Introduction to Computational Topology

Hsien-Chih Chang
HSIEN-CHIH.CHANG@DARTMOUTH.EDU

Course Description

Topology is the art of studying shapes without precise measurements. It is not surprising then that topology has found many applications in computer science, both in theoretical and applied research including algorithms and complexity theory, data analysis, robotics, computer graphics, etc., where often the input data is geometrically constrained, or noisy due to measurement errors.

The course serves as an introduction to the rapidly growing area(s) of computational topology. Naturally, due to the vast amount of work and literature, the topics covered in this class will be biased towards the interest and expertise of the instructor.

Comparison to similar course(s) offered at Dartmouth:

- MATH 74 (Algebraic Topology): This is a more rigorous treatment of the mathematical theory behind computational topology. We will focus more on the applications in computer science and not always give full proofs to the results. For students who took MATH 74 before or are familiar with the language of topology will benefit by seeing how the concepts are applied in computer science.

Course Objectives

By the end of the course we expect the students to be able to read articles and follow the basic terminology required to conduct research in computational topology.

Prerequisite

Students are assumed to have reasonable math maturity, in particular the ability to read and write proofs. COSC 30: discrete math or equivalent is required as prerequisite. Experience in the analysis of algorithms (say COSC 31: algorithms) is strongly recommended but not required. Previous exposure in linear algebra and graph theory will also help. We will hand out homework 0 in the first week and give the students an idea of the background knowledge expected. Background in general topology (the study of topological spaces) is not required.

Textbook

There is no *required* textbook. We will use a mix of books and class notes listed below (and sometimes papers) as references; none of them covers the exact set of topics we will talk about. The relevant reading materials will be emphasized on the course webpage each week for the interested students to read in their off-time.

- Jean-Daniel Boissonnat and Monique Teillaud, *Effective Computational Geometry for Curves and Surfaces*
- Éric Colin de Verdière, *Algorithms for Embedded Graphs*
- H. Edelsbrunner and J. L. Harer, *Computational Topology — An Introduction*
- H. Edelsbrunner, *A Short Course in Computational Geometry and Topology*

- R. Ghrist, *Elementary Applied Topology*
- Francis Lazarus and Arnaud de Mesmay, *Computational Topology: Lecture notes*
- Jiří Matoušek, *Using the Borsuk-Ulam Theorem*
- John Stillwell, *Classical Topology and Combinatorial Group Theory*
- Afra Zomorodian, *Topology for Computing*

Teaching Methods and Expectations

The class will be lecture-based, with a small set of (bi-)weekly homework to make sure that the students are following the material. Depending on the background and feedback from the students, we will adjust the topics covered. Outside the regular lecture hours the instructor will also provide office hour on a weekly basis; all students are encouraged to attend this discussion-based session.

The students are expected to spend reasonable amount of time working on the assignments, which is another main component of the course along with the lectures. Discussions with other students and using online resources are encouraged. Some homework problems require additional readings; all the extra readings required for the homework will be made available.

Grading

Due to COVID-19 it is unclear if physical exams are possible, and therefore the final grading will be (in principle) based on homeworks and a *class project*. Each student will choose a topic related to computational topology (broadly interpreted) of their interest; the instructor will provide a list of possible topics. Students who share the same interest are encouraged to work together. At the end of the semester, each person/group has to give an *oral presentation* in class, and provide a *final report* summarizing the work, including (but not restricted to) a summary of the papers read, identifying research questions related to their own field, and the progress made. Students who are interested in implementation are also encouraged to demo the result.

The final grading will take into account the difference between student groups (undergrads vs graduates) and the impact of pandemic in the Fall. The tentative weights are homework (60%), project presentation (20%) and final report (20%).

Tentative Course Schedule and Assignments

All materials are subject to change based on the actual class interaction and student feedback.

- Week 1: ***Curves***
  - polygons, Jordan polygon theorem, planar curves, winding number(s); planar graphs, duality, medial construction
  - applications: inside-polygon testing, Monge property; planar graph compressions
- Week 2: ***Surfaces***
  - topological space, homeomorphism, quotient spaces, terrains and polyhedrons, polygonal domains, surface with genus and boundary, polygonal schema, Euler characteristic; surface graphs and curves, cellular embeddability, tree-cotree decomposition, system of loops, cut-graphs

  - applications: metronomes and cultures, mean valence for triangle/quad mesh; textiles

- Week 3: ***Homotopy***
  - homotopy equivalence, groups, fundamental groups, covering spaces; homotopy moves, face and spike moves, contractibility, crossing sequence
  - applications: homotopy testing, shortest homotopic paths and nontrivial cycles

- Week 4: ***Optimization***
  - multiple-sourse shortest paths, separator decomposition, Monge heap, shortest paths in planar graphs

- Week 5: ***Complexes***
  - meshes, Delaunay triangulation and Voronoi diagram, Vietoris-Rips and Čech complexes, nerve theorem, manifolds and configuration spaces
  - applications: mesh generation, shape analysis, rotation distance/triangle flips

- Week 6: ***Homology***
  - Counting, simplicial homology, Euler characteristic redux, curvature, Gauss-Bonnet theorem, fixed point theorems (Sperner's, Browner's), PPA complexity
  - applications: algorithmic game theory

- Week 7: ***Cohomology***
  - impossible configurations, Stoke's theorem, Poincaré and Alexander duality, Helly theorem, degree theory and Hopf theorem, max-flow min-cut and Laplacians
  - applications: Borsuk-Ulam theorem (from degree or homotopy extension), Ham-sandwich theorem, fair division

- Week 8: ***Morse theory***
  - critical points, Morse functions, Reeb graphs; evasiveness

- Week 9: ***Advanced applications in CS***
  - persistence, disjoint paths, decision tree lower bounds; the amount of materials covered depend on student feedback and suggestions

- Week 10: ***Final project presentation***