



## Iconic and multi-stroke gesture recognition

Don Willems, Ralph Niels, Marcel van Gerven, Louis Vuurpijl\*

Donders Institute for Brain, Cognition and Behaviour, Radboud University Nijmegen, P.O. Box 9104, 6500 HE Nijmegen, The Netherlands

### ARTICLE INFO

#### Article history:

Received 11 August 2008  
Received in revised form 6 January 2009  
Accepted 12 January 2009

#### Keywords:

Iconic gestures  
Multi-stroke gesture recognition  
Feature selection

### ABSTRACT

Many handwritten gestures, characters, and symbols comprise multiple pendown strokes separated by penup strokes. In this paper, a large number of features known from the literature are explored for the recognition of such multi-stroke gestures. Features are computed from a global gesture shape. From its constituent strokes, the mean and standard deviation of each feature are computed. We show that using these new stroke-based features, significant improvements in classification accuracy can be obtained between 10% and 50% compared to global feature representations. These results are consistent over four different databases, containing iconic pen gestures, handwritten symbols, and upper-case characters. Compared to two other multi-stroke recognition techniques, improvements between 25% and 39% are achieved, averaged over all four databases.

© 2009 Elsevier Ltd. All rights reserved.

## 1. Introduction

The research described in this paper is motivated by the development of pen input recognition technologies for *iconic gestures*. Such gestures have a visually meaningful shape and are therefore easier to learn and remember by the users of pen-aware systems than abstract gestures which have no obvious relation between shape and semantics [1,2]. In the ongoing ICIS project [3], iconic gestures are used to indicate events or objects on interactive maps. ICIS aims at the domain of crisis management, where pen input devices like a tabletPC or PDA are used to convey messages. The typical pen interactions that emerge in these scenarios were explored in [4]. The categorization of the obtained pen gestures showed that next to route descriptions and markings of locations, the iconic sketchings of, e.g., cars, fires, casualties, accidents, or persons occurred quite frequently. In accordance with these observations, we designed and collected a suitable set of iconic gestures for specifying objects and events. The acquired database is called Niclcon [5] and is publicly available via <http://www.unipen.org>.

A wide range of pen gesture recognition systems have been described in the literature, like Rubine's GRANDMA system [6], Quickset [7], SILK [8], and iGesture [9]. For a recent review, the reader is referred to [1]. The majority of these systems target either the recognition of command gestures [10–12] (e.g., arrow up/down/left/right for scrolling, or gestures for performing delete/undo actions) or the sketches and drawings for design applications [9,13]. Most gesture

recognition systems employ Rubine's 13 global features, which are computed from a complete gesture shape. Rubine's features have mainly been used for recognizing single-stroke gestures like the unistroke [14] or graffiti alphabets [15]. Unfortunately, they are only moderately successful when applied to multi-stroke pen input [9,16].

Multi-stroke gestures pose similar problems to recognition technologies as handwritten characters or symbols. Shape variations, differences in stroke ordering, and a varying number of strokes have to be taken into account (see Fig. 1). There are several approaches to tackle these problems. The first employs modeling of stroke sequences. For example, using hidden Markov models (HMMs) [19] or dynamic graphical models [20], each stroke is mapped to an individual stroke model, which can be implemented as HMM states or nodes from a graphical model. The second approach captures variability in stroke length and stroke sequences through feature representations such as chain codes or spatio-temporal resampling [21]. Third, dynamic programming algorithms such as dynamic time warping (DTW) [22] can be employed for performing non-linear curve matching. Finally, to improve the processing of multi-stroke gestures, more elaborate and distinguishing features can be computed from a global multi-gesture shape [9,16], similar to Rubine's algorithms.

The current paper focuses on the latter approach: the design and evaluation of new features for multi-stroke gesture recognition. To this end, four publicly available databases containing multi-stroke gestures will be explored. The distinguishing properties of different groups of features are evaluated for these datasets, by using the best individual N (BIN) feature selection algorithm [23] and two well-known classification algorithms. The results will be compared to two alternative methods: classification based on spatio-temporally resampled gesture trajectories [21,24] and based on DTW [22,25].

\* Corresponding author.

E-mail address: [l.vuurpijl@donders.ru.nl](mailto:l.vuurpijl@donders.ru.nl) (L. Vuurpijl).

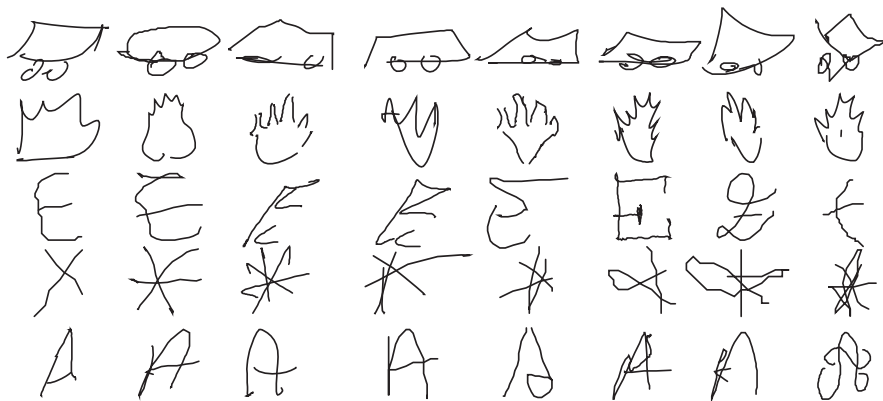


Fig. 1. Multi-stroke gestures with varying shapes and different number of strokes. Depicted are the classes 'car' and 'fire' from the Niclcon collection (first and second row), capitals 'E' and 'A' from both the UNIPEN [17] and the IRONOFF [18] databases (third and fifth row), and symbols from the UNIPEN database (fourth row).

In the next section, we will briefly describe the four databases. In Section 3, we will elaborate on different feature sets that can be employed for multi-stroke gesture recognition. In particular, new features will be presented that are based on the observation that 90% of the iconic gestures contained in the Niclcon dataset and a large portion of handwritten gestures contained in other data collections are drawn in multiple strokes. For each gesture, the features are computed along the complete gesture shape as well as along each individual stroke. As we will show through feature selection and recognition performances (Sections 4 and 5), adding mean and standard deviations of the individual stroke features has a very positive impact, which may also be of value for other applications in pattern recognition.

## 2. Databases containing multi-stroke gestures

For the experiments described in this paper, we have considered four different collections of multi-stroke gestures. The first is the Niclcon [5] database of iconic gestures which was recently made publicly available. Since for this paper, we have used a modified segmentation algorithm for isolating pen gestures, we briefly report on the differences with respect to [5] in Section 2.1. The other three databases are well known and comprise the UNIPEN 1d collection of handwritten symbols [17], the UNIPEN 1b collection of handwritten capital characters, and the handwritten capitals contained in the IRONOFF database [18]. From each collection, we excluded the single-stroked samples. From the remaining samples, three subsets were extracted (a training set and a test set for optimizing a classifier and an evaluation set which is kept hidden until final assessments). Stratified random sampling was used, such that each subset contains the same relative number of samples per class. The data were divided such that training, test, and evaluation sets contain 36%, 24% and 40% of the samples, respectively.

### 2.1. The Niclcon database of iconic pen gestures

The gesture repertoire from the Niclcon database was based on the icon lexicon from the IconMap application developed by Fitriani and Rothkrantz [2]. In IconMap, users can convey information about crisis situations by clicking on a well-designed set of icons. Although, as discussed in [2], iconic communication for this domain is new, the icon shapes used in IconMap are based on a standard set of icon classes used by the governments of the United States, Australia and New Zealand [26]. From the set of icons in IconMap, we constructed an icon lexicon containing the fourteen classes depicted in Fig. 2 and representing a representative subset of

the messages contained in [26]. It should be noted that these iconic gestures were collected in a laboratory setting where subjects were sitting at a desk, filling in well-designed boxed forms. Since this is far from the envisaged mobility contexts, these data should be considered as a first step toward the design of pen input recognition technology for interactive maps. On the other hand, collecting isolated handwritten characters or words for training handwriting recognition systems is a common approach. Consider, for example, the IAM database [27] and databases containing non-Western scripts like Japanese [28,29], Tamil [30,31], and Arabic [32].

The automated segmentation of the online data in iconic gestures reported in [5] employed both temporal information and spatial layout characteristics, resulting in 24,441 samples. However, due to a particular way of entering gestures the temporal stroke ordering for several samples was disturbed. We modified the segmentation algorithm such that we were able to recover these samples, resulting in a total of 26,163 iconic gestures. By discarding gestures with only one pendown stroke, in total 23,641 iconic gestures were selected. Table 1 shows the distribution of samples distinguished in the fourteen gesture classes. The table also shows the average number of strokes that users employ to draw an iconic gesture class. On average, 5.2 strokes are used for each iconic gesture.

### 2.2. Handwritten capitals and symbols

To assess the generalizability of our approach, three standard online databases containing multi-stroke gestures are explored as well. The data contain uppercase characters from the UNIPEN [17] and IRONOFF [18] collections and a selection of symbols from the UNIPEN collection. The IRONOFF database contains isolated characters, digits, and cursive words written by French writers. We used the IRONOFF 'B-forms' subset, containing 10,679 isolated capitals written by 412 different writers. The UNIPEN train\_r01\_v07 release contains a heterogeneous collection of characters, digits and words collected from writers from different countries of origin. From this collection, the '1b' subset contains 28,069 isolated uppercase characters and the '1d' subset contains 17,286 isolated symbols, from which 4833 samples containing ten classes were selected ('=', ':', ':', '!', '\$', '#', '%', '+', '?' and '\*'). Table 2 depicts the number of multi-stroke gestures selected from these three collections.

## 3. Features for multi-stroke recognition

From each of the four databases described above, three feature sets are computed, each on a different level of detail. The  $g$ -48 set contains 48 features computed from a global gesture trajectory.

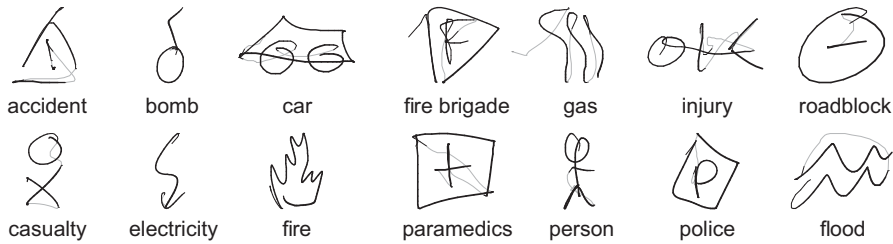


Fig. 2. Examples of fourteen different iconic gestures, produced by the same participant contributing to the Niclcon data collection. Penup strokes are depicted in light gray.

Table 1  
Distribution of the 23,641 gestures over the fourteen icon classes.

Description	Icon	# Samples	$\mu_s$	Description	Icon	# Samples	$\mu_s$
Accident		1831	5.9	Gas		1870	5.1
Bomb		1667	3.4	Injury		1862	7.4
Car		1842	5.9	Paramedics		1867	5.6
Casualty		1863	4.8	Person		1868	7.5
Electricity		1358	3.1	Police		1864	4.4
Fire		182	3.3	Roadblock		1839	3.1
Fire brigade		1858	7.2	Flood		1870	3.1

For each class, the average number of strokes  $\mu_s$  is given, counting both penup and penup strokes.

Table 2  
Statistics of the selected IRONOFF capitals and UNIPEN capitals and symbols.

Database	# Total	$\mu_s$	$\mu_{min}^s$	$\mu_{max}^s$
IRONOFF	4879	3.5	3.0 ('L', 'O')	4.7 ('I')
UpCaps	14 726	3.8	3.1 ('Z')	5.0 ('E')
UpSymbols	4471	4.0	3.2 ('+')	7.4 ('#')

Shown are the selected number of samples, average number of strokes (including both penup and penup) and the classes containing the lowest and highest average number of strokes ( $\mu_{min}^s, \mu_{max}^s$ ).

As we will argue in Section 3.2, these features cannot always distinguish between certain gesture classes, in particular if class separation requires a more detailed representation. The second set of features considers gestures at the stroke level and contains features computed from each stroke along the gesture trajectory, including the mean  $\mu$  and standard deviation  $\sigma$  of these feature values. At the finest level of detail, features are computed from each coordinate, as originally proposed by Guyon and LeCun in [21]. In the next subsections, we will describe these three feature sets: the g-48, the s- $\mu$ - $\sigma$ , and the c-n sets.

3.1. Global features: the g-48 feature set

As mentioned in the Introduction, most gesture recognition systems employ Rubine's thirteen features [6]. Among these features are the length and the angle of the bounding box diagonal, the distance between the first and the last point, the cosine and the sine of the angle between the first and the last point, the total gesture length, the total angle traversed, and the duration of the gesture. In [5], classification experiments on iconic gestures were presented which employed Rubine's features and an additional fifteen other global features (see Figs. 3 and 4 for some examples of these features). The classification accuracy using these 28 global features on the Niclcon database was significantly lower than when using features computed at the coordinate level from spatially resampled pen gestures. These findings corroborate other reports on using global

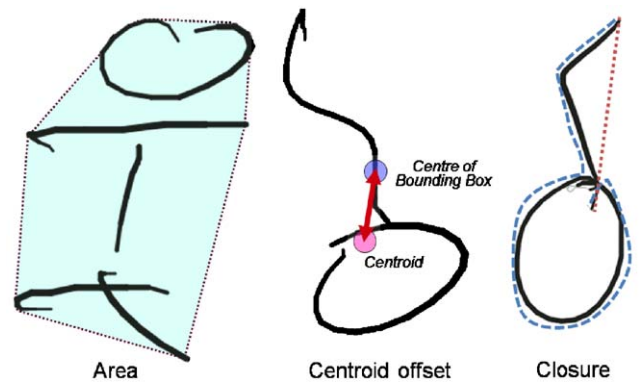


Fig. 3. Examples of g-48 features computed from the complete trajectory: area, centroid offset, and closure.

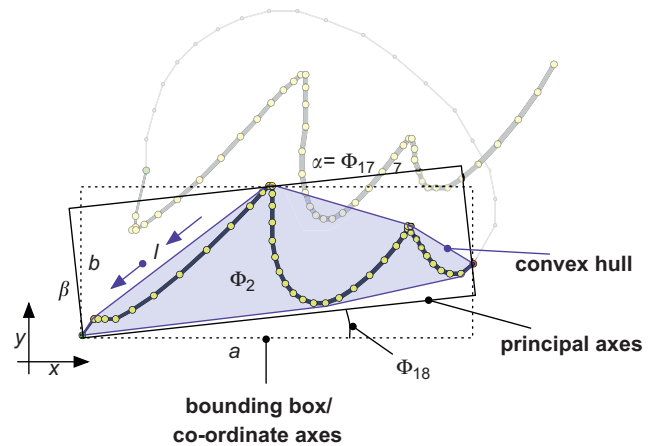
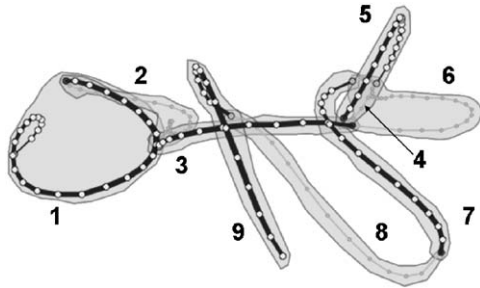


Fig. 4. The angle between the principal axis and x-axis determines the rotation angle. This figure depicts the g-48 features  $\Phi_2$ ,  $\Phi_{17}$  and  $\Phi_{18}$ : area of the convex hull and length and orientation of the principal axis.

features for gesture recognition [9,16] and indicate the need for improved feature representations.

As a result of our quest for more and better distinguishing features, we have recently compiled a survey on features for pen computing, which is available as a technical report [33]. We have



**Fig. 5.** Iconic gesture representing a casualty icon. Pendown strokes are odd numbered. The penup stroke '4' is marked with an arrow for distinguishing from stroke '6'. The new  $s$ - $\mu$ - $\sigma$  feature set contains features computed for each of the 9 strokes (segmented by transitions between pendown and penup). For each global feature  $f$  from the  $g$ -48 set,  $(\mu_f)$  and  $(\sigma_f)$  over all strokes were added as feature values. This was repeated for (i) all strokes, (ii) only the pendown strokes, and (iii) only the penup strokes.






included features from, e.g., [6,16,34–36]. The features described in [33] also contain local features computed along the coordinates from a gesture shape, such as chain code representations. These local features were excluded and for the current paper, we have selected 20 new global features, additional to the 28 features used in [5]. The resulting  $g$ -48 feature set is described in Appendix A. For details on the feature computation algorithms, the reader is referred to [33]. Table 7 in Appendix A depicts which of the  $g$ -48 features are rotation or size dependent. No features require both scaling and rotation. For features like area and trajectory length, size normalization is required. Other features, such as horizontal or vertical distance between samples, require rotation normalization. Scale normalization was performed by scaling a gesture's bounding box to unit size. Rotation was performed by aligning a gesture's principal axis to the  $x$ -axis, as depicted in Fig. 4.

### 3.2. Stroke-level features: the $s$ - $\mu$ - $\sigma$ feature set

In the current paper, we will compare the  $g$ -48 feature set to feature representations computed from each individual stroke. These stroke-level features comprise both the mean  $\mu$  and the standard deviation  $\sigma$  of the  $g$ -48 feature values computed over (i) all constituent strokes, (ii) penup strokes only, and (iii) pendown strokes only (see Fig. 5). Note that for the penup/pendown ratio and the pendown

**Table 3**

Examples for which classification using only global features ( $F_g$ ) yields a wrong result and classification with only stroke-level features yields the correct result.

Sample		$\bar{f}_c \pm \sigma(f_c)$	$f$	Offset	$\bar{f}_c \pm \sigma(f_c)$	$f$	Offset
		Length			Number of crossings		
 paramedics	$F_g$	$1.26 \pm 0.87$	8.42	8.27	$0.32 \pm 2.58$	12.54	4.74
	$F_m$	$0.45 \pm 0.82$	0.33	0.14	$0.19 \pm 2.34$	0.64	0.19
	$F_s$	$1.11 \pm 0.89$	1.05	0.08	$0.28 \pm 2.67$	1.88	0.60
		Length			Closure		
 gas	$F_g$	$0.20 \pm 0.61$	2.28	3.40	$0.89 \pm 0.53$	-0.24	2.12
	$F_m$	$-0.07 \pm 0.44$	0.30	0.85	$0.92 \pm 0.50$	0.60	0.64
	$F_s$	$-1.15 \pm 0.27$	-1.12	0.11	$-1.14 \pm 0.36$	-1.30	0.46
		Initial angle (sine)			Area		
 paramedics	$F_g$	$0.31 \pm 0.91$	-2.09	2.65	$1.52 \pm 0.76$	0.15	1.81
	$F_m$	$0.52 \pm 0.73$	0.97	0.62	$0.89 \pm 0.84$	1.45	0.66
	$F_s$	$-0.13 \pm 0.99$	-0.22	0.08	$1.34 \pm 0.69$	1.82	0.70
		Average Curvature			Rectangularity		
 firebrigade	$F_g$	$-0.49 \pm 0.66$	0.21	1.06	$-1.05 \pm 0.62$	0.32	2.20
	$F_m$	$0.34 \pm 1.21$	0.77	0.35	$-0.04 \pm 0.10$	-0.04	0.05
	$F_s$	$0.57 \pm 1.50$	0.45	0.08	$-0.04 \pm 0.11$	-0.04	0.05
		Absolute Curvature			Standard deviation pressure		
 firebrigade	$F_g$	$-0.77 \pm 0.64$	-0.12	1.01	$0.36 \pm 1.01$	1.57	1.19
	$F_m$	$-0.99 \pm 0.49$	-0.74	0.50	$0.04 \pm 1.04$	-0.12	0.15
	$F_s$	$0.06 \pm 0.72$	0.60	0.74	$0.55 \pm 1.10$	-0.06	0.55

For each sample the value for two features is given for the global feature ( $F_g$ ), for the mean feature value over the strokes ( $F_m$ ), and for the standard deviation value over the strokes ( $F_s$ ).  $\bar{f}_c$  and  $\sigma(f_c)$  denote the mean value and the value of the standard deviation of the feature over all samples in that class,  $f$  denotes the feature value of that sample, and 'Offset' denotes the offset of the feature value from the average feature value for the class in standard deviations for that class ( $\text{Offset} = |(f - \bar{f}_c) / \sigma(f_c)|$ ). If the offset has a high value, the feature value is an outlier for that class. Note that the feature values are normalized to a mean value of 0.0 and standard deviation of 1.0 over all samples from all classes.

count, the distinction in penup/pendown strokes is irrelevant. Care should be taken in cases where the pen is lifted too far from the tablet to be sampled as they may result in unreliable or incorrect feature values. For most tablets, such *penfar* events can be detected. However, since the amount of such cases is very limited (0.0% for the IRONOFF capitals, less than 0.1% for the UNIPEN capitals and symbols, and less than 0.2% for the Niclcon dataset) and since all feature values of the samples containing *penfar* events are within normal range, we decided to discard *penfar* events.

To understand why global features computed over the complete gesture shape cannot always properly distinguish between multi-stroke gestures, please consider Table 3. All these examples are correctly distinguished by the stroke-level features but not by the global features. As can be observed, these examples exhibit some specific characteristics making them harder to classify. In the first example (a), the participant made the cross bold, using multiple strokes, in the second example (b) the participant made a correction, in (c) the participant made a spurious double stroke on the left side, in (d) the icon is more rectangular than normal for this class (should be an elongated triangle), and in (e) the wrong box (a diamond instead of a triangle) was drawn.

Apparently, features computed at the level of individual strokes do allow classes to be discriminated in cases where the global features fail. These cases occur most often when gestures are written in a sloppy fashion, when writers make corrections, or when (parts of) a sample is re-drawn using multiple similar strokes.

The resulting feature set contains 758 features and is called the  $s$ - $\mu$ - $\sigma$  set. In Appendix A, a concise discussion is provided explaining how we arrived at this number of features. Although  $\mu$  and  $\sigma$  are very common measures in statistical pattern recognition (e.g., for estimating probability density functions or for regularization purposes), to our knowledge, the mean and standard deviation of features computed from sub-strokes of a gesture trajectory have not been used before as features for multi-stroke recognition.

### 3.3. Coordinate-level features: the $c$ -30 and the $c$ -60 feature sets

To assess the suitability of our new features, the  $s$ - $\mu$ - $\sigma$  feature set is compared to a set of features computed at the coordinate level. We use both the  $c$ -30 and the  $c$ -60 features, described in [24]. These are computed from gestures spatially resampled to  $n = 30$  or 60 coordinates. The  $c$ -30 features have extensively been used for character recognition. For each out of  $n$  points, the  $(x, y, z)$ -coordinates, the running angles and angular velocities are computed, resulting in  $3 \cdot n + 2 \cdot (n - 1) + 2 \cdot (n - 2)$  features. As explained in [24], a typical resampling of Western characters requires  $n = 30$  (204 features). Given that many of the collected iconic gestures have a more complex shape than the average Western character, we also explored resampling to  $n = 60$  (414 features), resulting in a better coverage of the original trajectory with resampled points.

## 4. Feature selection and classifier design

To explore the suitability of the 758 features from the  $s$ - $\mu$ - $\sigma$  feature set, the following method was employed. For each dataset, we computed the  $s$ - $\mu$ - $\sigma$  stroke-level features and the  $c$ -30 and  $c$ -60 coordinate features. All features were normalized through mean shifting [37] to a common scale with an average of zero and standard deviation of one. Second, eight different subsets were selected, as listed below:

Fg	containing the global features without $\mu$ and $\sigma$
Fm	containing only the $\mu$ features
Fs	containing only the $\sigma$ features
Fgm	containing the $g$ -subset with additional $\mu$ features
Fgs	containing the $g$ -subset with additional $\sigma$ features

Fgms	containing the $g$ -subset with additional $\mu$ and $\sigma$ features
Fc30	the coordinate-level features from trajectories resampled at $n = 30$
Fc60	the coordinate-level features from trajectories resampled at $n = 60$

The distinction between features with and without  $\mu$  and  $\sigma$  was made such that their individual contribution to recognition accuracy could be assessed. Third, each of these feature sets was divided into three subsets (36% train, 24% test, and 40% evaluation), using stratified random sampling. The fourth step in the comparison between features entailed feature selection using the BIN feature selection algorithm [23], which is described in Section 4.1. The final step in the feature assessment process used the selected features (through BIN) to design different classifiers based on the train and test sets. The description of the different classifiers involved is given in Section 4.2. The results of these elaborate explorations are presented in Section 5.

### 4.1. Feature selection from each feature subset

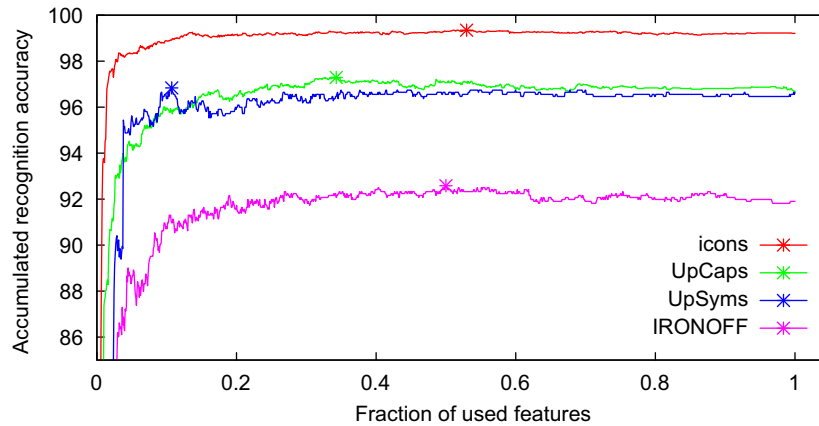
The BIN feature selection algorithm [23] was used to select a suitable subset of features from the first six feature sets (the stroke-level features). BIN tests the discriminative power of each of the features individually. The features are ranked according to the recognition performance for each individual feature using a support vector machine (SVM) classifier. The SVM was trained on each feature using the train set and the recognition performance was tested on the test set. Training the SVM was performed using the default parameter settings (see also Section 4.2.1). Like most other feature selection methods, the BIN method cannot guarantee to yield the best combination of features [38]. Furthermore, BIN does not explore linear or non-linear combinations of selected features, which in certain cases will lead to sub-optimal solutions [39]. Yet, we decided to use this method because it is relatively straight-forward, efficient and can be used to discard a relevant subset of poor features [38].

We used BIN feature selection for each dataset and each of the stroke-level feature sets, resulting in 24 feature rankings. For each ranking, the ‘accumulated’ recognition performance was computed by training and testing an SVM classifier on  $r$  features, where  $r$  is varied from 1 (the feature with the highest rank) to the number of features. As an example, consider Fig. 6. For each ranking, the fraction of features yielding the maximum accumulated recognition performance was determined. This fraction was subsequently used for further experiments and the remaining features were discarded. Each fraction is reported in Table 4, and ranges from 0.11 (for Fgms) to 0.99 (for Fg), both for the symbol dataset. On average, the fraction of features at which maximum performance was achieved was 0.5 ( $\sigma = 0.2$ ).

Another result from the BIN feature selection process concerns the relative amount of features from the categories  $g$ -48,  $\mu$ , and  $\sigma$ . For each combination of dataset and the three feature sets Fgm, Fgs, and Fgms, the relative amount of selected features from the corresponding BIN experiments was computed. Averaged over all datasets, the ratios ( $g$ -48/ $\mu$ / $\sigma$ ) are (0.48/0.52/0.0) for Fgm, (0.49/0.0/0.51) for Fgs, and (0.35/0.37/0.28) for the Fgms features, respectively. These results indicate that, according to the BIN method, each of the  $g$ -48,  $\mu$ , and  $\sigma$  features provides a similar contribution to recognition performance.

### 4.2. Classifier design and learning

Three classifiers (multi-layered perceptron (MLP), SVM, and DTW) were used for generating recognition scores on the various feature sets computed from each database. Each classifier used the train set and test set for learning and tuning of control parameters. Only



**Fig. 6.** Recognition scores on the Fgms features of SVMs trained with varying number of features ranked according to BIN. The maximum scores are indicated with starred dots and correspond to the highest recognition score, achieved with a limited number of features. As can be observed, incorporating additional features beyond these points does not improve performance.

**Table 4**

Classification results for each of the feature subsets, using SVM and MLP classifiers.

Group	Icons			IRONOFF			UpSym			UpCaps		
	MLP	SVM	$f$	MLP	SVM	$f$	MLP	SVM	$f$	MLP	SVM	$f$
Fg	97.3	98.7	0.47	88.3	90.8	0.61	94.8	95.4	0.99	94.4	95.5	0.70
Fm	97.0	98.2	0.62	88.1	89.3	0.42	94.4	95.2	0.64	91.9	93.8	0.27
Fs	96.4	97.7	0.55	88.5	89.0	0.26	93.2	94.4	0.69	92.1	93.5	0.47
Fgm	98.3	99.2	0.37	90.3	91.5	0.59	95.3	96.1	0.86	94.3	95.7	0.34
Fgs	97.9	99.1	0.35	91.3	92.8	0.55	94.9	96.3	0.58	94.2	96.4	0.55
Fgms	98.7	99.2	0.53	91.9	92.9	0.50	95.4	96.4	0.11	95.1	96.5	0.34

For each dataset and feature subset, the fraction  $f$  of features at which maximal BIN performance is achieved (on the testset) is depicted. The results show a consistent improvement when adding  $\mu$  and  $\sigma$  features.

after this optimization process, the classification performance on the evaluation set was used as the evaluation measure.

#### 4.2.1. Feature-based classification using MLP and SVM

Two common feature-based classifiers were used: the MLP and the SVM [40]. The MLP neural network implementation uses the generalized delta rule with momentum. The parameters varied for the MLP were learning rate, momentum, and number of hidden units. Training each MLP was performed until performance on the test set reached a maximum performance, as determined through cross-validation. We used LIBSVM [41], public domain software implementing a multi-class SVM-classifier (C-SVC). Besides the traditional linear kernel, non-linear kernels were employed in order to achieve the highest possible classification performance. We tested polynomial, radial basis function and sigmoid kernels. Each of these kernels has their own parameters which we varied:  $\gamma$  for all non-linear kernels and  $\text{degree}$  and  $\text{coef0}$  for the polynomial and the sigmoid kernel. Additionally, for all kernels, we tried different cost parameters  $C$ .

#### 4.2.2. Template matching using DTW

The dynamic time warping (DTW) algorithm described in [25] computes the DTW distance between two data samples by summing the normalized Euclidean distances between the matching coordinates of a known prototypical data sample  $A$  and an unknown sample  $B$ . For the experiments reported in this paper, the DTW classifier uses all training samples as prototypes. Whether two coordinates  $A_i$  and  $B_j$  match is decided using three conditions: (i) the continuity condition, which is satisfied when index  $i$  is on the same relative position on  $A$  as index  $j$  is on  $B$  (the amount in which the relative

positions are allowed to differ is controlled by a parameter  $c$ ), (ii) the boundary condition, which is satisfied if both  $i$  and  $j$  are at the first, or at the last position of their sample, (iii) the penup/pendown condition, which is satisfied when both  $i$  and  $j$  are produced with the pen on the tablet, or when they are both produced with the pen above the tablet.  $A_i$  and  $B_j$  match if either the boundary condition, or both other conditions are satisfied. Classification of a test sample is performed through nearest neighbour matching with the DTW distance function. Each DTW classifier was optimized by varying parameter  $c$ , which controls the strictness of the continuity condition.

## 5. Results

In this section the results of our feature assessments on four databases containing multi-stroke gestures are presented. First, in Section 5.1, the classification results on the 24 feature subsets derived from the  $s$ - $\mu$ - $\sigma$  features are discussed. Subsequently, in Section 5.2, these results are compared to the results achieved with DTW and with the classifiers trained on the  $c$ -30 and  $c$ -60 features.

### 5.1. Evaluation of feature subsets computed from the $s$ - $\mu$ - $\sigma$ features

For each feature subset and database, a SVM and an MLP classifier were optimized following the method described in the previous section. Table 4 shows the corresponding classification results on the evaluation datasets, containing the remaining 40% of the data.

As can be expected, the SVM classifier outperforms the MLP. Compared to the global features Fg, adding mean and standard deviation features computed at the stroke level improves classification accuracy. The results are consistent over different databases and both

**Table 5**

Performance comparison between the Fgms features, the coordinate features c-30 and c-60 and the DTW classifier.

Database	Fc30		Fc60		DTW	Fgms	
	MLP	SVM	MLP	SVM		MLP	SVM
Icons	96.2	97.0	95.9	96.8	98.5	98.7	99.2
IRONOFF	88.6	89.9	88.4	89.5	93.5	91.9	92.8
UpSym	92.6	93.3	93.1	94.1	94.0	95.4	96.4
UpCaps	94.3	95.4	95.1	95.6	95.5	95.1	96.4

**Table 6**

The notation and definitions used in Table 7.

	Notation
Unit vectors ( $x$ - and $y$ -axes) spanning $\mathbb{R}^2$	$\mathbf{e}_1 = (1, 0)$ and $\mathbf{e}_2 = (0, 1)$
Pen trajectory with $N$ sample points	$\mathcal{T} = \{\sigma_1, \sigma_2, \dots, \sigma_N\}$
Sample	$\sigma_i = \{\mathbf{s}_i, f_i, t_i\}$
Position	$\mathbf{s}_i = (x_i, y_i)$
Area of the convex hull	$A$
Angle between subsequent segments	$\psi_{s_n} = \arccos\left(\frac{(\mathbf{s}_n - \mathbf{s}_{n-1}) \cdot (\mathbf{s}_{n+1} - \mathbf{s}_n)}{\ \mathbf{s}_n - \mathbf{s}_{n-1}\  \ \mathbf{s}_{n+1} - \mathbf{s}_n\ }\right)$
Length along the $x$ -axis	$a = \max_{1 \leq i < j \leq N}  x_i - x_j $
Length along the $y$ -axis	$b = \max_{1 \leq i < j \leq N}  y_i - y_j $
Center of the bounding box	$\mathbf{c} = \left(x_{\min} + \frac{1}{2}(x_{\max} - x_{\min}), y_{\min} + \frac{1}{2}(y_{\max} - y_{\min})\right)$
Longest edge-length of the bounding box	$a' = a$ if $a > b$ else $a' = b$
Shortest edge-length of the bounding box	$b' = b$ if $a > b$ else $b' = a$
Lowest $x$ value	$x_{\min} = \min_{1 \leq i \leq N} x_i = \min_{1 \leq i \leq N} (\mathbf{s}_i \cdot \mathbf{e}_1)$
Lowest $y$ value	$y_{\min} = \min_{1 \leq i \leq N} y_i = \min_{1 \leq i \leq N} (\mathbf{s}_i \cdot \mathbf{e}_2)$
Principal components	$\mathbf{p}_i$
Angle of first principal axis	$\psi = \arctan \frac{\mathbf{p}_1 \cdot \mathbf{e}_2}{\mathbf{p}_1 \cdot \mathbf{e}_1}$
Length of first principal axis	$\alpha = 2 \max_{0 \leq n < N}  \mathbf{p}_2 \cdot (\mathbf{c} - \mathbf{s}_n) $
Length of second principal axis	$\beta = 2 \max_{0 \leq n < N}  \mathbf{p}_1 \cdot (\mathbf{c} - \mathbf{s}_n) $
Centroid	$\boldsymbol{\mu} = \frac{1}{N} \sum_n \mathbf{s}_n$
Velocity	$\mathbf{v}_i = \frac{\mathbf{s}_{i+1} - \mathbf{s}_{i-1}}{t_{i+1} - t_{i-1}}$
Acceleration	$\mathbf{a}_i = \frac{\mathbf{v}_{i+1} - \mathbf{v}_{i-1}}{t_{i+1} - t_{i-1}}$

classifiers. The best performances are achieved when using features from all three feature sets. Relative improvements in error rates when comparing the Fgms and Fg features range from 10% (for the MLP classifiers on the UNIPEN data) to 40% and 50% for the SVM and MLP on the IRONOFF database. Averaged over the databases and all classifiers, the improvement between Fgms and Fg is 25%.

The Fg, Fm, and Fs features achieve comparable recognition rates. This is in accordance with the observations from Section 4, where it was observed that the fractions of the global features and stroke-based mean and standard deviation features are similar. Considering the point at which maximum BIN performances are achieved (fraction  $f$ ) as listed in Table 4, no definite conclusions can be drawn. When averaging over the four datasets,  $f$  decreases from 0.69 (for Fg features) to 0.37 (for Fgms), but this is mostly due to the remarkable drop from 0.99 to 0.11 (for the UNIPEN symbols).

## 5.2. Comparison to other multi-stroke recognition techniques

Table 5 depicts the classification results of the MLP and SVM classifiers, optimized on the c-30 and c-60 features. Furthermore, the results from the DTW classifier are presented, using the complete trainset as prototype set. For convenience to the reader, we have included the results on the Fgms features from Table 4.

For both the SVM and the MLP types of classifiers, significant improvements are observed between the Fc30 and the Fc60 coordinate features and the Fgms features. Error rates drop between 0% (for the UNIPEN capitals) and 80%. Averaged over the four databases and all classifiers, the improvement is 39%. Comparing the results of the SVM classifiers trained on the Fgms features to DTW, the improvement is 25%, averaged over all databases. It should be noted that the DTW classifier employs all training samples as prototypes for matching. Allograph matching techniques like DTW in most cases employ a significantly lower amount of prototypes, e.g. obtained through hierarchical clustering [24]. This implies that the DTW classification results presented in Table 5 should be considered as an upper bound.

The performance on the IRONOFF dataset is much lower than the performance reported for the other three databases. This effect is consistent for all classifiers, feature representations, and feature subsets. Our explanation for this effect is that the isolated capitals from the IRONOFF database contain only one character class sample per writer, which makes the IRONOFF recognition results per definition writer-independent. For the other databases, more data samples are available per writer.

Nevertheless, when comparing the obtained recognition rates to reported performances from the literature, our achieved performances on UNIPEN and IRONOFF capitals are competitive. It should be noted that comparisons to the literature are hard to make, since we have excluded single-stroked gestures from these data, which in general are easier to recognize. Furthermore, the ratio between the available amount of training data versus the amount of evaluation data and the distribution of samples over the different data subsets may differ from experiments reported elsewhere.

## 6. Discussion and future research

Inspired by our research on pen-centric interactive map technologies, this paper focuses on the design and evaluation of feature sets and classifiers for multi-stroke pen gesture recognition. We have implemented and evaluated an elaborate set of 48 global features, compiled from various works from the literature. By computing these features on each constituent penup and pendown stroke along a gesture trajectory, additional mean and standard deviation features were devised. Through different normalizations on size and rotation, a large feature vector of 758 features was constructed. Feature selection using the BIN method was employed on features computed from four publicly available databases: the Niclcon collection of iconic pen gestures, the UNIPEN and IRONOFF uppercase characters, and a subset from the UNIPEN symbols category.

Several configurations of selected feature subsets were assessed on recognition accuracy, using different classifiers to compute recognition performances. The BIN feature selection method appeared to be very successful in selecting subsets of features. A particularly interesting conclusion to draw is that a major part of the selected features (about  $\frac{2}{3}$ ) comprise our new mean and standard deviation features. This implies that according to BIN, global features are equally important as  $\mu$  and  $\sigma$ .





The impact on recognition performance is significant: the new features yield an improvement between 10% and 50% compared to the global features. Furthermore, compared to the DTW trajectory-matching technique and to local features computed at the level of coordinate sequences, improvements between 25% and 39%, averaged over all four databases, were achieved.

We are currently further developing and assessing our pen input recognition technologies in more elaborate experiments, involving pen input data acquired in more realistic situations. As the envisaged context is emergency service personnel, working in stressful circumstances, we are considering experimental conditions including drawing from memory, drawing under pressure of time, or drawing in multi-task settings. Finally, given the large impact of our new features on recognition performance, we hope to achieve similar improvements for other tasks, like Asian or Arabic character recognition. Our results may also translate to other application areas where mean and standard deviations of features computed from sub-segments may prove to be successful.

## Acknowledgments

This research was supported by the Dutch Ministry of Economic Affairs, Grant no. BSIK03024 and the Netherlands Organization for Scientific Research (NWO), Project no. 634.000.434.

## Appendix A. Feature descriptions

In this appendix, the 48  $g$ -48 features are described. These features comprise a selection from the features described in our technical report [33]. The  $g$ -48 features contain purely global features computed from a complete gesture trajectory and are listed in Table 7. Some of the feature descriptions contained in Table 7 use the notation and definitions specified in Table 6.

Included in Table 7 are the equations used to calculate the features  $\Phi_i$  used in our technical report. Since this technical report also contains some features computed along the running trajectory, like chaincode information, some feature indices extend over 48 (like  $\Phi_{60}$ ). From each of the 48 plain features, various corresponding derived feature values were added. The column marked  $N_i$  specifies how many features were derived from a feature  $\Phi_i$ .  $N_i$  is computed as follows:

$$N_i = co_i \cdot (1 + 2 \cdot \mu\sigma_i) \cdot (1 + 2 \cdot pud_i) \cdot 2^{ns_i + nr_i}$$

where  $\mu\sigma_i$ ,  $pud_i$ ,  $ns_i$ , and  $nr_i$  can be considered as booleans with values  $\{0, 1\}$ . These booleans indicate, respectively, whether (i) the stroke-based mean and standard deviations can be computed, (ii) whether pressure-based information regarding penup and pendown can be determined, (iii) whether  $\Phi_i$  depends on size normalization or (iv) depends on rotation normalization. The parameter  $co_i$  indicates how many coordinates are required to represent a feature. For example, for angular information  $co_i$  is 2, represented by  $\sin(\phi)$  and  $\cos(\phi)$ . Certain features cannot be explained in one single equation. For these features, the reader is referred to [33].

## References

- [1] F. Tian, T. Cheng, H. Wang, G. Dai, Advances in Computer Graphics, in: Lecture Notes in Computer Science, vol. 4035/2006, Springer, Berlin/Heidelberg, 2006, Ch. Research on User-Centered Design and Recognition of Pen Gestures, pp. 312–323.
- [2] S. Fitriani, L. Rothkrantz, A visual communication language for crisis management, International Journal of Intelligent Control and Systems (Special Issue of Distributed Intelligent Systems) 12 (2) (2007) 208–216.
- [3] D. Willems, L. Vuurpijl, Designing interactive maps for crisis management, in: Proceedings of the Fourth International Conference on Information Systems for Crisis Response and Management (ISCRAM2007), 2007, pp. 159–166.
- [4] D. Willems, L. Vuurpijl, Pen gestures in online map and photograph annotation tasks, in: Proceedings of the Tenth International Workshop on Frontiers in Handwriting Recognition (IWFHR06), La Baule, France, 2006, pp. 297–402.
- [5] R. Niels, D. Willems, L. Vuurpijl, The Niclcon collection of handwritten icons, in: ICFHR8, the Eleventh International Conference on Frontiers of Handwriting Recognition, Montreal, Canada, 2008, pp. 296–301.
- [6] D. Rubine, Specifying gestures by example, Computer Graphics 25 (4) (1991) 329–337.
- [7] P.R. Cohen, M. Johnston, D. McGee, S. Oviatt, J. Pittman, I. Smith, L. Chen, J. Clow, Quickset: multimodal interaction for distributed applications, in: MULTIMEDIA '97: Proceedings of the Fifth ACM International Conference on Multimedia, ACM, NY, USA, 1997, pp. 31–40.
- [8] J.A. Landay, R.B. Dannenberg, Interactive sketching for the early stages of user interface design, in: CHI95 Computer Human Interaction, 1995, pp. 43–50.
- [9] B. Signer, U. Kurmann, M. Norrie, Igesture: a general gesture recognition framework, in: Proceedings of the Ninth International Conference on Document Analysis and Recognition (ICDAR2007), Curitiba, Brazil, 2007, pp. 954–958.
- [10] J. Lipscomb, A trainable gesture recognizer, Pattern Recognition 24 (9) (1991) 895–907.
- [11] M. Fonseca, J. Jorge, Experimental evaluation of an on-line scribble recognizer, Pattern Recognition Letters 22 (12) (2001) 1311–1319.
- [12] M. Egger, Find new meaning in your ink with tablet PC APIs in Windows Vista, Technical Report, Microsoft Corporation, May 2006.
- [13] J. Hong, J. Landay, Satin: a toolkit for informal ink-based applications, in: UIST00, Thirteenth Annual ACM Symposium on User Interface Software and Technology, San Diego, USA, 2000, pp. 63–72.
- [14] C. Blickenstorfer, Graffiti: Wow! Pen Computing Magazine, 1995, pp. 30–31.
- [15] D. Goldberg, C. Richardson, Touch-typing with a stylus, in: CHI '93: Proceedings of the INTERACT '93 and CHI '93 Conference on Human Factors in Computing Systems, 1993, pp. 80–87.
- [16] L. Zhang, Z. Sun, An experimental comparison of machine learning for adaptive sketch recognition, Applied Mathematics and Computation 185 (2) (2007) 1138–1148.
- [17] I. Guyon, L. Schomaker, R. Plamondon, M. Liberman, S. Janet, UNIPEN project of on-line data exchange and recognizer benchmarks, in: Proceedings ICPR'94, 1994, pp. 29–33.
- [18] C. Viard-Gaudin, P.M. Lallican, P. Binter, S. Knerr, The IRESTE on/off (IRONOFF) dual handwriting database, in: Proceedings of the International Conference on Document Analysis and Recognition, ICDAR'99, Bangalore, India, 1999, pp. 455–458.
- [19] W. Jiang, Z.-X. Sun, HMM-based on-line multi-stroke sketch recognition, in: Machine Learning and Cybernetics, 2005, Lecture Notes in Computer Science, vol. 7, Springer, Berlin, 2005, pp. 4564–4570.
- [20] T. Sezgin, R. Davis, Sketch recognition in interspersed drawings using time-based graphical models, Computers & Graphics 32 (2008) 500–510.
- [21] I. Guyon, P. Albrecht, Y. Le Cun, J. Denker, W. Hubbard, Design of a neural network character recognizer for a touch terminal, Pattern Recognition 24 (2) (1991) 105–119.
- [22] V. Vuori, E. Oja, J. Kangas, Experiments with adaptation strategies for a prototype-based recognition system for isolated handwritten characters, International Journal on Document Analysis and Recognition 3 (2001) 150–159.
- [23] A. Webb, Statistical Pattern Recognition, second ed., Wiley, 2002 (Chapter 9. Feature selection and extraction, pp. 305–359).
- [24] L. Vuurpijl, L. Schomaker, Finding structure in diversity: a hierarchical clustering method for the categorization of alphagrams in handwriting, in: Proceedings of ICDAR4, IEEE Computer Society, 1997, pp. 387–393.
- [25] R. Niels, L. Vuurpijl, L. Schomaker, Automatic allograph matching in forensic writer identification, International Journal of Pattern Recognition and Artificial Intelligence 21 (1) (2007) 61–81.
- [26] Homeland Security Working Group, Symbology Reference, Version 2.20, Released September 14, 2005 (<http://www.fgdc.gov/HSWG>).
- [27] U. Marti, H. Bunke, A full English sentence database for off-line handwriting recognition, in: Proceedings of the Fifth International Conference on Document Analysis and Recognition (ICDAR'99), Bangalore, India, 1999, pp. 705–708.
- [28] S. Jaeger, M. Nakagawa, Two on-line Japanese character databases in Unipen format, in: Proceedings of the International Conference on Document Analysis and Recognition, ICDAR'01, IEEE Computer Society, 2001, pp. 566–570.
- [29] M. Nakagawa, K. Matsumoto, Collection of on-line handwritten Japanese character pattern databases and their analysis, International Journal on Document Analysis and Recognition 7 (1) (2004) 69–81.
- [30] A.S. Bhaskarabhatla, S. Madhvanath, Experiences in collection of handwriting data for online handwriting recognition in Indic scripts, in: Proceedings of the Fourth International Conference on Linguistic Resources and Evaluation (LREC), 2004, CDROM.
- [31] U. Bhattacharya, Handwritten character databases of Indic scripts, 2004 (<http://www.isical.ac.in/~ujjwal/download/database.html>).
- [32] M. Pechwitz, S. Maddouri, V. Märgner, N. Ellouze, H. Amiri, Ifn/enit-database of handwritten Arabic words, in: Seventh Colloque International Francophone sur l'Écrit et le Document (CIFED02), Hammamet, Tunisia, 2002, pp. 1–8.
- [33] D. Willems, R. Niels, Definitions for features used in online pen gesture recognition, Technical Report, NICI, Radboud University Nijmegen, 2008 (<http://unipen.nici.ru.nl/Niclcon/>).
- [34] J. Iivarinen, M. Peura, J. Säreä, A. Visa, Comparison of combined shape descriptors for irregular objects, in: A. Clark (Ed.), Eighth British Machine Vision Conference, BMVC'97, Essex, UK, 1997, pp. 430–439.
- [35] M. Peura, J. Iivarinen, Efficiency of simple shape descriptors, in: L. Arcelli, C. Cordella, G. Sanniti di Baja (Eds.), Advances in Visual Form Analysis, World Scientific, Singapore, 1997, pp. 443–451.

- [36] J. LaViola, J.J.R. Zeleznik, A practical approach for writer-dependent symbol recognition using a writer-independent symbol recognizer, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29 (11) (2007) 1917–1926.
- [37] K. Fukunaga, L. Hostetler, The estimation of the gradient of a density function, with applications in pattern recognition, *IEEE Transactions on Information Theory* 21 (1975) 32–40.
- [38] A. Jain, R. Duin, J. Mao, Statistical pattern recognition: a review, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22 (1) (2000) 4–37.
- [39] S. Piramuthu, Evaluating feature selection methods for learning in data mining applications, *European Journal of Operational Research* 156 (2004) 483–494.
- [40] V. Vapnik, *The Nature of Statistical Learning Theory*, Springer, Berlin, 1995.
- [41] C.-C. Chang, C.-J. Lin, *LIBSVM: a library for support vector machines*, 2001 ([www.csie.ntu.edu.tw/~cjlin/libsvm](http://www.csie.ntu.edu.tw/~cjlin/libsvm)).

**About the author**—DON WILLEMS received his M.Sc. degree in Cognitive Science in 2003 at the Radboud University in Nijmegen in The Netherlands. From 2000 to 2004 he was a scientific software developer at the Max Planck Institute for Psycholinguistics. Since April 2004 he has been working as a researcher and a Ph.D. student at the Nijmegen Institute for Cognition and Information, where he is involved in research on pen gesture recognition systems for crisis management applications. He joined the Agrotechnology and Food Sciences Group at Wageningen University in September 2008 where he is working on knowledge intensive processes in the agrifood domain. His research interests focus mainly on the application of techniques from artificial intelligence and pattern recognition on human computer interaction.

**About the author**—RALPH NIELS received his M.Sc. degree in Artificial Intelligence from the Radboud University Nijmegen, The Netherlands, in 2004. His master thesis was about the use of Dynamic Time Warping for intuitive handwriting recognition. After his graduation, he joined the cognitive artificial intelligence group of the Nijmegen Institute of Cognition and Information as a Ph.D. student. His thesis, which is planned for 2009, focuses on the use of allographic information for forensic writer identification.

**About the author**—MARCEL VAN GERVEN received an M.Sc. degree in Cognitive Science and a Ph.D. degree in Computer Science both from the Radboud University Nijmegen. His Ph.D. studies dealt with the use of (dynamic) Bayesian networks for prognosis in clinical oncology and was partly conducted at the Dutch Cancer Institute and at the UNED in Madrid, Spain. Currently, Marcel is working as a postdoctoral researcher in Brain–Computer Interfacing (BCI) at the Institute for Computing and Information Science and the F.C. Donders Centre for Cognitive Neuroimaging; both are located at the Radboud University Nijmegen, The Netherlands. His research focuses mainly on the development of classification algorithms for BCI and on the analysis of cognitive neuroimaging data in general.

**About the author**—LOUIS VUURPIJL received his Ph.D. in computer science in 1998 for research on neural networks and parallel processing. He has been involved in various forms of image processing and neural network-based image recognition such as the detection of ground-cover classes in satellite imagery. Louis Vuurpijl has been affiliated with the NICI since 1995, conducting research on pen computing, image retrieval, online handwriting recognition, forensic document analysis, and multimodal interaction. He is an assistant professor and lectures on artificial intelligence, robotics, and cognitive science. Louis Vuurpijl is member of the board of the international Unipen Foundation and is involved in several national and European research projects.