

AR Monitoring System

Design Document

Team Number: Dec1714
Client: Andrew Gui
Advisor: Dr. Johnny Wong
Arbaaz: Team Leader
Ben: Key Idea Holder
Sam: Key Idea Holder
Nipun: Communication
Patrick: Webmaster
Dheeraj: Communication
Dec1714@iastate.edu
Dec1714.sd.ece.iastate.edu

April 27, 2017
Version 2

| | |
|------------------------------|----------|
| 1 Introduction | 2 |
| 1.1 Project Statement | 2 |
| 1.2 Purpose | 2 |
| 1.3 Goals | 2 |
| 2 Deliverables | 3 |
| Server | 3 |
| Raspberry Pi/Remote Device | 3 |
| Hololens App | 3 |
| 3 Design | 4 |
| 3.1 System Specifications | 4 |
| 3.1.1 Non-functional | 4 |
| 3.1.2 Functional | 4 |
| 3.1.3 Standards | 4 |
| 3.2 Proposed Design/Method | 5 |
| 3.3 Design Analysis | 5 |
| 4 Testing/Development | 6 |
| 4.1 Interface Specifications | 6 |
| 4.2 Hardware/Software | 6 |
| 4.3 Process | 6 |
| 5 Results | 7 |
| 6 Conclusion | 7 |
| 7 References | 7 |

1 Introduction

1.1 Project Statement

With the advancement in technology the ability to simplify life has become easier. With the possibilities of AR, we can minimize the human labor and make the tracking/oversight of autonomous vehicles easier. The intention of this project is to design an augmented reality monitoring system that will be integrated into the cab of an autonomous tractor. This system is based on a HoloLens application that will load up an aerial map view of a user defined location and show virtual 3D models of the actual vehicles moving throughout the map in real time. The user can then click on and access the information of a vehicle (i.e. current operator of vehicle, speed, etc.) and be able to experience a first person view of what the driver is seeing.

1.2 Purpose

This project aims to provide an easy-to-use and vertical solution to remotely monitor and/or control autonomous vehicles. With enough refinement, the solution could be used for efficient oversight of a work force or autonomous fleets.

1.3 Goals

Upon completion of the project, we hope to have a complete set of software and hardware needed to remotely manage, store, and visualize a wide array of devices and machinery. This would include but not be limited to a lightweight device mounted on a monitored machine, a centralized server, and software written for the Microsoft HoloLens. We also hope to gain a greater understanding and appreciation for AR in general and to improve our abilities to be effective team members.

2 Deliverables

By the end of the Spring semester, we plan to have a flow chart and wiring diagram for all hardware integration as well as a rough prototype for our application which demonstrates the primary functionalities which we plan to bring to completion by the end of the second semester. By the end of the fall semester of 2017 we plan to have the capability to both create and save user profiles, use head tracking technology to control a 2-axis camera gimbal, view a 2D map in 3D, and show where monitored machines are on the map in real time. The following lists all of the deliverables:

Server

- Authenticate connecting devices
- Handles and sets up all communication from gimbal setup and hololens

Raspberry Pi/Remote Device

- Have an assembled gimbal
- Research controlling the gimbal with streamed accelerometer data from Hololens - implement as proof of concept if have time
- Transmit GPS data from Raspberry Pi to server
- Transfer video output from Raspberry Pi to server
- NOTE: The Hololens will not communicate directly with the Raspberry Pi nor vice versa - instead all communication will pass through the server for both directions

Hololens App

- Draw a scrollable 2D map in visual space
- Display simplistic 3D models to represent monitored vehicles on 2D map
- User authentication on server

3 Design

3.1 System Specifications

The specifications for this project are outlined in the deliverables section. Our goal for this semester of the project is to have a set of applications that are able to act as an intermediary between the gimbal and the HoloLens. The specification that we have identified revolves around the three main components of this project: HoloLens app, server(s), and gimbal. What we have identified for the server is to be able to login and store multiple users. The specifications that we have for the gimbal are mainly related to the type of microprocessor that we choose to control the gimbal with. For this semester, we have chosen to use the Raspberry Pi for its flexibility in processing, variety of features, and low cost.

3.1.1 Non-functional

- The software should be able to be adapted for many different purposes and inputs.
- The software should be secure. Unauthorized devices should not be able to feed false data nor should they be able to view device data.

3.1.2 Functional

- Ability to view monitored machine on map using Hololens app
- Ability to view a custom area map on the Hollens app
- Ability to asses machine status from Hololens app
- Ability to control gimbal/camera with Hololens app

3.1.3 Standards

Workers who are being observed may feel like they are being intruded upon. This feeling of intrusion may come from the fact that there is a camera in cabin and also data on their workday is being recorded. However we don't believe that this will be a critical concern because data will only be stored only in emergency situations i.e. car crashes.

3.2 Proposed Design/Method

As we have discussed earlier, the main three components for this project are the: HoloLens app, the server, and the gimbal. Each of these systems are meant to communicate with each other and work in unison.

The Design Method that we have chosen to pursue for this project is to have a 3 axis gimbal, and use a third party microprocessor to connect to the HoloLens, and to use the accelerometers of the HoloLens to control the motors of the gimbal. Additionally the microprocessor should also oversee the video streaming of the camera. It should be taking the view from the camera and sending that data to the server so the the HoloLens can access and view what the camera sees.

The server will be the backbone of communications between all the devices in the system. Since having devices detect each other among a sea of devices or perhaps behind a firewall isn't optimal, this is the best solution. Acting as a bridge between the observer and the observee, there must be a degree of differentiation between the two. Because the observed devices are located in remote locations that may not be physically secure, these devices should have limited access to the system and should only "offer" data to the server. The server should only consume the data for the purposes of passing it on to the HoloLens.

The HoloLens application will be set up using Unity. Unity offers the best support for the HoloLens being the officially recommended engine upon release. Installing third-party software along with integrated versioning makes it an ideal candidate for development. Along the lines of third-party software, some excellent sub-solutions for features are only available via Unity. Using the engine, we will display a map on a physical surface and use it to pinpoint the locations of various remote devices.

3.3 Design Analysis

We have created a basic class diagram for the server. This diagram is likely to be revised as we progress through our project. We have also generated a basic wiring diagram for our camera-to-raspberry pi setup as well as a list of the parts needed for this. Finally, we have set up a server and a basic GUI.

For the work that we have accomplished during this semester, we have found out that there are some discrepancies in the frequency values that we input into our code and what the Raspberry Pi actually outputs. This ended up being a minor issue, but once we found a good range of frequency values for our code we noticed the gimbal started to work better. Through the course of this semester we have found that our primary choice of a gimbal was not necessarily the best, yes, it allowed for us to not worry about stabilization since it came with its own stabilizing software, however the functionality of its motors were very lacking. With the knowledge we have gained from working on this project we are confident in finding a more optimal gimbal to use. In terms of the overall functionality of the gimbal system, all parts work properly individually and when implemented together there is very few problems.

Our choice for server software needs to be revised or additional libraries need to be purchased as the Unity engine has limited networking options built-in for security concerns.

4 Testing/Development

4.1 Interface Specifications

The main application will be run on the Microsoft HoloLens. A corresponding unity project and C# application will provide the 3D graphics engine that renders the GUI and map application correctly from the user's point of view. This application also makes API calls to the server which will primarily serve to establish a relay of information to/from the Raspberry Pi. The Raspberry Pi will interface with several external hardware components. An external GPS module will be connected in order to gather accurate GPS data to be sent to the server. The Raspberry Pi will also communicate to the gimbal motors their position during runtime through an external motor controller. A camera will be used to stream video to the server via the Raspberry Pi. Finally, a bluetooth module will be used in the event that CAN bus information is integrated into the project plan next semester.

4.2 Hardware/Software

The software and hardware segments will be tested independent of each other for initial tests because the planned milestones to be accomplished by then do not support a full system. The software tests will consist of only the server and HoloLens software while the hardware tests will consist of the Raspberry Pi and the gimbal's motor controller.

4.3 Process

Each component and feature will be tested individually. For the HoloLens app, there will be a suite of tests to determine whether all deliverables for this sprint are complete. Connection to the server will be tested via scripts run within the editor. Other functionality will be tested on the actual HoloLens such as mapping and its scrolling functionality. The app should be able to request data from the server using any method and format and be able to update a map marker such that no two unique responses have the same position on the map.

The planned method for the Raspberry Pi is to run a basic script to locally control the gimbal using a gyroscope and another script to calculate GPS location. The expected results are the gimbal moving in sync with the tilt of the gyroscope and a reasonably accurate location for the device.

The server is tested using browser requests in combination with the connectivity tests for the HoloLens and Raspberry Pi. Google's Chrome Inspector built into their browser is the primary testing utility.

5 Results

On the HoloLens, the testing scripts work fine and are able to manipulate objects using data from the server. The map is scrollable and zoomable and also displays markers in predetermined positions. The Pi also successfully rotates the gimbal in sync with the gyroscope and also provided a position about 2 meters outside the room it was placed in. The corresponding server tests were all successful with both devices being able to submit authentication requests and send/retrieve data.

6 Conclusion

The work that we have accomplished so far has laid out the ground for us to start building and testing systems to see what will work best for our project. These systems will be parts of the goals that we have set for ourselves: a server that connects our mounted monitoring system and our HoloLens app. The best way to build and test our system is to build our 3 main components separately and test to see if they work on their own and then work to connecting them together.

After completing this initial semester, we have been successful in accomplishing the goals and deliverables that we had set. We are capable of controlling our gimbal system with a gyroscope and are able to control the camera and send recorded video data to the server. Our hopes for next semester for the gimbal system is to now be able to continuously track our Raspberry Pi while it is on the move, be able to continuously stream video to server and final output gimbal control from a 3rd party gyroscope to the gyroscope that is inside of the HoloLens.

The HoloLens application was also successfully completed with each part of its functionality shown through various tests. Communications with the server have been successful along with displaying a functioning world map. In the coming semester, the work will be focused on integrating all the features into one solid application and expanding on the user experience and quality of life.

7 References

Server Class Diagram: <http://dec1714.sd.ece.iastate.edu/docs/ServerClassDiagram.pdf>

System Block Diagram: <http://dec1714.sd.ece.iastate.edu/docs/BlockDiagram.pdf>