Research Statement
Samuel Micka (samuelmicka@montana.edu)

The goal of my research is to use tools from topological data analysis (TDA) to efficiently differentiate between shapes. TDA is a cross-disciplinary field of research focused on analyzing the shape of data using techniques from applied algebraic topology, computational geometry, and algorithms. Topology may be used to quantify features based on the connectedness of the data such as connected components, cycles, and higher dimensional holes. One of the most common approaches for measuring the magnitude of these topological features in TDA is the persistence diagram (PD). The $k$th PD encodes topological and geometric information about $k$-dimensional holes (i.e., connected components, cycles, voids, etc.) in shapes; see Figure 1 for an example of the zeroth PD on a filtered graph.

TDA has been used to uncover important substructures in data across several disciplines [19, 21, 15, 23, 15]. Furthermore, research into representing shapes as sets of topological descriptors is receiving significant attention [22, 4, 20, 3, 11, 5, 13]. These sets of descriptors serve as a method of differentiating between shapes but — unlike many other shape representations — encode information about both the geometry and topology of the shape. As such, my work is focused on exploring and understanding new methods of analyzing shapes using the PD by developing novel algorithms for extracting information from, and comparing, PDs. Figure 2 depicts two research problems, which are the focus of my dissertation work, we have worked to address: (1) is the representation of shapes using PDs and (2) is developing efficient methods of comparing PDs.

![Figure 1: An example of the zeroth persistence diagram quantifying different connected components of a shape as a height parameter (Time) varies from zero to infinity (i.e., a height filtration). The persistence diagram (right), marks times when components are born (are first included) and die (are connected) during the height filtration.](image1)

Figure 2: The general workflow for differentiating between shapes with topological descriptors. Each shape is represented using a unique set of persistence diagrams. These sets are compared to one another using the bottleneck distance to differentiate between the shapes.

**My Contributions**

In my dissertation, I made significant progress on two open research problems in the field of TDA using techniques from algorithms, computational geometry, and computational topology.

**Representing Shapes Using Topological Descriptors**

In practice, shapes are often finitely represented as simplicial complexes which are sets of vertices, edges, triangles, and higher-dimensional generalizations of triangles. In 2014, Turner et al. proved that an uncountably infinite set of PDs could be used to represent and differentiate between simplicial complexes in $\mathbb{R}^2$ and $\mathbb{R}^3$ [22]. The topology encoded in these descriptors may yield advantages over methods that compute dissimilarity between shapes that are insensitive to connectivity, such as the Hausdorff distance, which may return small values for non-homeomorphic shapes. However, many of the theoretical properties of these representations remains unknown and increasing interest in the field makes developing practical methods of computing finite sets of representative descriptors of utmost importance. As such, we considered an open problem that emerged from the results of [22]: can we develop an algorithm for identifying a finite set of PDs that is unique to each simplicial complex in $\mathbb{R}^d$? We developed algorithms to address this question by showing that for plane graphs in $\mathbb{R}^2$ with $n$ vertices that there was a finite representation of the graph using $O(n^2)$ augmented PDs [2] and extended the approach to embedded graphs in $\mathbb{R}^d$ [1]. We are currently working on a result generalizing these techniques to simplicial complexes in $\mathbb{R}^d$. We also identified several other problems that arise when representing simplicial complexes with topological descriptors that do not encode information about non-critical vertices [10, 17].

**Developing Efficient Methods of Comparing PDs**

My second area of research is motivated by the problem of efficiently querying for nearest neighbors (NN). The bottleneck distance is the most common method of computing distances between two PDs. As research solutions utilizing the PD continue to develop, the need to efficiently search sets of PDs for similar subsets will become increasingly important. However, the best known approach for computing the bottleneck distance between two PDs with $m$ points is $O(m^{1.5} \log m)$.

![Figure 2](image2)
Thus, identifying the NN in a set of \( n \) PDs to a query PD requires \( O(m n^{1.5} \log m) \) where \( m \) is the largest number of points in any of the \( n \) PDs. We provided the first result for more efficient queries for the NN and \( k \) NN neighbors \([9]\). Specifically, we show that we can determine a six-approximation, in terms of the bottleneck distance of the NN, to a query PD in \( O((m \log n + m^2) \log \tau) \) time where \( \tau \) is based on the distribution of the points in the PDs. For identifying a set of \( k \) NNs, we provide \( k \) PDs that are each a twenty-four factor approximation of the \( k^{th} \) NN in time \( O((m \log n + m^2 + k) \log \tau) \).

These are the first known approximation results for searching in the space of PDs.

**Combining Research and Education**

Throughout my time as a graduate student I spent a significant amount of time working on CS education research. While working on the Storytelling Project at MSU\(^1\) which brings CS education to underserved middle schoolers in rural Montana, I developed outreach events and lesson plans that introduce CS to students with little to no prior experience in the field \[^2^,\] \[^3^]. Additionally, I investigated student perceptions of plagiarism to better understand techniques for encouraging academic honesty in the classroom \[^4^]. These projects have given me significant experience working with students on research and developing methods to break complex problems into digestible pieces. These skills have translated well to my work in TDA. In my research, I have worked with several undergraduates and other graduate students \[^2^,\] \[^1^]. Furthermore, each semester I have participated in the seminar for the Computational Topology and Geometry (CompTaG) club which involves giving talks on my own research as well as other recent results in the field.

To introduce my research to students at a new college, I will develop a seminar dedicated to algorithms in TDA and computational geometry. In this seminar, we will discuss open problems, provide research updates, and present recent results. I will hold additional office hours specifically for research discussions with students working for me or interested in algorithms, TDA, or computational geometry. Furthermore, I plan to continue my community outreach and efforts to bring CS education to underserved groups. I will be looking to work with students to help with these programs.

**Future Work**

Currently, our approaches for reconstructing simplicial complexes from PDs require the PD to be augmented and we utilize predicates, dependent on geometry inferred from points in the PD. However, a rich amount of information can be inferred from the topological information provided by the PD. In fact, the topological information can generate constraints, highlighting regions where particular types of simplices may lie. We conjecture that if these constraints are generated carefully, it will be possible to reconstruct the complex with significantly fewer PDs when using a constraint satisfaction library. We will develop algorithms to generate constraints and prove the correctness of these approaches. Furthermore, we will run experiments to test the practicality of the approach on benchmark datasets such as MPEG7 or MNIST. Additionally, constraints can be generated from other topological descriptors. We plan to compare the constraints generated by these various descriptors and evaluate the complexity of the various constraint-based approaches.

As more research continues to be built around PDs, additional methods for determining similar PDs which are efficient and robust must be developed. As such, an implementation of the algorithm described in \[^9^] would be beneficial for projects oriented around large data sets. Furthermore, it may be possible to lower the approximation bounds on computing the \( k \)-NN using techniques from the literature on locality-sensitive hashing, spanners \[^14^]\, and vectorization techniques \[^6^]\.

I plan to hire students to help implement this algorithm as an add-on for Dionysus\(^2\), a library developed for computing persistent homology. Furthermore, I will work with students to develop new algorithms and approximation factors for this searching problem. The problem, being geometric and easy to visualize, is very accessible to undergraduate students.

To fund research in my lab and future work, I plan to apply for NSF grants. I will target the Computer and Information Science and Engineering (CISE) Research Initiation Initiative (CRII) grant through the NSF to fund work on the implementation of the algorithm in \[^9^]\ and for efforts developing new lower-bounds for approximate nearest neighbor querying. To facilitate my research committed to developing efficient methods of finding sets of PDs representative of shapes using constraint based approaches, I plan to submit a grant proposal to the Algorithmic Foundations (AF) Program at the NSF. This grant will include developing a framework for efficiently differentiating between shapes using the implementation of \[^9^]\ to quickly compare sets of PDs. To publish our results, I will encourage students to submit to several venues accessible to new researchers such as: Fall Workshop on Computational Geometry, Canadian Conference on Computational Geometry, and Young Researchers Forum at the Symposium on Computational Geometry. Furthermore, we will submit significant results to the Symposium on Computational Geometry, the Symposium on Discrete Algorithms, the Symposium on Foundations of Computer Science, the European Symposium on Algorithms, and the Symposium on Theory of Computing.

1. http://www.montana.edu/storytelling/
2. https://www.mrzv.org/software/dionysus2/
References


