

1. 实验名称及目的

1.1 实验名称

共享内存传输点云数据实验

1.2 实验目的

在虚拟机中运行程序发送取图请求，经过平台共享内存方式发出，`server_ue4.py`接收点云数据。

1.3 关键知识点

本实验主要是实现通过Python接口ReqCopterSim.py使用（见RflySimAPIs\RflySimSDK\ctrl目录）自动获取的IP，去建立远端电脑与本机RflySim3D（发送图片）与CopterSim（接收控制指令）的联机仿真。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，利用Open3DShow.py接口，进行作图。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Config.json中，SendProtocol[0]设置为了1，即UDP压缩传图模式。

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）。

关键知识点2：通过ReqCopterSim可以自动从局域网获取到仿真电脑的IP地址，从而自动建立连接，不再需要手动指定IP地址。不过，此种连接方式，可能在局域网中产生干扰（多台电脑同时打开多个CopterSim会产生误识别），不适合多个实验同时进行的场景。

1) 视觉接口使用

```
1 vis = VisionCaptureApi.VisionCaptureApi() \# 创建一个视觉传感器实例
2
3 vis.jsonLoad() \# 加载Config.json中的传感器配置文件
4
5 isSuss = vis.sendReqToUE4(0, TargetIP) \#向RflySim3D发送取图请求, 发给ip为TargetIP的地址
6
7 vis.startImgCap() \# 开启取图
8
9 vis.hasData[i] \# 图片i数据是否更新
10
11 vis.Img[i] \# 图片i数据 (像素矩阵)
12
13 cv2.imshow('Img'+str(i),vis.Img[i]) \# 显示图片i图像
```

2) ReqCopterSim接口使用 (自动获取ip接口)

```
1 req = ReqCopterSim.ReqCopterSim() \# 获取局域网内所有CopterSim程序的电脑IP列表
2
3 TargetIP = req.getSimIpID(StartCopterID) \#自动获取CopterSim的StartCopterID号程序所在电脑
4
5 req.sendReSimIP(CopterID) \# 请求mavlink数据到本电脑
```

3) Open3Dshow接口使用

```
1 show3d=Open3DShow.Open3DShow() \# 创建Open3DShow实例, 用于展示点云图像
2
3 show3d.CreatShow(0) \# 创建点云显示窗口
4
5 show3d.UpdateShow(vis.Img[0]) \# 更新点云
```

4) 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:

```
1 "SeqID":0 :使用自动更新ID的方式, 创建了SeqID为0的视觉传感器
2
3 "TypeID":20 :传感器类型为激光雷达
4
5 "TargetCopter":1 :相机绑定在1号飞机上
6
7 "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)。
8
9 "SensorPosXYZ":[0,0,-0.3] :相机分布位置。
```

5) 飞机控制指令

```
1 MavList = MavList+[PX4MavCtrl.PX4MavCtrl(CopterID,TargetIP)] \#
2 初始化并建立i号飞机的MAVLink通信连接, 连接上远端的电脑飞机, 创建一个飞机控制实例列表mavlist
3
4 MavList[i].InitMavLoop() \# 初始化Mavlink监听程序, 读取第i个飞机数据
5
6 MavList[i].initOffboard() \# 第i个飞机进入Offboard模式
7
8 Error2UE4Map =
9 Error2UE4Map+[-np.array([mav.uavGlobalPos[0]-mav.uavPosNED[0],mav.uavGlobalPos[1]-mav
10 \# 用于获取车辆的全局位置, 同时一个 Python 列表转换为一个 NumPy
11 数组, 便于进行计算处理
12
13 MavList[i].SendPosNED(0, 0, -8, 0) \# 对第i个飞机发送8米高的位置控制指令
14
15 mav=MavList[j] \# 从列表中取第i个实例作为当下处理的飞机实例
16
17 MavList[i].endOffboard() \# 对第i个飞机进入endoffboard模式
18
19 MavList[i].stopRun() \# 终止第i个飞机
```

6) UE控制

接口详细使用方法见: [UE4CtrlAPI.py](#)

```
1 ue = UE4CtrlAPI.UE4CtrlAPI() \# 创建UE控制实例
2
3 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
4
5 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
7 ue.sendUE4Cmd('RflyChangeMapbyName NeighborhoodPark', 0) \#更新地图场景, 类型为Neighborhood
8
9 ue.sendUE4PosScale(1, 3, 0, PosE=[2, 0, -3], AngEuler=[0, 0, -1.57]) \#创建一个障碍物, j
```

7) 其余代码说明

```
1 timeInterval = 1/30.0 \# 以30hz的频率进行控制
2
3 lastTime = lastTime + timeInterval \# 设置每一帧的处理结束时间
4
5 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
6
7 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\4.Point-CloudVisualize](#)

文件夹/文件名称	说明
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
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://rflsim.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。

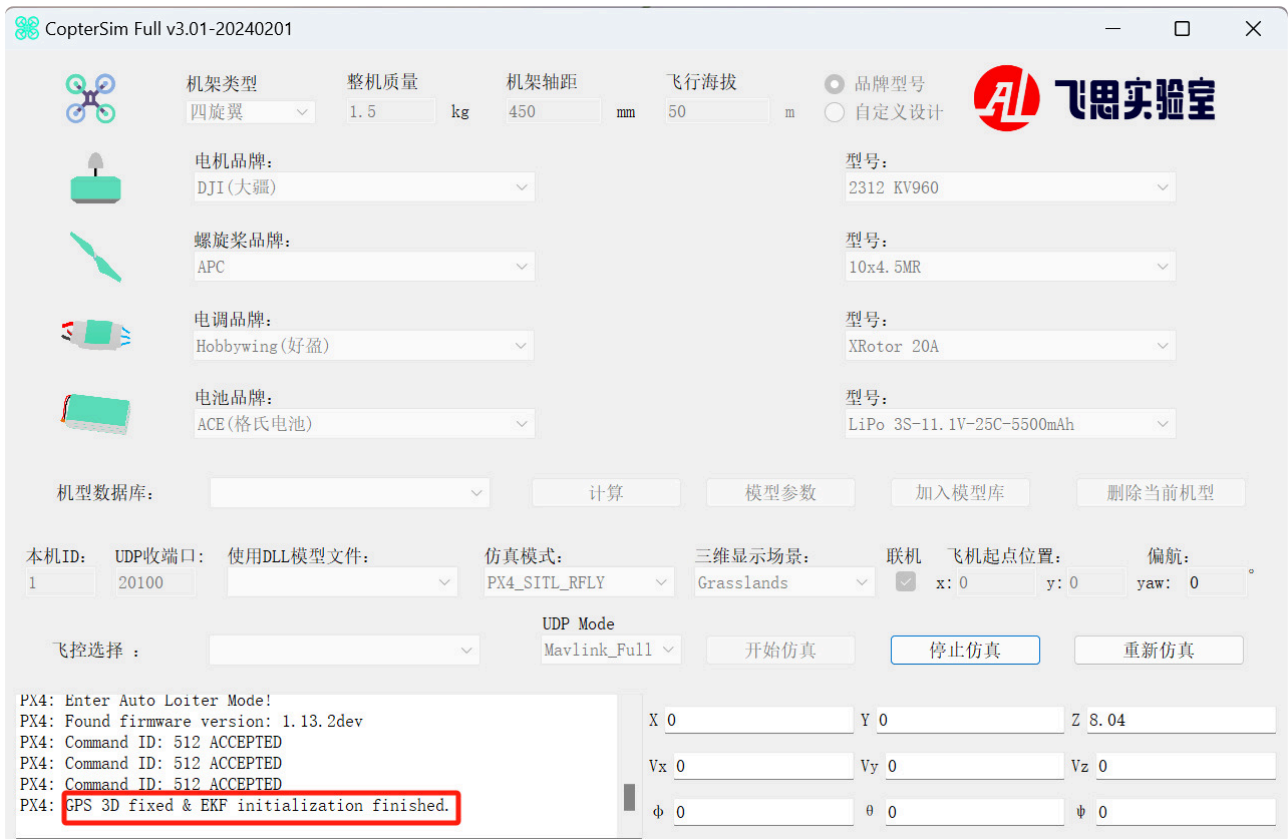
①：推荐配置请见：<https://rflsim.com/doc/zh/HowToInstall.pdf>

5. 实验步骤

5.1 必做实验：WinsWSL控制

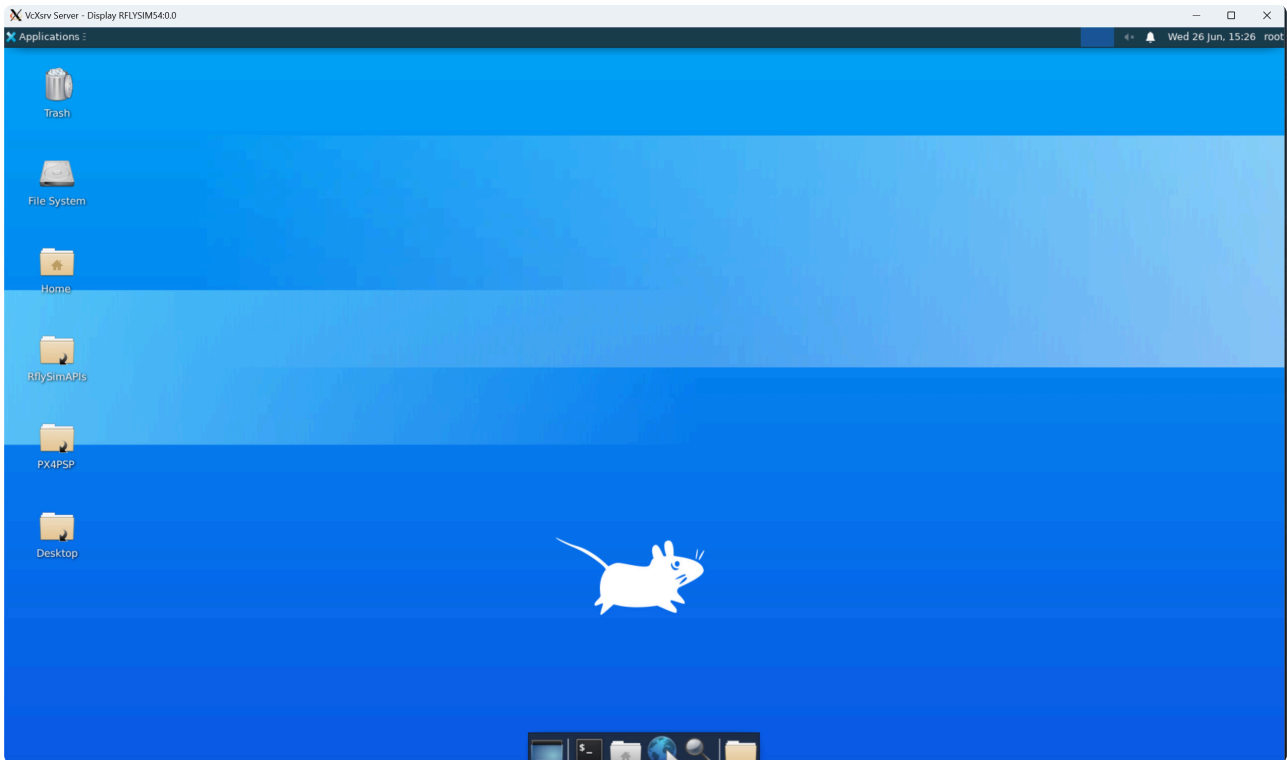
Step 1：开启仿真

双击运行 `client_ue4_SITL.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。）

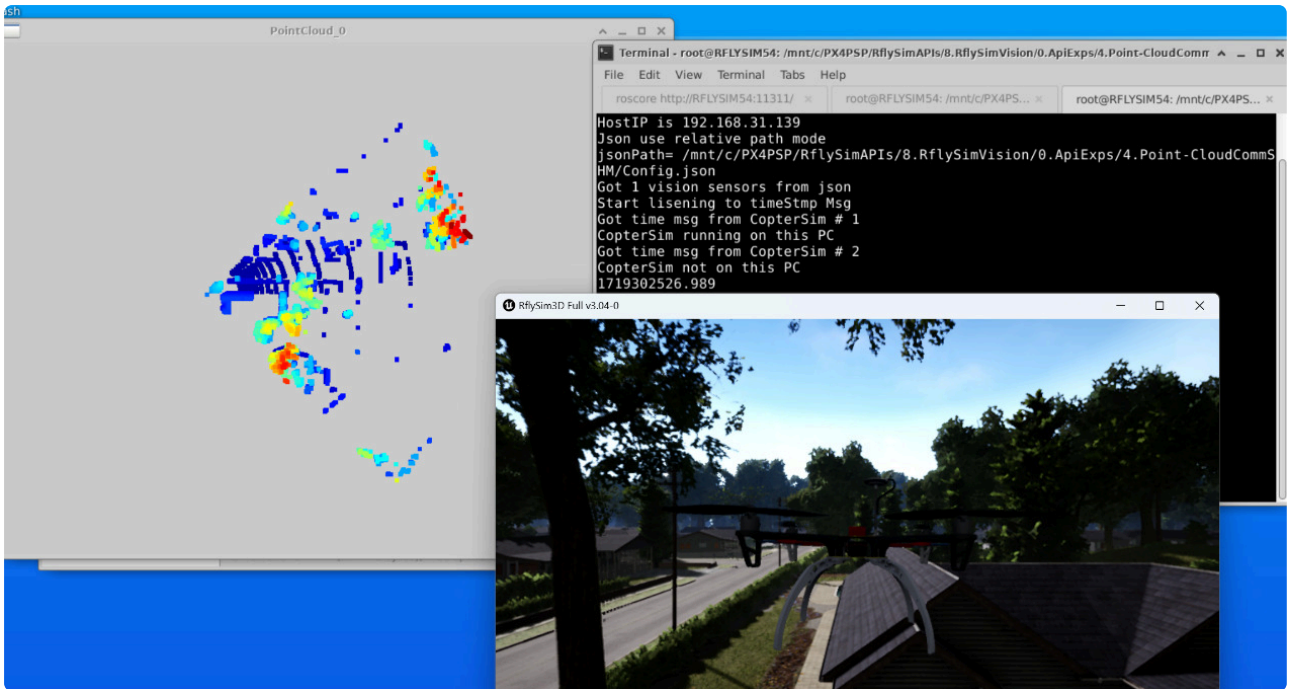


注意：参考 [安装目录]\RflySimAPIs\1.RflySimIntro\2.AdvExps\e7_WslGUI\Intro.pdf，来了解WslGUI的功能与使用。

Step 3: 运行控制程序

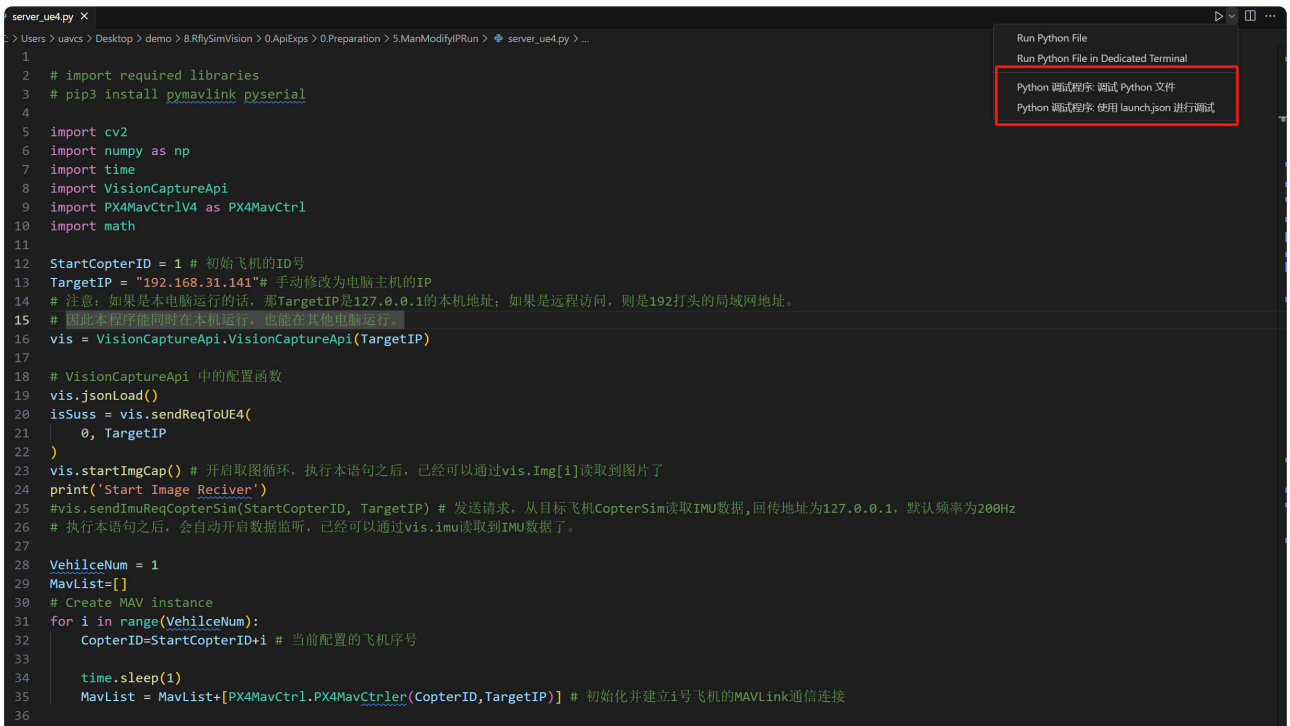
双击打开 [WinWSL.bat](#)，运行命令 `python3 server_ue4.py`，在WslGUI界面中可见绘制的点云图不断更新。

```
Terminal - root@RFLYSIM54: /mnt/c/PX4PSP/RflySimAPIs/8.RflySimVision/0.ApiExps/4.Point-CloudCommr ^ _ □ ×
File Edit View Terminal Tabs Help
roscore http://RFLYSIM54:11311/ × root@RFLYSIM54: /mnt/c/PX4PS... × root@RFLYSIM54: /mnt/c/PX4PS... ×
root@RFLYSIM54:/mnt/c/PX4PSP/RflySimAPIs/8.RflySimVision/0.ApiExps/4.Point-CloudCommSHM# python3 server_ue4.py
current ros environment noetic
HostIP is 192.168.31.139
Json use relative path mode
jsonPath= /mnt/c/PX4PSP/RflySimAPIs/8.RflySimVision/0.ApiExps/4.Point-CloudCommSHM/Config.json
Got 1 vision sensors from json
Start lisening to timeStmp Msg
Got time msg from CopterSim # 1
CopterSim running on this PC
Got time msg from CopterSim # 2
CopterSim not on this PC
1719302526.989
1719302529.3978224
Got time msg from CopterSim # 3
CopterSim not on this PC
```



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。



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.GeneralUbuntuConfig\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：

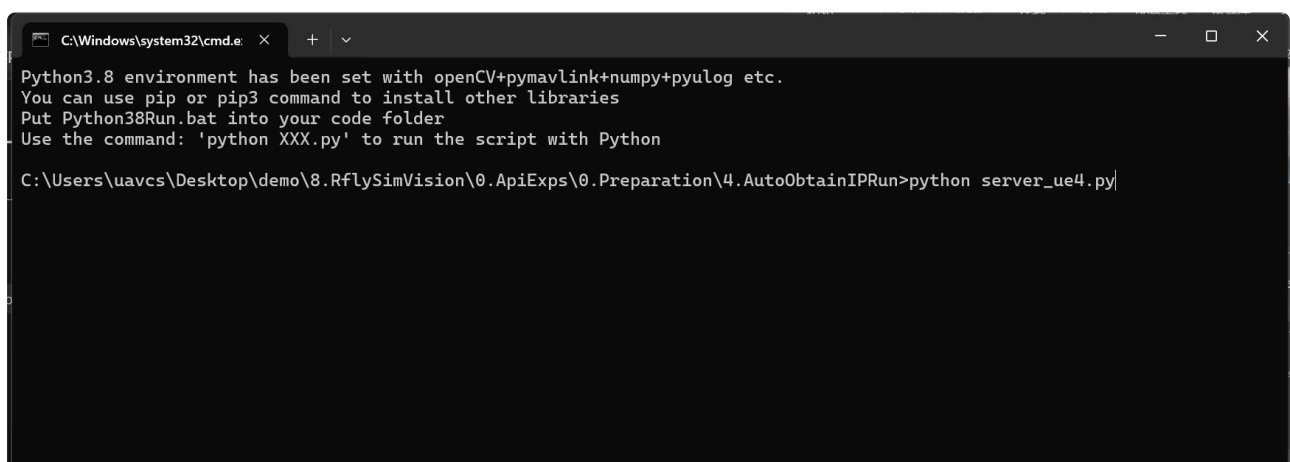
5.2.1 本机Windows接收图像实验

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`



```
C:\Windows\system32\cmd.e... Python3.8 environment has been set with openCV+pymavlink+numpy+pyulog etc. You can use pip or pip3 command to install other libraries Put Python38Run.bat into your code folder Use the command: 'python XXX.py' to run the script with Python C:\Users\uavcs\Desktop\demo\8.RflySimVision\0.ApiExps\0.Preparation\4.AutoObtainIPRun>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 `python 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

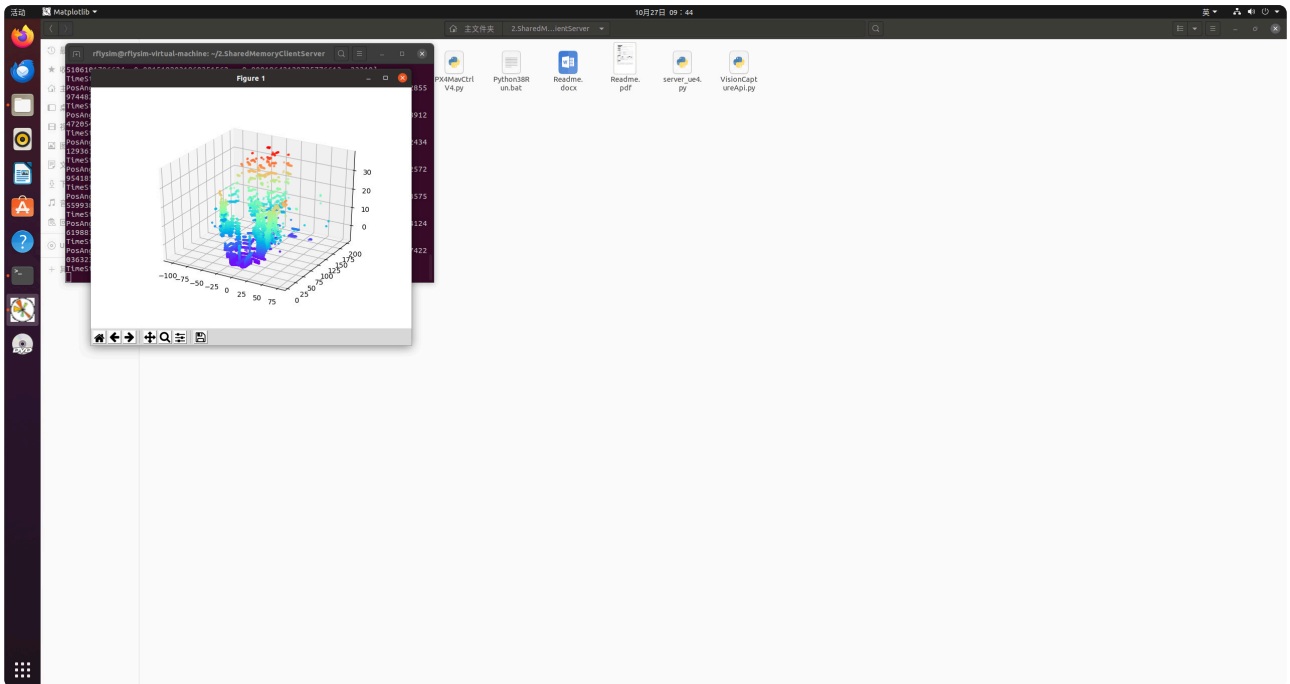
步骤1同上面的Step1步骤。

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，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：



```
uav11@ubuntu: ~/桌面/1.Point-CloudCommSHM
uav11@ubuntu:~/桌面/1.Point-CloudCommSHM$ python3 server_ue4.py
current ros environment noetic
Json use relative path mode
jsonPath= /home/uav11/桌面/1.Point-CloudCommSHM/Config.json
Got 1 vision sensors from json
Start lisening to timeStamp Msg
Got time msg from CopterSim # 1
CopterSim not on this PC
1704365568.73
1704365568.1045835
Got CopterSim time Data for img
Got start time for SeqID # 0
Start Image Reciver
PX4 Armed!
PosAngNum: [-0.05483430624008179, 0.009500881657004356, 9.397844314575195, -0.0
01651692553423345, -0.0016363858012482524, 0.00812564231455326, 24519]
TimeStmp: 67.28587921142578
```



Step 3: 结束仿真

在下图“

`client_ue4_SITL.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。

```
C:\WINDOWS\system32\cmd.exe

-----
Start QGroundControl
Kill all CopterSims
Starting PX4 Build
[1/1] Generating ../../logs
killing running instances
starting instance 1 in /mnt/c/PX4PSPFull/Firmware/build/px4_sitl_default/instance_1
PX4 instances start finished
Press any key to exit
```

按下回车键，快速关闭所有仿真窗口

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库的版本的版本的原因,可通过添加如下语句解决
`from mpl_toolkits.mplot3d import Axes3D`