

三维点云处理技术 和深度学习在点云处理中的应用

点云识别与跟踪特征描述

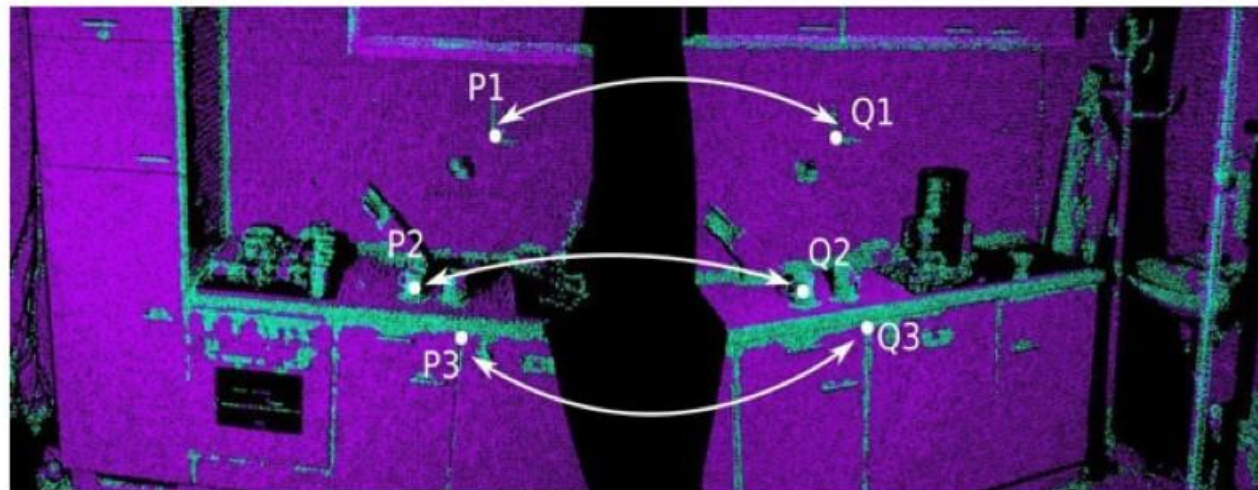
索传哲



内容概要

点云关键点检测
点云常用特征描述

点云关键点检测



什么是关键点(keypoint):

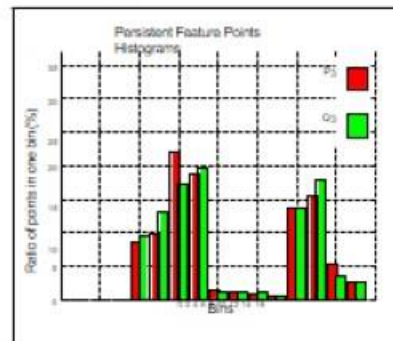
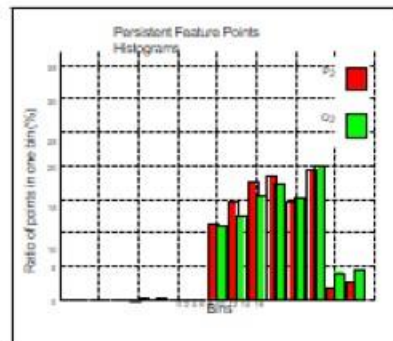
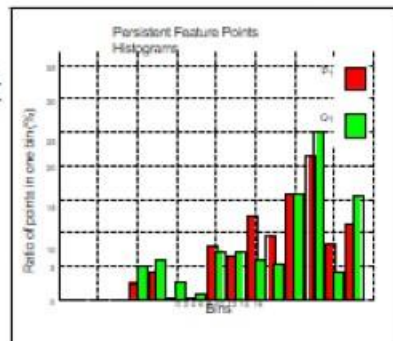
- ▶ 关键点 (或兴趣点) 由关键点检测器确定的具有特定相关关系的点

关键点的性质:

- ▶ 稀疏性: 只有小部分点作为关键点
- ▶ 独特性: 关键点附近应该具有独特的结构或特性可以作为关键点提取器的特征
- ▶ 可重复性(稳定性): 结构相似的点云结构的关键点应该是可重复的(稳定的)

关键点的好处:

- ▶ 提取特征时关键点具有代表性, 减少运算量
- ▶ 搜索匹配时降低噪声的影响, 减少匹配误差

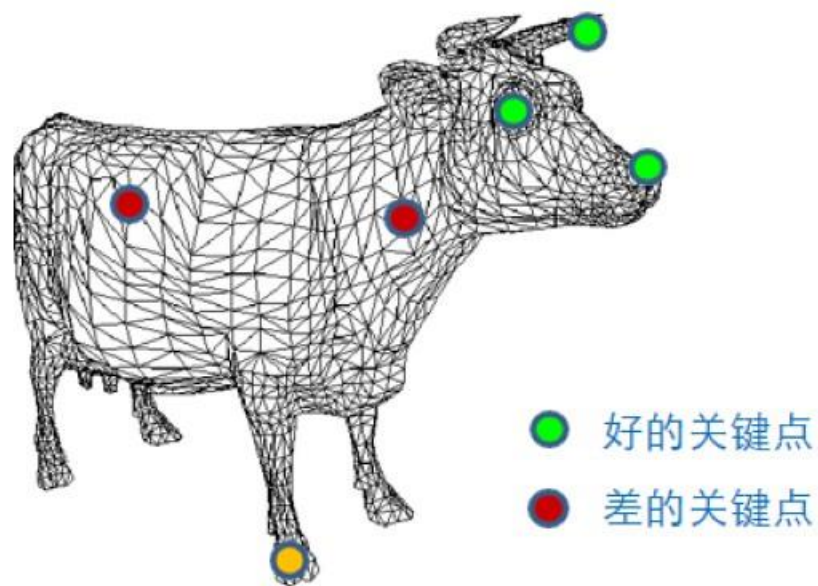


点云关键点检测

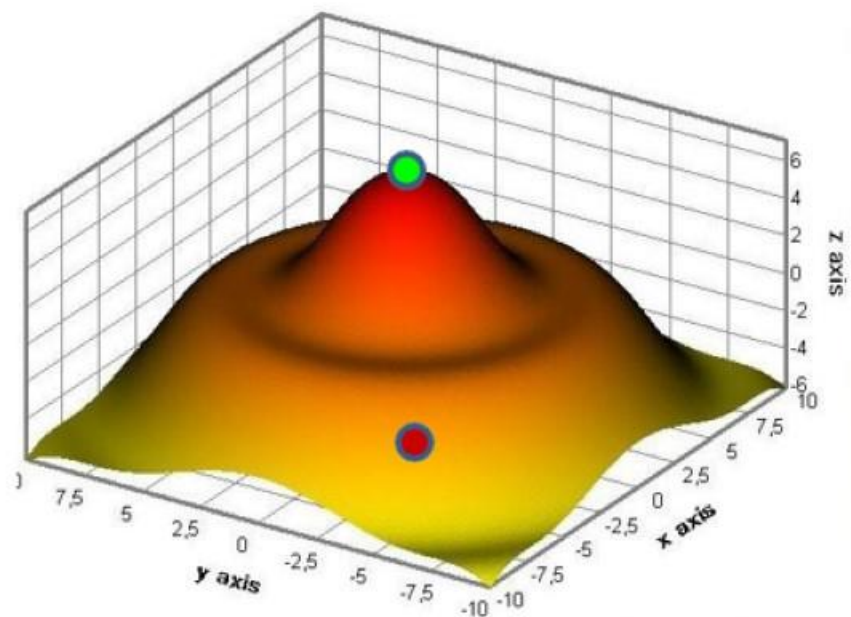
➤ 点云关键点检测

➤ 3D关键点

- 独特性：适合高效的描述和匹配（全局角度定义）
- 重复性：适应视角变化、噪声等（局部角度定义）
- 尺度不变性不是必须考虑的因素



独特性 vs. 可重复性



点云关键点检测

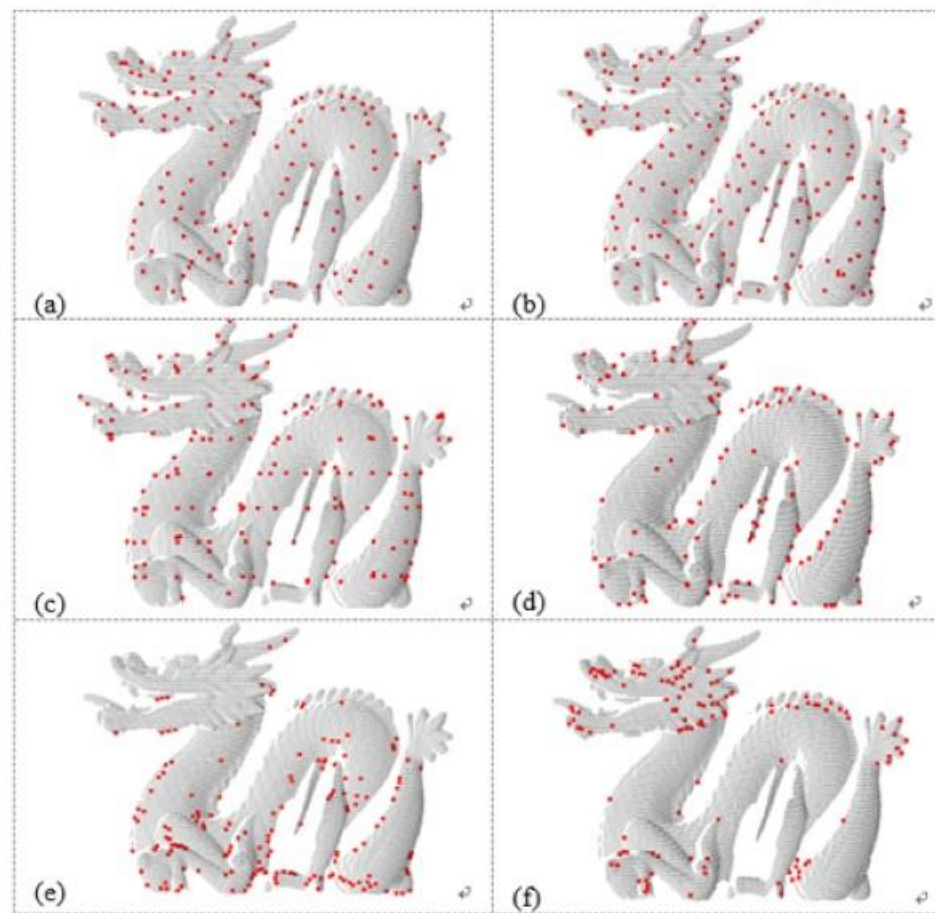
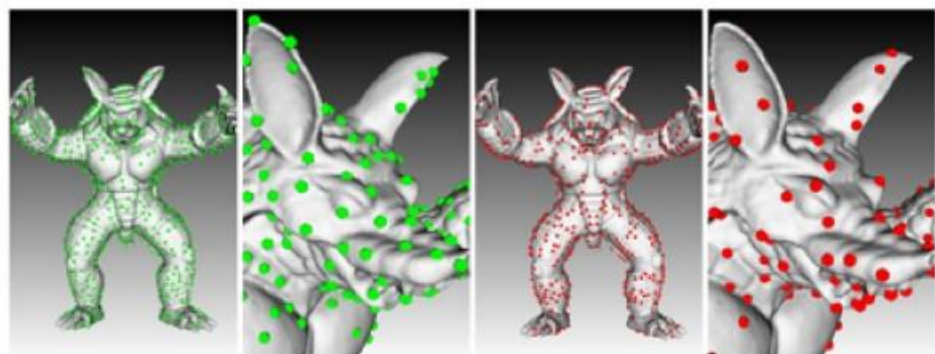
➤ 点云关键点检测 `pcl::Keypoints`

➤ 针对3D点云、深度图提出的检测器

- Intrinsic Shape Signatures(ISS) [Zhong ICCVW09]
- NARF [Steder ICRA11]
- 均匀采样/体素采样

➤ 从2D图像兴趣点检测器演变的检测器

- Harris (2D, 3D, 6D) [Harris AVC88]
- SIFT [Lowe IJCV04]
- SUSAN [Smith IJCV95]
- AGAST [Mair ECCV10]



点云关键点检测

➤ 点云关键点检测 `pcl::Keypoints`

➤ 3D关键点尺度问题:

在大多数PCL应用场景中，点云尺度(scale)并不是问题
但，尺度特性仍然很重要

针对尺度的关键点检测器主要分为两类:

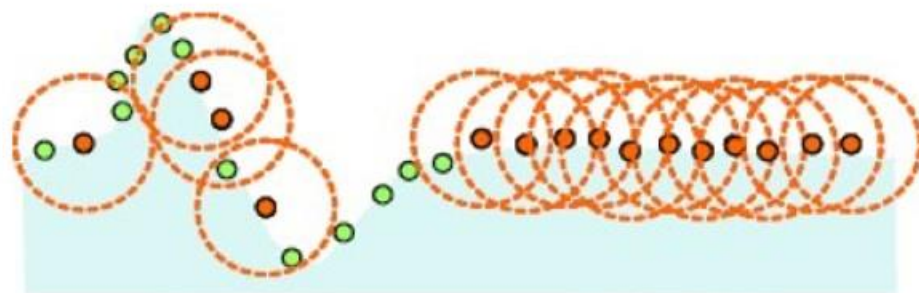
- 固定尺度检测器: 按照参数设定尺度
- 自适应尺度检测器: 尺度分析, 检测多个尺度下的显著结构

➤ 关键点检测器性能评估:

- 局部可重复性/数量
- 尺度可重复性
- 效率

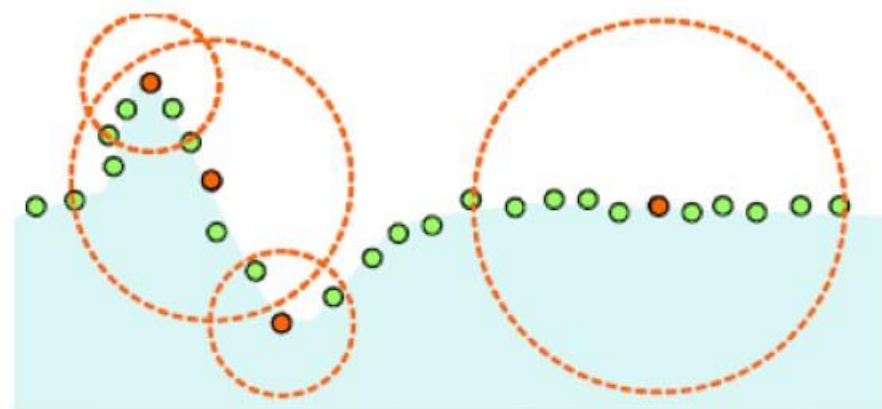


<http://www.vision.deis.unibo.it/keypoints3d/>



Sparse
but not representative

Exhaustive
but redundant



Data-driven selection
of both locations and neighborhoods

点云关键点检测

➤ 点云关键点检测 `pcl::Keypoints`

➤ 固定尺度检测器:

- Local Surface Patches (LSP) [Chen07]
- Intrinsic Shape Signatures (ISS) [Zhong09]
- KeyPoint Quality (KPQ) [Mian10]
- Heat Kernel Signature (HKS) [Sun09]

➤ 自适应尺度检测器:

点云/面元:

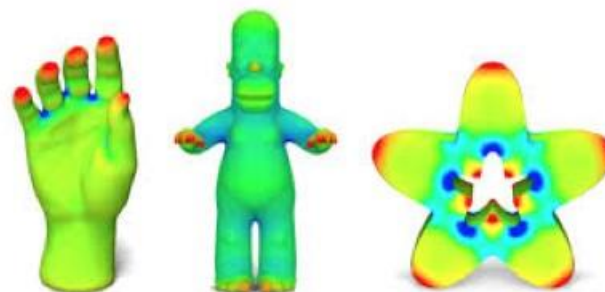
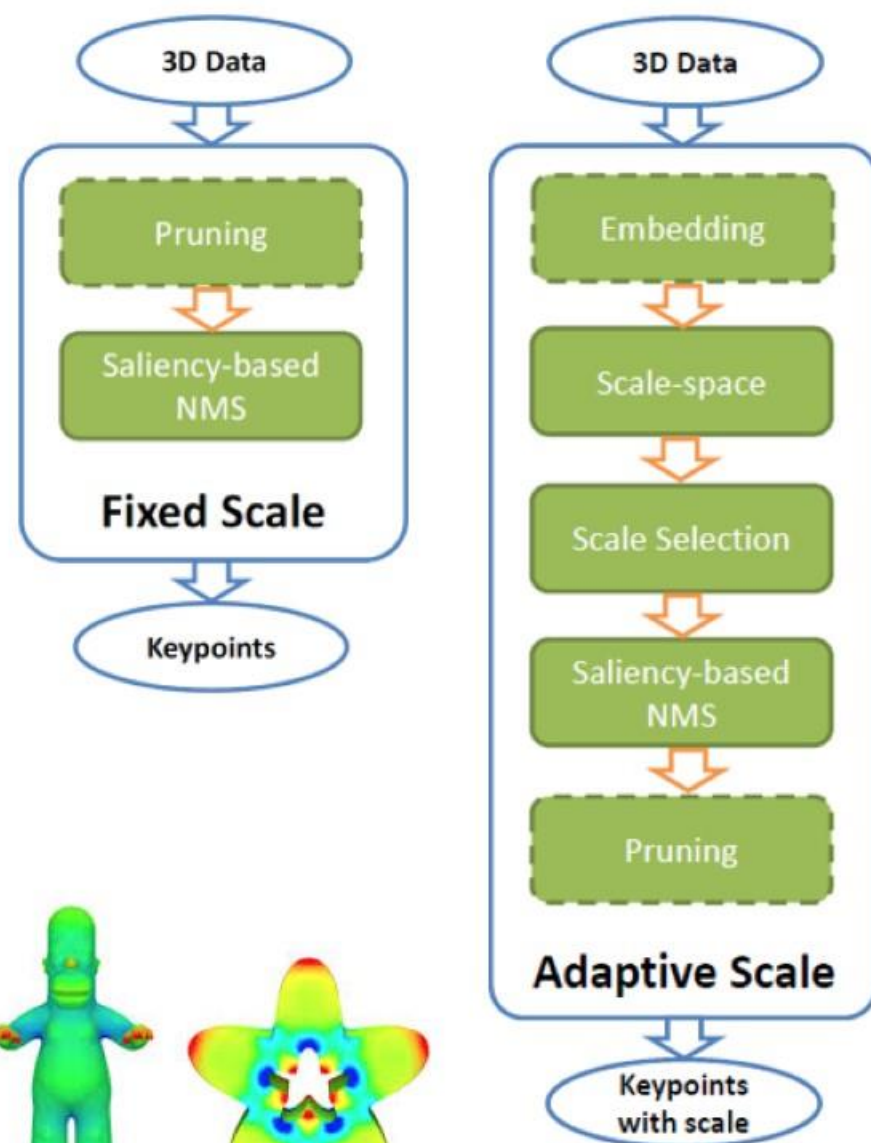
- KPQ Adaptive Scale (KPQ-AS) [Mian10]
- Salient Points (SP) [Castellani08]
- Laplace-Beltrami Scale-Space (LBSS) [Unnikrishnan08]
- MeshDoG [Zaharescu12]

体素地图:

- 3D-SURF [Knopp10]

深度图:

- Scale-dependent local shape detector [Novatnack08]
- HK Maps [Akagunduz07])



点云关键点检测

➤ 内在形状签名(ISS, Intrinsic Shape Signatures)

- 计算协方差矩阵:
$$\mathbf{M}(\mathbf{p}_i) = \frac{1}{\sum_{j=1}^k \rho_j} \sum_{j=1}^k \rho_j (\mathbf{p}_j - \mathbf{p}_i)(\mathbf{p}_j - \mathbf{p}_i)^T$$

- 特征值分解, 求特征值(由大到小):

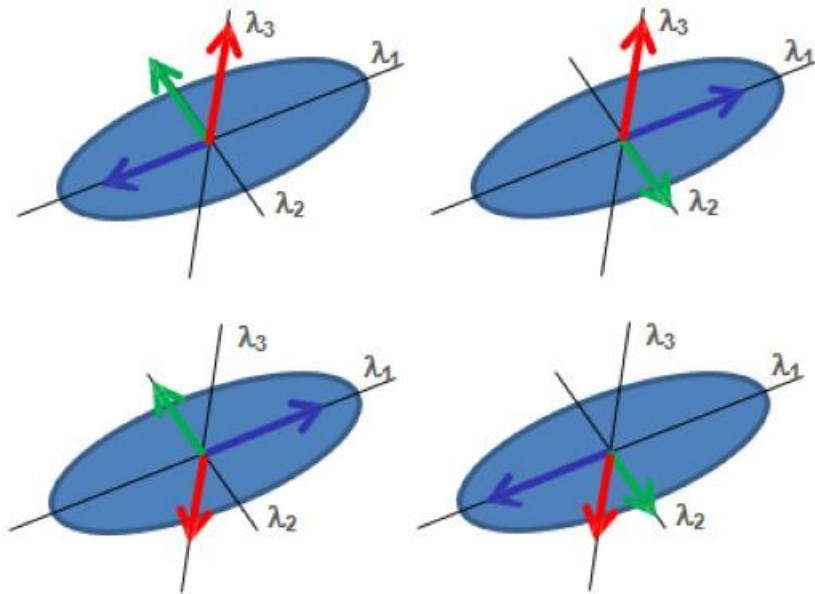
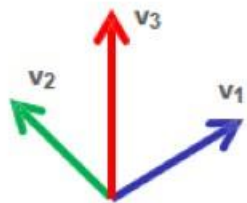
$$\lambda_1 \geq \lambda_2 \geq \lambda_3$$

- 根据以下规则选取关键点, 显著性特征为最小的特征值

$$\frac{\lambda_2(\mathbf{p})}{\lambda_1(\mathbf{p})} < Th_{12} \wedge \frac{\lambda_3(\mathbf{p})}{\lambda_2(\mathbf{p})} < Th_{23}$$

$$\rho(\mathbf{p}) \doteq \lambda_3(\mathbf{p})$$

- 对设置半径内的显著性特征进行非极大值抑制(NMS)
- 保留了沿主方向变化较大的点为关键点

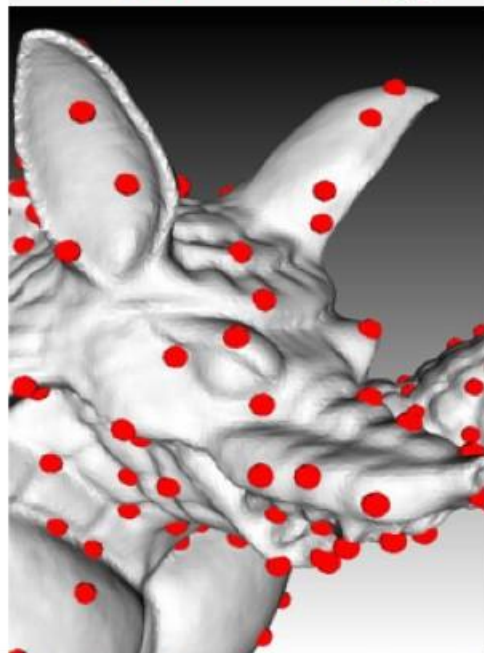


局部参考坐标系(Local Reference Frame, LRF)

点云关键点检测

➤ 内在形状签名(ISS) `pcl::ISSKeypoints3D`

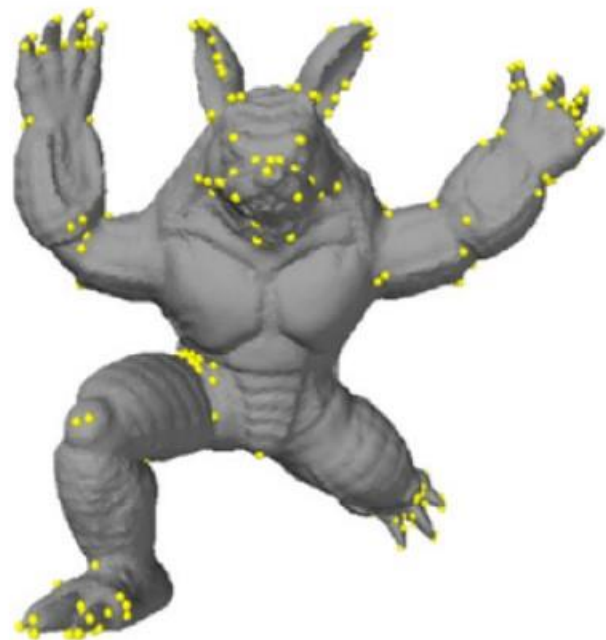
```
//UNIFORM SAMPLING
pcl::PointCloud<int> indices;
pcl::UniformSampling<pcl::PointXYZ> uniform_sampling;
uniform_sampling.setInputCloud (cloud);
uniform_sampling.setRadiusSearch (0.05f); //the 3D grid leaf size
uniform_sampling.compute (indices);
//ISS
pcl::PointCloud<pcl::PointXYZ>::Ptr keypoints (new
pcl::PointCloud<pcl::PointXYZ>());
pcl::search::KdTree<pcl::PointXYZ>::Ptr tree (new
pcl::search::KdTree<pcl::PointXYZ> ());
pcl::ISSKeypoint3D<pcl::PointXYZ, pcl::PointXYZ> iss_detector;
iss_detector.setSearchMethod (tree);
iss_detector.setSalientRadius (support_radius);
iss_detector.setNonMaxRadius (nms_radius);
iss_detector.setThreshold21 (0.975);
iss_detector.setThreshold32 (0.975);
iss_detector.setMinNeighbors (5);
iss_detector.setNumberOfThreads (4);
iss_detector.setInputCloud (cloud);
iss_detector.compute (*keypoints);
```



点云关键点检测

➤ Harris3D `pcl::HarrisKeypoint3D`

```
pcl::HarrisKeypoint3D<pcl::PointXYZ, pcl::PointXYZI, pcl::Normal>
harris;
harris.setInputCloud(point_cloud_ptr); // 设置输入点云 指针
harris.setNonMaxSupression(true);
harris.setRadius(0.02f); // 块体半径
harris.setThreshold(0.01f); // 数量阈值
// 注意Harris的输出点云必须是有强度(I)信息的 pcl::PointXYZI, 因为评估值保存在I分量里
pcl::PointCloud<pcl::PointXYZI>::Ptr cloud_out_ptr (new
pcl::PointCloud<pcl::PointXYZI>);
pcl::PointCloud<pcl::PointXYZI>& cloud_out = *cloud_out_ptr;
// 计算特征点
harris.compute(cloud_out);
```



点云关键点检测

➤ SIFT `pcl::SIFTKeypoint`

//设定参数值

```
const float min_scale = 0.002f; //尺度空间中最小的尺度
```

```
const int n_octaves = 3; //尺度空间层数, 小、关键点多
```

```
const int n_scales_per_octave = 3; //尺度空间层中计算的尺度个数
```

```
const float min_contrast = 0.0001f; //根据点云, 设置大小, 越小关键点越多
```

//sift关键点检测

```
pcl::SIFTKeypoint<pcl::PointXYZ, pcl::PointWithScale > sift_src;
```

```
pcl::PointCloud<pcl::PointWithScale> result_src;
```

```
pcl::search::KdTree<pcl::PointXYZ>::Ptr tree_src(new
```

```
pcl::search::KdTree<pcl::PointXYZ>());
```

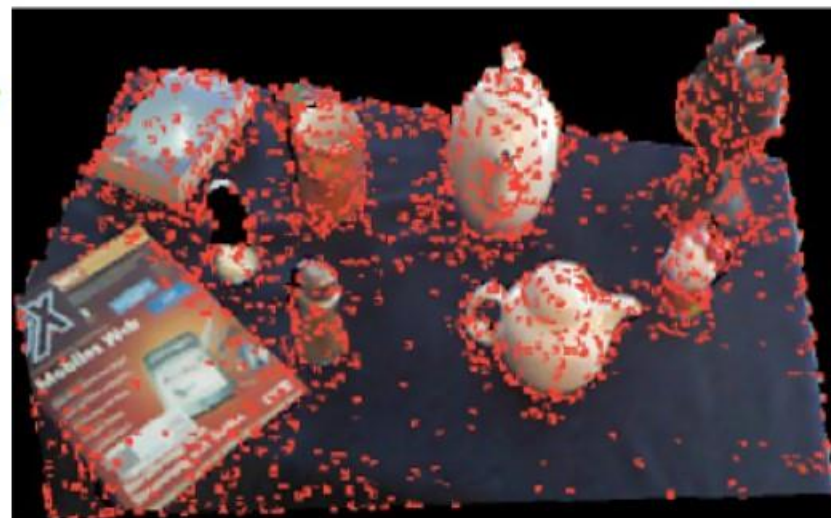
```
sift_src.setSearchMethod(tree_src);
```

```
sift_src.setScales(min_scale, n_octaves, n_scales_per_octave);
```

```
sift_src.setMinimumContrast(min_contrast);
```

```
sift_src.setInputCloud(cloud_src_o);
```

```
sift_src.compute(result_src);
```





内容概要

点云关键点检测
点云常用特征描述

点云常用特征描述

➤ 点云特征描述

➤ 全局描述vs局部描述

➤ 逐点描述子

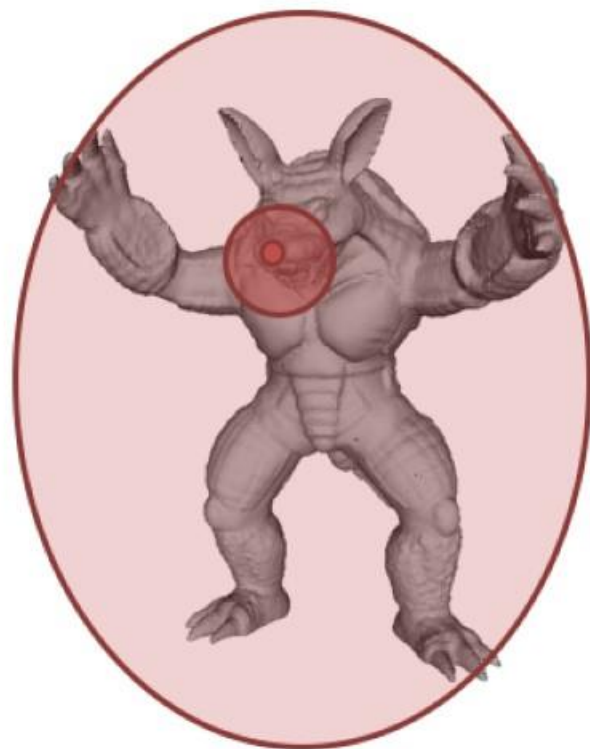
- 简单、高效，但对噪声不够鲁棒，往往不够有描述性

➤ 局部/区域描述子

- 非常适合杂乱和遮挡
- 在码本中可以量化为向量
- 适合杂乱的3D SLAM中分割、配准、识别任务

➤ 全局描述子

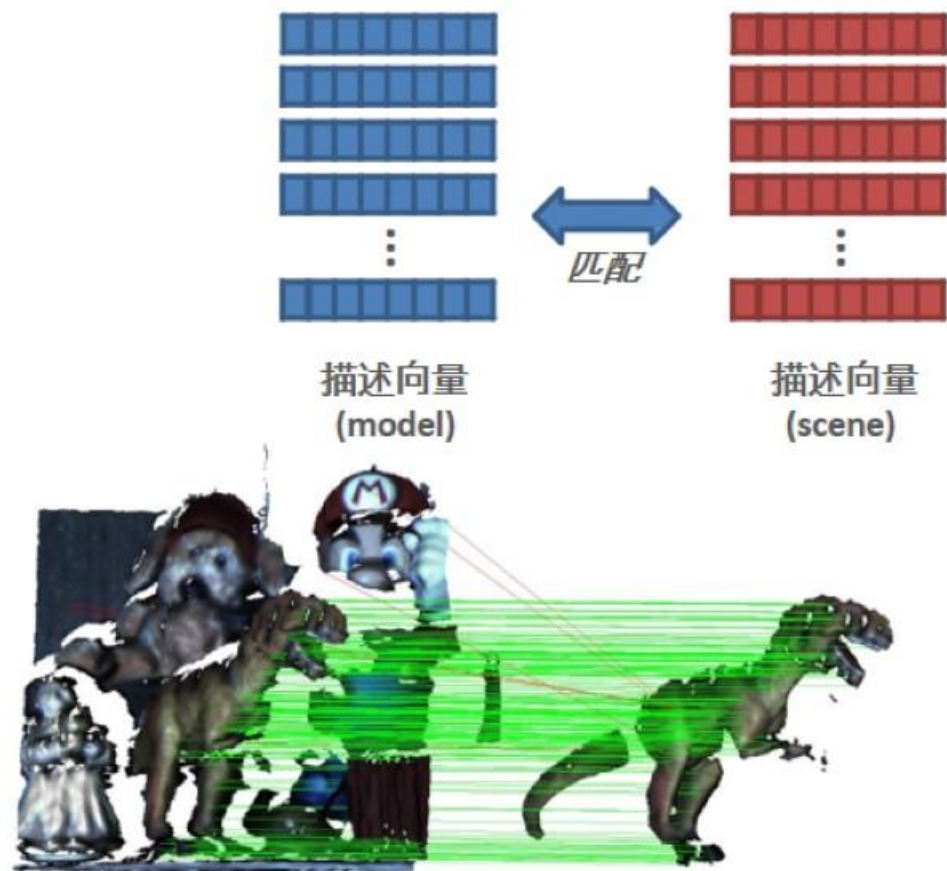
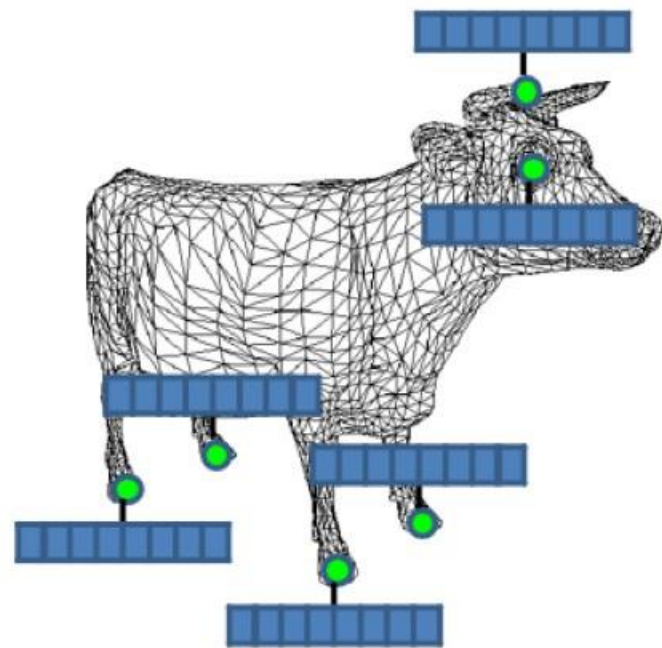
- 需要完整的表面信息(没有遮挡)
- 具有较高的不变性，适合检索和分类
- 具有更高的描述性，对于有较少几何结构的物品



点云常用特征描述

➤ 点云特征描述-局部描述子

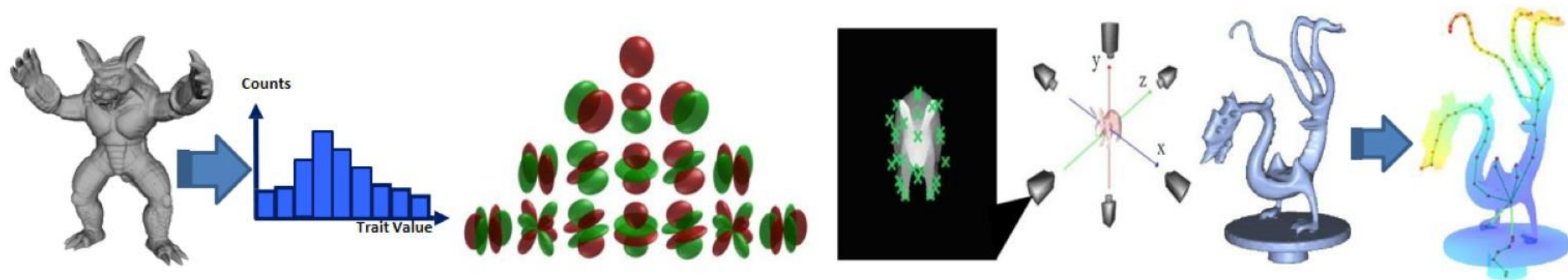
- 对点附近局部邻域的描述性表征
- 局部描述子包括密度和颜色信息(RGB-D)
- 匹配描述子用于两表面的点对点匹配



点云常用特征描述

➤ 点云特征描述-全局描述子

- 鲁棒，但也损失部分描述信息
- 基于直方图、变换、2D多视角图像、连通图



点云常用特征描述

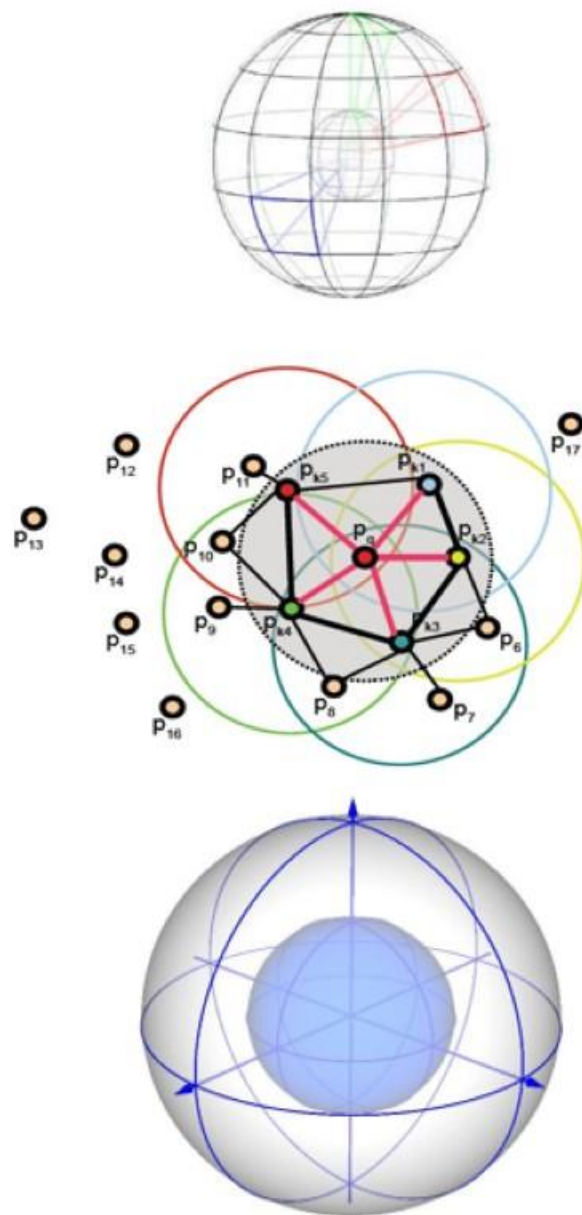
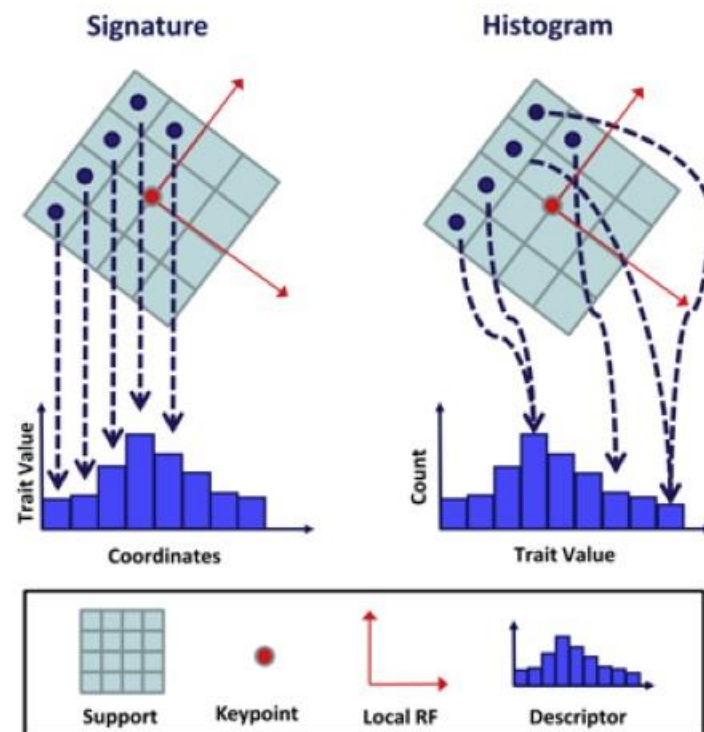
➤ 点云常用特征描述

➤ 局部描述特征

- Spin Image [Johnson99]
- 3DSC [Frome ECCV04]
- PFH [Rusu08]
- FPFH [Rusu09]
- Signatures of Histograms of Orientations, SHOT[Tombari10]
- SHOT for RGB-D data [Tombari11]

➤ 全局描述特征

- Viewpoint Feature Histogram [Rusu 10]
- Clustered VFH (CVFH) [Aldoma 11]

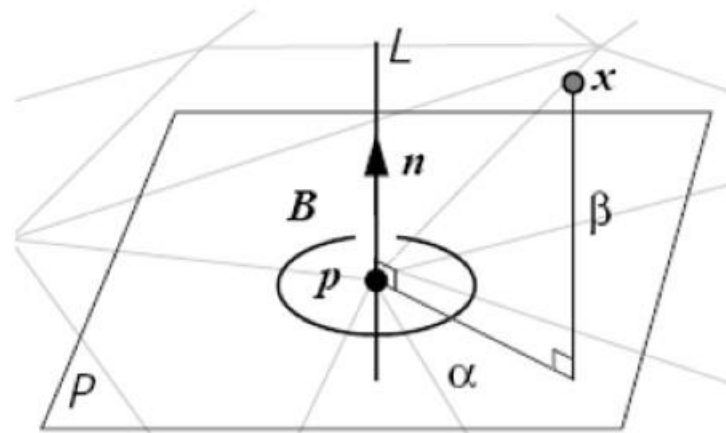


点云常用特征描述

➤ 局部描述特征

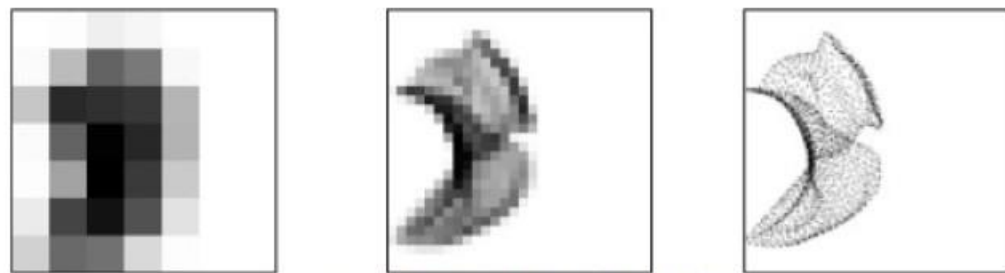
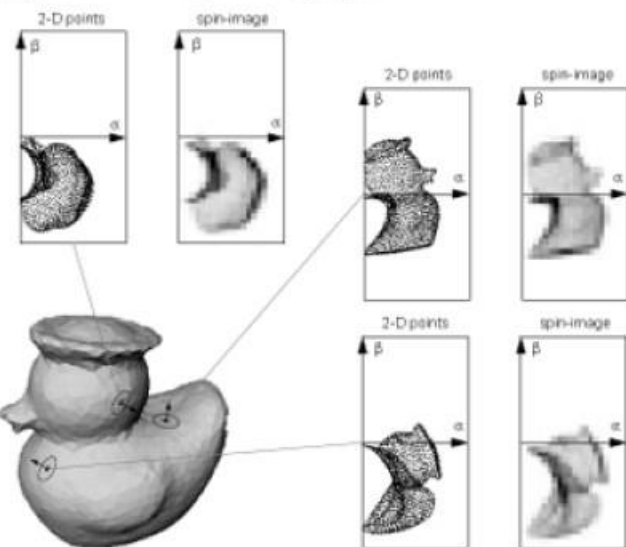
Spin Image `pcl::SpinImageEstimation`

- 通过围绕某个轴(法线)旋转来累积点的二维直方图
- 具有旋转平移不变性, 不具备尺度不变性
- 适合均匀采样



$$S_O: R^3 \rightarrow R^2$$

$$S_O(x) \rightarrow (a, \beta) = (\sqrt{\|x - p\|^2 - (n \cdot (x - p))^2}, n \cdot (x - p))$$



受图像分辨率影响

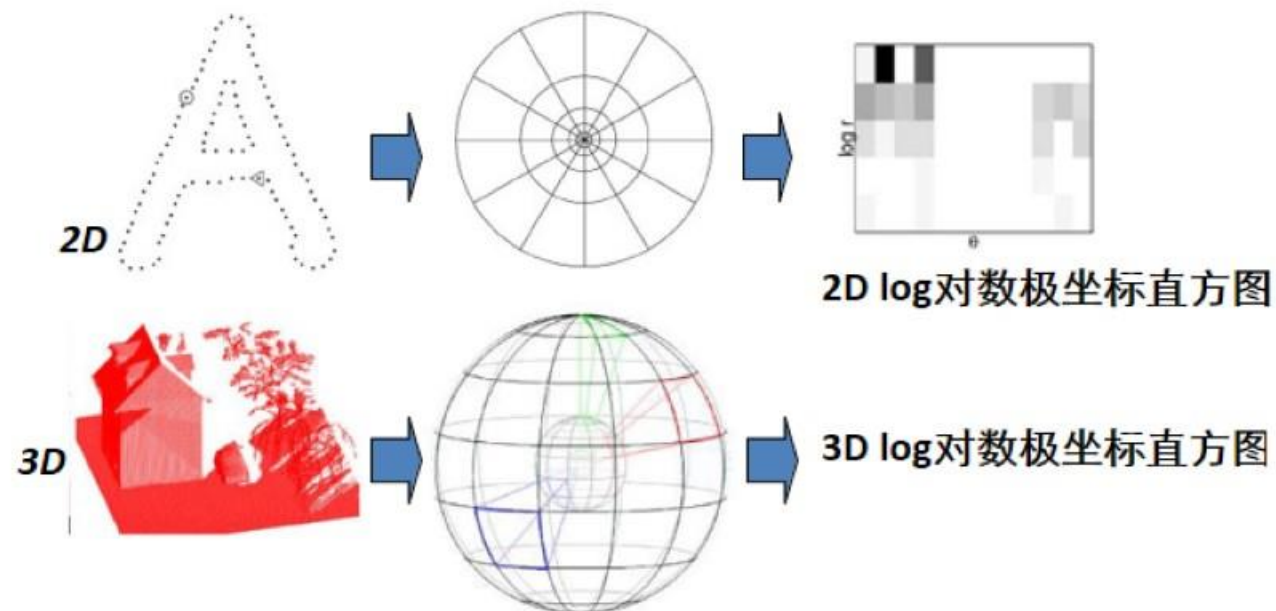
点云常用特征描述

➤ 局部描述特征

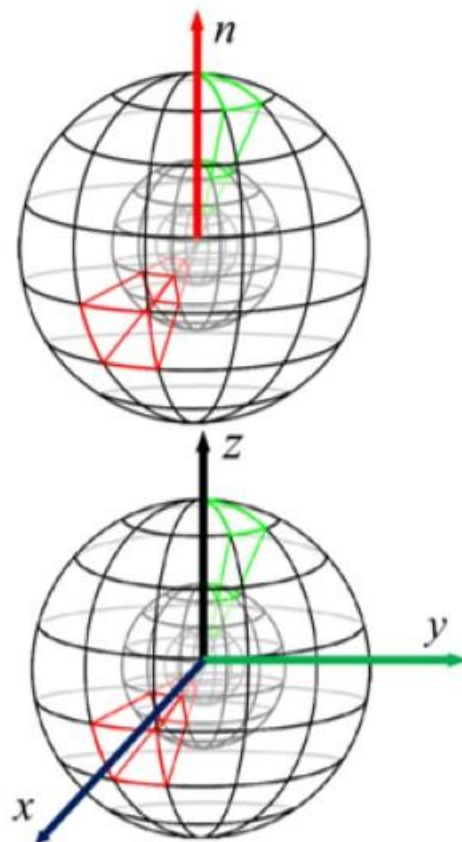
➤ 3D/Unique Shape Contexts

`pcl::ShapeContext3DEstimation` `pcl::UniqueShapeContext`

- 3D 形状上下文特征是扩展图像中的形状上下文特征
- 每个点在其所在的3D bin中累积，并与bin周围的局部点云密度和bin体积成比例加权
- USC通过建立局部参考坐标系LRF，消除3DSC 经度划分方向存在的歧义问题，提高了描述子的区分性，节省了存储空间，减少了匹配过程的虚假匹配



$$w(p_i) = \frac{1}{\rho_i \sqrt[3]{V(j, k, l)}}$$



点云常用特征描述

➤ 局部描述特征

➤ Point Feature Histogram `pcl::PFHEstimation`

- 无差别地计算邻域点集总体的参数直方图
- 对关键点，获取它的近邻，并计算这些近邻点的所有成对组合关系的直方图
- 用到近邻点的局部法向量信息
- 计算速度慢， $O(k^2)$

➤ 对每一点对 $((p_s, n_s); (p_t, n_t))$ ，在 p_s 建立LRF $u-v-w$

➤ $u = n_s, v = (p_t - p_s) \times u, w = u \times v$

$$f_1 = \langle v, n_t \rangle$$

$$f_2 = \|p_t - p_s\|$$

$$f_3 = \langle u, p_t - p_s \rangle / f_2$$

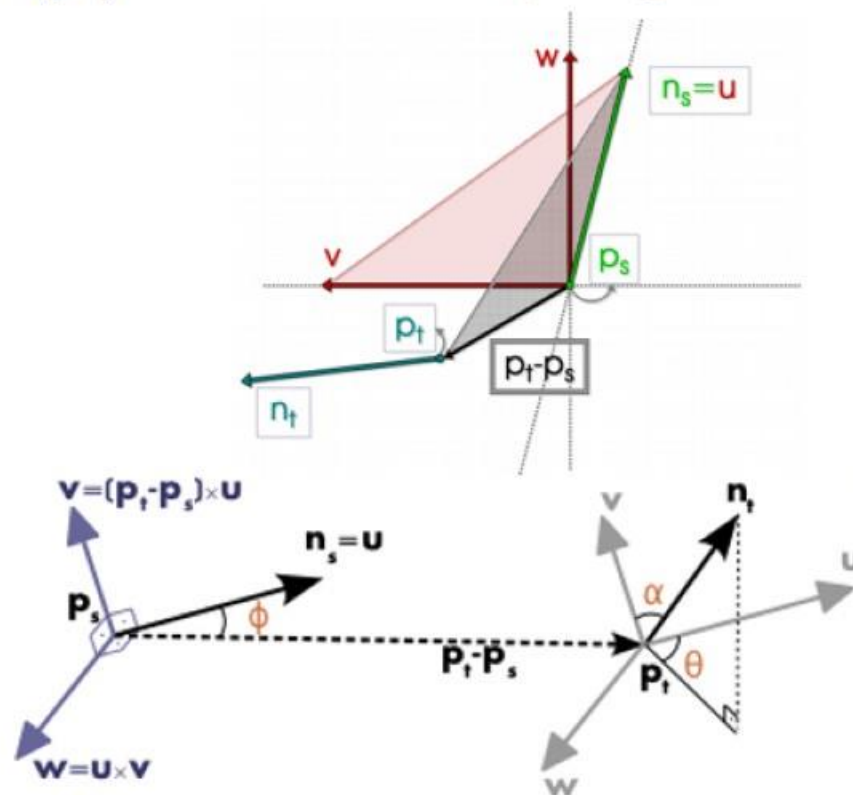
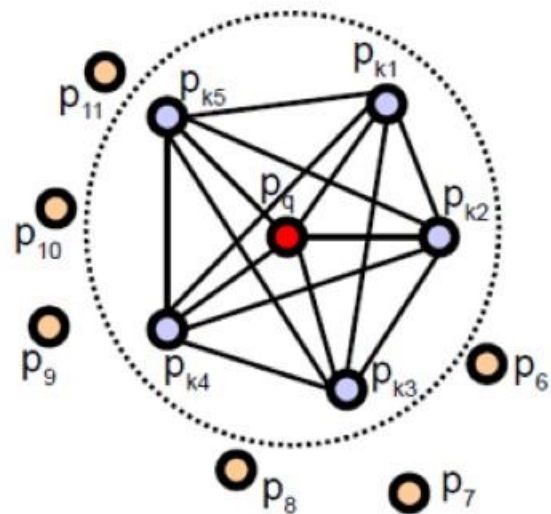
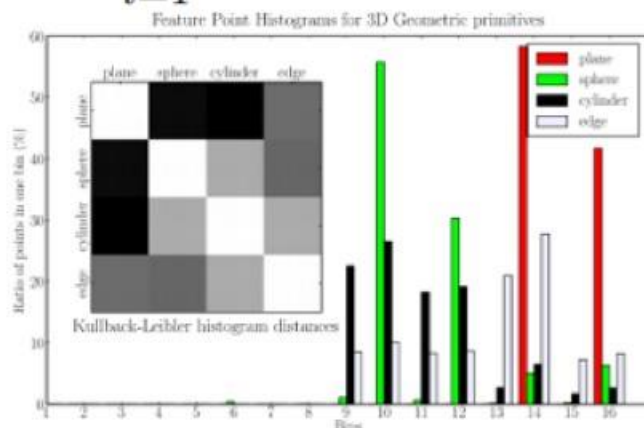
$$f_4 = \text{atan}(\langle w, n_t \rangle, \langle u, n_t \rangle)$$

$$\alpha = \arccos(v \cdot n_t)$$

$$\phi = \arccos\left(u \cdot \frac{(p_t - p_s)}{\|p_t - p_s\|_2}\right)$$

$$\theta = \arctan(w \cdot n_t, u \cdot n_t)$$

$$idx = \sum_{i=1}^{i \leq 4} \text{step}(s_i, f_i) \cdot 2^{i-1}$$

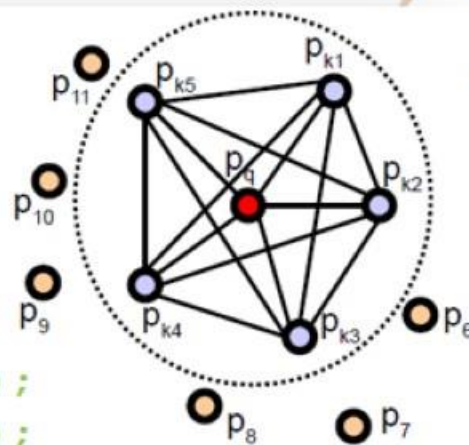


点云常用特征描述

➤ 局部描述特征

Point Feature Histogram `pcl::PFHEstimation`

```
pcl::PointCloud<pcl::PointXYZ>::Ptr cloud(new pcl::PointCloud<pcl::PointXYZ>);
pcl::PointCloud<pcl::Normal>::Ptr normals(new pcl::PointCloud<pcl::Normal>());
...//打开点云文件估计法线等
//创建PFH估计对象pfh，并将输入点云数据集cloud和法线normals传递给它
pcl::PFHEstimation<pcl::PointXYZ,pcl::Normal,pcl::PFHSignature125> pfh;
pfh.setInputCloud(cloud);
pfh.setInputNormals(normals);
//如果点云是类型为PointNormal,则执行pfh.setInputNormals (cloud);
//创建一个空的kd树表示法，并把它传递给PFH估计对象。
pcl::KdTreeFLANN<pcl::PointXYZ>::Ptr tree(new pcl::KdTreeFLANN<pcl::PointXYZ>());
pfh.setSearchMethod(tree);
pcl::PointCloud<pcl::PFHSignature125>::Ptr pfhs(new
pcl::PointCloud<pcl::PFHSignature125>());
//使用半径在5厘米范围内的所有邻元素。
//注意：此处使用的半径必须要大于估计表面法线时使用的半径!!!
pfh.setRadiusSearch(0.05);
//计算pfh特征值
pfh.compute(*pfhs);
```



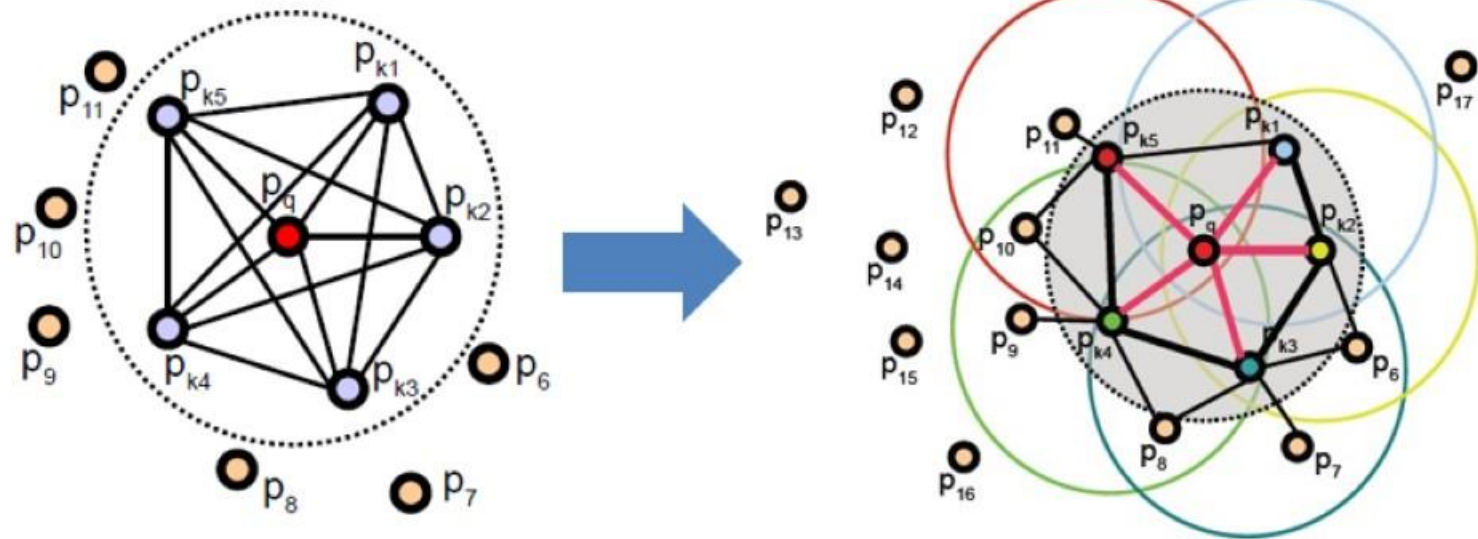
点云常用特征描述

➤ 局部描述特征

Fast PFH `pcl::FPFHEstimation` `pcl::FPFHEstimationOMP`

- 线性时间复杂度近似估计PFH $O(nk^2) \rightarrow O(nk)$
- step1: 计算p点邻域内与之直接相连的所有点对特征，生成Simplified Point Feature Histogram(SPFH)。
- step2: 计算邻域内其他点的SPFH特征。
- step3: p点的FPFH特征由邻域内所有点的SPFH特征加权得到。

$$FPFH(p_i) = SPFH(p_i) + \sum_{j=1} \frac{1}{\omega_j} SPFH(p_j)$$

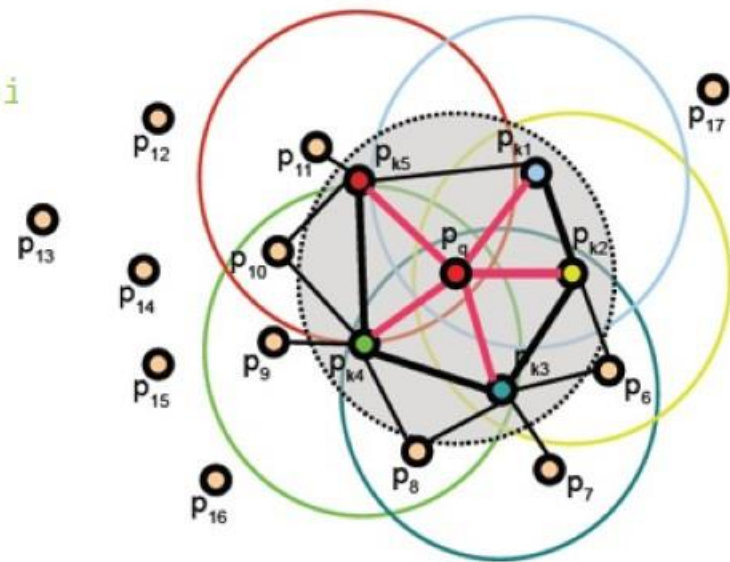


点云常用特征描述

➤ 局部描述特征

Fast PFH `pcl::FPFHEstimation`

```
pcl::PointCloud<pcl::PointXYZ>::Ptr cloud(new pcl::PointCloud<pcl::PointXYZ>);  
pcl::PointCloud<pcl::Normal>::Ptr normals(new pcl::PointCloud<pcl::Normal>());  
...//打开点云文件估计法线等  
//创建FPFH估计对象fpfh，并把输入数据集cloud和法线normals传递给它。  
pcl::FPFHEstimation<pcl::PointXYZ,pcl::Normal,pcl::FPFHSignature33> fpfh;  
fpfh.setInputCloud(cloud);  
fpfh.setInputNormals(normals);  
//如果点云是类型为PointNormal，则执行fpfh.setInputNormals (cloud);  
//创建一个空的kd树对象tree，并把它传递给FPFH估计对象。  
pcl::search::KdTree<PointXYZ>::Ptr tree(new pcl::search::KdTree<PointXYZ>);  
fpfh.setSearchMethod(tree);  
pcl::PointCloud<pcl::FPFHSignature33>::Ptr fpfhs(new  
pcl::PointCloud<pcl::FPFHSignature33>());  
//使用所有半径在5厘米范围内的邻元素  
//注意：此处使用的半径必须要大于估计表面法线时使用的半径!!!  
fpfh.setRadiusSearch(0.05);  
//计算获取特征向量  
fpfh.compute(*fpfhs);
```

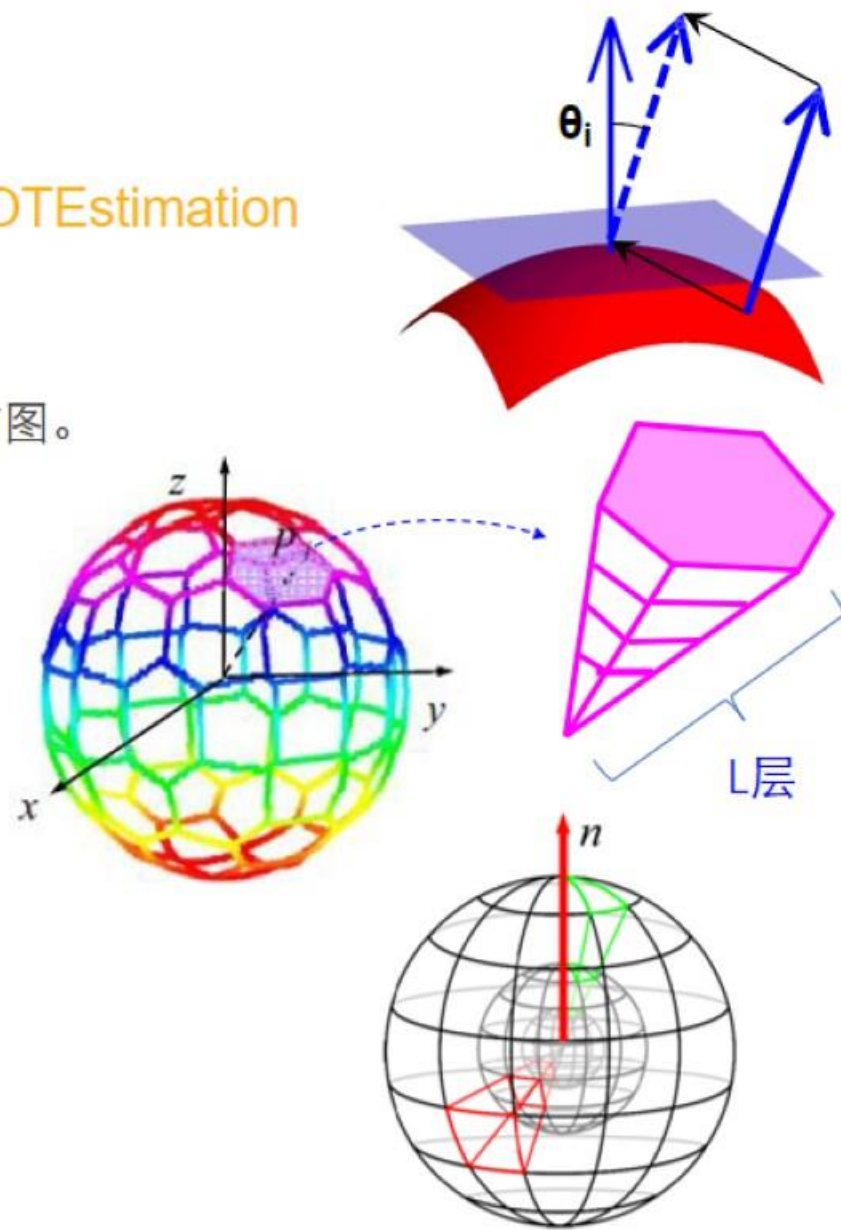
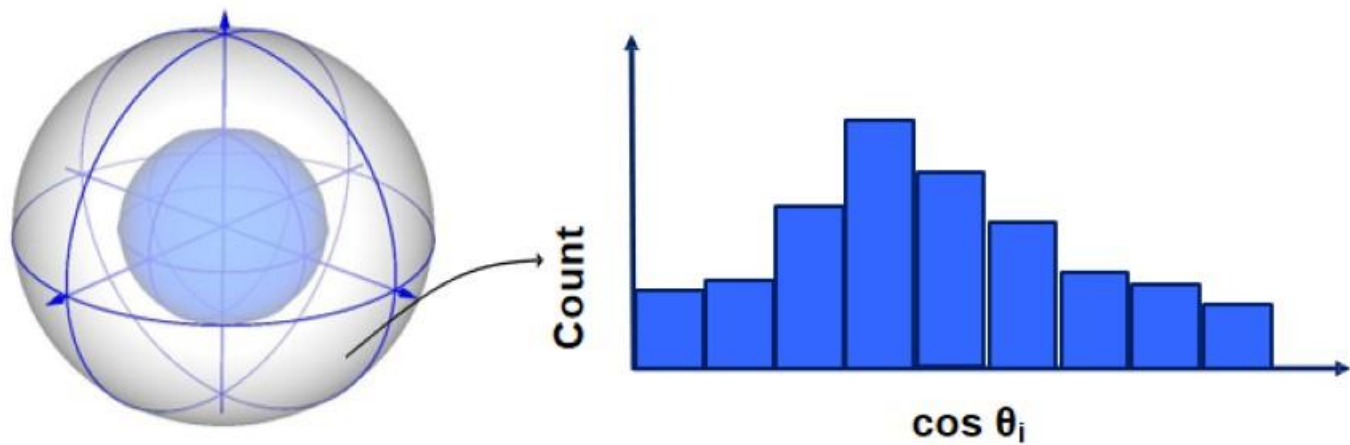


点云常用特征描述

➤ 局部描述特征

➤ SHOT(Signature of Histograms of Orientations) `pcl::SHOTEstimation`

- 灵感来自SIFT、3DSC、ISS:局部集的计算几何直方图。
- 通过球面网格分区
- 对于每个网格,每一点和特征点计算法向量夹角 θ_i 的余弦值构建直方图。
- 四线性插值平滑量化失真
- 归一化描述特征以对点密度变化的健壮性



点云常用特征描述

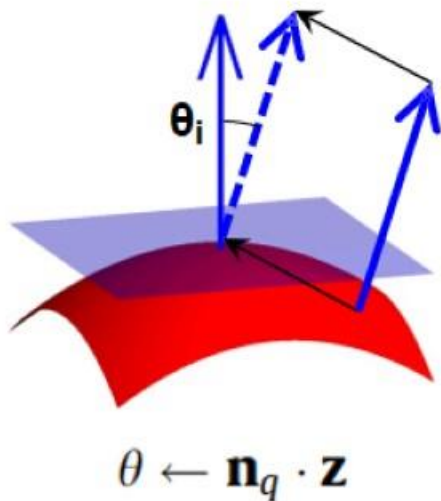
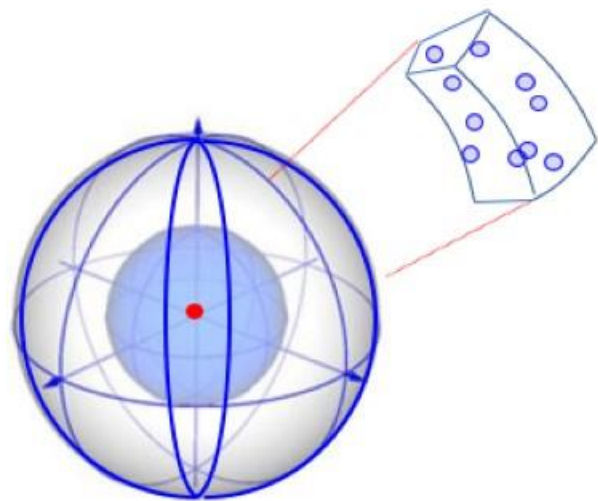
➤ 局部描述特征

➤ SHOT(Signature of Histograms of Orientations) `pcl::SHOTEstimation`

$$\mathbf{M} = \frac{1}{k} \sum_{i=0}^k (\mathbf{p}_i - \hat{\mathbf{p}})(\mathbf{p}_i - \hat{\mathbf{p}})^T, \quad \hat{\mathbf{p}} = \frac{1}{k} \sum_{i=1}^k \mathbf{p}_i$$

$$\mathbf{M}(\mathbf{p}_i) = \frac{1}{\sum_{j=1}^k \rho_j} \sum_{j=1}^k \rho_j (\mathbf{p}_j - \mathbf{p}_i)(\mathbf{p}_j - \mathbf{p}_i)^T \rho_j = \frac{1}{\|\mathbf{p}_j: \|\mathbf{p}_j - \mathbf{p}_i\| < r_{density}\|}$$

$$\mathbf{M} = \frac{1}{\sum_{i: d_i \leq R} (R - d_i)} \sum_{i: d_i \leq R} (R - d_i)(\mathbf{p}_i - \mathbf{p})(\mathbf{p}_i - \mathbf{p})^T d_i = \|\mathbf{p}_i - \mathbf{p}\|_2$$



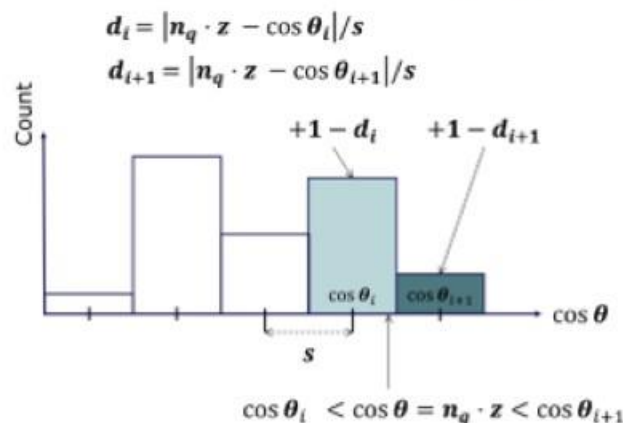
- step1: 根据特征点球邻域信息建立局部参考坐标系 LRF,对特征点的球邻域分别沿径向（内外球）、经度（时区）和纬度方向(南北半球)进行区域划分。通常径向划分为2，经度划分为8，纬度划分为2，总共32个小区域。
- step2:分别统计每个小区域内的法向量夹角余弦值分布情况，余弦值划分为11个bin。
- 最终SHOT的长度为：32x11=352。

点云常用特征描述

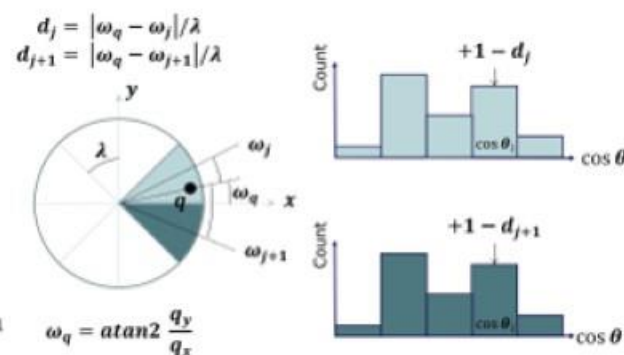
➤ 局部描述特征

➤ SHOT(Signature of Histograms of Orientations) `pcl::SHOTEstimation`

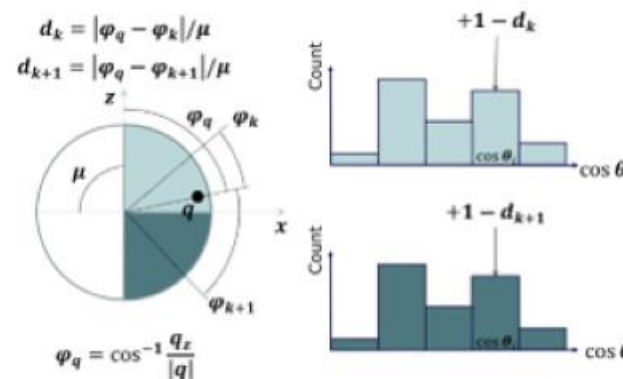
- 四线性插值平滑量化失真
- \cos 直方图
- 经度
- 纬度
- 距离(球层)



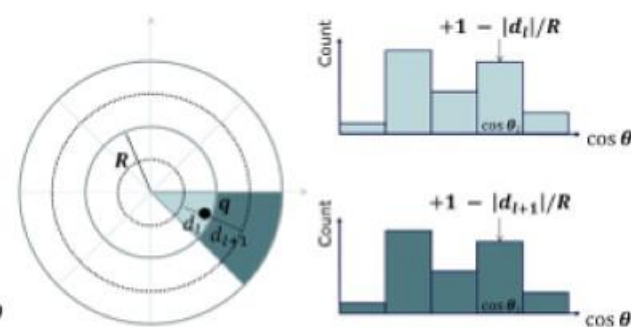
(a) Interpolation on normal cosines



(b) Interpolation on azimuth



(c) Interpolation on elevation



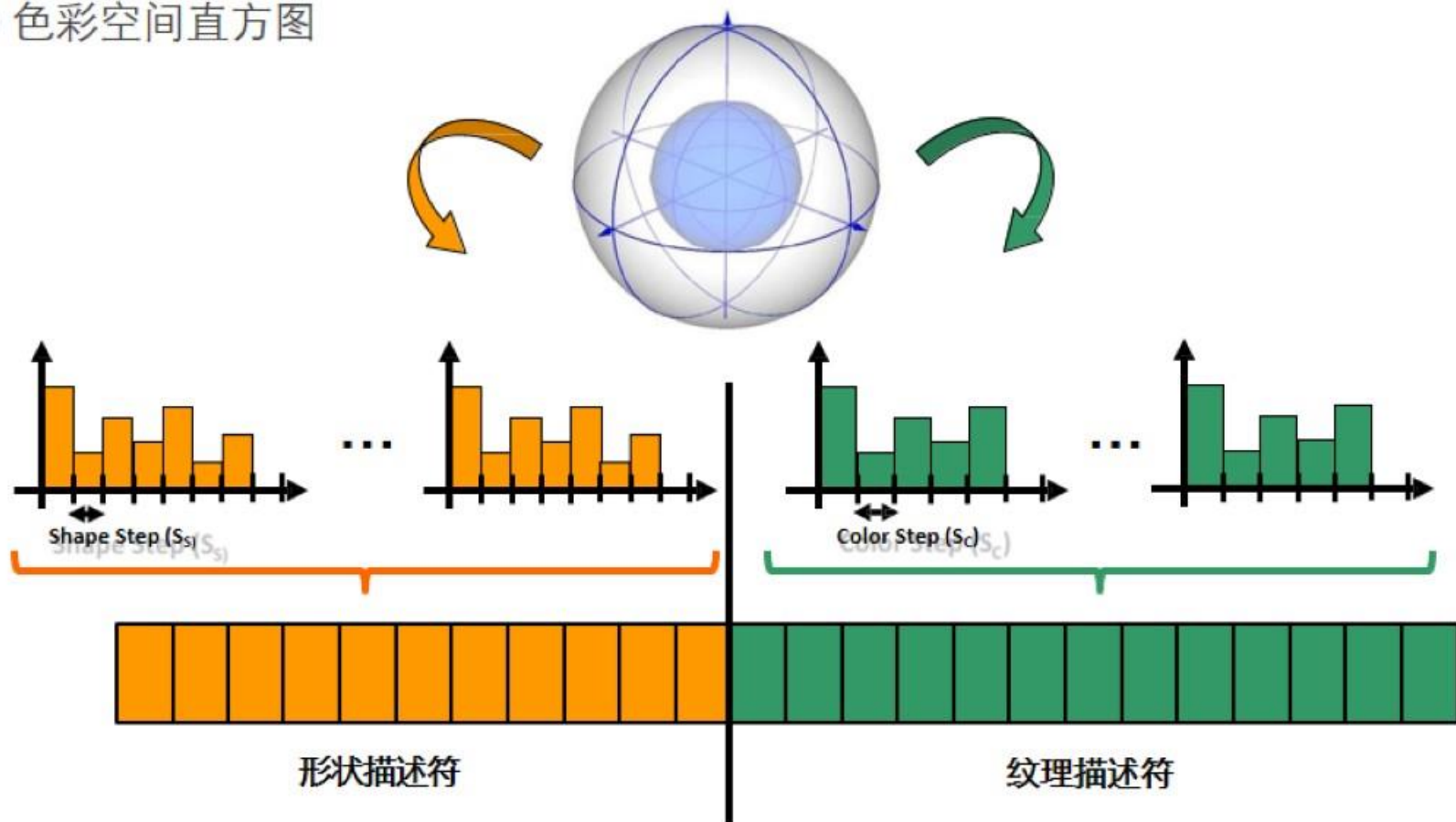
(d) Interpolation on distance

点云常用特征描述

➤ 局部描述特征

➤ SHOT for RGB-D `pcl::SHOTColorEstimation`

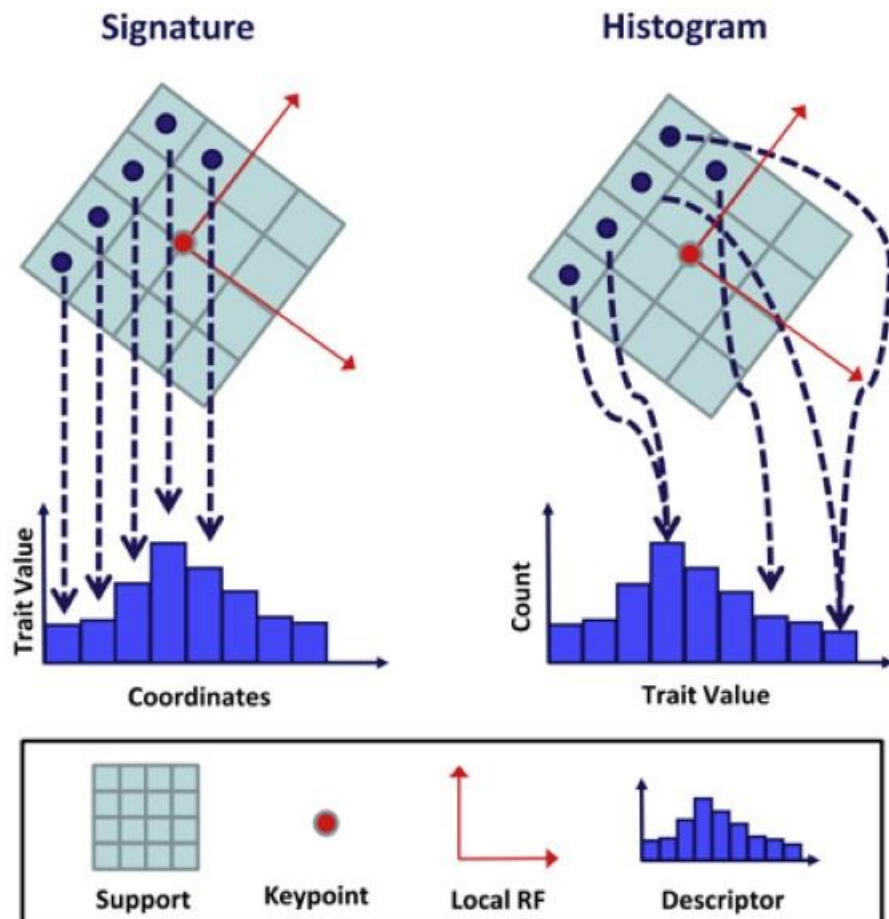
- 形状，用SHOT描述符
- 纹理，用Lab 色彩空间直方图



点云常用特征描述

➤ 局部描述特征总结

Signature vs Histogram



A comprehensive review of 3D point cloud descriptors

Xian-Feng Han^a, Jesse S. Jin^a, Juan Xie^a, Ming-Jie Wang^b, Wei Jiang^a

^aSchool of Computer Software, Tianjin University, 30072, Tianjin, China

^bFaculty of Science, Memorial University of Newfoundland, A1B3X9, Newfoundland and Labrador, Canada

Method	Category	Unique LRF	Texture
Struct. Indexing [Stein92]	Signature	No	No
PS [Chua97]	Signature	No	No
3DPF [Sun01]	Signature	No	No
3DGSS [Novatnack08]	Signature	No	No
KPQ [Mian10]	Signature	No	No
3D-SURF [Knopp10]	Signature	Yes	No
SI [Johnson99]	Histogram	RA	No
LSP [Chen07]	Histogram	RA	No
3DSC [Frome04]	Histogram	No	No
ISS [Zhong09]	Histogram	No	No
USC [Tombari10]	Histogram	Yes	No
PFH [Rusu08]	Histogram	RA	No
FPFH [Rusu09]	Histogram	RA	No
Tensor [Mian06]	Histogram	No	No
RSD [Marton11]	Histogram	RA	No
HKS [Sun09]	Other	-	No
MeshHoG [Zaharescu09]	Hybrid	Yes	Yes
SHOT [Tombari10]	Hybrid	Yes	Yes

点云常用特征描述

➤ 全局描述特征

➤ 基于直方图

- Shape Distributions [Osada02]
- 3D Shape Histograms [Ankerst99]
- Orientation Histograms [Horn84]
- **Viewpoint Feature Histogram (VFH) [Rusu10]**
- **Clustered-VFH [Aldoma11]**
- OUR-CVFH [Aldoma12]

➤ 基于变换

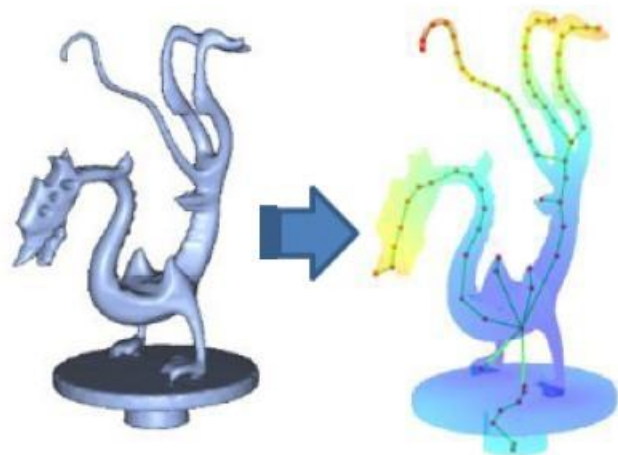
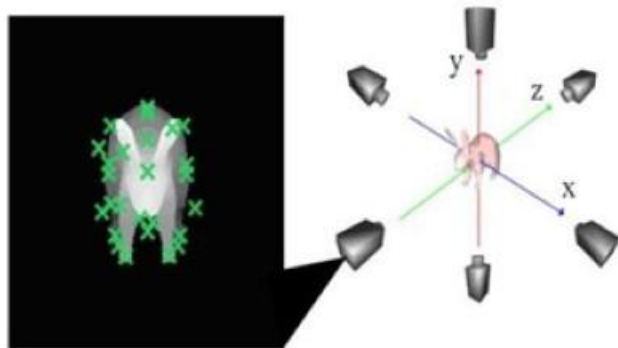
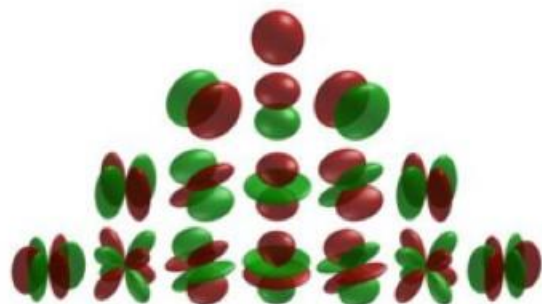
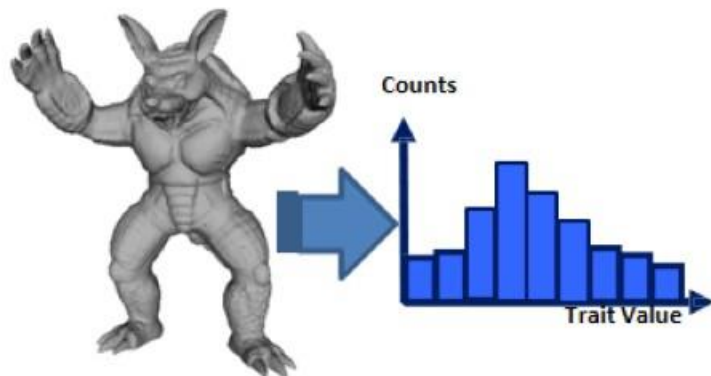
- 3D Fourier Transform [Dutagaci05]
- Angular Radial Tr. [Ricard05]
- 3D Radon Tr. [Daras04]
- Spherical Harmonics [Kazhdan03]
- wavelets [Laga06]

➤ 基于2D视角图像

- Fourier descriptors [Vranic 04]
- Zernike moments [Chen03]
- SIFT [Ohbuchi08]
- SURF, ...

➤ 基于连通图

- topology-based [Hilaga01]
- Reeb graph [Tung05]
- skeleton-based [Sundar03]

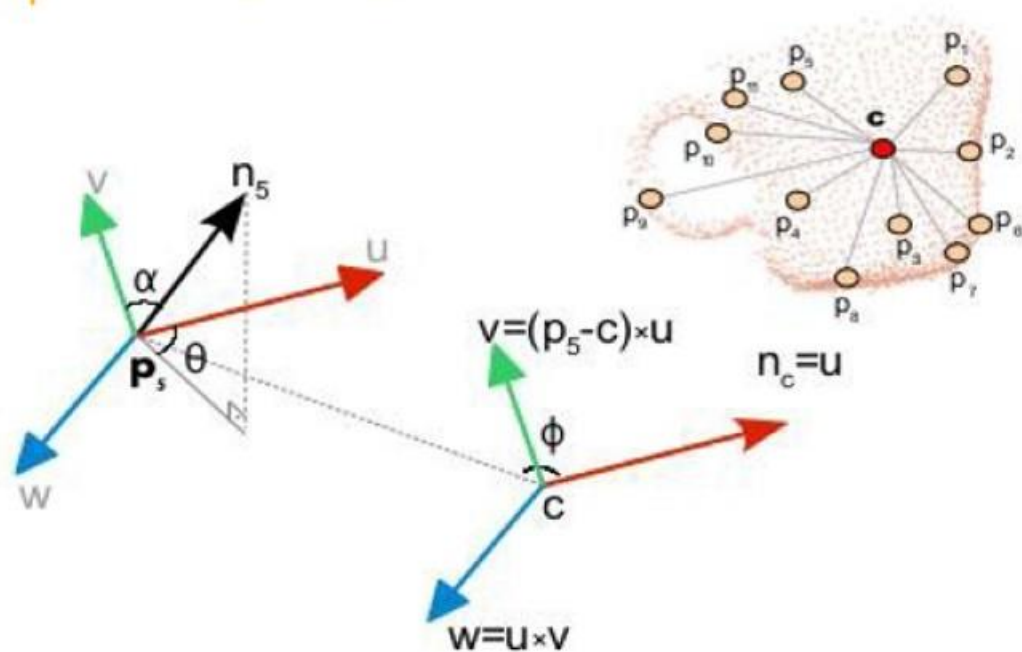


点云常用特征描述

➤ 全局描述特征

➤ 视点特征直方图VFH(Viewpoint Feature Histogram) `pcl::VFHEstimation`

- 每个3D模型都呈现在不同的视图中
- 每个视图提供一个描述符
显式地编码捕获/感知物体表面的视点
- 基于点特征直方图(PFH/FPFH)
- 1.扩展FPFH, 利用整个点云对象来进行计算估计, 计算FPFH时以物体中心点与物体表面其他所有点之间的点对作为计算单元。
- 2.添加视点方向与每个点估计法线之间额外的统计信息, 在FPFH计算中将视点方向变量直接融入到相对法线角计算当中。



- 在物体中心点建立LRF $u-v-w$
- $u = n_s, v = (p_t - p_s) \times u, w = u \times v$

$$\alpha = \arccos(v \cdot n_t)$$

$$\phi = \arccos\left(u \cdot \frac{(p_t - p_s)}{\|p_t - p_s\|_2}\right)$$

$$\theta = \arctan(w \cdot n_t, u \cdot n_t)$$

点云常用特征描述

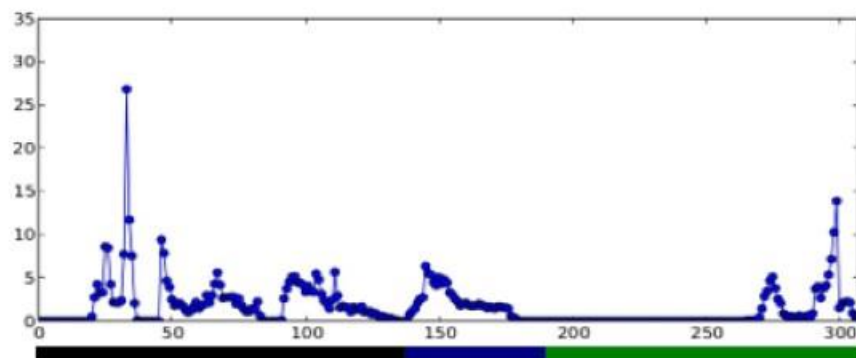
➤ 全局描述特征

➤ 视点特征直方图VFH(Viewpoint Feature Histogram) `pcl::VFHEstimation`

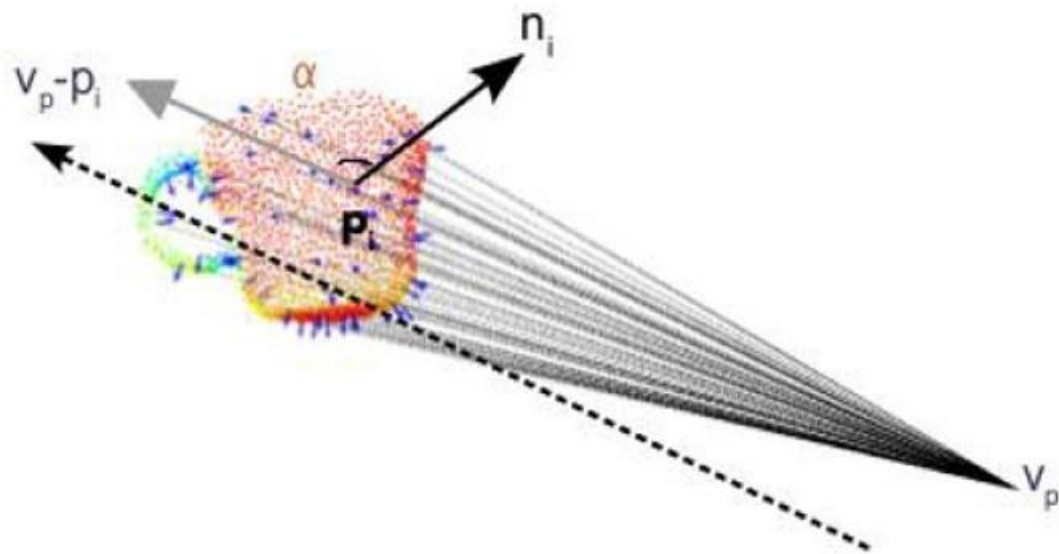
- 特征构成:
- 关于质心的3个"PFH"角度值(α , θ , Φ), (45 bins * 3)
- 关于质心的1个形状分布型描述值 (45 bins)

$$SDC = \frac{(p_c - p_i)^2}{\max ((p_c - p_i)^2)}$$

- 1个视角方向与点法向量之间的角度值(α) (128 bins)



FPFH (α , θ , Φ) Shape dist. Viewpoint (α)
(45*3 bins) (45 bins) (128 bins)

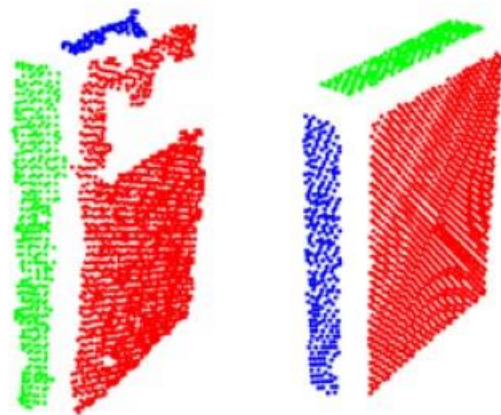


点云常用特征描述

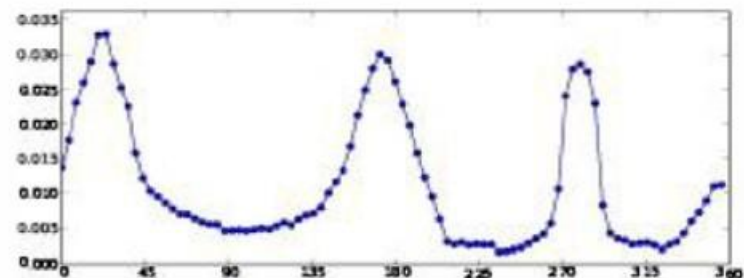
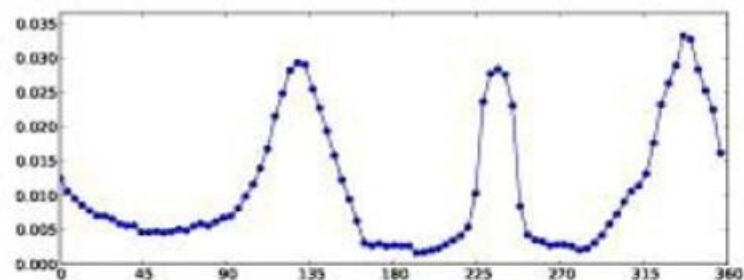
➤ 全局描述特征

➤ 聚类视点特征直方图 Cluster VFH `pcl::CVFHEstimation`

- VFH对表面的缺失部分很敏感
- Cluster VFH:
 - 对每个视图执行区域分割
 - 对每个聚类上应用VFH——不需要对实际大小进行编码



- VFH、CVFH对横滚角度变化的影响

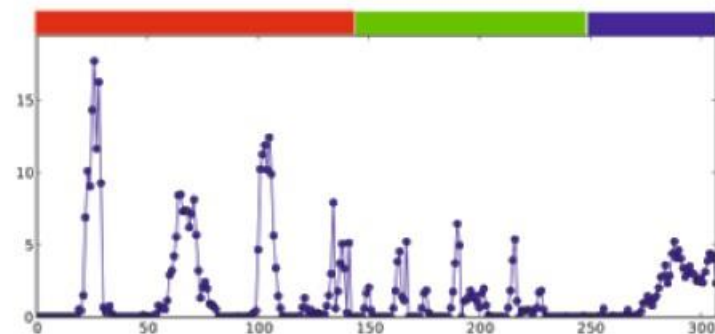
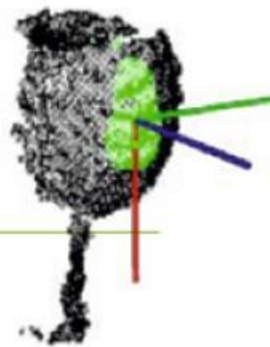


点云常用特征描述

➤ 全局描述特征

➤ VFH(Viewpoint Feature Histogram) `pcl::VFHEstimation`

```
pcl::PointCloud<pcl::PointXYZ>::Ptr cloud (new pcl::PointCloud<pcl::PointXYZ>);
pcl::PointCloud<pcl::Normal>::Ptr normals (new pcl::PointCloud<pcl::Normal> ());
...//打开点云文件估计法线等
//创建VFH估计对象vfh, 并把输入数据集cloud和法线normal传递给它
pcl::VFHEstimation<pcl::PointXYZ, pcl::Normal, pcl::VFHSignature308> vfh;
vfh.setInputCloud (cloud);
vfh.setInputNormals (normals);
//如果点云是PointNormal类型, 则执行vfh.setInputNormals (cloud);
//创建一个空的kd树对象, 并把它传递给FPFH估计对象。
pcl::KdTreeFLANN<pcl::PointXYZ>::Ptr tree (new pcl::KdTreeFLANN<pcl::PointXYZ> ());
vfh.setSearchMethod (tree);
pcl::PointCloud<pcl::VFHSignature308>::Ptr vfhs (new
pcl::PointCloud<pcl::VFHSignature308> ());
//计算特征值
vfh.compute (*vfhs);
```



点云常用特征描述

➤ 点云特征匹配

➤ 问题:

- 在一组 m 个 n 维候选向量中查找查询向量 q 的 k NN
- 变式:在以 q 为中心的半径为 r 的超球面上找到所有的邻域

➤ 加速暴力搜索(brute force), 使用快速索引方案

- Kd-tree [Freidman77]
- Hierarchical k-means tree [Fukunaga75]
- Locality Sensitive Hashing (LSH) [Andoni06]

➤ 高维情况下近似Kd-tree搜索

- Best Bin First [Beis97]
- Randomized kd-tree [Silpa-Anan08]
- FLANN [Muja09]

