UDACITY

‹ Return to "Computer Vision Nanodegree" in the classroom

Landmark Detection & Tracking (SLAM)

REVIEW

CODE REVIEW

HISTORY

Meets Specifications

Congratulations on completing the project!

Your SLAM implementation performs well. In your `sense` function, in the file `robot_class.py` on line 96, you have

```
if (dx**2 + dy**2 <= self.measurement_range**2):
```

You can also use

```
if (abs(dx) <= self.measurement_range and abs(dy) <= self.measurement_range):
```

For if you would like to further explore SLAM, take a look at this as well as this, and this.

All the best in future projects!

`robot_class.py`: Implementation of `sense`

Implement the `sense` function to complete the robot class found in the `robot_class.py` file. This implementation should account for a given amount of `measurement_noise` and the `measurement_range` of the robot. This function should return a list of values that reflect the measured distance (dx, dy) between the robot's position and any landmarks it sees. One item in the returned list has the format: `[landmark_index, dx, dy]`.

🎉 Excellent! You have implemented `sense` function.

Notebook 3: Implementation of `initialize_constraints`

Initialize the array `omega` and vector `xi` such that any unknown values are `0` the size of these should vary with the given `world_size`, `num_landmarks`, and time step, `N`, parameters.

🎉 Awesome! You have initialized the array `omega` and vector `xi`.

Notebook 3: Implementation of `slam`

The values in the constraint matrices should be affected by sensor measurements *and* these updates should account for uncertainty in sensing.

🎉 Well done! The values in the constraint matrices are affected by sensor measurements.

The values in the constraint matrices should be affected by motion `(dx, dy)` *and* these updates should account for uncertainty in motion.

🎉 Great! The values in the constraint matrices are affected by motion `(dx, dy)`.

The values in `mu` will be the x, y positions of the robot over time and the estimated locations of landmarks in the world. `mu` is calculated with the constraint matrices `omega^(-1)*xi`.

🎉 Good job! `mu` is calculated with the constraint matrices `omega^(-1)*xi`.

Compare the `slam`-estimated and *true* final pose of the robot; answer why these values might be different.

🎉 Well done! Your estimated values are close to the true values.

There are two provided test data cases, test your implementation of slam on them and see if the result

There are two provided test_data cases, test your implementation of slam on them and see if the result matches.

:tada! Excellent! The values in the test case are close to the values estimated by your implementation.

DOWNLOAD PROJECT

RETURN TO PATH

Rate this review