

ECE 445
SENIOR DESIGN LABORATORY
FINAL REPORT

Image acquisition, 3D reconstruction and a visual interactive digital heritage system

Team #9

CHUANRUI CHEN
(cc86@illinois.edu)
DENGLIN CHENG
(denglin3@illinois.edu)
QIANYAN SHEN
(qianyan2@illinois.edu)
ZIYING LI
(ziyingl4@illinois.edu)

Sponsors: Prof. Shurun Tan

May 10, 2024

Contents

- 1 Introduction** 1
 - 1.1 Purpose 1
 - 1.2 Visual Aid 2
 - 1.3 Functionality 2
 - 1.4 Subsystem Overview 3

- 2 Design** 4
 - 2.1 Subsystem Overview 4
 - 2.1.1 Assistive Scanning Subsystem 4
 - 2.1.2 3D Reconstruction Subsystem 6
 - 2.1.3 Database Subsystem 11
 - 2.1.4 Interactive Interface Subsystem 15
 - 2.2 Tolerance Analysis 20

- 3 Cost and Schedule** 21
 - 3.1 Cost Analysis 21
 - 3.1.1 Parts 21
 - 3.1.2 Labor 22
 - 3.1.3 Total 22
 - 3.2 Schedule 23

- 4 Conclusion** 24
 - 4.1 Accomplishment 24
 - 4.2 Uncertainties 25
 - 4.3 Future Work 26
 - 4.4 Ethic Consideration 26
 - 4.4.1 Ethics 26
 - 4.4.2 Safety 27

- 5 Appendix** 28

- References** 28

1 Introduction

1.1 Purpose

Cultural artifacts possess significant historical, cultural, and artistic value. However, due to the passage of time and the impact of natural deterioration, many artifacts face risks of damage, loss, or decay. During the course of study, exhibition, and inheritance, artifacts may also sustain deterioration and damage to cultural heritage[1]. Additionally, for history enthusiasts and researchers worldwide, detailed information about specific artifacts is not readily accessible. Furthermore, Traditional photographs often fail to capture the intricate details of artifacts, hampering comprehensive research and preservation efforts. Therefore, the conservation and exhibition methods of cultural relics have been a common concern of scholars around the world.

Therefore, our team aims to develop a system that can generate realistic 3D models of cultural artifacts and provide users with a user-friendly interactive interface for immersive exploration.

We plan to design a system that can capture the detailed geometric shapes of artifacts using advanced scanning and 3D reconstruction techniques, and create 3D models. Additionally, we need to establish a database to store the collected artifact information and design a user-friendly interface that allows users to easily browse and interact. This will enable us to accurately capture and preserve the features of artifacts and provide a platform for enthusiasts to interact with artifacts up close.

Based on the analysis above, our first requirement is an accurate and efficient system for collecting the visual information of artifacts. We need to address how to handle the positional relationship between artifacts and cameras, as well as how to convert RGBD data into 3D models. Therefore, we need a mechanical device that can ensure the artifacts rotate at the desired speed while controlling the position and angle of the camera relative to the artifacts. This will help us obtain accurate RGBD data. Secondly, we need an efficient and accurate system to convert RGBD images into 3D models. To address this issue, we have employed a point cloud reconstruction method. Firstly, we extract color and depth information from the RGBD images. By processing this information, we obtain point clouds that incorporate both position and color information. After applying denoising and filtering operations to the point cloud, we obtain a 3D model. Subsequently, we map the color information from the point cloud onto the mesh, resulting in a 3D model of the artifact that includes color information.

Nevertheless, considering our goal of better preserving, studying, and disseminating traditional cultural heritage, having only the 3D model data of artifacts is not enough. We also need to build a platform to store and render the models and provide interactive possibilities for users. We will export the reconstructed 3D models and load them into the database subsystem. Users can search for artifacts of interest in the database, and the rendered model data from the search results will be displayed in the interactive interface. Users can then appreciate and study the details of the artifacts up close through operations such as zooming and rotating.

By facilitating the digitalization of cultural heritage and sectors such as education, tourism, and grafting, our technology can generate greater economic benefits and serve as a solid theoretical foundation for the digital display, dissemination, and protection of historical relics[2].

1.2 Visual Aid

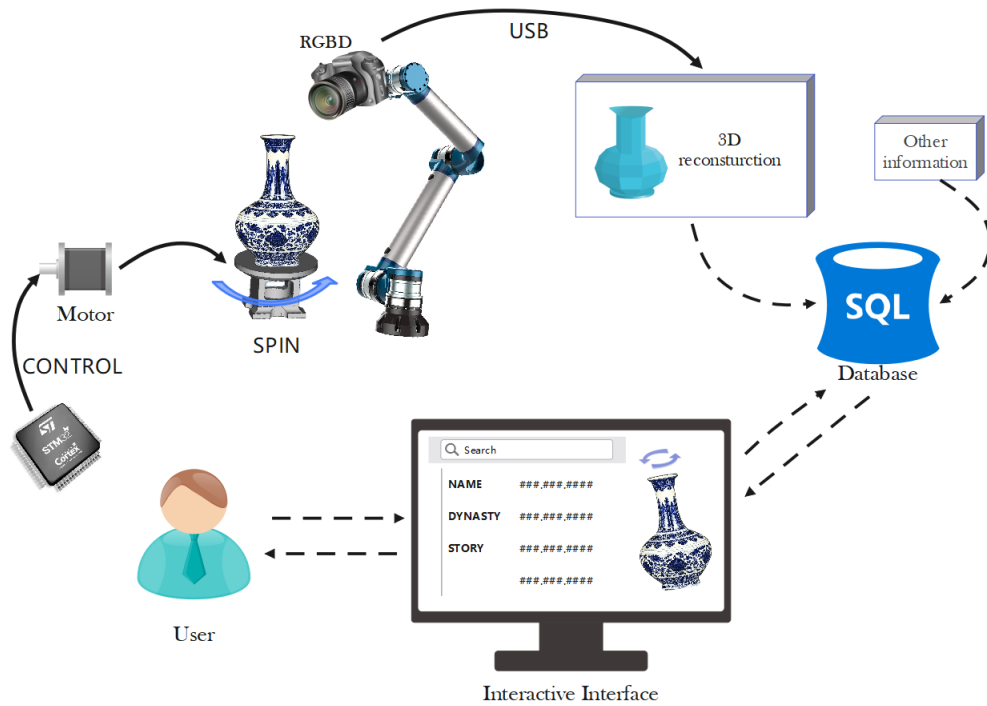


Figure 1: Visual Aid

1.3 Functionality

- **Overall Outcomes:** The system is capable of reconstructing objects with dimensions (including length, width, and height) ranging from 5cm to 20cm. The dimensional accuracy of the obtained 3D models is within 5% error precision. Our system has the capacity to store information on 2500 artifacts.
- **Modeling Outcomes and User Experience:** We are capable of converting color images and depth maps into point clouds, and we offer the ability to save and export the generated models in various formats. The database should store artifact information and 3D models accurately and securely. The system should execute the search and display the result page containing 3D reconstruction models and relevant information about the corresponding heritages within a response time of approximately 3 seconds. Users must be able to search for specific heritages by entering keywords through our website.

- Hardware Level:** The High-level Requirements for the Hardware Aspect of the Project focus on ensuring robust functionality and performance across the entire system. Key goals include achieving a minimum response time of less than 200 milliseconds for real-time control, maintaining precise rotational control of the platform with an accuracy of ± 1 degree and a speed of up to 15 degrees per second for comprehensive artifact coverage, and attaining a scanning resolution of 10 millimeters for capturing detailed artifact features. These requirements are set to ensure that the system is both efficient in operation and capable of producing high-quality 3D scans, supported by seamless integration and communication between the power supply, control unit, sensor module, and mechanical components.

1.4 Subsystem Overview

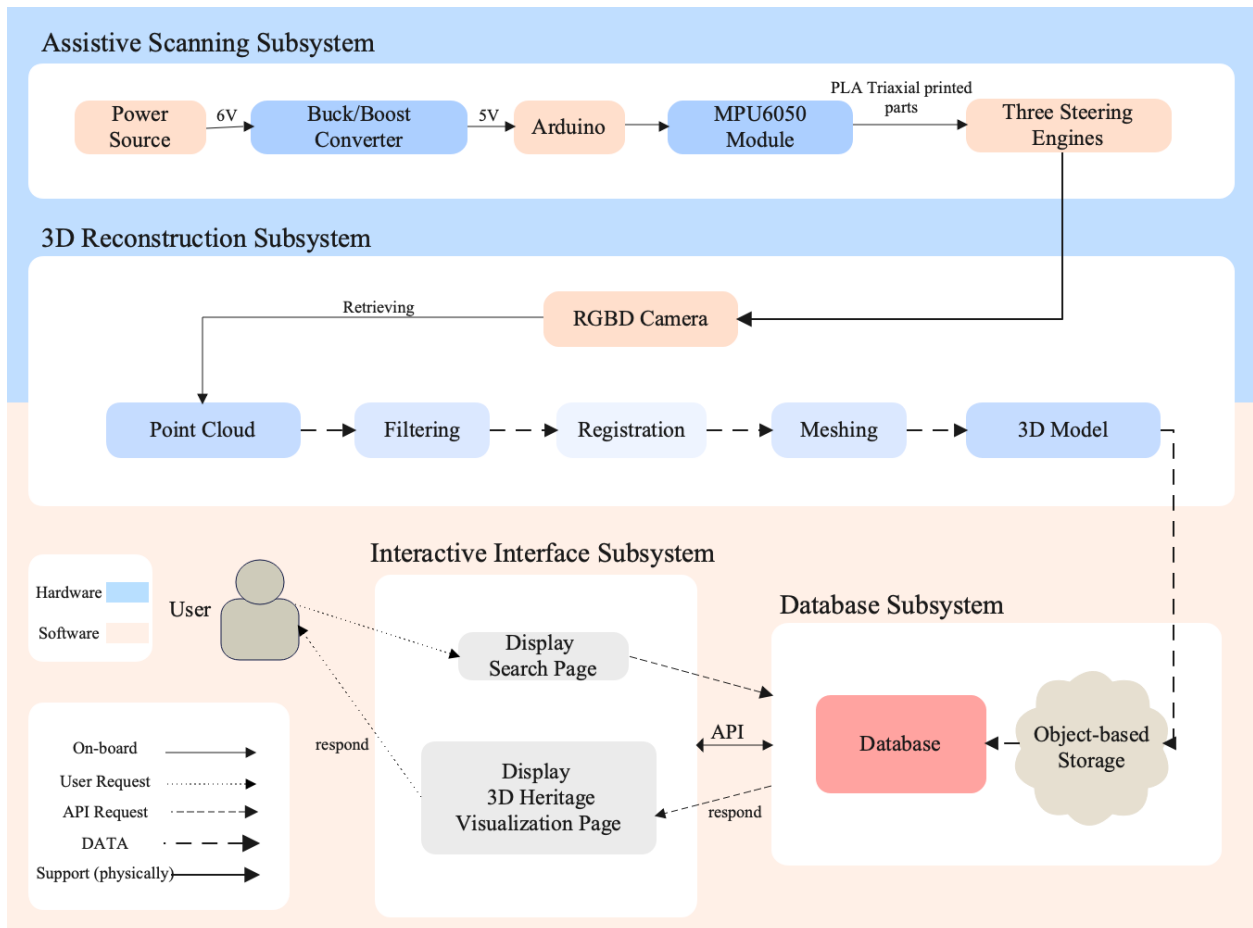


Figure 2: Block Diagram

Our system consists of four subsystems: the Assistive Scanning Subsystem, 3D Reconstruction Subsystem, Database Subsystem, and Interactive Interface subsystem.

The Assistive Scanning Subsystem is purely hardware-based and plays a crucial role in

supporting the cameras and artifacts. It helps determine the relative positions of the cameras and artifacts, facilitating subsequent point cloud processing.

The 3D Reconstruction Subsystem retrieves depth and color information from the RGBD cameras in the scanning subsystem to generate point clouds, ultimately completing the artifact reconstruction process.

The Database Subsystem stores the acquired 3D models and various information about the artifacts. It collaborates with the interactive interface subsystem to enable user search functionality.

The Interactive Interface Subsystem handles 3D model rendering, user selection and search, and provides basic interaction capabilities.

Our system aims to accurately obtain 3D models from physical artifacts and provide functions for storage, display, and interaction. This allows for better management and preservation of the artifacts' content and physical characteristics, while enabling users to conveniently access and appreciate artifacts from around the world.

2 Design

2.1 Subsystem Overview

2.1.1 Assistive Scanning Subsystem

The Assistive Scanning Subsystem is designed as a hardware component of the overall project. The RGBD camera will be placed on a gimbal built by this subsystem to fulfil the goal of achieving precise spatial positioning and movement detection..

At its core, the subsystem relies on a power supply of four 1.5V dry cell batteries, which collectively provide a 6V output. This power source is crucial for driving the various components of the system, including the Arduino Nano, which serves as the central processing unit. The voltage from the batteries is regulated to 5V using a buck converter, ensuring that the Arduino operates within its required voltage range. The Arduino is interfaced with the MPU6050 module, a motion sensor chip known for its dual capabilities of measuring acceleration and rotational angular velocity with a 3.3V/5V interface and a nominal operating current of 5mA.

Moreover, three distinct types of motors are considered for maneuvering the gimbal across three axes. Initially, three MG996R servo motors will be selected for the prototype, owing to its compatibility with the system's voltage requirements and its sufficient torque to control the gimbal's axes. However, for subsequent iterations, a transition to more advanced motors is contemplated. Two alternatives are proposed: the S42H40D20 stepper motor and the GM2804 brushless motor, which offer different benefits in terms of precision, torque, and operational speed. Each motor's specifications have been provided to ensure the selection aligns with the system's high-level requirements.

To refine the prototype into a seamless unit that harmonizes the MPU6050 sensor's input with the dynamic responsiveness of the servo motors. The forthcoming period will be dedicated to optimizing the codebase for enhanced accuracy and streamlining the interaction between hardware elements to bolster overall system stability. Looking ahead, the vision is set on elevating the prototype to a state-of-the-art scanning apparatus. This includes the strategic integration of stepper motors to replace the current servos, aiming to achieve unrivaled precision in the 3D scanning process. The transition to stepper motors is anticipated to necessitate a series of rigorous testing phases, which will be a primary focus in the upcoming weeks. Finally, the culmination of these endeavors will be the development of a fully functional 3D scanning subsystem. The success metrics for this phase are clearly defined: a minimum scanning resolution of 10 millimeters, agile response times, and a comprehensive scanning operation of artifacts within a 5-minute threshold. The steadfast pursuit of these milestones underscores our commitment to delivering a subsystem that meets the highest echelons of accuracy and efficiency in 3D scanning technology.

In summary, the 3-axis gimbal will stabilize and support the RGBD camera for a smoother and perfect scanning process. Together with the rotating platform of the object being scanned, the effect of the noise from the scanning will be physically reduced. Meanwhile, the data scanned by the RGBD camera will collaborate with several other subsystems, and the final scanned data will be transferred to the 3D reconstruction subsystem to support the efficiency and accuracy of the whole design.

As shown in the figure, in principle, we draw the Buck/Boost converter, Arduino and MPU6050 module. Then, we connect each parts based on the relationship between them.

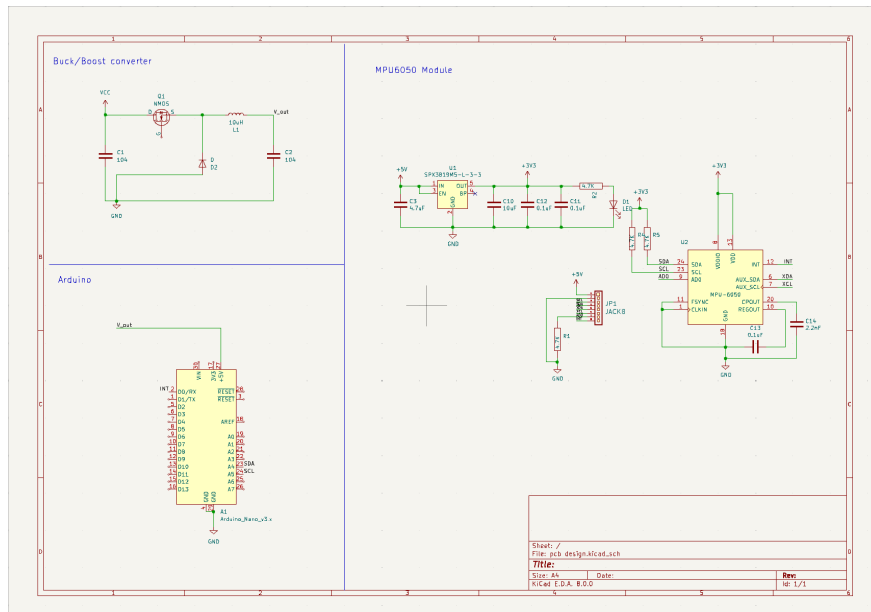


Figure 3: PCB Design

Requirement Description	Verification Procedure
1 Arduino Nano must receive a stable 5V supply, to maintain operational stability and prevent damage to the board	Verify voltage post-buck converter using a multimeter
2 MPU6050 must accurately capture motion data within manufacturer specs, to ensure precise control and feedback for motor movement	Test in Arduino coding software with known motion patterns and compare to MPU6050 output data
3 Servos must respond accurately to control signals within its operational range, to achieve the desired angular positioning for the scanning functionality	Test full range of motion and measure with protractor or similar device
4 Subsequent versions may use a S42H40D20 stepper motor or a GM2804 brushless motor, subject to power and performance requirements.	Compare motor specifications against system requirements.
5 The Assistive Scanning Subsystem shall perform reliably across all axes of the gimbal within the operational parameters of the motors.	Perform a system reliability test across all axes.

Table 1: Requirements and Verifications table for Assistive Scanning Subsystem

2.1.2 3D Reconstruction Subsystem

This subsystem aims to acquire point clouds through RGBD images and utilize them for 3D reconstruction.

The system takes depth images and colour images as input from the RGBD camera of the Assistive Scanning Subsystem. In order to generate 3D models with lower errors, the system not only needs to acquire point clouds and perform 3D reconstruction on them but also employs various algorithms for denoising and smoothing the point clouds. Through these processes, the system can obtain 3D models that are very close to the actual artifacts, enabling preservation and appreciation.

We choose Kinect v2 as our RGBD camera. In this subsystem, we need to obtain the depth and colour frame sequences of the rotating objects using the Kinect SDK, and then map the pixels on each depth image to 3D space through coordinate transformation to obtain the initial point cloud. Since we also need the artifact texture information, we also need to

perform a mapping between the depth image and the colour image to obtain the colour information for each 3D point. Then we store the spatial position and colour information of each point in the point cloud created by Open3D.

Algorithm 1 Obtaining point cloud from Kinect

Input: Kinect device

Output: Point cloud data

Initialize Kinect device

Create point cloud container, *PointCloud*

while *Capturing point cloud* **do**

 Get depth image, *DepthImage*, and color image, *ColorImage*

 Get depth value, *depth*, and color value, *color*, from *DepthImage* and *ColorImage*

for *each pixel* (*x, y*) **do**

 Convert pixel coordinates (*x, y*) to 3D coordinates (*X, Y, Z*)

if *depth is valid* **then**

 Create a point, *Point*, and assign (*X, Y, Z*) and *color* to *Point*

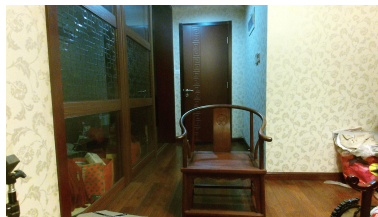
 Add *Point* to *PointCloud*

end

end

end

return *PointCloud*



(a) 1920 * 1080 input Color Image



(b) 512 * 424 input Color Image



(c) Retriwing Point Cloud

Figure 4: Chair Example for Point Cloud Retriwing

Note that images acquired from the Kinect SDK are mirrored. the Kinect SDK has no built-in way to address this, and it must be done by manually flipping the X-axis when reading pixels.

After obtaining the initial point cloud, we first need to segment the point cloud to separate the point cloud of the desired object from the background. In order to do this, we need to simultaneously acquire a background point cloud without the target object A and a point cloud with the target object B. Then we need to align the point cloud with B as the Target

and A as the Source. The result is two point clouds with uniform coordinate system. The two point clouds are then cropped to obtain a localized point cloud containing the target object and a localized point cloud of the ground on which the target object is located, respectively. Then we merge the two point clouds and perform planar segmentation to get the point cloud of the object with the ground removed. Then the clustering operation is performed on the point cloud to obtain the most aggregated point cloud block with the highest number of points, i.e., the point cloud data of the target object.



Figure 5: Point clouds of chair from different viewpoints

After segmentation the point cloud is first denoised. Statistical methods are used to find the Outlier. specifically a statistical analysis of the neighborhood of the points is needed to eliminate the points that do not meet specific criteria. We calculate the average distance from each point to all its neighbors. The resulting distance distribution is assumed to follow a Gaussian distribution with a mean and a standard deviation. Any point that exceeds the interval defined by the standard deviation can be recognized as an outlier and subsequently removed. This allows us to obtain point cloud data from a number of different angles. Since the scans are performed from different viewpoints, the obtained point clouds often have separate local coordinate systems and need to be converted to uniform global coordinates, a process known as point cloud alignment.

We perform coarse alignment first and fine alignment later. That is, the RANSAC method is used first for iteration. Then the Iterative Closest Point (ICP) method is used. The core idea is to iterate the nearest point method to make two point clouds close to each other, and finally minimize the distance error between two point clouds. However, the problem of this operation is that it is easy to fall into the local optimization and lack of overlapping region, which leads to poor alignment accuracy. Therefore, it is more suitable for objects with distinctive structural features. For objects with regular shapes (e.g. vases and water bottles), additional measures are needed. For such objects, we need to select overlapping regions with clear features for alignment, and then apply the Transformation matrix obtained from the alignment to the source point cloud.

Algorithm 2 Iterative Closest Point (ICP) algorithm

Input: Source point cloud A, Target point cloud B, an initial transformation matrix**Output:** Transformation matrix

```
 $T \leftarrow T_0$ 
while notconverged do
  for  $i \leftarrow 1$  to  $N$  do
     $m_i \leftarrow \text{findClosestPointInA}(T \cdot b_i)$ 
     $w_i \leftarrow 0$ 
    if  $\|m_i - T \cdot b_i\| \leq d_{\max}$  then
       $w_i \leftarrow 1$ 
    end
  end
   $T \leftarrow \underset{T}{\operatorname{argmin}} \{ \sum_i w_i \|T \cdot b_i - m_i\|^2 \}$ 
end
return  $T$ 
```



Figure 6: Registered Point clouds of chair from different viewpoints

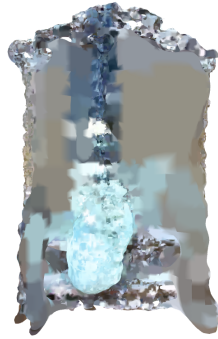
Reconstruction follows. We use the Ball Pivoting Algorithm (BPA) and the Poisson Surface Reconstruction Algorithm. The BPA algorithm starts from a seed triangle and defines a rolling ball with radius r . The ball is rolled along the edges, and it is determined whether there are other points inside the ball to form a new triangle. The BPA algorithm starts from a seed triangle and defines a rolling ball with radius r . The ball is rolled along the edges, and it is determined whether there are other points inside the ball to form a new triangle. repeated until all points are included in the triangle mesh. The BPA algorithm works well for uniformly distributed point clouds, but it becomes challenging to choose an appropriate rolling ball radius for non-uniform point clouds, which affects the quality of the 3D model. Therefore, in our project, the BPA algorithm is very accurate in portraying the shape of the object, but in the case of the BPA algorithm, the shape of the object is not as accurate as it should be. shape of the object, but it is prone to produce holes on the surface.

Poisson Surface Reconstruction is a commonly used 3D reconstruction method for generating smooth surface models from point cloud data. It is based on the numerical solution

of Poisson's equation, which infers the interior of the surface by considering the point cloud data as boundary conditions of the surface. The advantage is the ability to handle irregularly distributed point cloud data and generate smooth surface models. The disadvantages are but slower processing speeds and the inference of non-existent regions in low-density point cloud regions, which need to be additionally removed.



(a) Reconstruction
Result of BPA



(b) Reconstruction
Result of Poisson



(c) Reconstruction
Result of Poisson re-
moving low densities
part



(d) Real Color im-
age of This Ornament
(mirrored)

Figure 7: Example for Reconstruction

Requirement Description	Verification Procedure
1 The inputs to the Assistive Scanning Subsystem to this subsystem are $1920 * 1080$ colour images and $512 * 424$ depth images, and the subsystem is capable of acquiring a point cloud from images of this resolution.	For the subsystem, it is necessary to show the acquired colour and depth images, save and display the acquired point cloud.
2 The subsystem converts the point cloud into a smooth as possible 3D model.	The subsystem needs to visualize the reconstruction results. Manual checking of the model for continuity and smoothness.
3 The subsystem is capable of converting point clouds with 10,000 points or more.	Input large-scale point cloud data to observe the processing power and efficiency of the subsystem.
4 The reconstructed model can be saved as OBJ and other formats for storage in a database and visualisation in the interactive interface subsystem.	The saved model can be imported into professional software for visualisation in OBJ format.

Table 2: Requirements and Verifications table for 3D Reconstruction Subsystem

2.1.3 Database Subsystem

This subsystem aims to store the basic information of the artifacts, including countries, historical backgrounds, etc., and at the same time save the generated complex 3D model data. When a user wants to retrieve an artifact, the database will find the corresponding information from its own stored data according to the search item entered by the user and display it through the Interactive Interface Subsystem for users to view artifacts from around the globe.

The data generated by the 3D Reconstruction Subsystem will be loaded into the Database Subsystem. With this database, users can search and view artifacts from exotic countries. Based on this requirement, we opted for **Relational Cloud Database RDS MySQL (RDS)**. Selecting an MySQL database is pivotal in guaranteeing data precision and dependability owing to its stringent data consistency, transaction oversight, and intricate querying capabilities. However, it is primarily designed for storing structured data and is generally not suitable for directly storing large files or unstructured raw file content. Therefore, we also adopted **Object Storage Service (OSS)** to address this need. OSS can store files of any type, including OBJ format files generated by 3D modeling software. OSS provides data upload, download, management, and distribution services based on HTTP RESTful API, allowing OBJ format files to be directly uploaded to OSS buckets

OSS. RDS can store URL links pointing to OBJ files in OSS. This allows RDS to record the storage location information of OBJ files in OSS, achieving data association between the relational database and object storage. Therefore, in our Database Subsystem, RDS stores metadata of the model (such as model ID, name, historical background, etc.) and the URL link, while OSS stores the model files themselves. By recording the URL or other reference information of OSS objects in RDS, it enables the database to locate and retrieve the corresponding OBJ model files during queries.

Our database design contains one table, "Artifacts", which has "ID" as the primary key. The attribute "Link" represents the link to the corresponding OBJ files returned by the object-based storage. "Name", "Year", "Country" and "Historical background" are vital information about artifacts. The relationship is shown in the ER diagram below:

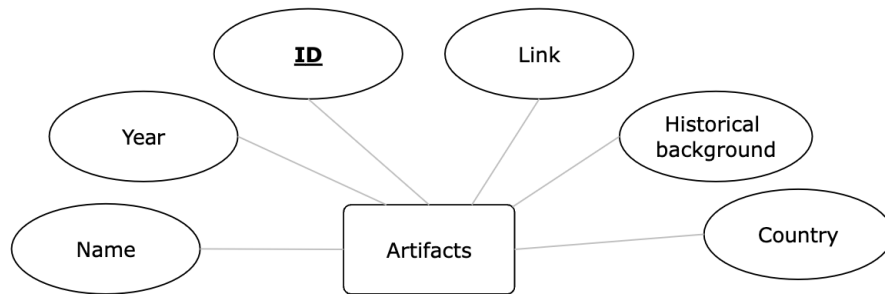


Figure 8: Entity-Relationship Diagram

In addition, for the creation and management of running virtual server instances, we also utilized **Elastic Compute Service (ECS)**. With 1M external bandwidth, it can meet the concurrent access of 10-15 people online at the same time. These servers not only support concurrent user accesses but are also highly flexible, accommodating various operating systems and application software for easy scalability. They can seamlessly adapt to changes in resource demands due to business growth. Moreover, they offer advanced functionalities such as hot migration and snapshot backup for enhanced management capabilities **ECS**.

Overall, the Database Subsystem serves as a connector in the software component of the project. The interactions with other subsystems utilize the Alibaba Cloud SDK. The Alibaba Cloud SDK is a pre-packaged dependency package for calling cloud product OpenAPIs. **Alibaba Cloud OpenAPIs** offer SDKs for multiple programming languages and development frameworks, facilitating developers in rapidly constructing applications related to Alibaba Cloud (including Java, Python, Node.js, Go, among others) **SDK**. The whole picture of interaction between the database subsystem and other subsystems is as follows:

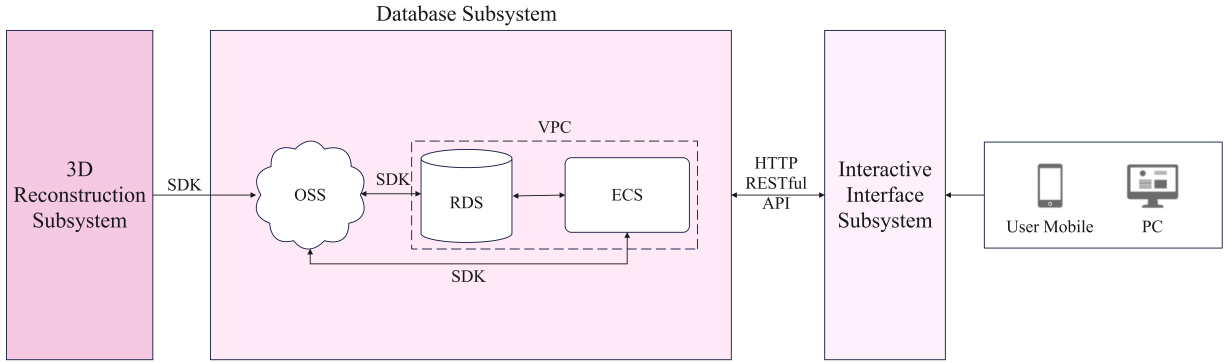



Figure 9: Interfaces between subsystems and within the database subsystem

In order to provide a more detailed explanation of the interaction process between the Database Subsystem and other subsystems, I will proceed to separately explain the interaction between the Database Subsystem, the 3D Reconstruction Subsystem, and the Interaction Interface Subsystem.

Regarding the interaction process between the Database Subsystem and the 3D Reconstruction Subsystem, this process is realised through the corresponding page on the website (as shown in figure 10a below), where Obj files from 3D Reconstruction Subsystem uploaded through the upload button on the website are uploaded to the node.js server via the **Multer storage engine**, and then uploaded to the OSS via the Alibaba Cloud SDK. After that, OSS generates a URL signature which is then stored in RDS with SDK. The pseudocode is as follows:



Contribute Artifacts You have

Artifact Contribution Steps

Step 1: Choose your OBJ file of your artifact below and click 'Upload'. Upon successful upload, you will be automatically redirected to Step 2 page.

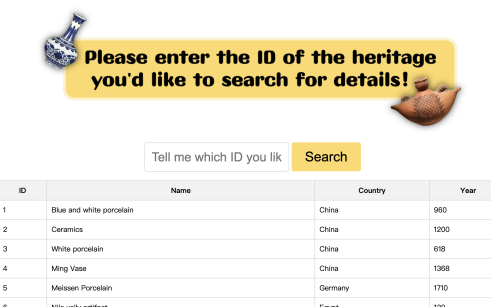
Step 2: Fill in the fields below with the relevant details of the artifact. Once you're done, click 'Submit' to complete the contribution process.

Your Artifact OBJ File:

Upload

uploadFile

Back



Please enter the ID of the heritage you'd like to search for details!

Tell me which ID you like **Search**

ID	Name	Country	Year
1	Blue and white porcelain	China	960
2	Ceramics	China	1200
3	White porcelain	China	618
4	Ming Vase	China	1368
5	Meissen Porcelain	Germany	1710
6	Nile valley artifact	Egypt	120

(a) Web page for showing information on all artifacts from the database

(b) Web page for showing information on all artifacts from the database

Figure 10: Database-related web pages.

Algorithm 3 Upload OBJ files using SDK

yourEndpoint \leftarrow the Endpoint corresponding to the region where the Bucket is located. For example, for East China 1 (Hangzhou), the Endpoint is filled in as <https://oss-cn-hangzhou.aliyuncs.com>.

BucketName \leftarrow the Bucket name, for example, *examplebucket*.

ObjectName \leftarrow the complete object path, the complete path cannot contain the Bucket name, for example, *exampledir/exampleobject.txt*.

Initialize network and other resources.

credentialsProvider \leftarrow access credentials from environment variables

Visit OssClient using Endpoint and credentialsProvider

content \leftarrow the file content with the complete local file path

request \leftarrow the request to get the object with BucketName, ObjectName, and content

outcome \leftarrow the result with request for uploading

if not success then

output "PutObject fail, code:" + outcome.error().Code() + ", message:" + outcome.error().Message() + ", requestId:" + outcome.error().RequestId()

return -1

end

Release network and other resources

return 0

For interaction between Database Subsystem and the Interaction Interface Subsystem, the detailed process is shown in figure 11. Users first input the ID of the artifact they want to search for on our website. Then, ECS forwards this information to RDS, which retrieves and finds the corresponding artifact information from its own database. If the retrieval fails, an error is returned to the webpage. If successful, the corresponding URL link is used to retrieve the OBJ file. The website takes all the information about the artefacts from the RDS and displays it in a tabular form on the web page, and the corresponding page is shown according to figure 10b.

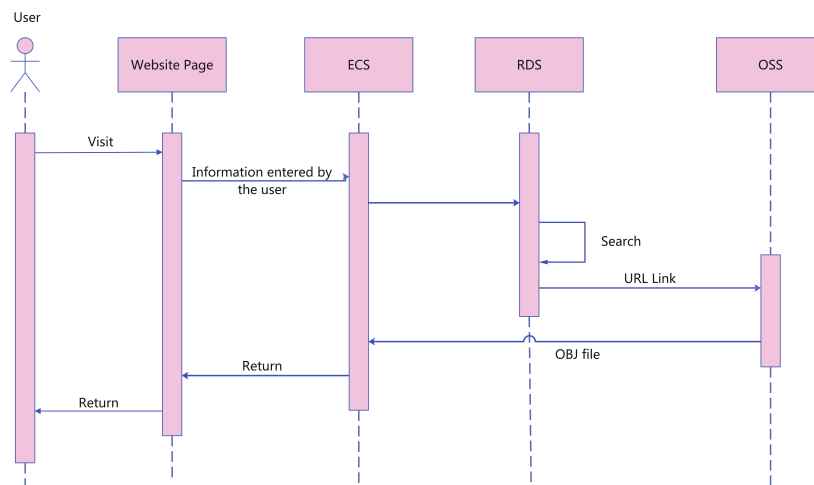


Figure 11: User search sequence diagram

Requirement	Description	Verification Procedure	Timeline
1	The subsystem should be able to store OBJ files in Object Storage Service (OSS).	Manually check if the OSS stores the OBJ files successfully.	Before the end of March
2	The subsystem should be able to store the metadata and the URL links that returned by OSS.	Manually check if RDS stores the metadata and the URL links that returned by OSS.	Before the end of April
3	The subsystem should be able to retrieve artifacts based on user search queries.	Manually check if the database correctly retrieve artifacts based on user search queries.	Before the mid of May
4	The subsystem should be able to establish a interaction between the database subsystem and the Interactive Interface Subsystem.	Manually check if the subsystem can provide data to the Interactive Interface Subsystem for artifact viewing.	Before the end of April

Table 3: Requirements and Verifications table for Database Subsystem

Components	Contingency plans
1 OSS SDK fails to upload OBJ files.	Use the OSS console to upload OBJ files.

Table 4: Detailed Contingency Plan

2.1.4 Interactive Interface Subsystem

The Interactive Interface Subsystem is responsible for facilitating user interaction with the visual heritage system. It provides a graphical user interface (GUI) through which users can search for artifacts, view relevant information, and explore 3D models. This subsystem interprets user requests, retrieves data from the Database Subsystem, and renders 3D models along with accompanying information. It also incorporates interactive control functions to enhance the user experience.

In the design of the Interactive Interface Subsystem, we opted to utilize the Express frame-

work combined with WebGL technology to achieve graphical rendering and web development.

The Express framework is a JavaScript library that supports a component-based development approach. This method simplifies the construction of user interfaces, making them more modular and enhancing application performance and maintainability. For the overall design and layout of the user interface, we incorporated elements such as buttons, forms, and navigation bars to facilitate user searches and browsing. Additionally, we implemented features such as model scaling and rotation, allowing users to interact with the modeled artifacts. These interactive functionalities not only enhance the usability of the user interface but also make the entire user interaction page subsystem more intuitive and dynamic. Users can interact with the 3D models through simple operations, gaining deeper insights into the structure and characteristics of the models.

Regarding rendering, this subsystem retrieves OBJ files from the database subsystem, obtaining vertex, texture, and normal information, as well as vertex information for the faces. Subsequently, we utilize shaders to render and visualize the retrieved information. For graphic rendering, we chose to use WebGL as the API. This is a web-based graphics rendering technology that allows the creation of 3D and 2D graphics using JavaScript and the OpenGL API within a web browser. As WebGL operates on the underlying GPU, we need to use GLSL language to implement shader functions for rendering. Specifically, we require paired vertex and fragment shaders. The vertex shader calculates the position of vertices, while the fragment shader computes the color value of each pixel in the currently drawn primitive. Based on this, WebGL rasterizes primitives such as points, lines, and triangles. Additionally, as we aim to display 3D models on a 2D screen, multiple sequential coordinate system transformations of the model's coordinate system are required during the visualization process. This process leads to obtaining Clip Space, followed by perspective divide to obtain normalized device coordinates. Finally, the shaders automatically perform viewport mapping to convert to Screen Space.

Requirement Description	Verification Procedure
1 The subsystem shall provide a search bar for users to search artifacts by keywords.	Verify that the search bar is visibly present on the website’s interface and that users can input keywords to initiate a search.
2 The subsystem shall display 3D models and relevant information within 3 seconds of initiating a search.	Conduct performance testing to measure the time it takes for the website to load search results and verify that it meets the specified requirement.
3 Users shall be able to manipulate and explore 3D models using interactive controls such as zooming and rotating.	Perform user testing to ensure that interactive controls for zooming, rotating, and panning 3D models are functional and intuitive to use.
4 The subsystem shall allow users to navigate back from the result page to the search page.	Manually test the website’s navigation functionality to confirm that users can return to the search page from the result page using provided controls or browser features.

Table 5: Requirements and Verifications table for Interactive Interface Subsystem

The workflow of Interactive Interface Subsystem involves two crucial stages. Firstly, user interaction design ensures an intuitive interface, allowing easy browsing of artifacts and interaction with 3D models. Upon user requests, the subsystem communicates with the Database Subsystem to retrieve artifact information, including OBJ format 3D model files. Secondly, rendering 3D model files onto the user’s screen is critical. This involves handling vertex, texture, and normal information from OBJ files and applying shaders for visualization. Interactive features allow users to manipulate and explore 3D models, enhancing the overall experience with seamless database integration for real-time updates.

The design of website structure offers a seamless journey. The structure is shown in figure 12. Users start at the welcoming page, then proceed to the mode choice page, offering options to explore existing artifacts or contribute new ones. Exploring artifacts leads to a search page with a database of artifacts. Users can locate artifacts quickly by ID, accessing detailed information including ID, name, country, year, background, and interactive 3D models for exploration. Alternatively, users can contribute artifacts by uploading OBJ files and relevant information on a dedicated page. Submitted artifacts enrich the platform’s collection, aiding in the preservation of cultural heritage. Consistent navigation via a back button ensures an intuitive experience, allowing users to easily navigate back

through their exploration or contribution journey. This streamlined structure ensures a seamless and user-friendly experience, facilitating both exploration and contribution efforts.

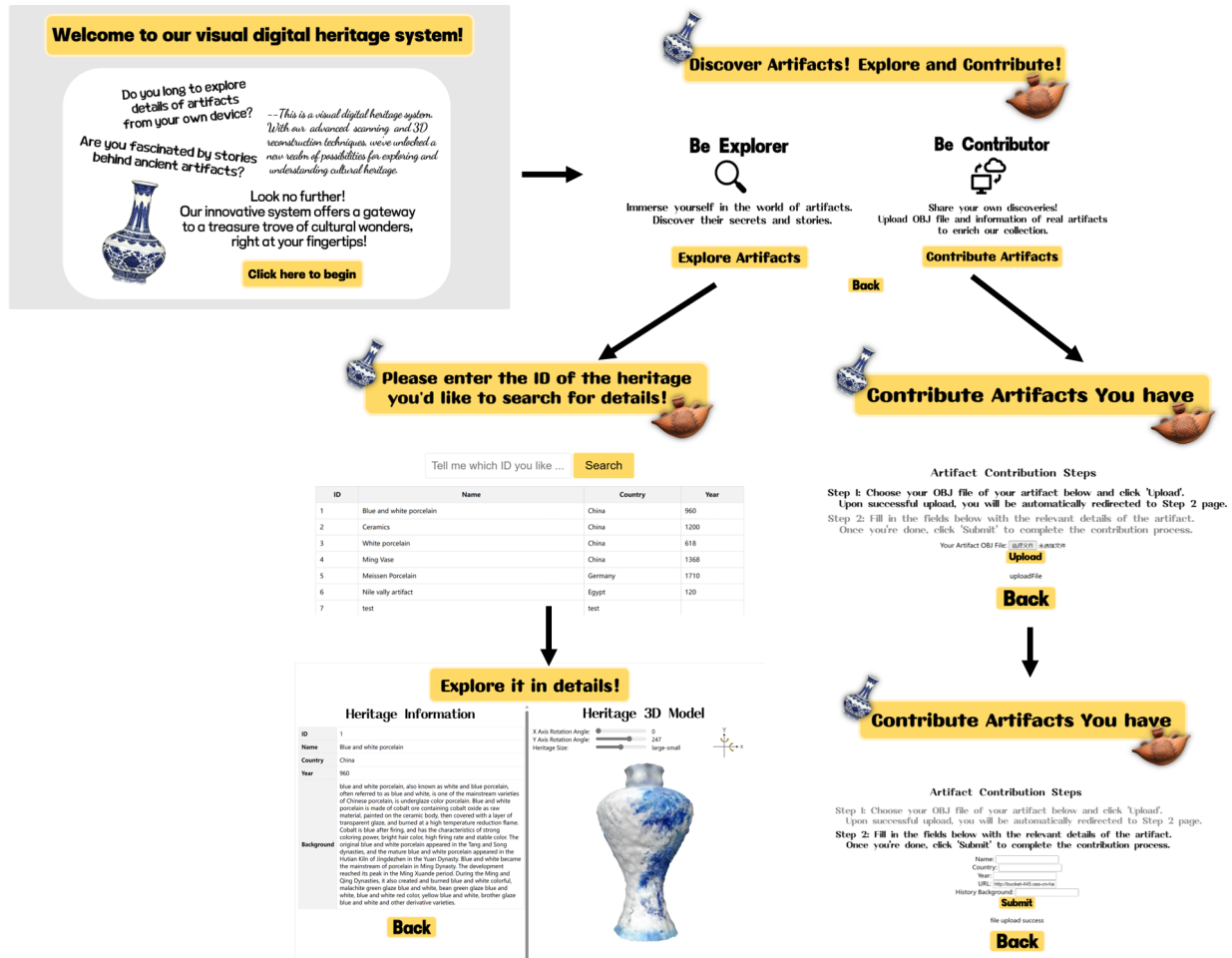


Figure 12: Website UI Structure

Handling and rendering OBJ files on the webpage involve several steps. Initially, the file is fetched using JavaScript's fetch function from a database URL. JavaScript then parses the file, extracting key data such as vertex positions and texture coordinates. This parsed data is structured for WebGL utilization to form the basis of the 3D model rendering. WebGL is prepared by creating a canvas element and acquiring context, storing parsed data in buffers for rendering. The camera's position, orientation, and projection matrix are established for scene projection. Interactive features are implemented, allowing users to dynamically manipulate the scene via sliders, controlling rotation angles and camera position. There are three sliders on the screen, including X axis rotation, Y axis rotation and heritage size. Here we set the model can rotate along x axis and y axis both for 360 degrees, step 0.1 and the size slider is used to zoom in and zoom out. Subsequently, WebGL draws the scene by passing vertex data, shader programs, and camera projection information to the GPU. Continuous updates ensure real-time responsiveness to user in-

teractions, providing a seamless and immersive experience. Figure 13 is a model example on the webpage.

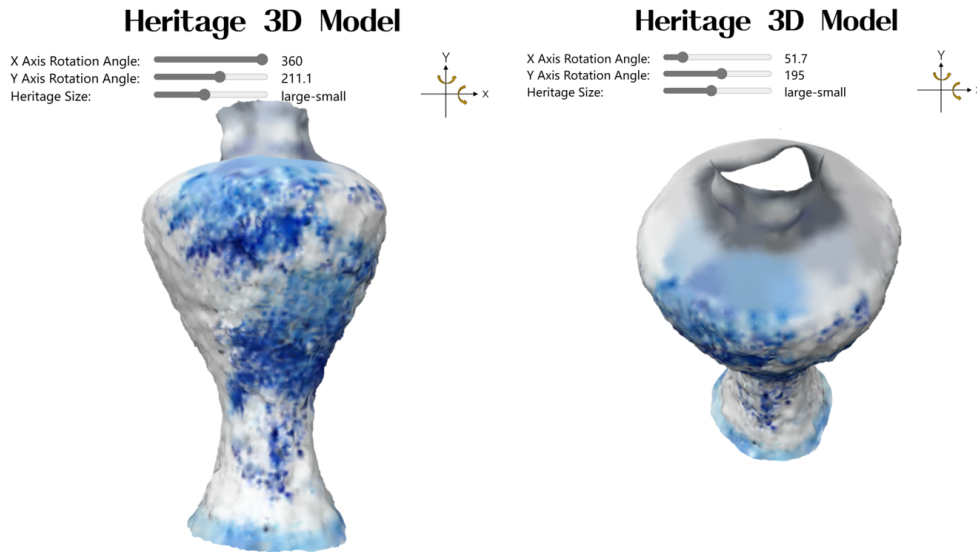


Figure 13: Website 3D Model Example

4 serves the purpose of parsing OBJ files, a standard format for representing 3D geometry, extracting essential data like vertex positions, texture coordinates, normals, and colors. It systematically processes each line of the OBJ file, identifying keywords (including v,vn,f,eta) and their corresponding arguments, and organizes the parsed data into suitable data structures for utilization in WebGL. Ultimately, it provides material libraries and geometries as output, facilitating the integration of 3D models into web applications for immersive user experiences.

Algorithm 4 Parsing OBJ file

Input: OBJ file text

Output: Material libraries and geometries

Initialize empty arrays for vertex positions, texture coordinates, normals, and colors Initialize empty arrays for WebGL vertex data and geometries Initialize arrays for material libraries and geometries Initialize default material, object, and groups

foreach *line in OBJ file* **do**

 Parse the line to extract keyword and arguments

if *keyword is a known OBJ keyword* **then**

 Call corresponding handler function with arguments

else

 Continue to the next line

end

end

Remove empty arrays from geometries

return *Material libraries and geometries*

5 orchestrates the process of rendering a 3D model on a canvas element using WebGL. It begins by setting up the WebGL context and compiling vertex and fragment shaders. After fetching and parsing the OBJ file containing geometry information, it processes each geometry and creates buffers for efficient rendering. Camera parameters are adjusted to fit the scene, and user interaction settings are defined. The function then enters a rendering loop, where it continuously updates the scene based on user input and renders each frame. This process ensures a dynamic and interactive 3D rendering experience.

Algorithm 5 Main function for rendering a 3D model

Input: Canvas element

Output: Rendered 3D model on the canvas

Obtain the Canvas container and set up the WebGL context If WebGL context is not available, exit the function Declare vertex and fragment shaders Compile and link the shaders, and find attribute and uniform locations Get the URL containing OBJ file information Fetch the OBJ file content via the URL Parse the OBJ file content and extract geometry information Process each geometry and create buffers Calculate the position and size of objects in the scene to fit the camera Initialize camera position and orientation Set camera view parameters Define user interaction parameters Enter the rendering loop **while** *rendering* **do**

 Update user interaction parameters Adjust Canvas size and viewport Enable depth testing and face culling Compute projection and view matrices Iterate over geometries, set uniform variables Draw geometries Request next frame rendering

end

2.2 Tolerance Analysis

In the design of our project, the precision control of the gimbal axes emerges as the paramount feature, governing the stability of the RGBD camera and consequently the fidelity of the scanned imagery. This control hinges on the performance of the MG996R servo motors, which operate within a voltage range of 4.8-6V and a stall current of 2.5A at the higher end of this spectrum. The motors are chosen for their ability to generate a torque ranging between 9.4 kg-cm at 4.8V and 11 kg-cm at 6V, with an allowable movement precision variance of $\pm 5\%$.

In a critical mathematical evaluation, if we consider the gimbal's stability to necessitate at least a 9 kg-cm torque, the MG996R's nominal torque at 4.8V sits precariously close to this threshold, particularly when factoring in a 5% tolerance which could result in an effective torque as low as 8.93 kg-cm. To mitigate this risk, a decision has been made to standardize the operating voltage at a slightly elevated level of 4.9V, thereby ensuring the motor's output remains above the stipulated requirement even at its tolerance limit.

Subsequent feasibility checks will entail empirical testing to measure the actual torque across a spectrum of voltages and loads, thus affirming compliance with our subsystem's stringent performance criteria. Should these tests reveal an inability of the MG996R motors to consistently deliver the requisite torque, we would be compelled to either source

motors with a more generous torque margin or to lighten the gimbal's load to accommodate the existing motors' capabilities.

This analysis consider the hardware system is tasked with executing precise and dependable 3D scans, specifically maintaining gimbal orientation control within a margin of ± 1 degree and achieving a response time shorter than 200 milliseconds, to support real-time processing. Our ambition is to attain a minimum scan resolution of 10 millimeters, optimizing the system to perform a full 360-degree scan in under five minutes, thereby harmonizing efficiency with the richness of data collected.

Mathematical Analysis and Solution:

Assuming the gyroscopic drift rate is approximately 0.01 degrees per second, over an extended scanning period of 5 minutes (300 seconds), the total potential drift could accumulate to 3 degrees. This deviation significantly exceeds the system's tolerance for precision control, which is within ± 1 degree.

To mitigate this risk and demonstrate the component's feasibility within our design constraints, a complementary filter combines the gyroscope and accelerometer data. The accelerometer provides long-term stability but with slower response times, while the gyroscope offers quick responses with short-term precision. By applying a complementary filter with a ratio of 98 % gyroscope data and 2 % accelerometer data, we can effectively correct for gyroscopic drift while maintaining real-time responsiveness.

For example, if the gyroscopic data drifts by 3 degrees over 300 seconds, the accelerometer's stable but slow-response data can correct this drift, reducing the effective error in orientation to within the ± 1 -degree requirement. This application of sensor fusion via a complementary filter ensures that our system maintains its high-level requirement for precision control, thus making the design feasible despite the inherent risk posed by gyroscopic drift.

This analysis underscores the importance of integrating both gyroscopic and accelerometer data to overcome the limitations of individual sensors, ensuring the Assistive Scanning Subsystem meets its high-level accuracy and efficiency requirements.

3 Cost and Schedule

3.1 Cost Analysis

3.1.1 Parts

To complete the cost analysis, we'll construct a table listing all the components along with their respective costs, and then sum these costs to find the grand total. Below is the table with the components provided.

Some information about manufacturer and Part # is not provided, actual details should be filled in when available.

Description with Manufacturer and Part #	Quantity	Unit Cost	Total Cost
#GM2804 Brushless Motor with AS5600 Encoder	3	¥103.9	¥311.7
42 Stepper Motor #S42H40D20	3	¥53	¥159
Arduino Nano Development Board	2	¥29.5	¥59
Servo #MG996R	4	¥18	¥72
#MPU6050 Module	2	¥35.7	¥71.4
Dupont Cables	1	¥16	¥16
Microsoft Xbox Kinect Sensor and Adapter	1	¥770	¥770
Alibaba Cloud Elastic Compute Service	1	¥162	¥162
Alibaba Cloud RDS MySQL	1	¥99	¥99
Alibaba Cloud Object Storage Service	1	¥4.98	¥4.98
Website Domain Name Registration	1	¥12	¥12
Total			¥1737.08

Table 6: Cost Analysis table listing all parts

The grand total for the cost of all components needed to complete the project is 1737.08 CNY. This includes all the hardware components such as motors, sensors, and development boards, as well as the necessary cables and connectors.

3.1.2 Labor

According to the 2023 China Undergraduate Employment Report^[3], the average monthly salary for undergraduates is 5990 CNY, or 34.56 CNY per hour. four of us in our group work an average of 2 hours a day, 5 days a week. Our weekly salary is $2 \times 5 \times 5 \times 34.56 = 1382.4$ CNY. For a total of nine weeks of work, our salary is $9 \times 1382.4 = 12441.6$ CNY.

3.1.3 Total

Based on the above analysis, we can get the total cost of the whole project is 14178.68 CNY.

3.2 Schedule

Table 7: Schedule

Date	Chuanrui Chen	Denglin Cheng	Qianyan Shen	Ziying Li
Week 1 (3.11 – 3.17)	Decide to use Vue and Visual Studio Code to design UI system.	Select and purchase modules, motors, and other system components.	Buy Kinect and connect it to your computer, install and learn to use the SDK	Discuss with professors and settle the structure of Database Subsystem.
Week 2 (3.18 – 3.24)	Learn based Html, CSS, JavaScript language and build the basic page of website	Design subsystem architecture and planned for future motor upgrades.	Get colour and depth information from Kinect and visualising it	Decide to use RDS, OSS, and ECS servers from Alibaba Cloud; ensure the environment deployment of a MySQL database and the object-based storage.
Week 3 (3.25 – 3.31)	Finish building the html page and try to build the front end structure.	Begin MPU6050 integration and code development for motion tracking.	Get the point cloud from the camera and save it locally	Ensure the functionality of RDS MySQL server and OSS server.
Week 4 (4.1 – 4.7)	Finish building the the front end structure.	Implement MG996R servo motor control and conducted initial tests.	Simple processing of point clouds and exploring registrations	Ensure the interaction between MySQL database instance and backend using cloud server.
Week 5 (4.8 – 4.14)	Make sure the webpage can communicate with database through API.	Calibrate sensors and fine-tune servo responses for accuracy.	Registration and post-processing of point clouds	Ensure the interaction between frontend and backend (stage 1).

Continued on next page

Table 7 – continued from previous page

Date	Chuanrui Chen	Denglin Cheng	Qianyan Shen	Ziying Li
Week 6 (4.15 – 4.21)	Make sure 3D model can be displayed on the webpage smoothly.	Assemble prototype; test MPU6050 and servo interaction.	Explore reconstruction algorithms to reconstruct point clouds	Ensure the interaction between frontend and backend (stage 2).
Week 7 (4.22 – 4.28)	Decorate the webpage and add more functions.	Analyze, optimize performance, and begin stress testing.	Interface with Assistive Scanning Subsystem to organise the whole process	Make sure the functionality of the whole subsystem.
Week 8 (4.29 – 5.5)	Interface with 3D Reconstruction Subsystem and Database Subsystem for the whole process.	Initiate stepper or brushless motor integration and testing.	Interface with Database Subsystem and Interactive Interface Subsystem.	Interface with 3D Reconstruction Subsystem and Interactive Interface Subsystem.
Week 9 (5.6 – 5.12)	Organize and prepare the Final Demo and Report.	Conduct comprehensive tests and prepare the Final Demo and Report.	Organize and prepare the Final Demo and Report.	Organize and prepare the Final Demo and Report.

4 Conclusion

4.1 Accomplishment

The 3D reconstruction subsystem should be able to convert point clouds with more than 10,000 points into OBJ format. The following are the vertex counts of each reconstructed OBJ file, which represents the number of points in the original point cloud before reconstruction:

Table 8: The number of points in meshed point clouds

Stone Ornament	Chair	Cup	Pen Container	Vase
69132	178630	15171	13498	96944

The following is one of the results of the reconstruction, where the system succeeded in obtaining relatively smooth models:



Figure 14: Meshed Model of chair from different viewpoints

The model obtained from the reconstruction of this subsystem should have a dimensional error of no more than 5%. Below is the height error analysis of the four models, the average error is 2.85% which is within the target range:

Table 9: Height error analysis of the Models

Model Name	Stone Ornament	Cup	Pen Container	Vase
Measured Height in Model (cm)	23.86	10.16	9.50	26.21
Real Height (cm)	24.90	10.10	9.60	24.80
Errors (%)	4.17	0.59	1.04	5.60

4.2 Uncertainties

The 3D reconstruction subsystem is faster and more effective when doing large, more distinctive structural features, simple colors, and less reflective objects. Such as antique furniture, large sculptures, etc. For objects with fine textures, such as colored glaze vases, details need to be obtained through a lot of repeated sampling. For objects with detailed texture engraving, the detail requirements cannot be met. For objects that may cause

strong specular reflection, such as ceramic mugs, holes are created. For objects that produce weak specular reflection, color distortion is produced. Also the color will react to changes in lighting and will not accurately get the true color of the object.

4.3 Future Work

For the 3D reconstruction subsystem, in the future I think real-time should be taken into account. The current reconstruction relies on a lot of manual labor and parameter matching. For the problem of needing to manually rotate the object for scanning, a motorized rotating turntable can be incorporated, and at the same time, the rotating speed of the turntable and the acquisition speed of the scanning algorithm need to be matched to achieve automated scanning and reconstruction. For parameter matching, firstly, user interaction can be added to the simple UI platform currently built, such as drop-down boxes and text boxes to fill in numbers. Secondly, the development of adaptive algorithms can be carried out to adopt different parameters at point clouds with different sparsity. For the problem of manually selecting feature points for alignment, image feature recognition can be added to select parts with distinctive matching features as the object of alignment.

4.4 Ethic Consideration

4.4.1 Ethics

In the development and implementation of our project, we face several ethical concerns that are integral to ensuring the integrity and societal value of our device. Given the cultural diversity of our potential user base, it is critical that we address the cultural sensitivities associated with the scents emitted by our hardware devices. Aligning with the IEEE Code of Ethics, particularly term 3, we commit to "avoid real or perceived conflicts of interest whenever possible, and to disclose them to affected parties when they do exist" [4]. This commitment underlines our dedication to cultural respect and awareness, ensuring that our scent cues are designed to be non-offensive and appropriate across different cultural contexts. Moreover, we are mindful of the potential for misuse of emerging technologies, including our own, which can lead to unintended negative consequences. Therefore, we adhere to term 6 of the IEEE Code of Ethics, obliging us to "maintain and improve our technical competence and to undertake technological tasks for others only if qualified by training or experience, or after full disclosure of pertinent limitations" [4]. This ensures that our technology is only distributed to organizations and companies holding valid and legal certifications, minimizing the risk of misuse.

Meanwhile, in line with ACM and IEEE ethics, we will maintain transparency in our operations and communications. As stipulated by the ACM Ethics guidelines, term 3, we will "Be honest and trustworthy," [5] providing full disclosure of system capabilities and limitations. Similarly, the IEEE Ethics guidelines, term 5, demands that we "seek, accept, and offer honest criticism of technical work, to acknowledge and correct errors" [4].

Furthermore, we are committed to fostering a positive and non-violent virtual environ-

ment, in adherence to the ACM Code of Ethics and Professional Conduct, which advocates against causing "unjustified physical or mental injury, unjustified destruction or disclosure of information, and unjustified damage to property, reputation, and the environment" [5]. This principle guides us to exclude any violent or bloody scenes from our content, recognizing the potential harm they could inflict, particularly on children. Our ethical framework is not merely a set of guidelines but a core component of our project's philosophy and operational strategy.

Finally, by following both the IEEE Ethics guidelines [4] and ACM Ethics guidelines [5], we aim to ensure that our technology not only advances in its technical capabilities but also contributes positively to society and operates within the highest standards of ethical conduct.

4.4.2 Safety

Ensuring the safety of both operators and subjects during the use of our 3D scanning system is paramount. We have identified several areas of focus to mitigate potential risks:

- **Electrical Safety:** Given our system's reliance on electronic components, we implement measures to prevent electrical shocks and hazards, ensuring all components are properly insulated and comply with relevant safety standards.
- **Mechanical Safety:** Our system includes moving parts; thus, we ensure these components are securely enclosed to prevent accidental injuries during operation.
- **Environmental Safety:** In our project's design and implementation, we consider the environmental sustainability and impact, respecting ACM's directive to "promote environmental sustainability," [5] and IEEE's instruction to "strive to comply with ethical design and sustainable development practices" [4]. Therefore, the use of materials, such as lithium batteries, requires careful handling and disposal to prevent environmental harm. Utilizing biodegradable and recyclable PLA materials for 3D printed components, promoting sustainable and eco-friendly practices. We adhere to guidelines for the safe use and disposal of hazardous materials.
- **Data Security:** To protect sensitive information collected during scanning, we employ robust security measures to prevent data breaches and unauthorized access.
- **User Training:** Comprehensive training for all users is essential to safely operate the 3D scanner, emphasizing awareness of potential hazards and adherence to established safety protocols.

By addressing these ethical and safety concerns, our project aims to not only advance technological capabilities but also ensure our work benefits society responsibly and safely.

5 Appendix

Our system have specific requirements regarding the model and configuration of the computer, as follows:

- Operating System: Windows 10
- Integrated Development Environment (IDE): Visual Studio 2019
- Libraries: PCL (Point Cloud Library) version 1.12.1; OpenCV version 4.5.5; Kinect SDK 2

References

- [1] W. Li, "Application of virtual reality technology in the inheritance of cultural heritage," in *Journal of Physics: Conference Series*, IOP Publishing, vol. 1087, 2018, p. 062 057.
- [2] J. Sun and H. Kim, "Digital display design of historical relics—using artistic projection of historical relics as an example," *TECHART: Journal of Arts and Imaging Science*, vol. 9, no. 1, pp. 35–48, 2022.
- [3] Wang Boqing, Wang Mengping, *2023 China Undergraduate Employment Report*. Beijing: Social Sciences Academic Press, 2023.
- [4] IEEE. "IEEE Code of Ethics." (2016), [Online]. Available: <https://www.ieee.org/about/corporate/governance/p7-8.html> (visited on 03/27/2024).
- [5] ACM. "ACM Code of Ethics and Professional Conduct." (2018), [Online]. Available: <https://www.acm.org/code-of-ethics> (visited on 03/27/2024).