

COMP417

Assignment 2

Posted: Friday, Oct 20, 2017

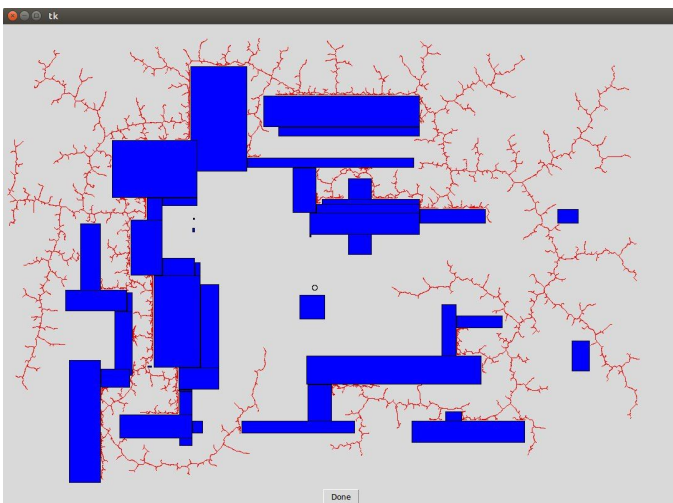
Due: Tuesday, Oct 31 at 12:00 pm, 2017

Rapidly-exploring Random Tree:

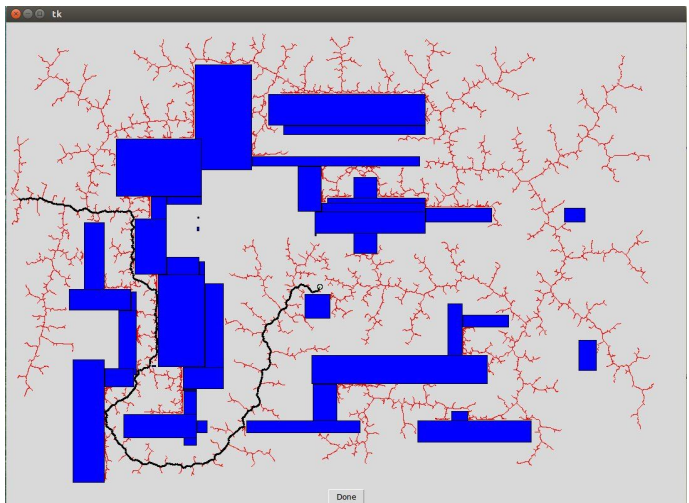
Implement the RRT algorithm for three different kinds of robots on a 2D plane. You are given starter code that implements some of the RRT functionality. You need to modify multiple functions which are annotated with TODOs in the file `rrt_planner.py`. You do not need to make any modifications in any other file. But it is recommended that you read the full code to understand it. Once you are done implementing the required functionality you can run this file as follows:

```
cd <path_to_rrt_directory>  
./rrt_planner.py example_world.png
```

You will see a GUI window opens up and displays your RRT planner steps as shown in the figures below:



(a) Planning Phase



(b) Final path to destination

Part 1 : A simple point robot with freedom to move in (x,y)

60%

Make a copy of the code `rrt_planner.py` to `rrt_planner_point_robot.py` and edit the TODO parts for the following Part1. Run your experiments on `world1.png`. Start_position = (10,270) and target_position = (900,300).

1a. Implement the RRT algorithm for an omnidirectional point robot on a 2D plane. You will sample the world uniformly to expand the RRT.

1b. Change the sampling to a Gaussian distribution around the destination. This means that while sampling the world you will have more number of samples around the destination target. Does sampling the world from different distributions for expanding the RRT have any effect on the planner? Please discuss.

[Hint: You can use python library : ‘random’ to achieve these samples]

1c. Run experiments from **1b** with varying extension_step_size (at least 10 different step_sizes). Run at least 10 trials for each extension_step_size and plot the following with error bars over 10 trials:

- Num_rrt_iterations vs. step_size
- Rrt_path_length vs. step_size

Things to submit:

- A report “rrt_planner_report.docx” with 2 images (Planning phase and Final path to destination as seen in above figures) for each of the sub-questions **1a** and **1b**. For **1b**, discuss the effect of uniform and Gaussian distributions on the planner. Add the plots from **1c** to the report and discuss the observations with varying extension_step_size. Discuss the plots and what do you think would be a good step_size according to your observations.
- Your code `rrt_planner_point_robot.py`

Part 2 : A Line robot of unit length that moves in (x,y,theta)

40%

Make a copy of the code `rrt_planner.py` to `rrt_planner_line_robot.py` and edit the TODO parts for the following Part2. Run your experiments on `world2.png`. Start_position = (100,630,0) and target_position = (800,150,0).

2a. Implement the RRT algorithm for a line robot on a 2D plane. You will sample uniformly to expand the RRT. Initial length of the line robot should be 50 units (Pixels).

2b. What is the effect of length of the line robot on the performance of the planner?

In this section, you will vary the length of the line robot (at least 10 different lengths) and run the code from **2a** to observe the effect of length of the line robot on the performance of the planner. Run at least 10 trials for each length of the line robot and plot **Num_rrt_iterations vs. length_of_the_robot** with error bars over 10 trials.

Things to submit:

- Add to the report “rrt_planner_report.docx”, 2 images (Planning phase and Final path to destination as seen in above figures) for sub-question **2a**. Add the plot from **2b** to the report and discuss the observations with varying **length_of_the_robot**.
- Your code `rrt_planner_line_robot.py`

We are also providing you with `example_world.png`. Try and run all of your implementations on other worlds provided to see how your planner works on different worlds. Also change the start and target locations to test your planner.