3D Region Segmentation Using Topological Persistence

William J. Beksi and Nikolaos Papanikolopoulos

Abstract—A ‘region’ is an important concept in interpreting 3D point cloud data since regions may correspond to objects in a scene. To correctly interpret 3D point cloud data, we need to partition the dataset into regions that correspond to objects or parts of an object. In this paper, we present a region growing approach that combines global (topological) and local (color, surface normal) information to segment 3D point cloud data. Using ideas from persistent homology theory, our algorithm grows a simplicial complex representation of the point cloud dataset. At each step in the growth process we compute the zeroth homology group of the complex, which corresponds to the number of connected components, and use color and surface normal statistics to build regions. Lastly, we extract out the segmented regions of the dataset. We show that this method provides a stable segmentation of point cloud data in the presence of noise and poorly sampled data, thus providing advantages over contemporary region-based segmentation techniques.

I. INTRODUCTION

The act of segmentation divides a dataset into perceptually meaningful clusters. It is an integral operation in robot vision systems. High level tasks such as the extraction of discriminative features, object recognition, and scene interpretation are directly consequent to the quality of the segmented data. Due to the availability of inexpensive RGB-D sensors [1], there has been a rise in the generation of 3D point cloud datasets. Robots equipped with RGB-D sensors can simultaneously capture both color and depth images, hence providing rich datasets along with the need to develop efficient algorithms for point cloud processing.

Within a 3D point cloud, a region can be defined as a group of connected points with similar properties. The concept of a region is essential in interpreting point cloud data since regions may correspond to objects in a scene. Segmentation of regions is a crucial preprocessing step towards pattern recognition and scene understanding. For example, segmentation based on regions is used in the semantic labeling of objects for indoor scenes [2], [3]. Structure discovery, useful in the area of personal robotics, is performed from multimodal data with a region segmentation approach [4]. Region segmentation is implemented on mobile robots for playing in RoboCup [5] and for person tracking [6].

Region segmentation algorithms are based on the expansion of a region whenever its interior is homogeneous according to certain features such as intensity, color, or texture. Unlike edge-based segmentation, which returns boundaries between regions, region-based segmentation is a method that allows the determination of regions directly. Region-based techniques are better than edge-based techniques in noisy data where edges are difficult to detect.

Region growing is one of the simplest and most popular algorithms for region-based segmentation. Traditional implementations start by choosing a starting point called a seed. Then, the region is grown by adding similar neighboring points according to a homogeneity criterion, thus increasing the size of the region step by step. The homogeneity criterion has the function of determining whether a point belongs to the growing region or not. The decision of merging is generally taken based only on the difference between the evaluated point and the region. However, it is not easy to decide when this variance is small (or large) enough to make a decision.

Defining a predicate for a homogeneity criterion which controls the region growing process involves the interplay between local and global considerations. Neighbors are joined together by a local or regional decision process, however what is desired is a satisfactory global or overall result when the algorithm terminates. In general, this dilemma can be resolved by using as much global information, as is available, to make local decisions.

In this work, we present a new 3D region segmentation technique utilizing persistent homology that builds upon our previous results [7]. Persistent homology is based on the concept that topological features detected over a varying range of scales are more likely to represent intrinsic features of the dataset in contrast to by-products of noise, bad sampling, or a selective pick of parameters [8], [9]. Our main contribution is the novel combination of global (topological) and local (color, surface normal) information, to produce a stable region growing segmentation of a 3D point cloud. This approach is fully automated and lacks the requirement of an initial seeding. Furthermore, the final segmentation does not depend on the order in which the regions are grown or joined.

The remainder of this paper is organized as follows. Related research work is presented in Section II. Section III states the problem followed by our region growing approach to segmentation in Section IV. Experimental results are provided in Section V. The paper concludes in the last section with a discussion on current issues and future work.

II. RELATED WORK

Region-based segmentation is a classic technique in computer vision and image processing with over forty years of history [10], [11]. ‘Region growing’ and ‘split and merge’ are the two most common region-based segmentation algorithms. In the following paragraphs we summarize several domains of region segmentation research that are relevant to our work.
The notion of seeded region growing is introduced in [12]. Later works furnish algorithms that do not require an initial seeding of the data [13]. In [14], a region growing sequence is built by incrementing a maximal homogeneity threshold from a small to large value. An assessment function is then used to determine the optimal homogeneity criterion.

Segmentation of regions is often used in the processing of medical images. A framework for segmenting 3D imaging data volumes with interactively guided initial seed points is developed in [15]. Interactive region segmentation with graph cuts is presented in [16]. Another work introduces a region growing algorithm based on a model that describes homogeneity and simple shape properties [17].

The authors of [18] present a color segmentation algorithm that combines region growing and merging processes. The algorithm generates a segmentation of the image into spatially disconnected but similar in color regions. A method that simultaneously segments and models a point cloud through a minimum spanning tree ellipsoidal region growing process is presented in [19].

A statistically robust segmentation algorithm using principle components analysis (PCA) is proposed for planar and non-planar surfaces extracted from laser scanning data in [20]. Fast segmentation of range images and organized point clouds is described in [21]. The main idea, geared towards polygonal meshing and then segment the resulting mesh planes, is to approximately reconstruct the surface by means of topological persistence.

A hybrid split-and-merge segmentation method using persistent homology is provided in [22]. The algorithm first performs edge detection on the input image. Then, the image is split into regions using edge-directed topology where regions with similar features are selected and merged in order of topological persistence.

III. PROBLEM STATEMENT

Given a topological space \( X = \{x_0, \ldots, x_m\} \subset \mathbb{R}^3 \) where \( x_0, \ldots, x_m \) are the points in a point cloud captured by an RGB-D sensor, our goal is to partition \( X \) into disjoint subsets \( X_1, X_2, \ldots, X_N \) such that

\[
\bigcup_{i=1}^{N} X_i = X \tag{1}
\]

\( X_i, \quad i = 1, 2, \ldots, N \) is connected \( \tag{2} \)

\( P(X_i) = \text{TRUE} \) for \( i = 1, 2, \ldots, N \) \( \tag{3} \)

\( P(X_i \cup X_j) = \text{FALSE} \) for \( i \neq j, X_i, X_j \) adjacent, \( \tag{4} \)

where \( P \) is a logical predicate defined on a set of proximate points.

The predicate \( P \) associated with the third and fourth conditions determines what kind of properties the segmented regions should have to satisfy the homogeneity criterion. The second condition implies that regions must be connected, i.e., constructed with contiguous points. This requirement plays a significant role. It affects the central structure of the segmentation algorithm, especially the region growing process, by enforcing the constraint that points are processed according to neighbor relationships. The first condition simply states that the space is composed of the disjoint regions.

IV. REGION SEGMENTATION USING TOPOLOGICAL PERSISTENCE

We formulate a solution to the problem outlined in III through a region growing algorithm that: 1) Constructs a simplicial complex representation of the input space; 2) Uses persistent homology in combination with a homogeneity criterion to grow regions; 3) Extracts the disjoint regions of the space. The output of this algorithm is a set of segmented regions where each region corresponds to an object or part of an object in a scene.

In the proceeding subsections we describe the details of the region growing algorithm. A detailed mathematical background for this work can be found in [7]. Comprehensive treatment on the subject of algebraic topology can be found in [23].

A. Simplicial Complex Construction

1) Simplices: Let \( X \) be a topological space. Our discrete representation of \( X \) is composed of simplices, Fig. 1. A d-simplex \( \sigma \) is the convex hull of \( d + 1 \) affinely independent vertices \( v_0, \ldots, v_d \in \mathbb{R}^n \). For any vertex, say \( v_i \), the \( d \) vectors \( v_j - v_i, j \neq i \), are linearly independent. This means that 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. A face \( \tau \) of \( \sigma \) is the convex hull formed by the subset \( \{v_0, \ldots, v_d\} \) of the \( d + 1 \) vertices. For example, the face of a triangle corresponds to its three edges.

![Fig. 1. A vertex, edge, and triangle are simplices of dimension 0, 1, and 2 respectively.](image)

2) Simplicial Complexes: We form simplicial complexes by joining together simplices, Fig. 2. A simplicial complex \( K \) is a collection of simplices such that if \( \sigma \in K \) and \( \tau \) is a face of \( \sigma \), then \( \tau \in K \). Furthermore, if \( \sigma, \sigma' \in K \) then \( \sigma \cap \sigma' \) is either empty or a face of both \( \sigma \) and \( \sigma' \).

![Fig. 2. A simplicial complex is constructed by gluing simplices together.](image)

Given a set of vertices \( V = \{v_0, \ldots, v_d\} \), an abstract simplicial complex, is a collection \( K \) of simplices closed under the operation of taking subsets. In other words, if \( \sigma \subset V \) is a simplex (\( \sigma \in K \)) and \( \tau \) is a face of \( \sigma \) (\( \tau \subset \sigma \subset V \)), then \( \tau \in K \).
3) Vietoris-Rips Complex: For a set of vertices $V \subset \mathbb{R}^n$ and a fixed radius $\epsilon$, the Vietoris-Rips complex of the space $X$ is an abstract simplicial complex whose $d$-simplices correspond to unordered $(d+1)$-tuples of vertices in $X$ that are pairwise within $\epsilon$ distance of each other. We use the Vietoris-Rips complex to approximate the topology of $X$, Fig. 3.

![Fig. 3. A Vietoris-Rips complex.](image)

**B. Persistent Homology Computation**

1) Homology Groups: Homology is a topological invariant that assigns a sequence of vector spaces to a space $X$. It provides us with an algebraic means to identify the holes in a topological space. This work makes use of simplicial homology which is based on the concept of a boundary homomorphism. The boundary homomorphism encodes how simplices are attached to their lower dimensional facets.

Specifically, for a given complex $K$ we choose an ordering for each simplex. An ordering on a $d$-simplex, $\sigma \subset V$, is a literal ordering of its vertices $[v_0, \ldots, v_d]$. After such an ordering is chosen, a boundary homomorphism is the linear mapping $\partial_d: C_d(K) \rightarrow C_{d-1}(K)$ where $C_d(K)$ is an $\mathbb{R}$-vector space whose basis consists of the oriented $d$-simplices in $K$. This mapping is produced by associating each basis element of $C_d(K)$ to the formal sum of its oriented faces of dimension $d-1$. The boundary operator, $\partial_d$, not only provides us with the information on how to construct $K$, but also allows us to reason about certain global topological features of $X$.

The $d$th homology group of $K$ is computed as $H_d(K) = \ker \partial_{d+1}/ \im \partial_d$, where $\ker \partial_d = \{a \in C_d | \partial_d(a) = 0\}$ and $\im \partial_{d+1} = \{b \in C_{d+1} | \exists a \in C_d : \partial_{d+1}(b) = a\}$. That is, $H_d(K)$ is the quotient vector space whose generators are $d$-cycles ($d$-dimensional subcomplexes surrounding a hole) modulo the equivalence relation that states two such $d$-cycles are the same whenever they are the oriented boundary of a $(d+1)$-dimensional subcomplex.

2) Filtrations: Note in the previous subsection we stated that a simplicial complex is constructed with respect to a scale parameter $\epsilon$. Nevertheless, when computing homology, topological features of this simplicial complex may be due to noise or inappropriately choosing the scale parameter. This problem is solved by persistent homology in the following way. A *filtration* is set up which tracks the growth of the complexes across the entire range of possible values of the scale parameter. Concretely, $\theta \subset K_0 \subset K_1 \subset \ldots \subset K_L = K$, where $L$ is the maximum value for constructing the complex. In essence, this allows us control by excluding short lived features, over how long a feature has to exist in the filtration before it is considered significant.

**C. Homogeneity Criterion**

Our criterion of homogeneity employs the analysis of the color characteristics of the nearest neighbors. The CIELAB color space, chosen for its perceptually uniform color distances, is used to compare the chromaticity between two neighboring points

$$\Delta E = \sqrt{(l_2 - l_1)^2 + (a_2 - a_1)^2 + (b_2 - b_1)^2}, \quad (5)$$

where $l$ corresponds to the luminance channel, $a$ and $b$ are the color channels, and $\Delta E$ is the Euclidean distance.

In addition, a similarity measure using the surface normal of a point is used. We estimate the coordinates of the surface normal by performing an eigendecomposition of the covariance matrix created from the nearest neighbors. The angle between a surface normal $u$ and its neighbor $v$ is computed as

$$\theta = \arccos \frac{u \cdot v}{\|u\|\|v\|}. \quad (6)$$

**D. Region Growing Algorithm**

The homogeneity criterion formulated in the previous subsection is used in conjunction with topological persistence to perform region growing. As regions are grown based on local similarities between nearest neighbors, the global connectedness of the region is preserved using topology. To grow regions we construct the 1-skeleton of the Vietoris-Rips complex, $K = \{\sigma \subset \{x_0, \ldots, x_m\} | \text{dist}(x_i, x_j) \leq \epsilon, \forall x_i \neq x_j \in \sigma\}$, from an input point cloud where $\text{dist}$ is the Euclidean metric and the points of $\sigma$ are pairwise within $\epsilon$ distance. To enforce the constraint that each disjoint region is a connected component we compute the zeroth homology group of the complex, Alg. 1.

The algorithm takes as input a set of points where each point is identified by a unique integer (ID). Range queries for finding the nearest neighbors of each point at a given radius are carried out using a kd-tree. For all nearest neighbors we compute the similarity measures based on (5) and (6). The indices of the nearest neighbors that satisfy the homogeneity criterion are saved. Among the indices of similar nearest neighbors we compute the parent ID of the current nearest neighbor to the nearest neighbor with the minimum ID. We proceed to find the root of the similar nearest neighbor point if the parent ID is greater. When a point has not been previously joined to any other point, we record the radius at which the points are to be connected. In the last step, we take the union of the points by making the parent ID the minimum of the pair. For the sake of clarity, Alg. 1 shows the nearest neighbor range search and feature computations. These values can be precomputed hence decreasing the runtime substantially.

**E. Region Growing Post-processing**

At the end of region growing, we may have regions that are smaller than a predefined region size. To satisfy property (1)
we post-process the remaining regions as follows. For each region representative, a nearest neighbors query is run. The point from the nearest neighboring region that best satisfies the homogeneity criterion is selected. A union operation is then performed thus joining the two regions. In practice, sufficiently small regions can be treated as noise and removed resulting in a cleaner segmentation of the data.

F. Extracting Regions

The outer loop of Alg. 1 is run for each step in the filtration. This yields an approximation of the underlying topological space at differing spatial resolutions. Initially, all points are born at time zero. During the filtration, existing homology classes can merge (die) as the complex grows. At each resolution we compute the 0-dimensional homology group ($H_0$) of the space, which corresponds to the number of disjoint regions, thus allowing us to track those homology classes that are present throughout many iterations of the filtration.

We trace the division of the point cloud dataset into disjoint regions with a union-find data structure. Initialization of the data structure is done by making each 0-simplex (point) its own set. While performing the filtration, nearby 0-simplices that meet the homogeneity criterion are united within the data structure. At the end of the filtration, we find the segmented regions based on the sets of points that are joined to 0-dimensional simplices of infinite persistence, Alg. 2.

Algorithm 1 growRegions(points)
1: tree = KDTreeSearcher(points)
2: for all points do
3:    query = [point.x point.y point.z]
4:    nn = rangesearch(tree, query, radius)
5:    for all nn do
6:      $\theta = $ computeNormalAngle()
7:      $\Delta E = $ computeColorDistance()
8:      if $\theta \in (\theta_{\text{min}}, \theta_{\text{max}})$ and
9:        $\Delta E \in (\Delta E_{\min}, \Delta E_{\max})$ then
10:       similar.nn = [similar.nn nn]
11:    end if
12: end for
13: min.nn = min(similar.nn)
14: for all similar.nn do
15:   if point(similar.nn).parent >
16:     point(min.nn).id then
17:       root = find(point(similar.nn))
18:       if point(root).id == point(similar.nn).id then
19:         point(similar.nn).death = radius
20:        end if
21:      end if
22:    union(point(similar.nn), point(min.nn))
23: end for
24: end for

Algorithm 2 computeRegionPersistence(points)
1: $K = $ growRegions(points)
2: for all points $\in K$ do
3:   Report death times
4: end for

V. EXPERIMENTAL RESULTS

In this section, we evaluate the 3D region segmentation algorithm described in Section IV. The following eight objects were used in the experiments: plant, water heater, paper roll, robot, box, shoe, shuttle, coffee can. The experimental runs were carried out using MATLAB on a 64-bit GNU/Linux machine with a single CPU core.

Prior to executing region segmentation we preprocess the data using the Point Cloud Library (PCL)\textsuperscript{3}. First, background subtraction is done to remove outlying points that are beyond range of the object to be segmented. Next, the floor is removed upon which the object is resting. To do this, we find all points that correspond to a planar model and discard them.

After preprocessing the point cloud, we run a filtration using topological persistence in combination with the homogeneity criterion to grow regions and then extract the segmented regions. The results of segmenting the eight objects can be seen in Fig. 4. For each object, we show the preprocessed representation of the point cloud before region segmentation proceeded by the randomly colored regions extracted after the segmentation has finished.

All filtrations are performed for ten steps up to the maximum distance value set for the complex. The threshold for the minimum number of points to be considered a region ranges from 8 to 128. For the homogeneity criterion, the color difference $\Delta E$ ranges from 3 to 10, and the angle difference $\theta$ between surface normals is set to $\pm 5^\circ$.

Under the existence of noise and the inherent limitations of the RGB-D sensor resolution, our region segmentation process is able to extract the major regions of each object. In the plant point cloud, the regions corresponding to the pot, soil, trunk, and leaves are captured. Note the existence of noise around each plant leaf in the original point cloud is removed in the segmented version resulting in a clean delineation of the leaves. Regions of uniform color distribution and/or surface normals such as the water heater, paper roll, and shoe are easily segmented. The robot, box, shuttle, and coffee can render much more challenging scenarios. These objects have regions that are very small, contain subtle color changes, and are disjoint when in fact they should not be. For example, in the segmented coffee can 112 regions are found. Some of these regions make up the product label, many are noise. Therefore, a trade-off in region segmentation must be made between granularity and noise.

To gain insight into the procedure of persistent homology we present the barcode diagrams for each object in Figs. 5(a)-5(h). In these diagrams the filtration values are represented on

\textsuperscript{3}http://www.pointclouds.org
Fig. 4. The results of 3D region segmentation using topological persistence.

Fig. 5. The barcode diagrams for the objects in Fig. 4. Lines with blue triangles indicate points of infinite persistence.
the x-axis, and the y-axis shows the 0-dimensional generators of homology. The lifespan of the generators of homology corresponds to the length of the blue lines. Points that die early in the filtration are denoted by shorter lines. Conversely, points that persist for a greater time interval map to longer lines. Points of infinite persistence are indicated by lines with blue triangles. This constitutes a 0-dimensional generating cycle that persisted past the end of the filtration, i.e. after all simplices of the complex had been added it did not get filled in as a boundary of a higher dimensional simplex.

Table I displays the following information for each object (plant, water heater, paper roll, robot, shoe, shuttle, coffee can): the number of points in the point cloud after filtering, the maximum distance value chosen for the Vietoris-Rips complex construction, the type of local features used, and the execution time for computing topological persistence.

### Table I

<table>
<thead>
<tr>
<th>Object</th>
<th>Points</th>
<th>Distance</th>
<th>Features</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>12,729</td>
<td>0.03</td>
<td>color+normals</td>
<td>1.508</td>
</tr>
<tr>
<td>2</td>
<td>10,395</td>
<td>0.035</td>
<td>color+normals</td>
<td>3.314</td>
</tr>
<tr>
<td>3</td>
<td>7,178</td>
<td>0.03</td>
<td>color+normals</td>
<td>1.884</td>
</tr>
<tr>
<td>4</td>
<td>25,790</td>
<td>0.02</td>
<td>color+normals</td>
<td>6.620</td>
</tr>
<tr>
<td>5</td>
<td>10,511</td>
<td>0.03</td>
<td>color</td>
<td>5.432</td>
</tr>
<tr>
<td>6</td>
<td>9,268</td>
<td>0.03</td>
<td>color</td>
<td>3.594</td>
</tr>
<tr>
<td>7</td>
<td>12,924</td>
<td>0.02</td>
<td>color</td>
<td>1.404</td>
</tr>
<tr>
<td>8</td>
<td>8,376</td>
<td>0.028</td>
<td>color</td>
<td>1.541</td>
</tr>
</tbody>
</table>

VI. CONCLUSION AND FUTURE WORK

This paper presents a new algorithm that uses persistent homology to combine global and local knowledge for segmenting regions in 3D point clouds. Persistent homology allows us to track topological features, e.g. the number of connected components, at varying spatial scales. Regions in point cloud data often match fundamental parts of an object or entirely complete objects. Extraction of regions is an imperative action in high-level robot vision tasks such as scene understanding.

Experimentally, our results show that a region growing approach using topological persistence is a capable method for identifying regions in noisy point cloud data. Construction of the simplicial complex is computationally intensive. Therefore, we are keen on parallelizing core parts of the algorithm using multiple CPU/GPU cores. Additionally, we are working on extending these ideas to a restricted form of region segmentation, i.e. supervoxel segmentation.

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.