Hough Transform Plane Detector

Release 0.00

Dorit Borrmann and David Doria

June 27, 2011

Abstract

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

The code is available at:
http://sourceforge.net/projects/slam6d/files/3dtk-houghplanes-for-vtk-release-1.0.tgz/download

Contents

1 Introduction **1**

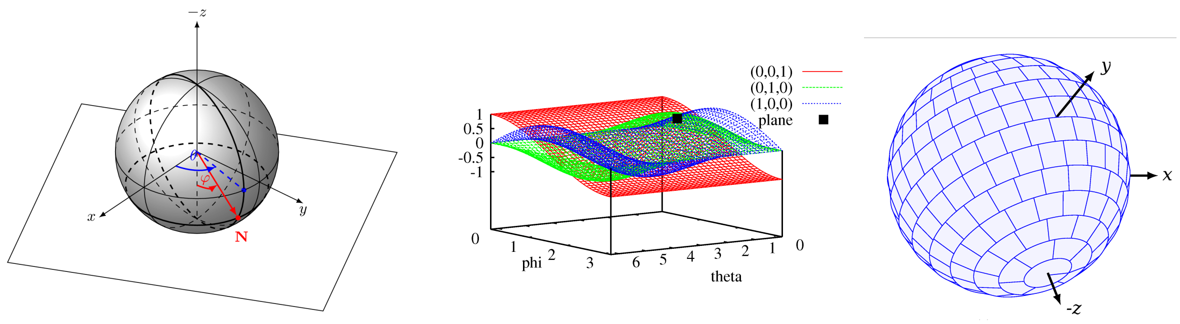
2 The Hough Transform **2**
2.1 Hough Methods 2

3 Demonstration **3**
3.1 Code Snippet 3
3.2 Explanation of parameters 4

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. The methods implemented in 3DTK are explained in detail in [1]. A very brief description is also given in the following section. For a more detailed view please refer to the original article.

2 The Hough Transform



(a) Polar coordinates of normal vector (b) Hough Transform of 3 points (c) Accumulator

Figure 1: The Hough Transform.

The Hough Transform [3, 2] is a method for detecting parametrized objects. In this software package it is used to detect planes in 3D point clouds. A plane is typically represented by its normal vector **n** and the distance ρ from the origin. Let θ be the angle of the normal vector on the *xy*-plane and φ the angle between the *xy*-plane and the normal vector in *z* direction (cf. Figure 1(a)). Then the plane is given by

$$\rho = \mathbf{p} \cdot \mathbf{n} = p_x \cdot \cos\theta \cdot \sin\varphi + p_y \cdot \sin\varphi \cdot \sin\theta + p_z \cdot \cos\varphi$$

for every point **p** on the plane. φ, θ and ρ define the 3-dimensional Hough Space (θ, φ, ρ). Each point in the Hough Space corresponds to one *plane* in $\mathbb{R}^3$. The Hough Transform of a point in $\mathbb{R}^3$ is a 3D sinusoid curve in Hough Space as shown in Figure 1(b). The point in Hough Space where the curves for three points intersect corresponds to the plane spanned by these three points. The more curves intersect in one point, the better this plane is represented by the point set.

To find planes in the point set, the Hough Space is discretized into a so-called accumulator. Different ways of discretizing the Hough Space are presented in [1]. One way is depicted in Figure 1(c). Transforming a point **p** into Hough Space means increasing the counter for all cells in the accumulator that represent a plane through **p**. The cell with the highest counter corresponds to the plane that is best represented by the point set.

2.1 Hough Methods

Applying the Hough Transform to a large point set with a fine discretization of the Hough Space is computationally expensive. To overcome this issue there are different variations of the Hough Transform.

Standardized Hough Transform (SHT) For all points from the input point set the complete Hough Transform is performed. Therefore this method is deterministic. After the voting phase the cells with the maximum votes are considered as plane candidates.

Randomized Hough Transform (RHT) Instead of performing the complete Hough Transform for the points triples of points are randomly selected and only the counter for the cell corresponding to the plane spanned by these points is incremented. Whenever one counter exceeds a threshold the points on the plane represented by this cell are removed from the point set. This procedure typically leads to more stable results.

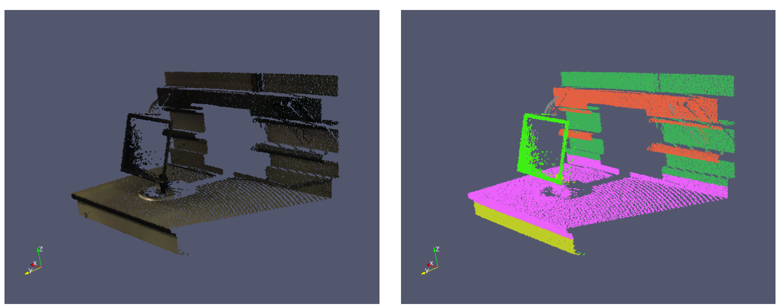
Probabilistic Hough Transform (PHT) The complete Hough Transform is only performed for p percent of the points from the input set. p has to be determined based on the expected noise level of the point set, e.g., the percentage of points that belong to no plane.

Progressive Probabilistic Hough Transform (PPHT) The complete Hough Transform is performed for randomly selected points from the input set. A threshold is calculated based on the number of points that have already voted. Once the counter of a cell exceeds this threshold the voting phase is stopped and the points that lie on the corresponding plane are deleted. This algorithm is less sensitive to noise in the data.

Adaptive Probabilistic Hough Transform (APHT) Sets of 10 points are randomly selected from the input point set. For these points the complete Hough Transform is performed. A list of maxima is maintained and updated after each voting round. Based on the stability of this list plane candidates are selected.

3 Demonstration

Figure 2 demonstrates the algorithm. Figure 2(a) shows a LiDAR scan of a flat panel monitor sitting on a counter. Figure 2(b) shows the points colored by the plane to which they belong.



(a) Input points. (b) Output points.

Figure 2: Demonstration

3.1 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();
```

3.2 Explanation of parameters

The Hough Transform needs many parameters to fine tune the application. The implementation does not depend on any unit of measurement. However, it is indispensible that the units of the parameters match with the units used for the input data.

General parameters:

`AccumulatorType` The software offers different accumulator types:

1: the Accumulator Array with a linear discretization of the Hough Space,

2: the Accumulator Ball with similar cell sizes when the Hough Space is mapped onto the unit sphere,

3: the Accumulator Cube where the discretization is based on projecting a cube onto the unit sphere,

4: the improved Accumulator Ball, where each of the two poles is represented by a single cell.

`HoughAlgorithm` The implemented Hough variations are are SHT, RHT, PHT, PPHT, and APHT.

The different Hough methods have various stopping rules. The following parameters are used to define the thresholds for these stopping rules:

`MinSizeAllPoints` Defines a threshold when to stop applying the Hough Transform. The number indicates the percentage of points from the input point cloud that have not been asigned to a plane.

`MaxPlanes` Defines a threshold for the maximum number of planes that are detected. The algorithms stops after this amount of planes has been reached.

`TrashMax` Threshold that defines after how many discarded planes the algorithm is stopped. If one cell in the accumulator receives enough votes, a plane is fitted through all points that belong to the represented plane. Planes are discarded if they are only represented by a small number of points, or if the planarity of the sample points is too low.

For practical applications the Hough Space is discretized. The following parameters define the discretization:

`RhoMax` Defines the maximum distance ρ_{max} of a plane from the origin for discretization.

`RhoNum` Defines the number of cells ρ_N in direction of ρ. Each cell has the size ρ_{max}/ρ_N.

`ThetaNum` Defines the number of cells θ_N in direction of θ.

`PhiNum` Defines the number of cells φ_N in direction of φ. Each cell has the size φ_N/φ_N.

Plane parameters:

`MaxPointPlaneDist` Maximal distance between a point and the plane so that the point is still considered to belong to the plane. This parameter is used to compensate measurement noise and discretization.

`MinPlanarity` A detected plane is discarded if the planarity falls below this threshold. Each plane candidate is fitted through all points that are considered to belong to it using a least squares method based on the Jacobi eigenvalue algorithm. The planarity is given by the value of the smallest eigenvalue devided by the number of points.

Additional parameters for certain algorithms:

`PlaneRatio` For the SHT and the PHT the maxima in the accumulator are determined after the accumulation phase is completed. The `PlaneRatio` is a threshold r that determines which maxima from the accumulator are considered to be detected planes. Each cell in the accumulator with a count lower than $r \cdot max_c$ is discarded, where max_c is highest score in the accumulator

`PeakWindow` For the SHT and the PHT the maxima in the accumulator are determined after the accumulation phase is completed. If `PeakWindow` is set to false the raw maxima in the accumulator are used, otherwise a sliding window technique is used to filter out maxima in the accumulator that are close to each other.

`WindowSize`	Sets the side lenght of the sliding window for peak detection.
`MinDist`	Sets the minimum distance for the three points selected to calculate the model plane for the Randomized Hough Transform.
`AccumulatorMax`	Sets the maximum distance for the three points selected to calculate the model plane for the Randomized Hough Transform.

References

[1] Dorit Borrmann, Jan Elseberg, Kai Lingemann, and Andreas Nüchter. The 3D Hough Transform for Plane Detection in Point Clouds: A Review and A new Accumulator Design. *3D Research*, 2:1–13, 2011. 10.1007/3DRes.02(2011)3. 1, 2

[2] R. O. Duda and P. E. Hart. Use of the Hough Transformation to Detect Lines and Curves in Pictures. Technical Note 36, Artificial Intelligence Center, SRI International, 1971. 2

[3] Paul V. C. Hough. Method and Means for Recognizing Complex Patterns. US Patent 3069654, December 1962. 2