Spring 17: CSci 8442—Computational Geometry and Applications

Instructor
Ravi Janardan
Dept. of Computer Science & Engineering
University of Minnesota–Twin Cities
Minneapolis, MN 55455

Office: 6–217 Keller Hall
Phone: (612)–625–7338
Email: janardan@umn.edu
URL: http://www.cs.umn.edu/∼janardan

Office hours: Mon. and Tue. 4:30–5:30 p.m.

Class Web page: http://www-users.cselabs.umn.edu/classes/Spring-2017/csci8442

Welcome to CSci 8442!

Computational Geometry is concerned with the design and analysis of efficient techniques for representing and manipulating geometric entities such as points, line-segments, polygons, and polyhedra. These techniques find applications in diverse areas such as, for instance, computer graphics, computer-aided design and manufacture, surgical planning and simulation, query-retrieval in spatial databases, wireless and sensor networks, robotics, and operations research.

This course is an introduction to several fundamental geometric problems and paradigms, with a focus on efficient computation and applications. Representative topics include geometric data structures, geometric searching, geometric intersection problems, triangulations, geometric transformations/duality, proximity problems, and methods for constructing useful geometric structures such as convex hulls, arrangements, Voronoi Diagrams, Delaunay Triangulations, and binary space partitions. (Note: The course will focus on "discrete geometry", as opposed to "continuous geometry" which deals with smooth curves and surfaces.)

This course will be useful to graduate students who have an interest in geometric problems and will provide a strong foundation for doing further research in the area. To get the most out of the course, you must have a good background in the design and analysis of algorithms and data structures (e.g., CSci 5421 or equivalent). It is *strongly recommended* that you have met this pre-requisite so that you are not at a disadvantage. If you are unsure about your preparation for the course, please talk to me right away. In addition, you should read ahead for class, attend the lectures regularly and participate actively in them, work out as many problems as possible from the text, and avail of the assistance offered by the Instructor.

The rest of this document describes the course in more detail. In particular, please pay attention to the policies governing coursework. They are designed to ensure that the course proceeds smoothly and will be implemented from the very beginning.

Once again, welcome to CSci 8442 and best wishes for a successful semester. I hope that you enjoy the course as much as I enjoy teaching it.

Detailed information

When/Where/etc.: MW 1:00–2:15 p.m. in 117 Bruininks Hall. Credits: 3. Prerequisite: As outlined earlier.

Teaching assistant(s): None. All work will be graded by the instructor.

Text: "Computational Geometry: Algorithms and Applications" by M. de Berg et. al., Springer, 2008 (3rd edition). This will be supplemented with papers, as needed. A free electronic copy of the book is available at `http://z.umn.edu/cgbook`. (This is a University-approved free copy.)

Coursework: Four written homework assignments and a project (including a presentation). More information on the project appears at the end of this document.

Evaluation: Grades will be based on a weighted average of the assignments (48%, weighted equally), the Project (47%), and class participation (5%). The latter involves attending classes regularly, contributing constructively to class discussions, and, in general, helping create and maintain a vibrant learning environment. (These are not "free" points.)

Grades will be assigned on the following scale, subject to the requirement set forth in the note on the bottom of page 4:

$A \geq 90\%$, $A- \geq 85\%$, $B+ \geq 80\%$, $B \geq 75\%$, $B- \geq 70\%$, $C+ \geq 65\%$, $C \geq 60\%$, $C- = S \geq 55\%$, $D+ \geq 50\%$, $D \geq 45\%$, and $F < 45\%$.

Please bear in mind that the weighted average is not rounded when computing the letter grade.

An 'Incomplete' will be considered only if a student is doing well but is unable to complete the course due to a documented medical emergency. In particular, an incomplete will not be given for grade improvement purposes.

Important class policies: To qualify for full credit, coursework must be submitted in class on the due date, at the start of class. Work submitted after class begins (i.e., 1:00 p.m.) is considered late; such work will be accepted only until 3:00 p.m. that day and will lose 10% of the total points that the assignment is worth. No work will be accepted after that. *To facilitate grading, all work must be typed (preferably in LaTeX).*

All work must be done independently. You may discuss an assignment problem in general terms with your classmates, but the final answer must be your own. Copying or interfering with the work of another student, plagiarizing from another source (including the Internet), or any other misrepresentation of your work constitutes cheating. This will result in a failing grade and a referral of the case to the CS&E department and to the Office of Student Conduct and Academic Integrity (`http://oscai.umn.edu`). You are urged to read the CS&E Department's official policy on academic conduct (`http://z.umn.edu/acadconduct`).

Web page: Throughout the semester, information about the course, including schedule updates, homework assignments, papers, and late-breaking class news, will be posted on the class web page. The page also links to an online forum, where students can discuss material related to the course, and to sites that contain additional information about computational geometry. Please check the class web page regularly.

Schedule

The following topics will be covered in (roughly) the indicated order; content and schedule are subject to change. Not all material in a chapter (from the text) will be covered and sometimes material will be drawn from papers that will be made available on the class web page. You should augment these readings with notes you take in class since discussions will often range beyond what is found in the readings.

Convex hulls: Definition, applications, complexity, algorithms in 2D and 3D. [Ch. 1, 11, papers]

Line-segment intersection: Algorithms, applications to map overlay. [Ch. 2]

Polygon triangulation: Definition, properties, algorithms, application to guarding. [Ch. 3]

Data structures for query-retrieval: Range trees, kd-trees, interval trees, priority search trees, segment trees, persistent trees. [Ch. 5, 10]

Beyond standard range search: Generalized range search, range-aggregation search. [Paper]

Planar point location: Definition, algorithms, applications. [Papers]

Voronoi Diagrams: Definition, applications, properties, algorithms for nearest and farthest point diagrams, variants. [Ch. 7, paper]

Delaunay Triangulations: Definition, applications, properties, algorithms, relationship between Voronoi Diagrams/Delaunay Triangulations and convex hulls. Other structures related to Delaunay Triangulations. [Ch. 9, 11, papers]

Arrangements and geometric duality: Definition, properties, complexity, construction of arrangements, zone theorem, applications to computing set discrepancy, convex hulls and half-space intersections. [Ch. 8, 11]

Binary space partitions: Definition, data structure, algorithms, application to hidden surface removal. [Ch. 12]

Parameteric search: Method, applications. [Paper]

Project and Homework dates:

The first class meeting is on Wed., 1/18 and the last one is on Wed., 5/3.

Spring Break is the week of Mon., 3/13.

PROJECT DATES	HOMEWORK DATES
2/20: Project proposal due	Hw1: Out 1/23, due 2/13
4/24: Final project due	Hw2: Out 2/13, due 3/6
4/26-5/3: Project presentations (approx.)	Hw3: Out 3/6, due 3/27
	Hw4: Out 3/27, due 4/17

Project Information

The goal of the project is to help you master some aspect of geometric computing. Towards this end, the project could be one of the following:

- An in-depth study of a topic with a significant geometric component. This will involve reading and assimilating a sizeable number of research papers on the topic and writing a survey paper. The survey paper should *not* be a recitation of the research papers; rather it should summarize the work in the area and present your unique viewpoints in a way that reveals deep understanding of the material. Done properly, this paper could be the foundation for a thesis topic and could become publishable.

- Software development, where you design, implement, and test software for some geometric problem(s), write a report on your work, and demo your work to the class. It could be, for instance, an experimental comparison of several known algorithms for a fundamental geometric problem (e.g., triangulation, point-location, etc.) or data structure (e.g., range trees, kd-trees, etc.), or a project that takes a substantial geometric problem embedded in some application (e.g., robotics, graphics, CAD etc.) from concept to design to implementation and testing, or a suite of applets that illustrate the working of several related geometric algorithms and/or data structures (e.g., Voronoi diagrams, Delaunay Triangulations, Convex Hulls etc.)

  Use of off-the-shelf software (e.g., CGAL; `http://www.cgal.org`) in your work is permissible to a limited extent. The project must demonstrate significant new software development effort on your part. The choice of programming language and environment is left to you.

Some of the links on the class web page may be a good starting point for project ideas. Please see me if you would like to discuss ideas prior to writing the project proposal.

Note that you cannot use for the class project any current or previous work done at Minnesota or elsewhere.

The project has three components.

- A project proposal which should be submitted by Feb. 20, for feedback. This should describe the planned work (including background and importance), supply references, and provide a tentative schedule for completion. (For software projects, describe any off-the-shelf software that you think you may use.) The write-up should be 2–3 pages long (single-spaced, $\geq$ 11-point font, excluding references), and is worth 7%.

- The final project report, and software if applicable, is due Apr. 24. This should be 12–15 pages long for survey papers and 6–8 pages long for software projects, including documentation (again, single-spaced, $\geq$ 11-point font, excluding references). This part is worth 30%.

- An in-class presentation, where you present your work to the class and engage in an interactive discussion. This will be scheduled during the class meetings of April 26–May 3. (approx.). This part is worth 10%.

Note: Completion of all three components and earning a score of at least 50% on the project is a necessary condition for passing the course.