Model Structure Selection & Training Algorithms for an HMM Gesture Recognition System

Nianjun Liu, Brian C. Lovell, Peter J. Kootsookos, and Richard I.A. Davis
Intelligent Real-Time Imaging and Sensing (IRIS) Group
School of Information Technology and Electrical Engineering
The University of Queensland, Brisbane, Australia 4072.
Email: {nianjunl, lovell, kootsoop, riadavis}@itee.uq.edu.au

Abstract

Hidden Markov models using the Fully-Connected, Left-Right and Left-Right Banded model structures are applied to the problem of alphabetical letter gesture recognition. We examine the effect of training techniques, in particular the Baum-Welch and Viterbi Path Counting techniques, on each of the model structures. We show that recognition rates improve when moving from a Fully-Connected model to a Left-Right model and a Left-Right Banded 'staircase' model with peak recognition rates of 84.8%, 92.31% and 97.31% respectively. The Left-Right Banded model in conjunction with the Viterbi Path Counting present the best performance. Direct calculation of model parameters from analysis of the physical system was also tested, yielding a peak recognition rate of 92%, but the simplicity and efficiency of this approach is of interest.

1 Introduction

Due to the widespread popularity of Hidden Markov Models in speech recognition [1] and handwriting recognition [2], HMMs have begun to be applied in spatio-temporal pattern recognition and computer vision [3, 4]. Their popularity is largely due to their ability to learn model parameters from observation sequence through Baum Welch and other re-estimation procedures. It is of great interest to find improved methods for training these models on multiple observation sequences, with a view to using the trained models for pattern classification purpose such as gesture recognition and other computer vision problems. This paper describes a Hidden Markov Model (HMM) based framework for hand gesture detection and recognition and investigates alternative models and learning strategies.

Human-Machine interfaces play a role of growing importance as computer technology continues to evolve. A large number of potential applications in advanced gesture interfaces for Human Computer Interaction (HCI) have been designed in the last decade [5, 6, 7, 9]. The goal of gesture interpretation is to push the envelope of advanced human-machine communication to bring the performance of human-machine interaction closer to human-human interaction. As example of the first generation pattern recognition application is the input of the Palm and Pocket PC PDAs, where keyboards have been replaced by handwriting recognition [8]. Previous attempts [5, 7] to develop a hand gesture recognition system include geometric feature-based methods, template-based methods, and more recently active contour and active statistical models.

A gesture is a spatio-temporal pattern. It is reasonable to assume that the temporal length of a gesture will vary amongst users. Thus we use the Hidden Markov Model (in a probabilistic approach incorporating time variability), to characterize a gesture in the recognition system. We define the trajectory of a letter gesture as a sequence of directional angles which are the observation symbols. Each letter is mapped to a different hidden Markov model.

The next section gives a brief introduction to the letter gesture input system we designed for HMM training and testing. Section 3 gives the basic theory of HMM Model structure and two training algorithms. Section 4 and 5 give the experimental results and analysis. The direct computation method is presented in section 6. The paper ends with a summary of the main findings.

2 Letter Gesture Input System

We designed a letter gesture input system (figure 2) to classify HMM model training methods. Its implementation details were presented in previous work [12] by the author.The system acquires 25 frame of video data and uses skin color segmentation to locate the hand. Various pre-processing steps are used to obtain hand trajectories. Figure 1 illustrates the letter V, X, P and S and shows how the trajectories were recorded. Along each trajectory, the orientation of each of the 25 hand movements is computed and quan-

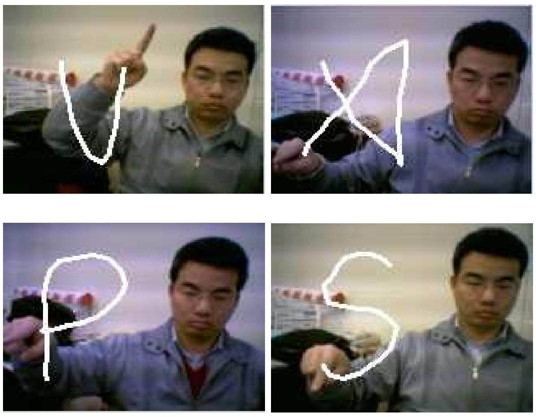


Figure 1: Single Hand Gestures for V, X, P and S

tized to one of 18 discrete symbols. This discrete observation sequence is fed into an HMM classification module for training and testing. Baum-Welch [1] and Viterbi Path Counting algorithms [10] are used to train the HMMs over a range of topologies including Fully Connected (FC), Left-Right (LR), and Left-Right Banded (LRB) with the number of states varying from 3 to 14. Our system recognizes all 26 letters from A to Z and the database contains 30 example videos of each letter gesture, including 20 training samples and 10 test samples, so there are a total of 760 gesture videos in the database.

One motivation for the development of this system was to provide an alternate text input mechanism for camera enabled handhold devices, such as video mobile phones and PDAs.

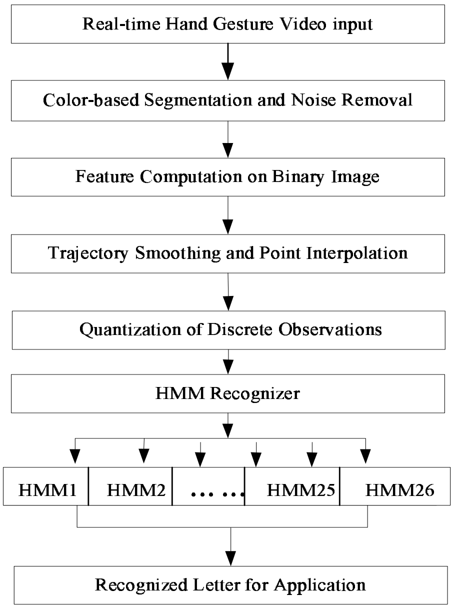

Real-time Hand Gesture Video input
Color-based Segmentation and Noise Removal
Feature Computation on Binary Image
Trajectory Smoothing and Point Interpolation
Quantization of Discrete Observations
HMM Recognizer
HMM1
HMM2
……
HMM25
HMM26
Recognized Letter for Application

Figure 2: System Flow Chart

3 Hidden Markov Model Structure and Training Methods

A Hidden Markov Model consists of N states, each of which is associated with a set of M possible observations. It includes the initial parameter π, a transition matrix A and an observation matrix B, being denoted as $\lambda = (A, B, \pi)$.

There are two basic types of model structures which are shown in figure 3. In a Fully Connected HMM, every state of the model can be reached from every other state of the model. The defining property of Left-Right HMMs is that no transitions are allowed to states whose indices are lower than the current state.

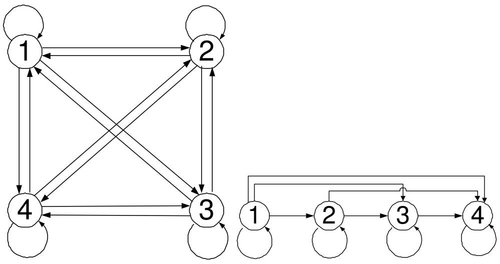


Figure 3: Fully connected and Left-Right structures

3.1 Baum Welch Algorithm

In our system, each gesture has multiple observation sequences (K), so we use a multi-sequence training algorithm proposed by Rabiner and Juang [1], which uses the K observation sequences at each stage of the Baum Welch re-estimation to iteratively update a single HMM parameter set. The re-estimation formulae for this type of iterative method are as follows:

$$\bar{a}_{ij} = \frac{\sum_k W_k \sum_{t=1}^{T_k} \alpha_i^k a_{ij} b_j(O_{t+1}^{(k)}) \beta_{t+1}^{(k)}(j)}{\sum_k W_k \sum_{t=1}^{T_k} \alpha_t^k(i) \beta_t^k(i)} \tag{1}$$

$$\bar{b}_{ij} = \frac{\sum_k W_k \sum_{O_t(k)=v_j} \alpha_t^k(i) \beta_t^k(i)}{\sum_k W_k \sum_{t=1}^{T_k} \alpha_t^k(i) \beta_t^k(i)} \tag{2}$$

where $W_k = 1/P_k, k \in [1, K]$ is the inverse of the probability of the current model estimate generating training sequence k, evaluated using the forward algorithm. $O_t^{(k)}$ is the observation symbol at time t emitted in sequence k. The forward and backward algorithms define $\alpha_t^k(i)$ and $\beta_t^k(i)$ for the sequence k, time t and state i respectively. The Baum Welch algorithm is an "iterative update" algorithm which re-estimates parameters of a given Hidden Markov Model to produce a new model which has a higher probability of generating the given observation sequence. This re-estimation procedure is continued until no more significant improvement in probability can be obtained and the local maximum is thus found. Therefore, the training results are highly dependent on the initial model.

3.2 Viterbi Path Counting Algorithm

Viterbi Path Counting training method is proposed by Davis and lovell [10], which is a fixed-structure variant of Stolcke and Omohundro Best-First Model Merging algorithm [11]. The characteristic of this method is to use the Viterbi algorithm to find the most likely path for a given sequence, and to modify the model parameters along that path by maintaining matrices of integers (π_c, A_c, B_c) corresponding to Viterbi Path statistics for π, transitions A and symbol emissions B respectively. It was thought that this would provide a simple and reliable way of maximizing the correspondence between a new sequence and the existing model structure, thus achieving good learning for both single and multiple sequence HMM training. This hypothesis was supported in synthetic trials [10].

4 Fully-Connected and Left-Right Results

We tested the Baum-Welch (BW) [1] and Viterbi Path Counting (VPC) [10] algorithms for HMM training on the fully connected (FC) and Left-Right (LR) HMM topologies, with the number of states ranging from 3 to 14. After extracting the observation sequence from the input gesture video, we calculate the probability of the observation sequence for all 26 HMMs corresponding to the letters. We output the letter corresponding to the HMM with highest likelihood.

State Num.	BW-FC%	BW-LR%	Improvement%
3	66.54	92.31	77.01
4	80.00	84.80	24.00
5	75.20	81.20	24.19
6	75.60	84.80	37.70
7	77.60	86.40	39.29
8	76.80	86.00	39.66
9	77.60	85.60	35.71
10	76.00	81.60	23.33
11	65.20	86.80	62.07
12	74.80	86.80	47.62
13	84.80	84.00	-5.26
14	72.80	81.60	32.35
Mean	75.24	85.16	40.05
Max	84.80	92.31	49.39

Figure 4: Baum Welch recognition FC/LR rates

State Num.	VPC-FC%	VPC-LR%	Improvement%
3	63.85	91.15	75.53
4	53.20	91.20	81.20
5	59.60	91.20	78.22
6	55.20	90.40	78.57
7	45.60	91.20	83.82
8	44.40	90.40	82.73
9	49.20	90.40	81.10
10	43.20	90.00	82.39
11	42.80	90.00	82.52
12	40.80	90.00	83.11
13	39.60	90.00	83.44
14	38.80	90.40	84.31
Mean	48.02	90.53	81.78
Max	63.85	91.20	75.66

Figure 5: VPC recognition FC/LR rates

Figures 4 and 5 present the recognition results on FC and LR structures for both training methods (BW and VPC) with the number of states ranging from 3 to 14. Here, the improvement is defined as the reduction percentage of error recognition rate. If the recognition ratio in the second column and third column is $C1$ and $C2$ separately, the improvement is $((1 - C1) - (1 - C2))/(1 - C1)$. From the experiment results, we can draw the following observations. For the Baum-Welch algorithm, LR structure, 3 states, achieved the best overall accuracy of 92.31%. For the VPC algorithm, the highest accuracy of 91.20% also occurred with the LR topology and 4, 5 and 7 states. The average overall recognition performance of LR topologies is always better than FC. For the average rates from 3 to 14, Baum-Welch LR achieves 85.16%, while FC is 75.24%, and for VPC- LR achieves 90.53% while FC is 48.02%. The recognition accuracy of LR HMMs trained with the Baum Welch algorithm was dependent on the number of states and accuracies ranged between 92.31 and 81.20 percent. HMMs trained with the VPC algorithm were much less sensitive to the number of states, and accuracies ranged between 90 and 91.20 percent.

VPC seems better-suited to more restrictive models in those cases, as there is less emphasis on structure learning and more emphasis on gaining accurate estimates of statistics. This suggests that separating the problems of structure learning and statistics collection may make it easier to obtain better quality models.

5 Left-Right Banded Experiment

From analyzing the results in the previous section, we found that the LR structure always performed better than the FC structure when using either Baum Welch or Viterbi Path

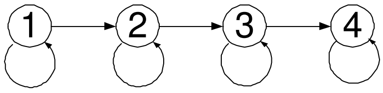


Figure 6: Left-right Banded Structure

Counting. It was therefore of interest to investigate another special model structure which we called the Left-Right Banded model (LRB). The structure is shown in figure 6. A Left-Right Banded model has a transition structure consisting of a single linear chain containing only self-transitions and transitions from any elements to the following element in the linear chain only.

From figures 7 and 8, we note that the LRB structure showed very encouraging performance levels, better than Left-Right for both Baum Welch and Viterbi Path Counting. VPC-LRB outperformed BW-LRB suggesting that VPC is the superior training method for this application. The average recognition rate of Baum Welch on Left-Right models is 85.16%, while Left-Right Banded reached 89.30%, and the improvement was 27.87%. The VPC-LR average ratio from state 3 to state 14 is 90.53%, and VPC-LR Banded obtained 94.33%, and its improvement was 40.10%. The maximum recognition rate was VPC-LRB which reached the peak value of 97.31%, an improvement of 69.41% on VPC-LR. Meanwhile, the maximum rate achieved by BW-LRB is 96.15%, an improvement of 50.00% over BW-LR. However BW was not nearly as consistently good as VPC over all numbers of states.

It seems that Left-Right Banded models perform better than Left-Right models in these trials because they make it easier for the training algorithm to extract the most information out of the data. Fully connected models tend to lose structure information because there is too much freedom for the algorithm to manage, whilst restricting the model to Left-Right form means that there is a greater chance that the training algorithm will be able to match the data to the model in a useful way.

For this data set, it appears that Left-Right Banded models are the simplest models which still enable the algorithm to distinguish well between gestures. Part of the reason may be that the quantisation scheme involves a single sequence of symbols taken from a continuous hand motion, which is therefore naturally modelled by a single Left-Right Banded model. This is in agreement with the comparison between Left-Right and Left-Right Banded models which also found that VPC produced better results on more restrictive models, but worse results on less restrictive models.

This is further evidence supporting the hypothesis that separating the tasks of learning structure and learning statistics enables better quality results to be produced. The VPC algorithm is a good algorithm for learning statistics from multiple training observation sequences given a particular model but the task of learning structure is best left to other methods, such as Best-First Model Merging or construction by hand, for example.

State Num	BW-LR%	BW-LRB%	Improvement%
3	92.31	96.15	50.00
4	84.80	85.38	3.85
5	81.20	90.77	50.90
6	84.80	85.77	6.38
7	86.40	89.62	23.64
8	86.00	89.62	25.82
9	85.60	90.00	30.56
10	81.60	88.46	37.29
11	86.80	89.23	18.41
12	86.80	88.08	9.67
13	84.00	90.00	37.50
14	81.60	88.46	37.29
Mean	85.16	89.30	27.87
Max	92.31	96.15	50.00

Figure 7: Baum-Welch recognition LR/LRB rates

State Num	VPC-LR%	VPC-LRB%	Improvement%
3	91.15	93.08	21.74
4	91.20	90.38	-9.27
5	91.20	95.00	43.18
6	90.40	93.85	35.90
7	91.20	94.23	34.44
8	90.40	94.23	34.90
9	90.40	94.62	43.91
10	90.00	95.00	50.00
11	90.00	95.00	50.00
12	90.00	95.77	57.69
13	90.00	97.31	73.08
14	90.40	93.46	31.89
Mean	90.53	94.33	40.10
Max	91.20	97.31	69.41

Figure 8: VPC recognition LR/LRB rates

6 Direct Computation Method

The defining feature of the Left-Right Banded model is that it allows the state to stay in one position for a while and then jump to the next one, and so on until finally reaching the last state, always preserving state occupation order. The expected number of observations (duration time in a state, conditioned on starting in that state) is calculated from the equation below:

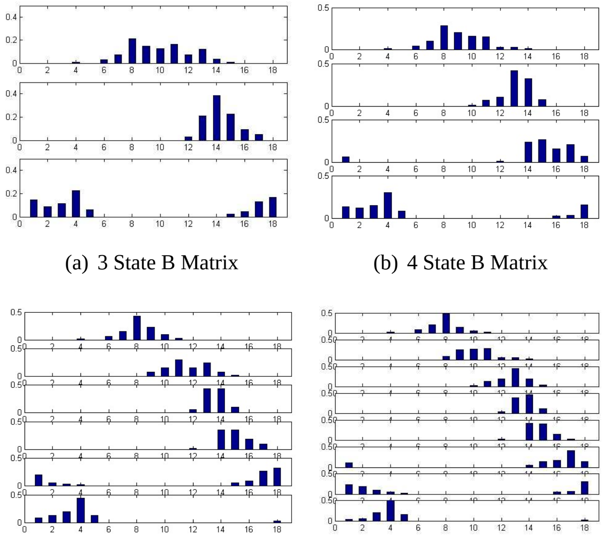
$$\bar{d}_i = \frac{1}{1 - a_{ii}} \tag{3}$$

Next it is natural to consider segmenting the observation sequence (T) evenly by the number of states, so each section is may be roughly considered as one state. Jumping from one section to the next corresponds to moving from one state to the next state. The final section corresponds to the duration of the last state. Therefore, we could use the duration time to compute the A matrix directly using the above equation. For example, if 3 states are used, with an observation sequence length of 24, then the duration time is $24/3 = 8$, and if $d = 8$, then $A_{ii} = 0.875$, because the row sum is 1, then the other value is 0.125, $A_{33} = 1$; so the A matrix is calculated. The A matrices of the other number states are computed in the same way. The example of A matrix in 4 states is shown and figure 9 shows the elements of the A matrix of 3, 4, 6 and 8 states.

$$\mathbf{A} = \begin{bmatrix} 0.83 & 0.17 & 0 & 0 \\ 0 & 0.83 & 0.17 & 0 \\ 0 & 0 & 0.83 & 0.17 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

States Num	a_{ii}	$a_{i,i+1}$	i
3	0.875	0.125	1 to 2
4	0.83	0.17	1 to 3
6	0.75	0.25	1 to 5
8	0.67	0.33	1 to 7

Figure 9: A Matrix elements in 3,4,6,8 States



(a) 3 State B Matrix

(b) 4 State B Matrix

(c) 6 State B Matrix

(d) 8 State B Matrix

Figure 10: B Matrix by 3,4,6,8 States

After we obtain the A matrix, to calculate the B matrix directly we use the idea that since each segmentation is one state, we consider which correspondings to one observation symbol. However, in a real system, a great deal of noise can be present. There are many probability distribution methods available to solve the problem, including Histogram, Gaussian Distributions, and the von Mises Distribution. In this paper, we only use the simple histogram distribution, and some other methods will be investigated in the future new paper. Because there are 18 observation symbols after quantizing the orientation angle of the trajectory in the letter gesture system and the training set is 20 observation sequences ($T \times 20$), we segment the training set by the number of states (N). Each segmentation is $T/N \times 20$ which is related to its state. Next we use the histogram to plot the distribution. For example, if there are 3 states then state 1 corresponds to the first 8×20, state 2 to the second 8×20, and state 3 to the third 8×20. Plotting them separately, we can treat the distribution directly as the observation probability matrix (B matrix). Figure 10 shows the observation probability B matrix in 3, 4, 6, and 8 states.

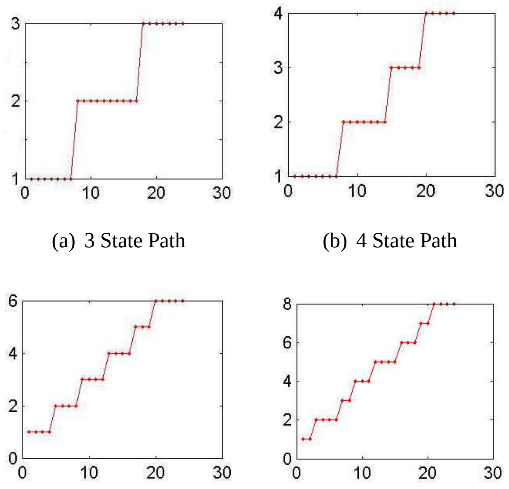
Number of States	Duration Time	Recognition rate
3	T/3	91.60
4	T/4	92
6	T/6	91.60
8	T/8	90.40

Figure 11: Direct Computation Method Results

After computing the A and B matrix, then since the model is LR-Banded, the initial probability can always be obtained from state 1, so we use the test set of the letter gesture system to estimate its performance. The result is shown in figure 11. In order to justify the state jumping processing, we calculate the Viterbi path of 3, 4, 6, and 8 states. The results are similar to our previous methods but do not achieve the performance of VPC.

7 Conclusion

The paper presents the Hidden Markov Models performance by varying the training algorithms, model structures and the number of states on the 26 Letter Gesture Recognition task. The Left-Right Banded model with Viterbi Path Counting training yielded the best overall recognition performance of about 97%, which is better than Left-Right and Full Connected models on both training algorithms. This suggests that banded models work well for gesture recognition because of their simplicity and also demonstrates that the Viterbi Path Counting is the most reliable algorithm for training HMMs for the letter recognition system. On the basis of the Left-Right Banded results and the state duration equation, we explored the direct computation method using the even state duration and the histogram distribution, and



(a) 3 State Path (b) 4 State Path

(c) 6 State Path (d) 8 State Path

Figure 12: 3,4,6,8 States Path

its performance is quite encouraging.

References

[1] L.R. Rabiner, and B.H. Juang, *Fundamentals of Speech Recognition*, New Jersy Prentice Hall,1993.

[2] J.J. Lee, and J.H. Kim, “Data-Driven Design of HMM Topology for online Handwriting Recognition”, *International Journal of Pattern Recognition and Artifical Intelligence*, *Volume* 15, Number 1(2001) pp.107-121. World Scientific Publishing Company.

[3] Andrew D. Wilson, Aaron F. Bobick, “Hidden Markov Models for Modeling and Recognizing Gesture under Variation”. International *Journal of Pattern Recognition and Artificial Intelligence (IJPRAI)*, Volume 15, Number 1, February 2001,pp.123-160.

[4] A. Nefian and M.Hayes, “Hidden Markov Models for Face Recognition”, *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP98)*, PP.2721-2724,Seattle, United States of America,1998.

[5] V. Pavlovic, “Visual Interpretation of Hand Gestures for Human-Computer Interaction”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, July 1997 (Vol. 19, No. 7) pp. 677-695

[6] Hyeon-Kyu Lee and Jin H. Kim, “An HMM-Based Threshold Model Approach for Gesture Recognition”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, October 1999 (Vol. 21, No. 10). pp. 961-973

[7] J. Yang, Y. Xu, and C.S. Chen, “Gesture Interface: Modeling and Learning,” *IEEE International Conference on Robotics and Automation*, Vol. 2, 1994, pp. 1747-1752.

[8] Palm Products-Ways to Enter Data into a Palm Handheld, Aug,2003, Available at (online): http://www.palm.com/us/products/input/

[9] D.Heckenberg and B. C. Lovell, “MIME: A Gesture-Driven Computer Interface”, *Proceedings of Visual Communications and Image. SPIE*, V 4067, pp 261-268, Perth 20-23 June, 2000.

[10] R. I. A. Davis and B. C. Lovell, “Comparing and Evaluating HMM Ensemble Training Algorithms Using Train and Test and Condition Number Criteria.” *To Appear in the Journal of Pattern Analysis and Applications*, 2003.

[11] A. Stolcke and S. Omohundro. ”Hidden Markov Model induction by Bayesian model merging”, *Advances in Neural Information Processing Systems*, pp. 11-18, Morgan Kaufmann, San Mateo, United States of America, CA1993.

[12] N. Liu, and B.C. Lovell, and P.J. Kootsookos. “Evaluation of HMM training algorithms for Letter Hand Gesture Recognition”, *IEEE International Symposium on Signal Processing and Information Technology*, December 2003.

Appendix:26 Letter Gestures

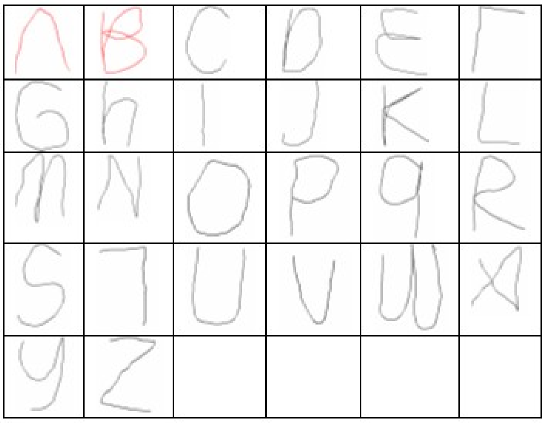


Figure 13: Samples of the 26 letter gestures used in the trials (traced onto a single bitmap)