SR3D:

Short Range 3D Scanner

Nirav Kotadia, Franco Zavagnini, Naoufal Rihani, and Gabriel Akenten.

Dept. of Electrical Engineering and Computer Science, University of Central Florida, Orlando, Florida, 32816-2450

***Abstract* — In an effort to combine multiple elements of both Electrical and Computer engineering we developed a short range scanner, SR3D is a tool to scan medium size object in a short period of time, by providing points to client that represent a object in 3 dimension space. It is composed of a line laser, camera, rotating motor, and a turntable. All these components will be enclosed in a convenient manner for the user. The SR3D utilizes a Linux operating system in order to manage the flow of information from the camera and able to show live feed of data being process. Program will also provide option to output file in formats that can be edited in many different programs. SR3D will be powered by a single power supply which is connected to 120 volt outlet.**

Index Terms — 3D, Scanner, Laser, Motor, Turntable

I. INTRODUCTION

Nowadays, the use of three-dimensional (3D) scanners has become a widely used tool**.** These innovative devices are being introduced or made to overcome existing problems that seem either impossible or extremely harder to solve before. Three Dimensional scanners can be used in different disciplines, such as reverse engineering, they can also be used to create detailed automotive parts, virtual reality, visual effects, animation , sculpture, mold making, graphic designing, video games, 3D printing, robotic mechanics, medicine, forensics, dentistry, movies, museum, toy, architecture and manufacturing.

But most of scanner available in market cost from a thousand to tens of thousands. So we want build inexpensive scanner that does the same job but cost way less. This project perfect fit for our group, since our team consists of half Computer Engineers and half Electrical Engineers. And we wanted to combine all of our experience and knowledge in these two related fields while pursuing new topics of interest for each team member. Since cheapest scanner we able to find on market was around $500 dollar. We build our scanner that cost less than $500. We able to design a low-cost system which can be implement hardware and software system for digitizing the shape of a medium sized object. We also added feature to show live scanning process and output file that can be open in many other programs to further use.

II. OVERVIEW OF THE SR3D

Our goal for this project was to design a low-cost scanning system and implement a laser to generate a 3D digital model of an object. Our project presents a hardware and software system for digitizing the shape of our chosen object. By the end of our design, we had completed the scanning of a 3D object in a given amount of time and accurately displayed the generated 3D model in MeshLab and final product should be comparatively cheaper than commercially available 3D scanners. Data acquisition from the scanning point was a critical aspect of this design since this enables the acquisition of the point cloud needed to be represented digitally. The scanner consists of a motor driver, turntable, laser, camera, and control logic unit which we will refer later in this paper.

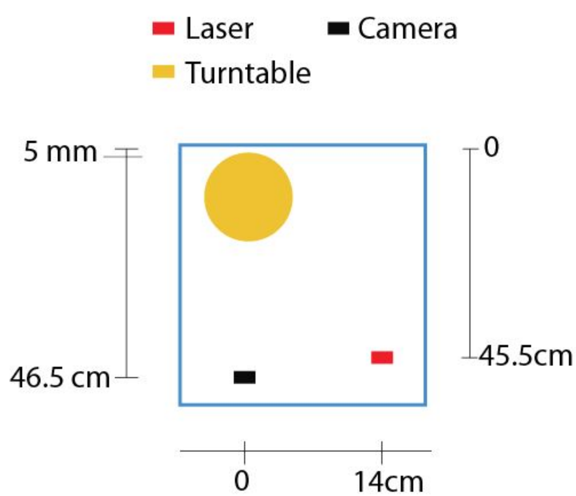


Fig. 1. Top view of SR3D

During operation, the turntable will provide a rotational movement, to allow the camera to capture and generate an image data of the object from every angle. The results from the system will be carefully monitored and displayed so that an end user could be able to tell if the device is efficiently performing to its desired specifications. These outcome images will be stored on our computer which then will process those images and finally create a 3D model in MeshLab.

The group has gained knowledge and experience from this project which will be beneficial since 3D applications are very new on the market and future employers might take it into consideration. Other relevant aspects of the design will be broadening our scope of knowledge along the way. Our goal was to create a 3D Scanner that would not exceed 5 kg in weight, also, it should be able to scan a medium sized object with a detailed resolution of 5 mm. The scanning speed will be approximately 10 min., and the maximum weight that we will be able to put in the turntable is 1 kg. Finally, the volume of the object should not exceed 5 in. in diameter and a maximum height of 6 in.

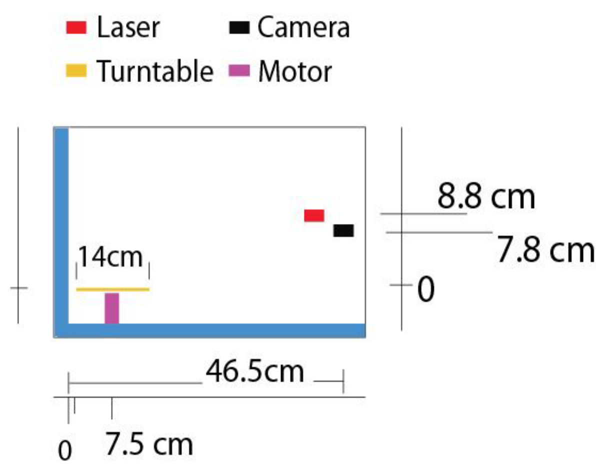


Fig. 2. Side view of SR3D

Figure 3 diagram is the overall block structure of the SR3D that we have developed. It is composed of the programming unit which is Linux base software that we develop. Which read the data from the webcam, and gives control signal to the processing unit which is microcontroller where motor driver and laser will be connected. Microcontroller can directly control laser light but to control motor, microcontroller need to send signals to motor driver to control the motor. We build our own power supply which will get power by normal plug outlet from wall. Power supply unit provides power to laser, motor and motor driver.

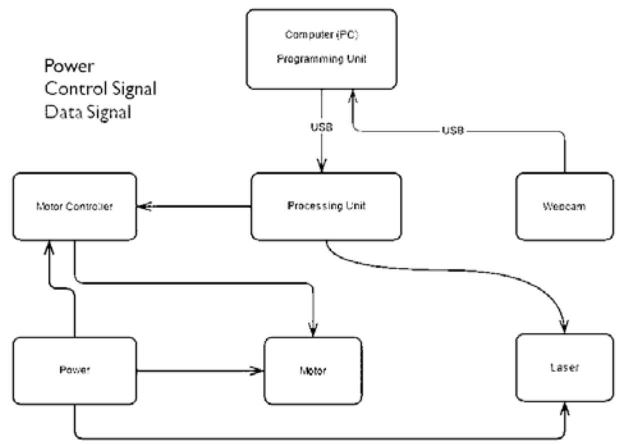


Fig. 3. SR3D System Block Diagram

III. SCANNING METHOD

When we talk about actually implementing the 3D modeled object in the data representation software, we are planning to get the series of picture from webcam and then pass it through the software. Some major issues that we might encounter along the way are how to completely model and render the scanned object. We are planning to write some algorithms for each part of the thesis. We want to implement some major mathematical and modeling algorithm starting from controlling the servo motor, calculating angle of camera and laser, and finally, the most important part that is actually modeling and rendering the scanned object. The controller part is the second to last most important part of this project. We will dedicate a whole section talking about this issue.

After obtaining the final input file from the controller via USB/FireWire cable, we will begin to pass the input file to the algorithm to finally display it in the data representation software that we have been discussing above. This and more important aspects of our project will be put into place later on when we start building and coding our project. For now, we have a good background programming wise on how this 3D scanner is going to work computationally and electronically.

For our SR3D we have chosen to implement the scan using laser triangulation since we already know the position of the hardware (camera, laser and platform) we can simplify the model without changing the problem (really small precision loss). Since we are doing math in ideal situation we are assuming some stuff like laser line is perfectly vertical to object in center. And the camera is pointing perfectly straight and perpendicular to the background. It is not always possible to achieve ideal condition but error its keep to the minimum since we are fixing position of the camera and the laser we know the measurement. Since the laser line is hitting the known background position we know original position of the laser and how far it's move once we place the object on rotating table.

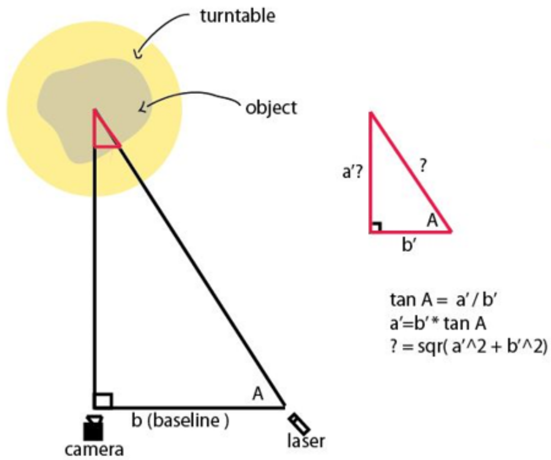


Fig. 4 Laser Triangulation

By fixing the camera and laser we know the angle where two lines meet. We also know the base line we can just measure it with ruler. And since we also fixing the position of the rotating table we also know what is the distance between camera and center of the table. One requirement for user is that object has to be place right at center of the object. After putting object on turn table, the positions of the reflected points on the object surface transformed from the image coordinate system to distance. Then we well do calculation one by one, this simplify the problem 3D problem to 2D line intersection problem. From figure above if we look up red right triangle which is form by center of object to current surface of the object. Once thing about know for that right triangle is angle A. But amount of information we know is not enough to calculate radius of current surface. So using camera we going figure out what is the value of b'. In Figure 4.1 we can see where laser is hitting object surface and knowing the turntable is at center of the camera / object we can measure how far laser is hitting the object by doing that problem simplify to geometry problem. So first we find we do "tan A = a'/b' " then solve for a'. Then using Pythagoras theorem we solve for hypotenuse. That will give us how far should we place that point away from center of the object.

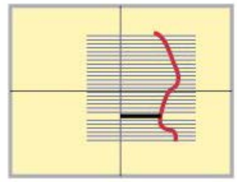


Fig. 4.1 Laser Triangulation

IV. SUBSYSTEMS

A. Processing Unit

We have chosen the Atmel AT-Mega 328 P because it is versatile and easy to program for our requirements, and also it works really well for 3D scanning purposes. Also, it is an 8-bit Microcontroller and has plenty I/O pins, specifically 20 which more than we need, so it can allow us to add more of features and options while controlling the scanning unit.

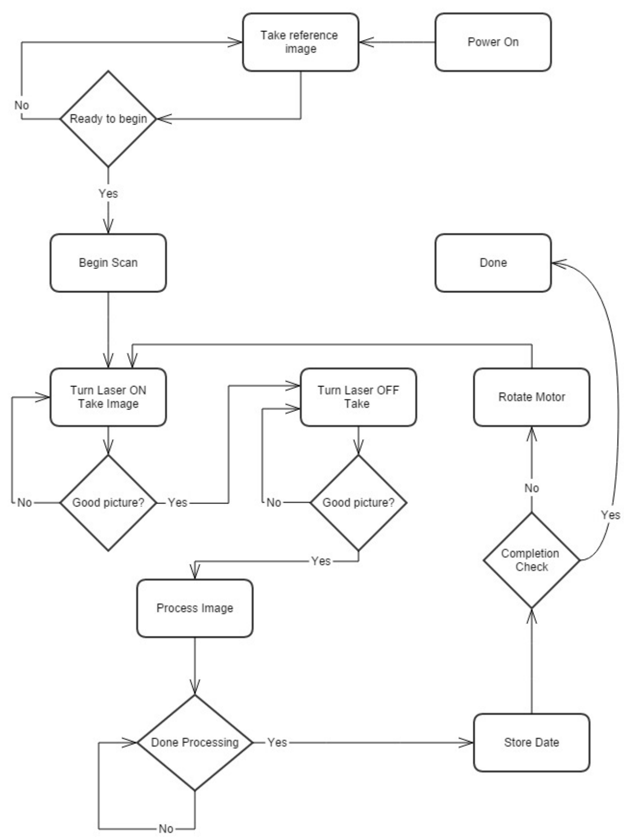


Fig. 5 System Flow Diagram

B. Laser and Camera Module

The Laser and Camera module produces the data which is to be processed. This module is implemented with a 25 mW 650nm red line laser. The vertical line will be about 25cm in length; it will then be projected at a blank background. When an object is placed on the turn table 0.5 millimeter away, the vertical laser line will be appear offset from the reference taken from the background.

Fig. 6 Laser offset base on distance and angle

This image will then be captured by a sensor which will then relay the data to the processing module. We used Arduino UNO mainly for testing purposes; mainly it would serve as transferring image data from camera to computer via USB cable quick as it can. The more samples we choose, the better the resolution of object scan can be and the less chance of leaving "holes" in middle of point cloud beside the bottom and top. But image file is always big so this will be our bottleneck for the project.

C. Motor Control Unit

The motor control unit will which brings the best performance and efficiency for our SR3D since we can adjust the velocity and the stepping of the turntable to acquire the desired output by simply programming it. Also, it is of great help since this is a very important part of our project. The motor driver will be in charge of controlling the motor which we will discuss later on in this paper.

D. Power Supply Unit

The power supply for the entire system will be powered from the lab's main power supply before being split up to the rest of the components. After choosing the recommended transformer of desirable power ratings, the power distribution analysis to the entire circuit will be designed making the needed voltage is met by each component of the circuit. This circuit to be designed will be simply made up of the various voltage regulator to be chosen for the project. The main relay will be a designed circuit which will begin with a 12 V regulated output voltage.

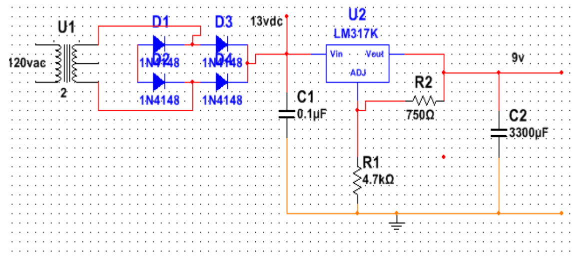


Fig. 7 Power Supply Circuit Diagram

E. Programming Unit

When talking about the programming unit in our thesis we will be dealing with basically every aspect of getting the data input file from the laser/camera to the data representation software. Also, we have talked about a lot of aspects regarding previous works done in this matter of 3D scanning algorithms. In the following block diagram we will discuss furthermore each and every part of the process of getting the image from the controller unit and passing it through the software that we will be using. Moreover, we are still looking forward to test many aspects of this important aspect of our SR3D.

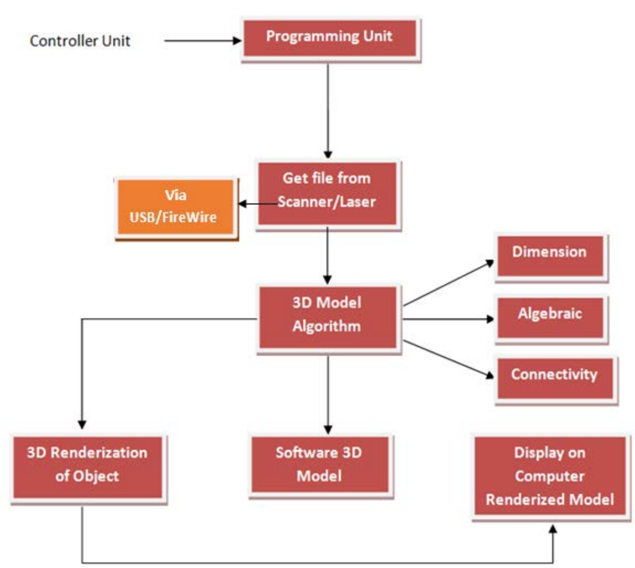


Fig. 8 Programming Unit Block Diagram

The idea is to basically get input file or point cloud from the controller unit and pass it Programming Unit will be in charge of getting the images and process them computationally as well as making sure that the object is of a decent size, shape, dimension, etc. We are planning to use Java as the programming language since there exist notable work on 3D Scanning Algorithm written in that language where we can take reference of to make our project simpler. However we plan to add special features like add color to the renderized objects displayed on software.

The 3D Model Algorithm builds upon earlier work in model synthesis. The user provides an example model as the input. The example model is a set of polygons that form closed polyhedral objects. Model synthesis generates a new model that resembles the scanned model gotten from the scanning process. Some factors in consideration or constraints that we have to deal with are applying the algorithm based on dimension of object since it will allow the user to model the object in a realistic dimension. Another constraint is the algebraic part where the user will have to calculate algebraically the size, shape, and height. Ultimately, is the connectivity constraint where user will have to pinpoint the object where holes occur in modeling so that it does not exist incomplete points in the renderized object.

In the 3D Software model we are planning to use MeshLab. Finally, after the scanned object has been synthesize and renderized, we are planning to display the smooth 3D modeled object in a computer screen via USB/FireWire Cable from Camera/Laser/Scanner to the CPU.

V. 3D OBJECT MODELING

When talking about the 3D Object modeling part of this project we want to recreate or model the three dimensional aspect of the object as accurate as possible. That is why mathematical as well as geometrical aspects should be considered in this part. The three dimensional space can be represented as X, Y, and Z axis. In order to obtain the data from the microcontroller we first need to inter communicate each device accordingly through a series of commands and algorithms. In this project we will utilize AT-Mega 328 and Arduino to communicate between devices.

Fig. 9. Modeling Algorithm

The information provided by these microcontrollers will allow us to synthesize the horizontal and vertical depth portions of the desired object. Also, the data files will be saved in memory and it will be not erased until we turn off the system. That is why we are planning to retain this data files and hopefully save them in the computer so that no data or point clouds are lost for future use. We will also obtain a basic 2D sample which will also be saved in local memory. The process of scanning will continue until we reach a whole 360 degrees around the object for perfect reconstruction.

Finally we would like to minimize the time scanning as much as possible so that this process does not become tedious and increments performance of the 3D scanners. We are planning to scan the object in less than 10 minutes. However, we still have to experiment with various cases. Depending on the resolution of the scanned object, more time to might be required to actually obtain the best possible rendered model of the object. Ultimately, more details might arise as we go through the actual construction of the 3D scanner and its sequence of steps to completely get the best representation of the scanned object.

VI. POINT CLOUD

This is probably the most important 3D data representation out there and the one we will be using and playing around with throughout this whole thesis. Point clouds are really important to represent modeled object since they give a rough sketch of how the objects will look like digitally as well as represent the characteristics of the scanned 3D model. It is a set of data points in the coordinate system. Specifically in the three dimensional coordinate system, these points are normally defined as {X,Y,Z} coordinates, and they are used to represent the external characteristics of the modeled object. These point cloud are created by a regular 3D scanner. The 3D scanners are really useful in creating these points to model 3D objects.

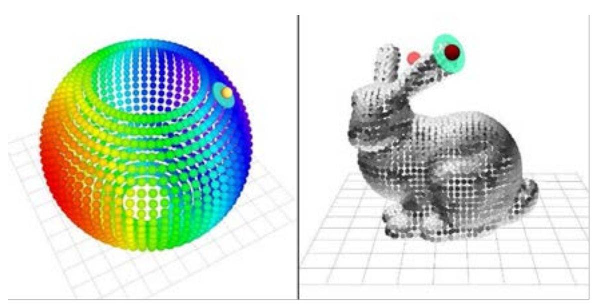


Fig. 10. Point Cloud Examples

Normally, the scanners get the input file or point clouds and translate these to a data representation software which gets the output file and models the object. Here are some examples of how point clouds look like after they have been scanned and rendered in the data representation software. We will be dealing with many point cloud in this thesis in order to get a better understanding on how these are really helpful in representing and visualizing the desired scanned 3D object.

VII. RENDERIZATION

Rendering of an object is just the process of generating the image of a modeled object. In our case we will be rendering the scanned object ideally to best represent the objects characteristics as well as real as possible from actual object. This includes all geometrical shapes and dimensions of the real object in the data representation software that we will be using. The final rendered modeled will contain many characteristics such as different point of views by means of 360 degrees view, texture, shades, illumination, etc. Many programs are available to render scenes which is the term given to actually rendering, and it is used in video games and animated movies development.

In our project we will be using a data representation software called MeshLab which is free to use and easy to manipulate. There are also many techniques for rendering, however, we will focus on basic strategies since the most important part of this project is actually communicating the devices and obtaining the point clouds for further use. Later on, we might actually implement these so called rendering techniques to better view the scanned object as real as possible digitally. Also, we might need a high speed computer with a high quality video card in order to maximize performance when rendering the scene or the scanned modeled object. In the following picture we can demonstrate how real a scene can be when rendered.

VIII. HARDWARE

A. Camera

We will be using the Logitech C270, it has USB 2.0 connectivity, and a resolution capture of 720p. We did not choose a higher end camera since we do not want unnecessary processing time. This one is just fine and it minimizes the scanning time by almost 20% compared with others. Also, it is very cheap, and we are working within a specific budget. We tried the Logitech C920 camera which resolution is 1080p and we encountered that scanning time is increased by almost 30% which translates in more time scanning. However, the quality increases considerably, but that is not the kind of quality that we are aiming for. The scanning time that we have so far is just fine as well as the quality.

Fig 11. Logitech C270

B. Laser

Our SR3D will use a line laser called Linien laser (103A2B). We choose this laser since it is line laser which means that we can scan the object vertically without having to move up and down and laser. It scans a complete line along the vertical axis. In the table below we can observe the characteristics of this laser.

SPECIFICATIONS		
INPUT VOLTAGE	3.5 - 4.5	v dc
OPERATING CURRENT	< 25	mA
WAVELENGTH	645-655	nm
OPTICAL POWER	< 5	mW

Table 1. Laser Specifications

Fig. 12. Linien Laser

C. Microcontroller

For the microcontroller part of our project we will be using ATMel ATMega328p which is an 8 bit microcontroller and has 20 I/O pins. We selected this controller since it is cheap, easy to program, and provides the best performance to our 3D scanner. The main purpose of using this microcontroller is to achieve maximum performance as well as efficiency.

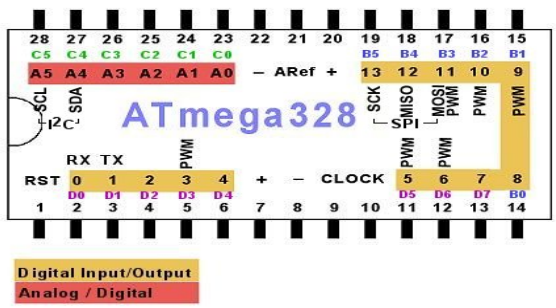


Fig. 13 ATMega328p Diagram

D. Motor

We will be using the stepper motor NEMA 17HS15-0404S which will provide the perfect stepping for our 3D modeling algorithm. In the following table we can see some of the specifications of this stepper motor.

Manufacturer Part Number	17HS15-0404S
Motor Type	Bipolar Stepper
Step Angle	1.8°
Holding Torque	40Ncm(56.6oz.in)
Rated Current/phase	0.4A
Phase Resistance	30.0ohms
Recommended Voltage	12-24V
Inductance	58mH±20%(1KHz)

Table 2. Stepper Motor Specifications.

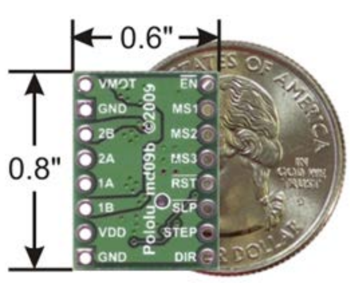


Fig 14. Allegro A4988 Stepper Motor Driver.

The motor driver that we will be using is the Allegro A4988 Stepper Motor Driver. It has a minimum operating voltage of 8V and a maximum operating voltage of 35V. The minimum and maximum logic voltage is 3V and 5.5V respectively. It manages the micro step resolution at different magnitudes like full, ½, and ¼.

E. Power Supply

We will be developing our own power supply. Some of the specification is that it must provide 9V to all of our components as well as 1A of current to all of our components.

IX. SOFTWARE

We will be using Ubuntu as our operating system to implement our SR3D application. Also, the program was written in C++, since almost all the libraries and components required so. The compiler that we used throughout this whole project was Qt Creator for GUI purposes.

- *OpenCV*

OpenCV is an open source data representation software mainly focused on computer vision. The functions in this section use a pinhole camera/laser model. In this model, a scene view is formed by projecting 3D points into the image plane using distinct algorithmic techniques. Real lenses usually have some distortion, mostly radial distortion and slight tangential distortion. OpenCV extracts 3D models of objects, produce 3D point clouds from stereoscopic cameras, stitch images together to produce a high resolution image of an entire scene among other features. OpenCV is mostly written in C++.

- *Point Cloud Library*

This open source library is the basis of our software part of the project since it will be the one to actually implement the point cloud acquisition as well as the large scale, standalone, and open project for 2D and 3D image and point cloud processing. Point Cloud offers many algorithms including filtering, feature estimation, reconstruction, and model fitting, among others.

- *MeshLab*

This software is an open source, open project application which allows the rendering of the output file or point cloud to look smoother. It also processes and edits 3D triangular meshes for correction and better viewing of the scanned object. This software application is the best open source out there for rendering, modeling, cleaning, remeshing, coloring, and inspection of 3D meshes.

X. CONCLUSION

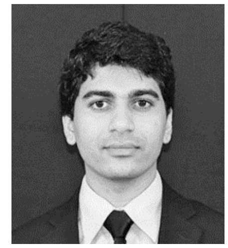
3D Scanners have been around for quite a few years and they still continue to be one of the most widely used devices for subsequent 3D printing. We will not get into details about 3D printing. However, it is something that we might consider in the future for future projects. Since we started this research, we have learned a lot about this technology which makes 3D modeling and rendering of object relatively easy and we will continue to strive for perfection as we go along this project. There are many 3D Scanning Devices out there in the market. The whole purpose of this project is to make one that is affordable, effective, with high performance, cheap, portable, not so heavy, and that it can get the job done meaning to scan and object in three dimensions, model it, and render it in data representation software.

We have dealt and informed ourselves with several previous works in this matter and we have reached to the fact that to make an affordable 3D Scanning Device we have to constraint ourselves into making su re that the components that we buy will have all the reference material as well as the desired specifications to actually implement our design. We must not lose time dealing with wrong components since we have not that much time to accomplish our goal which is building our 3D Scanning device, coding our microcontrollers, testing, debugging, assembling, and presenting it. Basically, every 3D Scanner out there is expensive, big, not so portable, and they require better performance.

XI. ACKNOWLEDGEMENT

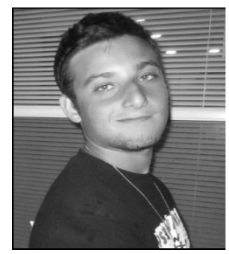
Group 9 of the Summer 2014 – Fall 2014 Senior Design class would like to acknowledge and thanks the Department of Electrical Engineering and Computer Science at the University of Central Florida, and Dr. Richie for the help and advice.

XII. TEAM



Nirav Kotadia is a graduating Computer Engineer from University of Central Florida. His areas of interests include low-level programming, FPGA, and microprocessors.

Franco Zavagnini will be graduating this semester as a Computer Engineer from University of Central Florida. His dream is to work in a big company such as Apple or Microsoft. His passions are computers and sports.



Naoufal Rihani is a graduating Electrical Engineer from the University of Central Florida. He has a high interest in the field communications systems but is also passionate about microcontrollers and computer architecture.

Gabriel Akenten – He will be graduating this semester as an Electrical Engineer from University of Central Florida. His specialty is power and electrical systems.

XII. REFERENCES

[1] Flexible calibration Method for 3D Laser Scanner System: Yang Zhangdong , March 2011.

[2] Design and calibration of a computer vision and structured light 3D imaging system: Tom Botterill, Steven Mills, July 2014.

[3] Laser Triangulation for shape acquisition in a 3D scanner: David Acosta, Olmer Garcia, May 2013

[4] Qt Project <qt-project.org/doc>