[1]PCL官网教程：<http://pointclouds.org/documentation/tutorials/>

1. 自己总结

* 3D重建分为：

1. 离线重建：PCL(见链接)、SFM

SDF：从某帧图像看过去得到的体素为表面，那么其他体素相对当前帧深度小于表面体素值的在表面前，大于表面体素值得在表面后；Curless[50]等人提出了TSDF (Truncated Signed Distance Field，截断符号距离场)算法，该方法只存储距真实表面较近的数层体素，而非所有体素。因此能够大幅降低KinectFusion的内存消耗，减少模型冗余点。

<https://blog.csdn.net/lianchenglian/article/details/81540159>

1. 在线重建：KinectFusion之后，陆续出现了Kintinuous，ElasticFusion，InfiniTAM，BundleFusion等非常优秀的工作。其中2017年斯坦福大学提出的BundleFusion算法，可以说是目前基于RGB-D相机进行稠密三维重建效果最好的方法了。

<https://zhuanlan.zhihu.com/p/37277511>

* 单视角 3D 重建现在已经成为热门话题，仅 CVPR 2019 就收到了大约 35 篇相关论文。一些方法能够基于全景图像生成场景布局，基于图像生成目标网格，基于单视角生成深度图。但是，就像我和 UCI 的研究者发表在 CVPR 2018 的论文《Pixels, voxels, and views: A study of shape representations for single view 3D object shape prediction》以及德国弗莱堡大学研究者发表在 CVPR 2019 的论文《What Do Single-view 3D Reconstruction Networks Learn?》所指出的那样，很多看似能解释图像几何的方法实际上只是在学习过程中记住了图像的几何信息，并通过检索与输入类似的样本来执行预测。预测得到的 3D 模型看似很好，但这些方法无法泛化到新的形状或场景。

3D 重建的两个重要问题：图像深度；自动进度监控。

<https://mp.weixin.qq.com/s/k1aZb-6xrftztDSifEWmQQ>

* 为什么三维重建才是计算机视觉的灵魂？三维重建的发展历程：

<https://mp.weixin.qq.com/s/vKGRVDZEfmn6Yy0MFeuEDA>

* 谷歌重磅推出TensorFlow Graphics：为3D图像任务打造的深度学习利器<https://mp.weixin.qq.com/s/2jq2e8z3CeIF2jc9QobjRA>

1. 1

PCL

1. IDE：Eclipse
2. 常用3D特征：用某个点周围的N个点合成的曲率或法向量表示，可以通过周围n个点求特征分解得到曲率和法向量，法向量为特征值最小的特征向量，曲率通过特征值求得。快速获得邻居的方法是 octrees or kD-trees，可以用他们获得距离最近的n个点或半径r内的点。
3. 常用特征点检测方法有2种：NARF（与描述子对应，用于某张深度图）和
4. Registration：不知道相对位姿的情况下合并几帧深度图像，首先应该找到图像之间的匹配点，然后通过位姿变换使所有匹配点的距离和最小（在同一坐标系下比较）。当距离小于某个值后，将所有帧3D点转换到同一坐标系下，如果某帧和其他帧都没有足够的匹配点将不能合并。PCL中包含了检测多个集合匹配和排除异常匹配及计算相对位姿的方法。
5. KdTree：k指k维点，3D云k=3，可以获得距离最近的n个点或半径r内的点，查找返回点在点云中的下标和与目标点的距离。3D用面分为2叉树，2D用线分为2叉树。

<https://blog.csdn.net/app_12062011/article/details/51986805>

1. Octree：可以获得距离最近的n个点或半径r内的点，还可以用于体素计算。
2. 用于压缩：如压缩体素中各点的颜色
3. 用于查找上面三种查找：OctreePointCloudSearch，返回下标和距离（体素没有距离），在有限正方体内需要设置分辨率。
4. 合并多个点云：OctreePointCloudChangeDetector，后面点云来时，只需添加新的体素。
5. Sample Consensus：如RANSAC，可以先随机采样最小n个点得到一个模型参数（如圆柱），在用个模型去尝试其他点，看是否在模型上，当点足够多时便可以说明这个模型是对的，这些点在该模型上，可将它们化为一类。
6. 描述子范围选择，若范围太大会考虑到两个曲面，太小则会受噪声影响
7. 特征：法向量、曲率、PFH、FPFH、VFH、NARF 、Moment of inertia and eccentricity based descriptors、RoPs (Rotational Projection Statistics)、GASD
8. How 3D Features work in PCL：
9. Estimating Surface Normals in a PointCloud
10. Normal Estimation Using Integral Images：积分图像是指从原点开始沿各轴积分得到的图像
11. Point Feature Histograms (PFH) descriptors
12. Fast Point Feature Histograms (FPFH) descriptors
13. Estimating VFH signatures for a set of points
14. How to extract NARF Features from a range image：
15. Moment of inertia and eccentricity based descriptors
16. RoPs (Rotational Projection Statistics) feature：
17. Globally Aligned Spatial Distribution (GASD) descriptors
18. Viewpoint Feature Histogram ([VFH])和Globally Aligned Spatial Distribution (GASD)特征适合作物体识别和pose估计。
19. 滤波器：常用1、2、3、6，滤波顺序1或3或6（可滤去噪声点）、3（降采样）
20. PassThrough：滤去某轴的值超出一定范围的点
21. VoxelGrid：体素滤波，使用体素下采样
22. StatisticalOutlierRemoval：统计点周围K个点的平均模长，若该点模长与该值超过一个阈值滤掉。
23. parametric model：使用模型滤波，如直线、平面，该模型可以通过RANSAC获得
24. Extracting indices：使用图像分割不断分割出不同物体并显示
25. Conditional or RadiusOutlier removal：C-自己设置滤波条件，可设置多个条件，如z大于有一个值；R-当某点半径r范围内点少于n个时滤去。
26. 3D点存储文件类型：PCD、PLY、STL、OBJ
27. 读写点云文件的2种方法：pcl::PCDReader reader; reader.read ("\*.pcd", \*cloud);和pcl::io：：load or save
28. 拼接点云的方法：把点和在一起（维度相同）可用+号，把fields（如维度）拼在一起，用pcl::concatenateFields (cloud\_a, n\_cloud\_b, p\_n\_cloud\_c);
29. 可以通过openNI等直接从摄像头读点云，通过HDL Grabber读激光雷达的数据，方便实时处理。
30. range\_image:
31. 从已知点云中得到一幅相当于相机拍摄的深度图
32. 查找深度图中边缘（可否用于检测房间的边缘）：物体及其阴影边缘，以及物体和阴影边缘的交界（深度图的特殊现象）
33. Recognition
34. 根据已有的模型识别点云中物体（先通过聚类等方式分割出来）
35. Implicit Shape Model：通过训练集来训练得到识别物体的模型，然后用于之后的识别
36. Hypothesis Verification for 3D Object Recognition：This tutorial aims at explaining how to do 3D object recognition in clutter by verifying model hypotheses in cluttered and heavily occluded 3D scenes.
37. Registration：将多帧图像注册到一个坐标系下

* 计算传感器位姿

1. 计算相对旋转和位移：PCL有多种计算匹配点和排除误匹配的方法。
2. 求特征点：有NARF, SIFT and FAST等
3. 求特征点描述子：NARF, FPFH, BRIEF or SIFT.
4. 特征匹配：文中有多种方法
5. Correspondences rejection：RANSAC或减少使用的比例
6. Transformation estimation：SVD、LM、ICP
7. Iterative Closest Point：直接迭代最近点
8. Feature based registration：提取匹配点后再求相对位姿
9. iterative closest point：计算两帧点云之间相对变换
10. incrementally register pairs of clouds：使用ICP跟踪帧的位姿变换，有提取特征的方法
11. interactive ICP：交互式界面，随意变换物体点云，然后计算变换矩阵
12. Normal Distributions Transform (NDT)：有相关论文，

* 计算物体位姿

1. 小物体的三维重建：首先要从每帧中分割出物体，还得具有计算两帧间相对位姿变换的能力。文中有相关前提条件。
2. Robust pose estimation of rigid objects：从杂乱和遮挡的环境中通过位姿变换匹配物体
3. Random Sample Consensus：
4. segmentation
5. Plane model segmentation：使用随机RANSAC方法得到模型平面进行分割
6. Cylinder model segmentation：分割圆柱体时先分割出圆柱两头
7. Euclidean Cluster Extraction：利用欧几里得距离分类
8. Region growing segmentation：利用曲率和法向量分割，假设同一物体曲率不会突变
9. Color-based region growing segmentation：利用颜色信息进行分割，也可以加上曲率信息
10. Min-Cut Based Segmentation：将点云分为物体和非物体两部分
11. Conditional Euclidean Clustering：在3）方法上加上了4）-6）的方法
12. Difference of Normals Based Segmentation：利用法向量分割，包含了其他方法
13. Clustering of Pointclouds into Supervoxels：用平面（3D的超像素）方格分割3D点云
14. Identifying ground returns:分割出地面和物体，有论文
15. Filtering a PointCloud using ModelOutlierRemoval：与1）2）相似，利用复杂些模型分割
16. surface
17. Smoothing and normal estimation based on polynomial reconstruction：使用Moving Least Squares (MLS)来光滑和去除噪声，多项式还能补充因为遮挡而缺失的部分
18. Construct a concave or convex hull polygon for a plane model：用多边体处理平面的情况，
19. Fast triangulation of unordered point clouds：用三角网格重建表面
20. Fitting trimmed B-splines to unordered point clouds：
21. Visualization
22. The CloudViewer：常用显示点云功能，复杂例程可以多线程
23. How to visualize a range image：用3D和深度颜色图（2D）两种方式展示深度图像
24. PCLVisualizer：it is also more powerful, offering features such as displaying normals, drawing shapes and multiple viewports.人机交互
25. PCLPlotter：画图多项式图或直方图等，包含多种参数输入方式，可以从excel直接读数
26. Visualization：简介页
27. Create a PCL visualizer in Qt with cmake
28. Create a PCL visualizer in Qt to colorize clouds：通过按钮改变点云颜色
29. Applications：综合前面的教程，完成更复杂的任务：
30. Aligning object templates to a point cloud：将在之前帧检测到物体的模型与当前帧重合，得到模型的位姿
31. Cluster Recognition and 6DOF Pose Estimation using VFH descriptors：通过训练来识别物体的位姿
32. Point Cloud Streaming to Mobile Devices with Real-time Visualization：点云从电脑到移动设备
33. Detecting people on a ground plane with RGB-D data：使用RGB-D相机检测人
34. GPU
35. Configuring your PC to use your Nvidia GPU with PCL：使用GPU加速
36. Using Kinfu Large Scale to generate a textured mesh：
37. Detecting people and their poses using PointCloud Library：
38. 1