

Fig. 2. Cameraman image (BSNR = 40 dB)  $\begin{bmatrix} a & b & c \\ d & e & f \end{bmatrix}$ : Top (left to right)—Original; blurred; fullband. Bottom (left to right)—Subband (LL); subband (LSI); subband (LSV).

TABLE I  
TWO DIFFERENT INITIAL CONDITIONS

type	A		B	
	subband	fullband	subband	fullband
AR Model	-0.09 .3	-9 .3	-36 .6	-36 .6
	.3 x	.3 x	.6 x	.6 x
Qv	700	400	350	200
Qw	650	800	650	800

TABLE II  
SNR IMPROVEMENT IN DIFFERENT CONDITIONS

type	subband		fullband	
	7 × 7	9 × 9	11 × 11	13 × 13
A	8.2 dB	8.5 dB	6.4 dB	-12.0 dB
B	10.1 dB	8.6 dB	7.9 dB	-12.5 dB

TABLE III  
SNR IMPROVEMENT OF DIFFERENT QMF'S

16A	24B	32C	48D
4.2 dB	7.3 dB	9.0 dB	9.5 dB

TABLE IV  
SNR IMPROVEMENT AT DIFFERENT BSNR'S

BSNR	Subband			Full band
	LL	Const.	Adap.	
20 dB	2.2 dB			2.3 dB
40 dB	3.9 dB	6.2 dB	7.3 dB	5.8 dB
60 dB	3.9 dB	9.2 dB	11.1 dB	8.8 dB

Table IV shows the numerical restoration results of a blurred image that has different noise levels. In the subband EM approach, all subbands (LL, HL, LH, HH) are restored and synthesized into one final image in 60 dB BSNR, three subbands (LL, HL, LH) are restored in 40 dB BSNR, and only one subband (LL) in 20 dB BSNR. In this experiment, we used the better type B initial condition since the restoration is sensitive to this. For this good initial condition, the fullband EM gives a better result for psf estimation, but the subband EM still gives a better restored image due to the adaptation in image modeling. In the subband EM image estimate, horizontal and vertical edges are restored well, but 45 and 135° edges are not well restored. This is shown in Fig. 2. This is due to the fact that the image is divided in the vertical and horizontal directions, and the HH subband was assumed to be white.

## VI. CONCLUSION

The subband EM method is less sensitive to initial conditions than fullband EM and permits a kind of parallel execution. In the upper frequency subbands, an inhomogeneous space-variant image model can be introduced to give better results. However, the subband analysis/synthesis filters must be quite good, i.e., near to ideal, to avoid distortion caused by transition band effects. Since many psf's are of the circular type, the application of a hexagonal subband method should be investigated.

## REFERENCES

- [1] R. L. Lagendijk, J. Biemond, and D. E. Boeke, "Identification and restoration of noisy blurred images using the expectation-maximization algorithm," *IEEE Trans. Acoust. Speech Signal Processing*, vol. 38, pp. 1180-1191, July 1990.
- [2] K. T. Lay and A. K. Katsaggelos, "Image identification and restoration based on the expectation-maximization algorithm," *J. Opt. Eng.*, vol. 29, pp. 436-445, May 1990.
- [3] F. C. Jeng and J. W. Woods, "Inhomogeneous Gaussian image models for estimation and restoration," *IEEE Trans. Acoust. Speech Signal Processing*, vol. 36, pp. 1305-1312, Aug. 1988.
- [4] F. C. Wagner, A. Macovski, and D. G. Nishimura, "A characterization of the scatter point-spread function in terms of air gaps," *IEEE Trans. Med. Imaging*, vol. 7, pp. 337-344, Apr. 1988.
- [5] J. W. Woods and S. D. O'Neil, "Sub-band coding of images," *IEEE Trans. Acoust. Speech Signal Processing*, vol. ASSP-34, pp. 1278-1288, Oct. 1986.
- [6] J. D. Johnston, "A filter family designed for use in quadrature mirror filter banks," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, pp. 291-284, Apr. 1980.

## Hidden Markov Models Applied to On-Line Handwritten Isolated Character Recognition

Stephan R. Veltman and Ramjee Prasad, *Senior Member, IEEE*

**Abstract**—Hidden Markov models are used to model the generation of handwritten, isolated characters. Models are trained on examples using the Baum-Welch optimization routine. Then, given the models for the alphabet, unknown characters can be classified using maximum-likelihood classification. Experiments have been conducted, and an average error rate of 6.9% was achieved over the alphabet consisting of the lowercase English alphabet.

## I. INTRODUCTION

With the introduction of integrated systems digital network, multimedia applications are gaining importance. Due to the integration of several services into one single terminal, the user interface, or the man-machine part of the system, becomes a very important aspect of the system. An interesting step in the evolution of the user interface was the introduction of the pen and digitizing tablet. Since a pen enables the input of handwritten script, machine interpretation of handwritten text becomes an important field of research. If a machine

Manuscript received June 10, 1992; revised June 3, 1993. The associate editor coordinating the review of this paper and approving it for publication was Dr. Michael Unser.

The authors are with Delft University of Technology, Telecommunications and Traffic Control Systems Group, Delft, The Netherlands.

IEEE Log Number 9216571.

can reliably interpret handwritten script, applications lie in a wide variety of systems, mainly those who require a high interactivity or the use of direct pointing and manipulation [7].

This paper focuses on character recognition. The performance of a system based on hidden Markov models (HMM's) is evaluated. HMM's have been successfully applied to automatic speech recognition [4], [5], [13] and to some extent also to character recognition [1], [3]. In [1] and [2], an HMM was used to model specific linguistic information, while [3] was a word-level approach. This paper focuses on recognition of isolated characters. Characters are input in a freely, unconstrained way using a tablet digitizer. HMM's for each character are obtained by training on example characters using optimization routines. During classification unknown characters are "scored" on each model, after which the model that has the largest likelihood of having produced the unknown character is chosen (ML- or Maximum-Likelihood classification). The training procedure is computationally very intensive, but can be done off-line, while the classification procedure can be done in real-time.

## II. HIDDEN MARKOV MODELS APPLIED TO THE MATCHING PROBLEM

Basically, an HMM is defined as a doubly stochastic process with an underlying stochastic process that is not observable, i.e., hidden, but can only be observed through another set of stochastic processes that produces observable symbols [2]. Here we present a short overview of the theory of application of HMM's to the matching problem according to [4] and [5]. At each time instant  $t$ , the HMM occupies a state  $s_t$  and emits an observable symbol  $O_t$ . Then it moves to the next state on the basis of state transition probabilities, contained in the **A** matrix. The emission of symbols from each state occurs on the basis of symbol emission probabilities, contained in the **B** matrix. To set the whole process in motion, only one more parameter is needed: the initial state probability vector  $\pi$ . For each character, an HMM is formed by training on example characters, so two problems can be distinguished: the classification and the training problem.

The classification problem is the easiest of the two. For each character, a model exists. An unknown observation sequence is "scored" on each model, and the character corresponding to the model that yields the highest probability of having emitted the sequence is selected. This type of classification requires computation of the probability that the sequence **O** was emitted by a given model  $M$ ,  $\Pr(O|M)$ . Since the state sequence that has produced an observation sequence is "hidden," i.e., not observable, we can either calculate  $\Pr(O|M)$  over all possible state sequences or over the most likely state sequence having emitted the observation sequence. In the present study, we have used  $\Pr(O|M)$  over all possible state sequences, which can be efficiently calculated using the forward-backward algorithm.

The training problem is typically a constrained optimization problem. The likelihood that the example characters are generated by the given model is optimized. This can be done by classical optimization methods, like with Lagrange multipliers, but here we have only considered the Baum-Welch algorithm. First, an initial estimate of the model is made. Next, the parameters of **A**, **B**, and  $\pi$  are reestimated using the Baum-Welch reestimation formulas. The model parameters are replaced by these reestimations, and this procedure is repeated until the increase in  $\Pr(O|M)$  is arbitrarily small. The Baum-Welch algorithm is guaranteed to increase  $\Pr(O|M)$  until a critical point, from which  $\Pr(O|M)$  no longer changes. A proof of this property can be found in [4]. The optimum value for  $\Pr(O|M)$  is typically a local optimum. The optimum found depends on the initial estimates of the reestimation algorithm. It may be necessary to optimize  $\Pr(O|M)$  multiple times under different initial estimates to obtain an optimum near the global optimum of the likelihood function.

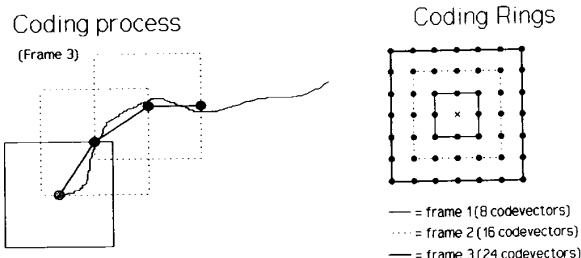


Fig. 1. Illustration of the coding process.

## III. TABLET QUANTIZATION

Since the character representation depends on the digitizing method, we first discuss the writing tablet. The writing tablet has a writable surface of  $21 \times 15$  cm and a resolution of  $2100 \times 1536$  grid lines on this surface. So, the resolution requirements according to [7] are sufficiently met. The sampling process involves the following steps (see Fig. 1):

1. registration of pen-down,
2. place a rectangular frame around the current pen position and wait for the pen to intersect this frame,
3. make this intersection the new current point, and repeat steps 2 and 3 until a pen-up is registered.

The frame size can be chosen in terms of gridlines as the first, second, or third frame around the current point. So, the tablet uses space sampling rather than time sampling. The result of the sampling process is an  $(x, y)$ -coordinate file. The temporal order of the strokes that compose the character is preserved in this manner, which, as we shall see, combines well with a special class of HMM's. Finally, we notice that this type of coding is very similar to Freeman coding of line drawings [8].

## IV. SYMBOL REQUIREMENTS

The next issue of importance is what features to use to represent a character. The requirements put on the features (i.e., the observable symbols) are all inspired by the same thought: the interclass scatter, i.e., the discriminant power between different characters should be as large as possible. On the other hand, the interclass variation, the variation between specimens of the same character, should be minimized. This leads to the following requirements on the features [2]: A good set of features should

1. be independent of translation, rotation, and linear scaling of the curve,
2. be chosen so that they do not replicate each other,
3. be easy computable, and
4. preferably (but not necessarily) employ the dynamic information provided by the writing tablet.

Since the writing tablet provides a time sequence of directional code vectors, we here use an angular curve description derived from the description used with the definition of generalized Fourier descriptors [9]–[12]. This description, discussed in Section VI, combines the following advantages:

1. The 2-D  $(x, y)$ -versus distance format is reduced to 1-D; angular direction versus distance along the trajectory of the curve. In addition, the temporal information is incorporated in the description.
2. By nature, angular information is independent of translation of the curve.
3. The description involves a size normalization procedure, thus yielding size-independency.

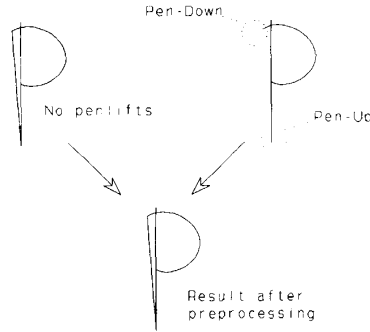


Fig. 2. Intra-class variability reduction through interpolation.

### V. INTRAClass-SCATTER REDUCING PREPROCESSING

Since characters sometimes consist of multiple strokes separated by pen-ups, a method has to be found to treat within-character pen-ups before the symbol generation can take place. Keeping in mind the formulation about intraclass scatter from the previous section, a very effective way of dealing with within-character pen-ups is to linearly interpolate the pen-up point with the consecutive pen-down point, i.e., drawing a straight line between these points in spatial domain. This will reduce the intraclass scatter, since some people retrace the pen in order to avoid lifting it from the paper [7], while others write the same character without lifting the stylus. The "pen-up version" will be mapped into the version without penlift. Fig. 2 illustrates this process for the character "p." Only characters "i" and "j" are treated differently: here, linear interpolation makes no sense, since the pen is not lifted to avoid retracing. Therefore, the length of the curve after the pen-down point is examined. When this length is very small compared to the total length of the curve, a dot is assumed, and a symbol "dot" is added to the feature vector. Now, the symbols that form the rest of the feature vector can be defined.

### VI. SYMBOL GENERATION

Let  $\phi(s)$  denote the absolute angular direction of the curve as a function of the distance  $s$  along the trajectory of the curve. Since the characters are captured by a tablet, we have  $N$  nonequidistant samples of  $\phi(s)$  (see also Fig. 1). Let  $s_i$  denote the length of an individual vector, then the total length of the curve is given by

$$L = \sum_{i=1}^N s_i \quad (1)$$

A reasonable independence of rotation is achieved by subtraction of the starting angle  $\phi(0)$  combined with a dehooking algorithm. Hooks are due to inaccuracies in pen-down detection and to rapid or erratic motion in placing the stylus on or lifting it off the tablet [7]. In any case, when subtraction of starting angle is used to achieve rotation-independency, these hooks must be removed, since they show a very unstable behavior. The used dehooking algorithm is very simple, yet effective. When the absolute (mod  $2\pi$ ) difference between the angular directions of two consecutive vectors within a predefined distance  $s_{\text{hook}}$  from the start of the curve exceeds a threshold  $\xi$ , all samples before this point are eliminated. Expressed in  $\phi$ , this criterion can be written as

$$\|\phi(s_i) - \phi(s_{i-1})\|_{2\pi} \leq \xi, \quad 0 \leq s_i \leq s_{\text{hook}} \quad (2)$$

where  $s_{\text{hook}}$  is chosen small compared to the total length  $L$  of the curve. Typical values for  $s_{\text{hook}} = 0.05 \cdot L$  and  $\xi = 1/2\pi$  yielded good results.

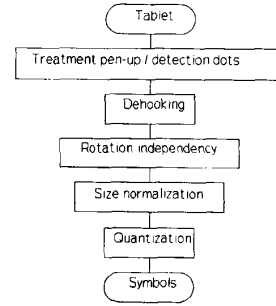


Fig. 3. Complete symbol generation procedure.

After dehooking, the  $s$ -axis is divided into  $T$  equidistant steps ( $T-1$  if a dot was found during preprocessing), and  $\phi[t]$  is obtained from  $\phi(s)$  by considering  $\phi(s)$  a piecewise linear function, i.e., linear interpolating the  $\phi(s_i)$ . Assuming no dot was found this can be written as

$$\phi[t] = \phi\left(t \cdot \frac{L}{T}\right), \quad t = 1, 2, \dots, T. \quad (3)$$

This normalization along the trajectory of the curve results in size-independency. Note that linear interpolation corresponds to a first-order reconstruction filter; in the spatial domain this means that the curve is reconstructed by circle segments.  $T$  must be chosen large enough to capture the highest spatial frequency of the characters. Since only the discrete HMM is considered, the  $\phi[t]$ -samples must be quantized in order to obtain discrete symbols. By straightforward uniform quantization  $\phi^*[t]$  is obtained

$$\phi^*[t] = Q\{\phi[t]\} = O_t, \quad t = 1, 2, \dots, T. \quad (4)$$

Resulting we have the desired observation sequence  $O$  containing  $T$  symbols. Fig. 3 shows the complete symbol-generation procedure. The next issue will be to determine the number of possible symbols per observation, i.e., the quantizer accuracy.

### VII. QUANTIZER ACCURACY

To determine the required accuracy of the quantizer, it is useful to recall the concepts from Section VI: the intraclass and interclass variation. Using these concepts, we can formulate the following requirement on quantizer accuracy: The difference between the quantized representation and its original may not be larger than the average intraclass variability according to some distance criterion.

Two relative error measures are defined,  $\epsilon_{\text{quantization}}$  and  $\epsilon_{\text{intraclass}}$

$$\epsilon_{\text{quantization}} = \frac{\sum_{t=1}^T \|\phi[t] - \phi^*[t]\|}{\sum_{t=1}^T \phi[t]} \quad (5)$$

$$\epsilon_{\text{intraclass}} = \frac{\sum_{t=1}^T \|\phi_1[t] - \phi_2[t]\|}{\sum_{t=1}^T \phi_1[t]} \quad (6)$$

where  $\phi[t]$  denotes the nonquantized description and  $\phi^*[t]$  its quantized counterpart.  $\phi_1[t]$  and  $\phi_2[t]$  are normalized descriptions of two different specimen of the same character.

To obtain an indication of the magnitude of the required number of quantization levels,  $\epsilon_{\text{intraclass}}$  and  $\epsilon_{\text{quantization}}$  were calculated. If

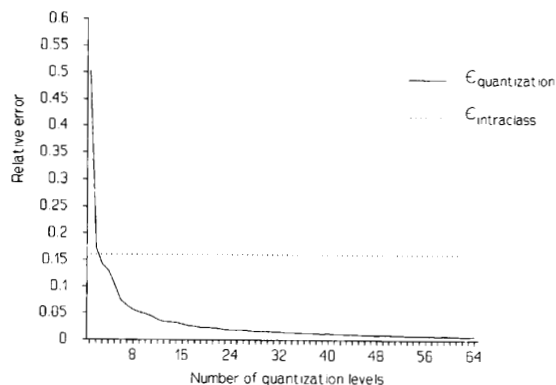


Fig. 4. Intraclass scatter and scatter due to quantization.

TABLE I  
OPTIMUM MODEL PARAMETERS

$N = 6$	LTR-1 structure
$M = 17$	16 quantization levels and symbol "dot"
$T = 64$	Observation sequence length
$K = 20$	20 examples per character
50 optimizations per character	

there are  $N$  specimens per character, then  $1/2 \cdot ((N-1)^2 + N-1)$  values for  $\epsilon_{\text{intraclass}}$  can be computed per character. Using  $N = 10$ ,  $\epsilon_{\text{intraclass}}$  was computed for  $26 \cdot 45 = 1170$  combinations. In addition,  $\epsilon_{\text{quantization}}$  was calculated for the 10 specimens, again for the complete alphabet. Fig. 4 shows a plot of the average of  $\epsilon_{\text{quantization}}$  as a function of the number of quantization levels. The average value for  $\epsilon_{\text{intraclass}}$  is also shown.

#### VIII. EMPIRICAL DETERMINATION OPTIMUM PARAMETERS

Extensive experimental work has been carried out in two stages: first, on a limited alphabet (typically characters "a"–"j"), an optimum set of parameters defining the HMM's was determined. With these optimum parameters, the performance of the system applied to the complete English alphabet was evaluated. The scheme used was to vary one parameter, while keeping all other parameters the same. Although the underlying assumption that the parameters are independent of each other probably is not entirely correct, this approach does limit the number of experiments

##### A. Model Structure

The result from the previous sections is an observation sequence (feature vector)  $\mathbf{O}$  that incorporates the temporal information provided by the tablet. This temporal order can be imposed on a HMM by constraining the model structure to left-to-right (LTR) [4], [5]. Once an  $a_{ij}$  is set to 0, it remains 0 when using the Baum–Welch reestimation formulas. Therefore, by initially setting  $a_{ij} = 0$  for a disallowed state transition, any model structure can be imposed. Left-to-right models must be trained on multiple sequences. Using one long observation sequence here makes no sense, since once the final absorbing state is reached, the further observations provide no information about earlier states. The performance of four different model structures was tested: an HMM with no constraints on state transitions, and 3 LTR HMM's (see Fig. 5): skipping of all states allowed (LTR-1), single skips only (LTR-2), and no skipping of states

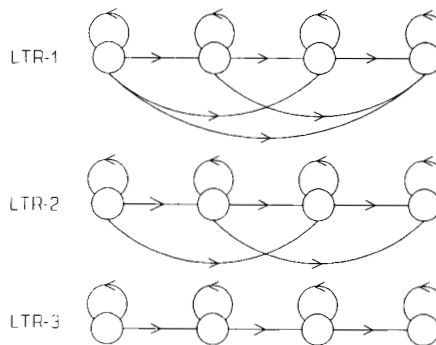


Fig. 5. Tested HMM structures.

allowed (LTR-3). In general, structure LTR-1 performed about 2–3 times better than structures LTR-2 and LTR-3 and about 4–5 times better than the structure without constraints. Clearly, the temporal order is an important aspect in the HMM structure. Since models LTR-2 and LTR-3 are subsets of model LTR-1, the obtained results are quite reasonable.

##### B. Initial Estimates

The main result of the first part of the experiments was that multiple optimizations per character, followed by selecting the model that yields the highest  $\Pr(\mathbf{O}|\mathbf{M})$  is an absolute necessity for good system performance. Experiments showed that even when an observation sequence  $\mathbf{O}$  is generated by a HMM using Monte Carlo simulation and the model is retrieved by the optimization procedure, for some initial estimates no correct model is found. One way to vary the initial estimates is by dividing the unit interval of each parameter into  $\mu$  sections, followed by optimizations for all possible combinations. For  $N$  states,  $M$  symbols, and  $\mu$  sections, the number of combinations is given by

$$C = \mu^{(N^2+N)/2+N \cdot M}. \quad (7)$$

Only the LTR-1 structure is assumed here. Setting  $\mu = 3$ ,  $N = 4$ , and  $M = 16$  already yields  $2 \cdot 10^{35}$  combinations, which is not a very attractive prospect. An alternative method is to use a random generator to generate initial estimates [5]. Experiments have shown that for approximately 50 optimizations per model, sufficiently strong local optima are found, i.e., with 100 optimizations under different initial estimates the same  $\Pr(\mathbf{O}|\mathbf{M})$  is found as under the 50 initial estimates.

##### C. Number of Symbols and States

The next issue under investigation was the number of symbols  $M$ , i.e., the quantizer accuracy. Based on the result from Section VII,  $M$  was varied from 8 to 24. The recognition error showed a descent from 12% ( $M = 8$ ) to 6.8% at  $M = 16$ . Then the error increased until 10.2% at  $M = 24$ . Similar experiments were conducted on the other parameters. For instance, for model size  $N = 6$ , the optimum recognition rate was achieved. In theory, using more states should lead to a better recognition performance due to the increased ability to capture variability. In practice, however, using more states does not necessarily increase the performance, probably because more states also need more training data to satisfy the training procedure [3]. The parameters shown in Table I show the experimentally optimal in terms of recognition accuracy, amount of training data needed, and classification time (set at less than 1 s on an AT-type PC).

TABLE II  
RECOGNITION ERROR RATES OF INDIVIDUAL AND COMBINED SYSTEMS

	Writer 1	Writer 2	Writer 3	Writer 4	Writer 5	Average error
Individual system	6.66%	6.54%	6.15%	2.69%	12.6%	6.94%
Combined system	14.2%	10.0%	17.7%	27.3%	26.5%	19.1%

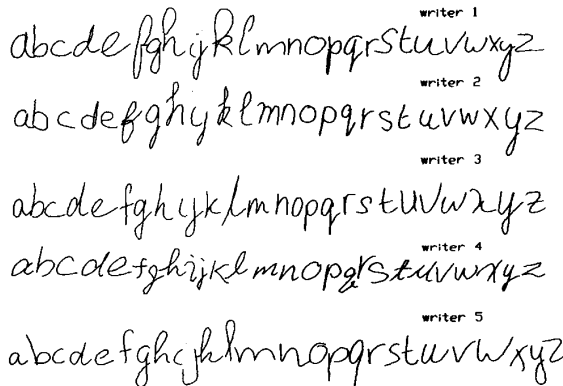


Fig. 6. Writing styles of five writers.

#### IX. SYSTEM PERFORMANCE EVALUATION

With the optimum set of parameters, recognition experiments were conducted on the alphabet consisting of the complete lower-case English alphabet. Five writers (three male, two female) were each instructed to enter 30 samples for each character in a fully unconstrained way and completely according to their own writing style. First, for each person independently HMM's were obtained by training on 20 samples per character, resulting in a system trained on its user (individual system). The other 10 samples were used as test characters, and the individual overall error rate, i.e., the average error rate over all characters was determined. Then, four training sequences from each of the five writers were used to obtain "combined" HMM's, i.e., one system trained on five users. Table II shows the recognition error rates that were obtained for both systems, and Fig. 6 shows the writing styles of the five writers. As can be seen, an average error rate of 6.9% was obtained for the systems trained specifically on its user (individual system). An error analysis showed that most confusions are between characters where the description differs only in a small durational way, like for instance the "a" and "d," who only differ in the higher extension of the vertical stroke for the "d." Apparently, the model is unable to capture these more subtle differences in durational information between two character descriptions. The confusion between two characters "a" and "b" may be due to the course quantization.

Due to the great difference in equipment, experimental protocols, data, etc., it is very difficult to compare results of experimental studies [6], [7]. Recognition rates varying from 71.9% to 100% have been reported [6], [7], depending on the alphabet, constraints, classification methods, the use of dictionaries, etc. Given the very unconstrained way in which the characters are allowed to be entered, the obtained results are quite good. Table II shows that the average performance of the system trained on five users (combined system) is about three times worse than the single-user system, so the main application lies in the individual system. Apparently the system is unable to effectively capture the enlarged variability introduced by multiple writers.

#### X. CONCLUSIONS

A system that classifies handwritten characters in an on-line fashion has been presented. If the system is trained on its user, an average error rate of 6.9% is achieved with fully unconstrained input of the characters. System performance degrades as the writer becomes more sloppy, which is obvious by comparing the results of writer 5, who wrote in a very hasty, sloppy way, with the result of writer 4. Performance can be increased by requiring writers to write neatly, which is a much more natural way of constraining than, for instance, constraints on the order or direction of the strokes [6]. This may even have an educational application.

Since ML-classification is used, the system can easily be extended depending on the application. For instance, weighting each  $\Pr(O|M)$  with occurrence probabilities for each character according to a language model can be used when the system is used mainly to enter text. A hypothesis generation scheme can easily be realized. In CAD applications, decision rules may be an alternative method to decide between ambiguous characters, although this method may put undesirable constraints on the handwriting.

#### ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their valuable suggestions and to E. Ooms for preparing the manuscript.

#### REFERENCES

- [1] A. Kundu, Y. He, and P. Bahl, "Recognition of handwritten word: First and second order hidden Markov model based approach," in *Pattern Recogn.*, vol. 22, no. 3, pp. 283-297, 1989.
- [2] A. Kundu and P. Bahl, "Recognition of handwritten script: A hidden Markov model based approach," in *Proc. ICASSP*, 1988, pp. 928-931.
- [3] R. Nag, K. H. Wong, and F. Fallside, "Script recognition using hidden Markov models," in *Proc. ICASSP*, 1986, pp. 2071-2074.
- [4] S. E. Levinson, L. R. Rabiner, and M. M. Sondhi, "An introduction to the application of the theory of probabilistic functions of a Markov process to automatic speech recognition," *Bell Syst. Tech. J.*, vol. 62, no. 4, pp. 1035-1074, 1983.
- [5] S. E. Levinson, L. R. Rabiner, and M. M. Sondhi, "On the application of vector quantization and hidden Markov models to speaker-independent, isolated word recognition," *Bell Syst. Tech. J.*, vol. 62, no. 4, pp. 1075-1105, 1983.
- [6] F. Nouboud and R. Plamondon, "On-line recognition of handprinted characters: survey and beta tests," *Pattern Recognition*, vol. 23, no. 9, pp. 1031-1044, 1990.
- [7] C. C. Tappert, C. Y. Suen, and T. Wakahara, "The state of the art in on-line handwriting recognition," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 12, no. 8, pp. 787-808, Aug. 1990.
- [8] H. Freeman, "On the encoding of the arbitrary geometric configurations," *IRE Trans. Electron. Comput.*, vol. EC-10, pp. 260-268, June 1961.
- [9] J. W. Vieveen and R. Prasad, "Generalised Fourier descriptors for use with line-drawings and other open curves," in *Proc. 6th Scandinavian Conf. Image Anal.* (Oulu, Finland), June 1989, pp. 820-827.
- [10] N. B. J. Weyland and R. Prasad, "Characterization of line-drawings using generalised Fourier descriptors," *Electron. Lett.*, vol. 26, no. 21, pp. 1794-1795, Oct. 1990.
- [11] "Criterion for characterization of line-drawings using generalised Fourier descriptors," in *Proc. 7th Scandinavian Conf. Image Anal.* (Ålborg, Denmark), Aug. 1991, pp. 48-55.
- [12] L. Yang and R. Prasad, "Recognition of line-drawings based on generalised Fourier descriptors," in *Proc. 4th IEEE Int. Conf. Image Processing Applicat.* (Maastricht, Netherlands), Apr. 1992, pp. 286-289.
- [13] R. Rosenfeld, X. Huang, and M. Furst, "Exploiting correlations among competing models with application to large vocabulary speech recognition," in *Proc. ICASSP*, 1992, pp. 1-5-1-8, vol. I.