GrandScan: A Portable, Scalable Laser Scanner

Noah Halford
nhalford@uchicago.edu

Catherine Moresco
cmoresco@uchicago.edu

Team Scanner Planner

June 3, 2016

1 Specific Aims of the Project

One possible use case for 3D printers is object replication. While it certainly is possible to design 3D models of real-world objects in CAD software, there are a number of drawbacks to this approach: it may require many iterations to get right, it may be time-consuming, and it requires a very particular set of skills. One alternative is to use a 3D scanner, which simplifies and automates the process of digitally fabricating the desired object.

The goal of this project was to design and build a 3D scanner that could ultimately be used to produce STL file approximations of real-world objects. Of course, because we were doing this cheaply, we did not anticipate results that might resemble those of an industrial-quality laser scanner: our attempt to create a laser scanner would be considered successful if we were able to produce STL files that were recognizable as the objects they represented. We did not expect that our resulting STLs would necessarily be completely faithful representations of the original objects.

As will be discussed later, many of the existing and easily available designs for DIY 3D scanners are lacking in either portability or scalability. This led us to attempt to create a laser scanner which would be both portable and scalable, as well as affordable.

From this project, we hoped to learn both how laser scanning works and what some of its limitations are.

2 Background Research

Laser scanning is a technology that has already been very well-developed; going into this project, we knew that it was unlikely for us to make significant improvements over existing designs. Our goals were to design a laser scanner that would be cheap and would work (as described in Section 1), but, perhaps most importantly, to learn about how laser scanners work. Instructions for a number of cheap DIY laser scanners already exist on the Internet, and we used these as inspiration for our project—although we did design our own hardware and software.

The two main preexisting DIY scanners that we usd as inspiration for this project were [4] and [1].

2.1 Approaches to design

We found [4] first and based our initial design off of this idea, because it was small and easily portable. However, in building and trying to use a scanner derived from this design, we decided that it was too small. In testing this design, we found ourselves wanting to scan larger objects; ideally, we would like to be able to scan a human.

The scanner presented in [1] is not scalable either, but it allows for the scanning of somewhat larger objects. However, this scanner is much less portable and much more unwieldy than [4]. We liked the concept of this design, but we were unhappy with the design itself.

It appears that there are no affordable, scalable, and portable DIY laser scanners. We sought to change this.

2.2 Approaches to measurement and mathematics

While designs vary wildly from laser scanner to laser scanner, the mathematical methods developed for the capturing of data and the construction of 3D models are fairly well-established; that being said, there are several different methods available, and it is interesting to note that each of these scanners take a different approach.

The Sarduscan takes the more canonical approach to laser scanning: that of triangulation. The laser is mounted a known angle to the line of sight of the camera; thus, distance from the camera to the target can be determined by the apparent location of the laser point in the camera's field of view. Laser locations are found by taking two images–one with the laser off and one with the laser on–and finding the difference between them. For a more detailed explanation of this, see [3].

In contrast, the Dentroman786 scanner doesn't use this approach, instead using a mathematical approach based on deformations of light more akin to structured-light scanning seen in devices like the Kinect. The angle between the camera and the laser is constantly changing; to find the location of the points in relation to the camera, we find the lines where the laser intersects the wall and the floor, and use these to define a basis for a coordinate transform to the camera's frame.

We experimented with both of these techniques in different iterations of GrandScan.

3 Approach

In our exploration of the different technologies and techniques used to create DIY laser scanners, our development cycle went through several iterations.

3.1 GrandScan 1.0

Initially, we modeled our laser scanner after the one described in [4]. We designed and constructed a laser scanner consisting of two webcams and tqo lasers between them mounted on a circular arc, with a turntable that was mounted on a stepper motor at the center of the arc. The target would sit on the turntable as it was scanned. At each step of the motor, each camera would capture an image; we the planned to combine data about rotation with a number of distance measurements (the distance between the cameras, between the cameras and the lasers, etc.) in order to produce a 3D model representation of the object we wished to scan using triangulation, as the Sarduscan scanner does. Our prototype used an Arduino attached to the stepper motor to control the turntable. A Python program communicating with the Arduino allowed us rotate the stepper motor (that is, the turntable) one step at a time, and take a photo from each webcam between steps. This produced 1024 images (512 per camera). After building a protoype of this scanner, we were dissatisfied with it for several reasons. Firstly, the configuration was not adjustable. The parts were 3D printed, and, after several tried, fit together nicely and looked quite nice; however, when actually scanning an object, we realized that we wanted to have the freedom to adjust the heights of the lasers and the angles of the cameras–something that our entirely 3D printed design did not allow. Additionally, the webcams did not offer the resolution we were hoping for. Their autofocus function was ultimately unsatisfactory, and the images obtained as data were blurry and poorly resolved; even to the naked eye, structure of the target was hard to discern–we were not optimistic about the capabilities of software to accurately generate a point cloud. Additionally, we realized that scanning on this level was–well–a little boring. After looking around for objects to test our scanner on, we realized that most objects that would fit on the print bed were rotationally symmetric, anyway (cups, salt shakers, bottles, etc); there were very few things that would fit on the 8-inch print bed that we would even want to scan. So, we decided to think a little bigger and begin a second iteration of our design.

3.2 GrandScan 2.0

Our second design was more heavily inspired by [1], using a similar laser-gantry-style scanning method and a similar mathematical approach. Ours, however, was much bigger and much simpler.

The design consists of three meter-long rods, joined at right angles at their ends by a 3D-printed joint. On the upright rod, there is a holder for an iPhone that can slide up and down to be of adjustable height. A gantry slides over one of the rods on the ground, on which is mounted the laser and the controlling Arduino. These three rods define a space of one cubic meter that represents the print volume; one "leg" is the track for the gantry and the width of the space, the other "leg" maintains a distance from the wall and provides support, and the third upright rod is a mount for the camera and represents the maximum printable height. Theoretically, a space of this size could even be used to scan a human–with eye protection, of course!

Mathematically, we also started from scratch, using the same technique as [1]. We first used a Hough transform to attempt to find the lines where the laser scans the floor and the wall; using these, we then transformed the other segments of the line using a coordinate transform to determine their location in our model.

We decided to switch to use of an iPhone to record data. This gives us the benefit of higher resolution than our cheap webcams, but also comes at the cost of not being directly able to transfer images to the controlling program, or to control the camera from the program. While, eventually, it might be interesting to write an app to manage this sort of control, we unfortunately did not have the time to do so. Instead, we recorded photos and videos from that fixed position, and then uploaded them to the computer for processing.

4 Technologies Used

Our project combined a number of technologies to create a laser scanner. In our initial design, we controlled an Arduino Uno in Python with the pySerial library; we also captured images with two Logitech webcams. We still ended up using the Arduino in the final design, but only because it was a convenient way to power our laser; any 5-volt power source would have been sufficient. When testing our initial design, we also found that the resolution of the Logitech webcams was too low to produce good results, so we instead designed a mount for an iPhone so that we would be able to scan with a higher-resolution camera. All of the image processing was performed with OpenCV.

5 Innovations

As discussed in Section 2, the concept of a DIY laser scanner is not new: a number of designs already exist on websites like Instructables. The three desirable characteristics of a DIY laser scanner are affordability, scalability, and portability. It appears that most preexisting DIY scanners have two out of three of these: we could not find any that had all three, so we created them.

One unique approach that we took was building a scanner that was highly modular. Many scanner designs on the Internet are designs for an entire scanner, meaning that it is not easy to make simple part swaps in order to change the scale at which the scanner can operate. Our scanner was designed specifically to come apart, so it is very easy to, with the same printed parts, change the scale by simply swapping out the rods. In fact, the advantage of the modularity of our design is twofold: it automatically gives us the easy portability that we wanted as well.

6 Results

We were able to successfully design a laser scanner that met all three of our goals: it was highly affordable, very scalable, and easy to transport. Not counting the Arduino, which, as discussed in Section 4, was only used as a power source and could easily be replaced by something simpler, or the iPhone that we used for image (actually, video) capture, this laser scanner is highly affordable: we bought two lasers from Amazon for under \$5, and the three dowels were about \$3 in total. The rest of the parts were 3D printed, so the scanner cost under \$10 total to build, taking into account an approximation of the cost of filament. This is more affordable than the designs shown in both [1] and [4].

Also as we had planned, our laser scanner is scalable and very portable. The only part of our design that limits the size of an object that can be scanned is the rod length: if one wants to scan larger objects, then larger dowels can be purchased. The entire scanner also easily comes apart into a few small parts which can be transported and quickly put together somewhere else; again, the limiting factor in portability is the size of the dowels.

The software was less successful than the hardware. Part of the reason for this may be that our image processing techniques were not highly sophisticated. As suggested by [2], we first produced a binary image by thresholding on the red value in each frame of the captured video, and then we performed the a Hough transform to determine which lines corresponded to the projection of the laser on the wall and the floor. However, the Hough transform had a tendency to find erroneous lines; we did our best to fix this issue, but in the end, it was not perfect, and the "floor" and "wall" that our software finds are not necessarily the true floor and wall. This may have led to some of our issues as well. [2] describes software in which, after the Hough transform is performed, the user is presented with multiple options for the "correct" lines. Implementing this likely could have improved our results.

We were surprised by the fact that the image processing used for producing scans was relatively simple: it essentially amounts to determining where the floor and the wall are and then performing a number of projections.

7 Lessons Learned

This project taught us about how laser scanners work, and about the image processing that needs to be done in order to successfully produce a 3D scan of a real-world object.

Something that we discussed while building this (but unfortunately at a time that was too late to change course) was creating a laser scanner that would be even more portable, by fabricating what would essentially amount to a smartphone case; the laser could then be mounted on the phone, and the phone could combine camera data with accelerometer and gyroscope data in order to produce a 3D scan. AutoCAD 123D Catch is software that has the aim of producing 3D scans by using this data, but, as we experienced during Lab 4, it has a multitude of issues. Perhaps using the laser scanner technique, rather than combining pictures, could yield better results. This would be an interesting approach to try in the future.

The main piece of advice that we would give next year's students would be to start early and to work on the software and hardware at the same time if possible. Before designing anything for our scanner, we took sample photos and videos in order to provide an approximation of the final result. This allowed us to divide the work and design the hardware and software in parallel, which significantly sped up the design process.

8 Conclusion

As discussed in Section 6, this project was a moderate success. On the hardware side, we were extremely successful: the design is simple, the total cost of the parts is very low, it is easy to scale, and transporting it is as easy as transporting the dowels that are used. We believe that, with these features, our laser scanner fills an important hole in the spectrum of existing DIY laser scanners.

Although we were not able to produce STL files or XYZ files (point clouds) that accurately captured the objects we wished to scan, the physical design of our laser scanner was successful and met our aims (on the hardware side). The only shortcoming of this project was in data processing. It is unclear exactly why this is, but the point clouds that we produced do not resemble the objects that we scan. The main reference that we used for implementing the software was [2], which had a tendency to be brief to a fault; it is entirely possible that a mistranslation from this document to code was responsible for our software issues. Perhaps with more time, it would have been possible to produce successful scans.

References

[1] Will Oursler. Make your own 3d scanner! `http://www.instructables.com/id/Make-your-own-3d-scanner/`, 2010. Accessed: 2016-06-02.

[2] Will Oursler. 3d scanner. `https://sites.google.com/site/oursler3dscanner/`, October 22, 2010. Accessed: 2016-06-02.

[3] George P. Pavlidis. Laser scanning using a laser pointing device and a camera. `http://georgepavlides.info/research/LaserScanningAndTriangulation.php`, 2006.

[4] Sardau. Build a 30$ laser scanner. `http://www.instructables.com/id/Build-a-30-laser/`. Accessed 2016-06-02.

9 Team Member Contributions

Noah Majority of programming, report

Catherine Design, OpenSCAD modeling, presentation

A Photos

B Code Listings

All of the code for this project (including OpenSCAD code, as well as some sample scanning photos and videos) is available on GitHub. Our Python program is `src/python/hough.py`.

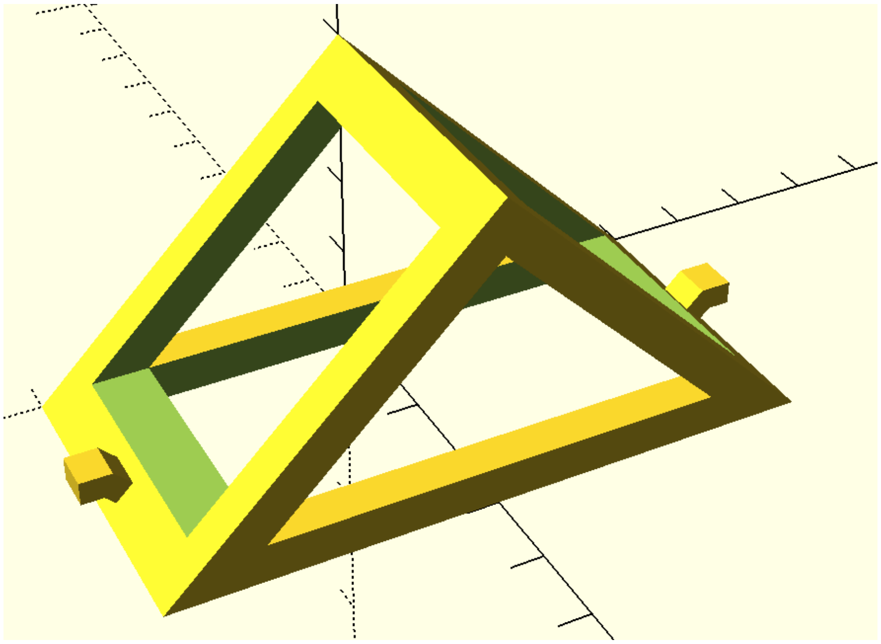

Figure 1: The SCAD representation of our original webcam mount.

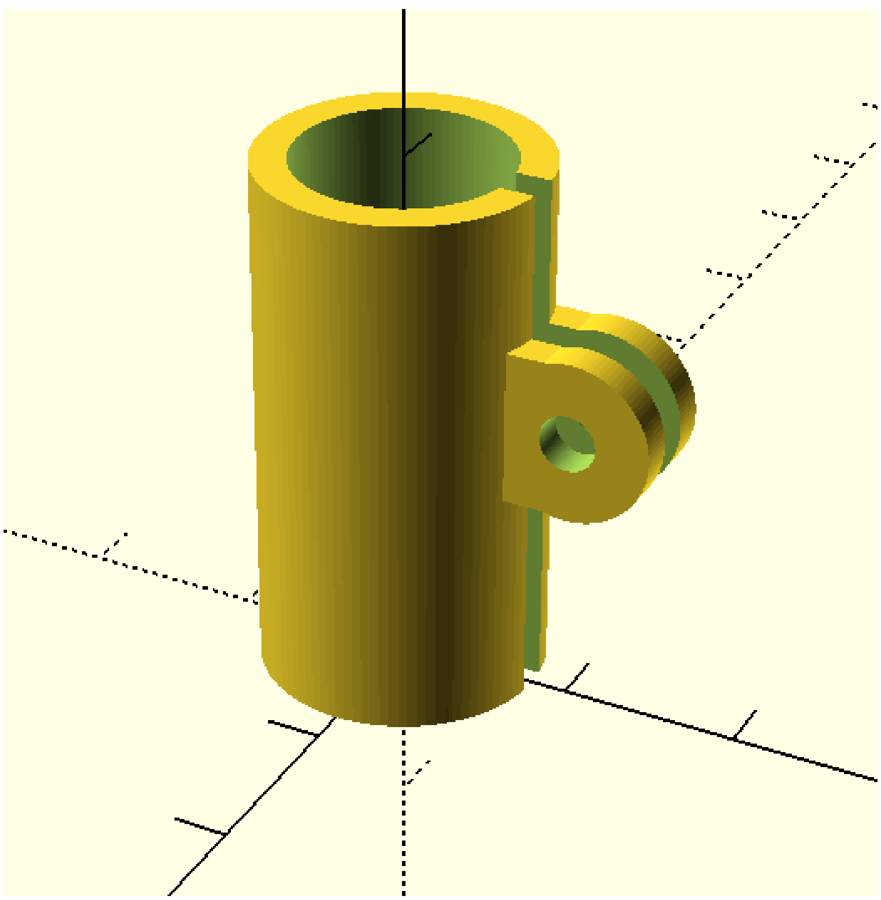

Figure 2: The SCAD representation of the laser housing.

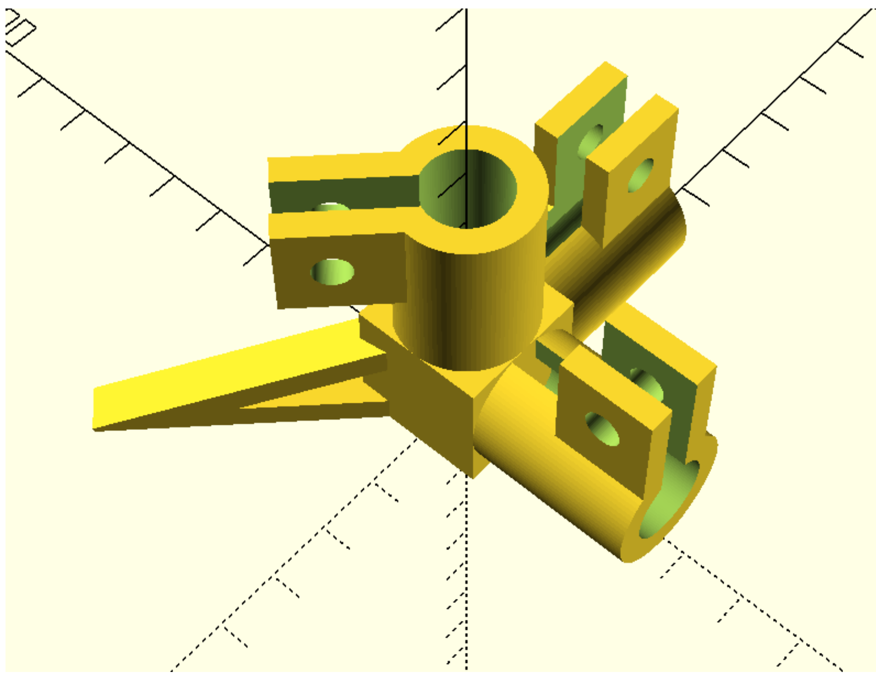

Figure 3: The SCAD representation of the corner piece which connects all of the dowels.

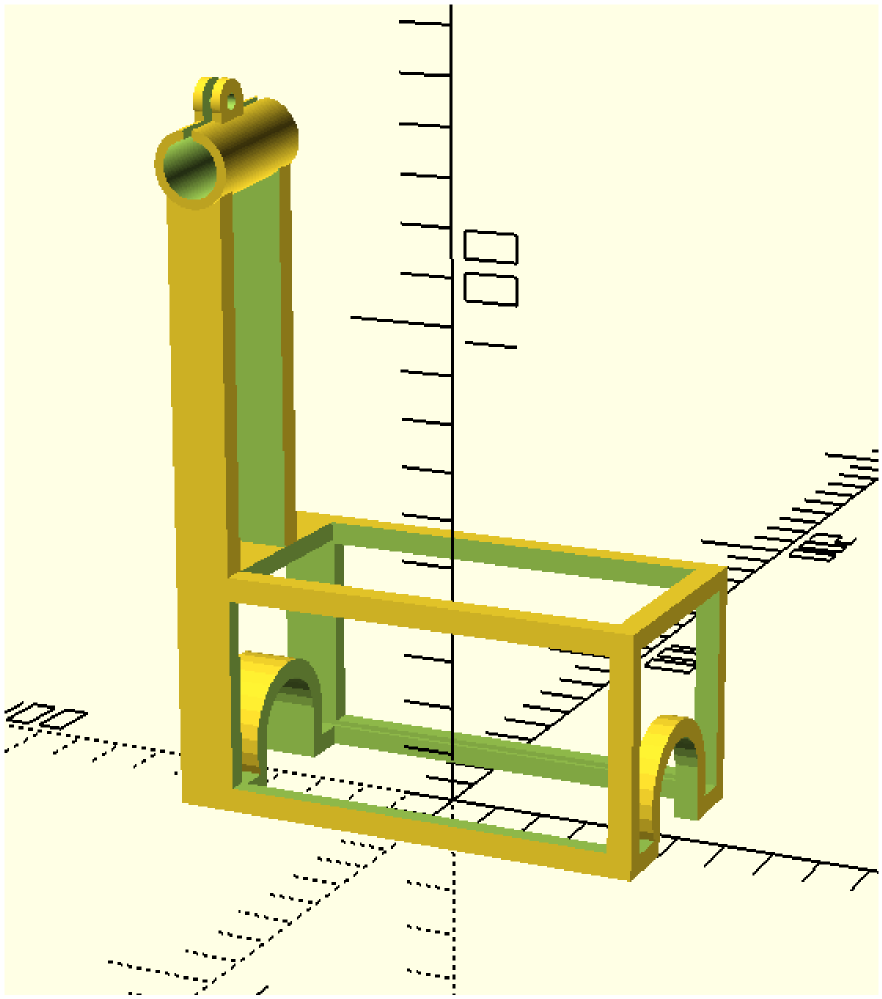

Figure 4: The SCAD representation of the gantry, on which the laser is mounted.

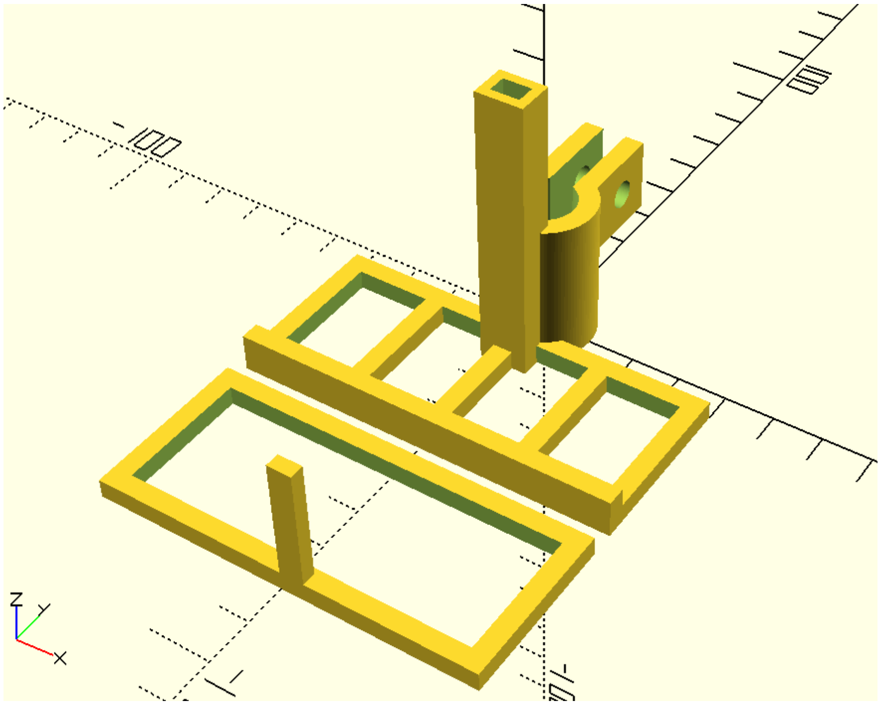

Figure 5: The SCAD representation of the housing for the iPhone, which replaces the webcams in our updated scanner.

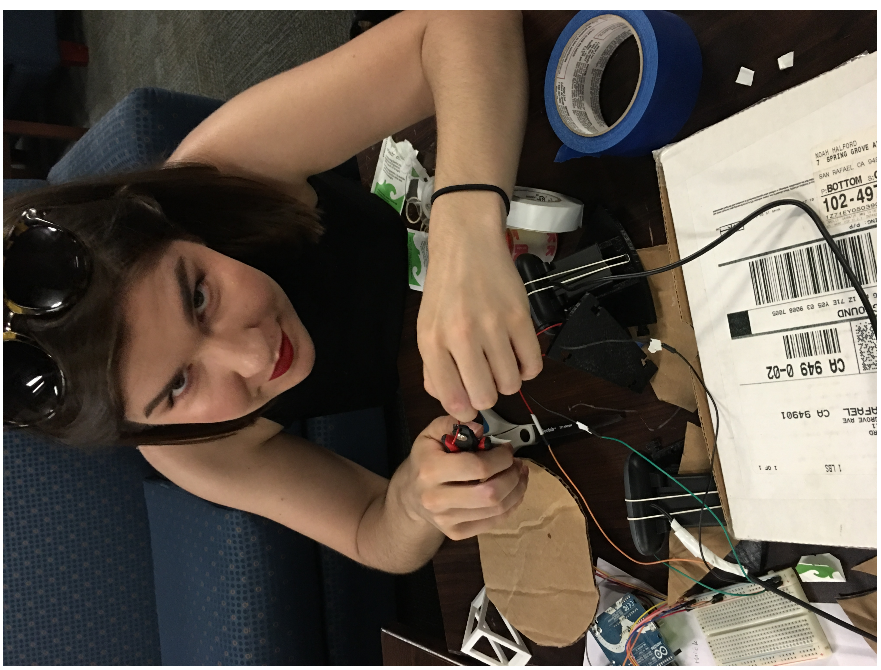

Figure 6: Constructing the first iteration.

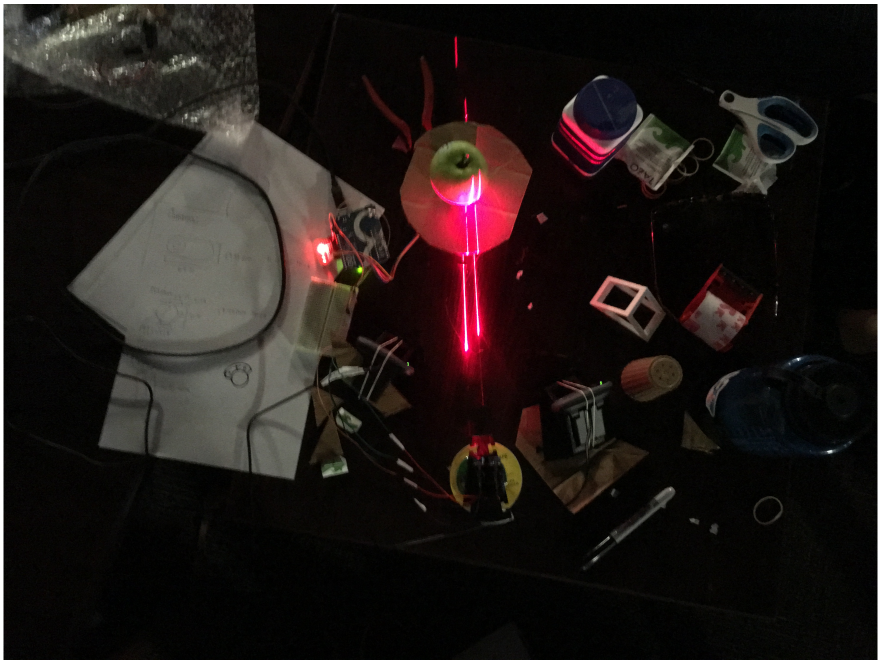

Figure 7: Scanning an apple with the first iteration of our scanner.

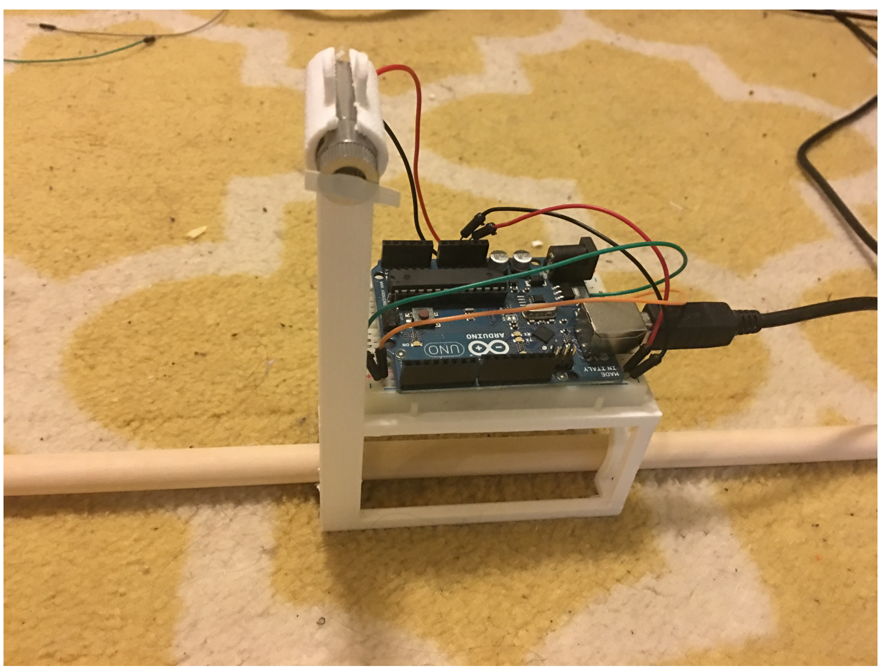

Figure 8: The gantry, with the laser and Arduino for power.

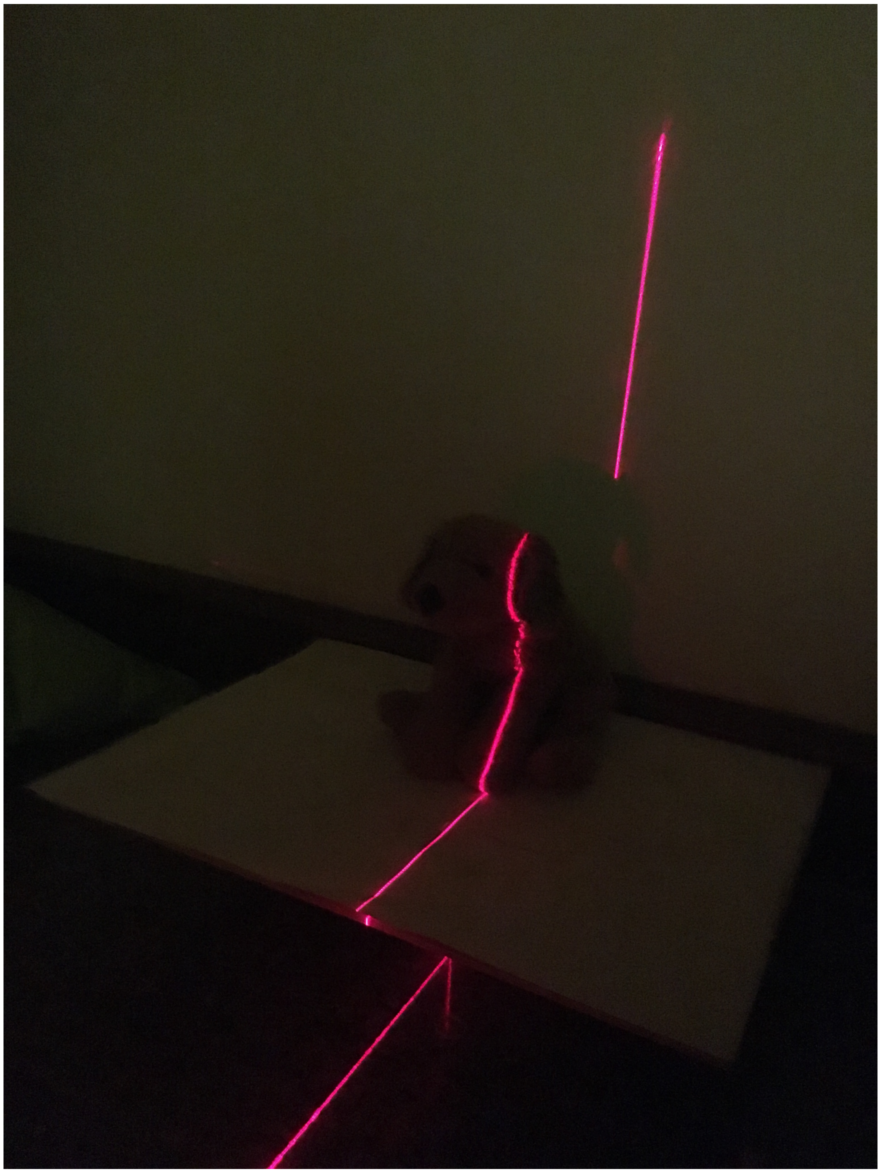

Figure 9: Scanning a stuffed dog with the second iteration of our scanner.

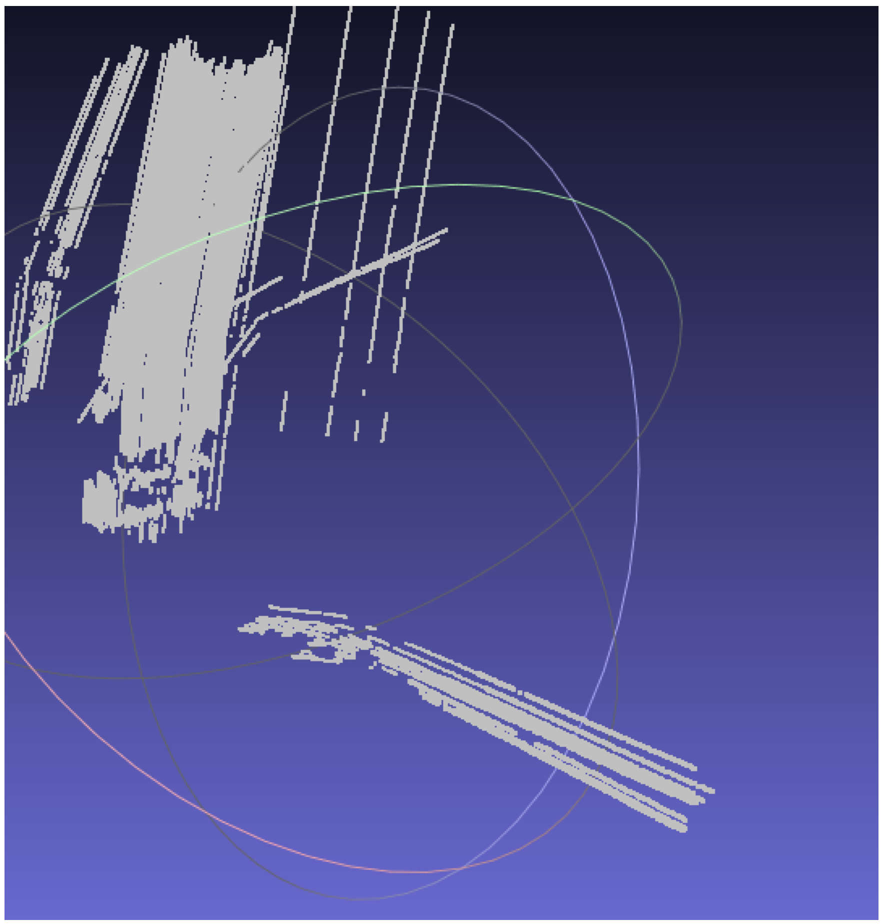

Figure 10: A point cloud produced from the scan of a cup. This is one indication that our code needs refinement: most of the data here is the wall or the floor, as the cup is relatively small compared to the volume that we scan.