An Approach to Glove-Based Gesture Recognition

Farid Parvini, Dennis McLeod, Cyrus Shahabi, Bahareh Navai, Baharak Zali, Shahram Ghandeharizadeh
Computer Science Department
University of Southern California
Los Angeles, California 90089-0781
[fparvini,mcleod,cshahabi,navai,bzali,shahram]@usc.edu

Abstract

Nowadays, computer interaction is mostly done using dedicated devices. But gestures are an easy mean of expression between humans that could be used to communicate with computers in a more natural manner. Most of the current research on hand gesture recognition for Human-Computer Interaction rely on either the Neural Networks or Hidden Markov Models (HMMs). In this paper, we compare different approaches for gesture recognition and highlight the major advantages of each. We show that gestures recognition based on the Bio-mechanical characteristic of the hand provides an intuitive approach which provides more accuracy and less complexity.

1. Introduction

Gestures are destined to play an increasingly important role in human-computer interaction in the future. Humans use gestures in their everyday communication with other humans, not only to reinforce the meanings that they convey through speech, but also to convey meaning that would be difficult or impossible to convey through speech alone. Hence, to make human-computer interaction truly natural, computers must be able to recognize gestures in addition to speech. Furthermore, gesture recognition is a requirement for the virtual reality environments, where the user must be able to manipulate the environment with his/her hands.

Closely related to the field of gesture recognition is the field of sign language recognition. Because sign languages are the primary mode of communication for many deaf people, and because they are full-fledged languages in their own rights, they offer a much more structured and constrained research environment than general gestures. While a functional sign language recognition system could facilitate the interaction between deaf and hearing people, it could also provide a good starting point for studying the more general problem of gesture recognition.

As an example, American Sign Language (ASL) is a complex visual-spatial language that is used by the deaf community in the United States and English-speaking parts of Canada. ASL is a natural language and it is linguistically complete. Some people have described ASL and other sign languages as 'gestural' languages.

ASL includes two types of gestures: static and dynamic. Static signs are the signs that according to ASL rules, no hand movement is required to generate them. All ASL alphabets excluding 'J' and 'Z' are static signs. In contrary to the static signs, dynamic signs are the ones which their generation require movement of fingers, hand or both.

One of the most challenging issues in gesture recognition is the ability to recognize a particular gesture made by different people. This problem, often called *user-dependency*, rises from the fact that people have different ergonomic sizes, so they produce different data for the same gestural experiment. As a solution device manufacturers suggest *Calibration*, which is the process makes the generated data as identical as possible. Calibration is the comparison of a measured value of unverified accuracy to a verified accuracy measure to detect any variation from the required performance specification.

Another relevant challenge in gesture recognition is device-dependency. That is the generated data by two different devices for the same experiment are completely different. In this paper, we compare three different approaches that are used for static sign language recognition. In our current implementations we only considered the static signs and dynamic sign detection is not supported by any of these three implemented systems.

While the accuracy is one of the aspects that we consider when comparing these approaches, we also consider other characteristics that can affect the accuracy and may favor one of these approaches for a specific application (e.g. user-independency and extensibility).

The remainder of this paper is organized as follows. Section 2 discusses the related work. We present each approach along with its strengths and limitations in Sections 3, 4 and 5, respectively. The results of our comparative experiments are reported in Section 6 . Finally, Section 7 concludes this paper and discusses our future research plans.

2. Related work

To date, sign recognition has been studied extensively by different communities. We are aware of two major approaches: Machine-Vision based approaches which analyze the video and image data of a hand in motion and Haptic based approaches which analyze the haptic data received from a sensory device (e.g., a sensory glove).

Due to lack of space, we refer the interested readers to [10] for a good survey on vision based sign recognition methods. With the haptic approaches, the movement of the hand is captured by a haptic device and the received raw data is analyzed. In some studies, a characteristic descriptor of the shape of the hand or motion which represents the changes of the shape is extracted and analyzed. Holden and Owens [4] proposed a new hand shape representation technique that characterizes the finger-only topology of the hand by adapting an existing technique from speech signal processing. Takahashi and Kishino [9] could recognize 34 out of 46 Japanese kana alphabet gestures with a data-glove based system using joint angle and hand orientations coding techniques. Newby [6] used a ”sum of squares” template matching approach to recognize ASL signs. Hong et. al [5] proposed an approach for 2D gesture recognition that models each gesture as a Finite State Machine (FSM) in spatial-temporal space. Su and Furuta [8] propose a ”logical hand device” that is in fact a semantic representation of hand posture.

More recent studies in gesture recognition have focused on Hidden Markov Model (HMM) or Support Vector Machines (SVM). While these approaches have produced highly accurate systems capable of recognizing gestures [11], we are more concerned about important characteristic of approaches such as extensibility and user-dependency rather than only the accuracy.

3. NEURAL NETWORK APPROACH

Neural networks are composed of elements, called neurons, which are inspired by biological nervous systems. The elements operate in parallel and form a network, whose function is determined largely by the connections between them. A neuron receives its inputs from a number of other neurons or from an external stimulus. Usually neural networks are trained so that a particular input leads to a specific target output. A comparison between the output and the target helps adjust the network. Typically, in order to train the network,many input/target pairs are required.

Neural networks have received much attention for their success in pattern recognition and gesture recognition is no exception to this. The main reason for their popularity is that once the network has configured, it forms appropriate internal representer and decider systems based on training examples. Since the representation is distributed across the network as a series of interdependent weights instead of a conventional local data structure, the decider has certain advantageous properties:

- recognition in the presence of noise or incomplete data
- pattern generalization

Generalization plays a crucial role in the system’s performance, because most gestures will not be reproduced even by the same user with perfect accuracy, and when a range of users are allowed to use the system, the variation becomes even greater. Other useful properties of this approach include performing calibration automatically and the ability to classify ’raw’ sensor data.

3.1. Strengths and limitations of the Neural Network Approach

Neural networks are very popular in these types of classification applications for their simplicity. Some of the advantages of neural networks are listed here:

1. No calibration is required
2. A trained neural network would recognize an unknown input sign very fast
3. Our experiments proved that with sufficient data, this approach produces very good results
4. With their current popularity, ready-to-use neural network packages are readily available, so there is almost no time spent on implementing the system

Neural networks have some drawbacks, too:

1. A large amount of labeled examples are required to train the network for accurate recognition
2. If a new gesture is added or one is removed, more accuracy can achieved by re-training the whole system.
3. Neural networks can over learn, if given too many examples, and discard originally learned patterns. It may also happen that one bad example may send the patterns learned into the wrong direction. This or other factors, such as orthogonality of the training vectors, may prevent the network from converging at all
4. Neural networks tend to consume large amount of processing power, especially in the training phase

5. The major problem in utilizing neural networks is to understand what the network had actually learned, given the ad-hoc manner in which it would have been configured
6. There is no formal basis for constructing neural networks, the topology, unit activation functions, learning strategy and learning rate. All these should be determined by trial and error

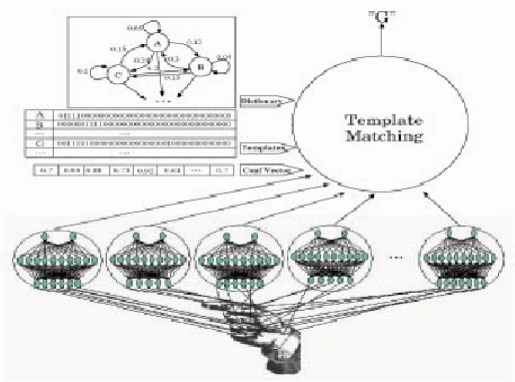

Figure 1. Multi-layer framework for detecting static ASL signs

4. MULTI- LAYER APPROACH

The Multi-Layer Approach, originally proposed in [3] combines template matching and neural networks, so it benefits from both approaches. Figure 4 shows the key modules in the implementation of this approach. By layering, the system achieves device independency and extensibility, two important properties that most other systems lack. Three layers of this approach are as follows:

1. The lowest layer, *raw data layer*, receives streams of noisy sensory data from the input device and translates them into a semantic description of the hand. The first layer is by nature device dependent but can be application independent.
2. In the next layer, *postural predicate layer* (or neural network layer), all possible postures of hand required for finger spelling of American Sign Language (ASL), are described by 38 Boolean postural predicates. This layer includes 38 3-layer feed-forward neural networks (with 10-hidden nodes), one neural network for each predicate. The number of inputs to each net is between 4 and 10 sensor values. Since neural networks are specifically trained for one predicate, and they have a limited range of input values, their accuracy is expected to be relatively higher than a single general neural network. The postural predicate layer, like the first layer, is device dependent and application independent. The results of this layer are device independent and can be used in different applications.
3. *Gestural template layer* is the final layer, which is completely application dependent and device independent. This layer contains a set of gestural templates that describe hand gestures for each sign in ASL. The system finds the nearest match to the gesture by computing a weighted distance between the gesture and all the gestural templates.

4.1. Strengths and limitations of the Multi- Layer Approach

Since this approach also uses neural networks, it has some of the drawbacks listed in 3.1, which are not repeated in this section.

Advantages of this approach can be summarized as the following:

1. No calibration is required
2. For a trained system, recognizing an unknown input sign is fast
3. Our experiments show that when we have limited number of data sets for training the neural networks, this approach behaves more accurately than a single layer neural network
4. Since the neural networks have limited number of inputs, training of the system (38 neural networks) takes much less time than training a single layer general-purpose neural network
5. Extensibility is another important feature; to add a new sign, it is not necessary to redefine and retrain the networks as it is in the single layer neural network. The postural templates are supposedly sufficient for defining all the simple postures required for ASL. Hence, we only need to add a gestural template to the system for the new sign. Even if a new postural predicate needs to be defined, we only need to map required sensors to a new predicate in the first layer, define a new postural predicate and train only the new corresponding neural network, and define the new gestures. Nothing in the rest of the system needs to be changed.
6. The application layer, or the gestural template layer, is device independent. The system can work with different types of haptic or visual devices only by some modifications in the first two layers.

7. The first two layers are also application independent, so that when the postures are detected in the second layer, any application can use this clean semantic description of hand for their own purpose.

As it is mentioned before, this system has some of the drawbacks of single layer neural networks as well as the followings:

1. When the number of training sets increased to 18 sets, both single layer neural network and GRUBC approaches behaved more accurately than the Multi-Layer Framework. Although we only represent our results for one training set vs. 18 training sets for each approach, we tested the system with different number of sets and it appears that somewhere between 1 and 18 this approach achieves its maximum accuracy.
2. Using multiple neural networks and combining them with template matching makes implementation of the system somehow complicated.

5. GRUBC: GESTURE RECOGNITION BY UTILIZING BIO-MECHANICAL CHARACTERISTICS

Gesture recognition by utilizing bio-mechanical characteristics, originally proposed in [7] is inspired by the observation that all forms of hand signs include finger-joint movements from a starting posture to a final posture. We utilize the concept of 'range of motion' from the Bio-Mechanical literature at each joint to abstract this movement.

Range of motion (ROM) is a quantity which defines the joint movement by measuring the angle from the starting position of an axis to its position at the end of its full range of the movement. For example, if the position of a joint axis changes from 20° to 50° with respect to a fixed axis, the range of motion for this joint is 30°.

We compute the range of motion per joint by using the sensor values acquired by the sensory device. The main intuition behind this approach is that the range of motion of each section of the hand participating in a sign, relative to the nonparticipating sections, is a user-independent characteristic of that sign. This characteristic provides a unique signature for each sign across different users. Given a sensory device with n sensors, each ASL static sign can be represented with a set of n values. Let us call this set $S_i = (s_1, s_2, \ldots, s_n)$ where i is an ASL sign and S is the set of sensor values. One issue in gesture recognition is that different users making the same sign (gesture) would generate different S_i's , i.e., S_i is not unique for different users.

Our objective is to transform S_i to NR_i where NR_i is unique across different users. Suppose S_0 and S_i represent the sets of the initial and final postures for a specific sign respectively. We calculate the range of motion tuple R_i as follows: $R_i = S_i - S_0$. Subsequently, we find the maximum and minimum values within R_i and represent them with M(R) and m(R) respectively. We then normalize each value in R_i and represent the result of this normalized R which consists of values between 0 and 1 with NR. Finally, we discretize the values of NR with a given discretization parameter k (> 1). For example, if k=2, we replace each value of NR with 0 if its value is less than 0.5.

Since NR represents the characteristic of movement of the sensors making a particular sign, it provides an abstraction for that sign. We call this abstraction the signature of the sign and observe that while the signature is unique for each sign, it is identical among different users making the same sign. That is, if different users wear the sensory device and make a specific ASL sign, while the raw data generated by the sensors are completely different, the calculated NR are almost identical across all of them. This also implies that by abstracting the sign with its signature, we eliminate the effect of inevitable noise produced by the sensors during the data collection process. The uniqueness of this signature provides us with the very important property of user independency.

In order to recognize an unknown static sign made by a user, we require comparing its signature with the signatures of some known samples. Consequently, the first step is collecting the data for each static sign once and calculating its corresponding NR. We call this process 'registration' and save all the registered signs in our registration database. While we have this database, we compare the signature of each unknown gesture with all registered signatures and the unknown gesture is labelled with the signature with the least distance (according to a distance metric, e.g. Euclidian distance).

Figure 2. CyberGlove and the location of its sensors

Registration which is the core of this approach is completely different from 'training' in the following aspects:

- In contrast to 'training' that requires several sets of data, we require one set of samples (one sample for each sign) to completely register all signs.
- While having more signers register each sign will potentially increase the accuracy, there is no direct relationship between the number of registered signs and accuracy, as is in the training.

5.1. Strengths and limitations of GRUBC

Based on our observations and the primary results of our experiments, this approach benefits from the following advantages:

1. User-independency, which implies that the registration can be done by any signer without having impacts on the overall accuracy.
2. Device independency which allows the registration to be done with another device or even without any sensory device. The reason is that the information we require to register a sign is the relative movements of the sensors, if this information can be provided by another device or even without using sensory device, we can still register the sign.
3. No calibration (a must in most traditional approaches) is required prior to data gathering from different users.
4. This approach is extensible, i.e. new signs can be recognized just by registering them and there is no requirement to train the system all over again.
5. While the focus of this approach is on the problem of classification, it has a broader applicability. With classification, each unknown data sample is given the label of its best match among known samples in a database. Hence, if there is no label for a group of samples, the traditional classification approaches fail to recognize these input samples. For example, in a virtual reality environment where human behavior is captured by sensory devices, every behavior (e.g., frustration or sadness) may not be given a class label, since they have no clear definition. This approach can address this issue by finding the similar behavior across different users without requiring to have them labeled.

On the other hand, the shortcomings of this approach can be listed as follows:

1. As we was mentioned before, the core of this approach is based on the signature matching of an unknown sign and the registered signs. If two signs have similar signatures (e.g. ASL alphabets R and U), this approach fails to differentiate them.
2. To match two signatures, we used two approaches:
   (a) We could recognize two signatures identical if their distances are less than a threshold (ε). In this approach, the threshold should be defined and we assume the value is application dependent.
   (b) Two signatures are identical, if they are the nearest-neighbors. In this approach, two signatures may be recognized identical while they are very far and completely different due to the fact that no other match could be found.

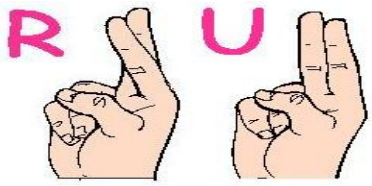

Figure 3. ASL signs R and U

Figure 4. ASL alphabets 'H' and 'K'

6. PERFORMANCE RESULTS

In this section, we present results of the experiments conducted to compare performance of three gesture recognition approaches in recognizing the ASL static signs.

6.1. Experimental Setup

For our experiments, we used CyberGlove [1] as a virtual reality user interface to acquire data. CyberGlove is a glove that provides up to 22 joint-angle measurements.

It uses proprietary resistive bend-sensing technology to transform hand and finger motions into real-time digital joint-angle data. This glove model has three flexion sensors per finger, four abduction sensors, a palm-arch sensor, and sensors to measure flexion and abduction. A picture of this glove, indicating the location of each sensor is displayed in Figure 2.

We initiated our experiments by collecting data from 19 different subjects performing static signs ('A' to 'Y', excluding 'J') from a starting posture while wearing the glove.

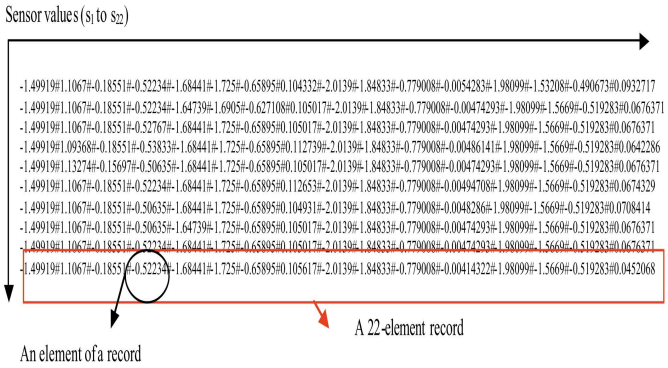

Figure 5. An illustration of part of the dataset (ASL A)

We assured the same starting posture for all the experiments. For each sign, we collected 140 tuples from the starting posture to the final posture, each including 22 sensor values.

While each sign was made by a transitional movement from a starting posture to a final posture, we conducted the experiments in a way that the last 40 tuples of each experiment belong to the static sign and do not include any transitional movement. Figure 5 shows a sample of data-set for one ASL sign, A. During our preliminary experiments, we discovered due to the limited number of sensors in the CyberGlove, it is not capable of capturing all the finger positions associated with every ASL alphabet. Hence, it cannot be used to differentiate some alphabets regardless of the approach. Figure 4 shows H and K are just different in some wrist rotation that is not captured by the glove. Figure 3 shows there are not enough sensors in the glove to capture the relationship between the index and middle fingers which is essential to differentiate R from U. Thus we left the signs H & R out of our experiments. For some data files, complete sensor values are not captured for some lines. We also excluded all the data lines which did not have all 22 sensor values.

For each approach, we conducted two different sets of experiments: Leave-One-Out and Keep-One-In. With the Leave-One-Out approach, all the data sets are used in training except one, which is used for testing. The same test is conducted 19 times, each time with a different set of data used for testing, until all the data sets are tested. The reported results for this set of experiments are the average of all 19 sets.With Keep-One-In approach, which is the opposite of the Leave-One-Out, only one data set is used for training the system, and the remaining 18 sets are used for testing. There again, we repeated this set of experiments 19 times, each time a different data set used for training and the remaining 18 sets for testing. The reported result of this set of experiments is again the average of all the 19 experiments. For the experiments both neural network frameworks used similar setups and experimental approaches.

As explained above, and illustrated in Figure 5, the data files contain records of sensor values taken in time. The last 40 tuples of each data file represent a static sign. Each record (tuple) consists of 22 values, one per sensor reading. We had several options in selecting which records to employ for training of a network. A simple approach is to use each record as a representative of an ASL sign. Another is to select a record randomly. Our approach is as follows. For each subject, we used average, minimum, maximum, and median of each sensor reading from the available 40 records. In addition, we also considered the use of the middle record as the representative entry for a subject. For our test data, we used the 40 records available from a subject as follows. The neural net is invoked with each record and the sign that is detected the maximum number of times is selected as the recognized ASL sign.

6.1.1. Single-Layer Neural Network Experimental Setup In these sets of experiments, we used a feed-forward, back propagation neural network from the MATLAB [2] neural network toolbox. This network has one hidden layer with 20 neurons. The input layer has 22 neurons (each for one sensor) and there are 22 neurons (one for each sign) in the output layer. The set of experiments conducted are exactly as explained above.

6.1.2. Multi-Layer Experimental Setup In this approach, in addition to the training set, there is also a tuning set. Although we followed the same guidelines to run the experiments as single-layer neural network approach, because of this difference we had to divide the training sets into two parts.

For the first set of experiments, Leave-One-Out, when 18 sets are used in training and one in testing, we tested two different setups; in the first one, we used 9 data sets for training, 9 other data sets for tuning the networks and the 19th set for testing the system (9-9-1 setup). In the second setup for Leave-One-Out, we used 18 sets for training

Alphabet	Neural Network	Neural Network	Multi-Layer	Multi-Layer	Multi-Layer	GRUBC	GRUBC
A	89.47%	29.83%	78.95%	84.21%	58.78%	94.44%	78.00%
B	100%	42%	100%	100%	92%	94%	78%
C	57.89%	23.98%	73.68%	89.47%	39.33%	83.33%	56.00%
D	84.21%	40.06%	89.47%	89.47%	69.11%	94.44%	72.00%
E	94.74%	36.55%	100.00%	94.74%	62.00%	88.89%	78.00%
F	100%	45%	100%	100%	81%	94%	83%
G	68.42%	22.52%	42.11%	57.90%	42.56%	55.56%	44.00%
I	89.47%	35.97%	84.21%	73.68%	73.22%	94.44%	89.00%
K	78.95%	23.98%	94.74%	100.00%	56.94%	72.22%	67.00%
L	73.68%	39.77%	42.11%	52.63%	41.28%	88.89%	72.00%
M	89.47%	29.24%	63.16%	52.63%	47.72%	83.33%	72.00%
N	73.68%	33.92%	42.11%	26.31%	58.11%	61.11%	39.00%
O	63.16%	26.90%	57.89%	47.37%	38.44%	55.56%	33.00%
P	94.74%	22.52%	84.21%	89.47%	19.11%	83.33%	78.00%
Q	78.95%	32.75%	68.42%	47.37%	39.11%	50.00%	17.00%
S	84.21%	36.84%	89.47%	68.42%	57.61%	94.44%	72.00%
T	57.89%	21.93%	21.05%	5.26%	27.72%	83.33%	67.00%
U	57.89%	21.05%	15.79%	5.26%	34.78%	88.89%	72.00%
V	84.21%	30.99%	15.79%	10.53%	59.44%	61.11%	39.00%
W	100%	26%	100%	100%	93%	100%	100%
X	84.21%	35.67%	0.00%	0.00%	41.32%	94.44%	83.00%
Y	94.74%	52.05%	94.74%	94.74%	72.11%	94.44%	89.00%
AVERAGE	81.82%	32.24%	66.27%	63.16%	54.75%	82.32%	67.18%
	Training set : 18	Training set : 1	Training set : 9 Tunining set : 9	Training set : 18 Tunining set : 18	Training set : 1 Tunining set : 1	Registration: 18	Registration: 1
	Testing set : 1	Testing set : 18	Testing set : 1	Testing set : 1	Testing set : 18	Testing set : 1	Testing set : 18
	Repeating: 19	Repeating: 19	Repeating: 19	Repeating: 19	Repeating: 19	Repeating: 19	Repeating: 19

Figure 6. Comprehensive result for two sets of experiments. The overall results for static ASL alphabet recognition. In the first column static signs are listed. The second column represents the average results achieved by single layer neural network approach for the Leave-One-Out experiments. The third column shows the results for Keep-One-In experiments of the same approach. Forth, fifth columns show the results of the multi-layer framework for Leave-One-Out experiments respectively for 9-9-1 and 18-18-1 setups. Column six lists the results for Keep-One-In experiments of the same framework. The last two columns represent the overall results for GRUBC approach, respectively for Keep-One-In and Leave-One-Out experiments.

the networks, and used the same 18 sets for tuning the networks and the last set for testing (18-18-1 setup).

In the second sets of experiments, which were Keep-One-In, one set is used for training and the same set is used for tuning the system, and then system is tested on all the remaining 18 data sets (1-1-18 setup). As in 7.2.1, all these experiments are repeated 19 times, each time for a different subject.

6.1.3. GRUBC Experimental Setup We repeated two sets of aforementioned experiments with GRUBC as we did for the neural network and multi-layer, except we did not have the training phase for this approach. The other dominant difference between the experiments of neural network and GRUBC is that in the former case, the data used to train the system was a preprocessed data, including statistical calculated data, e.g. min, max and average, while with GRUBC, only the raw data was used for registration. We conducted the Leave-One-Out experiment (i.e., registered with 18 sets and tested with the remaining set) and repeated it 19 times. We then conducted Keep-One-In experiment (i.e., registered with one set and tested with the remaining 18 sets) and repeated it 19 times.

6.2. Experimental Results and Observations

In this section the results of the experiments are listed, followed by the explanation regarding the differences between the results of the two sets of experiments and our observations. In each set of experiments, the results are represented after averaging the results of each sign across all users. The average results of static ASL alphabet signs in each set of experiments and each approach are shown in Table 6.

In the Leave-One-Out set of experiments, the single layer neural network approach had the overall accuracy of 81.82% while in the multi-layer approach, the accuracy for 9-9-1 setup was 66.27% and 63.16% for 18-18-1 setup respectively. We achieved the overall accuracy 82.32% for GRUBC approach, which was the maximum accuracy among all the approaches and setups.

For the Keep-One-In set of experiments, the overall accuracy for the Single layer neural network approach was 32.24%. The multi-layer approach showed 54.75% for the 1-1-18 setup and in GRUBC approach, the accuracy was 67.18%.

In the single layer neural network method, the results of the second experiments, Keep-One-In experiments, have degraded due to the fact that the neural network's training set is composed of input patterns together with the required response pattern as we mentioned earlier. If we don't provide the network with proper and adequate training set, the learning process will not complete. As a result, when we train the network with one subject and then test with the remaining 18, the accuracy will drop.

For the multi-layer approach, since the 38 neural networks in the second layer are specifically trained for a simple posture and the range of input data is very limited, they can behave more accurately than the general neural network when the training set is very small.

For the GRUBC approach, higher accuracy in Leave-One-Out is explained as follows: When registering each sign by 18 different signers, the error rises from having different sizes will be compensated a lot, i.e. the chance of having the signs registered with a similar hand is much higher. The second factor is that since the variety of registered signs is wider, the possibility of finding a similar sign in the registered signs is higher. If we consider each alphabet a point in 'n' dimensional space (in this case 'n' is equal to 22), we call each experiment a correct recognition if the unknown sign is the nearest neighbor of its matched alphabet, meaning that the unknown gesture (representing the point in 22-dimensional space) is a correct recognition if its nearest neighbor is its identical alphabet (e.g. nearest neighbor of 'a' is 'a'). In the second set of experiments, we have 18 similar points around the data representing the unknown gesture, so the possibility of its nearest neighbor being the matched sign is much higher.

7. CONCLUSION & FUTURE WORKS

In this paper, we compared different major approaches for recognizing static hand gestures and high-lighted the significant advantages of each approach. We also compared the results and showed while 'Gesture Recognition based on Biomechanical Characteristic' provides higher accuracy, it addresses detecting the similar hand gestures without having them labelled, a problem that most traditional classification methods fail to address

We plan to extend this work in two directions. First, we intend to extend our technique to recognize complex dynamic signs, e.g. continuous dynamic signs and compare the results with traditional approaches for dynamic gesture recognition. Second, we would like to show that in general, utilizing any other characteristic which defines the system on a higher level or abstraction rather than data layer (e.g. Bio-mechanical characteristic) provides both higher accuracy on result and less dependency on the data gathering process.

References

[1] Immersion corporation, www.immersion.com.

[2] Mathworks corporation, www.mathworks.com.

[3] J. Eisenstein, S. Ghandeharizadeh, L. Golubchik, C. Shahabi, D. Yan, and R. Zimmermann. Device independence and extensibility in gesture recognition. *IEEE Virtual Reality Conference (VR),LA, CA*, 2003.

[4] E. J. Holden and R. Owens. Representing the finger-only topology for hand shape recognition. *Machine Graphics and Vision International Journal*, 12(2), 2003.

[5] P. Hong, T. S. Huang, and M. Turk. Constructing finite state machines for fast gesture recognition. *International Conference on Pattern Recognition (ICPR'00)*, 3, 2000.

[6] G. B. Newby. Gesture recognition using statistical similarity.

[7] F. Parvini and C. Shahabi. Utilizing bio-mechanical characteristics for user-independent gesture recognition. *International Workshop on Biomedical Data Engineering (BMDE2005), Tokyo, Japan*, April 2005.

[8] S. A. Su and R. K. Furuta. Vrml-based representations of asl fingerspelling. *Proceedings of the third international ACM conference on Assistive technologies*.

[9] T. Takahashi and F. Kishino. Hand gesture coding based on experiments using a hand gesture interface device. pages 67–73, 2003.

[10] Y. Wu and T. S. Huang. Vision based gesture recognition a review. *International Gesture Workshop, GW 99, France*, March 1999.

[11] J. Yang and Y. Xu. Hidden markov model for gesture recognition. Technical report, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA.