Two-Stage Polygon Representation for Efficient Shape Retrieval in Image Databases

Lun Hsing Tung Irwin King Ping Fu Fung 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

Shape manipulation is an important and non-trivial task in Image Databases. We propose a method for polygon shape matching by using two levels of polygon representation. The first level uses the Binary String Descriptor to quickly find equivalent classes of polygons. The second stage uses a Multi-resolution Area Matching which operates on the subset of shapes having the same equivalent class by coarse-to-fine area matching strategy. We describe these techniques and demonstrate how this two-stage representation works for a sample shape image database.

1 Introduction

Shape representation, indexing and matching, are important issues in Image Databases. Users of Image Databases need to locate images containing the outline of specific objects from time to time. Since shape (outline) is a good visual measure of objects, Query-By-Shape facility is often provided in modern Image Database Systems. Instead of handling arbitrary shapes, we will concentrate on polygons manipulation because shapes are often quantized and represented as into polygons in computer system for processing. To be more specific, we only handle closed, simple and non-degenerate polygons:

Definition 1 *A polygon is represented by a list of vertices coordinates:* $P = \{V_1, V_2, \ldots, V_n\}$*, where* n *is the number of vertices of the polygon and* $V_i \in R^2$*.*

Definition 2 *A polygon is closed if* $V_1 = V_n$*.*

Definition 3 *A polygon is simple if there is no crossing within the polygon.*

Definition 4 *A polygon is non-degenerate if it has no collinear vertices.*

We combine two techniques in polygon matching task: Binary String Descriptor [1] and Multi-resolution Area Matching technique. Our idea is to perform polygon matching in two stages: perform a coarse but fast polygon classification first and then perform a precise polygon matching within a sub of polygons produced in previous step. In the first stage, polygons are represented by Binary String Descriptor (Section 2) and are partitioned into different equivalent classes with respect to their Binary String Descriptors (Section 2.2). The second stage of polygon matching is carried out within a equivalent class. In this stage, polygons are presented by multi-resolution area information (Section 3.1) and similarity measure between polygons is performed using coarse-to-fine area based matching (Section 3.2). How these two techniques are incorporated into Image Database Systems is presented in Section 4.

2 Binary String Descriptor (BSD)

2.1 Idea

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

Definition 5 *A Binary String Descriptor is a string* $\{0, 1\}^n$*, where* n *is the number of vertices of the polygon the descriptor is associated with.*

Since the measurement of convexity and concavity of a vertex is independent of scale, the BSD is scale invariant.

2.2 Standardizing Binary String Descriptor

A polygon can be represented by more than one BSD depending on the sequence of vertices being recorded. 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 of B becomes the MSB of B_i. Let $M(B_i)$ denotes the magnitude of the BSD B_i when B_i is regarded as a binary integer.

Definition 6 *The standardized BSD of B is B_j such that $M(B_j) = min\{M(B_i) | 1 \leq i \leq n\}$.*

Note that Standardized BSD is orientation invariant as well as scale invariant.

2.3 Number of equivalent classes for n-gons

BSD is a many-to-one mapping. Two polygons are said to be in the same equivalent class if they have the same standardized BSD. For BSD with n bits, there are 2^n possible BSDs. However, some of these possible BSD are invalid and some are exactly the same as those from its equivalent class after standardization. For n-gons, the number of distinct equivalent classes (E) is given in [1] as:

$$E = \frac{1}{n} \sum_{m \in D_n} m X_n(m) - (\lfloor \frac{n}{2} \rfloor + 2)$$

where D_n is the set of divisors of n and $X_n = 2^{\frac{n}{m}} - (X_n(m_1) + \cdots + X_n(m_k))$ where $m_1, \ldots, m_k$ are the multiples of m belonging to $D_n \backslash \{m\}$.

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

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

Table 1 shows the number of equivalent classes for polygons with sides from three to sixteen. The table indicates that the BSD may not be a good method for polygon classification when the polygons being handled are with small number of sides since the numbers of distinct equivalent classes are relatively small in these situations. Furthermore, as the number of sides increased, the number of equivalent classes increases quickly, therefore it is also inefficient. From empirical tests, we find that $8 < n < 15$ is ideal for dataset with several thousand of shape objects.

3 Multi-resolution Area Matching

We will now describe how Multi-resolution Area Matching of polygons is carried out.

3.1 Computing multi-resolution area information

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. Multi-resolution area information is computed as follow:

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

In our implementation, the frame buffer is 64×64 pixels large and we compute area information up to level 3.

3.2 Measuring similarity using multi-resolution area information

We use the L_p distance to measure the similarity of two polygons at a specific level of resolution. Given polygon A and B, which multi-resolution area information have already been computed, the similarity of these two polygons at level k is:

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

where S_k is the similarity measure, A_{ki} and B_{ki} are the portion of covered area of 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, performing similarity measuring from coarse resolution (level 0) to fine resolution (level 3, in our implementation). Two polygons A and B are said to be similar at level k if $S_k \leq threshold_k$ where

$threshold_k$ is a predefined threshold value for level k similarity measure. Two polygons are said to be matched if they are similar at all levels (again, up to level 3 in our implementation).

4 Polygon matching in Image Database

This section describes how the two mentioned techniques can be incorporated In Image Database for polygon matching task. As mentioned, our idea is to use the BSD as the first level filtering and then apply the Multi-resolution Area Matching technique on the polygons remained after filtering. The number of objects in Image Database Systems can be very large, thus, the coarse but very efficient Binary String Descriptor method can be used as a fast first level shape classification technique. Multi-resolution Area Matching gives better similarity measure but is relatively inefficient so it is used as second level matching technique within equivalent classes given by BSD.

4.1 Database population

When an image is added into the 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 automatic, by employing some automatic shape recognition algorithms such as [5] and [3], or outlined by user manually.

In order to facilitate further processing, we have to normalize the polygons and resize them to a comparable size. By finding the bounding box of the polygon and resizing the bounding box to be fit into a unit square box, we enforce the polygon to have a normalized size and area.

Along the perceived edges of the polygon, there may be many jerks and trivial edges (e.g. when automatic shape recognition algorithm is employed). We reduce the number of sides of the polygon by removing such disturbances and make the polygon smooth. Following normalization, the length of the edges are standardized. By setting a threshold value within the unit, we detect the short edges and merge them. Eventually, we have a smoothed, normalized polygon bounded by a unit square and it is ready for further manipulation.

For each polygon in an image defined and preprocessed, we compute its Binary String Descriptor and multi-resolution area information. A tuple <BSD, area information, image> is then added into database.

Figure 1(b) and (c) show an unprocessed polygon and a preprocessed polygon respectively.

4.2 Query by shape

This section describes how Query-By-Shape is carried out.

To initiate a shape query, user may either specify a polygon by sketching it out or by selecting a shape in the database. We then compute the BSD and multi-resolution area information of the target polygon. With the BSD of target polygon, we retrieve all polygons inside the database which have the exact BSD as the target polygon. With all the polygons found in previous step, we compute their similarity to the target polygon at different level of resolution, starting from level 0. At level k, only the N_k most similar polygons are selected as the candidates for level $k+1$. In our implementation, $N_0 = 100$, $N_1 = 50$, $N_2 = 25$ and $N_3 = 10$. After the above steps, a list of best matched polygons (and/or the images where they are associated with) are produced, which can be sorted in the order of similarity to the target polygon. Figure 2 shows an example of Query-By-Shape.

5 Conclusion

We proposed a method for providing Query-By-Shape facility in Image Databases. Our work only concentrates on closed, simple and non-degenerate polygons. Our approach incorporated both Binary String Descriptor and Multi-resolution Area Matching techniques.

References

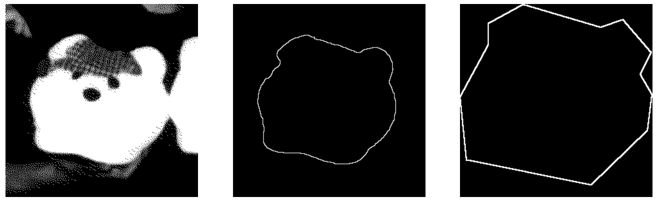
[1] B. Bhavnagri. A Method For Representing Shape Based On An Equivalent Relation On Polygons. *Pattern Recognition*, Vol. 27(No. 2):pp. 247–260, 1994.

[2] 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.

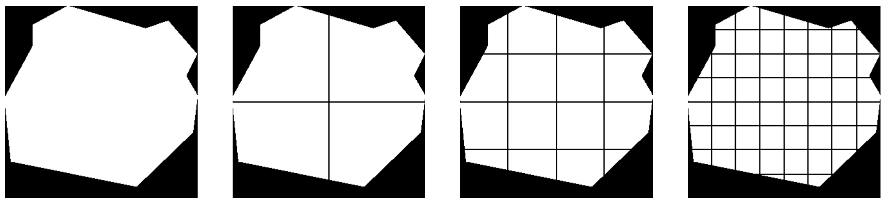
[3] P. F. Fung, W. S. Lee, and I. King. Randomized Generalized Hough Transform for 2-D grayscale object detection [submitted].

[4] 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.

[5] L. H. Tung and I. King. Partitioned Randomized Hough Transform [submitted].



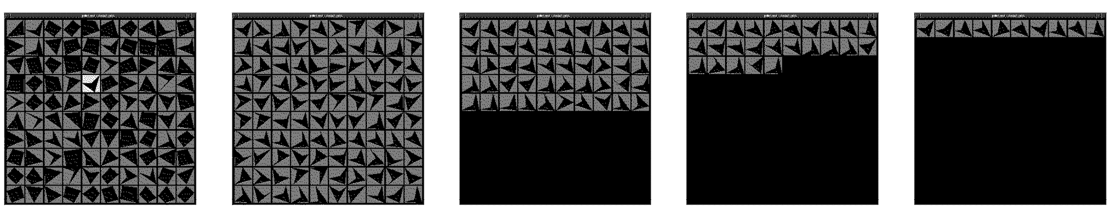
(a) (b) (c)



(d) (e) (f) (g)

Figure 1: Polygon preprocessing

(a) the original image. (b) user defined polygon, either outlined manually or automatic generated by shape segmentation algorithm. (c) polygon after normalization and jerky edges removal. (d) computing level 0 area information. (e) computing level 1 area information. (f) computing level 2 area information. (g) computing level 3 area information.



(a) (b) (c) (d) (e)

Figure 2: Query-By-Shape example

(a) the highlighted polygon is selected from a list of templates as the target for Query-By-Shape. (b) the 100 candidates after level 0 Multi-resolution Area Matching. (c) the 50 remaining candidates after level 1 Multi-resolution Area Matching. (d) the 25 remaining candidates after level 2 Multi-resolution Area Matching. (e) the 10 most similar polygons to the target after level 3 Multi-resolution Area Matching.