



Bridgewater State University Virtual Commons - Bridgewater State University

Honors Program Theses and Projects

Undergraduate Honors Program

12-19-2017

Developing a Virtual Reality Android Application to Remote Explore a Room

Jamie Nelson

Follow this and additional works at: http://vc.bridgew.edu/honors_proj

 Part of the [Computer Sciences Commons](#)

Recommended Citation

Nelson, Jamie. (2017). Developing a Virtual Reality Android Application to Remote Explore a Room. In *BSU Honors Program Theses and Projects*. Item 253. Available at: http://vc.bridgew.edu/honors_proj/253
Copyright © 2017 Jamie Nelson

This item is available as part of Virtual Commons, the open-access institutional repository of Bridgewater State University, Bridgewater, Massachusetts.

Developing a Virtual Reality Android Application to Remotely Explore a Room

Jamie Nelson

Submitted in Partial Completion of the
Requirements for Departmental Honors in Computer Science

Bridgewater State University

December 19, 2017

Dr. Michael Black, Thesis Director

Date

Dr. Li Enping, Committee Member

Date

Dr. Haleh Khojasteh, Committee Member

Date

Developing a Virtual Reality Android Application to Remotely Explore a Room

By

Jamie Nelson

Bridgewater State University
Bridgewater, MA

December 19, 2017

Acknowledgments

I would like to thank my advisor Dr. Michael Black for his guidance and support. This thesis was a dream of mine two years ago and he never gave up on me. He believed in me and my ability even when I did not. His support helped me through the difficult twists and turns of this project which taught me a lot about perseverance. It was a pleasure working with him.

I would like to thank Dr. Enping and Dr. Khojasteh for their feedback on this paper. Their input made a huge improvement. I appreciate their time and effort during the busiest part of the semester.

I would also like to thank my research partner Chris Lenoci. Thank you for always being there to answer any of my silly questions. I do not think Wall-E 2.0 would be as great without your help and expansive hardware knowledge.

Finally, I would like to thank the Office of Undergraduate Research for supporting my research. This thesis would not have been possible without the generous grants awarded to me, the Adrian Tinsley Program (ATP) for Undergraduate Research and Creative Work Summer Grant and the Semester Grant. Thank you for making this project possible.

Developing a Virtual Reality Android Application to Remotely Explore a Room

Jamie Nelson

Bridgewater State University
Bridgewater, MA
December 19, 2017

ABSTRACT

Virtual reality is a rapidly growing area of research. Many scholars are emphasizing its importance and creators are designing cheaper equipment to acquire it. The purpose of this research was to create a simple virtual reality mobile application that allows the user to explore a room. For this thesis, we created a robot and, with the help from a fellow researcher, we optimized the robot for this specific task. We programmed the robot to move forward in small increments and to spin a platform holding a Go Pro camera. The camera sends a live stream video to an Android mobile application. The video content is duplicated on the screen. The user can place the phone in a virtual reality headset achieving the immersive effect desired.

Table of Contents

1	Introduction	1
2	Exploring Immersive Images	3
3	Construction of Robot	8
4	Android Mobile Application	12
5	Conclusions	16
6	Future Work	17
	References	18

CHAPTER 1

Introduction

Virtual reality (VR) is a new hot word for this generation but what does it really mean? The Merriam-Webster dictionary defines virtual reality as "an artificial environment which is experienced through sensory stimuli (such as sights and sounds) provided by a computer and in which one's actions partially determine what happens in the environment (Merriam-Webster, 2017). Even with its recent increase in popularity, it is actually not a new topic for research. According to Merriam-Webster, the first known use of the word dates back to 1987 (Merriam-Webster, 2017). However, the essence of VR can be traced all the way back to 1838 with the creation of the stereoscope.

I began my research by compiling a complete understanding of virtual reality: the history behind VR, the psychological aspects in creating it, the hardware that makes it possible, and using that knowledge I built a software application. The psychology behind virtual reality is what really drives my passion for this subject and is the reason why virtual reality is so important to further investigate.

Two very important TED talks that discussed research in virtual reality was the basis for my research. The first described a research team that examined virtual reality and the news. They took a tragic news story of a man collapsing in a food bank line and created the actual scene as a graphical environment in VR. They then tested this scene on people and found that people had a huge emotional reaction to the news story, some even crying. The researcher pointed out that the user would be careful walking around the graphic of the man on the ground even though he was not actually there

(de la Pena, 2015).

The second TED talk was given by a film creator named Chris Milk. He discussed the virtual reality films he created. He filmed a girl from Syria that lived in a refugee camp. The story followed her as she described her family and her life in the refugee camp. The user could look around while she was talking to get a better idea of what a refugee camp really looks like. While watching the users could feel as though they were there with her at the refugee camp. The filmmakers showed the video to the United Nations and found that people were more likely to donate to Syrian refugees after watching the video than simply hearing about the refugees (Milk, 2015). With virtual reality, we are able to fully immerse the user in the experience creating more empathy.

My goal of my research into this topic is to create an application that gives an emotional response like the two mentioned case studies. But first I have to start with the basics of creating a VR application.

In this thesis, we consider how to create a basic virtual reality application. We explore the relevant history of VR and what is necessary in order to create an immersive artificial environment. In Chapter 2 we discuss how to implement this in a simple mobile application with images. Next in Chapter 3 we explain the hardware that we use to create the content for our application. Following with Chapter 4 where we expand the application to include live streaming video and motion sensors. Chapter 5 presents some conclusions. Finally, in Chapter 6 we propose possible improvements and an extension of this work.

CHAPTER 2

Exploring Immersive Images

In order to understand how virtual reality works, we must first understand non-virtual reality and how our brains interrupt what we see as the environment we are in. The main question to explore is what allows us to differentiate between the realistic world and a drawing or unrealistic computer screen. We can use that information to create an immersive environment in the mobile application.

We take for granted a lot of the visual world, including depth perception. In the article *Stereo Sue* by Oliver Sacks, he describes a woman named Sue Barry who was born without the ability of depth perception. Through testing and practice she gains the ability to see depth. She is astonished by this change and cannot imagine going back to seeing in two dimensions (Sacks, 2006). For those who can see with stereoscopic vision (three-dimensional), it is hard to understand the possibility of seeing our world in two dimensions. Our eyes and brain process images in ways that at times may seem miraculous or magical. However, this process is what helps us categorize a screen with a picture of our friend vs our friend standing in front of us.

One famous study, named the visual cliff study, performed by psychologists Eleanor Gibson and Richard Walk in 1960 highlighted this concept. Infants and babies from different species were placed on a platform. The platform was half Plexiglas and half checkerboard pattern. The Plexiglas covered the edge of the platform and below on the ground was a continuation of the checkerboard pattern. See Figure 2.1. This gave the illusion of a cliff. Almost all of the babies, no matter the species, did not move across

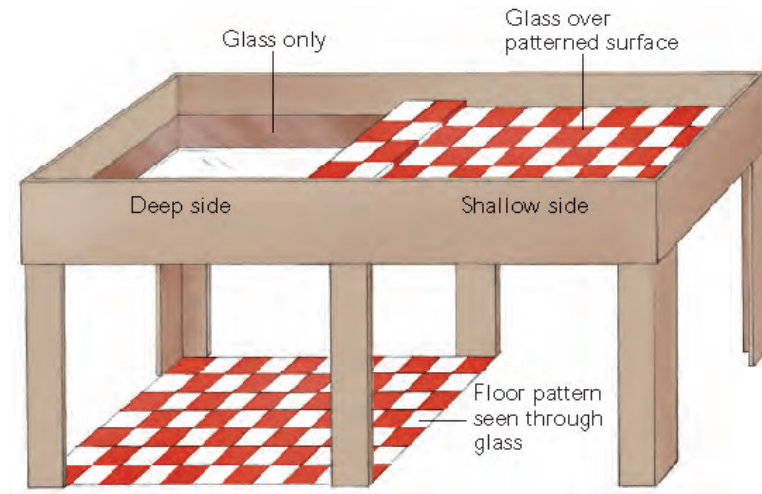


Fig. 2.1: The Visual Cliff Study

the Plexiglas. In this experiment, all of the babies relied on perceived depth of the fake cliff to navigate the world around them, specifically to make the decision to not cross the Plexiglas (Schwartz, 2016). We rely on depth perception every day to determine what objects are real. So, the next logical question is what creates depth perception and how can we create this illusion.

Depth perception is created partly from our eyes and partly from our brain. Since we have two distinct separated eyes, they each see different images of the world. Our brain combines those two images to define one vision. See Figure 2.2. This is a process we do not control and luckily all the work is done for us. However, it can be imitated by providing each eye with a different image and blocking the opposite eye from seeing the other image. Our eyes are roughly seven centimeters apart so each image has to be that distance to reflect what we see every day (VISION, 2012).

The stereoscope, created by Sir Charles Wheatstone, does just that. A stereoscope, see Figure 2.3, puts a piece of cardboard or any flat opaque material between your eyes so that each eye only sees the image presented to it. It also has two convex lenses so the images are magnified and the space between the eyes and the images is eliminated

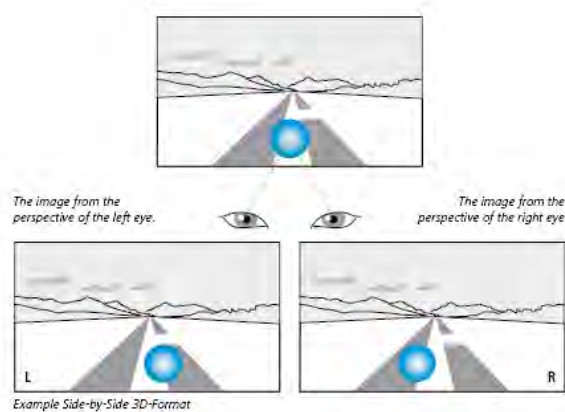


Fig. 2.2: Depth Perception

as much as possible. This duplicates what we see every day and therefore creates the three-dimensional environment (Smith, 2016).



Fig. 2.3: Stereoscope

In order to truly understand this important hardware for VR, we decided to create a simple stereoscope, see right image in Figure 2.3. The stereoscope was constructed out of a cereal box with two magnified glasses for lenses. This headset could be used with any mobile phone and VR mobile application. Google recently released a VR

application called Google Cardboard. This application splits the phone's screen into two images similar to the pictures placed in the stereoscope. We were able to use this application in our stereoscope. Google released a headset, called Google Cardboard, that is very similar to a stereoscope and allows users to put their phone into the headset causing the same three-dimensional effect (*Google Cardboard*, 2016). We soon decided to upgrade to a slightly better version than my stereoscope. Due to the simplicity of the project, we bought one with a very low cost, roughly twenty dollars. See Figure 2.4.



Fig. 2.4: Purchased VR Headset

We next studied other VR mobile applications to see how virtual reality is created by other developers. We found that there are three ways a user can experience virtual reality. The first is an all graphical interface with objects looking like cartoon drawings instead of real life video shots. This is not the immersive experience we were hoping for. The second is three hundred and sixty degree images. With Google maps, the user can go to any location in the world and look around three hundred and sixty degrees. However, it is limited to the one image. Therefore, the user can only look around and cannot move around in the space like we move around a room. The third and final currently available option is three-dimensional video. Through video technology, developers have created videos where the user is immersed in the environment and can look around three

hundred and sixty degrees while the video is playing. The one caveat is that the user does not control the movement. For the VR application we were creating, we hoped for the best possible scenario, that a user could look around a real-time video and control the movement. So, we started with the basic image to see if we could work up from there.

For the first mobile application we created, we decided to copy the stereoscope. We captured two images seven centimeters apart (see Chapter 3 for more information on how the images were taken) and placed the images side by side in the mobile application. The result was exactly as we hoped. The guitar in the image appeared three-dimensional and the artificial environment appeared realistic. See Figure 2.5.



Fig. 2.5: First Mobile Application using Images

This was a great first step in creating an immersive application. However, the limitation is that we are using a still shot. The environment is not realistic given that our world is constantly changing. We decided to improve the application by using video content instead of still images. This would require better image capture. Next, we focused on the optimization of the robot we were using to collect the content for the mobile application.

CHAPTER 3

Construction of Robot

We decided that a mobile application that explores a room would need a robot to control the camera's motion and to move around the room. We found a robot previous created in the robotics lab. The robot already had three wheels, two on both sides and one small one in the front. It had a frame that on top contained the Arduino Uno. The Arduino could be programmed using C++ in order to move the two big wheels. We took the old robot and also added a servo to the top with a platform. The platform contained two cameras approximately seven centimeters apart. The platform was able to spin by controls sent from the Arduino to the servo. See Figure 3.1.

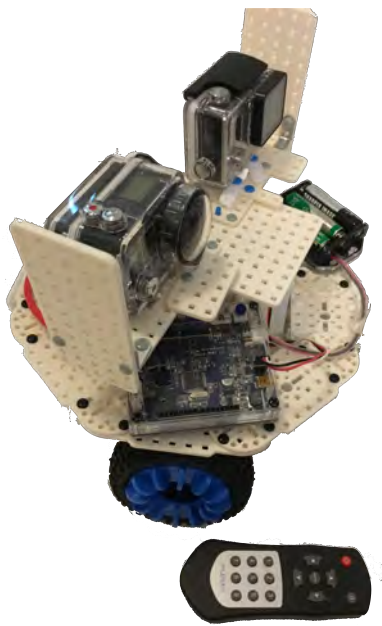


Fig. 3.1: The First Robot

This robot worked well for the first image application. However, it had many limitations when it came time to advance to an improved product. One big issue being that the robot seemed to lose power over time. The first time using the robot it would move forward at a different speed than the second time running the code. The code did not change so something was wrong with the hardware. We hypothesized that the battery was possibly dead or not powerful enough to control the two motors and the servo. There were other problems with the robot's design. The cameras were what we had available. One was a Go Pro Hero 3 provided by Dr. Black and the other was an AEE action camera. Since they were created by different companies, the lenses were in different locations. This made it hard to line up the cameras on the platform. We also had a hard time securing them to the platform at the same lens height. One smaller issue was the wheel in the front. It was prohibiting the motion from being fluid. The wheel would sometimes get stuck or would move the robot in inconsistent directions. After discussing with a fellow researcher, we decided to reconstruct a new robot to fix these issues. Researcher Chris Lenoci joined the project to help with the hardware challenges we were facing. He has a better background knowledge in hardware and translated many of the technical terms for me. His help greatly improved the project quality.

Together we agreed to construct a new robot with a more secure base. We decided four wheels would create more precise movement. We also opted for two Go Pro cameras instead of the two cameras with two different manufacturers. I had bought a Go Pro Hero 5 for personal use so that was the camera we used. Luckily, the Go Pro comes with attachments that can stick to a flat surface to secure it. We decided that would be better than creating a box to hold the cameras in place. Chris researched why the current robot was slowing down. We realized that we needed more than just an Arduino to run code to control the motors. We also needed an attachment called a motor shield.

We decided to add more battery power as well. Our new robot would need more power because we were adding two more wheels to control. We found an Arduino Uno, a motor controller shield, and a better battery all in the robotics lab. We considered buying a chassis for the robot but soon came across a more pressing issue for our proposed design.

As we were piecing together what we needed for the robot, we were also researching how to connect to the Go Pro cameras to the mobile application. The company Go Pro has its own application for live streaming to an IOS mobile phone. We were proposing to create an Android mobile application not IOS and our application needed to access that stream of data to manipulate it. We found an article online for live streaming to a computer (sdobbie, n.d.). The computer has to connect to the WiFi signal sent out by the camera. Then using the URL, <http://10.5.5.9:8080/live/amba.m3u8>, and a media player (we used QuickTime Player), the WiFi signal is converted into video data displayed on the computer. This was the only solution we found to get the content from the camera directly. This caused many problems for our project that will be discussed further in Chapter 4. The main issue at the time was that this capability is not available for all Go Pro cameras. We had a Go Pro Hero 3 and a Go Pro Hero 5 for the project. The WiFi streaming only works for version 3 of the Go Pro Hero. At this point, we decided it was best to spend our research budget on another Go Pro Hero 3 camera instead of a robot chassis.

Therefore, we used supplies in the robotics lab to construct the chassis for the robot. We added the four wheels, an Arduino Uno, a motor shield, and a battery. We also put the servo on top for the addition of the cameras once we were ready. With the last bit of our budget we also decided to buy an Arduino Primo, as we wanted the application to control the robot's movement. Since the WiFi signal would be used for sending data from the camera, we recognized we would need to use another wireless technology. We decided Bluetooth would be the next best option. The Arduino Primo

is compatible with both WiFi and Bluetooth. So that would be a massive upgrade from the Arduino Uno we were currently using that did not have either of those features. See Figure 3.2 of our final robot. We used simple starter code written in the C++ and

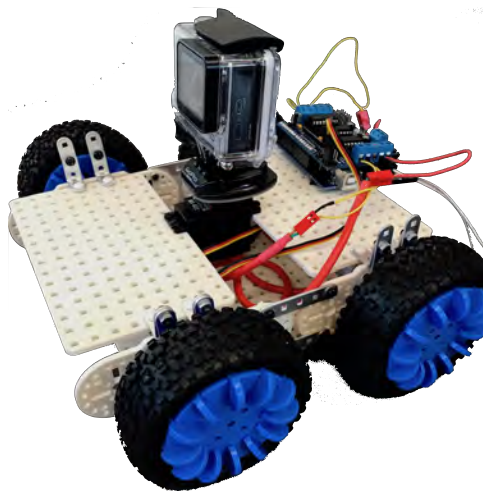


Fig. 3.2: The Second Robot

uploaded it to the Arduino Uno to make the robot move. We tweaked the code so that the robot only moves in increments of four inches and the axis spins from left to right then the motion is repeated. Once we had the robot prepared, we moved on to the mobile application implementing video.

CHAPTER 4

Android Mobile Application

The mobile application, named Robot's Life, starts at this simple icon. Once clicked, the user is taken to a welcome screen. See Figure 4.1. The video content only appears if the mobile phone is connected to the Go Pro's WiFi signal. We put the welcome page so the user does not encounter a continuous buffering message with no indication of why. The welcome screen has a simple VR background picture and a button that says, "Click When Ready". If the user clicks on the button but is not connected to the Go Pro WiFi, a toast message will appear to notify the user. Otherwise, the screen changes orientation to landscape, the dimmer is turned off so the screen will stay on while the mobile phone is in the VR headset, and the video buffers.

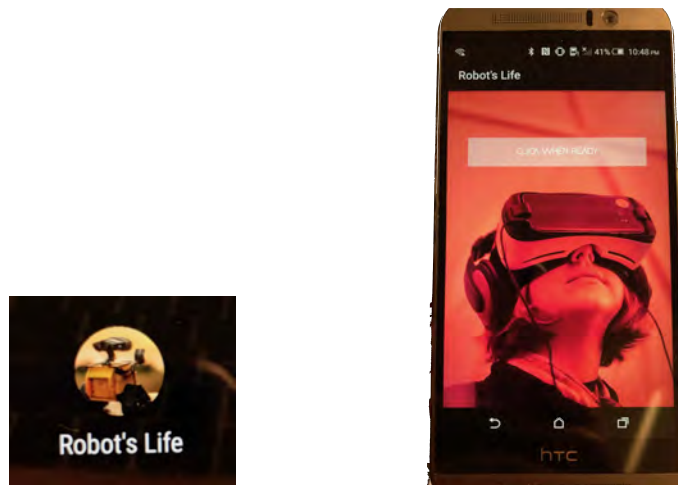


Fig. 4.1: Icon and Welcome Screen

There were complications in connecting to the camera, as previously mentioned. For the video streaming screen, we added an activity that creates a video view in the

application. Using the same URL we used on the computer to stream from the camera to QuickTime Player, we inserted that URL for the video content to be displayed. However, the phone is only able to connect to one WiFi signal at a time. We were only able to connect to the video streaming from one camera. Due to this limitation, we decided to duplicate the video on the screen. See Figure 4.2.



Fig. 4.2: Video Screen

We found a few more issues as we progressed. We were not able to alter the video since it was streaming straight from the camera to the phone. Therefore, we were not able to expand the video to fit the entire screen's height. To provide a small fix to this issue we made the background color of the application black. This made the problem less apparent but was not a complete fix. The two other large issues were that the video stream was delayed and if the robot moved too fast the video would freeze. These limitations were very disappointing but we decided to continue with the project by trying to control the robot's motion using the phone's application instead of the robot moving on it's own.

Our original plan was to control the robot by a VR controller that came with the VR headset we purchased. This controller connected to the phone via Bluetooth.

We could interpret the signal from the controller in the mobile application and then send a signal via WiFi to the Arduino Primo so it could adjust the robot's motions accordingly. Our ideal situation would be that the user could be in another state and through Internet connection could control the robot. This idea was ruined for many reasons. Mainly, by the Go Pro camera using the WiFi signal for the video content. The Go Pro camera connecting this way meant the phone has to be in a reasonable distance from the camera/robot. The VR remote was also a problem. Since it is a commercial product, we could not find any documentation on the signals the remote sends when the buttons are pushed. We discarded the remote idea and planned to use the phone's motions instead.

First, we researched what sensors are in the phone. We considered which ones are needed to sense when the user tilts the phone down and when the phone is moved from left to right. We found that the gyroscope and the accelerometer embedded in the phone would be best for our usage. We created test applications to understand what is changed in the sensor when the phone moves. The first test application displayed the x, y, and z coordinates of the gyroscope sensor. The second application was to test the accelerometer. This application moved a turtle image left and right across the screen based on the acceleration placed on the phone. See Figure 4.3. Finally, we tried to apply this test knowledge to utilize these sensors in the robot application.



Fig. 4.3: Second Test Application

Unfortunately, the Arduino Primo that was going to receive the Bluetooth signals from the application is not compatible with the motor shield that controls the wheels.

Hopefully, this is only due to the fact that the Arduino Primo is a very new product. We decided to use a raspberry pi instead. With a simple operating system uploaded to the raspberry pi, we tried to use it as an in between for the Bluetooth signal from the phone to the Arduino Uno to control the robot.

When we added the Bluetooth to the mobile application, we recognized that in the phone we were using we were not able to connect to both the WiFi and the Bluetooth at the same time. Also, adding the sensor detection to the video screen caused the application to crash. We believe, with more time, testing, and possibly better hardware, we could be able to get the application working as intended. However, we are satisfied with the current version that simply streams video from the robot's camera.

CHAPTER 5

Conclusions

Understanding how a simple virtual reality application works was a much larger task than expected. It combined the knowledge of psychology, hardware like the stereoscope, VR headset, robot motors, Arduinos, and raspberry pi, software like android development, java code for the application controls, XML for the application layouts, and C++ for the robot controls, and even an understanding of networking. Almost all of these concepts were completely new to me and the research took a great amount of effort. However, in the end we have a decent working product. We created a VR Android mobile application that streams live video from a Go Pro camera and allows the user to be immersed in the world of a robot. This is just the beginning of what a VR application is capable of and I cannot wait to see what the future hold for VR.

CHAPTER 6

Future Work

As mentioned many times in this paper, this project has many limitations. The first improvement I would propose in future work is fixing the video freezing and trying to tackle the delay problem. However, a better proposition would be using different cameras. Even though Go Pro cameras provide amazing quality, they also come with many downsides. A better camera would be one where the network from the camera can be managed easily. I would propose adding a server between the camera and application. This could allow two separate cameras to be streamed instead of just one. It could also change the video content to fit the full height of the screen. This would allow the WiFi signal on the phone to be able to not only receive data but also send data. In this way, the motion controls could be done over WiFi to the server and finally to the robot. Hopefully, it could be optimized so the delay is not as bad as the one created by the Go Pro network stream. The user could then be in another state or possibly country and see what the robot sees with the ability to also control it. In the future, I would like to propose to take this project one step further to store the video data from the robot for a room. Then, the mobile application could access that data and the user could move around the room based on the previously saved content. Hopefully, I will reach that goal one day in graduate school.

References

de la Pena, N. (2015). *The future of news? virtual reality*. TED.

Google cardboard. (2016). Google.

Merriam-Webster. (2017). *Virtual reality*.

Milk, C. (2015). *How virtual reality can create the ultimate empathy machine*. TED.

Sacks, O. (2006). Stereo sue. *The New York Times*.

Schwartz, J. (2016). *Are humans born with fears?*

sdobbie. (n.d.). *Gopro hero 3 black edition ip camera*.

Smith, J. P. (2016). *3d stereoscopic photography*.

VISION, B. (2012). *How 3d spectacles work*.