Gesture Recognition Controls for Audio Playback

CS 580I Term Project: Leap Music

Jeremy Meguira
Department of Computer Science
Binghamton University
Binghamton, NY USA
jmeguir1@binghamton.edu

Dandi Wang
Department of Computer Science
Binghamton University
Binghamton, NY USA
dwang84@binghamton.edu

ABSTRACT

There is widespread interest within the scientific community to improve and utilize hand gesture recognition. The far-reaching applications of this technology touch the fields of robotics, medical practice, human computer interaction, and everything in between. Through the use of the Leap Motion Controller, a lightweight IR sensor, this project explored one practical application of this technology-controlling audio playback-using two different approaches. In the first approach, a simple music player web-application was created and manipulated by distinct hand gestures/poses. In the second, a 3D augmented reality environment was constructed using the Unity engine. This application uses hand-tracking to control a virtual music player interface. Although there are many useful utilities that facilitate the use of Leap Motion technology, we discovered in our trials that performance is still lackluster at times. There is still much room for improvement with this project and with hand-tracking/gesture recognition at large. Future work should seek to improve reliability and performance of the Leap Motion Controller and its' related applications.

CCS CONCEPTS

• Gesture Recognition • Gesture Control • Augmented Reality

KEYWORDS

Leap Motion, Unity, Motion Detection

1 Introduction

The internet of things is ever-expanding. Regularly, sensors and new related technologies become available for consumer use. Recent advancements have brought sensors and sensor technology to the medical, defense, automotive, and entertainment industry. Logically, there is strong motivation among the scientific community to use these advancements to create new functionalities and enhance the user experience with existing functionalities. This is one of the core motivations for this project. Our team shares an appreciation for product features, such as the Apple Trackpad, DJ controllers, and the Windows 10 snap feature, that amplify or streamline a user's experience with technology. In addition, both of our team members enjoy electronic music and DJing technology. At the intersection of these two interests our team set out to use one of these sensors, the Leap Motion controller, as a means of capturing hand/digit signals and utilizing them to control audio playback. Our first approach involved use of LeapTrainer technology to train and recognize hand gestures and then using said gestures as input for a simple music playing web application. Second, we opted to develop a 3D AR (augmented reality) environment in which the Leap controller's input is used to interact with a music interface.

1 Hardware

The primary hardware component of our project is the Leap Motion Controller. We also made use of several open-source libraries and applications in the completion of this project. Notably, the LeapTrainer interface and the Unity development platform were used.

1.1 Leap Motion Controller

The Leap Motion Controller is a lightweight peripheral sensor device that registers hand and digit input. (Fig. 1)

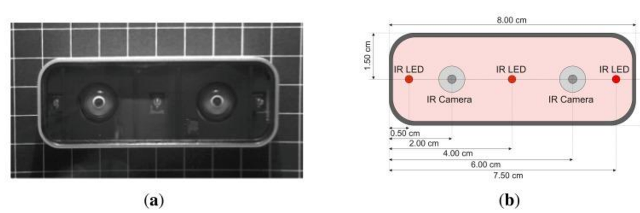


Figure 1: a.) Infrared image of the Leap Motion controller b.) Schematics for the Leap Motion Controller

The sensor can be positioned facing upward on a surface (as we have used it) or mounted on a VR headset. The controller utilizes two infrared (IR) cameras and three IR light-emitting diodes (LEDs) that enable a hemisphere of detection approximately 2 feet in radius. (Fig. 2) [1]

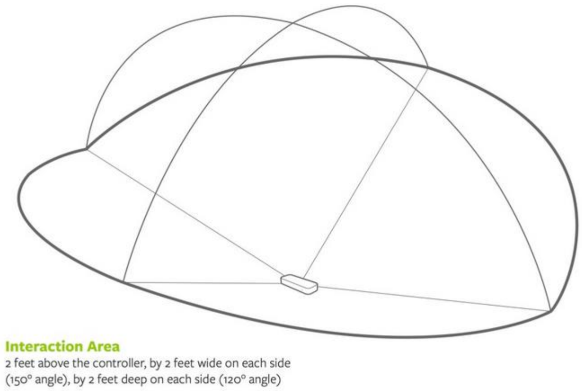


Figure 2: Hemispherical interaction area sensed by Leap Motion Controller

This raw sensor data is then fed through the proprietary Leap Motion Service. The Leap Motion Service extracts hands/digits from the raw data which are subsequently tracked. Leap's tracking algorithm also filters the image data and reconstructs/infers the position of occluded objects.

2 Gesture Controlled Web Application

There are two significant components of the gesture-controlled web application. The raw tracking data from the Leap Motion Controller must first be analyzed to find meaningful expressions of gestures or poses and subsequently these recognized gestures/poses must be used to manipulate an audio interface in some way. [2] The primary tool we used for gesture recognition is the opensource LeapTrainer library. We then linked the recognized gestures to a HTML/javascript audio player web-application using jQuery.

2.1 LeapTrainer

To implement gesture recognition in the web-based application we utilized the LeapTrainer javascript library. [3] The LeapTrainer library utilizes the tracking data provided by the LeapMotion Controller and enables the user to quickly train/export gestures and poses for use in other applications. (Fig. 3) LeapTrainer stores a template 3D image of a gesture or pose input by the user and then compares new sensor data against this template. If the new data meets a certain threshold for similarity, then a corresponding event is triggered. This event can be linked to features within a web application such as ours. This process is known as geometric template matching and LeapTrainer is an extrapolation of the 2D gesture recognition technique known as Point-Cloud Recognition. [4]

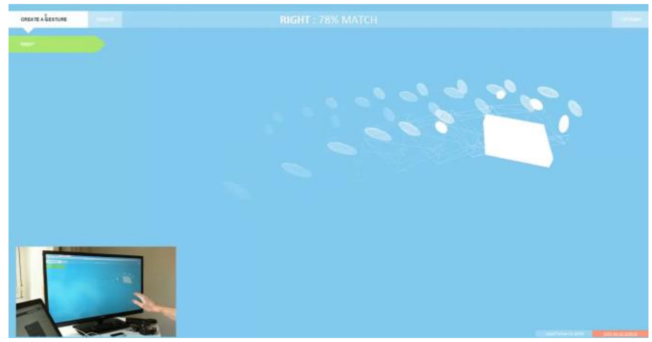


Figure 3: LeapTrainer UI

2.2 Results

Ultimately, the system that we implemented was functional but not as reliable as we would have hoped. Over the developmental period, we were able to make some aesthetic improvements to the audio interface and improve the layout. (Fig. 4)

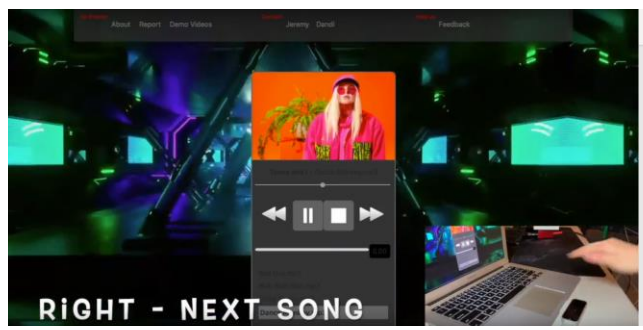


Figure 4: Gesture controlled web application

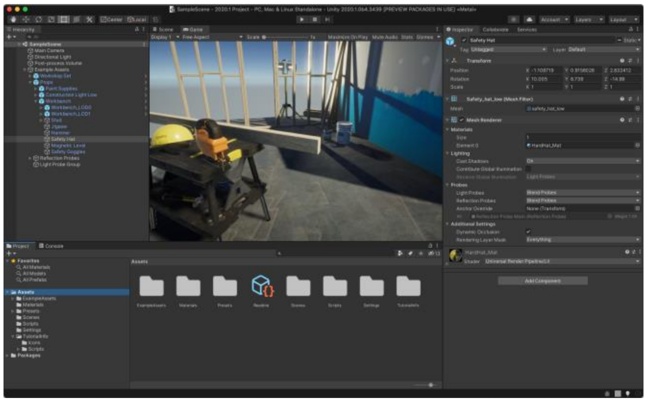
The system also enables us to rapidly train or retrain gestures and poses and feed those to the web application. However, the system does occasionally render false positives and incorrect labels for gesture input. Further, we have learned that trained gestures from one person may contain minor fluctuations to another person's trained gestures. This makes the system inconsistent from one user to another and requires retraining to correct. With more in-depth user testing, statistical analysis, and code modifications we could seek to amend some of the consistency issues.

3 Augmented Reality Unity Environment

Upon completion of the initial task, our team decided to experiment with a slightly different approach to audio playback. Our goal in creating this new environment was a proof-of-concept to see if raw tracking data in an AR 3D environment affords more reliability than our previous gesture-based implementation. For this purpose, we looked to the Unity development engine.

3.1 Unity

The Unity development engine has seen widespread use beyond its initial developmental purpose of game design. The development suite allows users to create applications in 2D and 3D and supports a wide variety of technologies. (Fig. 5)



Although it is still prominently utilized by industry leaders in video gaming, the software has been repurposed by virtually every other industry. For our project, the primary benefit of the Unity development engine is LeapMotion support. The LeapMotion V4 SDK allowed us to quickly add support for the raw tracking data to the application we developed.

3.2 Results

Although the environment is far from polished and is merely a first foray into 3D development we were satisfied with the proof of concept. We created a virtual environment with playback controls and some additional aesthetic touches. The system is functional but bare-bones at this point.

4 Conclusions

This project initially sought to explore methods of using gesture-recognition to control audio playback. At a fundamental level, we can confidently say that we have accomplished this task. Using our web application, we can quickly train gestures and poses and use them as a part of our application. Further, our approach extends to any other application with javascript support. Although functional, the web player did leave something to be desired in terms of precision and reliability. We also encountered minor gesture recognition obstacles from person-to-person. For these reasons, we additionally explored a 3D environment-based approach that was also moderately successful. With more time, user-testing, 3D development experience, and statistical analysis, we can work to improve the aesthetic and functional quality of the 3D project going forward. Future work in this area should continue to emphasize ease-of-use, cross-user compatibility, and precision as key indicators of success.

ACKNOWLEDGMENTS

We wish to thank Professor Sha and the course T.A. Junyang Shi for their guidance and the loan of the Leap Motion Controller throughout the completion of the project.

REFERENCES

[1] Frank Weichert, Daniel Bachmann, Bartholomäus Rudak and Denis Fisseler Analysis of the Accuracy and Robustness of the Leap Motion Controller. [https://www.mdpi.com/1424-8220/13/5/6380]. Published 14 May 2013

[2] Sushmita Mitra and Tinku Acharya Gesture Recognition: A Survey [https://ieeexplore.ieee.org/abstract/document/4154947] 16 April 2007

[3] Cleo Hayes-McCoy roboleary, LeapTrainer.js(v0.31) Gesture and pose learning and recognition for the Leap Motion [https://github.com/roboleary/LeapTrainer.js/#sub-classing-the-leaptrainercontroller] Nov 13, 2013

[4] Radu-Daniel Vatavu, Lisa Anthony, Jacob O. Wobbrock. Gestures as Point Clouds: A $P Recognizer for User Interface Prototypes [http://faculty.washington.edu/wobbrock/pubs/icmi-12.pdf]