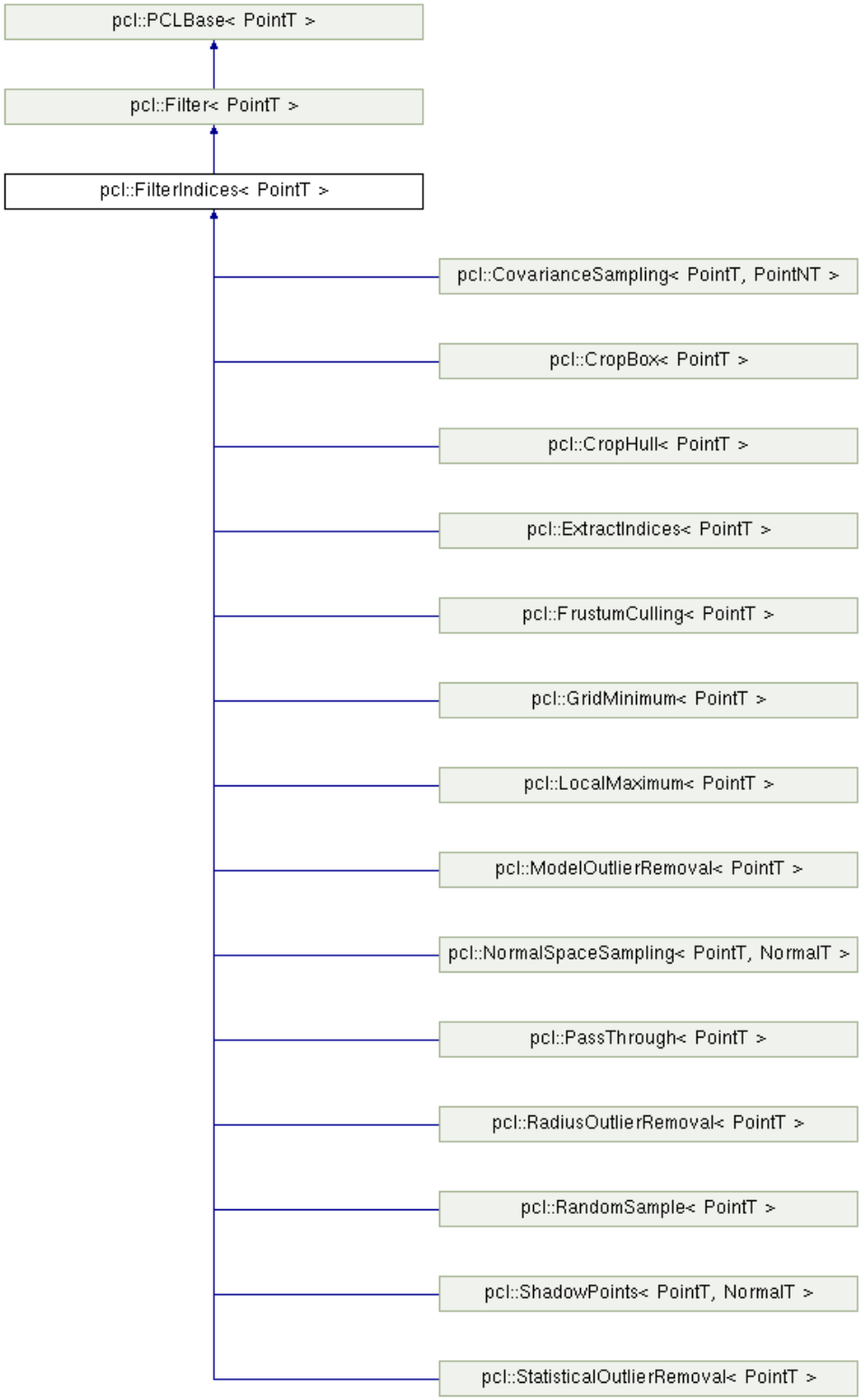
1 “针对性”滤波

点云滤波：旨在滤除非必要的噪声，噪声多表现为散列点，离群点等等。而点云的滤噪主要是为了后续的配准，特征提取等处理。

PCL中主要的处理场景

- 密度不均待平滑
- 离群点
- 减少数据量的降采样

-



1.1 常用滤波器

1.1.1 体素格滤波器 VoxelGrid filter

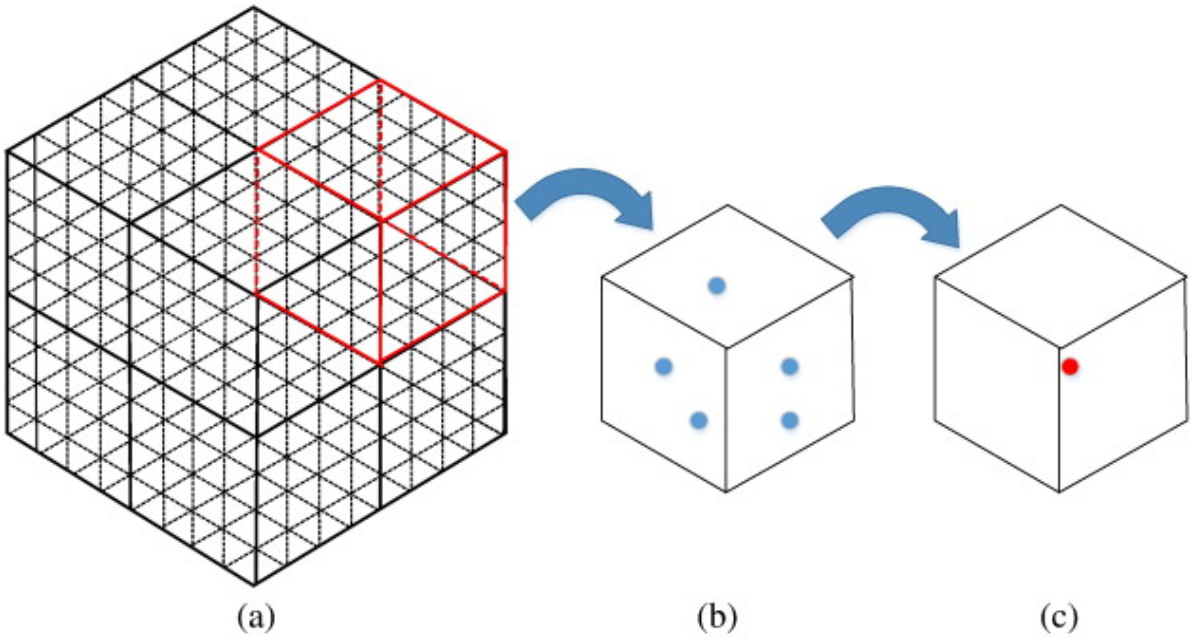
https://blog.csdn.net/qq_36686437/article/details/106628442

特点：在网格内减少点数量保证重心位置不变.

有一个很像的滤波器ApproximateVoxelGrid，用体素内所有点的中心近似表示体素中的其他点，这种计算方式基于哈希函数完成的，对于数据量巨大的场景点云，比VoxelGrid方法计算方式速度快很多。

Voxel意为体素，如果你知道像素一词-----pixel 就能理解体素，像素是构成图像的单位方片。体素就是三维空间的单位立方体。

体素滤波之所以能达到降采样的目的，是把聚集在单位体素里的n个点。求出质心点之后，作为一个代表点，由此数据量大幅减少。



3、VoxelGrid计算过程

1.依据的点云数据坐标集合，求取X、Y、Z三个坐标轴上的最大值X_{max}、Y_{max}、Z_{max}和最小值X_{min}、Y_{min}、Z_{min}。

2. 设置体素小栅格的边长r。

3. 根据X、Y、Z三个坐标轴上的最大、最小值求得点云最小包围盒的边长l_x、l_y、l_z。

$$\begin{cases} l_x = x_{\max} - x_{\min} \\ l_y = y_{\max} - y_{\min} \\ l_z = z_{\max} - z_{\min} \end{cases} \tag{1}$$

4.计算体素网格的尺寸。

$$\begin{cases} D_x = \lfloor l_x / r \rfloor \\ D_y = \lfloor l_y / r \rfloor \\ D_z = \lfloor l_z / r \rfloor \end{cases} \tag{2}$$

式中，$\lfloor\ \rfloor$表示向下取整。

5.计算点云中每一个在体素小栅格内的索引h。

$$\begin{cases} h_x = \lfloor (x - x_{\min}) / r \rfloor \\ h_y = \lfloor (y - y_{\min}) / r \rfloor \\ h_z = \lfloor (z - z_{\min}) / r \rfloor \\ h = h_x + h_y * D_x + h_z * D_x * D_y \end{cases} \tag{3}$$

6.将h里的元素按照从小到大的顺序进行排序，计算每个体素小栅格重心，以重心代替小栅格内的所有点。

体素滤波的一种改进

https://blog.csdn.net/qq_36686437/article/details/113529410

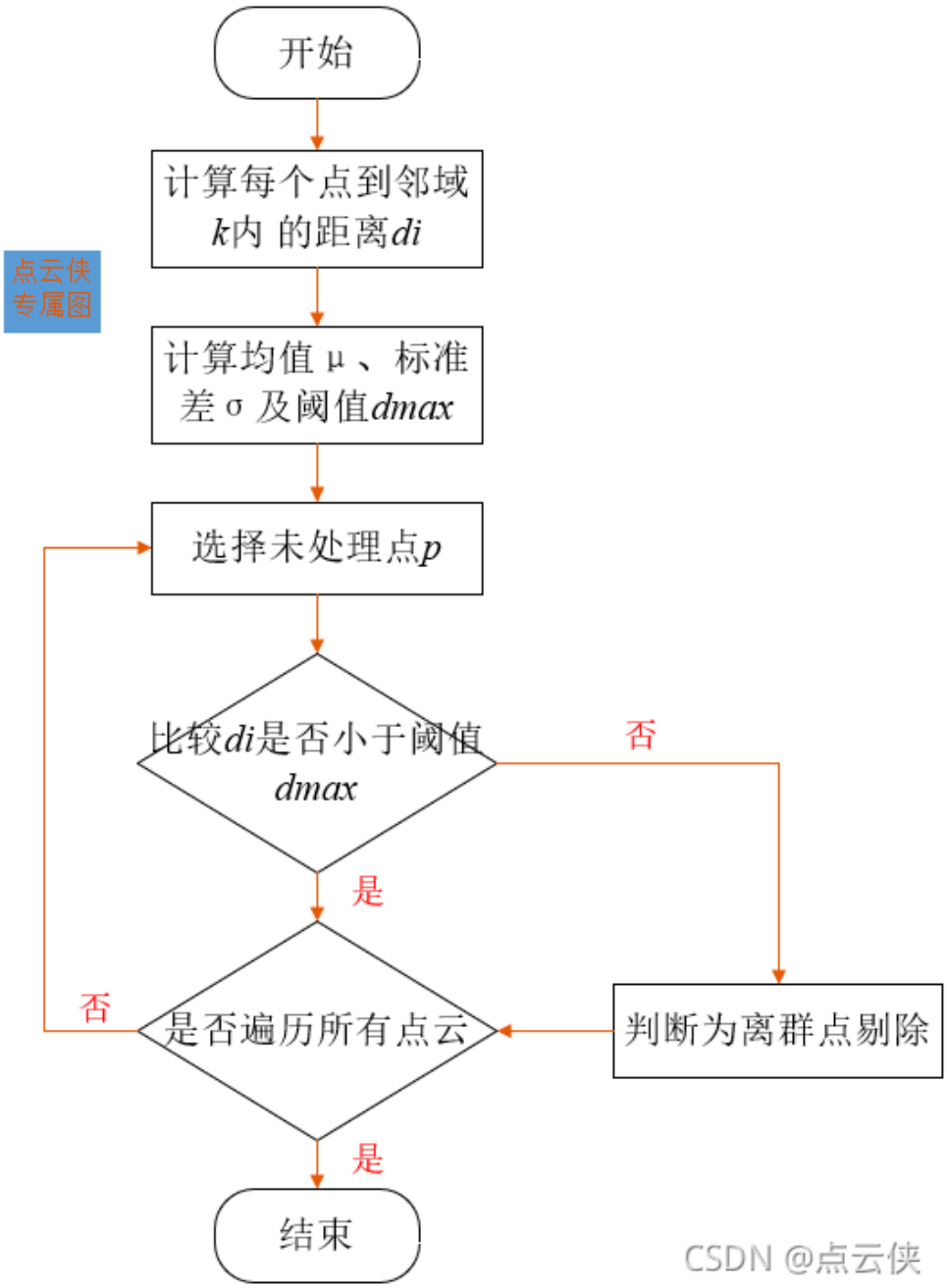
PCL中的VoxelGrid类通过输入的点云数据创建一个三维体素栅格，用重心来近似表示体素中的其他点，这样该体素内所有点都用一个重心点来表示，但是该重心点不一定是原始点云中的点，会失去原始点云的细小特征，这时，可以使用原始点云中最接近重心位置的那个点替代，可以提高点云数据的表达准确性。

PCL最小点数约束的VoxelGrid体素滤波

https://blog.csdn.net/qq_36686437/article/details/120515571

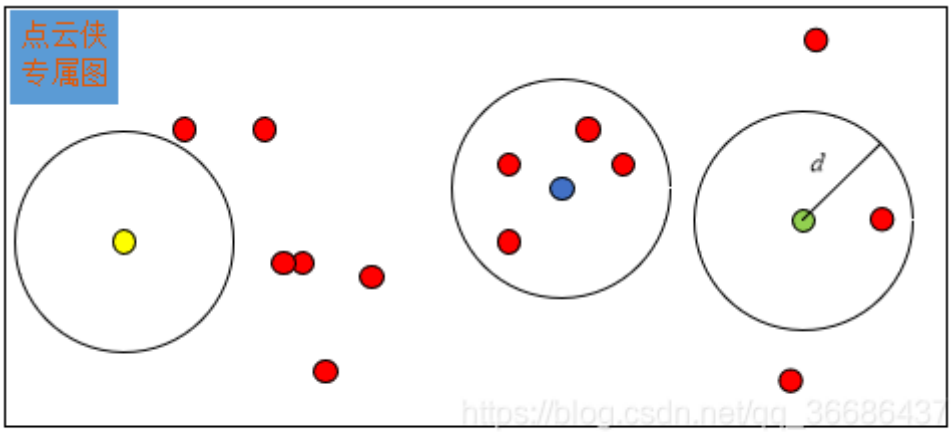
1.1.2 统计滤波器 pcl::StatisticalOutlierRemoval

激光扫描会产生密度不均匀的点云数据集，另外测量中的误差也会产生稀疏的离群点，此时，估计局部点云特征（例如采样点处法向量或曲率变化率）时运算复杂，这会导致错误的数值，反过来就会导致点云配准等后期的处理失败。**统计滤波器用于去除明显的离群点**，离群点往往由测量噪声引入，其特征是在空间中分布稀疏，可以理解为：每个点都表达一定信息量，某个区域点越密集则可能信息量越大。噪声信息属于无用信息，信息量较小，所以离群点没啥信息。定义如果某处点云小于一定密度阈值，则视为无效。以每个点与其邻近的 **K** 个点的平均距离作为密度度量。计算每个点到其最近的k个点平均距离。如果这个平均距离超过一定的阈值就把他剔除



1.1.3 半径滤波器

半径滤波器的滤波思想就是在点云数据中，设定每个点的一定半径范围内周围至少有足够多的近邻，不满足就会被删除，假如指定了一个半径d，并且指定了半径内至少有1个邻居，那么只有黄色点被删除，但是如果指定了半径内至少有2个邻居，那么黄色和绿色的点都会被删除



去除离群点原理图

2、实现流程

1. 点云数据构造K-D 树，建立点云拓扑关系。
2. 求点云中任意一点邻域范围内邻近点个数。
3. 判断邻近点个数是否小于判定阈值，若小于则认为该点为噪声点并去除。
4. 重复上述步骤，直至点云中所有点都处理完毕。

半径滤波器域统计滤波器相比更加简答粗暴，算法运行速度快，依序迭代留下的点一定是最密集的，但是球的半径和球内的点数需要人工指定。比较适合去除单个离群点，可以把边界做一个比较好的保留。

https://blog.csdn.net/qq_36686437/article/details/113730403

1.1.4 条件滤波器

顾名思义，满足条件的点被保留，反之剔除。

条件滤波器过滤满足特定条件的数据，可以一次删除满足对输入的点云设定的一个或多个条件指标的所有的数据点，删除点云中不符合用户指定的一个或者多个条件的数据点，但是用户必须提供条件

https://blog.csdn.net/qq_36686437/article/details/114025259

```
//创建条件定义对象
pcl::ConditionAnd<pcl::PointXYZ>::Ptr range_cond(new
pcl::ConditionAnd<pcl::PointXYZ>());
//为条件定义对象添加比较算子
range_cond->addComparison(pcl::FieldComparison<pcl::PointXYZ>::ConstPtr(new
pcl::FieldComparison<pcl::PointXYZ>("z", pcl::ComparisonOps::GT, 0.0)));
//添加在Z字段上大于（pcl::ComparisonOps::GT  great Then）0的比较算子
range_cond->addComparison(pcl::FieldComparison<pcl::PointXYZ>::ConstPtr(new
pcl::FieldComparison<pcl::PointXYZ>("z", pcl::ComparisonOps::LT, 0.8)));
//添加在Z字段上小于（pcl::ComparisonOps::LT  Lower Then）0.8的比较算子
```

1.1.5 模型滤波器

ModelOutlierRemoval 是基于模型和点之间的距离过滤 点云 中的噪点。对整个输入迭代一次，自动过滤非有限点和;setSampleConsensusModelPointer() 指定的模型之外的点，以及setThreholdFunctionPointer() 指定的阈值。

1.1.6 PCL投影滤波器

https://blog.csdn.net/qq_36686437/article/details/120578456

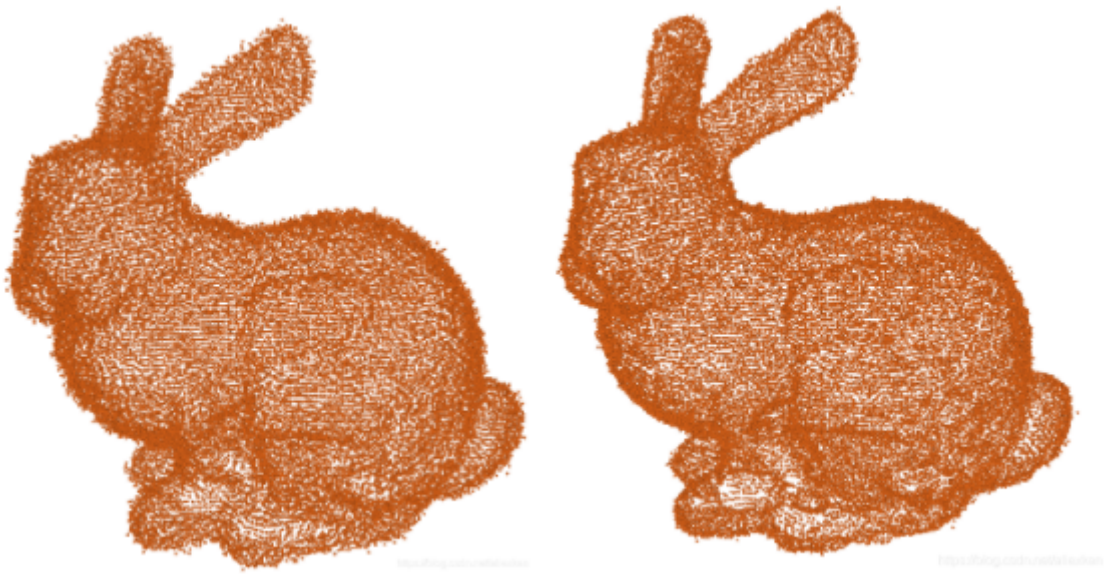
投影参数	对应描述
SACMODEL_PLANE	用于确定平面模型。平面的四个系数是它的Hessian Normal形式:
SACMODEL_LINE	用于确定线模型。直线的六个系数由直线上的一个点和直线的方向给出
SACMODEL_CIRCLE2D	用于确定平面上的二维圆。圆的三个系数由圆心和半径给出
SACMODEL_CIRCLE3D	用于确定平面上的三维圆。圆的七个系数由圆心、半径和法线给出
SACMODEL_SPHERE	用于确定球体模型。球体的四个系数由其三维中心和半径给出
SACMODEL_CYLINDER	用于确定圆柱模型。圆柱的七个系数由其轴上的点、轴方向和半径给出
SACMODEL_CONE	用于确定圆锥模型。圆锥的七个系数由其顶点的一个点、轴的方向和开口的角度给出
SACMODEL_TORUS	圆环，未实现
SACMODEL_PARALLEL_LINE	在规定的最大角度偏差范围内，确定与给定轴平行的线的一种模型。线系数类似于SACMODEL_LINE。
SACMODEL_PERPENDICULAR_PLANE	在指定的最大角度偏差范围内，确定垂直于用户指定轴的平面的模型。平面系数类似于SACMODEL_PLANE。
SACMODEL_PARALLEL_LINES	未实现
SACMODEL_NORMAL_PLANE	使用附加约束来确定平面模型的一种模型:在最大指定的角度偏差范围内，每个内点的表面法线必须与输出平面的表面法线平行。平面系数类似于SACMODEL_PLANE。
SACMODEL_NORMAL_SPHERE	类似于SACMODEL_SPHERE，但是附加了表面法线约束。
SACMODEL_PARALLEL_PLANE	在指定的最大角度偏差范围内，确定平行于用户指定轴的平面的模型。平面系数类似于SACMODEL_PLANE。
SACMODEL_NORMAL_PARALLEL_PLANE	定义使用附加表面法向约束的3D平面分割模型。平面法线必须与用户指定的轴平行。因此，SACMODEL_NORMAL_PARALLEL_PLANE等价于SACMODEL_NORMAL_PLANE + sacmodel_orthicular_plane。平面系数类似于SACMODEL_PLANE。
SACMODEL_STICK	一种三维棒分割模型。STICK是用户指定最小/最大宽度的一条线。

当前三维卷计算量太大且发展不是特别成熟，点云投影到二维平面，能够借助图像算法进行处理，当然会损失一点点云信息，如果多个维度进行投影，能够将损失的信息降低到比较低的水平，目前点云投影应用最多的一种投影方式就是生成俯视图。

1.1.7 高斯滤波

一般把异常值映射到曲面上，其实类似于3通道图片的高斯滤波，将图片像素点的颜色想象成在三维空间中的坐标。

https://blog.csdn.net/aliexken/article/details/107932964



适用于呈正态分布的数据。考虑到离群点的特征，则可以定义某处点云小于某个密度，既点云无效。计算每个点到其最近的k个点平均距离。则点云中所有点的距离应构成高斯分布。给定均值与方差，可剔除3∑之外的点。

1.1.8 双边滤波器 BilaternlFilter

https://blog.csdn.net/qq_36686437/article/details/115188001

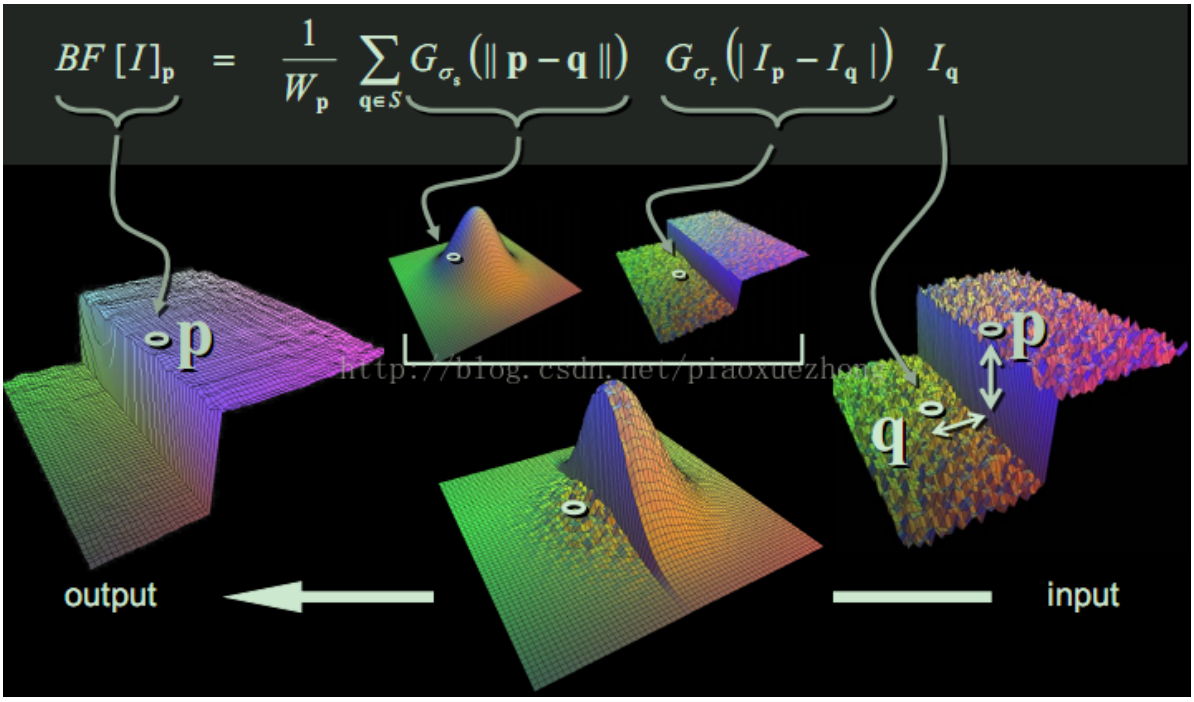
双边滤波主要作用是具有保边的功能，即在滤波的过程中不会连带边界一起都平滑掉，这样有利于计算准确的法线。

类 BilaternlFilter 是对双边滤波算法在点云上的实现， 该类的实现利用的并非XYZ 字段的数据进行，而是利用**强度**数据字段进行双边滤波算法的实现，所以在使用该类时点云的类型中字段必须有强度字段，否则无法进行双边滤波处理。

双边滤波算法，是通过取邻近采样点的加权平均来修正当前采样点的位置，从而达到滤波效果；同时也会有选择地剔除部分于当前采样点“差异”较大的相邻采样点，从而达到保持原特征的目的。

1. 在图像的平坦区域，像素值变化很小，对应的像素范围域权重接近于1，此时空间域权重起主要作用，相当于进行高斯模糊；
2. 在图像的边缘区域，像素值变化很大，像素范围域权重变大，从而保持了边缘的信息。

两个权重对图像的影响：



1.1.9 有序点云的中值滤波

有序点云：有组织的点云

pcl::MedianFilter：中值滤波器是一种最简单和广泛应用的图像处理滤波器。当用在 点云 上时，通常是在一个窗口中对点云数据进行扫描，把窗口内的数据点按其某一个坐标方向值(例如：Z值)进行升序或降序排序，把排序后中间数据点的方向值作为窗口输出时的对应方向坐标。中值滤波法采用各数据点的统计中值，对于消除数据毛刺，效果较好，但对彼此靠近的混杂点噪声滤除效果不好。
注意：该算法只过滤有序的和未转换的点云(即摄像机坐标)的深度(Z分量)。如果将一个无序的点云提供给类实例，将输出错误。

```
#include <pcl/filters/median_filter.h>

using namespace std;

int main()
{
    // 创建以下点云
      /* 1   2   3   4   5
       * 6   7   8   9   10
       * 10  9   8   7   6
       * 5   4   3   2   1
       * 100 100 500 100 100
       */
    pcl::PointCloud<pcl::PointXYZ> cloud_manual;
    cloud_manual.height = 5;
    cloud_manual.width = 5;
    cloud_manual.is_dense = false;
    cloud_manual.resize(5 * 5);

    for (size_t i = 0; i < 5; ++i)
    {
        cloud_manual(i, 0).z = static_cast<float> (i + 1);
        cloud_manual(i, 1).z = static_cast<float> (i + 6);
        cloud_manual(i, 2).z = static_cast<float> (10 - i);
        cloud_manual(i, 3).z = static_cast<float> (5 - i);
        cloud_manual(i, 4).z = static_cast<float> (100);
    }
    cloud_manual(2, 4).z = 500;
```

```
    pcl::MedianFilter<pcl::PointXYZ> median_filter;
    median_filter.setInputCloud(cloud_manual.makeShared());
    median_filter.setWindowSize(3);

    pcl::PointCloud<pcl::PointXYZ> out_1;
    median_filter.filter(out_1);

    // 结果应该是这样的
    /* 6   6   7   8   9
     * 7   7   7   7   7
     * 7   7   7   7   7
     * 10  9   8   7   7
     * 100 100 500 100 100
     */

}
```

1.1.10 移除边缘不连续点

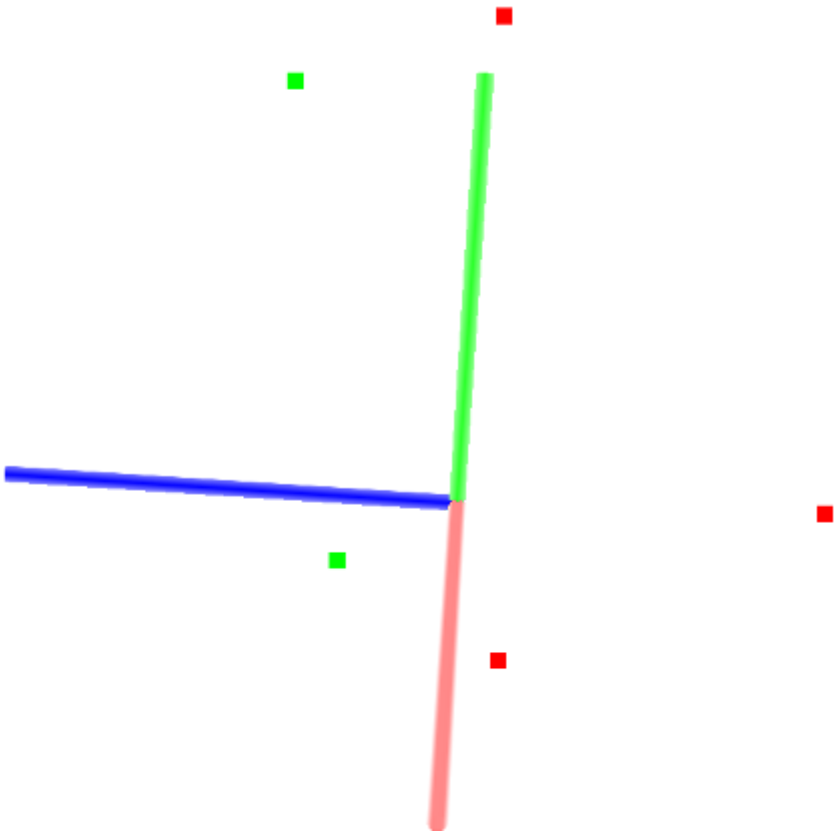
https://blog.csdn.net/qq_36686437/article/details/116722385

1.1.11 直通滤波器 pcl::PassThrough

直接指定保留哪个轴上的范围内的点

这个非常粗暴，划分好 X , Y, Z 方向， 给定范围值，就直接提出了不想要的点。慎用。

关键在于确定范围，必须要知道自己不想要的范围区间，在不知道的情况下也许还得先计算个包围盒。

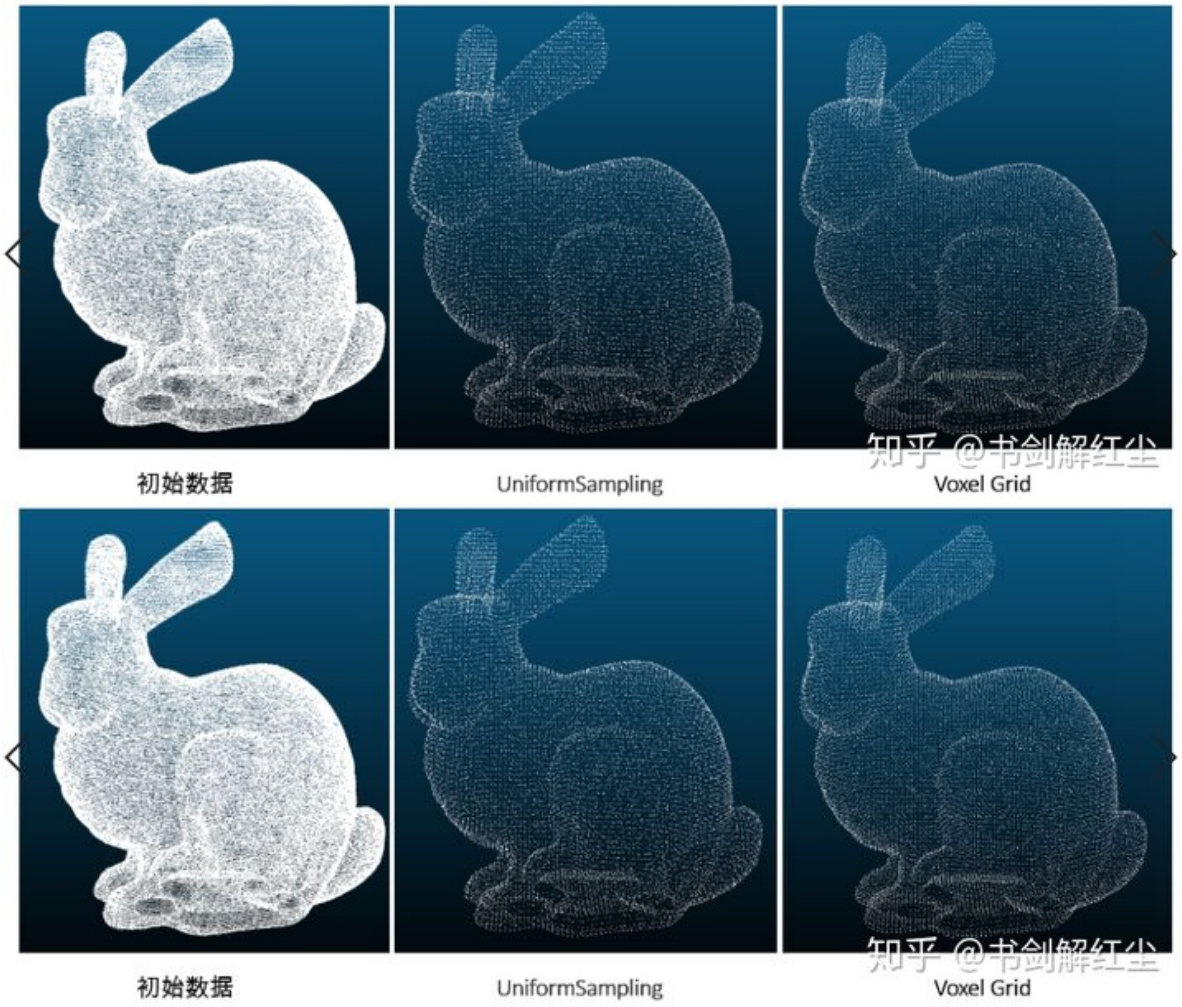


1.2采样滤波

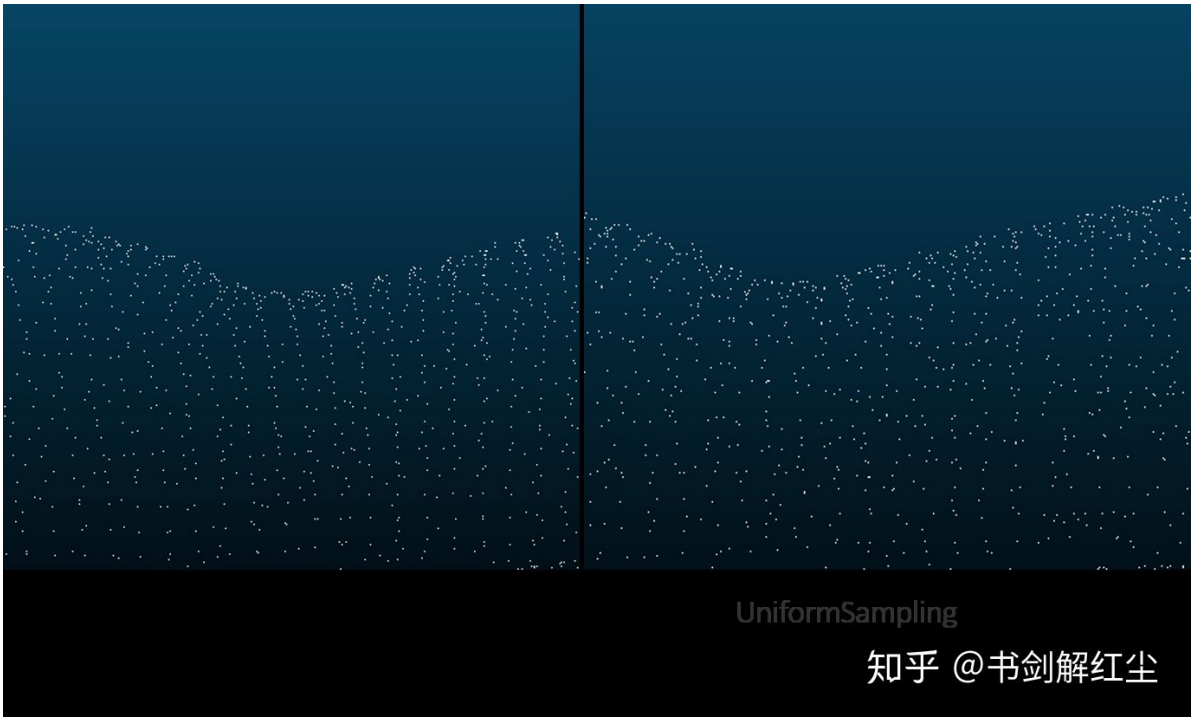
1.2.1 均匀采样 pcl::UniformSampling

原理和体素滤波是很相似的，但是体素采用单位立方体，而均匀采样取一定半径 r 的球体内的点离中心最近的点。

看一下对比：



可能不是特别突出，看一下细节：



栅格采样相比之下采出来分布会相对规律一些。

> 在使用CloudViewer类中实现可视化多个点云的过程时发现了一个问题：CloudViewer类是一个简单直接对简单点云进行可视化的类，比较简单，但是这个类不能用于多线程应用中，如果要在多线程中使用可视化，可以使用PCLVisualizer

均匀采样通过构建指定半径的球体对点云进行下采样滤波，将每一个球内距离球体中心最近的点作为下采样之后的点输出。

1.2.2 随机采样

随机抽样一致算法RANSAC

PCL 中以随机采样一致性算法(RANSAC) 为核心，实现了五种类似于RANSAC的随机参数估计算法，例如随机采样一致性估计（RANSAC) 、最大似然一致性估计 （MLESAC) 、最小中值方差一致性估计 (LMEDS)等，所有的估计参数算法都符合一致性准则。利用RANSAC可以实现点云分割，目前 PCL 中支持的几何模型分割有 空间平面、直线、二维或三维圆、圆球、锥体等 。

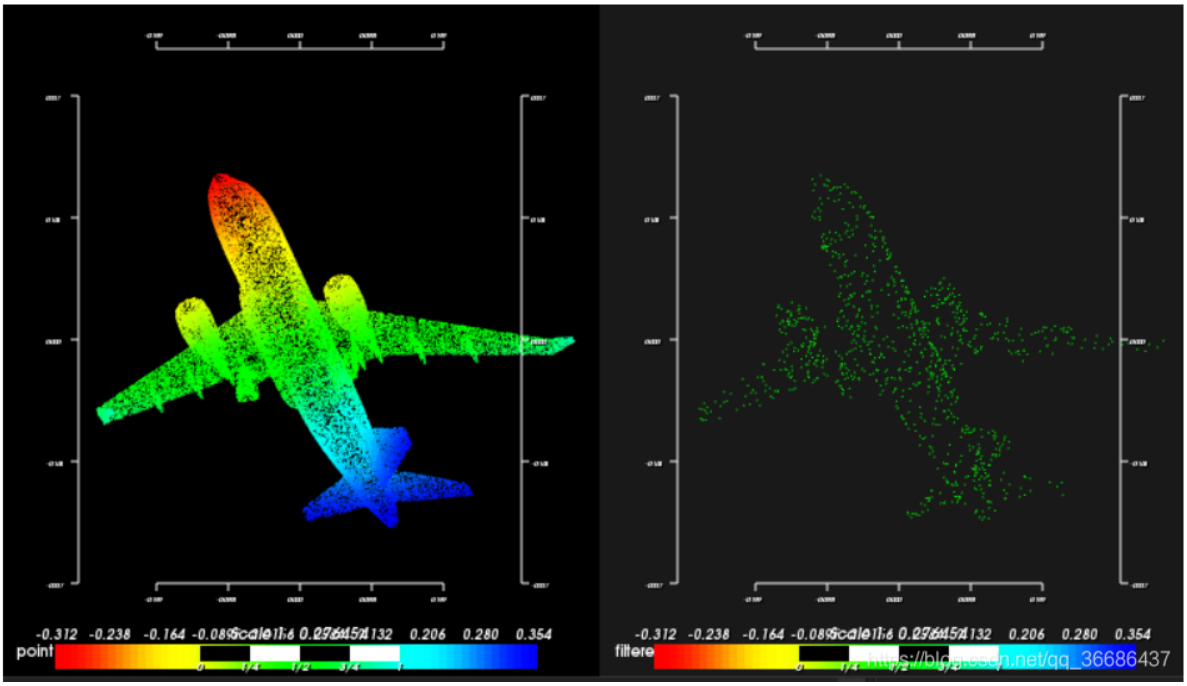
RANSAC从样本中随机抽选出一个样本子集，使用最小方差估计算法对这个子集计算模型参数，然后计算所有样本与该模型的偏差，再使用一个预先设定好的阈值与偏差比较，当偏差小于阈值时，该样本点属于模型内样本点 (inliers），或称内部点、局内点或内点，否则为模型外样本点（outliers），或称外部点、局外点或外点，记录下当前的 inliers 的个数，然后重复这一过程。每一次重复都记录当前最佳的模型参数，所谓最佳即是inliers的个数最多 ，此时对应的inliers个数为 best_ninliers 。 每次迭代的末尾都会根据期望的误差率、 best_ninliers、总样本个数、当前迭代次数，计算一 个迭代结束评判因子，据此决定是否迭代结束。迭代结束后，最佳模型参数就是最终的模型参数估计值 。

RANSAC理论上可以剔除outliers的影响，并得到全局最优的参数估计。但是RANSAC 有两个问题，首先在每次迭代中都要区分 inliers 和 outlieres，因此需要事先设定阈值，当模型具有明显的物理意义时，这个阈值还比较容易设定，但是若模型比较抽象时，阈值就不那么容易设定了。而且固定阈值不适用于样本动态变化的应用；第二个问题是，RANSAC的迭代次数是运行期决定的，不能预知迭代的确切次数（当然迭代次数的范围是可以预测的）。除此之外， RANSAC 只能从一个特定数据集中估计一个模型，当两个（或者更多个）模型存在时，RANSAC 同时找到多个模型。

1.2.3 法线空间采样 NormalSpaceSampling

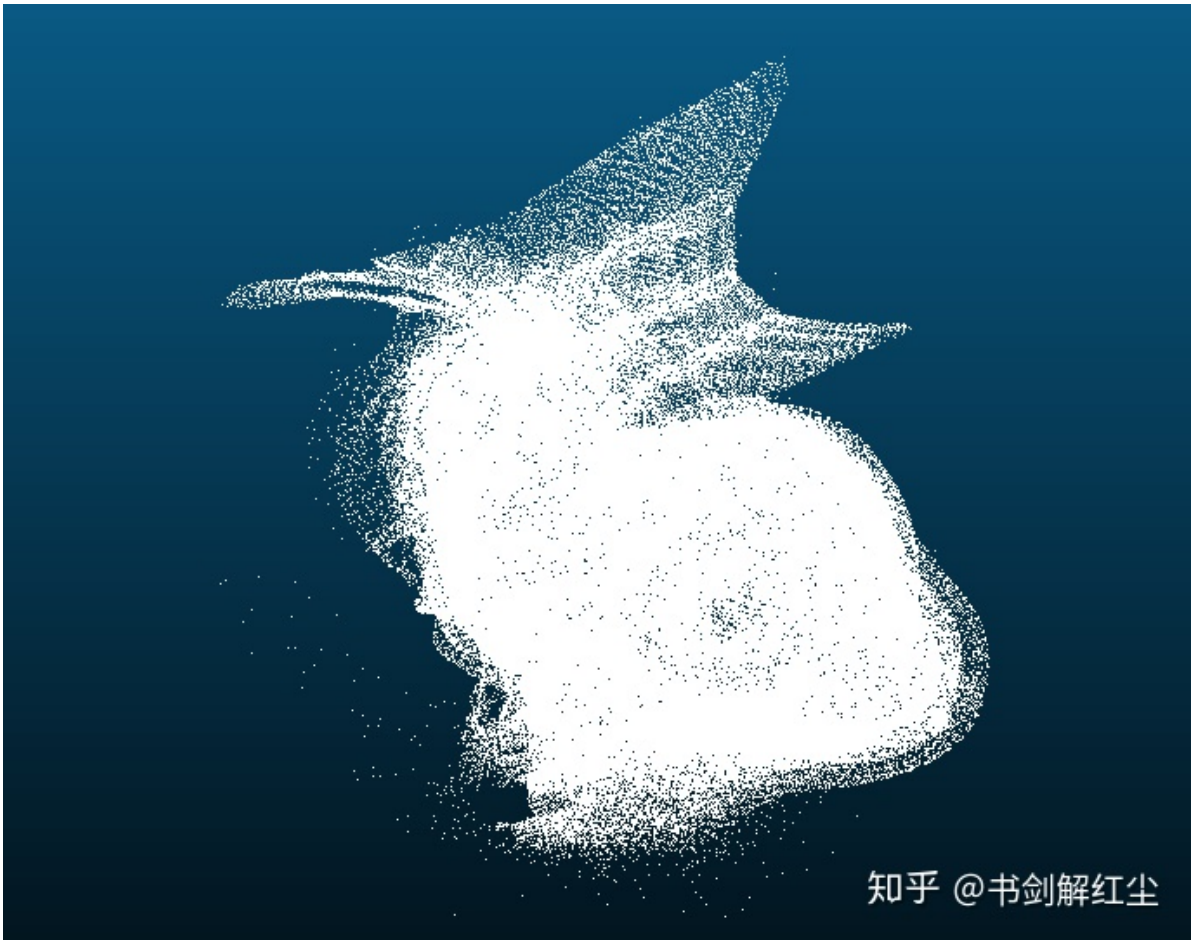
法线空间采样在法向量空间内均匀抽样，使得所选点之间的法线分布尽可能大，表现为特征变化大的地方剩余点较多，变化小的地方剩余点少，可以有效保持物体特性。

这种策略的动机是观察到，对于某些场景(如“切割平面”数据集)，模型的小特征对于确定正确的对齐至关重要。像随机抽样这样的策略通常只会在这些特征中选择几个样本，这导致某些部分无法确定正确刚体转换。因此，一种提高存在足够约束以确定转换所有组件的可能性的方法是根据角度空间中的法线位置来存储点，然后尽可能均匀地在这些存储点上采样。因此，法线空间采样是使用地表特征进行对齐的一个非常简单的例子;与传统的基于特征的方法相比，该方法具有较低的计算成本，但鲁棒性较差。



1.2.4 不算滤波的增采样setUpsamplingMethod

假如我们的原始数据太少，想要获取多一些的点的信息贡献于后面的处理。就要用到增采样（也叫上采样）。本质应该是一个插值的过程。值得提醒的是这种带着“猜想”性质的采样结果未必准确。比如。。。。



1.3 裁剪滤波

CropHull任意多边形内部点云提取

CropBox过滤提取给定立方体内的点云数据

平面裁剪器PlaneClipper3D的使用

LocalMaximum消除局部最大的点

GridMinimum获取栅格最低点

2.点云配准

https://www.sohu.com/a/321034987_715754

2.1基于对应3D匹配计算变换矩阵

点云配准中确定两点云中多个同名点对可实现点云的配准，具体的，三维点云构造点几何邻域特征或深度学习特征，进而进行特征匹配获得最优同名点对，最后计算对应点对间最优变换矩阵。

3 特征描述与提取features

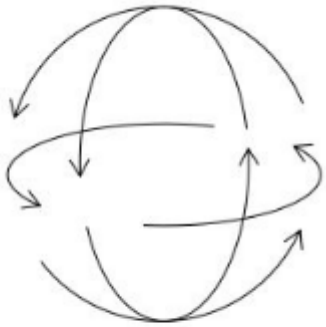
3D 点云特征描述与提取是点云信息处理中的最基础也是最关键的部分，点云的识别、分割、重采样、配准、曲面重建等大部分算法，都十分依赖特征描述与提取的结果。

从尺度上来划分，一般分为局部特征描述和全局特征描述。例如局部的法线等几何形状特征的描述，全局的拓扑特征描述，都属于3D点云特征描述与提取范畴。在 PCL 中，目前已有很多基本的特征描述子与提取算法。

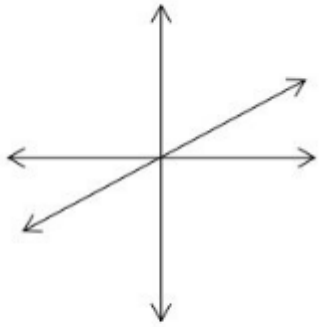
陀螺仪可以用来 检测人在空间中的姿态和朝向

3dof是指有3个转动角度的自由度，而 6dof 是指，除了3个转动角度外，再加上 上下、前后、左右等3个位置相关的自由度。

当我们说 3dof的VR眼镜或VR设备时，是指该VR设备可以检测到头部向不同方向的自由转动，但是不能检测到头部的前后左右的空间位移。而6dof的VR设备（眼镜），则除了检测头部的转动带来的视野角度变化外，还能够检测到由于身体移动带来的上下前后左右位移的变化。



角度



位置

用一张更为形象的图解释就容易理解了：

3DoF头显

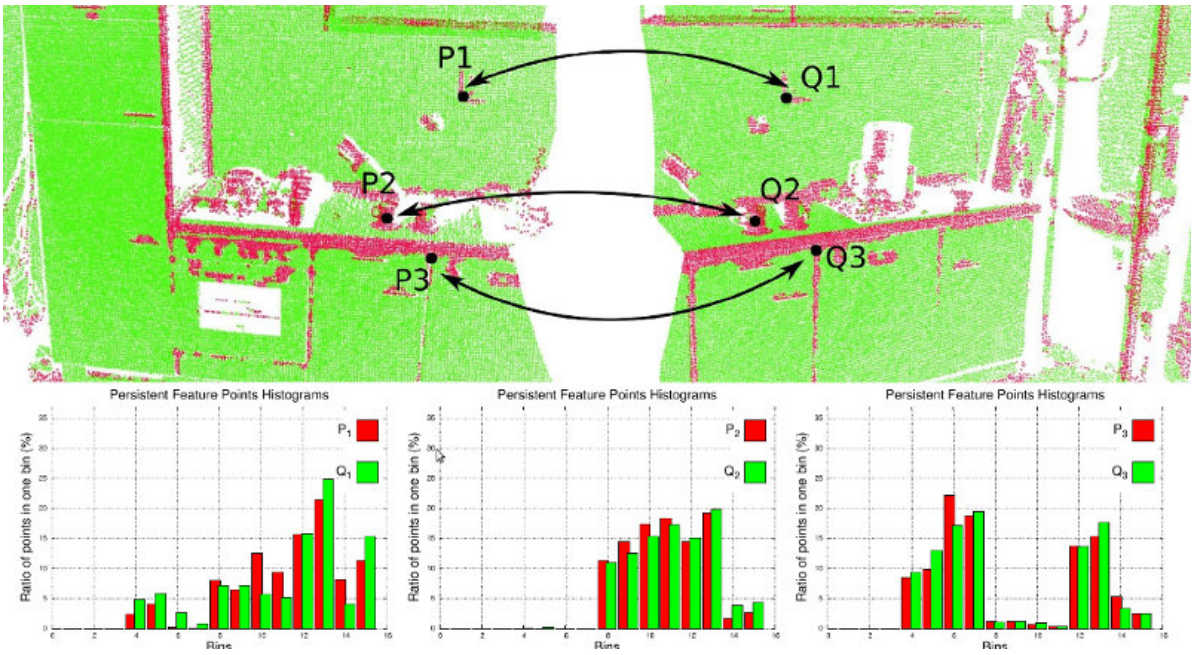
6DoF头显

点云密度是数据分辨率的指标：较高的密度意味着更多的信息（高分辨率），而较低的密度意味着较少的信息（低分辨率）。

3.1 特征点

理想情况下，在使用同一种度量规则情况下，相同或相似的表面上的点的特征值应该非常相似，不同表面上的点的特征描述子有明显的差异。通过以下几个条件的变化仍能够获取获取相同或相似的局部表面特征，则说明该特征表示方式比较优秀：

- **刚性变换(rigid transformations)**：即数据中的3D旋转和3D平移不应影响结果特征向量F的估计
- **多种采样密度(varying sampling density)**：原则上，在一个局部表面或多或少采样密度的应具有相同的特征向量
- **噪声(noise)**：在数据中存在轻微噪声的情况下，由特征点描述的特征向量必须相同或非常相近



2、关键点提取

- PCL ISS关键点提取
- PCL Harris3D关键点提取
- PCL 3D-SIFT关键点检测(Z方向梯度约束)
- PCL 3D-SIFT关键点检测(曲率不变特征约束)
- PCL 3D-SIFT关键点检测(RGB颜色特征约束)

3.1.1 ISS（内部形状描述子）关键点提取

是现有非尺度不变的关键点提取算子中效果较好较稳定的，