Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence (2008)

GLADDER: Combining Gesture and Geometric Sketch Recognition

Paul Corey and Tracy Hammond

Sketch Recognition Laboratory, Department of Computer Science, Texas A&M University
Mail Stop 3112, College Station, TX 77839
(979)696-1286, pfc9416@cs.tamu.edu (979)862-4284, hammond@cs.tamu.edu

Abstract

Sketch recognition systems usually recognize strokes either as stylistic gestures or geometric shapes. Both techniques have their advantages. This paper presents a method for integrating gesture-based and geometric recognition techniques, significantly outperforming either technique on its own.

Introduction

Sketch recognition is the automated understanding of hand-drawn shapes which are automatically simulated, animated, or otherwise processed. For example, circuit diagrams can be automatically recognized, interpreted, and analyzed, i.e. solving for the missing current value in a circuit.

Thus far, most sketch systems recognize objects in two ways: 1) by how they were drawn using gesture-based stylistic drawing features such as [Rubine 1991], or 2) by what they look like using geometric features that describe the shape and arrangement of the object such as [Hammond 2005]. Gesture-based recognition has the disadvantage that shapes must be drawn in a particular drawing style, but if they are drawn as intended the recognition rates can be high. Geometric recognition has the advantage that shapes can be drawn naturally; however, certain shapes are difficult to describe using their geometric subparts. Returning to the circuit diagram example, the components, such as resistors and capacitors, could be easily recognized geometrically, but a character such as the number '2' in a label is difficult to describe geometrically and would be recognized more accurately using a gesture system. Previous systems, such as Music Notepad, use spatial arrangement of gestures-based glyphs to aid in sketch interpretation but do not fully integrate low-level geometric and gesture classes [Forsberg 1998]. Unlike the combination methods presented in [Kittler 1998], our method combines two classifiers with potentially distinct classes.

Our goal is to combine the techniques to improve accuracy, add drawing flexibility, and enable recognition of a broader number of shapes. Here, we describe a method, called GLADDER, to combine the gesture-based technique of Rubine with the geometric recognition system, LADDER. Our combined implementation outperforms either system on its own.

Implementation

We built a system that allows new shapes to be defined either by LADDER in geometric rules or by Rubine features computed from class examples.

Modified Rubine Recognizer

The standard Rubine method creates a linear classifier based on class-specific average feature vectors and pooled covariance that is used for all glyph classes [Rubine 1991]. However, this pooled covariance loses any class specific information. Therefore, we use a modified Rubine method featuring a quadratic classifier. Class specific covariance matrices are maintained for each class, in addition average feature vectors. Input strokes are assigned to a specific class based on computation of Mahalanobis distances. The Malalanobis distance for example f to class i is computed as:

$$\sigma_i^2 = (f - \bar{f}_i)\Sigma_i^{-1}(f - \bar{f}_i)$$

where $\bar{f}_i$ is the average feature vector for class i and Σ_i is the covariance matrix of class i. Each example is assigned to the class whose Mahalanobis distance is smallest.

LADDER Recognizer

LADDER is a geometric recognition framework. Primitives are recognized by a low-level recognizer and combined into more complex shapes using geometric constraints [Hammond 2005]. The LADDER low-level recognizer used is that designed by Brandon Paulson [Paulson 2008]. It consists of a set of tests and a hierarchical classifier. Each test determines if an input stroke could be interpreted as one of the LADDER primitives and creates a fit for passed primitive types. These primitive fits are ordered based on the hierarchical classifier, and the top three are selected as possible interpretations. Simpler, less complex interpretations are added to the list of fits before more complex fits.

Recognizer Assignment

To correctly classify input examples, the recognition system must correctly determine which recognizer, Rubine or Paulson, should be used to interpret the stroke. To do this, a rejection method is used. First, the Rubine classifier is used to determine the minimal Mahalanobis distance to any Rubine class. This distance is then compared to a threshold and rejected from classification by the Rubine method if the threshold is exceeded. The average Mahalanobis distance of training Rubine glyphs was determined to be 24, and that of LADDER primitives 100. A threshold value of 35 was determined empirically to optimally separate Rubine strokes from Paulson.

For compatibility with the LADDER framework, the Paulson interpretation is computed, giving a list of potential fits. If a stroke's Mahalanobis distance falls below the threshold, the Rubine interpretation is added as the top of the fit list, and as the bottom, if not. This augmented fit list is then available for use in LADDER and permits users to define shapes that are composites of both the LADDER primitives and Rubine glyphs. Context can later be used to rectify an incorrect ordering of fits.

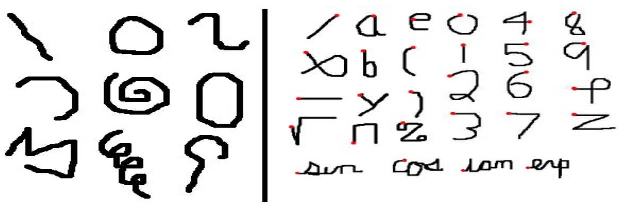

Figure 1: Example shapes. On the left, LADDER primitives. On the right, single-stroke math glyphs.

Modified Rubine	61.9%
LADDER	75.2%
Integrated	79.9%

Table 1: Overall top-result accuracy of each method.

Results

We performed a user study to measure the accuracy of our combined method. We collected a data set consisting of 28 single-stroke math symbols and simple geometric shapes. Figure 1 shows the shapes used in the study. Note that shapes on the left are LADDER geometric primitives that can vary in shape and style, while the mathematical shapes on the right are more naturally represented using gestures. We defined each shape in both gesture and geometric format and tested recognition using Rubine alone, LADDER alone, and our integrated method. Data from 23 users was collected to total 3520 examples. We used 1824 of these for training and the rest were reserved for testing.

For the integrated method, the correct classification was returned as the top result in 79.9% of test examples. Using the modified Rubine method alone resulted in the correct top classification of 61.9% of the examples. In 75.2% of examples, LADDER chooses the correct class as the top result. These results are summarized in Table 1.

Additionally, it should be noted that several Rubine glyphs overlap with LADDER primitives. For example, the '0' class is often drawn similarly to a circle or ellipse. As these shapes look similar, the incorrect interpretation often topped the list, accounting for much of the error. This is one of the primary reasons for the ranked list, allowing context to help determine the correct interpretation.

Future Work

Several areas of future extension still exist within this framework. Although we present a ranked list of interpretations, including both LADDER and Rubine interpretations, it is unclear how to effectively compare how to compare error rates to merge multiple interpretations from each recognizer. One method that could be to used a tiered thresholding system that inserts a Rubine fit with a higher Mahalanobis distance after a less complex Paulson interpretation like line or circle, but before a more complex curve or polyline interpretation. Also, overlapping low-level classifications should be combined into a single class that is distinguished at a higher level, increasing low-level recognition rates.

Conclusion

We have described a method for combining both geometric and gesture-based sketch recognition techniques into a single unified framework. This method can assign strokes to the correct type of classifier with a high degree of accuracy. Also, GLADDER gains the advantages of both geometric and gesture-based systems: naturally drawn shapes that do not depend on style and complex shapes that may be difficult to describe geometrically.

Acknowledgments

This work is funded in part by NSF IIS grant 0744150: Developing Perception-based Geometric Primitive-shape and Constraint Recognizers to Empower Instructors to Build Sketch Systems in the Classroom.

References

Forsberg, A., Dieterich, M., and Zeleznik, R. 1998. The Music Notepad, In Proceedings of UIST '98, ACM SIGGRAPH.

Hammond, T., Davis, R. 2005. LADDER, a sketching language for user interface developers. *Computers and Graphics* 29: 518-532.

Kittler, J.; Hatef, M.; Duin, R.P.W.; Matas, J., "On combining classifiers," *Pattern Analysis and Machine Intelligence, IEEE Transactions on* , vol. 20, no.3, pp.226-239, Mar 1998.

Paulson, B., Hammond, T. 2008. PaleoSketch: Accurate Primitive Sketch Recognition and Beautification. In Proceedings of IUI 2008.

Rubine, D. 1991. Specifying gestures by example. In Proceedings of the 18th Annual Conference on Computer Graphics and Interactive Techniques SIGGRAPH '91. ACM, New York, NY, 329-337.