Project Review

Search and Mapping

Student: chaichangkun
Reviewer: Qianglu

Hi Chaichangkun, your first project for Robotics Nanodegree is great. You wrote a clear writeup and notebook analysis. You only need some minor changes to make the project meet requirements. In the mapping part, the thresholded image should be shown in the Roversim program. Suggestions are made in the following corresponding part.

Write Up

Criteria: Provide a Writeup / README that includes all the rubric points and how you addressed each one. You can submit your writeup as markdown or pdf. Here is a template writeup for this project you can use as a guide and a starting point.

To meet specifications: The writeup / README should include a statement and supporting figures / images that explain how each rubric item was addressed, and specifically where in the code each step was handled.

Requires Changes:

You writeup is clear and easy to read. You included figures that explain how each rubric item was addressed. However, it would be helpful if you mention where in the code each step was handled.

Notebook Analysis

Criteria: Run the functions provided in the notebook on test images (either data you recorded or the test data provided). Add/modify functions to allow for color selection of obstacles and rock samples.

To meet specifications: Describe in your writeup (and identify where in your code) how you modified or added functions to add obstacle and rock sample identification.

Meets Requirements:

Great, you used different color ranges to apply thresholding on the original images which returns the navigable area, obstacles and rock samples. Good job!

Criteria: Populate the **process_image()** function with the appropriate analysis steps to map pixels identifying navigable terrain, obstacles and rock samples into a worldmap.
Run **process_image()** on your test data using the **moviepy** functions provided to create video output of your result.

To meet specifications: Describe in your writeup how you modified the **process_image()** to demonstrate your analysis and how you created a worldmap. Include your video output with your submission.

Meets Requirements:

Enough images are recorded to show the effectiveness of process_image(). Steps of the function process_image() are explained well how pixels are thresholded and mapped to worldmap. Movie is created to show the results.

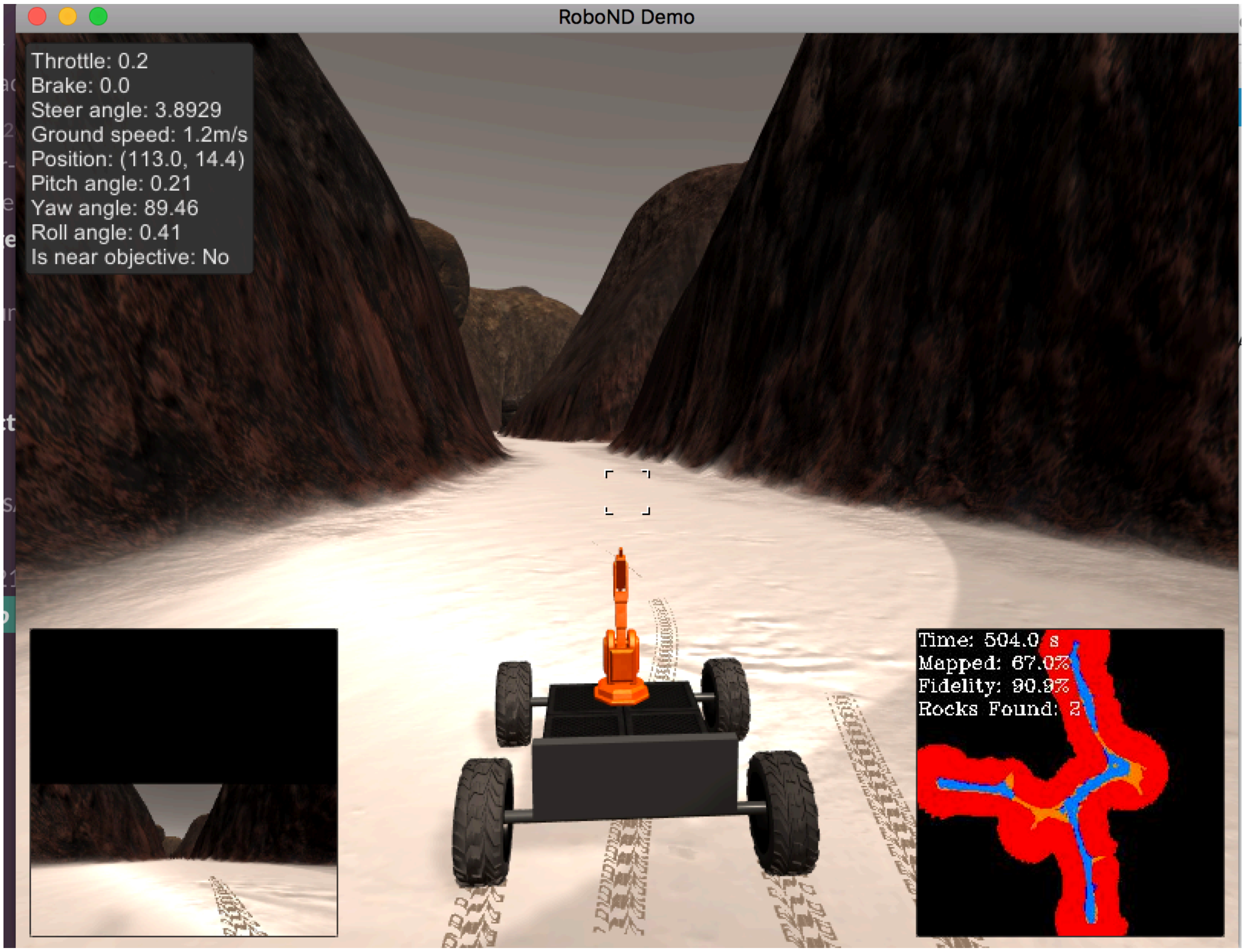
Autonomous Navigation and Mapping

Criteria: Fill in **the perception_step()** (at the bottom of the **perception.py** script) and **decision_step()**(in **decision.py**) functions in the autonomous mapping scripts such that your rover can navigate and map autonomously. Explain how you did this in your writeup and discuss your results.

To meet specifications: **perception_step()** and **decision_step()** functions have been filled in and by running **drive_rover.py** and launching the simulator in autonomous mode does a reasonably good job at mapping the environment.

Requires Changes:

The perception.py has been filled in with functions in the notebook. I ran 2 times of your code. The 1st time, the rover stuck at a big rock at about 200 seconds with 40.3% mapped. And the second time, the rover successfully mapped all navigable terrains. It is a bit weird to see the mapped percentage decreasing some times. And even the map is 100% mapped, the percentage is not the same. See figure below:



Throttle: 0.2
Brake: 0.0
Steer angle: 3.8929
Ground speed: 1.2m/s
Position: (113.0, 14.4)
Pitch angle: 0.21
Yaw angle: 89.46
Roll angle: 0.41
Is near objective: No

Time: 504.0 s
Mapped: 67.0%
Fidelity: 90.9%
Rocks Found: 2

The only thing you need to work more is figuring out why the thresholded image is not shown. The reason might be that you are using binary images which has a maximum value of 1 while the image shown in the simulator has a maximum value of 255.

Keep up your great work!