the UAV-target distance as well as the uncertainty level of a target. Finally, we consider the multi-UAV and multi-target case. We partition the targets into groups according to the distance information between the targets and the UAVs. Subsequently, each UAV takes care of the targets in its own partition.

The proposed autonomous navigation algorithm that navigates a UAV team in order to periodically survey a group of mobile ground targets is the main contribution of this paper. It is computationally efficient and easily implementable online, since it is a randomized RRT-based approach. Extensive simulation results are reported in order to confirm the effectiveness of the developed method.

The reminder of the paper is organized, as follows. Section 2 briefly reviews the relevant work. Section 3 presents the system models and formulates the problem. Section 4 presents the proposed UAV navigation approaches. Section 5 reports the computer simulation results, and Section 6 gives the concluding remarks.

2. Related Work

The target surveillance problem that is considered in this paper has not been considered in any existing publications. In this section, we present some closely relevant publications, so that we can distinguish the contributions of the paper with others.

The target surveillance problem has been investigated from different levels in the literature. In terms of sensing, a large number of image/video processing strategies have been developed in order to estimate states of the targets from the measured images/videos [10–14]. In these publications, attention has been paid to the quality of detection for a single target.

In the scenario with multiple targets, how to allocate the UAV resource becomes necessary to achieve a good quality of surveillance. Many operational research results, such as the conventional travelling salesman problem (TSP) [15] and the vehicle routing problem (VRP) [16], are the common tools for planning the UAVs' paths. When there are enough UAVs, the coverage control has been investigated to achieve the optimal sensing coverage of the targets [17,18]. In cases where moving targets are to be monitored, and to maintain the quality of sensing, the reactive algorithms have been proposed [2,19].

This paper focuses on the scenario where the number of UAVs is not enough to persistently monitor the targets, so a periodical surveillance is conducted by the UAVs. As the targets are moving in the considered region, the problem is more relevant to the time-dependent TSP [20] and the moving-target TSP [21,22]. In the time-dependent TSP [20], the common setting is to find the shortest tour for the salesman in a graph with time-dependent edges. In the moving-target TSP [21,22], the targets are assumed to move with a constant speed along a fixed direction. The problem considered here is different from them. Firstly, the targets move along some trajectories, so that their speeds and moving directions may change with time. Secondly, the existence of buildings in the urban environment requires the UAVs to avoid collision with the buildings. Thirdly, the UAVs need to harvest energy from the sun to enable the UAVs to operate in the given time period. However, each path depends on the UAV's initial position, the buildings' positions and the target's trajectory, which is challenging to be known in advance. Thus, both the moving-target TSP and the time-dependent TSP cannot be applied to address the problem that is considered in the paper.

Path planning plays an important role in this work. Among various path planning algorithms, RRT is a sampling-based approach. It generates a feasible (but may not be optimal) path quickly, even if the environment is complicated. Many publications have reported that this method can be easily applied in real-time applications, such as mobile ground robots [23] and autonomous driving [24]. To improve the solution quality and computing speed, many RRT variants have been developed. Specific attention has been paid to the generation of samples and the control of the searching step length. A lower bound tree-RRT is designed to find out the near optimal path [25]. Besides, a node control strategy is proposed in order to restrict the expansion of the random tree [26]. Because of the computational efficiency, RRT-based approaches are generally suitable to run in real-time, and it also has potential to be implemented in a decentralized manner [27]. We adopt the RRT approach in this paper. However, as will be shown in the following sections, we do not have a fixed destination for a UAV. Instead, the destination of a UAV moves. Our objective is generate a feasible path in real-time, so that the UAV can intercept the target as soon as possible.

3. System Model and Problem Statement

Suppose that we have a team of solar-powered UAVs labelled $1, 2, \dots, I$. We consider that these solar-powered UAVs fly at a fixed altitude z in an urban area to conduct some missions. For UAV i, let $p_i(t) = [x_i(t), y_i(t), z]$ be its position in the ground frame at t, $\theta_i(t)$ be the horizontal heading angle with respect to the x-axis; and, $v_i(t)$ and $\omega_i(t)$ be its linear and angular speeds, respectively. The motion of UAV i can be described by the following equations [28,29]:

$$\begin{cases} \dot{x}_i(t) = v_i(t)\cos(\theta_i(t)), \\ \dot{y}_i(t) = v_i(t)\sin(\theta_i(t)), \\ \dot{\theta}_i(t) = w_i(t), \end{cases} \tag{1}$$

In this paper, the effect of wind has not been considered. The following constraints hold for any UAV at any time:

$$\begin{cases} -V_i^{\max} \le v_i(t) \le V_i^{\max}, \\ -\Omega_i^{\max} \le w_i(t) \le \Omega_i^{\max}, \\ (x_i(t), y_i(t)) \in \mathcal{D}, \end{cases} \tag{2}$$

Here, Ω_i^{max} and V_i^{max} are the given constants, and $\mathcal{D} \subset \mathbf{R}^2$ is the considered area. The movement of many UAVs can be described by (1) and (2); see [30–33]. It is worth pointing out that, in (2), the linear speed $v_i(t)$ can take a negative value. This allows for a UAV to move backward when necessary. In Section 5, we will see some UAV trajectories with sharp turns, and the reason is that a negative linear speed is applied. This avoids making a large turn by moving along a circle. Table 1 summarizes the frequently used symbols in the paper, together with their meanings.

Let $P_i^{sun}(t)$ be the harvesting power of the solar energy. It can be computed as follows [34]

$$P_i^{sun}(t) = \eta A_i \cos\phi(t), \tag{3}$$

where η represents the solar cell efficiency, A_i represents the area of the solar cells, and ϕ gives the incidence angle. The incidence angle ϕ is further dependent on the azimuth angle α_z and the elevation angle α_e of the sun, and in the day time, both α_z and α_e vary with time.

Table 1. Symbols and meanings.

Symbol	Meaning
$p_i(t)$	UAV i's position at time t
$\theta_i(t)$	UAV i's heading angle
$v_i(t)$	UAV i's linear speed
$\omega_i(t)$	UAV i's angular speed
α	Observation angle
R	Radius of the vision cone
$P_i(t)$	UAV i's consuming power
$P_i^{sun}(t)$	UAV i's harvesting power
$Q_i(t)$	UAV i's residual energy
$q_i(t)$	Target j's position
τ_j	Target j's revisit time
$d_{ij}(t)$	Horizontal distance between UAV i and target j

The UAVs consume energy when they are flying. For the energy consuming power, we follow the model that was used in [35]:

$$P_i(t) = \lambda_0 \left(1 + \frac{3v_i^2}{U_{tip}^2}\right) + \frac{1}{2}\rho\kappa s S v_i^3(t) + \lambda_1 \left(\sqrt{1 + \frac{v_i^4(t)}{4\mu^4}} - \frac{v_i^2(t)}{2\mu^2}\right)^{\frac{1}{2}}, \tag{4}$$

where λ_0, λ_1 and μ are the blade profile power, the induced power and the mean rotor induced velocity in hovering, respectively; U_{tip} represents the tip speed of the rotor blade; κ is the fuselage drag ratio; s represents the rotor solidity; ρ is the air density; and, S_i rotor disc area.

Let $Q_i(t)$ denote the residual energy of the battery of UAV i. We have

$$\dot{Q}_i(t) = P_i^{sun}(t) - P_i(t). \tag{5}$$

Moreover, Q_i^{max} represents the upper bound of $Q_i(t)$.

Each UAV carries a ground-facing camera, and the camera's observation angle is denoted by $\alpha \in (0, \pi)$; see Figure 1. If a target locates in a disc centred at $p_i(t)$ of the radius

$$R = z \tan\left(\frac{\alpha}{2}\right), \tag{6}$$

and has LoS with the UAV, it can be observed by the UAV. We assume that a gimble is available on the UAV, so that, no matter how the UAV moves, the camera always faces the ground.

Now, we model the buildings. In this paper, each building is modelled as the smallest prism to enclose this building. Each prism has two parallel and congruent bases and a number of flat sides that are perpendicular to the xy-plane; see Figure 2. Each prism is characterized: Ψ, ψ and h. Ψ is a κ-by-2 matrix and ψ is a κ-by-one vector. They determine the shape and size of the base. h is a scalar indicating the height of the prism. For a point (x, y, z), if it is inside a prism, (7) holds:

$$\begin{cases} \Psi \begin{bmatrix} x \\ y \end{bmatrix} \leq \psi, \\ 0 \leq z \leq h. \end{cases} \tag{7}$$

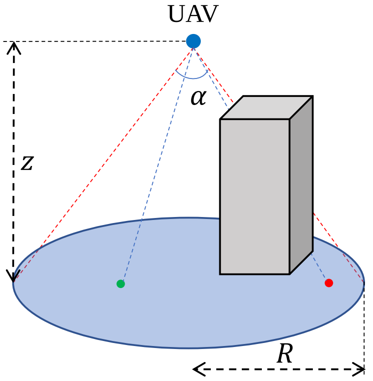
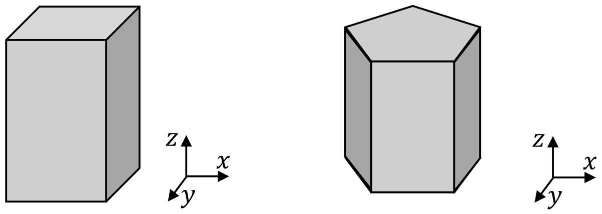


Figure 1. The observation of the ground-facing camera. The target at the green point can be viewed by the unmanned aerial vehicle (UAV) while that at the red point cannot due to the blockage of the building.

Given the environment information, Ψ, ψ and h are known for each building. Subsequently, we can have a subset of space, denoted by $\mathcal{X}_{building}$, which corresponds to these buildings. At any time, the UAVs must not be inside $\mathcal{X}_{building}$. Clearly, avoiding $\mathcal{X}_{building}$ is similar to the collision avoidance with steady obstacles [36,37].



(**a**) Four sides (**b**) Five sides

Figure 2. Prisms.

Having the model of buildings is not sufficient to characterize the observation region of a UAV. We also need a method to determine whether a position in the air and a position on the ground have LoS. For this purpose, we consider the straight line segment between two points p and q, which is described as

$$\begin{cases} x = x_q + \beta_x \tau, \\ y = y_q + \beta_y \tau, \\ z = z_q + \beta_z \tau, \\ \min\{x_p, x_q\} \leq x_q + \alpha\tau \leq \max\{x_p, x_q\}, \end{cases} \tag{8}$$

where $q = (x_q, y_q, z_q)$, $p = (x_p, y_p, z_p)$, $(\beta_x, \beta_y, \beta_z) = \frac{\vec{pq}}{\|\vec{pq}\|}$, and τ is a free variable.

Whether p and q have LoS can be tested by looking for the intersection points between the the line segment connecting p and q (8) and any prism (7). Because the environment information is known, whether p and q have LoS can be easily confirmed. We introduce a function $LOS(p, q, \mathcal{X}_{building})$:

$$LOS(p, q, \mathcal{X}_{building}) = \begin{cases} 1, & \text{if } p \text{ and } q \text{ have LoS}, \\ 0, & \text{otherwise}. \end{cases} \tag{9}$$

With this function, we can also test whether a UAV and the sun have LoS. To this end, the sun's location needs to be known. Let V, a unit vector, denote the sunlight direction. With V, we can imagine that the sun is at $q_{sun} = p - V\tau$, where the parameter τ takes a large value so that the sun is far from the point p. We need to place the sun at a relatively far position. The reason is that we use the line segment to verify whether two points have LoS. When the sun is placed closely to the point p, we may not obtain the correct verification.

Let $b_i(t)$ be a binary variable indicating whether UAV i has LoS with the sun. Subsequently, UAV i's residual energy can be computed by:

$$\dot{Q}_i(t) = P_i^{sun}(t)b_i(t) - P_i(t). \tag{10}$$

There are N ground mobile targets in the given urban environment to be periodically surveyed. These targets can be some sensor nodes to measure the environment information of interest. Instead of continuously transmitting the sensory data, they only upload their sensory data to a control unit via the UAVs in proximity. This setting can prolong the lifetime of the nodes when the sensory data are of large size, such as videos. We assume that the UAVs know the current positions of the targets, and this information can be provided by the targets, since the energy consumption of reporting the position information can be ignored compared to the large size of sensory data. We also assume that the targets' future positions are predictable. This assumption is reasonable, since, when the targets carry out some pre-defined missions, their trajectories can be known. Let $q_j(t) \in \mathbf{R}^3$ denote target j's location ($j = 1, \ldots, N$) at time t.

In this paper, we consider that $I < N$ and the targets spread in the considered environment. Subsequently, there may be some time in which a target is not under surveillance. From the common sense, the uncertainty level of a target relates to the time in which it is not under surveillance. Thus, a significant objective of the surveillance system is to maintain the uncertainty level of the targets as low as possible. This can be formulated as the minimization of the maximum target revisit time. Let τ_j denote the time gap between two consecutive visits of target j. Let $d_{ij}(t)$ denote the horizontal distance between target j and UAV i at time t.

Definition 1. *Target j is under surveillance of UAV i at time t, if $LOS(p_i(t), q_j(t), \mathcal{X}_{building}) = 1$ and $d_{ij}(t) \leq R$.*

Let $s_j(t)$ be a binary variable indicating if target j is under surveillance at time t. Subsequently, we have

$$s_j(t) = \begin{cases} 1, \text{ if } \exists i \text{ such that } LOS(p_i(t), q_j(t), \mathcal{X}_{building}) = 1 \text{ and } d_{ij}(t) \leq R, \\ 0, \text{ otherwise.} \end{cases} \tag{11}$$

Afterwards, we can use $s_j(t)$ to calculate τ_j. Specifically, we have

$$\tau_j = \max_{s_j(t_1)=1, s_j(t_2)=1, t_1<t_2, \forall t_1<t<t_2, s_j(t)=0} t_2 - t_1. \tag{12}$$

In other words, τ_j is the time instant gap between the two consecutive visits. Note that, if there is only one visit during the mission period $[0, T]$, i.e., at t_1, then, $\tau_j = T - t_1$. If there is not any visit during the mission period, then $\tau_j = T$.

The problem under investigation is to develop a navigation method for the UAVs modelled by (1) and (2) in order to minimize the maximum revisit time during the mission period $[0, T]$, i.e.,

$$\min \max_{j=1,\ldots,N} \tau_j. \tag{13}$$

subject to

$$Q_i(t) > 0, \ \forall i, \ \forall t \in [0, T]. \tag{14}$$

The problem under consideration is difficult to address optimally. Although we can have the trajectories of targets and predict their positions for the period of $[0, T]$, it is still hard to plan the trajectories of the UAVs in advance. The main reason lies in the complexity of the flying space in urban environments. In particular, due to the existence of buildings, the trajectory of UAV i (suppose that it is assigned to survey target j) depends on both the trajectory of target j and the position of UAV i at which it is assigned with this task. Furthermore, the UAV's position depends on its last task. The coupling of the high-level task assignment problem and the low-level trajectory planning problem makes it too complex to be addressed optimally. Even if an optimal solution can be obtained, it may also take so long time that it cannot be applied online.

4. Proposed Navigation Method

4.1. Lifetime of UAVs

Before presenting the navigation method, it is necessary to discuss the lifetime of the UAVs. As shown in (3), for a UAV, its energy harvesting power varies with time because of the movement of the sun. Subsequently, the maximum harvested energy amount by UAV i is given by $\int_0^T \eta A_i \cos\phi(t)dt$.

According to (4), the energy consuming power P_i increases with the speed v_i in the first and second terms, while decreases with v_i in the third term. Additionally, the third term weights more when v_i is relatively small, while the first and second terms weight more when v_i is relatively large. Thus, the energy consuming power P_i in (4) first decreases and then increases with v_i. Figure 3 shows an illustrative plot of the relationship between P_i and v_i. There exists an optimal linear speed, such that the energy consuming power is minimized. Let P_i^{opt} denote the minimum energy consuming power.

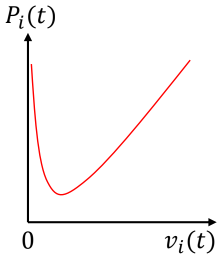


Figure 3. Illustration of the relationship between $P_i(t)$ and $v_i(t)$.

We assume that Q_i^{max} is the initial energy of UAV i. The necessary condition for the UAV to conduct the surveillance mission is as follows:

$$Q_i^{max} + \int_0^T \eta A_i \cos\phi(t)dt - TP_i^{opt} > 0. \tag{15}$$

Once (15) holds, the UAV can complete the surveillance mission with the optimal linear speed, although this does not guarantee the performance of surveillance. Moreover, a hidden assumption of (15) is that the UAV always has LoS with the sun in $[0, T]$. Otherwise, the harvested energy amount is smaller than $\int_0^T \eta A_i \cos\phi(t)dt$. However, if (15) does not hold, the UAV cannot conduct the mission. If the capacity Q_i^{max} takes some larger value, it is allowed to have some part of the UAV trajectory having no LoS with sun. The margin of capacity has the potential to reduce the revisit time. There are two feasible paths for the UAV to intercept the target, as shown in Figure 4. The red one has some part behind the building that prevents the UAV from harvesting energy, while the green path enables the UAV to harvest energy during the trip. The red one leads to a shorter time for the UAV to survey the target than the green. If there is no margin capacity, then the UAV has to follow the green path. Otherwise, it can follow the red one to reduce the revisit time.

Though the margin energy capacity brings benefit in terms of the reduction of the revisit time, it also creates challenges in the management of this amount of energy. Specifically, the UAV may need to survey several targets during the mission. Subsequently, it is difficult to decide how to allocate the margin capacity to the tasks. In the subsequent parts, we assume that there is no such margin capacity, and the UAVs fly at their corresponding optimal linear speeds. Accordingly, the UAVs always look for paths having LoS with sun. We leave the complex case with margin capacity for future study.

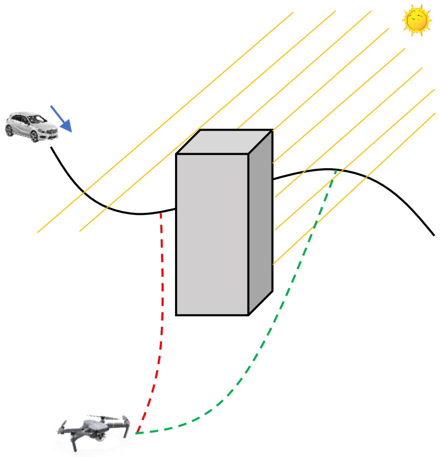


Figure 4. Illustration of the potential benefit offered by the margin energy capacity. The margin energy capacity allows the UAV to take a path with some part having no Line-of-Sight (LoS) with sun to reduce the revisit time.

4.2. One UAV and One Target

In the simplest case, we consider that one UAV is assigned to survey a target. Suppose that it is at t_0 the UAV starts to move to the target. We denote the initial positions of the UAV and the target as $p(t_0)$ and $q(t_0)$, respectively, and the initial heading angle of the UAV as $\theta(t_0)$. We use $\mathbf{x}(t) = [p(t), \theta(t)]$ to represent the state of the UAV. Suppose we know the trajectory of the target in the considered area $\mathcal{D}$, and we make a prediction for its future positions in the time interval $[t_0, t_0 + T_0]$. Formally, we know $q(t)$, where $t \in [t_0, t_0 + T_0]$. We select a suitable T_0, so that the UAV can intercept the target before $t_0 + T_0$.

For the purpose of implementing the method online, we adopt the computationally efficient RRT approach. The common setting of the RRT approach looks for a feasible path between a start position and a destination, and the obtained path can avoid obstacles. In our problem, the buildings that are taller than the UAV flying altitude are regarded obstacles. Our problem has some additional features compared to the common setting. Firstly, different from a stationary destination, the destination in our problem, i.e., the target, is moving. Secondly, the UAV should avoid shadow region, because it needs to always harvest energy during the trip.

The objective is to find a feasible UAV path, such that, at time instant $t \in [t_0, t_0 + T_0]$, the target locates inside the vision cone of the UAV and they have LoS. Starting from the current UAV position, i.e., $p(t_0)$, we generate a random tree. The termination condition of the tree generation process is that $d(t) \leq R$ and $LOS(p(t), q(t), \mathcal{X}_{building}) = 1$ at time t, where $d(t)$ is the horizontal UAV-target distance.

We present all of the procedures in Algorithm 1. Let $\mathcal{T}$ denote the random tree and δ be a sampling interval. $\mathcal{T}$ consists of a set of vertices. A vertex is annotated with control inputs, parent vertex and timestamp. Firstly, we initialize the tree $\mathcal{T}$ with $\mathbf{x}(t_0)$. Subsquently, we keep generating random samples in the space, find the nearest vertex from the tree to the sample, choose the suitable control inputs to generate a new vertex x_{new}. We further check the conditions that whether x_{new} belongs to $\mathcal{X}_{building}$ and whether it has LoS with the sun. When both are verified, we associate the parent vertex, the control inputs and the timestamp with this vertex, and add it to the random tree. Here, the timestamp is an integer indicating the number of steps from the initial vertex to this vertex. We stop growing the tree once there exist a timestamp k and a vertex $\mathbf{x}^*$ with the timestamp of k, such that $d(t_0 + k\delta) \leq R$ and the position of $\mathbf{x}^*$ has LoS with $q(t_0 + k\delta)$.

Algorithm 1: Rapidly-exploring random tree (RRT)-based path planning algorithm to intercept a target.

```
Input: $\mathbf{x}(t_0)$, $\mathcal{X}_{building}$, $q(t)$ where $t \in [t_0, t_0 + T_0]$
Output: $\mathcal{T}$
Initialize $\mathcal{T}$.
while There do not exist a timestamp $k$ and a vertex $\mathbf{x}^*$ with the timestamp of $k$, such that
  $d(t_0 + k\delta) \leq R$ and $\mathbf{x}^*$ has LoS with $q(t_0 + k\delta)$ do
    Randomly generate a node in the space.
    Select the closest vertex from $\mathcal{T}$ to the node.
    Choose the appropriate control inputs to generate $\mathrm{x}_{new}$, such that $\mathrm{x}_{new}$ cannot be closer to
      the node after $\delta$.
    if $x_{new} \notin \mathcal{X}_{building}$ and has LoS with the sun then
        Associate with $\mathrm{x}_{new}$ its parent, the applied control inputs and timestamp, and add $\mathrm{x}_{new}$
          to $\mathcal{T}$.
    end
end
```

When the tree growing process terminates, we obtain the final vertex on the UAV path, i.e., $\mathbf{x}^*$. We also know that it takes k steps for the UAV from its initial position to reach $\mathbf{x}^*$. By a standard backtracking algorithm, we can find the path from $\mathbf{x}^*$ back to the initial position. An example is shown in Figure 5, where t_0 is set as 0. In this example, we stop growing the random tree, since, at time 3δ, the condition $d(3\delta) \leq R$ holds (suppose $LOS(p(3\delta), q(3\delta), \mathcal{X}_{building}) = 1$).

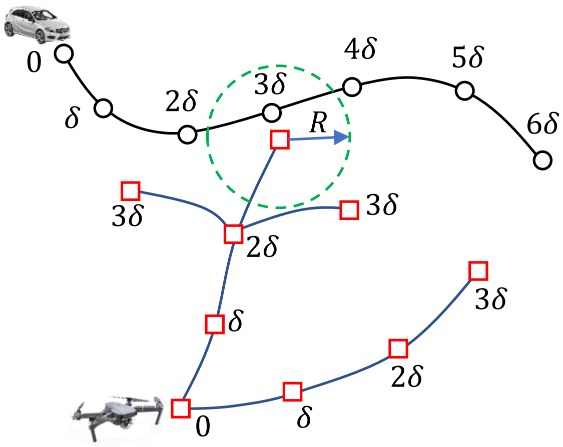


Figure 5. Illustration of the rapidly-exploring random tree (RRT)-based path planning.

4.3. One UAV and Multiple Mobile Targets

Now, we focus on the case, where one UAV periodically surveys multiple mobile targets.

The problem is a generalization of some variants of TSP. Different from the moving-target TSP [21,22], which assumes that the targets move at constant speeds in fixed directions, the targets in our problem may adjust their headings as well as speeds. Although we can make reasonably accurate predictions on the targets' movements, the time that is needed to intercept a target also depends on how the UAV moves. Thus, the time-dependent TSP [20], which assumes knowing the cost of the time-dependent arcs cannot be used directly to address our problem.

We present a nearest neighbour (NN) based navigation algorithm. Different from the common sense of NN, which uses distance information as a metric, we consider not only the distance, but also the uncertainty level of a target. A target's uncertainty level is modelled as a non-decreasing function of time since its last visit. A typical uncertainty function, denoted by $\gamma(t)$, is shown in Figure 6. Here, the target was last seen at instant t_0. For $t \in [t_0, t_0 + \sigma]$, $\gamma(t) = 0$. After $t_0 + \sigma$, $\gamma(t)$ increases with time. When $\sigma = 0$, $\gamma(t)$ is a monotonically increasing function of time.

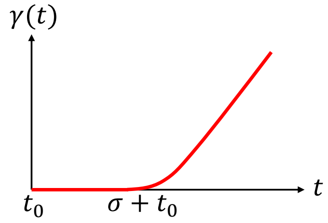


Figure 6. An illustrative example of the uncertainty level $\gamma(t)$.

We use the symbol $\lambda_j(t)$ to describe how close target j is to the UAV, and it is defined, as follows:

$$\lambda_j(t) := \frac{\gamma_j(t)}{L_j(t)}, \tag{16}$$

where $L_j(t)$ represents the length of the path for the UAV to intercept target j. The path can be found by the method that is discussed in Section 4.2. The metric λ couples the UAV-target distance and target's uncertainty level. The target with the maximum value of λ is the nearest neighbour (NN).

The navigation method is shown in Algorithm 2. It first selects the NN. Subsequently, the UAV follows the path to move towards the NN. The UAV will choose a new NN once the selected NN is surveyed. Note that since the targets are moving, it is possible that the UAV can have some other targets in view before the NN. In this case, the UAV will update the targets' status, but not change the selected NN. As a heuristic algorithm, it does not guarantee obtaining the optimal guidance. However, since it is a randomized method, it can complete the calculation quickly.

Algorithm 2: Navigating one UAV to survey multiple mobile targets.

```
while the time has not reached $T$ do
    Apply Algorithm 1 to find the paths to intercept the targets.
    Choose the NN.
    while the NN has not been surveyed do
        Keep moving on the planned path.
        Update the targets' status if they are surveyed during the movement.
    end
end
```

4.4. Multiple UAVs and Multiple Mobile Targets

Section 4.3 presents how to navigate one UAV to survey multiple mobile targets. This subsection extends the navigation algorithm to the scenario having multiple UAVs.

The target revisit time can be further reduced if multiple UAVs are available and their paths are well planned. Subsequently, an important question is how the UAVs coordinate. For coordinating, it requires some extra operations beyond intercepting targets. Firstly, the UAVs should exchange their positions and their selected NN. Exchanging positions can effectively avoid collision with other UAVs [37,38]. Exchanging the selected NN can avoid the case where two UAVs choose the same target as NN. Additionally, a UAV should share the status of a target once this target is surveyed. By doing this, all of the UAVs have a whole picture of the status of the targets, so that they can select their NNs more effectively. Without this sharing, a UAV may move towards a target that was recently surveyed, and this may increase the average revisit time of the targets.

With the shared positions of other UAVs, each UAV can construct a partition of the targets. In particular, a target is assigned to UAV i if

$$d_{ij}(t) \leq d_{hj}(t), \forall h = 1, \ldots, I, \;\; h \neq i, \tag{17}$$

where $d_{ij}(t)$ is the horizontal distance between target j and UAV i. Let $S_i(t)$ denote the set of targets that are more closer to UAV i.

Algorithm 3 summarizes the navigation algorithm for this scenario. Each UAV determines its partition according to the locations of other UAVs and the targets. Subsequently, it selects its NN in the partition. After that, it starts to pursuit the target. Whenever it views a target other than the NN, it shares the status of this target across the team. When the selected NN has been surveyed, the UAV repeats the above procedures. The difference between Algorithm 3 and Algorithm 2 is an additional operation in Algorithm 3 to update the partition of a UAV according to the UAVs' and targets' locations. Because a UAV needs to calculate the distance from a target to I UAVs, the additional computation complexity is $O(NI)$ for the situation with N targets.

Algorithm 3: The navigation algorithm running at a UAV in the team.

```
while the time has not reached $T$ do
    Determine the partition.
    Choose the NN in the partition.
    while the NN has not been surveyed do
        Keep moving on the planned path.
        Update and share the targets' status if they are surveyed.
    end
end
```

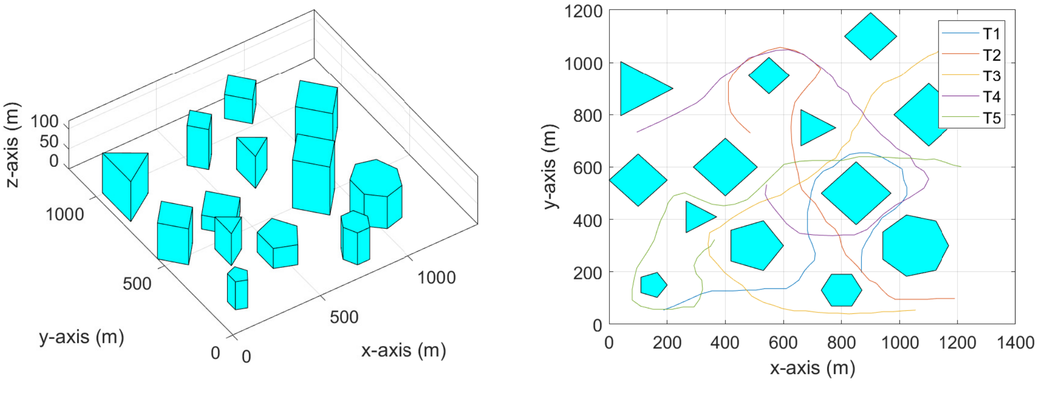
It is worth pointing out that partitioning the targets following (17) assigns a target to exact one group. Subsequently, the NN selected by each UAV is unique. Additionally, considering the existence of buildings and the randomness of the RRT-based path planning, two UAVs' paths may be close. In this case, i.e., when two UAVs are within a range of d_{safe}, one of the UAVs, such as the one with the smaller index, continues to fly as planned, while the other UAV tries to avoid the former UAV. To achieve this, the collision avoidance ability should be embedded. Many available UAV products that have such a function, such as DJI Mavic Pro 2, Mavic 2 Zoom, skydio 2, etc, can be used. Fortunately, this is a common ability in most commercial UAVs. When the two UAVs are at least d_{safe} apart, the latter re-starts to intercept its NN.

Remark 1. *The considered problem in this paper may not be stable. A simple example is that all the targets move away in different directions. Subsequently, the revisit time increases with time.*

5. Simulations

We show the performance of the proposed navigation algorithms. An urban environment is constructed in MATLAB with various buildings, see Figure 7a. The heights of the buildings are between 30 to 120 m. Five targets move on the xy-plane in the environment, and their trajectories are shown in Figure 7b.

We first consider using one UAV to survey the targets. The used parameters are listed, as follows: $V^{max} = 10$ m/s, $\Omega^{max} = 0.5$ rad/s, $z = 65$ m, $R = 50$ m, $T = 500$ s and $\sigma = 0$. It is worth pointing out that the parameters of V^{max} and Ω^{max} depend on the maneuverability of the UAV in use. Our method is not restricted to these parameters. The function $\gamma(t)$ is set to be increasing linearly with time. As will be seen in the below results, because the UAV flight height is taller than some buildings and lower than the others, a UAV needs to avoid the taller buildings but can fly above the lower buildings. To make results more understandable, we assume that during the 500 s, the incidence angle does not change. The fixed value can response to the average value during the 500 s. Subsequently, we can pre-compute the shadow range. Similar to $\mathcal{X}_{building}$, the shadow region is not allowed for the UAV to enter. Given an initial state, the trajectory of the UAV is obtained by applying Algorithm 2. In particular, the randomized Algorithm 1 is used to generate UAV trajectories to intercept the five targets. In our simulation, for the case with five targets, Algorithm 1 (runs on a normal computer with an Intel(R) Core(TM) i7-8565U CPU and 8.00 G RAM) takes less than 1 s to return five random trajectories. Although the onboard processor may not be as powerful as a normal computer, the algorithm can be coded in the more efficient language C. Moreover, even if the practical computation time may be longer than the simulation environment, in practice, the UAV can start to compute the random trajectories before it intercepts the intended target. Experimental verification of the proposed methods is left as our future work.

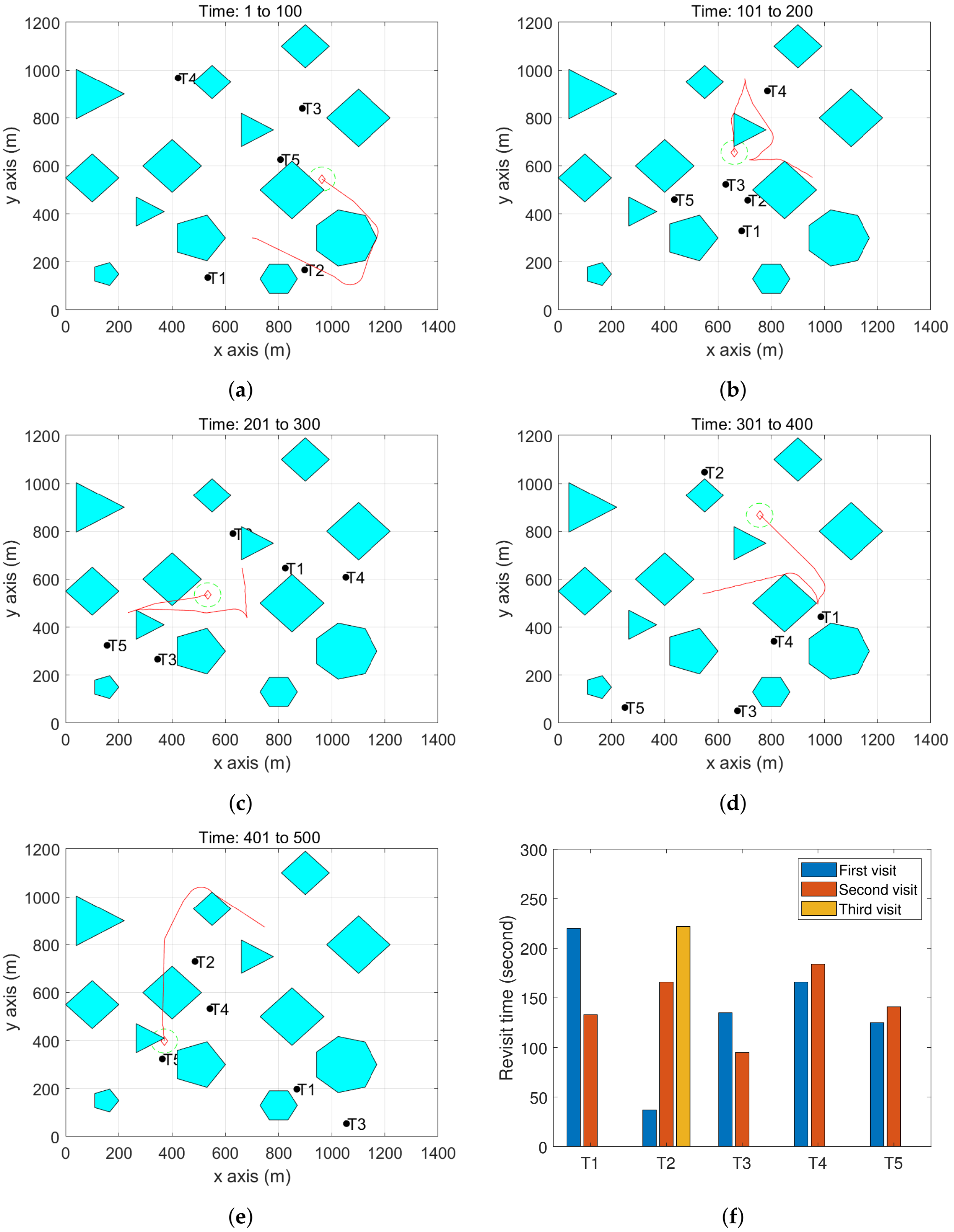


(**a**) (**b**)

Figure 7. (**a**) The simulated urban environment. (**b**) The trajectories of targets.

To make the trajectories visible, we demonstrate the 2D views in Figure 8 for each period of 100 s. We also record the movements of the UAV and the targets in some videos, and links for both 2D and 3D views are available at the caption of Figure 8. From Figure 8f, we can see that the five targets are visited twice or three times in the operation period, and the maximum revisit time is about 220 s. For the same targets movements, we increase the maximum of the UAV linear speed to $V^{max} = 15$ m/s, and the UAV's trajectory is shown in Figure 9. From Figure 9f, we can see that target 1 was visited three times and all the other targets are visited four times in the operation. The maximum revisit time is reduced from 220 s to 180 s.

Finally, we add one more UAV to survey the same targets. Here, $V^{max} = 10$ m/s, and all of the other parameters remain the same as above. We show the trajectories of the two UAVs in Figure 10. From Figure 10f we can see that compared to the results by one UAV, the maximum revisit time is reduced from 220 s to 150 s. The maximum angular speeds, i.e., Ω^{max}, has little impact on the maximum revisit time, since it only influences the UAV trajectory when the UAV makes a turn.



(**a**)

(**b**)

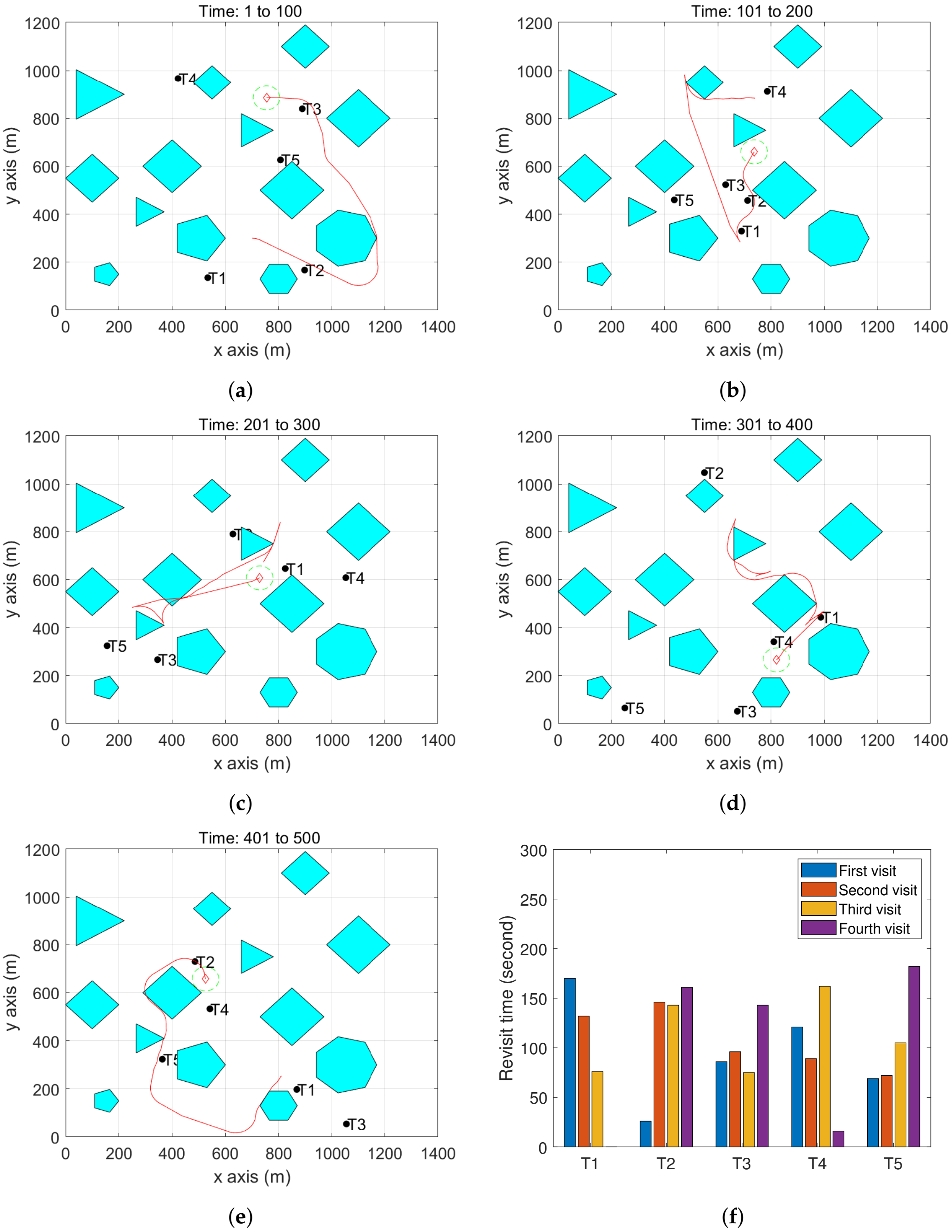
(**c**)

(**d**)

(**e**)

(**f**)

Figure 8. UAV trajectory when flying at 10 m/s. (2D view: https://youtu.be/7cx4jpr0W4I; 3D view: https://youtu.be/eXdSWdH1Yd8). (**a**) The UAV trajectory between 1 and 100 s. (**b**) The UAV trajectory between 101 and 200 s. (**c**) The UAV trajectory between 201 and 300 s. (**d**) The UAV trajectory between 301 and 400 s. (**e**) The UAV trajectory between 401 and 500 s. (**f**) The revisit time of each target during the simulation.



(**a**)

(**b**)

(**c**)

(**d**)

(**e**)

(**f**)

Figure 9. UAV trajectory when flying at 15 m/s. (2D view: https://youtu.be/smvPf-fFdfw; 3D view: https://youtu.be/anSTjbqGoa8).(**a**) The UAV trajectory between 1 and 100 s. (**b**) The UAV trajectory between 101 and 200 s. (**c**) The UAV trajectory between 201 and 300 s. (**d**) The UAV trajectory between 301 and 400 s. (**e**) The UAV trajectory between 401 and 500 s. (**f**) The revisit time of each target during the simulation.

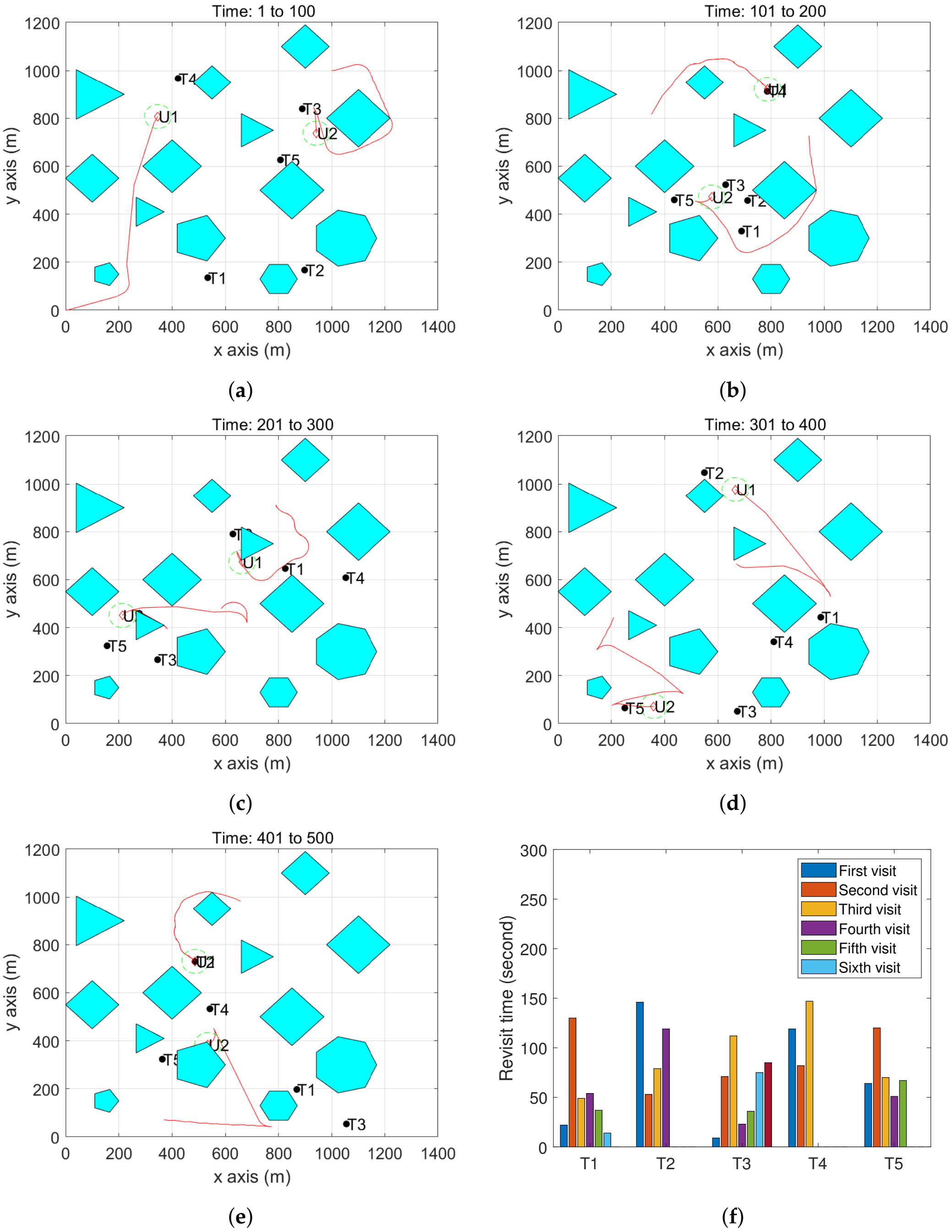


Figure 10. The trajectories of two UAVs flying at 10 m/s. (**a**) The UAV trajectory between 1 and 100 s. (**b**) The UAV trajectory between 101 and 200 s. (**c**) The UAV trajectory between 201 and 300 s. (**d**) The UAV trajectory between 301 and 400 s. (**e**) The UAV trajectory between 401 and 500 s. (**f**) The revisit time of each target during the simulation.

6. Conclusions

We considered using UAVs to conduct periodical surveillance of moving targets. We formulated an optimization problem in order to minimize the target revisit time. We proposed autonomous navigation algorithms for different cases. Because these algorithms are based on RRT, they inherit the advantage of computational efficiency, and they are promising to be applied in real-time. The simulation results showed their effectiveness.

One limitation of the current work lies in the assumption that the targets' trajectories are known. When these trajectories are unavailable, the accuracy of the target position predictions may significantly decrease. Consequently, the UAVs may lose some targets. One future research work is to extend the algorithms by including the searching operation. Another direction is to conduct experiments, since this is the most effective way to practically verify the computational efficiency of the proposed method. To this end, some ground mobile robots can play the role of targets and a small number of UAVs can be tested in order to conduct the surveillance mission in a laboratory environment.

Author Contributions: Conceptualization, A.V.S.; methodology, A.V.S.; software, H.H.; validation, H.H.; formal analysis, H.H.; investigation, H.H.; resources, H.H.; writing—original draft preparation, H.H.; writing—review and editing, A.V.S.; visualization, H.H.; supervision, A.V.S.; project administration, H.H..; funding acquisition, A.V.S. 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

1. Li, X.; Huang, H.; Savkin, A.V. A Novel Method for Protecting Swimmers and Surfers From Shark Attacks Using Communicating Autonomous Drones. *IEEE Internet Things J.* **2020**, *7*, 9884–9894. [CrossRef]
2. 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]
3. Wang, J.; Wang, G.; Hu, X.; Luo, H.; Xu, H. Cooperative Transmission Tower Inspection with a Vehicle and a UAV in Urban Areas. *Energies* **2020**, *13*, 326. [CrossRef]
4. Ješke, P.; Klouček, Š.; Saska, M. Autonomous Compact Monitoring of Large Areas Using Micro Aerial Vehicles with Limited Sensory Information and Computational Resources. In *International Conference on Modelling and Simulation for Autonomous Systesm*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 158–171.
5. Zhang, H.; Feng, T.; Yang, G.H.; Liang, H. Distributed cooperative optimal control for multiagent systems on directed graphs: An inverse optimal approach. *IEEE Trans. Cybern.* **2014**, *45*, 1315–1326. [CrossRef]
6. Sun, Q.; Han, R.; Zhang, H.; Zhou, J.; Guerrero, J.M. A multiagent-based consensus algorithm for distributed coordinated control of distributed generators in the energy internet. *IEEE Trans. Smart Grid* **2015**, *6*, 3006–3019. [CrossRef]
7. Sun, J.; Wang, Z. Event-triggered consensus control of high-order multi-agent systems with arbitrary switching topologies via model partitioning approach. *Neurocomputing* **2020**, *413*, 14–22. [CrossRef]
8. Huang, H.; Savkin, A.V.; Ding, M.; Huang, C. Mobile robots in wireless sensor networks: A survey on tasks. *Comput. Netw.* **2019**, *148*, 1–19. [CrossRef]
9. Huang, H.; Savkin, A.V.; Ni, W. Energy-Efficient 3D Navigation of a Solar-Powered UAV for Secure Communication in the Presence of Eavesdroppers and No-Fly Zones. *Energies* **2020**, *13*, 1445. [CrossRef]
10. 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.* **2019**, *49*, 2362–2372. [CrossRef]
11. Wang, S.; Jiang, F.; Zhang, B.; Ma, R.; Hao, Q. Development of UAV-Based Target Tracking and Recognition Systems. *IEEE Trans. Intell. Transp. Syst.* **2019**, *21*, 3409–3422. [CrossRef]
12. Wise, R.; Rysdyk, R. UAV coordination for autonomous target tracking. In Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit, Keystone, CO, USA, 21–24 August 2006; p. 6453.
13. Oh, H.; Turchi, D.; Kim, S.; Tsourdos, A.; Pollini, L.; White, B. Coordinated Standoff Tracking Using Path Shaping for Multiple UAVs. *IEEE Trans. Aerosp. Electron. Syst.* **2014**, *50*, 348–363. [CrossRef]
14. Zhang, M.; Liu, H.H. Cooperative tracking a moving target using multiple fixed-wing UAVs. *J. Intell. Robot. Syst.* **2016**, *81*, 505–529. [CrossRef]
15. Tang, H.; Miller-Hooks, E. Solving a generalized traveling salesperson problem with stochastic customers. *Comput. Oper. Res.* **2007**, *34*, 1963–1987. [CrossRef]
16. Toth, P.; Vigo, D. *The Vehicle Routing Problem*; SIAM: Philadelphia, PA, USA, 2002.
17. Savkin, A.V.; 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] [PubMed]

18. Savkin, A.V.; Huang, H. Asymptotically Optimal Deployment of Drones for Surveillance and Monitoring. *Sensors* **2019**, *19*, 2068. [CrossRef] [PubMed]
19. Huang, H.; Savkin, A.V. Reactive 3D deployment of a flying robotic network for surveillance of mobile targets. *Comput. Netw.* **2019**, *161*, 172–182. [CrossRef]
20. Cordeau, J.F.; Ghiani, G.; Guerriero, E. Analysis and branch-and-cut algorithm for the time-dependent travelling salesman problem. *Transp. Sci.* **2014**, *48*, 46–58. [CrossRef]
21. Helvig, C.S.; Robins, G.; Zelikovsky, A. The moving-target traveling salesman problem. *J. Algorithms* **2003**, *49*, 153–174. [CrossRef]
22. Jiang, Q.; Sarker, R.; Abbass, H. Tracking moving targets and the non-stationary traveling salesman problem. *Complex. Int.* **2005**, *11*, 171–179.
23. Karaman, S.; Walter, M.R.; Perez, A.; Frazzoli, E.; Teller, S. Anytime motion planning using the RRT. In Proceedings of the 2011 IEEE International Conference on Robotics and Automation, Shanghai, China, 9–13 May 2011; pp. 1478–1483.
24. Kuwata, Y.; Teo, J.; Fiore, G.; Karaman, S.; Frazzoli, E.; How, J.P. Real-time motion planning with applications to autonomous urban driving. *IEEE Trans. Control. Syst. Technol.* **2009**, *17*, 1105–1118. [CrossRef]
25. Salzman, O.; Halperin, D. Asymptotically near-optimal RRT for fast, high-quality motion planning. *IEEE Trans. Robot.* **2016**, *32*, 473–483. [CrossRef]
26. Wang, X.; Luo, X.; Han, B.; Chen, Y.; Liang, G.; Zheng, K. Collision-free path planning method for robots based on an improved rapidly-exploring random tree algorithm. *Appl. Sci.* **2020**, *10*, 1381. [CrossRef]
27. Wu, Y.; Low, K.H. An Adaptive Path Replanning Method for Coordinated Operations of Drone in Dynamic Urban Environments. *IEEE Syst. J.* **2020**, 1–12. [CrossRef]
28. Li, H.; Savkin, A.V. Wireless sensor network based navigation of micro flying robots in the industrial internet of things. *IEEE Trans. Ind. Inform.* **2018**, *14*, 3524–3533. [CrossRef]
29. Kang, Y.; Hedrick, J.K. Linear tracking for a fixed-wing UAV using nonlinear model predictive control. *IEEE Trans. Control. Syst. Technol.* **2009**, *17*, 1202–1210. [CrossRef]
30. Wang, C.; Savkin, A.V.; Garratt, M. A strategy for safe 3D navigation of non-holonomic robots among moving obstacles. *Robotica* **2018**, *36*, 275–297. [CrossRef]
31. Savkin, A.V.; Huang, H. Optimal Aircraft Planar Navigation in Static Threat Environments. *IEEE Trans. Aerosp. Electron. Syst.* **2017**, *53*, 2413–2426. [CrossRef]
32. Savkin, A.V.; Huang, H.; Ni, W. Securing UAV Communication in the Presence of Stationary or Mobile Eavesdroppers via Online 3D Trajectory Planning. *IEEE Wirel. Commun. Lett.* **2020**, *9*, 1211–1215. [CrossRef]
33. Savkin, A.V.; Huang, H. Bio-inspired Bearing Only Motion Camouflage UAV Guidance for Covert Video Surveillance of a Moving Target. *IEEE Syst. J.* **2020,** 1–4. Available online: https://ieeexplore.ieee.org/document/9228876 (accessed on 23 October 2020).
34. Klesh, A.T.; Kabamba, P.T. Solar-powered aircraft: Energy-optimal path planning and perpetual endurance. *J. Guid. Control. Dyn.* **2009**, *32*, 1320–1329. [CrossRef]
35. Zeng, Y.; Xu, J.; Zhang, R. Energy Minimization for Wireless Communication With Rotary-Wing UAV. *IEEE Trans. Wirel. Commun.* **2019**, *18*, 2329–2345. [CrossRef]
36. Matveev, A.S.; Teimoori, H.; Savkin, A.V. A method for guidance and control of an autonomous vehicle in problems of border patrolling and obstacle avoidance. *Automatica* **2011**, *47*, 515–524. [CrossRef]
37. 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]
38. Savkin, A.V.; Wang, C. A simple biologically inspired algorithm for collision-free navigation of a unicycle-like robot in dynamic environments with moving obstacles. *Robotica* **2013**, *31*, 993–1001.

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