CS 3250: Computational Geometry

Spring 2017: T, Th 1:00 - 2:25 in Searles 126

Piazza link | Schedule | Using svn for this class | Assignments | Code

---

Computational geometry studies algorithms for problems that involve geometric data, such as finite sets of points, line segments and polygons. Some classical examples of geometric problems: Given the location of post-offices on a map, we may 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, given two sets of segments, we may want to find their intersections. This type of geometric problems arise in computer graphics, computer vision and image processing, robotics, and GIS.

Tentative syllabus:

This class will cover some basic computational geometry topics:

- Introduction
  - finding collinear points

    - finding the closest pair of points
    - [finding fixed-radius 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
- Segment intersection
    - Bentley-Ottman sweep
- 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.
- Voronoi diagrams.
- Delaunay triangulations.
- 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.

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.

TAs

- Tucker Gordon
- Jason Nawrocki

Office hours:

Laura: Mondays 2:30-4; Tuesdays 2:30-4, Fridays 11:30-12:30
- Tucker: Mondays 7-9pm
- Jason: Wednesdays 7:30-9:30pm

For quick questions, you can drop by any time you see my door open. This term we will be using Piazza for class discussion. The system is highly catered to getting you help fast and efficiently from classmates, the TA, and myself. Rather than emailing questions to the teaching staff, I encourage you to post your questions on Piazza. If you have any problems or feedback for the developers, email team@piazza.com. Find our class page here.

Class webpage:

- you can access it via a link from my website
- http://www.bowdoin.edu/~ltoma/teaching/cs3250-CompGeom/spring17/
- does not have a Blackboard website.

Textbooks: The following are the classical, most popular textbooks and are suggested, not required. You can find them in Searles 224.

- 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 little project.