2023

Heads Up/Hands Free RSA Display

FINAL REPORT

Aashish Anand, Alec Uyematsu, Connor Morgan, Devam Patel, and Tony Hayden

UNIVERSITY OF PORTLAND | 5000 N Willamette Blvd, Portland, OR 97203

Table of Contents

1. Statement of Purpose

The primary aim of this Final Report is to present a well-rounded summary of the project's development and progress to Tektronix and future capstone teams. This document serves as an essential reference for everyone involved, providing a thorough understanding of the project's background, current standing, and potential direction. Through a transparent and in-depth evaluation of the project, we hope to ensure smooth handovers between capstone teams and keep Tektronix informed on all project advancements. In detail, we will discuss the previous team’s state of the project, our goals, challenges we encountered/overcame, our current minimum viable product (MVP), and discuss probable future steps which can be deployed.

2. Overview

The objective of this project was to develop an innovative heads-up and hands-free Realtime Spectrum Analyzer (RSA) display system, aimed at significantly enhancing the capabilities of Tektronix field surveyors during their monitoring and signal-hunting operations. By leveraging innovative Augmented Reality (AR) technology with the Microsoft HoloLens 2 we intended to revolutionize how field surveyors interact with and analyze spectrum data.

The integration of AR technology into this project provides many benefits for Tektronix field surveyors. By using the Microsoft HoloLens 2, an advanced and powerful AR headset, surveyors can visualize the crucial DPX graph right in front of their eyes. This graph, which represents the spectrum signals interpreted by the RSA, allows surveyors to monitor real-time data without the need for additional equipment or constant manual adjustments.

The hands-free and heads-up nature of this AR solution offers a significant improvement in operational efficiency and safety for field surveyors. With their hands unencumbered and their attention undivided, surveyors can focus on the task at hand. Whether that be identifying interference sources, tracking down rogue signal emitters, or conducting routine spectrum monitoring operations. This streamlined approach minimizes distractions and the risk of errors while optimizing the overall workflow.

The development of a heads-up and hands-free RSA display system using Microsoft HoloLens 2 seeks to elevate the capabilities of Tektronix field surveyors by optimizing the monitoring and signal-hunting processes. The integration of AR technology offers a more efficient, safer, and user-friendly solution, enabling surveyors to visualize critical spectrum data and interact with it in a more natural and immersive manner.

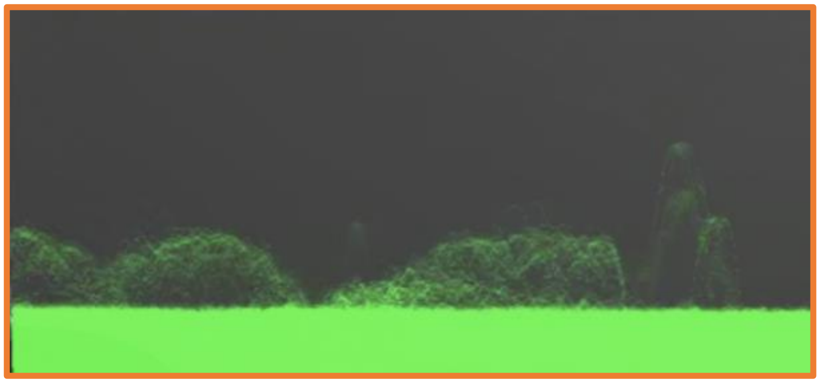


Figure 1
Figure displays what the user wearing the HoloLens would see while hunting signals.

3. Previous Capstone Work

Over the past couple of years, two distinct capstone groups have contributed to the development of this project, with each team tackling various aspects and building upon the foundation laid by their predecessors. The primary goal of the project has been to create a reliable and efficient Unity application that runs on the HoloLens, with the capacity to provide real-time visualization of Realtime Spectrum Analyzer (RSA) values.

The initial capstone team, responsible for the project's inception, successfully developed a Unity application that functioned on the HoloLens platform. However, the application's primary limitation was its reliance on simulated or "dummy" values in lieu of actual RSA data. Despite this drawback, the team's effort provided a valuable starting point for the subsequent capstone group.

The second capstone team focused on improving the application by incorporating real RSA values. They developed a Python script capable of generating PNG images, which were then uploaded to the HoloLens via a method known as Holographic Remoting. This approach allowed for the integration of actual RSA data into the application, representing a significant step forward in the project's development. Still, the second team's solution faced performance limitations, due to Python as the development language. The application's frame rate suffered, resulting in a suboptimal user experience, as it only managed to achieve 1.5 to 2 frames per second. This performance bottleneck made the application difficult to use in practice and highlighted the need for further optimization.

Additionally, the second capstone group's application lacked key features that would have allowed users to adjust settings such as frequency, span, and amplitude. The absence of these customization options limited the application's overall functionality and utility for its intended audience.

4. Our Goals

Our primary goals for the 2022-23 capstone team were the following:

1. Increase frame rate of the DPX graph.
2. Update the user interface (UI) to add RSA settings modification.
3. Clean up the old UI.
4. GPS tracking.
5. Voice commands.

5. Challenges

There were quite a few challenges which our team had to address throughout the year. Next, we will briefly discuss each challenge and how we overcame it.

5.1 Lack of Documentation

One of the year's biggest challenges was the lack of documentation presented to us by the previous capstone team. Due to there being almost no documentation on the previous project, our team spent the first half of the year learning about the old project and how we can set it up to run it. After learning about some of the project's basics, we were quickly able to adapt and begin working on our new optimal solution. The lack of documentation was an important challenge to face, as due to this, we as a team decided that we will make sure to write proper documentation for all our future work, so a potential future team does not encounter this challenge.

5.2 Unity Problems

To develop for the HoloLens headset, we had to use Unity to develop our UI and C# scripts. Unity was an incredibly challenging software to work with. We had to develop it into an older version of Unity because that is what the previous team had used. Due to this, we encountered various challenges with multiple features and packages being outdated or outright deprecated. Additionally, we had the issue of frequent crashes on Unity. Unity required a lot of computational power to render the UI and run other scripts, due to this the application resulted in an innumerable number of crashes which drastically hurt our team’s development process.

In the case of Unity, there was not much our team could do to overcome the challenges of frequent crashing.

5.3 Marshalling Errors

With the goal of switching the entire codebase to be in Unity, this meant all the API calls would have to be made in C#. Unity recompiles all the DLLs that are included in the project. This meant any existing examples showing how to use the Tektronix libraries would need tinkering just to be able to access the library calls required. This led to 3 months of debugging Marshalling errors (Marshalling is C#'s way of specifying how to interpret the library calls specified). Due to the outdated version of Unity and the nature of the bugs that would occur due to incorrect Marshalling, the only indication that a mistake was being made was Unity crashing with no error logs. This not only made debugging almost impossible, but it also gave us no insight into what needed to change with Marshalling.

Eventually, the fix was to go into the settings and disable “Safe Mode” which allowed us to forget about Marshalling and instead have the entire class that deals with the library calls exist in an unsafe environment. While we believe this is not the best way to get around this error, this was our last option and what we settled on. Newer versions of Unity could have improved this debug process while also allowing for different ways to include the DLLs.

5.4 ScottPlot Experiment

One of our goals was to increase the framerate of the product, and to do this we tried to shift away from the use of python to C# as the development language. We had to find an alternate library which could be used to replace python's *matplotlib* module which was used to generate the DPX graph. Our team found ScottPlot to be a suitable alternative to *matplotlib*, which was another promising graphing library.

However, after spending around three weeks trying to set up and implement ScottPlot graphs into our codebase, we discovered a new challenge with the library. ScottPlot uses the *System.Drawing* module to generate its graphs, unfortunately, the unity application did not support the *System.Drawing* module. Since our primary development was done in unity, we had to move on from using ScottPlot and look for a better option.

Finally, we shifted from plotting libraries to using the built-in functions in the RSA API. We used the RSA API to generate a bitmap image of the DPX graph, which we then plotted onto a simple material plane object in unity.

5.5 High Latency in new UI

One of the key issues faced during the integration of the new user interface was high latency, which the previous capstone team encountered. Despite extensive efforts to optimize the FPS, bottlenecks persisted due to insufficient system resources and software and hardware constraints. This latency issue significantly impacted the overall user experience and hindered the functionality of the system.

To mitigate this challenge, our development team worked tirelessly to find and eliminate all sources of latency. We analyzed and assessed the hardware components and identified the components that were not strong enough to handle the system's demands. We concluded that hardware upgrades were necessary to improve the system's performance, and thus, the user experience.

Although we were able to improve the system's FPS, bottlenecks persisted, and we realized that stronger hardware was required to achieve optimal results. With more robust hardware, the user interface would function optimally, and users would experience an improvement in the overall system performance.

5.6 Voice Commands

Originally, we wanted our voice commands to increase or decrease an RSA setting by a number spoken by the user. The problem with that was whenever the user says a number, it only processes the word version of said number. Mapping the numbers to their respective word versions was difficult for us to implement so we decided to limit them to the following: “Increase Frequency,” “Decrease Frequency,” “Increase Span,” “Decrease Span,” “Increase Amplitude,” and “Decrease Amplitude.” If a keyword from that list is spoken, then the RSA setting changes by a fixed increment. For frequency, the increment value is 10 MHz, the span increment is 1 MHz, and the amplitude increment is 1.

6. Minimum Viable Product (MVP)

At the time of writing this final report, our team has successfully developed an MVP for Tektronix. Our final product contains many of the goals which were previously defined, including increased framerate, voice commands to change RSA settings, and an updated cleaner looking user interface.

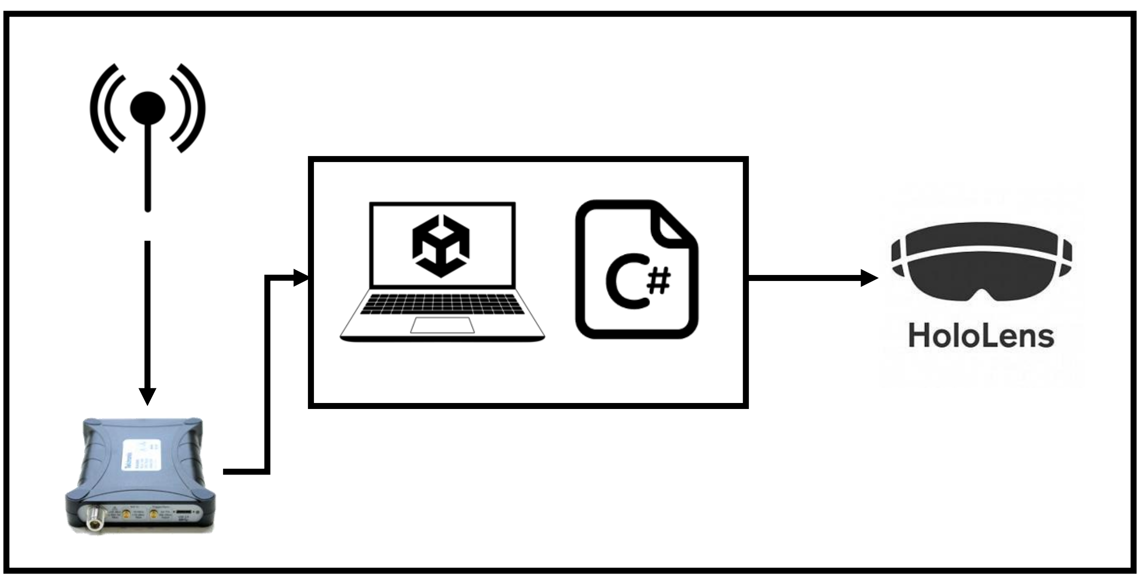


Figure 2: System Diagram

6.1 400% Increased Framerate

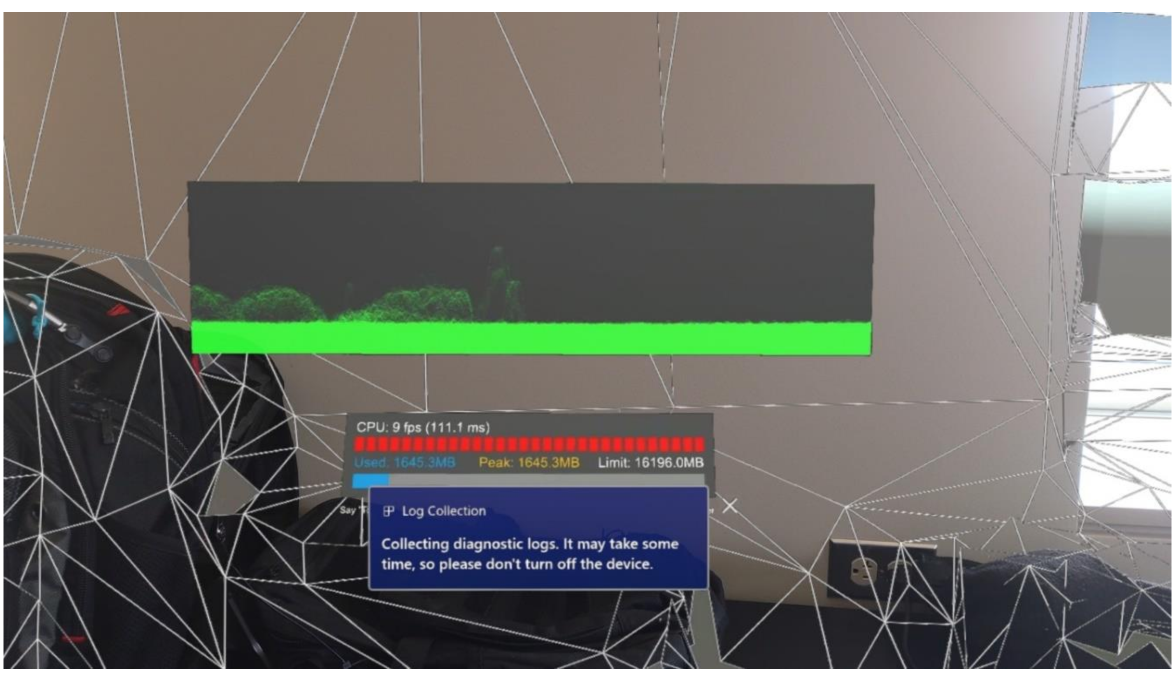
As mentioned previously, our team decided to shift from python as the programming language to C#. Due to this shift, along with not using python's *matplotlib* anymore, we were able to increase the product's framerate from **1.5 – 2fps** to **9-10 fps**, which is nearly a 400% increase. We concluded that Python was a much slower language and the graph generation in *matplotlib* was also very slow. The shift to C# allowed us to find alternate ways to improve the program's framerate.

6.2 Voice Commands

Another goal for our project was to update the UI such that it included interactivity to update the RSA settings. To do that, we added voice commands to the application such that the user would interact with the application without the use of their hands. In Unity, there is a class called KeywordRecognizer that listens to the speech input and maps phrases to certain keywords. Once the keywords are recognized, the recognizer invokes an action that changes the RSA settings accordingly via the OnPhraseRecognized handler.

6.3 Updated UI

A substantial portion of the deliverables was an improved user interface. We decided to take inspiration from the Tektronix SignalVu app. In this app there is a monochromatic version of the DPX graph with the signal density reflected in the transparency of the green (Higher signal density = lower transparency). This allows our intended users to easily interpret the signal density and eases the transition from other programs our intended users have experience with.



7. Future Steps

1. Look into other AR devices. As Microsoft has now cut funding for future HoloLens support it would be beneficial to investigate some different AR technologies that are still being supported.
2. Add buttons that change the settings. Right now, the only way we can change settings is through voice commands. It would be nice to add buttons to the UI that allow the user to change settings as well in case something goes wrong with the voice commands.
3. Add a geolocation display to the UI.
4. Interpret signals received into a log file accessible after field surveying is done.
5. Increase frame rate if possible.
6. Look into creating an application that does not take advantage of Holographic remoting.

8. Support

For any type of assistance needed with setting up/using the HHRD system, please contact any member of the support team below.

Contact	Organization	Email	Role
Aashish Anand	University of Portland	ananda23@up.edu	Developer
Alec Uyematsu	University of Portland	uyematsu@up.edu	Developer
Connor Morgan	University of Portland	morganco23@up.edu	Developer
Devam Patel	University of Portland	patelde23@up.edu	Developer
Tony Hayden	University of Portland	haydena23@up.edu	Developer
Kyle Bernard	University of Portland	bernardk@up.edu	Advisor
Surj Patel	University of Portland	patels@up.edu	Advisor
David Huddleson	Tektronix	david.huddleson@tektronix.com	Client
Keith Tinsley	Tektronix	keith.tinsley@tektronix.com	Client

APPENDIX A: Reference Documents and Software Links

Document Name	Document Location and/or URL
GitHub Repository	https://github.com/morganco23/UP_Tektronix_Capstone
RSA306 Manual	https://www.tek.com/en/spectrum-analyzer/rsa306-manual
RSA API Manual	https://download.tek.com/manual/RSA300-500-600-Series-SpectrumAnalyzers-API-Programming-Reference-077103104.pdf
Unity 2020.3f Manual	https://docs.unity3d.com/2020.3/Documentation/Manual/UnityManual.html
RSA API Repo	https://github.com/tektronix/RSA_API
Mixed Reality Toolkit	https://www.microsoft.com/en-us/download/details.aspx?id=102778
SignalVu	https://tinyurl.com/3bcrkv9t
Unity Hub	https://unity3d.com/get-unity/download