Sketch Recognition based on Manifold Learning

Heeyoul Choi and **Tracy Hammond**
Department of Computer Science, Texas A&M University
3112 TAMU, College Station, TX 77843-3112, USA
{hchoi, hammond}@cs.tamu.edu

Abstract

Current feature-based methods for sketch recognition systems rely on human-selected features. Certain machine learning techniques have been found to be good nonlinear features extractors. In this paper, we apply a manifold learning method, kernel Isomap, with a new algorithm for multi-stroke sketch recognition, which significantly outperforms the standard feature-based techniques.

INTRODUCTION

Sketching is a natural form of input for many domains, such as drawing mathematical symbols, graphs, binary trees, finite state machines, electrical circuit diagrams, military course of action diagrams and many more. As in other recognition domains (such as image and speech), feature selection is crucial for efficient and qualified performance in sketch recognition. Previous research has provided several suggestions for good feature sets. Rubine suggested 13 features based on drawing-style characteristics and time (Rubine 1991) and Long suggested 22 features modifying Rubine's (Long *et al*. 2000). Such hand-selected feature sets were well designed, but they are based on manual entry of methods to extract them, which becomes tiresome. Moreover, feature design is sensitive to problems such as jitters. More interestingly, machine learning research in other domains have shown that computers are able to select their own features and often times provide clear advantages in classification.

To our knowledge, however, manifold learning has not yet been applied to sketch recognition despite evidence that has shown manifold learning methods to be good nonlinear feature extractors. Manifold learning is an effective method for representation that works by recovering meaningful low dimensional structures hidden in high-dimensional data (Seung & Lee 2000). In this work, we apply the kernel Isomap manifold learning method (Choi & Choi 2007) to classify sketch data, because it has a projection property which provides the ability to project (map) new test data into (onto) the same feature space as training data. Kernel Isomap requires a dissimilarity matrix to find a low-dimensional mapping from which it produces a feature set. We introduce a new algorithm to measure dissimilarity distance between shapes that is based on both spatial and temporal information. We also show how this algorithm can be modified to accommodate for shapes drawn with multiple strokes. We compare our method with the widely used Rubine feature-based method for recognizing shapes (Rubine 1991).

IMPLEMENTATION

In order to use manifold learning for recognition, we need a projection property (or generalization property). A projection property provides the ability to project new test data into the same feature space as training data. In contrast to Isomap or LLE, kernel Isomap provides the projection property itself. Thus, we chose to use the kernel Isomap manifold learning technique. See (Choi & Choi 2007) for the algorithm and the projection property.

Dissimilarity from Sketch Data

To apply kernel Isomap to sketch classification, we need to use the time structure in data, as well as, the generalization property. In a kernel Isomap, what we need is not the data points, but a dissimilarity matrix.

The first step in our algorithm is to scale each character to be the same width and height. Then, we need a consistent (and large) number of points in each sketch. Thus, we interpolate points between any two consecutive points that are adjacent in time, but far away in distance to ensure that each gesture has the same number of points. Some characters such as '4' and '5' are drawn using multiple strokes. We simply connect and interpolate between the last point of the previous stroke and the first point of the next stroke.

To calculate the dissimilarity matrix, we calculate the sum of squared distance between two points of the same order in each sketch. The dissimilarity between ith sketch data and jth sketch data, D_{ij}, is then given by

$$D_{ij} = \sqrt{\sum_{k=1}^{S} (d_k)^2}, \tag{1}$$

where d_k is the Euclidean distance between kth points of two sketches, and S is the number of points in one stroke. In addition to normalizing each stroke by resampling, we use different weights for different points. When we connect the

two parts of '4' or '5', the connection causes confusion in calculating the distance. If we use different weights so that the fake points have less importance than real parts of the stroke, we can eliminate some of the confusion.

EXPERIMENTS

We carried out numerical experiments with three different data sets: (a) 26 small letters in English written by one user in one stroke with each letter having 25 data points; a third of the examples of each letter were drawn two weeks later, (b) 10 digits from 0 to 9, drawn by another user, where each class has 25 characters and (c) 8 different mathematical symbols ('+', '-', 'x', '/', '=', 'sin', 'cos' and 'tan'), drawn by 10 subjects, where each symbol was drawn 5 times in a single stroke. In order to show the improved accuracy of our method, we compared our method with Rubine's algorithm. Table 1 shows the hit rates of classification in three data sets, respectively, by three approaches: (1) Rubine's method, (2) kernel Isomap (K-Isomap) and (3) weighted K-Isomap (WK-Isomap). To make it fair, we extracted 13 features from the (W)K-Isomap[1] as Rubine's.

Table 1: Averages of the hit rates of 50 times cross validation with kNN classifier.

Methods	Alphabets	Digits	Math Symbols
Rubine's	77.31%	91.80%	60.46%
K-Isomap	87.19%	95.90%	92.77%
WK-Isomap	87.04%	97.80%	92.41%

Figure 1 shows us how good the features are. We plotted the 2 best features from the English letters data set. We can see that in K-Isomap, data points in the same class are grouped together. (W)K-Isomap is better than Rubine's for the classification because it tries to find the best features to represent the data set in an efficient way, whereas Rubine's features are already determined. We calculated the correlation matrix of each feature set (see Figure 2). We can see that the Rubine features have a large amount of redundency, whereas the (W)K-Isomap features are uncorrelated and more efficient.

CONCLUSION

In this paper, we developed a new technique for gesture classification using a manifold learning approach that combines temporal and spatial information, while handling multiple strokes with weighted distances. Experimental results confirmed the performance to be better than the Rubine feature set.

ACKNOLEDGEMENTS

This work was supported by KOSEF Grant-D00115. This work 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.

[1](W)K-Isomap means K-Isomap and WK-Isomap

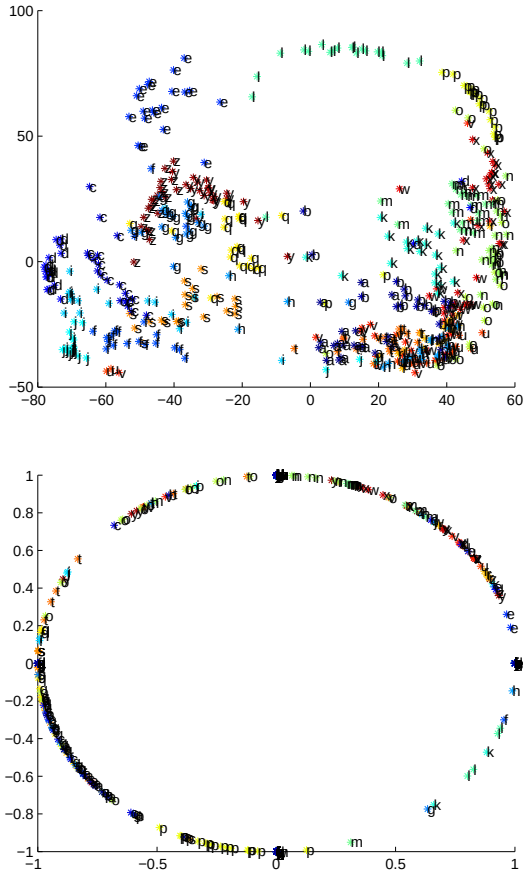

Figure 1: The features of K-Isomap (top) and Rubine method (bottom): Each axis means one feature.

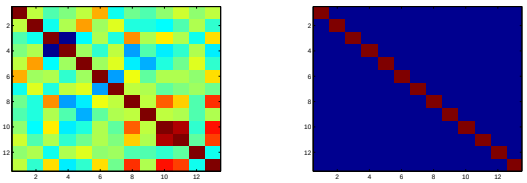

Figure 2: Correlation matrices between 13 features (Red color means high correlation): (Left) Rubine Features; (Right) (W)K-Isomap Features.

References

Choi, H., and Choi, S. 2007. Robust Kernel Isomap. *Pattern Recognition* 40(3):853–862.

Long, A. C.; Landay, J. A.; Rowe, L. A.; and Michiels, J. 2000. Visual similarity of pen gestures. In *Human Factors in Computing Systems*.

Rubine, D. 1991. Specifying gestures by example. *Computer Graphics* 25(4):329–337.

Seung, H. S., and Lee, D. D. 2000. The manifold ways of perception. *Science* 290:2268–2269.