ECE 445

SENIOR DESIGN LABORATORY

FINAL REPORT

3D Scanner

CHENCHEN YU
(cy32@illinois.edu)
JIAYI LUO
(jiayi13@illinois.edu)
PEIYUAN LIU
(peiyuan6@illinois.edu)
YIFEI SONG
(yifeis7@illinois.edu)

Instructor: Pavel Loskot

TA: Xinyi Xu

May 23, 2023

Acknowledgement

We would like to express our profound gratitude to all those who have made significant contributions to the success of this project.

First and foremost, our heartfelt thanks go to ZJUI for their generous sponsorship and steadfast support throughout the entire project. The financial backing they provided has been the bedrock upon which our aspirations and objectives were realized.

Our deepest appreciation is extended to our project advisor, Pavel Loskot, for his insightful guidance and expertise. His knowledgeable suggestions and encouraging spirit have been pivotal in the formation of our project and in pushing us towards excellence.

We would also like to express our gratitude to the course instructors and teaching assistant whose tireless dedication has left an indelible mark on our project. Their weekly discussions and insightful recommendations have greatly shaped the progress and quality of the project.

Regarding our team members, we want to acknowledge their significant efforts in this venture. Our mutual appreciation is directed towards the one who took charge of the mechanical design and manufacturing processes, the one who excelled in the control system design and implementation, the one who contributed to the wireless transmission module and the one who demonstrated significant prowess in the development and application of the 3D reconstruction algorithm. The commitment and expertise each one has brought to the table have been instrumental to the success of our project.

Finally, we extend our collective thanks to all individuals who have provided assistance, guidance, and moral support throughout this project. Their contributions, both technical and moral, have been a cornerstone in the accomplishment of our endeavor.

Thank you all for your unwavering faith in our potential and the support you've extended throughout the project.

Abstract

This study introduces the design and evaluation of an innovative 3D scanning apparatus, proficient in generating high-definition 3D point clouds from a collection of 2D images. The scanning apparatus utilizes a custom-engineered mechanical mechanism that maneuvers a smartphone around a given object, thereby securing a multitude of images from varied viewpoints. Subsequently, these images are wirelessly relayed to a computerized system. The latter utilizes a structure-from-motion technique for 3D reconstruction, which in turn yields a precise and intricate 3D point cloud of the object under inspection. System evaluations carried out on a range of objects demonstrated successful reconstructions, maintaining an average reprojection error of less than one pixel. This lends credence to the system's effectiveness and dependability. Further visual comparison substantiates the resemblance between the reconstructed models and their original counterparts. However, it is observed that the system's performance can be compromised under inadequate lighting conditions or when dealing with objects bearing a monochromatic surface. Future endeavors will concentrate on addressing these constraints to bolster the system's versatility and practicality.

Keywords: 3D scanning apparatus, point clouds, structure-from-motion, image-assisted 3D reconstruction, reprojection error.

Contents

1 Introduction

3D scanning technology, a pioneering instrument capable of producing high-accuracy digital 3D models, finds applications across a diverse range of industries, from manufacturing to healthcare [1]–[5]. Nevertheless, the implementation of conventional 3D scanning machinery is frequently deterred by its significant size, restricted mobility, lengthy processing periods, and operational intricacy.

1.1 Problem Statement

The widespread presence and advanced functionalities of contemporary mobile devices offer the potential to incorporate 3D scanning technology into these portable instruments. This integration could democratize 3D scanning technology, enabling more extensive utilization in daily life and professional environments [3]. However, the existing complexity and time-intensive characteristics of 3D scanning procedures present substantial hurdles to this anticipated integration. A compelling requirement exists for a solution that streamlines and expedites these procedures, making 3D scanning more efficient and user-friendly.

1.2 Project Overview

In response to these challenges, we have embarked on a project to develop an innovative system that leverages a mobile phone's camera and a mechanically controlled device. This system is designed to capture a series of 2D images from diverse angles, which are then used to construct high-quality digital 3D models. The solution is aimed at circumventing the limitations of traditional 3D scanning equipment, thereby expanding its accessibility and applicability, particularly for those without access to standard 3D scanning tools.

1.3 Motivation

Our motivation for this project is twofold. Firstly, we aspire to democratize access to 3D scanning technology, aiming to make it a readily accessible and practical instrument for a broader user base. Secondly, we are committed to expanding the utility of 3D scanning technology, promoting its integration into a plethora of fields and day-to-day applications. We are confident that by addressing the challenges intrinsic to traditional 3D

scanning, we can significantly contribute to extending the reach and impact of this revolutionary technology.

2 Literature Review

The landscape of 3D reconstruction methodologies has seen substantial research and development in recent years, with several techniques gaining noteworthy attention. Structure from Motion (SfM), a photogrammetric range imaging technique, estimates three-dimensional structures from two-dimensional image sequences. This method was initially explored by Ullman [6], who elucidated the potential of deriving 3D structure from sequences of images over time. Further enhancements, such as robust algorithms for large-scale reconstructions, have solidified SfM as a cornerstone in 3D reconstruction [7]–[9].

Simultaneously, Multi-View Stereo (MVS) has emerged as an essential technique in the field. By leveraging multiple images of a scene from diverse viewpoints, MVS recovers scene geometry. A comprehensive comparison of different MVS algorithms was presented by Seitz et al., shedding light on the strengths and weaknesses of each [10]. Subsequent research, including a patch-based method by Furukawa et al., has further enhanced MVS's precision and speed [11]. OpenMVS, an open-source library, encapsulates various MVS techniques, thereby providing a versatile tool for 3D reconstruction [12].

The photogrammetric computer vision framework Multi-view Geometry (MVG) has also made substantial strides in this arena. By incorporating SfM and MVS implementations, this open-source framework has encouraged contributions from researchers and developers worldwide, leading to more innovative solutions [13].

Lastly, the advent of Neural Radiance Fields (NeRF) represents a breakthrough in the field. This technique employs a fully connected deep network to model a continuous volumetric scene function, allowing for comprehensive scene analysis and interpretation. Mildenhall et al. demonstrated that NeRF could produce high-quality views of complex scenes from a sparse set of input images [14]. All these methods, each with their unique advantages, have found diverse applications in the broader realm of 3D reconstruction.

3 Methodology

This section presents a comprehensive overview of the proposed methodology for the 3D scanner system and outlines the different subsystems involved with their respective functionalities.

3.1 Block Diagram

The block diagram (Figure 1) illustrates the overall architecture of the 3D scanner system. It consists of five main subsystems: the Image Collection Subsystem, the Communication Subsystem, the Image Processing Subsystem, the Phone-holder Subsystem, and the Remote-control Subsystem. These subsystems work collaboratively to capture, transmit, process multiple 2D images to reconstruct high-quality 3D point clouds.

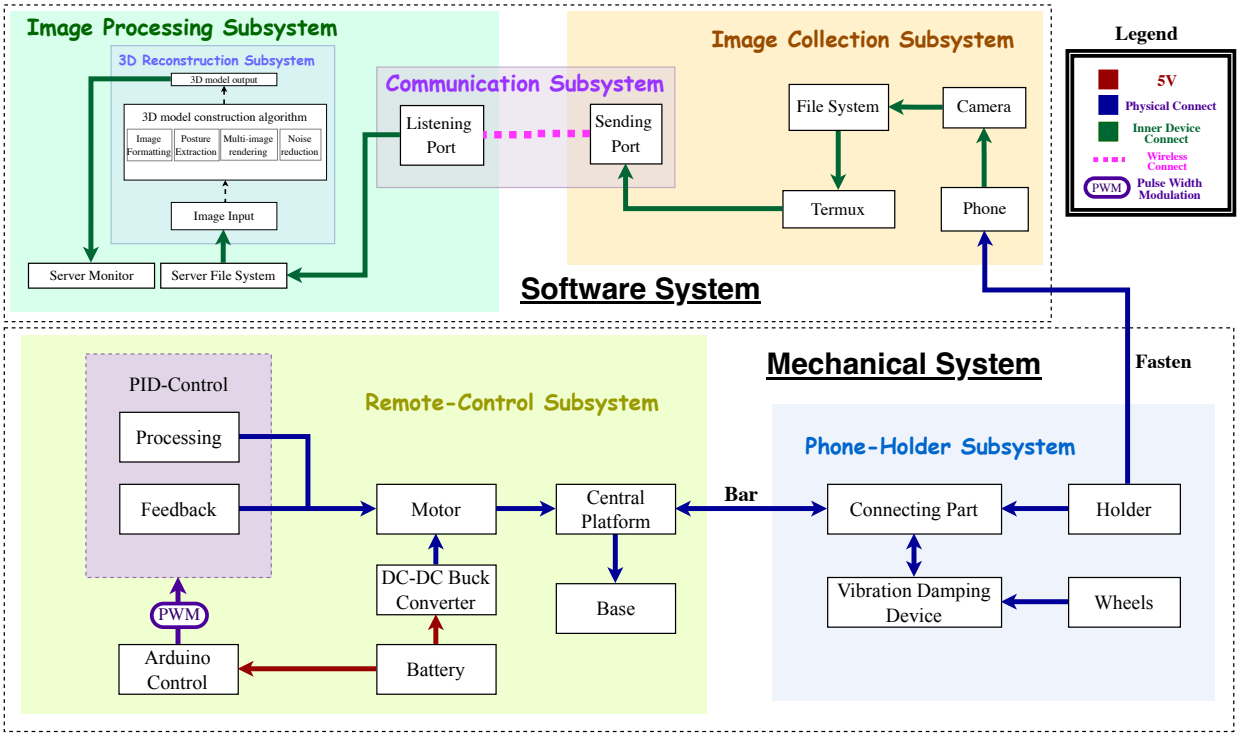


Figure 1: Block diagram of the 3D scanner system.

3.2 Image Collection Subsystem

3.2.1 Implementation of Remote Photo/Video Taking

This part mainly focuses on the implementation of the initial version of the design. We chose Python as the programming language for both the smartphone and computer due to the abundance of third-party libraries and frameworks available, as well as its high cross-platform compatibility. Since we decided to use an Android phone for the experiment, we conducted a search and investigation, and ultimately selected QPython and Termux platforms for mobile program development and testing. Once we finalized the programming language and development platforms, we proceeded to implement the functionality for specific message detection and handling. We provide pseudo-code for sock network programming in python, including sending data 1 and receiving data 2, respectively.

The design for remote transmission also involved optimizing the system for various devices and configurations. This included implementing error handling mechanisms, ensuring proper memory management, and improving the overall performance of the system.

3.2.2 Implementation of Stable Video Transmission

In order to enhance the speed and accuracy of video transmission, we implemented an IP camera application on the smartphone and utilized OpenCV [15] on the computer for video streaming. This system enabled the efficient transfer of video data, which was then saved on the computer in a designated location.

The implementation process began by setting up an IP camera on the smartphone, leveraging its built-in camera capabilities. This allowed the smartphone to function as a camera device that captures real-time video footage. We selected the IP camera application due to its ability to provide a reliable and secure connection for video streaming.

On the computer side, we utilized the OpenCV library to receive and process the video frames transmitted by the IP camera. OpenCV, a widely-used computer vision library, offers a comprehensive set of functions for video manipulation, analysis, and storage. By leveraging its features, we were able to efficiently handle the incoming video frames and perform any necessary processing tasks.

To ensure smooth and seamless video transmission, we established a network connection

between the smartphone and the computer. The IP camera application on the smartphone continuously captured video frames, which were then sent over the network to the computer running the OpenCV program. The computer received these frames and stored them in a specified location for further analysis or archival purposes.

This approach not only facilitated faster video transmission but also improved the accuracy of the captured frames. By utilizing the capabilities of both the IP camera application and the OpenCV library, the system achieved real-time video streaming with minimal delay and ensured the preservation of video data in a reliable and organized manner.

3.3 Communication Subsystem

The subsequent design objective entailed establishing effective communication and synchronization between the image collection subsystem and the image processing subsystem. To achieve this objective, a strategic decision was made to employ TCP (Transmission Control Protocol) communication. Through this approach, the computer would transmit TCP packets containing precise messages at predetermined intervals to trigger the smartphone's image capture process and store the acquired images in a designated location. Additionally, the TCP protocol would be utilized to facilitate image transfer from the smartphone to the computer for subsequent 3D reconstruction.

In order to ensure seamless communication and synchronization, a meticulous message handling mechanism was devised for the smartphone component. This mechanism facilitated the detection of specific messages, prompting corresponding actions. For instance, upon receipt of a message containing the designated keyword "start", the server would initiate the image capture process. Conversely, upon receiving a message containing the keyword "quit", the server would terminate the TCP connection. This well-designed message handling mechanism served to enhance the efficiency and effectiveness of communication between the subsystems.

3.4 Image Processing Subsystem

Figure 2 outlines the principal stages of our 3D reconstruction framework, encompassing six key phases: feature extraction, feature matching, sparse reconstruction, dense reconstruction, background removal, and outliers removal. Section 3.4.1 elucidates the first four stages of this process, while section 3.4.2 presents the Density-Based Spatial Cluster-

ing of Applications with Noise (DBSCAN) algorithm, utilized for background removal. Lastly, section 3.4.3 introduces the Statistical Outlier Removal (SOR) algorithm, a key component employed in the outlier removal phase.

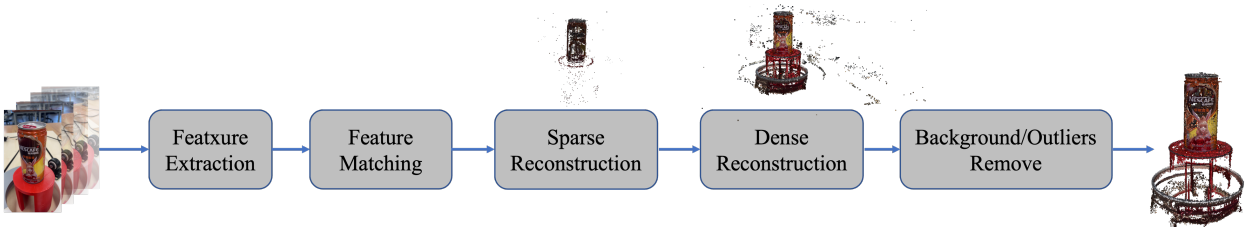


Figure 2: Schematic representation of the employed 3D reconstruction algorithm framework.

3.4.1 3D Reconstruction Algorithm

The core principle of the 3D reconstruction in this project is based on the COLMAP, a state-of-the-art tool for Structure-from-Motion (SfM) and Multi-View Stereo (MVS) algorithms [8]. The algorithm can be divided into two main stages: the Structure-from-Motion (SfM) stage and the Multi-View Stereo (MVS) stage.

In the SfM stage, features F_i are extracted for each image I_i using a scale-invariant feature transform (SIFT) algorithm [16]. The SIFT features are computed by convolving the image with a Gaussian kernel $G(x, y, \sigma)$ and taking the difference of Gaussians (DoG) to detect scale-space extrema:

$$D(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y), \tag{1}$$

where k is a constant factor and $*$ denotes the convolution operation.

Feature matching is then performed between image pairs (I_i, I_j) to identify correspondences using a similarity metric, such as the Euclidean distance between SIFT feature descriptors. Camera poses and a sparse point cloud are estimated using a perspective-n-point (PnP) algorithm [17] and bundle adjustment, which aims to minimize the reprojection error:

$$\min_{C_i, X_j} \sum_{i,j} w_{ij} \left| x_{ij} - \pi_{C_i}(X_j) \right|^2, \tag{2}$$

where C_i represents the camera pose, X_j are the 3D points, x_{ij} are the 2D image projections, π_{C_i} is the projection function, and w_{ij} are the weights for each correspondence [7]. The output of this stage is a sparse reconstruction $\mathcal{M}$.

The MVS stage initializes an empty dense point cloud $\mathcal{D}$. For each image I_i, it computes a depth map D_i by optimizing a photometric consistency measure, such as the normalized cross-correlation (NCC):

$$NCC(p,q) = \frac{\sum_{\Delta x,\Delta y}(I_p(\Delta x,\Delta y) - \bar{I_p})(I_q(\Delta x,\Delta y) - \bar{I_q})}{\sqrt{\sum_{\Delta x,\Delta y}(I_p(\Delta x,\Delta y) - \bar{I_p})^2(I_q(\Delta x,\Delta y) - \bar{I_q})^2}}, \tag{3}$$

where p and q are the pixel coordinates in the reference and target images, respectively, Δx and Δy represent shifts in the x and y directions from these pixel coordinates within the local windows, and $\bar{I_p}$ and $\bar{I_q}$ are the mean intensities of the local windows centered at p and q.

Utilizing a visibility-consistency criterion, the depth maps are amalgamated into consistent point clouds $\mathcal{P}_i$ which are subsequently merged to form the dense point cloud $\mathcal{D}$ [18]. Hence, the resultant output of this stage is the dense point cloud $\mathcal{D}$, as detailed in Algorithm 3.

3.4.2 Density-Based Spatial Clustering of Applications with Noise

Density-Based Spatial Clustering of Applications with Noise (DBSCAN) is a robust algorithm known for identifying clusters of varying shapes within noisy datasets, without relying on predefined cluster centers or shape assumptions. Instead, it bases its clustering on data point density within a given region. The algorithm employs two critical parameters: ε (the maximum radius of a data point's neighborhood) and $MinPts$ (the minimum number of data points required to form a dense region). It classifies each point as core, border, or noise, and expands clusters recursively by examining the ε-neighborhood of each core point, as detailed in Algorithm 5.

3.4.3 Statistical Outlier Removal

The Statistical Outlier Removal (SOR) method is a widely utilized technique in point cloud data processing, designed to eliminate noise or anomalous points. It estimates a

distance metric for each point based on its mean distance to its nearest neighbors, calculates an average distance and standard deviation, and subsequently classifies any point with a mean discrepancy exceeding a user-specified threshold (typically in terms of standard deviations) as an outlier, leading to its removal. The specifics of the algorithm are presented in Algorithm 4.

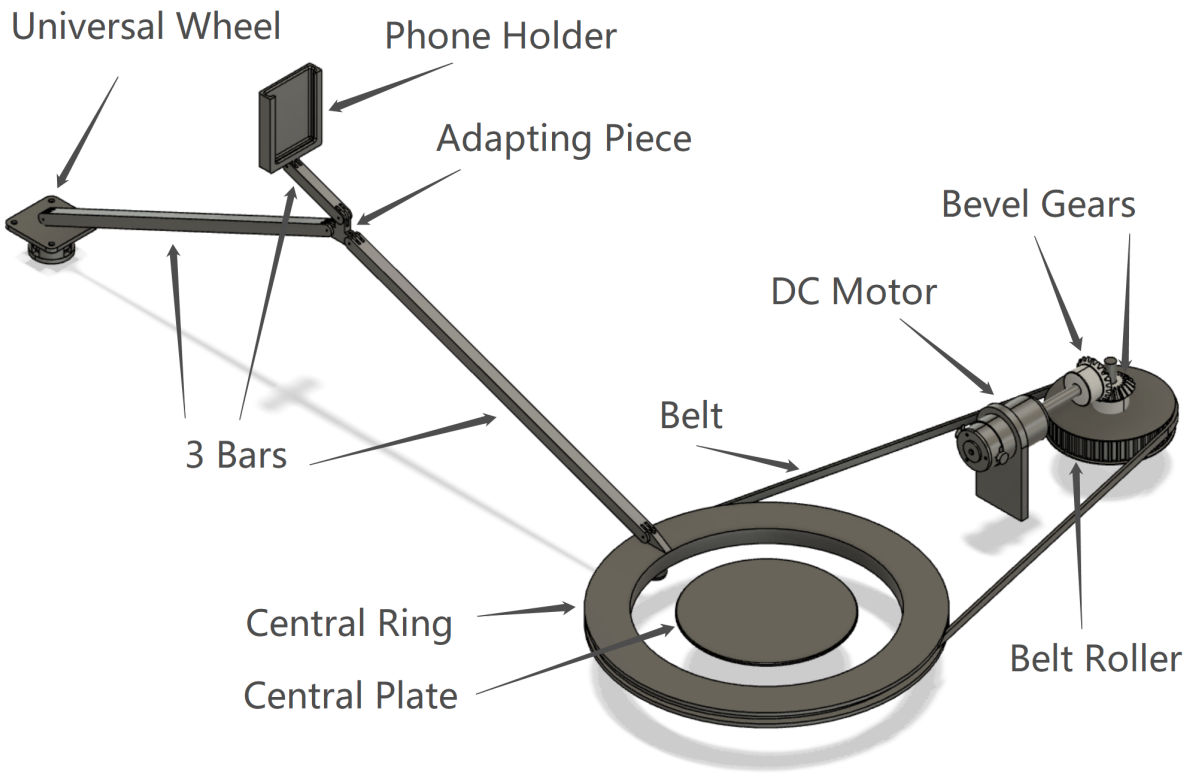


Figure 3: The CAD drawing of the final version of the physical design.

3.5 Phone-holder Subsystem

Figure 3 shows the CAD model we designed in Fusion 360, including labels of all the components. The phone-holder subsystem,which supports the image collection system, is driven by the remote-control subsystem. In our CAD model, the DC motor is designed to rotate a belt roller through a pair of Bevel gears. The belt roller then transmits the kinetic energy to the ”central ring” with the help of a belt. Since the phone holders and the bars are attached to the ring, the phone holder can rotate around the central plate as the central ring rotates. Besides, there is a universal wheel attached to one end of the bar to support the holder, and the combination of the 3 bars as well as the adapting piece

allows the adjustment of the height and angle of the phone.

To scan an object, we need to put the object on the central plate and adjust the position and orientation of the phone to make sure that the object can stay in the photo during the scanning process. After the motor is turned on, the phone holder will take the phone to rotate around the object. As long as the bars and wheel do not hit other components, the phone holder is able to rotate 360 degrees at a constant speed.

3.6 Remote-control Subsystem

To complement the image collection subsystem, we have envisioned the development of a hardware component that combines mechanical ingenuity and adaptability. This device is designed to enhance the functionality and versatility of capturing 3D images using mobile phones.

The remote-control subsystem is a crucial component of the proposed mechanical system. It is responsible for controlling the horizontal shaft that is connected to one side of the gear. The subsystem can rotate the central plate on which the object is placed, providing a clear and convenient way for the mobile phone to capture photos of the object from different angles. In this section, the design details of the remote-control subsystem will be discussed.

In this circuit, for controlling the speed of DC motor, we use a 100K ohm potentiometer to change the duty cycle of the PWM signal. 100K ohm potentiometer is connected to the analog input pin A0 of the Arduino UNO and the DC motor is connected to the 12th pin of the Arduino (which is the PWM pin). The working of Arduino program is very simple, as it reads the voltage from the analog pin A0. The voltage at analog pin is varied by using the potentiometer. After doing some necessary calculation the duty cycle is adjusted according to it.

The motor driver (L298N) can drive two DC motors or one stepper motor. It has four input pins, including Enable A, Enable B, Input 1, and Input 2, which control the direction and speed of the motor. To control the stepper motor, we need to use the two input pins to send the step pulses to the motor.

By integrating this mechanical device into the system, users gain a practical and adaptable tool for capturing 3D images using their mobile phones. The height adjustment feature ensures optimal scanning conditions for various objects, while the remote-controlled

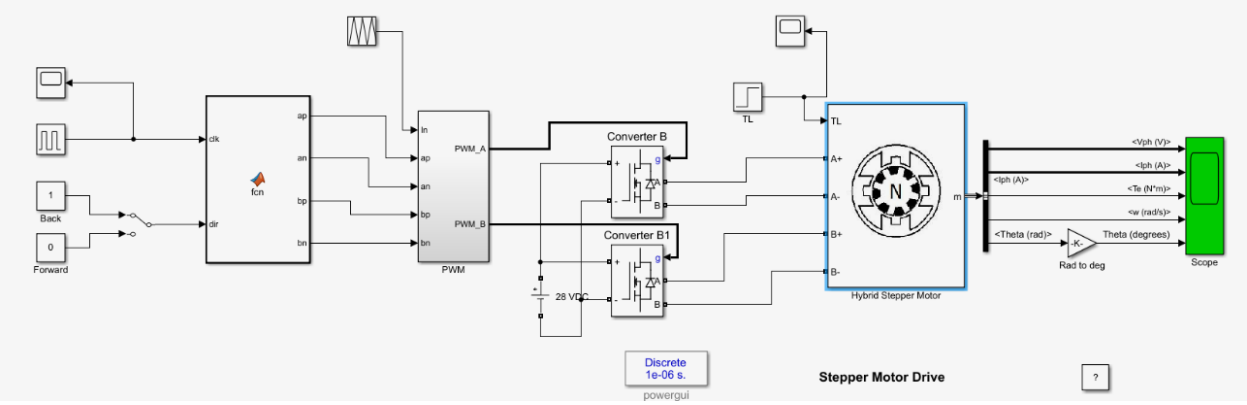


Figure 4: Circuit diagram of the remote-control subsystem.

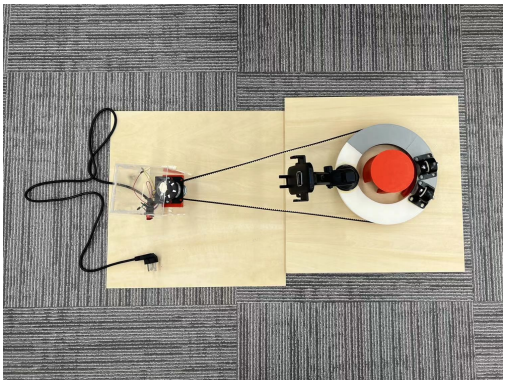
rotation of the phone holder offers flexibility in capturing images from different angles. This hardware component enhances the overall usability and effectiveness of the image collection subsystem, empowering users to achieve accurate and detailed 3D scans with ease.

4 Results

4.1 Mechanical Design

The 3D scanner we built is shown in Figure 5a. We used 2 wooden boards with different heights as the bases of the motor and the ring. For the motor, we made a box using acrylic plates to prevent users from touching the wires, and the motor needs 220 V AC to rotate at a constant speed. For the holder that is used to fix the phone, we purchased a phone holder, which is originally designed for cars. As a result, now the bar of the holder does not have the risk of hitting other components or breaking apart under large loads. Additionally, the size of the holder and the length of the bar are both changeable, making it convenient to place different types of phones and different sizes of objects. A detailed view of the device can be found in Figure 5b.

By the end of the project, our device can rotate around the platform at the center for 360 degrees without hitting anything at a constant speed, while the phone holder is capable of holding a maximum weight of 300 g. Therefore, the requirements of the phone-holder subsystem are successfully met.



(a) The top view of the physical device.

(b) The physical device at work.

Figure 5: The top view (a) and operational view (b) of the physical device.

4.2 Successful Transmission Rate

The reception rate of the frames from the smart phone with respect to the running time is shown in Figure 6. Most of the reception rates are above or close to 80% percent. Since we choose 50 pictures with same interval in hundreds of pictures, this transmission rate fully meets the requirements. The fluctuation of the line may be caused by network issues of the hotspot or the overheat of the smartphone.

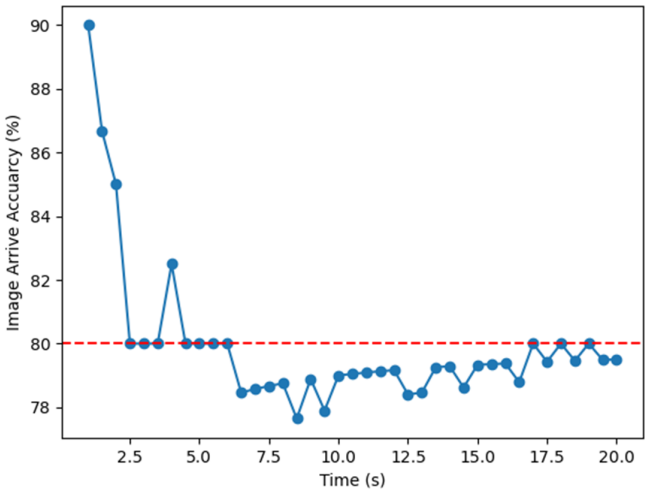


Figure 6: The relationship between time and image arrival accuracy in communication subsystem.

4.3 Simulation Results

To simulate the circuit, we have used software tools MATLAB, which allowed us to test and analyze the system's behavior under different conditions. When the load torque is zero, we simulate the circuit, and the results are shown in Figure 7. These results helped us to optimize the design and ensure the subsystem's proper operation.

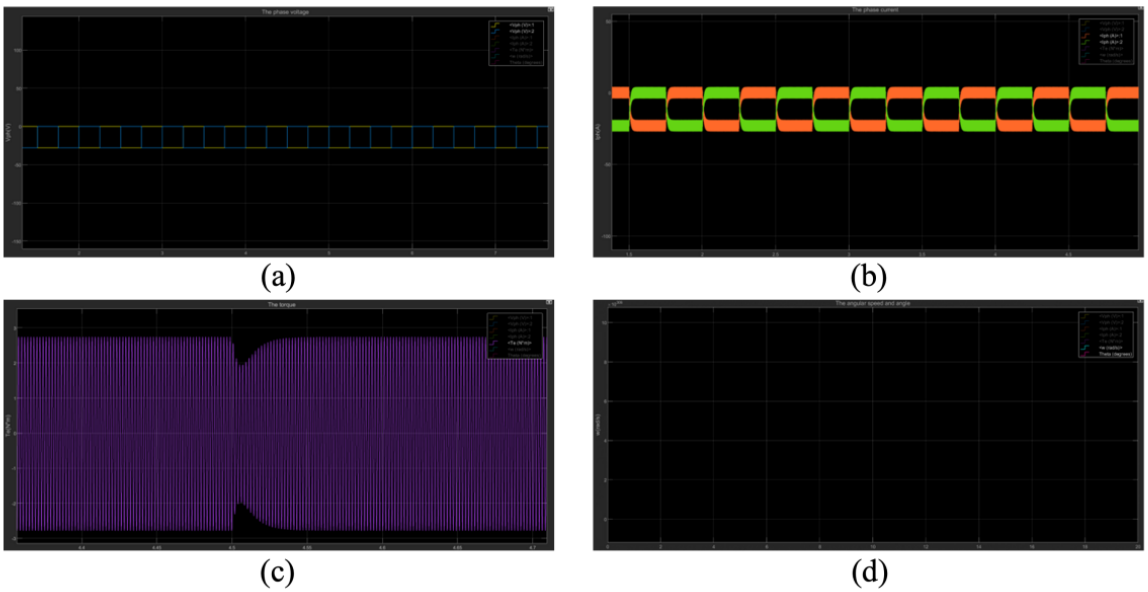


Figure 7: Simulation waveform of the motor when the load torque is 0. (a) represents the phase voltage. (b) represents the phase current. (c) represents the torque. (d) represents the angular speed.

The electromagnetic torque generated by the stepper motor is equal to the sum of the torque generated by the interaction between the phase current and the magnetic flux generated by the magnet and the braking torque generated by the salient pole of the rotor.

$$e_a(\theta) = -\phi_m sin(p\theta)\frac{dt}{d\theta} \tag{4}$$

When the load torque is large, from the simulation waveform, we can see that the voltage and torque are not stable. This will cause the speed of motor not stable, which will also influence the angle of photos the phone takes. Figure 8 is the result of simulated circuit under the conditional of the load torque is 8 N·m.

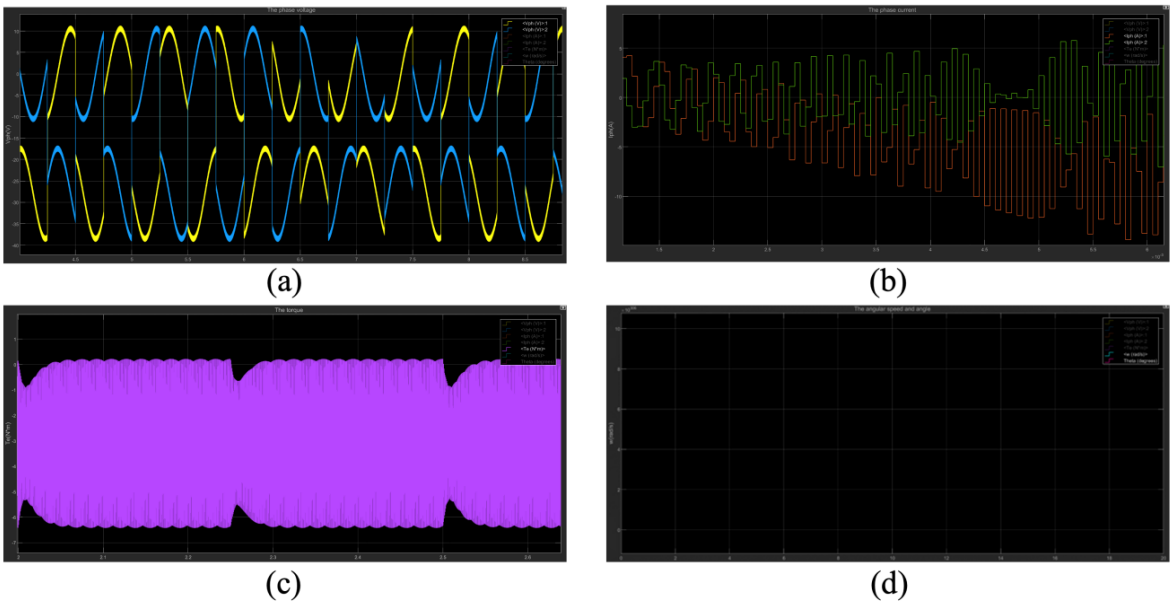


Figure 8: Simulation waveform of the motor when the load torque is 8 N·m. (a) represents the phase voltage. (b) represents the phase current. (c) represents the torque. (d) represents the angular speed.

4.4 Reconstruction Results

4.4.1 Hardware Configuration

In order to obtain high-quality 3D reconstruction results, which often require a significant amount of time, we have employed state-of-the-art testing equipment. For sparse reconstruction, we conducted our experiments using the Apple M2 CPU. As for dense reconstruction, we utilized the NVIDIA RTX 3090 GPU.

4.4.2 Evaluation Metrics

The primary evaluation metrics adopted in this study are: Registered Images Ratio (RIR), Points Number (PN), Mean Track Length (MTL), Mean Observations per Image (MOI), Mean Reprojection Error (MPE), Time, and Human Assessment (HA). RIR signifies the efficiency of the algorithm by quantifying the proportion of 2D images utilized for reconstruction from the total input. PN indicates the density of information within the reconstructed point cloud by representing the number of points therein. MTL and MOI measure the reliability of point generation and the quality of image reconstruction, respectively, by evaluating the average number of cameras observing a single 3D point and

the average number of matched feature points per input image. MPE quantifies the discrepancy between reprojected and actual 2D feature points based on estimated 3D structure and camera poses. HA involves human observers comparing the reconstructed 3D model with the scanned object and providing subjective feedback, while Time captures the total reconstruction duration.

4.4.3 Quantitative and Visual Results

Using our custom-designed scanner, we performed a comprehensive scan of the Nestlé coffee product, capturing 80 consecutive images from various evenly-distributed viewpoints, as shown in Figure 9. These images were subsequently utilized for the purpose of 3D reconstruction.

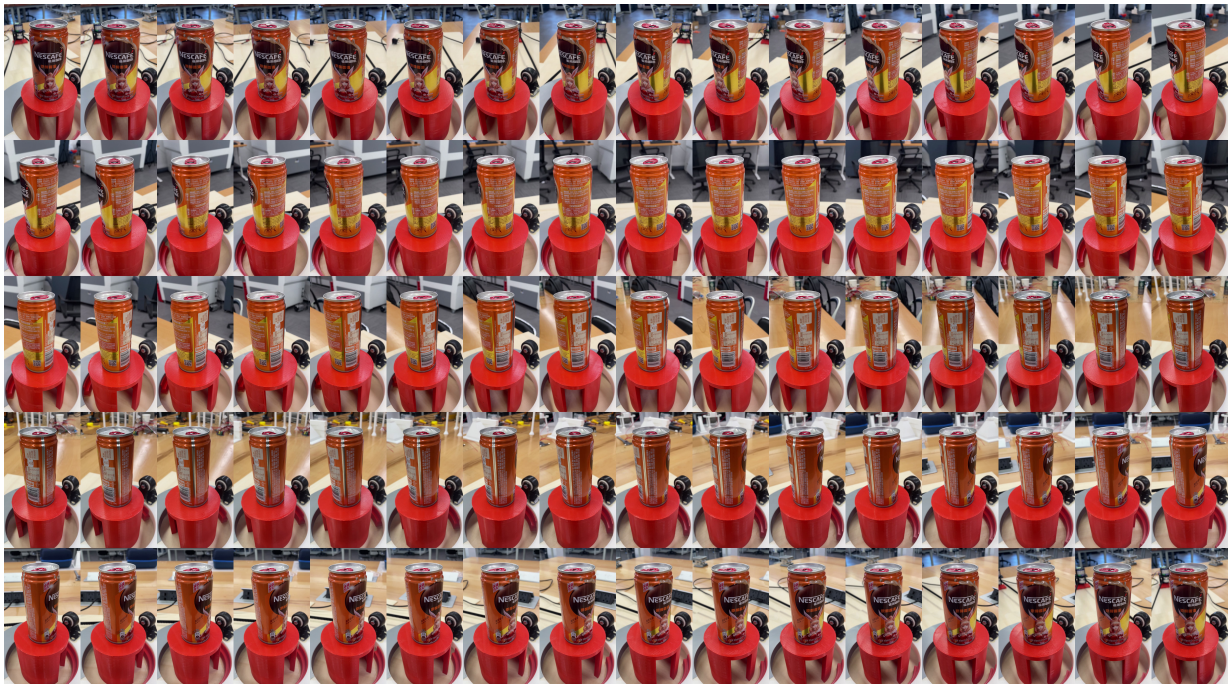


Figure 9: A set of 80 images of a Nestlé coffee bottle captured from various angles.

Following the reconstruction process and removal algorithm, we present our finalized 3D reconstruction results in Figure 10 and reconstruction qualitative result in Table 1. Additional reconstruction results can be found in Appendix D.

Figure 10: Visualization of the reconstruction results of a Nestlé coffee bottle.

Image Number	RIR	PN	MTL	MOI	MPE [px]	Time [s]	HA
80	1	946486	5.812	2079.625	0.537	1276	✔

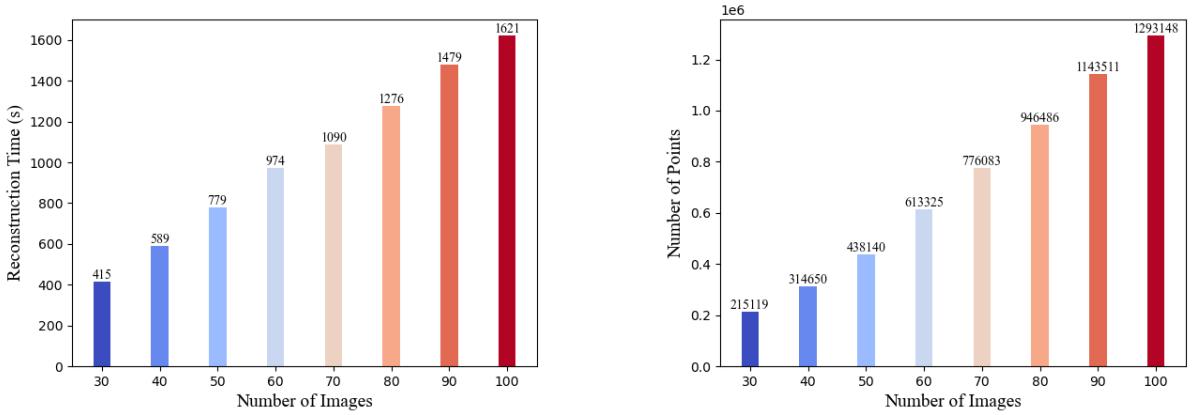
Table 1: Quantitative evaluation of the reconstruction results of a Nestlé coffee bottle.

5 Discussion

5.1 Number of Image vs Reconstruction Quality

We conducted tests on the quality of 3D model reconstruction using object images ranging from 30 to 100 images (with intervals of 10 images). The relationship between the number of images and the reconstruction time is depicted in Figure 11a, while the correlation between the number of images and the quantity of 3D model points is illustrated in Figure 11b. The visual representation of the 3D model as the number of images increases is shown in Figure 12. These results clearly demonstrate that as the number of images in-

creases, both the reconstruction time and the number of points in the reconstructed point cloud exhibit a linear growth pattern. Moreover, it is evident that the reconstruction outcomes tend to improve progressively with an increasing number of images.



(a) Correlation between the number of input images and the reconstruction time of the 3D model.

(b) Correlation between the number of input images and the quantity of points in the reconstructed 3D model.

Figure 11: Graphical analysis of the relationship between the quantity of input images and various reconstruction metrics.

Figure 12: Visual representation of the 3D model as the number of input images increases from 30 to 100 with intervals of 10 images.

5.2 Reconstruction Limitations

The primary limitation of the reconstruction algorithm lies in its inability to effectively reconstruct objects with few surface feature points, such as unicolor and transparent surfaces. This is due to the fact that the reconstruction algorithm lacks knowledge of the depth information of the images and relies solely on feature points extracted using the

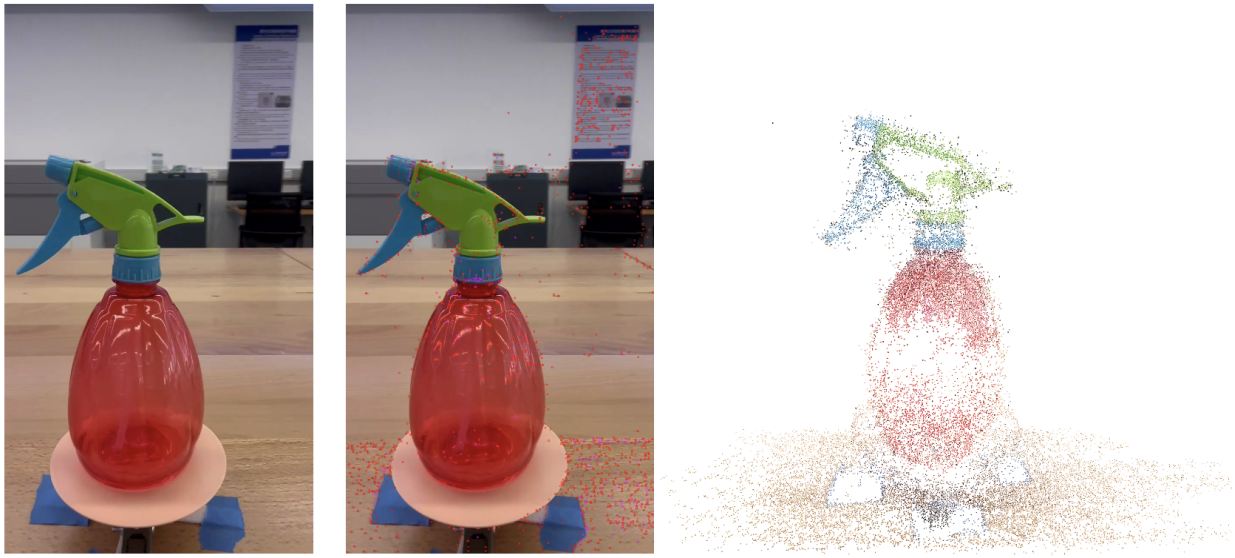


Figure 13: Illustration of a failed reconstruction case, where regions with fewer feature points in the 2D images correspond to sparse areas in the 3D point cloud.

SIFT algorithm for the reconstruction process. If there are only a few feature points extracted by SIFT, areas without feature points cannot be adequately estimated during image reconstruction. Figure 13 illustrates a failed case, where regions with fewer feature points in the 2D images correspond to sparse areas in the 3D point cloud. Additionally, upon examining the high-quality reconstruction results showcased in Appendix D, it is evident that there are still some hollow regions in the unicolor areas.

To overcome the aforementioned limitation, we propose several potential enhancements that could be explored. Firstly, the integration of additional depth sensing techniques, such as stereo vision [19] or time-of-flight cameras [20], may lead to more precise depth information for the reconstruction process. Secondly, examining advanced feature extraction methods beyond SIFT, including deep learning-based feature detectors [21], could improve the detection and extraction of surface features. Lastly, incorporating semantic information or prior knowledge about the object's surface characteristics [22] may aid in filling in gaps and refining the reconstruction quality in areas with scarce feature points.

5.3 Sizes of Objects

Due to the working principle of our scanner, the object been scanned should be placed within the central platform. Thus, for the device we built, the object must be smaller than

a cylinder with a diameter of 10 cm and a height of 25 cm. This is a significant limitation, which prevents users from scanning larger objects. Also, the platform we have now is simply a 3D printed plastic cylinder that cannot change its height, so the height of the platform partly limited the size of the object.

For now, we have come up with 2 possible solutions regarding this limitation. Firstly, we can scale up the ring and the platform. This will not be too difficult because the adjustment can be done on the CAD software and the components can be 3D printed easily. Our second idea is to make a electronically controlled lifting platform. In this way, we can control the height of the platform at a wide range based on the size of the object.

6 Conclusion

This paper presents a novel 3D scanner design, implementation, and validation. It utilizes a custom mechanical device to guide a mobile phone for multi-angle image capture, followed by structure-from-motion algorithm-based 3D reconstruction. The reconstructed models demonstrate exceptional accuracy, with an average reprojection error below 1 pixel. Visual comparisons confirm faithful replication of objects. System requirements ensure reliable image collection, efficient communication, precise image processing, and robust phone-holder mechanics. These achievements establish the scanner as a reliable and accurate tool in 3D reconstruction.

However, it is important to acknowledge the presence of uncertainties and potential ethical concerns within this research. Uncertainty arises from the possibility of errors or inaccuracies in the image capturing process, which may affect the fidelity of the reconstructed 3D models. Robustness and error analysis should be further explored to mitigate these uncertainties and provide a more comprehensive understanding of the scanner's performance limitations. Additionally, ethical concerns may arise regarding privacy and consent when capturing and processing images. It is essential to ensure that proper consent is obtained from individuals whose images are being captured, and to handle the data in accordance with relevant privacy regulations [23]. Ethical considerations should be given due attention to safeguard the rights and privacy of individuals involved in the scanning process.

Looking ahead, our future work on the mechanical system aims to enhance versatility,

accuracy, and ethical compliance. This involves integrating advanced image processing techniques, developing sophisticated algorithms for indistinct objects, and implementing key advancements. These include an electronically controlled lifting central platform, a larger scanning device for bigger objects, an automatic rotation-stop system, and the integration of electrical control parts onto a phone or laptop. For the software system, efforts will focus on improving efficiency and accuracy by utilizing depth-informative cameras, advanced algorithms for Scale-Invariant Feature Transform (SIFT), and a pre-reconstruction system to exclude the shelving unit from the final result.

By addressing these challenges, our aim is to enhance the scanner's capabilities while ensuring ethical compliance. This will contribute to the advancement of 3D reconstruction technology in a responsible and reliable manner.

References

[1] A. Agudo and F. Moreno-Noguer, "A scalable, efficient, and accurate solution to non-rigid structure from motion," *Computer Vision and Image Understanding*, vol. 167, pp. 121–133, 2018.

[2] Y. Cui, S. Schuon, D. Chan, S. Thrun, and C. Theobalt, "3d shape scanning with a time-of-flight camera," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, IEEE, 2010, pp. 1173–1180.

[3] M. Daneshmand, A. Helmi, E. Avots, *et al.*, "3d scanning: A comprehensive survey," *arXiv preprint arXiv:1801.08863*, 2018.

[4] M. Javaid, A. Haleem, R. P. Singh, and R. Suman, "Industrial perspectives of 3d scanning: Features, roles and it's analytical applications," *Sensors International*, vol. 2, p. 100 114, 2021.

[5] A. Haleem and M. Javaid, "3d scanning applications in medical field: A literature-based review," *Clinical Epidemiology and Global Health*, vol. 7, no. 2, pp. 199–210, 2019.

[6] S. Ullman, "The interpretation of structure from motion," *Proceedings of the Royal Society of London. Series B. Biological Sciences*, vol. 203, no. 1153, pp. 405–426, 1979.

[7] N. Snavely, S. M. Seitz, and R. Szeliski, "Modeling the world from internet photo collections," in *International journal of computer vision*, Springer, vol. 80, 2008, pp. 189–210.

[8] J. L. Schonberger and J.-M. Frahm, "Structure-from-motion revisited," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 4104–4113.

[9] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of rgb-d slam systems," in *2012 IEEE/RSJ international conference on intelligent robots and systems*, IEEE, 2012, pp. 573–580.

[10] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski, "A comparison and evaluation of multi-view stereo reconstruction algorithms," in *2006 IEEE computer society conference on computer vision and pattern recognition (CVPR'06)*, IEEE, vol. 1, 2006, pp. 519–528.

[11] Y. Furukawa, B. Curless, S. M. Seitz, and R. Szeliski, "Towards internet-scale multi-view stereo," in *2010 IEEE computer society conference on computer vision and pattern recognition*, IEEE, 2010, pp. 1434–1441.

[12] D. Cernea, "Openmvs: Multi-view stereo reconstruction library," *City*, vol. 5, no. 7, 2020.

[13] P. Moulon, P. Monasse, R. Perrot, and R. Marlet, "Openmvg: Open multiple view geometry," in *Reproducible Research in Pattern Recognition: First International Workshop, RRPR 2016, Cancún, Mexico, December 4, 2016, Revised Selected Papers 1*, Springer, 2017, pp. 60–74.

[14] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," *arXiv preprint arXiv:2003.08934*, 2020.

[15] G. Bradski, "The opencv library.," *Dr. Dobb's Journal: Software Tools for the Professional Programmer*, vol. 25, no. 11, pp. 120–123, 2000.

[16] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, pp. 91–110, 2004.

[17] V. Lepetit, F. Moreno-Noguer, and P. Fua, "Ep n p: An accurate o (n) solution to the p n p problem," *International journal of computer vision*, vol. 81, pp. 155–166, 2009.

[18] Y. Furukawa and J. Ponce, "Accurate, dense, and robust multiview stereopsis," *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 8, pp. 1362–1376, 2009.

[19] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[20] A. Kolb, E. Barth, R. Koch, and R. Larsen, "Time-of-flight cameras in computer graphics," in *Computer Graphics Forum*, Wiley Online Library, vol. 29, 2010, pp. 141–159.

[21] K. Xu, J. Ba, R. Kiros, *et al.*, "Show, attend and tell: Neural image caption generation with visual attention," in *International conference on machine learning*, PMLR, 2015, pp. 2048–2057.

[22] S. Gupta, R. Girshick, P. Arbeláez, and J. Malik, "Learning rich features from rgb-d images for object detection and segmentation," in *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part VII 13*, Springer, 2014, pp. 345–360.

[23] T. U. Senate. "Cybersecurity information sharing act." (2014), [Online]. Available: https://drexel.edu/it/security/policies-regulations/fed-laws/.

Appendix A Requirement and Verification

A.1 Image Collection Subsystem

Requirement:

- ✔ All photos taken should be stored in the local path of the server, and the number of images should match the user's specified quantity.

Verification:

- ❍ The images used for 3D reconstruction should be the same as specified in the GUI.

A.2 Communication Subsystem

Requirement:

- ✔ The vast majority of images collected by mobile devices can be transmitted quickly and reliably to the server-side.

Verification:

- ❍ The successful transfer rate should be higher than 80%.

A.3 Image Processing Subsystem

Requirement:

- ✔ The generated 3D model should bear a resemblance to the scanned object when viewed by the human eye.
- ✔ The mean reprojection error of the sparse reconstruction should be less than 1px.

Verification:

- ❍ Human evaluators should be employed to assess the quality of the generated model, with the vast majority of evaluators concluding that the generated 3D model bears a resemblance to the scanned object.
- ❍ The value of mean reprojection error should be below 1px.

A.4 Phone-holder Subsystem

Requirement:

- ✔ The phone should be able to move and take pictures of an object from many angles.
- ✔ The mechanical device should be able to stand the weight of a typical smart phone without any damage.

Verification:

- ❍ The phone holder can rotate at least 360° around the center without hitting any other parts (the belt, the belt roller, etc). The distance between the universal wheel of the phone holder and the center of the plate should be larger than 544 mm, which is the distance between the furthest end of the belt and the center of the plate.
- ❍ The whole mechanical system should be able to work normally when a load of 300 g (a weight of 240 g for the heaviest iPhone plus another 60 g of tolerance) is applied on the holder.

A.5 Remote-control Subsystem

Requirement:

- ✔ The scanning process takes 10 seconds on average to complete one round.

Verification:

- ❍ The angular velocity of the central ring is 0.628 rad/s ($\pm$10%). The angular velocity of the belt roller is 1.743 rad/s ($\pm$ 10%). The rotation speed of the DC motor should be 16 rpm ($\pm$1).

Appendix B Socket Programming Algorithm

Two algorithms are presented: the **Socket Sending Data** algorithm (Algorithm 1) transmits data over a network connection, while the **Socket Receiving Data** algorithm (Algorithm 2) receives data. They enable seamless client-server communication using sockets.

Algorithm 1: Socket sending data

Input : server_address, server_port, data_to_send

Output: None

```
1 sock ← create_socket()
2 connect(sock, (server_address, server_port))
3 for data in data_to_send do
4     send_data(sock, data_to_send)
5 close(sock)
```

Algorithm 2: Socket receiving data

Input : server_address, server_port, buffer_size

Output: client_data

```
1 server_socket ← create_socket()
2 bind(server_socket, (server_address, server_port))
3 listen(server_socket, 1)
4 while Connection Set do
5     client_socket, client_address ← accept(server_socket)
6     client_data ← receive_data(client_socket, buffer_size)
7 close(client_socket)
8 close(server_socket)
```

Appendix C Point Cloud Algorithm

Three algorithms are presented: **3D Reconstruct** 3 performs structure-from-motion-based 3D reconstruction, **Radius Search** 4 removes outliers using radius search, and **DBSCAN** 5 identifies the largest cluster using density-based spatial clustering.

Algorithm 3: 3D Reconstruct Algorithm

Procedure `Reconstruct` (*Images*)
    Initialize empty model $\mathcal{M}$
    Extract features F_i for each image $I_i \in$ Images
    Match features between image pairs (I_i, I_j)
    Estimate relative camera poses between image pairs (I_i, I_j)
    for *each image* I_i **do**
        Register image I_i to the model $\mathcal{M}$
        Refine camera pose and point positions using bundle adjustment
    return Sparse reconstruction $\mathcal{M}$
Procedure `MultiViewStereo` (*Sparse reconstruction* $\mathcal{M}$, *Images*)
    Initialize empty dense point cloud $\mathcal{D}$
    for *each image* I_i **do**
        Compute depth map D_i for image I_i
        Fuse depth maps into a consistent point cloud $\mathcal{P}_i$
    Merge point clouds $\mathcal{P}_i$ into dense point cloud $\mathcal{D}$
    return Dense point cloud $\mathcal{D}$
Procedure `Reconstruct_Pipeline` (*Images*)
    Sparse reconstruction $\mathcal{M} \leftarrow$ `Reconstruct` (*Images*)
    Dense point cloud $\mathcal{D} \leftarrow$ `MultiViewStereo` (M, *Images*)
    return 3D model $\mathcal{D}$

Algorithm 4: Radius Search Algorithm for Outlier Removal

Data: Dataset D, Radius (r), MinNeighbors ($min\_neighbors$)
Result: Dataset D' without outliers
Initialize dataset without outliers $D' = \emptyset$
for *each point* p *in* D **do**
    $NeighborPts =$ regionQuery(p, r, D)
    if *len*($NeighborPts$) $\geq min\_neighbors$ **then**
        Add p to D'
    end
end
Function `regionQuery` (p, r, D) **:**
    Initialize an empty list N
    for *each point* q *in* D **do**
        if *distance*(p, q) $\leq r$ **then**
            Add q to N
        end
    end
    return N

Algorithm 5: DBSCAN Algorithm for Selecting the Largest Cluster

Data: Dataset D, Eps (ε), MinPts (min_pts)
Result: Largest cluster C_{max}

Initialize set of visited points $V = \emptyset$
Initialize set of clusters $C = \emptyset$
for *each point p in D* **do**
    if $p \notin V$ **then**
        Add p to V
        $NeighborPts =$ regionQuery(p, ε, D)
        if *len*($NeighborPts$) $\geq min\_pts$ **then**
            $C_{new} =$ expandCluster($p, NeighborPts, \varepsilon, min\_pts, D, V$)
            Add C_{new} to C
        end
    end
end
$C_{max} = \arg\max_{c \in C}$ len(c)
Function `expandCluster`($p, NeighborPts, \varepsilon, min\_pts, D, V$)**:**
    Initialize new cluster $C_{new} = \{p\}$
    for *each point* p' *in* $NeighborPts$ **do**
        if $p' \notin V$ **then**
            Add p' to V
            $NeighborPts' =$ regionQuery(p', ε, D)
            if *len*($NeighborPts'$) $\geq min\_pts$ **then**
                $NeighborPts = NeighborPts \cup NeighborPts'$
            end
        end
        if p' *is not in any cluster in* C **then**
            Add p' to C_{new}
        end
    end
    return C_{new}
Function `regionQuery`(p, ε, D)**:**
    Initialize an empty list N
    for *each point q in D* **do**
        if *distance*(p, q) $\leq \varepsilon$ **then**
            Add q to N
        end
    end
    return N

Appendix D Reconstruction Result

Table 2 presents the results of the 3D scanning process for different objects. Each row corresponds to a specific object, and the columns provide information on the image number, Reconstruction Image Ratio (RIR), Point Number (PN), Mean Track Length (MTL), Mean Observations per Image (MOI), Mean Projection Error in pixels (MPE), processing time in seconds, and Human Assessment (HA) indicating whether the assessment passed ✔ or failed ✘.

Type	Image Number	RIR	PN	MTL	MOI	MPE [px]	Time [s]	HA
Cube 14	80	1	522572	4.810	482.913	0.766	1278	✔
Flower 15	80	1	899093	5.610	1632.713	0.598	1292	✔
Glue 16	80	1	843506	7.660	1066.608	0.487	1052	✔
Milk 17	80	1	938261	5.516	1619.863	0.549	1320	✔
Spray 18	80	1	603753	7.653	2432.625	0.546	1238	✔

Table 2: Results of the 3D scanning process for different objects.

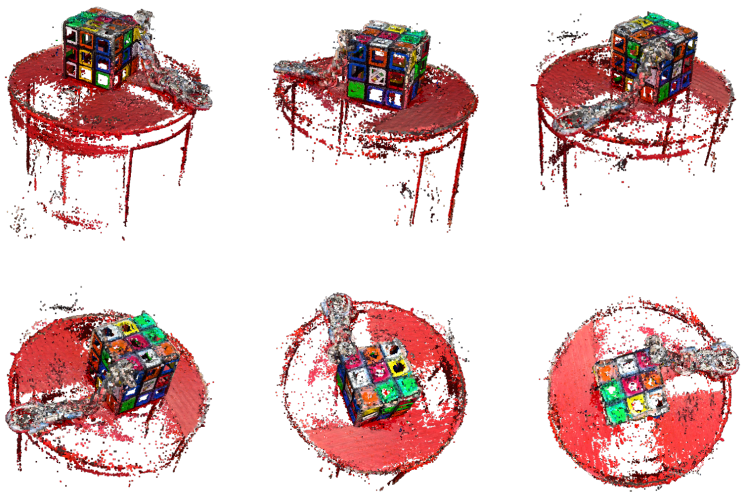


Figure 14: Reconstruction result of the Cube object.

Figure 15: Reconstruction result of the Flower object.

Figure 16: Reconstruction result of the Glue object.

Figure 17: Reconstruction result of the Milk object.

Figure 18: Reconstruction result of the Spray object.