

Perception and Decision Making in Intelligent Systems

Homework 3: A Robot Navigation Framework

Announce: 11/8 , Deadline: 12/06 23:55

1. Introduction

In the HW2, we successfully reconstructed a 3D semantic map of **apartment_0**. Our next goal is to have a robot move to a desired location (for example, **navigate the robot to find a specific item**). Therefore, in the third homework, we focus on how to navigate from point A to B on the **first floor of apartment_0** using the RRT algorithm.

There are three main tasks, i.e., **2D semantic map construction, RRT algorithm implementation, robot navigation**

1. 2D semantic map construction:

In the second homework, we already have a semantic point cloud.
We need a 2D semantic map for navigation.

2. RRT algorithm implementation:

We use the RRT algorithm to find a navigable path from starting point A to goal point B (a specific item).

3. Robot navigation:

An agent can navigate automatically by following the path calculated by RRT on the **first floor of apartment_0** to find specific items.

2. Requirements

- OS : ubuntu (recommend)
- Python 3.x
- opencv-contrib-python
- Open3d
- matplotlib
- Habitat sim and Habitat lab

3. Tasks

Part 1: 2D semantic map construction

The following describes the steps for generating a 2D semantic map.

1. Remove the ceiling and floor points of the point cloud constructed in HW2
2. Save coordinates and colors of the points (the color should be the same as HW2) [color map for 101 categories](#)
3. Use any libraries (such as matplotlib) to plot a scatter graph (x-coordinate, z-coordinate in the point clouds) with the points you saved
4. Save the map as “**map.png**”

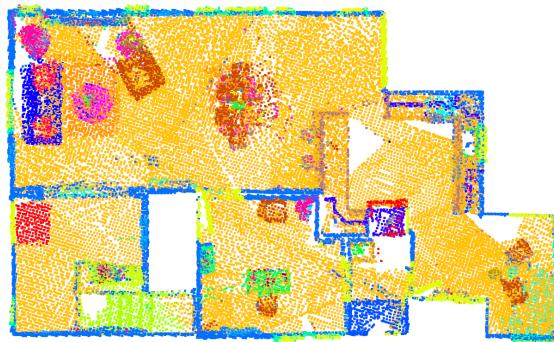
We provide a semantic 3d pointcloud of the **first floor in apartment_0 for your reference**. You can download it from this [link](#). Definitely, please **free to use your own reconstructed map from HW2**.

Notes for provided semantic 3d pointcloud :

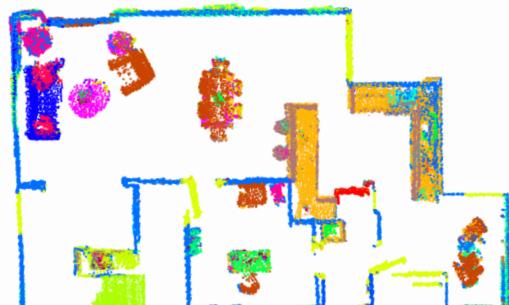
- **Scale relationship** between coordinates in **point.npy** and coordinates in **apartment_0** is:
apartment_0 = points * 10000. / 255.
- There are two versions of the color array, the value of rgb in **color_01.npy** is in range [0, 1] while **color_0255.npy** is in range [0, 255]. If you have problems using **color_01.npy** to find labels because of floating-point errors, you can use **color_0255.npy**.

Note: You need to find the coordinate relationship between the map (pixel) and the coordinates on the **first floor of apartment_0**. (Part 3)
(Feel free to add extra points to calculate the scale and create different map may help you do the job)

Example map:



[After removing ceiling]



[Removing ceiling and floor]

Part 2: RRT Algorithm

You will implement the RRT algorithm, which calculates a navigable path from the “[map.png](#)”. For more detail of the algorithm, please check details on this link [RRT](#) or the slide on the class

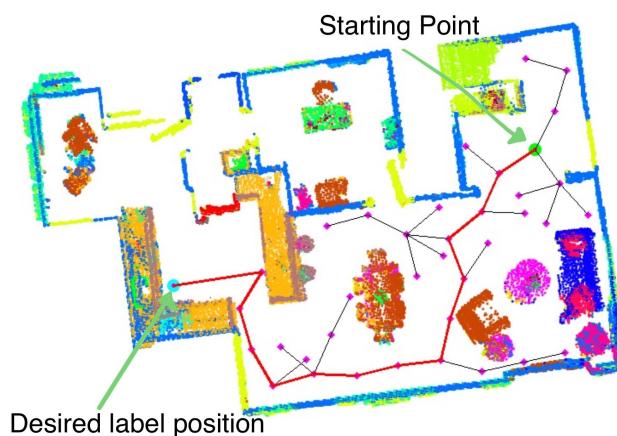
Input:

1. A target category you want to search. Just input a string (our target name ex: **rack**)
 - Check the colors of the corresponding categories from [color map for 101 categories](#)
Our target categories are refrigerator, rack, cushion, lamp, and cooktop
 - In this homework, you only need to search for designated categories
2. Choose a **starting point** on the map

Output:

1. A map with a **path** from a starting point to a target point. Note that the target point is a point in front of the target item. It is up to you to decide the location of the point.
2. Path points on this route
 - The xy-coordinate calculated by RRT is in the pixel coordinate. To navigate in Habitat, you need to convert them into the coordinate of the first floor of `apartment_0`

Example of a RRT output:



Part 3: Robot navigation

- The xy-coordinate calculated by RRT is in the pixel coordinate. We need to convert them from pixel coordinate to xyz-coordinate on the first floor of `apartment_0` in Habitat
- we can define the amount of how much the agent should move and turn for each step (By adding the following lines in function: `make_simple_cfg`)

```
# Here you can specify the amount of displacement in a forward action and the turn angle
agent_cfg.action_space = {
    "move_forward": habitat_sim.agent.ActionSpec(
        "move_forward", habitat_sim.agent.ActuationSpec(amount = 0.25)
    ),
    "turn_left": habitat_sim.agent.ActionSpec(
        "turn_left", habitat_sim.agent.ActuationSpec(amount = 10.0)
    ),
    "turn_right": habitat_sim.agent.ActionSpec(
        "turn_right", habitat_sim.agent.ActuationSpec(amount = 10.0)
    ),
}
```

- Based on those path points, let the agent move using these three commands : “`move_foward`”, “`turn_left`”, “`turn_right`”

- Use the [`load.py`](#) to navigate (please feel free to make any modifications to the sample code)
- Collect RGB images and save them as a video (or gif)
“{target_name}.mp4” (ex: refrigerator.mp4)

Note: In the video, please **highlight** the target with a **transparent mask** while navigating, so that we can clearly visualize the target category.

For instance, we search for “refrigerator”.



4. Examples

[Example Demo Video](#)

5. Submission

A. File Format:

StudentID_Name_hw3

| --- Readme (Explain how to run your code)

| --- report.pdf

| --- src

| | --- your code

| --- results

| | --- refrigerator.mp4 or refrigerator.gif

| | --- rack.mp4 or rack.gif

| | --- cushion.mp4 or cushion.gif

| | --- lamp.mp4 or lamp.gif

| | --- cooktop.mp4 or cooktop.gif

- B. Please put all the files into the folder **StudentID_Name_hw3** and compress your folder into **one “zip” file**. Submit the file to New E3 System with the naming format **StudentID_Name_hw3.zip**
- C. The **deadline** for this homework is **12/06 23:55**.
- D. Late submission leads to **-20 points per day**.
- E. **Wrong submission format leads to -10 point**

6. Grading

1. Online Questions and Demo (50%)

Note : We will specify a target label and a starting point, you should show us the corresponding navigation result on the first floor of apartment _0

2. Report (50%)

i. Code

1. **Detailed** explanation of your implementation

For example:

a. how do you do RRT algorithm

b. how do you convert route to discrete actions

ii. Results and Discussion

1. Show and discuss the results from the RRT algorithm with **different** start points and targets.

2. Discuss about the robot navigation results

3. Anything you want to discuss

iii. Reference

Any reference you take

3. Bonus (10%)

Try to improve the RRT algorithm, compare and discuss it with the original one.