WELCOME TO COS 451 — COMPUTATIONAL GEOMETRY

Tuesday/Thursday 1:30–2:50, Room 301

http://www.cs.princeton.edu/courses/archive/spring99/cs451/

When and where: Tuesday/Thursday 1:30–2:50, Room 301

Your instructor: Hervé Brönnimann, `hbr@cs.princeton.edu`, 205, x1978

Computational geometry

This course is intended for people who like geometry, mathematical reasoning, computers, and puzzles. The word *geometry* here bears little resemblance to the one you were taught in grade school (e.g. platonic solids, conic sections, trigonometry, etc.). The question here is not to prove existence and properties of these objects, but rather

how to represent geometric objects and manipulate them.

In the same way as mathematicians are happy knowing that water can be made to boil, but computer scientists find only rest once they devise the program to make tea, we will seek to find *algorithms* to solve geometric problems. An essential ingredient of our cuisine is to argue about the efficiency of the constructions. Sometimes the quality of the solutions need to be argued as well, such as when cutting a polyhedron into tetrahedra; then the number of tetrahedra matters. Geometric properties then become of interest for the analysis, but are not an end in itself.

The basic concepts will be applied to such problems as moving a piano out of an appartment, finding the closest phone on campus, reconstructing a toy from its cross sections, or picking up leaves with a minimum number of picks.

Prerequisites

The ideal prerequisites include the basic COS 226 and COS 423 (Theory of Algorithms). An elementary knowledge of probabilities will help in discussing randomized algorithm. Drawing and vizualizing skills a plus. Knowledge of at least one programming language (preferrably C or C++) is necessary for the final project.

Required work

There will be *two problem sets* during the semester (the first due approximately Feb 28 and the second due around April 15), plus minor problems given at the end of each class (but not graded). Finally, at the end of the class, there will be a *personal project,* focussing either on an algorithm implementation (from the textbook) or on the use of a geometric software library for devising a mini-application module (e.g., reconstruction, motion planning, geographic information system).

Text books

Three books have been placed on reserve in the Engineering Library. No books are required, but the first one is strongly recommended. The second is perhaps less intuitively presented and not as practically motivated, while the third is more for the graduate student wishing to deepen his understanding of the topic. The U-Store may not have them until later in the semester, due to reprinting delays, but the first one will be there at the end of the first week of class.

[1] *Computational Geometry: Algorithms and Applications*, Mark de Berg, Mark van Kreveld, Mark Overmars, Otfried Schwartzkopf, Springer, New York, 1997.

[2] *Computational Geometry: an Introduction*, Franco Preparata and Michael I. Shamos, Springer, New York, 1985.

[3] *Algorithmic Geometry*, Jean-Daniel Boissonnat and Mariette Yvinec, Cambridge University Press, 1998.

Topics

I lay out these possible topics in the form of a tentative schedule. However, the schedule will be adapted to the rythm of the class, and possible special interest from the students.

- overview, basic problems, history of the field, models of computation
- basic techniques, geometric computing
- convex hull problems
- triangulation and decomposition problems
- geometric searching
- planar maps, representation and point location
- sweep line algorithm, map overlay
- arrangements
- Minkowski sums, motion planning
- Voronoi diagrams
- application to reconstruction
- geometric optimization
- higher dimensional geometry