IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2023, London, UK. Cite as RA-L paper.

RAST: Risk-Aware Spatio-Temporal Safety Corridors for MAV Navigation in Dynamic Uncertain Environments

Gang Chen[1,2], Siyuan Wu[1], Moji Shi[1], Wei Dong[2], Hai Zhu[1] and Javier Alonso-Mora[1]

***Abstract*—Autonomous navigation of Micro Aerial Vehicles (MAVs) in dynamic and unknown environments is a complex and challenging task. Current works rely on assumptions to solve the problem. The MAV's pose is precisely known, the dynamic obstacles can be explicitly segmented from static ones, their number is known and fixed, or they can be modeled with given shapes. In this paper, we present a method for MAV navigation in dynamic uncertain environments without making any of these assumptions. The method employs a particle-based dynamic map to represent the local environment and predicts it to the near future. Collision risk is defined based on the predicted maps and a series of risk-aware spatio-temporal (RAST) safety corridors are constructed, which are finally used to optimize a dynamically-feasible collision-free trajectory for the MAV. We compared our method with several state-of-the-art works in 12000 simulation tests in Gazebo with the physical engine enabled. The results show that our method has the highest success rate at different uncertainty levels. Finally, we validated the proposed method in real experiments.**

***Index Terms*—Aerial Systems, Perception and Autonomy, Motion and Path Planning, Collision Avoidance**

I. Introduction

AUTONOMOUS navigation of micro aerial vehicles (MAVs) in complex, dynamic, and unknown environments has drawn significant attention in recent years, yet it remains an open problem due to many challenges [1]. These challenges include collision avoidance with both complex static obstacles and an unknown number of dynamic obstacles in the environment, and dealing with uncertainties arising from the MAV's noisy obstacle sensing and inaccurate localization.

To tackle these challenges, existing methods for the problem typically make one or more assumptions. For example, a popular pipeline for autonomous flight of MAVs is to use an occupancy grid map [2] to represent the complex environment, then construct a safe flight corridor [3] in it and finally optimize a collision-free trajectory within the safe corridor [4], [5]. However, these methods assume the environment is static. Other works that consider dynamic obstacles typically assume that they can be explicitly segmented from static obstacles [6], their number is known and fixed [7], or they can be modeled with given shapes, e.g. ellipsoids [8], [9] and polytopes [10]. However, these assumptions can hardly be satisfied in many real-world scenarios, where an unknown number of arbitrarily-shaped dynamic or static obstacles can appear, and the MAV's sensing and localization contain non-negligible noise.

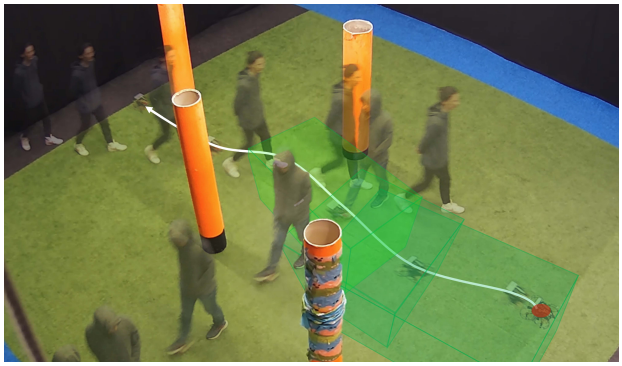

Fig. 1. Autonomous MAV navigation using our risk-aware spatio-temporal (RAST) safety corridors in a dynamic environment. The safety corridors are shown with semi-transparent green cuboids and the flight trajectory is illustrated by a white curve.

In this paper, we present a method for autonomous navigation of MAVs in dynamic uncertain environments without making any of the above assumptions. The novelty of our method is to construct risk-aware spatio-temporal (RAST) safety corridors from a particle-based map, which takes obstacle sensing and localization uncertainties into account. Specifically, first, the dual-structure particle-based (DSP) map [11] is adopted to represent the environment. The map can model arbitrarily-shaped static and dynamic obstacles simultaneously with particles and inherently considers the MAV's environment measurement noise. We then predict the map to the near future taking into account the MAV's localization uncertainty and define a risk metric to evaluate the collision risk of any region in the map. Based on this information, we generate the RAST safety corridors for the MAV, as shown by the green cuboids in Fig. 1. Unlike the safety corridors developed in previous works that guarantee a free space in static environments [3] [12] [13], our RAST safety corridors guarantee that the risk of staying inside the corridors during a future time interval

Manuscript received: July, 5, 2022; Revised October, 13, 2022; Accepted November, 27, 2022.

This paper was recommended for publication by Editor P. Pounds upon evaluation of the Associate Editor and Reviewers' comments. This work was supported in part by the U.S. Office of Naval Research Global (ONRG) NICOP-grant N62909-19-1-2027, National Natural Science Foundation of China (51975348), Shanghai Rising-Star Program (22QA1404400) and the European Research Council (ERC-StG 101041863 - INTERACT).

[1] Gang Chen, Siyuan Wu, Moji Shi, Hai Zhu and Javier Alonso-Mora are with Department of Cognitive Robotics, Delft University of Technology, Delft, The Netherlands. {g.chen-5; s.wu-14; m.shi-5; h.zhu; j.alonsomora}@tudelft.nl

[2] Gang Chen and Wei Dong are with State Key Laboratory of Mechanical System and Vibration, School of Mechanical Engineering, Shanghai Jiaotong University, Shanghai, China. {dr.dongwei}@sjtu.edu.cn

Digital Object Identifier (DOI): see top of this page.

IEEE Robotics and Automation Letters (RA-L) paper, presented at ICRA 2023, London, UK. Cite as RA-L paper.

is lower than a specified threshold. With the RAST safety corridors as constraints, we solve a quadratic programming (QP) problem to find a collision-free and dynamically-feasible trajectory for the MAV.

The main contributions of this paper are:

1) A method for autonomous flight of MAVs in dynamic uncertain environments without making specific assumptions on the obstacles.
2) An approach to construct RAST safety corridors from a particle-based map representation that considers the MAV's obstacle sensing and localization uncertainties.
3) We validate our approach and compare it to state-of-the-art approaches in extensive tests under different levels of uncertainty.

In addition, the source code of this paper, including environment mapping, trajectory planning, and execution is released at https://github.com/tud-amr/RAST_corridor_planning.

II. Related Work

A. Obstacle Avoidance in Dynamic Environments

Obstacle avoidance in dynamic environments requires predictions of the future states of dynamic obstacles, where the environment representation plays an important role. [8] uses depth images to detect and track dynamic obstacles and assume that they can be modeled with ellipsoids whose shapes are known and positions are Gaussian distributed. Collision avoidance trajectories are optimized via chance-constrained model predictive control [7]. In [9], [6] and [14], complex static obstacles are also considered and are modeled by a local voxel map separately [15]. Then a sampling-based [9] or optimization-based [6] [14] planner is used to plan trajectories for the MAV to avoid both static and dynamic obstacles. However, these works also assume that the dynamic obstacles have certain shapes such as cylinders or ellipsoids. To represent arbitrarily-shaped static and dynamic obstacles, researchers in [16] use the DSP map [11] to model all obstacles with particles. Then a sampling-based planner is used to search for a trajectory curve with a low collision cost. However, the method was discrete and localization uncertainty was not considered In the above works, one or more assumptions are made to reduce uncertainties and solve the problem (as discussed in Section I). Learning-based methods [17] [18] [19] can cope with uncertainties implicitly by using raw sensor data as input and letting the network learn the control policy. However, generalizing the learned models to various scenarios is challenging. An explicit method that tackles uncertainties without relying on those specific assumptions is hence required.

B. Autonomous Navigation with Safety Corridors

Safety corridors for MAV navigation originated from [20] a decade ago and were proposed to represent the free space with convex hulls. Since they are represented by linear constraints, trajectory optimization with safety corridors is computationally efficient. The common pipeline to build safety corridors is first to search for a collision-free reference path in the map and then expand convex hulls along with it. [12] searches a reference path with the A* algorithm in the Octomap [21] and generates corridors from the nodes of the path. Corridors are expressed in the voxel form. To include more free space in the safety corridor, [3] utilizes convex polyhedra generated from ellipsoids to express the corridors and turns the optimization problem into a QP problem. The reference path is searched with the Jump Point Search algorithm [22]. In [13], the reference path is given by a human operator's flight path, which is designed to follow the intention of the human operator. These safety corridors are constructed for static environments. To deal with dynamic environments, [23] builds spatio-temporal safety corridors, but the number and shapes of dynamic obstacles are assumed to be known. Spatio-temporal safety corridors have been employed in the autonomous driving field, by using axis-aligned bounding boxes along the reference trajectories to avoid obstacles [24], by building semantic corridors to handle both obstacles and traffic signals [25], or by leveraging S-T graph-based [26] and vertical cell decomposition [27] methods. Compared to these works, which consider deterministic obstacles, our corridor is computed in 3D space and considers risk. This paper investigates risk-aware safety corridors to cope with uncertainties in dynamic environments and realize efficient trajectory optimization.

III. System Overview

The structure of our system is shown in Fig. 2. The major modules include:

(a) DSP map building: We provide a brief introduction to the DSP map in Section IV-A. The DSP map is able to represent arbitrarily-shaped static and dynamic obstacles with particles and takes measurement uncertainty into account. In this paper, we further consider the localization uncertainty of the MAV when predicting the map to the near future for trajectory planning. By propagating the particles in the map, we can predict the collision risk of any region in the map during a given future time interval.

(b) RAST safety corridor generation: We employ a risk-aware kino-dynamic A* algorithm to search for a piece-wise reference path, which is then used to initialize and expand the RAST safety corridors. Fig. 2(b) illustrates the predicted risk of different regions in three future time intervals and the corresponding RAST safety corridors.

(c) Trajectory optimization: Taking the RAST safety corridors as constraints, we can optimize a flight trajectory by solving a QP problem. The optimized trajectory is finally sent to a trajectory tracker to control the MAV.

IV. Methods

We start by briefly introducing the DSP map based on our previous work [11]. Then, we present the method to utilize this map for MAV trajectory planning via the proposed RAST safety corridors.

A. DSP Map Building

The DSP map [11] is an egocentric local map that can represent arbitrarily-shaped static and dynamic obstacles in the environment and is built upon the random finite set (RFS)

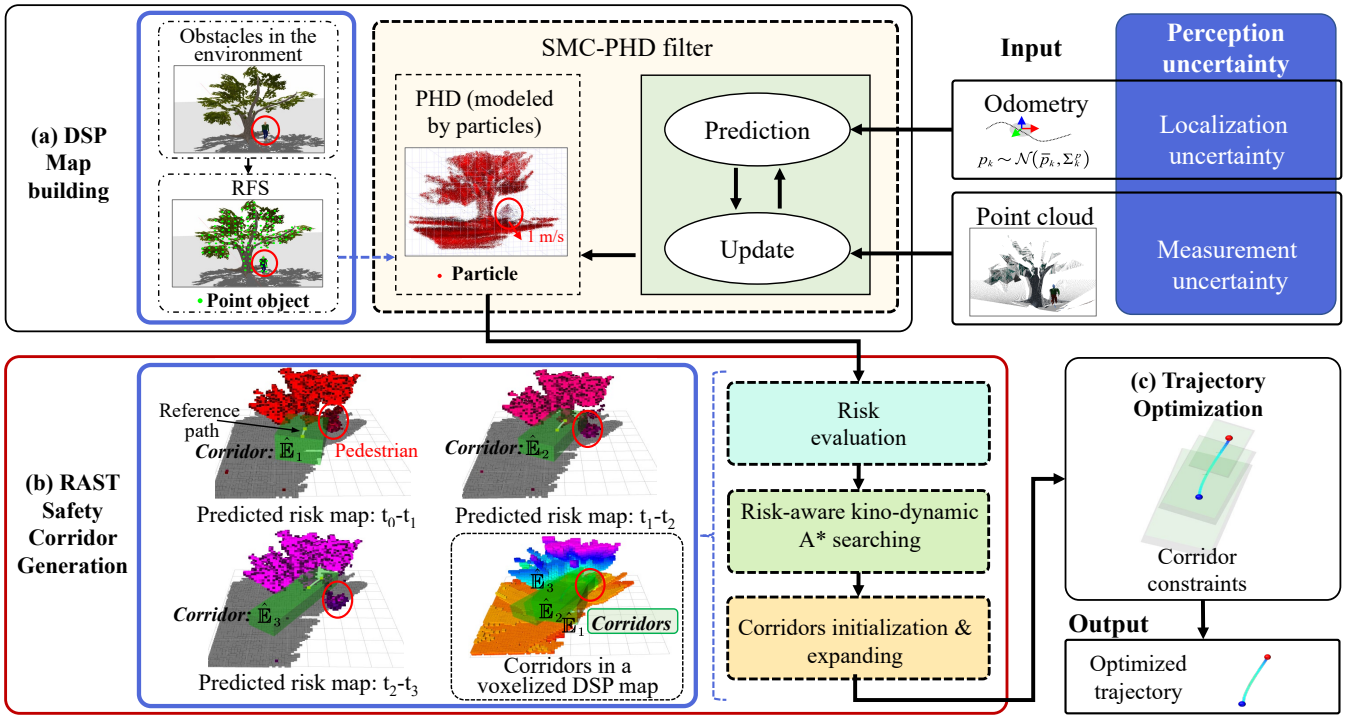

Fig. 2. System structure. The system consists of three modules: (a) DSP map; (b) RAST safety corridor generation; and (c) trajectory optimization. Black arrows show the procedures that our system take from input data with uncertainties to output trajectory. Plots in the blue box in (a) visualize the relationship between obstacles and point objects. The point objects form an RFS. The PHD of the RFS is estimated with particles in the SMC-PHD filter. Red ellipses outline a pedestrian. Plots in the blue box in (b) visualize the defined risk with risk maps in three time intervals, which are $t_0 - t_1$, $t_1 - t_2$, and $t_2 - t_3$. In each risk map of a time interval, the map space is divided into many voxel subspaces. The higher brightness of the voxel corresponds to a larger risk. The ground is painted gray. Red ellipses outline the region with risk caused by a pedestrian. The pedestrian has a downside velocity, and thus the red ellipse moves down as the time increases. The RAST safety corridors are shown with semi-transparent green cuboids. The yellow path in each corridor presents a piece of the reference path searched with the risk-aware kino-dynamic A* algorithm. Three safety corridors are constructed for $t_0 - t_3$ in the figure and are shown together in a voxelized DSP map of the current time (t_0). (c) shows an optimized trajectory where the blue point is the start point, and the red point is the end point. The optimized trajectory is sent to a PID trajectory tracker to control the MAV.

theory [28]. The RFS is defined as a set that has a random but finite number of elements, whose states are also random and finite. In the DSP map, obstacles are represented by many point objects, as illustrated by the green points in Fig. 2(a). Let $\boldsymbol{x}_{t_0} = [\boldsymbol{p}_{t_0}, \boldsymbol{v}_{t_0}]^\mathsf{T} \in \mathbb{R}^6$ be the state of a point object at the current time t_0, which includes its position $\boldsymbol{p}_{t_0} \in \mathbb{R}^3$ and velocity $\boldsymbol{v}_{t_0} \in \mathbb{R}^3$. All the point objects in the map space form a set $\mathrm{X}_{t_0} = \left\{\boldsymbol{x}_{t_0}^{(1)}, \boldsymbol{x}_{t_0}^{(2)}, \cdots, \boldsymbol{x}_{t_0}^{(K)}\right\}$, where $K \geq 0$ is the total number of point objects. Since the obstacles are unknown, K and $\boldsymbol{x}_{t_0}^{(k)}, \forall k \in \{1, 2, \cdots, K\}$ are random. Meanwhile, They are finite because the map is a local map with a limited range. Therefore, X_{t_0} is an RFS.

Instead of estimating the state of each point object, the DSP map estimates the probability hypothesis density (PHD) [28] of the RFS X_{t_0} during its construction process. Denote by $D_{\mathrm{X}_{t_0}}$ the PHD of X_{t_0}. It is the first-order moment of X_{t_0} that describes the density of the hypotheses with different states $\boldsymbol{x}_{t_0} \in \mathrm{X}_{t_0}$. The estimation of $D_{\mathrm{X}_{t_0}}$ is realized by a sequential Monte Carlo PHD (SMC-PHD) filter [29], where numerous particles are predicted and updated iteratively to estimate $D_{\mathrm{X}_{t_0}}$. The input of the SMC-PHD filter includes two channels of data, as shown in the top right of Fig. 2(a). One is the point cloud data from a depth camera which is used in the update procedure. The other is the localization data from the MAV visual odometry which is used in the prediction procedure. The output of the DSP map is a set of n_{t_0} particles. Each particle $i \in \{1, \cdots, n_{t_0}\}$ has a weight $w_{t_0}^{(i)}$ and a state $\tilde{\boldsymbol{x}}_{t_0}^{(i)} = [\tilde{\boldsymbol{p}}_{t_0}^{(i)}, \tilde{\boldsymbol{v}}_{t_0}^{(i)}]^\mathsf{T} \in \mathbb{R}^6$. Let $D_{\mathrm{X}_{t_0}}(\boldsymbol{x}_{t_0})$ be the current PHD at state $\boldsymbol{x}_{t_0}$. According to [11], $D_{\mathrm{X}_{t_0}}(\boldsymbol{x}_{t_0})$ can be estimated with particles by:

$$D_{\mathrm{X}_{t_0}}(\boldsymbol{x}_{t_0}) = \sum_{i=1}^{n_{t_0}} w_{t_0}^{(i)} \delta\left(\boldsymbol{x}_{t_0} - \tilde{\boldsymbol{x}}_{t_0}^{(i)}\right) \tag{1}$$

where $\delta(\cdot)$ is the Dirac function.$\tilde{\boldsymbol{x}}_{t_0}^{(i)}$, $w_{t_0}^{(i)}$ and n_{t_0} are updated dynamically in the filter. The estimation works on the basis of the Law of Large Numbers and n_{t_0} is usually as large as one million [11].

With the PHD calculated with particles in Eq. (1), the current occupancy status of any position (discretized by voxels) in the map can be estimated by checking the number of particles within the corresponding voxel. Details can be found in our previous work [11]. Note that the velocities of these particles are also estimated, which can be used to predict the map to the near future. In the following sections, we first present a map prediction method that takes the MAV localization uncertainty into account (Section IV-B). Then a risk definition is given to evaluate the collision risk of any region in the map (Section IV-C). Next, with the predicted map and defined risk, a risk-

aware spatio-temporal (RAST) corridor is constructed (Section IV-D) which is finally used to optimize a dynamically feasible collision-free trajectory for the MAV (Section IV-E).

B. Map Prediction with Uncertainties

Let X_t be the RFS formed by point objects in the map space at a near future time t and $D_{\mathrm{X}_t}(\boldsymbol{x}_t)$ be the PHD of X_t. To predict $D_{\mathrm{X}_t}(\boldsymbol{x}_t)$, we first predict the future state $\tilde{\boldsymbol{x}}_t^{(i)}$ of each particle in the map via the constant velocity model (CVM) [30], i.e.

$$\tilde{\boldsymbol{x}}_t^{(i)} = \phi(\tilde{\boldsymbol{x}}_{t_0}^{(i)}) = \begin{bmatrix} \boldsymbol{I}_{3\times3} & (t-t_0)\boldsymbol{I}_{3\times3} \\ \boldsymbol{0}_{3\times3} & \boldsymbol{I}_{3\times3} \end{bmatrix} \tilde{\boldsymbol{x}}_{t_0}^{(i)} + \boldsymbol{u} \quad (2)$$

where $\boldsymbol{u} \sim \mathcal{N}(0, \boldsymbol{\Sigma}_{6\times6})$ is the added Gaussian noise of the CVM. To take into account the MAV localization uncertainty which is assumed to be Gaussian distributed, we further add the uncertainty to the particle prediction:

$$\tilde{\boldsymbol{x}}_t^{(i)} = \phi(\tilde{\boldsymbol{x}}_{t_0}^{(i)}) + \boldsymbol{u}' \quad (3)$$

where $\boldsymbol{u}' \sim \mathcal{N}(0, \begin{bmatrix} \boldsymbol{\Sigma}'_{3\times3} & \boldsymbol{0}_{3\times3} \\ \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} \end{bmatrix})$ and $\boldsymbol{\Sigma}'_{3\times3}$ is the position covariance from the odometry. When predicting each particle using Eqs. (2) and (3), $\boldsymbol{u}$ and $\boldsymbol{u}'$ are obtained by sampling from their distributions. $\boldsymbol{\Sigma}_{6\times6}$ and $\boldsymbol{\Sigma}'_{3\times3}$ are determined empirically. We consider the weight of particle constant during the short prediction horizon, i.e. $w_t^{(i)} = w_{t_0}^{(i)}$. Hence, $D_{\mathrm{X}_t}(\boldsymbol{x}_t)$ can be obtained as follows:

$$D_{\mathrm{X}_t}(\boldsymbol{x}_t) = \sum_{i=1}^{l_t} w_t^{(i)} \delta\left(\boldsymbol{x}_t - \tilde{\boldsymbol{x}}_t^{(i)}\right) \quad (4)$$

where n_t is the number of particles within the map range at time t. Since some particles may have moved out of the map range from t_0 to t, then $n_t \leq n_{t_0}$.

The measurement uncertainty of the point cloud has been inherently considered during the DSP map building step [11]. Here, we further explicitly take into account the MAV localization uncertainty from visual odometry in the map prediction. Thus, the dispersion of $\tilde{\boldsymbol{x}}_t^{(i)}$ after the prediction gets larger and will affect the defined risk regions in the environment, which will be illustrated in the next section.

C. Risk Definition

The risk is defined to evaluate the safety level of a subspace in the map during some time interval. Denote by $\mathbb{E}_j \subset \mathbb{R}^3$ a subspace in the map, such as a cuboid subspace shown in Fig. 3. Then, at any time t the point objects within $\mathbb{E}_j$ form an RFS $\mathrm{X}_t^{\mathbb{E}_j}$. Trivially, there is $\mathrm{X}_t^{\mathbb{E}_j} \subset \mathrm{X}_t$ and the PHD of $\mathrm{X}_t^{\mathbb{E}_j}$ is

$$D_{\mathrm{X}_t^{\mathbb{E}_j}}(\boldsymbol{x}_t) = \sum_{i=1}^{n_t^{\mathbb{E}_j}} w_t^{(i)} \delta\left(\boldsymbol{x}_t - \tilde{\boldsymbol{x}}_t^{(i)}\right) \quad (5)$$

where $n_t^{\mathbb{E}_j} \leq n_t$ is the number of particles within the subspace $\mathbb{E}_j$ at time t.

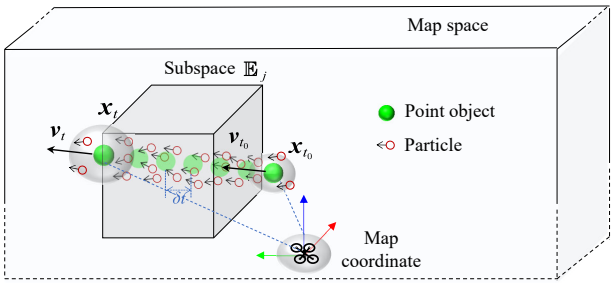

Fig. 3. Illustration of the risk calculation process. The map space is a cuboid space where the MAV is in the center. A point object is moving to the left through a subspace $\mathbb{E}_j$ from time t_0 to t_f. Particles with velocities follow the motion of the point object and model the PHD at the state of point object. Gray ellipsoids indicate the Gaussian distributions.

According to the property of the PHD [28], the cardinality expectation of $\mathrm{X}_t^{\mathbb{E}_j}$, namely the predicted number of point objects in $\mathbb{E}_j$ at time t, can be calculated by

$$E\left[|\mathrm{X}_t^{\mathbb{E}_j}|\right] = \int D_{\mathrm{X}_t^{\mathbb{E}_j}(t)}(\boldsymbol{x}_t)\mathrm{d}\boldsymbol{x}_t = \sum_{i=1}^{n_t^{\mathbb{E}_j}} w_t^{(i)} \quad (6)$$

where $|\cdot|$ is the cardinality and $E[\cdot]$ denotes the expectation. If $E\left[|\mathrm{X}_t^{\mathbb{E}_j}|\right] \geq 1$, the predicted number of point objects in $\mathbb{E}_j$ at time t is larger than one, which suggests that $\mathbb{E}_j$ is likely to contain obstacles and is risky at t.

The risk of the region $\mathbb{E}_j$ during some time interval $[t_0, t_f]$ is then defined with the integral of $E\left[|\mathrm{X}_t^{\mathbb{E}_j}|\right]$ from t_0 to t_f:

$$Risk(\mathbb{E}_j, t_0, t_f) = \int_{t_0}^{t_f} E\left[|\mathrm{X}_t^{\mathbb{E}_j}|\right] \mathrm{d}t \quad (7)$$

This risk indicates the predicted number of point objects in the region $\mathbb{E}_j$ from t_0 to t_f. The risk gets higher if more point objects enter $\mathbb{E}_j$ or the current point objects stay for a longer time. In practice, we discretize the integral with δt. Hence, by combining Eq. (6), we can rewrite Eq. (7) as

$$Risk(\mathbb{E}_j, t_0, t_f) \approx \sum_{t=\{t_0, t_0+\delta t, \ldots, t_f\}} \sum_{i=1}^{n_t^{\mathbb{E}_j}} w_t^{(i)} \delta t \quad (8)$$

Fig. 3 shows a situation where one point object moves through a subspace $\mathbb{E}_j$ from the current time t_0 to a future time t_f. The motions of the particles, which represent the PHD at the point object, are predicted using Eq. (3). $Risk(\mathbb{E}_j, t_0, t_f)$ is evaluated with the summation of the weights of the particles in $\mathbb{E}_j$ at $\{t_0, t_0 + \delta t, ..., t_f\}$ multiplying δt (Eq. (8)). The risk maps in Fig. 2(b) illustrate the risk in three future time intervals with small voxel subspaces. Only the high-risk voxels are presented for clear visualization. A pedestrian is moving downside in the scenario, and thus the high-risk voxels correspond to the pedestrian moving downside as time increases. When the perception uncertainties are considered, particles disperse to a larger area. Fig. 4 reveals the dispersion with the risk map. This dispersion increases the volume of the high-risk regions around the obstacles, especially for the regions that a dynamic obstacle may pass through in the future. As a consequence, the low-risk regions become smaller and the

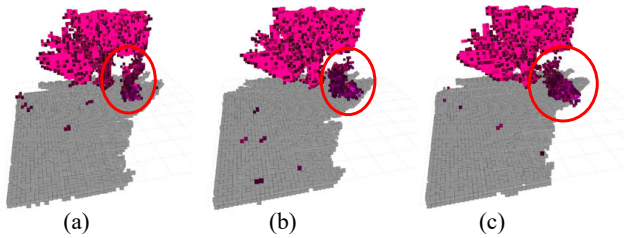

Fig. 4. Comparison of risk maps with and without the consideration of uncertainties in the same scenario as shown in Fig. 2. The predicted time interval is 0.5 s to 1.0 s. Subfigure (a) shows the risk map without considering measurement uncertainty or localization uncertainty. The risk map in (b) considers measurement uncertainty as the original DSP map [11] does. In (c), the localization uncertainty is also considered by using Equation (4) in the prediction step of the SMC-PHD filter. The red ellipses outline the high-risk regions corresponding to a pedestrian. The volume of high-risk regions increases when the uncertainties are considered.

flight path planned in the low-risk regions is safer in dynamic and uncertain environments.

D. Spatio-Temporal Safety Corridor

Using the risk definition in Eq. (8), we first compute a reference path composed of motion primitives using a risk-aware kino-dynamic A* algorithm and generate initial safety corridors. Then the initial safety corridors are expanded to the final RAST corridors.

The risk-aware kino-dynamic A* algorithm is modified from the kino-dynamic A* algorithm in [31] by checking the risk rather than the occupancy status of static grids in the feasibility checking procedure. We illustrate the risk checking process in Fig. 5. For visualization convenience, the process is shown on the 2D plane, but our algorithm works in 3D spaces. Each step in the kino-dynamic A* samples acceleration commands and generates motion primitives to obtain new nodes. Each motion primitive has a fixed planning time t_s. Since the sampled command is acceleration, the motion primitive is a quadratic curve. We first calculate a rectangle envelope for the quadratic curve, as shown in Fig. 5(a) and Fig. 5(b). Considering the size of the MAV, the envelope is expanded by a safety distance d as illustrated in Fig. 5(c) and Fig. 5(d). The expanded envelope is the initial safety corridor, denoted as $\mathbb{E}_j$. The risk of a motion primitive is evaluated by the risk of the corresponding initial safety corridor $\mathbb{E}_j$. Suppose the start time and end time of the motion primitive are t_{j-1} and t_j, respectively. Then the risk of the motion primitive is $Risk(\mathbb{E}_j, t_{j-1}, t_j)$, where $t_j - t_{j-1} = t_s$. Considering the trade-off between high computational efficiency and large searching space in kino-dynamic A*, we choose t_s =0.6 s in practice. Fig. 5(e) shows multiple motion primitives generated in one step of the kino-dynamic A* algorithm. The initial safety corridor $\mathbb{E}_j$ of the selected motion primitive is shown by a red box. No particle moves into $\mathbb{E}_j$ during t_{j-1} to t_j and thus the risk is zero.

By incrementally expanding the initial safety corridor $\mathbb{E}_j$, the final RAST safety corridor $\hat{\mathbb{E}}_j$ can be generated. The expanding process is shown in Fig. 6. To generate the RAST safety corridor $\hat{\mathbb{E}}_1$, we set a local coordinate for the initial corridor, and the expansion is along $+x$, $-x$, $+y$ and $-y$ directions of the local coordinate. Firstly, the initial corridor is

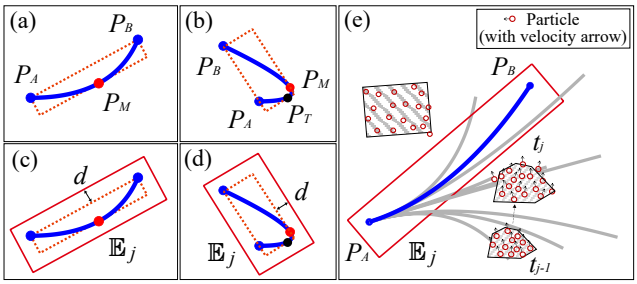

Fig. 5. Illustration of risk checking for each step of the kino-dynamic A* algorithm. Blue curves in (a) and (b) show two sampled motion primitives. Each motion primitive is a quadratic curve. P_A is the start point, and P_B is the endpoint of the curve. The rectangle envelope, shown by the red dashed box, is composed of Line P_AP_B, a parallel line of P_AP_B tangent to the curve, and two lines perpendicular to P_AP_B. The point of tangency on the parallel line is P_M, which locates at the $\frac{t_s}{2}$ point on the motion primitive. P_T in (b) is the point of tangency on a line perpendicular to P_AP_B. P_T occurs when the sampled acceleration command is large. In 3D space, the envelope is a cuboid. In (c) and (d), we expand the envelope by distance d, and get the initial safety corridor $\mathbb{E}_j$. Subplot (e) shows multiple motion primitives sampled in one step of the kino-dynamic A* algorithm in an environment with a static obstacle and a dynamic obstacle. The time interval this step takes is t_{j-1} to t_j. The selected motion primitive is shown in blue, while the others are in gray. The risk of the selected motion primitive is evaluated by calculating the initial corridor generated from this motion primitive, which is $Risk(\mathbb{E}_j, t_{j-1}, t_j)$.

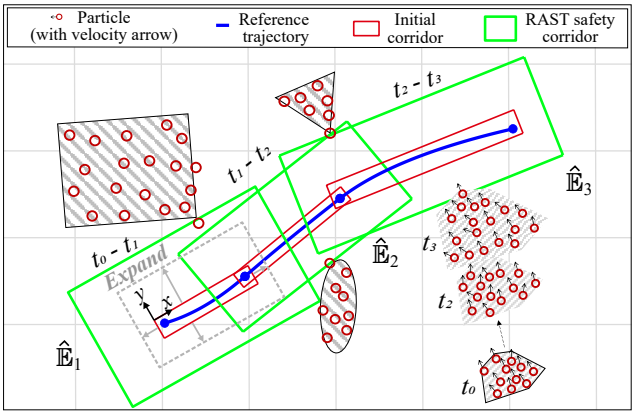

Fig. 6. Illustration of the safety corridor generation process. A three-piece reference path searched by the risk-aware kino-dynamic A* is shown with the blue curve. Red rectangles represent the initial safety corridors, which are then expanded to the RAST safety corridors $\hat{\mathbb{E}}_j$. In the first RAST safety corridor $\hat{\mathbb{E}}_1$, a local coordinate is built on the initial corridor, and an intermediate state in the expanding is shown with a gray dashed box. The expansion is conducted to $+x$, $-x$, $+y$, and $-y$ incrementally with different priorities.

expanded to all directions incrementally with a step distance of Δd. After each expansion, we check the risk of the expanded corridor. If the risk is higher than a threshold R_{max}, the expanding along the low-priority direction is banned, and the next step will only expand towards the remaining high-priority directions. As Fig. 6 shows, expansion along the x-axis mainly enlarges the overlap between this corridor and the next. Since the time step in each corridor is fixed in kino-dynamic A*, enlarging the overlap can hardly help increase the optimization space in practice. In comparison, expansion along the y-axis increases the optimization space and makes it easier to find a safe trajectory in a corridor. Therefore, $\pm x$ has a lower priority than $\pm y$. The priority of $+y$ over $-y$ is set randomly. The expansion stops when all the directions are banned. The risk

World A: Simple school World B: Pedestrian square World C: Pedestrian street

Fig. 7. Simulation worlds. World A: a simple school with very sparse static obstacles, and pedestrians walking slowly (0.5 m/s) in the same direction. World B: a square with dense pedestrians walking in different directions. World C: a street with oak trees and buildings in addition to pedestrians. The walking speed in World B and C ranges from 0.8 m/s to 1.5 m/s.

of the final RAST safety corridor is $Risk(\hat{\mathbb{E}}_j, t_{j-1}, t_j)$. In 3D space, the priority of $\pm z$ is between $\pm x$ and $\pm y$. The reason is that expansion along $\pm z$ can increase the optimization space, but the maneuver along $\pm z$ is realized by thrust rather than attitude changing and, thus, is less flexible than the maneuver along $\pm y$ [28]. Fig. 2(b) presents an example of the RAST safety corridors in 3D space. Note that each corridor has the same time interval (from t_{j-1} to t_j) and only guarantees the risk is lower than R_{max} in this time interval.

E. Trajectory Optimization

The trajectory is a N-order M-piece polynomial trajectory where M equals the number of safety corridors and $N = 7$, corresponding to the minimum snap cost [20]. Let $f_j(t), j \in \{1, 2, ..., M\}$ denote the trajectory piece in the j-th corridor. According to Section IV-D, $t \in [t_{j-1}, t_j]$ and $t_j - t_{j-1} = t_s$. Thus $f_j(t)$ can be formulated as:

$$f_j(t) = \mathbf{c}_j^{\mathsf{T}} \boldsymbol{\beta}(t_j), \quad t \in [t_{j-1}, t_j) \tag{9}$$

where $\mathbf{c}_j \in \mathbb{R}^3 \times \mathbb{R}^{N+1}$ is the coefficients' vector of the polynomial and $\boldsymbol{\beta}(t) = [1, t, t^2, t^3, \cdots, t^N]^{\mathsf{T}}$ is the natural basis. The optimization of $\mathbf{c}_j$ is:

$$\min_{\mathbf{c}_j} \sum_{j=1}^{M} \int_{t_{j-1}}^{t_j} \left\| \frac{\mathrm{d}^4 f_j(t)}{\mathrm{d}t^4} \right\|^2 dt \tag{10a}$$

$$\text{s.t.} \quad f_1(t_0) = f_0, \ f_M(t_M) = f_M \tag{10b}$$

$$f_j(t) \in \hat{\mathbb{E}}_j, \forall t \in [t_{j-1}, t_j] \tag{10c}$$

$$f_j(t_j) = f_{j+1}(t_j) \tag{10d}$$

$$f_j^{(m)}(t) \leq f_{max}^{(m)}, \forall t \in [t_0, t_M], \quad m = 1, 2 \tag{10e}$$

where f_0 is the initial state of the MAV given by the odometry; f_M is the terminal state provided by the last node in the kino-dynamic A* algorithm; $f_j^{(1)}(t)$ and $f_j^{(2)}(t)$ are first- and second-order derivatives, namely velocity and acceleration, of the trajectory; $f_{max}^{(1)}$ and $f_{max}^{(2)}$ are the velocity and acceleration limitations. The constraints include boundary value constraints (10b), corridor constraints (10c), continuity constraints (10d), and maximum acceleration and velocity constraints (10e). The corridor constraints (10c) can be transformed to the linear constraints to the poles of each polynomial piece [12] and the optimization problem is thus a QP problem that can be solved very efficiently [12].

V. Results

In this section, we first compare the obstacle avoidance performance of our method with several state-of-the-art methods in simulation tests. Then the real-world tests are presented to further validate our method.

A. Simulation Tests

Simulation tests were conducted with the IRIS quadrotor running PX4 firmware in the Gazebo simulation environment on a laptop equipped with an AMD R7-4900H CPU. The full physical engine in Gazebo was enabled to better simulate the dynamics of the MAV. Three simulation worlds shown in Fig. 7 were utilized in the tests. We first tested our method with different values of R_{max} to investigate the best risk threshold. Twenty tests were conducted in each simulation world with each risk threshold. The average success rate and planning time at different risk thresholds are shown in Fig. 8 (a). The top average success rate is 0.68 when $R_{max} = 0.2$. The planning time includes optimization, corridor generation, and A* searching time. As R_{max} increases, the size of the expanded corridor gets larger, and the trajectory optimizer takes less time to find a safe flight trajectory. The overall planning time decreases from 7.1 ms to 6.3 ms. Since 7.1 ms is already efficient, $R_{max} = 0.2$ is adopted for a high success rate. The average mapping time is 68.5 ms. Memory usage is mainly consumed by mapping and is about 300 MB.

Then our method was compared with three recent works: the Faster method [32], the FDF method [6], and the RAS method [33]. In these works, Faster utilizes a corridor-based method assuming obstacles are static. FDF and RAS use non-corridor-based methods and consider both static and dynamic obstacles. We did a full factorial experiment with four levels of measurement uncertainty and five levels of localization uncertainty. The input point cloud came from a simulated depth camera. Let $\bar{r}$ denote the accurate depth value of a pixel in the depth image from the camera. To simulate the measurement uncertainty, we added Gaussian noise with standard deviation (S.D.) $\sigma^m = \{0, \%2\bar{r}, \%5\bar{r}, \%10\bar{r}\}$ to the real depth $\bar{r}$. Thus the noised depth follows $r \sim \mathcal{N}(\bar{r}, \sigma^m)$, which simulates the depth from a real depth camera r is then used to calculate the point cloud with the pinhole model.

For the localization uncertainty, Gaussian noise with S.D. $\sigma^p = \{0, 5, 10, 15, 20\}$ cm was added to the ground truth odometry. Each method was tested 50 times under each noise condition in each simulation world. Totally, 12000 tests were conducted. A height limit of 1.8 m was added so that the MAV could not fly over the dynamic obstacles. The maximum velocity was set to 2 m/s. $\boldsymbol{\Sigma}_{6\times6}$ and $\boldsymbol{\Sigma}'_{3\times3}$ are set to be the same as the variance of measurement noise and localization noise, respectively.

We evaluate the performance with four metrics: the average success rate, the rate of collision to static obstacles, the rate of collision to dynamic obstacles, and the freezing rate. A freezing happens when the planner cannot find a feasible trajectory and the MAV is trapped. Fig. 8 (b) shows the average performance in different worlds. From World A to C, the environment is getting more complex, and thus, the success

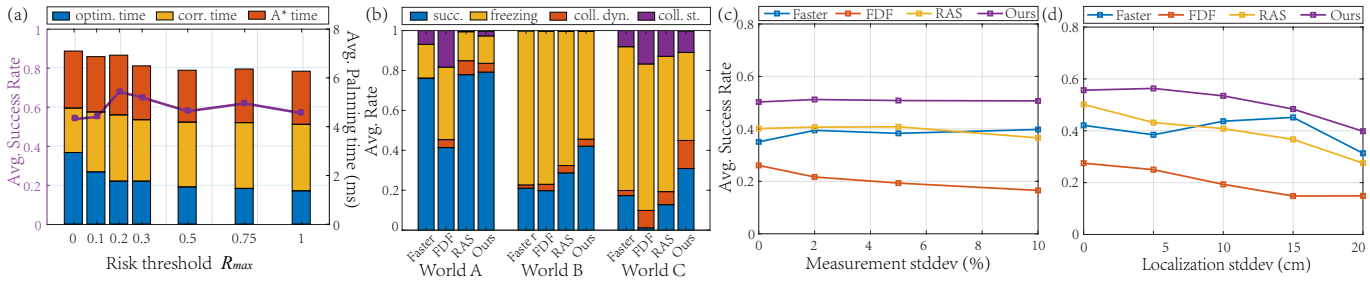

Fig. 8. Risk threshold investigation and performance comparison between our proposed method and Faster [32], FDF [6], and RAS [9]. In (a), the bar plot shows the average planning time with different risk thresholds and the line plot shows the average success rate. Subplot (b) shows the average success rate, freezing rate, collision rate to dynamic obstacles, and collision rate to static obstacles of different methods in different simulation worlds. Subplot (c) and (d) present the average success rate of different methods at different measurement noise level and localization noise level, respectively.

rate decreases for all methods. In a simple scenario World A, Faster, RAS, and ours have similar success rates. In more complex scenarios World B and World C, our method has a distinct advantage over other methods. Specifically, the success rate of our method is increased by 0.13 and 0.14, respectively, in World B and C, compared to the second-best method. Due to the consideration of uncertainties, our method searches for trajectories prudently, thus, its freezing rate is usually higher but the collision rate is lower.

Fig. 8 (c) shows the average success rate in all the simulation worlds at different measurement noise levels. Raising the measurement noise only has a significant influence on the FDF method [6], whose success rate drops by 0.10 when the measurement increases from 0 to 10%. In comparison, Faster, RAS, and our method are robust to the measurement noise. Fig. 8 (d) presents the average success rate in all the simulation worlds at different localization noise levels. The localization noise has a larger influence than the measurement noise. When S.D. increases from 0 to 5 cm, the success rate of FDF and RAS decrease by 0.03 and 0.07, respectively, while our success rate has no decrease. When S.D. increases to 15 cm, the success rate of FDF and RAS decrease by 0.13 and 0.14, respectively, while our success rate decreases by 0.07. The success rate of Faster grows when S.D. increases from 5 cm to 15 cm because the mapping module of Faster is prone to regard all noise points as obstacles, and thus, the trajectories are more conservative when the noise grows. However, when the localization S.D. is 20 cm, this conservativeness causes more collision and freezing cases, and the success rate drops again. Overall, our method outperforms the state-of-the-art methods in terms of navigation success rate and collision rate and also shows robustness against measurement and localization noise.

B. Real-world Tests

In the real-world tests, a MAV equipped with an NVIDIA Jeston Xavier NX computing board and an Intel Realsense D455 depth camera was used. The measurement noise S.D. of the camera is less than 2% within four meters. The position of the MAV was given by the Optitrack system. We validated the effectiveness of our method in obstacle avoidance tests at two levels of localization noise, whose S.D. was 2 cm and 5 cm. The corridors and trajectories were generated at a frequency of about 20 Hz. To avoid the influence of the delay in trajectory generation, a control command buffer (composed of position, velocity and acceleration commands from previously planned trajectory) was maintained, and the new trajectory was planned to start from the last element in the buffer unless the trajectory in the buffer was not safe.

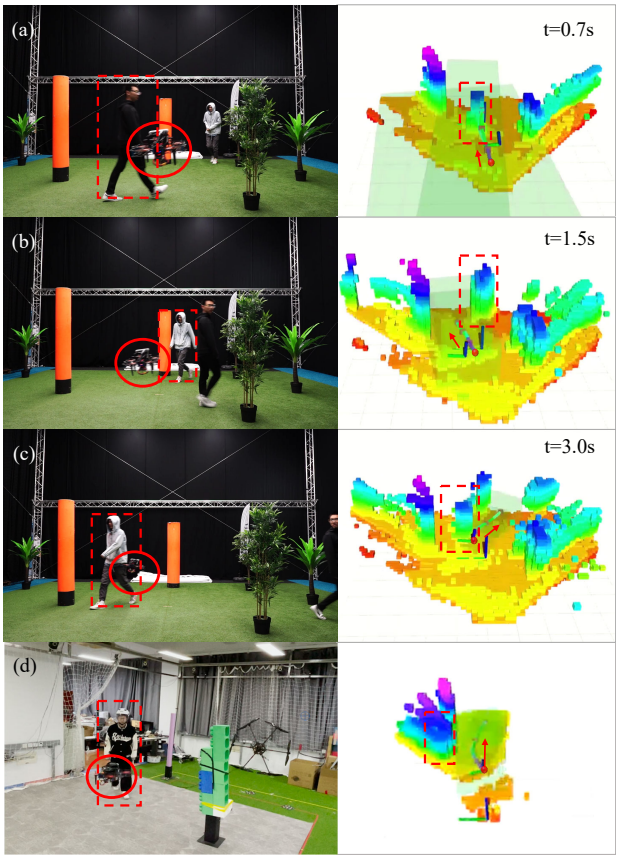

Fig. 9. Snapshots of our MAV flying in dynamic environments. Images on the right side show the risk map, the RAST safety corridors (semi-transparent green cuboids), and the planned trajectory (green curve in the corridors). Current high-risk regions in the map are shown in the voxel form colored by height. The red dashed rectangles outline the pedestrians that MAV is avoiding.

Fig. 9 (a) to (c) show snapshots of an obstacle avoidance test in an environment with two pedestrians and several static obstacles. In (a), a pedestrian is in front of the MAV. But the region occupied by the pedestrian currently is predicted to

be safe in the future because the pedestrian is moving to the right. Fig.9 (b) and Fig.9 (c) present the process in which the MAV avoids the second pedestrian. We conducted 43 tests in total. The success rate is 0.67. The freezing rate is 0.12. The static and dynamic obstacle collision rates are 0.12 and 0.09, respectively. We also conducted the tests with the localization provided by an onboard Intel Realsense T265 tracking camera. A snapshot is shown in Fig.9 (d). More results can be found in the attached video.

VI. Conclusion

This paper presents a risk-aware autonomous navigation method for MAVs in dynamic and uncertain environments. The method represents the local environment with the DSP map and builds RAST safety corridors in the map. MAV measurement and localization uncertainties are considered in the map building and prediction process. Within the RAST safety corridors, flight trajectories are optimized by solving a QP problem. Simulation results show that our method achieves the highest success rate compared to the state-of-the-art methods under different uncertainty levels. Physical experiments also proved the effectiveness of our method in the real world with localization and measurement noise embedded. However, the method is still not robust enough for application tasks. One major reason is that the limited FOV and the occlusion between obstacles interfere with the perception of dynamic obstacles. In future works, we will investigate perception-aware planning to improve the sensing and reacting robustness.

References

[1] S. Tang and V. Kumar, "Autonomous flight," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 1, pp. 29–52, 2018.

[2] 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, pp. 189–206, 2013.

[3] S. Liu, M. Watterson, K. Mohta, K. Sun, S. Bhattacharya, C. J. Taylor, and V. Kumar, "Planning dynamically feasible trajectories for quadrotors using safe flight corridors in 3-d complex environments," *IEEE Robot. Autom. Lett.*, vol. 2, no. 3, pp. 1688–1695, 2017.

[4] C. Richter, A. Bry, and N. Roy, "Polynomial trajectory planning for aggressive quadrotor flight in dense indoor environments," in *Robotics research*. Springer, 2016, pp. 649–666.

[5] F. Gao, W. Wu, Y. Lin, and S. Shen, "Online safe trajectory generation for quadrotors using fast marching method and bernstein basis polynomial," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 344–351.

[6] Y. Wang, J. Ji, Q. Wang, C. Xu, and F. Gao, "Autonomous flights in dynamic environments with onboard vision," in *Proc. of the IEEE/RSJ Intl. Conf. on Intell. Robots and Syst. (IROS)*, 2021, pp. 1966–1973.

[7] 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.

[8] J. Lin, H. Zhu, and J. Alonso-Mora, "Robust vision-based obstacle avoidance for micro aerial vehicles in dynamic environments," in *Proc. of the IEEE Intl. Conf. on Robot. and Autom. (ICRA)*, 2020, pp. 2682–2688.

[9] 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.

[10] J. Tordesillas and J. P. How, "Panther: Perception-aware trajectory planner in dynamic environments," *IEEE Access*, vol. 10, pp. 22 662–22 677, 2022.

[11] G. Chen, W. Dong, P. Peng, J. Alonso-Mora, and X. Zhu, "Continuous occupancy mapping in dynamic environments using particles," *arXiv preprint arXiv:2202.06273*, 2022.

[12] J. Chen, T. Liu, and S. Shen, "Online generation of collision-free trajectories for quadrotor flight in unknown cluttered environments," in *Proc. of the IEEE Intl. Conf. on Robot. and Autom. (ICRA)*. IEEE, 2016, pp. 1476–1483.

[13] F. Gao, L. Wang, B. Zhou, X. Zhou, J. Pan, and S. Shen, "Teach-repeat-replan: A complete and robust system for aggressive flight in complex environments," *IEEE Trans. Robot.*, vol. 36, no. 5, pp. 1526–1545, 2020.

[14] 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 *Proc. of the IEEE Intl. Conf. on Robot. and Autom. (ICRA)*, 2022, pp. 250–256.

[15] V. Usenko, L. V. Stumberg, A. Pangercic, and D. Cremers, "Real-time trajectory replanning for mavs using uniform b-splines and 3d circular buffer," in *Proc. of the IEEE/RSJ Intl. Conf. on Intell. Robots and Syst. (IROS)*. IEEE, 2017, pp. 215–222.

[16] G. Chen, D. Sun, W. Dong, X. Sheng, X. Zhu, and H. Ding, "Computationally efficient trajectory planning for high speed obstacle avoidance of a quadrotor with active sensing," *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 3365–3372, 2021.

[17] F. Sadeghi and S. Levine, "Cadrl: Real single-image flight without a single real image," in *Proc. of Robot.: Sci. and Syst. (RSS)*, 2017.

[18] G. Chen, H. Yu, W. Dong, X. Sheng, X. Zhu, and H. Ding, "What should be the input: Investigating the environment representations in sim-to-real transfer for navigation tasks," *Robotics and Autonomous Systems*, vol. 153, p. 104081, 2022.

[19] D. Dugas, J. Nieto, R. Siegwart, and J. J. Chung, "Navrep: Unsupervised representations for reinforcement learning of robot navigation in dynamic human environments," in *Proc. of the IEEE Intl. Conf. on Robot. and Autom. (ICRA)*, 2021, pp. 7829–7835.

[20] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," in *Proc. of the IEEE Intl. Conf. on Robot. and Autom. (ICRA)*. IEEE, 2011, pp. 2520–2525.

[21] 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, pp. 189–206, 2013.

[22] D. Harabor and A. Grastien, "Online graph pruning for pathfinding on grid maps," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 25, no. 1, 2011, pp. 1114–1119.

[23] J. Tordesillas and J. P. How, "Mader: Trajectory planner in multiagent and dynamic environments," *IEEE Trans. Robot.*, vol. 38, no. 1, pp. 463–476, 2022.

[24] B. Li, Y. Ouyang, L. Li, and Y. Zhang, "Autonomous driving on curvy roads without reliance on frenet frame: A cartesian-based trajectory planning method," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 9, pp. 15 729–15 741, 2022.

[25] W. Ding, L. Zhang, J. Chen, and S. Shen, "Safe trajectory generation for complex urban environments using spatio-temporal semantic corridor," *IEEE Robot. Autom. Lett.*, vol. 4, no. 3, pp. 2997–3004, 2019.

[26] J. Li, X. Xie, H. Ma, X. Liu, and J. He, "Speed planning using bezier polynomials with trapezoidal corridors," *arXiv preprint arXiv:2104.11655*, 2021.

[27] W. Xu, Q. Wang, and J. M. Dolan, "Autonomous vehicle motion planning via recurrent spline optimization," in *Proc. of the IEEE Intl. Conf. on Robot. and Autom. (ICRA)*, 2021, pp. 7730–7736.

[28] R. P. S. Mahler, *Statistical Multisource-Multitarget Information Fusion*. Artech House, Inc., 2007.

[29] B. N. Vo, S. Singh, and A. Doucet, "Sequential monte carlo methods for multi-target filtering with random finite sets," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 41, no. 4, pp. 1224–1245, 2005.

[30] C. Schöller, V. Aravantinos, F. Lay, and A. Knoll, "What the constant velocity model can teach us about pedestrian motion prediction," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 1696–1703, 2020.

[31] B. Zhou, F. Gao, L. Wang, C. Liu, and S. Shen, "Robust and efficient quadrotor trajectory generation for fast autonomous flight," *IEEE Robot. Autom. Lett.*, vol. 4, no. 4, pp. 3529–3536, 2019.

[32] J. Tordesillas, B. T. Lopez, M. Everett, and J. P. How, "Faster: Fast and safe trajectory planner for navigation in unknown environments," *IEEE Trans. Robot.*, vol. 38, no. 2, pp. 922–938, 2022.

[33] G. Chen, P. Peng, P. Zhang, and W. Dong, "Risk-aware trajectory sampling for quadrotor obstacle avoidance in dynamic environments," *arXiv preprint arXiv:2201.06645*, 2022.