(Lime-basil Triangulation, Dinara Kasko, 2016.)

CS 274 Computational Geometry

Jonathan Shewchuk

Spring 2017
Mondays and Wednesdays, 2:30-4:00 pm
310 Soda Hall

My office hours:
Mondays, 5:10–6 pm, 529 Soda Hall,
Wednesdays, 9:10–10 pm, 411 Soda Hall, and by appointment.
(I'm usually free after the lectures too.)

Combinatorial geometry: Polygons, polytopes, triangulations and simplicial complexes, planar and spatial subdivisions. Constructions: triangulations of polygons and point sets, convex hulls, intersections of halfspaces, Voronoi diagrams, Delaunay triangulations, restricted Delaunay triangulations, arrangements of lines and hyperplanes, Minkowski sums, Reeb graphs and contour trees; relationships among them. Geometric duality and polarity. Upper Bound Theorem, Zone Theorem.

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

Geometric data structures: Doubly-connected edge lists, quad-edges, face lattices, trapezoidal maps, conflict graphs, history DAGs, spatial search trees (a.k.a. range search), segment trees, binary space partitions, quadtrees and octrees, 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, curve reconstruction and surface reconstruction, robot motion planning.

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

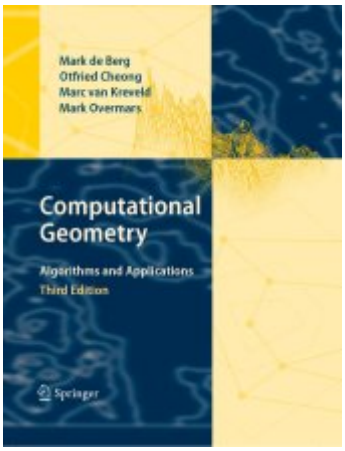
The best related sites:

- 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, Otfried Cheong, Marc van Kreveld, and Mark Overmars, Computational Geometry: Algorithms and Applications, third edition, Springer-Verlag, 2008. ISBN # 978-3-540-77973-5. Known throughout the community as the Dutch Book. Highly recommended; it's one of the best-written textbooks I've ever read. Note that one lecture will cover material available only in the third edition (BSP trees for low-density scenes; Section 12.5); earlier editions of the Dutch Book will probably suffice for everything else.

Lectures

The following schedule is tentative; changes are possible. Chapter headings refer to the third edition. Homeworks will be irregularly assigned, and are due at the start of class. Homeworks are mostly to be done alone, without help from or discussion with other humans; but each homework has one or two group problems, which you may do with one or two other students. (See Homework 1 for detailed rules.)

	Topic	**Readings**	**Assignment Due**
1: January 18	Two-dimensional convex hulls	Chapter 1, Erickson notes	.
2: January 23	Line segment intersection	Sections 2, 2.1	.
3: January 25	Overlay of planar subdivisions	Sections 2.2, 2.3, 2.5	.
4: January 30	Polygon triangulation	Sections 3.2–3.4	.
5: February 1	Delaunay triangulations	Sections 9–9.2; my Chapter 2	.
6: February 6	Delaunay triangulations	Sections 9.3, 9.4, 9.6	.
7: February 8	Voronoi diagrams	Sections 7, 7.1, 7.5	.
8: February 13	Planar point location	Chapter 6	.
9: February 15	Duality; line arrangements	Sections 8.2, 8.3	Homework 1
February 20	**Presidents' Day**	.	.
10: February 22	Zone Theorem; discrepancy	Sections 8.1, 8.4	.
11: February 27	Polytopes	Matoušek Chapter 5	.
12: March 1	Polytopes and triangulations	Seidel Upper Bound Theorem	Homework 2

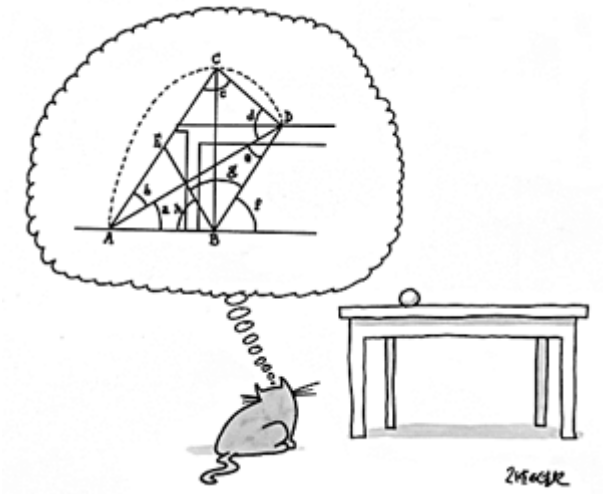
13: March 6	Small-dimensional linear programming	Seidel T.R.; Sections 4.3, 4.6	.
14: March 8	Small-dimensional linear programming	Section 4.4; Seidel appendix	.
15: March 13	Higher-dimensional convex hulls	Seidel T.R.; Secs. 11.2 and 11.3	.
16: March 15	Higher-dimensional Voronoi; point in polygon	Secs. 11.4, 11.5	.
17: March 20	Range trees	Sections 5.3–5.6	.
18: March 22	Segment trees	Section 10.3	Homework 3
March 27–31	**Spring Recess**		
19: April 3	Geometric robustness	Lecture notes	.
20: April 5	Binary space partitions	Sections 12–12.3	.
21: April 10	Binary space partitions	Section 12.5; BSP FAQ	.
22: April 12	BSP applications; nearest neighbors	Section 2.4; Arya et al.	Homework 4
23: April 17	Motion planning; Minkowski sums	Sections 13–13.4	.
24: April 19	Visibility graphs	Chapter 15; Khuller notes	.
25: April 24	Curve reconstruction	Dey & Kumar	Project
26: April 26	Surface reconstruction	Amenta et al.	.
27: May 1	Nonconvex polyhedra	.	Homework 5

For January 18, here are Jeff Erickson's lecture notes on two-dimensional convex hulls.

For February 1, you might (optionally) also be interested in Chapter 2 from my book: Siu-Wing Cheng, Tamal Krishna Dey, and Jonathan Richard Shewchuk, Delaunay Mesh Generation, CRC Press (Boca Raton, Florida), December 2012.

For February 27 and March 1, if you want to supplement my lectures, most of the material comes from Chapter 5 of Jiří 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 March 1, 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 (March 6 & 8), the Clarkson–Shor convex hull construction algorithm (March 13), 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 March 8, 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.

For April 3, here are my Lecture Notes on Geometric Robustness.

For April 10, here is the BSP Frequently Asked Questions.

For April 12, here is the Sunil Arya, David M. Mount, Nathan S. Netanyahu, Ruth Silverman, and Angela Y. Wu, An Optimal Algorithm for Approximate Nearest Neighbor Searching in Fixed Dimensions, Journal of the ACM **45**(6):891–923, November 1998.

For April 19, here are Samir Khuller's notes on visibility graphs.

On April 24, we study the NN-Crust algorithm by Tamal K. Dey and Piyush Kumar, A Simple Provable Algorithm for Curve Reconstruction, Proceedings of the Tenth Annual Symposium on Discrete Algorithms (Baltimore, Maryland), pages 893–894, January 1999. My lecture includes Lemma 1 from this pioneering paper, which Dey and Kumar use in their correctness proof: Nina Amenta, Marshall Bern, and David Eppstein, The Crust and the Beta-Skeleton: Combinatorial Curve Reconstruction, Graphical Models and Image Processing **60/2**(2):125–135, 1998.

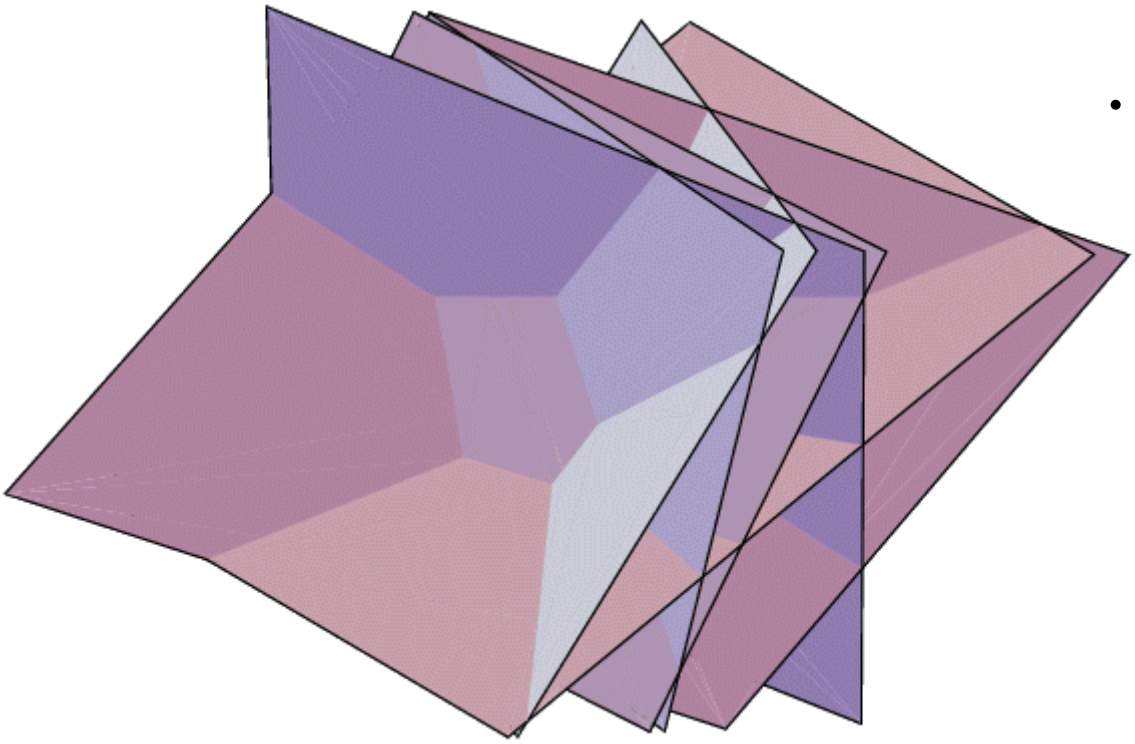
For April 26, I suggest reading the first paper below on the Cocone algorithm. Feel free to skip the proofs, but read the theorems. The second paper, on what has come to be known as the Wrap algorithm, is for reference. Nina Amenta, Sunghee Choi, Tamal K. Dey, and N. Leekha, A Simple Algorithm for Homeomorphic Surface Reconstruction, International Journal of Computational Geometry and Applications **12**(1–2):125–141, 2002. Herbert Edelsbrunner, Surface Reconstruction by Wrapping Finite Sets in Space, pages 379–404 of Discrete and Computational Geometry: The Goodman–Pollack Festschrift, Boris Aronov (editor), Springer-Verlag, 2003. Be warned that some heavy translation is required to get from the Edelsbrunner paper to how I describe it in lecture.

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, IIS-0915462, CCF-1423560, and EIA-9802069, in part by a gift from the Okawa Foundation, and in part by an Alfred P. Sloan Research Fellowship.

(Cake "Chocolate Block", Dinara Kasko, 2016.)

jrs@cs.berkeley.edu