Transactions on Machine
Learning and Data Mining
Vol. 10, No. 2 (2017) 41-55

ISSN: 1864-9734,
ISBN: 978-3-942952-54-5

ibai Publishing
www.ibai-publishing.org

American Sign Language Recognition using Hidden Markov Models and Wearable Motion Sensors

Rabeet Fatmi[1], Sherif Rashad[1,2], Ryan Integlia[2], Gabriel Hutchison[1]

[1]College of Innovation & Technology, Florida Polytechnic University
[2]College of Engineering, Florida Polytechnic University
4700 Research Way, Lakeland, Florida, 33805
{rfatmi3593, srashad, rinteglia, ghutchison2600}@flpoly.org

Abstract. In this paper, we propose an efficient and non-invasive solution to translate American Sign Language (ASL) to speech utilizing two wearable armbands called Myo. The compact Myo armbands that are used in this study are much more practical than existing solutions, which include glove-based techniques, camera-based systems, and the use of 3D depth sensors. We applied the Gaussian Mixture Model Hidden Markov Model (GMM-HMM) technique to achieve classification rates of up to 96.15% for ASL words (gestures). The HMM-based approach also sets a solid foundation for future work on the system, which includes continuous ASL recognition as well as signer independence.

Keywords: Sign language recognition, Hidden Markov Model (HMM), American sign language (ASL), Human Computer Interaction (HCI)

1 Introduction

American Sign Language is the leading method of communication between the speech and hearing-impaired population. It involves the use of visual gestures and body movements as a means of expression. One of the main difficulties that users of ASL face is engaging in effective verbal exchanges and collaboration with the hearing, as much of the general hearing population is not well-versed in ASL. As a result, the problem of sign language recognition has been well researched in an effort to bridge this communication gap. Many prior contributions to the field of sign language recognition use image/video based identification techniques [3]; while others have proposed a glove-based technique involving sensory gloves to be worn on either or both arms [4]. 3D depth sensors such as the Microsoft Kinect [9-10] and Leap Motion [8] have also evolved as a means of sign language recognition interfaces.

Compared to existing ASL recognitions systems involving a camera, sensory gloves or 3D depth sensors (see Figure 1), the proposed system is a much more portable in the sense that the user is not restricted to a certain area for signing and it is far less invasive than the 3D depth sensor based technique as the signer is free to move around while signing. Moreover, the armband based system that is being proposed in this paper is an end-to-end solution with no additional requirements other than a PC and two Myo armbands. A single armband is also supported if the gestures involved do not require movement of the secondary arm. A Myo is a wearable motion sensor that detects hand movements as well as electrical activity from the forearm.

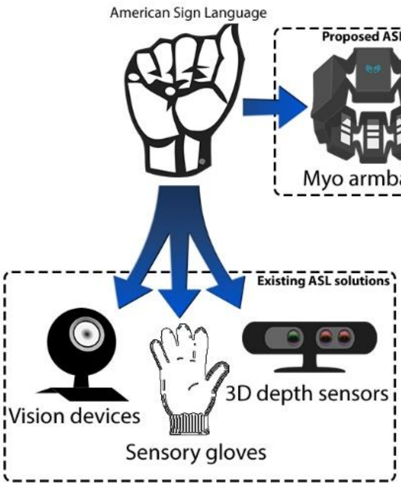

Fig. 1. ASL recognition solutions are divided into four broad categories by the sensors used. Vision-based techniques, sensory gloves and 3D depth sensors. Additionally, EMG/Accelerometer based sensors such as the Myo armband [18] have also been seen to aid in sign language recognition which is the topic of our study. (Myo image courtesy of [5])

We propose the use of the Hidden Markov Model (HMM) classification technique as HMMs have been widely used for the purposes of speech recognition over the past 20 years [13]. Furthermore, the intrinsic properties of HMM to represent doubly stochastic processes with signal segmentation allow them to be used effectively for sign language recognition [1]. Wang et al. [4] developed a Cyberglove based ASL recognizer with multi-dimensional HMMs to achieve an average of 95% correct recognition for 26 ASL alphabets and 36 basic ASL handshapes. Similarly, Starner et al. [12] described a system which used a one color camera to track hands in real time and interpret American Sign Language using HMMs. Experiments were conducted while wearing colored gloves and without gloves, and this yielded a word accuracy of 99%

and 92% respectively. Considering these promising results, we decided to follow an HMM based approach for our system and applied the Gaussian Mixture Model Hidden Markov Model (GMM-HMM) technique to classify 13 different words in ASL signed by three ASL users. We experimented with several different hidden state configurations and achieved classification rates of up to 96.15%.

In this paper, we reviewed related work done in the field of ASL recognition and compared the proposed system to existing solutions in section 2. Section 3 provides details of the entire system and each of its components, including data collection, processing, as well as recognition. In section 4 we provided a brief mathematical as well as an analytical overview of Hidden Markov Models, the machine learning technique of choice for sign language recognition for this paper. Lastly, the details of the experimental results are presented in section 5.

2 Related Work

Research on sign language recognition started percolating in the late-1980s. Tamura and Kawasaki presented an isolated sign image processing system in 1988 that recognized 20 Japanese signs based on matching cheremes [6]. Similarly, in 1992, Charayaphan and Marble published literature on an isolated image processing algorithm for the interpretation of American Sign Language. The system was able to correctly identify all 31 of the test samples by performing 3 different tests [7]. More recently, the use of Leap Motion [8] and Microsoft Kinect [9-10] has been observed to aid in the recognition of sign language.

Still, image recognition is the analysis of still images to convert the contents of the image into usable information. The idea is to detect the signed word or sentence using the sequence of images. One popular technique in still image recognition involves obtaining the Edge Images from the raw image and then using recognized methods to detect the signed word. The use of gradient masks and slope magnitude methods to obtain edge images is one of the most common ways [3]. Edge Histogram Descriptor algorithm has also been used as a means of recognizing the hand gesture and combining it with an artificial neural network for classification [14]. Image/video based recognition systems use features extracted from raw images or videos from users for sign language translation. The underlying drawback to this solution is that the system is still dependent on the presence of cameras which makes the system quite invasive and immobile. Real-time recognition also poses a problem since the image and video data usually require some form of computational heavy processing before it can be made functional [3]. Moreover, vision-based techniques usually require the signing environment to be “system-ready” so that the image or video can be processed as accurately as possible. Therefore, these techniques are affected by factors such as background colors or additional objects or movements in the surroundings. In contrast, the system that is introduced as part of this study is not affected by any of the aforementioned constraints as it detects raw motion data directly from the wearer’s arm. In addition, it is far less invasive than the camera-based techniques as the signer is free to move around while wearing the armbands as long as the host device is within bluetooth range.

Similarly, there has been significant work performed on sign language recognition utilizing Microsoft Kinect which is a motion sensor add-on to the Xbox gaming console but can also be used for non-gamification purposes such as digital signage, virtual shopping, and education. Ahmed et al. [9] and Usachokcharoen et al. [10] have done studies on the feasibility of using the Microsoft Kinect as a sign language sensor. Although both studies showed that the recognition rate was kept fairly high for certain gestures, the user's hand was still required to be in the view of the sensor at all times. Moreover, the use of a color-coded glove was also proposed to improve the performance of the hand recognition algorithm from the recorded video. In comparison, the armband based sign language translator that was developed as part of this study is an end-to-end solution with no additional requirements. A compatible PC that can run the Myo Connect application is the only requirement. It is also a more practical approach since the system can be taught new gestures with ease without the need for recording a video first or performing preliminary processing tasks such as extracting usable features from the image or video.

Chuan et al. [8] have done a study on using the Leap Motion sensor as an ASL recognition system. The leap motion controller is a compact sensor for tracking hand and finger movements in a 3D space of around 8 cubic feet above the device. It is therefore unfeasible to progress towards a motion-heavy recognition system as the leap motion sensor can only detect hand and finger data, such as position and spread of palm and fingers, from a single hand. Compared with the leap motion sensor, the Myo armband is a much more ubiquitous solution as it does not constrain the user to a certain signing space and supports the use of both hands.

In addition, sign language recognition is not limited to American Sign Language. Attempts have been made to translate sign languages of different regions to spoken language. Youssif et al [15] developed an Arabic Sign Language Recognition System using Hidden Markov Models (HMM). A video-based recognition engine was developed which used skin detection, edge detection, as well as hand fingertips tracking combined with a built HMM model to achieve reasonably high classification rates. Similarly, Raheja et al. [16] developed a system targeting the Indian sign recognition area based on dynamic hand gesture recognition techniques a real-time scenario. The captured video was preprocessed by converting it to an HSV color space and then segmenting it to image frames. Hu-Moments and motion trajectory were extracted from the image frames and the classification of the gestures was performed using Support Vector Machine.

Many of the aforementioned systems are signer-dependent, meaning they require prior knowledge about the user. A more efficient system would be one that is signer-independent and allows classification of unseen data without any information about the user. Fang et al. [11] presented a hybrid SOFM/HMM system for Chinese sign language recognition that is signer-independent. The system combined self-organized feature maps (SOFMs) with Hidden Markov Models (HMM) to recognize signer-independent CSL of 4368 samples from 7 signers with 208 isolated signs.

Fig. 2. Myo EMG pod configurations

3 Proposed Technique

We propose an EMG and accelerometer based technique for translating sign language to speech. Sensors based on accelerometers and surface electromyography (EMG) have previously been used for gesture-based HCI [17]. We made use of the Myo armband sensor for this purpose. An myo device is a compact and affordable commercial-grade armband sensor for free hand and finger movements. As shown in Figure 2 the sensor has 8 EMG sensors (named S1 – S8). Moreover, the sensor also has a highly sensitive nine-axis IMU containing three-axis gyroscope, three axis accelerometer, and three-axis magnetometer. Communication to the PC happens via Bluetooth low energy (BLE) connection with the effective range of up to 30 feet around the Bluetooth adapter.

The interactive software, as shown in Figure 4, that interfaces with the Myo devices was written in the Java programming language running on a 64-bit version of Windows 10 with an Intel Core i7-6500U CPU @ 2.50 GHz and 8.00 GB of ram. Myo's official API is written in C++ programming language so a third-party Java-based API was used. The MaryTTS Java API was used for the programming of speech synthesis. Hidden Markov Model ToolKit (HTK) was used for the modeling and training of the HMMs.

Figure 3 shows the overall structure of the current system and highlights the future work that can be performed on the model. The system that is presented in this paper performs data collection, preprocessing and recognition of isolated gesture data while future iterations of the study will focus on continuous gesture recognition and signer independence.

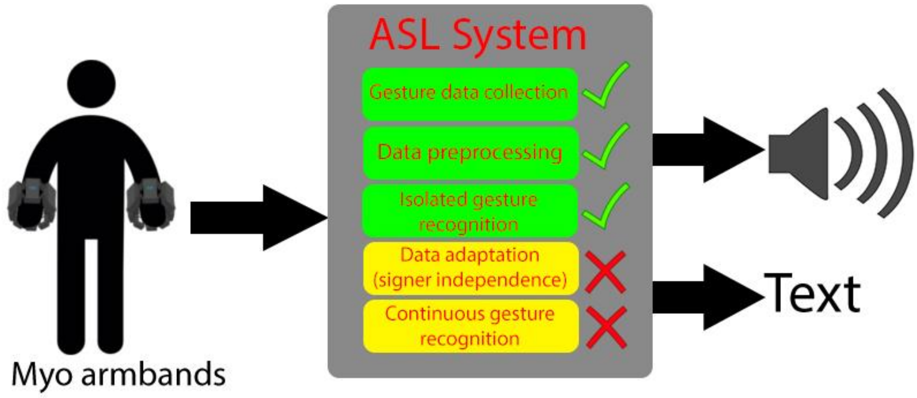

Fig. 3. System schematic of the proposed ASL recognition system

3.1 Data Collection

The features that were used for the training and recognition of the HMMs included orientation, gyroscope, and accelerometer data from both myos as well as the EMG data from the dominant hand myo. Data was formatted as a 26-element time-series array. The IMU data is updated at a speed of 50 Hz while the EMG data comes in at 200 Hz meaning that a 5-second recording will provide 250 new samples of IMU data and 1000 new samples of EMG data. Consolidating this data from both myos as one feature vector, the final feature space is shown in Table 1.

EMG (Electromyography) data. EMG data can be thought as electrical activity of the muscles. EMG data from the myo is reported as a byte array of 8 elements (representing 8 EMG pods (see Figure 2)) meaning that EMG values are between -128 to 127 and unit-less. EMG data is also extremely user-dependent as factors such as arm length, arm circumference, as well as the exact position of the arm where the data originates from, all come into play in realizing the EMG data.

IMU (Inertial measurement unit) data. IMU data is provided by the highly sensitive nine-axis IMU consisting of three-axis gyroscope for gyroscopic data, three-axis accelerometer for accelerometer data, and three-axis magnetometer for orientation data. IMUs work in part by detecting changes in roll, pitch, and yaw. The information obtained from each of these sensors is combined to get the best positional data since one is more sensitive than the other in certain scenarios [17].

Accelerometer. An accelerometer unit measures acceleration relative to a free-fall. The myo device includes a tri-axial accelerometer unit, where three accelerometers are aligned in x,y,z-axis, respectively, and each measure relative acceleration along its dimension.

Gyroscope. Gyroscopes sense angular velocity along one dimension. Meaning, it is possible to get changes in the rotational orientation of the device from a frame of reference or known orientation of the device. The Myo's gyroscope is tri-axial as well, where each gyroscope is measuring the change in rotational orientation along its dimension.

Orientation. The magnetometer determines the orientation of the device with respect to Earth's magnetic field in each of its axes. Simply put, a magnetometer measures direction and strength of the magnetic field along one dimension.

The orientation data from Myo comes in the form of Quaternion data (an array of w, x, y and z values), which can then be converted to Euler angles (roll, pitch and yaw) by using the following equations:

$$\text{roll} = \tan^{-1}((2(-w\cdot x+y\cdot z))/(1-2(x^2+y^2))) \quad (1)$$

$$\text{pitch} = \sin^{-1}(2(-w\cdot y-z\cdot x)) \quad (2)$$

$$\text{yaw} = \tan^{-1}((2(-w\cdot z+x\cdot y))/(1-2(y^2+z^2))) \quad (3)$$

Table 1. Features used for Machine Learning

	Dominant hand Myo		**Non-dominant hand Myo**	
	Name	**Type**	**Name**	**Type**
Features used	Roll (x)	Float [0,360]	Roll (x)	Float [0,360]
	Pitch (y)	Float [0,360]	Pitch (y)	Float [0,360]
	Yaw (z)	Float [0,360]	Yaw (z)	Float [0,360]
	X-Acceleration	Float	X-Acceleration	Float
	Y-Acceleration	Float	Y-Acceleration	Float
	Z-Acceleration	Float	Z-Acceleration	Float
	X-Gyroscope	Float	X-Gyroscope	Float
	Y-Gyroscope	Float	Y-Gyroscope	Float
	Z-Gyroscope	Float	Z-Gyroscope	Float
	EMG (S1-S8)	Integer [-128,127]		

3.2 Data Preprocessing

To prepare the data for training and classifying through machine learning, preprocessing was performed to improve recognition rates and reduce error. As the EMG data is streamed, the absolute value of the most recent thirty samples is saved for each sensor. These values are averaged for each sensor. The data is also normalized by dividing it by 127 (the maximum value). This is done to stabilize the EMG data and reduce any noise. See figure 5 for an example; The green lines represent the absolute value of the recorded data, the orange lines demonstrate the average over the past 30 samples, and the red lines demonstrate the overall average for the entire word. As expected, the averaged data reduces noise while still containing enough useful information for classification.

The vertical orientation (yaw) that the Myo armbands provide are not consistent across power cycles. Additionally, the yaw tends to change slowly over time, a widespread problem with gyroscopes known as “yaw drift”. To correct this, the raw orientation data is transformed through a calibration procedure, which has to occur at least once per usage session. The user is prompted to hold their arms straight forward. The orientation data at that moment is saved and used as the origin. All future recordings are calculated as an offset from the origin, to provide consistent measurements across sessions.

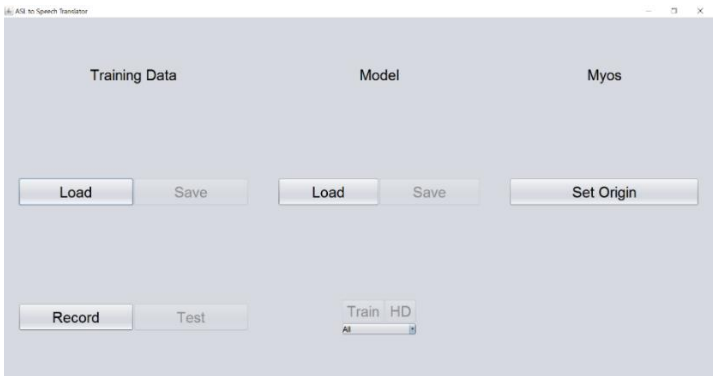

Fig. 4. ASL to speech translation software

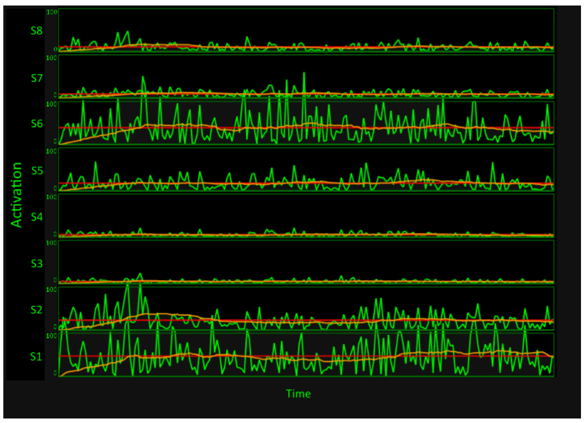

Fig. 5. EMG data reporting large amounts of muscle flex and movement

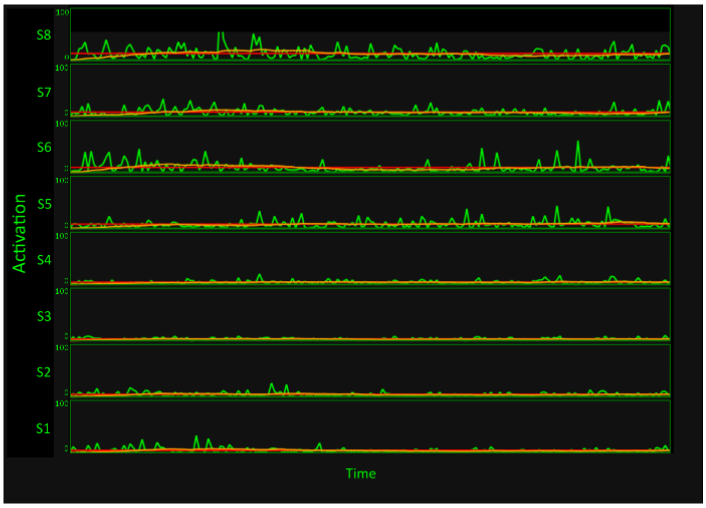

Fig. 6. EMG data reporting minimal movement of the hand

4 Hidden Markov Models

HMMs (Hidden Markov Models) have been used prominently in the field of sign language recognition with success. An HMM is a probabilistic framework for modeling time-series data and fall in the Dynamic Bayesian Network class. Since gesture data can typically be represented as a time-series vector of different extracted features, they work naturally with HMM.

The HMMs were trained with the simple left to right topology as shown in Figure 7 where each state is allowed to transit to itself and the next state (strictly linear). Moreover, since our feature values are real numbers we modeled the HMMs with continuous parameter densities with 16 Gaussian mixtures per state.

An HMM is formally defined as 5-tuple representing a given process with a set of states and transition probabilities between the states:

$$\Omega = \{N, M, A, \beta, \pi\} \tag{4}$$

where N indicates the number of distinct possible states in the model not directly observable except through a sequence of observations represented by M. Every state has a set of distinct observation symbols M, also called emissions, which are observable. β represents the discrete/continuous probabilities for these emissions. The state transition probability distribution is defined by A which indicates the chance that a certain state change might occur. These probabilities as well as the starting probabilities, π, are discrete.

Given the observation sequence $O = o_1\ o_2\ o_3\ \dots\ o_T$ where o_i represents the feature vector observed at time i, and a separate HMM, λ, for each ASL gesture, then the sign language recognition problem can simply be solved by computing:

$$\arg\max_i \{P(\lambda_i|O)\} \tag{5}$$

where i corresponds to the i'th ASL gesture. This probability is not computable directly but using Bayes' Rule [2]:

$$P(\lambda_i|O) = (P(O|\lambda_i)\cdot P(\lambda_i))/P(O) \tag{6}$$

Thus, for a given set of prior probabilities P(λ), the most probable signed gesture depends only on the likelihood $P(O|\lambda_i)$. Given the dimensionality of the observation sequence O, the direct estimation of the likelihood $P(O|\lambda_i)$ is not practical. However, if a parametric model such as a Markov model is assumed, then estimation from data is possible since this problem is replaced by a much simpler problem of estimating the Markov model parameters [2]. A Markov model is a finite state machine which changes its state only once per time unit *t* and each time *t* that a state *j* is entered, a feature vector o_t is generated from the probability density $b_j(o_t)$.

Furthermore, a "Hidden" Markov Model implies that the state changes themselves are not directly observable and the transition from state *i* to state *j* is also probabilistic and is governed by the discrete probability a_{ij}. An example of this is shown in Figure 7 where the 4-state model moves through the state sequence X = 1; 2; 2; 3; 3; 4. The entry and exit HMM states are non-emitting in the HTK Toolkit, which is used to build the models in this work. The joint probability O generated by the model λ moving through the state sequence X is calculated as the product of transition probabilities and output probabilities. For the state sequence X in Figure 7, O is calculated as follows:

$$P(O|\lambda) = a_{12}b_2(o_1)\cdot a_{22}b_2(o_1)\cdot a_{23}b_3(o_2)\cdot a_{33}b_3(o_2)\cdot a_{34}\dots \tag{7}$$

where the parameters $\{a_{ij}\}$ and $\{b_j(o_t)\}$ of the model are determined by an estimation procedure called the Baum-Welch re-estimation procedure [1].

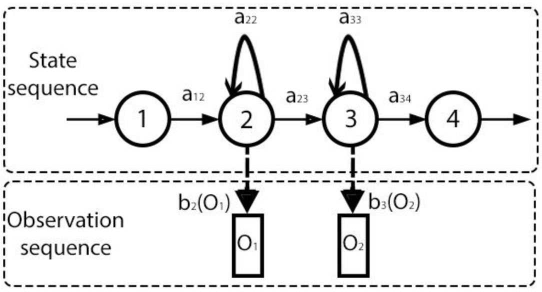

Fig. 7. A four-state left-to-right HMM topology with two emitting states

4.1 Training Phase

Given a number of ASL gestures in the dictionary, the training is performed by building an HMM for each gesture by using feature vectors for the sequence of observations for that gesture. We used a number of observations to estimate the optimum parameters for each gesture, giving model λ_i, for the i^{th} gesture. After some trials, we found that 200 observations with 26 features each were enough training data for our purposes.

4.2 Recognition Phase

Recognition in the context of our problem means to calculate the likelihood of each model generating the unknown data using the Viterbi Algorithm [1] and then choosing the most likely model that identifies the unknown gesture as shown in Figure 8.

Our final implementation of the HMM classifier comprised of the following steps:

1. Preprocess and transform the data so that it is in the HTK supported format,
2. Train N models, one for each gesture, by computing the model parameters using the Baum-Welch re-estimation algorithm
3. Use the generated models to determine the classification accuracies for each user.

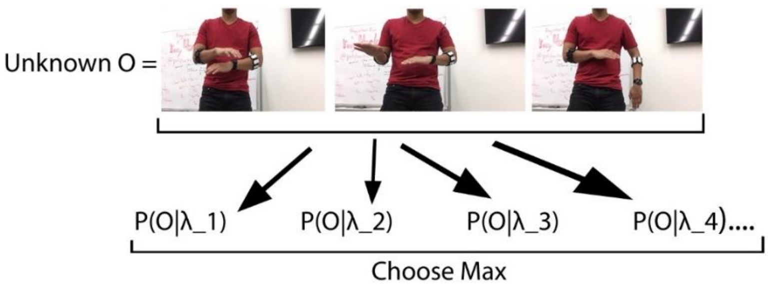

Fig. 8. Recognition of an unknown gesture is performed by choosing the maximum likelihood of the model that generates the gesture

5 Experimental Results

Table 2. Overall recognition results for all users for all different hidden states configuration

	User 1	User 2	User 3
4 hidden states	84.62%	96.15%	76.92%
6 hidden states	80.77%	92.31%	73.08%
8 hidden states	80.77%	76.92%	69.23%
Average / user	82.05%	88.46%	73.08%
	Overall Accuracy by User: 81.20%		

5.1 Experiment Setup

Data ~~s~~ were collected for 13 ASL gestures performed by 3 different ASL users. Gestures were chosen in a manner so that there was a good mix of words that involved motion of one hand as well as both hands. The following words were chosen for this experiment: 'Time', 'Me/I', 'Water', 'Think', 'Friend', 'Meet', 'Teacher', 'Your', 'You', 'Who', 'How', 'What', and 'Name'. Each word was repeated 10 times with 200 samples for each set. In total, 130 recordings were performed by each user. The recording of each word was carried out over a four second period from start to finish. To progress towards a robust ASL recognition system we wanted to determine the accuracy of the system with a minimal number of training and test samples, therefore, we deemed 8 samples for each word to be a reasonable requirement of the system.

5.2 Results

Evaluation results were obtained from the HMM recognizer for each gesture performed by 3 different ASL users. Eight samples were used for training; two were used for testing the system. Since there is no set rule of thumb on the number of hidden states to incorporate for each HMM, we performed multiple different benchmark tests and the results are displayed in table 2. The accuracy of the system is defined by the equation:

$$\text{Accuracy} = (S/N) \cdot 100\% \tag{8}$$

where S is the number of correctly recognized gestures and N is the total number of test gestures.

5.3 Discussion

We noticed that a 4-state topology yielded the best results as shown in Figure 9. This is possibly due to the fact that the current system is only a word-based recognizer, therefore selecting a low number of states per word is a reasonable decision. Moreover, according to [1] the number of states per word should roughly correspond to the average number of observations in the "spoken" version of the word. Consequently, choosing a 4-state topology in our model means that we are assuming that the signed word can be decomposed into 3 to 4 observations, which is a fair assumption if we consider the actual signing to happen in 3 different stages as sign-start → sign-mid → sign-end.

Table 3 shows the detailed results of training and recognition performed for user 2. As noted earlier, we will see that as we decrease the number of states per model our recognition accuracy increases. We also noticed that most of the errors occurred for gestures that involved motion of only one hand as shown in table 4. Lack of data from the secondary myo, in the case of 'Water', 'Me/I', 'Who', and 'You', can cause the system to misclassify the gesture. Currently, our system does not penalize for "static motion" data. In future versions of this system, a "weighting" option can be incorporated which assigns more weight to feature values obtained from the dominant hand myo in case of gestures that do not involve the heavy movement of the non-dominant hand.

Table 3. Detailed results of gestures with different hidden states per HMM for user 2

	Recognition results with different number of hidden states		
Gesture	4 hidden states	6 hidden states	8 hidden states
Friend	2	2	2
How	2	2	2
Meet	2	2	1
Me/I	2	2	1
Name	2	2	2
Teacher	2	2	2
Think	2	2	1
Time	2	1	2
Water	1	1	1
What	2	2	2
Who	2	2	1
You	2	2	1
Your	2	2	2

NOTE: Each cell indicates the number of correctly recognized test instances out of a total of 2 test instances.

Table 4. Detailed recognition results for all users

	Recognition results for all users (4 hidden states)			
Gesture	User 1	User 2	User 3	Overall
Friend	2	2	2	100%
How	2	2	2	100%
Meet	2	2	2	100%
Me/I	1	1	2	67%
Name	2	2	1	83%
Teacher	2	2	2	100%
Think	2	2	2	100%
Time	2	2	2	100%
Water	1	0	0	17%
What	2	2	2	100%
Who	2	1	0	50%
You	2	2	0	67%
Your	2	2	2	100%

NOTE: Each cell indicates the number of correctly recognized test instances out of a total of 2 test instances.

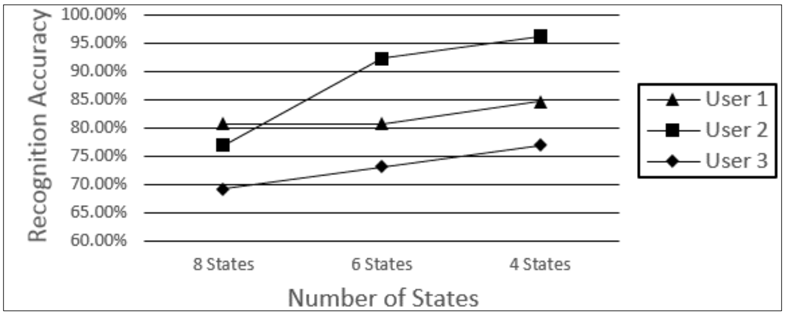

Fig. 9. Recognition accuracy vs. number of hidden states per model

6 Conclusion and Future Work

In this paper, we presented an ASL recognition system utilizing Myo armbands using multi-dimensional Hidden Markov Models. The system is an emerging application of the Hidden Markov Models machine learning technique as HMMs have not been applied to gesture data from the Myos before, to the best of our knowledge. The system currently can perform training and testing of basic hand gestures which involve motion. The evaluation results show that the HMM-based approach is promising for future research directions which include continuous ASL recognition of full sentences as well as an adaptation of data for a new signer (signer-independence).

References

1. Rabiner L.R., Juang B.H.: An introduction to hidden Markov models. IEEE ASP Magazine 3(1), 4-16 (1986).
2. Minker W., Waibel A., Mariani J.: Stochastically-based semantic analysis. Springer Science & Business Media, Vol. 514 (2012).
3. Thepade S.D., Kulkarni G., Narkhede A., Kelvekar P., Tathe S.: Sign language recognition using color means of gradient slope magnitude edge images. In: Intelligent Systems and Signal Processing (ISSP), pp. 216 – 220. Anand (Gujarat), India (2013).
4. Wang H., Leu M.C., Oz C.: American Sign Language Recognition Using Multi-dimensional Hidden Markov Models. Journal of Information Science and Engineering 22, 1109-1123 (2006).
5. Myo homapage, https://www.myo.com
6. Tamura S., Kawasaki S.: Recognition of sign language motion images, Pattern Recognition 21(4), 343-353 (1988).
7. Charayaphan C., Marble A.E.: Image processing system for interpreting motion in American Sign Language. Journal of Biomedical Engineering 14(5), 419-425 (1992).
8. Chuan C.H., Regina E., Guardino C.: American Sign Language Recognition Using Leap Motion Sensor. In: Proceedings of the 13th International Conference on Machine Learning and Applications (ICMLA), pp. 541–544, Detroit, MI, USA (2014).
9. Ahmed M., Idrees M., Abideen Z., Mumtaz R., Khalique S.: Deaf Talk Using 3D Animated Sign Language. In: SAI Computing Conference (SAI), pp. 330-335, London, UK (2016).

10. Usachokcharoen P., Washizawa Y., Pasupa K.: Sign Language Recognition with Microsoft Kinect's Depth and Colour. In: International Conference on Signal and Image Processing Applications (ICSIPA), pp. 186-190, Kuala Lumpur, Malaysia (2015).
11. Fang G., Gao W., Ma J.: Signer-independent sign language recognition based on SOFM/HMM. In: IEEE ICCV Workshop on Recognition, Analysis and Tracking of Faces and Gestures in Real-Time Systems, pp. 90-95, Vancouver, Canada (2001).
12. Starner T., Pentland A.: Real-time American sign language recognition from video using hidden markov models. In: Motion-Based Recognition, pp. 227-243. Springer, Netherlands (1997).
13. Wu J., Gao W., Song Y., Liu W., Pang B.: A simple sign language recognition system based on data glove. In: Proceedings of the 4th International Conference on Signal Processing, pp. 12-16, Beijing, China (1998).
14. Schneider B., Rashad S.: Hand Gesture Recognition for Automatic Translation of American Sign Language Using Machine Learning Techniques. In: Proceeding of the 31st International Conference on Computers and Their Applications, CATA, p. 207-214, Las Vegas, Nevada, USA (2016).
15. Youssif A., Aboutabl A.E., Ali H.: Arabic sign language (ArSL) recognition system using HMM. Journal of Advanced Computer Science and Applications (IJACSA), Vol. 4(1), 41-52 (2012).
16. Raheja J.L., Mishra A., Chaudhary A.: Indian sign language recognition using SVM. Journal of Pattern Recognition and Image Analysis 26(2), 434–441 (2016).
17. Zhang X., Chen X., Li Y., Lantz V., Wang K., Yang J.: A Framework for Hand Gesture Recognition Based on Accelerometer and EMG Sensors. IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans, 41(6), 1064-1076 (2011).
18. Paudyal P., Banarjee A., Gupta S.K.: SCEPTRE: A Pervasive, Non-Invasive, and Programmable Gesture Recognition Technology. In: Proceedings of the 21st International Conference on Intelligent User Interfaces ACM (2016).