Eric Schneider and Emily Wang
Due: November 6th, 2014
CompRobo!
Computer Vision Project Writeup
#SfM

What was the goal of your project?
The goal of our project was to understand and implement Structure from Motion. Structure from Motion is the recreation of a 3D scene from 2D images. This technique is potentially applicable to situations in which there is a moving camera and stationary objects, and has potential crossover with the eye-helper and ARL scope projects.

How did you solve the problem? (Note: this doesn't have to be super-detailed, you should try to explain what you did at a high-level so that others in the class could reasonably understand what you did).

- Building off of Paul's code (started with the epipolar/fundamental matrix code)
- Building off of the Baggio book code (c++ implementation)
- Gathering a variety of image frames:
    - Using items with lots of keypoints
    - Trying different transformations: rotation and translation alone
    - Took images of boxes, wall corners
- Making our own visualizations (homography things; rviz)

Describe a design decision you had to make when working on your project and what you ultimately did (and why)? These design decisions could be particular choices for how you implemented some part of an algorithm or perhaps a decision regarding which of two external packages to use in your project.

- A lot of our project involved trying to step through the OpenCV processes and trying to check that each step was correct.
- We did a quick and rough filter at the end of the project to
    - 1) filter out all points with negative Z (they are just wrong)
    - 2) filter out all points that had Z greater than 10 times the mean Z. This was a rough filter to get rid of the massive outlier points. This filter would do a bad job of capturing a large depth range, it's best for points that are all a similar distance away.

How did you structure your code?

- Converted Paul's code into an object-oriented architecture
- Made functions for different steps
- Plotted the calculated pcloud points into RVIZ after compute_depth calculations

What if any challenges did you face along the way?

- Essentially our project involved debugging each step of the SfM process, so each step proved a new challenge. When something just didn't work well, like the

depth triangulation, we often wouldn't know what to do, because we'd done what we thought was best.

What would you do to improve your project if you had more time?

- Incorporate multiple views into generating the point cloud; bundle adjustment
- Test this with more rotation/translation/scale variations
- The current program isn't robust enough to throw onto a neato/robot yet… right now the point cloud still has very nontrivial outliers and hasn't been tested on items with odd shapes. Perhaps coupling SfM with more sensors (odom, etc) would improve the functionality, though we couldn't describe exactly how to do that.

Did you learn any interesting lessons for future robotic programming projects? These could relate to working on robotics projects in teams, working on more open-ended (and longer term) problems, or any other relevant topic.

- Considerations for future project ideation: Algorithm-y projects vs. "Challenge Accepted!" projects; ideally there's a balance between the two project flavors but it needs to be scoped well. This project was more on the algorithm-y side and was definitely interesting to study and gain exposure to, but not as creative or hacker-minded.
- Tools for checking stuff ; validating matrices and things
- Plotting corr. lines in diff colors as a sanity check
- Also… order matters for recreating the scene

See our presentation for example images.