International Journal of User-System Interaction 12(4) 2019, 255-266

Interactive Gesture-based Sign Language Recognition for the Romanian Alphabet

Adrian Sabou, Daniela Georgiana Stanciu

Technical University of Cluj-Napoca
28, G. Bariţiu Street, 400027, Cluj-Napoca, Romania
E-mail: adrian.sabou@cs.utcluj.ro, georgiana.daniela.stanciu@gmail.com

Abstract. This paper presents an approach to interactive, gesture-based, sign language interpretation for deaf people. The application explores the suitability of the Leap Motion sensor for converting gestures and hand signs into text input. Data provided by the Leap Motion controller, such as hand and fingers positions, normals for every vector used, tip and length for each finger, are converted and fine-tuned to allow proper interpretation of the sign language. The recognized letters and words can then be easily fed into a natural language processing framework to work towards the final goal of providing a highly customizable natural language processing input interface for deaf people.

Keywords: gesture recognition, LeapMotion, sign language

Introduction

In the past years more and more applications have been developed to come in handy to all kind of users with the purpose of making life easier for those who use them. Applications that use artificial intelligence, such as virtual assistants, are one of the main concerns, because nowadays people need programs that understand them, that perform operations for them, thus allowing them to focus on other important tasks. One of the main drawbacks for such virtual assistants is that their natural language processing interface is usually aimed at people who can communicate with them through voice commands. While it is true that this category represents most users, unfortunately this leaves out persons that use gestures and sign language as their only method of communication such as deaf people.

Approximately 18.5% of all people experience some degree of hearing impairment (GBD 2015 Disease and Injury Incidence and Prevalence Collaborators, 2016), with over 23000 in Romania (Centenar Asociatia Nationala a Surzilor din Romania, 2018). One of the major problems deaf people encounter in their daily lives is communicating with ones unfamiliar with the sign language along with that of being unable to use communication interfaces such as speech recognition systems or virtual assistants. Our research tries to address this problem and aims at overcoming it, allowing users that are hard of hearing to have access to this kind of personal assistive technology, providing the choice to use personal assistants through sign language. Therefore, we aim to provide a bridge between deaf people and other applications that use natural language processing through vocal interfaces, so that it can give everyone, regardless of their mode of communication, the choice on how to interact with it.

Sign language is a way for deaf persons to express themselves and to interact with people around them. It uses hand movement in three-dimensional space along with other body parts to convey meaningful information. A sign language recognition system aims at accurately transforming sing language into text or speech. Such systems are highly complex and mostly consist of four essential components: gesture modelling, gesture analysis, gesture recognition and gesture-based application systems (Qutaishat Munib, 2007). Our research is focused on implementing a small part of such a complex system, mainly the gesture recognition, investigating the suitability of the Leap Motion controller for such a complex task. Therefore, we will limit ourselves at implementing sign recognition using a standard sign language with isolated letters that can be used and understood by a larger group. Because every country has its own sign language, it is difficult to integrate a common sign language known to all deaf people, thus, for the moment, we will focus on the Romanian sign language.

The application uses the Leap Motion controller to obtain hand gesture data, which are further processed in order to correctly interpret the user's gestures as letters and compose words. The process of detecting letters is defined and improved empirically, while observing and identifying the limitations of this technology. The application works interactively and provides an acceptable detection error rate. Letters are composed into words and words into sentences which can be fed into a Natural Language Processing (NLP) framework to generate actions.

The rest of the paper highlights similar applications, offers an overview of the Leap Motion controller and its capabilities, presents in detail how this gesture-based input controller was used to implement the application and discusses the preliminary test results.

Related Works

There are many applications for deaf people that try to help them in different situations they encounter in everyday life, from applications that turn voice messages into text and the other way around, to those that aid them in conversations with people that do not know sign language, or applications that help them to better understand and interact with their environment. Even though scarce, there also exist applications that try to translate user signs into a coherent conversation, but mostly the other way around.

Hand Talk (Hand Talk company, 2018) is an application that helps people that are hard of hearing. It translates text and audio messages into an animation of sign language using a virtual interpreter, Hugo, thus serving as a mediator between deaf people and persons that are not familiar with sign languages. Users that are hard of hearing can thus have the opportunity to visualize the messages they are receiving, either audio messages or written ones. This however does not offer the possibility of reversed communication or to be integrated with other applications that may use this library to generate signs.

ASL Tutor (Taylor, 2018) is an application for learning the sign language. It is aimed at persons that want to learn the American Sign Language and contains only this alphabet. It uses an algorithm to generate different signs in order to train the user and offers feedback on the correctness of her gestures. The project also uses data from the Leap Motion controller to determine if the user has created the correct sign or not.

SignAll (The SignALL consortium, 2018) is a pioneering commercial solution to automated sign language translation based on computer vision and natural language processing to enable everyday communication between individuals with hearing and deaf people. The prototype uses 3 ordinary web cameras, 1 depth sensor and an average PC. However, the system is at its early stages and the rig's setup is at current time too complicated to allow every-day life usage.

Escudeiro et al. (Escudeiro, et al., 2014) present an overall study that includes the developed model and the experiments performed with an automatic, bi-directional sign language translator between written and sign language. Their project aims to develop and evaluate a model that facilitates access for the deaf and hard of hearing to digital content, in particular to educational content and learning objects, creating the conditions for a greater social inclusion.

Naglot and Kulkarni (Naglot & Kulkarni, 2016) propose a sign recognition system for the American alphabet using LeapMotion while Chophuk et al. (Chophuk, Pattanaworapan, & Chamnongthai, 2018) investigate the problems arising from the similar symbols contained in it, especially fist signs. Chuan et al. (Chuan, Regina, & Guardino, 2014) also experiment with using the LeapMotion sensor for recognizing the American sign language using k-nearest neighbour and support vector machine, with an overall acceptable detection accuracy.

The LeapMotion Controller

General description

The Leap Motion sensor and controller, designed by the Leap Motion team, allows hands, fingers and all their attributes to be detected in real-time, offering useful information that can be used with other programming languages and platforms, like Unity, C++, Java and many others. The Leap Motion API offers a high precision tool for detecting and tracking hands and fingers.

The Leap Motion controller uses two cameras and three infrared LEDs, the latter tracing the light outside the visible spectrum. Orion beta, the main software package for using the controller, provides us with excellent detection within an 80 cm range, being limited by light propagation through space, but offering the possibility of detection even in poor light conditions.

The sensor is orientated such as its Y-axis points upward and, in the standard operating position, it has a field of view of nearly 150°. Data resulted from tracking are streamed via USB as a grayscale stereo image, and the only objects visible are those who are directly illuminated by the infrared LEDs.

Leap Motion Service is the computer software program that processes the images and constructs a 3D representation of the sensor's view. This is also the program that extracts information such as hand position, fingers and many other important data.

Leap Motion specifics for sign recognition

Even though the Leap Motion sensor was not specifically designed for sign language recognition, its detection capabilities and specific attributes make it a good candidate for the job.

The Leap Motion can either be used as a desktop placed sensor or as a head mounted one for applications using Virtual Reality. When placed on a desktop, it uses a right-handed Cartesian system of coordinates, with the origin right above the controller Figure 9. The Y-axis is vertical, with positive values increasing upwards, the X-axis is parallel to the long edge of sensor and the Z-axis has positive values increasing towards the user.

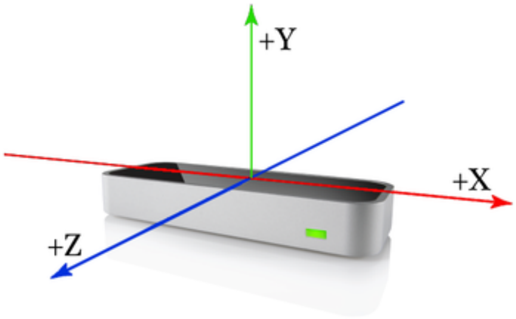


Figure 9- LeapMotion right-handed coordinate system (Leap Motion Inc., 2018)

The Leap Motion API provides measured physical attributes such as: distance in millimeters, time in microseconds, speed in millimeters/seconds and angle in radians. The data model consists of a top-level frame object encapsulating a set of data representing the model of the tracked hands and their properties at one specific moment in time, streaming it to the application.

The hand model gives information about the position, direction and other characteristics and most importantly, gives the list of fingers that are associated with the hand.

It provides data about each finger on a hand like its direction, the tip's position and more (Figure 10). Each finger has an identifier by type for every finger from one hand: Thumb, Index, Middle, Ring and Pinky. If an entire finger or just a part of it is not visible, the controller estimates the finger characteristics using recent observations and the anatomical model of the hand.

Figure 10- LeapMotion hand model: tip position and direction (Leap Motion Inc., 2018)

Because a finger has more bones, the controller also returns the list of all the bones from one finger. Each bone has a unique type that identifies it: Metacarpal, Proximal, Intermediate and Distal Figure 11. The thumb is the only finger that has three bones instead of four. However, to keep a consistent representation of data between different fingers, all of them have the same number of bones and the thumb has the metacarpal size zero.

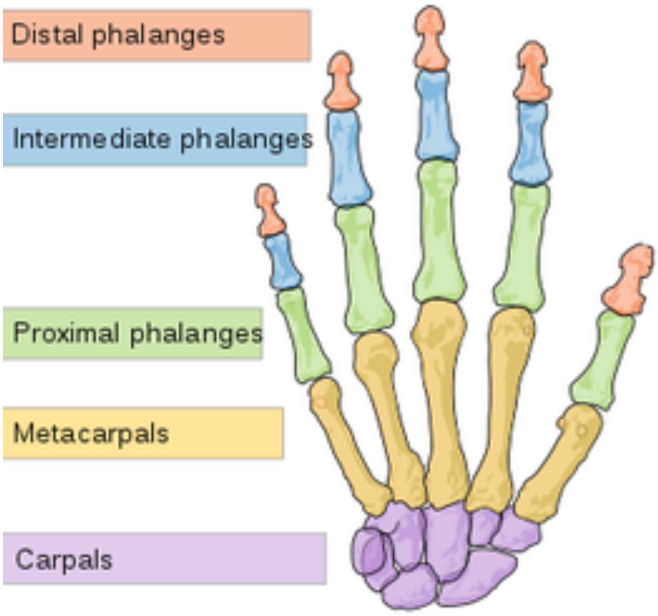


Figure 11- Finger bones and bone types (Leap Motion Inc., 2018)

Implementing sign language recognition

To detect a letter, we need to identify a unique combination of hand, finger and bones attributes that characterizes the letter's specific gesture. For that, we compute the angle between consecutive bones and test for matching configurations. The accepted error ranges for the computed angles was determined empirically, following a large number of experiments. The angle is computed using the orientation (direction) of each bone and since adjacent bones have limited movements, the values are constrained in the 0° and 90° interval.

However, the angle between adjacent of bones is not sufficient on its own to fully characterize a letter and thus must be complemented by the fingers' direction, mostly to remove false positive matching between letters with similar signs (e.g. to avoid confusion between the signs for letters "E" and "S" Figure 12).

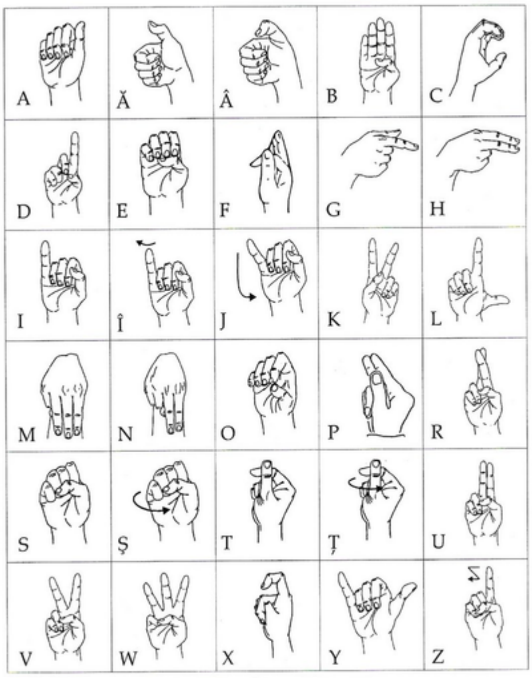
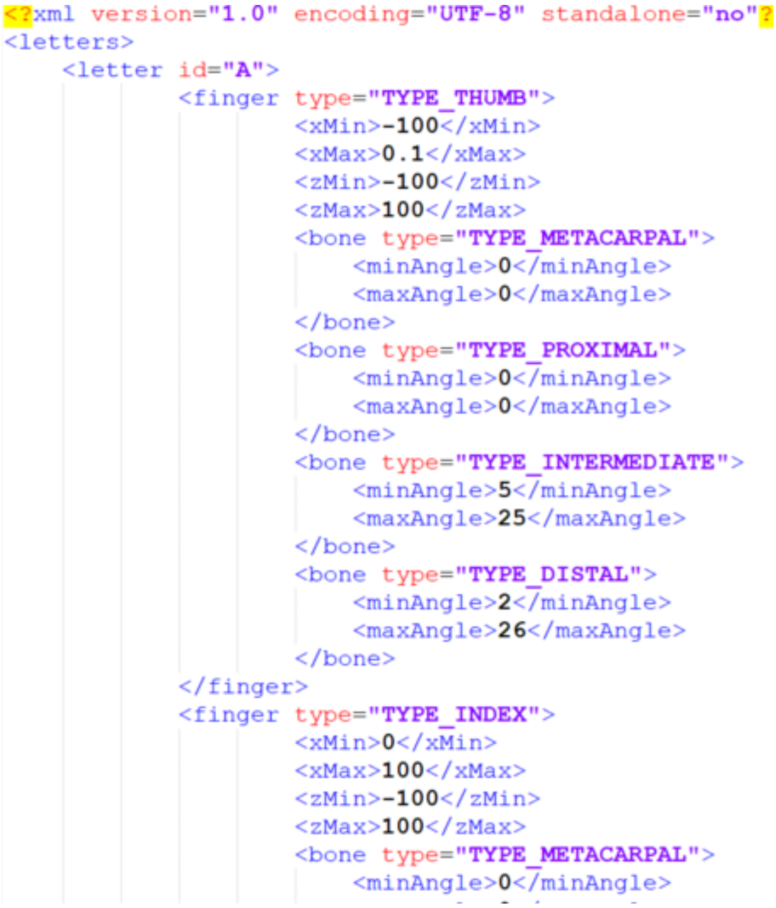


Figure 12 - Romanian sign language (Trufasu, 2018)

Because every letter is defined by its own unique combination between bone angles and finger directions, and this description does not change dynamically, this information is kept in an XML file. This also makes it easy to increase the number of detected and identified gestures in order to extend the application’s feature set, or even to add, either manually, or automatically, new, custom gestures.

The structure of the XML file (Figure 13) includes the parent tag „letters” and contains a set of „letter” tags. Each of these keep the ID, i.e. the name of the letter, and the tags „same” and „finger”. The „same” tag indicates letters that have similar signs and can cause false positive matches and the „finger” tag represents each finger from a hand. For easier identification, each finger tag also holds the finger’s type (e.g. thumb, index etc.). Each finger contains



```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<letters>
    <letter id="A">
            <finger type="TYPE_THUMB">
                    <xMin>-100</xMin>
                    <xMax>0.1</xMax>
                    <zMin>-100</zMin>
                    <zMax>100</zMax>
                    <bone type="TYPE_METACARPAL">
                        <minAngle>0</minAngle>
                        <maxAngle>0</maxAngle>
                    </bone>
                    <bone type="TYPE_PROXIMAL">
                        <minAngle>0</minAngle>
                        <maxAngle>0</maxAngle>
                    </bone>
                    <bone type="TYPE_INTERMEDIATE">
                        <minAngle>5</minAngle>
                        <maxAngle>25</maxAngle>
                    </bone>
                    <bone type="TYPE_DISTAL">
                        <minAngle>2</minAngle>
                        <maxAngle>26</maxAngle>
                    </bone>
            </finger>
            <finger type="TYPE_INDEX">
                    <xMin>0</xMin>
                    <xMax>100</xMax>
                    <zMin>-100</zMin>
                    <zMax>100</zMax>
                    <bone type="TYPE_METACARPAL">
                        <minAngle>0</minAngle>
```

Figure 13- XML data structure

four „bone” children, tags and the minimum and maximum values for the X and Z component of the finger’s direction vector. Each „bone” tag contains the minimum and the maximum angle, in degrees, between itself and the previous bone.

The tags are saved as different models in the application and stored in a Hash Map fast access to the list of fingers and bones and to easily identify the value using the type.

Tracking information from the Leap Motion Controller is fed into the application through a Listener class that implements the onFrame method and which continually receives the hand model’s state. If the right hand is detected, a match is attempted between the hand’s state and all the known signs described in the XML file, on a per finger basis. This allows for early filtering and removal of candidate signs that have no chance of matching the current hand’s configuration. Thus, each finger already tested reduces the number of candidate signs that need to be compared against the hand’s configuration for the remaining fingers.

To detect words from identified letters, the application continuously samples over a 2 seconds interval and evaluates the frequency of each matched letter during this interval. The final match for each sample (interval) will be the letter with highest frequency among all those detected. This allows for a better robustness in detection when matching against letters that have very similar signs.

When a finger is not seen by the sensor the controller estimates its data by considering past states. This may cause false positive matching. Problems mostly occur for signs that have finger directions perpendicular on XOZ plane such as the pairs “A” and “T”, “U” and “R” and “M” and “N”. However, such errors can be easily avoided by very small adjustment of the hand’s position relative to the sensor’s axes. Thus, to decrease errors for the “A” and “T” pair, the hand is to be rotated such as the index finger for “T”s sign is no longer perpendicular on the XOZ plane and is slightly higher than in its representation in Figure 12. For the “U” and “R” pair, the hand should be rotated such that “R”s index finger does not completely cover the middle one, but instead is slightly crossed and oriented a little to the left. For the “M” and “N” pair, the hand should be slightly rotated to the right.

Testing and Validation

The gesture recognition module was tested by the developers to obtain an estimation of its accuracy. Initial tests showed that the Leap Motion controller offers enough robustness in hand gesture detection to be fitted for an interactive automated sign recognition tool. Figure 14 shows the average correct detection rates of our application, computed as the number of successful detections out of a total of 20 attempts for each symbol. Best detection rates occur for “E”, “H” and “W”, while, even with the minor alterations proposed in the previous section, the “N”, “S” and ”U” remain a little problematic, generating more false positive matchings between each other than other letters, mainly because their similitude to other symbols and the approximations the LeapMotion API performs when one or more fingers are not visible. However, empirical results indicate that the greater the alteration from the original XOZ position for all fingers, the less the system will be prone to such errors.

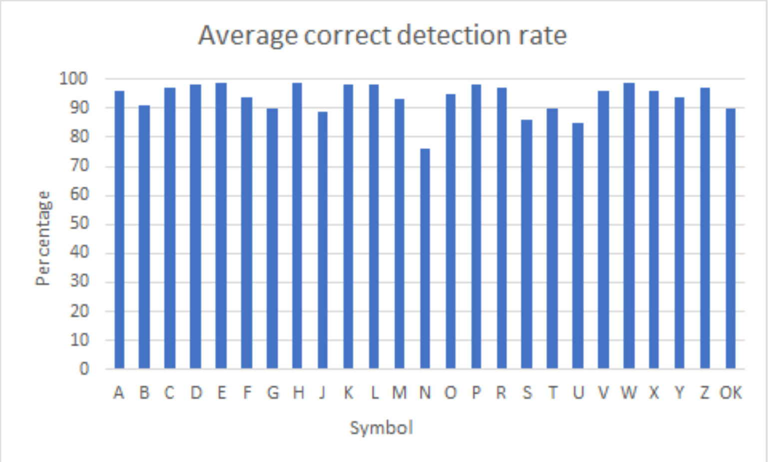


Figure 14 - Average correct detection rates

The main sources of errors in detection are illustrated in Figure 15:

- Fingers oriented perpendicularly on the vertical plane of the LeapMotion sensor are not detected and therefore approximated; In order to reduce erroneous detection rates, the gesture for the “T” symbol should be a little angled upwards instead of directly towards the sensor (Figure 15 – left)
- When not all fingers are visible to the sensor and thus their position

and orientation is approximated (e.g. for the “M” and “N” symbols), the hand performing the gestures should be slightly rotated as to bring as many fingers as possible in the LeapMotion’s range (Figure 15 – middle)

- Almost identical symbols will always generate high detection error rates and thus the gestures must be differentiated by slightly tilting the hand’s orientation (e.g. to help differentiate between the “U” and “R” symbols) (Figure 15 – right)

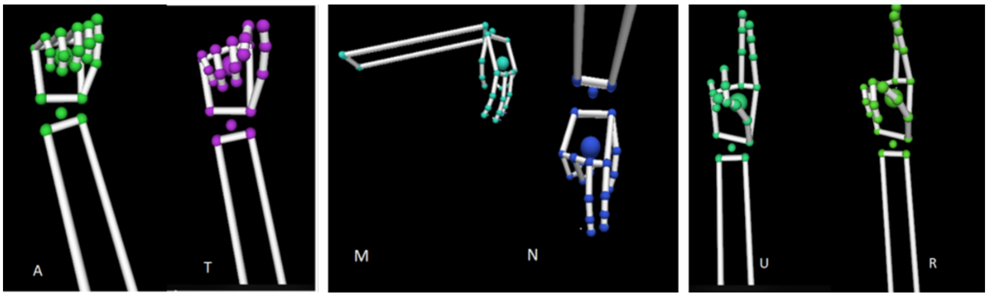


Figure 15- Similar symbols as seen by LeapMotion

Following these more than acceptable detection error rates, word and sentence composition were successful in most cases. Notable problems were observed only with words containing the same letters one after the other or when the user exceeds the 2 second time frame for gesture making. In this case the system experiences difficulties in correctly identifying which one of the aforementioned situations occurred and distinguishing between them.

Conclusions

This paper presented an approach to interactive, gesture-based, sign language interpretation for deaf people. The application assessed the Leap Motion controller’s suitability for converting gestures and hand signs into text. Data provided by the Leap Motion controller, such as hand and fingers positions, normals for every vector used, tip and length for each finger, were converted and fine-tuned to allow proper interpretation of the sign language. Letters were used to form words that can be easily fed into a natural language processing framework to work towards the final goal of providing a highly customizable natural language processing input interface for deaf people. Future work includes testing the application with deaf users to evaluate its accuracy and usefulness and to use Google's Dialogflow as a natural language interface to process the input.

References

Centenar Asociatia Nationala a Surzilor din Romania. (2018). Retrieved from ANSR - Asociatia Nationala a Surzilor din Romania: https://ansr.org.ro/centenar-asociatia-nationala-a-surzilor-din-romania/

Chophuk, P., Pattanaworapan, K., & Chamnongthai, K. (2018). Fist american sign language recognition using leap motion sensor. *2018 International Workshop on Advanced Image Technology (IWAIT)*, (pp. 1-4). Chiang Mai.

Chuan, C.-H., Regina, E., & Guardino, C. (2014). American Sign Language Recognition Using Leap Motion Sensor. *2014 13th International Conference on Machine Learning and Applications*, (pp. 541-544). Detroit.

Escudeiro, P., Escudeiro, N., Reis, R., Barbosa, M., Bidarra, J., & Gouveia, B. (2014). Automatic Sign Language Translator Model. *Advanced Science Letters, 20(2)*, 531-533.

GBD 2015 Disease and Injury Incidence and Prevalence Collaborators. (2016). Global, regional, and national incidence, prevalence, and years lived with disability for 310 diseases and injuries, 1990–2015: a systematic analysis for the Global Burden of Disease Study 2015. *The Lancet, 388(10053)*, 1545-1602.

Hand Talk company. (2018). *Hand Talk*. Retrieved from https://www.handtalk.me/en

Leap Motion Inc. (2018). *API Overview*. Retrieved from Leap Motion Developer: https://developer-archive.leapmotion.com/documentation/java/devguide/Leap_Overview.html

Naglot, D., & Kulkarni, M. (2016). Real time sign language recognition using the leap motion controller. *2016 International Conference on Inventive Computation Technologies (ICICT)*, (pp. 1-5). Coimbatore.

Qutaishat Munib, M. H.-M. (2007). American sign language (ASL) recognition based on Hough transform and neural networks. *Expert Systems with Applications, 32(1)*, 24-37.

Taylor, S. (2018). *ASL Tutor*. Retrieved from https://www.samueltaylor.org/projects/asl-tutor.html

The SignALL consortium. (2018). *SignAll*. Retrieved from https://signall.us/

Trufasu, R. (2018). *Limbajul Mimico Gestual*. Retrieved from Despre Surzi: http://despresurzi.blogspot.com/2009/03/limbajul-mimico-gestual.html