



Agile target capture with UAV swarm in dense environments

Ping Zhang¹ , Xin Yan¹ , Hongwei Liu¹ , Ran Liu² , Yury Furletov³, Jianwen Huo¹

Keywords:

Swarm control, state estimation, capture strategy, trajectory planning

Citation: Zhang, P.; Yan, X.; Liu, H.; Liu, R.; Furletov, Y.; Huo, J. Agile target capture with UAV swarm in dense environments. *Intell. Robot.* 2026, 6(3), 560-81. <https://dx.doi.org/10.20517/ir.2026.27>

Received: 26 May 2026

First Decision: 29 Jun 2026

Revised: 29 Jul 2026

Accepted: 18 Aug 2026

Published: 17 Sep 2026

Academic Editor:

Hao Zhang

Copy Editor:

Pei-Yun Wang

Production Editor:

Pei-Yun Wang

Abstract

Rapidly capturing moving targets is critical for maintaining border security and public safety. However, traditional swarm-based target acquisition models often rely on the navigator's observations and focus on fixed-formation capture methods. When target positions are uncertain and moving rapidly, deadlocks may occur due to target loss. To address this issue, this paper formulates a sliding-window pose graph optimization framework combined with trajectory prediction to estimate and predict the three-dimensional trajectory of agile targets in real time. By adopting a virtual center-of-mass extension strategy to optimize non-uniform circular capture formations, compact target enclosure and obstacle avoidance are achieved. This paper further integrates field-of-view awareness with yaw coordination by formulating a multi-constraint optimization problem. The proposed approach focuses on target-state fusion, prediction, and cooperative capture, demonstrating strong performance in multiple unmanned aerial vehicles (UAVs) cooperative tracking, target visibility maintenance, obstacle avoidance planning, and collision avoidance in dense environments.

1. INTRODUCTION

With advancements in drone technology and onboard sensing capabilities, expectations are growing for drone swarm systems to intercept agile, non-cooperative targets in dense environments (as shown in [Figure 1](#)). Typical scenarios include low-altitude defense of critical infrastructure, border surveillance in mountainous terrain, and emergency response in urban canyons. Under such demanding conditions, a single drone^[1] tracking a fast-moving target is highly susceptible to obscuration by buildings, vegetation, or infrastructure, and its local observational data suffers significant degradation due to sensor noise and changing viewpoints. Consequently, swarm-based collaborative target acquisition is emerging as a critical breakthrough, though it presents numerous challenges. The entire swarm must rely on collaborative perception^[2], prediction^[3], and planning^[4] capabilities to maintain a shared understanding of the target's state and execute safe, time-critical capture maneuvers.



¹School of Information and Control Engineering, Southwest University of Science and Technology, Mianyang 621000, Sichuan, China.

²School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798, Singapore.

³Chair of ground means of transportation, Moscow Polytechnic University, Moscow 101000, Russia.

Correspondence to: Prof. Jianwen Huo, School of Information and Control Engineering, Southwest University of Science and Technology, Mianyang 621000, Sichuan, China. E-mail: huojianwen2008@hotmail.com

These challenges exist not only within individual drones but are further amplified by the tight coupling between swarm systems. First, at the perception and state estimation level, the core difficulty stems from data uncertainty and heterogeneity. Targets frequently execute highly nonlinear and aggressive maneuvers. Visual and ranging data collected by different unmanned aerial vehicles (UAVs) at different times inherently exhibit asynchrony^[5]. Direct fusion of such unmodeled data leads to biased target state estimates, contaminating downstream planning modules. More complexly, lacking global positioning information, the swarm must construct a global state estimate of the target within its own coordinate system - a classic consensus problem^[6]. Any estimation error amplifies during collaborative decision-making, leading to formation chaos or capture failure.

At the trajectory prediction^[7] level, the challenge lies in balancing the model's expressive power with computational efficiency. Even when the target's instantaneous state is known, the swarm must predict its future trajectory in real time to form and update the capture formation before the target moves. This prediction model must possess sufficient expressive power to capture highly maneuverable behaviors such as sharp turns, acceleration, and deceleration; yet it must also be compact enough to be embedded within an online optimizer to meet real-time computational demands. Linear or simplified uniform acceleration models prove entirely inadequate in such scenarios, while overly complex models impose prohibitive computational burdens.

At the formation control^[8] and trajectory planning^[9] level, this problem evolves into a complex multi-objective optimization challenge. In dense environments, coordinated capture requires the swarm to achieve a delicate balance among multiple conflicting objectives. UAVs must rapidly contract their formation to encircle targets and prevent escape, while strictly adhering to safe inter-vehicle distances - avoiding collisions with obstacles while remaining within their own dynamic limits. Simple potential field methods or rule-based control strategies are prone to local deadlocks, oscillatory motion, or overly conservative behavior that misses critical capture opportunities.

A critical yet often overlooked challenge lies in the active coordination of perception and field of view (FOV)^[10]. Since all capture decisions ultimately rely on visual or ranging observations, the swarm must actively synchronize its yaw angles with its field-of-view direction. This creates a dilemma: on one hand, all drones must maintain continuous, reliable target lock to sustain "visual contact"; on the other hand, the swarm's overall perception architecture should prioritize outward observation to maximize situational awareness and detect unknown obstacles or threats early. This significantly increases the complexity and dimensionality of the system's planning.

Reliably capturing agile moving targets with drone swarms in complex environments is a classic "perception-prediction-planning-coordination" closed-loop problem. The overall framework of the system is shown in [Figure 2](#). This paper addresses the aforementioned challenge and makes the following contributions:

1. Target Estimation and Prediction: Proposes a collaborative target state estimation and prediction method, which integrates multi-UAV measurement data within a tightly coupled pose graph optimization (PGO) framework; Bézier curves are employed to achieve agile short-term trajectory prediction.
2. Capture in Dense Environments: Proposes an optimization-based capture formation strategy for dense environments, expanding outward from a virtual center of mass. This strategy synergistically achieves enclosure quality, obstacle avoidance, and inter-UAV safety through multiple cost functions.

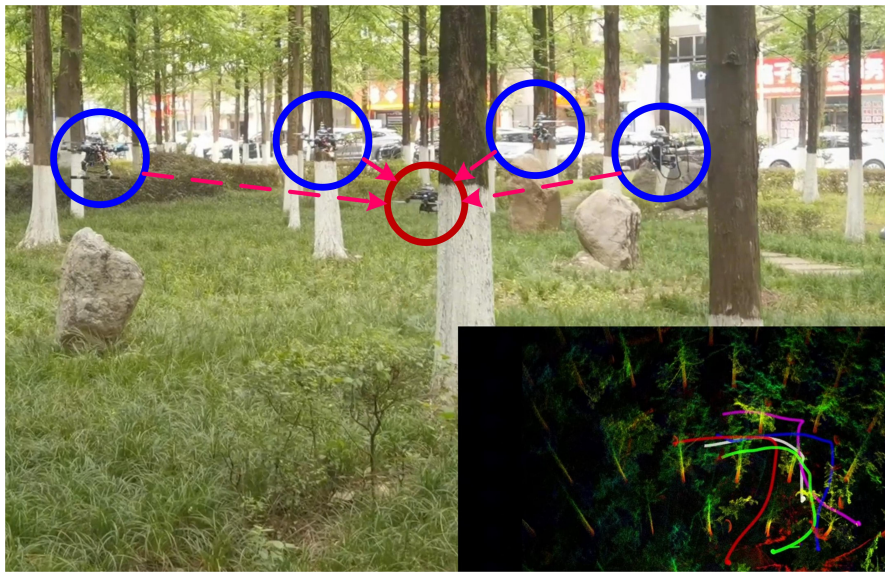


Figure 1. Agile target capture system for UAV Swarms. Blue circles represent capture drones, and red circles represent target drones. The bottom-right corner displays a real-time environment map and trajectory graph, where different colored curves represent the historical trajectories of the capture drone and the target drone, respectively. Image source: physical experiments and hand-drawn Visio images. UAV: Unmanned aerial vehicle.

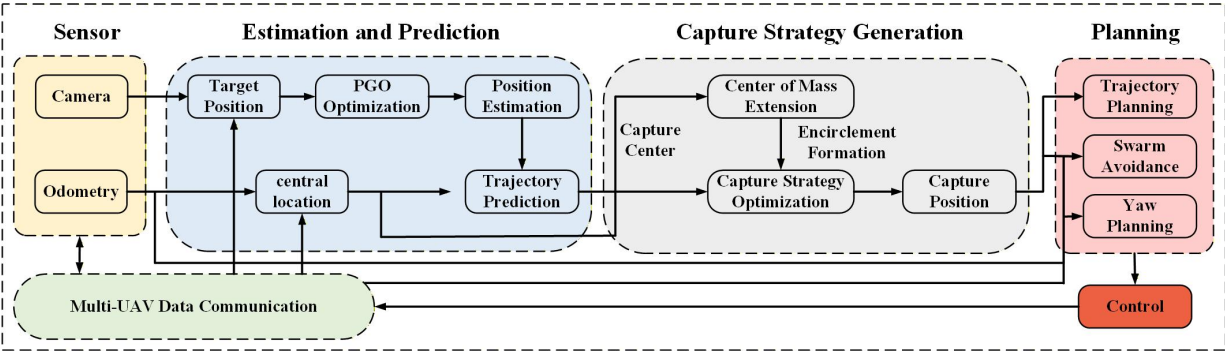


Figure 2. System Overview: This system is primarily divided into three aspects: Target Estimation and Prediction, Capture in Dense Environments, and Swarm Trajectory Planning. It employs a centralized computing approach to generate control commands for all drones, which are then transmitted via ROS communication. ROS: Robot operating system; PGO: pose graph optimization.

- 3. Swarm Trajectory Planning: Proposes a collision avoidance mechanism among unmanned aerial vehicles and a yaw coordination mechanism based on visual perception. This mechanism can predict future trajectory conflicts and adjust the yaw angle in real time while integrating multiple constraints.

2. RELATED WORK

2.1. Swarm collaborative target state estimation

To improve single-UAV state estimation accuracy, Li *et al.* [11] proposed a resilient unscented Kalman filtering fusion method with dynamic event-triggered mechanisms, solving the problem of computing cross-covariances between local filters through sequential covariance intersection fusion strategies. Zhou *et al.* [12] further improved filtering algorithms by proposing an adaptive robust unscented Kalman filter based on QR decomposition and singular value decomposition, effectively suppressing interference from outliers and non-Gaussian noise. Achieving accurate state estimation of agile targets by multiple UAVs in denied environments requires solving complex localization and information fusion problems. Dong *et al.* [13] proposed a method for dynamic target tracking using time-variant radio maps, developing grid-based and particle filter-based tracking algorithms based on time-varying received signal strength sequences from multiple UAVs, combined with radio maps generated from real-world terrain data. He *et al.* [14] further developed this concept by proposing a radio map-assisted multi-UAV target search method, addressing the problem of RSS being susceptible to terrain effects through a minimum mean square error estimator for memoryless observations. However, relying solely on wireless signal localization has limitations; thus, Wang *et al.* [15] introduced visual tracking methods, achieving 78.2% tracking accuracy by incorporating efficient channel attention modules and Swin Transformer blocks in the backbone network combined with the BYTE strategy. Li *et al.* [16] addressed visual occlusion issues by designing an image-based tracking controller based on YOLO deep neural networks and unscented Kalman filters, improving tracking performance under occlusion through motion models derived from quadratic programming.

To prevent issues arising from asynchronous communication, reference [17] explored distributed estimation methods, using single-timescale distributed estimation protocols to process time-of-arrival information shared by neighboring UAVs, reducing communication burden compared to multi-timescale protocols. Franchi *et al.* [18] addressed the mutual localization problem under anonymous relative position measurements, proposing a two-stage localization system based on the MultiReg algorithm, optimizing best estimates through multiple extended Kalman filters.

In batch data optimization, PGO is a technique derived from factor graph theory [19–22], which is crucial for robot localization and mapping in SLAM. The PGO method is widely used in multi-robot collaborative state estimation. Initial PGO applications utilized centralized servers to solve pose graph problems [20], with the core idea being to construct a factor graph to estimate the pose. This paper applies this method by analogizing the target to a robot in SLAM and the capture drone to a landmark in SLAM; it maps the capture drone's observations of the target to the target's observations of surrounding map points. This enables the use of PGO optimization methods to perform a tightly coupled estimation of the target's state.

2.2. Swarm target tracking and planning

At the cooperative tracking strategy level, Wu *et al.* [23] proposed a context-aware adaptive feature fusion method, achieving local-to-global situational analysis through graph attention convolutional networks and self-attention mechanisms. Peng *et al.* [24] considered limited field-of-view constraints, proposing a Normalized Actor with Graph Attention Critic algorithm, enhancing pursuers' search and pursuit capabilities through obstacle-target graph attention networks. Rao *et al.* [25] combined model predictive control with standoff algorithms, achieving formation maintenance and trajectory planning in complex 3D environments through fully connected communication topologies. Tang *et al.* [26] divided the cooperative attack process into cruise and attack phases, achieving multi-UAV cooperative target state estimation based on unscented Kalman filters with target localization accuracy reaching 10 meters. For system integration, Khosravi *et al.* [27] developed a search and detection system based on Bayesian inference and residual neural networks, significantly reducing mission execution time through optimized path planning algorithms. Mason *et al.* [28] utilized LoRaWAN communication and extended

constant turn rate and acceleration motion models, achieving reliable tracking of UAV swarms up to thousands in number at distances reaching 4 kilometers.

Early studies of cooperative UAV swarm pursuit of agile targets primarily focused on fundamental trajectory planning; for instance, Chen *et al.* [29] proposed an online trajectory generation method based on quadratic programming, embedding tracking errors and control costs into the cost function to achieve millisecond-level real-time trajectory generation. As research progressed, Chen *et al.* [30] addressed pursuit under speed disadvantage, analyzing strategies for slower UAV swarms to capture faster intruders using Apollonius sphere properties, transforming the game problem into swarm control through integrating coverage control and control barrier functions.

In trajectory planning and obstacle avoidance control, the EGO-Swarm system by Zhou *et al.* [9] adopts a gradient-based local planning framework, transforming collision risks into optimization penalty terms and introducing topological trajectory generation to achieve decentralized autonomous navigation. The E2CoPre method by Huang *et al.* [31] combines artificial potential fields and particle swarm optimization, achieving a balance between energy efficiency and collision avoidance. Jeon *et al.* [32] and Han *et al.* [3] respectively proposed tracking solutions for complex environments, with the former using covariance optimization to predict target motion and the latter employing dynamic search to generate spatiotemporal optimal trajectories. The Elastic Tracker framework by Ji *et al.* [33] achieves joint optimization of safety and visibility through occlusion-aware path planning.

2.3. Swarm cooperative capture

Collaborative perception forms the technological foundation for precise encirclement. Zhao *et al.* [34] proposed a two-layer distributed control architecture, where the leader layer obtains bearing information through target detection sensors while the follower layer uses relative distance sensors to perceive neighboring UAVs, eliminating dependence on global communication. Wang *et al.* [35] proposed perception-complementarity-driven trajectory generation, enhancing system perception through visual mutual observation between UAVs, avoiding dependence on prior maps while improving target visibility. Reinforcement learning brings breakthrough advances to cooperative decision-making. Chen *et al.* [36] proposed multi-agent reinforcement learning based on extrinsic and intrinsic rewards, simulating human curiosity mechanisms to cooperatively control heterogeneous UAVs. Chen *et al.* [37] applied deep reinforcement learning to pursuit-evasion scenarios, solving partial observability through enhanced evader prediction networks. Song *et al.* [38] combined deep neural networks with model predictive control, solving target loss problems through bystander algorithms. The high-sample-efficiency MARL method by Cheng *et al.* [39] employs hypernetwork-based embedding attention mechanisms, reducing training time to one-fifth and collision counts by an order of magnitude. In cooperative interception, Chen *et al.* and reference [30,40] proposed a “centroid expansion-containment” strategy, optimizing 3D interception paths via control barrier functions for low-speed UAVs. Recent formation-maintenance and collision-avoidance studies further discuss constrained coordination in dense environments [41,42].

3. METHODS

3.1. Target estimation and prediction

This section models target state estimation as a nonlinear optimization problem directly defined in the observed drone body coordinate system. By synergistically applying K-means clustering and PGO techniques, a smooth 3D trajectory of the target is derived from noisy measurement data. A sliding-window Bézier curve extrapolation method provides short-term predictions that adapt to the latest estimated motion patterns.

3.1.1. Target state estimation

In the real-time swarm capture operation described in this paper, N pursuer UAVs collaboratively estimate the state of a target. The global pose of the i -th pursuer UAV at timestamp t is assumed to be available as $(\mathbf{R}_{i,t}, \mathbf{p}_{i,t}) \in SO(3) \times \mathbb{R}^3$ in a common world frame, where $\mathbf{R}_{i,t}$ and $\mathbf{p}_{i,t}$ denote the rotation and translation of the pursuer body frame with respect to that frame. This pose can be provided by motion capture, visual/LiDAR SLAM, or a cooperative localization module; the localization and mapping backend is not the contribution of this paper. To reduce computational load, we estimate only the global position of the target UAV in three dimensions:

$$\mathbf{p}_t = [x_t, y_t, z_t]^T \in \mathbb{R}^3 \quad (1)$$

$\mathbf{z}_{i,t}^b$ indicates the 3D position observation of the target by UAV i in its own attitude at timestamp t . With the known pursuer pose, the observation is transformed into the global frame by

$$\mathbf{z}_{i,t}^w = \mathbf{R}_{i,t} \mathbf{z}_{i,t}^b + \mathbf{p}_{i,t} \quad (2)$$

where $\mathbf{z}_{i,t}^w$ is the global position measurement inferred from the i -th pursuer UAV.

All target measurements collected within a short fusion interval are assigned to the same optimization timestamp. This design reduces the impact of small communication delays without requiring strict hardware timestamp synchronization. Due to noise and occasional outliers, observation clustering is performed before graph optimization. Specifically, we need to perform K-means clustering on all observations $\{\mathbf{z}_{i,t}^w\}_{i=1}^N$ at each time step t to obtain a preliminary target estimate.

$$\{C_t^{(k)}\} = \text{KMeans}(\{\mathbf{z}_{i,t}^w\}_{i=1}^N), \quad \mathcal{I}_t = \arg \max_k |C_t^{(k)}| \quad (3)$$

Where $\{C_t^{(k)}\}$ denotes the obtained clusters, and $\mathcal{I}_t$ represents the set of interior points within the dominant cluster with the largest cardinality. After filtering out observations outside $\mathcal{I}_t$, the interior point set undergoes weighted fusion to yield a single stable and robust local measurement at time t :

$$\bar{\mathbf{z}}_t^w = \frac{\sum_{\mathbf{z}_{i,t}^w \in \mathcal{I}_t} w_{i,t} \mathbf{z}_{i,t}^w}{\sum_{\mathbf{z}_{i,t}^w \in \mathcal{I}_t} w_{i,t}}, \quad w_{i,t} \geq 0 \quad (4)$$

Here, $w_{i,t}$ reflects the confidence level of the i -th pursuer observation, derived from the covariance of the observation data. The fused measurement $\bar{\mathbf{z}}_t^w$ is used as the instantaneous estimate of the target's global position. Although it mitigates the impact of multi-view outliers, it remains subject to residual random errors and is therefore refined by the subsequent PGO module.

Compared to the Kalman filter method, which heavily relies on the target motion model, PGO provides a more robust and convenient tightly coupled method that does not depend on continuous observations of a single node. To achieve a globally optimal consensus target trajectory while avoiding computational overload, we address these challenges by integrating a pose map optimization algorithm with a sliding window approach. The detailed factor graph construction process is shown in [Figure 3](#).

We construct a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where the vertices correspond to the target locations at different timestamps:

$$\mathcal{V} = \{\mathbf{p}_t \mid t \in \mathcal{T}\} \quad (5)$$

Each vertex $\mathbf{p}_t$ indicates the target position to be estimated. The edge set $\mathcal{E}$ consists of four types of factors: (1) marginalized prior residual r_p from the previous window; (2) observation residuals aligned with fused measurements; (3) relative motion residuals across multiple timestamps within a sliding window; (4) second-order trajectory smoothing residuals.

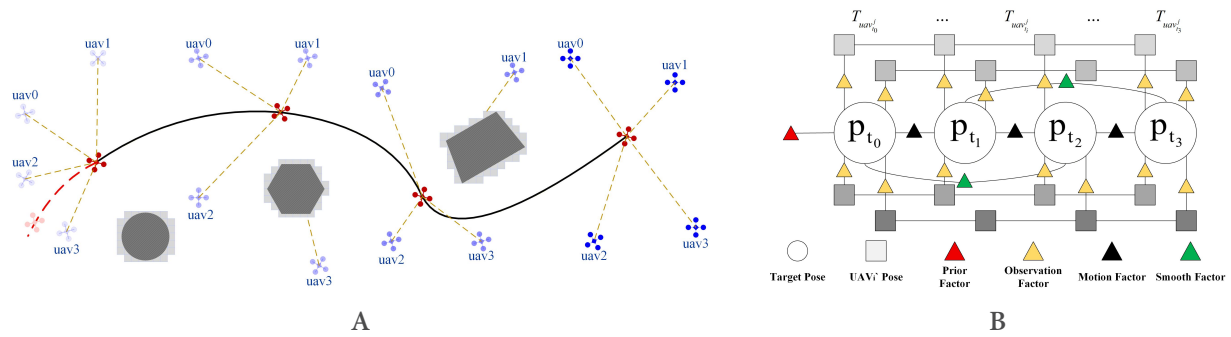


Figure 3. PGO target state estimation. (A) Data acquisition and factor distribution of the active sliding window across four consecutive frames during UAV swarm capture; (B) Factor graph constructed by the PGO method. As the window moves forward, the oldest state is marginalized into the prior residual. The red dashed lines and red triangles represent prior residual information; the yellow dashed lines and yellow triangles represent observation residuals; the black trajectory and black triangles represent motion residuals; and the green triangles represent the smoothing residuals of the optimized trajectory. PGO: Pose graph optimization; UAV: unmanned aerial vehicle.

The term r_p is the prior factor obtained after marginalizing the oldest target state when the window slides. It preserves information from past measurements and is computed using Schur-complement marginalization [6].

At each timestamp t , Observation residual constrains p_t to be close to the fused observation:

$$\mathbf{e}_t^u = \mathbf{p}_t - \bar{\mathbf{z}}_t^w \quad (6)$$

The corresponding information matrix Ω_t^u is determined by the uncertainty of $\bar{\mathbf{z}}_t^w$. To encode target motion continuity, we compute a relative displacement measurement between two consecutive fused observations:

$$\Delta \bar{\mathbf{z}}_{ij}^w = \bar{\mathbf{z}}_i^w - \bar{\mathbf{z}}_j^w \quad (7)$$

Relative motion residual between p_i and p_j is then defined as

$$\mathbf{e}_{ij}^b = (\mathbf{p}_i - \mathbf{p}_j) - \Delta \bar{\mathbf{z}}_{ij}^w \quad (8)$$

The second-order smoothing term constraint primarily constrains the target acceleration, ensuring the estimated value does not deviate excessively. The constraint error term is as follows:

$$\mathbf{e}_i^a = (\mathbf{p}_{i+1} - \mathbf{p}_i) - (\mathbf{p}_i - \mathbf{p}_{i-1}) \quad (9)$$

This factor encourages the optimized trajectory to follow the measured inter-frame displacement, while allowing errors to be distributed across the window rather than accumulating at the latest estimate. The information matrix Ω_{ij}^b [6] reflects the confidence of the relative displacement, which can be approximated by the propagated covariance of $\bar{\mathbf{z}}_i^w$ and $\bar{\mathbf{z}}_{i-1}^w$.

After constructing all vertices $\mathcal{V}$ and edges $\mathcal{E}$, they are incorporated into a sliding window pose optimization problem. To further address outliers during the fusion process, we incorporate a robust kernel function $\rho(\cdot)$ [6] into the residual construction.

$$\rho(s) = \begin{cases} s, & s \leq \delta^2 \\ 2\delta\sqrt{s} - \delta^2, & s > \delta^2 \end{cases} \quad (10)$$

where s is the squared Mahalanobis residual and δ is the kernel threshold. By minimizing the following optimization problem, a relatively accurate estimate of the target trajectory state can be obtained.

$$\min_{\{\mathbf{p}_t\}} r_p + \lambda_1 \sum_{t \in \mathcal{T}} \rho(\|\mathbf{e}_t^u\|_{\Omega_t^u}^2) + \lambda_2 \sum_{(i,j) \in \mathcal{E}_b} \rho(\|\mathbf{e}_{ij}^b\|_{\Omega_{ij}^b}^2) + \lambda_3 \sum_{i=t-L+2}^{t-1} \|\mathbf{e}_i^a\|_{\Omega_i^a}^2 \quad (11)$$

where $\mathcal{T} = \{t - L + 1, \dots, t\}$ denotes a fixed-size sliding window of length L , ε_p denotes the relative-motion edges inside the window, and Ω^g is the smoothing information matrix. Upon arrival of a new fusion measurement $\bar{z}_{i+1}^v$, a new vertex and corresponding unary/binary edges are added; the oldest vertex is marginalized into r_p if the window is full, and the optimization problem is solved using the Ceres solver. Where λ_1 , λ_2 and λ_3 represent the weights of the observation residual, the relative motion residual and the smoothing residual, respectively, and the specific values will be explained in subsequent experiments. The refined target position estimate is fed into downstream planning and control modules.

3.1.2. Target trajectory prediction

This paper uses Bernstein basis polynomials, called Bézier curves, to describe the target prediction trajectories. n th order Bézier curve is denoted as:

$$B(t) = \sum_{i=0}^n c_i b_n^i(t) \quad (12)$$

In this context, b_n^i denotes an n^{th} -order Bernstein polynomial basis, as outlined in reference [3]. The set of control points for the Bézier curve, denoted by $[c_0, c_1, \dots, c_n]$, is a crucial component in the analysis.

The 3D position of the target observed in the global frame at time t is denoted as $P_t \in \mathbb{R}^3$. Subsequently, a First-In, First-Out (FIFO) queue of length L is established to store the preceding observations and their corresponding timestamps (The value of L is related to the system operating frequency). The queue is denoted as $Q_{\text{target}} = [q_0, q_1, \dots, q_L]$, where $q_i = \{p_{t_i}, t_i\}$. Q_{target} contains the time range $[t_1, t_L]$, where t_L equals the current time. Upon the acquisition of a new target observation, a corresponding target prediction trajectory, denoted $B(t)$, is generated through the fitting of preceding observations.

As time elapses, the confidence of prior observations invariably diminishes. Consequently, the queue length L is fixed with precision, while values that fall outside the designated range are discarded. The trajectory is extrapolated to $(t_l, t_{pre}]$ during which the target motion is predicted.

In the capture process, the hyperbolic tangent function $\tanh(\cdot)$ is employed to select the prediction horizon according to the distance d between the center of mass and the current target location:

$$t_{pre} = f(d) = t_l + \tanh(d) \cdot t_{step} \quad (13)$$

where t_{step} denotes a preset prediction time step. Figure 4 shows the predicted trajectory of the target in a dense environment.

3.2. Capture in dense environments

This section introduces a virtual center-of-mass representation for the group. Using numerical optimization methods, capture points are selected and optimized for each UAV by minimizing a composite cost function. This cost function comprehensively accounts for enclosure tightness, obstacle spacing, and collision avoidance among UAVs, while promoting a uniform angular distribution under weaker environmental constraints.

3.2.1. Capture queue

At this stage, the UAV swarm position is known in three dimensions. The capture order and circular-formation assignment are planned on the horizontal projection because encirclement mainly constrains the target in the xoy plane; the altitude component is still handled by the 3D trajectory planner. Let $P_{i,t}^{xy} \in \mathbb{R}^2$ be the horizontal projection of UAV U_i , and let the projected centroid be $C = \frac{1}{N} \sum_{i=1}^N P_{i,t}^{xy}$. Let the projected target capture point be T^{xy} . The capture direction is then defined as $\vec{v} = T^{xy} - C$.

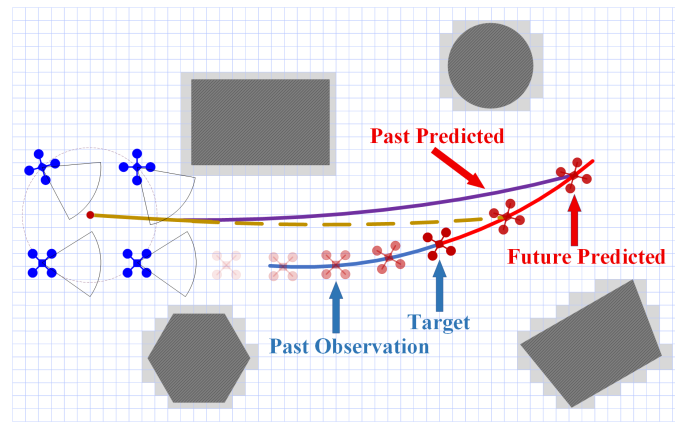


Figure 4. Trajectory prediction: the red dot on the left side of the picture represents the center of mass of the drone swarm, which is the center for surrounding and capturing the target. The blue curve represents the historical trajectory of the target, and the trajectory is predicted based on the first L historical observations. The red curve represents the expected trajectory. The brown curve represents the target prediction of the previous frame, while the purple curve represents the target prediction of the current frame. By piecing these trajectories together, the central point of the real-time captured trajectory is formed.

For each UAV U_i , define the projected vector with respect to the center of mass: $\vec{r}_i = P_{i,t}^{xy} - C$. Calculate the angle of this vector with respect to the capture direction $\vec{v}$. Let $\alpha_i = \text{mod}(\theta(\vec{r}_i) - \theta(\vec{v}), 2\pi)$, where $\theta(\vec{u})$ denotes the angle of the vector $\vec{u}$ with the x-axis, and takes on the range of $[0, 2\pi)$; here it is ordered in clockwise direction (i.e., the angle is gradually increasing from 0). Sort all α_i , and remember that the sorted UAVs are re-assigned the serial number (i.e., the initial capture order) as $S = (U_1, U_2, \dots, U_N)$, which satisfies $0 \leq \alpha_1 \leq \alpha_2 \leq \dots \leq \alpha_N \leq 2\pi$.

In the context of the center of mass position, if a UAV satisfies $|\vec{r}_i| = 0$, it can be deduced that said UAV is situated within the “center of mass region.” In order to ensure the continuity of the capture formation, the UAV is inserted between two neighboring UAVs in the circular formation in an orderly manner. All possible insertion sequences $I(J)$ are obtained, and all such candidate sequences are written into a FIFO candidate queue Q_{adj} .

If a set of candidate drones satisfies the condition of dense drone angle $|\Delta\alpha| < \delta$, where δ is a predefined small threshold, the set of these drones is denoted as J . The objective is to obtain all possible insertion sequences, denoted by $\Pi(J)$, by performing permutations and combinations of all drones in J . A specific permutation, denoted by $\sigma(J)$, is then selected and inserted into the spacing position of the original sequence to form a candidate neighboring sequence. The resultant candidate sequences are then written into a candidate queue, Q_{adj} , in a FIFO manner. The formation of the drone swarm capture team is shown in Figure 5 below.

$$S(U_1, U_2, \dots, U_i, \sigma(J), U_{i+1}, \dots, U_N). \quad (14)$$

The generation of this initial capture queue significantly accelerates the computation of the objective function, ensuring rapid capture decisions can be made for the target.

3.2.2. Generate capture points

In sparse environments, drone swarms can form circular formations with uniform angular spacing to capture targets. However, in dense environments, factors such as obstacles, in-swarm collisions, and rapid target acquisition make strictly uniform formations highly risky. To address this, we allow non-uniform angular spacing while maintaining a common capture radius to extend the capture formation.

Let $S = (U_{S(1)}, \dots, U_{S(N)})$ denote a candidate capture sequence in the queue Q_{adj} , and Let P denote the predicted value of the target mentioned earlier. For the j -th UAV in the sequence S , we assign an individual angular

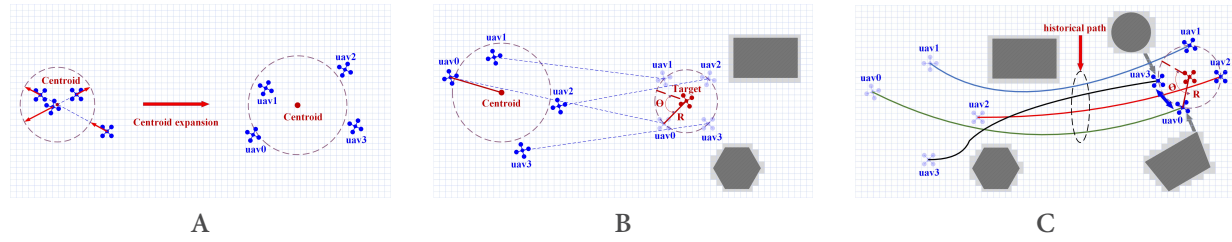


Figure 5. Target capture strategy generation: (A) illustrates the formation of a capture configuration by the drone swarm. The red dot represents the center of mass position of the swarm; (B) shows the capture points for each drone obtained by minimizing the objective function after the capture formation is established, when the surrounding area of the target is relatively open; (C) presents trajectory diagrams for dynamic target capture in dense environments. Curves of varying shades represent the historical trajectories of individual drones, with gray and blue arrows indicating obstacle avoidance factors and inter-drone collision avoidance factors, respectively. (C) also demonstrates how solutions to the objective function during capture alter the initial formation.

variable θ_j and a shared capture radius R , where $\Theta = [\theta_1, \dots, \theta_N]^T$ collects all angular parameters. In the special case where $\theta_j = \theta_0 + 2\pi(j-1)/N$ degenerates to the uniform circular formation. To rapidly respond to the capture of cluster targets, we constructed a comprehensive cost function that explicitly considers position fitting, tight enclosure, obstacle avoidance, intra-group collision avoidance, and formation regularity. By performing parallel computations across different candidate sequences s , we obtained the optimal capture queue and associated capture parameters (Θ, R) .

The desired capture point of the j -th UAV on the generalized circular arc is defined as

$$P_j^*(\Theta, R) = P + R \begin{bmatrix} \cos \theta_j \\ \sin \theta_j \end{bmatrix}, \quad j = 1, \dots, N \quad (15)$$

First, a fitting term measures the deviation between the actual UAV positions and their desired capture points:

$$J_{\text{fit}}(S, \Theta, R) = \sum_{j=1}^N \|P_{S(j)} - P_j^*(\Theta, R)\|^2 \quad (16)$$

To encourage groups to form a compact encirclement around the target while ensuring safe capture, without exceeding the minimum capture radius of $R_{\min}$, we introduce a distance regularization term.

$$J_{\text{dist}}(R) = (R - R_{\min})^2 \quad (17)$$

Environmental obstacles are considered through a soft penalty on the distance between capture points and obstacle boundaries. Let $d_{j,k}$ denote the shortest distance from $P_j^*(\Theta, R)$ to the surface of the k -th obstacle, and let d_{obs} be a safety margin. We define a hinge-type penalty function

$$\phi_{\text{obs}}(d) = [\max(0, d_{\text{obs}} - d)]^2 \quad (18)$$

and accumulate it over all UAV–obstacle pairs:

$$J_{\text{obs}}(\Theta, R) = \sum_{j=1}^N \sum_{k=1}^K \phi_{\text{obs}}(d_{j,k}) \quad (19)$$

Similarly, to prevent intra-swarm collisions in the final capture formation, we penalize small pairwise distances between capture points. Let

$$d_{j\ell} = \|P_j^*(\Theta, R) - P_\ell^*(\Theta, R)\|_2 \quad (20)$$

and let d_{col} denote a required minimum spacing. We define

$$\phi_{\text{col}}(d) = [\max(0, d_{\text{col}} - d)]^2 \quad (21)$$

and the collision avoidance cost as

$$J_{\text{col}}(\Theta, R) = \sum_{1 \leq j < \ell \leq N} \phi_{\text{col}}(d_{j\ell}) \quad (22)$$

To maximize the capture rate of targets and make escape difficult in any direction, we introduced formation rules. Define the angular gaps along the circular formation as

$$\Delta\theta_j = \begin{cases} \theta_{j+1} - \theta_j, & j = 1, \dots, N-1 \\ \theta_1 + 2\pi - \theta_N, & j = N \end{cases} \quad (23)$$

and compare them with the ideal uniform spacing $2\pi/N$. The regularity cost is given by

$$J_{\text{reg}}(\Theta) = \sum_{j=1}^N (\Delta\theta_j - \frac{2\pi}{N})^2 \quad (24)$$

When obstacles and collision constraints are inactive, this term encourages the angular parameters to converge back to a nearly uniform circular distribution.

Combining the above components, the dense-environment capture cost for a candidate sequence s is formulated as

$$\begin{aligned} \tilde{J}(S, \Theta, R) = & w_{\text{fit}} J_{\text{fit}} + w_{\text{dist}} J_{\text{dist}} \\ & + w_{\text{obs}} J_{\text{obs}} + w_{\text{col}} J_{\text{col}} + w_{\text{reg}} J_{\text{reg}} \end{aligned} \quad (25)$$

where $w_{\text{fit}}, w_{\text{dist}}, w_{\text{obs}}, w_{\text{col}}, w_{\text{reg}}$ are scalar weights that balance tracking fidelity, encirclement tightness, obstacle avoidance, collision avoidance, and formation regularity, respectively. In practice, the signs and magnitudes of these weights are determined empirically, enabling $J_{\text{fit}}, J_{\text{obs}}$, and J_{col} to serve as safety-related penalties, while J_{dist} and J_{reg} function as soft rewards for tight and regular capture formation.

For each neighboring candidate $s \in Q_{\text{adj}}$, the optimal formation parameters (Θ^*, R^*) are obtained by minimizing the dense-environment capture cost above:

$$(\Theta^*, R^*) = \arg \min_{\Theta, R} \tilde{J}(S, \Theta, R) \quad (26)$$

Among all candidate sequences s , the one achieving the minimum cost

$$\tilde{J}^*(S) = \min_{\Theta, R} \tilde{J}(S, \Theta, R) \quad (27)$$

is selected as s^* , and the corresponding optimal capture points are

$$Q_j^* = P_j^*(\Theta^*, R^*) = P + R^* \begin{bmatrix} \cos \theta_j^* \\ \sin \theta_j^* \end{bmatrix}, \quad j = 1, \dots, N \quad (28)$$

This enables the drone swarm to adapt its formation to local obstacles and inter-aircraft spacing constraints while capturing targets densely and uniformly throughout the environment. The cost functions are solved using the Ceres solver library.

3.3. Swarm trajectory planning

The trajectory planning module converts the optimized capture formation into executable UAV commands. The module generates polynomial trajectories and performs short-term inter-UAV collision checking through an event-triggered mechanism. Upon conflict detection, a target-distance-based priority rule selects which drones replan, mitigating hovering deadlock. Yaw planning is jointly optimized with position trajectories, incorporating outward alignment, target visibility, smoothness, and rate limits to ensure visual contact while enabling environmental observation.

3.3.1. Tracking trajectory planning

In the safe pursuit trajectory planning framework, the tracking trajectory generation module adopts the FAST-tracker framework proposed by [3]. Because planning differs between multi-machine and single-machine systems, the anti-collision mechanism among clusters must be considered. In multi-UAV cluster cooperative operations, preventing flight-path conflicts is pivotal to ensuring safe flight operations. In this paper, we propose an anti-collision mechanism based on priority ranking and short-term collision prediction that reduces collision risk by coordinating the movement priorities of individual UAVs within the swarm. It does so by coordinating the movement priority of each UAV in the cluster. The fundamental principles of the mechanism are outlined as follows.

The objective of the short-term collision risk detection machine is to detect collisions during agile capture. This machine has been developed to reduce computation time and improve real-time performance. It carries out collision detection for the next short moment (or a short time window) in the UAV prediction path. The specific method is as follows: for the native trajectory to be executed, a sequence of discrete sampling points is generated based on a hybrid A* search and trajectory optimization. This sequence is used as the predicted path. The trajectory of UAV u_i is defined by m-segment polynomials, thereby establishing the set of parameters:

$$\Gamma_i = \{t_k, n_k, C_k\}_{k=1}^m \quad (29)$$

Here, t_k denote the duration of the k^{th} segment, while n_k denotes the polynomial order of the k^{th} segment. The coefficient matrix, $C_k = [c_x^{(k)}, c_y^{(k)}, c_z^{(k)}] \in \mathbb{R}^{(n_k+1) \times 3}$.

In accordance with the per-segment polynomial that was obtained from the preceding tracking trajectory planning, the relative time of the present moment within each segment is calculated.

$$\begin{cases} t_{rel} = \max(t_{abs} - t_{start}, 0) \\ S = \arg \min_k \left(\sum_{i=1}^k t_i \geq t_{rel} \right) \\ t_{seg} = t_{rel} - \sum_{i=1}^{S-1} t_i \end{cases} \quad (30)$$

where t_{abs} is the absolute time, t_{start} is the trajectory start timestamp, S is the number of trajectory segments, and t_{seg} is the relative time within a segment. Calculate the 3D spatial position of the predicted trajectory:

$$P(t) = \begin{cases} P_{odom}, & t_i \text{ null} \\ \sum_{j=0}^{n_s} C_j^{(s)} t_{seg}^j, & \text{others} \end{cases} \quad (31)$$

in this context, $P_{odom} = [x_{odom}, y_{odom}, z_{odom}]$ denotes the current position of the UAV, and $C_j^{(s)} = [c_{x,j}^{(s)}, c_{y,j}^{(s)}, c_{z,j}^{(s)}]$ is the coefficient of the j^{th} term of the s^{th} segment. The sampling points at each moment in the local trajectory are then compared with the extrapolated positions of all other UAVs at the same predicted moment. If the distance is below the preset safety distance, the system determines the potential collision risk.

The prioritization of tasks is contingent upon the target distance. Upon the initiation of collision detection, each UAV is equipped with the capability to receive the state information of the target in real time, concurrently acquiring position information of other UAVs through integrated sensors or communication systems. Each UAV calculates the Euclidean distance between itself and the target and compares it with the distances of other UAVs that may collide, as obtained from the collision detection mechanism described above, to prioritize movement. The drone closest to the target (i.e., with priority 0) will maintain its predetermined trajectory even if a potential collision risk is detected; while other drones will replan to varying degrees according to their priorities. After prioritization, the replanning time of obstacle avoidance drones increases sequentially and equals the smallest time unit multiplied by the priority number. The calculation method for priority is as follows:

$$\mathcal{P}(i) = \text{rank}(\{\|P_j - p_{\text{target}}\|_2\}_{j=1}^N) \quad (32)$$

in this context, $P_j \in \mathbb{R}^3$ represents the positional coordinates of the j^{th} UAV, p_{target} denotes the target positional coordinates, and the $\text{rank}(\cdot)$ function retrieves the indexed order subsequent to ascending sorting. This method prevents all UAVs from hovering simultaneously during mission execution, thereby ensuring the robustness of cluster-persistent action and target capture.

3.3.2. Yaw coordination and field-of-view-aware planning

The yaw motion of each pursuer UAV plays a critical role in target visibility, trajectory planning, and obstacle perception. Onboard cameras provide a limited horizontal FOV. To balance all these constraints within the limited FOV, this paper proposes a method that enhances tracking trajectories through yaw planning under field-of-view, smoothness, and yaw-rate constraints.

Before entering the capture phase, one UAV, denoted by U_i , is assigned to follow the real target and provide an accurate estimate of its motion. At the switching time t_0 from tracking to capture, we use the terminal state of U_i to initialize the yaw planning. Let $P(t_0)$ be the estimated target position and Q_j^* be the position of U_i . The initial yaw of the tracking UAV is chosen such that its camera FOV is centered on the target:

$$\psi_f(t_0) = \arg \max_{\psi} d(\psi)^{\top} \frac{P(t_0) - Q_j^*}{\|P(t_0) - Q_j^*\|_2} \quad (33)$$

where $d(\psi)$ denotes the unit viewing direction of the camera in the horizontal plane associated with yaw ψ . For the remaining UAVs, the initial yaw is set to align with their outward radial directions (defined below), so that they immediately contribute to outward perception around the swarm boundary. This initialization ensures a seamless transition from the tracking stage to the capture stage, while preserving at least one reliable line of sight to the real target.

For each pursuer UAV U_i , the trajectory planner described earlier provides a time-parameterized positional trajectory $p_i(t)$. At each discrete sampling instant along this trajectory, we define a radial “outward” unit vector relative to the group center of mass $C(t_k)$ as

$$n_i(t_k) = \frac{p_i(t_k) - C(t_k)}{\|p_i(t_k) - C(t_k)\|_2} \quad (34)$$

The yaw angle of UAV u_i at time t_k is denoted by $\psi_i(t_k)$, and its corresponding camera boresight direction in the horizontal plane is $d(\psi_i(t_k))$. To make each UAV’s FOV preferentially “look outward”, we introduce an alignment cost over the planning horizon $\mathcal{T} = \{t_1, \dots, t_K\}$:

$$J_{\text{yaw}} = \sum_{i=1}^N \sum_{t_k \in \mathcal{T}} \|d(\psi_i(t_k)) - n_i(t_k)\|_2^2 \quad (35)$$

To ensure that the swarm does not lose sight of the target. Let $P(t_k)$ be the estimated target position at time t_k , and define the line-of-sight unit vector from U_i to the target as

$$\ell_i(t_k) = \frac{P(t_k) - p_i(t_k)}{\|P(t_k) - p_i(t_k)\|_2} \quad (36)$$

The angular separation between the camera boresight and the line of sight is

$$\gamma_i(t_k) = \arccos(d(\psi_i(t_k))^T \ell_i(t_k)) \quad (37)$$

The target is inside the horizontal FOV if $\gamma_i(t_k) \leq \varphi_{\max}$, where $\varphi_{\max} = \frac{1}{2}\varphi_{\text{FOV}} - \Delta\varphi$ includes a safety margin $\Delta\varphi$ for modeling errors and occlusion.

To encode the requirement that “at least N_{FOV} drones must observe the target”, we introduce a soft visibility penalty mechanism. At each time step, we select the optimal observing drone and impose a penalty for violating the FOV constraint:

$$\eta(t_k) = \text{rank}^{N_{\text{FOV}}}(\gamma_i(t_k) - \varphi_{\max}) \quad (38)$$

Where, $\text{rank}^{N_{\text{FOV}}}$ denotes the N_{FOV} number in this ascending sequence.

$$J_{\text{vis}} = \sum_{t_k \in \mathcal{T}} \rho(\eta(t_k)), \rho(s) = [\max(0, s)]^2 \quad (39)$$

If at least N_{FOV} UAVs satisfy $\gamma_i(t_k) \leq \varphi_{\max}$, then $\eta(t_k) \leq 0$ and the corresponding term vanishes. Otherwise, J_{vis} grows quadratically with the smallest FOV violation, which drives the optimizer to rotate one or more UAVs toward the target until visibility is re-established.

To avoid abrupt and aggressive yaw motions, we further impose a smoothness term on the yaw trajectories. Let $\Delta t_k = t_k - t_{k-1}$ and define the discrete yaw increment of U_i as

$$\Delta\psi_i(t_k) = \psi_i(t_k) - \psi_i(t_{k-1}), \quad k = 2, \dots, K \quad (40)$$

The yaw smoothness cost over the planning horizon is given by

$$J_{\text{sm}} = \sum_{i=1}^N \sum_{k=2}^K (\Delta\psi_i(t_k))^2 \quad (41)$$

This term penalizes large step-to-step changes in yaw angle, leading to continuous and physically feasible yaw trajectories that are more compatible with the attitude dynamics and onboard gimbal constraints. In implementation, we also limit the yaw increment by the maximum yaw rate $\dot{\psi}_{\max}$:

$$|\Delta\psi_i(t_k)| \leq \dot{\psi}_{\max} \Delta t_k, \quad i = 1, \dots, N, \quad k = 2, \dots, K \quad (42)$$

This rate limit prevents the FOV from snapping instantaneously between two directions.

Collecting all yaw angles over the horizon into a single decision vector,

$$x_{\psi} = [\psi_1(t_1), \dots, \psi_1(t_K), \dots, \psi_N(t_1), \dots, \psi_N(t_K)]^T \quad (43)$$

The yaw planning problem is formulated as the following nonlinear least-squares problem:

$$\min_{x_{\psi}} J_{\text{yaw}} + \lambda_{\text{vis}} J_{\text{vis}} + \lambda_{\text{sm}} J_{\text{sm}} \quad (44)$$

Where λ_{vis} and λ_{sm} represent the weights of the visualization constraint and the smoothing term constraint, respectively. The optimization problem is solved using Ceres. In each iteration, residuals are linearized around the current yaw estimate. The resulting yaw trajectory is then passed to the lower-level controller, along with position and velocity references.

4. SIMULATION AND EXPERIMENTATION

4.1. Simulation experiment

A $70\text{ m} \times 70\text{ m} \times 5\text{ m}$ environment was randomly generated in ROS and contained 100 obstacles. The heights of the cubic obstacles range from 1.0 to 1.3 m, and the widths range from 0.5 to 3.0 m. The diameters of the circular obstacles range from 0.5 to 0.7 m, and the inclination angles with respect to the xy plane are ± 0.5 rad. Four interceptor drones were used to track an autonomous target drone, each with a physical size of $0.6\text{ m} \times 0.6\text{ m} \times 0.25\text{ m}$ and a maximum speed of 5 m/s. The PGO sliding window L is set to a size of 25, and the weights λ_1 , λ_2 and λ_3 are 0.5, 1.5, and 1, respectively; the queue L in the prediction module is set to a size of 50, and the time step t_{step} for trajectory prediction is 2 s; the value of w_{fit} , w_{dist} , w_{obs} , w_{col} , and w_{reg} in the capture module are 1, 2, 5, 5, and 2; The parameters λ_{vis} and λ_{sm} in the visual observation optimization problem are set to 10 and 1, respectively. In the simulation, topic sharing represented communication among the UAVs. Since target detection is not the focus of this work, target observability was modeled by checking the line of sight between each UAV's FOV and the target. If the line segment was blocked by an obstacle, the target was regarded as unobserved by that UAV. Otherwise, the relative target observation was generated from this line-of-sight relation, and zero-mean Gaussian white noise with a standard deviation of 0.1 m was added to each coordinate component to simulate visual detection errors. Based on these noisy observations, the proposed framework performed target-state fusion, trajectory prediction, formation optimization, and swarm trajectory planning. The results show that the UAV swarm can maintain target visibility, avoid obstacles, and safely complete cooperative capture in complex environments. The proposed planning system has a startup time of approximately 2.5 s, and the replanning frequency is set to 15 Hz to support real-time capture. The simulation results are shown in Figure 6.

The apparent FOV snapping in the simulation is an optimization-level response rather than a physically executed motion. The yaw-coordination module uses multiple soft constraints, among which the visibility penalty requiring at least N_{FOV} UAVs to observe the target is assigned a high weight. Therefore, in cluttered or occluded scenes, the optimizer may produce relatively abrupt yaw-reference changes to recover target visibility, indicating that the visibility constraint is actively involved. In real-world experiments, however, the planned yaw is used as an expectation for the UAV controller and is executed under yaw-rate limits and vehicle dynamics constraints. As a result, the physical yaw motion remains smooth, and the snapping artifact observed in simulation is largely attenuated. Although yaw-rate limitation inevitably reduces the instantaneous visibility performance of the swarm to some extent, this trade-off is worthwhile as it ensures flight safety and dynamic feasibility, which are critical for reliable deployment in real-world dense environments. Because a minimum number of observers is set, the visibility of a target is lost when a drone slowly veers, but this does not affect the overall system's visibility of the target, as demonstrated in the experiment.

4.2. Experiment benchmark comparison

Our experimental benchmark comparison uses a hybrid approach that combines physical and simulation experiments. Through hybrid experiments, we can accurately measure the metrics and advantages of each method, which are validated by the physical experiments described later.

Due to the complex dynamic characteristics and wide applicability of the 3D figure-eight trajectory, the datasets for PGO pose estimation and target prediction in this paper are derived from a self-simulated figure-eight trajectory with Gaussian white noise. Each run lasted 30-35 s with a sampling frequency of 20 Hz, yielding over 600 sampling points per experiment. To compare the advantages and disadvantages of Bézier curve prediction and velocity prediction, the time setting for adaptive Bézier curve prediction and fixed-time velocity prediction was the same, both being 1 s. To ensure statistical reliability, all simulation experiments were independently repeated 10 times under identical initial conditions. The state estimation and prediction errors were recorded separately for each run. Since each experiment focuses on the performance evaluation of a single module (e.g., estimation or prediction), and the data sources are independent of other modules, the results from each run

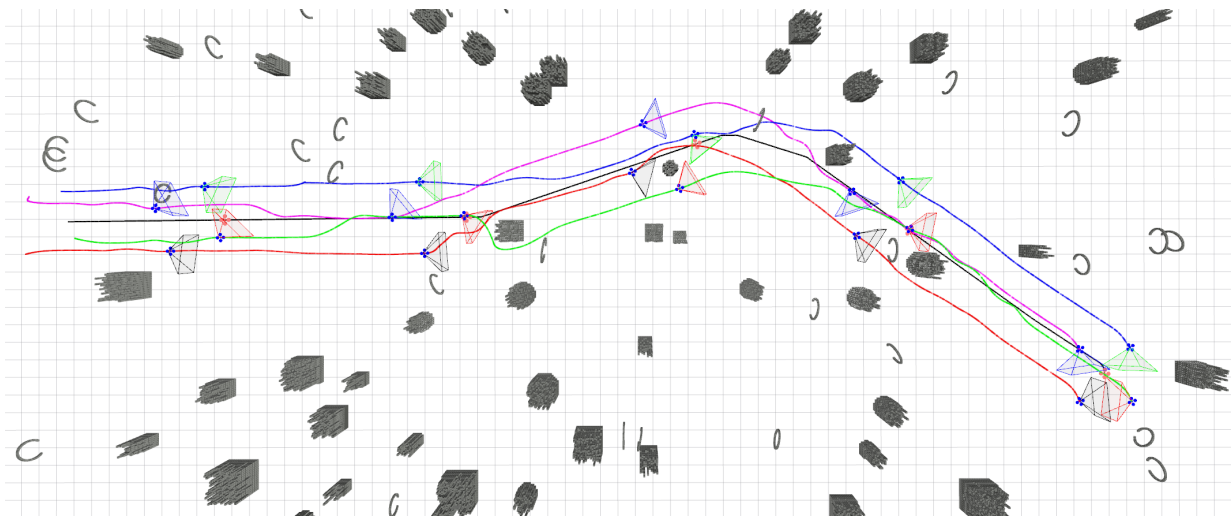


Figure 6. Simulation experiment. The four blue drones represent capture drones, while the red drone represents the target. Different colored curves indicate the movement trajectories of different drones, and different colored conical frames represent their fields of view. The corresponding video is provided as [Supplementary Video 1](#).

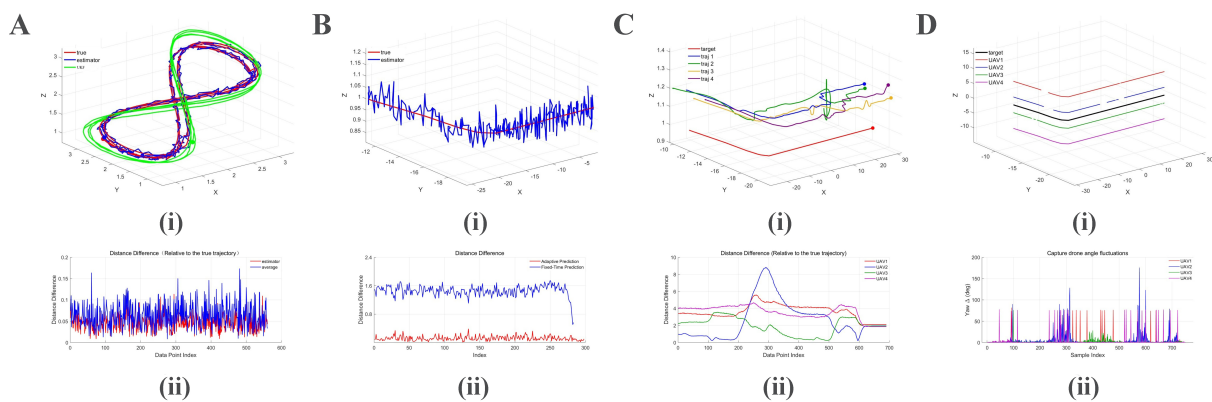


Figure 7. Metric validation. (A) show target position estimation results: [A(i)] compares the actual target trajectory, the proposed estimated trajectory, and the UKF estimated trajectory. [A(ii)] compares the estimation error of the proposed method and the average method; (B) show target state prediction results: [B(i)] compares the actual trajectory with the trajectory obtained by the adaptive prediction method. [B(ii)] compares the differences between adaptive prediction, current velocity fixed time difference prediction, and the actual trajectory; (C) show the trajectory status during the capture process: [C(i)] presents the historical trajectories of the drones, and [C(ii)] shows the distance between each drone and the target over time; (D) Visibility and yaw angle results during the capture process. [D(i)] Visibility status of each UAV with respect to the target: solid segments indicate that the target is visible, while blank gaps indicate that the target is not visible. [D(ii)] Yaw angle trajectories of all capturing UAVs over time. UKF: Unscented Kalman filter; UAV: unmanned aerial vehicle.

exhibit high reliability and consistency. For visualization, [Figure 7A\(i\)](#) and [B\(i\)](#) present the results from a single representative experiment, while [Figure 7A\(ii\)](#) and [B\(ii\)](#) show the pointwise averaged error curves across all 10 runs to evaluate the overall estimation and prediction performance. The trajectory and visual validation data during the encircling process are derived from the simulation data shown in [Figure 6](#).

For target-position state estimation, based on the three methods' trajectory smoothing and accuracy, we designed two experimental comparison charts; we compared performance metrics through the simulation experiment shown in [Figure 7A](#). Since the source code from paper [\[26\]](#) is not open-source, this comparison directly references data published in their paper. [Table 1](#) shows lower mean and maximum errors for the proposed

Table 1. Comparison of target state estimation errors

Method	Min (m)	Mean (m)	Max (m)
Proposed	0.0200	0.0459	0.1211
Average fusion	0.0280	0.0638	0.1736
UKF	0.1635	0.4276	0.6800
Ref. [26]	0.3300	0.5725	0.7400

UKF: Unscented Kalman filter.

Table 2. Comparison of target state prediction errors

Method	Min (m)	Mean (m)	Max (m)
Proposed	0.0412	0.1346	0.3503
Fixed-time prediction (1 s)	0.9878	1.4459	1.7493
Ref. [3]	1.8200	2.5400	3.4500
Ref. [29]	3.7000	4.6600	5.6000

Table 3. Comparison of visibility percentages

Method	Visibility (%)			
	3-vis	2-vis	1-vis	0-vis
Proposed	87.84	9.92	2.24	0.00
Ref. [25]	57.60	32.20	10.20	0.00
Ref. [4]	10.80	8.20	67.20	13.80

method, with the maximum estimated error remaining around 12 cm. Figure 7A further shows that although the graph optimization result is less smooth than the Kalman filter output, it tracks abrupt target direction changes more closely. The optimization method also reduces reliance on continuous observations from any single UAV, enabling effective application in cluttered environments.

In the context of target motion prediction, the system organizes all predicted points into a set. The adaptive time prediction method proposed in this paper is compared with a fixed-time velocity prediction in terms of real trajectory deviation, as shown in Figure 7B. The figure indicates that Bézier curves can provide accurate short-term prediction. A comparison is made with existing work^[3] and work^[29] on target motion prediction, as shown in Table 2 below. The present study is based on the methodology outlined in^[3]. The experimental conditions employed herein are consistent with those described in the aforementioned publication. Due to the non-open-source nature of the methodology presented in work^[29], direct referencing is employed to ensure congruence between the two methods. The table lists the minimum, average, and maximum prediction errors, highlighting the significant advantages of the proposed method.

During the initial system pursuit phase, all drones maintained stable pursuit trajectories and yaw angles. When

encountering obstacles, trajectory planning made compromises to avoid them, causing corresponding changes in visual yaw. However, the system rapidly responded to the current situation, replanning optimal pursuit strategies and yaw angles. The post-replanning state showed minimal deviation, thanks to the aforementioned $J_{fit}(S, \Theta, R)$ and J_{sm} , which ensured system stability. Figure 7C also demonstrates the event-triggered collision avoidance performance of the drones. After obstacle avoidance, the system swiftly returns to its previous state.

As shown in Table 3, the percentage of time the target remains visible to a specific number of drones during the total tracking duration. In this context, when i is less than N_{FOV} , “i-vis” refers to the case where exactly the specified number of UAVs have visibility of the target; when i is equal to N_{FOV} , “i-vis” refers to the case where at least the specified number of UAVs have visibility of the target. Compared to work^[25] and work^[4], our method offers higher target-visibility ratio and better sustained target observation performance for sustained target observation in dense environments. Furthermore, Figure 7D indicates that, apart from temporary large yaw deviations during obstacle avoidance, the yaw angles of all capturing drones are maintained within an 8-degree bound, demonstrating overall system stability.

4.3. Real world experiment

A series of empirical experiments was conducted on a self-built quadrotor platform in indoor and outdoor dense environments. The experiments were designed to validate the proposed estimation, prediction, and capture pipeline, not the target-recognition or SLAM. It should be noted that although the target shares its odometry with the interceptor UAVs, this setup does not imply a fully cooperative scenario. Instead, artificial noise is added to the shared states and line-of-sight occlusion checking is applied to determine observation availability, thereby simulating the intermittent and noisy visual observations that would be obtained from a non-cooperative target in real-world dense environments.

In the indoor experiment, all UAV and target positions were provided by the motion capture system with millimeter-level accuracy, and the size of each UAV and the target was $0.25 \text{ m} \times 0.25 \text{ m} \times 0.2 \text{ m}$. This experiment mainly validates target trajectory prediction and cooperative capture. The 3D perception data used for local planning were generated by the onboard camera-based visual SLAM/perception module. The average target speed was 0.6 m/s , the average UAV capture speed was 0.75 m/s , and the capture radius was set to 2-3 m. As shown in Figure 8A and B, the UAV swarm safely approached the target in a dense environment while maintaining smooth and dynamically feasible trajectories.

The outdoor experiment was conducted in a dense forest environment [Figure 8C and D]. It validates target-state fusion estimation, target trajectory prediction, and cooperative capture under onboard localization. The interceptor UAV size was $0.45 \text{ m} \times 0.45 \text{ m} \times 0.3 \text{ m}$, and the target size was $0.25 \text{ m} \times 0.25 \text{ m} \times 0.2 \text{ m}$. The UAV poses and local maps were obtained from each UAV’s LiDAR SLAM module, and the globally consistent pose fusion among UAVs was achieved via the back-end optimization method from reference^[2], with centimeter-level accuracy. At system initialization, the relative pose transformation between the target and each interceptor UAV was preset, so that each UAV had an initial estimate of the target position. With the target moving at an average speed of 0.7 m/s , the swarm achieved an average capture speed of 1 m/s . The tracking trajectories remained stable throughout the experiment. Detailed procedures are documented in the video recording.

5. CONCLUSIONS

This paper proposes an agile target acquisition drone swarm coordination system suitable for dense environments. Compared to existing approaches, this method shows superior performance in multi-UAV cooperative tracking, target visibility maintenance, and rapid capture of agile targets. To validate its feasibility and effectiveness, we conducted four-UAV flight experiments in indoor and outdoor environments. Target state estimation

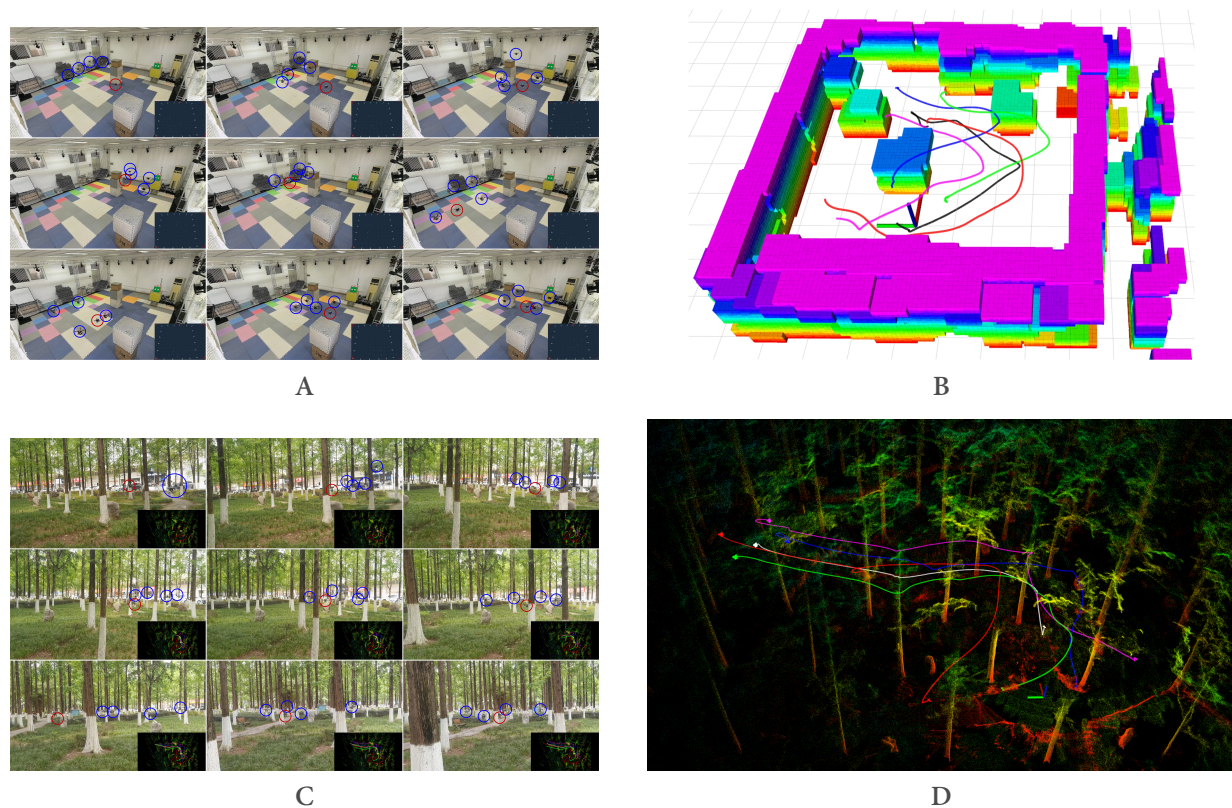


Figure 8. Real-world experiments. (A) Snapshot of the indoor experiment conducted in the motion capture laboratory, where four hunter drones track the target drone. The blue circles represent the hunter drones, and the red circle represents the target drone; (B) Real-time three-dimensional perception and trajectory planning process during the indoor capture experiment. The video of the indoor experiment is provided as [Supplementary Video 2](#); (C) Snapshot of the outdoor experiment in an unknown and cluttered environment; (D) Visualization of trajectories during the outdoor tracking experiment. The video of the outdoor experiment is provided as [Supplementary Video 3](#). Image source: physical experiments and hand-drawn Visio images.

and prediction achieved decimeter-level accuracy. Furthermore, the proposed solution maintained target visibility throughout the UAV swarm's real-time obstacle-avoidance planning process. However, our system has not been tested for real-world visual target detection, and the method assumes that UAV localization, mapping, communication, and target location information are reliable. To address these limitations, future work is needed.

1. Future work will integrate real-time visual detection algorithms (e.g., YOLO-based object detectors) to replace the current simulated observations, enabling fully autonomous vision-driven target acquisition and tracking.
2. In target motion prediction, the methods employed are based solely on historical three-dimensional coordinate trajectories, relying exclusively on kinematic reasoning. In the future, we will consider combining historical drone attitude data with dynamics for trajectory prediction.

DECLARATIONS

Authors' contributions

Conception and design of the study, manuscript writing, simulation and physical experiments, data analysis and interpretation: Zhang, P.; Huo, J.

Language polishing, structural revision of the manuscript, and data collection for the experiments: Yan, X.; Liu, R.

Administrative, technical, and material support: Liu, H.; Furletov, Y.

Availability of data and materials

All data presented in this study are derived from simulation experiments and physical experiments. The source code and datasets associated with the current study are available in the GitHub repository: <https://github.com/SWUST-ICAA/Fast-Capture>.

AI and AI-assisted tools statement

Not applicable.

Financial support and sponsorship

This work was supported in part by the Sichuan Provincial Science and Technology Program (2025YFRG0008); in part by the Postgraduate Innovation Fund Project by Southwest University of Science and Technology (25ycx1045).

Conflicts of interest

Huo, J. is a Junior Associate Chief Editor of the journal *Intelligence & Robotics*. Huo, J. was not involved in any steps of editorial processing, notably including reviewer selection, manuscript handling, and decision making, while the other authors have declared that they have no conflicts of interest.

Ethical approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Copyright

© The Author(s) 2026.

Supplementary Materials

[Supplementary Materials](#)

REFERENCES

1. Zhou, X.; Wang, Z.; Ye, H.; Xu, C.; Gao, F. EGO-Planner: an ESDF-free gradient-based local planner for quadrotors. *IEEE Robot. Autom. Lett.* **2021**, *6*, 478–85. [DOI](#)
2. Xu, H.; Zhang, Y.; Zhou, B.; et al. Omni-swarm: a decentralized omnidirectional visual-inertial-UWB state estimation system for aerial swarms. *IEEE Trans. Robot.* **2022**, *38*, 3374–94. [DOI](#)
3. Han, Z.; Zhang, R.; Pan, N.; Xu, C.; Gao, F. Fast-Tracker: a robust aerial system for tracking agile target in cluttered environments. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, Xi'an, China. May 30 - Jun 05, 2021. IEEE; 2021. pp. 328–34. [DOI](#)
4. Zhou, X.; Wen, X.; Wang, Z.; et al. Swarm of micro flying robots in the wild. *Sci. Robot.* **2022**, *7*, eabm5954. [DOI](#)
5. Qin, T.; Cao, S.; Pan, J.; Shen, S. A general optimization-based framework for global pose estimation with multiple sensors. *arXiv* **2019**, arXiv:1901.03642. Available online: <https://doi.org/10.48550/arXiv.1901.03642>. (accessed 2026-09-04)
6. Xu, H.; Liu, P.; Chen, X.; Shen, S. D²SLAM: decentralized and distributed collaborative visual-inertial SLAM system for aerial swarm. *IEEE Trans. Robot.* **2024**, *40*, 3445–64. [DOI](#)
7. Fauser, J.; Chadda, R.; Goergen, Y.; et al. Planning for flexible surgical robots via Bézier spline translation. *IEEE Robot. Autom. Lett.* **2019**, *4*, 3270–7. [DOI](#)
8. Quan, L.; Yin, L.; Zhang, T.; et al. Robust and efficient trajectory planning for formation flight in dense environments. *IEEE Trans. Robot.* **2023**, *39*, 4785–804. [DOI](#)
9. Zhou, X.; Zhu, J.; Zhou, H.; Xu, C.; Gao, F. EGO-Swarm: a fully autonomous and decentralized quadrotor swarm system in cluttered environments. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, Xi'an, China. May 30 - Jun 05, 2021. IEEE; 2021. pp. 4101–7. [DOI](#)
10. López-Nicolás, G.; Aranda, M.; Mezouar, Y. Formation of differential-drive vehicles with field-of-view constraints for enclosing a moving target. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, Singapore. May 29 - Jun 03, 2017. IEEE; 2017. pp. 261–6. [DOI](#)

11. Li, C.; Wang, Z.; Song, W.; Zhao, S.; Wang, J.; Shan, J. Resilient unscented Kalman filtering fusion with dynamic event-triggered scheme: applications to multiple unmanned aerial vehicles. *IEEE Trans. Control Syst. Technol.* **2023**, *31*, 370–81. [DOI](#)
12. Zhou, W.; Hou, J. A new adaptive robust unscented Kalman filter for improving the accuracy of target tracking. *IEEE Access* **2019**, *7*, 77476–89. [DOI](#)
13. Dong, Y.; He, C.; Wang, Z. J. Dynamic object tracking by multi-UAV with time-variant radio maps. *IEEE Trans. Wireless Commun.* **2024**, *23*, 7471–87. [DOI](#)
14. He, C.; Dong, Y.; Wang, Z. J. Radio map assisted multi-UAV target searching. *IEEE Trans. Wireless Commun.* **2023**, *22*, 4698–711. [DOI](#)
15. Wang, C.; Meng, L.; Gao, Q.; Wang, T.; Wang, J.; Wang, L. A target sensing and visual tracking method for countering unmanned aerial vehicle swarm. *IEEE Sens. J.* **2024**, *24*, 30340–51. [DOI](#)
16. Li, J. M.; Chen, C. W.; Cheng, T. H. Motion prediction and robust tracking of a dynamic and temporarily-occluded target by an unmanned aerial vehicle. *IEEE Trans. Control Syst. Technol.* **2021**, *29*, 1623–35. [DOI](#)
17. Doostmohammadian, M.; Taghieh, A.; Zarrabi, H. Distributed estimation approach for tracking a mobile target via formation of UAVs. *IEEE Trans. Autom. Sci. Eng.* **2022**, *19*, 3765–76. [DOI](#)
18. Franchi, A.; Oriolo, G.; Stegagno, P. Mutual localization in a multi-robot system with anonymous relative position measures. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, St. Louis, USA, Oct 10–15, 2009. IEEE; 2009. pp. 3974–80. [DOI](#)
19. Dellaert, F.; Kaess, M. Factor graphs for robot perception. *Found. Trends Robot.* **2017**, *6*, 1–139. [DOI](#)
20. Michael, N.; Shen, S.; Mohta, K.; et al. Collaborative mapping of an earthquake damaged building via ground and aerial robots. In *Field and service robotics*. Berlin, Germany: Springer; 2014. pp. 33–47. [DOI](#)
21. Cunningham, A.; Paluri, M.; Dellaert, F. DDF-SAM: fully distributed SLAM using constrained factor graphs. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Taipei, Taiwan, Oct 18–22, 2010. IEEE; 2010. pp. 3025–30. [DOI](#)
22. Cunningham, A.; Indelman, V.; Dellaert, F. DDF-SAM 2.0: consistent distributed smoothing and mapping. In *2013 IEEE International Conference on Robotics and Automation*, Karlsruhe, Germany, May 06–10, 2013. IEEE; 2013. pp. 5220–7. [DOI](#)
23. Wu, J.; Zhang, N.; Li, D.; Bi, J.; Han G. A context-aware feature fusion method for multi-UAV cooperative air combat. *IEEE Trans. Intell. Transp. Syst.* **2025**, *26*, 7197–210. [DOI](#)
24. Peng, Z.; Wu, G.; Luo, B.; Wang, L. Multi-UAV cooperative pursuit strategy with limited visual field in urban airspace: a multi-agent reinforcement learning approach. *IEEE/CAA J. Autom. Sin.* **2025**, *12*, 1350–67. [DOI](#)
25. Rao, K.; Yan, H.; Yang, P.; Wang, M.; Lv, Y. Multi-UAV trajectory planning with field-of-view sharing mechanism in cluttered environments: application to target tracking. *Sci. China Inf. Sci.* **2025**, *68*, 150206. [DOI](#)
26. Tang, Z.; Wang, Y.; Chen, Q.; Yang, X. Research on target state estimation and terminal guidance algorithm in the process of multi-UAV cooperative attack. In *2020 5th International Conference on Automation, Control and Robotics Engineering (CACRE)*, Dalian, China, Sep 19–20, 2020. IEEE; 2020. pp. 165–71. [DOI](#)
27. Khosravi, M.; Arora, R.; Enayati, S.; Pishro-Nik, H. A search and detection autonomous drone system: from design to implementation. *IEEE Trans. Autom. Sci. Eng.* **2025**, *22*, 3485–501. [DOI](#)
28. Mason, F.; Capuzzo, M.; Magrin, D.; Chiariotti, F.; Zanella, A.; Zorzi, M. Remote tracking of UAV swarms via 3D mobility models and LoRaWAN communications. *IEEE Trans. Wireless Commun.* **2022**, *21*, 2953–68. [DOI](#)
29. Chen, J.; Liu, T.; Shen, S. Tracking a moving target in cluttered environments using a quadrotor. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Daejeon, Korea, Oct 09–14, 2016. IEEE; 2016. pp. 446–53. [DOI](#)
30. Chen, L.; Xia, Y. Cooperative capture strategy for a group of slower UAVs to intercept one faster intruder in 3-D space. *IEEE Trans. Aerosp. Electron. Syst.* **2025**, *61*, 8757–69. [DOI](#)
31. Huang, S.; Zhang, H.; Huang, Z. E^2CoPre : energy efficient and cooperative collision avoidance for UAV swarms with trajectory prediction. *IEEE Trans. Intell. Transp. Syst.* **2024**, *25*, 6951–63. [DOI](#)
32. Jeon, B.; Lee, Y.; Kim, H. J. Integrated motion planner for real-time aerial videography with a drone in a dense environment. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, Paris, France, May 31 – Aug 31, 2020. IEEE; 2020. pp. 1243–9. [DOI](#)
33. Ji, J.; Pan, N.; Xu, C.; Gao, F. Elastic Tracker: a spatio-temporal trajectory planner for flexible aerial tracking. In *2022 International Conference on Robotics and Automation (ICRA)*, Philadelphia, USA, May 23–27, 2022. IEEE; 2022. pp. 47–53. [DOI](#)
34. Zhao, S.; Zheng, J.; Liu, K.; Liu, J.; Wang, X. Cooperative moving target fencing control for two-layer UAVs with relative measurements. *IEEE Trans. Autom. Sci. Eng.* **2025**, *22*, 7145–58. [DOI](#)
35. Wang, H.; Zhang, X.; Liu, Y.; Sun, G.; Zhang, X.; Zhuang, Y. PCDCT: perception-complementarity-driven collaborative trajectory generation for vision-based aerial tracking. *IEEE Trans. Autom. Sci. Eng.* **2025**, *22*, 10520–32. [DOI](#)
36. Chen, J.; Wang, Y.; Zhang, Y.; Lu, Y.; Shu, Q.; Hu, Y. Extrinsic-and-intrinsic reward-based multi-agent reinforcement learning for multi-UAV cooperative target encirclement. *IEEE Trans. Intell. Trans. Syst.* **2025**, *26*, 17653–65. [DOI](#)
37. Chen, J.; Yu, C.; Li, G.; et al. Online planning for multi-UAV pursuit-evasion in unknown environments using deep reinforcement learning. *IEEE Robot. Autom. Lett.* **2025**, *10*, 8196–203. [DOI](#)
38. Song, C.; Zhang, X.; She, Y.; Li, B.; Zhang, Q. Trajectory planning for UAV swarm tracking moving target based on an improved model predictive control fusion algorithm. *IEEE Internet Things J.* **2025**, *12*, 19354–69. [DOI](#)
39. Cheng, J.; Li, N.; Wang, B.; Bu, S.; Zhou, M. High-sample-efficient multiagent reinforcement learning for navigation and collision avoidance of UAV swarms in multitask environments. *IEEE Internet Things J.* **2024**, *11*, 36420–37. [DOI](#)
40. Oğuz, S.; Heinrich, M. K.; Allwright, M.; et al. An open-source UAV platform for swarm robotics research: using cooperative sensor

- fusion for inter-robot tracking. *IEEE Access* **2024**, *12*, 43378–95. [DOI](#)
41. Xiong, H.; Shi, X.; Liu, J.; Chen, Y.; Wang, J. A swarm model with constraint coordination mechanism for unmanned aerial vehicle swarm formation maintenance in dense environments. *Ind. Robot.* **2025**, *52*, 312–22. [DOI](#)
42. Wang, G.; Xu, Y.; Liu, Z.; Xu, X.; Wang, X.; Yan, J. Integrating human experience in deep reinforcement learning for multi-UAV collision detection and avoidance. *Ind. Robot.* **2022**, *49*, 256–70. [DOI](#)

Disclaimer/Publisher's Note: All statements, opinions, and data contained in this publication are solely those of the individual author(s) and contributor(s) and do not necessarily reflect those of OAE and/or the editor(s). OAE and/or the editor(s) disclaim any responsibility for harm to persons or property resulting from the use of any ideas, methods, instructions, or products mentioned in the content.



© The Author(s) 2026. Open Access This article is licensed under a Creative Commons Attribution 4.0 International License (<https://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, sharing, adaptation, distribution and reproduction in any medium or format, for any purpose, even commercially, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.