

Goldeneye Kinect Mini-Project

Raghav Anand

October 1, 2017

1 Documentation

1. **Dependencies:** Open CV 2. Open CV instructions are ubiquitous online, install the version relevant to your system.
2. **collect.py:** Collects data from the kinect and stores it frame-by-frame in numpy arrays. You will not need to interact with this. It is useful to look at it to see how to store numpy arrays. Storing some arrays might help speed up your iteration process by preventing the need to recompute intensive tasks everytime you tweak a program.

3. **display_images.py:** Run using the command line

```
python display_images.py
```

By default this runs 'rover_1' and displays all the images under that folder in RGB and depth format one at a time. This is just useful so you have some visual intuition, the actual data manipulation will have to be done directly from the numpy arrays. Look through this file to see how the numpy arrays are being retrieved from storage. If you want to run images from a different folder or space out each frame so you can see more clearly what is happening, run

```
python display_images.py 'FOLDERNAME'  
DELAY_TIME_BETWEEN_IMAGES
```

Note that to specify a delay you must include a folder name.

4. **Data format:** The data is stored under folders titled 'rover_x'. Every folder has depth and rgb subfolders. Depth information consists of a single (480 x 640) numpy array. RGB information is encoded in (480 x 640 x 3) numpy arrays i.e. (480 x 640) for each of red, green and blue.

2 Objectives

1. Identify the rover object every frame (you can assume that the rover exists in every frame). Feel free to use external libraries or build it from scratch.
2. Calculate the position, velocity and acceleration of the rover at every instant. Pretend like you're a dashcam on a self-driving car, these are three key parameters you need to be able to sense.
3. Calculate predicted trajectories at every instant for the rover. Try looking into Kalman filters or other techniques you find online to do this.
4. Do all the above tasks with and without depth information (you always have access to RGB information). This will help us gauge how much depth information from a purpose built sensor helps as opposed to just RGB computer vision.

3 Testing

We will meet as a group once you have your predictive models ready. Please test them to make sure they have a reasonable degree of accuracy. When we meet I will bring in the Kinect and we can test it in real time to see how your algorithms perform!

4 Miscellaneous

Here are the resources I used to get started with the Kinect. Note that each of you is not individually working with the Kinect because that is an unnecessary roadblock. Instead you get to dive in and make sensory predictions, while still being able to test it with the real hardware at a later point. If you want to use the Kinect yourself, follow these two very helpful tutorials (Windows users please just look up the same libraries, there will be tutorials for your specific OS)

<http://blog.nelga.com/setup-microsoft-kinect-on-mac-os-x-10-9-mavericks/>

<https://naman5.wordpress.com/2014/06/24/experimenting-with-kinect-using-opencv-python-and-open-kinect-libfreenect/>