MCS 481 Project Two : convex partitioning, range trees, and point location
due Monday 16 March 2009 at 10AM

The goal of this project is to use CGAL (an open source software available from `http://www.cgal.org`) to explore properties of algorithms we have seen to partition polygons, to work with range trees, and to perform point-location queries. The corresponding materials in the textbook occur in chapters 3, 5, and 6.

0. Using CGAL

Version 3.3.1 of CGAL was the most current version until very recently. For this project the differences between versions 3.3.1 or the most recent 3.4 most likely does not matter. CGAL is well documented and we recommend to start working from the examples that come with the installation of the library.

1. Convex Partitioning

Chapter 11 of the CGAL online manual deals with 2D Polygon Partitioning, (a problem we considered in chapter 3), see the directory `/examples/Partition_2` from the CGAL source code distribution. The directory contains four sample programs of different partition algorithms. The programs show how to call the algorithms and do validity checks on the output.

We have seen a sweep method to produce a y-monotone partition of a polygon. This method is implemented by the CGAL algorithm `y_monotone_partition_2`. The purpose of the first assignment is to verify whether the cost of $O(n \log(n))$ holds in practice for random inputs.

CGAL provides an algorithm for an optimal convex partition of a polygon requiring time $O(n^4)$ and space $O(n^3)$. Compare the performance of `optimal_convex_partition_2` with the other routines, record the number of pieces in the computed partitions along with the running times.

Can you find configurations for which the differences in performance are large?

2. Range Trees

In the directory `/examples/RangeSegmentTrees` of the source code distribution of CGAL, we find examples of 1-, 2-, 3-, and 4-dimensional range trees; see Chapter 46 of the online manual for definitions. In class we introduced range trees to reduce the query time of kd-trees.

For a one dimensional range tree, investigate with random data of size n the $O(n \log(n))$ construction and $O(k + \log(n)$ query time, where k is the size of the output. Make sure the values of n and k are large enough to observe significant differences.

For a two dimensional range tree, examine whether the distribution of the data matters. Compare the performance on random points drawn from a mix of discrete and continuous data, e.g.: from $[a, b] \times \{0, 1, \ldots, r\}$ and $\{0, 1, \ldots, r\} \times [a, b]$, where a and b are real numbers and r is some small natural number. Compare the performance of the queries when x comes from a discrete range (from the set $\{0, 1, \ldots, r\}$) and y comes from the interval $[a, b]$ with the opposite case.

3. Point Location Queries

The introduction of Chapter 20 of the online manual describes the representations of planar subdivisions as arrangements. The program `point_location.cpp` in the directory `/examples/Arrangement_2` in the source code distribution of CGAL illustration point location queries.

Compare the performance of `Arr_naive_point_location<Arrangement_2>` with the algorithm we discussed in class, available as `Arr_landmarks_point_location<Arrangement_2>`. Your experiments should demonstrate that the naive point location method requires linear time in the size of the subdivision, whereas the second query method has a logarithmic cost.

4. the Deadline is Monday 16 March 2009 at 10AM

The report you bring to class at 10AM on Monday 16 March 2009 consists of

1. Source code for the `c++` programs you used in the experiments. Also email the source code of these programs to `jan@math.uic.edu` so I can verify your results.
2. Tables with properly formatting of the essential characteristics for each experiment: the dimension of input and output, and the running time. I do not need the raw data.
3. Your interpretation of the results and conclusions.

This project must be an individual effort. Collaborations are not allowed.
If you have questions or difficulties with the project, feel free to come to my office for help.