2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

Continuous Robotic Tracking of Dynamic Targets in Complex Environments Based on Detectability

Zhihao Wang, Shixing Huang, Minghang Li, Junyuan Ouyang, Yu Wang, Haoyao Chen*

***Abstract*— Target tracking is a fundamental task in the domain of robotics. The effectiveness of target tracking hinges upon various factors, such as tracking distance, occlusions, collision avoidance, etc. However, few existing works can simultaneously tackle these considerations of tracking single and multiple targets in complex environments. In this study, the interaction mechanism of target tracking between the robot, the environment and the targets is analyzed, and a general measure named detectability is introduced to correlate the tracking performance for guiding robotic motion planning. Based on the detectability measure, the robotic motion planning framework based on Model Predictive Control (MPC) is proposed to achieve continuous and robust tracking of single, two and three targets in complex environments. Simulations and experiments are performed and verify the performances of our method better than the state-of-the-art methods.**

I. Introduction

Tracking of moving objects by robots has attracted increasing attention in various domains, such as security monitoring [1] and post-disaster rescue [2]. Tracking of multiple targets in an aggregated formation is common in these domains [3]. For example, two firefighters are performing a rescue task together; a robot is tasked with trailing behind them to carry supplies, and an additional task is to monitor the firefighters' status for safety guarantee. Thus, the continuous observation for each target in the formation is essential for target following and monitoring because lost detections may lead to the failure of target tracking [4].

From the vantage point of passive perception, the mainstream tracking-by-detection framework is widely used to infer and reconstruct the lost information for continuous detection using historical motion data [5], [6]. The literature [7] on person-following exploited the well-detected joints of a partially occluded person to estimate the person's location. These methods [5]–[7] all managed short-term or partial occlusions; however, long-term or complete occlusion remains an impediment to the success rate of target tracking [8]. Active motion planning of robots exhibits an efficient solution to achieve continuous observation in target tracking. The robots plan motion to avoid occlusion and ensure continuous target visibility within the sensor's field of view (FOV) [9]. The active planning-based solutions rely on two points: 1) how to model a measure of positive correlation with the tracking performance, and 2) how to plan the robotic motion based on the measure.

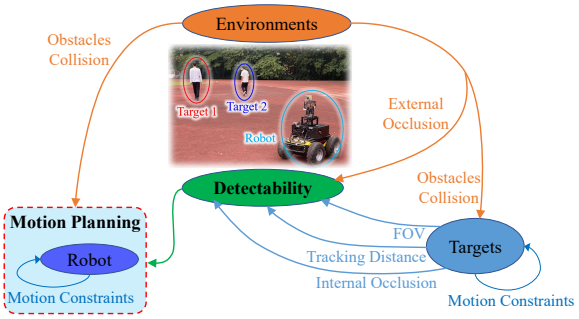


Fig. 1. Demonstration of the interaction mechanism between the robot, the environments, and the target in dynamic targets tracking task.

The tracking performance depends on many factors, such as the environment, the relative geometry between the robot and the targets [10], and so on. The tracking distance, the degree of external occlusion caused by a target behind obstacles, and the FOV constraint were commonly used to evaluate the tracking performance for single-target tracking [11]–[13]. Meanwhile, the degree of internal occlusion caused by a target behind other targets is essential for evaluating the tracking performance in multi-target tracking [3]. However, evaluating of the above metrics are based on a voxel map, such as Euclidean Signed Distance Fields (ESDF) [11], [12] or visibility score field [3], while constructing the voxel map is time-consuming in dynamic environments. In another way, the tracking performance was evaluated by using the relative geometry between the robot and the targets based on simplifying the dynamic persons into ellipsoid obstacles [14]. However, strong assumptions on the obstacle shape limited its applicability primarily to artificial or indoor structural environments [15].

Figure 1 shows the interaction mechanism of the target tracking process between the robot, the environments, and the targets. The robotic motion for target tracking is constrained by the complex environments, the moving targets, and the robotic kinematics and dynamic model. The environment includes many obstacles that cause collisions with the robot and external occlusions with the targets. The relative geometry between the robot and the targets, the sensor configuration, and the target detection algorithm also affect the tracking performance. Therefore, we introduce a measure named detectability to correlate the target tracking performance by a continuous and differentiable function. The detectability measure considers the tracking distance,

This work was supported in part by the National Natural Science Foundation of China (Grant No.U21A20119 and No.U1713206), and in part by the Shenzhen Science and Innovation Committee (Grant No.JCYJ20200109113412326 and No.JCYJ2021032412040003) . (Corresponding author: Haoyao Chen.)

Z.H. Wang, S.X. Huang, M.H. Li, J.Y. Ouyang, Y. Wang, and H.Y. Chen* are with the School of Mechanical Engineering and Automation, Harbin Institute of Technology Shenzhen, P.R. China, e-mail: hychen5@hit.edu.cn.

the external occlusion by obstacles, the internal occlusion by other targets, and the sensor FOV constraint. Besides, an obstacle modeling method is developed to represent complex environments with a series of simplified cylinders for calculating the detectability measure without the time-consuming process of updating the voxel map.

For the robotic motion planning for target tracking, trajectory optimization is the popular approach [3], [16] to ensure target tracking performance and collision avoidance. However, these trajectory optimization-based methods relied on a graph search algorithm to find an appropriate topology trajectory previously and subsequently need a controller to follow the optimized trajectory. MPC is another solution for optimizing the control commands directly to track targets [4], [17]. The existing methods based on MPC are designed for specific tasks, such as tracking only a single target in a simple and structural environment [4], [17]. To this end, using the detectability measure, a general framework based on MPC is developed to track single and multiple targets in complex environments. This framework guarantees the scalability of target-tracking task for different scenarios because the detectability shields the coupling relationship between the motion planner and the task requirements. For instance, our approach only changes the detectability function by adding the internal occlusion term when the task shifts from single-target tracking to two-target tracking.

This paper aims to address the above challenges in robotic target-tracking, i.e., simultaneously ensuring continuous target visibility and collision avoidance with limited sensors FOV in complex environments. The main contributions are twofold.

First, for continuous tracking of dynamic targets, the interaction mechanism between the robot, the environment, and the targets is analyzed, and the detectability measure considering tracking distance, occlusions, and FOV constraint is formulated to correlate the tracking performance for guiding robotic motion planning. The detectability measure is a continuous and differentiable function for efficient computation.

Second, based on the detectability measure, a robotic motion planning framework is proposed to achieve continuous and robust tracking of single, two, and three targets in complex environments. This framework based on MPC directly optimizes the control commands for tracking targets safely, making the framework more general for tracking different numbers of targets in various scenarios.

II. Method

The framework of the target tracking approach is given in Fig. 2; it mainly includes *Target Perception*, *Detectability Modeling*, and *Robotic Motion Planning* for target tracking. Based on our previous work [18], *Target Perception* completes the detection of targets and provides predicted trajectories $P_{pre} = P_{pre}^{t+1}, P_{pre}^{t+2}, ..., P_{pre}^{t+L}$ to compute the detectability for robotic motion planning. Dynamic SLAM performs robotic pose estimation with robustness by removing the points of dynamic targets. The estimated robotic pose

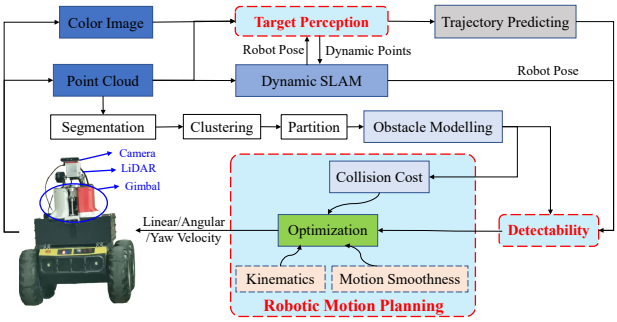


Fig. 2. System overview of continuous robotic tracking of dynamic targets.

is also used to help the target location estimation in target perception. Subsequently, we formulate a detectability function to correlate the tracking performance in *Detectability Modeling* module. Based on the idea of our previous work [19] for active horizontal sliding window-based planning, an MPC-based *Robotic Motion Planning* is formulated to tackle target tracking by predicting the target states over a finite horizon. After solving the optimization problem of MPC, the control commands of the robotic base and the gimbal are obtained and sent to the underlying controller to drive the robotic base and gimbal to track the targets.

A. Detectability Modeling

As indicated in Fig. 1, we consider three crucial indicators, including track distance, occlusions, and FOV constraint, to formulate the detectability measure. The measure relates to the performance of the robot's sensors in observing the targets from a viewpoint (including the position and orientation). Mathematically, the detectability function f_{det} is defined as follows:

$$f_{det}(\boldsymbol{x}_{t+1}, P_{pre}^{t+1}) = \lambda_d \cdot f_{dis} + \lambda_o \cdot f_{occ} + \lambda_f \cdot f_{fov} \quad (1)$$

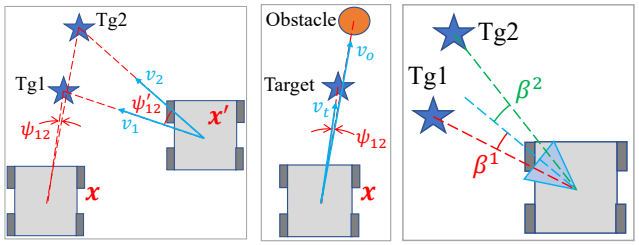
where $\boldsymbol{x_{t+1}}$ denotes the next viewpoint of the sensor; f_{dis}, f_{occ}, and f_{fov} denote the track distance term, occlusion term, and FOV deviation term, respectively; λ_d, λ_o, and λ_f are weighted coefficients being set empirically.

1) Track Distance: The track distance d^{trk} denotes the Euclidean distance between the sensor's position in $\boldsymbol{x}_{t+1}$ and the target's P_{pre}^{t+1}. d^{ref} denotes a reference value with the best observation distance empirically set according to the sensor configuration and target detection algorithm. The difference between the tracking distance d^{trk} and the preset reference distance d^{ref} indicates the detectability of the target. Thus, the track distance term f_{dis} is defined as:

$$f_{dis}(\boldsymbol{x}_{t+1}, P_{pre}^{t+1}) = e^{-(d^{trk} - d^{ref})^2} \quad (2)$$

2) Occlusion: If a target is occluded by other targets or obstacles, the information of the occluded targets is lost, leading the target tracking task to fail. Thus, the occlusion is closely related to the detectability. And the occlusion is divided into two categories: internal occlusion between targets and external occlusion between targets and obstacles.

Internal occlusion occurs when targets obstruct each other's visibility. The observation angle is introduced to



(a) (b) (c)

Fig. 3. Demonstration of detectability modeling. (a) internal occlusion between two targets; (b) fake external occlusion between the target and the obstacle; (c) FOV deviation.

represent the internal occlusion, as illustrated in Fig. 3(a). When the robot is at $\boldsymbol{x}$, the observation angle ψ_{12} formed between the robot, target 1, and target 2 tends to be 0. At this time, target 2 is occluded by target 1 and cannot be detected no matter how the sensor system rotates. When the robot is at $\boldsymbol{x}'$, the observation angle between the robot and the two targets ψ'_{12} is large, and both target 1 and target 2 can be observed. Therefore, the observation angle between the robot and the two targets is used to evaluate the inner occlusion. The definition is given as follows:

$$f_{occ}^{in}(\boldsymbol{x}_{t+1}, P_{pre}^{t+1}) = \psi_{12}^2 \tag{3}$$

where $f_{occ}^{in}(\cdot)$ indicates that a large observation angle brings higher detectability. The observation angle ψ_{12} is defined as:

$$\psi_{12} = \arccos\left(v_1 \cdot v_2 / \|v_1\| \cdot \|v_2\|\right) \tag{4}$$

where v_1 and v_2 are the unit direction vectors of the robot pointing to target 1 and target 2, respectively.

When the number of targets increases to N, the occlusion term is defined as follows:

$$f_{occ}^{in}(\boldsymbol{x}_{t+1}, P_{pre}^{t+1}) = \sum_{i=1}^{N} \sum_{j=i+1}^{N} \psi_{ij}^2 \tag{5}$$

where ψ_{ij} denotes the observation angle between the robot and two targets i and j. Notably, internal occlusion is inherently inevitable when tracking a large number of targets. While other high-level strategies, such as just tracking the leader target or tracking targets as many as possible, are needed to address this issue, they are beyond the scope of our current work.

External occlusion occurs when the target is behind the obstacle. Only using the observation angle is not compatible with calculating the external occlusion. As illustrated in Fig. 3(b), the target in front of the obstacle is a fake occlusion case even though the observation angle ψ_{12} is near 0. Thus, we define the external occlusion term as follows:

$$f_{occ}^{ex}(\boldsymbol{x}_{t+1}, P_{pre}^{t+1}) = \sum_{i=1}^{N} \sum_{j=1}^{O} \psi_{ij}^2 + \delta(v_t^i, v_o^j) * S_p \tag{6}$$

where $\delta(v_t^i, v_o^j)$ is a binary function, that is, if the norm of the direction vector v_t^i is smaller than v_o^j, then $\delta(v_t^i, v_o^j) = 1$, else $\delta(v_t^i, v_o^j) = 0$; v_t^i and v_o^j denote the direction vectors from the robot to the target and the obstacle, respectively; S_p is a constant positive value to identify the fake external occlusion; O denotes the number of obstacles near the target.

Given (5) and (6), the occlusion term of the detectability function in (1), including the internal occlusion and the external occlusion, is given as follows:

$$f_{occ} = f_{occ}^{in}(\boldsymbol{x}_{t+1}, P_{pre}^{t+1}) + f_{occ}^{ex}(\boldsymbol{x}_{t+1}, P_{pre}^{t+1}) \tag{7}$$

3) FOV Deviation: Due to the limited FOV of the sensors, the target is undetectable if the target exceeds the FOV. The track distance term f_{dis} and the occlusion term f_{occ} reveal the relationship between the relative geometry of the sensor's and the targets' position and the detectability, but the orientation of the sensors is not concerned. Therefore, the FOV deviation term f_{fov} constrains the detectability by keeping the targets remain centered within the sensors' FOV. f_{fov} is defined as follows:

$$f_{fov}(\boldsymbol{x}_{t+1}, P_{pre}^{t+1}) = \sum_{i=1}^{N} e^{-(\beta_{t+1}^i)^2} \tag{8}$$

where β_{t+1}^i denotes the angle difference between the direction from the current robot's position to the position of i-th target and the current yaw direction of the sensor, as depicted in Fig. 3(c).

According to (2), (7), and (8), the detectability measure in (1) is able to evaluate the tracking performance for guiding the robotic motion planning. A larger detectability value means better observation of the targets.

B. Single Step Optimization-based Robotic Planning

After defining the detectability function, an optimization problem is formulated to determine the next best viewpoint by maximizing the detectability of targets while maintaining a safe distance from obstacles. The mathematical formulation is presented as follows:

$$\begin{aligned} \boldsymbol{u}_{\boldsymbol{t+1}}^* &= \arg\min_{\boldsymbol{u_{t+1}}} \; J_c(\boldsymbol{x}_{t+1}, P_{pre}^{t+1}) \\ \text{s.t. } \boldsymbol{x}_{t+1} &= f\left(\boldsymbol{x}_t, \boldsymbol{u}_t\right) \end{aligned} \tag{9}$$

where $\boldsymbol{u}_t$ denotes the control command at time t; $\boldsymbol{x}_{t+1} = f\left(\boldsymbol{x}_t, \boldsymbol{u}_t\right)$ defines the kinematic equations of the robot; the objective function J_c is defined as follows:

$$J_c(\boldsymbol{x}_{t+1}, P_{pre}^{t+1}) = \; -f_{det}(\boldsymbol{x}_{t+1}, P_{pre}^{t+1}) + \lambda_c \cdot c_{col}(\boldsymbol{x}_{t+1}). \tag{10}$$

where c_{col} denotes collision cost, λ_c is a design weight.

1) Robotic Kinematic: Since the differential drive mobile robot is nonholonomic and the sensor is of limited FOV, it is difficult for the robot to track the targets while keeping the velocity consistent with the targets and the sensor is directed to the targets. To this end, a mobile robot system wherein sensors are affixed to a yaw-controllable gimbal is developed to augment sensor flexibility for target tracking. The kinematic equation of the robot with a yaw-controllable

gimbal in (9) is expressed as:

$$\boldsymbol{x}_{t+1} = \boldsymbol{x}_t + \begin{bmatrix} \cos\theta_t^b & 0 & 0 \\ \sin\theta_t^b & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \boldsymbol{u}_t \Delta t \tag{11}$$

where $\boldsymbol{x}_t = \begin{bmatrix} x_t & y_t & \theta_t^b & \theta_t^{\text{yaw}} \end{bmatrix}^T$ includes the robot position x_t and y_t, the yaw angle of the robot base θ_t^b and the yaw angle of the gimbal θ_t^{yaw}. $\boldsymbol{u}_t = \begin{bmatrix} v_t^b & \omega_t^b & \varphi_t \end{bmatrix}^T$ is represented by linear velocity v_t^b, the angular velocity ω_t^b of the robot base, and the yaw velocity φ_t of the gimbal.

2) Collision Avoidance: The soft-constrained method [20] is widely used for collision avoidance in trajectory optimization; it regards the distances between the robot and surrounding obstacles as cost function to drive the robot away from the obstacles. However, integrating the complex environment into the soft constraints remains an open issue. Many researches [17], [21] simplify the dynamic person to an ellipsoid, and the distance from the robot to the ellipsoid can be used as a soft constraint for collision avoidance. But the ellipsoid can not envelop snugly many irregularly shaped obstacles, such as a "L" shaped wall. We develop a method to simplify the complex obstacles into cylinders with different sizes, aiming at incorporating the obstacles into the collision cost function and the external occlusion function f_{occ}^{ex} for occlusion and collision avoidance. This obstacle modeling method is applicable to static and dynamic environments.

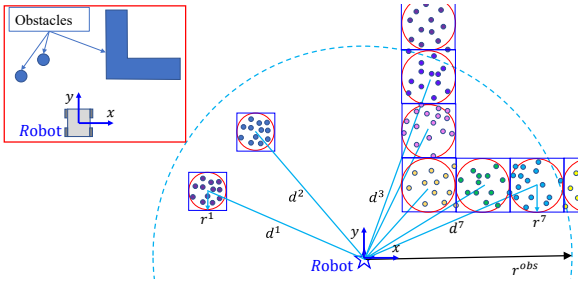


Fig. 4. Diagram of obstacle modeling for a complex environment with a "L" shaped wall and two cylinder obstacles in 2D projection. The points denote the point cloud scanned on the wall. The points with different colors belong to different clusters; the blue squares and the red circles are the bounding boxes and the minimal enclosing cylinders for each cluster, respectively. The blue pentagram denotes the position of the robot.

The obstacle modeling method includes point cloud clustering, segmentation, and shape-adaptive partition. Specifically, the point cloud is first segmented into ground point cloud and non-ground point cloud [22]. As shown in Fig. 4, the non-ground points are segmented into several clusters according to the x-y coordinate. Subsequently, we extract bounding boxes and identify the minimal enclosing cylinders for each cluster. The cylinders, located within a defined radius r^{obs} of the robot, are used to calculate collision cost for avoiding obstacles. The collision cost is formulated as:

$$c_{col}(\boldsymbol{x}_{t+1}) = \sum_{j=1}^{O} e^{-(d_{t+1}^j - r^j - d^s)} \tag{12}$$

where d_{t+1}^j denotes the distance between the robot and the center of the j-th cylinder; r^j denotes the radius of the j-th cylinder; d^s is the preset safe distance threshold.

By solving the optimization problem in (9), the next best action $\boldsymbol{u}_{\boldsymbol{t+1}}^*$ is computed by minimizing the objective function J_c for ensuring the detectability and collision avoidance.

C. Model Predictive Control-based Robotic Planning

Relying on single-step optimization to determine the next best viewpoint for target tracking leads to erratic and discontinuous robot motions. Because the single-step optimization only considers the performance of the next state, it is unable to involve future states to perform overall optimization for robot motion. Thus, a target tracking method based on MPC is developed, which optimizes a series of robotic actions to ensure the tracking performance under the robotic motion constraints over a finite prediction horizon. Specifically, we formulate an optimization problem of MPC to minimize $\mathcal{J}_c$ in (10) and the motion cost $\mathcal{J}_u$ under the robotic kinematic model and motion limitations, ultimately yielding optimal control commands $\boldsymbol{u}^*$ of the robotic base and the gimbal to track targets. The optimization problem is defined as follows:

$$\begin{aligned} \boldsymbol{u}^* = \arg\min_{\boldsymbol{u}} & \sum_{k=t+1}^{t+L} J_c(\boldsymbol{x}_k, P_{pre}^k) + \sum_{k=t+1}^{t+L} \mathcal{J}_u(\boldsymbol{u}_k) \\ \text{s.t. } & \boldsymbol{x}_{k+1} = f(\boldsymbol{x}_k, \boldsymbol{u}_k) \\ & \boldsymbol{x}_{\min} \le \boldsymbol{x}_k \le \boldsymbol{x}_{\max} \\ & \boldsymbol{u}_{\min} \le \boldsymbol{u}_k \le \boldsymbol{u}_{\max} \end{aligned} \tag{13}$$

where $\boldsymbol{x}_k$ and $\boldsymbol{u}_k$ represent the robot state and the control command at time k; $\mathcal{J}_u = \lambda_u \cdot \|\boldsymbol{u}_k\|$ encapsulates the avoidance of excessive robot motion; λ_u is a design weight.

To efficiently solve this optimization problem, we employ Cppad [23] for modeling and IPOPT [24] for optimization. The prediction horizon L is set to 15. Then, the first control action $\boldsymbol{u}_{t+1}^*$ is selected and sent to the underlying controller according to the MPC control law. Afterward, the robotic base and the gimbal are driven to chase targets.

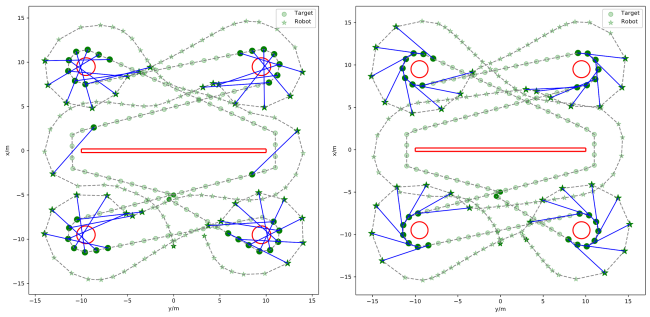
III. Simulations and Experiments

A. Simulation and Experimental Platforms

We conducted several simulations and real-world experiments to verify the performance of the proposed method. An experimental differential wheeled robot with a yaw-controllable gimbal is shown in Fig. 2, and a simulated robot is crafted in Gazebo. Our sensors include a camera with the FOV $87° \times 58°$ and an Ouster LiDAR possessing a wide FOV spanning $360° \times 45°$. The robot hardware configuration is CPU with i9-11950H and GPU with NVIDIA GeForce RTX 3080.

B. Evaluation of Target Tracking in Simulation

Three distinctive scenarios of tracking one, two, and three targets (i.e., pedestrians) were designed to verify the proposed target tracking algorithm.



(a) Elastic Tracker [12] (1.5 m/s) (b) Ours (1.5 m/s)

Fig. 5. Demonstration of the target tracking results under our method and Elastic Tracker [12]. The red rectangle and circles are the obstacles. The dotted lines are the trajectories of the target and the robot. The blue lines denote the selected line of sight from the robot to the target.

TABLE I

THE QUANTITATIVE ANALYSIS OF TRACKING SINGLE TARGET.

Target Velocity (m/s)		1	1.5	2
Distance Error (m)	Ours	**0.21**	**0.21**	**0.27**
	E-T [12]	0.88	1.09	1.25
External Occlusion	Ours	**15.03**	**15.12**	**14.65**
	E-T [12]	20.38	34.52	90.42
FOV Deviation(rad)	Ours	**0.06**	**0.07**	**0.08**
	E-T [12]	0.12	0.16	0.19

1) Single Target Tracking: The scenario of single target tracking includes a wall and four cylinders, indicated as the red rectangle and four red circles in Fig. 5. The target followed a predefined trajectory, going around the wall and cylinders. We conducted comparative analyses between our approach and Elastic Tracker (E-T) [12], focusing on target motions at three distinct velocities: 1 m/s, 1.5 m/s, and 2 m/s. Elastic Tracker serves for single target tracking that guarantees safety and visibility based on trajectory optimization.

The tracking results of both methods at a velocity of 1.5 m/s are shown in Fig. 5. Elastic Tracker faced numerous occlusion challenges, particularly when the target neared the wall or cylinders. In contrast, our method handled the external occlusion, ensuring uninterrupted target perception. A detailed performance comparison across various target velocities can be found in the supplementary video. The quantitative analysis of the tracking results is shown in TABLE I. We use the track distance error of $(d^{trk} - d^{ref})^2$, the external occlusion metric, and the square error of FOV deviation β, to evaluate the tracking performance. Our method outperformed Elastic Tracker across these evaluation metrics, underscoring its effectiveness. The occlusions of Elastic Tracker results from the fact that Elastic Tracker regards the occlusion avoidance as hard constraint in the trajectory optimization, which makes the target in edge areas of the sensor's FOV. This causes occlusion when the target changes the motion direction suddenly; for example the target suddenly moves behind the cylinders.

2) Two-target Tracking: The two-target tracking scenario, as shown in Fig. 6, encompasses targets traveling around

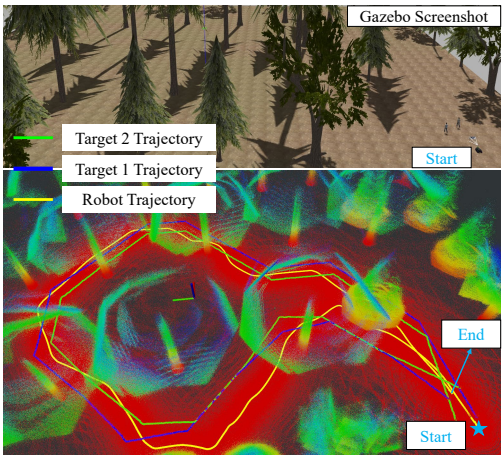


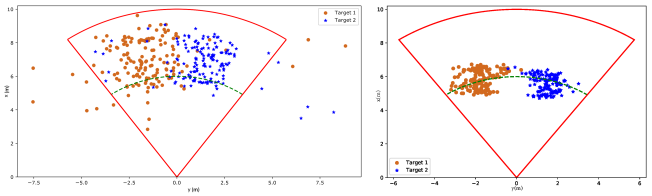
Fig. 6. Illustration of two-target tracking in a simulated field environment.

TABLE II

THE QUANTITATIVE ANALYSIS OF TRACKING TWO TARGETS.

		Ours	Dual Chaser [3]
Distance Error (m)		**0.36**	1.36
Occlusion		**11.31**	12.15
Out_of_FOV Times		**0**	12
Velocity (m/s)	Mean	2.23	2.41
	STD	**0.63**	0.98
Twist (rad/s)	Mean	-0.02	-0.02
	STD	**0.24**	0.48

the forest, following along the predefined trajectories with a velocity 2 m/s. Since Elastic Tracker serves for single target tracking, we compared our method with Dual Chaser, which is a method for chasing two targets with visibility assurance and collision avoidance [3].



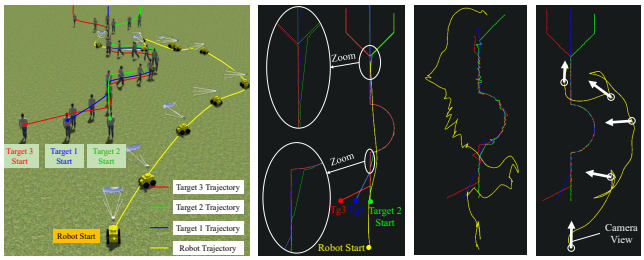
(a) Dual Chaser (b) Ours

Fig. 7. Distribution of the target positions under Dual Chaser [3] and our method. The red sector is the sensor's FOV, and the green dotted arc denotes the reference distance d^{ref}. The blue and orange points denote the position of the targets projected to the sensor's FOV.

The target positions projected to x-y plane in the sensor's FOV, shown as Fig. 7, is used to evaluate the tracking performance. Dual Chaser exhibits a scattered distribution of the target positions, with instances of targets exiting the FOV a dozen times. In contrast, our approach consistently maintains a centralized target distribution within the FOV, closely adhering to the reference distance. Quantitative metrics in TABLE II verify our method's superiority in motion stability, as indicated by the standard deviation (STD) values of velocity and robot twist. As shown in the material video, the tracking process of Dual Chaser is not smooth enough,

leading to scattered distribution and unstable motion.

3) Multi-targets Tracking: We conducted simulations for multi-target tracking, as depicted in Fig. 8. As there is no mature scheme for tracking three targets, we select two basic methods for comparison, which include (1) PID tracking - this method regards the center of the multi-target positions as the tracking point without considering the occlusion. (2) Sensor Fixed Tracking (SFT) - this method uses our tracking method but with the sensor system fixed on the robot.



(a) (b) PID (c) SFT (d) Ours

Fig. 8. Results of tracking three targets. (a) Simulated environment. (b), (c), (d) show the targets and robot's trajectories under different methods of PID, SFT, and Ours, respectively.

Actual trajectories generated by PID, SFT, and Our method are shown in Fig. 8(b), (c), and (d), respectively. Due to the inability of the PID tracking method to perceive internal occlusions, target 1 (blue trajectory) and target 2 (green trajectory) moving ahead were occluded by target 3 (red trajectory) moving behind in the two zoom areas shown in Fig. 8(b). In Fig. 8(c), the initial segment of the robot's motion trajectory is meandering, and ultimately leading to the failure of target tracking due to the insufficient motion flexibility of the mobile robot with the fixed sensor system. From Fig. 8(d), the robot tracks the three targets well by our method and ensures that the three targets are always in the robot's FOV as the targets' formation changes.

C. Targets Tracking in Real-world Environments

To verify the effectiveness of our tracking method in real-world environments, we conducted single target tracking in different scenarios and two targets moving on the playground with dynamic formations.

1) Single Target Tracking: The single target tracking process in the garden scenario is shown in Fig. 9. The robot under our method keeps the target in the FOV and maintains a suitable tracking distance. When the target travels around the parterre, the robot changes the motion to avoid external occlusion with obstacles.

2) Two-target Tracking: The two targets tracking process is shown in Fig. 10. The robot under our method keeps the two targets in the FOV and maintains a suitable tracking distance. The robot actively changes motion direction and speed to avoid internal occlusion when the targets' formation changes from S2 to S4.

IV. Conclusion

This paper achieved robotic target-tracking by simultaneously ensuring continuous target visibility and collision avoidance with limited sensors FOV in complex environments. We analyzed the interaction mechanism of target tracking between the robot, the environment, and the targets. The detectability measure considering tracking distance, occlusions, and FOV constraint is formulated to correlate the tracking performance. Furthermore, based on the detectability measure, the robotic motion planning framework was proposed to achieve continuous and robust tracking of single, two, and three targets in complex environments. This framework based on MPC directly optimizes the control commands for tracking targets safely, leading to superior tracking performance compared to baseline methods across various simulation scenarios. In the future, we will integrate the effect of rough terrains on the target tracking performance into the detectability and achieve multi-target tracking in rough terrains and complex outdoor environments like all-day time tracker [25].

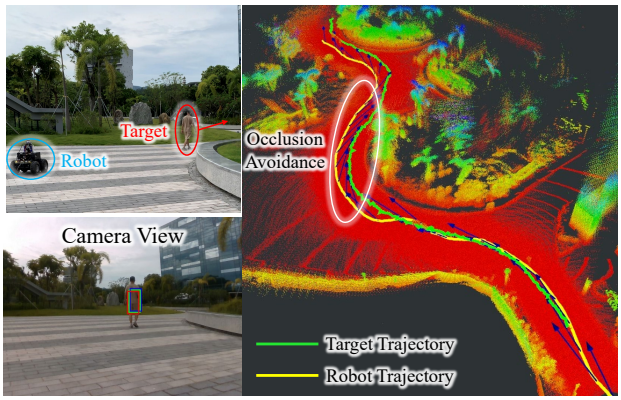


Fig. 9. The tracking process on single target in real-world experiment. The green trajectory is the estimated trajectory of the target, and the yellow trajectory is the trajectory of the robot. The blue arrows denote the viewpoints from the robot to the target.

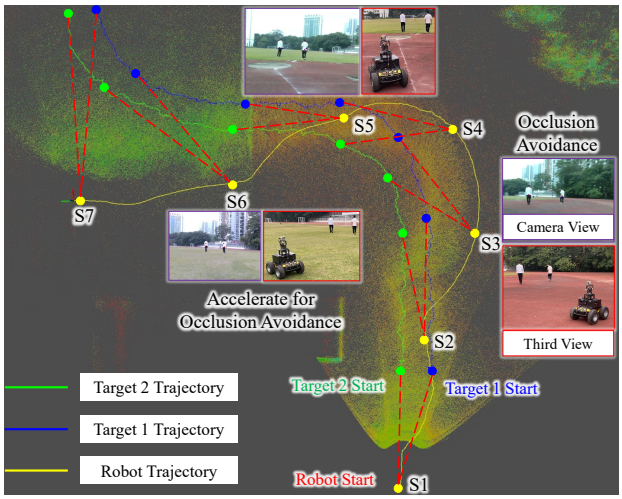


Fig. 10. The tracking process on two targets in real-world experiment. The blue trajectory is the estimated trajectory of target 1, the green trajectory is the estimated trajectory of target 2, and the yellow trajectory is the trajectory of the robot. S1, S2, S3, S4, S5, S6, and S7 are selected results during the tracking process.

REFERENCES

[1] I. Yadav, M. Sebok, and H. G. Tanner, "Receding horizon navigation and target tracking for aerial detection of transient radioactivity," *The International Journal of Robotics Research*, vol. 42, no. 3, pp. 66–82, 2023.

[2] C. Robin and S. Lacroix, "Multi-robot target detection and tracking: taxonomy and survey," *Autonomous Robots*, vol. 40, pp. 729–760, 2016.

[3] B. F. Jeon, Y. Lee, J. Choi, J. Park, and H. J. Kim, "Autonomous aerial dual-target following among obstacles," *IEEE Access*, vol. 9, pp. 143 104–143 120, 2021.

[4] B. Penin, P. R. Giordano, and F. Chaumette, "Vision-based reactive planning for aggressive target tracking while avoiding collisions and occlusions," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 3725–3732, 2018.

[5] X. Weng, J. Wang, D. Held, and K. Kitani, "3d multi-object tracking: A baseline and new evaluation metrics," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 10 359–10 366.

[6] X. Wang, C. Fu, Z. Li, Y. Lai, and J. He, "Deepfusionmot: A 3d multi-object tracking framework based on camera-lidar fusion with deep association," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 8260–8267, 2022.

[7] H. Ye, J. Zhao, Y. Pan, W. Chen, L. He, and H. Zhang, "Robot person following under partial occlusion," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 7591–7597.

[8] Y. Lee, J. Park, B. Jeon, and H. J. Kim, "Target-visible polynomial trajectory generation within an mav team," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 1982–1989.

[9] J. Gemerek, B. Fu, Y. Chen, Z. Liu, M. Zheng, D. van Wijk, and S. Ferrari, "Directional sensor planning for occlusion avoidance," *IEEE Transactions on Robotics*, vol. 38, no. 6, pp. 3713–3733, 2022.

[10] S. He, H.-S. Shin, and A. Tsourdos, "Trajectory optimization for target localization with bearing-only measurement," *IEEE Transactions on Robotics*, vol. 35, no. 3, pp. 653–668, 2019.

[11] 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.

[12] 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.

[13] L. Xi, X. Wang, L. Jiao, S. Lai, Z. Peng, and B. M. Chen, "Gto-mpc-based target chasing using a quadrotor in cluttered environments," *IEEE Transactions on Industrial Electronics*, vol. 69, no. 6, pp. 6026–6035, 2021.

[14] 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.

[15] H. Masnavi, J. Shrestha, M. Mishra, P. Sujit, K. Kruusamäe, and A. K. Singh, "Visibility-aware navigation with batch projection augmented cross-entropy method over a learned occlusion cost," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 9366–9373, 2022.

[16] 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*, 2023.

[17] 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. 29 009–29 031, 2022.

[18] S. Huang, Z. Wang, J. Ouyang, and H. Chen, "Multi-object detection, tracking and prediction in rugged dynamic environments," in *2023 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. IEEE, 2023, pp. 1–6.

[19] Z. Wang, H. Chen, S. Zhang, and Y. Lou, "Active view planning for visual slam in outdoor environments based on continuous information modeling," *IEEE/ASME Transactions on Mechatronics*, vol. 29, no. 1, pp. 237–248, 2024.

[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] Z. Jian, Z. Yan, X. Lei, Z. Lu, B. Lan, X. Wang, and B. Liang, "Dynamic control barrier function-based model predictive control to safety-critical obstacle-avoidance of mobile robot," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 3679–3685.

[22] M. Oh, E. Jung, H. Lim, W. Song, S. Hu, E. M. Lee, J. Park, J. Kim, J. Lee, and H. Myung, "Travel: Traversable ground and above-ground object segmentation using graph representation of 3d lidar scans," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 7255–7262, 2022.

[23] B. M. Bell, "Cppad: a package for c++ algorithmic differentiation," *Computational Infrastructure for Operations Research*, vol. 57, no. 10, 2012.

[24] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical programming*, vol. 106, pp. 25–57, 2006.

[25] Y. Wang, H. Chen, Y. Liu, and S. Zhang, "Edge-based monocular thermal-inertial odometry in visually degraded environments," *IEEE Robotics and Automation Letters*, vol. 8, no. 4, pp. 2078–2085, 2023.