agronomy

MDPI

Article

Design and Implementation of Evaluation Method for Spraying Coverage Region of Plant Protection UAV

Kun Zhang [1], Long Zhao [2,*], Jingying Cui [1], Pengjun Mao [3], Bohan Yuan [1] and Yuyang Liu [1]

[1] College of Physics and Electronic Engineering, Xinyang Normal University, Xinyang 464000, China; zk@xynu.edu.cn (K.Z.)
[2] College of Agricultural Equipment Engineering, Henan University of Science and Technology, Luoyang 471000, China
[3] School of Mechatronics Engineering, Henan University of Science and Technology, Luoyang 471000, China
* Correspondence: hkdzhaolong@haust.edu.cn; Tel.: +86-18603793588

Abstract: Plant protection UAVs are becoming the preferred plant protection method for agricultural pest control. At present, the evaluation of droplet distribution in aerial spraying is collected and evaluated after the completion of prevention and control operations, and there is a lack of real-time evaluation methods. Based on the flight parameter during the UAV plant protection process, real-time estimation of droplet distribution is the key to solving this problem and further improving the effectiveness of aerial spraying. This study proposes a merging algorithm for arbitrary polygonal regions, meshing the boundaries of the region, divide the mesh segments based on the overlapping meshes between the two regions, and connect the valid mesh connection segments of the two regions according to certain rules to obtain the intersection, union, and residual operation results between the regions. Afterwards, software based on this algorithm was developed and applied to generate spraying coverage regions, leakage spray regions, and repeated spray regions. The experimental results on theoretical and irregular routes show that the algorithm can accurately generate droplet distribution regions. The error of the calculation results with a mesh scale of 0.05 m is within 7‰, and the operating speed is above 30 Hz, meeting the real-time requirements. The smaller the mesh scale is, the higher the accuracy of the calculation results is, but the slower the calculation speed. Therefore, in practical applications, it is necessary to choose an appropriate mesh scale based on hardware computing power and accuracy level requirements. This study solves the problem of cumulative calculation of droplet distribution during the operation of plant protection UAVs, providing a basis for objectively evaluating the operation quality of plant protection UAVs and optimizing the setting of operation parameters.

Keywords: droplet deposition; plant protection UAV; pesticide spraying; region merging; spraying coverage

check for updates

Citation: Zhang, K.; Zhao, L.; Cui, J.; Mao, P.; Yuan, B.; Liu, Y. Design and Implementation of Evaluation Method for Spraying Coverage Region of Plant Protection UAV. *Agronomy* **2023**, *13*, 1631. https://doi.org/10.3390/agronomy13061631

Academic Editor: Yanbo Huang

Received: 24 May 2023
Revised: 16 June 2023
Accepted: 17 June 2023
Published: 18 June 2023

1. Introduction

Diseases and pests are the major threat to agricultural production. According to the Food and Agriculture Organization of the United Nations, the loss of world food production caused by diseases and pests is approximately 24% [1,2]. Diseases and pests spread rapidly and have a wide range of spread, which can lead to crop yield reduction or even crop failure, and quality degradation [3]. However, manual control efficiency is low, and it is easy to threaten the health of operators. Ground mechanical control cannot be applied to special terrains [4]. UAVs are becoming the preferred method of plant protection due to their high operational efficiency, good prevention and control effects, good maneuverability, wide applicability, water and drug conservation, and low risk to operators [5–8]. However, the harm of leakage spraying is enormous, and the pests at the leakage spraying region may spread to other regions again, leading to a resurgence of pests. The phenomenon of repeated spraying can reduce the efficiency of pesticide spraying and pollute the soil [9,10].

In recent years, scholars have conducted extensive research on the spatial distribution of droplets in plant protection UAVs, aiming to clarify the drift patterns of droplets in different situations, and then take effective measures to reduce droplet drift and reduce leakage, and repeated spraying. The droplet distribution of aerial spraying is closely related to the UAV model for plant protection [11], flight load [12], spray parameters [13,14], droplet size [15,16], UAV rotor wind field [17,18], external wind field [19,20], flight operational parameters [21–23], and other factors. Based on these studies [24–26], it is found that different UAV models, mission loads, and spray parameters will cause a large change in the droplet distribution, while the speed and direction of the external wind field is the most important factor that causes droplet drift, and the flight operational parameters of plant protection UAV are the secondary factor that affects drift. Through research [27–29], a preliminary exploration was conducted to predict droplet drift during aerial spraying based on wind speed and direction parameters and to dynamically adjust the motion trajectory of plant protection UAV, which can effectively reduce leakage spraying and repeated spraying, and improve target deposition performance. However, these studies did not provide real-time evaluation of droplet deposition, and the specific droplet deposition effect still needs to be determined through subsequent measurements.

Predictably, applying the drift law of droplets to generate and analyze the spatial distribution of droplets throughout aerial spraying can effectively evaluate and improve the deposition performance of the target. Decomposing the region covered by droplets during the aerial spraying process into multiple time interval subregions, the spatial distribution map of droplets can essentially be simplified as a series of region merging operations between these subregions and with the region to be worked on; that is, performing a series of Boolean operations on these polygonal regions, including intersection, union, and difference. In recent years, the polygon Boolean operation methods published by researchers include algorithms that only support convex polygon operations [30,31] and algorithms that support both convex and concave polygon operations [32–35], to adapt to operations with different data structures, vector data structure algorithms [36–38], and raster data structure algorithms [39–41] are currently the main methods for solving polygon Boolean operations. These methods can solve polygon Boolean operation problems in various situations, but the dynamic construction of the region covered by the droplets includes the Boolean operations of polygons with holes, convex polygon, concave polygon, and two polygons with overlapping polygons, which lacks a general solution. In addition, the situation of polygon Boolean operations merging changes dynamically with the aerial spraying process of the plant protection UAV. The existing algorithms cannot effectively solve the problem of polygon region dynamic merging construction in all the above situations.

Therefore, this study proposes a new algorithm for polygon Boolean operation based on boundary meshing, which converts the operation between points and line segments into the operation of meshes and mesh connected segments, and solves the Boolean operation results of arbitrary polygons. Based on on-site measured flight data, real-time calculation of droplet deposition distribution is carried out to construct spraying coverage regions, leakage spraying regions, and repeated spraying regions, providing a reference basis for precise spraying, variable spraying, and improving operational effectiveness.

2. Materials and Methods

2.1. Instruments and Equipment

In the field, with coordinates (23°09′76″ N, 113°20′37″ E), first, manually handheld differential GPS positioning equipment is used to collect the boundary coordinates of the site as the boundary of the region to be sprayed. As shown in Figure 1, the self-developed data acquisition module is installed on an eight-rotor plant protection UAV. The flight controller visually controls the UAV to simulate pesticide spraying operations in the field, including forward, backward, left, right, turn, and short hover. Finally, the longitude and latitude coordinates of an irregular route are collected by self-developed PC software, which includes various flight conditions during the actual operation of the plant protection UAV.

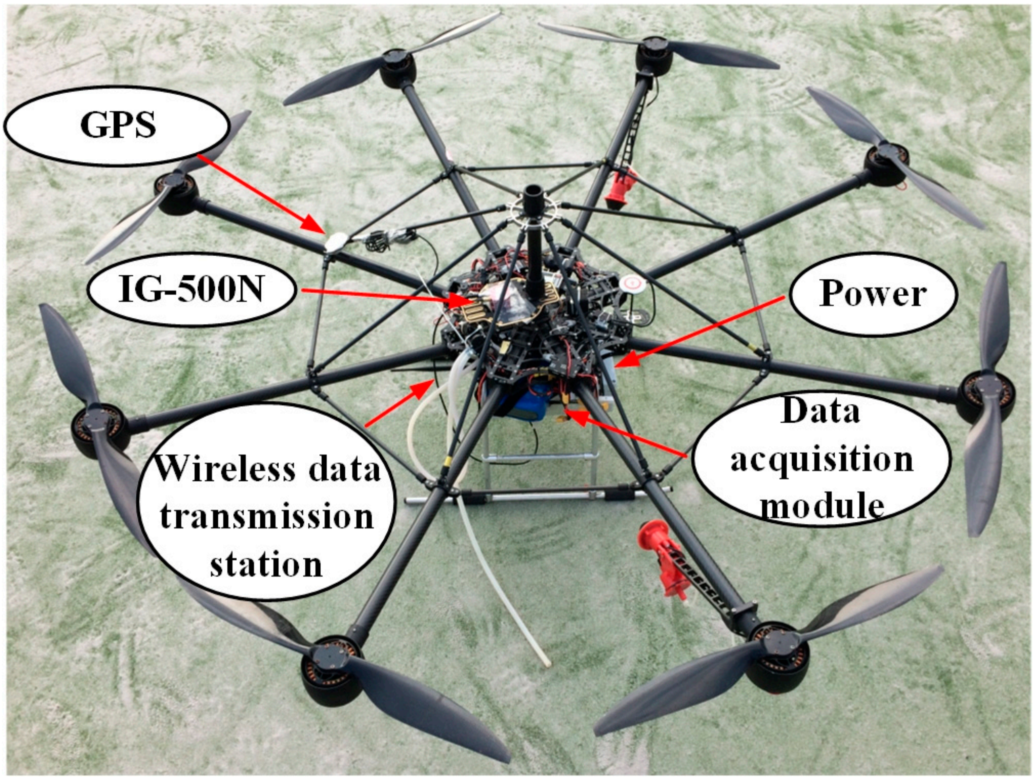

Figure 1. Experimental Platform.

The longitude and latitude coordinates are spherical coordinates and need to be converted into geodetic plane coordinates x and y through Gauss Kruger projection transformation.

$$x = X + \frac{{L_c}^2}{2} \cdot N \cdot sinB \cdot cosB + \frac{{L_c}^4}{24} \cdot N \cdot sinB \cdot cos^3B \cdot (5 - t^2 + 9\delta^2 + 4\delta^4) + \frac{{L_c}^6}{720} \cdot N \cdot sinB \cdot cos^5B \cdot (61 - 58t^2 + t^4) \quad (1)$$

$$y = N \cdot cosB + \frac{{L_c}^3}{6} \cdot N \cdot cos^3B \cdot (1 - t^2 + \delta^2) + \frac{{L_c}^5}{120} \cdot N \cdot cos^5B \cdot (5 - 18t^2 + t^4 + 14\delta^4 - 58\delta^2 \cdot t^2) + 500000 \quad (2)$$

L is the current longitude, L_0 is the longitude of the central meridian of the projection zone, and $L_c = L - L_0$. X is the meridian arc length from the equator to the parallel circle at latitude B, whose value is $X = c\int_0^B (1 + e \cdot c^2cos^2B^{-3/2})d$. E is the first eccentricity of the ellipsoid, whose value is $e = 2\partial - \partial^2$, and ∂ is the oblateness, whose value is $\partial = \sqrt{a^2/b^2 - 1}$. C is the radius of polar curvature, m; δ and t are auxiliary variables with values of $t = tanB$ and $\delta = e \cdot c \cdot cosB$. a and b are the long and short radii of the reference ellipsoid, respectively.

Based on the position coordinates, heading angle, and spray amplitude of the UAV, the operation coverage subregion is constructed. The construction rules are shown in Figure 2. The position coordinates of the drone are $P_n(x_n, y_n)$, and the rotation angle when the coordinate axis X rotates in the same direction as the heading angle is α, $P_n'P_n''$ is perpendicular to the heading of UAV, the coordinates of P_n' and P_n'' can be obtained through Equations (3) and (4).

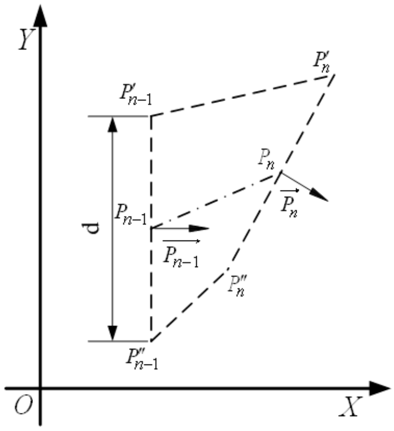

Figure 2. Construction of operation coverage subregion. Note: Points P_{n-1} and P_n represent the coordinate position of UAV, *d* represents the spray amplitude of the UAV, $\overrightarrow{P_n}$ and $\overrightarrow{P_{n-1}}$ represent the heading of UAV, P'_n, P'_{n-1}, P''_n and P''_{n-1} are the boundary vertices of the operation coverage subregion.

When $0 \leq \alpha < \pi/2$ or $3\pi/2 < \alpha \leq 2\pi$:

$$\begin{bmatrix} x'_n & y'_n \\ x''_n & y''_n \end{bmatrix} = \begin{bmatrix} x_n - \frac{d}{2} \cdot sin\alpha & y_n + \frac{d}{2} \cdot cos\alpha \\ x_n + \frac{d}{2} \cdot sin\alpha & y_n - \frac{d}{2} \cdot cos\alpha \end{bmatrix} \quad (3)$$

When $\pi/2 \leq \alpha < 3\pi/2$:

$$\begin{bmatrix} x'_n & y'_n \\ x''_n & y''_n \end{bmatrix} = \begin{bmatrix} x_n + \frac{d}{2} \cdot sin\alpha & y_n - \frac{d}{2} \cdot cos\alpha \\ x_n - \frac{d}{2} \cdot sin\alpha & y_n + \frac{d}{2} \cdot cos\alpha \end{bmatrix} \quad (4)$$

2.2. Basic Principles of the Algorithm

The basic principle of the region merging algorithm based on boundary meshing is shown in Figure 3, using the union of region A and region B as an example for analysis. Step 1: as shown in Figure 3A, the two-dimensional drawing space is decomposed into countless meshes based on the original coordinate axis. The meshes penetrated by the region boundary are solved, and these meshes are connected in the order of the original region vertices to form a grid sequence. The mesh sequence is used to replace the original region boundary, which is the meshing of the region boundary.

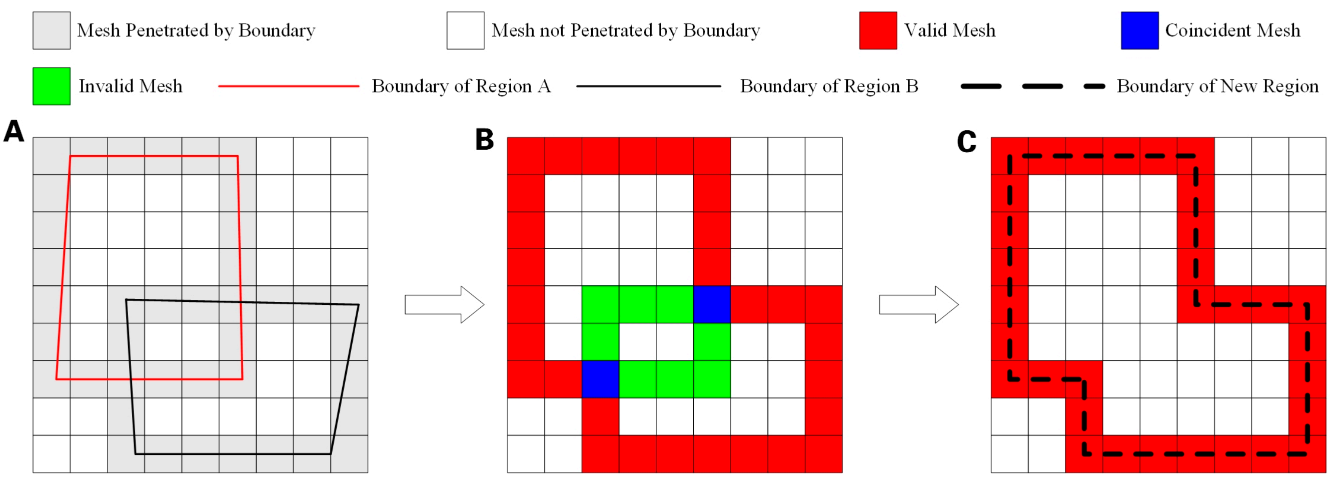

Figure 3. Schematic diagram of region merging algorithm based on boundary meshing. (**A**) Boundary meshing; (**B**) Mesh sequence classification; (**C**) New mesh sequence representing the result of region merging.

Step 2: Classify the mesh sequence representing the boundary of the region, as shown in Figure 3B. The meshes are preliminarily divided into coincident meshes, overlapping meshes, valid meshes and invalid meshes. The valid mesh refers to the mesh representing the boundary of the new region in the region merging result. Therefore, the mesh of region A outside region B is the valid mesh, and the mesh of region B outside region A is also the valid mesh. Invalid mesh refers to mesh that do not belong to the boundaries of the new region, that is, the mesh of region A inside region B and the mesh of region B inside region A. Coincident mesh refers to the mesh where two regions coincide with each other. Using coincident meshes, the mesh sequence of each region is divided into several valid and invalid mesh segments. Then, extract the connection points of the valid mesh segments of two regions in the coincident meshes, which are called overlapping meshes.

Step 3: Connect the valid mesh segments of the two regions at the overlapping meshes according to certain rules to form a mesh sequence representing the new region, as shown in Figure 3C. Finally, extract the center point coordinates of the meshes as the region vertex coordinates to construct the new region boundary, which is the result of region merging.

2.3. Meshing of Region Boundary

The meshing of region boundaries is to rephrase them in the form of mesh coordinates. In order to reduce the number of operations, all vertices of the original region are first meshed, and then one of the meshes is used as the starting mesh. According to the order of vertex connections in the original region, grow the mesh to the next vertex until it returns to the starting mesh.

2.3.1. Meshing of Regional Vertices

The mesh coordinates (x, y) represent the internal region contained by the boundaries composed of vertices (x, y), $(x, y + d_g)$, $(x + d_g, y + d_g)$, and $(x + d_g, y)$. Assuming that $A_{n-1}(x_{n-1}, y_{n-1})$ and $A_n(x_n, y_n)$ are the vertices of the original region, and d_g is the scale value of the mesh division, the mesh coordinates (x'_{n-1}, y'_{n-1}) and (x'_n, y'_n) of the two vertices can be preliminarily calculated by using Equation (5).

$$\begin{bmatrix} x'_{n-1} & y'_{n-1} \\ x'_n & y'_n \end{bmatrix} = \begin{bmatrix} \left\lfloor \frac{x_{n-1}}{d_g} \right\rfloor & \left\lfloor \frac{y_{n-1}}{d_g} \right\rfloor \\ \left\lfloor \frac{x_{n-1}}{d_g} \right\rfloor & \left\lfloor \frac{y_{n-1}}{d_g} \right\rfloor \end{bmatrix} \cdot d_g \tag{5}$$

Then the number of meshes n_x and n_y penetrated by line segment $A_{n-1}A_n$ in the X and Y directions can be determined.

$$\begin{bmatrix} n_x \; n_y \end{bmatrix} = \left[\left\lceil \frac{\left| x_n - x'_{n-1} \right|}{d_g} \right\rceil \left\lceil \frac{\left| y_n - y'_{n-1} \right|}{d_g} \right\rceil \right] \tag{6}$$

when using (x'_{n-1}, y'_{n-1}) as the starting mesh, the main growth direction of the mesh is the coordinate axis of the side with higher n_x and n_y values, and the auxiliary growth direction of the grid is the coordinate axis of the other side. The mesh grows along the boundary line $A_{n-1}A_n$ towards mesh (x'_n, y'_n). All meshes penetrated by the region boundary $A_{n-1}A_n$ along the way are sequentially added to the mesh sequence. Growth direction coefficient α determines the direction of mesh growth along the coordinate axis. When $\alpha = 1$, mesh growth is along the positive direction of the coordinate axis; when $\alpha = -1$, the mesh grows along the negative direction of the coordinate axis. α_x and α_y represents the growth direction coefficients in the X and Y directions, respectively.

$$\alpha_x = \begin{cases} \frac{x_n - x_{n-1}}{|x_n - x_{n-1}|} & (x_n - x_{n-1} \neq 0) \\ 1 & (x_n - x_{n-1} = 0) \end{cases} \tag{7}$$

$$\alpha_y = \begin{cases} \frac{y_n - y_{n-1}}{|y_n - y_{n-1}|} & (y_n - y_{n-1} \neq 0) \\ 1 & (y_n - y_{n-1} = 0) \end{cases} \quad (8)$$

2.3.2. Rules of Mesh Growth

The process of mesh growth is essentially searching for the mesh connected to the current mesh, and the connectivity rules of the mesh are its growth rules. This algorithm is based on a four mesh space composed of a starting mesh and three adjacent meshes. Firstly, the mesh connectivity rules within the four mesh space are determined, and then the last connected mesh is used as the starting mesh to search for the next four mesh space. Finally, all meshes penetrated by the boundary line are determined step by step. Assuming the starting mesh of the mesh growth is (x'_1, y'_1), the other three meshes of the four mesh space are $(x'_1, y'_1 + \alpha_y \cdot d_g)$, $(x'_1 + \alpha_x \cdot d_g, y'_1 + \alpha_y \cdot d_g)$, and $(x'_1 + \alpha_x \cdot d_g, y'_1)$. As shown in Figure 4, the number of meshes that the boundary line penetrates within the four mesh space is 2 or 3, i.e., the meshes are two connected or three connected.

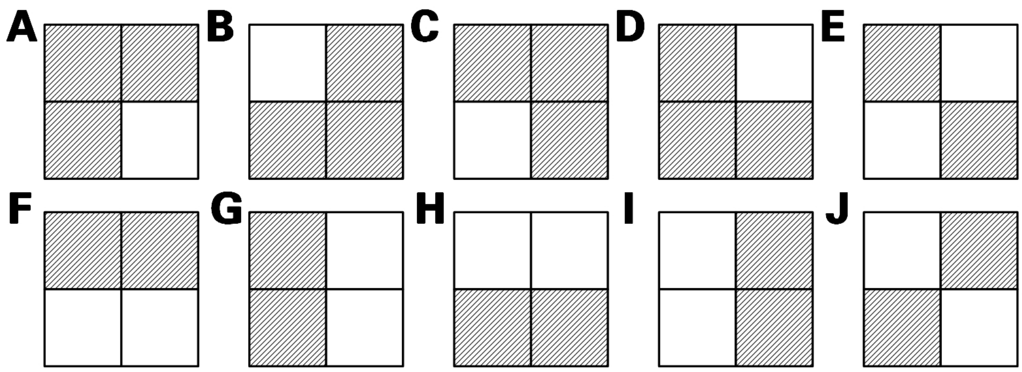

Figure 4. Mesh connectivity in four mesh space. (**A**–**D**) Triple connected meshes; (**E**–**J**) Two connected meshes.

By calculating and analyzing the coordinate values in Figure 5, the rules of mesh growth can be obtained. When $n_x \geq n_y$, a linear search should be performed along the direction of α_x. The first search should use a straight line of $x = x'_1 + \frac{1+\alpha_x}{2} \cdot d_g$. The intersection point between the straight line and the mesh boundary line is $(x'_1 + \frac{1+\alpha_x}{2} \cdot d_g, y_2)$. When $y_2 = y'_1 + \frac{1+\alpha_y}{2} \cdot d_g$, the meshes penetrated by $A_{n-1}A_n$ are two connected, and the connected mesh is (x'_2, y'_2). When $y_2 \neq y'_1 + \frac{1+\alpha_y}{2} \cdot d_g$, the number of meshes penetrated along the Y-axis direction should be calculate with $n'_{y1} = \left\lceil \frac{\left| y_2 - y'_1 - \frac{1-\alpha_y}{2} \cdot d_g \right|}{d_g} \right\rceil - 1$; when $n'_{y1} \leq 0$, continue searching along the direction of α_x to find the intersection point between $A_{n-1}A_n$ and the straight line $x = x'_1 + \frac{1+3\alpha_x}{2} \cdot d_g$, which is $(x'_1 + \frac{1+3\alpha_x}{2} \cdot d_g, y_3)$. Then, the number of penetrating meshes along the Y-axis direction can be calculated with $n'_{y2} = \left\lceil \frac{\left| y_3 - y'_1 - \frac{1-\alpha_y}{2} \cdot d_g \right|}{d_g} \right\rceil - 1$. When $n'_{y2} \leq 0$, the meshes penetrated by $A_{n-1}A_n$ are two connected, and the connected mesh is (x'_2, y'_1); When $n'_{y2} \geq 1$, the meshes penetrated by $A_{n-1}A_n$ are three connected, and the connected meshes are (x'_2, y'_1) and (x'_2, y'_2). When $n'_{y1} > 0$, the meshes penetrated by $A_{n-1}A_n$ are three connected, and the connected meshes are (x'_1, y'_2) and (x'_2, y'_2).

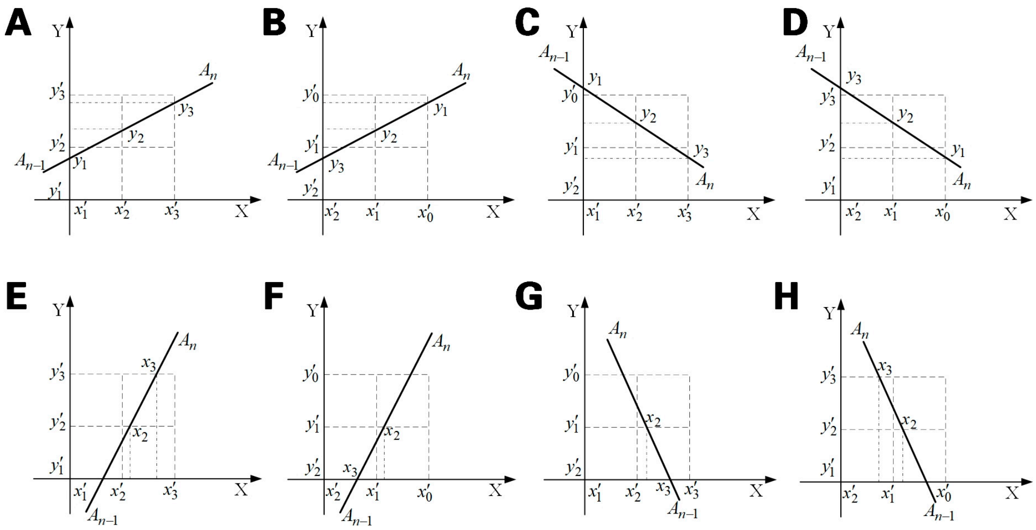

Figure 5. Mesh solving models correspond to different connectivity modes. As shown in (**A**–**D**), the mesh solving model is obtained when the X-axis direction is the main direction of mesh growth. As shown in (**E**–**H**), the mesh solving model is obtained when the Y-axis direction is the main direction of mesh growth. x'_0, x'_1, x'_2, x'_3, y'_0, y'_1, y'_2, and y'_3 are the horizontal and vertical coordinate values of the mesh boundary lines, where $x'_3 - x'_2 = x'_2 - x'_1 = x'_1 - x'_0 = \alpha_x \cdot d_g$ and $y'_3 - y'_2 = y'_2 - y'_1 = y'_1 - y'_0 = \alpha_y \cdot d_g$.

When $n_x < n_y$, a linear search should be performed along the direction of α_y. The first search should use a straight line of $y = y'_1 + \frac{1+\alpha_y}{2} \cdot d_g$. The intersection point between the straight line and the mesh boundary line is $(x_2, y'_1 + \frac{1+\alpha_y}{2} \cdot d_g)$. When $x_2 = x'_1 + \frac{1+\alpha_x}{2} \cdot d_g$, the meshes penetrated by $A_{n-1}A_n$ are two connected, and the connected mesh is (x'_2, y'_2). When $x_2 \neq x'_1 + \frac{1+\alpha_x}{2} \cdot d_g$, the number of meshes penetrated along the x-axis direction should be calculate with $n'_{x1} = \left\lceil \frac{\left| x_2 - x'_1 - \frac{1-\alpha_x}{2} \cdot d_g \right|}{d_g} \right\rceil - 1$; when $n'_{x1} \leq 0$, continue searching along the direction of α_y to find the intersection point between $A_{n-1}A_n$ and the straight line $y = y'_1 + \frac{1+3\alpha_y}{2} \cdot d_g$, which is $(x_3, y = y'_1 + \frac{1+3\alpha_y}{2} \cdot d_g)$. Then, the number of penetrating meshes along the x-axis direction can be calculated with $n'_{x2} = \left\lceil \frac{\left| x_3 - x'_1 - \frac{1-\alpha_x}{2} \cdot d_g \right|}{d_g} \right\rceil - 1$. When $n'_{x2} \leq 0$, the meshes penetrated by $A_{n-1}A_n$ are two connected, and the connected mesh is (x'_1, y'_2); When $n'_{x2} \geq 1$, the meshes penetrated by $A_{n-1}A_n$ are three connected, and the connected meshes are (x'_1, y'_2) and (x'_2, y'_2). When $n'_{x1} > 0$, the meshes penetrated by $A_{n-1}A_n$ are three connected, and the connected meshes are (x'_2, y'_1) and (x'_2, y'_2).

2.3.3. Processing of Original Mesh Sequence

Based on the above rules of mesh growth, starting from the mesh where the region's vertex is located, it gradually grows within the four mesh space in the order of the original region's vertex connection. When the mesh returns to the starting position again, all meshes that are penetrated by the region's boundary line form a mesh sequence. There may be some coincident meshes in the original mesh sequence, which can cause errors in subsequent calculation results. Further approximation processing is required for the original grid sequence, removing coincident meshes, and adding new meshes based on mesh connectivity rules to ensure that all adjacent meshes meet mesh connectivity rules (Figure 6).

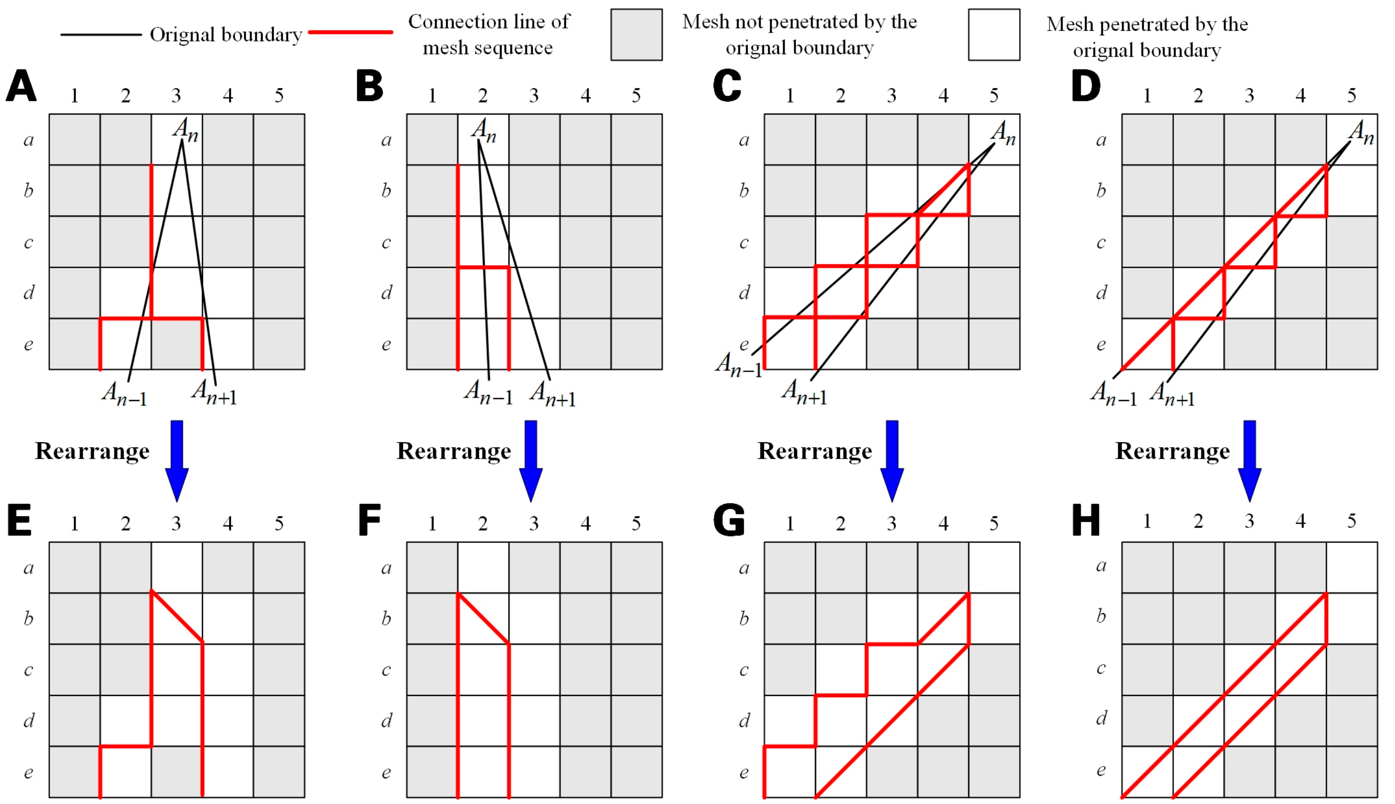

Figure 6. Schematic diagram of mesh sequence rearrangement. (**A–D**) Original mesh sequence; (**E–H**) Final mesh sequence.

2.4. Classification Processing of Meshes

Before merging two regions, it is necessary to classify the meshes in the two region mesh sequences into coincident meshes, overlapping meshes, valid meshes, and invalid meshes, and then the valid meshes connection segments are segmented.

2.4.1. Extracting Coincident Meshes

Firstly, it is necessary to extract meshes with the same coordinate in both regions and extract mesh index numbers of the mesh sequence to construct an array. The program flowchart is shown in Figure 7. By comparing each mesh coordinates of region A and region B in a loop, the mesh index numbers of two regions that coincide are found. The coincident mesh index numbers of region A are stored in array $N[m]$, and the coincident mesh index numbers of region B are also stored in array $N'[m]$. Therefore, there is a one-to-one correspondence between the coincident mesh numbers of region A and region B.

After extracting the coincident meshes, the mesh sequence shown on the left in Figure 8 can be obtained, which includes four typical cases. The first mesh A_0 of the mesh sequence in case 1 is coincident, and the last mesh is noncoincident, so it can be directly used for subsequent overlapping mesh extraction. In other cases, it is necessary to rearrange the mesh sequence so that the first mesh is the continuous coincident mesh with the minimum index number in the original sequence. All meshes before this mesh maintain the original sorting and are shifted to the last mesh of the original grid sequence to form a new mesh sequence, as shown on the right side of Figure 8.

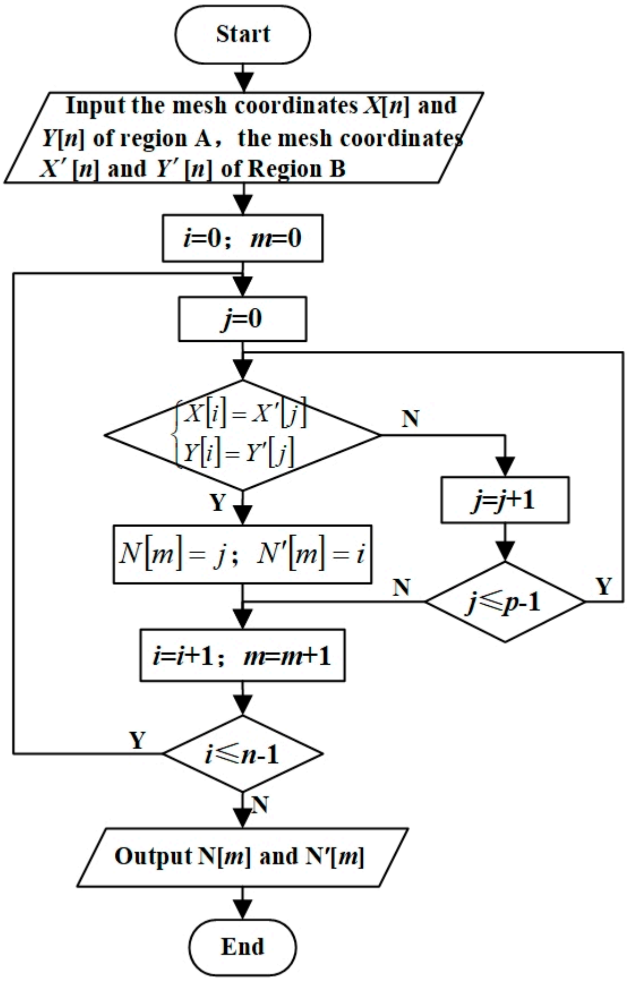

Figure 7. Flowchart of coincident mesh extraction subroutine.

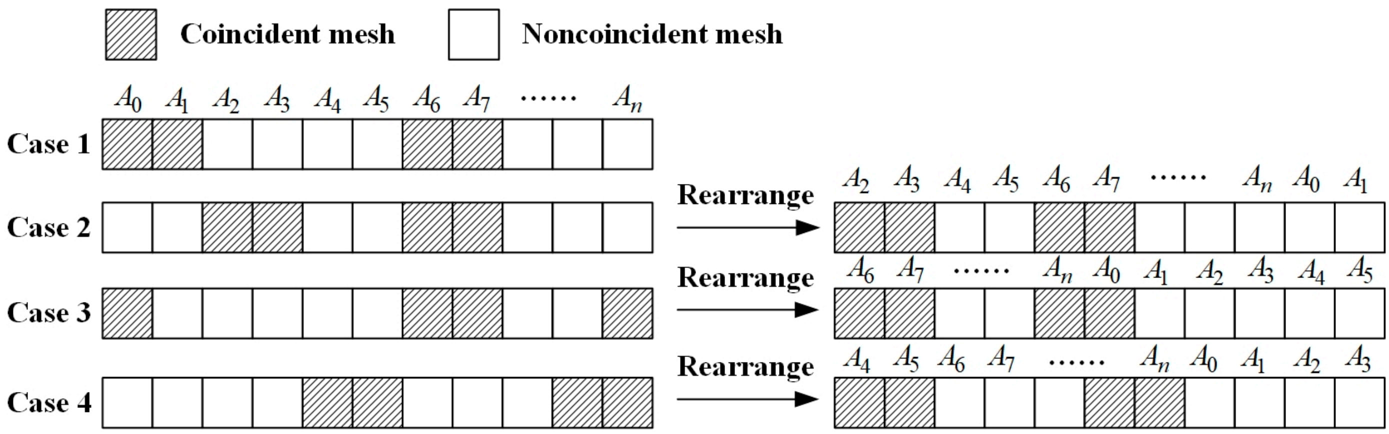

Figure 8. Schematic diagram of grid sequence rearrangement.

2.4.2. Extracting Overlapping Meshes

After determining the coincident meshes of the two regions, it is necessary to extract overlapping meshes and binarize the noncoincident meshes. The flow chart of the binarization program for grid in region A is shown in Figure 9. The program needs to input the coincident mesh index number array N′[m] of region A, and mesh coordinates array of both region A and region B. By detecting the difference between adjacent coincident mesh index numbers, mesh index numbers with a difference of more than 1 are obtained, which are the boundary points between coincident and noncoincident meshes. These meshes are called overlapping meshes, and they all exist in pairs.

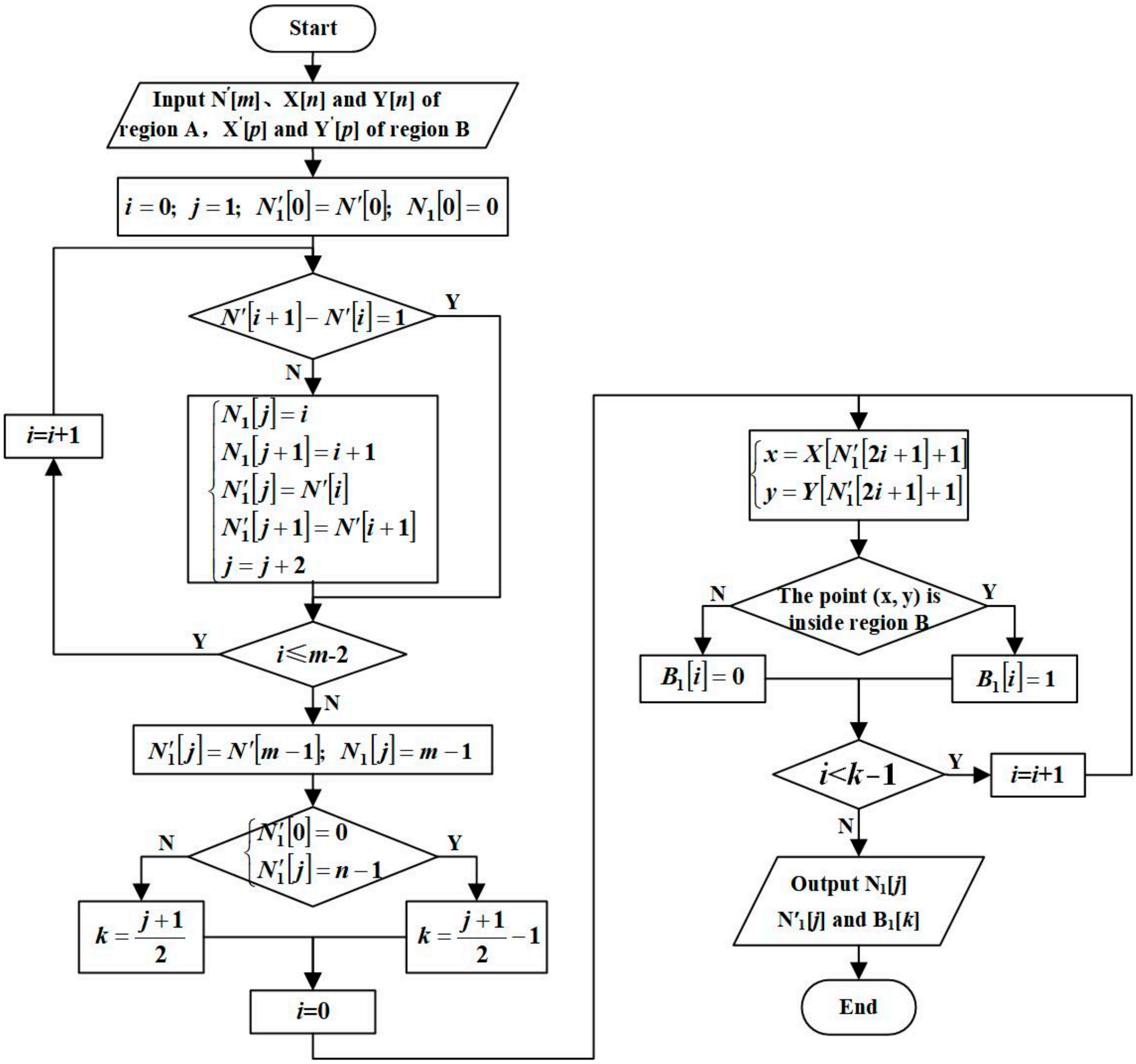

Figure 9. Extraction of overlapping meshes and binarization of mesh connection segment.

When $N'[i+1] - N'[i] \neq 1$, store i and $i + 1$ in the overlapping mesh serial number array $N_1[j]$, store $N'[i+1]$ and $N'[i]$ into the overlapping mesh index array $N_1'[j]$. Finally, add 0 and m-1 as the first and last elements of $N_1[j]$, and $N'[0]$ and $N'[m-1]$ as the first and last elements of $N_1'[j]$.

The noncoincident meshes between two adjacent overlapping meshes are called a connecting segment, and region A is divided into k connecting segments. When Equation $N_1'[0] = 0$ and $N_1'[j] = n-1$ are true, that is, the first and last overlapping meshes are the first and last meshes of region A, $k = \frac{j+1}{2}$ can be obtained; otherwise, $k = \frac{j+1}{2} - 1$. The connecting segment will be assigned a flag of 1 when all the meshes of this connecting segment are inside the other region; otherwise, the connecting segment will be assigned a flag of 0. The mesh segment with a flag of 1 is called a valid mesh segment.

Due to the fact that all mesh attributes within the same grid connecting segment are the same, this algorithm uniformly extracts the meshes after the odd overlapping mesh for binarizing, and stores the results of all connecting segments in the binary flag array $B_1[k]$ of region A. Similarly, the connecting segments of region B can be binarized.

2.4.3. Deletion Rules for Invalid Overlapping Meshes

Starting from overlapping meshes, connect all valid mesh segments of two regions one by one to form a new region. Therefore, the more overlapping grids, the more times the grid connection segments are connected. In order to reduce unnecessary calculations, overlapping meshes that cannot be directly connected to form new regions are removed from the overlapping mesh sequence in advance. These deleted overlapping meshes are called invalid overlapping meshes.

As shown in Figure 10, A_i–A_{k+3} are the coincident meshes of region A, and B_m–B_{m+13} are the coincident meshes of a region corresponding to region A in region B one by one. The overlapping meshes A_i and A_{k+3} correspond to B_m and B_{m+13}, which are also overlapping meshes of region B, and such overlapping meshes are called valid overlapping meshes. The meshes of region B corresponding to A_{i+3}, A_j, A_{j+5}, and A_k are not overlapping meshes, and such overlapping meshes are called additional overlapping grids.

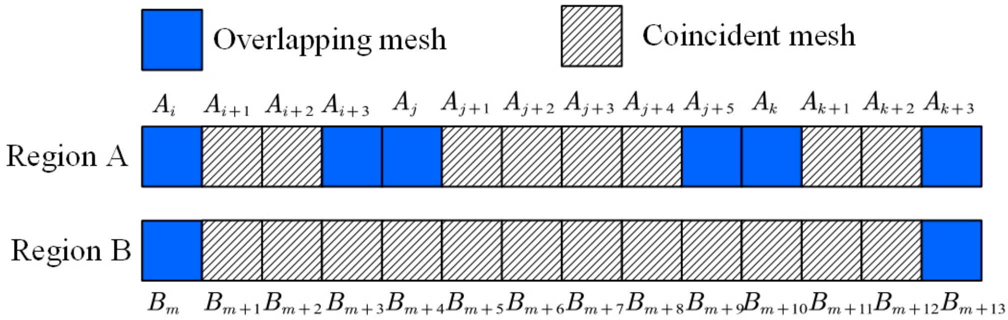

Figure 10. Schematic diagram of overlapping meshes.

Therefore, there are three kinds of overlapping grid pairs. Two overlapping meshes that are both valid overlapping meshes are called valid overlapping mesh pairs, such as B_m and B_{m+13}; if only one of the two overlapping meshes is a valid overlapping mesh, it is called an additional valid overlapping mesh pair, such as A_i and A_{i+3}; two overlapping grids that are both additional overlapping meshes are called additional overlapping grid pairs, such as A_j and A_{j+5}.

The flag of the mesh segment before the valid overlapping mesh A_i is a_1. The flag of the mesh segment between the additional overlapping meshes A_{i+3} and A_j is a_2. The flag of the mesh segment between the additional overlapping meshes A_k and A_{k+3} is a_3. The flag of the mesh segment after the valid overlapping mesh A_{k+3} is a_4; the flag of the mesh segment before the valid overlapping mesh B_m in region B is b_1. The flag of the mesh segment after valid overlapping mesh B_{m+13} is b_2.

Taking A_j and A_{j+5} as examples, analyze the deletion rules for additional overlapping mesh pairs. When $a_3 = 1$ and $a_1 = a_4 = 0$, retain additional overlapping grid pairs; otherwise, delete the additional overlapping mesh pairs and merge them into the mesh segment after A_{j+5}. Taking B_m and B_{m+13} as examples, analyze the deletion rules for valid overlapping mesh pairs. When $b_1 \neq b_2$ or $b_1 = b_2 = 1$ and $a_1 = a_4 = 1$, retain the valid overlapping mesh pairs; otherwise, delete the valid overlapping mesh pairs and merge the meshes between B_m and B_{m+13} into the mesh segment before B_m and delete the corresponding overlapping meshes in region A. Taking A_i and A_{i+3} as examples, analyze the deletion rules for additional valid overlapping mesh pairs. When $a_1 = a_4 = 0$ and $a_2 = 1$, retain additional valid overlapping mesh pairs; otherwise, delete additional valid overlapping meshes A_i and A_{i+3}.

2.4.4. Rearrange Mesh Segments

As shown in Figure 11, for the convenience of mesh connection, the mesh sequence after deleting invalid overlapping meshes is reordered, so that the first mesh segment is valid.

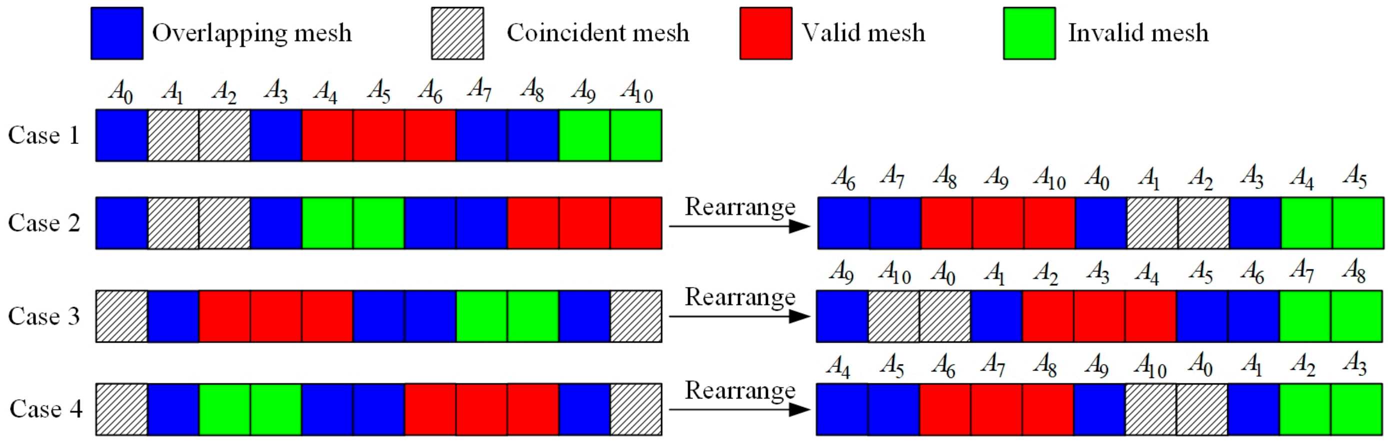

Figure 11. Schematic diagram of grid connection segment reordering.

2.5. Connection Rules for Mesh Segments

Use the region with a large number of valid mesh segments as the active region for region merging. When both ends of the valid mesh segment are valid overlapping mesh pairs, and the other sides of the two valid overlapping mesh pairs are invalid mesh segments, it is called a general valid mesh segment. When there is at least one valid mesh segment at the other side of two overlapping mesh pairs, it is called a dual valid mesh segment. When one end of the valid mesh segment is an additional valid overlapping mesh pair, it is called an additional valid mesh segment. Analyze the mesh connection rules during the region merging based on the types and numbers of valid mesh segments.

2.5.1. Connection Rules for General Valid Mesh Segments

As shown in Figure 12, the overlapping mesh index number n_1, n_2, n_3, n_4 of the active region. First, find the overlapping mesh index numbers n'_1 and n'_2 of the passive region corresponding to n_1 and n_2, and solve for the minimum value n'_{min} in n'_1 and n'_2, then the validity flag B_{min} of the mesh segment after n'_{min} is used to search for another overlapping mesh pair n'_3 and n'_4. B = 0 indicates that the mesh segment of the passive region needs to be reversed during mesh connection, otherwise the original order will be maintained. $B_C = 1$ indicates that the merging result is a closed region, otherwise it is an unclosed region. The starting point of the active region's valid mesh segment is s, and the length is l. The starting point of the passive region's valid mesh segment is s', and the length is l'. The merging situations of general valid mesh segments are shown in the Figure 13.

2.5.2. Connection Rules for Dual Valid Mesh Segments

The region formed by the dual valid mesh segment and another region must be closed, as shown in the program flowchart in Figure 14. It is necessary to extract the index numbers m'_1, m'_2, m'_3, and m'_4 of each overlapping mesh in the overlapping mesh array of the passive region. Equation $|m'_3 - m'_2| = 1$ represents that n'_3 and n'_2 are adjacent in the overlapping mesh array.

2.5.3. Connection Rules for Additional Valid Overlapping Mesh Segments

When the mesh segment of the active region is an additional valid, n'_2 and n'_3 are not overlapping meshes of the passive region. The valid mesh segment of the active region is connected to n_2 and n_3 to form a closed region, so $s = n_2$ and $l = n_3 - n_2 + 1$.

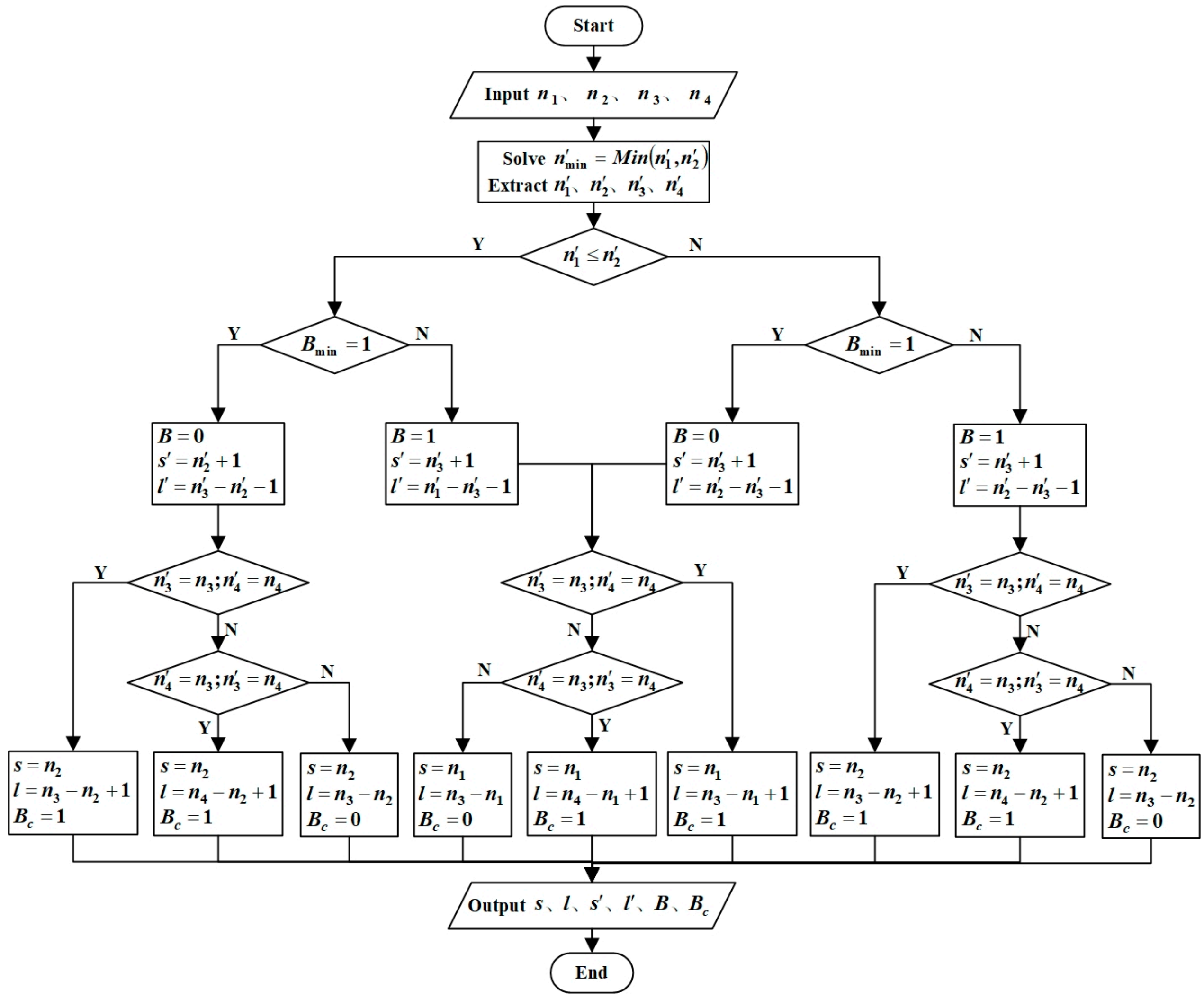

Figure 12. Flowchart of mesh connection subroutine for general valid mesh connection.

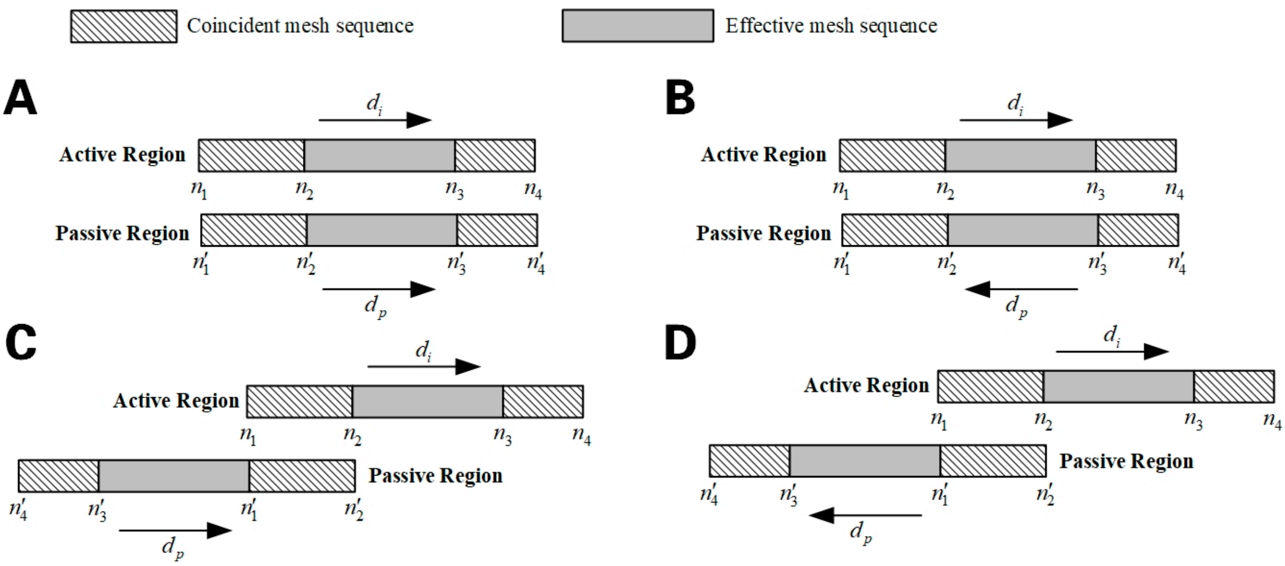

Figure 13. Schematic diagram of mesh connecting situations. Note: (**A**): $n'_{min} = n'_1, B_{min} = 1$, $n'_1 \le n'_2 \le n'_3 \le n'_4$. (**B**): $n'_{min} = n'_1, B_{min} = 0$, $n'_1 \le n'_2 \le n'_3 \le n'_4$. (**C**): $n'_{min} = n'_2, B_{min} = 1$, $n'_2 \le n'_1 \le n'_3 \le n'_4$. (**D**): $n'_{min} = n'_2, B_{min} = 0$, $n'_2 \le n'_1 \le n'_3 \le n'_4$.

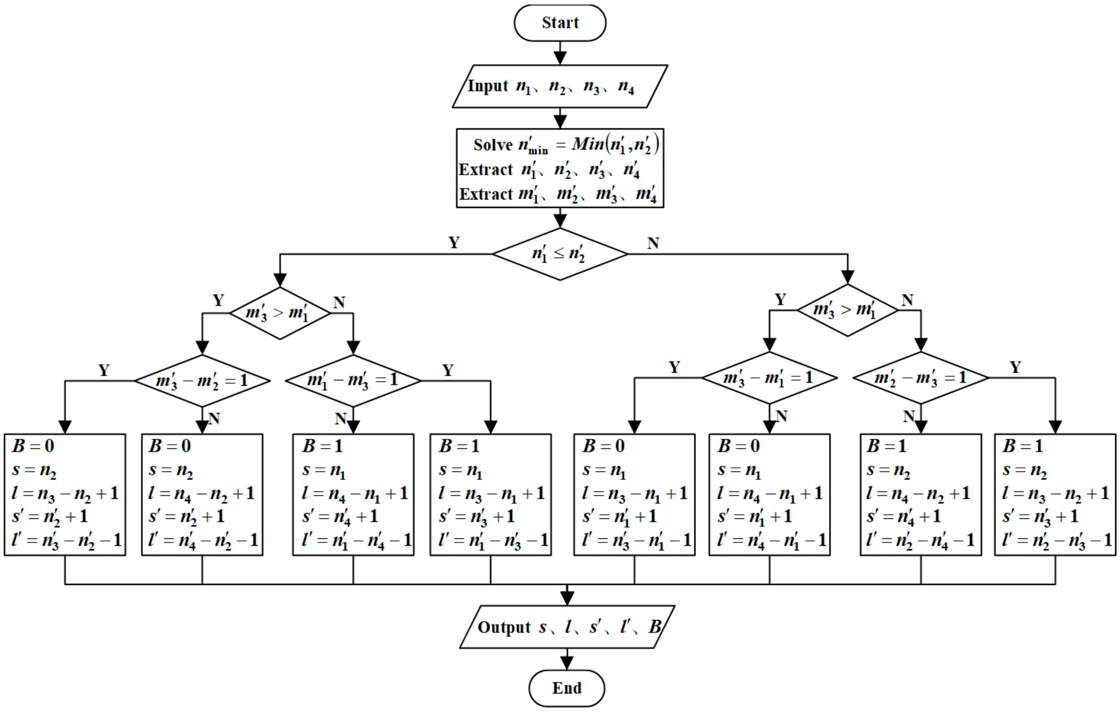

Figure 14. Flowchart of mesh connection subroutine for dual valid mesh segments.

2.5.4. Connection Rules for Special Situations

When the active region only contains one pair of overlapping grids and one segment of invalid or valid meshes, the program flowchart for mesh connection is shown in Figure 15. For active region A, n is the total number of meshes, n_1 and n_2 are overlapping meshes and B_a is the validity flag of the mesh segment. For passive region B, m is the total number of meshes, n'_1 and n'_2 are overlapping meshes and B_b is the validity flag of the mesh segment. The total number of coincident meshes in two regions is n'. The number of new regions is B_n.

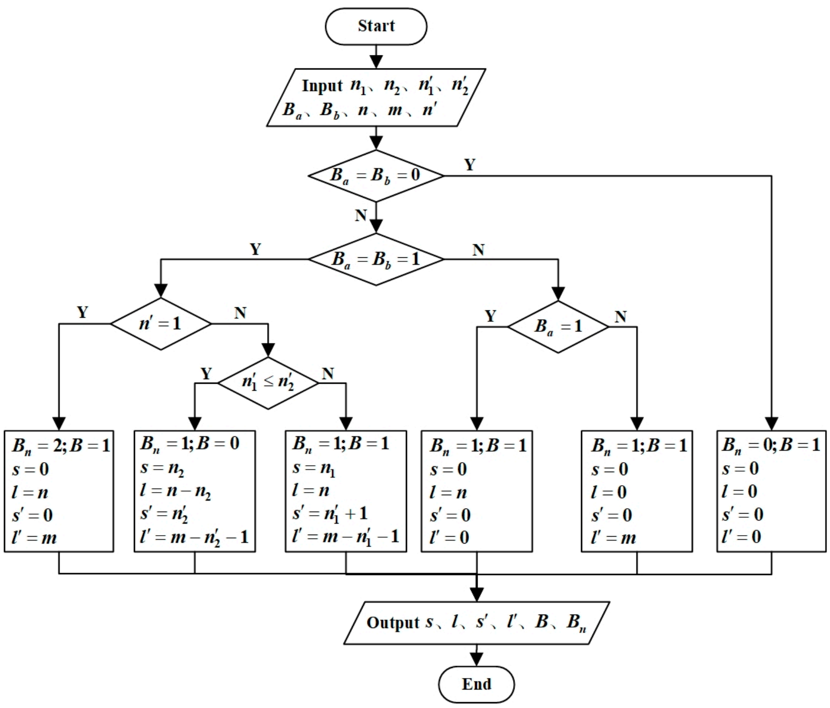

Figure 15. Flowchart of mesh connection subroutine in special situations.

When all the elements of the mesh sequence are coincident grids, the active region and passive region are completely coincident, and the connection result is a closed region, which is an active region.

When there is no overlapping mesh between two regions, the subroutine is shown in Figure 16, the mesh connection result is two independent regions, or one of them, or an empty set.

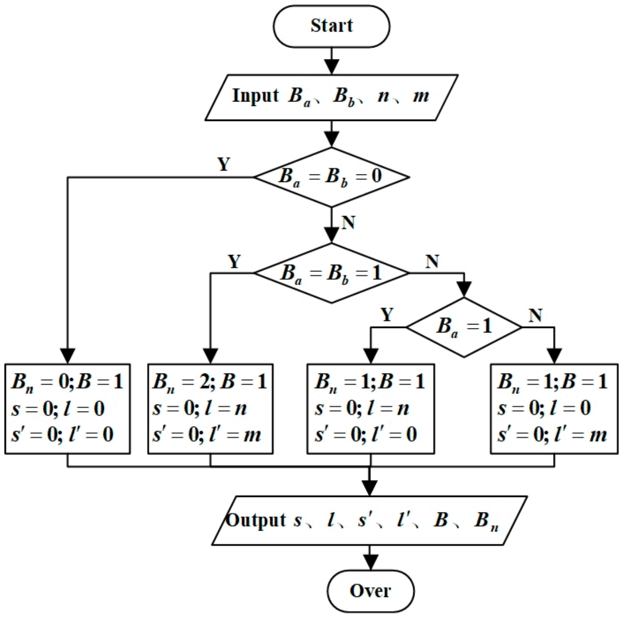

Figure 16. Flowchart of connection rules without coincident meshes.

3. Results

Design the software based on algorithm principles, and the software interface is shown in Figure 17. The software includes a data analysis mode selection area, which can be used for real-time online analysis or offline extraction of data documents for analysis. This interface selects the offline data document analysis mode. The mesh scale setting button can be used to set d_g.

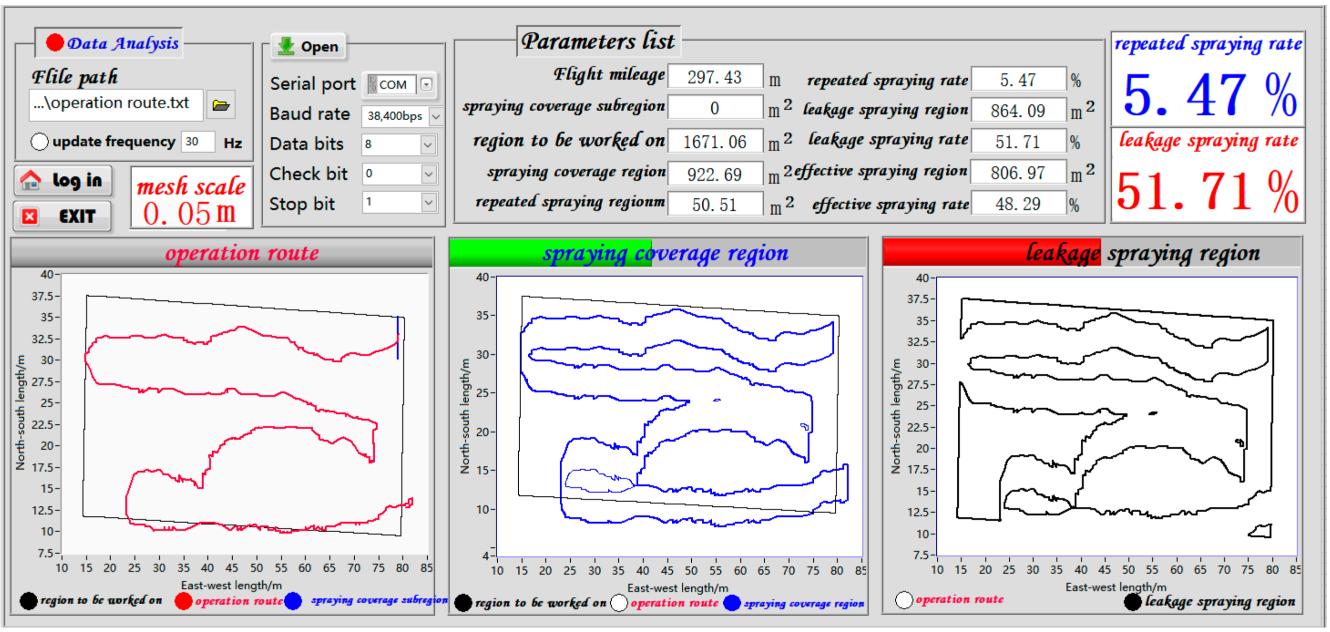

Figure 17. Screenshot of the software interface.

3.1. Experiment of Theoretical Route

The polygon composed of points A_1(7.5,5), A_2(32.5,5), A_3(42.5,30), A_4(5,37.5), and A_5(2,20) is taken as the region to be worked on. The theoretically planned routes [42] are imported into the designed plant protection UAV operation map reconstruction software. The mesh scale is set to 0.05 m, and the software automatically generates the operation coverage region. Compare it with the theoretical calculation value to verify the software's operation and calculation error under different starting operation directions. The experimental results are shown in Table 1. The experimental results show that the software calculation results are consistent with the theoretical calculation results. The maximum relative error of the spraying coverage region is 0.53 ‰, the maximum relative error of the repeated spraying coverage region is 6.8 ‰, the maximum relative error of the repeated spraying rate is 6 ‰, and the errors of the leakage spraying coverage region are all 0.

Table 1. Experimental results of theoretical route.

Starting Spraying Direction	Spraying Coverage Region (m^2)		Repeated Spraying Region (m^2)		Repeated Spraying Rate (%)	
	Theoretical	Experimental	Theoretical	Experimental	Theoretical	Experimental
$\overrightarrow{A_3A_2}$	1162.76	1162.84	70.66	71.07	6.08	6.11
$\overrightarrow{A_3A_4}$	1297.74	1298.43	130.42	131.31	10.05	10.11
$\overrightarrow{A_1A_2}$	1115.54	1115.67	58.37	58.15	5.23	5.21

3.2. Experiment of Irregular Route

Import the irregular route into the software. The starting spraying direction angle is 180°, the spraying span of the UAV is 4 m, and the mesh scale is 0.05 m. The algorithm automatically calculates the spraying coverage region and leakage region. The operation results are shown in Figure 17. Export the data to obtain Figure 18. The solving process of the algorithm includes union, intersection, and remainder operations between arbitrary polygons.

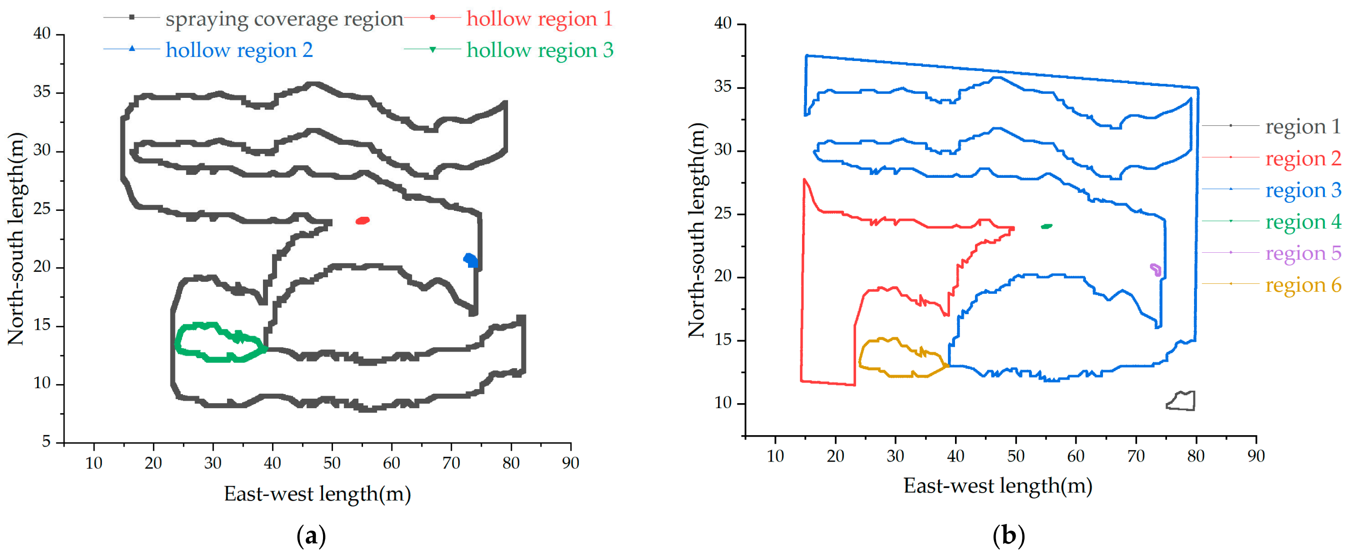

Figure 18. Experimental results of irregular route. (**a**) Spraying coverage region; (**b**) Leakage spraying region.

From Figure 18a, it can be seen that the software has calculated four spraying coverage regions, where the black curve represents the outer boundary of the spraying coverage region, meaning that the interior of the region is the spraying coverage region; the other three are the inner boundaries of the spraying coverage region, and the interior of this region is the leakage spraying regions. From Figure 18b, it can be seen that the software has calculated six mutually independent leakage spraying regions, and the regions inside each boundary line are the leakage spraying regions. The results of region merging are all closed regions, and the connections between each vertex comply with mesh connectivity rules, resulting in accurate and reliable calculation results.

Comparing the running time of the software constructed in different regions under different data points can obtain Table 2. From the table, it can be seen that with the increase in the number of regions and meshes, the running time of constructing the coverage region does not change significantly, while the running time of constructing the leakage spraying region significantly increases. As the number of regions and meshes increases, the difference in computational time between constructing spraying coverage region and constructing leakage spraying region becomes larger and larger. This is mainly due to the need to use the boundary of the pesticide spraying region to be worked on when calculating the leakage spraying region, which is relatively large and generates many meshes during the meshing process. Finding overlapping meshes and processing grid sequences require a lot of time.

Table 2. Running time of constructing different regions of the software.

Number of Data Points	Constructing Spraying Coverage Region			Constructing Leakage Spraying Region		
	Running Time (ms)	Number of Merging Regions	Maximum Number of Meshes in the Region	Running Time (ms)	Number of Merging Regions	Maximum Number of Meshes in the Region
100	783	1	888	1087	2	4882
200	1560	1	2562	5940	2	6542
300	2386	2	4340	16,982	3	9892
400	3274	3	5603	32,556	4	12,418
500	4267	4	6788	51,843	5	14,788
600	6004	4	9041	76,816	6	19,088
645	6663	4	9945	90,297	6	20,896

By comparing the running time of the software constructing spraying coverage region with different mesh scales, Table 3 can be obtained. From the data analysis in the table, it can be seen that when the mesh scale is 0.05 m, the average time for merging the regions of each data point is 10 milliseconds, which means that the algorithm can adapt to coordinate data update frequencies within 100 Hz; when the mesh scale is 0.02 m, the average time for merging regions of each data point is 33 milliseconds, which means that the algorithm can adapt to coordinate data update frequencies within 30 Hz.

Table 3. Running time of constructing spraying coverage region with different mesh scales.

Mesh Scale (m)	Running Time (ms)						
	100 Data	200 Data	300 Data	400 Data	500 Data	600 Data	645 Data
0.05	783	1560	2386	3274	4267	6004	6663
0.02	975	2274	4188	7198	12,166	18,445	21,021

Compared to the mesh scale of 0.02 m, the relative error of the spraying coverage region in the calculation result at the mesh scale of 0.05 m is 0.3‰, the error of the repeated spraying region is 5.9‰, and the relative error of the leakage spraying region is 6.8‰. Therefore, the error of the calculation result when the mesh scale is 0.05 m is within 7‰, but it increases the computational speed by three times.

Therefore, the performance of this algorithm is determined by the mesh scale. The larger the mesh scale, the simpler the boundary representation of the region, the smaller the storage space is occupied, and the shorter the calculation time of region merging, but the lower the accuracy of the calculation result. The smaller the mesh scale, the more accurate the boundary representation of the region, the larger the storage space occupied, and the longer the calculation time of region merging, but the higher the accuracy of the calculation results.

4. Discussion

In the process of solving the spraying coverage region, before forming a hollow region, the union calculation between regions is mainly carried out; after forming a hollow region, it is also necessary to complete the intersection operation between the boundary inside the hollow region and the new region. By solving the residual set of the spraying coverage region in the pesticide spraying region, the leakage coverage region can be obtained.

The algorithm in this study solves the dynamic calculation problem of union, intersection, and residual sets between arbitrary polygonal regions, and combines real-time flight parameter information of plant protection UAV to accumulate and calculate the distribution of fog droplets during the aerial spraying process in real time. Compared to the research results of [27–29], this study achieved the goal of real-time evaluation of droplet distribution, providing a reliable basis for precise planning, real-time replanning, and operation quality evaluation of plant protection UAV routes.

The performance of this algorithm is determined by the mesh scale. The larger the mesh scale, the simpler the boundary representation of the region, the smaller the storage space is occupied, and the shorter the calculation time of region merging, but the lower the accuracy of the calculation result. The smaller the mesh scale, the more accurate the boundary representation of the region, the larger the storage space occupied, and the longer the calculation time of region merging, but the higher the accuracy of the calculation results.

The spraying quality of plant protection UAV depends on the spraying coverage region and the uniformity of droplet distribution. The spraying coverage region in this manuscript can characterize the deposition area of the droplet on a two-dimensional plane, but it is not yet possible to characterize the uniformity of the deposition of the droplet in the spray coverage plane area and the deposition of the crop canopy at different heights. In further research, based on the experiments to obtain the uniformity data of droplet deposition under different parameters, the droplet deposition density model is established, and then the droplet deposition density is generated everywhere in the region while constructing the spraying coverage region, generating an accurate droplet distribution map.

5. Conclusions

This research proposes an algorithm for merging arbitrary polygonal regions based on boundary meshing, and develops software based on this algorithm. Combined with information collection hardware, a system for evaluating the spraying coverage region of plant protection UAV is constructed. The coverage region of pesticide droplets is constructed, displayed, and analyzed. The main conclusions were:

(1) This algorithm can be used to dynamically construct the spraying coverage region of plant protection UAV, and it can provide a new idea for solving the intersection, union and difference of arbitrary polygons;

(2) The system can calculate the spraying coverage region and leakage spraying region, with a calculation error of less than 7‰, generate spraying effect parameters, and mark the regions that need to be inspected;

(3) Import the operational parameters of the plant protection UAV planning route, and use the algorithm in this paper to generate spraying coverage region, which can be used to estimate the operational effect of the planned route, thereby selecting the optimal operational parameters and guiding the precise planning of the operational route. In the process of plant protection UAV spraying, the quality of the operation can also be evaluated based on the real-time generated spraying coverage region generated by this algorithm. At the same time, the operation route can be replanned in real-time based on the generated leakage spraying region, which can assist in

precise spraying and is expected to achieve the best control effect with the lowest pesticide consumption;

(4) Further research should be carried out to clarify the distribution of droplet deposition density under different operating parameters, add it to the spraying coverage region to generate a complete droplet distribution map, and combine the droplet distribution map at different heights to generate droplet deposition at different heights of crop canopy in three-dimensional space.

Author Contributions: Conceptualization and methodology, K.Z., J.C., and L.Z.; software, K.Z. and Y.L.; validation, K.Z., L.Z., and P.M.; formal analysis, investigation, and resources, J.C., P.M., B.Y., and Y.L.; data curation, J.C. and B.Y.; writing—original draft preparation, K.Z. and L.Z.; writing—review and editing, K.Z., L.Z., J.C., P.M., B.Y., and Y.L.; funding acquisition, K.Z. and L.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Key Research and Development Program of China (grant number 2022YFD1900805), Key R&D and Promotion Projects in Henan Province (Science and Technology Development) (grant number 232102110264), Key Research Projects of Higher Education Institutions in Henan Province (grant number 23A510014 and 22A510009), Xinyang Normal University Young Key Teacher Funding Project (grant number 2022GGJS-04)), and PhD research start up the foundation of Henan University of Science and Technology (No. 13480025).

Data Availability Statement: Not applicable.

Acknowledgments: We would like to thank South China Agricultural University for providing the experimental site.

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

References

1. He, X. Rapid development of unmanned aerial vehicles (UAV) for plant protection and application technology in China. *Outlooks Pest Manag.* **2018**, *29*, 162–167. [CrossRef]
2. Guru-Pirasanna-Pandi, G.; Adak, T.; Gowda, B.; Patil, N.; Annamalai, M.; Jena, M. Toxicological effect of underutilized plant, *Cleistanthus collinus* leaf extracts against two major stored grain pests, the rice weevil, *Sitophilus oryzae* and red flour beetle, *Tribolium castaneum*. *Ecotoxicol. Environ. Saf.* **2018**, *154*, 92–99. [CrossRef] [PubMed]
3. Gao, D.; Sun, Q.; Hu, B.; Zhang, S. A Framework for Agricultural Pest and Disease Monitoring Based on Internet-of-Things and Unmanned Aerial Vehicles. *Sensors* **2020**, *20*, 1487. [CrossRef]
4. Yang, K.; Yang, G.Y.; Fu, S.I.H. Research of control system for plant protection UAV based on Pixhawk. *Procedia Comput. Sci.* **2020**, *166*, 371–375. [CrossRef]
5. Wang, J.; Lv, X.; Wang, B.; Lan, X.; Yan, Y.; Chen, S.; Lan, Y. Numerical Simulation and Analysis of Droplet Drift Motion under Different Wind Speed Environments of Single-Rotor Plant Protection UAVs. *Drones* **2023**, *7*, 128. [CrossRef]
6. Lan, Y.; Chen, S.; Deng, J.; Zhou, Z.; Ouyang, F. Development situation and problem analysis of plant protection unmanned aerial vehicle in China. *J. South China Agric. Univ.* **2019**, *40*, 217–225.
7. Yang, S.; Xu, P.; Jiang, S.; Zheng, Y. Downwash characteristics and analysis from a six-rotor unmanned aerial vehicle configured for plant protection. *Pest Manag. Sci.* **2022**, *78*, 1707–1720. [CrossRef] [PubMed]
8. Qin, W.; Chen, P.; Wang, B. Productivity model and experiment of field crop spraying by plant protection unmanned aircraft. *Front. Plant Sci.* **2023**, *14*, 1168228. [CrossRef] [PubMed]
9. Hu, H.; Kaizu, Y.; Huang, J.; Furuhashi, K.; Zhang, H.; Li, M.; Imou, K. Research on Methods Decreasing Pesticide Waste Based on Plant Protection Unmanned Aerial Vehicles: A Review. *Front. Plant Sci.* **2022**, *13*, 811256. [CrossRef]
10. Chen, H.; Lan, Y.; Fritz, B.K.; Hoffmann, W.C.; Liu, S. Review of agricultural spraying technologies for plant protection using unmanned aerial vehicle (UAV). *Int. J. Agric. Biol. Eng.* **2021**, *14*, 8–49. [CrossRef]
11. Qi, H.X.; Chen, P.C.; Lan, Y.B.; Huang, X.Y.; Wang, G.B. Experimental Study on Droplet Deposit Distribution of Different Electric UAVs in Rice Fields. *J. Agric. Mech. Res.* **2019**, *41*, 147–151.
12. Jiang, K.; Wang, Z.H.; Fan, J.R.; Zhang, M.L. Droplet Deposition Rules of Multi Rotor UAV Flight Loads Impact. *J. Agric. Mech. Res.* **2020**, *42*, 25–32.
13. Wang, G.B.; Han, Y.X.; Li, X.; Andaloro, J.; Chen, P.C.; Hoffmann, W.C.; Han, X.Q.; Chen, S.D.; Lan, Y.B. Field evaluation of spray drift and environmental impact using an agricultural unmanned aerial vehicle (UAV) sprayer. *Sci. Total Environ.* **2020**, *737*, 139793. [CrossRef] [PubMed]
14. Vitória, E.L.d.; Krohling, C.A.; Borges, F.R.P.; Ribeiro, L.F.O.; Ribeiro, M.E.A.; Chen, P.; Lan, Y.; Wang, S.; Moraes, H.M.F.e.; Furtado Júnior, M.R. Efficiency of Fungicide Application an Using an Unmanned Aerial Vehicle and Pneumatic Sprayer for Control of *Hemileia vastatrix* and *Cercospora coffeicola* in Mountain Coffee Crops. *Agronomy* **2023**, *13*, 340. [CrossRef]

15. Chen, S.; Lan, Y.; Zhou, Z.; Ouyang, F.; Wang, G.; Huang, X.; Deng, X.; Cheng, S. Effect of Droplet Size Parameters on Droplet Deposition and Drift of Aerial Spraying by Using Plant Protection UAV. *Agronomy* **2020**, *10*, 195. [CrossRef]
16. Yang, F.; Xue, X.; Cai, C.; Sun, Z.; Zhou, Q. Numerical Simulation and Analysis on Spray Drift Movement of Multirotor Plant Protection Unmanned Aerial Vehicle. *Energies* **2018**, *11*, 2399. [CrossRef]
17. Lan, Y.; Qian, S.; Chen, S.; Zhao, Y.; Deng, X.; Wang, G.; Zang, Y.; Wang, J.; Qiu, X. Influence of the Downwash Wind Field of Plant Protection UAV on Droplet Deposition Distribution Characteristics at Different Flight Heights. *Agronomy* **2021**, *11*, 2399. [CrossRef]
18. Zhan, Y.; Chen, P.; Xu, W.; Chen, S.; Han, Y.; Lan, Y.; Wang, G. Influence of the downwash airflow distribution characteristics of a plant protection UAV on spray deposit distribution. *Biosyst. Eng.* **2022**, *216*, 32–45. [CrossRef]
19. Ling, W.; Du, C.; Ze, Y.; Shumao, W. Research on the prediction model and its influencing factors of droplet deposition area in the wind tunnel environment based on UAV spraying. *IFAC-Pap* **2018**, *51*, 274–279. [CrossRef]
20. Qi, P.; Zhang, L.; Wang, Z.; Han, H.; Müller, J.; Li, T.; Wang, C.; Huang, Z.; He, M.; Liu, Y.; et al. Effect of Operational Parameters of Unmanned Aerial Vehicle (UAV) on Droplet Deposition in Trellised Pear Orchard. *Drones* **2023**, *7*, 57. [CrossRef]
21. Wen, S.; Han, J.; Ning, Z.; Lan, Y.; Yin, X.; Zhang, J.; Ge, Y. Numerical analysis and validation of spray distributions disturbed by quad-rotor drone wake at different flight speeds. *Comput. Electron. Agric.* **2019**, *166*, 105036. [CrossRef]
22. Wang, C.; Song, J.; He, X.; Wang, Z.; Wang, S.; Meng, Y. Effect of flight parameters on distribution characteristics of pesticide spraying droplets deposition of plant-protection unmanned aerial vehicle. *Trans. Chin. Soc. Agric. Eng.* **2017**, *33*, 109–116.
23. Ribeiro, L.F.O.; Vitória, E.L.d.; Soprani Júnior, G.G.; Chen, P.; Lan, Y. Impact of Operational Parameters on Droplet Distribution Using an Unmanned Aerial Vehicle in a Papaya Orchard. *Agronomy* **2023**, *13*, 1138. [CrossRef]
24. Hanif, A.S.; Han, X.; Yu, S.-H. Independent Control Spraying System for UAV-Based Precise Variable Sprayer: A Review. *Drones* **2022**, *6*, 383. [CrossRef]
25. Xu, L.; Yang, Z.; Huang, Z.; Ding, W.; Buck-Sorlin, G. Effects of flight parameters for plant protection UAV on droplets deposition rate based on a 3D simulation approach. *Int. J. Agric. Biol. Eng.* **2023**, *16*, 66–72. [CrossRef]
26. Fan, G.; Liu, Z.; Qin, Y.; Long, B.; Li, H.; Li, J. Airflow characteristics of rotorcraft plant protection UAV operating in rice fields. *Biosyst. Eng.* **2023**, *226*, 209–222. [CrossRef]
27. Hu, J.; Wang, T.; Yang, J.; Lan, Y.; Lv, S.; Zhang, Y. WSN-Assisted UAV Trajectory Adjustment for Pesticide Drift Control. *Sensors* **2020**, *20*, 5473. [CrossRef]
28. Faiçal, B.S.; Costa, F.G.; Pessin, G.; Ueyama, J.; Freitas, H.; Colombo, A.; Fini, p.; Villas, L.; Osório, F.; Vargas, P.; et al. The use of unmanned aerial vehicles and wireless sensor networks for spraying pesticides. *J. Syst. Archit.* **2014**, *60*, 393–404. [CrossRef]
29. Faiçal, B.S.; Freitas, H.; Gomes, P.H.; Mano, L.Y.; Pessin, G.; de Carvalho, A.C.; Krishnamachari, B.; Ueyama, J. An adaptive approach for UAV-based pesticide spraying in dynamic environments. *Comput. Electron. Agric.* **2017**, *138*, 210–223. [CrossRef]
30. Liu, Y.K.; Wang, X.Q.; Bao, S.Z.; Gomboši, M.; Žalik, B. An algorithm for polygon clipping, and for determining polygon intersections and unions. *Comput. Geosci.* **2007**, *33*, 589–598. [CrossRef]
31. Bae, S.W.; Okamoto, Y.; Shin, C.S. Area bounds of rectilinear polygons realized by angle sequences. *Comput. Geom.* **2019**, *83*, 9–29. [CrossRef]
32. Rivero, M.; Feito, F.R. Boolean operations on general planar polygons. *Comput. Graph.* **2000**, *24*, 881–896. [CrossRef]
33. Landier, S. Boolean operations on arbitrary polygonal and polyhedral meshes. *Comput. Aided Des.* **2017**, *85*, 138–153. [CrossRef]
34. Hao, J.; Sun, J.; Chen, Y.; Cai, Q.; Tan, L. Optimal Reliable Point-in-Polygon Test and Differential Coding Boolean Operations on Polygons. *Symmetry* **2018**, *10*, 477. [CrossRef]
35. Venturini, G.; Grossi, N.; Morelli, L.; Scippa, A. A Non-Uniform Offset Algorithm for Milling Toolpath Generation Based on Boolean Operations. *Appl. Sci.* **2022**, *13*, 208. [CrossRef]
36. Martinez, F.; Rueda, A.J.; Feito, F.R. A new algorithm for computing Boolean operations on polygons. *Comput. Geosci.* **2019**, *35*, 1177–1185. [CrossRef]
37. Wei, S.L.; Li, Y. A Simple Polygon Boolean Operation Based on Sorted Intersection Points. *Comput. Technol. Dev.* **2019**, *29*, 81–85.
38. Dong, W.M.; Maira, B.B.; Zhou, D.W.; Sun, J.G. Boolean Operations on General Planar Curved-Edge Polygons. *J. Comput. Aided Des. Comput. Graph.* **2003**, *15*, 1134–1140+1144.
39. Fan, J.F.; Kong, W.H.; Ma, T.; Zhou, C.H.; Ji, M. RaPC: Arasterization-based Polygon Cliping Algorithm and Its Error Analysis. *Acta Geod. Et Cartogr. Sin.* **2015**, *44*, 338–345.
40. Gao, Y.; Luo, J.X.; Qiu, H.P.; Wu, B. Arbitrary Polygons Boolean Operation Based on GPU Rasterization. *Comput. Eng.* **2018**, *44*, 301–306+314.
41. Zhou, C.; Chen, Z.; Li, M. A parallel method to accelerate spatial operations involving polygon intersections. *Int. J. Geogr. Inf. Sci.* **2018**, *32*, 2402–2426. [CrossRef]
42. Zhang, k.; Cui, J.Y.; Tu, Y.C.; Gong, K.; Cang, Y.P.; Wang, P. Design and Verification of an Algorithm of Precision Coverage Path Planning for Unmanned Aerial Vehicle. *J. Agric. Mech. Res.* **2022**, *2*, 15–22.