Sign Language to English

Slate8

Final Report

by

Nathan Kebe El

Faculty Advisor: Dr. Mohamed Chouikha

Electrical Engineering and Computer Science

Howard University

Summary

The background for the American Sign Language communication was little or non-existent. The hearing impaired were discriminated against, left out or forgotten, and felt a lack of confidence or a burden or obligation to others. The National Association of the Deaf (NAD) gave the hearing impaired a voice, a sense of belonging, restoring a sense of confidence, and a degree of independence. Still, communication between the deaf was with each other or through interpreter. In the current state of the art idea thinkers (designers and inventors) said to themselves “How can I make communication between the hearing and hearing impaired better”? Well, they used audio or video in standalone devices. The devices were portable and convenient, and friendly”. Design cost constraints for sequencing software was small due to open source software available on the internet. Design cost constraints for hardware (Phone), hardware (camera, speaker, and microphone) was held to a minimum. Slate8 ASL is ADA Compliant – Adheres to ADA section 508, and WCAG 2.0 – Handicap Accessible. The final device will have a large screen display and color compliant, and have multimedia capabilities. In addition, the final product will be easy to navigate, designed using Accessibility Frameworks, text – to – speech, haptic (touch), and gesture technologies, safe for all ages, and most of all legal – does not interfere with other apps and hardware. Slate8 Top Level Design targets direct line of sight of the “Signer”, both hands and body are within view. Slate8 ASL software uses real time conversion of ASL to text and audio. The portable device has a larger onboard memory for multiple language conversions and contains ASL symbols and video tutorials. At present, slate8 ASL software is only able to recognized single syllables of the ASL symbols.

Problem Statement

The need to communicate with the Deaf Community or people experiencing hearing loss, or those who can't hear and need to communicate with those who can hear. The Sign Language App will serve both groups of people.

Team Project’s 2017-2018 Academic Year Goal: To build an app which converts ASL to text or voice.

Approximately twenty eight million hearing impaired individuals in America feel that one of their biggest difficulties in dealing with those who can hear as being looked at and treated by them as a handicapped individuals. According to the PBS documentary, "Sound and Fury", a fairly large percentage of deaf people---especially those deaf from birth---would opt out of the hearing world, no matter what current or future technologies will offer. The core principle of the slate8 project is to build an app that will work on any device to help integrate the deaf/mute community into the mainstream of society. By developing a standard app that is capable of understanding, translating, and communicating with the hearing and non-hearing world in any and all earthly languages. The app will be easy for all to use and convenient.

Design Requirements

Design Requirement Form		
Date:	October 4th, 2017	
Design Project Title:	Sign Language App	
Team Name:	Slate 8	
Team Advisor	Dr. Mohamed Chouikha, PhD.	
Team Assistant	Vanessa Galani	
Project's Long Term Goal	A Working Viable Sign Language App	
Project's 2017-2018 Academic Year Goal	A Working Viable Sign Language App	
Team Members (Design Class)	Nathan Kebe El	
Team Members (Others)	None	
Requirements	**Descriptions**	**Source**
Background (NEED)	The need to communicate with the Deaf Community or people experiencing hearing loss, or those who can't hear and need to communicate with those who can hear. The Sign Language App will serve both groups of people.	National Association of the Deaf - NAD
Objective (Problem)	To create a means of communication between the hearing and hearing impaired community.	Slate8
Performance	The sign language app once activated and focused on direct line of sight target, will automatically switch to the proper mode of operation: • Sign symbol to text display interpretation • Voice to text display interpretation The sign language app is equipped with such features as: • Symbols to text • Text to speech • Speech to symbol and text The Sign Language App comes with: • Database/storage unit consisting of the most general sign language symbols The Sign Language App is equipped with: • Search • Share • Store • Copy • Video copy/playback • Print	Sign Language App, HMM (Hidden Markov Models for Gesture Recognition). Python 2.7 for Windows 10.

Cost	Java Text, Java Runtime Environment, and App Inventor 2: Create Your Own Android Apps Oct 23, 2014 by David Wolber and Hal Abelson Kindle Edition $ 18.35 (Note: these text are starter text!)	Amazon
Safety	When using the app it is safe to use, does not contain upsetting, offensive content, will not cause any damage to their device, and can not cause any physical harm when used.	None
Compliance	The FCC's rules require any manufacturer certifying a device under the new process to take steps to prevent “unauthorized” changes to the software on the device that might alter its radio frequency and power parameters in a way that takes it out ofcompliance with the regulations known as FCC Part 15 regulations.Jul 6, 2007	FCC Website
Driver-Vehicle Interface	N/A	None
Energy, Power, and Environment	1. Sending or and receiving email over mobile network: 610mW 2. Video playback: 454mW 3. Sending and receive email over Wi-Fi: 432mW 4. Audio playback: 320mW 5. Sending a text message: 302mW	Internal battery.
Intellectual Property	**Anupam Yedida**	Get Hub
Size and Weight	N/A	None
Deliverables	Sign Language Software with single character symbol recognition only.	Single Syllables Only.

Constraints	Cost for tracking software is little to no cost. Open source software may be available or gain rights for design. Cost for sequencing software is little to no cost. Open source software may be available or gain rights for design. No cost for hardware (Phone), hardware (camera, speaker, microphone). Need specially design motions sensor for watch. Need specially design watch case for sensors. Little to no cost for Java app creation for ASL app.	Android App Development Website.
Standards and Regulations	ADA Compliant – Adheres to ADA section 508, and WCAG 2.0 – Handicap Accessible. Large screen display and color compliant. Multimedia capabilities. Easy to navigate. Design using Accessibility Frameworks, text – to – speech, haptic (touch), and gesture technologies. Safe for all ages. Legal – does not interfere with other apps and hardware.	FCC Website

Current Status of Art

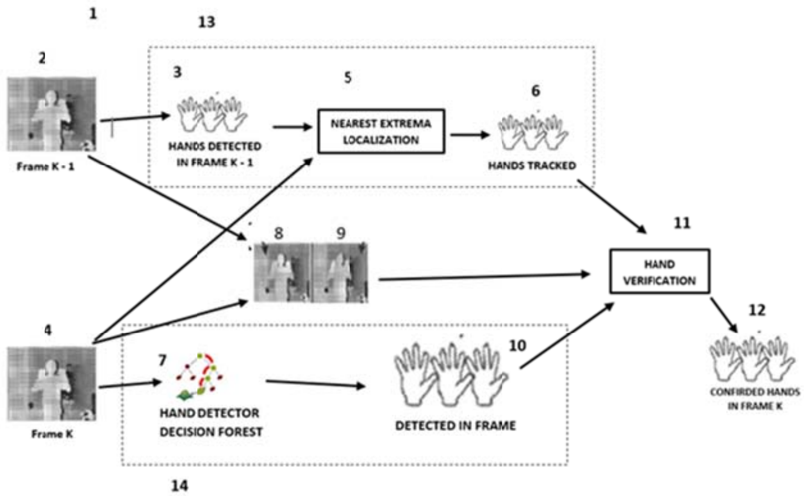
American Sign Language Apps teaches symbol identification through video recorded playback and give early warning detection of hazards. American Sign Language devices can be used as individual recording devices as base stations in group meetings which when used incorporate Automatic Speech Recognition software. At present, sign language gloves are used which incorporate wireless communication into text displayed on screens. This glove controls a virtual hand to mimic sign language gestures.

1. For the current state of the art, there are apps and devices which teach basic American Sign Language. These apps and devices contain video tutorials for learning proper hand and finger positions and facial expressions. An important safety feather of these apps and devices is early warning detection. For example, a fire alarm will cause the app to give off a series of vibrations and display warning text.
2. Another state of the art sign language device is designed for group meeting. Individual electronic devices send and receive text and audio to both hearing and hearing impaired members of the meeting. The base station is installed with Automatic Speech Recognition software which does the conversions between text and audio.
3. Finally the American Sign Language Glove wirelessly translates the American Sign Language alphabet into text by controlling a virtual hand to mimic sign language gestures.

Top Solution Design and Solution Approach

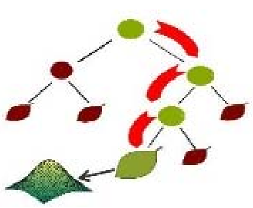
The figure 1 below, illustrates an exemplary process (1) for performing three dimensional hand tracking using depth sequences, in accordance with some embodiments.

First, the depth image for a previous frame (Frame K−1) is analyzed at Step (2). The depth image data for the previous frame may simultaneously be passed to tracking process (13) as well as motion detection process (8) and background modeling process (9)(for further verification. Within tracking process (13), the set of hands detected in the Frame K−1 are obtained (3).



Next, the hands from Frame K−1 and Frame K (i.e., the "current frame") may be subjected to a nearest extrema localization constraint (5), which effectively searches the area surrounding the location of the hand in the previous frame and constrains any hand candidate that is detected in the frame to be located at the nearest extrema of a foreground body object, otherwise it can be discarded. The hands that pass the nearest extrema localization constraint at step (5) are the confirmed set of hands that will be tracked (6) from Frame K−1.

Like the previous frame (Frame K−1), the information from the current frame (Frame K) may be passed to motion detection process (8) and background modeling process (9). The result of processing the previous and current frame using motion detection process (8) and background modeling process (9) is that the portions of the frames where a valid hand may be tracked are limited to only "moving pixel areas" within the frame, as well as pixels that are deemed "foreground pixels" within the frame. These two constraints come from the insight that true positive hands in received images are almost always both in the foreground, as well as moving. (Note: the motion detection process (8) and background modeling process (9) may "look back" a set number of frames, e.g., a few seconds' worth of frames, to determine whether there is a high enough probability of movement in a particular region of the image to deem it a "motion region" or a "background region.")



Finally, each frame, as it becomes the "current frame," is subjected to the hand detection process 565, which may involve the background-invariant hand detector decision forest (7) discussed in greater detail above, resulting in a set of candidate hands (10) detected in the "current frame," Frame K. Note that there may be some new hands in Frame K that were not present in Frame K−1, or some hands that were present in Frame K−1 that are not present in Frame K. This output of current frame hand detection process (14) is then passed to hand verification stage (11), along with: 1.) the output of motion detection process (8) and background modeling process (9) that limit the potential parts of the

frame where valid hands can appear to the moving portions of the foreground; and 2. The output of the hand tracker (13) from the previous frame.

At hand verification step (11), the detected (10) and tracked (6) hand candidates are again verified to make sure that they are likely to be hands. This process again leverages the fact that the hand is most often attached to the body with “single directional connectivity,” i.e., located at the end of an arm that is connected to the user's body via only a single connection point. This “single directional connectivity” check may be implemented by drawing a circle around the detected hand and checking the intersections with the foreground body mass of the user whose hand has been detected. If the detected hand is indeed a real hand, there will only be an intersection with the foreground body mass in one direction.

Additional verification steps would include ruling out candidate hands located in the background of the scene or in regions of the scene where there has not been any movement over a predetermined

amount of time. Finally, hands may be verified by tracking their IDs from frame to frame and removing those hands in the current frame that show unusual movement characteristics. For example, if there are two hands in the frame for many consecutive frames, and then there are suddenly six hands in the current frame, there is a high likelihood that four additional hands in the current frame may be false positives. Likewise, if Hand #1 has been on the left side of an image for many consecutive frames and Hand #2 has been on the right side of an image for many consecutive frames, it is unlikely that, in the current frame, either Hand #1 or Hand #2 would suddenly move all the way across to the other side of the image (i.e., move more than the distance by which a human subject could typically move their hands in the time it took the image sensor to capture the successive frames). Finally, those hand candidates that pass the hand verification step (11) are output as confirmed hands in the current frame, Frame K, (12).

A random decision tree from a random decision forest, in accordance with some embodiments, e.g., the random decision forest that is created during the hand detector training process or the random decision forest (7) that is used during the hand detection process once it has been trained. Each decision tree comprises a root node a plurality of internal nodes, called split nodes, and a plurality of leaf nodes.

Implementation Process

For my teams solution design is the extraction of human hand properties based on OpenCV then implement the proposed solution into a portable smartphone app that is user friendly an handicap accessible.

1. The human hand is captured by cameras and the images are converted into their digital representations. These digital images are made up of a matrix of scalar and vector values and the OpenCV library contains algorithms to process these images.
2. The contour of an object is defined by set of points, which describe the edge of object or its outline. For example, the contour of a tennis ball is a circle. OpenCV library contains a number of contour finding algorithms, which show approximation of contours and use extraction techniques.

3. Such contours represented as a set of points, can be enclosed inside an n-dimensional polygon, also known as hull. A hull, being a polygonal structure can be concave or convex in its shape. If a line can be drawn inside to its boarders then it is concave in shape and contains convexity defects. The human hand contain a large number convexity defects between fingers.
4. The creation of a threshold image is very crucial for Hand detection. Isolating the foreground from the background is essential as we want the hand to be in the region of Interest.
5. The hand is identified using OpenCV's contours analysis techniques which are then returned as an array of co-ordinates of the human hand.
6. Data from the contour analysis technique is then manipulated to obtain an entity known as "The Number of Convexity Defects". Based on the value of the convexity defects, it is possible to identify how many fingers are present and help to identify the symbolic letter that is being presented.
7. The number of contour defects is calculated by the following process. When the triangle is computed let the sides be a, b, and c. The triangle is formed by the starting point of the contour, the ending point of the contour and the farthest point of the contour (a, b, and c respectively).

   Angle 'a' is computed by this following formula:

   a = math.sqrt((end[0] - start[0])**2 + (end[1] - start[1])**2) [7]

8. Angle b and c are calculated in the same way and then the Cosine rule is applied. If the angle A is less than or equal to 90 degrees, it means that there is a convexity defect. Once a convexity defect is recognized, a variable by the name "count" increments by one. So, by an algorithm technique the number of convexity defects can be identified.
9. For identifying A, the difference between the area of a circle and the area of the contour was calculated. The circle is obtained by bounding the contour. The reason this method is adopted for A is that there is very little difference between the two areas (mentioned above) which makes the Letter A stand out from the other letters. Hence this algorithm was found to be very efficient.
10. For the letter B, the area of the contour was calculated. This method is adopted because the Letter B has the largest area among the other letters.
11. For letters C, and L if the number of convexity defects are equal to 1, the "angle" is calculated. OpenCV's inbuilt function then calculates the overall figure's orientation giving the angle and based on these values the letters C, and L are identified.

Conclusion

In conclusion the principal goal of this project is the recognition of American Sign Language Symbols and the extraction of features of the human hand using of OpenCV and Python software. At present 13 symbolic letters are recognized by this technique.