

Swarm Navigation Based on Smoothed Particle Hydrodynamics in Complex Obstacle Environments

Ruocheng Li¹, Bin Xin¹, Shuai Zhang², Mingzhe Lyu³, and Jinqiang Cui⁴

Abstract—In this letter, we propose a method for the navigation of swarm unmanned aerial vehicles (UAVs) in complex environments with obstacles. We propose an algorithmic framework based on Smoothed Particle Hydrodynamics (SPH). In this framework, each UAV is considered a particle, computing its motion information through local interactions with surrounding particles. Based on SPH, the UAV swarm can interactively adjust itself, allowing the entire cluster to advance in the flow pattern of an incompressible fluid. We introduce the Euclidean Signed Distance Field (ESDF) as a representation of the environment. The ESDF is constructed based on the obstacle information in the environment, enabling the swarm to deform and avoid obstacles within the environment. Simultaneously, we propose a swarm navigation function based on B-splines, rapidly obtaining executable trajectories by solving an unconstrained gradient optimization problem. Compared with existing methods, our algorithm exhibits significant improvements in success rate, stability, and scalability. Extensive simulations and physical experiments in both 2D and 3D environments have demonstrated the effectiveness of the proposed method. More elements can be referenced on our GitHub:<https://github.com/SmartGroupSystems/SPH-planning>.

Index Terms—Multi-UAV systems, swarm navigation, smoothed particle hydrodynamics

I. INTRODUCTION

HERE are a variety of complex and hazardous tasks that necessitate spatial movement and collaborative work over extensive areas in real-world situations. As task requirements become increasingly complex, UAV systems are evolving from individual units to swarm systems, which poses new challenges to motion planning technologies. At the motion planning level, the core problem that UAVs have to address is collision

Manuscript received: January 25, 2025; Revised: April 23, 2025; Accepted: May 28, 2025.

This paper was recommended for publication by Editor M. Ani Hsieh upon evaluation of the Associate Editor and Reviewers' comments.

This work was supported in part by the National Science Fund for Distinguished Young Scholars under Grant 62425304, in part by the Basic Science Center Programs of NSFC under Grant 62088101, in part by the Beijing Advanced Innovation Center for Intelligent Robots and Systems, in part by the Shanghai Municipal Science and Technology Major Project under Grant 2021SHZDZX0100, in part by the Shanghai Municipal Commission of Science and Technology Project under Grant 19511132101, in part by the Talent Program of Guangdong Province (2021QN02Z107).

¹Ruocheng Li and Bin Xin are with the School of Automation, Beijing Institute of Technology, Beijing 100081, China {ruochengli, brucebin}@bit.edu.cn

²Shuai Zhang is with the Department of Computer Science, The University of Hong Kong, China shuaizhang2017@gmail.com

³Mingzhe Lyu is with the Southern University of Science and Technology, Shenzhen 518066, China 12131040@mail.sustech.edu.cn

⁴Jinqiang Cui is with the Peng Cheng Laboratory, Shenzhen, China cuijq@pcl.ac.cn

Digital Object Identifier (DOI): see top of this page.

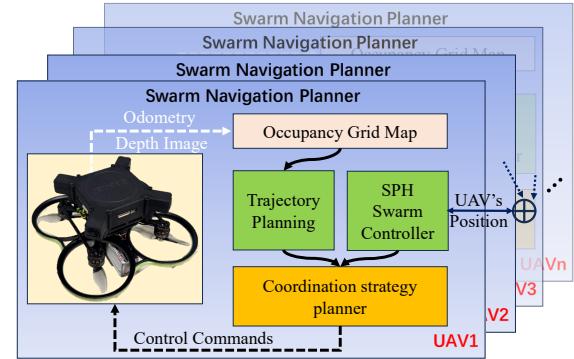


Fig. 1: A block diagram illustrating the full pipeline of the proposed navigation framework.

avoidance, which has two main aspects: collision avoidance: UAVs must avoid collisions with each other; and obstacle avoidance: UAVs must avoid obstacles in the environment.

A typical method is Multi-Agent Path Finding (MAPF). This method is based on setting up an environment and multiple robots with specified start and end positions. The goal is to find feasible paths for the robots that avoid collisions with obstacles and with each other. Wolfgang et al. [1] proposed a MAPF method with dynamic constraints. This method maintains a collision-free topology by constructing a Temporal Plan Graph (TPG) and minimizes the total operation time cost function by solving a linear program. This method performs well in centralized cluster topologies [2]. However, maintaining a centralized topology in practical tasks is often difficult, and the system scalability is poor. Jungwon et al. [3] proposed the Relative Safe Flight Corridor (RSFC) method. This method first uses the path generated by MAPF as an initial solution. Then it constructs time-linear equality constraints through backend optimization to ensure collision-free trajectories for multiple UAVs. This method has been validated in environments with static obstacles [4] and dynamic obstacles [5]. However, the core idea of RSFC is trajectory optimization based on MAPF, which requires a stable global communication topology for its maintenance. Another class of methods design navigation frameworks based on distributed trajectory planning. Zhou et al. [6] proposed a decentralized trajectory planning framework, where multi-robot collision avoidance is achieved by constructing cost functionals between positions. This framework can still guarantee safety through point cloud information when the communication topology cannot be ensured. Hou et al. [7] weakened the optimality guarantee of MAPF by

constructing collision-free paths using a suboptimal initial solution and generating dynamically feasible smooth trajectories through joint optimization. Wang et al. [8] designed a cost function with time capsule constraints and achieved multi-robot collision avoidance by solving unconstrained gradient optimization. Tordesillas et al. proposed the MADER [9] framework to achieve distributed navigation in obstacle-rich environments. This method relies on the trajectories of other UAVs for collision avoidance, so its effectiveness decreases significantly as swarm size increases. The essence of these methods lies in constructing and solving multi-objective cost functions. They cannot guarantee optimality or the existence of a solution, and the quality of the numerical solution depends heavily on the initial guess and the mathematical expression of the cost functions.

Smoothed Particle Hydrodynamics (SPH) has emerged as a powerful tool for swarm control, offering enhanced flexibility and scalability. Zhao et al. [10] proposed a swarm control method based on SPH. This method improves the original SPH, adapting it to meet the control requirements of swarm robotics while preserving the fundamental properties of fluid dynamics. Mitikiri et al. [11] designed a distributed controller based on SPH, enabling swarm robots to exhibit fluid-like swarm behaviors. Pimenta et al. [12] combined the SPH method with the artificial potential field approach, enabling obstacle avoidance and target-reaching in globally known obstacle environments. Zhang et al. [13], [14] proposed a density-based interaction method utilizing SPH to address multi-target trapping problem in swarm robotics. They extended their work to include an adaptive controller that allows the robot swarm to self-organize and deploy for target trapping, while dynamically adjusting the encirclement thickness and robot group size in response to target strength [15]. The aforementioned methods demonstrate the advantages of the SPH approach in achieving swarm expansion; however, further design is required to address unknown and complex environments effectively.

In summary, existing motion planning methods still have the following gaps [16]:

- **Adaptability.** Methods designed for open spaces or idealized conditions often fail in confined or obstacle-rich environments.
- **Scalability.** Approaches tailored to specific group sizes or velocities may become unstable when scaled to larger swarms or higher speeds.
- **Complexity.** Real-world swarm planning involves numerous parameters with nonlinear interactions, requiring fast adaptation across diverse scenarios.

Motivated by the above facts, this work integrates SPH swarm control with autonomous navigation to enable scalable and robust UAV swarm operation in complex environments. UAVs are modeled as particles and achieve local coordination through distributed SPH-based interactions. Building upon previous work [17], we extend the navigation framework from individual agents to multi-UAV systems by allowing each UAV to compute its trajectory independently based on local observations, without relying on a global map or centralized planning. To support adaptive coordination in cluttered spaces,

we incorporate a raycasting-based check to determine whether neighboring agents are occluded by obstacles. Each UAV interacts only with visible neighbors, allowing the swarm to dynamically adjust its coordination structure according to the environment. This selective interaction strategy enhances both safety and flexibility, enabling effective navigation in unknown and complex scenarios. Experimental results show that the proposed framework significantly improves swarm navigation success rates under varying environmental conditions. Compared to existing approaches [6], [9], [18], our method: (1) adapts robustly to cluttered environments via locally responsive strategies; (2) leverages the local support property of SPH for high scalability; and (3) requires minimal inter-agent communication without relying on neighbors' predicted trajectories, ensuring parameter flexibility. The main contributions of this paper are summarized as follows:

1) We propose a decentralized swarm navigation framework based on Smoothed Particle Hydrodynamics (SPH), where each robot independently computes a B-spline-guided trajectory while maintaining local coordination through SPH interactions. This enables scalable and robust navigation in complex unknown environments.

2) We develop a raycasting-based swarm splitting-and-merging strategy that allows agents to dynamically form local sub-swarms based on visibility constraints and merge when feasible. This mechanism significantly improves navigation success rates in cluttered and occlusion-rich environments.

3) We demonstrate the effectiveness of our approach through real-world experiments in both 2D and 3D environments using 10 autonomous UAVs, without relying on any external motion capture system. Comparative evaluations against baseline methods highlight superior performance, and large-scale simulations confirm scalability and robustness in dense obstacle fields.

The rest of this paper is structured as follows: Sec.II provides an overview of the foundational concepts and problem formulation; Sec.III elaborates on the development of the swarm framework utilizing SPH; Sec.IV showcases the results from simulation experiments and analyzes the system's performance; and finally, Sec.V summarizes the conclusions of the study and suggests directions for future research.

II. PROBLEM DESCRIPTION

Let $\mathcal{U}_c \subseteq \mathcal{U}_{\text{free}} \subset \mathbb{R}^3$ denote the region occupied by all UAVs and the obstacle-free space, respectively. Consider a swarm system consisting of M UAVs, where each UAV $i \in \{0, \dots, M-1\}$ can only interact with neighbors located within a sensing radius $R > 0$. The local neighbor set for UAV i at time step k is defined as

$$\mathcal{N}_i(k) = \{j \neq i \mid \|\mathbf{q}_j(k) - \mathbf{q}_i(k)\| < R\}. \quad (1)$$

Each UAV evolves according to a second-order discrete-time model:

$$\begin{aligned} \mathbf{q}_{k+1,i} &= \mathbf{q}_{k,i} + \Delta t \mathbf{v}_{k,i}, \\ \mathbf{v}_{k+1,i} &= \mathbf{v}_{k,i} + \Delta t \mathbf{u}_{k,i}, \end{aligned} \quad (2)$$

where $\mathbf{u}_{k,i} \in \mathbb{R}^3$ is the control input, subject to $\|\mathbf{v}_{k,i}\| \leq v_{\max}$, $\|\mathbf{u}_{k,i}\| \leq a_{\max}$. Let $Q_s = \{\mathbf{q}_{s,i}\}_{i=0}^{M-1}$ and $Q_g = \{\mathbf{q}_{g,i}\}_{i=0}^{M-1}$

denote the sets of initial and goal positions. Each UAV independently determines a control sequence $\{\mathbf{u}_{k,i}\}_{k=0}^{T-1}$ to reach its assigned goal while ensuring smooth motion and maintaining swarm cohesion:

$$\begin{aligned} \min_{\{\mathbf{u}_{k,i}\}_{k=0}^{T-1}} \quad & \mathcal{L}_i = \mathcal{L}_s(\mathcal{N}_i(k)) + \mathcal{L}_t(\{\mathbf{u}_{k,i}\}) + \mathcal{L}_g(\mathbf{q}_{T,i}, \mathbf{q}_{g,i}) \\ \text{s.t.} \quad & \mathbf{x}_{0,i} = \begin{bmatrix} \mathbf{q}_{s,i} \\ \mathbf{v}_{s,i} \end{bmatrix}, \\ & \mathbf{x}_{k+1,i} = \mathbf{A}\mathbf{x}_{k,i} + \mathbf{B}\mathbf{u}_{k,i}, \\ & \mathbf{q}_{k,i} \in \mathcal{U}_{\text{free}}, \quad \mathbf{q}_{k,i} \notin \mathcal{U}_c^{(-i)}(k), \\ & \|\mathbf{v}_{k,i}\| \leq v_{\max}, \quad \|\mathbf{u}_{k,i}\| \leq u_{\max}, \\ & \mathcal{N}_i(k) \neq \emptyset, \quad \forall k \in \{0, \dots, T-1\}. \end{aligned} \quad (3)$$

Here, $\mathbf{x}_{k,i} = [\mathbf{q}_{k,i}^\top, \mathbf{v}_{k,i}^\top]^\top \in \mathbb{R}^6$ denotes the full state of UAV i , and $\mathcal{U}_c^{(-i)}(k)$ represents the space occupied by other UAVs at time k . The cost function \mathcal{L}_i serves as a conceptual formulation of our framework, which integrates SPH-based interaction modeling (\mathcal{L}_s), local trajectory generation (\mathcal{L}_t), and target-directed motion planning (\mathcal{L}_g). Our goal is to achieve decentralized swarm coordination with local feasibility and convergence toward target positions.

III. FRAMEWORK FOR DISTRIBUTED UAV SYSTEMS

The pipeline of the framework is illustrated in Fig. 1. Each UAV uses onboard sensors to compute its odometry and constructs an occupancy grid map using odometry and depth images. Each UAV independently computes its navigation trajectory while acquiring position information of other UAVs through the broadcast network. Based on this position information, the UAV calculates its expected position within the swarm. Finally, a coordination planner is used to align the trajectory with the expected position.

A. Swarm Framework Based on SPH

Based on the work [14], we present the following kernel function:

$$W(\mathbf{q}, h) = \frac{c}{\pi h^3} \begin{cases} 1 - \frac{3}{2}k^2 + \frac{3}{4}k^3 & \text{if } 0 \leq k \leq 1 \\ \frac{1}{4}(2-k)^3 & \text{if } 1 < k \leq 2 \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

where h is the kernel function radius, $k = \|\mathbf{q}\|/h$ and c is a scaling constant. We present the SPH framework for swarm UAVs as shown in Alg. 1.

For each particle, we compute its density value based on its neighbors (line 5). In the SPH method, the pressure (attractive and repulsive forces) and viscosity forces between particles are calculated based on their density values. In the framework, each particle needs to compute the attractive force $\mathbf{F}_i^{\text{density}}$, the repulsive force $\mathbf{F}_i^{\text{repulsive}}$ and the viscosity force $\mathbf{F}_i^{\text{viscosity}}$. Forces $\mathbf{F}_i^{\text{density}}$ and $\mathbf{F}_i^{\text{repulsive}}$ ensure that the swarm maintains a fixed distance at steady state, driving the system's density value toward the reference density ρ_0 . Force $\mathbf{F}_i^{\text{viscosity}}$ helps the system quickly converge to the desired value. It is worth noting that the calculations of $\mathbf{F}_i^{\text{density}}$ and $\mathbf{F}_i^{\text{repulsive}}$ are simplified. They merely rely on the particle's velocity (itself) and the

relative distance. This makes the computation of repulsive and viscosity forces more aligned with the control description of multi-agent systems. Parameters k_{den} , k_{rep} , and k_{vis} are proportional coefficients for different terms. The parameter γ represents the model's sensitivity to changes in density. A larger value of γ indicates a more pronounced impact of density variations, causing the system to respond more quickly to small density fluctuations. We refer to the classical fluid dynamics model and set $\gamma = 7$ (lines 8-10). The entire process is shown in Fig. 2.

Algorithm 1 SPH Swarm Framework

```

1: for all particle  $i$  do
2:   find neighbors  $j$ 
3: end for
4: for all particle  $i$  do
5:    $\rho_i = \sum_j m_j W_{ij}$ 
6: end for
7: for all particle  $i$  do
8:    $\mathbf{F}_i^{\text{density}} = -k_{\text{den}} \frac{1}{\rho_i} \left[ \left( \frac{\rho_i}{\rho_0} \right)^\gamma - 1 \right] \sum_{j=1}^{\mathcal{N}_i} \nabla_i W(\mathbf{q}_{ij})$ 
9:    $\mathbf{F}_i^{\text{repulsive}} = k_{\text{rep}} \sum_{j=1}^{\mathcal{N}_i} \frac{1}{q_{ij}^2} \mathbf{q}_{ij}$ 
10:   $\mathbf{F}_i^{\text{viscosity}} = -k_{\text{vis}} \mathbf{v}_i$ 
11:   $\mathbf{F}_i^{\text{env}} = k_{\text{env}} \nabla \mathcal{F}(\mathbf{q}_i, d_{\text{thr}})$ 
12:   $\mathbf{F}_i(t) = \mathbf{F}_i^{\text{density}} + \mathbf{F}_i^{\text{repulsive}} + \mathbf{F}_i^{\text{viscosity}} + \mathbf{F}_i^{\text{env}}$ 
13: end for
14: for all particle  $i$  do
15:    $\mathbf{v}_i(t + \Delta t) = \mathbf{v}_i(t) + \Delta t \mathbf{F}_i(t) / m_i$ 
16:    $\mathbf{q}_i(t + \Delta t) = \mathbf{q}_i(t) + \Delta t \mathbf{v}_i(t + \Delta t)$ 
17: end for

```

Force $\mathbf{F}_i^{\text{env}}$ represents the force exerted on the particle by the environment. Function \mathcal{F} is constructed as follows:

$$\mathcal{F}(\mathbf{q}_i, d_{\text{thr}}) = \begin{cases} (d(\mathbf{q}_i) - d_{\text{thr}})^2 & , \text{ if } d(\mathbf{q}_i) < d_{\text{thr}} \\ 0 & , \text{ if } d(\mathbf{q}_i) > d_{\text{thr}} \end{cases} \quad (5)$$

Here, $d(\mathbf{q}_i)$ denotes the distance between \mathbf{q}_i and the nearest obstacle, with d_{thr} defining the threshold for obstacle clearance. The gradient is as follows:

$$\nabla \mathcal{F}(\mathbf{q}_i, d_{\text{thr}}) = 2(d(\mathbf{q}_i) - d_{\text{thr}}) d'(\mathbf{q}_i), \mathcal{F}(\cdot) \neq 0, \quad (6)$$

where $d'(\mathbf{q}_i)$ is the gradient of the particle in the Euclidean signed distance field (ESDF) [19]. Here, the ESDF serves as a continuous and differentiable representation of the environment that enables efficient computation of repulsive forces for smooth obstacle avoidance.

B. B-Spline Trajectory Optimization

In this work, we employ a 3rd-degree uniform B-spline to generate the reference trajectory. Given the spline degree p , knot vector $\{u_0, u_1, \dots, u_m\}$, basis functions N_i^p , and control points $\{\mathbf{C}_0, \mathbf{C}_1, \dots, \mathbf{C}_K\}$, the spline curve is defined as

$$\mathbf{q}(u(s(t))) = \sum_{i=0}^K \mathbf{C}_i B_i^p(u(s(t))), \quad B_i^p \in \mathbb{R}, \quad (7)$$

where $m = K+p+1$, and B_i^p are the degree- p basis functions constructed via the Cox-de Boor recursion. The parameter $s \in$

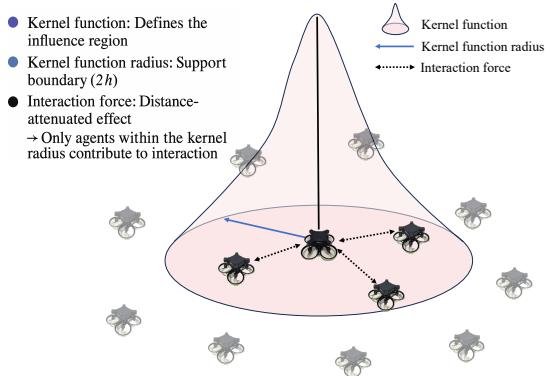


Fig. 2: Illustration of the kernel function in the SPH-based swarm interaction model. Each UAV is only influenced by nearby agents within the kernel radius, and the interaction force decreases with distance according to the kernel profile.

$[u_p, u_{m-p}]$ is mapped from physical time through $t = (s - u_p)/\beta$, where β is a temporal scaling factor, yielding the time domain $t \in [0, (u_{m-p} - u_p)/\beta]$.

We optimize the internal control points $\{\mathbf{C}_3, \dots, \mathbf{C}_{K-3}\}$, while keeping the first and last three fixed to enforce boundary conditions. This design guarantees C^2 continuity over the entire trajectory and allows precise control over initial and terminal states. The optimization objective is:

$$\min J_{total} = \lambda_s J_s + \lambda_c J_c + \lambda_f J_f, \quad (8)$$

where J_s , J_c , and J_f denote the costs for smoothness, collision avoidance, and dynamic feasibility, respectively, and λ_s , λ_c , λ_f are the corresponding weights.

The smoothness cost J_s penalizes the third-order finite difference of control points, which approximates the squared norm of jerk over the trajectory:

$$J_s = \sum_{i=0}^{K-3} \|\mathbf{C}_{i+3} - 3\mathbf{C}_{i+2} + 3\mathbf{C}_{i+1} - \mathbf{C}_i\|_2^2. \quad (9)$$

This formulation effectively models an elastic-band-like behavior that favors smooth transitions.

The collision avoidance cost J_c applies the second-order differentiable penalty $\mathcal{F}(\cdot)$ based on the ESDF evaluated at each control point:

$$J_c = \sum_{i=p}^{K-p} \mathcal{F}(\mathbf{C}_i, d_{thr}), \quad (10)$$

where d_{thr} is a safety threshold. By leveraging the convex hull property, this term provides an efficient surrogate for continuous-time obstacle clearance.

The feasibility term J_f imposes soft constraints on velocity and acceleration, where intermediate velocities and accelerations are approximated via finite differences of the control points. Specifically, velocity is computed as $\mathbf{V}_i = (\mathbf{C}_{i+1} - \mathbf{C}_i)/\Delta t$, and acceleration as $\mathbf{A}_i = (\mathbf{V}_{i+1} - \mathbf{V}_i)/\Delta t$. The cost is formulated as:

$$J_f = \sum_j \left(\sum_{i=0}^{K-1} \mathcal{G}(\mathbf{V}_i^{j,2} - v_{max}^2) + \sum_{i=0}^{K-2} \mathcal{G}(\mathbf{A}_i^{j,2} - a_{max}^2) \right), \quad (11)$$

with $\mathcal{G}(x) = \max\{x, 0\}^2$ and $j \in \{x, y, z\}$. This cost ensures the planned trajectory can be feasibly tracked under physical constraints.

The boundary conditions $[\mathbf{q}_s, \mathbf{v}_s, \mathbf{a}_s]$ and $[\mathbf{q}_e, \mathbf{v}_e, \mathbf{a}_e]$ are computed by differentiating the B-spline at the endpoints, yielding the following expression:

$$\begin{bmatrix} 1/6 & 2/3 & 1/6 \\ -\beta/2 & 0 & \beta/2 \\ \beta^2 & -2\beta^2 & \beta^2 \end{bmatrix} \begin{bmatrix} \mathbf{C}_0 & \mathbf{C}_{K-2} \\ \mathbf{C}_1 & \mathbf{C}_{K-1} \\ \mathbf{C}_2 & \mathbf{C}_K \end{bmatrix} = \begin{bmatrix} \mathbf{q}_s & \mathbf{q}_e \\ \mathbf{v}_s & \mathbf{v}_e \\ \mathbf{a}_s & \mathbf{a}_e \end{bmatrix}. \quad (12)$$

C. Swarm Coordination and Autonomous Planning

Sec. III-A designs a swarm coordination controller, while Sec. III-B presents an autonomous trajectory planner for individual UAVs. The former ensures global coordination, whereas the latter emphasizes local autonomy. In coordinated scenarios, each UAV exchanges information with at least one neighbor, and the swarm flows through the environment like a fluid despite limited perception. However, in cluttered environments, the SPH system lacks global guidance. The environmental force \mathbf{F}_i^{env} enables real-time obstacle avoidance via high-frequency control, but without trajectory-level planning, UAVs struggle to find viable paths.

To address this, we introduce a trajectory planner that equips each UAV with local autonomy. Yet full autonomy can undermine coordination, as the navigation cost is decoupled from swarm behavior. Moreover, the two modules operate on different time scales: swarm control runs at high frequency (≥ 100 Hz), while trajectory planning is slower (≤ 5 Hz). Joint optimization across these scales leads to trade-offs that degrade overall performance.

We use the following control law to achieve coordinated control of the swarm system in complex environments:

$$\begin{aligned} \mathbf{a}_i(t) = & \frac{\mathbf{F}_i(t)}{m_i} + k_p(\mathbf{q}^*(\tau_j) - \mathbf{q}_i(t)) + k_v(\mathbf{v}^*(\tau_j) - \mathbf{v}_i(t)) + \mathbf{a}^*(\tau_j), \\ \tau_j = & t_k + j \cdot \Delta t_s, \quad j = 0, 1, \dots, N_s - 1. \end{aligned} \quad (13)$$

Here, $\mathbf{q}^*(\tau_j)$, $\mathbf{v}^*(\tau_j)$, and $\mathbf{a}^*(\tau_j)$ are the reference position, velocity, and acceleration at the sampling time τ_j , and k_p, k_v are the proportional gains. t_k denotes the trajectory update time, Δt_s is the sampling interval, and N_s is the total number of sampled points along the trajectory. The trajectory is updated within a low-frequency planning period T_p , and the sampled values $[\mathbf{q}^*(\tau_j), \mathbf{v}^*(\tau_j), \mathbf{a}^*(\tau_j)]$ are sequentially sent to the controller for execution. The number of trajectory sampling points should match the control frequency to ensure smooth, accurate tracking and real-time computation of SPH-based interaction forces.

We implement a raycasting-based swarm splitting and merging strategy. Each UAV determines its set of actively interacting neighbors by performing raycasting on its locally perceived point cloud, identifying neighboring agents that are visible and not occluded by obstacles. Only these visible neighbors are considered in the SPH-based interaction computation. This selective coordination mechanism allows the swarm to maintain local cohesion even in cluttered environments where full connectivity cannot be ensured. Based on local perception, each UAV independently plans its trajectory and

Algorithm 2 Raycasting-based Swarm Splitting and Merging

```

1: Input: Agent state  $\mathbf{q}_i$ , neighbor  $\mathcal{N}_i$ , local point cloud  $\mathcal{P}_i$ 
2: Memory: Previous visible neighbor set  $\mathcal{N}_i^{\text{prev}}$ 
3: Output: Control command  $\mathbf{a}_i$ 
4:  $\mathcal{N}_i^{\text{vis}} \leftarrow \emptyset$ 
5: for each  $j \in \mathcal{N}_i$  do
6:   if Raycasting( $\mathbf{q}_i, \mathbf{q}_j, \mathcal{P}_i$ ) == Visible then
7:      $\mathcal{N}_i^{\text{vis}} \leftarrow \mathcal{N}_i^{\text{vis}} \cup \{j\}$ 
8:     if  $j \notin \mathcal{N}_i^{\text{prev}}$  then Merge with  $j$ 
9:     end if
10:    else if  $j \in \mathcal{N}_i^{\text{prev}}$  then Split from  $j$ 
11:    end if
12:  end for
13:  $\mathbf{F}_i \leftarrow \text{ComputeSPH}(\mathbf{q}_i, \mathcal{N}_i^{\text{vis}})$ 
14:  $\mathbf{q}^*(t), \mathbf{v}^*(t), \mathbf{a}^*(t) \leftarrow \text{TrajectoryPlanning}(\mathbf{q}_i, \mathcal{P}_i)$ 
15:  $\mathbf{a}_i(t) \leftarrow \text{Compute control using Eq.( 13)}$ 
16:  $\mathcal{N}_i^{\text{prev}} \leftarrow \mathcal{N}_i^{\text{vis}}$ 

```

computes control inputs using a coordination-aware reference trajectory. This approach integrates predictive trajectory planning for forward guidance with real-time SPH-based interactions for decentralized control, enabling the swarm to flow through complex environments in a cohesive and fluid-like manner. The overall system design follows three key principles: first, autonomous navigation based solely on local sensing; second, fully distributed and scalable coordination; and third, lightweight explicit communication neighbor position exchange, without requiring the transmission of planned trajectories or control commands. The complete procedure is summarized in Alg. 2.

IV. RESULTS

In the simulation, we assume that each UAV has omnidirectional perception, with its sensing range and communication radius represented by a single parameter. Unless otherwise specified, the parameter values are summarized in Tab. I.

A. Simulation Results in Complex Obstacle Scenarios

We conduct simulations using the Robot Operating System (ROS). The simulations are performed on a computer with two Intel Xeon Gold 6258R CPUs running at 2.70GHz and 256GB of memory, providing a total of 112 threads. Each particle performs perception, planning, and control computations using an independent thread.

Fig. 3(A) shows the simulation results that $M = 60$ particles are initialized within a rectangular region and gradually form a circular coordinated cluster under the SPH-based control law. Upon entering the obstacle field, each particle independently plans a collision-free B-spline trajectory without considering others, which may lead to spatially overlapping paths. However, the coordination force drives the swarm to maintain inter-agent separation and collectively follow the trajectory trend rather than exact paths, resulting in successful navigation without collisions.

Fig. 3(B) depicts a narrow-passage scenario with $M = 20$ and $d_{\text{thr}} = 0.5$, where the 2.2 m-wide passage is narrower than

TABLE I: SUMMARY OF VALUES OF CONTROL PARAMETERS

Parameter	Description	Values
Swarm		
c	normalization constant	54.97
R	Connection range of robot	20 m
h	Kernel function radius	10 m
m	Particle mass	1 kg
k_{den}	Strength of aggregation	1
k_{rep}	Strength of dispersion	1
k_{vis}	Strength of velocity damping	1.5
k_{env}	Strength of obstacle avoidance	10
v_{max}	Maximum velocity of particle	1 m/s
a_{max}	Maximum acceleration of particle	1.5 m/s ²
ρ_0	Initial reference density	10^6
Planning		
T_p	Replanning interval	0.2 s
λ_s	Smoothness weight	5
λ_c	Collision-avoidance weight	1000
λ_f	Dynamic feasibility weight	1
k_p	Strength of trajectory position tracking	5
k_v	Strength of trajectory velocity tracking	3.5
d_{thr}	Obstacle safety threshold	0.5

the swarm's original 4.12 m span. In this scenario, each robot performs trajectory optimization only once at the beginning of the simulation. The swarm deforms longitudinally to fit the constrained space and passes through as a whole without splitting. Particles rely on trajectory trends during execution, ensuring safety under tight spatial conditions.

To further validate robustness in 3D environments, Fig. 3(C1-C6) presents a highly cluttered obstacle field where $M = 30$ particles traverse complex geometries and partial visibility regions. Local motion details, including deformation and dynamic reconfiguration, are visualized in Fig. 3(C3-C6). Fig. 3(D1-D4) shows trajectories in a randomly generated 3D forest, emphasizing the swarm's capacity for local adaptation and coordination under irregular and unstructured constraints.

Statistical results across the above scenarios are summarized in Fig. 3(E-G). Fig. 3(E) shows the average swarm velocity, indicating consistent motion efficiency across varying conditions. Fig. 3(F) reports the minimum inter-robot distance, verifying the system's ability to maintain safe internal spacing. Fig. 3(G) presents the minimum robot-to-obstacle distance, confirming the effectiveness of the proposed planning and coordination scheme in preserving safe margins to environmental obstacles.

We conduct extensive comparative experiments across EGO-Swarm [6], MADER [9], LSC [20], NMPC [21] using 15 robotics. We define the obstacle density as $d_{\text{obs}} = \frac{V_{\text{obs}}}{V_{\text{total}}}$, where V_{obs} is the volume occupied by obstacles and V_{total} is the total environment volume. Each method is evaluated under four density levels with 20 randomized trials per setting. We compare average flight distance, flight time, success rate, energy consumption (based on jerk integration $E = \int_0^T \left\| \frac{d^3 \mathbf{q}(t)}{dt^3} \right\|^2 dt$), communication bandwidth, and average velocity. The results are summarized in Tab. II.

Compared with these methods, our method achieves shorter average flight distances by leveraging SPH formulation combined with decentralized trajectory sampling, enabling agents to adaptively distribute without enforcing strict trajectory sep-

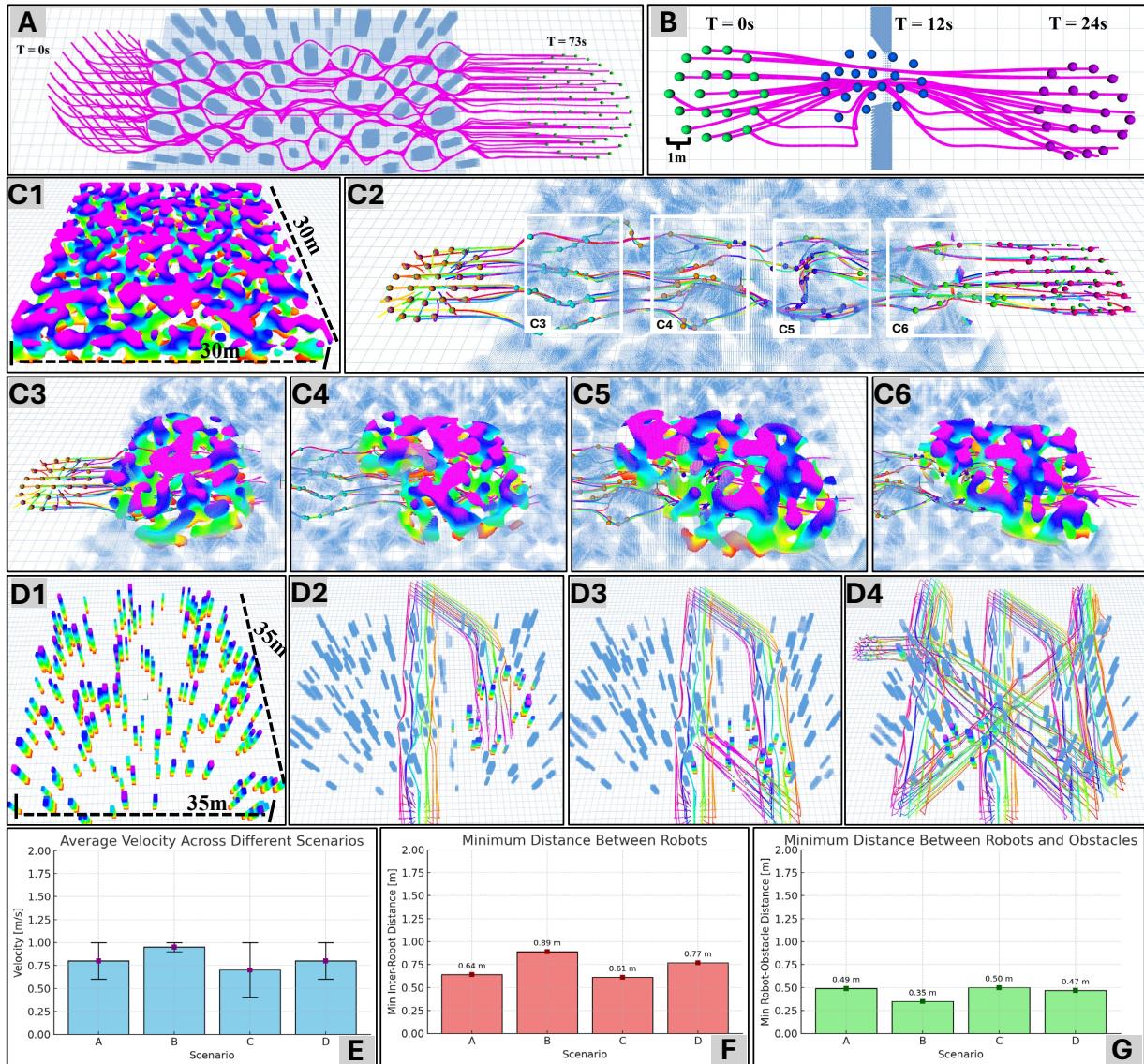


Fig. 3: Representative scenarios and evaluation metrics for swarm navigation. (A): $M = 60$ robots navigating through multiple static obstacles. (B): $M = 20$ robots passing through a narrow obstacle-constrained environment. (C1): Visualization of a complex 3D scenario with dense obstacles. (C2): $M = 30$ robots traversing the challenging 3D scene; (C3)-(C6) show local motion details. (D1): A randomly generated 3D forest-like environment, where a new environment is generated for each trial. (D2)-(D4): Trajectories of $M = 30$ robots navigating through the random forest. (E): Average swarm velocity under different scenarios. (F): Minimum inter-robot distances across scenarios. (G): Minimum robot-to-obstacle distances across scenarios.

eration. In contrast, other approaches tend to generate longer paths due to constrained optimization strategies that explicitly avoid trajectory intersection. In high-density environments, however, our method may encounter local congestion effects, leading to reduced flight speeds and increased energy consumption. The swarm's speed is influenced by the leading particle because SPH-based local interactions propagate deceleration through the group when the front robot slows down in response to environmental constraints. From a communication standpoint, our approach only requires broadcasting position data at 10 Hz, while existing methods typically transmit full trajectories such as B-spline control points or predicted motion states at a lower frequency of 5 Hz, often accompanied by additional mechanisms for temporal synchronization across

agents. This lightweight communication scheme significantly reduces bandwidth requirements and enhances scalability in large and cluttered environments.

B. Real World Experiments

To validate the proposed method in real-world settings, we conduct extensive indoor flight tests in both 2D and 3D environments using a swarm of 10 micro UAVs. Each quadrotor is equipped with omnidirectional vision and performs onboard localization [22], mapping [23], planning, and control without external infrastructure. Position sharing is achieved via Wi-Fi using UDP broadcasting, and the coordination gain is set to $k_{\text{vis}} = 6.5$ to reduce oscillations under network latency. The UAVs operate with a maximum velocity of 1.5 m/s.

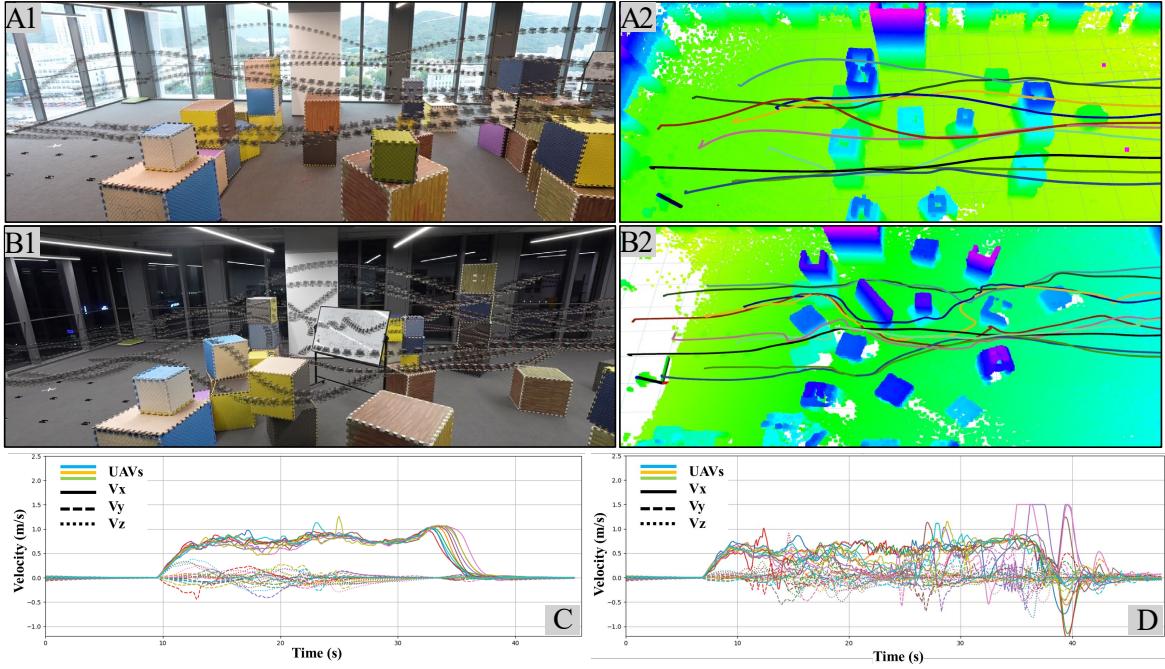


Fig. 4: Swarm navigation results from two real-world experiments with 10 UAVs. (A1): UAVs flying through a moderately cluttered environment. (B1): UAVs navigating a denser obstacle field with tighter spatial constraints. (A2): Corresponding trajectory visualization showing coordinated motion. (B2): Trajectories illustrating adaptive path adjustment in the complex scene. (C): Velocity profiles of all UAVs in the first experiment. (D): Velocity profiles in the more challenging environment.

TABLE II: METRICS COMPARISON UNDER VARYING OBSTACLE DENSITY d_{obs}

Metric Method	d_{obs} obs./m ²	Dist (m)	Time (s)	Rate (%)	Eng m^2/s^5	Comm (Mbps)	Vel (m/s)
EGO-swarm [6]	0.15	25.5±2.3	26.4±1.4	100	100±10	0.30±0.03	0.97
	0.25	28.9±2.5	29.4±1.5	100	138±12	0.30±0.03	0.98
	0.35	29.9±2.6	30.3±1.6	90	158±15	0.30±0.03	0.99
	0.45	32.5±2.9	34.9±2.0	75	180±18	0.30±0.03	0.93
MADER [9]	0.15	24.8±2.3	27.0±1.4	100	118±10	0.21±0.01	0.92
	0.25	28.1±2.5	30.1±1.6	100	140±12	0.21±0.01	0.93
	0.35	29.0±2.7	31.5±1.8	90	162±15	0.21±0.01	0.92
	0.45	30.8±3.0	36.0±2.2	85	185±18	0.21±0.01	0.86
LSC [20]	0.15	25.9±1.3	27.8±1.4	95	115±10	0.27±0.02	0.93
	0.25	28.5±1.4	30.9±1.5	90	138±12	0.27±0.02	0.92
	0.35	30.2±1.6	34.0±1.7	50	162±15	0.27±0.02	0.89
	0.45	33.3±1.9	38.1±2.3	40	186±18	0.27±0.02	0.87
NMPC [21]	0.15	26.1±2.4	28.3±1.5	100	110±10	0.33±0.03	0.92
	0.25	28.2±2.6	31.9±1.8	95	138±15	0.33±0.03	0.88
	0.35	30.5±3.8	35.7±2.1	65	163±18	0.33±0.03	0.85
	0.45	32.9±5.2	39.8±2.5	40	192±20	0.33±0.03	0.83
Proposed	0.15	23.1±1.2	26.8±1.3	100	112±10	0.075±0.01	0.86
	0.25	26.1±1.4	32.4±1.5	100	148±12	0.075±0.01	0.81
	0.35	28.9±1.5	39.1±1.7	95	164±15	0.075±0.01	0.74
	0.45	30.1±1.6	44.2±2.0	95	188±18	0.075±0.01	0.68

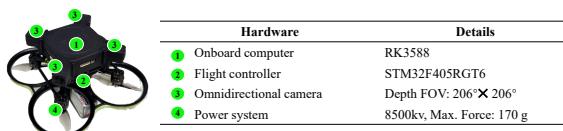


Fig. 5: Quadrrotor platform and its main parameters.

The experimental results are shown in Fig. 4. Fig. 4(A1) and Fig. 4(B1) display the swarm flight trajectories in environments of different complexity, where the collective motion and obstacle interactions are clearly visualized in Fig. 4(A2) and Fig. 4(B2). Fig. 4(C) and Fig. 4(D) show the velocity curves of all UAVs in the two scenarios. These results validate the effectiveness of the proposed coordination framework in achieving collision-free navigation under constrained conditions. We recommend that readers refer to the attached video for further details.

C. Scalability Analysis in Complex Obstacle Environments

We further analyze the system's scalability based on different swarm sizes and obstacle densities, and the results are given in Fig. 6. As the swarm size increases, the coordination time slightly increases, and the planning time remains nearly constant. Larger swarm sizes lead to longer coordination times because interaction forces spread more slowly through the group and require more steps for the swarm to reach a coordinated state. In confined environments, larger swarms face higher chances of spatial competition, which lowers the success rate. Higher obstacle density reduces feasible paths and causes a significant drop in navigation performance due to the lack of global guidance. Overall, the proposed method demonstrates good computational scalability. However, in scenarios where globally feasible paths are extremely limited, the swarm may fail to find a solution. In addition, due to the fully distributed nature of the planning framework, asynchronous decisions among agents in highly constrained spaces may lead to deadlocks or suboptimal routing outcomes.

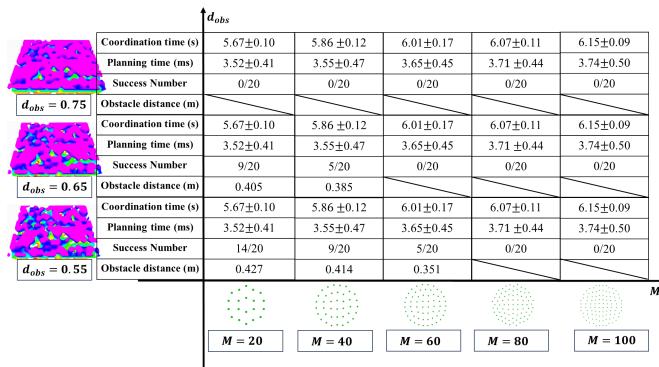


Fig. 6: Performance comparison under varying swarm sizes and obstacle densities. This figure shows that swarm scalability is constrained by both increasing obstacle density and swarm size.

V. CONCLUSION AND FUTURE WORK

In this paper, we propose a swarm navigation method based on smoothed particle hydrodynamics. We consider the swarm UAVs as a group of particles, utilizing density interactions, attraction, and repulsion forces between particles to maintain the dynamic equilibrium of the system. Meanwhile, we introduce a coordination mechanism that enables the particles to maintain a balance between swarm coordination and autonomous navigation. Compared to existing methods, our proposed swarm framework significantly reduces scalability challenges and communication overhead. We validated the proposed method through extensive simulations and real-world experiments.

However, current methods still require further improvement in the swarm's deformation capability. For instance, in a scenario with a narrow passage that allows only a single UAV to pass at a time, all UAVs may become stuck at the entrance, blocking one another and preventing further movement. This issue arises because SPH tends to maintain the coordination of the swarm, which weakens the swarm's ability to deform, making it unable to flow through such narrow spaces like a fluid. Future work will explore the integration of a global guidance vector field and active deformation control to enhance swarm adaptability and coordination efficiency in dense and constrained environments.

REFERENCES

- [1] W. Hönig, T. Kumar, L. Cohen, H. Ma, H. Xu, N. Ayanian, and S. Koenig, "Multi-agent path finding with kinematic constraints," in *Proceedings of the International Conference on Automated Planning and Scheduling*, vol. 26, 2016, pp. 477–485.
- [2] W. Hönig, J. A. Preiss, T. S. Kumar, G. S. Sukhatme, and N. Ayanian, "Trajectory planning for quadrotor swarms," *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 856–869, 2018.
- [3] J. Park and H. J. Kim, "Fast trajectory planning for multiple quadrotors using relative safe flight corridor," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 596–603.
- [4] J. Park, J. Kim, I. Jang, and H. J. Kim, "Efficient multi-agent trajectory planning with feasibility guarantee using relative bernstein polynomial," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 434–440.
- [5] J. Park and H. J. Kim, "Online trajectory planning for multiple quadrotors in dynamic environments using relative safe flight corridor," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 659–666, 2020.
- [6] X. Zhou, J. Zhu, H. Zhou, C. Xu, and F. Gao, "Ego-swarm: A fully autonomous and decentralized quadrotor swarm system in cluttered environments," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 4101–4107.
- [7] J. Hou, X. Zhou, Z. Gan, and F. Gao, "Enhanced decentralized autonomous aerial robot teams with group planning," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 9240–9247, 2022.
- [8] Z. Wang, C. Xu, and F. Gao, "Robust trajectory planning for spatial-temporal multi-drone coordination in large scenes," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 12 182–12 188.
- [9] J. Tordesillas and J. P. How, "Mader: Trajectory planner in multiagent and dynamic environments," *IEEE Transactions on Robotics*, vol. 38, no. 1, pp. 463–476, 2021.
- [10] S. Zhao, S. Ramakrishnan, and M. Kumar, "Density-based control of multiple robots," in *Proceedings of the 2011 American control conference*. IEEE, 2011, pp. 481–486.
- [11] Y. B. Mitikiri, M. Silic, and K. Mohseni, "Smoothed-particle-hydrodynamics for the control of robotic swarms, and parametric associations," *IEEE Transactions on Control of Network Systems*, vol. 8, no. 4, pp. 1942–1953, 2021.
- [12] L. C. Pimenta, G. A. Pereira, N. Michael, R. C. Mesquita, M. M. Bosque, L. Chaimowicz, and V. Kumar, "Swarm coordination based on smoothed particle hydrodynamics technique," *IEEE Transactions on Robotics*, vol. 29, no. 2, pp. 383–399, 2013.
- [13] X. Lei, S. Zhang, Y. Xiang, and M. Duan, "Self-organized multi-target trapping of swarm robots with density-based interaction," *Complex & Intelligent Systems*, vol. 9, no. 5, pp. 5135–5155, 2023.
- [14] S. Zhang, X. Lei, Z. Zheng, and X. Peng, "Collective fission behavior in swarming systems with density-based interaction," *Physica A: Statistical Mechanics and its Applications*, vol. 603, p. 127723, 2022.
- [15] S. Zhang, X. Lei, X. Peng, and J. Pan, "Heterogeneous targets trapping with swarm robots by using adaptive density-based interaction," *IEEE Transactions on Robotics*, 2024.
- [16] G. Vásárhelyi, C. Virág, G. Somorjai, T. Nepusz, A. E. Eiben, and T. Vicsek, "Optimized flocking of autonomous drones in confined environments," *Science Robotics*, vol. 3, no. 20, p. eaat3536, 2018.
- [17] R. Li, J. Lyu, A. Wang, R. Yu, D. Wu, and B. Xin, "Flagdroneracing: An autonomous drone racing system," *Unmanned Systems*, 2023.
- [18] A. Patwardhan, R. Murai, and A. J. Davison, "Distributing collaborative multi-robot planning with gaussian belief propagation," *IEEE Robotics and Automation Letters*, vol. 8, no. 2, pp. 552–559, 2022.
- [19] B. Zhou, F. Gao, L. Wang, C. Liu, and S. Shen, "Robust and efficient quadrotor trajectory generation for fast autonomous flight," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3529–3536, 2019.
- [20] J. Park, D. Kim, G. C. Kim, D. Oh, and H. J. Kim, "Online distributed trajectory planning for quadrotor swarm with feasibility guarantee using linear safe corridor," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4869–4876, 2022.
- [21] E. Soria, F. Schiano, and D. Floreano, "Predictive control of aerial swarms in cluttered environments," *Nature Machine Intelligence*, vol. 3, no. 6, pp. 545–554, 2021.
- [22] T. Qin, P. Li, and S. Shen, "Vins-mono: A robust and versatile monocular visual-inertial state estimator," *IEEE transactions on robotics*, vol. 34, no. 4, pp. 1004–1020, 2018.
- [23] B. Zhou, J. Pan, F. Gao, and S. Shen, "Raptor: Robust and perception-aware trajectory replanning for quadrotor fast flight," *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 1992–2009, 2021.