Radbound Repository

Radboud University Nijmegen

PDF hosted at the Radboud Repository of the Radboud University Nijmegen

The following full text is a publisher's version.

For additional information about this publication click this link.
http://hdl.handle.net/2066/56916

Please be advised that this information was generated on 2019-06-17 and may be subject to change.

Features for mode detection in natural online pen input

Don WILLEMS [a] , Stéphane ROSSIGNOL [a] and Louis VUURPIJL [a]

[a] *Nijmegen Institute for Cognition and Information (Radboud University Nijmegen)*
P.O.Box 9104
6500 HE Nijmegen, THE NETHERLANDS

Abstract. When the user is free to write anything, like handwriting, drawings, or gestures, techniques are required to distinguish between modes. Mode detection, preceding recognition, can be an important aid in applications that invite natural pen input. In this paper, a large amount of data, acquired in different contexts, is used to assess eight features on their suitability for mode detection. Six global features: length, area, compactness, eccentricity, circular variance, and closure, and two structural features: curvature, and perpendicularity, have shown to be particularly useful for determining whether a pen trajectory contains handwriting, lines, arrows, or geometric shapes. Using these eight features an overall performance on unseen data was achieved of 98.7%, using a KNN classifier. According to the principal components analysis of the data, the most important features were closure, curvature, perpendicularity, and eccentricity. The results of this study are employed in two large research projects on natural multi-modal interaction that pursue design, route map, and map annotation scenarios.

1. Introduction

In applications where users are allowed to generate any kind of natural pen input, traditional recognition techniques suffer from the uncertainty about the intention of the user. Since in such cases, the user is not constrained to enter one of a pre-defined number of shapes (e.g., as defined in a gesture lexicon), the number and form of shapes that can be expected is unknown. In particular with the advent of TabletPC applications, which may provide the option to write anything, anywhere on the screen, the early distinction of pen input into separate *modes* helps to steer subsequent recognition algorithms (Rossignol et al., 2004). The research presented in this paper pursues typical applications in which multiple input modes have to be distinguished. In the European IST-5 project COMIC (Boves et al., 2003), pen input in design applications is examined. In the Dutch ICIS project (ICIS, 2004), interactive route navigation and map annotation scenarios are being researched. Such applications require the recognition of pen drawn objects of many types, like handwriting, deictic gestures (e.g., arrows, encirclements, cross marks), complex drawings, and simple geometric shapes. Mode detection boils down to the detection of one of these input categories.

Research on mode detection is relatively new. In Jain et al. (2001) and Rossignol et al. (2004), handwriting and line segments were distinguished based on curvature and length features, resulting in recognition performances of respectively 99% and 98%. Bishop et al. (2004) achieved a performance of 95% on mode detection between handwriting and arbitrary drawn objects. In this paper we will try to sustain these performances while including more object classes. To this end, powerful features that have proven their value in both online and off-line shape matching will be explored. For a general overview of features used in off-line shape recognition see Peura and Iivarinen (1997) and Zhang et al. (2004).

In off-line pen input recognition, both region-based (finding salient features in a region of an image) and contour-based feature detection (using the features of contours in an image, see Zhang et al. (2004)) are employed. *Region-based* techniques include such techniques as the Hough transform (Ecabert et al., 2004; Wang et al., 1997), which uses data from the whole image to find lines, circles, or other shapes. *Contour-based* features include chain codes (Freeman, 1961; Zhang et al., 2004; Rossignol et al., 2004), Fourier descriptors (Osowski and Nghia., 2002), and curvature (Jain et al., 2001; Rossignol et al., 2004). Contour-based shape features use shape boundary information (Zhang et al., 2004) which has to be subtracted from an image. In this paper we will focus on using contour-based features for mode detection, since the outer contour of an object mimics the trajectory as acquired in the online applications we pursue.

Contour-based shape features can be distinguished in *structural* and *global* features. Global features use the dimensions of the object as principal features (Zhang et al., 2004), like area, length, compactness, and eccentricity. Structural features on the other hand describe relations between the path segments. They include chain codes, curvature and perpendicularity.

In the next section, a description of eight suitable features for mode detection is given. In Section 3., the results of using these features for mode detection in three distinct data sets is presented.

2. Shape features

The contour-based shape features we used were: *length* (λ), *area* (A), *compactness* (c), *eccentricity* (e), *circular variance* (v_{circle}), *closure* (c_l), *curvature* (κ), and *perpendicularity* (p_c). These features were

Notations	
sample	$\vec{s}_i = (x_i, y_i)$
pen stream	$\mathcal{S} = \{\vec{s}_i \| i \in [0, n-1], t_i < t_{i+1}\}$
area	A
length of enclosing perimeter of the area	l_c
length of the major axis of the bounding box	a
length of the minor axis of the bounding box	b
centroid position	$\vec{\mu} = \frac{1}{n}\sum_i \vec{s}_i$
mean radius	$\mu_r = \frac{1}{n}\sum_i \|\|\vec{s}_i - \vec{\mu}\|\|$
angle between samples	$\psi_{\vec{s}_i} = \cos^{-1}\left\{\frac{(\vec{s}_i - \vec{s}_{i-1})\cdot(\vec{s}_{i+1} - \vec{s}_i)}{\|\|\vec{s}_i - \vec{s}_{i-1}\|\|\|\|\vec{s}_{i+1} - \vec{s}_i\|\|}\right\}$

Feature equations	
length	$\lambda = \sum_{i=0}^{n-1} \|\|\vec{s}_i - \vec{s}_{i+1}\|\|$
compactness	$c = l_c^2 / A$
eccentricity	$e = \sqrt{1 - b^2/a^2}$
ratio between principal axis	$e_r = b/a$
circular variance	$v_{circle} = \frac{1}{n\mu_r^2}\sum_{i=1}^{n}(\|\|\vec{s}_i - \vec{\mu}\|\| - \mu_r)^2$
closure	$c_l = \frac{\|\|\vec{s}_0 - \vec{s}_n\|\|}{\lambda}$
curvature	$\kappa = \sum_{i=2}^{n-1} \psi_{\vec{s}_i}$
perpendicularity	$p_c = \sum_{i=2}^{n-1} \sin^2 \psi_{\vec{s}_i}$

Table 1
Notations and feature equations.

selected for their ability to be used in online shape recognition, where objects are generated by sequences of pen stream coordinates (for a definition see Table 1). If an object comprises multiple strokes, the pen stream contains the concatenation of all pen down strokes (no use is made of information such as pen pressure or tilt). The next two subsections describe the eight global and structural features.

2.1 Global features

The two most basic features are *length* (λ) of the path along the pen stream and *Area* (A). Since most pen streams are not closed, the area of the *convex hull* (Graham, 1972) of the stream is used instead.
Compactness or *circularity* is often defined as the ratio of the perimeter squared and the area of an object (Peura and Iivarinen, 1997). A circle is the most compact object and has a value of 2π. For an elongated object, compactness approaches infinity. If one uses the convex hull area, the perimeter of the convex hull should be used.
Eccentricity is a measure for the ratio between the major and minor axis of a bounding box. The eccentricity is 0 for a circle or square, and approaches 1 for a long elongated object. Often the ratio between the major and minor axis is used instead (Zhang et al., 2004).
Circular variance is a measure for the difference between the object and a circle as a template. The circular variance is the proportional mean-squared error with respect to the solid circle (Peura and Iivarinen, 1997). The centroid used is the centre of mass of the samples in the pen stream and not necessarily the centre of mass of the object. A better approximation for the centre of the object can be acquired by resampling the pen stream so that each sample point has the same distance to each of its neighbors ($||\vec{s}_i - \vec{s}_{i+1}|| = ||\vec{s}_j - \vec{s}_{j+1}||, \forall i, j \in [0, n-1]$).

2.2 Structural features

A simple structural feature is *Closure* which is the ratio between the distance between the first and last sample in a sample stream and the length of the stream.
In the COMIC project (Boves et al., 2003), a feature named mean *curvature* Rossignol et al. (2004) was used to distinguish handwriting from lines. The curvature measures the total (absolute) change

data set	Handwriting	Arrows	Lines	Geom. objects
development	121	51	48	0
training	375	68	99	200
test	540	103	127	290

Table 2
Mode categories per data set.

in orientation over the whole pen stream. For an ideal, perfectly straight line the curvature would be $\kappa = 0$, a perfect square or circle would result in: $\lim_{n\to\infty} \kappa = 2\pi$. Cursive handwriting, if it has sufficient characters results in a much larger curvature. A problem with curvature is that with online data, sample points are not necessarily equidistant, which results in sometimes large angles when the distance between subsequent points is very small in relation to the resolution of the digitizing tablet. Therefore it is necessary to resample the pen stream so that it contains equidistant sample points, whose distance should be larger than the resolution of the tablet.

A feature sensitive to straight angles is *perpendicularity* (p_c) which is the sum of squared sines of the angles between path segments of equal length. Perpendicularity is a measure for the number of sudden direction changes in the sample stream. As with curvature, the pen stream should be resampled to create path segments of equal length. A perfect straight line will have a perpendicularity of 0 while a perfect rectangle will have a perpendicularity of 4 (each right angled corner contributes 1). A circle on the other hand will have a smaller perpendicularity because the angle between each line segment is smaller than π if the number of sample points (n) is larger than 4. The perpendicularity of a circle is: $p_c^{circle} = n \sin^2 \frac{2\pi}{n}$ and will approach zero if more samples are used ($\lim_{n\to\infty} p_c^{circle} = 0$).

3. Experiments and results

To determine whether these global and structural features can be used efficiently for mode detection, we used three different data collections. These data were acquired in different contexts and from different writers. In the next sections, the three data collections, various instances of MLP and KNN classifiers, and experimental results assessing the recognition performance are presented.

3.1 Data

The data used in this paper were divided in three sets, a development set, a training set and a test set. The development set was used to (i) develop the feature extraction algorithms required to compute the features described in the previous section, (ii) to assess their distinctive properties through data analysis tools, and (iii) to optimize parameters like k in the KNN classifiers, the number of hidden units in the MLP neural networks and to perform initial training and testing. The remaining data was used for the experimental results presented below, where the training set was used for building the prototypes required for the KNN classifiers. The data originated from (i) the COMIC project (Boves et al., 2003), (ii) from Fonseca and Jorge (Fonseca and Jorge, 2001), (iii) and from the UNIPEN database (Guyon et al., 1994). The data acquired in the COMIC project was collected in two so-called human factors experiments with respectively 28 and 40 different writers. In both experiments, a writer had to describe the layout and dimensions of a bathroom, resulting in pen input comprising handwriting (typically digit strings referring to the dimensions of bathroom objects), drawing, sketches, and deictic gestures. The data generously made available by Fonseca (Fonseca and Jorge, 2001), contained six classes of geometrical objects (lines, rectangles, arrows, circles, diamonds and ellipses). The Fonseca data contained solid, bold, and dashed instances of each of these classes of objects and was acquired in the context of scribble recognition. From the UNIPEN data, a set of cursive words was randomly selected. Please note that the development set merely contained the data acquired in the COMIC experiment from 40 writers. The remaining data were divided in the training and test set (see Table 2).

As explained above, the mode of each pen input was determined based on pen stream information. Each pen stream was first resampled to $\lambda/5$ equidistant samples. This resulted in resampled pen streams where the distance of each sample to its neighbors was about 5 units. For each resampled pen stream the feature vector was calculated as described in the previous section. To prevent large scale features to dominate (Jain et al., 1999), features were normalized so they all had a common range (between 0 and 1).

3.2 Classifiers

The goal of this research was finding salient features for the mode detection of these four classes of hand drawn objects and test their effectiveness. Our goal was *not* to find the best classifiers. Therefore to test the effectiveness of the eight features we used six simple classifiers; three k-nearest-neighbour (KNN)

	Handwriting	Arrows	Lines	Geom.	Correct
Handwriting	99.4%	0.6%	0.0%	0.0%	99.4%
Arrows	4.9%	90.3%	1.9%	2.9%	90.3%
Lines	0.0%	0.0%	100.0%	0.0%	100.0%
Geom. objects	0.3%	0.0%	0.0%	99.7%	99.7%
Total					98.7%

Table 3
The confusion matrix for the KNN (k=4) classifier with the test class vertically and recognised class horizontally.

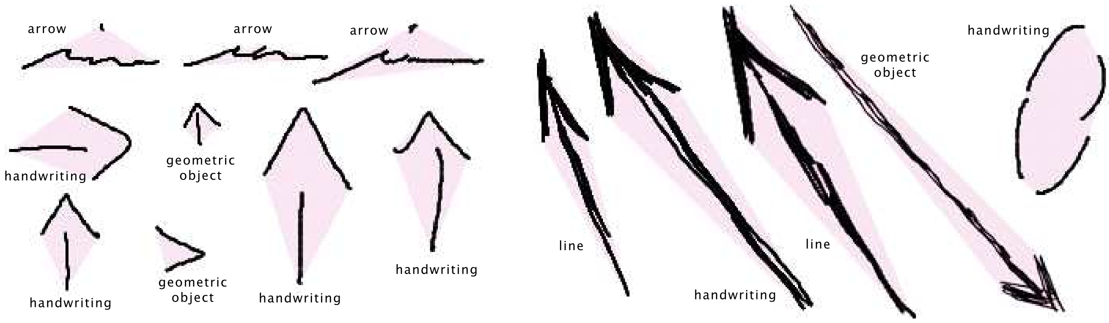

Fig. 1. All objects that were *mis*classified by the KNN classifier (k=4), labelled with the class of the object as recognised by the classifier.

classifiers (with $k = \{3, 4, 5\}$) and three four-layered perceptrons (MLPs). The three MLPs had 8 and 4 units, 8 and 8 units, and 16 and 8 units in the two hidden layers respectively. The KNN classifier used the Mahalanobis distance as the distance measure to counteract the effects of mutual linear dependent features (Jain et al., 1999). When the KNN classifier generated a draw, the class of the nearest neighbor was used as the result. For developing and training, only the development set was used.

3.3 Results

After the development and initial testing of the classification system, no features were added or changed. Because only a subset of COMIC data was used in the initial training set, no geometric objects (which are only present in the data from Fonseca et al.) were used during the development of the system. Nevertheless when geometrical objects were used during final testing, the results were quite satisfactory, as depicted in Table 3.
A maximum performance of 98.7% was reached by the KNN classifier with k=4. As can be seen in Figure 1, most errors (13 out of 14) were caused by the misrecognition of arrows. Handwriting was misclassified as arrows three times, five arrows were misclassified as handwriting, three arrows as geometric objects, and two arrows as lines. Finally one geometric object was misclassified as a geometric object.
The KNN classifiers with k=3 and k=5 achieved an overall performance of 98.6% and 98.4% respectively. All three MLP classifiers achieved the same overall performance (97.6%), although they performed differently on each class. On arrows, the performance of the 8x8x8x4 MLP classifier was better (93.2%) than the performance of the KNN classifier (90.3%).
To determine which features contribute most to correct classification, a *principal components analysis* (PCA) was performed on all data. According to the PCA, the most important feature is closure, followed by curvature, perpendicularity, and eccentricity. Using only these four features a mode-detection performance of 96.5% is obtained with the KNN classifier (k=4).

4. Discussion and future research

In this paper, eight features for mode detection are described and tested on three data sets, originating from different sources. Various configurations of KNN and MLP classifiers were used, yielding a performance of 98.7% on the distinction of four main classes: handwriting, arrows, lines and geometrical objects. This performance is comparable to the results published in Jain et al. (2001) and Rossignol et al. (2004) even though the number of object classes is twice as large. Moreover, since these results are achieved on unseen data from distinct sources, we may conclude that the features assessed here are robust and general purpose in the sense that they are writer-independent and suitable for a wide range of applications.
The resulting mode detection system made 14 errors in 1060 cases. An examination of these cases showed that the errors were mainly due to the deictic arrow category (see Figure 1). The largest number of

misclassifications involved arrows that were recognized as handwriting. A closer look at these arrows (Figure 1) shows that in four of these cases the arrows had a short tail compared to their heads and tail and head were disconnected from each other. This resulted in a (compared to other arrows) higher curvature which is characteristic of handwriting. On the other hand, the three strings that were misclassified as arrows were all small, containing only a few characters and were somewhat flattened, resulting in a (for handwriting) exceptionally low curvature.

Our current research is targeted on distinguishing more deictic gestures (e.g., cross marks, encirclements), which typically occur in the map annotation and route navigation scenarios envisaged in the ICIS project. In ICIS, a framework in which mode detection and object recognition techniques are merged is pursued. In this framework, the bottom up expectations made by the mode detection system will be combined with the results of subsequent recognition algorithms, feeding back information in case of uncertain results. Furthermore, it is expected that integrating the results from multiple classifiers (like the MLP classifier which performs better on certain categories than the KNN) will improve the confidence in the system's output. It is our believe that the combination of information on these different levels will result in more robust interactive pen-based systems that allow natural interactions, i.e., make very little assumptions on the pen input repertoire that users will employ in conditions where they are left completely free.

Acknowledgements

This work is supported by the Dutch ICIS project (grant BSIK03024) and by the European COMIC project (grant IST-2001-32311). We would also like to thank Manuel Fonseca for making his data available to us.

References

Bishop, C.M., Svensén, M. & Hinton, G.E., (2004). Distinguishing Text from Graphics in On-line Handwritten Ink. Proc. Int. Workshop on Frontiers in Handwriting Recognition – IWFHR'04, Tokyo, Japan, October 26–29, pp. 142–147

Boves, L., Neumann, A., Vuurpijl, L., ten Bosch, L., Rossignol, S., Engel, R. & Pfleger, N., (2003). Multimodal Interaction in Architectural Design Applications. 8th ERCIM Workshop on "User Interfaces for All", Vienna, Austria, June 28–29.

Ecabert, O. & Thiran, J. (2004). Adaptive Hough transform for the detection of natural shapes under weak affine transformations. Pattern Recognition Letters, 25(12), pp. 1411–1419.

Fonseca, M.J. & Jorge, J.A. (2001). Experimental evaluation of an on-line scribble recognizer. Pattern Recognition Letters, 22(12), pp. 1311–1319.

Freeman H. (1961). On the encoding of arbitrary geometric configurations. IRE Trans. Electron. Comput. EC-10, pp. 260–268.

Graham R. (1972). An Efficient Algorithm for Determining the Convex Hull of a Finite Point Set. Info. Proc. Letters, 1, pp. 132–133.

Guyon, I., Schomaker, L., Plamondon, R., Liberman, M. & Janet, S., (1994). UNIPEN project of on-line data exchange and benchmarks. Proc. of the 12th Int. Conf. on Pattern Recognition – ICPR'94, IAPR–IEEE pp. 29–33.

Interactive Collaborative Information Systems – ICIS (2004), http://www.decis.nl/projects/icis_program/.

Jain, A.K., Murty, M.N. & Flynn, P.J. (1999). Data clustering: a review. ACM Computing Surveys, 31(3), pp. 264–323.

Jain, A.K., Namboodire, A.M. & Subrahmonia, J., (2001). Structure in On-line Documents. Proc. of the 6th Int. Conference on Document Analysis and Recognition – ICDAR'01, Seattle, WA (USA), September, pp. 844–848.

Osowski, S. & Nghia, D.D. (2002). Fourier and wavelet descriptors for shape recognition using neural networks–a comparative study. Pattern Recognition, 35(9), pp. 1949-1957.

Peura, M. & Iivarinen, J. (1997). Efficiency of Simple Shape Descriptors. In Advances in Visual Form Analysis, Arcelli C., Cordella, L.P. & Sanniti di Baja, G. eds, Singapore: World Scientific, pp. 443–451.

Rossignol, S., Willems, D.J.M., Neumann, A. & Vuurpijl, L.G., (2004). Mode detection and incremental recognition. Proc. Int. Workshop on Frontiers in Handwriting Recognition – IWFHR'04, Tokyo, Japan, October 26–29, pp. 597–602.

Wang, J., Leung, M.K.H. & Cheung, S. (1997). Cursive word reference line detection. Pattern Recognition, 30(3), pp. 503–511.

Zhang, D. & Guojun, L. (2004). Review of shape representation and description techniques. Pattern Recognition, 37(1), pp. 1–19.