Haptic Enabled System for Augmented Reality

ECE4012 Senior Design Project

Section L2B, hapticAR Team

Project Advisor, Dr. West

Adwait Lele *adwaitlele27@gatech.edu*, Team Leader

Abraham Kancherla *akancherla3@gatech.edu*, Expo Coordinator

Varun Nambiar *vnambiar7@gatech.edu*, Documentation Coordinator

Samir Vedantham *svedantham3@gatech.edu*, Webmaster

Submitted

May 4, 2017

EXECUTIVE SUMMARY

hapticAR is an augmented reality system which aims to change the way users interact with the virtual world. Enabling haptic feedback, to the users fingers, this system immerses the user in the AR experience. Haptic feedback attempts to simulates touch, a component of UX design which many Augmented Reality (AR) and Virtual Reality (VR) solutions today lack. Any haptic feedback, if it exists at all, is often on the headset, which does not provide a life-like experience to the user. The design of this project will entail the development of haptic feedback enabled finger sleeves, an embedded device which provides processing power and Bluetooth connectivity, and an Android application which demonstrates three applications to the user.

This project utilizes Google Cardboard as a carrier system for the headset. To detect the orientation of the finger, an Android application reads markers that are on the index finger sleeve. The middle finger sleeve has a button which acts like a mouse so the user to interact with virtual objects projected into the virtual world. The finger sleeve uses an Arduino Uno to interface with the haptic motors, the button, and a Bluetooth module to receive and send information to the phone. To give sample use cases for our project we have created three demos: Block Stack, Paint, and Drag & Drop.

The cost of this project was $110.76, and the overall money spent to build this product was well under budget as per the allotted funds for the team. The outcome of this product received acclaim at Capstone Expo. The demos worked seamlessly with the system, and the judges enjoyed the experience. Several finger sleeves were constructed in order to accommodate users with larger or smaller finger sizes.

Table of Contents

Haptic Enabled System for Augmented Reality

1 Introduction

The Haptic Smart Glasses team proposes a design for mixed reality glasses which provide haptic feedback to the user. The system is designed to be cheaper than current industry Virtual Reality or Augmented Reality glasses. The proposed system will have two main functionalities: the ability to select colors and draw, and the ability to pick up and move virtual objects. The team requests $110.76 to fund this prototype.

1.1 Objective

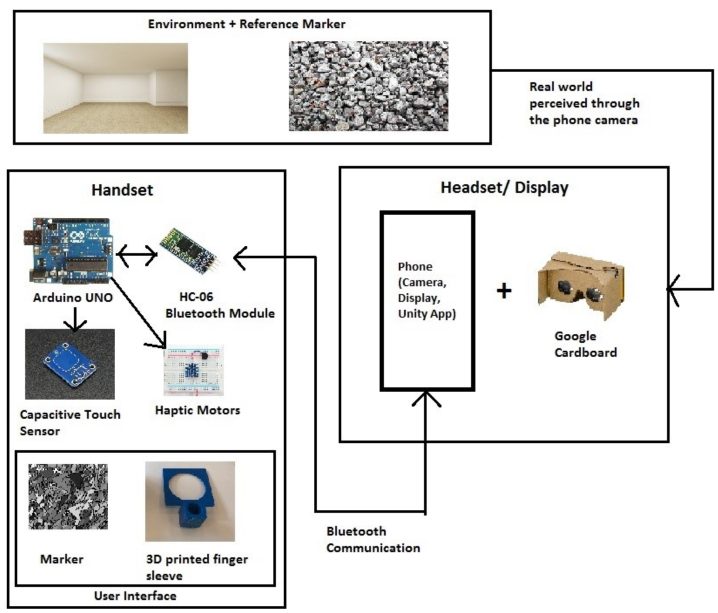


Fig. 1: Design Components.

The objective of this project is to the enhance the user experience of augmented reality headsets via haptic feedback through the users hands. The system will have a mounted headset as well as an extended device attached to the users arm which takes inputs from the user and outputs feedback in the form of vibration. The headset mount consists of the Google Cardboard AR glasses and a smartphone running a mobile application which produces the mixed reality. This prototype will showcase the ability for haptic vibration to significantly enhance the way humans interact with computers in the exciting new field of Augmented Reality. The block diagram in Figure 1 shows an overview of the technology stack.

1.2 Motivation

The inspiration for this project comes from a trend in the consumer electronics industry currently; many devices are now featuring haptic feedback to enhance user experience and overall usability. Devices such as game controllers vibrate when certain levels or tasks in a game are completed. The current iteration of the iPhone has replaced the home button with a haptic motor which allows for a much more interactive experience. The duration and strength of the vibration can be modulated as per the function of the application running on the phone. Augmented reality glasses have applications in various fields ranging from gaming and entertainment to manufacturing and engineering. The ability to interact with the headset will be paramount in exploring the varied uses of AR glasses in the present and future. For example, customer facing businesses will have the ability to immerse their potential buyers in their product in store before the purchase. This will significantly change the retail industry and advertising. The intended audience for this project include those looking for cheaper AR solutions. Current market prices make AR and VR headsets extremely expensive for the average user. By combining the ease of access of Google Cardboard and a mobile phone, this prototype will serve as a much cheaper alternative to other solutions sold currently. This prototype will also be intended for researchers in the field of Human-Computer Interaction investigating new methods for ways users interface with the virtual world.

1.3 Background

The field of Augmented Reality and Virtual Reality has the potential to change computing over the coming few decades. There are devices on market ranging from the Oculus Rift, HoloLens, and the Samsung Galaxy Gear Headset. These solutions are not widely used and have yet to capture the total potential market for these devices. When compared to the adaptation trend of mobile phones, AR solutions are still in the infancy stage with the next few decades promising incredible technological changes. This project attempts to create UX enhancements for Augmented Reality to allow the technology to be much more immersive. Augmented Reality has virtually unlimited applications, a few of which are discussed in this section.

1.3.1 Gaming

Games are incredibly entertaining when they completely involve the player. Game developers have created better graphics, more detailed plot lines and allowed players to interact with each other. Augmented and Virtual Reality will allow the player to feel more a part of the game, bridging the disconnect between the game and the human. Having the ability to feel vibration will enhance the users experience of the game and make it feel even more realistic.

1.3.2 Advertising and Sales

When potential customers are attempting to buy furniture, appliances or redesign their home, they would like no surprises during the transaction. For example, furniture shopping today involves multiple trips to the store and comparing of colors. This process can be shortened through AR. The ability to step into a virtual version of a room or closet and replace furniture or select colors and see the changes immediately would increase customer satisfaction and would save money, time, and frustration in this process. AR solutions would completely revolutionize the way advertising and sales are carried out as compared to today.

2 Project Description and Goals

The hapticAR system will contain a headset consisting of a Google Cardboard and a mobile phone running our applications. It will also include a box that attaches to the users upper arm with Velcro straps. This box will house the Arduino Uno microcontroller, Bluetooth module, and the haptic drivers. This embedded system will connect to the third component of our system, the two finger sleeves on the users hand. One finger sleeve will sit on the index finger. This sleeve has a marker that the mobile application will track; it serves as the pointer for the user. The middle finger sleeve will hold a button allowing the user to interact with the virtual world. Both the finger sleeves will have haptic vibrational feedback.

2.1 Capabilities

- Develop Three Demonstrations for the hapticAR Prototype
  - Choose a Color and Paint with your finger
  - Stack Blocks on a Grid
  - Drag and Drop Blocks

- Allow the User to Feel Vibrations in the Fingers
  - Have Different Vibrational Waveforms for Different Activities (ex. Choosing a color vs. Dropping a Block)
- Create a Headset to view the Virtual World
  - A Mobile Phone combined with a Google Cardboard set
- Develop an Embedded Application to connect the Headset to the Hardware on the Users Arm/Hand.

2.2 Target Users

- Users looking for a cheaper alternative to current on-market AR devices
- Researchers in the field of UX and Human-Computer Interaction
- Customers seeking a more immersive experience for AR.

The projected cost of materials for this prototype is $110.76. The cost breakdown is explained in Section 7.

3 TECHNICAL SPECIFICATIONS & VERIFICATION

TABLE 1: Proposed and Actual Hardware and Software Specifications

Hardware Specifications		
Item	**Actual Specification**	**Proposed Specification**
Glasses Dimensions	160.01mm×90mm×90mm	25 cm wide
Field of View	80°	Temporarily undecided
Finger Sleeve Dimensions	25mm×25mm×35mm	Adjustable Size
Processors & Microcontrollers	Qualcomm Snapdragon 820/Arduino Uno R2	1.2 GHz ARM-8 Cortex/Arduino Uno
Number of Buttons	1 Momentary Capacitive Button	1 Capacitive Button
Camera(s)	1 stream of 1080p @ 30 fps	1 stream of 1080p @ 30 fps
Haptic Motors	2 Linear Resonant Actuators	2 Linear Resonant Actuators
Screen Display	5.1″ AMOLED - 2560×1440	4.3″ 40-pin TFT Display - 480×272
Software Specifications		
Item	**Actual Specification**	**Proposed Specification**
Computer Vision Software	Vuforia	OpenCV
UI Software	Vuforia/Google Cardboard	Google Cardboard

4 Design Approach and Details

4.1 Design Approach

Fig. 2: hapticAR in action. You can see finger sleeves, a module strapped onto the user's bicep, and an Android phone in a Google Cardboard.

The system of hapticAR can be broken down into two components: (1) the augmented reality subsystem, and (2) the haptics peripherals. As seen in Figure 2, the Android phone and Google Cardboard provide the augmented reality experience to the user. The Google Cardboard has a cutout around the camera of the phone so that it can provide a camera feed of the world to the end user. The finger sleeves and haptics module strapped onto the user's arm provide the haptics based UX and communications with the Android phone.

4.1.1 Vuforia & Unity

The two main requirements of building augmented reality demos with the capability of drawing and interacting with objects, are the ability to track a user's hand and the ability to generate objects in 3D space. We achieved this by tracking 2D image targets (markers) in a 3D world using the Vuforia Augmented Reality SDK in conjunction with Unity for graphics. Unity is a cross-platform game engine that targets the OpenGL (graphics library) API on Windows, Mac, and Linux as well as the OpenGL Embedded Systems API on Android and iOS. With Unity, it was easy to create and manipulate 3D objects without having to learn and understand how all the graphics API works on the low level. Unity also has an asset store that has packages for integrating Bluetooth into projects, which we used to communicate to an Arduino. Vuforia is an SDK that can be imported into Unity and be used for various augmented reality applications - marker tracking in our case. Since Unity and Vuforia are both cross-platform software, development of the graphics and image tracking part of the demos could solely be done on a computer before building the projects for Android.

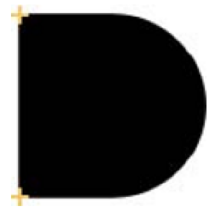
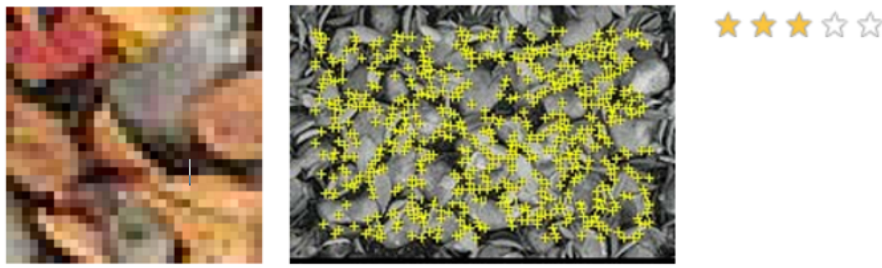


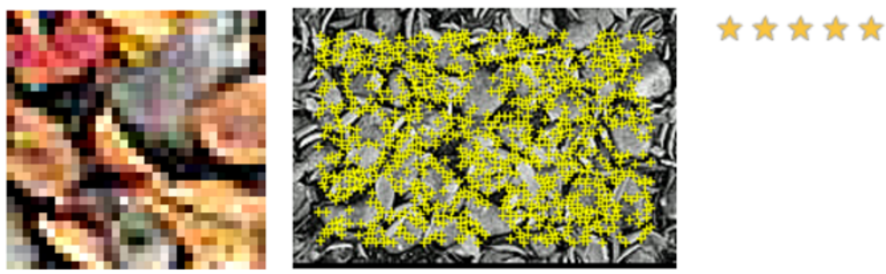
Fig. 3: Features of an image highlighted through Vuforia. [1]

To achieve consistent and reliable tracking of marker pose we needed to build a Vuforia package with good image targets. To work well with their detection and tracking algorithms, Vuforia attributes three characteristics to an ideal image target: (1) being rich in detail, (2) having good contrast, and (3) having no repetitive patterns [2]. The algorithms primarily detect features in an image, which are ”sharp, spiked, chiseled detail[s] in the image” [1]. An example of a feature is shown in Figure 3; the corners are recognized by Vuforia as features while the rounded edges are not. In order to satisfy the first criteria of an ideal image target, to be rich in detail, an image target must have a lot of features.

These features then need to be perceived by edge detection, and a way to improve that is by having high local contrast [1]. Edge detection works by detecting sharp changes in the brightness of an image so the greater the contrast, the higher the chance of the feature being detected. Vuforia's development portal allows you to see the features of sample images as well as personalized images that can be uploaded to the website. It also gives a star rating to the images, based on how well it thinks the images will be tracked using their algorithms. Figure 4 shows the features and ratings of two similar images with different amounts of contrast. After meeting the criteria of detail richness and high local contrast, the image also must not contain repetitive patterns. Even if there are a lot of features and good contrast, if the image has high rotational or translational symmetry, pose estimation will not work as well [1]. The camera may only be able to capture a few features in some frames and if these features are repeating on an image target, there are multiple locations the image target could be. Also, if the image has rotational symmetry, then the orientation of the marker could be ambiguous as shown in Figure 5.



(a) Zoomed in snippet of an image with LOW local contrast and with a corresponding low number of detected features and a 3-star rating by Vuforia.



(b) Zoomed in snippet of an image with HIGH local contrast and with a corresponding high number of detected features and a 5-star rating by Vuforia.

Fig. 4: Difference that local contrast can make on image target detection.

We used Vuforia's star ratings and guidelines for selecting image targets to help us find trackable images. There are two main targets used in our demos - one marks the reference frame in which objects are created and the other is used to track a user's index finger. The reference "stones" marker and the finger tracking "abstract" marker are both five star image targets and therefore track well. They are shown in Figure 6. Both these markers have good contrast, numerous features, and very few repetitive patterns.

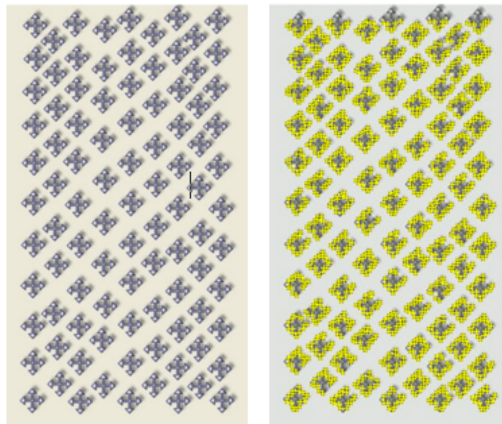
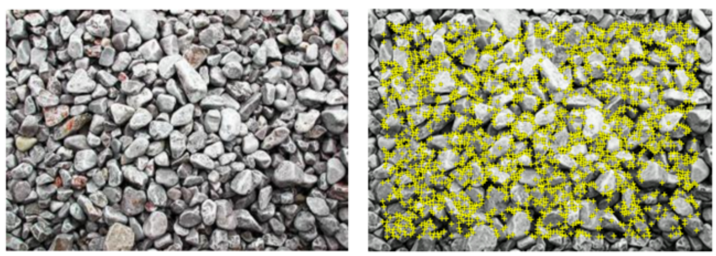
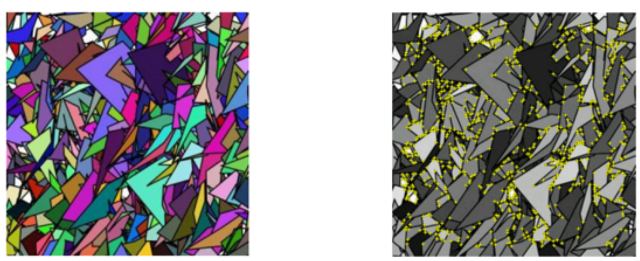


Fig. 5: Image with repeating pattern (left), and corresponding features detected by Vuforia (right). Because the image exhibits symmetry, there is no way of differentiating between the image and a 180° rotated version of the image. This makes the orientation of the marker ambiguous. [1]



(a) Image target ”stones” used as a reference marker for the virtual world and its features shown through Vuforia.



(b) Image target ”abstract” used as a marker for a user’s index finger and its features shown through Vuforia.

Fig. 6: Images we chose to use as markers in our demos

4.1.2 Computer Vision Algorithms (SURF & Optical Flow)

The Vuforia API does not disclose details about the internals of their feature tracking and pose estimation algorithms. However from understanding what the API deems as good images for tracking [1] and researching the underlying publications the API was built off of [3], one can, with good certainty, assume some of the core algorithms used for building the API.

Understanding the algorithms used in the API informed hapticAR's design and helped us understand its limitations.

A version of Speeded-Up Robust Features (SURF) is put into use in Vuforia. SURF provides a descriptor and detector to find the markers in the video stream at almost real time. The two features that SURF generates descriptors for are scale and in-plane rotation-invariant transformations. Other transformations such as skew, anisotropic scaling, and perspective effects are not explicitly handled by SURF but are partially handled due to the nature of the scaling and rotation transformations [4].

(a) Original Image **(b)** SURF Applied to the Original Image

Fig. 7: The image on the right represents the keypoints selected by the SURF algorithm for the image on the left.

To determine the scale and location of a marker, SURF uses the determinant of the Hessian matrix. For determining the orientation of the marker, SURF uses wavelet responses in horizontal and vertical directions [5]. SURF takes in the input marker seen in Figure 7a and generates the key points in red in Figure 7b. These key points are what the algorithm looks for in every frame of the video stream from the camera.

However some compromises of this algorithm end up being limitations of hapticAR's feature tracking capabilities. This algorithm can quickly find the location and orientation of a marker but is highly susceptible to changes in lighting. hapticAR required that the markers used be printed on non-glossy paper and the adhesive used to stick the marker be non-glossy as well.

Another computer vision algorithm that is most likely put into use is Optical Flow. Considering the fluidity and speed at which Vuforia recognized markers it is fair to assume that a sparse optical flow algorithm was used to bound the region in the frame for SURF to recognize the marker. Priority can be given to a region where there was motion so that SURF could quickly determine if a marker was present before the next frame is processed.

Optical flow allows a computer to understand the perceived motion of an object given a set of frames. The output of the algorithm is a 2D vector field of the image space with the vectors pointing in the direction of the motion [6]. Since Vuforia is performing this in near real time, a sparse implementation of optical flow is most likely in effect. Rather than looking at all the pixels in the image, Vuforia picks few critical points and observes the change in those pixels.

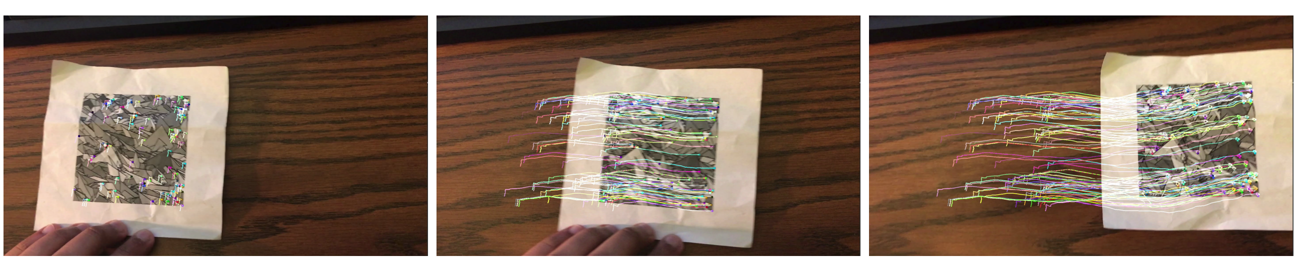
Optical flow works off the assumptions that pixel intensities remain constant between frames and neighboring pixels share similar motion. We can express this as

$$I(x, y, t) = I(x + dx, y + dy, t + dt)$$

Approximating the right hand side of the above equation would yield

$$f_x u + f_y v + f_t = 0$$

where f_x and f_y are the image gradients and f_t is the gradient along time. Different algorithms are employed to solve the u and v vectors associated with the motion of the object of interest [7]. One of the most common place almost real time algorithms is the Lucas-Kanade method.



(a) Initial Position **(b)** Midpoint Position **(c)** Final Position

Fig. 8: We see the sparse optical flow Lucas-Kanade algorithm tracking the abstract marker from the start to end frames.

As seen in Figure 8, the optical flow algorithm successfully tracks the key points of the abstract marker as it transitions from the left to the right. The multicolored lines in Figure 8c show the paths taken by each unique feature from the start frame to the end frame. By keeping track of certain keypoints on the marker, the algorithm can minimize the area that SURF needs to perform in to find the orientation of the marker.

It's important to note that as stated earlier the illumination of the scene heavily influences the quality of the tracking. In hapticAR's implementation, the lighting of the scene played a huge role in how well marker tracking performed. Using a mix of optical flow and SURF, the marker could be accurately and quickly found in the scene. We noticed that at times the marker would not be recognized if it started at a far distance but when it was brought closer to the camera and tracked, Vuforia would continue to track the marker even when it was moved back to the that same untracked far distance. This only worked if the transition from close to the camera to the far distance was visible to the camera. This is a telltale sign of optical flow working in the background.

4.1.3 Arduino UNO

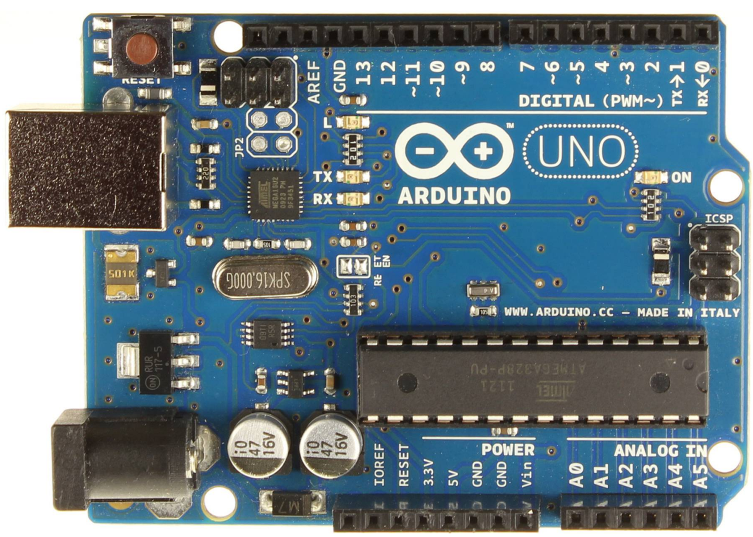


Fig. 9: Arduino UNO [8].

The Arduino UNO with the ATmega328P microcontroller controls the hardware component of the project. The dimensions of the board are 68.6 mm $\times$ 53.4 mm and it weighs about 25 grams [9]. It is shown in Figure 9. The board can be powered through multiple ways: USB (universal serial bus) connection (5 V), external (non USB) power that can come from an ACto-DC adapter or a DC battery (recommended 7-12 V), and through the power pins on the board (Vin and GND) by attaching the leads of the power source. These particular specs of the board give the device its programming the flexibility to be portable and wearable.

The Arudino UNO operates at a 5V voltage and is compatible with many peripheral devices. The board has 14 digital I/O pins. The digital I/O pins can be set to a digitalWrite() or digitalRead() mode. Each pin provides and is recommended to receive 20 mA with the maximum being 40 mA. It supports communications such as TWI (Two Wire Interface) which is I2C based, SPI (Serial Peripheral Interface), and the serial standard (asynchronous) that receives and transmits TTL (transistor-transistor logic) serial data. The UNO also has 6 analog inputs which measure from ground to 5 volts. The Arduino Software (IDE) includes a Wire library to simplify the use of I2C bus and a SPI library for the SPI communication. It also has a SoftwareSerial library and a serial monitor for the serial standard communication. The serial monitor allows simple textual data to be sent to and from the board. The board has a reset button and can also be reset through the IDE user interface which has an upload button on the toolbar. These particular set of specs and features of the UNO help incorporate many peripheral devices while simplifying the process of programming, debugging and modifying a project.

4.1.4 HC-06 Bluetooth Module

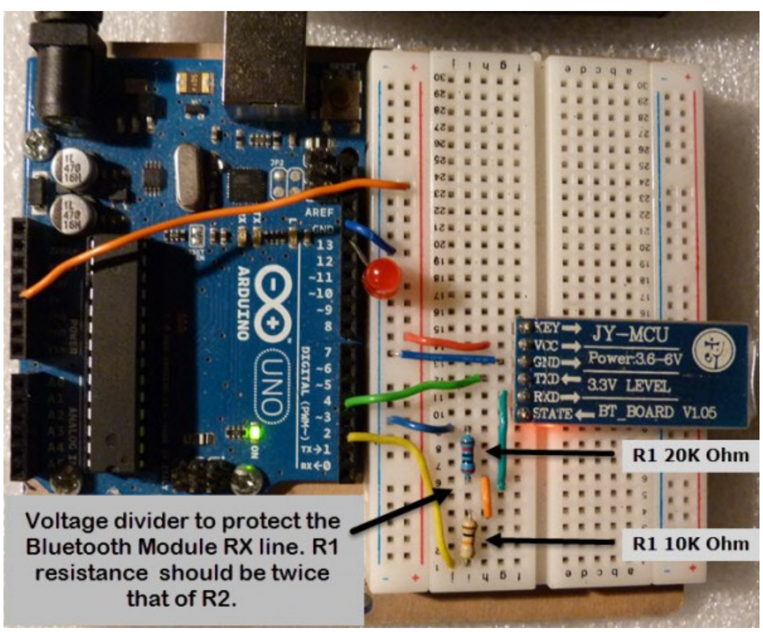


Fig. 10: HC-06 connected to the UNO with the voltage divider on the RX pin [10].

A wireless form of communication is required for the Arduino UNO to send sensor input to and receive signals from the Unity application running on the phone. A Bluetooth classic communication is used for this process since a basic android phone can support this form of wireless communication and also there are Bluetooth modules that can be attached to the UNO. The HC-06 module has signal coverage of 30 ft which is well within the range of the distance between a humans head (the phone will be placed in the google cardboard headset) and a humans arm (the module will be placed in the handset). It operates at the 3.3V logic level and draws a current of 30-40 mA while pairing and 8mA while communicating [11]. The mean current draw is about 25mA. The electrical specs are compatible with the UNOs. The UNO has a 3.3 V power pin and the use of a voltage divider can be used to step down the voltage on the connection between the TX pin on the UNO to the RX pin on the HC-06 module to mitigate any regulatory issues since all the pins of the UNO operate at 5V. This configuration can be seen in Figure 10. The dimensions of the module are 4.3cm by 1.6 cm by 0.7 cm and it weighs about 3 grams [12]. This lets the module to be in an enclosure with the UNO on the users arm.

4.1.5 Haptic Controller and Vibrating Motors

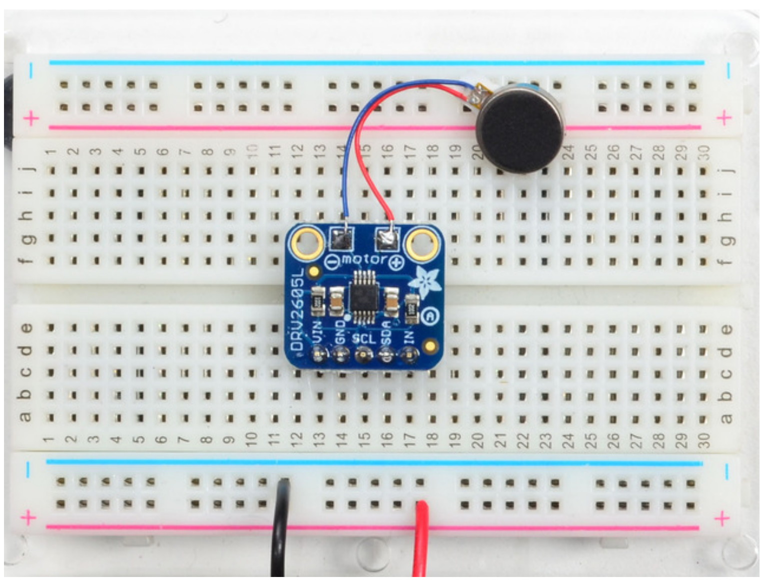


Fig. 11: Haptic Controller and Vibrating Motor configuration [13].

The haptic component plays a major role in this project. Haptics are used to improve the users experience by letting the him or her feel a light tap or heavy click on their fingers, depending on their interaction with the virtual objects. The DRV2605 haptic controller breakout board with the vibrating mini motor disc is a two-device component that can get this job done. The Vibrating Mini Motor is an ERM (eccentric rotating mass) motor that can operate at a voltage of 2-5 V [13]. It has a current draw ranging between 40-100 mA depending on the voltage its operating at. The current draw is too much for the UNO and it can be reduced through the use of serial resistors but reducing the current will lead to a weaker vibration. The DRV2605 haptic controller breakout board has the DRV2605 chip. It is especially made for haptic motors by TI (Texas Instruments). It operates at a voltage range of 3-5 V and can be controlled over I2C. The current draw is pretty low ranging from 1.9 uA to 3.5 mA. The UNOs I2C clock and data pins (SCL and SDA) are connected to the DRV2605 breakout boards SCL and SDA pins. It can be powered through the Vin and the GND pins. The IN/TRIG pin can be used to set off a haptic motor on particular type of trigger (more on this in Haptics Selection section). The haptic controller boards size is similar to that of a quarter. It is small enough to fit in the enclosure with the UNO. The haptic mini motor has a diameter of 10mm and a thickness of 2.7 mm [13]. It goes on a finger sleeve and small enough to be at the finger tip of the user without hindering any kind of finger movements. The configuration of the haptic driver and motor can be seen in Figure 11.

4.1.6 Capacitive Touch Sensor

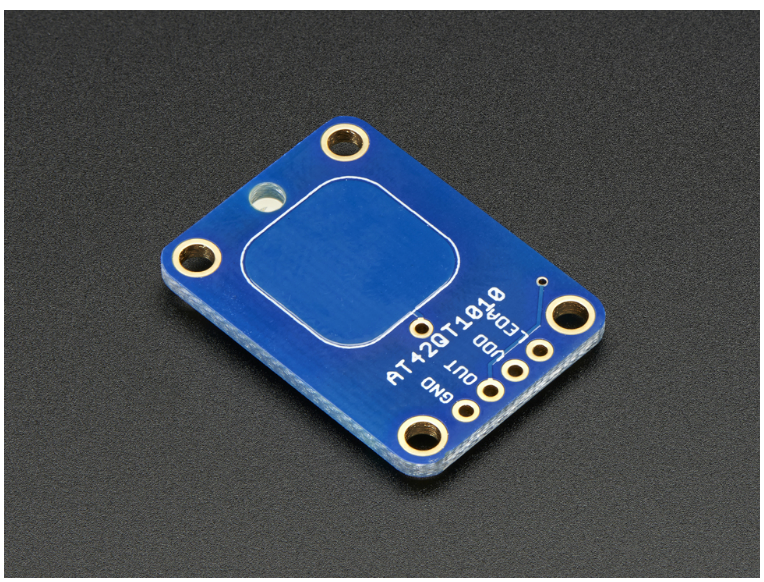


Fig. 12: Capacitve Touch Sensor Breakout Board [14].

The capacitive touch sensor breakout board (shown in Figure 12) is planted on the finger sleeve and allows the user to select, drag, drop, and interact with the virtual world. This device is one of the simplest parts of the design. It operates at a voltage range between 1.8-5.5 V and draws about 0.5 mA current. It has an OUT pin that can drive a signal of about 2 V with a current ranging between 1-4 mA [14]. These electrical specs are easily handled by the UNO. The OUT pin is activated once a capacitive load is detected (a touch). The sensor is powered through the VDD and GND pins. Since the capacitive touch sensor is very sensitive to fluctuations from its voltage source, it will be powered through the 3.3 V source rather than the 5 V source of the UNO which is connected to the haptic driver. The fluctuations caused at the 5V level because of the activation of a haptic motor can set off the capacitive touch sensor and can send an accidental touch/push signal back to the UNO. The sensor has the dimensions of 28 mm by 20 mm and weighs about 1.87 grams [14]. This simplifies the process of planting the sensor on the finger sleeve and also makes it easy for the user to move their hand around with less hindrance.

4.1.7 Powerboost 500c booster and 3.7 V LiIon/LiPoly Battery

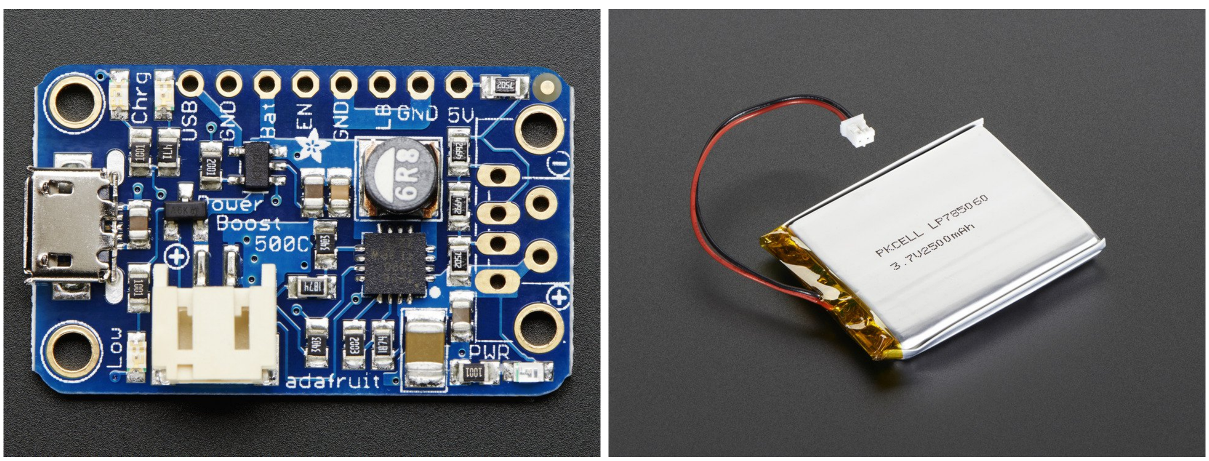


Fig. 13: The PowerBoost 500c booster (left) and the 3.7V LiIon/LiPoly battery (right) used in this project [15][16].

A portable power source is needed for the Arduino UNO so it can be powered in the enclosure. The PowerBoost 500c booster along with a 3.7 V LiIon/LiPoly battery is used for this function shown in Figure 13. The booster can convert the battery output to 5.2 V [15]. This perfect for the UNO which operates at 5V and draws a current about 50-70 mA when it is idle [17]. Given that there are two haptic drivers (3.5 mA max), a HC-06 module (40 mA max) and a capacitive touch sensor (0.5 mA max), the current draw of the overall circuit of the project should not be no more than 120 mA. Considering the RAM usage on the board for the software code which can add to the current draw, an educated estimate of 200 mA current draw by the UNO can be made. The booster can drive 500 mA at 5V. A 3.7 V LiIon/LiPoly 2500 mAh battery is the power source. The booster can draw 500 mA from the battery without any damage. The booster can also be used to recharge the battery using a basic charger cable such as the one used for an iPhone. Given the current power consumption estimate of the project, the battery should be able to power the project for at least 12 hrs without any recharge.

4.1.8 Demos

To illustrate the enhanced user experience of haptic enabled augmented reality, we created three demos: (1) Paint, (2) Block Stack (mini Minecraft), and (3) Drag & Drop. Each of these demos is an Android application that connects to the enclosure and finger sleeves through a Bluetooth connection with the Arduino. The Vuforia SDK converts the demos from a regular rectangular output to a shape that with the Google Cardboard's spherical lenses, creates a rectangular augmented reality for the user. In all the demos, there are three sources of input data external to the Android phone. The first two are the stones demo world reference marker and the abstract finger sleeve marker. These are sensed by the camera on the phone and used to determine the frame in which the demo will run and the pose of a user's finger with respect to this frame. The third source of input is a serial Bluetooth stream from the Arduino that is based on current state of the capacitive touch button as well as the pose of the user's finger with respect to the demo world frame. All three demos take in this information to change the user's interaction with the augmented world, and relay their own information back over Bluetooth to activate different waveforms on the haptic motors in the finger sleeves.

The first demo, Paint, lets a user paint onto a canvas whose location and orientation is determined by the stones marker. There is also a palette above the canvas from which a user can choose nine colors to paint with: blue, red, white, black, yellow, cyan, orange, green, and purple. The tip of the "paint brush" is determined by casting a ray from the index finger - based on the pose of the abstract finger sleeve marker - and finding where it intersects the canvas or the palette. This paint brush tip is shown to the user in the form a small white dot. If the tip is on the canvas, a user can paint on it by holding down the capacitive button on the middle finger sleeve. If the user wants to choose a different color, he/she can press the capacitive button when the tip is on that color. Figure 14 shows a first person view of the paint demo. There are also three different vibrations that this demo uses - one for hovering over a color in the palette, one for selecting a color from the palette, and while painting on the canvas.

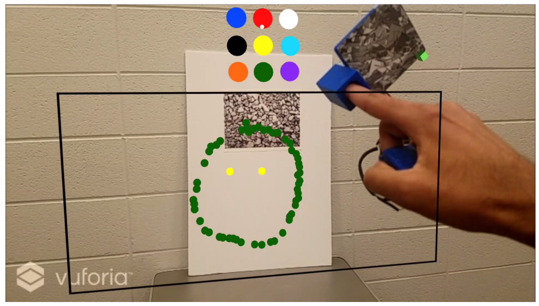


Fig. 14: First person view of the Paint demo

The Block Stack demo is akin to a primitive version of Minecraft, where a user can place and stack cubes on a flat surface determined by the stones marker. Figure 15 shows a first person view of the demo. There is a 2D grid generated that lies flat along the stones marker plane and blocks can only be placed in these discretized locations or directly above them, just like in Minecraft. Like the brush tip in the Paint demo, the Block Stack demo has a gray sphere pointer that tells the user where the ray projected by his/her index finger is intersecting the grid or objects on the grid. There is also a translucent white cube generated that tells the user where a box would be placed if the capacitive button on the middle finger is pressed. Once the button is pressed, a dark blue block is generated in that location. On placement of a block, there is haptic feedback on the index finger of the user as well.

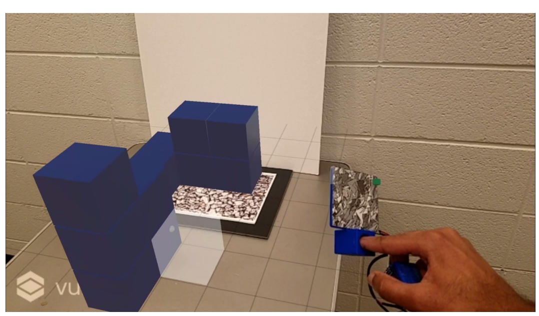


Fig. 15: First person view of the Block Stack demo

The last demo, Drag & Drop, is a basic application where a user can interact with a cube by moving it in space. Like the other demos, the cube's location and orientation is always relative to the stones marker, while the movement of the cube is based on the abstract finger marker. The user can pick up the cube by pointing at it with a gray spherical cursor and holding down the capacitive button on the middle finger. Then while holding it down, the user can move the cube wherever in space and can ”drop” it by letting go of the button. The location at which the button is released is where the cube will stay with respect to the stones marker. There is a white highlight around the cube when it is hovered over or selected. This demo also has two types of haptic responses, one when a cube is selected and being moved around and one when a user hovers over the cube. Having haptic feedback when hovering over the cube is really the key component in this demo because it lets the user know that the object being pointed at can be interacted with. Unlike a lot of other augmented reality experiences like the HoloLens, Drag & Drop tells the user which objects in the AR space are ”live” and which are not.

4.1.9 Finger Sleeve & Enclosure

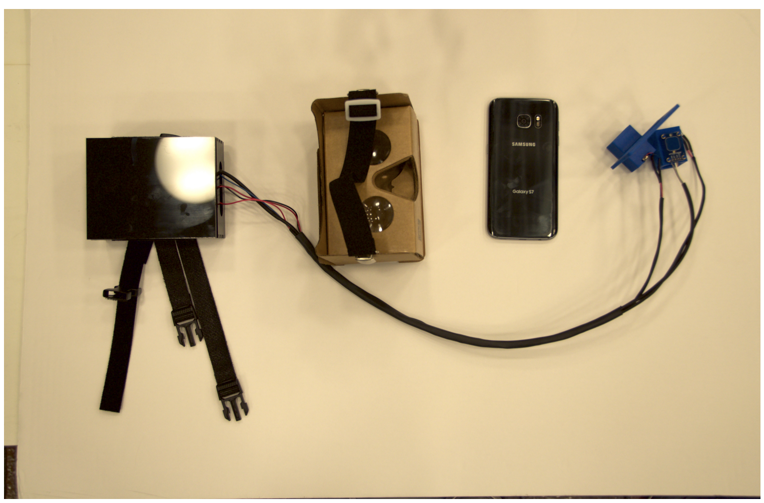
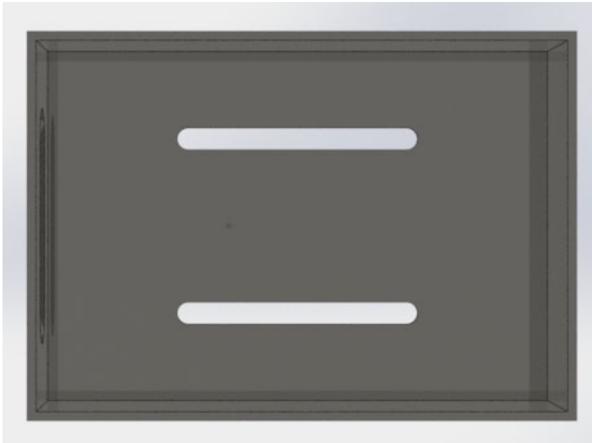


Fig. 16: The various components of hapticAR. From left to right: the enclosure that houses the Arduino, the haptic drivers, and battery, a Google Cardboard, an Android phone, and the finger sleeves with motors.

In conjunction with a Google Cardboard and Android phone, hapticAR needs finger sleeves to provide the haptics to the user and an enclosure to hold all the instruments that drive the finger sleeves. As seen in Figure 16, this was the final assembly of hapticAR. Many of the internals that drive communications and power of the finger sleeve subsystem are housed in the acrylic enclosure located on the very left of Figure 16. In the following paragraphs we will cover the design thought process behind the enclosure and the two finger sleeves.

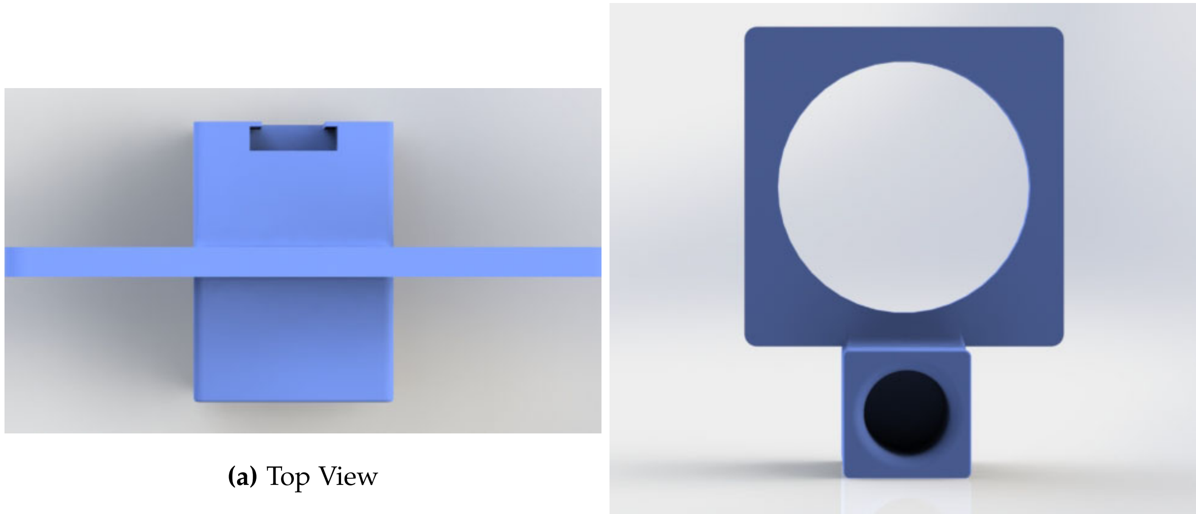


(a) Top View

(b) Side View

Fig. 17: The CAD render of the top and side views of the enclosure.

The enclosure's dimensions are 120mm $\times$ 85mm $\times$ 40mm. It was laser cut from a 3.17mm thick opaque black acrylic sheet and was assembled using superglue. The acrylic enclosure can house an Arduino Uno, two DRV2605 Haptic Controllers, a voltage regulator, a battery, and a HC-06 Bluetooth Dongle. All of these devices are then wired on a mini-breadboard in the enclosure. We decided to keep it simple and not design a breakout board due to time constraints and last minute design changes. In Figure 17a, we see two slots at the bottom of the board to provide a loop for the velcro straps to go through. To seal the enclosure, there's a lid that can be fastened with velcro. In Figure 17b there's a slot for the wires from the Arduino to connect to the haptic motors and capacitive button on the finger sleeves.



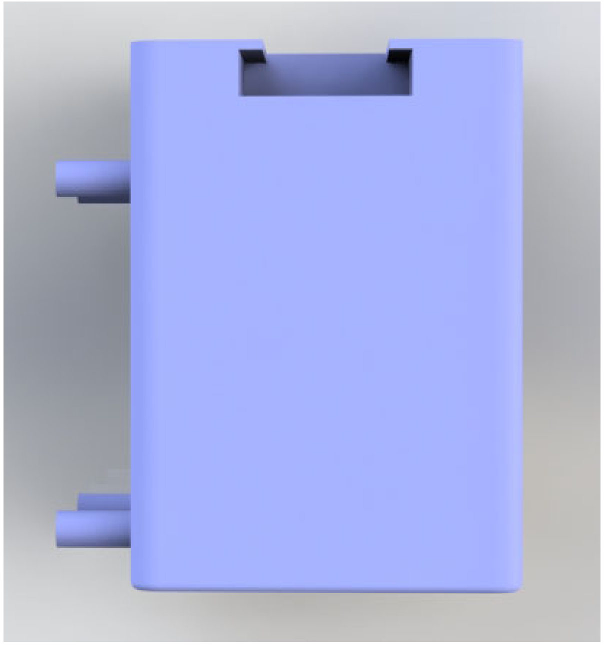
(a) Top View

(b) Side View

Fig. 18: The CAD render of the top and side views of the index finger sleeve.

Dimensions of the index finger sleeve are roughly 25mm × 25mm × 35mm. The marker support frame on the top of the sleeve has dimensions of 64.77mm × 64.77mm × 3.175mm. A top view of the marker can be found in 18a. An abstract marker was pasted on the support frame using glue. The sleeve was 3D printed on an Afina H800 using Acrylonitrile Butadiene Styrene (ABS) and normal print settings with a manufacturing time of roughly 1.5 hours. Having a one size fits all solution for index finger diameter was difficult so there were 3 sizes printed with 16mm, 16.5mm, and 17mm diameters. In Figure 18b, we see the 17mm version. For those that felt the sleeve was too loose, a cloth padding was provided during the demo. There's also a slot at the end of the finger for the haptic motor to snugly fit in.

Similar to the index finger sleeve, the middle finger sleeve has dimensions of 25mm × 25mm × 35mm. However since middle fingers on average have a larger diameter than index fingers the two sizes we printed for this sleeve were 17mm and 17.5mm. A side view of the 17mm diameter middle finger sleeve is visibe in Figure 19b. For demoing purposes, we also provided a cloth for those with thinner fingers. The sleeve was 3D printed on an Afina H800 using ABS with a manufacturing time of roughly 1 hour on normal print settings. There are also four mounting points on the side of the middle finger sleeve that can securely fasten a capacitive touch sensor using the precut mounting holes. A top view of the middle finger sleeve with the mounting pins is visible in Figure 19a.



(a) Top View

(b) Side View

Fig. 19: The CAD render of the top and side views of the middle finger sleeve.

We designed the enclosure and finger sleeves to have easily swappable components. The battery is mounted under the lid of the enclosure using velcro so that the user has easy access to charging it. Since our design has various finger sleeves at various diameters we made sure to have the haptic motors easily swappable and visible. The extrusion for the haptic motor housing is visible outside and allows the user to easily detach the motor if the user wishes to use a different sleeve. The wires for the haptic motors are incredibly flimsy and this open extrusion allows us to see if there's any damage to the motors.

4.1.10 Haptic Modes and Waveforms

The DRV2605 library provides several waveforms to be driven on the haptic motor. It also provides different modes for the driver. The driver can be set to have an internal trigger in which it waits for a Go command through the I2C interface to trigger the haptic motors. The driver can also be set to have an external trigger in which it waits for a HIGH signal on the IN/TRIG pin on the driver board to set off the motors. There are two types of external trigger modes: edge trigger and the voltage level trigger. The edge trigger is used in this project which sets off the motors at every positive edge of the clock signal given there is a HIGH signal on the IN/TRIG pin. This lets the motor go off as long as there is a high signal present. Overall the external trigger is important since it allows running the haptic motors independently through

EFFECT ID NO.	WAVEFORM NAME	EFFECT ID NO>	WAVEFORM NAME	EFFECT ID NO.	WAVEFORM NAME
1	Strong Click - 100%	42	Long Double Sharp Click Medium 2 – 80%	83	Transition Ramp Up Long Smooth 2 – 0 to 100%
2	Strong Click - 60%	43	Long Double Sharp Click Medium 3 – 60%	84	Transition Ramp Up Medium Smooth 1 – 0 to 100%
3	Strong Click - 30%	44	Long Double Sharp Tick 1 – 100%	85	Transition Ramp Up Medium Smooth 2 – 0 to 100%
4	Sharp Click - 100%	45	Long Double Sharp Tick 2 – 80%	86	Transition Ramp Up Short Smooth 1 – 0 to 100%
5	Sharp Click - 60%	46	Long Double Sharp Tick 3 – 60%	87	Transition Ramp Up Short Smooth 2 – 0 to 100%
6	Sharp Click - 30%	47	Buzz 1 – 100%	88	Transition Ramp Up Long Sharp 1 – 0 to 100%
7	Soft Bump - 100%	48	Buzz 2 – 80%	89	Transition Ramp Up Long Sharp 2 – 0 to 100%
8	Soft Bump - 60%	49	Buzz 3 – 60%	90	Transition Ramp Up Medium Sharp 1 – 0 to 100%
9	Soft Bump - 30%	50	Buzz 4 – 40%	91	Transition Ramp Up Medium Sharp 2 – 0 to 100%
10	Double Click - 100%	51	Buzz 5 – 20%	92	Transition Ramp Up Short Sharp 1 – 0 to 100%
11	Double Click - 60%	52	Pulsing Strong 1 – 100%	93	Transition Ramp Up Short Sharp 2 – 0 to 100%
12	Triple Click - 100%	53	Pulsing Strong 2 – 60%	94	Transition Ramp Down Long Smooth 1 – 50 to 0%
13	Soft Fuzz - 60%	54	Pulsing Medium 1 – 100%	95	Transition Ramp Down Long Smooth 2 – 50 to 0%
14	Strong Buzz - 100%	55	Pulsing Medium 2 – 60%	96	Transition Ramp Down Medium Smooth 1 – 50 to 0%
15	750 ms Alert 100%	56	Pulsing Sharp 1 – 100%	97	Transition Ramp Down Medium Smooth 2 – 50 to 0%
16	1000 ms Alert 100%	57	Pulsing Sharp 2 – 60%	98	Transition Ramp Down Short Smooth 1 – 50 to 0%
17	Strong Click 1 - 100%	58	Transition Click 1 – 100%	99	Transition Ramp Down Short Smooth 2 – 50 to 0%
18	Strong Click 2 - 80%	59	Transition Click 2 – 80%	100	Transition Ramp Down Long Sharp 1 – 50 to 0%
19	Strong Click 3 - 60%	60	Transition Click 3 – 60%	101	Transition Ramp Down Long Sharp 2 – 50 to 0%
20	Strong Click 4 - 30%	61	Transition Click 4 – 40%	102	Transition Ramp Down Medium Sharp 1 – 50 to 0%
21	Medium Click 1 - 100%	62	Transition Click 5 – 20%	103	Transition Ramp Down Medium Sharp 2 – 50 to 0%
22	Medium Click 2 - 80%	63	Transition Click 6 – 10%	104	Transition Ramp Down Short Sharp 1 – 50 to 0%
23	Medium Click 3 - 60%	64	Transition Hum 1 – 100%	105	Transition Ramp Down Short Sharp 2 – 50 to 0%
24	Sharp Tick 1 - 100%	65	Transition Hum 2 – 80%	106	Transition Ramp Up Long Smooth 1 – 0 to 50%
25	Sharp Tick 2 - 80%	66	Transition Hum 3 – 60%	107	Transition Ramp Up Long Smooth 2 – 0 to 50%
26	Sharp Tick 3 – 60%	67	Transition Hum 4 – 40%	108	Transition Ramp Up Medium Smooth 1 – 0 to 50%
27	Short Double Click Strong 1 – 100%	68	Transition Hum 5 – 20%	109	Transition Ramp Up Medium Smooth 2 – 0 to 50%
28	Short Double Click Strong 2 – 80%	69	Transition Hum 6 – 10%	110	Transition Ramp Up Short Smooth 1 – 0 to 50%
29	Short Double Click Strong 3 – 60%	70	Transition Ramp Down Long Smooth 1 – 100 to 0%	111	Transition Ramp Up Short Smooth 2 – 0 to 50%
30	Short Double Click Strong 4 – 30%	71	Transition Ramp Down Long Smooth 2 – 100 to 0%	112	Transition Ramp Up Long Sharp 1 – 0 to 50%
31	Short Double Click Medium 1 – 100%	72	Transition Ramp Down Medium Smooth 1 – 100 to 0%	113	Transition Ramp Up Long Sharp 2 – 0 to 50%
32	Short Double Click Medium 2 – 80%	73	Transition Ramp Down Medium Smooth 2 – 100 to 0%	114	Transition Ramp Up Medium Sharp 1 – 0 to 50%
33	Short Double Click Medium 3 – 60%	74	Transition Ramp Down Short Smooth 1 – 100 to 0%	115	Transition Ramp Up Medium Sharp 2 – 0 to 50%
34	Short Double Sharp Tick 1 – 100%	75	Transition Ramp Down Short Smooth 2 – 100 to 0%	116	Transition Ramp Up Short Sharp 1 – 0 to 50%

Fig. 20: The various waveforms provided by the DRV2605 library [13].

the use of different haptic drivers. This is important since there is only one I2C interface available for the UNO and using that to trigger the motors will result in both of them to run concurrently. The DRV2605 library provides about 116 different waveforms that can be run on the motors as seen in Figure 20. For a paint demo if the user tries to select a color from the virtual color palette, the waveform 24 which is a sharp tick goes off on the haptic motor placed near the capacitive sensor on the finger sleeve to notify the user of the selection. The sharp tick is supposed to resemble the feel of a button click when one uses a mouse to draw on a computer. Waveform 49 which is a type of buzz can be played on the haptic motor on the index finger sleeve as the user draws on the canvas trying to resemble the constant tactile force sensed when one draws on a wall or real canvas with their index finger. For a cube pick up demo, the haptic motor can play the waveform 27 which is a sharp double click strong to let the user know that virtual cube can be picked up while waveform 14 which is a strong buzz can be played on the motor to let the user feel that the object has been picked up. The double click serves as an alerting tactile force to really grab the attention of the user and the strong buzz resembles the constant force one might feel when picking up real objects. Several of these waveforms are incorporated into the project to give the user the best interactive AR experience.

4.2 Codes and Standards

4.2.1 I^2C protocol

The I^2C protocol is used to interface the haptic drivers to the Arduino UNO.

4.2.2 Bluetooth Classic

Bluetooth Classic is used to a have wireless communication between the HC-06 Bluetooth module and the phone running the Unity application.

4.2.3 Serial Standard (Asynchronous)

This communication protocol is used to send and receive information such as textual data from the HC-06 module to the Aruduino UNO.

4.2.4 USB-B

This is used to upload the code from a computer to the Arduino UNO.

4.3 Constraints, Alternatives, and Tradeoffs

Discuss other design alternatives you have considered and why you did not select them. What are the technical tradeoffs? In particular, include the constraints that affected your project. Include factors such as economic, environmental, sustainability, manufacturability, ethical, health and safety, social, and political where possible.

4.3.1 Hardware and Software Tradeoffs

There were two major tradeoffs that we had to decide on for this project. The first one was choosing between developing with OpenCV in Python or developing with Unity and Vuforia in C#. The benefit of developing in with OpenCV is that it is free and there is no watermark on the camera stream, but the downside is that everything has to be developed from scratch - calibrating cameras and image targets, working with rotation and translation matrices and figuring out how to actually create virtual objects. On the otherhand, using Vuforia and Unity take care of the small details and let you focus on refinement as well as adding extra functionality like Bluetooth communication easily. But Vuforia leaves a watermark on the demo. Initially, we started developing in OpenCV but realized that creating all the graphics for the demos would be difficult and less reliable that Unity/Vuforia and that we could live with the watermark.

The second major tradeoff was deciding what hardware to use with the visor: either using a phone or using a Raspberry Pi with a screen and camera peripherally attached. Since we had originally planned to develop with OpenCV in Python the obvious choice was the Raspberry Pi, because OpenCV could be installed on it whereas it would be difficult to do so easily on Android. The issue with this is that we would have needed to create an extra casing for the visor. The benefit of using a phone is that there don't need to be any peripherals - the camera, screen and processor are already packaged into one device. After we switched to using Unity and Vuforia for the demo development, this was the clear choice.

4.3.2 Finger Sleeve Design Process

Finding a good balance between design and engineering for the finger sleeves required many iterations of trial and error. We approached this problem by trying to keep the overall design as simple as possible. However engineering constraints forced us to make the design a lot more complicated than the simple sleeve found in Figures 21 and 22.

Fig. 21: Index finger sleeve design process from left to right and top to bottom: (1) the simplest finger sleeve (2) finger sleeve with flaps (3) a sleeve with the marker cube on top of the sleeve (4) a sleeve with a much larger marker cube on the top (5) a sleeve with just one marker with the same face dimensions as (4).

The initial design of the index finger sleeve was a simple rectangular prism with the markers pasted on the faces. However from tests conducted with the finger sleeve, the angle at which the camera viewed the finger sleeve made it difficult to track the marker. In addition the tape applied to the markers was too glossy and made it difficult for the camera to recognize the markers.

We then transitioned to a finger sleeve with flaps to make the markers more visible to the camera. We decided to use double sided tape and non-glossy paper for the markers but we never had a chance to test the sleeve since we decided to transition over to a Vuforia back end. Vuforia allows for different shaped targets and so we decided to use a cube target. On the cube target pictured in the top right of 21, all the faces have a marker for the camera to capture the position and orientation of the finger sleeve. However the markers were too small to be captured by a cellphone camera and we needed bigger targets. As a consequence, the team transitioned over to the larger cube target found in the bottom left of 21. The benefits of having a cube target is that the virtual world is properly layered behind the cube marker but we noticed a significant performance drop in finding the position and orientation of the marker. Another downside is that the finger sleeve became unwieldy.

Finally hapticAR settled for the bottom right finger sleeve. This provided a healthy balance of being comfortable to the user and providing accurate positional data of the finger sleeve. However the benefit of properly layering the virtual world with respect to the marker was lost.

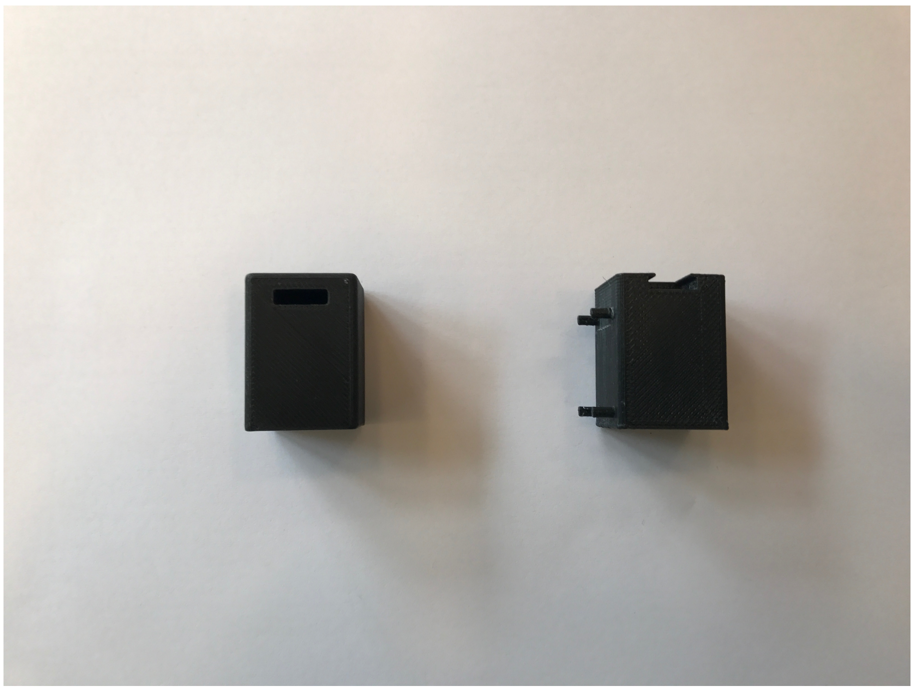


Fig. 22: Middle finger sleeve design process from left to right: (1) the simplest finger sleeve (2) finger sleeve with pins

The middle finger design roughly remained the same since it was not tracked. However from the left to the right in Figure 22 the design changes made were the slot location and the addition of four pins to holster the touch button. This provided a snug fit for both the button and the haptic motors.

4.3.3 Enclosure Design Process

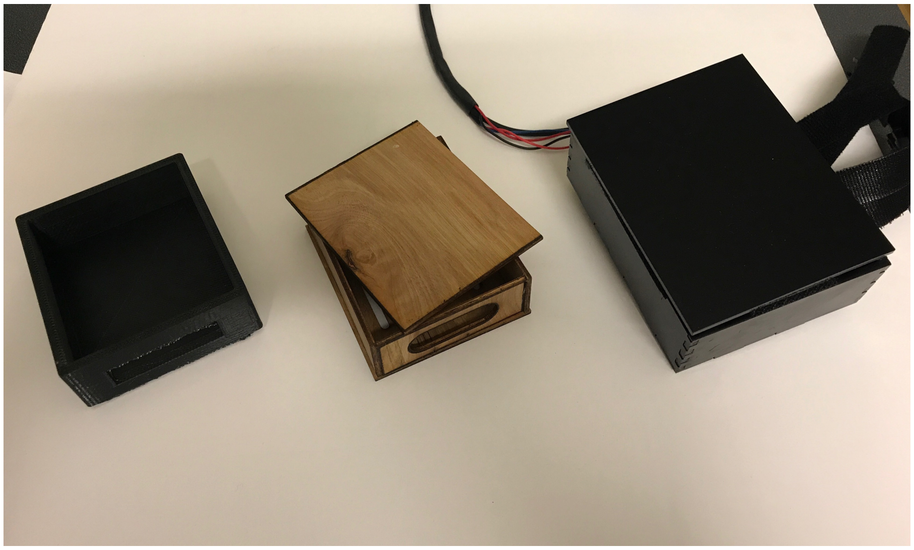


Fig. 23: Enclosure design process from left to right: (1) a 3D printed version of our enclosure (2) a laser cut balsa wood version of our enclosure (3) a laser cut acrylic version of our enclosure that is larger.

The thought process behind the enclosure design was driven more by manufacturability than ergonomics. hapticAR originally used ABS and was manufactured using a 3D printer. It took ˜2.5 hours to complete a turbo print on an Afina H800 and removing the support structure caused a lot of damage to the box. To cut production times, improve the yield rate, and reduce the weight of the device, the team decided to switch over to laser cutting the enclosure.

Our team used balsa wood for the enclosure and assembled the box using superglue. The process took ˜20 minutes. Unfortunately the finish of the enclosure did not look clean and there was not enough space to house all the components for the haptics subsystem. So in the final iteration of the device the team laser cut acrylic and super glued the pieces together.

5 Schedule, Tasks, and Milestones

The tasks that needed to be completed for this project are mainly divided into hardware development, software development, and testing. The Gantt chart in Appendix A shows the schedule for completing these tasks. The tasks are first divided into seven main groups: Planning, Glasses Construction, Finger Sleeves Construction, Development, Testing/Debugging, Oral Presentation/Design Review, and Capstone Expo. These are then further divided into specific sub-tasks that need to be realized for the main task to be fully completed.

The actual schedule followed was a bit different because some aspects of the design themselves were changed. For example, we moved on from a Lexan visor to a Google Cardboard. We also created a Block Stack demo instead of a biomedical demo which we had not planned in the beginning. There was also more testing done closer to the Capstone Expo and the Oral/Presentation/Design Review had to pushed back. Appendix B contains the division of labor for the modified major tasks from the Gantt Chart.

6 Final Project Demonstration

There were three main functional modules for this project: (1) Visor/Google Cardboard, (2) Finger Sleeves, and (3) the Enclosure. Initially, we planned on creating a Lexan visor with nonuniform lenses and mirrors that would project the image from a screen. This was later modified to a Google Cardboard headset where a phone would go instead of just a screen. The finger sleeve design changed a few times based on the trackability of the markers and the comfort of the sleeves. We tested out various sizes of markers and designed the final finger sleeves to accommodate the smallest size marker we could reliably track. The enclosure also went through a few iterations and was tested based on if the Arduino and the peripherals fit inside. Table 1 in Section 3 of this paper shows the changes in the hardware and software specifications for this project.

We demonstrated our project two times. Once before the Expo to Dr. West and once at the Expo as well. During the preliminary meeting with Dr. West, we showed him the demos with the enclosure and finger sleeves and let him test them out himself. During the Capstone Expo, we gave Dr. West our pitch for the product while using a poster and a video montage of the demos as visual aids. The main deliverables for the project were making the Drag & Drop and Paint demo with the hapticAR enclosure and finger sleeves. We accomplished this and also made an extra Block Stack demo.

During the Capstone Expo, the following items were demonstrated:

- Video montage of third-person and first-person view of a person using the Block Stack and Paint demos
- The pitch for the product itself
- Poster showing the motivation, design, and applications for our project
- Iterations of finger sleeve designs
- The whole device including the enclosure and the final finger sleeves

We used images of logos from external sources for our Expo poster. The following urls are links to the websites used to gather these images:

- Google Cardboard: https://developers.google.com/vr/
- Arduino: https://www.arduino.cc/arduino_logo.png
- Bluetooth: https://www.bluetooth.com/develop-with-bluetooth/marketing-branding/brand-guide-logos
- Unity: https://unity3d.com/public-relations/brand
- Vuforia: https://vuforia.com/
- Georgia Tech School of Electrical and Computer Engineering: https://www.ece.gatech.edu/media-resources

7 Marketing and Cost Analysis

7.1 Marketing Analysis

The final prototype will not be like other augmented reality headsets on the market. Most AR solutions do not have haptic feedback for the user on the fingers. A few products are haptic-enabled on the headset itself, but do one have finger sleeves. Furthermore, our product will be significantly cheaper than other headsets in that it is simply a mobile application which works with the finger sleeves as a pair. The headset, Google Cardboard, must be purchased separately. Augmented Reality solutions can be purchased for up to a $1000.00, whereas hapticAR will be significantly cheaper. The intended market for this product will range from entertainment to manufacturing. As this product is changing the way users interact with AR glasses, the intended market is almost limitless. There are applications in gaming, manufacturing, and engineering. This prototype can be customized and enhanced to suit specific market needs.

7.2 Cost Analysis

The total development cost for a prototype of Haptic AR is shown in the Figure 24.

Part	Cost
Adafruit ERM Haptic Motor (2)	$3.80
Adafruit DRV2605L Haptic Motor Controller (2)	$15.90
Adafruit Capacative TouchButton	$11.90
HC-06 Bluetooth Serial Module for Arduino	$9.99
ATmega328p Arduino Uno Evaluation Board	$24.99
Google Cardboard	$15.00
Adafruit 328 Li-Battery 3.7V 2500mAh	$15.30
Acrylic Casing Material	$13.78
Total Cost:	**$110.66**

Fig. 24: Cost breakdown of hapticAR device

The biggest costs of this project will be the Google Cardboard and the Arduino Uno. For the intended market, the Google Cardboard must be purchased separately. The calculated costs above show the price of materials needed to develop this project. The total cost of parts is well under the allotted budget for the senior design team.

Labor	Hours
Research and Planning	100
Headset Modification	50
Finger Sleeve Design	10
Software Development	200
Enclosure Development	25
Testing and Debugging	150
Design Review	60
Capstone Expo Review	50
Meetings	50
Total Hours:	**695**

Fig. 25: Breakdown of estimated labor cost

Assuming an average starting salary of $70,000/year, or $35.00/hour, the total labor cost is $24,325. The table below shows the estimated breakdown of hours for each category of tasks within the hapticAR project. It is estimated that a total of 695 hours will be spent on the engineering, design, and testing of this project. Assuming 30% fringe benefits of labor and 120% overhead on the materials, labor, and fringe benefits, the total development cost is shown in Figure 26.

Development Component	Cost
Parts	$110.66
Labor	$24,325.00
Fringe Benefits of Labor	$7,298.00
Subtotal	$31,733.66
Overhead	$38,080.39
Total Development Cost:	**$69,814.05**

Fig. 26: Breakdown of total development cost

We assume that a production run of 50,000 units will be sold over a 5-year period. The sale of these units will require a 10% sales expense for the sale of these units. The cost to manufacture 50,000 units is $50,107.00. The sale price is $210.00 for both the mobile app and the finger sleeves. The profit for each unit is $100.00 which is 67% profit on each unit. The total profit is expected to be $5,000,000.00 over 5 years for 50,000 units sold.

8 Conclusion and Lessons Learned

Currently hapticAR is in the completed proof of concept phase. Our team set out to showcase just how much of an improvement haptics provides to the UX of augmented reality and we think we succeeded. *Paint Demo*, *Cube Pickup*, and *Block Stack* gave glimpses into how haptics can amplify a user's augmented reality experience. When we demoed the product at the Expo the excitement in people's faces were palpable. Many industry sponsors stopped by our booth, took photos of our idea, and exclaimed that they wanted their research teams to do something similar to this. One person from Honeywell stopped by and remarked that our solution might be a better alternative to the HoloLens's finger tracking UX. From the feedback the team received we think this idea has potential to be implemented in real commercial products.

However our product needs a lot more refinement and money for it have commercial viability. For whoever wants to helm the next step of this project, it's important that the demos be related to spaces other than gaming. A feedback that really stood out was that people were not completely convinced that hapticAR could be used for applications other than gaming. Maybe the next set of applications can work in the medical space or utilities. Another piece of feedback we received was on the design of the finger sleeves. The sleeves are too large for commercial use and if it was brought down to the size of rings and wireless, the device would be sleeker and appealing to the masses. In addition, the entire product is too unwieldy. Strapping a user into it took a four man team two minutes when it should be easy for one person to put everything on.

A couple improvements can be made to the design. One particular design change would be switching from Bluetooth Classic to Bluetooth LE. The BLE consumes less energy than its other counterpart. The HM-10 module can substituted in for HC-06 to make this possible. This switch also helps to expand the project to iPhones. The Unity Asset store does provide a couple packages for BLE use on unity projects. A couple changes probably need to be made to the Arduino UNO sketch to accomodate for the Bluetooth communication change. There are online tutorials that can help with this process. Another possible change regarding the Bluetooth communication is to switch the Arduino UNO to Arduino 101 which has a built in Bluetooth. Not only does this change save energy but it will also help reduce the space for hardware component of the project. Less space would mean a smaller and lighter enclosure. This will make the product look much sleeker and would help in reducing the effort in wearing the product for the user.

To reduce the double wiring for the momentary capacitive sensor, a copper foil could be used to replace the sensor breakout on the finger sleeve. This will give the sleeve lighter and much neater look if the foil placement is properly handled. The foil could be wired (single wire) back to the capacitive breakout, which could be placed in the enclosure.

For the embedded device, the design we chose was an acrylic box to house the Arduino and Bluetooth devices. This box was chosen for its color and sleek look. We did not take into account how heat affects the material and how that would effect the embedded device. Some sort of chemical reaction occurs on the surface and the black acrylic turns white. For future projects, taking into consideration the various materials used for casings, boxes, and enclosures is of utmost importance. The look is just as important the engineering.

The second aspect of the enclosure which could have been improved is the attachment to the user. We attached the acrylic box to the arm via Velcro straps which both clicked on as well as strapped on. This was not user friendly and required a few people to put on correctly and safely. This design could be improved by taking a longer Velcro strap and just using that. This could be like the ones used by runners to attach their phones to their arms. This design would increase stability and ease of use.

One of the main lessons learned from developing the prototype is the importance of a complete and thorough research of tools and software to use in development. At the beginning of the semester there is a rush by all teams to start developing immediately. However, a more careful and deliberate planning phase would have saved an incredible amount of time later in the semester.

For example, an inordinate amount of time was was used in building the software stack using OpenCV and debating between using Python or Java. Although OpenCV is open-source and is heavily used within industry, there is a big ramp-up time. Furthermore, it requires an in-depth knowledge of computer vision algorithms. Developing with the Unity Platform did cost some money, but it was well within our budget. Unity was easier to develop in software, but it did however have a few problems with camera calibration and jitter.

All in all, the biggest takeaway is that there is no ”perfect” solution. There are always tradeoffs for every feasible piece of software and hardware. It is about finding the optimum solution which is easiest to use and gets the most done.

Aside from engineering problems, the ability to work in teams for an extended period of time without have personal issues get in the way is of utmost importance. In terms of the project management aspect of Senior Design, being organized and having the ability to effectively and politely communicate is key. This ability is required in each team member and is a pre-requisite for success in the project. An efficient method of communication and organization prevents burnout, negative emotions, and egoes from getting in the way of the project. Although we face no major issues on this front, project management could have been handled much better in retrospect.

9 Leadership Roles

There were four members on our team, each with his own leadership role. The roles and a brief description of their jobs are summarized below:

- Adwait *Team Leader*
  - Coordinated meetings and delegated tasks to team members
  - Acted as the main communicator for the team by emailing weekly reports to the advisor, Dr. Mick West, and ordering parts from James Steinberg
- Varun *Documentation Coordinator*
  - Organized the data storage for the group by managing the Dropbox, Google Drive, and GitHub repository for the project
  - Created project for design paper on ShareLaTex to make compiling and citing the project documentation easier and to be able to write the paper remotely as a group in LaTex
- Abraham *Expo Coordinator*
  - Figured out what was needed for the demo and presentation at the Capstone Expo
  - Created and helped attain a list of items necessary for the Expo, like poster easels, monitor, surge protector, etc.
- Samir *Webmaster*
  - Compiled the necessary documents for the senior design website: technical review papers, project summary form, project proposal, final design paper, etc.
  - Created the website for the project

REFERENCES

[1] *Natural Features and Image Ratings*. URL: https://library.vuforia.com/articles/Solution/Natural-Features-and-Ratings.

[2] Vuforia. *Attributes of an Ideal Image Target*. URL: https://library.vuforia.com/articles/Best_Practices/Attributes-of-an-Ideal-Image-Target (visited on 04/29/2017).

[3] Tobias Langlotz. *Publications*. Aug. 2011. URL: http://handheldar.icg.tugraz.at/publications.php.

[4] Herbert Bay et al. "Speeded-Up Robust Features (SURF)". In: *Comput. Vis. Image Underst.* 110.3 (June 2008), pp. 346–359. ISSN: 1077-3142. DOI: 10.1016/j.cviu.2007.09.014. URL: http://dx.doi.org/10.1016/j.cviu.2007.09.014.

[5] *Introduction to SURF (Speeded-Up Robust Features)*. URL: http://docs.opencv.org/3.0-beta/doc/py_tutorials/py_feature2d/py_surf_intro/py_surf_intro.html.

[6] Berthold KP Horn and Brian G Schunck. "Determining optical flow". In: *Artificial intelligence* 17.1-3 (1981), pp. 185–203.

[7] *Optical Flow*. URL: http://docs.opencv.org/3.0-beta/doc/py_tutorials/py_video/py_lucas_kanade/py_lucas_kanade.html#lucas-kanade.

[8] *The Absolute Beginner's Guide to Arduino*. Aug. 2016. URL: http://forefront.io/a/beginners-guide-to-arduino/.

[9] *Arduino - ArduinoBoardUno*. URL: https://www.arduino.cc/en/main/arduinoBoardUno.

[10] *HC-06 Bluetooth module datasheet and configuration with Arduino*. URL: http://42bots.com/tutorials/hc-06-bluetooth-module-datasheet-and-configuration-with-arduino/.

[11] *Product Data Sheet*. URL: http://silabs.org.ua/bc4/hc06.pdf.

[12] *Robot Check*. URL: https://www.amazon.com/Pass-Through-Communication-Compatible-Atomic-Market/dp/B00TNOO438.

[13] *Adafruit DRV2605 Haptic Controller Breakout*. URL: https://learn.adafruit.com/adafruit-drv2605-haptic-controller-breakout?view=all.

[14] Adafruit Industries. *Standalone Momentary Capacitive Touch Sensor Breakout*. URL: https://www.adafruit.com/product/1374.

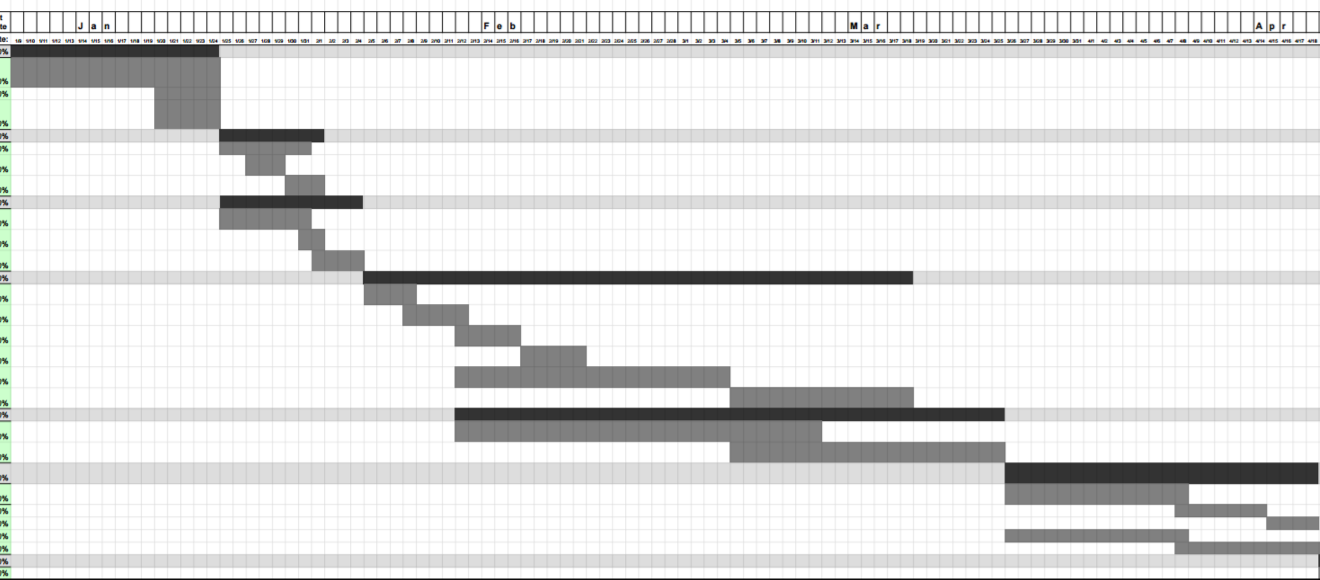
[15] Adafruit Industries. *PowerBoost 500 Charger - Rechargeable 5V Lipo USB Boost @ 500mA*. URL: https://www.adafruit.com/product/1944.

[16] Adafruit Industries. *Lithium Ion Polymer Battery - 3.7v 2500mAh*. URL: https://www.adafruit.com/product/328.

[17] *Gammon Forum : Electronics : Microprocessors : Power saving techniques for microprocessors*. URL: http://www.gammon.com.au/power.

Appendix A

Underwater Mapping and Navigation		Start Date:	January 9, 2017	
Task	Start Date	End Date	Duration (days)	Percent Complete
				Date:
1.0 Planning	2017-01-09	2017-01-24	16	0.00%
1.1 Research, Narrow Down, and Order Sensors and Microcontrollers	2017-01-09	2017-01-24	16	0.00%
1.2 Research GUI APIs to Use	2017-01-20	2017-01-24	5	0.00%
1.3 Research OpenCV and ArUco Marker Generators for Finger Tracking	2017-01-20	2017-01-24	5	0.00%
2.0 Glasses Construction	2017-01-25	2017-02-01	8	0.00%
2.1 Cut and Assemble Visor	2017-01-25	2017-01-31	7	0.00%
2.2 Attach Sensors and Screen to Visor	2017-01-27	2017-01-29	3	0.00%
2.3 Add Straps or Material to Secure Glasses to Head	2017-01-30	2017-02-01	3	0.00%
3.0 Finger Sleeves Construction	2017-01-25	2017-02-04	11	0.00%
3.1 Design and Print Finger Sleeves	2017-01-25	2017-01-31	7	0.00%
3.2 Attach Haptic Motors and Capacitive Button	2017-01-31	2017-02-01	2	0.00%
3.3 Interface Motors and Button to Arduino	2017-02-01	2017-02-04	4	0.00%
4.0 Development	2017-02-05	2017-03-18	42	0.00%
4.1 Interface Arduino with Raspberry Pi	2017-02-05	2017-02-08	4	0.00%
4.2 Develop Rudimentary GUI for Visor	2017-02-08	2017-02-12	5	0.00%
4.3 Calibrate Visor View with Camera	2017-02-12	2017-02-16	5	0.00%
4.4 Track Finger with ArUco Markers and Camera	2017-02-17	2017-02-21	5	0.00%
4.5 Develop Demo GUI with Movable Objects	2017-02-12	2017-03-04	21	0.00%
4.6 Develop Demo Paint/Draw Wall Feature	2017-03-05	2017-03-18	14	0.00%
5.0 Testing/Debugging	2017-02-12	2017-03-25	42	0.00%
5.1 Test and Debug Movable Object Demo	2017-02-12	2017-03-11	28	0.00%
5.2 Test and Debug Paint/Draw Wall Demo	2017-03-05	2017-03-25	21	0.00%
6.0 Oral Presentation/Design Review	2017-03-26	2017-04-18	24	0.00%
6.1 Develop BioMed Demo for Capstone	2017-03-26	2017-04-08	14	0.00%
6.2 Design Presentation	2017-04-08	2017-04-14	7	0.00%
6.3 Practice Presentation	2017-04-15	2017-04-18	4	0.00%
6.4 Design Paper Draft	2017-03-26	2017-04-08	14	0.00%
6.5 Design Paper Final	2017-04-08	2017-04-21	14	0.00%
7.0 Capstone Expo	2017-04-19	2017-04-25	7	0.00%
7.1 Design Poster	2017-04-19	2017-04-25	7	0.00%



Appendix B

Task Number	Major Tasks	Division of Labor
1	Planning	Samir, Abraham, Varun, Adwait
2	Google Cardboard Modifications	Varun, Samir, Adwait
3	Finger Sleeve Construction	Varun
4	Software Development	Abraham, Varun, Adwait
5	Enclosure Development	Varun
6	Testing/Debugging	Abraham, Varun, Adwait
7	Oral Presentation/Design Review	Samir, Abraham, Varun, Adwait
8	Capstone Expo	Samir, Abraham, Varun, Adwait