SEER Optics

S. Fick, EE, D. Gabriel, EE, D. Shosho, CSE, and O. Terry, EE

***Abstract* — SEER Optics looks to aid civil servants by providing an advanced visual aid allowing teammates to be seen in real time. Missions conducted by civil servants like Firemen and Police Officers are often very hectic and confusing environments to be in. This can hinder missions and cause them to become more dangerous. Our design will allow everyone wearing this device to see the locations of each other even with obstructions in the way and allow these users to call out or "ping" objects that they are seeing. This design sends GPS locations over radios and uses the difference in GPS coordinates to find relative location. This location is then displayed on augmented reality optics as a green dot. This allows the user to see their environment with the enhancement of having these green dots displaying the locations of the others. Our MDR deliverable makes the use of a Google Pixel with the camera running real time to show this augmented reality.**

I. Introduction

Our product, SEER optics, will be designed to provide a visual aid that takes in teammates GPS data through radio and displays where they are on the SEER Node. In addition, the product allows users to share a marked location using GPS data, called pinging, which allows for more efficient communication.

A. Significance

Civil servants (i.e. firefighters, policemen, search & rescue) suffer from a lack of situational awareness when conducting their missions. This creates a fog of war that leads to friendly fire, team members missing in action, delays in missions and other unnecessary issues. For example, there was a New York police officer killed by friendly fire during the capture of a suspect, another similar instance where an officer got caught in the crossfire, and an instance where the Coast Guard was unable to locate missing Fire Fighters that were lost at sea. All of these tragic incidents could have been remedied/prevented with our product. [1][2][5]

B. Context and Existing Products

Currently there are three main solutions that try to solve our problem but fall short in aspects that our product will succeed. These solutions are verbal communication, line-of-sight tracking via reflectors [3], and blue-force tracking [6]. Comms devices like phones are a great ally for these civil servants during their missions, but in the advent of disaster you can't depend on mass infrastructure to keep teams coordinated (i.e. cell towers/landlines being put out of commission) [4]. Similarly, reflectors are great for short range missions with little obstructions, but this solution is specific to low-light/high-light (fire) situations and doesn't work well in situations with abnormal environments like collapsed buildings or forests (a situation where being able to see through obstructions is advantageous). Finally, blue-force tracking is the closest competitor to our product, being that it offers a lot of the same benefits. The only issues with this solution being that some form of verbal communication is still required, and the team members depend on a central command operator that coordinates the team (our product is local to each teammate and doesn't require verbal communication).

C. Societal Impacts

With our product, these civil servants can remedy the short comings of the other currently employed solutions. Our solution operates on personal, closed-system hardware that updates the individual with real-time whereabouts of every team member and their pinged locations. This combines the team coordination of verbal communication, the individual real-time visuals of reflectors, and the technology used in blue-force tracking. See Figure 1 below for a rudimentary depiction of the system where "You" is the SEER Node and there is a "Teammate" Node.

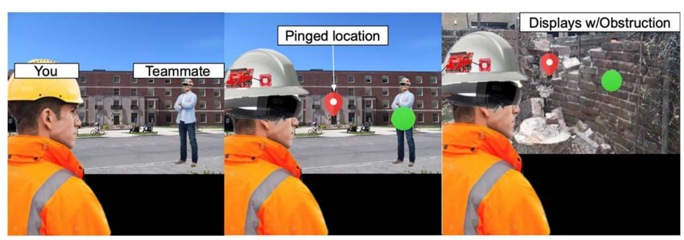

Figure 1: Depiction of the system in practice

D. Requirements Analysis and Specifications

As for the specifications of our product, seen in Table 1 below, we aimed to provide the most accurate and efficient system we could devise. For example, we chose to limit the range on our device to the specified due to inaccuracies of our GPS chip at the query timing we settled on for speed. When constrained to this range, we are able to achieve our desired

S. Fick from Pittsfield, MA (e-mail: sfick@umass.edu).
D. Gabriel from Portsmouth, RI (e-mail: dgabriel@umass.edu).
D. Shosho from Revere, MA (e-mail: dshosho@umass.edu).
O. Terry from Amherst, MA (e-mail: oterry@umass.edu).

display accuracy 95% of the time. To that end, we have plans to gather test data that shows we meet these specifications to the accuracy we claim. Additionally, the battery of the Google Pixel is rated at ~5 hours when using it constantly, this is the minimum battery life we want and aim to be at double or even triple that. And lastly, the Google Pixel is rated at IP53, which means it is protected against dust and should be unphased by a light rain. Our Node will at least match this specification if not supersede it.

Requirement	Specification
Seer Node sees teammates with or without obstruction	Range of 25<x<300m
Node can ping locations relative to them	Range of 0-50m relative to originator
Seer Node sees pings with or without obstruction	Range of 25<x<300m
Displayed teammates and pings will be accurate	Within specified range at +/-5 degrees
Node will have long-lasting battery	Node will last as long as the Optics
Teammates GPS data is updated quickly between Nodes	Location is queried <500ms
Node will withstand moderate weather	Up to the standards of the Optics

Table 1: Requirements and Specifications

II. DESIGN

A. Overview

Our solution involves allowing users to see their visually obscured teammates locations in real time through augmented reality optics. To determine the relative location, we will use GPS location data and gyroscope technology. The user will also be able to ping objects that they would like to mark. To show this location we will use a Google Pixel showing what is seen from the camera plus dots overlaid showing locations of the nodes. We also have put a lot of consideration into using the Microsoft Hololens as our augmented reality optics but have made the design decision to go with the Google Pixel based on our design to interface with the AR Optics. The full layout of our design can be seen in Figure 2 below. The subsystems shown here are the Node, the SEER Node, the Ping Controller, and the optics where

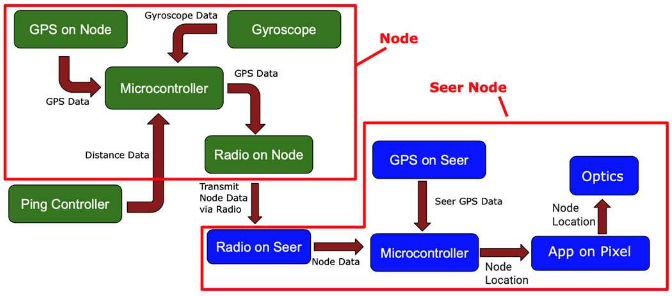

Figure 2: Block Diagram of the SEER Optics system.

B. Node

The Node will be the hardware that is carried by the person that is being seen by the SEER Node. This will include a GPS chip [11], microcontroller [10], Gyroscope, and a 915MHz radio transceiver [9]. This Node is untethered and powered by a 9V battery. This allows the node to have a longer battery than the limiting power factor of our project which is the Microsoft Pixel with a battery life of around 12 hours. This microcontroller will take inputs from the Ping controller which will be an analog input, The GPS chip, and the gyroscope. This info will be packaged by the microcontroller and sent over the RFM69 915MHz LoRa radio where it will be received by the SEER Node.

C. SEER Node

The SEER Node will be the user that is able to see the Nodes location in real time on the Google Pixel. The data from the Node will be received over another 915Mhz radio receiver and will be processed by the microcontroller on the SEER Node. This SEER Node has its own GPS chip. The microcontroller compares the data from the Nodes coordinates and the SEER Nodes coordinates to calculate the location of the node relative to the SEER. The gyroscope data will be used for direction of pinging and is explained more in section E. This data will then all be sent serially to the Google Pixel through a micro USB.

D. Google Pixel

This is the device that we have decided to use as our augmented reality optics, seen in Figure 2 below. We have developed an app for the google pixel that will take in the location and ping data from the SEER Node and display it as an overlay on live video that is being seen through the phone's camera. This app was developed using Unity software. This app will show relative location and distance of the node and the objects that the node pings.

E. Ping Controller

The Ping controller is a piece of custom hardware that we have designed. This will be wired into the Node so that an analog voltage can be read off this device. The interface can be seen in Figure 3 below. This controller has four buttons and one dial. The buttons can be used to mark different types of objects, up to three different types. There is also a button to clear all of the pinged objects. The dial can be used to estimate the distance away from the user that the object is. This dial can be scrolled to show 0-50 Meters. When a button is pressed the Node will take the info about the distance from the dial and the type of object and send it over the radio to the SEER Node. The direction of the object will be known by the gyroscope on the Node so the Node must be facing the object when pressing the ping button.

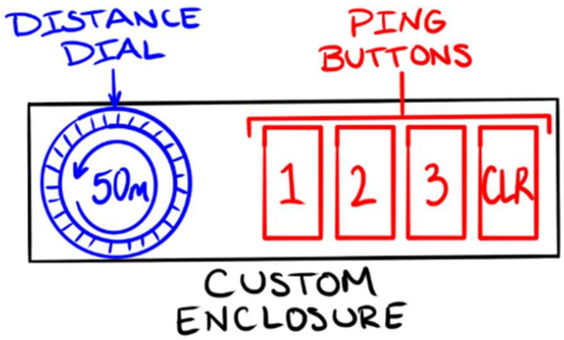

Figure 3: Ping Controller exterior design schematic.

III. Project Management

By the end of the fall semester, our team has successfully designed a prototype of SEER Optics with almost all of its functionality, besides the implementation of the gyroscopes. More specifically, the current state of the project exists as such: First, we have successfully interfaced the GPS component and the ping controller on the 'node' with the microcontroller (also on the node), which then interfaces with a radio connected to the node. Second, we have successfully sent our payload from the radio on the node to the radio on the 'seer'. Third, the microcontroller on the seer successfully interfaces with the radio (receiving the payload from the node) as well as receiving the second GPS payload from the GPS component. Fourth, the microcontroller on the seer does a computation with both sets of GPS data and sends a payload to the Google Pixel via USB serial communication containing pinging information as well as a distance/angle payload. Finally, the application on the Google Pixel displays the node's location in real-time as a green dot and displays pinged locations in different colors.

As for where we are going next semester and our plans for the future, our first goal is going to be to complete our design by implementing the gyroscope interfaces on both the node and the seer. This will make sure that the node is able to ping with a direction and distance as well as make sure that our application can update the location of the node accurately with both users walking around in any direction in the Engineering Quad. With our full design implemented, the next step will be to refine the application by: First, making the display more seem-less and easier to understand. Second, improving functionality and accuracy of the node being displayed. And third, speeding up the rate at which the node location is updating and ensure gyroscope functionality. The next step would be to gather data and information to ensure that our specifications for our project are met and are within our standards. This will consist of a lot of redundant testing to make sure the design is robust. After testing, we plan on getting some kind of optic so the user can put their phone inside of it and use our product as a heads-up display like it is intended too. We also plan on making custom enclosures for our hardware to protect the inner components from the elements as this product is intended to be used outside.

One significant difficulty we plan on running into is converting our design into a custom PCB. Since our design is complex and requires many interfaces, converting our significant hardware components into a PCB will be very difficult. We will tackle this difficulty by communicating with Prof. Goeckel every week in order to keep him up to date and ask for advice, by keeping an organized schedule so all group members are always working on the project at some point every week, and by using the abundance of resources the Engineering Department has to offer like M5 and instructors. Another difficulty we plan on running into is interfacing the gyroscope with both the microcontroller on the node as well as the microcontroller on the seer. This also includes interfacing the payload sent from the seer to the phone via the application using Unity (which comes hand in hand with implementing the gyroscopes in general). We will tackle this challenge by utilizing the vast knowledge and support on the internet, reading gyroscope documentation from SparkFun, and once again being committed to continuously work on the project in the lab. Below, in Figure 4, you can see a plotted our plan for next semester.

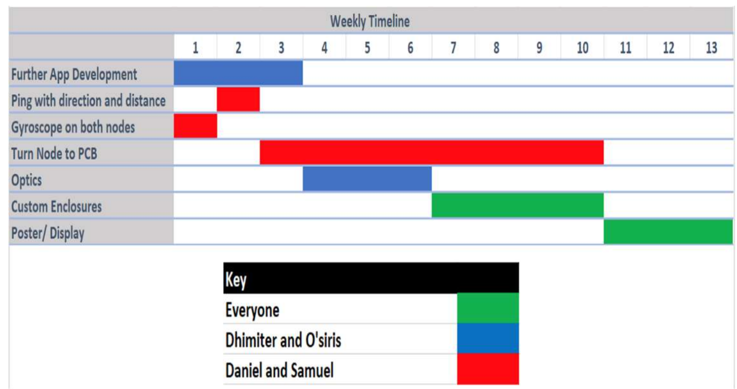

Figure 4: Gantt Chart for spring semester.

IV. Conclusion

Given the complexity and difficulty of this project, the result of all of this progress throughout the fall semester has directly come from the perseverance, passion, and dedication of every group member. From struggling with interfaces like the Microsoft Hololens and the radios, learning new software like Unity, and making difficult but crucial design decisions, our team has successfully put ourselves in good standing for the spring semester to clean up and refine our entire project.

All in all, our team is in good standing for next semester. Although we have a lot of different and difficult challenges to face in the spring, we have a solid and robust plan that will help us succeed. We greatly look forward to cleaning up, finalizing, and eventually providing a demo of our project to evaluators as well as others once we showcase our final design.

Acknowledgment

We would like to thank fellow SDP20 Team 1 and more specifically colleagues Ibrahim Tahoun and Cormac Kennedy for letting us borrow their android phone for development while we did not have our Google Pixel. This support from them was much appreciated during the debugging phase of our prototype as we had an actual unit to test the application that was developed for SEER Optics using Unity.

We would also like to thank our advisor, Professor Goeckel, and evaluators, Professors Gao and Aksamija, for thoughtful

insight, critique, and advice throughout the development of our project.

References

[1] A. Watkins and A. Southall, "N.Y.P.D. Officer Was Killed by 'Friendly Fire' During a Struggle With Suspect," *The New York Times*, 30-Sep-2019. [Online]. Available: https://www.nytimes.com/2019/09/30/nyregion/bronx-officer-brian-mulkeen-nypd.html. [Accessed: 24-Sep-2019].

[2] D. Wallace, "Coast Guard suspends search for firefighters lost at sea after Florida fishing trip," *Fox News*, 23-Aug-2019. [Online]. Available: https://www.foxnews.com/us/coast-guard-florida-suspends-search-two-firefighters-missing-fishing-trip-justin-walker-brian-mccluney. [Accessed: 24-Sep-2019].

[3] Fischer, C. and Hazas, M. (2019). *Firefighter - an overview | ScienceDirect Topics*. [online] Sciencedirect.com. Available at: https://www.sciencedirect.com/topics/engineering/firefighter [Accessed 24 Sep. 2019].

[4] Frank, T. (2019). *Cell Phone Service Must Be Restored Quicker after Hurricanes*. [online] Scientific American. Available at: https://www.scientificamerican.com/article/cell-phone-service-must-be-restored-quicker-after-hurricanes/ [Accessed 24 Sep. 2019].

[5] NBC, "NYPD Releases Bodycam Footage from Fatal Shooting of Det. Brian Mulkeen," *NBC New York*, 14-Dec-2019. [Online]. Available: https://www.nbcnewyork.com/news/local/nypd-releases-bodycam-footage-from-fatal-shooting-of-det-brian-mulkeen/2241734/. [Accessed: 24-Sep-2019].

[6] U.S. Army, Timeandnavigation.si.edu. (2019). *The Blue Force Tracker System | Time and Navigation*. [online] Available at: https://timeandnavigation.si.edu/multimedia-asset/the-blue-force-tracker-system [Accessed 24 Sep. 2019].

[7] "Developer Guides: Android Developers," *Android Developers*. [Online]. Available: https://developer.android.com/guide/index.html. [Accessed: 1-Dec-2019].

[8] *SIK Experiment Guide for Arduino - V3.3*. [Online]. Available: https://learn.sparkfun.com/tutorials/sik-experiment-guide-for-arduino---v33. [Accessed: 3-Nov-2019].

[9] "SparkFun RFM69 Breakout (915MHz)," *WRL-12775 - SparkFun Electronics*. [Online]. Available: https://www.sparkfun.com/products/12775. [Accessed: 26-Jan-2020].

[10] "SparkFun RedBoard Artemis," *DEV-15444 - SparkFun Electronics*. [Online]. Available: https://www.sparkfun.com/products/15444. [Accessed: 26-Jan-2020].

[11] "SparkFun GPS Breakout - Chip Antenna, SAM-M8Q (Qwiic)," *GPS-15210 - SparkFun Electronics*. [Online]. Available: https://www.sparkfun.com/products/15210. [Accessed: 26-Jan-2020].

Appendix

A. *Design Alternatives*

At a system level, we had a few design alternatives that we ended up discussing. One system level design alternative we discussed was using Wi-Fi and an Internet connection instead of using radios to transmit data between the Node and the Seer Node as well as transmitting data from the microcontroller on the Seer Node to the Microsoft HoloLens. This design alternative would have required a Wi-Fi chip that would give our microcontrollers the ability to access the internet. This design alternative would have also required a database in which we would interact with, updating the database and pulling information out of it as well every 500ms. We decided not to implement this alternative method of data transfer because we realized it would require a strong and consistent internet connection in both the Node and the Seer Node that would not be feasible outside in the Engineering Quad.

Another system level design alternative that we considered was an attempt to solve the same data transfer problem we were having in the previous paragraph. This time, we discussed using Bluetooth instead of using radios to transmit data between the Node and the Seer Node as well as transmitting data from the microcontroller to the Microsoft HoloLens. This system level alternative would have required two additional Bluetooth modules, one for the Node and one for the Seer node being implemented and would essentially replace the radios in our design. We did not go through with this design alternative because, after struggling to communicate with the Microsoft HoloLens, we learned that it did not support Bluetooth connections with devices besides Microsoft products like mice and keyboards.

As mentioned in our overview and conclusion, we have put a lot of effort in trying to interface with the Microsoft Hololens. After extensive research, we decided to move to our main alternative, the Google Pixel. The reason we moved away from the Microsoft Hololens was because the developer documentation revolving around the product was lackluster when compared to other augmented reality capable products. Furthermore, the lack of documentation sprouted to other problem when we attempted to interface with the product. For example, while Microsoft computers are capable of serial port communication, the Microsoft Hololens is not because of the operating system that it runs on.

We defined our criteria for an alternative to be any device that was capable of running a Unity program and could be used as an augmented reality device. The alternatives that met our criteria were smart phones and other augmented reality devices similar to the Microsoft Hololens such as Google Glasses. Due to the vast amount of developer documentation and cost efficiency we decided on the Google Pixel [7]. Creating alternatives gave us the intuition to know when we should try to interface with another product or to find a new interfacing method.

B. *Testing Methods*

Our product, Seer optics, requires a plethora of interfaces. Every interface we designed went through an excessive number of tests. Firstly, grabbing GPS data. This interface was between the Arduino Uno and the GPS chip, see Figure 2. Our test to make sure this interface worked was to first print out the GPS data to the serial port where we could view it on a serial port monitor. To further test our system, we walk around outside to see how reliable the GPS chip was. We concluded that the device is reliable outside but completely cuts out when in doors. We tested the GPS chip outside of the lab, but further tests need to be conducted in areas such as forests or under certain weather conditions in order for us to deem the reliability of this product.

In addition to printing out the GPS data, we made sure that the data changed accurately as we walked around with the GPS chip. We tested this by walking around the engineering quad with two GPS chips, measuring the distance between each other and then using the GPS data from the chips to calculate the distance between us. Then we compared the measured distance to the GPS distance and found that it was around a 3% error. When we conducted the same test in the rain, we found that the error jumped to 5%.

Secondly, sending GPS data over radios. This interface was between the Seer Node and the Node, see Figure 2. Our test to make sure this interface worked was to send a phrase of the radios to make sure the other end could receive data. After we were able to send a message over the radios, we tested their accuracy with distance and inside of buildings. We found that the frequency was effective for the distance that we specified in the specifications regardless of the obstacles that were in the way. Our future tests with this interface will revolve around testing which baud rates increase the speed at which we can send the data.

Lastly, serial port communication with the Google Pixel. This interface was between the Arduino Uno on the Seer Node and the Google Pixel, see Figure 2. Our test to make sure this interface worked was to write code to pull data from the serial port and display it onto the screen to make sure it was what the Arduino Uno was sending to the serial port. After we were able to get the data in the correct format, we attempted to try send data that would vary in length to see if the serial port buffer would cut off any of the data being sent. We found that none of the data being sent was corrupted. After our testing of the serial port communication we deemed that the interface met our goals and required no more testing. All of these interfaces were referenced with SparkFun's guide [8].

C. Team Organization

We as a team communicated about everything throughout the semester. The team had a shared idea on what the product would look like and to verify that it was shared vision we would communicate how we thought the design would operate whenever we made any changes to our original idea. When designing the interfaces all team members were present and were able to explain, face-to-face, how the interface would work and what the format would be of the output. This method of communication was extremely effective in our project because when a teammate would test the interface another teammate made both of them would be present and be able to fix any errors that occurred. The team was split into a hardware and software group since the work was even on both ends of the hardware-software spectrum. The software team, Dhimiter and O'siris, were able to test different input to their program before the hardware team finished their design. This was helpful when connecting the hardware and software at the end. The hardware team, Daniel and Samuel, communicated their design to the software team and the output of their system with extreme detail which was pivotal in the progression of the team.

The team never broke down and never sway from the idea that we had to all be there together until the product was finished. There were still road bumps that we had to get passed. The main one was having to switch from the Microsoft Hololens to the Google Pixel three days before MDR. This meant that the majority of progress that the software team made had to be scrapped because of the change in platforms. The software team spent their time before MDR leaving the lab no later than 3 a.m. During this time the hardware team had finished majority of their work but still stayed by the software teams' side until the product was finished. This proved to be very helpful for morale and efficiency as some aspects of the hardware needed to be altered in order to interface with the Google Pixel properly.

D. Beyond the Classroom

Most of the team shares the same problems. During presentations Dhimiter has made leaps and bounds on being more concise with his words in order to communicate his points better, but there is still improvement to be made for next semester. The other team members also have room for improvement for not only presenting but for also considering changes to the product they are developing. O'siris has improved with accepting new ideas when they come about but still has a lot more room for improvement when it comes to patience. For next semester both O'siris and Dhimiter's goals are to take a deep breath before they speak/listen and for both to orient themselves to the information they are receiving or the information they are about to give out. Samuel has improved greatly in his patience with teammates this semester. Samuels goal for next semester is to adapt to people's ideas and moods in order to bring out the best in them. That way in times that morale falls low he will be there to set the team back on track. Daniel contributes a lot to the team when a new idea is proposed because he uses his critical thinking skills and provides the team with foresight on how this addition could hurt or help the team. Daniels goal for next semester is to provide this constructive criticism in a way that improves upon the idea without bringing others down.

For technical skills, Dhimiter grew in his understanding on how we connect the software to the hardware. Dhimiter was heavily involved in this interface and had to overcome the road bump of not having taken many hardware classes. O'siris was involved on the same interface as Dhimiter but had a different challenge to overcome. O'siris overcame the discomfort of moving from his hardware environment to programming the software on the app. Samuel and Daniel grew in communicating the circuits they designed to the software team. They also grew in their ability to listen and implement the requests of the software team. Each member provides skills that are pivotal to the teams' effectiveness and chemistry.