1.实验名称及目的

1.1 实验名称

点云分割实验

1.2 实验目的

通过平台取图python接口并获取分割图点云数据进行实时显示。

注意：本实验只支持Windows下Python环境运行。不支持在WSL以及虚拟机下运行。
注意：本实验需要使用DesertTown场景，目前只支持完整版，免费版无法正常运行本例程。

1.3 关键知识点

本实验主要是实现通过Python接口Open3DShow.py（见RflySimAPIs\RflySimSDK\vision目录）对获得到的图像以及点云数据进行处理，通过死循环将不断得到的点云数据输入到自定义的绘制点云图函数从而绘制点云图，使得点云图一直显示在屏幕上。关键代码解析如下：

1）视觉接口使用

```
vis = VisionCaptureApi.VisionCaptureApi() \# 创建一个视觉传感器实例

vis.jsonLoad() \# 加载Config.json中的传感器配置文件

isSuss = vis.sendReqToUE4() \# 向RflySim3D发送取图请求

vis.startImgCap() \# 开启取图

vis.hasData[i] \# 图片i数据是否更新

vis.Img[i] \# 图片i数据（像素矩阵）

cv2.imshow('Img'+str(i),vis.Img[i]) \# 显示图片i图像
```

2）相机数量和参数配置

其中，视觉传感器的初始状态由本文件夹下的Config.json决定，主要包含以下配置项：

```
"SeqID":0：使用自动更新ID的方式，创建了SeqID为0和1的两个视觉传感器

"TypeID":4和"TypeID":20：传感器类型为分割和激光雷达

"TargetCopter":1：相机绑定在1号飞机上

"SendProtocol":[0,0,0,0,0,0,0,0]：传输模式为0共享内存机制，因此本例程只能运行在Windows环境下。

"SensorPosXYZ":[ 0.3,0,0]和"SensorPosXYZ":[ 0,0,-0.3]：两个相机的位置。
```

3）Open3DShow接口使用

```
show3d=Open3DShow.Open3DShow() \# 创建一个点云显示实例

show3d.CreatShow(0) \# 创建点云显示窗口

show3d.pcdTmp.points = open3d.utility.Vector3dVector(data[:, :3]) \#将data数据的前三列赋值show3d.pcdTmp.points，以便更新点云位置

show3d.pcdTmp.colors = open3d.utility.Vector3dVector(color) \#将生成的颜色数组赋值给点云，以更新每个点的颜色

show3d.UpdatePCD() \# 更新点云的渲染
```

4）飞机控制指令

```
mav = PX4MavCtrl.PX4MavCtrler(1) \# 创建飞机控制实例

mav.InitMavLoop() \# 初始化Mavlink监听程序，读取飞机数据

mav.initOffboard() \# 进入Offboard模式

mav.SendMavArm(True) \# 解锁飞控

mav.SendPosNED(0, 0, -2, 3.1415926) \# 发送2米高的位置以及旋转角设置控制指令
```

5）UE控制

接口详细使用方法见：UE4CtrlAPI.py

```
ue = UE4CtrlAPI.UE4CtrlAPI() \# 创建UE控制实例

ue.sendUE4Cmd('r.setres 1280x720w',0) \#发送指令，设置UE4窗口分辨率，注意本窗口仅限于显示，取图分辨率在json中配置，本窗口设置越小，资源需求越少。

ue.sendUE4Cmd('t.MaxFPS 30',0) \#发送指令，设置UE4最大刷新频率30Hz，同时也是取图频率
```

6）其余代码说明

```
timeInterval = 1/30.0 \# 以30hz的频率进行控制

lastTime = lastTime + timeInterval \# 设置每一帧的处理结束时间

sleepTime = lastTime - time.time() \#计算休息时间，从而保持按照设定的频率执行代码

rows,cols = data.shape \# 获取图像数据的维度（行和列）

points_intensity = (data[:, 3] \* 255).astype(int) \#获取点云第四列数据，将其转化为0到255之间的整数

unique_elements, indices = np.unique(points_intensity, return_inverse=True) \#找到强度值的唯一元素，并返回它们在数组中的位置

color = np.zeros((len(points_intensity),3)) \# 初始化一个零数组，用于存储颜色

color[indices] = np.random.rand(\*(1,3)) \#为每个唯一的强度值生成随机颜色，并赋值给相应的点
```

2.实验效果

本实验通过平台取图python接口并获取分割图点云数据进行实时显示。

3.文件目录

例程目录：
[安装目录]\RflySimAPIs\8.RflySimVision\0.ApiExps\1-UsageAPI\3.PointCloudAPI\1.SegmentPointCloudDemo

文件夹/文件名称	说明
SegmentPointCloud.bat	一键仿真启动脚本
SegmentPointCloud.py	Python实验代码

文件夹/文件名称	说明
Config.json	视觉传感器配置文件
Python38Run.bat	Python环境启动脚本

4.运行环境

4.1 软件要求

Windows 10及以上版本；RflySim工具链；VS Code。

> ①：若使用Pixhawk 6X飞控，平台安装时的编译命令为：px4_fmu-v6x_default，推荐PX4固件版本为：1.12.3。其他配套飞控及编译命令请见：https://rflysim.com/doc/zh/1/Hardware.html

4.2 硬件要求

笔记本/台式电脑① 1台。

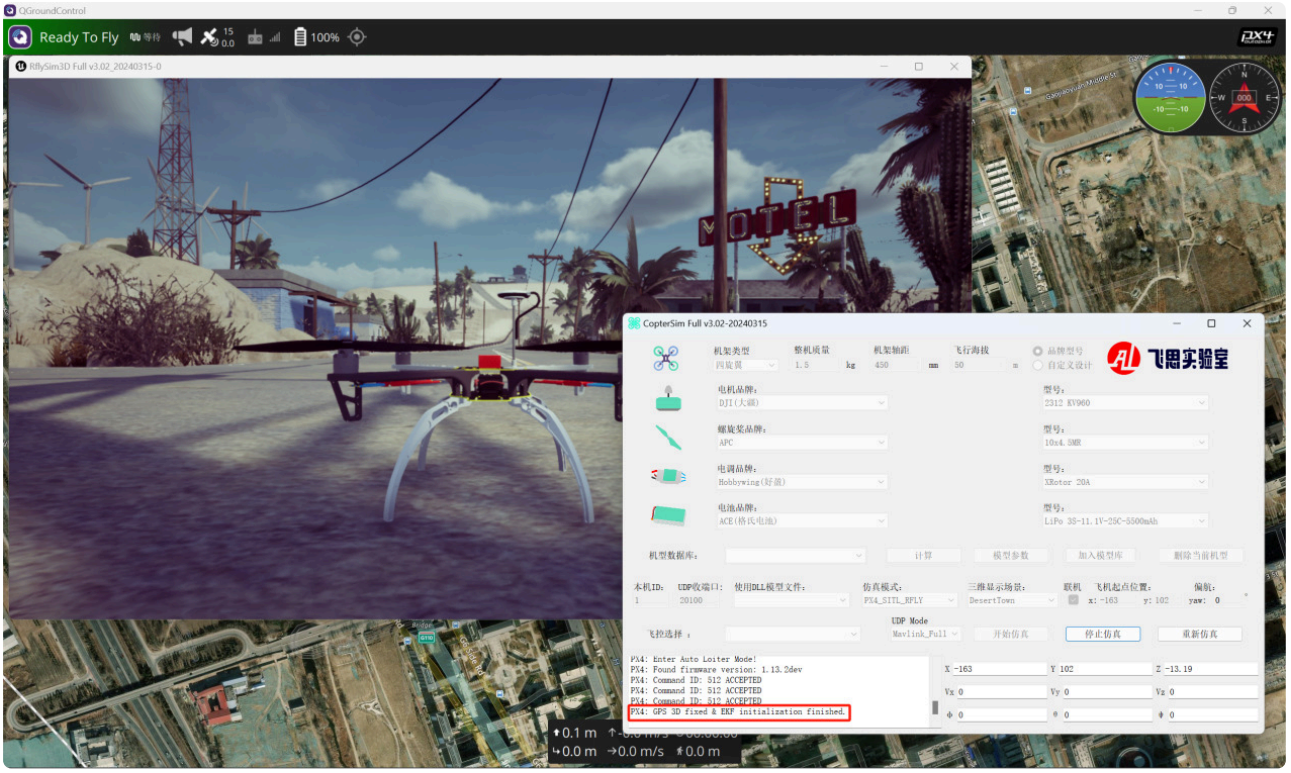
> ①：推荐配置请见：https://rflysim.com/doc/zh/HowToInstall.pdf

5.实验步骤

必做实验：Windows取图控制

Step 1：开启仿真

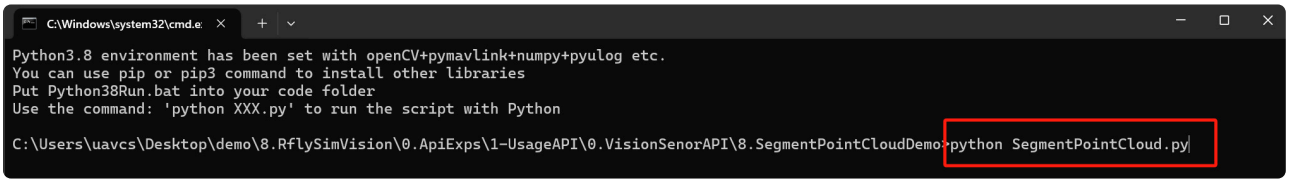
双击运行SegmentPointCloud.bat开启一个软件在环仿真。将会启动1个QGC地面站，1个CopterSim软件且其软件下侧日志栏必须打印出GPS 3D fixed & EKF initialization finished字样代表初始化完成，并且RflySim3D软件内有1架无人机。



Step 2：运行控制程序

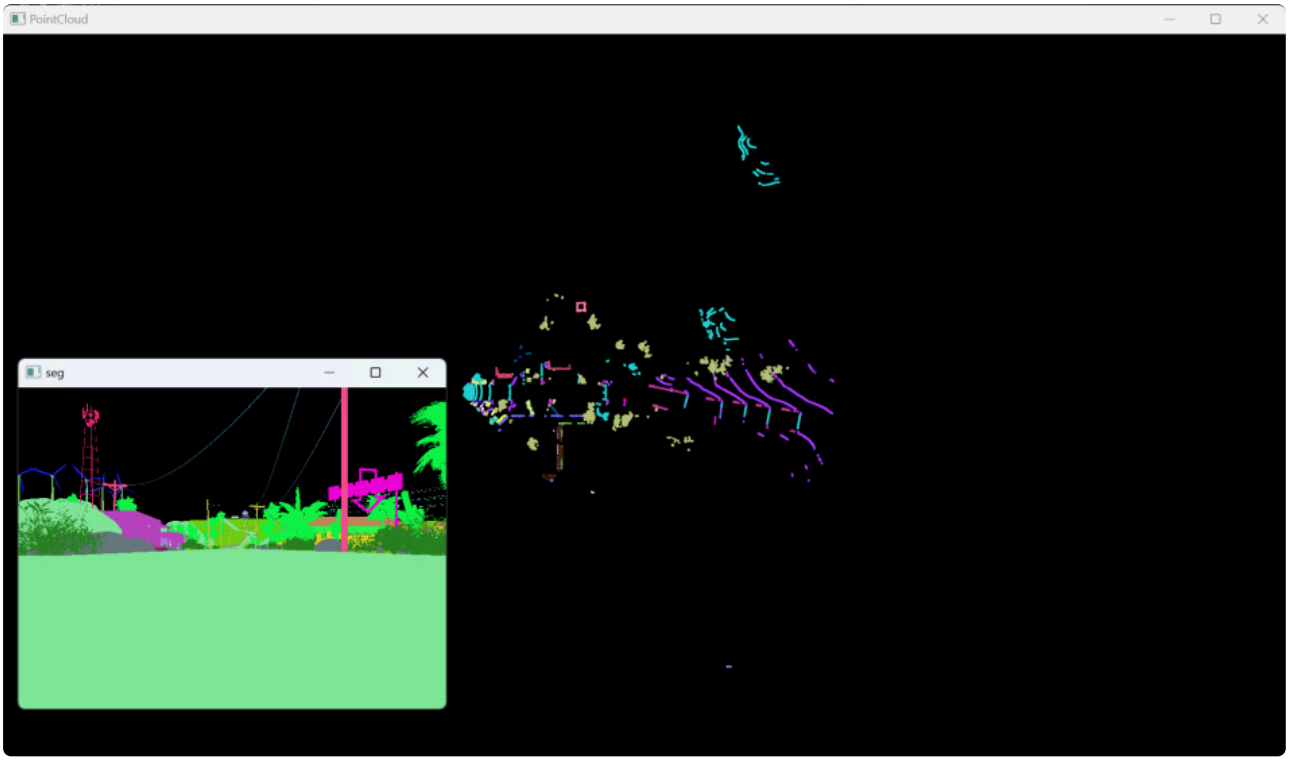
在文件夹下，双击 Python38Run.bat，打开集成好的python环境，在该环境下运行 SegmentPointCloud.py 文件，输入

```
python SegmentPointCloud.py
```



Step3：观察结果

可以看到飞机成功起飞，一个分割图像窗口和一个点云图像窗口。



Step 4：结束仿真

在下图“SegmentPointCloud.bat”脚本开启的命令提示符CMD窗口中，按下回车键（任意键）就能快速关闭CopterSim、QGC、RflySim3D等所有程序。

选作实验（VS Code调试运行）

准备工作

- 先确保已经按RflySimAPIs\1.RflySimIntro\2.AdvExps\e3.PythonConfig\Readme.pdf步骤，正确配置VS Code环境。或者配置了自己的Pycharm等自定义Python环境。
- 其他步骤与上文相同，在Step2运行SegmentPointCloud.py时，可使用VS Code（或Pycharm等工具）来打开SegmentPointCloud.py文件，并阅读代码，修改代码，调试执行等。

扩展实验

- 请自行使用VS Code阅读SegmentPointCloud.py源码，通过程序跳转，了解每条代码的执行原理；再通过调试工具，验证每条指令的执行效果。

VisionCapAPIDemo.py

> RflySimAPIs > 8.RflySimVision > 0.ApiExps > 1-UsageAPI > 0.VisionSenorAPI > 1.Came

```
ue = UE4CtrlAPI.UE4CtrlAPI()

#Create a new MAVLink communication instance, UDP sending
mav = PX4MavCtrl.PX4MavCtrler(1)

# The IP should be specified by the other computer
vis = VisionCaptureApi.VisionCaptureApi()

# Send command to UE4 Window 1 to change resolution
ue.sendUE4Cmd('r.setres 1280x720w',0) # 设置UE4窗口分辨率，
ue.sendUE4Cmd('t.MaxFPS 30',0) # 设置UE4最大刷新频率，同时也
time.sleep(2)

# VisionCaptureApi 中的配置函数
vis.jsonLoad() # 加载Config.json中的传感器配置文件
```

- 请尝试修改代码，实现飞机位置改变、相机姿态角改变、相机参数改变等功能。

6.参考资料

无

7.常见问题

Q1：无

A1：无