

1、实验名称及目的

点云数据传输实验：通过平台接口在 `client_ue4.py` 客户端共享内存接收点云数据，经过平台共享内存方式发出，`server_ue4.py` 接收点云数据。

2、实验原理

首先更改 `client_ue4_SITL.bat`、`client_ue4.py`、`server_ue4.py`、`Config.json` 的 IP 地址，使用指定 IP 传输，使得传输更加稳定与快捷，然后运行 `client_ue4_SITL.bat` 开启一个飞机的软件在环仿真，然后运行 `client_ue4.py` 文件，通过 `VisionCaptureApi` 接口创建一个取图接口，并通过 `Config.json` 文件加载传感器配置，其中配置文件参数含义如下

“SeqID”代表第几个传感器。此处表示第 1 个传感器（免费版只支持 2 个图）。

“TypeID”代表传感器类型 ID，此处 4 代表输出点云为激光雷达坐标系；

“TargetCopter”传感器装载的目标飞机的 ID，可改变。“TargetMountType”代表坐标类型，0：固定飞机上（相对几何中心），1：固定飞机上（相对底部中心），2：固定地面上（监控）也可变。

“DataWidth”为激光雷达一个 ring 内的点云个数为 900

“DataHeight”为激光雷达线束数量 32。

“DataCheckFreq”点云发布频率此处为 10HZ。

“SendProtocol[8]”为传输方式与地址，`SendProtocol[0]`取值 0，表示使用共享内存进行数据传输（0：共享内存（免费版只支持共享内存），1：UDP 直传）

“CameraFOV”为相机视场角（仅限视觉类传感器），单位度也可改变。

“SensorPosXYZ[3]”为传感器安装位置，单位米也可改变。

“SensorAngEular[3]”为传感器安装角度，单位度°也可改变。

“otherParams[8]”：[激光最远距离(m),精度(m),水平扫描角度下限值(度),水平扫描角度上限值(度),垂直扫描角度下限值(度),垂直扫描角度上限值(度),预留,预留]

然后运行 `client_ue4.py` 向 `RflySim3D` 发送取图请求，通过 UDP 直传 png 压缩的方式将图像数据传输到 Ubuntu 虚拟机，Ubuntu 虚拟机运行 `server_ue4.py` 通过 `PX4MavCtrlr` 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。

3、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。

4、文件目录

文件夹/文件名称	说明
<code>client_ue4.py</code>	Windows 下客户端文件
<code>server_ue4.py</code>	Ubuntu 下服务端文件

Config.json	视觉传感器配置文件
lidar.rviz	Rviz 配置文件

5、运行环境

序号	软件要求	硬件要求	
		名称	数量(个)
1	Windows 10 及以上版本	笔记本/台式电脑 ^①	1
2	RflySim 平台高级版及以上		
3	Visual Studio Code		
4	Ubunt 虚拟机		

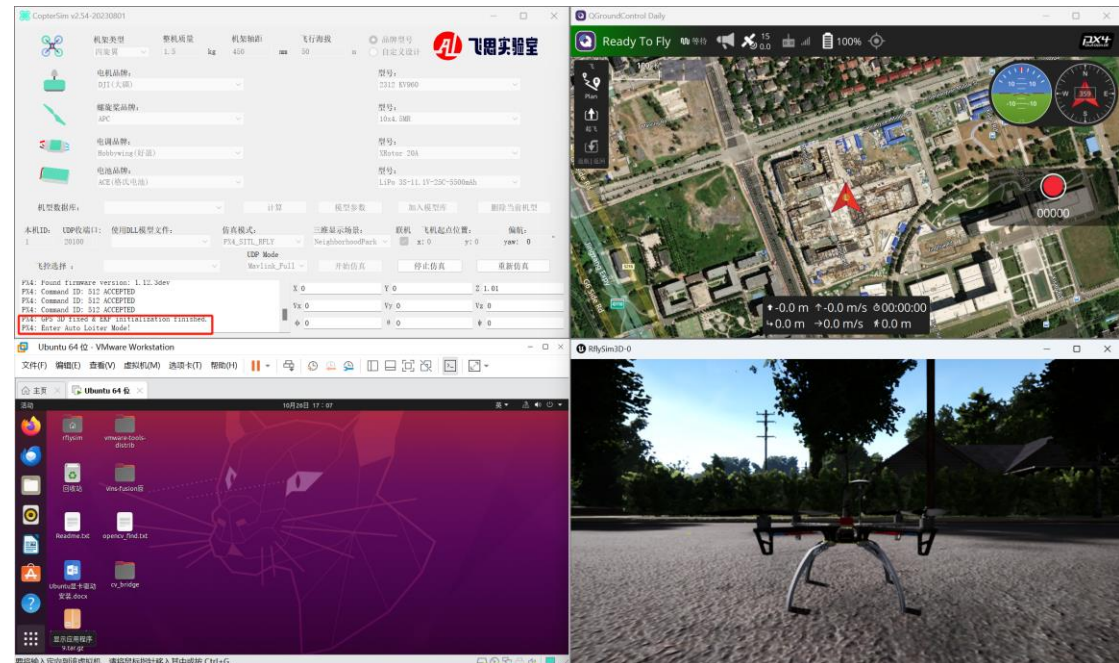
①：推荐配置请见：<https://doc.rflysim.com/1.1InstallMethod.html>

6、实验步骤

Step 1:

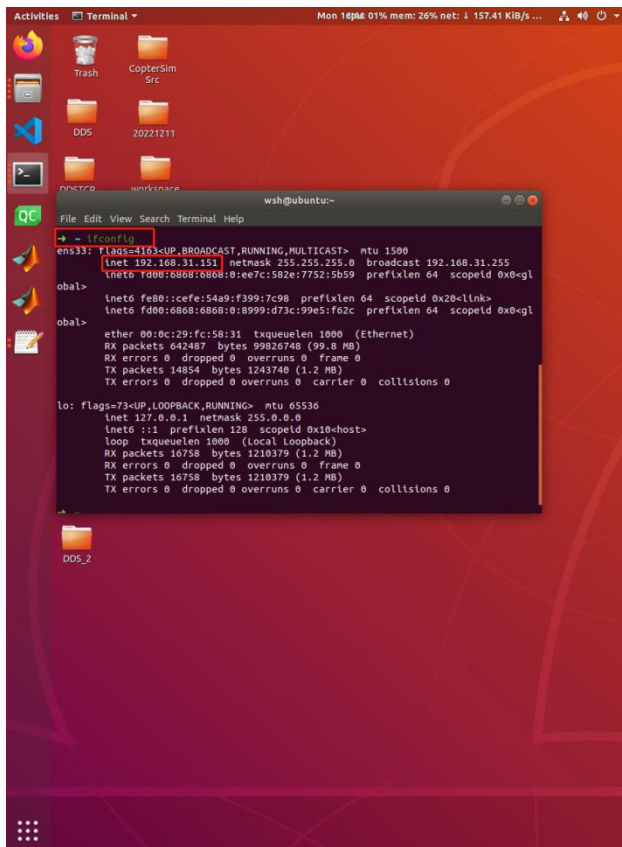
以管理员方式运行 LidarAPIDemo.bat 开启一个飞机的软件在环仿真。将会启动 1 个 QGC 地面站，1 个 CopterSim 软件且其软件下侧日志栏必须打印出 GPS 3D fixed & EKF initialization finished 字样代表初始化完成，并且 RflySim3D 软件内有 1 架无人机。

并且启动一个 Ubuntu 虚拟机。



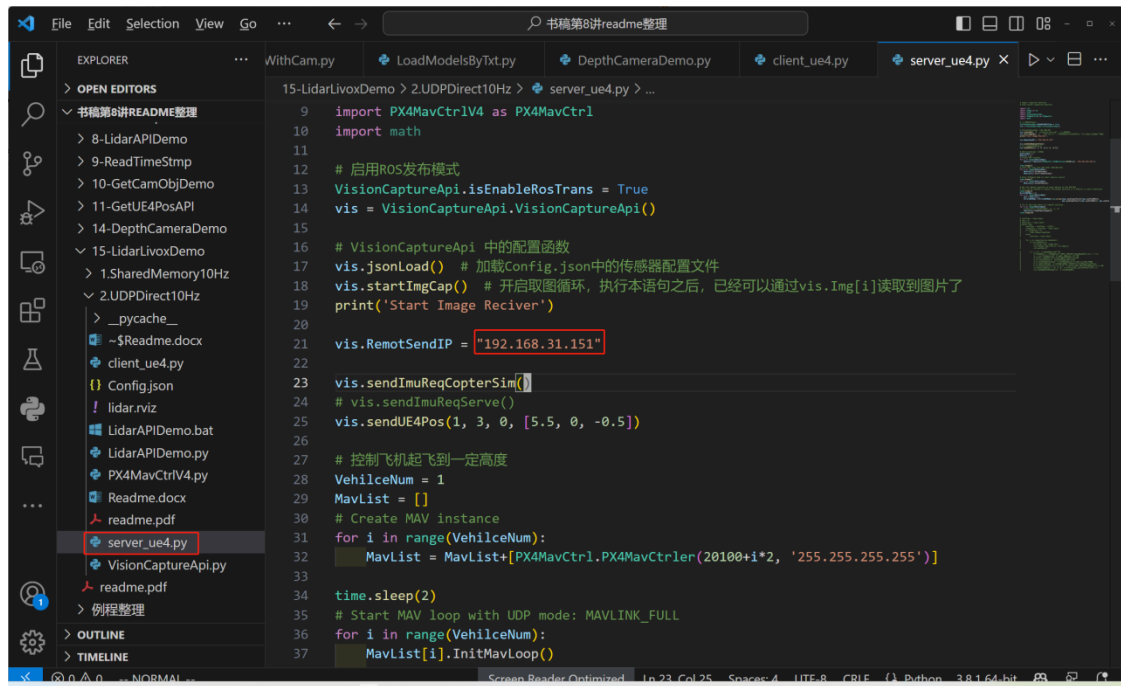
Step 2:

在虚拟机终端中输入 ifconfig 命令，查找该虚拟机的 IP 地址，如下图：



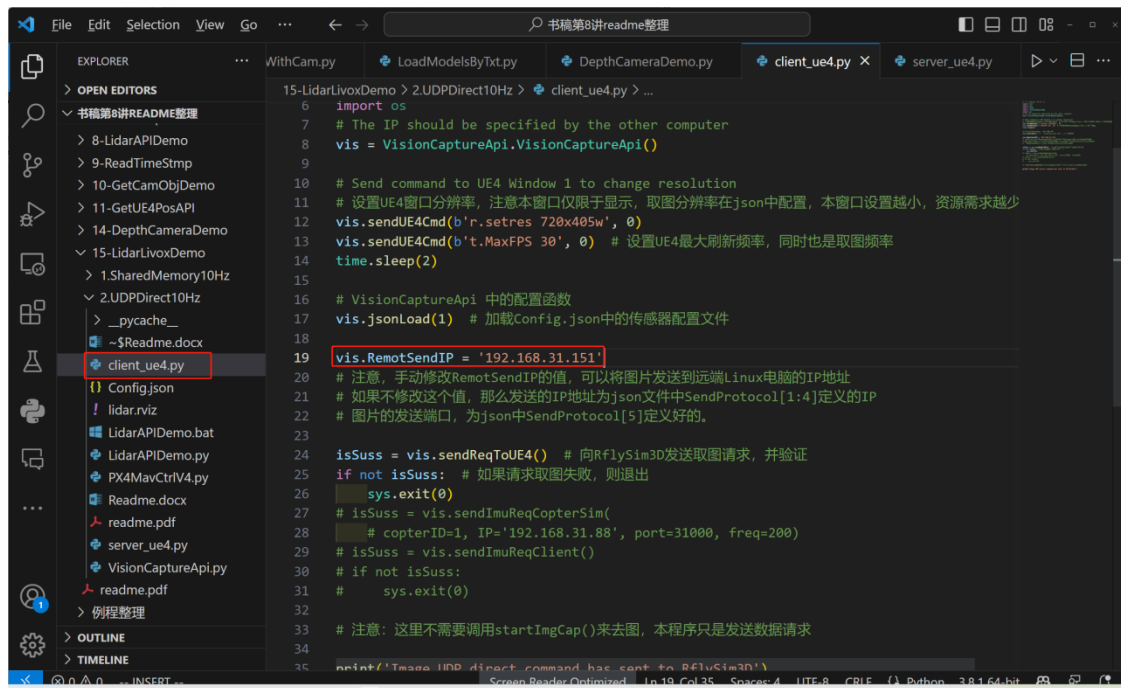
Step 3:

修改 client_ue4.py 与 server_ue4.py 中的代码 vis.RemotSendIP 值，改成自己的虚拟机地址。在 Windows 运行 client_ue4.py。



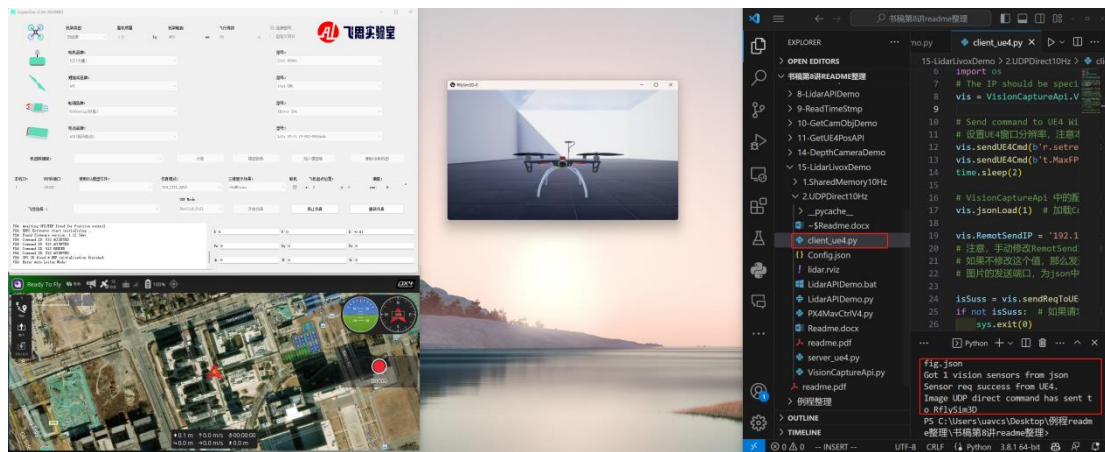
This screenshot shows the VS Code editor with the file explorer on the left and the editor window displaying `server_ue4.py`. The file explorer shows a project structure with folders like `8-LidarAPIDemo`, `9-ReadTimeStmp`, `10-GetCamObjDemo`, `11-GetUE4PosAPI`, `14-DepthCameraDemo`, `15-LidarLivoxDemo`, `1.SharedMemory10Hz`, and `2.UDPDirect10Hz`. The `server_ue4.py` file is selected in the explorer. The editor window shows the following code:

```
9 import PX4MavCtrlV4 as PX4MavCtrl
10 import math
11
12 # 启用ROS发布模式
13 VisionCaptureApi.isEnableRosTrans = True
14 vis = VisionCaptureApi.VisionCaptureApi()
15
16 # VisionCaptureApi 中的配置函数
17 vis.jsonLoad() # 加载Config.json中的传感器配置文件
18 vis.startImgCap() # 开启取图循环, 执行本语句之后, 已经可以通过vis.Img[i]读取到图片了
19 print('Start Image Receiver')
20
21 vis.RemotSendIP = "192.168.31.151"
22
23 vis.sendImuReqCopterSim()
24 # vis.sendImuReqServe()
25 vis.sendUE4Pos(1, 3, 0, [5.5, 0, -0.5])
26
27 # 控制飞机起飞到一定高度
28 VehicNum = 1
29 MavList = []
30 # Create MAV instance
31 for i in range(VehicNum):
32     MavList = MavList+[PX4MavCtrl.PX4MavCtrl(20100+i*2, '255.255.255.255')]
33
34 time.sleep(2)
35 # Start MAV loop with UDP mode: MAVLINK_FULL
36 for i in range(VehicNum):
37     MavList[i].InitMavLoop()
```



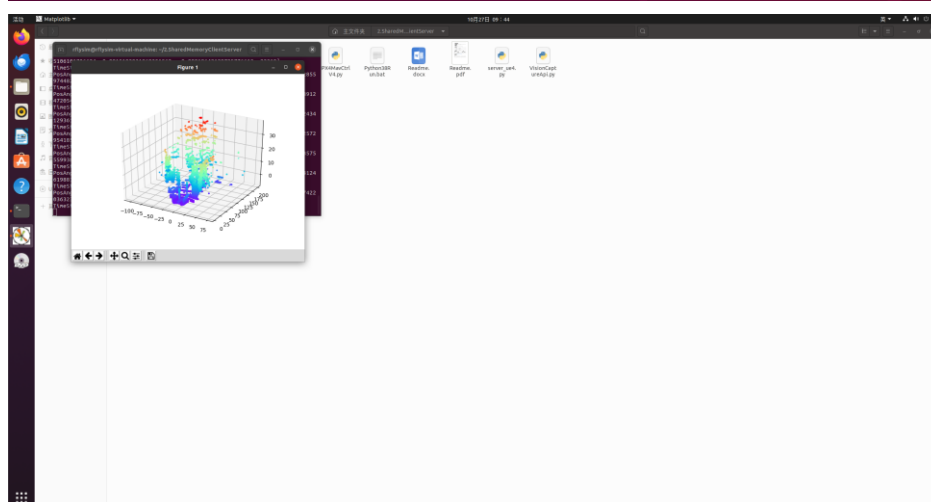
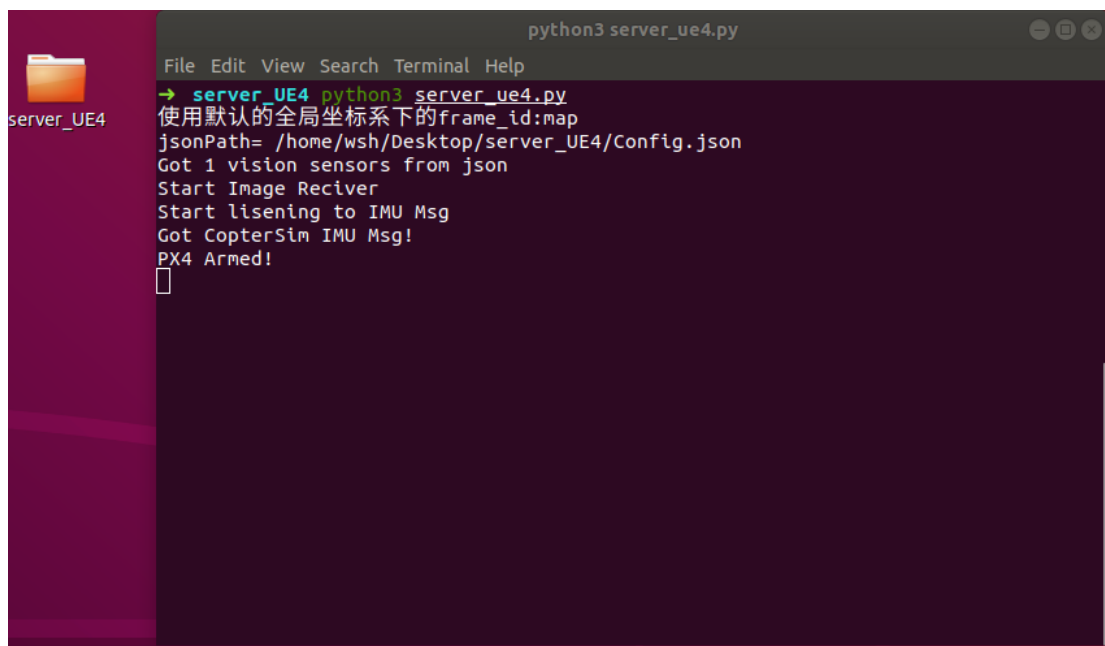
This screenshot shows the VS Code editor with the file explorer on the left and the editor window displaying `client_ue4.py`. The file explorer shows the same project structure as the previous screenshot. The `client_ue4.py` file is selected in the explorer. The editor window shows the following code:

```
6 import os
7 # The IP should be specified by the other computer
8 vis = VisionCaptureApi.VisionCaptureApi()
9
10 # Send command to UE4 Window 1 to change resolution
11 # 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
12 vis.sendUE4Cmd(b'r.setres 720x405w', 0)
13 vis.sendUE4Cmd(b't.MaxFPS 30', 0) # 设置UE4最大刷新频率, 同时也是取图频率
14 time.sleep(2)
15
16 # VisionCaptureApi 中的配置函数
17 vis.jsonLoad(1) # 加载Config.json中的传感器配置文件
18
19 vis.RemotSendIP = '192.168.31.151'
20 # 注意, 手动修改RemotSendIP的值, 可以将图片发送到远端Linux电脑的IP地址
21 # 如果不修改这个值, 那么发送的IP地址为json文件中SendProtocol[1:4]定义的IP
22 # 图片的发送端口, 为json中SendProtocol[5]定义好的。
23
24 isSuss = vis.sendReqToUE4() # 向RflySim3D发送取图请求, 并验证
25 if not isSuss: # 如果请求取图失败, 则退出
26     sys.exit(0)
27 # isSuss = vis.sendImuReqCopterSim(
28 #     # copterID=1, IP='192.168.31.88', port=31000, freq=200)
29 # isSuss = vis.sendImuReqClient()
30 # if not isSuss:
31 #     sys.exit(0)
32
33 # 注意: 这里不需要调用startImgCap()来去图, 本程序只是发送数据请求
34
35 print('!Teage UDP direct command has sent to RflySim3D')
```



Step 4:

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：



Step 5:

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

Step 6:

在下图 VScode 中，点击“终止终端”，可以彻底退出脚本运行。



7、参考文献

[1]. 无

8、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
```