IEEE Access
Multidisciplinary Rapid Review Open Access Journal

Received 27 November 2023, accepted 15 January 2024, date of publication 17 January 2024,
date of current version 25 January 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3355275

RESEARCH ARTICLE

An Improved Artificial Potential Field UAV Path Planning Algorithm Guided by RRT Under Environment-Aware Modeling: Theory and Simulation

JILONG LIU[1], **YUEHAO YAN**[2], **YUNHONG YANG**[1], **AND JUNLIN LI**[1]
[1]School of Automation and Information Engineering, Sichuan University of Science and Engineering (SUSE), Yibin 644000, China
[2]Artificial Intelligence Key Laboratory of Sichuan Province, Yibin 644000, China

Corresponding author: Yuehao Yan (yanyuehao@126.com)

This work was supported in part by the Innovation Fund of Chinese Universities under Grant 2021ZYA09002.

ABSTRACT Unmanned Aerial Vehicles (UAVs) have been extensively researched and used in civil and military applications due to their effectiveness and flexibility. However, when identifying obstacles and avoiding them, most of the existing path planning methods fail to accurately perceive the environment, such as without considering the differences between obstacles, which leads to low timeliness and easy fall into a local minimum. In this work, an improved artificial potential field UAV path planning algorithm (G-APF) guided by the rapidly-exploring random tree (RRT) based on an environment-aware model is designed to overcome the limitations of traditional methods. The model can perceive different objects in the environment through the addition of supervised environment modeling to traditional unsupervised path planning. Specifically, an environment-aware model based on YOLOv8 is used to establish the UAV flight environment model, and an adaptive optimal threat distance calculation module is used to construct the repulsive potential field. Secondly, to improve the timeliness of path planning and the global awareness of the model, we first use the G-APF algorithm to plan the rough flight path based on the UAV flight environment. Then, the initially generated trajectory is replanned by building an attractive potential field and combining it with a repulsive potential field. Finally, the problems of local minimum and target unreachability and local trajectory oscillation generated by the artificial potential field (APF) algorithm are solved by G-APF. Experiments with generated regions are performed to demonstrate the efficiency and effectiveness of the proposed approach.

INDEX TERMS UAV, path planning, object detection, perception, supervised learning, modeling.

I. INTRODUCTION

In recent years, unmanned aerial vehicles (UAVs) have seen significant advancements due to the rapid development of autonomy technology, automatic control, and artificial intelligence [1], [2], [3]. The applications of UAVs include civilian or scientific deployments such as agricultural plant protection [4], [5], target tracking [6], [7], traffic monitoring [8], [9], as well as military utilizations such as disaster relief [10], [11], anti-terror campaign [12]. Ensuring accurate obstacle avoidance and stable path planning is pivotal for UAV mission success. UAVs must navigate around obstacles in the field while maintaining stability and finding optimal paths.

Thus, precise obstacle avoidance tailored to different objects is crucial for successful and effective task completion. Moreover, in complex environments, path planning using local planning algorithms fails to provide the UAV with global information, which easily leads to path planning failure. Combining global and local planning approaches

The associate editor coordinating the review of this manuscript and approving it for publication was Mark Kok Yew Ng.

TABLE 1. The relevant studies of apf.

Methods	Main features
Improved APF [27]	Defining threat factors by considering the relative speed and improving the potential model with new threat factors.
Improved APF [28]	The method designs a new APF shape based on a potential density that is inversely proportional to the Nth power of the distance. A new path planning strategy called the "quasi-geodesic method" is proposed to avoid obstacles.
Improved APF [29]	An APF path planning method for moving around the nearest obstacle is proposed to avoid local minimum traps while solving the problem of unreachable targets by a parallel search algorithm.
PSO-APF [30]	The length and smoothness of the global path are used as a fitness function for Particle Swarm Optimization (PSO) to obtain the obstacle impact area, the gravitational coefficient, and the repulsive force in the APF. At the local path planning level, a fuzzy control scheme based on DWA is used to assess the hazard level of moving obstacles using the collision risk index and relative distance.
SA-APF [31]	The APF method is extended to 3D space and combined with the UAV kinematics to constrain the step size and generate a path that conforms to the UAV kinematics. By designing a two-layer repulsive potential field and establishing collision prediction and obstacle avoidance mechanisms, the shortcomings of the traditional APF are overcome.
ACO-APF [32]	An improved ant colony optimization (ACO) mechanism is used to search for the globally optimal path of the USV from the origin to the destination in the grid environment, and subsequently, an improved APF algorithm is used to avoid unknown obstacles in the USV navigation process.
A*-APF [33]	The method increases the repulsive force between UAVs based on the optimized obstacle repulsion, sets up a temporary virtual target point, and uses the A* algorithm to make UAVs pass the obstacle and move to the target point smoothly.
JPS-APF [34]	The method establishes an APF and a directional map, representing the resultant force distribution and node directionality of the target node, respectively. Then, considering the influence of the APF and the guidance of the directional map, the extended directional priority of the path planning is calculated to guide and improve the search of subsequent jump points.

enhances work efficiency, accelerates target task completion, reduces system consumption, and bolsters UAV flight robustness [13]. Algorithm fusion planning methods will be the trend for current and future UAV path planning [14], [15], [16], [17].

To achieve efficient and stable path planning, various algorithms have been proposed, which can be divided into two groups [18]: Heuristic algorithms and deterministic algorithms.

Heuristic algorithms are a class of search algorithms based on problem domain knowledge and experience for solving complex optimization problems, such as the particle swarm algorithm [19], the ant colony algorithm [20], and the simulated annealing algorithm [21]. Due to their non-deterministic nature, heuristic methods exhibit considerable variability in results, making them less suitable for scenarios requiring high timeliness. While swarm intelligence algorithms, such as the artificial fish swarm algorithm [22], enhance the efficiency of heuristic algorithms, their reliance on collaboration and self-organizing behavior may lead to challenges influenced by factors like population size and iteration frequency [23]. Furthermore, a significant limitation they face remains their relatively low timeliness in certain applications.

Deterministic algorithms are methods that use deterministic models and algorithms for computation and decision-making in path planning, without considering random or probabilistic factors, such as the A* algorithm [24], the Dijkstra algorithm [25], and the artificial potential field algorithm (APF) [26]. As the computation time of A* and Dijkstra algorithms is proportional to the size and complexity of the search space, they are typically unsuitable for high-dimensional environments. APF, known for its ease of implementation and excellent timeliness, is a primary deterministic algorithm in high-dimensional environments, often used in combination with other algorithms or enhancements. However, APF, as a local path planning method, faces the challenge of local minima. Once the UAV becomes trapped in a local minimum region, equal-magnitude repulsive and gravitational forces impede path planning progression. Consequently, APF is frequently used in conjunction with other algorithms or after self-improvement. Relevant APF studies are summarized in Table 1.

The Rapidly-exploring Random Tree (RRT) algorithm is a global path planning algorithm based on random sampling. It is capable of handling complex environments, rapidly exploring solution spaces, and does not require an explicit environmental model, making it one of the most widely used global path planning algorithms [35]. However, the stochastic nature of the RRT algorithm makes it very sensitive to the choice of initial sampling points, and different initial sampling points may lead to different path results, which leads to uncertainty and instability in path planning. In certain cases, the RRT algorithm may fall into a situation of poor search quality.

However, conventional methods often apply uniform obstacle avoidance criteria to diverse obstacles, leading to two challenges:

1) Using a smaller obstacle avoidance distance for both immovable obstacles (e.g., rocks, walls) and movable obstacles (e.g., twigs, electric wires). Due to the limited deformation ability of immovable obstacles, the drone can still safely avoid obstacles within a safe distance using a smaller obstacle avoidance distance. Applying the same short distance to movable obstacles could lead to collisions, for example, tree branches susceptible to wind could collide with the UAV;
2) Employing a larger obstacle avoidance distance for both immovable and movable obstacles significantly increases the flight path length.

Effective path planning relies on accurate, real-time, and robust perception of the environment. UAVs need to categorize and locate "flight environmental objects," defined as objects related to flight, such as buildings, trees, and other obstacles. Therefore, building an explicit model of the UAV flight environment is equally crucial. Most current methods use unsupervised learning for object detection and obstacle avoidance, employing sensors like distance sensors and binocular vision cameras to determine obstacle locations and employ algorithms for avoidance. UAVs perform obstacle avoidance based on distance information only, and may not be able to accurately identify and recognize the characteristics of complex obstacles such as trees and buildings. This results in the UAV being unable to perform effective path planning and obstacle avoidance.

Synthesizing the above discussion, this paper proposes a guided Artificial Potential Field (G-APF) algorithm based on environmental awareness. The algorithm is improved through the following key steps:

- YOLOv8-based environment awareness model: Explicitly modeling the UAV flight environment using a YOLOv8-based environment-aware model, enabling the UAV to gain a comprehensive understanding of the environment and its obstacles.
- Adaptive threat distance calculation: An adaptive threat distance calculation module is introduced to construct the repulsive potential field model for the Artificial Potential Field (APF). This allows the UAV to dynamically adjust its repulsion from obstacles based on their characteristics, preventing potential collisions.
- Global path planning with improved rapidly-exploring random tree (IRRT): By incorporating the idea of sampling the APF gravitational potential field, the IRRT efficiently guides the tree toward the target point, ensuring faster convergence.
- Path replanning using improved artificial potential field (IAPF): To address the APF algorithm's tendency to fall into local minima, sub-target points are created by using IRRT, and path replanning is performed using the IAPF.
- Overcoming target unreachability and path oscillation: In this paper, the target unreachability and local path oscillation problems associated with APF are solved by incorporating a distance weighting function for target points and introducing a direction weighting factor.

Furthermore, the model is enriched through supervised learning, where it is trained to learn and recognize the characteristics and attributes of different types of obstacles. This empowers the UAV to more accurately identify various obstacles and provides more detailed and precise obstacle information to the path planning algorithm. By combining the strengths of RRT and APF for path planning, the proposed G-APF algorithm achieves more efficient and safer trajectory planning for UAVs. The incorporation of environmental awareness and adaptive parameters ensures the UAV can avoid obstacles effectively, maintain stability, and reach its target points with improved accuracy.

The structure of this paper is organized as follows: Section II introduces two key algorithms. The limitations and drawbacks of the APF algorithm are discussed, emphasizing the need for alternative approaches. Then explores the benefits and importance of accurate environment modeling in the context of path planning. Section III presents the G-APF algorithm as a solution to address the path planning problem. Section IV evaluates the superiority of the proposed algorithm through simulation experiments conducted in different environments.

II. RELATED WORKS

A. THE RAPIDLY-EXPLORING RANDOM TREE

RRT is a widely used path planning algorithm for finding collision-free paths in complex environments. It was first proposed in 1998 by Steven M. LaValle [36]. It is designed to find collision-free paths in complex environments.

The core idea of RRT is to randomly sample points in the feasible space and gradually connect these sampled points to the growing tree structure to form a path. As the algorithm expands the branches of the tree, it rapidly explores different regions of the environment and adapts to complex obstacle shapes, making it suitable for planning paths in complex and dynamic scenarios.

The initialization phase commences with the selection of an initial point, which serves as the starting position for the mobile entity. This selected point functions as the root node for the creation of the exploration tree. The construction of the path tree unfolds as an iterative process. Initially, a random sampling point is generated within the unoccupied space. Subsequently, we identify the nearest node to this randomly sampled point within the existing exploration tree. A new node is then generated by taking a small step from the nearest node towards the randomly sampled point. Subsequently,

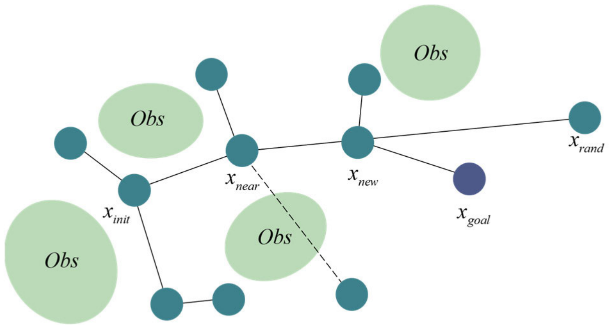


FIGURE 1. The expansion schematic of the RRT tree.

Algorithm 1 RRT Algorithm

```
Input: $\mathcal{R}$, $X_{init}$, $X_{goal}$, $X_{obs}$, $StepSize$
Result: A path $\mathcal{T} = \{X_1, \ldots X_n\}$ from $X_{init}$ to $X_{goal}$
$\mathcal{T}.init()$;
for $i = 1$ to $n$ do
    $X_{rand} \leftarrow Sample(\mathcal{R})$;
    $X_{near} \leftarrow Near(X_{rand}, \mathcal{T})$;
    $X_{new} \leftarrow Steer(X_{rand}, X_{near}, StepSize)$;
    if $CollisionFree(X_{new}, X_{obs})$ then
        $\mathcal{T}.addNode(X_{new})$;
    if $X_{new} = X_{goal}$ then
        return $\mathcal{T}$;
```

a collision check is executed to ascertain whether the newly generated node collides with any obstacles. If no collisions are detected, the new node is incorporated into the exploration tree, with its parent node being designated as the closest node. Moreover, we evaluate whether the new node lies in proximity to the target point or has reached the target point itself. If the target conditions are met, the algorithm concludes. Upon successful termination of the algorithm with a path discovery, the path can be extracted from the exploration tree through a backtracking process. This process involves tracing from the target node back to the starting node and establishing connections via the parent nodes to derive the complete path.

The expansion schematic of the RRT tree is visually illustrated in Figure 1, showing how the tree structure expands to explore the environment and find potential paths.

The detailed algorithm flow of RRT is outlined in Algorithm 1, providing a step-by-step description of the process, from random sampling to tree expansion, until the algorithm reaches its termination condition or finds a valid path.

RRT is recognized for its efficiency and effectiveness in addressing complex environments featuring obstacles, rendering it a pivotal algorithm in the realms of robotics and path planning, enjoying widespread adoption. Nevertheless, the RRT algorithm exhibits a propensity for rapid expansion within the search space, without a guarantee of uncovering the global optimal solution. Additionally, its performance may be suboptimal in high-dimensional spaces, with slower convergence or potential failure in instances of expansive search spaces and heightened complexity.

In Algorithm 1, the variables are: $\mathcal{R}$, $\mathcal{T}$, X_{init}, X_{goal}, X_{obs}, $StepSize$, X_{rand}, X_{near}, X_{new}; The functions are: $init()$, $Sample()$, $Near()$, $Steer()$, $CollisionFree()$, $addNode()$. The variables and functions along with their meanings are shown in Table 2.

B. THE ARTIFICIAL POTENTIAL FIELD

The basic idea of the artificial potential field method is to detect the obstacle situation of the environment by several environmental sensors and to generate gravitational and repulsive potential fields for the target point and the obstacle, respectively. In a combined potential field environment consisting of a gravitational potential field and multiple repulsive potential fields, the UAV will move in the direction of the falling potential field. The combined gravitational and repulsive forces act on the UAV to generate a safe path from the start point to the target point. Compared to generating a full path, the algorithm only considers the combined potential field at the current path point, resulting in high real-time performance.

The gravitational and repulsive potential field functions can be expressed as follows:

$$U_{att}(X) = ck_{att}[d(X, X_e)]^2 \tag{1}$$

$$U_{rep}(X) = \begin{cases} ck_{rep}(\dfrac{1}{d(X, X_t)} - \dfrac{1}{\rho})^2, & d(X, X_t) \leq \rho \\ 0, & d(X, X_t) > \rho \end{cases} \tag{2}$$

where, c is the constant term, k_{att} and k_{rep} are the gravitational and repulsion coefficient, $X = (x, y, z)$ is the current position of the UAV, $X_e = (x_e, y_e, z_e)$ is the location of the target point, $X_t = (x_t, y_t, z_t)$ is the location of the obstacle, $d(X, X_e) = \sqrt{(X_e - X)^2}$is the Euclidean distance between the UAV and the target point, $d(X, X_t) = \sqrt{(X_t - X)^2}$ is the Euclidean distance between the UAV and the obstacle, and ρ is the obstacle threat distance.

The magnitude of the gravitational and repulsive forces can be obtained by finding the negative gradient of the gravitational and repulsive potential fields, expressed as follows:

$$F_{att}(X) = -\nabla(U_{att}) = -2ck_{att}d(X, X_e)\frac{\partial d(X, X_e)}{\partial X} \tag{3}$$

$$\begin{aligned} F_{rep}(X) &= -\nabla(U_{rep}) \\ &= \begin{cases} \dfrac{2ck_{rep}}{[d(X, X_t)]^2}(\dfrac{1}{d(X, X_t)} - \dfrac{1}{\rho})\dfrac{\partial d(X, X_t)}{\partial X}, \\ \qquad\qquad d(X, X_t) \leq \rho \\ 0, \qquad\quad d(X, X_t) > \rho \end{cases} \end{aligned} \tag{4}$$

The combined force of the UAV (as shown in Figure 2) is determined by several repulsive forces and a gravitational force together. The combined force provides the UAV's next

TABLE 2. Variables and functions of RRT.

Variables and functions	Meaning
R	Free Space
T	Waypoint collection
$StepSize$	Random tree expansion step
$X_{init}, X_{goal}, X_{obs}$	Start points, target points, obstacle points
$X_{rand}, X_{near}, X_{new}$	Extension points, proximity points, feasible points
$init(\,)$	Initialization function
$Sample(\,)$	Random sampling function
$Near(\,)$	Find the nearest point function
$Steer(\,)$	Extended Functions
$Steer(X_{rand}, X_{near}, StepSize)$	Expanding from X_{near} to X_{rand} in steps of $StepSize$
$CollisionFree(\,)$	Detects collision function
$addNode(\,)$	Add node function

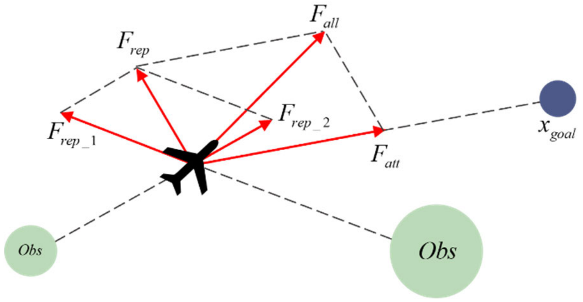


FIGURE 2. The combined force model of the UAV.

flight direction, which allows the UAV to safely reach the target point while avoiding obstacles, expressed as follows:

$$F_{all}(X) = F_{att}(X) + \sum_{i=1}^{n} F_{rep\_i}(X) \tag{5}$$

The APF method is simple to implement, but since the combined force is only a simple iteration of the gravitational force and the repulsive force, this method has the following drawbacks:

1) Target unreachability problem: As the UAV approaches the target point, the gravitational potential field generated by the target decreases, while the repulsive potential field generated by nearby obstacles increases. In certain situations, the repulsive force may exceed the gravitational force, causing the combined force to point away from the direction of the target point. This can result in the UAV being unable to reach the target, leading to a target unreachability problem (Figure 3(a)).
2) Local minimum trap: During path planning, the UAV only has access to local information about its current position, lacking a global perspective. In complex environments, this limited local information may lead to situations where the combined force becomes zero, leaving the UAV unable to choose a better path. This can result in the UAV becoming trapped in a local minimum, hindering its ability to find an optimal path (Figure 3(b)).
3) Local path oscillation: As the UAV approaches the target point, the gravitational potential field dominates, guiding the UAV towards the target. However, when the UAV encounters an obstacle, the repulsive potential field takes precedence, causing the UAV to move away from the obstacle. This continuous interaction of gravitational and repulsive forces in the vicinity of obstacles can lead to a back-and-forth oscillatory motion pattern, causing the UAV to repeatedly approach and move away from obstacles. This behavior results in local path oscillation (Figure 3(c)).

C. ACCURATE OBJECT DETECTION IN PATH PLANNING

Most UAVs typically rely directly on sensors for obstacle location information during obstacle avoidance, without adequately considering crucial factors such as the type, size, and possible deformation or movement of the obstacle. This generalized categorization of all objects as "obstacles" gives rise to the following issues:

1) The same avoidance strategy for objects of different sizes or categories (e.g., trees, poles, walls) may result in the UAV not being able to distinguish between obstacle types and accurately calculate the avoidance space, which may cause the UAV to be overly conservative or risky in the avoidance process and not fully utilize the

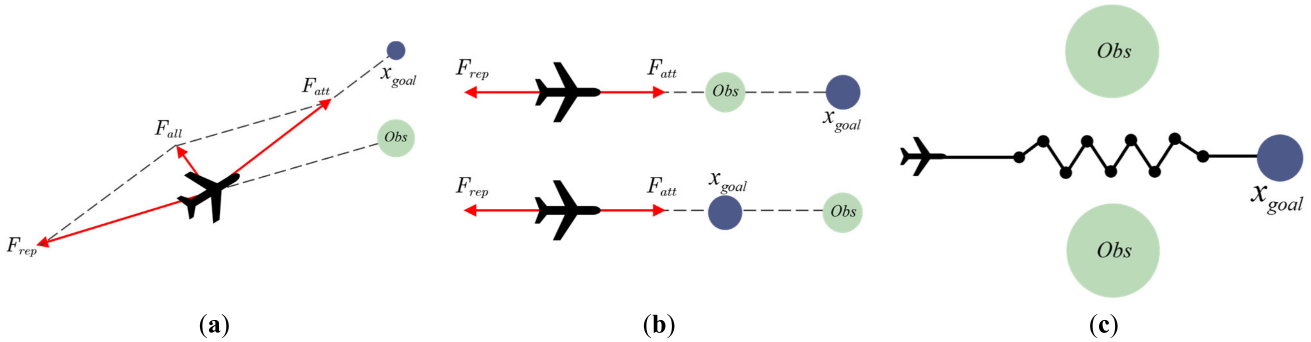


FIGURE 3. Problems with APF in different situations. (a) Target unreachability problem; (b) Local minimum problem; (c) Local path oscillation problem.

available space. This limits the UAV's maneuverability and mission effectiveness.

2) Some objects in real-world scenarios do not remain stationary, they may deform or move (e.g., trees, flags), for fixed objects, UAVs can choose to go around or static avoidance, while for shifting or moving obstacles we need a more flexible avoidance strategy.

To ensure the safe operation of UAVs, a precise perception and perception of the environment are paramount. The perception system of UAV transforms sensory data into semantic information, such as identification and recognition of objects' positions and classes. Notably, the object detection task is of fundamental importance, as any failure to accurately identify and recognize objects has the potential to lead to safety-related incidents [37]. Therefore, it is crucial to address the aforementioned issues in UAV obstacle avoidance systems to enhance safety and overall performance.

III. PROPOSED METHODOLOGY

A. OVERVIEW

From the above discussion, since the UAV is incapable of detecting the size and category of objects in the environment, the traditional unsupervised learning method for obstacle avoidance strategy will reduce the efficiency of path planning and even lead to planning failure. In addition, using a single global algorithm for path planning will have uncertainty, resulting in increased planning time costs or even planning failure. By using a single local algorithm for path planning, the UAV will be unable to obtain global information and will be prone to falling into a local minimum, resulting in planning failure.

Traditional path planning algorithms typically rely on sensor data and image extraction algorithms for environment understanding and single algorithm path planning. Object detection through supervised learning fusion provides more accurate and richer perceptual information for path planning [38], [39]. The fusion of planning algorithms fully exploits the advantages of different algorithms to compensate for their respective shortcomings and improve the performance, robustness, and reliability of the system [40].

The method in this paper consists of an environment-aware module and a path-planning module. The environment sensing module adopts the idea of deep learning to detect objects in the UAV flight environment through self-aware learning and proposes an adaptive threat distance calculation module to establish the explicit model and potential field model of the flight environment. The path planning module improves RRT and APF and fuses them to obtain G-APF, a local path replanning method guided by global planning. The path planning module improves and fuses RRT and APF to obtain G-APF, a local path replanning method under the guidance of global planning. We introduce the concept of APF to direct the expansion of the exploration tree towards the target point in RRT. This addition enhances the goal-oriented nature of tree expansion, moving away from random expansion strategies. To enhance the effectiveness of the APF, we tackle three pivotal issues: firstly, we resolve the local minimum problem by establishing sub-target points. Secondly, we address the challenge of target unreachability by introducing a target point distance weight function. Finally, we mitigate the problem of path oscillation by incorporating a directional weighting factor. A schematic of the above methodology is shown in Figure 4. By integrating environment awareness and fusion of planning algorithms, the approach presented in this paper aims to overcome the limitations of traditional methods, ultimately leading to improved efficiency, robustness, and reliability in UAV path planning and obstacle avoidance.

B. ENVIRONMENT AWARE MODULE

For high-accuracy path planning, precise obstacle detection is essential. However, most algorithms suffer from imprecise obstacle detection. In addition, different types of obstacles have different effects on path planning. We design an environment-aware model based on YOLOv8 to address these issues.

YOLOv8 is a powerful deep learning model that enables fast and accurate detection and classification of objects in images [41]. Our environment-aware model uses YOLOV8 to detect and classify obstacles in the UAV flight environment. Through the integration of the environment-aware model,

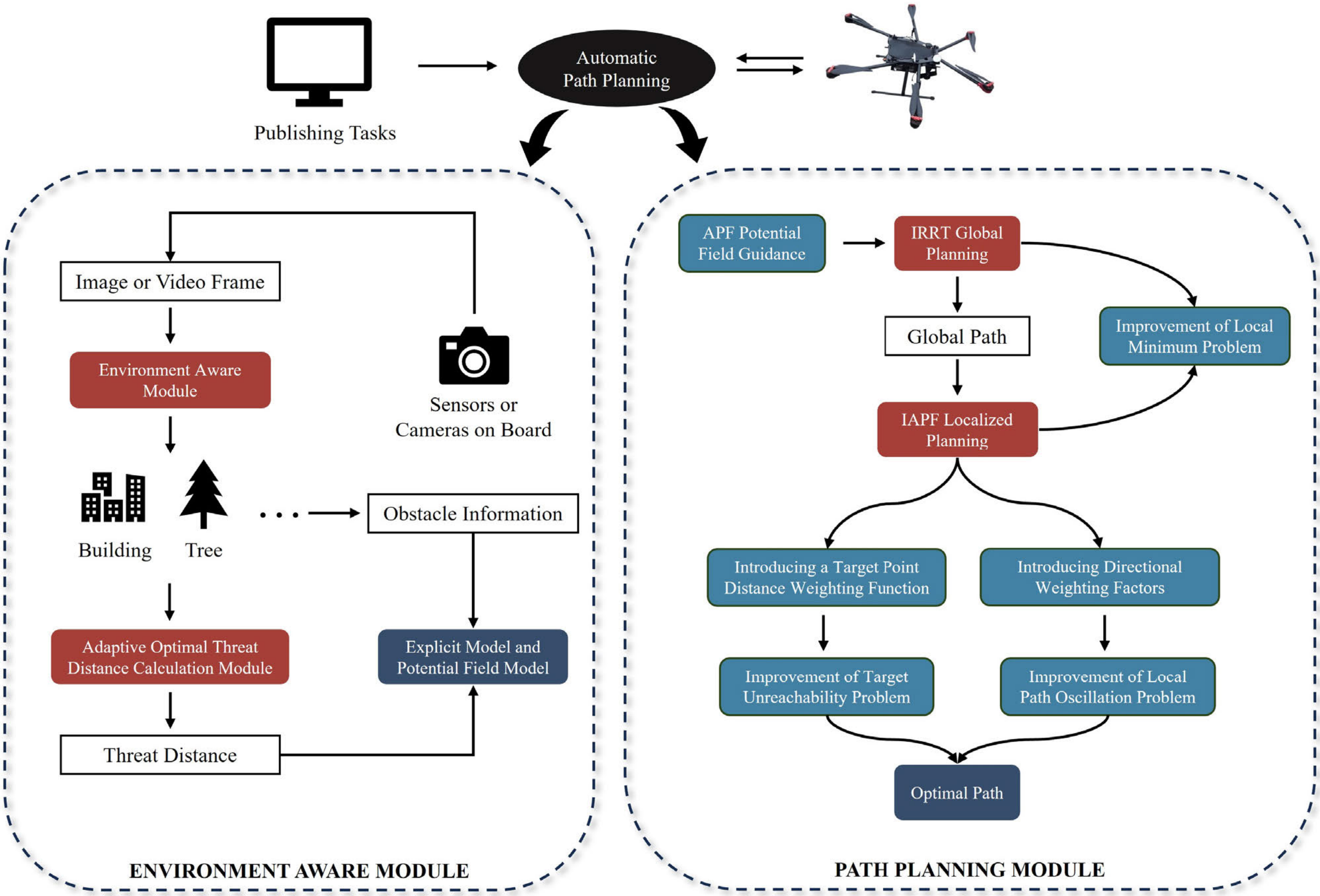

Publishing Tasks
Automatic Path Planning
Image or Video Frame
Environment Aware Module
Sensors or Cameras on Board
Building
Tree
Obstacle Information
Adaptive Optimal Threat Distance Calculation Module
Explicit Model and Potential Field Model
Threat Distance
ENVIRONMENT AWARE MODULE
APF Potential Field Guidance
IRRT Global Planning
Global Path
Improvement of Local Minimum Problem
IAPF Localized Planning
Introducing a Target Point Distance Weighting Function
Introducing Directional Weighting Factors
Improvement of Target Unreachability Problem
Improvement of Local Path Oscillation Problem
Optimal Path
PATH PLANNING MODULE

FIGURE 4. Schematic of UAV path planning.

we can attain highly accurate obstacle detection results, which serve as essential inputs for the path planning process. Moreover, we recognize that different types of obstacles can have varying impacts on path planning. To account for this, our model takes into consideration the specific effects of different obstacle types, enabling a more intelligent and optimized path planning solution for the UAV.

Considering that the flight environment of the UAV in this paper is low to medium altitude, cars and pedestrians are less threatening to the UAV, thus only objects such as trees, buildings, traffic poles, and billboards are detected and classified. We used the VisDrone dataset [42] for validation, and the detection results are shown in Figure 5.

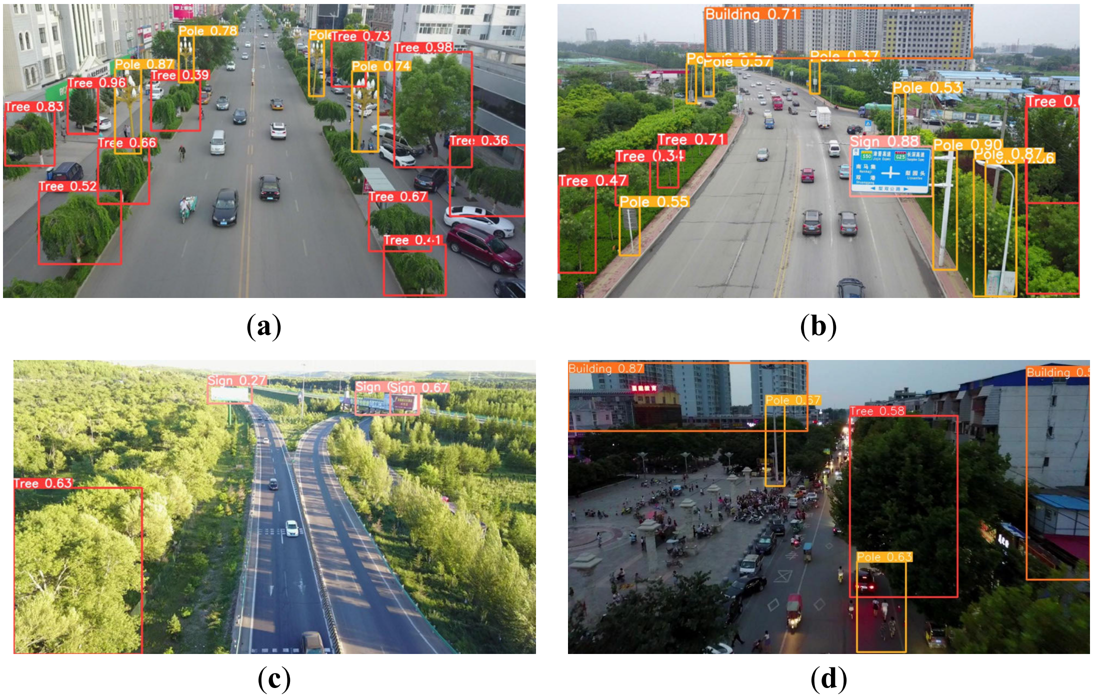
In object detection, objects such as trees are easily deformed and are more threatening to UAVs and are considered "non-rigid objects"; objects such as buildings and traffic poles are less threatening to UAVs and are considered "rigid objects". The classification is shown in (6).

To store essential information about each detected object, we utilize the data structure $\left[\hat{L}\right]_i$. This data structure contains four key attributes for each object:

- Position X: This attribute represents the position of the object in the UAV's flight environment, providing spatial information for further analysis and planning.
- Classification result k: The classification result denotes the category or type of the detected object, distinguishing between non-rigid and rigid objects.
- Confidence level τ: The confidence level represents the certainty or reliability of the object detection process, indicating how confident the model is in its classification result.
- Threat distance ρ: The threat distance indicates the proximity of the object to the UAV, signifying the potential risk or danger it poses to the UAV during its flight.

$$Class = \begin{cases} RigidObs(Storeto\left[\hat{L}\right]_i), & k \in \{Sign, Pole, Building\} \\ NonRigidObs(Storeto\left[\hat{L}\right]_i), & k \in \{Tree\} \\ Discard, & others \end{cases} \tag{6}$$

Based on this feature, we established an adaptive optimal threat distance calculation module based on environment



(a)

(b)

(c)

(d)

FIGURE 5. Detection results.

awareness: This module assigns different threat distances to different types of objects, ensuring a more tailored and intelligent approach to obstacle avoidance.

The threat distance for each detected object is determined by the adaptive threat distance calculation module, as depicted in (7).

$$\rho_i = \rho_0 \cdot \tau_i^{-1} \cdot \eta_k, k \in \{Tree, Pole, Sign, Building\}, \quad i = 1, 2, \ldots n \tag{7}$$

The calculation incorporates three main factors:

- Base threat distance ρ_0: This represents the initial threat distance assigned to all objects as a baseline.
- Confidence level τ_i: The confidence level associated with each detected object is taken into account. Objects with higher confidence levels are considered more reliable in their detection results, potentially warranting a reduced threat distance.
- Weighting factor η_k: The weighting factor, denoted as η_k, is utilized to further adjust the threat distance based on the type of object. Different weighting factors are specified for various categories of objects. Specifically, $\eta_{Tree} = 1.5$, $\eta_{Pole} = 1.3$, $\eta_{Sign} = 1.2$, $\eta_{Building} = 1.0$ are the predefined values for the weighting factors.

The aforementioned steps enable us to gather comprehensive information about each obstacle's location, shape, size, category, and threat distance, allowing us to construct an explicit flight environment model. This model serves as a representation of the environment surrounding the UAV, providing crucial data for path planning and flight decision-making processes.

In the explicit model of the flight environment, obstacles are identified and depicted on the map or virtual environment based on their precise location and shape. Additionally, the adaptive threat distance calculation module determines the threat distance for each obstacle, which is then visually differentiated using distinct icons or colors. This intuitive representation helps in better understanding the complexity of the flight environment.

The different threat distances assigned to each obstacle correspond to various obstacle avoidance decisions. Consequently, when establishing the artificial potential field, these varying threat distances result in different potential field sizes, effectively addressing the issue of using the same obstacle avoidance strategy for all obstacles.

The repulsive potential field is formulated as shown in (8), which differs from (2) in that the repulsive potential field of different classes of objects is influenced by their respective threat distances. By incorporating the threat distance information, the repulsive potential field becomes more adaptable to the specific characteristics of each obstacle, allowing the UAV to plan its path with greater precision and safety.

$$U_{rep}(X) = \begin{cases} ck_{rep}(\frac{1}{d(X, X_t)} - \frac{1}{\rho_i})^2, & d(X, X_t) \leq \rho_i \\ 0, & d(X, X_t) > \rho_i \end{cases} \tag{8}$$

C. PATH PLANNING MODULE

1) THE GLOBAL PATH PLANNING

IRRT algorithm builds upon the RRT by incorporating the concept of the APF as a guiding function. While RRT involves free sampling in space, which can lead to significant randomness, the addition of the APF potential field in IRRT serves as a guidance mechanism.

In Algorithm 2, the detailed flow of the IRRT algorithm is presented. The algorithm employs the APF potential field function to guide the sampling process. By doing so, each sampling point is steered gradually towards the target point. This intelligent guidance provided by the potential field ensures that the sampling points are selected in a more informed manner, resulting in improved efficiency during the path search. Consequently, the search space and search time are reduced.

Algorithm 2 The Global Path Planning Algorithm

```
Input: $\mathcal{R}, X_{init}, X_{goal}, X_{obs}, Stepsize$
Result: A path $\mathcal{T} = \{X_1, \ldots X_n\}$ from $X_{init}$ to $X_{goal}$
$\mathcal{T}.init()$;
$\mathcal{T} = \{X_{init}\}; X_{new} = X_{init}$;
for $i = 1$ to $n$ do
    $X_{rand} \leftarrow Sample(\mathcal{R})$;
    $F_{rand} \leftarrow CalPotential(X_{rand})$;
    $F_{new} \leftarrow CalPotential(X_{new})$;
    if $F_{rand} < F_{new}$ then
        $X_{near} \leftarrow Near(X_{rand}, \mathcal{T})$;
        $X_{new} \leftarrow Steer(X_{rand}, X_{near}, StepSize)$;
    else
        continue;
    if $CollisionFree(X_{new}, X_{obs})$ then
        $\mathcal{T}.addNode(X_{new})$;
    if $X_{new} = X_{goal}$ then
        return $\mathcal{T}$;
```

In the algorithm, *CalPotential*() is the function to calculate the potential field. If the potential field of X_{rand} is smaller than the potential field of X_{new}, it means that X_{rand} is the point closer to the endpoint than X_{new}, otherwise X_{rand} will be removed and resampled.

The above steps are taken to ensure that each sampling point gradually approaches the target point. By increasing the guidance of the potential field, the sampling points are selected more intelligently and the efficiency of the path search is improved, thus reducing the search space and search time.

2) THE LOCAL PATH PLANNING

In the path planning process, we first obtain a global path using the IRRT algorithm. The UAV will follow this global path during its flight. However, to ensure safe obstacle avoidance, we perform path replanning using the IAPF method.

As the UAV encounters obstacles during its flight, IAPF dynamically adjusts the path to safely avoid these obstacles. If the UAV becomes trapped in a local minimum during path planning, IRRT guides the UAV to break free from the local minimum and continue its path towards the target.

In Section III, we leverage supervised learning to detect and classify obstacles in the flight environment, assigning different threat distances based on their characteristics. During path planning with IAPF, the UAV adopts distinct obstacle avoidance strategies for obstacles with different threat distances. For objects with small threats, the UAV flies as close as possible to minimize the flight range. In contrast, for objects with larger threats, the UAV maintains a greater distance to ensure flight safety.

Furthermore, we have addressed two challenges associated with the APF algorithm. To overcome the target unreachability problem, we have improved the potential field model. Additionally, to mitigate the local path oscillation problem, we introduced directional weighting factors.

The detailed steps are argued as follows:

1) Improvement of local minimum problem

To effectively overcome the issue of local minima in the potential field, our algorithm introduces the concept of sub-target points. When the UAV becomes trapped in a local minimum during the potential field-based path planning, we identify the closest global path point that was planned by the RRT from the UAV's current position. This closest point is then designated as a sub-target point, which becomes the new target point for the local planning process.

The UAV follows an optimal path determined by local planning, which guides it out of the local minimum and towards the sub-target point. Once the UAV successfully navigates out of the local minimum and reaches the sub-target point, the local goal point is reset to the original goal point of the path planning process. This allows the algorithm to continue generating the path until the UAV ultimately reaches the final target point.

The detailed flow of this process is presented in Algorithm 3. This approach ensures that the UAV effectively jumps out of local minima, thus preventing it from getting stuck in these situations during the path planning process. By incorporating sub-target points and redefining the local goal point, the algorithm optimizes the UAV's path planning, enhancing its efficiency and robustness in complex environments.

CalForce() is a function to calculate the potential field force, *MinDist*() is a function to calculate the point corresponding to the shortest distance, and *CalLocation*() is a function to calculate the coordinates of the next position. Through the above steps, the robustness and reliability of path planning can be improved by setting sub-goal points to guide the UAV to jump out of the local minimum and continue path planning. Local path planning uses the nearest point in the global path as a subgoal point, allowing the UAV

Algorithm 3 Improvement of Local Minimum Problem

```
Input: $\mathcal{R}, \mathcal{T}, X_{init}, X_{goal}, X_{obs}, Stepsize$
Result: A path $\zeta = \{X_1', \ldots X_n'\}$ from $X_{init}$ to $X_{goal}$
$\zeta.init()$; $X_1' = X_{init}$;
for $i = 1$ to $n$ do
    $F_i \leftarrow CalForce(X_i', X_{obs}, X_{goal})$;
    if $F_i = 0$ then
        $X_{sub} \leftarrow MinDist(X_i', \mathcal{T})$;
        $F_i' \leftarrow CalForce(X_i', X_{obs}, X_{sub})$;
        $X_{i+1}' \leftarrow CalLocation(X_i', F_i', Stepsize)$;
    else $X_{i+1}' \leftarrow CalLocation(X_i', F_i, Stepsize)$;
    $\zeta.addNode(X_{i+1}')$;
    if $X_i' = X_{goal}$ then
        return $\zeta$;
```

to quickly optimize the path in a local area. This reduces the path length and improves the efficiency and optimization of the path. After jumping out of the local minimum, reset the final target point as the target point for path planning to ensure continuity and consistency of the path. This avoids interruptions or discontinuities in the path, allowing the drone to reach the target point smoothly.

2) Improvement of target unreachability problem

For the target unreachability problem, this paper improves the potential field function to ensure that the gravitational force of the target point can still pull the UAV to the target position when the target is unreachable. In improving the potential field function, the gravitational potential field function is defined as in the traditional method, but the target point distance weighting function $\varphi(X, X_e) = \frac{1}{2} - \frac{1}{1+e^{-d(X,X_e)}}$ is added to the repulsive potential field function, and the improved repulsive potential field function is as follows:

$$U_{rep}(X) = \begin{cases} ck_{rep}(\frac{1}{d(X,X_t)} - \frac{1}{\rho_i})^2 \varphi(X, X_e), & d(X,X_t) \\ & \leq \rho_i \\ 0, & d(X,X_t) \\ & > \rho_i \end{cases} \tag{9}$$

where, c is a constant term. The new repulsive function is obtained by finding the negative gradient of the repulsive potential field function as:

$$F_{rep}(X) = -\nabla(U_{rep}) = \begin{cases} F_{rep1} + F_{rep2}, & d(X,X_t) \leq \rho_i \\ 0, & d(X,X_t) > \rho_i \end{cases} \tag{10}$$

where, F_{rep1} is the repulsive force of the obstacle facing the UAV and F_{rep2} is the gravitational force of the target point facing the UAV, defined as follows:

$$F_{rep1}(X) = (\frac{1}{d(X,X_t)} - \frac{1}{\rho_i})\frac{2ck_{rep}}{d(X,X_t)^2}\varphi(X, X_e)\frac{\partial d(X,X_t)}{\partial X} \tag{11}$$

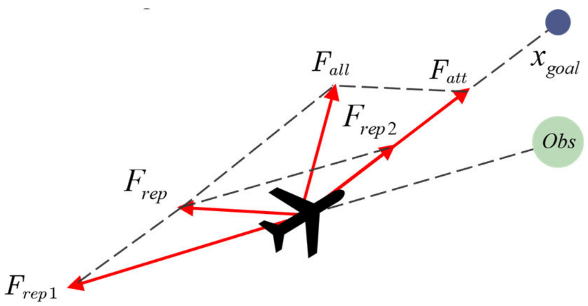


FIGURE 6. The Improved combined force model.

$$F_{rep2}(X) = (\frac{1}{d(X,X_t)} - \frac{1}{\rho_i})^2 e^{-d(X,X_e)} \frac{ck_{rep}}{(1+e^{-d(X,X_e)})^2} \times \frac{\partial d(X,X_e)}{\partial X} \tag{12}$$

As the UAV gradually approaches the target point, $d(X, X_e)$ gradually decreases and F_{rep1} gradually converges to 0. At this point, the UAV's repulsive force is mainly composed of F_{rep2}. F_{rep1} and F_{rep2}, which together determine the magnitude and direction of the new repulsive force, as shown in Figure 6.

Through the above steps, when the UAV gradually approaches the target point, the gravitational force of the target point on the UAV decreases, thereby causing the problem that the target cannot be reached. By introducing a target point distance weighting function, the balance of the combined force can be effectively regulated when the UAV approaches the target point. As the distance between the UAV and the target point decreases, the influence of the repulsive force is gradually weakened, while the gravitational force is enhanced to ensure that the UAV continues to be attracted by the gravitational force when it encounters obstacles, while the influence of the repulsive force is reduced to continue to guide the UAV close to the target point.

3) Improvement of local path oscillation problem

For the local path oscillation problem, this paper proposes an optimization method based on the steering angle and the introduction of a directional weighting factor, which suppresses the steering angle and thus the oscillation when the UAV makes a large steering at two adjacent path points.

The combined force of the UAV at the current position X_n to fly to the next path point at the theoretical position X_{n+1} is calculated as F_{n+1}, and by introducing the combined force F_n from the previous path point X_{n-1} to the current position X_n, the angle change $\Delta\alpha$ between F_n and F_{n+1} is calculated, and the actual position to fly to the next path point is determined by the magnitude $\Delta\alpha$ of the expression as follows:

$$X_{n+1} = \begin{cases} X_n + L \cdot (m_1 \cdot w_n + m_1' \cdot w_{n+1}), & \Delta\alpha \leq \theta \\ X_n + L \cdot (m_2 \cdot w_n + m_2' \cdot w_{n+1}), & \Delta\alpha > \theta \end{cases} \tag{13}$$

where, m_i and m_i' $(m_i + m_i' = 1)$ are the directional weight factors, $w_n = X_n - X_{n-1}$ and $w_{n+1} = X_{n+1} - X_n$ are the

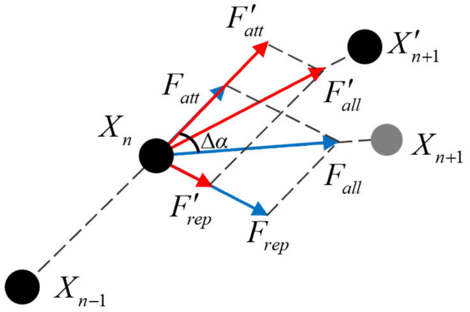


FIGURE 7. Force diagram of UAV when calculating the next position.

directional vectors of F_n and F_{n+1}, respectively, L is the motion step, and θ is the angular threshold. The values of the directional weighting factor and the angular threshold can be referred to as follows: $m_i = m'_i$, if $\Delta\alpha \leq \frac{\pi}{6}$; $m_i > m'_i$, if $\frac{\pi}{6} < \Delta\alpha \leq \frac{\pi}{3}$; $m_i \gg m'_i$, if $\Delta\alpha > \frac{\pi}{3}$.

When the UAV's angle of deviation from the current heading is small, it is considered a minor deviation, and the UAV will continue to fly in the preset direction without making significant adjustments. This fine-tuning ensures that the UAV maintains its course when encountering slight deviations, avoiding unnecessary corrections that could lead to instability. However, when the UAV's angle of deviation from the current heading is large, the path is prone to oscillation. To mitigate this oscillation, the algorithm introduces a direction weighting factor. The direction weighting factor works to guide the UAV's heading as close as possible to the initial heading. By doing so, excessively large turning angles are avoided, and the oscillation is suppressed.

As shown in Figure 7, X_{n-1} and X_n are the previous path point position and the current position of the UAV; X_{n+1} is the theoretical position of the next path point of the UAV; F_{att}, F_{rep} and F_{all} are the theoretical gravitational force, repulsive force, and the combined force of the UAV; X'_{n+1} is the actual position of the next path point of the UAV; F'_{att}, F'_{rep} and F'_{all} are the actual gravitational force, repulsive force and combined force of the UAV, respectively.

IV. SIMULATION EXPERIMENT AND ANALYSIS

This paper presents a comprehensive evaluation of the G-APF algorithm's performance through a series of simulation experiments conducted in MATLAB. The experiments cover various environments to validate the method's feasibility, and it is compared against several existing algorithms, including RRT and APF, as well as improved versions IRRT and IAPF. Additionally, to provide a more comprehensive assessment, we incorporate comparative experiments with two recently introduced algorithms, namely APF-PSO [43] and MOD-RRT* [44].

To simulate the obstacles, spherical obstacles are approximated as non-rigid objects, while columnar obstacles are approximated as rigid objects. To provide a visual representation of the different threat distances associated with these obstacles, the spheres and columns are colored accordingly, this article sets a total of four different threat distances. Lighter colors are used to depict small threat distances, while darker colors represent larger threat distances.

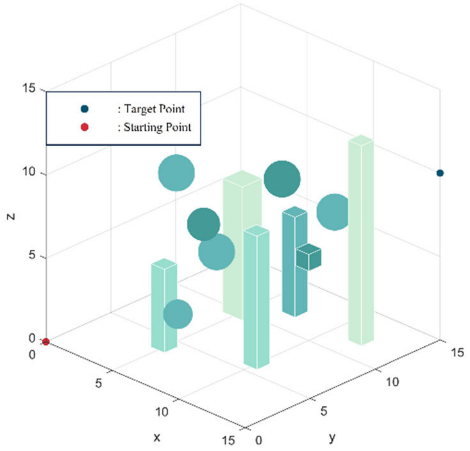


FIGURE 8. A simple simulation environment for path planning.

TABLE 3. Parameter setting.

Parameter Name	Parameter Value
Start Point	(0,0,0)
Target Point	(15,15,10)
Step Size(m)	0.2
ρ_0 (m)	1
Threat Distance ρ_1 (m)	1.2
Threat Distance ρ_2 (m)	1.3
Threat Distance ρ_3 (m)	1.4
Threat Distance ρ_4 (m)	1.5
APF Gravitational Coefficient	15
APF Repulsion Coefficient	10

A. PATH PLANNING IN A SIMPLE ENVIRONMENT

The path planning simulation environment is shown in Figure 8, the size of the area is 15 m*15 m*15 m. The red point is the starting point with coordinates (0, 0, 0) and the blue point is the target point with coordinates (15, 15, 10). The parameter settings in this experiment are shown in Table 3.

Figure 9(a-g) shows the planning results of each algorithm. The comparison of the different path planning methods shows that there are significant differences in the path characteristics. The characteristics are as follows:

- The paths generated by APF (Figure 9(a)) have a higher degree of tortuosity and unnecessary turns compared to those generated by IAPF (Figure 9(b)). This observation highlights the inherent limitations of APF in achieving optimal and smooth path planning.
- On the other hand, the RRT algorithm (Figure 9(c)) shows a higher level of complexity when compared to

the sampling process used by IRRT (Figure 9(d)). The inherent complexity of the RRT algorithm stems from its more involved sampling strategy, resulting in a more extensive exploration of the configuration space.

- The inherent complexity of the RRT algorithm stems from its more involved sampling strategy, resulting in a more extensive exploration of the configuration space. Conversely, the expansion of the tree structure in IRRT is significantly hampered by the influence of the potential field, thus limiting its potential for extensive exploration and path expansion.
- In comparison to the path planning results of the APF-PSO algorithm (Figure 9(e)), the generated path by G-APF (Figure 9(g)) is more direct, successfully avoiding unnecessary turns. In contrast to APF-PSO, the path planning of G-APF exhibits more efficient and intuitive characteristics during obstacle avoidance. On the other hand, when compared to the path planning results of the MOD-RRT* algorithm (Figure 9(f)), we distinctly observe that the exploration of the MOD-RRT* tree surpasses that of G-APF. In G-APF, the tree expansion is notably more efficient, resulting in a simpler and more straightforward planned path. In contrast, the tree expansion of MOD-RRT* is more complex, leading to increased curvature in the planned path.
- The G-APF algorithm incorporates a localized replanning mechanism. Building on the initial path generated by IRRT, this adaptive algorithm uses the insights gained from the IRRT exploration process to identify areas of suboptimal path planning. By recalculating the path locally, the algorithm aims to minimize unnecessary turns and ensure closer adherence to obstacles, thereby improving the overall efficiency and effectiveness of the trajectory planning process.

Table 4 presents the performance evaluation of each algorithm based on 100 iterative trials. The results show that the G-APF algorithm outperforms the other algorithms in terms of planning time. It achieves an average planning time reduction of 23.1% compared to the traditional APF algorithm, 9.8% compared to IAPF, 51.7% compared to RRT, 35.1% compared to IRRT, 18.9% compared to APF-PSO, and 53.3% compared to MOD-RRT*.

Regarding planning path lengths, the IAPF algorithm produces the shortest lengths among all the algorithms tested. However, even though the G-APF algorithm increases the path length by 47.4% compared to IAPF, 41.2% compared to APF, and 31.5% compared to APF-PSO, it still manages to decrease the path length by 20.2% compared to RRT, 19.9% compared to IRRT, and 18.4% compare to MOD-RRT*.

Figure 10 visually illustrates the time performance data for each algorithm. The error bars in the figure represent the minimum and maximum planning times for the corresponding algorithms, and the columnar bars represent the average planning time for each algorithm.

TABLE 4. Performance of each algorithm.

	Minimu m time(s)	Average time(s)	Maximum time(s)	Average path length(m)
APF	/	1.244	/	18.195
IAPF	/	1.061	/	**17.432**
RRT	1.601	1.983	2.393	32.239
IRRT	1.018	1.474	1.891	32.126
APF-PSO	0.905	1.178	1.475	19.546
MOD-RRT*	1.564	2.049	2.579	31.517
G-APF	**0.601**	**0.956**	**1.290**	25.703

In a simple environment, the APF algorithm demonstrates high time efficiency and generates short path lengths compared to other algorithms. The reason for this lies in the local nature of APF path planning. It only needs to calculate the local potential field of each position during the planning process, resulting in faster computation. Additionally, the paths generated by APF tend to be shorter due to its focus on local obstacle avoidance.

In contrast, the RRT algorithm explores the entire free space, which can be highly time-consuming. Moreover, the paths generated by RRT may not always be optimal as its tree expansion process relies on random sampling.

The IRRT algorithm, which incorporates the concept of a potential field, achieves a better balance between time efficiency and path length compared to RRT.

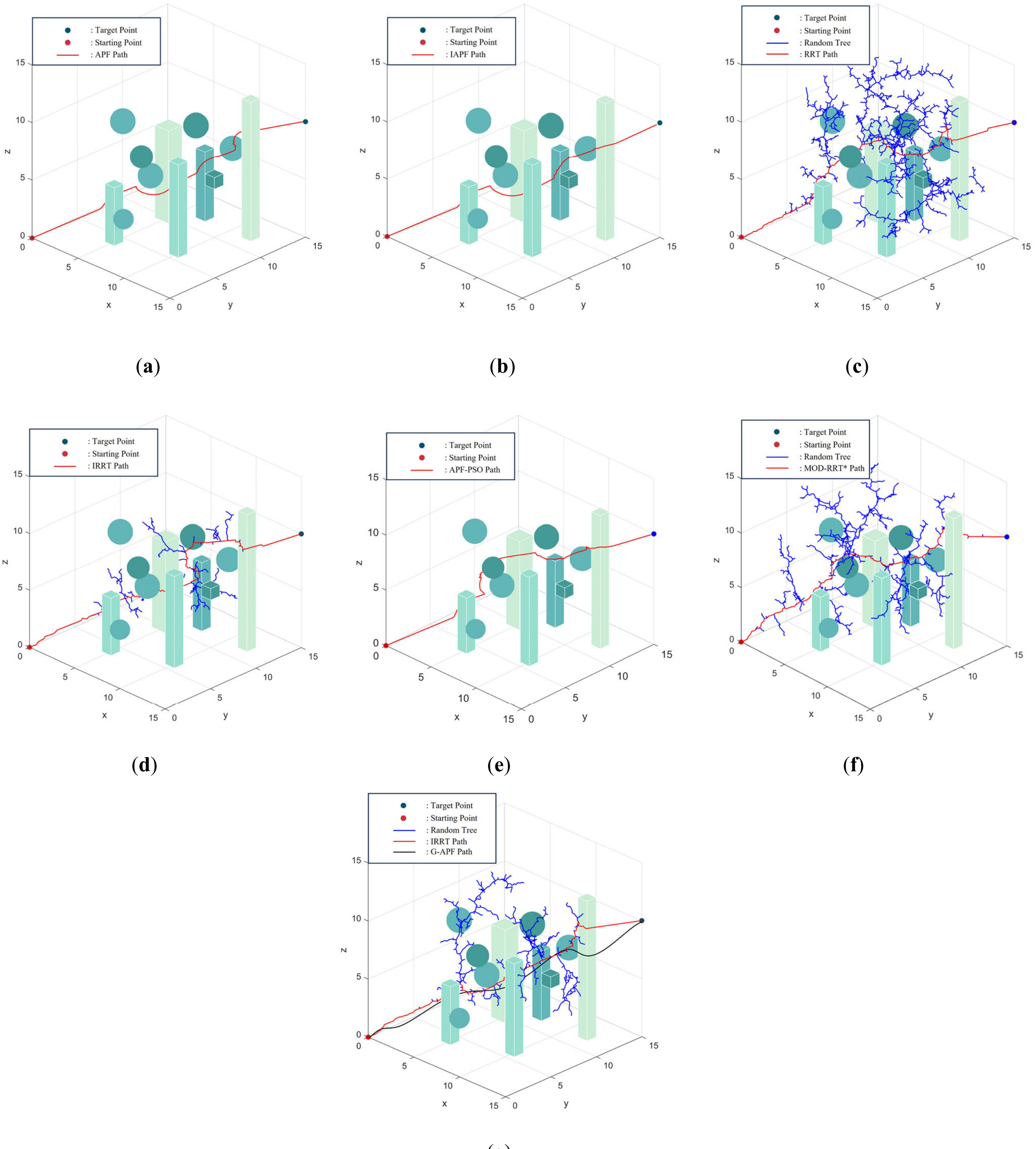
The APF-PSO algorithm lags in time performance compared to G-APF as it requires iterations to find the optimal value for the artificial potential field factor. Additionally, due to APF's sole reliance on potential field computation for optimal path determination, G-APF falls slightly short in terms of path length compared to APF-PSO. The MOD-RRT* algorithm introduces a multi-objective mechanism and employs RRT for path planning as prior knowledge. However, its performance, both in terms of time and path length, is slightly behind that of G-APF.

By combining the advantages of RRT and APF, the G-APF algorithm is able to effectively improve the timeliness of path planning and reduce path lengths. This hybrid approach benefits from the efficient exploration capabilities of RRT and the local optimization of APF, leading to more efficient and effective path planning in simple environments.

B. PATH PLANNING IN COMPLEX ENVIRONMENT

The simulation environment of path planning is shown in Figure 11(a-c), the size of the area is 30 m*30 m*30 m. The red point is the starting point with coordinates (0, 0, 0) and the blue point is the target point with coordinates (30, 25, 15). Other settings and parameter definitions are the same as those in the previous section.

Figure 12(a-f) depicts the path planning outcomes in diverse environmental scenarios, showing the comparative performance of the APF and IAPF algorithms. The characteristics are as follows:



(a)

(b)

(c)

(d)

(e)

(f)

(g)

FIGURE 9. Results of path planning: (a) APF; (b) IAPF; (c) RRT; (d) IRRT; (e) APF-PSO; (f) MOD-RRT*; (G) G-APF.

- In Figure 12(a), the APF algorithm encounters a local minimum, causing the path planning process to abruptly halt. However, in Figure 12(d), the IAPF algorithm effectively resolves the local minimum issue, allowing for continued path planning by dynamically adjusting the path around obstacles.
- Figure 12(b) illustrates a scenario in which the APF algorithm faces a target unreachability problem due to an obstacle obstructing the direct path to the target point. In contrast, Figure 12(e) demonstrates the successful resolution of this problem using the IAPF algorithm. By dynamically adjusting the gravitational

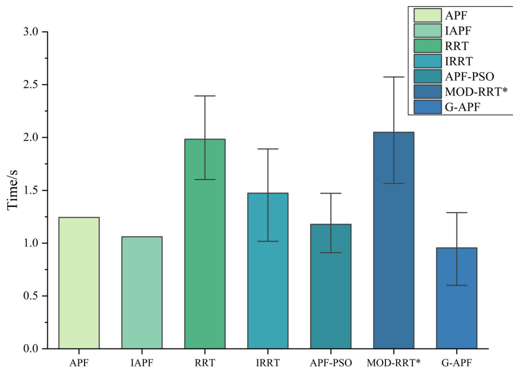


FIGURE 10. Time performance of each algorithm.

TABLE 5. Performance of each algorithm.

	Minimu m time(s)	Average time(s)	Maximu m time(s)	Average path length(m)
APF	Failure	Failure	Failure	Failure
IAPF	/	8.123	/	**33.706**
RRT	6.010	7.106	7.998	65.336
IRRT	5.052	5.998	6.996	64.785
APF-PSO	6.936	7.835	8.615	35.741
MOD-RRT*	3.684	4.459	5.157	44.724
G-APF	**1.716**	**2.989**	**3.970**	42.945

force to exceed the repulsive force near the obstacle, the IAPF enables the UAV to navigate closer to the target point. In addition, since the IAPF in Figure 12(e) does not incorporate a mechanism to improve path oscillations, it can be clearly seen that the problem of localized oscillations arises.

- Figure 12(c) shows a path planning scenario near an obstacle, which is prone to oscillations when using the APF algorithm. However, Figure 12(f) illustrates the improvement achieved by employing the IAPF algorithm, which effectively smooths out the path and mitigates oscillation-related issues.

These comparative results clearly highlight the advantages of the IAPF algorithm over the traditional APF method. The IAPF's ability to address local minimum, target unreachability, and oscillation problems contributes to smoother and more efficient path planning.

Figures 13(a-f) show the planning results for IAPF, APF-PSO, RRT, IRRT, MOD-RRT* and G-APF in the environment of Figure 11(a). Since APF-PSO introduces the PSO mechanism, it can effectively jump out of the local minimum to complete the planning. In contrast to IRRT, the tree expansion process employed by RRT is characterized by a higher level of complexity, while MOD-RRT* adopts a multi-objective mechanism to expand the tree more simply than IRRT. Building upon the foundation of IRRT, the G-APF algorithm further enhances path planning capabilities by incorporating path replanning techniques. By considering the influence of the potential field, G-APF aims to optimize and refine the path, resulting in improved trajectory planning outcomes. G-APF offers the potential to achieve better-quality paths and enhance overall path planning performance.

Table 5 presents the performance of each algorithm under 100 iterative trials in the environment depicted in Figure 10(a). The results demonstrate that the G-APF algorithm outperforms the other algorithms in terms of planning time.

Specifically, G-APF achieves a remarkable 63.2% reduction in average planning time compared to the IAPF. It also exhibits a 57.9% reduction in planning time compared to the RRT and a 50.2% reduction compared to the IRRT. The average planning time is also significantly reduced by 61.8% compared to the APF-PSO and 32.9% compared to MOD-RRT. However, when considering planning path lengths, the IAPF performs the best, producing the shortest path lengths among all the algorithms tested, followed by APF-PSO. On the other hand, the G-APF shows a 27.4% increase in path length compared to IAPF, a 20.1% increase compared to the APF-PSO, a 34.3% decrease compared to RRT, a 33.7% decrease compared to IRRT, and a 3.9% decrease compared to the MOD-RRT*.

Despite the fact that the average path length of G-APF is slightly longer compared to IAPF, G-APF outperforms algorithms like IAPF in terms of overall performance. This superiority stems from G-APF's exceptional timeliness. In UAV path planning, high timeliness is a critical performance indicator as it determines whether the UAV can swiftly plan a path to avoid obstacles. In fast decision-making and obstacle avoidance scenarios, the G-APF is able to provide timely and efficient path planning, even if this results in a slightly longer average path length. Ultimately, the ability to navigate safely and efficiently in dynamic environments takes precedence over path length considerations. Figure 14 visually illustrates the time performance of six algorithms.

As the complexity of the scene increases, the efficiency of the APF algorithm diminishes compared to the RRT algorithm. This is because APF needs to calculate the potential fields of all obstacles in the environment, which can be time-consuming. While APF excels at obstacle avoidance based on local information and often produces shorter path lengths than RRT, its computational burden becomes a limitation in complex environments where numerous obstacles need consideration.

On the other hand, RRT utilizes random sampling to explore paths towards the target point. Although these paths might not be optimal, RRT's search efficiency is significantly higher in complex scenes as it doesn't need to consider every obstacle explicitly.

By combining the strengths of both algorithms, it is possible to achieve excellent performance in complex environments. The optimization of the APF algorithm with RRT enables the efficient search capabilities of RRT to support the

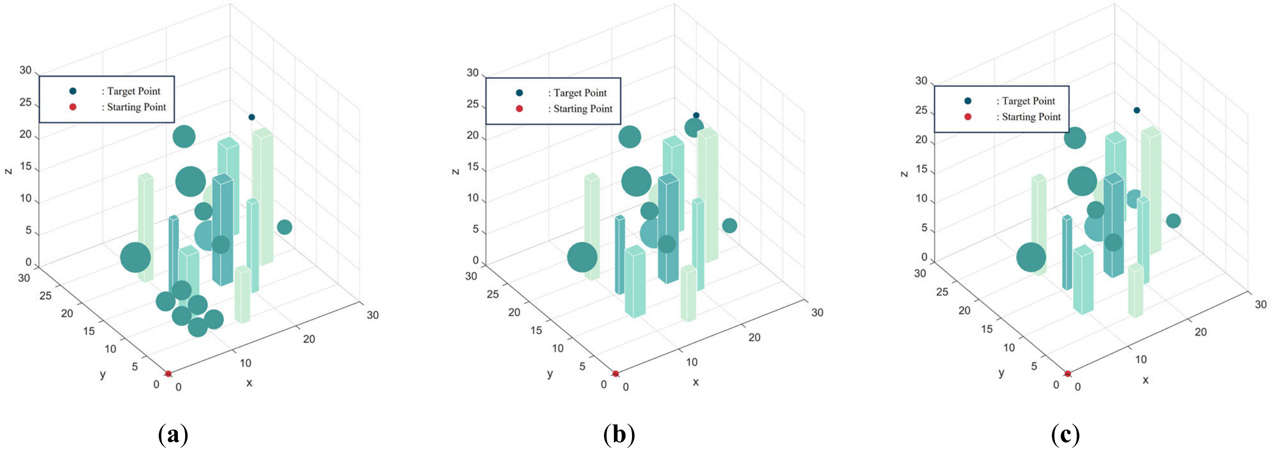


FIGURE 11. Complex simulation environments for path planning.

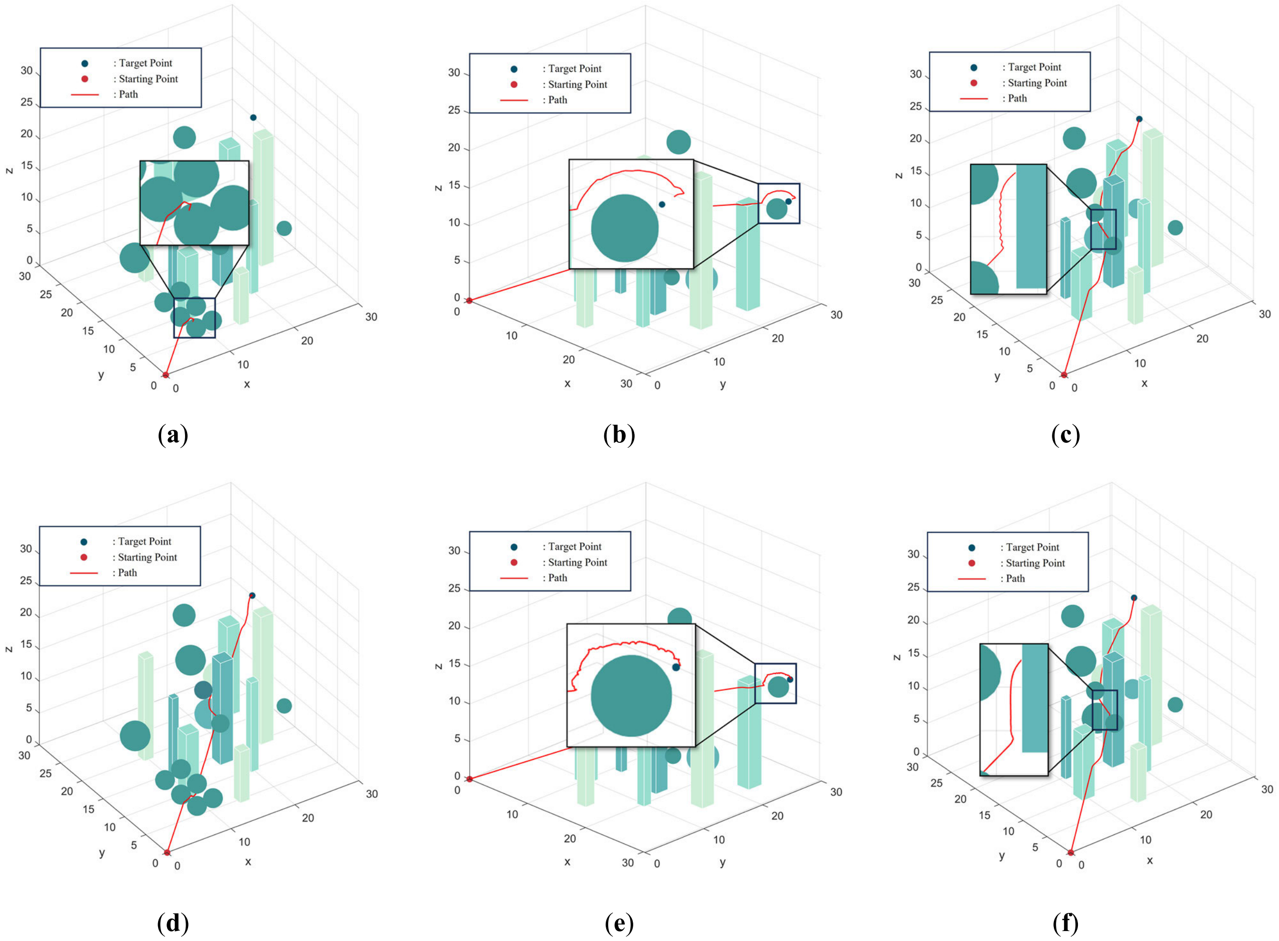
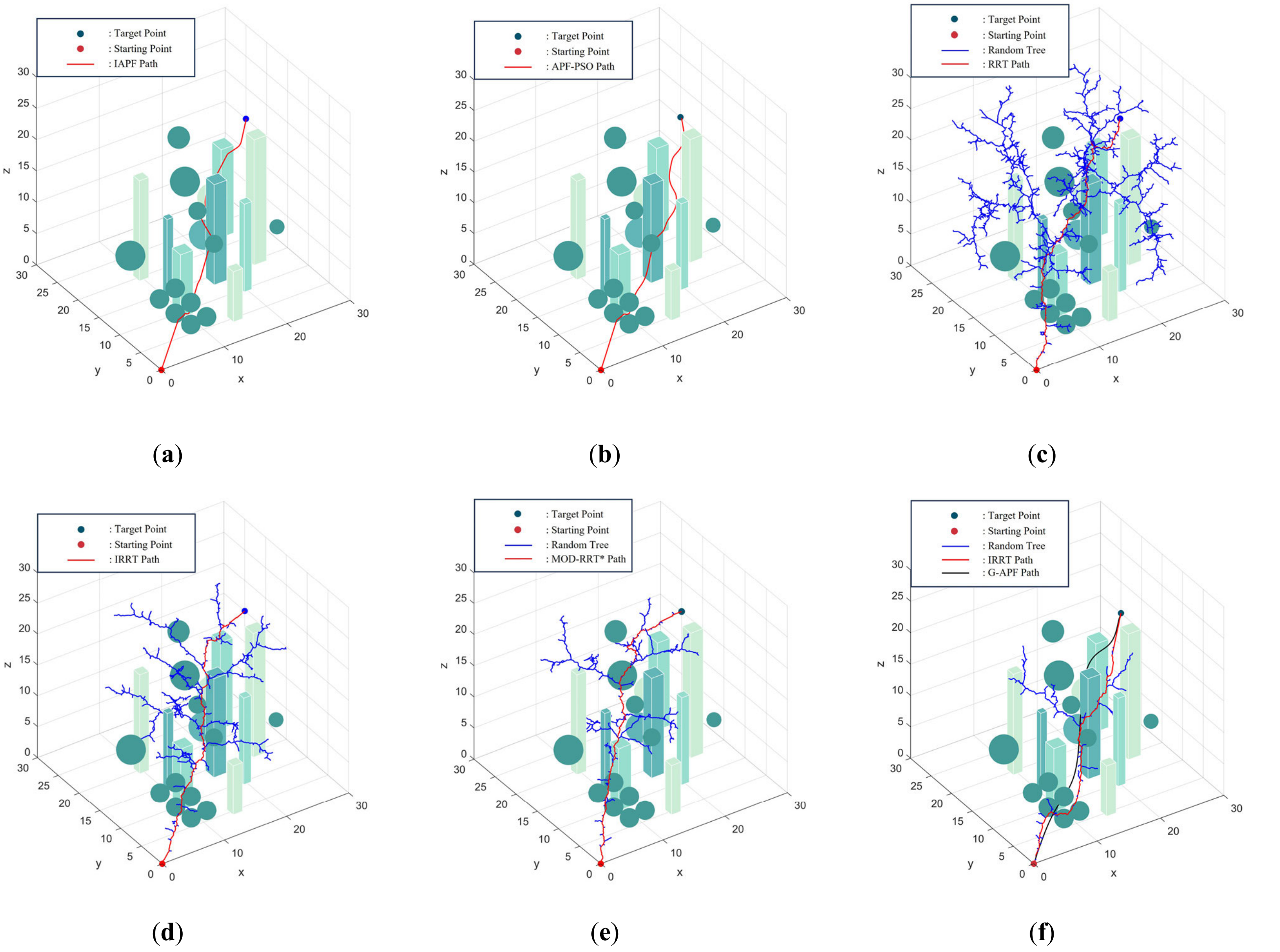


FIGURE 12. The planning situation of APF and IAPF in three different environments: (a, b, c) demonstrates that APF creates local minimum, target unreachability, and local path oscillations problems in a particular setting; (e, d, f) demonstrates that the IAPF method is effective in solving these problems alone, respectively.

path planning process of APF. This hybrid approach benefits from the optimal planning of APF in the local environment and the efficient exploration of RRT to navigate through complex scenarios. As a result, the combined method leads to improved path planning outcomes in challenging and intricate environments.



(a)

(b)

(c)

(d)

(e)

(f)

FIGURE 13. Results of path planning: (a) IAPF; (b) APF-PSO; (c) RRT; (d) IRRT; (e) MOD-RRT*; (f) G-APF.

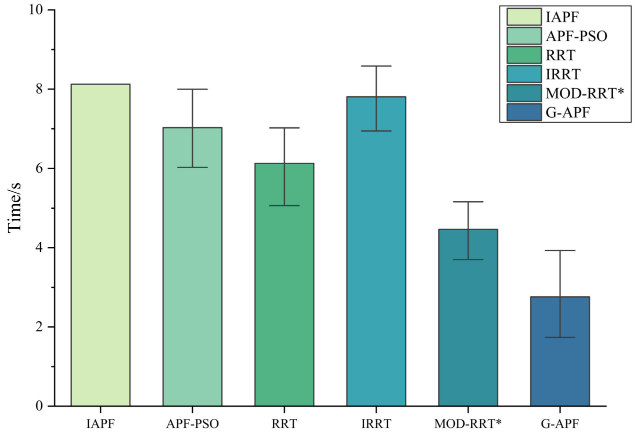


FIGURE 14. Time performance of the algorithm.

V. CONCLUSION

In this paper, we propose an efficient path planning algorithm called G-APF, which combines the strengths of the RRT and APF algorithms and integrates the idea of supervised learning based on environment awareness. The algorithm leverages supervised learning for object detection and creates an explicit environment model using an adaptive threat distance calculation module. By incorporating the APF potential field into the RRT for global path planning, the algorithm achieves faster tree search.

The introduction of environment awareness allows the algorithm to overcome the limitations of traditional methods that struggle to accurately perceive the environment. The ability to assign different threat distances to each object enhances the path planning process, leading to improved results.

During local path replanning, the algorithm optimizes the path based on the global path generated by the RRT. This approach addresses the problem of APF getting stuck in local minima. Furthermore, improvements to the potential field model and the introduction of a directional weighting factor overcome the issues of target unreachability and local path oscillations in the APF.

We conduct simulation experiments to evaluate the performance of G-APF in both simple and complex environments. The results demonstrate that G-APF outperforms APF in terms of planning time, reducing it by 23.1% in simple environments and overcoming APF's limitations in complex environments. Compared to IAPF, G-APF achieves planning time reductions ranging from 9.8% to 63.2%. Planning times are between 51.7% and 57.9% shorter than for RRT. Finally, compared to IRRT, planning time is reduced by 33.7% to 35.1%. In comparison with the APF-PSO and MOD-RRT∗, the G-APF also demonstrated its high performance. Compared to APF-PSO, planning time in complex environments was reduced by a staggering 61.8% with only a 20.1% increase in path length. Compared to MOD-RRT∗, the path length in complex environments was reduced by 3.9% and the planning time by 32.8%.

Overall, the proposed G-APF algorithm shows significant improvements over existing approaches and provides superior performance in both simple and complex environments. The algorithm demonstrates faster planning times and addresses key limitations of APF, making it a promising solution for efficient path planning.

The idea of G-APF is to introduce the APF potential field when RRT performs random sampling to guide the random sampling to move to the target point. In the follow-up work, in order to make G-APF get a shorter path, we can improve the random sampling strategy to be more precise. Guide the RRT exploration process in an intelligent way. The heuristic search algorithm can also be used to guide path search, which can better explore the optimal path while maintaining time efficiency. The method in this article will be deployed on UAVs in future work, and the feasibility of the method will be further verified through real experiments.

REFERENCES

[1] O. S. Oubbati, M. Atiquzzaman, T. A. Ahanger, and A. Ibrahim, "Softwarization of UAV networks: A survey of applications and future trends," *IEEE Access*, vol. 8, pp. 98073–98125, 2020.

[2] B. Alzahrani, O. S. Oubbati, A. Barnawi, M. Atiquzzaman, and D. Alghazzawi, "UAV assistance paradigm: State-of-the-art in applications and challenges," *J. Netw. Comput. Appl.*, vol. 166, Sep. 2020, Art. no. 102706.

[3] F. Nex and F. Remondino, "UAV for 3D mapping applications: A review," *Appl. Geomatics*, vol. 6, no. 1, pp. 1–15, 2014.

[4] S. Yang, X. Yang, and J. Mo, "The application of unmanned aircraft systems to plant protection in China," *Precis. Agricult.*, vol. 19, no. 2, pp. 278–292, Apr. 2018.

[5] H. Xiongkui, J. Bonds, A. Herbst, and J. Langenakens, "Recent development of unmanned aerial vehicle for plant protection in East Asia," *Int. J. Agricult. Biol. Eng.*, vol. 10, no. 3, pp. 18–30, 2017.

[6] W. Zhou, J. Li, Z. Liu, and L. Shen, "Improving multi-target cooperative tracking guidance for UAV swarms using multi-agent reinforcement learning," *Chin. J. Aeronaut.*, vol. 35, no. 7, pp. 100–112, Jul. 2022.

[7] K. Li, Y. Han, and X. Yan, "Distributed multi-UAV cooperation for dynamic target tracking optimized by an SAQPSO algorithm," *ISA Trans.*, vol. 129, pp. 230–242, Oct. 2022.

[8] Y. Yang, W. Wang, L. Liu, K. Dev, and N. M. F. Qureshi, "AoI optimization in the UAV-aided traffic monitoring network under attack: A Stackelberg game viewpoint," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 1, pp. 932–941, Jan. 2023.

[9] E. V. Butilă and R. G. Boboc, "Urban traffic monitoring and analysis using unmanned aerial vehicles (UAVs): A systematic literature review," *Remote Sens.*, vol. 14, no. 3, p. 620, Jan. 2022.

[10] Q. Diao, J. Zhang, M. Liu, and J. Yang, "A disaster relief UAV path planning based on APF-IRRT fusion algorithm," *Drones*, vol. 7, no. 5, p. 323, May 2023.

[11] S. H. Alsamhi, A. V. Shvetsov, S. Kumar, S. V. Shvetsova, M. A. Alhartomi, A. Hawbani, N. S. Rajput, S. Srivastava, A. Saif, and V. O. Nyangaresi, "UAV computing-assisted search and rescue mission framework for disaster and harsh environment mitigation," *Drones*, vol. 6, no. 7, p. 154, Jun. 2022.

[12] J. Keller, D. Thakur, M. Likhachev, J. Gallier, and V. Kumar, "Coordinated path planning for fixed-wing UAS conducting persistent surveillance missions," *IEEE Trans. Autom. Sci. Eng.*, vol. 14, no. 1, pp. 17–24, Jan. 2017.

[13] S. Aggarwal and N. Kumar, "Path planning techniques for unmanned aerial vehicles: A review, solutions, and challenges," *Comput. Commun.*, vol. 149, pp. 270–299, Jan. 2020.

[14] J. Fan, X. Chen, and X. Liang, "UAV trajectory planning based on bi-directional APF-RRT algorithm with goal-biased," *Exp. Syst. Appl.*, vol. 213, Mar. 2023, Art. no. 119137.

[15] A. Ait-Saadi, Y. Meraihi, A. Soukane, A. Ramdane-Cherif, and A. B. Gabis, "A novel hybrid chaotic Aquila optimization algorithm with simulated annealing for unmanned aerial vehicles path planning," *Comput. Electr. Eng.*, vol. 104, Dec. 2022, Art. no. 108461.

[16] B. Abhishek, S. Ranjit, T. Shankar, G. Eappen, P. Sivasankar, and A. Rajesh, "Hybrid PSO-HSA and PSO-GA algorithm for 3D path planning in autonomous UAVs," *Social Netw. Appl. Sci.*, vol. 2, pp. 1–16, Nov. 2020.

[17] H. Haghighi, S. H. Sadati, S. M. M. Dehghan, and J. Karimi, "Hybrid form of particle swarm optimization and genetic algorithm for optimal path planning in coverage mission by cooperated unmanned aerial vehicles," *J. Aerosp. Technol. Manag.*, vol. 12, Sep. 2020.

[18] Y. Liu and R. Bucknall, "Path planning algorithm for unmanned surface vehicle formations in a practical maritime environment," *Ocean Eng.*, vol. 97, pp. 126–144, Mar. 2015.

[19] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. Int. Conf. Neural Netw.*, vol. 4, 1995, pp. 1942–1948.

[20] M. Dorigo, V. Maniezzo, and A. Colorni, "Ant system: Optimization by a colony of cooperating agents," *IEEE Trans. Syst., Man, Cybern., B*, vol. 26, no. 1, pp. 29–41, Feb. 1996.

[21] S. Kirkpatrick and M. P. Vecchi, *Optimization by Simulated Annealing. Spin Glass Theory and Beyond: An Introduction to the Replica Method and Its Applications*. Singapore: World Scientific, 1987.

[22] X. L. Li, "An optimizing method based on autonomous animats: Fish-swarm algorithm," *Syst. Eng.-Theory Pract.*, vol. 22, no. 11, pp. 32–38, 2002.

[23] J. Tang, G. Liu, and Q. Pan, "A review on representative swarm intelligence algorithms for solving optimization problems: Applications and trends," *IEEE/CAA J. Autom. Sinica*, vol. 8, no. 10, pp. 1627–1643, Oct. 2021.

[24] P. Hart, N. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Trans. Syst. Sci. Cybern.*, vol. SSC-4, no. 2, pp. 100–107, Jul. 1968.

[25] *Edsger Wybe Dijkstra: His Life, Work, and Legacy*. New York, NY, USA: Association for Computing Machinery, Jul. 2022, pp. 287–290.

[26] O. Khatib, "The potential field approach and operational space formulation in robot control," in *Adaptive and Learning Systems: Theory and Applications*, K. S. Narendra, Ed. Boston, MA, USA: Springer, 1986, pp. 367–377.

[27] C. Qixin, H. Yanwen, and Z. Jingliang, "An evolutionary artificial potential field algorithm for dynamic path planning of mobile robot," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2006, pp. 3331–3336.

[28] J. Agirrebeitia, R. Avilés, I. F. de Bustos, and G. Ajuria, "A new APF strategy for path planning in environments with obstacles," *Mechanism Mach. Theory*, vol. 40, no. 6, pp. 645–658, Jun. 2005.

[29] T. Huang, D. Huang, N. Qin, and Y. Li, "Path planning and control of a quadrotor UAV based on an improved APF using parallel search," *Int. J. Aerosp. Eng.*, vol. 2021, pp. 1–14, Jul. 2021.

[30] Z. Lin, M. Yue, G. Chen, and J. Sun, "Path planning of mobile robot with PSO-based APF and fuzzy-based DWA subject to moving obstacles," *Trans. Inst. Meas. Control*, vol. 44, no. 1, pp. 121–132, Jan. 2022.

[31] Y. Liu, J. Qi, M. Wang, C. Wu, and H. Sun, "Path planning for large-scale UAV formation based on improved SA-APF algorithm," in *Proc. 41st Chin. Control Conf. (CCC)*, Jul. 2022, pp. 4472–4478.

[32] Y. Chen, G. Bai, Y. Zhan, X. Hu, and J. Liu, "Path planning and obstacle avoiding of the USV based on improved ACO-APF hybrid algorithm with adaptive early-warning," *IEEE Access*, vol. 9, pp. 40728–40742, 2021.

[33] X. Wu, X. Long, S. Yuan, Q. Hu, and P. Xie, "Multi-UUV coordinated path planning with collision avoidance (CPP/CA) based on combination of improved APF and A?" in *Proc. 8th Int. Conf. Control, Autom. Robot. (ICCAR)*, Apr. 2022, pp. 218–223.

[34] Q. Su, S. Ma, L. Wang, Y. Song, H. Wang, B. Li, and Y. Yang, "Artificial potential field guided JPS algorithm for fast optimal path planning in cluttered environments," *J. Brazilian Soc. Mech. Sci. Eng.*, vol. 44, no. 12, p. 602, Dec. 2022.

[35] C. Zammit and E.-J. van Kampen, "Comparison between A* and RRT algorithms for 3D UAV path planning," *Unmanned Syst.*, vol. 10, no. 2, pp. 129–146, Apr. 2022.
[36] S. LaValle, "Rapidly-exploring random trees: A new tool for path planning," Res. Rep. 9811, 1998.
[37] E. Arnold, O. Y. Al-Jarrah, M. Dianati, S. Fallah, D. Oxtoby, and A. Mouzakitis, "A survey on 3D object detection methods for autonomous driving applications," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 10, pp. 3782–3795, Oct. 2019.
[38] X.-H. Liu, D.-G. Zhang, H.-R. Yan, Y.-Y. Cui, and L. Chen, "A new algorithm of the best path selection based on machine learning," *IEEE Access*, vol. 7, pp. 126913–126928, 2019.
[39] A. Gupta, A. Anpalagan, L. Guan, and A. S. Khwaja, "Deep learning for object detection and scene perception in self-driving cars: Survey, challenges, and open issues," *Array*, vol. 10, Jul. 2021, Art. no. 100057.
[40] L. Yang, J. Qi, J. Xiao, and X. Yong, "A literature review of UAV 3D path planning," in *Proc. 11th World Congr. Intell. Control Autom.*, Jun. 2014, pp. 2376–2381.
[41] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788.
[42] D. Du, P. Zhu, L. Wen, X. Bian, H. Lin, Q. Hu, T. Peng, J. Zheng, X. Wang, Y. Zhang, and L. Bo, "VisDrone-DET2019: The vision meets drone object detection in image challenge results," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshop (ICCVW)*, Oct. 2019, pp. 213–226.
[43] Y. Zhao, K. Liu, G. Lu, Y. Hu, and S. Yuan, "Path planning of UAV delivery based on improved APF-RRT algorithm," *J. Phys., Conf.*, vol. 1624, no. 4, Oct. 2020, Art. no. 042004.
[44] J. Qi, H. Yang, and H. Sun, "MOD-RRT: A sampling-based algorithm for robot path planning in dynamic environment," *IEEE Trans. Ind. Electron.*, vol. 68, no. 8, pp. 7244–7251, Aug. 2021.

JILONG LIU received the B.S. degree from the School of Electronic Engineering, Chengdu Technological University, Chengdu, China, in 2022. He is currently pursuing the M.S. degree with the Sichuan University of Science and Engineering, Yibin, China. His main research interests include UAV path planning and obstacle avoidance.

YUEHAO YAN received the B.S. and M.S. degrees in computer science and software engineering from the University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 2001 and 2008, respectively, and the Ph.D. degree in advanced manufacturing (aviation) from Northwestern Polytechnical University, China, in 2022. He is currently a Professor with the School of Automation and Electronic Information, Sichuan University of Science and Engineering (SUSE). His research interests include UAV obstacle avoidance and UAV navigation and control.

YUNHONG YANG received the B.S. degree in communication engineering from the Sichuan University of Science and Engineering, Yibin, China, in 2021, where he is currently pursuing the M.S. degree. His main research interests include UAV formation and path planning.

JUNLIN LI received the B.S. degree in Internet of Things engineering from the Sichuan University of Arts and Science, Dazhou, China, in 2020. He is currently pursuing the M.S. degree with the Sichuan University of Science and Engineering, Yibin, China. His main research interests include UAV path planning and obstacle avoidance.