1.实验名称及目的

1.1 实验名称

UDP直传点云数据入门实验

1.2 实验目的

使用UDP直传方式进行点云数据传输实验

1.3 关键知识点

本实验主要是实现通过Python接口VisionCaptureApi.py使用（见RflySimAPIs\RflySimSDK\vision目录）发送取图请求。通过PX4MavCtrler.py（见RflySimAPIs\RflySimSDK\ctrl目录）使用创建控制接口，以此向RflySim发送控制指令，并使用UDP获取点云数据，将数据传输到定义的绘制点云数据的值，从而在虚拟机中绘制出点云图。关键代码解析如下：

本例子和其他分布式例子的区别，主要在于直接通过UDP直传传出点云数据。

关键知识点1：SendProtocol[0]决定了图像的传出模式。SendProtocol[0]=0：共享内存（仅限Windows下获取图像），1：UDP直传png压缩，2：UDP直传图片不压缩（只适用图片类传感器），3：UDP直传jpg压缩（只适用图片类传感器）。如果是激光雷达数据只有0或1（共享内存和UDP网络传输）。

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(0, TargetIP) \#向RflySim3D发送取图请求，发给ip为TargetIP的地址

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）
```

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的视觉传感器

"TypeID":20：传感器类型为激光雷达

"TargetCopter":1：相机绑定在1号飞机上

"SendProtocol":[1,0,0,0,0,0,0,0]：传输模式为1：UDP网络传输模式（图片使用jpeg压缩，点云直传）。

"SensorPosXYZ":[0,0,-0.3]：相机分布位置。
```

3）Open3DShow接口使用

```
show3d=Open3DShow.Open3DShow() \# 创建3D点云显示实例

show3d.CreatShow(0) \# 创建点云显示窗口

show3d.UpdateShow(vis.Img[0]) \# 更新点云
```

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.InitTrueDataLoop() \# 初始化来自 CopterSim 的 UDP True 数据监听循环

mav.initOffboard() \# 进入Offboard模式

mav.SendMavArm(True) \# 解锁飞控

mav.SendPosNED(0, 0, -8, 0) \# 发送8米高的位置控制指令

mav.SendCopterSpeed(3) \# 发送速度为3m/s控制指令

mav.endOffboard() \# 结束offboard模式

mav.stopRun() \# 结束mav飞机实例运行
```

5）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

targetPosE=targetPosE+Error2UE4Map[j] \# 设置飞机位置
```

2.实验效果

本实验通过平台接口进行RflySim3D直接10hz频率UDP直传点云数据。

3.文件目录

例程目录：[安装目录]\RflySimAPIs\8.RflySimVision\0.ApiExps\8.LidarAPIDemo\1.SharedMemoryClientServer

文件夹/文件名称	说明
client_ue4_SITL.bat	Windows客户端启动仿真配置文件
server_ue4.py	取图控制程序（支持Windows或Linux分布式运行）
Config.json	视觉传感器配置文件

文件夹/文件名称	说明
lidar.rviz	Rviz配置文件
Python38Run.bat	Windows下Python程序运行脚本
WinWSL.bat	WSL1/Ubuntu 20.04环境程序运行脚本
WslGUI.bat	WSL1/Ubuntu 20.04可视化界面脚本

4.运行环境

4.1 软件要求

Windows 10及以上版本；RflySim工具链；Visual Studio Code；Linux（Ubuntu 20.04）；Linux（Ubuntu 20.04）。

> 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；WinWSL 1台；虚拟机/视觉盒子/其他板卡 可选台。

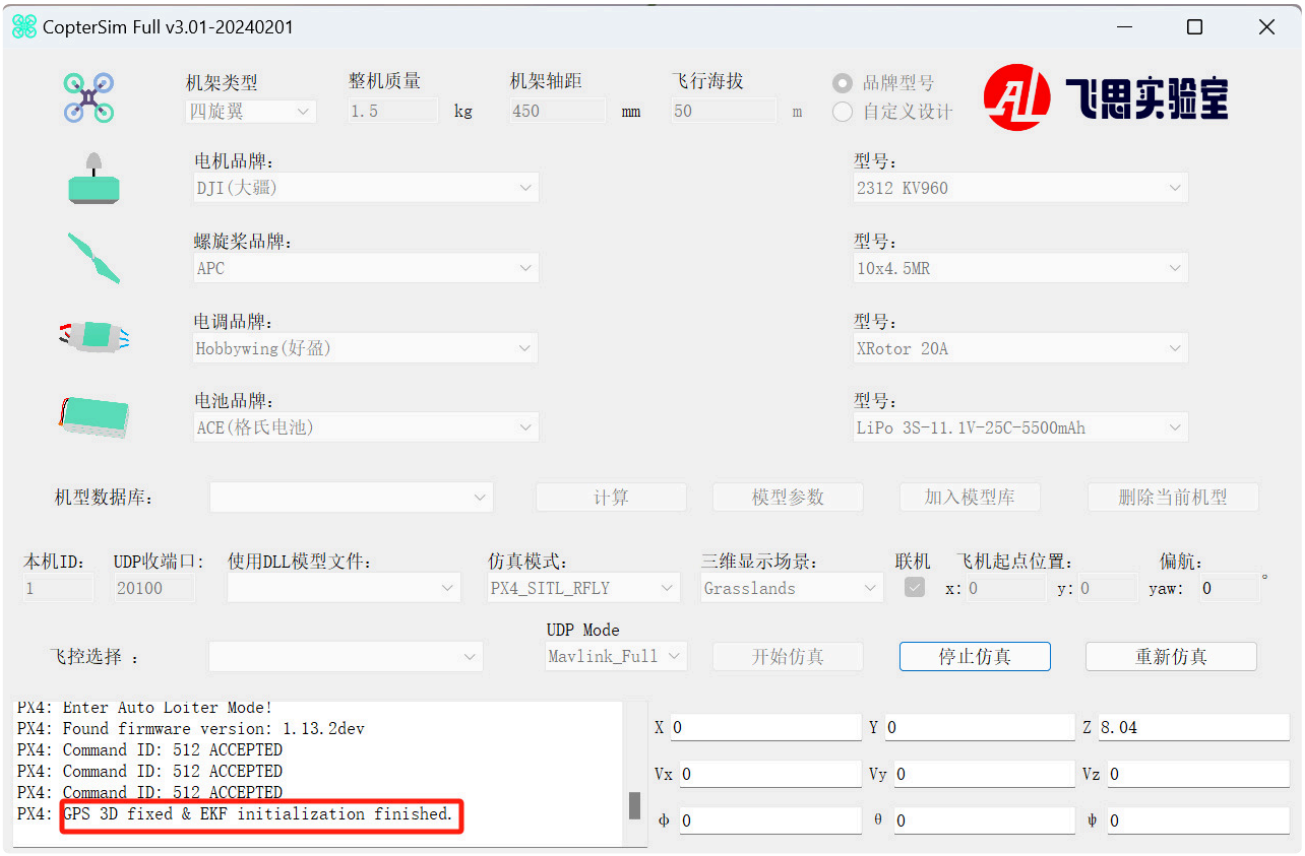
> ①：推荐配置请见： https://rflysim.com/doc/zh/HowToInstall.pdf

5.实验步骤

5.1. 必做实验：WinsWSL控制

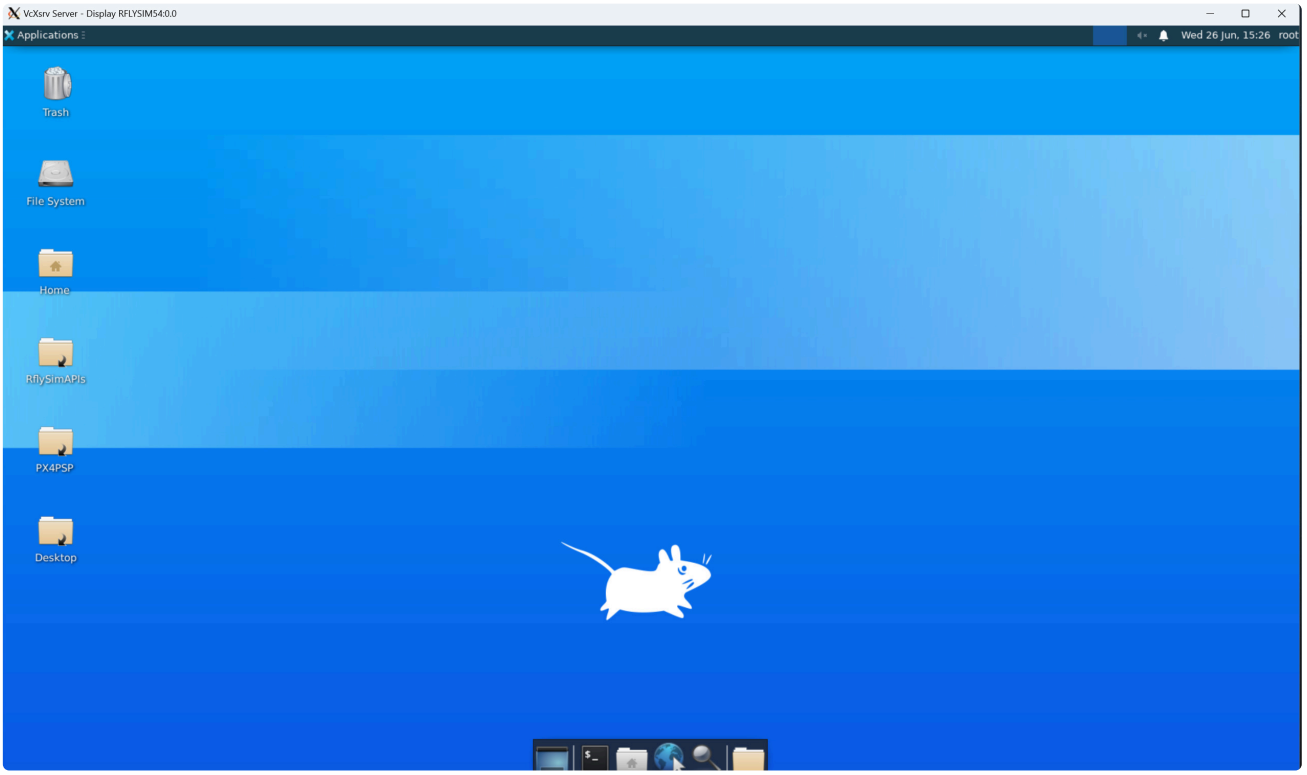
Step 1：开启仿真

双击运行LidarAPIDemo.bat开启一个飞机的软件在环仿真。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：开启WSL可视化界面

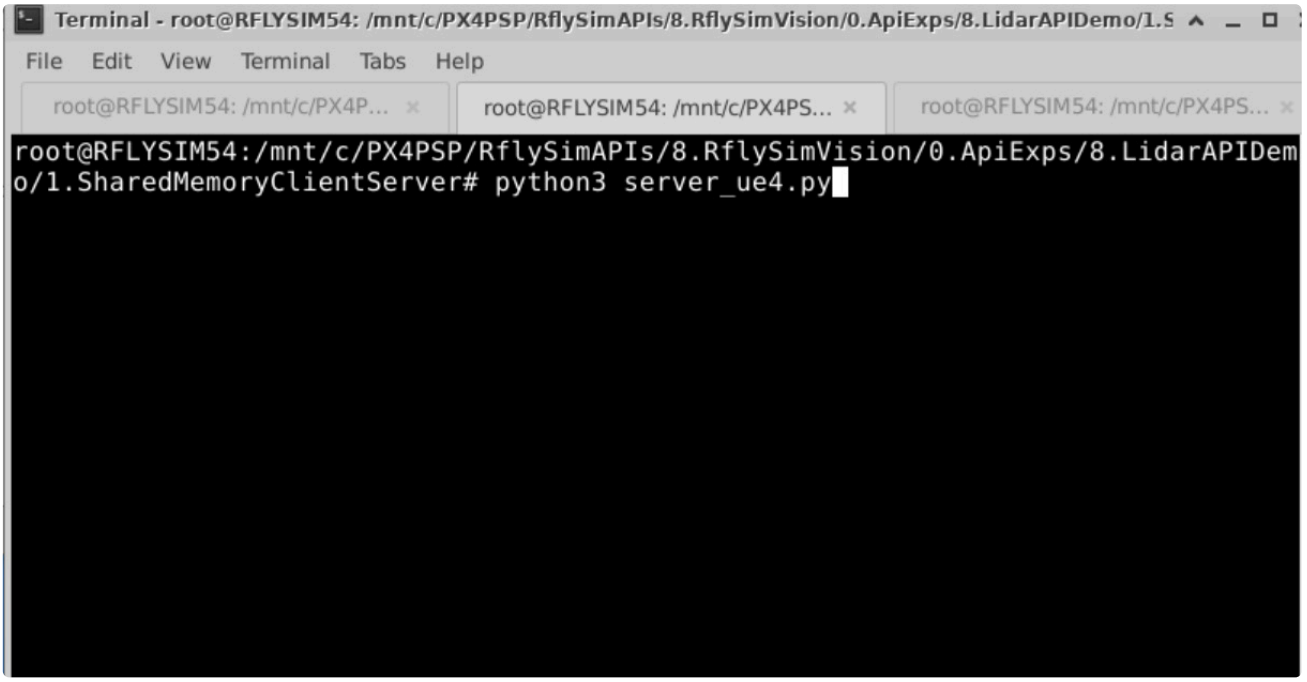
双击打开WslGUI.bat，启动WSL可视化界面。（注：如果打开发现窗口白屏，没有桌面，则关了重开一两次。）

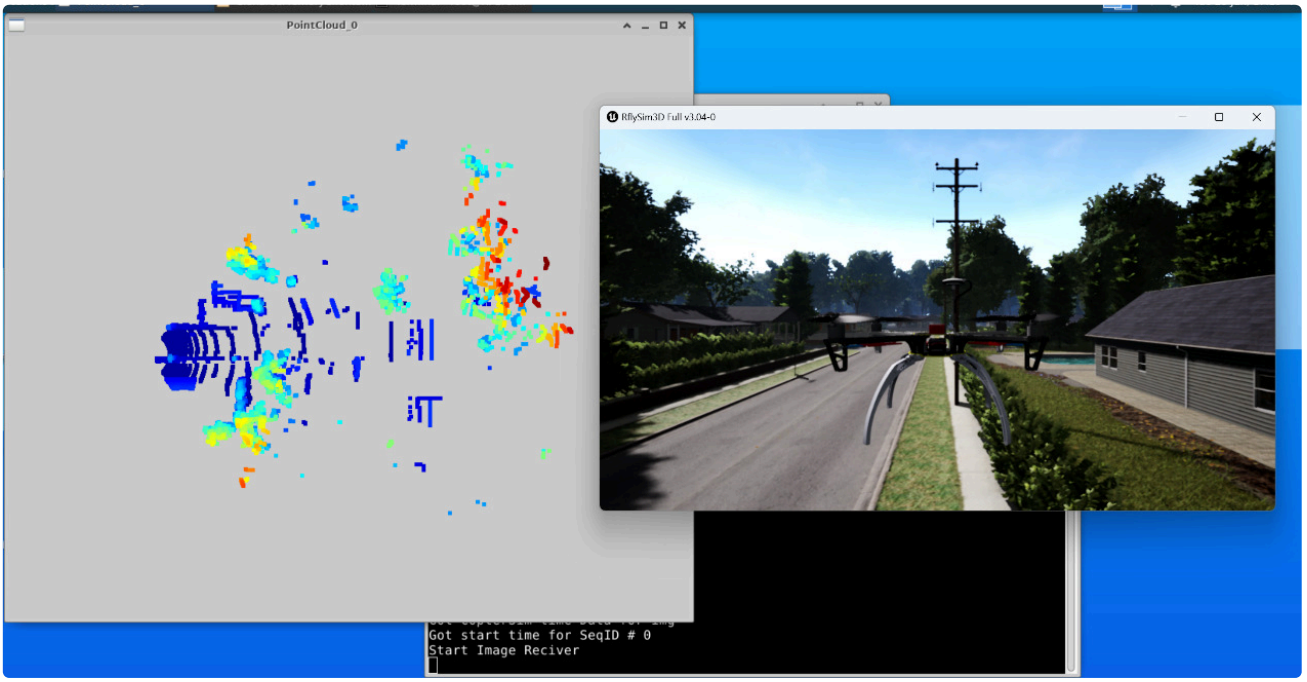


注意：参考`[安装目录]\\RflySimAPIs\\1.RflySimIntro\\2.AdvExps\\e7_WslGUI\\Intro.pdf`，来了解WslGUI的功能与使用。

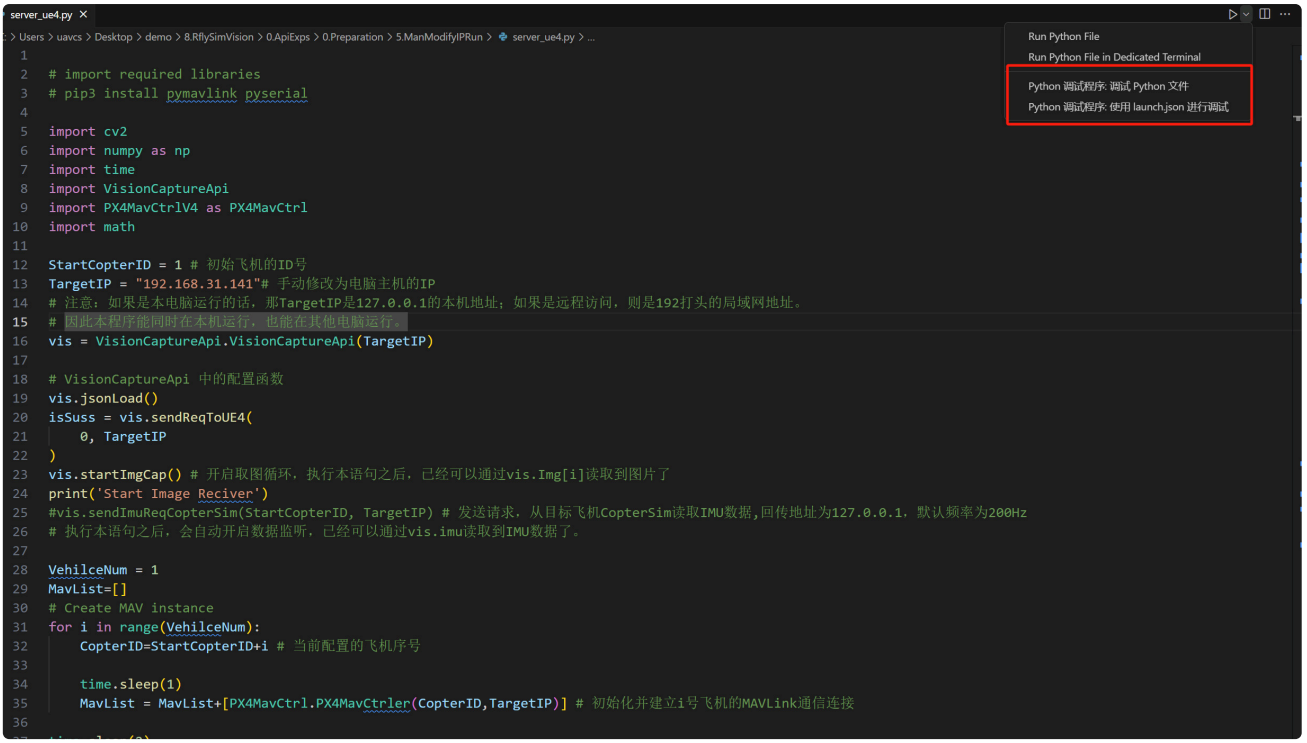
Step 3：运行控制程序

双击打开WinWSL.bat，运行`roscore`指令，双击打开WinWSL.bat，运行`pip install matplotlib –upgrade`，更新matplotlib库，再运行命令`python3 server_ue4.py`运行脚本server_ue4.py。可见绘制的点云图不断更新：





备注：可以参考`[安装目录]\\RflySimAPIs\\1.RflySimIntro\\2.AdvExps\\e8.WslVsCode\\Intro.pdf`来使用VS Code开发并调试Ubuntu下python文件。



Step 4：结束仿真

在下图“LidarAPIDemo.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。

5.2. 选作实验

准备工作：

虚拟机或NX的配置方法是相同的。

1）Ubuntu虚拟机环境下，进行分布式联机实验。先参考`[安装目录]\\RflySimAPIs\\8.RflySimVision\\0.ApiExps\\0.Preparation\\1.VMwareUbuntu\\Readme.pdf`，完成虚拟机的下载与配置。

2）用第二台Ubuntu电脑或NX板卡，实现联机实验。其他Ubuntu电脑的配置，先看`[安装目录]\\RflySimAPIs\\8.RflySimVision\\0.ApiExps\\0.Preparation\\2.GenenralUbuntuConfig\\Readme.pdf`；NX板卡的配置方法，先看[安装目录]\RflySimAPIs\8.RflySimVision\0.ApiExps\0.Preparation\3.NXwithPX4Config\Readme.pdf。

扩展实验：

5.2.1 本机Windows接收图像实验

Step 1：开启仿真

步骤1同6、实验步骤的Step1步骤。

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server_ue4.py文件，输入`python server_ue4.py`

注：也可以使用VScode打开运行（如果安装好了VScode），VS Code打开server.py并执行

5.2.2 远端Windows电脑接收图像实验

Step 1：开启仿真

步骤1同6、实验步骤的Step1步骤。

Step 2：运行控制程序

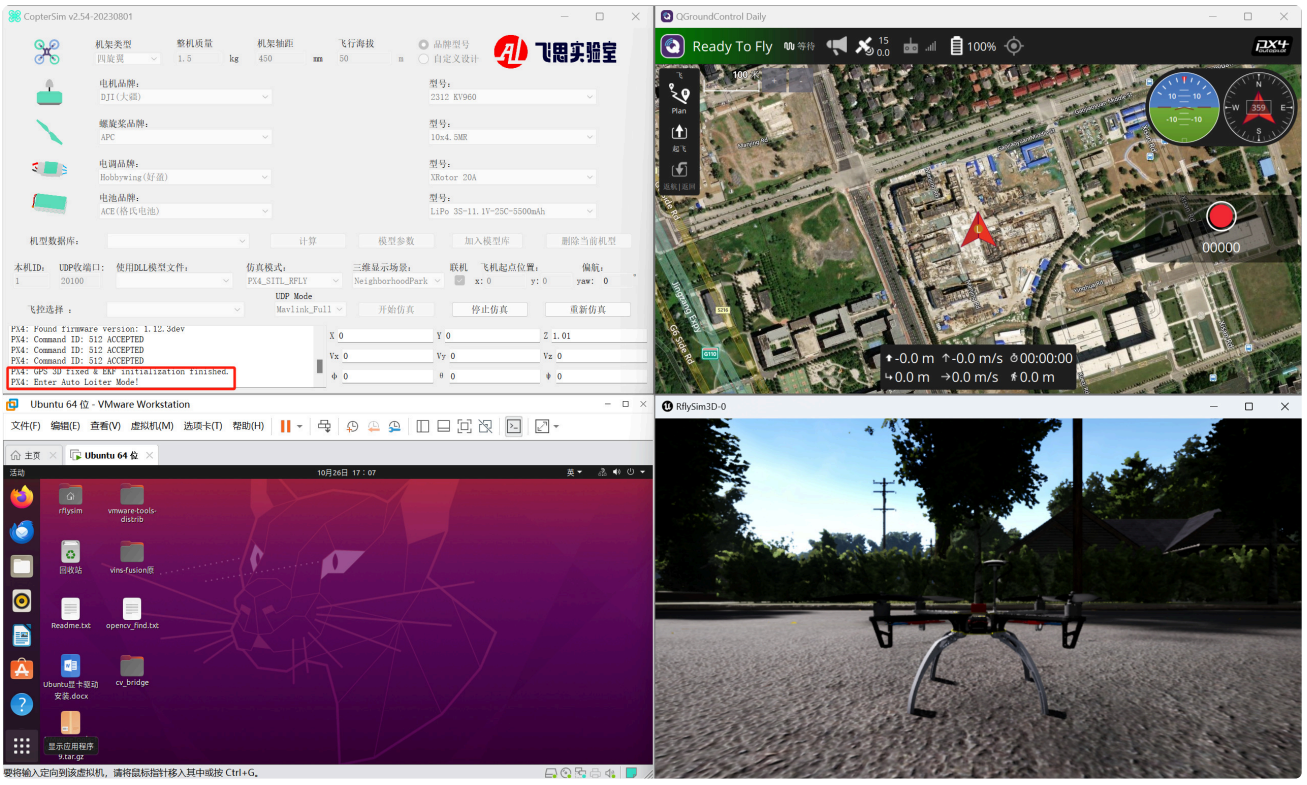
在另一台Windows电脑上双击Python38Run.bat， 再运行server.py。

5.2.3在虚拟机/视觉板卡/另一台Ubuntu上接收图像实验

Step 1：开启仿真

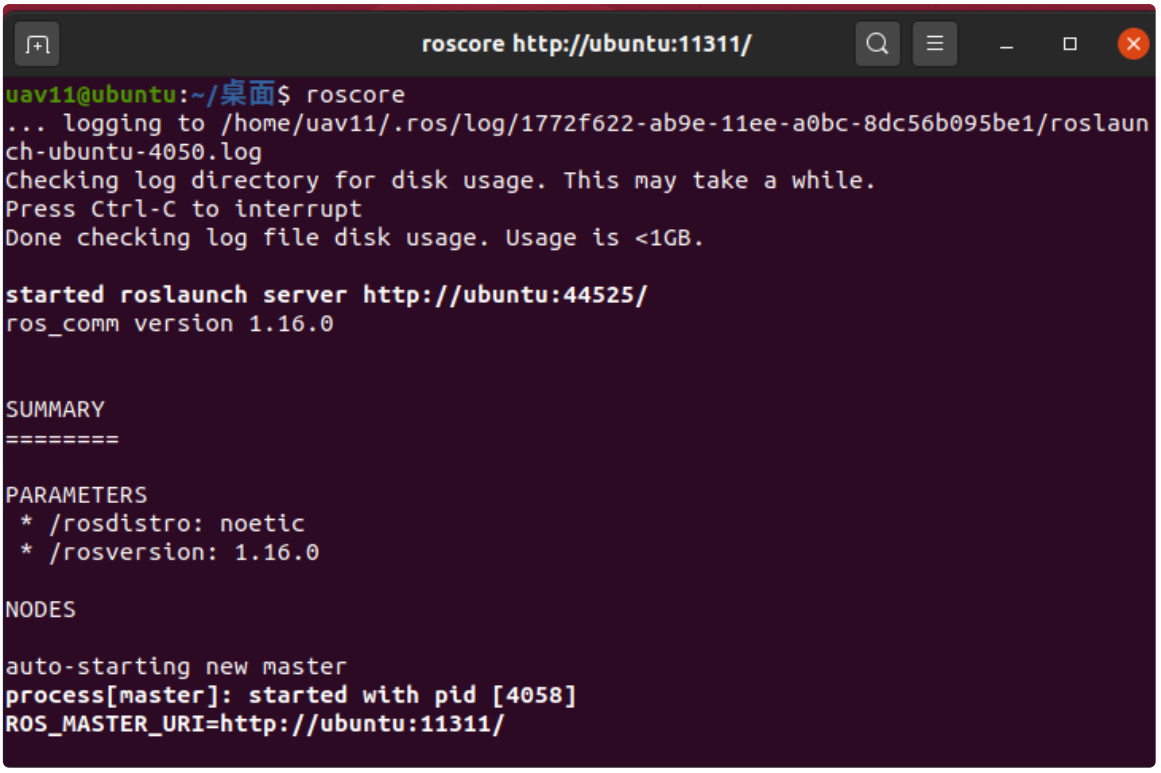
步骤1同上面的Step1步骤。

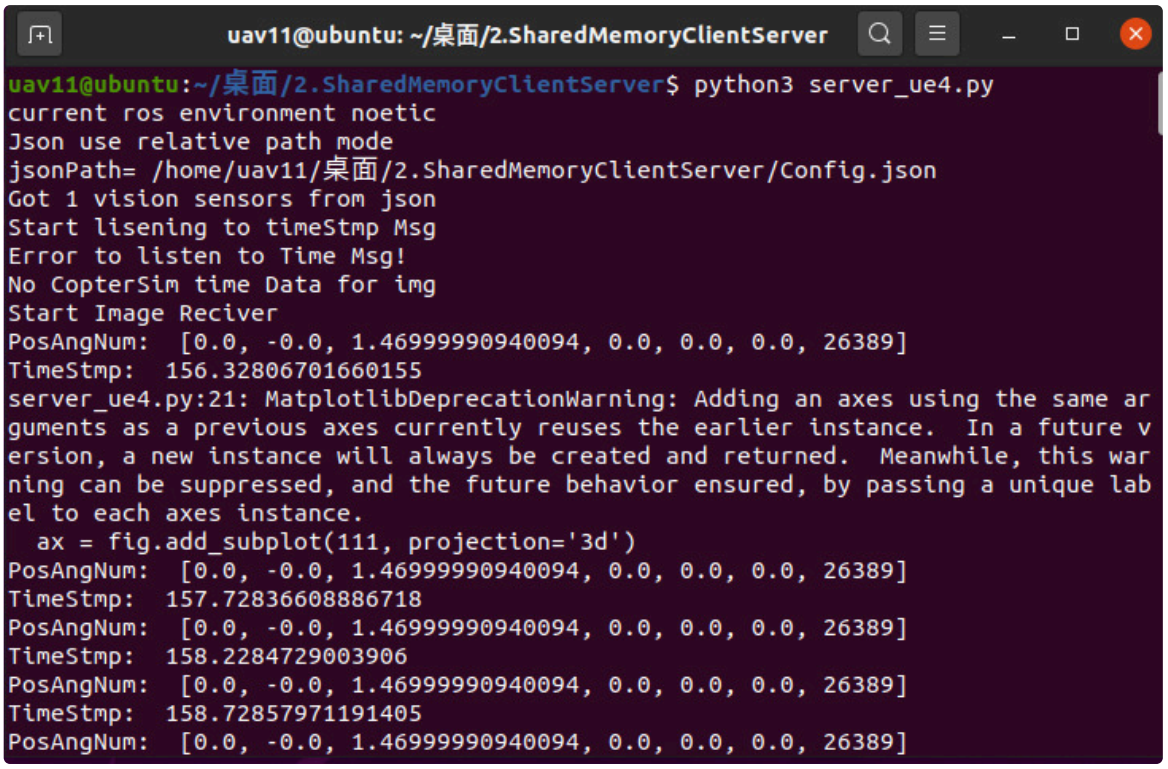
并且启动一个Ubuntu虚拟机。

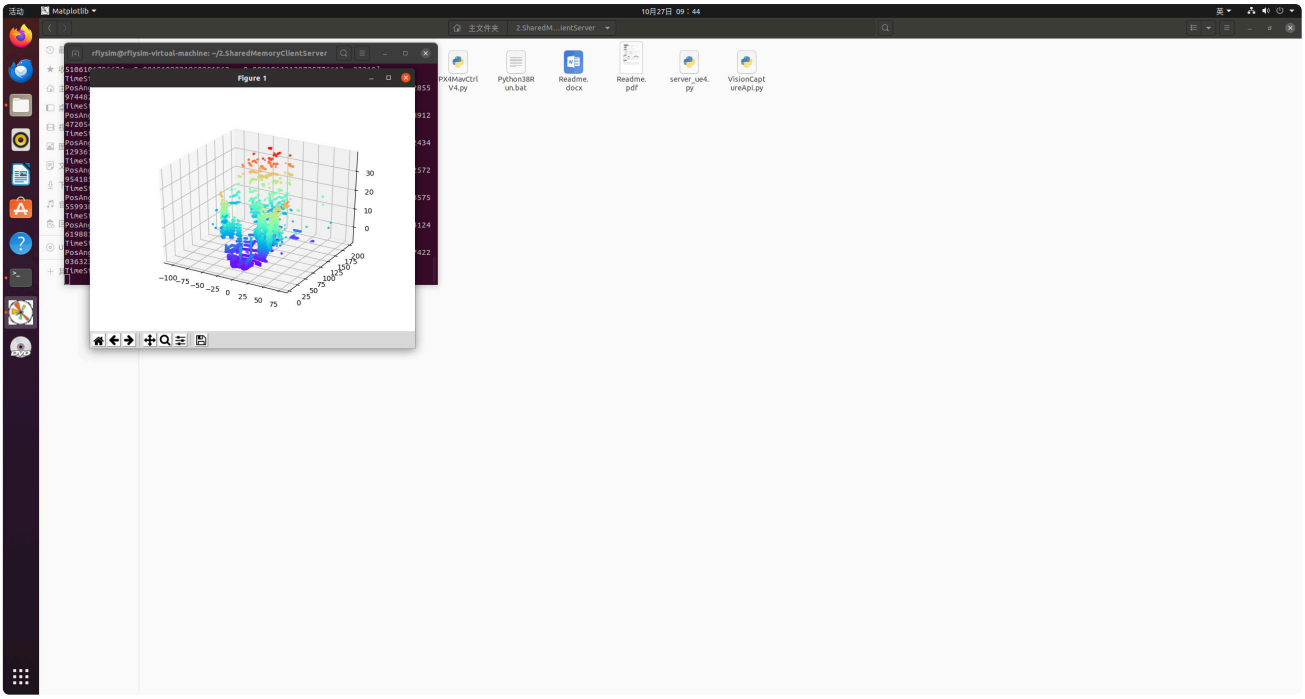


Step 2：运行控制程序

在虚拟机中新建一个文件夹，并将该实验文件全拷贝过去，然后在终端中运行roscore指令，将起新终端路径选择到新建的文件夹路径，运行`pip install matplotlib –upgrade`，更新matplotlib库，再运行命令`python3 server_ue4.py`运行脚本 server_ue4.py。可见绘制的点云图不断更新：


roscore http://ubuntu:11311/
uav11@ubuntu:~/桌面$ roscore
... logging to /home/uav11/.ros/log/1772f622-ab9e-11ee-a0bc-8dc56b095be1/roslaunch-ubuntu-4050.log
Checking log directory for disk usage. This may take a while.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.
started roslaunch server http://ubuntu:44525/
ros_comm version 1.16.0
SUMMARY
========
PARAMETERS
* /rosdistro: noetic
* /rosversion: 1.16.0
NODES
auto-starting new master
process[master]: started with pid [4058]
ROS_MASTER_URI=http://ubuntu:11311/


uav11@ubuntu: ~/桌面/2.SharedMemoryClientServer
uav11@ubuntu:~/桌面/2.SharedMemoryClientServer$ python3 server_ue4.py
current ros environment noetic
Json use relative path mode
jsonPath= /home/uav11/桌面/2.SharedMemoryClientServer/Config.json
Got 1 vision sensors from json
Start lisening to timeStmp Msg
Error to listen to Time Msg!
No CopterSim time Data for img
Start Image Reciver
PosAngNum: [0.0, -0.0, 1.46999990940094, 0.0, 0.0, 0.0, 26389]
TimeStmp: 156.32806701660155
server_ue4.py:21: MatplotlibDeprecationWarning: Adding an axes using the same arguments as a previous axes currently reuses the earlier instance. In a future version, a new instance will always be created and returned. Meanwhile, this warning can be suppressed, and the future behavior ensured, by passing a unique label to each axes instance.
ax = fig.add_subplot(111, projection='3d')
PosAngNum: [0.0, -0.0, 1.46999990940094, 0.0, 0.0, 0.0, 26389]
TimeStmp: 157.72836608886718
PosAngNum: [0.0, -0.0, 1.46999990940094, 0.0, 0.0, 0.0, 26389]
TimeStmp: 158.2284729003906
PosAngNum: [0.0, -0.0, 1.46999990940094, 0.0, 0.0, 0.0, 26389]
TimeStmp: 158.72857971191405
PosAngNum: [0.0, -0.0, 1.46999990940094, 0.0, 0.0, 0.0, 26389]



6.参考资料

无

7.常见问题

Q1：运行server_ue4.py时报错ValueError: Unknown projection '3d'

A1：这是因为matplotlib库的版本的原因,可通过添加如下语句解决

from mpl_toolkits.mplot3d import Axes3D