

CMSC 754: Computational Geometry

Fall 2002

<http://www.cs.umd.edu/~mount/754/>

Instructor: Dave Mount. Office: AVW 3209. Email: mount@cs.umd.edu. Office phone: (301) 405-2704. Office hours: Mon 3:00-4:00, Wed 3:30-4:30. I am also available immediately after class for questions. Feel free to send me email if you cannot make these times to set up another time.

Teaching Assistant: Justin Wan. Office: AVW 1151. Email: ycwan@cs.umd.edu. Office hours: Mon 2:30-3:30, Thu 5:00-6:00.

Class Time: Tue, Thur 2:00-3:15. Room: CSI 1122.

Course Objectives: This is an introductory course to computational geometry and its applications. We will discuss techniques needed in designing and analyzing efficient algorithms for problems in geometry, including convex hulls, geometric intersections, Voronoi diagrams, Delaunay triangulations, geometric data structures, and motion planning.

Text: (Required) *Computational Geometry: Algorithms and Applications* (2nd Edition), M. de Berg, M. van Kreveld, M. Overmars, O. Schwarzkopf, Springer-Verlag, 2000.

Prerequisites: CMSC 451, or its equivalent.

- Knowledge of basic algorithm design techniques, such as divide-and-conquer, greedy algorithms, dynamic programming,
- Knowledge of basic analysis techniques such as asymptotic notation, solving summations and recurrences, basic probability theory.
- Knowledge of basic data structures (e.g. balanced binary trees and heaps).

Course Work: There will be roughly 5 homework assignments. Normally, late homeworks will *not* be accepted. If you think that you have a legitimate excuse for handing in a homework late, contact me *before* the due date. There will be a midterm and comprehensive final exam. The midterm is tentatively scheduled for Thu, Oct 31. The final exam will be Thu, Dec 19, 10:30am-12:30pm. There will be no programming projects. Approximate weights: Homeworks: 25%, Midterm: 30%, Final: 45%.

Academic Dishonesty: All class work is to be done independently. You are allowed to discuss class material, homework problems, and general solution strategies with your classmates. When it comes to formulating/writing/programming solutions you must work alone. If you make use of other sources in coming up with your answers you *must* site these sources clearly (papers or books in the literature, friends or classmates, information downloaded from the web, whatever).

It is best to try to solve problems on your own, since problem solving is an important component of the course. But I will not deduct points if you make use of outside help, provided that you cite your sources clearly. Representing other people's work as your own, however, is plagiarism and is in violation of university policies. Instances of academic dishonesty will be dealt with harshly.

Topics: The following list of topics is very tentative.

Preliminaries: Basic Euclidean geometry.

Convex Hulls: Graham's scan, QuickHull, Chan's algorithm.

Intersections: Plane-sweep and line segment intersection, representation and intersection of planar subdivisions.

Polygon Triangulation: Art gallery problems, monotone partitions.

Linear Programming: Half-plane intersection, incremental method, applications.

Range Searching: kd-trees, range trees, fractional cascading.

Point Location: Kirkpatrick's method, trapezoidal decompositions, history DAGs.

Voronoi Diagrams: Post office problem, divide and conquer algorithm, randomized incremental algorithm, history graphs, Fortune's algorithm.

Arrangements and Duality: Point/line duality, incremental construction of arrangements, levels, applications.

Delaunay Triangulations: Triangulations of point sets, properties of the Delaunay triangulation, incremental construction.

Motion planning: Shortest paths, visibility graphs, roadmaps.

Approximation Methods: Dudley's theorem and applications, well-separated pair decompositions, spanners.