CS 4960: Introduction to Computational Geometry

Administrative Details and Course Syllabus

Spring 2016

Instructor: Dr. Bei Wang

Course Information

Instructor: Dr. Bei Wang
Instructor Homepage: `http://www.sci.utah.edu/~beiwang`
Meeting Time: Tuesdays, Thursdays, 3:40 pm - 5:00pm
Classroom: WEB 1230
Textbook: Discrete and Computational Geometry by S. L. Devadoss and J. O'Rourke, 2011.
Web page: `http://www.sci.utah.edu/~beiwang/teaching/cs4960.html`
Office Hours: See time and location from the course webpage or by appointment.
Contact Information:
Office: WEB 4819
Email: beiwang@sci.utah.edu

TA: See office hour information and location on the course webpage.

Course Description

This course is an undergraduate elective that provides an accessible introduction to computational geometry. Computational geometry is a relatively young and emerging area in computer science that has made exciting advancements in recent years. It has close connection to many application areas such as integrated circuit design, computer-aided engineering, computer vision, molecular biology, geometric databases, sensor networks, visualization, robotics, computer graphics and geometric modeling. This course focuses on solving application-driven, data-centric problems with geometric input and output. This course includes 4-6 programming assignments designed to provide hands-on experience in solving problems in 2D and 3D geometry, using a C++ computational geometry library. This class would be ideal for undergraduate students who are interested in data analysis, computer graphics, visualization, robotics, computer vision, image processing, gaming and animation. Specifically, in this course, we will touch on the following topics with an emphasis on data and applications: geometry foundations; polygons; convex hull in 2D and 3D; Delaunay triangulations and Voronoi diagrams; curves; meshes; polyhedra; as well as selected topics. This course will be followed by an advanced graduate course in computational geometry (CS 6160: Computational Geometry) for highly motivated students.

The class is formulated where the lectures focus on theoretical and algorithmic foundations for computational geometry, while the assignments are designed to give hands-on experience solving geometric problems in 2D and 3D. Instead of programming from scratch, most projects are centered around programming with computational geometry libraries.

Course Topics (subject to change)

- Geometry foundations: motivation, primitives, transformation
- Polygons and Art Gallery Theorem
- Convex hull: convex hull in 2D and 3D, and applications
- Triangulations and Voronoi diagrams: Delaunay triangulation and special cases; graphs, 2D, 3D and weighted constructions, duality
- Curves: medial axis, reconstruction
- Tetrahedron meshes
- Polyhedra
- Surfaces: reconstruction, surface simplification
- Selected topics and open problems

Prerequisite

The recommended prerequisite is CS 4150. However, if you have not taken CS 4150, you will need at least some programming background (C++, Python or Java), preferably C++, such that you are able to use computational geometry libraries. For example, CGAL (`http://www.cgal.org/`) or its Python/Java bindings (for example, `https://github.com/CGAL/cgal-swig-bindings`).

Communication & Getting Help

A key responsibility for a student in this course is to use the online class website and to check it regularly for due dates, updated materials, and corrections. The class website is at:

`http://www.sci.utah.edu/~beiwang/teaching/cs4960.html`

Take advantage of the TA and the instructor office hours. Both the instructor and the TA will work hard to be accessible to students. Please send us email if you need to meet by appointments. Please do not hesitate to ask questions: come to office hours, send emails, or speak up in class!

The TA will provide full support to programming using CGAL in C++, and some support for using the CGAL-bindings for Python/Java.

Students are encouraged to use discussion group for additional questions outside of class and office hours. The class will rely on the **Canvas** discussion group. Feel free to post questions regarding the class: assignments, schedule, class materials, etc. Also feel free to answer questions: learn from each other! The instructor and TA will also be actively answering questions. However, please **do not post potential homework answers or source codes central to the projects**. Such posts will be removed immediately and not answered. All important announcement will be made through Canvas. There is no class mailing list.

Students are expected to check their email, Canvas account, and the class website regularly.

Course Materials

There required textbook is Discrete and Computational Geometry by S. L. Devadoss and J. O'Rourke, 2011.

There might be supplementary readings available online, or through materials posted on the course website. There are also a few computational geometry libraries that the class might use to complete class projects. These should all run on a standard computer or can be found on machines in the college of engineering CADE Linux computer labs.

Lectures

The instructor will make use of slides and mostly white boards during lecture, and the slides (along with any other materials pertinent to the lecture) will be posted on the class website. Students are encouraged to take notes in class and should not expect to rely solely on posted slides to recall the material covered in each lecture. Students are expected to participate actively by asking and answering questions. The lecture schedule is posted on the class website as a weekly outline, and some lectures have corresponding reading assignments.

Assignments, Late Policy & Testing

Students will practice the concepts learned in the classroom by completing class assignments. Most assignments will be computer-based projects and will entail investigating computational geometry concepts using computational geometry library. Each assignment will clearly indicate how and when students should submit their solutions. Most assignments should be submitted via Canvas as PDF files indicating projects outputs; if the assignment includes programming, source code should also be submitted via Canvas.

Students are expected to submit completed assignments by the due date and time. To get full credit for an assignment, it must be turned in through Canvas by the start of class, specifically 3:40 p.m. Once the deadline is missed, those turned in late will lose 10% of its total points for each subsequent hour until it is turned in. Therefore, assignments will not be accepted more than 10 hours late, and will be given 0.

Student progress will be evaluated throughout the semester by a number of short, simple quizzes to be given in class. Quiz dates will be announced with at least one-week notice.

If you believe there is an error in grading (assignments or quizzes), you may request a regrading within one week of receiving your grade. Requests must be made in writing, explaining clearly why you think your solution is correct.

There is no final exam, but a final project.

Grading

The final course grade will be based on a number of assignments (mostly projects) (70% total), a number of evenly-weighted quizzes to encourage class participation(10% total), a final project (20%). There might be opportunities throughout the class to obtain bonus points.

The total 100 points are distributed across projects and exams as:

- In class quizzes (10 points): 5 quizzes, each 2 points, roughly 10 minutes each
- Project Hello World of CGAL (10 points)
- Project on Convex Hull (15 points)
- Project on Delaunay Triangulation (15 points)
- Project on Voronoi Diagram (15 points)
- Project on Surface Mesh (15 points)
- Final Project in 3D Geometry (Get creative!) (20 points)

Most projects should be completely individually, with the exception for the final project, where students can choose to work alone, or work in groups of at most 2 people. Students who choose to complete the final project as a group need to state clearly in their final project report what they have individually contributed to the project. And projects done by a group is expected to have a higher quality and complexity than those done by individuals.

Scale for assigning letter grades is as follows. This scale might be curved based on overall class performance, while ensuring fairness to all.
A 100-93 **A-** 93-90
B+ 90-87 **B** 87-83 **B-** 83-80
C+ 80-77 **C** 77-73 **C-** 73-70
D+ 70-67 **D** 67-63 **D-** 63-60
E 60-0

Working Together

Students are encouraged to discuss assignments with fellow classmates, but each student is responsible for writing an individual answer or writing their own codes (with the exception of final project). Cheating is: sharing written or electronic work either by copying, retyping, looking at, or supplying a copy. Cheating is not: discussing concepts, answering questions about concepts or clarifying ambiguities, or helping someone understand how to use the class tools and software.

Of course, there must be no collaboration during quizes. Please see the University of Utah Student Code (www.regulations.utah.edu/academics/6-400.html) for a detailed description of the university policy on cheating.

Students with Disabilities

The University of Utah seeks to provide equal access to its programs, services, and activities for people with disabilities. Students who need accommodations in this class should give reasonable prior notice to the Center for Disability Services, 162 Olpin Union Building, 581-5020 (V/TDD). CDS will work with the student and instructor to make arrangements for accommodations.