Utilizing Bio-Mechanical Characteristics For User-Independent Gesture Recognition[*]

Farid Parvini, Cyrus Shahabi
Computer Science Department
University of Southern California
Los Angeles, California 90089-0781
[fparvini,shahabi]@usc.edu

Abstract

We propose a novel approach for recognizing hand gestures by analyzing the data streams generated by the sensors attached to the human hands. We utilize the concept of 'range of motion' in the movement and exploit this characteristic to analyze the acquired data. We show that since the relative 'range of motion' of each section of the hand involved in any gesture is a unique characteristic of that gesture, it provides a unique signature for that gesture across different users. Based on this observation, we propose our approach for hand gesture recognition which addresses two major challenges: user-dependency and device-dependency. Furthermore, we show that our approach neither requires calibration nor involves training. We apply our approach for recognizing ASL signs and show that we can recognize static ASL signs with no training. Our preliminary experiments demonstrate more than 75% accuracy in sign recognition for the ASL static signs.

1. Introduction

Human motion recognition has many important applications such as improving human-computer interaction in the virtual reality application domain. This research area concerns the tracking, detection and recognition of the movement of people and more generally, understanding the human behavior. Due to the diversity of human sizes and ergonomic measures, one of the most challenging issues in this context is recognizing the movement of different people robustly regardless of this diversity. A more specific problem arises during gesture recognition while various users making the same gesture.

In order to recognize the movement or specifically a gesture, the user is traced and monitored through various sensory devices such as tracking devices on her hands or haptic devices. Data collected from the sensors of these devices are considered as *Multi-Stream Human Sensor Data*, or MSHSD as we call , which is the continuous immersive data stream which is generated by the sensors attached to human body. This data type has the following special characteristics, it is:

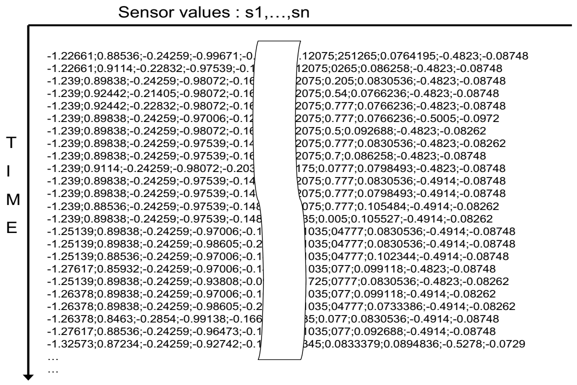
- user-dependent
- device-dependent
- noisy

User-dependency implies that the data generated by different users for the same experiment are not identical. *Calibration* is the process that device manufacturers suggest to make the generated data as identical as possible. It is the comparison of a measured value of unverified accuracy to a verified accuracy measure to detect any variation from the required performance specification. Machine learning techniques (e.g., neural networks, decision trees or case-based reasoning) are also addressing user-dependency. They are the algorithms that generate a model based on the previously seen data (i.e., training data) in order to classify the new data. 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 propose an approach for recognizing gestures based on the bio-mechanical characteristics of the movement of the hand. In our approach, we abstract out a unique signature for each sign based on the 'range of motion' of the sensors (joints) involved in that gesture.

Our research is distinct and novel in the following three aspects. First, to the best of our knowledge,

[*] This research has been funded in part by NSF grants IIS-0238560 (PECASE) and IIS-0307908, and unrestricted cash gifts from Microsoft. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.



```
-1.22661;0.88536;-0.24259;-0.99671;-0.        .12075;251265;0.0764195;-0.4823;-0.08748
-1.22661;0.9114;-0.22832;-0.97539;-0.1        12075;0265;0.086258;-0.4823;-0.08748
-1.239;0.89838;-0.24259;-0.98072;-0.16        2075;0.205;0.0830536;-0.4823;-0.08748
-1.239;0.92442;-0.21405;-0.98072;-0.16        2075;0.54;0.0766236;-0.4823;-0.08748
-1.239;0.92442;-0.22832;-0.98072;-0.16        2075;0.777;0.0766236;-0.4823;-0.08748
-1.239;0.89838;-0.24259;-0.97006;-0.12        2075;0.777;0.0766236;-0.5005;-0.0972
-1.239;0.89838;-0.24259;-0.98072;-0.16        2075;0.5;0.092688;-0.4823;-0.08262
-1.239;0.89838;-0.24259;-0.97539;-0.14        2075;0.777;0.0830536;-0.4823;-0.08262
-1.239;0.89838;-0.24259;-0.97539;-0.16        2075;0.7;0.086258;-0.4823;-0.08748
-1.239;0.9114;-0.24259;-0.98072;-0.203        175;0.0777;0.0798493;-0.4823;-0.08748
-1.239;0.89838;-0.24259;-0.97539;-0.14        2075;0.777;0.0830536;-0.4914;-0.08748
-1.239;0.89838;-0.24259;-0.97539;-0.14        2075;0.777;0.0798493;-0.4914;-0.08748
-1.239;0.88536;-0.24259;-0.97539;-0.148        075;0.777;0.105484;-0.4914;-0.08262
-1.239;0.89838;-0.24259;-0.97539;-0.148        35;0.005;0.105527;-0.4914;-0.08262
-1.25139;0.89838;-0.24259;-0.97006;-0.1        1035;04777;0.0830536;-0.4914;-0.08748
-1.25139;0.89838;-0.24259;-0.98605;-0.2        1035;04777;0.0830536;-0.4914;-0.08748
-1.25139;0.88536;-0.24259;-0.97006;-0.1        1035;04777;0.102344;-0.4914;-0.08748
-1.27617;0.85932;-0.24259;-0.97006;-0.1        035;077;0.099118;-0.4823;-0.08748
-1.25139;0.89838;-0.24259;-0.93808;-0.0        725;0777;0.0830536;-0.4823;-0.08262
-1.26378;0.89838;-0.24259;-0.97006;-0.1        035;077;0.099118;-0.4914;-0.08262
-1.26378;0.89838;-0.24259;-0.98605;-0.2        1035;04777;0.0733386;-0.4914;-0.08262
-1.26378;0.8463;-0.2854;-0.99138;-0.166        35;0.077;0.0830536;-0.4914;-0.08748
-1.27617;0.88536;-0.24259;-0.96473;-0.1        1035;077;0.092688;-0.4914;-0.08748
-1.32573;0.87234;-0.24259;-0.92742;-0.1        345;0.0833379;0.0894836;-0.5278;-0.0729
…
…
```

Figure 1. An illustration of a sample data set

our approach in utilizing bio-mechanical characteristics is unique among all the studies who have intended to analyze the collected raw data for gesture recognition. Second, our approach addresses the major challenges involved in analyzing MSHSD: user-dependency, device-dependency and noisy data. Finally, our approach does not require any sort of training or calibration, a must in most machine learning based approaches for gesture recognition.

Finally, while the focus of this paper is on the problem of classification, our approach has 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. Our approach can address this issue by finding the similar behavior across different users without requiring to have them labelled.

The remainder of this paper is organized as follows. Section 2 explains the motivating application. Section 3 formalizes the problem of analyzing data for gesture recognition. Section 4 discusses the related work. We present our approach in Section 5. The results of our experiments are reported in Section 6. Finally, Section 7 concludes this paper and discusses our future research plans.

2. Motivation

To motivate our research of utilizing bio-mechanical characteristics for gesture recognition, we focus on recognizing American Sign Language (ASL) signs as an example of a well-defined set of hand motions. Unlike general gestures, sign languages are highly structured, which makes the recognition problem easier, since their structures can be used to form abstraction and exploit context. Thus, sign language recognition provides a good starting point for studying a more general problem of gesture recognition. In addition, a functional sign language recognition system could facilitate the tedious process of transcribing conversations for sign language research tremendously, as well as facilitating the interaction between deaf and hearing people.

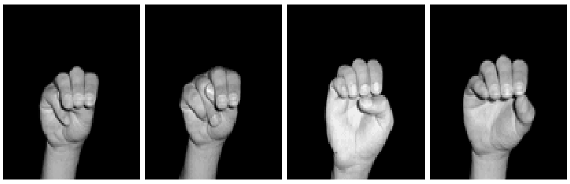


Figure 2. ASL alphabets ‘M’,‘N’, ‘E’ and ‘O’

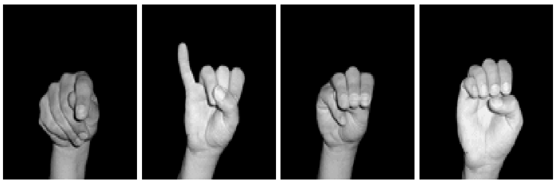


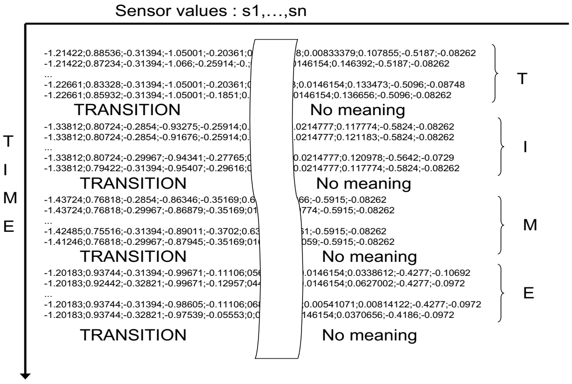
Figure 3. The signing of the word ‘time’ in ASL which consists of the individual alphabets ‘T,‘I’,‘M’ and ‘E’

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. It is a linguistically complete and natural language. Some people have described ASL and other sign languages as ‘gestural’ languages. ASL also has the advantage of including two types of gestures, static and dynamic, hence it is a perfect application for investigating our approach addressing different challenges involve in recognizing both of these types. The main challenges of our approach are analyzing the data and recognizing the gesture or gestures a user makes in real-time. We show that while our approach is user and device independent, it requires neither training nor calibration. This distinguishes our work from other studies and we address this in more details in Section 4.

3. Formal Definition

In this section, we formally describe the problem and define the notations we use through out the paper.

As we mentioned, the process of gesture recognition starts with collecting data from the sensors attached to the hand of a user. At each sensor clock, the sensory device driver captures one sample by acquiring data from all of the n sensors of the device. Each sample is stored in a tuple with n fields where each field is associated with one sensor. We represent a sample at time t by $S_t = (s_1, s_2, \ldots, s_n)$, where each s_i is a real number indicating the value that is acquired from sensor i at time t. As time evolves, a *data set* of samples is acquired and generated. An example of such a data set of samples is shown in Figure 1.



```
-1.21422;0.88536;-0.31394;-1.05001;-0.20361;0          8;0.00833379;0.107855;-0.5187;-0.08262
-1.21422;0.87234;-0.31394;-1.066;-0.25914;-0.;          0146154;0.146392;-0.5187;-0.08262
...
-1.22661;0.83328;-0.31394;-1.05001;-0.20361;0          3;0.0146154;0.133473;-0.5096;-0.08748
-1.22661;0.85932;-0.31394;-1.05001;-0.1851;0.          0146154;0.136656;-0.5096;-0.08262
-1.33812;0.80724;-0.2854;-0.93275;-0.25914;0.          .0214777;0.117774;-0.5824;-0.08262
-1.33812;0.80724;-0.2854;-0.91676;-0.25914;0.          .0214777;0.121183;-0.5824;-0.08262
...
-1.33812;0.80724;-0.29967;-0.94341;-0.27765;0          0.0214777;0.120978;-0.5642;-0.0729
-1.33812;0.79422;-0.31394;-0.95407;-0.29616;0          0.0214777;0.117774;-0.5824;-0.08262
-1.43724;0.76818;-0.2854;-0.86346;-0.35169;0.6          66;-0.5915;-0.08262
-1.43724;0.76818;-0.29967;-0.86879;-0.35169;01          774;-0.5915;-0.08262
...
-1.42485;0.75516;-0.31394;-0.89011;-0.3702;0.63          61;-0.5915;-0.08262
-1.41246;0.76818;-0.29967;-0.87945;-0.35169;01          059;-0.5915;-0.08262
-1.20183;0.93744;-0.31394;-0.99671;-0.11106;056          .0146154;0.0338612;-0.4277;-0.10692
-1.20183;0.92442;-0.32821;-0.99671;-0.12957;044          .0146154;0.0627002;-0.4277;-0.0972
...
-1.20183;0.93744;-0.31394;-0.98605;-0.11106;068          0.00541071;0.00814122;-0.4277;-0.0972
-1.20183;0.93744;-0.32821;-0.97539;-0.05553;0;0          146154;0.0370656;-0.4186;-0.0972
```

Figure 4. Data samples representing the finger-spelling of the word 'TIME' in ASL

In general, we represent the data set, which is a collection of samples, with $\mathcal{C} = \{S_{t_0}, \ldots, S_t\}$, where t_0 denotes the starting time, t the ending time and $\triangle t = t - t_0$ the period of sampling, respectively. Each S_i is the sample we collected at time i. We also show the collection of our data with an interchangeable notation as $\mathcal{C}_{[t_0,t]}$. Since each sample can be considered as a point in an n-dimensional space, we define two samples equal if their distance according to a metric criteria is less than a threshold. In other words: $S_i \equiv S_j \Leftrightarrow Distance(S_i, S_j) \leq \varepsilon$. The most straightforward approach for measuring the similarity between two samples is using a *Minkowski* measure such as the *Euclidean* distance. Given two samples $S_i = (s_1, \ldots, s_n)$ and $S_j = (s'_1, \ldots, s'_n)$, the *Euclidean* distance between S_i and S_j is defined as:

$$Distance(S_i, S_j) = (\sum_{i=1}^{n} |s_i - s'_i|^2)^{1/2} \tag{1}$$

We compare the result of applying different distance metrics in Section 6.

Considering this notation for each sample, we accordingly classify the ASL signs into the following two categories:

1. Static Signs: These 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 spite of the fact that one sample should be enough to represent each alphabet, due to the nature of data acquisition process, we require to represent each static sign with several samples. Hence we represent each sign with $\mathcal{S}_{[t_0,t]} = \{S_{t_0}, \ldots, S_t\} | \forall j \in [t_0, t], S_j \equiv \mathcal{A}_i$, where $\mathcal{A}_i$ represents an ASL alphabet in which $i \in \{A, \ldots, Z\} - \{J, Z\}$. Ideally, for a static sign, all the samples in the time period of $\triangle t = t - t_0$ should be identical. However, since the data acquired from the sensors are usually noisy and some minor movements are inevitable, the data set has some deviation. In other words, ignoring some unlikely exceptions, the consequent samples in $\triangle t$ would not be identical, but mostly their distances from each other are bounded by a threshold and then can be considered equal. Hence, we represent a static sign more accurately by $\mathcal{S}_{[t_0,t]} = \{S_{t_0}, \ldots, S_t\} | \forall j \in [t_0, t] \Rightarrow (Distance(S_j, \mathcal{A}_i) \leq \epsilon \Leftrightarrow S_j \equiv \mathcal{A}_i)$.

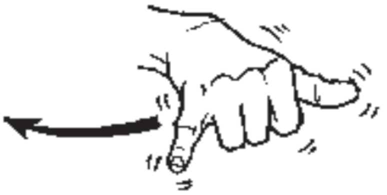


Figure 5. The sign for 'yellow' is made by forming the letter 'y' with the right hand in a circular movement

2. Dynamic Signs: In contrary to static signs, the generation of these signs require movement of fingers, hand or both. We divide this group of signs into two subcategories:

   - Type I: These signs consist of a series of different static signs with some transitional movements between each pair. The movements involve in these signs do not convey any meaning by themselves and are just the transitions between different static signs. The ASL finger-spelling words fall in this category.

     We represent a dynamic sign type I with $\mathcal{D}_{[t_0,t]}$, where the time duration of $\triangle t = t - t_0$ should be considered as a superset of some smaller durations each of which represents a static sign. That is : $[t_0, t] = [t_0, t_1] \bigcup [t_1, t_1 + \triangle t_1] \bigcup [t_1 + \triangle t_1, t_2] \bigcup [t_2, t_2 + \triangle t_2] \bigcup \ldots + [t_n, t]$ where $\mathcal{C}_{[t_1, t_1 + \triangle t_1]}$ represents the first static sign and $\mathcal{C}_{[t_n, t_n + \triangle t_n]}$ represents the nth static sign. With this notation, $[t_1 + \triangle t_1, t_2]$ denotes the time period for transitional samples which represents transitional movement between the first and second static sign and does not convey any meaning by itself. An example of this type of dynamic signs is shown in Figure 3 and its collected data is shown in Figure 4.

- Type II: The rest of ASL dynamic signs fall in this category. They can either represent an alphabet, i.e., 'J' or 'Z' or convey a word, e.g., 'yellow'. An example of this type of signs is shown in Figure 5. We represent this type of signs with $(D_{[t_0,t]}, M)$, where M represents the movement information in the sign. This movement is either the result of the movement of the hand, e.g., a static sign in movement or is the result of both fingers and hand moving simultaneously, e.g., another dynamic sign in movement.

In this paper, we address the problem of recognizing the static signs as well as the dynamic signs of type I. The recognition of the dynamic signs of type II involves two steps:

1. Recognizing the static part (if any), and
2. Recognizing the movement of the fingers and/or the hand

While we address the former in this paper, the latter is the focus of our future research. Consequently, the elaboration of M is beyond the scope of this paper.

The recognition of a static sign requires finding the best match (i.e., the minimum distance) between a sample in $\mathcal{C}[t_0, t]$ (or $\mathcal{D}[t_0, t]$ for a dynamic sign type I) and a known sample $\mathcal{A}_i$. In Section 5, we will show that this problem cannot be addressed by a simple search and comparison. Before proceeding to describe our approach for sign recognition, we survey the studies related to our work in the following section.

4. Related Work

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 [12] for a good survey on vision based sign recognition methods. Within 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 [5] 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. Takashi and Kishino [11] could recognize 34 out of 46 Japaneese kana alphabet gestures with a data-glove based system using joint angle and hand orientations coding techniques. Newby [8] used a 'sum of squares' template matching approach to recognize ASL signs. Hong et. al [6] proposed an approach for 2D gesture recognition that models each gesture as a Finite State Machine (FSM) in spatial-temporal space.

More recent studies in gesture recognition have focused on *hidden Hidden Markov* Model (HMM) or *Support Vector Machines* (SVM), which have produced highly accurate systems capable of handling dynamic gestures [13]. While classification can be accomplished by these model-based learning approaches, they fail to match the signs which have no labels or detect the similar unknown signs. Many researchers proposed using neural networks to address the sign recognition problem. Murakami and Taguchi [7] use recurrent neural networks to detect signs from Japanese Sign Language. Boehm et. al [2] proposed using neural network for dynamic gesture recognition.

The dominant distinction between all these approaches and ours is that they rely on a training phase and that requires a lot of training data. In addition, the results of these techniques are directly affected by the data set chosen for the training phase and consequently they are user-dependent. The idea of device independency has been addressed in a number of studies. Su and Furuta [10] propose a 'logical hand device' that is in fact a semantic representation of hand posture. This was done with the express purpose of achieving device independency, but to our knowledge it was never implemented.

5. ROMAS: Range Of Motion Abstraction for Sign recognition

Our solution to recognize a sign is based on the observation that all forms of hand signs include finger-joint movements from a starting posture to a final posture. To abstract this movement, we utilize the concept of 'range of motion' from the Bio-Mechanical literature [9] at each joint. 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 our approach is that the range of motion of each section of the hand participating in a sign, relative to the non-participating sections, is a user-independent characteristic of that sign. This characteristic provides a unique signature for each sign across different users. We now discuss our approach in more details.

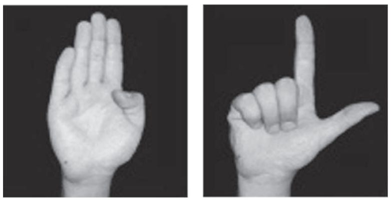


Figure 6. Starting Posture and final Posture

Suppose that a user U is making the static sign 'L' by wearing a sensory device. The user is required to start making the sign from a starting posture toward a final posture. An example of one possible starting posture and the posture representing sign 'L' are shown in Figure 6. Note that our technique does not rely on a specific starting posture, in fact, *any* starting posture will work for us, as long as it is consistent through all experiments. The samples associated with these two postures can be represented as S_{t_0} and S_t, respectively. For the user U and sign 'L', the values of the samples may be similar to $S_{t_0} = (-1.57353, \ldots, -0.04374)$ and $S_t = (-1.59831, \ldots, -0.05346)$. The sensory device and what each sensor value represents are explained in more details in Section 6. If the user repeats making the same sign, due to noisiness of data and some inevitable movements, the raw data would be different as compared to the first experiment. In addition, this raw data is completely user dependent, i.e., the tuples generated by users U and U' making the same sign are completely different.

Due to these circumstances, having an exact match (i.e., an identical sample) for an unknown sample among existing samples is almost impossible. Consequently, recognizing a sign by attempting to search through existing samples in order to find an exact match is not possible either. The best that can be achieved by searching is finding a sign with distance less than a threshold. As it was mentioned before, this approach also fails due to the diversity of raw data.

Hence, our objective is to transform the diverse raw data sample to an abstracted unique sample. This transformation compensates the dissimilarity of the collected data and provides a unique signature across different experiments for the same sign (sign 'L' in this case).

Towards this end, we introduce *ROMAS*, the algorithm that provides this transformation by utilizing the concept of 'range of motion'. The pseudo-code for this algorithm is shown in Figure 9.

Suppose $S_{t_0} = (s_1, \ldots, s_n)$ and $S_t = (s'_1, \ldots, s'_n)$ represent the samples of the start and final postures for making a sign, respectively. We calculate the range of motion tuple $R_{\triangle t}$ as follows :

$R_{\triangle t} = S_t - S_{t_0} = (r_1, \ldots, r_n) | \forall i \in (1, \ldots, n), r_i = s'_i - s_i$ and $\triangle t = t - t_0$

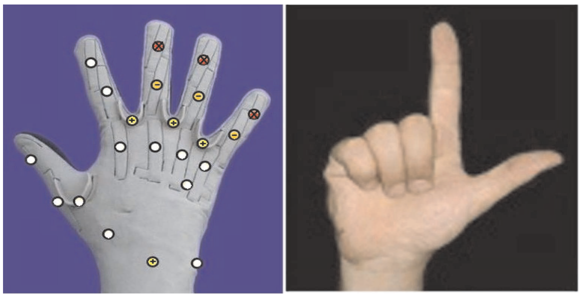


Figure 7. ASL sign 'L' and its representation with k=4

For example, for sign 'L', $R_{\triangle t}$ is calculated as $(-1.59831 - (-1.57353), \ldots, -0.05346 - (-0.04374))$ or $(-0.02478, \ldots, -0.00972)$

$R_{\triangle t}$ is a tuple consisting of n positive or negative real numbers depending on the direction of the movement. In order to normalize the values of the movement, we first construct $\overline{R} = (|r_1|, \ldots, |r_n|)$. The rationale behind using absolute values is that smaller values (i.e., larger negative numbers) in R do not necessarily mean less movement. To capture the direction of the movement and differentiate between movements in opposite directions, for each R, we calculate D which holds directional information as follows:

$D_{\triangle t} = (d_1, \ldots, d_n)$ where

$$d_i = \begin{cases} 1 & \text{if } r_i \geq 0 \\ -1 & \text{otherwise} \end{cases}$$

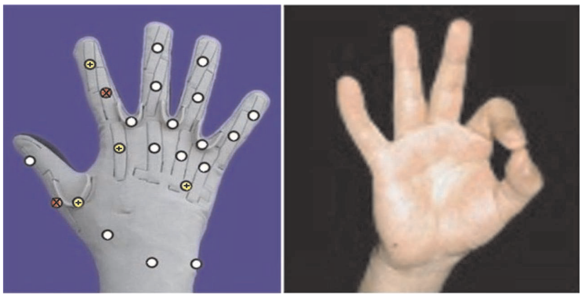


Figure 8. ASL sign 'F' and its representation with k=4

Subsequently, we find the maximum and minimum values within $\overline{R}$ and represent them with $M(R)$ and $m(R)$, respectively. We then normalize each value in $\overline{R}$ by subtracting $m(R)$ and dividing the result by $(M(R) - m(R))$. We 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, and 1 otherwise. The resulting NR for sign 'L' with $k = 4$ is : NR = (0.25; 0.25; 0.25; 0.25; 0.25; 0.25; 0.25; 0.25; 0.75; 1.0; 0.5; 0.25; 0.75; 1.0; 0.5; 0.25; 0.75; 1.0; 0.5; 0.25; 0.5; 0.25). While this tuple contains valuable information about relative movement of each sensor with respect to others for sign 'L', it does not provide any information regarding the direction of the movements. Multiplying each item of NR by its correspondent item in D results a new tuple $NRD = (\bar{n_1}, \ldots, \bar{n_n}) | \forall i \in \{1, \ldots, n\}, \bar{n_i} = n_i * d_i$ which provides both the direction and the relative value of the movements.

The calculated NRD for sign 'L' is shown as follows: NRD =(-0.25; -0.25; 0.25; 0.25; -0.25; 0.25; 0.25; -0.25; -0.75; -1.0; -0.5; 0.25; -0.75; -1.0; -0.5; 0.25; -0.75; -1.0; -0.5; 0.25; 0.5; 0.25).

Since NRD represents the characteristic of the 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 NRDs 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.

For $k = 4$, we visualize the resulting NRD with the following coding for each sensor:

- White color (no mark) if $\bar{n_i} \in [0, 0.25]$
- Yellow color (+ mark) if $\bar{n_i} \in [0.25, 0.5]$
- Orange color (- mark) if $\bar{n_i} \in [0.5, 0.75]$
- Red color (X mark) if $\bar{n_i} \in [0.75, 1.0]$

Figures 7 and 8 show the ASL representation of sign 'L' and 'F', respectively, with our sensor value coding for k = 4.

In the following section, we discuss our method for recognizing ASL signs based on ROMAS.

5.1. Recognizing ASL signs by ROMAS

In this section, we first present our approach for recognizing the ASL static signs and then extend our approach to show how to recognize a dynamic signs type I. Finally we address the problem of finding similar movements without labelling them. In order to recognize an unknown static sign made by a user, we require to compare 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 NRD. We call this process 'registration' and save all the registered signs in our registration database. The 'registration' is completely different from 'training' in the following aspects:

```
Recognize Static Sign( $S_i, k, (\mathcal{A}_1, \ldots, \mathcal{A}_n)$ ) {
  $S_{t_0}$ := Sample of Starting Time
  while( $S_i \leq S_t$ ) {
    $R_i = S_i - S_{t_0}$;
    $\overline{R_i} = |R_i|$;
    $Calculate D_i$;
    $M(R) = Maximum(\overline{R_i})$;
    $m(R) = Minimum(\overline{R_i})$;
    $NR_i = \frac{\overline{R_i} - m(R)}{(M(R) - m(R))}$
    $NR_i = Discretisize(NR_i, k)$;
    $NRD_i = NR_i \times D_i$

    for ($j = 1$ to $n$) {
      if Distance(NRD, NRD($\mathcal{A}_j$)) $\leq \epsilon$
        return $\mathcal{A}_j$
    }
  }
  Return null
```

Figure 9. ROMAS Algorithm for recognizing a static sign

- 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.
- Registration is user-independent, i.e., any user can register any sign while combinations of users are also acceptable.
- Registration is device-independent, i.e., a sign can be registered 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 still can register the sign.
- Registration is extensible, that is if we want to recognize some new signs, we can register them without having any effect on previously registered signs. While with some training based approaches, adding new signs requires training with all signs again, in our approach, we just register new signs and there is no requirement to register previous signs.

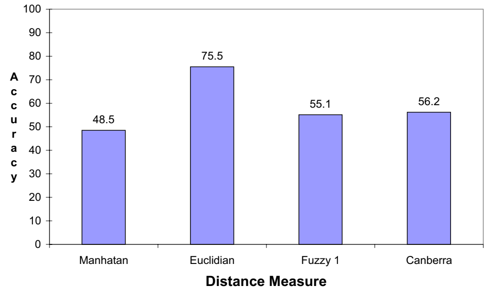


Figure 10. Comparison of different distance measures

To recognize an unknown sign, we identify the best match for that sign in our database. We first collect its sample data and calculate its corresponding NRD. The only requirement is that the starting posture for making this unknown sign should be identical with the starting postures of previously registered signs. We then compute the distance between the calculated NRD of the unknown sign and NRDs of all 'registered signs' in the database and find the one which has the least distance. If the distance is $\leq \epsilon$, the unknown sign is considered identical with the one with the minimum distance.

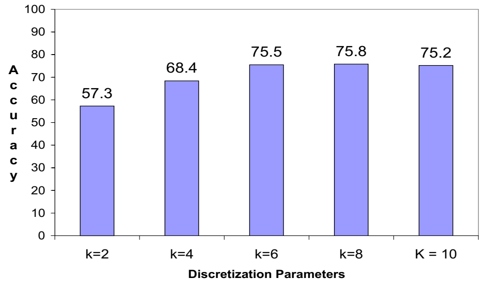


Figure 11. The impact of the discretization value, k

It is necessary to mention that since we are utilizing a high-level semantic data rather than low-level sensor values, it is possible to employ simpler and more extensible approaches for finding the best match (i.e., minimum distance). It is also possible to take advantage of multi-dimensional indexing structures(e.g., R-Tree) to expedite the search process. We have experienced with several distance measures metrics and found that the *Euclidean* distance measure provides the fastest and most accurate result (refer to Section 6).

We now move forward and apply our method to address a more challenging problem, recognizing dynamic signs of type I. In order to do so, we require to identify a series of static signs in the data set. Towards this end, we follow the same procedure as we did in the process of static sign recognition, that is for each coming sample, we calculate its corresponding NRD and find the best match compared to the registered signs . If there exists any NRD with distance less than ε from the unknown NRD, we consider it a match and save it in an output buffer. Since all samples in $C_{[t_i,t_i+\triangle t_i]}|[t_i,t_i+\triangle t_i] \subset [t_0,t_0+\triangle t]$ are representing one static sign, we utilize this buffer to discard repeating signs and consider only one sign for each time period of $[t_i,t_i+\triangle t_i]$. Hence for each set of samples within $[t_i,t_i+\triangle t_i]$, we generate one sign in the output buffer.

We observed that it is not required to calculated NRD for each coming sample. Our experiments demonstrate that as long as one sample within $[t_i,t_i+\triangle t_i]$ is recognized, we achieve the same accuracy as calculating NRD for all samples.

In order to find similar gestures without having them labelled, we consider the case that a large amount of signers make some unknown static signs which do not belong to ASL. Since these signs have no label, we are not able to recognize them by traditional sign recognition techniques. However, in our approach, we can generate a database which keeps all the signs along with their corresponding signatures. To find the similar matches for each sign of interest, we search through the database and find the samples with the the minimum distances. We can find the best match (nearest neighbor) among signatures or find several matches (k-nearest neighbors). We also can utilize various indexing techniques to perform this task faster.

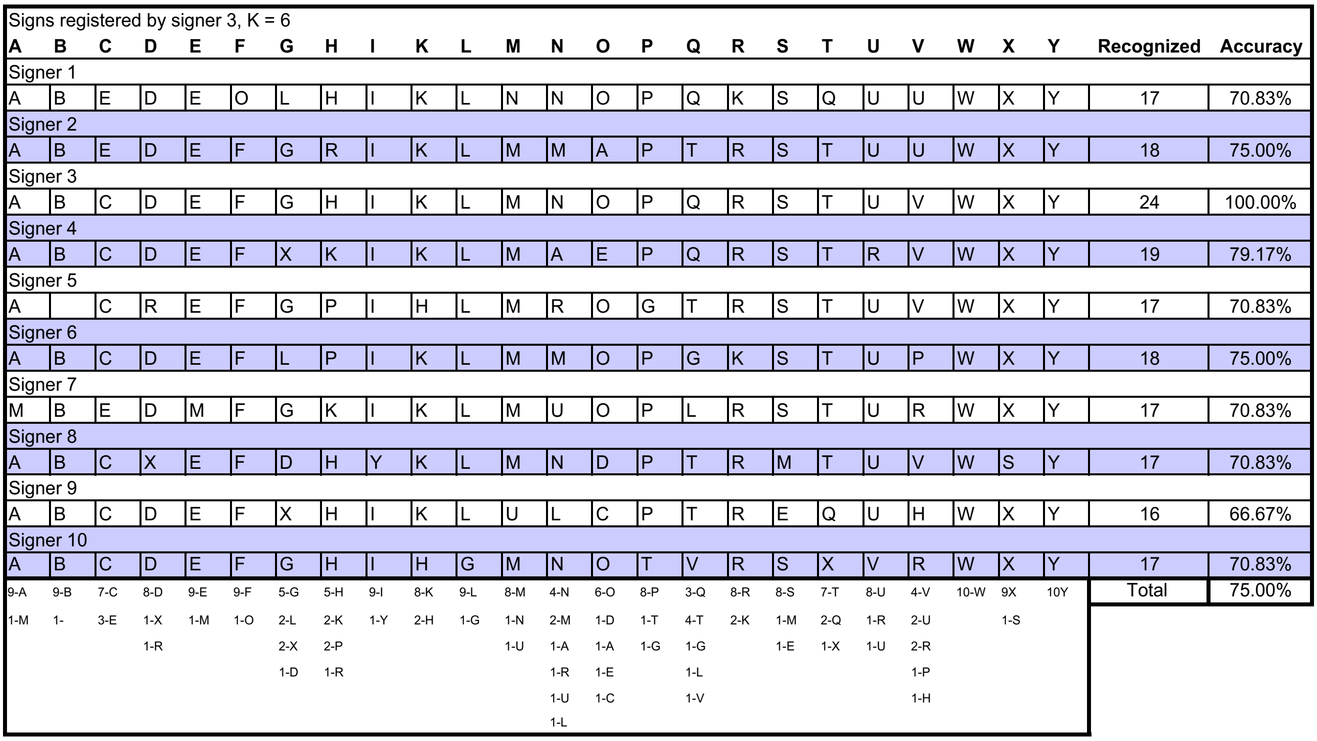
In the next section, we present the result of our experiments.

6. Performance Results

In this section, we present the results of our experiments that we conducted to evaluate our approach in recognizing ASL static signs. The objectives of our experiments were:

1. Specifying the best values for our parameters (e.g., k) to achieve the highest possible accuracy and determining this accuracy (i.e., the number of correctly recognized signs out of 24).
2. Comparing our approach with a conventional approach utilizing neural network.

We will show that our approach achieves at least a comparable performance to the conventional approach, while providing a level of user-independency and device-independency that is beyond the capabilities of any conventional approach. We first explain the setup that we used for our experiments and then present the results.



Signs registered by signer 3, K = 6																									
A	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**I**	**K**	**L**	**M**	**N**	**O**	**P**	**Q**	**R**	**S**	**T**	**U**	**V**	**W**	**X**	**Y**	**Recognized**	**Accuracy**
Signer 1																									
A	B	E	D	E	O	L	H	I	K	L	N	N	O	P	Q	K	S	Q	U	U	W	X	Y	17	70.83%
Signer 2																									
A	B	E	D	E	F	G	R	I	K	L	M	M	A	P	T	R	S	T	U	U	W	X	Y	18	75.00%
Signer 3																									
A	B	C	D	E	F	G	H	I	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	24	100.00%
Signer 4																									
A	B	C	D	E	F	X	K	I	K	L	M	A	E	P	Q	R	S	T	R	V	W	X	Y	19	79.17%
Signer 5																									
A		C	R	E	F	G	P	I	H	L	M	R	O	G	T	R	S	T	U	V	W	X	Y	17	70.83%
Signer 6																									
A	B	C	D	E	F	L	P	I	K	L	M	M	O	P	G	K	S	T	U	P	W	X	Y	18	75.00%
Signer 7																									
M	B	E	D	M	F	G	K	I	K	L	M	U	O	P	L	R	S	T	U	R	W	X	Y	17	70.83%
Signer 8																									
A	B	C	X	E	F	D	H	Y	K	L	M	N	D	P	T	R	M	T	U	V	W	S	Y	17	70.83%
Signer 9																									
A	B	C	D	E	F	X	H	I	K	L	U	L	C	P	T	R	E	Q	U	H	W	X	Y	16	66.67%
Signer 10																									
A	B	C	D	E	F	G	H	I	H	G	M	N	O	T	V	R	S	X	V	R	W	X	Y	17	70.83%
9-A	9-B	7-C	8-D	9-E	9-F	5-G	5-H	9-I	8-K	9-L	8-M	4-N	6-O	8-P	3-Q	8-R	8-S	7-T	8-U	4-V	10-W	9X	10Y	Total	75.00%
1-M	1-	3-E	1-X	1-M	1-O	2-L	2-K	1-Y	2-H	1-G	1-N	2-M	1-D	1-T	4-T	2-K	1-M	2-Q	1-R	2-U		1-S			
			1-R			2-X	2-P				1-U	1-A	1-A	1-G	1-G		1-E	1-X	1-U	2-R					
						1-D	1-R					1-R	1-E		1-L					1-P					
												1-U	1-C		1-V					1-H					
												1-L													

Figure 12. Comprehensive result for the first set of experiments with k=4 and *Euclidean* distance measure

6.1. Experimental Setup

For our experiments, we used CyberGlove [1] as a virtual reality user interface for acquiring 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 and the location of each sensor is shown in Figure 8.

We initiated our experiments by collecting data from one signer wearing the glove and making all static signs ('A' to 'Y' , excluding 'J') from the starting posture as shown in Figure 6. For each sign, we collected 140 tuples from the starting posture to the final posture, each including 22 sensor values. We registered all these signs for this signer and stored the data to be used in the next phase. In the next step, we collected the same amount of samples (140 tuples, 22 sensors) from ten different signers (including the original signer) wearing the glove and making all 24 static signs. The collected data were saved in 240 files, each consist of 140 samples acquired while making one static sign. Since the application that we implemented for our experiment is capable of replaying these stored data as on-line data streams, the collected data would be sufficient for running multiple experiments with various parameters. During our experiments, we varied the discretization parameter k from 2 to 10. We also tested with 4 different distance metrics. Finally, we compared the accuracy of our approach with the result of a neural network based system. We now present the results of our experiments.

6.2. Experimental Results

Clearly the performance of a similarity query (as a component of gesture recognition) is determined largely by the chosen distance metric. So in the first set of experiments, we tested four different distance measures: *Euclidean*, *Manhattan*, *Canberra* [4] and fuzzy [3]. We read all 240 files one by one to recognize the sign. For this experiment, we chose $(k = 6)$ as the discretization parameter and calculated the distances between samples based on each of these distance metrics. The result of this set of experiments is shown in Figure 10. The result reveals that the Euclidean distance measure provides the best result with the highest accuracy of 75.5%.

In the second set of experiments, we repeated the same experiments as the first set while varying the discretization parameters from 2 to 10 incrementing by 2. The result is shown in Figure 11. The figure illustrates that the accuracy improves by increasing the discretization parameter from 2 to 4 and then to 6, but does not change significantly over 6. The observation is that our normalization and discretization methods can improve the accuracy only to some extends. Investigating other normalization and discretization methods are the focus of our future studies.

Considering these results, we repeated the same experiments with $k = 6$ and the *Euclidean* distance measure to find the overall accuracy of our approach. The result of this set of experiments is shown in Figure 12. In this figure, the result of the experiment for each user is shown in one row. Two columns in the right represent the number of correctly recognized signs and the corresponding accuracy, respectively. The table at the bottom of the figure shows what and how many were the signs recognized in 10 experiments for each sign. The result of this experiment shows that our proposed approach is capable of recognizing most of the static signs correctly and the signs which were not recognized correctly are the very similar ones, e.g.,'M' and 'N' or 'E' and 'O', as it shown in Figure 2.

In the last set of experiments, we used a feed-forward back propagation neural network from the *Matlab* neural network package. This network has one hidden layer with 40 neurons. The input layer has 22 neurons (each for one sensor) and 24 neurons (one for each sign) in the output layer. We performed the standard leave-one-out experiment with the subjects, that is, the network was trained with the data sets belonging to nine different users. The training data included the tuples consisting of 22 values received from 22 sensors plus the minimum, maximum, average and median of the items in the tuple. We then tested for the 10th user who was excluded from trainees. We repeated the test 10 times, each time left a different signer out of training and tested with her data set. After averaging, the neural net had the overall accuracy of 67%. We performed a different version of experiments, called keep-one-in with ROMAS. That is we registered all signs with one user and then tested with the data sets belonging to 9 other signers. We repeated this test 10 times, each time registered the signs with a different signer . The overall accuracy of ROMAS for this experiments was 75%. Though keep-one-in is considered a tougher test over leave-on-out (registration by one versus training with nine), the accuracy of ROMAS was higher than neural network approach. In Figure 13, the result of this comparison for each sign is shown.

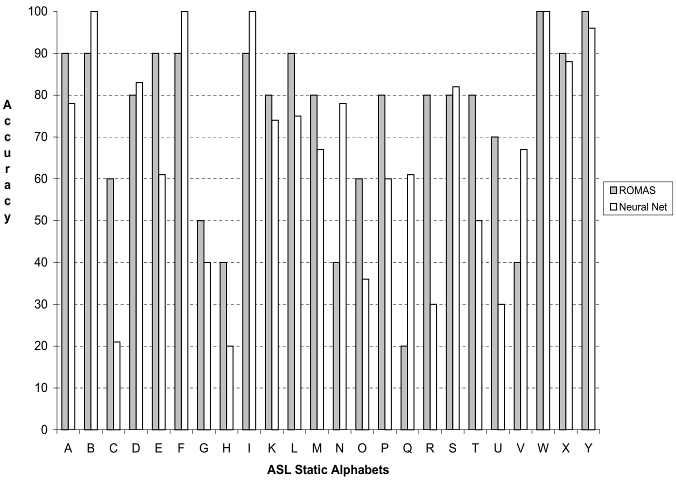


Figure 13. Comparing ROMAS with Neural Network

7. Conclusion and Future work

We proposed an approach for recognizing hand gestures based on the bio-mechanical characteristics of the movement of the hand during the formation of the gesture. Our approach has the following advantages over previous studies:

- It is user-independent
- It is device-independent
- It does not require calibration
- There is no training involved

Our approach is also capable of detecting the similar hand gestures without having them labelled, a problem that most traditional classification methods fail to address. Our experimental results confirm the effectiveness of our approach for recognizing both static ASL signs and dynamic signs type I.

In our future research, we plan to focus on the following issues:

- Investigating different methods of normalization and discretization to improve the accuracy.
- Approaching the recognition process in two steps. In the first step, we will recognize the similar signs (e.g. 'A','C' and 'S') and then fine tune within the recognized group.
- Utilizing the concepts of Context, N-gram letter model and Dictionary to improve the accuracy of fingerspelling.
- Converting our numerical signature of the signs to some form of strings and then investigating the string matching approaches for finding the similar match between signatures.

- Recognizing the dynamic sign of type II.

References

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

[2] K. Boehm, W. Broll, and M. Sokolewicz. Dynamic gesture recognition using neural networks; a fundament for advanced interaction construction. *SPIE Conference Electronic Imaging Science and Technology, San Jose, CA*.

[3] D. V. der Weken, M. Nachtegael, and E. Kerre. Some new similarity measures for histograms. *Proceedings of ICVGIP'2004 (4th Indian Conference on Computer Vision, Graphics and Image Processing, Kolkata, India*, December 2004.

[4] S. M. Emran and N. Ye. Robustness of canberra metric in computer intrusion detection.

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

[6] 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.

[7] K. Murakami and H. Taguchi. Recognition using recurrent neural networks. *Human Interface Laboratory, Fujitsu Laboratories LTD*.

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

[9] N. B. Reese. *Joint Range of Motion*. ISBN 0721689426, 2001.

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

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

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

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