**计算机图形学2022秋**

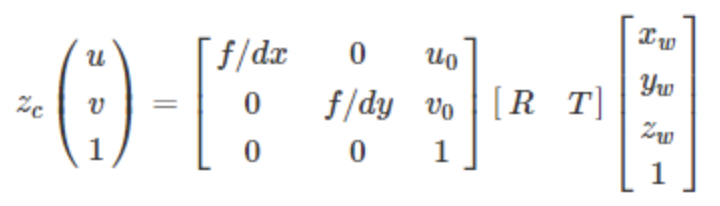
编程作业：三维重建

本次作业需要需实现三维重建中的点云拼接算法，即给定深度图和RGB图片，相机的位姿和内参，还原对应场景的三维点云，了解相机内外参的基本概念，掌握三维坐标系之间的转换和重建的基本思想。

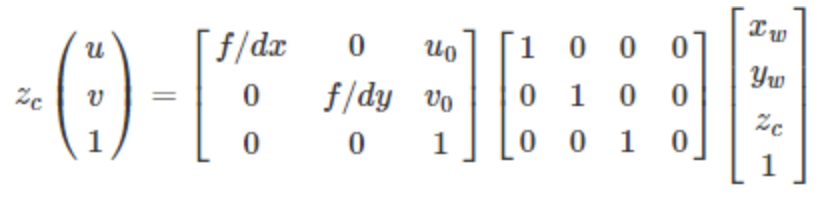
1. 环境：
2. Linux或Windows系统皆可，编程语言为C++或python；
3. C++可能用到库有opencv，pcl，Eigen等等，python可能用到的库有numpy，opencv，pcl等等
4. 算法流程：

数据提供5张深度图和RGB图，及相机的内外参。

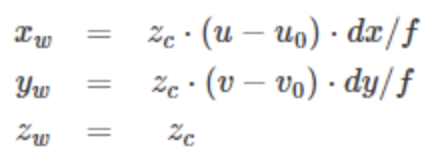
1. 读取相机内参(fx,fy,cx,cy)、外参数(R,T)，将外参的四元数转化成R矩阵；
2. 读取相机对应的深度图和RGB图，注意opencv读取RGB图的结果是BGR格式；
3. 对深度图进行处理，除以实际的尺度depth\_scale=1000；
4. 遍历所有的像素，利用公式计算对应的3D点位置。



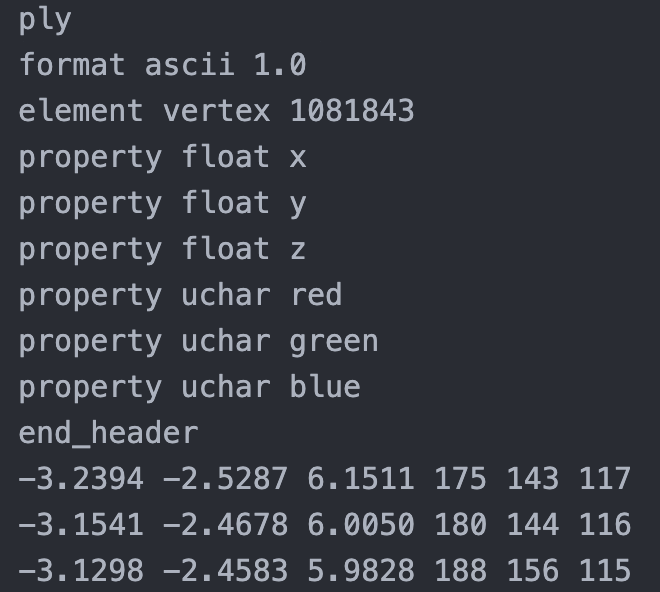
在相机有外参的情况下，可以先默认把相机放在世界坐标系原点处，转化成如下公式：



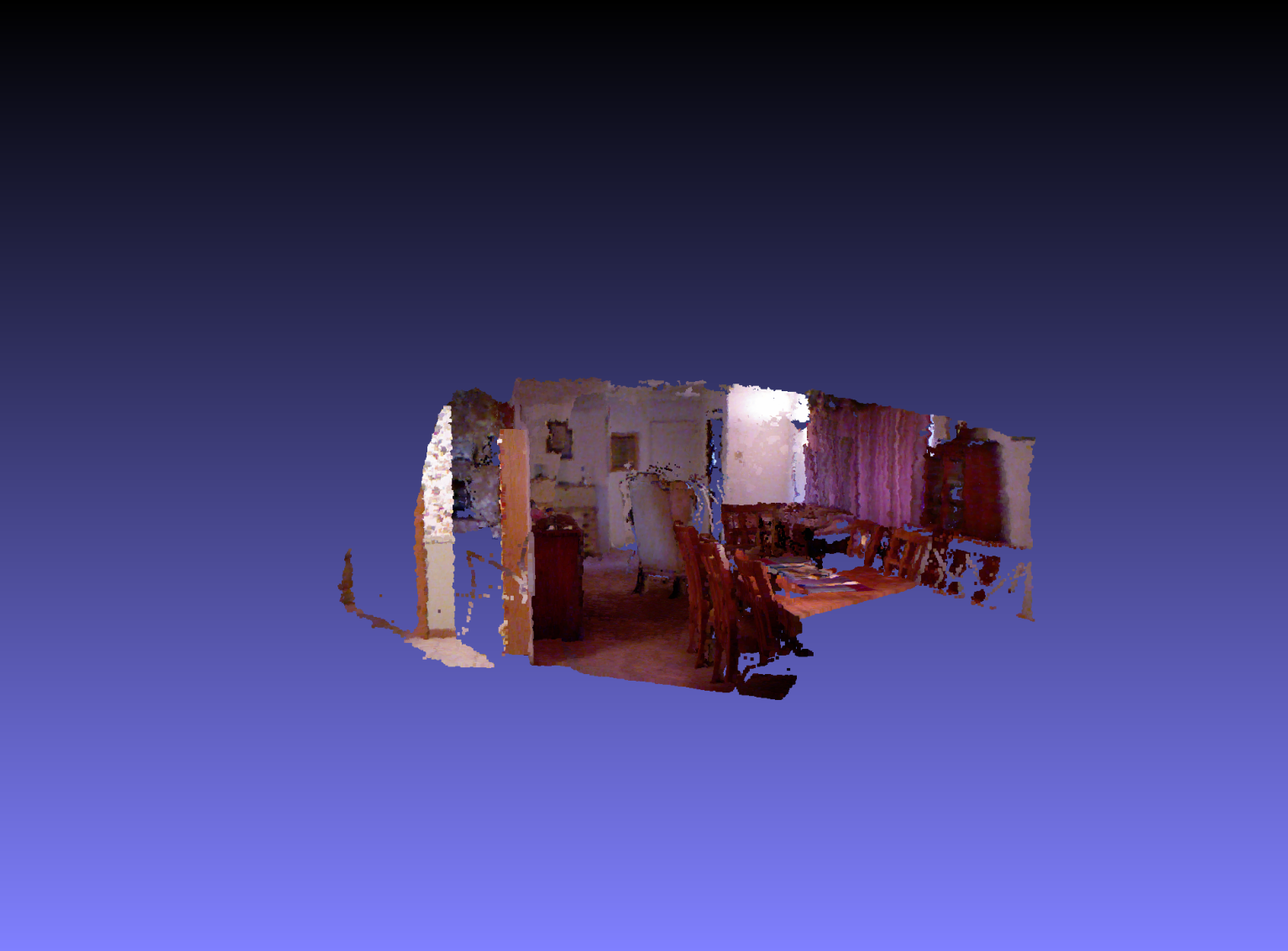
便可以得到每个像素(u,v)和对应的3D点坐标(x\_w,y\_w,z\_w)，再利用相机外参数4\*4的[R T]与[x\_w,y\_w,z\_w,1]相乘即可转换到实际位置(也可旋转和平移分开进行)



1. 重复上述1-4步，将每一张图获得的点云累加；
2. 对重建的点云进行可视化(截图保存)，可以利用第三方库可视化或保存成ply等格式用meshlab软件可视化；
3. 直接拼接累加的点云往往存在噪声，需对最终的点云进行点云滤波(如统计滤波、半径滤波、体素滤波等)，这一步可使用第三方库如pcl或者open3d，对点云进行过滤，并可视化结果，需提供对比图片，可以对比不同参数的过滤结果，选择效果最好的；
4. 将最终对点云保存成ply，可以将点云信息直接写入文件，保存示例如下，也可用第三方库保存；



如果实现正确，在滤波前你应该得到类似的结果：



如果实现正确，在滤波后你应该得到类似的结果：



实验报告中请提供滤波前后的点云可视化结果截图，并对比滤波前后点云数量(需截图)，可对比不同参数，不同滤波器的结果，分析当前数据不同。

1. 要求：

基础部分(100分，折算为80分)：

1. (60分)能正常重建出点云，且颜色正确，并能正确保存及可视化；
2. (20分)对点云进行了滤波，且实现结果正确，有滤波前后对比结果；

加分部分：

1. (10分)滤波步骤自行实现了第三方库的相关滤波方法；
2. (10分)提出了自己设计的去除噪声点的方法，并且具有实验验证；
3. (10分)利用多线程或其他方法对整个算法进行了加速，并有前后时间开销对比；
4. (10分)能对比不同滤波器的效果，不同参数的对比，需有实验截图，并分析得出当前数据最适合的滤波器和参数，提供分析过程；

实验文档撰写(20分)

文件命名：“学号\_姓名.zip”，需包含**1.实验报告.pdf**(环境配置，算法实现，实验结果，过滤前后的点云可视化截图，点云数量对比，实验分析，总结），**2.代码源文件**，**3.可视化结果.ply**。代码会进行查重(含互联网上的代码)，请同学们独立完成。