Point Cloud Segmentation Exercise for the VCG Interview

Prerequisites

Download the example.pts file provided along with exercise. This is a plain text file which contains a **boundary-annotated** point cloud of a relatively simple CAD model (10 thousand points). By boundaries we mean points that lie on sharp edges of the original CAD model or areas of high concavity/convexity. Each row represents a unique point of the point cloud with the following format:

- x y z nx ny nz boundary_label boundary_probability
    - **x y z (float)**: 3D coordinates of the point
    - **nx ny nz (float)**: the normal of the point
    - **boundary_label (int)**: ground truth label - 1 if the point lies on a boundary, 0 if the point does not lie on a boundary
    - **boundary_probability (float)**: probability for the point to lie on a boundary, as returned by a deep neural network

Recommended software tools for inspecting the given point cloud:

- CloudCompare
- MeshLab

Introduction

Segmentation is one of the most important tasks in the analysis of 3D shapes. The goal of this task is to divide the surface of a 3D shape into homogeneous regions. There are several types of segmentation, for example, semantic segmentation: segment a shape into semantic parts, or, geometric segmentation: decompose the shape into regions enclosed by sharp boundaries. In this exercise we focus on geometric segmentation. For this task, a representation of the 3D shape is taken as input. There are many possible representations for a 3D shape, e.g. a 3D mesh, a 3D point cloud, a NURBS surface etc. In this exercise we focus on a point cloud. The goal of the task is to provide a label as output for each input point. This label is just an integer corresponding to either a region ID. By using a simple watershed approach, and a normalized cuts approach, you are asked to decompose the provided 3D point cloud into geometric segments. We expect to see code and results for the provided example.pts file for both of the methods described below. Please answer the questions in each of the two methods as well.

1. Watershed Segmentation

Watershed segmentation is a simple morphological method based on region processing, mainly applied on image segmentation of medical images. It can be applied in the same way for 3D point clouds if there is some notion of connectivity between neighboring points. In our case, we want to use the thresholded boundary points as seen in Figure 1 (left) as a guide to segment our point cloud. The procedure is relatively simple. First, you will have to threshold the boundary

probabilities to infer boundaries that match the given ground-truth boundaries i.e. maximize the F1-score[1] between ground-truth and inferred boundaries.
Question: What is the threshold you chose?
Question: What is the F1-score you get?
Then you need to construct a k-nn[2] graph over the point cloud and perform a BFS (Breadth-First Search) starting from a random seed point and stopping at inferred thresholded boundary points. All visited points result in a segment. Then, repeat the same procedure by selecting a random seed point from the rest of the non-visited points, until no points are left unvisited. Save and show your results. They should look similar to Figure 1 (right).
Question: What value did you choose for k and why?
Question: How many segments do you get ?

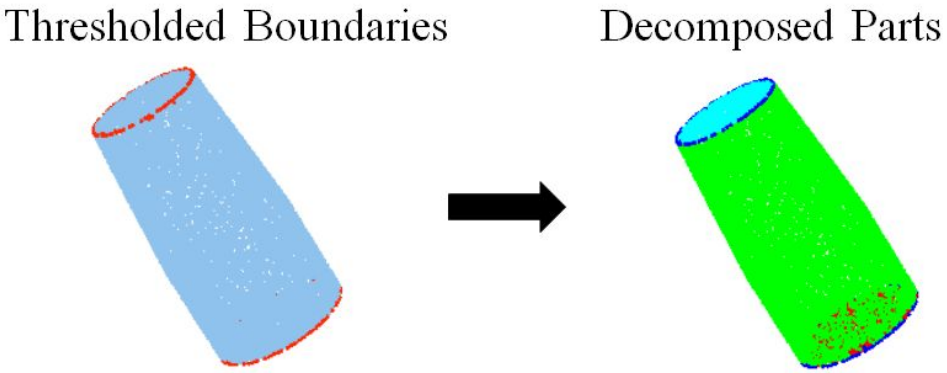

Figure 1

2. Normalized Cuts Segmentation

By using Normalized Cuts[3] we can split a weighted graph into disjoint sets, i.e. unlabelled segments, whereby some measure of the similarity within a group is high and that across the group is low[4]. First you will have to construct a k-nn graph over the point cloud and assign a weight to each edge of the graph. To assign a weight to each edge there are several options. In this task you should use a function of: a) the normal angle between the two adjacent points (edge's endpoints), and b) the max-pooled boundary probabilities of the adjacent points. Finally, you need to use Normalized Cuts to partition the graph, i.e. the point cloud into individual sets of points (regions). *Hint: You can use Spectral Clustering since Normalized Cuts is a special case of Spectral Clustering e.g. check scikit-learn for this.*
Question: Which of the two options for the edge weight works best?
Question: What function did you use for each of the two options?
Question: Can you think of a better option for an edge weight that gives better results?

Save and show your results as in watershed segmentation above.

---

[1] https://en.wikipedia.org/wiki/F1_score
[2] The choice of k is yours to decide.
[3] See https://people.eecs.berkeley.edu/~malik/papers/SM-ncut.pdf for the original paper
[4] See http://www.sci.utah.edu/~gerig/CS7960-S2010/handouts/Normalized%20Graph%20cuts.pdf