3D Point Cloud Segmentation Using Topological Persistence

William J. Beksi and Nikolaos Papanikolopoulos

Abstract— **In this paper, we present an approach to segment 3D point cloud data using ideas from persistent homology theory. The proposed algorithms first generate a simplicial complex representation of the point cloud dataset. Next, we compute the zeroth homology group of the complex which corresponds to the number of connected components. Finally, we extract the clusters of each connected component in the dataset. We show that this technique has several advantages over state of the art methods such as the ability to provide a stable segmentation of point cloud data under noisy or poor sampling conditions and its independence of a fixed distance metric.**

I. Introduction

Segmentation algorithms aim to divide an image or point cloud into constituent objects or clusters that are perceptually meaningful. It is a vital preprocessing step in robotic systems. The performance of high level tasks such as object localization, feature extraction, and classification are dependent upon the quality of the segmented data. In recent years, 3D point cloud datasets have become increasingly ubiquitous due to the availability of low-cost RGB-D sensors [1]. These sensors can capture both color and depth images simultaneously, and have created much interest in the robotics community towards developing efficient algorithms for point cloud processing [2].

Point clouds can be filtered, segmented, compressed, etc., to support a given job. Segmentation of point cloud data is an active area of research due to its significance in many robotic perception applications. For example, a reconstructed model can be computed from objects segmented out from partial views and used in manipulation and grasping [3], segmentation of fused laser and camera data is suitable for mobile robotic tasks (object recognition, obstacle avoidance, terrain classification) [4], and a robotic system that can perform real-time 3D perception of the surrounding environment to reliably detect obstacles incorporates point cloud segmentation [5].

Segmenting objects in point clouds is a challenging problem. 3D points in a point cloud are often incomplete, sparse, unorganized, lack connection information, and have an uneven dispersion. Object clusters can be highly entangled. In addition, the topological features of an object's surface can be arbitrary and with no statistical distribution pattern in the data. Real-world data is noisy; the physical limitations of sensors, boundaries between features, multiple areas of reflectance, occlusions, etc., lead to the creation of outliers that make the process of segmentation difficult.

The authors are with the Department of Computer Science and Engineering, University of Minnesota, Minneapolis, USA. Emails: {beksi, npapas}@cs.umn.edu.

In this work, we implement and evaluate algorithms for segmentation based on computing the topological persistence of a point cloud dataset at different spatial resolutions [6], [7]. This procedure, known as persistent homology, is based on the concept that topological features detected over a range of varying scales are more likely to represent true features of the underlying dataset rather than artifacts of noise, poor sampling, or a particular choice of parameters. The key contributions of this paper are (1) the introduction of persistent homology to the area of point cloud processing for 3D perception; and (2) a novel approach for segmenting 3D point clouds based on topological persistence.

The remainder of this paper is organized as follows. Previous research work is presented in Section II. In Section III, we provide a brief overview of the mathematical concepts used in this work. Section IV states the problem followed by our proposed solution in Section V. Experimental results are given in Section VI. The paper concludes in the last section and discusses future work.

II. Previous Work

Data segmentation has been a heavily worked on research problem for decades with a rich history of literature. Many different approaches have been taken in segmenting both 2D and 3D data. In the following paragraphs we highlight several areas of research that are relevant to our work.

A variety of algorithms have been proposed in computer graphics on the segmentation of 3D models of single objects where the objects are typically represented by meshes [8]. These algorithms include watersheds [9], k-means [10], hierarchical [11], and spectral clustering [12]. The objective of these methods is to decompose an object into meaningful parts. Typically a graph is constructed from an input mesh. Clustering is then performed on the graph to produce a segmentation using graph cuts.

Prior to the appearance of cheap RGB-D sensors, datasets produced by laser scanning and stereo vision technologies fueled segmentation-based research. [13] segments foreground objects from background clutter in outdoor scenes using a min-cut method on a nearest neighbors graph. A multi-class point cloud segmentation technique that proposes multiple seeding methods and a min-cut framework is developed in [14]. In [15], researchers demonstrate a set of segmentation methods for 3D point clouds of varying densities.

With the introduction of RGB-D sensors, interest in segmenting point clouds continues. Pre-segmentation based on surface normals followed by surface patch estimation and model selection to find a learned representation for the given data is presented in [16]. The authors of [17] develop

a segmentation strategy that extracts objects, defined as compact regions enclosed by the depth and contact boundary in a scene, as single regions using color, texture, and 3D information. Efficient planar segmentation of organized point clouds based on connected components is proposed in [18].

The idea of topological persistence has been used in computer vision for performing image segmentation. Hierarchical segmentation using the mean shift method is discussed in [19]. The authors apply these concepts to a density function in a 5-dimensional space that combines the x- and y-coordinates of a pixel along with its RGB color components. A hybrid split-and-merge image segmentation algorithm is given in [20]. The algorithm first performs edge detection on the input image. It then splits the image into regions using edge-directed topology based on persistent homology. Regions with with similar features are selected and merged in order of topological persistence. TopoCut, a way to integrate knowledge about topological properties into a random field segmentation model, is introduced in [21]. In contrast to these works on 2D images, we utilize topological persistence for processing point cloud data to support 3D perception tasks.

III. Mathematical Background

This work makes use of ideas stemming from algebraic topology. In the following subsections we provide a brief index of key definitions and constructs. A comprehensive treatment of the subject can be found in [22].

A. Simplices

The discrete space that we work in uses simplices as the building blocks.

Definition 1: A d-simplex σ is the convex hull of $d+1$ affinely independent vertices $v_0, \ldots, v_d \in \mathbb{R}^n$. We denote $\sigma = \text{conv}\{v_0, \ldots, v_d\}$ where the dimension of σ is d.

For any of the vertices, v_i, the d vectors $v_j - v_i$, $j \neq i$, are linearly independent. In other words, given a set of $d+1$ vertices a simplex is the set of points each of which is a linear combination of these vertices with nonnegative coefficients summing to 1. The convex hull is simply the solid polyhedron determined by the $d+1$ vertices. Examples of simplices can be seen in Fig. 1.

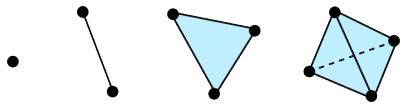


Fig. 1. A vertex, edge, triangle, and tetrahedron are simplices of dimension 0, 1, 2, and 3 respectively.

Definition 2: A face of σ is conv$S$ where $S \subset \{v_0, \ldots, v_d\}$ is a subset of the $d+1$ vertices.

For example, the four faces of a tetrahedron correspond to the four subsets of S obtained by removing vertices one at a time from σ. These four triangle faces are themselves 2-simplices. There also exists six edge faces and four vertex faces.

B. Simplicial Complexes

Our space of interest is composed by gluing together simplices to form simplicial complexes, Fig. 2.

Definition 3: A simplicial complex K is a finite collection of simplices such that if $\sigma \in K$ and τ is a face of σ, then $\tau \in K$ and if $\sigma, \sigma' \in K$ then $\sigma \cap \sigma'$ is either empty or a face of both σ and σ'.

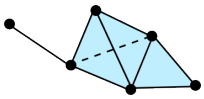


Fig. 2. A simplicial complex is constructed by gluing together simplices.

An *abstract simplicial complex*, based on a set of vertices $V = \{v_0, \ldots, v_d\}$, is a collection K of simplices closed under the operation of taking subsets: if $\sigma \subset V$ is a simplex ($\sigma \in K$) and τ is a face of σ ($\tau \subset \sigma \subset V$), then $\tau \in K$ as well.

C. Vietoris-Rips Complex

We make use of the following abstract simplicial complex to capture the topology of our space, Fig. 3.

Definition 4: Given a set of points $X = \{x_0, \ldots, x_m\} \in \mathbb{R}^n$ in Euclidean n-space and a fixed radius ϵ, the Vietoris-Rips complex of X is an abstract simplicial complex whose d-simplices correspond to unordered $(d+1)$-tuples of points in X that are pairwise within ϵ distance of each other.

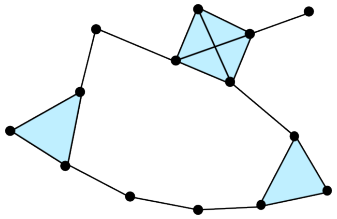


Fig. 3. An example of a Vietoris-Rips complex.

D. Homology Groups

Homology is an algebraic means to identify the holes in a topological space. It is based on the concept of a boundary homomorphism where simplicial homology encodes how simplices are attached to their lower dimensional faces.

Given a complex K, we choose an ordering for each simplex. Concretely, an ordering on a d-simplex, $\sigma \subset V$, is a literal ordering of its vertices $[v_0, \ldots, v_d]$, up to the equivalence generated by even permutations. The orientation of the simplex can be reversed by multiplying by -1. After choosing such an ordering, consider the $\mathbb{R}$-vector space $C_d(K)$ with its basis consisting of the oriented d-simplices in K.

Definition 5: A boundary homomorphism is the linear mapping $\partial_d : C_d(K) \to C_{d-1}(K)$ produced by associating each basis element of $C_d(K)$ to the formal sum of its oriented faces of dimension $d-1$.

An essential property of the boundary operator is that the boundary of a boundary is empty, $\partial_{d-1} \circ \partial_d = \emptyset$. With the

boundary operator $\partial = \{\partial_d\}$ providing the information on how to construct K, we can deduce certain global topological features from the resulting linear algebra. The dth homology group of K is expressed as

$$H_d(K) = \ker \partial_d / \text{ im } \partial_{d+1}, \quad (1)$$

$$\ker \partial_d = \{a \in C_d \mid \partial_d(a) = \emptyset\}, \quad (2)$$

$$\text{img } \partial_{d+1} = \{b \in C_{d+1} \mid \exists a \in C_d : \partial_{d+1}(b) = a\}. \quad (3)$$

In other words, $H_d(K)$ is the quotient vector space whose generators are d-cycles (which correspond to d-dimensional boundaryless subcomplexes surrounding a hole) modulo the equivalence relation which states that two such d-cycles are the same (homologous) whenever they are the oriented boundary of a $(d+1)$-dimensional subcomplex. Additionally, a main property of $H_d(K)$ is that its dimension (number of generators) corresponds to the number of d-dimensional holes in the simplicial complex.

IV. Problem Statement

Let X be a topological space where $X = \{x_0, \ldots, x_m\} \in \mathbb{R}^3$ and $x_0, \ldots, x_m$ are the points in a point cloud captured by an RGB-D sensor. To find the persistent homology of X it is necessary for the space to be in the form of a simplicial complex. Therefore, our first objective is to represent the topology of the space with a Vietoris-Rips complex. Our next objective is to determine the zero-dimensional homology group ($d = 0$) of X. When $d = 0$, a cycle is a linear combination of 0-simplices in X. A boundary is a linear combination of 0-simplices contained in the same connected component of X such that the sum of their coefficients is zero. This implies that ker $\partial_0 /$ im ∂_1 is precisely the number of connected components in X. Our final objective is to extract out the connected component clusters.

To achieve these objectives, we make the following assumptions:

Assumption 1: The input space is voxelized: Points in the space are approximated (downsampled) with their centroid.

Assumption 2: The input space is locally connected: Each point consists of a neighborhood containing open and connected sets. This constraint is enforced by using only the neighboring points during the segmentation process.

V. Segmentation Using Topological Persistence

Our solution to the problem outlined in IV comes in a set of algorithms and data structures for: 1) Constructing a Vietoris-Rips complex of the input space; 2) Using persistent homology to compute the zeroth homology group; 3) Extraction of the connected components in the space. The output of these algorithms is the set of clusters where each cluster is a set of points that are considered to be a part of the same connected component.

A. Constructing the Vietoris-Rips Complex

Given an input point cloud our goal is to form the Vietoris-Rips complex, $K = \{\sigma \subset \{x_0, \ldots, x_m\} \mid \text{dist}(x_i, x_j) \leq \epsilon, \forall x_i \neq x_j \in \sigma\}$, where dist is the Euclidean metric and the vertices of σ are pairwise within distance ϵ. We observe that the 1-skeleton of the Vietoris-Rips complex is sufficient to compute the zeroth homology group of the space. Thus, to generate K, we use the following algorithm (Alg. 1).

Algorithm 1 generateRipsComplex(points)

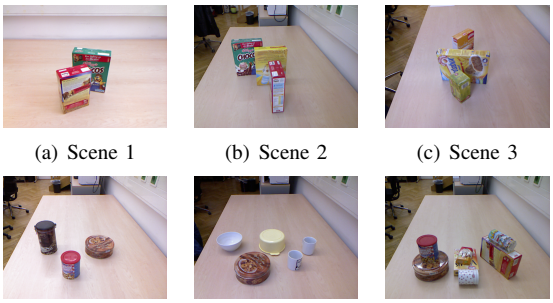
```
tree = KDTreeSearcher(points)
for all points do
    query = [point.x point.y point.z]
    nn = rangesearch(tree, query, radius)
    min_nn = min(nn)
    for all nn do
        if point(nn).parent > point(min_nn).id then
            root = find(point(nn))
            if point(root).id == point(nn).id then
                point(nn).death = radius
            end if
            union(point(nn), point(min_nn))
        end if
    end for
end for
```

The input to the algorithm is a list of points sorted in non-decreasing order. A kd-tree is used to carry out range searches for finding the nearest neighbors of each point at a given radius. Among the returned indices of nearest neighbors, we compare the parent of the current nearest neighbor to the minimum nearest neighbor. If the position of the parent occurs later on in the sorted list, then we proceed to find the root of the nearest neighbor point. In the case of a point not connected to any other point, we record the radius at which the points are to be joined. Finally, we union the points by making the parent the minimum of the pair. Since all faces of a simplex in a simplicial complex also belong to the complex, K is just the union of edge faces formed by this method.

B. Computing Persistent Homology

In the previous subsection, we constructed our simplicial complex with respect to the scale parameter ϵ. When computing homology, topological features of this simplicial complex may be due to noise or an inappropriately chosen scale parameter. Persistent homology solves this problem in the following way. We construct a *filtration* that stores the complexes across the entire range of possible values of the scale parameter. By allowing us to exclude short lived topological features, we can control how long a topological feature has to exist in the filtration before we consider it significant. Note that this implicitly assumes that the length of time (persistence) of the feature reflects its importance.

Given a simplicial complex K, let $f : K \to R$ be a non-decreasing function. Here non-decreasing means that if τ is a face of σ, then $f(\tau) \leq f(\sigma)$. The level subcomplexes are then defined to be $K(\alpha) = f^{-1}(-\infty, \alpha]$. If we denote by α_i the values of f on the simplices of K in increasing order, then the level subcomplexes define a filtration of K. Let



(a) Scene 1 (b) Scene 2 (c) Scene 3

(d) Scene 4 (e) Scene 5 (f) Scene 6

Fig. 4. RGB images of selected scenes from the Object Segmentation Database [25] used for testing.

$K_L = K(\alpha_L)$ where L is the index of the largest value α_L, then we have a filtration

$$\emptyset \subset K_0 \subset K_1 \subset \ldots \subset K_L = K. \tag{4}$$

As the scale parameter increases, new simplices are added until eventually we obtain the complete complex $K(\infty) = K$.

During the filtration process, we are interested in non-bounding cycles that remain nonbounding. For example, if we consider the step from $H_d(K_i)$ to $H_d(K_{i+1})$ the following changes may occur: new homology classes can be created (born), or existing homology classes can merge or become trivial (die), Alg. 2. The purpose of persistent homology is to keep track of those homology classes which are present throughout many iterations of the filtration. This information is coded in a barcode or persistence diagram. These diagrams are stable and can accommodate small perturbations (e.g. sensor noise) in the point cloud data [23].

Algorithm 2 computePersistence(points)

```
1: sort(points)
2: K = generateRipsComplex(points)
3: for all cycles ∈ K do
4:     Report death times
5: end for
```

C. Extracting Connected Components

We track the partitioning of the point cloud dataset into clusters of connected components (Alg. 1) using a disjoint-set data structure [24]. The data structure is initialized by making each 0-simplex its own set, i.e. all points are born at time zero. While performing the filtration, connected 0-simplices are merged within the data structure. At the end of the filtration, we find the connected components based on the sets of points that are joined to 0-dimensional simplices of infinite persistence.

VI. Experimental Results

In this section, we perform an evaluation of the topological persistence segmentation method outlined in Section V. The experiments are conducted using the Object Segmentation Database (OSD) [25]. The OSD provides RGB-D data in several subcategories to enable evaluation of object segmentation approaches. Six scenes selected from the database for testing are shown in Fig. 4. These scenes, labeled 1-6, contain a mixture of occluded and stacked objects. All experimental runs were done using MATLAB on a 64-bit GNU/Linux machine with a single CPU core.

Before segmentation occurs the data is preprocessed using the Point Cloud Library (PCL) [26]. First, we remove outlying points (background clutter) that are beyond range of the objects of interest. Next, the objects that we want to segment are typically resting on a flat surface such as a floor or table top. Therefore, we find all points that correspond to a planar model and remove them. We then downsample the point cloud with a voxelgrid filter using 1 cm^3 voxels. Finally, we construct the Vietoris-Rips complex, compute homology, and extract the connected components.

The results of segmenting the six scenes can be seen in Fig. 5. For each scene, we show the filtered representation of the point cloud prior to segmentation followed by the color coded clusters extracted after segmentation. We set a threshold of 32 points as the minimum number of points to be considered a cluster. The filtration is performed for 10 steps up to the maximum distance value set for the complex. Note that under the existence of noise and occlusions in the dataset, the topological filtration process cleanly segments each object cluster. However, this method fails to segment the individually stacked objects in scene 6. Therefore, the segmentation procedure may benefit from the incorporation of additional features such color and surface normals when processing complex scenes.

To aid our understanding of the results we present the barcode diagrams for each scene in Figs. 6(a) - 6(f). In these diagrams, the x-axis represents the filtration values and the y-axis represents the 0-dimensional generators of homology. The length of the blue lines corresponds to the lifespan of the generators of homology. Shorter lines are points that die early in the filtration while longer lines are points that persist for a greater number of steps. A line with a blue triangle equates to a point with infinite persistence; the corresponding 0-dimensional generating cycle persisted past the end of the filtration and did not get filled in as a boundary after all simplices of the underlying complex had been added.

Table I displays the following information for each scene: the number of points in the point cloud after filtering, the maximum distance value chosen for the Vietoris-Rips complex construction, the size of the complex in the number of simplices, and the total CPU execution time.

VII. Conclusion and Future Work

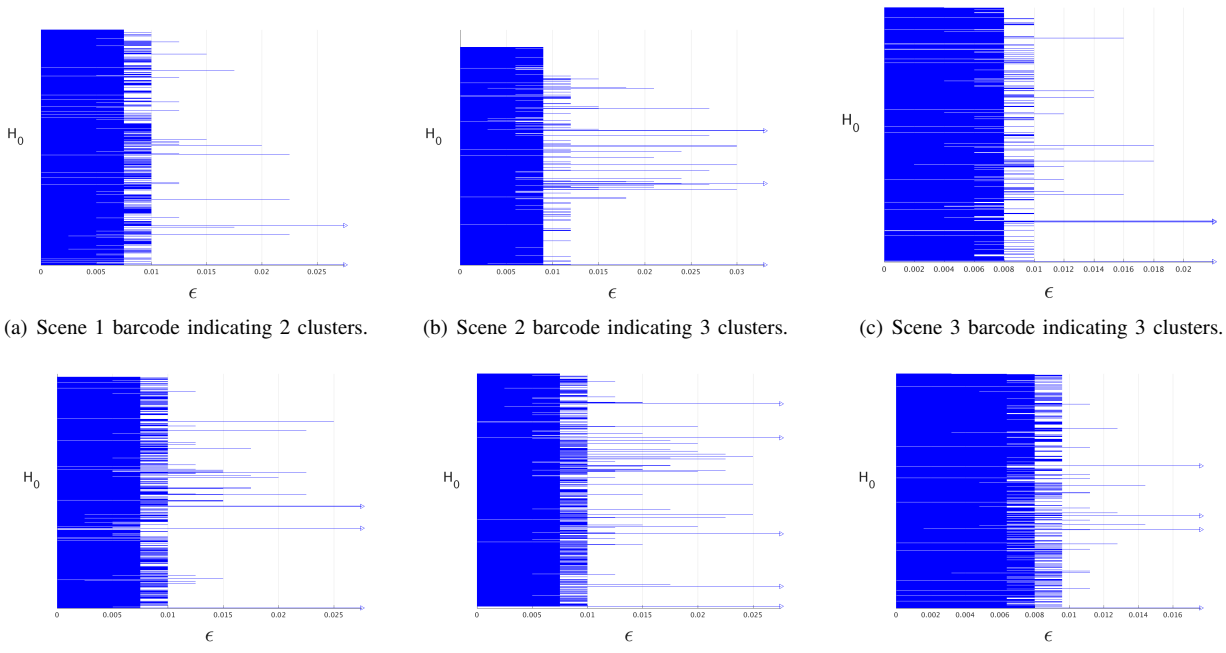
This paper presents and evaluates algorithms and data structures for segmenting 3D point clouds based on computing topological persistence. This notion of persistence derives from persistent homology, an algebraic invariant that captures topological features at varying spatial resolutions. The purpose of this work is to not only introduce a new way of performing segmentation, but also to introduce the

(a) Scene 1 filtered. (b) Scene 1 segmented. (c) Scene 2 filtered. (d) Scene 2 segmented.

(e) Scene 3 filtered. (f) Scene 3 segmented. (g) Scene 4 filtered. (h) Scene 4 segmented.

(i) Scene 5 filtered. (j) Scene 5 segmented. (k) Scene 6 filtered. (l) Scene 6 segmented.

Fig. 5. The results of 3D point cloud segmentation using topological persistence.



(a) Scene 1 barcode indicating 2 clusters. (b) Scene 2 barcode indicating 3 clusters. (c) Scene 3 barcode indicating 3 clusters.

(d) Scene 4 barcode indicating 3 clusters. (e) Scene 5 barcode indicating 5 clusters. (f) Scene 6 barcode indicating 4 clusters.

Fig. 6. The barcode diagrams for the scenes in Fig. 5. Lines with blue triangles show points of infinite persistence.

TABLE I

RESULTS OF SEGMENTATION USING TOPOLOGICAL PERSISTENCE. THE COLUMN CATEGORIES ARE AS FOLLOWS: POINTS - NUMBER OF POINTS IN THE POINT CLOUD, DISTANCE - MAXIMUM VALUE FOR THE COMPLEX CONSTRUCTION, COMPLEX - NUMBER OF SIMPLICES IN THE COMPLEX, TIME - CPU TIME IN SECONDS.

Scene	Points	Distance	Complex	Time
1	1289	0.025	4,813	2.211
2	1924	0.030	12,264	4.025
3	1410	0.020	3,744	2.344
4	1524	0.025	5,903	2.826
5	1581	0.025	5,994	3.179
6	2444	0.016	5,365	4.866

idea of topological persistence into the area of point cloud processing for 3D perception tasks.

Our experimental results show that 3D point cloud segmentation using topological persistence is an effective and robust method for finding and extracting clusters in the presence of noise. Nevertheless, much more work in this area is needed. The simplicial complex construction is the most computationally intensive step in the topological persistence segmentation pipeline. Therefore we are interested in efficient algorithms, in terms of speed and storage, for constructing simplicial complexes that scale with the size and dimensionality of the dataset [27]. We are also investigating the use of simplicial complexes that are better suited for processing voxelized datasets such as cubical complexes [28]. In addition, we are working on speeding up the filtration process through parallelization by making use of multiple CPU/GPU cores.

For future work, we would like to combine the global information provided by topology with local information, such as color and surface normals, to perform region-based segmentation of objects in point clouds. Exploring the use of higher dimensional homology groups for segmenting groupings of voxels (supervoxels) is on our agenda. Lastly, we intend to make available an open source release of our software to the point cloud processing community for use in 3D perception research.

ACKNOWLEDGEMENTS

This material is based upon work supported by the National Science Foundation through grants #IIP-0934327, #IIS-1017344, #IIP-1332133, #IIS-1427014, #IIP-1432957, #OISE-1551059, #CNS-1514626, #CNS-1531330, and #CNS-1544887.

REFERENCES

[1] Kinect, 2016. [Online]. Available: http://www.xbox.com/en-US/xbox-one/accessories/kinect-for-xbox-one

[2] L. Cruz, D. Lucio, and L. Velho, "Kinect and RGBD images: Challenges and applications," in *25th SIBGRAPI Conference on Graphics, Patterns and Images Tutorials (SIBGRAPI-T)*, 2012, pp. 36–49.

[3] R. B. Rusu, N. Blodow, Z. C. Marton, and M. Beetz, "Close-range scene segmentation and reconstruction of 3D point cloud maps for mobile manipulation in domestic environments," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2009, pp. 1–6.

[4] J. Strom, A. Richardson, and E. Olson, "Graph-based segmentation for colored 3D laser point clouds," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2010, pp. 2131–2136.

[5] D. Holz, S. Holzer, R. B. Rusu, and S. Behnke, "Real-time plane segmentation using RGB-D cameras," in *RoboCup 2011: Robot Soccer World Cup XV*. Springer, 2012, pp. 306–317.

[6] H. Edelsbrunner, D. Letscher, and A. Zomorodian, "Topological persistence and simplification," *Discrete and Computational Geometry*, vol. 28, no. 4, pp. 511–533, 2002.

[7] A. Zomorodian and G. Carlsson, "Computing persistent homology," *Discrete & Computational Geometry*, vol. 33, no. 2, pp. 249–274, 2005.

[8] A. Shamir, "Segmentation and shape extraction of 3D boundary meshes," *Proceedings Eurographics State-of-the-Art Report*, vol. 2006, pp. 137–49, 2006.

[9] A. P. Mangan and R. T. Whitaker, "Partitioning 3D surface meshes using watershed segmentation," *IEEE Transactions on Visualization and Computer Graphics*, vol. 5, no. 4, pp. 308–321, 1999.

[10] S. Shlafman, A. Tal, and S. Katz, "Metamorphosis of polyhedral surfaces using decomposition," in *Computer Graphics Forum*, vol. 21, no. 3. Wiley Online Library, 2002, pp. 219–228.

[11] M. Garland, A. Willmott, and P. S. Heckbert, "Hierarchical face clustering on polygonal surfaces," in *Proceedings of the 2001 Symposium on Interactive 3D graphics*. ACM, 2001, pp. 49–58.

[12] R. Liu and H. Zhang, "Segmentation of 3D meshes through spectral clustering," in *Proceedings of the 12th Pacific Conference on Computer Graphics and Applications*, 2004, pp. 298–305.

[13] A. Golovinskiy and T. Funkhouser, "Min-cut based segmentation of point clouds," in *IEEE 12th International Conference on Computer Vision Workshops (ICCV Workshops)*, 2009, pp. 39–46.

[14] M. Johnson-Roberson, J. Bohg, M. Björkman, and D. Kragic, "Attention-based active 3D point cloud segmentation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2010, pp. 1165–1170.

[15] B. Douillard, J. Underwood, N. Kuntz, V. Vlaskine, A. Quadros, P. Morton, and A. Frenkel, "On the segmentation of 3D lidar point clouds," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2011, pp. 2798–2805.

[16] A. Richtsfeld, T. Mörwald, J. Prankl, M. Zillich, and M. Vincze, "Segmentation of unknown objects in indoor environments," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2012, pp. 4791–4796.

[17] A. K. Mishra, A. Shrivastava, and Y. Aloimonos, "Segmenting "simple" objects using RGB-D," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2012, pp. 4406–4413.

[18] A. J. Trevor, S. Gedikli, R. B. Rusu, and H. I. Christensen, "Efficient organized point cloud segmentation with connected components," *Semantic Perception Mapping and Exploration (SPME)*, 2013.

[19] S. Paris and F. Durand, "A topological approach to hierarchical segmentation using mean shift," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007, pp. 1–8.

[20] D. Letscher and J. Fritts, "Image segmentation using topological persistence," in *Computer Analysis of Images and Patterns*. Springer, 2007, pp. 587–595.

[21] C. Chen, D. Freedman, and C. H. Lampert, "Enforcing topological constraints in random field image segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011, pp. 2089–2096.

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

[23] D. Cohen-Steiner, H. Edelsbrunner, and J. Harer, "Stability of persistence diagrams," *Discrete & Computational Geometry*, vol. 37, no. 1, pp. 103–120, 2007.

[24] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to algorithms, third edition*, 3rd ed. The MIT Press, 2009.

[25] A. Richtsfeld, "The object segmentation database (OSD)," 2012. [Online]. Available: http://www.acin.tuwien.ac.at/?id=289

[26] R. B. Rusu and S. Cousins, "3D is here: Point cloud library (PCL)," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2011, pp. 1–4.

[27] A. Zomorodian, "Fast construction of the Vietoris-Rips complex," *Computers & Graphics*, vol. 34, no. 3, pp. 263–271, 2010.

[28] H. Wagner, C. Chen, and E. Vuçini, "Efficient computation of persistent homology for cubical data," in *Topological Methods in Data Analysis and Visualization II*. Springer, 2012, pp. 91–106.