

3D Scanner MRD Project Report

Vangel F. Author, CSE, Siyan L. Author, EE, Chenkai Z. Author, EE

Abstract—As we look throughout history, the wealthy and privileged have had access to useful technologies, before the masses. When cars, televisions, and even computers first entered the market, they were revolutionary but overpriced. This discouraged people from acquiring them. These technologies are not necessities to life, but they play big roles in making life more convenient. 3D scanners are relatively new technologies that have many useful applications, but yet very few people own them, mostly due to the expense of the product. We will talk about a way to create a cheap standalone 3D scanning system, which through the use of processing information taken by a camera and line lasers, can create digitized 3D models. These models can be used in digital animation or printed with 3D printers for a wide verity of applications.

I. INTRODUCTION

The world is advancing technologically faster and faster every year. However, the more revolutionary technologies tend to be too expensive for the average consumer. One of these technologies is a 3D scanner. These scanners have a starting price of \$400. This is a high price for an item that can be useful, but not necessary to have, “Generally, most 3D scanners are too large and expensive to be used by consumers. That’s at odds with the culture of 3D printing right now, considering most of the developments are in desktop 3D printers made for home use. Most of the 3D scanners out there are more than \$1,000...” [1]. Besides the cost of just the scanner, there are other issues.

Scanners that exist have another flaw as well, which is that they are not standalone systems. This means that they do not run unless there is an external device that is necessary to control the scanner. In most cases, this external device is a desktop or a laptop. With the advancement of technology there are appliances with applications that can control the scanner such as smart phones and tablets. Scanners are expensive enough, and consumers shouldn’t have to purchase another device, which is also hundreds or even thousands of dollars, to use their scanner.

Before 3D scanning had a huge role in 3D printing, people used Computer Aided Design (CAD) tools in order to create the object for 3D printing (they would design the object on a computer digitally). Although the use of CAD tools is still a very viable option, it has its own set of issues. The use of these tools is a skill; A skill that most people do not have. Not to mention that for a lot of practical uses, sizing of the objects that

are to be printed need to be exact (nuts and bolts), which takes time and effort. The time it takes to learn how to use the tool, and the time it takes to implement your knowledge every time you want to print something, is a thing of the past. With 3D scanners you can save time and just place the object in the scanner and let the computer do the work for you [2].

3D scanners can be used to scan parts, building supplies, props for movies, digital animation, and much more. The usefulness of 3D scanners is wide and vast; however, it seems to be more of a nuisance, due to the cost of owning one.

In order to provide a solution, we designed a 3D scanner that can be made for under \$250 dollars. In addition to this, this scanner will be able to perform scans by being plugged into a wall, without the need of another device to run. Because of necessary precision, the scans must be accurate to within 1mm of the actual object size. The scanner will have a length of 12 inches a width of 10 inches and a height of 8 inches, making it a convenient size relative to most scanners. We also are designing the scanner to perform the scans in under 15minutes. Scanning something should not be a time consuming processes that makes scanning an object a hassle for the end consumer. Especially since people using scanners in the business world are usually busy people and time is a commodity of the essence.

TABLE I
SPECIFICATIONS

Specification	Value
Weight	<1kg
Height	>20cm
Length	>40cm
Width	~20cm

Design

A. Overview

There are very inexpensive ways to create a 3D scanner. Essentially, you need to capture the image of an object, then once you have that data, you need to capture another image of that object at a different angle. Repeat this process until you have images for the entire 360-degree view of the object. The data is then processed together through an algorithm in order to create a 3D model of that object in a STereo Lithography file (STL files are the files containing information about the 3D model that a 3D printer can take in as input to create a 3D object). In order to provide the 360-degree view, we are going to use a turn table to rotate the object that we are trying to scan. This turn table is driven by a stepper motor, which will receive .35A of current to take a step. It is controlled to do so

by A9488 chip, which takes instructions for our computing component.

In order to run the algorithm and do the computing, we will be using a Raspberry Pi 2 (RP2). We decided to go with the RP2 because of its convenient size, it is compatible with a small camera that fits our purposes for the data collection. We will be using both a camera (the RP2 camera) and lasers retrieve information about the dimensions of the object that we are scanning.

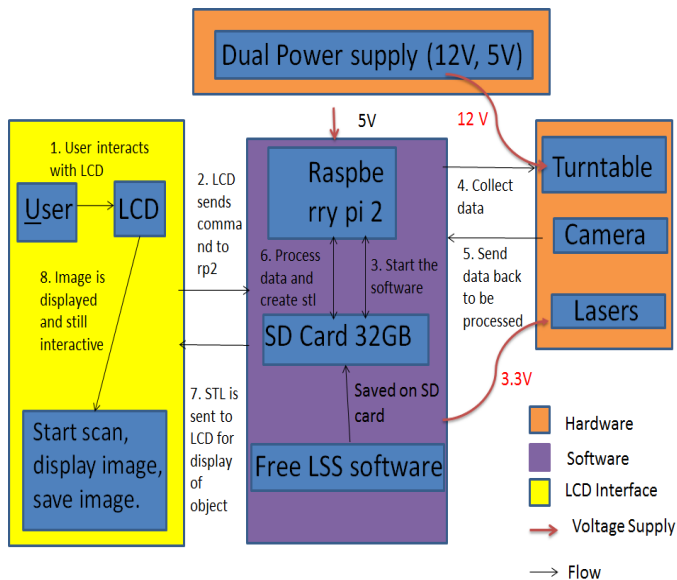


Fig. 1. This is a block diagram of our overall design. The arrows are numbered in the flow of how to run a scan, starting from when a user interacts with the GUI to start.

Also, in order to create the standalone system, we need to be able to have a user interface to communicate with the system. The RP2 is compatible with LCD screens. This component is relatively inexpensive, considering the alternatives mentioned earlier cost hundreds.

The overall design consists of the camera being fixed at a certain location. Everything else is in relation to the camera. The lasers are each located at an arbitrary distance to the left and to the right of the camera, however, both lasers must be the same distance away from the camera. The turn table is placed so that the center of the table aligns perfectly with the camera, and it can be any distance away. Each laser must shine their light directly through the center point of the turn table (refer to Figure 2).

The distances between the camera and the lasers, as well as the camera and the center of the table must be known in order to properly take a scan. Knowing how far the center of the table is from the camera provides a reference z- coordinate in 3D space. Knowing the height of the camera provides a reference y- coordinate, and having the table centered provides an origin for the x-coordinate. Then by having the lasers some fixed known distance from the camera, and having them cross the center of the table, we can calculate the hypotenuse when the light hits an object.

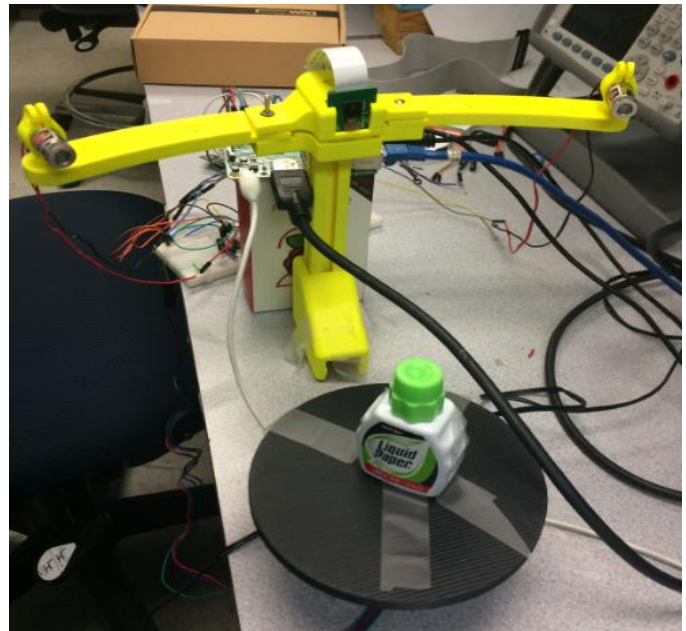


Fig. 2. This is a representation of how the lasers should be in relation to the camera. Note that the turn table is not in the shot, it would be placed in the direct path of the camera lens and the laser lights would be adjusted to shine through the center

There were a few alternatives that we considered when developing our design. We considered using only a camera for the data collection. This method of 3D scanning is called photogrammetry. This requires taking multiple pictures, but you also need to take the images at different angles. This would make the design much more difficult to implement, due to the necessity of having more than one camera, each at a different angle. This is problematic because it drives up cost if we need more than one camera. The RP2 also only has one slot for a camera input. With the laser component, we can get the information we need, which we will go over later.

A. Hardware design

Lasers

Two red LED line laser were selected to be the transmitter which its light will be projected onto the target sample and is monitored by the camera. The distance between the camera and laser diode is fixed and known. A picture of the object is taken without any lasers being turned on. One laser is switched on and another picture is taken. Since the laser is by far the brightest element in the image, the line it makes on an object is easily distinguishable from the rest of the image. [4] Our software will be computing the difference between the two images, a laser trace is obtained. We chose red LED since it consumes the least amount of power in comparison with other LEDs of different colors.

We realized some of the irregular shapes are unable to be scanned, for example a wine glass due to its concave part. We found out that when you scan an irregularly shaped object, there are areas

that one laser can't see, but the other laser does. Using two lasers does increase the quality because it reduces the number of "gaps" in the model due to concave areas on the part. For further development, we may try to use a different colored laser and try scan in different ambient light.

After wiring the LED's to the bread board, I connected them to a GPIO pin on the RP2. I wrote a python code that would turn that particular pin to HIGH, which supplied enough voltage to turn the laser on (3.3V). The python code would enter a for loop that toggled the laser on and off. As a parameter it took in how many times to blink and how long to stay on before turning off. This was important because we need the lasers to turn on and off again. I used what I learned in electronics lab and computer systems lab to accomplish these tasks, as well as learning a little bit of python on my own.

Stepper Motor

Our scanner able to do a 360-degree scans through the implantation of a rotation table platform. Due to the high precision of control, we decided to use a stepper motor for our rotation motor. We choose NEMA-17 stepper motor; it is 4-wire bipolar stepper motor has 200 steps per revolution which is 1.8 degree per steps [5]. It has a maximum current of 350mA and therefore it required a stepper motor driver which driven with 2 full H-bridge.

To test this component, I also wrote a short python code to drive the stepper motor. It took the number of steps as a parameter, as well as the direction (clockwise or counterclockwise) to turn. After my success, I wrote yet another short python code to control the lasers and motor at the same time, to ensure that we had enough current to drive both at the same time.

Camera

We ended up using the raspberry pi camera module. It is 5MP camera module is capable of 1080p video and still images and it can be connecting directly to the raspberry Pi. Here is some data we get from the software of how long it takes to scan when we run a scanner on different resolution of the camera. The size of file is about 2MB-100MB depends on the resolution the user chooses.

Resolution	time
5 Megapixel(2592 X 1944)	55mins
1.9Megapixel(1600 X 1200)	10 mins
1.2Megapixel (1280 X 960)	8 mins
1.2Megapixel (1280 X 960)	4 mins

The RP2 has a command to test whether the camera is correctly functioning.

Power

The raspberry pi requires high performance to run the Freelss software which required a lot power. As a result, we are not building a portable 3D scanner. Instead using battery, we decided to use a dual power supply which can power the 5V raspberry pi and 12V stopper motor at the same time. The 12V/5V AC to DC power adapter takes 100-240VAC as input

voltage and output 12V and 5V. The output current will be 2A. The maximum power is 34W and it is calculated below. $12V * 2A = 24W$, $5V * 2A = 10W$, $24W + 10W = 34W$. The raspberry pi takes 5V and 1.8A as input source, 50mA to draw the GPIO pins and roughly 250mA to drive the keyboard and mouse. As a result, the power consumption roughly is $1.8A * 5V$ (raspberry pi) + $1A * 12V$ (table) + $250mA * 5V$ (Camera) = 22.25W. Our dual power supply should good enough to handle the power consumption.

B. Software

The software block (the purple block in Figure 1) is the block that will control the hardware components, use these hardware components to control the hardware to gather data, and process the data being gathered to produce the output STL file. These processes are being controlled by the RP2.

The RP2 uses the Rasbian OS which is downloaded onto the SD card. With this OS, the RP2 is able to run open source software on GitHub known as freelss. Working with the terminal required some knowledge from software intensive engineering. The RP2 is driven on 5V, which will be supplied from a dual power supply that outputs 12V as well as 5V (the 12V are required to drive the stepper motor).

As far as building this block is concerned, it is necessary to get the Raspbian OS onto the RP2, and attaching the camera. The software running on the RP2 needs to command the physical hardware, which requires a library call WiringPi. Finally, wiring must be done connecting the hardware to the GPIO pins on the pi, in order to send commands to the hardware.

When the scan begins, the RP2 sets the pin for the right laser to HIGH, this enables the laser to shine a vertical red line over the object that we are scanning. Similar to setting signals to HIGH in computer systems lab. By capturing the image with the camera, the software calculates an (x, y, z) coordinate for each point of the vertical line that hits the object. After, that pin is set to LOW and the laser turns off. The other laser gets set to HIGH and the process is repeated again. Note that you can choose to do a scan with either of the lasers instead of both, however the quality of your scan would be lower due to less data points and more gaps.

Once both lasers have been had their data obtained, the RP2 sends a signal to the pins connected to the A9488 chip which controls the stepper motor. The stepper motor turns in steps of 1.8 degrees. Once it steps, the data collection process begins again. The process is repeated for 3200 steps, which is equivalent to 5760 degrees, or 16 revolutions.

By taking all the data points of the line lasers from each step of the way, the software combines them all in order to create the 3D model of the object.

In order to build this block, we had to familiarize ourselves with wiringPi. We ran into multiple roadblocks while integrating our blocks together due to our lack of understanding of this concept. WiringPi had its own pin layout which the code corresponded to, not the pin layout of the RP2. Once I figured this out I was able to drive the current to the correct locations in order to operate the hardware. I also had to learn how to work

with open source code, this was the first time I work with GitHub code.

C. The user Interface

This is a block of a user interface. This consists of a capacitive LCD panel and a user interface, which will allow user to interact with the device. At the beginning, according to our understanding of Freelss software, such software is a completely terminal based software. But as soon as we compiled the software and ran it, we found out Freelss has its own interface. However, opening such interface involved typing commands into terminal. You also needed to enter the I.P address into web browser.

Based on situation discussed above, we've decided our user interface should save the trouble of going through all the tedious steps and directly bring user to start a scan. Hence, we will be using a touchscreen LCD panel as our monitor and input device. The reason we choose touchscreen over regular one is because we want to build a highly integrated system that takes minimum space. And we believe with our interface it is sufficient to only use a finger other than keyboard and a mouse. We also used Java to write our basic user interface. Even though we've had certain level of Java experience, using Java to create a user interface is not one of them, especially when we haven't use Java in a while. Most importantly, in order to achieve our goal, we need to learn how to use Java to change the parameter of a C++ code, which is even more challenging. Also, we need to use Java to get right into the web browser, without having to type in the terminal commands. Furthermore, this user interface we are developing can also recognized touch operation since on our design there are no keyboard and mouse.

II. PROJECT MANAGEMENT

Each team member's individual roles were first assigned after the team broken down the project requirement.

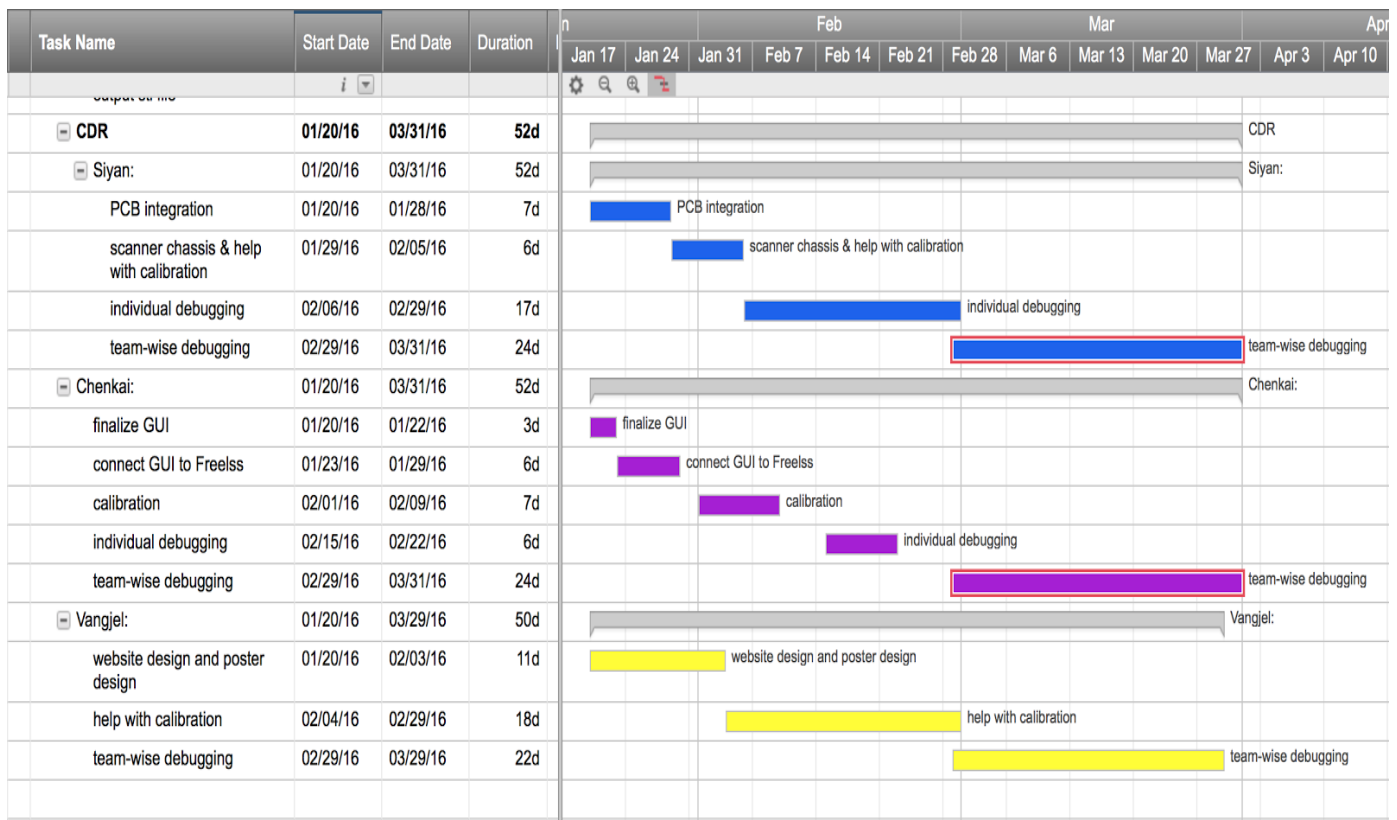
Siyan Lin has experience with microcontrollers and thus took the responsibility for the putting together the hardware parts, supplying power to these parts, and designing the physical structure of the 3D scanner. This included building a fully functions stepper motor to precisely control the rotation of the turn table. Siyan also made sure the hardware was able to retrieve signals when the Pi sent them out. To date, Siyan has successfully demonstrated functioning hardware.

Vangjel expressed interest in software, so his responsibilities include installed the Freelss software on the raspberry pi and making sure that it compiled. Vangjel made all the necessary changes corresponding to our 3D scanner project, including the use of WiringPi to which helps the software control the hardware. To date, Vangjel has gotten the software to do a full scan and output an STL file.

Chenkai expressed interest in building a user-friendly graphical interface with start and how to button to guide the user to use the software. To date, Chenkai was able to demonstrated the user interface on LCD to displace connectivity between GUI and raspberry pi 2

Team communication

All the team members produced a schedule of their available times. We met at least once every week. Occasionally, team members will meet with one another to clarify any question or receive help and support. There was constant communication though Facebook group chat. Weekly team meetings were scheduled for Thursday and Sunday and meetings with our Advisor, Professor Tessier were on Wednesday morning. During each team meeting with our advisor, we each start off



with an overview of each member's accomplishments. We had different interactions depend on what stage of the project we were in, and also what to try to improve on before the next meeting. If we had any problems or concerns, we would discuss how to move forward. We collaborate to share information as well as work on PowerPoint presentations using Google drive.

III. CHALLENGES

Although we accomplished a huge milestone by being able to output the STL file, it wasn't without some serious challenges. When we read through the code, we located the section that assigned hardware components to certain pins (see Figure 3). This code was written to be used by the original RP, so we modified the pin numbers in the code, by looking up the pin layout for the RP2. We connected the lasers and turntable to the numbers that corresponded. However when we ran the software, neither the turn table nor the lasers were working. At first we considered that there might be something wrong with the physical hardware of our system. We reran the python code for the lasers and turntable, and they both worked fine. Our next thought was that we weren't getting the 3.3V that we needed to operate the lasers.

By taking the voltmeter to the pins we thought were correct, we realized that the voltage at this pin was constantly low. After many attempts at debugging on the software level, we took a new approach. We ran the code, and used the voltmeter on all the pins to see which ones would toggle from low to high. Once we found the pins, we hooked up the lasers and tried running the code again. This time the lasers were working to the level we expected! However, this strategy would only work for the lasers because we knew that they would receive 3.3V or 1V (high or low). The stepper motor, however, had three different components that needed to be wired correctly (the enable pin, the direction pin, and the step pin). Because they received the voltage they needed from the 12V power source, finding the right combination of the three pins was not as simple. Considering there are forty pins on the RP2, finding the correct three would be near impossible and inefficient.

However, being low on options and ideas, Siyan and Chenkai, started using the voltmeter anyway to try and locate the correct pins. Meanwhile, Vangjel started to look into WiringPi. Upon his research, we figured out that WiringPi had its own set of pins, which mapped to certain pins on the RP2. The pins for each of the lasers mapped to the same pins we discovered by using the voltmeter. We reconnected the turntable by following this mapping and finally got all the hardware components to work, after many days of troubleshooting.

```
rightLaserPin(5),
leftLaserPin(4),
motorEnablePin(0),
motorStepPin(7),
motorDirPin(1),
motorDirPinValue(0),
```

Fig 3. This is a snippet of the code which corresponded to the pins.

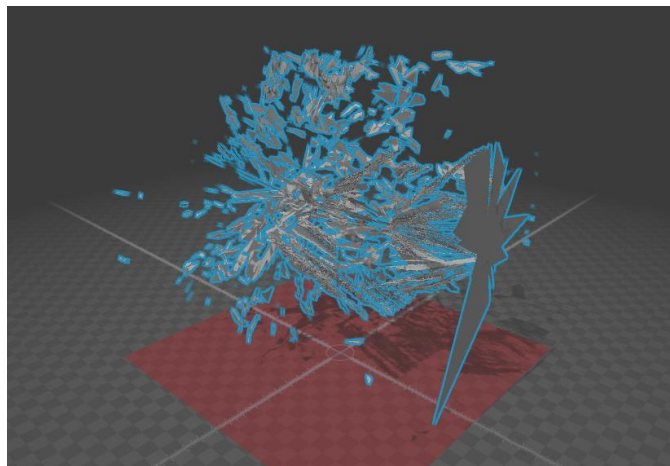


Figure 4. Output STL file

Here is the STL file we get after scanning the object. This shows that the hardware is working together with the software and producing raw data. However, this is not perfect due to lack of calibration or movement during scan. We can see there are many points connected together to create the whole STL file. The big gray part on the right is the surface of the turntable. The part on the left is the object being scan. Since the calibration is off, we didn't successful getting the exact shape of the object. We also need to find a better way to hold or stable the part when scanning the object. Calibration will be the most important part toward CDR, and we promise to get the exact shape

VI. CONCLUSION

By the 8th of December, Siyan has successfully assembled all necessary hardware and tested them. Vangjel has complied Freelss software and Chenkai has written a basic user interface. We also managed to output an STL file generated by Freelss. Such file proved that we have finished MDR deliverable. However, our team believe that following 3 months is very critical to our project. Hitting the milestone of getting a file does not ensure our success. Because of the fact that our device has not done any calibration, the STL file we have right now is nothing but a huge amount of random data, which is completely unrecognizable when comparing to the scanned object.

That's why the next phase of our project will be mainly focus on calibration. However, there are still things to be done for each of us before we all get to gather to calibrate the device. We have made plans till end of March as refer to Gantt chart. According to the Gantt chart, each one of us has an individual goal to achieve in about a week or two. Siyan will be focusing

on further polishing and organizing hardware as well as any hardware problems that remain unsolved before MDR. For example, the turn table is too slippery to hold an object. While scanning, the object moves while table is turning. Also, the stepping motor is stepping too strong causing displacement of the whole turn table.

The goal for Chenkai for the first week is to have our own user interface to call Freelss, and a mini tutorial, or help page on the user interface to briefly explain our software. As for Vangjel, according to the plan he will spend his time on create our project website on SDP 2016 website as well as design our team poster.

After finishing our individual project, we will all focus on the hardest and the most tedious part of this projecting – calibration. Freelss cannot output correct scanning result without appropriate parameter such as distance between two lasers, distance between laser and camera as well as distance between object and camera. Since we cannot view real time visual data while scanning, at least for now, we cannot determine which parameter goes wrong until we view the result on a different computer. We also need to perform various experiments to determine which parameter effect which part of the scan the most. Such as finding out what part or aspect of the result would be changing the distance between object and camera affect the most. We promise to deliver a printed out object, within +/-2 mm in terms of size, by the time of CDR.

- [5] Pololu.com, "Pololu - A4988 Stepper Motor Driver Carrier", 2016. [Online]. Available: <https://www.pololu.com/product/1182>. [Accessed: 17-Oct- 2016].

REFERENCES

- [1] L. Gilpin, '8 things you need to know about 3D scanners - TechRepublic', *TechRepublic*, 2015. [Online]. Available: <http://www.techrepublic.com/article/8-things-you-need-to-know-about-3d-scanners/>. [Accessed: 08- Dec- 2015].
- [2] 3dscanco.com, 'Benefits of 3D Scanning, 3D Scanners and Reverse Engineering', 2015. [Online]. Available: <http://www.3dscanco.com/about/3d-scanning/benefits.cfm> . [Accessed: 08- Dec- 2015].
- [3] LIM Innovations, 'Photogrammetry: Turning photos into 3D scans - LIM Innovations', 2015. [Online]. Available: <http://www.liminnovations.com/photogrammetry-photos-3d-scans/>. [Accessed: 09- Dec- 2015].
- [4] D. Lanman and G. Taubin, "Build Your Own 3D Scanner: Optical Triangulation for Beginners", *Mesh.brown.edu*, 2016. [Online]. Available: <http://mesh.brown.edu/byo3d/links.html>. [Accessed: 20-Nov- 2016].