

ECE 445  
SENIOR DESIGN LABORATORY  
FINAL REPORT

---

# AR Sandbox

---

**Team #3**

HAOZE GAO (haozeg2@illinois.edu)

HAOWEN ZHENG  
(haowenz5@illinois.edu)

QIRAN PAN (qiranp2@illinois.edu)

YIHENG ZHANG  
(yihengz5@illinois.edu)

Advisor: Timothy Lee

May 7, 2024

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Problem Statement . . . . .	1
1.2	Solution Overview . . . . .	1
1.3	High-Level Requirements . . . . .	2
1.4	Block Diagram . . . . .	2
<b>2</b>	<b>Design</b>	<b>4</b>
2.1	Sensor Subsystem . . . . .	4
2.2	Processing Subsystem . . . . .	4
2.2.1	Processor Selection . . . . .	4
2.2.2	IR Markers for Sandbox Boundary Detection . . . . .	5
2.2.3	Display – Generating Colored Contours from Depth Information . . . . .	7
2.2.4	Concurrent Handling of Web and Display Using Multithreading . . . . .	8
2.3	Projection Subsystem . . . . .	9
2.3.1	Projector Selection . . . . .	9
2.3.2	Projector Auto Focusing . . . . .	9
2.3.3	Projector-Camera System Calibration . . . . .	13
2.4	Powering Subsystem . . . . .	19
2.5	Database and Wireless Display Subsystem . . . . .	20
2.6	Structure Subsystem . . . . .	24
2.6.1	The Main Part of Sandbox . . . . .	24
2.6.2	The Separable Part . . . . .	25
2.6.3	The Recovery Subsystem . . . . .	26
2.6.4	The Mobility System . . . . .	26
<b>3</b>	<b>Verification</b>	<b>28</b>
3.1	Sensor Subsystem . . . . .	28
3.2	Projection Subsystem . . . . .	28
3.2.1	Auto Focusing . . . . .	28
3.2.2	Projector-Camera System Calibration . . . . .	28
3.3	Database and Wireless Display Subsystem . . . . .	30
3.4	Structure Subsystem . . . . .	30
<b>4</b>	<b>Costs</b>	<b>34</b>
4.1	Labor . . . . .	34
4.2	Components . . . . .	34
<b>5</b>	<b>Ethics and Safety</b>	<b>36</b>
5.1	Ethical Issues . . . . .	36
5.2	Safety Concerns . . . . .	36
<b>6</b>	<b>Conclusion</b>	<b>38</b>
	<b>References</b>	<b>39</b>

<b>Appendix A</b>	<b>Requirements and Verification Table</b>	<b>41</b>
<b>Appendix B</b>	<b>PCB Design of Stepper Motor Drive Board</b>	<b>44</b>
<b>Appendix C</b>	<b>Pseudocode for Motor Control</b>	<b>45</b>
<b>Appendix D</b>	<b>Code for Converting Camera's Pixel Coordinates to World Coordinates</b>	<b>46</b>
<b>Appendix E</b>	<b>Code for Converting World Coordinates to Projector's Pixel Coordinates</b>	<b>47</b>
<b>Appendix F</b>	<b>Simulation Results of the Structure Subsystem</b>	<b>48</b>

# 1 Introduction

## 1.1 Problem Statement

We are introducing a smart sandbox with augmented reality (AR) capabilities that projects contour maps in real-time onto the sand surface, making geography education for children not only informative but also significantly more enjoyable. Currently available educational sandboxes [1] [2] are mostly cumbersome and limited to public spaces like activity centers rather than serving as personalized learning tools. For AR sandboxes which are able to project contours onto the sandbox, problems emerge. A high lagging time can be noticed when sand is manipulated manually causing contours being recalculated and re-projected onto the terrain. Meanwhile, contours are flickering on the terrain parts that remain still for the whole time, showing heavy amount of computational resources are wasted on the unnecessary parts thus restricting more delicate and faster signal processing procedure on the terrain parts that need to be updated as soon as possible. Furthermore, the existing projectors designed for sandboxes exhibit primitive features, characterized by a notably low refresh rate and harsh direct light that arise safety concerns for children's eyesight.

We introduce a smart sandbox with AR capabilities that projects contour maps in real-time onto the sand surface, making geography education for children not only informative but also significantly more enjoyable. We are committed to addressing these drawbacks and are working towards the development of a new and improved AR sandbox. This innovative solution aims to overcome the limitations of bulkiness, offering a more accessible and personal learning experience. Additionally, we are focused on enhancing the AR functionality to deliver a smoother experience with higher refresh rates and reduced glare, ensuring a more comfortable and engaging educational tool for children.

## 1.2 Solution Overview

We would develop a next-generation sandbox with AR projection and interaction capabilities. Basic criteria for our sandbox are: safe, easy to use and high refresh rate. The overall structure will be designed to be portable while ensuring both high load-bearing capacity and stability. The AR module, including depth camera, processing unit and projector, can be removed from the sandbox and installed on other sand tables. In general, our project will ensure a latency of less than 1 s, and the contours can be accessed wirelessly from portable mobile devices. The projector can be auto focused to ensure display clarity when projector's installation is changed to another sand table or the height of sandbox is altered. The depth camera will be used to detect the height of the sand and collect RGB and IR image for excluding human hands from the height change. The projector will be high luminance to display clearly on sand and short throw to match the camera's field of view and lower the center of mass for safety. The AR module will interact with people through the projected screen or mobile devices.

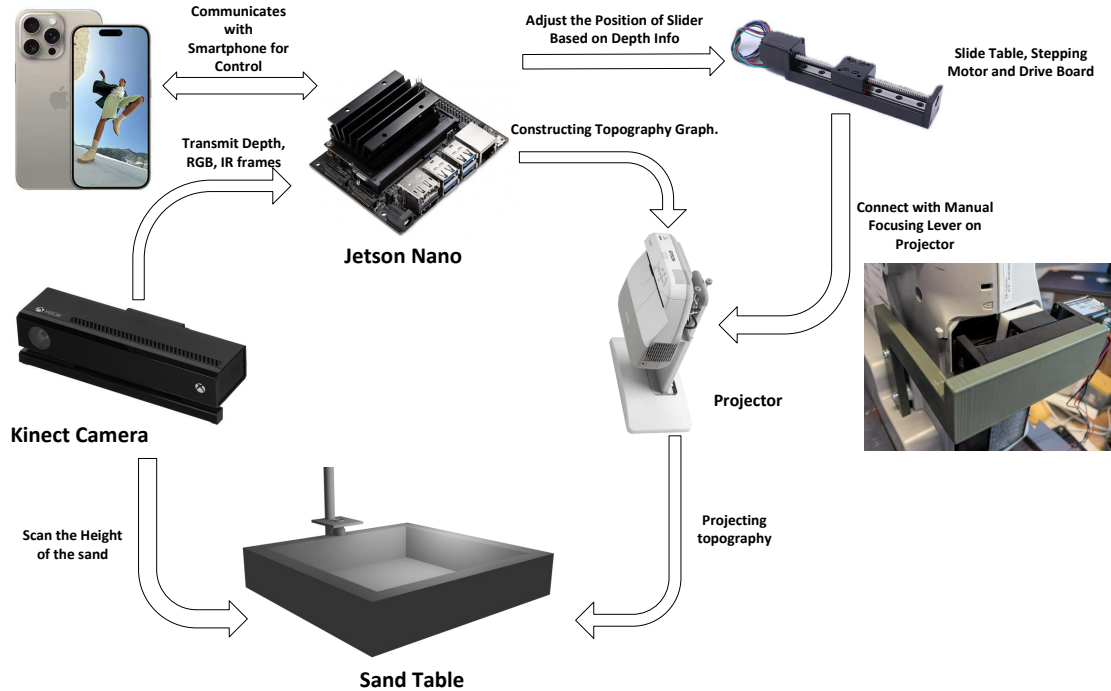


Figure 1: Visual Aid for AR Sandbox Project

### 1.3 High-Level Requirements

- The AR module must have a latency less than 1 s when projecting the contour map onto the sand surface.
- The AR module must be able to correctly exclude human hands from the height change of sand.
- The AR module must be easy to use, pushing one button to start and interact with people through the projected screen directly or a webpage.
- The sandbox must be portable and foldable for easy transportation and storage.
- The AR module must be able to adapt to different kinds of sand and sand table.

### 1.4 Block Diagram

The AR sandbox project consists of six subsystems: sensor, processing, projection, powering, structure and wireless display. The sensor subsystem is responsible for acquiring depth information from the sand surface. The processing subsystem is responsible for processing the depth information and projecting the correct topography graph onto the sand. The projection subsystem is responsible for projecting the topography map onto the sand surface. The powering subsystem is responsible for providing power to all subsystems. The structure subsystem is responsible for storing the sand and connecting to all the other parts as a whole AR module. The wireless display subsystem provides a webpage to wirelessly view realtime contours and control the sandbox. The block diagram

for the AR sandbox project is shown in Figure 2.

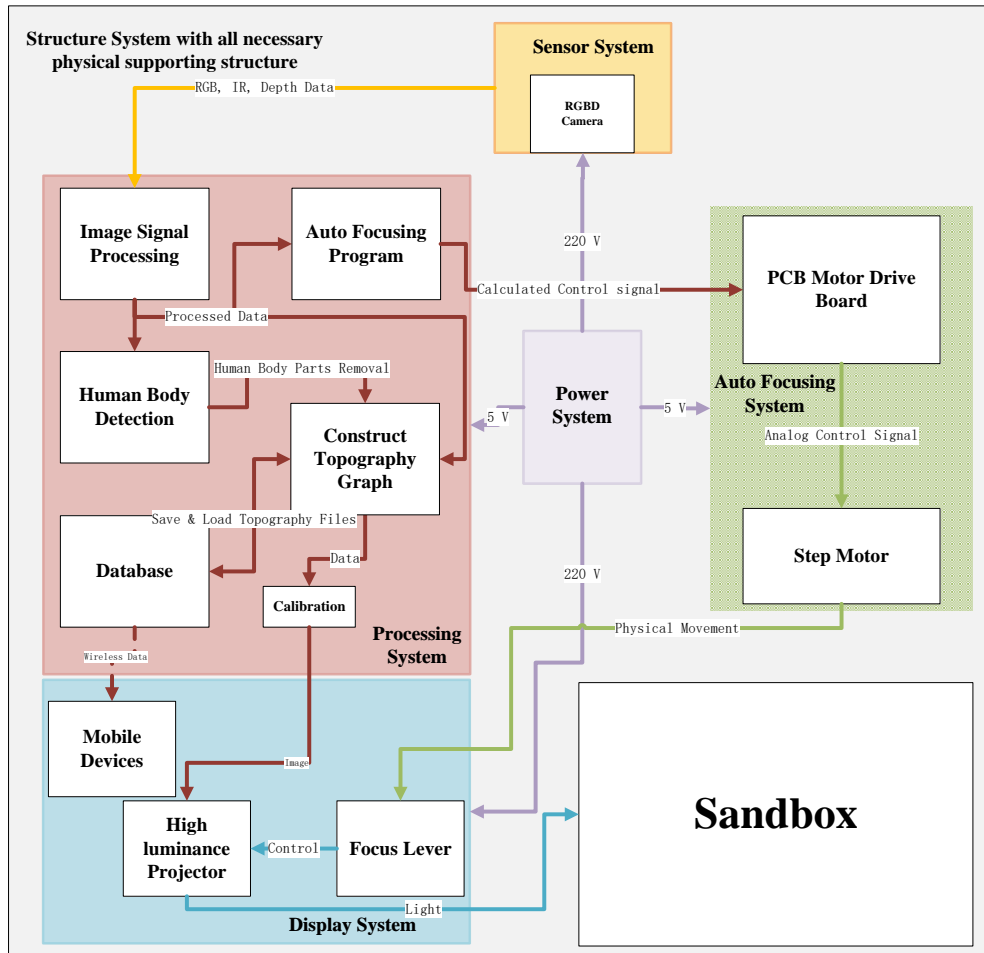


Figure 2: Block Diagram for AR Sandbox Project

## 2 Design

### 2.1 Sensor Subsystem

The sensor subsystem in the AR sandbox project refers to the RGB-D camera that is responsible for acquiring depth information from the sand surface and collecting RGB and IR image for excluding human hands interference and detecting the shape of the sandbox.

For depth information acquisition, there are two main categories of sensors: pressure sensors and depth cameras. Pressure sensors provide a simple and cost-effective solution for measuring the height of the sand. They can be easily integrated into the sandbox structure and are not affected by ambient light conditions. However, pressure sensors have limited accuracy and resolution and are prone to human hands putting on sand, which would largely affect the quality of the projected topography map. On the other hand, depth cameras, such as the Intel RealSense D435 or Microsoft Kinect, provide a more accurate and detailed depth map of the sand surface. They can capture RGB and IR images simultaneously, which is additional information for excluding human hands interference. Although depth cameras are more expensive and need to be installed overhead, they are more suitable for our project due to their high accuracy, resolution and abilities to adapt to different kinds of sand and sand tables and provide RGB and IR images for excluding human hands interference. Therefore, we choose the Microsoft Kinect V2 as the sensor for our AR sandbox project.

Microsoft Kinect V2 is capable of providing depth information in low light levels, providing RGB and IR frames and is color and texture invariant. It has a color sensor with a  $1920 \times 1080$  pixels resolution and a depth sensor with a  $512 \times 424$  pixels resolution. The operational area is delineated by a depth span from 0.5 to 4.5 meters, with a viewing angle of  $70^\circ$  horizontally and  $60^\circ$  vertically [3]. When detecting objects at depth range from 0.5 m to 2 m, the average depth accuracy error is less than 4 mm according to Yang et al. [4]. The Kinect V2 camera is connected to the processing unit through a USB 3.0 cable.

The sensor communicates with the processing unit through USB protocol. We use an open source library called "libfreenect2" [5] to access the RGB, IR and depth information from the Kinect V2 camera.

### 2.2 Processing Subsystem

#### 2.2.1 Processor Selection

The chosen processor for our project is the Jetson Nano, a compact yet powerful computer designed specifically for embedded applications and AI IoT. The Jetson Nano delivers the performance and capabilities required to run modern AI workloads efficiently. It offers a swift and straightforward method for integrating advanced AI into future products.

**Reason for Selection** The primary consideration for selecting the Jetson Nano was its balance of cost and performance. Our project involves considerable computer vision (CV)



Figure 3: Jetson Nano Board

tasks, which require robust GPU support. The specifications of the Jetson Nano are as follows:

- GPU: NVIDIA Maxwell architecture with 128 NVIDIA CUDA® cores
- CPU: Quad-core ARM Cortex-A57 MPCore processor
- Memory: 16 GB
- Operating System: Built-in Linux, facilitating development
- Price: 1199 RMB

These features, combined with the price, provide a compelling option for our needs.

**Additional Functionalities** The Jetson Nano board includes a developer kit pre-installed with essential libraries for CV applications such as CUDA, cuDNN, TensorRT, and OpenCV, which assist in the development of visual processing and display features. Furthermore, the board is equipped with GPIO pins that can output PWM signals. These GPIO pins can be utilized to connect with a PCB controlling motors, enabling an autofocus functionality that is vital for our project's objectives.

## 2.2.2 IR Markers for Sandbox Boundary Detection

**Introduction to the Approach** Initially, to display the contours precisely on the sandbox, my method involved using advanced object detection techniques. Specifically, we considered employing a machine learning model, such as the YOLO (You Only Look Once) model, to detect and define the bounding region of the sandbox. This approach,



while sophisticated, proved to be computationally expensive and somewhat inefficient, particularly as it utilized substantial processing resources just to delineate the sandbox edges.

To achieve robustness and scalability, adaptable to sandboxes of various shapes, a more direct and resource-efficient method was devised. Utilizing the Kinect v2 camera's ability to capture infrared (IR) information, infrared reflective markers were selected for this task. These markers were strategically placed at each corner of the sandbox, enabling the camera to clearly capture these markers with minimal interference. The markers are easily distinguishable from other objects in the IR spectrum, which facilitates the use of OpenCV to identify these markers and ultimately delineate the sandbox boundaries.

After successfully detecting the sandbox boundaries, a region of interest matrix (*roi\_mat*) is established, where the sandbox area is represented as 1, and other areas are set to 0. This matrix is then used to filter the processed colored contours, multiplying it with the *roi\_mat* to isolate and display only the relevant sandbox area in the final image. This additional step ensures that only the specific area of interest is highlighted, enhancing the visual output's relevance and clarity for further analysis or display purposes.

---

**Algorithm 1** Locate Sandbox Edge Pseudocode

---

```
1: function LOCATESANDBOXEDGE(img)
2:   Apply histogram equalization to the input image
3:   img_hist ← CV2.EQUALIZEHIST(img)
4:   Apply a binary threshold to the image
5:   binary ← CV2.THRESHOLD(img_hist, 250, 255, cv2.THRESH_BINARY)
6:   Initialize parameters for blob detection
7:   params ← CV2.SIMPLEBLOBDETECTOR_PARAMS
8:   params.blobColor ← 255
9:   params.filterByArea ← True
10:  params.filterByCircularity ← True
11:  Create a blob detector with specified parameters
12:  detector ← CV2.SIMPLEBLOBDETECTOR_CREATE(params)
13:  Detect blobs in the binary image
14:  keypoints ← DETECTOR.DETECT(binary)
15:  if LEN(points) ; 3 then
16:    Print "No sandbox detected"
17:    return im_with_keypoints, points, binary
18:  end if
19:  Compute the convex hull of points
20:  hull ← CV2.CONVEXHULL(points)
21:  Draw the convex hull as the contour on the image
22:  CV2.DRAWCONTOURS(im_with_keypoints, [hull], 0, (0, 255, 0), 3)
23:  return im_with_keypoints, points
24: end function
```

---

**Pseudocode for IR Marker Detection and Boundary Calibration** This pseudocode delineates a structured framework for implementing IR marker detection by leveraging the Kinect v2 camera alongside image processing capabilities provided by OpenCV. The process commences with histogram equalization of the input image, which enhances the contrast, making IR markers more discernible. A binary threshold is then applied to isolate these markers based on their intensity.

The blob detection feature in OpenCV is particularly pivotal in this application. A Blob is a group of connected pixels in an image that share some common property. The BlobDetector object is configured to detect white blobs (high-intensity areas) in a binary image. The parameters are meticulously set to filter blobs by area and circularity, which are essential for recognizing the characteristic near-square shape of the IR markers used to delineate the sandbox boundary. Since the markers are expected to be uniform in size and shape, these settings allow the algorithm to effectively ignore other objects in the scene that do not meet these specific criteria.

Upon detecting the blobs, which represent potential IR markers, a check is performed to ensure that at least three markers are identified to form a valid boundary. If fewer than three markers are detected, the algorithm reports that no sandbox is detected. Otherwise, the points corresponding to these markers are used to compute a convex hull, which is then drawn onto the image. This step visualizes the sandbox's perimeter, helping to confirm the accurate detection and positioning of the IR markers. The use of the convex hull algorithm ensures that the boundary encompasses all identified markers, thereby forming the most concise perimeter possible around them.

The efficiency and accuracy of this method hinge on the ability of the blob detection algorithm to correctly identify and segregate the IR markers based on predefined physical characteristics. This approach proves particularly suitable for applications where precise spatial delineation is required in varied lighting conditions, making it a robust choice for sandbox boundary calibration in embedded AI applications.

### 2.2.3 Display – Generating Colored Contours from Depth Information

**Introduction to the Approach** The purpose of this function is to visualize depth information in a more intuitive manner by applying a color mapping to depth data. This approach helps in identifying features at various depths within a scene by representing them with different colors. The process involves several key steps: clamping the depth values to a specified range, normalizing these values, and then mapping them to a set of colors.

#### Detailed Steps

1. **Normalizing Depth Values:** The depth values are first clamped to ensure they fall within a user-specified range. This step is crucial as it filters out depth values that are either too close or too far, which may not be relevant to the analysis.

2. **Mapping to Color Indices:** These normalized values are then mapped to color indices. The mapping translates the normalized depth into a discrete index, which corresponds to a specific color in the colormap.
3. **Creating a Custom Colormap:** A custom colormap is created from the JET colormap, which contains a gradient of colors. Only a specific number of colors defined by  $n$  are selected to create this colormap, which helps in distinguishing between different depth values more effectively.
4. **Applying the Colormap:** Finally, the custom colormap is applied. This step maps the previously obtained color indices to actual RGB color values, which are then used to produce the final visual representation of the depth data.

**Python Code Implementation** The function `plot_depth_contour` demonstrates this process. It takes a depth array and parameters defining the depth range and the colormap specifics, and it returns an image where the depth information has been transformed into a visually comprehensible format using colors.

This method not only enhances the visual appeal of the depth data but also significantly aids in the interpretation of the spatial and depth-related complexities of the scene. By transforming raw depth information into a colored depth map, we facilitate a more accessible understanding of the data for analysis and decision-making processes.

## 2.2.4 Concurrent Handling of Web and Display Using Multithreading

**Introduction to the Approach** To enhance the responsiveness and efficiency of our system, which involves both displaying processed images and serving these images via a web interface, we employ a multithreading approach. This method allows the web server and the image display processes to operate simultaneously without blocking each other, ensuring smooth execution. The key to managing the shared resources, particularly the image frames, is the use of threading with a lock mechanism to handle concurrency safely.

**Detailed Implementation** The implementation involves two main threads: one for the Flask web server and another for the image processing task. A global variable, `current_frame`, is used to store the latest processed frame, which is accessed by both threads. This variable is protected by a threading lock, `current_frame_lock`, to prevent concurrent access issues such as race conditions.

1. **Web Server Thread:** This thread runs the Flask application, which includes a route `/video_feed` specifically for streaming video frames to clients. The `video_stream` function continuously checks for the latest available frame held in the `current_frame` global variable. When a new frame is available, it copies this frame (ensuring thread safety with a lock) and encodes it into JPEG format before yielding it as part of a multipart HTTP response.

2. **Image Processing Thread:** Simultaneously, the main application thread processes incoming video data to generate color-mapped contours or other image processing outputs. Once a frame is processed, it updates the `current_frame` with a new image. This update is also performed under the protection of `current_frame_lock` to ensure that the web server thread can safely access the latest frame without encountering incomplete or corrupted data.

Using multithreading with careful management of shared resources allows our system to handle intensive processing tasks while maintaining a responsive web interface. It effectively decouples the processing workload from the web serving tasks, leading to improved performance and user experience. This setup is particularly beneficial in environments where real-time data visualization is crucial, such as in surveillance, remote monitoring, or interactive media applications.

## 2.3 Projection Subsystem

### 2.3.1 Projector Selection

The projector in projection subsystem is responsible for projecting the topography map onto the sand surface. Considering that topography is displaying on sand, the projection subsystem should be able to project the topography map with high luminance to overcome the strong diffuse reflection of sand. To mount the projector lower on one side of the sandbox for an easier installation and match the camera's field of view, the projection subsystem aims for a short throw projector. For this, this AR sandbox uses a high luminance ultra short throw projector from EPSON with a resolution of  $1024 \times 768$  pixels and zoom factor of 0.31 in horizontal direction and 0.41 in vertical. It has a refresh rate of 60 fps and 2600 ANSI lumens. The projector connects with the processing unit through an HDMI cable, which is compatible with the Jetson development board.

One feature of this projector comparing to the projector previously used in AR sandbox project is that its projection is significantly off-axis, which means that the optical axis of the projector does not coincide with the center of the projected image. This feature is beneficial for the installation of the projector as it can be mounted lower on one side of the sandbox and still project the topography map onto the sand surface. However, this feature also brings challenges to the calibration of the projector-camera system, which will be discussed in the Projector-Camera System Calibration section.

### 2.3.2 Projector Auto Focusing

As the projection subsystem uses a ultra short throw projector, a small distance change between the projector and the surface it projects to would bring a blurry image and requires refocusing with the focusing lever on the projector. To automatically adjust the focus of the projector according to the height of the sand surface as the projector only allows for manual focusing through a lever shown in Figure 5, an auto focusing subsystem is designed. It uses a stepper motor with sliding table to control the projector's lever automatically. The stepper motor connects with the focusing lever on the projector through

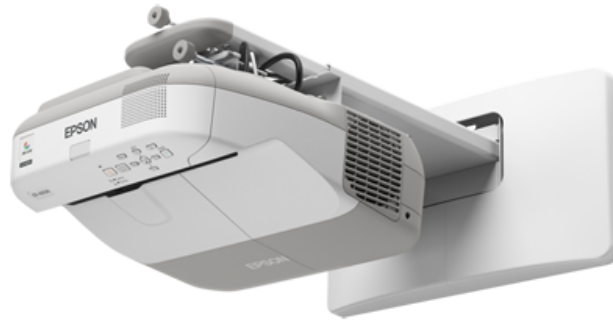


Figure 4: EPSON EB-CU600X Projector

a custom connector on a sliding table shown in Figure 5. The stepper motor would be controlled by the processing unit and connected with the lever to change its position. The processing unit would use the depth information from the sensor subsystem to adjust the focus of the projector according to the height of the sand table.

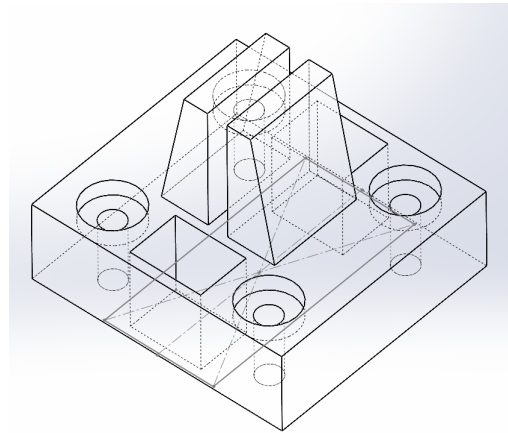
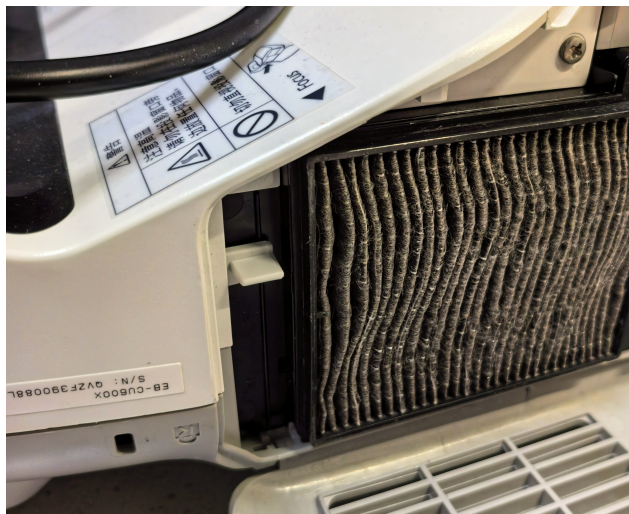


Figure 5: Focus Lever on EPSON EB-CU600X Projector and Connector Design

The auto focusing system uses a 28HD2830 two-phase stepper motor with a  $1.8^\circ$  step angle and a holding torque of  $0.07 \text{ N} \cdot \text{m}$ . It is installed on a linear sliding table with 50 mm stroke. The stepper motor connects with the processing unit through a drive board. The sliding table will go for 1 mm per rotation of the stepper motor. The drive board connects with the processing unit through GPIO pins and receives PWM signal from the processing unit. The Jetson Nano has GPIO pins that can be used as PWM output. The acceleration and deceleration of the stepper motor can be controlled by the PWM signal. Figure 6

shows the relationship between distance and speed of the stepper motor.

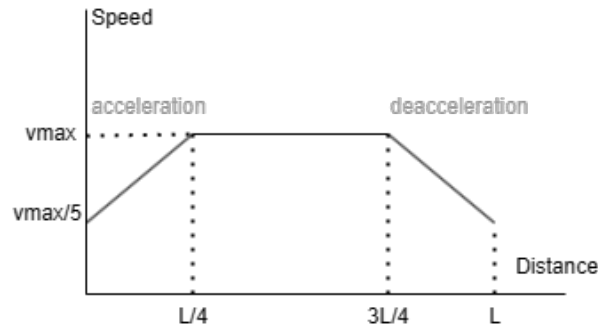


Figure 6: Stepper Motor Speed vs. Distance

For a more precise control of the stepper motor's position, the motor is driven by half-step sequence. The half-step sequence is a sequence of 8 steps that the motor goes through to complete one full activation loop. Comparing to full-step sequence that only has 4 steps per activation loop, the half-step sequence provides a higher precision and less vibration at low-speed operation [6]. With a  $1.8^\circ$  step angle and a sliding table with 1 mm lead, a half-step sequence could achieve 0.0025 mm accuracy in theory, which is enough for focusing lever control. Figure 7 compares the full-step and half-step sequences. By controlling the delay time between each step, the stepper motor can be controlled to move at different speeds. The motor function pseudocode is shown in Appendix C.

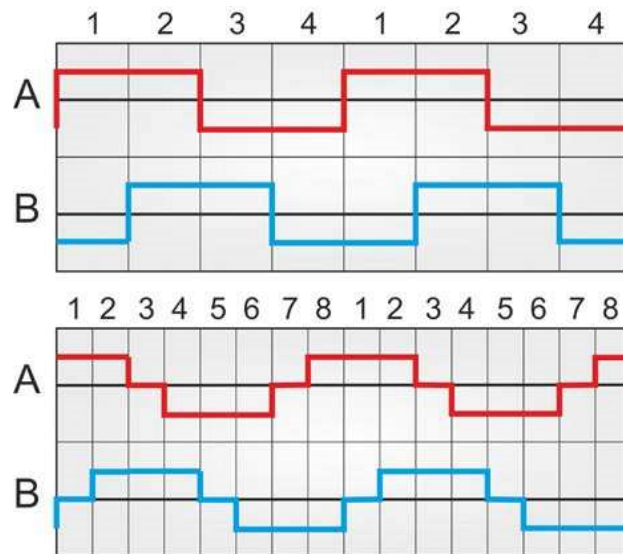


Figure 7: Comparison of Full-Step and Half-Step Sequences

The drive board connects with the stepping motor through a 4-pin connector. L298N chip drives the stepping motor as shown in Figure 9. The L298N is a dual H-bridge motor driver integrated circuit. It is a high voltage, high current dual full-bridge driver designed to accept standard TTL logic levels and drive inductive loads such as relays,

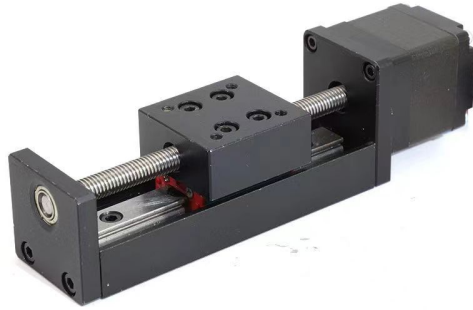


Figure 8: Stepper Motor and Sliding Table

solenoids, DC and stepping motors. The L298N can drive up to 2A per channel and has a peak current of 3A per channel. The L298N has a thermal shutdown feature to prevent overheating and a short-circuit protection feature to prevent damage to the stepping motor. The L298N is simple to use and provides low noise and high stability, which satisfies the need of the stepping motor.

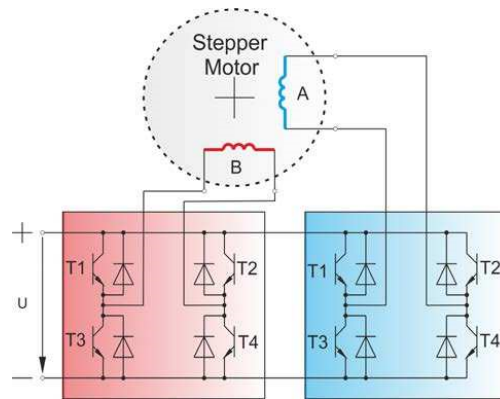


Figure 9: Control of Stepper Motor by H-Bridge

The relationship between the position of the focusing lever and the distance between the projector and the focused surface is determined after getting datapoints and fitting linear relationship as shown in Figure 11. As the standard for the projector in focus is not determined by numerical measurements but by visual judgement, a linear relationship is used for its simplicity and effectiveness. The distance information is obtained from the sensor subsystem.

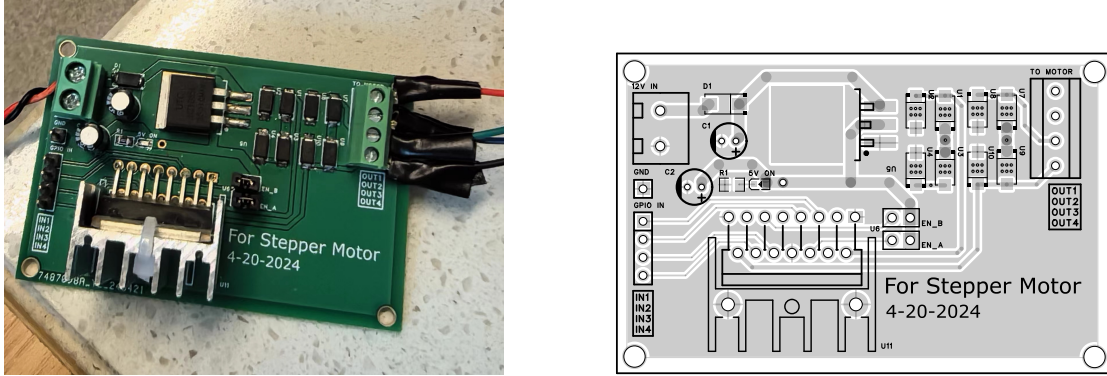


Figure 10: PCB Design of Stepper Motor Drive Board

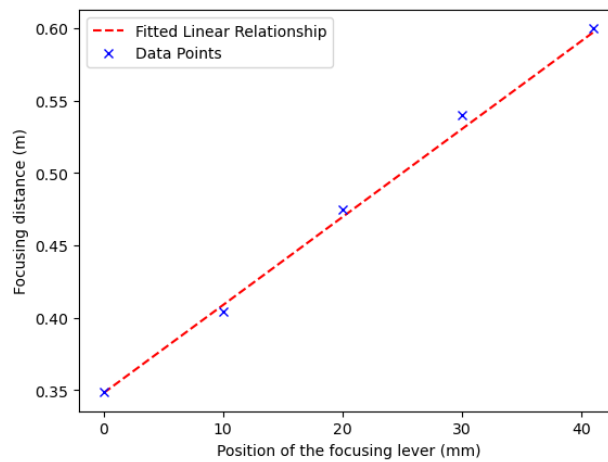


Figure 11: Datapoints and Fitting Results of Projector's Focusing System

### 2.3.3 Projector-Camera System Calibration

Projector-camera system calibration is a crucial step in the AR sandbox project. Comparing with existing methods such as the KinectProjectorToolkit [7] that uses a printed checkerboard pattern, this AR sandbox use a different method to calibrate the off-axis projector-camera system that does not require a printed checkerboard pattern. Considering that this AR sandbox aims for educational use and the printed checkerboard pattern may be easily lost or damaged, this would be a much better solution comparing to the existing methods.

As most AR applications, the relative position between the projector and the Kinect camera are fixed. To speed up the calibration process when applying transformation to colored contours, this AR sandbox first uses a simple homography transformation to warp the colored contours to the projector's perspective. The equation used is defined as:



$$\begin{bmatrix} x_p \\ y_p \\ 1 \end{bmatrix} = H_{3 \times 3} \begin{bmatrix} x_c \\ y_c \\ 1 \end{bmatrix} \quad (1)$$

Here,  $H_{3 \times 3}$  is the homography matrix that maps the camera coordinate system to the projector coordinate system.  $x_p$  and  $y_p$  are the pixel coordinates in the projector coordinate system, and  $x_c$  and  $y_c$  are the pixel coordinates in the camera coordinate system. The benefit of this method is that the processing speed is very fast and make the AR sandbox more responsive as it does not consider the depth information of the sand surface. The drawback of this method is that it is not accurate and the final topography map offsets more than 5 cm with the sand surface when the sand surface has a more than 5 cm height difference. One homography applies when points in the image are on the same plane and multiple planes in the image requires multiple homographies, one for each plane [8]. Consider that the sandbox this project uses contains sand with a maximum height difference of around 10 cm, this method is not suitable for this project. Besides the depth difference of sand, another major contributing factor causing large image offsets is that the projector is significantly off-axis as it is installed on the side of the sandbox rather than above head and it is short throw that a small change in distance between the projector and the sand surface would bring a large position change in the projected image. Therefore, the AR sandbox uses a more accurate method to calibrate the projector-camera system.

Our AR sandbox takes an additional step that takes the depth information into account, which requires the intrinsic, extrinsic and distortion parameters of both the camera and the projector. There are two main steps in the calibration process: camera calibration and projector calibration. The camera calibration considers the intrinsic parameters of the camera, such as focal length, principal point, and distortion coefficients. The projector calibration considers the intrinsic and extrinsic parameters of the projector, which is modelled as a camera. The detailed calibration process is shown in Figure 12. The projector-camera system is shown in Figure 13. Some simplifications taken and assumptions made in the calibration process will be discussed in the following sections.

**Camera Calibration** Camera calibration is the process of estimating the intrinsic parameters of the camera, such as focal length, principal point, and distortion coefficients. These parameters are crucial for converting the pixel coordinates of the camera to the real-world coordinates. The relation between 2D data points in the camera coordinate system and 3D data points in the world coordinate system is given by the following equation according to the pinhole camera model proposed in Zhang’s paper [9]:

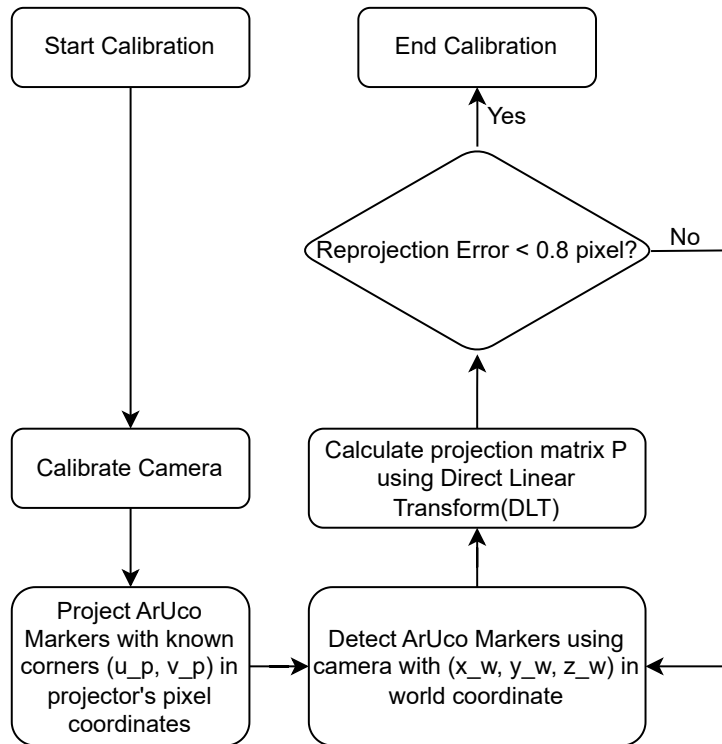


Figure 12: Flowchart of Projector-Camera System Calibration

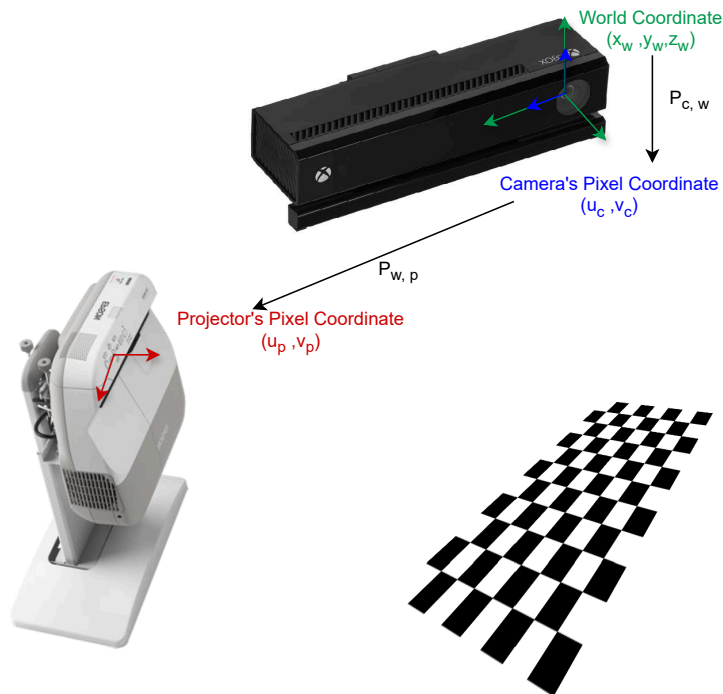


Figure 13: Camera Projector System

$$Z_c \begin{bmatrix} x_c \\ y_c \\ 1 \end{bmatrix} = K_c \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \quad (2)$$

$$K_c = \begin{bmatrix} \frac{1}{dx} & 0 & C_x \\ 0 & \frac{1}{dy} & C_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f_x & 0 & 0 & 0 \\ 0 & f_y & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (3)$$

By defining the world coordinate system's origin to be the same as the camera's pixel coordinate system and align pixel coordinate's u-v plane with the x-y plane of the world coordinate, the rotation matrix  $R$  is the identity matrix and the translation vector  $t$  is a zero vector. For the distortion coefficients, the camera model is using a standard Brown-Conrady model, which is a radial-tangential distortion model. The equation describing distortion is in Equation 4 with tangential and distortion terms [10]. The intrinsic parameters and distortion parameters of the camera are provided in a paper [11] and source code of a library for Kinect [12] as shown below:

```
CameraParams = {
    "cx":254.878,
    "cy":205.395,
    "fx":365.456,
    "fy":365.456,
    "k1":0.0905474,
    "k2":-0.26819,
    "k3":0.0950862,
    "p1":0.0,
    "p2":0.0,
}
```

$$\begin{bmatrix} x_{\text{distorted}} \\ y_{\text{distorted}} \end{bmatrix} = \begin{bmatrix} x_{\text{undistorted}} \\ y_{\text{undistorted}} \end{bmatrix} \times (1 + k_1 r^2 + k_2 r^4 + k_3 r^6) + \begin{bmatrix} 2p_1 x_{\text{undistorted}} y_{\text{undistorted}} + p_2 (r^2 + 2x_{\text{undistorted}}^2) \\ p_1 (r^2 + 2y_{\text{undistorted}}^2) + 2p_2 x_{\text{undistorted}} y_{\text{undistorted}} \end{bmatrix} \quad (4)$$

**Projector Calibration** Projector calibration is the process of estimating the intrinsic and extrinsic parameters of the projector, which is modelled as a camera. Considering that this projector exhibits strong off-axis projection as shown in Figure 14, intrinsic matrix  $K_p$  is different from the camera's intrinsic matrix  $K_c$ . The relation between 2D data points

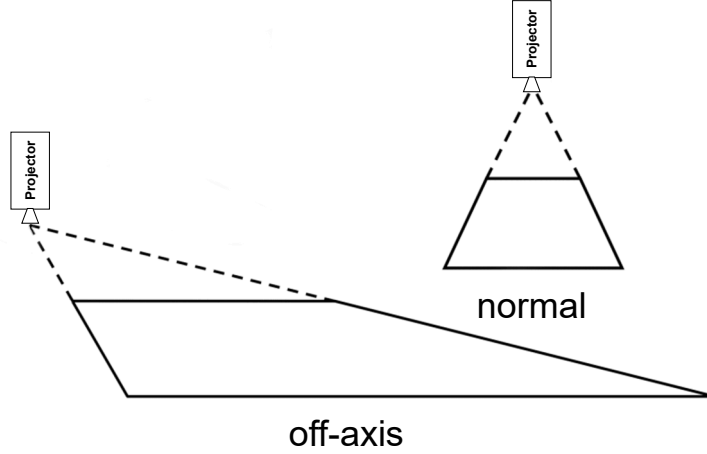


Figure 14: Comparison between On-axis and Off-axis Projection

in the projector coordinate system and 3D data points in the world coordinate system is given by the following equation:

$$Z_p \begin{bmatrix} x_p \\ y_p \\ 1 \end{bmatrix} = K_p \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \quad (5)$$

where  $K_p$  is the intrinsic matrix of the projector,  $R$  is the rotation matrix, and  $t$  is the translation vector. According to Hancock and Carpendale's modelling [13], The intrinsic matrix  $K_p$  considering the off-axis projection is defined as:

$$K_p = \begin{bmatrix} \frac{1}{dx} & 0 & C_x \\ 0 & \frac{1}{dy} & C_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f_x & 0 & A & 0 \\ 0 & f_y & B & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (6)$$

where additional parameters  $A$  and  $B$  are introduced as a multiplier of the distance. This means that functions in OpenCV such as `cv2.calibrateCamera()` cannot be directly used to get the intrinsic and extrinsic matrix of off-axis projector. However, what remains unchanged is the 3 by 4 projection matrix  $P_{w,p} = K_p * [R_p | t_p]$  that maps the 3D data points in the world coordinate system to the 2D data points in the projector coordinate system. Therefore, the calibration process chooses to directly estimate the projection matrix using the Direct Linear Transformation (DLT) algorithm.

For a 3 by 4 projection matrix  $P_{w,p}$ , 12 unknown parameters needs to be solved, which needs at least 6 corresponding points between the world coordinate system and the pro-

jector coordinate system. These points are obtained by projecting an ArUco [14] marker board onto a surface and detecting the corners of the markers in the camera coordinate system. The ArUco marker board is a planar board with a set of markers with known dimensions and ids, which does not require the camera to capture the whole board. The world coordinates of detected corners of the markers is then calculated with previously calibrated depth camera. The following equation is used to calculate the projection matrix  $P_{w,p}$ :

$$\begin{bmatrix} x_p \\ y_p \\ 1 \end{bmatrix} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \quad (7)$$

To solve for the parameters  $p_{ij}$  in  $P_{w,p}$ , the Direct Linear Transformation (DLT) [15] transform the equation into a linear system  $Lp = 0$  as shown in the following equation:

$$L = \begin{bmatrix} X_1 & Y_1 & Z_1 & 1 & 0 & 0 & 0 & 0 & -u_1 X_1 & -u_1 Y_1 & -u_1 Z_1 & -u_1 \\ 0 & 0 & 0 & 0 & X_1 & Y_1 & Z_1 & 1 & -v_1 X_1 & -v_1 Y_1 & -v_1 Z_1 & -v_1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ X_n & Y_n & Z_n & 1 & 0 & 0 & 0 & 0 & -u_n X_n & -u_n Y_n & -u_n Z_n & -u_n \\ 0 & 0 & 0 & 0 & X_n & Y_n & Z_n & 1 & -v_n X_n & -v_n Y_n & -v_n Z_n & -v_n \end{bmatrix} \quad (8)$$

$$p = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} & p_{21} & p_{22} & p_{23} & p_{24} & p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix}^T \quad (9)$$

The projection matrix  $P_{w,p}$  is then estimated by solving the linear system  $Lp = 0$  using the singular value decomposition (SVD) method. The reproduction error of the projection matrix is calculated by the following equation:

$$\text{Reproduction Error} = \sqrt{\frac{\sum_{i=1}^n (u_i - \hat{u}_i)^2 + (v_i - \hat{v}_i)^2}{n}} \quad (10)$$

where  $u_i$  and  $v_i$  are the pixel coordinates of the  $i$ th point in the projector coordinate system, and  $\hat{u}_i$  and  $\hat{v}_i$  are the pixel coordinates of the  $i$ th point in the projector coordinate system calculated by the projection matrix  $P_{w,p}$ . The projection matrix  $P_{w,p}$  is then used to project the colored contours to the projector coordinate system. The reproduction error of the projection matrix is required to be less than 0.8 pixel for the calibration to be considered successful.

Considering that the projector has a built-in distortion correction feature that cannot be turned off and performs correction well, the distortion coefficients of the projector are ignored.

**Applying Projector-Camera System Calibration** To apply the projector-camera system calibration, two steps are taken. First, the depth information of the sand surface is obtained from the sensor subsystem. The depth information is then converted to the real-world coordinates based on the calibrated camera. The real-world coordinates are then projected to the projector coordinate system using the projector's projection matrix. The final topography map is then projected onto the sand surface.

When converting from the camera coordinate system to the projector coordinate system, a vectorization method is proposed. The open source library called "libfreenect2" [5] provides a function `getPointXYZ` that can only perform the conversion for one point at a time. To speed up the process, the function is vectorized to convert multiple points at a time. The code is shown in Appendix D.

When projecting the real-world coordinates to the projector coordinate system, the projection matrix is used. The code is also vectorized to speed up the process. The vectorized code is shown in Appendix E. By vectorizing the function, the processing speed is shortened by 20 times. The final topography map is then projected onto the sand surface. The final topography map is shown in Figure 15.

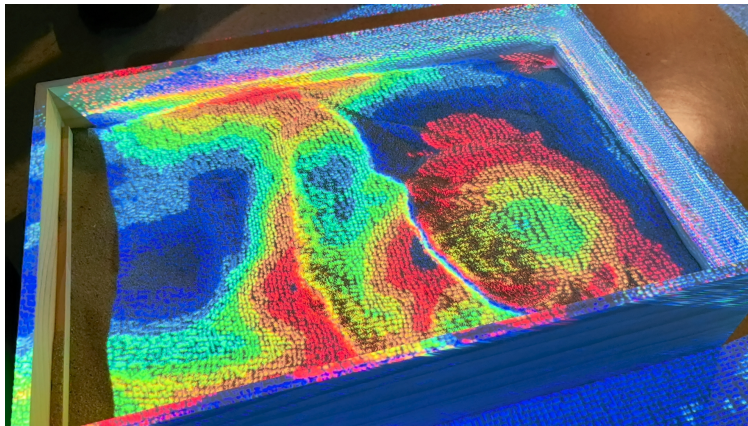


Figure 15: Topography Map Projected onto Sand Surface

## 2.4 Powering Subsystem

The powering subsystem is responsible for providing power to the sensor, processing, and display subsystems. The powering subsystem should be able to provide power to the sensor, processing, and display subsystems with sufficient power and voltage. The powering subsystem should be able to provide power to the sensor, processing, and display subsystems with high reliability and high safety.

The Kinect sensor and projector take 220 V AC power, while the processing unit and the drive board of stepping motor takes 5 V DC power. Two power converters safely convert the 220 V AC power to 5 V DC power and provide 20 W at maximum for the processing unit and the drive board of the stepping motor. The Jetson Nano takes 5 V DC power and 2 A current at maximum. The stepping motor takes 5 V DC power and 0.35 A current. The power consumption of these two parts is within the power supply's capacity. A power strip with high current and temperature protection connects the Kinect sensor, projector, stepping motor and processing unit to one plug and provides at most 2500 W power to the whole system. It is safe and reliable to use.

In the auto focusing subsystem, a UZ1085 LDO voltage regulator is used to convert the 12 V DC power to 5 V DC power for chips on PCB board and the stepper motor. The LDO voltage regulator can provide a stable 5 V DC power and up to 2 A current, which satisfies the need of the stepper motor. The LDO voltage regulator has a thermal shutdown feature to prevent overheating, a short-circuit protection and overcurrent protection feature to prevent damage to the processing unit. It is simple to use and provides low noise and high stability voltage output.

## 2.5 Database and Wireless Display Subsystem

**Database** The database serves as a centralized repository for storing the voluminous data generated by the depth camera. Images captured by the depth camera contain intricate depth information, making them essential assets for calibration, segmentation and contour painting. By structuring these images within a database, we ensure efficient organization, retrieval, and management of the data, enabling seamless access for analysis and processing on the back end of the project. As the volume of captured images grows over time, a well-designed database architecture allows for seamless expansion without compromising performance. Scalability ensures that the system can accommodate increasing data loads without experiencing degradation in responsiveness or reliability.

SQLite was chosen as the database solution for this project due to several distinct advantages it offers over alternative options such as PostgreSQL, MySQL, MongoDB, Firebase, and Neo4j:

- **Simplicity:** SQLite is a self-contained and file-based database engine that requires minimal setup and administration. Unlike client-server databases like PostgreSQL or MySQL, SQLite does not require a separate server process to be running, making it easy to integrate into Python applications without additional dependencies. Its simplicity makes it ideal for our project where ease of deployment on the front end is paramount.
- **Seamless Integration with Python:** SQLite's seamless integration with Python via the built-in "sqlite3" module makes it a natural choice for our project which segmentation, calibration and contour-painting are developed in Python. The module provides a straightforward API for interacting with SQLite databases, allowing developers to perform CRUD (Create, Read, Update, Delete) operations with minimal

boilerplate code. As database in our project is designed to perform read, write and delete, SQLite provides the perfect solution.

**Wireless Display** The integration of wireless transmission capabilities within the web page enables real-time streaming of the camera feed to end-users who are using either mobile devices or PC. Wireless transmission eliminates the need for physical connections, offering flexibility and mobility in accessing the camera feed from diverse locations and devices. Moreover, the web page serves as an intuitive interface for users to interact with the camera feed and manage the stored images effectively. Through the web page, users can initiate live streams, navigate through archived frames, and enjoy the view of colored contours. As the overseer of the sandbox, parents are able to monitor their children's behavior even they are not near the sandbox, providing extra safety precautions.

Flask paired with pure HTML and CSS offers a lightweight, flexible, and efficient approach to web page development that aligns well with the requirements and constraints of the project. Here's why this choice stands out compared to the other alternatives in our project: 1. Flask, a lightweight micro-framework, provides the essentials for building web applications without imposing unnecessary complexity. Its minimalist design and intuitive API make it easy to learn and use, particularly for smaller project like ours. By leveraging pure HTML and CSS alongside Flask, it is easy to maintain a simple and straightforward development workflow, and debugging process can be a lot easier. 2. Using pure HTML and CSS allows for granular control over the web page's structure, layout, and styling, enabling precise customization to achieve the desired design and responsiveness. Unlike front-end frameworks or preprocessors, which may come with predefined styles and conventions, pure HTML and CSS offer complete flexibility and make it possible to develop a simple but fully functional responsive web page layout which can adapt to all kinds of viewing devices well.

**Design Details** Database and wireless display are deeply connected with each other in our project, and design details of both shall be discussed together.

As shown in Figure 16, overall 10 Python Functions and 3 Javascripts have been implemented to realize both database setup and web page layout.

Decorator: `@app.route('/')`

Decorator: `@limit_users(10)`

Function: `index`

Description: This function serves as a route handler for the root URL ('/'). It limits access to a maximum of 10 users concurrently using the `limit_users` decorator. The function increments the count of active users when a new user accesses the page and deletes the existing 'frames.db' database file. It then clears the set of deleted frame timestamps and fetches the list of captured frames from the database using the `get_frame_list` function. After generating a QR code for the current URL, it saves the QR code image and



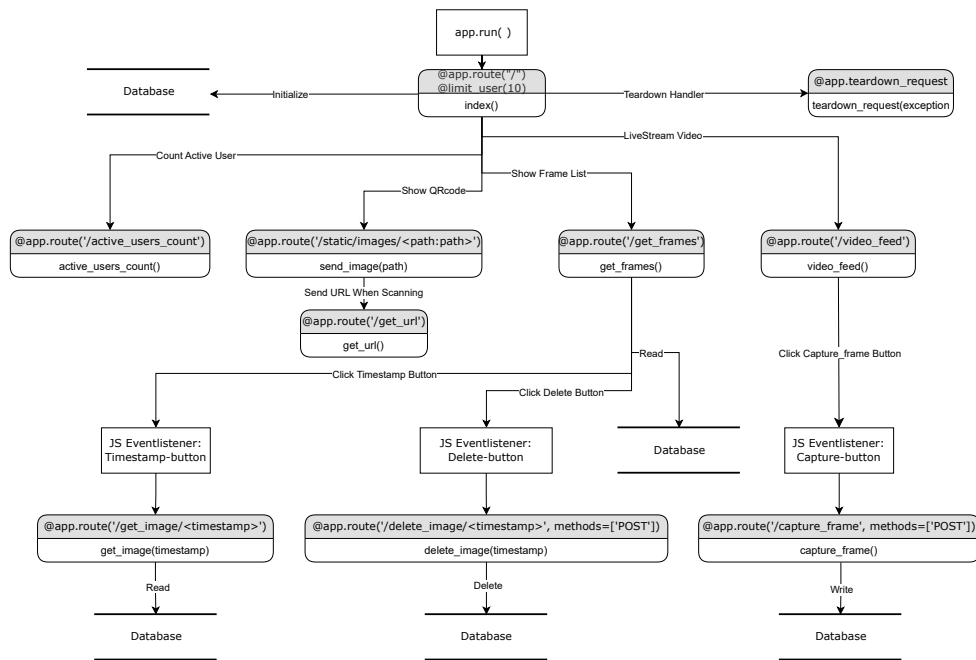


Figure 16: Flowchart on database and wireless transmission subsystem

renders the 'index.html' template, passing the frames and QR code image path as context variables.

Decorator: `@app.route('/static/images/<path:path>')`

Function: `send_image`

Description: This function serves static images located in the 'static/images' directory. It takes a path parameter representing the image file path relative to the 'static/images' directory. It uses Flask's `send_from_directory` function to locate and send the requested image file to the client.

Decorator: `@app.route('/video_feed')`

Function: `video_feed`

Description: This function serves as a route handler for `video_feed`. It streams video frames using the `generate_frame` function, which continuously generates frames for a video stream. It returns a Flask Response object with the generator function `generate_frame()` as its content.

Decorator: `@app.route('/get_frames')`

Function: `get_frames`

Description: This function serves as a route handler for `get_frames`. It fetches the list

of captured frame timestamps from the database using the `get_frame_list` function. It then renders the `frame_list.html` template, passing the fetched frames as context variables.

Decorator: `@app.route('/capture_frame', methods=['POST'])`

Function: `capture_frame`

Description: This function serves as a route handler for capturing a frame via POST request. It starts by checking if the current frame is available, returning a status code of 500 if it's not. It then converts the current frame to a JPEG image, retrieves the current timestamp, and saves the JPEG image to a BytesIO object. After connecting to the SQLite3 database, it creates the 'frames' table if it doesn't exist and checks the count of frames currently stored in the database. If the count exceeds the maximum frame limit (MAX\_FRAMES), it deletes the oldest frame. Finally, it inserts the current frame into the database with its timestamp. The function returns an empty response with a status code of 204 (No Content) to indicate that the frame has been successfully captured and saved to the database.

Decorator: `@app.route('/get_image/<timestamp>')`

Function: `get_image`

Description: This function serves as a route handler for fetching an image corresponding to a given timestamp from the database. It takes the timestamp as a parameter in the URL. The function connects to the SQLite3 database, executes a query to select the image data for the given timestamp, and fetches the row. If the row exists, it retrieves the image data and returns it as a response using Flask's `send_file` function with the mimetype set to 'image/jpeg'. If the image is not found for the given timestamp, it aborts with a status code of 404 (Not Found). In case of any exception during the process, it prints an error message and aborts with a status code of 500 (Internal Server Error).

Decorator: `@app.route('/delete_image/<timestamp>', methods=['POST'])`

Function: `delete_image`

Description: This function serves as a route handler for deleting an image corresponding to a given timestamp from the database via a POST request. It takes the timestamp as a parameter in the URL. The function connects to the SQLite3 database, executes a delete query to remove the image data for the given timestamp, and commits the changes. If the deletion is successful, it prints a success message and returns an empty response with a status code of 204 (No Content). In case of any exception during the deletion process, it prints an error message and aborts with a status code of 500 (Internal Server Error).

Decorator: `@app.route('/active_users_count')`

Function: `active_users_count`

Description: This function serves as a route handler for fetching the count of active users. It returns a JSON object containing the count of active users, which is obtained by taking the length of the `active_users` set.

Decorator: `@app.teardown_request`

Function: `teardown_request` Description: This function is registered as a teardown handler by Flask using the `@app.teardown_request` decorator. It is called after each request, regardless of whether an exception occurred during the request processing.

## 2.6 Structure Subsystem

### 2.6.1 The Main Part of Sandbox

The main part of the sandbox is dedicated to the containment of sand and constitutes the central and largest portion that interconnects with all other segments. This integral section comprises the main body of the sandbox and a removable lid as shown in Figure 17. With the lid in place, the sandbox assumes external dimensions of 600x400x150 millimeter, while the internal cavity measures 578x378x128 millimeter. The lid is a composite structure made of two boards that are pivotally affixed using hinges, enabling a full 180-degree swivel. The interface between the lid and the main body is established through a pliable silicone sheet, which guarantees a unfettered 270-degree movement. The design rationale for this feature is to allow the lid to be manipulated with ease, enabling it to be stored flush against the side of the sandbox during utilization. This configuration ensures unobstructed access and convenience for the user around the entire perimeter of the sandbox.

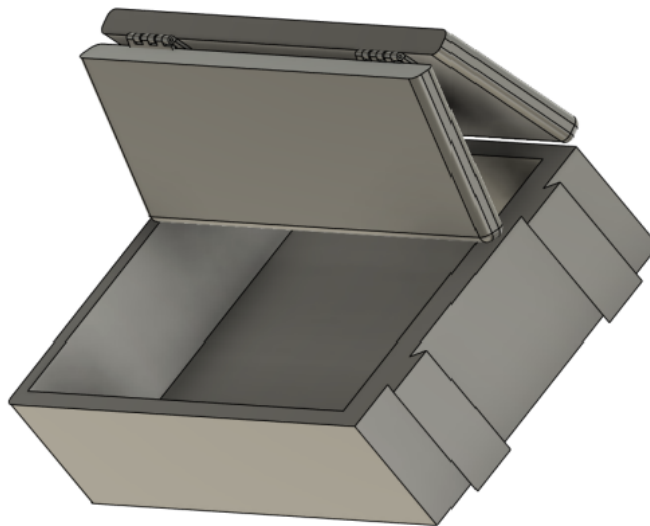


Figure 17: The CAD model of the main part of sandbox

The material used for this section is pine wood. The box body is constructed by joining pine boards with a thickness of 12 millimeters. The selection of materials for this part was greatly influenced by budgetary constraints. The ideal material for fabrication would be PLA (Poly Lactic Acid) plastic, which offers higher strength properties. However, calculations and simulation validations have confirmed that the structural integrity of

pine wood is sufficient to support the weight of the sand, while also making the sandbox more portable.

## 2.6.2 The Separable Part

The overall structure of the separable part is depicted in the diagram. Its function is to house the projector, connect sensors, and accommodate all electronic devices. The top of this part features a recess measuring 160x380x10 mm, slightly larger than the projector's base, allowing the projector to be securely inserted. The sensors will be mounted on a pan-tilt mechanism and secured by a rod to the upper edge of the separable part. The pan-tilt mechanism enables free rotation, facilitating the convenient use of the sensors. The rod is made of aluminum alloy and is adjustable in length. The hollow center of the separable part is designed for storing the processing system and power supply system, ensuring that the entire system can be powered by a single wire extending from the box. The open side will be covered with a rotating plate to maintain a sleek, integrated appearance. The plate features a rectangular opening to aid in positioning the clamping point of the rod connecting to the sensors. The connection between the main body of the sandbox and the separable part is achieved using a dovetail joint structure. This design is based on the reason that when the sandbox is placed horizontally, the joint will not be subjected to significant force, but it must ensure stability. The simple structure also allows for easier assembly and disassembly. The significance of this section lies in the ability to use our electronic devices independently. This means that our electronic devices are not dependent on the original sandbox but can be adapted to various scenarios and different sand tables, increasing the flexibility of product use.

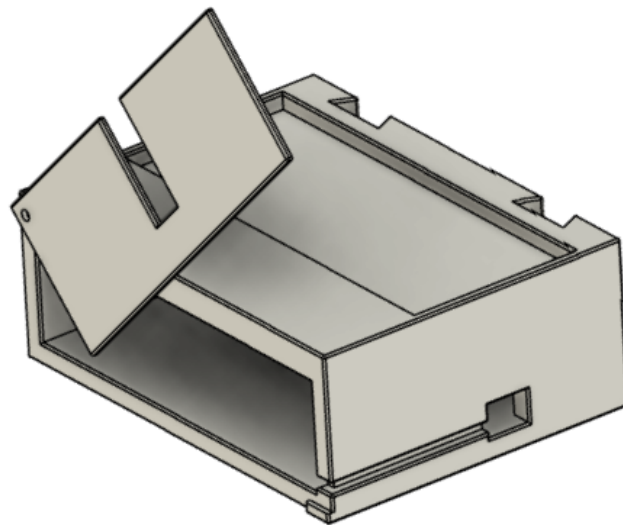


Figure 18: The CAD model of the separable part

The material for the separable part is PLA, produced by 3D printing. In actual production, more cost-effective methods such as injection molding can be considered. The strength of the PLA-made separable part fully meets safety requirements. Cheaper materials with slightly lower strength could also be suitable alternatives.

### 2.6.3 The Recovery Subsystem

The primary function of this system is to facilitate the rapid and convenient re-leveling of used sand tables. To avoid complicating the use and maintenance of the sandbox with electric equipment, a special shovel with a unique structure was designed to achieve this purpose. The structure of the shovel is depicted in the diagram. Its mechanism operates by pushing excess sand into the shovel when thrust forward, where it is stored in the hollow segment in the center, and upon returning, the stored sand fills the gaps through the bottom slit. By placing the shovel on the guide rail as shown, the lower end of the shovel is set at the original height of the sand surface in the sandbox. This design allows the user to restore the sand surface simply by pushing the shovel. The edges of the shovel have been blunted after optimization to prevent accidental injury to the user. Furthermore, the manual operation of the recovery system eliminates the risk of user harm, making it decidedly more convenient and safe.

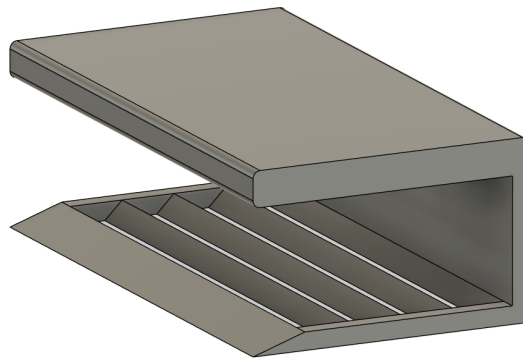


Figure 19: The CAD model of the shovel

The shovel is fabricated using PLA material via 3D printing, and in actual production, more cost-effective methods such as injection molding can be considered. The guide rail maintains the same pine wood material as the main body of the sandbox, which is rougher than anticipated in the design phase. The friction can be reduced by applying polytetrafluoroethylene (PTFE) tape to the surface of the rail.

### 2.6.4 The Mobility System

The purpose of this section is to make the entire AR sandbox portable. The weight of all the structures of the AR sandbox, including the electronic equipment, approaches 30

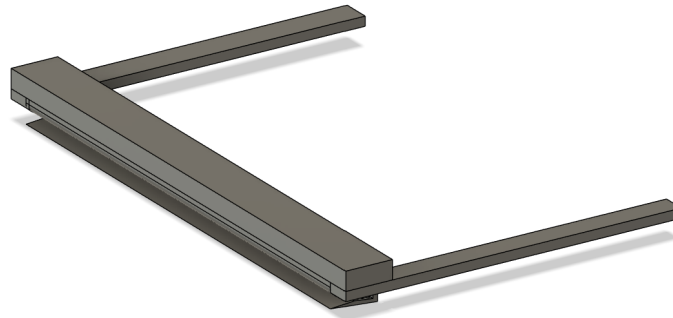


Figure 20: The CAD model of the recovery system

kilograms. Manual handling is impractical and also increases the risk of damage to the device. The mobility system consists of two parts: a metal frame and wheels. The metal frame is welded from angle steel with a thickness of 3 millimeters and a width of 50 millimeters, and it is reinforced with one longitudinal and three transverse cross braces at the bottom to enhance strength and stability. The dimensions of the metal frame are 1500x400 millimeters, taking into account the need to move the projector for focusing. The material used for this part is 304 steel, which has a very high hardness. There are a total of six wheels: the four outer wheels are swivel casters with brakes, and the two in the middle are ordinary swivel casters. The wheels are made of nylon and are 2 inches in size. This ensures that the system can be fixed in place during use and will not move easily. Testing has shown that the mobility system can support a weight of 200 kilograms without deformation, which fully meets the usage requirements.

## 3 Verification

### 3.1 Sensor Subsystem

By logging the frame rate and latency when fetching RGB, IR and depth frames from Kinect camera, the performance of the sensor system can be verified. The frame rate of the camera is higher than 30 fps with latency less than 20 ms. The log is shown in Figure 21.

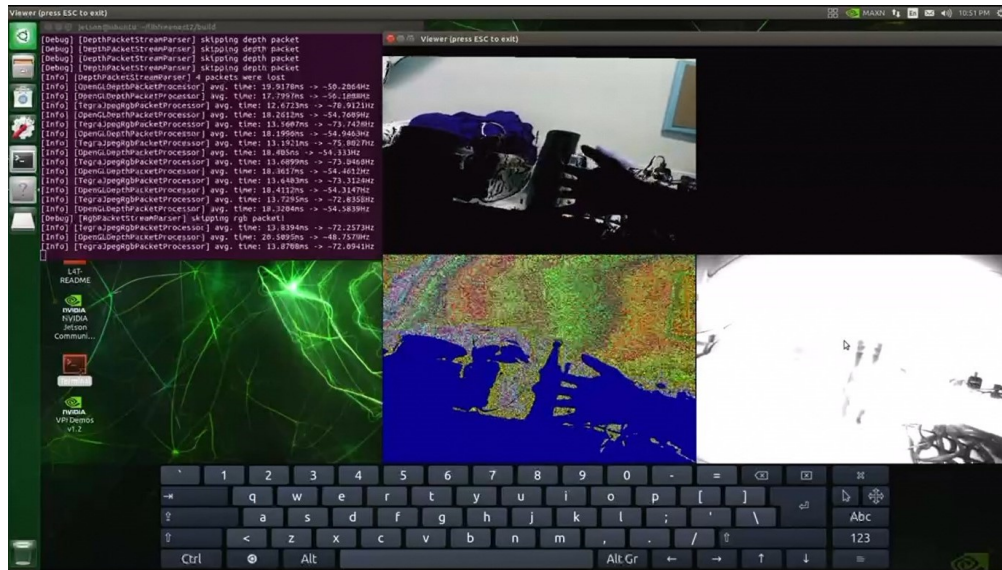


Figure 21: Log of Sensor System

### 3.2 Projection Subsystem

#### 3.2.1 Auto Focusing

Although the current driving capability of the GPIO pins on Jetson Nano is only around 0.5 mA, it is sufficient to drive the H-bridge chip and control the stepper motor. Figure 22 is the output of one of the GPIO pins on Jetson development board, which shows acceleration, steady speed and deceleration periods.

The stepper motor can control the position of the sliding table correctly and accurately. The sliding table can move to the desired position and stop at the correct position. The motor can also move in the opposite direction. The error of the movement is within 0.5 mm.

#### 3.2.2 Projector-Camera System Calibration

The verification of the projection system is first achieved by checking the reproduction error of the projected image. In Figure 23, 19 corner points are detected as valid datapoints

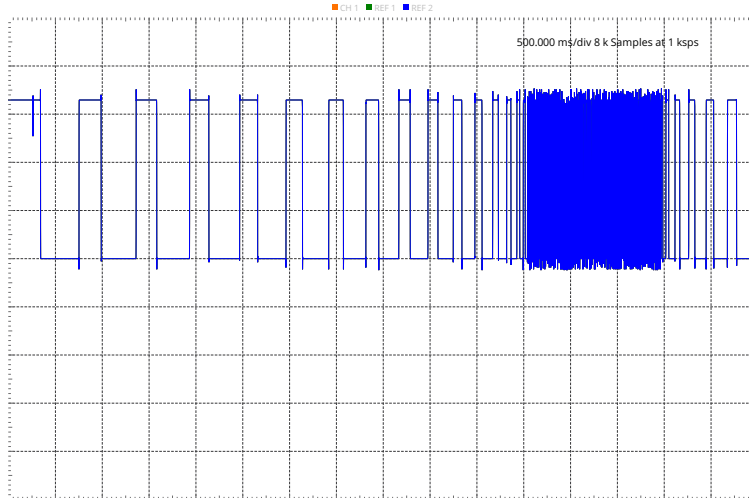


Figure 22: PWM Output of Jetson Nano

and used for projection matrix estimation. Below is the projection matrix estimated with a projection error of 0.7941067124646294 using these points:

$$\begin{bmatrix} -2.86547238e-01 & 2.77225866e-01 & -5.01825389e-01 & 3.49394704e+02 \\ -1.98248169e-02 & 6.11044295e-01 & -5.56395603e-01 & 4.34260722e+02 \\ -3.97751643e-05 & 7.87623720e-04 & -1.39620048e-03 & 1.00000000e+00 \end{bmatrix}$$

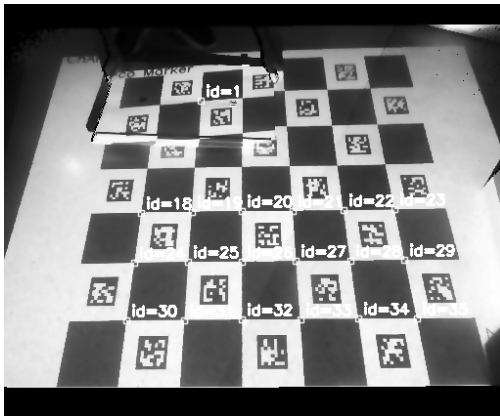


Figure 23: Calibration of Projection System and Results

The effectiveness of calibration process is also verified by checking the alignment of an elevated surface with the topography projected on it, considering that this system is built to deal with the alignment of projected image with objects at different heights. Here a box with height of 11 cm is placed in front of the projector. The error of the alignment is within 1 cm for both the top and bottom of the box. The result is shown in Figure 23.



The speed of the calibration process is verified by timing the calibration process. The calibration process takes 0.17 ms to complete over 50 consecutive runs, which is within the acceptable range of time for the calibration process.

### 3.3 Database and Wireless Display Subsystem

For database as well as the web page, testing is rather intuitive.

In order to testing database functionality, which specifically are Read, Write and Delete, , another `DBreader.py` is written so that directly viewing contents of the database is available. All contents will be unzipped into a folder named output, and by visually checking the number of frames captured as well as the contents of each figure, it is easy to determine whether Writing into the database is correct or not. Reading functionality is checked by clicking the timestamp button on the web page after capturing frames. If the frame correctly shows up on the desired area, then Reading functionality is good. There are two ways to check whether Delete is implemented correctly or not. First of all, by clicking delete button beside the timestamp button, the corresponding line shall be erased from the page. Secondly, `DBreader.py` is used before and after the delete button is clicked, and if the number of frames decreases, then the delete functionality is checked. There is hard max number of frames can be stored in the database. After reaching the hard cap, newest frame will replace the most-outdated frame. Using `DBreader.py` to twice on the contents of the database can decide whether this functionality is implemented right or not.

The web page is designed to be responsive so that every element can comfortably scale up and down based on display devices of users. If the user opens the web page on mobile devices, functional buttons including Capture frame, timestamp and delete shall work just like on PC. The Positions of every element shall move accordingly to provide a comfortable user experience. The top navigation line shall fix on top and when scrolling down the page it is always visible. On the menu list in the navigation bar, users can be able to click the titles and jump immediately to corresponding areas. For example, by clicking the check button in the menu list, user shall be able to jump to the area where the frame list and the stored image display area is. The QRcode shall have consistency that whether user scans from screen of a PC or a phone the web page can jump out correctly. The livestream area shall display the contours correctly and constantly. The Back to Top button on bottom of the page shall correctly scrolling the page to top after being clicked.

### 3.4 Structure Subsystem

We think it is necessary to carry out tolerance analysis on the mechanical structure of the sandbox. We have confirmed that the model using ABS plastic can fully meet the functional needs of the AR sandbox. The sample we used was produced using 3D printing, which can almost perfectly reproduce the CAD model. However, such accurate production is difficult to achieve in actual production. Whether it is the production method of 3D

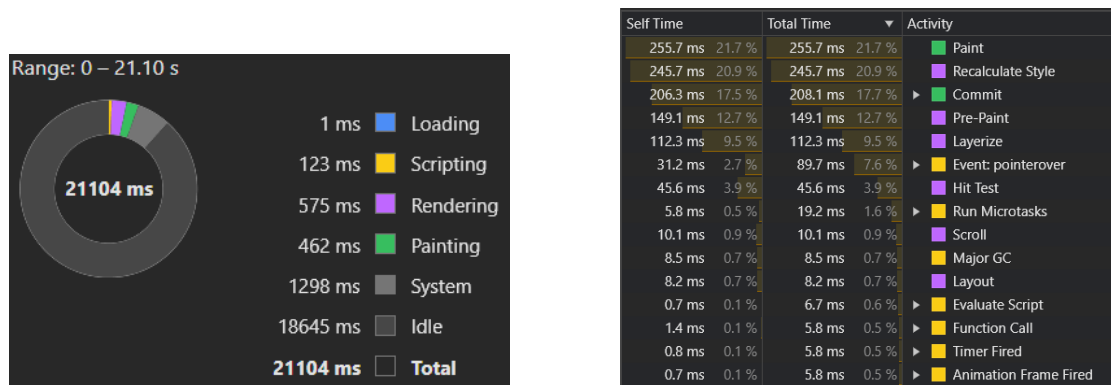


Figure 24: Performance Summary and Call Tree when using web page

printing, or the selection of ABS materials will increase the cost of the product. Therefore, our tolerance analysis focuses on whether structural errors and the influence of different materials on the performance of our products are acceptable.

First of all, in actual production, the production method with a high probability is injection molding. This is suitable for fast, large-scale production, and can effectively reduce costs. However, the biggest drawback of the injection molding process is that the accuracy is not as good as 3D printing. According to the data, the typical tolerance for injection molding is usually  $\pm 0.1$  mm, while the very strict tolerance is  $\pm 0.025$  mm [16]. In order to ensure the results were convincing, I enlarged the tolerance of 0.1 mm tenfold in the analysis. This means that the thickness of all parts of the model will be reduced by 2 mm.

The second is the choice of materials. Even though acrylonitrile butadiene styrene (ABS) is a good choice, the significantly lower price of polystyrene (PS) and polypropylene (PP) for the same weight can undoubtedly reduce production costs. It is worth noting that the injection molding process is also suitable for acrylic (PMMA), nylon polyamide (PA), polycarbonate (PC), polyethylene (PE), thermoplastic elastomer (TPE), Thermoplastic polyurethane (TPU) and other common materials. Some of these materials are too expensive (e.g., PE, whose price is almost 1.5 times of ABS), while others have obvious defects in structural strength (e.g., TPE is too ductile). So we don't take that into consideration. For PP and PS, we ignore the process of filling glass fibers to increase their strength. Although this approach is often used in daily production, it reduces the universality of our analysis.

To sum up, the model of thinner PP and PS materials will be simulated to confirm that the design exhibits very good tolerance. The part chosen for analysis is the separable part with the most hollow sections, the highest pressure, and the greatest susceptibility to external disturbances. If this part performs adequately, it indicates that the other parts are functioning effectively.

Starting with a control group, which represents what is considered a perfect model, an equivalent force of 5 kg was added to the groove to simulate the weight of the projector

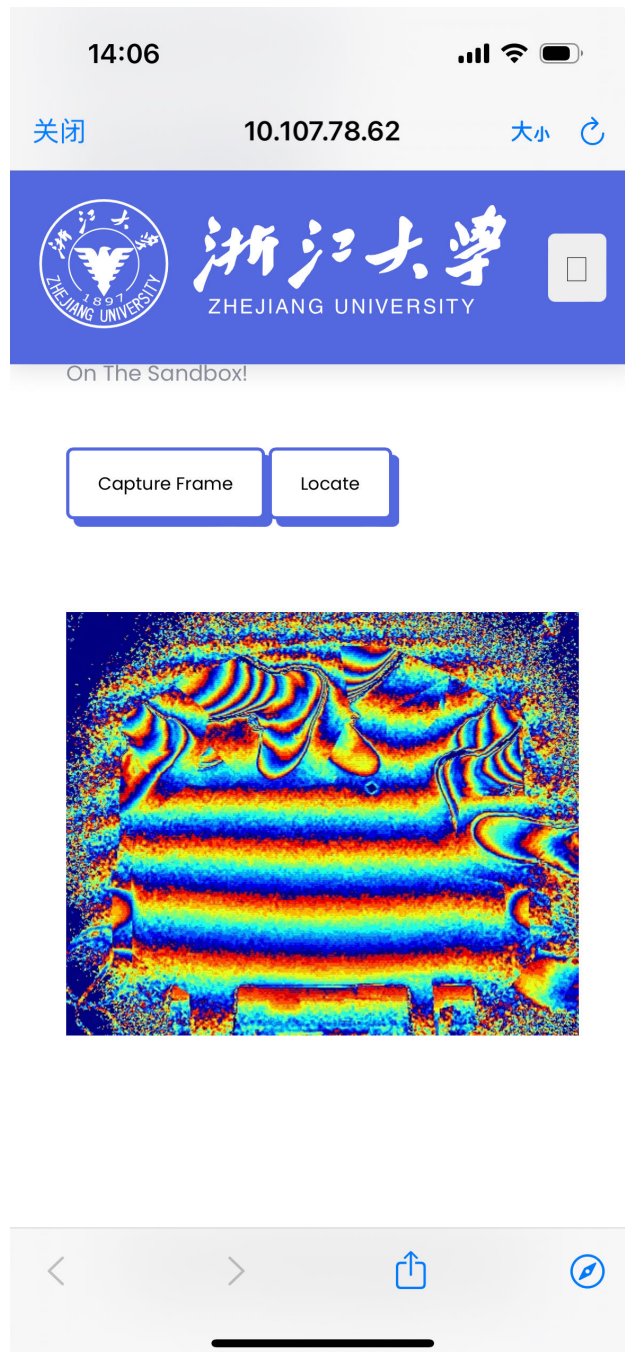


Figure 25: Responsive web page testing on iPhone

being used. The force applied to the edge consists of two parts. Firstly, this position will hold a camera, and secondly, after analysis, it is determined that this position is the most prone to deformation, making it the most suitable point for simulating external forces. As shown in Figure 33 and 34, a downward force of 500 N was applied to this position to simulate the worst-case scenario. It is found that under such conditions, the pressure and deformation of the detachable part remain negligible.

Next is the simulation results of PP material with error model. According to the results shown in Figure 31 and 32, it can be concluded that under this condition, even if the value of the deformation increases a lot compared with the original, it is still small enough to be ignored for the whole model.

Finally comes to the simulation results of PS material in Figure 29 and 30 and there are errors in the model. We are pleasantly surprised to find that the PS material shows increased structural strength. This proves that the design of the structure is completely feasible.

In conclusion, we believe that the implementation feasibility of our design is very strong. It can tolerate errors generated in industrial production and can also be applied to a variety of materials. So overall it was a very successful design. However, there is one more thing to add about the selection of materials. An important audience for our sandbox design is children, and safety and health factors must be fully considered. Although PP and PS are very satisfactory in terms of strength and price, there are some defects in physical properties. PP and PS are extremely easy to burn, and PP will decompose harmful aromatic hydrocarbons such as benzene and toluene at high temperatures [17]. Therefore, from a designer's point of view, I still recommend the use of ABS plastic.

## 4 Costs

The cost of our project consists mainly of the cost of the hardware and the cost of labor.

### 4.1 Labor

The labor cost is calculated based on the assumed salary of \$20 per hour. There are four members in our team. The time spent on the project for each people is estimated to be 25 hours per week. The total duration for this project is 10 weeks. Thus, the total labor cost would be:

$$\text{Labor cost} = 20 \times 25 \times 10 \times 4 = \$20000$$

### 4.2 Components

The cost of the components is shown in Table 1. The non-standard parts mainly include the structure part, which we plan to produce in the school laboratory by 3D printing. The material will be ABS plastic, and the filling rate should be between 30% and 50%. Since it has not yet been officially put into production, it is difficult for us to estimate the exact quantity of consumables. Therefore, we will not include the cost of consumables in the cost estimate.

name	description	manufacturer	part #	quantity	cost (CNY)
JETSON NANO B01 with 5V, 4A power supply	NVIDIA Jetson Nano developer motherboard	NVIDIA	B01	1	1199
Children's Plastic Sandbox	sandbox to support the sand	Foshan Liasheng Plastic Manufacturing Plant	No.8 (40 cm * 80 cm * 12 cm)	1	59
EPSON CU600X Ultra Short Throw HD Projector	Projector used to project contour maps onto a sand table	EPSON	CU600X	1	584.5
Microsoft Kinect V2 camera	Gather images of the sandbox as well as depth information	Microsoft	V2	1	770

RTL 8822CE Wireless Network Card	network card for wireless connection for Jetson Nano	REALTEK	8822CE	1	25
Delta electronics brushless fan	fan for cooling the Jetson Nano	Delta Electronics	AFB0412VHA	1	5
superfine river sand	Placed in a sandbox for children to use	TIAN-SHISHUIZU	unknown	3*5 kg	53.4
Screw Type Terminal Block	Fixed input and output wires	KEFA	C474881	4	2.5
Dual Brushed DC Motor Drive Circuit	Stepper motor driver chip	mixic	C5119044	1	1.682
Desktop retractable stand	Potential projector support components	Chenxin Digital Accessories	1/4 inch - external teeth	1	23.4
Linear Sliding Table with Stepper Motor	Stepper motor for adjusting the focus of the projector	Ouli Transmission	28 T6*1-50mm	1	85

Table 1: Cost of the Components

Total cost of the components is:

$$\text{Component cost} = 1199 + 59 + 584.5 + 770 + 25 + 5 + 53.4 + 2.5 + 1.682 + 23.4 + 85 = 2809.02$$

## 5 Ethics and Safety

### 5.1 Ethical Issues

High luminance projector would be energy-consuming. It is possible that the user forget to turn of the AR sandbox after use. This would bring potential environmental issue. As stated in ACM code of Ethics and Professional Conduct, "human well-being requires a safe natural environment. Therefore, computing professionals should promote environmental sustainability both locally and globally"[18]. In order to achieve this goal, we would explore ways to lower the power consumption during use and automatically detect the leave of users.

Also, consider the fact that people with different skin colors may have different reflection rates, the depth camera may not work well for people with dark skin. This would be a potential ethical issue and we need to consider seriously, as mentioned in IEEE Code of Ethics that "to treat all persons fairly and with respect, and to not engage in discrimination based on characteristics such as race, religion, gender, disability, age, national origin, sexual orientation, gender identity, or gender expression"[19]. We would consider adding more light sources to make the reflection more uniform.

### 5.2 Safety Concerns

There are a list of safety concerns towards our project:

- Electric shock: the projector and the sensor would be connected to power supply. The structure needs to be designed to prevent users from touching the wires and the power supply.
- PCB exposure: Using only PCB to design a 220 / 5 V power converter may not be the safest option. The inner connection on the foot may be broken, causing electricity leakage or short circuit.
- High luminance light: the projector would be high luminance to display clearly on sand considering sand as a diffuse reflection surface. Users might look directly to the projector accidentally that might cause harm to eyes. We would consider adding human eye detection function to turn off the projector when necessary.
- Laser: the depth camera use laser to help measure the distance. We choose an off-the-shelf depth camera whose laser safety has been verified by Microsoft and regulation authorities. The laser should be safe for human eyes.
- Step-motor malfunction: we will use the step-motor as part of the control system for auto focusing of the projector. As the whole system will be built from ground up and connected with projector with irregular shape, the step-motor may not be supported well, thus collapsing with PCB and the projector, or entangling into wires.
- Dust hazard: Fine sand may be inhaled by children when it is lifted into the air. In addition, fine dust may obscure the lenses of the projector and the camera, causing

interference with the input data and an unattractive projected image.

- Breach of structural integrity of the projector: In order to realize the auto focusing function, we will tear down parts of the purchased projector and punching holes on it, thus making the projector's air vents and the focusing lever exposed. This process may cause more dust entering the projector doing damage.
- Falls or deformation: the sand in the box could be heavy. After adding about 25 kg of sand into the box, center of mass will be lifted up, resulting in greater potential of falling down. The structure needs to withstand the weight of sand and consider users leaning on it. We performed simulation on the structure to ensure its stability shown in Figure 26 and Figure 27.

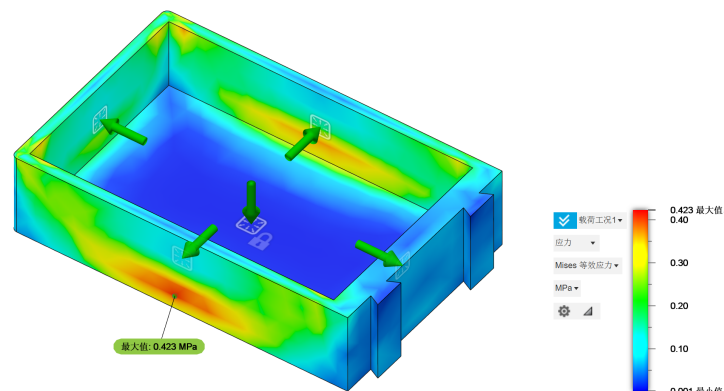


Figure 26: Bottom Pressure Simulation of the Sandbox with Sand in the Box

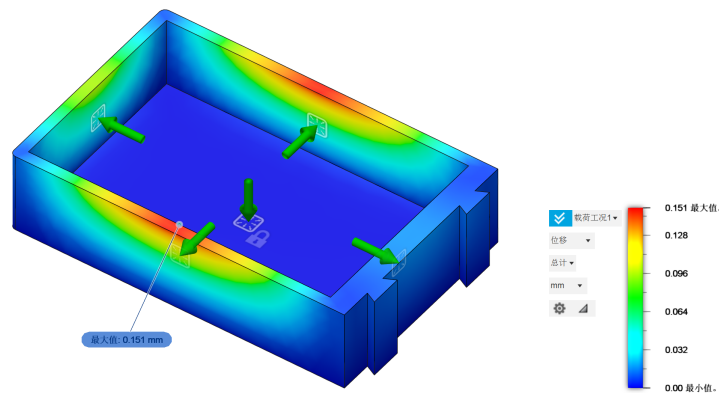


Figure 27: Side Pressure Simulation of the Sandbox with Sand in the Box



## 6 Conclusion

This redesigned AR sandbox has a faster response, safer design and better human machine interaction than existing solutions. We wish it can be used for education and make kids interested in geology in an interactive way. We also hope that the AR sandbox can be used in other fields such as urban planning, water management, and environmental protection. We believe that the AR sandbox can be a useful tool for education and research in the future.

## References

- [1] O. Kreylos, *Oliver Kreylos' research and development homepage - Augmented Reality Sandbox*. [Online]. Available: <https://web.cs.ucdavis.edu/~okreylos/ResDev/SARndbox/>.
- [2] F. Wellmann, S. Virgo, D. Escallon, *et al.*, "Open AR-Sandbox: A haptic interface for geoscience education and outreach," *Geosphere*, vol. 18, no. 2, pp. 732–749, Feb. 2022, ISSN: 1553-040X. DOI: 10.1130/GES02455.1. eprint: <https://pubs.geoscienceworld.org/gsa/geosphere/article-pdf/18/2/732/5576191/732.pdf>. [Online]. Available: <https://doi.org/10.1130/GES02455.1>.
- [3] A. Corti, S. Giancola, G. Mainetti, and R. Sala, "A metrological characterization of the kinect v2 time-of-flight camera," *Robotics and Autonomous Systems*, vol. 75, pp. 584–594, 2016, ISSN: 0921-8890. DOI: <https://doi.org/10.1016/j.robot.2015.09.024>. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0921889015002195>.
- [4] L. Yang, L. Zhang, H. Dong, A. Alelaiwi, and A. E. Saddik, "Evaluating and improving the depth accuracy of kinect for windows v2," *IEEE Sensors Journal*, vol. 15, no. 8, pp. 4275–4285, 2015. DOI: 10.1109/JSEN.2015.2416651.
- [5] L. Xiang, F. Echtler, C. Kerl, *et al.*, *Libfreenect2: Release 0.2*, version v0.2, Apr. 2016. DOI: 10.5281/zenodo.50641. [Online]. Available: <https://doi.org/10.5281/zenodo.50641>.
- [6] I. Virgala, M. Kelemen, A. Gmitterko, and T. Lipták, "Control of stepper motor by microcontroller," *Journal of Automation and Control*, vol. 3, no. 3, pp. 131–134, 2015.
- [7] G. Kogan, *Kinect projector toolkit for image mapping and calibration.[online, github].(july 2014)*, 2014.
- [8] S. Mallick, *Homography examples using opencv ( python / c ++ )* —, Jan. 2024. [Online]. Available: <https://learnopencv.com/homography-examples-using-opencv-python-c/>.
- [9] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330–1334, 2000. DOI: 10.1109/34.888718.
- [10] D. Brown, *Close-range camera calibration photogeom*, 1971.
- [11] Zhang, R. Huang, and Z. Zhao, "A new model of rgb-d camera calibration based on 3d control field," *Sensors*, vol. 19, p. 5082, Nov. 2019. DOI: 10.3390/s19235082.
- [12] Shiffman, *Openkinect-for-processing at master · shiffman/openkinect-for-processing*. [Online]. Available: <https://github.com/shiffman/OpenKinect-for-Processing/blob/master/OpenKinect-Processing/examples/Kinect%5C.v2/CameraPointCloud2/CameraParams.pde%7D>.
- [13] M. Hancock and S. Carpendale, "Supporting multiple off-axis viewpoints at a tabletop display," in *Second Annual IEEE International Workshop on Horizontal Interactive Human-Computer Systems (TABLETOP'07)*, 2007, pp. 171–178. DOI: 10.1109/TABLETOP.2007.9.
- [14] S. Garrido-Jurado, R. Muñoz-Salinas, F. J. Madrid-Cuevas, and M. J. Marín-Jiménez, "Automatic generation and detection of highly reliable fiducial markers under occlusion," *Pattern Recognition*, vol. 47, no. 6, pp. 2280–2292, 2014.

- [15] K.-Y. Shin and J. H. Mun, "A multi-camera calibration method using a 3-axis frame and wand," *International Journal of Precision Engineering and Manufacturing*, vol. 13, pp. 283–289, 2012.
- [16] W. Qicai, F. Xiaohui, S. Ban, *et al.*, "Research on defect detection system of injection molding parts based on machine vision," *DefectDetection*, vol. 51, pp. 113–119, 2024, ISSN: 1000-3932. DOI: 10.20030/j.cnki.1000-3932.202401018.
- [17] L. Jun, Z. Jing-Lian, and T. Ji-Bo, "Study on modification of polypropylene plastics," *Petrochemical Technology*, vol. 22, p. 14, 2015, ISSN: 1006-0235.
- [18] D. Gotterbarn, B. Brinkman, C. Flick, *et al.*, "Acm code of ethics and professional conduct," 2018.
- [19] *IEEE Code of Ethics*. [Online]. Available: <https://www.ieee.org/about/corporate/governance/p7-8.html>.

## Appendix A Requirements and Verification Table

Requirement	Verification
The RGB, IR and depth frames should transmit to the processing unit at a frame rate of at least 30 fps.	Run a program with basic frames retrieving and displaying function. Log the frame rate to verify if it reaches desired value.
The depth sensor should provide depth information with a resolution of $512 \times 424$ pixels.	Check the dimension readings of depth data in the code.
The depth sensor should run for a long time without overheating or performance issues.	Run the sensor for at least one hour and log the frame rate. Check if the frame rate keeps above our desired value 30 fps.
The body detection module should be able to process images at a rate of 30 fps. And the latency should be negligible for human eyes.	Set up a test environment where the body detection module processes a continuous stream of images above 30 fps and count number of images processed in a certain period of time.
The body detection module should predict the human body with an accuracy higher than 80% per pixel.	Prepare several labeled images and run the body detection on test datasets to calculate its accuracy.
The topography rendering module should be able to construct a topography map from depth information and project the topography map onto the sand surface.	Build sand models in different shapes and check if the contour line and color is correctly assigned.
The system must ensure that the wireless connection, facilitated by the REALTEK 8822CE network card, is stable and can support multiple users. Also, the database should be updated correctly when multiple users are saving models or loading models at the same time.	Use more than 5 devices to access the website at the same time and each do operations like reloading homepage, saving or loading a model. Then check the connection stability and see if the save and load operations are correct. Each user connected wirelessly should not experience operation delay longer than 3 seconds.
The calibration algorithm should be able to track the projected image alignment with the sandbox and the alignment error should be less than 1 cm when installed with our sandbox.	Use the calibration algorithm to track the projected image alignment with the sandbox. Put boxes with known dimensions on the sandbox and project the image. Check if the offset between the projected image and the boxes is within the acceptable range.

<p>The auto focusing algorithm should be able to adjust the focus of the projector according to the height of the sand surface.</p>	<p>Use the auto focusing algorithm to adjust the focus of the projector according to the height of the sand surface. Adjust the height of sand table and project image with thin lines with one pixel wide onto it. Check if the projected lines are clear.</p>
<p>The powering subsystem should be able to provide power to the sensor, processing, and display subsystems with sufficient power and voltage.</p>	<p>Connect the sensor, processing, and display subsystems to the powering subsystem. Check if the sensor, processing, and display subsystems are powered on. Put all subsystems at work for more than an hour, check if sufficient power is provided and no components failed.</p>
<p>Design a sturdy and stable support structure capable of safely holding the weight of the AR sandbox (25 kg) and the projector set(5 kg) without risk of collapse or instability during use. The weight of the sandbox and projector should be distributed evenly across the support structure.</p>	<p>Conduct load testing to verify that the support structure can safely withstand the combined weight of the AR sandbox, sand and the projector set. Apply enough sand to simulate real-world usage scenarios. Check for any signs of stress concentration or uneven loading that could compromise stability.</p>
<p>Ensure that the design minimizes potential hazards for children interacting with the AR sandbox. Eliminate sharp edges, protruding components, or other features that could cause injury.</p>	<p>Perform a safety inspection of the support structure to identify any potential hazards or safety concerns. All edges should be rounded, sharp corners be eliminated, and there are no small parts that could pose choking hazards to children.</p>
<p>The projector set on top should be separable, and the separable part should be able to be used separately and can adapt to different use environments and scenarios.</p>	<p>Conduct a practical test where the projector is detached from the support structure. Verify that the separable part functions independently and can be easily moved to different locations or setups as needed. Then assess the reattachment process to confirm that it can be securely and quickly reconnected to the support structure without compromising stability or safety.</p>

The restoration system should achieve the re-tiling of the sand surface in a simple way.

Conduct a practical test where the sand is deliberately disturbed or unevenly distributed. Use the restoration system and observe its ability to re-tilt the sand surface to a uniform level. Then evaluate the consistency and smoothness of the restored sand surface to by projecting topography onto the sand.

# Appendix B PCB Design of Stepper Motor Drive Board

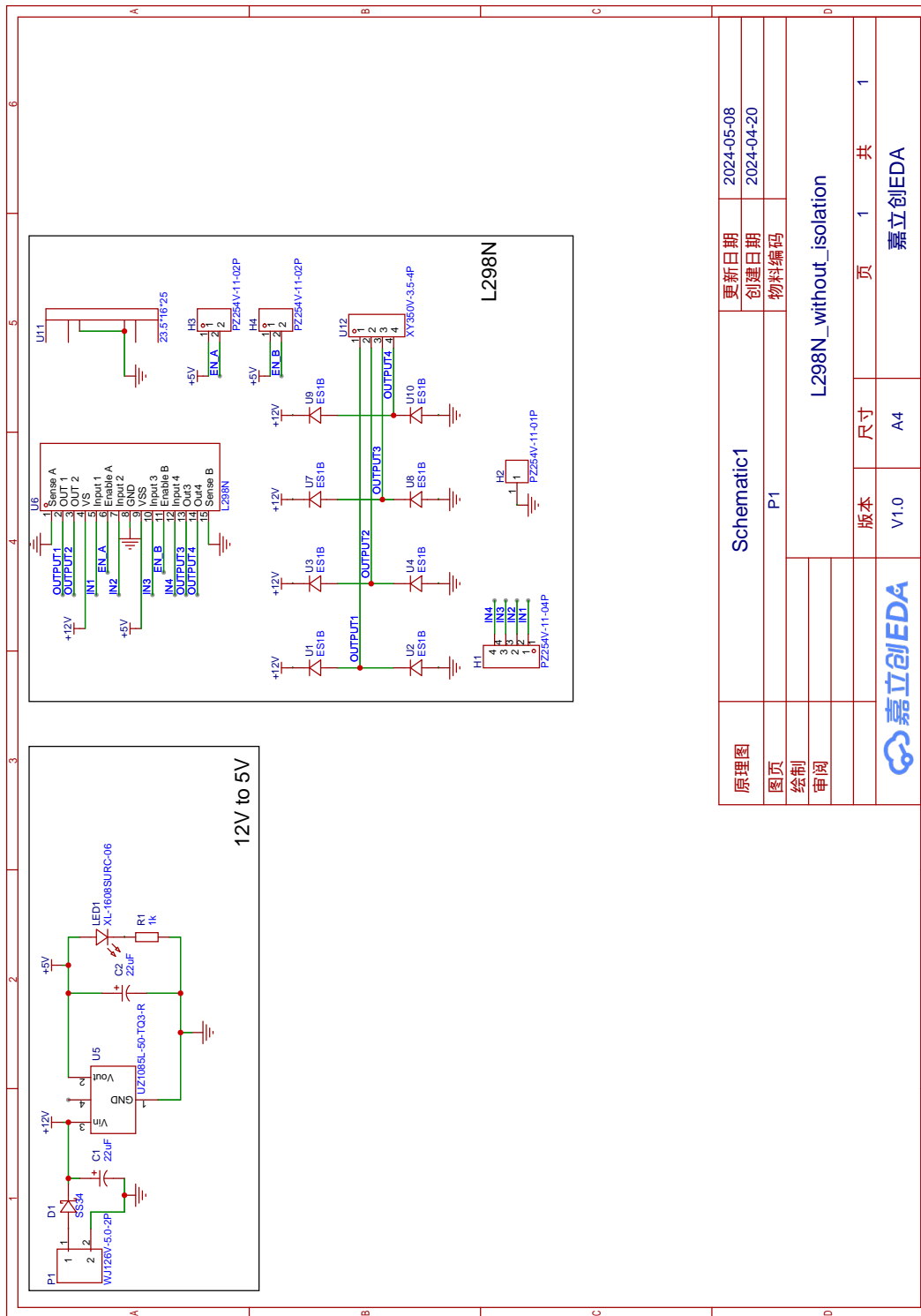


Figure 28: Circuit Design of Stepper Motor Drive Board

更新日期	2024-05-08		
创建日期	2024-04-20		
物料编码			
Schematic1			
P1			
L298N_without_isolation			
版本	尺寸	页	共
V1.0	A4	1	1
嘉立创EDA			
嘉立创EDA			

## Appendix C Pseudocode for Motor Control

---

### Algorithm 2 Motor Function Pseudocode

---

```
1: procedure MOTOR(distance, direction)
2:   Import necessary libraries (GPIO and time)
3:   if distance is not valid or direction is not valid then
4:     Exit procedure
5:   end if
6:   Set GPIO mode to BOARD
7:   Define control pins and set them as GPIO output
8:   Define halfstep sequence for the motor
9:   if direction is -1 then
10:    Reverse the halfstep sequence
11:  end if
12:  Calculate total steps required
13:  Define minimum and maximum delay times for motor control
14:  Define acceleration, deceleration and steady time
15:  Initialize counter i to 0
16:  while i is less than total steps do
17:    Get the current halfstep based on i
18:    Calculate the delay time based on the current step number
19:    Output the corresponding value in halfstep
20:    Pause for delay time
21:    Increment i by 1
22:  end while
23:  Cleanup GPIO
24: end procedure
```

---



## Appendix D Code for Converting Camera's Pixel Coordinates to World Coordinates

```
def depthMatrixToPointCloudPos(z, scale=100):
    C, R = np.indices(z.shape).astype(np.float64)

    np.subtract(R, CameraParams['cx'], out=R)
    np.multiply(R, z, out=R)
    np.divide(R, CameraParams['fx'] * scale, out=R)

    np.subtract(C, CameraParams['cy'], out=C)
    np.multiply(C, z, out=C)
    np.divide(C, CameraParams['fy'] * scale, out=C)

    return np.column_stack((R.ravel(), -C.ravel(), z.ravel() /
        scale))
```

## Appendix E Code for Converting World Coordinates to Projector's Pixel Coordinates

Here is the code for converting world coordinates to pixel coordinates in the projector. This is optimized for speed by using vectorized operations in NumPy and changing the data types to float32.

```
def cameraToProjector_f32(color, depth, P_projector):
    if P_projector is None:
        return color

    points_in_world = depthMatrixToPointCloudPos(depth.astype(np.
        float32))

    points_in_world = points_in_world.reshape(-1, 3).astype(np.
        float32)
    world_points_h = np.hstack((points_in_world, np.ones((
        points_in_world.shape[0], 1), dtype=np.float32)))

    points_projector = np.dot(P_projector.astype(np.float32),
        world_points_h.T).T
    points_projector = points_projector[:, :2] / points_projector
       [:, 2:]

    color_projector = np.zeros((DISPLAY_HEIGHT, DISPLAY_WIDTH, 3)
        , np.uint8)

    valid_indices = ~np.isnan(points_projector).any(axis=1)

    valid_points = points_projector[valid_indices].astype(int)

    valid_indices = (valid_points[:, 0] >= 0) & (valid_points[:,
        0] < DISPLAY_WIDTH) & \
        (valid_points[:, 1] >= 0) & (valid_points[:,
        1] < DISPLAY_HEIGHT)

    valid_points = valid_points[valid_indices]
    valid_indices = valid_indices.nonzero()[0]

    color_values = color.reshape(-1, 3)[valid_indices]

    color_projector[valid_points[:, 1], valid_points[:, 0]] =
        color_values

    return color_projector
```

# Appendix F Simulation Results of the Structure Subsystem

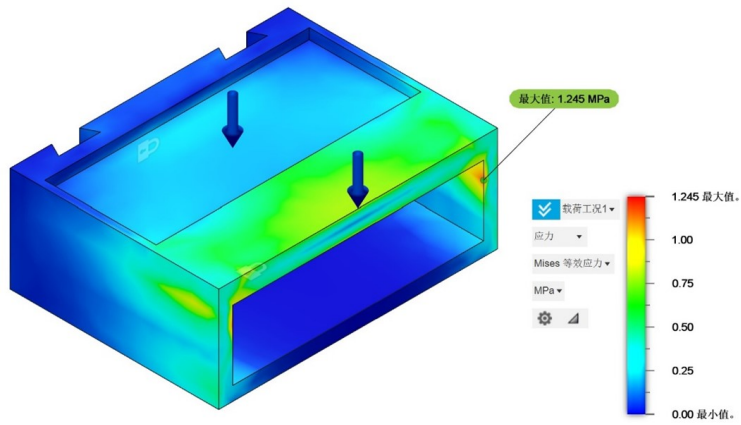


Figure 29: Pressure Analysis on PS Material Defects

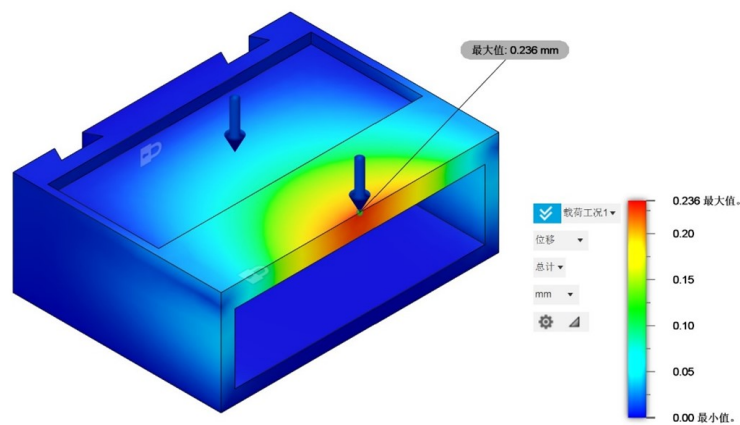


Figure 30: Deformation Analysis on PS Material Defects

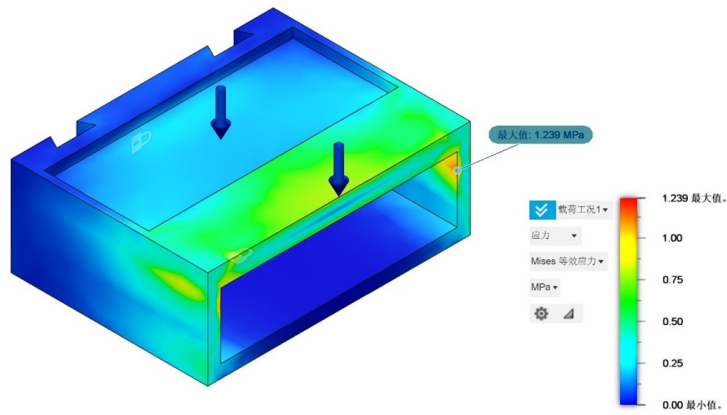


Figure 31: Pressure Analysis on PP Material Defects

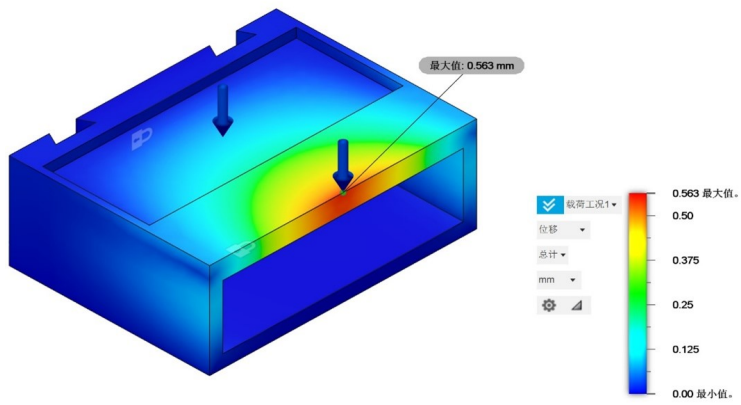


Figure 32: Deformation Analysis on PP Material Defects

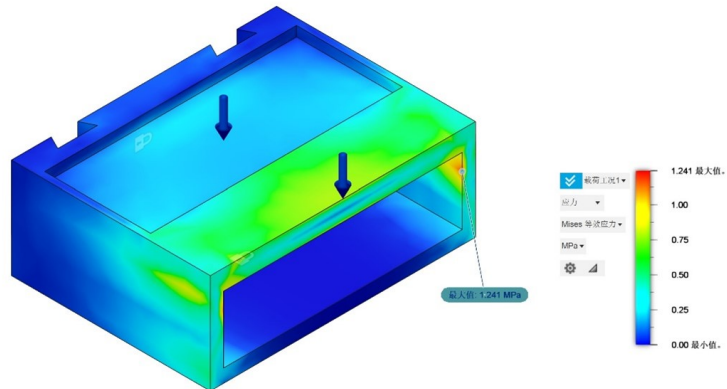


Figure 33: Pressure Analysis on ABS Material Standard Part

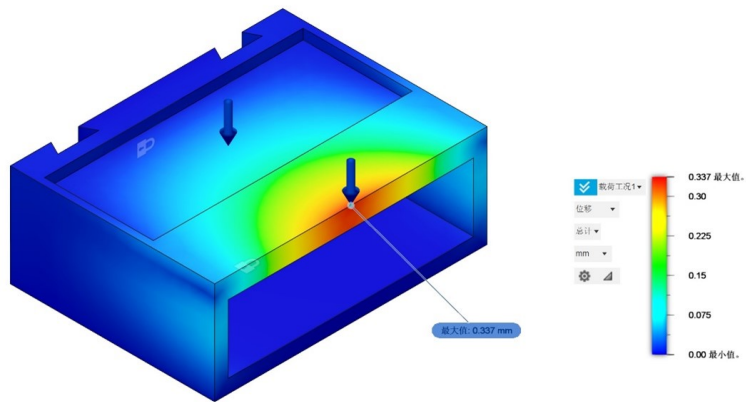


Figure 34: Deformation Analysis on ABS Material Standard Part