

DMVC-Tracker: Distributed Multi-Agent Trajectory Planning for Target Tracking Using Dynamic Buffered Voronoi and Inter-Visibility Cells

Yunwoo Lee , Jungwon Park , and H. Jin Kim 

Abstract—This letter presents a distributed trajectory planning method for multi-agent aerial tracking. The proposed method uses a Dynamic Buffered Voronoi Cell (DBVC) and a Dynamic Inter-Visibility Cell (DIVC) to formulate the distributed trajectory generation. Specifically, the DBVC and the DIVC are time-variant spaces that prevent mutual collisions and occlusions among agents, while enabling them to maintain suitable distances from the moving target. We combine the DBVC and the DIVC with an efficient Bernstein polynomial motion primitive-based tracking trajectory generation method, which has been refined into a less conservative approach than in our previous work. The proposed algorithm can compute each agent's trajectory within several milliseconds on an Intel i7 desktop. We validate the tracking performance in challenging scenarios, including environments with dozens of obstacles.

Index Terms—Path planning for multiple mobile robots, distributed robot systems, aerial tracking.

I. INTRODUCTION

AERIAL target tracking has been widely applied in fields such as cinematography and surveillance. Although a single micro aerial vehicle (MAV) is usually employed in these applications, the utilization of multiple MAVs can bring benefits. For example, multiple views of actors captured by a team of MAVs provide movie directors with more footage [1], [2], [3]. Also, multi-agent tracking can be deployed in moving motion-capture systems [4], [5], [6], and a large number of cameras increases the accuracy of pose estimation.

Despite great attention and research on motion generation for multi-robot systems, multi-agent target tracking remains a challenging task. The main challenge is finding constraints that prevent both inter-agent occlusion and inter-agent collision, while considering the moving target. Additionally, several other requirements should be considered: occlusion and collision

Received 4 November 2024; accepted 3 March 2025. Date of publication 14 March 2025; date of current version 7 April 2025. This article was recommended for publication by Associate Editor M. Alhafnawi and Editor M. Ani Hsieh upon evaluation of the reviewers' comments. This work was supported by IITP-ITRC under Grant IITP-2025-RS-2024-00437268. (Corresponding author: H. Jin Kim.)

Yunwoo Lee is with the Artificial Intelligence Institute of Seoul National University (AIIS), Seoul 08826, South Korea (e-mail: snunoo12@snu.ac.kr).

Jungwon Park is with the Department of Mechanical System Design Engineering, Seoul National University of Science and Technology, Seoul 01811, South Korea (e-mail: jungwonpark@seoultech.ac.kr).

H. Jin Kim is with the Department of Aerospace Engineering, Seoul National University, Seoul 08826, South Korea (e-mail: hjinkim@snu.ac.kr).

This article has supplementary downloadable material available at <https://doi.org/10.1109/LRA.2025.3551255>, provided by the authors.

Digital Object Identifier 10.1109/LRA.2025.3551255

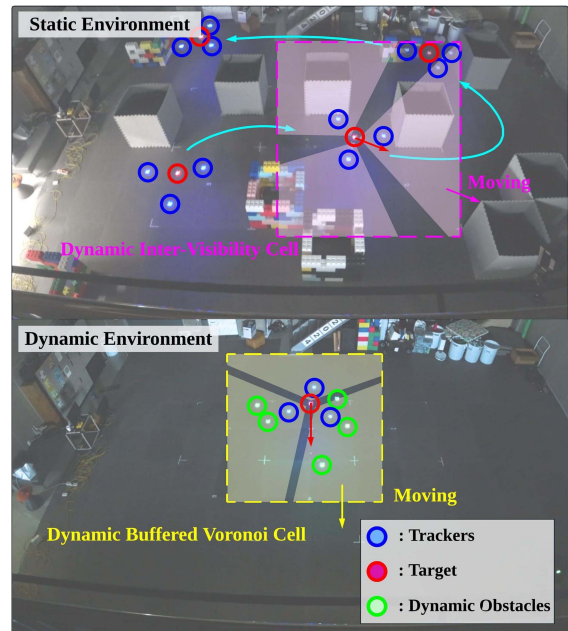


Fig. 1. Hardware demonstration of multi-agent target tracking.

against obstacles, actuator limits, and tracking distances. To respond to frequent changes in the motion of the target and moving obstacles, the trajectory planning should be updated quickly to reflect these considerations.

In this letter, we present an online distributed trajectory planning algorithm that can generate a target-visible and safe trajectory. The key ideas of the proposed method are Dynamic Buffered Voronoi Cell (DBVC) and Dynamic Inter-Visibility Cell (DIVC). The DBVC is a time-varying, inter-collision-free region developed from Buffered Voronoi Cell (BVC) [7]. This time-varying region helps maintain consistent distances between the target and the agents, whereas the BVC can cause an agent to become stuck in a static space, making it difficult for the agent to follow the target. We design the DIVC to prevent inter-agent occlusion. Specifically, the DIVC ensures that the *Line-of-Sight* connecting the target and an agent does not collide with other agents. Based on the DBVC and the DIVC, the proposed planner generates a target-visible and safe trajectory using Bernstein polynomial motion primitives. We employ a *sample-check-select* strategy and leverage properties of Bernstein polynomials to speed up the calculation, enabling us effectively to tackle complex tracking problems. This paper

reduces the conservativeness of the *check* method from our previous work [8] to achieve a higher success rate. The main contributions of this paper are as follows.

- A distributed multi-agent trajectory planning algorithm for target tracking that generates a collision-free and occlusion-free trajectory in complex environments, such as dozens of obstacles.
- Dynamic Buffered Voronoi Cell (DBVC) and Dynamic Inter-Visibility Cell (DIVC) that impose inter-agent collision and inter-agent occlusion avoidance constraints, while adapting to the future movements of the target.
- An integration of the DBVC and the DIVC with an improved Bernstein polynomial motion primitive-based trajectory generator, which lessens the conservativeness of our previous method [8].

II. RELATED WORKS

A. Target Tracking Using a Single Drone

Various studies have proposed methods that consider target visibility for single-drone path planning in obstacle environments. [9], [10], [11] address target occlusion in the presence of static and unstructured obstacles. To be specific, [9] and [10] design polytope-shaped target-visible regions and formulate a spatio-temporal optimization problem. [11] proposes a visibility metric using Euclidean Signed Distance Fields (ESDF) and formulates a graph optimization to select optimal viewpoints.

There are works that address target tracking problems in dynamic obstacle environments [12], [13], [14]. [12] designs a visibility cost to avoid target occlusion, inspired by the GPU ray casting model. [13] utilizes partial convex structures of the target tracking optimization problem to handle non-convex visibility constraints, while [14] designs a time-varying half-space-shaped region to apply visibility constraints within a quadratic programming problem.

The above approaches can be utilized for multi-agent target tracking missions by treating neighboring drones as dynamic obstacles. However, such approaches may result in collisions between agents or occlusions of the target by neighboring agents, as the newly updated trajectories may differ from the predicted movements of other agents. In contrast, our planner enables cooperative target tracking, in a *distributed* manner.

B. Target Tracking Using Multiple Drones

Research on target tracking using multiple MAVs has been widely studied in recent years. [3], [4], [5] focus on the formation of MAVs to follow the moving target. [3] and [4] decompose the multi-MAV trajectory optimization to tackle non-convex constraints, which allows replanning in real-time. Both methods calculate viewpoints considering formations, and then successive optimization generates a safe [3] and smooth [4] trajectory. [5] develops external control inputs to prevent inter-agent collision, enabling the conversion of the optimization problem into a convex one. However, the above approaches address only collision avoidance with respect to the target and neighboring agents, while omitting considerations of visibility issues caused by neighboring agents.

On the other hand, there are studies that consider mutual visibility. [1] optimize MAVs' viewpoints in discretized state and time spaces in a centralized manner. The finer the discretized states, the more computation time and memory are

TABLE I
NOMENCLATURE

Symbol	Definition
$\mathbf{x}_i^k(t)$	Trajectory of the i -th agent.
$\mathbf{x}_q(t)$	Predicted trajectory of the target.
$\mathbf{x}_o^k(t)$	Predicted trajectory of the k -th dynamic obstacle.
N_c, N_o	The number of agents and dynamic obstacles.
\mathcal{I}_c	An agent index set. $\mathcal{I}_c = \{1, \dots, N_c\}$
\mathcal{I}_o	A dynamical obstacle index set. $\mathcal{I}_o = \{1, \dots, N_o\}$
r_c, r_o, r_q	Radius of agents, dynamic obstacles and a target.
T	Planning horizon.
$\mathcal{X}, \mathcal{T}(t)$	Environment, Occupied space by the target.
$\mathcal{C}_i(t), \mathcal{C}(t)$	Occupied space by the i -th agent and all agents.
$\mathcal{C}^i(t)$	Occupied space excluding $\mathcal{C}_i(t)$ from $\mathcal{C}(t)$
$\mathcal{F}_s, \mathcal{F}_d(t)$	Static-obstacle-free space, obstacle-free space
$\mathcal{F}_q(t)$	The space in $\mathcal{F}_d(t)$ excluding the target's area.
$\mathcal{F}_c^i(t)$	The space in \mathcal{X} excluding $\mathcal{C}^i(t)$.
$\mathcal{H}_s^{ij}(t)$	DBVC for the i -th agent against the j -th agent.
$\bigcap_{\mu=1,2} \mathcal{H}_{\mu k}^{ij}(t)$ $\mu = o$ or a	DIVC for the i -th agent against the j -th agent. o : obtuse case, a : acute case.
$\mathcal{B}(\mathbf{x}, r)$	A ball with center at \mathbf{x} and radius r .
$\mathcal{L}(\mathbf{x}_a, \mathbf{x}_b)$	A segment connecting points at \mathbf{x}_a and \mathbf{x}_b .
\cup, \cap, \setminus	Set operator: union, intersect, except
\oplus	Minkowski-sum operator
$\ \mathbf{x}\ , \mathbf{x}^\top$	Euclidean norm and transpose of a vector \mathbf{x} .
$[A; B]$	Row-wise concatenation of matrices A and B
AB	A segment connecting points A and B .

required. Therefore, the authors set the number of time steps small to enable real-time execution of the algorithm. The methods in [2] and [15] try to avoid the appearance of the other MAVs in the MAV's camera image. Both methods operate in a distributed manner to meet real-time criteria; however, [2] uses a priority-based approach, while [15] employs a sequential consensus approach. Therefore, there may be situations where all mutual visibility constraints are not simultaneously satisfied. In contrast, our method ensures inter-agent occlusion avoidance and does not require sequential planning updates, resulting in a fully distributed approach.

C. Target Tracking in Crowded Environments

Most trajectory planning research aims to ensure real-time performance. The works [12], [16], and [17] formulate non-convex optimization methods to generate a target-visible trajectory; however, as the number of obstacles increases, computation time quickly increases. To address this, this work employs Bernstein polynomial motion primitives to efficiently find a tracking trajectory by leveraging the properties of Bernstein polynomials. This work reduces the conservativeness of the approach in our previous work [8], further improving the tracking success rate in challenging conditions, such as environments with numerous dynamic obstacles.

III. PROBLEM FORMULATION

In this section, we formulate a trajectory planning problem for multiple tracking agents. We suppose that N_c homogenous agents are deployed to follow a target in a 2-dimensional space $\mathcal{X} \subset \mathbb{R}^2$ with static and N_o dynamic obstacles. We aim to generate trajectories so that the agents 1) avoid collisions, 2) maintain the visibility of the target and 3) do not exceed dynamical limits.

For consistent image acquisition of the target, we fix the flying altitude, which means we address the 2D problem. Unlike methods that allow drones to move obstacle-free high altitudes [18], [19], since the drone must find feasible movements only on the x - y plane, our problem is more challenging. Throughout this paper, the notations in Table I are used.

A. Environments

The environment \mathcal{X} consists of target space $\mathcal{T}(t)$, obstacle space $\mathcal{O}(t)$, and drone space $\mathcal{C}(t)$. $\mathcal{O}(t)$ is divided into spaces occupied by static and dynamic obstacles, \mathcal{O}_s and $\mathcal{O}_d(t)$, respectively. $\mathcal{C}(t)$ is an occupied space by all agents and is partitioned into the i -th agent space $\mathcal{C}_i(t)$, $i \in \mathcal{I}_c$. We define the following free spaces for target-tracking missions.

$$\begin{aligned} \mathcal{F}_s &= \mathcal{X} \setminus \mathcal{O}_s, \mathcal{F}_d(t) = \mathcal{F}_s \setminus \mathcal{O}_d(t), \mathcal{F}_q(t) = \mathcal{F}_d(t) \setminus \mathcal{T}(t), \\ \mathcal{F}_c^i(t) &= \mathcal{X} \setminus \mathcal{C}^i(t), \text{ where } \mathcal{C}^i(t) = \mathcal{C}(t) \setminus \mathcal{C}_i(t). \end{aligned} \quad (1)$$

B. Trajectory Representation

Due to the virtue of differential flatness of the dynamics of various MAV platforms, such as quadrotors, we represent trajectories of all agents as polynomial functions of time t . Employing the Bernstein basis, we represent the trajectory of the i -th agent, $\mathbf{x}_c^i(t) \in \mathbb{R}^2$, $i \in \mathcal{I}_c$ as follows:

$$\mathbf{x}_c^i(t) = \mathbf{C}^{i\top} \mathbf{b}_{n_c}(t), \quad t \in [0, T] \quad (2)$$

where T is the planning horizon, n_c is the degree of the polynomials, $\mathbf{C}^i \in \mathbb{R}^{(n_c+1) \times 2}$ is a coefficient matrix for the agent i , and $\mathbf{b}_{n_c}(t) \in \mathbb{R}^{(n_c+1) \times 1}$ is a vector that consists of n_c -th order Bernstein bases for time interval $[0, T]$. We represent the trajectory of the target, $\mathbf{x}_q(t)$, the k -th dynamic obstacles, $\mathbf{x}_o^k(t)$, $k \in \mathcal{I}_o$ in the same manner.

C. Assumptions

In this study, we make the following assumptions.

- *Agents*: All agents share their current positions, and they start trajectory planning at the same time.
- *Obstacles*: The information about the static obstacle space \mathcal{O}_s is given as a point cloud, and the current positions of dynamic obstacles can be acquired.
- *Target*: The current position of the target can be observed.
- *Shape*: The moving objects, such as the target, dynamic obstacles, and tracking agents are modeled as balls with radius r_q , r_o and r_c , respectively.

In this paper, we use extended Kalman filters to estimate the velocity of each moving object, and the future trajectories of the obstacles and the target are predicted using a constant velocity model and the method described in [8], respectively.

D. Mission Description

In the multi-agent tracking trajectory planning, we focus on the following objectives.

1) *Collision Avoidance*: For safety, all agents should not collide with obstacles, the target, and the other agents.

$$\mathcal{B}(\mathbf{x}_c^i(t), r_c) \subset \mathcal{F}_q(t) \cap \mathcal{F}_c^i(t), \quad \forall i \in \mathcal{I}_c, \quad \forall t \in [0, T] \quad (3)$$

2) *Occlusion Avoidance*: To avoid target occlusion caused by obstacles or neighboring agents, the *Lines-of-Sight* between the agents and the target should not intersect with any obstacles and other agents.

$$\mathcal{L}(\mathbf{x}_c^i(t), \mathbf{x}_q(t)) \subset \mathcal{F}_d(t) \cap \mathcal{F}_c^i(t), \quad \forall i \in \mathcal{I}_c, \quad \forall t \in [0, T] \quad (4)$$

3) *Target Distance*: In order to avoid being too close or too far from the target, we formulate the distance constraints.

$$\|\mathbf{x}_c^i(t) - \mathbf{x}_q(t)\| \in [d_{\min}, d_{\max}], \quad \forall i \in \mathcal{I}_c, \quad \forall t \in [0, T] \quad (5)$$

4) *Dynamical Limits*: Due to the actuator limits of the drone, the trajectory should not exceed the maximum velocity, v_{\max} , and acceleration, a_{\max} . Also, the yaw rate should not exceed $\dot{\psi}_{\max}$ to prevent motion blurs in the camera.

$$\begin{aligned} \|\dot{\mathbf{x}}_c^i(t)\| &\leq v_{\max}, \quad \|\ddot{\mathbf{x}}_c^i(t)\| \leq a_{\max}, \quad |\dot{\psi}_c^i(t)| \leq \dot{\psi}_{\max}, \\ &\forall i \in \mathcal{I}_c, \quad \forall t \in [0, T] \end{aligned} \quad (6)$$

IV. DYNAMIC CELLS FOR MULTI-AGENT TRACKING

In this section, we design time-varying regions free from inter-agent collisions and inter-agent occlusions, which are suitable for dynamic target tracking.

A. Dynamic Buffered Voronoi Cell

We consider the inter-collision-free region that moves along the target's predicted trajectory. We define a Dynamic Buffered Voronoi Cell (DBVC) to avoid collisions between the i -th and j -th agents as follows:

$$\mathcal{H}_s^{ij}(t) = \left\{ \mathbf{x}(t) \in \mathbb{R}^2 \mid (\mathbf{x}_{c0}^j - \mathbf{x}_{c0}^i)^\top \left(\mathbf{x}(t) - \mathbf{x}_q(t) + \mathbf{x}_{q0} - \frac{\mathbf{x}_{c0}^i + \mathbf{x}_{c0}^j}{2} \right) + r_c \|\mathbf{x}_{c0}^j - \mathbf{x}_{c0}^i\| \leq 0 \right\}, \quad (7a)$$

$$\mathcal{H}_s^{ji}(t) = \left\{ \mathbf{x}(t) \in \mathbb{R}^2 \mid (\mathbf{x}_{c0}^i - \mathbf{x}_{c0}^j)^\top \left(\mathbf{x}(t) - \mathbf{x}_q(t) + \mathbf{x}_{q0} - \frac{\mathbf{x}_{c0}^j + \mathbf{x}_{c0}^i}{2} \right) + r_c \|\mathbf{x}_{c0}^i - \mathbf{x}_{c0}^j\| \leq 0 \right\} \quad (7b)$$

Lemma 1: If $\mathbf{x}_c^i(t) \in \mathcal{H}_s^{ij}(t)$ and $\mathbf{x}_c^j(t) \in \mathcal{H}_s^{ji}(t)$, $i \neq j \in \mathcal{I}_c$, the i -th and j -th agents do not collide with each other.

Proof: When $\mathbf{x}_c^i(t) \in \mathcal{H}_s^{ij}(t)$ and $\mathbf{x}_c^j(t) \in \mathcal{H}_s^{ji}(t)$, the summation of constraints in (7) yields:

$$\begin{aligned} (\mathbf{x}_{c0}^j - \mathbf{x}_{c0}^i)^\top (\mathbf{x}_c^i(t) - \mathbf{x}_c^j(t)) + 2r_c \|\mathbf{x}_{c0}^j - \mathbf{x}_{c0}^i\| &\leq 0 \\ \Leftrightarrow (\mathbf{x}_{c0}^i - \mathbf{x}_{c0}^j)^\top (\mathbf{x}_c^i(t) - \mathbf{x}_c^j(t)) &\geq 2r_c \|\mathbf{x}_{c0}^i - \mathbf{x}_{c0}^j\| \end{aligned} \quad (8)$$

By using Cauchy-Schwartz Inequality and (8), we have

$$\begin{aligned} \|\mathbf{x}_c^i(t) - \mathbf{x}_c^j(t)\| &\geq \frac{\|(\mathbf{x}_{c0}^i - \mathbf{x}_{c0}^j)^\top (\mathbf{x}_c^i(t) - \mathbf{x}_c^j(t))\|}{\|\mathbf{x}_{c0}^i - \mathbf{x}_{c0}^j\|} \\ &\geq \frac{2r_c \|\mathbf{x}_{c0}^i - \mathbf{x}_{c0}^j\|}{\|\mathbf{x}_{c0}^i - \mathbf{x}_{c0}^j\|} = 2r_c. \end{aligned} \quad (9)$$

Hence, we can conclude that $\mathcal{H}_s^{ij}(t)$ and $\mathcal{H}_s^{ji}(t)$ do not make inter-agent collisions. \square

B. Dynamic Inter-Visibility Cell

Similar to the Dynamic Buffered Voronoi Cell (DBVC), we define a Dynamic Inter-Visibility Cell (DIVC) that moves in accordance with the target's future trajectory and prevents inter-agent occlusion between the i -th and j -th agents. We design the DIVC by dividing the cases where an angle formed by the two *Lines-of-Sight*, $\mathcal{L}(\mathbf{x}_{c0}^i, \mathbf{x}_{q0})$ and $\mathcal{L}(\mathbf{x}_{c0}^j, \mathbf{x}_{q0})$, is either obtuse or acute.

1) *Obtuse Case*: Let V_i , V_j , and V_Q be points at \mathbf{x}_{c0}^i , \mathbf{x}_{c0}^j , and \mathbf{x}_{q0} , respectively. Then, let points V_{oi} and V_{oj} be the points

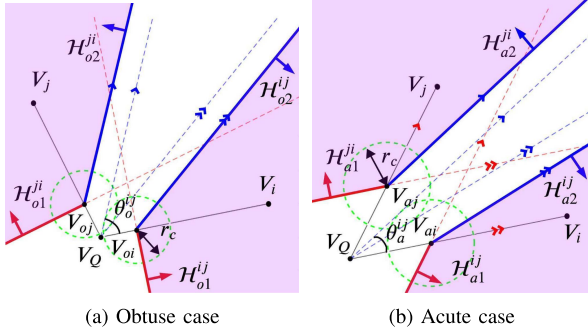


Fig. 2. DIVC formulation.

on line segments $\overline{V_i V_Q}$ and $\overline{V_j V_Q}$ that are each at a distance $\alpha_o^{ij} r_c$ ($\alpha_o^{ij} \geq 1$) from V_Q . We draw lines (red in Fig. 2(a)) that are perpendicular to $\overline{V_i V_Q}$ and $\overline{V_j V_Q}$, crossing V_{oi} and V_{oj} , and we call half-spaces made by the lines \mathcal{H}_{o1}^{ij} and \mathcal{H}_{o2}^{ij} , respectively. In addition, we draw the rays (blue in Fig. 2(a)) starting from V_{oi} and V_{oj} , which are parallel to the lines that pass through V_Q and are tangential to the balls, $\mathcal{B}(V_{oj}, r_c)$ and $\mathcal{B}(V_{oi}, r_c)$. Half-spaces made by the rays are represented as \mathcal{H}_{a1}^{ij} and \mathcal{H}_{a2}^{ij} . To satisfy the conditions that V_{oi} and V_{oj} are on $\overline{V_i V_Q}$ and $\overline{V_j V_Q}$, and the rays do not intersect, α_o^{ij} should be in the following range.

$$1 \leq \alpha_o^{ij} \leq \min \left(\frac{\min(\|\mathbf{x}_{cq0}^i\|, \|\mathbf{x}_{cq0}^j\|)}{r_c}, \sqrt{\frac{2}{1 - (\mathbf{n}_{cq0}^i)^\top \mathbf{n}_{cq0}^j}} \right)$$

where $\mathbf{x}_{cq0}^i = \mathbf{x}_{c0}^i - \mathbf{x}_{q0}$, $\mathbf{x}_{cq0}^j = \mathbf{x}_{c0}^j - \mathbf{x}_{q0}$,

$$\mathbf{n}_{cq0}^i = \mathbf{x}_{cq0}^i / \|\mathbf{x}_{cq0}^i\|, \quad \mathbf{n}_{cq0}^j = \mathbf{x}_{cq0}^j / \|\mathbf{x}_{cq0}^j\| \quad (10)$$

With the α_o^{ij} satisfying (10), for $\forall \mathbf{x}^i \in \mathcal{H}_{o1}^{ij} \cap \mathcal{H}_{o2}^{ij}$ and $\forall \mathbf{x}^j \in \mathcal{H}_{a1}^{ij} \cap \mathcal{H}_{a2}^{ij}$, the following inequalities are satisfied.

$$\begin{aligned} \min_{\epsilon \in [0,1]} \|\epsilon \mathbf{x}^i + (1-\epsilon) \mathbf{x}_{q0} - \mathbf{x}^j\| &> r_c, \\ \min_{\epsilon \in [0,1]} \|\epsilon \mathbf{x}^j + (1-\epsilon) \mathbf{x}_{q0} - \mathbf{x}^i\| &> r_c \end{aligned} \quad (11)$$

The above inequalities mean that the distances between neighboring agents and *Line-of-Sight* connecting the tracker and the target are always greater than the size of the trackers r_c ; therefore, the inter-occlusion constraints are satisfied. To make the inter-occlusion-free regions move with the target, we maintain the shapes of $\mathcal{H}_{o1}^{ij} \cap \mathcal{H}_{o2}^{ij}$ and $\mathcal{H}_{a1}^{ij} \cap \mathcal{H}_{a2}^{ij}$ and translate them by $\mathbf{x}_q(t) - \mathbf{x}_{q0}$, equivalent to the amount of target's movement. The moving half-spaces are represented as $\mathcal{H}_{o1}^{ij}(t)$, $\mathcal{H}_{o2}^{ij}(t)$, $\mathcal{H}_{a1}^{ij}(t)$, and $\mathcal{H}_{a2}^{ij}(t)$. Their mathematical expressions are simplified as follows, and $\mathcal{H}_{o1}^{ij}(t)$ and $\mathcal{H}_{o2}^{ij}(t)$ can be constructed by exchanging the indices i and j in (12).

$$\mathcal{H}_{o1}^{ij}(t) = \{\mathbf{x}(t) \in \mathbb{R}^2 | (\mathbf{n}_{cq0}^i)^\top (\mathbf{x}(t) - \mathbf{x}_q(t)) - \alpha_o^{ij} r_c \geq 0\}, \quad (12a)$$

$$\mathcal{H}_{o2}^{ij}(t) = \left\{ \mathbf{x}(t) \in \mathbb{R}^2 \mid \begin{bmatrix} z^{ij} \sin(\theta_{cq}^j + z^{ij} \theta_o^{ij}) \\ -z^{ij} \cos(\theta_{cq}^j + z^{ij} \theta_o^{ij}) \end{bmatrix}^\top (\mathbf{x}(t) \right.$$

$$\left. - \mathbf{x}_q(t) - \alpha_o^{ij} r_c \mathbf{n}_{cq0}^i \leq 0 \right\} \quad (12b)$$

where z^{ij} , θ_{cq}^i , θ_{cq}^j , and θ_o^{ij} are defined as follows.

$$z^{ij} = \text{sign}(\det([\mathbf{x}_{cq0}^j]^\top; [\mathbf{x}_{cq0}^i]^\top)),$$

$$\begin{bmatrix} \cos(\theta_{cq}^i) \\ \sin(\theta_{cq}^i) \end{bmatrix} = \mathbf{n}_{cq0}^i, \quad \begin{bmatrix} \cos(\theta_{cq}^j) \\ \sin(\theta_{cq}^j) \end{bmatrix} = \mathbf{n}_{cq0}^j,$$

$$\sin(\theta_o^{ij}) = (\alpha_o^{ij})^{-1}, \quad \cos(\theta_o^{ij}) = \sqrt{1 - \sin^2(\theta_o^{ij})} \quad (13)$$

2) *Acute Case*: Fig. 2(b) illustrates an acute case. First, we draw the lines that are parallel to $\overline{V_i V_Q}$ and $\overline{V_j V_Q}$ and apart by $\alpha_a^{ij} r_c$ ($\alpha_a^{ij} \geq 1$) from them. The half-spaces made by the lines are represented as \mathcal{H}_{a1}^{ij} and \mathcal{H}_{a2}^{ij} (red in Fig. 2(b)), and the intersections between the segments and lines are denoted as V_{aj} and V_{ai} , respectively. Then, we draw the rays (blue in Fig. 2(b)) starting from V_{ai} and V_{aj} , which are parallel to the lines that pass through V_Q and are tangential to the balls $\mathcal{B}(V_{aj}, r_c)$ and $\mathcal{B}(V_{ai}, r_c)$. Half-spaces divided by the rays are represented as \mathcal{H}_{o1}^{ij} and \mathcal{H}_{o2}^{ij} . To satisfy conditions that V_{ai} and V_{aj} are on $\overline{V_i V_Q}$ and $\overline{V_j V_Q}$, and the rays do not intersect, α_a^{ij} should satisfy the following range.

$$1 \leq \alpha_a^{ij} \leq \min \left(\frac{\|\det([\mathbf{x}_{cq0}^i]^\top; [\mathbf{x}_{cq0}^j]^\top)\|}{r_c \max(\|\mathbf{x}_{cq0}^i\|, \|\mathbf{x}_{cq0}^j\|)}, \sqrt{2(1 + (\mathbf{n}_{cq0}^i)^\top \mathbf{n}_{cq0}^j)} \right) \quad (14)$$

Similarly to the *Obtuse case*, $\forall \mathbf{x}^i \in \mathcal{H}_{a1}^{ij} \cap \mathcal{H}_{a2}^{ij}$ and $\forall \mathbf{x}^j \in \mathcal{H}_{o1}^{ij} \cap \mathcal{H}_{o2}^{ij}$ satisfy (11). The moving version of these half-spaces are denoted as $\mathcal{H}_{a1}^{ij}(t)$, $\mathcal{H}_{a2}^{ij}(t)$, $\mathcal{H}_{o1}^{ij}(t)$, and $\mathcal{H}_{o2}^{ij}(t)$ and mathematically formulated as follows.

$$\mathcal{H}_{a1}^{ij}(t) = \left\{ \mathbf{x}(t) \in \mathbb{R}^2 \mid z^{ij} \det([\mathbf{x}(t) - \mathbf{x}_q(t)]^\top; [\mathbf{x}_{cq0}^j]^\top) + \alpha_a^{ij} r_c \|\mathbf{x}_{cq0}^j\| \leq 0 \right\}, \quad (15a)$$

$$\mathcal{H}_{a2}^{ij}(t) = \left\{ \mathbf{x}(t) \in \mathbb{R}^2 \mid \begin{bmatrix} z^{ij} \sin(\theta_{cq}^j + z^{ij} \theta_o^{ij}) \\ -z^{ij} \cos(\theta_{cq}^j + z^{ij} \theta_o^{ij}) \end{bmatrix}^\top (\mathbf{x}(t) - \mathbf{x}_q(t) - \frac{\alpha_a^{ij} r_c \mathbf{n}_{cq0}^i}{\|\det([\mathbf{n}_{cq0}^i]^\top; [\mathbf{n}_{cq0}^j]^\top)\|)} \leq 0 \right\} \quad (15b)$$

where θ_o^{ij} are defined as follows.

$$\begin{aligned} \sin(\theta_o^{ij}) &= \frac{1}{\alpha_a^{ij}} \|\det([\mathbf{n}_{cq0}^i]^\top; [\mathbf{n}_{cq0}^j]^\top)\|, \\ \cos(\theta_o^{ij}) &= \sqrt{1 - \sin^2(\theta_o^{ij})} \end{aligned} \quad (16)$$

Lemma 2: The intersection of DBVC and DIVC is non-empty.

Proof: By moving the i -th agent by the same relative displacement as the target's movement: $\mathbf{x}_c^i(t) = \mathbf{x}_{c0}^i + (\mathbf{x}_q(t) - \mathbf{x}_{q0})$, both constraints $\mathbf{x}_c^i(t) \in \mathcal{H}_s^i(t)$ and $\mathbf{x}_c^i(t) \in \mathcal{H}_{\mu 1}^{ij}(t) \cap \mathcal{H}_{\mu 2}^{ij}(t)$, $\mu = o$ or a , are satisfied. \square

Lemma 2 indicates that the DBVC and DIVC offer feasible constraints to handle inter-agent collisions and occlusions.

V. TRACKING TRAJECTORY PLANNING

In this section, we formulate the tracking trajectory generation using Bernstein polynomial motion primitives. We begin by sampling a set of motion primitives and filtering out those that do not satisfy the given constraints. Next, we select the optimal trajectory. To enhance computational efficiency during both the feasibility check and trajectory selection processes, we fully exploit the convex hull property and the integral property of the Bernstein polynomial, respectively.

A. Primitive Sampling

Based on the i -th agent's current position \mathbf{x}_{c0}^i and velocity $\dot{\mathbf{x}}_{c0}^i$, we formulate the following optimal control problem to sample primitives, which can be solved in the closed form.

$$\begin{aligned} \min_{\mathbf{u}_c^i(t)} \quad & \frac{1}{T} \int_0^T \|\mathbf{u}_c^i(t)\|^2 dt \\ \text{s.t.} \quad & \dot{\mathbf{z}}_c^i(t) = F_c \mathbf{z}_c^i(t) + G_c \mathbf{u}_c^i(t), \\ & \mathbf{z}_c^i(t) = \begin{bmatrix} \mathbf{x}_c^i(t) \\ \dot{\mathbf{x}}_c^i(t) \end{bmatrix}, F_c = \begin{bmatrix} 0_2 & I_2 \\ 0_2 & 0_2 \end{bmatrix}, G_c = \begin{bmatrix} 0_2 \\ I_2 \end{bmatrix}, \\ & \mathbf{x}_c^i(0) = \mathbf{x}_{c0}^i, \dot{\mathbf{x}}_c^i(0) = \dot{\mathbf{x}}_{c0}^i, \mathbf{x}_c^i(T) = \mathbf{x}_{cf}^i \end{aligned} \quad (17)$$

0_2 is a 2×2 zero matrix, and I_2 is a 2×2 identity matrix. The \mathbf{x}_{cf}^i are terminal points, sampled around the target's terminal points $\mathbf{x}_q(T)$:

$$\begin{aligned} \mathbf{x}_{cf}^i &= \mathbf{x}_q(T) + [r_{cs}^i \cos \psi_{cs}^i, r_{cs}^i \sin \psi_{cs}^i]^\top, \\ r_{cs}^i &\sim U[r_{cs}^i, \bar{r}_{cs}^i], \psi_{cs}^i \sim U[\underline{\psi}_{cs}^i, \bar{\psi}_{cs}^i]. \end{aligned} \quad (18)$$

U represents the uniform distribution, and $(r_{cs}^i, \bar{r}_{cs}^i)$ and $(\underline{\psi}_{cs}^i, \bar{\psi}_{cs}^i)$ are pairs of the lower and upper bound of distribution of radius and azimuth, respectively.

B. Feasibility Check

1) *Collision Check*: To make the i -th agent safe, we check whether trajectories satisfy (19a): being confined in safe and visible corridors, and (19b)–(19d): avoiding collision with dynamic obstacles, target, and the other agents.

$$\Xi(\mathbf{C}_m^i) \oplus \mathcal{B}(\mathbf{0}, r_c) \subset S_m^i, \forall m = 1, \dots, M_i, \quad (19a)$$

$$\|\mathbf{x}_c^i(t) - \mathbf{x}_o^k(t)\|^2 - (r_c + r_o)^2 \geq 0, \forall k \in \mathcal{I}_o \quad (19b)$$

$$\|\mathbf{x}_c^i(t) - \mathbf{x}_q(t)\|^2 - (r_c + r_q)^2 \geq 0, \quad (19c)$$

$$\mathbf{x}_c^i(t) \in \bigcap_{j \in \mathcal{I}_c \setminus i} \mathcal{H}_s^{ij}(t), \quad (19d)$$

S_m^i is the m -th visible and safe corridors generated by [8], \mathbf{C}_m^i is the m -th Bernstein coefficients, which is split from \mathbf{C}^i . Ξ is a set of points in Bernstein coefficients. For (19a), we utilize convex hull property, and since the left-hand-side of (19b) and (19c) can be expressed as Bernstein polynomials, we select primitives with nonnegative coefficients for all bases. Similarly to (19a), we verify whether the control points of (19d) belong to the intersections of the affine spaces by using the convex hull property to determine if trajectories are located within DBVC

where inter-collision does not occur. The primitives that pass the tests in (19) satisfy (3).

2) *Visibility Check*: We check whether the primitives of the i -th agent satisfy (20a), (20b), and (20c) to avoid occlusion by static obstacles, dynamic obstacles, and the other agents, respectively, and consequently satisfy (4).

$$\Xi(\mathbf{C}_m^i) \subset S_m^i, \forall k = 1, \dots, M_i, \quad (20a)$$

$$\|\epsilon \mathbf{x}_c^i(t) + (1 - \epsilon) \mathbf{x}_q(t) - \mathbf{x}_o^k(t)\|^2 - r_o^2 \geq 0, \quad (20b)$$

$$\forall k \in \mathcal{I}_o, \forall \epsilon \in [0, 1],$$

$$\mathbf{x}_c^i(t) \in \bigcap_{j \in \mathcal{I}_c \setminus i} \left(\mathcal{H}_{\mu 1}^{ij}(t) \cap \mathcal{H}_{\mu 2}^{ij}(t) \right), \mu = o \text{ or } a \quad (20c)$$

(20a) is a necessary condition for (19a). The convex hull property is used to check (20c), the DIVC constraints. (20b) means that *Line-of-Sight* does not collide with dynamic obstacles, and the left-hand side of (20b) can be reformulated as follows.

$$\begin{aligned} \text{L.H.S of (20b)} &= \epsilon^2 \sigma_1(t) + 2\epsilon(1 - \epsilon) \sigma_2(t) + (1 - \epsilon)^2 \sigma_3(t) \\ &\geq \epsilon^2 \sigma'_1(t) + 2\epsilon(1 - \epsilon) \sigma'_2(t) + (1 - \epsilon)^2 \sigma'_3(t), \end{aligned}$$

$$\text{where } \sigma_1(t) = \|\mathbf{x}_c^i(t) - \mathbf{x}_o^k(t)\|^2 - r_o^2,$$

$$\sigma_2(t) = (\mathbf{x}_c(t) - \mathbf{x}_o^k(t))^\top (\mathbf{x}_q(t) - \mathbf{x}_o^k(t)) - r_o^2,$$

$$\sigma_3(t) = \|\mathbf{x}_q(t) - \mathbf{x}_o^k(t)\|^2 - r_o^2,$$

$$\sigma'_1(t) = \sigma_1(t) - (r_o + r_c)^2 + r_o^2,$$

$$\sigma'_2(t) = \sigma_2(t) + (r_o + \min(r_q, r_c))^2 - r_o^2,$$

$$\sigma'_3(t) = \sigma_3(t) - (r_o + r_q)^2 + r_o^2 \quad (21)$$

Since ϵ^2 , $2\epsilon(1 - \epsilon)$, and $(1 - \epsilon)^2$ are nonnegative for $\forall \epsilon \in [0, 1]$, (20b) holds if the terms multiplied by them are also nonnegative. This paper examines whether $\sigma'_1(t)$, $\sigma'_2(t)$, $\sigma'_3(t) \geq 0$ hold, whereas our previous work [8] checks whether $\sigma_1(t)$, $\sigma_2(t)$, $\sigma_3(t) \geq 0$ hold. By tightening the conditions for $\sigma_1(t)$ and $\sigma_3(t)$, the proposed work relaxes condition for $\sigma_2(t)$. Fig. 3 shows that the proposed check method is less conservative compared to [8]. By employing the less conservative feasibility check, our planner can find more feasible motions.

3) *Distance Check*: To keep suitable distance from the targets, the following conditions are established.

$$\|\mathbf{x}_c^i(t) - \mathbf{x}_q(t)\|^2 - d_{\min}^2 \geq 0, d_{\max}^2 - \|\mathbf{x}_c^i(t) - \mathbf{x}_q(t)\|^2 \geq 0 \quad (22)$$

We set $d_{\min} \geq r_q + r_c$ to avoid collision with the target, and the primitives that pass the tests in (22) satisfy (5).

4) *Dynamical Limit Check*: To ensure the drone does not exceed dynamic limits (6), we verify whether the primitives satisfy the following inequalities. (23b) is calculated under the assumption that the agents directly head toward the target.

$$v_{\max}^2 - \|\dot{\mathbf{x}}_c^i(t)\|^2 \geq 0, a_{\max}^2 - \|\ddot{\mathbf{x}}_c^i(t)\|^2 \geq 0, \quad (23a)$$

$$\dot{\psi}_c(t) = \frac{\det([\mathbf{x}_q(t) - \mathbf{x}_c^i(t)]^\top; [\dot{\mathbf{x}}_q(t) - \dot{\mathbf{x}}_c^i(t)]^\top)}{\|\mathbf{x}_q(t) - \mathbf{x}_c^i(t)\|^2}, \quad (23b)$$

$$-\dot{\psi}_{\max} \leq \dot{\psi}_c(t) \leq \dot{\psi}_{\max} \quad (23c)$$

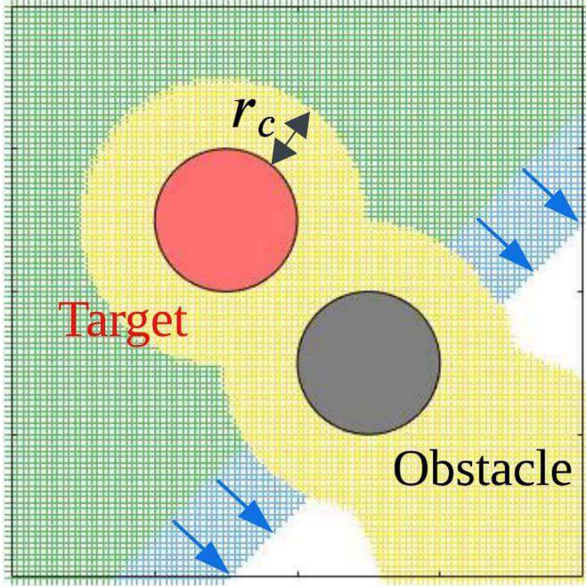


Fig. 3. Comparison of the feasibility checks. Yellow: an area where the agent either cannot see the center of a target or collides with the target or an obstacle. Green: an area where [8] considers the collision- and occlusion-free area. Blue: an expanded feasible area by the proposed method.

C. Best Trajectory Selection

Among the primitives that pass (19)–(23), we select the best tracking trajectory. We evaluate the following cost, which consists of a penalty term for jerkiness of trajectory and a cost term to maintain appropriate distance from the target.

$$\min J_1 + J_2, \quad (24a)$$

$$J_1 = w_j \int_0^T \|\ddot{\mathbf{x}}_c^i(t)\|_2^2 dt, \quad (24b)$$

$$J_2 = \sum_{i=1}^{N_q} \int_0^T (\|\mathbf{x}_c^i(t) - \mathbf{x}_q(t)\|_2^2 - d_{des,i}^2)^2 dt \quad (24c)$$

w_j is a weight factor, and $d_{des,i}$ is the desired distance between the target and the i -th agents, which is set to $\frac{1}{2}(r_{cs}^i + \bar{r}_{cs}^i)$ in the validation.

VI. VALIDATIONS

In this section, the proposed method is validated through various target-tracking settings. We measure the distance between the drone and environments, defined as (25a), the distance among agents (25b), and the distance between the target and drone, defined as (25c), to evaluate drone safety.

$$\chi_1(t) = \min_{i \in \mathcal{I}_c} \min_{\substack{\mathbf{x}(t) \in \mathcal{C}_i(t) \\ \mathbf{y}(t) \in \mathcal{O}(t)}} \|\mathbf{x}(t) - \mathbf{y}(t)\|, \quad (25a)$$

$$\chi_2(t) = \min_{i \in \mathcal{I}_c} \min_{\substack{\mathbf{x}(t) \in \mathcal{C}_i(t) \\ \mathbf{y}(t) \in \mathcal{C}^i(t)}} \|\mathbf{x}(t) - \mathbf{y}(t)\|, \quad (25b)$$

$$\chi_3(t) = \min_{i \in \mathcal{I}_c} \min_{\substack{\mathbf{x}(t) \in \mathcal{C}_i(t) \\ \mathbf{y}(t) \in \mathcal{T}(t)}} \|\mathbf{x}(t) - \mathbf{y}(t)\| \quad (25c)$$

Also, we measure the distance between the *Lines-of-Sight* and environments (26a) and the distance between the *Lines-of-Sight*

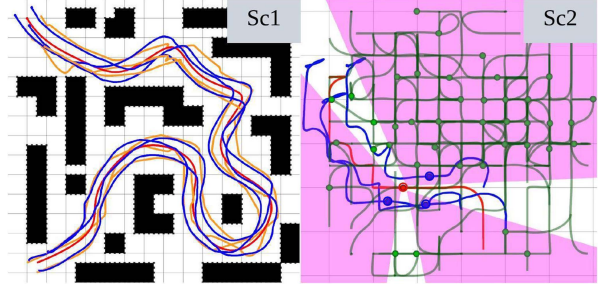


Fig. 4. **Left:** Three trackers (blue: ours, orange: [20]) follow the red target among black static obstacles. **Right:** Three blue trackers follow the red target among forty green dynamic obstacles. Magenta regions represent DIVCs.

and the other agents (26b) to assess the target visibility.

$$\phi_1(t) = \min_{i \in \mathcal{I}_c} \min_{\substack{\mathbf{x}(t) \in \mathcal{L}(\mathbf{x}_c^i(t), \mathbf{x}_q(t)) \\ \mathbf{y}(t) \in \mathcal{O}(t)}} \|\mathbf{x}(t) - \mathbf{y}(t)\|, \quad (26a)$$

$$\phi_2(t) = \min_{i \in \mathcal{I}_c} \min_{\substack{\mathbf{x}(t) \in \mathcal{L}(\mathbf{x}_c^i(t), \mathbf{x}_q(t)) \\ \mathbf{y}(t) \in \mathcal{C}^i(t)}} \|\mathbf{x}(t) - \mathbf{y}(t)\| \quad (26b)$$

Using the above performance metrics, we validate the operability of the proposed planner under two environmental conditions: 1) unstructured static obstacles, and 2) dynamic obstacles. Through simulations and hardware experiments, we show successful target tracking. Table II shows the reported performance, and the details of the tests are explained in the following subsections. In addition, we validate the effectiveness of the proposed method for multi-agent tracking through a comparative analysis and show applicability to 3D scenarios.

For the target trajectory prediction, we employ a method in [8], applicable to both test conditions: environments with unstructured-but-static obstacles and dynamic-but-structured obstacles. The radii of the targets, trackers, and dynamic obstacles are set to 7.5 cm, matching the size of Crazyflie quadrotors. Also, since the test environments are either narrow or crowded, we set the sampling range of the tracking distance ($\underline{r}_{cs}, \bar{r}_{cs}$) to (0.3, 0.6) [m].

In the tests, we use computers with an Intel i7 12th-gen CPU and 16 GB RAM. The number of sampled primitives is set to 1000, and four threads are used for parallel computation. The reported computation time in all tracking scenarios is less than 10 milliseconds.

A. Simulations

Scenario 1. (Unstructured Environment): We compare the proposed planner with the state-of-the-art planner [20] in an environment with various shapes of static obstacles, as shown in Fig. 4. As Table II shows, the trackers controlled by the baseline [20] experience inter-agent occlusions and collisions with the target several times, despite finely tuned parameters to adapt to the narrow tracking environment. In contrast, our planner successfully tracks the target without collision and occlusion by thoroughly checking safety and visibility constraints, including the tracking distance (d_{\min}, d_{\max}) and DIVC constraints.

Scenario 2. (Dynamic Environment): We test our approach in a dense moving-obstacle environment. The target moves in a 6×6 m² space with 40 obstacles. The target and obstacles move

TABLE II
REPORTED VALIDATION PERFORMANCE

Scenarios	Environments	Planner	Safety Metrics [m]			Visibility Metrics [m]	
			Obstacle (25a)	Agent (25b)	Target (25c)	Obstacle (26a)	Agent (26b)
Sc1	unstructured	proposed	0.609/0.132	0.561/0.727	0.215/0.359	0.202/0.609	0.290/0.727
		baseline [20]	0.098/0.567	0.225/0.639	0.000/0.399	0.173/0.567	0.000/0.639
Sc2	dynamic	proposed	0.061/0.327	0.480/0.666	0.195/0.319	0.136/0.327	0.270/0.666
Sc3	unstructured	proposed	0.013/0.480	0.554/0.683	0.148/0.346	0.089/0.480	0.223/0.683
Sc4	dynamic	proposed	0.065/1.201	0.421/0.711	0.240/0.353	0.140/1.201	0.315/0.711
Sc5	unstructured	proposed	0.413/1.014	0.467/0.848	0.275/0.654	0.401/1.014	0.319/0.848
Sc6	dynamic	proposed	0.135/0.397	0.378/0.431	0.096/0.217	0.210/0.397	0.171/0.431

0 indicates a collision or occlusion. The values for the above metrics indicate the minimum/mean performance.

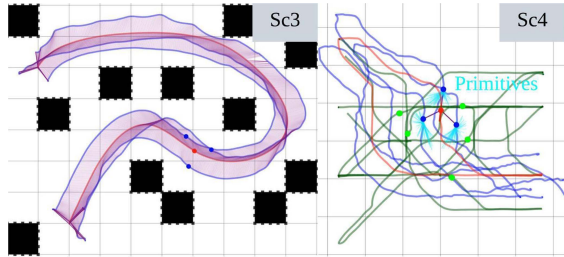


Fig. 5. Target tracking experiments. The total flight paths of Crazyflies serving as trackers (blue), target (red), and dynamic obstacles (green), along with boxes (black), are plotted in a top-down view. The purple areas are the accumulated histories of the *Lines-of-sight* connecting the target and the trackers.

around with the maximum speed 1.0 m/s. The right side of Fig. 4 shows a flight history.

B. Hardware Experiments

For hardware demonstration, we use Crazyflie 2.1 quadrotors. One serves as the target, three serve as the trackers, and the remainder act as dynamic obstacles. The target and dynamic obstacles are both controlled by a single Intel NUC, and they follow a pre-calculated path generated by [21]. On the other hand, to implement distributed calculation, each tracker is connected to a separate mini PC.

Scenario 3. (Unstructured Environment): The target moves in a $10 \times 7 \text{ m}^2$ space with twelve cube-shaped obstacles. The top image in Fig. 1 is a snapshot of the flight, and the left side of Fig. 5 shows the reported results.

Scenario 4. (Dynamic Environment): The target moves in a $7 \times 7 \text{ m}^2$ space with five moving obstacles. The bottom image in Fig. 1 is a snapshot of the mission, and the right side of Fig. 5 summarizes the flight test.

C. Comparison Analysis

The performance of the proposed approach is investigated through two tests. The tests evaluate the effectiveness of the DBVC, the DIVC, and improved feasibility check methods. In each test, we define success as the absence of occlusion or collision until all objects come to a stop. We conduct 1000 tests for each tracking setup and measure the success rate.

First, we validate the effectiveness of the DBVC and the DIVC by comparing the success rates with cases where these two cells are not applied and where static versions of the cells are used. Specifically, when the cells are not used, each tracker treats the neighbor trackers as dynamic obstacles. In the case of using static cells, we use BVC, $\mathcal{H}_{\mu 1}^{ij}$, and $\mathcal{H}_{\mu 2}^{ij}$, $\mu = o$ or a , instead of $\mathcal{H}_s^{ij}(t)$, $\mathcal{H}_{\mu 1}^{ij}(t)$, and $\mathcal{H}_{\mu 2}^{ij}(t)$.

TABLE III
SUCCESS RATE IN AN EMPTY SPACE WHEN USING (NO/STATIC/DYNAMIC) CELLS [%]

N_c	3	4	5
Short [0.4, 1.2] [m]	95.6/79.4/ 99.5	44.7/44.0/ 99.3	1.9/14.6/ 96.3
Medium [0.8, 1.6] [m]	97.9/ 99.5/99.5	74.2/96.6/ 99.4	16.5/67.6/ 98.2
Long [1.2, 2.0] [m]	98.1/ 99.6/99.6	88.0/98.4/ 99.5	48.5/96.6/ 98.9

The ranges in the first column mean the sampling distance $[L_{cs}, \bar{r}_{cs}]$. The dynamic cells denote the DBVC and the DIVC.

We test these setups in an obstacle-free space, where the target moves around at the maximum speed 1.0 m/s for an average of 30 seconds. We varied the number of trackers and distance to the targets, and Table III summarizes the results. As the number of targets increases and the tracking distance decreases, the difficulty of tracking increases. Without using cells, the level of interference due to the movement of neighboring agents increases as the tracking distance shortens, leading to a higher failure rate. In contrast, the spatially separated characteristics of the DBVC and the DIVC implicitly make a balanced formation, resulting in a higher success rate. Also, static cells can conflict with other constraints, particularly the distance between the target and the trackers. However, the dynamic properties of the DBVC and the DIVC maintain consistent tracking distance, which results in a higher success rate. Moreover, a higher number of trackers narrows the cells, bringing the agents closer to the cell boundaries. Such conditions make it difficult to satisfy all the constraints. However, even in the short tracking distance, the planner using the DBVC and the DIVC achieves the highest success rate because the dynamic properties result in fewer constraint violations under such conditions.

Second, we conduct a benchmark test to validate the superiority of our planner in a dynamic obstacle environment. We compare our method with a noncooperative approach and a conservative approach. For the noncooperative approach, we use the BPMP-Tracker [8]. Since the method in [8] is designed for single-agent tracking, we adapt the problem setting so that each tracker treats neighboring agents as obstacles. The conservative approach utilizes the DBVC and the DIVC but applies the conservative feasibility check methods from [8].

The target and obstacles move around for an average of 40 seconds in a $6 \times 6 \text{ m}^2$ space at the maximum speed of 0.5 m/s. We measure the success rate while varying the number of obstacles. Our approach generates tracking trajectories in a cooperative manner, whereas the noncooperative setup causes consistent interference among trackers. This results in a higher success rate for the proposed planner for all tracking conditions, as shown in Table IV. Moreover, our approach discovers more feasible motions than the conservative approach in tight conditions, leading to a higher success rate.

TABLE IV
SUCCESS RATE COMPARISON IN DYNAMIC ENVIRONMENTS [%]

#Agent (N_c)	Planner	# Obstacle (N_o)		
		5	10	20
2	noncooperative	99.2	98.7	95.2
	conservative	99.9	99.9	99.2
	proposed	99.9	99.9	99.4
3	noncooperative	70.1	68.4	58.1
	conservative	96.9	93.5	88.3
	proposed	97.6	94.6	90.1
4	noncooperative	1.01	0.93	0.72
	conservative	89.7	79.6	68.5
	proposed	91.2	82.1	71.2

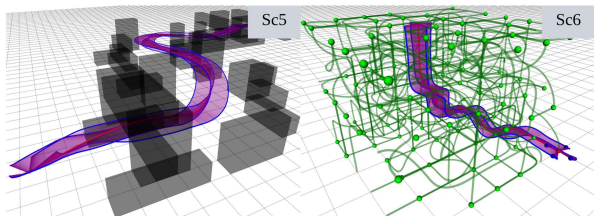


Fig. 6. Tracking in 3D spaces. Reported paths of trackers (blue), target (red), and dynamic obstacles (green), with static obstacles (black) and *Line-of-sights* (purple) connecting the trackers and the target.

As the number of trackers increases, the cells become smaller, making it difficult to find a feasible motion within those reduced areas in the presence of adjacent dynamic obstacles. In future work, we plan to design inter-occlusion- and inter-collision-free cells that not only translate but also change shape over time, to enhance robustness against the interference of dynamic obstacles.

D. 3D Extensions

The proposed planner, initially designed for 2D, is extended to 3D environments. Allowing trackers to move along the z-axis enables a wider range of feasible motions and improves tracking performance in challenging scenarios. The following tests show its effectiveness in 3D settings.

Scenario 5. (3D Unstructured Environment): The aerial target navigates through height-varied structures and tunnel-like passages while adjusting its altitude within the range of 1 to 2 meters, with the maximum speed of 2.19 m/s. The left side of Fig. 6 shows the paths of the target and the four trackers.

Scenario 6. (3D Dynamic Environment): The aerial target flies in a $6 \times 6 \times 4$ m³ space with 100 moving airborne obstacles, with the maximum speed of 1.0 m/s. The right side of Fig. 6 illustrates the results of tracking using four trackers.

VII. CONCLUSION

We presented a distributed multi-agent trajectory generation method for aerial tracking, which prevents both occlusion and collision caused by obstacles. DBVC and DIVC were constructed by dividing the space into multiple cells to avoid inter-agent collision and inter-agent occlusion, respectively. Since both cells are designed based on the shared current positions of agents and the predicted target's trajectory, it enables distributed planning. We achieved fast computation by combining the DBVC and the DIVC with the Bernstein-polynomial-motion-primitive-based trajectory planner and validated the operability

of our planner through tests under various tracking conditions. Lastly, we confirmed that the proposed method achieved a higher success rate in tracking missions in environments with dynamic obstacles, outperforming the state-of-the-art methods.

REFERENCES

- [1] A. Bucker, R. Bonatti, and S. Scherer, "Do you see what i see? coordinating multiple aerial cameras for robot cinematography," in *Proc. 2021 IEEE Int. Conf. Robot. Automat.*, 2021, pp. 7972–7979.
- [2] A. Alcántara, J. Capitán, R. Cunha, and A. Ollero, "Optimal trajectory planning for cinematography with multiple unmanned aerial vehicles," *Robot. Auton. Syst.*, vol. 140, 2021, Art. no. 103778.
- [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 Robot. Automat. Lett.*, vol. 6, no. 4, pp. 7580–7587, Oct. 2021.
- [4] R. Tallamraju et al., "Active perception based formation control for multiple aerial vehicles," *IEEE Robot. Automat. Lett.*, vol. 4, no. 4, pp. 4491–4498, Oct. 2019.
- [5] C. Ho, A. Jong, H. Freeman, R. Rao, R. Bonatti, and S. Scherer, "3D Human reconstruction in the wild with collaborative aerial cameras," in *Proc. 2021 IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2021, pp. 5263–5269.
- [6] S. Zhuge et al., "Markerless motion capture for humans through a multi-UAV system," *IEEE Trans. Instrum. Meas.*, vol. 73, 2023, Art. no. 5003112.
- [7] D. Zhou, Z. Wang, S. Bandyopadhyay, and M. Schwager, "Fast, on-line collision avoidance for dynamic vehicles using buffered Voronoi cells," *IEEE Robot. Automat. Lett.*, vol. 2, no. 2, pp. 1047–1054, Apr. 2017.
- [8] Y. Lee, J. Park, B. Jeon, S. Jung, and H. J. Kim, "BPMP-Tracker: A versatile aerial target tracker using Bernstein polynomial motion primitives," *IEEE Robot. Automat. Lett.*, vol. 9, no. 12, pp. 10938–10945, Dec. 2024.
- [9] J. Ji, N. Pan, C. Xu, and F. Gao, "Elastic tracker: A spatio-temporal trajectory planner for flexible aerial tracking," in *Proc. 2022 Int. Conf. Robot. Automat.*, 2022, pp. 47–53.
- [10] Z. Zhang, Y. Zhong, J. Guo, Q. Wang, C. Xu, and F. Gao, "Auto filmer: Autonomous aerial videography under human interaction," *IEEE Robot. Automat. Lett.*, vol. 8, no. 2, pp. 784–791, Feb. 2023.
- [11] B. Jeon, Y. Lee, and H. J. Kim, "Integrated motion planner for real-time aerial videography with a drone in a dense environment," in *Proc. 2020 IEEE Int. Conf. Robot. Automat.*, 2020, pp. 1243–1249.
- [12] 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 Robot. Automat. Lett.*, vol. 2, no. 3, pp. 1696–1703, Jul. 2017.
- [13] H. Masnavi, V. K. Adajania, K. Kruusamäe, and A. K. Singh, "Real-time multi-convex model predictive control for occlusion-free target tracking with quadrotors," *IEEE Access*, vol. 10, pp. 29009–29031, 2022.
- [14] Y. Lee, J. Park, S. Jung, B. F. Jeon, D. Oh, and H. J. Kim, "QP chaser: Polynomial trajectory generation for autonomous aerial tracking," 2023, *arXiv:2302.14273*.
- [15] T. Nägeli, L. Meier, A. Domahidi, J. Alonso-Mora, and O. Hilliges, "Real-time planning for automated multi-view drone cinematography," *ACM Trans. Graph.*, vol. 36, no. 4, pp. 1–10, 2017.
- [16] R. Bonatti, Y. Zhang, S. Choudhury, W. Wang, and S. A. Scherer, "Autonomous drone cinematographer: Using artistic principles to create smooth, safe, occlusion-free trajectories for aerial filming," in *Proc. Int. Symp. Exp. Robot.*, 2018, pp. 119–129. [Online]. Available: <https://api.semanticscholar.org/CorpusID52125807>
- [17] B. Penin, P. R. Giordano, and F. Chaumette, "Vision-based reactive planning for aggressive target tracking while avoiding collisions and occlusions," *IEEE Robot. Automat. Lett.*, vol. 3, no. 4, pp. 3725–3732, Oct. 2018.
- [18] L. Zhou and V. Kumar, "Robust multi-robot active target tracking against sensing and communication attacks," *IEEE Trans. Robot.*, vol. 39, no. 3, pp. 1768–1780, Jun. 2023.
- [19] L. Zhou, V. Tzoumas, G. J. Pappas, and P. Tokekar, "Resilient active target tracking with multiple robots," *IEEE Robot. Automat. Lett.*, vol. 4, no. 1, pp. 129–136, Jan. 2019.
- [20] X. Zhou et al., "Swarm of micro flying robots in the wild," *Sci. Robot.*, vol. 7, no. 66, 2022, Art. no. eabm5954.
- [21] J. Park, Y. Lee, I. Jang, and H. J. Kim, "DLSC: Distributed multi-agent trajectory planning in maze-like dynamic environments using linear safe corridor," *IEEE Trans. Robot.*, vol. 39, no. 5, pp. 3739–3758, Oct. 2023.