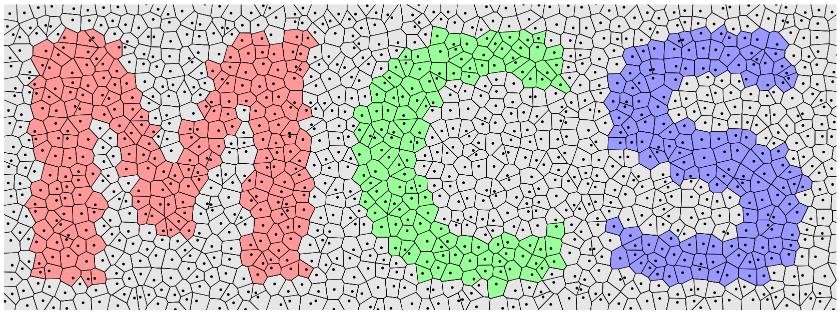
MCS 481 – Computational Geometry – Spring 2012



1. General Information

Instructor	David Dumas (ddumas@math.uic.edu)
Web Page	http://www.math.uic.edu/~ddumas/mcs481/
Textbook	de Berg, Cheong, van Kreveld, and Overmars, *Computational Geometry: Algorithms and Applications, 3rd edition,* Springer, 2008. ISBN-13: 978-3-540-77973-5
Meeting Time	MWF 2:00-2:50pm
Location	Taft Hall 219
CRN	31103 (undergrad) 31104 (grad)
Office Hours	Wed and Fri 11:00am-12:00pm in SEO 503 or by appointment

2. Overview

Computational geometry is the study of data structures that represent geometric objects and algorithms that solve geometric problems. In this course we will discuss some fundamental problems of computational geometry, their mathematical foundations, and algorithmic solutions. We will often compare several approaches to a problem that optimize different measures of efficiency, such as storage space, running time, or algorithmic complexity.

Computational geometry has important applications in fields such as computer graphics, electrical engineering, and geographic information systems. The methods of computational geometry are also useful as a set of tools for computer exploration of geometric problems in pure mathematics.

3. Prerequisites

The prerequisites for this semester's MCS 481 differ slightly from the catalog description:

- MCS 401 is not required.
- Previous programming experience would be helpful but is not required.
- Familiarity with some basic linear algebra will be assumed; Math 310, 320, or equivalent experience would suffice.

4. Topic Outline

(1) Introduction to computational geometry
- What, why, and how?
- Planar convex hull as a prototypical problem
- Goals: Correctness, space and time efficiency

(2) Basic planar problems
- (a) Convex hull
- (b) Line segment intersection
- (c) Map overlay
  - Example: How much of the land area of Illinois is forest?
  - Applications to clipping and polygon intersection
- (d) Triangulating a simple polygon
  - The art gallery problem: How many security cameras are needed? Where should they go?

(3) Geometric approach to linear programming
- (a) Reduction to half-plane intersection
- (b) Incremental solution
- (c) Randomized solution and expected running time analysis
- (d) Application to casting problems: can you remove this object from a mold without breaking it?

(4) Finding things — geometric query problems
- (a) Range searching
  - Organize data geometrically for fast queries
  - Preprocessing / query time trade-off
- (b) Point location
  - Trapezoidal decompositions
  - DAG search structures

(5) Special planar decompositions
- (a) The Voronoi decomposition of a point set
  - Definition and properties
  - Fortune's sweep algorithm
- (b) Generalized Voronoi decompositions (segments, weights)
  - Medial axis and shape recognition applications
- (c) Delaunay triangulations
  - Duality
  - Angle optimality

(6) Additional topics (as time permits)
- (a) Convex hulls in n-space, $n \geq 3$
- (b) Computer graphics
  - (i) The painter's algorithm
  - (ii) Binary space partition
- (c) Motion planning
- (d) Mesh generation

5. Coursework

There will be three types of graded work in this course:

- *Homework exercises*—usually from the textbook, posted on the course web page, and collected approximately every two weeks.
- *Projects*—Three computer projects involving CGAL, a C++ library implementing most of the computational geometry algorithms from the course. Some projects will have options allowing either coding or experimental work depending on your background and preferences.

  The projects will be due on the following dates:

  - Project 1: Monday, February 6
  - Project 2: Friday, March 2
  - Project 3: Friday, March 30
- *Final project*—An in-class presentation (20 minutes) and a written report, due at the end of the semester, on a topic selected in consultation with the instructor.

Please check the course web page regularly to ensure that you have the most up-to-date information about the various assignments.

6. CGAL

For the computer projects you will use the Computational Geometry Algorithms Library (CGAL), an open source C++ implementation most of the geometric algorithms discussed in the course. The library and its documentation are available from the CGAL web page (`http://www.cgal.org`).

You will need regular access to a computer with CGAL and a compatible C++ build environment. It may be easiest for you to install CGAL on your own computer, especially if CGAL has been packaged for easy installation on your operating system (e.g. Ubuntu and Debian GNU/Linux). A detailed installation guide is also available on the CGAL web page.

Please try to install CGAL as early in the semester as possible. If you have trouble installing CGAL, ask your instructor for assistance. If installing CGAL on your home computer is not an option, we will create an account for you on a department computer that has CGAL installed (for remote login only).

7. Grading

Your final grade will be based on your homework assignments, assigned projects, and your final project presentation and report. These items will be weighted as follows:

Homework	30%
Projects	30%
Final project (report & presentation)	40%

8. Attendance

Attending the lectures is mandatory. If you must miss a lecture, you should make arrangements to get notes and any class materials from someone else in the class. You are responsible for the contents of all lectures, including any that you cannot attend.

9. Academic honesty

All UIC students are expected to maintain the standards of academic honesty described in the *Guidelines for Academic Integrity* in the Undergraduate Catalog:

http://www.uic.edu/ucat/catalog/GR.shtml#qa

In particular, this policy prohibits plagiarism and giving or receiving aid on an examination.