Clustering Segmentation for VTK

Release 0.00

David Doria

August 2, 2011

Rensselaer Polytechnic Institute

Abstract

This document presents a VTK implementation of the algorithm described in "A clustering method for efficient segmentation of 3D laser data" by Klasing, Klaas Wollherr, Dirk, and Buss, Martin. The algorithm .

The code is available here: `https://github.com/daviddoria/ClusteringSegmentation`

Latest version available at the Insight Journal [`http://hdl.handle.net/10380/3309`]

Contents

1 Introduction

This document presents a VTK implementation of an algorithm to find clusters in a point cloud. It implicitly uses a Radially Bounded Nearest Neighbor (RBNN) graph. This implementation is based on [1].

2 Algorithm

- Iterate over all points. For each point:
  - If the point already belongs to a cluster, skip it.
  - Find all neighbors within distance r.
  - If any of these neighbors is already in a cluster, assign the current point to the same cluster, then assign all neighbors without a cluster to the same cluster.
  - If the current point has been assigned to a cluster and there exist neighbors assigned to different clusters, merge all these clusters.

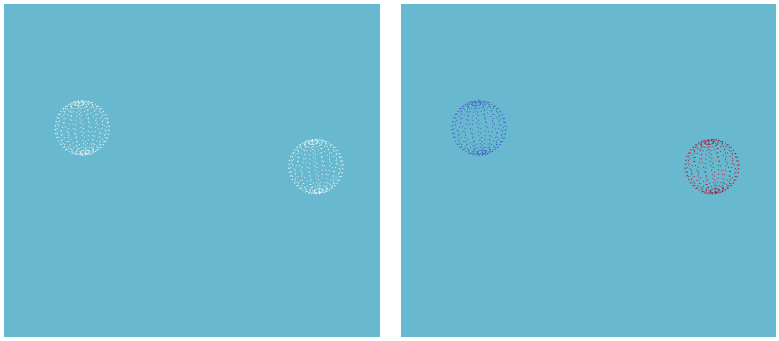
In this implementation, the cluster that each point belongs to is stored in a vtkPointData array of the output vtkPolyData called "ClusterID".

3 Parameters

The only parameter to the algorithm is the radius of the nearest neighbor lookup. We provide a flag "UseAutoRadius" that attempts to select a reasonable radius for the nearest neighbor lookup based on the extent of the data.

4 Demonstration

A demonstration of the algorithm is shown in Figure 1.



(a) Input point cloud. (b) Output point cloud.

Figure 1: An example clustering.

5 Code Snippet

The interface to the filter is very straight forward. Simply pass it a vtkPolyData, and it will return a vtkPolyData with a new field "ClusterID" attached to each point.

```
vtkSmartPointer<vtkXMLPolyDataReader> reader =
  vtkSmartPointer<vtkXMLPolyDataReader>::New();
reader->SetFileName(inputFileName.c_str());
reader->Update();

vtkSmartPointer<vtkClusteringSegmentation> clusteringSegmentation =
  vtkSmartPointer<vtkClusteringSegmentation>::New();
clusteringSegmentation->SetInputConnection(reader->GetOutputPort());
clusteringSegmentation->SetUseAutoRadius(true);
clusteringSegmentation->Update();

vtkSmartPointer<vtkXMLPolyDataWriter> writer =
  vtkSmartPointer<vtkXMLPolyDataWriter>::New();
writer->SetFileName(outputFileName.c_str());
writer->SetInputConnection(clusteringSegmentation->GetOutputPort());
writer->Write();
```

References

[1] Klasing, Klaas Wollherr, Dirk, and Buss, Martin, *A clustering method for efficient segmentation of 3D laser data*. 2008 IEEE International Conference on Robotics and Automation 1