



Article

A Hierarchical Heuristic Architecture for Unmanned Aerial Vehicle Coverage Search with Optical Camera in Curve-Shape Area

Lanjun Liu ¹ , Dechuan Wang ¹, Jiabin Yu ^{2,3} , Peng Yao ^{1,*}, Chen Zhong ¹ and Dongfei Fu ¹

¹ College of Engineering, Ocean University of China, Qingdao 266100, China; hdliuj@ouc.edu.cn (L.L.); wangdechuan@stu.ouc.edu.cn (D.W.); zhongchen4917@stu.ouc.edu.cn (C.Z.); fudongfei@ouc.edu.cn (D.F.)

² School of Computer and Artificial Intelligence, Beijing Technology and Business University, Beijing 100048, China; yujiabin@th.btbu.edu.cn

³ Key Laboratory of Industrial Internet and Big Data, China National Light Industry, Beijing 100048, China

* Correspondence: yp@ouc.edu.cn

Abstract: This paper focuses on the problem of dynamic target search in a curve-shaped area by an unmanned aerial vehicle (UAV) with an optical camera. Our objective is to generate an optimal path for UAVs to obtain the maximum detection reward by a camera in the shortest possible time, while satisfying the constraints of maneuverability and obstacle avoidance. First, based on prior qualitative information, the original target probability map for the curve-shaped area is modeled by Parzen windows with 1-dimensional Gaussian kernels, and then several high-value curve segments are extracted by density-based spatial clustering of applications with noise (DBSCAN). Then, given an example that a target floats down river at a speed conforming to beta distribution, the downstream boundary of each curve segment in the future time is expanded and predicted by the mean speed. The rolling self-organizing map (RSOM) neural network is utilized to determine the coverage sequence of curve segments dynamically. On this basis, the whole path of UAVs is a successive combination of the coverage paths and the transferring paths, which are planned by the Dubins method with modified guidance vector field (MGVF) for obstacle avoidance and communication connectivity. Finally, the good performance of our method is verified on a real river map through simulation. Compared with the full sweeping method, our method can improve the efficiency by approximately 31.5%. The feasibility is also verified through a real experiment, where our method can improve the efficiency by approximately 16.3%.

Keywords: dynamic target search; unmanned aerial vehicle (UAV); curve-shaped area; curve segment; coverage path; transferring path



Citation: Liu, L.; Wang, D.; Yu, J.; Yao, P.; Zhong, C.; Fu, D. A Hierarchical Heuristic Architecture for Unmanned Aerial Vehicle Coverage Search with Optical Camera in Curve-Shape Area. *Remote Sens.* **2024**, *16*, 1502. <https://doi.org/10.3390/rs16091502>

Academic Editors: Andrea Nascetti, Lu Ma, Jianmin Yang, Yiping Duan, Qin Lu and Gang Qiao

Received: 13 March 2024

Revised: 22 April 2024

Accepted: 22 April 2024

Published: 24 April 2024



Copyright: © 2024 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

Over the past few decades, unmanned aerial vehicles (UAVs) have been increasingly used in many applications, including mapping [1], surveillance [2], and tracking [3]. Compared with ground vehicles or surface vehicles, UAVs have better maneuverability, wider field of view (FOV), and less restriction by terrain [4], and thus they are very suitable for target search missions. This paper particularly focuses on using UAVs to search a dynamic target located in a curve-shaped area, e.g., river, coastline, road, etc. Assuming that a person falls into a flowing river, their probability of survival will decrease rapidly over time. In addition, because the person flows downstream by the water current, the dynamically changing information of the person will make the search and rescue mission more difficult. Therefore, UAVs need to cover and detect the area of interest as quickly as possible. The dynamic target search problem here can be formulated as optimizing the searching path of UAVs, which aims to obtain the maximum detection reward in the shortest possible time under the constraints of UAV maneuverability and obstacle avoidance. The work in [5]

proposes two methods, namely Equivalent Area Iteration (EAI) and Probability Estimation Diffusion (PED), for situations where environmental information is relatively accurate with certain errors, respectively. However, there is no specific quantification of the relatively accurate range of environmental information. It should be noted that, although the target search problem also needs to plan the UAV path, it is quite different from the traditional path planning problem between the starting point and destination for autonomous navigation. In [6], a new DAALO method based on natural-inspired ALO is proposed to solve the problem of UAV route planning. The DAALO method can plan flight paths faster and smoother, but it is only applicable to single aircrafts and has not been extended to multi-UAVs.

Researchers have already presented various methods for searching targets located in a 2D wide area. For static target search, the work in [7] proposes a two-step search and investigation algorithm for multi-UAV systems for searching forest fires. For dynamic target search, the work in [8] proposes a search improvement algorithm based on the theory of dynamical system traversal for searching drift areas on the ocean surface. Although the proposed method improves the search efficiency by an order of magnitude, this improvement is achieved by abandoning the optimal utilization of fuel and resources. Without any priori information, UAVs can use a geometry method or a completely random way to perform search missions. In [9], the researchers propose a progressively spiral-out-based method for UAV searching for ground moving targets, but the method itself requires a sensor detection probability of 100%. In [10], an online distributed algorithm is proposed for UAV search of targets, taking into account energy loss, but the accuracy of the algorithm will decrease as the search area expands. The work in [11] proposes a lost target search algorithm based on recursive regional clustering and target trajectory prediction in marine environments. However, the efficiency of these patterns will be low if some prior information is already known in a complex search problem. In this case, graph-based heuristic methods (such as biologically inspired swarm intelligence [12], the greedy method [13,14], receding-horizon approach, coverage control method [15], mixed-integer linear programming [16], and reinforcement learning) are widely used, relying on various graphs (such as a target probability map, information density map, pheromone map). For example, based on the target probability map of the search area, our previous work [14] models the Gladius bio-inspired neural network to solve the target search problem and introduces the Gaussian mixture model (GMM)-based global guidance mechanism to improve the searching efficiency. The different flight heights of UAVs can also affect the resolution of camera sensors, so the work in [17] proposes a method for determining the optimal height of UAVs by considering their power consumption, high-value areas, and image resolution. After determining the altitude, the motion trajectory of the UAV only considers a 2D wide space.

As mentioned above, the existing searching methods mainly focus on target search mission in a 2D wide space, where UAVs can cover the entire region completely or detect the area of interest hierarchically according to prior information. This paper, however, emphasizes a dynamic target located in a curve-shaped area, which can be modeled as abstract curves subjected to geographical constraints resulting in the limitation of the UAV's coverage motion. Hence, the existing 2D searching methods cannot be utilized for this mission directly [18]. In fact, this mission is similar to the road-network search path planning problem to some extent [19], but the existing methods mainly guide UAVs to cover the areas of interest without any quantified heuristic architecture [20,21]. Inspired by the hierarchical heuristic architecture for target searches in a 2D wide area, our previous work has achieved the coverage search of stationary targets in a river by extracting and sorting some high-value curve segments [20]. Although the scenario is simple, this proposed scheme has shown its huge potential and high efficiency in searching targets located in a curve-shaped area.

Based on the above analysis, a three-layer heuristic framework is used to solve the dynamic target search problem in a curve-shaped area. First, based on prior information,

the curve-shaped area should be modeled and several high-value curve segments can be extracted. Second, the optimal visiting sequence of segments for UAVs should be determined. Third, the whole path of the UAV is planned by combining transferring paths and coverage paths. For the first step, the traditional distance-based clustering methods, such as GMM or K-means [21], require some knowledge in advance of the number of clusters and can only handle regularly shaped clusters. Density-based spatial clustering of applications with noise (DBSCAN) is a density-based clustering method widely used in 2D clustering problems, and it does not require any settings of the number of clusters and can discover arbitrarily shaped clusters [22]. For the second step, it leads to a variant of the well-known traveling salesman problem [23], and many methods have been proposed for this, e.g., approximation insertion, generalized solution [24], self-organizing map (SOM) neural network [25]. But, these methods may not be suitable for the target search problem in a dynamically changing area. For the third step, the constraints of UAV maneuverability and obstacle avoidance need to be considered at the same time in a complex scenario.

There has been a lot of research on the autonomous control of UAVs in regular cases [26,27]. This paper aims to solve the problem of dynamic target searching in a curve-shaped area, which can be modeled as a path planning problem for UAVs to obtain the maximum detection reward. The main contributions are as follows.

- (1) Based on the original target probability map generated by Parzen windows with 1D Gaussian kernels, several high-value curve segments can be extracted by DBSCAN.
- (2) Considering the dynamic characteristics of drifting targets, the boundary of each curve segment is predicted, and the coverage sequence of curve segments is dynamically determined by the rolling SOM (RSOM) neural network.
- (3) The whole path of UAVs is a successive combination of coverage paths and transferring paths planned by the Dubins method with modified guidance vector field (MGVF) satisfying the constraints of maneuverability and obstacle avoidance.

The paper is organized as follows. Section 2 gives the problem formulation. Section 3 proposes the three-layer hierarchical heuristic architecture. Section 4 shows the experimental results. Section 5 draws the conclusion.

2. Problem Formulation

This paper assumes that an UAV equipped with a gimbaled camera performs the search task and takes a water falling event as an example that a person falls into a river and waits to be rescued. The real position of the person is undetermined. As time passes, the area of interest will gradually expand since the person may be carried downstream by the current, while the person's probability of survival will decrease rapidly. Therefore, the UAV must detect the area of interest as quickly as possible to find the person.

2.1. Simplified UAV Model

Suppose that the UAV flies at a fixed altitude, and its kinematic model can be described by

$$\begin{cases} \dot{x} = v \cos \varphi \\ \dot{y} = v \sin \varphi \\ \dot{\varphi} = \omega \end{cases} \quad (1)$$

where $(x, y)^T$ is the inertial position, v the speed, φ the heading angle, and ω the turning rate with constraint $|\omega| \leq \omega_{\max}$. In order to ensure the confidence of target detection and decrease the cruising time, the flying speed when detecting the target (i.e., the UAV is directly covering the curve-shaped area) is $v_{\text{cov}} = v_{\min}$, and in other cases it is the maximum cruising speed $v_{\text{cru}} = v_{\max}$. Here, the change between v_{cov} and v_{cru} is ignored. In addition, this paper assumes that the UAV's maneuverability can satisfy the geographical curvature of the curve-shaped area, and thus only the maneuverability during cruising is considered with the minimum turning radius $R_{\min} = v_{\text{cru}}/\omega_{\max}$.

2.2. Environment Model

This paper takes a portion of the Weihe river in Weinan City, Shaanxi Province, China, as an example of a curve-shaped area, as shown in Figure 1. Usually the detection range of the camera on a UAV is larger than the width of river, so the river can be modeled as a curve with the appropriate resolution. From Figure 1b, the whole curve can be divided into $M = L/L_s$ cells uniformly with respect to the forward distance s , where L is the total length of the curve and L_s is the cell length. Thus, the 2D position $\mathbf{x}_m = (x_m, y_m)^T$ of cell m can be reduced to 1D expression by $s_m = m \cdot L_s$. This paper utilizes the target probability map as the basic information for target searching, which will be described in Section 3. Each cell has a probability $p(\mathbf{x}_m) \in [0, 1]$ determined by prior information and it represents the detection payoff of this cell. Additionally, if only one target exists, the volume of the whole probability map is equal to 1:

$$\sum_{m=1}^M p(\mathbf{x}_m) = 1 \quad (2)$$

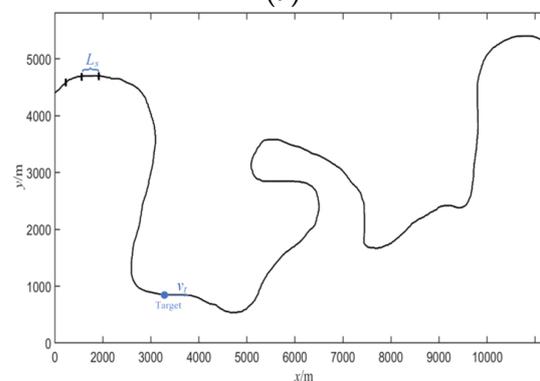
We define $z_m^{1:t} = \{z_m^1, \dots, z_m^t\}$ as the historical detection events of cell m at time t . Using Bayes' rule, the target probability $p(\mathbf{x}_m | z_m^{1:t})$ after detection can be updated by

$$p(\mathbf{x}_m | z_m^{1:t}) = \lambda \cdot p(\mathbf{x}_m | z_m^{1:t-1}) \cdot \left(1 - g(z_m^t | \mathbf{x}_m)\right) \quad (3)$$

where $p(\mathbf{x}_m | z_m^{1:t-1})$ at the initial time $t = 1$ is equal to the prior value of $p(\mathbf{x}_m)$, $\lambda = 1 / \sum_{m=1}^M (p(\mathbf{x}_m | z_m^{1:t-1}) \cdot (1 - g(z_m^t | \mathbf{x}_m)))$ is a normalization parameter, and $g(z_m^t | \mathbf{x}_m)$ is the detection confidence related to the observation ability of the camera. This paper assumes the sensor performance to be perfect, i.e., when the UAV completes the coverage of cell m at time t , a new observation occurs with $g(z_m^t | \mathbf{x}_m) = 1$ and $z_m^t = 1$, and otherwise $g(z_m^t | \mathbf{x}_m) = 0$ with $z_m^t = 0$.



(a)



(b)

Figure 1. An example of curve-shaped area. (a) Partial GIS map of Weihe river in Weinan, Shaanxi, China. URL: <https://ditu.amap.com/place/B0391001MN> (accessed on 1 March 2024). (b) Abstracted curve of the river.

In addition, since the target drifts downstream by the water current in the river, the prior target probability map will be changed as well. This paper uses the beta distribution to describe the probability of drifting speed:

$$f(v_t) = \frac{c}{v_{t\max}} \left(\frac{v_t}{v_{t\max}} \right)^{a-1} \left(1 - \frac{v_t}{v_{t\max}} \right)^{b-1} \quad (4)$$

where $v_t \in [0, v_{t\max}]$ denotes the drifting speed of the target, and $a, b, c = (a + b - 1)! / ((a - 1)!(b - 1)!)$ are parameters of beta distribution. Based on the Chapman–Kolmogorov equation, the target probability can further be predicted by

$$p(\mathbf{x}_m | z_m^{1:t}) = \sum_{l=1}^M \left(p(\mathbf{x}_l | z_l^{1:t}) \cdot \bar{f}_l(v_t) \right) \quad (5)$$

where the value of v_t is determined by the transition from any cell l to the current cell m , and $\bar{f}_l(v_t)$ is an integration of $f(v_t)$ in the relevant cell. This prediction model has good convergence and low complexity.

2.3. Problem Description

The single conditional probability of non-detection for UAVs can be obtained by the volume of multiplying the target probability and the sensor non-detection confidence:

$$\bar{D}^t = \sum_{m=1}^M \left(p(\mathbf{x}_m | z_m^{1:t-1}) \cdot (1 - g(z_m^t | \mathbf{x}_m)) \right) \quad (6)$$

Thus, the probability that the target has been found so far, i.e., the cumulative detection reward, is given by

$$D^{1:t} = 1 - \prod_{i=1}^t \bar{D}^i \quad (7)$$

It should be noted that $\bar{D}^t < 1$ and $D^{1:t} \rightarrow 1$ with $t \rightarrow \infty$. Next, we define ζ as a candidate path. Hence, the problem of dynamic target search can be formulated as finding the optimal path ζ^* attached with the maximum searching efficiency (i.e., the maximum detection reward in the shortest possible time) under the constraints of UAV maneuverability and obstacle avoidance:

$$\begin{aligned} \zeta^* &= \operatorname{argmax}_{\zeta} (D^{1:t}(\zeta) / t), \\ \text{s.t. } \zeta \cap S_F &= \emptyset, R \geq R_{\min} \end{aligned} \quad (8)$$

where S_F denotes the obstacle region, and R is the turning radius of the UAV.

3. Three-Layer Hierarchical Heuristic Architecture

This paper utilizes a three-layer heuristic architecture to solve the dynamic target search problem in a curve-shaped area, as shown in Figure 2. First, the original target probability map for the curve-shaped area is generated by the Parzen windows method, and on this basis, some high-value curve segments are extracted by DBSCAN preliminarily. Second, given an example of a drifting target along the river, the downstream boundaries of curve segments in the future time can be updated and predicted by the mean speed of beta distribution, and the coverage sequence of these high-value curve segments is determined by RSOM dynamically. Third, the transferring path between curve segments (or the transferring path between the starting point and curve segment) is planned by the Dubins method with MGVE, and thus the whole path is obtained by connecting the transferring paths and the coverage paths (i.e., the curve segments) successively.

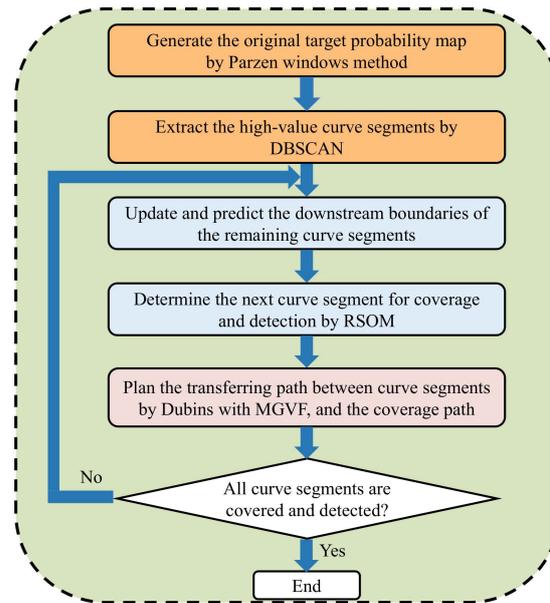


Figure 2. Three-layer heuristic architecture.

3.1. Preliminary Processing of Curve-Shape Area

3.1.1. Generation of Original Target Probability Map

In most cases, there usually exists prior qualitative information about the target to be searched, which may be obtained from historical incidents, reports from commons, inference from experts, etc. Suppose there are J discrete data and each one $Q = \{(s_j, \gamma_j, \lambda_j)\}$, $j \in \{1, \dots, J\}$ has three attributes, i.e., the position s_j , the reliability γ_j , and the accuracy λ_j . As seen in Section 2.2, the position s_j can be reduced to a 1D expression with respect to forward distance. The reliability $\gamma_j \in [0, 1]$ denotes the confidence of this prior information. The inverse of accuracy λ_j denotes the possible region around s_j . Thus, a standard 1D Gaussian function $G(s_j, 1/\lambda_j)$ with reliability γ_j can be used to describe this prior information.

Given that the prior information matches the Gaussian distribution, it is appropriate to utilize Parzen windows with Gaussian kernels to build the original target probability map. Because the amount of the prior qualitative data is usually insufficient, each item should first be expanded to $E_j = E_{\max} \gamma_j$ samples from its corresponding Gaussian function. Then, the original target probability map can be generated by summing the Gaussian kernels of all the samples:

$$p(s) = \frac{1}{E} \sum_{e=1}^E G_e(s_e, \sigma) \quad (9)$$

where $E = \sum_{j=1}^J E_j$ denotes the number of samples after expansion, s_e the position of each sample, σ the standard deviation for each Gaussian kernel calculated by

$$\sigma = 0.5E^{-1/5} \min(\sigma_{\text{std}}, IQR/1.34) \quad (10)$$

where IQR denotes the difference between the 75th and 25th percentile of samples, and σ_{std} the standard deviation of samples. In addition, the continuous form of (9) should be discretized into $p(x_m)$ for ease of processing.

3.1.2. Extraction of High-Value Curve Segments by DBSCAN

DBSCAN is a density-based clustering algorithm [24] that is suitable for extracting high-value curve segments from the original target probability map. First, we choose

N points ($N \geq M$) to form the training samples. To reflect the prior information objectively, the number of points placed in the center of each cell is

$$N_m = [p(x_m) \cdot N] \quad (11)$$

where the symbol $[\]$ denotes the rounding operation. Additionally, the number of points in some cells needs to be fine-tuned to satisfy $\sum_{m=1}^M N_m = N$.

In DBSCAN, we define R_η as the neighborhood radius for each cell in the 1D space, which is an integer multiple of L_s . We define $minPts$ as the minimum number of points in the neighborhood. The algorithm works as follows. First, an unmarked cell m from M cells is selected randomly. If there exists at least $minPts$ points in the neighborhood of cell m , this cell is marked as a core one. Then, all the cells in the neighborhood of cell m are added to the same high-value segment. These steps are repeated until all the cells have been visited. Those cells which are not core cells but fall within the neighborhood of a core cell are considered as boundary cells, since they are not sufficient to form a segment but can be assigned to another segment. Figure 3 illustrates the principle of the DBSCAN algorithm in a 1D space, where the red core cells and the blue boundary cells form a cluster. Finally, each cluster corresponds to an extracted high-value curve segment.

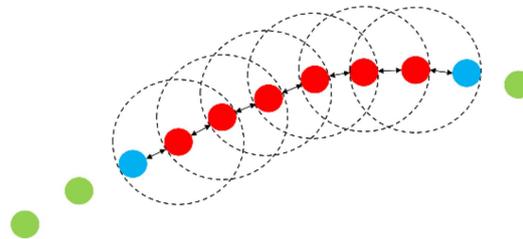


Figure 3. The principle of DBSCAN algorithm in 1D space.

3.2. Coverage Sequence of Curve Segments

3.2.1. Prediction of Downstream Boundary by Beta Distribution

The geographical restriction for the target makes only the coverage motion of the UAV along the curve be valuable. Thus, the UAV should cover and detect the high-value curve segments in an optimal order to obtain the maximum detection reward in the shortest time. However, the target probability map is dynamically changing by (4) and (5), and consequently the extracted high-value curve segments are changing as well. Although the DBSCAN method could be used repeatedly to update the curve segments at each moment, it is too time consuming. Here, this paper takes an example of a drifting target at the speed conforming to beta distribution in a river and utilizes its dynamic characteristics to update and predict the boundaries of the original high-value curve segments directly.

As described in (4), although the speed of a dynamic target is uncertain at each moment, its mean speed tends to be constant:

$$v_{\text{mean}} = \frac{av_{\text{tmax}}}{a+b} \quad (12)$$

Considering the dynamic characteristics of a drifting target, the upstream boundary of the curve segment is unchanged, but the downstream boundary will extend a distance of mean speed multiplied by time.

3.2.2. Sorting Curve Segments by RSOM

Unlike most neural networks, the SOM neural network only contains an input layer and output layer [28]. In this paper, each boundary point of the high-value segment denotes a neuron of the input layer. The starting point of the UAV is also an input neuron. Additionally, to ensure the region constraints, i.e., the upstream and downstream boundary points of the same segment are continuously accessed by the UAV, we add some virtual

points evenly distributed in each curve segment as neurons of the input layer. The number of virtual points inserted in each segment is

$$\zeta_k = \left\lceil \zeta \cdot L_k / \sum_{k=1}^K L_k \right\rceil \quad (13)$$

where ζ is a constant denoting the total number of virtual points, L_k denotes the length of a curve segment, and K is the number of curve segments. Each neuron of the input layer is denoted by its coordinates $P_k = (x_k, y_k)^T, k \in \{1, \dots, \zeta + 2K + 1\}$. The spatial relationship of these $\zeta + 2K + 1$ points is the pattern that the neural network needs to learn, and finally a circular topological structure of neurons $c_h, h \in \{1, \dots, H\}$ in the output layer will be obtained. It is noticed that H is usually much larger than $\zeta + 2K + 1$, and the initial positions of output neurons are set randomly. Because the upstream/downstream boundary points of the same curve segment can be seen as adjacent with the help of virtual points, and each output neuron c_h gets close to any one input point P_k gradually, the final circular topological structure with neuron connections can reflect the coverage sequence of curve segments, as well as the entry/exit point for each curve segment. Figure 4 illustrates the structure of the SOM, where each neuron of the output layer is connected with all the neurons of the input layer.

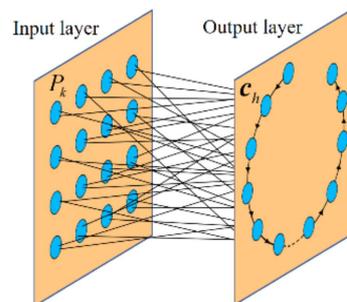


Figure 4. The structure of SOM neural network.

In this paper, three evaluation factors including the transferring time, the coverage time, and the reward are used in the winner selection step of the SOM framework. The shortest transferring path between the input neuron and output neuron under the constraint of UAV maneuverability can be obtained using the Dubins method, where the heading of input/output neuron is determined by the geographical restriction. Then, the transferring time can be calculated by

$$T_{P_k \rightarrow c_h} = \frac{d_{P_k \rightarrow c_h}}{v_{cru}} \quad (14)$$

where $d_{P_k \rightarrow c_h}$ denotes the length of Dubins path. As the UAV should cover the corresponding curve segment after arriving at the boundary point, the coverage time can be calculated by

$$T_h = \frac{L_h}{v_{cov}} \quad (15)$$

and L_h denotes the length of the curve segment which c_h is closest to. And the reward of the curve segment is the sum of the target probability of cells contained in this segment:

$$R_h = \sum_{x_m \in S_h} p(x_m) \quad (16)$$

where S_h denotes the curve segment. Therefore, for any input neuron P_k , the winner neuron of the output layer can be selected by

$$c_{win} = \arg \max_{h \in \{1, \dots, H\}} \frac{R_h}{T_{P_k \rightarrow c_h} + T_h} \quad (17)$$

Then, the neighbor neurons of c_{win} can be updated by

$$c'_h = c_h + \delta^t \cdot \exp\left(-d_h^2 / \sigma_c^{t2}\right) \cdot (c_{win} - c_h) \quad (18)$$

where d_h is the distance between the winner neuron and the neighbor neuron, δ^t is a learning factor, and σ_c^t is the radius of neighborhood decayed over iteration:

$$\begin{aligned} \delta^t &= \gamma_\delta \cdot \delta^{t-1} \\ \sigma_c^t &= \gamma_{\sigma_c} \cdot \sigma_c^{t-1} \end{aligned} \quad (19)$$

where γ_δ and γ_{σ_c} denote the decay factors. The above process of winner selection and weight update repeats until reaching the maximum number of iterations or the threshold of neighborhood radius.

The traditional SOM is more suitable for a static scenario assuming all the curve segments remain unchanged. However, the boundaries of curve segments in this paper will expand slowly in real time, so a rolling strategy is introduced into the SOM. The proposed RSOM algorithm consists of four main steps. First, the boundary points of the remaining curve segments are updated and predicted. Then, for the remaining curve segments, we generate some virtual points according to the length of segment. Then, the optimal coverage sequence for these segments is planned by the iterative process of winner selection and weight update in the SOM. Finally, the UAV only covers the first one of these sequenced segments and updates the current state. The above steps repeat until all the curve segments are covered.

3.3. UAV Path Planning

3.3.1. UAV Path in Free Environment

Based on the sequenced curve segments, the whole path of the UAV is actually a successive combination of transferring paths and coverage paths. The curve segment can be used as the coverage path directly. The Dubins path can be taken as the transferring path between the starting point and the first curve segment, or that between two curve segments (i.e., the boundary points with specified heading angle under geographical restriction). Figure 5 illustrates a scenario including three high-value curve segments S_1 , S_2 and S_3 . Obviously, the transferring paths (red) and the coverage paths (blue) can combine the whole path $c = \{S_{0 \rightarrow 1}, S_1, S_{1 \rightarrow 2}, S_2, S_{2 \rightarrow 3}, S_3\}$ in sequence.

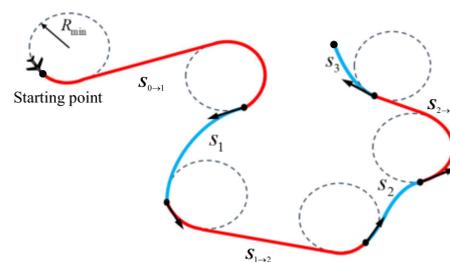


Figure 5. Illustration of Dubins transferring paths and coverage paths.

3.3.2. UAV Path in Obstacle Environment

In some complicated cases, there may be some obstacles that bring risk for the UAV. When obstacles exist on the transferring path, for example, the UAV cannot avoid them along the original Dubins path. Thus, this paper utilizes the MGVF method to achieve the Dubins path tracking and obstacle avoidance simultaneously, assuming that the UAV is equipped with lidar sensors. First, to achieve Dubins path tracking in a free environment, the original velocity of the UAV can be obtained by

$$v = v_{cru} \cdot \frac{(d_D/d_0 \cdot \delta + \tau)}{|d_D/d_0 \cdot \delta + \tau|} \quad (20)$$

where d_D denotes the vertical distance between the current UAV position and the Dubins path, d_0 is a constant, $\delta = (-\partial\zeta/\partial x, \partial\zeta/\partial y)^T$ is the normal vector pointing towards the path, and $\tau = (-\partial\zeta/\partial y, \partial\zeta/\partial x)^T$ is the tangent vector along the path.

Then, the original velocity needs to be adjusted when the UAV detects some obstacles. Considering the distance and orientation of the point cloud from lidar, the threat level for each point on the obstacle is calculated by

$$\Gamma = \frac{1 + \cos \theta}{2d_{\text{obs}}} \quad (21)$$

where d_{obs} denotes the distance between the UAV's position and the lidar's sampling point, and θ is the angle between the UAV's heading and the lidar's ray direction. Then, we select the obstacle point $P_o = (x_o, y_o)^T$ with the highest threat level to define the normal vector \mathbf{n} and tangent vector \mathbf{t} by

$$\mathbf{n} = \frac{1}{\sqrt{(x_o - x)^2 + (y_o - y)^2}} \begin{bmatrix} x_o - x \\ y_o - y \end{bmatrix} \quad (22)$$

$$\mathbf{t} = \frac{1}{\sqrt{(x_o - x)^2 + (y_o - y)^2}} \begin{bmatrix} y - y_o \\ x_o - x \end{bmatrix} \quad (23)$$

Thus, the correction matrix is defined as

$$\bar{\mathbf{M}} = \mathbf{I} - \frac{\mathbf{n}\mathbf{n}^T}{(e^{d_{\text{obs}}})^{\frac{1}{\rho}} |\mathbf{n}|^2} + \frac{\mathbf{t}\mathbf{t}^T}{(e^{d_{\text{obs}}})^{\frac{1}{\varepsilon}} |\mathbf{t}||\mathbf{n}|} \quad (24)$$

where \mathbf{I} denotes the unit matrix, and ρ and ε are constants. After that, the desired velocity of the UAV can be corrected by

$$\bar{\mathbf{v}} = v_{\text{cru}} \frac{\bar{\mathbf{M}}\mathbf{v}}{|\bar{\mathbf{M}}\mathbf{v}|} \quad (25)$$

Using the MGVF method, both the stability of the Dubins path tracking and the safety of obstacle avoidance can be ensured.

3.4. Constraint of UAV Communication Network

The above solution framework is also suitable for multi-UAVs, once the subregions are allocated to corresponding UAVs [29]. In this case, the connectivity of the multi-UAV communication network should be ensured to achieve the continuous information transmission. Here, we quantify the communication connectivity between UAVs by the successful transmission probability. First, we suppose that UAV i sends a message to UAV j under the sending power P_i and the transmitter output noise with average power σ_j^2 , so the received signal-to-noise ratio (SNR) of the signal can be defined as

$$\Gamma_{ij} = \frac{P_i G_{ij}}{\sigma_j^2} \quad (26)$$

The channel gain is

$$G_{ij} = \frac{C_{ij} |h_{ij}|^2}{D_{ij}^\alpha} \quad (27)$$

where C_{ij} is related to the antenna gain and shading, h_{ij} is a factor of multipath decay, D_{ij} is the distance between two UAVs, and α is the propagation loss factor. To ensure successful information transmission with acceptable packet loss, the SNR should be higher than the

minimum level of link quality γ . Thus, if h_{ij} conforms to a standard Gaussian distribution, the probability of successful transmission can be expressed as

$$P_r^{ij}(\Gamma \geq \gamma) = \exp\left(-\frac{\sigma_j^2 \gamma D_{ij}^\alpha}{C_{ij} P_i}\right) \quad (28)$$

If P_r^{ij} is larger than a certain threshold δ_P , the two UAVs are connected. The above constraint should always be satisfied in order to form a communication network with strong connectivity. Therefore, if the communication constraint is not satisfied, one UAV should fly straight to the other one, i.e., the original velocity of the UAV in (20) should be redefined:

$$v = \begin{cases} v_{\text{cru}} \cdot \frac{(d_D/d_0 \cdot \delta + \tau)}{|d_D/d_0 \cdot \delta + \tau|}, & P_r^{ij} > \delta_P \\ v_{\text{cru}} \cdot \frac{(d_D/d_0 \cdot \delta + \tau + \lambda(\delta_P - P_r^{ij}) \cdot \eta)}{|d_D/d_0 \cdot \delta + \tau + \lambda(\delta_P - P_r^{ij}) \cdot \eta|}, & P_r^{ij} \leq \delta_P \end{cases} \quad (29)$$

where η denotes the normal vector of the UAV pointing towards the other one, and λ is a constant. It should be noted that even if only a single UAV is involved for a target search, we still need to consider the constraint of the communication network between the UAV and base station.

4. Experiment

The effectiveness of the proposed method is verified via simulation in the river map of Figure 1 by MATLAB R2022a on a computer with 64-b Intel Core i7-11800H CPU with 2.3 GHz frequency. Unless otherwise specified, the parameter values are as follows: cell length $L_s = 10$ m, minimum turning radius of UAV $R_{\min} = 150$ m, cruising speed of UAV $v_{\text{cru}} = 80$ m/s, detecting speed of UAV $v_{\text{cov}} = 20$ m/s, maximum speed of target $v_{\text{tmax}} = 3$ m/s, $a = 3$, $b = 2$, maximum number of expanded samples $E_{\max} = 400$, number of training samples in DBSCAN $N = 5000$, neighborhood radius in DBSCAN $R_\eta = 20$ m, and minimum number of neighbor points in DBSCAN $\text{minPts} = 10$, and the parameters of SOM and MGVF can be chosen in a relatively broad range. Additionally, a simple real-world flight experiment is conducted on a quadrotor UAV to verify the feasibility of our method.

4.1. Coverage Search by DBSCAN-RSOM

Figure 6 gives the original target probability map of a dynamic target in the Weihe river generated by the Parzen windows method based on some prior qualitative information about the target. Obviously, there are several high-value clusters conforming to Gaussian distribution in this map. On this basis, six high-value segments can be extracted by DBSCAN accurately, as shown in Figure 7. Additionally, the expanded downstream boundary of each high-value segment at $t = 1500$ s is shown in Figure 8, considering the beta distribution-based speed of the target drifting along the river (from left to right). It is obvious that the dynamically changing scenario cannot be ignored, since the length of each expansion (green) will be long. We suppose that the UAV starts from 3000 m, 0 m with a heading angle 90° and performs the coverage search task. Figure 9 illustrates the planned path by RSOM and MGVF, where the green areas denote the obstacles, the blue circles denote the original upstream/downstream boundary points of segments, and the yellow circles mean the expanded downstream boundary points. The RSOM method can determine the optimal coverage sequence (3,2,1,4,5,6) and the corresponding entry point (1,1,0,1,0,0, where 0 and 1 denote upstream/downstream boundary point, respectively) of these dynamically changing segments. Additionally, the transferring path by MGVF can avoid both regular and irregular obstacles.

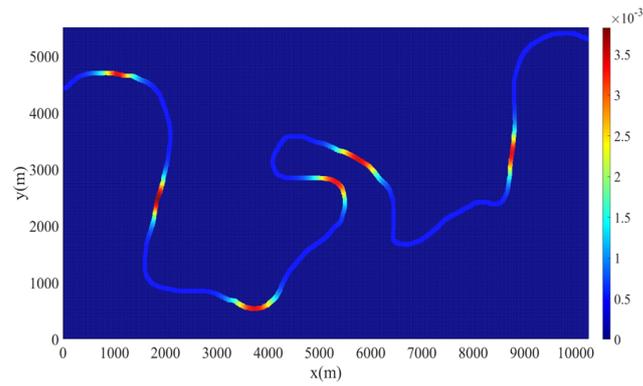


Figure 6. Original target probability map by Parzen windows.

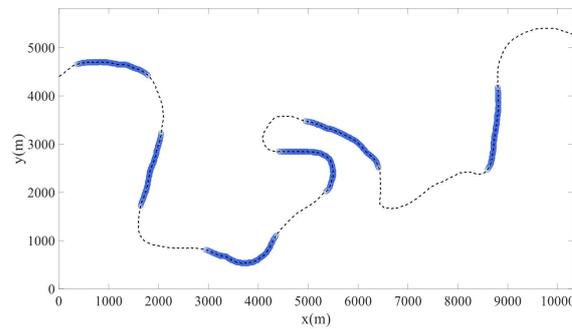


Figure 7. Six high-value segments extracted by DBSCAN.

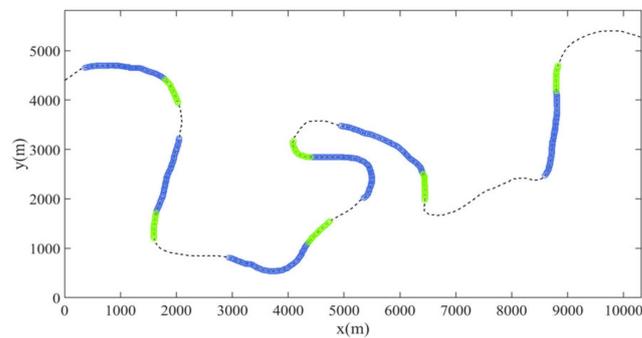


Figure 8. Expanded downstream boundary of segments at $t = 1500$ s.

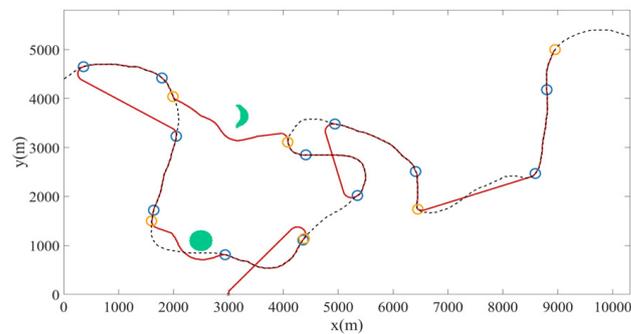


Figure 9. Planned path of UAV by RSOM and MGVF in obstacle environment.

4.2. Comparison of Different Methods

To verify the advantages of the DBSCAN-RSOM, some other methods are utilized as well for comparison. It should be noticed that the obstacle avoidance is not considered when using these methods. The first is the full sweeping method, i.e., the UAV flies along

the river and covers the whole river after a long time, as shown in Figure 10a. The second is the GMM-RSOM, i.e., the architecture is similar to our method but the high-value curve segments are extracted by the GMM. However, the clustering result by the GMM is heavily dependent on the given number of clusters. Suppose that the preset number of clusters is 4, for example, it will result in some low-probability areas being included in the extracted segments. As shown in Figure 10b, the UAV will waste some time on detecting these low-value areas. The third is the DBSCAN-DW, where the dynamic window (DW) strategy is utilized to sort the segments extracted by the DBSCAN. The principle of the DW is similar to the well-known model predictive control method, and the optimal sequence for the next three segments is computed but only the first one is covered by the UAV in a rolling way. For the DBSCAN-SOM, the coverage sequence of segments is only computed once by the traditional SOM method, where the dynamic characteristic of the target is ignored. The planned paths using different methods can be seen in Figure 10. In addition, Figure 11 shows the curve of cumulative detection reward of each method. The RSOM is a global optimization strategy and, inversely, the DW emphasizes a local myopia, so the reward of the DBSCAN-RSOM is less than the DBSCAN-DW in the early stage but the DBSCAN-RSOM will complete the search task in the shortest time. The performance of each method is also provided in Table 1. Obviously, the hierarchical heuristic architecture (i.e., the latter four methods) has a much higher efficiency (less flight time) than the full sweeping method, which can improve the efficiency by approximately 31.5%. And the DBSCAN-RSOM can obtain almost all the detection rewards (0.99) in the shortest time (620 s). Furthermore, although the computation time of our method is longer than others, it is still acceptable for the time-sensitive mission in this paper.

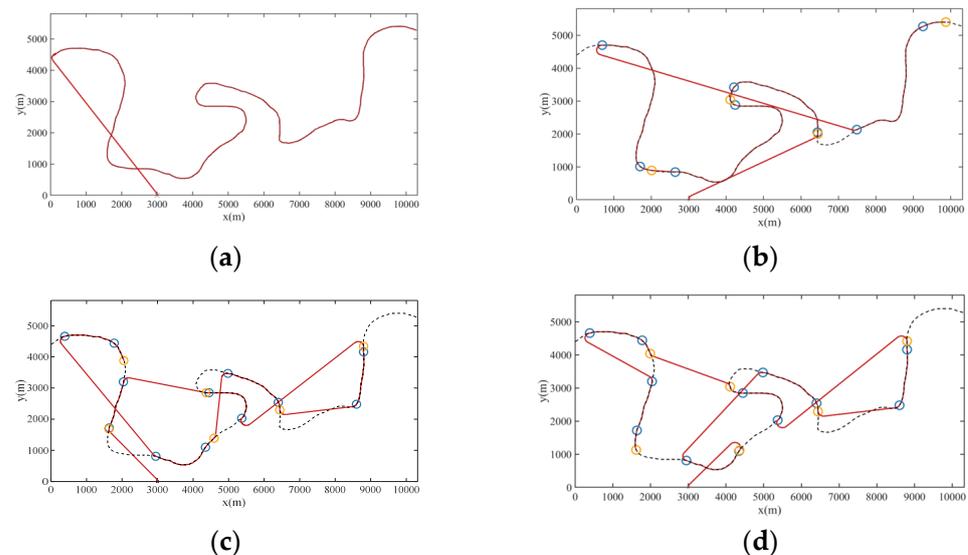


Figure 10. Planned paths using different methods in free environment. (a) Full sweeping. (b) GMM-RSOM. (c) DBSCAN-DW. (d) DBSCAN-SOM.

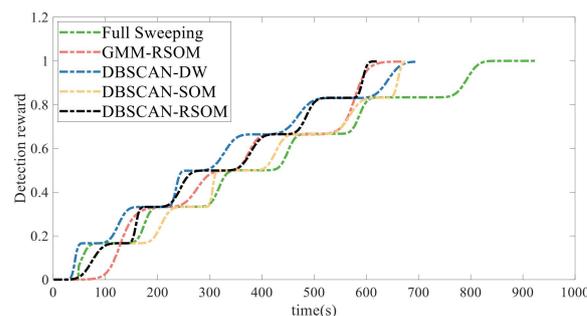
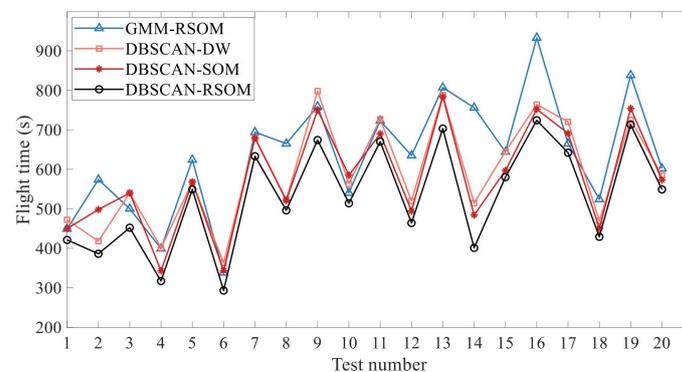


Figure 11. Comparison of cumulative detection reward using different methods.

Table 1. Performance of different methods.

Method	Final Reward	Flight Time (s)	Computation Time (s)
Full sweeping	1	923	0.91
GMM-RSOM	0.99	671	2.33
DBSCAN-DW	0.99	694	2.28
DBSCAN-SOM	0.99	674	3.38
DBSCAN-RSOM	0.99	620	4.03

We also ran 20 tests in various random scenarios to verify the robustness of our method. In each test, 10~30 discrete qualitative data with a random position, reliability, and accuracy were set. Because the final reward of each method was almost the same, we only took the flight time as the comparison index. As the flight time by the full sweeping method was much larger than others, Figure 12 mainly displays the statistical results of four other hierarchical methods. Overall, the flight time by our method is the shortest in most cases, showing its high search efficiency in various scenarios.

**Figure 12.** Statistical results by different methods.

4.3. Real-World Flight Experiment

To verify the feasibility of the DBSCAN-RSOM, we utilize a quadrotor UAV to perform a simple real-world flight experiment on a $80\text{ m} \times 60\text{ m}$ field, where a virtual target is assumed to exist in the curve-shaped road and a series of prior data points are set artificially. With a wheelbase of 450 mm, the quadcopter is equipped with a Raspberry Pi 4B on-board computer, monocular and binocular cameras, GPS and IMU, etc., as shown in Figure 13. Three high-value curve segments with true coordinates of boundary points are extracted by the DBSCAN. Assuming that the virtual target moves at 0.1 m/s, the cruising speed and the detecting speed of the quadrotor is 1 m/s and 0.5 m/s, respectively, and the path of quadrotor is then planned by the RSOM. Figure 14 shows the real flight path, and obviously the quadcopter completes the coverage of high-value segments. As seen in Table 2, our method can improve the efficiency by approximately 16.3%, compared with the full sweeping method.

**Figure 13.** The quadcopter UAV.



Figure 14. The real flight path of UAV.

Table 2. Performance of different methods in real-world flight experiment.

Method	Final Reward	Flight Time (s)
Full Sweeping	1	92
DBSCAN-DW	0.99	85
DBSCAN-RSOM	0.99	77

5. Conclusions

The dynamic target search mission in a curve-shaped area is formulated as an optimal UAV path planning problem, which aims to obtain the maximum detection reward in the shortest flight time under various constraints. To improve the mission efficiency, this paper proposes a three-layer hierarchical heuristic architecture, including the extraction of high-value curve segments by the DBSCAN, the sorting of high-value curve segments by the RSOM, and path planning by Dubins with MGVF. The effectiveness, robustness, and feasibility of our method is verified through simulations and a real flight experiment. Compared with the full sweeping method, our method can improve the efficiency by approximately 31.5% in simulations and 16.3% in real experiments, respectively. The proposed hierarchical heuristic architecture is not limited to curve-shaped areas, and it can be generalized to 2D or even 3D wide cases (e.g., sea surface, farmland, hilly region, etc.).

Author Contributions: Conceptualization, L.L. and P.Y.; methodology, P.Y. and C.Z.; software, C.Z.; validation, D.W.; resources, P.Y.; writing—original draft preparation, C.Z.; writing—review and editing, D.W.; supervision, D.F.; project administration, L.L.; funding acquisition, J.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This work was partially supported by the Open Research Fund Program of Key Laboratory of Industrial Internet and Big Data, China National Light Industry, Beijing Technology and Business University (Grant No. IIBD-2022-KF04) and the Natural Science Foundation of Shandong Province, China (Grant No. ZR2023ME009).

Data Availability Statement: Data are contained within the article.

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

References

1. Wilhelm, J.P.; Clem, G. Vector field UAV guidance for path following and obstacle avoidance with minimal deviation. *J. Guid. Control Dyn.* **2019**, *42*, 1848–1856. [\[CrossRef\]](#)
2. Akter, R.; Doan, V.S.; Huynh-The, T.; Kim, D.-S. Rfdoa-net: An efficient convnet for rf-based doa estimation in uav surveillance systems. *IEEE Trans. Veh. Technol.* **2021**, *70*, 12209–12214. [\[CrossRef\]](#)
3. Ye, J.; Fu, C.; Lin, F.; Ding, F.; An, S.; Lu, G. Multi-regularized correlation filter for UAV tracking and self-localization. *IEEE Trans. Ind. Electron.* **2021**, *69*, 6004–6014. [\[CrossRef\]](#)
4. Booth, K.E.C.; Piacentini, C.; Bernardini, S.; Beck, J.C. Target search on road networks with range-constrained UAVs and ground-based mobile recharging vehicles. *IEEE Robot. Autom. Lett.* **2020**, *5*, 6702–6709. [\[CrossRef\]](#)
5. Lun, Y.; Wang, H.; Wu, J.; Liu, Y.; Wang, Y. Target Search in Dynamic Environments with Multiple Solar-Powered UAVs. *IEEE Trans. Veh. Technol.* **2022**, *71*, 9309–9321. [\[CrossRef\]](#)
6. Yao, P.; Wang, H. Dynamic Adaptive Ant Lion Optimizer applied to route planning for unmanned aerial vehicle. *Soft Comput.* **2017**, *21*, 5475–5488. [\[CrossRef\]](#)

7. Sarkar, M.; Yan, X.; Erol, B.A.; Raptis, I.; Homaifar, A. A novel search and survey technique for unmanned aerial systems in detecting and estimating the area for wildfires. *Robot. Auton. Syst.* **2021**, *145*, 103848. [[CrossRef](#)]
8. Ivić, S.; Crnković, B.; Arbabi, H.; Loire, S.; Clary, P.; Mezić, I. Search strategy in a complex and dynamic environment: The MH370 case. *Sci. Rep.* **2020**, *10*, 19640. [[CrossRef](#)] [[PubMed](#)]
9. Brown, D.; Sun, L. Dynamic exhaustive mobile target search using unmanned aerial vehicles. *IEEE Trans. Aerosp. Electron. Syst.* **2019**, *55*, 3413–3423. [[CrossRef](#)]
10. Wang, T.; Qin, R.; Chen, Y.; Snoussi, H.; Choi, C. A reinforcement learning approach for UAV target searching and tracking. *Multimed. Tools Appl.* **2019**, *78*, 4347–4364. [[CrossRef](#)]
11. Boulares, M.; Barnawi, A. A novel UAV path planning algorithm to search for floating objects on the ocean surface based on object's trajectory prediction by regression. *Robot. Auton. Syst.* **2021**, *135*, 103673. [[CrossRef](#)]
12. Quamar, M.; ElFerik, S.; Saif, A.-W.A. A Bioinspired Adaptation-Based Algorithm for Swarm of UAVs in multiple threat prone Environments. In Proceedings of the 2023 20th International Multi-Conference on Systems, Signals & Devices (SSD), Mahdia, Tunisia, 20–23 February 2023.
13. Zhu, K.; Han, B.; Zhang, T. Multi-UAV distributed collaborative coverage for target search using heuristic strategy. *Guid. Navig. Control* **2021**, *1*, 2150002. [[CrossRef](#)]
14. Yao, P.; Zhao, Z. Improved Glasius bio-inspired neural network for target search by multi-agents. *Inf. Sci.* **2021**, *568*, 40–53. [[CrossRef](#)]
15. Cho, S.W.; Park, H.J.; Lee, H.; Shim, D.H.; Kim, S.-Y. Coverage path planning for multiple unmanned aerial vehicles in maritime search and rescue operations. *Comput. Ind. Eng.* **2021**, *161*, 107612. [[CrossRef](#)]
16. Oh, H.; Kim, S.; Tsourdos, A.; White, B.A. Coordinated road-network search route planning by a team of UAVs. *Int. J. Syst. Sci.* **2014**, *45*, 825–840. [[CrossRef](#)]
17. Abubakar, A.; Farah, M.; Alsheikh, M.; Saif, A.-W.A. Optimal Altitude Range for Image Collection UAV from Agricultural Areas with Image Resolution and Power Consumption Constraints. In Proceedings of the 2023 20th International Multi-Conference on Systems, Signals & Devices (SSD), Mahdia, Tunisia, 20–23 February 2023; pp. 689–693. [[CrossRef](#)]
18. Li, Y.; Zhang, Z.; Sun, Q.; Huang, Y. A distributed framework for multiple UAV cooperative target search under dynamic environment. *J. Frankl. Inst.* **2024**, *361*, 106810. [[CrossRef](#)]
19. Huang, H.; Savkin, A.V.; Huang, C. Decentralized autonomous navigation of a UAV network for road traffic monitoring. *IEEE Trans. Aerosp. Electron. Syst.* **2021**, *57*, 2558–2564. [[CrossRef](#)]
20. Yao, P.; Xie, Z.; Ren, P. Optimal UAV route planning for coverage search of stationary target in river. *IEEE Trans. Control Syst. Technol.* **2017**, *27*, 822–829. [[CrossRef](#)]
21. Li, J.; Li, W.; Zhang, W. Path Planning of UAV Navigation Mark Inspection Using a K-means Clustering ACA. *J. Mar. Sci. Technol.* **2023**, *31*, 10. [[CrossRef](#)]
22. Qiu, Z.; Ma, Y.; Fan, F.; Huang, J.; Wu, L.; Du, Y. Improved DBSCAN for Infrared Cluster Small Target Detection. *IEEE Geosci. Remote Sens. Lett.* **2023**, *20*, 3329372. [[CrossRef](#)]
23. Zhao, L.; Bi, X.; Dong, Z.; Xiao, N.; Zhao, A. Robust traveling salesman problem with UAV: Balancing risk and makespan in contactless delivery. *Int. Trans. Oper. Res.* **2024**, *31*, 167–191. [[CrossRef](#)]
24. Yu, K.; Budhiraja, A.K.; Buebel, S.; Tokekar, P. Algorithms and experiments on routing of unmanned aerial vehicles with mobile recharging stations. *J. Field Robot.* **2019**, *36*, 602–616. [[CrossRef](#)]
25. Yao, P.; Lou, Y.; Zhang, K. Multi-USV cooperative path planning by window update based self-organizing map and spectral clustering. *Ocean. Eng.* **2023**, *275*, 114140. [[CrossRef](#)]
26. Saif, A.-W.A.; Aliyu, A.; Al Dhiaifallah, M.; Elshafei, M. Decentralized backstepping control of a quadrotor with tilted-rotor under wind gusts. *Int. J. Control Autom. Syst.* **2018**, *16*, 2458–2472. [[CrossRef](#)]
27. Elfeky, M.; Elshafei, M.; Saif, A.-W.A.; Al-Malki, M.F. Modeling and simulation of quadrotor UAV with tilting rotors. *Int. J. Control Autom. Syst.* **2016**, *14*, 1047–1055. [[CrossRef](#)]
28. Zhu, D.; Cao, X.; Sun, B.; Luo, C. Biologically inspired self-organizing map applied to task assignment and path planning of an AUV system. *IEEE Trans. Cogn. Dev. Syst.* **2018**, *10*, 304–313. [[CrossRef](#)]
29. Yao, P.; Wei, X. Multi-UAV Information Fusion and Cooperative Trajectory Optimization in Target Search. *IEEE Syst. J.* **2022**, *16*, 4325–4333. [[CrossRef](#)]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.