CS311 Computational Geometry

Dianna Xu

A study of algorithms and mathematical theories that focus on solving geometric problems in computing, which arise naturally from a variety of disciplines such as Computer Graphics, Computer Aided Geometric Design, Computer Vision, Robotics and Visualization. The materials covered sit at the intersection of pure mathematics and application-driven Computer Science and efforts will be made to accommodate Math majors and Computer Science majors of varying math/computational backgrounds. Topics include: graph theory, triangulation, convex hulls, geometric structures such as Voronoi diagrams and Delaunay triangulations, as well as curves and polyhedra surface topology.

Prerequisites

CS/MATH 231

Textbooks

Discrete and Computational Geometry by Devadoss Satyan and Joseph O'Rourke, 2011. Princeton University Press. ISBN 978-0691145532.

Schedule of Topics

- Week 1: Course overview, intro to graph theory.
- Week 2: Visibility, art gallery theorems and polygon triangulation.
- Week 3: Cross product, polygon partitioning.
- Week 4: Dissections
- Week 5: Convex hull
- Week 6: Triangulations of a point set
- Week 7: Delaunay and other special triangulations
- Week 8: Spring break
- Week 9: Voronoi diagrams
- Week 10: Arrangements and duality
- week 11: Medial axis, Minkowski sums, curve shortening and curve reconstruction
- Week 12: Polyhedra surfaces, Gauss-Bonnet Theorem, Cauchy Rigidity

- Week 13: Shortest Paths and Geodesics
- Week 14: Motion planning and configuration spaces
- Week 15: Search and intersection, space partitioning

Grading For all graded work that receive numerical scores, guidelines of letter grades corresponding to lab/exam score levels will be given during the semester. At the end of the semester, a total score (to which the corresponding final grade is assigned) will be calculated from a weighted average of all scores according to the following weights:

Class participation	5%
Assignments	30%
Midterm	35%
Final project	30%
Total	100%

There will be weekly to bi-weekly assignments for this course, involving both theory (mathematical proofs or algorithm design) and implementations (any programming language of choice). The topics are sufficiently new that there are many open and unsolved problems. We will explore serveral, both in class and in assignments/projects. Students are encouraged to collaborate on all assignments, except for the final project. The definition of collaboration differs depending on whether the assignment is theory or implementation. On theory, collaboration means discussion but individual write-ups of solutions. If collaborating on programming exercises, only one copy of code needs to be handed in. The final project includes a brief, preliminary in-class presentation. More details will be given as the semester progresses.