Spotter 1000

Jay Davila, Wandy Pierre, Richard Macias

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

Abstract **—** At the University of Central Florida, business students have the option of classes that can enroll 1000 students. The problem with these classes is that there are no classes at the university to accommodate 1000 students. This is an issue because if a student shows up to class when there are no seats available, they are either left standing or going to a computer to watch the lecture capture. The Spotter1000 will detect open seats using a computer vision camera and stream information about class room seat availability to web and mobile interfaces available to students.

Index Terms — Computer vision, servomotors, databases, servers, image color analysis, object detection**.**

I. INTRODUCTION

With so many students in the college of business, finance, and accounting, the University of Central Florida cannot accommodate enough teachers per classroom for the classes these students need to take. Therefore there is a system in place which allows students to have the option of either showing up to class or watching the class from their computer from wherever they want. These are called lecture capture. Although this is a great way of solving the problem of not having enough teachers for students, it presents a new problem. The problem it presents is the student who shows up to class but does not have a seat because all the seats are taken. This problem causes students to either stand during the class or forces them to watch the lecture capture for the class that day. If a student is going to watch a lecture capture, they might as well have stayed home or not rushed to class. This problem can be solved by providing an accurate representation of the how many seats are available in the class at any given time.

The initiative behind Spotter1000 is to give the user the ability to see the availability of the seats in the classroom before being physically present there. It also allows students to see what seats are available once they are in the classroom. Since there are many students in these lecture halls, it is difficult to see what seats are not taken. There are over one hundred fifty million people in the United States who own smartphones. Our plan is to give them access to the wealth of information gathered with the Spotter1000.

The Spotter1000 will be a user-friendly mobile and web application that provides users with information regarding their classroom destinations. In the first phase of the application the user will be able to view the different buildings and classroom numbers capacities of available seats. The proceeding phases will include percentage of available seats in classrooms and extend the same functionality to parking lots and garages for available parking spaces.

The Spotter1000 will use one camera with a narrow angle lens strategically mounted on a motor that sweeps the room to provide a clear visibility of the individual seats in several distinguished classrooms.
Each seat will contain a rectangular light colored rectangle that represents its availability. The sensor, a camera developed by Carnegie Mellon University, will provide a live feed to a remote server running an application to determine the spaces availability. The data will be stored in a database which can be accessed by the web and mobile user interface.

II. SPECIFICATIONS

In order for the Spotter1000 to be considered successful, the following specifications must be met:

- The Camera must have a field of view no greater than 75 degrees, a line of sight above 6 feet.
- The image analysis software must be able to determine the availability of 250 seats.
- The data transfer speeds of the wireless transceiver must be at least 100Mbps.
- The database must have a MySQL Default port of 3306 available and the tables must be at least 2.09kB per table.
- The servo motors must be able to turn 180 degrees in 2 seconds.
- The Mobile application must be clean, simple, and user friendly.
- The refresh rate of the applications must be at most 1 minute.

III. MAJOR SUBSECTIONS

The Spotter1000 design consists of the research done by each member of the group and deciding collectively on a parts list as well as design summaries broken down by each subsystem. The system has been divided into three equal parts in order for the total work done by each member to be evenly distributed. These three parts consist of multiple elements which are required for the spotter1000's functionality.

The first subsystem is the camera subsystem. This subsystems purpose is to detect any empty seats in a classroom. It consists of the cmucam5 open source camera developed by Carnegie Mellon University, and a lens of specific field of view for the camera.

The camera then passes the information on to the second sub system which is the microcontroller. The second subsystem is the heart of the project. The microcontroller controls the servos which set the tilt of the camera and which side of the classroom is being analyzed, runs the cameras algorithm for detection of seats and passes the location of the open seats to the server using a wireless transceiver.

The last subsystem is the server subsystem. This subsystem consists of a server to store and process the data that is sent in, a database to update the end clients, and a web application to provide the information to the clients. The subsystem processes the information sent to the server to determine what seats are still open, updates the database, then sends the information back to the server to output to the web and mobile application.

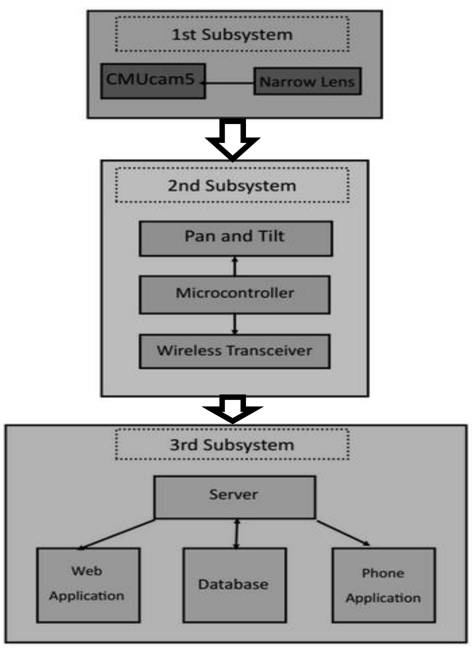

FIGURE 1: THE DIFFERENT SUBSYSTEMS

IV. DESIGN

A. Camera

The image sensor used is the Omnivision 0v9715. It has a display resolution of 1280x800 at 25fps. It can also run at 50fps using 640x400 resolution. The sensor is placed inside an open source packaged camera called the CMUcam5 developed by Carnegie Mellon University or simply the "Pixy Cam"[2]. This camera contains a NXP LPC4330 processor, which will do image processing, specifically a hue-based color filter to distinguish objects. There is a total of 264K bytes of RAM, which is needed to store frames needed for the image processing. The camera can be set to track up to seven distinguished color signatures. A newer firmware update allows it to take in almost twice as many signatures.

The NXP processor sends data via SPI, UART, or I2C communication. This data consists of the number of objects detected, their X and Y pixel coordinate locations, and the height and width of the objects. The data is sent using the Serial peripheral interface and share the data lines with the wireless transceiver, which is described later in the paper.

The max number of objects it can detect is well over one hundred. The information sent will be for 1 frame and update at 25 frames per second. The original camera came with a 75 degree horizontal field of view lens. This lens was undesirable according to the project specifications. The system needs as many pixels as possible. To help identify the available seats, a narrower field of view lens of 50 degrees was used. This means the pixels have less area to cover and an object can be defined with more pixels. The more pixels available to cover an object provided, the better the hit rates of the object identification.

The object detection of the open to closed seats is done with the hue based color filter on the camera. The camera used RGB format and would compare pixel intensities of the image. Similar pixel intensities of the color the camera is set to look for are considered one object. The system is set to follow unique colors students would not wear in class like bright pink or yellow. In the real application, the seats will be painted, but for demo purposes sticky notes are used to identify open seats. The sticky notes are significantly smaller than the actual seats, but help determine one seat apart from a seat in a nearby location. To help identify the sticky notes, several signatures of its color were taken at different lighting conditions. This creates a larger color signature to look for.

The Pixy Cam was a great choice because of its high resolution and digital output of information. As an extra feature, the Pixy Cam took care of storing the large sized images and the performed image processing needed to identify objects. This means that the microcontroller would not require a fast central processing unit or a large amount of memory. This allowed the purchase of an affordable microcontroller, that everyone in the group was familiar with using.

B. Microcontroller

The Microcontroller used in the Spotter 1000 is the T.I. MSP430 F5229. It processes at 25MHz, 8kb of SRAM, and 53 available GPIO.[1] It's able to communicate via SPI, UART, I2C, and produce pulse width modulation signals. The T.I. board is also low powered, only running 404 uA per MHz.

The microcontroller was set to receive data from the Pixy Camera. Information from the camera was parsed, so only the total number of objects are detected and their X and Y coordinates. The microcontroller will also parse the data of one frame and output it in the following order, "s" for start, to signal to the server the beginning of a frame. N number of objects to let the server know how many objects it should expect from the microcontroller. "x", and "y" values of the pixel location of the center of the object. It will then repeat the x and y locations for every object detected in the frame.

The T.I. MSP430 will send this data through a wireless transceiver module which is another T.I family module, the CC3100. It will be discussed further in the paper. Finally the controller is in charge of moving the motors in our pan and tilt system after the data is sent.

The MSP430 F5229 was chosen over other microcontrollers because the group has had experience working on these microcontrollers in the past. For the printed circuit board, using an older TI microprocessor like the MSP430 G2553 would have been more convenient to use due to its small amount of pins on the processor. Since there was an insufficient amount of RAM and ROM on the G2553 to run the software, the older versions of the MSP430 could not be used. The system required a sufficient amount of RAM to account for the camera sending large amounts of data, in which contained the X and Y pixel locations of the objects and varied frame from frame.

C. Wireless Transceiver

Texas instruments CC3100 Booster pack was chosen to give the MCU access to the World Wide Web. The CC3100 has on chip WLAN and TCP/IP stack that help cut the amount of work and time that would be required to implement these functions. Besides creating the internet of things, the CC3100 has built in security module with Embedded Crypto engine that protect the data and provide quick connection. The features that are most important for the project are the 2X20 pin stackable connectors which are connected to the pins from the msp430f5529LP, the on-board chip antenna with the option for U.FL based conducted testing which enable the device to receive and send the data, it is powered from on-Board LDO using USB or 3.3v from MCU Launchpad and for this project it will be powered with 3.3v from the MSP430f5529LP, it has 0.8 Mbit serial flash, 40MHz crystal.

D. Pan and Tilt System

The pan and tilt system was purchased through Charmed Labs. The kit uses two servo motors to control the tilt and panning of the camera for object tracking. The project needed to have manual control of these motors with the MSP430. Specifically the microcontroller uses Pulse Width Modulation signals to vary the positions of the camera. One servo motor of 180 degrees of movement is enough for the camera to capture the entire room since the camera will be placed in the front of the room. The Servo motor for the tilting movement is only needed to change a few degrees when the system initializes based on the height of the mount.

The Servo motors chosen are the GS-9018. They have a torque of 1.1kg-cm and a speed of .12 seconds to turn 60 degrees. These servo motors were purchased because they are provide enough torque to easily move our camera and are available an affordable price. We chose these motors over other variations of motors because our system does not require high accuracy movement or a high speed to move the camera from one half of the class to the other. Since the camera cannot capture the entire class room in one angle, the system will need to pan the camera to capture the rest of the seats. This means after sending the locations of the open seats, the camera halts the transmission of data, pans to the other side of the classroom, and then continues to acquire data.

E. Server

The Spotter1000 server is designed to perform a multitude of functions that are essential for the functionality of the system. These primary functions include the cloud computing subsystem which takes in the live data stream, extracts the information, and stores it in the database. As it stores the information in the database, it is converting that information to the proper format of X and Y pixel positions of the seats to compute. It also enables the end user to access the system from the phone or web application.

The daemon will run in the background constantly feeding the information obtained from the TI CC3100 shield to the database. Using PHP in conjunction with JQUERY and JSON, this information will be transmitted to the Internet making it available for the web based and mobile applications to access. The daemon will have the ability to remotely connect to the database by using one of the php files created which is called “update.” The daemon will also have access to read and to write data to the file. Although this primary implementation will have both the database and the web server on the same machine as the server, the system will be designed for the daemon to have remote access to the database.

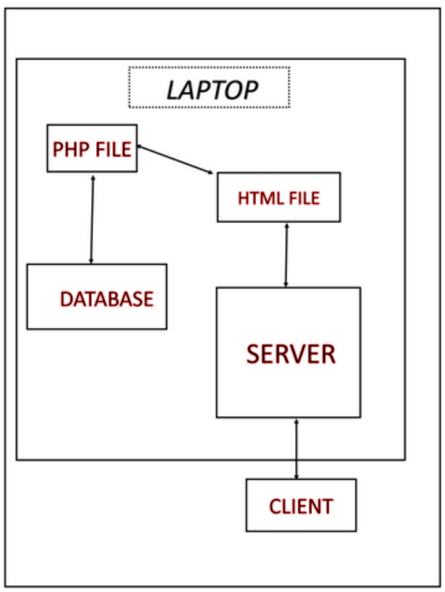

FIGURE 2: SERVER SUBSYTEM

F. Algorithms

The spotter algorithm relies on the data taken in from the camera. This data is in the form of horizontal and vertical positions on the frame of the picture and the number of objects in the frame. The way in which the camera sends the blocks of data is in order of size and not position. On top of that, during testing, slight variations were noticed in the seats’ positions every time the camera sends data which would cause the seats to be incorrectly sorted.

It is important for the seats to be sorted correctly because the information displayed on the web application is displayed by seat number. This issue was solved with the realization that the classrooms empty seats will be static. This allows for the seats to be hardcoded into the algorithm. Ideally the cameras will be fixed in one location, so once the positions of the empty seats in the classroom are recorded they will not change.

Once the x and y positions of the seats are recorded, the unique number is manually calculated and assigned to a corresponding seat number. Then all the seats are initialized to “closed”. This completes the hard coded portion of the algorithm.

The second part of the algorithm depends on the unique number function that was used for the hardcode. After receiving the data from the camera, (which is out of order) the x and y values of the seats location are plugged into the same unique number function used to produce a unique value for the seat number. These values that the function returns will then be compared to the hardcoded values.

Since there are slight variations in the x and y positions, a sort of tolerance or buffer must be accounted for. The values that are a “tolerance number” close to the hard code are the seats that are still available and therefore outputted as “open” to indicate that it is so.

There are two ways in which the comparisons can be done. The first way, after the x and y values are passed through the function, the number is saved into an array which will represent what seats are still open. This array will then be looped through as the hardcoded empty seat array is looped through and when a match is found, the index of the hardcoded empty seat array is taken note of and in that index of the open or close array, an open is recorded. This is brute force type of method and will cause the runtime to be longer.

The second way implements a hash function to make the runtime faster and more efficient. After the x and y values are passed through the function, they represent an index on the seat number array which contains the seat number that corresponds to that unique value. An output of open is then written into the “open close” array index that matches the number inside the unique value index. This may sound confusing so the figure below should demonstrate it more clearly.

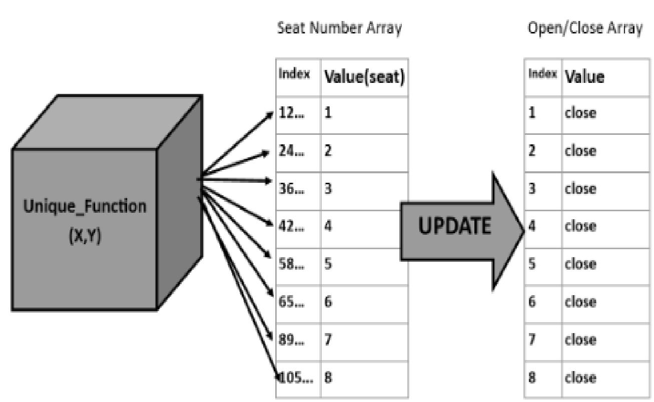

FIGURE 3: HASH ARRAY

All together there are four arrays. One of the arrays is the unique number array which maps the unique number to the seat number that we hard coded. Two of the arrays are the arrays that hold the X & Y values of the seat and the last array holds the "open or close" seat status of the seats.

G. *Machine Virtualization*

While the system is under development, the absence of a real server forces the team to come up with ways to simulate a real server by implementing a virtual machine atmosphere, and also make the software scalable, modifiable, portable and robust throughout the development and prototyping. The main computer running the virtual machine will be capable of emulating the desired server and exceed the recommended system requirements. The core machine used to run the virtual machine environment will be an existing TOSHIBA SATELLITE laptop with an Intel 2.4GHz processor and 4GB of DDR2 Memory. The Virtual Machine application selected is the VirtualBox. It is used widely in the industry and the guide's resources available for it are countless. It is also free of charge and it is backed by Oracle.

H. *Application*

The application will be implemented in both the world wide web and a mobile phone running the android operating system.

The first page of the web application is the home page. Once the user loads the spotter1000.com website they will see a list of buildings to choose from whose names correspond to the building names on a student's schedule. At the top of the page there is a menu that directs you to the project and the members of the project. To maintain consistency and a little bit of school pride to make the application user interface more user friendly, the color scheme is none other than black and gold which represent the University of Central Florida's main colors.

After clicking on a building, the next page is the list of class numbers in that building. The class number list is in a chart format that runs in an increasing order of class numbers from left to right. At the top of the page the section of the site in which the user is in is displayed.

Once a class number is selected, the list of open and closed seats is presented in a descending order beginning with the left most seat of the classroom. If a seat is available a green "open" is displayed next to that seat number. If the seat is not available, a red "close" is displayed instead. The color scheme once again is kept at a minimum number of colors in order to meet the simple design specification for the user interface.

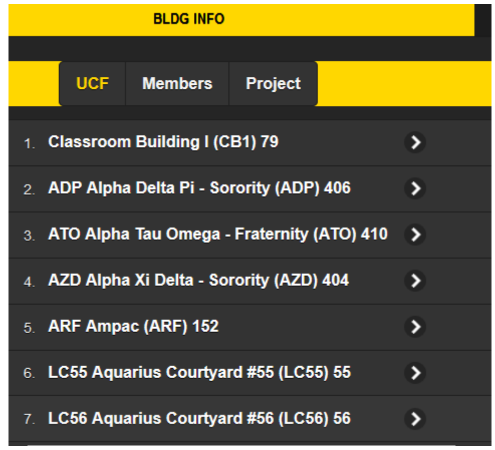

FIGURE 4: SCREEN SHOT OF THE WEB USER INTERFACE

The mobile application has a similar layout. The first page that loads when you open the application is the home page which matches the color scheme of the website. A UCF athletics logo is in the background to promote the University. The home page also displays an analog clock that represents the time of day. Once the user chooses the classroom button, the next page is loaded. The next page that is loaded is the webpage view. This makes the mobile application easier to use and it also conserves space on the mobile device.

FIGURE 5: MOBILE APPLICATION SCREENSHOT

I. Server Operating System and Other Software

To run the project on a server operating system, the Linux distribution of Ubuntu Server 14.04 LTS has been chosen. It is one of the most widely used platforms in the industry. In order for the virtual machine to have full network access and for the user to have more control, the networked device will be bridged.

The virtual machine will be used to run some of the main applications used in the spotter1000 system. As previously mentioned, MySQL is intended to be used as the systems database collecting data from the cloud computing operation subsystem and storing it in the local host.

In order to serve HTTP request on a local network, the latest version of Apache HTTP Server v2.4.9 will be installed and configured to the project's needs. The Apache HTTP Server is renowned and generally recognized as the world's most popular web server. It was originally designed for the UNIX environment but recently ported to windows and other network operating systems.

The web server provides a full range of features including DDL, CGI and virtual domains. The database will communicate with the Apache web server through various configured server side scripting languages for local communication and via TCP/IP with the daemon. The selected server side scripting languages used in the spotter1000 system are JQUERY and PHP and will serve the function of updating and generating the HTML formats displayed on the web and mobile applications for the end user.

J. Database

The database will be organized in a way to make all the buildings' name and the classes' number as PHP Clause, which is the script used to create, update, and destroy a classroom in the database. This is done to avoid updating or viewing the wrong classroom. In order to do that, The latest version of MySQL will be installed and configured on the virtual server machine, along with an installation of MySQL Workbench, a visual database design application that can be used to efficiently design, manage and document databases.

To be consistent with the naming, the primary database that holds the seats will be created and given the name Spotter. Every column will represent one particular classroom. Since the majority of the classrooms that will be tested are very similar, as mentioned earlier, the first row will be the building's name followed by the classroom's number.

K. Mounting the System

Ideally the system will be located in the front of the classroom, placed in a corner, and close to the ceiling. A pole was chosen to mount the system for two reasons. Hanging a camera may damage the walls of the classroom and it gives the system the ability to be portable. Using a pole, the Spotter1000 can be used to detect seats in other locations besides the classroom.

The pole has a base height of 6 feet and having an extended range up to around 11 feet. There is now an option to adjust the height for testing purposes and it gives the flexibility of using the system in classrooms with different number of seats. The reason the system needs to elevate is to obtain a field of view of all of the seats in the classroom without having one chair or student block out another.

It's made of aluminum and has a plywood base for both the electronics on top and mounting the pole at the ground level. Bolts are used to secure the pole and an inner pole with a smaller diameter of 1" is used to extend out of the base pole of 1 ½" diameter. The pole is decorated in the UCF colors.

L. PCB and Power design

A printed circuit board was created to distribute power and route the data signals to the components. 4PCB and Quality management system services were used to create the board and mount the components. The Serial Peripheral communication protocol pins on the MSP430 are shared to communicate with the camera and the wireless transceiver. A slave select is required for both external devices since they both share the same data lines. The pulse width modulation signal pins are connected from the MSP430 to the signal lines of the servo motors in order to control the pan and tilt manually. The PCB is a two layer board, having routing done on both sides. The bottom layer is used as ground and vias serve to as connections for ground on most of our components.

Besides the signal routing, the system needs to supply the camera with 6-10 volts. A 7.5 Volt power supply with 1.5 amps is used to power the entire system. The 7.5 Volts splits into the camera and into a 5 volt regulator with 1.5amp capability, specifically the UA7805CKTTR from Texas instruments. The 5 volts is used to power the MSP430 and the two Servo motors. Table () shows the max power consumption of the components and the power provided by our power supply. This table shows the power supply can support our system.

Component	Max Power consumption
MSP430	49.5mW
Pixy Camera	700mW
CC3100	726mW
2X Servo Motors	1.6W
Component	**Max power Supplied**
7.5V AC/DC power supply	11.25W

TABLE 1: POWER SPECIFICATIONS

V. TESTING

The first type of testing is testing the components of the system in a lab. First individually testing each component to verify they meet the correct specifications and requirements and then testing them together to verify that they are compatible. The three main components to be tested are the camera, the wireless transceiver, the server, the web application and the phone application. The camera was tested in the lab with both the microcontroller and the printed circuit board.

Color codes were placed at several distances from the camera to ensure it would detect the number of objects and the color signatures correctly. Other objects were placed in the same frame to test the accuracy of the object detection.

The wireless transceiver was also tested in the lab to ensure it could connect to the server and send data at a rate which met specifications. This was done by connecting and sending ASCII values manually through a terminal. The web application was tested by updating information through the server and the phone application was tested by pulling information from the web application.

Different test scenarios were done to see the accuracy of the open seat detection. Group members would sit in different seats and observe the seating chart update on their mobile phones. The time required to update and the consistency of the chart when no one moved from their seats was observed.

Testing for blind spots was also done to make sure the camera was mounted at the right height and tilted in the correct position, in order for it to see most of the seats in the classroom. The camera was tested at six feet and blind spots were observed in some of the seat locations. After trying several different heights, it was determined that nine to ten feet would be the ideal height for the least amount of blind spots.

The camera was also placed in corners of the classroom. It was not able to identify the seats in the far back. The hit rate of empty seats decreased when compared to the having the camera in front of the classroom. The best position for the camera would have to be the middle of the classroom. Figure 6 displays the testing area of the project. It is the left and right hand side of a class. The boxes on the chair indicate the area needed to identify the open seat. If the camera cannot detect a chair due to it being too far or a person blocking its field of view, our hit ratio of empty chairs detected would decrease.

Figure 6

The timing and positions of the servo motors was also tested and calculated to be in sync with the sending of the x and y locations of the objects to the server. Different pulse width modulation signals varying in duty cycle were used to observe the speeds and positions of the servo motors.

VII. CONCLUSION

The Spotter 1000 introduced us to designing a system with the knowledge we have learned in our classes at University of Central Florida throughout the past four years. We used an embedded microcontroller, servo motors, an image processing camera, and created our very own server and android application. We applied computer science theories to maximize our run time.

The project had an equal balance of hardware and software design. The work ranged from programming in JAVA and C to creating a schematic and printed circuit board for our very own embedded controller. Helping the students of UCF find seats in class motivated us to work hard this semester. We know how important it is to have this information before commuting to class and wanted to make it available for students to access it. An android application was the perfect solution to distributing and viewing this information.

In the future we can use this system in parking lots and for reliability have a panoramic view instead of a pan and tilt camera system. To reduce the cost, a standard color camera with a sufficient amount of resolution can be used, and all of the heavy image processing can be done on our server. The design we currently have can only have a max range of about 6 rows of seats. To increase the range the mount would need to be raised higher and the resolution of the camera needs to be increased.

Due to using a hue-based color filter, our design relies on students not to wear the same color as the signature saved on the camera to detect. This approach is not reliable in real life application. Machine learning would help increase the reliability of the system by having the system trained to differentiate between positive and false objects. Our design can be improved greatly, but as of now it successfully detects a high percentage rate of seats in a classroom.

ACKNOWLEDGEMENT

The authors wish to acknowledge the assistance and support of the various online communities such as Carnegie Mellon University for the creation of the Pixy Camera and Robert Wessels for starting the open sources electronic prototyping platform Energia. These sources played a significant role in the completion of both the hardware and software portions of this project.

Wandy Pierre is a Senior at the University of Central Florida and studying Computer Engineering. He is looking to pursue a career to provide for his family and his newborn daughter.

Richard Macias is an Electrical Engineering Senior at the University of Central Florida. He is looking to go to Graduate School for his Masters while working at Saint John's Optical Systems.

Jay Davila is a Senior at the University of Central Florida studying Computer Engineering. He is looking to pursue a Career as a Communications Patent Examiner with the United States Patent and Trademark office.

REFERENCES

[1] Texas Instruments "Mixed Signal Microcontroller," IEEE., May 2013.

[2] Anthony Rowe, Rich LeGrand, Scott Robinson, "How Pixy got started", CMUCam.org