Multi-Quadrotor Cooperative Encirclement and Capture Approach in Obstacle Environments

Zhenyu Li*
School of Mechanical Engineering
Zhejiang University
Hangzhou, 310058, China
lizy9918@zju.edu.cn

Ruixin Wu*
School of Mechanical Engineering
Zhejiang University
Hangzhou, 310058, China
22325053@zju.edu.cn

Jin Wang†
School of Mechanical Engineering
Zhejiang University
Hangzhou, 310058, China
dwjcom@zju.edu.cn

Chuanqi Hu
Polytechnic Institute
Zhejiang University
Hangzhou, 310058, China
fanxingml@zju.edu.cn

Zhi Zheng
School of Mechanical Engineering
Zhejiang University
Hangzhou, 310058, China
z.z@zju.edu.cn

Jun Meng
College of
Electrical Engineering
Zhejiang University
Hangzhou, 310027, China
junmeng@zju.edu.cn

Guodong Lu
School of Mechanical Engineering
Zhejiang University
Hangzhou, 310058, China
lugd@zju.edu.cn

***Abstract*—This article proposes an algorithm designed for autonomous quadrotors to capture target quadrotors in complex obstacle environments, enabling multiple pursuers to perform distributed capture while simultaneously avoiding obstacles and collisions. We first introduce a 3D obstacle space representation method based on safety-priority reachable region (SPRR) and implement real-time construction for general point cloud maps. Subsequently, leveraging SPRR, we employ an area minimization approach to ensure the effective capture of the evader within a limited time. The outcomes are then integrated into a MINCO-based trajectory planning framework to ensure their practical execution by real quadrotors. Relying on SPRR, the optimization problem is simplified and the solution efficiency is improved. In the scenario of simulating the dynamics of real quadrotors, we compare the proposed algorithm with state-of-the-art evasion and pursuit methods, validating its effectiveness and advancements. This work presents a practical solution for multi-quadrotor collaborative capture.**

***Index Terms*—multi-quadrotor, collaborative capture, trajectory planning.**

I. Introduction

Multi-quadrotor collaborative capture involves coordinating multiple quadrotors to track and encircle one or more evading targets using coordinated control strategies to prevent escape. Fundamentally, this represents a multirobot pursuit-evasion (MPE) game, which has gained significant attention due to its practical applications, including area searching and monitoring [1], and target tracking [2].

A classical approach to solving the multi-agent collaborative capture problem relies on differential equations, specifically differential games based on the Hamilton-Jacobi-Isaacs (HJI) equation, which provide optimal control strategies for efficient capture in complex environments [2] [3] [4]. Another formulation of the MPE problem is based on intelligent methods. These methods leverage reinforcement learning to train cooperative pursuit strategies, enabling quadrotors to encircle and approach evading targets [5] [6]. They also allow pursuers to learn the evader's escape strategies and adapt their pursuit strategies, ensuring that the cooperative strategy cannot be exploitable by the evader [7] [8]. These methods improve capture efficiency through innovative pursuit algorithms, ensuring robust collaboration.

However, due to the difficulty in obtaining analytical solutions to HJI partial differential equations and the curse of dimensionality in numerical solutions, differential game methods face significant limitations in large-scale, high-dimensional scenarios. Similarly, reinforcement learning-based methods encounter high computational complexity and challenges related to environmental adaptability. In contrast, geometric control methods, by simplifying the model, demonstrate higher efficiency in handling multi-quadrotor collaboration and collision avoidance, offering superior real-time performance.

Advanced studies on geometric control primarily focus on pursuing the evader within a bounded environment, where the evader often needs to be forced into a corner before it can be captured [9] [10]. Pierson et al. developed a distributed algorithm based on Voronoi partitioning and Voronoi area minimization strategies to ensure that multiple quadrotors can effectively capture an evader in a limited time [11]. Tian et al. proposed a distributed algorithm using Voronoi diagrams, dynamically adjusting each pursuer's safe region to ensure

This work was supported in part by the National Natural Science Foundation of China under Grant 52475033, in part by the "Pioneer" and "Leading Goose" R&D Program of Zhejiang under Grant 2024C01170 and Grant 2023C01070, and in part by the Robotics Institute of Zhejiang University under Grant K12107 and Grant K11805. (**These authors contributed to the work equally.*)

Zhenyu Li, Ruixin Wu, Jin Wang, Chuanqi Hu, Zhi Zheng and Guodong Lu are with State Key Laboratory of Fluid Power and Mechatronic Systems, Zhejiang University, Hangzhou 310058, China, also with Zhejiang Key Laboratory of Industrial Big Data and Robot Intelligent Systems, Zhejiang University, Hangzhou 310058, China, and also with Robotics Research Center of Yuyao City, Ningbo 315400, China.

Jun Meng is with Center for Data Mining and Systems Biology, College of Electrical Engineering, Zhejiang University, Hangzhou 310027, China.

†Corresponding author: Jin Wang.

efficient cooperative capture in both obstacle-free and cluttered environments [12]. Wang et al. introduced a method utilizing B-ECBVC (Buffered Evader-Centered Bounded Voronoi Cells) to generate safe regions, aiding multiple pursuers in capturing evaders in complex environments [13]. In follow-up studies, Wang et al. further proposed the Obstacle-Aware B-ECBVC method, which effectively avoids collisions and ensures pursuit efficiency in dynamic environments [14].

These geometry-based control methods perform well in 2D scenarios but face limitations when applied to quadrotor systems that rely on visual obstacle perception. Furthermore, these methods often ignore the dynamics and kinematic constraints of real quadrotors, which can negatively impact their performance in practical scenarios.

To address these issues, this article proposes an safety-priority reachable region (SPRR)-based approach for multi-quadrotor cooperative encirclement and capture. The proposed method can adapt to the 3D flight environment and fully consider the dynamics and kinematic constraints of quadrotor to improve the feasibility and efficiency in practical applications.The main contributions are summarized as follows.

1)**Fast and Efficient Strategy Generation**: By combining Voronoi partition with obstacle-aware safety regions, we construct SPRR for effective 3D obstacle space representation. To enhance practicality and real-time performance, we also implement a rapid SPRR construction on point cloud maps. Finally, an area minimization strategy is applied for efficient encirclement.;

2)**Trajectory Planning Based on MINCO**: To address the challenge that current research often cannot be directly applied to quadrotor systems, we integrate our strategy with the advanced trajectory planning framework MINCO [15]. This allows for spatiotemporal joint optimization of the quadrotor trajectory, ensuring safety and effectiveness while enabling the quadrotor to fully execute the cooperative capture strategy;

3)**Experimental Validation**: We conduct comprehensive simulation experiments to validate the effectiveness and advancement of our proposed method, demonstrates its ability to rapidly and effectively achieve encirclement in a quadrotor system based on visual obstacle perception.

The remainder of this article is organized as follows. In Section II, the problem formulation is provided. Section III first presents the construction of SPRR, then proposed SPRR based capture strategy. Section IV presents trajectory planning for executing cooperative capture strategy. Section V shows experiments on the effectiveness and advancement of the proposed method. Finally, Section VI concludes this article.

II. PROBLEM FORMULATION

Consider an MPE problem in a bounded convex environment $Q \subset \mathbb{R}^3$, involving N pursuers and a single evader, with static obstacles. Let $x_i, x_e \in Q$ for $i \in \Gamma = \{1, \ldots, N\}$ be the position of pursuer and evader, respectively. Obstacle regions are denoted as Q_o, where $o \in \Phi = \{1, \ldots, m\}$. The dynamics of each agents are typically modeled as integrators [13].

$$\begin{aligned} &\dot{x}_e = v_e, \|v_e\| \leq v_{e,\text{ max}} \\ &\dot{x}_i = v_i, \|v_i\| \leq v_{p,\max} \forall i \in \Gamma \end{aligned} \tag{1}$$

where v_e and v_i are the velocities of evader and pursuers, respectively. We assume $v_{e,\max} = v_{p,\text{ max}}$ to model scenarios where the pursuers do not possess speed advantage.

The evader's policy is unknown. The pursuers' objective is to employ teamwork to corner the evader and capture it within a limited time, ensuring that at least one pursuer enters the capture radius r_c. The evader is captured at time t_c when

$$\mathcal{D}_c = \min_{i \in \Gamma} \|x_i(t_c) - x_e(t_c)\| < r_c \tag{2}$$

Safety is ensured by maintaining a minimum distance between quadrotors and obstacles, expressed as:

$$\begin{aligned} &\min_{i,j \in \Gamma, i \neq j} \|x_i(t) - x_j(t)\| > 2r_s \text{ and } x_i(t) \notin Q_o \\ &\text{for } \forall t \in [0, t_c], \forall i \in \Gamma \text{ and } \forall o \in \Phi \end{aligned} \tag{3}$$

The MPE problem is formally defined as follows::

Definition 1(MPE Problem): Given a random initial configuration $x_i(0), x_e(0) \in Q$ with $\mathcal{D}_c(0) > r_c$, find a cooperative control law v_i for each pursuer i such that $\mathcal{D}_c(t_c) \leq r_c$ for some $t_c < \infty$ and safety constraints(3) are satisfied for all t.

III. COOPERATIVE CAPTURE STRATEGY

This section addresses solving the MPE problem outlined in *Definition 1*, focusing on devising control strategies for the pursuers to capture the evader in a bounded convex environment with obstacles, while ensuring safety constraints are met. Due to the fact that such an environment is typical non-convex, directly modeling and solving strategies is challenging. To address this, we simplify the 3D obstacle space by constructing Safety-Priority Reachable Region (SPRR).The pursuer can adopt the SPRR-based area minimizing capture strategy to effectively capture the evader.

A. The Construction of Priority Reachable Region

Although the evader's motion strategy is dynamic and unpredictable, the effective capture can be achieved by reducing the evader's priority reachable region(PRR), which is defined as the set of positions within Q that the evader can reach faster than any other quadrotor [11]. In an obstacle-free environment, for agents that follow single-integrator kinematic model with equal maximum velocities, such a region can be represented by Voronoi partition:

$$\begin{aligned} &\mathcal{V}_e = \{x \in Q \mid \|x - x_e\| \leq \|x - x_i\|, \forall i \in \Gamma\} \\ &\mathcal{V}_i = \{x \in Q \mid \|x - x_i\| \leq \|x - x_j\|, \forall j \neq i, i, j \in \Gamma\} \end{aligned} \tag{4}$$

where $\mathcal{V}_i$ and $\mathcal{V}_e$ represent the PRR of pursuers and the evader in an obstacle-free environment, respectively.

Based on the hyperplane separation rule [16], the Voronoi partition of each quadrotor can be represented by a set of hyperplanes, which divide $Q \subset \mathbb{R}^3$ into two parts, inside and outside part of the plane. As shown in Fig.1, the combination

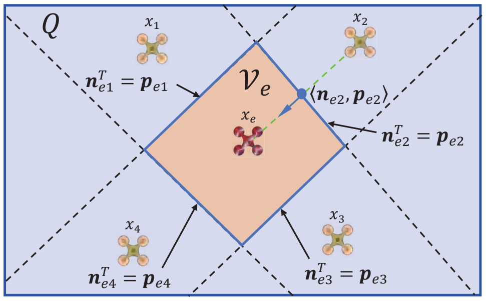
Q
x_1
x_2
$\mathcal{V}_e$
$\boldsymbol{n}_{e1}^T = \boldsymbol{p}_{e1}$
$\langle \boldsymbol{n}_{e2}, \boldsymbol{p}_{e2} \rangle$
x_e
$\boldsymbol{n}_{e2}^T = \boldsymbol{p}_{e2}$
x_4
x_3
$\boldsymbol{n}_{e4}^T = \boldsymbol{p}_{e4}$
$\boldsymbol{n}_{e3}^T = \boldsymbol{p}_{e3}$

Fig. 1: Priority Reachable Region (PRR).Dashed lines indicate hyperplanes separating each region. The illustration is in 2D for clarity, with the 3D construction being analogous.

of the inside part of multiple hyperplanes can form a convex region. Then, the definition of the PRR can be transformed from (4) into the following form:

$$\begin{gathered}\mathcal{V}_e = \left\{ \mathbf{X} \in Q \mid \mathbf{n}_{ei}^{\mathrm{T}} \mathbf{X} \le \boldsymbol{p}_{ei} \right\} \\ \mathcal{V}_i = \left\{ \mathbf{X} \in Q \mid \mathbf{n}_{ij}^{\mathrm{T}} \mathbf{X} \le \boldsymbol{p}_{ij}, \forall j \neq i, i, j \in \Gamma \right. \\ \left. \text{and } \; \mathbf{n}_{ie}^{\mathrm{T}} \mathbf{X} \le \boldsymbol{p}_{ie} \right\}\end{gathered} \tag{5}$$

where $\mathbf{X}$ denotes the vector from the origin to any point x in Q, $\langle \mathbf{n}_{l_1 l_2}, \boldsymbol{p}_{l_1 l_2} \rangle$ with $l_1, l_2 \in \{\Gamma, e\}$ and $l_1 \neq l_2$ denote the parameters of the hyperplane. The hyperplane normal vector $\mathbf{n}$ and point on the plane $\boldsymbol{p}$ can be calculated by as follow:

$$\mathbf{n}_{l_1 l_2} = \mathbf{X}_{l_1 l_2} = \mathbf{X}_{l_1} - \mathbf{X}_{l_2}, \boldsymbol{p}_{l_1 \kappa_2} = \mathbf{X}_{l_1 l_2}^{\mathrm{T}} \frac{\mathbf{X}_{l_1} + \mathbf{X}_{l_2}}{2} \tag{6}$$

B. The Construction of Safety-Priority Reachable Region

Different with previous studies [11], which focused on obstacle-free scenarios, our work considers environments with obstacles. Building upon the PRR, we further construct SPRR based on obstacle perception.

The separating hyperplane in (5) ensures collision avoidance between multiple quadrotors, so we only need to focus on obstacle avoidance during the capture phase to guarantee safety. To enable real-time obstacle detection and avoidance, we represent the quadrotor's SPRR as a convex polyhedron formed by multiple separating hyperplanes between the quadrotor and obstacles.

$$\mathcal{V}_l = \left\{ \mathbf{X} \in Q \mid \sum_{h=1}^{H} \left(\boldsymbol{n}_{lo}^{h} \right)^{\mathrm{T}} \mathbf{X} \le \boldsymbol{p}_{lo}^{h}, l \in (\Gamma, e), o \in \Phi \right\} \tag{7}$$

where $\mathcal{V}_l$ represent the safety reachable region of quadrotor $l \in (\Gamma, e)$. $\langle \boldsymbol{n}_{lo}^{h}, \boldsymbol{p}_{lo}^{h} \rangle, h = \{1, \ldots, H\}$ defined the set of parameters for the hyperplanes forming $\mathcal{V}_l$. Note that the number of hyperplanes is not fixed, as it depends on the relative position of obstacle region and the quadrotor. Different with other studies that assume obstacles are strictly convex regions with known vertices [14], we directly calculate the separating hyperplane parameters based on the point cloud map Ω_obs constructed by the quadrotor's visual perception system. The algorithm is outlined in Algorithm 1.

Algorithm 1 Construction of SRR Based on Point Cloud Map

```
Input: $x_p \in Q, r_s, \Omega\_obs$
Output: $\mathcal{V}_l$
for $p = 1$ to $N$ and $e$ do
    $\Omega$_remain = $\Omega\_obs$;
    while PointCloudNum ($\Omega$_remain ) $> \xi$ do
        Find $o\_near$: the nearest obstacle point to $x_p$;
        Calculate the parameters $\langle \boldsymbol{n}_{po}^{h}, \boldsymbol{p}_{po}^{h} \rangle$ using (8);
        Update $\Omega$_remain: remove point clouds outside the hyperplane;
    end while
    Add $\langle \boldsymbol{n}_{po}^{\hbar}, \boldsymbol{p}_{po}^{\hbar} \rangle$ to the set of hyperplane parameters $\mathcal{V}_l$.
end for
```

To ensure real-time strategy generation, we accelerate the search process by using the KD-Tree for rapid retrieval of the nearest obstacle point o_near in the point cloud map. The parameters$\langle \boldsymbol{n}_{lo}^{h}, \boldsymbol{p}_{lo}^{h} \rangle$can be expressed as follows:

$$\begin{aligned}\boldsymbol{n}_{lo}^{h} &= \mathbf{X}_{lo\_near} = \mathbf{X}_l - \mathbf{X}_{o\_near} \\ \boldsymbol{p}_{lo}^{h} &= \mathbf{X}_{o\_near} + r_s \frac{\mathbf{X}_{lo\_\mathrm{near}}}{\|\mathbf{X}_{lo\_\mathrm{near}}\|}\end{aligned} \tag{8}$$

Considering the quadrotor's physical size, we ensure effective obstacle avoidance in practical scenarios by offsetting the hyperplane away from the obstacle by a safety radius r_s. As shown in Fig.2, by combining (5) and (7), the final representation of the SPRR is defined as follows:

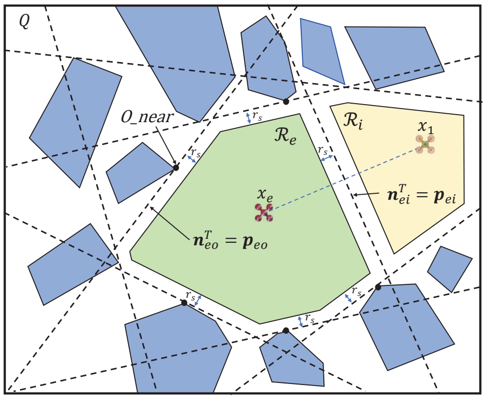
Q
O_near
r_s
$\mathcal{R}_e$
$\mathcal{R}_i$
x_1
r_s
x_e
$\boldsymbol{n}_{ei}^T = \boldsymbol{p}_{ei}$
$\boldsymbol{n}_{eo}^T = \boldsymbol{p}_{eo}$
r_s

Fig. 2: Safety-Priority Reachable Region (SPRR).

$$\begin{aligned}\mathcal{R}_e &= \left\{ \mathbf{X} \in Q \mid \mathbf{n}^{\mathrm{T}} \mathbf{X} \le \boldsymbol{p}, \langle \mathbf{n}, \boldsymbol{p} \rangle \in \{ \langle \mathbf{n}_{ei}, \boldsymbol{p}_{ei} \rangle, \langle \mathbf{n}_{eo}, \boldsymbol{p}_{eo} \rangle \} \right\} \\ \mathcal{R}_i &= \left\{ \mathbf{X} \in Q \mid \mathbf{n}^{\mathrm{T}} \mathbf{X} \le \boldsymbol{p}, \langle \mathbf{n}, \boldsymbol{p} \rangle \in \left\{ \begin{array}{c} \langle \mathbf{n}_{ie}, \boldsymbol{p}_{ie} \rangle, \langle \mathbf{n}_{ij}, \boldsymbol{p}_{ij} \rangle \\ \langle \mathbf{n}_{io}, \boldsymbol{p}_{io} \rangle \end{array} \right\} \right\}\end{aligned} \tag{9}$$

C. SPRR-Based Area Minimizing Capture Strategy

This section presents the capture strategy from [17], which minimizes the evader's activity space. However, we generalize it to 3D obstacle environments by leveraging the SPRR.

The area of the region that evader can reach faster than other quadrotors is defined as $\forall_e = \int_{\mathcal{R}_e} dq$. According to the definition in (9), the dynamics of $\forall_e$ are:

$$\dot{\forall}_e = \frac{\partial \forall_e}{\partial x_e} \dot{x}_e + \sum_{i=1}^{N} \frac{\partial \forall_e}{\partial x_i} \dot{x}_i, i \in \Gamma \tag{10}$$

It is obvious that $\dot{\forall}_e$ is determined by both the evader's motion strategy $\dot{x}_e$ and the pursuer's strategy $\dot{x}_i$. The evader's motion strategy is unpredictable to the pursuer, but the pursuer can implement a strategy that gradually reduces evader's SRPP area $\forall_e$ over time to achieve capture. We can decouple the pursuer's terms in (10), isolating them as $\frac{\partial \forall_e}{\partial x_i} \dot{x}_i$.

After decoupling, the pursuer can follow the motion strategy in (11), which aligns with the gradient descent direction of $\dot{\forall}_e$, ensuring the fastest reduction of the evader's SPRR.

$$v_i = \dot{x}_i = -\frac{\frac{\partial \forall_e}{\partial x_i}}{\left\| \frac{\partial \forall_e}{\partial x_i} \right\|} \tag{11}$$

Next, we will derive how to calculate this strategy. Using the Leibniz Integral Rule, $\dot{\forall}_e$ can be simplified as:

$$\begin{aligned} \dot{\forall}_e &= \sum_{i \in N_s} \int_{s_i} \left[\frac{(x_i - q)^T \dot{x}_i}{\|x_i - x_e\|} - \frac{(x_e - q)^T \dot{x}_e}{\|x_i - x_e\|} \right] dq \\ &= \sum_{i \in N_s} \frac{A_i (x_i - C_{s_i})^T}{\|x_i - x_e\|} \dot{x}_i - \sum_{i \in N_s} \frac{A_i (x_e - C_{s_i})^T}{\|x_i - x_e\|} \dot{x}_e \end{aligned} \tag{12}$$

where N_s denotes the set of pursuers i sharing a boundary with the evader e, and s_i represents the shared boundary. The area of shared boundary is described as $A_i = \int_{s_i} dq$, and the centroid of shared boundary is $C_{s_i} = \frac{1}{A_i} \int_{s_i} q dq$.

By comparing (10) and (12), it is evident that $\frac{\partial \forall_e}{\partial x_i} = \frac{\dot{\forall}_e (x_i - C_{s_i})^T}{\|x_i - x_e\|}$. Substituting this conclusion into (11), the final capture strategy for the pursuer becomes:

$$v_i = \frac{C_{s_i} - x_i}{\|C_{s_i} - x_i\|} \tag{13}$$

Physically, this strategy directs the pursuer towards the centroid of shared boundary with the evader.

IV. TRAJECTORY PLANNING

In this section, we mainly discuss how to effectively execute capture strategy on real quadrotor. Our trajectory planning consists of two main parts, namely front-end path search based on Hybrid-A* and back-end trajectory optimization based on MINCO [15]. Guided by cooperative capture strategy in (13), a safe and smooth executable trajectory that satisfies the quadrotor's dynamics is ultimately generated.

A. Hybrid-A Based Path Search*

To find an executable trajectory from the current position to the target point in (13), we use the hybrid-A* [18] for front-end path search. The framework is similar to A*, but straight-line edges are replaced by motion primitives that represent quadrotor dynamics. Due to space constraints, we only discuss the motion primitive generation and penalty-heuristic design in hybrid-A*, without repeating the classic A* framework.

Given the initial state $x(0)$ and control input $u(t)$, the trajectory is determined. In hybrid-A*, each node's initial state is fixed, and with a discretized control input set $U_D \subset U$ and duration τ, each axis $[-u_{\max}, u_{\max}]$ is discretized into $\{-u_{\max}, -(r- \ 1) \ /r u_{\max}, \ldots, (r-1)/r u_{\max}, u_{\max}\}$, generating $(2r+1)^3$ motion primitives per state node.

To minimize trajectory time and control effort, we define the cost function as:

$$J(T) = \int_0^T \|u(t)\|^2 dt + \rho T \tag{14}$$

In the A* graph, the edge cost from discrete control input u_d and duration τ is $e_c = \left(\|u_d\|^2 + \rho \right) \tau$. The total cost from the start to the current state is $g_c = \sum_{k=1}^{k} \left(\|u_{d_k}\|^2 + \rho \right) \tau$.

We design the heuristic using Pontryagin's minimum principle to compute the optimal trajectory. The cost function is defined as $J^*(T) = \sum_{\mu \in \{x,y,z\}} \left(\frac{1}{3} \alpha_\mu^2 T^3 + \alpha_\mu \beta_\mu T^2 + \beta_\mu^2 T \right)$.

To minimize $J^*(T)$, we solve for T_h using $\frac{\partial J^*(T)}{\partial T} = 0$, and use $J^*(T_h)$ as the heuristic. The total cost is $f_c = g_c + h_c = \sum_{k=1}^{k} \left(\|u_{d_k}\|^2 + \rho \right) \tau + J^*(T_h)$.

In summary, hybrid-A* improves upon A* by considering dynamics in front-end path search, offering a better initial guess for back-end trajectory optimization.

B. MINCO Based Trajectory Optimization

The basic requirements for the planned trajectory $x(t)$ include dynamic feasibility and safety. Meanwhile, minimizing control effort for a smoother trajectory is preferred. To effecively execute the pursuit strategy, we aim to reach the target in the shortest time, considering both the terminal state and execution time. In conclusion, the requirements for the shortest-time pursuit lead to the following problem:

$$\min_{x(t), T} \mathcal{J}_o = \int_0^T \left\| x^{(3)}(t) \right\|^2 \mathrm{d}t + \rho T \tag{15a}$$

$$s.t. \quad T > 0, \tag{15b}$$

$$x^{(s-1)}(0) = \bar{x}_o, x^{(s-1)}(T) = \bar{x}_f, \tag{15c}$$

$$\left\| x^{(1)}(t) \right\| \leq v_m, \left\| x^{(2)}(t) \right\| \leq a_m, \forall t \in [0, T], \tag{15d}$$

$$\mathbf{n}^{\mathrm{T}} x(t) \leq \boldsymbol{p}, \forall t \in [0, T], \tag{15e}$$

where (15a) formulates a spatiotemporal optimization problem for the trajectory, balancing trajectory smoothness and time optimality. (15b) represents general constraints on time feasibility. (15c) imposes terminal state constraints on the start and end positions. (15d) ensures dynamic feasibility, with v_m, a_m as velocity and acceleration boundaries. (15e) defines safety constraints, ensuring obstacle avoidance by constraining the trajectory within the SPRR defined in Section III-B. $\langle \mathbf{n}, \boldsymbol{p} \rangle$ are hyperplane parameters described in Section III.

The polynomial trajectory $\mathfrak{T}_{\text{MINCO}}$ defines trajectories as:

$$\begin{aligned} \mathfrak{T}_{\text{MINCO}} = \Big\{ & x(t) : [0, T] \mapsto \mathbb{R}^3 \mid \mathbf{c} = \mathcal{M}(\mathbf{q}, \mathbf{T}), \\ & \mathbf{q} \in \mathbb{R}^{3(M-1)}, \mathbf{T} \in \mathbb{R}_{>0}^M \Big\} \end{aligned} \tag{16}$$

An 3-dimensional trajectory $x(t)$ is represented by a piecewise polynomial with M segments and degree $N = 2s - 1$, where $s = 3$ for obtaining a smoother trajectory. Trajectories are compactly parameterized by the waypoint vector $\mathbf{q}$ and time vector $\mathbf{T}$, using the linear-complexity mapping $\mathbf{c} = \mathcal{M}(\mathbf{q}, \mathbf{T})$. The i-th piece is denoted by $p_i(t) = \mathbf{c}_i^{\mathrm{T}} \beta(t), t \in [0, T_i], \mathbf{c}_i \in \mathbb{R}^{2s \times 3}$ is the coefficient matrix of the piece and $\beta(t) = \left(1, t, \ldots, t^N\right)^{\mathrm{T}}$ is the natural basis.

C. Optimization Problem Solving

A constrained non-convex optimization problem cannot be solved directly. We transcribe the constraints into penalty terms, transforming (15) into a unconstrained nonlinear optimization problem as

$$\min_{\mathbf{c}, \mathbf{T}} \mathcal{J}_o + w_r \mathcal{J}_r \tag{17}$$

where w_r is the weight. $\mathcal{J}_o$ is described in (15a), the gradients can be evaluated as

$$\begin{aligned} \frac{\partial \mathcal{J}_o}{\partial c_i} &= 2\left(\int_0^{T_i} \beta^{(3)}(t) \beta^{(3)}(t)^{\mathrm{T}} \, \mathrm{d}t\right) c_i, \\ \frac{\partial \mathcal{J}_o}{\partial T_i} &= c_i^{\mathrm{T}} \beta^{(3)}\left(T_i\right) \beta^{(3)}\left(T_i\right)^{\mathrm{T}} c_i + \rho. \end{aligned} \tag{18}$$

Continuous time penalty function, corresponding to (15d) and (15e), can be described as

$$\begin{cases} \mathcal{Y}_h = \mathbf{n}^{\mathrm{T}} x_i(t) - \boldsymbol{p} \leq \mathbf{0}, & \forall t \in [0, T_i], \\ \mathcal{Y}_v = \left\| x_i^{(1)}(t) \right\|^2 - v_m^2 \leq 0, & \forall t \in [0, T_i], \\ \mathcal{Y}_a = \left\| x_i^{(2)}(t) \right\|^2 - a_m^2 \leq 0, & \forall t \in [0, T_i]. \end{cases} \tag{19}$$

Through sampling methods , infinite constraints on continuous time can be transcribed into finite constraints [15], which can be further used to construct the penalty term $\mathcal{J}_r$ as follow.

$$\begin{aligned} \mathcal{I}_i^{\star} &= \frac{T_i}{\kappa_i} \sum_{j=0}^{\kappa_i} \bar{\omega}_j \max\left[\mathcal{Y}_{\star}\left(\mathbf{c}_i, T_i, \frac{j}{\kappa_i}\right), \mathbf{0}\right]^3, \\ \mathcal{J}_r &= \sum_{i=1}^{M} \mathcal{I}_i^{\star}, \star = \{h, v, a\}, \\ \frac{\partial \mathcal{J}_r}{\partial c_i} &= \frac{\partial \mathcal{I}_i^{\star}}{\partial \mathcal{Y}_{\star}} \frac{\partial \mathcal{Y}_{\star}}{\partial c_i}, \frac{\partial \mathcal{J}_r}{\partial T_i} = \frac{\mathcal{I}_i^{\star}}{T_i} + \frac{j}{\kappa_i} \frac{\partial \mathcal{I}_i^{\star}}{\partial \mathcal{Y}_{\star}} \frac{\partial \mathcal{Y}_{\star}}{\partial t}, \\ \frac{\partial \mathcal{I}_i^{\star}}{\partial \mathcal{Y}_{\star}} &= 3 \frac{T_i}{\kappa_i} \sum_{j=0}^{\kappa_i} \bar{\omega}_j \chi^{\mathrm{T}} \max\left[\mathcal{Y}_{\star}\left(\mathbf{c}_i, T_i, \frac{j}{\kappa_i}\right), \mathbf{0}\right]^2. \end{aligned} \tag{20}$$

where $\bar{\omega}_j = \begin{cases} \frac{1}{2} & j = 1, \kappa_i \\ 1 & \text{other} \end{cases}$ is a weight based on the equidistant integration rule, which has been proven to better estimate continuous time penalty terms numerically [15].

V. SIMULATIONS

We design two experiments that provide an in-depth analysis of the proposed SPRR-based cooperative capture strategy, demonstrating its advantages in terms of both effectiveness and advancement. We set the capture distance $r_c = 0.4m$ and the safety radius $r_c = 0.3m$. Without losing generality, we set $v_{e,\max} = v_{p,\text{ max}} = 1m/s$. As shown in Fig.3, the experimental scenario is a 25 m × 25 m × 5 m bounded area with randomly generated obstacle point clouds simulating the feedback from the quadrotor's visual perception system. The data is visualized as voxel grid map, which is commonly used in quadrotor systems. The colored convex regions represent the SPRR regions constructed by the proposed method. In this large-scale scenario, the time to generate the SPRR and complete strategy generation ranges from 3 to 10 milliseconds, meeting the real-time requirements of quadrotor systems.

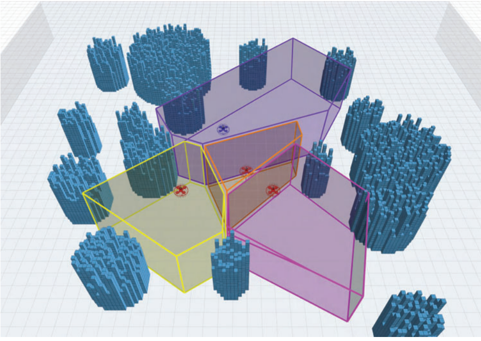

Fig. 3: Simulation experiment environment, where the regions composed of blue squares represents obstacle Q_o, the red spherical quadrotor represents the pursers, the blue spherical quadrotor represents the evader, and the colored convex region represents the SPRR $\mathcal{R}_e$ and $\mathcal{R}_i$ of each quadrotor.

A. Exp1: Effectiveness Validation Experiment

In this section, we apply the proposed SPRR-based capture method against state-of-the-art escape strategies to evaluate its effectiveness in achieving capture within a limited time. The advanced escape strategies include:

1) The Greedy Policy (GP) aims to move the evader away from the nearest pursuer x_n. The control strategy under GP is given by $\dot{x}_{e1} = \|v_{e,\max}\| \frac{x_n - x_e}{\|x_n - x_e\|}$.

2) The Move-to-Centroid Policy (MCP) in [11] balances the pressures from surrounding pursuers by moving the pursuer towards the centroid of the Voronoi partition. The control strategy under MCP is given by $\dot{x}_{e2} = \|v_{e,\max}\| \frac{C_{\mathcal{R}_e} - x_e}{\|C_{\mathcal{R}_e} - x_e\|}$.

3) The Potential Field Policy (PFP) in [19] enables the evader to react more agilely to the pursuer team's movements. The control strategy under PFP is given by $\dot{x}_{e3} = \|v_{e,\max}\| \frac{\sum_i \frac{k_e(x_e - p_i)}{r_i} + x_e}{\left\|\sum_i \frac{k_e(x_e - p_i)}{r_i} + x_e\right\|}$.

To demonstrate the effectiveness of the proposed method more effectively, we introduce three performance metrics: (1)**Capture Distance**: the distance between evader and nearest pursuer;(2)**Obstacle Avoidance**: the nearest distance between the pursuer and the nearest obstacle;(3)**Collision Avoidance**: the nearest distance between pursuers. As shown in Fig.4, the results confirm that the proposed method achieves effective capture while ensuring safety when countering the state-of-the-art evasion methods.

B. Exp2: Advancement Validation Experiment

In this section, we compare the proposed method with state-of-the-art multi-agent collaborative capture methods, using capture time to demonstrate the algorithm's advancement. The current advanced pursuit methods include:

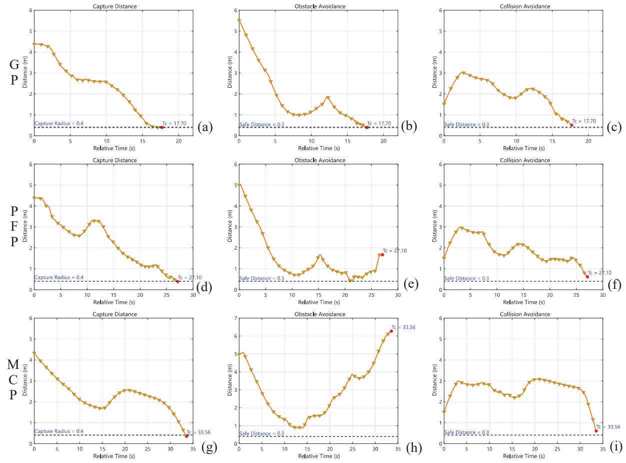

Fig. 4: (a), (b), and (c) represent the capture distance, obstacle avoidance, and collision avoidance under GP, respectively. Similarly, (d), (e), and (f) represent the performance under PFP. (g), (h), and (i) represent the performance under MCP.

1)Buffered Evader-Centered Bounded Voronoi Cell (B-ECBVC) methods [12], in which pursuers initially leverage their speed advantage to encircle the evader before capturing;

2)Buffered Voronoi Cell-based Greedy Capturing (BVC-G) methods [13], where pursuers move towards the point in the BVC that is closest to the evader to capture it.

The results of Exp1 clearly indicate that the MCP method outperforms other methods by effectively balancing the pressure exerted by all pursuers, resulting in superior evasion performance. Therefore, in Exp2, all evaders adopt the Move-to-Centroid Policy uniformly.

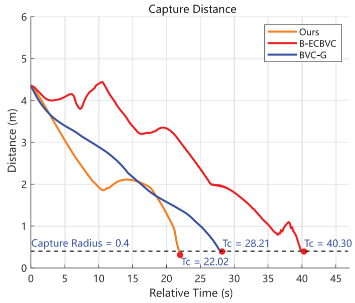

Fig. 5: Single experiment

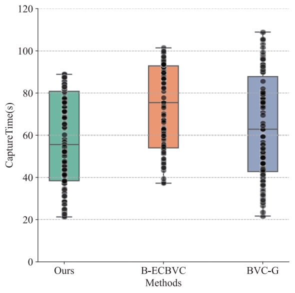

Fig. 6: Repeated experiments

As shown in Fig.5, in a single trial, the proposed SRPP method achieves effective capture of the evader in a shorter time. Note that the safety throughout the pursuit process is consistently maintained. Due to space constraints and the validation of algorithm's safety in Exp1, safety data is not repeated in Exp2. To further demonstrate the advantage of the proposed method in capture time, we conducted 100 repeated experiments with different initial positions in the same scenario. As shown in Fig.6, our algorithm achieves the minimum average capture time.

VI. CONCLUSION

This article proposes a multi-quadrotor cooperative algorithm for encirclement and capture in obstacle environments. Compared with previous research, its significant advantage lies in its ability to be applicable to real quadrotor systems based on visual obstacle perception. Extensive adversarial and comparative experiments demonstrate that our algorithm can fully ensure safety during the capture, and its capture efficiency is also superior to state-of-art methods. And the entire strategy generation process can be completed within $3-10ms$, meeting the real-time requirements of real quadrotor. The limitation lies in the algorithm depends on the global positions of all agents, which is not addressed here. Future work will focus on localization and target estimation to enhance autonomy.

REFERENCES

[1] T. H. Chung, G. A. Hollinger, and V. Isler, "Search and pursuit-evasion in mobile robotics," *Auton. Robots*, vol. 31, no. 4, pp. 299–316, 2011.

[2] L. Xi, X. Wang, L. Jiao, S. Lai, Z. Peng, and B. M. Chen, "GTO-MPC-based target chasing using a quadrotor in cluttered environments," *IEEE Trans. Ind. Electron.*, vol. 69, no. 6, pp. 6026–6035, Jun. 2021.

[3] J. Chen, W. Zha, Z. Peng, and D. Gu,"Multi-player pursuit-evasion games with one superior evader," *Automatica*, vol. 71, pp. 24-32, 2016.

[4] L. Cheng and Y. Yuan, "Multiplayer obstacle avoidance pursuit-evasion games with adaptive parameter estimation," *IEEE Trans. Ind. Electron.*, vol. 70, no. 5, pp. 5171-5181, May 2022.

[5] M. D. Awheda and H. M. Schwartz, "A fuzzy reinforcement learning algorithm using a predictor for pursuit-evasion games,"*2016 Annual IEEE Systems Conference (SysCon)*, pp. 1-8, 2016.

[6] R Lowe, Y Wu,A Tamar, "Multi-agent actor-critic for mixed cooperative-competitive environments,"*Advances in Neural Information Processing Systems*, vol. 30, pp. 6379-6390, 2017.

[7] A. T. Bilgin and E. Kadioglu-Urtis, "An approach to multi-agent pursuit evasion games using reinforcement learning," *2015 International Conference on Advanced Robotics*, pp. 164-169, Istanbul, Turkey, 2015.

[8] H. Wang and Y. Zhang, "Impulsive maneuver strategy for multi-agent orbital pursuit-evasion game under sparse rewards," *Aerospace Sci. Technol.*, vol. 155, p. 109618, 2024.

[9] J. Liao, C. Liu, and H. H. Liu, "Model predictive control for cooperative hunting in obstacle-rich and dynamic environments," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2021, pp. 5089-5095.

[10] C. Wang, H. Chen, J. Pan, and W. Zhang, "Encirclement guaranteed cooperative pursuit with robust model predictive control," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2021, pp. 1473-1479.

[11] A. Pierson, Z. Wang and M. Schwager, "Intercepting Rogue Robots: An Algorithm for Capturing Multiple Evaders With Multiple Pursuers," in *IEEE Robot. Autom. Lett*, vol. 2, no. 2, pp. 530-537, April 2016.

[12] B. Tian, P. Li, H. Lu, et al., "Distributed pursuit of an evader with collision and obstacle avoidance," *IEEE Trans. Cybern.*, vol. 52, no. 12, pp. 13512-13520, Dec. 2021.

[13] X. Wang, L. Xi, Y. Ding, et al., "Distributed encirclement and capture of multiple pursuers with collision avoidance," *IEEE Trans. Ind. Electron.*, vol. 70, no. 9, pp. 8084-8095, Sep. 2023.

[14] X. Wang, Y. Ding, Y. Chen, et al., "OA-ECBVC: A cooperative collision-free encirclement and capture approach in cluttered environments," in *Proc. 62nd IEEE Conf. Decision Control (CDC)*, 2023, pp. 5416-5422.

[15] Z. Wang, X. Zhou, C. Xu and F. Gao, "Geometrically Constrained Trajectory Optimization for Multicopters," *IEEE Transactions on Robotics*, vol. 38, no. 5, pp. 3259-3278, Oct. 2022.

[16] O. Arslan and D. E. Koditschek, "Sensor-based reactive navigation in unknown convex sphere worlds,"*Int. J. Robot. Res.*, vol. 38, no. 2–3, pp. 196–223, 2019.

[17] H. Huang, Z. Zhou, W. Zhang, J. Ding, D. M. Stipanovic, and C. J. Tomlin, "Safe-reachable area cooperative pursuit,"*IEEE Trans. Robot.*, vol. 10, no. 5, 2012.

[18] D. Dolgov, S. Thrun, M. Montemerlo, and J. Diebel, "Path planning for autonomous vehicles in unknown semi-structured environments," The International Journal of Robotics Research, vol. 29, no. 5, pp. 485–501, 2010.

[19] X. Fang, C. Wang, L. Xie, and J. Chen, "Cooperative pursuit with multi-pursuer and one faster free-moving evader,"*IEEE Trans. Cybern.*, vol. 52, no. 3, pp. 1405–1414, Mar. 2022.