The City College of New York

Grove School of Engineering

American Sign Language Translator using Gesture Recognition

By:

Eun-Jung Park

Namita Sehdev

Roger Frogoso

Senior Design I

CSC 59866

Prof. Camp & Prof. Wei

Fall 2016

Executive Summary

Our team has composed a report on a potential system design that will translate American Sign Language numbers, letters, and a set of words and phrases to English. We are addressing the problem of the unique language barrier between the deaf population, who uses sign language as their primary way to communicate, and the rest of the verbal speaking population. The lack of a fully functioning sign language translator that is available to the public puts the deaf population at a severe disadvantage. Our report includes a summary of some of the most recent works on creating an American Sign Language translator with the use of gesture recognition. We discussed the current state of the field by mentioning the technology and approaches involved, as well as the most current and prominent inventions that serve the same purpose of our design. Out of all the devices available in the field, we have chosen to work with a regular camera input device because it is the most accessible option out of all the devices; and as it turns out, there are no systems for this input device that can translate the non-static gestures for ASL. Our technical approach is broken down into three major sections: gesture detection, gesture tracking, and gesture recognition. The gesture detection section describes the use of image processing techniques to discriminate and extract the hands, face, and body from it's background. The gesture tracking section extracts information about the relative position and movement of the hand, which provides crucial information for non-static gestures. The gesture recognition technique section describes the use of machine learning to accurately determine the meaning of gestures despite deviations from the input signs. We intend to create a functioning real-time executable program for computers; however, if we can finish before our timeline, we would like to develop an application version for the Android and iOS platforms as well.

Contents

1. Problem Definition

1.1 Problem Scope

The ability to communicate is more than just a fundamental right; it is an integral catalyst for the development and growth of any social culture. In our culturally diverse world, a significant amount of effort is invested in aiding the communication among cultures with different languages. Machine translation, for example, can be traced back to the 1950s and has been a growing field ever since. It is defined as "computerized systems responsible for the production of translations with or without human assistance."[1] This field of study has provided us with many useful, albeit imperfect applications such as online translators; however, most, if not all of these tools only apply to verbal languages.

The deaf community, which is part of the large deaf population that reaches 29.5 million[2] in the United States alone, possesses a unique culture that uses sign language as their primary way of communication. Sign language is a visual language that uses gestures instead of speech; and just like verbal languages, it possesses its own linguistic complexity. Because of the significant differences of gestures and speech, translating between sign languages and verbal languages has proven to be a much more complicated task than the traditional verbal to verbal language translations. While translating verbal languages (in the form of text) to sign languages can use similar approaches to traditional verbal to verbal machine translations, the reverse involves another layer of complexity: providing an input method for sign language gestures. As a result, gesture recognition approaches have emerged to provide a solution for this problem.

The lack of systems that provide translations for sign languages puts the deaf community, and other communities that rely on sign language, at a severe disadvantage. It prevents them from effectively communicating with the vast majority of cultures that use verbal languages.

1.2 Technical Review

While there are numerous sign languages around the world, this section will focus on the field of work on American Sign Language (ASL). There are many disputes on the prevalence and prevalence ranking of ASL, but most sources estimate the ASL user population to be in the

[1] W. John Hutchins. "Machine Translation: A Brief History". 1995.

[2] Matthew W. Brault, "Review of Changes to the Measurement of Disability in the 2008 American Community Survey", September 22, 2009

range of 250,000 to 2,000,000, and rank ASL to be the 3rd to 10th most used language as well as being the predominant sign language used in the United States.[3]

1.2.1 Machine Translation: English to ASL

One of the earliest attempts to develop a sophisticated machine translation system was done by Liwei Zhao et al. from the University of Pennsylvania in the year 2000. Prior to their work, the field of machine translation has neglected ASL largely because it was only recently accepted as a natural language at the time.[4] ASL machine translation systems back then, and even today, used oversimplified approaches such as using the ASL alphabet gestures to spell out english words. Generally speaking, a lot of information was lost in those translations, and they failed to capture the complexity of ASL. One of Zhao's team's main contributions was their insight on the essential design principles needed for translating ASL. They emphasized the complexity of ASL by comparing it to spoken languages. The intonation, pitch, and timing of a spoken language contains information that is lost when converted to text. Similarly, the intensity of the signs and facial expressions while using ASL also convey information that can be lost. This lead to the conclusion that a full natural language processing approach and full graphics in real time are needed to translate english to ASL, which was demonstrated in their work.

Zhao's team described a prototype machine translation system that took into account the linguistic, visual, and special information associated with ASL[4]. Their output goal was a fully articulated 3D human model, which consisted of 80 joints and 135 degrees of freedom. Their approach in making this computer animated output was based on previous work that built computational models of a system called Laban Movement Analysis. Their model took into account the phonology and morphology of ASL to capture its linguistic complexity. Finally, to translate English into this 3D ASL output, their team used a Lexicalized Tree Adjoining Grammar based system to translate between English and ASL glosses.

1.2.2 Gesture Recognition: ASL to English

Unlike machine translation, the technology and resources involved in the gesture recognition field is quite extensive. The main reason being is the broad range of applications gesture recognition is being used in today (i.e. smartphones). It is worth mentioning some of the common input devices that are used for gesture recognition applications: wired gloves (or datagloves), gesture-based controllers, radar, depth-aware cameras, stereo cameras, and regular cameras. Each input device produces its own unique set of data, as well as its own set of approaches for analysis. Our design will utilize a regular camera, and the reason will be included in our "Objectives" section.

[3] Mitchell et al. "How Many People Use ASL in the United States? Why the Estimates Need Updating". 2005.
[4] Liwei Zhao et al. "A Machine Translation System from English to American Sign Language". 2000.

The amount of technical work that involves gesture recognition paired with regular camera input devices is quite underwhelming. A possible explanation could be because of the relatively low amount of information regular cameras provide compared to the other more advanced input devices mentioned earlier. While there is a sufficient framework for analyzing simple gestures using a regular camera input device, which was summarized by Zabulis and his team from the University of Crete Heraklion in Greece[5], it is insufficient for analyzing the more complicated gestures of ASL. To realize why this is so, we will briefly discuss the work of Jayashree R. Pansare and his team from the M. E. Society's College of Engineering in Pune, India, who created a system that can translate the ASL gestures for the English alphabet[6].

The flowchart of Pansare's gesture recognition system is shown in figure 1. There are two main processes in this system: image processing and machine learning. The first step involves multiple image processing techniques to extract meaningful information from the input image. This step is important because it simplifies the data that needs to be analyzed, and by removing the excess noise from the background, it increases the accuracy of recognition. The second step uses a feature matching recognition technique, which compares the input data to a set of training data. The method for comparison uses Euclidian distance, which is an example of a classification machine learning technique, namely the k-Nearest Neighbors. The system was tested with 2600 samples of hand gestures, more specifically 100 samples for each alphabet, and was able to successfully recognize 90% of the samples.

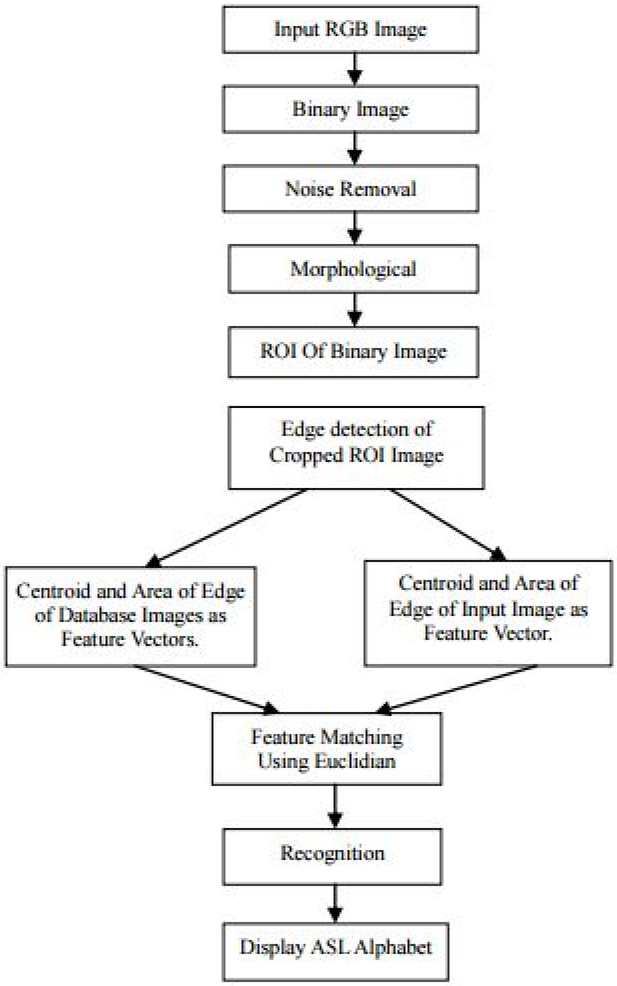

Input RGB Image
Binary Image
Noise Removal
Morphological
ROI Of Binary Image
Edge detection of Cropped ROI Image
Centroid and Area of Edge of Database Images as Feature Vectors.
Centroid and Area of Edge of Input Image as Feature Vector.
Feature Matching Using Euclidian
Recognition
Display ASL Alphabet

Figure 1: Pansare's Gesture Recognition System Flowchart

[5] Zabulis et al. "Vision-based Hand Gesture Recognition for Human-Computer Interaction". 2009.

[6] Pansare et al. "Real-TIme Static Hand Gesture Recognition for American Sign Language (ASL) in Complex Background". 2012.

While Pansare and his team's system provides a rough framework for building a gesture recognition system for ASL, it is incomplete for a number of reasons. First, their system only analyzes static gestures; the ASL gestures for the English alphabet does not involve any movement (with the exception of J and Z), however, most words and sentences in ASL utilize a significant amount of movement. Furthermore, their system only takes in consideration the image information of the hands, which is sufficient for the alphabet gestures. However, in ASL, the position of the hands relative to the face and body contains information about the meaning of the gesture. This analysis suggests the need for a tracking system that will extract information based on the movement of the hand and relative position of the hands and other parts of the body.

1.2.3 Current ASL Translation Systems

There are currently two promising inventions that serve the purpose of translating ASL into audible English: SignAloud and MotionSavvy. SignAloud, made by Thomas Pryor and Navid Azodi from the University of Washington, is a system that uses gloves to translate ASL into English text and speech[7]. The gloves contain sensors that extract and record data on the movement and position of the hands. This data is then sent to a computer via Bluetooth, and after analysis and translation, is output through speakers. On the other hand, MotionSavvy is a device that uses specialized (Leap Motion) cameras, which utilizes two cameras to create a real-time 3D representation of the user's hands[8]. The device is able to translate the gestures detected by the Leap Motion camera into text, and also provides an audible output. Although the availability of both these inventions are still in the works, it is worth noting that there are no other ASL translation devices that are currently available in the market.

[7] Lemelson.mit.edu "Thomas Pryor and Navid Azodi". 2016

[8] Motionsavvy.com "MotionSavvy". 2016

1.3 Project Objectives

Our team aims to expand the current ASL gesture recognition translation systems that use regular camera input devices. Among all the input devices used for gesture recognition, a regular camera is arguably the most accessible and convenience as it is shown in the table 1[9]. Ideally with this approach, any electronic device with a camera (specifically smartphones) would be able to use a sign language translator program. Because of the large scope of ASL, and the time constraints for the development of this project, our team will limit the scope of our project; our system will use a computer webcam to detect and translate ASL gestures to English text and speech. Because of the magnitude of ASL, we will limit the gestures to the alphabet, numbers, and some common words and phrases. This will allow us to develop and understand gesture recognition for static gestures (alphabet and numbers), which will provide more insight in developing non-static gesture recognition (words and phrases). We will first create an executable computer program, and if time permits, an android application.

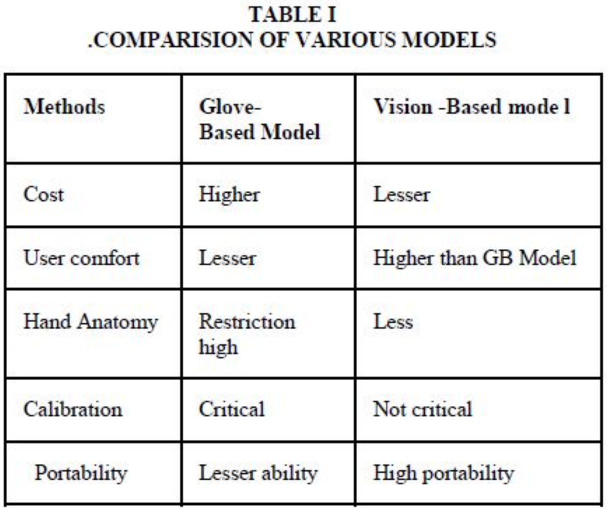


TABLE I
.COMPARISION OF VARIOUS MODELS

Methods	Glove-Based Model	Vision -Based mode l
Cost	Higher	Lesser
User comfort	Lesser	Higher than GB Model
Hand Anatomy	Restriction high	Less
Calibration	Critical	Not critical
Portability	Lesser ability	High portability

[9] R. Pradipa et al. "Hand Gesture Recognition - Analysis of Various Techniques, Methods and Their Algorithms". 2014

2. Technical Approach

To achieve our objective of creating a system that can translate the ASL gestures for numbers, letters, and common words and phrases, we will first implement a system that is similar to the work of Pansare and his team. This will imply implementing a system with a phase that can detect and process an image (which will be discussed in section 2.1), as well a phase that can extract the meaning of, or recognize the gesture (which will be discussed in section 2.3). However, as mentioned in the technical review, these approaches need to be modified to account for the movement and relative position of the hands when analyzing words and phrases. This may imply the need to include the detection and processing of the face and body, as well as including them to the the training data that will be used for the gesture recognition phase. Furthermore, to obtain and analyze information on motion and relative position, a new phase for tracking the hands will also be developed (which will be discussed in section 2.2).

2.1 Gesture Detection

Because ASL heavily relies on hand gestures, the first step in our approach to recognize our target gestures is to detect the hands. Our goal in this phase is to discriminate and isolate the hands from the rest of the background. This is a necessary step because it simplifies the data that will be used for interpreting the gesture. In order to discriminate and isolate the hands from the image, we will use image processing techniques to derive our target visual features; skin color, silhouettes, and contours. (Zabulis et al.)

Skin color is a convenient feature because it is usually distinct in hue-saturation space, and is not greatly affected by changes in lighting. Techniques that differentiate skin color from its surroundings usually rely on histogram matching or tables that consist of training data for the skin. One drawback of this feature, however, is the variability of skin color in different lighting conditions, which can lead to failures in, or falsely detecting skin color; ligth skin colors may be harder to detect in bright lighting conditions, while dark skin colors might be too similar to their background in dim lighting conditions. (Pavlovic et al.)

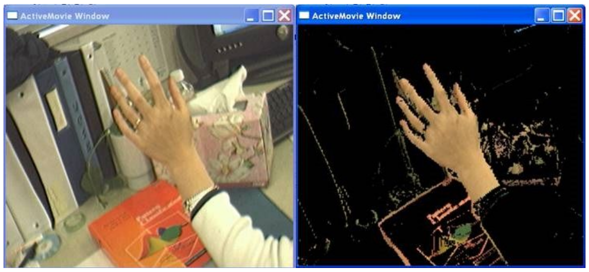
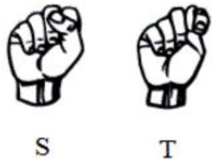
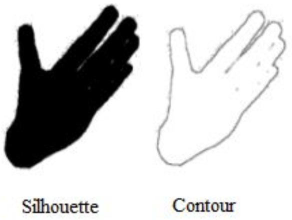


Figure: An example of hand segmentation based on skin color by Four Eyes Lab from the University of California, Santa Barbara (2012)

Silhouettes are another parameter of interest because it is very simple and easy to extract. The only drawback with this approach is the risk of losing information. Especially, the gestures for the alphabet S and T are all very similar. The only difference is in the placement of the thumb. If the silhouette of the hand was taken it might not be possible to determine which letter is being presented.



Contours are similar to silhouettes but focus on extracting the edges on an image. Some methods derive the edges from silhouettes, making them equivalent; however, edge detection techniques can also be applied to colored and gray-scale images.



We will overcome the drawbacks of each features by applying those them in layers when necessary, but applying those layers flexibly for the performance.

To achieve these image processing techniques, we will be using OpenCV. OpenCV is a library of programming functions mainly aimed at real-time computer vision. The detailed process will be done in three steps. First, extract the region of interest from the image obtained from web camera. If we continue the explanation with an example for the case that we use silhouette feature, it can be done by turning the image obtained from camera input into gray level. A threshold will be applied to make the image simplified silhouette removing the background. Secondly, the we will find the contour of the image to approximate the location of hand using the contour function on OpenCV. On third, based on the contour and silhouette of hand gesture, we will draw convex hull and find convexity defects for further process.

2.2 Gesture Tracking

Other than detecting hand gesture, we need to keep track of the hand for the interpretation of non-static ASL. The tracking target are mainly divided into two; movement of hand location and movement of fingers.

For the hand location tracking, the information that was obtained during gesture detection like convexity defects will be used. Since the center of the palm is a good representation of hand location, tracking the hand location will be done by tracking the position of a palm. Using the fact that the defect points are approximately located in the middle of palm, we will get x and y for the center of palm by averaging the x's and y's of the convexity defects, which are the deepest points of valleys between the fingers. By executing this series of image processing and calculations repeatedly in few seconds, we will keep track of the movement of hand location.

Another important movement that we need to find out is that of fingers. For this movement, we need to find out the angles between fingers as well as the location of fingers. For this, we will apply linear fingertip model assuming that the most finger movements are linear and comprise very little rational movement.[10] The locations for each fingertips were already obtained from gesture detection process through convex points. Using those points, the trajectories of them will be calculated according to the motion correspondence. Then, the fingertip trajectory through space will be represented as simple vectors with magnitude for each. The observed set of records about the finger movement will be labeled to be used later on the gesture interpretation phase to find the corresponding meaning of the gesture.

Most of the ASL keeps its simplicity if it includes some motion in the sign. If the sign is composed of the movement of hand location, there's no change in the shape of fingers during the movement of hand location. If the sign includes movement of fingers, the movement of hand location is minimized to make the finger movement captured by the sign reader. For example, the index finger signing alphabet Z stays straight while the hand movement draw Z on the air. As another instance, while the thumb is moving right and left to sign number 10, the movement of hand location is stabilized. This tendency will be also true for the set of our limit gestures to be translated into English, so we don't need to consider the complex movement case in which the movement of hand location and movement of fingers happening simultaneously.

Z

10

2.3 Gesture Interpretation

There will be two levels of complexity for interpreting gestures: the first level involving static gestures, and the second level involving non-static gestures. We will use template matching for both approaches, with an extra level of analysis for the non-static gestures. The general idea of template matching is a classification type machine learning problem: an input image will be compared to a set of premade templates corresponding with datavalues for each classification. If the input image reaches a certain threshold score, it will be classified as the template with the closest matching score. We will use the Scikit-learn library for its machine learning algorithms, specifically the k-Nearest Neighbors.

[10] R. Pradipa et al. "Hand Gesture Recognition - Analysis of Various Techniques, Methods and Their Algorithms". 2014

2.3.1 Static Gesture Template Matching

The first step for developing our template matching system is to gather training data for each symbol we will use; for static gestures we will need multiple sample images of each number and alphabet symbols, and each template needs to be defined by datavalues. Multiple images are needed for each symbol because the template has to account for the variabilities in shape due to rotation and scale. Multiple techniques exist to cope with rotational and scale variability, such as normalization methods, and the use of multiple view templates. Our team will choose a method that will be the best compromise between computational efficiency and accuracy, relative to our purpose. Once the templates are made, they will be tested with samples of increasing amounts of variability to evaluate how well they can recognize gestures. Another interesting function we could implement would be to “personalize” these templates for specific users by using the specific user’s input signs as training data. This will help increase the accuracy of our system if it will be used for personal use versus public use.

2.3.2 Non-Static Gesture Template Matching

Template matching for non-static gestures will not be as simple because we need to somehow include the information of the movement and position of the gesture to the templates. Given that we are using a regular camera, spatial information can easily be lost in 2D images. One approach we could utilize is to compare each frame to the previous frame to detect changes in position. Given a set period of time, or when a brief pause in movement is detected, we can accumulate the features that involve movement and create a 2D image, which is similar to the time lapse image shown in the figure 2. The formed image can be used to represent the trail of the gesture; and along with the starting and ending position, we could use these three images as potential template categories. Our team will have to test and experiment on multiple possibilities to achieve an accurate and efficient solution.

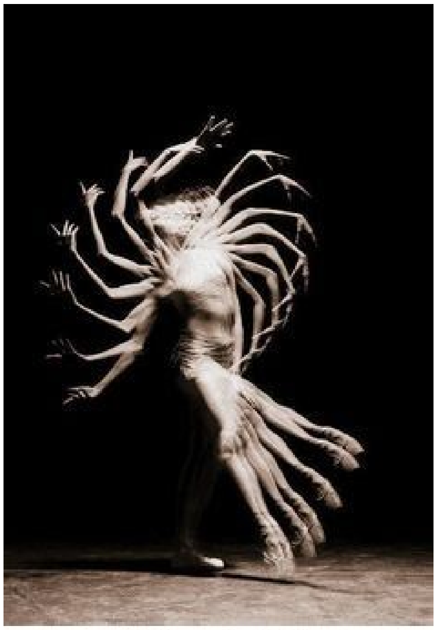


Figure 2: An Example of a Time Lapse Image

3. Project Management

3.1 Deliverables

Our team’s primary goal is to create a computer executable program that does the following functions:

1. Uses a webcam, built in or exterior, to detect a short range surrounding
2. In real-time, use the webcam’s video output as an input to our program
3. Our program will continuously and efficiently perform image processing and machine learning algorithms on this video input, and extract the English translation in text form
4. Our program will be able to translate numbers, letters, and a set of common words and phrases from ASL to English
5. This text form will then be converted to an English speech output through speakers.

Once we achieve this, we will work on improving the efficiency and accuracy of our program. If we can achieve these goals, we will work on a secondary goal of creating an application version of this program. We will work on making an android application before extending it to the iOS platform.

3.2 Timeline

Our timeline provides a rough estimate on our progression for our primary goal. If we are able to accomplish our goals before our intended schedule, will will continue working on the next phases to create time for implementing our secondary objectives.

3.2.1 January - March

The first two weeks of January will be dedicated to further research on choosing optimal techniques and developing a more concrete layout of our program. We will develop a flowchart for each process mentioned in our technical approach. We will then implement this system for translating numbers and letters from ASL to English by the end of February: We estimate spending 5 weeks to develop a working system that can translate numbers, and another 5 weeks or less to implement alphabet translation using similar techniques used for numbers.

3.2.2 April - May

In the beginning of April, we will start implementing word translation. We will choose a variety of commonly used words that can capture a wide range of motion and position variability. This phase will involve rigorous testing and adjustments to our algorithms. Once we establish a working frame, we will start including common useful phrases.

4. Conclusion

Our team hopes to expand the current field on ASL gesture recognition by developing algorithms for detecting and analyzing non-static gestures using a single camera input device. While our project will not fully translate ASL, we hope to develop a framework for single camera input approaches to gesture recognition. While our project only focuses ASL, our design approach can be used for other gesture-based languages as well. Finally, by choosing a regular camera input device approach, we hope to raise awareness and inspire the need for an accessible tool for aiding gesture-based language barriers. An interesting topic to consider is the reverse processes of translating English to ASL. This will involve the area of machine translation and computer graphics, which was mentioned in our Problem Scope Definition.

5. References

[1] W. John Hutchins. “Machine Translation: A Brief History”. 1995.
[2] Matthew W. Brault, “Review of Changes to the Measurement of Disability in the 2008 American Community Survey”, September 22, 2009
[3] Mitchell et al. “How Many People Use ASL in the United States? Why the Estimates Need Updating”. 2005.
[4] Liwei Zhao et al. “A Machine Translation System from English to American Sign Language”. 2000.
[5] Zabulis et al. “Vision-based Hand Gesture Recognition for Human-Computer Interaction”. 2009.
[6] Pansare et al. “Real-TIme Static Hand Gesture Recognition for American Sign Language (ASL) in Complex Background”. 2012.
[7] Lemelson.mit.edu “Thomas Pryor and Navid Azodi”. 2016
[8] Motionsavvy.com “MotionSavvy”. 2016
[9] R. Pradipa et al. “Hand Gesture Recognition - Analysis of Various Techniques, Methods and Their Algorithms”. 2014