2025 IEEE International Conference on Robotics and Automation (ICRA 2025)
May 19-23, 2025. Atlanta, USA

Dynamic Perception-Enhanced Motion Planning and Control for UAVs Flights in Challenging Dynamic Environments

Luyao Liu[1], Jiarui Xu[1] and Hong Zhang[1,2]

Abstract— **The autonomous flights of unmanned aerial vehicles (UAVs) in unknown environments have garnered significant attention. However, most existing methods only achieve safe navigation in static environments or spacious scenes with few moving obstacles. Motivated by this open problem, this paper presents a complete system for safe and autonomous UAVs flights in unknown clustered environments with multiple dynamic obstacles. To properly represent complex dynamic environments, we develop a 3D dynamic Euclidean Signed Distance Field (ESDF) mapping method that initially segments and tracks dynamic obstacles using a novel feature-based association strategy, while fusing the remaining static obstacles into ESDF map. Then, we propose a joint trajectory planning and motion control framework for safely avoiding surrounding obstacles. Specifically, the gradient-based B-spline trajectory optimization algorithm is employed to generate a collision-free static trajectory with respect to static obstacles. To avoid dynamic obstacles while adaptively tracking the static trajectory, we utilize time-adaptive model predictive control combined with Dynamic Control Barrier Function (D-CBF), which maps the collision avoidance constraints of dynamic obstacles onto the control inputs. Extensive simulated and real-world experiments confirm that our proposed method outperforms previous approaches for UAVs flights in challenging dynamic environments.**

I. Introduction

With the advantages of agility and cost-effectiveness, lightweight UAVs have developed widespread applications, such as cargo transportation [1], agricultural spraying [2], and rapid rescue operations [3]. Some complex scenarios impose high demands on the autonomy, intelligence, and reliability of UAVs. In such situations, the UAVs must swiftly adapt to the continuously changing environments while fulfilling their objectives.

There are two significant challenges for UAVs navigation in complex dynamic environments, namely environment perception and path planning. Plenty of existing planning methods [4][5][6][7] focus on static environments, relying on occupancy voxel map [8] or ESDF map [9][10] to represent static environments. These mapping techniques neglect the representation of moving objects, which could degrade the performance of autonomous robots in obstacle avoidance. To safely and proactively avoid obstacles, an effective perception method, encompassing the ability to distinguish moving obstacles and estimate their inherent motion states, is essential for safe navigation in complex dynamic environments. Recently, some advanced planning methods [11][12] have incorporated dynamic obstacle collision constraints, proving effectiveness in dynamic environments with sparse moving obstacles. Due to the consideration of both static and dynamic obstacle avoidance constraints during trajectory planning, these studies easily fail in cluttered dynamic environments with restricted feasible space, struggling to solve a collision-free trajectory. Given the complicated static structures and the unknown number of dynamic obstacles, developing a safe and reliable navigation system is crucial to meet the escalating demands of UAVs flights in increasingly complex applications.

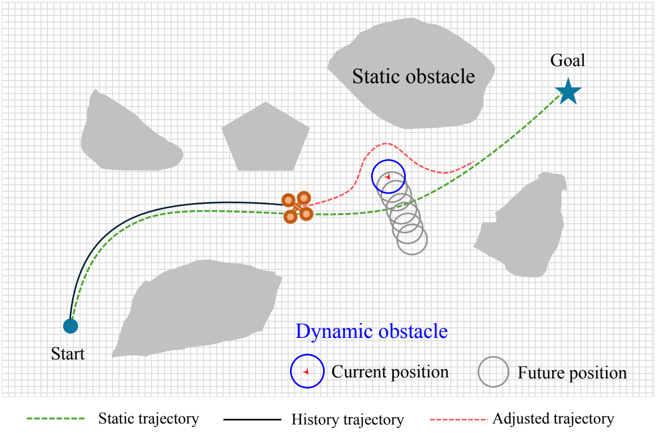


Fig. 1. Illustration of our method working in a complex environment including static and dynamic obstacles. While tracking the static trajectory, the time-adaptive MPC uses D-CBF constraints to avoid dynamic obstacles, resulting in an adjusted trajectory.

To address the above challenges, this paper presents a complete system consisting of a perception module for complex dynamic environments and an obstacle avoidance method utilizing planning and control co-design. For perception, we adopt the feature vector of each obstacle to enhance obstacle association and tracking performance, followed by Kalman filter to distinguish dynamic obstacles and estimate their velocities. Based on this, a 3D dynamic ESDF mapping method is developed, where dynamic obstacles are represented using bounding boxes, while the remaining static obstacles are integrated into ESDF map.

For obstacle avoidance, a unified framework combining trajectory planning and motion control is constructed, accounting for dynamic and static obstacle avoidance strategies separately. As illustrated in Fig. 1, we employ the B-spline trajectory optimization method to generate a feasible and

This work was supported in part by the Shenzhen Key Laboratory of Robotics and Computer Vision (ZDSYS20220330160557001). (Corresponding author: Hong Zhang)

[1] Department of Electronic and Electrical Engineering, Southern University of Science and Technology, Shenzhen, China. liuly2023@mail.sustech.edu.cn

[2] Shenzhen Key Laboratory of Robotics and Computer Vision, Shenzhen, China.

979-8-3315-4139-2/25/$31.00 ©2025 IEEE

smooth trajectory that satisfies static obstacle constraints. Subsequently, the time-adaptive model predictive control is applied to track the static trajectory, with Dynamic Control Barrier Function (D-CBF) serving as the avoidance constraints for dynamic obstacles.

The main contributions of this paper could be summarized as follows:

- For dynamic environment perception, we design a 3D dynamic ESDF mapping method, which represents dynamic obstacles using 3D bounding boxes and static obstacles with an ESDF map. Therein, a feature vector is introduced to facilitate dynamic obstacle matching and tracking, and the Kalman filter is used to distinguish dynamic obstacles and estimate their velocities.
- We propose a unified framework that integrates trajectory planning and motion control for obstacle avoidance in dynamic and complex environments. It combines time-adaptive MPC with D-CBF, enabling active avoidance of dynamic obstacles while tracking the static trajectory.
- Extensive experiments demonstrate the safety and robustness of our proposed method for UAVs flights in challenging dynamic environments.

II. Related Work

A. Perception in Dynamic Environments

To ensure the safe flights of UAVs in unknown dynamic environments, precise environmental perception is crucial. This involves accurately distinguishing moving objects within the environment, estimating their motion states, and mapping complex scenes. With the advancement of deep learning technologies, some learning-based methods [13][14] possess the ability to precisely segment dynamic obstacles. However, these methods require extensive computational resources, making it difficult to run them on lightweight onboard computers. In [15][16], the authors aim to detect and track moving objects by utilizing color and depth images. These studies are limited to detecting particular objects and do not consider integration with mapping functionality. Moreover, [17] presents a Timed-ESDF mapping framework for dynamic environment, which supports distance and metric queries with state-time keys, facilitating gradient-based path optimization. Nevertheless, this approach requires maintaining ESDF map information at multiple time steps, which increases the computational burden. Researchers in [18] utilize the U-depth map [19] to detect both static and dynamic obstacles, and represent them with 3D ellipsoids, but this method is unable to effectively model unstructured static obstacles in complex environments. In [20], the authors utilize a hybrid mapping approach to represent complex scenes. However, this method relies on the displacement of the central position for obstacle matching, which negatively impacts the tracking accuracy of dynamic obstacles.

Inspired by these methods, we first identify dynamic obstacles in the environment, and then represent static and dynamic obstacles using different data structures to better perceive complex dynamic environments.

B. Obstacle Avoidance in Dynamic Environments

In recent years, significant progress has been made in ensuring the safe flights of UAVs in unknown environments. There are several reactive methods that have been developed for dynamic obstacle avoidance, such as the artificial potential field (APF) [21][22] and velocity obstacles (VO) [23][24]. However, these methods only consider the current positions of moving obstacles while neglecting their future states, which makes it challenging to handle dense environments containing both static and dynamic obstacles. Some state-of-the-art methods [4][5][6] have successfully enabled UAVs to navigate safely in complex static environments. Inspired by these works, [20] develop a planning method consisting of modified kinodynamic path searching and gradient-based optimization. In [12], Xu et al. utilize the motion states of dynamic objects to design the receding-horizon distance field aimed at preventing collisions with dynamic obstacles. Based on the DSP map [25], Chen et al. [11] construct a sequence of risk-aware spatio-temporal safety corridors to optimize dynamically feasible and collision-free trajectories. These methods simultaneously incorporate both static and dynamic obstacle avoidance constraints during path planning to generate safe trajectories. However, they encounter challenges in trajectory optimization within crowded dynamic environments, where finding a feasible solution becomes increasingly difficult.

Additionally, the application of Model Predictive Control (MPC) for dynamic obstacle avoidance has attracted considerable attention from researchers. In [26], Guo et al. construct the dynamic obstacle risk region based on the speed of moving obstacles. However, this method is overly conservative, treating the entire area that the moving obstacles could potentially occupy over several future time intervals as hazardous. To address uncertainties caused by various disturbances, [27] formulates a chance-constrained nonlinear MPC problem to enhance obstacle avoidance in dynamic environments. Building on [27], the authors combine reactive chance-constrained MPC with temporal goal tracking strategy to avoid uncertain moving obstacles [28]. However, these methods are limited in their ability to describe complex environmental structures, which undermines the UAV's capacity to navigate effectively in complex environments with unknown quantities of dynamic obstacles.

III. Methodology

This section provides detailed explanation of each component of the proposed system, including the 3D dynamic ESDF mapping in Sec. III-A, the static trajectory planning in Sec. III-B, and the motion control for dynamic obstacle avoidance in Sec. III-C. The algorithm flow of our system is illustrated in Fig. 2.

A. 3D Dynamic ESDF Mapping

For avoiding dynamic obstacles safely, it is essential to distinguish them and estimate their motion states. Drawing inspiration from hybrid mapping paradigms [20], we develop a 3D dynamic ESDF mapping algorithm tailored for complex

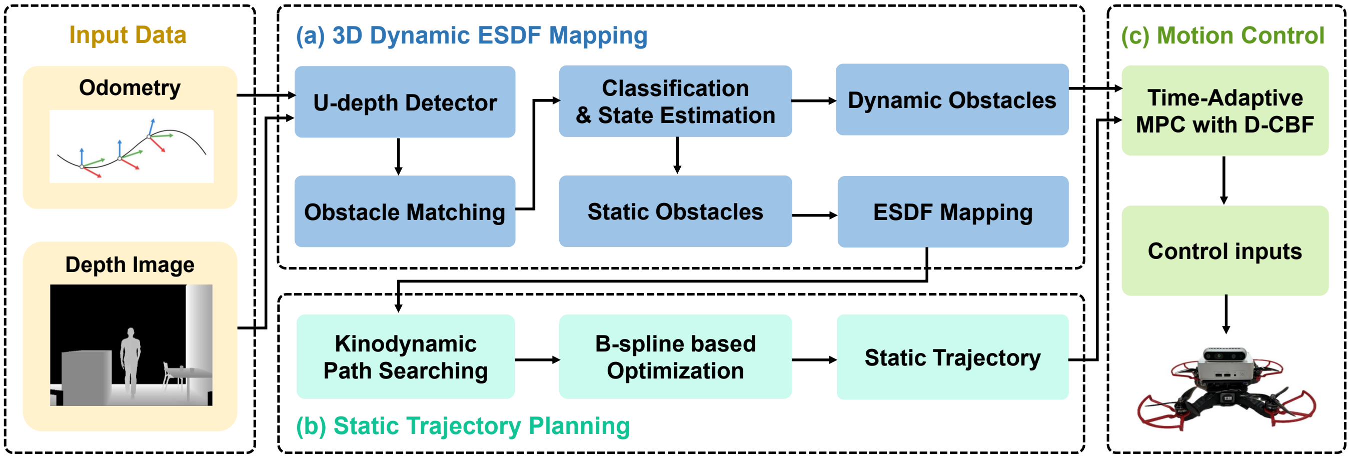

Input Data
Odometry
Depth Image
(a) 3D Dynamic ESDF Mapping
U-depth Detector
Classification & State Estimation
Dynamic Obstacles
Obstacle Matching
Static Obstacles
ESDF Mapping
(b) Static Trajectory Planning
Kinodynamic Path Searching
B-spline based Optimization
Static Trajectory
(c) Motion Control
Time-Adaptive MPC with D-CBF
Control inputs

Fig. 2. System overview. The proposed system consists of three modules: (a) 3D Dynamic ESDF Mapping, (b) Static Trajectory Planning, and (c) Motion Control. The input data, including odometry and depth images, are processed by the 3D Dynamic ESDF Mapping module. This module classifies dynamic obstacles and estimates their velocities, while the static obstacles are represented using ESDF map. Based on this, the Static Trajectory Planning module performs kinodynamic path searching and B-spline-based optimization to generate a collision-free trajectory with respect to the static obstacles. Finally, the Motion Control module tracks the planned static trajectory using time-adaptive MPC integrated with D-CBF constraints, ensuring collision avoidance with dynamic obstacles. The resulting control inputs are sent to the UAV.

environments with multiple moving obstacles. Specifically, static obstacles are modeled using the ESDF map, while dynamic obstacles are represented by the 3D bounding boxes that include velocity information. This method is computationally efficient, enabling it to operate on onboard computers without requiring Graphics Processing Unit (GPU). The following provides the detailed description of our proposed dynamic perception method.

Firstly, the U-depth detector is adopted to obtain 3D bounding boxes for potential obstacles based on depth camera. It directly utilizes the column depth histogram of the raw depth image to produce the U-depth map, which offers a top-view image of these objects. Considering the continuous variation in depth values within the image, a line grouping technique is applied to extract 2D bounding boxes from the U-depth map, as illustrated in Fig. 3(a), which provides rough estimation of each obstacle's length and width. Subsequently, a depth continuity check is performed on the original depth image, using the obstacle's width to determine its height, thereby obtaining the corresponding 2D bounding boxes shown in Fig. 3(b). The 3D points are then triangulated into the camera coordinate system and transformed into the world coordinate system, resulting in 3D bounding boxes that approximate the positions and dimensions of potential obstacles. Notably, these bounding boxes may include both static and dynamic obstacles. To improve the above results, a map refinement strategy is implemented to enhance the detected accuracy of the obstacle positions and dimensions. This strategy inflates the raw 3D bounding boxes using a predefined coefficient in the 3D space, and then searches for occupancy information within the static occupied voxels to determine the minimal bounding box that fully encloses all occupied voxels.

Given the set with n detected obstacles $O_t = \{o_t^0, o_t^1, o_t^2, \cdots, o_t^{n-1}\}$ at the current time t, we need to match them with their positions in the previous frame at $t-1$ for obstacle tracking. In the recent work [20], obstacle matching across two different frames relies on the displacement of their central positions, which may result in mismatches when obstacles are in close proximity. To help obstacle matching and tracking, we construct the statistical feature vector from the point cloud contained within the obstacle's 3D bounding box. It is defined as follow:

$$F(o^i) = [len(o^i), pos(o^i), vol(o^i), M_p(o^i), V_p(o^i)] \quad (1)$$

where $len(o^i)$ is the function that returns the point cloud size of the obstacle, $pos(o^i)$ is the center position of the 3D bounding box, $vol(o^i)$ returns the volume of the 3D bounding box, $M_p(o^i)$ and $V_p(o^i)$ is the position mean and variance of the obstacle's point cloud. Considering the varying magnitudes of each element, the feature vector is normalized. Then, to find the best match $o^j \in O_{t-1}$ for $o^i \in O_t$ at previous time $t-1$, we use the Euclidean distance $d_F = \|F(o^i) - F(o^j)\|_2$ as matching criteria. Only the two with the smallest scores, provided they are below the threshold, are regarded as a match. By matching the i-th obstacle's state with its previous time, we could get its historical trajectory $H_t^i = \{o_{t-1}^i, o_{t-2}^i, o_{t-3}^i, \cdots, o_{t-k}^i\}$. Subsequently, we use Kalman filter to estimate the velocities of obstacles. If the estimated velocity exceeds the predefined threshold, the obstacle is classified as dynamic.

We then clean the specific historical regions traversed by dynamic obstacles. For the remaining regions containing static obstacles, we apply efficient ESDF mapping algorithm to model them. To ensure computational efficiency, we exclusively maintain a local ESDF map, which adequately fulfills the requirements of our task. Finally, the constructed 3D dynamic ESDF map is shown in Fig. 3(c).

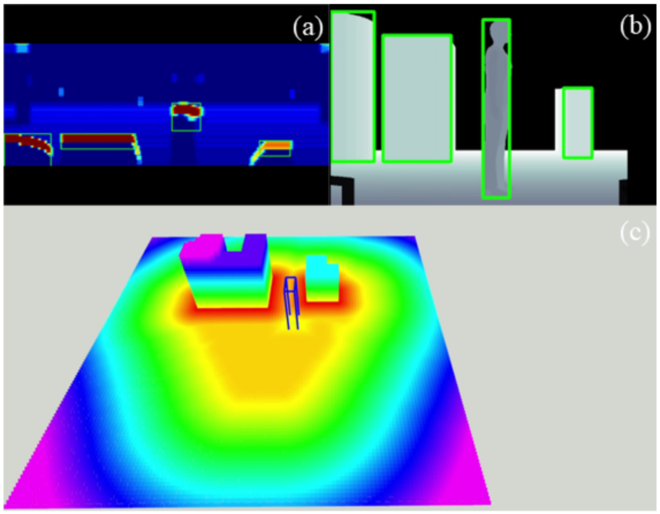


Fig. 3. (a) The 2D bounding box in U-depth map. (b) The depth image with 2D bounding box. (c) The visualization of our proposed dynamic ESDF mapping method, where the blue 3D bounding box represents dynamic obstacle, and the plane is a slice of ESDF map representing static obstacle.

B. Static Trajectory Planning

At this stage, we solely focus on generating a static trajectory that satisfies the static obstacle avoidance constraints. With the ESDF map representing the static regions from the previous phase, it is straightforward to obtain the distance and gradient information regarding the static obstacles for trajectory optimization. Upon receiving the global goal, we employ popular path search algorithms, such as A* and Dijkstra, to derive an initial trajectory. Subsequently, the uniform B-spline optimization method is applied to refine the trajectory by taking account of the distance to obstacles, dynamical feasibility and smoothness.

A B-spline curve is defined by its order μ and consists of several degree $\mu-1$ polynomial basis functions, each associated with control points distributed along a knot vector. The B-spline trajectory could be parameterized by M control points $\{Q_0, Q_1, Q_2, \cdots, Q_{M-1}\}$. We optimize the subset of $M-2(\mu-1)$ control points, as the first and last $\mu-1$ control points should remain unchanged to preserve the boundary conditions. The total cost function is formulated as follows:

$$J_{total} = \omega_c J_c + \omega_d J_d + \omega_s J_s \tag{2}$$

where J_c is the collision cost for static obstacles, J_d denotes feasibility cost, J_s is the smoothness cost. ω_c, ω_d and ω_s are the weighting coefficients for the respective terms.

Leveraging the advantages of the ESDF map representation, we can readily query the distance $s(\mathbf{Q}_i)$ between control point $\mathbf{Q}_i$ and the closest static obstacle. To prevent control points from colliding with static obstacles, we formulate a penalty function $F_c(\cdot)$ for the collision cost. It is defined as the equation:

$$F_c(s(\mathbf{Q}_i)) = \begin{cases} (s(\mathbf{Q}_i) - d_{safe})^2, & \text{if } s(\mathbf{Q}_i) \le d_{safe} \\ 0, & \text{if } s(\mathbf{Q}_i) > d_{safe} \end{cases}, \tag{3}$$

where d_{safe} is user-defined safety distance threshold. Thus, the collision cost for each control point can be incorporated as follow:

$$J_c = \sum_i F_c(s(\mathbf{Q}_i)). \tag{4}$$

The dynamical feasibility of the trajectory is guaranteed by constraining the higher-order derivatives along each individual dimension. With the same time interval Δt between control points, we can calculate the velocity $\mathbf{V}_i$ and acceleration $\mathbf{A}_i$ by taking the first and second derivatives of the B-spline trajectory, respectively. They could be described as follow:

$$\mathbf{V}_i = \frac{\mathbf{Q}_{i+1} - \mathbf{Q}_i}{\Delta t}, \quad \mathbf{A}_i = \frac{\mathbf{V}_{i+1} - \mathbf{V}_i}{\Delta t}. \tag{5}$$

Consequently, based on the permissible maximum velocity v_m and acceleration a_m, the cost function for dynamical feasibility is formulated as:

$$\begin{aligned} J_d = &\sum_i \mathbf{max}\left(\|\mathbf{V}_i\|_2^2 - v_m^2, 0\right)^3 \\ &+ \sum_i \mathbf{max}\left(\|\mathbf{A}_i\|_2^2 - a_m^2, 0\right)^3. \end{aligned} \tag{6}$$

On account of the convex hull property, it is sufficient that making the whole B-spline trajectory smooth by minimizing high order derivatives jerk $\mathbf{J}_i$ of the control points. Therefore, the cost function penalizing smoothness can be expressed as:

$$J_s = \sum_i \|\mathbf{J}_i\|_2^2, \quad \text{where} \quad \mathbf{J}_i = \frac{\mathbf{A}_{i+1} - \mathbf{A}_i}{\Delta t}. \tag{7}$$

C. Motion Control for Dynamic Obstacle Avoidance

The static trajectory generated in the previous stage is treat as reference trajectory. In this subsection, we address the potential collisions with dynamic obstacles while tracking the reference trajectory. Specifically, we design a safety-critical and feasible motion control framework that integrates time-adaptive model predictive control [29] with dynamic control barrier function (D-CBF) [30]. It outputs low-level control inputs for UAV. We begin by introducing the definition of time-adaptive MPC, followed by the explanation of D-CBF.

Time-Adaptive Model Predictive Control: The model predictive controller adjusts the reference trajectory to satisfy constraints while generating control inputs. However, the adjusted trajectory for avoiding dynamic obstacles may cause excessive distance between waypoints, resulting in a mismatch with the predefined time interval. This discrepancy could compromise the UAV's safety. Compared to the traditional MPC with fixed time interval, time-adaptive MPC treats the time interval as variable. The adaptive time interval dt is then determined by minimizing the objective function. Therefore, the problem of simultaneously tracking static trajectories and performing dynamic obstacle avoidance can be formulated as follows:

$$\min_{\mathbf{x}_{t+k|t}, \mathbf{u}_{t+k|t}} \quad \sum_{k=1}^{H-1} \left(\|\mathbf{x}_{t+k|t} - \mathbf{x}_{\text{ref},t+k|t}\|_{\mathbf{Q}}^2 + \|\mathbf{u}_{t+k|t}\|_{\mathbf{R}}^2 \right) \tag{8a}$$

$$\text{s.t.} \quad \mathbf{x}_0 = \mathbf{x}_{\text{init}}, \tag{8b}$$

$$\mathbf{x}_{t+k+1|t} = \mathbf{f}(\mathbf{x}_{t+k|t}, \mathbf{u}_{t+k|t}, dt_{t+k|t}), \tag{8c}$$

$$s\left(\mathbf{x}_{t+k|t}\right) - d_{safe} \geq 0 \tag{8d}$$

$$\text{safety constraints}, \tag{8e}$$

$$t_{\text{lb}} \leq dt_{t+k|t} \leq t_{\text{ub}}, \tag{8f}$$

$$\mathbf{u}_{\text{lb}} \leq \mathbf{u}_{t+k|t} \leq \mathbf{u}_{\text{ub}}. \tag{8g}$$

where $\mathbf{x}_{t+k|t}$ and $\mathbf{u}_{t+k|t}$ denote the state and control inputs, H represents the receding horizon, $\mathbf{Q}$ and $\mathbf{R}$ are weighting matrices, $s(\cdot)$ and d_{safe} have been defined in Sec. III-B, the safety constraints for dynamic obstacles refer to Eq. (11).

Dynamic Control Barrier Function: Compared to traditional CBF [31], the D-CBF takes into account the future states of dynamic obstacles, making it particularly well-suited for application in complex dynamic environments. Based on the proposed 3D dynamic ESDF mapping module, we can obtain the 3D bounding boxes that contain the position and velocity information of dynamic obstacles. Assume that at timestamp t, there are N_d detected dynamic obstacles, which are denoted as a set $O_{d,t} = \{o_{d,t}^0, o_{d,t}^1, \cdots, o_{d,t}^{N_d-1}\}$. And $\mathbf{p}_t$ is defined as the robot's current position. To facilitate subsequent distance calculation, we convert the 3D bounding box model of each dynamic obstacle into ellipsoid in three-dimensional space, where the length, width, and height of the bounding box correspond to the axis lengths of the ellipsoid's x, y, and z dimensions, respectively. For the ellipsoid representing the i-th dynamic obstacle, we adopt $\mathbf{v}_{o_d}^i$, l_x^i, l_y^i, l_z^i and $R_{o_d}^i$ to signify its velocity, axis-length, and rotation matrix relative to the world frame and $\mathbf{p}_{o_d,t}^i$ represent its center position at time t. Besides, the distance vector in the coordinate frame of the i-th dynamic ellipsoid could be described as $\xi_{o_d,t}^i = \mathbf{R}_{o_d}^{i\;\top}\left(\mathbf{p}_t - \mathbf{p}_{o_d,t}^i\right)$. By applying the equations of the line and ellipsoid, the distance $d_{o_d,t}^i$ from the center to its surface of i-th moving obstacle can be calculated by the following equation:

$$d_{o_d,t}^i = \sqrt{\frac{\sin^2(\psi)\cos^2(\phi)}{l_x^{i\,2}} + \frac{\sin^2(\psi)\sin^2(\phi)}{l_y^{i\,2}} + \frac{\cos^2(\psi)}{l_z^{i\,2}}} \tag{9}$$

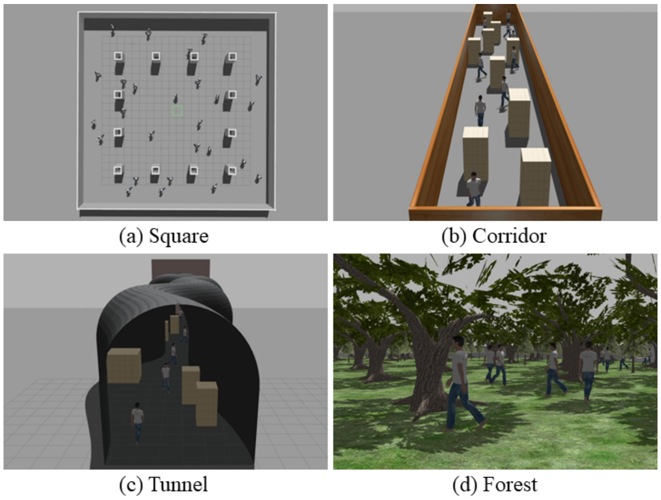
where ψ represents the angle between $\xi_{o_d,k}^i$ and the z-axis of the i-th ellipsoid, and ϕ denotes the angle between the projection of $\xi_{o_d,t}^i$ on the xOy plane and the x-axis of the i-th ellipsoid. As a result, the safety D-CBF constraint for dynamic obstacle avoidance could be expressed as:

$$h(\mathbf{p}_t, \mathbf{p}_{o_d,t}^i) = \|\mathbf{p}_t - \mathbf{p}_{o_d,t}^i\|_2 - d_{o_d,t}^i - d_{safe}. \tag{10}$$

For the discrete-time system, we could get the following function:

$$h(\mathbf{p}_{t+1}, \mathbf{p}_{o_d,t+1}^i) \geq (1-\gamma)h(\mathbf{p}_t, \mathbf{p}_{o_d,t}^i), \tag{11}$$

where function γ is a constant, and $\gamma \in (0, 1]$.


(a) Square

(b) Corridor

(c) Tunnel

(d) Forest

Fig. 4. The four constructed Gazebo environments for simulation test.

IV. Experimental Verification and Results

A. Experimental Setup

Extensive experiments are implemented to validate the performance of our proposed system. The simulation experiments are conducted on a desktop computer equipped with an Intel Core i7-10700F CPU. We use the physics engine-based Gazebo simulator to perform realistic simulation tests. Our programs are developed under the Noetic release of Robot Operating System (ROS). Moreover, the open-source nonlinear optimization solver CasADi [32] is applied to solve constrained optimization problems. In the real-world experiments, our quadrotor platform is equipped with D435i depth camera for environmental sensing and Intel NUC13 as the onboard computer. We utilize visual-inertial odometry (VIO) [33] to estimate the position and velocity of the quadrotor.

B. Dynamic Obstacle Perception

Firstly, we quantitatively evaluate the performance of our proposed perception method for dynamic obstacle tracking and velocity estimation. To facilitate evaluation, we build several simulation scenarios containing walking person as dynamic obstacle in Gazebo. The mean position error and velocity error are adopted as evaluation metrics, with comparison results against other methods presented in Tab. I. It could be seen that our method achieves the lowest position error of 0.09 m and velocity error of 0.15 m/s, which demonstrates that the designed feature vector based on object point clouds improves the performance of dynamic obstacle tracking and estimation.

TABLE I

Dynamic Obstacles Perception Comparison

Method	Position Error (m)	Velocity Error (m/s)
[18]	0.14	0.36
[20]	0.11	0.19
Ours	**0.09**	**0.15**

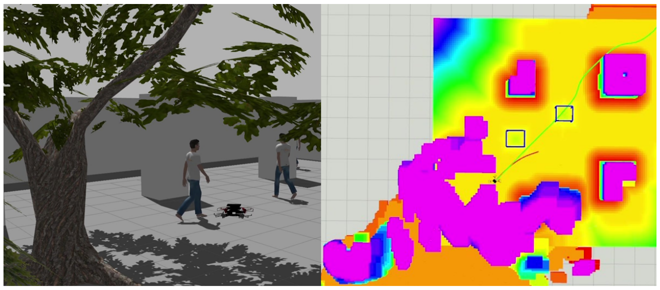


Fig. 5. The left image depicts a UAV flight in dynamic environment within the Gazebo simulation, while the right image shows the corresponding visualization in RViz. The green trajectory represents the static trajectory designed to avoid static obstacles, while the red trajectory indicates the adjusted path for avoiding dynamic obstacles.

C. Simulation Flight Tests

As shown in Fig. 4, we construct four challenging dynamic environments including wide square, narrow corridor, curved tunnel, and clustered forest. The speed of the dynamic obstacles is set between 0.6 and 1.2 m/s. Each method is tested 140 times in each simulation environment, totaling 2,240 tests. There are three metrics utilized to quantitatively evaluate the performance of our system, comprising freezing rate, success rate, and collision rate. The freezing condition occurs when the planner fails to solve a feasible trajectory, causing the UAV to become trapped. To ensure fairness and accuracy, we utilize the Gazebo bumper plugin[1] to detect collisions. It should be noted that if both freezing and collisions occur during a single flight, the first event is regarded as the result of this test.

TABLE II

THE BENCHMARK OF THE FREEZE RATE, COLLISIONN RATE, AND SUCCESS RATE IN THE SIMULATION TESTS.

Method	Freeze Rate	Collision Rate	Success Rate
Fast-planner [4]	47.14%	21.43%	31.43%
RAST [11]	19.27%	27.87%	52.86%
ViGO [12]	14.28%	24.29%	61.43%
Ours	**3.57%**	**9.28%**	**87.15%**

We then benchmark the proposed method against previous state-of-the-art approaches, with the comparison results shown in Table II. It can be observed that our method achieves the highest success rate of 87.15% and the lowest freezing rate of 3.57%, demonstrating its safety and robustness in challenging dynamic environments. Without specifically accounting for the motion of dynamic obstacles, Fast-planner exhibits poor obstacle avoidance performance in dynamic environments. Moreover, ViGO and RAST simultaneously consider both static and dynamic collisions to plan a safe trajectory. However, in narrow dynamic environments with limited navigable space, they often fail to optimize the trajectory, resulting in freezing or collisions. In contrast, our approach first focuses on avoiding static obstacles to easily generate feasible trajectory, and then enhances dynamic obstacle avoidance performance by applying the D-CBF, which considers the future states of moving obstacles as safety constraints. The visualization of the UAV avoiding obstacles is shown in Fig. 5.

D. Real-World Flight Tests

We construct an indoor scene with static obstacle and pedestrian to evaluate our proposed system in a real-world environment. The velocity of real UAV is limited to 1.5m/s. The runtime of each module in real flight is summarized in Table III. A succession of snapshots from one flight test is depicted in Fig. 6, where the UAV safely flies from the starting point to the destination, avoiding various obstacles along the way. The results demonstrate that our proposed system achieves effective obstacle avoidance on a real UAV equipped only with onboard sensors and computer, highlighting its significant potential for applications in dynamic environments.

TABLE III

The Runtime of Each Module of Our Proposed System

Modules	Time(ms)	Portion(%)
3D Dynamic ESDF Mapping	21.43	24.23
Static Trajectory Planning	4.85	5.48
Motion Control	62.17	70.29
System Total	**88.45**	**100**

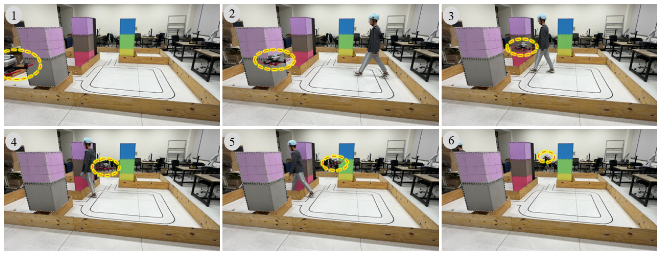


Fig. 6. Snapshots illustrating obstacles avoidance during flight. The number in the top-left corner of each subfigure represents the frame sequence as time progresses.

V. CONCLUSIONS

In this paper, we propose a comprehensive system that integrates environment perception, path planning, and motion control. We develop a 3D dynamic ESDF mapping method, which could identify and estimate dynamic obstacles, and build an ESDF map to represent the remaining static obstacles, facilitating trajectory optimization. Then, a unified planning and control framework is presented for obstacle avoidance. Leveraging the constructed static ESDF map, we adopt the B-spline optimization algorithm to generate a static trajectory that complies with static obstacle avoidance constraints. We utilize time-adaptive MPC to track the static trajectory, while incorporating D-CBF constraints to avoid dynamic obstacles. Benchmark comparisons highlight the safety and robustness of our proposed method during UAVs flights in challenging dynamic environments. In the future, we plan to apply our method to decentralized multi-agent navigation system.

[1] https://classic.gazebosim.org/tutorials?tut=ros_gzplugins#Bumper

References

[1] S. J. Lee, D. Lee, and H. J. Kim, “Cargo transportation strategy using t3-multirotor uav,” in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 4168–4173.

[2] J. Kim, S. Kim, C. Ju, and H. I. Son, “Unmanned aerial vehicles in agriculture: A review of perspective of platform, control, and applications,” *IEEE Access*, vol. 7, pp. 105 100–105 115, 2019.

[3] S. Kulkarni, V. Chaphekar, M. M. Uddin Chowdhury, F. Erden, and I. Guvenc, “Uav aided search and rescue operation using reinforcement learning,” in *2020 SoutheastCon*, vol. 2, 2020, pp. 1–8.

[4] 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.

[5] L. Jarin-Lipschitz, X. Liu, Y. Tao, and V. Kumar, “Experiments in adaptive replanning for fast autonomous flight in forests,” in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 8185–8191.

[6] J. Tordesillas, B. T. Lopez, M. Everett, and J. P. How, “Faster: Fast and safe trajectory planner for navigation in unknown environments,” *IEEE Transactions on Robotics*, vol. 38, no. 2, pp. 922–938, 2022.

[7] W. Zhang, J. Jia, S. Zhou, K. Guo, X. Yu, and Y. Zhang, “A safety planning and control architecture applied to a quadrotor autopilot,” *IEEE Robotics and Automation Letters*, vol. 8, no. 2, pp. 680–687, 2023.

[8] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, “Octomap: an efficient probabilistic 3d mapping framework based on octrees,” *Auton. Robots*, vol. 34, no. 3, p. 189–206, 4 2013. [Online]. Available: https://doi.org/10.1007/s10514-012-9321-0

[9] H. Oleynikova, Z. Taylor, M. Fehr, R. Siegwart, and J. Nieto, “Voxblox: Incremental 3d euclidean signed distance fields for on-board mav planning,” in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 1366–1373.

[10] L. Han, F. Gao, B. Zhou, and S. Shen, “Fiesta: Fast incremental euclidean distance fields for online motion planning of aerial robots,” in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 4423–4430.

[11] G. Chen, S. Wu, M. Shi, W. Dong, H. Zhu, and J. Alonso-Mora, “Rast: Risk-aware spatio-temporal safety corridors for mav navigation in dynamic uncertain environments,” *IEEE Robotics and Automation Letters*, vol. 8, no. 2, pp. 808–815, 2023.

[12] Z. Xu, Y. Xiu, X. Zhan, B. Chen, and K. Shimada, “Vision-aided uav navigation and dynamic obstacle avoidance using gradient-based b-spline trajectory optimization,” in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 1214–1220.

[13] X. Chen, B. Mersch, L. Nunes, R. Marcuzzi, I. Vizzo, J. Behley, and C. Stachniss, “Automatic labeling to generate training data for online lidar-based moving object segmentation,” *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 6107–6114, 2022.

[14] B. Mersch, X. Chen, I. Vizzo, L. Nunes, J. Behley, and C. Stachniss, “Receding moving object segmentation in 3d lidar data using sparse 4d convolutions,” *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 7503–7510, 2022.

[15] M. Lu, H. Chen, and P. Lu, “Perception and avoidance of multiple small fast moving objects for quadrotors with only low-cost rgbd camera,” *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 11 657–11 664, 2022.

[16] G. Chen, W. Dong, X. Sheng, X. Zhu, and H. Ding, “An active sense and avoid system for flying robots in dynamic environments,” *IEEE/ASME Transactions on Mechatronics*, vol. 26, no. 2, pp. 668–678, 2021.

[17] D. Zhu, T. Zhou, J. Lin, Y. Fang, and M. Q.-H. Meng, “Online state-time trajectory planning using timed-esdf in highly dynamic environments,” in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 3949–3955.

[18] J. Lin, H. Zhu, and J. Alonso-Mora, “Robust vision-based obstacle avoidance for micro aerial vehicles in dynamic environments,” in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 2682–2688.

[19] H. Oleynikova, D. Honegger, and M. Pollefeys, “Reactive avoidance using embedded stereo vision for mav flight,” in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 50–56.

[20] Y. Wang, J. Ji, Q. Wang, C. Xu, and F. Gao, “Autonomous flights in dynamic environments with onboard vision,” in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 1966–1973.

[21] N. Malone, H.-T. Chiang, K. Lesser, M. Oishi, and L. Tapia, “Hybrid dynamic moving obstacle avoidance using a stochastic reachable set-based potential field,” *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1124–1138, 2017.

[22] D. Falanga, K. Kleber, and D. Scaramuzza, “Dynamic obstacle avoidance for quadrotors with event cameras,” *Science Robotics*, vol. 5, no. 40, p. eaaz9712, 2020. [Online]. Available: https://www.science.org/doi/abs/10.1126/scirobotics.aaz9712

[23] P. Fiorini and Z. Shiller, “Motion planning in dynamic environments using velocity obstacles,” *The International Journal of Robotics Research*, vol. 17, no. 7, pp. 760–772, 1998. [Online]. Available: https://doi.org/10.1177/027836499801700706

[24] N. Piccinelli, F. Vesentini, and R. Muradore, “Mpc based motion planning for mobile robots using velocity obstacle paradigm,” in *2023 European Control Conference (ECC)*, 2023, pp. 1–6.

[25] G. Chen, W. Dong, P. Peng, J. Alonso-Mora, and X. Zhu, “Continuous occupancy mapping in dynamic environments using particles,” *IEEE Transactions on Robotics*, vol. 40, pp. 64–84, 2024.

[26] B. Guo, N. Guo, and Z. Cen, “Obstacle avoidance with dynamic avoidance risk region for mobile robots in dynamic environments,” *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 5850–5857, 2022.

[27] H. Zhu and J. Alonso-Mora, “Chance-constrained collision avoidance for mavs in dynamic environments,” *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 776–783, 2019.

[28] Z. Xu, D. Deng, Y. Dong, and K. Shimada, “Dpmpc-planner: A real-time uav trajectory planning framework for complex static environments with dynamic obstacles,” in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 250–256.

[29] Z. Zhou, G. Wang, J. Sun, J. Wang, and J. Chen, “Efficient and robust time-optimal trajectory planning and control for agile quadrotor flight,” *IEEE Robotics and Automation Letters*, vol. 8, no. 12, pp. 7913–7920, 2023.

[30] 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)*, 2023, pp. 3679–3685.

[31] J. Zeng, B. Zhang, and K. Sreenath, “Safety-critical model predictive control with discrete-time control barrier function,” in *2021 American Control Conference (ACC)*, 2021, pp. 3882–3889.

[32] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, “CasADi – A software framework for nonlinear optimization and optimal control,” *Mathematical Programming Computation*, vol. 11, no. 1, pp. 1–36, 2019.

[33] T. Qin, P. Li, and S. Shen, “Vins-mono: A robust and versatile monocular visual-inertial state estimator,” *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 1004–1020, 2018.