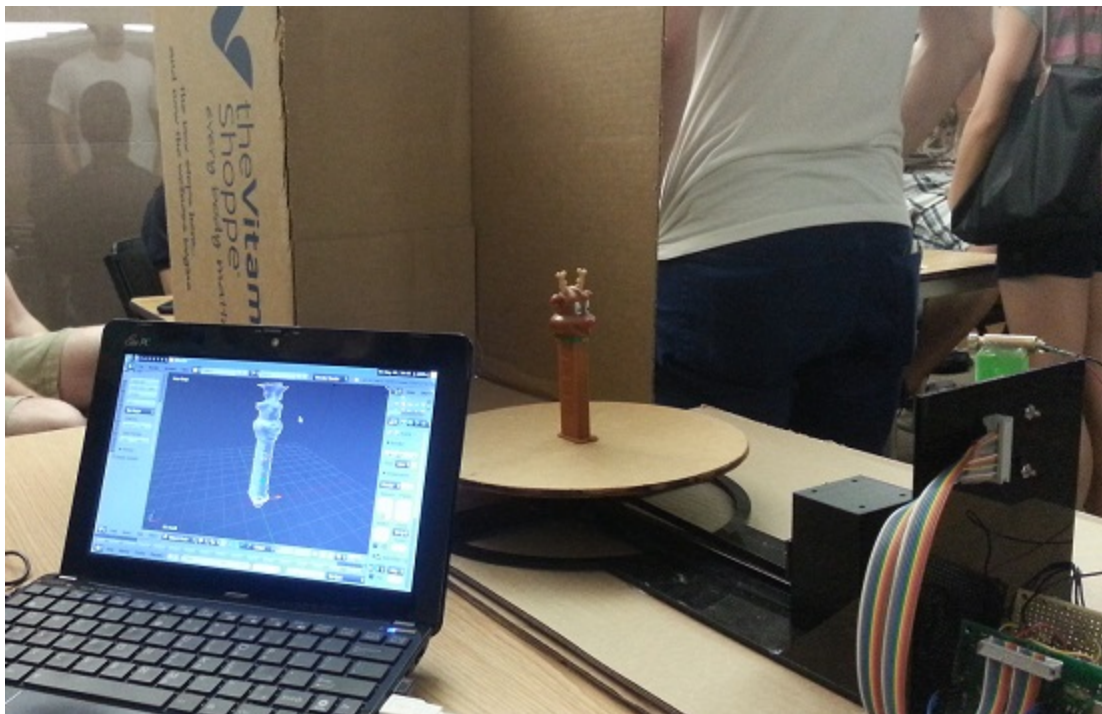


Team 19
May 12, 2015

Alex Franchuk
Madhav Iyengar
Miguel Sotolongo
Nathaniel Jeffries

SCANNERDUDE



Final Report

Abstract

The market for cheap, hobbyist-accessible 3D printers is growing and healthy, yet there is an absence of 3D scanners. As useful as prototyping and creating objects from three-dimensional models is, being able to scan existing objects into model files is arguably more useful, because this allows someone to take an existing object and make modifications and alterations, without needing to measure it and manually recreate it in modeling software. ScannerDude was designed to be a low-cost, yet fairly accurate, 3D scanner which would be useful for both hobbyist and industry applications. Through the use of camera feedback and lasers, ScannerDude triangulates data points and creates 3D models in the PLY format. For maximal ease of use, the scanned model file appears as if it were a file on a flash drive when ScannerDude is plugged into a computer. This makes it very easy (without any external software or drivers to install) for users to scan objects and immediately have a model file available to them.

Table of Contents	
Project Description	4
Design Requirements	4
Functional Architecture	5
Design Trade Studies	6
System Description/Depiction	9
Project Management	11
Related Work (Competition)	14
Requirements Met and Numerical Analysis	15
Conclusion	16
References	18

Project Description

We aim to create a functioning 3D scanner that has a work envelope of at least a cubic foot, while being generally affordable (from \$400-\$1000). The 3D scanner will be able to scan items within the work envelope, create a point-cloud of the scanned points, and finally provide an exportable file or data stream representing the scanned object. It will be targeted toward hobbyists and consumers, as its cost will not be extraordinary and its accuracy will be relatively reliable and practical for small- to medium-sized objects.

We also want the 3D scanner to be able to acquire color data, so the generated model can have a color mapping. The 3D scanner will be reasonably accurate, such that generated models will be quite similar to the real object. It will also be reasonably fast, since a 3D scanner that takes hours does not make much sense for the size of objects it will be scanning. The scanner should be easy to interface with, ideally using a USB interface, and hopefully registering as a USB Scanner or USB Mass Storage Device, so that it wouldn't be necessary for the user to install additional, specific drivers to immediately be able to use the scanner.

Design Requirements

The project has a number of requirements, which are itemized in no particular order below.

- *Cost*: Building and prototyping the project cannot cost more than the \$700 budget that is provided.
- *Safety*: The project should be safe to use. Thus, if lasers are used, they should be rated for reasonable safety (Classes 1 and 2).
- *Time*: The project must be designed and constructed in a 10-week period.
- *Scan Time*: The scanner should be fast, performing a scan of any object in at most 30 minutes. If necessary, model construction should be done in parallel with scanning to decrease overall scan time.
- *Physical Attributes*: The scanner should be a reasonable size and weight such that it can be put on a tabletop.
- *Power*: The power required by the scanner should not exceed that which is available from a 120V power outlet.
- *Scan Accuracy*: Each point of the scanned object's model data should be within 2mm of the corresponding point on the object, to account for possible unforeseen noise.

- *Autonomy*: The scanner should be autonomous, in that once it is set in motion, it will complete on its own, except perhaps minimal intervention to reorient an object.
- *Ease of Use*: The scanner should be as easy to use as possible. It should not be difficult for a user to immediately connect the scanner to their computer and scan an object, receiving the file of the object's data from the scanner.
- *Error Handling*: If an error occurs while scanning is taking place, the scanner should be able to gracefully degrade performance, or if urgency dictates, should alert the user of the issue. Ideally, this alert would be textual, however an LED indicating that the device should be reset may be adequate as well.

Functional Architecture

Functional components of the system are listed below:

- *Point Sampling*: The system must be able to select certain points of the object to be examined at a time, since it is infeasible to sample every point composing the object at once.
- *Point Data Gathering*: The system will need a way of taking a point or points selected by the sampling component, and getting location and color information for them.
- *Object Manipulation*: There must be some way of moving the object with respect to the data gathering and sampling components.
- *Point Aggregation*: A subsystem must exist which aggregates point data into a model, and preferably does some coalescing or smoothing of the data as well.
- *Export*: Point or model data must be exported from the device, preferably into a documented format that other 3D model software can interpret.

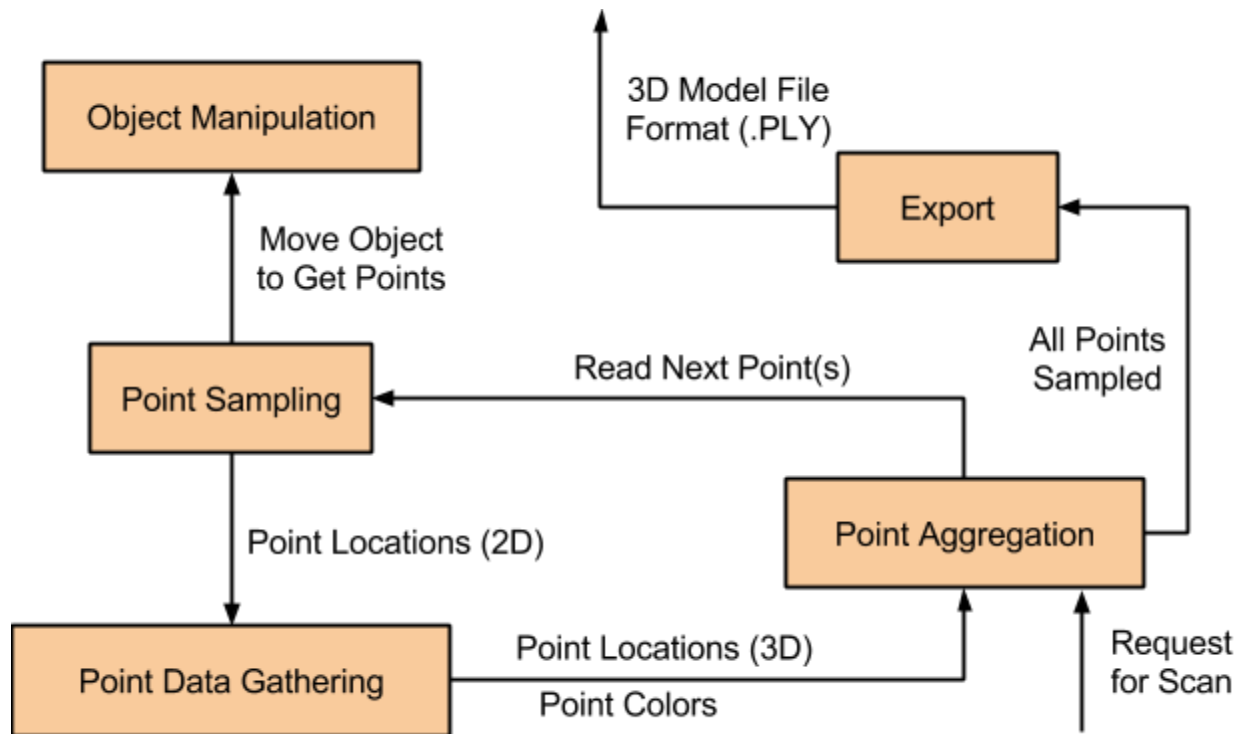


Figure 1: Functional Architecture of the 3D Scanner

Design Trade Studies

Design Trade-offs being considered:

- *Time of Flight vs. Triangulation*: Time of flight sensing is one way to achieve distance calculations at points on an object. The downside of time of flight measurement is that only one data point can be gotten at a time. It also requires highly accurate laser/sonar range finders which tend to be expensive. Triangulation allows for an entire line of points to be collected. Triangulation requires image processing to determine the outline of the object along the laser line. Triangulation requires fewer expensive parts and results in quicker scans, so we chose triangulation using a laser line over time of flight.
- *Structured Light vs. Line Scanning*: Structured light scanning involves projecting a pattern on the object being scanned. The features of the pattern are distorted based on the surface of the object, and a large number of points can be calculated from each scan. Line scanning takes points only along a single line at a time. Structured light scanning is much quicker to aggregate a large number of data points, but requires lots more processing and results in a greater volume of

data. Line scanning is simpler and less processing intensive, but takes a longer time to complete a scan. For simplicity sake we decided to use line scanning. In this way we avoid buying a projector, and we avoid potential issues of having too much data for our system to handle.

- *Construction Materials:* Several materials would be reasonable choices for the construction of our 3D scanner. First, we could machine and CNC mill aluminium parts to create the chassis. Alternatively we could use Polycarbonate, another easy to machine material which has the added bonus of being see-through so internal components are visible. Finally, in order to quickly and accurately prototype our chassis, we could use a 3D printed frame designed to fit our components. Since we are most focused on demonstrating the feasibility of our design, we have decided to use 3D printing wherever possible. This will allow us to potentially make multiple iterations of our design, as well as quickly prototype parts designed in CAD. A long-lasting and highly durable chassis, which could be achieved using a metal design, is lower on our priorities since we have only one semester to complete our project. We may use Polycarbonate and Aluminium for parts of our design for which durability or machining ease outweighs the benefits of rapid prototyping.
- *Rotating Platter vs Rotating Sensors/Camera:* 3D scanners mostly are based on rotation of the object being scanned. This can be done easily in two ways. Some rotate the object being scanned and others rotate the sensors and lasers doing the scanning. The advantage of rotating the sensors and lasers is that the object being scanned is less likely to move during the scan. The benefit of rotating the scanned object is simplicity and reduction in number of moving parts. We decided that it is best for our design to rotate the platter holding our object, since that will result in fewer moving parts.
- *Item Platter Rotation Through Direct Drive vs Gearing vs Belt Driven:* The platter holding the object to be scanned could be driven in one of several ways. First, a high accuracy stepper motor could be attached directly to the platter, driving the platter. Alternatively, the platter could be geared and attached to a geared down stepper motor. This adds some mechanical complexity but allows for greater resolution of platter movement. Finally, a belt can be used to drive the platter. The belt approach is good because it allows the motor to be placed in a more optimal location relative to the base of the assembly. The main

downside is complexity and possible faults related to the belt. We decided the geared approach is best, because it is a good compromise between mechanical complexity and accuracy. Gearing allows the platter to be controlled in a finer grained way, while minimizing the opportunities for moving parts to fail.

- *Choice of Processor/Architecture:* When it comes to the embedded processor used to control our design and process the data coming from our camera, there are many options which are equally valid. We need to choose a platform which is powerful enough to stream in video content and process it. This lead us to decide we will probably use a 32 bit ARM platform with sufficient clock speed to do the processing we need.
- *External Data Processing vs Internal Data Processing:* When streaming and processing point data, there are two key options. First is to stream raw data from our device out to a powerful computer and develop software to do the processing on the computer. Alternatively, we could do all the processing internally on our processor. The benefits of internal processing is reduced volume of data to communicate, and the ease with which adjacent line scans can be linked with polygons. (Each line can be scanned, and linked to the previous line.) The benefit of external processing is that the computer has more processing power, so it may be a faster option. We chose to do processing internally, because adequately powerful embedded processors are relatively cheap, and it is highly advantageous to develop a system which can plug and play with any computer, without having to install additional software to get the 3D object formatted data.

System Description/Depiction

Below in figure 2 is a diagram of the hardware architecture of our system. We haven't decided on the precise hardware devices for our scanner yet, however we have looked into different systems of turning the turntable, camera devices and their varying properties which may benefit our project, and lasers and how to make them into line lasers. We have also discussed which MCU to use for the project, arriving on the conclusion that a moderately powerful AVR MCU or an ARM would be best, and preferably one with a built-in USB interface (which isn't uncommon or particularly expensive).

We have looked into getting differently-colored lasers, so that we can use the laser lines to extract color data for the entire object. With a red, green, and blue laser, this would be trivial to convert to computer color data, assuming that we will be able to read color values with fine enough resolution. This is still to be determined.

We have determined that a CMOS camera would be best for our needs. However, it is difficult to set up a working camera system from a CMOS collector chip. For this reason, we plan to get a breakout board for a CMOS camera, such that we can command it to take an image, and read the image from it, without much hassle.

The turntable would best perform with a stepper motor attached by a gear to it to get fine-resolution movement. This is preferable to a belt system because, while the belt would filter out some motor noise, the gears are mechanically simpler and more reliable, and shouldn't introduce or propagate that much noise into the system, since we'll only be moving in one direction with very small steps.

3D Scanner Components

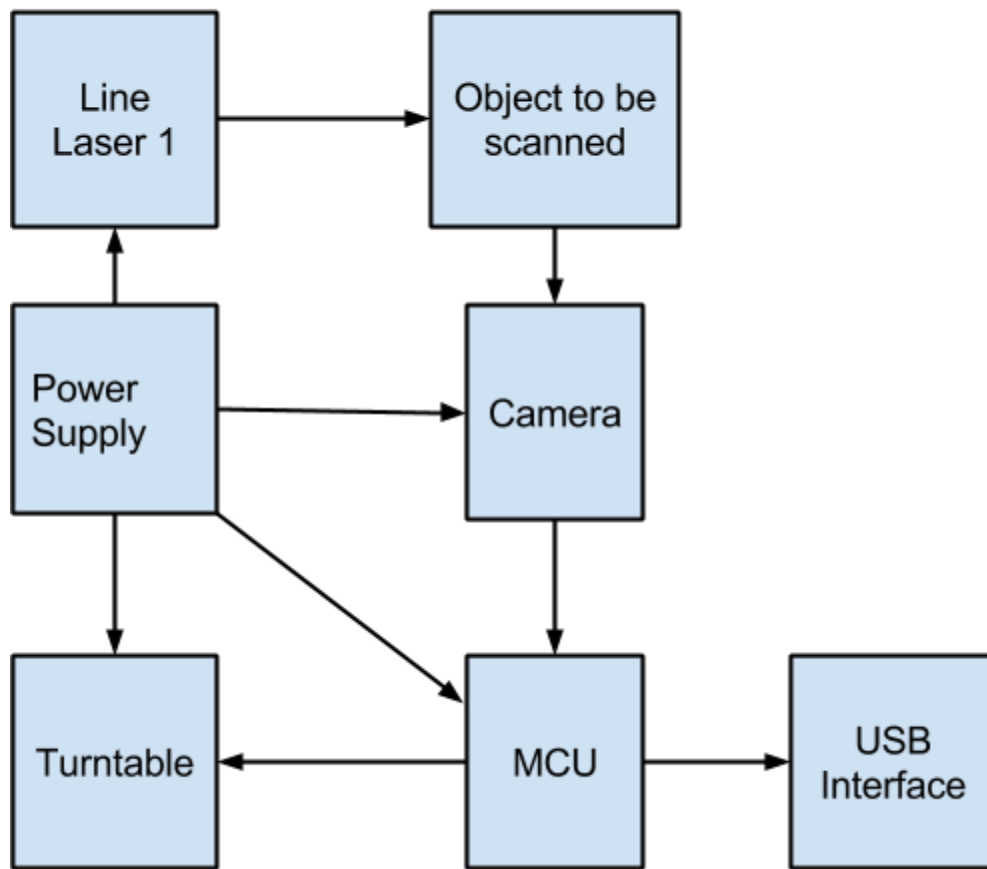


Figure 2: 3D Scanner Hardware Overview

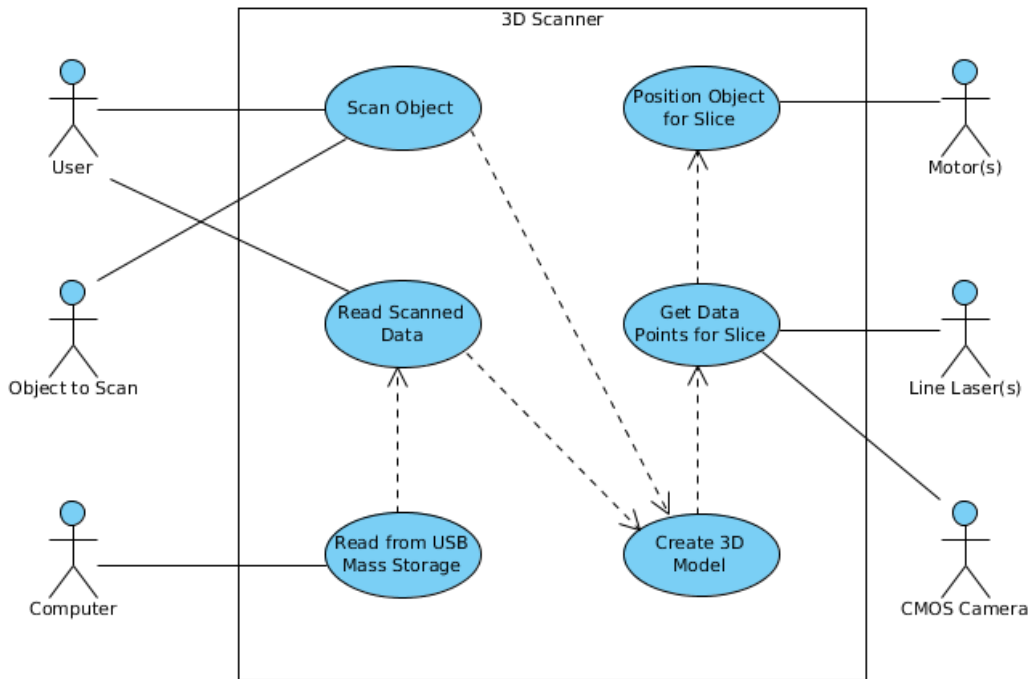


Figure 3: 3D Scanner Use Cases

Figure 3 shows a use case diagram for our 3D scanner, detailing the primary use cases of the system, and how they interact, both with each other and with external hardware or actors. The main interaction with users is to command the scanner to scan an object, and then to read the scanned object data from the device, preferably from a simulated USB Mass Storage Device interface.

Project Management

Schedule		
Iteration 0	2/20/2015	Project Website Check 1 Parts List for Prototyping
Iteration 1	3/6/2015	Final parts list submitted Work on Use Case: Position Object for Slice Work on Use Case: Get Data Points for Slice *Risk Check
Iteration 2	3/20/2015	Complete Use Case: Get Data Points for Slice Work on Use Case: Generate 3D Model Work on Use Case: Scan Object

Iteration 3	4/3/2015	Project Website Check 2 Complete Use Case: Position Object for Slice Complete Use Case: Generate 3D Model Work on Use Case: Read Scanned Data Work on Use Case: Read from USB * Risk Check
Iteration 4	4/17/2015	Complete Use Case: Scan Object Complete Use Case: Read Scanned Data Complete Use Case: Read from USB * Final Risk Check
Iteration 5	4/30/2015	Final Demo Project Website Final Check Last minute panic

Responsibilities	
Alex Franchuk	image reconstruction, image data acquisition, coding, documentation, USB interface
Madhav Iyengar	image data acquisition, image reconstruction, coding, documentation
Miguel Sotolongo	website, image reconstruction, coding, documentation
Nathaniel Jeffries	machining, data formatting, usb interface, coding, documentation

Total Spent		
Product	Vendor	Unit Price
Line Laser (x2)	AixiZ	\$26.00 (650nm) + \$60.50 (455nm) + \$38.50
Lenses	AixiZ	\$3.00 * 3
Stepper Motors	DigiKey (1460-1075-ND)	\$36.64
CMOS Camera	Aptina (MT9P401)	\$10.00
Structural Materials	McMaster-Carr	\$100.00
MCU	Mouser (Freescale MCU)	\$3.39
Custom PCB	??	\$89.52 + \$25.00 Shipping
Miscellaneous Components	Various	\$106.90
Total Spent		\$505.45

ScannerDude System Cost		
Product	Vendor	Unit Price
Line Laser (x1)	AixiZ	\$38.50 (650nm)
Lense	AixiZ	\$3.00

Stepper Motors	DigiKey (1460-1075-ND)	\$36.64
CMOS Camera	Aptina (MT9P401)	\$10.00
Structural Materials	McMaster-Carr	\$50.00
MCU	Mouser (Freescale MCU)	\$3.39
PCB	??	\$30
Misc. Components	Various	\$50.35
Total Cost		\$221.88

Risk Management	
Risk	Solution
Accuracy of Slice Data too Low	Switch to a higher resolution camera. Since this is a major component of our project it will be the first use case to be worked on.
Issues Associating Separate Red, Green, and Blue Laser Line Data at High Resolution	Abandon this feature. Intended to be an additional capability to set the device apart from competitors in a similar price range.
Difficulties with Moving Parts	Scanning the object should not be reliant on working motors. A system that positions the object for slice should not affect the progress we can make towards completing other use cases. This particular use case can be discarded as the user manually positions the object for slicing. This particular aspect of the project has the least priority from a design perspective.

Related Work (Competition)

- *MakerBot 3D Scanner (\$799)*: This baseline 3D scanner performs many of the tasks we are looking to implement in our scanner. Notably, it does not collect any color data from the item being scanned. Also, it relies on a software package to be installed on the computer it is connected to in order to make sense of its data.

- *Matter and Form's 3D Scanner (\$600)*: This is a cheaper alternative to the MakerBot scanner, and according to reviews it performs comparably. It relies on the same line laser scanning techniques as the MakerBot, and has the same shortcomings mentioned for the MakerBot, just at a slightly lower price. According to reviews it sometimes suffers from background noise if it is not placed with a solid color backdrop.
- *Fuel3D Scannify 3D Handheld Scanner (~\$1000)*: At a slightly higher price-point than the MakerBot, this scanner has the ability to perform high accuracy scans under optimal conditions. It is handheld and relies on stereoscopic imaging to map each angle of an object. It can scan an object from one angle in approximately .1 seconds. It has trouble scanning uniform color objects, objects with sharp corners, and any object which has obscured features from a single angle. This scanner fills a slightly different product niche than our 3D scanner would, being handheld and relying on stereoscopic imaging rather than laser triangulation. It is able to capture color information about a scanned object, like we hope our scanner will.
- *LMI Technologies HDI 3D Scanner (~\$15000-\$20000)*: A higher price scanner with extremely high definition imaging at high speed, it can take scans from each of six angles each scan taking only .88 seconds. Once all six angles are scanned, a high quality 3D object is available in multiple formats. The accuracy, ability to scan color, and speed of this scanner make it a great choice but its price range puts it outside the price range we are shooting for.

Requirements Met and Numerical Analysis

- *Cost*: Cost of the entire project and development was under \$700.
- *Safety*: Lasers and components used were safe for consumers.
- *Time*: The project was completed on schedule.
- *Scan Time*: The scanner takes about 2 minutes for a 10-degree step scan, and 8 minutes for a 2.5-degree step scan.
- *Physical Attributes*: The scanner is reasonably sized and is easy to carry.
- *Power*: The scanner uses a 120VAC to 12VDC wall wart, and draws about 1.25A from the 12VDC supply, when scanning.

- *Scan Accuracy*: Points are within 2mm of the target point (most noise is $\pm 0.5\text{mm}$).
- *Autonomy*: The scanner operates autonomously, and when done provides the resulting model file.
- *Ease of Use*: Since the scanner appears as a USB mass storage device, no extra software or drivers are needed for a user to be able to scan an object and get the resulting model file. The scanner is also mechanically sturdy, so the user doesn't need to fiddle with it.
- *Error Handling*: It is obvious when an error occurs, and the scanner is easy to reset if such an event happens (although in the final product this is a very rare, if ever happening, occurrence).
- *Average size of scanned file*: At high resolution (2.5-degree steps), the generated file of a large object is about 200 KB. Varying the object and the step size changes this, but only a small portion of the 8GB memory card is used.

Conclusion

During the course of this project we succeeded in developing a complete 3D scanner by pairing a camera with a laser line projected onto the object to be scanned. We custom designed our entire system, from the PCB layout to the device drivers, point cloud generation, surface construction and physical structure. We successfully met all of the criteria we set out to complete and demonstrated sub-millimeter accuracy, short scan times, and a low-budget solution for custom standalone 3D scanning. The product we developed has many desirable features, including platform independent plug-and-play usage, automatic .PLY file generation, and 8GB of on-board storage for generated 3D files. After a scan has completed, the system may be attached to any machine's USB port and it will appear as a USB mass storage device. Files can be copied and opened directly using any 3D rendering software. Upon startup, the scan completes entirely autonomously, meaning a scan can be started from a device attached to the scanner, and after the scan has completed, the resulting file can be copied onto one or more machines to be opened in modelling software. The automatic scanning process first generates a point cloud calculated based on the laser line position in the image captured, then the point cloud is used to calculate faces and generate a solid 3D object. This final step takes on the order of 1 second to complete at the end of the scan. The SD card used to

store the generated model file has 8GB of available storage space, so there is no object too big or detailed for our system to handle. While developing ScannerDude, our team not only completed a challenging and multi-faceted project involving lots of bare-metal programming and construction of a system completely from scratch, but we experienced the challenges all too common in the real world. We overcame a constantly developing set of hardware requirements resulting from trial-and-error testing by adding a second proto-board with additional components attached. We encountered difficult bugs where flash memory clocking resulted in occasional data corruption, camera data rates overwhelmed our processor, and damaged hardware failed to perform properly. We overcame these challenges through intensive debugging using a USBDM debugging interface we included in our hardware platform, along with hardware continuity testing and a logic analyzer to interpret malfunctioning communication lines. As with any large project, the final product we demoed represented a successful completion of our project, but only the tip of the iceberg with regards to lessons learned and experience gained.

References.

1. G. Bradshaw. (1999). *None-Contact Surface Geometry Measurement Techniques* [Online]. Available: <https://www.cs.tcd.ie/publications/tech-reports/reports.99/TCD-CS-1999-46.pdf>
2. "3D Scanner." *Wikipedia*. Wikimedia Foundation, n.d. Web. 11 Feb. 2015.
3. "Build Your Own 3D Scanner: Optical Triangulation for Beginners." *Build Your Own 3D Scanner: Optical Triangulation for Beginners*. N.p., n.d. Web. 12 Feb. 2015. <<http://mesh.brown.edu/byo3d/notes.html>>
4. Volume 21 (2002), Number 2 Pp. 149–1. *The 3D Model Acquisition Pipeline* (n.d.): n. pag. Web. <<http://www1.cs.columbia.edu/~allen/PHOTOPAPERS/pipeline.fausto.pdf>>.
5. "What's the Right 3D Scanner for You?" *LAGOA*. N.p., n.d. Web. 12 Feb. 2015. <<http://home.lagoa.com/2014/04/whats-the-right-3d-scanner-for-you/>>.