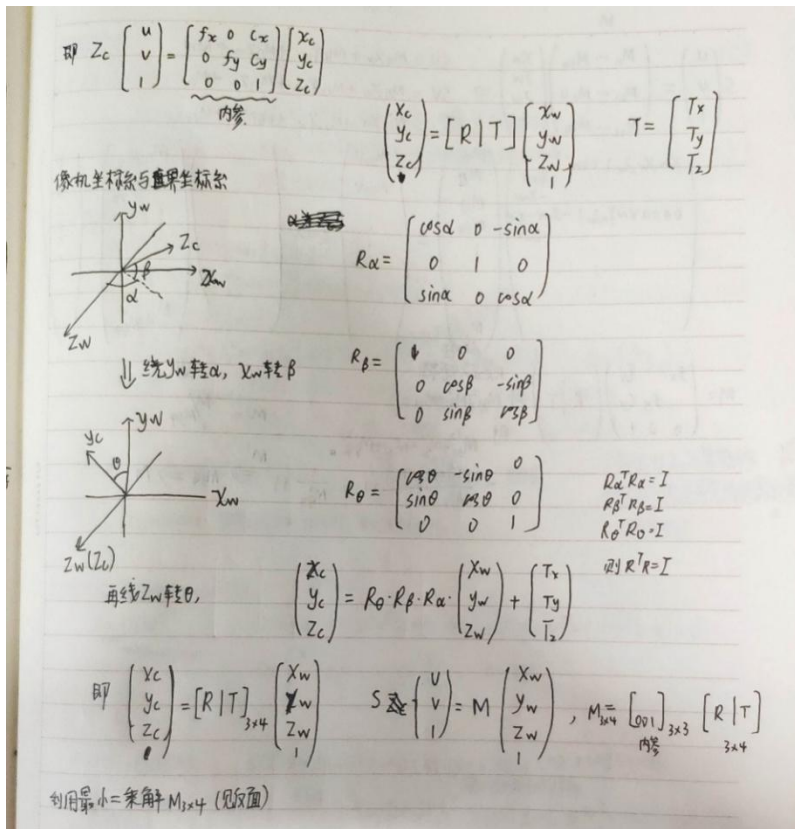
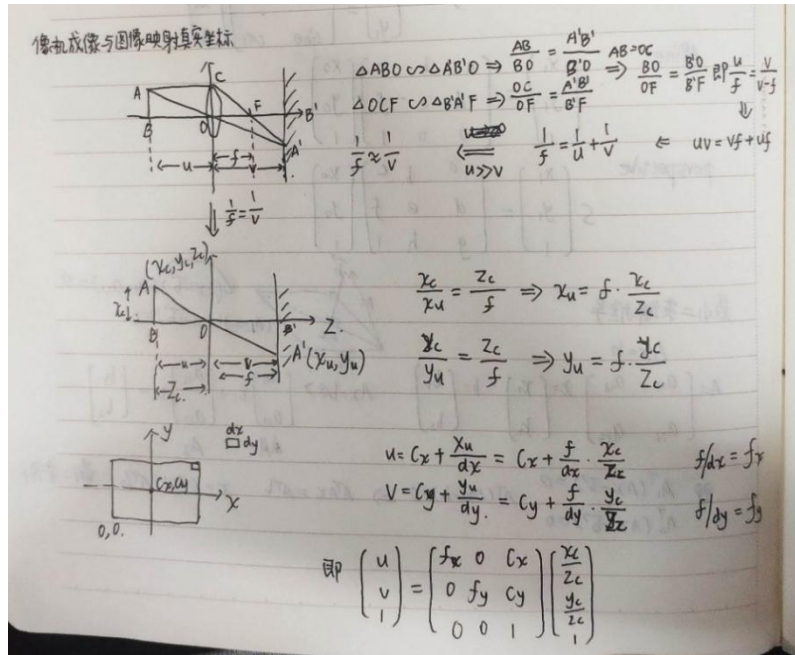


# 三维重建

小组成员：陈豪博、杨晓峰、周长颖、龙润灵、周学荣

## 一、算法概述

三维重建的理论原理如下图：



可以清楚看出，上图完成了三维世界坐标与二维图片的像素坐标的一一映射。在具体实现时的重建过程实际上是将三维空间的每一个点 $[x; y; z]$ 通过各个图片对应的变换矩阵  $M_{34}$ （即  $M_{3 \times 4}$ ，上述原理中的  $M$ ）映射到各个图片中去，然后根据此点是否在重建目标上确定是否保留此点，从而得到目标物体的三维点云，同时可根据此点在各个图片中的颜色确定此点在三维空间中的颜色。

## 二、文件总述

本项目由以下几部分组成：

1. bdb：存放所有图像，其矫正图像及分割结果
2. src：源码，包括 main.h 头文件；main.cpp 主文件；HsvSplit.cpp 图片分割的文件；World2Img.cpp 世界坐标系到图片坐标系的映射，主要运算部分；undistortion.cpp 图片畸变矫正函数；draw3D.py 三维图像绘制
3. bin：可执行文件，使用时在**主目录**键入 “ ./bin/3Dconstruction”
4. build：make 目录,CMakeList.txt 文件
5. 3dSlice：存放三维图形的水平每一层，即切片 slice，用于快速校验结果，无实际意义
6. params：3dPoint.txt 存放 main.cpp 产生的三维点云信息，用于 draw.py 进行图像绘制，考虑 c++ 三维绘图需 opengl，故使用 python 的 matplotlib 绘图；RotationMatrices.txt 存放 matlab 导出的所有图像的旋转矩阵外参，依照图像序号存放，被 main.cpp 读取并使用；TranslationVector.txt 存放 matlab 导出的所有图像的平移外参，同上

## 三、算法实现

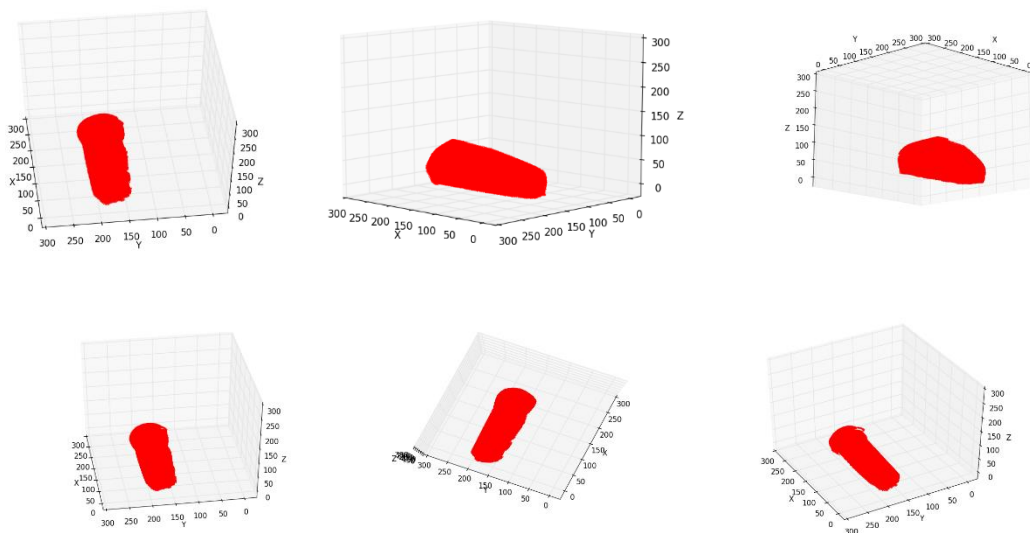
1. 利用 matlab 对相机进行标定，并导出其内参和对应每张图片的外参，内存直接使用于 main.cpp 的 getIntParam() 函数中，外参分别存放在 RotationMatrices.txt 和 TranslationVector.txt；
2. 利用 matlab 的相机畸变系数，使用畸变矫正方法（undistortion.cpp）进行畸变矫正；
3. 利用图像分割方法（HsvSplit.cpp）对图片进行 HSV 分割，并将结果二值

化，即有目标的像素为 255；

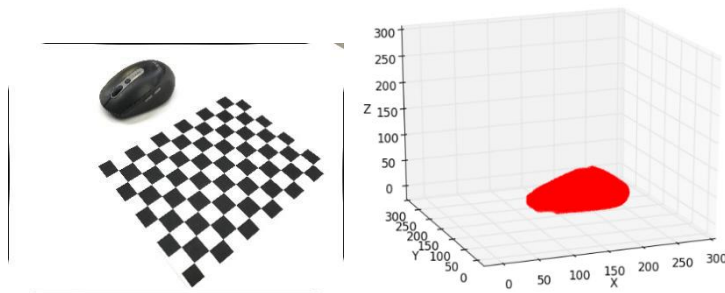
4. 设置合理的真实坐标区间，使用映射方法（World2Img.cpp）将真实坐标系映射到各个图片中；
5. 采用投票机制，对映射结果进行投票，将符合条件的点保留下来；

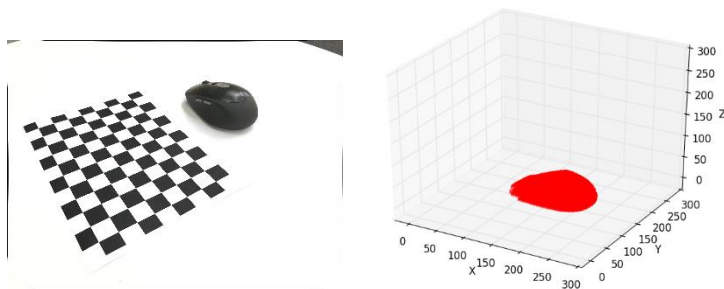
## 四、结果展示

下面是我们对 draw3D.py 中绘制的函数进行旋转，并选择不同的角度进行图片保存，值的注意的是在绘图前可以根据电脑的性能选择不同的步长对点云进行过滤（step 位于 read\_data 函数中），每隔几个点绘制一个，以提高流畅度。默认值为 3。



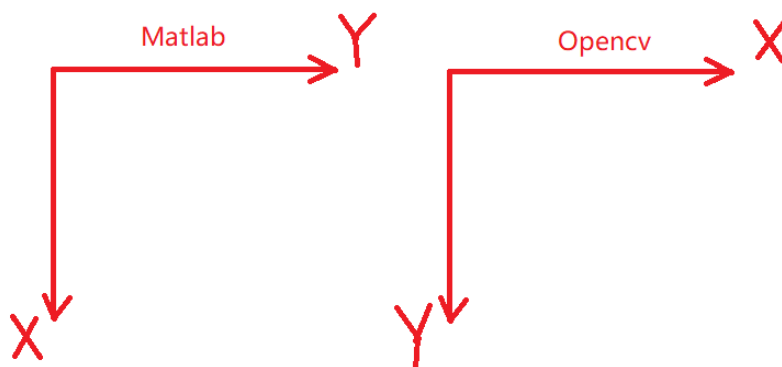
可以看出，因为原图拍摄的原因，导致三维重建出的模型并不是很好，但是整体的杯子形状已经很不错了，而且因为拍摄光线、灯光下的阴影以及被子透明度影响，即使在优化了分割程序的情况下，结果还是不够完美。为了验证原图影响，我们借用了别的数据进行重建，部分效果如下：





## 六、特别说明

1. 我们使用 Matlab 进行标定，后使用 opencv 进行三维重建，而这两个图像的存储体系是完全不同的，如下：



我们事前只对内参矩阵进行了转置，并未对旋转矩阵和投影后的  $u$ ,  $v$  进行判断，这导致我们在完成了最初版的算法实现后代码迟迟不能 work，后面经一个同学提醒才考虑到 opencv 坐标系的不同，同时又使用 python 对别组的数据（鼠标）进行了重新实现（为了避免数据对算法的影响），这才发现问题所在，最后又重回 C++，改出了最终代码。

2. 在实现坐标映射时，因为需要对三维空间中某一范围内所有点的四维向量  $[x; y; z; 1]$  进行矩阵运算，因此可将所有点组成一个  $4*n$  矩阵，进行一次矩阵运算，从而省去循环计算的复杂，达到运算加速效果。
3. 因为原图影响，14 张图只有 9 张能被同时映射参与投票，经过分析原因是拍摄的照片有横屏和竖屏两种格式，标定的时候未经过仔细思考便统一旋转成横屏图像，导致相机外参部分出错，因此在建模的时候删除了 5 张竖屏旋转的照片，用 9 张而不是 14 张图片建模，得到了合理的结果。为了验证这一点，我们使用鼠标数据，所有图都参与了投票，所以可以确

定是数据的问题。

## 七、使用说明

1. 环境 ubuntu/C++/OpenCV/python
2. 在主目录键入 “./bin/3Dconstrusction” ， 获取点云数据 ./params/3dPoint.txt（提交文件中已含有，因此可不运行此步）
3. 在主目录键入” python ./src/draw3D.py” 绘制三维图（可根据需要调整其中 step 变量值，上面有提到原因，同时需要安装 matplotlib 绘画库）。