

**Research/Technical Note**

# Point Cloud Processing System Development Based on PCL and Qt

**Liu Dingning, Ding Qiong\***

School of Civil and Transportation Engineering, Guangdong University of Technology, Guangzhou, China

**Email address:**

ldning12@163.com (Ding Qiong)

\*Corresponding author

**To cite this article:**Liu Ding-ning, Ding Qiong. Point Cloud Processing System Development Based on PCL and Qt. *International Journal of Engineering Management*. Vol. 3, No. 1, 2019, pp. 33-39. doi: 10.11648/j.ijem.20190301.16**Received:** June 10, 2019; **Accepted:** July 12, 2019; **Published:** July 31, 2019

---

**Abstract:** LiDAR technology has been widely applied in various disciplines as it can obtain 3D information of targets directly and accurately. However, it is still a challenge to processing LiDAR point clouds efficiently as its huge datasets and complicated processing procedures. Current processing methods need integrate multiple software to complete the whole processing procedures to produce final results which needs lots of time effort and cause low efficiency. By analyzing the theories and methods of LiDAR data processing procedures, this research aims to develop a new point cloud processing software based on PCL and Qt. Firstly, the overall design and modules of the processing system was introduced. The main modules include data management, visualization, filtering, segmentation modeling and auxiliary function. Secondly, to improve system security and maintenance convenience, the system adopts the object-oriented programming method to encapsulate private members and methods of classes, and only open public member variables and methods are available to users. The main classes which were employed in this research were explained. Finally, indoor environments datasets were used to verify the point cloud processing system. The results showed that system has strong interactivity, intuitive display, easy to use and comprehensive features and good results can be derived.

**Keywords:** Point Cloud Processing, Filtering, Modeling, System Design, PCL

---

## 1. Introduction

With the rapid development of technologies such as internet, 3S, computers, and virtual reality, large numbers of innovative applications of 3D geographic information are emerging. They are based on high-precision 3D spatial data and virtual reality technology to restore the real world and give people an immersive experience. 3D laser scanning technology can quickly and efficiently obtain 3D information of research objects, and is widely used in various fields. While to meet the increasing need for high-precision three-dimensional data, how to deal with massive point cloud data quickly and efficiently is still a big challenge for scholars [1-2].

During the last few years, many scholars paid attention on point cloud data processing methods in filtering, segmenting and modeling. Vosselman [3] proposed a slope-based filtering algorithm, Zhang [4] developed a progressive morphological

filter to classify ground points and nonground points. Tan [5] improved the progressive morphological filter for UAV-based point clouds. Sui Lichun [6] introduced "bandwidth" into point cloud morphological filtering. Zuo [7] proposed a knowledge-based triangular progressive filtering method. Lin [8] proposed a method for line segment from point clouds. Raina [9] used deep learning to reconstruct features. Zou [10] employed a deep neural network and Du [11] combined point-based and grid-based features to recognize and extract buildings from point clouds. Many other scholars paid attention on modeling based on the segmented point clouds [12-15].

While for point cloud processing software, there are Terrasolid series software, REALM, SCOP++, etc. However, they still need a lot of manual operations to complete denoising, classification and modeling for point cloud.

This research aims to develop a comprehensive system for

point cloud processing which is called as PCLab (Point Cloud Lab). It was developed based on PCL and Qt and includes the basic procedures of point cloud data processing which contains reading, denoising, filtering, segmentation, modeling and displaying. In addition, it provides key parameters adjustment for each algorithm to achieve a more practical and robust processing results.

## 2. PCLab Point Cloud Processing Modules and Procedures

PCLab includes six functional modules, and other sub-modules are developed based on the first level module as shown in Figure 1.

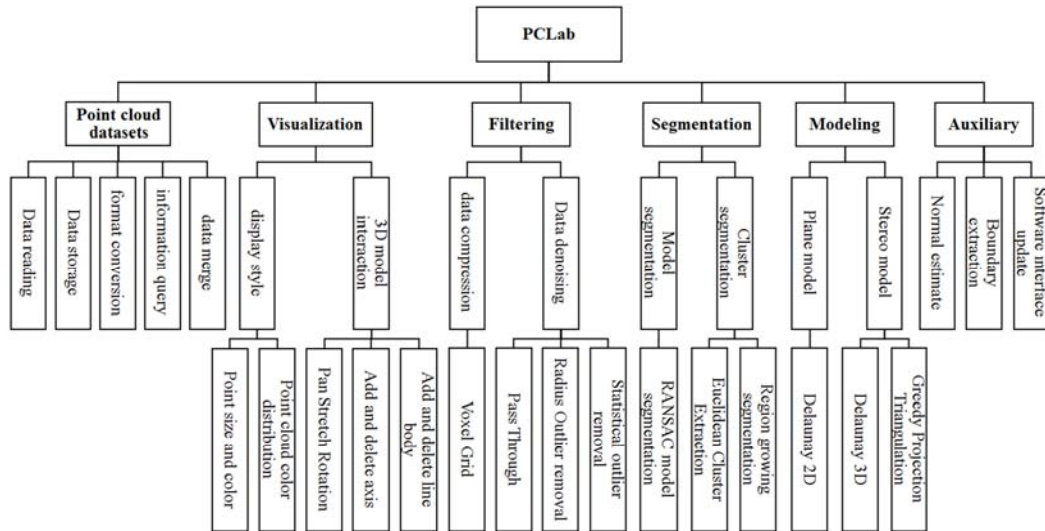


Figure 1. PCLab organization chart.

(1) Point cloud datasets. This module supports typical point cloud data reading, storage, conversion and data structure information query. It provides multi-point cloud data registration to provide integration basis for large scene multi-data projects.

Point cloud data structure: PCD format files are supported, the characteristics of point cloud data are declared in the header file, including fields: VERSION (file version), FIELDS (point dimension), SIZE (dimension size), TYPE (dimension type), COUNT (number of elements per dimension), WIDTH (point cloud dataset width), HEIGHT (point cloud dataset height), VIEWPOINT (viewpoint), POINTS (point cloud total), DATA (point cloud storage type, There are ascii and binary). The content after the header file is regarded as point cloud information. Point cloud merge operation is carried out by reading the data attribute information in the header file.

(2) Visualization. It supports data visualization, browsing and interaction. The color and size of point cloud can be set. Some fundamental operation, such as zooming, dragging and rotating are also achieved.

(3) Point cloud filtering. When acquiring point cloud data, noise points and redundant points are inevitably generated due to the influence of instrument and environment. This module provides various filtering methods for data denoising and data compression. For example, voxels were used for data compression. The barycenter of a voxel is used to replace other points in the voxel to achieve data compression. Data denoising module provides multiple choices for users to derive better results, such as through-pass filtering, limited

radius filtering, and statistical analysis filtering. Pass-through filter user can directly specify coordinate range to cut the point cloud. Limited radius filter draws a circle centered on a point to calculate the number of points falling in the circle. When the quantity is greater than the given value, the point is retained. If it is less than the given value, the point is eliminated. Statistical analysis filter assumes that each point expresses a certain amount of information. The more dense the point, the more information there is. Noise is useless information, it appears with small points. When a given threshold of point number is given, noise can be removed.

(4) Point cloud segmentation. Segmentation divides raw point cloud into multiple point cloud segments, and each point cloud segment corresponds to an independent object. The module provides RANSAC, European clustering and region growing method classical point cloud segmentation algorithms, and provides key parameter setting and modification for each algorithm.

(5) Model reconstruction. Based on each cluster after segmentation, a variety of modeling methods are provided to realize 3D reconstruction of point cloud, including Dionysian 2D reconstruction, Dionysian tetrahedron and greedy projection triangulation reconstruction.

(6) Auxiliary function. Provides normal line estimation, boundary extraction and software interface update auxiliary functions.

### 3. System Development and Environment Building

The system was developed based on VC++, PCL and Qt. PCL is a powerful large-scale cross-platform open source C++ programming library for processing point cloud data, including a variety of advanced point cloud processing algorithms [16]. A large number of point cloud processing algorithms and efficient data structures were included, including point cloud acquisition, filtering, segmentation, registration, retrieval, feature extraction, recognition, surface reconstruction, visualization, etc.. It supports multiple operating system platforms, such as Windows, Linux, Mac OS X and other systems. Qt is a cross-platform C++ application development framework widely used to develop GUI programs. This study uses VS2013 to compile PCL1.8.0 from source code and build a PCL-based GUI in Qt.

PCLab's interface design fully considers the user

experience, and promotes the principle of easy learning and use. The interface of the software is shown in Figure 2. PCLab has a total of 5 partitions: menu, tool, function, object and visualization. The top bar of the interface contains four modules: file, filter, segmentation, and 3D modeling. The tool bar (the second top bar) uses the corresponding icons to reflect common functions. Object area (left side of the interface) shows all datasets, and provides functions for modifying names and deleting objects. Function area (on the right side of the interface) used multiple-tab design for tool setting, different setting tabs are arranged on the top of labels, where the user can modify the key parameters to get satisfactory results and enhance the interaction. Dock design, all functional areas can be changed into a floating window, which is convenient for users to operate multiple tools at the same time and personalized design. Visualization area (central area) supports two forms of display modes which are point cloud data and three-dimensional models.

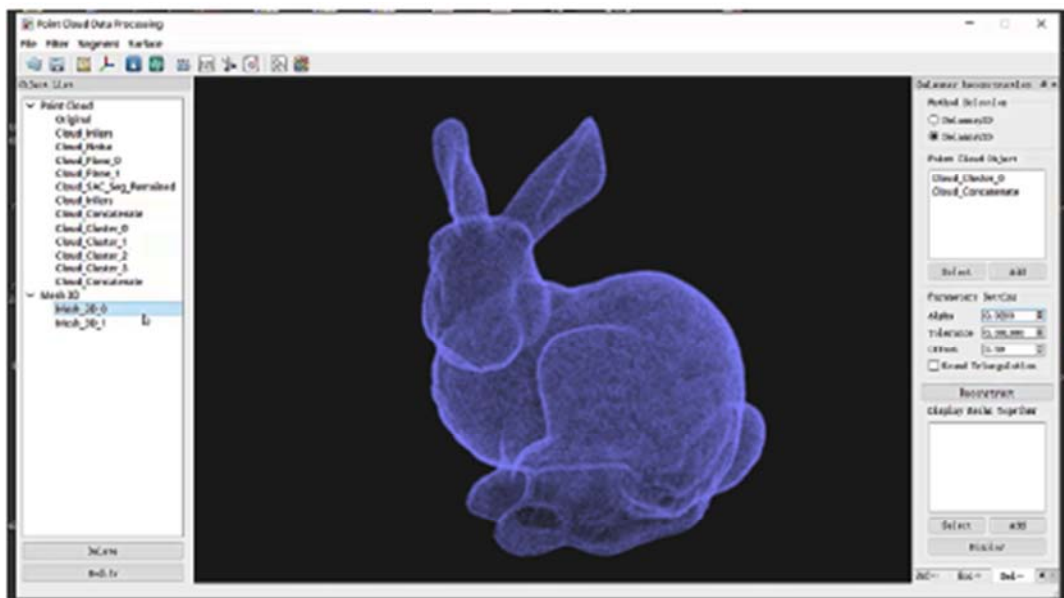


Figure 2. PCLab interface.

To improve system security and maintenance convenience, PCLab adopts the object-oriented programming method to encapsulate private members and methods of classes, and only open public member variables and methods are available to users. The class inheritable and polymorphic properties are improved. The class structure design of the software system will be described as follows.

#### 3.1. Point Cloud Data Class

Point cloud data class is responsible for the management of point cloud data, realizing the function of point cloud data type conversion, point cloud splicing, point cloud information output, etc. It derives the point cloud data file class to realize the access and conversion function of point cloud data files. The relationship between class and point cloud file classes is shown in Figure 3.

#### 3.2. Point Cloud Filter Class

Point cloud filter class (Filter) is an abstract class that implements the point cloud filtering. It derives three-dimensional voxel grid filter (VoxelGridFilter) for data compression, PassThroughFilter and statistical analysis. The class (StatisticalOutlierRemovalFilter) is used to implement the data noise reduction function. The relationship between the point cloud filter class and its derived class is shown in Figure 4.

#### 3.3. Point Cloud Segmentation Class

As an abstract class for implementing point cloud segmentation, the point cloud segmentation class derives SAC Model Segment, Euclidean Cluster Segment and region growth based on random sampling consistency. The Region Growing Segment is used to implement the point cloud model

segmentation function and cluster segmentation function. The relationship between it and the derived class is shown in Figure 5.

### 3.4. Surface Reconstruction Class

Surface reconstruction class (Surface) is an abstract class that implements 3D modeling. It derives the Delaunay 2D modeling class, Delaunay 3D modeling class and Greedy Projection class. They can be used to implement 2D and 3D surface reconstruction functions. The relationship between it and derived classes is shown in Figure 6.

### 3.5. Visualization Class

Visualization is an abstract class for visualizing 3D data. It derives point cloud display style class (PointCloudDisplayStyle) and module visualization class (ModuleVisualization). Among them, point cloud display style class realizes the custom rendering function of point cloud. The module visualization class customizes a variety of visualization solutions for other functional modules. The relationship between visual classes and their derived classes is shown in Figure 7.

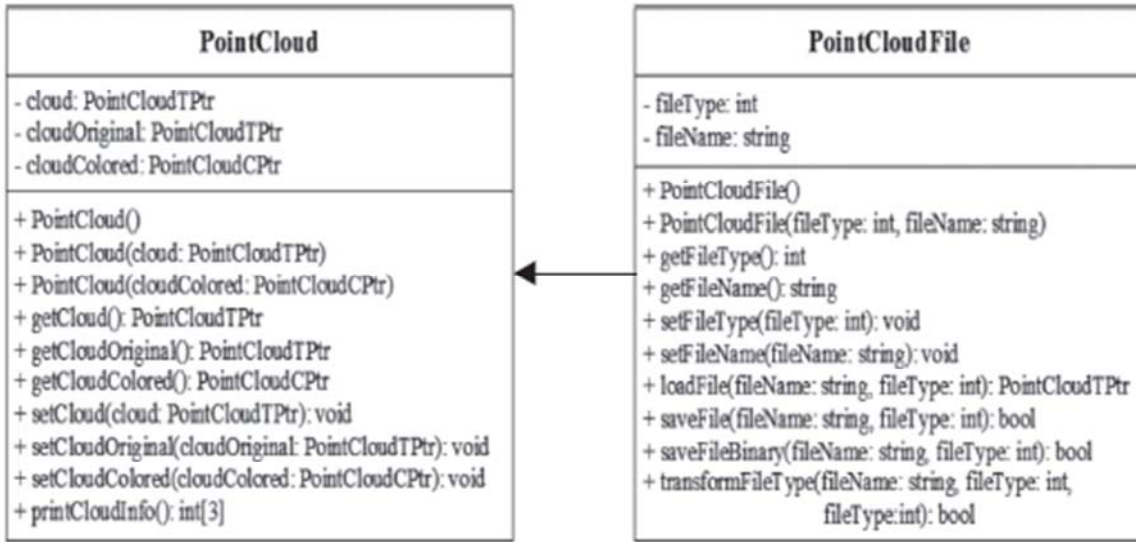


Figure 3. Point cloud structure.

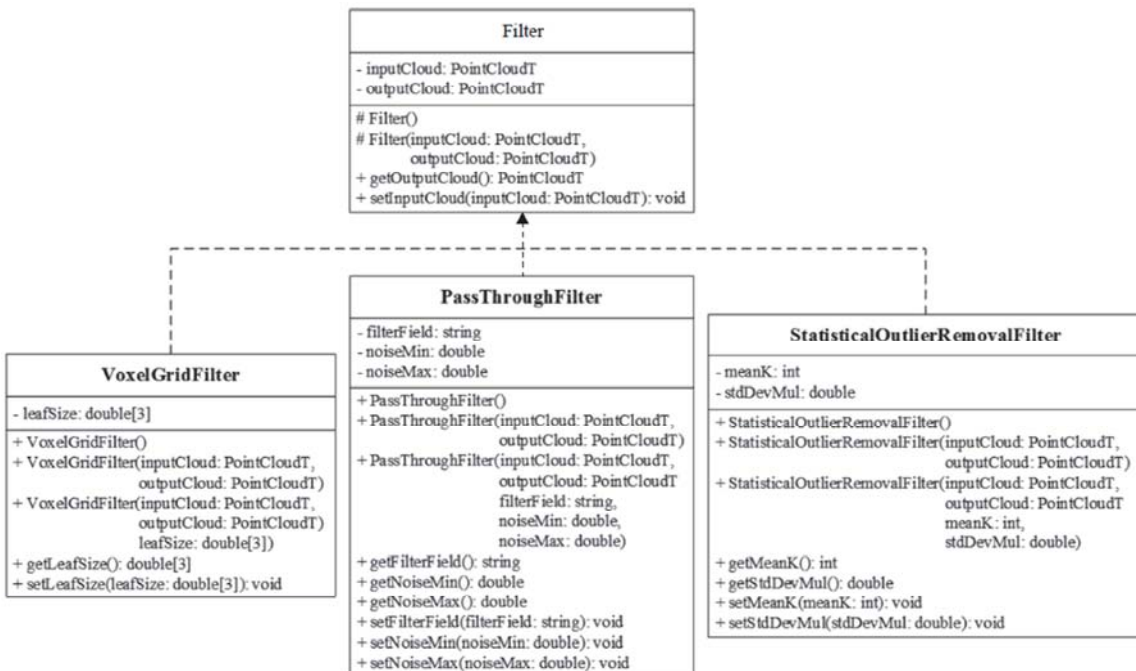


Figure 4. Point cloud filter structure.

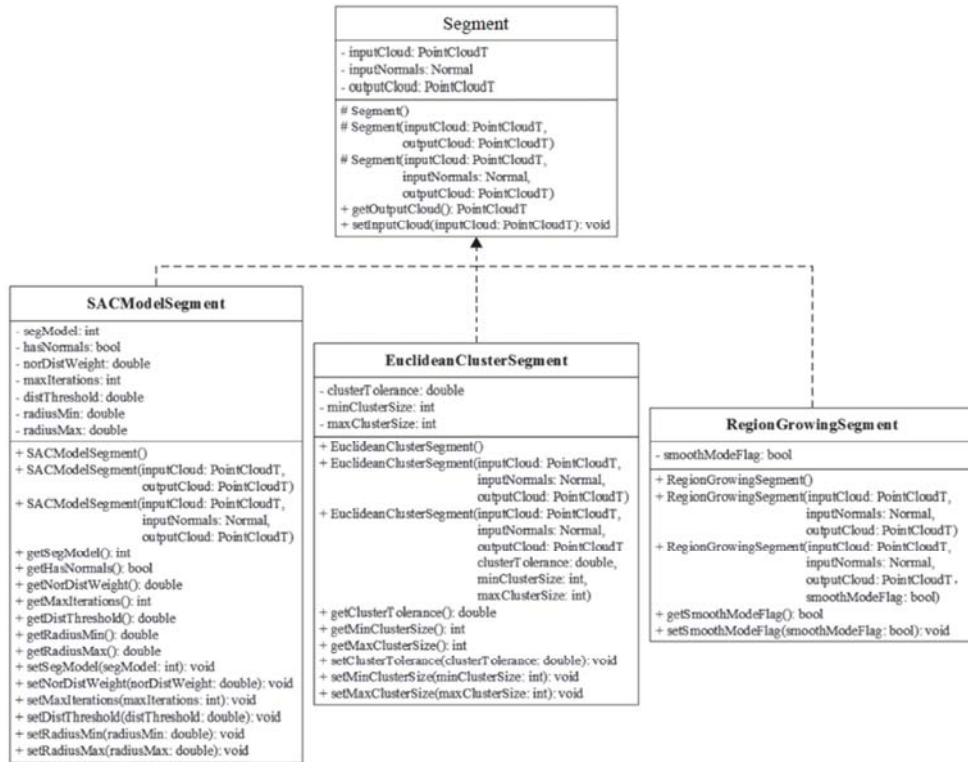


Figure 5. Point cloud segmentation structure.

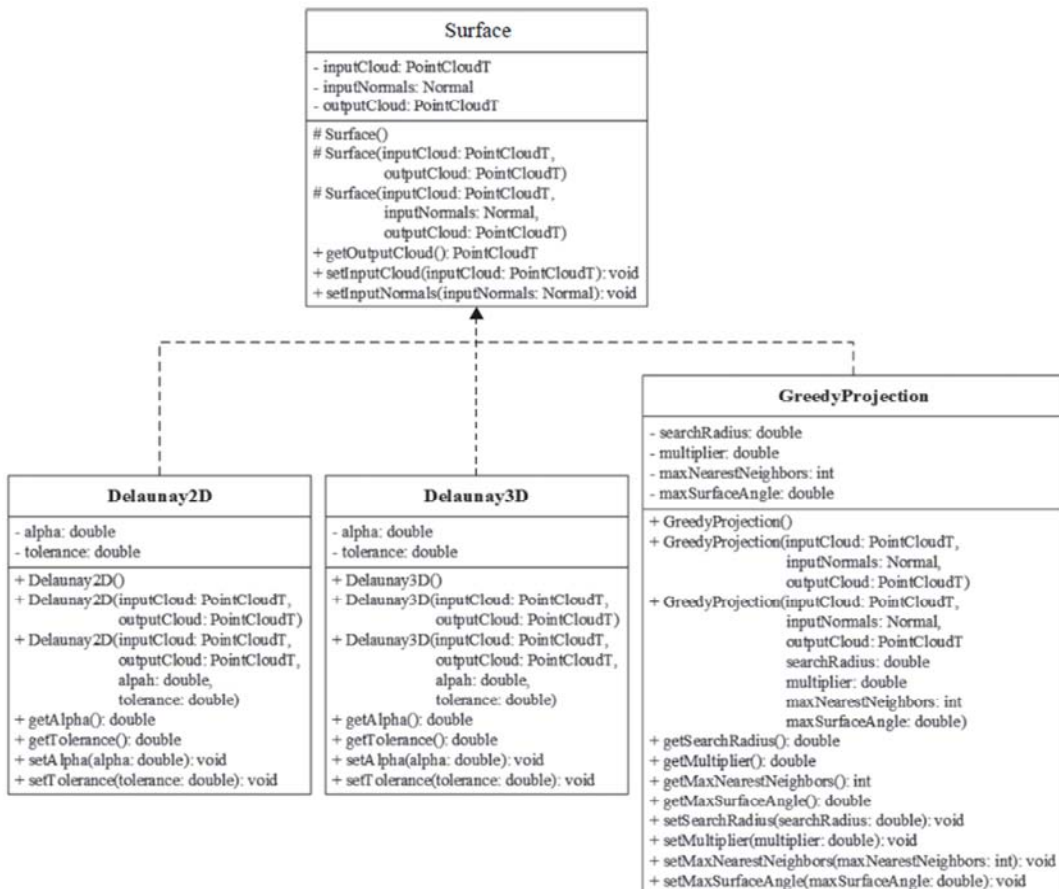


Figure 6. Surface reconstruction structure.

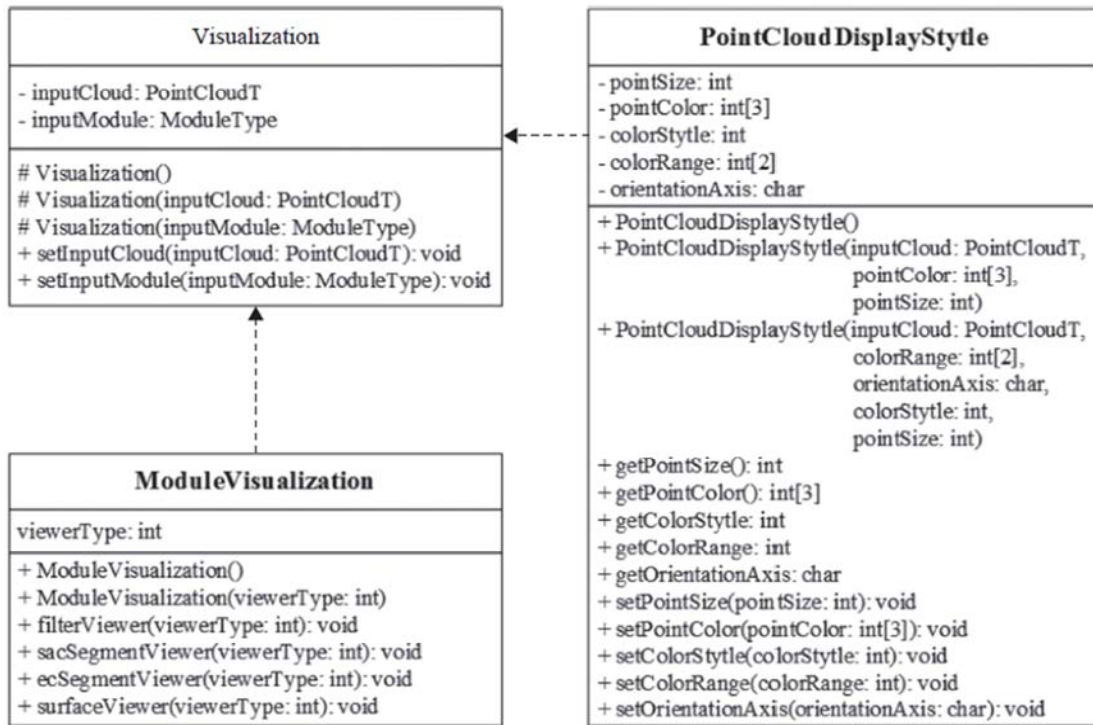


Figure 7. Visualization structure.

### 4. Results and Analysis

In order to verify the point cloud data processing effect of PCLab, Structure Sensor was used to obtain point cloud data. Structure sensor is a 3D depth sensing device with multiple functions that is placed on the back of the ipad/iphone. The sensor can derive three-dimensional point cloud of scanned object.

The point cloud data in this paper is collected in an office. The objects in the scanned environment are wall, floor, chair and table, as shown in Figure 8-1. The specific process is as follows: (1) experimenter holds the Structure Sensor to scan the target to obtain experimental data; (2) import the collected data into the PCLab software and view the basic information of the point cloud data. As shown in Figure 8-2, the point cloud information including wall surfaces, ground, table and chair; (3) the original data is filtered and denoised, outliers are eliminated, and the intra-points are obtained (Figure 8-3), where red is noise point; (4) ground point and wall point are removed by the plane segmentation algorithm provided by PCLab, as shown in the figure. Three walls shown in Figure 8-4 have been divided into three clusters and marked with yellow, purple and blue, leaving the remaining scene features interested (red point cloud); (5) adopts the European clustering segmentation algorithm to produce various segments, as shown in Figure 8-5; (6) 3D reconstruction algorithm is selected to perform three-dimensional modeling of each point cloud segment (Figure 8-6), and modeling results are saved to the hard disk. As can be seen from Figure 8, PCLab can realize point cloud processing flow such as automatic denoising, segmentation and modeling of indoor scene features and provide key parameter interaction options

on each module algorithm to obtain better results.

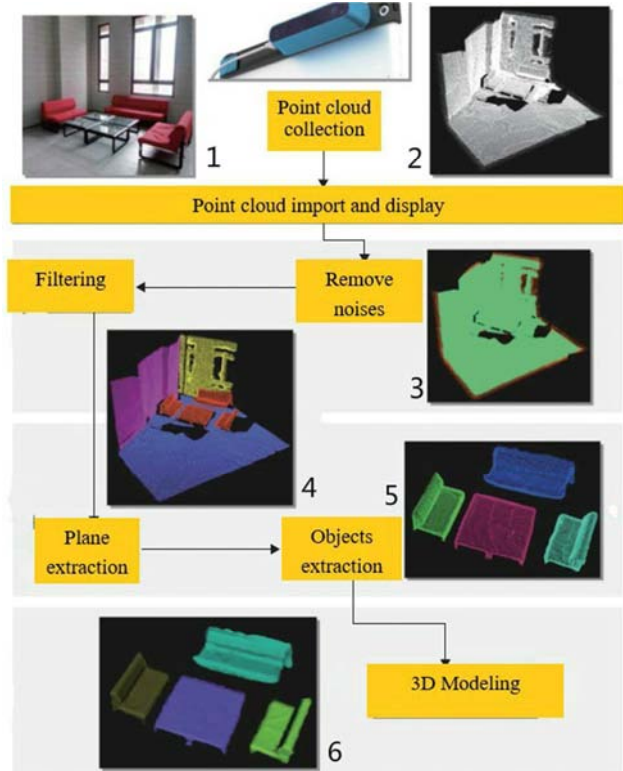


Figure 8. PCLab point cloud processing experiment.

### 5. Conclusion

With the increasing of massive 3D point cloud data, how to

carry out fast and efficient processing to meet application requirements is a hot topic for scholars of different disciplines. This paper develops a PCLab system which integrates point cloud data reading, display, filtering, segmentation, modeling and other functions based on PCL and Qt. Compared with other point cloud system, this system provides alternative key parameter setting options to help users to produce better processing results and reduces the complexity of point cloud processing. In addition, the system is easy to use, open to user and general for all types of operation system. It lays a solid foundation for further development of a more stable, practical and efficient point cloud processing system.

## Acknowledgements

This research was funded in part by the by the Natural Science Fund of China with Project No. 41874031.

## References

- [1] Zhang X H. Theory and Method of Airborne Lidar Measurement Technology. Wuhan: Wuhan University Press, 2007.
- [2] Cheng X J, Jia D F, Cheng X L. Theory and Technology of Massive Point Cloud Data Processing. Shanghai: Tongji University Press, 2014.
- [3] Vosselman G. Slope based filtering of laser altimetry data. International Archives of Photogrammetry and Remote Sensing, 2000, 33 (63/2): 935-942.
- [4] Zhang K Q, Chen S C, Shyu M L, Yan J H, Zhang C C. A progressive morphological filter for removing nonground measurements from airborne LiDAR data. IEEE Transactions on geoscience and remote sensing, 2003, 41 (4): 872-882.
- [5] Tan Y M, Wang S, Xu B, Zhang J B. An improved progressive morphological filter for UAV-based photogrammetric point clouds in river bank monitoring. ISPRS Journal of Photogrammetry and Remote Sensing, 2018, 146: 421-429.
- [6] Sui L C, Yang Y. Filtering of airborne LiDAR point cloud based on car (p, q) model and mathematical morphology . Acta Geodaetica et Cartographica Sinica, 2012, 41 (2): 219-224.
- [7] Zuo Z Q, Zhang Z X, Zhang J Q, et al. A high-quality filtering method with adaptive TIN models for urban LiDAR points based on priori-knowledge. Acta Geodaetica et Cartographica Sinica, 2012, 41 (2): 246-251.
- [8] Lin Y B, Wang C, Cheng J, et al., Line segment extraction for large scale unorganized point clouds. ISPRS Journal of Photogrammetry and Remote Sensing, 2015, 102: 172-183.
- [9] Raina P, Mudur S, Popa T. Sharpness fields in point clouds using deep learning. Computers & Graphics, 2019, 78: 37-53.
- [10] Zhou Z X, Gong J. Automated residential building detection from airborne LiDAR data with deep neural networks. Advanced Engineering Informatics, 2018, 36: 229-241.
- [11] Du S J, Zhang Y S, Zou Z R, Xu S H, He X, Chen S Y. Automatic building extraction from LiDAR data fusion of point and grid-based features. ISPRS Journal of Photogrammetry and Remote Sensing, 2017, 130: 294-307.
- [12] Li M L, Rottensteiner F, Heipke C. *Modelling* of buildings from aerial LiDAR *point clouds* using TINs and label maps. ISPRS Journal of Photogrammetry and Remote Sensing, 2019, 152: 24-33.
- [13] Zhong S K, Zhong Z C, Hua J. Surface reconstruction by parallel and unified particle-based resampling from point clouds. Computer Aided Geometric Design, 2019, 71: 43-62.
- [14] Kang Z Z, Yang J T. A probabilistic graphical model for the classification of mobile LiDAR point clouds. ISPRS Journal of Photogrammetry and Remote Sensing, 2018, 143: 108-123.
- [15] Nguyen H L, Belton D, Helmholtz P. Planar surface detection for sparse and heterogeneous mobile laser scanning point clouds. ISPRS Journal of Photogrammetry and Remote Sensing, 2019, 151: 141-161.
- [16] Rusu R B. Semantic 3D object maps for everyday manipulation in human living environments. Garching: Institut für Informatik, Technische Universität München, 2009.