HIERARCHICAL MATCHING FOR RETRIEVAL OF 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

As pen-based devices become more popular, it has motivated several novel research issues in the area of pen computing. One interesting and practical problem is the storage and retrieval of hand-drawn sketches. A sketch can consist of handwritten notes, symbols, free-form hand-drawings, annotations on a document etc. It will be very useful to store the sketches in a database and then retrieve them later. In our prior work, we proposed a method for hand-drawn sketch retrieval by performing stroke-by-stroke matching and by considering the spatial relationship between them. In this paper, we propose a method to simplify a sketch and perform matching in a hierarchical manner. With this approach, a shaded region or a region of complex strokes can be detected automatically and represented as a single hyper-stroke. In the matching stage, the similarity is computed by combining the similarity scores across the feature spaces at each level and the similarity of the stroke hierarchies. A sketch may be better represented by exploiting the structural relationship between the strokes. In addition, as a general rule to the hierarchical approach, it has the advantage of reduced computation thus the retrieval process would speed up.

1. INTRODUCTION

Pen computing has become more and more important in our society due to the popularity of pen-based devices such as TabletPC [1] that recently came out. Pen-based devices provide users with a natural way of input for drawing sketches. A sketch can consist of handwritten notes, symbols, free-form hand-drawings, annotations on a document etc. It will be very useful to store the sketches in a database and then retrieve them later by providing a simple sketch query. For example, in a classroom, the teacher may write and draw the lecture notes on the whiteboard that can be captured and stored in a database. Later students can retrieve relevant lecture sketches from the database by drawing a sketch as the query.

Query by sketch falls into the category of content-based image retrieval (CBIR). QBIC [2] was the first CBIR system and it also supports query by sketch. Global features such as area, circularity, eccentricity, etc., are used in shape matching. Matusiak et al. [3] proposed another approach to sketch-based images database retrieval by using Curvature Scale Space (CSS) to match contours. In Sciascio and Mongiello's system [4], the Fourier descriptors are used for shape comparison and they use relevance feedback to improve the retrieval performance for content-based image retrieval over the web. All the above systems assume that the query consists of a single shape.

Lopresti et al. [5][6] reported their work on matching hand-drawn pictures which they call "pictograms". This approach has a drawback that it treats the same hand-drawings with different stroke orders as a poor match. In order to make the system less sensitive to the stroke order, Lopresti and Tomkins [7][8] proposed to match the strings block by block. However, poor match may still result if a stroke is drawn in reverse direction (i.e., when the start point and the end point of a stroke interchange). Under these approaches, string matching is performed for the alignment based on the time sequence. They may work well for hand-writings or pen gestures [9] when the strokes have certain sequence pattern but may not be suitable for unstructured free-form hand-drawings.

The Query by Visual Example (QVE) reported by Kato et al. [10] used correlation of the corresponding blocks between the edge maps for evaluating similarity. Due to the variations in drawing style, this correlation approach will hardly match two rough sketches. Del Bimbo and Pala [11] proposed to use elastic matching to retrieve images from the database based on the user sketch. However, this energy minimization technique may be too time consuming when it requires many iterations for the solution to converge.

In our prior work [12], we proposed a retrieval method for hand-drawn sketches. It is based on string matching by the alignment of the spatial order among the boundaries of the minimum bounding rectangles of the strokes in each of the x and y projections. In [13], we included the similarity in spatial relations between strokes in the computation of the overall similarity score. We have introduced another application of query by sketch in trademark retrieval [14].

In this paper, we propose a novel retrieval method for simplifying the hand-drawn sketches and then matching them in a hierarchical manner. Sketch simplification is performed to detect shaded regions or regions with complex strokes, and represent each of them by a single hyper-stroke. For example, in Figure 1, there are two sketches of traffic light whose red light state is indicated by the shaded region. It can be seen that the strokes used to cover the shaded regions in these two sketches are quite different. As a result, it would be more appropriate to represent the shaded region by a single hyper-stroke containing features about that region. On the other hand, a stroke hierarchy is constructed for each sketch by exploiting the structural relationship between the strokes. In the matching stage, the similarity scores are computed for three feature spaces at each level of the stroke hierarchy. The purpose of matching in three different feature spaces is to consider various aspects of similarity including the overall region similarity, stroke similarity and spatial relation similarity. The final similarity score is determined by linearly combining the overall similarity scores for each feature space and the similarity of the stroke hierarchies.

Work supported in part by VACE/ARDA.

The advantages of hierarchical matching are: 1) a structural comparison is made possible by matching stroke hierarchies; and 2) the matching process can speed up.

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.

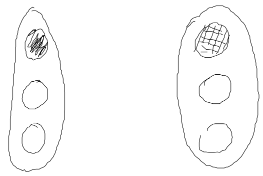


Figure 1 Example sketches with shaded regions

2. SYSTEM DESCRIPTION

Figure 2 shows the system diagram of our approach:

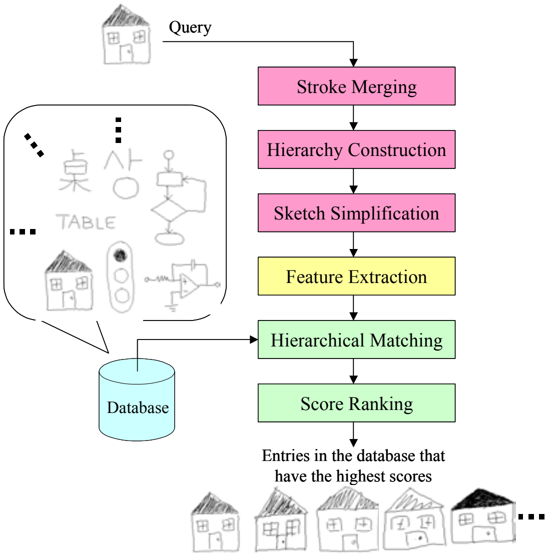


Figure 2 System diagram of our approach

When the user sketches a query, the 2-D *x* and *y* coordinates of each stroke of the sketch are captured. Firstly stroke merging is applied to reduce the effect of the variation in drawing style. Next the stroke hierarchy is constructed based on the structural relationship between the strokes. The sketch is then simplified to identify shaded regions such that all the strokes inside a shaded region will be represented by a single hyper-stroke. Afterwards three kinds of features are extracted from the strokes: hyper-stroke features, stroke features and bistroke features. In the matching stage, the features of the query are compared with the precomputed features from each entry in the database in a hierarchical manner to determine the similarity scores. These scores are sorted and the system will retrieve the entries in the database that have the highest similarity scores.

2.1 Stroke Merging

A single stroke may sometimes be drawn as smaller broken strokes. In order to account for these different styles, stroke merging is performed to merge broken strokes based on the proximity of the end points. A stroke is first checked whether its two end points are close. If they are not close, then that stroke is considered as open stroke and it may be merged with another open stroke if the end point of one stroke is close to the end point of the other. For example, in Figure 3(a), the two end points of one of the strokes are very close thus it is not an open stroke and the two strokes will not be merged. On the other hand, in Figure 3(b), the four strokes are open strokes and the end point of one stroke is close to the end point of another stroke, therefore these four strokes will be merged to become one stroke.

(a) Strokes not to be merged (b) Strokes to be merged

Figure 3 Stroke Merging

2.2 Hierarchy Construction

This module is responsible for building a stroke hierarchical representation for a sketch. The relationship between a parent stroke and a child stroke is that the child stroke is *inside* the parent stroke. The stroke hierarchy describes the structural relationship between the strokes in a sketch. For example, in drawing a house, the windows and the door are drawn inside the front part of the house; and the door knob is drawn inside the door. As a result, the corresponding stroke hierarchy is the one shown in Figure 4. In the actual implementation of this module, a quick bounding box test will be first performed between every pair of strokes to determine whether their bounding boxes overlap. If there is a significant amount of overlap between their bounding boxes, then a further convex hull test will proceed. In the convex hull test between two strokes, if most of the samples of a stroke fall inside the convex hull bounding the other stroke, then the first stroke is considered to be inside the second stroke. Under this rule, when there are two overlapping strokes, it is possible for each stroke to be considered inside the other stroke simultaneously. In this case, we will cancel out their effect by removing the "inside" relationships on both sides. After getting the "inside" relationship between the strokes, these relationships are examined to get the hierarchical information. For example, if stroke 3 and stroke 4 are both inside stroke 2; and if stroke 4 is also inside stroke 3, then stroke 2 is considered as the parent stroke of stroke 3; and stroke 3 will be considered as the parent stroke of stroke 4. A sketch of house and its corresponding stroke hierarchy are shown in Figure 4.

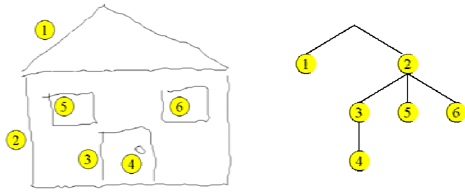


Figure 4 A sketch and its corresponding stroke hierarchy

2.3 Sketch Simplification

For a shaded region, a user may draw many strokes to describe the shade. It is better to consider this shaded region as one unit instead of considering each stroke separately. This is because two sets of strokes forming the same shaded region may not look similar as explained before with the examples in Figure 1. As a result, a shaded region is represented by a single hyper-stroke containing features describing the shaded region. The process of detecting and representing the shaded area as a single unit is called sketch simplification. A region consisting of a stroke with all its descendent strokes is more likely to be a shaded region if the ink density in that area is high. In order to determine whether a region has high ink density, we consider the total stroke length (the sum of the length of all the descendent strokes) and the convex hull area of the parent stroke. We plot these features from a set of training data in Figure 5, where the shaded region is characterized by large total stroke length with small convex hull area. The training data consists of shaded and non-shaded regions with both large and small areas as shown in Figure 5. The decision boundary is selected as follows:

$$x \underset{\text{non-shaded}}{\overset{\text{shaded}}{\gtrless}} Ay^2 + By + C$$

where x is the stroke length; y is the convex hull area; A, B and C are parameters determined from the training data. A second order function is chosen as the decision boundary because the training data cannot be well separated by a linear function.

After a region is decided as a shaded region, all those strokes forming this region will be replaced by a single hyper-stroke keeping only features about that region in the stroke hierarchy.

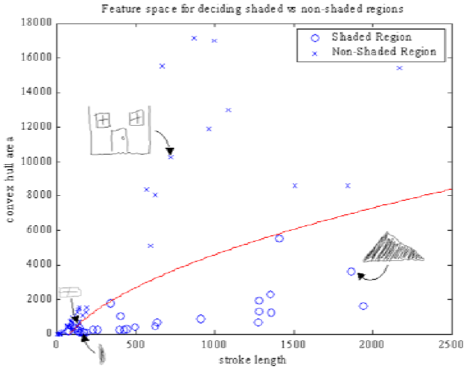


Figure 5 Feature space for deciding shaded vs non-shaded region

2.4 Feature Extraction

Three kinds of features are extracted from a sketch: hyper-stroke features, stroke features and bistroke features. The purpose of matching in three different feature spaces is to consider various aspects of similarity including overall region similarity, stroke similarity and spatial relation similarity.

2.4.1 Hyper-Stroke Features

A hyper-stroke consists of a parent stroke with all its descendent strokes. As described above, a hyper-stroke can also be used to represent a shaded region. The hyper-stroke features are the Hu moments and the histogram of edge directions. Hyper-stroke features provide an overall description about a region.

2.4.2 Stroke Features

Stroke features are geometric features extracted from each stroke. Different features are used to determine the likelihood that each stroke falls in each basic shape: line, circle and polygon. Some examples of the features are shown in Figure 6 [12].

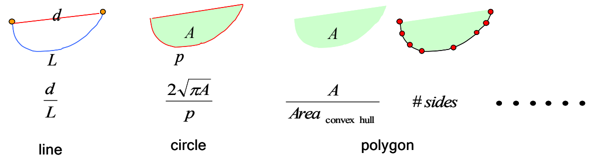


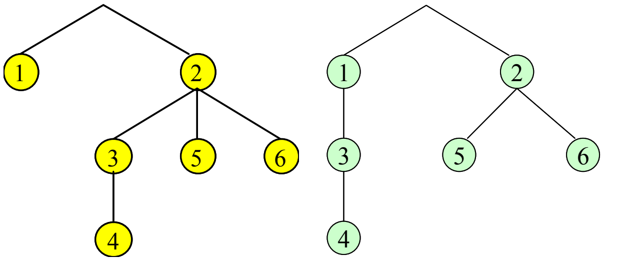
Figure 6 Example features

2.4.3 Bistroke Features

A bistroke feature consists of stroke features of a pair of strokes within the same level together with the spatial relation between them. The spatial relation is represented by the distance between the centroids of the minimum bounding rectangle of the pair of strokes, normalized by the sum of stroke length to make it scale invariant.

2.5 Hierarchical Matching

The similarity is evaluated in a top to bottom hierarchical manner. Starting at the top level, the similarity scores are computed for the hyper-stroke features, stroke features and the bistroke features at each level of the query stroke hierarchy. In the full hierarchical matching, all the bistroke features of the query are considered. On the other hand, to reduce computation, only bistroke features of the query from strokes within the same level are considered in the reduced hierarchical matching. The scores are first combined linearly across different levels for each of the three feature spaces. Afterwards the final similarity score is computed as the weighted sum of the scores from the three feature spaces and the similarity score from the stroke hierarchical structures. The similarity in the stroke hierarchical structures is determined by counting how many corresponding stroke pairs also preserve the parent-child relationship in the stroke hierarchies. For example, two stroke hierarchies are shown in Figure 7 where nodes with the same numbers are corresponding strokes. Between these two stroke hierarchies, three corresponding stroke pairs (2-5; 2-6 and 3-4) are also parent-child strokes in both hierarchies. As a result, the similarity in the stroke hierarchical structure in this case is 3.



(a) Stroke hierarchy 1 (b) Stroke hierarchy 2

Figure 7 Two stroke hierarchies

3. EXPERIMENTS AND RESULTS

For the data collection, a PDA device (Compaq iPaq Pocket PC) is used to capture the sketches. There are sketches from 11 people in our database. During each session, we ask each person to draw 1 to 3 repetitions for each of the 38 classes of sketches. The classes include Chinese characters, Korean characters, English words, mathematical equations, chemical structure, flow diagram and free-form hand-drawings. Figure 8 shows a few examples of different classes of sketches drawn by several people.

For the experiment, each of the sketches in the database is used as the query to retrieve other sketches within the same class. In the database, there are 722 sketches in total. For each of the 38 classes of sketch, there are 19 sketches in the database that belong to that class. For each query, we retrieve the elements from the database in the descending order of similarity scores.

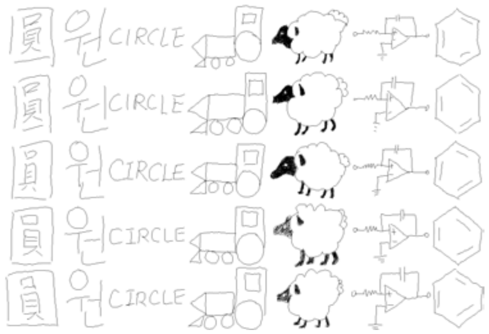


Figure 8 Some classes of sketches drawn by different people

To evaluate the retrieval performance, the precision and recall graph [16] is plotted based on the ranks of those sketches from the same class as the query sketch. The resulting graph shown in Figure 9 is obtained by averaging over all the queries. In a recall and precision graph, the higher the curve, the better the retrieval performance since for the same recall value, a higher curve signifies a higher precision value. We compare our retrieval result with several other approaches. From Figure 9, it can be seen that the retrieval performance by matching the edge histogram is very low. With the seven Hu moment invariants [17] as features, the retrieval performance is higher. By using the full hierarchical matching, the result is better than the previous two approaches. The performance with the reduced hierarchical matching is very close to the performance with the full hierarchical matching. On the other hand, in terms of computation time, conducting this experiment for all the query sketches with the reduced hierarchical matching is roughly two times faster than with the full hierarchical matching.

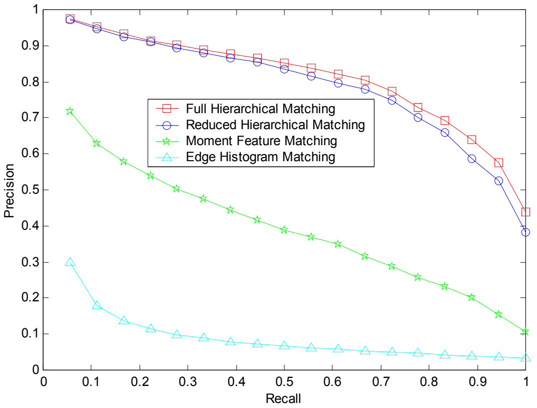


Figure 9 Retrieval Performance

4. CONCLUSIONS AND FUTURE WORK

We proposed a method for simplifying and retrieving hand-drawn sketches in a hierarchical manner. Sketch simplification allows the shaded region to be represented by a single hyper-stroke. Experiments show that the hierarchical matching outperforms some existing methods in terms of the retrieval performance. The processing time can be decreased with the reduced hierarchical matching whose retrieval performance is very close to the retrieval performance with the full hierarchical matching. For future work, we are working on further improving the hierarchical matching algorithm by exploring other possibilities of evaluating the similarity of the stroke hierarchy structures. In addition, we will also study relevance feedback to refine the retrieval result.

5. REFERENCES

[1] Microsoft XP TabletPC Edition. [Online]. Available: http://www.microsoft.com/windowsxp/tabletpc/

[2] Faloutsos C., Barber R., Flickner M., Niblack W., Petkovic D. and Equitz W. "Efficient and Effective Querying by Image Content". In *Journal of Intelligent Information Systems*, 3:231-262, 1994.

[3] Matusiak S., Daoudi M. and Blu T. "Sketch-Based Images Database Retrieval". In *Proc. of the Fourth Intl. Workshop on Multimedia Information Systems (MIS'98)*, pp. 185-191, Istanbul, Turkey, September 1998.

[4] Sciascio E. D. and Mongiello M. "Query by Sketch and Relevance Feedback for Content-Based Image Retrieval over the Web". In *Journal of Visual Languages and Computing*, 10(6), pp. 565-584, 1999.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] Long A. C., Landay, J. A., Rowe, L. A. and Michiels J. "Visual similarity of pen gestures". In *CHI 2000, ACM Conf. on Human factors in Computing Systems*, 2(1), pp. 360-367, April 2000.

[10] Kato T., Kurita T., Otsu N. and Hirata K. "A Sketch Retrieval Method for Full Color Image Database". In *Proc. of the 11th Intl. Conf. On Pattern Recognition*, pp. 530-533. The Hgues, The Neitherlands, August 1992.

[11] Del Bimbo, A and Pala, P. "Visual image retrieval by elastic matching of user sketches". In *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 19, No. 2, pp. 121-132, February 1997.

[12] Leung W. H. and Chen T. "User-Independent Retrieval of Free-Form Hand-Drawn Sketches". *IEEE Intl. Conf. on Acoustics, Speech, and Signal Processing*, Vol. 2, pp. 2029-2032, Orlando, FL, May 2002.

[13] Leung W. H. and Chen T. "Retrieval of Sketches Based on Spatial Relations Between Strokes". *IEEE Intl. Conf. on Image Processing*, Vol. 1, pp. 908-911, Rochester, NY, September 2002.

[14] Leung W. H. and Chen T. "Trademark Retrieval with Contour vs Skeleton Classification". *IEEE Intl. Conf. on Multimedia and Expo.*, Lausanne, Switzerland, August 2002.

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

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

[17] Hu M. "Visual Pattern Recognition by Moment Invariants". *IRE Trans. on Information Theory*, vol. IT-8, 1962.