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

pcl::BRISK2DEstimation< PointInT, PointOutT, KeypointT, IntensityT >算子功能分析说明书

|  |  |  |  |
| --- | --- | --- | --- |
| 拟制 |  | 日期 | 2021-10-07 |
| 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 |
| 2021-10-07 | 1.0 | 初稿完成 |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

目录

[pcl::BRISK2DEstimation< PointInT, PointOutT, KeypointT, IntensityT >算子功能分析说明书 1](#_Toc84857204)

[修订记录 1](#_Toc84857205)

[目录 1](#_Toc84857206)

[1.1 pcl::ShapeContext3DEstimation< PointInT, PointNT, PointOutT > 2](#_Toc84857207)

[1.1.1 功能介绍 2](#_Toc84857208)

[1.1.2 使用场景介绍 2](#_Toc84857209)

[1.2 void BRISK2DEstimation<PointInT, PointOutT, KeypointT, IntensityT>::compute (PointCloudOutT &output) 4](#_Toc84857210)

[1.2.1 接口功能 4](#_Toc84857211)

[1.2.2 接口和IR描述 4](#_Toc84857212)

[1.2.3 (高性能)实现方案 4](#_Toc84857213)

[1.3 void BRISK2DEstimation<PointInT, PointOutT, KeypointT, IntensityT>::generateKernel (std::vector<float> &radius\_list, std::vector<int> &number\_list, float d\_max, float d\_min, std::vector<int> index\_change) 6](#_Toc84857214)

[1.3.1 接口功能 6](#_Toc84857215)

[1.3.2 接口和IR描述 6](#_Toc84857216)

[1.3.3 (高性能)实现方案 6](#_Toc84857217)

[1.4 inline int BRISK2DEstimation<PointInT, PointOutT, KeypointT, IntensityT>::smoothedIntensity (const std::vector<unsigned char> &image,int image\_width, int,const std::vector<int> &integral\_image,const float key\_x, const float key\_y, const unsigned int scale,const unsigned int rot, const unsigned int point) 8](#_Toc84857218)

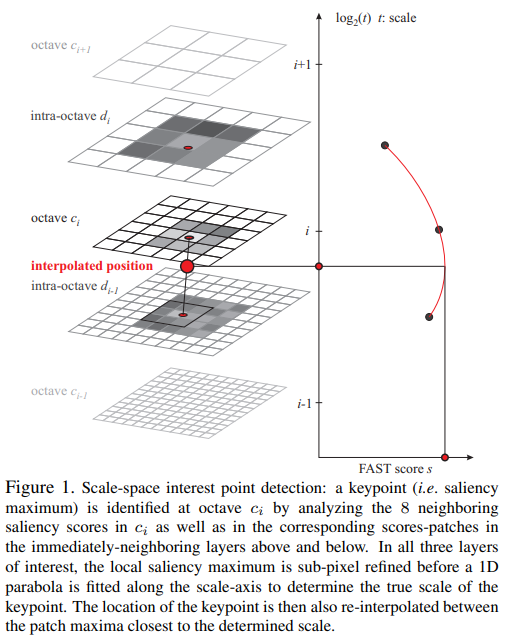
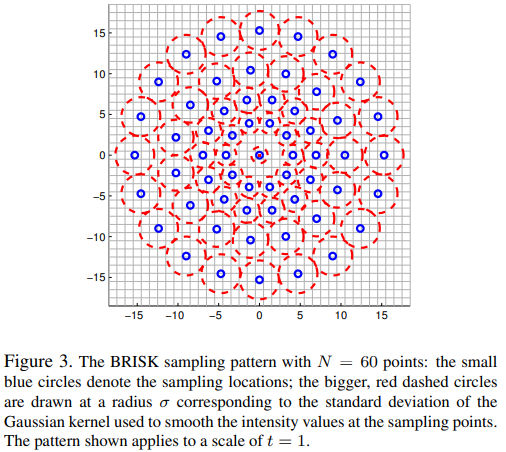
[1.4.1 接口功能 8](#_Toc84857219)

[1.4.2 接口和IR描述 8](#_Toc84857220)

[1.4.3 (高性能)实现方案 9](#_Toc84857221)

## **pcl::BRISK2DEstimation< PointInT, PointOutT, KeypointT, IntensityT >**

### 功能介绍

1. 该类基于源码和文献BRISK: Binary Robust Invariant Scalable Keypoints，实现了BRISK-descriptor的相关算法
2. 文章将BRISK算法分成两大块进行描述，首先是Scale-space keypoint detection，其次是Keypoint description。
3. 尺度空间关键点检测
   1. 在尺度空间中利用FAST的score:s作为显著性的测量指标，并计算连续尺度空间中每一个关键点的真实尺度。
   2. 在BRISK架构中，尺度空间金字塔包含n个octaves层，记作ci。以及n个intra-octaves层，记作di;其中i={0,1,...,n-1}，n=4。如图1所示。
   3. octaves层由原始图像C0通过half-sampling得到。而每一个intra-octave层(即di)则位于ci与ci+1之间。
   4. 特别的，第一个intra-octave(即d0)由原始图像C0通过下采样得到。而其他的intra-octave则皆由d0通过half-sampling得到。
4. 关键点描述
   1. BRISK描述子使用了一个pattern来对关键点的邻域点进行采样。该pattern如图3所示，定义了N个点，平均的分布在关键点的不同半径上的同心圆中。
5. 该类实现的关键接口
   1. 计算输入点集的描述子

BRISK2DEstimation<PointInT, PointOutT, KeypointT, IntensityT>::compute (PointCloudOutT &output)

* 1. 生成内核

BRISK2DEstimation<PointInT, PointOutT, KeypointT, IntensityT>::generateKernel (

     std::vector<float> &radius\_list,

std::vector<int> &number\_list,

float d\_max,

float d\_min,

std::vector<int> index\_change)

* 1. 计算图像中给定 x/y 位置的平滑强度

BRISK2DEstimation<PointInT, PointOutT, KeypointT, IntensityT>::smoothedIntensity (

const std::vector<unsigned char> &image,

int image\_width, int,

const std::vector<int> &integral\_image,

const float key\_x,

const float key\_y,

const unsigned int scale,

const unsigned int rot,

const unsigned int point)

1. 该类属于Feature模块下，但不继承任何类。

### 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BRISKEstimation的实例，记作brisk\_est |
| 2~3 | 设置参数 |
| 4 | 调用compute接口，计算输入点集的描述子。 |

## void BRISK2DEstimation<PointInT, PointOutT, KeypointT, IntensityT>::compute (PointCloudOutT &output)

### 接口功能

该计算输入点集的描述子。

### 接口和IR描述

1. IR原型定义

IR定义：

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

1. 算子的实现接口定义

BRISK2DEstimation<PointInT, PointOutT, KeypointT, IntensityT>::compute (PointCloudOutT &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 | vec\_ln，vec\_rec，vec\_mul，vec\_max，vec\_add，vec\_sub |
| 无法用Tik接口实现的部分 | 三角函数，同接口中调用的其他接口 |

行号见源码附件：

|  |  |  |
| --- | --- | --- |
| 行号 | 解释 | Tik接口 |
| 26 | 调用了std::log2接口 | vec\_ln、vec\_rec、vec\_mul |
| 35~42  50~55 | 标量计算 | vec\_max、vec\_rec、vec\_mul、vec\_add |
| 83~89 | 标量混合四则运算 | vec\_mul、vec\_add、vec\_sub |
| 131 | 调用smoothedIntensity接口 |  |
| 145~146 | 标量除法 | vec\_rec、vec\_mul |
| 150 | 调用std::atan2接口  标量混合四则运算 | Tik接口中无三角函数  vec\_rec、vec\_mul |
| 166、168、170 | 标量混合四则运算 | vec\_mul、vec\_rec、vec\_add |
| 182 | 调用smoothedIntensity接口 |  |

## void BRISK2DEstimation<PointInT, PointOutT, KeypointT, IntensityT>::generateKernel (std::vector<float> &radius\_list, std::vector<int> &number\_list, float d\_max, float d\_min, std::vector<int> index\_change)

### 接口功能

生成内核。

### 接口和IR描述

1. IR原型定义

IR定义：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Op** | **Classify** | **Name** | **Type** | **Type Range** | **Default\_value** | **Format** |
| generateKernel | Input | radius\_list | std::vector<float> |  |  |  |
| Input | number\_list | std::vector<int> |  |  |  |
| Input | d\_max | float |  | 5.85f |  |
| Input | d\_min | float |  | 8.2f |  |
| Input | index\_change | std::vector<int> |  |  |  |

1. 算子的实现接口定义

BRISK2DEstimation<PointInT, PointOutT, KeypointT, IntensityT>::generateKernel (std::vector<float> &radius\_list, std::vector<int> &number\_list, float d\_max, float d\_min, std::vector<int> index\_change)

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 说明 |
| radius\_list | std::vector<float> |  |
| number\_list | std::vector<int> |  |
| d\_max | float |  |
| d\_min | float |  |
| index\_change | std::vector<int> |  |

### (高性能)实现方案

|  |  |
| --- | --- |
| 该接口需要用到的Tik接口 | vec\_ln，vec\_rec，vec\_mul，vec\_add，for\_range，scalar\_conv |
| 无法用Tik接口实现的部分 | 三角函数 |

行号见源码附件：

|  |  |  |
| --- | --- | --- |
| 行号 | 解释 | Tik接口 |
| 21 | 标量运算：对数，除法 | vec\_ln、vec\_rec、vec\_mul |
| 22 | 标量除法 | vec\_rec、vec\_mul |
| 31 | 标量幂运算 | vec\_mul、vec\_add |
| 38、44、51 | 标量四则运算 | vec\_mul、vec\_rec |
| 47、48、53 | 标量运算：四则运算、三角函数 | tik中似乎没有三角函数相关的接口 |
| 79 | 调用std::iota接口 | for\_range、vec\_add |
| 77、88~99 | 标量混合四则运算 | vec\_mul、vec\_add、vec\_rec |
| 114 | 调用std::ceil向上取整 | scalar\_conv |

## inline int BRISK2DEstimation<PointInT, PointOutT, KeypointT, IntensityT>::smoothedIntensity (const std::vector<unsigned char> &image,int image\_width, int,const std::vector<int> &integral\_image,const float key\_x, const float key\_y, const unsigned int scale,const unsigned int rot, const unsigned int point)

### 接口功能

计算图像中给定 x/y 位置的平滑强度。

### 接口和IR描述

1. IR原型定义

IR定义：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Op** | **Classify** | **Name** | **Type** | **Type Range** | **Default\_value** | **Format** |
| smoothedIntensity | Input | image | std::vector<unsigned char> |  |  |  |
| Input | image\_width | int |  |  |  |
| Input | image\_height | int |  |  |  |
| Input | integral\_image | std::vector<int> |  |  |  |
| Input | key\_x | float |  |  |  |
| Input | key\_y | float |  |  |  |
| Input | scale | unsigned int |  |  |  |
| Input | rot | unsigned int |  |  |  |
| Input | point | unsigned int |  |  |  |
| Output | intensity | int |  |  |  |

1. 算子的实现接口定义

BRISK2DEstimation<PointInT, PointOutT, KeypointT, IntensityT>::smoothedIntensity (const std::vector<unsigned char> &image,int image\_width, int,const std::vector<int> &integral\_image,const float key\_x, const float key\_y, const unsigned int scale,const unsigned int rot, const unsigned int point)

|  |  |  |
| --- | --- | --- |
| 参数 | 类型 | 说明 |
| image | std::vector<unsigned char> |  |
| image\_width | int |  |
| image\_height | int |  |
| integral\_image | std::vector<int> |  |
| key\_x | float |  |
| key\_y | float |  |
| scale | unsigned int |  |
| rot | unsigned int |  |
| point | unsigned int |  |

### (高性能)实现方案

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

行号见源码：

|  |  |  |
| --- | --- | --- |
| 行号 | 解释 | Tik接口 |
| 10~214 | 该接口中涉及计算的部分全部是标量的混合四则运算 | vec\_mul、vec\_add、vec\_rec |