IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

ROAR: A Robust Autonomous Aerial Tracking System for Challenging Scenarios

Tong Zhang[1], Chenghao Li[1], Kezhen Zhao[1], Hao Shen[1], Tao Pang[2]

***Abstract*—Autonomous tracking represents a significant advancement in the evolution of unmanned aerial vehicles (UAVs), offering applications in areas such as aerial photography and infrastructure inspection. Despite its potential, many autonomous tracking systems encounter challenges in maintaining consistent and reliable target tracking, particularly in complex and dynamic environments. This study presents a robust approach to autonomous tracking aimed at overcoming two primary challenges that often lead to tracking failures: target loss and poor tracking trajectory quality. To tackle these issues, a Markov chain-based motion prediction method is introduced to estimate the target's future position probabilities over time. Based on these predictions, a re-capture strategy is designed to enhance decision-making, ensuring effective recovery when the target exits the field of view (FOV). Additionally, the tracking process is modified by formulating cost functions that incorporate tracking distance and optimal yaw angle, while ensuring safety, dynamic feasibility, and operational constraints. Simulations and real-world experiments validate the proposed method, demonstrating stable and efficient tracking performance in challenging environments.**

***Index Terms*—Aerial Systems: Applications, Motion and Path Planning, Task and Motion Planning.**

I. Introduction

In recent years, advancements in autonomous navigation technology for robots have significantly contributed to the application of UAV-based target tracking in various fields, including aerial photography and surveillance [1]–[4].

In challenging scenarios where the target exhibits high uncertainty and maneuverability, becomes obscured by a complex environment, or falls outside the UAV sensor's limited FOV, it may frequently disappear from the sensor's detection range. This phenomenon, known as target loss, disrupts tracking stability and can ultimately lead to tracking failure, highlighting the need for robust tracking algorithms.

To address target loss, most existing methods focus on generating tracking trajectories by incorporating carefully designed constraints, such as visibility and distance constraints, to minimize the duration for which the target remains outside the FOV and thereby enhance tracking stability. However, some methods employ complex cost functions [5], [6], while others overlook yaw angle optimization [7]. This increases the optimization burden on the position trajectory, making it more susceptible to local minima and ultimately degrading trajectory quality. Furthermore, in dynamic and cluttered environments, target loss is inevitable, and trajectory optimization alone cannot ensure reliable tracking. Existing approaches lack effective target re-capture mechanisms, as current relocation strategies primarily plan paths to the target's last known position [8], which becomes inefficient in complex scenarios.

To address the susceptibility of tracking algorithms to target loss, this paper introduces **R**obust Aut**o**nomous **A**erial T**r**acking (**ROAR**) System. This system enhances trajectory quality while enabling efficient re-capture of lost targets. The approach begins by generating a series of nodes with probabilistic target appearance information, leveraging motion primitives expansion and Markov chains. Instead of directly embedding visibility constraints into the trajectory optimization process, the method generates a sequence of viewpoints to improve target observability, thereby reducing the burden of trajectory optimization. When target loss occurs, a re-capture strategy is employed to relocate the target using these predicted viewpoints. On the backend, a B-spline joint trajectory optimizer is implemented to refine trajectory quality, incorporating a time-dependent tracking distance penalty term. This ensures the generation of high-quality trajectories with a lightweight cost function, maintaining an optimal observation position for the target. Additionally, the yaw angle is treated as an independent optimization variable, ensuring that the target remains within the UAV's FOV while relaxing strict position trajectory constraints.

In comparison to other state-of-the-art approaches, the proposed method achieves higher tracking success rates and extends target retention within the UAV's FOV under challenging conditions, demonstrating excellent robustness. Additionally, real-world experiments conducted in outdoor environments further validated the effectiveness of the proposed method. The key contributions of this study are as follows:

1) A motion prediction framework based on a Markov chain is introduced, incorporating viewpoint generation and a re-capture strategy derived from prediction outcomes. This ensures sustained target visibility and facilitates efficient re-capture following target loss.
2) A novel cost function is designed, integrating both tracking distance and optimal yaw angle. This cost function is integrated into a joint trajectory optimizer, which incorporates additional factors to effectively reduce the risk of target loss from the FOV.
3) Extensive evaluations in both simulated and real-world environments confirm that the proposed method consis-

This work is supported by National Natural Science Foundation of China (NSFC) (61603297) and Natural Science Foundation of Shaanxi Province (2023JC-YB-503) Foundation.

[1]Tong Zhang, Chenghao Li, Kezhen Zhao, Hao Shen are with Unmanned System Research Institute, Northwestern Polytechnical University, Xi'an 710072, China zhangtong@nwpu.edu.cn, chenghao620@foxmail.com , zhenke@mail.nwpu.edu.cn, 2022204583@mail.nwpu.edu.cn

[2]Tao Pang is with the The 32nd Research Institute of China Electronics Technology Group, Shanghai 201808, China t_pang@126.com

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

tently achieves stable and reliable tracking performance in challenging scenarios.

II. Related Works

Previous studies have approached the target tracking challenge from a control-based perspective. For instance, Dmytruk et al. [9] formulated UAV target tracking as a nonlinear model predictive control (NMPC) problem, integrating visual perception constraints to maintain continuous target visibility while fulfilling mission objectives. Similarly, Nageli et al. [10] developed an model predictive control (MPC) solver that optimizes the trajectories of both the UAV and its gimbal while incorporating obstacle avoidance strategies. However, these approaches rely on simplified assumptions that represent obstacles as spheres or ellipsoids, which restricts their applicability in complex environments. Several methods have been developed within the frameworks of image-based visual servoing (IBVS) [3], [11] and position-based visual servoing (PBVS) [12], utilizing controllers that incorporate the full dynamic model of the UAV. While these approaches perform well in open environments, their lack of dedicated obstacle avoidance mechanisms limits their effectiveness in cluttered settings. Control-based methods share a common drawback: they generate control commands based solely on the current frame, lacking a long-term perspective on tracking task, which increases the risk of target loss from the FOV.

Several studies have explored target tracking through trajectory generation to address the limitations of control-based methods. Han et al. [8] and Chen et al. [13] utilized polynomial fitting to predict target motion, optimizing the fitted polynomial for trajectory planning. It should be indicated that Chen et al. [13] leveraged UAV-predicted trajectories as initial values for solving quadratic programming (QP) problems to generate safe polynomial trajectories, while Han et al. [8] employed path search results as initial inputs for spatiotemporal joint optimization to refine the final trajectory. However, neither approach incorporated tracking-related constraints, such as visibility and tracking distance, making it challenging to track targets with unpredictable movements. Joen et al. [2], [14], [15] introduced a series of aerial tracking strategies. [2], [14] maintained visibility by defining the shortest distance between the UAV-target line and nearby obstacles, with safe trajectories generated via viewpoint skeletons using graph search methods. However, these approaches rely on a known global map, limiting their effectiveness in dynamic or unknown environments. [15] derived target trajectory predictions from past position observations and sampling-based inspection strategies. Safe and dynamically feasible trajectories were then generated using a viewpoint-based approach. Similarly, Ji et al. [7] applied an extended kalman filter (EKF) to estimate target positions and integrated these predictions into a spatiotemporal joint optimizer that considered tracking distance and visibility. A major drawback of [7], [15] is their lack of yaw angle optimization, which increases the probability of target loss when tracking highly maneuverable objects. Wang et al. [5] and Gao et al. [16] employed target prediction strategies similar to [8], using dynamic A* for path planning and incorporating a joint optimizer that accounts for tracking distance and target visibility, while also optimizing the yaw angle trajectory. Moreover, Wang et al. [6] followed a similar optimization framework but emphasized environmental perception to enhance tracking safety. However, in complex environments, incorporating excessive optimization constraints can lead to convergence failures, as evidenced by an increased frequency of planning failures in simulations when the target speed rises. A shared limitation among these trajectory generation methods is their inability to adapt when the target moves out of the UAV's FOV or remains occluded by obstacles for an extended duration. These approaches lack a robust decision-making mechanism to effectively re-capture lost targets.

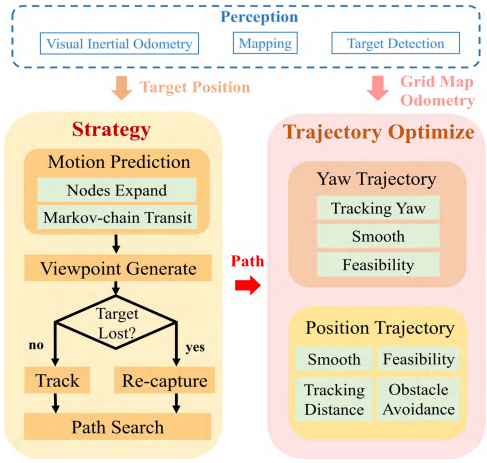

Fig. 1. An overview of the proposed autonomous tracking framework.

To reduce tracking failures, the proposed method concentrates on two main elements: target re-capture and trajectory optimization. The main objective of this approach is to reduce instances of the target escaping from the UAV's FOV and enhance the overall quality of the tracking trajectory, thereby improving both tracking success and system robustness.

III. System Overview

The proposed UAV tracking framework, as depicted in Fig.1, consists of two main components: strategy and trajectory planning. At the start of the tracking task, motion nodes are expanded based on the target's movement information from the previous period. Using a Markov chain, the probability of the target's future position at these nodes is calculated. Accordingly, a series of viewpoints are generated through a heuristic approach, leveraging the potential nodes. When the target is lost from the UAV's FOV, the re-capture strategy is activated, guiding the UAV to explore all potential nodes to re-capture the lost target.

In the trajectory planning phase, the kinodynamic A* algorithm is employed to generate initial path, and B-splines are used for trajectory parameterization. This phase considers dynamic constraints, tracking distance, and trajectory smoothness. The yaw angle is also optimized using the same approach, ensuring it stays aligned with the target while maintaining smooth motion and satisfying dynamic constraints.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

IV. Motion Predict and Track Strategies

In the motion prediction phase, a technique similar to the hybrid-state A* algorithm is employed to expand motion primitives. Additionally, the probability of the target appearing at each motion primitive in the future is determined using a Markov chain.

A. Nodes Expansion

The state vector $\mathbf{s}_t$ of the drone in world frame at time t are represented by Equ.1, including position vector $\boldsymbol{\varrho}_t$ and velocity vector $\boldsymbol{v}_t$:

$$\boldsymbol{s}_t = [\boldsymbol{\varrho_t}, \boldsymbol{v_t}]^T \tag{1}$$

The control input $u \in [-a_{max}, a_{max}]^3 \subset \mathbb{R}^3$ is represented as acceleration. It is discretized in each dimension as $u_d \in u_k = \{-a_{max}, -\frac{n-1}{n}a_{max}, \ldots, \frac{n-1}{n}a_{max}, a_{max}\}$. In these expressions, n indicates the degree of discretization; a smaller value of n leads to more discrete control inputs. With a time interval Δt, the state transition equation is expressed as Equ.2:

$$\boldsymbol{s}_{t+\Delta t} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \boldsymbol{s}_t + \frac{1}{2}\begin{bmatrix} \Delta t^2 \\ 2\Delta t \end{bmatrix} u_d \tag{2}$$

After a single expansion, $(2n+1)^3$ nodes are generated, each containing position, velocity, and acceleration data from the previous time step. To enhance the predictive information, these nodes are used as the starting point for another round of node expansion. After m expansions, a total of $(2n+1)^{3m}$ nodes are obtained.

B. Markov Chain-Based Motion Prediction

Inspired by the research conducted by Xu et al. [17], a motion prediction approach based on a Markov chain is proposed, where the discretized accelerations are treated as states within the Markov chain. The probability of the target transitioning to each state during every node expansion is then computed. To model this, it is assumed that the probability distribution follows a Gaussian probability density function:

$$f(a_x, a_y, a_z) = \frac{1}{(2\pi)^{3/2}\sigma_x\sigma_y\sigma_z} \cdot \exp \left(-\frac{1}{2}\left(\frac{(a_x - a_x^n)^2}{\sigma_x^2} + \frac{(a_y - a_y^n)^2}{\sigma_y^2} + \frac{(a_z - a_z^n)^2}{\sigma_z^2}\right)\right) \tag{3}$$

a_x, a_y, and a_z represent the state of the nodes after expansion, while a_x^n, a_y^n, and a_z^n denote the mean values of the probability distribution function, corresponding to the accelerations of the nodes before expansion. This study assumes that the motion is most likely to maintain a constant acceleration, keeping the mean acceleration equal to its pre-expansion value. The covariances of the probability distribution, represented by σ_x, σ_y, and σ_z, are set to 1.5 in the experiments. Using this probability density function, the probability vector can be defined as follows:

$$\boldsymbol{p}_i = [p_{i1}, p_{i2}, \cdots, p_{ij}]^T \tag{4}$$

In this context, p_{ij} represents the normalized probability of transitioning from acceleration $a_j \in u_k$ to acceleration $a_i \in u_k$. Each state $a_j \in u_k$ has a corresponding probability vector that describes the probability of transitioning to all other states within u_k. By integrating all the individual probability vectors, a probability transition matrix P can be constructed.

$$P = [\boldsymbol{p}_1, \boldsymbol{p}_2, \cdots, \boldsymbol{p}_i] \tag{5}$$

Considering the target's natural avoidance of obstacles, the probability vector $\boldsymbol{p}_f$ will be calculated to reduce the probabilities for nodes that are located near obstacles. When expanding a node, the probability vector $\boldsymbol{p}_i^t$ at time stamp t for transitions from the current node to all other nodes is calculated and can be expressed by Equ.6.

$$\boldsymbol{p}_i^t = P \cdot \boldsymbol{p}_i^{t-1} * \boldsymbol{p}_f \tag{6}$$

$$\boldsymbol{p}_f = \frac{[d_1, d_2, \cdots, d_i]^T}{\sqrt{{d_1}^2 + {d_2}^2 + \cdots + {d_i}^2}} \tag{7}$$

The symbol $*$ represents elementwise multiplication. The term d_i refers to the distance from each node to the closest obstacle. This value is readily available through the Euclidean signed distance field (ESDF) map, which is continuously updated during the planning process. Once the probabilities for all nodes are computed, a filtering procedure is applied to remove nodes where the target appearance probability falls below a specified threshold, p_{thr}, which in the experiments is set at 0.3. As a result, a set of motion primitives, referred to as potential nodes, is generated, each with an associated probability, which will be utilized for subsequent trajectory planning.

C. Viewpoint Generation

The set of potential nodes derived in Sect.IV-B is denoted as $\mathcal{N} = \{N_1, N_2, \ldots, N_k\}$, where each node N_k is represented by $\{\boldsymbol{x}_n, p, i\}$. In this expression, $\boldsymbol{x}_n$ refers to the three-dimensional position of the node, p is the probability of the target appearing at that node, and i is the number of times the nodes have been expanded. Initially, the mean position of all $\boldsymbol{x}_n$ in the potential nodes is calculated and denoted as $\boldsymbol{x}_{avg}$. Using $\boldsymbol{x}_{avg}$ as the center, a set of viewpoints $\mathcal{V}_j = \{V_1, V_2, \ldots, V_k\}$ is sampled on circles with varying radii. Each viewpoint V_k is represented as $\{\boldsymbol{x}, \xi\}$, where $\boldsymbol{x}$ is the position of the viewpoint and ξ is the optimal observation yaw angle for the potential nodes. This yaw is determined using a method similar to the one in [18]. However, instead of relying on the number of potential nodes, the sum of the probabilities p_{sum} of the potential nodes within the UAV's FOV is used as the criterion for selecting the optimal yaw ξ.

In contrast to [5], [6] and , in which the visibility cost function relies on computationally intensive optimization to generate high-quality trajectories, the proposed method incorporates target visibility directly into the viewpoint generation process. This approach alleviates the optimization burden at the trajectory planning level.

D. Re-capture Strategy

Inspired by autonomous exploration strategies, an exploration-based approach is applied to re-capture the target after it is lost. When the UAV begins planning the

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

target tracking trajectory, it generates i expansions of the potential nodes. In this study, i is set to 3. If the target remains within the UAV's FOV, viewpoints will be generated based on all potential nodes, and the viewpoint with the highest p_{sum} is chosen as the goal for path planning. If the target is lost from the UAV's view, a re-capture mechanism is triggered, as depicted in Fig.2. Viewpoints will now be generated based on each potential node expansion rather than using all the potential nodes obtained from i expansions. For $i = 3$, viewpoints V_1, V_2, V_3 are generated based on the potential nodes expanded at each time interval Δt. The cost associated with the viewpoints during re-capture is defined as:

$$Cost_{vpi} = \lambda_p p_{sum,i} + \lambda_e c_e + \lambda_d c_d \tag{8}$$

In this context, $p_{sum,i}$ represents the cumulative probability of the potential nodes that are observable from the optimal yaw angle of the viewpoint. The term c_e indicates the distance from each viewpoint to the nearest obstacle. Obstacles near the viewpoint can interfere with the UAV's ability to continuously observe the target and pose safety risks to its flight path. Therefore, a greater distance between the viewpoint and obstacles signifies a higher-quality viewpoint. The final term, $c_d = min\{1/||\boldsymbol{x}_j^v - \boldsymbol{x}_{j-1}^v||_2, 1\}$, where $\boldsymbol{x}_j^v$ denotes the 3D position of the viewpoint, aims to avoid excessive separation between viewpoints, which could result in inefficiencies during the search process, potentially causing the UAV to lag behind the target. The parameters λ_p, λ_e, and λ_d represent the respective weights for each cost component. Since our viewpoint selection prioritizes safety as the primary criterion, followed by identifying the optimal observation position for the target, with final minor refinement of viewpoint placement using. Through extensive experimental iterations, we determined the optimal parameter setting as $\lambda_e = 5, \lambda_p = 3, \lambda_d = 1$ in practice.

From each cluster of viewpoints generated during the expansion of potential nodes, the viewpoint with the highest total cost will be selected. The UAV will then navigate through all the viewpoints, with its yaw direction adjusted to the optimal observation yaw associated with each viewpoint.

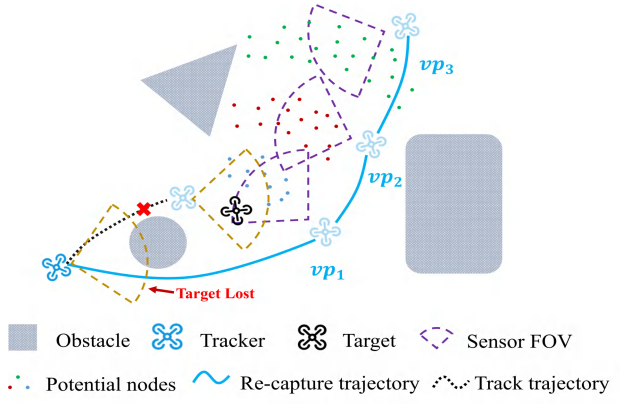

Fig. 2. When the target exiting the tracker's field of view, the original tracking trajectory (black dashed line) is aborted, and a re-capture trajectory (light blue line) is initiated to find target.

V. Trajectory Optimization

In Section IV, viewpoints were determined based on potential nodes to provide decision-making information. The next step is to develop a safe and smooth trajectory to guide the drone. Given the differentially flat property of quadrotor drones [19], planning can be performed on the flat outputs $\{x, y, z, \phi\}$. The kinodynamic A$*$ algorithm [20] is employed to generate an initial path, which is parameterized as a B-spline curve for optimization. To ensure consistent target observation, a time-dependent tracking distance cost function is introduced. This cost function, along with safety and smoothness constraints, forms a joint optimization framework to generate the position trajectory [21]. Existing methods [7], [8] simplify the yaw control by aligning it either with the trajectory's tangent direction or directly toward the target. However, such approaches may result in target loss, unstable yaw motion, or excessively high angular velocities. To address these limitations, the yaw angle is incorporated as an optimization variable in this work, guaranteeing a smooth yaw trajectory that satisfies dynamic constraints.

A. Trajectory Representation

A B-spline curve is uniquely determined by its degree q_b, a set of $N+1$ control points $\{\boldsymbol{Q}_0, \boldsymbol{Q}_1, \ldots, \boldsymbol{Q}_N\}$, and the knot vector $\{t_0, t_1, \ldots, t_M\}$, where each control point $\boldsymbol{Q}_i \in \mathbb{R}^3$ and each knot value $t_m \in \mathbb{R}$, with $M = N + q_b + 1$. This curve is a piecewise polynomial that is parameterized by t. The properties of B-spline curves, including the preservation of higher-order derivatives and the convex hull property, guarantee dynamic feasibility and safety.

B. Construction of Cost Function

For the B-spline trajectory defined by $N+1$ control points $\{\boldsymbol{Q}_0, \boldsymbol{Q}_1, \ldots, \boldsymbol{Q}_N\}$, a subset of $N+1-2q_b$ control points $\{\boldsymbol{Q}_{q_b}, \boldsymbol{Q}_{q_b+1}, \ldots, \boldsymbol{Q}_{N-q_b}\}$ is optimized. The cost function is formulated as follows, where $g(x) = \max\{0, x^3\}$ is used in the construction of the cost function:

$$\min_{\boldsymbol{Q}} J = [J_f, J_s, J_e, J_a,] \cdot \boldsymbol{\omega}, \tag{9}$$

where $\boldsymbol{\omega}$ represents a weighting vector used to balance the contribution of each cost.

1) Feasibility Cost J_f: The velocity, acceleration, and jerk of the B-spline trajectory can be calculated using the following equations:

$$\boldsymbol{V}_i = \frac{\boldsymbol{Q}_{i+1} - \boldsymbol{Q}_i}{\Delta t}, \quad \boldsymbol{A}_i = \frac{\boldsymbol{V}_{i+1} - \boldsymbol{V}_i}{\Delta t}, \quad \boldsymbol{J}_i = \frac{\boldsymbol{A}_{i+1} - \boldsymbol{A}_i}{\Delta t} \tag{10}$$

To ensure that the drone's velocity, acceleration, and jerk remain within their maximum allowable limits, a penalty term is introduced, as expressed in Equ.11:

$$J_f = \sum_{i=1}^{N-1} g(||\boldsymbol{V}_i||^2 - v_m^2) + \sum_{i=1}^{N-2} g(||\boldsymbol{A}_i||^2 - a_m^2) \tag{11}$$

where a_m and v_m denote the maximum allowable acceleration and velocity, respectively.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

2) Smoothness Cost J_s: Leveraging the convex hull property of B-spline trajectories, the second and third derivatives of the control points are minimized to enhance trajectory smoothness:

$$J_s = \sum_{i=1}^{N-1} ||\boldsymbol{A}_i||^2 + \sum_{i=1}^{N-2} ||\boldsymbol{J}_i||^2 \tag{12}$$

3) Safety Cost J_e: In trajectory planning, control points that are positioned too close to obstacles will experience a force that pushes them away from the obstacles. The penalty term is formulated in the form below:

$$J_e = \sum_{i=q_b}^{N-q_b} F_c(d(\boldsymbol{Q}_i)) \tag{13}$$

$$F_c(d(\boldsymbol{Q}_i)) = \begin{cases} (d(\boldsymbol{Q}_i) - d_{thr})^2 & d(\boldsymbol{Q}_i) \le d_{thr} \\ 0 & d(\boldsymbol{Q}_i) > d_{thr} \end{cases} \tag{14}$$

Where $d(\boldsymbol{Q}i)$ represents the Euclidean distance from the control point to the nearest obstacle, which can be derived from the ESDF map. A thrust is applied when this distance falls below the threshold d_{thr}.

4) Distance Cost J_a: Since the proposed tracking strategy selects the optimal observation point for potential nodes as the endpoint, only the tracking distance needs to be considered at the trajectory optimization level. The approach used in [5] overlooks the tracking duration, which can result in parallel tracking problems, as depicted in Fig.3. "Parallel" tracking refers to the scenario where the target can be easily obstructed by obstacles. To address this issue, a distance-based cost function is developed that optimizes the tracker's trajectory during the tracking process. This ensures that the trajectory remains closer to the target's path, reducing the risk of target loss.

$$D_a(\boldsymbol{Q}_i) = \begin{cases} ||\boldsymbol{Q}_i - \boldsymbol{x}_{avg,1}|| & 0 < t_i < \Delta t \\ ||\boldsymbol{Q}_i - \boldsymbol{x}_{avg,2}|| & \Delta t \le t_i < 2\Delta t \\ \vdots & \\ ||\boldsymbol{Q}_i - \boldsymbol{x}_{avg,j}|| & (j-1)\Delta t \le t_i < j\Delta t \end{cases} \tag{15}$$

$$J_a(\boldsymbol{Q}_i) = \begin{cases} (D_a(\boldsymbol{Q}_i) - d_{min})^2 & 0 < D_a(\boldsymbol{Q}_i) < d_{min} \\ (d_{max} - D_a(\boldsymbol{Q}_i))^2 & D_a(\boldsymbol{Q}_i) \ge d_{min} \\ 0 & else \end{cases} \tag{16}$$

$\boldsymbol{x}_{avg,j}$ is the average position of the potential nodes' j-th expansion . d_{min},d_{max} are threshold of optimal tracking distance.

C. Construction of Yaw Cost Function

To ensure continuous optimal observation of the target during tracking, the drone's yaw trajectory is formulated as an optimization problem using B-spline curves. Similar to the previous section, the overall cost function is given by Equ.17. The vector $\boldsymbol{\lambda}$ represents the weights used to balance the various costs involved in the optimization.

$$J = [J_{\psi a}, J_{\psi f}, J_{\psi s}] \cdot \boldsymbol{\lambda} \tag{17}$$

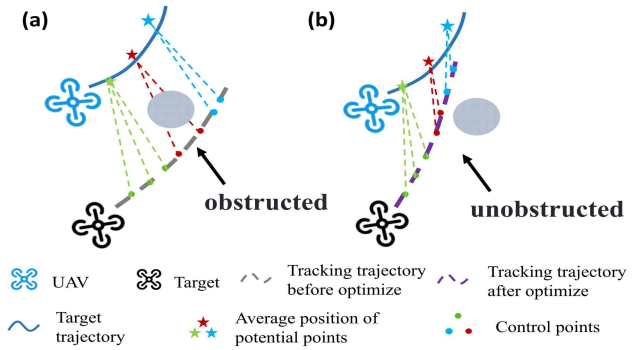

Fig. 3. (a): "Parallel" tracking trajectory before distance optimize, which means target is easily obstructed by an obstacle. (b):Tracking trajectory after distance optimize.The tracker follows the target more closely.

1) Tracking Yaw Cost $J_{\psi a}$: During tracking, it is desired that the drone's yaw continuously aligns with the target to achieve optimal tracking performance. This cost is formulated as shown in Equ.18:

$$J_{\psi a} = \sum_{i=1}^{N} (\psi_{c,i} - \psi_{target,i})^2 \tag{18}$$

$\psi_{c,i}$ denotes the yaw of the current control point, while $\psi_{best,i}$ represents the yaw direction towards the target.

2) Feasibility Yaw Cost $J_{\psi f}$: The second term addresses the dynamic constraints, formulated similarly to the positional trajectory:

$$J_{\psi f} = \sum_{i=1}^{N-1} g(||\dot{\psi}_i||^2 - \dot{\psi}_m^2) + \sum_{i=1}^{N-2} g(||\ddot{\psi}_i||^2 - \ddot{\psi}_m^2) \tag{19}$$

where $\dot{\psi}_m$ and $\ddot{\psi}_m$ represent the maximum yaw angular velocity and acceleration, respectively.

3) Smoothness Yaw Cost $J_{\psi s}$: The third term guarantees that the entire yaw trajectory maintains sufficient smoothness:

$$J_{\psi s} = \sum_{i=1}^{N-1} ||\ddot{\psi}_i||^2 + \sum_{i=1}^{N-2} ||\dddot{\psi}_i||^2 \tag{20}$$

VI. SIMULATION AND BENCHMARK

This section presents three simulation scenarios to evaluate the proposed method in complex environments. The first scenario introduces obstacles causing occlusion and target loss, assessing the effectiveness of the re-capture strategy. The second evaluates trajectory optimization by tracking a highly maneuverable target. The third tests the whole system's robustness in environments with dense obstacles and dynamic target movements. We define average distance error E_d (distance between UAV and target), average yaw error E_y (yaw angular deviation from target direction), tracking success rate R_t, and track maintenance percentage M_p (target-in-FOV duration proportion during tracking) as metrics to evaluate the tracking robustness of each method. The results are compared against state-of-the-art methods like Fast Tracker [8], Vis Planner [5], and Elastic Tracker [7] to highlight the benefits of the proposed approach. The tracker was equipped with a front-facing camera with a $90° \times 60°$ FOV, and the maximum tracking distance

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

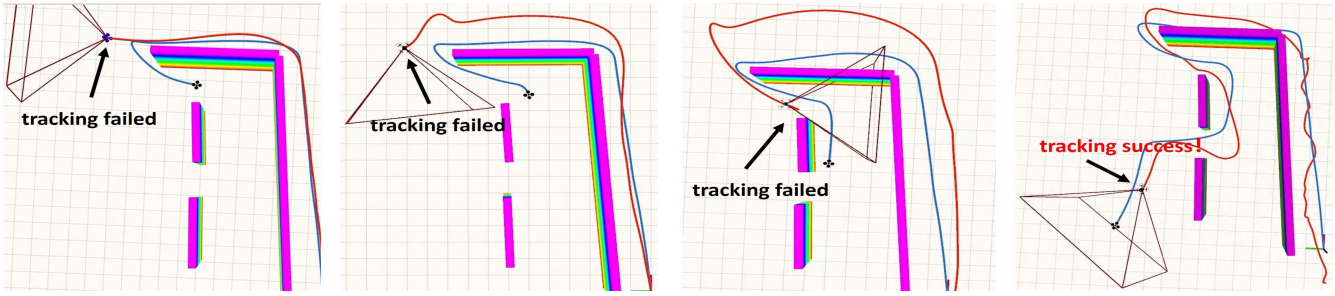

Fig. 4. The results of track strategy test are shown. The cone-shaped frame indicates the tracker's FOV, the red trajectory represents the tracker, the blue trajectory represents the target. (The following images are represented in the same way.)

was set to 4.5 meters, with the target being another high-speed drone.

The maximum speeds of both the target and the tracker vary across the different simulation scenarios, with the tracker's speed always being greater than that of the target. To ensure a fair comparison of the strengths and weaknesses of each method, the maximum speed, maximum acceleration, and the tracker's maximum speed and acceleration were kept consistent across all scenarios. All simulations were conducted on a laptop equipped with an Intel i7-12650 CPU, Nvidia RTX 4060 GPU, and 16GB of RAM.

A. Track Strategy Test

To evaluate the effectiveness of the proposed motion prediction and re-capture strategy, a maze-like environment was created, where the target navigates obstacles and attempts to evade the tracker's FOV by accelerating around the first corner. The target starts at 2 m/s and accelerates to 4 m/s after the corner. Once lost, the tracker receives no further position updates. Each method was tested 20 times, and the tracking success rate was recorded.

As shown in Fig.4, Elastic Tracker and Vis Planner lack re-capture mechanisms, making continuous tracking in the maze difficult. Fast Tracker, though employing a relocation strategy, relies heavily on motion prediction accuracy, limiting its effectiveness when the target makes large maneuvers or remains lost for extended periods. In contrast, the proposed method conducts a "reconnaissance" to estimate the target's potential locations instead of solely extrapolating its last known trajectory. This approach significantly improves target reacquisition and ensures successful tracking in the maze. Since the tracking process remains stable before target loss and no error data exists after loss, we select the R_t as the sole evaluation metric for this scenario, as shown in Tab.I.

TABLE I
TRACKING ROBUSTNESS METRICS IN TRACK STRATEGY TEST

Method	Fast Tracker	Vis Planner	Elastic Tracker	Ours
R_t	30%	0%	45%	**75%**

B. Track Trajectory Planning Test

To evaluate the performance of the proposed trajectory tracking optimization method, a scenario was set where the

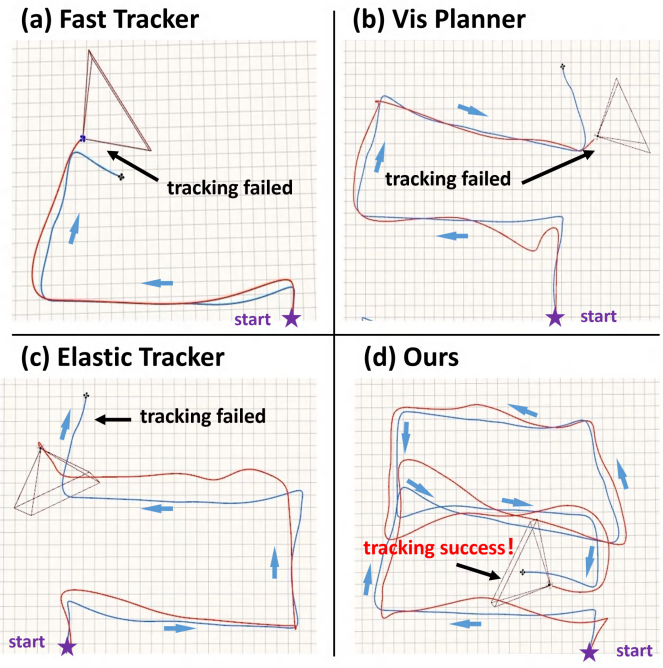

Fig. 5. The comparison of various approaches in track trajectory planning tests. Blue arrows depict the movement directions of both the tracker and the target.

UAV tracks a high-speed target performing continuous sharp turns in an environment without obstacles. The target's maximum speed is 5.5 m/s, and its position is continuously shared with the tracker. This scenario provides a substantial challenge for tracking trajectory planning algorithms. Without any yaw angle planning during tracking, Fast Tracker fails to keep up with the target when it makes sharp turns. In the Vis Planner framework, poor robustness is observed in this scenario, always leading to crashes. The Elastic Tracker utilizes distance-based optimization with elastic constraints. However, it exhibits limitations in high-speed scenarios due to insufficient consideration of temporal factors, leading to substantial fluctuations in the tracker-target distance. In contrast, the proposed method optimizes the yaw angle trajectory during tracking and integrates time-dependent distance constraints into the planning process, enabling the tracker to maintain an optimal distance for continuous target observation, as shown in Fig.5. The methods' tracking robustness metrics are shown

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2026, Vienna, Austria. Cite as RA-L paper.

in Fig.6 and Tab.II.

TABLE II
TRACKING ROBUSTNESS METRICS IN TRACK TRAJECTORY PLANNING TEST

Method	Fast Tracker	Vis Planner	Elastic Tracker	Ours
R_t	65%	0%	**100%**	**100%**
M_p	63.4%	\	76.8%	**84.3%**

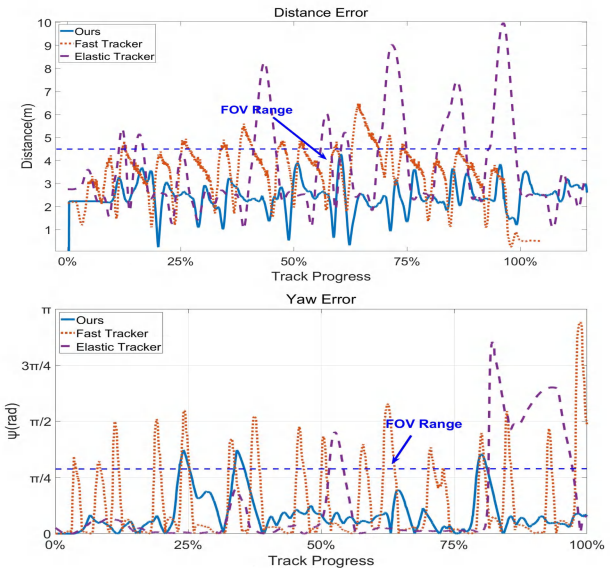

Fig. 6. The figure above compares the yaw angle errors of the three methods, while the figure below shows the tracking distance errors for each method.

C. General Test

To thoroughly evaluate the algorithm's performance, a simulation environment with a randomly generated forest containing 80 obstacles was used. In this environment, the UAV tracks targets with varying maximum speeds V and the optimal tracking distance was set at $2.5m$. To better simulate real-world experimental conditions, we introduce Gaussian noise $g \sim \mathcal{N}(0, \sigma^2)$ to the target's position to evaluate method performance under noisy observations. For guaranteeing equitable assessment in noisy environments, we implemented a standardized outlier-rejection filter across all methods, which removes inter-frame positional discontinuities exceeding dynamically adjusted thresholds. Each method was tested in 20 trials, shown as Fig.7 and Tab.III. Fast Tracker struggles to maintain the target when it performs sharp turns and noise is introduced. Vis Planner faces challenges in successfully planning trajectories within this scenario. Elastic Tracker, lacking a re-localization strategy after losing the target, tends to lose track of the target when it navigates through dense obstacles, rendering it unable to continue tracking. In contrast, the proposed tracking method performs exceptionally well across all scenarios and noise demonstrating superior robustness in this environment.

VII. REAL-WORLD EXPERIMENTS

To further assess the effectiveness of the proposed re-capture strategy and trajectory optimization method, practical outdoor

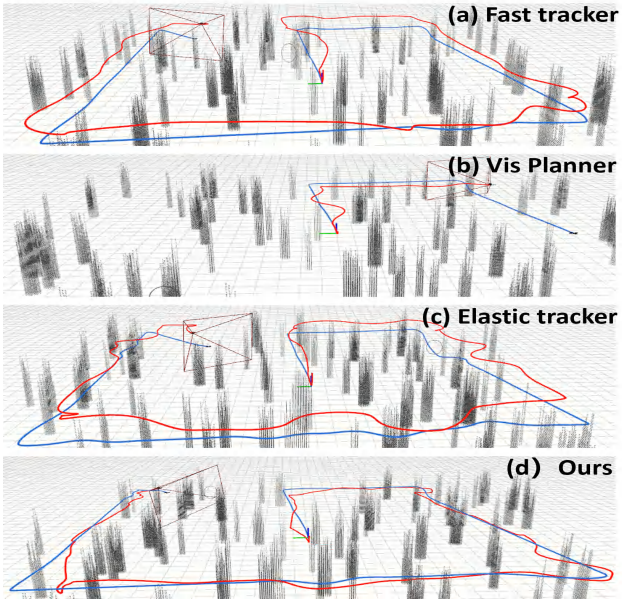

Fig. 7. Comparison of tracking methods in random environment.

TABLE III
TRACKING ROBUSTNESS METRICS IN GENERAL TEST

		V(m/s)	R_t	E_y(rad)	E_d(m)	M_p
Fast Tracker	$\sigma^2 = 0$	2	100%	0.19	1.11	85%
		3	100%	0.25	2.01	84%
		4	90%	0.26	3.24	64%
	$\sigma^2 = 3, 5$			\		
Vis Planner	$\sigma^2 = 0, 3, 5$			\		
Elastic Tracker	$\sigma^2 = 0$	2	100%	0.12	**2.56**	91%
		3	100%	0.18	2.90	83%
		4	100%	0.22	3.53	75%
	$\sigma^2 = 3$	2	100%	0.18	2.36	**94%**
		3	100%	**0.15**	3.23	69%
		4	85%	0.24	3.81	51%
	$\sigma^2 = 5$	2	100%	**0.18**	2.74	83%
		3	90%	0.26	2.93	65%
		4	85%	**0.23**	5.01	53%
Ours	$\sigma^2 = 0$	2	100%	**0.11**	2.60	**95%**
		3	100%	**0.10**	**2.44**	**92%**
		4	100%	**0.12**	**2.32**	**84%**
	$\sigma^2 = 3$	2	100%	**0.09**	**2.49**	89%
		3	100%	0.17	**2.33**	**88%**
		4	100%	**0.14**	**2.53**	**81%**
	$\sigma^2 = 5$	2	100%	0.20	**2.30**	**94%**
		3	75%	**0.19**	**2.22**	**83%**
		4	85%	0.30	**2.34**	**75%**

experiments were conducted in a forest environment, as shown in Fig.8. The experimental platform consisted of quadrotors equipped with an Intel i7-11650 processor, an Intel RealSense Depth Camera D435i, and a USB camera for perception. The onboard computers were responsible for tasks such as state estimation, mapping, coordination, motion planning, and control. The maximum speed of the target was 2.13 m/s. For detection and localization, AprilTag [21] was utilized to

provide the target's position in the real world.

During the experiment, the target navigated irregularly through the forest and attempted to escape UAV's FOV to test the re-capture strategy. As shown in Fig.9 and the attached video[1], the target vanished from the UAV's FOV 4 times. Each time, after the re-capture mechanism was activated, the UAV successfully re-capture the target and resumed the tracking task. The trajectory optimization method, which incorporated time-dependent distance penalties and yaw angles, ensured the generation of high-quality trajectories throughout the tracking process. The robustness metrics obtained from real-world experiments are $E_y = 0.16$ rad , $E_d = 2.23$ m and $M_p = 89.39\%$, demonstrating the effectiveness of our method.

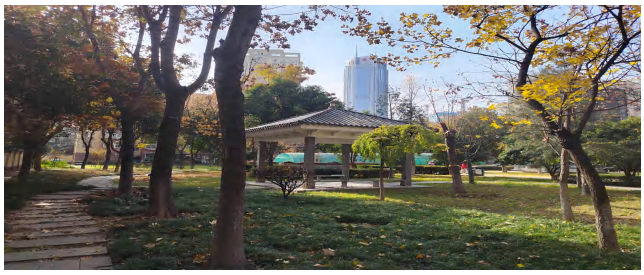

Fig. 8. Outdoor scenario for the tracking tests.

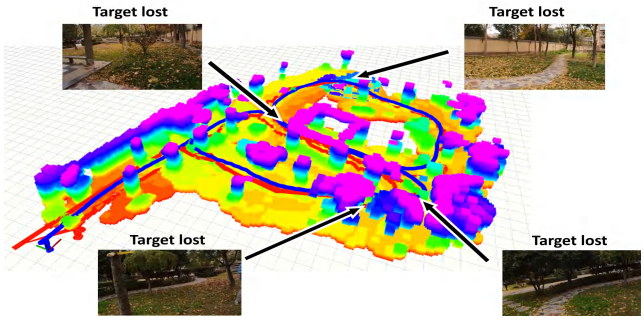

Fig. 9. Visualization of the online-generated map and tracking trajectories. The four inset images depict the UAV's first-person camera view. At these four positions, the target is lost, triggering the re-capture mechanism.

VIII. Conclusion

This paper introduces a framework for autonomous target tracking, utilizing a Markov-chain-based method for predicting target motion to estimate the likelihood of the target's future position. Leveraging these predictions, a tracking strategy is devised alongside a re-capture mechanism for situations when the target is lost, thereby minimizing the risk of tracking failure. Additionally, various constraints, including tracking distance, are integrated into trajectory optimization, with the yaw angle incorporated as an optimization variable, enhancing the system's robustness. Both simulations and real-world experiments validate the superiority of the proposed approach.

References

[1] P. Pueyo, J. Dendarieta, E. Montijano, A. C. Murillo, and M. Schwager, "Cinempc: A fully autonomous drone cinematography system incorporating zoom, focus, pose, and scene composition," *IEEE Transactions on Robotics*, vol. 40, pp. 1740–1757, 2024.

[1]https://youtu.be/kNcbNbwhgMY.

[2] B. F. Jeon and H. J. Kim, "Online trajectory generation of a mav for chasing a moving target in 3d dense environments," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 1115–1121.

[3] V. Krátký, A. Alcántara, J. Capitán, P. Štěpán, M. Saska, and A. Ollero, "Autonomous aerial filming with distributed lighting by a team of unmanned aerial vehicles," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 7580–7587, 2021.

[4] G. Wang, J. Qin, Q. Liu, Q. Ma, and C. Zhang, "Image-based visual servoing of quadrotors to arbitrary flight targets," *IEEE Robotics and Automation Letters*, vol. 8, no. 4, pp. 2022–2029, 2023.

[5] Q. Wang, Y. Gao, J. Ji, C. Xu, and F. Gao, "Visibility-aware trajectory optimization with application to aerial tracking," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 5249–5256.

[6] H. Wang, X. Zhang, Y. Liu, X. Zhang, and Y. Zhuang, "Svpto: Safe visibility-guided perception-aware trajectory optimization for aerial tracking," *IEEE Transactions on Industrial Electronics*, vol. 71, no. 3, pp. 2716–2725, 2023.

[7] J. Ji, N. Pan, C. Xu, and F. Gao, "Elastic tracker: A spatio-temporal trajectory planner for flexible aerial tracking," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 47–53.

[8] Z. Han, R. Zhang, N. Pan, C. Xu, and F. Gao, "Fast-tracker: A robust aerial system for tracking agile target in cluttered environments," in *2021 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2021, pp. 328–334.

[9] A. Dmytruk, G. Silano, D. Bicego, D. B. Licea, and M. Saska, "A perception-aware nmpc for vision-based target tracking and collision avoidance with a multi-rotor uav," in *2022 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, 2022, pp. 1668–1673.

[10] T. Nägeli, J. Alonso-Mora, A. Domahidi, D. Rus, and O. Hilliges, "Real-time motion planning for aerial videography with dynamic obstacle avoidance and viewpoint optimization," *IEEE Robotics and Automation Letters*, vol. 2, no. 3, pp. 1696–1703, 2017.

[11] K. Zhang, Y. Shi, and H. Sheng, "Robust nonlinear model predictive control based visual servoing of quadrotor uavs," *IEEE/ASME Transactions on Mechatronics*, vol. 26, no. 2, pp. 700–708, 2021.

[12] W. Zhao, H. Liu, F. L. Lewis, K. P. Valavanis, and X. Wang, "Robust visual servoing control for ground target tracking of quadrotors," *IEEE Transactions on Control Systems Technology*, vol. 28, no. 5, pp. 1980–1987, 2019.

[13] J. Chen, T. Liu, and S. Shen, "Tracking a moving target in cluttered environments using a quadrotor," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 446–453.

[14] B. Jeon, Y. Lee, and H. J. Kim, "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)*. IEEE, 2020, pp. 1243–1249.

[15] B. F. Jeon, C. Kim, H. Shin, and H. J. Kim, "Aerial chasing of a dynamic target in complex environments," *International Journal of Control, Automation and Systems*, vol. 20, no. 6, pp. 2032–2042, 2022.

[16] Y. Gao, J. Ji, Q. Wang, R. Jin, Y. Lin, Z. Shang, Y. Cao, S. Shen, C. Xu, and F. Gao, "Adaptive tracking and perching for quadrotor in dynamic scenarios," *IEEE Transactions on Robotics*, vol. 40, pp. 499–519, 2023.

[17] Z. Xu, X. Zhan, B. Chen, Y. Xiu, C. Yang, and K. Shimada, "A real-time dynamic obstacle tracking and mapping system for uav navigation and collision avoidance with an rgb-d camera," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 10 645–10 651.

[18] C. Witting, M. Fehr, R. Bähnemann, H. Oleynikova, and R. Siegwart, "History-aware autonomous exploration in confined environments using mavs," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 1–9.

[19] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," in *2011 IEEE international conference on robotics and automation*. IEEE, 2011, pp. 2520–2525.

[20] 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.

[21] X. Zhou, Z. Wang, H. Ye, C. Xu, and F. Gao, "Ego-planner: An esdf-free gradient-based local planner for quadrotors," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 478–485, 2020.