A Two-Stage Framework for Efficient Simple Polygon Retrieval in Image Databases *

Lun Hsing Tung, Irwin King, Ping Fu Fung and Wing Sze Lee
{lhtung, king, pffung, wslee}@cs.cuhk.edu.hk

Department of Computer Science and Engineering,
The Chinese University of Hong Kong,
Shatin, New Territories, Hong Kong

Abstract

We propose a two-stage framework for efficient polygon matching in image databases. The first stage performs a coarse polygon classification based on *qualitative* features of polygons and the second stage performs *quantitative* measure of polygons. We use Binary String Descriptor to quickly find equivalent classes of polygons in the first stage. In the second stage, we have two possible approaches: (1) a Multi-Resolution Area Matching technique based on Quad-Tree method and (2) the Hausdorff Distance method. The technique incorporated in the second stage will only operate on a subset of polygons belonging to the same equivalent class which is produced in the first stage. This two-stage framework can prune the search space of a polygon matching query and speed up the matching process. We have built an experimental system. We will also discuss the experimental results.

1 Introduction

Query-by-shape is a fundamental operation in an image database system. It provides intuitive way to access an object by its outline. Hence, shape representation, indexing, and matching are important issues for image databases.

Considerable work has been carried out on the shape manipulation problem. Some techniques use model-driven approach, in which the query shape is compared individually against each shape in the database. Other techniques use data-driven approach, in which shapes are mapped into some multidimensional index structures and matching is conducted by performing searching in the index tree [7, 8, 9].

In this paper, we propose a two-stage framework which performs polygon matching using an extended model-driven approach. It is augmented with method for reducing the number of polygons needed to be compared and thus improves the efficiency of model-driven based polygon matching. We perform shape matching in two stages. In the first stage, a qualitative and coarse shape classification method is employed to partition shapes into a number of classes. In this stage, polygons are represented by Binary String Descriptor (BSD) [3] and are partitioned into different equivalent classes with respect to their Standardized Binary Shape Descriptors (SBSDs). In the second stage, a quantitative and precise shape matching is employed to perform matching of shape within a shape class produced in the first stage. We have investigated two approaches on implementing this second stage. The first approach is to represent polygon by Multi-Resolution Are Information (MRAI) and similarity measure between polygons is performed using coarse-to-fine area

*This work is supported in part by Hong Kong's Industry Grant #AF/17/95 (2427 01300) and a RGC Grant # CUHK 485/95E(CU95513). A preliminary version of this work was presented at IMDB MMS.

based matching. In the second approach, polygon matching are performed using Hausdorff Distance as similarity measure. We choose to incorporate the Multi-Resolution Area Matching (MRAM) technique into our experimental system. The reason and the comparison between the two approaches will be presented later. Also note that it is the first stage of the framework that reduce the number of shapes to be compared and improves the efficiency of the matching process.

As this work is aimed at providing image database systems with *fast* query-by-shape facility, and it is sufficient to use approximated polygons instead of arbitrary shapes in shape processing of most current applications, we decided to concentrate on matching of simple polygons rather than arbitrary shapes since polygons are easier and less computational expensive to handle.

This paper is organized as follows. We will introduce the main idea of BSD in Section 2. In Section 3, we propose the Multi-Resolution Area Matching technique that can be use in the second stage of our framework. The issue of incorporating our approach into image database systems for providing query-by-shape facility will be discussed in Section 4. Comparison between the MRAM technique and Hausdorff Distance method, as well as the reason why MRAM is selected in our experimental system, is presented in Section 5. Conclusion is made in Section 6.

2 Binary String Descriptor

In our present work, instead of handling arbitrary shapes, we only handle closed, simple and non-degenerate polygons. We assume that a polygon is represented by a list of vertices: $P = \{V_1, V_2, \cdots, V_n\}$, where n is the number of vertices of the polygon and $V_i \in \mathbb{R}^2$.

2.1 Basic idea

A BSD [3] is a binary string recording the convexities and concavities of the vertices of a polygon. Let '0' denotes a convex vertex (interior angle less than π) and '1' denotes a concave vertex (interior angle larger than π).

Definition 1 *A Binary String Descriptor (BSD) is a string* $\{0,1\}^n$*, where* n *is the number of vertices of the polygon the descriptor is associated with.*

BSD is scale and orientation invariant since the measurement of convexity and concavity of a vertex is independent of these properties. However, the specific instance of the BSD of a polygon depends on the selection of the *anchor vertex* (the vertex of the polygon at which we start recording the BSD).

2.2 Standardizing Binary String Descriptor

A polygon can be represented by more than one BSD depending on the sequence of vertices being recorded. For example, a polygon represented by BSD '0010' can also be represented by '0100', '1000' or '0001', depending on the anchor vertex. The idea of standardizing BSD is introduced in [3] in order to obtain a unique BSD for a given polygon.

Given a BSD $B = \{0,1\}^n$, a rotated BSD B_i, for $1 \leq i \leq n$, is another BSD resulted by rotating the bits of B such that the ith MSB (Most Significant Bit) of B becomes the MSB of B_i. Let $M(B_i)$ denotes the magnitude of B_i regarding it as a binary integer.

Definition 2 *The standardized Binary String Descriptor (SBSD) of* B *is* B_j *such that* $M(B_j) = \min_i M(B_i), 1 \leq i \leq n$.

SBSD inherits the scale and orientation invariant properties from BSD and it is independent of the selection of anchor vertex.

2.3 Number of equivalent classes for n-gons

BSD function is a many-to-one mapping, i.e. more than one polygon may have the same BSD. Two polygons having the same SBSD are said to be in the same equivalent class. For polygons with n sides, there are 2^n possible BSDs. However, some of them are invalid and some are the same after standardization. For n-gons, the number of equivalent classes (E) is given in [3] as $E = \frac{1}{n}\sum_{m \in D_n} mX_n(m) - (\lfloor \frac{n}{2} \rfloor + 2)$ where D_n is the set of divisors of n, $X_n(m) = 2^{\frac{n}{m}} - (X_n(m_1) + \cdots + X_n(m_k))$ and $m_1, \cdots, m_k$ are the multiples of m belonging to $D_n \backslash \{m\}$.

Table 1 shows the number of equivalent classes for polygons with sides from 3 to 16. When the polygons being handled are with small number of sides, the numbers of equivalent classes are relatively small. Thus, SBSD may not be a good method for polygon classification in these situations.

A possible solution to this problem is to record the angle of a vertex in more discrete levels (rather than convex and concave only). For example, if 4 discrete levels are used ($0 < \theta \leq \frac{\pi}{2}$, $\frac{\pi}{2} < \theta \leq \pi$, $\pi < \theta \leq \frac{3\pi}{2}$ and $\frac{3\pi}{2} < \theta < 2\pi$), there will be 2 distinct equivalent classes for triangles instead of 1. If 8 discrete levels are used, then there will be 6 distinct equivalent classes for triangles. Thus, the number of discrete levels used to record the vertex angle can be chosen according to the number of sides of polygons a system is expected to process.

3 Multi-Resolution Area Matching

Once the equivalent class is found, the subset of polygons will then be matched by MRAM. We will now describe how Multi-Resolution Area Information (MRAI) of polygons is computed and how similarity between polygons is measured using MRAI.

3.1 Computing MRAI

A polygon, which is normalized to have a unit bounding box (Section 4.1), is first scan-converted onto a frame buffer with $W \times W$ pixels. MRAI is computed using a *Quad-Tree* [10] like approach:

1. MRAI is recorded starting at level 0.
2. At level 0, the whole frame buffer is regarded as a cell. The portion of area covered by the polygon is recorded.
3. At level k, cells are obtained by quartering every cell of level $k-1$. The portion of area covered by the polygon in each level-k cell is recorded. There are 4^k cells at level k.

The MRAI at each level is concatenated into a complete MRAI vector. The size of this vector depends on K, the maximum resolution level to be recorded, and is given as $L = \sum_{i=0}^{K} 4^i = \frac{4^{K+1}-1}{3}$.

3.2 Measuring similarity using MRAI

We use the L_p distance to measure the similarity of two polygons at a specific level of resolution. Given polygon A and B, with their MRAI, the similarity of these two polygons at resolution level k is:

$$S_k(A, B) = (\sum_{i=1}^{4^k} |A_{ki} - B_{ki}|^p)^{\frac{1}{p}}$$

where $S_k(A, B)$ is the similarity measure of A and B at resolution level k, A_{ki} and B_{ki} are the portion of covered area in level k cells of polygon A and B respectively, and $p = 2$ in our implementation.

Matching of two polygons can be done in stages, that is, perform similarity measuring from coarse resolution (level 0) to fine resolution (the maximum resolution level K, where $K = 3$ in our implementation).

Definition 3 *Two polygons A and B are said to be similar at level k if* $S_k(A, B) \leq \delta_k$ *where* δ_k *is a predefined threshold value for level k similarity measure.*

Definition 4 *Two polygons are said to be matched if they are similar at all levels, i.e. the two polygons are similar at level* $0, \cdots, K$.

4 Polygon Matching in Image Databases

4.1 Database population

When an image is added into the an image database, some preprocessing tasks are carried out. First of all, user has to define a number of polygons on the input image for future queries, if this image should be involved in query-by-shape operation. This task can be automated, by employing some shape segmentation algorithms such as [6, 2, 11], or the shape can be outlined by user manually. Each defined polygon will then go through the following preprocessings:

1. **Removal of collinear vertices** For any three successive vertices V_i, V_j and V_k of a polygon, V_j is removed if the three vertices are collinear such that the resultant polygon will be non-degenerate.

2. **Orientation normalization** We compute the BSD, B, of a polygon and then standardize B to obtain the SBSD, B_j, using the algorithm in Section 2.2. The representation of the polygon is then changed to $\{V_j, V_{j+1}, \cdots, V_n, V_1, \cdots, V_{j-1}\}$. After the re-arrangement of vertices, the coordinates of $V_1, \cdots, V_n$ are rotated regarding V_j as the origin such that the edge (V_j, V_{j+1}) is aligned with the y-axis.

3. **Scale normalization** We find out the bounding box of a polygon, then scale the bounding box to a unit square and transform the coordinates of $V_1, \cdots, V_n$ according to the same scaling factor.

4. **Polygon smoothing** Along the perceived edges of the polygon, there may be many jerks and trivial edges (e.g. when automatic shape segmentation algorithm is employed). We reduce the number of sides of the polygon by removing such disturbances and make the polygon smooth.

After the above preprocessing steps, we should have a smoothed, orientation and scale normalized non-degenerate polygon and it is ready for further manipulation. For each preprocessed polygon defined in an image, we compute its SBSD and MRAI as stated in Section 2 and Section 3. A tuple ⟨SBSD, MRAI, image⟩ is then added into database for future queries and retrievals.

4.2 Query-by-shape

This section describes how query-by-shape can be carried out by incorporating our framework into image database systems.

4.2.1 Initiating a query

To initiate a shape query, user may either specify a polygon by sketching it out or by selecting a shape from the database. For sketched shapes, automatic shape segmentation algorithm and shape preprocessing techniques, mentioned in Section 4.1, have to be employed in order to obtain a polygon. This polygon is regarded as the *target polygon* in remaining discussion. We then compute the SBSD and MRAI of the target polygon.

4.2.2 Exact matching query

Exact matching query refers to queries like "find me images containing objects having exactly this shape (a target polygon)". Exact matching query is carried out in following steps:

1. Select the set of polygons $Q = \{p_i \mid p_i \in P \ \wedge \ SBSD(p_i) = SBSD(T)\}$ where P is the set of polygons in the database and T is the target polygon.

2. Select the set of polygons $R = \{p_i \mid p_i \in Q \ \wedge \ matched(p_i, T)\}$ where $matched(p_i, T)$ denotes the predicate which measures the similarity between p_i and T using the algorithm proposed in Section 3.2.

3. The set of images in the database containing polygons in R is the result of the query.

4.2.3 Similar matching query

Besides Exact matching query, users of image database need to perform Similar matching query like "find me images containing objects having similar shape as this one (a target polygon)". Similar matching query is carried out in the following steps:

1. Select the set of polygons $Q = \{p_i \mid p_i \in P \ \wedge \ SBSD(p_i) = SBSD(T)\}$ where P is the set of polygons in the database and T is the target polygon.

2. **for** $i = 0$ **to** K **do**
    sort Q in descending order of $S_i(p_j, T)$ where $p_j \in Q$
    $Q \leftarrow \{p_j \mid p_j \in Q, 1 \leq j \leq N_i\}$
    /* $K = 3, N_0 = 100, N_1 = 50, N_2 = 25, N_3 = 10$ */
    /* in our implementation */
   end for

3. The set of images in the database containing polygons in Q is the result of the query.

Figure 2 shows how similar matching works in our experimental system.

5 Discussion

We use a two-stage framework for efficient polygon matching. Different matching techniques, both qualitative and quantitative, can be incorporated into the framework to replace the BSD and MRAM techniques that we used in our experimental system. Thus, when newer and better matching techniques are available, they can be incorporated into the framework in order to improve its performance.

We have investigated two approaches on implementing the second stage matching. One of them is the MRAM technique, which is introduced in Section 3. Another one is the Hausdorff Distance method. Hausdorff Distance has been used as shape matching technique [1, 5]. Basically, Hausdorff Distance is used to measure the difference between two sets of points.

Definition 5 *Given two finite point sets $A = \{a_1, \ldots, a_n\}$ and $B = \{b_1, \ldots, b_m\}$, the Hausdorff Distance is defined as*

$$H(A, B) = \max(h(A, B), h(B, A))$$

where

$$h(A, B) = \max_{a \in A} \min_{b \in B} ||a - b||$$

and $|| \cdot ||$ is some underlying norm on the points of A and B.

In our experiment, we use the L_2 (Euclidean) norm. The idea of Hausdorff Distance method for polygon matching is to measure the dissimilarity of two polygons by the Hausdorff Distance of the two vertex sets. Table 2 shows some characteristics of the Hausdorff Distance method and our MRAM technique.

- **Perceptual similarity**
  From observation, the two method both produce perceptual similarity of polygons similar to human perception. Fig. 1(a) and 1(b) show similarity ranking of polygons using MRAM technique and Hausdorff Distance method respectively.

- **Implementation difficulty**
  Both Hausdorff Distance method and MRAM technique is straight forward in idea and easy to implement. But Hausdorff Distance method is relatively easier to implement than MRAM.

- **Storage requirement**
  The two techniques have different storage requirements. Hausdorff Distance method requires polygon data (list of vertex coordinates) to be stored since it will be used, along with the polygon data of the target shape, to calculate the Hausdorff Distance in realtime. Thus, Hausdorff Distance method requires a $O(n)$ storage complexity per stored polygon where n is the maximum number of sides of polygons to be handled. In MRAM technique, we choose to store MRAI of a polygon instead of its polygon data. Therefore, MRAM technique requires a $O(L)$ storage complexity per stored polygon where L is the size of MRAI used in the system which is depended on the highest resolution used in MRAM. Though in general MRAI requires more storage space than polygon data, we can save a lot of computation by not computing MRAI in realtime. Our choice can be justified by the fact that storage is relatively cheap nowadays and fast response time is critical for interactive applications such as image database systems. Note that there is no way to compute Hausdorff Distance beforehand since it can be computed only when the target polygon is specified by user. Thus, Hausdorff Distance method does not share the advantage of saving computation time through preprocessing, as MRAM does.

- **Efficiency in terms of computation time**
  For a query-by-shape query, the computation complexity when Hausdorff Distance method and MRAM technique are used, are $O(mn^2)$ and $O(mL)$ respectively. It is hard to analysis the relative efficiency of the two methods as their complexities depend on different parameters. Yet, in average, the MRAI technique is more efficient as a result of its multi-resolution approach which terminates comparison in low resolution when a stored polygon is dissimilar to the target polygon. But for Hausdorff Distance method, you have to spend $O(n^2)$ computation to compute the Hausdorff Distance between a stored polygon and the target polygon no matter how similar, or dissimilar, the stored polygon is comparing with the target polygon. Our experimental system use an average time of 1.02 second for a query-by-shape query when Hausdorff Distance method is used. However, it only uses 0.67 second on average to handle a query-by-shape query when MRAM technique is used. The above statistic come from experiments based on databases containing 3000 polygons.

- **Indexing support**
  The fact that Hausdorff Distance must be computed in realtime also indicates that no indexing technique can be incorporated into Hausdorff Distance method since there is no preprocessed data to index. However, indexing techniques can easily be incorporated into the MRAM technique since MRAIs can be computed when the stored polygons are added into a database. This is why we claim that MRAM support indexing while Hausdorff Distance method does not.

Because we are applying our two-stage framework to image database systems and efficiency is a critical requirement for such systems, we choose to used MRAM technique in our experimental system instead of

Table 1: N-gons and number of their distinct equivalent classes

n	3	4	5	6	7	8	9	10	11	12	13	14	15	16
E	1	2	4	9	15	30	54	101	181	343	624	1173	2183	4106

Hausdorff Distance method. Not only that the MRAM is more efficient, in terms of computation time, than the Hausdorff Distance method in average, but also that MRAM has the ability to cooperate with indexing techniques which can further improve the overall system performance.

6 Conclusion

We proposed a two-stage framework for matching closed, simple, and non-degenerate polygons using an improved model-driven approach. This method can be used to provide query-by-shape facility in image database systems. Our approach incorporated Binary String Descriptor, Multi-Resolution Area Matching technique, and Hausdorff Distance method for qualitative and quantitative measure of polygons. From our empirical experiments, systems using Multi-Resolution Area Matching technique are more efficient than those using Hausdorff Distance method.

References

[1] H. Alt, B. Behrends, and J. Blömer. Approximate Matching of Polygonal Shapes. *Annals of mathematics and Artificial Intelligence*, Vol. 13:pp. 251–265, 1995.

[2] D. H. Ballard. Generalizing the Hough Transform to Detect Arbitrary shapes. *Pattern Recognition*, Vol. 13(No. 2):pp. 111–122, 1981.

[3] B. Bhavnagri. A Method for Representing Shape Based on an Equivalent Relation on Polygons. *Pattern Recognition*, Vol. 27(No. 2):pp. 247–260, 1994.

[4] E. Bribiesca and A. Guzman. How to Describe Pure Form and How to Measure Differences in Shapes Using Shape Numbers. *Patter Recognition*, Vol. 12:pp. 101–112, 1980.

[5] D. P. Huttenlocher and W. J. Ruckidge. A Multi-Resolution Technique for Comparing Image Using the Hausdorff Distance. *Technical Report TR92-1321, Department of Computer Science, Cornell University.*

[6] J. Illingworth and J. Kittler. The Adaptive Hough Transform. *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. 9:pp. 690–698, 1987.

[7] H. V. Jagadish. A Retrieval Technique for Similar Shapes. *Proc. of the ACM SIGMOD International Conference on the Management of Data*, pages pp. 208–217, May 1991.

[8] R. Mehrotra and J. E. Gray. Feature-Based Retrieval of Similar Shapes. *Proc. 9th International Conference on Data Engineering*, pages pp. 108–115, 1993.

[9] R. Mehrotra and J. E. Gray. Similar-Shape Retrieval in Shape Data Management. *IEEE Computer Magazine*, pages pp. 57–62, September 1995.

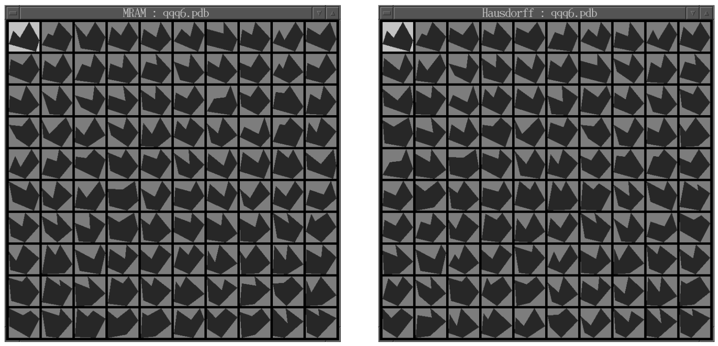
[10] H. Samet. *Design and Analysis of Spatial Data Structure, Addison-Wesley*, 1989.

[11] L. Su, E. Oja, and P. Kultanen. A New Curve Detection Method: Randomized Hough Transform (RHT). *Pattern Recognition Letters*, Vol. 11:pp. 331–338, 1990.

[12] W. H. Tsai and S. S. Yu. Attributed string matching with merging for shape recognition. *IEEE Trans. Pattern Analysis Mach. Intell.*, Vol. 7:pp. 453–462, 1985.

Table 2: Comparison of MRAM and Hausdorff Distance

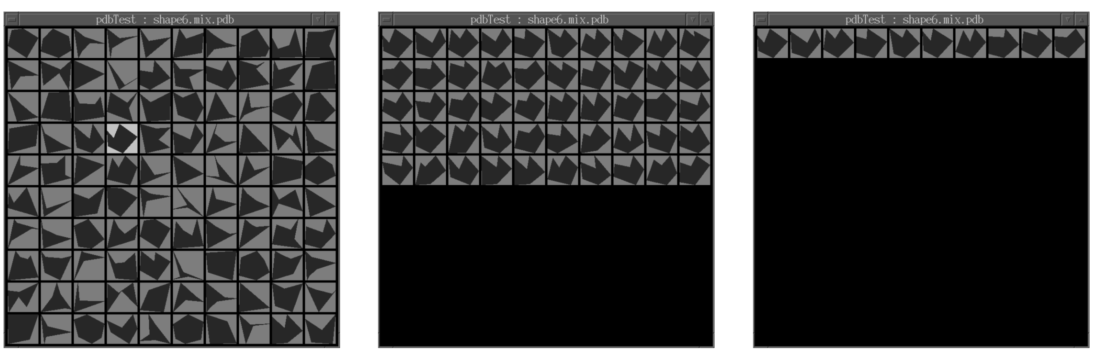
	MRAM	**Hausdorff Distance**
Perceptual Similarity	Good	Good
Implementation	Relatively hard	Relatively easy
Storage requirement	$O(L)$	$O(n)$
Time Complexity	$O(mL)$	$O(mn^2)$
Indexing	Supported	Not supported



(a) (b)

Figure 1: Query-by-shape (similar matching) example

(a) Ranking 100 6-sided polygons using MRAM. The highlighted one is the target polygon. (b) Ranking the same 100 6-sided polygons using the same target polygon as (a), but use Hausdorff Distance method.



(a) (b) (c)

Figure 2: Query-by-shape (similar matching) example

(a) the highlighted polygon is selected from a list of templates as the target for query-by-shape. (b) the 50 remaining candidates after level 1 Multi-resolution Area Matching. (c) the 10 most similar polygons to the target after level 3 Multi-resolution Area Matching.