Project Anywhere

Wearable and Ubiquitous Computing

Ekta Bindlish
Ekta@vt.edu

Wei Huang
wei90@vt.edu

Stephanie Mencia
smencia@vt.edu

Eric Refour
erefour@vt.edu

Introduction

Literature Review

Ubiquitous computing is a concept in software engineering and computer science where computing is made to appear everywhere and anywhere. A user interacts with the computer, which can exist in many forms such as laptops, tablets, and terminals in everyday objects. This is also described as pervasive computing, ambient intelligence, ambient media, or ‘everyware’. Ubiquitous computing touches on a wide range of research topics, which include distributed computing, mobile computing, location computing, mobile networking, context aware computing, sensor networks, human-computer interaction, and artificial intelligence (Hansmann 2003).

Mark Weiser, a chief scientist at Xerox Parc, is widely considered to be the father of ubiquitous computing. He coined the term in 1988 (In Memory of Dr. Weiser 2008). Xerox PARC, now known as PARC (Palo Alto Research Center Incorporated), is a research and development company in Palo Alto, California. It was founded in 1970 as a division of Xerox corporation. They were responsible for many technological developments including ubiquitous computing. Mark Weiser wrote in Scientific American; “The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it.” (Weiser 1991)

One of the earliest ubiquitous systems was “Live Wire”, also known as “Dangling String”. This was the invention of artist Natalie Jeremijenko. It was a piece of string attached to a stepper motor and controlled by a LAN connection. Network activity caused the string to twitch, yielding a peripherally noticeable indication of traffic. This was installed at Xerox PARC during Mark Weiser’s time there, who was Natalie’s advisor at the time. Weiser called this an example of calm technology (Weiser 1999).

Ubiquitous computing research has focused on building an environment in which computers allow humans to focus attention on select aspects of the environment. It emphasizes the creation of human computer interface that can interpret and support a user’s intentions. This is a fundamental transition that does not seek to escape the physical world and enter cyberspace but rather bring computers and communications to the user (Winter 2008).

With the growing interest of ubiquitous computing, many people are conducting research and attempting to build systems of their own. An example of an institute that is doing this is Virginia Tech.

Overviw of Existing Technology

Mirror Worlds (ICAT)

Virginia Tech has recently built a new Center for the Arts (Figure 1). It is a multifaceted building and a research institute. One of the many ongoing projects at the Moss Art Center is Mirror Worlds (Figure 2). “It is a funded project by the National Science Foundation that researches human interaction by studying behavior and emotion in both the physical and virtual environment. Sensors placed around the Moss Art Center track people in the physical space and then generate digital version of those people at the same locations in a virtual building. People will also be able to access the virtual model of the building online and see both online visitors and tracked representations of real people currently in the physical building. The project hopes to create a shared space, with portals between the physical and virtual, where people can interact with each other and explore the Moss Arts Center. Mirror Worlds has the potential to connect people around the world to computational research on human behavior, interaction and understanding” (MirrorWorlds).

Figure 1: Moss Art Center

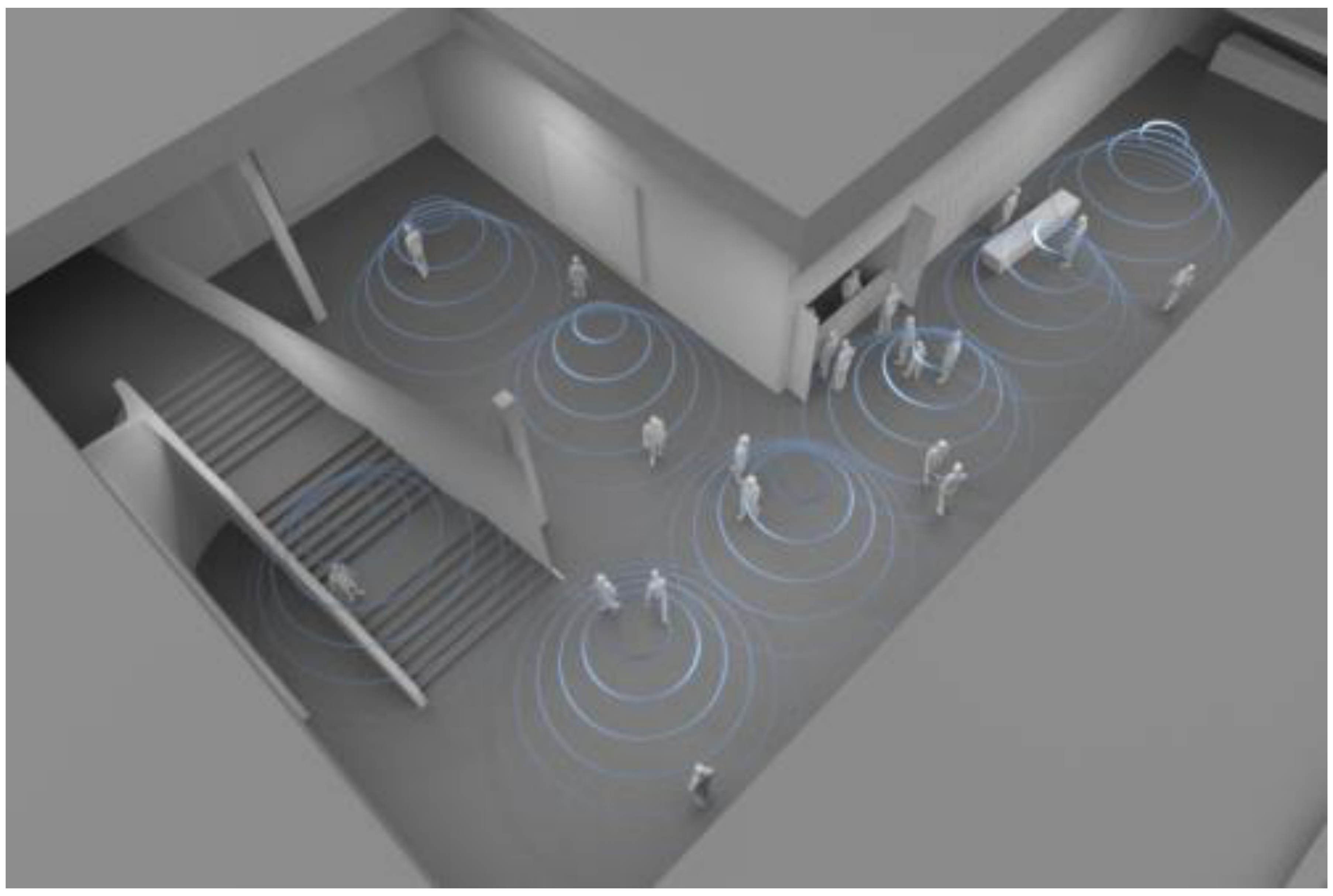

Figure 2: Mirror Worlds

Our goal for our particular project was to design a way for the virtual users to project an image in the real world for the real world users to interact with. This is just a small piece of what is Mirror Worlds. Similar to Claudio Pinhanez, we sought to explore an alternative to create ubiquitous graphical interfaces (Pinhanez 2001). Our idea was to take an LCD projector and combine it with a rotating mirror. The rotation of the mirror would allow the image being projected to be displayed on multiple surfaces in a given room. This approach has been named *Project Anywhere*.

Methods

Needs Analysis

The main function to be expected out of the system was to give the virtual user an easy way to project an image in the real world so that the real world user could view it and possibly interact with. For the system to work as intended, there are many things it has to support. The system has to be intuitive and visually appealing. It needs to be an interface that the average user would be able to grasp quickly. It needs to be able to support multiple users and their different knowledge levels. The systems needed to leverage the already in place Unity 3D source code used in Mirror Worlds. The system must also be robust enough that it can be implemented and used in any room.

Our approach was based on the design created by Pinhanez et al for IBM. We decided to combine a stationary LCD projector with a rotating mirror. The projector would point towards the ceiling of the room. The mirror would be attached to the projector in a way such that the rotation of the mirror would allow the image being projected to be displayed on different surfaces in a given room.

Brainstorming and Initial Sketching

Before we started working on the project we brainstormed on a variety of topics such as the placement of the mirror, the mechanism used to rotate the mirror, a mechanism to allow a user to control the rotation of the mirror etc.

We started with a basic visualization of the system. We had decided that the projector would be pointed towards the ceiling. Keeping this in mind we came up with many different ideas to place the mirror and the motors.

There were many sketches that were done to visualize what the system would look like. These are shown in Figures 3 and 4.

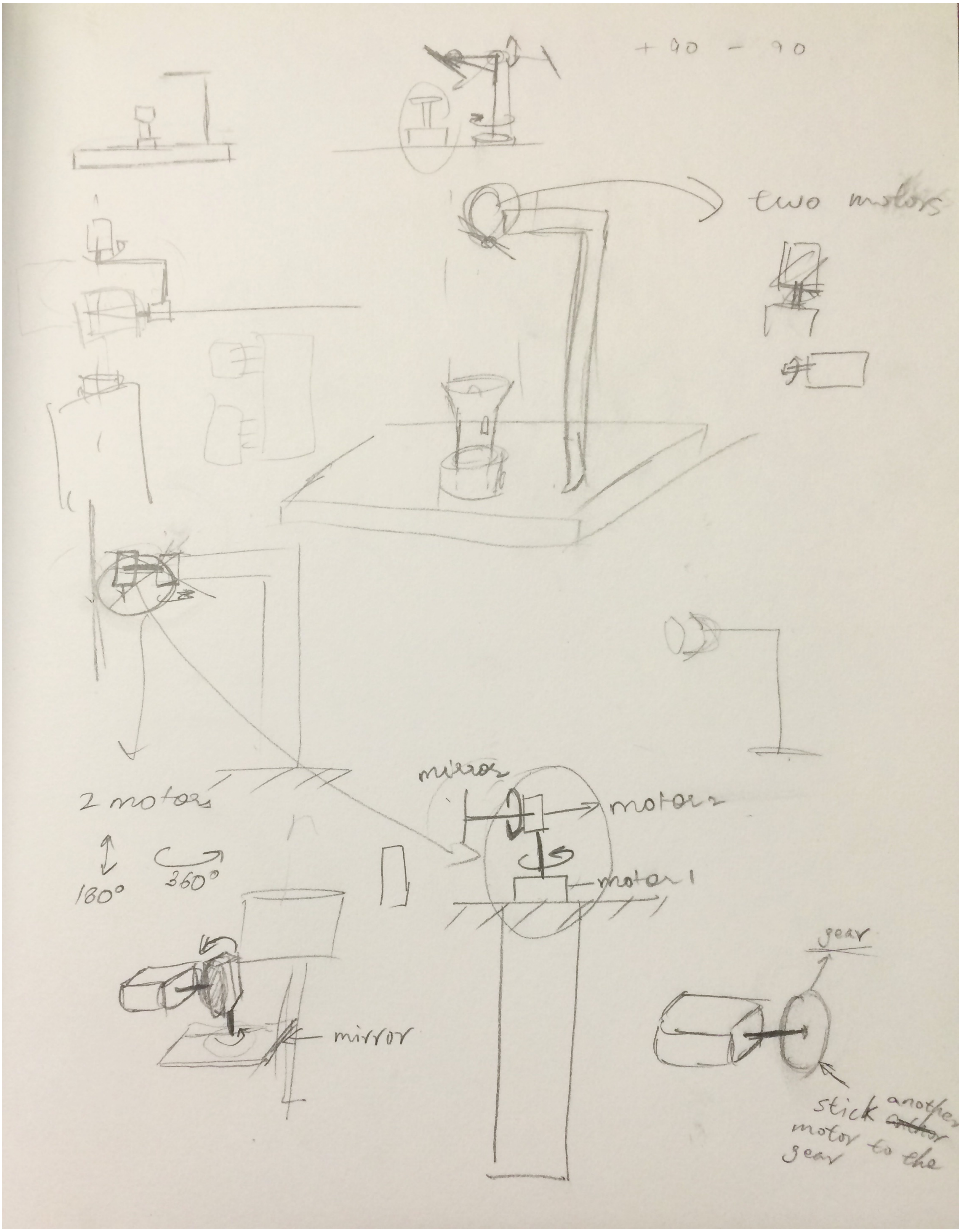

Figure 3: Initial Sketches

From the initial sketches, we were able to see that it was necessary to align the center of the projector and the center of the mirror at all times. Based on this, we came up with an initial design for the rotation harness. We decided against attaching the motors to a fixed arm. Instead, we decided that one motor would be fixed somewhere above the projector. The second motor would be attached to the rotating arm of the first motor.

Based on this we came up with a sketch for the rotation harness (Figure 4).

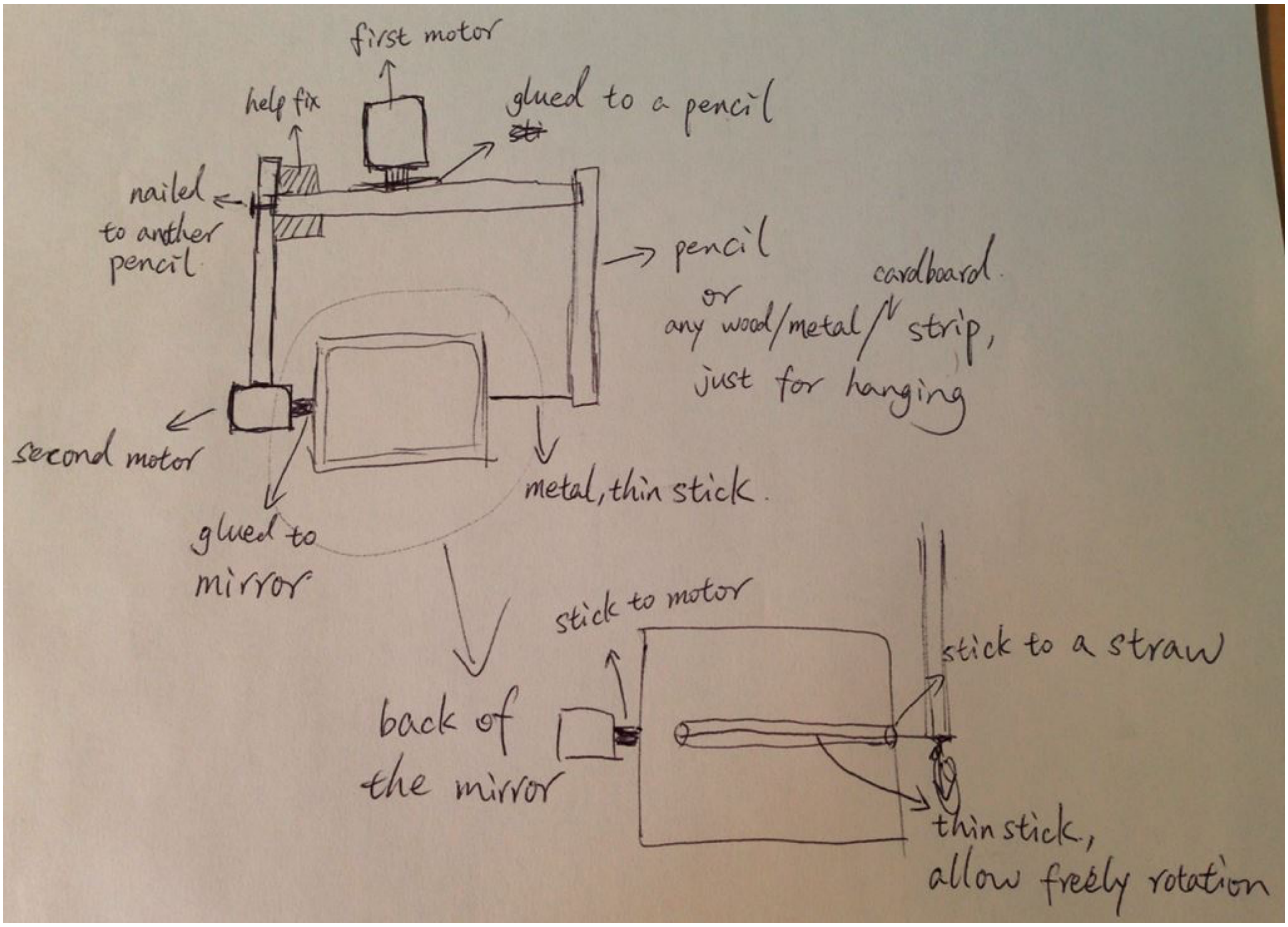

Figure 4: Sketch of Rotating Harness

Prototype

When it came to the prototype, it was decided to use a very simple set up so as to not waste time or money on an extravagant set up that would not work. It was decided to use a cardboard box as a casing for the entire system. A small, lightweight rectangular mirror was chosen to be rotated. A combination of servo and stepper motors were tested to rotate the mirror and various items such as metal, bamboo sticks and straws were used to build the support system skeleton (Figure 5). There were many other design decisions that took place when building the prototype that affected the outcome of our design.

Projector vs. Laser Pointer

For the beginning prototypes of this project, we originally wanted to use a projector to allow for image projection. However, as we proceeded with our prototyping, we decided to go with a laser pointer to test the prototype. We felt that the laser pointer was a better small step forward. After getting the system to work with the laser pointer, we felt it would be easier to eventually generalize the system to a projector.

Rotating Mirror vs. Rotating Projector

It was originally planned to have a rotating projector to aim the projection anywhere in the room. While doing research on rotating projectors, we realized that tackling our project by this angle would be a difficult feat. We continued to do more research and found a paper in which the researchers were trying to do something similar to our project. They used a projector and a mirror in front of the projector to direct the light source. After reading this paper, we decided it would be best to try this approach as a mirror would be much easier to rotate than a projector.

Servo Motor vs. Stepper Motor

When first coming with designs and ideas for the prototype, we decided to order servo motors to move the motors. While beginning to write code for the motors, we realized that it was hard to get the servo motors to stop rotating as they kept rotating on a continuum. We then decided to try stepper motors, seeing as those are easier to write code for to allow a couple “steps” that will move the mirror. Specifically, stepper motors are designed to turn in equal steps each time and would have thus provided a more accurate angle of rotation for the mirror projection. Once coming up with a mechanism to hold the mirror, we realized that the step motors could not support the weight of the mirror and the mechanism, so we had to go back to the servo motor. Code was written to fix the issue of the motor continuously rotating and they were able to support the weight of the entire fixture much better than the stepper motors could.

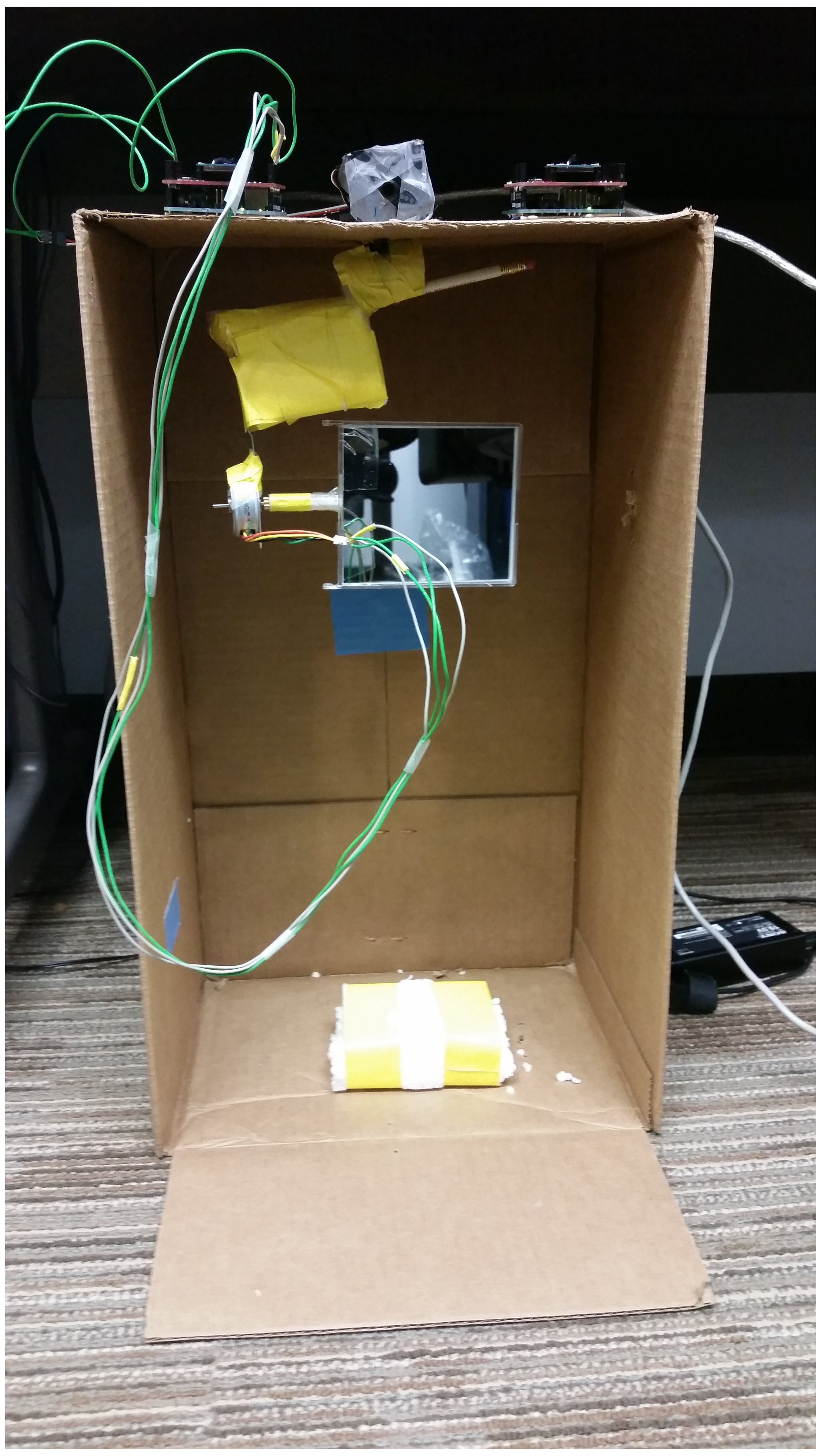

Figure 5: Initial Prototype

Tracking Real World User

When designing the user interface of the system, the issue of tracking the users in the real world came up. The idea was that the user interface would be aware of the location of a user in the real world, and the user in the virtual world would use this information to direct the projection to get the real user's attention. We went through many ideas, such as having the users wear wristbands or tracking their motion via a Kinect. We had originally decided to attempt using the Kinect to track the user's motion and possibly their location.We later learned that mirror worlds already has their own way of tracking users in the real world space. We decided to assume that a tracking system was already in place and that we could later integrate with our prototype system and interface.

Unity3D Interface

A user interface was designed using Unity3D. We chose to use Unity 3D as the existing Mirror Worlds interface is implemented in Unity 3D. The interface allowed for the virtual user to click anywhere, in the virtual rendering of the real world room (Figure 6), and project an image or a light source for the real world user to see.

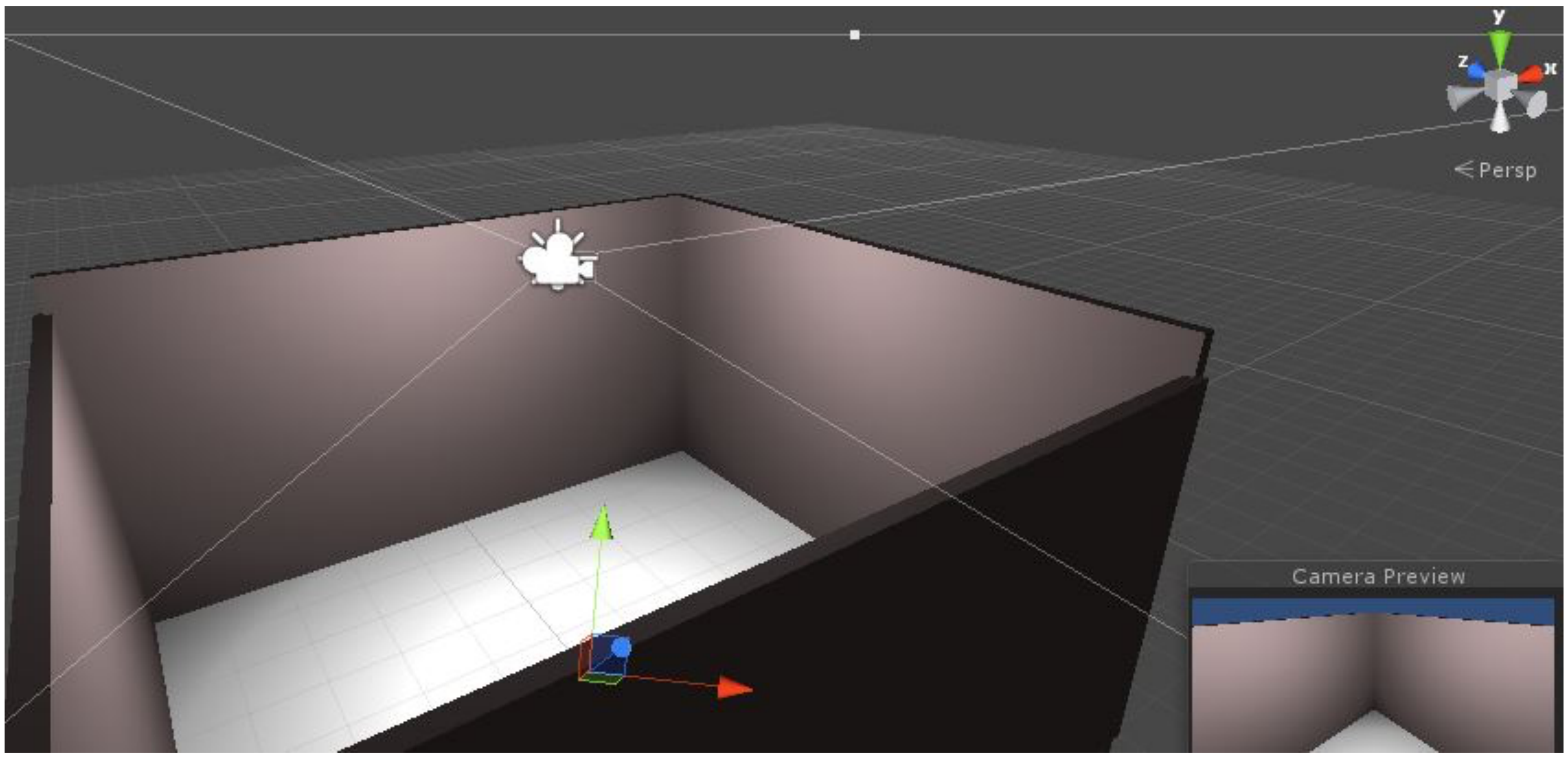

Figure 6: Rendition of real world room in Unity 3D

The size and dimensions of this virtual room will ideally be proportional to the real room. Our system, once calibrated, will scale to any size room. For the time being, only square or rectangular rooms will be scaled properly. The camera in the middle corresponds to the position of our system in real world.

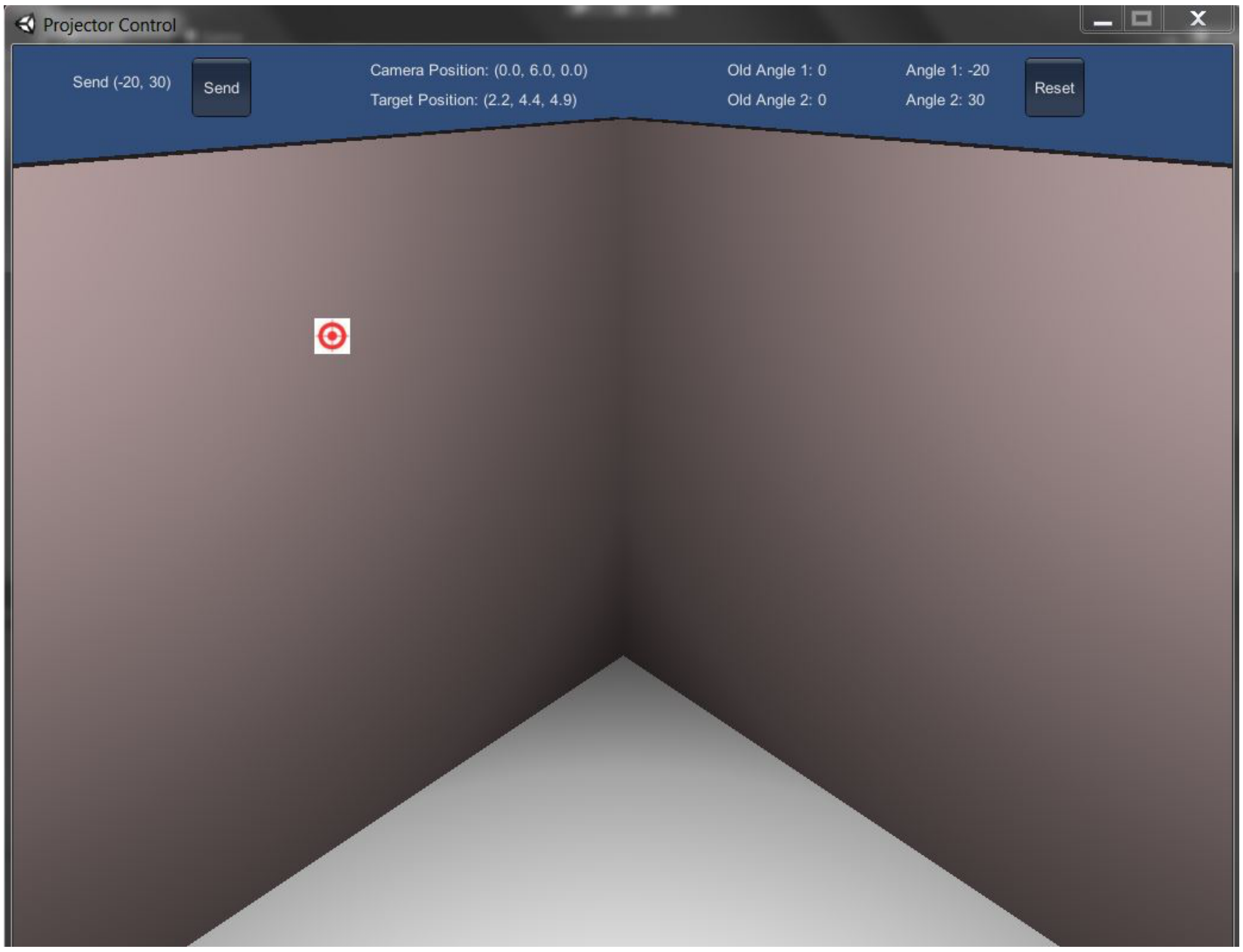

Figure 7: Virtual User Interface on Unity 3D

The virtual user is able to click on the position where they want to send a projection (Figure 7). By simply clicking on the “Send” button, they transmit the parameters, which are needed by the motors, to the Arduino. These parameters are the differences in position between current position, which is being transmitted, and the previous position.

1. The “Reset” button will reset the values to initial position.
   The initial position consists of the mirror facing downward and flat.
2. “Angle 1” is the angle of the horizontal rotation. This is from the initial position to desired target.
3. “Angle 2” is the angle of the vertical rotation. This is from the initial position to desired target.
4. “Old Angle” are the angles of rotation, which refer to previous target.

Angle Calculation

Step 1: The intersection of the wall and the direction/axis of the projector must be calculated first to obtain the 3D world space coordinates of the desired point. This is the ray from the camera in the direction of the point clicked by the user. This is necessary because clicking on the wall only gives us 2D coordinates and 3D coordinates are necessary to actually move the mirror to the proper location.
There are some useful methods provided by Unity3D:

Camera.ScreenPointToRay: This creates a ray in 3D virtual world space, starting from the camera through the position’s (x,y) pixel coordinates on the screen.

Physics.Raycast: This detects where the ray intersects a specific collider.

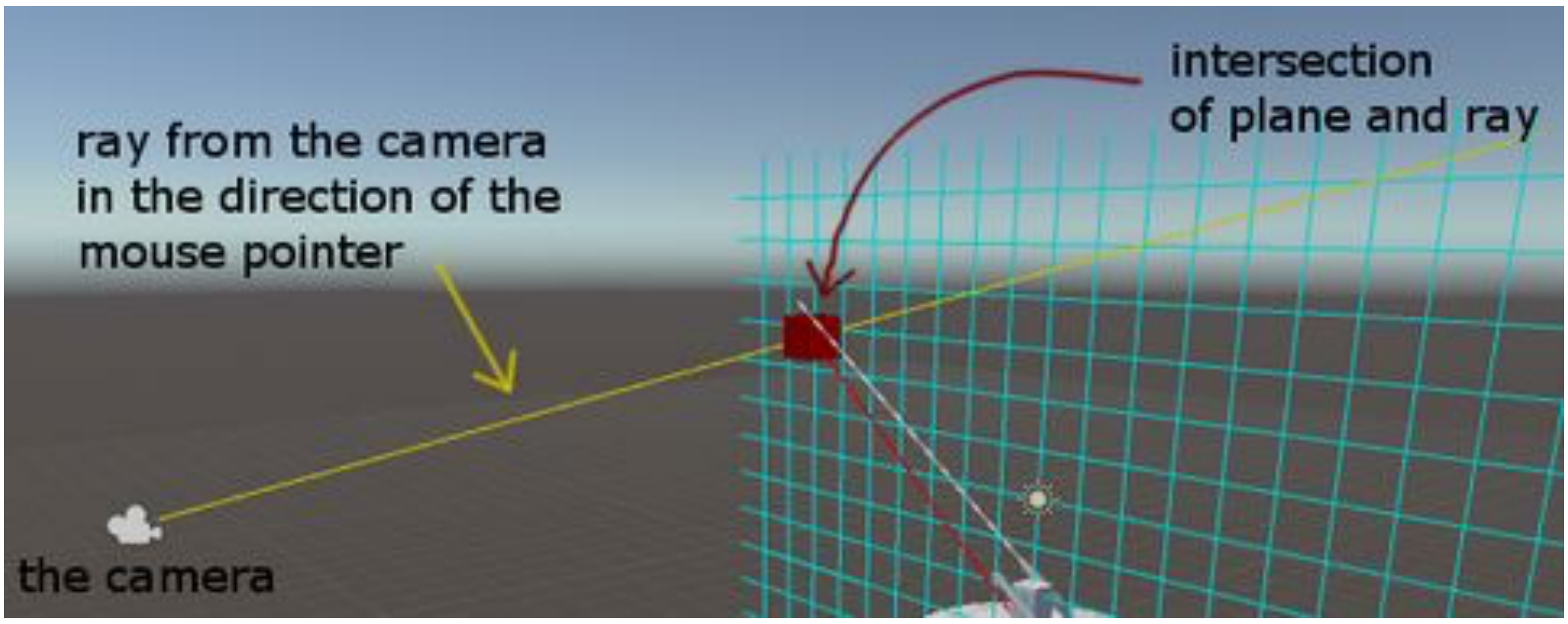

Figure 8: Unity 3D Calculation of Angle

Step 2: Angle Calculation Formulas

1. Angle 1: horizontal rotation

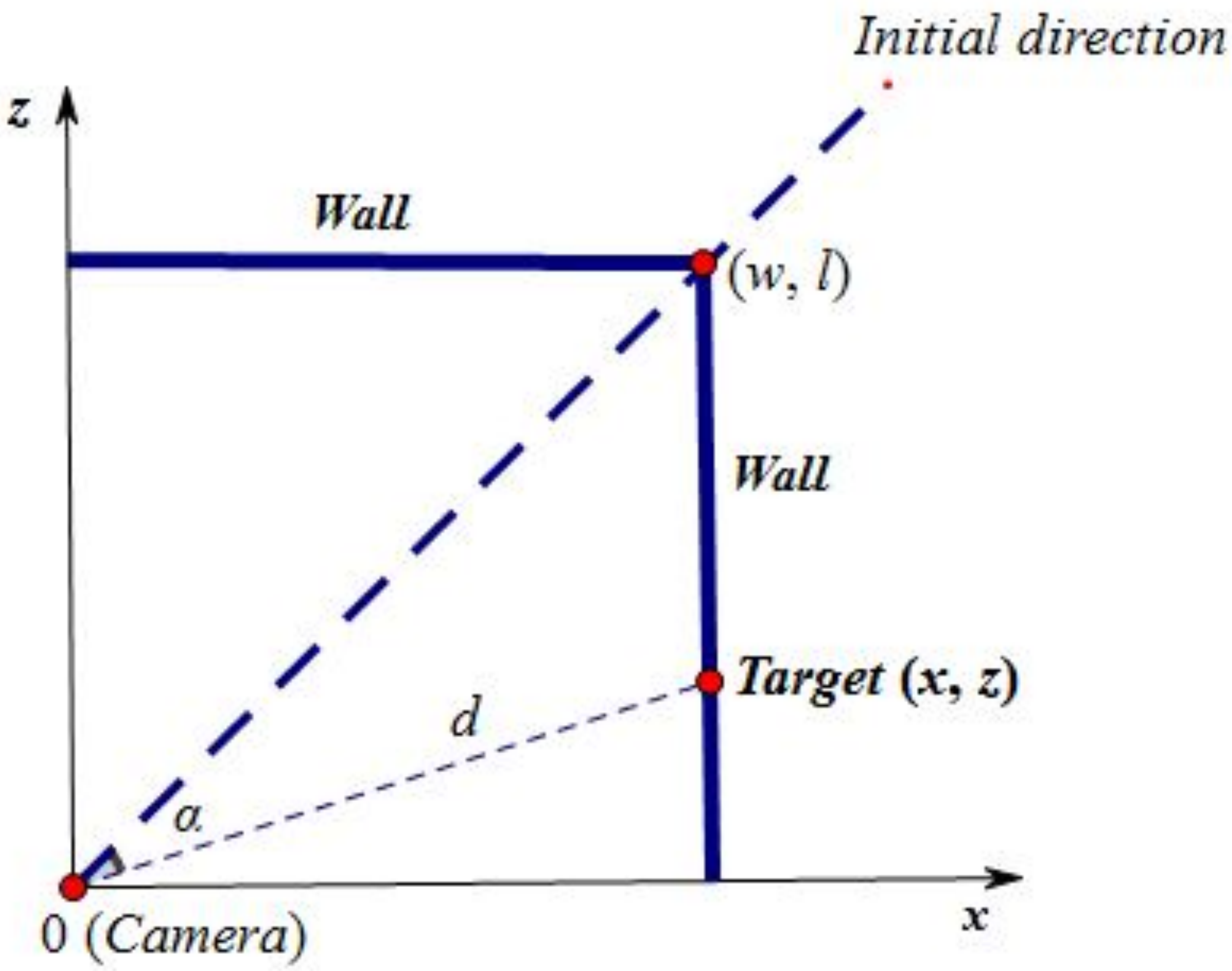

Figure 9: Calculation of angle 1 Diagram

$$\alpha = \arctan\left(\frac{l}{w}\right) - \arctan\left(\frac{z}{x}\right)$$

2. Angle 2: vertical rotation

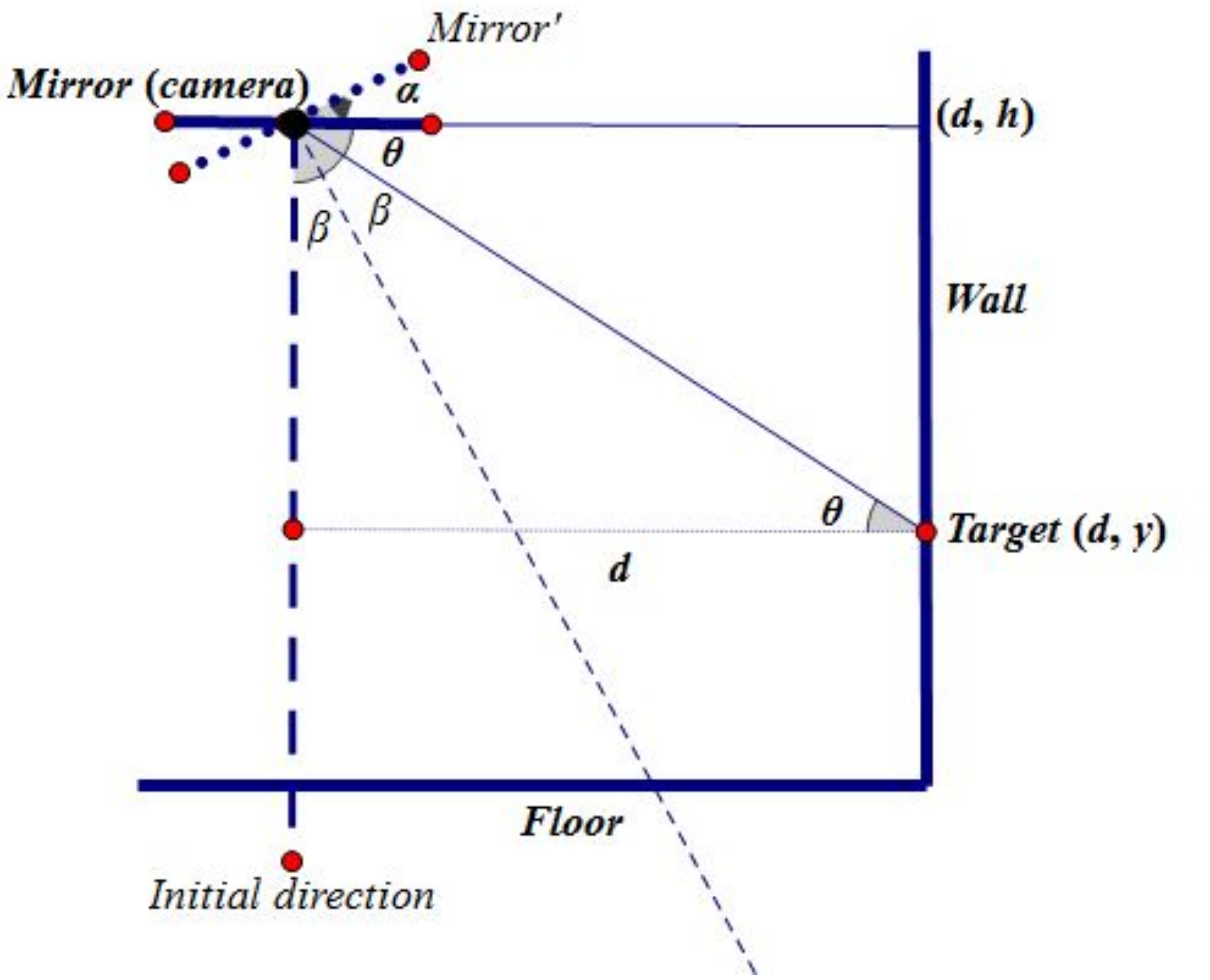

Figure 10: Calculation of angle 2 Diagram

$$d = x^2 + z^2$$

$$\theta = \arctan\left(\frac{h-y}{d}\right)$$

$$\begin{cases} 2\beta + \theta = 90^\circ \\ \alpha + \beta + \theta = 90^\circ \end{cases} \rightarrow \alpha = 45^\circ - \frac{\theta}{2} = 45^\circ - \frac{1}{2} arctan \frac{h-y}{d}$$

Motor Control

The final system consisted of two servo motors connected to an Arduino Uno board with Xbee Shield.

Figure 11 shows the pin connections of the two motors and the Arduino Uno. The signal pin of Motor A is connected to pin 9 of the Arduino. The signal pin of Motor B is connected to pin 10 of the Arduino.

A second Arduino Uno with Xbee Shield was used to transmit the angles of rotation from the user interface to the first Arduino board.

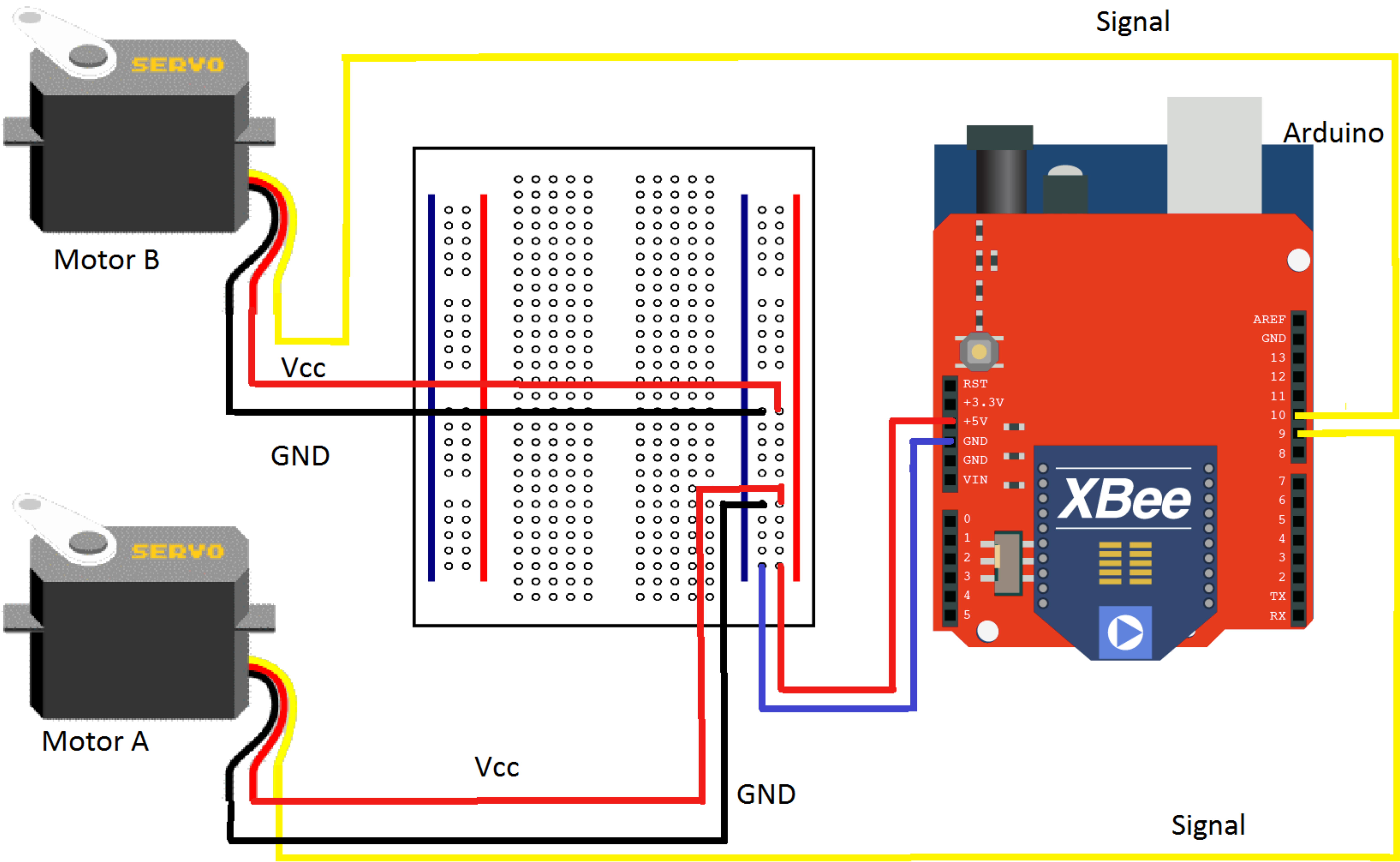

Figure 11: Connection of motors to Arduino Boards

Rotation Harness Design

Based on the previous sketches, we came up with a model as shown in figure 12.

Motor A is fixed to the roof of a cardboard box. It rotates the projected image along the horizontal axis. This motor controls the rotation of the entire rotation harness, and has to bear its weight. The rotation harness has an L shaped arm, and motor B is attached at the end of this arm. This motor rotates the projection image vertically. The length of the rotating arms is adjusted so that the center of the mirror is always aligned with the center of the light source.

A supporting arm is added to the motor harness to help motor B support the weight of the mirror.

Due to the weight limitations of the rotation harness, we had to use lightweight materials to construct the harness. We used bamboo sticks to build the supporting arm. The mirror was attached to motor B using an attachment device that came with the motor. Motor A's rotating arm was made out of a wooden pencil, nailed to a metal piece. Motor B was attached to the end of the metal piece using screws. The whole harness was secured with construction tape.

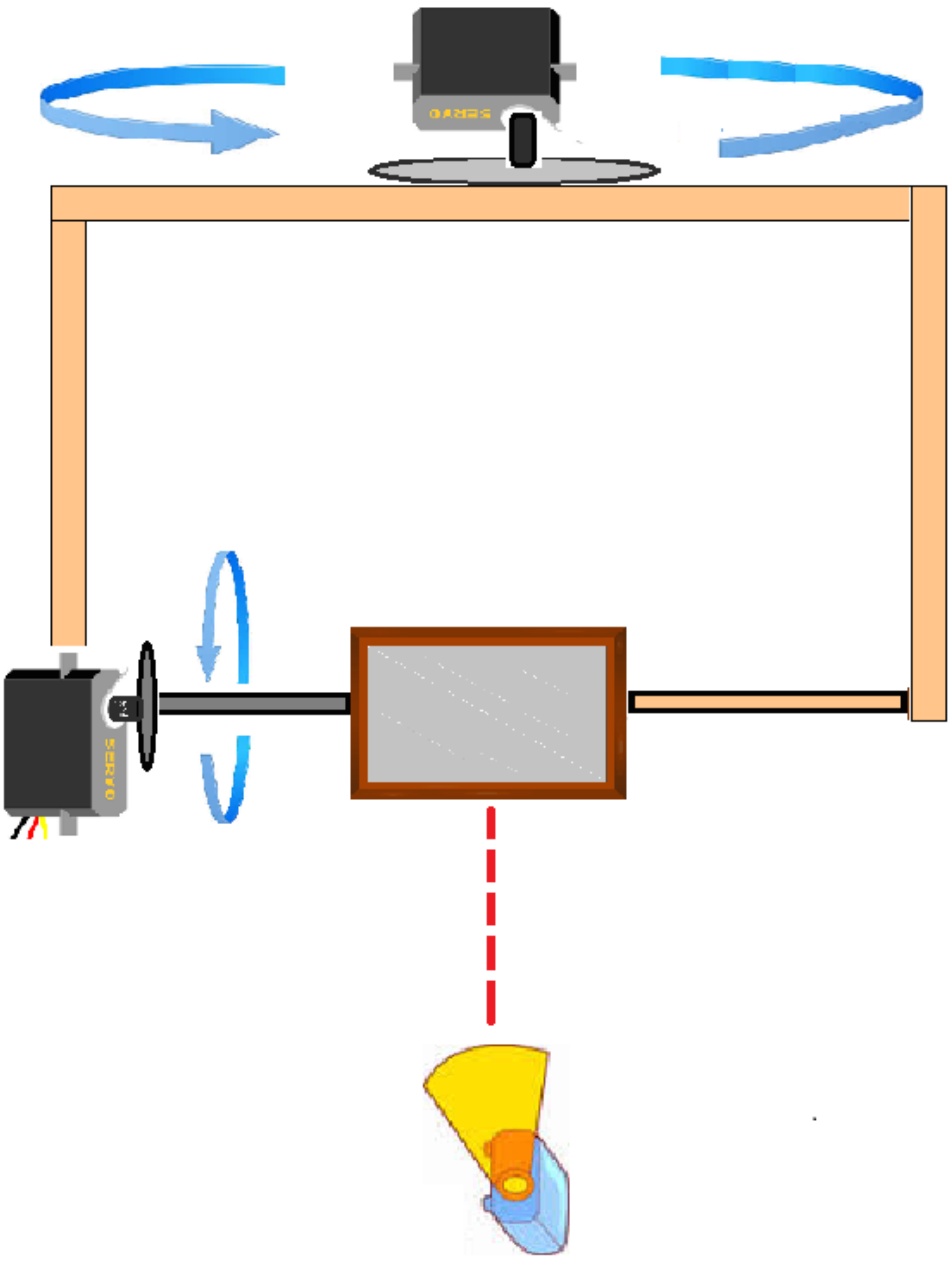

Figure 12: Diagram of Rotation Harness Design

Communication Network

To bridge communication between the user interface and the motor control, two XBee wireless radios were implemented. As shown by the schematic below in figure 13; the virtual user interacts with the Unity3D interface to select their desired point of projection. The angle measurements needed to rotate the mirror to project to this new desired location is calculated by the Unity3D program (using the current project position as the reference location).The angle calculations are transmitted from the Unity3D interface to an XBee radio that is connected to an Arduino Uno board connected to the computer running the user interface program. This XBee radio communicates wirelessly to another XBee radio which is connected to the Arduino Uno board that controls the two motors for the mirror projection. Once the data is sent between the two XBees, the Arduino parse the data and sends the corresponding commands to the motors A and B, which then move the entire structure that holds the mirror on the vertical and horizontal axis. The information about the angles is then sent back to the user interface to let the virtual user know the position of the projection.

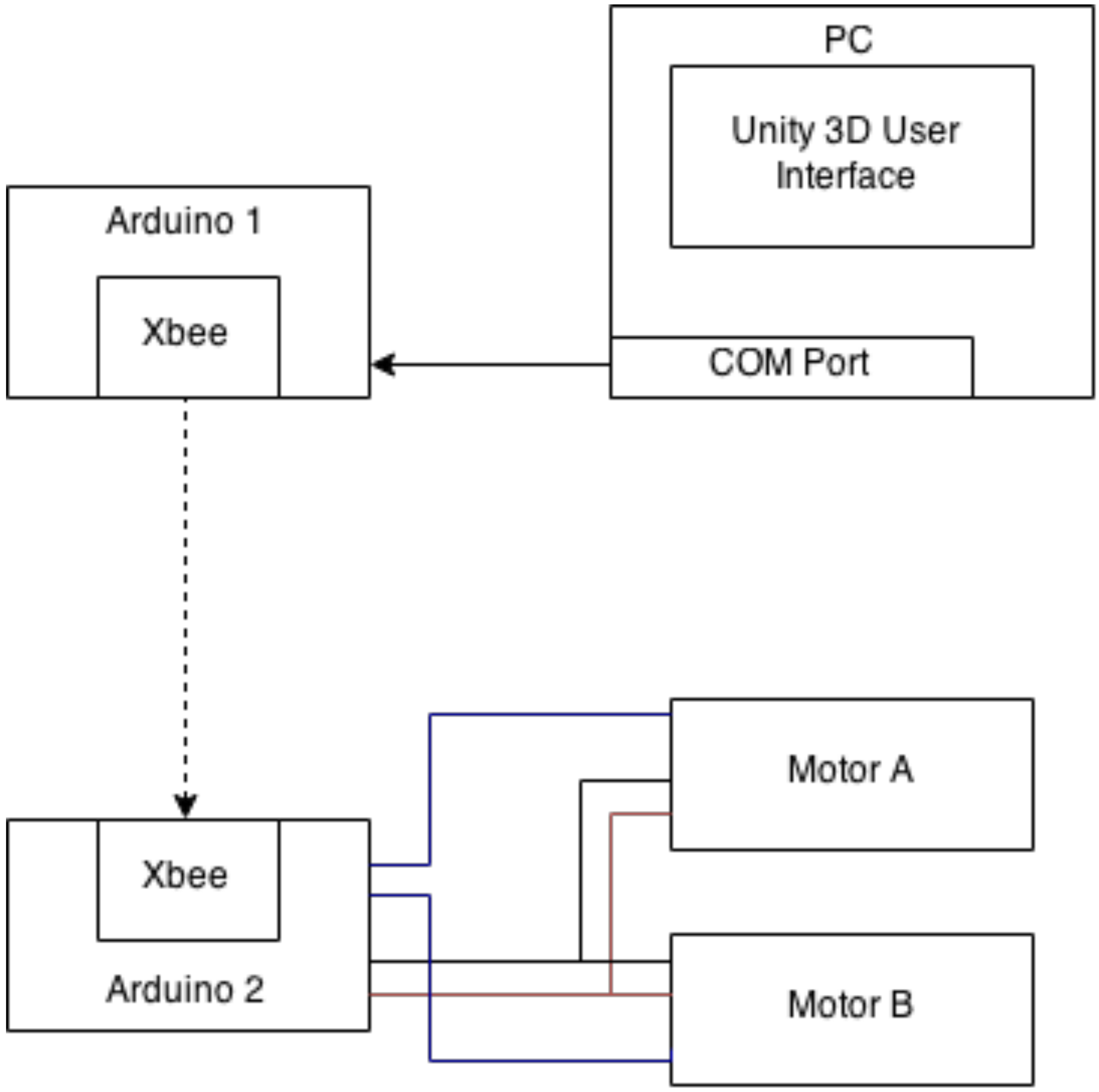

Figure 13: Schematic of Communication Network

Calibration of System

During the initial stage of prototyping, we read on many online sources that a value of 90 would have to be written to the servo motors to make them stop rotating.
However, we found that this value varied from motor to motor. We tested several values, and found that values 100 and 91 were ideal for the two motors.

While testing the initial prototype, we also observed that the servo motors did not always rotate by an equal angle, unlike the stepper motor which always moved by a fixed angle. The speed of the servo motors depends on various factors, like the supplied power voltage, supported weight etc. We also saw that the clockwise and counterclockwise speeds of rotation were different for the servo motors. So if it moves 360 degree over 40 ms, and we split the 40 ms into 4 10s, the first step may be equal to 90 but the second may not

This variation in angles of rotation led to inaccurate results. To get the projected image within an acceptable range of the point selected on the user interface, the angles of rotation were calibrated based on the results of several test attempts.
During our testing, we found that there was more variation while rotating by larger angles than by smaller angles. We also saw that when rotating by angles smaller than 10 degrees, the servo motors were sometimes not able to overcome the resistance presented by the rotation harness.

Since our aim was to project onto two walls, we decided to limit the range of rotation to 80 degrees (+40 to -40) both horizontally and vertically, varying in steps of 10 degrees.
We tested our system quite intensely and calibrated the time it took a servo motor to turn 10, 20, 30 and 40 degrees in the clockwise and counterclockwise directions. We calibrated both the motors separately. The calibrated time values were then directly used in the program.

We also modified the user interface to calculate the angles of rotation in multiples of 10.
The accuracy with the calibrated values was still not as precise as it was with stepper motors, but we were now able to project within a small range of the desired point on a wall.

Limitations due to design

We faced some limitations as a result of the rotation harness design.

First, we needed to attach Motor A to a firm surface parallel to the floor. We chose to use a cardboard box as a casing as it was a flexible way to build and test prototypes.

However, due to this, our range of projection was limited. We could only project on two walls. The cardboard casing was not very stable, and made it difficult for us to rotate the mirror too fast.

The attachment of the rotation harness to motor A was also a cause of concern. The stepper motors were not powerful enough to rotate the harness. The wires connected to motor B offered resistance to the rotation of the harness. As a result we used servo motors instead as they were much more powerful.

The servo motor, in turn, had to be calibrated to increase the accuracy of the projection. Unlike the stepper motors that turns in steps of equal and fixed angles, the servo motors only rotate at a fixed speed. As a result, we had to calibrate a set of fixed angles that the servo motors could turn with decent accuracy. This impacted the user interface, as it had to be modified to send angles of rotation in multiples of 10.

The distance of the system from the walls also impacted the scale and scope of the projection. As the system got further from the wall, even smaller angles of projection covered wide distances. As a result, we tested our prototype by keeping it closer to the wall.

Balancing the load on the rotation harness was a challenging task. The balancing had to be done in way that did not significantly increase the weight of the harness. The balancing also had to take care of aligning the center of the mirror with the center of the light source. We opted to add a lightweight load balancing arm that would help bear some of the weight of the mirror.

Due to the weight restrictions, the rotation harness was made of lightweight materials, instead of metal.

Product Features

Arduino Uno Microcontroller Board

- Processor: ATmega328
- Operating Voltage/Input Voltage: 5V/7-12V
- CPU Speed: 16MHz
- Analog in/out: 6/0
- Digital IO/PWM: 14/6
- EEPROM [KB]: 1
- SRAM [KB]: 2
- FLASH [KB]: 32
- USB: Regular
- UART: 1

XBee Series 2 Modules

- Indoor/Urban range: up to 133 ft. (40m)
- Outdoor RF line-of-sight range: up to 400 ft (120m)
- Transmit power output: 2 mW (+3dbm)
- RF Data Rate: 250 Kbps
- Receiver Sensitivity: -98dbm (1% PER)
- Supply Voltage: 2.8 - 3.6 V
- Transmit Current (typical) 40 mA (@ 3.3V)
- Idle/Receive Current (typical): 40 mA (@3.3 V)
- Power-down Current (1 uA)
- Frequency: ISM 2.4 GHz
- Dimensions: 0.0960” x 1.087”
- Operating Temperature: -40 to 85 C
- Antenna Options: PCB, Integrated Whip, U.FL, RPSMA
- Network Topologies: point to point, Star, Mesh
- Number of Channels: 16 Direct Sequence Channels
- Filtration Options: PAN ID, Channel and Source/Destination

Micro Servo Motors
- Voltage: 4.8-6.0 Volts
- Torque: 16.6/20.8 oz-in. (4.8/6.0V)
- Speed: 0.15/0.10 sec/60° (4.8/6.0V)
- Rotation: ~160°
- Single Top Ball Bearing
- Nylon Gears
- 3-Pole Ferrite Motor
- 31.8 x 11.7 x 29mm
- Wire Length: 160mm
- Weight: 9g

Stepper Motors
- Step Angle: 7.50 deg
- Frame Size: 35 mm
- Max Holding Torque: 20 mNm
- Steps per rotation: 48
- Drive Circuit: Bipolar
- Drive Voltage: 24 V
- Coil Resistance per Phase: 7 Ohms
- Drive IC: UDN2916B-V
- Magnet Material: Nd-Fe-B bounded magnet/Ferrite plasic magnet/Polar anisotriphy ferrite sintered magnet
- Insulation Resistance: 100
- Dielectric Strength: 500
- Class of Insulation: E
- Operating Temperature: -10 ~ +50
- Storage Temperature: -30 ~ +80
- Operating Humidity: 20 ~ 90 RH
- Standard: No

Unity 3D
- Version: Unity 4.6.3
- OS: Windows XP SP2+, 7 SP1+, 8; Mac OS X 10.8+.
- GPU: Graphics card with DX9 (shader model 2.0) capabilities.
- Anything made since 2004 should work.

Final Design

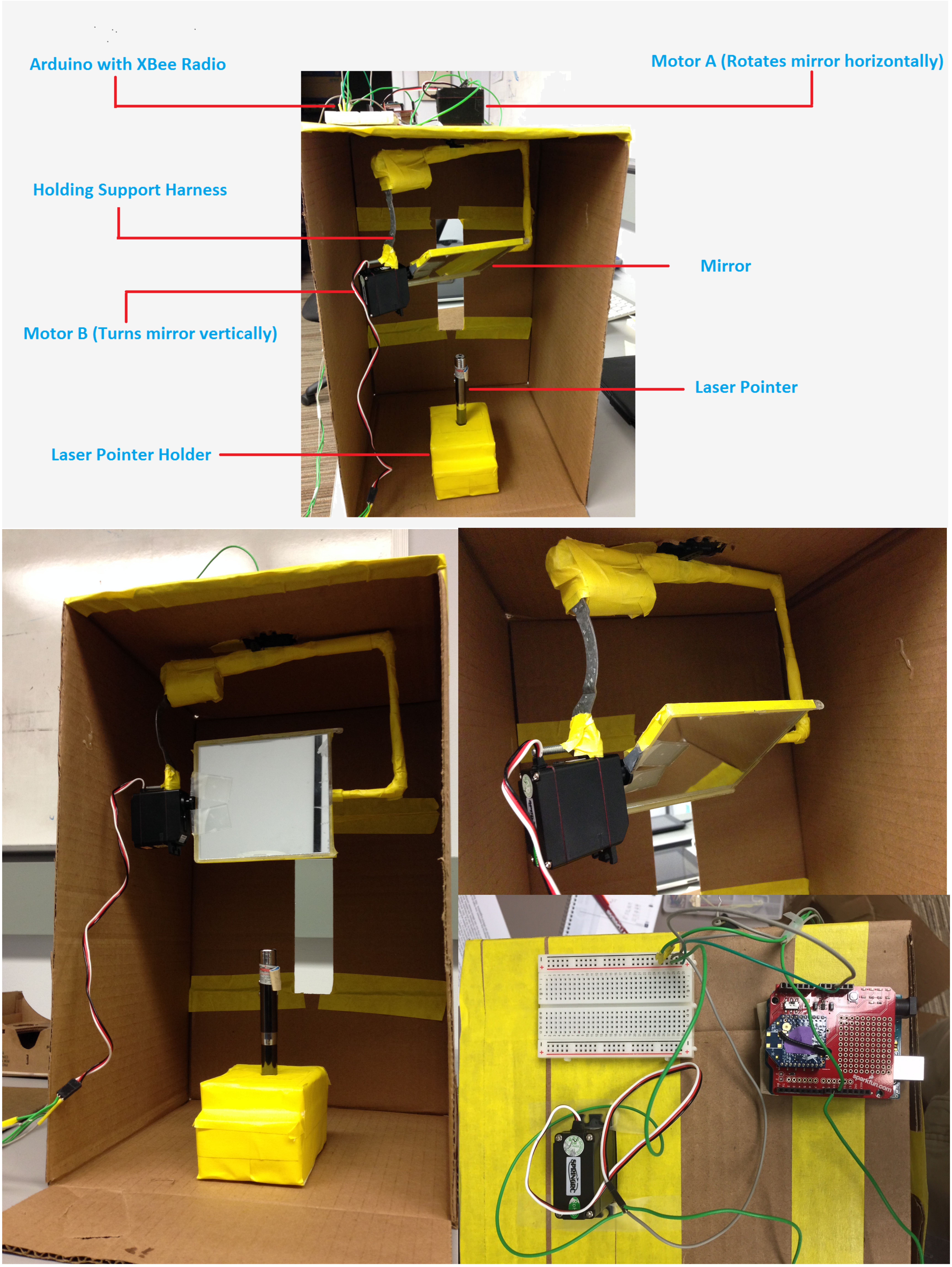

Figure 15: Final Prototype Design

Evaluation
After the final prototype was built, it underwent an evalution. To evaluate success for our system there were many factors that went into play. The biggest aspect that we wanted to focus on was how accurate was the projection based on the point that was clicked within Unity 3D and how easily people could interact with the interface. We selected a small group of participants to test out the system. We had them click on the interface to see where they could project images. After they ran through that simple tasks, we asked them for feedback on what they did and what they did not like about the system. The majority of the users were enthusiastic about the novelty of the system and found the user interface to be fairly easy to use. They felt that the simple clicks was nice and appreciated the fact that they could see the position of the projection before they moved to a different point. They mentioned that they had hoped for a more accurate system. Although the projection itself was fairly accurate, it was not as precise as they had imagined. Some users also mentioned some safety issues they would want to be addressed. As the system is designed now, there are some exposed wires that the users would like to see covered up not only because of safety but also for aesthetic purposes. Overall, the users were please with the system and are eager to see future developments.

Conclusion

Learning Outcomes

This project presented several learning opportunities. It presented us with the true reality of research and design. Designing or developing any new product requires dedication to overcome obstacles and research solutions to difficult problems in order to achieve the final goal. Throughout our design process, we faced several shortcomings. We had to learn how to first prioritize our tasks to solve and then manage our process in terms of milestones. Initially we were overwhelmed from focusing a lot on too many aspects and the finer details of the overall project instead of getting the key functionality to work. Applying the engineering design process, we started with basic prototyping and added features and functionalities as we went along. The biggest learning in the whole project was to keep divide the prototype into sub modules. This way, revisions done to one module did not require significant rework on the others, and also made it easier for us to test individual modules and then integrate them. … We also learned how to work within a multidisciplinary team. There were different backgrounds within the group and everyone had different strengths. This allowed for a division of workload and also an acknowledgement of different viewpoints.

If we could do it all over again

If given the chance to start over from scratch, we would do a few things differently. First, we would select more powerful and stable stepper motors, preferable those that came with a potentiometer to adjust the speed and angle of rotation. We would also have looked into motor jitter earlier. We would use wires that were more flexible so as to present minimum resistance to the rotation harness, instead of learning it the hard way.
We would also use a slightly bigger, and lighter, mirror. We would also have looked into 3D printing to make a model of the rotation harness.

Further Work

In the future we would take larger steps to complete this system. The first major step that we would like to take would be to allow the virtual user to project the image or light source anywhere in the room and not just on two walls in the corner of a room. After that is completed we would like to generalize the system to be able to use and actual projector so that more that just a laser pointed dot could be projected throughout the whole room. Using the projector would allow for a variety of images and messages to be projected around the room. Because of this, our next step would be to create another subsystem that would allow the real world user to interact with that projection and in turn interact with the virtual user as well.

References

Mirror Worlds. from http://www.icat.vt.edu/impact/project/mirror-worlds

In memory of Dr. Mark Weiser. (2008). from http://www.parc.com/services/focus-area/ubicomp/

Hansmann, U., Merk, L., Nicklous, M. S., & Stober, T. (2003). *Pervasive Computing: The Mobiel World*. New York: Springer-Verlag Berlin Heidelberg.

Pinhanez, C. S. (2001). *The Everywhere Displays Projector: A Device to Create Ubiquitous Graphical Interfaces*. Paper presented at the UbiComp '01 Proceedings of the 3rd international conference on Ubiquitous Computing, London, UK.

Weiser, M. (1991). The Computer for the 21st Century. *Scientific American, 265*(3), 94-104. doi: 10.1038/scientificamerican0991-94

Weiser, M., Gold, R., & Brown, J. S. (1999). The origins of ubiquitous computing research at PARC in the late 1980s. *IBM Systems Journal, 38*(4), 693-696.

Winter, J. (2008). Emerging Policy Problems Related to Ubiquitous Computing: Negotiating Stakeholders' Vision of the Future. *Knowledge and Policy, 21*(4), 191-203.