T.C

MARMARA UNIVERSITY

FACULTY OF TECHNOLOGY

DEPARTMENT OF MECHATRONICS ENGINEERING

3D Scanning System Using the Photogrammetry Method

Abdelwahab Rayis

Muhammed Şeyhoğlu

Dr.Öğr.Üyesi ERSİN TOPTAŞ

İSTANBUL,2025

T.C

MARMARA UNIVERSITY

FACULTY OF TECHNOLOGY

DEPARTMENT OF MECHATRONICS ENGINEERING

3D Scanning System Using the Photogrammetry Method

Abdelwahab Rayis

(170219922)

Muhammed Şeyhoğlu

(170221997)

Dr.Öğr.Üyesi ERSİN TOPTAŞ

İSTANBUL,2025

ACKNOWLEDGEMENT

We would like to express our sincere gratitude to our advisor, Asst. Prof. Dr. Ersin Toptaş, for his invaluable guidance and support during the research and implementation stages of this project. Lastly, we are deeply grateful to our families for their patience and understanding throughout this long journey.

Contents

Contents

LIST OF SYMBOLS

N : Number of Turns

R : Radius

nF : Nanofarad

uF : Microfarad

LIST OF ABBREVIATIONS

2D	: Two-Dimensional
3D	: Three-Dimensional
SFM	: Structure from Motion
MVS	: Multi-View Stereo
PMVS	: Patch-Based MVS
CMVS	: Clustering Views for MVS
CAD	: Computer-Aided Design
CMM	: Coordinate Measuring Machine
CPU	: Central Processing Unit
GPU	: Graphics Processing Unit
GUI	: Graphical User Interface
IP	: Internet Protocol
SCP	: Secure Copy Protocol
MP	: Megapixel
GB	: Gigabyte
CSI	: Camera Serial Interface
GPIO	: General Purpose Input/Output
EMF	: Electromotive Force
DC	: Direct Current
LCD	: Liquid Crystal Display
LED	: Light Emitting Diode
SMD	: Surface Mount Device
PCB	: Printed Circuit Board

LIST OF FIGURES

LIST OF TABLES

Abstract

In this study, a semi-automatic scanning system capable of 3D modeling using the photogrammetry method has been developed. The system integrates mechanical motion capability, image processing, and software support to accurately model small- and medium-sized objects in a digital environment. Photogrammetry offers a low-cost, portable, and accessible solution as it enables 3D modeling using only photographs. In the developed system, motors and a camera were synchronized and controlled via a Raspberry Pi microcontroller. The captured images were transferred to a computer via Wi-Fi, and the 3D modeling process was completed by generating a point cloud using the VisualSFM software.

Although fully automated functionality on the software side has not yet been achieved, key steps such as image capture and data transfer have been carried out in a near-automated manner. The system was designed with a modular structure encompassing mechanical, electronic, and software components, and it was tested as an integrated unit. The results demonstrate that the system operates with the targeted accuracy and provides a practical and feasible solution.

We believe that in future stages, this project can evolve into a more stable, efficient, and high-performance 3D scanning system through enhanced mechanical design, standardized electronic circuits, and fully automated software architecture.

Keywords: Photogrammetry, 3D Scanning, Raspberry Pi, Image Processing, VisualSFM, Point Cloud, Structure from Motion (SfM), 3D Reconstruction, Camera Calibration, Portable Scanning Device, Digital Modeling, Automated Imaging System, Python Programming, Hardware Integr

1. INTRODUCTION

In today's world, digital transformation has significantly increased interest in 3D modeling techniques across various fields—from manufacturing and design to cultural heritage preservation and industrial applications. The growing need to digitally capture objects in a precise, measurable, and visually realistic manner has accelerated the development of 3D scanning systems. Within this development, photogrammetry has gained prominence due to its affordability and portability.

Photogrammetry is a technique that enables the extraction of geometric data from photographs taken from different angles of an object. Compared to high-cost systems such as laser scanning, this method requires minimal hardware and is therefore widely preferred in small- and medium-scale projects. Thanks to advancements in open-source software and microcontroller technologies, photogrammetry systems have become increasingly accessible.

In this study, a photogrammetry-based, portable, and user-friendly 3D scanning machine was designed and prototyped. The system was developed with a near-automated structure that integrates mechanical design, electronic hardware, and software components. During the scanning process, a camera collects object images from various angles under motorized control, the images are transferred to a computer via Raspberry Pi, and 3D model generation is completed using VisualSFM software. Figure 1.1 shows the actual part alongside the model output of the system.

The methods adopted in this project aim to achieve high-accuracy scanning with low-cost components, ensure the system is portable and modular, and establish an infrastructure that can be fully automated in future developments. In this context, the project's development as an engineering product provides valuable contributions in terms of both hands-on experience and industrial applicability.

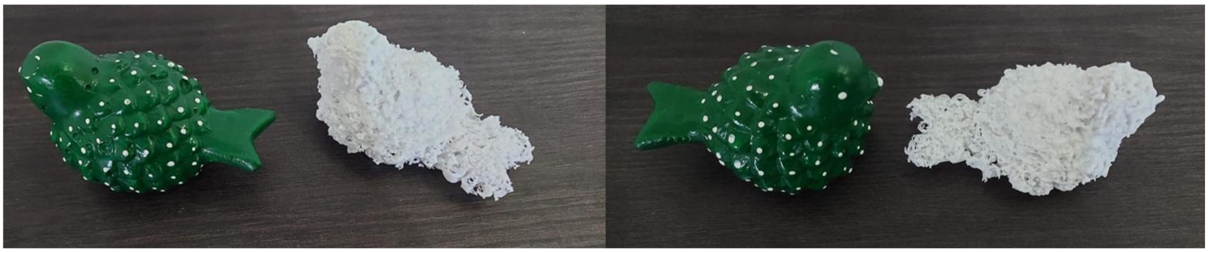

Figure 1-1 Comparison Between the Actual Object and the Scanned Image

1.1. Project Objective

The objective of this project is to develop a portable and cost-effective scanning system capable of creating 3D models of small- and medium-sized objects using the photogrammetry method.

The system is designed to feature a user-friendly structure in which software and hardware components work in full integration. The maximum scanning volume is limited to 11 cm^3, and the system can scan objects weighing up to 1 kg.

The main goals of the project are as follows:

- To develop a scanning algorithm based on photogrammetry and generate 3D models from images taken at multiple angles,
- To establish a semi- or fully-automated photo capture process by integrating a stepper motor and camera with a Raspberry Pi-based microcontroller,
- To generate point clouds and models using open-source software such as VisualSFM,
- To design a simple and effective control structure that enables users to operate the system easily,
- To develop a portable, modular, and economical system architecture that can be used in fieldwork and laboratory settings,
- To offer an accessible, locally developed alternative to high-cost professional 3D scanning systems.

1.2. Project Scope

This project encompasses the design, prototyping, and basic functionality testing of a 3D scanning system based on photogrammetry. The scope involves three main technical components: mechanical design, electronic circuit architecture, and software development. Additionally, the system's integration, testing, and evaluation based on the outcomes are also key parts of the study.

The scope of the project includes the following core elements:

- **Mechanical Design:** Design and fabrication of a two-axis motion system that enables object rotation and camera positioning from various angles. Components were modeled using CAD software and prototyped using a 3D printer.
- **Electronic System:** Integration of a Raspberry Pi-based control unit, NEMA17 stepper motors, DRV8825 motor drivers, and other essential components. Motor control and system feedback circuits were initially tested on a breadboard, followed by PCB design in later stages.
- **Software Infrastructure:** Semi-automation of motor control, photo capture, image transfer, and 3D model creation using VisualSFM. Python was chosen as the programming language, and camera drivers as well as GPIO controls were implemented in it.

- **Photogrammetric Processing:** Generating point clouds and 3D models from multiple images using the Structure from Motion (SfM) algorithm, implemented via VisualSFM.
- **Performance Testing:** The system was tested with images taken from various angles and in different quantities, and model density and accuracy were compared.

Although the system is not yet fully automated, it has been designed with a scalable structure that supports future improvements. The result is a semi-automatic system in which hardware, software, and mechanical components operate in harmony.

1.3. Project Significance

With advancing technology, the need for 3D modeling and scanning systems is rapidly increasing not only in industrial sectors but also in fields such as education, healthcare, cultural heritage preservation, arts, and archaeology. However, commercially available professional 3D scanning systems are often expensive, complex, and limited in terms of portability. This makes it difficult for educational and low-budget research and development activities to access such technologies.

In this context, the developed project aims to meet this need by offering a low-cost, modular, portable, and user-friendly 3D scanning system. Since the photogrammetry method can be implemented using only a camera and open-source software, it enables access for a broad user base. This advantage has been fully leveraged in the project, with the system optimized to achieve high accuracy using minimal hardware.

Furthermore, the fact that the system is built on open-source software and hardware provides an important example of domestic technological production and offers a solid foundation for future development work. The semi-automated nature of the software minimizes user intervention and simplifies the process, making the system usable by users of varying skill levels.

In conclusion, this project stands out as a significant contribution in terms of technological accessibility, practical training opportunities, support for research and development activities, and promotion of local production technologies.

2. Literature Review

In this section, photogrammetry and other 3D scanning methods used in the field of three-dimensional modeling are examined, with a comparative analysis of the advantages and disadvantages of current technologies.

2.1. Fundamentals of Photogrammetry

Photogrammetry is a technique that enables the creation of three-dimensional models by determining the geometric features of objects through photographs taken from different angles [1]. It is not merely a data collection method but a multidisciplinary approach involving various theoretical and software concepts. Photogrammetry is widely applied, especially in fields such as the documentation and preservation of cultural heritage [2].

Today, numerous open-source and commercial software tools have been developed for photogrammetry-based 3D modeling. Tools such as VisualSFM [3], Meshroom [4], and Regard3D [5] have made this method accessible to a broader range of users. This section introduces key concepts that help explain the methods and software used in the project:

- **Single-Image Photogrammetry:** Allows for the extraction of planar coordinates from a single photograph. Commonly used in fields like archaeology and architecture.
- **Stereo Photogrammetry:** Based on estimating 3D coordinates using at least two images taken from different angles. It operates by analyzing parallax differences and is illustrated in Figure 2.1 [1].

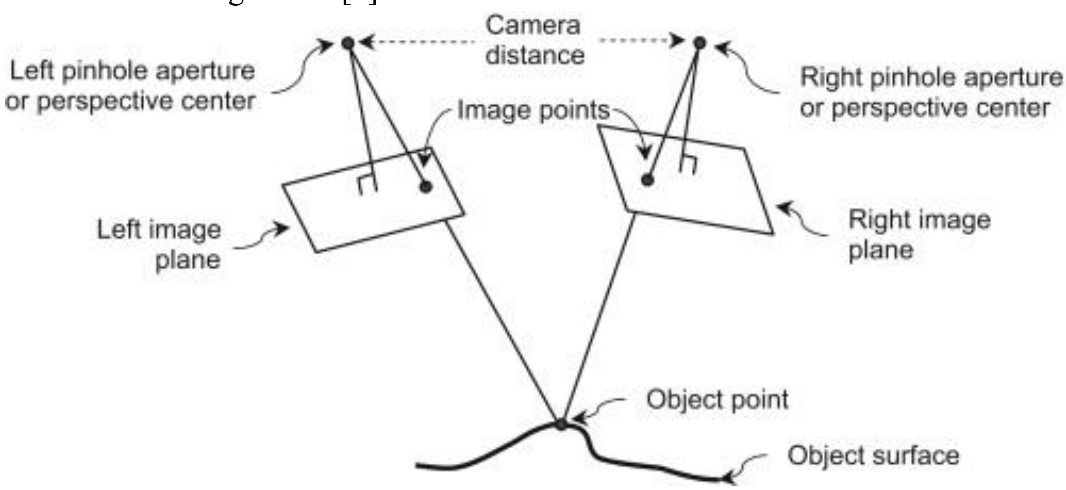

Figure 2-1 Representation of Stereo Photogrammetry Technique Showing Triangulation Point [3]

- **Graphical, Analog, and Analytical Photogrammetry:** Graphical photogrammetry refers to early techniques involving measurements made through drawings. Analog photogrammetry is based on optical-electromechanical systems, while analytical photogrammetry represents the transition to digital modeling.
- **Image-Based Modelling:** Refers to the process of converting images taken from different angles into 3D models using specialized software tools.
- **3D Reconstruction:** The digital re-creation of a physical object or environment in three dimensions.

- **Structure from Motion (SfM):** A method that estimates both camera positions and scene geometry from sequential or multi-angle images. The VisualSFM software used in this project operates based on this technique. It is illustrated in Figure 2.2 [2].

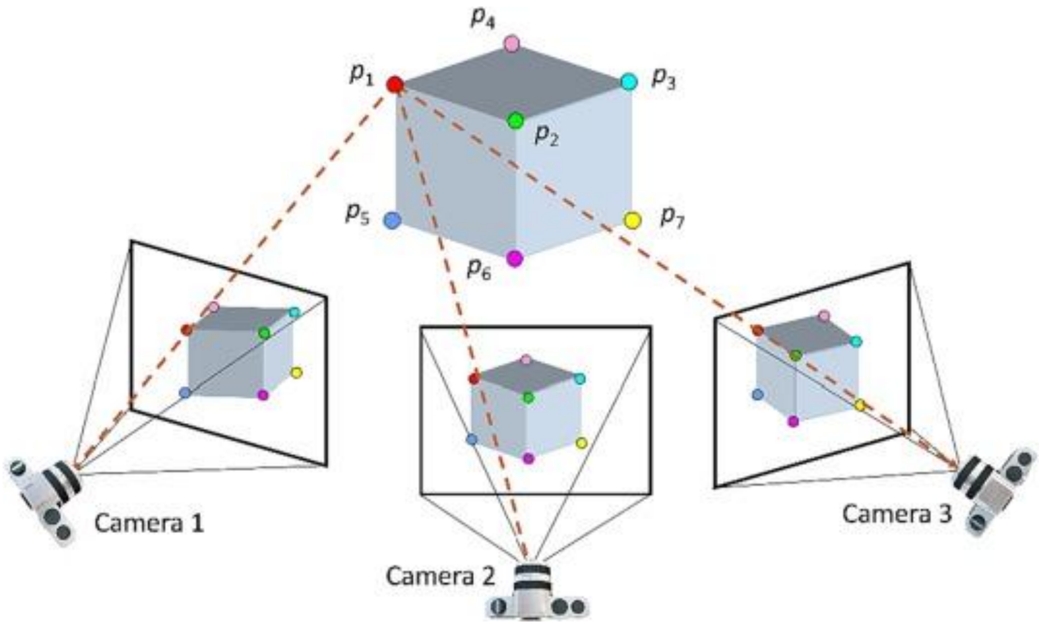

Figure 2-2 SfM Process [2]

- **Multi-View Stereo (MVS):** Builds upon the camera positions estimated through SfM to generate more detailed and dense 3D data (dense point clouds). Modules such as CMVS and PMVS are specific implementations of MVS [6].
- **Point Cloud:** A data set in which objects are represented in space using X, Y, Z coordinates. A sparse point cloud reflects camera alignment, while a dense point cloud includes detailed surface information [7].
- **Mesh, Polygon, and Surface Models:** These models are created after the point cloud and represent the geometry of the object. A mesh model defines only the edges, a polygon model defines surfaces, and a surface model represents the outer shell. Solid models, in contrast, offer more detailed structures by including internal volume information [8].

These concepts are fundamental for understanding how photogrammetry-based systems operate and for evaluating the quality of the resulting output. In particular, SfM and MVS techniques play a critical role in determining factors such as processing time, model detail level, and point cloud quality [9].

2.2. 3D Scanning Methods

Various three-dimensional scanning methods offer distinct advantages and disadvantages depending on the requirements of specific application areas. In general, 3D scanning technologies are categorized into two main groups: non-contact and contact methods [10]. Below is a summary of commonly used scanning techniques and their key features:

2.2.1. Non-Contact Methods

- **Laser Scanning:** Provides highly accurate and detailed measurements by projecting laser beams onto the object surface, resulting in a point cloud [11]. It is especially effective in engineering and industrial applications. However, laser scanners are typically expensive and bulky, which limits their portability.
- **Structured Light Scanning**: Projects a specific pattern (e.g., light stripes) onto the object using a projector to rapidly capture surface geometry. It offers high resolution and speed for scanning small- to medium-sized objects. However, reflective or shiny surfaces can distort the projected patterns, leading to detection issues. As a result, structured light scanners may struggle to deliver reliable results on reflective or transparent objects [12].
- **Photogrammetry:** A passive technique that enables the creation of 3D models using only a standard digital camera and software. As it requires no additional specialized hardware, it is a much more affordable and flexible method compared to others. Photos can be captured in the field using an ordinary camera and then processed via software.

However, achieving high-quality results requires proper photography techniques and adequate lighting conditions [13]. Poor lighting or incorrect exposure may negatively affect the quality of the resulting model. Furthermore, photogrammetry traditionally faces challenges when modeling reflective or transparent surfaces.

2.2.2. Contact Methods

- **Coordinate Measuring Machines (CMM):** This method involves direct contact with the object's surface using a probing device. The coordinates of specific points can be measured with extremely high precision, making CMM a reliable choice in industrial quality control. However, due to their large and stationary nature, CMM devices operate slowly and have limited portability.

Thus, while scanning an entire object in detail with CMM may be impractical, they are frequently used for reference measurements.

2.3. Comparison with Other Technologies

Compared to other 3D scanning techniques, photogrammetry offers several advantages:

- **Cost:** Since it can be performed using just a camera and computer software, it does not require additional scanning hardware, making it a low-cost solution. Entry costs for photogrammetry are significantly lower than those of active systems like laser scanners.
- **Portability:** Most photogrammetry equipment consists of lightweight and portable components such as cameras and tripods. It can be easily used in the field or on-site and even applied in hard-to-reach areas using drones. Its compact and lightweight nature allows the method to be set up and used virtually anywhere.
- **Accessibility:** Due to its relative ease of use and reliance on widely available camera technologies, photogrammetry is accessible to a broad user base, from academics to hobbyists. The availability of free and open-source photogrammetry software further promotes its adoption [14].

However, photogrammetry also has some limitations. For instance, when an object has reflective (bright or shiny) or transparent surfaces, photogrammetry software may struggle to generate reliable point clouds in those areas, resulting in gaps in the model.

Additionally, a sufficient number of photographs must be taken, and proper lighting must be ensured to produce a successful 3D model. Poor lighting conditions or low image quality can negatively impact the model's accuracy [15].

In conclusion, laser scanners are preferred for applications requiring extremely high geometric precision (e.g., large-scale engineering projects), whereas photogrammetry proves superior in scenarios where low cost, portability, and ease of use are prioritized.

3. System Architecture

The developed three-dimensional scanning system is designed as a fully automated and synchronized structure in which the mechanical framework, electronic control unit, and software architecture operate in full integration. The primary goal is to capture images of small- and medium-sized objects from multiple angles to generate high-accuracy photogrammetric 3D models.

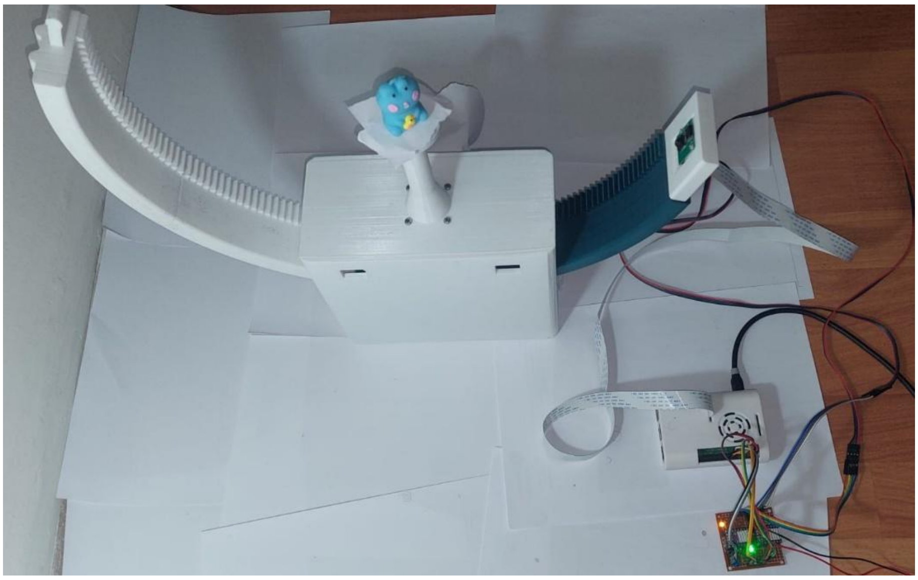

Figure 3-1 Designed System

As shown in Figure 3.1, the physical core of the system consists of a body structure housing a mechanism capable of two-axis movement. The Y-axis movement enables 360° horizontal rotation of the turntable on which the object is placed. This motion is achieved through a large 217 mm radius ring gear with a 20:1 gear ratio, driven by a small stepper motor. Each motor step precisely rotates the turntable, aligning it to predetermined angular positions for multi-angle image capture.

In parallel, the X-axis movement allows the camera carrier to move vertically along a spring-shaped rail integrated into the inner surface of the system. This rail enables the camera to be positioned at different heights incrementally, allowing the object to be scanned both horizontally (Y-axis) and vertically (X-axis). This systematic coverage ensures that the required multi-angle images for photogrammetry are accurately collected. The stepper motor responsible for vertical motion is mounted and controlled precisely to follow this curved path.

The control of this mechanical structure is entirely handled by a Raspberry Pi 4. Both stepper motors are independently and synchronously controlled through DRV8825 motor drivers. While one motor rotates the turntable, the other adjusts the camera's height. At each new position, the software triggers the camera to capture an image. Based on user input specifying the number of horizontal and vertical shots, the software calculates and executes the required motor steps and determines when and how each photo will be taken and named.

The camera module is directly connected to the Raspberry Pi via the CSI (Camera Serial Interface). Each image captured is saved sequentially by the software. Once all X and Y positions are completed, the system returns to its initial position, and all captured images are transferred wirelessly to the target computer.

This integrated system—combining the mechanical components for linear and rotational motion, the electronic control unit for motor and camera coordination, and the software for process scheduling—provides a fully automated 3D data acquisition and modeling workflow. In the final stage, the collected images are processed on an external computer to first generate a sparse point cloud, then a dense point cloud, and optionally followed by mesh and texture stages to produce a complete 3D model. The system's workflow diagram is illustrated in Figure 3.2.

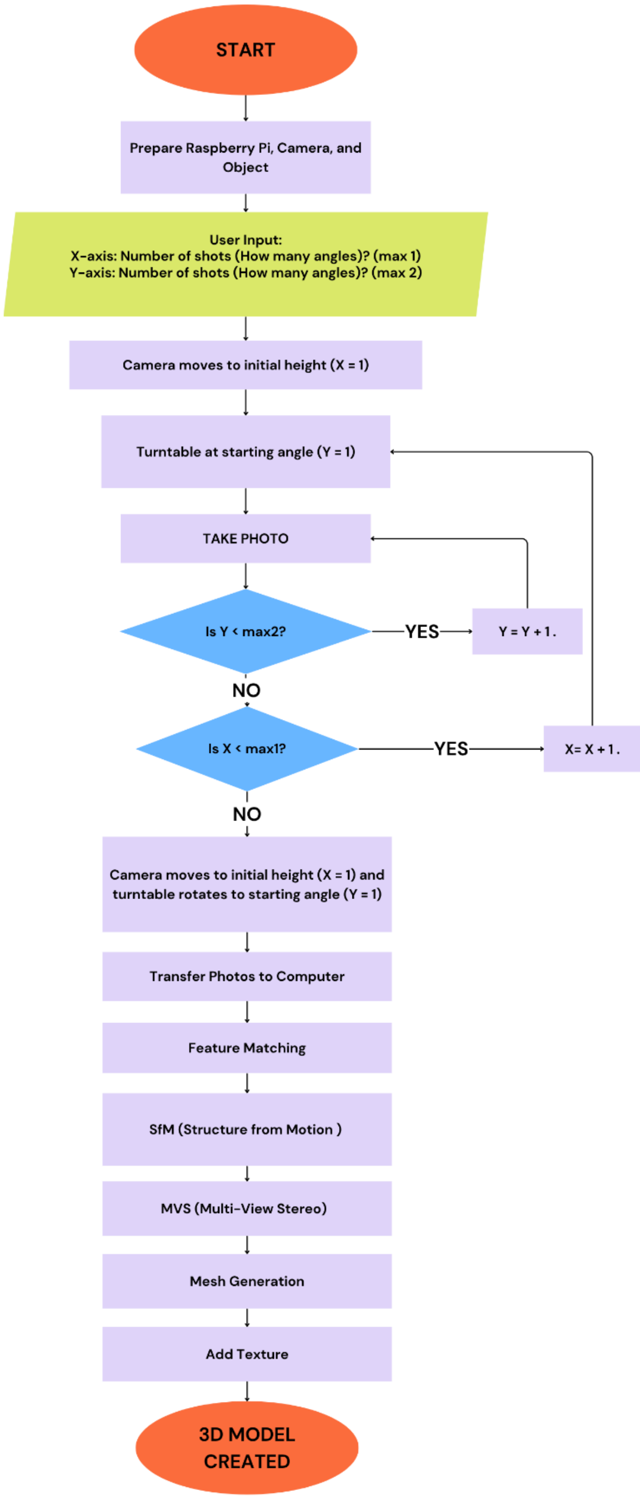

Figure 3-2 System Workflow Diagram

3.1. Mechanical Design

This section explains the mechanical infrastructure of the developed photogrammetry-based 3D scanning system within the context of the overall system architecture. The aim is not only to describe individual components, but also to clarify their functional relationships within the system, how their movements are coordinated, and why this particular mechanical setup was chosen.

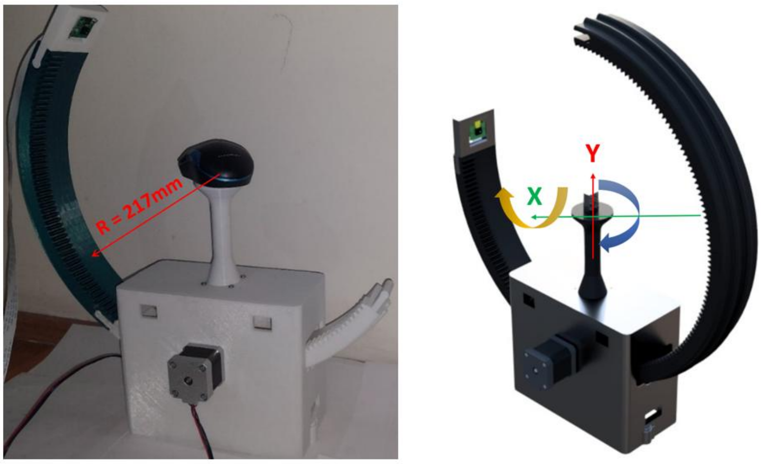

Figure 3-3 Mechanical Design of the System

The developed system shown in Figure 3.3 enables the automatic acquisition of multi-angle images required for photogrammetry by providing movement along two axes. These axes are defined as follows:

- Y-Axis: Horizontal rotation of the object on the turntable through a full 360° circle.
- X-Axis: Vertical movement of the camera and the large ring gear along a rail, allowing the system to capture images from various heights.

3.1.1. Motion Mechanism and Gear Transmission Structure

At the core of the system lies a turntable on which the object is placed. This turntable is rotated via a stepper motor that provides movement along the Y-axis. The motor drives a small gear with a 20 mm diameter, which in turn rotates a large ring gear with a 217 mm diameter, as shown in Figure 3.4. The gear ratio is defined as 1:20, meaning that for every 20 rotations of the small gear, the large gear completes only one full rotation.

This gear ratio enables high precision and fine incremental angular positioning of the turntable. Mathematically, this relationship can be expressed as:

$$\frac{N_{Small}}{N_{Large}} = \frac{1}{20}$$

where **N** denotes the number of rotations for the respective gear.

Figure 3-4 Large and Small Gear Rings

The rotation of the turntable provides the angular diversity required for photogrammetry. This part of the system is directly related to the Y-axis movement and is illustrated in Figure 3.5.

Figure 3-5 Turntable

3.1.2. Camera Carrier Structure and X-Axis Movement

The camera is mounted on a carrier that moves vertically along a spring-shaped rail integrated into the large ring of the system, as shown in Figure 3.6. This carrier structure is controlled by a second stepper motor, which enables precise vertical positioning along the X-axis. The shape of the rail is optimized so that the camera maintains a central viewing angle toward the object at every level.

This motorized carrier moves the camera to specific vertical positions. At each height, the turntable rotates again along the Y-axis to capture images from various angles. As a result, each photograph provides a unique perspective of the object, combining both a different vertical level and a different horizontal angle.

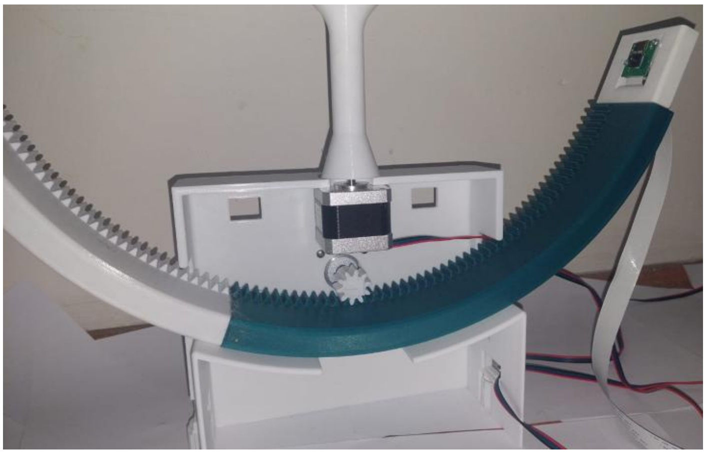

Figure 3-6 Internal Design of the System

3.1.3. System Integration and Design Rationale

This mechanism was chosen based on the following design considerations:

- Y-Axis: Enables rotation of the object, providing a wide range of viewing angles for image diversity.
- X-Axis: Allows vertical movement of the camera, capturing data related to the object's depth and contributing to accurate 3D structure modeling.
- Gear Ratio (1:20): Offers high angular resolution for precise camera positioning.
- Spring-Shaped Rail: Ensures the camera remains focused on the center of the object at all vertical levels.

- Centrally Symmetrical Structure: The entire system is designed with central symmetry, which helps minimize alignment errors during 3D reconstruction.

Figure 3-7 System During Multi-Angle Image Capture

The dimensions of the turntable have been set to approximately 110 × 110 × 110 mm, ensuring the safe placement of small- and medium-sized objects. The camera carrier mechanism is designed to position the camera up to a maximum height of 217 mm along the vertical axis. This sizing allows the system to remain compact and portable while still enabling the capture of a sufficient number of images with the necessary angular and height variations required for photogrammetry.

In summary, the system—comprising the turntable, large ring–small gear drive mechanism, and camera—provides precise and repeatable movement along two axes. This enables the acquisition of high-quality image data suitable for photogrammetry. While the structure is built from low-cost components, it ensures sufficient mechanical stability and accuracy from an engineering standpoint. All images are captured in synchronization with motor movements controlled by the software, resulting in a consistent and multi-angle dataset essential for successful 3D modeling.

3.2. Electronic Design

This section provides a detailed examination of the electrical and electronic infrastructure of the developed 3D scanning system. The system consists of stepper motors that control the movement of mechanical components, motor driver modules, an image capture module, and a microcontroller that serves as the central control unit. The electronic system design is illustrated in Figure 3.8.

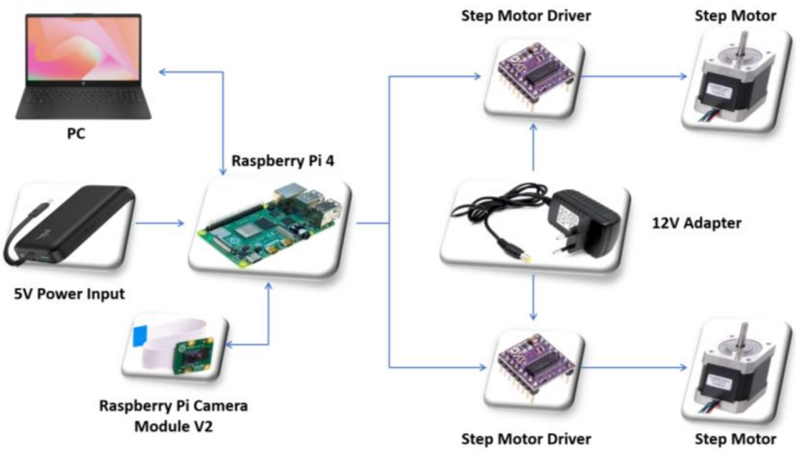

Figure 3-8 Electronic System Design

In the electronic design process, low-cost yet reliable components were selected with the aim of making the system portable, modular, and user-friendly. The precise control of the motors, synchronized with image capture, has a direct impact on the accuracy of the photogrammetric modeling process.

3.2.1. Electronic Components, Their Functions, and Specifications

- **Raspberry Pi 4 8GB – Model 4B:** In this project, the Raspberry Pi 4 was selected as the system's central control unit. Its high processing power, hardware compatibility, and low energy consumption make it a suitable platform. The pin layout of the Raspberry Pi is shown in Figure 3.9. To prevent overheating during intensive processing, a cooling fan was added. Key features include:
  - 8GB RAM and a 64-bit quad-core processor offering high computational performance
  - Direct support for the camera module via the CSI
  - Easy integration with hardware such as motor drivers and LEDs through GPIO pins
  - Energy efficiency due to low power consumption
  - Wireless data transfer capability via built-in Wi-Fi
  - Compatibility with the V2 camera module, enabling image processing and control to be managed simultaneously
  - Use of a flat cable allows for a compact and organized system design

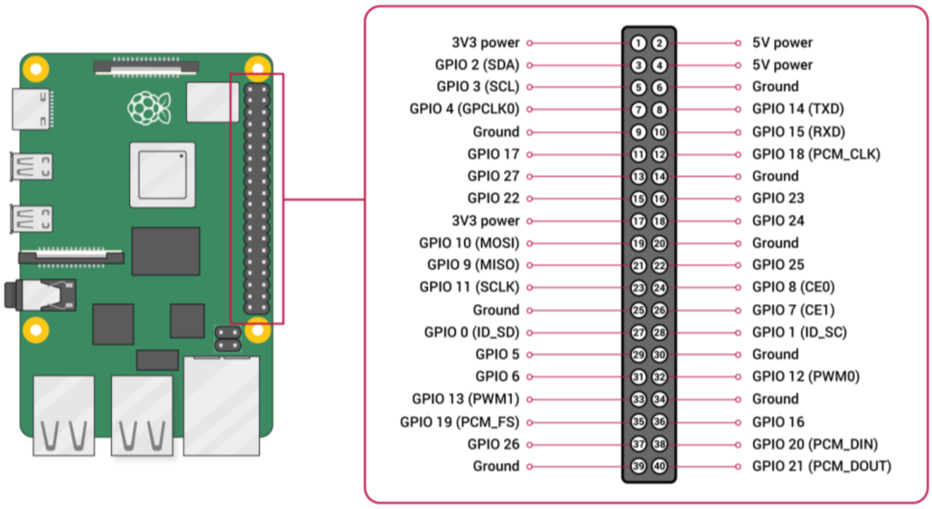

Figure 3-9 Raspberry Pi 4 Pinout [16]

- **Raspberry Pi Camera Module V2 (Sony IMX219):** This camera module serves as the system's image capture unit and is optimized for photogrammetry applications. It is shown in Figure 3.10. Key features of this camera include:
  - 8 MP resolution (3280×2464 pixels) and a fixed-focus lens, offering high-quality images
  - Connects directly to the Raspberry Pi via the CSI interface, enabling a compact and integrated design
  - Uses a 15-pin ribbon cable for connection, supporting neat and space-efficient system assembly

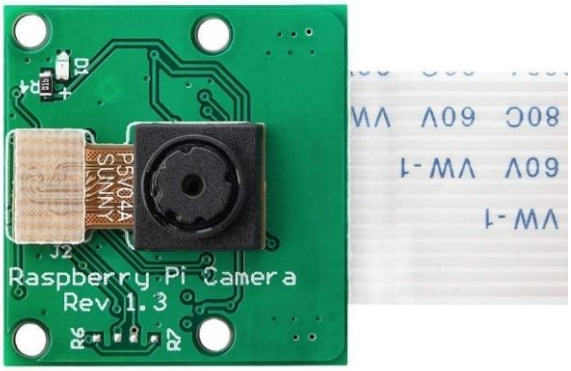

Figure 3-10 Raspberry Pi Camera Module V2 [17]

- **NEMA17 Stepper Motors (2 units):** These motors are used to drive mechanical movement along two different axes in the system:
  - **MOTOR1:** Rotates the turntable on which the object is placed, providing 360° motion along the horizontal axis.
  - **MOTOR2:** Moves the camera carrier vertically along the semicircular rail, enabling vertical positioning along the X-axis.

Figure 3-11 NEMA 17 Step Motor [18]

As shown in Figure 3.11, NEMA17 motors are compact and capable of providing precise motion, making them commonly used in low-cost automation systems. A comparison of stepper motors with other types of motors is presented in Table 1.

Due to their body weight, the motors can remain stable without the need for additional supports, helping to reduce structural vibrations within the system. A technique called microstepping, used to increase step precision, will be discussed in later sections. Key specifications of the motor are as follows:

- Dimensions: 42.3 mm × 42.3 mm × 38 mm (excluding shaft)
- Weight: 280–285 g
- Shaft Diameter: 5 mm (D-type shaft)
- Step Angle: 1.8° (200 steps per revolution)
- Number of Phases: 2 (bipolar configuration)
- Phase Voltage: 2.6–2.8 V
- Phase Current: 1.68–1.7 A
- Cable Length: 30–100 cm (4-wire)

Table 1 Comparison Between Step, DC, and Servo Motors

Feature / Motor Type	STEP MOTOR	DC MOTOR	SERVO MOTOR
Position Control	Yes (step by step, precise control)	No (position unknown without feedback)	Yes (precise with feedback)
Angular Precision	High (1.8°/step, more precise with microstepping)	Low (PWM speed control, position difficult)	High (generally below 0.1°)
Cost	Low	Low	Medium–High
Driver Requirement	Simple (like DRV8825)	Simple (motor driver or direct)	Requires feedback driver
Feedback System	Not required (can work open loop)	Required (encoder necessary for accuracy)	Present (built-in encoder included)
Performance Under Load	Stable, no slip	May slip, speed varies	Stable (PID controlled)
Size & Mounting	Compact, common sizes	Compact, various sizes	Large and heavy
Software Control	Easy, direct step count input	Requires speed and duration control	C omplex PWM and PID tuning may be required

- **DRV8825 Stepper Motor Driver (2 units):** To control the stepper motors, driver boards equipped with the DRV8825 integrated chip were selected. These drivers meet the system's precision control requirements thanks to their compact design, wide voltage range, and support for microstepping. Figure 3.12 shows the driver and its connection configuration. Key features of the DRV8825 stepper motor driver include:
  - Motor supply voltage range: 8.2V to 45V
  - Equipped with a current-limiting potentiometer, allowing adjustment of the maximum current delivered to the motor coils
  - STEP and DIR pins operate at logic level, making it easy to interface with microcontrollers such as Raspberry Pi
  - Includes an onboard 5V regulator, eliminating the need for a separate logic power supply
  - Standard 2.54 mm (0.1") pin spacing, enabling easy installation on breadboards and PCBs
  - It is recommended to use a 47–100μF / 35V capacitor to protect against reverse EMF generated in the motor coils

- When operating above 1A current, the driver can reach high temperatures and must be used with a heatsink for safe operation

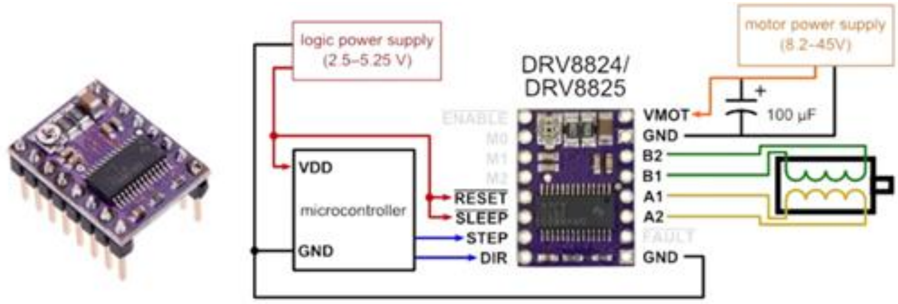

Figure 3-12 DRV8825 Stepper Motor Driver and Its Connection with the Microcontroller [18]

- **Other Electronic Components:** To ensure power stability, suppress electrical noise, and protect components from potential voltage-related damage, several passive electronic elements have been incorporated into the system. One of the most critical components is the 100 nF ceramic capacitor, which is placed on the power lines supplying the Raspberry Pi and camera module. It effectively suppresses high-frequency noise and ensures cleaner power delivery. To filter potential reverse voltages (back EMF) generated by the DRV8825 motor driver in the motor coils, a 47–100 µF 35V electrolytic capacitor is connected to the motor power supply line. The entire system is powered by a 12V DC adapter. While the stepper motors and their drivers operate directly from this main power source, a separate 5V power line is used for the Raspberry Pi. Wiring has been selected with appropriate thickness and minimal length to ensure stable power and data transmission. Additionally, a 10kΩ resistor is connected in series with an LED placed in the system to indicate operational status. This setup not only informs the user when the system is powered on but also protects the LED from overcurrent damage.

3.2.2. Circuit Design

The electronic connections and PCB designs of the system were prepared using KiCad, an open-source electronic schematic and PCB design software. The circuit layout is shown in Figure 3.13.

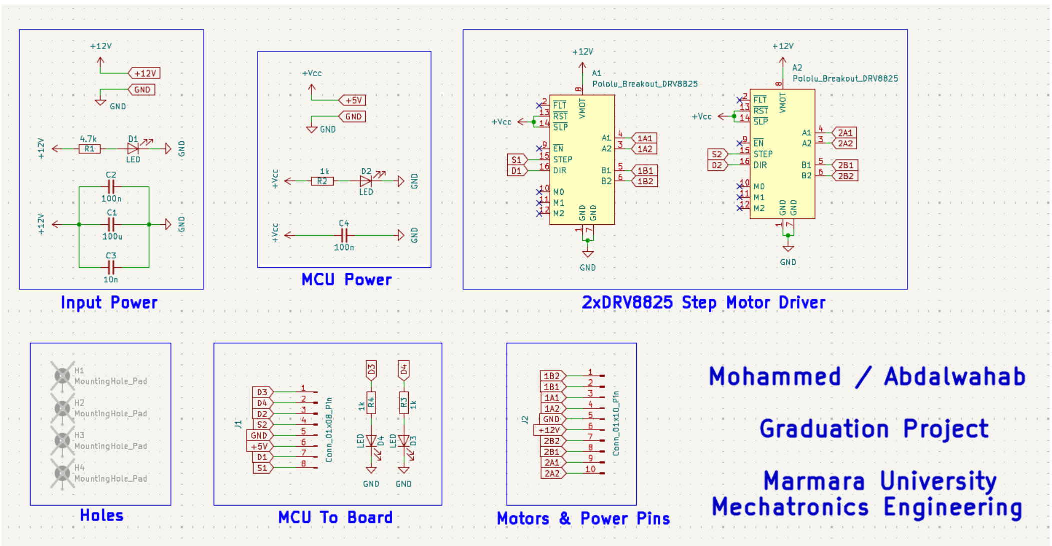

Figure 3-13 Circuit Diagram

During the design process, the placement of components, power lines, and signal traces was optimized to ensure stable and reliable system operation. The components on the circuit were arranged to facilitate easy testing and maintenance.
In the initial phase, system testing was carried out on a perforated protoboard with manually soldered connections, as shown in Figure 3.14. The DRV8825 motor drivers, LEDs, and resistors were soldered onto the board, and their connections to the Raspberry Pi were established using jumper wires.
Once the functionality of the design was validated during the prototype phase, a custom PCB (Printed Circuit Board) was developed to ensure a more professional and stable long-term solution.

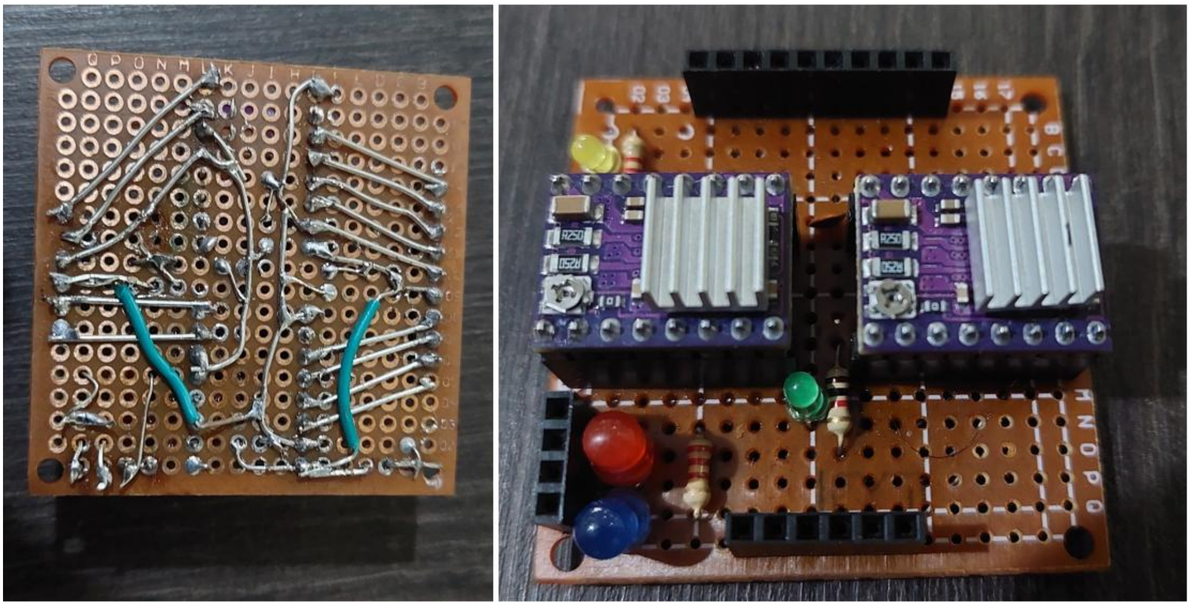

Figure 3-14 Soldered Circuit

In summary, the system’s electrical structure integrates power distribution, motor control, and image processing functions into a cohesive operation. The stepper motors, camera module, and drivers—synchronized via the Raspberry Pi 4 microcontroller—form the foundation of a low-cost, portable, and reliable 3D scanning system.

3.3. Software Architecture

The developed 3D scanning system operates in full integration with a software infrastructure that coordinates the sequential operation of hardware components and synchronizes image capturing with these mechanical movements. The software manages the capture cycle, saves the acquired images in an organized manner, and automatically transfers the data to an external computer. As a result, a large volume of visual data can be collected systematically without operator intervention.
The software is written in Python and runs on the Raspberry Pi 4. Two-axis motion in the system is achieved via stepper motors, which are controlled using DRV8825 motor drivers. Image capture is performed with the Raspberry Pi Camera Module V2, and both motors and the camera are connected directly to the microcontroller through GPIO pins and the CSI interface. This configuration allows all timing and control processes to be synchronized through a single processor.

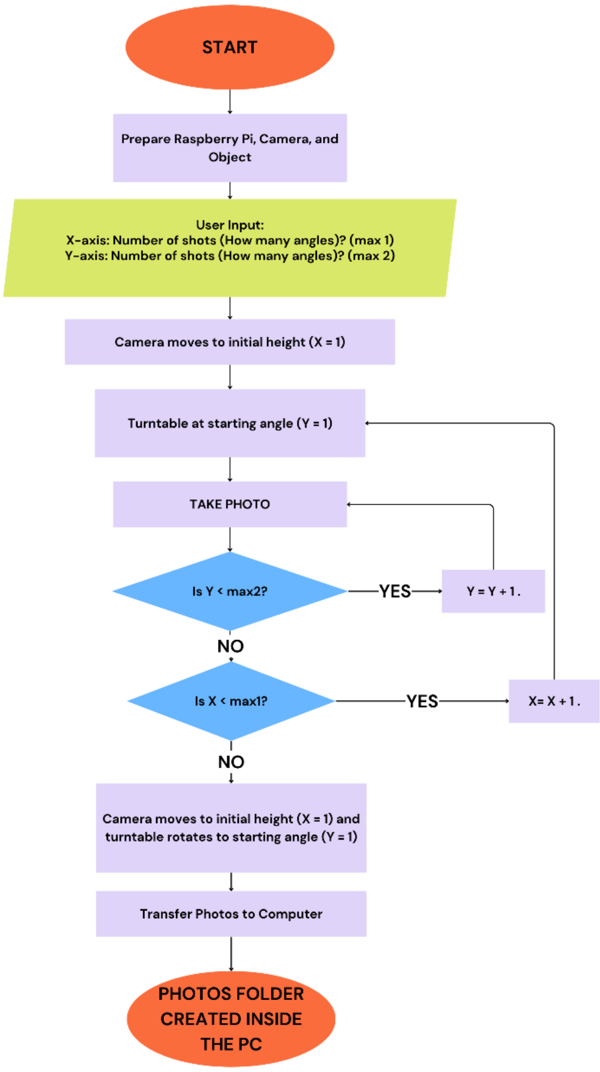

Figure 3-15 Raspberry Pi Code Algorithm

When the code is executed, it initializes all hardware components. Based on the capture parameters provided by the user—namely the number of vertical and horizontal positions—the system moves the camera carrier to each vertical level. At every level, the object is rotated on the turntable to specific angles, and high-resolution images are captured from each angle. The captured images are saved using a naming convention in the format: “v_d_h.jpg”, where v stands for vertical level, d for dataset, and h for horizontal angle.

Once the scanning process is complete, the system automatically returns to its starting position and transfers all images to a designated computer using the SCP (Secure Copy Protocol). This eliminates the need for manual file transfers and ensures well-organized data management.

While the data collection and transfer processes are fully automated, the 3D modeling phase still requires manual user input. The transferred images must be manually loaded into photogrammetry software (such as VisualSFM or COLMAP) to initiate the model reconstruction process. For this reason, the system is classified as a semi-automatic 3D scanning turntable with highly automated data acquisition and transfer capabilities.

The software prompts the user—via the terminal interface—to input the number of photos to be taken in the horizontal and vertical directions. As shown in Figure 3.16, the user is asked to enter these values when the script is executed.

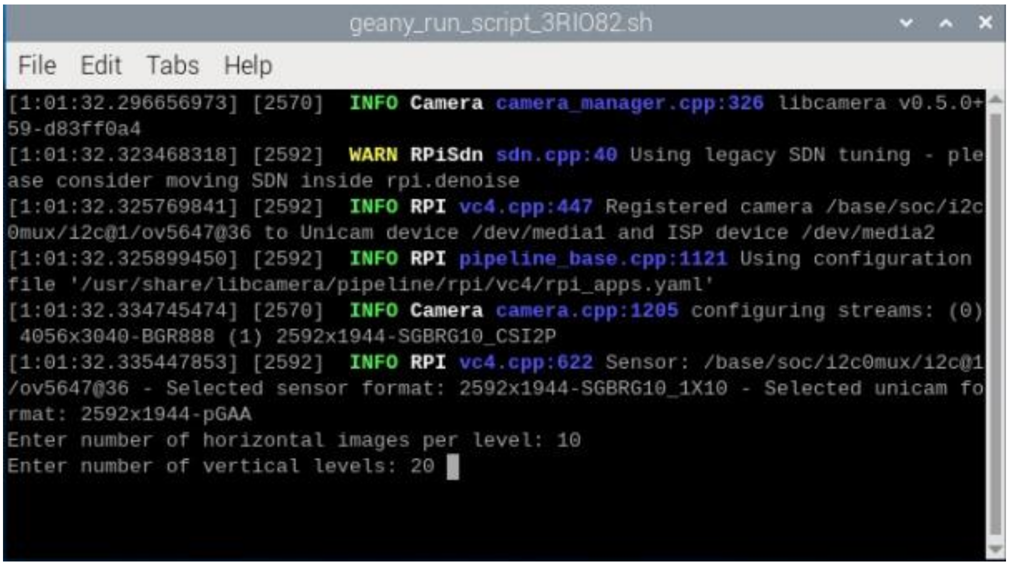

Figure 3-16 User Input for Number of Photos

The system then proceeds to perform image capture by moving the motors in fixed increments at each predefined vertical level. At every vertical position, a corresponding number of equally spaced horizontal rotations is performed, with a photo taken at each angle.

All images are systematically saved in the format v(vertical)_h(horizontal).jpg in the directory */home/pi/Desktop/photos/*.... For example, an image named v2_h6.jpg indicates that it was taken at the second vertical level and the sixth horizontal position.

Figure 3.17 illustrates how the captured images are stored on the Raspberry Pi.

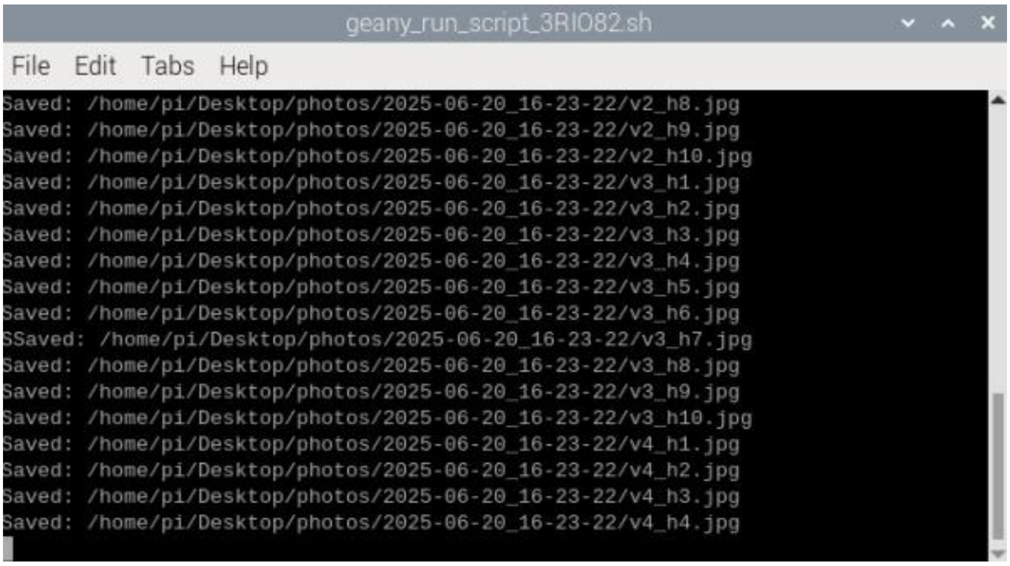

Figure 3-17 Saving Captured Photos on the Raspberry Pi

After the image capture process is complete, the system returns the motors and camera to their initial positions, effectively ending the scanning sequence. Following this, all captured images are automatically transferred to predefined computers with known IP addresses using the SCP (Secure Copy Protocol). This approach eliminates the need for manual data transfer, ensuring seamless automation.

Throughout the process, terminal output displays status messages to help the user track each step. As shown in Figure 3.18, the user is required to enter a password in order to authorize the transfer of all captured photos to the computer.

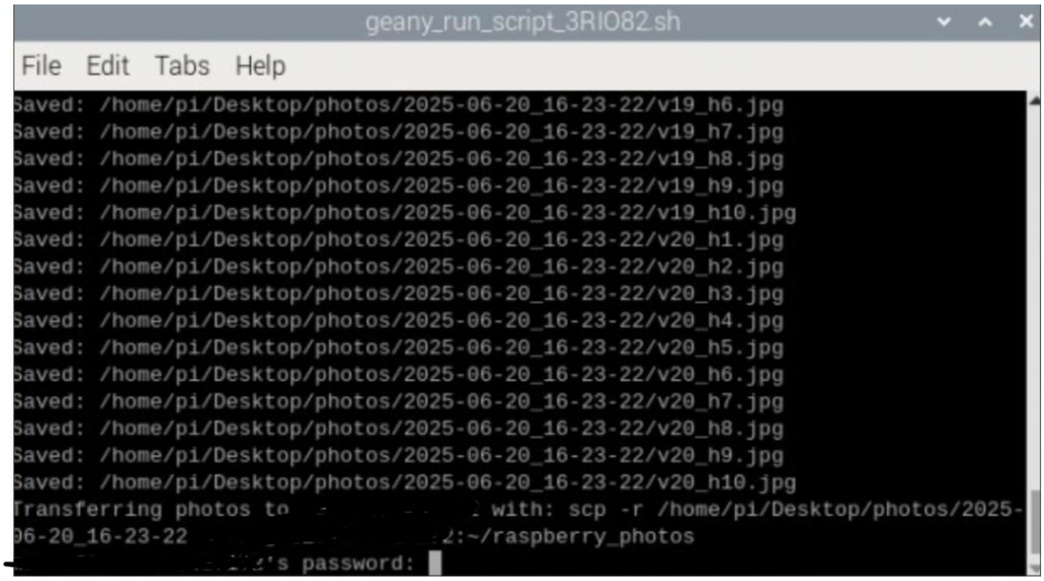

Figure 3-18 User Password Entry for Saving Captured Photos to the Computer

3.4. Image Processing and Modeling Software

The photos captured and systematically named by the Raspberry Pi are transferred to a computer, where they are converted into three-dimensional models using open-source photogrammetry software. The primary tool used in this process is VisualSFM.

Alternative software options such as COLMAP, OpenMVS, and Regard3D were also explored. However, VisualSFM was selected based on its compatibility with the system's hardware and its ability to perform effectively without requiring high computational resources.

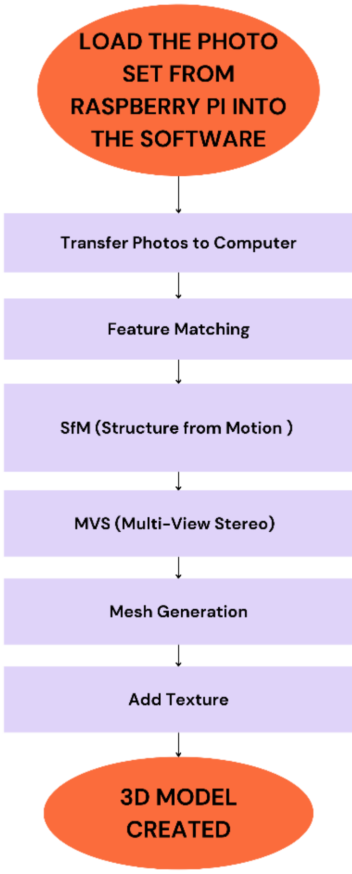

Figure 3-19 Software Program Algorithm

VisualSFM, with its user-friendly graphical interface, makes the generation of sparse point clouds highly accessible and straightforward. Additionally, it supports the creation of dense point clouds through integration with the CMVS and PMVS modules.

One of the software's key advantages is that it can operate entirely on the CPU, without requiring a CUDA-supported GPU. This makes VisualSFM fully compatible with the hardware used in our system, enabling effective 3D modeling without the need for high-end graphics processing capabilities.

The figures below illustrate the VisualSFM interface and the step-by-step usage process of the software.

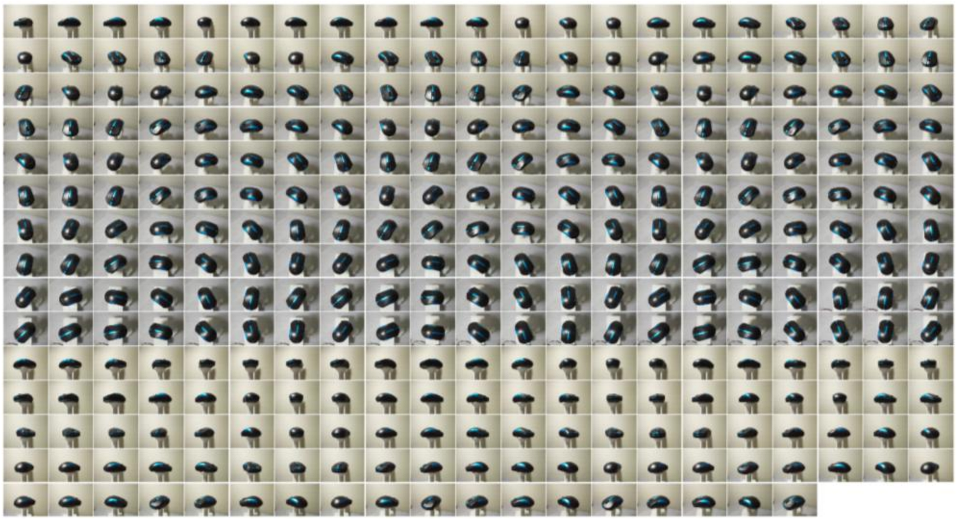

Figure 3-20 VisualSFM Interface

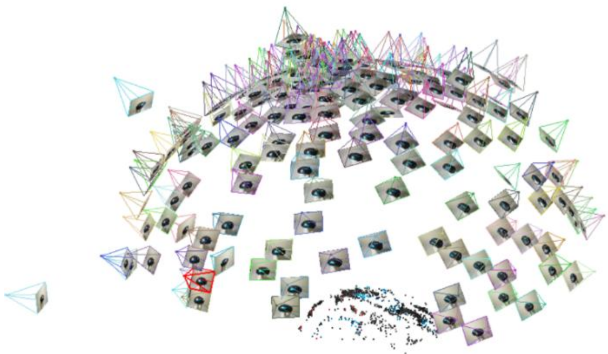

Figure 3-21 3D Reconstruction

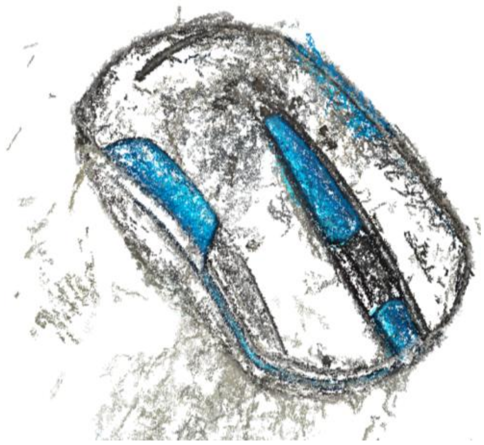

Figure 3-22 Dense Reconstruction

The COLMAP software, which was considered as an alternative, offers highly advanced SfM (Structure from Motion) algorithms. However, it requires a CUDA-supported GPU for dense reconstruction tasks. Since this requirement exceeds the processing capabilities of our system's current computer hardware, COLMAP could not be used effectively in practice. Similarly, OpenMVS, which operates based on COLMAP outputs, also demands high computational power and thus presents similar limitations.

Choosing VisualSFM aligns well with the low-cost and portable nature of our system, as it allows all image processing tasks to be handled via CPU-based computation. In addition, its GUI-based interface enables users to manually intervene and control the modeling process, which adds flexibility and ease of use.

Other GUI-based alternatives such as Regard3D were also evaluated. However, VisualSFM was ultimately selected due to its combination of user-friendliness and hardware compatibility.

For SfM and MVS algorithms to perform successfully, accurate image alignment is crucial. Therefore, motor movements are executed at fixed angular increments without vibration, and images are systematically named in the format v_d_h_k.jpg. This structured approach ensures sufficient overlap between images, creating an optimal dataset for high-quality 3D model generation.

4. Application and Results

In this section, a practical test of the developed photogrammetry system was carried out, and its performance was evaluated through both quantitative and qualitative outputs. A scanning procedure was conducted on a selected sample object to assess the mechanical and software integration of the system. Throughout the 3D modeling process, the quality of the captured images, the impact of camera and lighting positioning, and the effect of using a uniform background on the final results were analyzed.

Additionally, point cloud densities obtained under different test scenarios were compared, and the resulting 3D model visuals are included in the appendix section.

4.1. Sample Scanning Application

To evaluate the system's performance, a small duck figurine was selected as the test object. The model was scanned under varying lighting and background conditions to observe the system's effectiveness in generating a 3D model. As shown in Figure 4.1, the system's modeling performance was monitored and visually assessed.

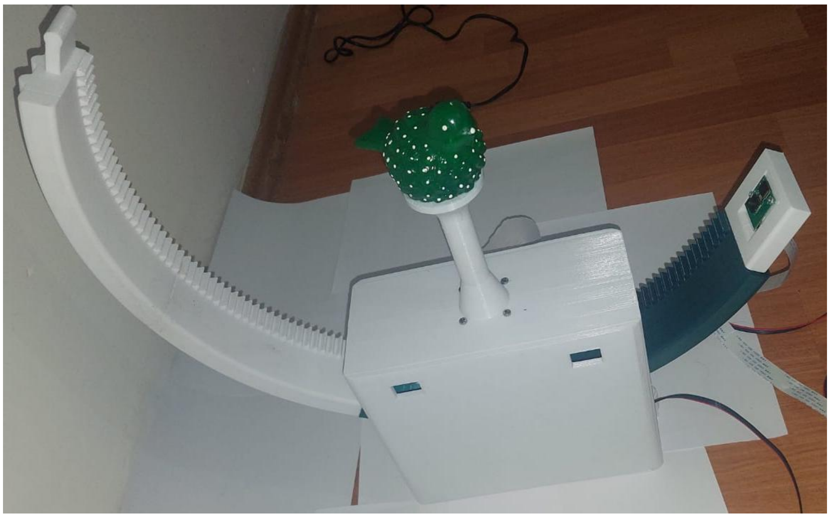

Figure 4-1 Scanning Process

During the application phase, the duck model was placed on a fixed turntable and photographed from a total of 300 different angles as determined by the predefined motor steps. A consistent distance of approximately 10 to 15 cm was maintained between the camera and the model, ensuring sharp and scale-consistent images. As shown in Figure 4.2, a white background was used to enhance image clarity and segmentation.

Figure 4-2 With White Background

The model’s background was intentionally selected as a completely white, uniform surface to provide ideal conditions for image processing. This choice allowed the software to more accurately detect the model’s edges, significantly reducing segmentation errors caused by background interference. All photos were captured automatically via the Raspberry Pi controller and transferred wirelessly (via Wi-Fi) to the computer for processing.

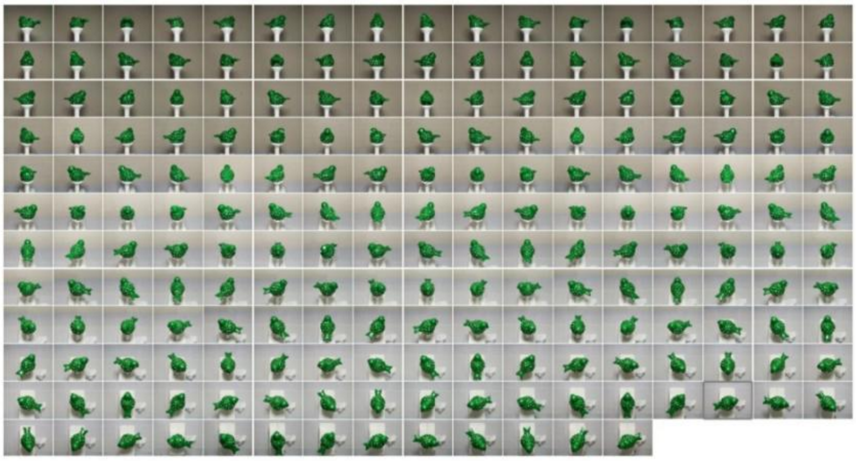

Figure 4-3 Photo Set

With proper lighting, the surfaces of the model were evenly illuminated, reducing shadows and enhancing overall image quality. The uniform white background allowed the software to more clearly detect the model's boundaries, minimizing segmentation errors to the greatest extent possible.

4.2. Generated 3D Models

Following the image capture and data transfer process, the collected images were imported into VisualSFM software. Using the Structure from Motion (SfM) method, a sparse point cloud was first generated.

At this stage, the system identified common feature points across the images, enabling the estimation of camera positions in 3D space and constructing the basic geometric framework of the model.

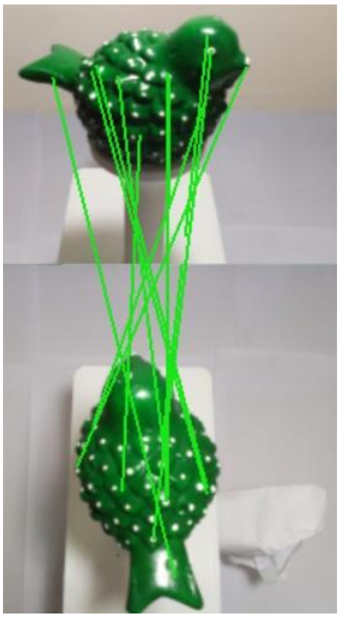

Figure 4-4 Feature Matching Between Two Different Images

Figure 4.4 shows the feature matching process between two images taken from different horizontal and vertical angles.
On average, approximately 4,000 to 9,000 feature points were detected per scan. This sparse point cloud provided a sufficient foundation for defining the overall geometry of the model.

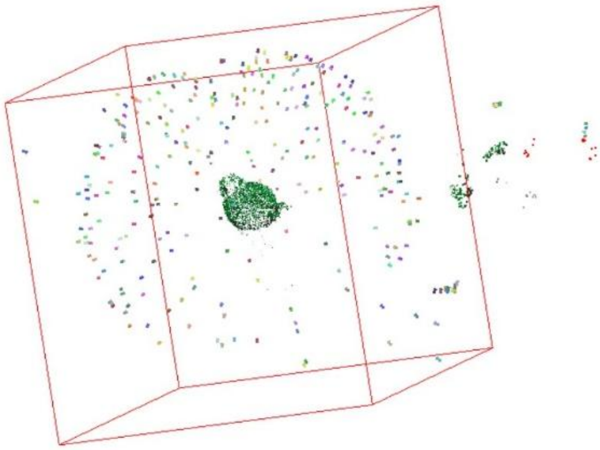

Figure 4-5 3D Reconstruction Application

Following the SfM (Structure from Motion) phase, a dense point cloud was generated. In this stage, the sparse point cloud served as a reference, and all images were matched with depth information to enhance surface detail.
During the dense reconstruction process, approximately 10,000 to 800,000 points were produced, enabling a much more detailed representation of the model's surface features.

Figure 4-6 Dense Reconstruction

Finally, a surface mesh was generated using the mesh reconstruction module. In this step, the dense point cloud was converted into a network of triangular surfaces, completing the 3D geometric structure of the model.
The resulting mesh model was then subjected to optical refinement and surface smoothing processes to improve visual quality. Apart from some minor overlap deficiencies at the top and bottom areas of the model—caused by camera angle limitations—the overall form of the object was successfully reconstructed.

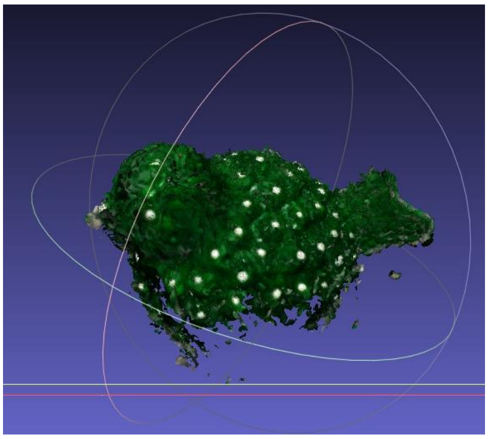

Figure 4-7 Meshing Process

4.3. System Performance Analysis

In practical applications, the system demonstrated stable and consistent performance. Particularly in environments supported by a white, uniform background, the boundaries of the object within the camera's field of view were accurately detected by the software, which positively influenced point cloud generation. Under adequate lighting conditions, surface details were more clearly extracted, and shadowing was minimized.

In this implementation, a scan performed with 300 medium-resolution photos successfully preserved structural integrity in both the sparse and dense point clouds, and the modeling process was completed without issues. However, a higher number of images taken from more diverse angles could have resulted in an even higher-quality model. Similarly, if more uniform lighting and studio-grade environmental conditions had been provided, surface deficiencies in the upper and lower regions of the model could have been further minimized.

In conclusion, the system operates effectively under suitable environmental conditions and has the potential to produce significantly higher-quality outputs when supported by improved imaging setups and a larger dataset.

Table 2 Effect of Lighting

Condition	Point Density	Mesh Quality	Observations
No lighting, colored background	Low	Poor	High noise, distorted edges
With lighting, white background	High	Medium-Good	Clear boundaries, smooth surface
Extra large number of photos	Very High	Good	Detailed, smoother mesh

5. Discussions

This section evaluates the performance of the developed photogrammetry-based 3D scanning system within an engineering framework. The system is analyzed separately in terms of mechanical stability, electronic control reliability, software workflow, and output quality. The results obtained during implementation are compared with the initial design objectives, identifying both the strengths of the system and the areas that require improvement. In addition, technical limitations encountered under real-world operating conditions, their causes, and potential solutions are discussed in detail.

5.1. Achieved Goals

The project successfully accomplished its goal of designing a low-cost, modular, and automation-ready photogrammetry system. Both motor control and image capture were executed in synchronization via the Raspberry Pi microcontroller, and the system operated consistently in accordance with the defined algorithm. Thanks to the proper dimensioning of mechanical components (gear system, camera carrier), the turntable exhibited stable movement, minimizing image distortions caused by vibrations.

The captured image data was processed with VisualSFM, resulting in the generation of a sparse point cloud, followed by a dense point cloud, and finally a mesh model. These stages demonstrated that the system is capable of successfully reconstructing 3D models under specific conditions, confirming the practical feasibility of the theoretical design. Moreover, the automatic data transfer and workflow streamlined the user experience by minimizing manual intervention.

5.2. Issues Encountered and Solutions

Throughout system testing, several factors affecting output quality were examined, including the number of photos, background uniformity, and lighting level. These variables were found to have a direct impact on model accuracy and point cloud density.

5.2.1. Comparison by Lighting Conditions

In each test scenario, the same object (duck model) was used, and the system was evaluated under different lighting conditions. The camera was positioned at 10 horizontal and 20 vertical angles, capturing a total of 200 images per test. The images were processed in VisualSFM to generate dense point clouds. In each case, the number, continuity, and density of the reconstructed objects were analyzed.

- **Very Low Lighting:** Due to insufficient lighting, images suffered from excessive shadowing and a loss of contrast. The software was able to detect only 12 distinct objects, which appeared as disconnected and fragmented structures. None of the objects were fully reconstructed. Dense point cloud count: 3,830 This value indicates a level that is insufficient for effective 3D modeling. See Figure 5.1 for the visual result.

Figure 5-1 Objects and Point Cloud Count Under Very Low Lighting

- **Low Lighting:** Under these conditions, the system was able to partially reconstruct 4 objects. Among them, one appeared relatively more complete, while the others exhibited significant structural gaps. Although image clarity improved compared to the very low lighting scenario, light distribution remained uneven. Dense point cloud count: 62,473 While the overall quality improved compared to the previous case, model detail remained limited. This outcome is illustrated in Figure 5.2.

Figure 5-2 Objects and Point Cloud Count Under Low Lighting

- **Medium-Level Lighting:** This lighting condition provided the most efficient environment for the system. The captured images were clear, well-contrasted, and largely free of shadows. The software successfully reconstructed a single, complete duck model without any fragmentation or missing parts. Dense point cloud count: 215,382. This level yielded optimal results in terms of both model integrity and point cloud density. The outcome is shown in Figure 5.3.

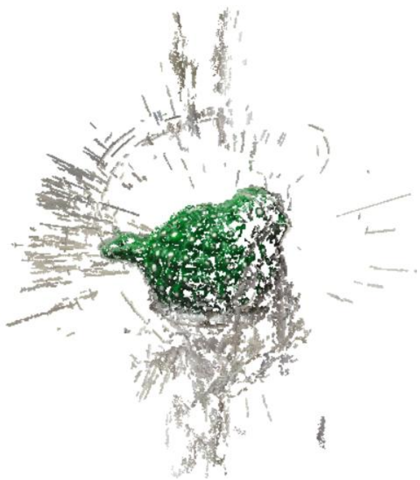

Figure 5-3 Objects and Point Cloud Count Under Medium Lighting

- **High Lighting:** Under high lighting conditions, the system detected two separate objects. One of them successfully represented a complete duck model, while the other exhibited a partially incomplete structure. In areas with excessive brightness, noticeable loss of detail was observed due to overexposure.

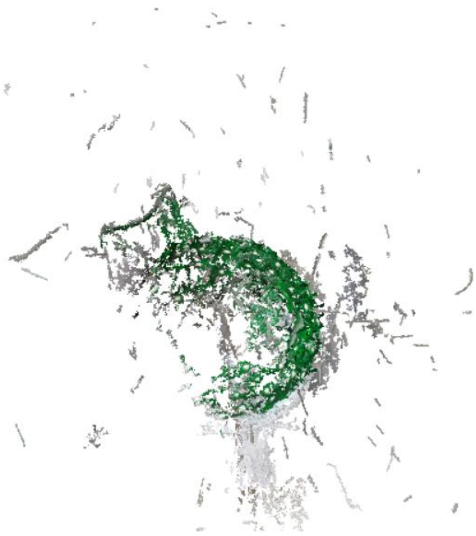

Figure 5-4 Objects and Point Cloud Count Under High Lighting

As shown in Figure 5.4, the dense point cloud count was 162,543. This value is lower than that of the medium lighting condition, and overexposure was observed in certain areas due to excessive brightness.

- **Very High Lighting:** In this scenario, the system identified 10 separate fragments (objects); however, none were successfully or completely modeled. Due to intense light reflections and blown-out white areas, the software struggled to distinguish surface details, leading to significant degradation in model integrity. Dense point cloud count: 41,825. This result demonstrates that the system loses stability under very high lighting conditions. The outcome is presented in Figure 5.5.

Figure 5-5 Objects and Point Cloud Count Under Very High Lighting

Lighting level is a critical parameter that directly affects both the integrity of the model and the density of the point cloud in photogrammetry-based modeling systems.
Test results indicate that medium lighting conditions yield the most efficient outcomes in terms of model accuracy and data density. In contrast, very low and very high lighting levels led to model fragmentation and prevented the algorithms from making accurate matches, thereby reducing the overall quality of the output. The dense point cloud results under varying lighting conditions are illustrated in Figure 5.6.

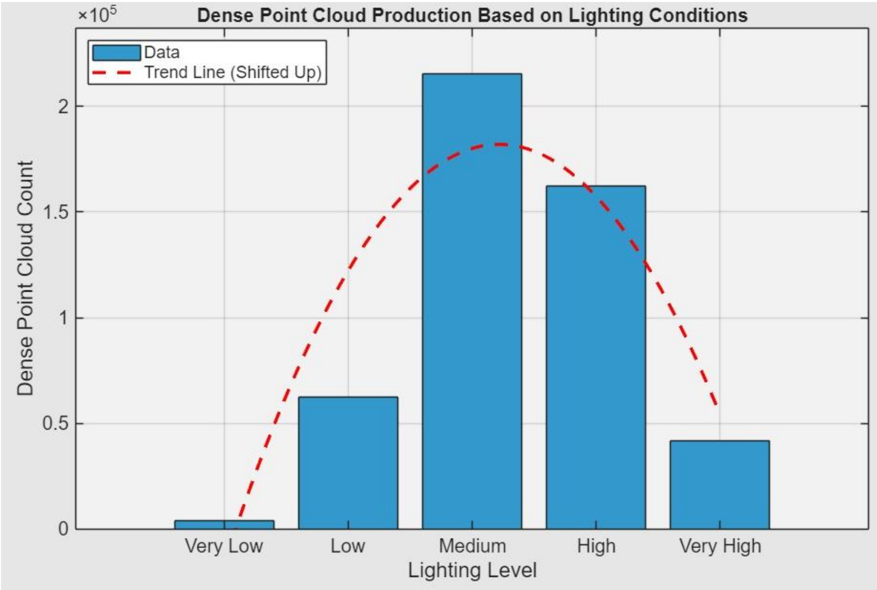

Figure 5-6 Dense Point Cloud Generation Under Different Lighting Conditions

5.2.2. Comparison by Number of Photos

- In the scan performed with **25 photos**, a total of 30,300 dense point cloud points were generated. The system reconstructed 3 objects, but all were incomplete and fragmented. This result is shown in Figure 5.7.

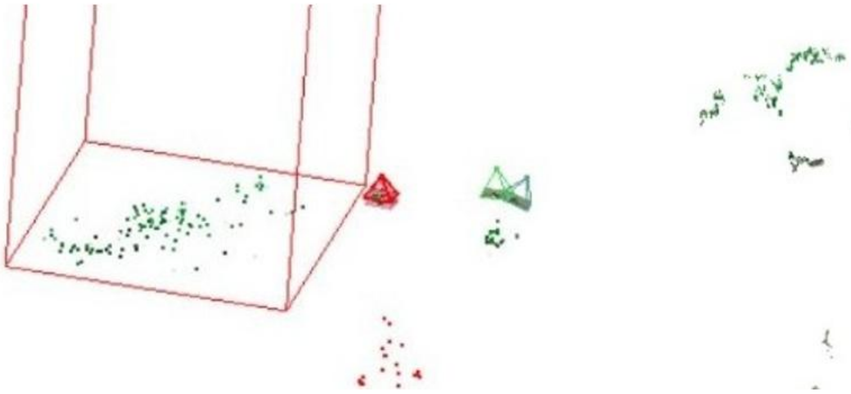

Figure 5-7 3D Reconstruction, Dense, and Point Cloud Count with 25 Photos

- With **50 photos**, 45,128 dense point cloud points were generated. Again, 3 objects were detected, but model integrity was not achieved. This result is shown in Figure 5.8.

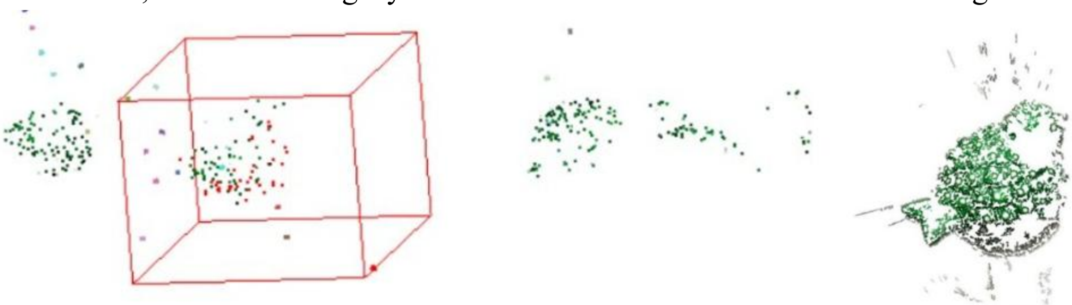

Figure 5-8 3D Reconstruction, Dense, and Point Cloud Count with 50 Photos

- In the process performed with **100 photos**, 77,631 dense point cloud points were generated. This time, a single object was produced; its overall form was clear, although some gaps remained. The result is shown in Figure 5.9.

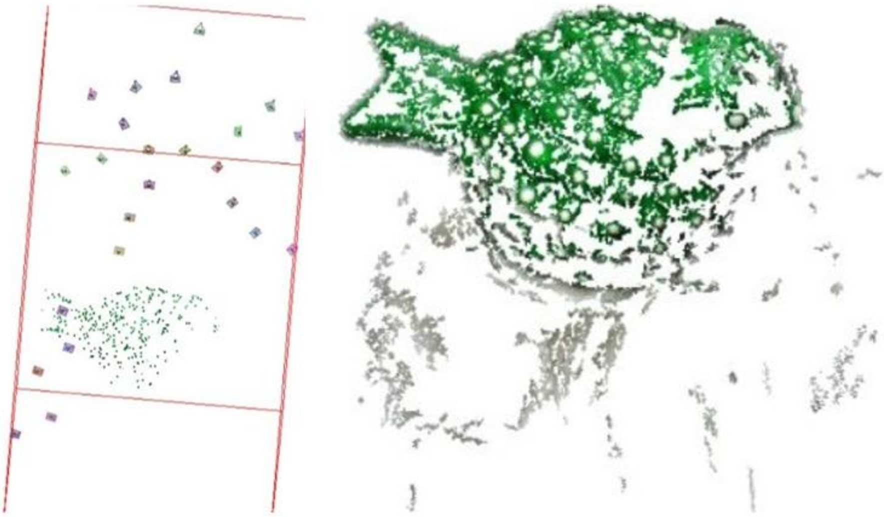

Figure 5-9 3D Reconstruction, Dense, and Point Cloud Count with 100 Photos

- With **200 photos**, a dense point cloud consisting of 233,376 points was obtained. A single, cohesive model was generated, resulting in a satisfactory outcome. This is illustrated in Figure 5.10.

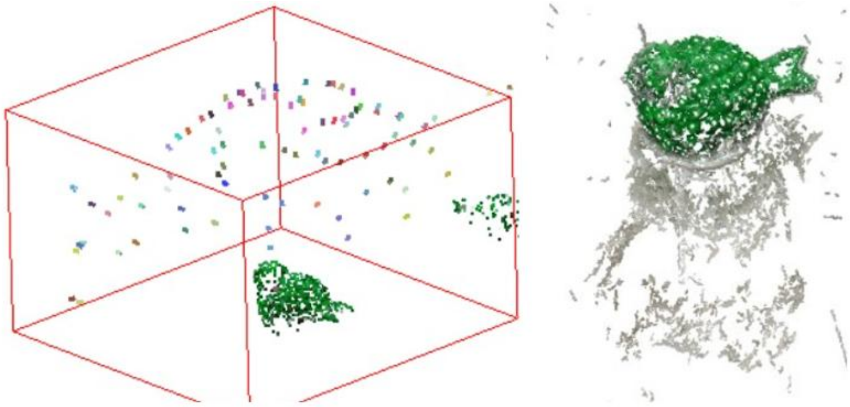

Figure 5-10 3D Reconstruction, Dense, and Point Cloud Count with 200 Photos

- In the scan performed with **300 photos**, a dense point cloud of 709,285 points was obtained. The resulting object exhibited a more detailed and refined structure compared to the model generated with 200 photos. This outcome is shown in Figure 5.11.

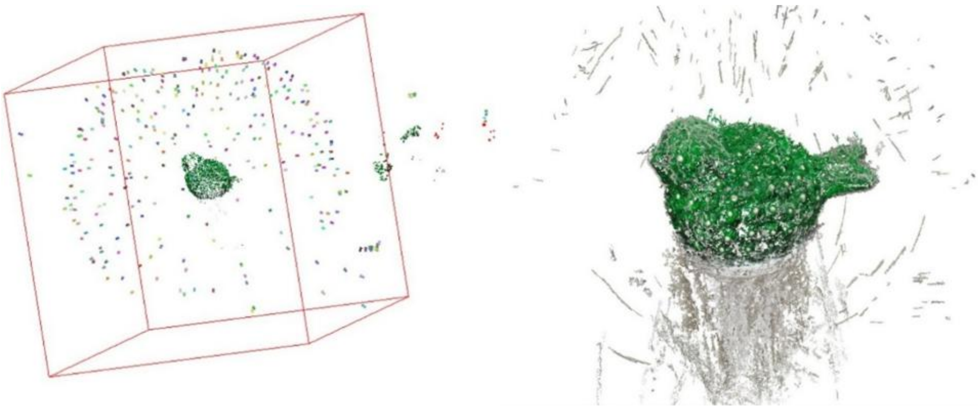

Figure 5-11 3D Reconstruction, Dense, and Point Cloud Count with 300 Photos

In brief, as the number of photos increases, both the amount of dense point cloud data and the overall model quality generally improve. Figure 5.12 illustrates the relationship between the number of photos taken and the resulting dense point cloud density.

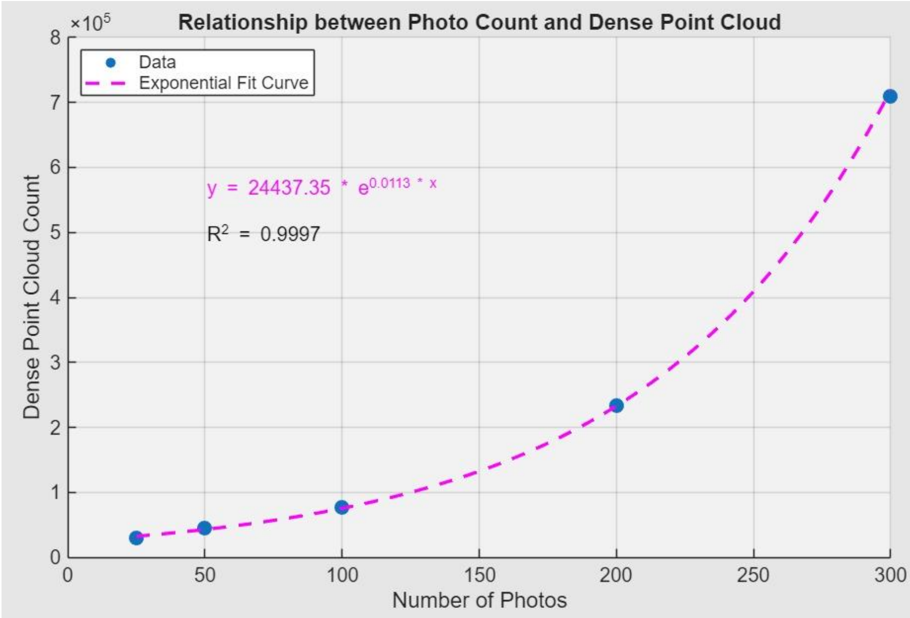

Figure 5-12 Dense Point Cloud Generation Based on Different Numbers of Photos Taken

5.2.3. Comparison Based on Background,

In photogrammetry-based 3D modeling, the number of dense point cloud points does not always fully reflect the quality of the model. The color and uniformity of the background play a critical role in the software's ability to accurately separate the object from the scene.

Multicolored or complex backgrounds can be interpreted as noise by the software, causing the model's edges to become blurred or fragmented. Therefore, having a completely uniform and single-colored background is essential to maintain the integrity of the model, even if the data density is high.

Below are results obtained under different background conditions with the same number of photos:

- With a **completely white background** and 200 photos, a single, cohesive object was created. A total of 162,543 dense point cloud points were obtained, and the model's geometry was largely reconstructed accurately. This is shown in Figure 5.13.

Figure 5-13 Dense and Point Cloud Count with White Background

- Using a **half-white, half-brown background,** 200 photos were captured again. A single object was generated; however, distortions appeared in the edge details. The dense point cloud consisted of 146,098 points. This result is illustrated in Figure 5.14.

Figure 5-14 3D Reconstruction and Point Cloud Count with Half White – Half Brown Background

- When using a **typical environmental background** (such as a table, wall, etc.) with 200 photos, the modeling process failed. A single, complete object was not formed; instead, fragmented detections appeared in the scene, as shown in Figure 5.15. Although 132,871 dense point cloud points were generated, the model's structural integrity was not preserved.

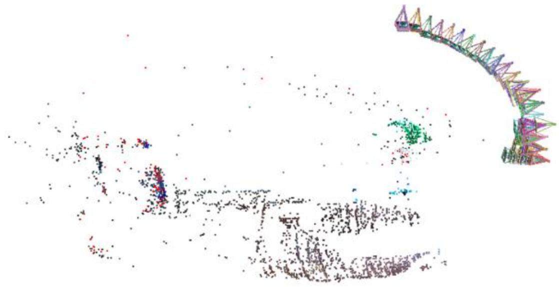

Figure 5-15 3D Reconstruction with Normal Environmental Background

5.2.4. Comparison Based on the Geometric Shape of the Turntable

During photogrammetric scanning, the stable rotation of the turntable is critically important for the clarity and consistency of the images. If the turntable wobbles during rotation, misalignment occurs between the captured photos, which directly degrades the quality of the resulting point cloud. Therefore, the mechanical stability of the components used in the system is one of the fundamental factors determining scan quality.

- **Wobbling Turntable:** The first turntable used in the project, shown in Figure 5.16, was mechanically unstable and exhibited wobbling during rotation. This caused inconsistencies in the angular positions of the photos, negatively affecting the overall imaging process.

Figure 5-16 Initial Turntable (Prone to Wobbling)

As a result, alignment errors occurred during the SfM stage, producing a distorted, scattered, and low-accuracy point cloud. Notably, clustering and gaps were prominent in the edge regions. Figure 5.17 illustrates the dense point cloud of the object placed on the wobbling turntable.

Figure 5-17 Object and Dense Point Cloud Formation of the Model Captured on a Wobbling Turntable

- **Redesigned Balanced Turntable:** Following issues encountered in the initial trials, the turntable was redesigned to be more rigid and balanced. The new structure rotated smoothly around a fixed axis without vibrations, keeping the object consistently centered. As a result, the captured images were consistent, and the point cloud generated was well-formed, gap-free, and highly accurate. The redesigned balanced turntable is shown in Figure 5.18.

Figure 5-18 Redesigned Balanced Turntable

5.2.5. Effect of Motor Speed and Microstepping in the Mechanism

In the system, the large ring gear mechanism on the vertical axis is driven by a stepper motor, enabling the camera's upward and downward movement. This motor is controlled using the microstepping method, and the system's behavior has been analyzed at different speed levels to assess performance.

Table 3 Micro Steps

MODE 0	MODE 1	MODE 2	Micro steps
LOW	LOW	LOW	**Full**
HİGH	LOW	LOW	**Half**
LOW	HİGH	LOW	**1/4**
HİGH	HİGH	LOW	**1/8**
LOW	LOW	HİGH	**1/16**
HİGH	LOW	HİGH	**1/32**
LOW	HİGH	HİGH	**1/32**
HİGH	HİGH	HİGH	**1/32**

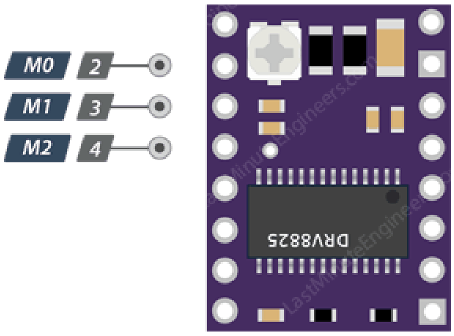

Figure 5-19 Microstepping Pins [14]

Experimental observations revealed the following key findings:

- **At very low speeds**, the motor caused significant vibrations in the camera carrier's movement. Due to the large diameter of the ring gear, this vibration propagated throughout the system, reducing image stability during capture.
- **At moderate speeds**, the mechanical system operated more steadily, providing smoother motion for both the motor and gear mechanism. This improved image sharpness and enhanced the quality of the dense point cloud outputs. Operating away from the system's resonance frequencies minimized vibrations.
- **At high speeds**, noticeable vibration increased again. The inertia of the large ring gear combined with rapid movement led to loss of balance, causing blurring and alignment issues in some images. Moreover, torque loss in the stepper motor increased the risk of missed steps.

Based on these results, careful selection of the motor speed range on the X-axis's large ring gear is critical for both mechanical stability and photogrammetric output quality. While microstepping improves motion precision, speeds chosen without considering the system's physical dynamics can degrade data quality.

5.3. System Limitations

Although the developed photogrammetry-based 3D scanning system successfully meets certain objectives, it has several technical and operational limitations. These constraints, stemming from both hardware and software aspects, significantly affect overall system performance:

- **Hardware Limitations:** Although the image capture process is automated by the Raspberry Pi, the computationally intensive image processing stages (SfM and MVS) must be run on an external computer. This divides the workflow across two separate hardware platforms, detracting from a fully embedded and portable solution. The Raspberry Pi Camera V2 used is fixed-focus, lacking adjustable focal length. This results in focus loss when scanning objects of varying sizes and depths, potentially causing small details to be missed. Additionally, the use of a protoboard (perforated phenolic board) with manually wired connections prolongs setup time and reduces system robustness.
- **Mechanical Limitations:** The mechanical design imposes certain geometric constraints. The camera carrier's range limits the system to scanning only small- and medium-sized objects. To accommodate larger objects, the system's physical dimensions or degrees of freedom must be increased.

- **Environmental Limitations:** The system is sensitive to environmental conditions. Uneven lighting, complex backgrounds, and external vibrations directly affect model quality. It performs best in controlled indoor environments; output quality may degrade significantly in natural or uncontrolled settings. Finally, some missing surfaces and distortions have been observed in the post-processing models.

6. Conclusion and Recommendations

This section provides an overall evaluation of the developed photogrammetry-based 3D scanning system and presents suggestions for future improvements based on the obtained results. The conducted applications have demonstrated that the system successfully fulfills its core functions; however, certain technical enhancements are necessary to achieve higher accuracy and flexibility.

6.1. General Conclusions

In this study, a low-cost, open-source, and semi-automatic photogrammetry-based 3D scanning system was successfully designed and implemented. Using a Raspberry Pi-controlled platform, automatic photography from different angles was performed with the aid of stepper motors and a camera. The captured images were processed through VisualSFM software to sequentially generate a sparse point cloud, dense point cloud, and finally a 3D mesh model.

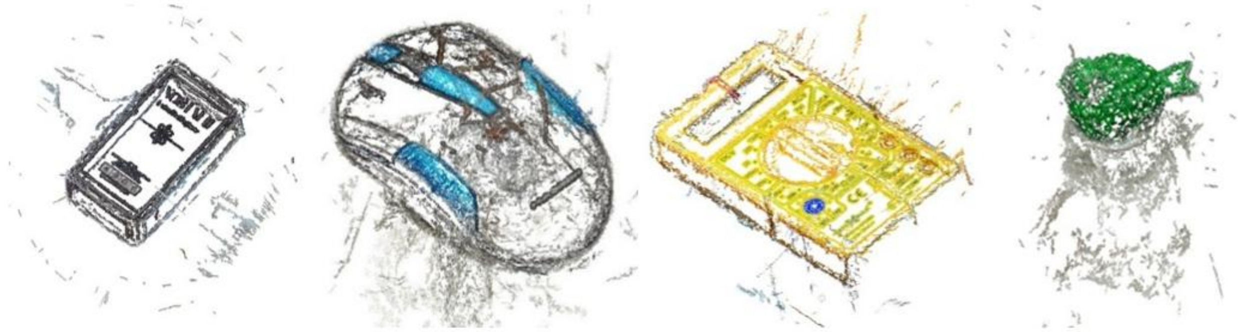

Figure 6-1 Scanning Examples

The experiments demonstrated that medium-level lighting and a uniform (white) background are critical factors influencing model quality. Additionally, as the number of photos increases, both the detail level and surface accuracy of the model improve significantly. Scans with more than 200 high-quality images successfully produced a single, coherent object with effective photogrammetric matching. Additional examples are shown in Figure 6.1.

Technically, the system operates stably, with mechanical structure, electronic control, and software components integrated harmoniously. However, hardware limitations and sensitivity to environmental factors remain key influences on model quality.

6.2. Recommendations for Future Work

To enhance system stability, compactness, and intelligence in the future, several improvement suggestions are proposed under four main categories:

6.2.1. Mechanical Improvements

- **New Object Fixation Method:** Currently, objects are directly adhered to the turntable. An alternative would be to design a clamp or support system that securely fixes and isolates the object without leaving marks.
- **Turntable Expansion:** Increasing the range of motion for the camera carrier system would enable scanning of larger objects.

- **Enclosed Indoor System Assembly:** Designing a more compact, ergonomic, and enclosed housing for the electronic circuit board, motors, and wiring would improve portability and safety.

6.2.2. Electrical Improvements

- **Integration of Autofocus Camera:** Replacing the Raspberry Pi camera with an autofocus system would allow sharper images of objects with varying sizes.
- **SMD Design:** Developing an SMD (Surface-Mount Device) type circuit board design would reduce board size, facilitate automated production, and enhance mechanical durability.

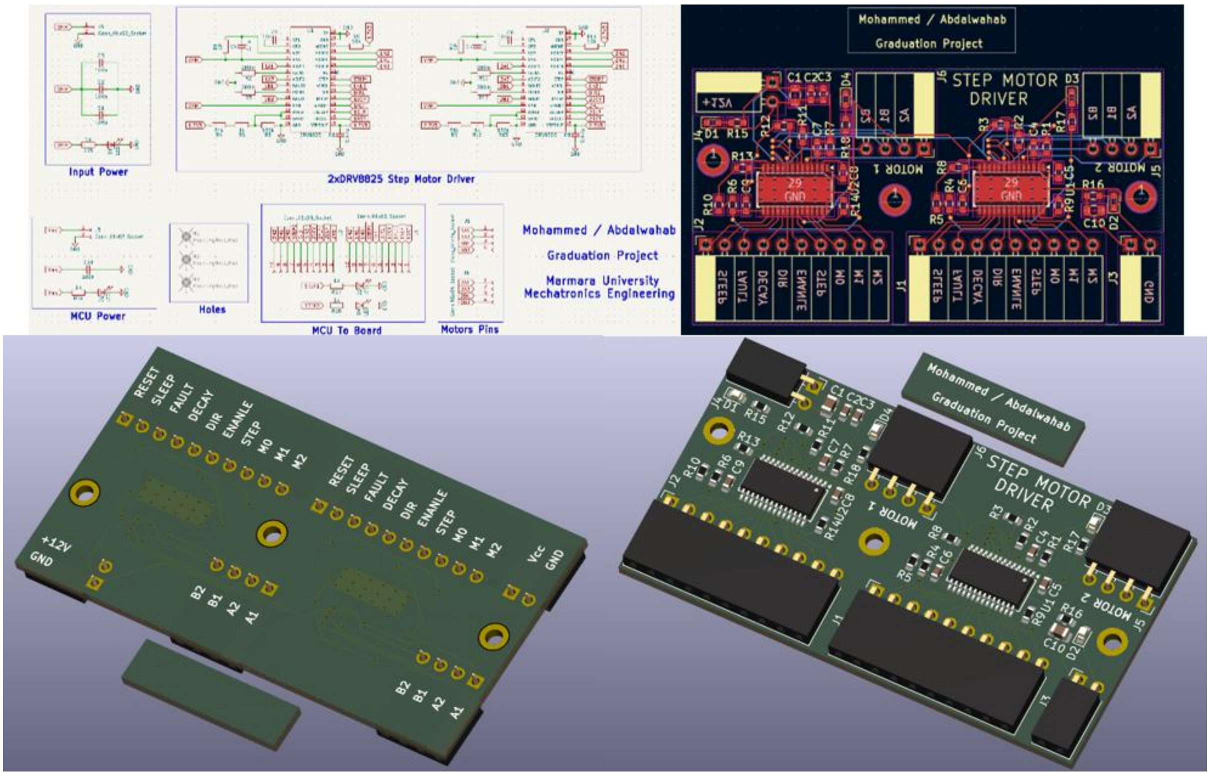

Figure 6-2 PCB Circuit Design, Layering, and 3D Modeling for SMD Design

- **Integrated Power Management:** An integrated power module should be added. Voltage regulation and protection circuits embedded on the board can reduce cable clutter and minimize the risk of failures.
- **LCD Screen Usage:** An LCD screen should be incorporated for the user interface. This display would allow users to monitor system status and visually track settings.
- **Input Control Interface:** A user control unit for entering the number of photos should be added. Using a joystick, buttons, or a touchscreen, users could easily adjust the number of horizontal and vertical shots.
- **Product-Oriented Design:** The board design should be suitable for productization. All electronic components should be housed within a single enclosure to enhance system ergonomics and portability.

6.2.3. Software and Automation Improvements

- **Embedded Image Processing:** Shifting image processing tasks from the Raspberry Pi to a more powerful embedded system (e.g., NVIDIA Jetson) would accelerate processing and increase system autonomy.
- **Cloud-Based Automation:** Performing all data transfer and processing in the cloud would enable remote monitoring and fully automate the modeling workflow.
- **AI-Driven Image Selection and Background Filtering:** AI-powered algorithms could automatically discard blurry or unusable images. Additionally, complex backgrounds could be algorithmically interpreted as uniform white backgrounds, improving modeling accuracy.

7. References

[1]. Sürmen HK., 2022, Fotogrametri ile Üç Boyutlu Tarama – 3B Fotogrametri, Mühendislik Alanında Uluslararası Araştırmalar - I, Serüven Yayınevi, ISBN: 978-625-6399-08-2.

[2]. Remondino F, El-Hakim S., 2006, Image-based 3D modelling: a review, The Photogrammetric Record, 21(115):269–291.

[3]. Wu C., 2011, VisualSFM: A Visual Structure from Motion System, http://ccwu.me/vsfm/

[4]. AliceVision, 2024, Meshroom: Photogrammetric Computer Vision Framework, https://alicevision.org/#meshroom

[5]. Regard3D, 2024, Open Source Structure-from-Motion Program, http://www.regard3d.org

[6]. Luhmann T, Robson S, Kyle S, Boehm J., 2013, Close-Range Photogrammetry and 3D Imaging, De Gruyter.

[7]. Cignoni P, Callieri M, Corsini M, Dellepiane M, Ganovelli F, Ranzuglia G., 2008, MeshLab: an open-source mesh processing tool, Eurographics Italian Chapter Conference, 129–136.

[8]. MeshLab, 2024, 3D Mesh Processing Software, https://www.meshlab.net

[9]. OpenScan, 2024, Open-source 3D scanning solutions, https://openscan.eu

[10]. Hartley R, Zisserman A., 2004, Multiple View Geometry in Computer Vision (2nd ed.), Cambridge University Press.

[11]. Westoby MJ, Brasington J, Glasser NF, Hambrey MJ, Reynolds JM., 2012, Structure-from-Motion photogrammetry: A low-cost, effective tool for geoscience applications, Geomorphology, 179:300–314.

[12]. Schönberger JL, Frahm JM., 2016, Structure-from-Motion Revisited, Proc. CVPR, https://colmap.github.io

[13]. Blender Foundation, 2024, Blender: Free and Open Source 3D Creation Suite, https://www.blender.org

[14]. Instructables, 2023, Open 3D Scanner: DIY Photogrammetry Rig, https://www.instructables.com/Open-3D-Scanner/

[15]. Solomon C, Breckon T., 2010, Fundamentals of Digital Image Processing: A Practical Approach with Examples in Matlab, Wiley-Blackwell.

[16]. Raspberry Pi Foundation, 2024, Raspberry Pi 4 Documentation, https://www.raspberrypi.com/documentation

[17]. Gonzalez RC, Woods RE., 2008, Digital Image Processing (3rd ed.), Pearson Prentice Hall.

[18]. Pololu Robotics, 2024, DRV8825 Stepper Motor Driver Carrier, https://www.pololu.com/product/2133

8. Appendices

Appendix 1: Mechanical Design (CAD + Prototype)

This appendix presents the CAD drawings and physical prototype images of all mechanical components of the system. The main body, gear mechanisms, turntable, camera stand, and slider components collectively enable the two-axis movement necessary for photogrammetry. All parts have been designed and tested to operate with precision, stability, and repeatability.

Figure 8-1 Mechanical Components

Appendix 2: PCB Design and Prototype Images

This appendix presents the electronic infrastructure of the system, including the circuit schematic, PCB layout, 3D renderings, and photos of the prototype test board. The designs were created using KiCad software, with component placement and trace routing optimized to ensure stable system operation. Initial testing was conducted on a soldered perfboard (protoboard), followed by a professional transition to a custom-designed PCB.

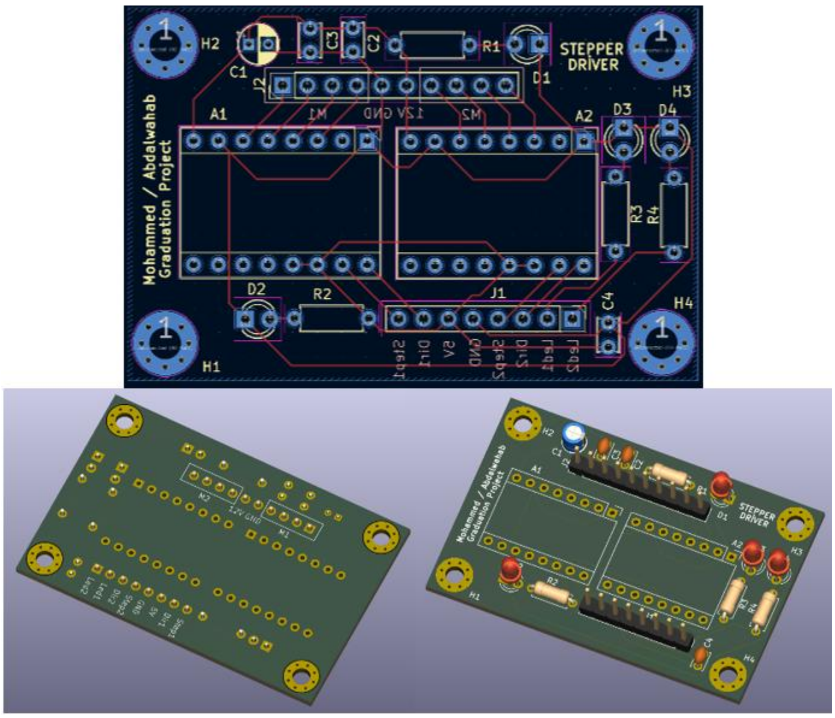

Figure 8-2 PCB Layer and 3D Model of the Designed Circuit in the Electronics Section

Appendix 3: Software System (Python Automation)

This appendix presents the automation code running on the Raspberry Pi. The software controls motor movements, captures photos with the camera, saves images, and automatically transfers them to a computer. The code is organized modularly, including initialization, functions, main loop, and error handling. The system operates in a fully automated manner.

```python
import RPi.GPIO as GPIO
import time
import os
from picamera2 import Picamera2
from datetime import datetime
import subprocess
import sys
STEP1, DIR1 = 17, 18
STEP2, DIR2 = 23, 24
LED1, LED2 = 22, 27
DELAY_US = 1000 # Delay between motor steps in microseconds
WAIT_TIME = 1   # Delay between 2 motor movement and camera capture
camera = Picamera2()
camera.configure(camera.create_still_configuration(main={"size": (4056, 3040)}))
camera.start()
GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)
for pin in [STEP1, DIR1, STEP2, DIR2, LED1, LED2]:
    GPIO.setup(pin, GPIO.OUT)
GPIO.output(LED1, GPIO.LOW)
GPIO.output(LED2, GPIO.LOW)

def wait():
    time.sleep(WAIT_TIME)

def move_motor(step_pin, dir_pin, enable_led_on, enable_led_off, steps, direction):
    GPIO.output(dir_pin, GPIO.HIGH if direction == 'forward' else GPIO.LOW)
    GPIO.output(enable_led_on, GPIO.HIGH)
    GPIO.output(enable_led_off, GPIO.LOW)
    for _ in range(steps):
        GPIO.output(step_pin, GPIO.HIGH)
        time.sleep(DELAY_US / 1_000_000)
        GPIO.output(step_pin, GPIO.LOW)
        time.sleep(DELAY_US / 1_000_000)
    GPIO.output(enable_led_on, GPIO.LOW)

def capture_photo(path):
    camera.capture_file(path)
    print(f"Saved: {path}")

def transfer_photos(photo_dir, targets):
    for pc_ip, pc_user in targets:
        target_path = f"{pc_user}@{pc_ip}:~/raspberry_photos"
        scp_command = ["scp", "-r", photo_dir, target_path]
        print(f"Transferring photos to {pc_ip} with: {' '.join(scp_command)}")
        subprocess.run(scp_command)

def main():
    horizontal = int(input("Enter number of horizontal images per level: "))
    vertical = int(input("Enter number of vertical levels: "))

    photo_dir = f"/home/pi/Desktop/photos/{datetime.now().strftime('%Y-%m-%d_%H-%M-%S')}
    os.makedirs(photo_dir, exist_ok=True)

    motor1_steps_per_rotation = 200 # Steps required for motor 1 to make one full turn
    motor2_steps_total = 1200 # Steps required for motor 2 to make one full turn
    motor1_steps = motor1_steps_per_rotation // horizontal
    motor2_steps = motor2_steps_total // vertical

    for v in range(1, vertical + 1):
        move_motor(STEP2, DIR2, LED2, LED1, motor2_steps, 'forward')
        wait()

        for h in range(1, horizontal + 1):
            filename = os.path.join(photo_dir, f"v{v}_h{h}.jpg")
            capture_photo(filename)
            wait()
            if h != horizontal:
                move_motor(STEP1, DIR1, LED1, LED2, motor1_steps, 'forward')
                wait()

        move_motor(STEP1, DIR1, LED1, LED2, motor1_steps * (horizontal - 1), 'backward')

    move_motor(STEP2, DIR2, LED2, LED1, motor2_steps * vertical, 'backward')

    camera.stop()
    GPIO.cleanup()

    targets = [
        ("IP_ADDRESS_1", "User_Name_1"),
        ("IP_ADDRESS_2", "User_Name_2")
    ]
    transfer_photos(photo_dir, targets)
    print("Done! Photos transferred to all targets.")

try:
    main()
except KeyboardInterrupt:
    print("Interrupted by user.")
    camera.stop()
    GPIO.cleanup()
    sys.exit(0)
except Exception as e:
    print(f"An error occurred: {e}")
    camera.stop()
    GPIO.cleanup()
    sys.exit(1)
```

Figure 8-3 Raspberry Pi Code (Python)

CVs

PERSONAL INFORMATION

Full Name	: Abdelwahab Rayis
Year of Birth	: 2002
Native Language	: Arabic
Foreign Language	: English, Turkish
Email	: rayis2021@gmail.com

EDUCATIONAL BACKGROUND

Degree	Department/Program	University	Mezuniyet Yılı
Bachelor	Mechatronics Engineering	Marmara University	2025

PERSONAL INFORMATION

Full Name	: Muhammed Şeyhoğlu
Year of Birth	: 2003
Native Language	: Arabic
Foreign Language	: English, Turkish
Email	: muhammet.elsik@gmail.com

EDUCATIONAL BACKGROUND

Degree	Department/Program	University	Mezuniyet Yılı
Bachelor	Mechatronics Engineering	Marmara University	2025