Analysis of Haptic Data for Sign Language Recognition[1]

C. Shahabi, L. Kaghazian, S. Mehta, A. Ghoting, G. Shanbhag and M. McLaughlin

Integrated Media Systems Center and Computer Science Department
University of Southern California
Los Angeles, CA 90089-0781
{shahabi, kaghazia, srm, ghoting, shanbhag, mmclaugh}@usc.edu

Abstract: For the past two years we have been addressing the challenges involved in managing the data generated within immersive environments. We together with many other researchers have addressed the management of obvious data types such as image, video, audio and text. However, we identified a set of less familiar data types, collectively termed as *immersidata*, that are specific to immersive environments. In this paper, we focus our attention on analysis of a kind of immersidata known as *haptic* data. We propose to analyze the haptic data acquired from CyberGlove to recognize different static hand signs automatically. The ultimate objective is to understand how to model and store haptic data in a database, for similar types of applications. We propose several techniques to analyze subtle changes in hand signs and words (a series of signs). We show that our techniques can recognize the most important features to distinguish between two letters and several preliminary experiments demonstrate more than 84.66% accuracy in sign recognition for a 10-sign vocabulary.

1 INTRODUCTION

Recently, several research efforts have been directed towards immersive environments. Such environments can facilitate the virtual interaction between people, objects, places and databases. Data types such as image, audio, video and text are an integral part of immersive environments and many in the past have addressed their management. However, we identified a set of less familiar data types, collectively termed as *immersidata* [SBE99], that are specific to immersive environments. *Haptic* data-type is a kind of immersidata. CyberGlove is used as a haptic user interface and it consists of several sensory devices that generate data at a continuous rate. The acquired data can be stored, queried and analyzed for several applications. In this paper, we focus our attention on the analysis of haptic data with the objective of modeling these data in a database. An application may need haptic data stored and modeled at different levels of abstraction. For now, we consider three levels of abstraction. First, in [SBK 01], we made our first attempt to model haptic data conceptualized as time series data sets, at the lowest level of abstraction. Second, in this paper, we move a level up from our previous work in using raw haptic data, by trying to understand the *semantics* of hand actions, and we employ several learning techniques to develop this understanding. The application that we focus on is *limited vocabulary American Sign Language recognition* that involves the translation of American Sign Language (ASL) to spoken words. Finally, for the third level of abstraction, there exists a class of applications that need to analyze *pre-processed* data as opposed to analyzing raw haptic data. We intend to study this final level of abstraction as part of our future work.

We investigate three different analysis techniques for the automatic recognition of signs and evaluated their accuracy over a 10 sign vocabulary. We use C4.5 Decision Tree, Bayesian Classifier and Neural Networks for the recognition of static signs. Our experiments show that Bayesian Classifier can classify haptic data with an average error of 15.34%. Bayesian Classifier appears to be the fastest classification technique providing the best classification accuracy for our experiments.

Our research is distinct and novel in the following three aspects. To begin with, we are distinct with respect to the framework we have used for our research and experiments. All our analysis and experimentations were performed on raw haptic data without any kind of pre-processing. In addition to a novel framework, we have taken a new approach for modeling haptic data, which is based upon learning techniques such as Decision Trees, Bayesian Classifier and Neural Networks. The comparison of these three techniques within the same environment and

[1] This research has been funded in part by NSF grants EEC-9529152 (IMSC ERC) and ITR-0082826, NASA/JPL contract nr. 961518, DARPA and USAF under agreement nr. F30602-99-1-0524, and unrestricted cash/equipment gifts from NCR, IBM, Intel and SUN.

experimental setup is also novel and unique. Finally, the ultimate objective of our research is to model and store haptic data at different levels of abstractions.

The remainder of this paper is organized as follows. Section 2 discusses how we acquire haptic data using CyberGlove. Section 3 explains the three learning techniques that we use for sign recognition. The results of our experiments in comparing the three analysis techniques have been reported in Section 4. Section 5 covers various other research efforts in the area of sign language recognition. Finally, Section 6 concludes this paper and provides pointers to our future research plans.

2 DATA ACQUISITION

The development of haptic devices is in a very early stage. We have focused our research and experiments on the CyberGrasp exoskeletal interface and accompanying CyberGlove from Virtual technologies. It consists of 33 sensors as described in [SBK 01]. We developed a multi-threaded double buffering technique to sample and record data asynchronously. We used 10 alphabets (A to I and L) from the American Sign Language (ASL) for our experiments. We term each of these 10 alphabets a *sign*. The 22 sensor values (excluding sensors 23 to 33 in [SBK 01]) are recorded in a log file for each sign made by a *subject*, termed as a *session*. Each session log file contains thousands of rows of sensor values sampled at some frequency, which depends on the sampling technique used. We denote each such row as a *snapshot*. We thus have thousands of snapshots for each session.

2.1 Sampling Techniques

In order to record several snapshots for each static sign, made within a session, we need to sample the values of sensors for each subject making a sign. Thus sampling the sensors at a rate, which would lead to lower storage space requirements and better accuracy is central to the task of data acquisition for any haptic device. We designed and implemented the following sampling techniques for our experiments (see [SBK 01] for more details).

First, Fixed Sampling was used to record the session. This technique is wasteful since it records data for each sensor at each possible opportunity regardless of the sensor type or the semantics of the session. Next, Group Sampling was used to record the session. We can isolate a sampling rate for each group and acquire data at different rates, based upon the group membership for each sensor. The advantage of this technique is its improvement over the fixed sampling technique by further reducing storage space and transmission requirements, while maintaining accuracy. Finally, Adaptive Sampling was used to sample the sensors. This is a dynamic form of sampling where we try to find an optimum rate r_{ij} for each sensor i during a given window j of the session. The benefit of this technique is that the sampling rate changes with the nature of the sessions. This makes the approach more efficient and robust as compared to the fixed or group sampling approach. In [SBK 01], we provide details on these sampling techniques and the various tradeoffs among factors like bandwidth, storage and computational complexity.

3 CLASSIFICATION METHODS

In this paper we explore three different classification techniques, and evaluate the accuracy of each technique to detect 10 different hand signs. The employed techniques are C4.5 Decision Tree, Bayesian Classifier and Neural Networks. Each classification technique is implemented in two different stages, the training phase and the recognition phase.

3.1 C4.5 Decision Tree

Tree induction methods are considered to be supervised classification methods, which generate decision trees derived from a particular data set. C4.5 uses the concept of information gain to make a tree of classificatory decisions with respect to a previously chosen target classification [Quinlan 93]. The output of the system is available as a symbolic rule base. The cases, described by any mixture of nominal and numeric properties, are scrutinized for patterns that allow the classes to be reliably discriminated. These patterns are then expressed as models, in the form of decision trees or sets of *if-then* rules, which can be used to classify new cases, with an emphasis on making the models understandable as well as accurate [Quinlan 93]. For real world databases the decision trees become huge and are always difficult to understand and interpret. In general, it is often possible to prune a decision tree to obtain a simpler and more accurate tree.

We employed C4.5 Decision Tree because it provides a model to build a sign recognition language. In addition, decision trees in general and C4.5 in specific provide results as a set of understandable and interpretable rules. Finally, C4.5 has been used as a benchmark in several other works in machine learning, artificial intelligence and data mining literature. C4.5 complexity is $O(nt)$ where t is the number of tree nodes and the number of tree nodes often grows as $O(n)$ where n is the number of sessions. The complexity for non-numeric data would be $O(n^2)$, for numeric $O(n^2 \log n)$, and for mixed-type data, somewhere in between.

3.2 Bayesian Classification

Bayesian Classifier is a fast-supervised classification technique. Bayesian Classifier is suitable for large-scale prediction and classification tasks on complex and incomplete datasets. Naïve Bayesian Classification performs well if the values of the attributes for the sessions are independent. Although this assumption is almost always violated in

practice, recent work [DP96] has shown that naïve Bayesian learning is remarkably effective in practice and difficult to improve upon systematically. We have decided to use the naive Bayesian Classifier in our application, for the following reasons. First, it is efficient for both the training phase and the recognition phase. Second, its training time is linear in the number of examples and its recognition time is independent of the number of examples. Finally, It provides relatively fine-grained probability estimates that can be used to classify the new session [Elk97]. The computational complexity of Bayesian Classification is fairly low as compared to other classification techniques. Consider a session with *f* attributes, each with *v* values. Then with naive Bayesian classifier with *e* sessions, the training time is *O(ef)* and hence independent of *v*.

3.3 Neural Network

We use Neural Networks for the recognition of static signs with a limited vocabulary. Supervised learning is being used for the classification. A Multi Layer Perceptron is a feed-forward network with one or more layers of nodes between the input and output layers of nodes. These additional layers contain hidden nodes that are not directly connected to both the input and the output nodes. The capabilities of the multi-layer perceptrons stem from the non-linearity used in these nodes. The number of nodes in the hidden layer must be large enough to form a decision region that is as complex as required by a given problem. A Multi Layer Perceptron (MLP) is trained using the Supervised-learning rule. The most commonly used algorithm for such training is the Error-back-propagation-algorithm. A three-layer perceptron can form arbitrarily complex decision regions. Hence, usually, most problems can be solved by 3-layer (1 hidden layer) perceptrons.

3.3.1 Implementation of Neural Network Classification over Static Signs:

We used 22 nodes for the input layer, in addition to one threshold value node for the next layer. The hidden layer required 10 nodes. Output layer required 10 nodes, each corresponding to one sign. With each of the 23 inputs (22 haptic glove values + 1 threshold) connected to each of the 11 hidden layer neurons (10 neurons + 1 threshold), and again each of these hidden layer neurons being connected to each of the 10 output neurons, the total number of weights in the network is equal to $(23\times10)+(11\times10) = 340$ weights.

For our experiments, we establish the cardinality of the training set, to achieve a good generalization as propounded in [VC71], approximately 10 times more, to cross the "*VCDim*" threshold. Hence, we have training data set of 10 subjects, each making 10 signs, and for each session we have 40 snapshots, resulting in 4000 sets of sensor values. We train the network for 500 epochs. We generate pseudo-random weights, the range of which is –1.0 to +1.0. We strived to make the neural network learn on raw haptic data so that it learns to handle noisy data. This can be useful when we try to use the classifier in real-time immersive applications. Work done by [WH99] yields very good results on the training set, but the ability of this approach to be generalized needs to be ascertained. Our approach provides a good promise for an overall generalization.

4 PERFORMANCE EVALUATION

4.1 Experimental Setup

We conducted several preliminary experiments to evaluate each classification method. Fifteen subjects were asked to generate the following signs: *a*, *b*, *c*, *d*, *e*, *f*, *g*, *h*, *i* and *l*, and data has been stored in a database. The signs *j and k* are complicated and taking the novice subjects into consideration, the signs were skipped for simplicity reasons. To evaluate each algorithm we used the cross validation technique. We split the data into 3 sets, trained the system using two of the sets and conducted the tests using the third set. We implemented the test procedure in a round robin fashion (shuffling the training and test sets) and computed the average error.

4.1.1 Storage of the Input

Neural Network is trained using 4000 snapshots, as described earlier. These data are retrieved from 100 session log files (10 subjects, 10 signs each, 40 different snapshots). For our experiments on static signs, we analyzed the recorded log files stored in a database and extracted the snapshot that has the sensor values consistent over a substantial period of time. Classification algorithms can then be developed using incremental learning.

4.2 Results

Figures 1and 2 compare the average recognition error for each sign using the three different classification techniques. The naive Bayesian Classifier has the highest average accuracy with 50 training examples: 84.66% (with standard deviation sd = 2.94).

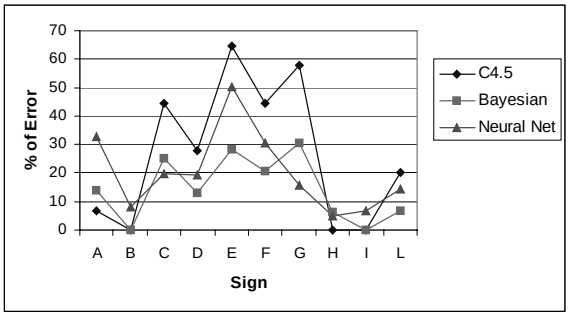

Figure 1: Sign recognition error

Sign	A	B	C	D	E	F	G	H	I	L
										
C4.5	**6.6**	**0.0**	44.4	27.7	64.4	44.4	57.7	**0.0**	**0.0**	20.0
Bayesian	14	**0.0**	25	**13.06**	**28.4**	**20.83**	30.52	6.4	**0.0**	**6.4**
Neural Net	32.8	7.87	**19.6**	19.5	50.04	30.55	**15.73**	4.9	6.67	14.24

Figure 2: Sign recognition error comparison

	Error	Standard Derivation
C4.5	22	8
Bayesian	15.34	2.94
Neural Net	20.18	7.92

Figure 3: Overall classification error

4.3 Analysis

The Bayesian Classifier shows a very efficient and accurate result as compared to other classification techniques. The results of our experiments illustrate that C4.5 Decision Tree is not suited to the task of sign recognition. Both Neural Networks and C4.5 have a large amount of variation in their performance. However, most often, C4.5 results are more interpretable and understandable. In contrast, the Neural Network architecture and procedure are not interpretable, and it is similar to a black box in which case, we only have access to input and output. Our experiments indicate that all of the classifiers relatively performed quite well on signs 'b', 'h', 'i' and 'l'. Inspecting the signs, it occurs that it was intuitive for subjects to perform these signs most consciously. Considering all the signs as points in 22-dimension hyperspace, and computing the Euclidian distance among them, we realized that on the average, these four signs are quite apart from the rest of the signs, which justifies our observation. On the other hand, letter 'e' was quite close in distance to all the other signs and hence all classifiers were confused one way or the other with the recognition of letter 'e'.

The performance variation of individual classifiers over the signs can be traced back to the performance characteristics of each classifier. A neural network, inherently tries to draw crisp distinguishing boundaries between groups of signs in the 22-dimensional hyperspace. Hence, it distinguished all the signs made when the hand is in the horizontal position (i.e., 'c', 'g' and 'h') quite well. Note that although C4.5 was the best classifier for letter 'h', it had the minimum recognition error among the other letters with the neural net. With C4.5 and Bayesian Classifiers, the main assumption is that all features in a given space are independent. In general any strong dependency increases the level of error for both methods, while a low degree of dependency among features might be negligible. Further, since C4.5 produces decisions based on a set of 'if-then' rules, it tends to be relatively rigid, resulting in a high standard deviation as well as a high overall error. Since Bayesian classifier decides based on probability distribution of the input samples, it tends to perform quite well overall despite intuitive variations in performance of signs by different subjects. We illustrate that even with a small pool of snapshots, a fast learner such as Naïve Bayesian Classifier and an appropriate I/O design we can achieve an acceptable performance.

5 RELATED WORK

Various research groups worldwide have been investigating the problem of sign recognition. We are aware of two main approaches. Machine-Vision based approaches analyze the video and image data of a hand in motion. The

Haptic based approaches analyze the haptic data from a glove. These efforts have resulted in the development of devices such as CyberGlove. Due to lack of space, we refer the interested readers to [WH99] for a good survey on vision based sign recognition methods.

Using gloves and haptic data, Fels et. al [FH95] employ a VPL Glove to carry out sign recognition. Sandberg [San97] 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 signs. One more variant exists [MT91] using Recurrent Neural Networks to classify Japanese sign language. Hidden Markov Models are popular here too, which is reflected in [NW96] and [LY96]. The work of Lee et. al in [LY96] is particularly relevant because it presents an application for learning of signs through Hidden Markov Models taking the data input from CyberGlove. Other neural network algorithms- Radial Basis Function Network, Orthogonal Least Squares and Self – Organizing maps have also been tested on various kinds of data-glove inputs, in [LIN98] and [IM99].

Our work is distinguished from all of the above-mentioned works, because we provide a complete system including I/O unit, data acquisition module, database structure and classification methods for American Sign Language recognition. All our analysis is carried out on raw haptic data. We are the first to use Decision Tree for the analysis of haptic data. We are also the first to use Bayesian Classifier for *raw* (i.e. no pre-processing) haptic data analysis. Taking our framework into consideration, we are also the first to use and compare Back Propagation Neural Networks with Bayesian Classifier and C4.5 Decision Tree for the recognition of static signs.

6 CONCLUSION AND FUTURE WORK

In this paper, we analyzed three different classification techniques for sign language recognition. We showed that Decision Tree, Bayesian Classifier and Neural Networks could be used for American Sign Language recognition. Bayesian Classifier proved to be the fastest classification technique amongst the three classification techniques. It also proved to have the best classification accuracy for static sign recognition. We carried out several preliminary experiments and the results of our experiments suggest that Bayesian Classifier can be used to develop a real time sign language recognition system. However, more work needs to be carried out in order to establish the validity of our results, which are very encouraging in the early stages of experimentation.

We intend to extend our work in several ways. First, we intend to investigate Time Delay Neural Networks and Evolving Fuzzy Neural Networks for the recognition of *dynamic* signs. It would be interesting to compare the effectiveness of these two techniques. Second, we want to utilize the lessons learned in our analysis of haptic data to model haptic data in a database. Our analysis would determine what data we need to store at which level of abstraction for a given application. Third, we would like to analyze haptic data at the third level of abstraction, which requires us to analyze pre-processed haptic data. Finally, we propose to use shape recognition techniques for the recognition of dynamic signs based upon a fixed sign language vocabulary. Using a shape to represent the dynamic part of a sign would let us view the dynamic sign in a time independent manner.

7 REFERENCES

[DP 96] P. Domingos, , M. Pazzani.Beyond Independence: Conditions for the Optimality of the Simple Bayesian Classifier Proceedings of the Thirteenth International Conference on Machine Learning (pp. 105-112), 1996. Bari, Italy: Morgan Kaufmann.

[ELK 97] Elkan, C. (1997), Boosting and Naive Bayesian learning, In proceeding of KDD -97, New Port beach, CA.

[FH95] S. Fels and G. Hinton. Glove-talkii: An adaptive gesture-to-format interface. In Proceedings of CHI95 Human Factors in Computing Systems, 1995.

[IM99] Ishikawa, M.; Matsumura, H, Recognition of a hand-gesture based on self-organization using a DataGlove. Neural Information Processing, 1999. Proceedings. ICONIP '99. 6th International Conference on , Volume: 2 , 1999 Page(s): 739 -745 vol.2

[LIN98] Daw-Tung Lin, Spatio-temporal hand gesture recognition using neural networks Neural Networks Proceedings, 1998. IEEE World Congress on Computational Intelligence. The 1998 IEEE International Joint Conference on , Volume: 3 , 1998 Page(s): 1794 -1798 vol.3

[LY96] C. Lee and X. Yangsheng. Online interactive learning of gestures for human/robot interfaces. In Proceedings of IEEE International Conference on Robotics and Automation, pages 2982-2987, 1996.

[MT91] K. Murakami and H. Taguchi. Gesture recognition using recurrent neural networks. In Proceedings of CHI91 Human Factors in Computing Systems, 1991.

[NW96] 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, pages 51-58, 1996.

[Quinlan 93] J.R. Quinlan, C4.5: Programs for Machine Learning, Morgan Kaufmann, 1993.

[San97] A. Sandberg. Gesture recognition using neural networks, 1997.

[SBE99] C. Shahabi, G. Barish, B. Ellenberger, N. Jiang, M. Kolahdouzan, S. Nam and R. Zimmermann, Immersidata Management: Challenges in Management of Data Generated within an Immersive Environment, In proceedings of the Fifth International Workshop on Multimedia Information Systems, 1999

[SBK 01] C. Shahabi and M. R. Kolahdouzan and G. Barish and R. Zimmermann and D. Yao and K. Fu and L. Zhang, Alternative Techniques for the Efficient Acquisition of Haptic Data, to appear in Sigmetrics'2001.

[VC71] Vapnik, V. N. and Chervonenkis, *On the uniform convergence of relative frequencies of events to their probabilities*. Theory of Probability and its Applications, 16:264--280. 1971

[WH99] Y. Wu and T. Huang. Vision-based gesture recognition: A Review. In Proceedings of the International Gesture Recognition Workshop, pages 103-115, 1999. [RJ93] Rabiner, L. and Juang, B.-H. (1993). Fundamentals of Speech Recognition, Prentice Hall, Englewood Cliffs, NJ.