MCS for Online Mode Detection: Evaluation on Pen-Enabled Multi-Touch Interfaces

Markus Weber*†, Marcus Liwicki*, Yannik T. H. Schelske†, Christopher Schoelzel†, Florian Strauß†, Andreas Dengel*†
* *German Research Center for AI (DFKI GmbH)*
Knowledge Management Department,
Kaiserslautern, Germany
{firstname.lastname}@dfki.de
† *Knowledge-Based Systems Group,*
Department of Computer Science, University of Kaiserslautern,
P.O. Box 3049, 67653 Kaiserslautern

***Abstract*—This paper proposes a new approach for drawing mode detection in online handwriting. The system classifies groups of ink traces into several categories. The main contributions of this work are as follows. First, we improve and optimize several state-of-the-art recognizers by adding new features and applying feature selections. Second, we use several classifiers for the recognition. Third, we perform multiple classifier combination strategies for combining the outputs. Finally, a large experimental evaluation on two data sets is performed: the publicly available Touch & Write database which has been acquired on a pen-enabled multi-touch surface; and the publicly available IAMonDo-database which serves as a benchmark. In our experiments on the IAM-OnDo-database we achieved a recognition rate of 97 %, which is much higher than other results reported in the literature. On the more balanced multi-touch surface data set we achieved a recognition rate of close to 98 %.**

***Keywords*-online mode detection, multiple classifier combination, multitouch, touch & write**

I. Introduction

Drawing mode detection is the task to automatically detect the mode of online handwritten strokes. Instead of having the user of a real-time pen-enabled interface to switch manually between handwriting recognition, shape detection, and gesture recognition, the mode-detection system should be able to guess the user's intention based on the strokes themselves. For example, mode detection should be able to determine whether a user is producing deictic gestures (e.g. to mark an object on a map or to specify a route), handwritten text, or iconic object drawings (people, cars, etc.) [1].

The main motivation of this paper is pen-enabled multi-touch interfaces. A mayor trend in human computer interaction is the emerge of surfaces which enable the user to directly interact with the information presented on the screen via interaction. After decades of research as summarized in [2] nowadays several interfaces are available, such as mobile touch phones, the iPad, and the Microsoft Surface. Some of these interfaces allow for pen-based interaction besides touch or multi-touch. The device of choice in this work is the Touch & Write [3], which inherently distinguishes between touching and pen and touch interaction.

Mode detection can be seen as a special case of text/graphics segmentation for online documents [4], [5]. Instead of analyzing the online document as a whole, the handwritten strokes are analyzed shortly after putting them on the surface. This can be considered as a more difficult task because of several reasons. First, there is no information about the context of the other handwritten strokes, especially those entered in the future. Second, drawing modes typically increase the amount of classes to be distinguished from, i.e., besides the classes of text and graphics also gestures are possible. Finally, the result of drawing mode detection should be available in real-time, i.e., the processing time should be only a few milliseconds. However, despite these differences mode detection systems can be evaluated on online document databases, as done in this paper, in order to make detection results comparable to other work.

Research work which is related to this paper has been performed in the area of mode detection and online document analysis. Willems et al. [6] proposed a set of features for the direct task of mode detection. They improved their methods in [1] where also more sophisticated classifiers are used. In this work we also use features of Willems' approach in some of our systems. However, we investigate a larger set of features as well as more classifiers.

Methods for text/graphics segmentation in online documents have been proposed in [4], [5]. Further studies have been performed in [7], where the first database of online handwritten documents with different content types was made available to the public. In this paper we compare our approach to the methods of [4] and [7].

Note that a simple system for mode detection for Touch & Write surfaces has already been presented in [8]. However, the system in [8] just used the features of [1] and applied a simple nearest neighbor approach for classification. In this paper we investigate several features and classifiers for mode detection. Furthermore we combine all systems in a multiple classifier systems. Finally, the results are compared to state-of-the-art approaches on a publicly available benchmark database.

Table I
ONLINE FEATURES

ID	Feature	Description	Note
0	Number of Strokes	N	
1	Length	$\lambda = \sum_{i=0}^{n-1} \|\|\vec{s}_i - \vec{s}_{i+1}\|\|$	s_i denotes a sample.
2	Area	A	
3	Perimeter Length	λ_c	Length of the path around the convex hull.
4	Compactess	$c = \frac{\lambda_c^2}{A}$	
5	Eccentricity	$e = \sqrt{1 - \frac{b^2}{a^2}}$	a and b denote the length of the major respectively minor axis of the convex hull.
6	Principal Axes	$e_r = \frac{b}{a}$	
7	Circular Variance	$v_c = \frac{1}{n\mu_r^2} \sum_{i=0}^{n} (\|\|\vec{s}_i - \vec{\mu}\|\| - \mu_r)^2$	μ_r denotes the mean distance of the samples to the centroid μ.
8	Rectangularity	$r = \frac{A}{ab}$	
9	Closure	$c_l = \frac{\|\|\vec{s}_0 - \vec{s}_n\|\|}{\lambda}$	
10	Curvature	$\kappa = \sum_{i=1}^{n-1} \psi_{\vec{s}_i}$	ψ_{s_i} denotes the angle between the segments $\overline{s_{i-1}s_i}$ and $\overline{s_i s_{i+1}}$ at s_i.
11	Perpendicularity	$p_c = \sum_{i=1}^{n-1} \sin\left(\psi_{\vec{s}_i}\right)^2$	
12	Signed Perpendicularity	$p_{sc} = \sum_{i=1}^{n-1} \sin\left(\psi_{\vec{s}_i}\right)^3$	
13	Angles after Equidistant Resampling (6 line segments)	$\sin(\alpha), \cos(\alpha)$	The five angles between succeeding lines are considered to make the features scale and rotation invariant.

Table II
OFFLINE FEATURES

ID	Feature	Description	Note
14	Number of Strokes	N	Same value as in the online version
15	Accumulated length	λ	Same value as in the online version
16	heigth of bounding box	$h_{bb} = y_{max} - y_{min}$	$y_{max/min}$ is the maximum/minimum value on the y-axis that is touched by the sample.
17	width of bounding box	$w_{bb} = x_{max} - x_{min}$	
18	ratio between width and height of bounding box	w_{bb}/h_{bb}	
19	area of bounding box	$w_{bb} * h_{bb}$	
20	number of connected components	n_{cc}	Determined after connected component analysis
21	average area of bounding box of connected components	$\frac{1}{n_{cc}} \sum_{i=1}^{n_{cc}} \left(c_{xmax}^{(i)} - c_{xmin}^{(i)}\right) \cdot \left(c_{ymax}^{(i)} - c_{ymin}^{(i)}\right)$	$c_{xmax}^{(i)}$ is the maximum value on the x-axis that is touched by connected component nr i - analogue for min and y.

The rest of this paper is organized as follows. First, Section II describes the several classifiers which have been used for our multiple classifier approach. Next, Section III summarizes the approaches used for multiple classifier combination. Subsequently, Section IV reports on the experiments performed on two databases. Finally, Section V concludes the paper..

II. CLASSIFIER DESCRIPTION

As mentioned above, several systems have been used in the work described in this paper. These systems differ in the way of feature extraction and classification. This section describes the features and classifiers used for our multiple classifier system.

The input of the classifier is a trace consisting of several stroke sequences (A stroke is the sequence of points between a pen-down and the next pen-up movement). This trace has been acquired by an online acquisition device, i.e., the Anoto[1] pen which works on a multi-touch surface. In the real-time environment the recognition of a trace is triggered 500 milliseconds after the last pen-up movement.

We have extracted two sets of features out of the traces, online features and offline features. The online features where extracted directly from the stroke sequence. The offline features where extracted from a pseudo-offline version of the trace derived by connecting succeeding points of the point sequence of each stroke.

Our online features are basically taken from the features set proposed in Willems et al. [6]. A complete list of these features appear in Table I.

In order to derive the offline features the online-data was converted to a binary image of at most 1000×1000 pixels by drawing all the strokes with a 3 pixel wide pen. Subsequently

[1] www.anoto.com - last visited 2011-03-14

Table III
OPTIMIZED k FOR EACH DISTANCE MEASURES FOR THE KNN

Distance Measure	definition	k
Mahalanobis Distance	$(u-v)\frac{1}{V}(u-v)^T$	5
Cosine Distance	$\frac{1-uv^T}{\|\|u\|\|_2\|\|v\|\|_2}$	5
Standardized Euclidean Distance	$\sqrt{\sum_i \frac{(u_i-v_i)^2}{V}}$	5
Cityblock Distance	$\|\|u-v\|\|_1$	3
Canberra Distance	$\sum_i \frac{u_i-v_i}{\|u_i\|+\|v_i\|}$	3

Table IV
META-PARAMETERS OF THE SVM

Kernel Function	Cost	Degree
Polynomial	15	3
Radial Basis	15	5
Linear	15	1
Sigmoid	15	7

the images were normalized by putting the center of gravity at the center of the image and aligning the major axis with the y-axis to achieve rotation invariance. A complete list of these features appear in Table II.

The IDs given in Tables I and II will be used for describing the different classifiers in the remainder of this section. Note that the actual number of feature values extracted per feature might be larger than one (refer to the descriptions in the tables). All feature values were normalized by subtracting the mean and dividing through the standard deviation among the values obtained from the training set.

As mentioned above, we have used several statistical classifiers for our mode detection systems. In order to derive diverse systems for our multiple classifier system we have used different subsets of the features for different classifiers. In the following the statistical classifiers are shortly summarized.

All meta-parameters of the classifiers described above have been optimized by performing a leave-one-out cross-validation (LOOCV) with each parameter combination on the training set. As a further optimization strategy a sequential forward search (SFS) has been applied on each selected classifier.

The first classifier is a k-nearest neighbor classifier (KNN) (we used the implementation of [9]). This classifier is applied on the Features 1–12 of Table I. For this kind of classifier the first meta-parameter is the number of neighbors k. The second parameter is the applied distance measure. Table III summarizes the distance measures used and the obtained optimal values of k. The output of the KNN is the class of the nearest sample together with a confidence, i.e., the normalized distance to this sample.[2] Note that this strategy leads to 10 systems using the KNN classifier as five distance measures are used and for each distance measure two classifiers are trained, i.e., one classifier without feature subset selection and one classifier after feature subset selection. Another 10 KNN systems are derived similarly from the Features 14–21 appearing in Table II.

The second classifier is a support vector machine (SVM) (we used the implementation of [10]). This classifier is again applied on the Features 1–12 of Table I. The meta-parameters of the SVM are the kernel function, the cost, and the degree in the kernel function (if applicable). The output of the SVM is the selected class together with a confidence. Again note that this strategy leads to 10 systems using the SVM classifier as five kernel functions are used and for each kernel two classifiers are trained. Another 10 SVM systems are derived similarly from the Features 14–2 appearing in Table II.

The third classifier is a multi-layer Perceptron (MLP) (we used the implementation of [11]). Since the training of MLPs is computationally expensive we have only selected the 12 feature values of Features 0, 1, and 13 appearing in Table I. Thus the MLP has 12 input neurons. For training, backpropagation with momentum is used. The meta-parameters of the MLP are the number of neurons in the hidden layer (40 is the optimal value), the learning rate η, the momentum, and the maximum number of backpropagation iterations. Using more than one hidden layer did not lead to improvements. The output of the MLP is the selected class together with a confidence (the value at the corresponding output neuron). Note that this results in two MLP classifiers, one classifier using all features and one classifier after SFS.

In summary 42 classifier systems have been used.

III. MULTIPLE CLASSIFIER COMBINATION

If several classifiers are available for a classification task it is advisable to combine the results of the classifiers in a multiple classifier system (MCS). From such a combination an improved recognition can be expected. A general overview and an introduction to the field of MCS is given in [12].

Several voting strategies have been evaluated for our MCS. In the following description we assume the task of discriminating text (t) and graphics (g). However, these strategies can easily be applied to more than two classes (c). Each classifier participating in the MCS outputs a recognition result together with a confidence.

The following typical voting strategies are investigated:

- Number of occurrences of each class n_c
- Maximum confidence for each class $\max C(conf_c)$
- Average confidence for each class
- Weighted voting $\alpha \cdot \frac{n_c}{\sum_{i=1}^{n} n_i} + (1-\alpha) * C(c)$, where $C(c)$ can be the maximum or the average confidence.

Furthermore, we applied statistical classifiers for the combination. The input of these classifiers are the outputs of

[2] Normalization is performed by taking into account all the distances obtained during LOOCV-training. The normalized distance is obtained by subtracting the mean and dividing through the standard deviation of all values.

Table V
Recognition accuracies (in %) of selected individual classifiers on the Touch & Write test data (the numbers in Column 3 correspond to IDs in Tables I and II). Note that the number in parentheses denote the IDs after SFS

ID	Class.	Note	Features	Accuracy	After SFS
1(8)	kNN	Mahalanobis	1–12	97.04	97.04
2(9)	kNN	Cityblock	1–12	97.04	94.70
3(10)	kNN	Cosine	1–12	95.57	96.43
4(11)	kNN	Canberra	1–12	96.80	95.07
5(12)	SVM	Polynomial	1–12	91.26	86.95
6(13)	SVM	Radial Basis	14–21	92.12	93.47
7(14)	MLP	40 Neurons	13–15	91.26	92.12

Table VI
Recognition accuracies (in %) of the MCS on the Touch & Write test data (the numbers in Column 4 correspond to classifier IDs in Table V)

Strategy	Accuracy	After Feature Selection	
		Accuracy	Selected Systems
Best Individual	97.04		
Majority Voting	96.30	95.93	1,5,8,13,14
Weighted Voting	96.12	**97.17**	1,5,8,13,14
MLP	96.43	96.92	1,5,8,13,14
kNN ($k = 3$)	96.80	**97.54**	1,5,8,13,14
Oracle	100.00	99.26	1,5,8,13,14

each participating system, i.e., the confidence value of the selected class and 0 for the other class(es). If all 42 classifiers participate in the MCS this would result in a 84-dimensional feature vector.

The following classifiers were applied on the above mentioned feature vector:

- KNN
- MLP
- SVM

As explained previously (see Section II), there are 42 individual classifiers available for potential inclusion in the ensemble. However, it is well known that ensemble performance does not necessarily monotonically increse with ensemble size. Therefore, the question arises which of the individual classifiers actually to include in the ensemble.

In order to select the optimal classifier ensemble we have applied a genetic algorithm for each above mentioned classification strategy. This results in 8 optimized classifier ensembles. Note that all optimization of the MCS has been performed on the training set or on the validation set, if available.

IV. Experimental Results

As mentioned above we have performed experiments on two datasets. The data set has been acquired on the Touch & Write surface (with the Anoto-pen as an acquisition device) and was introduced in [8]. The second data set is called IAMonDo-database and was introduced in [7]. Again, the Anoto pen has been used as an acquisition device. Both data sets are publicly available.[34].

A. Touch & Write Data and Results

The Touch & Write dataset contains drawings and text samples (letters, words, or sentences) contributed by 20 writers. Each writer contributed with at least 40 samples for each category, resulting in roughly $1,600$ samples. The data of 10 writers has been used for training the classifiers, as well as all meta-parameters during classification and combination. The remaining data has been used as test data for the final evaluation. The recognition task is to decide if a trace corresponds to either text or graphics.

Table V shows the results of selected individual classifiers. The kNN approach using the Mahalanobis distance performed best on the Touch & Write database with a recognition accuracy of 97.04 %. In Table V it can be observed that the recognition accuracy on the test set sometimes drops after SFS. The main reason for this seems to be an overfitting on the training set. This problem will be tackled by applying a cross validation in the future.

The recognition results of the MCS appear in Table VI. Recognition accuracies which outperform the best individual classifier are given in bold font. All MCS selected the same individual classifiers for the ensembles. It is interesting to note that the kNN-Mahalanobis classifier is selected twice, once before SFS and once after SFS is applied. These two classifiers also produce divers output on the test set. The best MCS is the kNN-combination with a performance of $97.54\,\%$, which is $0.5\,\%$ higher than the best individual classifier.

Again, it can be observed that the recognition accuracy on the optimized features sometimes drops. In another test, where we applied the genetic algorithm for selecting the members of the classifier ensemble, we have achieved a recognition accuracy of $98.65\,\%$. This shows that our system would perform better, as soon as more training data is available.

The last row in Table VI shows the recognition accuracy of an oracle classifier. This hypothetical classifier knows the ground-truth and chooses the correct output if at least one classifier has chosen it correctly. Noteworthy, an oracle classifier applied on all 42 classifiers would achieve a perfect performance. This indicates that there is still some room for improvement.

B. IAMonDo-database Data and Results

In order to access the performance of our system we have evaluated the performance on another publicly available benchmark database, the IAMonDo-database. This database consists of $1,000$ documents containing handwritten text, drawings, diagrams, formulas, tables, lists, and marking

[3] The Touch & Write dataset is publicly available at http://www.touchandwrite.de/dataset

[4] The IAMonDo-database is available at http://www.iam.unibe.ch/fki/databases

Table VII
RECOGNITION ACCURACIES (IN %) OF SELECTED INDIVIDUAL CLASSIFIERS ON THE IAMONDO-DATABASE.

ID	Class.	Note	Features	Accuracy	After SFS
1(9)	kNN	Mahalanobis	1–12	92.55	97.00
2(10)	kNN	Cityblock	1–12	96.29	96.50
3(11)	kNN	Cosine	1–12	94.04	91.58
4(12)	kNN	Canberra	1–12	95.88	96.07
5(13)	kNN	Euclidian	1–12	96.21	96.92
6(14)	SVM	Radial Basis	1–12	96.64	95.03
7(15)	SVM	Radial Basis	14–21	93.17	93.39
8(16)	MLP	40 Neurons	13–15	93.17	93.17

elements arranged in an unconstrained way. 200 persons produced 5 documents, each.

In [7] benchmark experiment for the task of distinguishing text and graphics has been performed. There the method of [4] achieved a performance of 91.3 % and the offline method proposed by [7] achieved 94.4 %.[5]

The recognition results of the individual classifiers appear in Table VII. Again, the kNN classifier achieves the best performance with 97 %. Noteworthy, the feature selection increased the recognition performance in most cases. This can be explained by the fact that the IAMonDo-database contains much more training data. Furthermore, a separate validation set has been used for classifier optimization.

The best multiple classifier system is a simple weighted voting and also performs with 97 %. Especially the discriminative voting strategies (SVM, kNN, MLP) did not succeed on this data set, which might be explained by the imbalanced distribution of text traces (93 %) and non-text traces. However, the recognition performance is already significantly better than the performance in the reference system.

V. CONCLUSION

In this paper we presented an online mode detection system for pen-enabled multi-touch environments. We have applied several state-of-the-art strategies and introduced new features. Furthermore, we applied a multiple classifier combination systems using several voting strategies.

In our experiments on the IAMonDo-database benchmark our final classifier could outperform previous approaches. In another experiment on more balanced data which has been acquired on the Touch & Write table we have achieved a performance of nearly 98 %. This system is used on our multi-touch table as a running prototype.

ACKNOWLEDGMENT

This work has been financially supported by the ADIWA project.

REFERENCES

[1] D. Willems and L. Vuurpijl, "A bayesian network approach to mode detection for interactive maps," in *Proceedings of the Ninth International Conference on Document Analysis and Recognition - Volume 02*. Washington, DC, USA: IEEE Computer Society, 2007, pp. 869–873. [Online]. Available: http://portal.acm.org/citation.cfm?id=1304596.1304913

[2] NUI Group Authors, *Multi-Touch Technologies*. NUI Group, 2009. [Online]. Available: http://nuicode.com/attachments/download/115/Multi-Touch_Technologies_v1.01.pdf

[3] M. Liwicki, O. Rostanin, S. M. El-Neklawy, and A. Dengel, "Touch & write: a multi-touch table with pen-input," in *9th Int. Workshop on Document Analysis Systems*, 2010, pp. 479–484.

[4] A. Jain, A. Namboodiri, and J. Subrahmonia, "Structure in on-line documents," in *Proceedings of the Sixth International Conference on Document Analysis and Recognition*. Washington, DC, USA: IEEE Computer Society, 2001, pp. 844–848. [Online]. Available: http://portal.acm.org/citation.cfm?id=876867.877719

[5] C. M. Bishop, M. Svensen, and G. E. Hinton, "Distinguishing text from graphics in on-line handwritten ink," in *Proceedings of the Ninth International Workshop on Frontiers in Handwriting Recognition*, ser. IWFHR '04. Washington, DC, USA: IEEE Computer Society, 2004, pp. 142–147. [Online]. Available: http://dx.doi.org/10.1109/IWFHR.2004.34

[6] D. Willems, S. Rossignol, and L. Vuurpijl, "Features for mode detection in natural online pen input," in *Proceedings of the 12th Biennial Conference of the International Graphonomics Society*. Citeseer, 2005, pp. 113–117.

[7] E. Indermühle, M. Liwicki, and H. Bunke, "IAMonDo-database: an online handwritten document database with non-uniform contents," in *9th Int. Workshop on Document Analysis Systems*, 2010, pp. 97–104.

[8] M. Liwicki, M. Weber, and A. Dengel, "Online mode detection for pen-enabled multi-touch interfaces," in *Proc. 15th Conf. of the International Graphonomics Society*, 2011, p. 4 pages.

[9] E. Jones, T. Oliphant, P. Peterson *et al.*, *SciPy: Open source scientific tools for Python*, 2001, software available at http://www.scipy.org/.

[10] C.-C. Chang and C.-J. Lin, *LIBSVM: a library for support vector machines*, 2001, software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[11] W. M., *Feed-forward neural network for python*, Technical University of Lodz (Poland), Department of Civil Engineering, Architecture and Environmental Engineering, software available at http://ffnet.sourceforge.net/.

[12] L. I. Kuncheva, *Combining Pattern Classifiers: Methods and Algorithms*. John Wiley & Sons Inc, 2004.

[5] Note, however, that the offline results are not directly comparable to the online results, because another evaluation method is applied. [7]