CSCI 5511 - Final Project

Srijan Kumar Pal (pal00036@umn.edu)

Codebase: https://github.com/srijanpal07/3D-Path-Planning-in-Dynamic-Environment.git

Problem description: This project focuses on 3D path planning in environments containing both static and dynamic obstacles, using a voxelized 3D grid representation and interactive visualizations. The goal is to implement a planning algorithm extended to dynamic environments that require replanning under sensing. As a baseline, I implemented offline A* planning with two neighborhood configurations, where each node has either 6 or 26 neighbors in 3D space. In this setting, the planner has full knowledge of the environment (dynamic and static obstacles) and computes a path only once at the start. Building on this baseline, I implemented an online A*-based replanning approach that operates under local sensing constraints. The planner initially plans a path using only known static obstacles. During execution, if any obstacle enters the probe's sensor radius, it starts replanning.

The visualization shows a probe (purple sphere) moving along the planned trajectory (magenta). The probe's sensing region is illustrated using concentric spheres centered on the probe, which is used only in the online planning. In the environment, static obstacles are shown in blue, while dynamic obstacles are shown in orange.

All components of the system were implemented from scratch by me, including the planners, environment representation, online replanning logic, visualization tools, metric computation, and configurable experiment parameters. The core planning and visualization logic can be found in the examples/ and planners/ directories of the codebase.

Literature review: This project is formulated from the perspective of a drone navigating an urban environment with limited onboard sensing, where the map of static obstacles is known, and dynamic obstacles are unknown. Several works have studied precomputed or offline path planning using A* in static environments [1], [2]. However, fewer works focus on online replanning with unknown dynamic obstacles, especially under local sensing constraints. Some related studies consider online planning where the edge traversal cost is influenced by wind or environmental factors [3]. More general formulations of planning and replanning in dynamic environments have been explored in prior work [4], [5], [6] along with existing codebases [7], [8], [9], which served as inspiration for the structure and design of this project.

Software required: Python 3.10, numpy 2.2.6, plotly 6.3.1, and any browser.

Instructions on how to run the code:

More details on how to run the code can be found in the README.md.

Step 1: Clone the repository:

```Shell
git clone
https://github.com/srijanpal07/3D-Path-Planning-in-Dynamic-Environment.git
```

Step 2: Set up a virtual environment and install dependencies:

```shell
cd 3D-Path-Planning-in-Dynamic-Environment/
python3 -m venv .venv
source .venv/bin/activate
pip install -r requirements.txt
```

Step 3: Run the main planning and visualization script:

```shell
python -m examples.plan_and_visualize --planner online_astar --env
environments/env3.json --sensor-radius 0.5
```

Running the above command generates two outputs - an interactive visualization file (.html) showing the environment, obstacles, and planned trajectory, and a metrics file (.json) containing planning time, path length, success status, and number of replans. Both outputs are saved automatically, and the metric values are also printed to the console. The following command-line arguments can be configured -

```
"--env": Path to environment JSON file.
"--outfile": Output HTML file name.
"--planner": Planner type (offline A* or online A*).
"--neighbors": Number of neighbors for A* (26 or 6) used by offline A*.
"--sensor-radius": Sensor radius (m) for detecting obstacles used by online A*.
```

(Optional) To generate a new environment or to change the start and goal location in the existing environment, follow the instructions mentioned in the section below, "**Supporting files required to run the code.**"

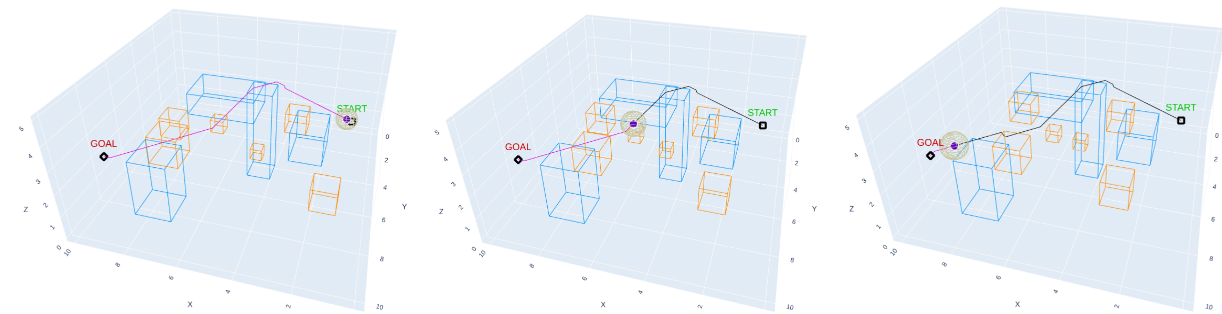
Results: The results generated by all planners across all the environments are stored in the directory (results/), with visualizations in (results/visualizations), and the corresponding metric scores in (results/metric_scores). The visuals could not be included as they are not still images.
Environments: 4 environments are used for evaluation. Environments env1, env2, and env3 contain the same set of obstacles (4 static and 6 dynamic), with only the goal location changing across environments.
The fourth environment (env_complex) includes a larger number of static and dynamic obstacles with varying sizes and positions, making it significantly more challenging.
The following metrics are used for evaluation:

1. Success/failure: Whether a valid path from start to goal is found. If unsuccessful, the reason for failure is recorded.
2. Planning time: For offline A*, this is the one-time planning duration. For online A*, this includes the total time spent across all replanning steps.
3. Path length: The total distance traveled along the planned trajectory.

Additionally, the number of replans performed by the online planner is recorded.

For each run, with a specific planner on an environment, the code generates a metrics file (metrics_{planner}_{env}.json) and a visualization file (viz_{planner}_{env}.html) in the root directory. Some of the generated results are reported below:

Environments	Env1 Start: (1.0, 1.0, 1.0) Goal: (4.5, 3.0, 2.0)		Env2 Start: (1.0, 1.0, 1.0) Goal: (7.2, 6.0, 1.8)		Env3 Start: (1.0, 1.0, 1.0) Goal: (9.0, 9.0, 3.0)	
	Planning Time (s)	Path Length (m)	Planning Time (s)	Path Length (m)	Planning Time (s)	Path Length (m)
Offline A* (6 neighbors)	0.002629	6.0	0.033398	13.50	0.068493	19.50
Offline A* (26 neighbors)	0.010711	5.5480	0.016106	12.3896	0.003851	14.3287
Online A*	0.000722	4.8775	0.002121	8.5594	0.001462	13.2681



Images above show the visualization of an online A* planner in action at 3 time steps. Interestingly, the online planner, even more than 1 replan, takes less time to compute in total than the offline planners, probably because it starts with a much shorter path.

Supporting files required to run the code:

The only required files to run the code are the environment configuration files (.json) located in the environments/ directory. New environments can be generated using:

```shell
python -m examples.generate_env
```

The start and goal positions, along with the environment name, can be modified at the top of generate_env.py under the Configurable Parameters section.

```python
#---------------------- Configurable Parameters ----------------------
START = (1.0, 1.0, 1.0)
GOAL  = (9.0, 9.0, 3.0)
```

The name of the generated environment can be changed by modifying the global constant *VIZ_SAVE_PATH* in the "Configurable Parameters" section. When generate_env.py is executed, the environment file is saved in the environments/ directory using this name. If *SAVE_VIZ* is enabled, the script also produces a visualization of the generated environment with a trajectory planned using the offline A* (default).

Note: D*-Lite-based online planner is under development and is not included in the report.

References:

[1] Rienecker, H., Hildebrand, V. & Pfifer, H. Energy optimal 3D flight path planning for unmanned aerial vehicle in urban environments. CEAS Aeronaut J 14, 621–636 (2023). https://doi.org/10.1007/s13272-023-00666-x

[2] Carola Ebert, Christopher Ruwisch, Julien Weiss, Maarten Uijt De Haag, and Flávio Silvestre. "Trajectory Planning in Windy Urban Environment – a Gappy POD Approach for Wind Field Estimates with Sparse Sensors," AIAA 2022-3757. AIAA AVIATION 2022 Forum. June 2022.

[3] E. A. Pensado, G. F. Carrera, F. V. López, H. G. Jorge, and E. M. Ortega, "Turbulence-Aware UAV Path Planning in Urban Environments," 2024 International Conference on Unmanned Aircraft Systems (ICUAS), Chania - Crete, Greece, 2024, pp. 280-285, doi: 10.1109/ICUAS60882.2024.10556934.

[4] Zammit, C., and Kampen, E.V. (2021). 3D real-time path planning of UAVs in dynamic environments in the presence of uncertainty. AIAA Scitech 2021 Forum.

[5] Karaman, S., & Frazzoli, E. (2011). Sampling-based algorithms for optimal motion planning. IJRR, 30(7), 846–894.

[6] Fiorini, P., & Shiller, Z. (1998). Motion planning in dynamic environments using velocity obstacles. IJRR, 17(7), 760–772.

[7] https://github.com/VBot2410/A_Star_3D.git

[8] https://github.com/danielesartori/3D-grid-path-planning.git

[9] https://github.com/HamzaSaddour/Path-planning-for-Multi-UAV-.git