

UNIVERSITY OF CALIFORNIA, MERCED

Building Custom Real-Time Sensors for Virtual Reality Applications

A Thesis submitted in partial satisfaction of the requirements
for the degree of Master of Science

in

Electrical Engineering & Computer Science MS IGEC

by

Hanbin Tao

Committee in Charge:

Professor Marcelo Kallmann, Chair

Professor Shawn Newsam

Professor Mukesh Singhal

Summer, 2017

Copyright

Hanbin Tao, 2017

All rights reserved

The Thesis of Hanbin Tao is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

Shawn Newsam

Mukesh Singhal

Marcela Kallmann, Chair

University of California, Merced
Summer, 2017

Abstract:

In virtual reality (VR), real-time motion tracking is essential for the synchronization between virtual scenes and the real world. However, due to the limited availability of specific devices and the often expensive cost of existing tracking devices, this may prevent more people from having the opportunity to use VR technologies. In this thesis, I explore the process of connecting specific sensor configurations to a VR application in order to be able to customize motion sensors to specific applications, instead of having to rely on solutions that not often match the application needs. The MPU-6050 sensor is very accurate, as it contains 16-bits analog to digital conversion hardware for each channel. Therefore, it captures the x, y, and z channel at the same time. The sensor uses the I2C-bus to interface with the Arduino. The Arduino Ethernet Shield making use of UDP communication procedure provides us a convenient way to interact between Arduino and any desired software application. Finally, three experiments are carried out to demonstrate the application of real-time simulation under virtual environment, and the results show that this work can provide an accurate motion tracking for VR applications in real-time.

1. Introduction

Virtual reality is multisensory computer generated environments that allow the user to interact in these simulated worlds. Using different technological devices, VRs may artificially create sensory experiences in the user. These environments can be almost indistinguishable from real ones, trying to create lifelike experiences (e.g., VRs for pilot training) or fantastic imagined worlds that are completely different from reality (e.g., in video games or interactive stories). Virtual reality technology is widely used for various purposes, such as video games, engineering, entertainment, education, design, films, media, medicine and many more. Virtual reality (VR) technologies nowadays could change many aspects of our existing way of life. The objective of VR is to render the virtual object in a real-world context with an accurate posture, thus the system needs to know where the user is and what the user is looking at by computing. If we want to create a feeling of presence in a synthetic VR environment, simulating the user's posture is also essential.

1.1 Motion Tracking in Virtual Reality Applications

There are quite a few ways motion can be tracked, here I will break them down into two broad categories: optical and non-optical tracking.

Optical tracking is where an imaging device is used to track body motion. Non-optical tracking uses a variety of sensors that are often attached to the body for the purpose of motion tracking, but can also involve magnetic fields or sound waves. Many motion tracking systems use a combination of these methods for added accuracy, but as the sensors improve we are seeing systems that are more compact and accurate. Optical

methods of motion tracking usually use cameras of one sort or another. The person being tracked has optical markers. Usually dots of highly reflective material attached on certain known points of the body or on the equipment such as the HMD or handheld controllers. As mentioned earlier, optical tracking methods are usually expensive, and they suffer from occlusion. Full body tracking that rivals professional systems in terms of movement freedom and accuracy is not yet available as a low-cost consumer device. At the moment, most virtual reality solutions for consumers focus on a “sit-down experience” or a standing in place experience. Products such as the Oculus Rift are designed for this sort of experience at the moment. Many games require you to stand up and wander around while wearing their HMD, but conservatively since you’ll likely trip over something and hurt yourself if you move too much.

Non-optical methods of motion tracking in VR often utilizes micro-electromechanical sensors such as accelerometers, gyroscopes and magnetometers. The gyroscopes measure 360- degree rotation, accelerometers measure movement along the XYZ axes and magnetometers can determine orientation towards a magnetic field. Combining these two sensors can provide a device with low-latency and precision motion data, which allows us to have some interesting animations in virtual world, such as running, jumping, turning left/right, and flying.

1.2 Work proposed in this thesis

In this thesis, I describe how low-cost and customizable motion tracking solutions can be put together to address specific needs of virtual reality applications.

In the first presented experiment, I use a low-cost Google cardboard and Unity3D free software to develop a test application utilizing customized tracking. The presented experiments show how the MPU-6050 sensor works and how accurate it is. The tested results are mainly based on communication latency and the accuracy of the motion behavior in virtual world.

The second and third experiments are carried out during my internship at EON Reality last summer. We utilize the EON software and Arduino Ethernet Shield to develop featuring applications for virtual reality movies or medical insertion tube simulation.

2. Related Work

Virtual reality has made huge strides in headgear recently, but developers are still grappling with control. How do we interact with an artificial world in a natural way? There are plenty of devices to answer this: gamepads, the Sixense Stem system and even purpose-built game controllers. Leap Motion company proposes that they have a better solution by just using our hands. Then they have announced that its both creating a VR mount for its existing hardware and building a new sensor specifically for virtual reality. VR projects using leap vary in application: some mount the device on headgear, others place it on a nearby table. It creates a kind of chaos, an inconsistent experience across Leap-supported VR projects. By introducing a standardized mount, Leap's CEO Buckwald hopes to make all Leap-based VR consistent. To help facilitate this, the company is pushing a massive update to its beta SDK, enabling native top-down tracking and giving developers direct access to the Leap sensor's raw image.

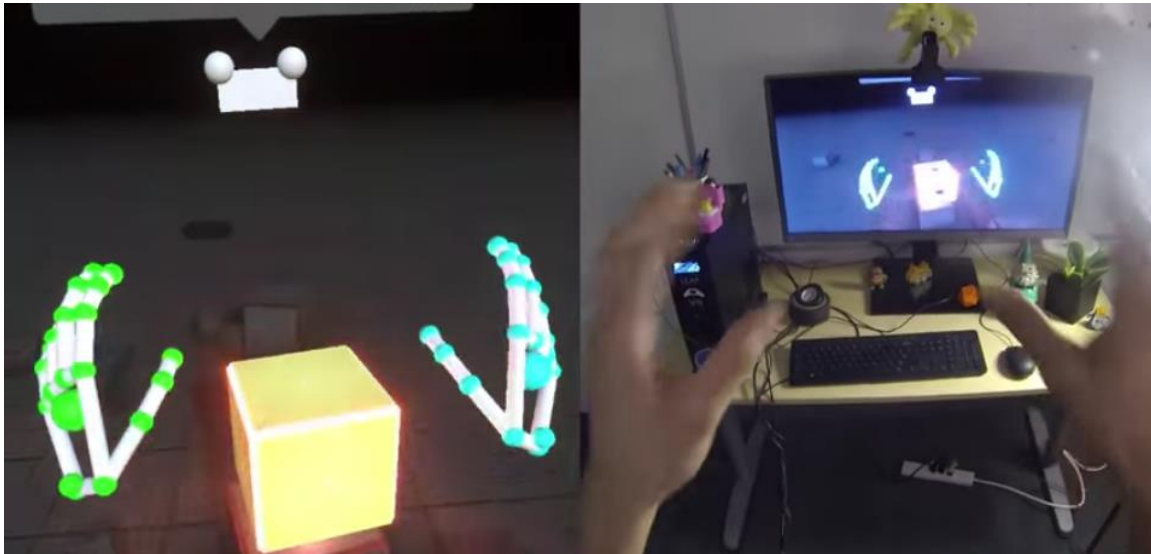


Figure 1: Leap motion VR demo

Until now, there hasn't been a way to truly immerse yourself into virtual worlds in a way that allows natural, full-body interaction with only a few sensors. PrioVR, as one example of a full-body solution from inertial sensors is changing the game by bringing movements into virtual environments where we can see our body move as we move, manipulate items as if they were right in front of us, and interact as naturally as we do in the real world. Other motion sensor tracking system, like Leap, only allows us to track our hands in virtual environment. But PrioVR uses high-performance inertial sensors to provide 360 degrees of low-latency, real-time motion tracking without the need for cameras, optics, line-of-sight, or large, awkward equipment. PrioVR's sensors are placed on key points of our body to capture your movements and translate them on-screen in real-time. PrioVR is wireless, allows for multiple simultaneous users, and will

work anywhere - indoors or out, in capture spaces of any size. PrioVR is available in three versions, the Core, the Lite, and the Pro, all of which are wireless and compatible with each other. All suits include two hand-controllers with action buttons, triggers, and joysticks. The joysticks let you easily navigate through large worlds without actually walking, while the trigger supports a familiar shooting experience. Other non-optical tracking system, like Perception Neuron is adaptive and affordable. It is very efficient for capturing level of detail and body movements as we need. It also provides us an SDK with a C/C++ API and integrations into Unity3D and Unreal Engine. Examples of Oculus Rift application are also included. Xsens is another leader in 3D motion tracking technology. Its sensor fusion technologies enable a seamless interaction between the physical and the digital world in consumer electronics devices and professional applications such as 3D character animation, motion analysis, and industrial control & stabilization. Finally, Vicon plays an important role in motion tracking market. Its software is tailored to each individual market – Nexus for Clinical Science and Biomechanics and Sports Science, Tracker for Object Tracking and VR followed by Blade for entertainment. All Vicon software uses the same core engine, bringing the benefits of Vicon's market leading motion capture algorithms, calibration, the best real time in the business, labeling and solving to everyone.

PrioVR Lite - an upper-body only suit, perfect for sitting on the couch. You can still punch, aim, shoot, lean around corners, dodge, duck, and more.

PrioVR Core - for the gamer who wants full-body tracking, including the legs.

PrioVR Pro - for the gamer who demands the most realistic experience. Adds more precision with the feet, shoulders, and hips. Can also be used for professional motion capture.

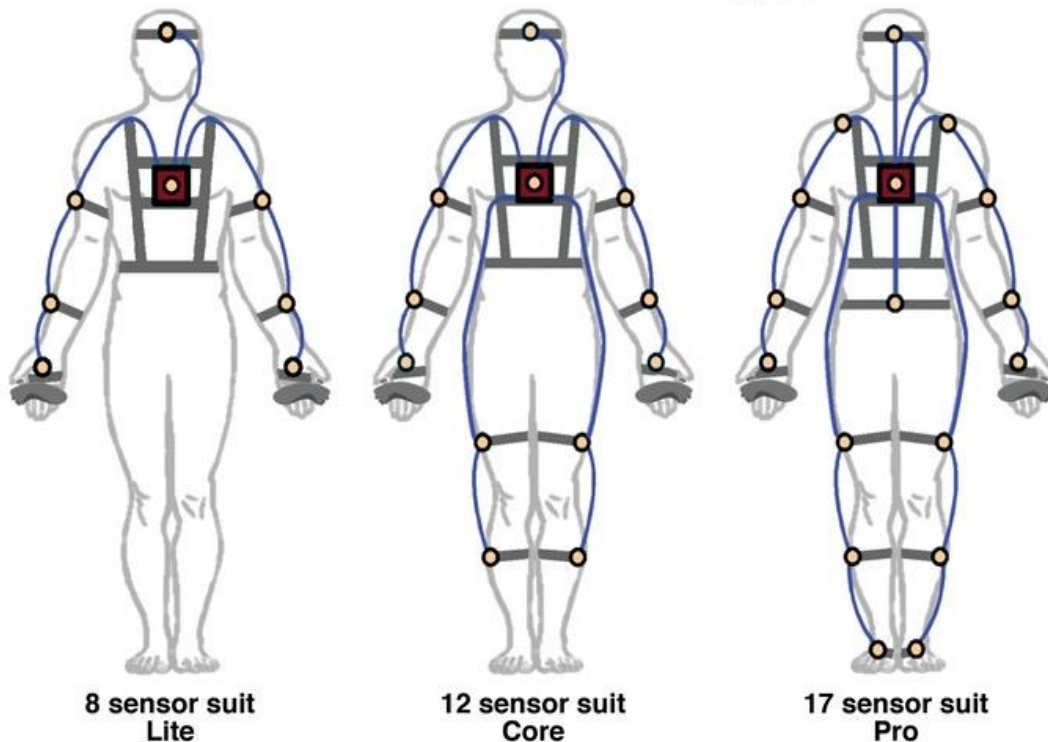


Figure 2: PrioVR different tracking methods

Motion recognition is another topic in software engineering and dialect innovation with a goal of interpreting human signals through mathematical algorithms. Hand gesture is a strategy for nonverbal communication for individuals as it expresses more liberally than body parts. [Sagayam, et al.] proposes a study on various motion recognition methodologies which is given specific accentuation on available motions.

Hand gesture acknowledgment has more prominent significance in planning a proficient human computer interaction framework, utilizing signals as a characteristic interface favorable to circumstance of movements. Regardless, the distinguishing proof and acknowledgment of posture, gait, proxemics and human behaviors is furthermore the subject of motion to appreciate human nonverbal communication, thus building a richer bridge between machines and humans than primitive text user interfaces or even graphical user interfaces.

The iCinema Centre for Interactive Cinema Research at UNSW has created a versatile virtual reality theatre that, by combining real-time 360-degree omni-stereo projection with surround audio and marker-less motion tracking, provides a highly immersive and interactive environment for up to 20 users. The theatre, codenamed AVIE, serves as the Centre's principal platform for experiments in interactive and emergent narrative, artificial intelligence, human-computer interfaces, virtual heritage, panoramic video and real-time computer graphics, as well as our primary platform for public exhibition of iCinema projects. [McGinity, et al.] briefly discusses the design of the system, technical challenges, novel features and current and future applications of the system. The authors believe their system to be the first and only 360-degree cylindrical stereo virtual reality theatre constructed to date, which later contributed to the technologies of immersive dome(Idome) cinema.

[Bun, et al.] presents a process of adaptation of the Oculus Rift and Samsung Gear VR devices for needs of immersive training application. The process was described on the example of virtual 3D human body atlas, created using EON Studio software. The aim of this application is to facilitate and make it more attractive to learn anatomy for students of medical disciplines. Possibilities of the application are, among others: selection of visibility level of certain organs or groups of organs, creation of complex sections in 3 planes and launching animations of selected organs. Application of each selected device required additional programming work. The work included both adaptation of Graphical User Interface for different display resolutions, as well as preparation of communication between integrated orientation tracking systems and the application, because none of the selected low-cost HMDs has a position tracking system, to allow navigation in the virtual space, two low-cost solutions were proposed, as well as traditional approach in form of a professional tracking system. First of the proposed low-cost solutions is application of a dedicated, customized controller built in the PUT VR Laboratory using commercial electronics and 3D printing technologies. The second solution is controlling the navigation via specific gestures recognized by the Kinect low-cost tracking device. After application was developed, a group of students tested all possibilities of interaction with the virtual environment using a professional HMD and both low-cost display and tracking solutions. During the evaluation performed by the test group, features like field of view, weight of devices and general impressions and feeling after prolonged use were taken into consideration. Intuitiveness of the proposed navigation solutions was evaluated separately. The evaluation performed by the test

group can be used during work on subsequent versions of the medical educational application and design of new peripheral devices.

Overall, this related work discussion regarding motion tracking and immersive applications provides a general idea of how optical and non-optical tracking systems work. The discussed points are all key factors in virtual reality motion tracking processes. The immersive dome cinema and medical training process will also be discussed later.

3. Materials and Methods

3.1: MPU-6050 Motion Tracking Project

For this experiment, we used the Arduino and Unity3D free software to simulate the motion tracking in virtual reality environment. It could be done with either Windows or Mac OSX system. The hardware we used are Arduino Uno which cost about \$20, the MPU6050 accelerometer with gyro which costs about \$5, and the google cardboard which costs about \$10. As we can see, it is a simple set-up device with very low cost.

3.1.1. Brief Introduction of Device Properties:

The inertial sensor that we used in this experiment is MPU-6050. Mainly it has three advantages compared with other sensors. First of all, it is accurate as it contains 16-bit analog to digital conversion hardware for each channel. In addition, it is fast as it uses the I2C-bus to interface with the Arduino. I2C is not only used on single boards but also to connect components which are linked via cable. Simplicity and flexibility are key characteristics that make this bus attractive to many applications. What's more, it is cheap and the total cost is less than \$10, but it combines both an accelerometer and a gyro.

The microcontroller that we used in this experiment is Arduino-Uno. It is a microcontroller board based on the ATmega328. Arduino-Uno is one of the most popular microcontrollers as it contains everything needed to support the microcontroller. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. Simply connect it to a computer with a USB cable or power it with an AC-to-DC adapter or battery to get started.

3.1.2. Integrating Arduino as a Motion Tracker:

We first read the raw values from the accelerometer and gyro. But the sensor also contains a 1024-byte FIFO buffer. The sensor values can be programmed to be placed in the FIFO buffer. And the buffer can be read by the Arduino. The FIFO buffer is used together with the interrupt signal. If the MPU-6050 places data in the FIFO buffer, it signals the Arduino with the interrupt signal so the Arduino knows that there is data in the FIFO buffer waiting to be read. The accelerometer and gyro values are called the "raw" values. This is just as with other accelerometer and gyro sensors. A more sophisticated application is using the DMP to retrieve specific computed values from the sensor. Then we read those data and send them to unity. For example, we used data 'a' to represent acceleration in the x direction. The baud rate is set at 115200.

3.1.3. Unity3D Software Interface:

Reading data from Arduino is straight forward. Once we set up the port communication, we use those represented letters to read the exact data we want. After that, some attunement must be done with those data to make our positional locations

more precise. The 3D model we used is one aircraft, which could rotate and move according to the movement of our MPU-6050 motion sensor.



Figure 3.1a: screenshot of aircraft motion simulation in Unity3D

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using System.IO.Ports;
4 using UnityEngine;
5
6 public class mpufly : MonoBehaviour {
7     private float AmountToMove;
8     public float x;
9     public float y;
10    public float z;
11    SerialPort serial = new SerialPort("COM3",115200);
12
13    // Use this for initialization
14    void Start () {
15        serial.Open ();
16        serial.ReadTimeout = 2000;
17    }
```

Figure 3.1b: partial code from Unity showing how to read data from Arduino

Then we set the display mode to our virtual reality mode. Here, we use google cardboard to demonstrate the results. The build platform can be switched between Windows, IOS, or Android.

Finally, we put our mobile phone into the Google cardboard. We may simulate the motion behaviors in virtual reality by moving the MPU-6050 motion sensor with our hand. The picture below demonstrates the process of this.



Figure 3.1c: Real-time motion simulating with MPU-6050 and Google Cardboard

3.2: Idome Project

The Idome project is based on the simulation of Arduino Uno and Arduino Ethernet Shield. The major goal is to simulation some specific effects during the movie when necessary. For example, if we want to have some mist during certain duration of the movie, this would be a nice solution. In addition, the EON Idome offers an immersive 3D experience for up to 24 people. By projecting onto the inside of a geodesic dome, the Idome gives the viewer a seamless full field-of-view 3D experience. The Idome includes motion seats that offer a “4D” experience that adds motion, scents, wind, and mist adding even more immersion to the Idome experience.

Regarding EON Idome, it offers an immersive 3D experience for up to 24 people. By projecting onto the inside of a geodesic dome, the Idome gives the viewer a seamless full field-of-view 3D experience. The EON Idome is designed to blend high resolution Hollywood cinematography with interactive 3D elements to create compelling, inspiring, and fun edutainment experiences for guests, but the technology can be adapted to other applications.

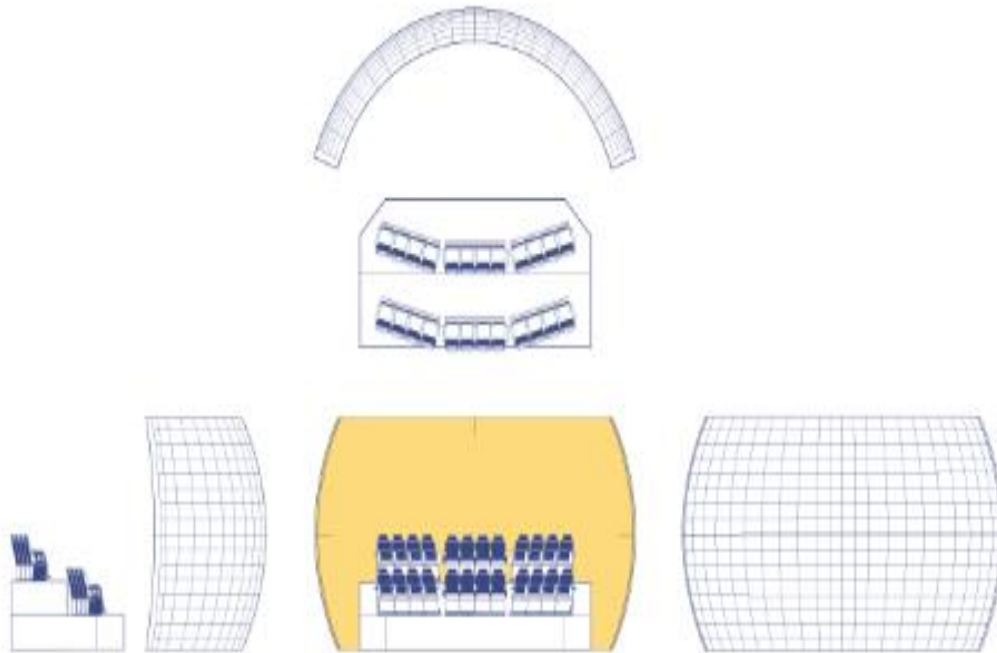


Figure 3.2a: Overview of EON Idome

Based on the film trailer ‘Dragon’, I have added wind and mist effects according to the trailer timeline. We import the film to EON software, and then send proper information through UDP to Arduino Shield. For example, ‘F1’ means fan on and ‘F0’ means fan off. Then our Arduino code can process the information and tell the relay to go on or off according to the packets received from EON software. A relay is an electrically operated switch which may control four different devices at one time. So here switch 1 controls the wind machine and switch 2 controls the fog machine. There is also one indicating light which indicates whether the overall system works properly.

```

void loop() {
  // if there's data available, read a packet
  int packetSize = Udp.parsePacket();
  if (packetSize) {
    Serial.print("Received packet of size ");
    Serial.println(packetSize);
    Serial.print("From ");
    IPAddress remote = Udp.remoteIP();
    for (int i = 0; i < 4; i++) {
      Serial.print(remote[i], DEC);
      if (i < 3) {
        Serial.print(".");
      }
    }
    Serial.print(", port ");
    Serial.println(Udp.remotePort());

    // read the packet into packetBuffer
    Udp.read(packetBuffer, UDP_TX_PACKET_MAX_SIZE);
    Serial.println("Contents:");
    Serial.println(packetBuffer);

    // send a reply to the IP address and port that sent us the packet we received
    Udp.beginPacket(Udp.remoteIP(), Udp.remotePort());
    Udp.write(ReplyBuffer);
    Udp.endPacket();
  }
  delay(10);
}

```

Figure 3.2b: Arduino Ethernet Shield code using UDP protocol to communicate

The picture below shows how the overall system looks like after soldering together the circuit parts.

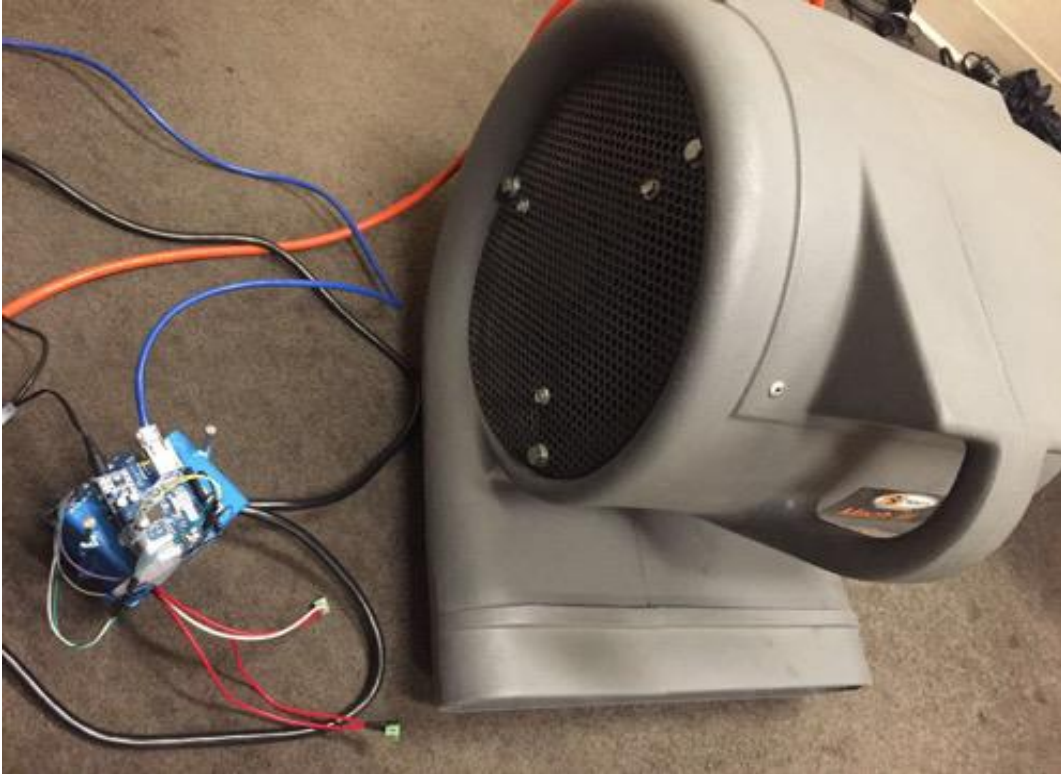


Figure 3.2c: Wind machine simulation with Arduino Ethernet Shield

3.3: Medical Tube Insertion Procedure Simulator

The purpose of this project is trying to simulate medical tube insertion with the control of mouse. It is quite similar to the Idome project as we also use Arduino Uno and Arduino Ethernet Shield as our communication devices.

Similarly, we send useful packets through UDP from EON software to Arduino Ethernet Shield. We create a slider inside the EON panel to control our servo, which will be later used to control some mechanical devices. There is also one indicating light which indicates whether the overall system works properly. We use P2P connection for mouse and Arduino since it is the most convenient way so far. There are four wires with power, ground, data, and clock. After all the connections, it will send information back to the EON software as position changes to control the virtual medical tube there. For example, +20 and -20 will tell the tube to go up in the x axis by 20 and go below in the y axis by 20. It is also possible to change the scale of position changes if necessary. As we may see from the picture below, slider controls how the servo rotates while the yellow bar is like our virtual medical tube.

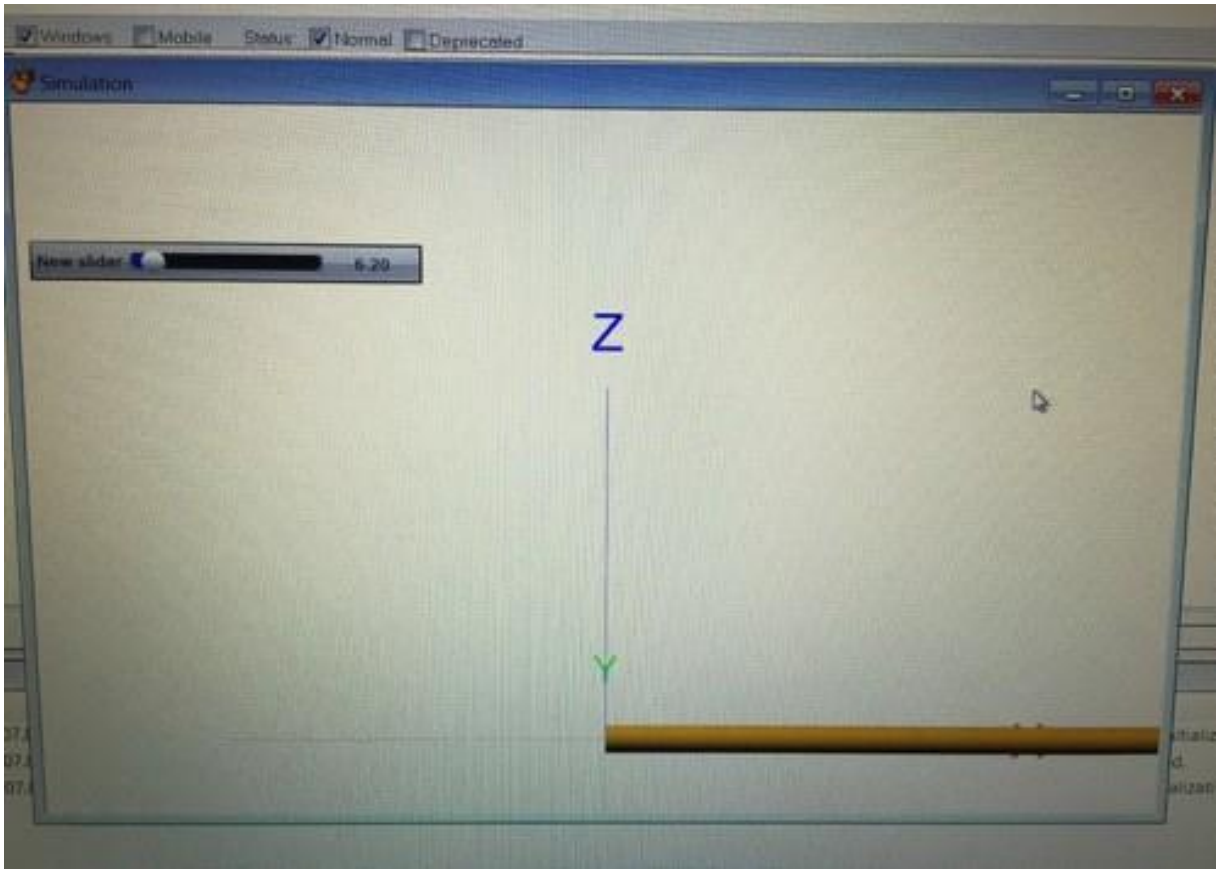


Figure 3.3: Medical Tube Insertion simulation with EON software GUI

4. Experiments and Results

4.1: MPU-6050 Motion Tracking Project

In order to evaluate the performance of the proposed real-time motion sensor tracking for VR, a qualitative experiment is carried out. The mobile phone is placed inside the google cardboard and the Arduino with motion sensor is controlled by hands.



Figure 4.1: Arduino-Uno with MPU-6050 inertial sensor

Table 1:

| Type | Cardbox+MPU6050-Arduino | Oculus Rift | Samsung Gear VR |
|----------------------|-------------------------|-----------------|-------------------|
| Platform | PC/Android/IOS | PC | Android |
| Resolution | Depends on platform | 640*800 per eye | 1280*1440 per eye |
| Cost (entire system) | \$35+ | \$500+ | \$100+ |
| Tracking Latency | ~1ms | ~2ms | <20ms |
| low-persistence | No | No | No |
| Tracking | Motion sensor | 3 DOF angular | 3 DOF angular |
| End-to-end latency | 100ms | 50-60ms | 50-60ms |

As we can see from the above table, our project was successfully carried out with mostly ideal results. We fully take the advantages of inertial sensor and make the tracking latency very low. It can have the same resolution as those commercial products, i.e. Oculus Rift and Samsung Gear VR. However, non-optical tracking method has its limits. Its tracking accuracy is a bit worse than those camera-based tracking systems. Accumulated drift may get higher as time goes on, which also indicates that inertial sensor tracking cannot work well in the long run.

4.2: Idome Project

By making use of the Arduino Ethernet Shield, we could send as much information as we want to the EON software, from which we could control the timeline and add all the effects we want for a movie. The communication latency is about 50ms and the mist effect can last as long as we wish. We can also control multiple machines at the same time, like one wind machine and one mist machine with only one small Arduino and Ethernet Shield.

The overall project cost depends on the cost of our external machines. The Arduino Uno plus Arduino Ethernet Shield should cost around \$50. Some typical external machines, like large wind machine costs about \$50 and mist machine around \$20. The robustness can be guaranteed as all the pins and wires are soldered solidly. Arduino Ethernet Shield also provides stable wireless transformation between devices.

Regarding UNSW Idome project, the spherical movie is projected inside the dome with a coverage of 180 degree. A track ball allows the user to rotate the projection freely while the movie is still running. Together with the image, the multi-channel sound field is rotated accordingly. A simple vector based panning algorithm is used to distribute the multi-channel sound dependent of the user's principle point of view. The tracking or controlling process is complicated. It also requires human interaction to be able to control certain effects during the movie. While our EON Idome project provides automatically started timeline machine to control whatever effects we want during the movie.



Figure 4.2: UNSW Idome track ball system

4.3: Medical Tube Simulation Project

This project allows us to control some complicated actions under virtual environments by simply using a mouse. Since we are using the same communication method as Idome project with the Arduino Ethernet Shield and EON software, the communication latency is about the same as Idome project. The P2P connection between mouse and Arduino is very stable and provides us very accurate raw data for the coordinates. Thus, this method provides an accurate and stable simulation for the medical tube insertion procedure.

The overall cost of this project is around \$50 as the price of P2P mouse is only \$1-2. It is kind of hard to find these types of mouse nowadays as I bought it from China. The connection is very stable as I mentioned above because it has 6 pins. One is ground, one is power; 2 pins are for clock and data, and the other 2 are not connected.

The communication latency is about 50ms as we are still using Arduino Ethernet Shield here. All the simulation results can be tuned from the EON software conveniently. Arduino provides robustness connections with 4 pins connected with the P2P mouse with clock and data gives us precise positional information.

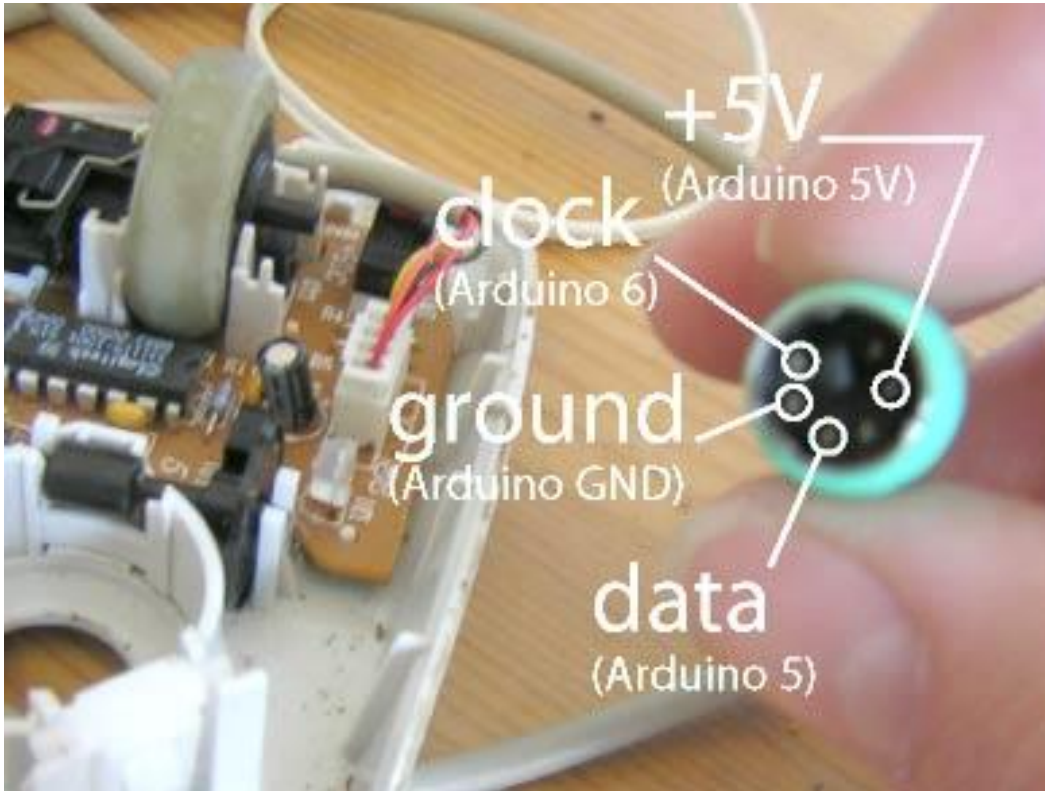


Figure 4.3a: PS2 mouse interface with Arduino

The Tube Feeding Simulator is a commercial product which offers effective training for caregivers and medical professionals who work with patients receiving enteral tube nutrition. But one single simulator costs more than \$2300. Some may argue that the advantages of real operation feeling cannot be replaced. But virtual reality technologies can already provide us an authentic environment to learn and train whatever we want. The project only costs around \$50 with the servo motor working. However, we can replace the servo with any mechanical machines that we want. For example, using automatically moving light bulbs could help the medical professionals to better observe the patients without using their own hands. We may also add many authentic training practices with little software development and avoid the possibility of damaging the real tube.



Figure 4.3b: commercial tube feeding simulator

5. Discussion

Real-time motion tracking is a significant issue for any VR systems, and there are different methods to realize the tracking performance. I have shown in this thesis that it is possible to build specific sensors connected with Arduino or to communicate through UDP to Arduino for a given non-standard application and achieve equivalent/better results than what is provided from commercial products.

It is worth noting that all of our proposed experiments take the advantage of Arduino. Using an Arduino simplifies the amount of hardware and software development we need to do in order to get a system running. The Arduino hardware platform already has the power and reset circuitry setup as well as circuitry to program and communicate with the microcontroller over USB. On the software side, Arduino provides a number of libraries to make programming the microcontroller easier. The simplest of these are functions to control and read the I/O pins rather than having to fiddle with the bus/bit masks normally used to interface with the Atmega I/O.

There is a decent amount of great software to choose so as to implement the simulations under virtual reality environment. Unity3D is overall the best choice here in my opinion. First of all, it has free version for personal development as EON Reality is only free for their employees. Many other 3D game engines would require you to subscribe in order to gain most useful features. In addition, Unity supports build for Mac, Android, and IOS. We also get good and helpful documentation if we spend some time researching their official website. Scripting can be done in Javascript or C#. Adding scripts is trivial, and Unity comes with excellent scripting examples that concentrate on only a few different tasks. Overall, adding functionality by scripts was pretty easy. The EON software, on the otherside, only support Javascript as programming language. Commercial products, like PrioVR requires us to download their own software. However, we still have to learn how to use it and pay some money to their development engineers. Thus, I would conclude that Unity3D is overall the best option for VR application development.

6. Conclusions and Future Work

This thesis describes feasible user-designed solutions for custom real-time motion tracking solutions for VR applications, leading to cheaper and more convenient solutions than relying on commercial sensors. With the support of Arduino, we can make interesting simulations in virtual reality environment conveniently. The Google cardboard also provides us an easy way to interact with the virtual reality. In addition, the temperature sensor on the MPU-6050 could also bring us some cool features when necessary.

In the MPU-6050 project, only one motion sensor is used for demonstration purpose. If we want a more accurate motion tracking system, multiple sensors could be applied. Thus, future work could be done by applying more sensors with cheap costs for our motion tracking system.

With the help of simple devices, we can use our VR technologies for education, medicine, and many other potential areas. For example, the Idome project could make '5D' movie in the future with the effect of wind and mist. The MPU-6050 sensor project can help pilot to simulate flying under virtual reality environment. I believe that virtual reality technologies will become more useful and popular in the next few years.

Acknowledgements: I would like to thank Professor Marcelo Kallmann for the help through my graduate research, and all the EECS department professors from UC Merced who have helped me during my graduate studies. Also, special thanks to EON Reality as they provided me a good opportunity to do internship last summer and learn much useful knowledge about VR.

References:

1. Wei Fang, Lianyu Zheng, Huanjun Deng and Hongbo Zhang, Real-time motion tracking for mobile augmented/virtual reality using adaptive visual-inertial fusion
2. Sagayam, K.M. & Hemanth, D.J. Virtual Reality (2017) 21: 91.
doi:10.1007/s10055-016-0301-0
3. Pawel Bun , Filip Gorski, Damian Grajewski, Radoslaw Wichniarek, Przemyslaw Zawadzki Poznań University of Technology, Piotrowo 3 STR, 60-965 Poznan, Poland
4. P. Bun, et al, Application of Professional and Low-cost Head Mounted Devices in Immersive Educational Application Procedia Computer Science, 75 (2015), pp. 173–181
5. Cristina Botella, Rosa M. Baños, Azucena García-Palacios, Soledad Quero, The Science of Cognitive Behavioral Therapy, 2017, Pages 551-590
6. Matthew McGinity, Jeffrey Shaw, Volker Kuchelmeister, Ardrian Hardjono, Dennis Del Favero, AVIE: A Versatile Multi-User Stereo 360° Interactive VR Theatre
7. <https://www.kickstarter.com/projects/yeitechnology/priovr-suit-up-game-on>
8. <https://playground.arduino.cc/Main/MPU-6050>
9. <http://er.educause.edu/articles/2016/3/the-promise-of-virtual-reality-in-higher-education>
10. <https://www.engadget.com/2014/08/28/leap-motion-s-next-sensor-is-designed-specifically-for-virtual-r/>
11. <https://playground.arduino.cc/ComponentLib/Ps2mouse>
12. <https://www.eonreality.com/eon-idome/>
13. <https://playground.arduino.cc/ComponentLib/Ps2mouse>
14. <http://digital.csic.es/bitstream/10261/127788/7/D-c-%20Arduino%20uno.pdf>

15. <http://icinema.edu.au/technologies/idome/>

16. <https://www.limbsandthings.com/us/our-products/details/tube-feeding-simulator1>