Winter Term Retrospective

Group 09: Handheld AR Device

Jacob Sowanick

Christopher Miyasako

Lane Thompson

Section 1: Introduction

The purpose of this paper is to provide a retrospective view of the Handheld AR project for the Winter term of CS 462. We will be looking at the progress over the previous two terms to consider what we could have done differently, both with our current constraints and if we were able to change the constraints given to us.

The Handheld AR project that we have been developing is a device that allows the user to scan a marker, and then would project data onto a surface for the user to see. As an example, someone who is working on a building would be able to scan the marker on a certain wall, and then could point to various places on that wall to see a representation of the wires, pipes, and studs underneath.

This project is currently being designed with guidance from Dr. Joseph Louis, and the current plan is to develop this to aid the construction workflow in order to make construction projects much faster and easier. The original plan for the device was for it to be a handheld device that would contain a camera for scanning the markers, a raspberry pi to handle all of the computations, an IMU board that would measure any movements or rotations that happen as the device moves, and then finally a projector that would be used to display images out onto surfaces that are being pointed towards. More details about how each aspect was planned to work and how we've handled each aspect at this point will be discussed in our next section.

Section 2: Tools and Technical Approaches

In this section, we are going to discuss the different tools and approaches that we've used with this project in further detail. We plan to cover both the tools and components for the device, like the camera or projector, as well as the tools for our workflow, like the online repository and the various ways we've communicated with each other. After discussing the tools themselves, we will also discuss what we would have changed or looked into differently if we were to start this project over again.

Online Communication - Our team communicated mainly through two different applications. The first one was discord, where we would reach out to each other and discuss our plans. Discord was very useful for us because it was something that we were already familiar with and used a fair amount, so we would be able to reliably see the messages on there. The second tool that we used was GitHub. GitHub is an online repository, and we have used it to collect and share all of our code with each other. We would also use other applications like Google or Zoom to help communicate with each other or with our project partner as needed.

Raspberry Pi - The project primarily relies on its computation through a Raspberry Pi, which is a lightweight and simple computer, which all runs on a board that can fit in one's hand. The Raspberry Pi that we used for this project was the Raspberry Pi 4 model B. Setting up the Raspberry Pi and making sure it would work for this project took a significant amount of time, but once we were able to get everything connected to the raspberry pi, it proved to be a very useful tool for us. Looking back, we would probably choose to continue with this again, as it has been very useful and there isn't anything else that would be a better choice for us.

Programming Languages - We decided to use the Python programming language for this project because it is a very high level language that we are all very familiar with. Each of us have used Python for projects in the past, and it has libraries that work specifically with Augmented Reality. It also has a library for the Augmented Reality markers that we've decided to use. Because the raspberry pi is basically a small computer, we are able to use the Python language to develop and test the programs directly on the device.

Camera - The camera that we used for this project is the Raspberry Pi Camera Board V2. We chose this camera because it plugs directly into the Raspberry Pi and works well with it. It's also very lightweight, but it still delivers a clear enough picture to scan the scene correctly. It took some time to configure the camera correctly, but once it was set up, we had an easy time accessing and using the camera.

Augmented Reality Markers - In order for the handheld device to successfully scan a scene and project an image into the correct place, we chose to use AR markers that would be scanned first. The other option would be to simply scan the wall and use some sort of computer vision algorithm to locate where the camera is pointing, but we realized that would be much more difficult than what we would be able to achieve in a reasonable amount of time. First, we looked into using QR codes as markers, because they are very well known and can be created easily. They are also very useful because you can include any data that you may need since QR codes are essentially an encoded string of text. The downside is that the markers have a lot of small detail in order to contain all of the data, which makes them hard to scan for. Instead of using QR codes, we recently found ArUco markers. These have much larger grain to them, so they are much easier to scan, but they need to be created specifically from the ArUco program. Another benefit to using ArUco markers is that they have Python libraries that work specifically with Augmented Reality. If we were able to start this project, we would look into ArUco markers or similar markers from the beginning.

Inertial Momentum Unit - We received an IMU device for the device, but we have not yet been able to apply it to the project. An IMU uses multiple devices like accelerometers to keep track of the position and orientation of the device. This would eventually allow us to scan a marker only once, and then be able to move to different areas while continuing to display the image onto the wall. At our current pace, we are not sure if this will fall into the scope of this project. We may need to attach this to our Raspberry Pi by soldering it, but we have no experience with soldering and don’t have the tools to do it.

Projector - The original planned scope for this project would be to eventually project a graphic onto the surface, but we do not believe that it will be applied to our scope. We currently have not found a projector that would work well for this situation, and have a couple concerns that we would have to tackle in order to get an appropriate projector. First would be the size. We would need a projector that could reasonably be held with one hand, otherwise the project would be pointless. The second thing we would need to consider is the attachment. Much like the IMU, we would need something that we could connect to the Raspberry Pi easily enough. The last thing would be the strength of the light. If we were able to find a projector that is small enough to be attached and held in one hand, we are concerned that it would not be powerful enough to be seen.

Section 3: Project Management

When we were approaching the tasks of the project, what we did was set up broad functionality goals that we would all attempt to accomplish. For example, one goal was to use augmented reality markers to get the raspberry pi to recognize specific parts of the visual input. This was completed using ArUco markers that worked really well with our goal of being able to use computer vision to recognize specific parts of the visual input taken in by the camera. The ArUco markers are much better suited for accomplishing the goals of the project in that it is easier to use them to tell the position of the camera and where specific points are in relation to the ArUco markers. Knowing this, the task of getting some of the functionality for the computer vision portion of the project working would be much easier as ArUco markers would have been the focus rather than QR codes. If there was time wasted then it would be due to the focus that we had on using QR codes to implement the computer vision functionality rather than finding an alternative that better fit our needs.

It might have been more beneficial to our progress to split up the tasks into even smaller components rather than focusing on broader goals. This could also have been a place where time was wasted focusing on a single broad goal in order to progress. There was a bit of difficulty in integrating the code that would receive visual input from the camera and the code to recognize AR markers within the visual input, but it was quickly handled. A lot of the resources and knowledge that were needed during the course of this term were acquired through the internet on different websites that covered topics such as ArUco markers and homography. This term was mainly spent on implementing computer vision functionality and researching and learning more about that, as well as homography in order to help with other parts of the project. Homography would particularly be important for finding the orientation of the camera module in relation to the augmented reality markers.

Section 4: Advice and Recommendations

As a recommendation to our project partner, Dr. Joseph Louis, we would say that, if he is able to select students by their skillset, to choose students with a broader knowledge base. As it is currently, all team members are CS students with applied path of simulation and game design. While that skillset is very important for this project, none of us had knowledge in Raspberry PI, computer vision, or soldering. Any of these would have been useful in keeping our progress from stagnating. We were able to continue our project, but only after large delays and by dropping certain features from our project scope.

Conversely, we could instead recommend understanding our abilities better when creating the scope. Computer vision is a difficult concept for people with no training in it. If the project was meant only to pick up markers in the environment and then create a graphic in response it would be easier to complete the original scope. Then the project could be passed on to finish the project.