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

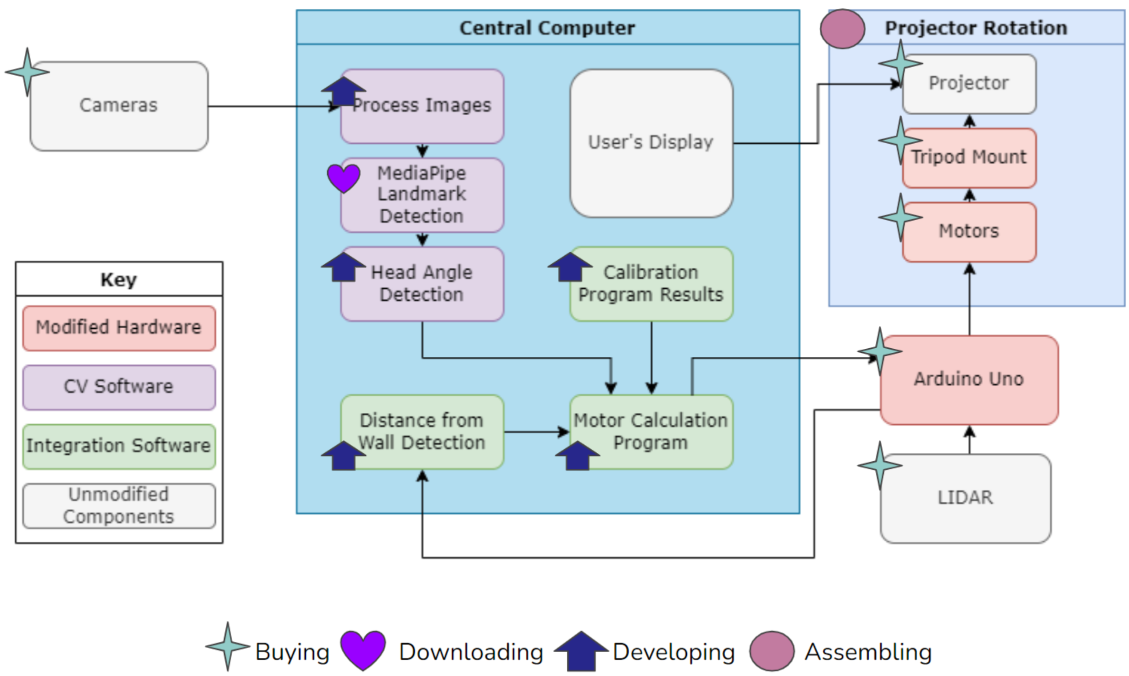


Figure 1: Block Diagram

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.

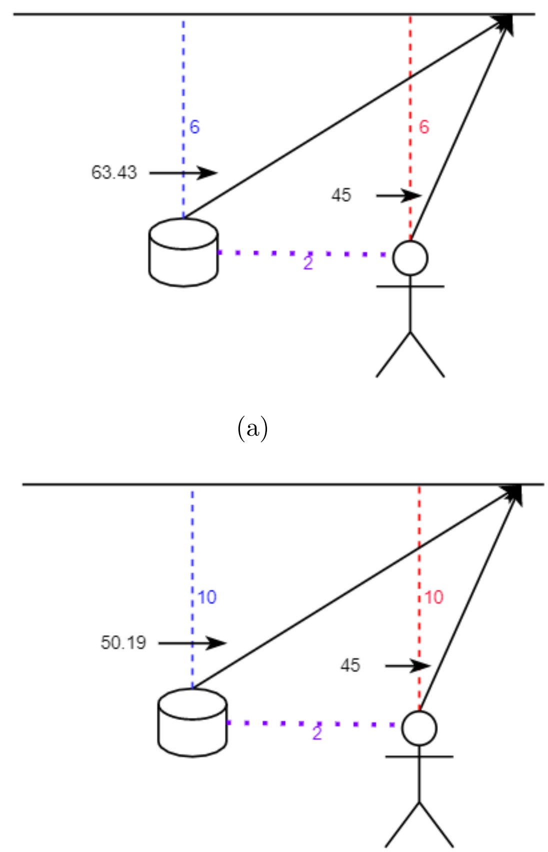
Requirement 5: Small head movements, which would result in less than 0.25 meters of change for the projection center, must be ignored by the system. This is to prevent the projection from moving in an annoying, unnecessary way.

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 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 Wyze Cam v3 camera which has night-vision capability, is easy to mount, is inexpensive, and has an HD resolution spec of 1080p. These specs are more than sufficient for Mediapipe and should allow facial landmark detection in dim settings.

5.3 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.

5.4 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. 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. In this case, the Garmin LIDAR-Lite v4 is the better choice for our usecase.

5.5 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.

5.6 Battery Selection

The servo motors at most require 7-7.2V. Since we want the system to be portable, we do not want a wall charger so we decided to use rechargeable batteries. We considered the "Ovonic 2s 50C 5200mAh 7.4V LiPo Battery," and the "Powerextra 3600mAh 6 cell 7.2V NiMH Battery Pack."

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 LiPo battery with a 5200mAh capacity, a single charge would last 5.2 / 3 h = 1.73 h which is about 1 h 44 min. This is a suitable battery life because the user will be able to have a focus time of nearly 2 hours, and needing to plug this in can serve as a break time. The LiPo battery can safely be charged at a rate of 1C which equates to 5.2A. Thus, it would take 1 hour to charge.

We then consider the NiMH battery. With a 3600mAh capacity, a single charge would last 3.6 /3 h = 1.2 h which is about 1h 12 min. This is also an acceptable battery life. This battery has a recharge rate of 1.8A so it would take 2 hours to charge, but reviews say that it may take up to 5 hours depending on the charger. Though it is more expensive, the LiPo battery has a better battery life and charge time so we selected the Ovonic battery.

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.

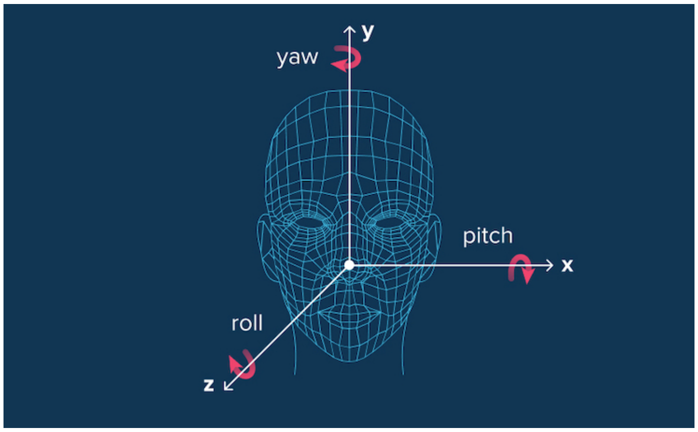


Figure 3: Euler Angles

6.2 Integration System

6.2.1 Calibration

When 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. The user will have to first 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.

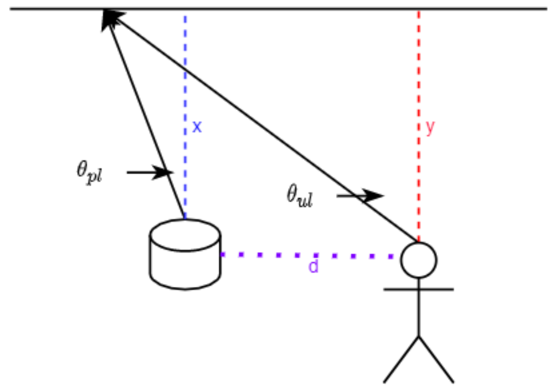
The user will then center the projector on their field of vision while looking approximately 45 degrees to the left. The more exact angle can be calculated by our CV program, but having the user look approximately at this angle should give our program clearer data. Again, the user will line up the projector with their field of vision using the CV program, giving us the angle of the projector (θ_{pl}) and the angle of the user (θ_{ul}). Both of these measurements will also be stored as radian floats in our program.

Finally, the user will then center the projector on their field of vision while looking approximately 45 degrees to the right. The angle will again be error corrected by our CV program, and again the user can adjust and line up the projector using their head movements. Our program will then store this new angle of the projector (θ_{pr}) and angle of the user (θur).

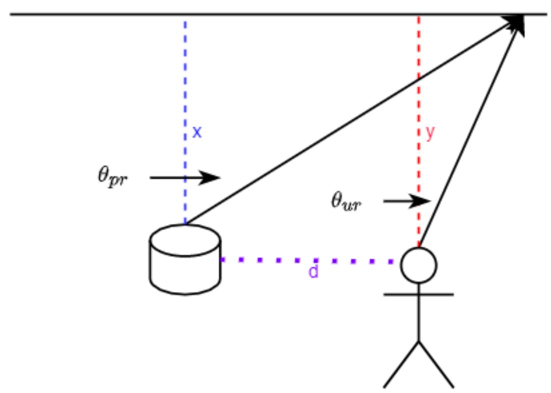
Both the left and right calibration phases are illustrated by Fig. 4. These are in a 2D plane facing downwards on the system, with the d line representing the 'horizontal axis' and the y line representing 'vertical axis'. Using the four angles gathered from the left and right calibration measurements, 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 below 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.

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

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



(a)



(b)

Figure 4: Top-Down Calibration. (a) Left side (b) Right side

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. We can then read the serial string into a float representation of the user's head angle, and pass this into another function that will convert this angle to a projector angle, using the results from calibration. Then, once we return the results, the projector angle will be converted to motor commands, beginning the mechanical process.

6.3 Hardware Connections

Two PWM pins on the Arduino would be used for the two servo motors. The Arduino requires 7V and the servo motors would be connected in parallel and require 7.2V. The 7.2V NiMH battery pack would power the Arduino and the servo motors. Refer to the diagram below in Fig. 5 – note that orange lines correspond to PWM pins, blue lines correspond to power, and black lines correspond to ground.

6.4 Software Communication

Communication between the Arduino and the integration modules will be through a USB cable. The previous and current position of the user's head and the speed at which they rotated in each direction is used by the Arduino functions to move the motors accordingly. The speed of the motors will follow almost a parabolic pattern, where the motors move slower at first, then faster, and slow down again into the final position. This is to avoid jerky movements of the projector.

7 TEST & VALIDATION

7.1 Tests for Design Specification 1

We will run 20 trials and time from when CV information is calculated to when the motor receives the command. The average time for all trials should be less than 200 ms.

7.2 Tests for Design Specification 2

After the calibration process, the user will rotate their head left, right, up and down. The angle at which head rotation calculations break down will be identified for each direction. All angles should be greater than 45 degrees.

7.3 Tests for Design Specification 3

After the calibration process, the user will perform translations in the x, y, and z directions. The distance from the initial position at which projection placement accuracy breaks down will be identified for each directions. All distances should be greater than 1 meter.

7.4 Tests for Design Specification 4

We will run 20 trials where a user moves their head, within the specified bounds, from point A to point B. We intend for the projection to match up with the user's line of sight 95% of the time.

7.5 Tests for Design Specification 5

Based on the calibration calculations, the maximum head rotation that would result in less than 0.25 meters of change in the projection will be calculated. Then, we will perform these small head rotations to ensure the projection stays in place. We will run 20 trials and aim for a success rate to be 95% or greater.

7.6 Tests for Design Specification 8

In between prototypes, we intend to gather roughly 5 users to test the system and rate their experience on a 1-10 scale (1=strongly disagree, 10=strongly agree) using the following statements:

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

We aim for each statement to have an average rating of 9 or above by the final prototype, resulting in an overall satisfaction rate of 90% or above. During earlier prototypes, we do not expect the ratings to be high. However, we do intend for the average ratings to increase for each statement as each prototype is tested.

8 PROJECT MANAGEMENT

8.1 Schedule

The schedule is shown in Fig. 7.

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.

Rama Hassabelnabi is responsible for the mechanics section. This will involve 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 will involve 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 will involve 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.

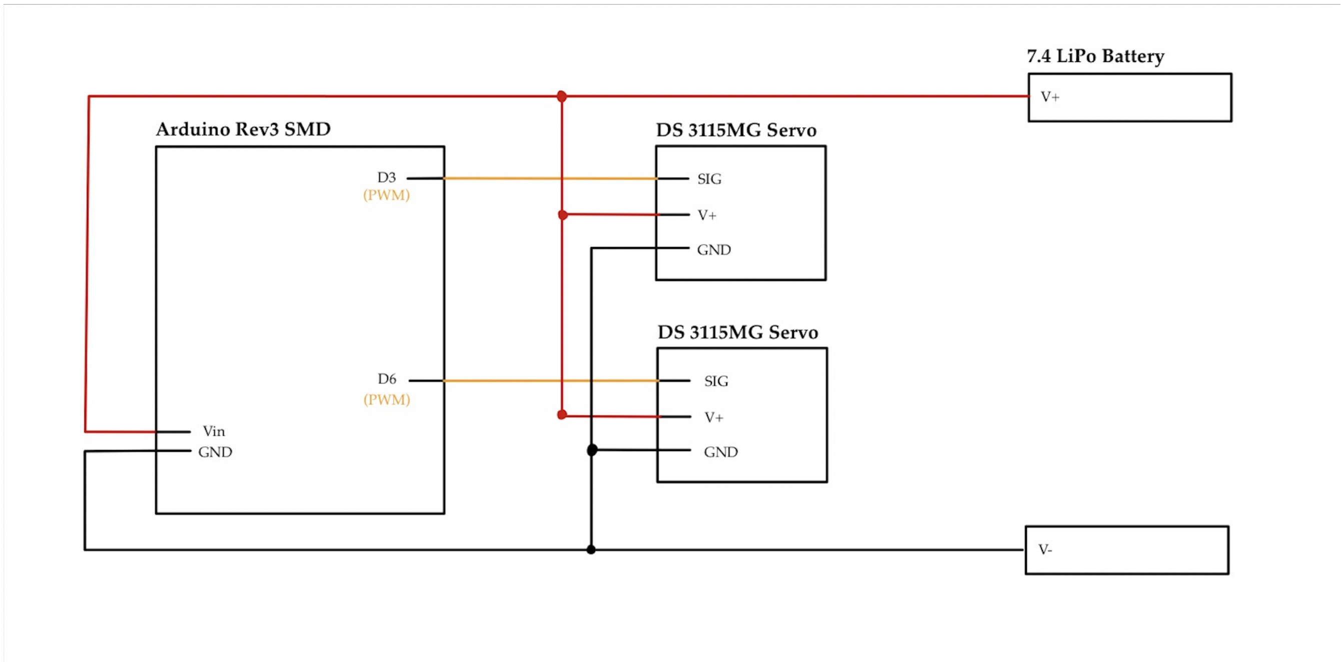


Figure 5: Hardware connections for Servo motors

8.4 Risk Mitigation Plans

Risk 1: Our calibration program will need a significant amount of time to refine. This means we will need to get an end-to-end system working by mid-March at the latest in order to begin user testing. Currently we are on track to completing this deliverable, but may run into additional roadblocks as we begin testing different calibration methods. To mitigate this risks, we're going to keep our product fairly flexible and use the results from user testing to fit the user as best as possible.

Risk 2: Our CV module may have difficulties distinguishing between unintentional and intentional head movements. For instance, if a user sneezes or nods their head yes or no, our system might interpret that as a head movement and send a command for the motor to move. To avoid this, we intend to put caps on the motor movement. If the user moves too rapidly or only moves their body slightly, the projection should not move. We will likely need to average the head pose between frames to account for the small body movements.

Risk 3: Our camera must be able to detect head pose in dim settings. To mitigate this risk, we opted for a camera with night vision capability and high resolution to capture the user's movement.

9 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.

10 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.

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.

Table 1: Lidar Tradeoffs

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

Table 2: Bill of materials

Description	Model #	Manufacturer	Quantity	Cost @	Total
PVO Projector	YG300Pro	PVO	1	$58.99	$58.99
LIDAR-Lite v4	v4	Garmin	1	$64.50	$123.49
Pre-Assembled 2 DoF Pan Tilt Digital Servo Kit	U6115	UCTronics	1	$89.99	$89.99
Shipping from Vendor				$38.00	$38.00
7.4V 5200mAh 2S 50C LiPo Battery		Ovonic	1	$17.00	$34.99
Balance LiPo Battery Charger (2S-3S)	B3AC	HTRC	1	$12.97	$12.97
Arduino Uno Rev3 SMD	A000073	Arduino	1	$21.90	Borrowed
Wyze Cam v3	v3	WYZE	1	$35.98	$35.98
					$394.32

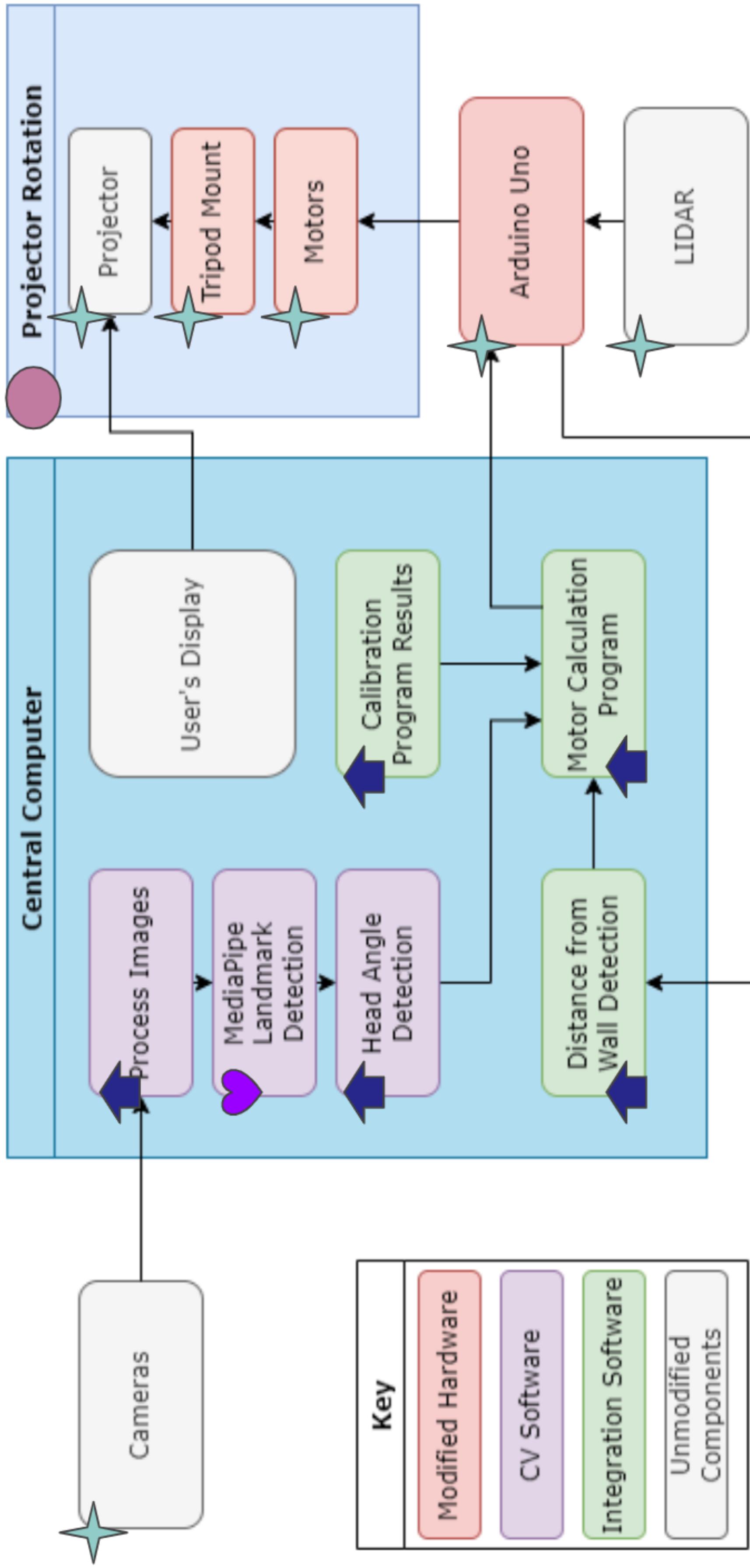


Figure 6: A full-page version of the same system block diagram as depicted earlier.

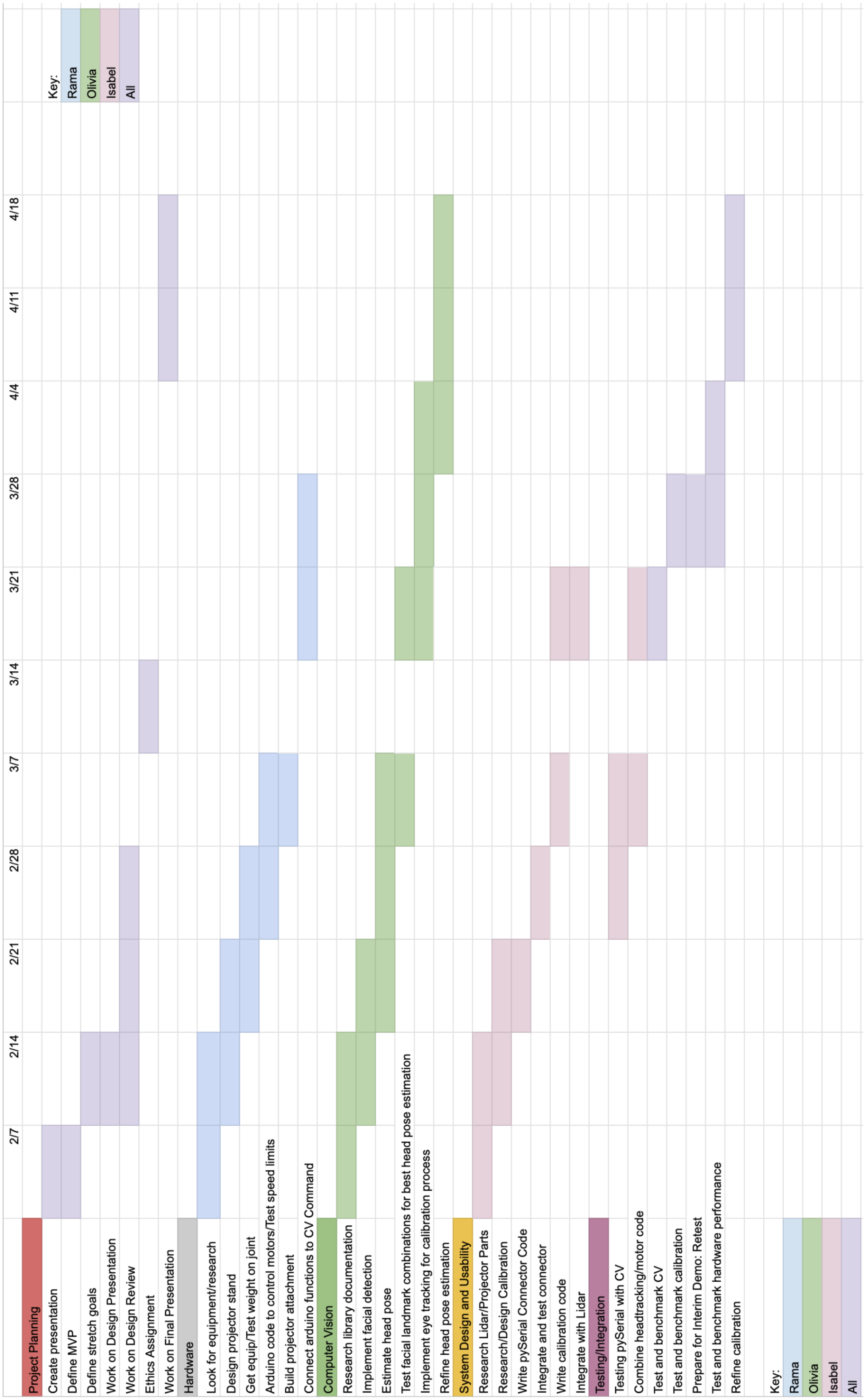


Figure 7: Gantt Chart