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

pcl::ShapeContext3DEstimation< PointInT, PointNT, PointOutT >算子功能分析说明书

|  |  |  |  |
| --- | --- | --- | --- |
| 拟制 | 赵宇轩 | 日期 | 2021-09-25 |
| 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 |
| 2021-09-25 | 1.0 | 初稿完成 | 赵宇轩 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

目录

[pcl::ShapeContext3DEstimation< PointInT, PointNT, PointOutT >算子功能分析说明书 1](#_Toc84857593)

[修订记录 1](#_Toc84857594)

[目录 1](#_Toc84857595)

[1.1 pcl::ShapeContext3DEstimation< PointInT, PointNT, PointOutT > 2](#_Toc84857596)

[1.1.1 功能介绍 2](#_Toc84857597)

[1.1.2 使用场景介绍 2](#_Toc84857598)

[1.2 init pcl::ShapeContext3DEstimation<PointInT, PointNT, PointOutT>::initCompute() 4](#_Toc84857599)

[1.2.1 接口功能 4](#_Toc84857600)

[1.2.2 接口和IR描述 4](#_Toc84857601)

[1.2.3 (高性能)实现方案 5](#_Toc84857602)

[1.3 bool pcl::ShapeContext3DEstimation<PointInT, PointNT, PointOutT>::computePoint (std::size\_t index, const pcl::PointCloud<PointNT> &normals, float rf[9], std::vector<float> &desc) 7](#_Toc84857603)

[1.3.1 接口功能 7](#_Toc84857604)

[1.3.2 接口和IR描述 7](#_Toc84857605)

[1.3.3 (高性能)实现方案 8](#_Toc84857606)

[1.4 void pcl::ShapeContext3DEstimation<PointInT, PointNT, PointOutT>::computeFeature (PointCloudOut &output) 10](#_Toc84857607)

[1.4.1 接口功能 10](#_Toc84857608)

[1.4.2 接口和IR描述 10](#_Toc84857609)

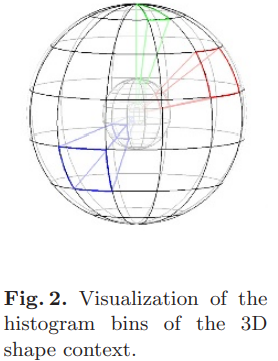
[1.4.3 (高性能)实现方案 10](#_Toc84857610)

## pcl::ShapeContext3DEstimation< PointInT, PointNT, PointOutT >

### 功能介绍

1. 该类实现了3D形状内容描述子(3D shape context descriptor, 3DSC)算法。可用于三维模型的识别、检索等领域。
2. 3DSC的支撑域是一个以基点p为球心，以p的平面法线N为北极构造的一个球体。该支撑域被方位角(azimuth)，俯仰角(elevation)以及经对数处理过的球体径向(radial)平均地分割成1980个bins（源代码中方位角区间划分成12份，俯仰角区间划分成11份，径向区间划分成15份，则特征向量的总长度为12\*11\*15=1980）。（这里的平均指的是按一定的区间长度平均地划分方位角和俯仰角，而不是指bins体积上的平均）。
3. 通过半径集合R={R0...RJ}划分出J+1个径向区间；俯仰角集合Θ={Θ0...ΘK}划分出K+1个俯仰角区间；方位角集合Φ={Φ0...ΦK}划分出L+1个方位角区间。此外，3DSC中的每一个bin都与特征向量(J\*K\*L)中的一个元素对应。

上方公式用于计算径向区间，其中rmin, rmax分别表示预定义的最小半径与最大半径。

1. 俯仰角Θ的取值范围为[0°,180°]；方位角Φ的取值范围为[0°,360°]。
2. 右图形象的展示了bins在空间中是如何进行划分的。
3. Bin(j,k,l)中累加了与点p相关的每一个空间临点pi的权重值w(pi)。其计算公式如下：

其中pi表示bin中的局部点密度，即等于搜索半径σ定义的空间中其他临点的数量。V(j,k,l)表示bin的体积，立方根由经验得出。

1. 该类实现的关键接口
   1. 计算初始化接口

bool pcl::ShapeContext3DEstimation<PointInT, PointNT, PointOutT>::initCompute ()

* 1. 计算点云中某点的描述子

bool pcl::ShapeContext3DEstimation<PointInT, PointNT, PointOutT>::computePoint (

     std::size\_t index, const pcl::PointCloud<PointNT> &normals, float rf[9], std::vector<float> &desc)

* 1. 计算点云的特征

void pcl::ShapeContext3DEstimation<PointInT, PointNT, PointOutT>::computeFeature (PointCloudOut &output)

1. 下图是该类的继承关系。

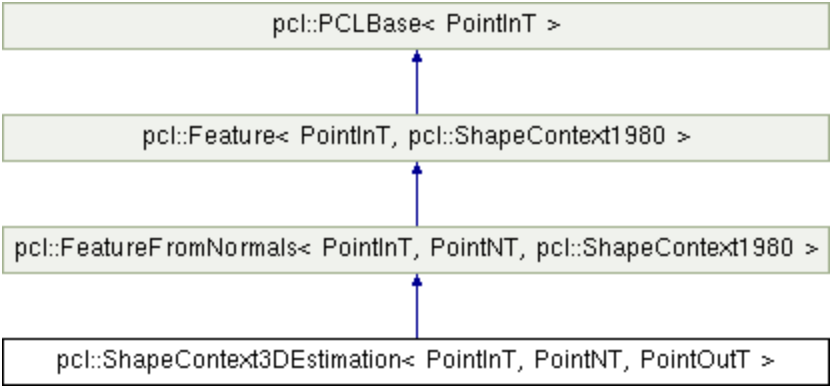


图 1 类的继承关系

### 使用场景介绍

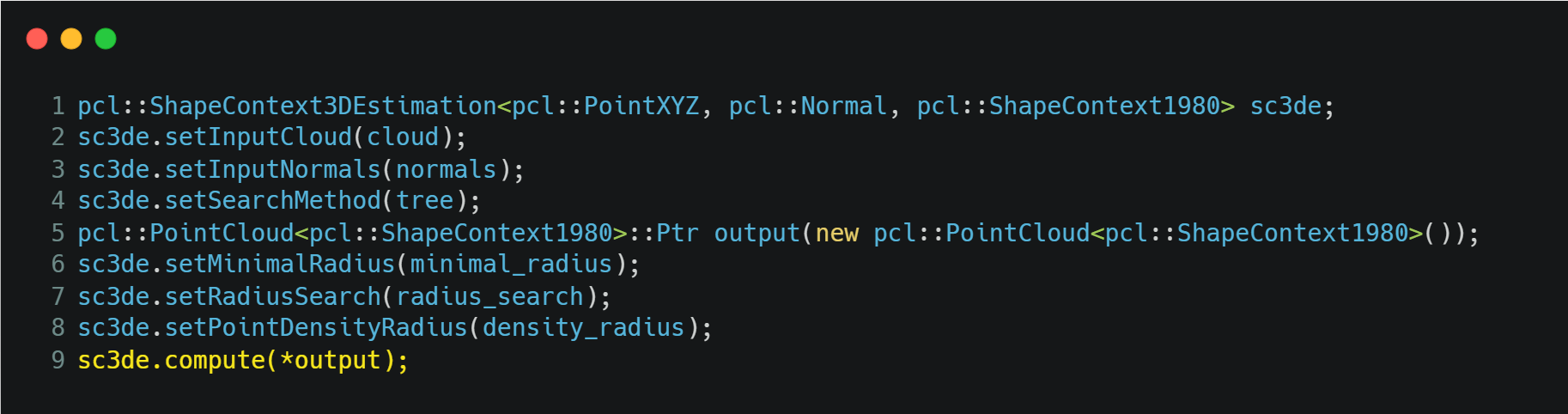


图 2 使用场景

（1）使用场景说明

|  |  |
| --- | --- |
| 行号 | 解释 |
| 1 | 声明一个ShapeContext3DEstimation的实例，记作sc3de |
| 2 | 设置输入的点云，记作cloud |
| 3 | 设置输入的法线，记作，normals |
| 4 | 设置使用kd-tree搜索邻近点 |
| 5 | 声明一个PointCloud的指针，记作output |
| 6~8 | 为sc3de对象设置一系列参数 |
| 9 | 调用compute接口，计算输入点云cloud的特征。其中，compute接口是ShapeContext3DEstimation的父类的父类Feature实现的接口。 |

## init pcl::ShapeContext3DEstimation<PointInT, PointNT, PointOutT>::initCompute()

### 接口功能

该接口对计算查询点的描述子时需要用到的一系列参数做校验及初始化。

### 接口和IR描述

1. IR原型定义

IR定义：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Op** | **Classify** | **Name** | **Type** | **Type Range** | **Default\_value** | **Format** |
| initCompute | Input | search\_radius\_ | double |  | 0 |  |
| Input | min\_radius\_ | double |  | 0.1 |  |
| Input | elevation\_bins\_ | std::size\_t |  | 11 |  |
| Input | azimuth\_bins\_ | std::size\_t |  | 12 |  |
| Input | radius\_bins\_ | std::size\_t |  | 15 |  |
| Input | radii\_interval\_ | std::vector<float> |  | 0 |  |
| Input | phi\_divisions\_ | std::vector<float> |  | 0 |  |
| Input | theta\_divisions\_ | std::vector<float> |  | 0 |  |
| Input | volume\_lut\_ | std::vector<float> |  | 0 |  |
| Output | y1 | bool |  |  |  |

1. 算子的实现接口定义

pcl::ShapeContext3DEstimation<PointInT, PointNT, PointOutT>::initCompute()

该实现接口定义中没有入参

除参数以外需要接口需要使用到的成员变量

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 说明 |
| search\_radius\_ | double | 临近点的搜索半径 |
| min\_radius\_ | double | 搜索的最小半径 |
| elevation\_bins\_ | std::size\_t | 按俯仰角维度划分的bins的数目 |
| azimuth\_bins\_ | std::size\_t | 按转动角维度划分的bins的数目 |
| radius\_bins\_ | std::size\_t | 按径向维度划分的bins的数目 |
| theta\_divisions\_ | std::vector<float> | 按俯仰角维度划分的bins |
| phi\_divisions\_ | std::vector<float> | 按转动角维度划分的bins |
| radii\_interval\_ | std::vector<float> | 按径向维度划分的bins |
| volume\_lut\_ | std::vector<float> | 一个查找表，用于存储按俯仰角、转动角、径向维度划分的bins中的数值。 |

### (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\_adds)，vec\_rec，vec\_exp，vec\_ln，for\_range，set\_as |
| 无法用Tik接口实现的部分 | 三角函数、反三角函数 |

行号见源码附件：

|  |  |  |
| --- | --- | --- |
| 行号 | 解释 | Tik接口 |
| 4~8 | 调用FeatureFromNormals::initCompute()接口 |  |
| 17 | 标量乘法 | vec\_muls |
| 20、21 | 标量除法 | vec\_muls、vec\_rec |
| 30~32 | 对参数radii\_interval\_做初始化，参考论文：  Frome2004\_Chapter\_RecognizingObjectsInRangeDataU的section 2.1的公式(1) | vec\_add、vec\_rec、vec\_mul、vec\_exp、vec\_ln |
| 35~42 | 张量加法 | for\_range、vec\_adds |
| 47 | 张量减法 | vec\_add、vec\_muls（用于deg2rad） |
| 49 | 标量除法 | vec\_rec、vec\_muls |
| 56 | 张量混合四则运算 | vec\_adds  vec\_muls vec\_rec |
| 61 | 标量除法 | vec\_muls(用于deg2rad)  vec\_adds  tik中似乎没有三角函数的接口 |
| 63 | 标量乘法 | vec\_muls |
| 73 | 张量混合四则运输+幂运算 | vec\_muls、vec\_adds vec\_rec |

## bool pcl::ShapeContext3DEstimation<PointInT, PointNT, PointOutT>::computePoint (std::size\_t index, const pcl::PointCloud<PointNT> &normals, float rf[9], std::vector<float> &desc)

### 接口功能

计算点云中某点的描述子。

### 接口和IR描述

1. IR原型定义

IR定义：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Op** | **Classify** | **Name** | **Type** | **Type Range** | **Default\_value** | **Format** |
| computePoint | Input | index | std::size\_t |  |  |  |
| Input | normals | pcl:PointCloud<PointNT> |  |  |  |
| Input | rf | float[9] |  |  |  |
| Input | desc | std::vector<float> |  |  |  |
| Input | indices\_ | PCLBase<PointInT> |  |  |  |
| Input | search\_radius | double |  | 0 |  |
| Input | input\_ | PCLBase<PointInT> |  |  |  |
| Input | surface | PointCloudInConstPtr |  |  |  |
| Input | elevation\_bins\_ | std::size\_t |  | 11 |  |
| Input | radius\_bins\_ | std::size\_t |  | 15 |  |
| Output | desc | std::vector<float> |  |  |  |
| Output | y1 | bool |  |  |  |

1. 算子的实现接口定义

pcl::ShapeContext3DEstimation<PointInT,PointNT,PointOutT>::computePoint(std::size\_t index, const pcl::PointCloud<PointNT> &normals, float rf[9], std::vector<float> &desc)

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 说明 |
| index | std::size | 需要计算描述子的查询点的下标 |
| normals | const pcl::PointClout<PointNT> | 一个指向点云的法向量集合的指针 |
| rf[9] | float | 查询点的参考系 |
| desc | std::vector<float> | 用于存放计算好的描述子 |

除参数以外需要接口需要使用到的成员变量

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 说明 |
| indices\_ | PCLBase<PointInT> | 用于存放点云中所有查询点的下标 |
| search\_radius | double | 搜索临近点的搜索半径 |
| input\_ | PCLBase<PointInT> | 输入点云 |
| surface | PointCloudInConstPtr | 一个指向点云表面的静态指针 |
| elevation\_bins\_ | std::size\_t | 按俯仰角划分的bins的数量 |
| radius\_bins\_ | std::size\_t | 按半径划分的bins的数量 |

### (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，for\_range，set\_as，scalar\_sqrt，scalar\_max、scalar\_min |
| 无法用Tik接口实现的部分 | 三角函数、反三角函数 |

行号见源码附件：

|  |  |  |
| --- | --- | --- |
| 行号 | 解释 | Tik接口 |
| 13 | 调用searchForNeighbors接口，尚未找到该接口的源码 |  |
| 21 | 调用std::min\_element接口 | for\_range、 scalar\_min |
| 41~45 | 标量混合四则运算 | vec\_mul、vec\_add、vec\_rec、vec\_sub |
| 47 | 向量标准化 | vec\_rec |
| 53 | 向量叉乘 | vec\_mul、vec\_sub、vec\_add |
| 65 | 调用std::sqrt接口 | scalar\_sqrt |
| 69 | 调用pcl::geometry::project接口 | vec\_sub、vec\_mul、vec\_add |
| 73 | 向量标准化 | vec\_rec |
| 76 | 向量叉乘 | vec\_mul、vec\_sub  vec\_add |
| 77 | 向量点乘  调用pcl::rad2deg接口  调用std::atan2接口 | vec\_mul、vec\_add  Tik中似乎没有三角函数相关的接口 |
| 78 | 向量点乘、四则运算 | vec\_mul、vec\_sub |
| 81 | 向量标准化 | vec\_mul、vec\_add |
| 83 | 调用pcl::rad2deg接口  调用std::acos接口  调用std::min、std::max接口 | vec\_mul、scalar\_max、scalar\_min  Tik中似乎没有三角函数相关的接口 |
| 86 | 调用std::lower\_bound接口 | for\_range, scalar\_min |
| 91~93 | 调用std::distance接口 | vec\_sub |
| 98 | 调用searchForNeighbour接口 |  |
| 103、112 | 标量混合四则运算 | vec\_rec、vec\_mul、vec\_add |

## void pcl::ShapeContext3DEstimation<PointInT, PointNT, PointOutT>::computeFeature (PointCloudOut &output)

### 接口功能

该接口实现输入点云的描述子的计算。

### 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 | Input | descriptor\_length\_ | std::size\_t |  | 1980 |  |
| Input | is\_dense | bool |  | true |  |
| Input | indices\_ | PCLBase<PointInT> |  |  |  |
| Input | input\_ | PCLBase<PointInT> |  |  |  |
| Output | output | PointCloudOut |  |  |  |

1. 算子的实现接口定义

pcl::ShapeContext3DEstimation<PointInT, PointNT, PointOutT>::computeFeature (PointCloudOut &output)

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

除参数以外需要接口需要使用到的成员变量

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 说明 |
| descriptor\_length\_ | std::size\_t | 描述子的长度 |
| is\_dense | bool | 点云的所有点是否是有限的？  True表示所有点有限，反之False |
| input\_ | PCLBase<PointInT> | 输入的点云 |
| indices\_ | PCLBase<PointInT> | 输入的点云的点的下标集合 |

### (高性能)实现方案

|  |  |
| --- | --- |
| 该接口需要用到的Tik接口 | 同computePoint接口 |
| 无法用Tik接口实现的部分 | 同computePoint接口 |

行号见源码：

|  |  |  |
| --- | --- | --- |
| 行号 | 解释 | Tik接口 |
| 23 | 调用computePoint接口 |  |
| 25 | 调用std::copy接口 |  |