Notes for mini-course on computational topology

Erin Wolf Chambers
Department of Computer Science
Saint Louis University
erin.chambers@slu.edu

1 Introduction

Computational topology is a growing area that straddles mathematics and computer science; it consists of diverse pool of results and techniques which are a natural growth of fundamental computational problems. Computation in topology is not a new development; the first such "algorithms" date back to Dehn or even earlier. However, a vast array topological tools and structural results are now used by computational geometry and the broader algorithms community, as well as in a varity of other disciplines like machine learning [5], programming language design [18, 35, 25, 26], and distributed computing [29].

This document is meant to be a reference guide for a series of five talks given at the Intensive Research Program in Discrete, Combinatorial and Computational Geometry; as such, each of these talks will focus on various uses of computational topology that overlap most directly with algorithms, computational geometry, and graphics. This document is not a comprehensive overview of the lectures, or of computational topology; rather, it is meant to pull together a sequence of references from algorithmic topology that will be source material for these 5 lectures, as well as providing links for those who wish further information. (I am also happy to share the actual talk slides if anyone is interested; please contact me directly via email to request them.)

Lecture 1: An introduction to core tools

The first lecture will begin with the common tools from computational geometry that are most utilized in computational topology algorithms. Namely, we will briefly review constructs such as the Delaunay triangulation, the Rips complex, the Čech complex, and general simplicial complexes, with a focus on 2 and 3d. Finally, we will introduce homotopy and homology, ending with a discussion on computatbility and how these connect to more standard graph theory parameters.

For additional references, the book by Edelsbrunner is a nice introduction to many of these core tools in more depth [19]. I also really recommend the book by Ghrist [27], which is freely available online and gives an intuitive, high level introduction of topological concepts and notation with a focus towards applied topology. For a deeper understanding of topology, there are several classic textbooks [28, 33]. Of course, the book by de Berg et. al is considered the classic computational geometry reference [14]; I am also a fan of the book by O'Rourke and Devadoss [15].

For topics related to complexity or computation of homotopy and homology, I also find the course notes by Lazarus and de Mesmay [31] particularly helpful and nice to read.

Lecture 2: Topological notions of similarity

The question of how to measure similarity between curves in various settings has received much attention recently, motivated by applications in GIS data analysis, medical imaging, and computer graphics. While geometric measures such as the Hausdorff and Fréchet distance [1] have efficient algorithms, measures that take the underlying topology of the ambient space into account are less well understood. Several candidates have been proposed in recent years that are based on homotopy [6, 10, 12, 7] or homology [17, 11], but many of these are only tractable in restricted settings and surprisingly little is known about their practicality or utility.

In this lecture, we will survey known results (both geometric and topological), and then focus on some of the recent algorithmic results and remaining open questions for the topological measures. As there are unfortunately no books that really cover these topics, I will suggest any of the cited papers for further reading.

Lecture 3: Persistent homology

Since its introduction in several different contexts, persistent homology – the notion of calculating and updating homology groups in spaces that vary or change over time – has become a primary tool for data analysis, object recognition, and surface reconstruction and simplification. We will continue the discussion of homology from lecture 1, focusing on some of the first algorithms developed to compute persistent homology in common settings, based primarily on [21, 37]. No prior knowledge of persistence is necessary, besides a general familiarity with graphs and simplicial complexes and some basic familiarity with matrices and linear algebra.

A full discussion of persistence and its many uses is well beyond the scope of one lecture. The book by Edelsbrunner and Harer [20] is a classic look at persistence algorithms, although it is a bit more out of date in terms of recent results and developments. For those that wish to dive deeper, the recent book by Oudot [34] gives an excellent overview for both mathematicians and computer scientists, and includes a nice array of applications and open questions as well. For further examples and understanding the basic algorithms, I again like the course notes by Lazarus and de Mesmay [31] as well as the course taught by Wang [36].

Lecture 4: Graphs on surfaces

Many algorithms on planar graphs can leverage the simpler structure of the graph in order to gain considerable speedup; examples include algorithms to compute useful objects like shortest paths, or computing maximum flows and minimum cuts in these graphs. Motivated by this body of results, we will consider these problems on surface embedded graphs, which arise as an output of the surface reconstruction algorithms considered later in the week. Specifically, in this lecture we will focus on several algorithms to compute minimum cuts in these graphs [9, 24, 23], and then move to the slightly harder problem of maximum flows [8].

For further reading, there is book in preparation by Klein and Mozes that focuses on optimization algorithms for planar graphs that is a nice starting point for these techniques [30]. For a deeper overview of graphs on surfaces, the book by Bohar and Thomassen [32] is a slightly older look at the combinatorial structures and some algorithmic results. More recent coverage can be found in the notes by E. Colin de Verdiére, which are available on his webpage [13], or in the lecture notes by Jeff Erickson [22].

Lecture 5: Surface reconstruction

Finally, in this last lecture, we will discuss some of the most common surface reconstruction algorithms, based primarily on the Crust [2] and Powercrust [3] algorithms. We will conclude with some structural open questions in this setting.

For more discussion or information, there are several good books on these topic. Tamal Dey's book [16] covers the mathematical foundations of these algorithms in at very understandable framework. For a more practical discussion that covers implementations and software, I recommend the book edited by Boissonnat and Teillaud [4].

References

[1] H. Alt and M. Godau. Computing the Fréchet distance between two polygonal curves. *IJCGA*, 5(1–2):75–91, 1995.

[2] N. Amenta, M. Bern, and M. Kamvysselis. A new voronoi-based surface reconstruction algorithm. In *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*, volume 415–421 of *SIGGRAPH '98*, 1998.

[3] N. Amenta, S. Choi, and R. K. Kolluri. The power crust. In *Proceedings of the Sixth ACM Symposium on Solid Modeling and Applications*, SMA '01, pages 249–266, 2001.

[4] J.-D. Boissonnat and M. Teillaud, editors. *Effective Computational Geometry for Curves and Surfaces*. Springer, 2004.

[5] G. Carlsson, R. Jardine, D. Feichtner-Kozlov, and D. Morozov. Topological data analysis and machine learning theory: Final report from birs workshop. http://www.birs.ca/workshops/2012/12w5081/report12w5081.pdf, 2012.

[6] E. Chambers, E. Colin de Verdiére, J. Erickson, S. Lazard, F. Lazarus, and S. Thite. Homotopic fréchet distance between curves, or walking your dog in the woods in polynomial time. To appear in *Computational Geometry: Theory and Applications*, 2009.

[7] E. W. Chambers, A. de Mesmay, and T. Ophelders. On the complexity of optimal homotopies. In *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms*, 2018.

[8] E. W. Chambers, J. Erickson, and A. Nayyeri. Homology flows, cohomology cuts. In *Proc. 41st ACM Symp. on Theory of Computing*, 2009.

[9] E. W. Chambers, J. Erickson, and A. Nayyeri. Minimum cuts and shortest homologous cycles. In *Proc. 25th Symp. on Computational Geometry*, 2009.

[10] E. W. Chambers and D. Letscher. On the height of a homotopy. In *Proceedings of the 21st Canadian Conference on Computational Geometry*, pages 103–106, 2009.

[11] E. W. Chambers and M. Vejdemo-Johansson. Computing minimum area homologies. *Computer Graphics Forum*, 34(6):13–21, 2015.

[12] E. W. Chambers and Y. Wang. Measuring similarity between curves on 2-manifolds via homotopy area. In *Proceedings of the Twenty-ninth Annual Symposium on Computational Geometry*, SoCG '13, pages 425–434, New York, NY, USA, 2013. ACM.

[13] E. Colin de Verdière. Algorithms for embedded graphs. Course lectures notes available at http://monge.univ-mlv.fr/ colinde/cours/all-algo-embedded-graphs.pdf.

[14] M. de Berg, O. Cheong, M. van Kreveld, and M. Overmars. *Computational Geometry: Algorithms and Applications*. Springer, 2008.

[15] S. L. Devadoss and J. O'Rourke. *Discrete and Computational Geometry*. Princeton University Press, 2011.

[16] T. K. Dey. *Curve and Surface Reconstruction: Algorithms with Mathematics Analysis*. Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press, 2007.

[17] T. K. Dey, A. Hirani, , and B. Krishnamoorthy. Optimal homologous cycles, total unimodularity, and linear programming. In *Proceedings of the 42nd ACM Symposium on Comput. Theory*, pages 221–230, 2010.

[18] A. Edalat. Domains for computation in mathematics, physics and exact real arithmetic. *Bulletin of Symbolic Logic*, 3(4):401–452, 1997.

[19] H. Edelsbrunner. *Geometry and Topology for Mesh Generation.* Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press, 2001.

[20] H. Edelsbrunner and J. Harer. *Computational Topology: An Introduction.* AMS Press, 2009.

[21] H. Edelsbrunner, D. Letscher, and A. Zomorodian. Topological persistence and simplification. *Discrete Comput. Geom.*, 28:511–533, 2002.

[22] J. Erickson. Lecture notes from topology course. http://jeffe.cs.illinois.edu/teaching/topology17/schedule.html.

[23] J. Erickson, K. J. Fox, and A. Nayyeri. Global minimum cuts in surface-embedded graphs. In *Proceedings of the 23rd Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1309–1318, 2012.

[24] J. Erickson and A. Nayyeri. Minimum cuts and shortest non-separating cycles via homology covers. In *Proceedings of the 22nd Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1166–1176, 2011.

[25] M. Escardo. Mathematical foundations of functional programming with real numbers. Notes for a course delivered at the Midlands Graduate School in the Foundations of Computer Science, posted at http://www.cs.bham.ac.uk/ mhe/papers/mgs.pdf, April 2003.

[26] M. Escardó. Synthetic topology of data types and classical spaces. *Electronic Notes in Theoretical Computer Science*, 84, 2004.

[27] R. Ghrist. *Elementary Applied Topology.* Createspace, 1.0 edition, 2014.

[28] A. Hatcher. *Algebraic topology.* Cambridge University Press, 2002.

[29] M. Herlihy, D. Kozlov, and S. Rajsbaum. *Distributed Computing Through Combinatorial Topology.* Elsevier, 2013.

[30] P. Klein and S. Mozes. Optimization algorithms for planar graphs. http://planarity.org/.

[31] F. Lazarus and A. de Mesmay. Computational topology course notes. http://www.gipsa-lab.fr/ francis.lazarus/Enseignement/compuTopo.html, 2017.

[32] B. Mohar and C. Thomassen. *Graphs on Surfaces.* Johns Hopkins University Press, 2001.

[33] J. R. Munkres. *Elements of Algebraic Topology.* CRC Press, 1996.

[34] S. Y. Oudot. *Persistence Theory: From Quiver Representations to Data Analysis*, volume 209 of *Mathematical Surveys and Monographs.* American Mathematical Society, 2015.

[35] J. V. Tucker and J. I. Zucker. Abstract versus concrete computation on metric partial algebras. *CoRR*, 2001.

[36] Y. Wang. Lecture notes for introduction to computational topology. http://web.cse.ohio-state.edu/ wang.1016/courses/788/, 2011.

[37] A. Zomorodian. *Topology for Computing.* Cambridge University Press, 2005.