USER-INDEPENDENT RETRIEVAL OF FREE-FORM HAND-DRAWN SKETCHES

Wing Ho Leung and Tsuhan Chen

Carnegie Mellon University, 5000 Forbes Avenue, Pittsburgh, PA 15213, USA

Email: {wingho, tsuhan}@andrew.cmu.edu

ABSTRACT

In this paper we propose a method to retrieve free-form hand-drawn sketches stored in the form of multiple strokes, by extracting the shape information for each stroke and by considering the geometric relationship between the strokes. To extract the shape information, a number of shape estimators are applied to each stroke to provide a soft decision about how similar it is to a particular shape type. Then, two strokes are matched according to a specific set of features for each shape type. The proximity of the corresponding strokes is used to account for the geometric relationship between multiple strokes during the matching stage. Our approach is robust to different drawing styles, thus making our retrieval system user-independent. Sketch retrieval is useful in applications where a user can easily search through a database of hand-draw sketches by inputting a sketch about what he/she is looking for, without the trouble of describing it using keywords.

1. INTRODUCTION

Recently pen-based devices such as Personal Digital Assistants (PDA) and electronic whiteboards have become more and more common to the general public. Using a pen as the input device provides the user a more natural way of interaction compared with other input devices such as the keyboard. As a result, this motivation opens up a new field of research for improving pen-computing technologies in order to justify the use of these pen-based devices. For example, in the classroom, the lecture notes written by the teacher on the whiteboard can be captured electronically by a collaboration device such as "mimio" [1]. Later students can retrieve relevant lecture materials from the hand-drawn sketch database by sketching a query drawing. Each sketch (hand-drawing) in the database consists of strokes that are sequences of coordinates of the points sampled by the pen-based device. Retrieval in the hand-drawn sketch database is equivalent to finding a stroke or multiple strokes from the database that are a good match to the query stroke(s). This is different from hand-written recognition, since hand-written recognition deals with a limited set of alphabets while the hand-drawn sketch domain is a more general set that allows unstructured free-form hand-drawings. Besides, the goal of hand-written recognition is to recognize the context of the hand-writing by mapping it to a set of pre-defined alphabets. On the other hand, the goal of hand-drawn sketch retrieval is to find a hand-drawing from the database that is similar to the query.

Lopresti et el [2][3] reported their work on matching hand-drawn pictures which they call "pictograms". In their ScriptSearch algorithm, the electronic ink is broken into small chucks at local *y*-minima. The features such as the length of the stroke, the total angle traversed are extracted to form the feature vector. Vector quantization is performed to map the feature vector into one of the 64 symbols. The resulting hand-drawing becomes a string of symbols and dynamic programming can be used to compute the distance between strings. The data used in their experiment is hand-written text. This approach has the drawback that it treats the same hand-drawings with different stroke orders as a poor match. Besides, the algorithm is sensitive to different writing styles. In order to make the system less sensitive to the stroke order, Lopresti and Tomkins [4][5] proposed to match the strings block by block. Under the modified scheme, the string is divided into blocks which are used as the basic unit for matching between the query and the database. This modification allows the matching of the hand-drawings with arbitrary drawing order of the strokes. However, poor match will still result if a stroke is drawn in reverse direction (i.e., when the start point and the end point of a stroke interchange). As a result, there is a certain degree of user-dependency in their retrieval scheme.

Kamel and Barbaras [6] proposed a two-stage scheme for stroke retrieval. In the first stage global features are used to filter out unmatched hand-drawings and a list of candidates from the database is generated. In the second stage, a sequential algorithm (similar to the one used by Lopresti and Tomkins [2]) is performed on the candidate set to find the best *k* matches to the query. Their experimental data is cursive handwriting and they did not perform any experiments on free-form hand-drawings.

In this paper, we propose a sketch retrieval method for general unstructured free-form hand-drawings. One goal of this approach is to support different human drawing styles by extracting the semantic shape information to match a stroke with some basic types (lines, circles and polygons), thus making it user-independent. In [7], geometric shapes, namely rectangles, ellipses, circles, diamonds, triangles and lines, are recognized from a hand-drawing. This is different from our approach in two ways. Firstly, in our approach, instead of trying to recognize a shape, a confidence value is computed for each shape type in order to provide a soft decision when two strokes are compared. Secondly, the shapes that can be recognized using the approach in [7] have limited orientation. We are able to provide a high confidence value to detect shapes with more arbitrary orientations. Our approach exploits the geometric relationship between multiple strokes for matching. The proximity of the corresponding strokes is used to account for the geometric relationship between multiple strokes during the matching stage.

This paper is organized as follows. In Section 2 we provide the system description of our approach. In Section 3 we describe our experiment and presents the results. The conclusions and future work are in Section 4.

2. SYSTEM DESCRIPTION

Figure 1 shows the system diagram of our approach:

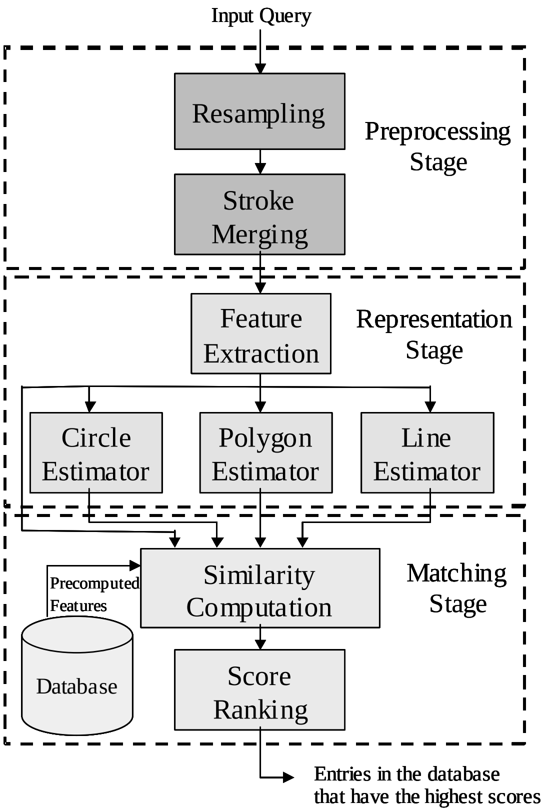


Figure 1 System diagram of our approach

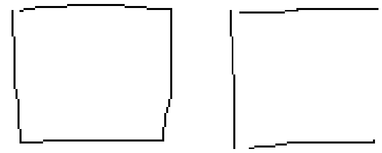
2.1 Preprocessing Stage

2.1.1 Resampling

The input query, a single stroke or multiple strokes, is resampled to 256 points according to the arc-length. When the input query consists of multiple strokes, the number of resample points are distributed according to the proportion of the arc-length of each stroke. This resampling process reduces inconsistencies due to different writing speed.

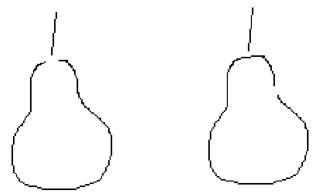
2.1.2 Stroke Merging

There can be various ways of combining strokes to form a similar sketch. For example, as shown in Figure 2, when drawing a square, a person may draw it as one stroke (Figure 2(a)), or draw it as four line segments (Figure 2(b)). In order to account for these different styles, two strokes are merged into one stroke if the distance between a start/end point of a stroke and a start/end point of the other stroke is small. However, if the start and end points of the same stroke are already very close, i.e., a closed loop, then this stroke should not be merged with other strokes because the location of the start and end points of a closed-loop stroke can be arbitrary, as shown in Figure 3.



(a) one stroke (b) 4 line segments

Figure 2 A sketch formed by different combination of strokes



(a) (b)

Figure 3 A sketch containing a closed-loop stroke formed by different location of start and end points

2.2 Representation Stage

Features are extracted from each stroke. These features are used for the shape estimators to determine the likelihood that each stroke falls in each basic shape type: line, circle and polygon. A confidence measure that takes the value between 0 and 1 is assigned for each stroke with respect to each shape type.

2.2.1 Feature Extraction

Some basic features are extracted from each stroke such as the center of the stroke, the perimeter, the area, the convex hull, etc. Other features are computed by combining the basic features. For example:

1) Perimeter efficiency $k = \frac{2\sqrt{\pi A}}{p}$, where A is the area and p is the perimeter.

2) The ratio between the area formed by the original stroke and the area of its convex hull.

3) The ratio between the number of points of the convex hull and the number of point of the original stroke samples.

2.2.2 Circle Estimator

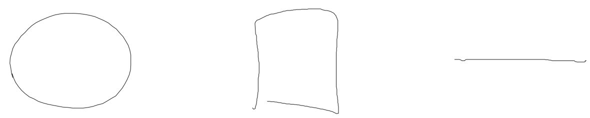
To estimate the confidence of a stroke being a circle, we use the following properties:

1) The perimeter efficiency of a circle is 1.

2) A circle requires a large number of points (ideally every sample) to form the convex hull that surrounds all the stroke samples.

3) The stroke samples go 360° around the center of the stroke.

Figure 4 shows the confidence output of the circle estimator for some stroke examples.



(a) c_{circle} = 0.8536 (b) c_{circle} = 0.6806 (c) c_{circle} = 0.273

Figure 4 Some sketches and their confidence values from the circle estimator

2.2.3 Polygon Estimator

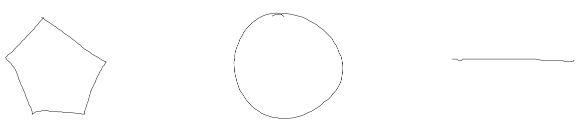
To estimate the confidence of a stroke being a polygon, we use the following properties:

1) The ratio between the area formed by the stroke samples to the area of the convex hull is close to 1.

2) A polygon requires a relatively few number of points (ideally equal to the number of vertices) to form the convex hull that surrounds all the stroke samples.

3) The stroke samples go 360° around the center of the stroke.

Figure 5 shows the confidence output of the polygon estimator for some stroke examples.



(a) $c_{polygon}$ = 0.9772 (b) $c_{polygon}$ = 0.5390 (c) $c_{polygon}$ = 0

Figure 5 Some sketches and their confidence values from the polygon estimator

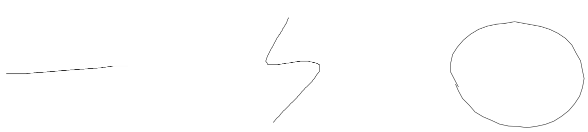
2.2.4 Line Estimator

To estimate the confidence of a stroke being a line, we use the following properties:

1) Consider the triangle formed by the two end points and each sample point of the stroke, the height is small compared to the base.

2) The ratio between the sum of distances of each pair of neighboring points and the distance between the end points is close to 1.

Figure 6 shows the confidence output of the line estimator for some stroke examples.



(a) c_{linen} = 0.9953 (b) c_{line} = 0.4958 (c) c_{line} = 0

Figure 6 Some sketches and their confidence values from the line estimator

2.3 Matching Stage

In the matching stage we first compute a matching score between the query and each of the sketch in the database, then the results are retrieved according to the descending ranking of the matching scores.

2.3.1 Matching between two strokes

Two strokes are matched by first computing the matching score of each corresponding shape type of the two strokes and then choosing the maximum matching score among all the shape types. The matching score for a shape type between two strokes is computed by finding the proximity between the features of that shape type as defined in the previous section, weighted by their confidence values. For example, if the two strokes to be matched are similar to a circle, then the product of the confidence of the circle estimator is high. In addition, if the two strokes are similar to each other, then it will yield a high similarity between their circle features. The resulting equation of the matching score between two strokes is shown in the following:

$$MS(s_{Qi}, s_{Dj}) = \max_p \left\{ c_p(s_{Qi}) c_p(s_{Dj}) \prod_k G(f_{p,k}(s_{Qi}), f_{p,k}(s_{Dj})) \right\}$$

where s_{Qi} is the i-th stroke of the query, s_{Dj} is the j-th stroke of the sketch from the database, p is the shape type which can be line, circle, polygon or non-basic type, $c_p(s)$ is the confidence of the estimator of the shape type p for the stroke s; $f_{p,k}(s)$ is the k-th feature from the estimator of the shape type p about stroke s; $G(f_1,f_2)$ is the similarity measure between features f_1 and f_2. The non-basic type is one of the shape types and its confidence value is derived from the confidence values of the other basic shape types. The confidence value of the non-basic type is high when the confidence values of all the basic shape types are low. The features used for the similarity measure for the non-basic type are taken from each of the estimators of the basic shape type.

2.3.2 Matching between two sketches

Since a sketch may consist of multiple strokes, the geometric relationship between the multiple strokes of a sketch should also be considered for matching two sketches. In our system, for each stroke in the query sketch, the stroke in the sketch from the database that yields the maximum matching score is weighted inversely by the distance between the center of the two strokes. The resulting matching score is obtained by summing up the distance-weighted matching scores for all the strokes, minus a cost for unmatched strokes. There are two cases of unmatched strokes: 1) no match is found in the sketch from the database against a stroke in the query; 2) a stroke in the sketch from the database that does not match any of the strokes in the query sketch. Summarizing, the matching score is determined as follows:

$$MS(s_Q, s_D) = \sum_i \max_j \left\{ \frac{MS(s_{Qi}, s_{Dj})}{D(s_{Qi}, s_{Dj})} \right\} - \text{unmatched cost}$$

where s_Q is the query sketch, s_D is the sketch from the database, $D(s_{Qi}, s_{Dj})$ is a distance measure between the i-th stroke of the query, and the j-th stroke of the sketch from the database.

3. EXPERIMENTS AND RESULTS

We perform an experiment to analyze the retrieval performance based on our approach. Our database consists of 35 categories of free-form sketches. Initially, for each sketch we have 20 repetitions made by 4 different people to account for the different drawing styles. Eight months later, the same people are asked to redraw the sketches 5 more times per person. Figure 7 shows a few examples of these sketches in the database. Figure 8 shows example sketches drawn by the same person at two different time instants (8 months apart). There is significant variation in the drawings even if they are drawn by the same user. We then compute the matching scores between a query and the sketches from the database. Based on the rank of those sketches that fall in the same category, we plot the precision and recall graph [8] in order to analyze the retrieval performance. The result is shown by the two curves in Figure 9. The top curve corresponds to the retrieval performance when each of the sketches from the initial collection is used as the query. The bottom curve corresponds to the retrieval performance when each of the sketches drawn 8 months later is used as the query, matching with the initial collection of the sketch database. At the recall rate of 0.5, the precision rate is higher than 0.8 for both cases. This means that as we keep retrieving sketches from the database, when 50% of the sketches from the same category as the query (relevant sketches) are retrieved, 80% of the total retrieved sketches are relevant sketches. It can also be seen from Figure 9 that the two sets of query yields similar retrieval performance, showing that our system is robust to variation over time.

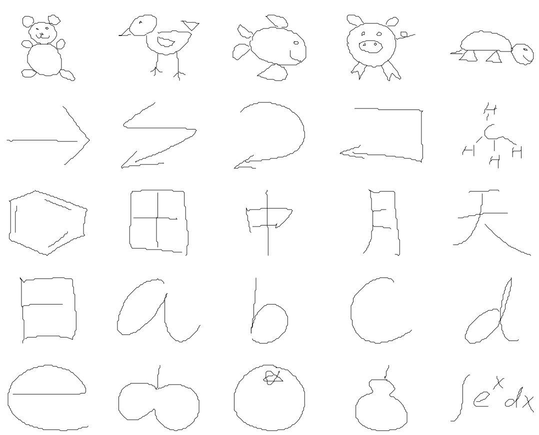


Figure 7 Some examples of the sketches in the database

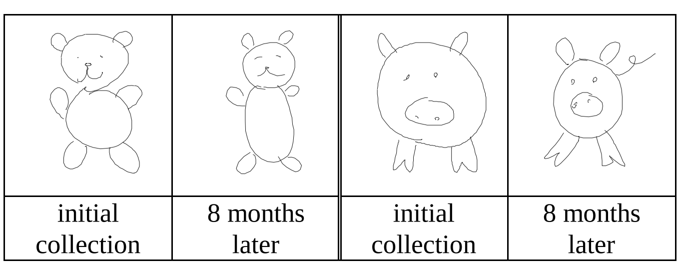


Figure 8 Example sketches drawn by the same user at two different time instants (8 months apart)

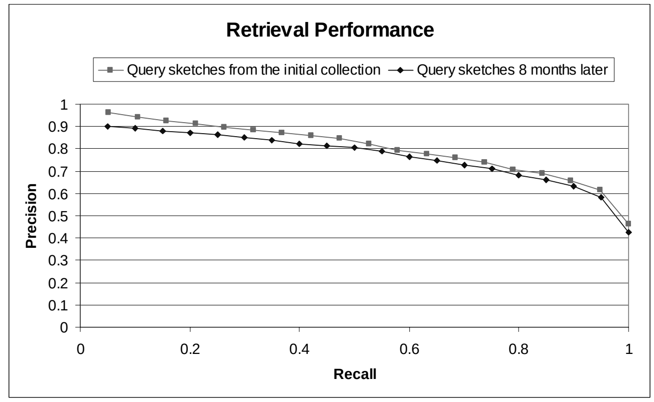


Figure 9 Retrieval Performance by Precision and Recall graph

4. CONCLUSIONS AND FUTURE WORK

This paper presents our system for retrieving free-form hand-drawn sketches by using soft decision shape estimators and geometric relationship between multiple strokes. Experiments show that our algorithm works well to retrieve user-independent sketches. For future work, we would like to extend the shape types to include more shapes. Moreover, we will perform more studies about which features to be used for each shape type. Currently we use heuristics to generate the confidence value of each shape type. In the future, we will perform statistical analysis based on training data. We will also improve the stroke merging algorithm to increase the retrieval performance.

5. REFERENCES

[1] Virtual Ink Corporation, mimio. [Online]. Available: http://www.mimio.com/

[2] Lopresti D. and Tomkins A. "On the searchability of electronic ink". In *Proc. of the Fourth Intl. Workshop on Frontiers in Handwriting Recognition*, pp. 156-165, December 1994.

[3] Aref W., Barbara D., Lopresti D. and Tomkins A. "Ink as a first-class datatype in multimedia databases". In S. Jajodia and V. S. Subrahmanian, editors, *Multimedia Databases*. Springer-Verlag, New York, 1995.

[4] Lopresti D. and Tomkins A. "Temporal domain matching of hand-drawn pictorial queries". In *Proc. of the Seventh Conf. of The Intl. Graphonomics Society*, pp. 98-99. August 1995.

[5] Lopresti D., Tomkins A. and Zhou J. "Algorithms for matching hand-drawn sketches". In *Proc. of the Fifth Intl. Workshop on Frontiers in Handwriting Recognition*, pp. 233-238, September 1996.

[6] Kamel I. and Barbara D. "Retrieving Electronic Ink by Content". In *Proc. of the Intl. Workshop on Multimedia Database Management Systems*, pp. 54-61, 1996.

[7] Apte A., Vo V. and T. D. Kimura. "Recognizing Multistroke Geometric Shapes: An Experimental Evaluation". In *Proc. of the Sixth ACM Symposium on User Interface and Software Technologies*, pp. 121-128, 1993.

[8] van Rijsbergen C. J. *Information Retrieval,* Butterworths, London, 1979.