# Auto-Focusing and Color Identifying Lenses for Mild Visual Impairments (P.A.U.L.)

Genaro Morales, Julie Gallagher, Kayla Stanley, Midkhat Satdanov

Dept. of Electrical Engineering and Computer Science, College of Optics and Photonics, University of Central Florida, Orlando, Florida, 32816

***Abstract* – This paper presents the methodology and parts used to design a headset that enables the user to receive accommodation for the focusing power of their eyes and to identify colors in their surroundings. These portions are sectioned into: (1) Color Identification; (2) Autofocusing; (3) PCB Design; and (4) Software Design. Detailed lens design drawings and simulations comprise a large portion of the paper, and there is a strong focus on attaining proper optical power through the combination of lenses. The system relies on both an ESP32 and a Raspberry Pi 4 to function, with a full suite of accompanying programs**

***Index Terms – Autofocusing lenses, color identification, optical power correction, RGB color detection, raspberry pi 4***

## I. Introduction

Colorblindness and vision disorders affect billions of people worldwide [1]. Our project was born from the desire to help these people with a relatively cheap and versatile tool that would assist them in day-to-day tasks.

P.A.U.L. is the end result of these efforts. Our project includes a system of moving lenses that help focus the user's vision on both near and far objects, coupled with two color identification systems that will help users identify wavelengths their eyes may not be able to perceive. These systems work in tandem to increase quality of life for those who struggle with such issues.

Each component was thus researched with both cost-effectiveness and accessibility in mind, as we wanted to ensure a lightweight headset that would feel practical to wear and use.

The expectations placed on each member of the team was to research and design systems that would conform to IEEE (Institute of Electrical and Electronics Engineers) and CIE (International Commission on Illumination) standards while adhering to our mission statement of cheap, accessible technology.

It is important that we first discuss how we went about correcting minor visual impairments. Myopia, for example, is a refractive error [2], meaning that it can be corrected by adjusting the optical power of the user so that their eyes are recorrected to 60 diopters (D). In our system we accomplish this by moving a set of lenses forwards and backwards to adjust optical power, thus correcting any values that veer from the desired 60 D. The same principle can be applied towards any focusing power impairments.

As for color identification, P.A.U.L. first relies on the use of a camera to identify colors in front of it. The camera connects to a raspberry pi, where this information is calculated on a set of RGB values. A similar principle is applied to our colorimeter, which shines the light from an LED onto the object in focus and detects the intensity of the RGB values that are reflected off it. To better understand these, we have included a block diagram of our hardware design as can be seen in Figure 1 below.

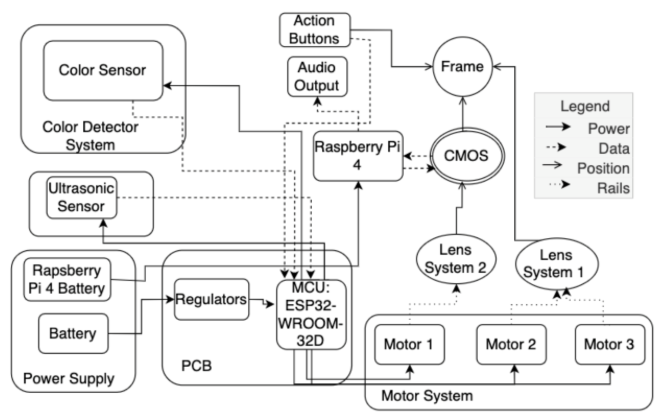



*Figure 1 – Block Diagram*

## II. Color Identification System

P.A.U.L. relies on a wide variety of lenses to function. Our colorimeter utilizes a telephoto lens to focus reflected light onto the photodetectors that comprise the central portion of its board. To determine what kind of telephoto lens we would need, extensive work was undertaken in Zemax OpticStudio to plot out potential lenses. We knew that we needed a small lens system to fit onto our board, and thus constrained our design to a maximum length of 15 mm, although ideally even shorter, and a diameter of 10 mm. After researching telescope systems we found that a common design was to incorporate a large positive lens

| | Surface Type | Comment | Radius | Thickness | Material | Coating | Clear Semi-Dia | Chip Zone | Mech Semi-Dia | Conic | TCE x 1E-6 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | OBJECT Standard | | Infinity | Infinity | | | 0.000 | 0.000 | 0.000 | 0.0... | 0.000 |
| 1 | STOP (a Standard | | Infinity | 2.000 | | | 6.500 U | 0.000 | 6.500 | 0.0... | 0.000 |
| 2 | Standard | | 44.391 F | 2.500 | BK7 | | 3.250 | 0.000 | 3.250 | 0.0... | - |
| 3 | Standard | | Infinity | 1.233 V | | | 3.191 | 0.000 | 3.250 | 0.0... | 0.000 |
| 4 | (aper) Standard | | 5.000 | 2.000 | BK7 | | 3.250 U | 0.000 | 3.250 | 0.0... | - |
| 5 | Standard | | Infinity | 1.983 V | | | 2.851 | 0.000 | 3.250 | 0.0... | 0.000 |
| 6 | (aper) Standard | | -7.140 | 1.500 | BK7 | | 3.250 U | 0.000 | 3.250 | 0.0... | - |
| 7 | Standard | | Infinity | 2.043 V | | | 1.858 | 0.000 | 3.250 | 0.0... | 0.000 |
| 8 | (aper) Standard | | 6.250 | 3.000 | BK7 | | 5.000 U | 0.000 | 5.000 | 0.0... | - |
| 9 | Standard | | Infinity | 2.000 V | | | 0.724 | 0.000 | 5.000 | 0.0... | 0.000 |
| 10 | IMAGE Standard | | Infinity | - | | | 0.060 | 0.000 | 0.060 | 0.0... | 0.000 |

*Figure 2 – Lens Data*

with a smaller negative lens [3], thus producing magnification based on this formula:

$$M = -\frac{P_I}{P_O}$$

Where $P_I$ of the lens closest the image and $P_O$ is the power of the lens closest to the object or light source.

To simulate potential designs, we created a lens system involving a larger planoconvex with a smaller planoconcave in Zemax. We found two potential lenses from Edmund Optics, #37-782 and #21-303, with their properties listed in table 1 below:

| Lens | Radius of Curvature | Center Thickness (mm) | Diameter (mm) | Focal Length |
|---|---|---|---|---|
| **Plano-convex** | 9.045 | 2.80 | 10.0 | 17.5 |
| **Plano-concave** | -6.200 | 1.5 | 6.00 | -12.0 |

*Table 1 – Edmund Optics Lenses*

They were chosen to fit our parameters and to achieve the cheapest price possible. After putting them in Zemax and attempting to optimize their air gaps, the lens system seen in Figure 3 was produced:

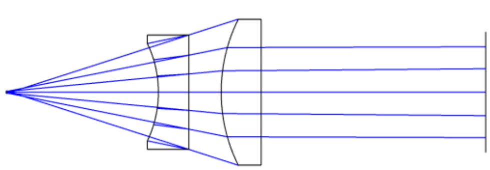


*Figure 3 – Simple Telescopic System*

While this does produce reliable telescopic vision, the primary drawback is the total length that would be required for such a system. The TOTR (total track length) value for this system was 18.8 mm, well beyond the desired 15 mm maximum. Although not as much of a concern, a two-lens system would also run into issues with chromatic aberrations and would overall be far more expensive, as the example lenses chosen would have run a cost of $70.50 before tax.

Thus, we instead investigated 3-lens and 4-lens systems. A different approach was additionally taken here. After researching existing telescopic lenses on the market, we found that they often incorporated a large planoconvex lens, two smaller planoconvex lenses, and a powerful planoconcave lens. With this information in mind, we decided to try and recreate such a lens system in Zemax and then purchase an existing product that resembled our design.

Within Zemax we set the values of our merit function editor to make it such that the optimization algorithms would attempt to make the lenses as close to an effective focal length (EFFL) and total system length (SUMM) of 12 mm each. Setting the weight to 1 ensured that the optimization procedure would do its best to hit these targets. The functions, as well as actual values after optimization, are shown in Table 2 below.

| Function | Target | Weight | Actual Value |
|---|---|---|---|
| **EFFL** | 12.000 | 1.000 | 9.916 |
| **SUMM** | 12.000 | 1.000 | 19.833 |

*Table 2 – Merit Functions*

The resulting lens data is noted in Figure 2 at the top of this page. The lens design is shown below in Figure 4:

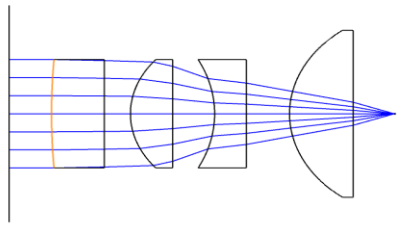


*Figure 4 – Final Lens System*

After scouring the market, we found that OpenMV's 4x telephoto lens was a very close match to our design and thus incorporated it into the colorimeter

As for the colorimeter itself, it incorporates four Chanzon LEDs that are 2 x 5 x 7 mm in dimension and boast an incredibly bright 1.5 cd. Their color temperature of 9000 K and brightness made them worthwhile additions due to their similarity to daylight. They are necessary to illuminate the surface of whatever the colorimeter is focused on, after which the reflected light falls onto its matrix of photodiodes, which interpret the reflected color. Our final colorimeter can be seen in Figure 5.

The photodiodes are arranged in an 8x8 grid, with three sets of 16 diodes possessing red, green, and blue filters, and the final set lacking a filter. By connecting the colorimeter to an ESP32, we can calibrate the sensor by shining it onto a black and a white object to obtain the extremes of its calibration, and then the colorimeter is set to begin sensing data.

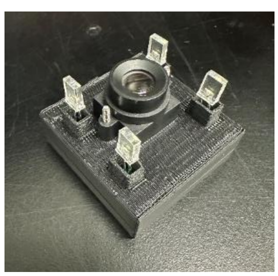


*Figure 5 - Colorimeter*

The colorimeter interfaces with a CMOS sensor, the Pi Camera Module v2, that performs a similar function. The v2 runs a program that will be discussed in section V to interpret colors in front of it. The colorimeter and camera then compare results with one another in the raspberry pi, and if the same color is detected then the color identification is considered a success. These functions are activated by buttons on the side of the system, which will be explored further in PCB design.

## III. Autofocusing System

As for the autofocusing system, which encompasses the lenses that will be used to provide focusing power for the user, we had to incorporate the human eye in consideration for this subsystem. As such, we'll be showing a model of the human eye to demonstrate where the lens setup will place the focal point within it.

We required three sets of two lenses for this setup that would provide the necessary optical power claimed in our specifications of -1.5 D to +2.5 D. To accomplish this, we purchased a set of biconvex and biconcave lenses from Eisco. The biconvex lenses had a 38 mm diameter, and the biconcave lenses had a 50 mm diameter. Additional biconvex lenses with diameters of 38 mm were purchased for the CMOS sensor lens design.

For our design, we aimed to create linear motion of multiple lens systems with the help of motors, carefully considering factors such as weight, torque, and the dimensions of various motors. We narrowed our options down to two main categories: servo motors and stepper motors, as both met most of our required characteristics.

Although stepper motors are commonly employed for linear motion with lead screw mechanism, our concerns arose from the additional circuitry and weight they would introduce to the design. As a result, we explored continuous rotation servo motors to identify the best option.

Ultimately, our decision came down to which motor would minimize the overall weight, ensuring that the design remained wearable for longer periods of time without weighing on the user quickly. The motor we picked was the MG90S servo motor, a continuous bidirectional servo. Table 3 below illustrates some of the main capabilities of the motors.

| Motor | Torque (5V) | Weight | Operating Voltage | Speed (5V) |
|---|---|---|---|---|
| MG90S | 2.0 kg*cm | 13.4 grams | 4.8V – 6.0V | 0.12 seconds per 60° |

*Table 3 – Motor Data*

Another critical consideration for ensuring the proper functionality of our design was the selection of the lead screw type. Initially, we experimented with a small pitch size to achieve finer step sizes. However, we encountered a recurring issue where one of the motors in the dual motor lens system would become jammed, while the other would continue running. To address this, we transitioned to a trapezoidal lead screw with a higher pitch, despite the trade-off of increased weight and size compared to the original lead screw. This adjustment significantly improved the system's performance, resulting in smoother movement and eliminating the jamming issue. As a result, we ultimately selected a 100mm long trapezoidal lead screw with a 2mm pitch and an 8mm lead.

The movement of this system produces the desired changes in optical power while their carefully considered size and focal length minimizes changes in magnification so as not to become disorienting for the user. As an example, consider Figure 6 below which demonstrates the system when it provides 0 D of optical power for a standard human eye. Figure 7 below displays a zoomed in view on the retina:

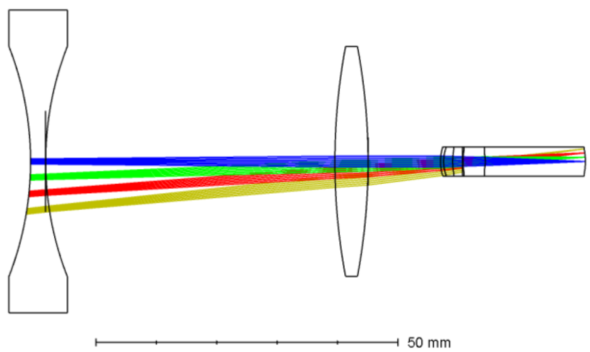



*Figure 6 – 0 D Primary Lens System*

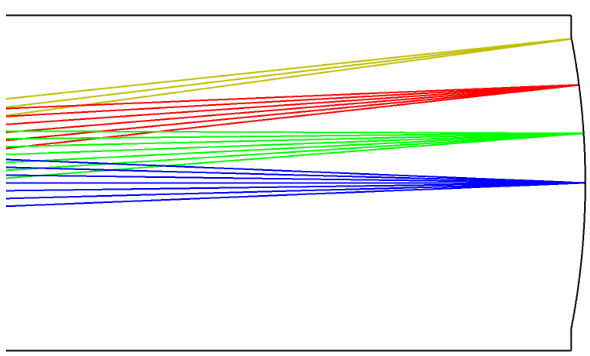


*Figure 7 – Retina*

Here, the complex makeup of the eye is represented by a system of small lenses at the right side of the figure. As can be seen in Figure 7 above, light passing through the biconcave and biconvex lenses perfectly focuses onto the back of the eye (the retina), ensuring perfect focus for an individual with no focusing issues. For users who do have focusing issues however, the movement of the lenses will fit to their prescription. Table 4 demonstrates our achieved values:

| Distance between Lenses (mm) | AMAG | EFFL (mm) | RECI (D) |
|---|---|---|---|
| 40.38 D | 0.573 | -0.667 | 1.5 |
| 47.922 | 0.497 | 1.00E+10 | 1.00E-10 |
| 60.492 | 0.370 | 0.4 | 2.5 |

*Table 4*

The distance between lenses determines the optical power, with the scenarios ranging from -1.5 D, 0 D, and +2.5 D respectively. AMAG denotes magnification, EFFL denotes effective focal length, and RECI takes the reciprocal of EFFL, giving us actual diopters. It should be noted that due to Zemax's limitations, the EFFL of scenario 0 D should be infinite but must instead be calculated to a very large, yet finite value. This makes its RECI value very small, approximating 0.

The secondary set of lenses that are attached to the CMOS camera are used for the autofocusing system, along with the HC SR04 ultrasonic sensor, to determine how far and in which direction the primary lenses must move to properly accommodate the user. The exact program is discussed section V. Since the lenses in front of the CMOS are identical in function to those in front of the eyes, the system attempts to match their optical powers while accounting for the user's prescription.

## IV. PCB DESIGN

### A. Microcontroller Unit (MCU)

For our project, we decided to design and build a custom ESP32 development board rather than designing a Raspberry Pi 4 board, as it was a strong option during our research portion. The primary reason for this decision was based on its capabilities: low-power consumption, simplified architecture and smaller form. Considering too that it has strong community support as it offers available documentation and development resources that will come helpful in the coding portion of our project. The ESP32 allowed us to design a custom board that was tailored to all our requirements without complexity of working with a full single-board computer like the Raspberry Pi 4.

While we are not building a Raspberry Pi 4 we will be acquiring it to run the primary computing tasks. We have chosen to build the ESP32 development board to handle specific peripheral interfaces and communication tasks. This hybrid approach allows us to focus the Raspberry Pi on the CMOS camera that will require strong image processing, while utilizing the ESP32 for the Ultrasonic Sensor, three MG90S servo motors and TCS3200 Color

Sensor Board. We will be utilizing the Arduino IDE software to program our ESP32.

B. Printed Circuit Boards (PCB)

For our project, we have designed four custom boards, using the online design tool EasyEDA for creating both the schematics and PCB layouts. The decision to use four boards, rather than a single, was driven by the strategic placement of our sensors and peripherals.

The first board, denoted as the Main PCB, houses the microcontroller ESP32, CP2102 chip, AMS1117 3.3V regulator and all necessary pin headers for the peripherals. The second board is dedicated to the 5V regulator, the LM2596. This is on its own separate board to facilitate easier replacement and troubleshooting. The schematic of our main PCB and 5V regulator PCB is available in the PCB design section of our team's final report.

The third board incorporates the TCS3200 color sensor, following a standard online schematic for integration. This houses an array of photodiodes with RGB filters that convert light into frequency signal that allows us to detect colors using the ESP32. The schematic of this board is presented in Figure 8 below.

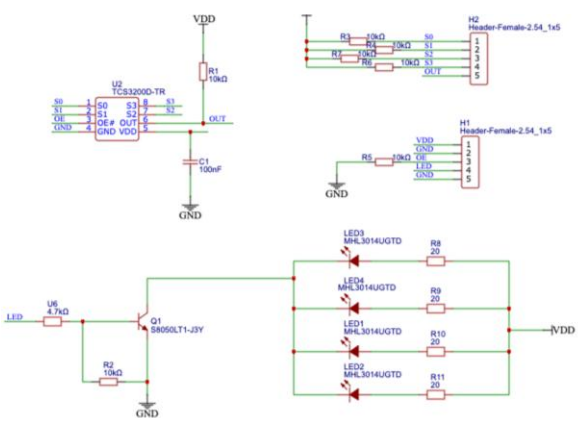


Figure 8 – TCS3200 Integration Schematic

The final board is the Buttons board, which features four push-down switches used to trigger autofocusing and color detection functionalities.

We manufactured our boards to an overseas supplier, JLCPCB to be specific. JLCPCB is well known for its competitive pricing and user-friendly website. Their platform integrates greatly with the EasyEDA PCB design software we used, making it straightforward to place orders directly from the design tool. The boards we have received so far show no errors or missing components, indicating that we have received quality products.

The assembly of the PCBs was carried out at the senior design lab, facilitating efficient testing and troubleshooting throughout the process. While the first set of boards required some troubleshooting due to design mistakes, the second iteration of the boards performed as expected, with no further issues. Figure 9 below shows the second iteration of the Main PCB board and the 5V regulator after assembling.

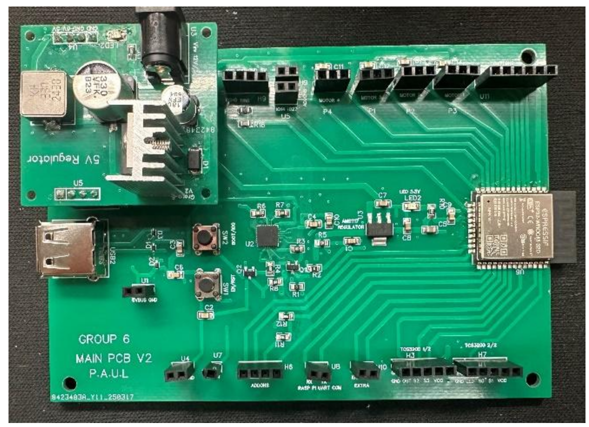


Figure 9 – 5V Regulator

C. Battery Selection

We conducted an analysis of various battery types to ensure our design would be both rechargeable and capable of meeting all the power demands for the integrated peripherals, including the ESP32 microcontroller, motors, ultrasonic sensor and colorimeter. Given the importance of long-lasting performance and reliable power delivery, we focused on selecting a battery that would ensure the longevity and durability of the overall system.

After evaluating multiple options, we chose Li-Ion (Lithium-Ion) batteries due to their outstanding characteristics, including high energy density, long lifespan, and reliable performance. Their batteries are known for their ability to deliver consistent power over extended periods, making them well-suited for our design.

To meet our specific power requirements, we selected the Talentcell 12V/6Ah Rechargeable Li-Ion battery pack. This model was chosen for its optimal balance between power demands of all components in our system. The battery pack comes with a jack connection therefore our regulator has an input jack connector. Given the size and weight of the battery pack, we determined that the most practical placement for the battery would be in an arm pocket. This placement allows for better weight distribution and ensures both functionality and comfort.

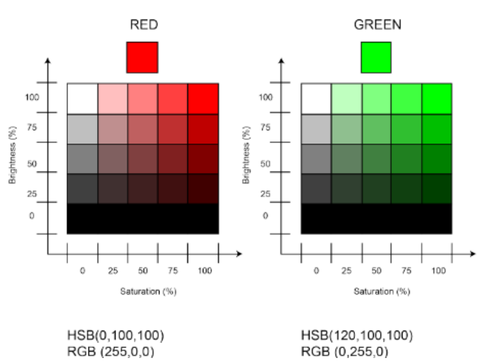
## V. Software Design

For our software design and setup, P.A.U.L. utilizes a Raspberry Pi 4 as its primary central processing unit due to its strong computational power, affordability, and compatibility with popular open-source libraries. Its ability to efficiently execute image processing along with its compact size benefits the build of our project. In conjunction with the ESP32 that will run code to remaining peripherals.

For our specific design, we have two separate systems: a color-identification system and an auto-focusing lens system. The color-identification system utilizes an algorithm to capture and process images to identify colors through the RGB color system. The auto-focusing lens system uses a Sobel filter edge detection algorithm that controls a dual lens setup to adjust lens positioning continuously until focus is achieved.

Firstly, let's break down the color-identification algorithm. Our goal is to achieve at least 80% accuracy in identifying one or more colors simultaneously. We utilized the Python OpenCV library which handles image processing, region-of-interest (ROI) isolation, and color conversion to RGB (red-green-blue) color model. The program is written entirely in Python in the Visual Studio Code IDE.

The pressing of the color button triggers the process to start, it then utilizes the Raspberry Pi Camera Module to capture an image which is then uploaded to the Pi and a region of interest (ROI) is extracted from the middle of the image to cut out unnecessary background noise. The pixels RGB values are then compared to a preset library of standard colors value ranges: red, green, blue. Visual examples of the preset library can be seen in Figure 10:



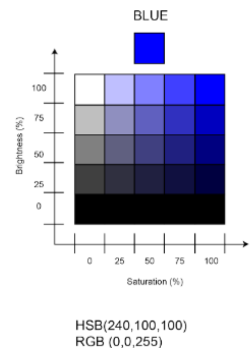




*Figure 10 – Preset Library*

The program compares each separate value to identify the color closest to the image pixels. This is done using a python NumPy formula to calculate the Euclidean distance between color vectors. This equation:

$$distance = \sqrt{(R_1 - R_2)^2 + (G_1 - G_2)^2 + (B_1 - B_2)^2}$$

will tell know how close two values are, the smaller the distance value, the more the colors are similar. The code will then look at which color had the smallest Euclidean distance, and that value is the one selected as the final color.

Once an initial color is identified, the second verification step is to cross-reference the results taken from the colorimeter. This helps to improve accuracy and adjusts the color if necessary due to potential outside interference in the color identification due to lighting or reflection inconsistencies.

The colorimeter detects color by using photodiodes with red, green, and blue filters that convert light into frequency signals. The sensor's output frequency is proportional to the intensity of the selected color, and it can be controlled using S2 and S3 pins to choose the desired filter. In our code, the ESP32 reads the frequency corresponding to each color and maps the values to a range of 0 to 255, representing RGB color intensities. The program then calculates the Euclidean distance, as mentioned before, between the detected color's RGB values and predefined values for our range of colors to identify the closest matching color. The detected color is then sent to the Raspberry Pi 4 to compare with the color detected from the CMOS camera.

The program then, utilizing Bluetooth, outputs an audible response using a text-to-speech conversion library which reads out the identified color along with a message of completion.

Moving onto the edge-detection algorithm, P.A.U.L implements a Sobel filter-based edge detection algorithm with a dual lens system. One lens system is set up in front of the Raspberry Pi Camera Module and the other is the lens system set as the glasses frame on the face. The first system in front of the camera is the one being adjusted during the auto-focusing process until focus is reached working in conjunction with the ultrasonic sensor. Then the overall distance that the camera module lenses moves is then transferred over to the frames to move those into focus as well.

The ultrasonic sensor is the initial step in the edge-detection algorithm, providing a distance measurement that guides the system in determining the starting point/ The ultrasonic sensor operates with two primary components: the trigger and the echo. To program the sensor, we configured the trigger pin to emit the pulse and used the echo pin to measure the duration of the returned pulse. In the code, we calculate the distance in millimeters, which provides a precise starting point for the autofocusing process.

Continuing this process an initial image is captured from the Raspberry Pi Camera Module in real-time. The image is then converted to be in grayscale as grayscale processing enhances and emphasizes intensity variations which we need for determining the sharpness of the image. Working without color also helps reduce computational overhead which is important to manage with our project given our limited resources and time limit goals. Next, we use a Sobel filter calculation which calculates the gradient magnitude of the image by applying convolution operations in both the horizontal and vertical directions. The following gradient equations are used:

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} * I$$

$$G_y = \begin{bmatrix} -1 & 0 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} * I$$

where we represent the greyscale image, and the final magnitude is computed with:

$$G = \sqrt{G_x^2 + G_y^2}$$

The final magnitude is used as a "score" as this algorithm calculates the total edge intensity across the image and we can use this as a sharpness metric. A higher edge intensity corresponds to sharper focus, and a lower value indicates blurriness. These "in-focus" values do vary by lenses, lighting, camera type, etc. so they required some calibration.

We've found through testing in the senior design lab standard lighting, that our acceptable range of "in-focus" goes for values larger than 300. Values appear a bit fuzzy or out of focus.

*Figure 11 – Focused Image*

The image in Figure 11 above received a score of 311 which is classified as in focus.

The program then determines if initially the image is in focus or not. If the focus metric score is within a calibrated acceptable range, the image is determined to be in focus and the same text-to-speech library is utilized to read out a message of completion. If focus is determined not to be sufficient, this triggers the program to request data from the ESP32 from the ultrasonic sensor to get distance reading. This distance is then used to calculate an estimated initial correction on the lenses. It'll adjust the camera lens system by moving the motors clockwise or initially away from the face. The Sobel filter calculation will be once again run and another updated image taken. If the score is lower, the lenses will then move in the opposite counterclockwise direction for double that distance as we are assuming it needed to be corrected in the other direction. Then moving in smaller increments, the Pi will continue this same process moving in one direction, checking focus and then moving in the other direction, if necessary, until the focus score is determined to be within acceptable range.

The program will then read out a message of completion. If the Pi repeats this process over a time of 2 minutes, a message of failure will be read out and prompt the user to press the button to reset the lenses and then restart the process. This added time limit ensures that the frames are still usable and reliable given outside interference in environments where focus adjustments are difficult.

## VI. Conclusion

P.A.U.L. has been designed with the aforementioned specifications to create a product that enhances the lives of those with minor visual impairments by assisting them with focusing power and identifying colors. Throughout the development process, we encountered several challenges such as the jamming of our motors. However, we managed to tackle each complication. The final design reflects our commitment to overcoming these challenges and obstacles and delivering effective solutions. The success of this project would not have been possible without the collaborative effort of the team.

## VII. Acknowledgement

We would like to thank Dr. Amezcua, Dr. Das, Dr. Li Kam Wa, Dr. Chan, Dr. Lei Wei, Dr. Delfyett, Dr. Weeks, and Dr. Kar for their guidance, support and expertise throughout our senior design project. Their mentorship has been crucial in helping us navigate the challenges of the design process. Additionally, we would like to thank the College of Engineering and Computer Science and CREOL for providing us the workspace and resources to develop our project to its full extent. We would also like to thank the professors who agreed to review our project, offering their insights and constructive feedback. Finally, we are grateful for the supportive and collaborative environment created by our departments, which allowed us to grow both individually and as a team throughout this project. This project could not have been possible without the help of everyone mentioned.

## VIII. Biography

Julie Gallagher is currently a senior at the University of Central Florida. Aiming to expand her expertise and knowledge in the field, while utilizing engineering skills and tools developed over the years to drive innovative solutions. Driven by a strong desire to address challenges and provide solutions for those in need.

Genaro Morales Jr is currently a senior at the University of Central Florida. He will be graduating from the University of Central Florida with a Bachelor's of Photonic Science and Engineering in May of 2025. He is currently working as a Research Assistant in the Microstructured Fibers and Devices research group under Dr. Rodrigo Amezcua Correa. After graduation, he plans on working at Relativity Networks as a Photonic Component Engineer.

Midkhat Satdanov is currently a senior at the University of Central Florida. He is hoping to pursue biomedical photonics and engage in bettering the lives of others through engineering.

Kayla Stanley is currently a senior at the University of Central Florida. She will be graduating with a degree in Computer Engineering and is planning on pursuing additional education to pursue a career working with the research, development, and maintenance of medical devices.

## IX. References

[1] "Blindness and vision impairment." World Health Organization. https://www.who.int/news-room/fact-sheets/detail/blindness-and-visual-impairment (accessed 11/25/2024, 2024).

[2] "Myopia (Nearsightedness)." Cleveland Clinic. https://my.clevelandclinic.org/health/diseases/8579-myopia-nearsightedness (accessed 3/20/2025, 2025).

[3] "Galileo's telescope." Institute and Museum of the History of Science. https://brunelleschi.imss.fi.it/esplora/cannocchiale/dswmedia/esplora/eesplora2.html (accessed 3/22/2025.

[4] T. V. d. Vall, "RGB Primary Colors," ed: Dutch Renaissance Press, 2013.