#### 《机器视觉与应用》--姿态估计--说明文档

* **程序使用方法：**

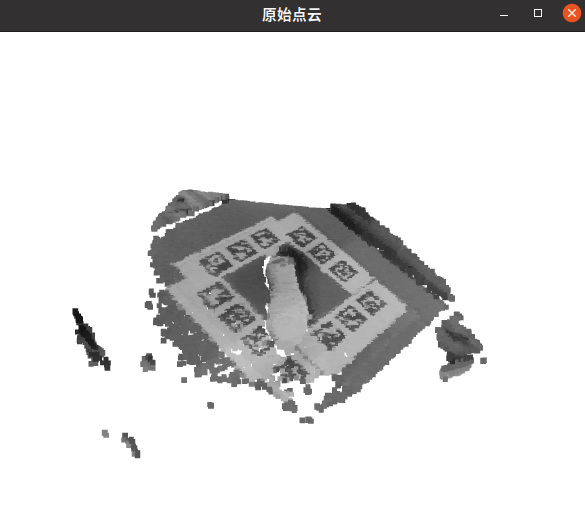
训练步骤：采用传统姿态估计算法，直接设置参数，无训练过程。

测试步骤：在o3d.io.read\_image("data/originaldata/image/1.jpg")中更改需要处理的rgb图像和对应的depth图像的路径，然后直接运行代码即可。

* **算法步骤介绍：**

1. 将彩色图与其对应的深度图转换为场景点云。
   1. 调用o3d.io.read\_image函数分别将rgb图像和对应的depth图像读入，调用create\_from\_sun\_format函数将两者整合成RGBD图像。
   2. 按照所给的相机内参在代码中进行定义，调用PinholeCameraIntrinsic函数将相机内参写入intrinsic中。
   3. 调用create\_from\_rgbd\_image函数，传入RGBD图像和相机内参后，得到RGBD图像对应的场景点云source\_pcd。

图片1对应的场景点云如下图所示：



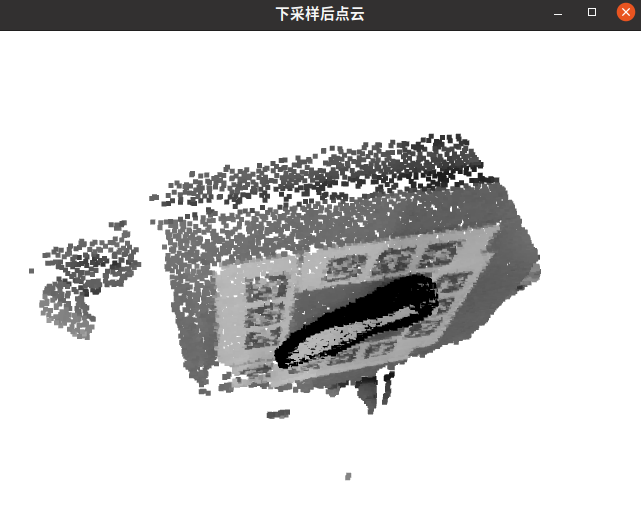
1. 读入模板点云
   1. 调用o3d.io.read\_point\_pcd函数将shoes.ply文件以xyzrgb格式读入，得到模板点云target。

模板点云如下图所示：



1. 对场景点云和模板点云进行体素下采样
   1. 体素下采样使用规则体素栅格从输入点云创建均匀下采样点云。
      1. 首先，将点云进行进行体素划分。
      2. 对所有非空体素，取体素内点云的质心作为该体素的点的位置。
   2. 调用voxel\_down\_sample函数进行体素下采样，其中体素大小设为0.007m^3。

图片1对应的下采样后的场景点云和模板点云如下图所示：



1. 使用RANSAC算法对体素降采样后的场景点云进行平面分割

**RANSAC**(**RA**ndom **SA**mple **C**onsensus,随机采样一致)算法是从一组含有“外点”(outliers)的数据中正确估计数学模型参数的迭代算法。其本质是是通过反复选择数据集去估计出模型，一直迭代到估计出认为比较好的模型。

具体的实现步骤可以分为以下几步：

1. 选择出可以估计出模型的最小数据集；
2. 使用这个数据集来计算出数据模型；
3. 将所有数据带入这个模型，计算出“内点”的数目；
4. 比较当前模型和之前推出的最好的模型的“内点“的数量，记录最大“内点”数的模型参数和“内点”数；
5. 重复1-4步，直到迭代结束或者当前模型已经足够好了。

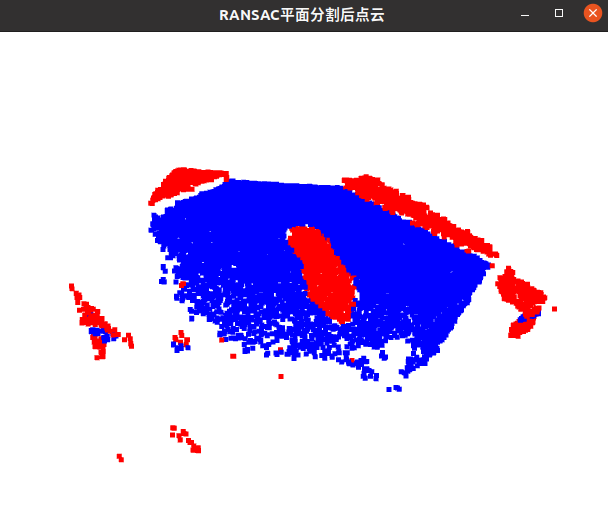
这里将鞋垫认为成平面的外点，将平板认为成平面的内点，需要去除内点保留外点，也就是去除鞋垫的背景。

其中的超参数设置：

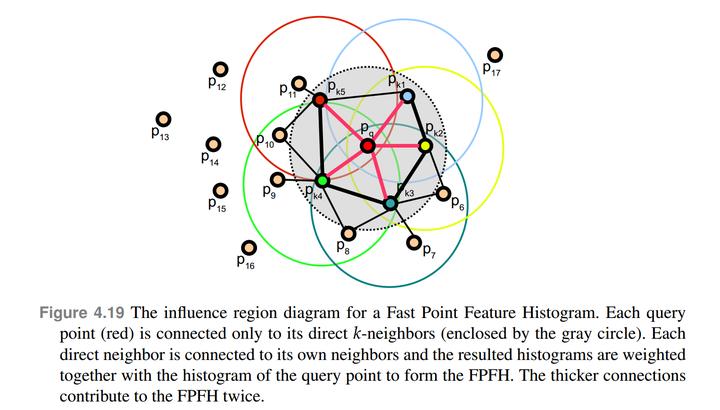
* + 1. 内点到平面模型的最大距离：0.008m
    2. 用于拟合平面的采样点数：20
    3. 最大迭代次数：100

最终得到场景点云去除平面后的点云not\_plane\_pcd

图片1对应的场景点云进行平面分割的结果（平面为内点，其余为外点）：

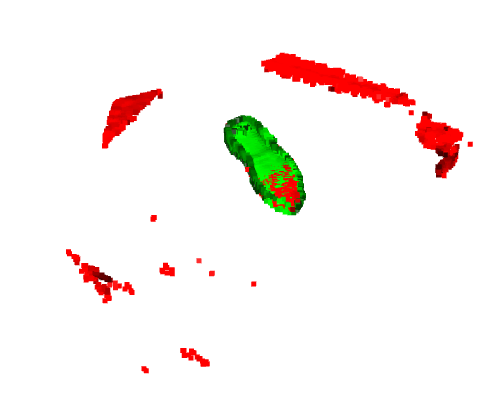
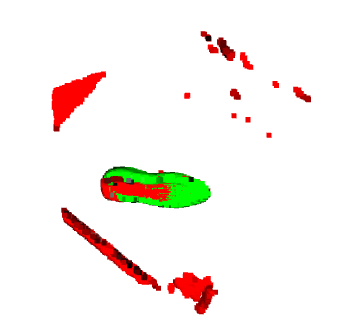


1. 对上一步得到的场景点云和下采样后的模板点云分别提取FPFH特征。
   1. FPFH特征是一个描述点的局部几何属性的33维的向量，在33维空间中进行最近邻查询可以返回具有近似几何结构的点.
   2. 为查询点求得它和其k邻域内每个点之间的三个特征元素值，然后统计成一个SPFH；
   3. 分别对k邻域中的每个点确定k邻域，按第一步分别形成自己的SPFH；
   4. 对邻域中的各个SPFH进行加权统计，生成该范围的FPFH



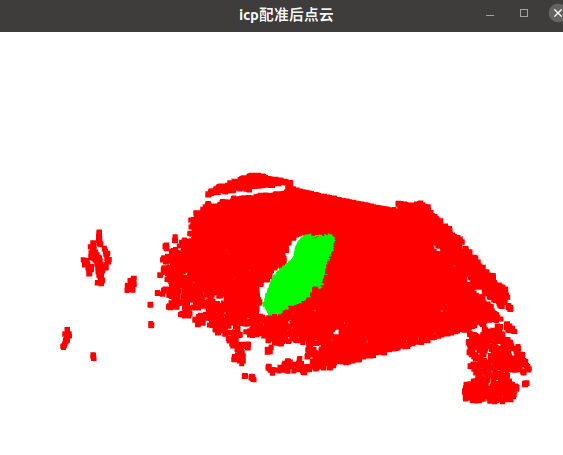
1. 对去除平面后的场景点云和下采样后的模板点云进行全局配准。
   1. 调用open3d.pipelines.registration.registration\_ransac\_based\_on\_feature\_matching函数。
   2. registration\_ransac\_based\_on\_feature\_matching 函数是 Open3D 中的一个函数，用于将两个点云进行配准。其中各个参数的意义如下：
      1. source: 待配准的源点云。
      2. target: 目标点云。
      3. source\_feature: 源点云的特征描述符，用于点云间的特征匹配。可以通过 compute\_fpfh\_feature 等函数来计算。
      4. target\_feature: 目标点云的特征描述符。
      5. distance\_threshold: 特征匹配的距离阈值，单位为米。
      6. ransac\_n: RANSAC 算法采样的点数。
      7. ransac\_t: RANSAC 算法的阈值，表示采样的点与模型之间的最大距离，单位为米。
      8. ransac\_iterations: RANSAC 算法的迭代次数。
      9. transformation\_estimation: 用于估计刚性变换的方法，例如mo li pin o TransformationEstimationPointToPoint 或 TransformationEstimationPointToPlane。
      10. criteria: 停止迭代的标准，例如最大迭代次数或者最小变换量。
      11. 这些参数用于控制点云配准的精度和速度。函数的返回值是一个 registration\_result 对象，其中包含了两个点云之间的变换矩阵和对齐后的点云。

图片1对应的场景点云与模板点云配准后的结果（去除平面后的场景点云与体素下采样后的模板点云）：

1. 将上一步全局配准得到的变换矩阵作为初值，再进行点到平面的icp配准。
   1. 使用点到平面（point-plane）误差度量的迭代最近点 (ICP) 算法已被证明比使用点到点（point-point）误差度量的算法收敛得更快。在 ICP 算法的每次迭代中，产生最小点到平面误差的相对位姿变化通常使用标准的非线性最小二乘法来解决。
2. 显示点云配准结果。

图片1对应的原始场景点云与模板点云最终配准后的结果：

* 结果分析与总结

经过对点云做一系列的处理得到了最终的配准结果，配准可视化结果和位姿变换矩阵见**附件：测试结果。**

每个测试用例配准用时在4s左右。

本项目为减少计算量，提高运算效率做了以下几点工作：

* 1. 第一步就对场景点云和模版点云进行体素降采样，可以在不失点云特征的前期下尽量减少后续操作处理的点云点数。
  2. 第二步的平面分割操作的是第一步降采样后的点云，并且去除平面的这一操作可以有效利用“鞋垫放在平板上”这一先验信息来最大限度地去除鞋垫以外的点，可为后续配准进一步精简计算量，还可减少平面点对配准过程的干扰，以尽量减少迭代次数。
  3. 去除平面后的场景点云直接用ICP算法做点到平面的配准的话，当场景点云与模板点云距离较远时候，配准结果很不理想。所以需要在此之前进行全局配准，将场景点云和模板点云变换到较近的距离，以此位姿为初值再进行ICP局部配准，这样可以得到一个很好的配准结果。
  4. 先进行全局配准，再进行局部配准的操作可以大幅减少两者的迭代次数，以最快得到一个较好的配准结果。