Final Report – Submitted 4/26/2018

Design, Fabrication, and Testing of a 3D Image Reconstruction Process

A Project Report
Submitted to the Faculty
of the
WORCESTER POLYTECHNIC INSTITUTE

in partial fulfillment of the requirements for

Major Qualifying Project
By

_____

Stephen Jendritz

_____

Logan Mendelson

_____

Brad Scuzzarella

Approved By:

_____

Prof. John M. Sullivan, Project Advisor

Abstract

This project involves optimizing a workflow for photogrammetry. This includes the redesign of an existing 3D scanning stage, the use of multiple Raspberry Pi cameras for image acquisition, as well as the selection of the best software and settings to use in order to render the 3D model. Once a 3D model is generated, it is brought into a model editing environment or sent directly to a 3D printer. Through the methods and product iterations explained in this project, the team was able to successfully create and 3D print realistic models of actual objects. The main objective was to develop a complete 3D image reconstruction package that is easier to use for the general user.

Executive Summary

3D image reconstruction is useful for many different disciplines. Typical methods include one or multiple cameras, and some techniques involve laser projection or LiDAR. This equipment is usually expensive and difficult to use effectively for an inexperienced user. Photogrammetry is convenient, but the quality of the resulting models are inconsistent when image capture varies. We've compiled various tools to standardize the process for home enthusiasts and to create viable 3D-printable models.

Photogrammetry is a method of 3D reconstruction which uses photographic cameras as the sole data collector. A point in an image corresponds to an observed point in the 3D world. From an unordered image set, photogrammetry software strives to orient the images to a global coordinate system and find corresponding points. Triangulation can then be done to acquire X Y and Z coordinates of the object surface.

Our project objectives were to design a workflow to simplify and optimize a 3-Dimensional model generation process, create a stage utilizing a rotating platform specifically for photogrammetry, select appropriate hardware for imaging and software for processing, and provide a low-cost solution for 3D image reconstruction.

The first step was to select a stage that had potential for modification. For hardware, we chose to use a Raspberry Pi and its camera for image acquisition. This allowed for easy programming of the capture timing and manipulation of image qualities. We experimented with multiple software products on the market to determine which option would perform best. We also described categories for an object's levels of difficulty to reconstruct. The levels "easy, medium, and hard" were defined by how complex of geometry and the variance of color that the

object had. An easier object would have detailed patterns or multiple colors, while a harder object would have a more uniform shape, texture, and color.

We modified a design to utilize a circular platform rather than a hexagonal one. We also introduced a solid white background to the scene rather than keeping the background unconsidered. All of this helps the software find better anchor points for image orientating. We then added a second camera for more convenience and improved the white background with a light box. This final setup returned the cleanest models after processing.

The final stage allows for a consistent method of imaging. The notched rotating platform and the light box is a simple solution to help generate quality 3D models. Using two Raspberry Pi cameras was for convenience alone, as it expedited the process of imaging multiple angles. Of the different software products we tested, Agisoft Photoscan Pro provides the most flexibility for image pairing, point cloud generation, and tiling/meshing. Results supported our object difficulty categories, as larger objects with more diverse geometry and colors tended to reconstruct more accurately.

Acknowledgements

Professor Sullivan - The team would like to thank Professor Sullivan for his help and guidance throughout the development and execution of the project. Professor Sullivan’s contributions helped make this project as successful as it was.

Contents

List of Figures

Chapter 1 - Introduction

3D image reconstruction is useful for many different disciplines. Typical methods include one or multiple cameras, and some techniques involve laser projection or LiDAR. This equipment is usually expensive and difficult to use effectively for an inexperienced user. [1]

Photogrammetry is convenient, but the quality of the resulting models are inconsistent when image capture varies. Various tools have compiled to standardize the process for either home enthusiasts and to create viable 3D-printable models, or for professionals looking at technology for reverse engineering applications. For example converting difficult real world geometry to a digitalized model or part.

The goal of this project was to solve the problem of 3D image reconstruction for a less experienced user. Generally, the techniques available today are expensive and difficult to operate without any prior knowledge of the equipment.

There are four main objectives for this project. The first objective is to design a workflow to simplify and optimize 3-Dimensional model generation. The next objective was to modify an existing image capturing stage to be used specifically for photogrammetry. The third objective for this project was to select the hardware and software the user could use for image capture and processing into a 3D model. The final objective is to make sure that the end results are low cost and accessible to all levels of users.

Chapter 2 - Background

2.1 - 3D Scanning

Optical 3D scanning is a rather new technology based on a very old idea. The principles of trigonometry were established about 2200 years ago by Euclid and Archimedes. [2] In the early 1980s, through years of theoretical analyses about the possibilities of scanning a 3-Dimensional object to compile an image of said object, the first 3D scanners of sufficient resolution were invented using triangulation techniques. [3] The two stationary objects of known locations are cameras that rapidly take pictures of the object, and the images can use trigonometry to determine the location of the object relative to the fixed points. Taking these two 2-Dimensional pictures, the scanners then use stereography to render a 3-Dimensional image of the original two pictures. [4] The use of two fixed points creates depth perception in the same way the human eyes do, also known as disparity. [5] Over the time of the scanner constantly taking pictures of the object, the scanner makes use of the archived photos and data to create a 3D point cloud of the object. [3] Another technique of 3D scanning comes from fringe projection. The way fringe projection works is an image, or fringe, is projected onto a surface from one angle. This fringe can either be a scrolling image or a consistent projection. The projection is then recorded by another camera from another angle, preferably perpendicular to the fringe, to take note of the topographical changes of the fringe due to the object it is projected on. [6] For example, if there is a bump in the middle of a flat surface, as the fringe goes across the flat surface, it will follow the geometry of the bump. The camera takes note of this and analyzes that the fringe is not as protruded as it was before. From this, it can be concluded that the surface isn’t topographically uniform. [7] Using frame by frame analysis, the difference in the length of

the lightwave projected on the flat surface and the length of the lightwave projected on the bump can be determined. This difference in wavelength is then the difference in topographical surface.

2.1.1 - History and innovation

The first scanners made use of hardware in which now would be considered barbaric. They used analogue video cameras and read the data on CPUs with miniscule RAM, as low as 64 KB. Also, the capacity of the storage devices was as low as 5 MB. [3] To put this into perspective, the Apple iPhone 7 released in 2016 has 2 GB RAM, and a minimum capacity of 32 GB. [8] In other words, the iPhone in your pocket has 31,250 times the amount of random access memory and 6,400 times as much storage space. Throughout the years, technology has adapted and innovated so much that the capabilities of these scanners have gone through the roof. While some researchers were looking into the ideas of triangulation, and other were looking into the tactics of fringe projection on topographical surfaces, the usefulness and capabilities of 3D scanners flourished in the following 30 years. [3] By the 2000s, technology had improved enough to supply the scanners with 200 MB RAM, further technological improvements equipped scanners with 64 GB RAM. [3] Video cameras were replaced by digital cameras with 256 KB resolution by the mid-1980s, and improved to 16 MB resolution by 2014. [3] Lastly, and equally as important as it is impressive, the capacity of storage skyrocketed up to 10 TB in the same amount of time as the other innovations. [3] This increased the amount of storage space to place captured images by 2 million times. All of these innovations pushed the limitations of 3D scanners further and further. The increase in processing power allows pictures to be taken more rapidly while noting more detailed information about each picture. The increase in resolution of the cameras used provides much more accurate scanning. Since the scanners rely on the resolution of the cameras to make the picture clearer, having increased resolution will increase the quality of the scan captured. [9] The storage of these devices allows

the scanner to keep track of much more data. Pictures of higher resolution take up more space than lower resolution pictures, due to them storing more data points. [10] The improvements in ram and camera power allow for many more frames of data per second. [11] Together, the increase in number of images and space that each of those image takes up means the extra storage space can be utilized to its fullest potential to acquire very high resolution and high quality 3D scans.

2.2 - Photogrammetry

Photogrammetry is a method of 3D reconstruction which uses photographic cameras as the sole data collector. Assuming pinhole cameras where all light passes through a single hole and projects onto the pixel array, a point in an image corresponds to an observed point in the 3D world along a single ray from the camera's center. When the same point on an object can be seen on multiple images from different camera orientations, a 3D coordinate frame can be produced. Triangulation can then be done using known camera orientations and pixel correlations to acquire full 3D coordinates of the object surface. [12]

"For photogrammetric 3D reconstruction the camera orientations are in fact only an unavoidable by-product, whereas the actual goal is to reconstruct 3D points." [12] In order to reference corresponding points in the triangulation process, an unordered image set with unknown alignments around an object must be oriented properly. The relationship in figure 1 below, illuminates how a camera forms an image of a scene; where x y and 1 form a pixel location, λ is a zoom scaling factor, K is the camera calibration matrix, R and t are camera rotation and translation matrices from the origin of a global coordinate system, and X Y and Z are object point coordinates. Photogrammetric methods first approximate the camera matrix P of each image (= [|]). [13]

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \lambda K \left(R \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + t \right)$$

Figure 1: Pixel location in an image of a scene coordinate

Algorithms have been developed to find point correspondences using known properties of the camera, projective geometry, and linear algebra and calculus methods. For example, projective geometry shows that one point on an image can lie anywhere along a line in a paired image. This is known as the epipolar constraint, as the corresponding point in the second image is called the epipole and the line it exists on is the epipolar line. When pairing images and finding corresponding points, the software use the epipolar constraint as a check in the process. [12][13]

Once images are oriented and corresponding points are found, the software begins with the 3D reconstruction. Methods of coordinate calculations from images alone are called Multi-View Stereo (MVS) algorithms. There are four general approaches to MVS: volumetric methods, surface deformation methods, patch-based methods, and depth-map fusion. In addition to these a technique called space-time stereo can be used, which projects structured light onto the object solely for the MVS algorithms to “grab onto.” [13]

Described below are some commercial and consumer software options that have photogrammetry capabilities.

2.2.1 - Agisoft Photoscan Professional

Agisoft Photoscan Professional edition is a commercial photo stitching software developed by Agisoft LLC beginning in late 2010. [14] This photogrammetry software is capable of bringing in images to be processed and configured into a dense point cloud, 3D model, tiled model, and orthomosaic models. The outputs in which the team is interested in are the dense point cloud, 3D generated model, and the finally textured, tiled model. This tiled model may then be exported from

the software as either a *.obj or *.stl file. The former is useful for viewing the generated model, while the latter is compatible with a majority of commercially available 3D printing software.

The *.obj file generated from this software is a type of geometry definition file that portrays the position of each vertex, the UV positioning on the model, and the faces that make up each polygon. The UV positioning is a process in which a 2D texture map of the model is wrapped around the textureless model. This information is defined as a list of vertices and texture vertices. [15]

The *.stl file, short for stereolithograph, that is generated is another type of geometry definition file that contains the unstructured, triangulated surface of a 3D model. Stl files are very widely used throughout rapid prototyping and are accepted by various software, namely Solidworks, AutoCAD and Creo. This is the most commonly desired file format used in 3D printing. [16]

Agisoft Photoscan Professional edition is a powerful and versatile software allowing for a lot of post image acquisition processing as well as post model generation processing. Once the images are imported into the software, a masking image may be added to the photoset as a whole or to each photo individually. The masking image is a picture of the background without the target object in the frame. This allows for much faster processing and point cloud generation. The masking process also allows for a much more accurately generated model by ensuring the background scene of the image is not modeled as well and merged with the target object. In addition, the user can set camera calibration settings if the values are known; otherwise, the camera matrices are created using metadata from the images themselves.

The use of Agisoft Photoscan Professional edition is limited by the team. A 30 day full trial version of the software is offered by the company in which the team may use to process the images gathered. Other means of using the software include buying a license for the limited edition of Photoscan for $179 or purchasing a full license of the professional edition of the software

for $3499. As the cost of the license for the professional edition far exceeds the budget of the project, the trial version or the limited version are the only viable options for using this tool.

2.2.2 - Autodesk ReCap Photo

ReCap Photo is another powerful image processing software developed by Autodesk. It was originally available as Autodesk ReMake, but was bundled with Autodesk ReCap Pro in late 2017. [17] Through Worcester Polytechnic Institute, the team has access to a 3-year student subscription to the service. Using the student version of the software, the team is able to import up to 100 images to be processed by the software and generated into a model. The software is also available in a professional version for $305, which would allow for faster processing of models as well as the ability to import up to 300 images to be processed. Use of the student version of the software would be adequate for this project.

While using ReCap Photo, all the team is required to do is upload the images to a project inside of the software. The images are then sent to AutoDesk on a cloud server and are processed by AutoDesk. Once the model is completely generated, it is sent back to the team to be downloaded onto a personal machine for post processing of the model. Post processing may include clean up and trimming of the model.

The model generated by ReCap Photo is a *.rcm file. This file type includes the point cloud used for generation, a textured *.obj file of the model as well as a *.stl file. This is all of the data that is desired by the team.

2.2.4 - SCANN3D

SCANN3D is a mobile application developed for the Android operating system by SmartMobileVision in late 2017. [18] This application allows for the images to be captured directly on the device in which the data processing is done. This application has a free to use,

limited version allowing for the lowest quality scans. Another option for this software is a $5.99 monthly subscription for further access of the applications features. This subscription cost would not be too much for the project to fund, so both options for using this software are viable for the project.

An interesting feature of this application is the image capture assistance built into it. While the user is moving around the object, the app shows anchor points on screen. The anchor points on screen turn from red to green when it is the correct time to take an image. This would be useful to understand what would be the ideal interval of rotation done by the device following each picture. Another useful feature offered by this application is fast model generation. The generation of the model is done right on the phone used to gather the images. There are three different options for model quality offered by this application, which are low, medium and high. The low quality scan may be generated in under 10 minutes, while the medium and high quality models process in about a half hour and an hour respectively.

This application may export models in the *.obj format. A downside to this application is the inability to export the file as an *.stl file for printing.The use of this application would require the model to be imported into another software for any post processing and needed modifications, as well as for a file conversion from *.obj to *.stl for further use of the model in the rapid prototyping field.

While testing out this application, the team ran into several errors during the processing phase. While generating a higher quality model, the application would oftentimes crash, causing the data that was generated to be lost. This would cause for a need to restart the generation of the model from the image set from scratch, without any guarantee that the application will not crash again. This problem was mainly encountered while generating medium and high quality models of the target object

2.2.5 - Qlone

Qlone is an iPhone and Android application that uses photogrammetry to Render 3D images. Similar to SCANN3D images are captured by walking around the object of interest, and once the object is fully scanned the processing is done on the device itself.

Contrary to SCANN3D however, the app uses video rather than individual images in conjunction with a patterned mat to gather its data. Qlone has a very user friendly GUI that is accessible to almost anyone. While using the app, the phone display shows a, “to be scanned” image using the patterned mat as anchor points, and as the “to be scanned” area is being scanned, the app generates a point cloud in real time. Once the user has fully scanned an object, Qlone asks if the user is finished or if the user would like to reorient the object to cover sides that may have been out of view. Once the entire object is fully captured the app finally meshes and textures the model to more precisely represent the object.

Qlone is a free application, however if the user wishes to download an.stl or a *.obj file they can pay a small fee of $.99 per export, $1.99 for 5 exports, $7.99 for 3 months of unlimited exports, or $24.99 for 1 year of unlimited exports.

2.3 - 3D Printing

3D printing has been one of the largest technological breakthroughs to hit the market recently. Originally, it was slower, less accurate, and could only print flimsy plastics. [19] Over the years, the innovation in this field was tremendous; the idea of 3D printing went from being science fiction to commercially available and affordable for someone as young as a college student. Thirty one years ago, in 1986, the first pioneers of 3D printing emerged. [20] This method of 3D printing was named Stereolithography, or SLA. SLA was patented by Charles Hull of 3D Systems, Inc and is still a very common method of 3D printing today. [20] It is relatively quick and cheap which makes it good for prototyping. This method includes extruding melted, liquid plastic, generally

PBT or ABS, onto a bed initially and done layer by layer until the part in completed. The file required for this printing type is a *.stl file, which can be exported from a majority of 3D modeling software.

Following SLA came Digital Light Processing, or DLP for short. This was introduced the following year and was relatively similar to SLA. Larry Hornbeck of Texas Instruments created this technology, which is useful for projectors and 3D printing. This process uses a digital micro-mirror to project light coming from arc lamps. This technology utilizes a liquid crystal display panel that is applied to the product each run. [19] This process is faster than the use of SLA, as well as using and wasting less excess materials. Fused Deposition Modeling (FDM) was the next technology developed for 3D printing in the late 1980s. [19] Invented by Scott Crump of Stratasys Ltd., FDM brought 3D printing to the next level. Prior to FDM, 3D printing was more conceptual than useful for prototyping. The products of FDM may be used for conceptual understanding of products as well as functional prototypes, and even end-use products. Like SLA, this process heats up and extrudes engineering grade thermoplastics such as ABS and PC onto a bed and builds a model layer by layer from the bottom up. [19]

Carl Deckard of Texas University developed the next pioneer in 3D printing technology in the late 1980s, titled Selective Laser Sintering (SLS). [19] [21] SLS is much different from the last three processes, notably in the fact that SLS has the capability of printing materials other than plastics. SLS has the capabilities of printing nylon, ceramics, glass and even some metals. Using a laser as a power source, the machine will form solid 3D objects inside of a powder bed. [21] This powder bed allows for the product to be produced without the use of supports, as it supports the printed material itself, saving on material costs. Following SLS came something very similar. Invented by German researchers at the Fraunhofer Institute ILT, was Selective Laser Melting, or SLM, in 1995. [22] Also utilizing a high powered laser, this technology takes metal powders and completely melts them and fuses them together. [22] From here, this process takes an SLA approach and produces the model layer by layer onto a metal plate. This technology can print

many metals such as stainless steel, titanium, cobalt, and aluminum. [22] Moving into the early 2000s, EBM, or Electronic Beam Melting was introduced. [19] This is a very expensive 3D printing technique in which metals and formed together in a powder bed as in SLS technology. However, it makes use of an electron beam, uses materials that are limited, and on top of all of that, it is rather slow comparatively. [23] Lastly, there is Laminated Object Manufacturing (LOM) which fuses sheets of laminates together. [19] These fused laminates are then either laser cut or machine cut with a knife. [24]This process isn't very popular, however it is very cheap and fast. The materials are plentiful, and the final products have wood-like characteristics. [19][24] While this method isn't as dimensionally accurate as a method like SLA however, depending on the application, it might be the best bet.

Although SLA was one of the first methods developed, it is the cheapest, most available, and most well-known method of 3D printing today. [19] The printers themselves have been improved tremendously over the past few decades, which in turn have improved the quality of the products they produce. [25]

2.4 - Hardware

Each camera option could be used with photogrammetry. The iPhone and the Android devices had available apps that could create the model directly, and the Raspberry Pi could possibly be programmed to perform the photogrammetry operations. Any unordered image collection about an object could be uploaded into a computer software in order to form a 3D model. The images taken would be moved from the device to the computer in order to use those pieces of software.

2.4.1 - iPhone

iPhones run on an iOS operating system developed by Apple Inc. The iPhone used in initial testing was an iPhone 7. This device features a 12 MegaPixel camera with 2 GB of RAM. [8] This device was used in a few different ways. Early on the application Qlone was used in order to generate a 3d model using the iPhone. Next the phone device was used to take pictures rotating around the object of interest. Later the phone was used in conjunction with the scanning stage in order to compare point cloud quality with what the Raspberry Pi cameras produced. In this final test the phone was stationary.

2.4.2 - Android

Android is a mobile operating system developed by Google. The Android device used in the testing of the SCANN3D software was a Samsung Galaxy s5 model G900T. This device is equipped with a 16 MegaPixel camera and has 2 GB of RAM. [26] The camera was not on stationary hardware as the app required the camera to be rotated. This may have caused some of the images to blur and give the software a hard time generating a model.

2.4.3 - Raspberry Pi

2.4.3.1 - Raspberry Pi 3 Model B

The Raspberry Pi processor is a great tool for developers. It has multiple accessories that can be attached and used for specific projects. Because the Raspberry Pi is python based for programming, writing scripts for imaging is relatively simple enough when following the camera module documentation. The team looked into open-source code for photogrammetry modeling to be done on the Raspberry Pi processor itself as well. There were examples of

multiple processors used together for imaging and modeling, but no specific photogrammetry package that could be utilized with the programming language.

While the Raspberry Pi is connected to the internet, it offers VNC connection capabilities. Using the Raspberry Pi to host a VNC server, it can be remotely accessed to run python scripts and see the images taken by the camera. Using OpenCV, a script was adapted from pyimagesearch.com to access the live video stream of the raspberry pi camera. [27] OpenCV is essentially a language that wraps C++ coding in Python wrappers. This is advantageous to the team due to the increased operating speed of C++ coding over Python scripts. [28]

The team looked into coding using both the Python and OpenCV coding languages. With the use of the python language alone, the raspberry pi is capable of taking pictures.

2.4.3.2 - Raspberry Pi NoIR Camera Board v2.1

The Raspberry Pi has its own camera manufactured to be used in conjunction with it. This camera module comes in both a typical RGB version as well as a NoIR filtered version. The NoIR camera was chosen as the team had already had them accessible. The Raspberry Pi 3 and the Raspberry Pi NoIR Camera combination was an option for taking images due to the picamera Python package. The picamera package allows for access to the on board camera module as well as provides a multitude of different camera settings that may be set in Python. [29]

The NoIR camera is an 8-megapixel camera, which is a high resolution for performing pixel operations. The camera board is easily mountable on the case of the raspberry pi to ensure steady operation as well. [30,31]

Chapter 3 - Methodology

Photogrammetry was chosen as the method for 3D reconstruction. The goal was for a basic user to utilize a cost-effective and simple workflow to obtain high-quality results. In order to achieve this, it was necessary to design and test the best and most reasonable options for an imaging stage, a camera, and a modeling software. For the imaging stage, the team began by searching for an existing design or idea as a basis.

When considering designs to optimize, the following functional specifications were used in order to guide the decision as well was how to make improvements.

Functional Specifications:

- A stage that would capture 360 degrees of images around an object.
- A stage that would be able to capture small scale items.
    - Max Base dimensions: 7 in. by 7 in.
    - Max Height dimensions: 7 in.
- A stage that would hold a camera.
- A stage that can be adjusted for multiple sized items.
- A stage that can capture multiple angles of an object.
- Generate a viable 3D model of only the object of interest.

3.1 - Original Designs Comparison

Three stages were examined and compared in order to decide on a design to optimize. Stages A, B, and C are shown below in figures 2, 3, and 4. The first design was a simple lazy susan and tripod setup. It would have been manually turned and the camera would have been placed on a tripod. This design has good potential to be modified because of its simplicity, however it would be harder to take precise measurements

Figure 2: Lazy Susan [32]

The next design is a 3D printable stage that implements a smartphone as well as a stage turned by gears. The camera in this design is at a fixed point. This design promotes the least amount of modification because of its complexity and fixed camera position. The size of the objects to be scanned is severely limited because of this design as well. The gears make precise measurement easier than the Lazy Susan.

Figure 3: Compact, Gear-rotate [33]

The final design includes a stage as well as a camera holding arm that can be adjusted based on the object of interest's size. This stage also features notches every ten degrees. This allows us to take images at precise angles. This design allows for the most flexibility as far as object size and potential modification.

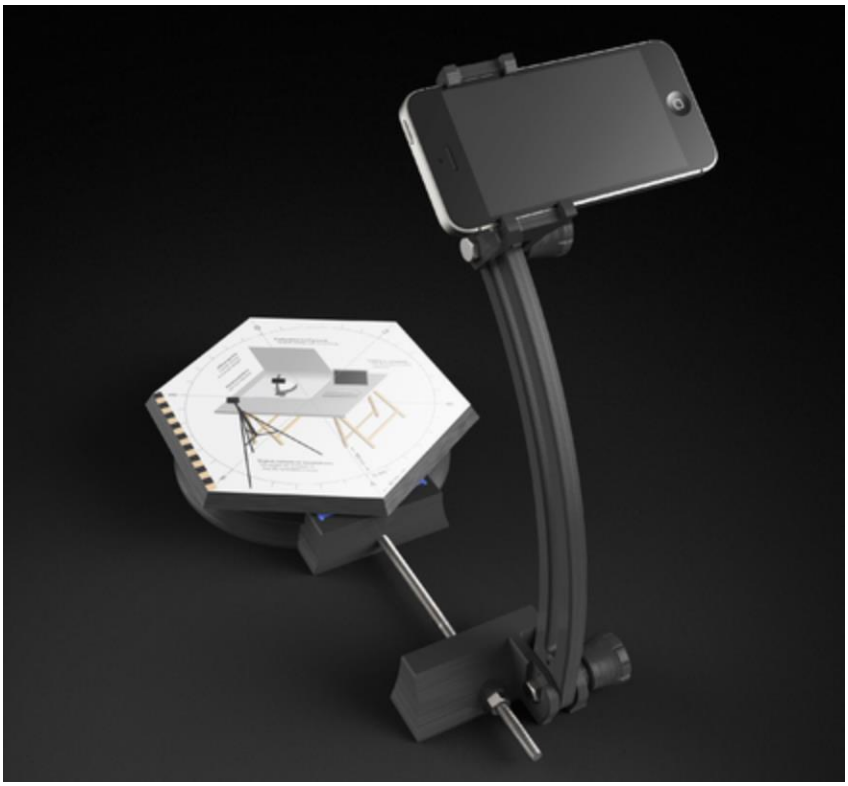

Figure 4: Camera arm, Manual-rotate [34]

3.1.1 - Chosen Design

Stage B was chosen as an initial prototype based on 3D-printability and potential for modification. Besides the 3D printed parts, the stage required marbles for the ring bearing, hex bolts and nuts, and a threaded rod to attach the camera arm. The camera arm is designed to mount a smartphone to take pictures. The clamp holding the phone can be adjusted up and down the arm to capture multiple angles of the object of interest. The stage allows for a

maximum object base of 6 in. by 6 in. and a maximum object height of 7 in. The stage also is notched every ten degrees making it easier for the user to gather images at consistent angles.

3.2 Alternative Designs

Other types of designs that were considered include rotating the object itself or revolving a camera arm about an object. A preliminary sketch idea was brainstormed for lifting and rotating the object of interest. It involved a stage platform with two brackets extending upward. Each bracket had a parallel bar with a soft tip extending toward the center that could compress together and grip the object as well as rotate. Ultimately, the team decided against this idea because it limited the number of scannable objects too drastically.

For a revolving camera arm, the team worked with the idea a bit more. While a rotating platform seemed to be a more typical stage design, a stage for a revolving camera could prove to be another viable method. A reasonable concept for a stage platform was modeled in SolidWorks CAD software based on the original chosen design, as seen in figure 5 below. Controlling the lighting and shadows for a design like that would be much more difficult then in a static camera setup, so the team decided to stick with the chosen design for improvements.

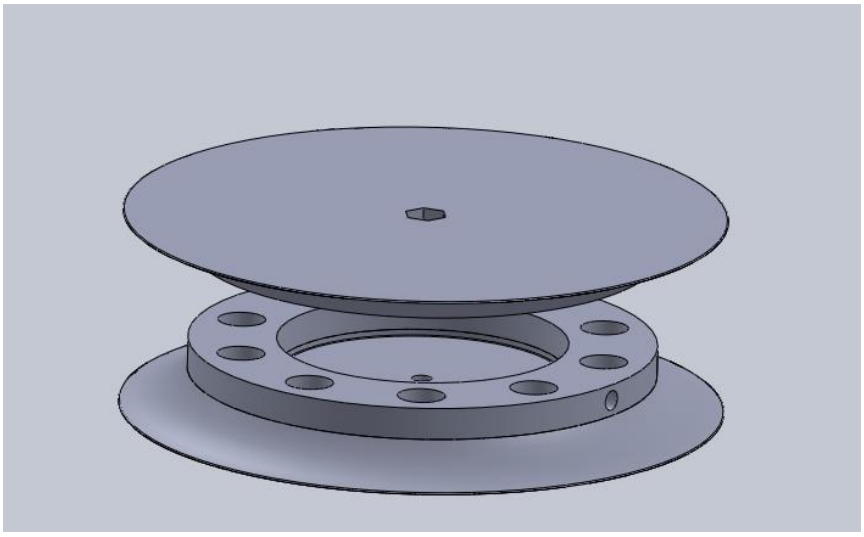

Figure 5: Simple example of static stage and revolving camera arm

3.3 - Design Optimization

Figure 6 below shows a model using only the stage as a setup. Using a tape dispenser as the object, we rotated the stage and took 36 total images.

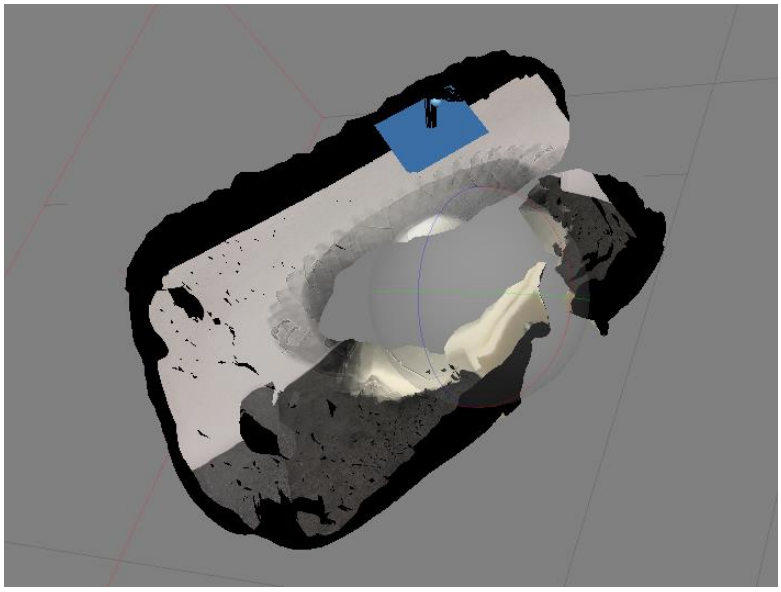

Figure 6: Model using original stage setup alone

In order to improve upon the chosen design, modifications were made to make the operation of the stage more user friendly, while increasing accuracy and speed of image capture for 3D model generation. The following design specifications were set into place:

Design Specifications:

- Rotating platform.
- 8 in. diameter staging table.
- Whiteout/light box to mask background.
- LED lights to eliminate shadows.
- Arm height 8 in.
- Light box 16 in. cubic.

3.3.1 - Modification 1

The first modification implemented to the stage was making the stage larger as well as more uniform. According to the functional specifications, the stage must be able to capture objects with maximum base dimensions of 7 in. by 7 in., therefore the stage was increased in size to an 8 in. diameter disk. This will allow objects in the target rage to be captured. This was done by laser cutting a circular piece of acrylic and mounting it to the already existing platform. In addition to increasing the stage table size, it was also made uniform in shape. This was done in accordance with the functional specification stating that the stage must only capture the object of interest. When taking images with the previous hexagonal stage, the stage would be changing in each image captured. This would prohibit the software from masking the background and multiple objects would be reconstructed. Figure 7 below shows an example model from this setup (with color removed).

Figure 7: Model from first optimization

3.3.2 - Modification 2

To further increase the ability to capture only the object of interest, it was decided that the background should be consistent in every picture. In order to achieve this, white printer paper was introduced below and behind the scanning stage as well as on top of the scanning table. This essentially whites out the background, keeping it consistent in every image. This will not only allow the software to focus on the object of interest and not get confused by a potentially changing background, but it also allows the masking or the background easier because of its consistency. Once the team saw success with this method, a light box was introduced to the staging, replacing the printer paper background. This will allow for a completely uniform background and forces the software the lock onto the object of interest. Furthermore, 2 led lights were introduced to the design to help eliminate unwanted shadows that could confuse the software. Figure 8 below shows a model from using this setup.

Figure 8: Model from second optimization

3.3.3 - Modification 3

In order to make the design more efficient, a Raspberry Pi and camera were chosen as the method of image acquisition. A python script allows for a time-lapse of photos to be taken rather than individually taking each image. This allows for quicker and more convenient imaging. The Raspberry Pi camera also allows those without high quality smartphone cameras to gather high quality images for photogrammetry. After seeing results, it was then decided that two cameras would be helpful. With this design, the user may gather more images per stage rotation, allowing for better point cloud results in half the time that it would with just one camera. See “Results” section for reconstructed models using this setup.

3.4 - Object Selection

Levels of assumed difficulty. Intricate objects such as the tape dispenser below in Figure 9 were deemed easier to scan and render because there is more geometry that the software can anchor to and compare in subsequent images. Objects with more uniform geometry and colors, i.e. a colored plastic bottle, or the Raspberry Pi pictured below in figure 10, are examples of a medium difficulty object. There is less for the software to help differentiate between images, therefore the object will be harder to render. Finally a hard difficulty would be an object with hard to distinguish geometries. For example a sphere or monotone cylinder. Objects like these look the same from every side making it very difficult for the software to anchor to a point and compare subsequent images. This is the same for the entire back of the turtle pictured in figure 11 below. While going around the back of the object, the images all seem to be of a red spherical object. Some of the photogrammetry algorithms favor more organic-looking objects with smooth surfaces, so objects with sharp angles also could prove difficult to model.

Figure 9: Easy - “Tape dispenser”

Figure 10: Medium - “Raspberry Pi”

Figure 11: Hard - “Plastic Turtle”

3.5 - Software Selection

Part of the methodology was to test and compare different software. Described below are four products that were tested including smartphone applications, consumer software, and commercial software.

3.5.1 - SCANN3D

Through preliminary testing, the software to be used for the photo stitching was narrowed down. Beginning with SCANN3D, the team tested the software with two different objects, a chair and bottle. Starting with the chair, figure 12 below, the team generated a photo set while walking around the object. The photos were gathered with the camera on a Samsung Galaxy s5 in an amply lighted environment. The medium quality model generated is represented by figure 13 below along with the model viewer packaged with the application.

Figure 12: “Chair” Object

Figure 13: SCANN3D model of chair

The model that was generated was not bad, as it had the basic shape of the chair, but it was not extremely accurate in terms of quality. Following the generation of this model on both the low and medium settings, the team became aware of a bug in the software that caused the app to crash when generating higher quality scans. This bug was inconsistent, but apparent nonetheless. An Arizona Green Tea bottle was also used as a test model. This object would consist of harder geometry to model, due to the fact it is not very irregular. After multiple attempts at trying to construct a model of the bottle, none were successful.

Though the convenience of having the image acquisition and data processing take place all on one device through a GUI would have been a very easy and user friendly way to take a scan, it was determined that this app was not sufficient enough to be used by the team moving forward. With the constant possibility of the app crashing mid processing, along with the quality of the models being mediocre at best, it was not worth the time to continue processing with this app, as there were better options available for the team to use.

3.5.2 - Qlone

The next software looked into was Qlone. The object used to test this application was a table plug, shown below in figure 14. The device used to capture the object was an iPhone 7. The Table plug was set on the Qlone patterned mat shown in figure 15 below, in the middle of a table in a well lit room.

In figure 16 the result of the scan can be seen. The image is better than that of SCANN3D, however, there are still noticeable flaws and small details are left out. A mechanical pencil was also scanned with similar results.

Besides the less than optimal image quality of Qlone, there is one major flaw of the app and that is the fact that it only works with the patterned mat. Because of this, there is a limitation of what objects can be scanned and the size of the objects chosen to be scanned.

After these limitations were realized, the team decided to continue to find a more robust software for model generation.

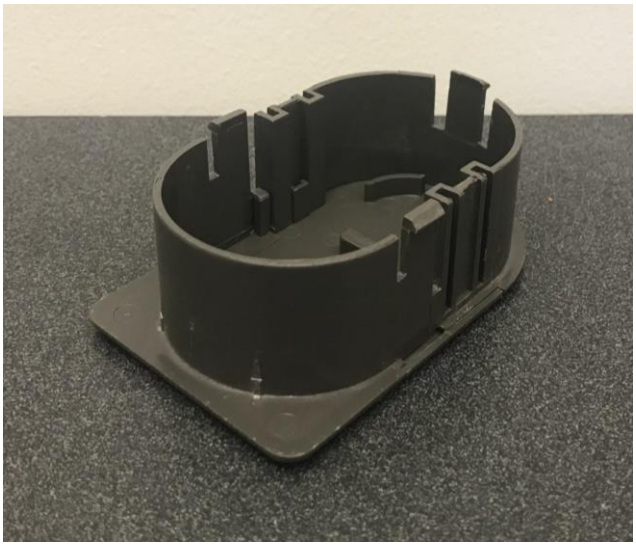

Figure 14: Table Plug

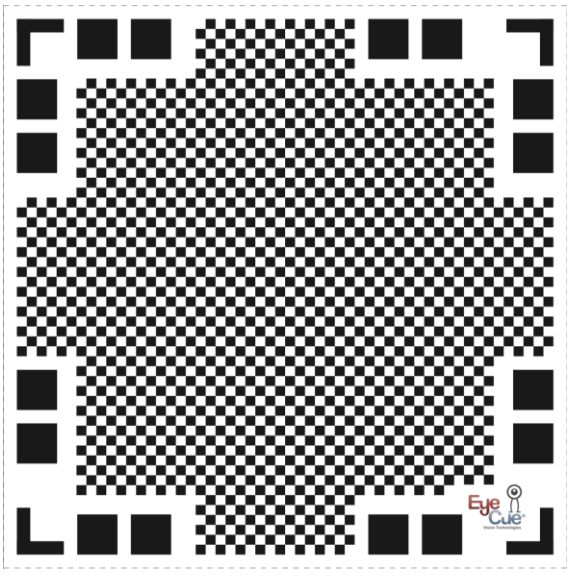

Figure 15: Qlone Mat [32]

Figure 16: Table Plug scan

3.5.3 - AutoDesk ReCap Photo

Preliminary testing of this software showed that it was easily the most user friendly of those tested. All that is required by the user is for the images to be uploaded into a project folder. Once the images are updated into the project folder, the images are sent off to AutoDesk and held on a cloud server. Once the images are processed into a model, the resulting ReCap model is sent back to the user to download and perform any post generation modifications. The most useful of these modification tools gives the ability to cut directly along a plane. This allows the user to cut out any portion of the model that may have been generated, but is not desired. An example of this would be the base in which the target object rests. During the construction of some models, it is added in along with the target object.

During testing of this software, a model of a tape dispenser, figure 17 below, was generated from photos taken with an iPhone while walking around the test object. The quality of the scan was very impressive compared to those obtained before.

Figure 17: ReCap model of tape dispenser

The downside to this software is the fact that the models are not generated locally. Due to the use of ReCap being on a student subscription, the models are placed in a queue with the rest of the models uploaded to the student server. This server is much more crowded than the professional server. Due to this, it is hard to predict how long a model will take to generate. There is no indication from AutoDesk when the model will begin to be processed.

Even with the downside of the models being generated on a server, the team has decided to move forward using this software to generate its models. With the accuracy of the model generated above, it was clear that this software is capable producing the results desired.

3.5.4 - Agisoft Photoscan Professional

Throughout preliminary testing of this software, the team was impressed with all that was available for the user. Compared to the other software programs being looked into, this was the only one in which you can do any processing with the images prior to generating a point cloud of the model.

One really useful feature of this software is the masking feature. Since a turntable is being used to rotate the test object in one location, there will be a static background throughout

all photos taken from the same orientation. If an image of just the background is taken without the test object in the frame, this image may be masked over the other images. This mask helps the software detect exactly what is the test object in each picture, based on the background image. This causes the scans to be more accurate, as the software knows not to blend the background into the target object. Utilizing this tool, the software is capable of distinguishing what needs to be modeled and what is irrelevant.

Objects used for preliminary testing of this software include a tape dispenser and an Arizona Green Tea bottle. Comparatively, the scan of the bottle came out just as poor as it did while trying to model it in SCANN3D. Upon further use of the software however, it was determined that the image set used to model the bottle was of poor quality, which was the reasoning for the poor quality resulting scan. Figure 18 below depicts the model of the tape dispenser, which was far more accurate than the model of the bottle.

Figure 18: Tape Dispenser model from Agisoft

Another useful feature of Photoscan Professional is that all of the processing is done locally. The photo sets are not uploaded to a cloud to be processed on a server, unlike AutoDesk ReCap Photo. This is desirable for the team due to the processing time being much more predictable. Immediately following importing the image sets into the software, they may be processed at our will. Along with the use of the Batch Process function within the software,

models may be generated by scheduling out step by step what processes to perform on the photo set. For the purpose of this project, the batch process includes aligning the photos, generating a sparse point cloud, generating a model, generating a texture, and finally wrapping the texture around the generated model. Along the process, the model may be checked to ensure there are no errors during its generation.

Due to the versatility and multitude of features offered by Agisoft Photoscan Professional edition, the team has decided to move forward with using this software to generate models along with the stage. The capabilities of this software far exceed those of the ones on either smartphone application, which was expected.

Chapter 4 - Results and Discussion

4.1 Scanning Stage

The final stage, shown below in figure 19, allows for an easy and consistent method of imaging. A slotted arm is used in order to capture images of objects at multiple different angles. This not only allows for more detail to be captured but also increases the amount of images available for use for 3D reconstruction, which increases the accuracy of the overall model. The stage also features a threaded rod that is used to adjust the working distance of the cameras. This helps capture different sized objects more precisely. The notched rotating platform ensures the user is taking pictures at equivalent angles so virtually every perspective on the object of interest is captured. The light box is another simple solution to help generate quality 3D models by ensuring there are minimal shadows that would otherwise interfere with the modeling software. Finally, using two Raspberry Pi cameras was a convenient addition, as it expedited the process of imaging multiple angles simultaneously.

Figure 19: Final Stage setup

4.2 Final 3D Models

Shown below are fully rendered models of selected objects. Observations to note include accurate dimensions relative to major entities as well as angles and textures. Our final setup allowed for models that are exceptionally more realistic than previous versions.

Figure 20: Recap model of Tape Dispenser

The model pictured above in figure 20 was the model generated from AutoDesk ReCap. The model generated by recap is much smoother and more accurate than the model produced by Agisoft, pictured below in figure 21. The Agisoft model is very pixelated and missed out on modeling the bridge of tape. The model generated by ReCap is much more printable in this case, and is also of much better quality. This shows that even with the ability to edit the photos before they are processed by the software may not be the best case in every scenario. This may also be due to the fact that Agisoft is better used when rotating the camera about the object and not the other way around.

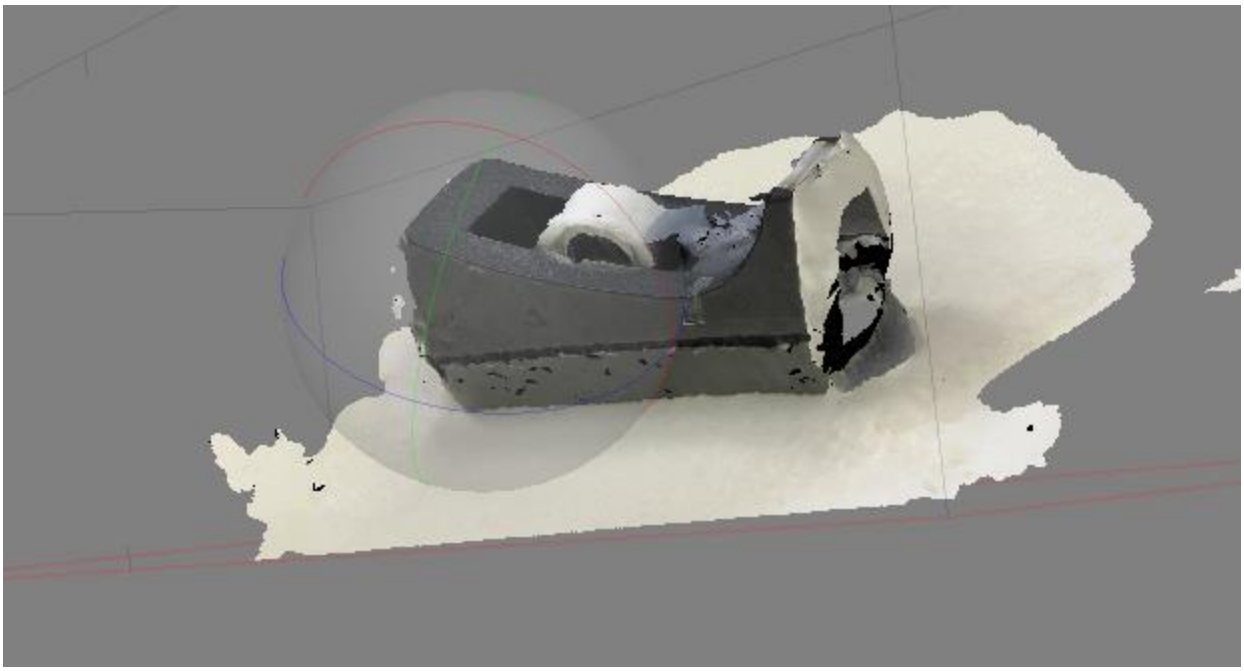

Figure 21: Tape Dispenser model from Agisoft

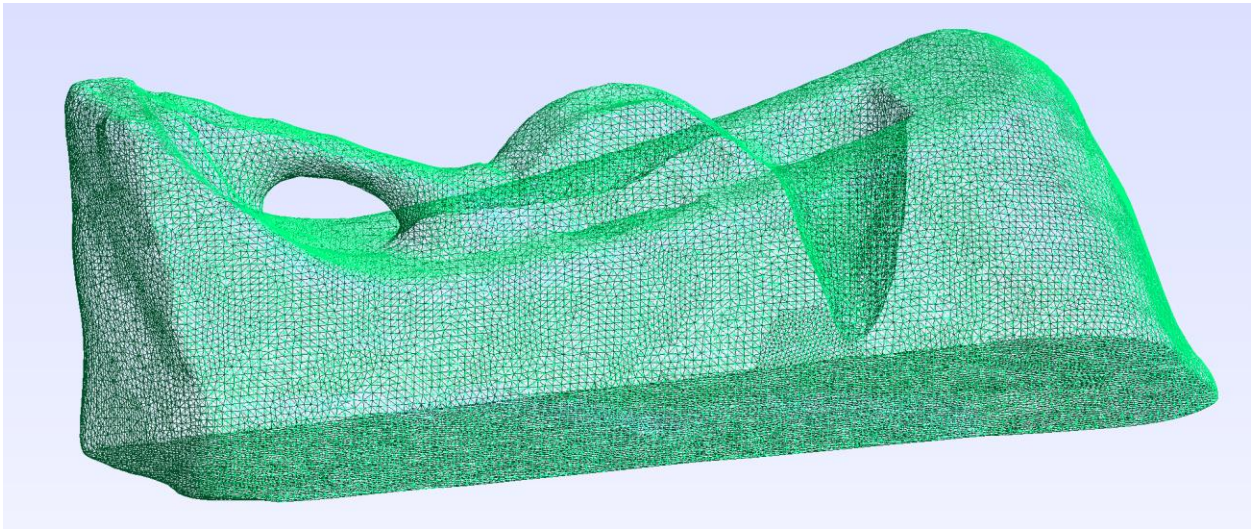

Figure 22: Mesh of ReCap model of tape dispenser

The model pictured below in figure 23 depicts the best model of the Raspberry Pi that was generated. This model was also generated with AutoDesk ReCap Photo. The models generated by ReCap tend to be smoother and sharper. Although the texture was less defined, the overall geometry of the model was better generated through ReCap.

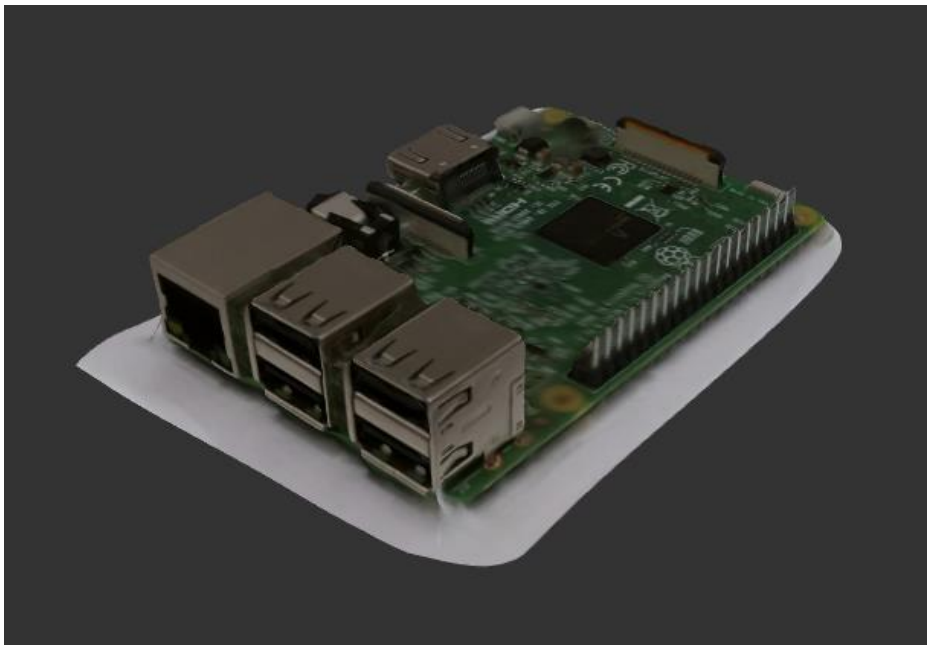

Figure 23: Recap model of Raspberry Pi

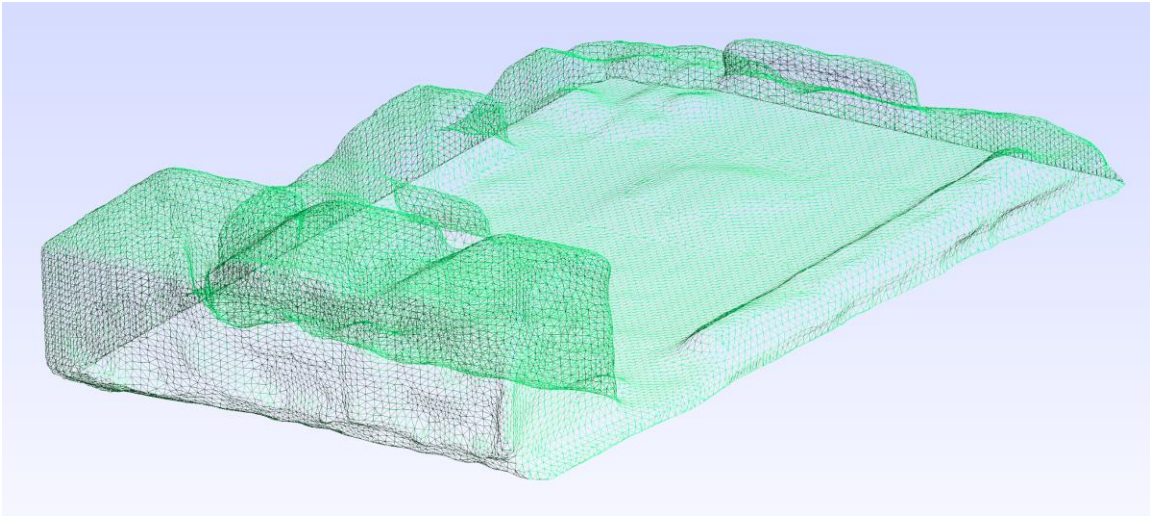

Figure 24: Mesh of Recap model of Raspberry Pi

In figure 25 below, an Agisoft generated model of the Raspberry Pi is shown. Along with the model is various blue rectangles circulating the object. These rectangles represent the camera positions of each individually taken photo. During this model, the capability to scan the bottom of an object was tested. Since the entire backdrop is whitewashed out, the software was

able to stitch together the photos and generate an underside of an object. This is very useful when modeling an object that is not flat on the bottom, as it will give more accurate resulting geometry.

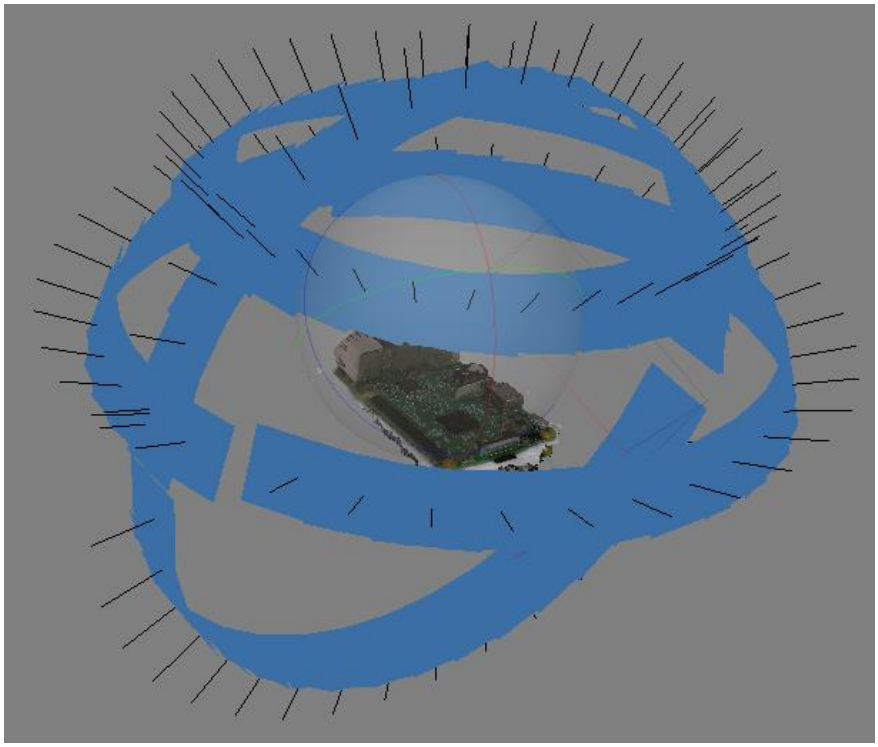

Figure 25: Agisoft model of Raspberry Pi

In figure 26 below, is the best model that was generated of the turtle. This turtle took many attempts both within Agisoft (figure 27) and AutoDesk Recap (figure 28). The first few attempts of the model generation resulted in a geometry with zero thickness. This is extremely undesirable as the model does not portray the target object at all and is absolutely not printable. Throughout many attempts of capturing image sets of the turtle, a model was finally generated. AutoDesk Recap again generated the smoothest model of this target object.

Figure 26: Recap model of plastic turtle

Figure 27: Agisoft model of plastic turtle

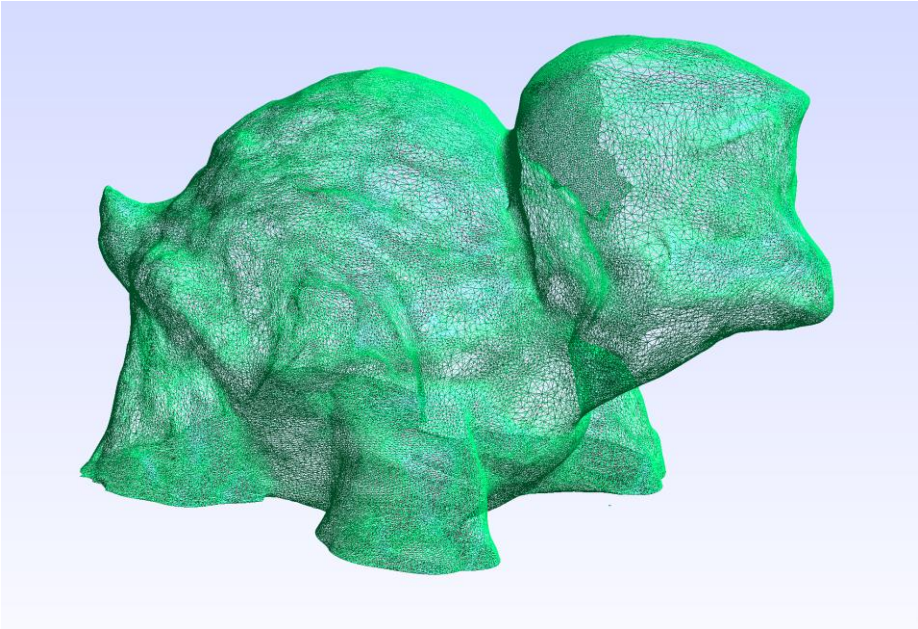

Figure 28: Mesh of Recap model of plastic turtle

In the figures above showing a green mesh (22, 24, and 28), the resolution of the final *.stl models can be easily observed. Each point is connected to neighboring points, creating a “triangle” mesh. The triangle size in the mesh is used to describe to resolution of the mesh, and the invariability of the triangles dimensions describe the quality of the mesh.

Using a binary file export from Agisoft containing a list of coordinates of each point in the point cloud, we were able to model the point cloud in SolidWorks as well. The idea was that in a CAD environment, a point cloud can mesh and surface in order to create a more workable model than just a *.stl file. We wrote a macro to import and write each point to a 3D sketch. However, each model contained tens of thousands of points in a point cloud. We made the assumption that every point is not necessary, as a dense point cloud captures more than general surface geometry. Using Microsoft Excel, we took every tenth point and wrote it to a file for the macro. The resulting point cloud is shown below in figure 29.

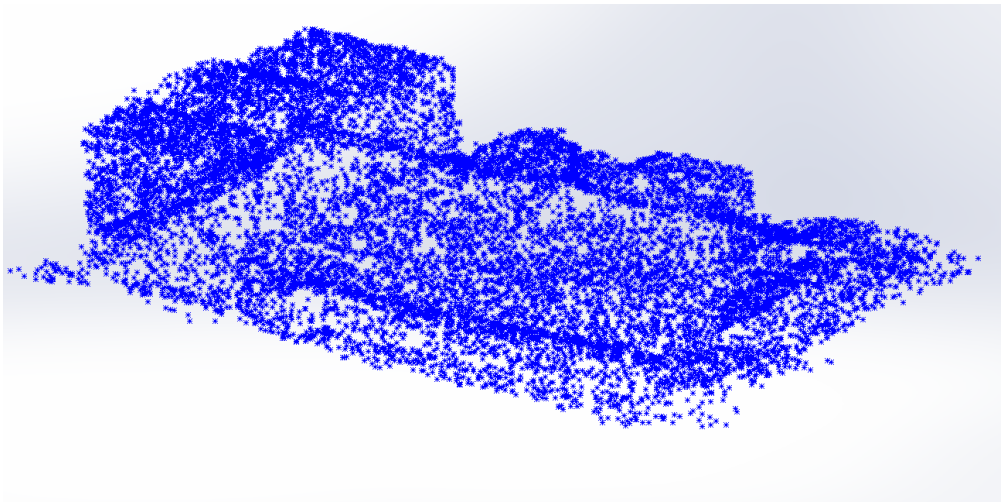

Figure 29: Point Cloud in Solidworks of Raspberry Pi

4.3 3D Printed Models

For minor post-processing of the models, we utilized Autodesk MeshMixer to edit the *.stl files. Simple surface smoothing and creating a flat bottom were among the few edits completed. These edits were done specifically to improve the 3D print itself.

The first 3D model that was taken to the printer was the model of the turtle. The printer used was a Prusa i3 MK3 printer, along with PLA filament. The most printable model of the turtle came from AutoDesk Recap Photo. Along with some post-processing touch up through MeshMixer, the turtle model was ready to be printed. After one failed print, the turtle was successfully printed out of PLA, shown below in figure 30.

Figure 30: 3D printed model #1

After 3D printing the turtle, the team moved forward with printing the model of the tape dispenser. This model was 3D printable without the use of MeshMixer. The only post processing that was needed for this print was to add a flat bottom to the model, which was done in Recap. Once this was done it was sent to the printer. The print of the tape dispenser came out well and only took one print, seen in figure 31 below.

Figure 31: 3D printed model #2

4.4 Future Work

Future work could include moving forward with a revolving camera idea for a stage design. When the stage itself is rotating, anchor points are assigned to the object being scanned. If the background is not whitewashed or there are shadows in the images taken, because the background isn't moving, the software will also attempt to use those points, thinking they are points of interest because they appear in every frame. For a basic user who could have a difficult time getting the optimal setup of a whitewashed background or light box, a revolving camera could prove to be a great option for standardized image collection. With this method, the object would appear in different orientations in every image as with a rotating platform, but the background scene would be constantly changing as well. The issue would then be modeling the object independent of the platform or background, rather than accidentally merging the background with the object.

Chapter 5 - Conclusions

Of the different software products we tested, Agisoft Photoscan Pro provides the most flexibility for image pairing, point cloud generation, and tiling/meshing. Although Agisoft is the more professional option for parameter settings, Autodesk Recap actually created the more 3D printable models. Results supported our object difficulty categories, as larger objects with more diverse geometry and colors tended to reconstruct more accurately.

Once the models were generated, two out of the three were successfully 3D printed. The Raspberry Pi model was not attempted to be printed due to the fact that it is not a very interesting model without a texture. Without a texture it is merely a rectangle with a few rectangles sitting on top of it. It was decided that it was not worth the time or materials to print this model. The successful printing of the models shows that the scanning stage as well as the process of image collection is justified. From being able to take an object, take pictures of it, and importing them into a software for processing, the team is able to generate a fully functional and printable 3D model of the object along with a wrapped texture. The models are also able to be brought into other 3D modeling software such as Solidworks to be further worked with. This may be useful for both tweaking the geometry of an existing object, or if the target object needs to incorporated into a larger assembly.

References

[1] Markus-Christian Amann, Thierry M. Bosch, Marc Lescure, Risto A. Myllylae, Marc Rioux, "Laser ranging: a critical review of unusual techniques for distance measurement," Opt. Eng. 40(1) (1 January 2001)

[2] Wildberger, N J. “Pythagoris, Euclid, Archimedes and a New Trigonometry.”*Http://Web.maths.unsw.edu.au/~Norman/*, web.maths.unsw.edu.au/~norman/papers/Pythagoras.pdf.

[3] Breuckmann, Bernd. “25 Years of High Definition 3D Scanning: History, State of the Art, Outlook.” The British Computer Society, doi:http://dx.doi.org/10.14236/ewic/eva2014.31.

[4] Exercises in Three Dimensions: About 3D, Tom Lincoln, 2011

[5] Blake, R., & Sekuler, R. (2006) Perception (5th ed.). New York, NY: McGraw-Hill.

[6] Song Zhang (2009) Recent progresses on real-time 3D shape measurement using digital fringe projection techniques, Optics and Lasers in Engineering, Volume 48, Issue 2, https://doi.org/10.1016/j.optlaseng.2009.03.008.

[7] Gorthi, et al. “Fringe Projection Techniques: Whither We Are?” Optics and Lasers in Engineering, Elsevier, 2010, infoscience.epfl.ch/record/140745.

[8] Apple. “IPhone 7 - Technical Specifications.” Apple, www.apple.com/iphone-7/specs/.

[9] R. Ghazali et al, "Evaluating the relationship between scanning resolution of laser scanner with the accuracy of the 3D model constructed," *2011 IEEE International Conference on Control System, Computing and Engineering*, Penang, 2011, doi: 10.1109/ICCSCE.2011.6190595

[10] Guideline for Noting Digital Camera Specifications in Catalogs. "The term 'Resolution' shall not be used for the number of recorded pixels"

[11] “The Advantages Of Increasing The RAM Of A Computer.” Pervasive Technology, www.pervasive-postgres.com/increasing-ram-computer/.

[12] Schindler, Konrad. "Mathematical Foundations of Photogrammetry." *Handbook of Geomathematics* (2014)

[13] Radke, Richard J. Computer Vision for Visual Effects. Cambridge University Press, 2013.

[14] Agisoft. http://www.agisoft.com/pdf/photoscan-pro_1_2_en.pdf

[15] Burkard, J. “Material Definitions for OBJ Files.” People.sc.fsu.edu, people.sc.fsu.edu/~jburkardt/data/obj/vp.mtl.

[16] All3DP. “STL File Format for 3D Printing - Simply Explained.” All3DP, 16 Oct. 2017, all3dp.com/what-is-stl-file-format-extension-3d-printing/.

[17] AutoDesk. “Learn and Explore.” Autodesk Support & Learning, knowledge.autodesk.com/support/recap/learn-explore?sort=score.

[18] SmartMobileVision. “Scann3D.” Scann3D: SmartMobileVision, scann3d.smartmobilevision.com/./

[19] 3dprintingfromscratch.com. “Types of 3D Printers or 3D Printing Technologies Overview.” 3D Printing from Scratch, 2 Feb. 2016, 3dprintingfromscratch.com/common/types-of-3d-printers-or-3d-printing-technologies-overview/.

[20] “History of 3D Printing Timeline: Who Invented 3D Printing.” 3D Insider, 2 Feb. 2018, 3dinsider.com/3d-printing-history/.

[21] Deckard, C., "Method and apparatus for producing parts by selective sintering", U.S. Patent 4,863,538, filed October 17, 1986, published September 5, 1989.

[22] "DMLS | Direct Metal Laser Sintering | What Is DMLS?". www.atlanticprecision.com. Retrieved 2017-04-10.

[23] “The Complete Guide to Electron Beam Melting 3D Printing (EBM).” 3Dnatives, 22 Dec. 2017, www.3dnatives.com/en/electron-beam-melting100420174/.

[24] Palermo, Elizabeth. “What Is Laminated Object Manufacturing?” LiveScience, Purch, 9 Oct. 2013, www.livescience.com/40310-laminated-object-manufacturing.html.

[25] Wellington, Matthew. “How 3D Printing Has Evolved in the Last 10 Years.” 3D Print HQ, 20 Mar. 2013, 3dprinthq.com/how-3d-printing-has-evolved-in-the-last-10-years/.

[26] T-Mobile. “Tech Specs: Samsung Galaxy S 5.” T-Mobile Support, support.t-mobile.com/docs/DOC-28502.

[27] Rosebrock, Adrian. “Accessing the Raspberry Pi Camera with OpenCV and Python.”PyImageSearch, 14 June 2015, www.pyimagesearch.com/2015/03/30/accessing-the-raspberry-pi-camera-with-opencv-and-python/.

[28] Rongala, Arvind. “Benefits of C / C++ over Other Programming Languages.” Invensis Blog, 6 Apr. 2018, www.invensis.net/blog/it/benefits-of-c-c-plus-plus-over-other-programming-languages/.

[29] Jones, Dave. “Picamera Package.” Picamera - Picamera 1.13 Documentation, picamera.readthedocs.io/en/release-1.13/.

[30] “Raspberry Pi 3 Is out Now! Specs, Benchmarks & More.” The MagPi Magazine, 14 Mar. 2018, www.raspberrypi.org/magpi/raspberry-pi-3-specs-benchmarks/.

[31] Adafruit Industries. “Raspberry Pi NoIR Camera Board v2 - 8 Megapixels.” Adafruit Industries Blog RSS, www.adafruit.com/product/3100.

[32] Lazy Susan - retrieved from https://www.kohls.com/product/prd-1913909/lipper-acacia-18-in-lazy-susan

[33] Thingiverse.com. “The $30 3D Scanner V7 Updates by Daveyclk.” By Daveyclk - Thingiverse, www.thingiverse.com/thing:1762299.

[34] Thingiverse.com. “Rotating Platform for 3D Scanning - with Scaling and Alignment Features! by Tbeerkens.” By Tbeerkens - Thingiverse, www.thingiverse.com/thing:2351574.

[35] EyeCue Vision Technologies LTD (2018) [Qlong -3D Scanning and AR Solution] retrieved from https://itunes.apple.com/us/app/reuters/id602660809?mt=8.

Appendix

Appendix A: Raspberry Pi 3 Model B Technical Specs [30]

SoC: Broadcom BCM2837
CPU: 4× ARM Cortex-A53, 1.2GHz
GPU: Broadcom VideoCore IV
RAM: 1GB LPDDR2 (900 MHz)
Networking: 10/100 Ethernet, 2.4GHz 802.11n wireless
Bluetooth: Bluetooth 4.1 Classic, Bluetooth Low Energy
Storage: microSD
GPIO: 40-pin header, populated
Ports: HDMI, 3.5mm analogue audio-video jack, 4× USB 2.0, Ethernet, Camera Serial Interface (CSI), Display Serial Interface (DSI)

Appendix B: Raspberry Pi NoIR Camera v2 Technical Specs [31]

Sensor type: Sony IMX219PQ[TM] Color CMOS 8-megapixel

Sensor size: 3.674 x 2.760 mm (1/4" format)

Pixel Count: 3280 x 2464 (active pixels) 3296 x 2512 (total pixels)

Pixel Size: 1.12 x 1.12 um

Lens: f=3.04 mm, f/2.0

Angle of View: 62.2 x 48.8 degrees

Full-frame SLR lens equivalent: 29 mm

Video Modes:

1 - 1080P30 cropped (680 pixels off left/right, 692 pixels off top/bottom), up to 30fps

2 - 3240x2464 Full 4:3, up to 15fps

3 - 3240x2464 Full 4:3, up to 15fps (identical to 2)

4 - 1640x1232 binned 4:3, up 40fps

5 - 1640x922 2x2 binned 16:9 (310 px crop T/B before binning), up to 40fps

6 - 720P bin+crop (360 px L/R, 512 px T/B before binning), 40..90fps (OC: 120fps)

7 - VGA bin+crop (1000 px L/R, 752 px T/B before binning), 40..90fps (OC: 120fps)

Board size: 25 x 23.86 x 9mm

Mounting Holes: 4x D=2.20 mm on 12.5 x 21.0 mm centers

Appendix C: Python Code from “Top Camera”

```
from picamera.array import PiRGBArray
from picamera import PiCamera
import os
import time
import cv2

pics = int(float(raw_input('Number of images: ')))
int = int(float(raw_input('Length of Interval (seconds): ')))*1000
length = (pics-1)*int

time.sleep(2)

os.system("raspistill -w 3280 -h 2464 -q 100 -t " + str(length) + " -tl " + str(int) + " -rot 270 -o
/home/pi/opencv-3.3.0/frames/high/image1%04d.jpg")
```

Appendix D: Python Code from “Bottom Camera”

```
from picamera.array import PiRGBArray
from picamera import PiCamera
import os
import time
import cv2

pics = int(float(raw_input('Number of images: ')))
int = int(float(raw_input('Length of Interval (seconds): ')))*1000
length = (pics-1)*int

time.sleep(2)

os.system("raspistill -w 3280 -h 2464 -q 100 -t " + str(length) + " -tl " + str(int) + " -rot 270 -o
/home/pi/opencv-3.3.0/frames/low/image%04d.jpg")
```

Appendix E: Python Code for viewing images at camera positions before official acquisition

```
from picamera.array import PiRGBArray
from picamera import PiCamera
import time
import cv2
import os
import numpy

try:
    if not os.path.exists('frames'):
        os.makedirs('frames')
except OSError:
    print ('Error: Creating directory of data')

camera = PiCamera()
camera.resolution = (640, 480)
camera.framerate = 32
rawCapture = PiRGBArray(camera, size=(640, 480))
currentFrame = 0

time.sleep(0.1)

for frame in camera.capture_continuous(rawCapture, format="bgr", use_video_port=True):
    image = frame.array
    cv2.imshow("Frame", image)
    key = cv2.waitKey(1) & 0xFF
    rawCapture.truncate(0)
    if key == ord("w"):
        name = './frames/frame' + str(currentFrame) + '.jpg'
        cv2.imwrite(name, image)
        currentFrame +=1
    if key == ord("q"):
        break
```