SMART Garage Parking Aid

Jennifer Castillo, Jorge Dardon, Daniela Otero, Nicholas Zollo

School of Electrical Engineering and Computer Science, University of Central Florida, Orlando, Florida, 32816-2450

Abstract — **There is no parking aid system on the market that is made for home use and offers useful convenience features for users. Using laser triangulation, image processing techniques and the Raspberry Pi, we have created a system that gives a driver visual feedback as they are parking their vehicle, to facilitate this process. With both lights that change color depending on distance as well as a graphical display that shows actual distance and configurable settings, the SMART Garage Parking Aid aims to bring an easier experience when parking at any home garage. This paper will focus on the electronic and physical components and mechanisms that make this device possible.**

Index Terms **— Diode lasers, distance measurement, engineering students, graphical user interfaces, image analysis**

I. Introduction

Many devices used today are considered "smart" devices, from a smartphone to smart appliances like the Smart Fridge. More and more people are using "smart" devices as a means of making their everyday lives just a bit easier, and our Parking system will benefit users in the same way.

Our SMART Garage Parking Aid (SMART GPA) will be a device that will give the user real-time, accurate feedback on their distance from their system as they pull into their home garage. Using this feedback, the drivers can judge their distance and will be able to pull into their driveway more safely and without stress.

The Smart GPA will also, like its name implies, include extra features and customization that will allow the user to personalize it to their liking.

With all the add-ons and features we added to our garage parking aid, we came up with the name Smart GPA, and we believe that it is different than any at home garage parking aid available to the market today.

II. Goals and Objectives

The goal for this project is for it to be a self-contained, portable system that can be attached to the wall of any typical home garage where a vehicle can be parked so the device can detect the position of the car to avoid any possible collision with the wall. This device will also offer additional features such as an interactive display with a GUI (Graphical User Interface) and lighting components for extra back up lighting.

The fundamental function of the garage parking aid is to alert a driver parking either forward or in reverse in a parking space if they are within near proximity from the wall. This device will be attached to the wall with all its components compartmentalized in an enclosure. A visual light LED display would alert the driver when it is okay to continue driving near the wall and then the color or pattern of light would change significantly enough to alert the driver of high-risk proximity.

III. System Concept

A. Distance Measurement

The main function of the Smart GPA is visual feedback to the user based on the constant distance measurements that are performed by the system. For the distance measurement system, we decided to implement a basic laser triangulation system, using a laser at an angle from a camera and calculating the distance the laser dot is from the camera from its position in the camera's image [1].

The extended theory and calculations for the distance measurement system is included in section V of this document.

B. Visual Feedback

Visual feedback for the user will be provided in two main ways. For one, a number of lights will be on the surface of the Smart GPA, and will turn on and change colors as the user pulls closer to the system. The distance measurement will control how the lights turn on and change color, and the properties and behavior of the lights will be customizable.

The other main method of feedback in the Smart GPA is the touch display which will show the current measured distance in large text when it detects a car pulling into the garage. Together, these two methods will provide both broad and fine data to allow the user to judge distance either at a glance or with a closer look.

IV. Hardware System Components

The Smart GPA has certain requirements and specifications, detailed in Table I below.

TABLE I
LIST OF REQUIREMENTS & SPECIFICATIONS

Parameter	Specification
Operating Range	1 to 10 feet
Measurement Rate	≥ 20 Hz
Measurement Accuracy	Error ≤ 10 cm
Output Power of Laser	≤ 5 mW
Output Power of LEDs	≤ 15 W
Input Voltage	≤ 12 V
Enclosure Size	1.5 m x 0.2 m x 0.5 m

In order to satisfy these physical requirements, we needed hardware that can perform at these specifications.

A. Microcontroller

The first component decided upon for the Smart GPA was the microcontroller used to control all of the components and handle their inputs and outputs. Our selected choice for this project was the Raspberry pi 3 Model B. The raspberry pi is more functional than a normal microcontroller, being more like a laptop or a desktop computer. It features a 4-core 1.2 GHz CPU, 1 GB RAM, USB communication between devices, 40 GPIO pins, and has native compatibility with external cameras and LCD displays. Our microcontroller's main purpose is running intensive image processing algorithms and controlling the major components like the camera and lights.

B. RGB LEDs

For the RGB LEDs we wanted to have then be powered at around 5 Volts as well as be bright enough for the driver to see them and use them for their visual aid. We chose to work with Addressable LED Lights to be able to individually code them to show specific colors and patterns [2]. We decided to go with the Addressable LED strip version WS281B because it suited our needs perfectly. The Addressable LED lights have 3 wires for connectivity, one connects to ground, another to power the LED strips and the last for data input from the Raspberry Pi to program the variations of each individual LED light.

C. Laser Diode

For the laser used for the SMART GPA, we wanted the laser dot to be clearly visible and distinguishable in the camera capture, but also fulfill the safety requirements for lasers so that it would not have a large chance of injuring a user. For this purpose we decided that the output power of the laser should be 5 mW, which is around the maximum power of a Class 3R laser, which is the highest class of laser that does not pose an injury risk to the skin or to the eye for momentary accidental exposures.

For our laser diode we decided on the HiLetGo red dot laser diode, which operates at 5 V with an output power of 5 mW and a central wavelength of 650 nm, satisfying all of our requirements. In addition, the laser diodes could be ordered in bulk, which allowed us to use multiple for testing purposes.

D. Touch Screen Display

For the GUI to be operational, we chose a portable touch screen display, namely the Raspberry Pi 7" Touchscreen display, or piDisplay. First we installed the piDisplay and connected it to the Raspberry Pi. The piDisplay has a native resolution of 800x480 pixels. The viewable screen size is 1.55mm x 86mm. Because of this and the fact that most of the GUI components will have to be visible from a far distance. The Raspberry Pi has a special port called the DSI port on the PCB that is responsible for handling the signal between the two components.

The piDisplay has its own printed circuit board that is responsible for transferring power from the pi and producing a visual display. It also has a physical touch screen module that must be connected to the printed circuit board in order for the display to have touch screen capability. This will ensure that the Raspberry Pi and the display are within range of each other.

Next, a ribbon cable that has laminated wire for all of the individual port connectors must be hooked up between the raspberry pi's DSI port and the piDisplay's connector port.

E. Power Supply

To power our whole device, a power supply was needed that would provide enough DC power to the device at maximum peak performance. Our peak performance load was calculated to be 30W. One of our specifications was that the input voltage of the overall device be less than 12VDC. Since this was a device intended for home use the options included powering the device by a general home use power receptacle, that outputs normally at 180 VA, which is more than enough power needed to power our device. To comply with our input voltage specification, this would require designing an adapter that would step down the voltage. Another option was powering the device with a rechargeable battery that would output at least 5VDC. The rechargeable battery option seemed like a better idea because it would allow the device to be more portable and convenient to the user.

The rechargeable battery selected was an EXP 635 6V 4.5 AH. Since the battery output was 6V, a step down voltage regulator was needed to safely power components at a steady 5VDC.

Ultimately, a PHEVOS 5V 12A 60W universal Switching Power Supply was used in the final prototype of the device. This power supply eliminated the need for a voltage regulator integrated into the PCB, since the power supply outputted a regulated 5V required by all components in the device.

F. Camera

The creators of the Raspberry Pi developed a camera suited for the raspberry pi board called PiCamera. The PiCamera is a low-power device that supports video streaming and image capture, that can be powered directly by the Raspberry Pi microcontroller. The pi camera can be controlled through python code via a library called PiCamera. It can be powered simultaneously with the PiDisplay, and still have sufficient current for GPIO output. For our project we will be deploying a video stream and using real time image processing on the frames of the video captured through the camera lens.

V. LASER TRIANGULATION SYSTEM

A. Range of Measurement

In order for the distance measurement to function at all, the laser light must always be reflected from the car pulling towards the Smart GPA at a location which is within the field of view of our camera [3]. The Raspberry Pi Camera Module V2 has a listed horizontal FOV of 62.2 degrees. Using trigonometric calculations shown in Figure 17 below, we can find the lateral distance W that the laser dot must be away from the camera to be seen for a given longitudinal distance L between the Smart GPA and the incoming car.

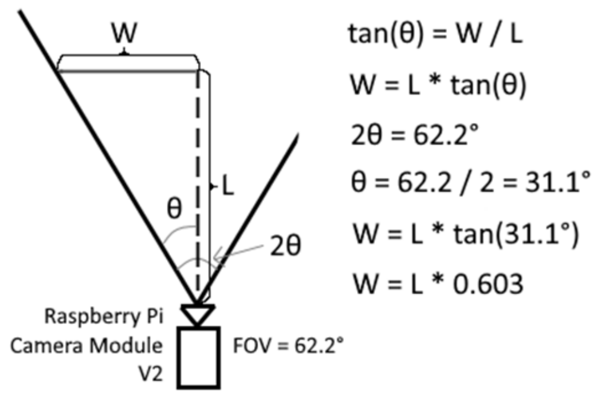


Fig. 1. Calculation of relationship between maximum visible width of an object at a certain length and the camera's field of view.

The relationship calculated in Fig. 1 is vital for designing our system layout. Using the equation, we can calculate that the laser dot will be visible if it is at most a lateral distance of 0.603 feet, or about 7.2 inches, from our camera at our closest operating distance of 1 foot. At our maximum operating distance of 10 feet, the laser dot must be a lateral distance of at most 6.03 feet from our camera.

To maximize the accuracy of our measurement, it is best to use as much of the horizontal FOV of our camera as possible. Therefore, we will create a triangulation system layout where the dot is far to the right for our closest operating range, and far to the left for our farthest operating range. To give us some room for error and still see the dot, the lateral distance for these key ranges will be set to 80% of the maximum that would be visible, so 0.482 feet to the right for the minimum range of 1 foot and 4.82 feet to the left for the maximum range of 10 feet. Fig. 2 on the next page illustrates the trigonometric calculations for the laser position and angle needed to create this system.

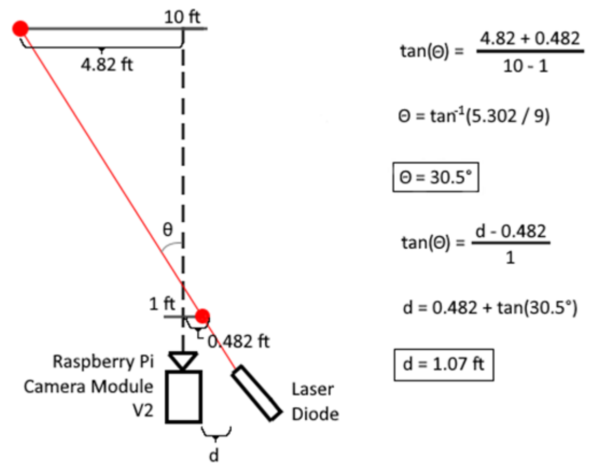


Fig. 2. Calculation of upper limit of the system parameters of the distance and angle between laser diode and camera.

These calculated values for the distance and angle between the laser diode and the camera are an upper limit for how they can be set. In our actual design, we used values slightly below the upper limit, setting our camera and laser diode 9 inches apart with an adjustable angle which is usually set to around 23 degrees. A special holder was made for our camera and laser diode to ensure the distance and angle was kept stable, pictured in Fig. 3 below.

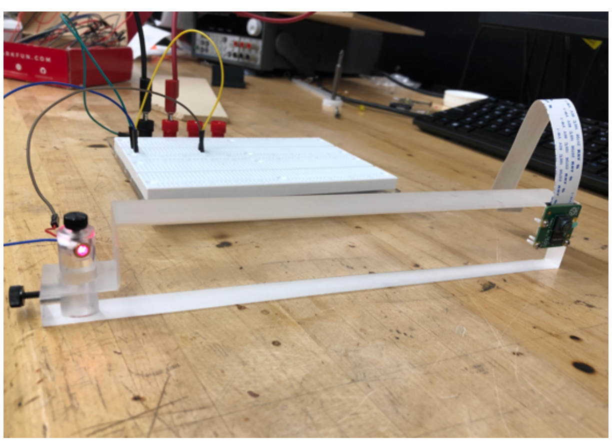


Fig. 3. Apparatus used to hold camera and laser diode still, with adjustable angle for the laser diode

B. Distance Calculation

For our laser triangulation distance sensor system, the main input that our distance calculation receives is the horizontal position of the laser dot in the image. Finding the laser dot quickly and efficiently is the main programming difficulty of the design, but once that position is found, we must convert it to the distance, which turns out to be a rather involved calculation. First, we must convert the pixel we read to the angular displacement of the laser dot from the center axis of the image. The trigonometric calculations leading to this conversion are listed in Fig. 4 below, with W being the number of pixels in the horizontal axis of the image, HFOV being the horizontal angular field of view of the camera, x being the horizontal position of the laser dot, and ϕ represents the angular displacement of pixel x from the image axis.

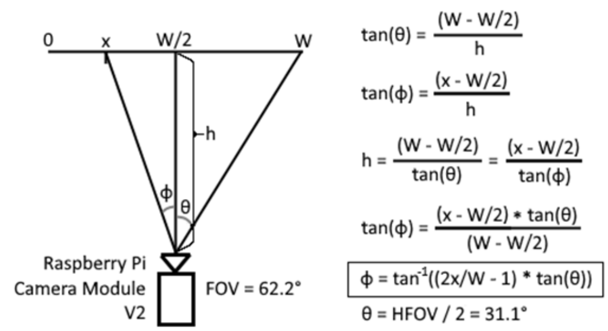


Fig. 4. Calculation of angular position of object relative to camera from its pixel position on camera image.

The relationship calculated above can be used to convert the pixel measurement to the angular position of the laser dot. Each angular position of the laser dot is uniquely tied to one distance of the object from the camera. However, we also need to calculate this relationship between the distance of the object from the camera, z, and the angular position of the laser dot. This relationship is specifically dependent on the position and angle of the laser diode relative to the camera, so it will change if we use a different system layout than the one described in Fig. 2. Fig. 5 on the next page shows the calculation of the distance of the object using the angular position of the laser dot ϕ as well as the distance d and angle Θ between the laser diode and camera.

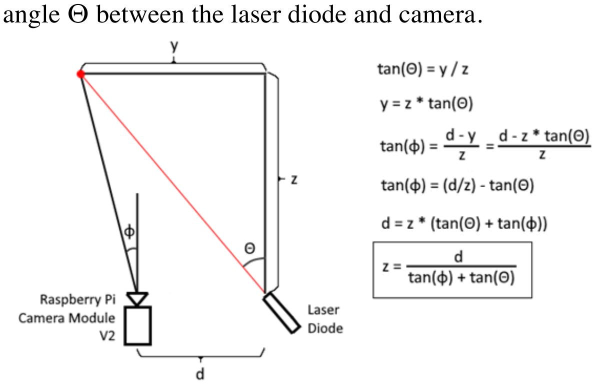


Fig. 5. Calculation of distance of object from camera from its angular position relative to camera.

By combining the equations in Fig. 4 and Fig. 5, we can create a general equation for our laser triangulation system that holds true for any system layout and camera HFOV or image resolution. This general equation is marked as Eq. 1. Note that the entire denominator of Eq. 1 is unitless, so the unit of z will be equal to the unit of the value of d substituted into the equation. Note also that for a given system layout and image format, that d, W, HFOV, and Θ are all constants. This means that the only variable for Eq. 3 is x, the horizontal position of the laser dot on the image taken by the camera.

$$tan(\varphi) = (2x\ /\ W\ -\ 1) * tan(HFOV/\ 2)$$

$$z = \frac{d}{\left(\frac{2x}{W}-1\right)*\tan\left(\frac{HFOV}{2}\right)+\tan(\theta)} \quad (1)$$

VI. Hardware Design

Our hardware design consisted of power distribution and component communication. All of our components required a 5V input voltage to operate efficiently, but the power requirements and consumption vary between components. Thus, it is essential to supply regulated power to all components safely. Component communication was also an essential part of the hardware design and will be detailed below. The block diagram below outlines our final PCB design power and communication flowchart.

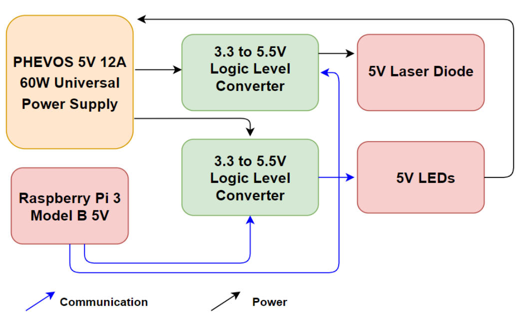


Fig. 6. Block diagram of final PCB design and system for communication.

A. Power Distribution

The input power supply that met power specifications and allowed the device to power efficiently was a PHEVOS 5V 12A 60W Universal Switching Power Supply. This was ultimately the power supply used in the final prototype. This power supply outputted a regulated 5V to all components and enough power for the device to steadily work at maximum performance [4].

The initial idea for the power supply was to use a 6V rechargeable battery. This would have required a step down voltage regulator that would have converted the output 6V of the battery to 5V, and handle a current load of at least 6A to safely and efficiently power the device. The voltage regulator chosen was the LM25085MY/NOPB by Texas Instruments, a PFET switching regulator controller. This voltage regulator satisfied power requirements for converting 6V to 5V in online simulations provided by TI Webench.

B. Communication

Communication between components was essential to be able to synchronize and control when components should turn on and off. Our components such as the LEDs and laser diodes needed 5V to power on, but the raspberry pi output GPIO pin logic outputted only 3.3V maximum. Therefore logic level converters were integrated to allow our devices to power on and be controlled by our MCU.

Our other components such as the PiCamera and piDisplay did not need logic level converters as their input power is fed directly from the Raspberry Pi. Because these components are fed from the Raspberry Pi, the raspberry pi's consumption was increased and taken into account when estimating loads to select the appropriate power supply.

VII. Software Design

In this section we will cover the software and programming tools that we used to make the SMART GPA.

A. Operating System /Programming Language

The Raspberry Pi has its own unique operating system specifically tailored for its user base; it is a flavor of Linux based on the Debian OS called Raspbian. For our project the Raspbian is important because it features many important packages and commands that are unavailable in other operating systems, many of which are often built in commands for the linux terminal.

Python is the clear winner for this project because it has native support to program the GPIO pins and control the camera and display. The python language includes support for many of the programming tools, libraries, and frameworks for different components of the project we will be using, such as QT, PiCamera, among others. These libraries and frameworks are essential for creating a quick and responsive GUI and controlling the camera and lights.

B. Image Processing for Distance Detection Algorithm

The goal of the algorithm is to isolate the pixels in each frame that match the laser beam. First for the algorithm to work, we ensure that the camera has a low shutter speed (found to be beneficial during testing). This limits the amount of light entering and prevents the laser beam from deviating from its original color. In the current algorithm, with a native PiCamera function called capture_continuous, we can start the camera in video mode and work with each frame individually [5]. This can be done in an endless loop until stopped.

For each frame we convert its representation in Python from a regular 2-dimensional array to a numpy array. After converting the image to a numpy array and flattening the array, we use numpy.where() to separate the pixels based on whether any of its pixel values (R, G, or B) are greater than a threshold AND if the R values are greater than the sum of G and B values above a threshold. Using both of these conditions, we can reduce the amount of false positives and improve the general accuracy of detection. Fig. 7 below is an illustration of the current pixel isolation algorithm.

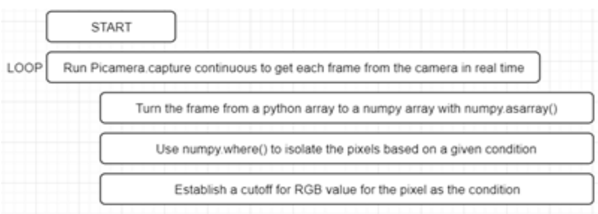


Fig. 7. Illustration of current pixel isolation algorithm.

C. Graphical User Interface Design

Fig. 8. Idle mode GUI window created with PyQt framework.

Fig. 8 above shows a window created on the PyQt5 framework for Python. This is the Idle Mode main menu, which will be explained in section VIII. It is an event-driven application that detects mouse clicks, changes, and responds with a programmed action. Each of the buttons creates a new dialog window that has its own style and operation and features.

VIII. Subsystems

In the SMART GPA we have three main different subsystems which include the distance sensor, the GUI display, and the light display.

A. Distance Sensor System

The distance sensor system is made up of the raspberry pi, camera, and laser. While the camera continuously captures frames that the laser appears in, the raspberry pi processes them. A moving object in a forward or backward direction will have the position of the dot change horizontally on the image.

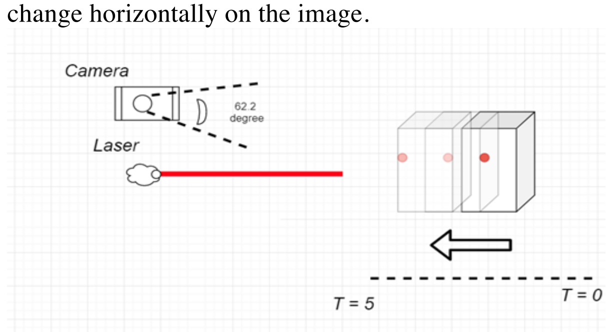


Fig. 9. Illustration of basic function and use case of distance sensor subsystem.

B. GUI Display System

The GUI will communicate with the camera as well as the light display. It will have two different modes the user can see, called the idle mode or operation mode.

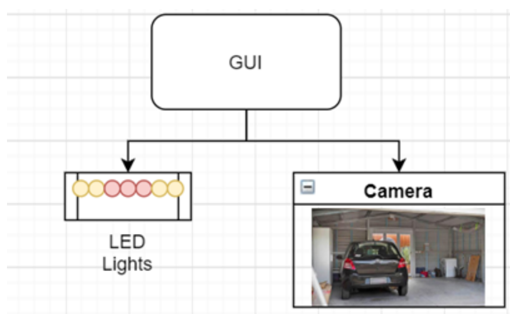


Fig. 10. Basic block diagram of control system of the GUI.

The idle mode is where the user can access different settings, either account settings or general settings. In account settings the user can enable and disable the option to login to the GUI as well as change the password or create a new account. In general settings, the user can change the date and time, enable or disable the light display, and customize the system to their liking. Idle mode will be enabled automatically when the distance measurement remains the same for a period of time, signifying that there is no moving object in the range of the Smart GPA.

In operation mode, the GUI will connect to the camera and distance sensor and display the current distance measurement from the distance sensor subsystem. The distance measurement will be displayed in large text so as to be readable to the user as they pull towards the Smart GPA. Operation mode will be enabled automatically when the distance measurement is changing, signifying that an object is within the measurement range of the Smart GPA.

C. Light Display System

The light display system will be made out of LED lights attached to the enclosure. They will light up a certain color when the distance sensor senses a car within distance.

The light display will light up green when the incoming car is around 10 feet away from the device. When the distance sensor senses the vehicle to be a bit closer, at around 5 feet, the lights will change color from green to yellow and start blinking, to alert the driver to start driving slower.

When the distance sensor senses the vehicle to be super close, around 2 feet, the lights will change from yellow to red and will start blinking more rapidly to alert the driver to stop driving and park their car.

The colors and settings described above are the default, but are customizable with our touch display.

IX. TESTING

A. Lab Prototype Testing

In order to test the initial design of the pixel isolation algorithm, we started with an outline of its expected behavior. After designing the algorithm, we set up the environment that would be used. We had already tested the laser diode on the breadboard and confirmed it working, so we clamped the laser to have it firing at a desired angle and at the same spot. For testing purposes since our power system was not yet set up, we used the senior design lab's power supply to turn on the laser diode. We also clamped the camera so it would not move.

Once the laser and the camera were set up, we ran the python script we had developed to start capturing frames as well as turned on the laser. We took a white box and moved it back and forth in front of the camera, noting that the laser dot on the box would move horizontally as the box moved. Our output showed that we could get an accurate distance measurement for our specifications, even with this crude system.

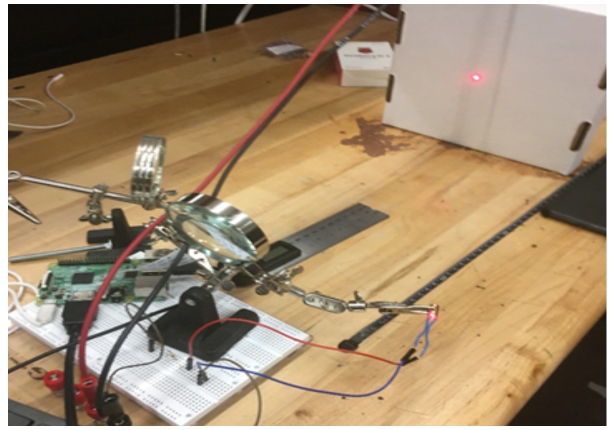


Fig. 11. Basic testing system used for initial tests.

B. Final Testing

We ran tests for verifying that the system works together as a whole as well as for individual components and subsystems. Before the prototype was constructed, we had verified each of the components like the camera, pi display, lights, etc. would work alone. Then, we integrated the camera, Raspberry Pi and laser to form the distance sensor. We ran unique tests on the distance sensor alone, testing at various distances like a few centimeters from the camera, almost 15 feet away, and anything in between.

We tested multiple surfaces starting with the white box surface we were using in the lab followed by various items we had available like dark transparent plastic, metallic non reflective surfaces, and each showed few problems during measurement. During this phase we realized that rotating the camera and making the laser go in a straight line towards our object is a superior design as it allows us to measure in our entire range without worrying about the width of an object.

We tested the GUI's individual windows, making sure each performed its prescribed functionality and that spawning other windows would not interfere with the current window. We tested its event driven architecture and functionality, making sure that changing an internal value that was a property or linked to an event would fire the appropriate function.

C. Problems

One problem with the algorithm is that false positives can arise. When we are detecting for pixel values we may encounter pixels that are within the threshold of the separating condition to identify the laser dot. False positives during testing have shown up often when testing new lighting conditions, and we have to adjust the threshold to get rid of them

Another problem encountered during testing is that light can be scattered on reflective surfaces and the laser beam might not show up on the camera. Dark surfaces also have the problem of absorbing the laser beam. We can compensate for this by increasing the shutter speed of the camera and allow more light to enter the lens for a longer period of time, but this may create more false positives. A balance is required. If ambient lighting is too high, it may be very difficult avoid false positives while still detecting the laser dot on difficult surfaces.

X. CONCLUSION

The Smart GPA has been an 8 month project. We spent the first few weeks choosing the right project and the first few months selecting the appropriate components and writing 120 page documentation detailing every part of the system, as well as light assembly and testing. The next few months were spent rigorously constructing and assembling the subsystems, and finally testing, debugging, fixing issues.

Because of the Covid19 pandemic currently going on, two of our group members are out and our project was cut a bit short on the full expectations such as a lack of a PCB for power distribution as well as a wall-mountable enclosure.

However, we were able to accomplish the vast majority of our goals with the Smart GPA. If we had more time, we would likely expand the customization options, add more convenience features, and create a wall-mountable enclosure for the system.

ACKNOWLEDGEMENT

We would like to acknowledge Dr. Lei Wei at the University of Central Florida for his guidance and direction during the earlier stages of the project.

REFERENCES

[1] "What is Laser Triangulation?" MoviMED, n.d., www.movimed.com/knowledgebase/what-is-laser-triangulation/

[2] "What is an Addressable LED?" sim-ple-ton Says, 2 January 2016, www.simpletonsays.com/how-do-i/what-is-an-addressable-led/.

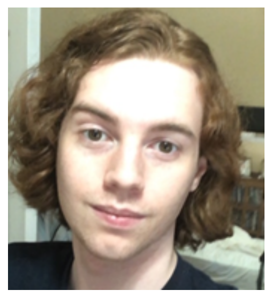
[3] "An Introduction to Laser Triangulation Sensors." AZO Sensors, 28 August 2014, www.azosensors.com/article.aspx?ArticleID=523.

[4] "Power Supply." Raspberry Pi Documentation, Raspberry Pi, n.d., www.raspberrypi.org/documentation/hardware/raspberrypi/power/README.md.

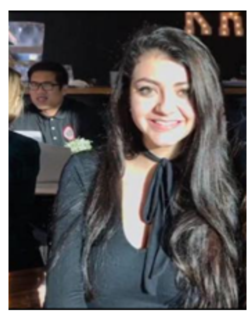
[5] "Camera Module." Raspberry Pi Documentation, Raspberry Pi, n.d., www.raspberrypi.org/documentation/hardware/camera/.

BIOGRAPHIES

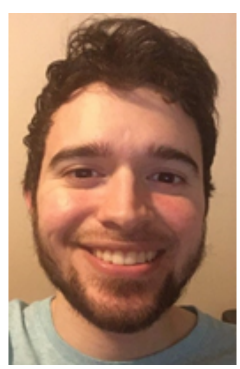
Jennifer Castillo is studying Electrical Engineering at UCF with a focus on Power & Renewable Energy. Post graduation plans include taking the FE exam and being an electrical engineering building design consultant in the current firm employed in.



Nicholas Zollo is a Photonic Science and Engineering student at UCF, with side interests in Mathematics and Computer Science.



Daniela Otero is studying Computer Engineering at UCF with a side interest in Math. After Graduation, will pursue a Masters degree in Industrial Engineering as well as look into taking the FE exam and being a software engineer at the company currently interning for.



Jorge Dardon is a 23 year old computer engineering senior at UCF specializing in full stack development. After graduation he plans to find a job in programming or software engineering.