A Sign-to-Speech Glove

Olga Katzenelson
Solange Karsenty
Hadassah Academic College
HaNeviim 37, Jerusalem, Israel
olga.katzenelson@gmail.com
solange@hadassah.ac.il

ABSTRACT
In this paper, we describe a smart glove – JhaneGlove - that turns sign language gestures into vocalized speech via a computer to help deaf people communicate easily with people who do not understand the sign language. We have developed a handmade device: a glove with sensors, and the software that will transform signs into text. Text is converted into speech using standard software. A neural network based agent can learn the gestures interactively and allows users to define new signs. As a result, each user can have a custom sign language independent from other users. We present the system and early experimental results.

Author Keywords
Sign language, neural networks, hand-shape recognition, gesture recognition, glove

ACM Classification Keywords
K.4.2 [Computers and Society]: Social Issues—Assistive technologies for persons with disabilities; H.5.2 [Information Interfaces and Presentation]: User interfaces—Input devices and strategies (e.g., mouse, touchscreen)

INTRODUCTION
Our environment is continuously evolving with new computerized and electronic systems that are changing the way we live and interact with others. Electronics become smart, and sensors are everywhere, thereby enabling the development of devices that can analyze, recognize and interpret our actions. Communication is perhaps one of the major concerns when developing such systems, and it is a major concern at the Hadassah Academic College where projects of the Computer Science department can be initiated in direct collaboration with the Communication Disorders department.

While most deaf individuals with various degrees of deafness can lip-read and understand someone speaking, they almost always communicate with sign language. This language is usually unknown by people around them. A typical situation is a person (student) who wants to use sign language to give a presentation to other students who do not understand the sign language. To enable this type of communication, we have built an experimental system for the automatic translation of sign language into text and speech.

PREVIOUS WORK
A sign is a language that uses manual communication and body language to convey meaning. This can involve simultaneously combining hand shapes, orientation and movement of the hands, arms or body, and facial expressions to fluidly express thoughts. Some systems attempt to recognize as a whole the body language, for example using the kinect [1] that combines cameras and microphone to capture 3D motion and voice, enabling a wide range of applications. Other systems, based on more portable devices such as a glove, focus on the hands only.

The hands are used in sign language to code all letters of the alphabet (Figure 1). This is the primary means of communication for the deaf. We are focusing on this type of sign language, aiming at recognizing all letters to enable the constructions of words and sentences. Furthermore, we generalize the concept of a letter to represent a word or sentence as well: the signs can be mapped to a letter, a word or even a sentence.

Figure 1. The Sign Alphabet

There have been previous attempts in solving the problem of sign language translation with a glove. A team of Ukrainian students (in the Software Design Competition of the 2012 Microsoft Imagine Cup) developed a glove to turn signs into text [3]. Their system relies on many sensors, including sensors between the fingers allowing very precise gesture recognition. It runs on an Android device, at relatively low speed, and the gestures are translated into text.

Similarly, we have built a custom glove with fewer sensors, but with an accelerometer and gyroscope, and greater flexibility since we can reprogram and run pre-processing computations on the glove itself. Also, the accelerometer and gyroscope helps us detect hand orientation, therefore enabling recognition of rotated hand signs.

Calibration of the glove is an essential initial process to ensure precise gesture recognition. It can be a time consuming operation, particularly for deaf users who need clarifications on about how they should be positioning or moving their hands.

Another glove system aiming at accurate calibration [4] uses a Cyberglove 22 flex sensor for gesture recognition. The system is built on 22 predefined gestures and prompts the user to learn how to operate the glove using a simulator.

Our system is different from both prior attempts in two respects. First, in both prior systems the set of gestures is predefined (after some initial calibration). Our goal is to provide a generic system that enables the user to define any other gesture or sign language. Furthermore, each user's set of signs is stored and enabled after authentication.

Second, the calibration process for the Cyberglove is rather cumbersome and requires the user to repeat 44 gestures. Our system on the other hand has very simple calibration process from a user perspective. The user only needs to move his hand in all directions, and bend and unbend the fingers.

Note that there are a number of systems for sign language translation that have been developed for the kinect [1,2]. The kinect offers very accurate sensors. Systems using the kinect require to be set up in permanent areas such as classrooms, while our vision is to build a wearable device that can easily be carried around. Therefore we focus on hands only, and we rely on fewer sensors provided with a custom made glove.

THE JHANEGLOVE

The human hand has 27 degrees of freedom: four in each finger, three for extension and flexion and one for abduction and adduction. The thumb is more complex: it has five degrees of freedom, and six degrees of freedom for wrist rotation and translation.

The JhaneGlove (figure 2) was built using three types of sensors: 5 flex sensors, 8 contact pads sensor, and an accelerometer and gyroscope. Each flex sensor has three states. Each sensor can be turned on or off. It enables 20 degrees of freedom, allowing more than a hundred different gestures. The flex sensor is a device that changes its resistance proportional to its form (figure 3), therefore allowing the detection of a bending movement when placed on the finger. The contact pads allow identifying the contact between fingers and the hand palm, such as the "Y" sign.

Finally the gyroscope allows identifying hand motion and thereby the recognition of gestures based on hand orientation.

This last feature is unique and makes our system more flexible in terms of recognition, as opposed to others. We can track hand movements and rotation as part of the gesture. For example the A sign can be trained and recognized in different orientations.

THE SIGN LANGUAGE

What makes our glove different from other systems is its ability to easily create custom sign languages. Most systems focus on implementing the conventional sign language. Not only we can support this language, but we can also train and store one different language per user.

As a trade-off between the recognition precision and time needed to train the system, our system currently allows the definition of 30 signs that can be processed by the system's recognizer. In addition to these signs, we have the *space* sign that enables marking the end of a word, and the *dot* sign that marks the end of a sentence. These are necessary for the recognizer to form the words. Once the words are captured, the text-to-speech agent can convert them into speech.

If the system is used to enter the 26 letters of the English alphabet, then four more signs remain that can be assigned to frequent occurring words or even full sentences. If the system is used to define a domain-specific language, then one can assign full words or sentences to the 30 signs and create a custom sign language.

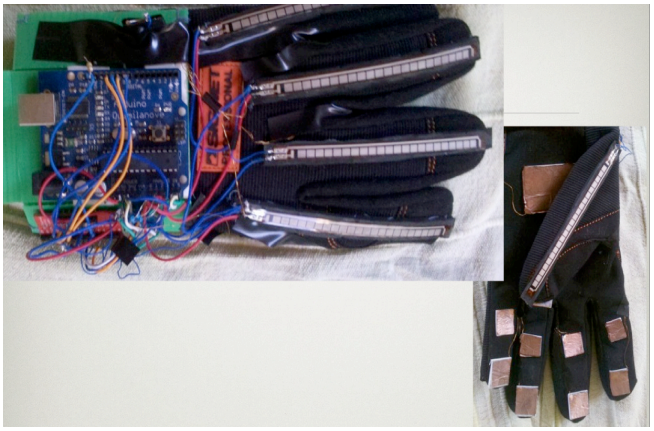

Figure 2. The JhaneGLove

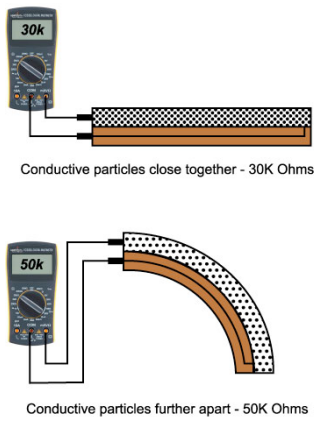

Figure 3. Flex sensors

CALIBRATION AND TRAINING

The glove is a device that continuously sends data from its sensor. In order to pre-process the data to be sent to our main agent (a server side component), we have used an electronic board, the Arduino, to develop the pre-processing software. The board is attached directly to the glove.

Arduino is an open-source electronics prototyping platform based on flexible, easy-to-use hardware and software. It can be used in various ways to create interactive systems. The Arduino board is shown in Figure 2. It can be used with a variety of sensors to control for example lights or motors. The microcontroller on the board can be programmed using a custom programming language. This enabled us to implement pre-processing computations on the glove itself. Arduino projects can be stand-alone or they can communicate with software running on a computer (e.g. Flash, Processing). We have used this ability to develop a computer client that handles communication between the glove and the recognizer. It also contains everything needed to support the microcontroller; we connect it to the computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.

Calibration

In order to use the system, the user must first calibrate the glove. The first stage is to record the min/max values that the user can emit with the glove. This process requires the user to bend and unbend all fingers, as well as move the hand in all directions. The calibration is very short and non tedious for the user. The min/max values are stored on the Arduino board. The pre-processing software embedded in the Arduino board also performs normalization (0-128) on the values that range between (-200,200) for flex sensors and (4000-8000) for the gyroscope. From that point, all data emitted from the glove is sent to our server in a normalized form and can be processed by the gesture recognition agent.

From a user point of view, the software provides a simple user interface. In order to calibrate the glove, the user presses the "Start calibration" button. At that moment, the system stores the timestamp of startup. The glove keeps the calibration signal and starts to send raw data to the server until the "Stop calibration" event generated by the stop button. On generation of the "Stop calibration" event, the system stores the timestamp of shutdown, loading all the raw data from the database which was received from the user during the start-stop time range. Then it computes the minimum and maximum value per sensor. These values are then sent back to the glove.

When the glove receives the last bit of calibration data, it send a "Ready" message to the server and continues to send the calibrated sensor's data to the server by using the new set of minimum/maximum values.

Training

After calibration, the user can train the system to recognize his signs. The training process is relatively intuitive, with a simple user interface where each sign must be entered four times (Figure 4). The user controls the training progress: he can take a break when necessary and can complete the training at a later time.

The problem we are solving a multiple class classification problem. The input dimension corresponds to the sensors input: 20 dimensions with different input ranges for each sensor, multiplied by the number of continuous values recorded for each gesture. The more the user gesture is precise, the better the recognition engine will perform. In addition, the more gestures there are, the more there is a probability of similarity (close distance between gestures in our 20 dimension space). This is why the recognition agent can function with more than 30 signs in practice, but it actually performs more accurately with fewer signs (see experiments below).

Different users use the glove: we implemented an authentication system that guarantees a personal set for each user. After successful authentication, the system loads the user specific training set, and the user can start using the glove with the signs he previously entered.

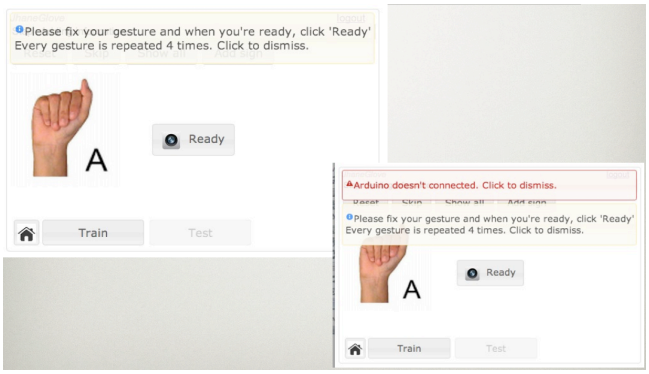

Figure 4. Training

THE SOFTWARE ARCHITECTURE

Our system is built in a client-server architecture where the server side component includes the gesture recognition agent, the database that collects the training and calibration data, and each user's personal language data (the neural network serialization).

The glove with the Arduino is the "wearable" device. On the Arduino board we have the processing software that stores calibration data. The Arduino is connected to a local computer (the client) that transfers the glove data to the server side. The whole setup at this stage is not as mobile as we would hope that the operational system would be. Nevertheless it is sufficient to allow us to experiment and test the feasibility and usefulness of such system. Note that, in this project, we are facing both computing, algorithmic, as usability challenges.

The recognition agent produces text. Then we convert the text into voice using the third-party available text-to-speech software.

THE GESTURE RECOGNITION AGENT

The gesture recognition agent is implemented as a back propagation neural network [5]. The back propagation algorithm is a common learning algorithm. It is used for training multilayer networks by means of error propagation. It aims at minimizing the sum of squared approximation errors by adjusting the network parameters. The neural network uses data that was stored in the database during the training process. This data is user specific so each user benefits from a different customized recognizer. The system is initialized with a predefined dataset including the alphabet, allowing unregistered users to be trained on this set.

The neural network is composed of an input linear layer of size 20, a middle sigmoid layer of size 13, and an output layer of size 1. Data is continuously fed into the neural network and the recognition of a sign is validated after five consecutive recognitions of the same sign. A sign generally last for less than half a second up to one second: slow as well as fast signs can be recognized.

EXPERIMENTS

The recognition accuracy of the device was tested under various conditions: users were asked to enter a full sentence using the alphabet, then they were asked to add a new sign to the dataset and use this sign within a sentence. The results are summarized in Table 1: the percentage represents the amount of correctly recognized symbols. It clearly shows a very good percentage for 15 signs but the recognition accuracy drops and it is too low for a full set of 30 signs. This can be explained both from a software and hardware aspect. The handmade glove requires a lot more sensors to provide more accurate data, and the neural network configuration could be improved.

Natural sign language is certainly much faster without the glove and also much faster than typing. We measured the time required for the user to make the sign and for its recognition. We found that a user sign lasts for about half a second (that includes collecting data and storing it in the database). The recognition is achieved in about half a second (that includes retrieving the data from the database and activating recognition engine). So about a second for a sharp gesture or more for a shaky gesture, is required for each sign. That gives us a maximum rate of 60 signs per minute.

The transition to from one gesture/letter to another works well, but we have so far been unable to identify the repetition of a gesture, for example where a letter appears twice in a word. Recall that we are not using specific signs to signal the end of a sign, although we could. Many systems do use a start/end signal sign to facilitate the parsing. Adding such feature makes the user experience a lot slower (twice as many signs for the same word) and tedious. This is why we did not implement such functionality.

# of signs	accuracy
15	92%
20	88%
30	80%

Table 1: Recognition accuracy

CONCLUSION

We have described early results of a homemade glove that allows defining and recognizing the conventional and also custom sign languages. The core system functions were completed and tested. The system is open for extensions both on the hardware and software sides. As a first experiment, we identified several aspects that could benefits from significant improvements. First we could use more sensors, better quality sensors. Two of five flex sensors have a very small range and the gyroscope sometimes shows inconsistent values. We could also add flex sensors between fingers to enhance the precision of gestures. Second we could rely on a wireless Bluetooth connection between the glove and the computer to free completely the user from the physical computer. On the long term, we would aim for a smartphone application where the system would really become wearable. The smartphone would contain the application front-end (user interface of the system), while the server side would perform the computations of the recognition agent and hold the database.

Our current focus is on improving the gesture recognition agent since we still have not reached a satisfying recognition of the full set of signs. We will investigate different neural network configurations with less input in

the middle layer for instance, while keeping an acceptable training time. After we will have improved the recognition agent, we plan on performing user tests to evaluate the system from a user perspective. For example we are not sure about using separator signs for experienced users. A simple long pause might be enough instead of a *space* sign. Finally we are exploring other directions such as an auto-complete ability to enhance the user experience and shorten the numbers of signs required to express a word or sentence.

REFERENCES

1. Zahoor Zafrulla, Helene Brashear, Thad Starner, Harley Hamilton, and Peter Presti. 2011. American sign language recognition with the kinect. In Proceedings of the 13th international conference on multimodal interfaces(ICMI '11). ACM, New York, NY, USA, 279-286.
2. EnableTalk http://enabletalk.com
3. Simon Lang, Marco Block, and Raúl Rojas. 2012. Sign language recognition using kinect. In Proceedings of the 11th international conference on Artificial Intelligence and Soft Computing - Volume Part I (ICAISC'12)
4. Matt Huenerfauth and Pengfei Lu, Accurate and Accessible Motion-Capture Glove Calibration for Sign Language Data Collection, ACM Transactions on Accessible Computing, 3, 1, Article 2 (2010).
5. P. Sibi, S.Allwyn Jones and P.Siddarth, Analysis of Different Activation Functions Using Back Propagation Neural Networks, *Journal of Theoretical and Applied Information Technology*, (2013). Vol. 47 No.3