深蓝学院
shenlanxueyuan.com

3D点云第一章作业分享

主讲人 张点堃

第一题

●Perform PCA for the 40 objects, visualize it.

●思路：从PCA原理出发，计算点云矩阵的特征值/SVD。

●注：需要注意点云数据矩阵的维度，计算出的主方向应该是3维，而不是N维。

第一题

深蓝学院 shenlanxueyuan.com

- 代码实现：3D可视化
- 可以创建一个o3d.geometry.LineSet()对象，并在空间中沿主方向定义3个点，实现在3维视图中主方向的可视化。

```
#三维主方向可视化
pca_vector=o3d.geometry.LineSet() #创建一个线集
lines = [[0, 1], [1, 2]]  #定义两条线(实际上是共线的)
colors = [[0, 0, 1] for i in range(len(lines))] #定义每条线的颜色
vector_points = np.array([origin_point-point_cloud_vector, origin_point, origin_point+point_cloud_vector], dtype=np.float32) #录入三个点，其均在主方向上
pca_vector.lines = o3d.utility.Vector2iVector(lines)
pca_vector.colors = o3d.utility.Vector3dVector(colors)
pca_vector.points = o3d.utility.Vector3dVector(vector_points)

vis = o3d.visualization.Visualizer()
vis.create_window(window_name='Principal direction', width=1920, height=1080, left=10, top=10, visible=True) #定义画图窗口
vis.get_render_option().point_size = 2  # 设置点的大小
vis.add_geometry(point_cloud_o3d)
vis.add_geometry(pca_vector)
vis.create_window()
vis.run() #绘图
vis.destroy_window() #关闭窗口
```

第一题

深蓝学院 shenlanxueyuan.com

- 代码实现：2D投影可视化

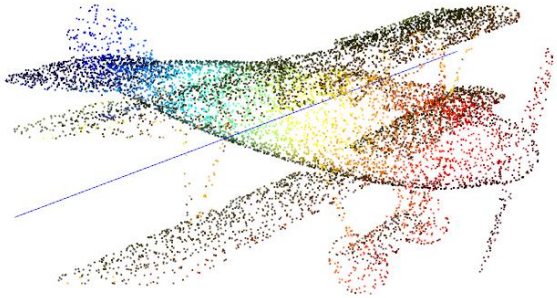
- 根据投影原理，直接将点云数据与主方向和次主方向点积可以得到投影值。增加一个空维度，使其变成三维空间中同一平面的点，便于可视化。

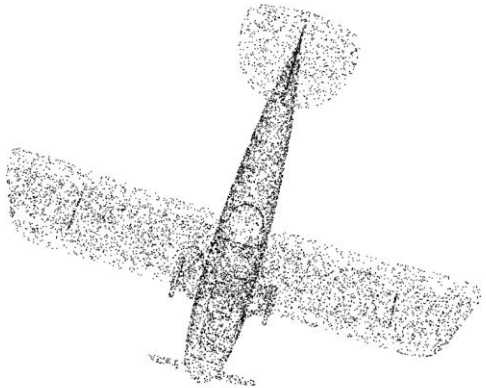
```
#在主方向和次主方向做投影，形成二维可视化图片
proj_vector=v[:, 0:2] #取前两个向量
proj_point=np.matmul(points,proj_vector)
proj_point=np.concatenate((proj_point,np.zeros([point_num,1])),axis=1) #添加一个统一的Z轴数据，形成3D点云，方便可视化
point_cloud_proj_pca = o3d.geometry.PointCloud(o3d.utility.Vector3dVector(proj_point))
vis_pca = o3d.visualization.Visualizer()
vis_pca.create_window(window_name='PCA projection', width=1920, height=1080, left=10, top=10,
                      visible=True) # 定义画图窗口
vis_pca.get_render_option().point_size = 2 # 设置点的大小
vis_pca.add_geometry(point_cloud_proj_pca)
vis_pca.create_window()
vis_pca.run() # 绘图
vis_pca.destroy_window() # 关闭窗口
```

第一题

深蓝学院
shenlanxueyuan.com

●可视化结果





第一题

深蓝学院
shenlanxueyuan.com

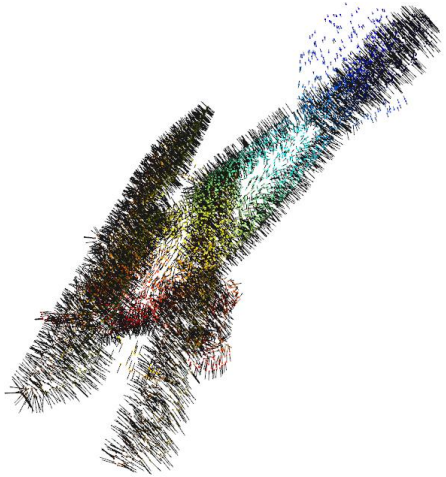
- Perform surface normal estimation for each point of each object, visualize it.

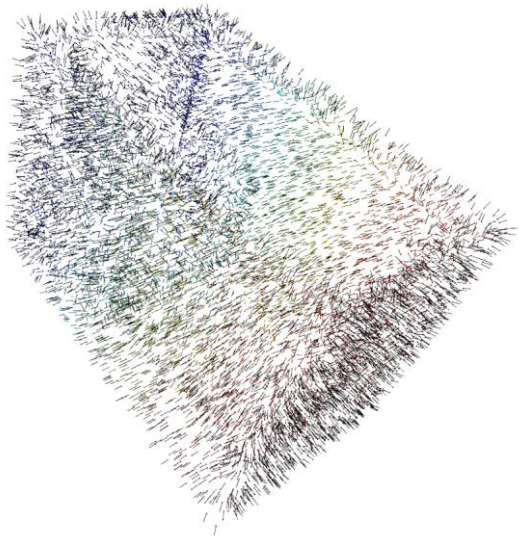
- 思路：第一章还没有讲点云的邻域搜索算法，可以直接使用open3d给出的方法o3d.geometry.KDTreeSearchParamKNN()。

第一题

深蓝学院
shenlanxueyuan.com

●可视化结果





第二题

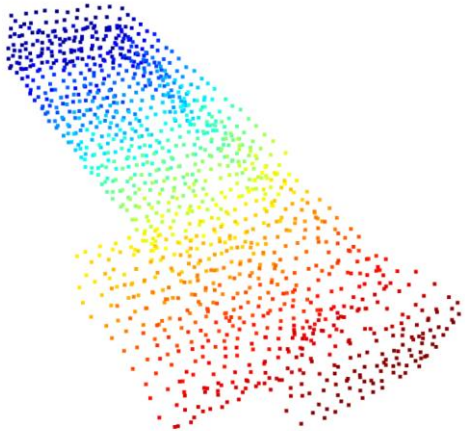
- Downsample each object using voxel grid downsampling(exact, both centroid & random). Visualize the results.

- **思路：根据体素降采样原理：1）划分网格；2）创建容器（hash表）；3）向容器中压入/弹出点；4）弹出容器中剩余的所有点。弹出的方法可以是centroid或random。**

第二题

深蓝学院
shenlanxueyuan.com

- 可视化结果：体素降采样



第三题

深蓝学院
shenlanxueyuan.com

- Perform depth upsampling / completion for the validation dataset

- 思路：根据Bilateral Filter原理，对深度图进行滤波，可以使用RGB/灰度图信息作为额外的权重。

- 由于深度图是稀疏的，因此在权重分配上需要单独做一些处理。

第三题

$$BF[I]_{\mathbf{p}} = \frac{1}{W_{\mathbf{p}}} \sum_{\mathbf{q} \in \mathcal{S}} G_{\sigma_s}(\|\mathbf{p} - \mathbf{q}\|)\, G_{\sigma_r}(I_{\mathbf{p}} - I_{\mathbf{q}})\, I_{\mathbf{q}}$$

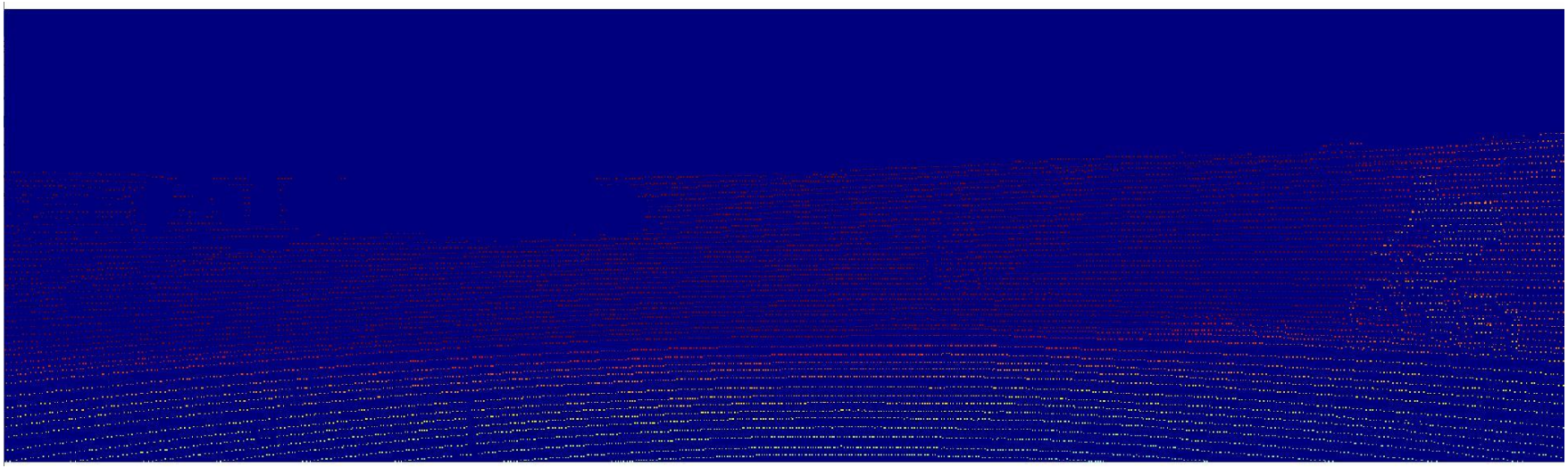
$$W_{\mathbf{p}} = \sum_{\mathbf{q} \in \mathcal{S}} G_{\sigma_s}(\|\mathbf{p} - \mathbf{q}\|)\, G_{\sigma_r}(I_{\mathbf{p}} - I_{\mathbf{q}})$$

- 使用RBG/灰度图的信息作为额外的权重(替换Ip-Iq)。
- 将等式最右端的加权对象Iq（图像灰度值）更换为该点的深度值。
- 稀疏性：没有深度的点（深度为0/-1）不应该影响周围点的加权求和。

第三题

深蓝学院
shenlanxueyuan.com

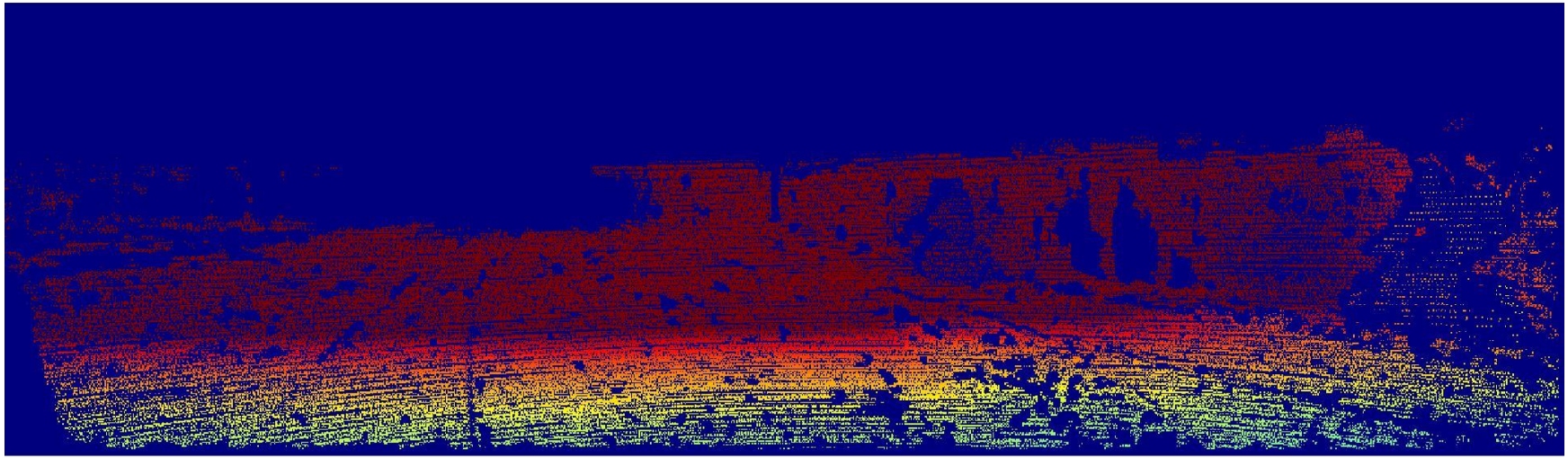
●可视化结果：输入深度图



第三题

深蓝学院
shenlanxueyuan.com

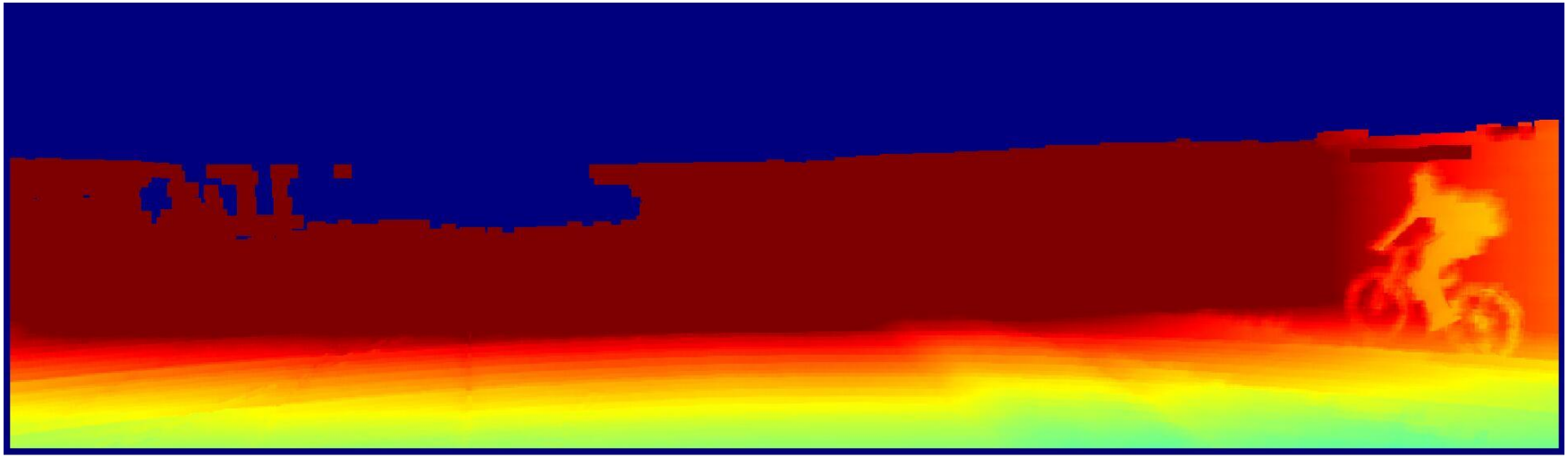
●可视化结果：深度图GT



第三题

深蓝学院
shenlanxueyuan.com

●可视化结果：深度图滤波结果



第三题

深蓝学院
shenlanxueyuan.com

- 量化结果：From KITTI depth Evaluation（左：滤波后结果，右：原始数据结果，仅供参考）

mean mae: 0.528272
min mae: 0.209994
max mae: 1.850485
mean rmse: 1.900804
min rmse: 0.612379
max rmse: 7.132728

```
mean mae: 2.41848
mean rmse: 6.01417
mean inverse mae: 0.0093934
mean inverse rmse: 0.023794
mean log mae: nan
mean log rmse: -nan
mean scale invariant log: -nan
mean abs relative: 0.10948
mean squared relative: 0.0628753
```

总结与经验

- 第一次作业的很多同学较多时间都花在了熟悉相关的库，数据的使用以及相关的数据格式等问题上，但详细了解一下相关的函数以及对应的数据格式可以避免以后再次踩坑。
- 前三个必做作业提交了的同学我看完成情况都比较好，问题不大。
- Open3d的Visualizer好像在显示深度图的时候有距离限制（我个人遇到），过大的距离显示不出来。可以直接使用draw_geometries()方法绘制完整的深度图。
- 双边滤波由于进行类似卷积的操作（变化权重的卷积），这种计算如果单纯按照公式来写程序，运行速度较慢，这是正常的。
- 双边滤波当使用共享距离权重时，要注意对共享的距离权重变量进行深拷贝，否则调整权重时会影响到共享距离权重的值，导致计算错误。变量间的引用关系也应该在编程时格外注意。

在线问答

深蓝学院
shenlanxueyuan.com

Q&A

深蓝学院
shenlanxueyuan.com

感谢各位聆听！

Thanks for Listening