Retrieval of On-line Hand-Drawn Sketches

Anoop M. Namboodiri and Anil K. Jain
Department of Computer Science and Engineering, Michigan State University
East Lansing, MI 48823, USA
{anoop,jain}@cse.msu.edu

Abstract

Sketch matching algorithms are commonly used for indexing and retrieval of documents based on printed or hand-drawn sketches. One could use a hand-held computer to do sketch-based queries to a database containing hand-drawn and printed sketches. We present an on-line hand-drawn sketch matching algorithm based on a line-based representation of sketches. A distance measure is defined for comparing two sketches based on this representation. The algorithm is computationally efficient and achieves a recall rate of 88.44% *at the same precision, when tested on a database of* 150 *sketches collected from* 5 *users.*

1. Introduction

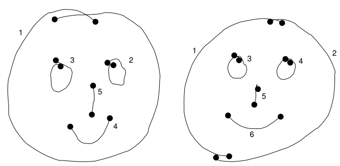
As handheld and portable devices such as PDAs, Pocket PCs and Tablet PCs, which accept handwritten data as input, are increasingly being used for communication and data entry, the number of handwritten documents available for storage and retrieval are also increasing. Many applications, which store and transmit handwritten data are available commercially [1]. Sketches are commonly present in handwritten documents, in the form of concept illustrations, flow charts, graphs, drawings of objects, annotations, etc. Hence, a document retrieval system, which allows sketches as queries can considerably improve the user's ability to retrieve relevant documents.

The problem of retrieving on-line handwritten data based on matching of digital ink (on-line handwritten data) has been studied before. Russell et al. [2] store the top N candidates of the recognition output of each word along with the handwritten data in the document database. Lopresti and Tomkins [3] use a stroke representation, where the handwritten data is divided into smaller segments at points of local minima of the y coordinates (vertical axis). Kamel [4] uses a set of features extracted from the strokes to represent the documents in a database. However, these studies deal with matching of handwritten words, where the order of writing of strokes is generally preserved. Matching of hand-drawn sketches poses a different set of challenges due to the large amount of variability among multiple instances of a figure. Lopresti et al. [5] have studied the problem and provided some initial results of experiments, which are encouraging. They describe the need for efficient algorithms to match components of sketches and utilize spatial arrangement of the components. Leung and Chen [6] represent the diagrams in each document using a feature vector, whose elements are confidence values corresponding to different primitive shapes derived from a shape estimator.

One could also retrieve printed or hand-drawn sketches and images using sketches as queries. Jain et al. [7] addressed the problem of image retrieval using a deformable template, which is a binary edge image. The matching process takes into consideration the energy required to deform the model and the goodness of fit of the deformed model to the target image based on gradient of the image. Del Bimbo and Pala [8] used a similar approach to match user drawn sketches to images in a database for retrieval. Manmatha et al. [9] used the edit distance to measure similarity between word shapes for retrieval. The problem of shape matching also involves the computation of a global transformation for aligning the two shapes. Belongie et al. [10] use a set of shape descriptors named 'shape context' to align two shapes for matching, which could be applied in the context of matching sketches.

The problem of matching on-line sketches is specifically interesting due to the challenges it poses, and its differences to off-line matching of shapes. On-line data captures the temporal sequence of strokes[1] while drawing a sketch. Most of the off-line sketch matching algorithms deal with the problem of matching a hand-drawn shape to an image. The inter-class variability of shapes in images are usually limited, since the images are formed from natural objects. However, in the case of hand-drawn sketches, the intra-class variability can be much higher due to a variety of reasons. They include variability due to drawing styles, draw-

1 A stroke is defined as the locus of the tip of the pen from pen-down to the next pen-up position.



(a) (b)

Figure 1. On-line sketch variability: Two similar sketches, drawn using different stroke orders and number of strokes. The numbers indicate the order of the strokes and the dots indicate the end points of the strokes.

ing sequence, overwriting or corrections, etc. We can convert the on-line sketch to an off-line image to avoid some of these variations. However, we will lose the ability to use the stroke information for efficient matching in that case.

2. On-line Sketch Matching

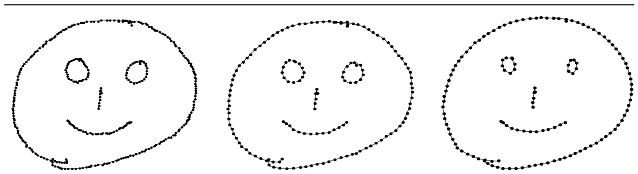
A sketch matching algorithm could compare two sketches using a distance measure such as dynamic time warping. However, such an approach will not be able to deal with the variability in hand-drawn sketches due to changes in the order of drawing the strokes. Figure 1 shows two similar shapes, drawn using different stroke orders and different number of strokes. In addition, a holistic matching of sketches would be computationally expensive for retrieval applications, where we need to compare a query sketch against a large set of sketches in a database. To deal with this problem, the sketch matching algorithm should be able to represent each sketch in a compact form, which can be compared efficiently and effectively. Representing a sketch using a set of global properties, such as total stroke length, perimeter efficiency etc., would be compact, but it cannot capture the salient details of all the possible sketches and hence, will not be an effective representation.

A common approach, used by Lopresti et al. [5] and Leung and Chen [6] is to represent a sketch in terms of a set of basic shapes. Lopresti et al. [5] refer to these basic shapes as 'motifs', whose identification and matching are not provided in detail in their paper. Leung and Chen [6] proposed a recognition-based approach for matching hand drawn sketches by identifying basic shapes such as circles, straight lines and polygons in the sketch. The problem of developing a set of basic shapes for representing a generic set of sketches is difficult at best. Once the basic shapes in a sketch are identified, one needs to compare the two sketches, based on the identified shapes. This involves a comparison of two shapes and their spatial arrangement in two sketches. We note that an exhaustive comparison of two sets of shapes to arrive at the best matching is exponential in time. We present a matching algorithm, which tries to overcome the problems with an efficient representation and similarity measure.

3. Proposed Algorithm

The matching algorithm consists of three stages. The pre-processing stage tries to eliminate the noise introduced in the sketches during data capture due to noise from the sensor, quantization noise, etc. In the second stage, we divide the traces in the sketch into smaller units. This is followed by a matching stage, where the sketches are compared based on their global properties and the properties of the individual portions of the strokes in the two sketches.

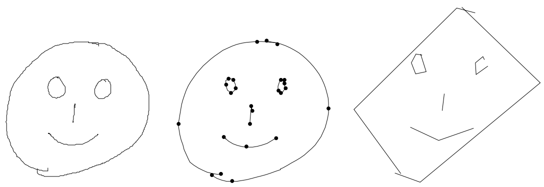
The input data for the matching algorithm can be noisy due to a variety of factors. These include the noise from the digitizer, noise due to pen vibrations, and inherent variabilities in the drawing process. The pre-processing stage tries to eliminate this noise to facilitate the matching process. During pre-processing, the strokes are resampled to make the sampled points equidistant. This helps to reduce the intra-class variations in the shapes due to different drawing speeds and to avoid anomalous cases such as having a large number of samples at the same position when the user holds the pen down at a point. The strokes are then smoothed using a Gaussian (lowpass) filter. This reduces the noise due to pen vibrations and errors in the sensing mechanism. Figure 2 shows an example of preprocessing an on-line sketch.



(a) (b) (c)

Figure 2. Pre-processing. (a) An input sketch. The dots represent the sampling points. (b) The sketch after equidistant resampling. (c) Result after lowpass filtering.

One of the basic problems in comparing sketches based on basic shapes is the identification of the basic shapes from a sketch. We overcome this problem by representing the sketches as a set of straight line segments instead of a predefined set of shapes. After pre-processing, each stroke is divided at the points where the x or y direction of a stroke changes. Each resulting stroke segment is represented as a straight line. Figure 3 shows the sketch in figure 2, where the stroke segments are identified and represented as a set of lines.



(a) (b) (c)

Figure 3. Segmentation of Strokes. (a) An input sketch. (b) The critical points in the sketch. (c) Representation of the sketch as a set of straight lines.

Let $\{\zeta_1, \zeta_2, \cdots, \zeta_n\}$ be the set of segments identified from a stroke ξ in the sketch S. A sketch with m strokes is hence represented as:

$$S = \xi_1 \cup \xi_2 \cup \cdots \cup \xi_m = \{\zeta_1, \zeta_2, \cdots, \zeta_k\}, \qquad (1)$$

where k is the total number of stroke segments identified from all the m strokes in the sketch. Each stroke segment ζ_i, is represented using its position, direction and length, $<P_i, D_i, L_i>$. Consider a stroke segment, ζ_i, containing p sample points.

$$\zeta_i =< (x_1, y_1), (x_2, y_2), \cdots, (x_p, y_p) > . \qquad (2)$$

The position, direction and length of the segment are computed as:

$$\begin{aligned} Position &= \left(\sum_{j=1}^{p} x_j/p, \sum_{j=1}^{p} y_j/p \right) \\ Direction &= arctan\left(\frac{y_p - y_1}{x_p - x_1} \right) \\ Length &= \sqrt{(}(x_p - x_1)^2 + (y_p - y_1)^2) \qquad (3) \end{aligned}$$

The value of direction, in degrees, is restricted to the range $[-180, 180]$ to avoid the difference between strokes drawn in opposite directions during comparisons. The value of position and length features are normalized using the size of the sketch ($length + width$).

The comparison of two sketches involves computing the best match between the two sets of stroke segments. The matching distance between two stroke segments is defined as a weighted euclidian distance.

$$\begin{aligned} d(\zeta_i, \zeta_j) &= d(< P_i, D_i, L_i >, < P_j, D_j, L_j >) \\ &= w_p.|P_i - P_j| + w_d.|D_i - D_j| + w_l.|L_i - L_j|, \quad (4) \end{aligned}$$

where $w_p, w_d,\ and\ w_l$ are the weights corresponding to each feature, which were determined empirically to be $5.0, 2.0\ and\ 5.0$.

The computation of the optimal match between two sets of line segments is exponential in terms of the number of lines in the sets. Let the number of stroke segments in the two sketched to be compared be p and q. An optimal matching requires $\frac{p!}{(p-q)!}$ comparisons. To reduce the time complexity of this computation, we use a greedy algorithm, which is possibly sub-optimal. In this method, we repeatedly select the most similar line segment pair between the two sets. The matching of two sketches takes only $O(p^2.q)$ comparisons with this algorithm. One could further refine this algorithm by incorporating heuristics for selecting the matching segment pair at each iteration, thus moving towards an optimal solution. However, the algorithm works well in practice as indicated by the experiments.

4. Results and Discussions

The database consists of 10 instances of 15 different sketches, collected from 5 users, with each user contributing two instances of each sketch over a period of two weeks. Figure 4 shows an example of each of the sketches in the database. The data was collected using the *CrossPad®* . The CrossPad has a pen and paper interface along with the ability to digitally capture the (x; y) position of the pen tip at a resolution of 254 dpi. The pen position is sampled at a constant rate of 132 Hz.

To compute the accuracy of the sketch matching algorithm, each of the sketches was compared against every other sketch in the database. The resulting distance matrix is used to analyze the matching performance. The equal error rate (point at which the precision equals the recall) is 11.56%. The results of matching revealed that most of the errors were committed by false matching between similar looking shapes such as the circle and the pentagon, where the line segment representations match well (see figure 5). The algorithm was also tested with leave-one-out cross validation, where one sketch from each class was randomly removed. The equal error rate for cross validation was approximately 11.7%, averaged over 10 trials.

The algorithm is also sufficiently fast to be used in real time searches of databases. The time to compare two sketches is approximately 25 $msec$ on a Pentium-III 650 MHz machine with 128 MB RAM. In practice the matching can be much faster as the pre-processing and feature extraction of the sketches in the database can be done off-line.

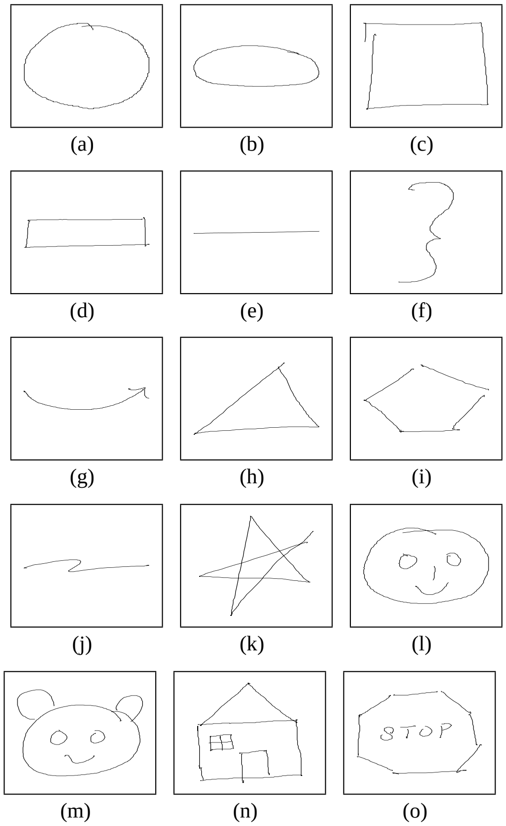


Figure 4. Examples from the on-line database.

5. Summary

We have presented an on-line sketch retrieval algorithm, which computes the similarity between two sketches based on a line-based representation of the sketches. The algorithm achieves a precision of 88.5% at the same recall rate. The representation allows us to overcome the difficulties associated with shape-based matching algorithms. We note that the high-level temporal features of the sketches are not very useful for the purpose of matching in a database of generic sketches. The algorithm can also be used for retrieval of printed or off-line hand-drawn sketches using an appropriate line detection algorithm.

We are currently working on extending the work to match sketches using only a part of the sketch as query. The algorithm could also be modified for rotation invariant matching. In addition, one could employ a matching stage based on global properties of sketches for indexing to improve the computational efficiency of the algorithm. We are also working on improving the matching accuracy by incorporating additional features to represent each stroke segment.

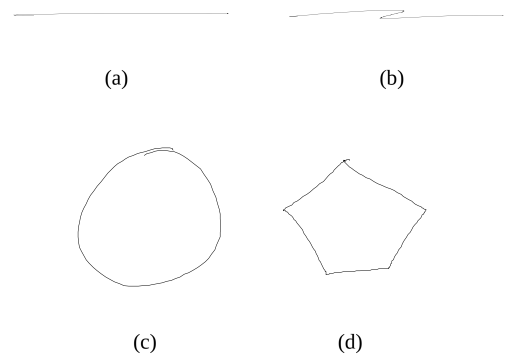


Figure 5. Examples of incorrect retrievals by the system. (a) and (c) were query sketches which were incorrectly matched to (b) and (d) respectively, by the algorithm.

References

[1] "riteMail note taking application," 2003. http://www.ritemail.net.

[2] G. Russell, M. P. Perrone, Y.-m. Chee, and A. Ziq, "Handwritten document retrieval," in *Proc. of the* 8^{th} *International Workshop on Frontiers in Handwriting Recognition*, (Ontario, Canada), August 2002.

[3] D. Lopresti and A. Tomkins, "On the searchability of electronic ink," in *Proc. of the* 4^{th} *IWFHR*, (Taipei), pp. 156–165, December 1994.

[4] I. Kamel, "Fast retrieval of cursive handwriting," in *Proceedings of the Fifth International Conference on Information and Knowledge Management*, (Rockville, Maryland), pp. 91–98, November 1996.

[5] D. Lopresti, A. Tomkins, and J. Zhou, "Algorithms for matching hand-drawn sketches," in *Proc. of the* 5^{th} *IWFHR*, pp. 233–238, September 1996.

[6] W. H. Leung and T. Chen, "User-independent retrieval of free-form hand-drawn sketches," in *Proc. of the IEEE ICASSP*, vol. 2, (Orlando, Florida), pp. 2029–2032, May 2002.

[7] A. K. Jain, Y. Zhong, and S. Lakshmanan, "Object matching using deformable templates," *IEEE Trans. on PAMI*, vol. 18, pp. 267–278, March 1996.

[8] A. Del Bimbo and P. Pala, "Visual image retrieval by elastic matching of user sketches," *IEEE Trans. on PAMI*, vol. 19, pp. 121–132, February 1997.

[9] R. Manmatha, C. Han, and E. Riseman, "Word spotting: A new approach to indexing handwriting," in *Proc. of the IEEE Computer Vision and Pattern Recognition Conference*, (San Francisco), pp. 631–637, June 1996.

[10] J. Belongie, Malik and J. Puzicha, "Matching shapes," in *Proceedings of the Eighth International Conference on Computer Vision*, vol. 1, (Vancouver, Canada), pp. 454–461, July 2001.