3D point cloud datasets are becoming more common due to the availability of low-cost sensors. Light detection and ranging (LIDAR), stereo, structured light, and time-of-flight are examples of sensors that capture a 3D representation of the environment. These sensors are increasingly found in mobile devices and machines such as smartphones, tablets, robots, and autonomous vehicles. As hardware technology advances, algorithms are needed to process sensor data in innovative and meaningful ways.

My research focuses on developing algorithms and data structures for fundamental robotic perception tasks such as 3D reconstruction, segmentation, and object detection and classification. I investigate the use of applied topology, specifically topological data analysis (TDA) for processing massive sensor datasets. Additionally, I work on problems in networked and cloud robotics where the goal is to allow under-resourced robots remote access to computing, data, and learning resources.

In this statement, I describe two current research projects that illustrate the broad spectrum of possible applications for my research as well as specific instances. In both projects I have made novel contributions and published in premiere peer-reviewed venues. As a faculty member, I plan to continue to extend the theoretical foundations of my work and apply the theory to relevant applications.

In the first project, I introduce topological methods to process 3D point cloud data. These methods include the capability to reconstruct, segment, and detect objects in 3D point clouds using persistent homology, a way to extract topological features such as connected components, holes, and voids. I address a problem in computational topology that allows non-bounding 1-cycles (holes) to be computed in nearly linear time on 3D datasets.

In the second project, I develop a perception architecture and algorithms for processing data between a robot and a remote computing infrastructure. This architecture is able to train on large-scale datasets, perform classification of 3D point cloud data, and efficiently transfer data in a robotic network. The source code is publicly available and is being used by researchers around the world.¹

**Topological Methods for 3D Point Cloud Processing**

TDA refers to a collection of tools for extracting topological features from data. The main tool, persistent homology, allows us to study homology (i.e. connected components, holes, and voids) at multiple scales. Concretely, it provides a framework to quantify the evolution of the homology of a parameterized family of topological spaces. For example, we can study the persistent homology of a distance-varying region of a 3D point cloud. We track the homological changes that occur as the distance parameter varies. This information can be visualized through a barcode diagram: a collection of horizontal line segments in a plane where the horizontal axis corresponds to a scale parameter and the vertical axis represents an arbitrary ordering of the homology generators. As the scale parameter changes, homology

¹https://github.com/wjbeksi
is born and dies. The pairing between births $b$ and deaths $d$ over varying spatial resolutions is known as a filtration, Figure 1. Topological features that exist for a long interval $d - b$ can be perceived as important, while features with small intervals are indistinguishable from noise.

Figure 1: The evolution of a scale parameter defining the neighborhood radius about each point (left) and the corresponding barcode diagram (right). When the neighborhoods of two points overlap, one dies while the other survives. At the end of this process, the lone surviving point (red) is a point of infinite persistence and is the representative of the set of points which forms a connected component.

**Contributions.** The contributions of my work in this area include the introduction of persistent homology theory to low-level robotic perception tasks. High-level tasks (e.g. object localization, feature extraction, classification, etc.) are dependent upon the quality of segmented data. Segmentation algorithms aim to divide a point cloud into constituent objects that are perceptually meaningful. In my research, I design and implement algorithms for segmentation based on computing the topological persistence of a point cloud at different spatial resolutions. To do this, the topology of the dataset is represented by generating an abstract simplicial complex (a collection of vertices, edges, and triangles closed under the operation of taking subsets). Then, the zeroth homology group of the complex, which corresponds to the number of connected components, is computed. Finally, the clusters of each connected component in the dataset are extracted. This method can be used, for example, to segment objects in a point cloud for a robotic vision system, Figure 2.

I have extended the aforementioned method to perform region-based segmentation. The notion of a region is essential in interpreting point cloud data since regions may correspond to objects, or parts of an object, in a scene. Moreover, region segmentation is a crucial preprocessing step towards pattern recognition and scene understanding. My contribution is the unique combination of global (topological) and local (color, surface normal) information, to produce a stable region growing segmentation of a noisy point cloud. Compared to related work in this area, this technique is fully automated and lacks the requirement
of an initial manual seeding of the dataset. As regions are grown based on local similarities between nearest neighbors, the global connectedness of the region is preserved using topology. Furthermore, the final segmentation does not depend on the order in which the regions are grown or joined. Currently, I am working on generalizing this idea to supervoxel segmentation.

Computing persistent homology entails the reduction of a boundary matrix, a matrix encoding the relationship between a simplex and its lower dimensional face, into a special form. An incremental approach to computing persistent homology was introduced by Edelsbrunner et al. for datasets restricted to three dimensions or less [9]. However, incrementally computing persistent 1-cycles has a runtime that is cubic in the number of simplices. I’ve addressed this problem by devising an algorithm for calculating topologically persistent 1-cycles in nearly linear time. A second algorithm utilizes the output of the first to generate a spanning tree upon which the location of non-bounding minimal 1-cycles can be found in a distributed manner. In practical terms, we can now take a topological approach to reconstructing point clouds with missing data, Figure 3. As part of my dissertation work, I plan to make available to the research community a Fast Library for Incrementally computing Persistence (FLIP).

Given the capability to quickly compute homology on large-scale 3D datasets by the previous algorithm, I’ve proposed a global shape descriptor STPP (Signature of Topologically Persistent Points) for object detection and classification. Unlike related work that uses the distances between persistence diagrams as topological proxies for the input data, STPP forms a feature vector comprised of the birth-death pairing of the homology generators. The end result is the ability to distinguish between noisy point clouds using only topological features. Moreover, STPP is competitive with state of the art geometrical feature descriptors. I’m currently working on designing methods that combine the differentiating power of geometry with the classifying power of topology for 3D point cloud description.
Published and Recognition. The research community has indicated interest in my work on topological methods for 3D point cloud processing. 3D object and region segmentation have been published in the International Conference on Robotics and Automation (ICRA) [3] and the International Conference on Intelligent Robots and Systems (IROS) [5], respectively. I was invited to present highlights of my work at the ICRA 2016 Workshop on Emerging Topological Techniques in Robotics [4]. A fast algorithm for computing topologically persistent 1-cycles with application to 3D point cloud reconstruction is under review in the Journal of Applied and Computational Topology [8]. Another paper for computing a signature of topologically persistent points for 3D point cloud description is being reviewed in the IEEE Robotics and Automation Letters (RA-L) [6].

Future Work. Although the focus of my research develops methods to process sensor datasets in robotic applications using TDA, these techniques are by no means limited to this domain. There are a rich variety of significant applications within the realm of $\mathbb{R}^3$. As a faculty member, my first step will be to submit an NSF proposal to apply these methods to robotics and other areas. In addition, I would like to establish connections with faculty in other departments. I am interested in working with collaborators to pursue several new research directions, including the application of multidimensional persistent homology.

Perception Architecture and Algorithms for Networked and Cloud Robotics

A cloud infrastructure, and its substantial set of readily available resources, has the potential to provide powerful benefits to robots and automation systems. Researchers have outlined four areas where connection to remote facilities is an asset: 1) Big data: access to databases of images, point clouds, and maps; 2) Cloud computing: access to parallel grid computing on demand for learning, motion planning, and statistical analysis; 3) Collective robot learning: robots sharing control policies, trajectories, and machine learning.
models; 4) Human computation: using crowdsourcing to access human expertise for image analysis, classification, and error recovery [10]. The capability of robotics and automation systems can be further enhanced by facilitating access to: benchmarks, simulation tools, and open-source software.

**Contributions.** RGB-D sensors are able to simultaneously capture both color and depth images. These sensors operate at high frame rates, and can produce over 10 MB/s of data thus allowing for potential bottlenecks in robotic networks. To mitigate this problem, I’ve developed algorithms that use information and communication theory concepts to intelligently throttle RGB-D data from a client (robot) to a server (cloud). The client makes use of redundant information in point cloud frames to reduce the amount of data transmitted. The server analyzes the received data and makes adjustments to the client’s rate of transmission by employing an adaptive entropy threshold. Together, the client and server work to maintain the usability of the network. I’ve shown that this approach makes it possible to conserve bandwidth and reduce network latency while allowing a mobile robot to perform vision tasks such as object tracking and classification.

Processing and classifying 3D point cloud data within a group of robots is a resource intensive task. Robots may not have the necessary on-board resources to perform this task, however computations and access to classifier models can be provided by a remote computing infrastructure or by other robots in the network. I’ve introduced a framework that allows for a distributed and scalable object classification paradigm among a group of robots. This framework uses RGB-D covariance descriptors (compact and low dimensional feature descriptors) in conjunction with dictionary learning (a collection of atoms that represent a sparse approximation of a covariance descriptor). I’ve demonstrated the increased performance of 3D object classification utilizing covariance descriptors and dictionary learning over previous results with experiments performed on a publicly available RGB-D dataset.

My work on this project has culminated in the design of a Cloud-based Object Recognition Engine (CORE), Figure 4. CORE is a distributed and scalable architecture that can utilize a cloud computing infrastructure for performing object recognition on a stream of 3D data. The architecture handles the following two use cases: 1) Multiple robots connect to the cloud to make use of computationally demanding and data intensive object recognition services; 2) When no cloud connection is available, robots can share their knowledge of machine learning models thus enhancing their object recognition capabilities.

**Publications and Recognition.** My work in this area has been published in top robotics conferences. I’ve presented object classification using dictionary learning and RGB-D covariance descriptors at the International Conference on Robotics and Automation (ICRA) [2]. Point cloud culling for robotic vision tasks under communication constraints [1] and a cloud-based object recognition engine for robotics [7] have been publish in the proceedings of the International Conference on Intelligent Robots and Systems (IROS). I was also invited to attend the NSFCloud 2014 Workshop on Experimental Support for Cloud Computing to provide input on experimental requirements for cloud robotics.
Figure 4: The architecture of the cloud-based object recognition engine. Robots perform as much on-board processing (e.g. culling, filtering, segmentation) as possible before sending compressed point clouds to the object recognition engine. The object recognition engine can be configured to train classifiers on large-scale datasets, and passes the results of object recognition queries along with a transmission entropy threshold back to connected robots.

**Future Work.** I will be working on extending the functionally of CORE in the area of semantic scene understanding. Experimental evaluations will continue to be carried out on CloudLab, an NSF funded testbed that allows researchers to experiment with cloud architectures. I’m interested in exploring the nascent Internet of Things (IoT) especially in the areas of privacy and security, time-varying network latency and quality of service, and the effective handling of big data. I would like to work with faculty members across different departments and initiate partnerships with industry to usher in smart homes, factories, and cities.

**References**


