ITC 3/53

Information Technology and Control
Vol. 53 / No. 3/ 2024
pp. 736-757
DOI 10.5755/j01.itc.53.3.36021

Agricultural UAV Obstacle Avoidance System Based on a Depth Image Inverse Projection Algorithm and B-Spline Curve Trajectory Optimization Algorithm

Received 2024/01/11 | Accepted after revision 2024/04/25

HOW TO CITE: Liu, Z., Wen, S., Huang, G., Li, S., Deng, Z. (2024). Agricultural UAV Obstacle Avoidance System Based on a Depth Image Inverse Projection Algorithm and B-Spline Curve Trajectory Optimization Algorithm. *Information Technology and Control*, 53(3), 736-757. https://doi.org/10.5755/j01.itc.53.3.36021

Agricultural UAV Obstacle Avoidance System Based on a Depth Image Inverse Projection Algorithm and B-Spline Curve Trajectory Optimization Algorithm

Zhanlin Liu, Sheng Wen, Gaofeng Huang, Sipei Li, Zhiyin Deng

College of Engineering, South China Agricultural University, Guangzhou 510642, China; National Center for International Collaboration Research on Precision Agricultural Aviation Pesticide Spraying Technology, Guangzhou 510642, China; e-mails: liuzhanlin11@163.com; vincen@scau.edu.cn

Corresponding author: vincen@scau.edu.cn

As multirotor plant protection drones evolve, using them for pest control has emerged as a mainstream trend. Nevertheless, lemon orchards often feature trees planned in an unstructured manner, with irregularly distributed obstacles such as utility poles posing safety threats to the operational tasks of unmanned aerial vehicles (UAVs) used for plant protection. To address such issues, this study proposes a novel UAV obstacle avoidance system designed for lemon orchards. The system has two components: a sensing and mapping subsystem using a depth image inverse projection algorithm, and a path planning subsystem that utilizes B-spline curve trajectory optimization. The system comprises a hardware description and software integration of the UAV, a map construction algorithm to sense obstacles in front of the UAV, and a path planning algorithm for obstacle avoidance. Two experimental scenarios were developed to evaluate the system's flight performance: a flight test using the Gazebo simulation platform and a real-world test in a lemon orchard. In the simulation results, the flight trajectory's average deviation from the original path was 2.77 m, and the maximum yaw angular velocity reached 1.001 rad/s. In the real-flight experiments, the flight trajectory's average deviation from the original path was 2.90 m, and the maximum yaw angular velocity reached 1.545 rad/s. Both the simulation and real-flight experiments demonstrate that the system is effective in avoiding obstacles and planning paths in lemon orchards, providing a safe, smooth, and stable flight trajectory that meets the operational safety requirements for plant protection UAVs.

KEYWORDS: Plant protection UAV, Lemon orchard, Depth image, Inverse projection, B-spline curve, Trajectory optimization.

1. Introduction

The lemon is a green fruit that is known for its high nutritional and medicinal value [29, 33]. Since the turn of the 21st century, improvements in the living standards of Chinese people have considerably increased the demand for lemons, thereby driving rapid development in lemon cultivation within China [15, 18].

The southern region of China, which is characterized by a subtropical monsoon climate, offers optimal light, temperature, and precipitation for lemon growth. However, this environment also renders the lemons grown therein susceptible to various pests and diseases, including citrus red and yellow spiders, whiteflies, aphids, citrus thrips, and gray mold [3, 12, 26]. Therefore, the daily management of lemon cultivation, particularly in terms of controlling pests and diseases that affect lemon trees, requires substantial effort.

In southern China, lemon fruit trees are commonly planted in mountainous regions due to the scarcity of arable land per capita. Moreover, nonstructural planting techniques are widely utilized in these regions [22]. Given the restricted planting conditions and challenging topography, ground spraying equipment encounters challenges in maneuvering through mountainous orchards. Consequently, most plant protection activities in lemon orchards involve manual spraying, entailing a heavy workload, high operational intensity, and the need for considerable material and human resources.

In recent years, there has been a rapid development of multi-rotor plant protection UAVs. These UAVs offer advantages such as high speed, efficiency, simple structure, terrain versatility, and vertical takeoff and landing capabilities without requiring a runway [19, 28]. This development has positioned multi-rotor plant protection UAVs as crucial tools in combating agricultural pests and diseases in China. Moreover, agricultural aerial plant protection technology has emerged as a prominent research focus within the field of agricultural plant protection [6, 30]. Nevertheless, the implementation of agricultural aerial plant protection technology in plant protection operations within hilly lemon orchards faces several critical challenges that require immediate resolution.

Solving the path avoidance problem during UAV operations in unstructured planting territories is an urgent requirement. In general, path planning for UAV spraying operations in plantations is conducted offline, with path points selected via ground station maps to generate a “ $\prod$ ”-shaped path. However, orchards often contain static obstacles, including utility poles, streetlights, and trees of varying heights. These obstacles pose major safety threats to plant protection UAVs during path navigation. To address this issue, plant protection UAVs should be capable of generating and executing paths online for obstacle avoidance. Consequently, to ensure safety during tasks, particularly in hilly orchards, research into real-time obstacle avoidance path generation algorithms for plant protection UAVs is imperative.

Most existing agricultural UAV obstacle avoidance systems utilize both traditional vision and deep learning techniques to address the obstacle detection problem, formulating corresponding avoidance strategies based on the size of the obstacles. Wang et al. capitalized on the benefits of deep learning and sophisticated cameras, enabling UAVs to discern obstacle attributes such as categories, contours, and 3D spatial locations. Based on these detections, they detailed the development of collision avoidance strategies and the computation of optimal flight paths for evasion [38]. Mannar et al. introduced a monocular vision-based quadrotor control algorithm tailored for obstacle avoidance in forest environments, employing a weighted combination of texture features to calculate distances to the nearest obstacle across various longitudinal sections of image frames [25]. Stefas et al. presented a UAV system with vision-based obstacle detection and avoidance capabilities for navigation in apple orchards, utilizing binocular vision to construct a local obstacle map and identifying vacant cells within it as the next safe position for the UAV [32]. Castro et al. devised an online adaptive path planning scheme by integrating fast exploratory random trees and deep reinforcement learning algorithms. This approach is particularly effective in generating and controlling autonomous UAV trajectories, especially for tasks like detecting olive string traps, thereby circumventing potential obstacles in the environment [5].

However, the obstacle avoidance systems mentioned above still exhibit certain limitations. Autonomous navigation systems employing traditional vision techniques rely heavily on a single feature; when this feature changes, their robustness and reliability significantly diminish. Moreover, deep learning navigation schemes cannot assure that the deployment of deep learning models will meet the real-time requirements of UAV navigation, given the limited computational resources of current airborne equipment. Furthermore, the obstacle avoidance trajectory planning based on these schemes remains confined to the task level, lacking in path optimization for effective obstacle avoidance.

Prior to path planning, understanding the distribution of obstacles in an orchard is crucial to devising a collision-free operational path. Furthermore, the orchard map's accuracy considerably influences the quality of the path planning. Thus, substantial research efforts have been directed toward orchard map generation. Eviatar et al. developed a computer vision algorithm using UAV aerial images to distinguish trees from the background, converting RGB images to HSV, followed by a series of image manipulations to generate obstacle maps [9]. In contrast to traditional vision algorithms, Sun et al. proposed SADNet, a semantic segmentation network that uses UAV aerial images, designed to predict pixel labels in orchard images and construct orchard grid maps from the segmentation results [34]. Nevertheless, the orchard grid map's accuracy is inherently associated with that of the semantic segmentation results; in other words, poorer segmentation results lead to reduced map accuracy. Martina produced a dense three-dimensional (3D) point cloud map of a vineyard by deploying a fixed-wing UAV to capture multispectral images beforehand [23], utilizing the motion-based commercial software Agisoft Metashape (Agisoft LLC, located in St. Petersburg, Russia). This point cloud map was subdivided into a grid, enabling the manual customization of impassable areas, which culminated in the creation of a global map for UAV path generation. However, the aforementioned maps are generated in advance by UAVs equipped with diverse sensors to capture orchard imagery. These maps necessitate extensive postprocessing before being utilized for path planning, a process that falls short of the automation requirements necessary for efficient map construction.

Path planning after the generation of maps has become a common practice, with numerous scholars achieving preliminary results in studying UAV path planning for plant protection. Common obstacle avoidance algorithms for UAVs include Dijkstra [39], A* [14, 21], RRT* [13, 17], inform RRT* [11], Dubins [7, 21], the artificial potential field method [4, 35], the simulated annealing algorithm [2, 41], and the ant colony algorithm [27, 42]. Zhang et al. proposed an enhanced Dubins curve algorithm, which established an obstacle circle model and a small obstacle model based on UAV flight parameters and field obstacle types. This algorithm selects appropriate Dubins curves to generate multiple obstacle avoidance paths, followed by the application of a genetic algorithm to determine the optimal path [43]. Huang et al. enhanced the traditional A* algorithm by integrating dynamic heuristic functions, merging millimeter-wave radar and monocular camera data, and optimizing search and inflection points [16]. One drawback is the UAV's requirement to decelerate at inflection points, which results in a discontinuous flight process. Li et al. proposed a 3D spatial path planning model based on the R5DOS model, enhancing the model's directional layer, and developed the RJA star algorithm using an improved jump-point searching A star algorithm. This approach effectively avoids obstacles and reduces the UAV reaction time compared to traditional algorithms [20]. Zhang employed the traditional artificial potential field method for agricultural UAV trajectory planning, noted its tendency to fall into local minima in obstacle-rich environments [40]. To address this, he proposed an improved repulsion function that randomly generates a virtual target point near the local minima, effectively solving the problem of UAVs becoming trapped in local minima before reaching the target. Martina et al. utilized remote sensing maps of vineyards, obtained beforehand, to generate global maps. Both unmanned ground vehicles and UAVs conduct operational path planning based on these global maps, using RRT* for global path planning and DWA for local path planning [23]. The existing literature primarily concentrates on enhancing and optimizing traditional path search algorithms, resulting in improved algorithmic efficiency. Nevertheless, researchers often overlook back-end optimization, particularly in terms of trajectory smoothness, safety, and dynamic feasibility. Such oversights can result in poor trajectory quality, including unnecessary turns, which makes ensuring the flight stability of plant protection UAVs a complicated task.

To address the current challenges in map construction redundancy and trajectory optimization for plant protection UAVs in orchard path planning, we introduce a novel obstacle avoidance system. This system consists of two key components: a sensing and mapping subsystem based on the depth image inverse projection algorithm and a path planning subsystem utilizing B-spline curve trajectory optimization. The method used by the aforementioned scholars to obtain a global orchard map through prior acquisition lacks automation, and the map accuracy comprises postprocessing algorithms. By contrast, the sensing and mapping subsystem proposed herein employs a UAV with a depth camera to capture the depth images of obstacles. This subsystem constructs a map using an inverse projection algorithm, combining the rotation matrix and translation vectors to determine the point cloud positions of obstacles in a global coordinate system.

The problem of unoptimized trajectories is addressed by the path planning subsystem proposed in this study through the construction of global B-spline trajectories. This approach utilizes the inherently easy-to-optimize nature of B-spline curves for generating global paths. Furthermore, using a local obstacle map generated by the sensing and mapping subsystem, the A* algorithm is applied to identify a collision-free path within this map. Moreover, an optimization function concerning control points is established for the global B-spline trajectory based on this path, and path-guided optimization is conducted to generate the warm-up trajectory. Finally, a cost function encompassing trajectory safety, smoothness, and kinetic feasibility is formulated based on the control points, convex envelopes, and derivatives of the B-spline curve. This function is used to optimize the warm-up trajectory, generating a trajectory that meets all the requirements.

The study conducted herein can be summarized as follows, as shown in Figure 1, which illustrates the overall framework.

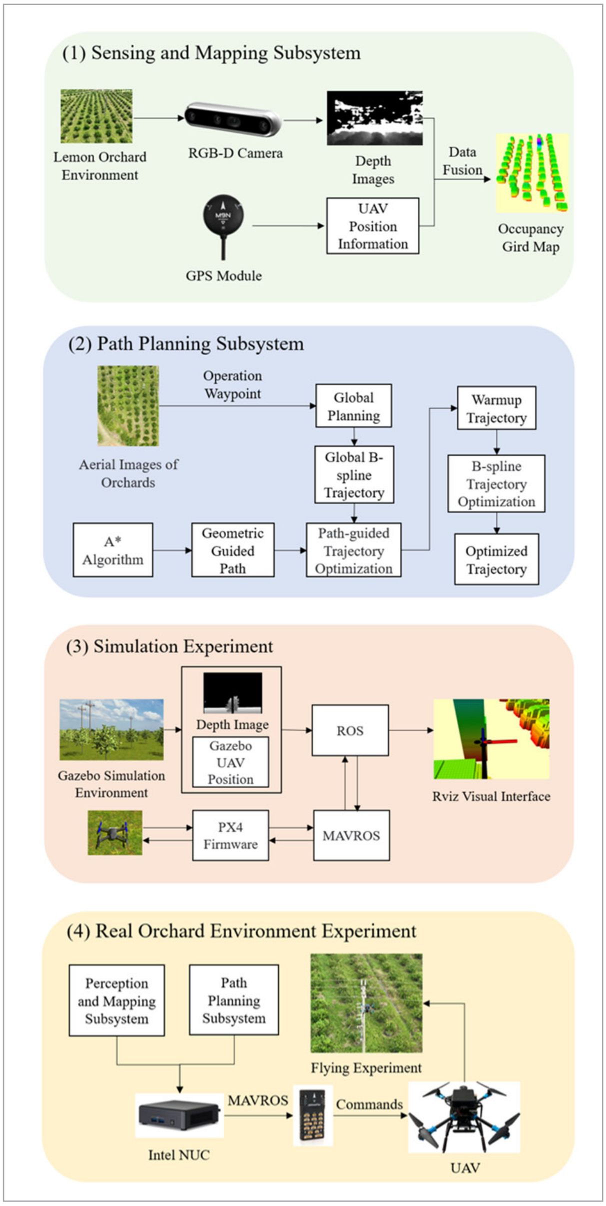
1 To address the redundancy and complexity in current map construction methods used for orchard path planning, a sensing and mapping subsystem utilizing a depth image back-projection algorithm is proposed, comprising the following steps:

a A UAV is equipped with a depth camera to capture the depth images of the orchard environment during flight.

b Utilizing the depth images of the orchard environment and the UAV's GPS data, the inverse projection algorithm is applied. The rotation matrix and translation vector are integrated to ascertain the point cloud positions of obstacles within the global coordinate system, thereby constructing a local map reflecting the UAV's current position.

Figure 1

Overall framework of the research content


(1) Sensing and Mapping Subsystem
Lemon Orchard Environment
RGB-D Camera
Depth Images
Data Fusion
Occupancy Gird Map
GPS Module
UAV Position Information
(2) Path Planning Subsystem
Aerial Images of Orchards
Operation Waypoint
Global Planning
Global B-spline Trajectory
Warmup Trajectory
B-spline Trajectory Optimization
A* Algorithm
Geometric Guided Path
Path-guided Trajectory Optimization
Optimized Trajectory
(3) Simulation Experiment
Gazebo Simulation Environment
Depth Image
Gazebo UAV Position
ROS
Rviz Visual Interface
PX4 Firmware
MAVROS
(4) Real Orchard Environment Experiment
Perception and Mapping Subsystem
Path Planning Subsystem
Flying Experiment
Intel NUC
MAVROS
Commands
UAV

2 To address the problem of insufficient trajectory optimization in current orchard path planning, a path planning subsystem utilizing B-spline curve trajectory optimization is proposed, which comprises the following steps:

a Considering the operational range of the lemon orchard, the locations and connection sequence of the UAV's operational path points are established, and these points are then connected to form a global path.

b Utilizing the property of B-spline curves being easy to optimize, the global path is adapted to create a global B-spline trajectory.

c The UAV follows the global B-spline trajectory, conducting trajectory feasibility checks against the local map from step 1(b). If an obstacle is detected along the intended trajectory, the A* algorithm is deployed to identify a preliminary collision-free path within this map.

d Leveraging the characteristics of the B-spline curve's control points, a path guidance optimization function is developed. This function minimizes the distance between the B-spline trajectory's control points and the geometric guidance path's sampling points, which ensures that the B-spline trajectory aligns closely with the guidance path and initially secures trajectory safety. The global B-spline trajectory is thus optimized into a preliminary trajectory at this stage.

e Considering the control points, convex hulls, and derivatives of the B-spline curve, a cost function is established to evaluate the trajectory's safety, smoothness, and dynamic feasibility. This function is employed to further refine the preliminary trajectory, yielding a trajectory that meets all the requirements.

3 Utilizing the robot operating system (ROS) in the Gazebo software package, an orchard environment is simulated to validate the obstacle avoidance algorithm.

4 An experimental platform for drones is constructed, and field experiments are conducted in a lemon orchard to evaluate the effectiveness of the drone's obstacle avoidance capabilities.

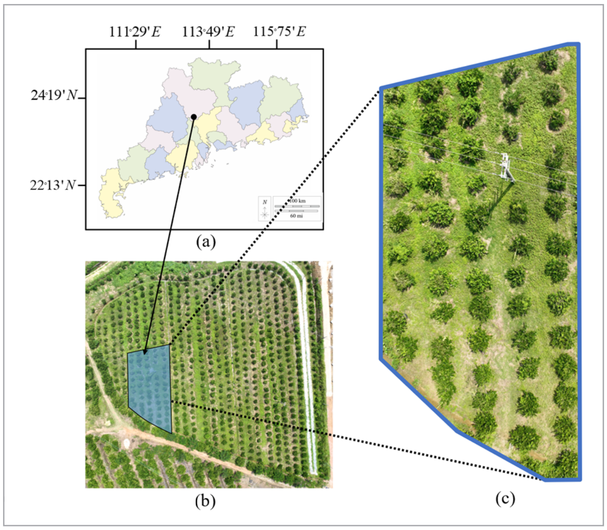
The remainder of this paper is structured as follows: Section 2 details the experimental site of the lemon orchard. Section 3 introduces the sensing and mapping subsystem. Section 4 outlines the path planning subsystem, encompassing global path generation and trajectory optimization. Section 5 presents the outcomes of the proposed algorithm within a simulation environment. Finally, Section 6 highlights the UAV experimental platform and the outcomes of the field experiments conducted in the lemon orchard.

2. Experimental Site of a Lemon Orchard

The experimental site was situated within a lemon orchard (23°27′ N, 112°68′ E) in Huadu District, Guangzhou City, Guangdong Province, China, as shown in Figure 2(a)-(b). The site features a subtropical monsoon climate conducive to lemon cultivation. Overall, the orchard exhibits an irregular shape and an unstructured distribution. As shown in Figure 3, the orchard exhibits the presence of scattered utility poles, which pose a major risk to the operating safety of plant protection UAVs. Based on the spatial arrangement of the obstacles inside the orchard, a designated portion was selected as the flight test area, as shown in Figure 2(c). Notably, the utility poles are positioned at the midpoint of the lemon tree columns, which represent a common obstruction encountered by plant protection UAVs during operational flights.

Figure 2

Geographic location and division of the experimental area.
(a) Geographic location of the experimental plot.
(b) Aerial view of the experimental field in the orchard. (c) Designated area for flight experiments



3. Sensing and Mapping Subsystem

To equip the UAV with the ability to detect obstacles in its path, an onboard sensing and mapping subsystem was developed, encompassing both hardware and software integration, as illustrated in Figure 4. In addition to facilitating obstacle detection for the UAV, this subsystem constructs maps detailing upcoming obstacles, which aids the path planning subsystem in its operations. These details are elaborated upon in this section.

Figure 3

The presence of electric poles dispersed throughout the lemon orchard, positioned between two lemon trees, represents a potential safety risk for plant protection drones

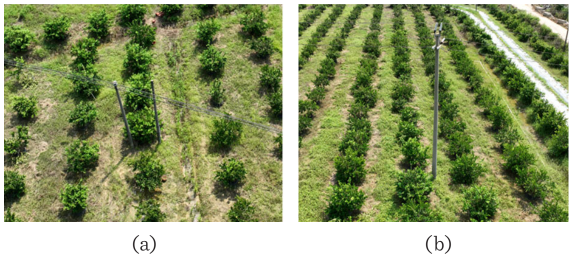
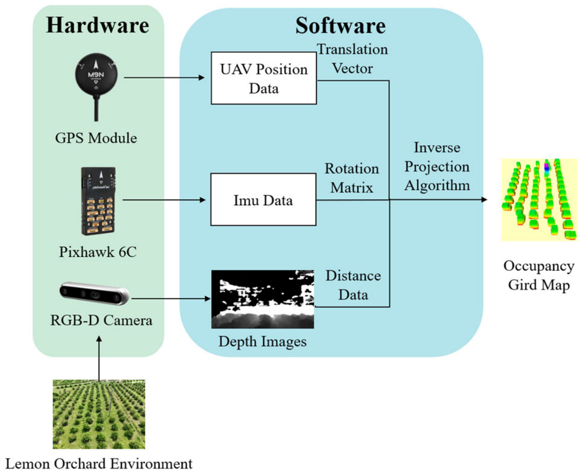


Figure 4

Sensing and mapping subsystem, including hardware integration and software integration



The lemon orchard presents an unstructured environment wherein obstacle distribution is unpredictable, including varied elements such as scattered trees, utility poles, and lemon canopies of differing heights. Figure 3 illustrates the distribution of obstacles in the orchard, highlighting utility poles as the most prevalent and disorganized form of obstruction. As shown in Figure 3, most utility poles are positioned between lemon trees, thereby posing a major safety hazard to the plant protection UAVs that operate in the orchard. A fundamental requirement for a drone to navigate these obstacles is its ability to accurately determine the location of the obstacles interspersed among the fruit trees. To address this challenge, we use a depth camera as a sensor to capture obstacle data. This camera generates depth images, enabling the UAV to easily ascertain the positions of the obstacles ahead.

The depth camera parameters are divided into internal and external categories. Internal parameters, set during factory production, define the camera's intrinsic properties. External parameters, conversely, represent the camera's relationship with the world coordinate system. The internal reference matrix **K** of the depth camera is represented as follows:

$$\mathbf{K}=\begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}, \tag{1}$$

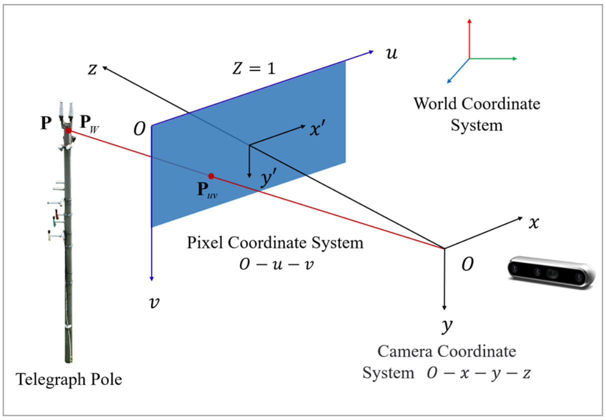
where f_x, f_y, c_x , and c_y are the internal parameters of the depth camera. In camera parameters, f_x represents the focal length in the horizontal direction, and f_y represents the focal length in the vertical direction. c_x is the horizontal coordinate of the optical center on the image plane, whereas c_y indicates the vertical coordinate of the principal point on the same plane. The outer reference matrix **T** of the depth camera is denoted as follows:

$$\mathbf{T}=\begin{bmatrix} \mathbf{R} & \mathbf{t} \\ 0 & 1 \end{bmatrix}. \tag{2}$$

The external reference matrix denotes the transformation relationship between the camera coordinate system and the world coordinate system. **R** refers to the rotation matrix, which signifies the rotational relationship between the world coordinate system and the camera coordinate system. Furthermore, **t** represents the translation vector, which indicates the translational relationship between the world coordinate system and the camera coordinate system. A utility pole represents a typical obstacle in a lemon orchard. In Figure 5, point $\mathbf{P}$ is identified as a point on the pole. Point $\mathbf{P}$ is denoted as $\mathbf{P}_W$ in the world coordinate system, $\mathbf{P}_C$ in the camera coordinate system, and $\mathbf{P}_{uv}$ in the pixel coordinate system, where $\mathbf{P}_W = \begin{bmatrix} X_W & Y_W & Z_W \end{bmatrix}^T$, $\mathbf{P}_c = \begin{bmatrix} X_c & Y_c & Z_c \end{bmatrix}^T$, and $\mathbf{P}_{uv} = \begin{bmatrix} u & v & 1 \end{bmatrix}^T$.

Figure 5

Process of projecting a point on a telegraph pole into a pixel coordinate system



To accomplish this process, point $\mathbf{P}_W$ on the pole in the world coordinate system is first projected to the camera coordinate system to obtain $\mathbf{P}_C$ and then transformed to the pixel coordinate system to obtain $\mathbf{P}_{uv}$. The projection of $\mathbf{P}_C$ from the camera coordinate system to the pixel coordinate system involves passing through the normalization plane. This plane, defined as $Z_c = 1$, coincides with the location of the pixel plane. In summary, obtaining $\mathbf{P}_{uv}$, which represents the position of point $\mathbf{P}$ on the pole in the orchard in the image, necessitates multiple coordinate transformations. The sequence of projection is as follows: world to camera, camera to normalization plane, and normalization plane to pixel. For UAV obstacle avoidance, the objective is to use the depth camera to ascertain $\mathbf{P}_W$, the position of point $\mathbf{P}$ on the pole in the world coordinate system, facilitating the construction of an obstacle map. Furthermore, acquiring $\mathbf{P}_W$ involves the back-projection of $\mathbf{P}_{uv}$. First, $\mathbf{P}_{uv}$ is back-projected into the camera coordinate system to obtain $\mathbf{P}_C$:

$$Z_C \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_C \\ Y_C \\ Z_C \end{bmatrix} \triangleq \mathbf{KP}_C, \tag{3}$$

where Z_c represents the distance from point $\mathbf{P}_C$ to the camera, which is equivalent to the distance between the UAV and the utility pole and directly measurable via the depth camera. The internal parameter $\mathbf{K}$ is determined via prior calibration. Subsequently, Equation (3) is transformed as follows:

$$X_C = \frac{(u - c_x) Z_C}{f_x}, \tag{4}$$

$$Y_C = \frac{(v - c_y) Z_C}{f_y}. \tag{5}$$

The coordinates of $\mathbf{P}_C$ are thus solved for, and $\mathbf{P}_W$ is obtained by back projecting $\mathbf{P}_C$ into the world coordinate system:

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X_W \\ Y_W \\ Z_W \\ 1 \end{bmatrix}. \tag{6}$$

Subsequently, Equation (6) is transformed into

$$\begin{bmatrix} X_C \\ Y_C \\ Z_C \end{bmatrix} = \mathbf{R} \begin{bmatrix} X_W \\ Y_W \\ Z_W \end{bmatrix} + \mathbf{t}. \tag{7}$$

Finally, Equation (7) is transformed into

$$\begin{bmatrix} X_W \\ Y_W \\ Z_W \end{bmatrix} = \mathbf{R}^{-1} \begin{bmatrix} X_C \\ Y_C \\ Z_C \end{bmatrix} - \mathbf{t}. \tag{8}$$

As indicated by the above equations, acquiring the external parameter matrix of the depth camera is essential to obtaining $\mathbf{P}_W$. Given that the depth camera is mounted on the UAV, its external reference matrix can be derived from the UAV's sensor data. The UAV's flight controller, equipped with an inertial measurement unit (IMU) sensor, acquires the UAV's attitude information to calculate the rotation matrix. Moreover, the UAV's onboard GPS gathers the position information required for determining the translation vector.

The two-dimensional pixel point coordinates and depth information of the utility pole obstacle in the orchard's depth image can be converted into point cloud coordinates of the utility pole in the global coordinate system. By integrating the UAV's position data with the point cloud coordinates of the utility poles, an orchard map detailing the utility pole obstacles is created. The obstacle map is represented using an occupancy grid map. In an occupancy grid map, space is discretized into cell grids approximating real-world obstacles, with each cell storing the probability of occupancy by an obstacle. A 3D grid map reflecting the current UAV position is constructed in real time using the point cloud data of obstacles in the global coordinate system. In consideration of factors such as navigation accuracy, sensor precision, and computational cost, the cell grid size is set at 0.1 m. A real-time local map updates the global occupancy grid map, reflecting the UAV's current global attitude, with the occupancy probability of each grid updated using Bayesian probability [36]. As the subsystem updates each grid state by incorporating previous data with each new observation, it achieves a high degree of reproducibility in environmental information. This strategy is widely employed in mapping applications utilizing sensors such as depth cameras and LiDAR.

4. Path Planning Subsystem

To enable a UAV to navigate and avoid obstacles, it must not only sense obstacles and generate an obstacle map but also execute path planning from its current location to a designated target point. Consequently, we developed a path planning subsystem primarily designed to facilitate safe UAV travel from point A to point B, ensuring the generation of smooth, dynamic, and collision-free trajectories. Figure 6 shows the three stages involved in trajectory optimization.

During operations in a lemon orchard, a plant protection UAV typically follows a " ∏ "-shaped path. Given the orchard's unstructured layout, unavoidable obstacles such as utility poles between lemon trees (as shown in Figure 2(c)) pose safety risks to the UAV's operation. Consequently, to ensure the UAV's safe operation, generating collision-free trajectories that the UAV can execute is essential. Our proposed path planning algorithm, delineated in Figure 7, encompasses three main phases:

Figure 6

Three stages of trajectory optimization: (a) obstacles in the trajectory, (b) generation of warm-up trajectory, (c) B-spline trajectory optimization

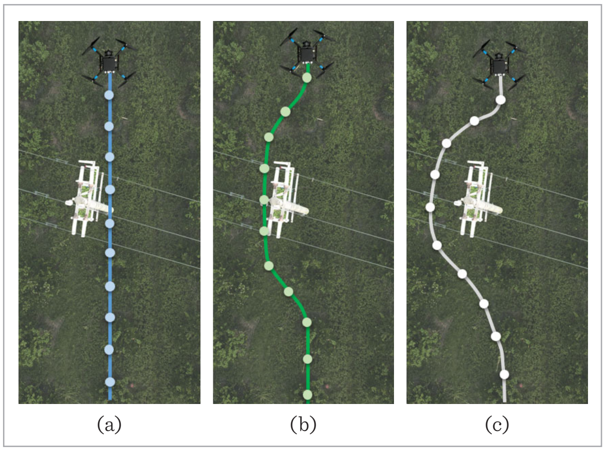
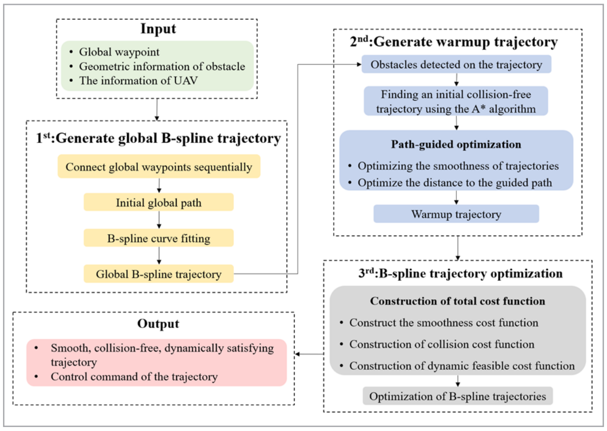


Figure 7

Flowchart of the path planning algorithm


Input
• Global waypoint
• Geometric information of obstacle
• The information of UAV
1st:Generate global B-spline trajectory
Connect global waypoints sequentially
Initial global path
B-spline curve fitting
Global B-spline trajectory
2nd:Generate warmup trajectory
Obstacles detected on the trajectory
Finding an initial collision-free trajectory using the A* algorithm
Path-guided optimization
• Optimizing the smoothness of trajectories
• Optimize the distance to the guided path
Warmup trajectory
3rd:B-spline trajectory optimization
Construction of total cost function
• Construct the smoothness cost function
• Construction of collision cost function
• Construction of dynamic feasible cost function
Optimization of B-spline trajectories
Output
• Smooth, collision-free, dynamically satisfying trajectory
• Control command of the trajectory

a Initially, all path points are sequentially connected to form the global operation path. Taking advantage of the B-spline's property of easy optimization, B-spline curve fitting is applied to the global path points to generate the initial B-spline trajectory, setting the stage for subsequent optimization. At this stage, the global B-spline trajectory may contain obstacles, as obstacle avoidance processing has not yet been implemented.

b The UAV follows a global B-spline trajectory, conducting trajectory feasibility checks using the local map. Upon detecting an obstacle along the intended trajectory, a geometric collision-free path is identified within the local map using the A* algorithm. The global B-spline trajectory intersecting the obstacle is then optimized for path guidance based on this geometric collision-free path. Moreover, a path guidance optimization function is constructed based on the nature of the B-spline control points. The control points of the B-spline trajectory are aligned with the sampling points on the geometric path, which results in the trajectory being optimized into a preliminary trajectory.

c An optimization function is constructed to further enhance the warm-up trajectory. This function is based on the characteristics of the control points, convex envelopes, and derivatives of B-spline curves. The trajectory is tuned to ensure smoothness, safety, and kinetic feasibility. Ultimately, a collision-free trajectory that is both smooth and executable is obtained.

4.1. Path Initialization

Prior to initiating operations, the plant protection UAV must first plan its waypoints. These waypoints are connected to form the global operational path, which initially includes right-angle turns. Such angles can result in considerable changes in the UAV's yaw angle and lead to flight oscillations. To address this issue, preprocessing the global path is necessary. This involves fitting the path with a B-spline curve, which not only smoothens the path but also facilitates subsequent trajectory optimization.

A B-spline is a segmented polynomial curve. Its shape is determined by its order p_b, $n+1$ control points $\{\mathbf{Q}_0, \mathbf{Q}_1, \ldots \mathbf{Q}_n\}$, and node vectors $[t_0, t_1, \ldots, t_M]$, where $\mathbf{Q}_i \in \mathbb{R}^3$, $t_m \in \mathbb{R}$, and $M = n + p_b + 1$. A B-spline is a trajectory parameterized with respect to time t, where $t \in [t_{p_b}, t_{M-p_b}]$. B-spline fitting of the global path involves determining the sequence of global path points, setting the B-spline order, selecting the number of control points, and incorporating these elements into the B-spline fitting formula. The number of control points substantially influences the fitting accuracy of the global path; that is, a larger value of n results in a more precise fit. Nevertheless, an excessively large value of n may lead to a trajectory that lacks smoothness, adversely affecting the flight of the plant protection UAV. In light of the flight stability of the plant protection UAV, the B-spline curve's order p_b is defined as 3. The number of control points is set to $2l$, where l denotes the length of the global path. If the B-spline curve is linear, the distance between each control point is maintained at 0.5 m.

The equation that defines the B-spline curve is as follows:

$$\mathbf{C}(t) = \sum_{i=0}^{n} N_{i,p_b}(t)\mathbf{Q}_i, \tag{9}$$

where $\mathbf{Q}_i$ is the i control point and N_{i,p_b} is the B-spline basis function with the transfer equation:

$$N_{i,0}(t) = \begin{cases} 1 & \text{if } t_i \le t < t_{i+1} \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

$$N_{i,p_b}(t) = \frac{t - t_i}{t_{i+p_b} - t_i} N_{i,p_b-1}(t) + \frac{t_{i+p_b+1} - t}{t_{i+p_b+1} - t_{i+1}} N_{i+1,p_b-1}(t). \tag{11}$$

The B-spline curve can be expressed for each path point $\mathbf{D}_i$ using the following expression:

$$\mathbf{C}(t_i) = \sum_{i=0}^{n} N_{i,p_b}(t_i)\mathbf{Q}_i. \tag{12}$$

$N_{i,p_b}(t_i)$ represents the B-spline basis function value at the i control point for the parameter t_i. The distance error, calculated as $|\mathbf{D}_i - \mathbf{C}(t_i)|$, arises between the path point $\mathbf{D}_i$ and its corresponding point on the curve for t_i. This distance error is squared, and a least squares formula is applied to resolve the issue:

$$f(\mathbf{Q}_0, \ldots, \mathbf{Q}_n) = \sum_{i=0}^{n} |\mathbf{D}_i - \mathbf{C}(t_i)|^2. \tag{13}$$

Ultimately, the global B-spline trajectory is derived by calculating the control points $\mathbf{Q}_0$, ..., $\mathbf{Q}_n$ in a manner that minimizes the function $f(\mathbf{Q}_0, \ldots, \mathbf{Q}_n)$. The global B-spline trajectory possesses the following characteristics:

a Four control points govern a segment of the curve, and adjusting their positions alters the curve's shape. This characteristic is instrumental in the development of the path-guided optimization function.

b The convex hull property ensures that the curve lies within the convex shape formed by the four control points. This characteristic is utilized in optimization to ensure trajectory safety.

c The derivative of a B-spline curve is also a B-spline curve, and this property is leveraged to ensure the dynamic feasibility of the trajectory.

Following the global path fitting, the resulting global B-spline trajectory ensures smooth transitions at turns. Nevertheless, as obstacle information is not yet considered, obstacles may be present on the trajectory, requiring further optimization.

4.2. A* Algorithm

Upon initiating operation, the UAV initially follows the global path, during which the depth proximity sensor acquires obstacle information in front of the UAV and generates a local map. Given that the global path does not account for obstacle information, obstacles present on the path may pose a threat to the UAV's flight safety. Should obstacles be detected on the impending path on the local map, the UAV's current position will serve as the starting point for employing the A* algorithm to identify a collision-free path. The A* algorithm has identified the yellow line in Figure 8(a) as the initial collision-free path.

The strategy of the A* algorithm is to maintain a prioritized queue to store all the nodes to be expanded, and the cost function of each node is defined as follows:

$$f(n) = g(n) + h(n). \tag{14}$$

Here, $g(n)$ is the best estimate of the current cumulative cost from the start state to the node, $h(n)$ is the minimum cost estimated from the node to the end node, and the $h(n)$ heuristic function uses the Euclidean distance.

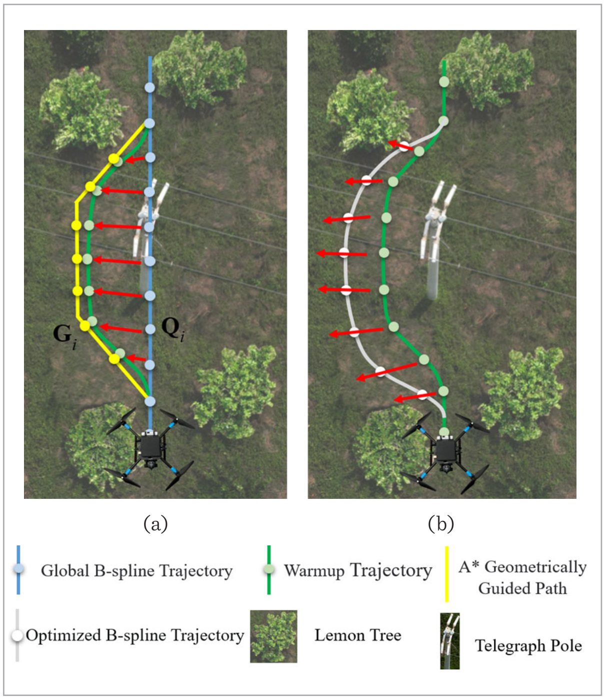
4.3. Path-guided Trajectory Optimization

The trajectory found by A* in Section 4.2, which avoids collisions, is considered a geometric guidance trajectory. This trajectory is used to guide the global B-spline trajectory, pulling it away from obstacles and toward free space, which ensures the initial safety of the trajectory. This is the initial stage of global B-spline trajectory optimization for creating the warm-up trajectory in unoccupied space. The overall cost function for this phase is as follows:

$$f_{total1} = \lambda_{1s} f_s + \lambda_{1d} f_d. \tag{15}$$

Figure 8

Range of the current local map where the utility poles appear on the global path. (a) Use the A* algorithm to find a collision-free path from the current location, and use the collision-free path as a guide to pull the global path away from free space as a warm-up trajectory. (b) The warm-up trajectory is further optimized into a smooth, safe, and dynamically feasible trajectory


G_i
Q_i
(a)
(b)
Global B-spline Trajectory
Warmup Trajectory
A* Geometrically Guided Path
Optimized B-spline Trajectory
Lemon Tree
Telegraph Pole

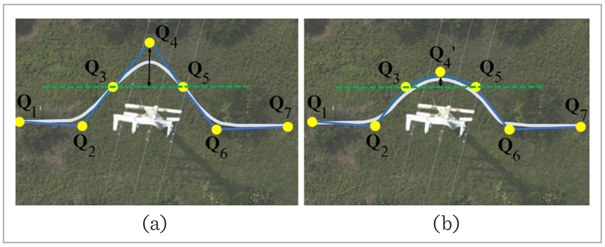
Here, f_s is the trajectory smoothness cost function, and f_d is the distance function between the penalized bootstrap path and the global trajectory B-spline. f_s is defined as follows:

$$f_s = \sum_{i=p_b-1}^{N-p_b+1} dis\left\|\mathbf{Q}_i - line\left(\mathbf{Q}_{i-1}\mathbf{Q}_{i+1}\right)\right\|, \tag{16}$$

where $\mathbf{Q}_{i-1}\mathbf{Q}_{i+1}$ is a straight line that connects the control points $\mathbf{Q}_{i-1}$ and $\mathbf{Q}_{i+1}$; the above equation represents the distance between minimizing $\mathbf{Q}_i$ to the straight line $\mathbf{Q}_{i-1}\mathbf{Q}_{i+1}$. In Figure 9(a), the pre-optimization B-spline trajectory is presented, showing the seven control points $\mathbf{Q}_1$, $\mathbf{Q}_2$, ..., $\mathbf{Q}_7$. Currently, the trajectory exhibits a greater curve at control point $\mathbf{Q}_4$ and lacks smoothness. The spikes in the B-spline trajectory occur because of the substantial distance between $\mathbf{Q}_4$ and the straight $line\left(\mathbf{Q}_3\mathbf{Q}_5\right)$. To provide

Figure 9

(a) There is a large distance between $\mathbf{Q}_4$ and the straight $\mathbf{Q}_3\,\mathbf{Q}_5$. (b) The distance between the optimized $\mathbf{Q}_4'$ and the straight $\mathbf{Q}_3\,\mathbf{Q}_5$ is reduced



a smooth and gentle trajectory, the curve in Figure 9(b) is optimized, which results in the selection of the optimum control point $\mathbf{Q}_4'$. At present, the distance from the straight $line(\mathbf{Q}_3\mathbf{Q}_5)$ is decreased, leading to a reduction in curvature. Consequently, the optimized trajectory exhibits greater smoothness and seamlessness compared to the trajectory prior to optimization.

Given that a B-spline's shape is determined by its control points, steering the global path into free space requires aligning the B-spline's control points as closely as possible with the geometric steering trajectory. Figure 8(a) illustrates the assignment of an associated point $\mathbf{G}_i$ on the bootstrap path to each control point $\mathbf{Q}_i$ on the B-sample. The points $\mathbf{G}_i$ are evenly sampled on the bootstrap path. f_d is defined as the sum of the Euclidean distances between $\mathbf{Q}_i$ and $\mathbf{G}_i$:

$$f_d = \sum_{i=p_b}^{N-p_b} \left\| \mathbf{Q}_i - \mathbf{G}_i \right\|^2. \tag{17}$$

Figure 8(a) illustrates the optimal global B-spline trajectory (blue) achieved by the geometric bootstrap path (yellow). This trajectory results in a warm-up trajectory (green), during which the trajectory moves away from the utility pole barrier toward free space. Nevertheless, this trajectory is not yet optimal and necessitates a secondary stage of optimization.

4.4. B-spline Optimization

During the second stage of optimization, the objective is to further refine the preliminary trajectory into a smooth, safe, and dynamically feasible one. To this end, the overall cost function is formulated according to the control points, convex hull, and derivative properties of the B-spline trajectory:

$$f_{total2} = \lambda_{2s} f_s + \lambda_{2c} f_c + \lambda_{2d} \left(f_v + f_a \right). \tag{18}$$

The cost functions f_s and f_c represent the trajectory smoothness and collision costs, respectively. Moreover, f_v and f_a are cost functions related to UAV velocity and acceleration, respectively, whereas λ_{2s}, λ_{2c}, and λ_{2d} denote the respective weights of these three cost functions. The cost function fs is defined similarly to Equation (14), as optimized in the first stage.

The definition of f_c is as follows:

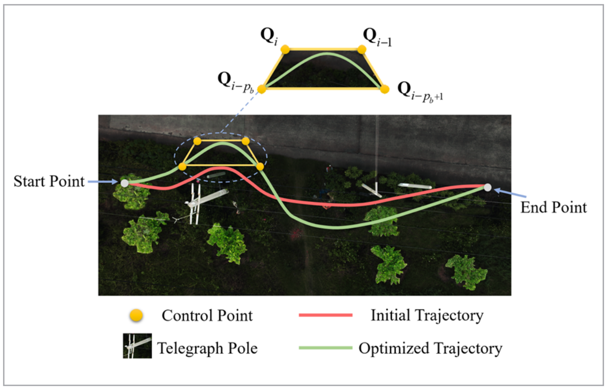
$$f_c = \sum_{i=p}^{N-p} L_c\left(d\left(\mathbf{Q}_i \right) \right), \tag{19}$$

$$L_c\left(d\left(\mathbf{Q}_i \right) \right) = \begin{cases} 2\sqrt{\left(d\left(\mathbf{Q}_i \right) - d_{thr} \right)^2} & d\left(\mathbf{Q}_i \right) \le d_{thr} \\ 0 & d\left(\mathbf{Q}_i \right) > d_{thr} \end{cases}. \tag{20}$$

The variable $d(\mathbf{Q}_i)$ represents the distance between the control point $\mathbf{Q}_i$ of the obstacle avoidance trajectory and the closest orchard utility pole obstacle. $L_c(d(\mathbf{Q}_i))$ is a function that is divided into segments based on $d(\mathbf{Q}_i)$, and d_{thr} is a specified threshold for the minimum distance from the control point to the closest obstacle of an orchard utility pole. If the distance between the control point $\mathbf{Q}_i$ and the utility pole is below the specified threshold, the trajectory is adjusted to maximize the movement away from the utility pole. Figure 10 depicts the initial trajectory in red, which is in close proximity to the utility poles. By contrast, the enhanced track in green is noticeably distant from the

Figure 10

Property of convexity in B-spline packets. The initial trajectory exhibits proximity to the obstacle, whereas the improved trajectory's convex envelope resides outside the impediment, guaranteeing safety



utility pole obstacles. Because of the B-spline's convex packet nature, the convex packet is outside the obstruction, guaranteeing the safety of the trajectory.

Given that the derivative of a B-spline curve remains a B-spline curve, the control points $\mathbf{V}_i$ and $\mathbf{A}_i$ for the derived velocity and acceleration B-spline curves, respectively, can be expressed as follows:

$$\mathbf{V}_i = \frac{\mathbf{Q}_{i+1} - \mathbf{Q}_i}{\Delta t},\ \mathbf{A}_i = \frac{\mathbf{V}_{i+1} - \mathbf{V}_i}{\Delta t}. \tag{21}$$

The definitions of f_v and f_a are as follows:

$$f_v = \sum_{u\in\{x,y,z\}} \sum_{i=p_{b-1}}^{N-p_b} L_v\left(V_{ui}\right), \tag{22}$$

$$f_a = \sum_{u\in\{x,y,z\}} \sum_{i=p_{b-2}}^{N-p_b} L_a\left(A_{ui}\right), \tag{23}$$

$$L_v\left(V_{ui}\right) = \begin{cases} \left(v_u - v_{\max}\right)^2 & v_u > v_{\max} \\ 0 & v_u \le v_{\max} \end{cases}, \tag{24}$$

$$L_a\left(A_{ui}\right) = \begin{cases} \left(a_u - a_{\max}\right)^2 & a_u > a_{\max} \\ 0 & a_u \le a_{\max} \end{cases}; \tag{25}$$

where v_u and a_u represent the velocity and acceleration in a single dimension of the UAV, respectively. $v_{\max}$ and $a_{\max}$, respectively, signify the maximum linear velocity and acceleration in that dimension. Should the UAV's current velocity and acceleration exceed its dynamic limits, these aspects are optimized to remain below these thresholds. Following the second stage of optimization, as shown in Figure 8(b), the global B-spline trajectory diverges from the utility pole, moving toward a safe area in free space, which results in a trajectory that is smooth and meets dynamic requirements.

5. Simulation Experiment

5.1. Simulation Environment and System Construction

To validate the effectiveness of our proposed method, a lemon orchard simulation environment was established using the Gazebo software package, with the simulated UAV controlled via ROS. The simulation environment, depicted in Figure 11, features a blue sky with white clouds, lemon trees, grass, utility poles, and a simulated UAV. The simulation UAV utilizes the Iris UAV model, with its nose equipped with a RealSense depth camera module to capture RGB-D images of the environment. The lemon orchard simulation consists of seven rows of lemon trees, with each row containing varying numbers of trees. The spacing between trees in each row ranges from 2.5 m to 4 m, with a 6-m interval between rows. Four utility poles, serving as obstacles, are strategically positioned in the second, third, and sixth rows. Specifically, to assess the UAV's performance in continuous obstacle avoidance, two poles are consecutively placed among the fruit trees in the sixth row.

Figure 11

A lemon orchard simulation environment was constructed in Gazebo, featuring lemon trees, utility poles, and Iris drones. Subpanels (a)–(c) depict various angles of the simulation environment. Subpanels (d)–(e) show close-up details of the lemon orchard trees and Iris drones

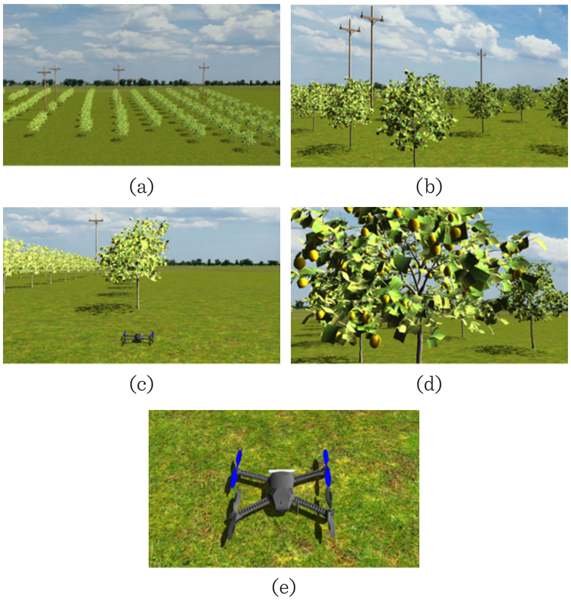
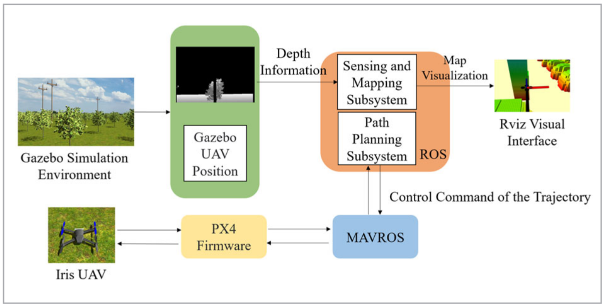


Figure 12 displays the overall block diagram of the simulation system. The sensing and mapping subsystem captures the depth image information within the lemon orchard simulation environment and merges this information with the simulated UAV's

Figure 12

Simulation system framework



position data in Gazebo to create a grid map, which is subsequently visualized in Rviz. The path planning subsystem identifies a collision-free trajectory using the current UAV position and the constructed map, thereby generating control commands for navigating this trajectory. These control commands are relayed to the simulation system's PX4 firmware via MAVROS communication. Finally, the PX4 firmware produces the fundamental control instructions to pilot the simulated UAV, enabling it to perform obstacle avoidance simulation tasks.

5.2. Simulated Flight Experiment

In the lemon orchard simulation environment, the flight path, as illustrated in Figure 13, includes 14 path points. These points are positioned 3.5 m above the center of each column of the head and tail lemon trees and are sequentially connected to form the initial flight path. The UAV is programmed to take off from the starting point, maintaining a flight altitude of 3.5 m and a maximum speed of 1 m/s. The UAV follows the initial flight path, which spans a distance of 306.43 m. The orientation of the fruit tree columns aligns with the X-axis, whereas the rows of fruit trees align with the Y-axis. In this simulation, the Iris UAV is equipped with a depth camera to capture the depth images of the simulated lemon orchard and utilize its positional data to create a grid map of the orchard.

Figure 14 displays various flight states during the execution of the operation path by the simulated UAV. In each subpart, the leftmost image displays the depth and RGB images captured using the UAV from the front. The image in the middle depicts the UAV's flight scenario in the Gazebo simulation. The rightmost image shows the UAV's flight trajectory along the estab-

Figure 13

UAV flight paths in simulated environments

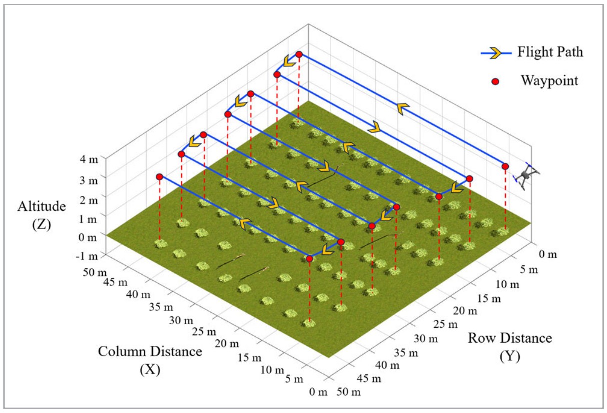
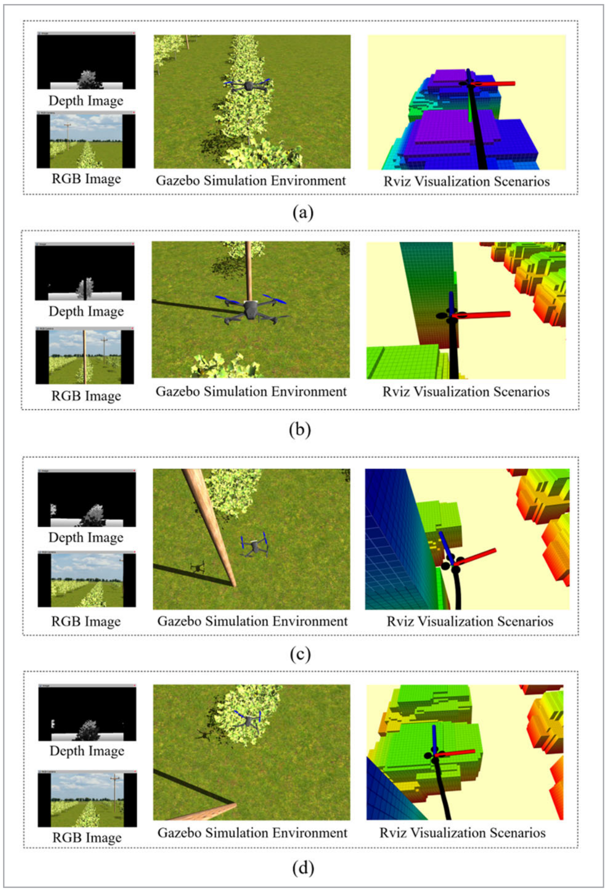


Figure 14

The UAV exhibits four distinct states during the simulated flight: (a) executing an operational path flight; (b) detecting obstacles on the path; (c) navigating an obstacle avoidance path; and (d) successfully completing the obstacle avoidance task


Depth Image
RGB Image
Gazebo Simulation Environment
Rviz Visualization Scenarios
(a)
Depth Image
RGB Image
Gazebo Simulation Environment
Rviz Visualization Scenarios
(b)
Depth Image
RGB Image
Gazebo Simulation Environment
Rviz Visualization Scenarios
(c)
Depth Image
RGB Image
Gazebo Simulation Environment
Rviz Visualization Scenarios
(d)

lished occupancy grid map of the lemon orchard in Rviz. In Figure 14, the UAV follows a predetermined flight path, initially flying above a column of lemon fruit trees, as illustrated in Figure 14(a). Subsequently, obstacles are detected on the path ahead, as depicted in Figure 14(b), leading the UAV to execute the obstacle avoidance maneuver shown in Figure 14(c). This maneuver culminates in the UAV successfully bypassing the obstacles, as shown in Figure 14(d).

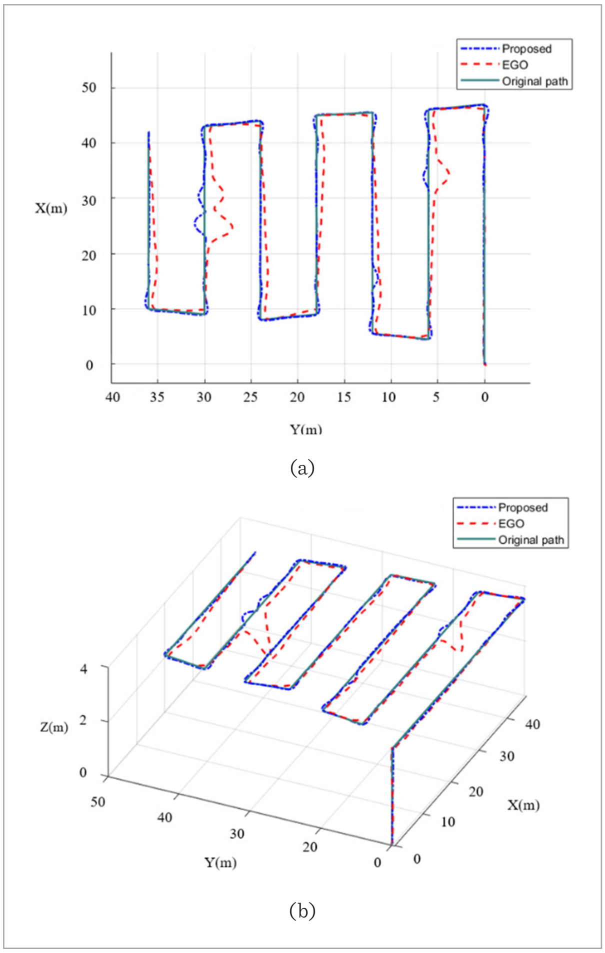
For performance comparison, our proposed algorithm and the ego-planner [44] were tested under identical flight parameters. The simulated UAV was set to fly at an altitude of 3.5 m with a maximum speed of 1 m/s, following a predetermined path. The comparison covered various parameters, including the flight distance, flight time, average deviation from the original path, three-axis linear velocity, and maximum and average yaw angular velocities. Table 1 presents the values of these parameters for both methods during the simulated flight.

Figure 15 shows a comparison of the flight trajectories of the proposed algorithm, the ego-planner, and the originally set path. The proposed algorithm covers a flight distance of 311.36 m, takes 333.54 s, and reaches a maximum speed of 1.279 m/s. By contrast, the ego-planner covers a distance of 302.87 m, requires 352.80 s, and attains a maximum speed of 1.270 m/s. Thus, while adhering to the UAV's maximum dynamic requirements, the proposed algorithm demonstrates higher operational efficiency compared to the ego-planner.

Moreover, as presented in Figure 15(b), the ego-planner's strategy involves a decrease in UAV flight altitude during obstacle avoidance. This algorithm is not viable for plant protection UAV operations in lemon orchards due to its adverse impact on the efficacy of fog droplet deposition. Consequently, the algorithm results in inconsistent deposition on the fruit trees before and after the obstacle [31]. The average deviation of the proposed algorithm's flight trajectory from the original path is 2.77 m, whereas that of the ego-planner is 3.08 m, which indicates a closer fit of the proposed algorithm to the original path.

Figure 15

Comparison of the three trajectories. The proposed method trajectory is shown in blue, the ego-planner trajectory is shown in red, and the original trajectory is shown in green. (a) Trajectory in the X-Y plane. (b) Three-dimensional trajectory



(a)

(b)

Furthermore, Figure 15(a) demonstrates that the proposed algorithm's trajectory deviation when avoiding

Table 1

Comparison of the parameters of the proposed method with the ego-planner in the Gazebo simulation flight

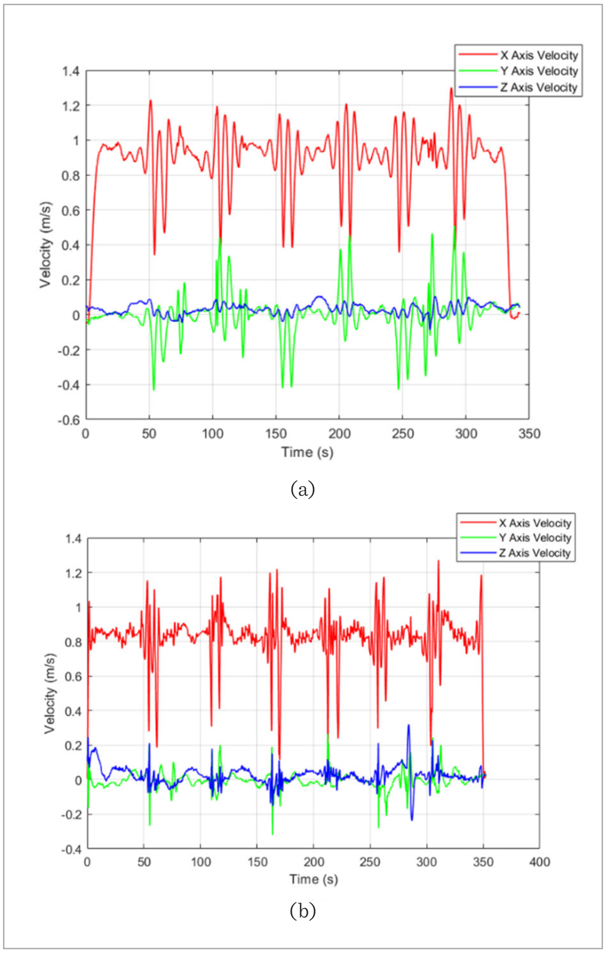
Method	Distance (m)	Time (s)	Average deviation (m)	Max yaw angular velocity (rad/s)	Average yaw angular velocity (rad/s)
Proposed	311.36	333.54	2.77	1.001	0.078
ego	302.87	467.67	3.08	3.348	0.067

obstacles is smaller than that of the ego-planner. In the context of plant protection UAV operations, where the original paths traverse the center of the fruit trees, a better alignment of the flight trajectory with these paths leads to minimal droplet offset. Moreover, this alignment results in reduced pesticide drift to adjacent fruit trees [10].

To analyze flight stability, we examine three key parameters: triaxial linear velocity, maximum yaw angular velocity, and average yaw angular velocity. Figure 16 illustrates the time-based graphs of triaxial velocities for both the proposed algorithm and the ego-planner. The proposed algorithm's velocity change curve is smoother than that of the ego-planner. The ego-planner's velocity change curve exhibits spikes, which indicate considerable variations in UAV speed over brief intervals. In an orchard environment, frequent acceleration and deceleration within brief periods can result in substantial fluctuations in UAV flight. Such fluctuations can lead to increased unnecessary energy consumption and are unfavorable for efficient UAV operations in agriculture [37].

Figure 16

Three-axis velocity profiles in the Gazebo simulation flight. (a) Proposed method. (b) Ego-planner



(a)

(b)

The ego-planner's maximum acceleration reaches 0.9 m/s2, whereas the proposed algorithm's maximum acceleration is limited to 0.27 m/s2. The proposed algorithm exhibits a maximum yaw angular velocity of 1.001 rad/s and an average yaw angular velocity of 0.078 rad/s. By contrast, the ego-planner has a maximum yaw angular velocity of 3.348 rad/s and an average of 0.067 rad/s. The considerably lower maximum yaw angular velocity of the proposed algorithm compared to that of the ego-planner implies that the plant protection UAV will not undergo drastic steering maneuvers. In particular, in orchard environments, excessive yaw angular velocities can result in unstable control, thereby influencing the coverage and uniformity of applications [1].

The analysis of the simulation flight results indicates that the proposed algorithm outperforms the ego-planner in terms of flight operation efficiency, obstacle avoidance effectiveness, and flight stability. Specifically, the proposed algorithm achieves a 28.7% reduction in the flight time, a 10.1% decrease in the deviation from the original path, and a 70.1% reduction in the maximum yaw angular velocity.

6. Flight Experiment

This section details the UAV flight hardware platform, the UAV flight experiment carried out in a lemon orchard, and the subsequent analysis of the experimental results.

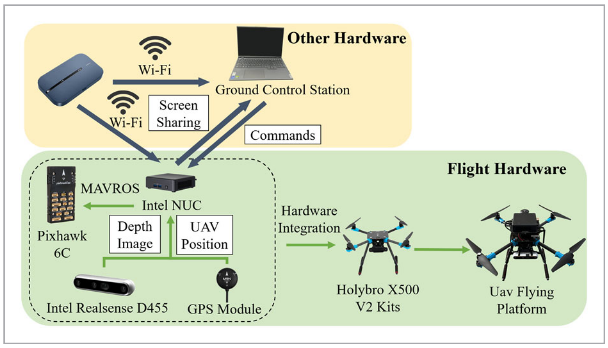
6.1. UAV Flight Hardware Platform

The X500 (v2, Holybro, China) development kit was selected as the drone platform due to its affordability, lightweight design, and robustness as a professional carbon fiber drone kit. The kit includes motors (2216 KV920, Holybro, China), an ESC (BLHeli S, Holybro, China), propellers (1045, Holybro, China), a switchboard (XT30 Pre-soldered, Holybro, China), and a GPS module (M9N, Holybro, China), along with a depth camera mount, all of which facilitate easy development on the platform. The M9N GPS system boasts a measurement accuracy of 1.5 m CEP and a navigation update frequency of 25 Hz. Our research employed a depth camera (Realsense D455, Intel, America) that offers a depth detection range from 0.6 m to 6 m and a depth image output at a frame rate of 90 fps. The depth technology of the Realsense D455, based on stereoscopy, is well-suited for outdoor applications, offering reliable data across varying weather lighting conditions, diverse crop growth stages, and obstacle densities [8]. The algorithms operate on an onboard computer (NUC11TNKv7, Intel, America), powered by an Intel 11th i7 processor. Its compact size, lightweight, and high performance make it ideal for UAV integration. Equipping a UAV with GPS, depth cameras, and on-board computers incurs a cost of $950, a figure that represents a cost-effective deployment relative to obstacle avoidance solutions utilizing LIDAR sensors. This setup, suitable for agricultural environments, seamlessly integrates into existing agricultural UAV platforms. In terms of power supply, the selected battery (4 s-5300 mAh, Gens, China) offers approximately 10 minutes of UAV endurance.

Figure 17 illustrates our comprehensive hardware system, wherein the depth camera and GPS module transmit the depth images and position information of the UAV to the onboard computer. The sensing and mapping subsystem and the path planning subsystem operating on the onboard computer process this information to construct an environmental map, develop an obstacle avoidance trajectory, and relay flight control commands to the flight controller via MAVROS communication. A laptop computer, designated as the ground station, monitors the onboard computer's operations and controls it via remote control software. The ground station communicates with the onboard computer via the same Wi-Fi network.

Figure 17

Comprehensive hardware system

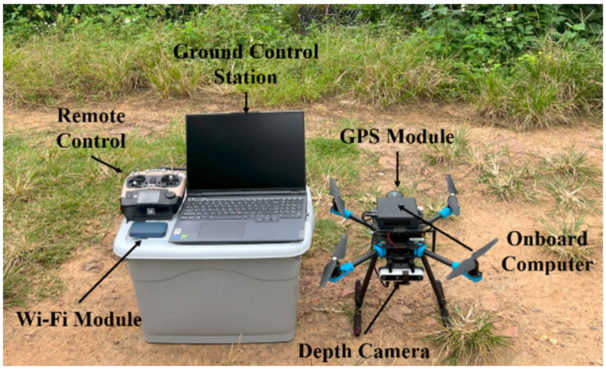


6.2. Real-flight Experiment

Figure 18 presents the setup of the UAV flight hardware platform at the experimental site. This setup includes the integrated UAV flight platform, a laptop ground station, a mobile Wi-Fi module (3Pro, Huawei, China), and a remote control (AT9S Pro, RadioLink, China). Prior to obtaining the path point location information, preliminary data collection at the site is required. This process involved measuring the distances between lemon tree columns, the distances between rows, and the height of the lemon trees in the experimental area, as shown in Figure 19.

Figure 18

Laboratory equipment in the lemon orchard experiment



As illustrated in Figure 20, the flight path for the lemon orchard experiment involves 10 path points, akin to the Gazebo simulation environment. These points are positioned approximately 3 m above the center of the lemon trees at both ends of each row, linked sequentially to form the initial flight path. The experiment was conducted on September 18, 2023, under clear weather conditions, with the UAV flying at an altitude of 3 m. The UAV initiated its flight from the starting point, following the predefined initial path of 224.33 m. The direction of the fruit tree rows aligns with the X-axis, whereas the columns align with the Y-axis. Equipped with a depth camera, the UAV captures the depth images of the lemon orchard and, combined with its own localization data, generates a grid map of the area.

Figure 19

Process of gathering information about the lemon orchard to determine path point locations. (a) Measuring the distance between columns of the lemon trees. (b) Measuring the distance between rows of the lemon trees. (c) Assessing the height of the lemon trees

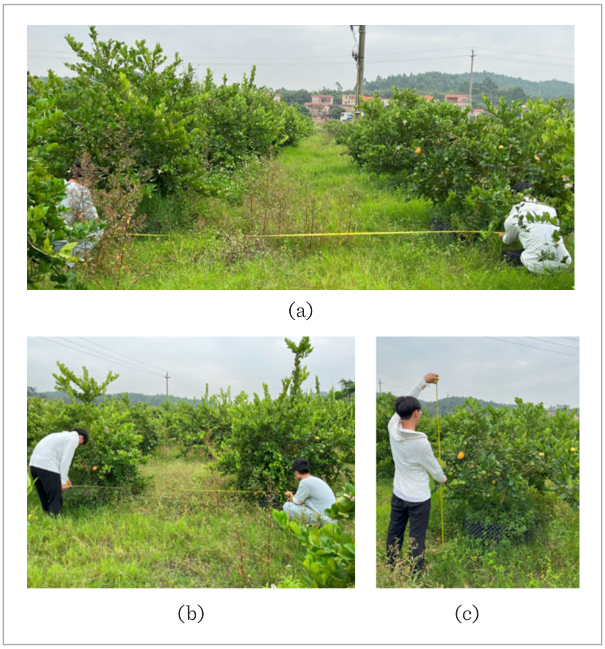


Figure 20

Three-dimensional path of plant protection drones operating in the orchard; there are electric poles on the operation path, which pose a security threat to the drones

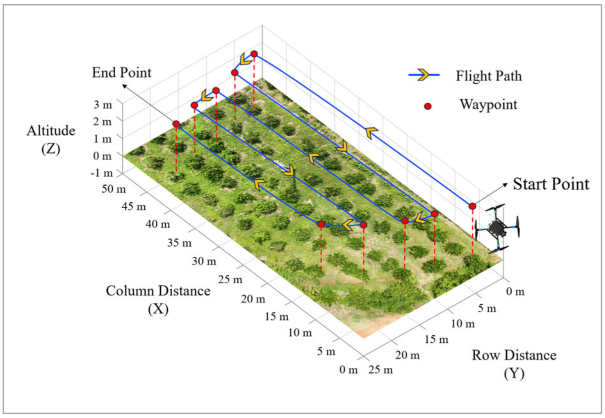
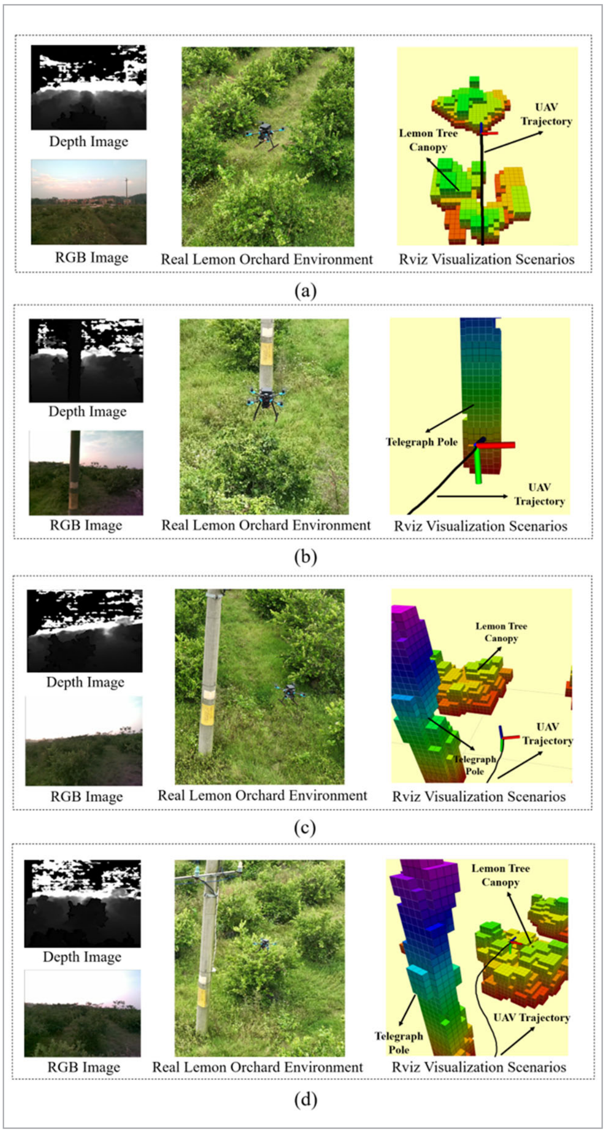


Figure 21 depicts the multiple flight states of the UAV as it carries out the operational path. In each subpart, the leftmost image shows the real-time depth and RGB images captured using the depth camera. The image in the middle shows UAV's flight over the lemon orchard. The rightmost image depicts the UAV's flight trajectory along the established occupancy grid map of the lemon orchard in Rviz. Similar to the simulation flight of UAVs, the experimental UAV deployed in the lemon orchard must undergo four distinct flight states while conducting the predetermined flight tra-

Figure 21

There are four distinct flight states of the UAV in the lemon orchard experiment: (a) executing an operational path flight; (b) detecting obstacles on the path; (c) navigating an obstacle avoidance path; (d) successfully completing the obstacle avoidance task



jectory. As shown in Figure 21(a), the UAV flies above the lemon tree canopy. Figure 21(b) illustrates the detection of a utility pole obstacle ahead, signaling the UAV's readiness to initiate obstacle avoidance. In Figure 21(c), the UAV commences its obstacle avoidance maneuver around the utility pole. Figure 21(d) illustrates the UAV completing the obstacle avoidance maneuver, successfully bypassing the utility pole. Thus, as demonstrated in Figure 21, the proposed method ensures the UAV's safe flight in the actual environment of a lemon orchard.

To evaluate the performance of our proposed method, the ego-planner algorithm was executed using identical flight parameters. The comparison encompasses several aspects: flight distance, flight time, average deviation from the original path, three-axis linear velocity, maximum yaw angular velocity, and average yaw angular velocity. Table 2 details the values of these parameters for both methods in the real lemon orchard flight experiment.

In the real experiment, the proposed algorithm covered a distance of 226.53 m, consumed 244.44 s of flight time, and reached a maximum speed of 1.29 m/s. The ego-planner, conversely, flew 220.90 m, took 257.13 s, and achieved a maximum speed of 1.17 m/s. Therefore, the proposed algorithm, while adhering to the maximum kinetic speed constraints, operates more efficiently in the lemon orchard environment than the ego-planner.

Figure 22 presents a comparison of the flight trajectories in the real lemon orchard experiment, including those of the proposed algorithm, the ego-planner, and the originally set path. Figure 22(b) illustrates the ego-planner engaged in obstacle avoidance, where the UAV exhibits the same drop in altitude observed during the simulation. The average deviation of the proposed algorithm's flight trajectory from the original path is 2.90 m. By contrast, the ego-planner's average deviation is 3.18 m, indicating that the pro-

Figure 22

Comparison of the three trajectories in the lemon orchard flight experiment. The proposed method trajectory is shown in blue, the ego-planner trajectory is shown in red, and the original trajectory is shown in green. (a) Trajectories in the X-Y plane. (b) Three-dimensional trajectories

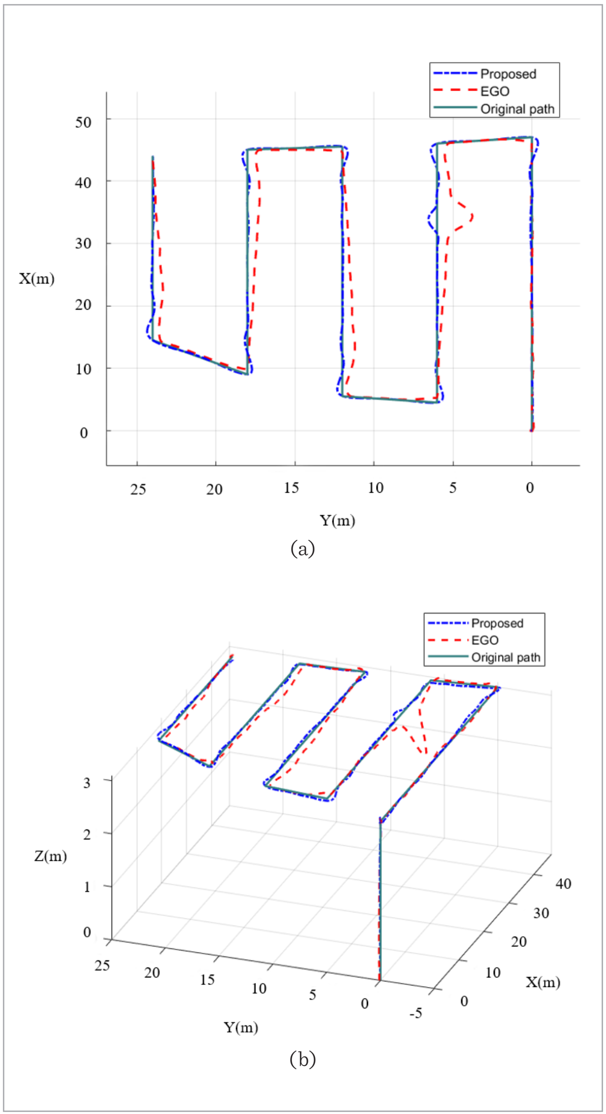


Table 2

Comparison of the parameters of the proposed method with the ego-planner in the real lemon environment flight experiment

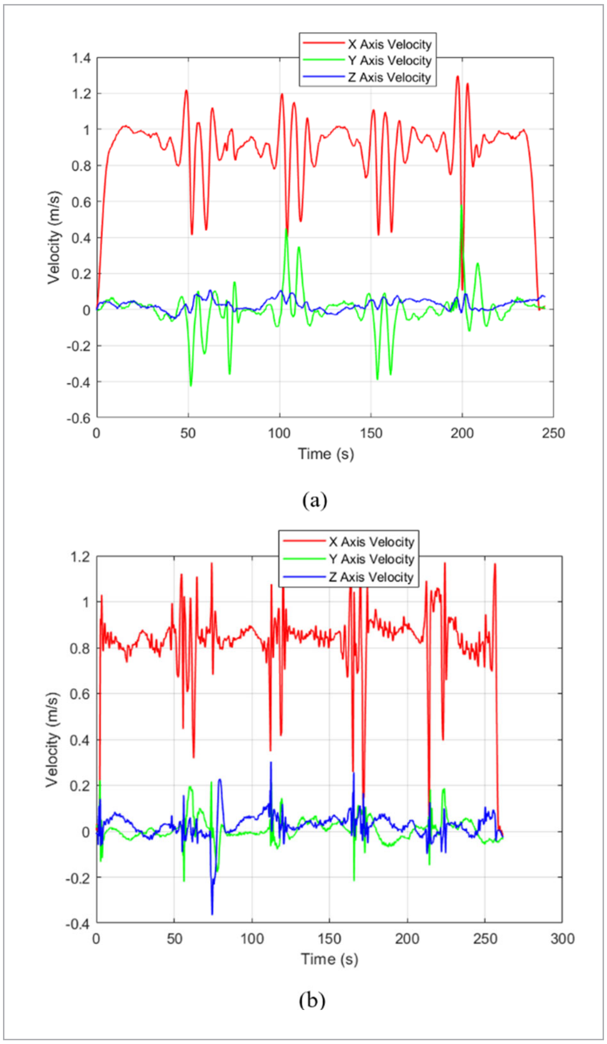
Method	Distance (m)	Time (s)	Average deviation (m)	Max yaw angular velocity (rad/s)	Average yaw angular velocity (rad/s)
Proposed	226.53	244.44	2.90	1.545	0.068
ego	220.90	338.55	3.18	3.523	0.059

posed algorithm has a more precise alignment with the original path in the real lemon orchard environment. Similarly, as in the simulated flight, Figure 22(a) shows that the proposed algorithm's trajectory deviation when avoiding obstacles is less than that of the ego-planner.

Figure 23 displays the three-axis linear velocity versus time graphs for both the proposed algorithm and the ego-planner in the real lemon orchard experiment. These graphs illustrate that the proposed algorithm features smoother velocity change curves in comparison to those of the ego-planner. The ego-planner's velocity change curves exhibit spikes, with a maximum acceleration of 0.9 m/s^2, which indicates drastic velocity changes over short intervals. This results in additional energy consumption during the flight of heavily loaded plant protection UAVs. The proposed algorithm's maximum acceleration is limited to 0.27 m/s^2, and its velocity change is smooth, aligning with the flight requirements of plant protection UAVs. Furthermore, the proposed algorithm exhibits a maximum yaw angular velocity of 1.545 rad/s and an average yaw angular velocity of 0.068 rad/s. Conversely, the ego-planner's maximum yaw angular velocity reaches 3.523 rad/s, with an average of 0.067 rad/s. Although the proposed algorithm's average yaw angular velocity is marginally higher than the ego-planner's, its maximum yaw angular velocity is substantially lower. This aligns with the simulation results, indicating that the proposed algorithm ensures less intense steering in a real lemon orchard environment. As a result, the smoother steering contributes to the flight stability of heavily loaded plant protection UAVs.

Figure 23

Three-axis velocity profiles of the flight experiment at the lemon orchard. (a) Proposed method. (b) Ego-planner



(a)

(b)

The analysis of the lemon orchard flight results indicates that the proposed method outperforms the ego-planner in terms of flight operational efficiency, obstacle avoidance effectiveness, and flight stability in a real orchard environment. Specifically, the proposed algorithm achieves a 27.8% reduction in the flight time, an 8.8% decrease in the deviation from the original path, and a 56.1% reduction in the maximum yaw angular velocity. The results from the lemon orchard flight experiment show slight variances when compared with the simulation flight experiment. Overall, the proposed method demonstrates satisfactory performance for the safe operation of plant protection UAVs in actual lemon orchard scenarios.

7. Conclusion

This study proposed a novel UAV obstacle avoidance system specifically designed for lemon orchards. The system comprises two main components: a sensing and mapping subsystem utilizing a depth image inverse projection algorithm and a path planning subsystem based on B-spline curve trajectory optimization. To evaluate the performance of the proposed system within a lemon orchard environment, a simulated lemon orchard environment was constructed, and corresponding simulation experiments were conducted. The simulation experiments demonstrate that the average deviation of the flight trajectory from the original path is 2.77 m. Moreover, the maximum yaw angular velocity recorded is 1.001 rad/s, with an average yaw angular velocity of 0.078 rad/s. Furthermore, the obstacle avoidance system was integrated into the UAV, and a UAV flying platform was constructed to conduct on-site flight experiments in an actual lemon grove. The field experiments demonstrate that the flight trajectory deviates from the original course by an average of 2.90 m. The maximum yaw angular velocity is 1.545 rad/s, whereas the average yaw angular velocity is 0.068 rad/s, proving its effectiveness.

Both simulation and field experiments demonstrate that our proposed algorithm equips plant protection UAVs with the capability to effectively navigate obstacles in unstructured lemon orchards, ensuring a stable flight process and the safe operation of the UAVs. This research offers a technical solution for the autonomous flight of planting UAVs in precision agriculture, particularly in lemon orchards. It enhances the intelligent operational capabilities of these UAVs and diminishes the risk of crashes, thereby safeguarding the economic interests of orchard farmers. Consequently, this study contributes to the broader adoption of UAVs in precision agriculture.

The hardware and algorithms implemented in this system are adaptable to other similar orchard environments, like apple orchards and vineyards, requiring minimal modifications. While this system employs GPS for positioning, scaling it to diverse agricultural scenarios, such as greenhouse and under-canopy flights, may encounter signal occlusion challenges. In our future work, we aim to integrate visual inertial odometry (VIO) algorithms to enhance the UAV's positional accuracy and bolster the system's robustness. Additionally, this advancement is expected to improve the cost-benefit ratio of the obstacle avoidance system.

Acknowledgement

This work was supported by the National Natural Science Foundation of China under Grant 32271985, the Basic and Applied Basic Research Foundation of Guangdong Province, China under Grant 2022A1515011008, the Scientific and Technological Planning Project of Guangzhou City, China under Grant 2024B03J1268 and 2024E04J0232, and the China Agriculture Research System, China under Grant CARS-15-22.

References

1. Ahmed, S., Qiu, B., Kong, C.-W., Xin, H., Ahmad, F., Lin, J. A Data-Driven Dynamic Obstacle Avoidance Method for Liquid-Carrying Plant Protection UAVs. Agronomy, 2022, 12(4), 873. https://doi.org/10.3390/agronomy12040873
2. Ait-Saadi, A., Meraihi, Y., Soukane, A., Ramdane-Cherif, A., Benmessaoud Gabis, A. A Novel Hybrid Chaotic Aquila Optimization Algorithm with Simulated Annealing for Unmanned Aerial Vehicles Path Planning. Computers and Electrical Engineering, 2022, 104, 108461. https://doi.org/10.1016/j.compeleceng.2022.108461
3. Atakan, E., Pehlivan, S., Achiri, T. D. Pest Status of the Hawaiian Flower Thrips, Thrips Hawaiiensis (Morgan) (Thysanoptera: Thripidae) in Lemons in the Mediterranean Region of Turkey. Phytoparasitica, 2021, 49(4), 513-525. https://doi.org/10.1007/s12600-021-00899-8
4. Bounini, F., Gingras, D., Pollart, H., Gruyer, D. Modified Artificial Potential Field Method for Online Path Planning Applications. 2017 IEEE Intelligent Vehicles Symposium (IV), 2017, 180-185. https://doi.org/10.1109/IVS.2017.7995717
5. Castro, G. G. R. d., Berger, G. S., Cantieri, A., Teixeira, M., Lima, J., Pereira, A. I., Pinto, M. F. Adaptive Path Planning for Fusing Rapidly Exploring Random Trees and Deep Reinforcement Learning in an Agriculture Dynamic Environment UAVs. 2023, 13(2), 354. https://doi.org/10.3390/agriculture13020354
6. 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). International Journal of Agricultural and Biological Engineering, 2021, 14(1), 38-49. https://doi.org/10.25165/j.ijabe.20211401.5714
7. Chitsaz, H., LaValle, S. M. Time-optimal Paths for a Dubins Airplane. 2007 46th IEEE conference on decision and control, 2007, 2379-2384. https://doi.org/10.1109/CDC.2007.4434966

8. Condotta, I. C. F. S., Brown-Brandl, T. M., Pitla, S. K., Stinn, J. P., Silva-Miranda, K. O. Evaluation of Low-cost Depth Cameras for Agricultural Applications. Computers and Electronics in Agriculture, 2020, 173, 105394. https://doi.org/10.1016/j.compag.2020.105394

9. Edlerman, E., Linker, R. Autonomous Multi-robot System for Use in Vineyards and Orchards. 2019 27th Mediterranean Conference on Control and Automation (MED), 2019, 274-279. https://doi.org/10.1109/MED.2019.8798538

10. 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. Computers and Electronics in Agriculture, 2017, 138, 210-223. https://doi.org/10.1016/j.compag.2017.04.011

11. Gammell, J. D., Srinivasa, S. S., Barfoot, T. D. Informed RRT*: Optimal Sampling-based Path Planning Focused Via Direct Sampling of an Admissible Ellipsoidal Heuristic. 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2014, 2997-3004. https://doi.org/10.1109/IROS.2014.6942976

12. Goane, L., Casmuz, A., Salas, H., Lizondo, M., Gastaminza, G., Vera, M. T. Spatial and Temporal Variation in Chaetanaphothrips orchidii Moulton (Thysanoptera: Thripidae) Population and Its Damage on Lemon. Neotropical Entomology, 2013, 42(1), 72-81. https://doi.org/10.1007/s13744-012-0081-5

13. Guo, J., Xia, W., Hu, X., Ma, H. Feedback RRT* Algorithm for UAV Path Planning in a Hostile Environment. Computers & Industrial Engineering, 2022, 174, 108771. https://doi.org/10.1016/j.cie.2022.108771

14. Hart, P. E., Nilsson, N. J., Raphael, B. A Formal Basis for the Heuristic Determination of Minimum Cost Paths. IEEE Transactions on Systems Science and Cybernetics, 1968, 4(2), 100-107. https://doi.org/10.1109/TSSC.1968.300136

15. He, X., Zhang, H., Li, J., Yang, F., Dai, W., Xiang, C., Zhang, M. The Positive Effects of Humic/Fulvic Acid Fertilizers on the Quality of Lemon Fruits. Agronomy, 2022, 12(8), 1919. https://doi.org/10.3390/agronomy12081919

16. Huang, X., Dong, X., Ma, J., Liu, K., Ahmed, S., Lin, J., Qiu, B. The Improved A* Obstacle Avoidance Algorithm for the Plant Protection UAV with Millimeter Wave Radar and Monocular Camera Data Fusion. Remote Sensing, 2021, 13(17), 3364. https://doi.org/10.3390/rs13173364

17. Karaman, S., Frazzoli, E. Sampling-based Algorithms for Optimal Motion Planning. The International Journal of Robotics Research, 2011, 30(7), 846-894. https://doi.org/10.1177/0278364911406761

18. Khokhar, I., Chen, J., Wang, J., Jia, Y., Yan, Y., Mukhtar, I. First Report of Postharvest Blue Mold Decay Caused by Penicillium Expansum on Lemon (Citrus limon) Fruit in China. Plant Disease, 2021, 105(11), 3747. https://doi.org/10.1094/PDIS-10-20-2239-PDN

19. Kose, O., Oktay, T., Özen, E. Simultaneous Arm Morphing Quadcopter and Autonomous Flight System Design. Aircraft Engineering and Aerospace Technology, 2023, 95(10), 1624-1632. https://doi.org/10.1108/AEAT-05-2023-0146

20. Li, J., Zhang, W., Hu, Y., Fu, S., Liao, C., Yu, W. RJA-Star Algorithm for UAV Path Planning Based on Improved R5DOS Model. Applied Sciences, 2023, 13(2), 1105. https://doi.org/10.3390/app13021105

21. Lin, Y., Saripalli, S. Path Planning Using 3D Dubins Curve for Unmanned Aerial Vehicles. 2014 International Conference on Unmanned Aircraft Systems (ICUAS), 2014, 296-304. https://doi.org/10.1109/ICUAS.2014.6842268

22. Liu, Y., Liu, H., Xu, X., Zou, Y., Zhang, Y., Wang, R. Potential Geographical Distribution and Ecological Suitability of Lemon in Sichuan Based on MaxEnt. Polish Journal of Environmental Studies, 2022, 31(3), 2197-2212. https://doi.org/10.15244/pjoes/143117

23. Mammarella, M., Comba, L., Biglia, A., Dabbene, F., Gay, P. Cooperation of Unmanned Systems for Agricultural Applications: A Case Study in a Vineyard. Biosystems Engineering, 2022, 223, 81-102. https://doi.org/10.1016/j.biosystemseng.2021.12.010

24. Mandloi, D., Arya, R., Verma, A. K. Unmanned Aerial Vehicle Path Planning Based on A* Algorithm and Its Variants in 3d Environment. International Journal of System Assurance Engineering and Management, 2021, 12, 990-1000. https://doi.org/10.1007/s13198-021-01186-9

25. Mannar, S., Thummalapeta, M., Saksena, S. K., Omkar, S. N. Vision-based Control for Aerial Obstacle Avoidance in Forest Environments. IFAC-PapersOnLine, 2018, 51(1), 480-485. https://doi.org/10.1016/j.ifacol.2018.05.081

26. Meng, L., Wang, Y., Wei, W.-H., Zhang, H. Population Genetic Structure of Diaphorina Citri Kuwayama (Hemiptera: Liviidae): Host-driven Genetic Differentiation in China. Scientific Reports, 2018, 8(1), 1473. https://doi.org/10.1038/s41598-018-19533-5

27. Mousa, M. H., Hussein, M. K. Efficient UAV-based Mobile Edge Computing Using Differential Evolution and

Ant Colony Optimization. Computer Science, 2022, 8, e870. https://doi.org/10.7717/peerj-cs.870

28. Özen, E., Oktay, T. Optimizing the Performance of an Unmanned Aerial Vehicle that Can Detect and Avoid Obstacles. 9th International Zeugma Conference on Scientific Research, 2023, 1-5.

29. Pei, Y., He, C., Liu, H., Shen, G., Feng, J. Compositional Analysis of Four Kinds of Citrus Fruits with an NMR-Based Method for Understanding Nutritional Value and Rational Utilization: From Pericarp to Juice. Molecules, 2022, 27(8), 2579. https://doi.org/10.3390/molecules27082579

30. Radoglou-Grammatikis, P., Sarigiannidis, P., Lagkas, T., Moscholios, I. A Compilation of UAV Applications for Precision Agriculture. Computer Networks, 2020, 172. https://doi.org/10.1016/j.comnet.2020.107148

31. Richardson, B., Rolando, C. A., Somchit, C., Dunker, C., Strand, T. M., Kimberley, M. O. Swath Pattern Analysis from a Multi-rotor Unmanned Aerial Vehicle Configured for Pesticide Application. Pest Management Science, 2020, 76(4), 1282-1290. https://doi.org/10.1002/ps.5638

32. Stefas, N., Bayram, H., Isler, V. Vision-based Monitoring of Orchards with UAVs. Computers and Electronics in Agriculture, 2019, 163, 104814. https://doi.org/10.1016/j.compag.2019.05.023

33. Sun, L., Xu, J., Nasrullah, Wang, L., Nie, Z., Huang, X., Sun, J., Ke, F. Comprehensive Studies of Biological Characteristics, Phytochemical Profiling, and Antioxidant Activities of Two Local Citrus Varieties in China. Frontiers in Nutrition, 2023, 10. https://doi.org/10.3389/fnut.2023.1103041

34. Sun, Q., Zhang, R., Chen, L., Zhang, L., Zhang, H., Zhao, C. Semantic Segmentation and Path Planning for Orchards Based on UAV Images. Computers and Electronics in Agriculture, 2022, 200, 107222. https://doi.org/10.1016/j.compag.2022.107222

35. Sun, Y., Chen, W., Lv, J. Uav Path Planning Based on Improved Artificial Potential Field Method. 2022 International Conference on Computer Network, Electronic and Automation (ICCNEA), 2022, 95-100. https://doi.org/10.1109/ICCNEA57056.2022.00031

36. Thrun, S. Learning Occupancy Grid Maps with Forward Sensor Models. Autonomous Robots, 2003, 15, 111-127. https://doi.org/10.1023/A:1025584807625

37. Tian, H., Mo, Z., Ma, C., Xiao, J., Jia, R., Lan, Y., Zhang, Y. Design and Validation of a Multi-objective Waypoint Planning Algorithm for UAV Spraying in Orchards Based on Improved Ant Colony Algorithm. Frontiers in Plant Science, 2023, 14, 1101828. https://doi.org/10.3389/fpls.2023.1101828

38. Wang, D., Li, W., Liu, X., Li, N., Zhang, C. UAV Environmental Perception and Autonomous Obstacle Avoidance: A Deep Learning and Depth Camera Combined Solution. Computers and Electronics in Agriculture, 2020, 175, 105523. https://doi.org/10.1016/j.compag.2020.105523

39. Wang, J., Li, Y., Li, R., Chen, H., Chu, K. Trajectory Planning for UAV Navigation in Dynamic Environments with Matrix Alignment Dijkstra. Soft Computing, 2022, 26(22), 12599-12610. https://doi.org/10.1007/s00500-022-07224-3

40. Yingkun, Z. Flight Path Planning of Agriculture UAV Based on Improved Artificial Potential Field Method. 2018 Chinese Control and Decision Conference (CCDC), 2018, 1526-1530. https://doi.org/10.1109/CCDC.2018.8407369

41. Yue, X., Zhang, W. UAV Path Planning Based on k-means Algorithm and Simulated Annealing Algorithm. 2018 37th Chinese Control Conference (CCC), 2018, 2290-2295. https://doi.org/10.23919/ChiCC.2018.8483993

42. Zhang, C., Zhen, Z., Wang, D., Li, M. UAV Path Planning Method Based on Ant Colony Optimization. 2010 Chinese Control and Decision Conference, 2010, 3790-3792. https://doi.org/10.1109/CCDC.2010.5498477

43. Zhang, X., Fan, C., Cao, Z., Fang, J., Jia, Y. Novel Obstacle-avoiding Path Planning for Crop Protection UAV Using Optimized Dubins Curve. International Journal of Agricultural and Biological Engineering, 2020, 13(4), 172-177. https://doi.org/10.25165/j.ijabe.20201304.3205

44. Zhou, X., Wang, Z., Ye, H., Xu, C., Gao, F. Ego-planner: An esdf-free gradient-based Local Planner for Quadrotors. IEEE Robotics and Automation Letters, 2020, 6(2), 478-485. https://doi.org/10.1109/LRA.2020.3047728