Analysis of Clustering Techniques to Detect Hand Signs

J. Eisenstein, S. Ghandeharizadeh, L. Huang, C. Shahabi, G. Shanbhag, R. Zimmermann
Computer Science Department
University of Southern California
Los Angeles, California 90089

Abstract

The term multimedia has a different meaning to different communities. The computer industry uses this term to refer to a system that can display audio and video clips. Generally speaking, a multimedia system supports multiple presentation modes to convey information. Humans have five senses: sight, hearing, touch, smell and taste. In theory, a system based on this generalized definition must be able to convey information in support of all senses. This would be a step towards virtual environments that facilitate total recall of an experience. This study builds on our previous work with audio and video servers and explores haptic data in support of touch and motor skills. It investigates the use of clustering techniques to recognize hand signs using haptic data. An application of these results is communication devices for the hearing impaired.

I. Introduction

Hand signs are one form of communication for the hearing impaired. Similar to spoken languages, there is no universal sign language. Even for a single sign language, such as English, there exists multiple sign languages, e.g., the British Sign Language and the American Sign Language. Some sign languages, such as Auslan (the Australian sign language), have a very different grammar than English. Auslan reduces the number of words per sentence to enable an individual to sign a sentence in approximately the same amount of time as speaking it.

To communicate unfamiliar words and proper names, sign languages also use an alphabet of signs for each corresponding letter. Spelling words in this manner can also be used between people who are not familiar with the concise sign languages. Some signs are expressed as static gestures while others incorporate some dynamic hand movement. For static gestures, a specific moment in time captures when the sign is most prominently displayed. For dynamic gestures a complete sequence of finger and hand positions needs to be recognized. Segmenting a continuous sequence of sensor values into the proper signs (either static or dynamic) is one of the challenging aspects of automated sign recognition.

The focus of this study is on static gestures with a single hand. We strive the detect a hand sign from a continuous stream of haptic data generated by a glove. The rest of this paper is organized as follows. Section II provides a general framework to detect hand signs. This framework consists of three steps. Next, Sections III and IV flush out the details of the first two steps of this framework. In particular, Section IV presents two clustering algorithms: K-Means and Adaptive. The experimental results of Section V demonstrate that a simple application of these clustering techniques is sensitive to the training data set size and its presentation. The related work is presented in Section VI. Section VII offers brief conclusions and future research directions.

II. A Framework for Sign Recognition

This paper explores a simple framework to detect hand signs that consist of two different stages: a training phase, and a lookup phase. During the training phase, the user generates different signs multiple times while wearing a haptic glove. The system employs a clustering technique [1], [2] and constructs clusters that correspond to these signs. During the lookup phase, the user generates signs while wearing the haptic glove. The system identifies signs by comparing it with the clusters and identifying the best match. These two stages, along with a third, are formalized as follows:

1. Gather raw data from a haptic glove as a function of time. While this may appear straightforward with our simplified objective, it is important for our future extensions that detect continuous hand signs as a function of time, see Section VII.
2. This step either (a) constructs clusters when being trained or (b) detects signs by accepting input from the user.
3. This step translates the detected signs, representing individual characters, into words. This step may employ context to compensate for temporal noise introduced such as repeated characters introduced by Step 2.

Appeared in International Symposium on Intelligent Multimedia, Video and Speech Processing, Hong Kong, May 2001. This research was supported in part by the following NSF grants: IIS-0091843 (SGER), EEC-9529152 (IMSC ERC) and ITR-0082826.

In the following, we describe the first two steps in detail.

III. Raw Data Gathering

Haptic device development is still in a very early stage. We have focused our study on one such device, the CyberGrasp exoskeletal interface and the accompanying CyberGlove from Virtual Technologies. It consists of 33 sensors, see Table I. In our experiments we use the CyberGrasp SDK to write handlers that record sensor data whenever a particular sampling interrupt was called. The rate at which these handlers were called was thus the maximum rate we could sample varied as a function of the CPU speed.

To sample and record the data asynchronously we developed a simple multi-threaded double buffering approach. One thread was associated with responding to the handler call and copying sensor data into a region of system memory. A second thread asynchronously writes this data to disk. The CPU was never 100% utilized during this process. This prevents our recording process from interfering with the rendering process itself. Further there is obvious room for optimization here as we could run our experiments on dual processor machines and adjust priority for the second thread. Other optimization techniques are described in [3].

IV. Clustering of Data

Clustering classifies objects with no supervision. This study assumes a simplified environment consisting of two steps: training and data lookup. During training, a user issues a fixed number of hand signs and repeats them several times. The system detects the different clusters that represent each class with no aprior knoledge of classes. The user then assigns a label to each cluster. During lookup, the user repeats a hand sign and the system compare it with the available clusters to identify the best match. In this section, we describe: a) how the system constructs clusters during training, and b) how the system looks up a sign and compares it with a cluster.

The training phase may utilize different clustering algorithms. K-Means [1], [4], [5] is probably the most popular clustering algorithm. It requires the user to specify the number of classes K, where each class corresponds to a sign. It forms the clusters by minimizing the sum of squared distances from all patterns in a cluster to the center of the cluster. The pattern samples are constructed using the first 22 sensor values of Table I that pertain to the position of different joints that constitute a hand. A main assumption of K-Means is that clusters are hyper-ellipsoidal.

Adaptive clustering [2], [6] determines the number of clusters based on training data. It is more general than K-Means because it does not require aprior knowledge of K. It chooses the first cluster center arbitrarily. It assigns an input training record to a cluster when the distance from the sample to the cluster is below $\theta \times \tau$ where: θ is the distance threshold while τ is a fraction between 0 and 1. Adaptive does not create a new cluster when this distance is greater than τ. Moreover, it does not make a decision when the sample record falls in an intermediate region. Once the training ends, it assigns all patterns to the nearest class according to the minimum distance rule, i.e., Euclidian distacne. It may leave some patterns as unclassified if their distances to all cluster centers are greater than τ.

V. Experimental Results

For experimental purposes, we used 10 subjects performing 10 different hand signs (nine corresponding[1] to letters 'A' through 'I' plus letter 'L'). We used an implementation of K-Means and Adaptive provided by a package named Numerical Cruncher [7] for experimental purposes. With Adaptive, we used $\theta = 0.8$ and $\tau = 4.1$. These values were chosen to guide[2] Adaptive to construct 10 clusters.

The results are dependent on the input data, its size and how it is presented to a given algorithm. Generally speaking, K-Means is most sensitive, providing an accuracy that ranges between 55% to 83% (depending on the input data and the order it is presented to the algorithm). Adaptive is less sensitive with its accuracy ranging between 66% to 77%.

In our first experiment, we used a hand sign from each subject in order to come up with 100 data points for training purposes. With this data set, K-Means can detect input hand signs with 80% accuracy while Adaptive can accurately identify 77% of the signs. Next, we varied the number of hand signs from each subject for training purposes. This was varied from 2 to 4 and 6 signs per subject for a total of 200, 400 and 600 samples, respectively. The results demonstrate that both algorithms provide varying degrees of accuracy depending on how the samples are presented to the clustering algorithm. In our experiments, Adaptive proved to be less sensitive than K-Means.

As a comparison, we used a classification algorithm, K Nearest Neighbor (termed KNN) to compare with both K-Means and Adaptive clustering. For each data point X, KNN constructs a hypersphere centered on X that is just big enough to include K nearest neighbors (its similarity func-

[1] The letter 'J' was skipped because it is not a static sign. It requires the subjects to move their fingers while performing the sign.

[2] This is based on trail and error by manipulating θ and τ multiple times. It took us four trails to realize ten clusters.

Sensor Number	Sensor Description	Sensor Number	Sensor Description
1	Thumb roll sensor	14	Ring outer joint
2	Thumb inner joint	15	Ring middle abduction
3	Thumb outer joint	16	Pinky inner joint
4	Thumb index abduction	17	Pinky middle joint
5	Index inner joint	18	Pinky outer joint
6	Index middle joint	19	Pinky ring abduction
7	Index outer joint	20	Palm arch
8	Middle inner joint	21	Wrist flexion
9	Middle middle joint	22	Wrist abduction
10	Middle outer joint	23,24,25	X,Y,Z location
11	Middle index abduction	26,27,28	X,Y,Z abduction
12	Ring inner joint	29 to 33	Forces for each finger
13	Ring middle joint		

TABLE I
CyberGrasp sensors.

tion is based on Euclidean distance). Unless stated otherwise, we set $K = 5$ for all experiments. The results demonstrate that KNN provides the best accuracy when compared with both K-Means and Adaptive, providing 81% to 88% accuracy. With large training sets, 2000 samples (20 samples per sign per subject), the accuracy of KNN increases to 95%. With such a large sample set, the accuracy of KNN drops when we increase the value of K from 1 to 15.

In a final experiment, we focused on a single subject repeating a sign ten different times for training purposes. (With ten different signs, the training sample size consists of one hundred elements.) Once again, K-Means was most sensitive with its accuracy ranging between 48% to 66%. Adapative provided better accuracy ranging between 57% to 77%. KNN provides the best performance with 96% to 100% accuracy when K equals to one. With larger K values, the accuracy of KNN drops (90% accuracy when K equals to 5). These results demonstrate K-Mean and Adaptive's sensitivity to the training data set size and how it is presented. KNN, as a classification algorithm, provides superior performance when it is provided with a large, redundant training set size.

VI. Related Work

Gesture Recognition is investigated by various research groups world-wide. We are aware of two main approaches:

1. Machine-Vision: this approach analyzes the video and image data of a hand in motion. This includes both 2D and 3D position and orientation of one or two hands.

2. Haptic approach: similar to this study, the basic idea is to gather and analyze haptic data from a a glove. The data is basically quantified values of the various degrees of freedom of the hand. These efforts resulted in development of devices such as the CyberGlove.

We taxonomize the first approach based on the employed technique. Darrell et. al [8] discusses vision based recognition with "Template Matching". Heap et. al [9] employs *Active Shape* models. Several studies, [10] and [11] propose to use the Principal Component Analysis. Yet another method of recognition using linear fingertips is described in [12]. Banarase [13] uses a *neocognitron* network. Just as us, various researchers have tried to "recognize" various sign languages all over the world with different methods. These include the American (ASL), Australian (AUSLAN), Japanese (JSL), and Taiwanese (TWL) Sign Languages, to name a few. Getting to the most relevant, the ASL recognition has been performed by numerous groups; [14], [15] and [16], to cite a few who use Hidden Markov Models. An excellent survey of vision-based gesture-recognition methods is provided in [17].

With gloves and haptic data, Fels et. al [18] employs a VPL Glove to carry out gesture recognition with Back-Propagation neural networks. Sandberg [19] provides an extensive coverage and employs a combination of *Radial Basis Function Network* and *Bayesian Classifier* to classify a hybrid vocabulary of static and dynamic hand gestures. One more flavor is added with [20] using *Recurrent* neural networks to classify Japanese sign language. As with vision based, Hidden Markov Models is a popular tool here too, which is reflected in [21] and [22]. The work of Lee et. al in [22] is particularly relevant because it presents an application for learning of gestures through Hidden Markov Models and the data input is from CyberGlove. In [23], they use instance based learning to classify Australian Sign Language. Newby and Gregory [24] have done a glove based template matching. In [25], they propose feature extraction, while Takahashi et. al [26] preferred to use PCA on the glove input.

Similar to those studies that analyze haptic data, this study is orthogonal to the machine-vision approaches. It is different than studies that utilize haptic data in that is focuses on the role of clustering techniques and how well they detect hand signs.

VII. Conclusion and Future Research Directions

This study compares two different clustering techniques to detect hand signs: K-Means and Adaptive. Detection of hand signs is important for the design and implementation of communication devices for hearing impaired. Our obtained results demonstrate that the accuracy of clustering techniques is sensitive to the characteristics of training data. A larger training set does not necessarily improve the accuracy of a clustering technique. This is because the clusters are formed incrementally as the training data is presented to the algorithm. A more dynamic version of clustering, Adaptive, is less sensitive because it delays assignment of samples to clusters when the sample does not match with a cluster well[3].

This study is preliminary and we plan to extend it in many directions. First, we intend to compare clustering with other machine learning approaches investigated in [27] to detect hand signs from haptic data. Second, a realistic algorithm must consider temporal characteristics of data. For example, in this study, we skipped the letter 'J' because it is not a static sign. Instead, it requires the subject's hand to move as a function of time. The role of temporal data becomes more profound with more complex sign languages that require temporal gestures to express a sentence, e.g., Auslan. In this case, the three step process of Section II must be re-visited in order to accommodate the temporal characteristics of data. The simple task of gathering data becomes complex because it must now detect when to start sampling the data stream. If it does not start in a timely manner then it might miss the start portion of a gesture. In this case, it might be useful to extend the concept of a context from layer 3 down to the lower layers when gathering data.

References

[1] J. B. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, 1967, pp. 281–297.

[2] M. J. Martin-Bautista and M. A. Vila, "A survey of genetic selection in mining issues," in *IEEE Conference on Evolutionary Computation*, 1999, pp. 1314–1321.

[3] C. Shahabi, M. R. Kolahdouzan, G. Barish, R. Zimmermann, D. Yao, K. Fu, and L. Zhang, "Alternative Techniques for the Efficient Acquisition of Haptic Data," in *To Appear in the Proceedings of ACM SIGMETRICS 2001*, June 2001.

[4] A. K. Jain and R. C. Dubes, *Algorithms for Clustering Data*, Prentice Hall, Reading, MA, 1988.

[5] R. Ng and J. Han, "Efficient and Effective Clustering Method for Spatial Data Mining," in *Proceedings of 20th International Conference on Very Large Databases*, 1994.

[6] R. Carrasco, J. Galindo, J.M. Medina, and M.A. Vila, "Clustering and classification in a finantial data mining environnment," in *Third International ICSC Symposium on Soft Computing*, 1999, pp. 713–772.

[7] F. B. Galiano and J. C. C. Talavera, "Data mining software," in *Front DB Research*, 1999.

[8] T. Darrell and A. Pentland., "Recognition of space-time gestures using a distributed representation," Tech. Rep., 1993.

[9] A. Heap and F. Samaria, "Real-time hand tracking and gesture recognition using smart snakes," in *Proceedings of Interface to Real and Virtual Worlds*, 1995.

[10] J. Martin and J. Crowley, "An appearance-based approach to gesture recognition," in *Proceedings of Ninth International Conference on Image Analysis and Processing*, 1997, pp. 340–347.

[11] H. Birk, T. Moeslund, and C. Madsen, "Real-time recognition of hand Alphabet gestures using Principal Component Analysis.," in *Proceedings of The 10th Scandinavian Conference on Image Analysis*, 1997.

[12] D. James and M. Shah, "Gesture recognition," Tech. Rep., 1993.

[13] D. Banarase, "Hand posture recognition with the neocognitron network," Tech. Rep., 1993.

[14] C. Vogler and D. Metaxas, "Adapting Hidden Markov Models for asl recognition by using three-dimensional computer vision methods," in *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, 1997, pp. 156–161.

[15] T. Starner and A. Pentland, "Real-time american sign language recognition from video using hidden markov models," Tech. Rep., MIT, 1996.

[16] T. Starner, "Visual recognition of american sign language using hidden markov models," 1995.

[17] Y. Wu and T. Huang, "Vision-based gesture recognition: A Review," in *Proceedings of the International Gesture Recognition Workshop*, 1999, pp. 103–115.

[18] S. Fels and G. Hinton, "Glove-talkii: An adaptive gesture-to-format interface.," in *Proceedings of CHI95 Human Factors in Computing Systems*, 1995.

[19] A. Sandberg, "Gesture recognition using neural networks," 1997.

[20] K. Murakami and H. Taguchi, "Gesture recognition using recurrent neural networks.," in *Proceedings of CHI91 Human Factors in Computing Systems*, 1991.

[21] Y. Nam and K. Wohn, "Recognition of space-time hand-gestures using hidden markov model," in *Proceedings of ACM Symposium on Virtual Reality Software and Technology*, 1996, pp. 51–58.

[22] C. Lee and X. Yangsheng, "Online interactive learning of gestures for human/robot interfaces.," in *Proceedings of IEEE International Conference on Robotics and Automation*, 1996, pp. 2982–2987.

[23] W. Kadous, "Grasp: Recognition of australian sign language using instrumented gloves," 1995.

[24] G. Newby, "Gesture recognition using statistical similarity," in *Proceedings of Virtual Reality and Persons with Disabilities*, 1993.

[25] D. Rubine, "Specifing Gestures by Example," in *Proceedings of SIG-GRAPH91*, 1991.

[26] T. Takahashi and F. Kishino, "Hand gesture coding based on experiments using a hand gesture interface device," *SIGCHI Bulletin*, vol. 23, no. 2, 1991.

[27] C. Shahabi, L. Kaghazian, S. Mehta, A. Ghoting, G. Shanbhag, and M. McLaughlin, "Analysis of Haptic Data for Sign Language Recognition," in *To Appear in the International Conference on Universal Access in Human-Computer Interaction*, August 2001.

[3] One can quantify how well a sample matches by analyzing the value of θ and τ.