

*Technical Report*

HASHMAP vs MAP  
C++

2021/01/28

Brycen Co., Ltd

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Revised parts and contents | Người viết |
| 1.0 | Jan 28, 2021 | -First version | Tran Van Quang Huy |
| 1.1 | Jan 30, 2021 | - Write about Map and HashMap | Tran Van Quang Huy |
| 1.2 | Jan 31,2021 | - Write about the difference between Map and HashMap  - Conclude | Tran Van Quang Huy |

# Mục lục

[Revision History 2](#_Toc63053432)

[Mục lục 3](#_Toc63053433)

[Danh mục hình ảnh 4](#_Toc63053434)

[Giới thiệu 5](#_Toc63053435)

[1. Map là gì ? 6](#_Toc63053436)

[2.1 Flow diagram of the program 6](#_Toc63053437)

[2.2 Explain the function blocks 6](#_Toc63053438)

[2. HashMap là gì? 8](#_Toc63053439)

[3. Sự khác nhau giữa Map và và HashMap trong C++ 11](#_Toc63053440)

[4. Kết luận 12](#_Toc63053441)

[Reference 13](#_Toc63053442)

# Danh mục hình ảnh

[Figure 1 Flow diagram of the program 6](#_Toc60930179)

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

[Figure 3 Filtering object Function 7](#_Toc60930181)

[Figure 4 Segmentation Function 7](#_Toc60930182)

[Figure 5 Extract Fuction 8](#_Toc60930183)

[Figure 6 Get plane equation 8](#_Toc60930184)

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

[Figure 8 Original Point Cloud Data 9](#_Toc60930186)

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

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

[Figure 11 Detect plane 3 10](#_Toc60930189)

# Giới thiệu

Tài liệu báo cáo về các cấu trúc dữ liệu Map và HashMap (unorder\_map) trong C++:

1. Map là gì ?
2. HashMap là gì ?
3. Sự khác nhau giữ Map và HashMap trong C++
4. Kết luận

# 1. Map (std::map) là gì ?

## 2.1 Flow diagram of the program

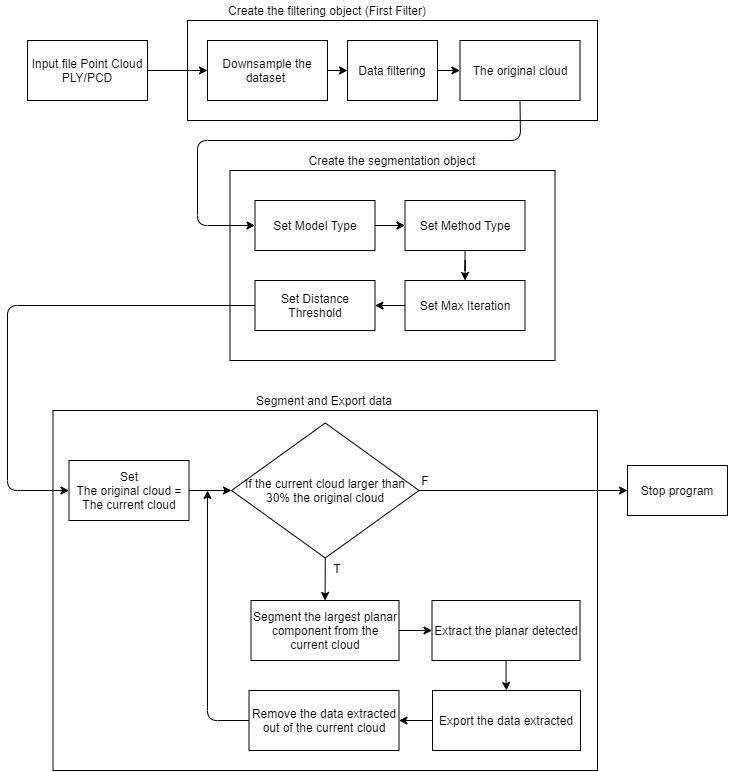


Figure 1 Flow diagram of the program

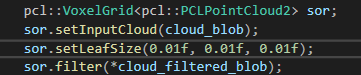
## 2.2 Explain the function blocks

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*.



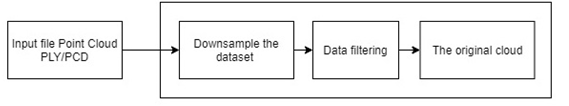
**

Figure 3 Filtering object Function

Then, lines

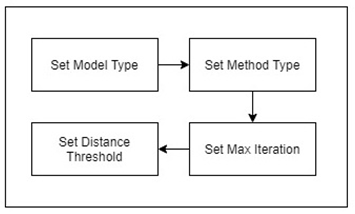
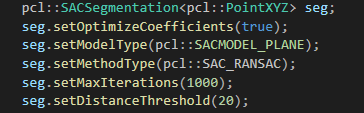


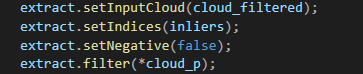
Figure 4 Segmentation Function

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:



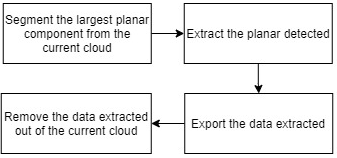


Figure 5 Extract Fuction

Finaly

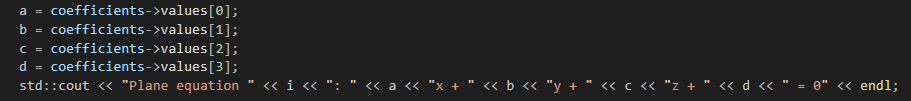


Figure 6 Get plane equation

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

# 2. HashMap (std::unorder\_map) là gì?

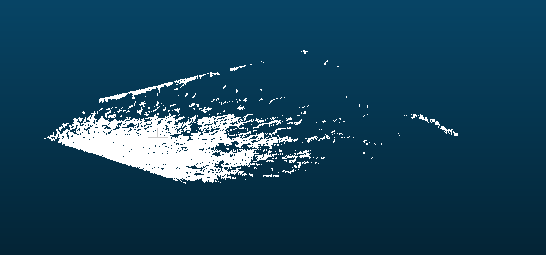


Figure 9 Detect plane 1

PointCloud representing the planar component: 39524 data points.

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

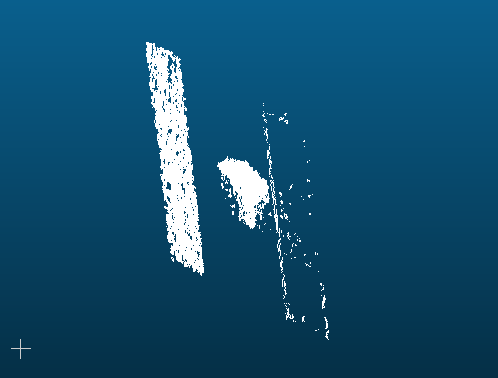


Figure 10 Detect plane 2

PointCloud representing the planar component: 17089 data points.

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

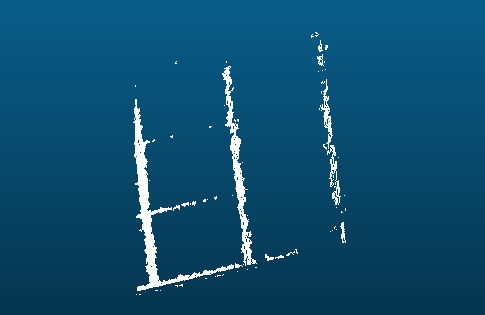


Figure 11 Detect plane 3

# 3. Sự khác nhau giữa Map (std::map) và và HashMap (std::unorder\_map) trong C++

# 4. Kết luận

## 4.1 Khi nào nên sử dụng Map (std::map) ?

* Muốn dữ liệu của mình được sắp xếp.
* Truy cập dữ liệu của phần tử theo thứ tự.
* Khi muốn sử dụng ít bộ nhớ hơn để lưu trữ dữ liệu lớn.

## 4..2 Khi nào nên sử dụng HashMap (std::unorder\_map)

* Khi muốn tìm kiếm cũng như thêm và xóa nhanh mà không quan tâm đến thứ tự các phần tử.

Cả hai cấu trúc dữ liệu trên đều giúp tổ chức và lưu trữ dữ liệu hiệu quả, nên sử dụng một cách hợp lý để đạt được hiệu năng tốt nhất cho từng trường hợp. Đối với dữ liệu nhỏ (dưới một triệu phần tử trở xuống) thì sự khác biệt về tốc độ tìm kiếm thực sự không nhiều nên ta có thể sử dụng Map (std::map) hoặc HashMap (std::unorder\_map). Còn đối với dữ liệu lớn thì ta cần đánh đổi giữa tốc độ tìm kiếm và bộ nhớ, nếu muốn tiết kiệm bộ nhớ thì hãy sử dụng Map (std::map), còn nếu muốn tốc độ thì sử dụng HashMap (std::unorder\_map).

# Tài liệu tham khảo

1. https://www.geeksforgeeks.org/map-vs-unordered\_map-c/
2. <https://www.youtube.com/watch?v=M2fKMP47slQ>