ECE 445

SENIOR DESIGN LABORATORY

DESIGN DOCUMENT

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

March 27, 2024

Contents

1 Introduction

1.1 Problem and Solution Overview

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. Additionally, for history enthusiasts and researchers worldwide, detailed information about specific artifacts is not readily accessible. Therefore, the conservation and exhibition methods of cultural relics have been a common concern of scholars around the world.

Traditional photographs often fail to capture the intricate details of artifacts, hampering comprehensive research and preservation efforts. Furthermore, the absence of user-friendly interactive interfaces limits the interaction between enthusiasts and artifacts, impeding immersive experiences in a virtual exploration of cultural heritage. 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.

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.[1], [2]

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. We employ the point cloud reconstruction method, which involves converting RGBD images into point clouds, applying denoising and filtering operations to the point clouds, and finally reconstructing the processed point clouds to obtain a smooth 3D model.

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.

1.2 Visual Aid

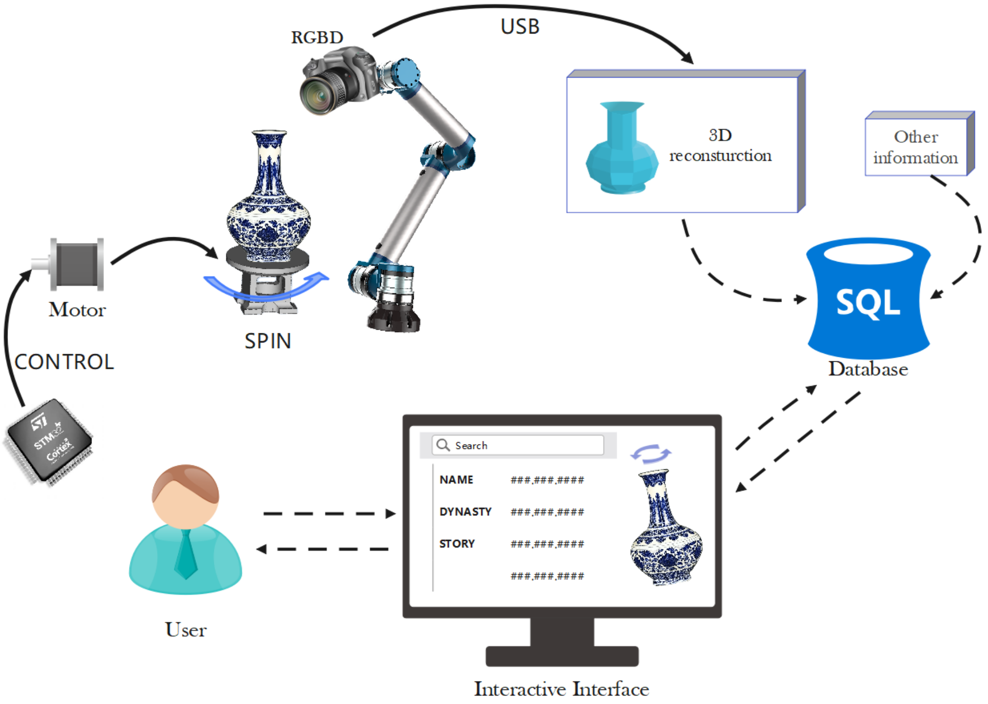

Figure 1: Visual Aid

1.3 High Level Requirements List

- **Modeling Outcomes:** We are capable of converting 920x1080 color images and 512x424 depth maps into point clouds. Our system can achieve the reconstruction of point clouds into 3D models. Additionally, we offer the ability to save and export the generated models in various formats.
- **User Interface and User Experience:** 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 in the search bar.
- **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.

2 Design

2.1 Block Diagram

Our system consists of four subsystems: the scanning subsystem, 3D reconstruction subsystem, database subsystem, and interactive interface subsystem.

The 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.

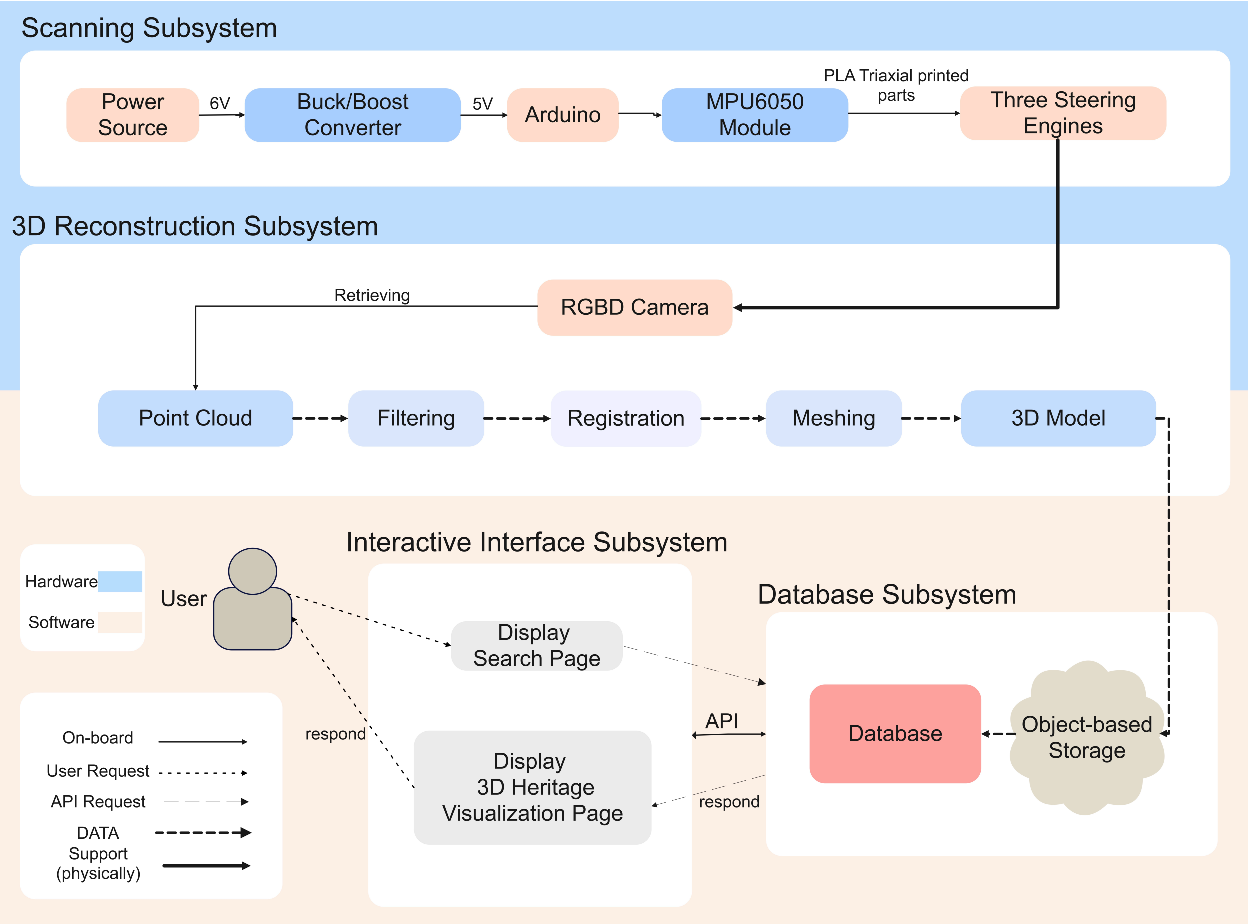

Figure 2: Block Diagram

2.2 Physical Design

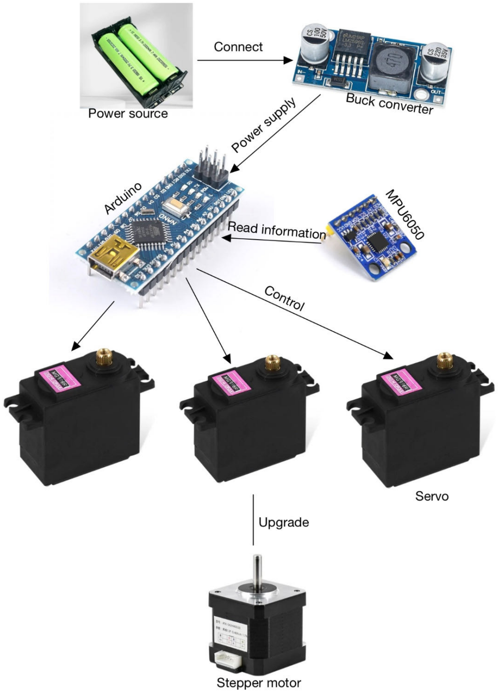

Figure 3: Physical Design

The system begins with a power source composed of batteries. The regulated power from the buck converter supplies an Arduino board, which acts as the main control of the operation. The Arduino is responsible for processing data and orchestrating the actions of the connected components. It reads information input from the MPU6050 module, a motion-tracking device that likely provides real-time spatial information, such as orientation and acceleration data, which is crucial for accurate 3D scanning. From the Arduino, the flow of control splits towards an array of servos, which are used to manipulate the scanning apparatus, providing precise movements and adjustments. The three motors shown will be mounted at the end of each of the three axes to accomplish three-axis control. This level of control is essential for detailed scanning and model reconstruction. The diagram also indicates an upgrade path to stepper motors, suggesting that in future iterations, the servos might be replaced for even greater precision and control over the scanning process.

2.3 Subsystem Overview

2.3.1 Scanning Subsystem

The scanning subsystem is designed as a hardware component of the overall project, with the main objective 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.

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.

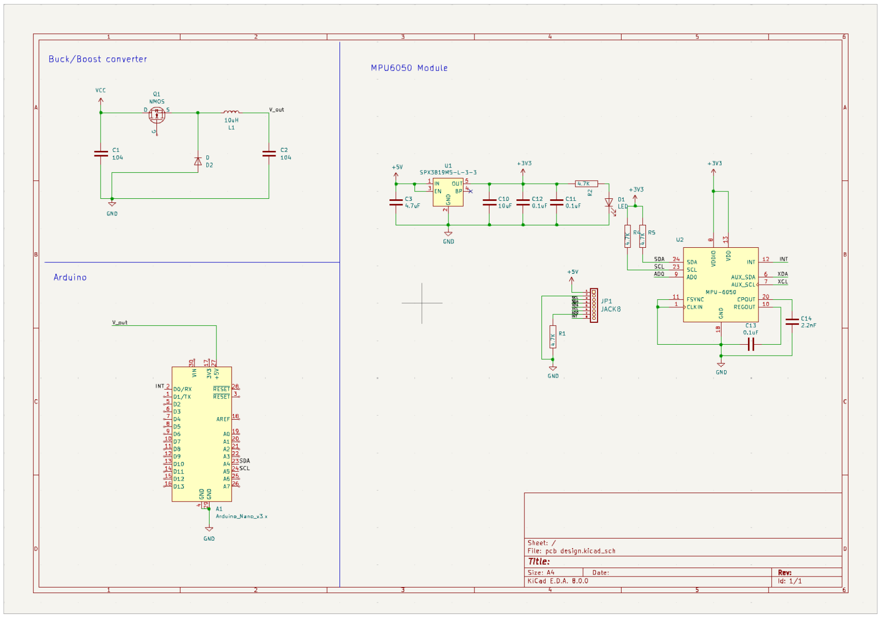

Figure 4: PCB Design

	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 scanningh 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 Scanning Subsystem

2.3.2 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 scanning subsystem. 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 PCL, and set various saving modes, such as pressing the space bar to save and saving at fixed time intervals, to help to obtain point clouds with different angles according to their individual needs, on top of better obtaining point clouds with fixed angles by using a turntable.

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
    Extract point cloud data from DepthImage and ColorImage
    for each pixel $(x, y)$ do
        Get depth value, $depth$, and color value, $color$, from DepthImage and ColorImage
        Convert pixel coordinates $(x, y)$ to 3D coordinates $(X, Y, Z)$

        if depth value 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
```

After obtaining the initial point cloud, we first need to downsample the point cloud.The number of point clouds obtained by Kinect is huge, in the hundreds of thousands. Performing point cloud operations on such a volume of data would take a lot of time, so we need to reduce the complexity by obtaining a new point cloud that has fewer points but is equivalent to the previous one. We achieve this by performing a voxelisation operation[3]. Subsequently, we have to segment the downsampled point cloud to separate the point cloud of the object we need from the background. Specifically, we need to filter out points within a specified region under a certain dimension by setting[4]. We then need to filter our target point cloud more finely: scanning inevitably produces outlier points, which complicates the operations of estimating local point cloud features, and also leads to erroneous values that make post-processing such as point cloud registration fail. We perform denoising by statistically analysing the neighbourhood of each point and pruning out points that do not meet the criteria[5]. Finally we smooth the point cloud using the Moving Least Squares (MLS) algorithm to cope with problems such as holes and unsmoothness on the model surface that may result from inaccurate sampling[6].

Algorithm 2 PCL Point Cloud Passthrough(as an example)

```
Input: Point cloud data
Output: Filtered point cloud data
Create a filtered point cloud container, $FilteredPointCloud$
for each point, Point, in PointCloud do
    Extract $(X, Y, Z)$ and $color$ from Point
    if Passes passthrough filter then
        Create a filtered point and assign $(X, Y, Z)$ and $color$ to $FilteredPoint$
         Add $FilteredPoint$ to $FilteredPointCloud$
    end
end
return FilteredPointCloud
```

For the obtained point cloud data, we need to process the point cloud registration. The ICP[7] algorithm is mainly used to maximise the overlap between point clouds by finding the best rotation and translation values. In this way, the point clouds scanned from different angles can be effectively spliced. For the spliced point cloud, we need to perform filtering, smoothing and outliers removal again to get a smoother point cloud.

Algorithm 3 Iterative Closest Point (ICP) algorithm

```
Input: Source point cloud A, Target point cloud B, an initial transformation matrix
Output: Transformation matrix
$T \longleftarrow T_0$
 while $notconverged$ do
    for $i \longleftarrow 1$ to $N$ do
        $m_i \longleftarrow findClosestPointInA(T \cdot b_i)$
         $w_i \leftarrow 0$
         if $\|m_i - T \cdot b_i\| \leq d_{\max}$ then
             $w_i \leftarrow 1$
        end
    end
    $T \leftarrow \underset{T}{\operatorname{argmin}} \left\{ \sum_i w_i \|T \cdot b_i - m_i\|^2 \right\}$
end
return $T$
```

Finally, we will use a variety of reconstruction algorithms, including Greedy Projection Triangulation, Poisson Surface Reconstruction[8], etc., to provide a wealth of options for different scenes. The aim is to obtain the best possible 3D model. The obtained 3D model will be exported in OBJ format, stored in a database and visualised in Interactive Interface Subsyste for user interaction.

	Requirement Description	Verification Procedure
1	The inputs to the scanning subsystem to this subsystem are $\mathbf{920 * 1080}$ colour images and $\mathbf{512 * 424}$ 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 of order of magnitude **10,000** or higher.	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

2.3.3 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. 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. 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.

In addition, for the creation and management of running virtual server instances, we also utilized Elastic Compute Service (ECS). 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.

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:

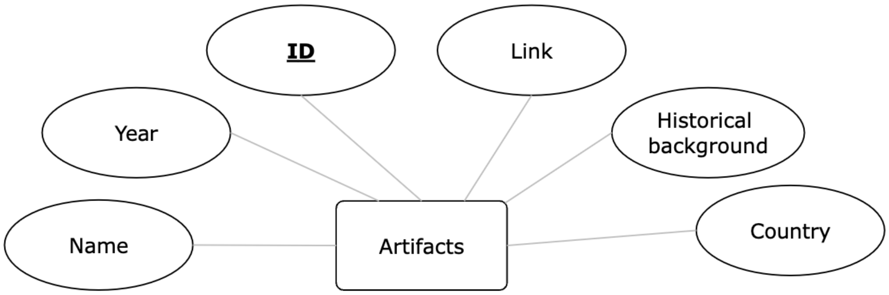

Figure 5: Entity-Relationship Diagram

	Requirement Description	Verification Procedure
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.
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.
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.
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.

Table 3: Requirements and Verifications table for Database Subsystem

2.3.4 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 React framework combined with WebGL technology to achieve graphical rendering and web development.

The React 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 4: Requirements and Verifications table for Interactive Interface Subsystem

2.4 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 ±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 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.

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
Total			**¥1725.08**

Table 5: Cost Analysis table listing all parts

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

The grand total for the cost of all components needed to complete the project is 1725.08 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[9], 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.

3.2 Schedule

Table 6: 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.

Continued on next page

Table 6 – continued from previous page

Date	Chuanrui Chen	Denglin Cheng	Qianyan Shen	Ziying Li
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).
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 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 Ethics and Safety

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” [10]. 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” [10]. 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,”[11] 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”[10] .

Furthermore, we are committed to fostering a positive and non-violent virtual environment, 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” [11]. 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 [10] and ACM Ethics guidelines [11] , 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.

4.2 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,"[11]and IEEE's instruction to " strive to comply with ethical design and sustainable development practices"[10]. 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] Point Clouds. “Voxel Grid Downsampling.” (), [Online]. Available: https://pcl.readthedocs.io/projects/tutorials/en/latest/voxel_grid.html (visited on 03/27/2024).

[4] Point Clouds. “PassThrough Filter.” (), [Online]. Available: https://pcl.readthedocs.io/projects/tutorials/en/latest/passthrough.html (visited on 03/27/2024).

[5] Point Clouds. “Statistical Outlier Removal.” (), [Online]. Available: https://pcl.readthedocs.io/projects/tutorials/en/latest/statistical_outlier.html (visited on 03/27/2024).

[6] Point Clouds. “Smoothing and normal estimation based on polynomial reconstruction.” (), [Online]. Available: https://pcl.readthedocs.io/projects/tutorials/en/latest/resampling.html#smoothing-and-normal-estimation-based-on-polynomial-reconstruction (visited on 03/27/2024).

[7] P. J. Besl and N. D. McKay, “Method for registration of 3-d shapes,” in *Sensor fusion IV: control paradigms and data structures*, Spie, vol. 1611, 1992, pp. 586–606.

[8] M. Kazhdan, M. Bolitho, and H. Hoppe, “Poisson surface reconstruction,” in *Proceedings of the fourth Eurographics symposium on Geometry processing*, vol. 7, 2006.

[9] Wang Boqing, Wang Mengping, *2023 China Undergraduate Employment Report*. Beijing: Social Sciences Academic Press, 2023.

[10] IEEE. “IEEE Code of Ethics.” (2016), [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html (visited on 03/27/2024).

[11] ACM. “ACM Code of Ethics and Professional Conduct.” (2018), [Online]. Available: https://www.acm.org/code-of-ethics (visited on 03/27/2024).