

1、实验名称及目的

自定义 ROS 系统 tf 树实验：通过平台接口自定义更改 frame_id 接口。

2、实验原理

首先更改 RunRflysim3DSITL.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，通过 Config.json 文件配置传感器，其中参数含义如下

“SeqID”代表第几个传感器。此处共有四个相机 0、1、2、3 分别表示第一、二、三、四个相机。

“TypeID”代表传感器类型 ID，1:RGB 图（免费版只支持 RGB 图），2:深度图，3:灰度图；

“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”为数据或图像宽度此处为 640

“DataHeight”为数据或图像高度此处为 480。

“DataCheckFreq”检查数据更新频率此处为 30HZ。

“SendProtocol[8]”为传输方式与地址，SendProtocol[0]取值 1 表示使用 UDP 直传 png 压缩（0：共享内存（免费版只支持共享内存），1：UDP 直传 png 压缩，2：UDP 直传图片不压缩，3：UDP 直传 jpg 压缩）

“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。

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，并将包含传感器数据的话题发布，将 vision.rviz 文件进行修改使得 Rviz 能够订阅到 server_ue4.py 发布的传感器数据话题，从而根据仿真的情况创建 tf 树。

3、实验效果

本实验通过平台接口进行 ROS 系统 tf 树自定义更改 frame_id。

4、文件目录

文件夹/文件名称	说明
RunRflysim3DSITL.bat	启动仿真配置文件
client_ue4.py	Windows 下客户端文件

server_ue4.py	Ubuntu 下服务端文件
Config.json	视觉传感器配置文件
vision.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		

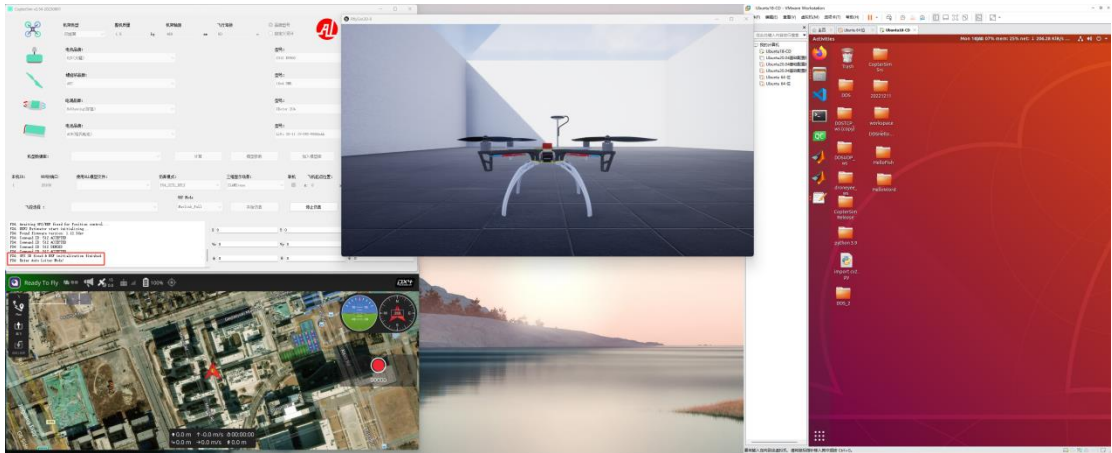
① ：推荐配置请见：<https://doc.rflysim.com/1.1InstallMethod.html>

6、实验步骤

Step 1:

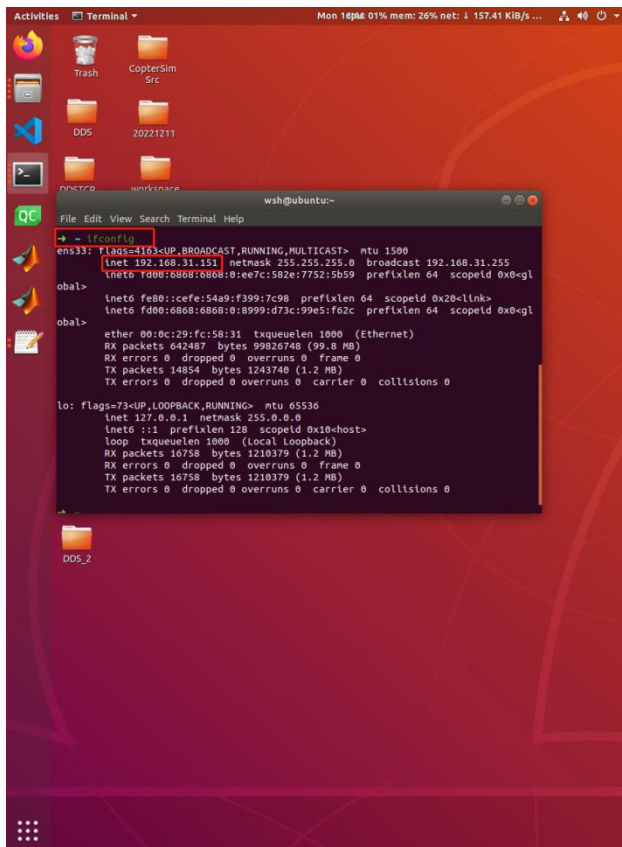
以管理员方式运行 RunRflysim3DSITL.bat 开启一个飞机的软件在环仿真。将会启动 1 个 QGC 地面站，1 个 CopterSim 软件且其软件下侧日志栏必须打印出 GPS 3D fixed & EKF initialization finished 字样代表初始化完成，并且 RflySim3D 软件内有 1 架无人机。

并且启动一个已安装了 ros 的 Ubuntu 虚拟机。



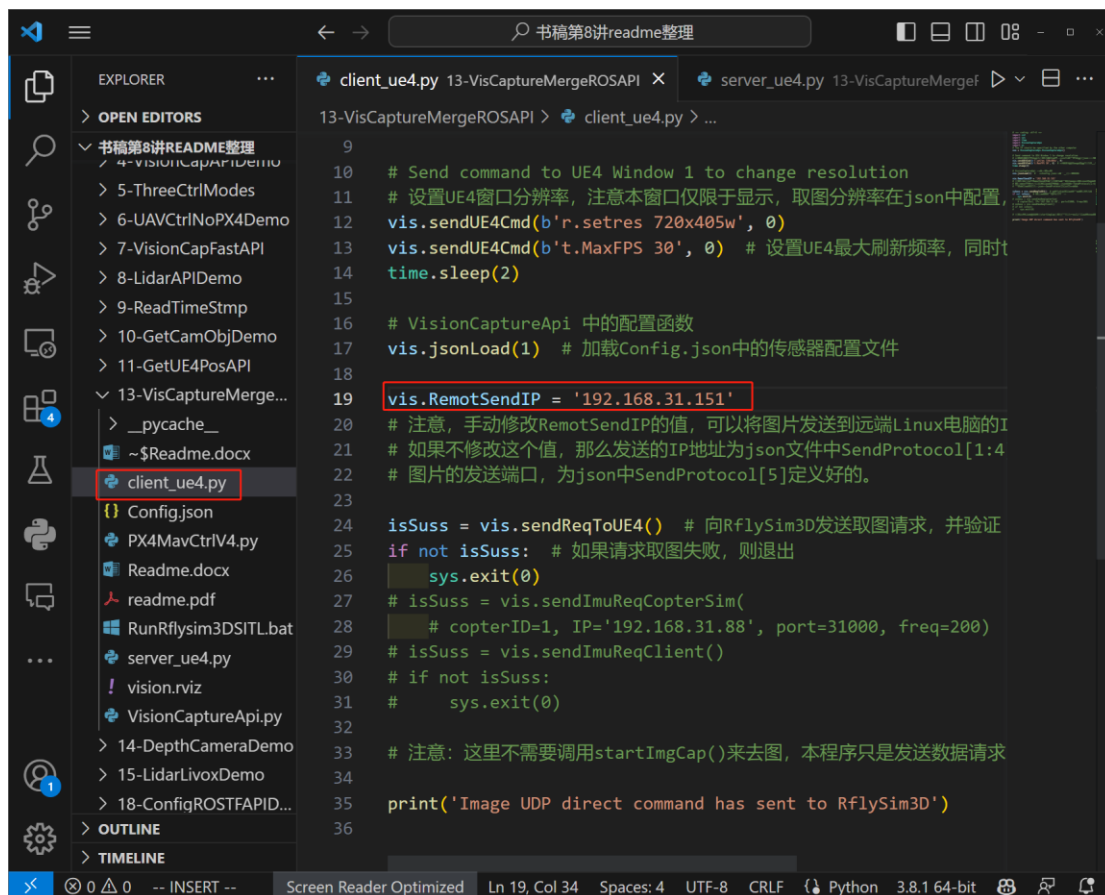
Step 2:

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：



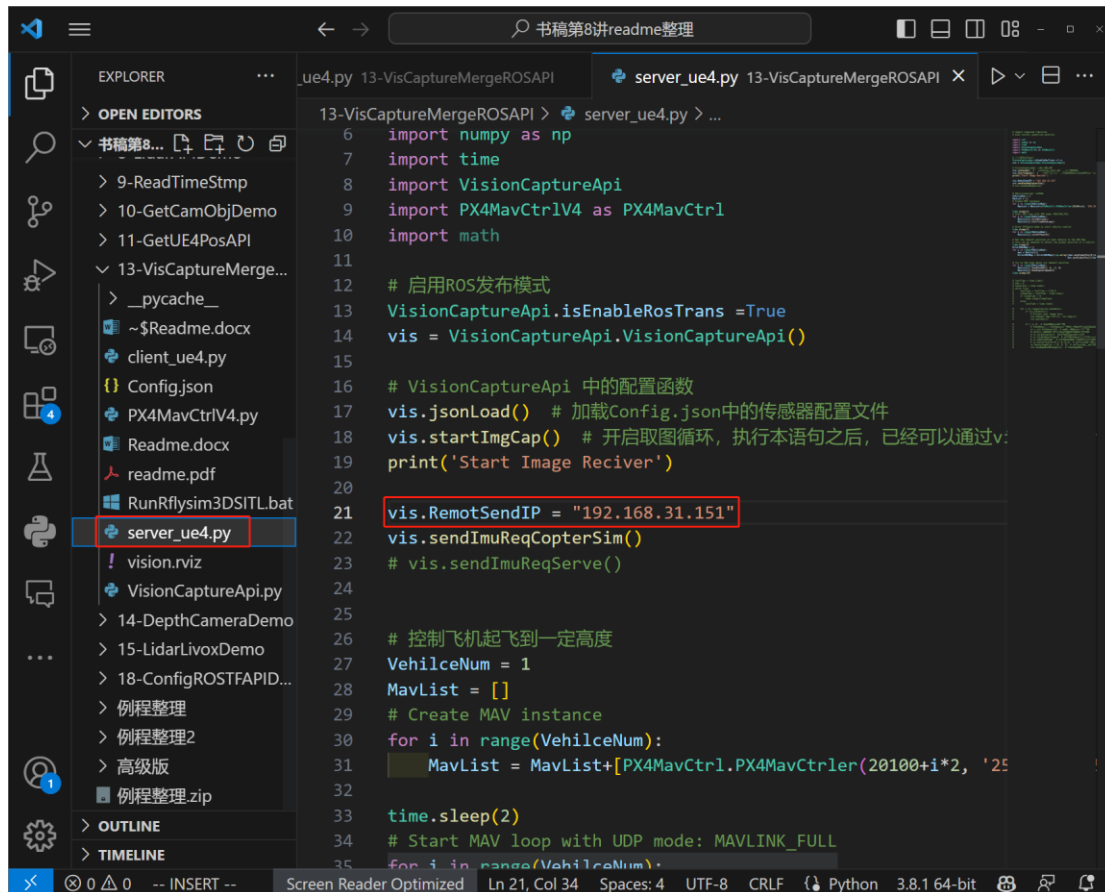
Step 3:

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。并修改文件夹下 Config.json 配置文件下的 SendProtocol 的第一个数为 1，以及将 IP 改成虚拟机下的 IP 地址，确保可以与虚拟机通信，如果是 0 则是 Windows 下本地通信。



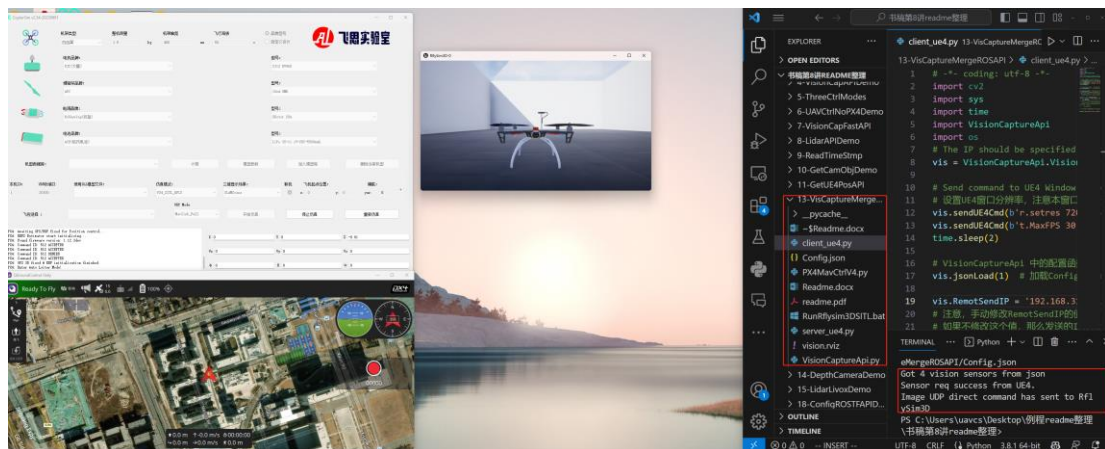
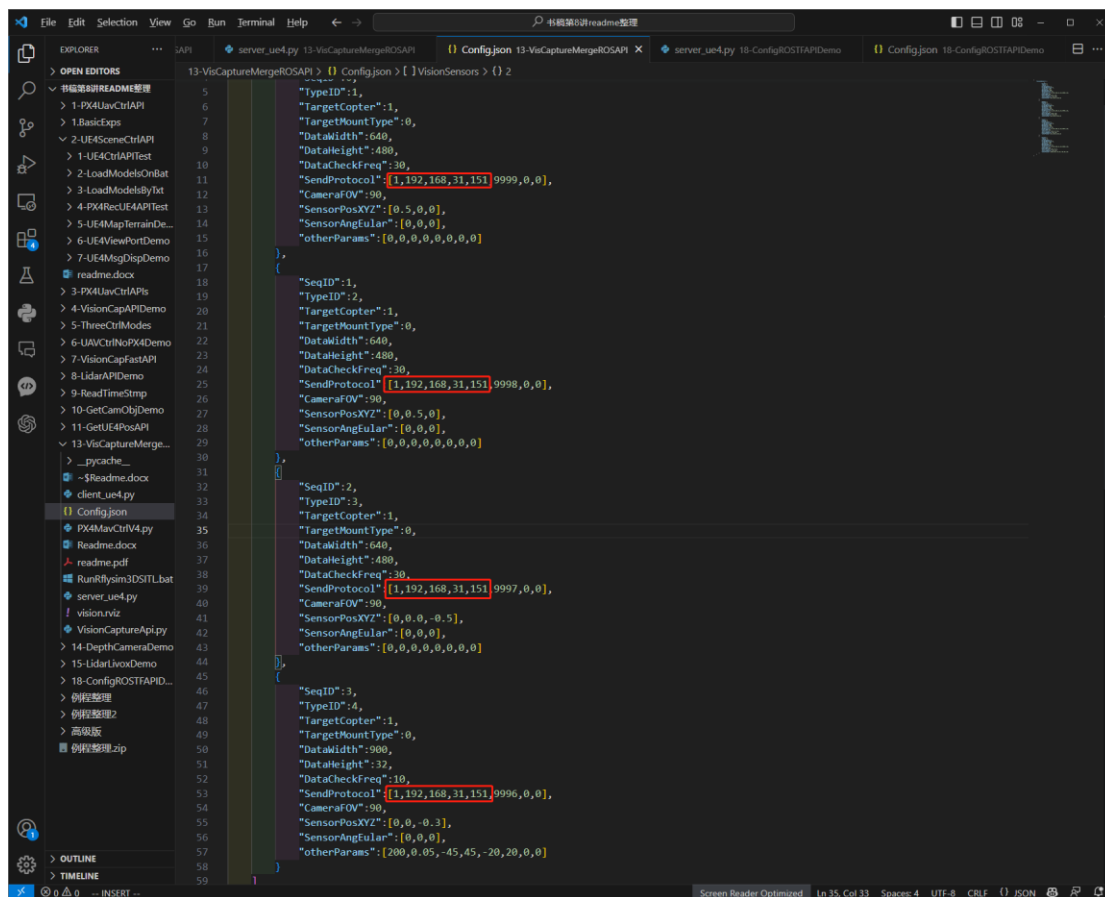
This screenshot shows the VS Code editor with the file explorer on the left and the code editor in the center. The file explorer shows a project structure with a folder named '书稿第8讲README整理' containing several files. The file 'client_ue4.py' is selected and highlighted in red. The code editor displays the contents of 'client_ue4.py', which is a Python script for sending commands to a UE4 window. The script includes comments in Chinese and Python code for setting resolution, FPS, and sending a UDP command. The line 'vis.RemotSendIP = '192.168.31.151'' is highlighted in red. The status bar at the bottom indicates the file is at line 19, column 34, with a UTF-8 encoding and CRLF line endings.

```
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,
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t
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I
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
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('Image UDP direct command has sent to RflySim3D')
36
```



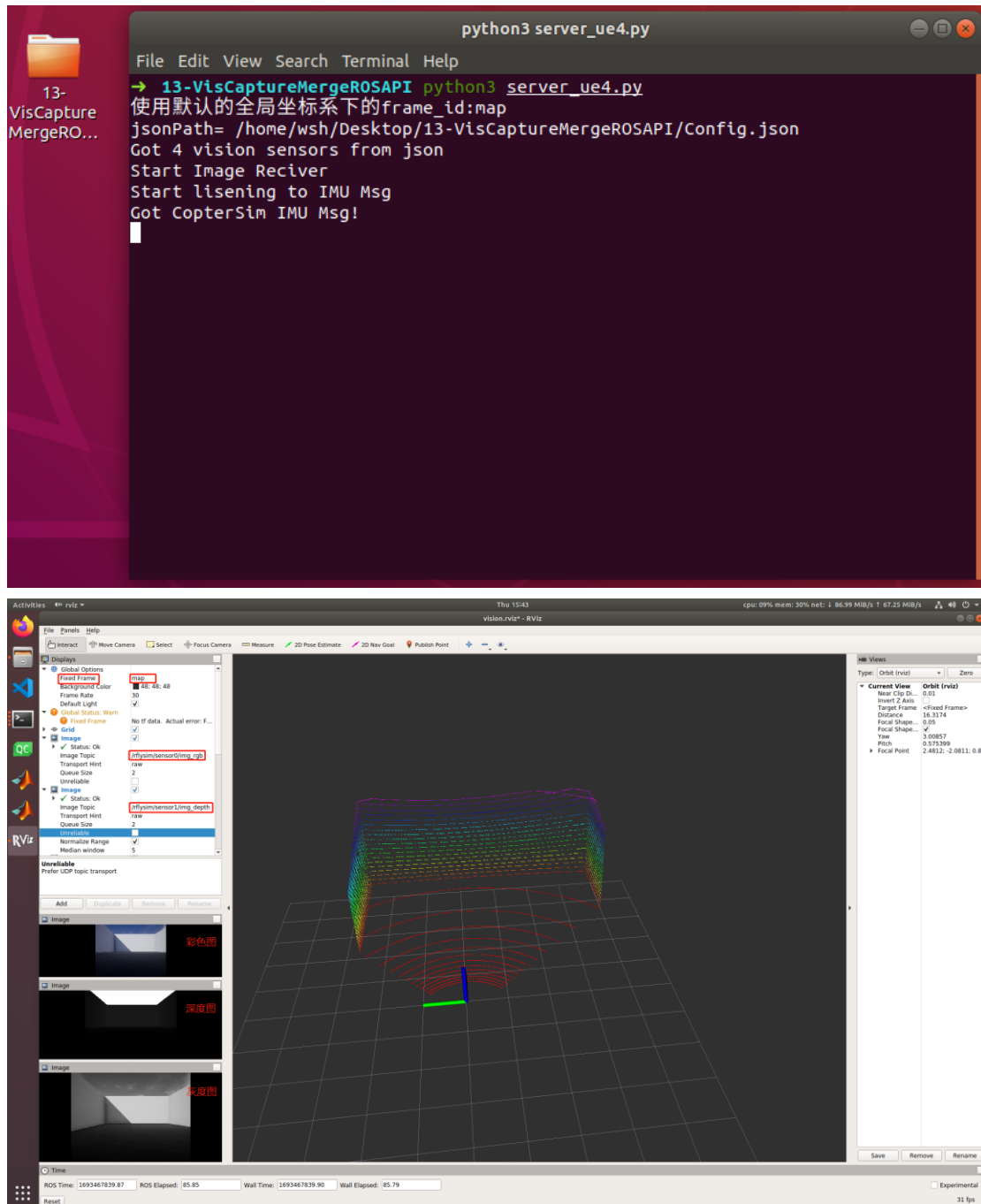
This screenshot shows the VS Code editor with the file explorer on the left and the code editor in the center. The file explorer shows the same project structure as the previous screenshot, but the file 'server_ue4.py' is selected and highlighted in red. The code editor displays the contents of 'server_ue4.py', which is a Python script for receiving commands from a UE4 window. The script includes imports for numpy, time, VisionCaptureApi, PX4MavCtrlV4, and math. It sets 'VisionCaptureApi.isEnableRosTrans = True' and creates a 'vis' object. The line 'vis.RemotSendIP = "192.168.31.151"' is highlighted in red. The script also includes a loop for sending commands to a MAV instance. The status bar at the bottom indicates the file is at line 21, column 34, with a UTF-8 encoding and CRLF line endings.

```
6 import numpy as np
7 import time
8 import VisionCaptureApi
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v:
19 print('Start Image Reciver')
20
21 vis.RemotSendIP = "192.168.31.151"
22 vis.sendImuReqCopterSim()
23 # vis.sendImuReqServe()
24
25
26 # 控制飞机起飞到一定高度
27 VehilceNum = 1
28 MavList = []
29 # Create MAV instance
30 for i in range(VehilceNum):
31     MavList = MavList+[PX4MavCtrl.PX4MavCtrl(20100+i*2, '25
32
33 time.sleep(2)
34 # Start MAV loop with UDP mode: MAVLINK_FULL
35 for i in range(VehilceNum):
```



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`。然后另起一个终端运行 `rviz` 命令，打开 `rviz` 可视化工具，加载文件夹下的 `vision.rviz` 配置文件，然后修改 Fixed Frame 为 `map`，然后在三个 Image 以及 PointCloud 2 下的话题填入图像数据话题及点云数据话题 `/rflysim/sensor0/img_rgb`(彩色图)、`/rflysim/sensor1/img_depth`(深度图)、`/rflysim/sensor2/img_gray`(灰度图)、`/rflysim/sensor3/vehicle_lidar` 可看到如下效果：



Step 5:

在下图“RunRflysim3DSITL.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