

**BÁO CÁO TÌM HIỂU**

Các bộ lọc được sử dụng trong Detect Cuboid

Version 1.0

2021/02/01 Brycen Co., Ltd

# Lịch sử sửa đổi

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | Người viết | Phiên bản | Thời gian | Chú thích |
| 1 | Trần Văn Quang Huy (tvq\_huy@brycen.com.vn). | 1.0 | 2020/02/01 | Phiên bản đầu tiên. |

# Mục lục

[Lịch sử sửa đổi 2](#_Toc63093420)

[Mục lục 3](#_Toc63093421)

[Mục lục hình ảnh 3](#_Toc63093422)

[1. Lấy mẫu dữ liệu với DownSample (Voxel Grid) 4](#_Toc63093423)

[2. Bộ lọc PassThrough 4](#_Toc63093424)

[3. Phép chiếu các điểm lên mặt phẳng với ProjectInliers 5](#_Toc63093425)

[4. Bộ lọc Euclidean Cluster 6](#_Toc63093426)

[Tài liệu tham khảo 8](#_Toc63093427)

# Mục lục hình ảnh

[Hình 1. Hàm DownSample(). 4](#_Toc63093428)

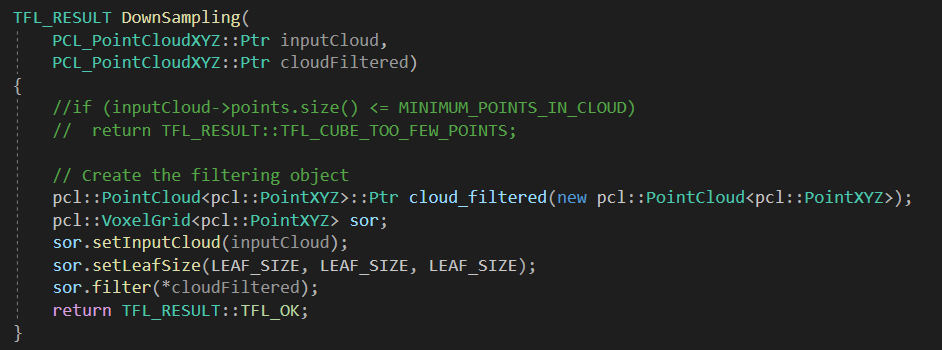
[Hình 2. Bộ lọc Pass Through. 4](#_Toc63093429)

[Hình 3. Hàm ProjectPlaneInliers(). 5](#_Toc63093430)

[Hình 4. Bộ lọc Euclidean Cluster. 6](#_Toc63093431)

# Lấy mẫu dữ liệu với DownSample (Voxel Grid)

Detect Cuboid sử dụng hàm DownSample để lấy mẫu dữ liệu (là triển khai của bộ lọc Voxel Grid), dữ liệu đầu vào sẽ được đi qua hàm này trước khi tiến hành Detect. Mục đích giúp giảm số lượng điểm gây nhiễu và tăng độ chính xác trong quá trình Detect.



Hình 1. Hàm DownSample().

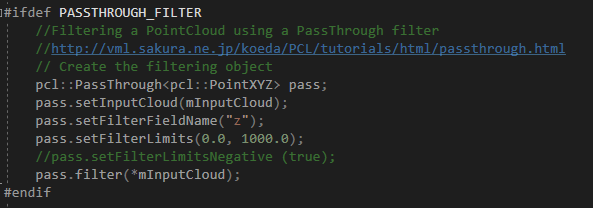
Tham số LEAF\_SIZE của Detect Cuboid sử dụng khá lớn (3.5f) so với Detect Planar (0.01f). Với tham số LEAF\_SIZE càng lớn thì số điểm được giữ càng ít và thời gian xử lý sau đó cũng giảm theo nhưng độ chính xác thì không thể đảm bảo. Hay nói cách việc tăng LEAF\_SIZE sẽ tỉ lệ thuận với tốc độ và tỉ lệ nghịch với chất lượng.

? Nên tìm tham số hợp lý cho Detect Planar.

\*Hàm trên được triển khai ở file pclUtils.cpp

# Bộ lọc PassThrough

Bộ lọc PassThrough được triển khai trong Detect Cuboid nhưng không được bật (chưa biết lý do tắt).



Hình 2. Bộ lọc Pass Through.

Cách thức hoạt động chính của bộ lọc PassThrough rất giống hàm FilterXYZ() trong TFLIB. Giúp lọc các điểm trong vùng được chỉ định theo trục và giới hạn phạm vi.

*Cách thức sử dụng:*

// khai báo

pcl::PassThrough<pcl::PointXYZ> pass;

*// Gán dữ liệu đầu vào*

*pass.setInputCloud(input\_cloud);*

*// Chọn trục muốn lọc*

*pass.setFilterFieldName(“z”);*

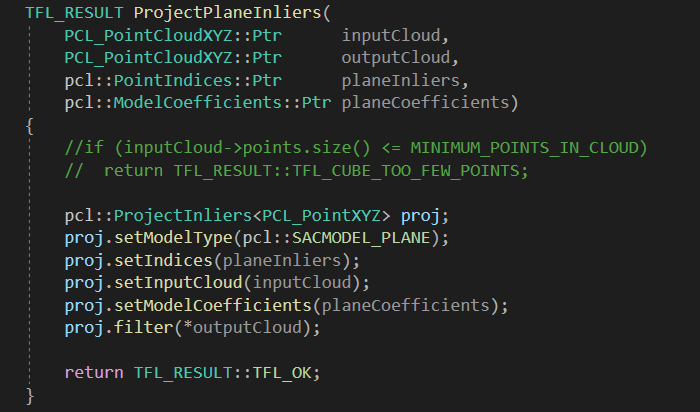
*// Xác định giới hạn vùng chọn*

*pass.setFilterLimits(0.0, 1000.0);*

*// Xuất dữ liệu*

*pass.filter(\*output\_cloud);*

# Phép chiếu các điểm lên mặt phẳng với ProjectInliers



Hình 3. Hàm ProjectPlaneInliers().

Trong quá trình Detect với giá trị DISTANCE\_THRESHOLD > 0 luôn tồn tại các điểm thỏa điều kiện đã cho và được xem là điểm của của mặt phẳng nhưng chưa thực sự nằm chính xác trên mặt phẳng trong không gian 3 chiều. Nên thư viện pcl hỗ trợ hàm giúp điều chỉnh giá trị (X, Y, Z) của các điểm đó về đúng chính xác vị trí trên mặt phẳng. Hay là phép chiếu một điểm lên mặt phẳng cho trước.

*Cách thức sử dụng:*

// khai báo

pcl::ProjectInliers<pcl::PointXYZ> proj;

*// Xác định kiểu mô hình*

*proj.setModelType (pcl::SACMODEL\_PLANE);*

*// Vị trí các điểm thỏa mãn điều kiện mặt phẳng*

*proj.setIndices (inliers);*

*// Gán dữ liệu đầu vào*

*proj.setInputCloud (input\_cloud);*

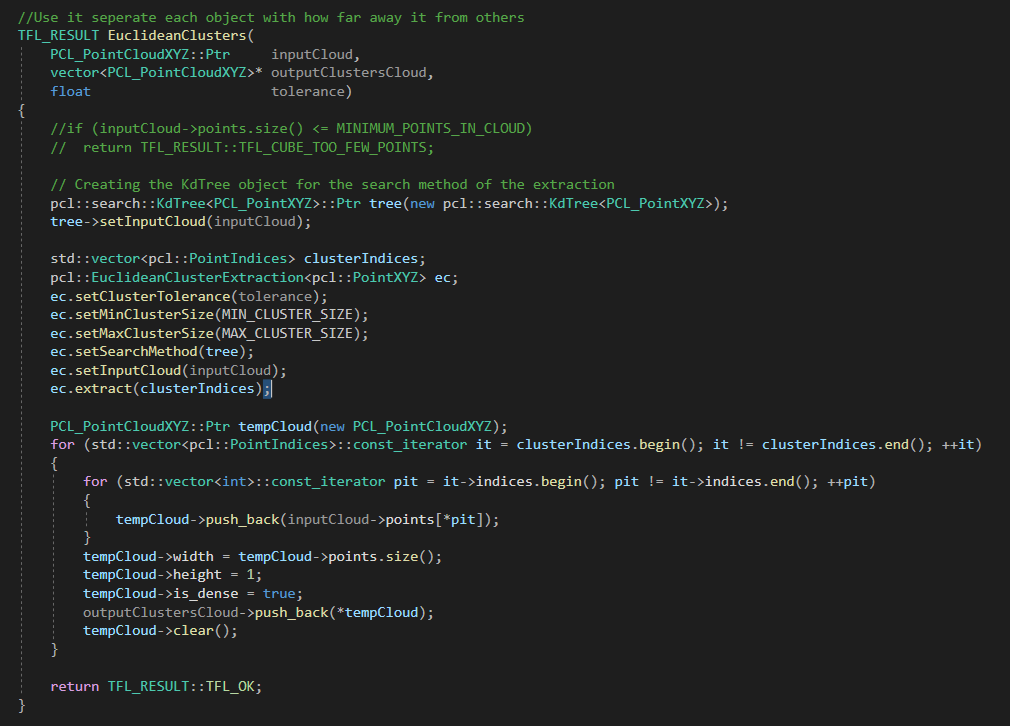
*// Gán giá trị tham số phươn trình mặt phẳng*

*proj.setModelCoefficients(coefficients)*

*// Xuất dữ liệu*

*proj.filter(\*output\_cloud);*

# Bộ lọc Euclidean Cluster



Hình 4. Bộ lọc Euclidean Cluster.

Detect Cuboid cũng sử dụng bộ lọc Euclidean Cluster để loại bỏ các cụm dữ liệu nhỏ không cần thiết. Ngược với cách xử lý của Detect Planar là loại các cụm nhỏ không cần thiết thì Detect Cuboid lấy các cụm lớn cần sử dụng.

Các thông số được thiết lập của Detect Cuboid:

+ setClusterTolerance = 10: khoảng cách tối thiểu mà các điểm cách nhau trong 1 cụm (đơn vị m).

+ setMinClusterSize = 10000: Số điểm tối thiểu có trong 1 cụm

+ setMaxClusterSize = 200000: Số điểm tối đa có trong một cụm

? Nên tìm tham số ClusterTolerance hợp lý cho Detect Planar.

# Tài liệu tham khảo

1. Research Report.docx – Dương Bảo Quý (SVN: https://dvu-pmo.brycen-vn.local:8443/svn/fpga2/personal/tvq\_huy/v1.1 (planar)/document )
2. <http://vml.sakura.ne.jp/koeda/PCL/tutorials/html/passthrough.html>
3. http://vml.sakura.ne.jp/koeda/PCL/tutorials/html/project\_inliers.html