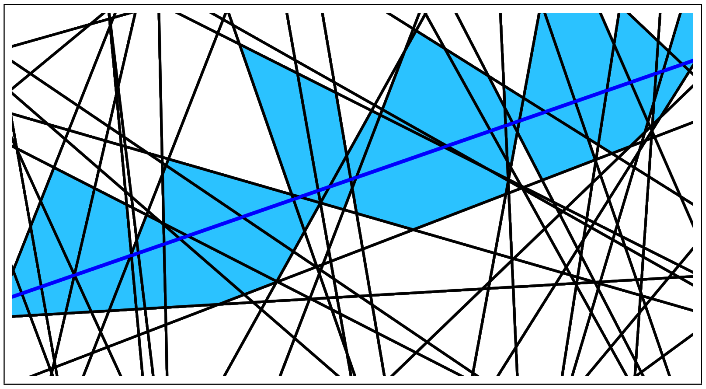
MCS 481 – Computational Geometry – Spring 2014



1. General Information

Instructor	David Dumas (ddumas@math.uic.edu)
Web Page	http://www.math.uic.edu/~ddumas/mcs481/ (or at dumas.io/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 10:00-10:50am
Location	Taft Hall 207
CRN	35483 (undergrad) 35484 (grad)
Office Hours	Mon & Fri, 2-3pm 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

The precise course content is subject to minor changes to account for student background and interests. Current and upcoming topics will be discussed in lecture.

(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) 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

(4) 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

(5) Convex hull in $\mathbb{R}^3$
  (a) Complexity
  (b) Incremental algorithm and analysis
  (c) The quickhull algorithm
  (d) Computing 2D Voronoi with 3D convex hull

(6) Binary space partition
  (a) Background: Application to rendering 3D graphics
  (b) Construction of a BSP tree
  (c) Pathological configurations and size bounds

(7) Motion planning

(a) Background: Application to robotics
(b) Configuration spaces
(c) Translating polygonal robot case
(d) Minkowski sums
(e) Allowing rotation
(f) Finding shortest paths
(g) Computing visibility graphs

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 longer-term projects that involve either implementing an algorithm discussed in the course using a program language of your choice, or studying the implementation in CGAL, a C++ computational geometry library. Some projects may allow a custom component in which students have some freedom to choose their own topic, with instructor approval. Details will be posted on the course web site.

  The projects will be due on the following dates:

  - Project 1: Monday, February 10
  - Project 2: Friday, March 7
  - Project 3: Friday, April 4
- *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

The Computational Geometry Algorithms Library (CGAL) is 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).

Students in MCS 481 will need regular access to a computer with CGAL installed and with a compatible C++ build environment. CGAL packages and installation instructions are available for a variety of operating systems (see the CGAL web page for details). Installation on most recent GNU/Linux distributions is particularly easy.

The point of requiring CGAL access for MCS 481 students is two-fold: The three assigned projects can be completed by studying CGAL implementations of geometric algorithms, and thus access to CGAL is necessary to make this a meaningful option. But even for students who choose to complete projects through coding or other options, it is helpful to have access to a working implementation of the geometric algorithms under consideration.

Please try to install CGAL as early in the semester as possible. **Establishing access to a working CGAL installation and compiling an example program is the first homework assignment, due on Friday, January 24.** Check the course web page for details.

If you have trouble installing CGAL, ask your instructor for assistance.

Though it will probably more convenient to have CGAL on your own computer, we can also create an account for you on a department computer that has CGAL installed. In that case your account will only be accessible through SSH (remote login) and it will not be possible to run programs requiring a graphical user interface. If you would like such an account, please contact your instructor as early in the semester as possible.

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.