Joshua, Hannah, Lucas

Statement of Work

Our refocused capstone will largely look similar to our proposed project outlined in our design document. We will be refining and implementing our design for an error detection system for the Printrbot 3D printer that Lucas currently has. While we hope to be using multiple cameras for stereo imagery in the project demo, we will focus first on the blob detection discussed in our paper. The main challenge to the current situation is the additional testing involved with integration. Since Lucas is the only group member with a Printrbot, and each of us currently has at most one camera from the trade studies, focusing first on the blob detection (which only requires one camera) and quantifying the detection and accuracy rates for that algorithm will help guide us on next steps. As we receive parts and complete the various tasks on our timeline, we can determine how best to approach the stereo imagery aspect of the project, if we even have time for that. To further help us reach our goals, we decided to only attempt to detect two instead of four errors. We chose to detect faulty bed adhesion and mid-print extruder errors because these are common and tend to produce the most drastic and easily captured errors.

In terms of what each person is doing, Hannah has already started on implementing the blob detection algorithm. She will also help implement the set of gcode parser functions. For the most part, she will follow the Gantt chart in the paper. Joshua will write the main function which performs the necessary function calls and appropriate error checks. One possible challenge was that the Octoprint documentation warned against putting computationally expensive operations (including I/O) in the Octoprint plugin handlers. The current plan is to write the program and quantify how messed up (or not) the plugin will behave with the edge detection function calls, and then move to a multithreaded or multiprocess solution. We originally chose to develop with the Python language because Octoprint plugins are usually written in Python;

however, Python, being interpreted, might produce unintentional latencies that affect our final system. If necessary, we can use the Cython superset language to provide typing and give access to C functions, allowing for more performance. Lucas will continue to work with the Printrbot and Octoprint. As mentioned previously, integration will be our main obstacle because fixing bugs relies on an ability to detect when something goes wrong with our system. Really, Lucas will have to be the main communicator on that end since he is the one with the Printrbot, and so communication will be extremely important so that the three of us can efficiently detect and debug errors in our implementation.