Alex Rich and John Allard
Harvey Mudd College
Summer 2014 REU

Using the 3D Localization Program

1 Locations

The 3D Localization programs are located at `github.com/jhallard/3DLocalization`.
Some example Robot Control Programs are located at `github.com/aarich/ROS-Controllers`.

2 Quick Start

See the next section on how to visualize what's happening. To run the program using the already present database of images, open three terminal windows (or tabs):

1. `roscore`

2. `cd` into `3DLocalization/Localization/build/` If needed, compile the code using the command `make`. If this is done already, run

   ```
   ./devel/lib/localization/3DLocalization 2ndFloorSprague
   ```

3. `cd` into `3DLocalization/Localization/RobotTest/build/`. Again, `make` as necessary. Run `./robot 2ndFloorSprague`.

4. Once you allow these programs to run, they should stop. The first will tell you to press enter, and the second will say "Done Loading Images." You should press enter on the first, then press enter on the second. This initializes the handshake.

The program runs, displaying the virtual robot image and the top match if found. Debug statements are printed, and can be turned off in ProgramIO.cpp.

3 Visualization

All visualizers are in the `3DLocalization/Localization/src/GUI/` folder.

3.1 MapViewer

BLAHBLAH

3.2 MATLAB

This is run after the localization is done. Run `weightovertime` to get a graph of the top weight over the iterations as well as a graph of the average of the top 20 particles.

3.3 Meta

In here you can run `python createTrace.py` or `python WeightOverTime.py`. The first animates the path of the localization guess, and can be run at any time during the localization or after. The second graphs the same as the matlab, but less pretty.

3.4 PCLViewer

Run `./pcl_viewer 2ndFloorSprague` to view a PCL visualization of the particles. Each particle is colored according to its weight. Only one particle is shown even if multiple particles exist at a certain spot.

3.5 PyViewer

This is located in `GUI/PyViewer/` Simply run `python Viewer.py`. To adjust the number of particles displayed, click on the top section of the visualizer. Clicking towards the right increases the number of particles shown, while clicking towards the left decreases it. To close the window, click on the lower portion of the window.

4 PreLocalization

4.1 Start with an Object File

We will refer to the object file with "ModelName." You should decide on a model name for your map, then be consistent. Changing the ModelName will mean that the program won't be able to find files. Open object in MeshLab, select "Remove Unreferenced Vertices," then export as obj file.

4.2 Rendering Images

4.3 Creating the Database of Features

Run CreateDB: `./CreateDB ModelName`. CreateDB does the following:

1. Reads images from directory in Data/RenderedImages.
2. Computes various features, including.
3. SURF Descriptors.
4. Average Pixel Sum images of size 50 × 50.
5. Above/Below images of same size.
6. Save images to corresponding folder.
7. Both descriptors are saved in yaml files.
8. Images are stored to gsimages and bwimages.

5 Localization

Once the database has been created, simply run the MCL program with the name of the file:

```
./devel/lib/localization/3DLocalization 2ndFloorSprague
```

The

6 Using an established ROS-Controllers program

Connect to respective robots, then run the ROS-Controller program after running the main MCL Program, as instructed above.