Task1

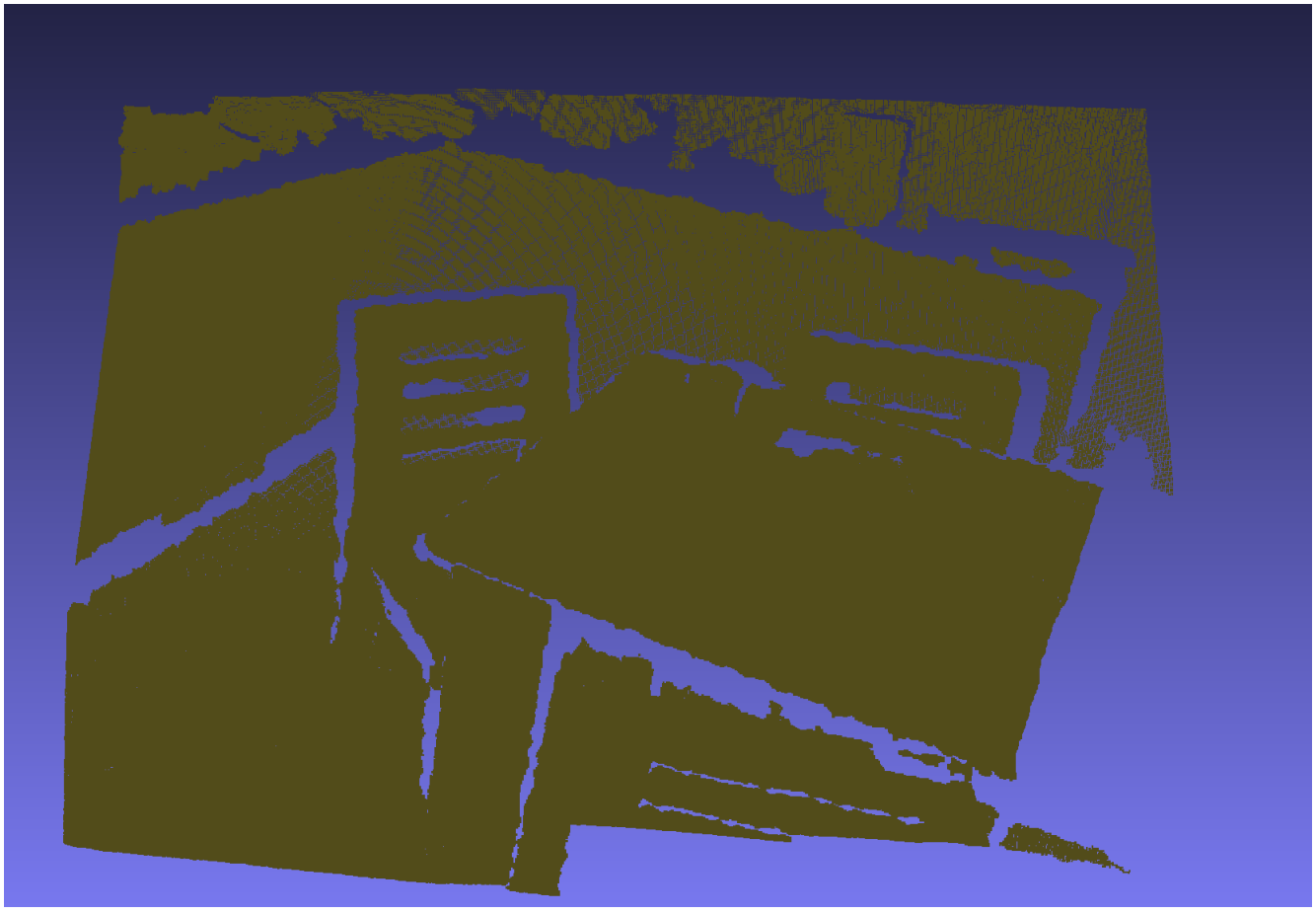
这一步中，主要区分所给的相机位姿矩阵与计算机视觉课上提及的相机外参矩阵的区别。实际上，两者是互逆的过程。相机外参矩阵描述了世界坐标向相机坐标的转换，相机位姿矩阵描述了相机坐标到世界坐标的转换。弄清楚这个关系后，即可写出表达式：

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = cam\_intr \times cam\_pose^{-1} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{1}$$

现已知像素坐标(可以利用对深度图高和宽网格化来获取u，v，而深度图中uv位置对应的值就是z)，对这个过程逆向进行即可得到世界坐标。

具体地，我先手动把像素坐标转化到相机坐标系，在用矩阵乘法转换到世界坐标系。前一步使用手动而不是求逆的方法，是因为内参矩阵大小很小，手动计算比求逆精确度更高，计算更快。

得到的点云如图：



接下来只需要在每遍历一张图片时记录下此次坐标的最大最小值，取全局最大和最小即可得到 TSDF体素场的范围。

```
Volume bounds: [[-2.9715983   3.80868932]
 [-1.91094864  1.02850464]
 [ 0.29656917  3.90544781]]
```

Task2

先初始化TSDFVolume对象的一些属性，根据vol_bnds划分体素场，创建网格。
然后在integrate中把创建的网格坐标一步步转化成像素坐标，即先转化成世界坐标，在转化成相机坐标，最后转化成像素坐标。其中，在转化为世界坐标时，注意体素网格的划分实际上选取了vol_bnds的$(x_{min}, y_{min}, z_{min})$作为原点，所以要多一步平移操作。另外，在转换到像素坐标后，要进行归一化。然后，对像素坐标取整并选出合法的坐标（即落在图片范围内的TSDF采样点）去深度图中采样深度。

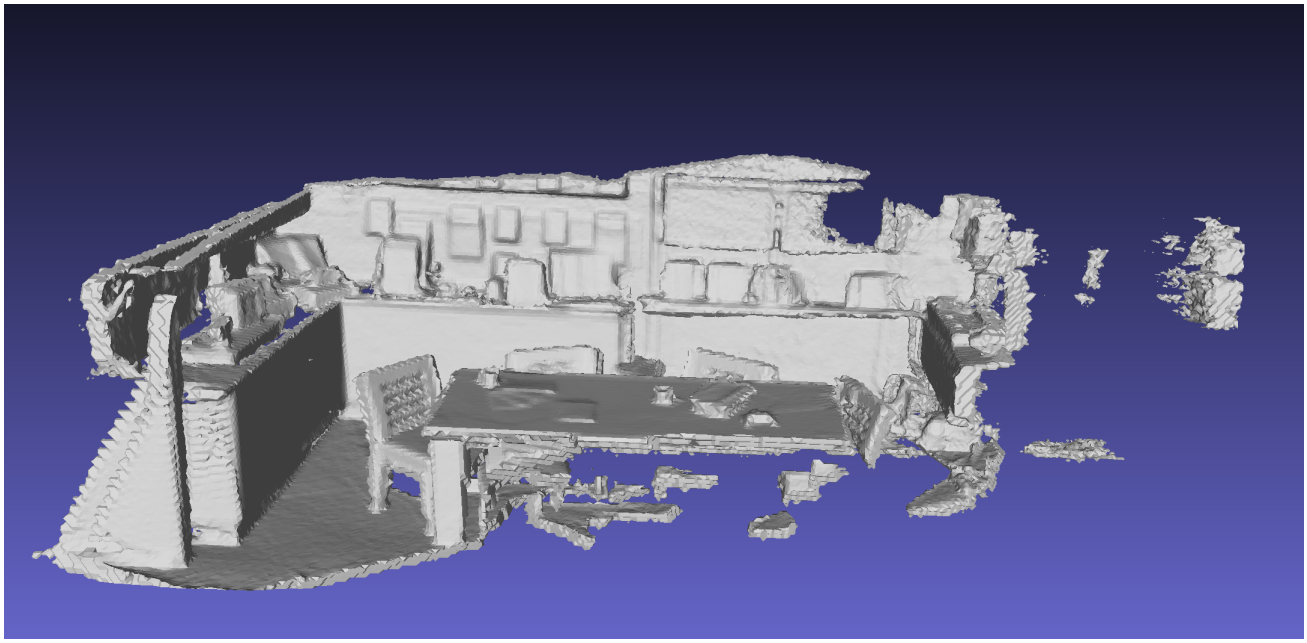
Task3

根据定义，先计算出sdf值，进而算出tsdf值。

Task4

先选出在这一轮中要更新tsdf值的点，根据算出的TSDF值，带入加权公式得到新的TSDF值，并更新新的权重。待计算完所有的TSDF值后，使用measure.marching_cubes抽取mesh，并用trimesh生成ply文件。

效果如图：



Additional task

选择实现带颜色的TSDf fusion。

在TSDFVolume中，初始化一个与网格点TSDF值形状相同的三维矩阵来记录每个点的颜色值。这里为了储存以及integrate中计算方便，将rgb三通道值压缩到一个通道中。具体的方法是：

$$rgb(i) = b(i) * 256^2 + g(i) * 256 + r(i)$$

再还原回三通道时，

$$b(i) = \lfloor (rgb(i) \div 256^2) \rfloor$$

$$g(i) = \lfloor ((rgb(i) - b(i) * 256^2) \div 256) \rfloor$$

$$r(i) = rgb(i) - b(i) * 256^2 - g(i) * 256$$

以$b(i)$为例，$rgb(i) = 256^2 * [b(i) + \frac{g(i)}{256} + \frac{r(i)}{256^2}]$，显然 $\frac{g(i)}{256} < 1$ and $\frac{r(i)}{256^2} < 1$，所以 $\lfloor (rgb(i) \div 256^2) \rfloor$就是$b(i)$。同样的，可以还原出$g(i)$，$r(i)$。

使用这样的颜色表示方式，在每一帧循环中，首先把输入的图片转换成这样的表示格式，然后与计算TSDF值一样，选出要更新点的颜色信息，加权融合每一个通道，更新颜色即可。

最大的bug出现在使用trimesh把颜色信息一同写入ply文件中。

一开始我直接将循环得到的tsdf_vol.color作为颜色信息写入到ply中，结果MeshLab可视化始终无颜色显示。查看ply文件发现，文件头中并不含有property uchar red,property uchar green,property uchar blue,property uchar alpha 等信息，再打印中间过程发现传入的color不是(N,3)的形状。

因此，应该根据marching cubes产生的vertices处理循环得到的color，提取出每一个vertex对应的体素网格颜色，然后处理成(N,3)形状，传入trimesh写入ply文件，才能在MeshLab中可视化带颜色的重建场景。

如图：

Reference

TSDF算法原理1

TSDF算法原理2

ply文件格式