

Problem 3: Region Query Problem

Source: Avant!, Taiwan

February 15, 2000

1 Introduction and Problem Definition

Searching for objects that intersect a specified area in two-dimensional space is commonly referred to as region query. For this problem, a set of simple polygons are given. Simple polygons are regions enclosed by a single closed polygonal chain that does not intersect itself. Thus we do not allow polygons with holes. After reading in the polygons, the program expects three types of query commands and gives the correct answers. Those commands are:

1. Find the polygons contained by a given window area.
2. Find the polygons interacting with a given window area. Interaction means overlap or touching among polygons.
3. Find the polygons that do not overlap with others.
4. Find the polygons whose area are less than a given number.
5. Find the percentage of the given window area that are covered by polygons.

To efficiently answer the queries, the program has to put the polygons into a proper data structure. Usually, bins or quad-trees are used. Please refer to the reference materials.

2 Input

Input is a list of polygons. A polygon is described by a list of points. Each point consists of some two-dimensional (X, Y) coordinates. Those points define the vertices of the polygon. The number of vertices and a keyword "poly" are following the point list and conclude the definition of the polygon.

Example:

```
-1 -1 1 4 3 1 3 poly
4 4 4 6 3 6 3 4 4 poly
3 5 4 5 4 7 2 7 2 6 3 6 6 poly
```

There are five query command key words:

1. FIND_INSIDE - find the polygons inside the given window.
2. FIND_INTERACT - find the polygons interacting with the given window area. (Even point touching counts.)
3. FIND_NO_OVERLAP - find the polygons that do not overlap with any other polygons.
4. FIND_AREA_LESS_THAN - find the polygons whose area are less than the given number.
5. FIND_DENSITY_INSIDE - find the percentage of the given window area that are covered by polygons.

The window area is defined by the lower-left and upper-right coordinates:

Example:

```
FIND_INSIDE 2 2 4 6
FIND_INTERACT 2 2 4 6
FIND_NO_OVERLAP
FIND_AREA_LESS_THAN 2.5
FIND_DENSITY_INSIDE 2 5 4 7
```

3 Output

Report the polygons in its input format.

Example:

```
command: FIND_INSIDE 2 2 4 6
answer:  4 4 4 6 3 6 3 4 4 poly
command: FIND_INTERACT 2 2 4 6
answer:  -1 -1 1 4 3 1 3 poly
          4 4 4 6 3 6 3 4 4 poly
          3 5 4 5 4 7 2 7 2 6 3 6 6 poly
command: FIND_NO_OVERLAP
answer:  -1 -1 1 4 3 1 3 poly
command: FIND_AREA_LESS_THAN 2.5
answer:  4 4 4 6 3 6 3 4 4 poly
command: FIND_DENSITY_INSIDE 2 5 4 7
answer:  75%
          4 4 4 6 3 6 3 4 4 poly
          3 5 4 5 4 7 2 7 6 3 6 6 poly
```

4 Language/Platform

- Language: no restriction.
- Platform: SUN workstation (Sparc) is preferred.

5 Evaluation

- Correctness: 50%
- CPU time and memory usage: 50% (Please expect that the problem size could reach one million polygons.)

6 Questions

Please report any questions regarding this problem to cad@cis.nctu.edu.tw with the email subject "CAD Contest: Problem 3." Your question(s) will be answered in two weeks, and the Q&A's will be posted

at the contest web site

References

- [1] Hanan Samet, *The Design and Analysis of Spatial Data Structures*, Addison Wesley, 1989
- [2] M. De Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf, *Computational Geometry, Algorithms and Applications*, Springer, 1991
- [3] Anucha Pitaksanonkul, Suchai Thanawastien, and Chidchanok Lursinsap, "Comparisons of Quad Trees and 4-D Trees: New Results," *IEEE Trans. on Computer-Aided Design*, pp.1157-1164, Nov. 1989.
- [4] Glenn G. Lai, Donal S. Fussell, and D.F. Wong, "Hinted Quad Trees for VLSI Geometry DRC Based on Efficient Searching of Neighbors," *IEEE Trans. on Computer-Aided Design*, pp.317-324, Mar. 1996.