Evaluating Dynamic Environment Difficulty for Obstacle Avoidance Benchmarking
Supplementary Material

Moji Shi, Gang Chen, Álvaro Serra Gómez, Siyuan Wu, and Javier Alonso-Mora

I. Metrics Evaluation

This appendix section contains details of calculation for metrics evaluation. Assuming we have a map pool containing n different maps, each denoted by M_i. The map pool can be represented as $\mathscr{M} = \{M_1, M_2, ..., M_n\}$. m different planners are tested on these maps. The map difficulty metric is defined as a function D that maps a map M_i to a real number $\mathrm{D}(M_i)$ and the success rate of a planner j on map M_i is defined as SR_{ij}. We can introduce two quantitative indicators for evaluating representativeness:

Spearman's Rank Correlation Coefficient(SRCC): This metric measures the monotonic relationship between two variables. Principally, a good metric $\mathrm{D}(M_i)$ should show a good monotonical relationship with the success rate SR_{ij} for each planner j. The Spearman's Rank Correlation Coefficient $SRCC_j$ for planner j is defined as:

$$SRCC_j = 1 - \frac{6\sum_{i=1}^{n}(\mathrm{R}(\mathrm{D}(M_i)) - \mathrm{R}(SR_{ij}))^2}{n(n^2-1)} \tag{1}$$

where the $\mathrm{R}(\mathrm{D}(M_i))$ and $\mathrm{R}(SR_{ij})$ are the rank of $\mathrm{D}(M_i)$ and SR_{ij} in all the sampled data respectively. The $SRCC_j$ is between -1 and 1. The larger the $|SRCC_j|$ is, the more monotonically related the $\mathrm{D}(M_i)$ and SR_{ij} are. Since a good difficulty metric should have this monotonical relationship with the success rate for every planner, we will use the average of $SRCC_j$ as the representativeness of the map difficulty metric $\mathrm{D}(M_i)$:

$$SRCC = \frac{1}{m}\sum_{j=1}^{m} SRCC_j \tag{2}$$

Coefficient of Variation(CV): This metric measures the variation of the success rate SR_{ij} under the same map difficulty level. We want to know whether the performance of one specific planner under different maps with the same difficulty level is stable. The maps are made into groups according to the map difficulty metric. For example, if the range of the function D is $[0, 10]$, we will divide the maps into ten groups:

$$\mathscr{M}_k = \{M_i | \mathrm{D}(M_i) \in [k, k+1]\} \tag{3}$$

For each group $\mathscr{M}_k$, we can calculate the Coefficient of Variation CV_{jk} for planner j:

$$CV_{jk} = \frac{\sigma_{jk}}{\mu_{jk}} \tag{4}$$

where σ_{jk} is the standard deviation of the success rate SR_{ij} in group $\mathscr{M}_k$ and μ_{jk} is the mean of the success rate SR_{ij} in group $\mathscr{M}_k$. The CV_{jk} is between 0 and 1. The smaller the CV_{jk} is, the more stable the performance of planner j is under the same map difficulty level. Since we want to have this stability for every planner, we will use the average of CV_{jk} as the representativeness of the map difficulty metric D_i:

$$CV = \frac{1}{m}\sum_{j=1}^{m} CV_{jk} \tag{5}$$

II. Trajectory Planning Algorithms

This appendix contains details of the algorithms used to generate the trajectories for the UAV including the global motion primitives [1], local motion primitives [2], and MPC [3]. All these trajectory planners share the format of inputs and outputs as shown in Table I.

Input Parameter	Symbol
Target Position	$\vec{p}_t = (x_t, y_t)$
Start Position	$\vec{p}_0 = (x_0, y_0)$
Start Velocity	$\vec{v}_0$
Occupancy Map	$\mathscr{M} = \{m_{ij}\}$, $m_{ij} = \begin{cases} 1 & \text{if not occupied} \\ 0 & \text{if unexplored} \\ -1 & \text{if occupied} \end{cases}$
Active Trackers	$\mathscr{T} = \{\vec{t}_i\}$, $\vec{t}_i = \{\vec{p}_i, \vec{v}_i, \Sigma_{\vec{p}_i,\vec{v}_i}\}$

(a) Input parameters of the trajectory planners

Output Results	Symbol
Future Trajectory	$\mathscr{F} = \{\vec{p}_i, \vec{v}_i, \vec{a}_i\}$

(b) Output results of the trajectory planners

Constraint Parameters	Symbol
Maximum MAV Velocity	v_{max}
Maximum MAV Acceleration	a_{max}
MAV Radius	r_u

(c) Other constraint parameters of the trajectory planners

TABLE I: Inputs, outputs, and other parameters of constraints of the trajectory planners

A. Global Motion Primitives

In Global Motion Primitives, the trajectories are decomposed into several motion primitives, each with constant acceleration. The duration of all primitives is T. The connection between two trajectories is defined as a node with the position and velocity of the end of the first trajectory and the start of the second trajectory. The Global Motion Primitives algorithm tries to expand a graph from the starting node to the goal node:

Algorithm 1 Global Motion Primitives Algorithm

```
Input: Starting position $\vec{p}_0$, starting velocity $\vec{v}_0$, target position $\vec{p}_\mathrm{t}$, duration $T$, sampled accelerations $\mathscr{A}$
Output: Path from start to target
Initialize open list $\mathscr{O}$ with node $N(\vec{p}_0, \vec{v}_0, 0)$
Initialize closed list $\mathscr{C} = \emptyset$
while $\mathscr{O}$ is not empty do
    $N_\mathrm{current}$ = node from $\mathscr{O}$ with the lowest cost
    Move $N_\mathrm{current}$ from $\mathscr{O}$ to $\mathscr{C}$
    if $N_\mathrm{current}.position$ is close to $\vec{p}_\mathrm{target}$ then
        return reconstructed path from $N(\vec{p}_0, \vec{v}_0, 0)$ to $N_\mathrm{current}$, add to $\mathscr{F}$
    end if
    for each $\vec{a}$ in $\mathscr{A}$ do
        Generate trajectory $\tau$ using $\vec{a}$ over duration $T$ starting from $N_\mathrm{current}.position$ and $N_\mathrm{current}.velocity$
        $\vec{p}_\mathrm{new}$, $\vec{v}_\mathrm{new}$ = endpoint of $\tau$
        $\mathrm{cost}_\mathrm{new}$ = cost of $N_\mathrm{current}$ + cost of $\tau$
        if $\tau$ is collision-free and $N(\vec{p}_\mathrm{new}, \vec{v}_\mathrm{new}, \mathrm{cost}_\mathrm{new})$ is not in $\mathscr{C}$ then
            Add $N(\vec{p}_\mathrm{new}, \vec{v}_\mathrm{new}, \mathrm{cost}_\mathrm{new})$ to $\mathscr{O}$
        end if
    end for
end while
return no path found
```

The heuristic cost of each node is derived from the time-optimal control problem:

$$C = \frac{12\left\|\vec{p}_t - \vec{p}\right\|^2}{T^3} - \frac{12\vec{v}\cdot(\vec{p}_t - \vec{p})}{T^2} + \frac{4\left\|\vec{v}\right\|^2}{T} + \rho T \qquad (6)$$

where $\vec{p}$ and $\vec{v}$ denote the position and velocity of the node. The MAV will follow the trajectory after getting the future trajectory $\mathscr{F}$. At each timestep, the MAV will execute the collision check based on the updated occupancy map $\mathscr{M}$ and the active trackers $\mathscr{T}$:

$$\|\vec{p} - (\vec{p}_i + k\Delta t\vec{v}_i)\| \geq r_i, \forall t_i \in \mathscr{T} \qquad (7)$$

$$m(\vec{p}_i) \neq -1 \qquad (8)$$

If every point on the future trajectory is collision-free, the MAV will execute the trajectory. Otherwise, the MAV will start replanning.

B. Local Motion Primitives

Local Primitives do not generate full trajectories toward the target position in one shot. Instead, they generate a motion primitive from the current position to a temporary target position with minimum heuristic cost. The planning process can be explained in Algorithm 2.

Algorithm 2 Local Primitives with Optimal Jerk Control

```
Input: Initial position $\vec{p}_0$, velocity $\vec{v}_0$, and acceleration $\vec{a}_0$; final acceleration $\vec{a}_f$; duration $T$; time steps $t$
Output: Optimal collision-free trajectory
Generate temporary target list $\mathscr{P}_t$
for $\vec{p}_{tt} \in \mathscr{P}_t$ do
    Generate a trajectory $\tau$ from $\vec{p}_0$ to $\vec{p}_{tt}$ using the optimal jerk control [?]
end for
Sort temporary targets by their heuristic costs $h(\tau)$ in ascending order
for each ranked temporary target do
    If its trajectory is collision-free:
        Return the trajectory
end for
```

The list of temporary targets is generated by sampling around the current position of the MAV with a fixed radius and uniformly sampled angles:

$$\mathscr{P}_t = \left\{ \vec{p}_0 + \begin{bmatrix} r\cos\theta \\ r\sin\theta \end{bmatrix} \mid \theta \in [0, 2\pi) \right\} \qquad (9)$$

C. MPC

In MPC, the state of the MAV is defined as $\vec{x}^k = \begin{bmatrix} \vec{p}^k \\ \vec{v}^k \end{bmatrix}$, where $\vec{p}^k$ and $\vec{v}^k$ are the position and velocity of MAV at time step k. The control input is defined as $\vec{u}^k = \begin{bmatrix} \vec{a}^k \end{bmatrix}$, where $\vec{a}^k$ is the acceleration of MAV at time step k. The dynamics of the MAV can be defined as:

$$\vec{x}^{k+1} = \vec{f}\left(\vec{x}^k, \vec{u}^k\right) = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \vec{x}^k + \begin{bmatrix} \frac{\Delta t^2}{2} & 0 \\ 0 & \frac{\Delta t^2}{2} \\ \Delta t & 0 \\ 0 & \Delta t \end{bmatrix} \vec{u}^k \qquad (10)$$

where Δt is the time step. The MPC formulation is as follows:

$$\min_{\vec{x}^{1:N}, \vec{u}^{0:N-1}} \sum_{k=0}^{N} J^k\left(\vec{x}^k, \vec{u}^k\right) + J^N\left(\vec{x}^N\right) \qquad (11)$$

$$\text{s.t. } \vec{x}^0 = \vec{x}(0), \quad Vecx^k = \vec{f}\left(\vec{x}^{k-1}, \vec{u}^{k-1}\right), \qquad (12)$$

$$\vec{u}^{k-1} \in \mathscr{U}, \quad \vec{x}^k \in \mathscr{X}, \qquad (13)$$

$$\forall k \in \{1, \ldots, N\}. \qquad (14)$$

where J^k is the cost function at time step k, $\mathscr{U}$ and $\mathscr{X}$ are the control input and state constraints. For each tracked dynamic obstacles $t_i \in \mathscr{T}$, the collision constraint is defined as:

$$\left\| \vec{p}^k - (\vec{p}_i + k\Delta t\vec{v}_i) \right\| \geq r_i \qquad (15)$$

Where the constant velocity model is used to estimate the future position of the obstacle, the cost function is defined as:

$$J^k\left(\vec{x}^k, \vec{u}^k\right) = \vec{x}^{k\top} Q \vec{x}^k + \vec{u}^{k\top} R \vec{u}^k \qquad (16)$$

where Q and R are the weight matrices that can be tuned to achieve different performance.

III. Gaze Planning Algorithm

This appendix section explains the details of the gaze planners which update the MAV ψ_r yaw angle. It takes the MAV's current state and observation, then outputs the yaw angle speed $\dot{\psi}_r$ for the next time step. The yaw angle is updated by $\hat{\psi}_r = \psi_r + \dot{\psi}_r \Delta t$. The direction of the updated yaw angle is denoted as $\vec{d}_r = \begin{bmatrix}\cos(\psi_r + \dot{\psi}_r \Delta t) & \sin(\psi_r + \dot{\psi}_r \Delta t)\end{bmatrix}$.Multiple gaze planners are implemented in this work, including the following:

Gaze Planner	Reference
LookAhead	[4], [5]
LookGoal	[6], [7]
Rotating	-
Finean et al.	[8]
Owl	[2]

TABLE II: Reproduced trajectory planners.

A. LookAhead

LookAhead planner is a simple planner that aligns the heading of the MAV to the direction of current velocity $\vec{d}_v = \begin{bmatrix}\dot{x}_r & \dot{y}_r\end{bmatrix}$. It can be formulated as the following optimization problem:

$$\min_{\dot{\psi}_r} \quad \arccos\left(\frac{\vec{d}_r \cdot \vec{d}_v}{\left\|\vec{d}_r\right\| \left\|\vec{d}_v\right\|}\right) \tag{17}$$

$$\text{s.t.} \quad \dot{\psi}_r \in [-\dot{\psi}_{max}, \dot{\psi}_{max}] \tag{18}$$

B. LookGoal

LookGoal planner aligns the heading of the MAV to the direction of the target position $\vec{d}_t = \begin{bmatrix}x_t - x_r & y_t - y_r\end{bmatrix}$. It can be formulated as the following optimization problem:

$$\min_{\dot{\psi}_r} \quad \arccos\left(\frac{\vec{d}_r \cdot \vec{d}_t}{\left\|\vec{d}_r\right\| \left\|\vec{d}_t\right\|}\right) \tag{19}$$

$$\text{s.t.} \quad \dot{\psi}_r \in [-\dot{\psi}_{max}, \dot{\psi}_{max}] \tag{20}$$

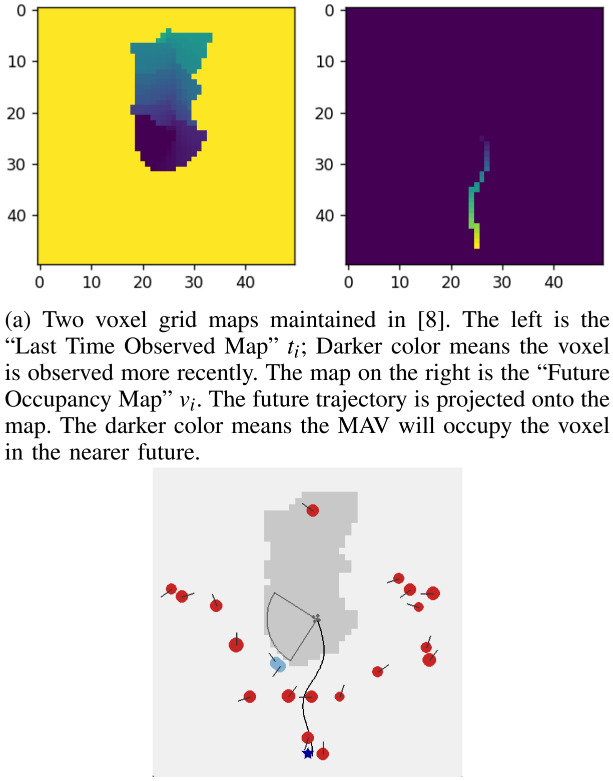
C. Finean et al. [8]

Finean et al. [8] is an optimization-based gaze planning method. Two voxel grid maps are maintained. One is the "Last Time Observed Map" t_i, representing when the voxel i is last observed. The other is the "Future Occupancy Map" v_i, which represents whether the MAV will occupy the voxel i in the future. Each voxel is assigned the value:

$$v_i = \begin{cases} 0 & \text{if the future trajectory does not occupy the voxel} \\ \tau & \text{if the future trajectory occupies the voxel at time step } \tau \end{cases} \tag{21}$$

One example of the two grid maps at one moment is shown in Figure 1. With the two maps, the planner calculates the reward for each voxel i as:

$$r_i = \begin{cases} c_1 & 0 < v_i \leq \tau_s \quad \text{and} \quad t_i \geq \tau_c \\ c_2 & v_i > \tau_s \quad \text{and} \quad t_i \geq \tau_c \\ \max(c_3 t_i, 1) & 0 < \text{otherwise} \end{cases} \tag{22}$$



(a) Two voxel grid maps maintained in [8]. The left is the "Last Time Observed Map" t_i; Darker color means the voxel is observed more recently. The map on the right is the "Future Occupancy Map" v_i. The future trajectory is projected onto the map. The darker color means the MAV will occupy the voxel in the nearer future.

(b) The environment where the two maps above are generated.

Fig. 1: Example of Oxford planner. One example of the two grid maps in the cost function at one moment is shown.

where τ_s, τ_c, c_1, c_2, and c_3 are hyperparameters. The planner then selects the yaw angle velocity $\dot{\psi}_r$ that maximizes the reward in the FOV:

$$\dot{\psi}_r = \arg\max_{\dot{\psi}_r} \sum_{i \in FOV(\vec{d}_r)} r_i \tag{23}$$

D. Owl

Owl planner is proposed in [2]. It also considers gaze planning as an optimization problem. Compared to the Oxford planner, the Owl planner considers more factors in a dynamic environment. Four directions are prioritized in four cost functions f_1 to f_4: the direction of the target position, the direction of the current velocity, the direction of the observed dynamic obstacles, and the direction that has not been updated for a period. The last cost function f_5 is defined so that the yaw angle velocity $\dot{\psi}_r$ is not too large. Finally, the gaze planning problem is formulated as a multi-objective optimization problem:

$$\min_{\dot{\psi}_r} \quad \sum_{i=1}^{5} \lambda_i f_i(\vec{d}_r) \tag{24}$$

$$\text{s.t.} \quad \dot{\psi}_r \in [-\dot{\psi}_{max}, \dot{\psi}_{max}] \tag{25}$$

IV. Experiment Results on *Dataset II*

This appendix section presents details of the experiment results of survivability metric in *Dataset II*. Specifically, there will be three different categories of maps in *Dataset II*:

1) the environment of obstacles with various velocities
2) the environment of obstacles with various sizes
3) the environment of obstacles with a different motion profile RVO [9]

A. Environment of obstacles with various velocities

To test the survivability metric on the environment of various obstacles of various velocities, we generate 45 maps in which the obstacle number is chosen from $\{10, 20, 30\}$, the obstacle size is chosen from $\{0.5, 1.0, 1.5\}m$, and the obstacle velocity is randomly sampled from $[2, 6]m/s$. The survivability metric is computed for each map. The scatter plot of the survivability metric and the success rate is shown in Figure 2.

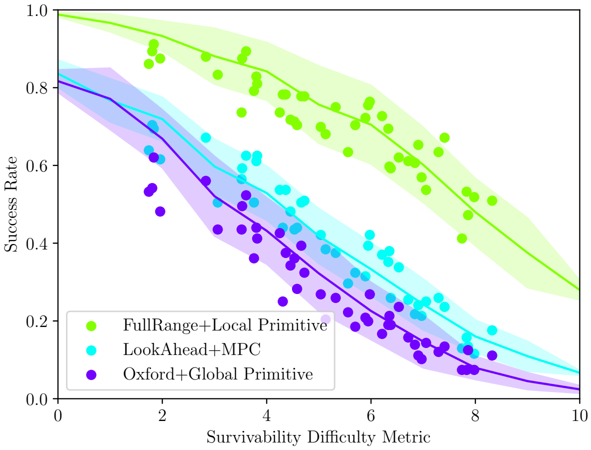


Fig. 2: Scatter plot of the survivability metric and the success rate on the environment of obstacles with various velocities. The curves are fitted curves from *Dataset I*. The scatters are the survivability metric and the success rate on *Dataset II*.

Here three different planners are tested on *Dataset II*. The *FullRange+Local Primitive* and *Oxford+Global Primitive* are chosen for having the best and worst performance on *Dataset I*. The *LookAhead+MPC* is chosen for having a different trajectory and gaze planner and increasing diversity. The results of the proportion of data points within one standard deviation, two standard deviations, and three standard deviations from the Gaussian distribution fitted in *Dataset I* are shown in Table III.

Map Dataset	Within 1 σ	Within 2 σ	Within 3 σ
Dataset I	68%	95%	99.7%
Dataset II (Type 1)	76.3%	98.5%	100%

TABLE III: The percentage of *Dataset II* data points(with various obstacle velocities) within one standard deviation, two standard deviations, and three standard deviations from the Gaussian distribution fitted in *Dataset I*.

B. Environment of obstacles with various sizes

To test the survivability metric on the environment of various obstacles of various sizes, we generate 45 maps in which the obstacle number is chosen from $\{10, 20, 30\}$, the obstacle velocity is chosen from $\{2, 4, 6\}m/s$(All obstacles in one map have the same velocity), and the obstacle sizes are randomly sampled from $[0.5, 1.5]m$. The survivability metric is computed for each map. The scatter plot of the survivability metric and the success rate is shown in Figure 3. The testing planners are chosen the same as in the previous section.

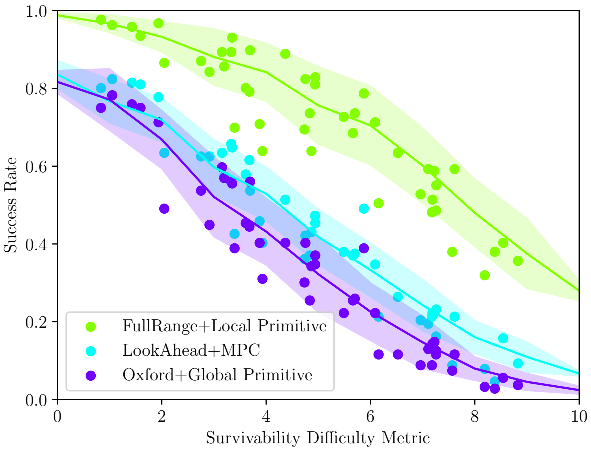


Fig. 3: Scatter Plot of Survivability Metric and Success Rate in Maps with Various Obstacle Sizes. The curve is the fitted curve from *Dataset I*. The scatters are the data from *Dataset II*.

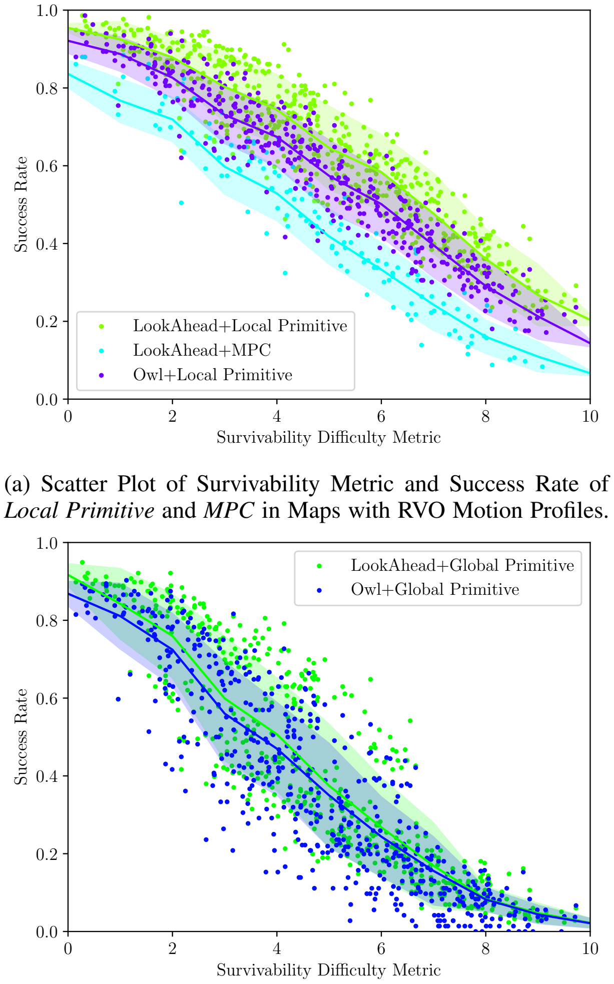
The results of the proportion of data points within one standard deviation, two standard deviations, and three standard deviations from the Gaussian distribution fitted in *Dataset I* are shown in Table IV.

Map Dataset	Within 1 σ	Within 2 σ	Within 3 σ
Dataset I	68%	95%	99.7%
Dataset II (Type 2)	71.1%	94.8%	100%

TABLE IV: The percentage of *Dataset II* data points(with various obstacle sizes) within one standard deviation, two standard deviations, and three standard deviations from the Gaussian distribution fitted in *Dataset I*.

C. Environment of obstacles with RVO motion profiles

To test the survivability metric on the environment of obstacles with RVO motion profiles, we generate 45 maps in which the obstacle number is chosen from $\{10, 20, 30\}$, the obstacle velocity is chosen from $\{2, 4, 6\}m/s$(All obstacles in one map have the same velocity), and the obstacle sizes are randomly sampled from $[0.5, 1.5]m$. The survivability metric is computed for each map. The scatter plot of the survivability metric and the success rate is shown in Figure 4. The testing planners are chosen the same as in the previous section.



(a) Scatter Plot of Survivability Metric and Success Rate of *Local Primitive* and *MPC* in Maps with RVO Motion Profiles.

(b) Scatter Plot of Survivability Metric and Success Rate of *Global Primitive* in Maps with RVO Motion Profiles.

Fig. 4: The environment of obstacles with RVO motion profiles.

The results of the proportion of data points within one standard deviation, two standard deviations, and three standard deviations from the Gaussian distribution fitted in *Dataset I* are shown in Table V.

Type of Map	**Within 1 σ**	**Within 2 σ**	**Within 3 σ**
Controlled	68%	95%	99.7%
Uncontrolled (RVO)	63.7%	92.5%	98.0%

TABLE V: The percentage of *Dataset II* data points(with RVO motion profile) within one standard deviation, two standard deviations, and three standard deviations from the Gaussian distribution fitted in *Dataset I*.

References

[1] S. Liu, N. Atanasov, K. Mohta, and V. Kumar, “Search-based motion planning for quadrotors using linear quadratic minimum time control,” in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 2872–2879.

[2] 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.

[3] 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.

[4] C. Lim, B. Li, E. M. Ng, X. Liu, and K. H. Low, “Three-dimensional (3D) Dynamic Obstacle Perception in a Detect-and-Avoid Framework for Unmanned Aerial Vehicles,” in *2019 International Conference on Unmanned Aircraft Systems (ICUAS)*, pp. 996–1004.

[5] B. Zhou, J. Pan, F. Gao, and S. Shen, “Raptor: Robust and perception-aware trajectory replanning for quadrotor fast flight,” *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 1992–2009, 2021.

[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, 2021.

[7] A. Patel, B. Lindqvist, C. Kanellakis, and G. Nikolakopoulos, “Fast Planner for MAV Navigation in Unknown Environments Based on Adaptive Search of Safe Look-Ahead Poses,” in *2022 30th Mediterranean Conference on Control and Automation (MED)*.

[8] M. N. Finean, W. Merkt, and I. Havoutis, “Where should i look? optimized gaze control for whole-body collision avoidance in dynamic environments,” *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 1095–1102, 2021.

[9] J. van den Berg, Ming Lin, and D. Manocha, “Reciprocal Velocity Obstacles for real-time multi-agent navigation,” in *2008 IEEE International Conference on Robotics and Automation*, pp. 1928–1935.