
Hough Transform Plane Detector

Release 0.00

Dorit Borrmann and David Doria

May 31, 2011

Abstract

This document presents a wrapper of an extracted portion of 3DTK (<http://threedtk.de>) to enable a developer to find planes in 3D point cloud data.

The code is available here: <https://github.com/daviddoria/VTKHoughPlanes>

Latest version available at the [Insight Journal](http://hdl.handle.net/10380/3250) [<http://hdl.handle.net/10380/3250>]
Distributed under [Creative Commons Attribution License](#)

Contents

1	Introduction	1
2	Demonstration	1
	Code Snippet	2

1 Introduction

Finding planes in 3D point clouds is a very common operation. This code uses the Hough transform to find the strongest planes in a point cloud and then labels each point with the label of the plane to which it belongs.

2 Demonstration

Figure 1 demonstrates the algorithm. Figure 1(a) shows a LiDAR scan of a flat panel monitor sitting on a counter. Figure 1(b) shows the points colored by the plane to which they belong.

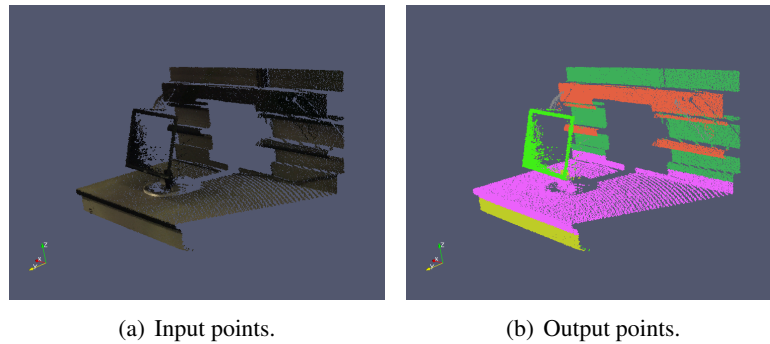


Figure 1: Demonstration

Code Snippet

The `vtkHoughPlanes` class must be passed an input `vtkPolyData` via `SetInputConnection`. The many parameters can also be set as demonstrated below.

```
// Read the input file
vtkSmartPointer<vtkXMLPolyDataReader> reader =
    vtkSmartPointer<vtkXMLPolyDataReader>::New();
reader->SetFileName(inputFileName.c_str());
reader->Update();

vtkSmartPointer<vtkHoughPlanes> houghPlanes =
    vtkSmartPointer<vtkHoughPlanes>::New();
houghPlanes->SetInputConnection(reader->GetOutputPort());
houghPlanes->SetMaxDist(2.00);
houghPlanes->SetMinDist(0.10);
houghPlanes->SetAccumulatorMax(100);
houghPlanes->SetMinSizeAllPoints(5);
houghPlanes->SetRhoNum(100);
houghPlanes->SetThetaNum(360);
houghPlanes->SetPhiNum(176);
houghPlanes->SetRhoMax(5.00);
houghPlanes->SetMaxPointPlaneDist(0.050);
houghPlanes->SetMaxPlanes(30);
houghPlanes->SetMinPlaneSize(100);
houghPlanes->SetMinPlanarity(0.300);
houghPlanes->SetPlaneRatio(0.5);
houghPlanes->SetPointDist(0.050);
houghPlanes->SetPeakWindow(false);
houghPlanes->SetWindowSize(8);
houghPlanes->SetTrashMax(20);
houghPlanes->SetAccumulatorType(1);
houghPlanes->SetHoughAlgorithm(vtkHoughPlanes::Randomized);
houghPlanes->Update();
```

```
// Write output points colored by plane
vtkSmartPointer<vtkXMLPolyDataWriter> writer = vtkSmartPointer<vtkXMLPolyDataWriter>::New();
writer->SetInputConnection(houghPlanes->GetOutputPort());
writer->SetFileName(outputFileName.c_str());
writer->Write();
```