Journal Report 8

10/28/19-11/7/19
Aneesh Boreda
Computer Systems Research Lab
Period 4, White

Daily Log

Monday October 28

I looked to find a paper with a more detailed description of the overall algorithm, as implementing the algorithm is much more difficult in C++, so I'm trying to make it as easy for myself as I can. I didn't find exactly what I was looking for, but I did find a paper specifying more on the calibration step.

Tuesday October 29

I wasn't at school today.

Thursday October 31

After searching for a while, I finally found a paper with a detailed description of the stereovision algorithm, so now I have a lot better start for programming the calibration part. Also, I accidentally overwrote the calibration training images with test images, so I had to retake the training images.

Wednesday November 6

I started programming the calibration algorithm in C++. It involves taking training images and comparing the difference in location of points throughout the image for a picture of a chessboard. Since the chessboard has regular spacing between each of the squares, transformation matrices can be calculated as the location of the chessboard in space is much easier to find.

Thursday November 7

I found some OpenCV sample code for calibration on github, so I downloaded that and tried to run it. However, some of the dependencies weren't working, so I spent the class installing them and debugging why C++ wouldn't recognize the new libraries I had to download.

Timeline

Date	Goal	Met
Week of 10/17	Figure out if segfault can be fixed, otherwise find another method for creating a point cloud	It was a short week, so I started on this
Week of 10/24	Continued from previous week	Yes, I determined that I need to find another method
Week of 10/31	Start setting up configuration of stereo vision prototype in C++, make sure all necessary libraries are installed	Combined with next week
Week of 11/7	Continued from previous week	I made good progress on the calibration step, as I found a sample on github that I can adapt.
Week of 11/14	Figure out necessary information to store for camera calibration, work on generating calibration from chessboard training images	

Reflection

These two weeks, I worked almost all of the time on generating the calibration, which is necessary to rectify images later on based on the known positions of the cameras. This ensures that certain objects in the image are positioned in the same orientation and along the same epipolar lines, which ensures that both images are positioned on the same plane. This means that comparing the points in the images can almost directly give the distance to that point based on the difference in position. From this, a 3D point cloud can be generated when calculated for as many points as possible across both images. Since both chessboard images have regular spacing between the corners, those are used to calculate necessary information about the cameras. The code I found on github calculates but doesn't save this information, so I need to figure out how to properly store this for later retrieval.

Winter Goal

My goal is to have working point cloud generation, and to be able to store these distances at different pixel locations.