Team B3 - 2D23D

Authors: Alex Patel, Chakara Owarang, Jeremy Leung: Electrical and Computer Engineering, Carnegie Mellon University

Introduction— **Because of the unfortunate current situation of the Coronavirus Outbreak, our team members are scattered in different places with different timezones: Alex will be in Connecticut, Chakara will be in Bangkok, and Jeremy will likely return to Toronto once circumstances allow. This means that our project must be able to be completed remotely, which would void any hardware or physical requirement for our project. Below is the details of our refocused project in order to achieve most of our project requirements.**

1 Updated Requirements and Validation

Since we are removing our physical components and requirements for our project, we also have updated our project requirements and validation. We are still keeping the **Accuracy** and **Affordability** requirements of our project. Below are changes in our other requirements.

Usability and Portability

For the portability requirement, since we do not have physical components anymore and would be using simulated sensor data, we can completely remove this requirement as well as the validation method of testing the platform with a load. We are also removing the usability requirement that the device is easy to setup and weighs less than 7kg as we would not have the actual device anymore. Since we would not be testing with real objects, we would also remove usability testing and user testing.

We are still keeping our main usability requirement that the program outputs a common 3D format that we will be able to input to a 3D printer (STL format). This requirement can still be easily evaluated and do not require any special quantitative tests.

We will maintain the same object size requirement of the input object being 5cm to 30cm along all axes - this is because the requirement is also tied with how we setup the camera's field of vision and we want to imitate our original setup as much as possible.

Efficiency

Since we are now simulating sensor data and we would not be using NVIDIA Jetson GPU as our processor anymore and due to uncertainties, we are relaxing our efficiency requirement to **10 minutes** which only include processing time. We would be using Jeremy's computer (GTX 1070 with 8GB of RAM) to test our benchmarks.

Accuracy

Although we are not relaxing our accuracy requirement, we are changing our validation method for this requirement. We would not longer be 3D printing objects and test them since we no longer have the physical components of our objects. Instead, the database of models we found will act as absolute ground truth, and our simulated scans of the objects will be compared directly to those models for verification. The method to calculate the difference between two triangular meshes will be the same as originally proposed.

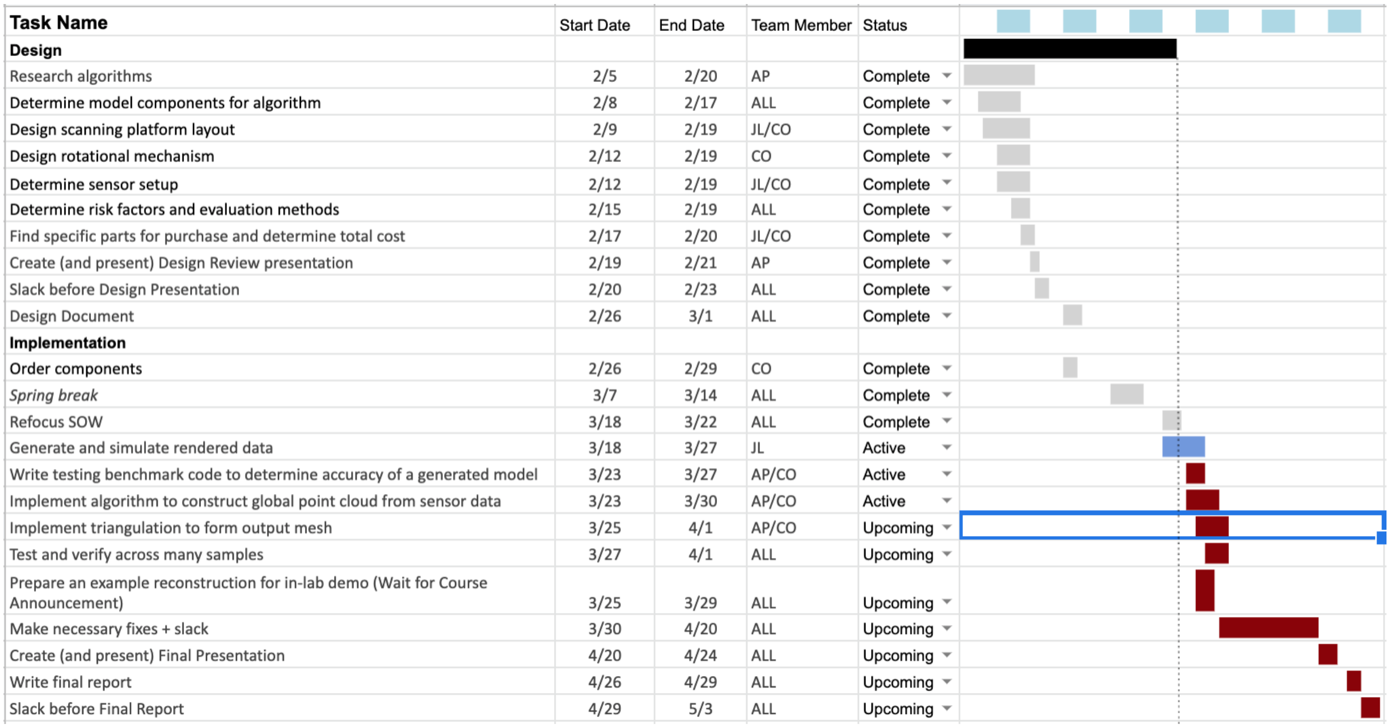
2 Changes in Design and Implementation

Since we are unable to create a physical device to carry out 3D scanning, we have changed our project to instead utilize a virtual simulation of a 3D scan. This simulation will be done by taking renders of a 3D scene, which act as the camera input from our original design. The remainder of our software pipeline will remain unchanged. This decision to simulate data capture allows us to stay true to our original design requirements, while adapting to a changing environment.

3 Project Management

3.1 Schedule

Since we removed all the physical components of the project and added a few more important tasks, we adjusted our project schedule. Below is our updated Gantt Chart.



Task Name	Start Date	End Date	Team Member	Status
Design				
Research algorithms	2/5	2/20	AP	Complete
Determine model components for algorithm	2/8	2/17	ALL	Complete
Design scanning platform layout	2/9	2/19	JL/CO	Complete
Design rotational mechanism	2/12	2/19	CO	Complete
Determine sensor setup	2/12	2/19	JL/CO	Complete
Determine risk factors and evaluation methods	2/15	2/19	ALL	Complete
Find specific parts for purchase and determine total cost	2/17	2/20	JL/CO	Complete
Create (and present) Design Review presentation	2/19	2/21	AP	Complete
Slack before Design Presentation	2/20	2/23	ALL	Complete
Design Document	2/26	3/1	ALL	Complete
Implementation				
Order components	2/26	2/29	CO	Complete
Spring break	3/7	3/14	ALL	Complete
Refocus SOW	3/18	3/22	ALL	Complete
Generate and simulate rendered data	3/18	3/27	JL	Active
Write testing benchmark code to determine accuracy of a generated model	3/23	3/27	AP/CO	Active
Implement algorithm to construct global point cloud from sensor data	3/23	3/30	AP/CO	Active
Implement triangulation to form output mesh	3/25	4/1	AP/CO	Upcoming
Test and verify across many samples	3/27	4/1	ALL	Upcoming
Prepare an example reconstruction for in-lab demo (Wait for Course Announcement)	3/25	3/29	ALL	Upcoming
Make necessary fixes + slack	3/30	4/20	ALL	Upcoming
Create (and present) Final Presentation	4/20	4/24	ALL	Upcoming
Write final report	4/26	4/29	ALL	Upcoming
Slack before Final Report	4/29	5/3	ALL	Upcoming

Figure 1: Gantt Chart

3.2 Team Member Responsibilities

To be able to still achieve our goals despite the current situation of different team members being in different locations and our updated plan for the project, we have also updated the team member responsibilities. Our team divides work among each team member equally. The team deals with logistics, integration, and decision-making together but each member still has his main tasks assigned as follows:

Jeremy:

- 3D simulation construction (Blender) and assembly of testing models
- Outlier removal and noise reduction
- ICP for combining multiple scans

Chakara:

- Point cloud generation from sensor data
- Optimization of software components for GPU
- Point cloud triangulation

Alex:

- Laser stripe detection and mapping to world coordinates
- Setting camera calibration parameters in simulation
- Testing benchmark code

3.3 Risk Management

All of the risks we listed in the design document are related to the physical component of the project: part of the input object is obscured, vibrational noise, the stepper motor angle data is inaccurate, the laser stripe doesn't have enough intensity, potential holes in the point cloud, and NVIDIA Jetson Nano and motor driver integration. Thus, we can ignore these risks. However, another main risk we have right now is the **accuracy of simulated data**. We need to make sure that the the laser data we get and the rotation angle of the object are accurate and that we can retrieve these data from the rendering software we use. We can mitigate this risk by, if we can't get the data we need for our algorithms, then we can order the Intel RealSense Depth camera and use a depth map instead. This means we would need to add back some physical components but mainly just the platform and lazy susan. When using the depth camera, we can manually rotate the input objects.