Persistent Homology-Based Approaches to Localization

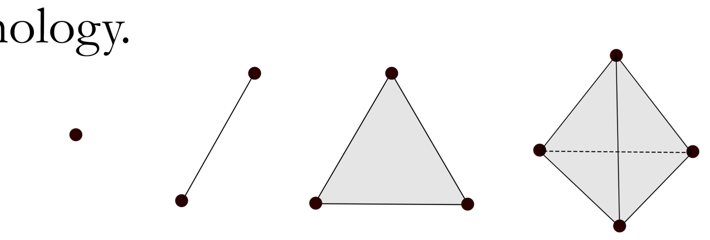
Luke Askew, Clare DuVal, George Engel, Brittany Terese Fasy, David L. Millman, Binhai Zhu

Abstract

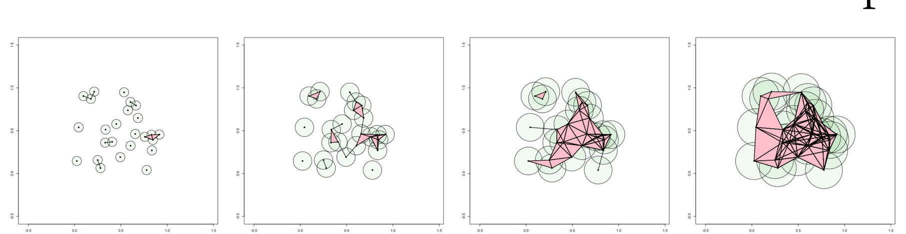
The goal of this study was to explore techniques to identify our position in a given space without the support of GPS, and identify key characteristics for building a strong solution for GPS denied localization. In particular, we use geometric and topological features extracted from LiDAR scans to index a database of locations. When a user wishes to perform a localization query, they perform a LiDAR scan locally, generate a set of features, and search for similar features in the database. In this project, we explore the sensitivity of the searching under varying features.

Introduction

Simplicial complexes are combinations of generalized triangles. Upon a simplicial complex, a triangulable topological space can be constructed, allowing for the use of tools from persistent homology.

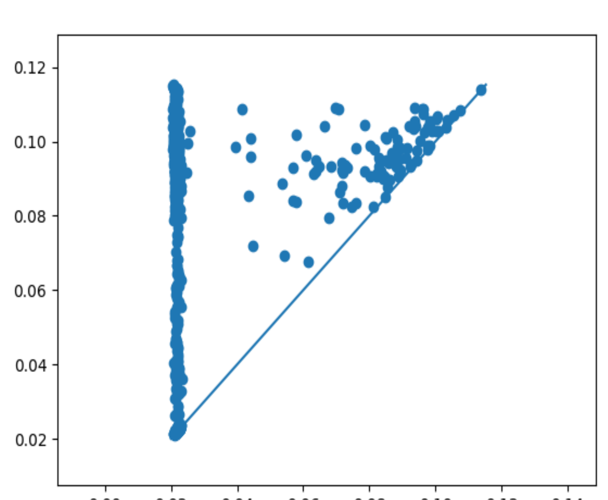


A **filtration** is a sequence of inclusions of nested topological spaces. In a **Vietoris-Rips Filtration** a triangulation of a point cloud is parameterized by the distance between points. For our data, we considered the Euclidian distance between points.

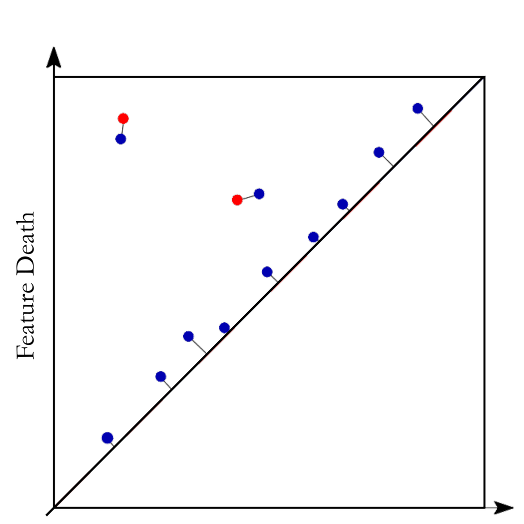


We also consider parameterizing by position of topological features with respect to a direction vector to construct a lower or upper star filtration.

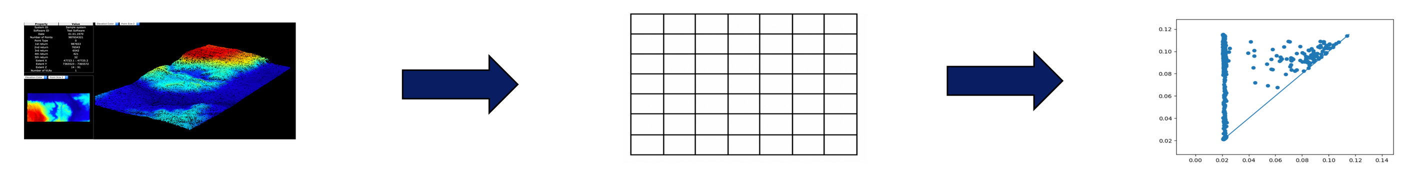
A **persistence diagram** of a filtration is a representation of the topological features of that filtration. By plotting the parameter at which topological features appear, their birth and death times, on the x-axis and where they merge with other features, their death times, on the y-axis we can construct a persistence diagram.



Bottleneck distance is a measure of similarity between two persistence diagrams. It measures the minimum of the maximal pairwise distances between points in a persistence diagram.



Preparation of Data

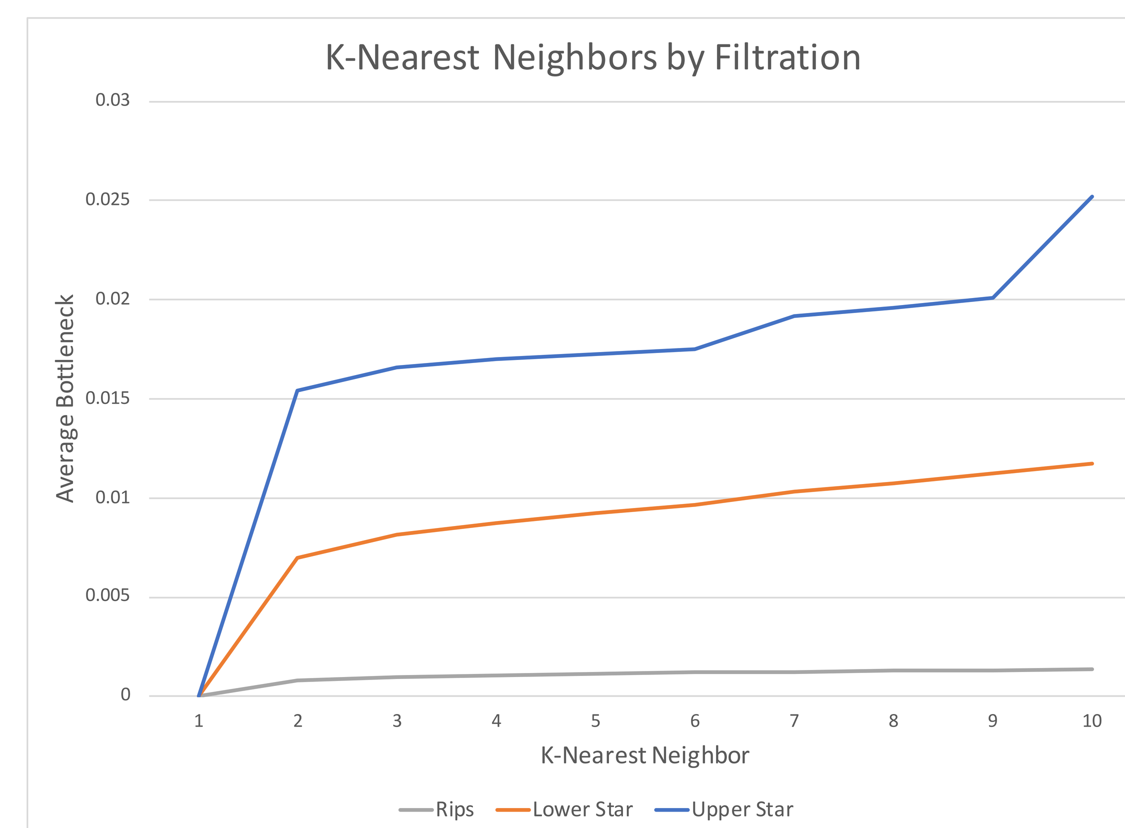


With a LiDAR data file of a 3-dimensional point cloud from Open Topography broken down into n^2 partitions by sectioning the x and y-axis, we calculated a persistence diagram for each partition using one of three filtrations. We then stored each partition's point cloud and persistence diagram in an indexed Point Cloud Persistence Diagram Section (PCPDS) object. The index is determined by the matrix index positioning from the 2-dimensional partitioning.

K-Nearest Neighbors

By searching for random PCPDS object by comparing it's persistence diagram to those from all other partitions and by ranking these comparisons by which had the lowest bottleneck distance, we identify the distance to the k-nearest neighbors.

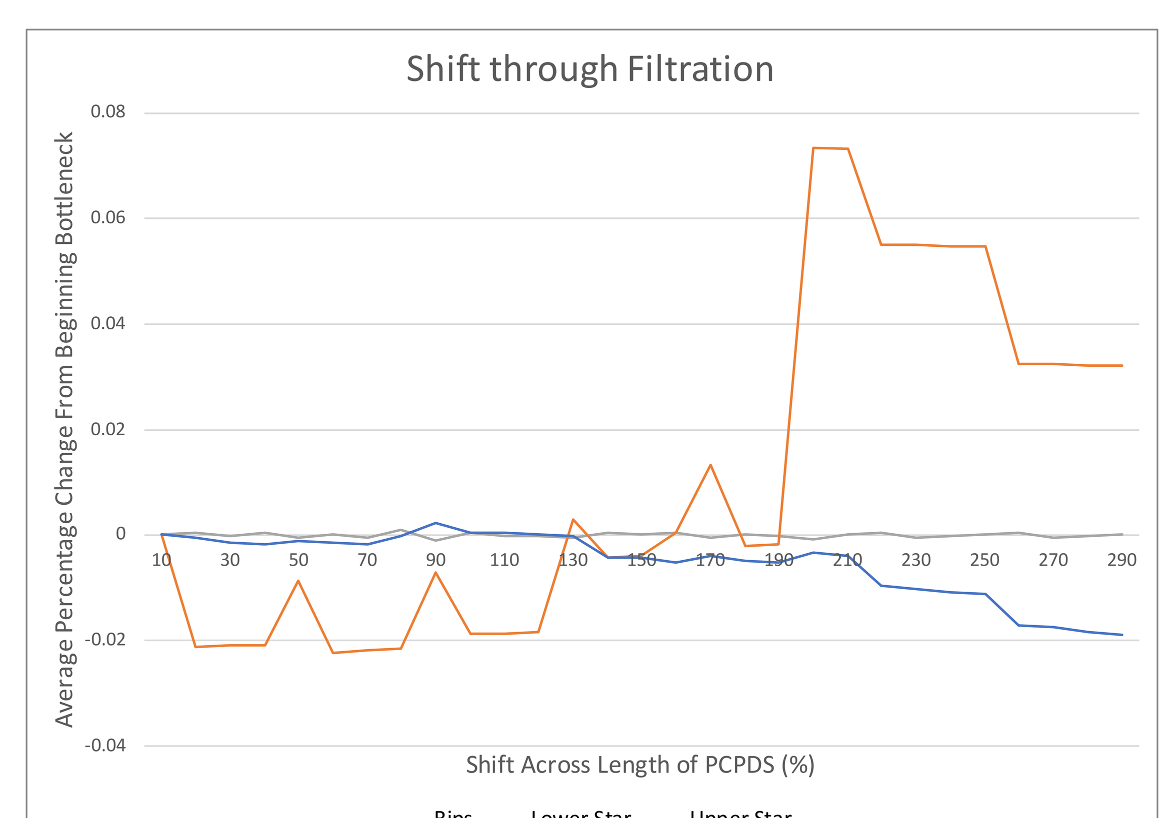
Knowing the average distance to the k-nearest neighbors of a search filtration in our point cloud for each filtration technique will allow for setting bounds on a heuristic search algorithm. Identifying filtrations with distances found to be anomalously low is information we would like to utilize in a heuristic.



Shift and Search

By shifting the bounds of PCPDS point cloud within the original dataset by increments of 10% of the size of the PCPDS's bounds, we compared the bottleneck distance to this new point cloud using different filtrations.

Because the average bottleneck distance between point clouds decreased for both filtrations as the distance between centers was increased, we conclude that accurate determination of local position using bottleneck distance requires overlap of more than 90% of area between point clouds when using our method of partitioning.

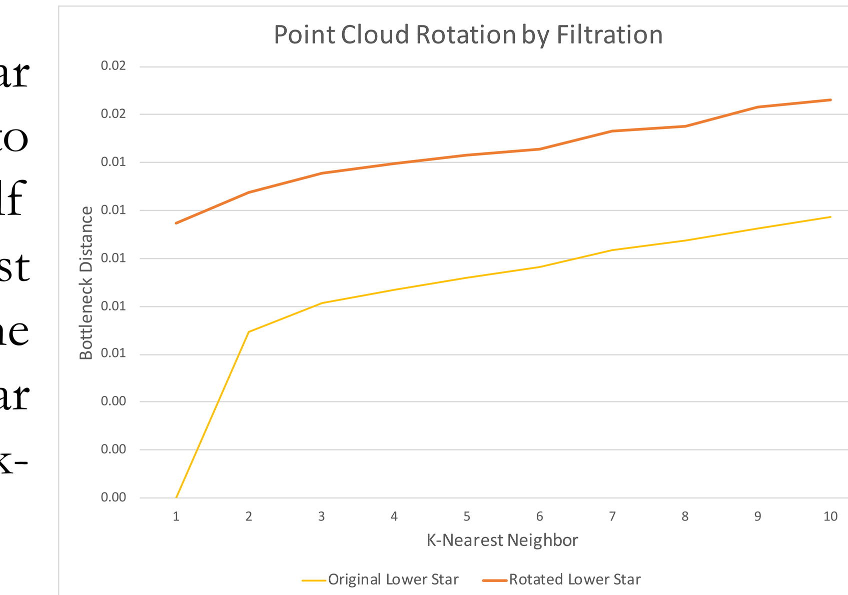


Check out the GitHub used to run to the experiments at https://github.com/compTAG/localization

Rotate and Find

We repeated our k-nearest neighbors experiment with a search point cloud rotated by $\pi/6$ radians about the data's z-axis to see how well the lower star filtration was able to identify its source point cloud.

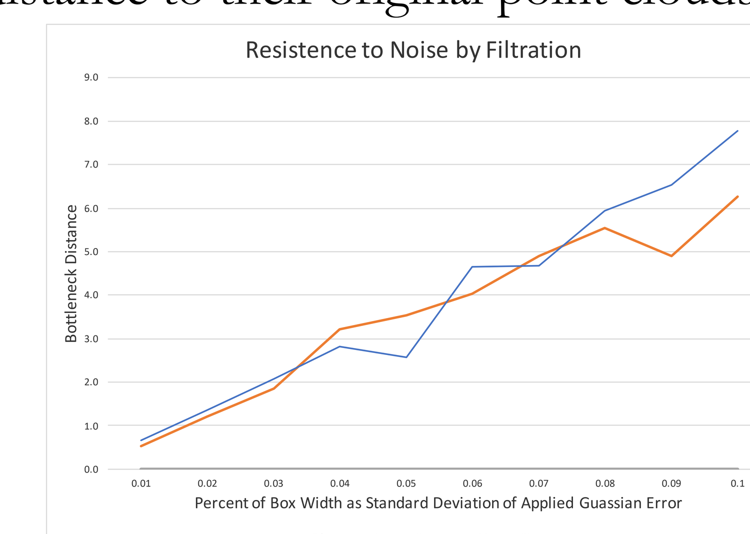
The lower star filtration was able to identify its rotated self as its first nearest neighbor 68.5% of the time, and had a similar slope to it's original k-nearest neighbors.



Add Noise and Compare

Bottleneck distances between persistence diagrams are provably stable for small perturbations in input, but measuring the impact of point cloud error on bottleneck distance directly lends itself to naïve search. We found that error of even 1% the size of the box width had significant impacts on the lower and upper star's bottleneck distance to their original point clouds.

However, the Vietoris-Rips filtration showed it was able to identify itself effectively even with applied error, with most bottleneck distances in the ten millionths.



Future Work

Our team's plans for this research is to seek to provide a probability driven solution to localization using partial search methods. This includes narrowing down the search by splitting existing partitions, then focusing on a collection of subsections with high match rates.

Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant No. NSF CCF 1618605. and NSF CSF 1658971. We would also like to thank Jordan Schupbach for reviewing and providing image software.

References

[1] Edelsbrunner, Herbert and Harer, John L., Computational Topology: An Introduction, American Mathematical Society,
Providence, Rhode Island, 2013
[2] Morozov, Dimitriy. A Practical Guide to Persistent Homology, Lawrence Berkeley National Lab, 2017. https://www.mrzv.org/software/dionysus/_downloads/dionysus-slides.pdf
[3]NSF OpenTopography Facility. EarthScope Intermountain Seismic Belt LiDAR Project. NSF OpenTopography Facility. 2009, https://doi.org/10.5069/g9vd6wcs
[4] The GUDHI Project. User and Reference Manual, GUDHI Editorial Board, 2016. http://gudhi.gforge.inria.fr/doc/latest/group__rips__complex.html