Preprint of: Schomaker, L.R.B., & Teulings, H.-L. (1992). Stroke- versus Character-based Recognition of On-line, Connected Cursive Script. In J.-C. Simon & S. Impedovo (Ed.), *From Pixels to Features III* (pp. 313-325), Amsterdam: North-Holland.

Stroke- versus Character-based Recognition of On-line, Connected Cursive Script.

L.R.B. Schomaker and H.-L. Teulings*

*Nijmegen Institute for Cognition Research and Information Technology (NICI).
P.O.Box 9104, 6500 HE Nijmegen, The Netherlands. E-mail: schomaker@nici.kun.nl.

Abstract

In this study, a comparison is made between a stroke-based, and a character-based recognizer of connected cursive script. Experiments were performed with Kohonen's topology-preserving neural network. In one method, a feature vector was based on single velocity-based strokes and a subsequent symbolic character classification stage, as reported earlier. In an alternative approach, kinematically segmented characters were time-normalized and represented in a feature vector as a whole. In the latter case, the character classification is done completely within the Kohonen method and does not require a separate symbolic matching stage. Two distance measures are used, Euclidean and variance-weighted Euclidean. The method displays a good unsupervised clustering of allographs in a single network. Weighted Euclidean distance performs better than a standard Euclidean distance, and results in less "allonyms" (name confusion) per allograph. It is concluded that by combining the discriminative stroke-matching method and the more generalizing character-matching method in a single system, recognition rate can be improved.

1. INTRODUCTION

In an earlier study [1], an on-line cursive-script recognition system was described that is based on kinematical strokes, i.e., pieces of handwriting movement bounded by minima in the tangential pen-tip velocity. These time-reference points are robust in adult handwriting [2-5] and are associated with peaks in the curvature time function. In such an approach, a character is defined as a "sequence of strokes". Shape classification is done at the stroke level. A Kohonen [6] self-organizing network (Topology Preserving Map) was used to obtain feature vector quantization, reducing a 14-dimensional stroke feature vector to two indexes i,j into a 2-D Kohonen network with hexagonal connectivity between neurons [7]. After stroke feature vector quantization, a symbolic classification procedure

Figure 1. Typical errors produced by a stroke-based cursive script recognizer. Left: Correct shape classification of */lazy/*. Right: misclassification of */b/* as */lr/* by overdiscrimination, in the word */brown/*. Also note a human writing error in */w/*

is performed on strings of symbolic stroke codes to identify characters. This approach yields an estimated 70-80% correct character classification of neatly written on-line cursive script without the use of any linguistic statistics and/or context. This figure is an estimate, since shape ambiguity in cursive script makes an elegant assessment of the recognition rate complicated. From the geometric point of view, often more than one character candidate is perfectly acceptable, whereas linguistically, only a single character is considered "correct". A notorious example is the cursive word */minimum/* written without punctuation, which allows for a very large number of interpretations containing */i/*,*/u/*,*/n/*,*/m/*, and even */c/*,*/v/*. However, the stroke-based approach lacks the generalization capability that is necessary to display invariance to the geometric distortions and variabilities inherent to cursive script. This incomplete generalization capability, leading to a undesired proportion of "rejected" handwriting fragments, is mainly due to the discrete and symbolic matching phase, where all metric information on structural relations within characters is lost. There are two motives for investing efforts in the improvement of bottom-up shape classification.

First, it is still evident to the human observer that our recognition system could have incorporated geometrical information that is present in the input, but fails to do so (Figure 1). Since there is no between-stroke information represented, the system cannot make use of high-order relations between parts of a character.

Second, although incorporation of linguistic context theoretically improves the recognition rate, there are currently no generally accepted, robust and broadly applicable large-capacity (> 100000 word dictionary) techniques available. A large variety of techniques is being explored, varying from hash coding [8], triphone analysis [9], dictionary tree [10] and trie search [11], to Hopfield networks [12], but there is apparently no consensus about basic issues. Furthermore, finite dictionaries often deteriorate particular texts by replacing partially distorted words (names, abbreviations) by completely wrong word alternatives.

Thus, before considering the use of linguistic knowledge, the quality of geometric shape classification should be optimized. To improve the performance in this area, an alternative approach is studied as an enhancement to the existing stroke-based system. In pilot studies it was noticed that taking Euclidean distances between user-identified prototypical characters and a given stream of cursive handwriting input yields a large proportion of spurious hits often even at non-character positions (Figure 2). The reason for the spurious matches lies in (1) *inherent ambiguities*, and (2) in the *redundance* or periodicity of the up-down movement sequence.

Inherent *ambiguities* exist when the Euclidean distance between a part of a character* and a prototype is very small, as in, e.g., the first stroke of a cursive */a/* or */d/* that may have a perfect fit (zero distance) with the single-stroked character */c/*.

The *redundance* problem occurs when, e.g., down-up-down-up-down sequences display a spuriously small distance from */m/*. This phenomenon will also lead to a poor performance when classifying connected cursive script by means of multi-layer perceptrons (MLPs) trained by back propagation (BP) as we have found. However, these networks do appear to work well on isolated characters [13] where this type of redundance is absent. Simon & Baret [14] makes a distinction between regularities and singularities in handwriting, to solve the redundance problem by only concentrating on the geometric singularities in the classification process, but the singularities are defined a priori, which is a disadvantage from the trainability point of view.

The purpose of the present study is to find a method that reduces the "overgeneralization" that occurs in the case of (Euclidean) character matching, but that still displays a degree of generalization that is sufficient to fill in the gaps (misses and false hits) that are present in the existing "overdiscriminating" symbolic stroke-based approach [1]. To implement a character-base recognition system for on-line connected cursive script, a number of problems must be solved:

1.1 Data acquisition

Pen-tip movements are recorded by a Calcomp 2500 or 9060 digitizing tablet at 100 samples/second, measuring displacement (X,Y) and axial pen force ("pressure") (Z). The recorded handwriting is of an adult male writer and consists of English and Dutch paragraphs of text, as well as a limited number of isolated block print letters and digits, yielding a total of 629 words, and 5327 labeled characters. The labeling is done manually with an interactive "mouse"-pointer user interface. However, it is desirable to do this (semi-)automatically, for which methods are being developed [15]. A separate test set consisted of 244 cursive words.

---

*In cursive script, it is often preferable to substitute the term "characters" by "allographs", i.e., different shapes for the same symbol. Also, there exist "allonyms", i.e., different symbol names for the same shape ({ */0/*, */o/* and */O/* }, or cursive {*/e/* and */l/*}).

Figure 2. Misclassification in template matching of cursive script using an Euclidean distance measure. The correct */k/* actually has a higher distance from the input than an */h/*-shaped prototype. The smallest distance is obtained by matching the first two strokes with an */l/* shape. The word */gemakkelijks/* is Dutch and means "something easy".

1.2 Preprocessing

The handwritten input word is pre-processed (low-pass filtered, differentiated, unrotated, unslanted, size-normalized) and segmented on the basis of minima in the tangential velocity by standardized software [1,2,16].

1.3 The coding of the movement signals

In the newly proposed system, a feature vector does not describe a stroke, but a complete character. The character feature vector consists of a sequence of consecutive x,y coordinates of pen-tip movement, normalized to a centroid position of $(0,0)$ and a maximum excursion of 1. Thus, the feature set is homogeneous: all features are in the same unit, and there are no structural features. No provisions have been made for pen lifting within characters, as yet. In real connected cursive script, pen-lifting is not supposed to occur. Note that pen-lifts actually are incorporated in the stroke-based recognition system [1].

1.4 The varying-duration problem

The varying duration of different characters, expressed in the number of samples and/or the number of strokes leads to feature vectors of varying sizes for different characters. In an attempt to solve the varying-duration problem, Morasso et al. [7] propose the employment of separate Kohonen networks for 2 to 7-stroked characters separately. The disadvantage is that one cannot make use of the clustering capability of the Kohonen network over all possible n-stroked characters: the "Voronoi tesselation" only takes place within a network dedicated to characters of a fixed number of strokes and there is no mutual influence between these networks during training. In the current study, however, we will try to represent all allographs in a single Kohonen network. In order to achieve this, characters must be represented by a single fixed-size feature vector. The adopted solution is to take the kinematic stroke segmentation moments in time as anchor points for a time axis normalization. The number of normalized time samples is chosen to be 30 samples, yielding a worst case of 5 samples per stroke in a 6-stroked character (e.g., some allographs of */m/*). At the other end, single-stroked characters like */i/* and */c/* are represented with 30 samples per stroke. The time-axis normalization is performed using a quadratic interpolation procedure [17] which has the advantage of requiring substantially less computing time than more sophisticated methods [18, 19] but exhibiting better performance than linear interpolation at the peaks and dips of the displacement time functions.

1.5 Creating a training set and handling allographic variation

Allographic variation refers to the phenomenon that a given letter can be written in several, topologically different ways, such as an upper-case variant, a block print variant, and a cursive variant of the letter */t/*. In cursive script, with its idiosyncratic writer styles, labeling a character by its letter name only does not automatically make it unique compared to its competitors or similar to its namechildren. Different letter labels do not automatically ensure that the allograph shape indeed is unique. Without a scale context, cursive */e/*'s and */l/*'s may have exactly the same (size-normalized) shape. On the other hand, a writer may have several variants (allographs) for a */t/*, topologically differing to such an extent that it is not sensible to define the different shapes as belonging to the same shape class. As an example, in a MLP, containing 26 output nodes for the alphabet, allographic variation for a given letter will very soon lead to mutual interference of weights and a consequently reduced convergence and recognition rate in such an architecture. By normalizing the time axis and using a fixed-dimension feature vector (see above), we can now take advantage of a single self-organizing Kohonen network that settles down to a least-squares, topologically correct map of all the allographs in a training set. A sufficient number of the obtained feature vectors must be labeled with its letter name to obtain a robust representation of allographs. As a rule of thumb, Jain & Chandrasekaran [20] mention a number of examples per class that is five times the number of features. In the current case, this implies 300 examples per class, which is a number that cannot easily be reached in an interactive system were the user must provide for the complete training set.

1.6 Training

The training of the Kohonen network was done in 50 epochs, starting with a learning rate of 0.5 and ending with 0.01, with a steepness factor of 5. The steepness factor is used in:

$$x_k = \left((\sqrt[s]{x_1} - \sqrt[s]{x_N}) \frac{(k-1)}{(N-1)} + \sqrt[s]{x_N} \right)^s \tag{1}$$

where $s(>0)$ is the steepness factor, x is a decreasing training parameter (here learning rate or Kohonen bubble radius), $k = [1, N]$ is the epoch number and N is the total number of epochs. If $s = 1$, x_k is a linear function. Network bubble radius similarly decreased from full network size to 0 with a steepness factor of $s = 5$. This relatively high steepness speeds up the self-organizing process by reducing the duration of the initially irregular state space evolution. The network size was 20x20 neurons, organized in a hexagonal-connectivity 2-D grid.

1.7 Operation: The segmentation of the data into characters

Assuming there exists a fixed-dimensional feature vector for each allograph as a class prototype, segmenting unknown input handwriting into characters can be done in a similar fashion as described above, with the difference, that we are now dealing with unlabeled input. This means that for each given stroke position, handwriting fragments must be time-normalized and presented to the trained Kohonen network. For each stroke of an input word, pieces of handwriting containing from 1 to 6 strokes are resampled to 30 samples and presented to the Kohonen network, determining the closest match is determined. Thus, for each stroke position, 6 hypotheses concerning the character identity are generated, along with a measure of their fit with the actual input.

2. RESULTS

After training, a Kohonen network of cursive characters evolved that is presented in Figure 3. The labeled training set was presented to the Kohonen network once again, this time not to update the weights (prototypical feature vectors), but to label the nearest neighboring neuron with the name of its matching example. Of the 400 neurons, 387 were actually labeled, the remaining neurons representing intermediate character shapes and irrelevant mixtures. The fact that not all neurons were needed shows that the number of neurons was sufficient to represent the topology of the allographs present in the training set. Note that the presence of a large number of connecting strokes in the training set caused a salient presence in the network.

During the automatic labeling of neurons, the software kept track of the squared deviations between prototype and input for each feature of each neuron to obtain an estimate of the variability of the feature per allograph σ_i^2. Normalizing by also correcting for covariance (Mahalonobis distance) was avoided because of the anticipated error of the covariance estimates in the high-dimensional (=60) space used [21,22]. The resulting distance measure is more sensitive to essential (i.e. invariant) sections within the feature vector. Table 1 shows the distribution of neurons as a function of the number of interpretations.

Figure 3. The complete 20x20 Kohonen network representing the allographs in the training set after 50 epochs.

Table 1.
The histogram of neurons per number of interpretations (labels).

	D^2_{kj}	$\tilde{D}^2_{kj}$
N_{labels}	$N_{neurons}$	$N_{neurons}$
1	102	130
2	145	133
3	81	51
4	33	32
5	26	36
empty	13	18
Total	400	400

Column D^2_{kj} denotes the use of the simple, column $\tilde{D}^2_{kj}$ the use of the weighted Euclidean distance.

Two distance measures were used to find the closest prototype of an input character, Euclidean:

$$D^2_{kj} = \frac{1}{N_{feat}} \sum_{i=1}^{N_{feat}} (f_{ki} - f_{ji})^2 \tag{2}$$

and feature-variance normalized (weighted) Euclidean:

$$\tilde{D}^2_{kj} = \frac{1}{N_{feat}} \sum_{i=1}^{N_{feat}} \frac{(f_{ki} - f_{ji})^2}{\sigma^2_{ki}} \tag{3}$$

where k denotes the prototype index, j is the input pattern index, f_i are the feature values and σ^2_{ki} is the variance of differences for feature i between prototype k and all corresponding examples in the training set. In using the Euclidean distance (middle column, Table 1), most neurons can have 2 interpretations (145), followed by 1 interpretation/neuron (102), then dropping steeply. In case of the weighted Euclidean distance, the number of neurons with a single interpretation increases (130), right column: the extent of the confusion decreases.

Table 2 shows the confusion list. Numbers indicate the frequency of occurrence in the training set. Up to four confusions are shown (the maximum is five, see Table 1). Confusions are sorted in order of decreasing frequency of occurrence. Note that there are special codes: # refers to a horizontal spacing movement, a _ refers to a connecting stroke, and a ˜ refers to unclassifiable fragments.

Figure 4 shows the variation around some character prototypes of the 20x20 Kohonen network using ellipsoids, representing the horizontal and the vertical rms error. As can be seen, there are differences between the characters with respect to their motoric writing variability. Also, the variance is not homoscedastical over features (e.g. the /r/ in panel (8,8). Occasionally, neurons contain meaningless shapes when they lie a transition area of very different allographs. These outliers belong mostly to the group of unlabeled neurons. Table 3 displays preliminary recognition results for a single writer. The results are expressed as the percentage of words that could be identified in the solution space of best matching characters.

Table 2.
The confusion list for the labels attached to the (20x20) Kohonen neurons.

!	!(10)	~(1)	?(1)	j(1)	T	T(6)	t(2)		
”	_(73)	~(22)	r(19)	”(9)	U	a(50)	n(35)	u(19)	U(8)
#	_(109)	~(93)	#(33)	e(13)	V	o(45)	v(38)	V(8)	_(1)
’	_(66)	~(61)	’(28)	”(6)	W	w(31)	W(8)	m(1)	r(1)
,	r(11)	c(8)	~(8)	,(8)	X	X(6)	x(5)	Y(1)	
.	~(68)	_(59)	z(17)	.(15)	Y	g(11)	Y(7)	X(6)	y(6)
0	o(64)	0(21)	e(14)	i(8)	Z	z(20)	2(12)	Z(8)	7(2)
1	i(44)	1(20)	~(10)	f(7)	_	_(1174)	~(132)	r(26)	o(25)
2	2(17)	c(8)	~(8)	j(8)	‘	~(24)	’(7)	”(5)	‘(3)
3	3(16)	?(9)	.(4)	~(3)	a	a(171)	o(62)	n(14)	k(13)
4	4(14)	e(13)	b(7)	h(5)	b	b(73)	h(51)	t(14)	k(10)
5	5(11)	y(9)	s(4)	~(4)	c	c(69)	i(69)	n(49)	e(22)
6	6(13)	t(2)	j(1)	G(1)	d	d(126)	m(35)	l(9)	x(6)
7	z(17)	7(13)	Z(5)	J(3)	e	e(541)	_(297)	l(85)	n(36)
8	8(14)	j(3)	‘(1)		f	f(45)	t(39)	6(7)	e(5)
9	g(60)	9(24)	q(23)	y(16)	g	g(78)	y(65)	q(29)	9(23)
:	f(2)	(1)			h	h(86)	k(28)	b(14)	t(11)
?	!(10)	?(10)	3(8)	j(1)	i	i(182)	u(32)	c(31)	r(30)
A	~(18)	A(10)	t(6)	N(3)	j	y(30)	j(27)	!(10)	~(8)
B	B(7)	K(5)	Q(4)	R(3)	k	k(102)	h(53)	b(5)	K(5)
C	c(52)	C(17)	e(7)	i(5)	l	e(244)	l(105)	t(34)	b(12)
D	D(12)	n(10)	H(5)	Y(4)	m	m(83)	o(10)	w(9)	~(4)
E	E(6)	F(5)	~(3)	n(1)	n	n(249)	u(46)	~(29)	x(17)
F	F(6)	U(2)	d(2)	R(1)	o	o(228)	a(53)	n(42)	v(42)
G	o(46)	6(6)	0(5)	G(4)	p	p(74)	o(25)	H(8)	x(6)
H	x(13)	H(10)	D(7)	p(5)	q	g(64)	y(45)	q(33)	9(19)
I	j(8)	1(5)	I(5)	i(4)	r	r(219)	i(52)	o(47)	~(34)
J	j(9)	J(6)	7(3)	i(3)	s	s(141)	r(71)	~(22)	i(17)
K	K(5)	B(5)	Q(4)	R(3)	t	t(220)	f(36)	y(26)	l(23)
L	t(4)	k(4)	L(4)	h(2)	u	n(125)	u(77)	a(50)	U(6)
M	o(6)	D(6)	H(4)	M(3)	v	o(129)	v(106)	V(8)	_(6)
N	n(11)	N(9)	A(2)	w(1)	w	w(56)	W(8)	v(5)	~(4)
O	o(75)	a(50)	0(10)	O(9)	x	n(145)	x(30)	e(14)	a(13)
P	p(47)	P(6)			y	y(90)	g(51)	q(19)	9(17)
Q	K(5)	B(5)	Q(4)	R(3)	z	z(63)	h(16)	r(9)	Z(6)
R	K(5)	B(5)	R(4)	Q(4)	~	_(654)	~(408)	e(25)	r(24)
S	s(44)	S(13)	t(4)	L(3)					

There were 5327 characters in total. The reader is referred to the text for more details.

Figure 4. Some prototypical feature vectors (allographs) detected by the Kohonen neural network and the horizontal and vertical rms error at each normalized-time sample represented as an ellipsoid.

Table 3.
Average recognition results for one writer (N=244 test words).

Recognizer	% Identifiable words	Number of competing letter hypotheses
A. Stroke-based	47	4.0
B. Character-based	65	13.2
A. & B.	83	16.6

The second column gives the percentage of identifiable words (if lexical post-processing were applied), the third column displays the average number of competing letter hypotheses per cursive stroke in the input stream (A: stroke-based recognizer, B: character-based recognizer, A & B: the results if the solution space is a combination of the output of A and B).

The character-based recognizer yields a better (+18%) performance at the cost of a larger number (+9) of competing letter hypotheses per stroke. Combining letter hypotheses from the stroke-based approach and the character-based approach yields a 83% word identification. It should be noted, however, that the character-based approach requires a large amount of computation due to the high dimensionality of the feature vectors (60 vs 14) and the multiple (6x) matching per serial stroke position.

3. CONCLUSION

The presented self-organizing approach offers new possibilities in the recognition of connected cursive script. A useful system will heavily rely on user trainability, given the broad range of cursive writing styles. Among other things, the approach illustrates that spatial resampling is not necessary because of the existing movement replicability. The use of the writing velocity as the basis for finding temporal anchor points in matching appears equally applicable to single strokes and to characters as a whole. Results on one writer show that by combining a discriminative stroke-matching method with the more generalizing character-matching method presented here, recognition rate can be improved. The post-hoc use of a weighted Euclidean distance measure raises the question whether this measure could be also used during the training of a Kohonen self-organizing map, instead of the usual Euclidean or city-block distance. Theoretically, this is possible if each neuron has a representation of a running-average squared-feature vector as the basis for the feature variance estimation. Future work will be concerned with questions of training set requirements and with more detailed comparisons between the two recognition methods.

This study was supported by an Esprit grant, project P5204.

4. REFERENCES

1 Schomaker, L.R.B. & Teulings, H.-L. (1990). A handwriting recognition system based on properties of the human motor system. *Proceedings of the first international conference on frontiers in handwriting recognition, Concordia University.* Montreal: CENPARMI.

2 Teulings, H.-L., Schomaker, L.R.B., Morasso, P., & Thomassen, A.J.W.M. (1987). Handwriting-analysis system. In R. Plamondon, C.Y. Suen, J.-G. Deschênes, & G. Poulin (Eds.), *Proceedings of the Third International Symposium on Handwriting and Computer Applications* (pp. 181-183). Montreal: Ecole Polytechnique.

3 Morasso, P. (1986). Trajectory Formation. In Morasso, P. & Tagliasco, V. (Eds.), *Human Movement Understanding.* Amsterdam: North-Holland.

4 Plamondon, R., & Maarse, F.J. (1989). An evaluation of motor models of handwriting. *IEEE Transactions on Systems, Man and Cybernetics, 19*, 1060-1072.

5 Schomaker, L.R.B. (1991). Simulation and Recognition of handwriting movements: A vertical approach to modeling human motor behavior. Doctoral Dissertation. Nijmegen (The Netherlands): University of Nijmegen.

6 Kohonen, T. (1987). Adaptive, associative, and self-organizing functions in neural computing. *Applied Optics, 26*, 4910-4917.

7 Morasso, P., Kennedy, J., Antonj, E., Di Marco, S., & Dordoni, M. (1990). Self-organisation of an allograph lexicon. International Joint Conference on Neural Networks, Lisbon, March 1990.

8 Kohonen, T. (1988). The "Neural" Phonetic Typewriter *IEEE Computer, 3,* 11-22.

9 Van Berkel, B. & De Smedt, K. (1988) Triphone analysis: a combined method for the correction of orthographical and typographical errors. *Proceedings of the Second Conference on Applied Natural Language Processing, Austin TX.* American Association for Computational Linguistics

10 Maier, M. (1986). Separating characters in scripted documents. Use of a dictionary in conjunction with a handwritten text recognizer. *Proceedings of the 8th International Conference on Pattern Recognition, Paris, October 27-31,* 1056-1058.

11 Wells, C.J., Evett, L.J., Whitby, P.E. & Whitrow, R.J. (1990). Fast dictionary look-up for contextual word recognition. *Pattern Recognition, 23,* 501-508.

12 Jagota, A. & Hung, Y.-S. (1990). A neural lexicon in a Hopfield-style network. *Proceedings of the International Joint Conference on Neural Networks (IJCNN-90-WASH-DC), Vol. 2,* 607-610.

13 Guyon, I., Albrecht, P., Le Cun, Y., Denker, J. & Hubbard, W. (1991). Design of a neural network character recognizer for a touch terminal. *Pattern Recognition, 24,* 105-119.

14 Simon, J.C. & Baret, O. (1990). Handwriting Recognition as an application of Regularities and Singularities in Line Pictures. *Proceedings of the International Workshop on Frontiers in Handwriting Recognition (IWFHR).* (pp. 23-37). Montreal: CENPARMI Concordia.

15 Teulings, H.-L., & Schomaker, L.R.B. (1991). Unsupervised learning of prototype allographs in cursive-script recognition using invariant handwriting features. *Proceedings of the 2nd International Workshop on Frontiers in Handwriting Recognition, 23-27 sept 1991, Bonas, France,* pp. 45-55.

16 Thomassen, A.J.W.M., Teulings, H.-L., Schomaker, L.R.B., Morasso, P., & Kennedy, J. (1988). Towards the implementation of cursive-script understanding in an online handwriting-recognition system. In Commission of the European Communities: D.G. XIII (Ed.), *ESPRIT '88: Putting the technology to use. Part 1* (pp. 628-639). Amsterdam: North-Holland.

17 Teulings, H.L., & Maarse, F.J. (1984). Digital recording and processing of handwriting movements. *Human Movement Science, 3,* 193-217.

18 Woltring, H.J. (1986). A FORTRAN package for generalized, cross-validatory spline smoothing and differentiation. *Advances in Engineering Software, 8,* 104-113.

19 Schomaker, L.R.B., & Thomassen, A.J.W.M. (1986). On the use and limitations of averaging handwriting signals. In H.S.R. Kao, G.P. van Galen, & R. Hoosain (Eds.), *Graphonomics: Contemporary research in handwriting* (pp. 225-238). Amsterdam: North-Holland.

20 Jain, A.K. & Chandrasekaran, B. (1982). Dimensionality and sample size considerations in pattern recognition practice. Handbook of Statistics, Vol. 2, Amsterdam: North Holland.

21 Van Ness, J. (1980). On the dominance of non-parametric Bayes rule discriminant algorithms in high dimensions. *Pattern Recognition, 12,* 355-368.

22 Deinet, W, Linke, M & Rieger (1987). *Analyse der Schreibdynamik.* Technical Report, Bundeskriminalamt (BKA-Technische Forschung), Wiesbaden.