



# Homework13 3D Reconstruction

胡成成 2101210578

## Question

根据RGB-D数据图，生成点云数据（文件），并显示。

## Answer

### PLY简介

- RGB-D数据图是ply格式的文件。PLY文件格式是Stanford大学开发的一套**三维mesh模型数据格式**，图形学领域内非常多著名的模型数据，比方Stanford的三维扫描数据库（当中包含非常多文章中会见到的Happy Buddha, Dragon, Bunny兔子），Geogia Tech的大型几何模型库。
- PLY多边形文件格式的开发目标是建立一套针对多边形模型的，结构简单可是可以满足大多数图形应用须要的模式格式，并且它同意以ASCII码格式或二进制形式存储文件。PLY的开发人员希望。这样一套既简单又灵活的文件格式，可以帮助开发人员避免反复开发文件格式的问题。然而因为各种各样的原因，在工业领域内，新的文件格式仍然在不断的出现，可是在图形学的研究领域中，PLY还是种经常使用且关键的文件格式。
- PLY作为一种多边形模型数据格式，不同于三维引擎中经常使用的场景图文件格式和脚本文件。每一个PLY文件仅仅用于描写叙述一个多边形模型对象(Object)，该模型对象能够通过诸如顶点、面等数据进行描写叙述，每一类这种数据被称作一种元素(Element)。相比于现代的三维引擎中所用到的各种复杂格式。PLY实在是种简单的不能再简单的文件格式，可是假设细致研究就会发现，就像设计者所说的，这对于绝大多数的图形应用来说已经是足够用了。

### PLY结构

- PLY的文件结构简单：文件头加上元素数据列表。当中文件头中以行为单位描写叙述文件类型、格式与版本号、元素类型、元素的属性等，然后就依据在文件头中所列出元素类型的顺序及其属性，依次记录各个元素的属性数据。
- 典型的PLY文件结构：        头部        顶点列表        面片列表        （其它元素列表）
- 头部是一系列以回车结尾的文本行。用来描写叙述文件的剩余部分。
- 以下是一个立方体的完整ASCII描写叙述。大括号里的凝视不是文件的一部分，它们是这个样例的注解。文件里的凝视一般在“comment”开始的关键词定义行里。

```
<span style="font-size:16px;">ply
format  ascii  1.0  {  ascii/二进制。格式版本号数  }
comment  made  by  anonymous  {  凝视关键词说明，像其它行一样  }
comment  this  file  is  a  cube
element  vertex  8  {  定义“vertex”（顶点）元素，在文件里有8个  }
property  float32  x  {  顶点包括浮点坐标“x”}
property  float32  y  {  y  坐标相同是一个顶点属性  }
property  float32  z  {  z  也是坐标  }
element  face  6  {  在文件中有6个“face”（面片）  }
property  list  uint8  int32  vertex_index  {  “vertex_indices”（顶点索引）是一列整数  }
end_header  {  划定头部结尾  }
0  0  0  {  顶点列表的开始  }
0  0  1
0  1  1
0  1  0
1  0  0
1  0  1
1  1  1
1  1  0
4  0  1  2  3  {  面片列表开始  }
4  7  6  5  4
4  0  4  5  1
4  1  5  6  2
4  2  6  7  3
4  3  7  4  0 </span>
```

## 生成点云代码

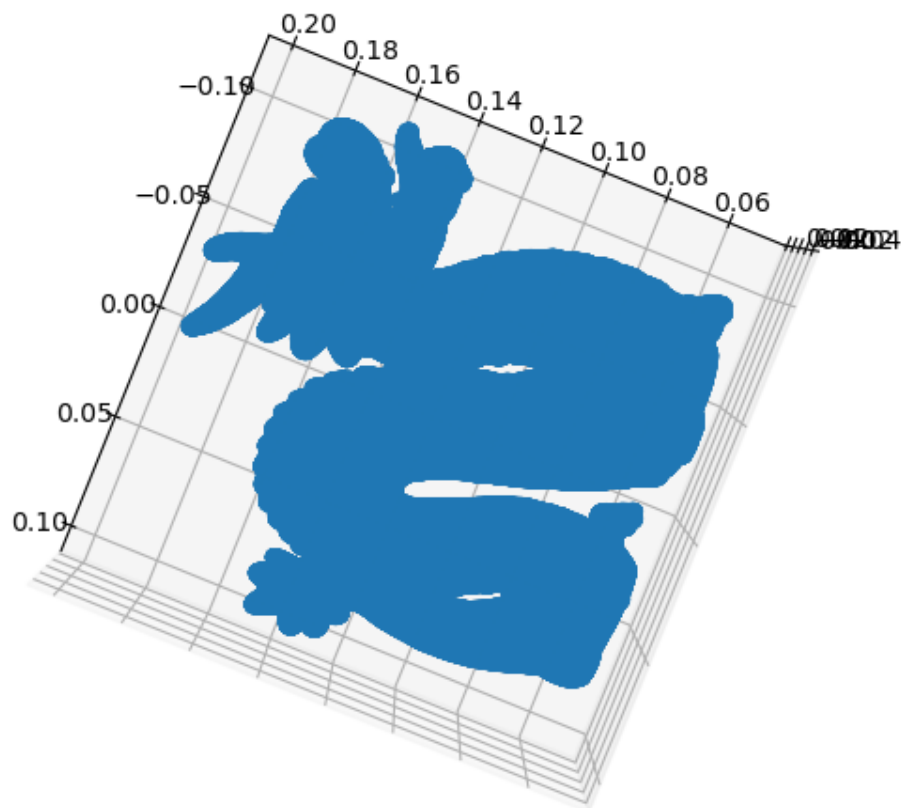
```
#!/usr/bin/env python
# -*- coding: UTF-8 -*-
import matplotlib.pyplot as plt
from plyfile import *
from mpl_toolkits.mplot3d import Axes3D

# plydata = PlyData.read('./bun045.ply')
plydata = PlyData.read('./dragon.ply')
# plydata = PlyData.read('./ellipsoid.ply')
```

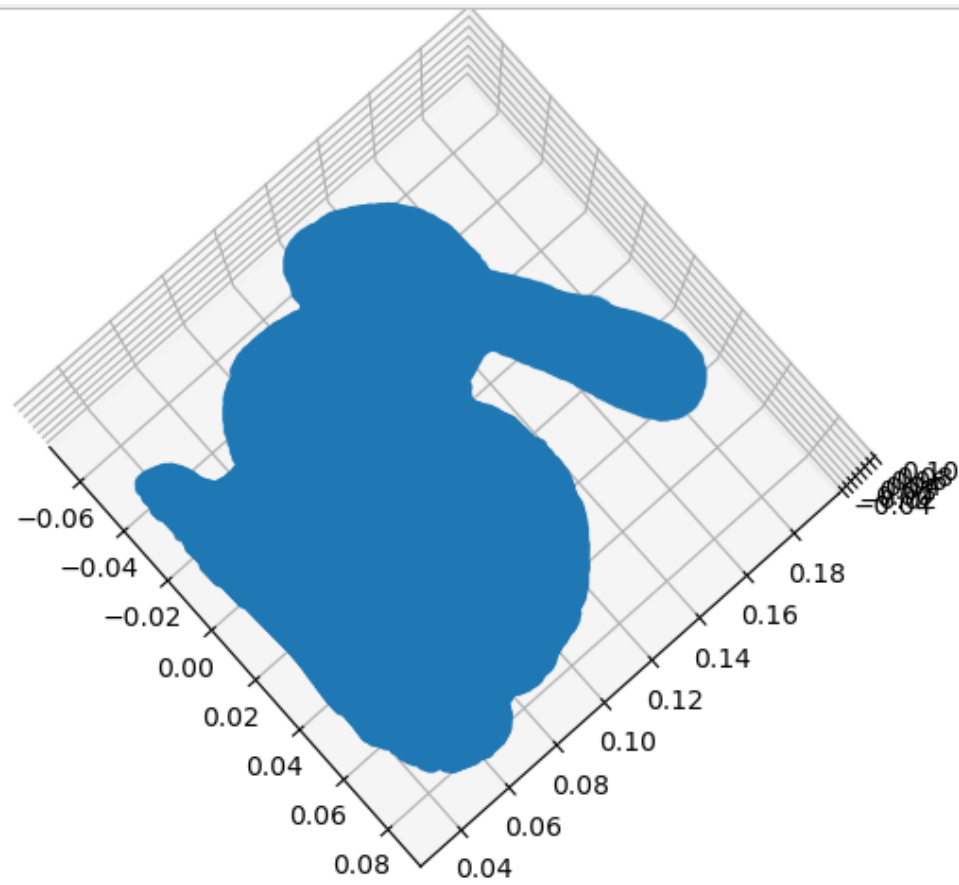
```
xlist = plydata['vertex']['x']
ylist = plydata['vertex']['y']
zlist = plydata['vertex']['z']

fig = plt.figure()
ax = Axes3D(fig)
ax.scatter(xlist, ylist, zlist)
plt.show()
```

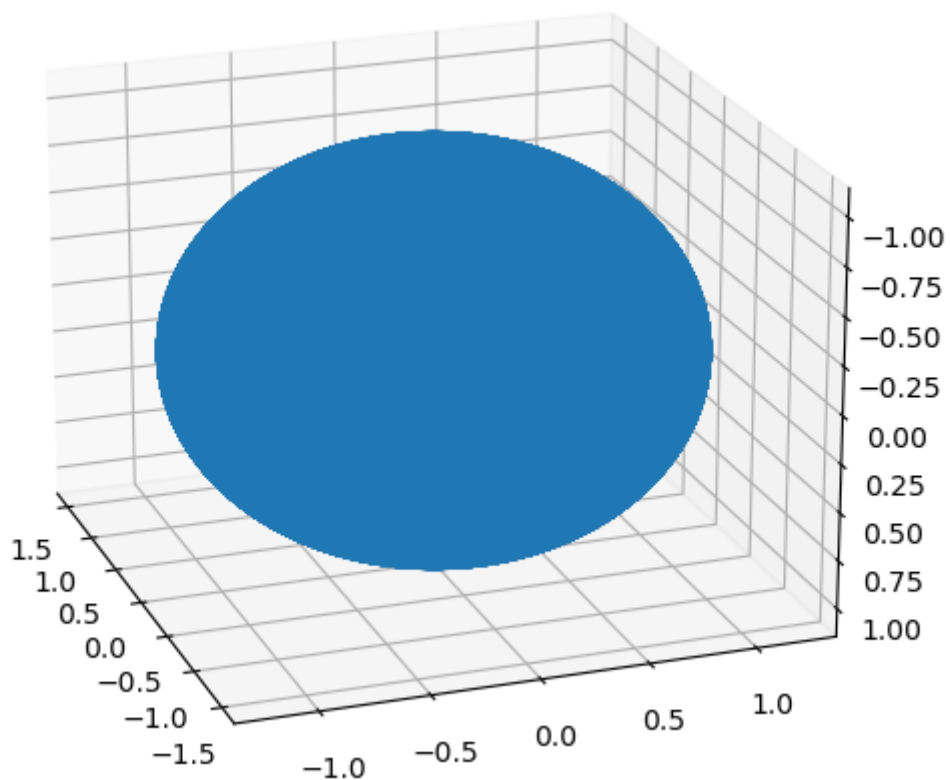
- 输入dragon.ply，显示结果为：



- 输入bun045.ply，显示结果为：



- 输入ellipsoid.ply，显示结果为：



## 总结

点云是存储三维信息表面的一种形式，主要固定的格式记录采样的点，通过 `Axes3D` 能够将其展示出来。