CS 3250: Computational Geometry

Spring 2015: T, Th 1:00 - 2:25 in Searles 126

Schedule

Computational geometry studies algorithms for geometric problems that arise in Computer Science applications. These problems typically involve finite sets of points, line segments or polygons and use discrete and combinatorial geometry (as opposed to continuous geometry).

For example, we might be given the location of post-offices on a map, and want to find the closest pair of post-offices; Or we might want to find a triangulation of a set of surface samples that maximizes the minimum angle of a triangle ---- this type of meshing is often used in solid modeling, where small angles cause numerical instability; or, we might be given two sets of segments, and want to find their intersections. These are some simple examples of geometric problems.

Geometric problems arise in many fields in Computer Science, such as computer graphics, computer vision and image processing, robotics, and GIS.

This class will cover basic computational geometry topics such as convex hulls, line segment intersection, guarding and art gallery, segment intersection, point location, Voronoi diagrams, Delaunay triangulations, as well as geometric data structures for geometrci searching, such as kd-trees, range trees, and interval trees.

Prerequisites: Data Structures (cs2100) and Algorithms (cs2200) (or permission of instructor). In other words:

- knowledge of basic analysis techniques: asymptotic notation, growth, solving recurrences.
- knowledge of basic algorithm design techniques: divide-and-conquer, greedy.
- knowledge of basic algorithms and data structures: searching, sorting, binary search trees, priority queues.

Instructor: Laura Toma
Office: 219 Searles

Office hours: Tue, Wed, Wed 2:30pm-3:30pm. For quick questions, you can drop by any time you see the door open. Or, send me an email to setup a (different) time; I'll do my best to accomodate you, but keep in mind my schedule can get pretty full.

Class mailing list: csci3250 at bowdoin.edu. Use it to ask questions. Since somebody else in the class might have the same question, using a mailing list is better for everyone.

Class webpage: http://www.bowdoin.edu/~ltoma/teaching/cs3250-CompGeom/spring15/"

Approximate syllabus::

- Introduction: finding collinear points, finding the closest pair of points, finding fixed-radious near neighbors.
- Geometric primitives (area of triangle, orientation, segment intersection).
- Convex hulls in 2D (naive, gift wrapping, quickhull, Graham scan, incremental, divide-and-conquer algorithm); CH lower bound.
- Art gallery problem. Fisk sufficiency proof.
- Polygon triangulation (quadratic, based on ear removal algorithm).
- Triangulation of monotone polygons.

- Polygon triangulation in O(n lg n) via trapezoidalization.
- Orthogonal range searching. kd-trees and range trees.
- Orthogonal segment-segment intersection queries (window queries). Interval trees.
- Voronoi diagrams.
- Delaunay triangulations. Tandomized-incremental construction.
- Polygon-polygon intersection for convex and non-convex polygons.
- Finding the extreme point of a convex polygon in O(lg n) time.
- Point-in-polygon intersection with Dobkin-Kirkpartick hierarchy.
- Motion planning in 2D:
    - shortest path in a simple (non-convex) polygon with the funnel algorithm.
    - Shortest paths among polygonal obstacles via visibility graph.
    - Computation of the VG with plane sweep.
- Motion planning in 3D.

Textbooks:

- Computational geometry in C. J. O'Rourke.
- Computational geometry: algorithms and applications. Mark de Berg, Otfried Cheong, Mark van Kreveld, Mark Overmars.

Grading policy: The grade will be based on:

- programming assignments (approximately biweekly)
- class work

Naturally the assignments will not be equal in length/difficulty. The last assignment might consitute a project.