

## Lab05

姓名：万晨阳

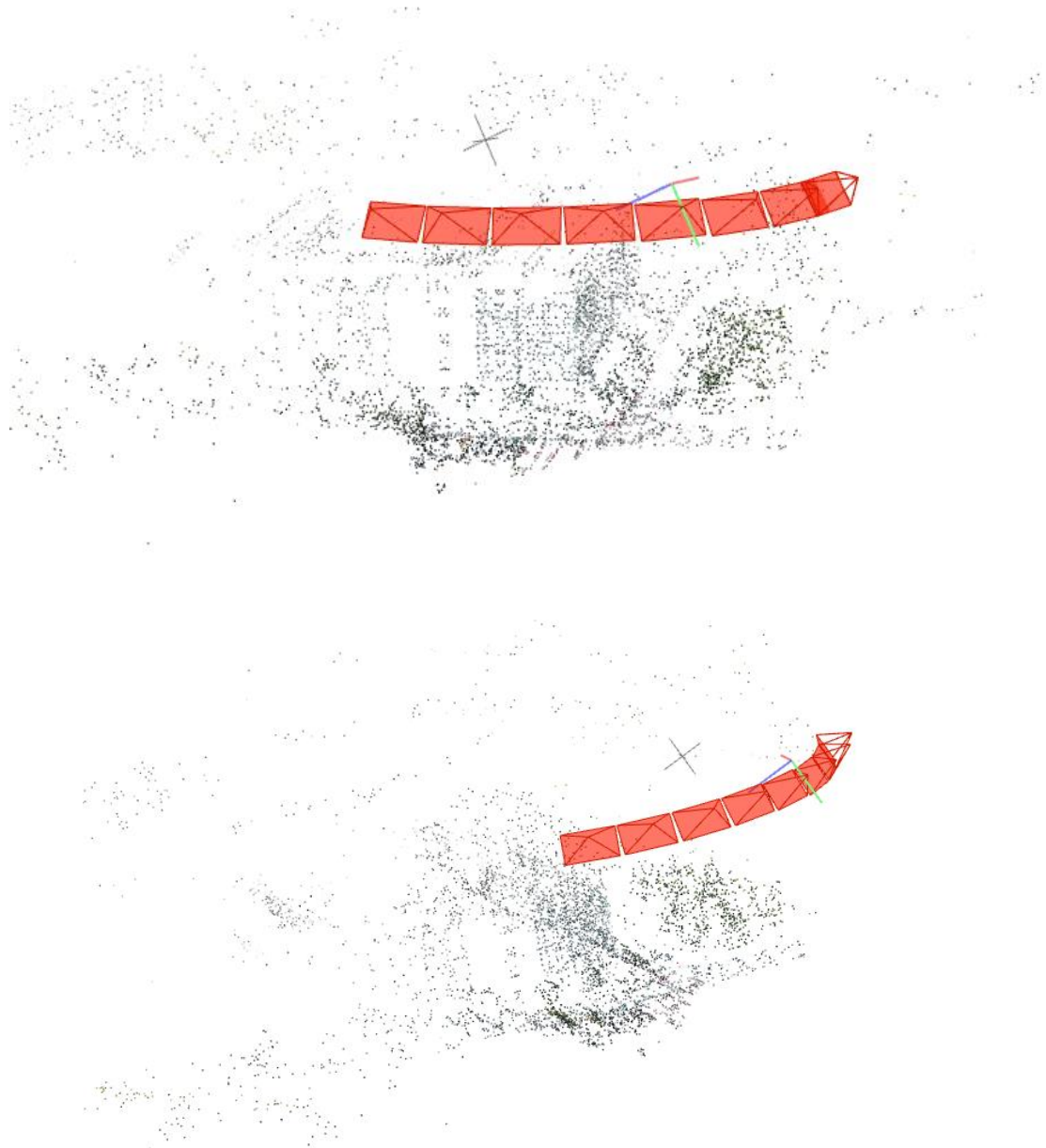
学号：3210105327

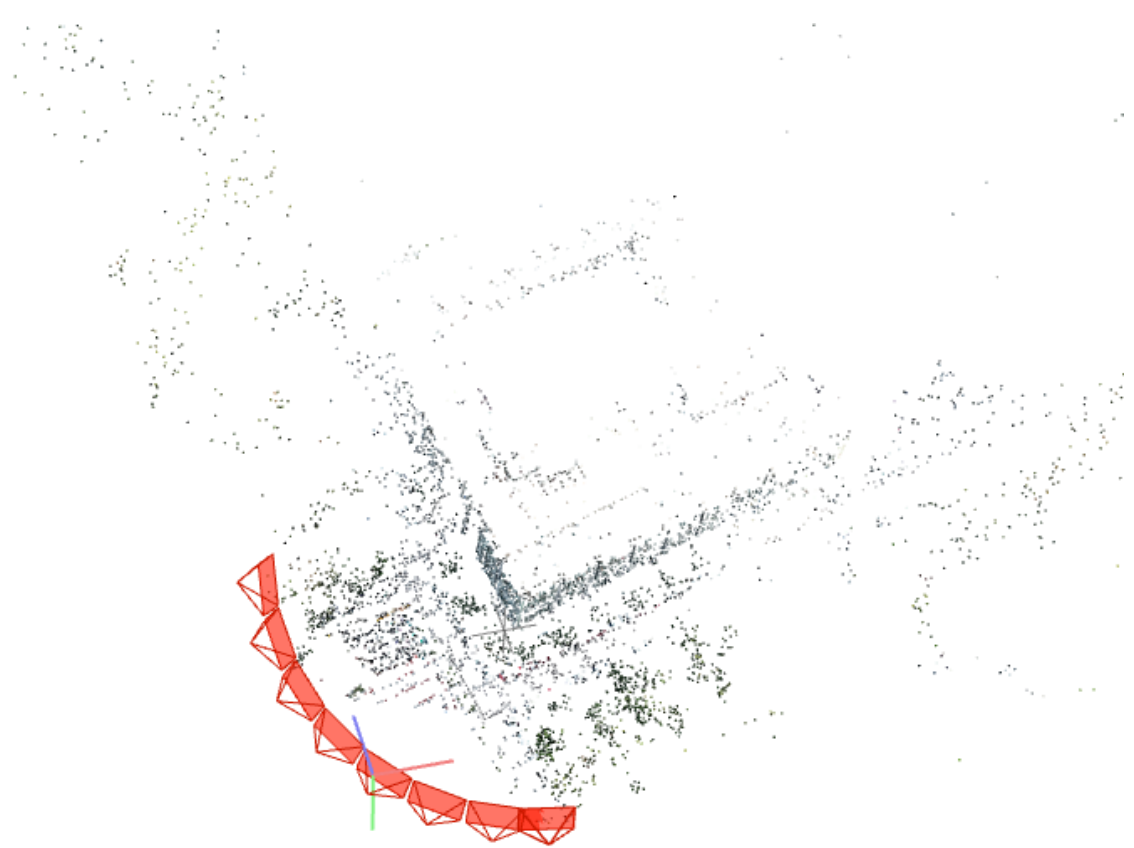
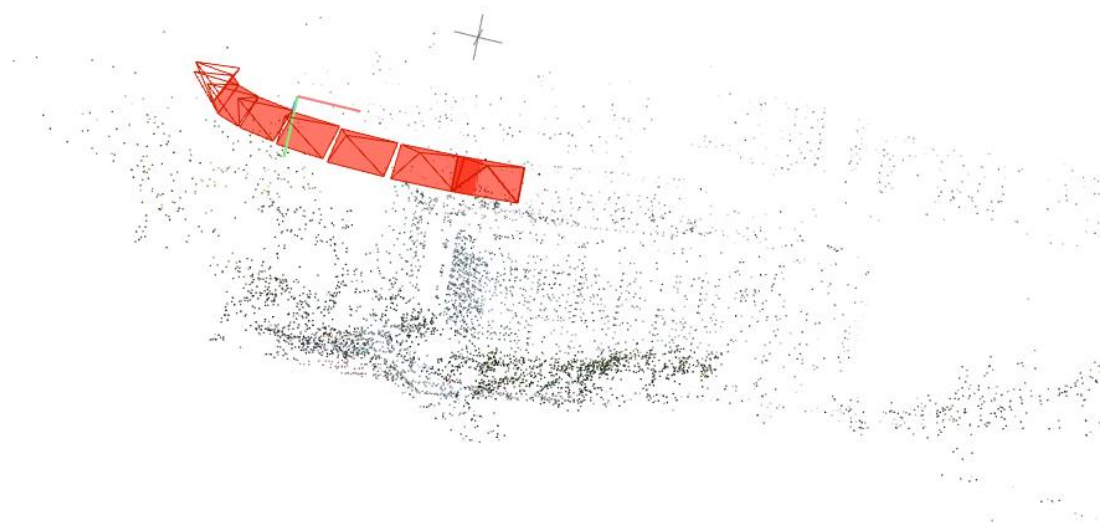
作业要求：

1. 查阅相关文档，完成稀疏重建
2. 提取 00001.jpg 的二维关键点、特征描述子。在图片上可视化二维关键点
3. 在 matplotlib 中可视化三维稀疏模型点云，将和 00001.jpg 关联的关键点用特别的颜色标出
4. 将截图与回答填写到 lab5\_name\_id.docx 上传 pdf

### 一、 稀疏重建结果

按照特征提取->特征匹配->稀疏重建流程，得到的结果如下：





## 二、 可视化 00001.jpg 的二维关键点

首先，我将模型导出为.txt 文件，然后分析了其中各个文件的构成。在 images.txt 中包含了 2D 图像的特征点信息。因此，我通过编写脚本完成了对于 txt 文件的解析，提取了其关于第一张图（00001.jpg）的特征点信息，然后利用 matplotlib 在图像上进行了标注。

脚本中的关键代码部分如下：

```
1. interested_points = []
2. points = image_points_data[5].split(' ')
3. print(points)
4. for i in range(len(points)):
5.     if (i+1) % 3 == 0 and points[i] != '-1':
6.         # if (i+1) % 3 == 0:
7.             interested_points.append(np.array((float(points[i-2]), float(points[i-1]))))
8. interested_points = np.array(interested_points)
```

最终结果如下：



我注意到在 COLMAP 中也有内置的特征点可视化功能（Database Management -> Processing -> Show image），这里将两者进行一下比较。右图为我可视化结果，左图为 COLMAP 内置的可视化结果，显示的都是图像中提取出的所有关键点。





可以看到两个结果是一致的。

另外，通过解析得到的模型可以发现，图像中并不是所有的关键点都在稀疏的 3D 模型找到匹配（部分特征点 POINT3D\_ID=-1 意味着没有在 3D 场景中出现），下图展示了图像 00001.jpg 在三维模型中得到匹配的关键点：



### 三、 可视化三维模型并标记出 00001. jpg 的点

对于这一任务，我首先通过对于导出的 points3D.txt 进行解析，得到三维场景中所有点的坐标，然后根据其中存储的 2D-3D 匹配关系，判断每个关键点是否存在于图像 00001.jpg 中，最后使用matplotlib进行三维场景的可视化，将和 00001.jpg 关联的关键点用红色标出，其他点为蓝色。脚本中的关键部分如下：

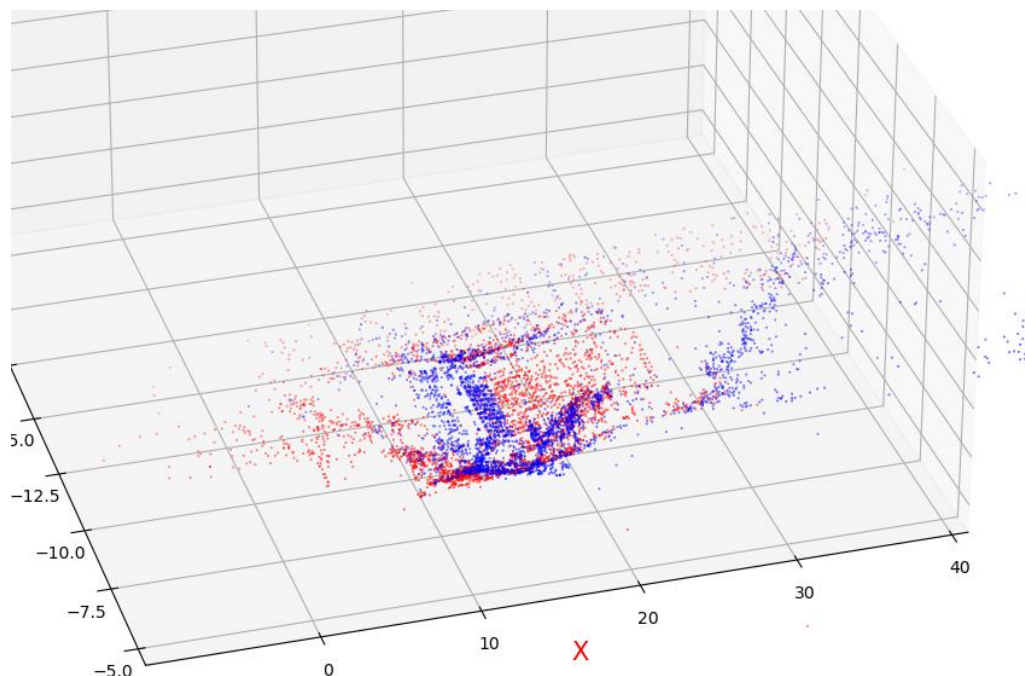
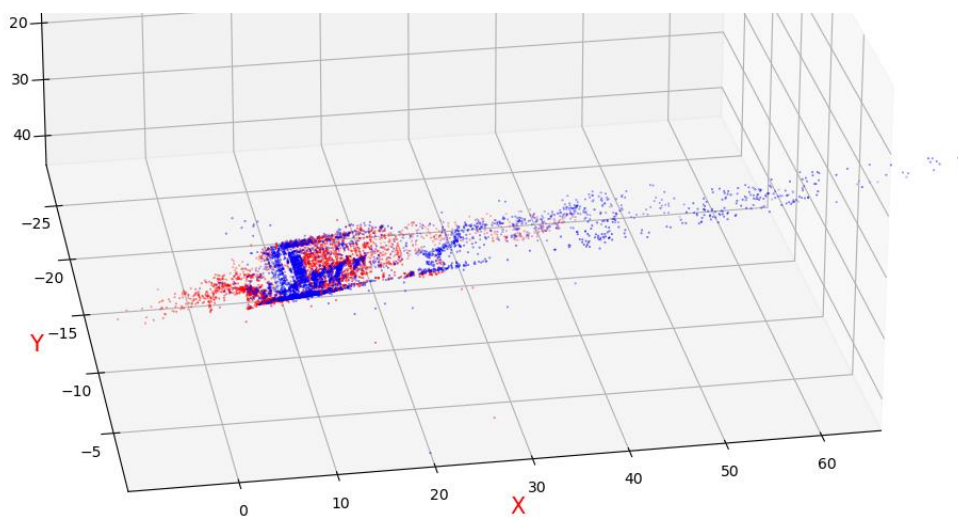
```
1. with open('./points3D.txt', 'r') as f:
2.     points_data = f.readlines()
3.
4. interested_points = []
5. other_points = []
6.
7. # divide the 3d points into 2 groups
8. for i in range(3, len(points_data)):
9.     point = points_data[i].split(' ')
10.    pos = point[1:4]
11.    for k in range(len(pos)):
12.        pos[k] = float(pos[k])
13.    for j in range(8, len(point)):
14.        if '1' == point[j]:
15.            flag = 1
16.            break
```

```

17.     if flag:
18.         interested_points.append(pos)
19.     else:
20.         other_points.append(pos)
21.     flag = 0
22.
23. # my scatter plot function
24. scatter_3d(np.array(interested_points), np.array(other_points))

```

结果如下：



可以看到由于是以第一张图象（相机）的坐标系为世界坐标系，所以画出来的点云整体存在一定的倾斜，但结果是正确的。

#### 四、COLMAP 中三种相机模型 Pinhole, Radial, OPENCV 之间有什么区别？

PINHOLE 模型：这是最简单的相机模型，假设光线通过相机的一个点（针孔）进入，然后投影到成像平面上。Pinhole 模型没有考虑透镜畸变，如果我们事先认为图像是无畸变的可以使用这种模型。那么 COLMAP 中 SIMPLE\_PINHOLE 使用一个焦距参数即  $f$ ，PINHOLE 使用两个焦距参数  $f_x$  和  $f_y$ 。在一些情况下这种模型不够准确，特别是在使用具有较大畸变的相机时。

RADIAL 模型：Radial 模型考虑了透镜的径向畸变。SIMPLE\_RADIAL 用了一个畸变参数，RADIAL 用了两个畸变参数，这两个模型都是 OPENCV 模型的简化模型。如果内参未知且每张图像具有不同的相机标定，那这种模型应该是我们的首选。

OPENCV 模型：COLMAP 中的 OpenCV 模型是一种更为复杂的相机模型，通常包括切向畸变和更多的畸变参数。在我们知道先验的相机内参的时候，应该使用这 OPENCV 相机模型。我们可以选择让 COLMAP 自己估计内参，也可以将自己知道的内参告诉 COLMAP（但是 COLMAP 只能共享一套内参，也即不能指定多个相机模型），让 COLMAP 去估计外参。如果每张图片有不同的内参的话，自动重建可能会失败。