Donald Sheehy

Geometry, Topology, and Data

1 Executive Summary

When I lived in Pittsburgh, the local classical music station had a slogan reminding listeners that "all music was once new". The same holds in computer science where the basic concepts that any undergraduate might be expected to learn were once the cutting edge of theory. It is easy to forget this. I approach theoretical computer science with the clear vision that I am searching for the algorithms and data structures that will be standard, ubiquitous, even "obvious" in 10, 15, or 20 years. I do this by focusing on high impact areas of scientific importance where foundational questions remain wide open.

My work started mainly in mesh generation, the essential preprocess for numerical solution of partial differential equations by the finite element method. Basic questions in mesh generation have lingered unanswered for decades, despite being the focus of a massive engineering effort distributed across major industries including big money players in automobiles, aviation, and petroleum. I have been working to answer the primary algorithmic questions about the construction of finite element meshes with an eye towards the major pain points for those using meshes in the field.

Recently, I have leveraged my expertise in mesh generation, geometric algorithms, and data structures to attack problems in data analysis. The underlying structure in large, high-dimensional data sets can be described geometrically and topologically. The growing fields of topological data analysis and geometric inference attempt to reveal, describe, and exploit this structure. This inherently multidisciplinary area brings together tools from statistics, differential geometry, algebraic topology, combinatorics, and algorithms.

I try to bring creativity, ambition, and perseverance to my research, seeking novel methods to solve the most important problems in mesh generation and topological data analysis. In the following sections, I have detailed some of the ways I have brought together different areas of research to solve problems, whether its using ϵ-nets in mesh generation or metric spanners in topological data analysis. Here are some of the highlights of my research career thus far.

1. The first time-optimal, output-sensitive algorithm for quality mesh generation [MPS11].

2. The first proof that the ubiquitous technique of placing a bounding box around an input to a mesh generator does not add more than a constant factor more work [HMPS09].

3. The first use of mesh generation to give guaranteed approximations in topological data analysis [HMOS10].

4. The first linear-size approximation to what is arguably the most widely used tool in topological data analysis, the Vietoris-Rips filtration [She12a].

5. A simple output-sensitive algorithm for computing Voronoi diagrams and Delaunay triangulations in higher dimensions that achieves polylog running time per output face without the near-quadratic preprocessing of previous methods.

2 Computational Geometry

Geometric structure lurks explicitly or implicitly in all kinds of computing applications. It is most explicit in geometric modeling problems for physical objects and spaces, but it is implicitly present in all manner of data sets where the underlying space has fewer degrees of freedom than the number of measurements per data point. The goal of computational geometry is to efficiently represent, store, search, summarize, and transform geometric objects. Although primarily a subfield of data structures and algorithms, it has intimate ties with linear algebra, discrete geometry, differential geometry, and algebraic topology.

A recurring theme in my work (perhaps *obsession* is a better word) is the interaction of geometric, combinatorial, and topological structure in algorithms and data structures. Consider for example the divide-and-conquer paradigm of algorithm design. Dividing a geometric object such as a set of points can be done geometrically (dividing into equal radius sets) or combinatorially (dividing into equal cardinality sets). There is a tension between these approaches as one yields running times that depend on the geometry of the input, the other yields running times that depend on the size of the input, and neither is strictly better than the other. My work on fast point location algorithms for mesh generation exploits this tension to give the first optimal-time algorithm (see below) [MPS11].

The geometric/combinatorial distinction is analogous to the difference between the mean and the median of a set of numbers. For points in Euclidean space, the mean is a geometric center of the points, whereas the median is not so easy to define. I have worked on algorithms for computing good medians, that is to say, combinatorially central points [MS10, MPS09]. These are used both to partition data sets for divide and conquer and also to give short summaries of data sets.

I am also interested in output-sensitive algorithms in computational geometry. Such algorithms are useful because there is often a large difference between the best-case and worst-case complexity of the objects we want to compute. A prime example is the Voronoi diagram, a classic target for output-sensitive algorithms. Recently, I developed a new algorithm for computing Voronoi diagrams in higher dimensions that achieves the best known output-sensitive running time for a wide range of inputs. It is illustrated in Figure 1.

My research plan in computational geometry going forward is to extend my previous work and to explore relaxations of the type of incremental algorithms commonly used in practice. To some extent, these algorithms maintain a topological invariant as an algorithmic

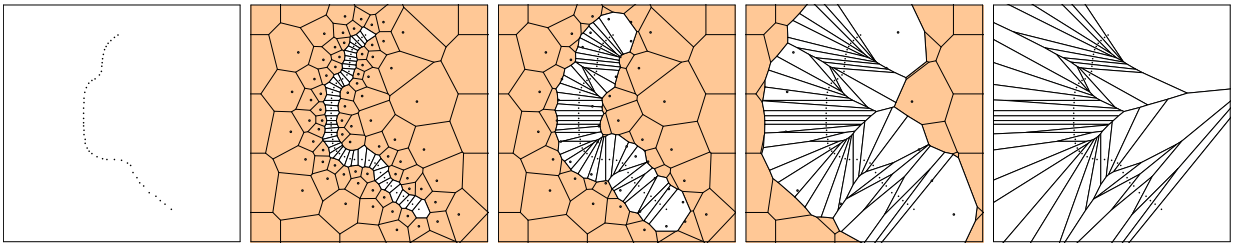

Figure 1: An illustration of our output-sensitive Voronoi diagram algorithm from left to right. Starting with a point set, it is extended to a quality mesh. Then the weights of the input points are increased until the extra cells disappear.

invariant, but this is often unnecessary. It can be problematic because in many cases, the underlying combinatorial structure is representing an underlying topology, but that combinatorial structure is determined by numerically unstable geometric computations. I am looking at how relaxations in this area lead to what I call "intrinsically robust" geometric algorithms. This would be a cheap alternative to the exact computation methods currently employed.

3 Mesh Generation

In many industrial processes, there is a person known officially as the mesh generation expert and unofficially as the "mesh monkey". This person has the unglamorous job of interfacing with meshing software and often to manually clean up the output.

Mesh generation is a classic field within computational geometry as it is linked to classic methods for the numerical solution of partial differential equations. A mesh is a decomposition of a space into simple elements such as simplices (triangles, tetrahedra, etc.) or hexahedra (like cubes) to provide a basis for representing (approximations to) functions on that space. The challenge problem in mesh generation is to build adaptive, dynamic, and kinetic meshes of high quality, of optimal size, and in optimal time. Moreover, this should all happen automatically; there should be no need for "mesh monkeys".

In our work on the Scaffold Theorem with Hudson, Miller and Phillips, we showed a precise condition under which a pervasive assumption in the meshing literature was actually true [HMPS09]. That result depended on a new approach to analyzing the complexity of meshes that gave a simple geometric method for interpreting the mess of integrals ubiquitous in the meshing literature [MPS08, She12b] The highlight of this line of research was the first time-optimal algorithm for meshing point sets [MPS11].

The subject of my ongoing and future research in mesh generation is the search for time- and space-optimal algorithms for meshing piecewise linear complexes as well as new algorithms for meshing in higher dimensions. I also plan to target some of the major pain points in this area such as numerical robustness issues in higher dimensions and badly formed simplices called slivers that lead to incorrect solutions.

4 Topological Data Analysis

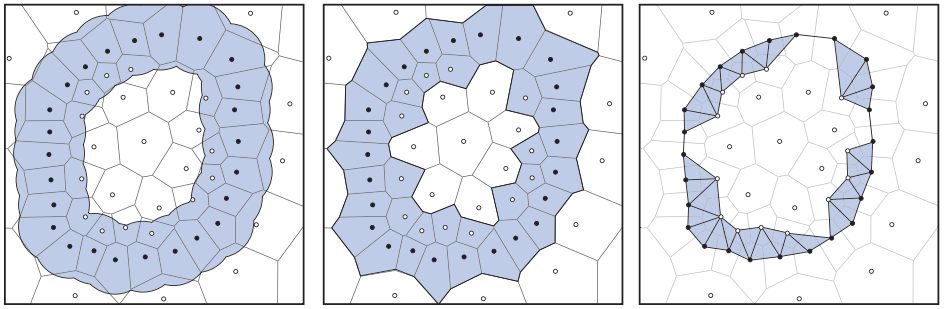

Figure 2: From left to right: the offsets of the black points for some radius, a corresponding subset of Voronoi cells in the mesh, and the subcomplex dual to the shaded Voronoi cells. This is the basic pipeline for approximating and analyzing persistent homology using meshes.

Persistent Homology and Mesh Generation Topological data analysis is an emerging field in data analysis that lives in the intersection of computational geometry, data mining, and algebraic topology. Persistent homology is a popular method in topological data analysis that gives a multiscale view of the intrinsic topological structure of a data set.

I am most interested in methods for exploiting geometric structure to improve the efficiency of computations. Persistent homology methods and mesh generation are a natural fit. In a persistent homology data analysis pipeline, one often looks at the simplicial complexes representing sublevel sets of a (Lipschitz) smooth function that is determined by the density of a set of points in Euclidean space. Mesh generation is exactly that field which aims to build simplicial complexes giving good approximations to smooth functions where the fidelity adapts according the density of a set of input points. This connection and its effective use was first detailed in my thesis work, and a specific application to distance functions was published in joint work with Hudson, Miller, and Oudot [HMOS10].

Sparse Approximations for Metric Data In topological data analysis, it is also common to consider data that comes equipped only with some *metric* structure, not necessarily *geometric* structure, i.e. coordinates. By far the most common way to impose a multi-scale topological structure on a finite metric space is to use Vietoris-Rips filtration. This object may be viewed as an ordering on all subsets of points based on their diameter (the distance between the farthest pair). Naturally, this object is huge, even if one only considers sets up to some fixed size. I proved that the Vietoris-Rips filtration for intrinsically low-dimensional data could be approximated with a linear-size construction [She12a]. Prior to my work, it was not possible to construct it in full for large data sets because its size is $n^{O(d)}$. The key idea was to incorporate ideas from metric approximation algorithms, thus revealing a connection between the problem of sparse approximations to metric spaces and sparse approximations to the underlying topological structure.

Future work in TDA I am currently working on several extensions to my work on sparse Vietoris-Rips filtrations, one of which shows how the same methods can be applied to fil-

trations that are more robust to outliers in data such as that induced by the distance to a measure (see [CCSM11]).

I am also exploring connections between homology computation and techniques from algebraic graph theory. For example, nested dissection is a combinatorial approach the classic problem of solving a symmetric positive definite linear system by Gaussian elimination that exploits graph separators to keep the complexity down. I am extending the theory of geometric graph separators to simplicial complexes to use nested dissection for homology computation on the types of simplicial complexes common in topological data analysis. This exploits the geometric structure in novel way to improve known algorithms. I believe this research holds the key to one of the outstanding problems of persistent homology, namely, why is there a large gap between the cubic worst-case performance of the persistent homology algorithm and the near-linear performance of the algorithm in practice.

References

[CCSM11] Frédéric Chazal, David Cohen-Steiner, and Quentin Mérigot. Geometric inference for probability measures. *Foundations of Computational Mathematics*, 11:733–751, 2011.

[HMOS10] Benoît Hudson, Gary L. Miller, Steve Y. Oudot, and Donald R. Sheehy. Topological inference via meshing. In *SOCG: Proceedings of the 26th ACM Symposium on Computational Geometry*, pages 277–286, 2010.

[HMPS09] Benoît Hudson, Gary L. Miller, Todd Phillips, and Donald R. Sheehy. Size complexity of volume meshes vs. surface meshes. In *SODA: ACM-SIAM Symposium on Discrete Algorithms*, 2009.

[MPS08] Gary L. Miller, Todd Phillips, and Donald R. Sheehy. Linear-size meshes. In *CCCG: Canadian Conference in Computational Geometry*, pages 175–178, 2008.

[MPS09] Gary L. Miller, Todd Phillips, and Donald R. Sheehy. The centervertex theorem for wedge depth. In *CCCG: Canadian Conference in Computational Geometry*, 2009.

[MPS11] Gary L. Miller, Todd Phillips, and Donald R. Sheehy. Beating the spread: Time-optimal point meshing. In *SOCG: Proceedings of the 26th ACM Symposium on Computational Geometry*, pages 321–330, 2011.

[MS10] Gary L. Miller and Donald Sheehy. Approximate centerpoints with proofs. *Computational Geometry: Theory and Applications*, 43(8):647–654, 2010.

[She12a] Donald R. Sheehy. Linear-size approximations to the Vietoris-Rips filtration. In *SOCG: Proceedings of the 28th ACM Symposium on Computational Geometry*, 2012.

[She12b] Donald R. Sheehy. New Bounds on the Size of Optimal Meshes. *Computer Graphics Forum*, 31(5):1627–1635, 2012.