Automated Photogrammetry Station

Alhusain Ali Al Badi, Ryan Dimmig, Isaac Liljeros, Nelson Vargas

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

***Abstract* — The Automated Photogrammetry Station is designed to increase the ease and accessibility to photogrammetry by streamlining the picture capturing process. Photogrammetry is the process of generating a 3D model from pictures of an object taken from various angles. We aim to make the picture taking part of the process easier by asking the user for a few settings with a simple user interface, then taking the pictures with the help of motors and a microcomputer to control the camera. The station utilizes three motors, a camera, a hall-effect sensor, a touchscreen, various mechanical parts, and two microcontrollers, one of which is included on a custom Printed Circuit Board (PCB).**

Index Terms — Belts, DC Motors, Displays, Microcontrollers, Photography, Structure from Motion, Universal Serial Bus.

I. Introduction

The Automated Photogrammetry Station is a device designed to streamline the photogrammetry process by simplifying picture capturing in a reliable way to save the user time and effort. Photogrammetry is the process of using pictures of a three-dimensional object from various angles to reconstruct the object as a 3D model. This is done by processing the images in reconstruction software that utilizes Structure-from-Motion (SfM) algorithms to generate a model [1]. Our device automates the process of image capturing by allowing the user to place an object on a turntable, select how many pictures they want taken from different angles, and then run the device which rotates the object and angles the camera to take pictures automatically. It then sends the pictures to the user's computer to begin the reconstruction process using the third party software Meshroom.

Our station is designed to be cost-effective, user friendly, and scalable. The station is comprised of readily available components and 3D printed parts to reduce manufacturing costs, and is designed to utilize a DSLR camera that can be provided by the user. The user experience involves a touchscreen that requests the IP address of the Linux workstation and gathers settings such as the number of angles from which the user wants pictures taken, the history of which is saved for easy selection. The design is also scalable by swapping out parts such as the turntable or stand railing to allow for larger and heavier objects to be captured.

II. Project Components

The photogrammetry station consists of various mechanical and electrical hardware components, most notably the railing, motors, camera, touchscreen, Raspberry Pi microcomputer, and a custom printed circuit board (PCB) with a microcontroller. This section will give an overview of the various components used in our project.

A. Mechanical Structure

The mechanical structure allows for three degrees of motion, including moving the camera vertically, tilting the camera, and rotating the object on the turntable. The vertical motion is achieved by a timing-belt driven linear actuator made of an aluminum rail, which uses a motor connected to a belt to lift the camera along the railing. The tilting is achieved via a carriage attached to the belt that holds the camera and houses another motor that controls the camera's angle. This, along with the base of the stand, are 3D printed. The rotational motion is achieved with a turntable design consisting of two aluminum rail profiles, a lazy Susan bearing, and 3D printed base.

B. Camera

We used a Canon T7i DSLR camera for our image capturing due to its high image quality and excellent integration ability with software. The camera is mounted to the carriage that is lifted along the linear actuator, and is connected to another motor that allows it to tilt to face the object from a variety of angles. We utilized the gPhoto2 software to connect to the camera and control picture taking from software instead of hardware. The higher quality of the pictures allows for 3D models of higher detail and quality, which was an important goal for the project. Other options we considered were using a phone camera or the Pi Cam 3, but the phone camera did not offer the best image quality and would be difficult to control through software without developing an app, and the Pi Cam 3 had lower image quality and offered less modularity, as the user could swap out this specific brand of camera for another that they have.

C. Microcontroller Units

Our project utilizes two microcontrollers, each of which have important tasks for which they are better suited. We used an ATmega2560 microcontroller unit to control the motors and touchscreen and a Raspberry Pi 3 to control the camera and transmission of pictures to the workstation. The ATmega2560 was a part of our PCB, while we used the Raspberry Pi development board. The ATmega2560 provided a better programming experience for the touchscreen and motors and could easily handle the coordination of these components. This offered an advantage during development compared to the MSP430, which we also considered. However, the microcontroller was unable to interface with the camera to the extent which we required, and so a second microcomputer that could run an operating system was necessary. The two communicate through a wired USB connection, which allows the ATmega2560 to control motors to get the camera in position, then instruct the Pi to take pictures with the camera and send them to the workstation that runs on a separate PC.

D. Motors and Drivers

We are utilizing three motors in our project, one on the stand to provide vertical motion for the camera, one on the carriage to tilt the camera to aim at the object from different angles, and one on the turntable to rotate the object to get pictures from more even angles. We used StepperOnline NEMA-17 stepper motors in our design, as they provided the required precision for controlling the motion of the camera and object, and they also provided enough torque to lift the camera and hold it at the necessary angles. We also considered the NEMA-23 stepper motors, and although they provide higher torque, this was not necessary for our project, and they had a higher weight and cost. We selected the A4988 motor driver design for the motors, which we integrated onto our PCB. This driver supports micro-stepping that allows us to have more control over the small angles at which the camera rotates.

E. Auto-Homing Sensors

In order to calibrate the vertical position of the camera, we implemented two MXRS KY-003 hall-effect sensors that detect the position of the camera at the top and bottom of the stand, and this information is used to translate the number of steps of the motor to the distance that the camera travels. We chose these sensors because of their small size and precise detection.

F. Display Module

Our user interface uses a touchscreen to gather user input and display messages. The input needed is the IP address of the workstation for connecting as well as the height of the object they're taking pictures of and how many angles from which they want pictures. We chose the HXD8357D 3.5" TFT touchscreen because of its ease of use due to the graphics and touch libraries, as well its size, providing enough space for the UI to display important information and buttons for collecting user input. The touchscreen was chosen over a basic LCD display with accompanying buttons due to the ease of programming and improved user experience.

G. Power System

Our device is designed to plug into a wall outlet, and thus utilizes AC power. We chose to use an ALITOVE AC to DC 12V Power Supply to convert to DC power, and a switching regulator implemented onto the PCB to step the voltage down to 5V. The power systems needed to provide 12V DC for the motors, and 5V for the remaining components including the microcontrollers, touchscreen, sensors, and motor drivers.

H. 3D Reconstruction Software

For our software that generates a 3D model based on the images our stand captures, we chose to use Meshroom by ALICEVISION, which is open-source photogrammetry software. This software will run on our Linux workstation and will have the pictures transmitted to it from the Raspberry Pi. We chose this software because it is free and open-source and provides the required functionality for our project in a well-documented and user friendly way. It produces high quality models while also being relatively easy to work with and use for automation.

III. Hardware Design

The hardware of the project consists of mechanical and electrical components. This section will elaborate on the structure of the project's hardware and how various aspects connect. The hardware block diagram in Fig. 1 illustrates the major components of the project and how they interact. The major components include the PC, Raspberry Pi 3, PCB, motors, sensors, touchscreen, camera, router, and power supply.

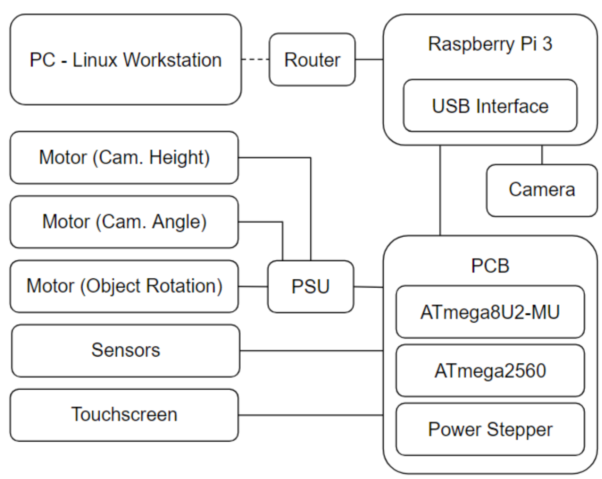


Fig. 1: Hardware block diagram that illustrates the hardware in use and how to connect to each other.

A. Mechanical Hardware

The first element of the mechanical design is the stand that moves the camera vertically. The main structure is the camera stand that lifts the camera vertically and changes its angle. This consists of a base made of 3D printed PLA that supports a vertical aluminum railing. The base houses a motor that connects to a belt that spans the height of the railing, which attaches to a carriage. The carriage holds the camera and houses the motor that tilts it, and it has the sensors on the top and bottom. The turntable is a separate piece of hardware that connects to the stand via wires to power the motor. The turntable plate is 3D printed with a lazy susan bearing that connects to the motor via a belt.

B. Electrical Hardware

The main electrical components are the PC that runs the 3D reconstruction software, the Raspberry Pi that controls the camera, and the PCB that houses the microcontroller that manages most of the other components. The Raspberry Pi is a predesigned development board that does not require any hardware design to modify it, which is the same for the PC running Meshroom. However, we do need to connect the Pi to a router for PC communication and connect it to the PCB via USB. The major elements of hardware design were related to the PCB. Our PCB involves the ATmega2560 microprocessor, the DC voltage converter for the power supply, a motor driver, and pins to connect the touchscreen and sensors.

The ATmega2560 CPU is the major component of our PCB, and the necessary pins are wired through traces to external pins for peripherals, and connected directly to elements such as the motor driver. Fig. 2 shows the connections for the microcontroller that handle the motors and communication with the touchscreen. We also utilized an ATmega8U2-MU chip to act as a USB to serial interface. This serial port interface allows for the ATmega2560 to communicate with the Pi over USB, and acts as a method for loading software onto the ATmega2560 once it has been soldered onto the board.

The motor driver design we chose has a handful of useful features for our project. One driver is implemented on the PCB, and the others are purchased boards that will connect to it via pins. For driver setup, the current draw limit needed to be set to 1A. The drivers can handle up to 2A with a 12V supply, but we chose to run it at a lower amperage to reduce heat and ensure longevity. A potentiometer is used to adjust the current draw limit. The driver needs two digital pin inputs for controlling the motor, which are connected to the ATmega2560. Another feature of the drivers is micro-stepping, which allows us to increase the number of steps that equal a full rotation above the value of 200 at which the stepper motor functions. This will allow us to gain more precise control over the rotation and allow us to achieve smaller angles when tilting the camera or rotating the object. We also inserted a 100µF capacitor at the 12V input line to protect against voltage spikes, as was suggested by the datasheet for the driver design [2]. The motor drivers that are not built into the PCB will be connected via the external pins and wired to the corresponding motors.

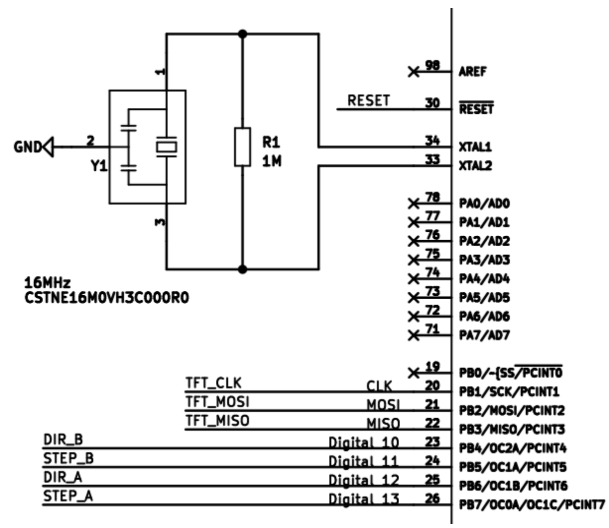


Fig. 2: Schematic for the ATmega2560 microcontroller pin connections on the PCB related to motors and communication.

The hall-effect sensors used for calibration each required one pin for the sensor output, The camera carriage houses the sensors, which detect magnets connected to the vertical railing at the top and bottom. The sensors are placed on the top and bottom of the carriage so that they come into close proximity to the magnets.

The touchscreen required some hardware modification to set up. We soldered pin connectors to the board that allow us to connect it to the PCB, and the connection requires 5 pins for the SPI communication and 4 pins for the touch capabilities. We also soldered the SPI jumper pin on the back of the breakout board to set the communication mode permanently to SPI.

The power supply is another important piece of hardware, with the AC to DC converter being a separate component that plugs into a wall outlet and connects to the PCB, and the voltage regulator being on the PCB. The voltage regulator schematic is shown in Fig. 3, and is used to step down the 12V that is used to power the motors to 5V, which is used to power the remaining components including the ATmega2560 and the touchscreen. We used a buck converter based on the design of the TPS56623 chip.

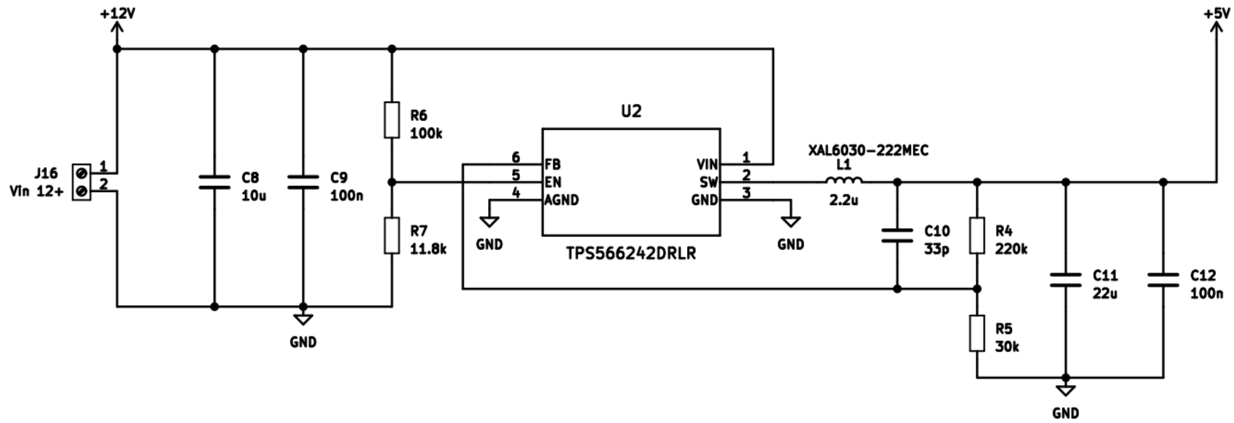


Fig. 3: Schematic for the voltage regulator on the PCB.

IV. Software Design

The software design of the project involves multiple pieces of code that we needed to write, as well as software with which we needed to interface. There is C++ code on the ATmega2560 that handles the motors, touchscreen, and communication. A Python script runs on the Raspberry Pi that communicates with the ATmega2560 and controls the camera via the gPhoto2 software. There is also a Python script in the Linux workstation that receives the pictures from the Pi and processes them in Meshroom, which is the open source SfM software. Communication between the ATmega2560 and Raspberry Pi is done via a USB connection, which removes the need for level shifting circuits due to the differing voltages [3].

A. Raspberry Pi 3 Software

The Raspberry Pi serves three main functions: interfacing with the camera, communicating with the Linux workstation, and saving previous user settings. The Pi will boot upon being powered on, and the script will establish communication with the ATmega2560 through the USB and the camera through gPhoto2. After the ATmega2560 provides the Pi with an IP address through user input, the Pi will connect to the workstation. The Pi then receives commands from the ATmega2560 to take pictures with the camera and transmit them to the workstation when the capturing process has finished. The Pi has the main control over the entire software flow at the beginning of the process and at the end, but it passes control back and forth between itself and the ATmega2560 throughout the capturing process. Fig. 4 shows the control flow of the Pi and when it communicates with the ATmega2560.

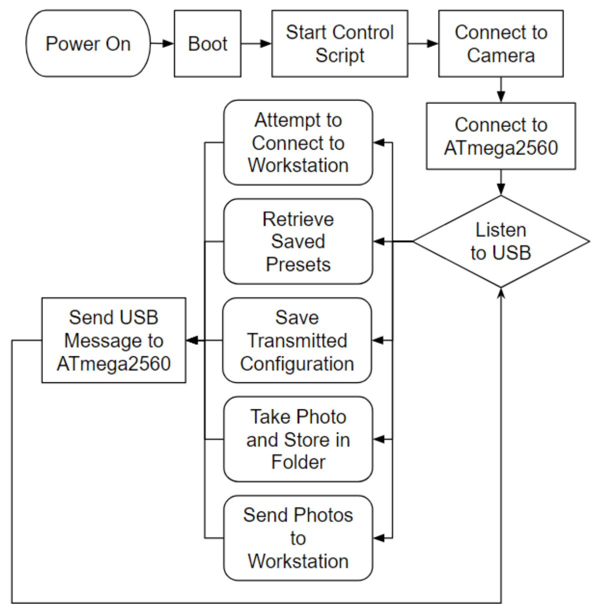


Fig. 4: Software flowchart that illustrates how the Raspberry Pi 3 controls the program flow and passess control between itself and the ATmega2560.

B. ATmega2560 Software

The ATmega2560 serves three main functions: display options on the touchscreen and gather user input, calibrate the position of the camera with the motors and sensors, and position the camera and object using the motors for pictures to be captured. Once the ATmega2560 boots up, it displays a loading screen until it connects to the Pi. Then, the screen will ask for the IP address of the workstation and attempt to connect. Once a connection is made, the ATmega2560 will request the presets from the Pi, and will display the previous settings to the user for quick selection. Selecting a preset will load in the settings, or the user can enter the height of the object, the number of vertical angles, and the number of rotational angles for a new capturing process.

Once the user is ready, the ATmega2560 will send the selected settings to the Pi to be saved, then instruct the user to focus the camera on the object, and calibration will begin. This process involves moving the camera to the bottom and top of the vertical railing, which is detected by the hall-effect sensors, and results in the ATmega2560 knowing the position of the camera and how to translate distance to motor steps.

The ATmega2560 will then ask the user to focus the camera on the object, and confirm to begin the picture taking process. During this process, the ATmega2560 controls the three motors to position the object and camera at the correct height and angles, then sends a signal to the Pi to take a picture with the camera. The motors position the camera at a certain height and angle to aim it at the object, then rotate the turntable so that each angle of the object faces the camera. Once all of the pictures have been taken, the ATmega2560 instructs the Pi to transmit the pictures and displays another loading screen. Once this is done, the ATmega2560 receives a message from the Pi and it will ask the user if they would like to begin the process again. Fig. 5 shows the states of the touchscreen during the entire process.

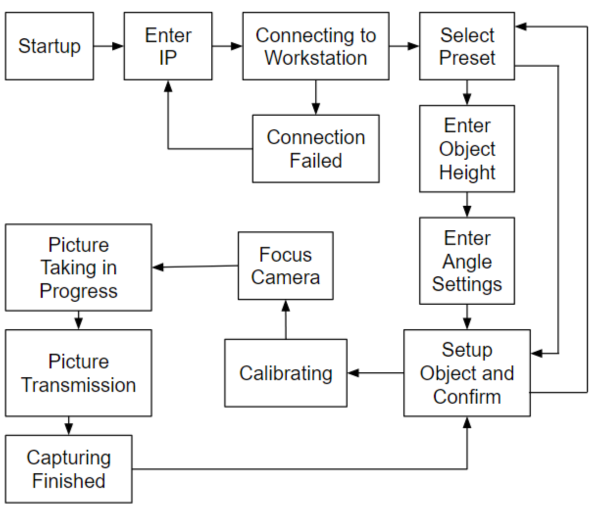


Fig. 5: Touchscreen state machine that shows which screens the display shows to the user.

C. Linux Workstation Software

The main function of the Linux workstation is to accept the pictures from the Raspberry Pi and run them through the Meshroom 3D reconstruction algorithm. The workstation script will begin upon running the application, which will check that Meshroom is available on the user's PC. If not, it will instruct the user to install Meshroom. The workstation UI will then allow the user to set the folder where the 3D model will be saved. The workstation will then establish a connection to the Raspberry Pi and wait for the Pi to have pictures for processing. Once the station has taken a set of pictures of an object, the Pi will transmit them to the workstation to be placed in a folder. The folder of pictures will be put into a queue for processing by Meshroom, which will allow for another set of pictures to be taken while the previous set is being processed. The user will be informed via the workstation UI when the 3D reconstruction process is complete. Fig. 6 shows the workstation flow.

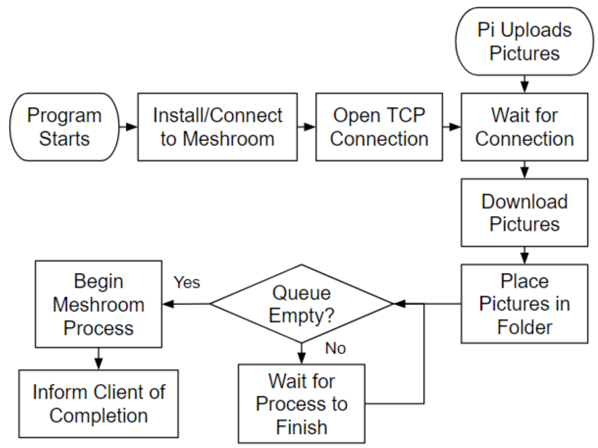


Fig. 6: Software flowchart that illustrates the functions of the Linux workstation.

V. Prototypes & Testing

The design of our project has gone through various changes as we developed prototypes and tested them. The PCB, mechanical structure, and software design all underwent changes during development to better achieve our goals.

A. Printed Circuit Board

Our PCB went through 3 major iterations with moderate changes being made in each revision. V1 was designed to have the microcontroller and power supply built into the board, with all other components including the motor drivers, screen, and USB chip being separate components that would connect to the PCB via external pins. During testing, we found that V1 had inadequate component selection for the power supply. The capacitors and inductors were not all rated for the necessary 12V and 5A, meaning that they could not handle the operation and needed to be upgraded.

For V2, we corrected the parts selection for the power regulator, and upon testing, the power circuit functioned as intended, correctly converting 12V DC to 5V DC. We were then able to test the motor, touchscreen, and sensor control, which functioned as intended, and the communication with the Raspberry Pi which was successful. We tested all individual aspects of the ATmega2560 to ensure our design up until this point worked. However, we still needed to add a peripheral onto the PCB itself, so we designed V3.

For V3, we kept the power supply the same as well as all other components, except for the addition of one motor driver onto the PCB. This design meets the requirements of significant PCB design, but was not too complicated as to introduce design problems. V3 was tested to function as intended, just as V2 did, and meets the requirements for our PCB.

B. Mechanical Structure

The mechanical structure has gone through a few iterations throughout the design process. The initial design for the base of the stand involved legs on four sides. After testing the stability of the stand with the camera at higher positions, we altered the design to be a box which would provide more support. The initial turntable design was constructed and tested, and we found that it supported the required weight and worked well with the motors we chose, so that design remained the same. The carriage design also remained largely the same from its initial design, being a box that houses a motor, with two sensors and the camera mounted to the outside.

C. Software Design

With software design, we began by identifying all the requirements that our design had to meet. For the ATmega2560, this included things such as controlling motors, displaying output and taking input on the touchscreen, and communicating with the Raspberry Pi. For the Pi, the important requirements included communication with the ATmega2560, taking pictures with the camera, and transmitting them to the Linux workstation. For the workstation, this included running pictures through Meshroom and collecting the model. For each of these pieces of software, we tested the capabilities of our designs individually, with separate code to test things such as motors, the touch screen, and camera connectivity.

We wrote software for the ATmega2560 to test the motors, touchscreen, and communication. For the motors, a stepper motor library was used to make controlling them easier, and so we wrote code that was designed to just test the stepper motors and their capabilities. We determined that the library would make utilizing the motors simple, and that they would be able to handle the tasks we required of them, that being lifting the camera and carriage along the vertical railing and rotating the plate with the object on it. We then developed this code to meet the requirements of the entire image capturing process and integrated it with the other functionality. For the touchscreen, we wrote code to test the graphics library and touch library to make sure the screen was able to display all of the information needed and take in input. We programmed test screens that would be adapted into the final UI design to make sure that we could display buttons at a reasonable size to be pressed by the user, and display text at a reasonable size to be read. For communication with the Pi, we wrote code to send messages back and forth to confirm that communication was simple and that the USB chip on our PCB was functioning correctly. This code was then integrated with the rest of the ATmega2560 code to allow communication during the picture taking process.

We wrote software for the Raspberry Pi to test its ability to communicate with the ATmega2560, take pictures with the camera, and transmit them to the workstation. For the ATmega2560 communication, we wrote code to send messages as previously mentioned, which demonstrated that communication was functioning as intended. For the camera, we wrote code to utilize gPhoto2 to take a picture with the camera, which was successful and was implemented into the project. For transmission to the workstation, we wrote code to test communication and we were able to get the picture from the camera testing code to transmit successfully.

We wrote software for the Linux workstation to test the ability to feed pictures into Meshroom and get a 3D model, which was successful, and showed us that Meshroom was a good choice for our 3D reconstruction software.

VI. Demo Process & Engineering Requirements

The demonstration of our project will involve going through the entire photogrammetry process with a few objects of varying sizes. We will capture a varying number of images depending on the object's size and desired level of detail then show the generated 3D model.

There are various engineering specifications that have guided the design of our project, which are shown in Table I. The ones marked as "Demo" will be made explicitly clear during the demonstration process that these have been achieved. During the demo, which showcases the photogrammetry process multiple times for varying objects, we will show that these requirements have been met. Due to the nature of the demonstration, various other specifications will be shown to have been met in addition to the main ones.

The photogrammetry process will now be elaborated upon. Before beginning, the user must download our Python application onto their computer, which will also prompt them to download Meshroom so the 3D reconstruction can take place after the pictures are captured.

TABLE I
ENGINEERING REQUIREMENTS

Specification		Spec Type	Requirement
Object Plate	Object Capacity Limit	Demo	Turntable should handle objects of at least 5 kg in weight
	Object Dimensions	Demo	Minimum plate radius of 10 cm
	Table Rotation Rate	-	Minimum rotational speed of 1 revolution/minute
	Table Stability	-	Table should tilt < 1° with maximum object weight
Camera	Camera Stand Height	Demo	Stand should be at least 70 cm in height
	Camera Adjustability	-	Should point 45° downward
	Camera Quality	-	At least 12 Megapixels
	Capture Speed	-	Shutter speed of around 1/60 sec
MCU	MCU Feature Set	-	Coordinate 3 motors, 1 touchscreen, 1 camera, and 2 sensors
	Transmission Rate	-	Transmit at rate of 100 MB/s
UI	Desktop App Scope	-	Initiate photogrammetry and transfer to algorithm automatically
	UI Polish	-	User interface should feel intuitive and display text and buttons at appropriate sizes
	LCD Size	-	Display should be approx. 9 cm across
Power	Power Output	-	Output at least 12V for the motors

After setup, the process begins by powering on the stand, which will cause the Raspberry Pi and ATmega2560 to boot. The touchscreen will display a loading screen until they connect. Once a connection is made, the screen will ask the user to input the IP address of the workstation, to which the station will then connect. The screen will then ask the user to select a preset if any are available. If this is the first time using the stand, or if the user wants to enter in settings for a new object, they will then enter the height of the object and the number of angles from which they want to take pictures. They can select from between 2 and 5 vertical angles, and 4, 8, 12, 16, or 20 rotational angles. The user will then be prompted to set up the object on the turntable at a certain distance from the stand, and the stand will calibrate the camera's position. The user will then make sure the object is fully in frame and adjust the zoom and focus until they are satisfied. Once they confirm, the picture taking process will begin. The camera will move to a vertical location and angle itself toward the object, then the turntable will rotate the object to capture pictures of it from every desired angle. This will repeat for every vertical position according to the entered settings until the process is complete. The pictures will then be transmitted to a folder in the workstation and put into a queue to be processed by Meshroom.

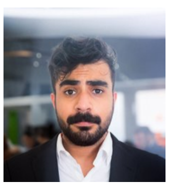
We will demonstrate that the photogrammetry station can take a varying number of pictures of objects of varying sizes, and produce a photo set that can be fed into the 3D reconstruction algorithm to generate a 3D model.

As for the engineering specifications we plan to demo, we have already tested them and ensured we have met the requirements. We have placed objects that we have weighed to be 5 kg on the table and it was able to rotate as expected. We were also able to design the turntable to have a radius of 10cm that turns, as we selected motors with the necessary torque. Finally, the aluminum railing that makes up the bulk of the camera stand is 1m long, and thus our target height for the stand has been achieved.

VII. Conclusion

We believe that our project has successfully achieved our goals and met the engineering requirements. Our goals and requirements informed our design process and technology selection, which allowed us to achieve a working device. We selected components that allowed for the successful implementation of an automated photogrammetry stand that provides a streamlined user experience to capture pictures of an object and feed them into SfM software that creates a 3D model. Our team divided the design work in an effective way to match our experience and strengths. The photogrammetry station can automatically take pictures of an object within a reasonable range of size, and allows for the user to select the number of angles they want pictures from, which allows for flexibility with objects that require more pictures for a higher level of detail.

The Team



Alhusain Ali Al Badi is an electrical engineering major with the most experience in photogrammetry. His major responsibilities included designing the PCB and the mechanical structure of the stand.

Ryan Dimmig is an electrical engineering major with experience in microcontroller programming and user interface design. His major responsibilities included coding for the ATmega2560, with a focus on the touch screen user interface.

Isaac Liljeros is a computer engineering major with extensive experience in software engineering and UI design. His major responsibilities included coding for the Raspberry Pi and Linux workstation, as well as the camera.

Nelson Vargas is an electrical engineering major with experience in Arduino development and motor control. His major responsibilities included working on software and hardware for the motors.

The team is composed of three electrical engineers and one computer engineer, and each member was knowledgeable enough to work on various software and hardware related tasks.

Acknowledgements

The team would like to thank the Senior Design instructors, Dr. Lei Wei, Dr. Aravinda Kar, and Dr. Chung Yong Chan for their guidance in the past two semesters, as well as Dr. Samuel Richie for his assistance in Senior Design 1.

We would also like to thank our review committee, Dr. Suboh Suboh, Dr. John Aedo, and Dr. Mike Borowczak for taking time out of their busy schedules to review our project and provide feedback.

References

[1] B. Kreimer, “A Guide to Photogrammetry Photography,” Online News Association, https://journalists.org/resources/a-guide-to-photogrammetry-photography/. (accessed Jul. 5, 2023).

[2] Allegro MicroSystems, “DMOS Microstepping Driver with Translator And Overcurrent Protection,” A4988 datasheet, (updated April 2022)

[3] J. Woolsey, “Controlling an Arduino From a Raspberry Pi,” Woolsey Workshop, https://www.woolseyworkshop.com/2020/02/05/controlling-an-arduino-from-a-raspberry-pi/ (accessed Jul. 5, 2023).