Institute for Aerospace Studies
UNIVERSITY OF TORONTO

AER1217: Development of Unmanned Aerial Vehicles
Project: Autonomous Drone Geodashing

1 Introduction

Geodashing is a form of geocaching, and is defined according to http://gpsgames.org:

> ... In each game, a large set of waypoints, called dashpoints, from all over the world is posted on the Web. Dashpoint locations are chosen at random by computer, with all the unpredictability that presents ... Then, the race is on to see who can reach the most dashpoints before the deadline...
>
> There is nothing to find at the waypoint. No box. No log book. No prizes...
>
> **Getting there is all the fun.**

The final project is divided into two phases:

- **Phase 1:** In the first phase, you are given a bag file of flight data from the ARDrone Gazebo Simulator. Similar to Lab 3, the dataset contains the ARDrone bottom images, vicon pose, and commands. The bottom camera images contain obstacle indicators and photos of notable landmarks in Toronto. Your task is to determine the size and position of the obstacle indicators and the poses of the landmark photos for the second phase of the project.
  The simulated ARDrone bottom camera has the following intrinsic matrix (no distortion parameters).

$$K = \begin{bmatrix} 604.62 & 0.0 & 320.5 \\ 0.0 & 604.62 & 180.5 \\ 0.0 & 0.0 & 1.0 \end{bmatrix}$$

  The extrinsic transformation matrix from the vehicle body frame (base frame) to the camera frame is

$$T_{CB} = \begin{bmatrix} 0.0 & -1.0 & 0.0 & 0.0 \\ -1.0 & 0.0 & 0.0 & 0.0125 \\ 0.0 & 0.0 & -1.0 & -0.025 \\ 0.0 & 0.0 & 0.0 & 1.0 \end{bmatrix}.$$

  This data can also be found in the `camera_info` topic.
- **Phase 2:** In the second phase, with the information gathered in the first phase, you are to generate a desired optimized path to visit all the landmarks in a predefined sequence, while avoiding obstacles in the environment. The obstacles will be located at the red targets.

Two deliverables are required for the project, your **code** containing all the localization, control, and planning algorithms, and a **final report** documenting your approach and results to be submitted.

2 Loading the Project Simulation

The Gazebo models and project world file for the final project are located in the bitbucket repository, open Terminal and pull from the repo:

```
cd $HOME/aer1217/gazebo
git pull origin master
```

To open the project environment, change the world file in the `world.launch` file from `empty.world` to `project2022.world`, and then launch the file. The `project2022.world` file can be accessed here.

```
roslaunch aer1217_ardrone_simulator ardrone_simulator.launch
```

You should edit the launch file include your position/ROS controller and desired position nodes.

The simulation environment should look like the figure below. **Note that the landmark and obstacle locations in the world file are only for display, and do not correspond to the actual locations in the dataset**.

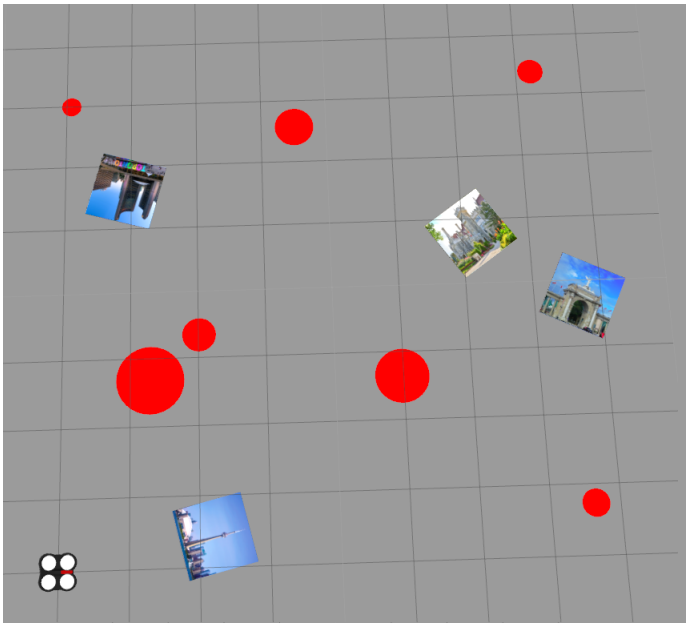

Figure 1: Project Environment in Gazebo

Please let the TAs know if you need help with getting the latest files from the git repositories.

3 Requirements

3.1 Obstacle Indicators

There are **7 obstacle indicators** in the environment which are red circles with **diameter between 0.3 m to 1m**. These red circles correspond to the locations of cylindrical obstacles in phase 2. The diameters of these obstacles are **scaled by a factor of 3** and extend to the maximum environment height (no flying over).

3.2 Landmark Images

There will be 4 images in the environment with size 1m by 1m. These are images of notable landmarks in Toronto:

1. Casa Loma
2. CN Tower
3. Nathan Philips Square
4. Princes' Gates

Figure 2: Landmark Images

The images are provided in the project module for your use. You are required to estimate the position and orientation of all 4 images, with the position being at the image center. You may use the OpenCV library image processing algorithms including any of its image feature detectors.

3.3 Modifying the Simulation Environment

Once you obtain the estimated poses of the obstacle indicators and landmark images, edit the project world file `project2022.world` to add these poses into the simulator.

For the obstacle indicators, starting from line 1583 to 1663. Increase the scale in the x and y direction to 3 times the diameter and set the scale in the z direction to 1000. Then, in the line after `link_frame`, add the pose of the target in the form $(x, y, z, roll, pitch, yaw)$.

For example, if the estimated location of an obstacle indicator is (1.2, 3.4) with 0.4 m diameter. The following lines should be changed:
Line 1595 : `<scale> 1.2 1.2 1000</scale>`
Line 1597 : `<pose frame=''>1.2 3.4 0 0 0 0</pose>`
The figure below shows this change to the example world file.

Figure 3: Example Project Environment with Obstacles

For the landmark images, only the pose needs to be changed. This corresponds to lines 1667, 1677, 1687, and 1697 in the world file.

3.4 Trajectory Planning

With the obstacles and landmarks in place, you will need to plan a collision free trajectory that reaches all 4 landmarks and returns to the starting location. At each landmark, the orientation of the ARDrone should be the orientation of the image **+90 degrees**, such that the bottom camera shows the landmark image upright. The starting location is at **(1,1)**. Your code should accept a **sequence of four numbers** indicating the order of visitation. For example, given the order (3,2,1,4), the ARDrone should traverse in the order: Nathan Philips Square, CN Tower, Casa Loma, Princes' Gates, and finally return to its initial location.

Your controller will be used to follow this trajectory, where the ARDrone must fly fully autonomously without colliding into obstacles. Each team will be required to use either **(1) search-based** or **(2) sampling-based** algorithms for trajectory planning, which will be announced on Quercus before the final project.

3.5 Flight Boundaries

The environment boundaries are $x \in [0, 9]$, $y \in [0, 9]$, and $z \in [0, 3]$. All the obstacles and landmark locations will be within this range. Do not fly outside these boundaries!

4 Final Project Grading

The final project grading is separated into three parts: (1) simulation demonstration (15%), (2) presentation (10%), and project report (25%). The project demonstration is on **Apr. 25th, 2022** in simulation.

The project report is worth 25% and is limited to ten pages. Please submit the code and report in a .zip format, to Blackboard Portal, latest by **April 30, 2022 11:59PM EST**. Grading is based on the following

- **Performance: (10%)**
  - Estimation accuracy of obstacle indicators and landmark poses (5 %)
  - Trajectory planning results with different sequence inputs (5 %)
- **Code: (5%)** All code that was written and modified by your team that was used in the simulation (5%). Code should be commented appropriately.
- **PDF report(10%)** containing the following:
  - Overview of the algorithms and equations used and considered for the final project, as well as the justifications behind your decisions (3.5%).
  - Discussion of the simulation results (3.5%).
  - A brief summary of the general mobile robotics framework. Discussion of the difficulties faced in control, planning and estimation, including what went well or as expected, what did not go well, and what you would do differently if faced with the same problem (3%)

The presentation is worth 10%. The following contents are needed for grading.

- Overview of the final project robotic system. (2%)
- What to do if there is no VICON? (2%)
- What if there are dynamic obstacles? (2%)
- How to improve if it's a racing competition? (2%)
- What to consider if swarming UAVs are going to collaborate? (2%)

You are required to submitted a 10 minute video for the final presentation on Quercus, latest by **April 30, 2022 11:59PM EST**.

5 Good luck!