Mode detection in on-line pen drawing and handwriting recognition

Don Willems, Stéphane Rossignol, and Louis Vuurpijl
Nijmegen Institute for Cognition and Information
Radboud University Nijmegen
P.O.Box 9104, 6500 HE Nijmegen, The Netherlands
DD.Willems@nici.ru.nl

Abstract

On-line pen input benefits greatly from mode detection when the user is in a free writing situation, where he is allowed to write, to draw, and to generate gestures. Mode detection is performed before recognition to restrict the classes that a classifier has to consider, thereby increasing the performance of the overall recognition. In this paper we present a hybrid system which is able to achieve a mode detection performance of 95.6% on seven classes; handwriting, lines, arrows, ellipses, rectangles, triangles, and diamonds. The system consists of three kNN classifiers which use global and structural features of the pen trajectory and a fitting algorithm for verifying the different geometrical objects. Results are presented on a significant amount of data, acquired in different contexts like scribble matching and design applications.

1. Introduction

When users are allowed to generate any kind of input when using a tablet application, it becomes necessary to distinguish between different modes of pen input. The recognition task is particularly complex in applications such as interactive map navigation, map annotation, and design tasks. For example, the user can enter handwriting, drawings of geometric shapes, or deictic gestures. The recognition system should be able to recognize all of these types of pen input. Because of the many object classes involved in natural pen input recognition, using only one single, monolithic classifier may not result in optimal performance. In this paper, this complex task is performed by using a mode detection system to determine whether a pen drawn object is handwriting, a shape, or a gesture, before a specialized classifier is employed to determine the content of the handwriting or the position and orientation of an object. Shape/handwriting recognition is thereby distinguished in several subtasks. The focus in this paper is on mode detection, first to distinguish between handwriting and shapes and then to distinguish between several important classes of shapes, such as lines, arrows, and geometrical objects.

Research on mode detection is relatively new. A performance of 95% was achieved for mode detection between handwriting and graphics by classification systems presented in [2]. Mode detection between handwriting and lines was tested using curvature and length features in [16] and [13] achieving performances of 98% and 99% respectively. It is the goal of this paper to explore other powerful features such as those presented in [15] and [18] for achieving similar performances while distinguishing between more classes of input modes.

This paper is organized as follows. First, in Section 2, the system architecture that was developed to test the classifiers and the features used in mode detection will be discussed. Second, in Section 3, we will present the data used to train and test the system, and the results of the tests.

2. System architecture

The system we developed for on-line pen drawing and handwriting recognition is depicted in Figure 1. It consists of four mode detection classifiers organized in a decision tree, a handwriting recognition classifier, and several fitting algorithms to determine the position, size and orientation of each object.

This paper focuses on maximizing the performance of the four mode detection subsystems. Various feature combination schemes have been explored in order to determine the best set of features for each subsystem. Seven features we used are described in the literature [15, 16, 17, 18] and in this paper, five new features are presented. The classifiers used for mode detection between handwriting, lines, arrows, and geometrical objects, all used subsets of these twelve features. For distinguishing between the four geometrical object types (rectangle, triangle, ellipse, and diamond), we used a fitting algorithm which implements a chain code histogram to create a template of the drawn ob-

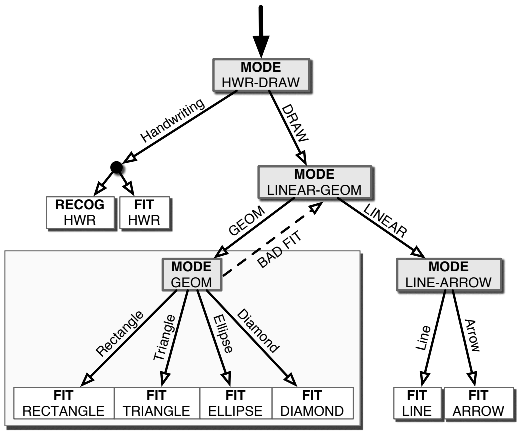

Figure 1. System architecture, consisting of four mode detection classifiers (MODE). The result of these classifiers are fed into the relevant fitting (FIT) subsystem and/or recognition (RECOG) classifier.

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

Table 1. Notations.

length	$\lambda = \sum_{i=0}^{n-1} \|\|\vec{s}_i - \vec{s}_{i+1}\|\|$
area	A
compactness	$c = l_c^2 / A$
eccentricity	$e = \sqrt{1 - b^2/a^2}$
principal axes	$e_r = b/a$
circular variance	$v_c = \frac{1}{n\mu_r^2}\sum_{i=1}^{n}(\|\|\vec{s}_i - \vec{\mu}\|\| - \mu_r)^2$
rectangularity	$r = \frac{A}{ab}$
centroid offset along major axis	$o_c = \frac{\|(\vec{\mu} - \vec{a} \cap \vec{b}) \cdot \vec{a}\|}{\|\vec{a}\|}$
closure	$c_l = \frac{\|\|\vec{s}_0 - \vec{s}_n\|\|}{\lambda}$
curvature	$\kappa = \sum_{i=2}^{n-1} \psi_{\vec{s}_i}$
perpendicularity	$p_c = \sum_{i=2}^{n-1} \sin^2 \psi_{\vec{s}_i}$
signed perpendicularity	$p_{sc} = \sum_{i=2}^{n-1} \sin^3 \psi_{\vec{s}_i}$

Table 2. Features.

ject. The results of this fitting algorithm (containing a confidence value for the fit and a definition of the fitted template) are used for both the verification of mode detection and the final recognition (fit) of the object.

Before discussing the results of this system on the data sets, we will first describe the features used in the classifier and the fitting algorithm for the geometrical object classes.

2.1. Features

Shape features can be distinguished in global and structural features [17, 18]. Global features are features that use the dimensions of a complete pen stream. Examples are the length and area of the pen stream. Structural features describe the relations between the path segments. Examples of structural features are curvature [16] and perpendicularity (both real-valued scalars). Structural features may also include chain codes [7, 16, 18] and their histograms [16]. For an overview of the features and the equations describing the features, see Table 2.

The global features used in this paper are: length (of the pen stream), area, compactness, eccentricity, ratio between the principal axes, circular variance, rectangularity, and the centroid offset along the major axis. These global features describe properties of the area of the pen stream. For the area, which is also used in the calculation of compactness and rectangularity, the area of the convex hull [9] (enclosing the pen stream) is used.

Structural features describe the relationships between individual samples. Closure for instance gives the ratio of the distance between the first and last sample in a pen stream and the length of the stream. Closed objects such as rectangles will have a smaller closure than lines. Other structural features are curvature [16], perpendicularity [17], and the signed perpendicularity.

The features rectangularity, centroid offset along the major axis, closure, perpendicularity, and signed perpendicularity were introduced by us, the other features were taken from [15, 16, 18].

Feature Selection: Using all twelve features, a performance is reached of 99.1% for mode detection between handwriting and drawing (**MODE** HWR-DRAW in Figure 1). Since it is known that particular feature combinations will improve specific class separation boundaries, for tuning the mode detection subsystems, different combinations of feature sets were explored. To find the best feature set for each classifier, a simple genetic algorithm [11] was used, optimizing on the development data (see Section 3).

The best performance for the **MODE** HWR-DRAW classifier was reached by using stream length, curvature, circular variance, closure, and perpendicularity. For the **MODE** LINEAR-GEOM) classifier the best performance was attained using compactness, curvature, average curvature, circular variance, eccentricity, and closure, and for **MODE** LINE-ARROW the best performance was reached by compactness, curvature, eccentricity, closure, and the centroid offset along the major axis. The classification of geometrical shapes (**MODE** GEOM) used a different algorithm and was not optimized using the genetic algorithm.

2.2. Fitting algorithm for mode verification

None of the feature combinations were sufficient for distinguishing between the four different classes of geometrical objects (**MODE** GEOM in Figure 1). The performance using a kNN was never better than 78% on the development set. To improve the performance of the system on these object classes we used a fitting algorithm. For each unknown pen input, the algorithm (i) generated best-matching templates for the four object classes; (ii) fitted the input to the four templates and (iii) verified whether the input could be properly matched to one of the templates.

There is a variety of different algorithms such as the Hough transform [1, 3, 5] or generative models with bayesian model comparison [14] that can be used for on-line shape recognition. In the research presented in this paper we used the chain code histogram.

Detecting corner points: Our fitting algorithm does not use a discrete chain code as in the original implementation [7, 8] but a modified chain code histogram (CCH) that uses the sequence of the angles of each line segment with the coordinate system (a continuous value) instead. The histogram value at each angle ($0 < \alpha < 2\pi$) was calculated by counting all angle values found in the pen stream within a certain range around the angle in the histogram being calculated ($\alpha - \theta \ < \ \beta \ < \ \alpha + \theta$). Depending on the value of θ the histogram became smoothed to a higher or lesser degree, resulting in more or less local maxima in the histogram.

The chain code histogram was used to find the straight lines and the corners of the pen drawn object. With these lines and corners one can create templates for the different object classes which one can compare with the pen drawn object. The class whose template differs least from the object is used as the result of the classifier. This process does not only result in mode detection for the geometrical objects but also in the recognition of the object itself as the template provides the position, orientation, and other properties of the recognized object.

To determine the position of straight lines (from which rectangles, triangles and diamonds can be constructed), the important maxima were extracted from the chain code histogram. This was done using two different thresholds t_{peak} and t_{wing}, both based on the median of the angles of the line segments. If the value of the histogram at a certain angle ($H_{cc}(\alpha)$) came above t_{peak}, a new peak in the histogram was found. The width of the peak was determined by the second (lower) threshold t_{wing}. Next, the pen samples which contributed to the peak were determined, and a line was fitted through these pen samples. Each line was then compared with the other lines found, and if the lines had the same orientation (within 0.1 radians) and were close to each other, they were merged. Next, the intersection points between each of the lines was determined. For each intersection point the sample closest was chosen as a possible corner for the templates.

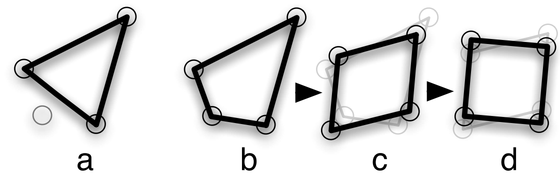

Figure 2. The creation of templates for (a) a triangle and (b to d) a rectangle. See text (Section 2.2) for an explanation.

Template construction: If more than three corner points were found, all possible triangle templates using these corner points were created (see Figure 2a). To create the templates for the rectangle and diamond classes, all possible templates for a quadrangle were constructed. Subsequently, each pair of opposite (non-intersecting) edges was made parallel by changing their orientation by an amount of half the orientation difference between the two opposing edges (Figure 2c). The center of the edge is used as pivot. During this process the centroid of the four corners of the quadrangle (now a parallelogram) remained at the same location.

The final step in the creation of the rectangle template is changing the angles between the edges to $\pi/2$ radians. This was done by taking one set of opposing edges and rotating both edges (again with the center of the line as pivot) so that all angles between the edges became $\pi/2$ radians (Figure 2d). The set of opposing edges with the smallest length were chosen to be rotated as the change was less than when the other set was chosen.

Diamonds are parallelograms with four edges of the same length. To create a template for the diamond class, a parallelogram is created as above (Figure 2c), but instead of changing the angles between the edges to $\pi/2$, the lengths of the four edges is equalized by changing each edge so that its length is equal to the average edge length in the parallelogram. Again, the centroid of the corners remains the same.

Finally, the template for an ellipse is created by using the intersection point of the principal axes as the center, the orientation of the major axis, and the eccentricity of the pen stream.

	Development	Training	Test	Total
HWR	220	202	614	1036
LINEAR	88	109	299	496
Arrow	30	50	142	222
Line	58	59	157	274
GEOM	97	99	294	490
Ellipse	35	35	112	182
Rectangle	17	15	62	94
Diamond	30	26	65	121
Triangle	15	23	55	93
Total	405	410	1207	2022

Table 3. Number of items per category per set.

	Hwr	Arr	Lin	Tri	Ell	Rec	Dia
Hwr	99.8	4.2	0.6			1.6	1.5
Arr	0.2	92.3	0.6	1.8			
Lin		2.1	98.1				
Tri		1.4	0.6	96.4		6.6	3.1
Ell				1.8	86.6	8.2	1.5
Rec					4.5	78.7	6.2
Dia					8.9	4.9	87.7

Table 4. The confusion matrix (in percentages) for the kNN (k=3) classifier with the test class horizontally and recognized class vertically. (Hwr=handwriting, Arr=arrow, Lin=line, Tri=triangle, Ell=ellipse, Rec=rectangle, and Dia=diamond.)

Fitting the input: The quality of fit of each of the templates is calculated by first selecting which pen sample belongs to which edge in the template and then calculating the correlation coefficient for each edge in the template and taking the average over all edges. The edge to which a sample belongs is taken to be the edge whose distance to the sample is smallest. This distance is then used in the calculation of the correlation coefficient of that edge.

The fitting algorithm automatically produced a measure (the fit for each template) for determining whether the result of the classifier was *good* enough. If the fit was not good enough (determined by a threshold on the fit), the system retraced back to the previous classifier (**MODE** LINEAR-GEOM) and the next classifier (**MODE** LINE-ARROW) was automatically called.

3. Experiment and Results

To determine whether this system with multiple mode detection classifiers can be used efficiently for mode detection, we tested the performance of the system on a test set of pen stream data. The data was acquired from different writers in different contexts, and was attained from three different data collections. In the following sections the data collections, the classifiers used in the system, and the resulting performances are discussed.

3.1. Data

Before starting development of the system, the data was divided in three sets, a training set, a development test set, and an experimental test set. The development set was used to (i) develop and select the best features for each mode detection classifier, (ii) to assess the results of combinations of features using data analysis tools, and (iii) optimize the classifiers used in the process. The development set was not used during final testing of the system.

The training set was used both during development and while testing. It was only used to train the classifiers. The largest set, the test set (see Table 3), was only used for the final evaluation of the system.

The data used in the experiment originated from three different sources and was already segmented. Most of the handwriting data came from the UNIPEN database [10], which included cursive handwriting from many different writers. Other handwriting data (measures) originated in experiments from the COMIC project [4] which also provided line and arrow data. Fonseca and Jorge [6] provided more line and arrow data and all geometric object data.

The development, training, and test sets were created beforehand by randomly selecting pen streams from the complete data set.

3.2. Classifiers

Since we were mainly interested in determining salient features, and not in the differences between different types of classifiers, only k-nearest-neighbor (kNN, with k=3) classifiers were used. To prevent effects from multiple dependent features, the Mahalanobis distance was used instead of the Euclidean distance [12]. When a kNN classifier generated a draw, the nearest neighbor was used as the result of the classifier.

3.3. Results

Using the kNN classifier we were able to attain an overall performance of 95.6%. The classifiers **MODE** HWR-DRAW and **MODE** LINEAR-GEOM achieved a performance of 99.2% and 98.8% respectively. The **MODE** LINE-ARROW classifier achieved 99.0% and the **MODE** GEOM classifier which used templates instead of features reached a performance of 87.9%.

As one can observe in the confusion matrix for the complete system (Table 4), ellipses, rectangles and diamonds are easily confused. Analysis of the pen streams that lead to the confusion shows that confusion mainly takes place between thickly drawn objects, where the user draws multiple lines to create one thick line. Confusion also takes place between diamonds and rectangles where the rectangle has almost equal sides (a square), or when the diamond has perpendicular angles (also a square). These situations are ambiguous as both diamonds and rectangles can be squares.

4. Discussion

In this paper, a hybrid system for mode detection between seven different classes of hand-drawn objects is described, using both global and structural features and template fitting classifiers. Using this system we were able to attain an overall performance of 95.6% for mode detection between all of these seven classes. Since most errors were caused by misclassifications of geometrical objects, our current efforts concentrate on improving the corresponding fitting and verification algorithms. Mode detection between handwriting and drawings reached a performance of 99.2%, which compares favorably to the results published in [13] and [16], even though the drawing class we used contained more varied data.

The work presented in this paper aims at the design of general purpose mode detection algorithms. Such techniques are indispensable for complex pen input applications in which users are not constrained to a specific pre-defined dictionary of object shapes. As shown here, the exploration of suitable features for mode detection, tested on data acquired in different contexts, has provided us with a proper baseline system. Our future research will not only improve the current technologies, but also employ the system in interactive multimodal map navigation.

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

[1] D. H. Ballard. Generalizing the hough transform to detect arbitrary shapes. *Pattern Recognition*, 13:111–122, 1981.

[2] C. M. Bishop, M. Svensén, 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*, pages 142–147, 2004.

[3] N. Bonnet. An unsupervised generalized hough transform for natural shapes. *Pattern Recognition*, 35:1193–1196, 5 2002.

[4] L. Boves, A. Neumann, L. Vuurpijl, L. ten Bosch, S. Rossignol, R. Engel, and N. Pfleger. Multimodal interaction in architectural design applications. In *8th ERCIM Workshop on "User Interfaces for All"*, Palais Eschenbach, Vienna, Austria, 28-29 June 2004.

[5] O. Ecabert and J.-P. Thiran. Adaptive hough transform for the detection of natural shapes under weak affine transformations. *Pattern Recognition Letters*, 25:1411–1419, 9 2004.

[6] M. J. Fonseca and J. A. Jorge. Experimental evaluation of an on-line scribble recognizer. *Pattern Recognition Letters*, 22:1311–1319, 2001.

[7] H. Freeman. On the encoding of arbitrary geometric configurations. *IRE Trans. Electron. Comput. EC-10*, pages 260–268, 1961.

[8] H. Freeman. Computer processing of line-drawing images. *Computer Surveys*, 6:57–97, 1974.

[9] R. Graham. An efficient algorithm for determining the convex hull of a finite point set. *Info. Proc. Letters*, 1:132–133, 1972.

[10] I. Guyon, L. Schomaker, R. Plamondon, M. Liberman, and S. Janet. UNIPEN project of on-line data exchange and benchmarks. In *Proceedings of the 12th International Conference on Pattern Recognition*. IAPR-IEEE, http://www.unipen.org, 1994.

[11] J. Holland. *Adaptation in Natural and Artificial Systems*. The University of Michigan, 1975.

[12] A. K. Jain, M. N. Murty, and P. J. Flynn. Data clustering: a review. *ACM Computing Surveys*, 31:264–323, 1999.

[13] A. K. Jain, A. M. Namboodiri, and J. Subrahmonia. Structure in on-line documents. In *Proceedings of the 6th International Conference on Document Analysis and Recognition (ICDAR'01)*, pages 844–848, Seattle, Washington, September 2001.

[14] B. Krishnapuram, C. M. Bishop, and M. Szummer. Generative models and bayesian model comparison for shape recognition. In *Proceedings of the ninth International Workshop on Frontiers in Handwriting Recognition*, pages 20–25, 2004.

[15] M. Peura and J. Iivarinen. Efficiency of simple shape descriptors. In C. Arcelli, L. P. Cordella, and G. S. di Baja, editors, *Advances in Visual Form Analysis*, pages 443–451, Singapore, 1997. World Scientific.

[16] S. Rossignol, D. Willems, A. Neumann, and L. Vuurpijl. Mode detection and incremental recognition. In *Proceedings of the ninth International Workshop on Frontiers in Handwriting Recognition*, pages 597–602, 2004.

[17] 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 (IGS2005)*, 2005 In Press.

[18] D. Zhang and G. Lu. Review of shape representation and description techniques. *Pattern Recognition*, 37:1–19, 1 2004.