**(Untitled, Till Rickert, Shift 2005 Calendar.)**

# CS 274
# Computational Geometry

Jonathan Shewchuk

Autumn 2006
Mondays and Wednesdays, 1:00-2:30 pm
320 Soda Hall

**Combinatorial geometry:** Polygons, polytopes, triangulations, planar and spatial subdivisions. Constructions: triangulations of polygons, convex hulls, intersections of halfspaces, Voronoi diagrams, Delaunay triangulations, arrangements of lines and hyperplanes, Minkowski sums; relationships among them. Geometric duality and polarity. Numerical predicates and constructors. Upper Bound Theorem, Zone Theorem.

**Algorithms and analyses:** Sweep algorithms, incremental construction, divide-and-conquer algorithms, randomized algorithms, backward analysis, geometric robustness. Construction of triangulations, convex hulls, halfspace intersections, Voronoi diagrams, Delaunay triangulations, arrangements, Minkowski sums.

**Geometric data structures:** Doubly-connected edge lists, quad-edges, face lattices, trapezoidal maps, history DAGs, spatial search trees (a.k.a. range search), binary space partitions, visibility graphs.

**Applications:** Line segment intersection and overlay of subdivisions for cartography and solid modeling. Triangulation for graphics, interpolation, and terrain modeling. Nearest neighbor search, small-dimensional linear programming, database queries, point location queries, windowing queries, discrepancy and sampling in ray tracing, robot motion planning.

Here are Homework 1, Homework 2, Homework 3, Homework 4. and Homework 5.

**The best related sites:**

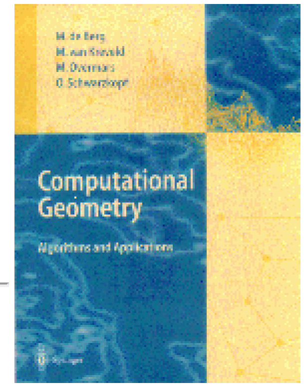
- David Eppstein's **Geometry in Action** and **Geometry Junkyard**.
- Jeff Erickson's **Computational Geometry Pages**.
- Lists of open problems in computational geometry from Erik Demaine et al., Jeff Erickson, and David Eppstein.

**Resources for dealing with robustness problems** (in increasing order of difficulty):

- My robust predicates page (floating-point inputs, C).
- Chee Yap's **CORE Library** (C/C++).

- David Bailey's extensive **MPFUN** arbitrary precision arithmetic package (floating-point, C++ or Fortran).
- Olivier Devillers' predicates (integer inputs).
- Stefan Näher et al.'s **LEDA** contains several arbitrary precision numerical types, including integers and floating-point (C++). Commercial; you have to pay for it.

---




## Textbook

Mark de Berg, Marc van Kreveld, Mark Overmars, and Otfried Schwarzkopf (presently known as Otfried Cheong), Computational Geometry: Algorithms and Applications, second revised edition, Springer-Verlag, 2000. ISBN # 3-540-65620-0.
Known throughout the community as the Dutch Book.

---

## Lectures

Homeworks will be irregularly assigned, and are due at the start of class on a Wednesday. Homeworks are to be done alone, without help from or discussion with other humans.

| | **Topic** | **Readings** | **Due Wednesday** |
|---|---|---|---|
| **1**: August 28 | Two-dimensional convex hulls | Chapter 1, Erickson notes | . |
| **2**: August 30 | Line segment intersection | Sections 2, 2.1 | . |
| **September 4** | **Labor Day** | . | . |
| **3**: September 6 | Overlay of planar subdivisions | Sections 2.2, 2.3, 2.5 | . |
| **4**: September 11 | Polygon triangulation | Sections 3.2-3.4 | . |
| **5**: September 13 | Delaunay triangulations | Sections 9-9.2 | . |
| **6**: September 18 | Delaunay triangulations | Sections 9.3, 9.4, 9.6 | . |
| **7**: September 20 | Voronoi diagrams | Sections 7, 7.1, 7.3 | . |
| **8**: September 25 | Planar point location | Chapter 6 | . |
| **9**: September 27 | Geometric robustness | Lecture notes | Homework 1 |
| **10**: October 2 | Duality; line arrangements | Sections 8.2, 8.3 | . |
| **11**: October 4 | Zone theorem; discrepancy | Sections 8.1, 8.4 | . |
| **12**: October 9 | Polytopes | Matoušek Chapter 5 | . |
| **13**: October 11 | Polytopes and triangulations | Seidel Upper Bound Theorem | Homework 2 |
| **14**: October 16 | Small-dimensional linear programming | Sections 4.3, 4.6; Seidel T.R. | . |
| **15**: October 18 | Small-dimensional linear programming | Section 4.4; Seidel appendix | . |

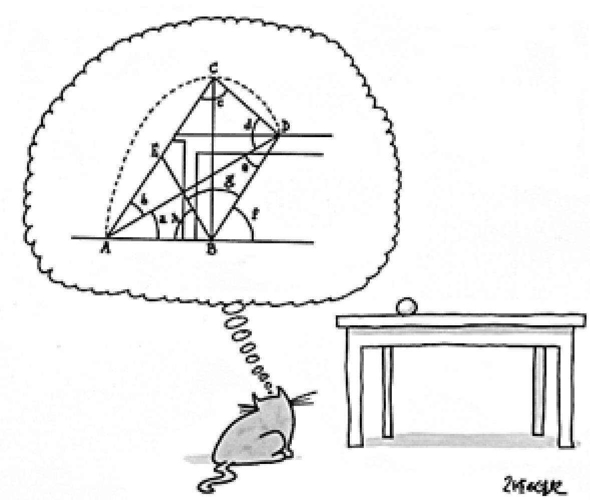
| | | | |
|---|---|---|---|
| **16**: October 23 | Higher-dimensional convex hulls | Seidel T.R.; Secs. 11.2 and 11.3 | . |
| **17**: October 25 | Higher-dimensional Voronoi; point in polygon | Secs. 11.4, 11.5 | . |
| **18**: October 30 | k-d trees | Sections 5-5.2 | . |
| **19**: November 1 | Range trees | Sections 5.3-5.6 | Homework 3 |
| **20**: November 6 | Interval trees; closest pair in point set | Sections 10-10.1; Smid Sec. 2.4.3 | . |
| **21**: November 8 | Segment trees | Section 10.3 | . |
| **22**: November 13 | Binary space partitions | Sections 12-12.3 | . |
| **23**: November 15 | Binary space partitions | Sections 12.4, 2.4, BSP FAQ | Homework 4 |
| **24**: November 20 | Robot motion planning | Sections 13-13.2 | . |
| **25**: November 22 | Minkowski sums | Sections 13.3-13.5 | . |
| **26**: November 27 | Visibility graphs | Chapter 15; Khuller notes | . |
| **27**: November 29 | Constrained triangulations | . | Project |
| **28**: December 4 | Dobkin-Kirkpatrick hierarchies | . | . |
| **29**: December 6 | Homework review | . | Homework 5 |

For August 28, here are Jeff Erickson's lecture notes on two-dimensional convex hulls.

For September 27, here are my Lecture Notes on Geometric Robustness.

For October 9 and 11, if you want to supplement my lectures, most of the material comes from Chapter 5 of Jirí Matoušek, Lectures on Discrete Geometry, Springer (New York), 2002, ISBN # 0387953744. He has several chapters online; unfortunately Chapter 5 isn't one of them.

For October 11, I will hand out Raimund Seidel, The Upper Bound Theorem for Polytopes: An Easy Proof of Its Asymptotic Version, Computational Geometry: Theory and Applications **5**:115-116, 1985. Don't skip the abstract: the main theorem and its proof are both given in their entirety in the abstract, and are not reprised in the body at all.



Seidel's linear programming algorithm (October 16 & 18), the Clarkson-Shor convex hull construction algorithm (October 23), and Chew's linear-time algorithm for Delaunay triangulation of convex polygons are surveyed in Raimund Seidel, Backwards Analysis of Randomized Geometric Algorithms, Technical Report TR-92-014, International Computer Science Institute, University of California at Berkeley, February 1992. Warning: online paper is missing the figures. I'll hand out a version with figures in class.

For October 18, I will hand out the appendix from Raimund Seidel, Small-Dimensional Linear Programming and Convex Hulls Made Easy, Discrete & Computational Geometry **6**(5):423-434, 1991. For anyone who wants to implement the linear programming algorithm, I think this appendix is a better guide than the Dutch Book.

On November 6, I will teach a randomized closest pair algorithm from Section 2.4.3 of Michiel Smid, Closest-Point Problems in Computational Geometry, Chapter 20, Handbook on Computational Geometry, J. R. Sack and J. Urrutia (editors), Elsevier, pp. 877-935, 2000. Note that this is a long paper, and you only need pages 12-13.

For November 15, here is the BSP FAQ.

For November 27, here are Samir Khuller's notes on visibility graphs.

For the Project, read Leonidas J. Guibas and Jorge Stolfi, Primitives for the Manipulation of General Subdivisions and the Computation of Voronoi Diagrams, ACM Transactions on Graphics **4**(2):74-123, April 1985. Feel free to skip Section 3, but read the rest of the paper. See also this list of errors in the Guibas and Stolfi paper, and Paul Heckbert, Very Brief Note on Point Location in Triangulations, December 1994. (The problem Paul points out can't happen in a Delaunay triangulation, but it's a warning in case you're ever tempted to use the Guibas and Stolfi walking-search subroutine in a non-Delaunay triangulation.)

---

## Geometry Applets

These applets can be quite helpful in establishing your geometric intuition for several basic geometric structures and concepts.

- Convex hulls
- Delaunay triangulations
- Voronoi diagrams and Delaunay triangulations I
- Voronoi diagrams and Delaunay triangulations II
- Line sweep
- Fortune's sweep-line Delaunay triangulation algorithm
- Quadtrees of points in the plane

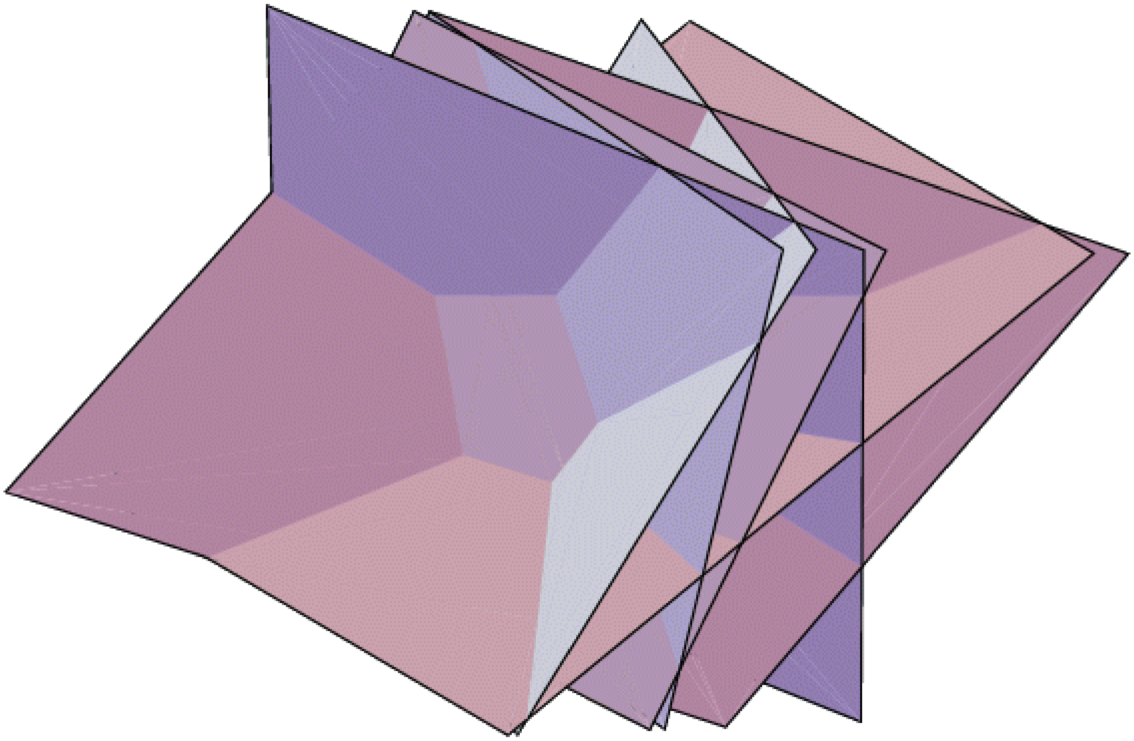
---

## Prerequisites

- CS 170 (Advanced Algorithms) or the equivalent. In particular, you should know and understand amortized analysis; how to solve recurrences; sorting algorithms; graph algorithms like Dijkstra's shortest path algorithm, connected components, and topological sorting; and basic data structures like binary heaps, hash tables, and balanced binary search trees (splay trees or AVL trees or red-black trees or 2-3-4 trees or B-trees). Every one of these will make an appearance at least once.
- A basic course in probability.
- Experience doing mathematical proofs. If you've never taken a class where you did lots of proofs, consider working your way through Bruce Ikenaga's notes and Larry Cusick's notes and exercises.

## Grading

- **80%** for the homeworks.
- **20%** for the project: a Delaunay triangulation implementation, or an alternative by arrangement with the instructor.



Supported in part by the National Science Foundation under Awards ACI-9875170, CMS-9980063, CCR-0204377, CCF-0430065, CCF-0635381, and EIA-9802069, in part by a gift from the Okawa Foundation, and in part by an Alfred P. Sloan Research Fellowship.

(Radiolarian Color Painting. **[Ernst Haeckel]**, **zoologist, 1834-1919.**)

jrs@cs.berkeley.edu