# On the Selection of a Classification Technique for the Representation and Recognition of Dynamic Gestures

Héctor H. Avilés, Wendy Aguilar, and Luis A. Pineda

Department of Computer Science
Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas
Universidad Nacional Autónoma de México
Circuito Escolar, Ciudad Universitaria, D.F.
04510 México
{haviles,weam}@turing.iimas.unam.mx, luis@leibniz.iimas.unam.mx

**Abstract.** Previous evaluations of gesture recognition techniques have been focused on classification performance, while ignoring other relevant issues such as knowledge description, feature selection, error distribution and learning performance. In this paper, we present an empirical comparison of decision trees, neural networks and hidden Markov models for visual gesture recognition following these criteria. Our results show that none of these techniques is a definitive alternative for all these issues. While neural nets and hidden Markov models show the highest recognition rates, they sacrifice clarity of its knowledge; decision trees, on the other hand, are easy to create and analyze. Moreover, error dispersion is higher with neural nets. This information could be useful to develop a general computational theory of gestures. For the experiments, a database of 9 gestures with more than 7000 samples taken from 15 people was used.

**Keywords:** Gesture recognition, hidden Markov models, neural networks, decision trees, human-machine interaction.

## 1 Introduction

Visual recognition of dynamic gestures is important for machines to naturally interact with humans. Hidden Markov models (*HMMs*) and artificial neural networks (*ANNs*) have become standard classification techniques in this problem. However, previous efforts have focused on recognition performance [1], at the expense of feature selection, learning performance, error distribution and clarity and comprehension of knowledge representation. Notwithstanding, accurate recognition is not the only factor to take into account for developing computational models and for judging knowledge representation techniques.

In this paper we present a set of experiments to evaluate ANNs, HMMs and decision trees (*DTs*) to recognize gestures following the criteria mentioned above. Decision trees have become a cornerstone in many pattern classification problems; however, it is not a commonly used technique to represent dynamic gestures. Our experimental results show that ANNs and HMMs obtained high recognition results, but sacrifice clarity, while DTs are clearer, although have a marginal decrease in performance. We also

H. Geffner et al. (Eds.): IBERAMIA 2008, LNAI 5290, pp. 412–421, 2008.

present a gesture database with more than 7000 samples taken from 15 people and executed with the person's right-arm. These gestures are oriented to instruct a mobile robot. We propose this database as a testbed for comparison of different techniques in gesture recognition, as well as in machine learning and analysis of sequential data.

Section 2 discusses various techniques proposed for gesture recognition, and previous experimental comparisons of these techniques. Section 3 briefly describes DTs, ANNs and HMMs. Section 4 describes the gesture database used in our experiments. Section 5 presents our experiments, results and analysis. Finally, in section 6 we discuss our conclusions and future work.

## 2 Related Work

Recognition of dynamic gestures can be seen as a pattern classification problem; a wide range of techniques have been used for this purpose, like temporal templates [2], comparison of body posture streams using Viterbi-based alignment [3] and dynamic time warping. However, the most succesfull and widely used techniques for gesture recognition are HMMs [4] and ANNs [1]. On the one hand, HMMs effectively represent gestural phases *via* state transition probabilities, and noisy observations by means of observation conditional probability functions. On the other hand, ANNs are successful classifiers in many pattern recognition problems, due to their capability to model complex pattern functions [5].

Some of the previous work have evaluated recognition performance of these techniques. In [6] is presented a comparison between support vector machines and feed forward neural nets for 13 mouse cursor movements executed by three people. In those experiments, ANNs slightly outperformed support vector machines by 2% of recognition rate. Corradini and Gross [1] present a comparison of three architectures based on combinations of ANNs and HMMs, and dynamic time warping. Their database is composed by 1350 samples of 6 gesture classes executed by 5 people. They obtained better recognition results by combining HMMs and radial basis fuctions networks to compute state probabilities. These authors state that due to the small training set their results do not mean that one classifier always outperform the other. Surveys on gesture recognition with comprenhensive reviews on gesture classification have been presented as well [7,8]. Other comparison efforts have focused on extensive tests of different gesture attributes using HMMs [9].

Decision trees have been frequently used to classify static gestures –or postures [10,11]. However, its most important usage is in areas such as machine learning and data mining to discover useful information in large size datasets. Despite its usefulness, DTs is not commonly selected as a classifier to recognize dynamic gestures [12].

## 3 Classification Techniques

Decision trees are frequently used in classification problems where classes can be represented by a set of features. DTs are a tree-based representation of a collection of *if-then* rules. The antecedent part of a rule is composed by *and* operations between feature values. The consequent corresponds to the desired class or target. Following the tree representation, each node represents a unique feature. Each branch emerging from the node represents one possible value of the feature. Leaves correspond to the values of the class variable. Decision trees are constructed by iterative selection of the most discriminative features [13]. Some features may not appear in the tree since they do not provide with a large enough discriminative capability *–i.e., attribute selection*. This technique allows us to represent continuous and categorical data, and its knowledge and classification process is easy to interpret and understand.

Hidden Markov models are probabilistic models that represent statistical properties of dynamic processes [15]. Dynamics is described in terms of the system states and their transitions. States are not visible directly, they are estimated only through the observations generated by the process –states are "hidden". HMMs suppose independence of the future with respect to the past given the present, and that the probabilities do not change over time. In gesture recognition, HMMs-based classifiers are frequently constructed by training a single HMM for each gesture class. The parameters of a HMM are computed through well-known training procedures such as the *Baum-Welch* algorithm. For the recognition, the probability –or *likelihood*– of an observation sequence for each HMM is calculated using the *Forward* algorithm [15]. It can assumed that the HMM with the highest probability correspond to the desired gesture class.

Artificial neural networks [5] are composed by a set of interconnected elements called artificial neurons. Neurons are usually grouped into input, hidden and output layers. When used as classifiers, a learning step consists on training neurons to be activated or inhibited given an input pattern. The activation of a neuron is propragated as input to other neurons of the network. On testing, a given example is presented to the input nodes, and the learnt responses are propagated through the network to the output nodes. The activation states of the output neurons define the class of the original input pattern. Common learning strategies include 3-layer topologies, backpropagation learning to take into account error changes, sigmoidal activation functions for non-linear neural responses, and adaptive number of hidden neurons.

## 4 Gesture Database

In this work we used a gesture database with a set of 9 dynamic gestures performed by 10 men and 5 women. Gestures were executed with the user's right-arm –See Fig. 1. The complete set of examples is composed of 7308 gestures. Every person contributed a different number of samples; however, at least 50 samples of each gesture per person were recorded [1].

Each sample is composed by the length $T$ of the gesture observation sequence –that ranges from 6 to 42 observations in the entire database– and the gesture data itself. Every observation of the sequence is composed of: i) $(x, y)$-coordinates of the upper and lower corners of the rectangle that segments the right-hand, ii) $(x, y)$-coordinates of the upper and lower corners of the rectangle that segments the user's torso, and iii) $(x, y)$-coordinates of the center of the user's face. These coarse posture data enable us to convert the information to different feature sets easily [9]. All coordinates are

[1] This database is available at:
`http://sourceforge.net/projects/visualgestures/`

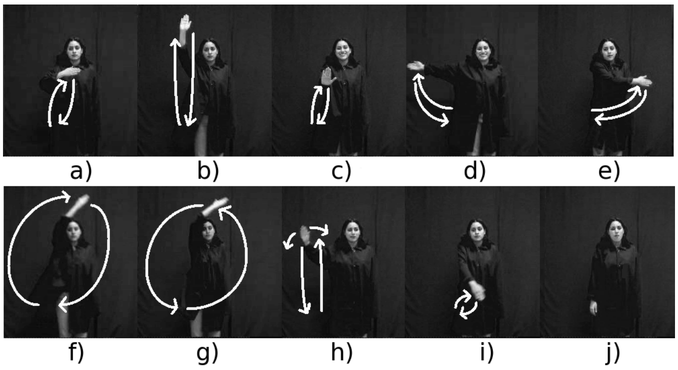


**Fig. 1.** Gesture set: a) come, b) attention, c) stop, d) right, e) left, f) turn-left, g) turn-right, h) waving-hand and i) pointing; j) initial and final position for each gesture

relative to the usual upper-left corner of the image and assuming a resolution of $640 \times 480$ pixels. Data were recorded on plain text files. Gestures were obtained using our monocular visual system[2] based on skin–color described in [16]. A spatial criterion about the position of the hand was used to start and end the capture of each gesture example. Every person executed his gestures in front of the camera at a distance of aproximately 3m. Observations were sampled every 4 images at a rate of 30 images per second.

## 5 Experiments and Results

### 5.1 Gesture Attributes

From the coarse posture information described in section 4 the following 7 gesture attributes were extracted: a) 3 features to describe motion, and b) 4 to describe posture. Motion features are $\Delta area$ –or changes in hand area–, $\Delta x$ and $\Delta y$ –or changes in hand position of the $XY$-axis of the image. The conjuction of these three attributes allows us to estimate hand motion in the Cartesian space $XYZ$. Each one of these features takes only one of three possible values: $\{+, -, 0\}$ that indicate increment, decrement or no change, depending on the area and position of the hand in a previous image of the sequence. For example, if the hand moves to the right, then $\Delta x = +$, if its motion is to the left, $\Delta x = -$ and if there is no motion in the $X$-axis, $\Delta x = 0$. Posture features named *form*, *above*, *right* and *torso* describe hand appearance and spatial relations between the hand and other body parts, such as the face and torso. Hand appearance is represented by *form*. This feature is discretized into one of three values: $(+)$ if the hand is vertical, $(-)$ if the hand is horizontal, or $(0)$ if the hand is leant to the left or right over the $XY$ plane. *right* indicates if the hand is to the right of the head, *above* if the hand is above the head, and *torso* if the hand is over the user's torso. These features take binary values, **true** or **false**, that represent if their corresponding condition is

[2] Available at the same location of the gesture database.

satisfied or not. The number of all possible combinations of these feature values is 648. In this work, every feature observation $o_t$ is a vector of 7 features in the following order $(\Delta x, \Delta y, \Delta area, form, right, above, torso)$. In this setting motion attributes requires two consecutive observations to be calculated. This way, once features are extracted the first observation is eliminated and hence the new length of the feature observation sequences is $T - 1$.

### 5.2 Data Preparation

To obtain a more compact representation of the gestures and avoid missing data, gesture samples were normalized to 5 feature observations by subsampling the sequences at equal intervals. Five is the minimum sequence length found in our feature sequences. This preprocessing step is not unusual in gesture recognition [9,1]. We recorded our gesture data in the following order: $T, o_1, o_2, o_3, o_4, o_5$, where $T$ is the original gesture length before feature extraction and normalization, and $o_1, o_2, o_3, o_4, o_5$ is the sequence of feature observations. $T$ was included as a feature for DTs and ANNs.

### 5.3 Experimental Setup

We used the J4.8 tree learning algorithm, that is a Java implementation of C4.5 [13], and the 3-layer Perceptron implemented in the WEKA machine learning toolkit [17]. The learning strategy of Weka for multilayer Perceptron include adaptation of the number of hidden neurons and error backpropagation with gradient descendent. For HMMs, a modified version implemented in [18] to consider multiple observation sequences [15] was used. Training is performed using the Baum-Welch algorithm. Experiments were carried out using a PC with an Intel Core 2 Duo 2.33Ghz processor and 2Gb of RAM. These software suites were used to enable others to reproduce our experiments. We conducted experiments to evaluate recognition results, knowledge description, feature selection, error distribution and learning capabilities of ANNs, DTs and HMMs independently for each one of the participants. We did this for two reason: i) to compare the behaviour of these models for different people, and ii) because it has been suggested the need to construct personalized recognition systems for gesture recognition [16].

For DTs, WEKA's default parameters were used and no special setup was established –e.g., confidence threshold for pruning to 0.25, and 2 as the minimum number of instances per leaf. For ANNs we set learning rate parameter to 0.6 and the number of epochs to train through to 100. HMMs were set to discrete uniform distributions for observations and transition probabilities to follow a standard linear 5-state transition topology. The EM algorithm with *Forward* and *Backward* logarithmic probabilities was used to train HMMs. Convergence criteria are: 1) $10^{-26}$, as the minimum difference between two consecutives estimations of a HMM and, 2) $10,000$ as the maximum number of iterations to train each model. However, the latter criterion was not observed to be triggered. For testing, the probability of each gesture sequence was computed using the scaled version of the *Forward* algorithm. Training parameters were selected arbitrarily; however, on various tests we observed that there is not a considerable impact of these values on the recognition performance of the classifiers. For

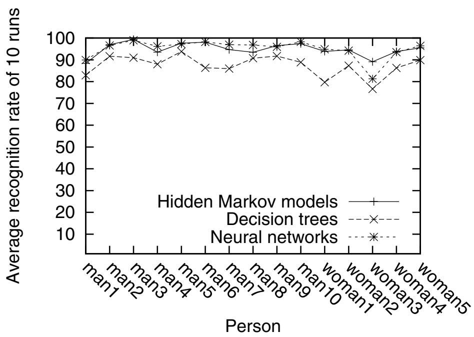


**Fig. 2.** Average recognition rates of 10 runs for each person using hidden Markov models, decision trees and neural networks

example, by modifying the minimum training threshold of HMMs to $10^{-6}$, only the number of training iterations is decreased, without affecting recognition rates significantly.

Initially, for each person, we extracted 50 samples of each gesture to construct 15 personal databases. These examples are the base of all our experiments: from these 50 examples, we selected randomly 30 samples to construct the training data set, and the remaining 20 samples for testing. The three models were trained and tested using the same examples.

### 5.4 Results

We performed 10 repetitions of the previous experiment for each person. Following this configuration, 150 DTs, 150 ANNs and 1350 HMMs –9 HMMs per classifier– were constructed. Figure 2 shows the average recognition rates obtained with these classification techniques. Average recognition rates of all the participants are $95.07\%$ for neural networks, $94.84\%$ for HMMs and $87.3\%$ for DTs. To analyze how classification erros are distributed among classes we computed cumulative confusion matries by adding the 150 individual confusion matrices of each model. Tables 1, 2, and 3 show these matrices. Rows are true classes and columns corresponds to classification results. Percentages on each row account for recognition results of 3000 classification tests. Table 4 presents the best and worst case of: i) computational time used for training and ii) the number of parameters required by the classifiers. For HMMs classifiers, the number of parameters is fixed and stands for discrete observation distributions –640 parameters– of each state, plus transition probabilities –25 parameters– and initial state distribution –5 parameters– for each one of the 9 hidden Markov models. Decision tree parameters correspond to the number of tree nodes. Parameters of neural networks are the number of connection weights between nodes required by networks with 122 and 155 hidden neurons in the best and worst cases, respectively. We conducted a visual inspection of these classifiers to give us an idea about the description capabilities of these models.

**Table 1.** Cumulative confusion matrix for recognition results of each person with HMMs. Gestures are: **C** = come, **A** = attention, **R** = right, **L** = left, **S** = stop, **T-L** = turn-left, **T-R** = turn-right, **P** = pointing, **W-H** = waving-hand. Percentage on each row corresponds to 3000 classification tests.

| | **C** | **A** | **R** | **L** | **S** | **T-L** | **T-R** | **P** | **W-H** |
|---|---|---|---|---|---|---|---|---|---|
| **C** | **95.1** | 0.7 | 0.57 | | 1.33 | | | 1.43 | 0.87 |
| **A** | 0.87 | **94.83** | | | 0.47 | 0.7 | 0.13 | | 3 |
| **R** | 0.13 | | **99.17** | | 0.17 | 0.03 | | | 0.5 |
| **L** | 0.17 | | | **93.83** | | | | 6.0 | |
| **S** | 1.93 | 0.47 | 0.07 | | **90.63** | | | 0.87 | 6.03 |
| **T-L** | 0.1 | 0.1 | 0.03 | | 0.1 | **99.47** | | 0.03 | 0.17 |
| **T-R** | | 0.17 | 0.03 | 0.03 | | | **99.67** | 0.07 | 0.03 |
| **P** | 1.6 | | 0.1 | 5.37 | 0.67 | 0.07 | | **91.87** | 0.33 |
| **W-H** | 0.47 | 3.43 | 0.2 | | 6.7 | 0.03 | 0.03 | 0.1 | **89.03** |

**Table 2.** Cumulative confusion matrix for classification results of each person using DTs

| | **C** | **A** | **R** | **L** | **S** | **T-L** | **T-R** | **P** | **W-H** |
|---|---|---|---|---|---|---|---|---|---|
| **C** | **85.73** | 0.87 | 2.97 | 0.13 | 4.1 | 0.5 | | 2.87 | 2.83 |
| **A** | 2.13 | **87.67** | 0.23 | | 0.97 | 6.4 | 0.77 | | 1.83 |
| **R** | 0.47 | 0.17 | **97.33** | | 0.6 | 1.43 | | | |
| **L** | 0.77 | | | **96.73** | | | 0.67 | 1.83 | |
| **S** | 4.13 | 1.67 | 0.57 | | **87.37** | 0.70 | 0.10 | 1.47 | 4 |
| **T-L** | 4.37 | 8.5 | 4.23 | | 0.73 | **80.7** | 1.13 | 0.13 | 0.2 |
| **T-R** | 6.73 | 0.83 | | 4.33 | | 1.23 | **85.03** | 1.60 | 0.23 |
| **P** | 6.70 | 0.03 | | 4.77 | 0.47 | 0.23 | | **86.77** | 1.03 |
| **W-H** | 5.27 | 5.03 | 1.07 | | 7.20 | 1.07 | 0.2 | 0.03 | **80.13** |

### 5.5 Analysis

Recognition rates of the proposed techniques show that there are no considerable differences between recognition rates of ANNs and HMMs. This is somewhat coincident with previous comparisons of these classifiers, althought we are using neither the same type of models, nor the same implementations. Recognition performance of DTs is below the other classifers in all cases. However, results of DTs are positive enough in accordance with their small number of parameters. Notwithstanding, more experimentation must be executed to test different criteria for tree learning, and evaluate its impact on the classification performance. An interesting case of low recognition rate is woman3. We analyzed her individual confusion matrices. Misclassifications are concentrated in attention and waving-hand gestures for the three models. By displaying these gestures we found there are fairly similar examples of these classes –*i.e.*, raising the hand around the heads top with a little hand waving.

In addition to recognition rates, we used three different measures to quantify error of these models. Results are presented on Table 5. The first one is *Shannon's entropy* [20] to evaluate error dispersion. This measure shows that dispersion for decision trees is higher in comparison to HMMs and ANNs. However, entropy is sensible to the error

**Table 3.** Cumulative confusion matrix for the recognition results of each person using ANNs

| | **C** | **A** | **R** | **L** | **S** | **T-L** | **T-R** | **P** | **W-H** |
|---|---|---|---|---|---|---|---|---|---|
| **C** | **95.77** | 0.83 | 0.23 | 0.03 | 1.83 | 0.1 | 0.07 | 1.03 | 0.1 |
| **A** | 1.1 | **91.6** | 0.83 | | 0.67 | 2.17 | 1.23 | 0.27 | 2.13 |
| **R** | 0.03 | 0.03 | **99.3** | | 0.2 | 0.1 | 0.07 | 0.13 | 0.13 |
| **L** | 0.17 | | | **96.13** | | 0.03 | 0.1 | 3.57 | |
| **S** | 1.8 | 0.6 | 0.07 | 0.37 | **93** | 0.17 | 0.07 | 0.9 | 3.03 |
| **T-H** | 0.13 | 1.27 | 0.67 | 0.03 | 0.23 | **97.43** | 0.03 | 0.07 | 0.13 |
| **T-R** | 0.03 | 0.3 | 0.03 | 0.23 | 0.03 | 0.03 | **98.57** | 0.73 | 0.03 |
| **P** | 1.37 | | 0.03 | 3.4 | 0.93 | 0.27 | 0.23 | **93.63** | 0.13 |
| **W-H** | 0.2 | 2.27 | 1.3 | 0.03 | 4.33 | 1.23 | 0.8 | 0.23 | **89.6** |

**Table 4.** Minimum and maximum number of parameters and training time (in seconds) for the three classifiers

| | **Number of parameters** | | **Training time (*Sec*)** | |
|---|---|---|---|---|
| | **Best case** | **Worst case** | **Best case** | **Worst case** |
| **Hidden Markov models** | 29,070 | | 1.03 | 849.83 |
| **Neural Networks** | 25,173 | 42,483 | 207.06 | 760.67 |
| **Decision trees** | 31 | 519 | 0.02 | 0.09 |

rate. To avoid this, we follow the method introduced by R. van Son to calculate error dispersion measures independent of the error rate, with information taken directly from the confusion matrix [19]. This method relies on entropy-based measure *perplexity* to calculate the "effective" mean number of error classes. Measures are $d_s$ and $d_r$. $d_s$ can be interpreted as the mean number of wrong responses per correct class; $d_r$ is the mean number of samples incorrectly classified on each possible response. These measures account for dispersion through the horizontal and vertical dimensions of the confusion matrix, respectively. The higher the dispersion is, the higher the value of these measures should be. Error dispersion values show that NNs generated more error dispersion in comparison to HMMs and DTs. Error dispersion is important because confusion matrices could be used by machines to decide whether a gesture has been executed or not, and classifiers with low error distribution should be preferred. Finally, we propose *Confusion Ratio*. This measure is defined as the quotient of the total error over the total correct recognition rate; the lowest the confusion ratio, the lowest the value of this measure is. Here, again the results are similar for HMMs and NNs, and better than DTs.

Clarity in knowledge representation has not been correctly valued in the past on gesture recognition. While it has been shown that ANNs and HMMs provide good recognition engines, gestural information is better described by decision trees. In particular, HMMs represent internal information as numerical data, making it difficult to assign physical meanings and make judgments without the aid of adequate graphical tools. With ANNs the situation is even worse. Decision trees represent their information in a suitable form to be readable for everyone with a little understanding of them and using only a few parameters. For example, DTs enable us to visually analyze the structure

**Table 5.** Dispersion measures for cumulative confusion matrices of HMMs, DTs and NNs

| | *Shannon's entropy* | $d_s$ | $d_r$ | *Confusion Ratio* |
|---|---|---|---|---|
| **Hidden Markov models** | 3.51 | 1.88 | 1.88 | 0.054 |
| **Decision trees** | 3.94 | 3.36 | 3.46 | 0.142 |
| **Neural Networks** | 3.54 | 3.60 | 3.62 | 0.052 |

of gestures by identifying relevant observations for each gesture class –that it is a first step for feature selection. We have seen that gestures with similar evolutions are frequently grouped together into the structure of DTs. This help us to identify in advance which gestures are similar and could be potentially misclassified before making extensive testing. In addition to descriptiveness capabilities of DTs, the low training time of these models can be important for fast prototyping when designing gestural interfaces.

## 6 Conclusions and Future Work

In this paper, an empirical comparison of decision trees, neural networks and hidden Markov models in gesture recognition has been presented. Our analysis extends previous efforts to issues not considered before such as knowledge description, feature selection, error distribution and computational time for training. We have found that there is not a single best alternative to cope with all these questions. Neural nets and hidden Markov models obtained high recognition rates in comparison to decision trees. However, knowledge description of decision trees allows us to analyze interesting information such as the similarity of gestures or relevant observations. Moreover, due to the required computational time for training, decision trees could be adequate for fast prototyping in the design of gestural interfaces. We used a gesture database with more than 7000 samples performed by 15 people.

We believe decision trees can be applied to gestural analysis beyond gesture recognition. As a future work we plan to test different configurations for the current attributes and different sets of feature vectors to evaluate its impact on recognition performance and gesture description. In addition, we plan to develop a methodology for using decision trees as a preprocessing step to automatically analyze gestures, their relevant attributes, and to identify possible confusions before testing other more complex representations.

**Acknowledgments.** The authors would like to thank Dr. Rob van Son for his valuable comments and for proofreading the document, and anonymous reviewers for their constructive suggestions.

## References

1. Corradini, A., Gross, H.M.: Implementation and Comparison of Three Achitectures for Gesture Recognition. In: IEEE International Conference on Acoustics, Speech, and Signal Processing, pp. 2361–2364 (2000)
2. Davis, J.W., Bobick, A.F.: The Representation and Recognition of Human Movement Using Temporal Templates. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, p. 928 (1997)

3. Waldherr, S.: Gesture Recognition on a Mobile Robot, Diploma's thesis Carnegie Mellon University. School of Computer Science (1998)
4. Yang, J., Xu, Y.: Hidden Markov model for Gesture Recognition, Techincal report CMU-RI-TR-94-10, Carnegie Mellon University (1994)
5. Duda, R.O., Stork, D.G.: Pattern Classification, 2nd edn. Wiley-Interscience, Chichester (2001)
6. Dadgostar, F., Sarrafzadeh, A., Fan, C., De Silva, L., Messom, C.: Modeling and Recognition of Gesture Signals in 2D Space: A comparison of NN and SVM approaches. In: 18th IEEE International Conference on Tools with Artificial Intelligence, pp. 701–704 (2006)
7. Watson, R.: A Survey of Gesture Recognition Techniques, Technical Report TCD-CS-93-11, Trinity College (1993)
8. Pavlović, V., Sharma, R., Huang, T.S.: Visual Interpretation of Hand Gestures for Human-Computer Interaction: A Review. IEEE Trans. on Patt. Anal. and Mach. Intell. 19(7), 677–695 (1997)
9. Campbell, L.W., Becker, A.D., Azarbayejani, A., Bobick, A.F., Pentland, A.: Invariant features for 3-D Gesture Recognition, Techinical report 379, M.I.T. Media Laboratory Perceptual Computing Section (1996)
10. Spiegel, D.: A Gesture Recognition System (1997), `http://alumni.media.mit.edu/~spiegel/papers/gesture.pdf`
11. Hai, W., Sutherland, A.: Dynamic gesture recognition using PCA with multi-scale theory and HMM. In: Proc. SPIE, vol. 4550, pp. 132–139 (2001)
12. Mardia, K.V., Ghali, N.M., Hainsworth, T.J., Howes, M., Sheehy, N.: Techniques for online gesture recognition on workstations. Image Vision Comput. 11(5), 283–294 (1993)
13. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann, San Francisco (1993)
14. McNeill, D.: Hand and Mind: What gestures reveal about thought. University of Chicago Press, Chicago (1992)
15. Rabiner, L.R.: Readings in Speech Recognition, A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition, pp. 257–286. Morgan Kaufmann Publishers, San Francisco (1990)
16. Avilés, H.H., Sucar, L.E., Mendoza, E.: Visual Recognition of Gestures. In: 18th International Conference on Pattern Recognition, pp. 1100–1103 (2006)
17. Witten, I.H., Frank, E.: Data Mining: Practical machine learning tools and techniques, 2nd edn. Morgan Kaufmann, San Francisco (2005)
18. Kanungo, T.: Hidden Markov Models Software (Last retrieved May 26, 2008), `http://www.kanungo.com/`
19. van son, R.J.J.H.: The Relation Between the Error Distribution and the Error Rate in Indentification Experiments. Proceedings 19, Institute of Phonetic Sciences, University of Amsterdam, pp. 71–82 (1995) (Last retrieved May 26, 2008), `http://fonsg3.hum.uva.nl/Proceedings/Proceedings_19/ErrorDispersion_RvS/ErrorDispersion.html`
20. Shannon, C.E.: A Mathematical Theory of Communication. Bell System Technical Journal 27, 379–423, 623–656 (1948)