American Sign Language to Speech Translator Using Leap Motion Technology

Daniel Fong
Graduate Student, Electrical Engineering,
University of California, Davis
2064 Kemper Hall, University of California
One Shields Avenue, Davis, CA 95616
dfong@ucdavis.edu

ABSTRACT

In order to lower the communication barrier between the deaf and hard of hearing and the hearing population in the United States, an American Sign Language (ASL) to speech translator was developed using the Leap Motion technology. This technology captures gestural information using infrared LEDs and cameras and provides an API that allows 3rd party app development. A simple pattern-matching engine was developed to match the gestural information to a specific sign and translate that into text. This text was then fed into a text-to-speech synthesizer to generate the audible speech. A visualizer was developed to provide visual feedback to the signer as to what the Leap Motion device contextually sees. Several simple signs in ASL were successfully distinguished by the system and turned into speech, thereby successfully allow communication from the deaf population to the hearing population without the need for both parties to learn ASL. This system provides a base from which future works can expand and improve upon. One such aspect should be the integration of a communication channel allowing information to flow from the hearing population to the deaf population.

1. INTRODUCTION

The relative employment rate of persons with a severe hearing impairment is much lower than those who are in the hearing population, according to a study done in 2011 [1]. Furthermore, the study also reports a lower college graduation rate for those who have a hearing impairment than their hearing counterparts. This disparity can be explained by the communication barrier that separates the hearing and hard of hearing. Developing technology that can act as a translator between the two populations can bridge this gap. The most commonly used form of communication for the deaf population in the United States is American Sign Language (ASL). This work uses the Leap Motion technology to develop an American Sign Language to speech translator.

2. LEAP MOTION TECHNOLOGY

David Holz, co-founder of Leap Motion Inc., developed the technology behind Leap Motion in 2008 while he was studying for a Ph.D. in Mathematics at the University of North Carolina [2]. The Leap Motion hand-capture system, hereafter referred to as Leap Motion, is described in the following sections broken down into two parts, consisting of the hardware and the software.

2.1 Hardware

The hardware portion of the Leap Motion, hereafter referred to as the Leap Motion device, consists of three infrared (IR) LED emitters, two small IR monochromatic cameras with a fish-eye lens, and supporting hardware [3]. This small module (3.0x1.2x0.5 inches, 1.6 ounces) emits IR light, upward from the LEDs and captures the reflected light using the IR cameras. The captured images are then sent to the computer via a USB 3.0 port for post-processing and hand position and orientation estimation. The hardware uses an EZ-USB FX3 USB 3.0 controller developed by Cypress Semiconductor Corporation [4]. The reported capture rate of the IR cameras is at 300 fps.

Figure 1. The Leap Motion device connected to a laptop and displaying simple hand location using the closed-source visualizer provided with a purchase of the device [2].

2.2 Software

The software portion of the Leap Motion, hereafter referred to as the Leap Motion service, consists of a service that connects to the operating system and receives the captured frames from the Leap Motion device. The service, with a native C++ backend, translates the images into information about the hand, and sends the information to any application requesting the data using a Publisher-Subscriber software architecture. In order to encourage developers to write applications for the Leap Motion system, there are API wrappers for Objective-C, Java, JavaScript, Python, Unity, C#, and the native C++ languages.

2.2.1 Hand-extraction from image

Unlike other popular 3D mapping technology, the Leap Motion system does not use a depth-map creation approach. Rather, it exploits the fallout of light intensity from a source to distinguish between background and object pixels. After extracting the object pixels, the hand or hands, the Leap Motion service then uses the edges of the hands to infer the tangential lines of those parts of the hands, and creates estimations of the cross-sectional areas using a simple closed curve (ellipses). This recreated hand can be seen in Figure 2, provided by David Holz' US Patent Application (US2013/0182079) [5]. By combining successive frames and the

Submitted: December 14, 2014

position of the ellipses, hand position and orientation is inferred. The service then translates and wraps this inferred information about the hand into an interface that is easier to use for developing higher-level applications.

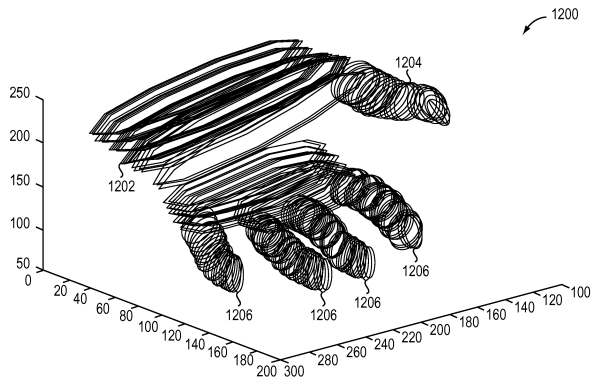

Figure 2. The recreation of a hand using the Leap Motion technology. After using infrared reflectance to separate the distinguish the background, tangential lines are inferred at the edges of the hand and cross-sectional ellipses of the fingers and palm are formed using a closed-curve estimation [5].

2.2.2 Skeletal Tracking Model

In order to create a generic interface that would allow developers to create applications that can exploit this touchless interface, the Leap Motion service presents the captured image data in the form of a skeletal tracking model. Excluding the carpal bones, located near the wrist and which connects the five metacarpal bones to each other, there are four bone types (three for the thumb) that define each finger. From distal to proximal locations, the four bone types are the distal phalanx, middle phalanx (excluded in the thumb), proximal phalanx, and the metacarpal bones, shown in Figure 3.

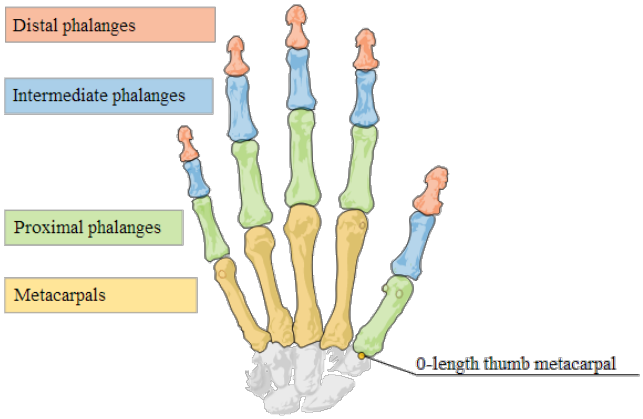

Figure 3. Anatomy of the human hand. This is used in the skeletal tracking model of the LeapAPI [2].

The Leap Motion API, hereafter referred to as LeapAPI, presents these bones as normalized vectors, allowing users to extract direction and position using a defined Cartesian coordinate system shown in Figure 4. The service also wraps these bones into higher-level objects, namely fingers and hands, and presents many useful convenience methods for extracting data and performing calculations with vectors. These higher-level objects, hereafter referred to as gestural data, are immutable and do not have public constructors, thereby securing the LeapAPI for proprietary use only with the Leap Motion service.

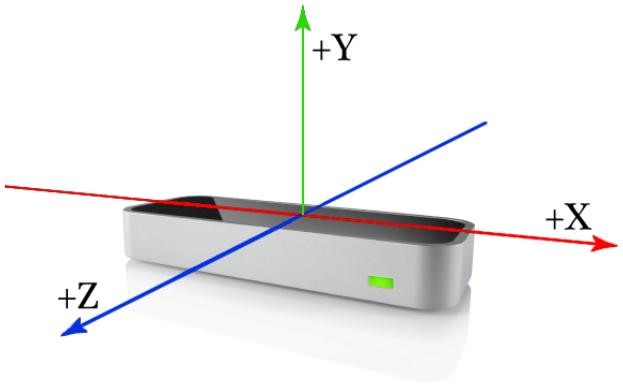

Figure 4. Cartesian coordinate system used by the LeapAPI. Objects that portray information displayed in this coordinate system are defined as vectors to signify direction and position [2].

3. ASL TRANSLATION METHOD

In order to translate signs in ASL into speech, a simple pattern matching system was developed. These patterns would analyze the gestural information published by the Leap Motion service and determine if a known sign was presented. At the core of this system, there are three main modules: the pattern-matching engine, the speech synthesizer, and the visual feedback renderer. The following system was developed in the Eclipse IDE using the Java programming language and interfaces with the LeapAPI via the aforementioned Java API wrapper.

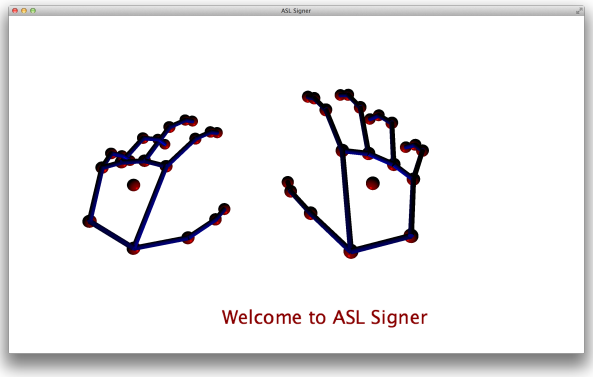

Figure 5. The visualizer developed for visual feedback portion of this translation system.

3.1 Pattern-Matching Engine

The ASL vocabulary consists of more than 7200 words and signs. Of these signs many use other parts of the body, aside from the hands, to convey a word. The author of this paper chose to exclude these signs to simplify the complexity of the problem and develop a pattern matching system using the Leap Motion technology for a smaller subset of signs, consisting of only hands with minimal movement, and allow future works to develop a more robust system.

While there are many common elements of different ASL signs, it is clear that different signs emphasize different actions and not all

Submitted: December 14, 2014

elements are important to a specific sign. For these reasons, a simple flat and modularized pattern-matching scheme was created and used. This was accomplished by allowing each pattern to describe a sign and the important elements of the gestural data for that sign, and build a larger dictionary of patterns to match each sign on. The gestural data published by the Leap Motion service is then compared against all of the patterns in the dictionary, which reports if a match was found and the corresponding word. In order to reduce the amount of work the dictionary needs to do, some pre-processing is done to determine if the gestural information contains a single hand (meaning only one hand can be seen by the Leap Motion system) or two hands. If two hands are seen, then the dictionary will only compare the gestural information against patterns that need two hands. Similarly, the dictionary will only compare against a single hand if one hand is seen. No comparisons are made if no hands are contained in the gestural information. Furthermore, the dictionary allows the patterns to be compared in an ordered manner, depending on the order they are input into the comparison list, thereby allowing a decreased overall processing time by ordering the patterns by most frequently used. This scheme also makes it possible for future works to allow the user to create patterns unique to them, and input it to the pattern matching system through some form of persistent storage, like a database file.

In order to overcome the problem of oversampling (repeated matches on the same pattern) the sampling time for pattern matching was reduced to two frames per second. This allows ample time for all patterns to be processed and for the speech synthesizer to complete speaking. Future works could develop a queuing system for the speech synthesizer and increase the sampling time for pattern matching.

The patterns developed in this work use the coordinate system shown in Figure 4 and various unit vectors to determine if the gestural information result in a match. Several unit vectors commonly used for the pattern matching, described as (x, y, z) coordinates, were the unit vectors along the axes, namely Forward (0, 0, -1), Backward (0, 0, 1), Left (-1, 0, 0), Right (1, 0, 0), Upward (0, 1, 0), and Downward (0, -1, 0). Other useful, but less commonly used, unit vectors include Left-forward, Right-forward, Left-backward, Right-backward, and other intermediate vectors between the axes unit vectors. Many patterns were matched by comparing the angle between a unit vector and the direction vector reported by the finger, bone, and/or hand. This allowed the pattern to infer pointing gestures. Furthermore, by comparing angles between the directions of different bones, patterns can also determine if the fingers are bent. It is important to note that left and right directions can mean different things, depending on if the hand(s) being compared are left or right hands. Lastly, in order to determine palm orientation, the roll, pitch, and yaw angle magnitudes were calculated based off of the direction and normal vector of the palm. These were then compared using different angle thresholds to determine if a palm were facing right, left, up, down, forward, or backward. If all compared angles were below (or between) a certain threshold, then we could infer that the gestural information matched that pattern. The patterns that were developed for this engine and successfully distinguishes their pattern were the signs for the alphabetical letters “a”, “b”, and “d”, and the signs for the words “hello”, “me”, “you”, “name”, “who/what/where/when” (these are all the same sign), “attention”, and “thank you”.

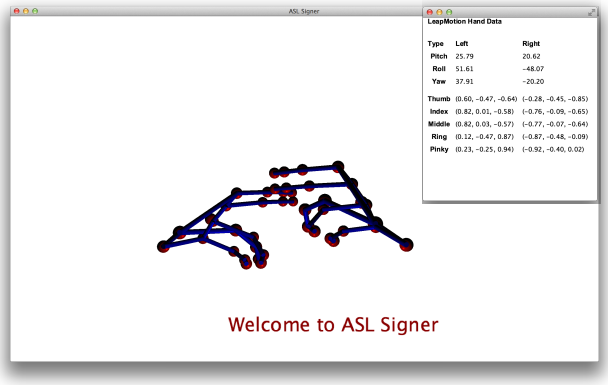

Figure 6. The ASL sign for "Name" and corresponding data as displayed in this translation system. This required that the middle and index fingers of the hands be pointed to around 45º from the front in the opposite hand's direction. The other fingers needed to be pointing in other directions and curled. Once this sign was reached, the synthesizer would produce the audio word for “name”.

Various vectors provided by the gestural information that were useful in pattern matching were the direction vectors of the fingers, bones, and hands, and the normal vector of the palm. It is important to note that if a pattern needed more information to determine if the gesture matches its sign, it has full access to the LeapAPI, which also provides information like palm velocity, and can determine palm rotation angles between previous frames. If a pattern reports a hit, all subsequent patterns are skipped, and the resulting word is reported as text. This text is then passed over to the text-to-speech synthesizer.

3.2 Text-to-Speech Synthesizer

This text is then passed over to the text-to-speech synthesizer, which uses the computers built-in speakers to play the audible speech. The pattern matching system used FreeTTS, an open-source java text-to-speech synthesizer, which provided a convenient API to input text and output audible speech. This text-to-speech synthesizer was developed by the Speech Integration Group of Sun Microsystem Laboratories, and was based off of Flite: a small speech synthesis engine developed at Carnegie Mellon University, which was derived from the Festival Speech Synthesis System from the University of Edinburgh and the FestVox project from Carnegie Mellon University [6]. This work uses FreeTTS v1.2, as-is, and used the voice Kevin16 for synthesis, a configurable property in the API.

3.3 Visual Feedback

The pattern matching system developed also provides visual feedback to the signer by displaying what the Leap Motion device sees to help the signer help the system match patterns. The visual feedback was built using JavaFX and a partially implemented open source framework provided by another developer also using Leap Motion [7] and accessed through the code sharing cloud service called github. The author completed the relevant portions of the framework (syncing with and subscribing to the Leap Motion service, and position translation from the Leap Motion device’s coordinate system to the JavaFX frame’s display coordinate system for drawing). In order to reduce the amount of processing time due to image rendering, not every image frame published by the Leap Motion service was rendered. Instead, a refresh rate of 62 Hz (period of 16 ms) was used since many

common computer screens also use this rate to maintain visual consistency and protect the user experience. Furthermore, in order to help the developer, a data frame was also created which displays statistics on the gestural information being published by the Leap Motion service. This was useful in helping to determine the appropriate threshold ranges of angles for various patterns developed. The resulting threshold range was ±20º of the desired angle for most pointing gestures, and expanded to ±33º for more complicated gestures.

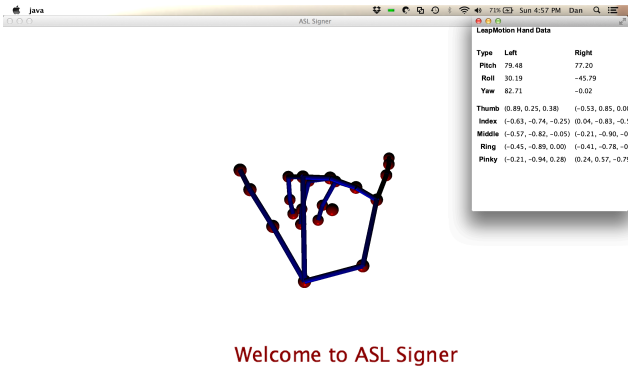

Figure 7. The visual feedback with corresponding debug data. The data in the named *Left* column corresponds to the left hand, which was in the frame earlier. This data is the last seen data of that hand. The *Right* column shows the data of the hand currently being displayed. The hand is in the form of a common Hawaiian greeting known as the 'shaka'.

4. DISCUSSION

Using the Leap Motion technology to develop an American Sign Language to speech translator is feasible for simple signs, but can be improved. It is important to note that ASL is normally displayed and viewed anterior of the body. Many signs were developed so that features of the hand were easier to distinguish when viewed from that orientation. However, in this set up of this work, the Leap Motion device was placed upright on the table, such that the IR cameras and emitters were looking upward at the bottom of the hand. This made it much harder for the system to distinguish features of the hand where fingers were covered by other fingers. For example, the sign for the letter 'n', in Figure 8, require that the index and ring fingers wrap themselves around the thumb. This makes it much harder for the Leap Motion technology to determine the location of the thumb since the technology relies on the reflectance of the IR light and hard edges to estimate a cross section of the finger. However, future works should explore the feasibility of orienting the Leap Motion device in a way such that the IR cameras capture an anterior view of the signs.

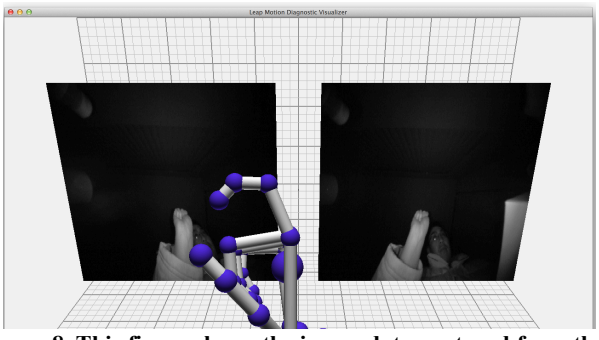

Figure 8. This figure shows the image data captured from the IR cameras on the Leap Motion device. This shows how difficult it is to distinguish between the letter "n" being signed.

The purpose of this work is to help bridge the communicational gap between the deaf and the hearing population, however, the system developed only allows for one-way communication from the deaf to the hearing population. This system still requires the hard of hearing to be able to read lips, and or requires the hearing population to be able to communicate via ASL. The integration of this step fully defines the functionality of a minimally viable product needed to bring this ASL to Speech translator system to a beta version. Future works could realize this by integrating a speech to text component using the computers built-in microphone.

The pattern matching engine developed above is useful in being able to plug-in signs that are frequently used and in attempting to reduce the overall processing burden by splitting the pattern matching into distinctive parts, mainly one hand or two hand, and allowing the patterns to be ordered by how frequently it is used or matched. An alternative ordering could be accomplished by giving order priority to the more stringent patterns first. While this type of ordering would promote the accuracy of a sign by imposing an implicit filtering scheme, it would increase overall processing time for all patterns.

5. CONCLUSION

An American Sign Language to speech translation system was built using the Leap Motion technology and was able to successfully translate simple signs into audible speech. The work derived here can be used as a base from which more robust systems can be developed. Future works should take into consideration that a communication system from the hearing to the hard of hearing still needs to be integrated into this system to create a minimally viable product.

6. ACKNOWLEDGMENTS

Our thanks to Dr. William Vicars of California State University, Sacramento for providing free online lessons on American Sign Language, the engineers at Leap Motion Inc. for providing the technology that makes this system possible, Zoltan Ruzman for developing an open-source framework that was useful in creating the visualizer, Jessica Ogihara for her interest in American Sign Language leading to the initial thought of this project, Dr. Ghiasi and the Fall 2014 EEC284 classmates at UC Davis for their enthusiasm which helped enhance the authors motivation towards completion, and the members of the MicroNano Innovations Laboratory at UC Davis for introducing the author to Leap Motion technology.

7. REFERENCES

[1] S. Schley, G. G. Walter, R. R. Weathers, 2nd, J. Hemmeter, J. C. Hennessey, and R. V. Burkhauser, "Effect of postsecondary education on the economic status of persons who are deaf or hard of hearing," *J Deaf Stud Deaf Educ,* vol. 16, no. 4, pp. 524-36, Fall, 2011.

[2] "Leap Motion Inc.," 2014; https://www.leapmotion.com/.

[3] S. Parvin, "LEAP MOTION: Operate your gadgets without touching them!," *LEAP MOTION: Operate your gadgets without touching them!*, 2014.

[4] C. S. Corp., " Leap Motion Selects Cypress's EZ-USB® FX3™ Solution for Controller Components," Cypress Semiconductor Corp., 2013.

[5] D. Holz, *MOTION CAPTURE USING CROSS-SECTIONS OF AN OBJECT*, United States,to OCUSPEC, U. S. P. A. Publication, 2012.

[6] W. Walker, P. Lamere, and P. Kwok, *FreeTTS: a performance case study*, Sun Microsystems, Inc., 2002.

[7] Z. Ruzman. "LeapFX," 2014; https://github.com/RuZman/LeapFX.

Submitted: December 14, 2014