

*Report*

PLANAR DETECTION

2021/01/07 Version: 1.0

Brycen Co., Ltd

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Revised parts and contents | Writer |
| 1.0 | Jan 07, 2021 | First version | Do Ngoc Hoai, Duong Bao Quy |

Table of Contents

[1. Functions 6](#_Toc60929516)

[2. Flow diagrams and program functions 6](#_Toc60929517)

[2.1 Flow diagram of the program 6](#_Toc60929518)

[2.2 Explain the function blocks 7](#_Toc60929519)

[3. Results 9](#_Toc60929520)

Table of Figure

[Figure 1 Flow diagram of the program 7](#_Toc60930179)

[Figure 2 PCL Reader Function 7](#_Toc60930180)

[Figure 3 Filtering object Function 8](#_Toc60930181)

[Figure 4 Segmentation Function 8](#_Toc60930182)

[Figure 5 Extract Fuction 9](#_Toc60930183)

[Figure 6 Get plane equation 9](#_Toc60930184)

[Figure 7 Result on the terminal screen 9](#_Toc60930185)

[Figure 8 Original Point Cloud Data 10](#_Toc60930186)

[Figure 9 Detect plane 1 10](#_Toc60930187)

[Figure 10 Detect plane 2 11](#_Toc60930188)

[Figure 11 Detect plane 3 11](#_Toc60930189)

Introduction

This document describes the program and functions in the Planar Detection app.

1) Functions used in the program

2) Flow diagram of the program's operation flow

3) Results detected

# 1. Functions

|  |  |  |
| --- | --- | --- |
| No. | Function | Description |
| 1 | pcl::PCDReader | Point Cloud Data (PCD) file format reader. |
| 2 | pcl::VoxelGrid | Create a filter to read the input file. |
| 3 | pcl::fromPCLPointCloud2 | Convert to the temlated Point Cloud. |
| 4 | pcl::PCDWriter | Point Cloud Data (PCD) file format writer |
| 5 | pcl::ModelCoefficients | Get the model coefficient. |
| 6 | pcl::PointIndices | Data Detected. |
| 7 | pcl::SACSegmentation | Create the segmention object. |
| 8 | pcl::ExtractIndices | Extract data detected. |

Table 1 Functions

# 2. Flow diagrams and program functions

## 2.1 Flow diagram of the program

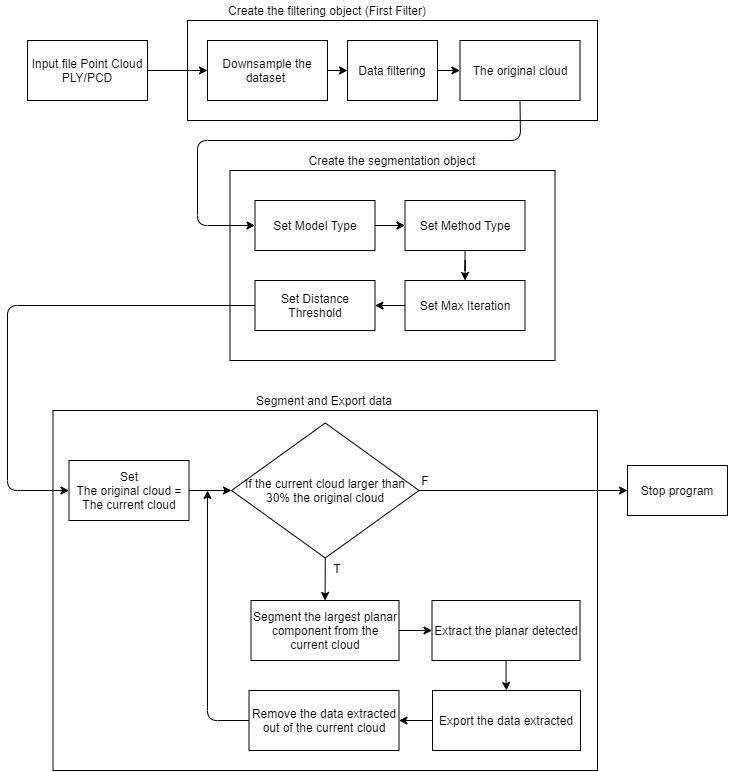


Figure 1 Flow diagram of the program

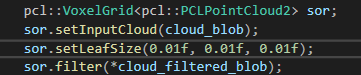
## 2.2 Explain the function blocks

The following lines of code will read the point cloud data from disk.



Figure 2 PCL Reader Function

Then, a *pcl::VoxelGrid* filter is created with a leaf size of 1cm, the input data is passed, and the output is computed and stored in *cloud\_filtered\_blob*.



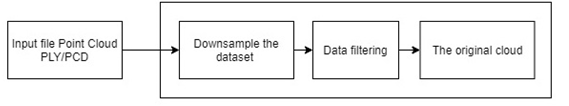
**

Figure 3 Filtering object Function

Then, lines

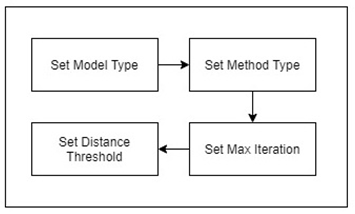
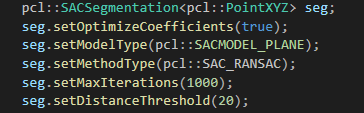


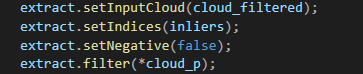
Figure 4 Segmentation Function

Create the: *pcl::SACSegmentation <pcl::SACSegmentation>* object and set the model, method type and the interation. This is also where we specify the “distance threshold”, which determines how close a point must be to the model in order to be considered an inlier. In this tutorial, we will use the RANSAC method *pcl::SAC\_RANSAC* as the robust estimator of choice.

The line:



And:



Represent the actual indices: *pcl::extraction filter <pcl::ExtractIndices>*. To process multiple models, we run the process in a loop, and after each model is extracted, we go back to obtain the remaining points, and iterate. The inliers are obtained from the segmentation process, as follows:



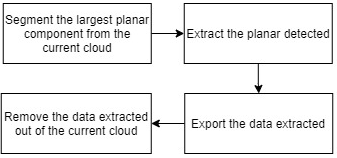


Figure 5 Extract Fuction

Finaly

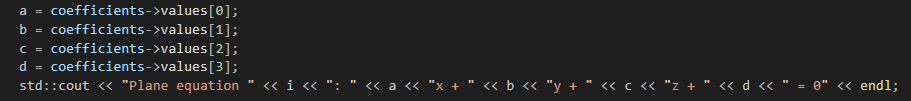


Figure 6 Get plane equation

Are used to show the contents of the inlier set, together with the estimated plane parameters (in ax + by + cz + d = 0 form).

Results on the terminal screen

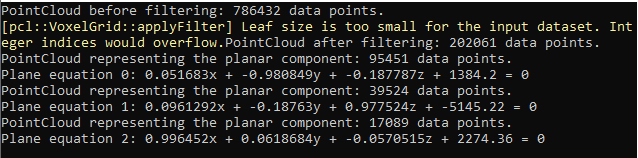


Figure 7 Result on the terminal screen

# 3. Results

Original data: PointCloud before filtering: 786432 data points.

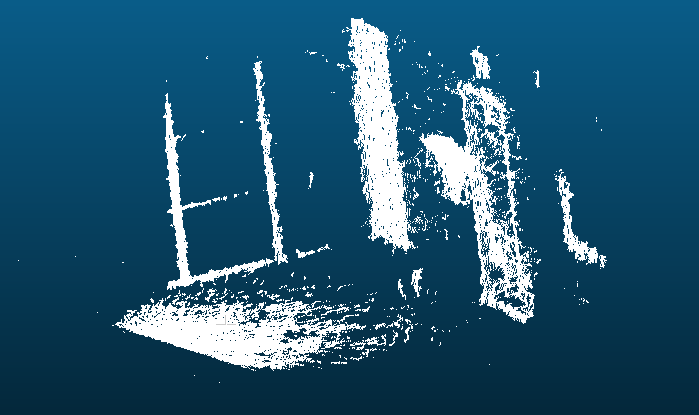


Figure 8 Original Point Cloud Data

The detected planes were detached and saved to files.

PointCloud representing the planar component: 95451 data points.

Plane equation 0: 0.051683x + -0.980849y + -0.187787z + 1384.2 = 0

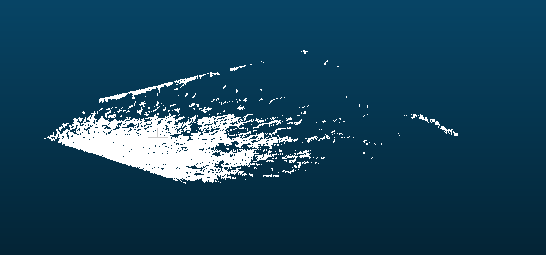


Figure 9 Detect plane 1

PointCloud representing the planar component: 39524 data points.

Plane equation 1: 0.0961292x + -0.18763y + 0.977524z + -5145.22 = 0.

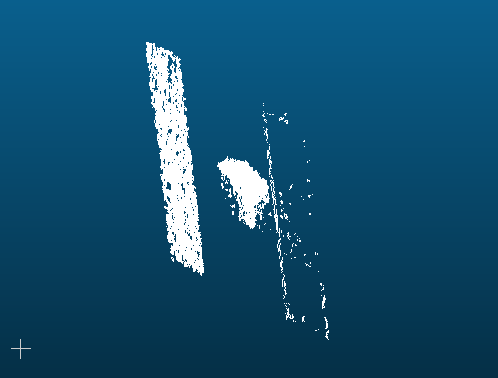


Figure 10 Detect plane 2

PointCloud representing the planar component: 17089 data points.

Plane equation 2: 0.996452x + 0.0618684y + -0.0570515z + 2274.36 = 0

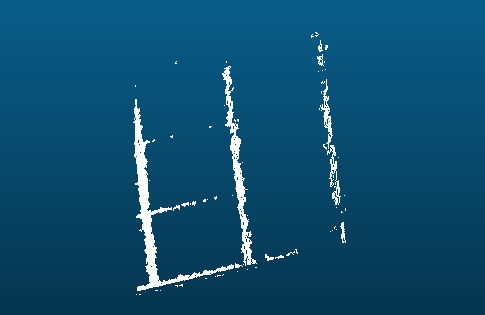


Figure 11 Detect plane 3

Reference

[1]https://blog.csdn.net/soaryy/article/details/82884691

[2]https://blog.csdn.net/weixin\_41758695/article/details/85322304

[3]https://www.programmersought.com/article/8989751838/

[4]https://pcl.readthedocs.io/en/latest/planar\_segmentation.html