Project Projective

Authors: Olivia Fernau, Isabel Gardner, Rama Hassabelnabi
Affiliation: Electrical and Computer Engineering, Carnegie Mellon University

***Abstract*—A system capable of allowing workers more mobility and freedom while still being able to complete tasks requiring heavy focus. By using computer vision and a motorized projector, users can project their work to any location in the room. A calibration process ensures only intentional movement is detected so the user is free to move, stretch, and feel better during their workday.**

***Index Terms*—Arduino, Calibration, Camera, Computer Vision, Facial Detection, Head Pose, Lidar, Mediapipe, Motor, Projection,**

1 INTRODUCTION

Virtual work has become the new normal with the recent pandemic. People now spend eight plus hours a day at a computer screen. This static method of working has increased eye strain, headaches, neck pain and has created an overall dissatisfaction with working days.[1] We aim to improve mobility for these workers while still allowing them to complete their tasks in a focused, productive manner. Project Projective introduces a system that enables a projection to follow the gaze of a user. This way, the user can move their neck and body around while allowing their screen to seamlessly stay within their gaze in real-time.

Augmented reality headsets perform a similar function as they also display information into space that moves with a user's body. However, these headsets require a user to wear a bulky piece of equipment on their head and cost about $1000-$3500.[2] Our system utilizes computer vision to track the head movement of a user. The movement of the head is then sent to a motor that is attached to a projecter. This way, the projection can move with the head movements of the user. The system also must undergo a calibration process to ensure the projection moves with the user's line of sight in a comfortable, enjoyable manner.

2 USE-CASE REQUIREMENTS

1. The latency for the full pipeline, from detecting the user's movement to the initial movement of the projection, should be 200 ms.
2. The projection should follow natural head movement when one is sitting. From a centered position, we do not expect a user to comfortably rotate their head beyond a 45 degree angle in the left, right, up and down directions. Hence, the projection movement should follow a head rotation up to 45 degrees in the aforementioned directions.
3. The user can comfortably move their body from their initial, calibrated position without the system failing. Our system must be able to handle a 1 meter translation from the initial, calibration position in the x,y, and z directions.
4. The projection should match the user's line of sight 95% of the time (measured within the bounds of use-case requirements 2 and 3).
5. The projection will be stable and will not vibrate during or after adjustment.
6. The projector should be placed between 6 to 10 meters of the wall for ideal use.
7. The user should be within 2 meters on the horizontal axis of the camera.
8. The user experience should have a satisfaction rate of 90% or above.

3 ARCHITECTURE AND/OR PRINCIPLE OF OPERATION

A block diagram visualizing our system is shown in Fig. 1. Our system will consist of three major subsystems: Computer Vision, System Translation, and Mechanics. The first subsystem, which takes in the user's input, is the Computer Vision system. A camera specialized for low lighting will capture images of the user's face. From these images, our CV program will extrapolate facial landmarks and use these to detect the user's head angle in space. The second subsystem in this process is System Translation, which will translate the head angle coordinates into those usable by an arduino program. Then, using these coordinates along with stored user information from the initial calibration process, the translation code will provide motor commands for our mechanical projector arm. This mechanical projector arm is the main product of our third subsystem, Mechanics. This subsystem will include a mounted projector on a motorized tripod, which will turn to match the projector's projection with the user's view, using the arduino commands.

3.1 Computer Vision

This system will take input from the camera and convert this input into head pose data as head pose data directly infers the gaze estimation of the user. First, the camera images are processed by the OpenCV library into data that is usable by our program. Then, we will use MediaPipe to

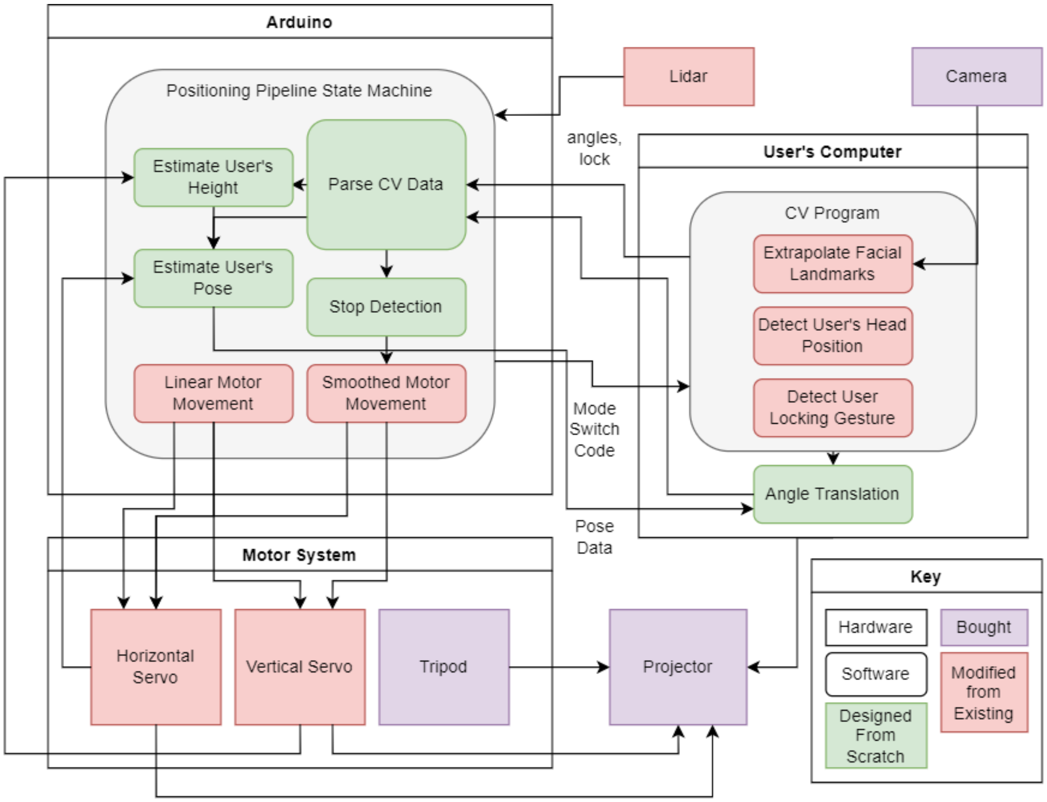


Figure 1: Block Diagram

detect facial landmarks in the processed images. By honing in on the location of the facial landmarks, we can convert this data into a reasonable guess of the head pose of the user. Then, we extrapolate the head rotation amount as an angle which will be sent through our translation module to the arduino program. All computer vision software will be written in Python.

3.2 Integration Modules

This system is focused primarily on two modules, Calibration and Translation, which will both be used to translate the CV head angle outputs into arduino motor commands. The Calibration module will be a program that is run before the user begins a session of using the product. By recording the user's preferred place of the projector's projection for three head angles, our program should calculate the user's position in space, particularly their head's position in relation to the projector. This is to match the user's eye gaze as much as possible. After this initial calibration, the user's field of view information will be stored and used by the program in the Translation module. For translating the head angles into the arduino program, we will be using Python's pySerial package. Our program then needs to utilize our stored user information to calculate how the arduino should command motor movement from our mechanical projector arm.

3.3 Mechanics

Our mechanical system will take the motor commands from the integration modules and use this to move the actual projection in line with the user's face. The projector itself will be mounted on a motor, which will be the endpoint where the mechanics system interacts with the integration system. The motor will be mounted in order to turn up, down, left, or right, so that the projector can follow the user's head movements focusing on one wall.

4 DESIGN REQUIREMENTS

Each design requirement number corresponds to the matching use-case requirement number it fulfills.

> *Requirement 1a:* The system must respond to a head movement within 200 ms. We believe a 200ms latency is within reasonable bounds of following the user's head in a way that feels instantaneous while still giving our program a bandwidth of time to process the movement.
>
> *Requirement 1b:* The user's head rotation is detected and calculated in real-time (less than 30 ms). The program we are using to measure facial landmarks offers real-time detection which allows this specification to be possible. This allocates 170 ms for the motor to receive the command to begin movement.

Requirement 2: From a centered head position that faces the camera, the user's head rotation can be detected up to 45 degrees in each direction for the yaw and pitch degrees of freedom. This implies sufficient facial landmarks must be used that are visible from the 45 degree rotation. Beyond this angle, it will be difficult for the camera to identify the needed facial landmarks.

Requirement 3: An initial calibration is necessary to understand where the user is located in space relative to the camera, projection, and wall. Moving far from this initial, calibrated location has the potential to mess up the projection placement without undergoing a new calibration process. To prevent the user from calibrating their position constantly and improve their mobility, our system must be able to handle a person moving 1 meter in the x, y, or z directions from their initial, calibrated location.

Requirement 4: The calibration system must be accurate enough such that the center of the projector is no more than 0.125 * (projector width) away from the user's central focus. A user's eyes can adjust to the projector being slightly off, and it is natural for a user to not have their focus directly on the center of the screen. Additionally, small head movements should not be picked up by the CV program, so the user can slightly adjust their gaze to be more centered if they wish. However, more than 1/8th off of center may cause the user discomfort, or cause the user to trigger the mechanism again while adjusting their head to where they want to focus on the projected screen.

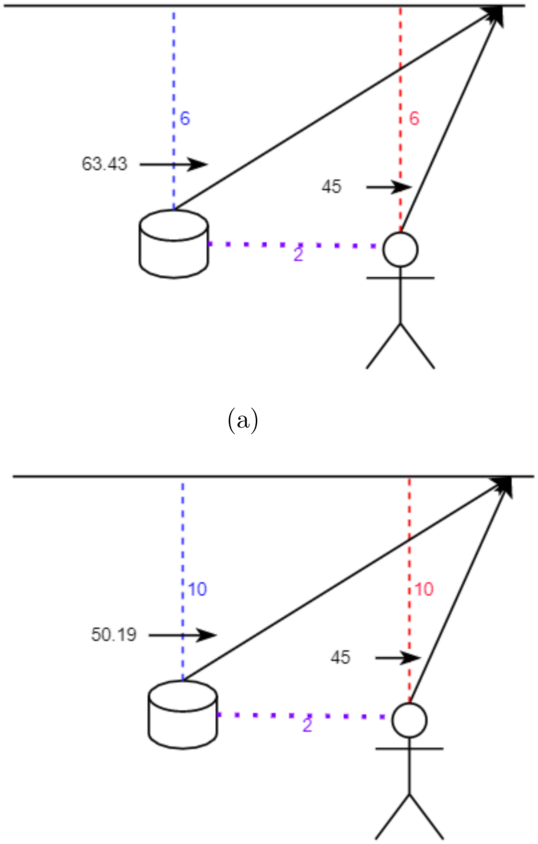
Requirement 5: Small head movements, which would result in less than 0.25 meters of change for the projection center, must be ignored by the system. This is to prevent the projection from moving in an annoying, unnecessary way.

Requirement 6a: The projector must be within 10 meters from a wall. The average living room size is 16x20 meters, and a living room is most likely the largest space a user might use our product. It's likely the projector would be placed in the middle of the room or closer to the wall, so that the user could stand behind the glare to avoid blocking the projection. Thus by halving the longest side, we get 10 meters as approximately the longest distance the projector may have to be from the wall.

Requirement 6b: It follows from requirement 6a that the lidar range must be at least 10 meters to account for this projector distance.

Requirement 7: The distance between the camera and the user should be below 2 meters on the horizontal axis. The reasoning behind this has to do with projector skew. If a projector is projected to a wall between 6-10 meters away, we do not want this projection to be at an angle larger than 65 degrees. This is because at a large tilt, the screen may be skewed, particularly if it is against a flat wall. The skew with the user 2 meters away is shown below, first at a 6 meter distance and then at a 10 meter distance. We can see that the skew at 6 meters is 63.43 degrees, which is just under our maximum of 65.

Requirement 8: When our system is in between prototypes, we will gather roughly 5 users to test the system and gather their feedback. Our design specs may fluctuate from this feedback with the intention of creating the most comfortable, enjoyable, and helpful experience for the user.



(a)

(b)

Figure 2: Maximum Projector Skew Angle (a) 6 meters (b) 10 meters

5 DESIGN TRADE STUDIES

5.1 Facial Detection Algorithm

The two main goals for facial landmark detection are high accuracy and real-time computation. Two commonly used facial landmark detection libraries are known as Dlib and Mediapipe. Dlib and Mediapipe both have deep learning based facial detection capabilities. Although both libraries meet the aforementioned goals, Mediapipe is known for it's "super real-time" computation speed and tends to have a higher accuracy, making it a better choice than Dlib for our purposes.[3] Another strong factor for choosing Mediapipe over Dlib is that it can detect 3D facial landmark coordinates whereas Dlib only detects 2D coordinates. Hence, if Dlib was chosen, more computation and time would need to be allocated for the head pose estimation.[4]

5.2 Head Pose Estimation

When testing the head pose estimation module, we encountered a tradeoff between robustness and accuracy. Initially, the yaw and pitch were solely dependent on the current frame being processed. After testing, we realized the yaw and pitch should be averaged with their 5 previous values because of how quickly the yaw and pitch can change. This calculation decreased the immediate accuracy of the yaw and pitch but made the program more robust to error.

5.3 Camera

Mediapipe is known to work exceptionally well even on mobile device cameras, allowing the camera choice in our system to be quite flexible.[2] However, since a projector is generally used in dim lighting, our camera must be able to clearly process images in low light. We originally thought a camera on a Macbook Pro would be sufficient to conduct the facial detection process. To test this out, we implemented Mediapape facial detection on a Macbook Pro and tested out the facial landmark detection in bright and dim settings. In bright settings, facial landmarks are detected in real-time with high accuracy as a person rotates their head. However, in dark settings, the facial landmark detection breaks down as a person rotates their head. As a result, the option for using a Macbook Pro camera was thrown out. We settled on the Arducam 1080P Day and Night Vision USB Camera that is compatible with Mac OS, inexpensive, and easy to mount. These specs are more than sufficient for Mediapipe and allow for facial landmark detection in dark settings.

5.4 Projector

Considering our design requirements, our projector needed to be lightweight, easily mounted, affordable, and reasonably functional. We considered three projectors as possibilities, the "Funtustic-Ci Projector", the "ELLEPHAS Mini-Projector", and the "PVO Projector".

We first considered the Funtustic-Ci projector. This projector has a weight of 3.97 lbs, which while not as light as the PVO projector, is still fairly lightweight. The display resolution is 1920x1080, which is better compared with the other two. However, in all depictions this projector was on a table rather than on anything mountable, which led us to believe it's possible this product may not have an underside that can be mountable. We considered making some special mounting modifications to accommodate this projector, if it ended up being our best option. However it is also the most expensive out of the three, with a price of $140. With these negatives, other options needed to be considered.

The ELLEPHAS projector has a weight of 5.68 lbs, and although it's still classified as a 'mini-projector', it is the heaviest out of the three. On the other hand, the display resolution is 1280x720, which is pretty sharp and better than the PVO projector. Another positive point is it comes with an attached strap and screw on the bottom, which makes it the most easily mountable out of the three options. However in terms of price, it's not much cheaper than the Funtustic-Ci, at a price of $130.99. Aside from the mountability, this option would not be ideal.

The PVO projector has a weight of 0.7 lbs, which already makes it highly preferable to the other two since it has the lowest weight by far. The most important consideration for the projector tradeoff is weight, because to meet our main requirement to following a user's gaze, we need the projector to be as lightweight as possible so that the motor's are not slowed by it. However the display resolution is 800x480, which is a worse resolution than the other two. On the other hand, it does meet our baseline requirement of easy attachment, with a screw hole on the bottom. Finally, it is also the most affordable of the three, with only a price of $58.99. Since this projector is ideal in terms of weight and budget, and according to reviews, has reasonable resolution for functionality, we decided to choose this projector for our product.

5.5 Lidar

The lidar device is an integral part of our calibration process, as it will measure the distance between the projector apparatus and the wall. We needed a lidar that could fit our needs to satisfy our requirement that our product could be used in many different spaces. We considered two lidar devices, the LIDAR-Lite version 3 (Lite v3) and the Garmin LIDAR-Lite version 4 (Lite v4).

First we should consider the range and accuracy requirement. The lidar device we choose must have a range up to 10 meters, so to satisfy design requirement 6b. Both devices satisfy this requirement, but notably the Lite v4 fluctuates more in it's error as it gets further away from an object. The Lite v4 has a +/-2cm error at a range of 5 meters, and then a +/-5cm error at 10 meters. In comparison, the Lite v3 has a constant approximate error of no more than +/-2.5cm.

However, even though the Lite v3 is better in accuracy, the Lite v4 is significantly better in both cost and weight properties. Weight is an important consideration because we want our device to be reasonably portable to meet our multi-space requirement, and not hinder the projector's speed as it needs to follow the user's head in as close to real-time as possible. Due to the many parts we need for this project, it's important that the cost is as low as possible as well.

There has been an update to this section since the design report, we ended up getting a different LiDAR device because the Lite v4 actually ended up being bought out of stock. In review, this worked in our favor, because after doing more searches on SparkFun, I ended up finding an option that was better in terms of weight and price. The TFMini-S weighs only about 5g and costs less than the other options. The only difficulty with this device is that it is slightly less accurate, and involves more building and soldering to connect with the Arduino. However, with it's other benefits, and the time we'd have to wait for the Lite v4 to get back in stock, we decided this was the best option by far. In our document's tables section, near the end, is a comparison between the two devices' cost and weight, in Table 1 where the difference is easily highlighted.

5.6 Motor and Mount Selection

The purpose of the motors is to rotate the projector such that it mimics the way a person moves their head. Thus, there are four considerations that we need to keep in mind.

- The projector must rotate accurately
- We require two degrees of freedom
- Can hold load of 5 lb
- Power consumption

Thus, we aimed for two servos mounted in the form of a joint. The power required is important here because some servos are known to be power drainers. Since we are using two motors simultaneously we want to limit power consumption.

Our first option was the "LD 20MG Digital Servo." This motor requires 6.6V of power, can rotate 180 degrees, and moves at a speed of 0.2s/60 degrees when powered with 7.4V. We will likely need to operate the motor at the maximum speed or nearly the maximum speed, so we expect to need at least 7 V at some times. It takes 200ms to rotate 60 degrees, so we need to keep in mind that it may need more time to complete the CV to motor pipeline for larger rotations of the head. To mount the motors, we can purchase a pan and tilt motor bracket made of aluminum. Extra parts are needed to secure the bottom motor so that the projector can horizontally rotate without shaking.

Our second option was the "UCTronics Preassembled Pan and Tilt Servo Kit." This is commonly used for robotic arms and the system consists of two DS 3115MG servo motors in a pan and tilt bracket attached to a base. It allows for 180 degree rotation horizontally and 90 degrees vertically. It is made of aluminum and requires 4.8V of power. It moves at a speed of 0.15s/60 degrees when powered with 7.2V. This system would also need at least 7 V at some times. These motors are faster than the LD 20MG, and we feel that the difference in milliseconds is significant since we are aiming for real time response. Also, this is preassembled so no extra time or parts are needed to get basic motor tests done.

Both of these options use servo motors which offer better accuracy when rotating. Both use a pan and tilt system to achieve two degrees of freedom. Both motor brackets are made of aluminum, so it is able to carry the weight of a small projector. The UCTronics system is slightly faster, and uses slightly less power. The UCTronics system also saves us time assembling the motor and mounting them, and waiting on parts to make it fit our needs. Thus, we chose the UCTronics system.

5.7 Battery Selection

The servo motors typically require 6V and at most 7-7.2V. Since we want the system to be as portable as possible, we want a long lasting battery that supplies a safe amount of current. We considered the "Talentcell 2000 Cycles 6V 6Ah Rechargeable LiFePO4 Battery," and 4 AA Energizer batteries.

First we calculate the energy needed for the motor system. The current going through the motor when operating at 7.2V is 1.5A. The system includes two motors connected in parallel, so the total current drawn would be 3A. For the LiFePO4 battery with a 6Ah capacity, a single charge would last 6 / 3 h = 2 h. This is a suitable battery life because the user will be able to have a focus time of 2 hours, and needing to plug this in can serve as a break time. The LiFePO4 battery can safely be charged at a rate of 1C which equates to 6A. Thus, it would take 1 hour to charge.

We then consider the 4 AA Energizer batteries. With a 2800mAh capacity, a single charge would last 2.8 /3 h = 0.933 h which is about 56 min. This is also an acceptable battery life, but since they are not rechargeable, the batteries need to be replaced fairly often.

The rechargeable battery is known to output too much current and people have had issues when powering servo motors. Regular AA batteries do need to be switched out often but ensure that the servo motors will not get ruined so we decided to use 4 AA Energizer batteries to make up 6V of power.

6 SYSTEM IMPLEMENTATION

6.1 Computer Vision

We will be using Python to implement the facial landmark detection and head pose estimation modules. The library OpenCV will be used to process the images of the user from the camera. These images will then be processed with the library MediaPipe to detect facial landmarks. MediaPipe detects 468 3D facial landmarks in real-time.[2] However, not all landmarks are necessary to detect head rotation. We intend to test various combinations of landmarks but will likely be using, at the very least, the tip of the nose, the corners of the mouth and eyes, and the bottom of the chin.

Based on the facial landmarks and the intrinsic parameters of the camera (optical center, focal length, radial distortion), the head pose can be calculated. The head pose, which is fully characterized by a rotation matrix and a translation vector, will be calculated using the Perspective-n-Point function in OpenCV. This will output the pose

$$p = [w_x, w_y, w_z, t_x, t_y, t_z] \tag{1}$$

The rotation parameters w_x, w_y, w_z correspond to the Euler angles pitch, yaw, and roll depicted in Fig. 3. The translation of the head is represented by t_x, t_y, t_z. This data will then sent to the translation module to be converted into motor rotation angles to ensure the projection is placed in the user's line of sight.

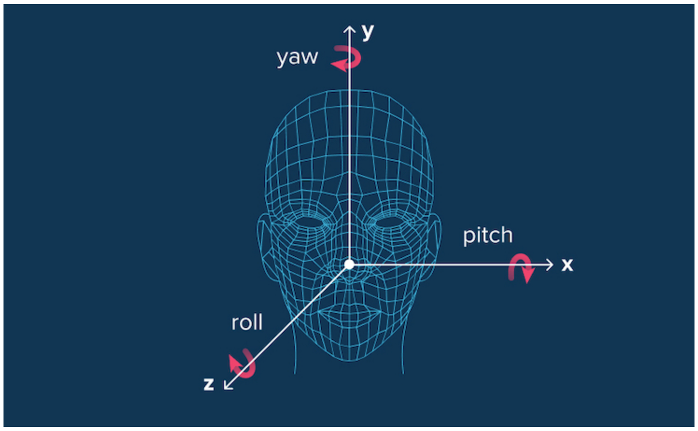


Figure 3: Euler Angles

6.2 Integration System

6.2.1 Calibration

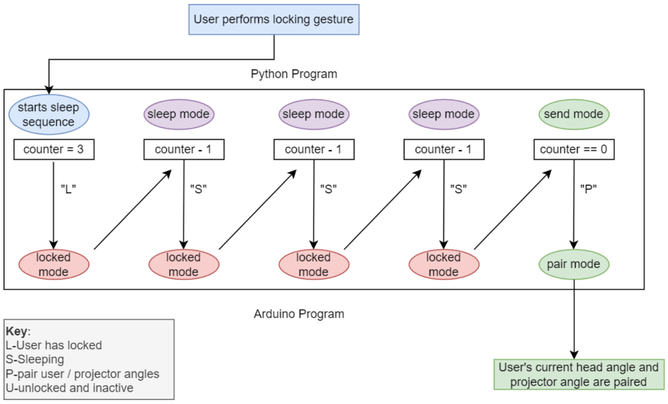


Figure 4: Locking and Pairing Process Serial Communication

Beginning use of our product, the user will first have to perform some manual calibration with three poses. When the user is satisfied with the projector position in each of the three stages, they will blink three times to bind the projector pose to the head angle. This initiates a locking and pairing process, depicted above in Fig. 4. First, there is a locking phase, where the signal is sent to the arduino that the motor should stop moving. When the locking phase begins, a ringing sound plays on the user's computer so they know their lock has been effective. There is then a 'sleeping' phase, where the CV program is performing two tasks. First, there is an intentional delay during this phase to give the user time to get their head into the position they want to be taken by the program. Second, the program is flushing the older head angle data in order to get the most accurate and up-to-date head angle as it's recorded pairing data. This is necessary because we average 4 of the user's past detected angles with their current detected angle, which makes our program more robust to small errors, but also means the angles that are sent to the arduino may be affected by past angles, which could lead to some inaccuracy if there has been a large change in user position. When the sleeping process is over, then the user's head angle data and the projector's angle data are taken and recorded in the pairing stage.

Algorithm 1 Calibration State Machine

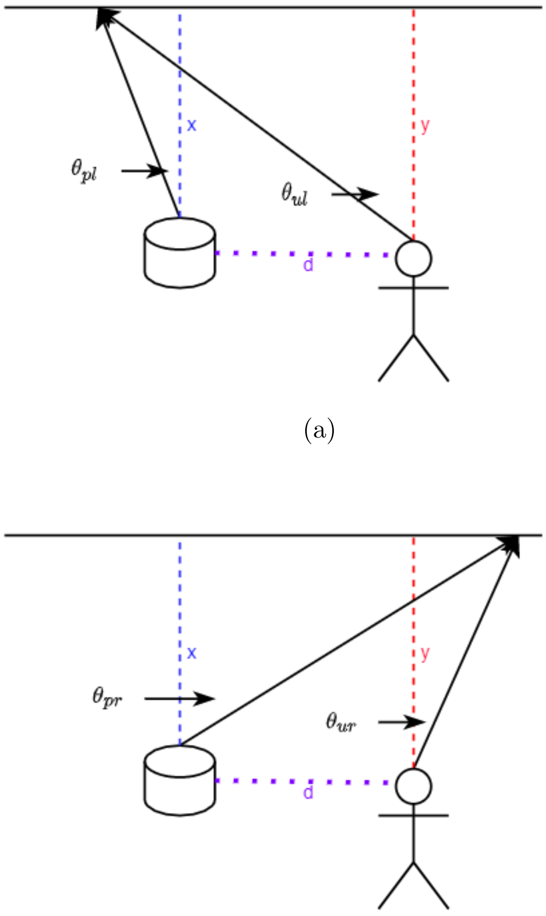
```
while User is Calibrating do
    if Calibration Phase == 1 then
        if User has not locked then
            Allow user to move projector around with their head
        else
            Initialize locking and pairing process
            Save user's vertical offset from the projector
        end if
    else if Calibration Phase == 2 then
        if User has not locked then
            Move or keep the projector at 30
        else
            Initialize locking and pairing process
            Save the first user/projector angle pair
        end if
    else if Calibration Phase == 3 then
        if User has not locked then
            Move or keep the projector at -30
        else
            Initialize locking and pairing process
            Save the second user/projector angle pair
            Use the angle pairs to calculate user position
            Flag user is no longer calibrating
            Break out of the loop
        end if
    end if
end while
```

The three phases of calibration are implemented as a state machine in our Arduino program, depicted as Algorithm 1. In the first phase, the user will have to center the projector on their field of vision while looking forward. Since our CV program will already be active, the user can use their head movements to move the projector around in two dimensions, until it is lined up with their central pose. Using this configuration, the height of the user will be calculated and stored as a float.

The projector will then turn 30 degrees to the right. Again, the user should blink 3 times when they are ready to bind their head angle to the projector's angle. After locking the projector, the user will repeat the process of lining up the projector with their field of vision by looking at the screen. The CV program will giving us the angle of the user (θ_{ur}), and the angle of the projector (θ_{pr}) is already known to be 30 degrees. Both of these measurements will also be stored as radian floats in our program as the first angle pair.

Finally, the projector will then turn 30 degrees to the left. Again, the user should blink 3 times when they are ready to bind their head angle to the projector's angle. The user should again look at the screen while the CV program collects their head angle. Our program will then store this new angle of the projector (θ_{pl}) and angle of the user (θul), and perform calculations to detect the user's position.

Both the second and third calibration phases are illustrated by Fig. 5, and the data collected in this phases is used for the final calculations. The diagrams in Fig. 5 are in a 2D plane facing downwards on the system, with the d line representing the 'distance between the user and the projector' and the y line representing 'distance between the user and the wall'. Using the four angles gathered from the left and right calibration measurements, we then get a solvable system of equations for d (distance horizontally between user and projector) and y (distance vertically between user and wall), shown in "(2)" and "(3)". Considering we already know x (distance between projector and wall) and the height of the user from calibration 1, we then have the full set of information to calculate offsets between the user's position in space and the projector's position in space. We perform these calculations, and then send them back to the user's computer over the serial port, at the end of the last calibration phase.



(a)

(b)

Figure 5: Top-Down Calibration. (a) Left side (b) Right side

$$d = y \tan \theta_{ul} - x \tan \theta_{pl} \tag{2}$$

$$d = y \tan \theta_{ur} - x \tan \theta_{pr} \tag{3}$$

6.2.2 Translation

The translation module will be written in both python and arduino code. This is because the python code needs to convert the head angle data to be sent over the serial port. We will do this using python's pySerial package. Before use, we must set the arduino program such that it should be listening on a COM port connected to the pySerial channel for the incoming data. The python program first uses the results from calibration to convert the user's angle into what the projector angle should be to line up with the user's gaze. We can then read the serial string into a float representation of the projector's angle. Then, the projector angle will be converted to motor commands, beginning the mechanical process.

The speed of the motors will be set to 60 degrees per second and will follow a cubic curve. This means the motors move slower at first, then dramatically move faster, and slow down again into the final position. This is to help smooth out the movement of the projection and mimic the way a person's head moves.

6.3 Hardware Connections

Two PWM pins on the Arduino would be used for the two servo motors. The Arduino will be powered through a USB cable connected to the user's laptop (since it will already be connected to send serial data). The servo motors would be connected in parallel and require 6V. The 6V from 4 AA Energizer batteries would power the servo motors. Refer to the diagram below in Fig. 5 – note that orange lines correspond to PWM pins, red lines correspond to power, and black lines correspond to ground.

For the lidar circuit, two PWM pins on the Arduino are also used, for TX and RX. These connections have to be connected to the lidar through a bidirectional logic level converter. The logic level converter is required for reading the sensor values through the serial UART pins, and requires both 5V and 3.3V for the high voltage and low voltage sides, respectively. The lidar device is powered by 5V from the Arduino. For this diagram, red lines correspond to power, and black lines correspond to ground. The green lines correspond to the TX of the lidar sensor values, and the blue lines correspond to the RX.

6.4 Software Communication

Communication between the Arduino and the integration modules will be through a USB cable. The distance from the wall will be collected from the LiDAR sensor and sent from the Arduino to the modules. Then the serial port will close off its connection to the LiDAR data and will send the current position(s) of the user's head in each direction to the Arduino to move the motors accordingly.

7 TEST & VALIDATION

7.1 Results for Design Specification 1

We ran 20 trials and timed from when the user moved to a new position to when the motor received the command to move to that new position. Between all trials, the time for these commands to be transferred through this pipeline ranged from 200 ms to 450 ms. We did meet the design spec that the user's head rotation is calculated in real-time (less than 30 ms). This means it takes anywhere from 170 ms to 420 ms for the motor to receive the command to move. As a result, we only met our design spec about 50% of the time. After the motor received the command, it would then take roughly 100 ms to 500 ms for the motor to move the projection to the new position, dependent on the angle the motor had to move.

Prior to performing the 20 trials, the motor received the commands in less than 200 ms. However, when moving the projector during the translation phase, we noticed if the projector responded immediately to every movement, the movement would be jerky and unpleasant for the user. We implemented a routine in the Arduino code that compared past user angles to detect if the user was stopped, and only moved the motor if the user stopped. Although this increased the lag time of our system, this did improve the user experience significantly, so we decided to prioritize the motor's smooth movement over the lag time.

7.2 Results for Design Specification 2

We ran 20 trials to identify the angles at which head rotation calculations break down for each direction. We met our specification since, in all trials, the angle at which the yaw and pitch calculations broke down were always above 45 degrees. In fact, the yaw was able to be detected up to ± 60 degrees while the pitch was able to be detected up to ± 55 degrees.

7.3 Results for Design Specification 3

During the translation phase, we identified the distance at which the projection placement accuracy breaks down in the x, y, and z directions. We discovered the user can move up to ± 0.5 meters in any direction without a noticeable negative change to the projection placement accuracy. This means we did not meet our design specification as we wanted the user to be able to move at least ± 1 meter in any direction.

This is because we did not take into account the user's change in their position in space within our design. This is mainly because our current camera did not have a wide enough field-of-view to keep track of large changes to their position in space. We also thought it was more important to spend our time testing the overall system on users than focusing on this design spec. Moreover, it is quite easy to move the projector and camera to accommodate the user's change in position in space without having to re-enter the

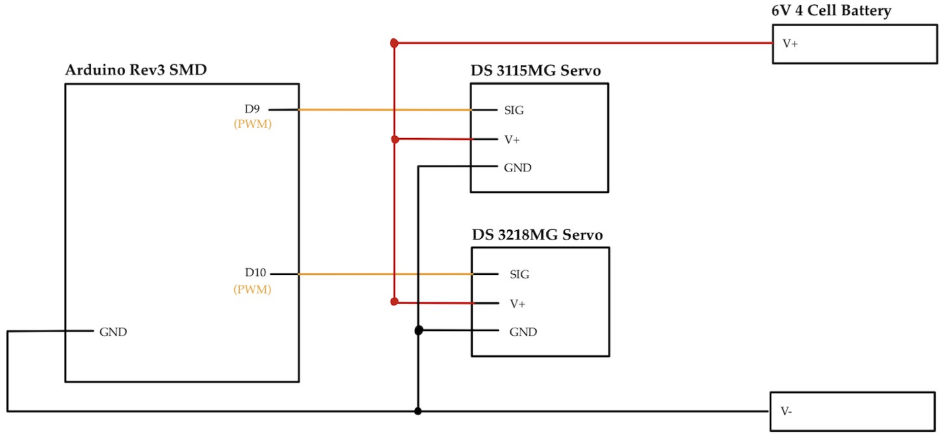


Figure 6: Hardware connections for Servo motors

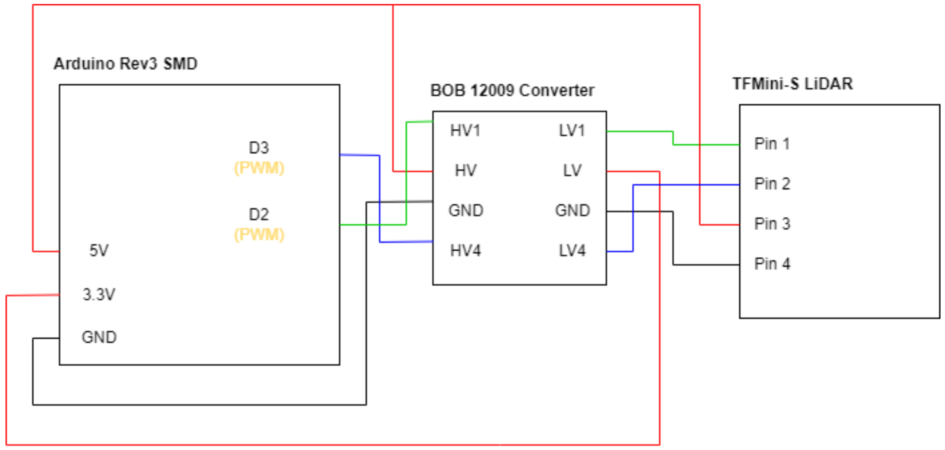


Figure 7: Hardware connections for LiDAR

calibration phase again. Plus, the LiDAR automatically adjusts to different distances from the wall, so it is a relatively simple process to move positions. However, in the future, this is a design spec we would like to focus on.

7.4 Results for Design Specification 4

When the calibration was performed correctly, the projection matched the user's line of sight 100% of the time, as long as the user turned their head within the bounds of use-case requirements 2 and 3. However, if the calibration was not performed correctly, the projection generally did not match the user's line of sight.

7.5 Results for Design Specification 5

We ran 20 trials and had the user read from the left to the right and also up and down. This was to mimic small head movements that a typical user might perform. This correlated to the user altering their current yaw and pitch up to roughly ±8 degrees. We did not want the projection to move at all in these cases. In all trials except one, the projection did not move. Thus, we achieved our success rate of 95% for this specification.

7.6 Results for Design Specification 8

In between prototypes, we gathered 10 students to test the system. We looked for feedback on the overall system and also used this opportunity to figure out which speed and easing function users liked for the movement of the projection. The three easing functions were linear, quadratic, and cubic. The three speeds (in x,y deg/sec) were slow (30, 15), medium (45, 22.5), and fast (60,30). We then had users rate their experience on a 1-10 scale (1=strongly disagree, 10=strongly agree) using the following statements:

1. The projection matches my gaze accurately.
2. The system follows my head movement in real-time.
3. The projection movement is stable and moves cleanly.
4. The system and calibration is easy to use.

All users preferred the cubic easing function at the fast speed. Statement 1 had a satisfaction rate of 70%. We later changed the multiplication factors of the pitch and yaw to increase the accuracy of gaze estimation. Statement 2 had a satisfaction rate of 90%. This is because the system initially had small latency issues for large panning distances. Statement 3 had a satisfaction rate of 70%. Users noted that there was a jitter even after they stopped moving and that there was some jerkiness during panning. We decided to check when the user has stopped moving in either direction before sending the motor commands. We also added some thresholds so that the projection does not move for minor head movements (such that those that occur when reading). This made the movement of the projection very smooth but did increase latency issues. We also tweaked the speed to be slower for smaller panning distances and faster for larger panning distances. This further decreased the jerkiness for small head movements while still allowing the system to move as quick as possible. Statement 4 had a satisfaction rate of 80% and we improved this by simplifying the calibration process so that it takes less time to complete and is easier to understand.

8 PROJECT MANAGEMENT

8.1 Schedule

The schedule is shown in Fig. 9.

8.2 Team Member Responsibilities

We have split up this project into three distinct portions of the architecture, and assigned each to a team member. Each team member assisted in testing and editing the system based on user feedback.

Rama Hassabelnabi is responsible for the mechanics section. This involves researching hardware, testing the motor, securing the projector, writing arduino code, and testing different methods of motor movement.

Olivia Fernau is responsible for the computer vision section. This involves working with MediaPipe to get the facial landmark detection, calculating gaze estimation, working with the raw camera input, and testing the head angle accuracy with the standalone CV gaze estimation code.

Isabel Gardner is responsible for the integration section. This involves working with pySerial to translate the CV output to the mechanical input, researching, designing programming, and testing the startup calibration process, and writing arduino code to synthesize the computer vision input with the saved calibration metrics to provide motor command output.

8.3 Bill of Materials and Budget

The bill of materials and budget are shown in Table 2. Note that we did not end up using the Wyze Cam v3. There were issues integrating this camera into our overall system, so we bought the Arducam 1080P Day and Night vision camera instead. We also had to buy extra batteries and a USB extension cord for our system.

8.4 Risk Management

Risk 1: Our calibration program will need a significant amount of time to refine. We initially thought we would need to get an end-to-end system working by mid-March at the latest in order to begin user testing. However, this was a bit unrealistic due to how much work needed to go into getting the motor functionality, and the completion of the full calibration state machine. To help minimize the amount we would have to correct in calibration, we wrote a testing suite for the user position calculation functions, which helped us verify the correctness of the code without having to test it on the full system. We were able to finish the whole system by early April, which still left us a good amount of time to refine the calibration. We had to redesign the calibration system twice, for optimization and then usability, so this time was truly needed.

Risk 2: Our CV module must distinguish between unintentional and intentional head movements. For instance, if a user is reading something on the projection, their head may be moving slightly in any direction. When this is occurring, we do not want the projection to move or else the user might become irritated. To manage this risk, we placed thresholds on the motor movement to not move with small yaw and pitch changes. We tested this out and confirmed the projection does not move with very small head movements.

Risk 3: Our camera must be able to detect head pose in dim settings. To manage this risk, we opted for a camera with night vision capability and high resolution to capture the user's movement in light and dark settings. We were successfully able to use the system in all ranges of lighting with high accuracy.

Risk 4: Our pan and tilt system must be able to support the weight of the projector. To manage this risk we use a servo motor with a torque rating that is one higher than necessary and limit the angle range of the vertical motion so that the projector never moves all the way up or down. Over the semester, the projector weight never gave our motor any problems, and the vertical angle limit worked very well. We did experience some risk with respect to the servo's battery. The initial rechargeable batteries we chose overpowered and broke the vertical servo, so we had to find a replacement servo and different batteries as soon as possible. In the meantime, we were able to continue testing and refining our system using the horizontal servo until our replacements came in.

9 ETHICAL ISSUES

Our project is intended to help people work from home more comfortably. A central downside of this goal is that working more comfortably can lead to more efficiency and tolerance of longer work hours, which could potentially lead to worker exploitation. For example, one of our main users that we are targeting is radiology technicians in the medical field. The medical field already has an overwork problem,[6] and we intend for this product to be used to endure difficult or strenuous workdays, not lengthen them. Both screen time and overwork have also been noted to be associated with mental health problems.[6]

Another edgecase for operation, on the other end of the spectrum, is that this product could encourage excessive non-work-related screentime. Screen time is already a well-known issue, as computers and smartphones have become an integral part of our lives. Too much screen time can have negative effects on a person's health.[7] Although our product can help increase mobility, mobility isn't a requirement to use our product, and may encourage a more sedentary lifestyle if user's choose not to move or stand while using the product. This can lead to increased blood pressure and a risk for diabetes and other health problems. It could also lead to attention span problems, particularly in children, due to the fast pace of the internet's media.[8]

It can be difficult to untangle products like ours from the way that society in general underestimates and sometimes even encourages overwork. In some sense, our product is a way to cope with an already unhealthy standard of jobs that require employees to stare at a screen for an excessive amount of time daily. It's important to note that if our product was developed and sent out for consumption, it wouldn't be a perfect solution. The perfect solution would be to limit screen time altogether. Unfortunately, due to how interconnected work has become with technology and computers, that would be impossible. The more realistic solution is to develop tools that will help mitigate the difficulties associated with screen time and long work hours.

Some of the mitigations for our ethical risks lie in the legal sphere. Particularly, companies should not make unreasonably long work days a cultural norm, and should limit the number of hours employees can work. However, our product should also discourage overuse in both cases for work or for leisure. One thing we could do to mitigate this is make a warning system that would activate if it seems our user has been using the projector for more than 3 hours straight. For some video game consoles, they have a popup message that suggests that the user should take a break, and we could implement something similar. We could also recommend in our user manual that this product is not intended for children, since children are more susceptible to building harmful habits around electronics.

10 RELATED WORK

As mentioned in the introduction, there are quite a few augmented reality headsets on the market that allow users to project information into space. For instance, the Microsoft HoloLens 2 is on the market for $3500 which allows users to project and interface with holographic material in their field of view using their hands, voice, and eyes. Since the camera and projection are placed on the headset, the user has great mobility. Our system mimics the design of having projected information follow the movement of a user's head. It is much less expensive and the user does not need to wear a bulky headset. However, the user has less mobility and cannot interface with the projection.

11 SUMMARY

Our system's aim is to create a projector viewing experience for the user that can help augment ways to work remotely. By designing a CV program that takes in the user's head angle and estimates their gaze, and integrating it with a mechanic system that moves the projector into a place that matches the projection with the user's eye gaze, we hope to create an interesting and exciting way to look at digital materials. One upcoming challenge is testing the motor and making sure it follows the user's gaze estimation smoothly, and coming up with calibration and refinement mechanisms to improve the user experience. We hope to begin user testing mid-March so we have a buffer of time to cater our project's features towards what user's are looking for. We may have to add to or modify our use-case and design requirements based on user feedback, but our design requirements in this document are our current end goals for a polished system.

11.1 Future work

In the future, we would like to have a better projector and learn how to deal with the skew of the projection when it is turned far to one side. It would also be wonderful to add more interactive gestures into our system. For example, we would have liked to incorporate hand gestures that allow the user to zoom in and out, click buttons, or even draw on the projection.

Currently, our calibration will only lead to successful gaze matching if the user is in relatively the same sitting or standing position. In the future we would like to make it so that the user can move around the room as long as they are within the camera's view.

11.2 Lessons Learned

One lesson we learned was that testing and external input is very important. We had our MVP done fairly quickly but we learned that if we had it done even a week earlier, we would have more time to test out our system on more people and have more time to implement the feedback. We also learned that we should buy double of most of our materials just in case something goes wrong. Waiting for orders to be confirmed, placed, and delivered can take up to a week and this is too much time to be blocked from making progress.

Glossary of Acronyms

- CV – Computer Vision

References

[1] "Is Too Much Screen Time Giving You Eye Fatigue?" *Harvard Business Review*, 11 Oct. 2021, https://hbr.org/2021/07/is-too-much-screen-time-giving-you-eye-fatigue

[2] "Face Mesh." *Mediapipe*, https://google.github.io /mediapipe/solutions/face_mesh.html.

[3] Anwar, Taha, and Rizwan Naeem. "5 Easy amp; Effective Face Detection Algorithms in Python." *Bleed AI*, 12 Jan. 2022, https://bleedai.com/5-easy-effective-face-detection-algorithms-in-python/.

[4] Khalid, Irfan Alghani. "Face Landmark Detection Using Python." *Medium*, Towards Data Science, 6 Sept. 2021, https://towardsdatascience.com/face-landmark-detection-using-python-1964cb620837.

[5] "Serial Communication between Python and Arduino." Arduino Project Hub, https://create.arduino.cc/projecthub/ansh2919/serial-communication-between-python-and-arduino-e7cce0.

[6] Portoghese, Igor et al. "Burnout and workload among health care workers: the moderating role of job control." Safety and health at work vol. 5,3 (2014): 152-7. doi:10.1016/j.shaw.2014.05.004

[7] Mosley, Aris. "The Negative Effects of Screen Time for Adults and Children-Blog-Valleywise Health." Blog–Valleywise Health, 10 October 2020, https://blog.valleywisehealth.org/negative-effect-of-screen-time-adults-children/. Accessed 7 May 2022.

[8] Doheny, Kathleen. "Too Much Screen Time Can Threaten Attention Span." MedicineNet, https://www.medicinenet.com/script/main/art.asp?articlekey=117789. Accessed 7 May 2022.

Table 1: Lidar Tradeoffs

	Range	Weight	Error	Price
Lite v3	40m	22g	+/-2.5cm	129.99
Lite v4	10m	14.6g	+/-5cm	64.5
TFMini-S	12m	5g	+/-6cm	45.95

Table 2: Bill of materials

Description	Model #	Manufacturer	Quantity	Cost @	Total
PVO Projector	YG300Pro	PVO	1	$58.99	$58.99
TFMini-S	A01	Benewake	1	$42.95	$42.95
Logic Level Converter	BOB-12009	SparkFun	1	$3.50	$3.50
Pre-Assembled 2 DoF Pan Tilt Digital Servo Kit	U6115	UCTronics	1	$89.99	$89.99
Shipping from Vendor				$38.00	$38.00
DS3218 20kg High Torque Servo Motor		ANNIMOS	1	$16.99	$16.99
Projector Stand		Nelhat	1	$37.99	$37.99
Standard 1/4"-20 Threaded Tripod Screw Adapter		Anwenk	1	$1.60	$1.60
AA Battery		Energizer	24	$0.75	$17.99
4 Cell AA Battery Holder			1	$7.00	Borrowed
Mini Solderless Breadboard		DaFuRui	2	$1.16	Borrowed
Arduino Uno Rev3 SMD	A000073	Arduino	1	$21.90	Borrowed
Arducam 1080P Day/Night Vision USB Camera		Arducam	1	$34.99	$34.99
USB 2.0 Extension Cable		AmazonBasics	1	$6.40	$6.40
Double sided mounting tape		Gorilla	1	$9.94	$9.94
Wyze Cam v3	v3	WYZE	1	$35.98	$35.98
					$395.31

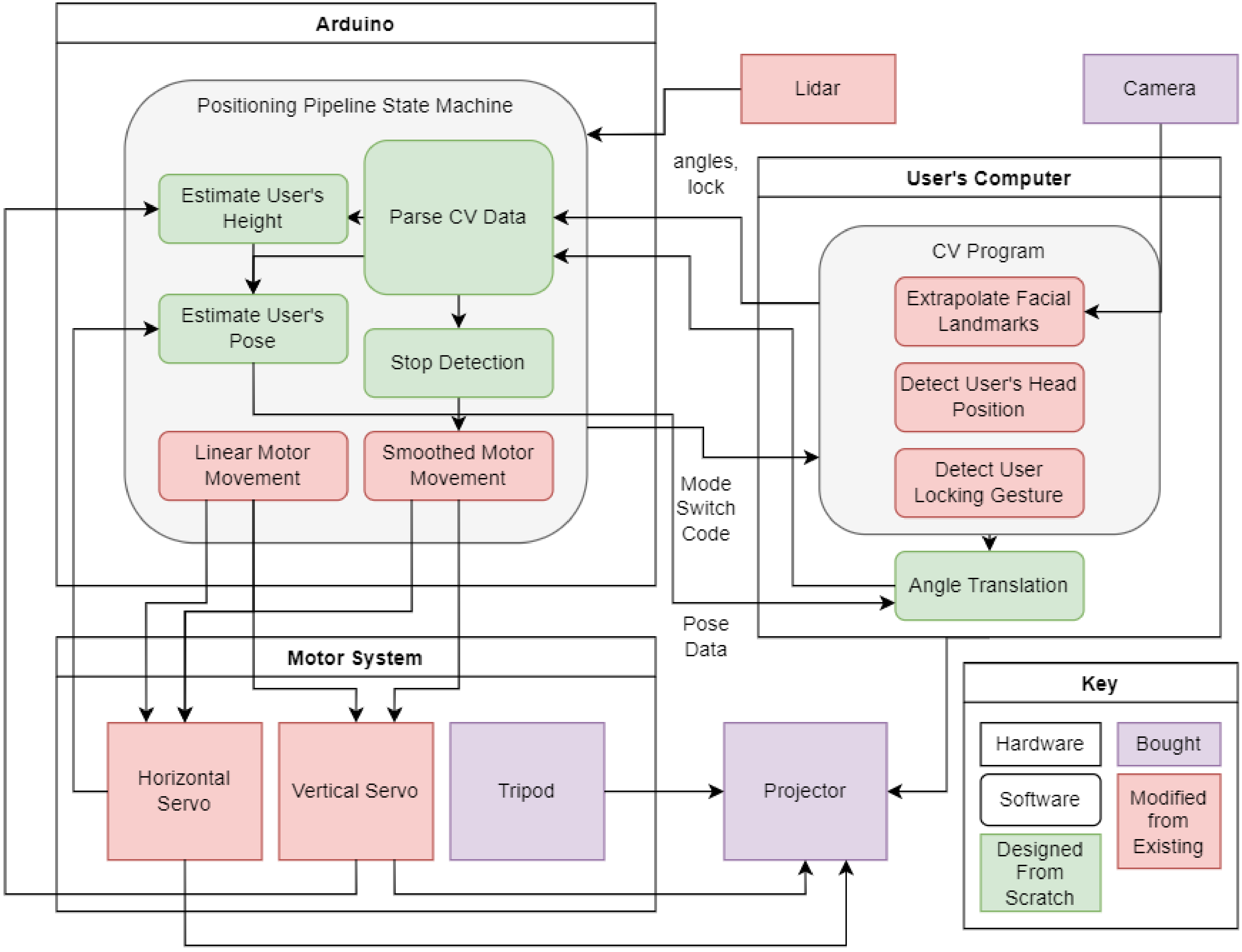


Figure 8: A full-page version of the same system block diagram as depicted earlier.

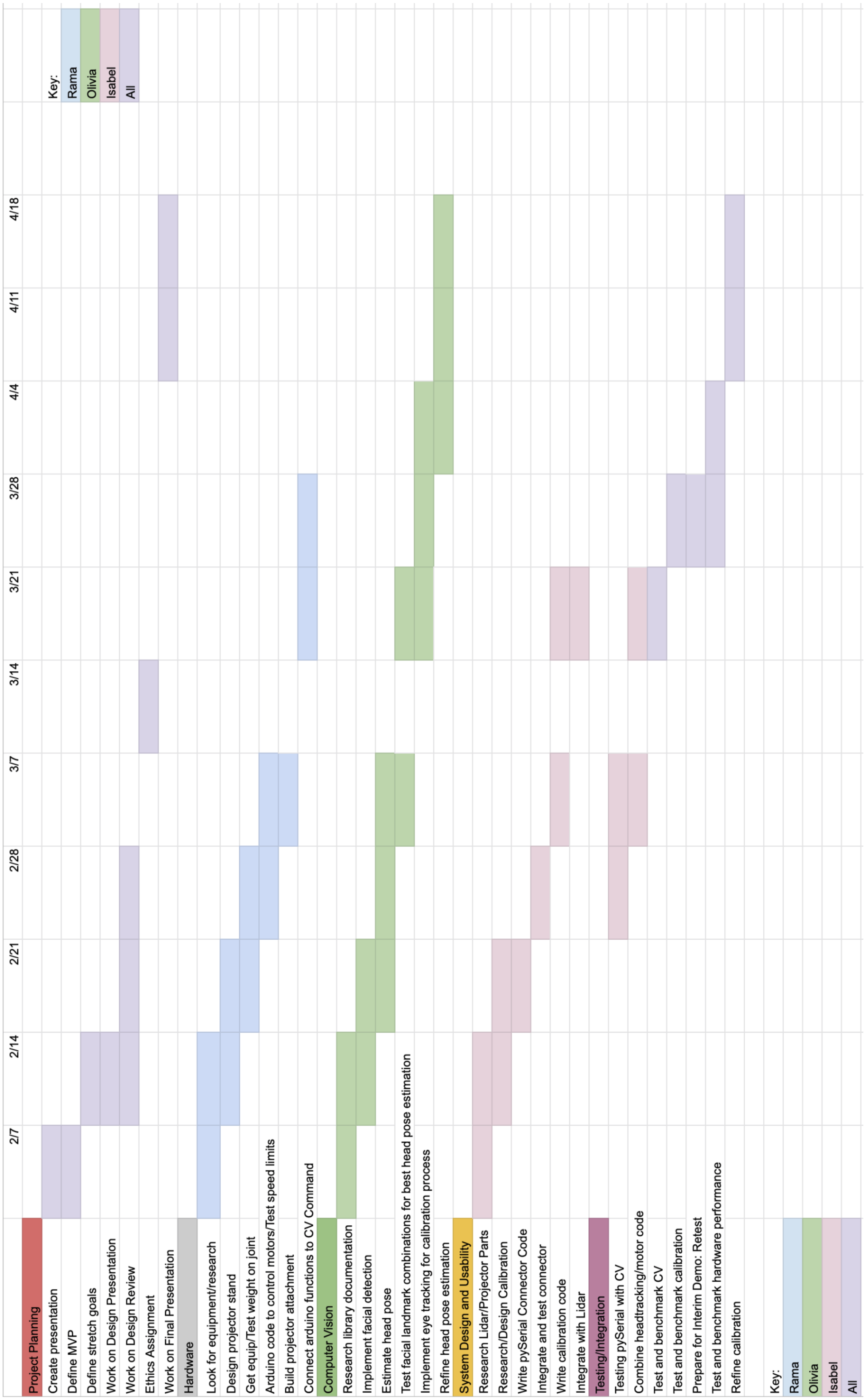


Figure 9: Gantt Chart