

MCS 481 Project 3: Voronoi Diagrams, Arrangements of Lines, Delaunay Triangulations due Wednesday 29 April 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 compute Voronoi Diagrams, Arrangements of Lines, and Delaunay Triangulations. The corresponding materials in the textbook occur in chapters 7, 8, and 9.

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. Voronoi Diagrams

In the directory `/examples/Voronoi_diagram_2` from the CGAL source code distribution is the sample program `point_location.cpp`. Consider the following questions.

1. For n sites, we have shown that it takes $O(n \log(n))$ operations to construct a Voronoi diagram. Verify whether the CGAL implementation corresponds to this complexity, for sites in general position. For reliable timings, use many different and sufficiently large values for n .
2. For sites in general position, the degree of each vertex of a Voronoi diagram equals three. Examine the behavior of the CGAL implementation when all sites occur on the same circle. Is there a difference when the sites are almost on a circle?

2. Arrangements of Lines

In `/examples/Arrangement_2` we find the program `vertical_ray_shooting.cpp` to answer point queries to solve the discrepancy problem we considered in Chapter 8. Examine whether the implementation can solve the discrepancy problem in time $O(n^2)$ for n points.

In reporting times, distinguish between the time it takes to construct an arrangement and the time a point query takes.

3. Delaunay Triangulations

The program `terrain.cpp` in the directory `/examples/Triangulation_2` in the source code distribution of CGAL illustrates the computation of a Delaunay triangulation. The program `voronoi.cpp` builds a Delaunay triangulation from the dual of a Voronoi diagram. Consider the following questions:

1. The complexity of computing a Delaunay triangulation of n points is expected to be $O(n \log(n))$. Take several point configurations in random position and report how long it takes to build a Delaunay triangulation. Make sure to take sufficiently large data sets.
2. Compare the computation of a Delaunay triangulation using the method as in `terrain.cpp` with the method used in the program `voronoi.cpp`. Although their complexity is expected to be the same, could you find point configurations for which one method is preferable to the other?

4. the Deadline is Wednesday 29 April 2009 at 10AM

The report you bring to class at 10AM on Wednesday 29 April 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.