|  |  |
| --- | --- |
| 产品名称 | 密级 |
|  | 机密 |
| 产品版本 | 共 页 |
|  |

pcl::BoundaryEstimation< PointInT, PointNT, PointOutT >算子功能分析说明书

|  |  |  |  |
| --- | --- | --- | --- |
| 拟制 | 赵宇轩 | 日期 | 2021-10-06 |
| 审核 |  | 日期 | yyyy-mm-dd |
| 批准 |  | 日期 | yyyy-mm-dd |



华为技术有限公司

版权所有 侵权必究

修订记录

|  |  |  |  |
| --- | --- | --- | --- |
| 日期 | 修订版本 | 修改描述 | 作者 |
| 2021-10-06 | 1.0 | 初稿完成 | 赵宇轩 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

目录

[pcl::BoundaryEstimation< PointInT, PointNT, PointOutT >算子功能分析说明书 1](#_Toc4856)

[修订记录 1](#_Toc28474)

[目录 1](#_Toc8030)

[1.1 pcl::BoundaryEstimation< PointInT, PointNT, PointOutT > 2](#_Toc20648)

[1.1.1 功能介绍 2](#_Toc1629)

[1.1.2 使用场景介绍 2](#_Toc7282)

[1.2 pcl::BoundaryEstimation<PointInT, PointNT, PointOutT>::isBoundaryPoint (const pcl::PointCloud<PointInT> &cloud, int q\_idx, const pcl::Indices &indices, const Eigen::Vector4f &u, const Eigen::Vector4f &v, const float angle\_threshold) 4](#_Toc14537)

[1.2.1 接口功能 4](#_Toc31602)

[1.2.2 接口和IR描述 4](#_Toc4637)

[1.2.3 (高性能)实现方案 4](#_Toc278)

[1.3 pcl::BoundaryEstimation<PointInT, PointNT, PointOutT>::isBoundaryPoint (const pcl::PointCloud<PointInT> &cloud, const PointInT &q\_point, const pcl::Indices &indices, const Eigen::Vector4f &u, const Eigen::Vector4f &v, const float angle\_threshold) 6](#_Toc28267)

[1.3.1 接口功能 6](#_Toc15902)

[1.3.2 接口和IR描述 6](#_Toc29852)

[1.3.3 (高性能)实现方案 6](#_Toc23121)

[1.4 pcl::BoundaryEstimation<PointInT, PointNT, PointOutT>::computeFeature (PointCloudOut &output) 8](#_Toc24534)

[1.4.1 接口功能 8](#_Toc29899)

[1.4.2 接口和IR描述 8](#_Toc10509)

[1.4.3 (高性能)实现方案 8](#_Toc18390)

## **pcl::BoundaryEstimation< PointInT, PointNT, PointOutT >**

### 功能介绍

1. 该类实现估计一组点集是否处于指定点的投影区域的边缘位置。
2. 该类实现的关键接口
   1. 判断某点是否在给定点集投影到平面上的投影区域的边缘。

pcl::BoundaryEstimation<PointInT, PointNT, PointOutT>::isBoundaryPoint (

       const pcl::PointCloud<PointInT> &cloud, int q\_idx,

       const pcl::Indices &indices,

       const Eigen::Vector4f &u, const Eigen::Vector4f &v,

       const float angle\_threshold)

* 1. 判断某点是否在给定点集投影到平面上的投影区域的边缘。

pcl::BoundaryEstimation<PointInT, PointNT, PointOutT>::isBoundaryPoint (

       const pcl::PointCloud<PointInT> &cloud, const PointInT &q\_point,

       const pcl::Indices &indices,

       const Eigen::Vector4f &u, const Eigen::Vector4f &v,

       const float angle\_threshold)

* 1. 计算点集是否属于平面的边界上，返回包含边界点的点云

pcl::BoundaryEstimation<PointInT, PointNT, PointOutT>::computeFeature (PointCloudOut &output)

1. 下图是该类的继承关系。

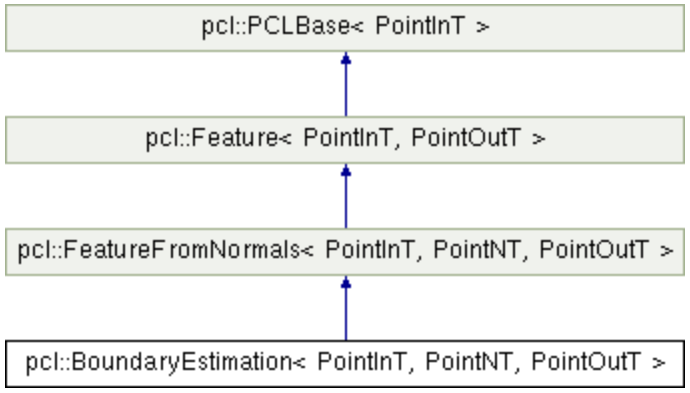


图 1 类的继承关系

### 使用场景介绍

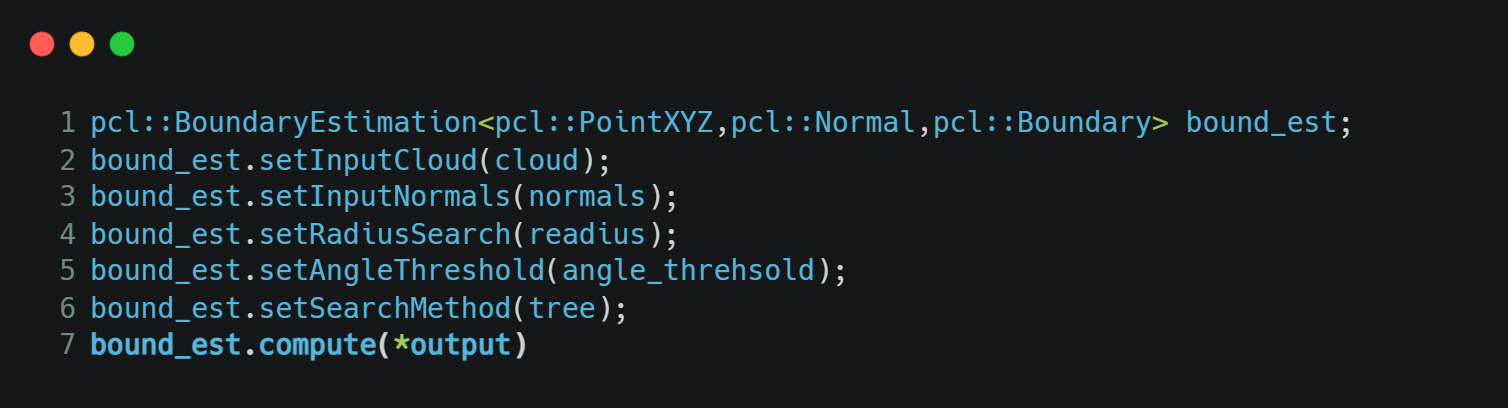


图 2 使用场景

（1）使用场景说明

|  |  |
| --- | --- |
| 行号 | 解释 |
| 1 | 声明一个BoundaryEstimation的实例，记作bound\_est |
| 2~6 | 设置一系列参数 |
| 7 | 调用compute接口，计算输入点云cloud的特征。其中，compute接口是BoundaryEstimation的父类的父类Feature实现的接口。 |

## pcl::BoundaryEstimation<PointInT, PointNT, PointOutT>::isBoundaryPoint (const pcl::PointCloud<PointInT> &cloud, int q\_idx, const pcl::Indices &indices, const Eigen::Vector4f &u, const Eigen::Vector4f &v, const float angle\_threshold)

### 接口功能

判断某点是否在给定点集投影到平面上的投影区域的边缘。

### 接口和IR描述

1. IR原型定义

IR定义：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Op** | **Classify** | **Name** | **Type** | **Type Range** | **Default\_value** | **Format** |
| isBoundaryPoint | Input | cloud | pcl::PointCloud<PointInT> |  |  |  |
| Input | q\_idx | int |  |  |  |
| Input | indices | pcl::Indices |  |  |  |
| Input | u | Eigen::Vector4f |  |  |  |
| Input | v | Eigen::Vector4f |  |  |  |
| Input | angle\_threshold | float |  |  |  |
| Output | isBoundaryPoint | bool |  |  |  |

1. 算子的实现接口定义

pcl::BoundaryEstimation<PointInT, PointNT, PointOutT>::isBoundaryPoint (const pcl::PointCloud<PointInT> &cloud, int q\_idx, const pcl::Indices &indices, const Eigen::Vector4f &u, const Eigen::Vector4f &v, const float angle\_threshold)

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 说明 |
| cloud | pcl::PointCloud<PointInT> | 输入点云 |
| q\_idx | int | 查询点的下标 |
| indices | pcl::Indices | 查询点的临近点集的下标 |
| u | Eigen::Vector4f | u方向 |
| v | Eigen::Vector4f | v方向 |
| angle\_threshold | float | 阈值角度 |

### (高性能)实现方案

行号见源码附件：

|  |  |  |
| --- | --- | --- |
| 行号 | 解释 | Tik接口 |
| 7 | 调用重载的isBoundaryPoint接口 | 7 |

## pcl::BoundaryEstimation<PointInT, PointNT, PointOutT>::isBoundaryPoint (const pcl::PointCloud<PointInT> &cloud, const PointInT &q\_point, const pcl::Indices &indices, const Eigen::Vector4f &u, const Eigen::Vector4f &v, const float angle\_threshold)

### 接口功能

判断某点是否在给定点集投影到平面上的投影区域的边缘。

### 接口和IR描述

1. IR原型定义

IR定义：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Op** | **Classify** | **Name** | **Type** | **Type Range** | **Default\_value** | **Format** |
| isBoundaryPoint | Input | cloud | pcl::PointCloud<PointInT> |  |  |  |
| Input | q\_point | PointInT |  |  |  |
| Input | indices | pcl::Indices |  |  |  |
| Input | u | Eigen::Vector4f |  |  |  |
| Input | v | Eigen::Vector4f |  |  |  |
| Input | angle\_thresholded | float |  |  |  |
| Output | isBoundaryPoint | bool |  |  |  |

1. 算子的实现接口定义

pcl::BoundaryEstimation<PointInT, PointNT, PointOutT>::isBoundaryPoint (const pcl::PointCloud<PointInT> &cloud, const PointInT &q\_point, const pcl::Indices &indices, const Eigen::Vector4f &u, const Eigen::Vector4f &v, const float angle\_threshold)

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 说明 |
| cloud | pcl::PointCloud<PointInT> | 输入点云 |
| q\_point | PointInT | 查询点 |
| indices | pcl::Indices | 查询点的临近点集的下标 |
| u | Eigen::Vector4f | u方向 |
| v | Eigen::Vector4f | v方向 |
| angle\_thresholded | float | 阈值角度 |

### (高性能)实现方案

行号见源码附件：

|  |  |  |
| --- | --- | --- |
| 行号 | 解释 | Tik接口 |
| 25 | 向量相减 | vec\_add |
| 29 | 向量点乘、计算反正切 | vec\_mul、vec\_add  tik中似乎没有与三角函数相关的接口 |
| 38~43 | 变量赋值 | for\_range、vec\_add |
| 45 | 变量赋值 | vec\_add |

## pcl::BoundaryEstimation<PointInT, PointNT, PointOutT>::computeFeature (PointCloudOut &output)

### 接口功能

计算点集是否属于平面的边界上，返回包含边界点的点云。

### 接口和IR描述

1. IR原型定义

IR定义：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Op** | **Classify** | **Name** | **Type** | **Type Range** | **Default\_value** | **Format** |
| computeFeature | Output | output | PointCloudOut |  |  |  |

1. 算子的实现接口定义

pcl::BoundaryEstimation<PointInT, PointNT, PointOutT>::computeFeature (PointCloudOut &output)

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 说明 |
| output | PointCloudOut | 输出点云 |

### (高性能)实现方案

行号见源码：

|  |  |  |
| --- | --- | --- |
| 行号 | 解释 | Tik接口 |
| 17 | 搜索临近点 | 尚未找到searchForNeighbors接口的源码 |
| 30 | 调用isBoundaryPoint接口 |  |
| 52 | 调用isBoundaryPoint接口 |  |