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

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

|  |  |  |  |
| --- | --- | --- | --- |
| 拟制 | 赵宇轩 | 日期 | 2021-10-05 |
| 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 |
| 2021-10-05 | 1.0 | 初稿完成 | 赵宇轩 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

目录

[pcl::BOARDLocalReferenceFrameEstimation< PointInT, PointNT, PointOutT >算子功能分析说明书 1](#_Toc84857519)

[修订记录 1](#_Toc84857520)

[目录 1](#_Toc84857521)

[1.1 pcl::BOARDLocalReferenceFrameEstimation< PointInT, PointNT, PointOutT > 3](#_Toc84857522)

[1.1.1 功能介绍 3](#_Toc84857523)

[1.1.2 使用场景介绍 4](#_Toc84857524)

[1.2 void pcl::BOARDLocalReferenceFrameEstimation<PointInT, PointNT, PointOutT>::randomOrthogonalAxis (Eigen::Vector3f const &axis, Eigen::Vector3f &rand\_ortho\_axis) 5](#_Toc84857525)

[1.2.1 接口功能 5](#_Toc84857526)

[1.2.2 接口和IR描述 5](#_Toc84857527)

[1.2.3 (高性能)实现方案 5](#_Toc84857528)

[1.3 void pcl::BOARDLocalReferenceFrameEstimation<PointInT, PointNT, PointOutT>::projectPointOnPlane (Eigen::Vector3f const &point, Eigen::Vector3f const &origin\_point, Eigen::Vector3f const &plane\_normal, Eigen::Vector3f &projected\_point) 6](#_Toc84857529)

[1.3.1 接口功能 6](#_Toc84857530)

[返回某点到某一平面的投影。 6](#_Toc84857531)

[1.3.2 接口和IR描述 6](#_Toc84857532)

[1.3.3 (高性能)实现方案 6](#_Toc84857533)

[1.4 void pcl::BOARDLocalReferenceFrameEstimation<PointInT, PointNT, PointOutT>::planeFitting ( Eigen::Matrix<float, Eigen::Dynamic, 3> const &points, Eigen::Vector3f &center, Eigen::Vector3f &norm) 8](#_Toc84857534)

[1.4.1 接口功能 8](#_Toc84857535)

[1.4.2 接口和IR描述 8](#_Toc84857536)

[1.4.3 (高性能)实现方案 8](#_Toc84857537)

[1.5 void pcl::BOARDLocalReferenceFrameEstimation<PointInT, PointNT, PointOutT>::normalDisambiguation ( pcl::PointCloud<PointNT> const &normal\_cloud, pcl::Indices const &normal\_indices, Eigen::Vector3f &normal) 10](#_Toc84857538)

[1.5.1 接口功能 10](#_Toc84857539)

[1.5.2 接口和IR描述 10](#_Toc84857540)

[1.5.3 (高性能)实现方案 10](#_Toc84857541)

[1.6 float pcl::BOARDLocalReferenceFrameEstimation<PointInT, PointNT, PointOutT>::getAngleBetweenUnitVectors ( Eigen::Vector3f const &v1, Eigen::Vector3f const &v2, Eigen::Vector3f const &axis) 12](#_Toc84857542)

[1.6.1 接口功能 12](#_Toc84857543)

[1.6.2 接口和IR描述 12](#_Toc84857544)

[1.6.3 (高性能)实现方案 12](#_Toc84857545)

[1.7 void pcl::BOARDLocalReferenceFrameEstimation<PointInT, PointNT, PointOutT>::directedOrthogonalAxis ( Eigen::Vector3f const &axis, Eigen::Vector3f const &axis\_origin, Eigen::Vector3f const &point, Eigen::Vector3f &directed\_ortho\_axis) 14](#_Toc84857546)

[1.7.1 接口功能 14](#_Toc84857547)

[1.7.2 接口和IR描述 14](#_Toc84857548)

[1.7.3 (高性能)实现方案 14](#_Toc84857549)

[1.8 float pcl::BOARDLocalReferenceFrameEstimation<PointInT, PointNT, PointOutT>::computePointLRF (const int &index, Eigen::Matrix3f &lrf) 16](#_Toc84857550)

[1.8.1 接口功能 16](#_Toc84857551)

[1.8.2 接口和IR描述 16](#_Toc84857552)

[1.8.3 (高性能)实现方案 16](#_Toc84857553)

[1.9 void pcl::BOARDLocalReferenceFrameEstimation<PointInT, PointNT, PointOutT>::computeFeature (PointCloudOut &output) 18](#_Toc84857554)

[1.9.1 接口功能 18](#_Toc84857555)

[1.9.2 接口和IR描述 18](#_Toc84857556)

[1.9.3 (高性能)实现方案 18](#_Toc84857557)

## pcl::BOARDLocalReferenceFrameEstimation< PointInT, PointNT, PointOutT >

### 功能介绍

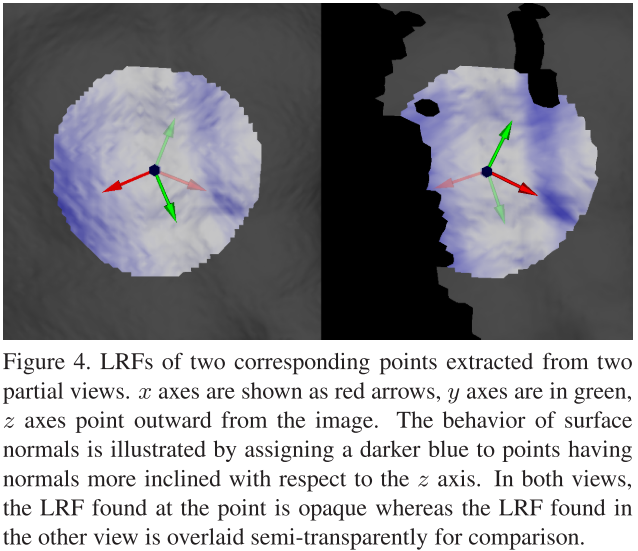
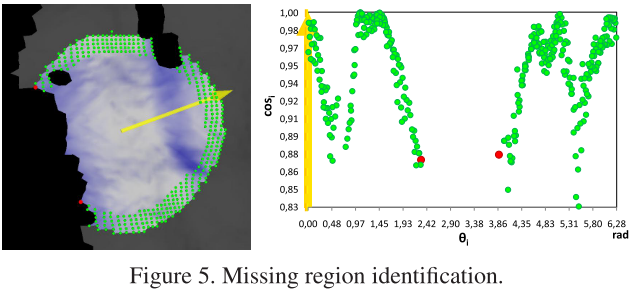
1. 该类实现文献（On the repeatability of the local reference frame for partial shape matching）中讲的局部坐标系估计方法，该方法在估计局部坐标系时对处理点云边缘或有孔洞处有特殊的处理方式，比以往常用的局部坐标系估计方法稳定，可重复性好。
2. 算法实现对点云中的所有点计算他们的局部坐标系(Local Reference Frame, LRF),下面从对点云中某点p计算其LRF来对算法进行说明。
3. 首先确定z轴：通过点p的邻居点求出拟合平面的法线np，通过邻居点的法线计算平均法线na（对邻居点的法线进行求和后标准化），最后将np作为z轴正半轴（通过np与na做内积来确定np的方向）。
4. 然后确定x轴：x轴由z轴及点p（以点p作为LRF的原点）和点pmin求得（x轴为从点p出发指向pmin的一个向量在z轴所定义的法平面的投影）。pmin表示使得z轴与法向量nr,i所形成的夹角的cosine值最小的某个点，而nr,i则表示点pi所定义的ringr(pi)中的点所计算出的平均法线。算法对ringr(pi)的定义如下，其中A(pi)为某点pi的邻居点的集合。

ring0(pi)=pi

ring1(pi)=ring0(pi)∪A(pi)

ringr(pi)=ringr-1(pi)∪{A(pr):pr∈ringr-1(pi)}

考虑到时间复杂度，算法只对与点p相距大于Tm×Rm(Tm=0.85)的pi进行计算，且论文中**r的取值为2**。

1. 最后计算y轴：z轴和x轴做叉乘即可得出y轴。
2. 但是，点云图像的数据并不总是完美的（如Figure 4的右半部分所示），图像边界的遗失会极大地影响可复用的LRF的计算，因此本文提出了一个改进方法，使得我们能对边界数据遗失了的点云数据进行LRF的计算。改进方法分为3步走。
3. 第一步，寻找消失域(missing region)：在拟合平面上随机生成一个向量，记作Vr，将从点p指向点pi的向量在拟合平面的投影记作Vp，计算Vr与Vp的夹角θ。认为当连续两点pa, pb的θ的差值Δθ>2π×Th（Th=0.2）时，由pa, pb组成了消失域的边界（Figure 5）。
4. 第二步，判断pmin是否在消失域中：当S>TS（TS=0.1）时，认为pmin在该消失域之中，若有多个消失域，选择S值最大的进行计算，计算公式如下。
5. 第三步，定位pmin：按下方公式计算出t，然后将第一步生成的随机向量Vr在拟合平面上转动t度后形成的新向量作为x轴。
6. 该类实现的关键接口
   1. 输入一个坐标轴，返回一个指向随机且与输入坐标轴正交的坐标轴

pcl::BOARDLocalReferenceFrameEstimation<PointInT, PointNT, PointOutT>::randomOrthogonalAxis (

                                                                                             Eigen::Vector3f const &axis,

                                                                                             Eigen::Vector3f &rand\_ortho\_axis)

* 1. 返回某点到某一平面的投影

pcl::BOARDLocalReferenceFrameEstimation<PointInT, PointNT, PointOutT>::projectPointOnPlane (

                                                                                            Eigen::Vector3f const &point,

                                                                                            Eigen::Vector3f const &origin\_point,

                                                                                            Eigen::Vector3f const &plane\_normal,

                                                                                            Eigen::Vector3f &projected\_point)

* 1. 通过JacobiSVD求空间中点集的拟合平面

pcl::BOARDLocalReferenceFrameEstimation<PointInT, PointNT, PointOutT>::planeFitting (

                                                                                     Eigen::Matrix<float,

                                                                                     Eigen::Dynamic, 3> const &points,

                                                                                     Eigen::Vector3f &center,

                                                                                     Eigen::Vector3f &norm)

* 1. 消除某法线在方向上的歧义

pcl::BOARDLocalReferenceFrameEstimation<PointInT, PointNT, PointOutT>::normalDisambiguation (

                                                                                             pcl::PointCloud<PointNT> const &normal\_cloud,

                                                                                             pcl::Indices const &normal\_indices,

                                                                                             Eigen::Vector3f &normal)

* 1. 计算两向量之间的夹角，用弧度表示

pcl::BOARDLocalReferenceFrameEstimation<PointInT, PointNT, PointOutT>::getAngleBetweenUnitVectors (

                                                                                                   Eigen::Vector3f const &v1,

                                                                                                   Eigen::Vector3f const &v2,

                                                                                                   Eigen::Vector3f const &axis)

* 1. 给定一个点和一个向量，计算与给定向量正交且指向给定点的向量

pcl::BOARDLocalReferenceFrameEstimation<PointInT, PointNT, PointOutT>::directedOrthogonalAxis (

                                                                                               Eigen::Vector3f const &axis,

                                                                                               Eigen::Vector3f const &axis\_origin,

                                                                                               Eigen::Vector3f const &point,

                                                                                               Eigen::Vector3f &directed\_ortho\_axis)

* 1. 计算给定点的LRF描述子（基于给定点的部分3D邻近点和法线）

pcl::BOARDLocalReferenceFrameEstimation<PointInT, PointNT, PointOutT>::computePointLRF (const int &index, Eigen::Matrix3f &lrf)

* 1. 抽象特征计算接口

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

1. 下图是该类的继承关系。

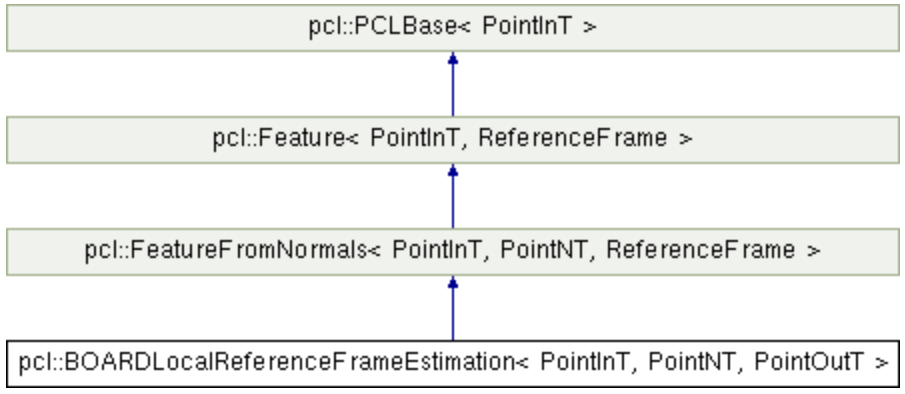


图 1 类的继承关系

### 使用场景介绍

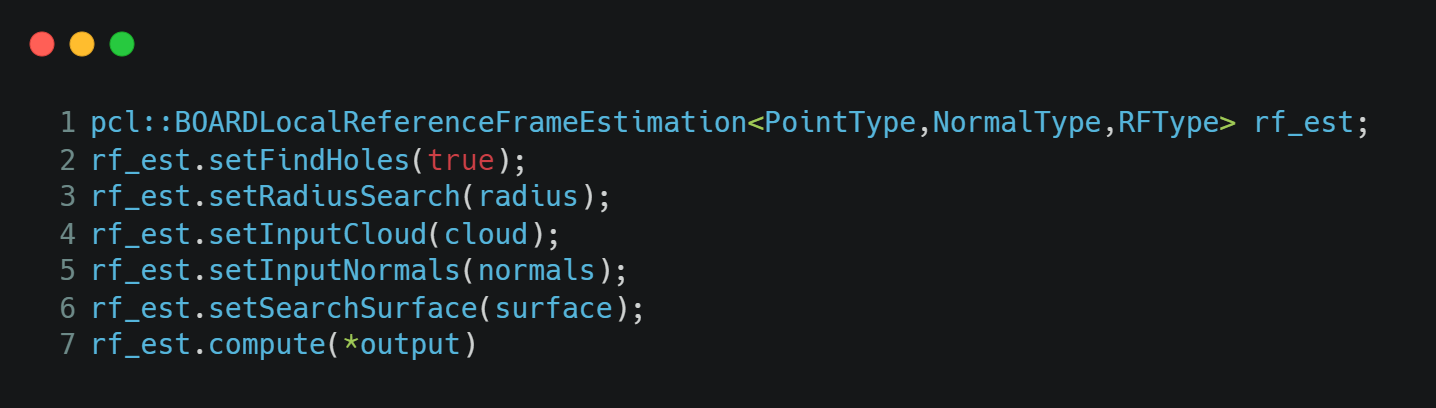


图 2 使用场景

（1）使用场景说明

|  |  |
| --- | --- |
| 行号 | 解释 |
| 1 | 声明一个BOARDLocalReferenceFrameEstimation实例，记作rf\_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BOARDLocalReferenceFrameEstimation的父类的父类Feature实现的接口。 |

## void pcl::BOARDLocalReferenceFrameEstimation<PointInT, PointNT, PointOutT>::randomOrthogonalAxis (Eigen::Vector3f const &axis, Eigen::Vector3f &rand\_ortho\_axis)

### 接口功能

输入一个坐标轴，返回一个指向随机且与输入坐标轴正交的坐标轴。

### 接口和IR描述

1. IR原型定义

IR定义：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Op** | **Classify** | **Name** | **Type** | **Type Range** | **Default\_value** | **Format** |
| randomOrthogonalAxis | Input | axis | Eigen::Vector3f |  |  |  |
| Output | rand\_ortho\_axis | Egien::Vector3f |  |  |  |

1. 算子的实现接口定义

pcl::BOARDLocalReferenceFrameEstimation<PointInT, PointNT, PointOutT>::randomOrthogonalAxis (Eigen::Vector3f const &axis, Eigen::Vector3f &rand\_ortho\_axis)

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 说明 |
| axis | Eigen::Vector3f | 输入的坐标轴 |
| rand\_ortho\_axis | Egien::Vector3f | 输出的坐标轴 |

### (高性能)实现方案

|  |  |
| --- | --- |
| 该接口需要用到的Tik接口 | vec\_mul，vec\_muls，vec\_add，vec\_rec，scalar\_sqrt |
| 无法用Tik接口实现的部分 | 无 |

行号见源码附件：

|  |  |  |
| --- | --- | --- |
| 行号 | 解释 | Tik接口 |
| 6~23 | 标量混合四则运算 | vec\_rec、vec\_add、vec\_mul |
| 25 | 向量标准化 | vec\_muls、vec\_rec、scalar\_sqrt |

## void pcl::BOARDLocalReferenceFrameEstimation<PointInT, PointNT, PointOutT>::projectPointOnPlane (Eigen::Vector3f const &point, Eigen::Vector3f const &origin\_point, Eigen::Vector3f const &plane\_normal, Eigen::Vector3f &projected\_point)

### 接口功能

### 返回某点到某一平面的投影。

### 接口和IR描述

1. IR原型定义

IR定义：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Op** | **Classify** | **Name** | **Type** | **Type Range** | **Default\_value** | **Format** |
| projectPointOnPlane | Input | point | Eigen::Vector3f |  |  |  |
| Input | origin\_point | Eigen::Vector3f |  |  |  |
| Input | plane\_normal | Eigen::Vector3f |  |  |  |
| Output | projected\_point | Eigen::Vector3f |  |  |  |

1. 算子的实现接口定义

pcl::BOARDLocalReferenceFrameEstimation<PointInT, PointNT, PointOutT>::projectPointOnPlane (Eigen::Vector3f const &point, Eigen::Vector3f const &origin\_point, Eigen::Vector3f const &plane\_normal, Eigen::Vector3f &projected\_point)

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 说明 |
| point | Eigen::Vector3f | 空间中一点 |
| origin\_point | Eigen::Vector3f | 平面上一点 |
| plane\_normal | Eigen::Vector3f | 平面法线 |
| projected\_point | Eigen::Vector3f | 空间中一点在平面的投影 |

### (高性能)实现方案

|  |  |
| --- | --- |
| 该接口需要用到的Tik接口 | vec\_mul，vec\_add，vec\_sub |
| 无法用Tik接口实现的部分 | 无 |

行号见源码附件：

|  |  |  |
| --- | --- | --- |
| 行号 | 解释 | Tik接口 |
| 12 | 向量点乘 | vec\_add、vec\_mul |
| 14 | 标量混合四则运算 | vec\_sub、vec\_mul |

## void pcl::BOARDLocalReferenceFrameEstimation<PointInT, PointNT, PointOutT>::planeFitting ( Eigen::Matrix<float, Eigen::Dynamic, 3> const &points, Eigen::Vector3f &center, Eigen::Vector3f &norm)

### 接口功能

通过JacobiSVD求空间中点集的拟合平面。

### 接口和IR描述

1. IR原型定义

IR定义：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Op** | **Classify** | **Name** | **Type** | **Type Range** | **Default\_value** | **Format** |
| planeFitting | Input | points | Eigen::Matrix<float, Eigen::Dynamic,3> |  |  |  |
| Output | center | Eigen::Vector3f |  |  |  |
| Output | norm | Eigen::Vector3f |  |  |  |

1. 算子的实现接口定义

pcl::BOARDLocalReferenceFrameEstimation<PointInT, PointNT, PointOutT>::planeFitting ( Eigen::Matrix<float, Eigen::Dynamic, 3> const &points, Eigen::Vector3f &center, Eigen::Vector3f &norm)

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 说明 |
| points | Eigen::Matrix<float, Eigen::Dynamic,3> | 三维空间中的点集 |
| center | Eigen::Vector3f | 输入点集的质心 |
| norm | Eigen::Vector3f | 拟合平面的法线 |

### (高性能)实现方案

|  |  |
| --- | --- |
| 该接口需要用到的Tik接口 | vec\_mul，vec\_add，vec\_sub |
| 无法用Tik接口实现的部分 | 无 |

行号见源码：

|  |  |  |
| --- | --- | --- |
| 行号 | 解释 | Tik接口 |
| 12~16 | 判断输入点集是否为空 |  |
| 19 | 张量按列求均值并转置 | vec\_add、vec\_mul、vec\_rec |
| 22 | 张量减法 | vec\_sub |

## void pcl::BOARDLocalReferenceFrameEstimation<PointInT, PointNT, PointOutT>::normalDisambiguation ( pcl::PointCloud<PointNT> const &normal\_cloud, pcl::Indices const &normal\_indices, Eigen::Vector3f &normal)

### 接口功能

消除某法相在方向上的歧义。

### 接口和IR描述

1. IR原型定义

IR定义：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Op** | **Classify** | **Name** | **Type** | **Type Range** | **Default\_value** | **Format** |
| normalDisambiguation | Input | normals\_cloud | pcl::PointCloud<PointNT> |  |  |  |
| Input | normal\_indices | pcl::Indices |  |  |  |
| Input/Output | normal | Eigen::Vector3f |  |  |  |

1. 算子的实现接口定义

pcl::BOARDLocalReferenceFrameEstimation<PointInT, PointNT, PointOutT>::normalDisambiguation ( pcl::PointCloud<PointNT> const &normal\_cloud, pcl::Indices const &normal\_indices, Eigen::Vector3f &normal)

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 说明 |
| normals\_cloud | pcl::PointCloud<PointNT> | 用于计算的法线集合 |
| normal\_indices | pcl::Indices | 需要被用于计算的法线的下标的集合 |
| normal | Eigen::Vector3f | 需要消除歧义的法线 |

### (高性能)实现方案

|  |  |
| --- | --- |
| 该接口需要用到的Tik接口 | vec\_mul，vec\_rec，vec\_add |
| 无法用Tik接口实现的部分 | 无 |

行号见源码：

|  |  |  |
| --- | --- | --- |
| 行号 | 解释 | Tik接口 |
| 17 | 向量标准化 | vec\_mul、vec\_rec、vec\_add |
| 19 | 向量点乘 | vec\_mul、vec\_add |

## float pcl::BOARDLocalReferenceFrameEstimation<PointInT, PointNT, PointOutT>::getAngleBetweenUnitVectors ( Eigen::Vector3f const &v1, Eigen::Vector3f const &v2, Eigen::Vector3f const &axis)

### 接口功能

计算两向量之间的夹角，用弧度表示。

### 接口和IR描述

1. IR原型定义

IR定义：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Op** | **Classify** | **Name** | **Type** | **Type Range** | **Default\_value** | **Format** |
| getAngleBetweenUnitVectors | Input | v1 | Eigen::Vector3f |  |  |  |
| Input | v2 | Eigen::Vector3f |  |  |  |
| Input | axis | Eigen::Vector3f |  |  |  |
| Output | angle | float |  |  |  |

1. 算子的实现接口定义

pcl::BOARDLocalReferenceFrameEstimation<PointInT, PointNT, PointOutT>::getAngleBetweenUnitVectors ( Eigen::Vector3f const &v1, Eigen::Vector3f const &v2, Eigen::Vector3f const &axis)

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 说明 |
| v1 | Eigen::Vector3f | 向量v1 |
| v2 | Eigen::Vector3f | 向量v2 |
| axis | Eigen::Vector3f | 旋转轴，与v1、v2所定义的平面垂直 |
| angle | float | 向量v1与v2的夹角 |

### (高性能)实现方案

|  |  |
| --- | --- |
| 该接口需要用到的Tik接口 | vec\_mul，vec\_add |
| 无法用Tik接口实现的部分 | 反三角函数 |

行号见源码：

|  |  |  |
| --- | --- | --- |
| 行号 | 解释 | Tik接口 |
| 8 | 向量叉乘 | vec\_mul、vec\_add |
| 9 | 计算两向量之间的夹角 | 需要用到反三角函数 |
| 11 | 向量点乘 | vec\_mul、vec\_add |

## void pcl::BOARDLocalReferenceFrameEstimation<PointInT, PointNT, PointOutT>::directedOrthogonalAxis ( Eigen::Vector3f const &axis, Eigen::Vector3f const &axis\_origin, Eigen::Vector3f const &point, Eigen::Vector3f &directed\_ortho\_axis)

### 接口功能

给定一个点和一个向量，计算与给定向量正交且指向给定点的向量。

### 接口和IR描述

1. IR原型定义

IR定义：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Op** | **Classify** | **Name** | **Type** | **Type Range** | **Default\_value** | **Format** |
| directedOrthogonalAxis | Input | axis | Eigen::Vector3f |  |  |  |
| Input | axis\_origin | Eigen::Vector3f |  |  |  |
| Input | point | Eigen::Vector3f |  |  |  |
| Output | directed\_ortho\_axis | Eigen::Vector3f |  |  |  |

1. 算子的实现接口定义

pcl::BOARDLocalReferenceFrameEstimation<PointInT, PointNT, PointOutT>::directedOrthogonalAxis ( Eigen::Vector3f const &axis, Eigen::Vector3f const &axis\_origin, Eigen::Vector3f const &point, Eigen::Vector3f &directed\_ortho\_axis)

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 说明 |
| axis | Eigen::Vector3f | 给定向量的终点 |
| axis\_origin | Eigen::Vector3f | 给定向量的起点 |
| point | Eigen::Vector3f | 给定点 |
| directed\_ortho\_axis | Eigen::Vector3f | 指向给定点且与给定向量正交的向量 |

### (高性能)实现方案

|  |  |
| --- | --- |
| 该接口需要用到的Tik接口 | vec\_mul，vec\_add，vec\_sub，vec\_rec |
| 无法用Tik接口实现的部分 | 同该接口中调用到的接口 |

行号见源码：

|  |  |  |
| --- | --- | --- |
| 行号 | 解释 | Tik接口 |
| 9 | 调用projectPointOnPlane接口 |  |
| 10 | 向量减法 | vec\_sub |
| 11 | 向量标准化 | vec\_mul、vec\_add、vec\_rec |

## float pcl::BOARDLocalReferenceFrameEstimation<PointInT, PointNT, PointOutT>::computePointLRF (const int &index, Eigen::Matrix3f &lrf)

### 接口功能

计算给定点的LRF描述子（基于给定点的部分3D邻近点和法线）。

### 接口和IR描述

1. IR原型定义

IR定义：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Op** | **Classify** | **Name** | **Type** | **Type Range** | **Default\_value** | **Format** |
| computePointLRF | Input | index | const int |  |  |  |
| Output | lrf | Eigen::Matrix3f |  |  |  |

1. 算子的实现接口定义

pcl::BOARDLocalReferenceFrameEstimation<PointInT, PointNT, PointOutT>::computePointLRF (const int &index, Eigen::Matrix3f &lrf)

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 说明 |
| index | const int | 输入点集的下标 |
| lrf | Eigen::Matrix3f | 计算得到的local reference frame |

### (高性能)实现方案

|  |  |
| --- | --- |
| 该接口需要用到的Tik接口 | vec\_mul，vec\_add，vec\_sub，vec\_rec，scalar\_min，scalar\_conv |
| 无法用Tik接口实现的部分 | 同该接口中调用到的接口 |

行号见源码：

|  |  |  |
| --- | --- | --- |
| 行号 | 解释 | Tik接口 |
| 10 | 调用searchForNeighbors接口，搜索查询点的临近点 | 尚未找到该接口的源码 |
| 36 | 调用planeFitting接口 |  |
| 39 | 调用normalDisambiguation接口 |  |
| 50 | 调用searchForNeighbors接口 | 尚未找到该接口的源码 |
| 62~63 | 标量乘法 | vec\_mul |
| 79 | 标量混合四则运算 | vec\_mul、vec\_rec |
| 96 | 向量点乘 | vec\_mul、vec\_add |
| 108 | 调用directedOrthogonalAxis接口 |  |
| 110 | 调用getAngleBetweenUnitVectors接口 |  |
| 112 | 调用std::min、std::floor接口  标量除法 | scalar\_min、vec\_rec、scalar\_conv |
| 142 | 向量点乘 | vec\_mul、vec\_add |
| 158 | 调用directedOrthogonalAxis接口 |  |
| 160、194、239 | 向量叉乘 | vec\_mul、vec\_add、vec\_sub |
| 175、192、219 | 调用directedOrthogonalAxis接口 |  |
| 285~332、351 | 标量混合四则运算 | vec\_mul、vec\_add、vec\_sub、vec\_rec |
| 360、372 | 调用directedOrthogonalAxis接口 |  |
| 377 | 向量叉乘 | vec\_mul、vec\_add、vec\_sub |

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

### 接口功能

抽象特征计算接口。

### 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 | out | output | PointCloudOut |  |  |  |

1. 算子的实现接口定义

pcl::BOARDLocalReferenceFrameEstimation<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

|  |  |
| --- | --- |
| 该接口需要用到的Tik接口 | 同该接口中调用到的接口 |
| 无法用Tik接口实现的部分 | 同该接口中调用到的接口 |

行号见源码：

|  |  |  |
| --- | --- | --- |
| 行号 | 解释 | Tik接口 |
| 21 | 调用computePointLRF接口 |  |