Methods for Multi-touch Gesture Recognition

Daniel Wood

Abstract

This paper presents a number of different methods for gesture recognition. A gesture is a form of non-verbal communication in which the body performs visible actions in order to communicate a particular message. Gesture recognition, in this context, refers to the target of interpreting these gestures via mathematical algorithms. An analysis describing each of the algorithms advantages and disadvantages when applied to a resource-constrained mobile device in terms of recognition speed, accuracy and training time, with a certain bias towards gaming, is also presented. The most appropriate algorithm to implement in terms of these metrics is found to be Hidden Markov models. Additionally, they are easy to implement and understand. Furthermore, they require an easily managable amount of memory and disk space available on mobile devices.

1 Introduction

Gesture recognition is a pracitice that has been around since 1964, when W. Teitelman developed the first trainable gesture recognizer [Myers 1998]. Since then, gesture recognition has been used in a wide range of applications, such as commercial CAD applications since the early 1970's [Myers 1998], and more recently, sign language interpretation, hand and facial gesture recognition, and multi-touch gesture recognition [Vogler and Metaxas 2001].

With the recent rise and commercialization of gesture-recognition enabled mobile devices, the need for algorithms that minimize the computational load on the limited resources of mobile devices, whilst still remaining strong in recognition speed and accuracy, has become of great importance.

Being an expressive form of human-to-human communication, gestures allow users to perceive their bodies as an input mechanism, without having to rely on the limited input capabilities of current mobile devices [Chen et al. 2007].

This paper will provide a description of the most tried and tested gesture recognition techniques and mothodologies, as well as describe the various strengths and weaknesses of these techniques. In addition, a critical analysis of each technique based on gesture recognition speed, accuracy and training time will be conducted.

2 Gesture Recognition

When designing a gesture recognition system, careful attention must be paid to pattern representation, feature extraction and selection, classifier design and learning, training and testing, and performance evaluation [Jain et al. 2000].

The best known approaches to pattern recognition are template matching, statistical classification, syntactic or structural matching [Jain et al. 2000].

Template matching involves finding small subsections of an image, being the image of the input gesture, which matches a template image [Wang 1985].

Statistical classification involves each pattern being represented as features in an n-dimensional space, based on quantitative information on one or more characteristics, and assumes that patterns are generated by a probabilistic system [van der Walt and Barnard 2007]. It is widely used with Bayesian Networks (2.1), Hidden Markov Models (2.2), Artificial Neural Netorks (2.4) and Support Vector Machines (2.5).

In the synaptic or structural matching approach, patterns are seen as being made up of simpler sub-patterns, which are themselves built up of even simpler sub-patterns [Duda et al. 2001].

These approaches are implemented via a number of methods, described below.

2.1 Bayesian Networks

Bayesian Networks are, in essence, directed acyclic graphs (DAGs) where each node is a random variable, and specific independence assumptions hold [Charniak 1991]. The edges in the Bayesian Network are the independence assumptions that must hold between the random variables. These independence assumptions determine what probability information is required to specify the probability distribution among the random variables in the network.

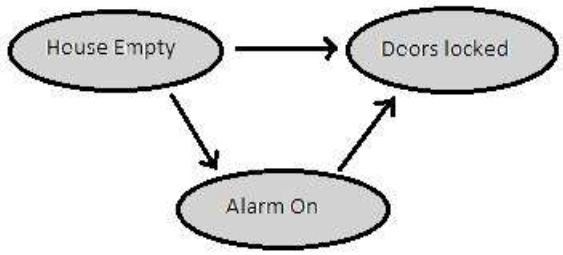

Figure 1: *A casual graph of a Bayesian Network. When the house is empty, there is a high probability that the doors are locked, and the alarm is on.*

In Figure 1, the edges denote casuality, whereas in a Bayesian Network, they specify things about the probability distribution (the identification of the probability of each value of a random variable). This is done by assigning prior probabilities to the root nodes, and conditional probabilities to non-root nodes (training the network for gestures) [Charniak 1991]. Gesture data is preprocessed before being passed into the Bayesian Network. Based on this data, the various states in the network can make probability decisions about the gesture.

An advantage of Bayesian Networks is that they continue to improve over time with the addition of new data [Kyprianidou 2003]. This means that accuracy in gesture recognition improves over time, or with additional training. However, Bayesian Networks are not practical for large implementations (or in this case, large sets of gestures), as their computations become too costly, slowing down recognition speed [Kyprianidou 2003].

2.2 Hidden Markov Models

Hidden Markov Models (HMMs) are statistical models and simplest versions of dynamic Bayesian Networks, where the system being modeled is a Markov process with an unobserved state. It

is a collection of finite states connected by transitions, much like Bayesian Networks. Each state has two probabilities: a transition probability, and an output probability distribution.

Parameters of the model are determined by training data. These trained models represent the most likely way that a human will perform a gesture, and are used to classify new incoming gestures.

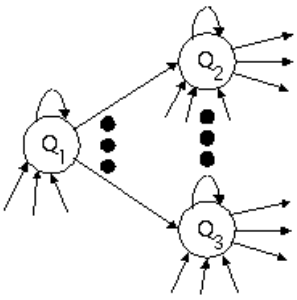

Figure 2: *A simplified representation of a Hidden Markov Model. The Q's represent the states, while the lines represent the transitions.*

An approach to using HMMs to recognize and classify gestures is described by [Yang and Yangsheng 1994]:

Meaningful gestures must first be specified, such as a vocabulary. Each gesture is then described or modeled in terms of a multi-dimensional HMM, and has a set of hidden states, as well as N dimensional observable symbols. Each HMM has a transition matrix, and N discrete output matrices. Gestures are then specified by the training data, and the HMMs are then trained on this data. This is important, as the HMMs are then adjusted in such a way that they can maximize accuracy. This trained model can now evaluate incoming gestures. Algorithms such as the Forward-Backward algorithm and Viterbis algorithms are then used to classify isolated gestures, and decode continuous gestures [Yang and Yangsheng 1994].

HMMs use only positive data and they scale well, meaning that new gestures may be added without affecting already learnt HMMs [Kadous 2002]. However, HMMs require a large amount of data in order to train [Kadous 2002], implying that initial accuracy is low. It should not be a surprise that HMM characteristics are similar to those of a Bayesian Network, in that they share advantages and disadvantages, as an HMM is simply a modified version.

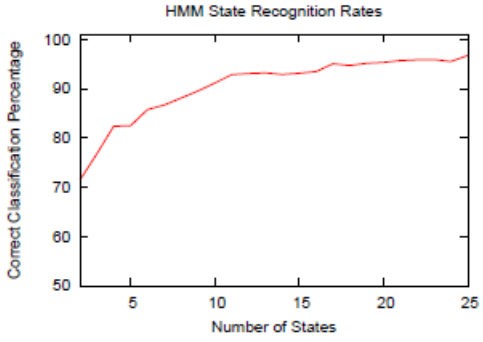

Figure 3: *The number of states in a Hidden Markov Model has a great effect on accuracy. It is, however, a toss up between accuracy and recognition speed, as can be seen in Figure 4.*

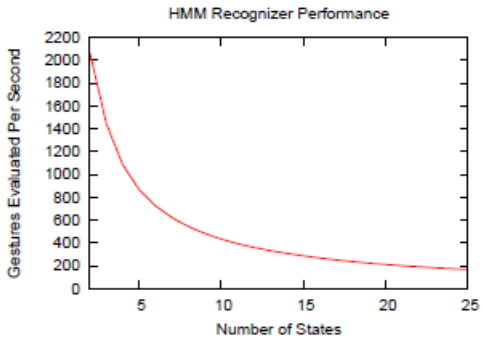

Figure 4: *The number of states effects recognition speed.*

2.3 Fuzzy Logic

There are a number of algorithms for gesture recognition based on fuzzy logic. Fuzzy logic is a form of multi-valued logic, with reasoning that is approximate rather than precise [Novak et al. 1999]. This paper will focus on Bimbers algorithm.

Bimbers algorithms works by analyzing a gesture for 56 different attributes, 28 of which are based on the position information of the gesture, and 28 of which are based on the orientation. Each of the 56 attributes has a value between 0 and 100, and provides important information about the gesture [Bimber 1999].

For eg. Consider the height-to-width ratio of a gesture. If this particular attribute has a value approaching 100, then the gesture image is elongated vertically. If the attribute has a value approaching 0, then the gesture image is elongated horizontally.

The recognition system of the algorithm is trained by allowing a user to perform a gesture, before adding the analysis of that gesture to a database. Naturally, having many representations of the same gesture increases recognition rate, as it provides a wider gap in the attribute values, increasing the chances that the same gesture will fall into this gap. Gestures being input are compared to the representations in the database, and a score is calculated for each comparison. The lower the score, the closer the match [Bimber 1999].

Bimber et al. describes the strengths of his algorithm as its ability to recognise a gesture with a minimum of information [Bimber 1999]. Consequently, the algorithm offers a high gesture recognition speed. However, inherent in Fuzzy Logic, there are many ways of interpreting fuzzy rules, combining the outputs of several fuzzy rules and de-fuzzifying the output [Russel and Norvig 2010]. This could cause decreases in recognition accuracy.

2.4 Artificial Neural Networks

Artificial Neural Networks (ANNs) are networks of weighted, directed graphs where the nodes are artificial neurons, and the directed edges are connections between them [Nasution and Khan 2008].

The most common ANN structure is the feedforward Multi-Layer Perceptron. Feedforward means that the signals only travels one way through the net [Heaton 2005]. Neurons are arranged in layers, with the outputs of each neuron in the same layer being connected to the inputs of the neurons in the layer above [Heaton 2005]. ANNs consist of three layers, namely the input layer, the hidden layer and the output layer [Dayhoff 1989]. Once a gesture has been performed, its features are broken down into parts before being passed

to different neurons of the input layer. The data is then passed to the Hidden Layer, where decisions are made (based on weightings) in order to classify the gesture; there may be more than one hidden layer. Finally, the output layer neurons are assigned a value. Each output layer neuron represents a particular class of gesture, and the record is assigned to whichever class's neuron has the highest value [Dayhoff 1989].

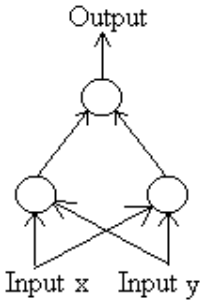

Figure 5: *A two-layer representation of an Artificial Neural Network.*

During training, the gesture class for each neuron in the output layer is known, and the nodes can be assigned the "correct" value (0.9 for the node corresponding to the correct class, and 0.1 for the others) [Dayhoff 1989]. Consequently, it is possible to compare the ANNs calculated values for these nodes to the "correct" values they have been assigned, and hence calculate an error term. These error terms can then be used to adjust the weightings in the hidden layers, and thereby training the network to better calculate the correct values [Dayhoff 1989].

Although powerful, using and implementing ANNs is not straightforward as HMMs and Baysian Networks, and a good understanding of the underlying theory is essential [Zemel et al. 1995]. Additionally, large training times are required for large sets of gestures [Drossu and Obradovic 1996]. However, once the training has been completed, the classification of gestures is quick [Drossu and Obradovic 1996].

2.5 Support Vector Machines

A Support Vector Machine (SVM) is a technique for seperating data points (or gestures, represented as data points) into particular classes. SVM models are very close to that of ANNs.

If we had a set of training examples, and each set was marked to belong to one of two categories, the SVM training method creates a model that can predict if a new example falls into the one category, or the other. The model represents the examples as a p-dimensional vector (a list of p numbers) in space, mapped so that examples from each of the separate categories are separated by as wide a gap as possible. New examples can then be mapped onto the same space and predicted to belong to a specific category depending on which side of the gap they fall on [Shawe-Taylor and Cristianini].

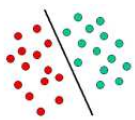

Figure 6: *A graphical representation of the division of points associated with a Support Vector Machine.*

SVMs are very robust due to inter-changable kernels [Burgress 1988]. Kernels in SVMs implicitly define the class of possible patterns by introducing an idea of similarity between data. For example, the similarity between documents by length, topic, language etc [Cristianini 2001]. However, they are very limited in both recognition speed, and gesture set size [Burgress 1988]. Additionally, they are very slow during test phases [Burgress 1988].

2.6 Dynamic Time Warping

Dynamic Time Warping (DTW), introduced by Sakoe and Chiba in 1978, is an algorithm that compares two different sequences that may possibly vary in time [Sakoe and Chiba 1978]. For example, if two video clips of different people walking a particular path were compared, the DTW algorithm would detect the similarities in the walking pattern, despite walking speed differences, accelerations or deaccelerations.

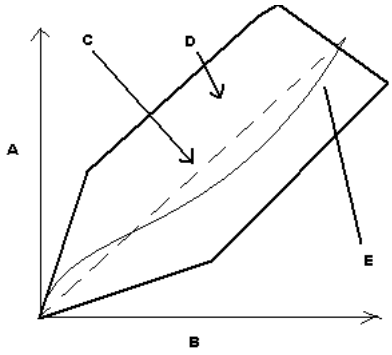

Figure 7: *B represents time interval of the input stream, and A represents the time of the template stream. C represents the linear time warp, D the dynamic timewarp search space, and E the Minimum distance mapping between the input and the template.*

The algorithm begins with a set of template streams, describing each gesture available in the system database. Taking an unlabeled input stream of templates, the minimum distance between the input, and each template stream is calculated, and the input stream is classified as whichever template stream it matches most closely [Kadous 2002]. The warping comes in with an algorithm that is used to search the space of mappings from the time sequence of the input stream to that of the template streams, so that the total distance is minimized [Kadous 2002].

DTW has several weaknesses. It is O, where N is the length of the sequence, and V is the number of templates to be considered [Kadous 2002]. This results in high computation time, and hence, limitations in recognition speed. Additionally, the storing of many templates for each gesture results in costly space usage on a resource-constrained device.

3 Critical Analysis

A critical analysis based on the results achieved by [Niezen 2008] is shown in this section. ANNs, HMMs, and DTW algorithms were implemented on a mobile phone, and measured in performance according to recognition speed, accuracy and time needed to train. Since Bayesian Networks are a superclass of HMMs which have been tweaked towards gesture classification, they are not considered. Additionally, since Support Vector Machines are based on ANN models, they are not considered either.

HMMs were implemented with both 4 states, and 8 states, but it

was found that 4 states were not able to correctly classify enough samples to accurately compare to other algorithms. Therefore, 4 states were not considered. For HMMs with 8 states, 77 of 80 samples were correctly classified after training. DTW also identified 77 samples correct, and ANN classified only 72 samples correctly [Niezen 2008]. These results are summarized below.

Algorithm	Recognition Speed	Accuracy	Training Time
HMMs	10.5ms	95.25%	Long
ANNs	23ms	90%	Medium
DTW	8ms	95.25%	No Training

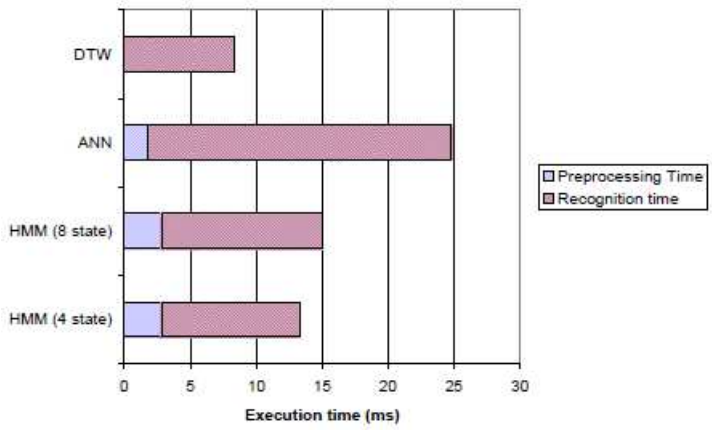

Figure 8: *A comparison of execution times between the considered algorithms.*

From the results, it is easily seen that DTW is more effiecient in both recognition speed, and training time. However, for this experiment, DTW was not fully implemented. This is discussed further in section 4.

4 Conclusion

In section 3, Artificial Neural Networks, Dynamic Time Warping and Hidden Markov Models were optimized, and tested on resource constrained devices (in this instance, cellular phones), and compared against each other in terms of accuracy, and computational performance. ANNs proved to have the slowest computation performance due to the large size of the neural network (mentioned in disadvantages). HMMs performed better, but the DTW algorithm proved to be the fastest, with comparable recognition accuracy. DTWs also did not require training, as is the case with HMMs and ANNs [Niezen 2008].

The evidence and testing does seem to point toward using Dynamic Time Warping as the preferred gesture recognition algorithm for resource constrained devices, however, in [Niezen 2008], the algorithm was not fully implemented, in that the minimum cost path obtained from the distance matrix was used as a similarity measurement, without calculating the final time warped signal. According to [Niezen 2008], this approach to the algorithm has never been applied to gesture recognition. Therefore, DTW will not be considered until a full implementation can be tested.

The optimal method for gesture recognition on a mobile device is then the use of HMMs. They have been tested with great success in many projects and implementations, perform well in recognition time and accuracy, and are easy to understand and implement. ANNs prove to be too slow for large sets of gestures, and require a lot of disk space and memory, which is not ideal for resource constrained devices. Although Bimber's algorithm is another possibility, there is not enough comparison for this against the other techniques.

References

ALDRICH, J. 1997. R.a. fisher and the making of maximum likelihood 1912-1922. *Statistical Science 12*, 3, 162–176.

AURIA, L., AND MORO, R. 2008. Support vector machines (svm) as a technique for solvency analysis. *DIW Berlin Discussion Paper No. 811*.

BAILADOR, G., ROGGEN, D., TROSTER, G., AND TRIVINO, G. 2007. Real time gesture recognition using continuous time reccurent neural networks. *Proceedings of the ICST 2nd International Conference on Body Area Networks*, 15.

BIMBER, O. 1999. Continuous 6d gesture recognition: A fuzzy-logic approach. *Proceedings of 7-th International Conference in Central Europe on Computer Graphics, Visualization and Interactive Digital Media 1*, 24–30.

BURGRESS, C. 1988. From simple associations to the building blocks of language: Modelling meaning in memory with the hal model. *Behaviour Research Methods, Instruments, and Computers 30*, 188–198.

CHARNIAK, E. 1991. Bayesian networks without tears. *AI Magazine 12*, 4, 50–63.

CHEN, Q., PETRIU, E., AND GEORGANAS, N. 2007. 3d hand tracking and motion aanalysis with a combination approach of statistical and syntactic analysis. *Proc. IEEE HAVE*, 56–61.

CHOI, E. 2005. Beatbox music phone: Gesture-based interactive mobile phone using a tri-axis accelerometer. *IEEE International Conference on Industrial Technology*, 97–102.

COLEMAN, J. 2005. Introducing speech and language processing.

CRISTIANINI, N., 2001. Support vector and kernel machines. Support-vector.net Lecture Notes.

DAYHOFF, J. 1989. Neural network architectures.

DROSSU, R., AND OBRADOVIC, Z. 1996. Rapid design of neural networks for time series prediction. *IEEE Computational Science and Engineering 3*, 2.

DUDA, R., HART, P., AND STORK, D. 2001. Pattern classification (2nd edition).

HEATON, J. 2005. Introduction to neural networks with java. Heaton Research, Inc.

JAIN, A., DUIN, R., AND MAU, J. 2000. Statistical pattern recognition: a review. *Transactions on Pattern Analysis and Machine Intelligence 22*, 1, 4–37.

KADOUS, W. 2002. *Computer Based Machine Learning Unit*. PhD thesis, University of New South Wales.

KO, M. 2005. Online context recognition in multi-sensor systems using dynamic time warping. *Proceedings of the 2005 International Conference on Intelligent Sensors, Sensor Networks and Information Processing*, 283–288.

KYPRIANIDOU, C. 2003. *Analysing Basic Genetics Using Bayesian Networks and the Impact of Genetic Testing on the Insurance Industry*. Master's thesis, Cass Business School, City University.

MARQUARDT, N., GROSS, T., AND EGLA, T. 2004. Research, technology: Bayesian networks, mobile interface, project sensation.

MYERS, B. 1998. A brief history of human-computer interaction technology. 44–54.

NASUTION, B., AND KHAN, A. 2008. A hierarchical graph neuron scheme for real time pattern recognition. *IEEE Transactions on Neural Networks 19*, 2, 212–229.

NIEZEN, G. 2008. *The Optimization of Gesture Recognition Techniques for Resource-Constrained Devices*. Master's thesis, University of Pretoria.

NOVAK, V., PERFILIEVA, I., AND MOCKOR, J. 1999. Mathematical principles of fuzzy logic.

RUSSEL, AND NORVIG, 2010. Fuzzy sets and fuzzy logic. University of Iowa Lecture Notes.

SAKOE, H., AND CHIBA, S. 1978. Dynamic programming algorithms optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing 26*, 1, 43–49.

SHAWE-TAYLOR, J., AND CRISTIANINI, N. Support vector machines and other kernel based learning methods. *IEEE Transactions on Acoustics, Speech, and Signal Processing*.

VAN DER WALT, C., AND BARNARD, E. 2007. Data characteristics that determine classifier performance. *SAIEE Africa Research Journal 98*, 3, 87–93.

VOGLER, C., AND METAXAS, D. 2001. A framework for recognizing the simultaneous aspects of american sign language. *Computer Vision and Image Understanding 81*, 3, 358–384.

WANG, C. 1985. *Edge Detection Using Template Matching (Image Processing, Threshold Logic, Analysis, Filters)*. PhD thesis, Duke University.

YANG, J., AND YANGSHENG, X. 1994. Hidden markov model for gesture recognition.

ZEMEL, R., WILLIAMS, K., AND MOZER, M. 1995. Lending direction to neural networks. *Neural Networks 7*, 565–579.