



(Untitled, Till Rickert, [Shift 2005 Calendar](#).)

# CS 274

## Computational Geometry

[Jonathan Shewchuk](#) and [Marc Houry](#).

Spring 2015  
Mondays and Wednesdays, 5:30-7:00 pm  
320 Soda Hall

Jonathan's office hours:  
Mondays, 2:10-3 pm, 510 Soda Hall, and by appointment  
(I'm usually free after the lectures too.)

Marc's office hours:  
Tuesdays, 2:10-3 pm, 510 Soda Hall

**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, arrangements of lines and hyperplanes, Minkowski sums, Reeb graphs and contour trees; 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, 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), 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, 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
<b>1:</b> January 21	Two-dimensional convex hulls	Chapter 1, <a href="#">Erickson notes</a>	.
<b>2:</b> January 26	Line segment intersection	Sections 2, 2.1	.
<b>3:</b> January 28	Overlay of planar subdivisions	Sections 2.2, 2.3, 2.5	.
<b>4:</b> February 2	Polygon triangulation	Sections 3.2–3.4	.
<b>5:</b> February 4	Delaunay triangulations	Sections 9–9.2; <a href="#">my Chapter 2</a>	.
<b>6:</b> February 9	Delaunay triangulations	Sections 9.3, 9.4, 9.6	.
<b>7:</b> February 11	Voronoi diagrams	Sections 7, 7.1, 7.5	.
<b>February 16</b>	<b>Presidents' Day</b>	.	.
<b>8:</b> February 18	Planar point location	Chapter 6	<a href="#">Homework 1</a>
<b>9:</b> February 23	Duality; line arrangements	Sections 8.2, 8.3	.
<b>10:</b> February	Zone Theorem; discrepancy	Sections 8.1, 8.4	.

25			
<b>11: March 2</b>	Polytopes	Matoušek Chapter 5	.
<b>12: March 4</b>	Polytopes and triangulations	Seidel <a href="#">Upper Bound Theorem</a>	<a href="#">Homework 2</a>
<b>13: March 9</b>	Small-dimensional linear programming	<a href="#">Seidel T.R.</a> ; Sections 4.3, 4.6	.
<b>14: March 11</b>	Small-dimensional linear programming	Section 4.4; <a href="#">Seidel appendix</a>	.
<b>15: March 16</b>	Higher-dimensional convex hulls	Seidel T.R.; Secs. 11.2 and 11.3	.
<b>16: March 18</b>	Higher-dimensional Voronoi; point in polygon	Secs. 11.4, 11.5	.
<b>March 23–27</b>	<b>Spring Recess</b>		
<b>17: March 30</b>	k-d trees	Sections 5–5.2	.
<b>18: April 1</b>	Range trees	Sections 5.3–5.6	<a href="#">Homework 3</a>
<b>19: April 6</b>	Interval trees; closest pair in point set	Sections 10–10.1; <a href="#">Smid</a> Sec. 2.4.3	.
<b>20: April 8</b>	Segment trees	Section 10.3	.
<b>21: April 13</b>	Geometric robustness	<a href="#">Lecture notes</a>	.
<b>22: April 15</b>	Binary space partitions	Sections 12–12.3	<a href="#">Homework 4</a>
<b>23: April 20</b>	Binary space partitions	Section 12.5	.
<b>24: April 22</b>	BSP applications; nearest neighbors	Section 2.4; <a href="#">BSP FAQ</a> ; <a href="#">Arya et al.</a>	.
<b>25: April 27</b>	Motion planning; Minkowski sums	Sections 13–13.4	<a href="#">Project</a>
<b>26: April 29</b>	Visibility graphs	Chapter 15; <a href="#">Khuller notes</a>	.
<b>27: May 4</b>	Reeb graphs (and contour trees)	<a href="#">Harvey et al.</a>	<a href="#">Homework 5</a>

For January 21, here are [Jeff Erickson's lecture notes on two-dimensional convex hulls](#).

For February 4, 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.

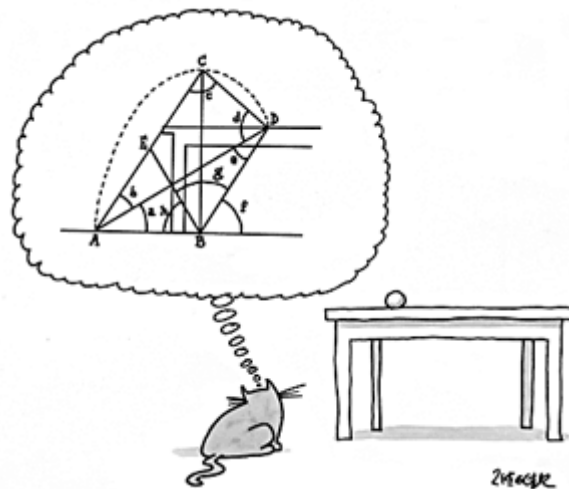
For March 2 and 4, if you want to supplement my lectures, most of the material comes from Chapter 5 of [Jiri Matoušek, Lectures on Discrete Geometry](#), Springer (New York), 2002, ISBN # 0387953744. He has several chapters online; unfortunately Chapter 5 isn't one of them.

For March 4, 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 9 & 11), the Clarkson–Shor convex hull construction algorithm (March 16), 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 11, 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 April 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 April 13, here are my [Lecture Notes on Geometric Robustness](#).

For April 22, here is the [BSP FAQ](#), and here is [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 29, here are [Samir Khuller's notes](#) on visibility graphs.

For May 4, I will hand out [William Harvey, Yusu Wang, and Rephael Wenger, A Randomized  \$O\(m \log m\)\$  Time Algorithm for Computing Reeb Graphs of Arbitrary Simplicial Complexes](#), *Proceedings of the Twenty-Sixth Annual Symposium on Computational Geometry*, pages 267–276.

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.)

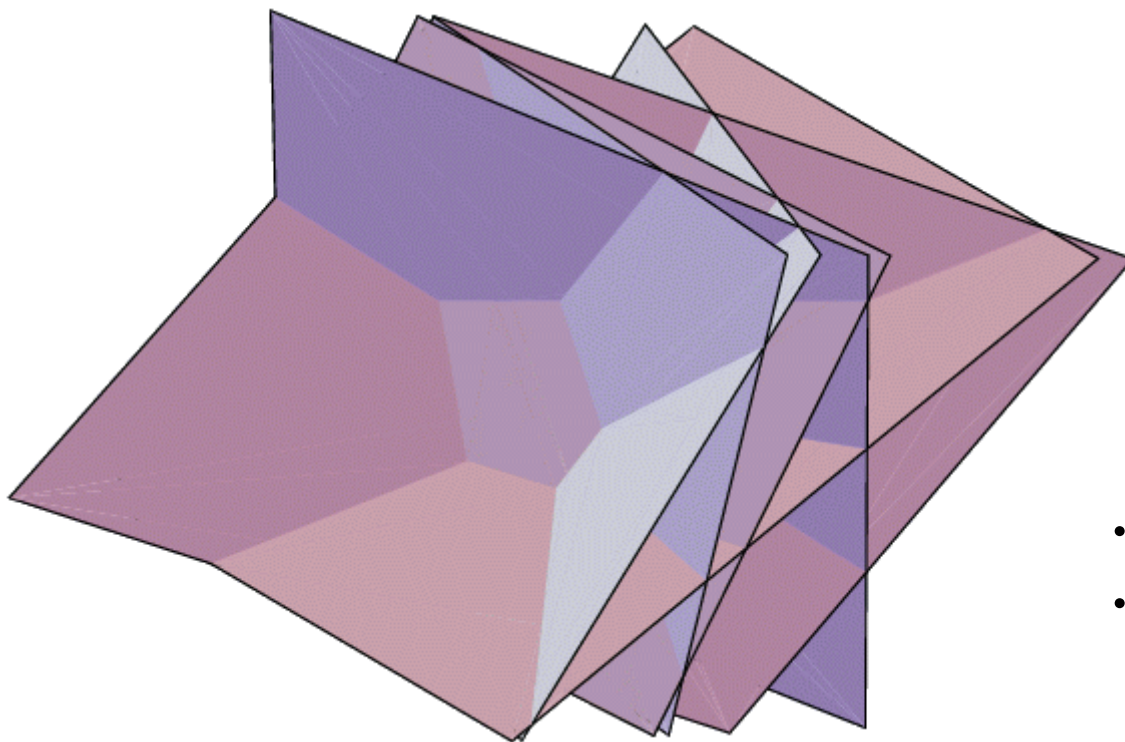
## Geometry Applets

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.

---

(Radiolarian Color Painting. [Ernst Haeckel](#), zoologist, 1834–1919.)

[jrs@cs.berkeley.edu](mailto:jrs@cs.berkeley.edu)

