

1. 实验名称及目的

1.1 实验名称

数据UDP直传png压缩实验

1.2 实验目的

尝试使用UDP直传png压缩的传输的方式传图，并在远端Linux系统（WinWSL、虚拟机、机载板卡、智能视觉盒子）或另一台Windows电脑中接收图像，并回传飞机控制指令。

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。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.jsonLoad() \# 加载Config.json中的传感器配置文件
2
3 isSuss = vis.sendReqToUE4(0, TargetIP) \#向RflySim3D发送取图请求, 发给ip为TargetIP的地址
4
5 vis.startImgCap() \# 开启取图
6
7 vis.hasData[i] \# 图片i数据是否更新
8
9 vis.Img[i] \# 图片i数据 (像素矩阵)
10
11 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 vis = VisionCaptureApi.VisionCaptureApi(TargetIP) \#创建一个视觉传感器实例, 这个实例对应的i
6
7 req.sendReSimIP(CopterID) \# 请求mavlink数据到本电脑
```

3) 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, 1和2的三个视觉传感器
2
3 "TypeID":1 : 传感器类型为RGB彩色图像
4
5 "TargetCopter":1 : 相机绑定在1号飞机上
6
7 "SendProtocol":[3,0,0,0,0,0,0,0] : 传输模式为3 : UDP直传图片png压缩。
8
9 "SensorPosXYZ":[0.3,0,0] : 相机分布位置。
```

4) 飞机控制指令

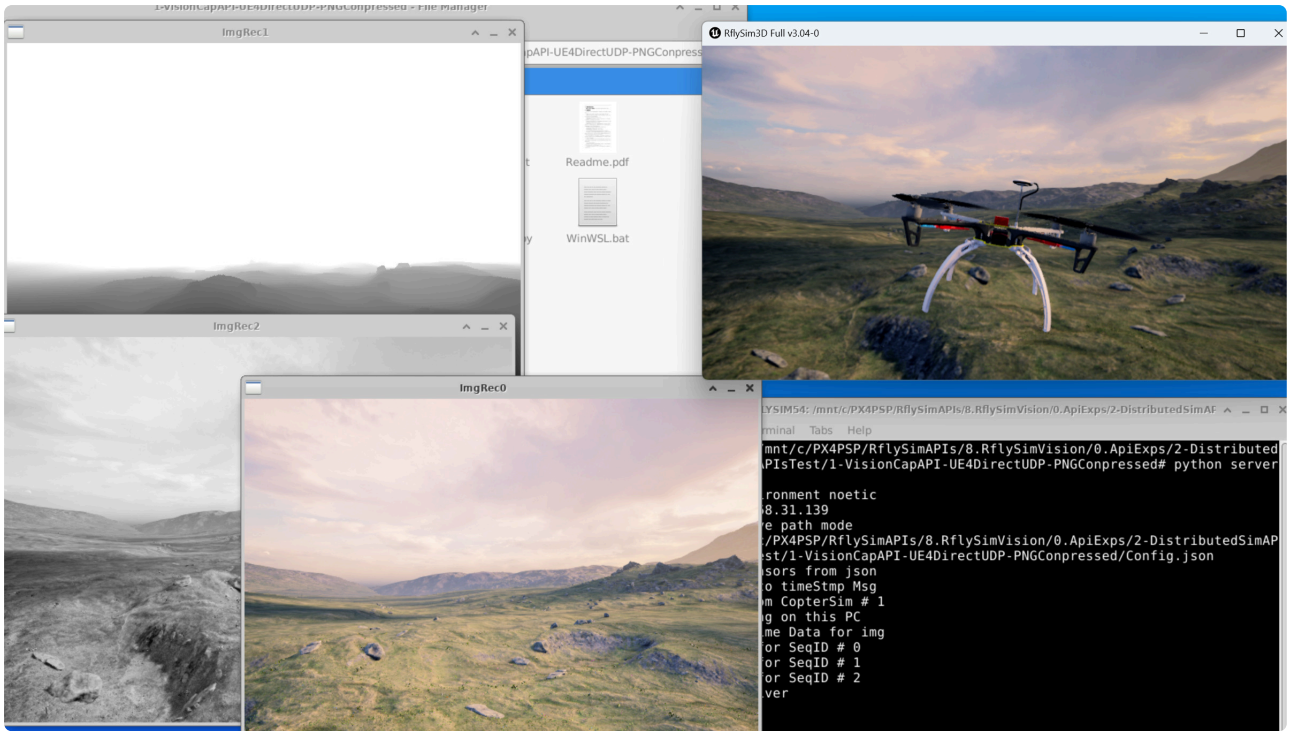
```
1 MavList = MavList+[PX4MavCtrl.PX4MavCtrl(CopterID,TargetIP)] \#初始化并建立i号飞机的MA
2
3 MavList[i].InitMavLoop() \# 初始化Mavlink监听程序, 读取第i个飞机数据
4
5 MavList[i].initOffboard() \# 第i个飞机进入Offboard模式
6
7 Error2UE4Map =Error2UE4Map+[-np.array([mav.uavGlobalPos[0]-mav.uavPosNED[0],mav.uavGl
8
9 MavList[i].SendPosNED(0, 0, -10, 0) \# 对第i个飞机发送10米高的位置控制指令
10
11 mav=MavList[j] \# 从列表中取第i个实例作为当下处理的飞机实例
12
13 MavList[i].endOffboard() \# 对第i个飞机进入endoffboard模式
14
15 MavList[i].stopRun() \# 终止第i个飞机
```

5) 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

在启动服务端后飞机起飞并按照一定的路线飞行, 并生成如图所示三个窗口。



3. 文件目录

例程目录：

[[安装目录](#)]\RflySimAPIs\8.RflySimVision\0.ApiExps\1-UsageAPI\4.SendProtocolAPI

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）
server_ue4Ros.py	Ros环境下的取图控制程序 （支持Windows或Linux分布式运行）
Config.json	视觉传感器配置文件
client_ue4_HITL.bat	硬件在环仿真脚本
client_ue4_SITL.bat	软件在环仿真一键脚本（MAVLinkFull模式）
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。

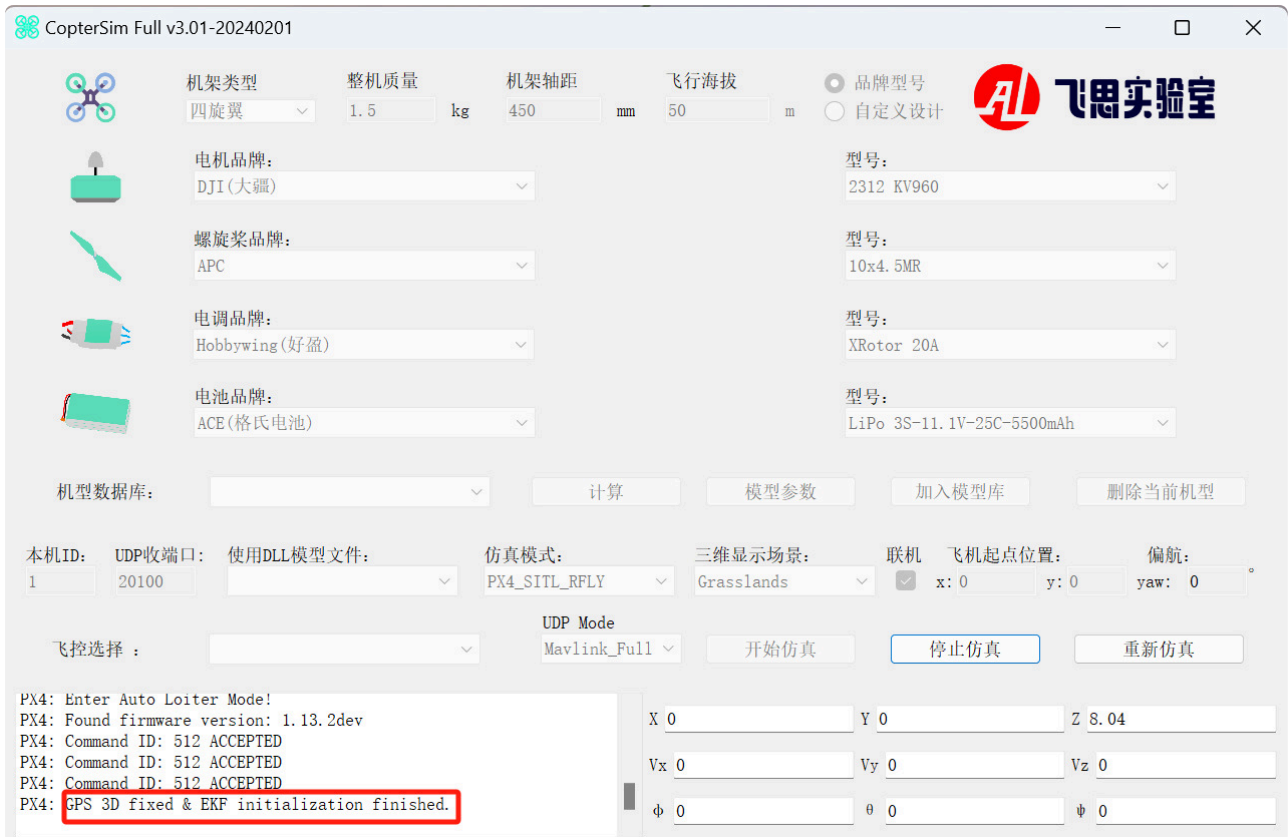
①：推荐配置请见：<https://rflysim.com/doc/zh/HowToInstall.pdf>

5. 实验步骤

必做实验：WinsWSL控制

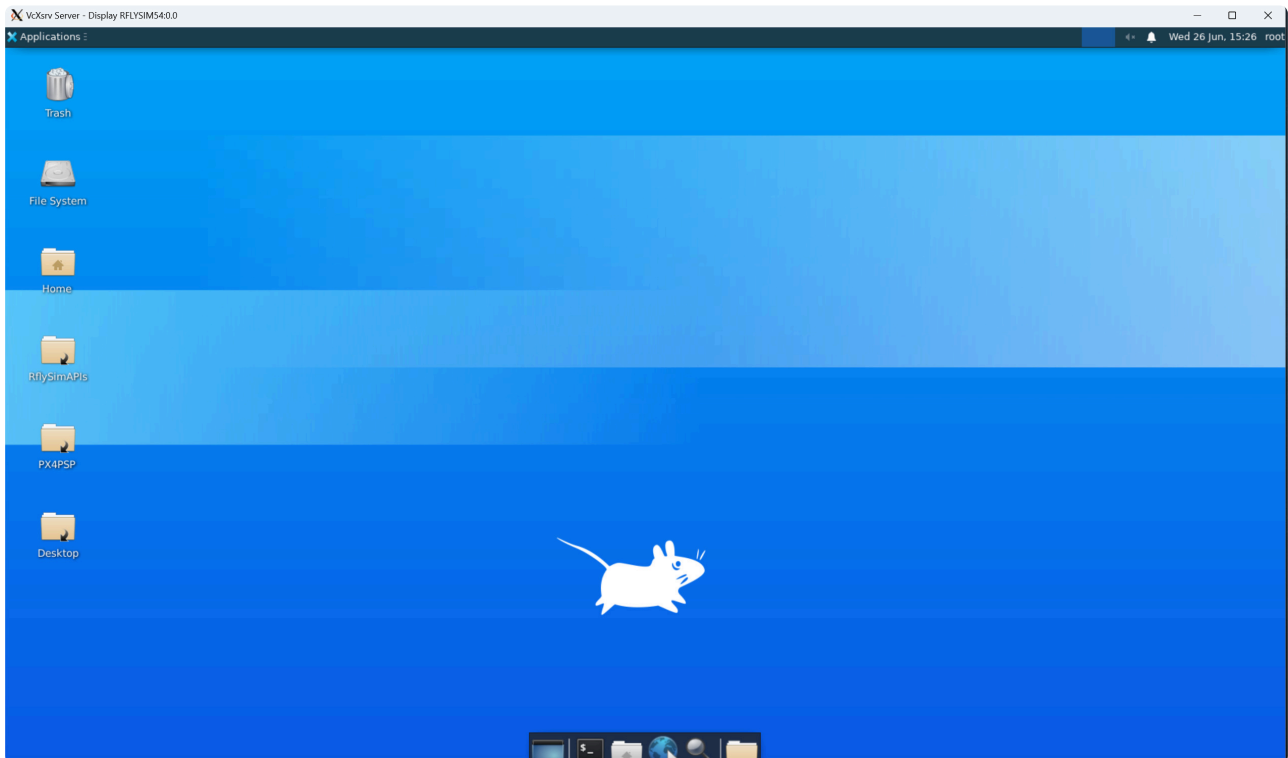
Step 1：开启仿真

双击运行 `client_ue4_SITL.bat` 启动仿真界面。等待CopterSim界面初始化完成。



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。）

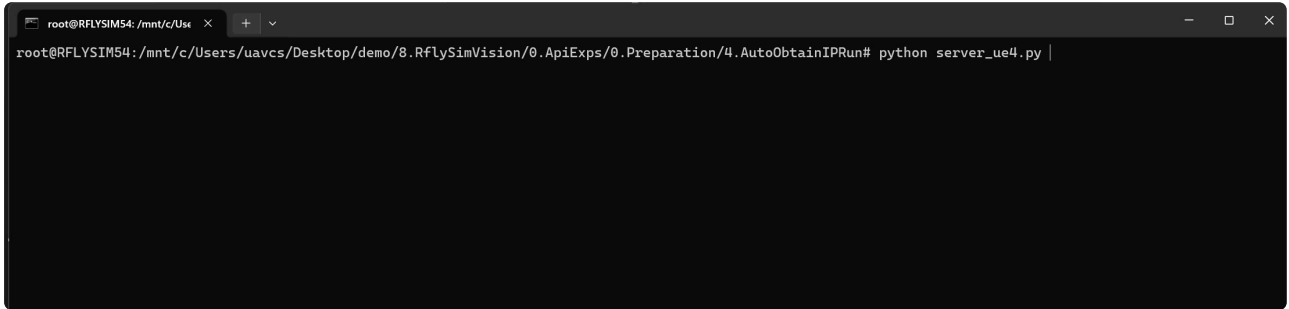


注意：参考 [\[安装目录\]\RflySimAPIs\1.RflySimIntro\2.AdvExps\e7_WslGUI\Intro.pdf](#)，来了解WslGUI的功能与使用。

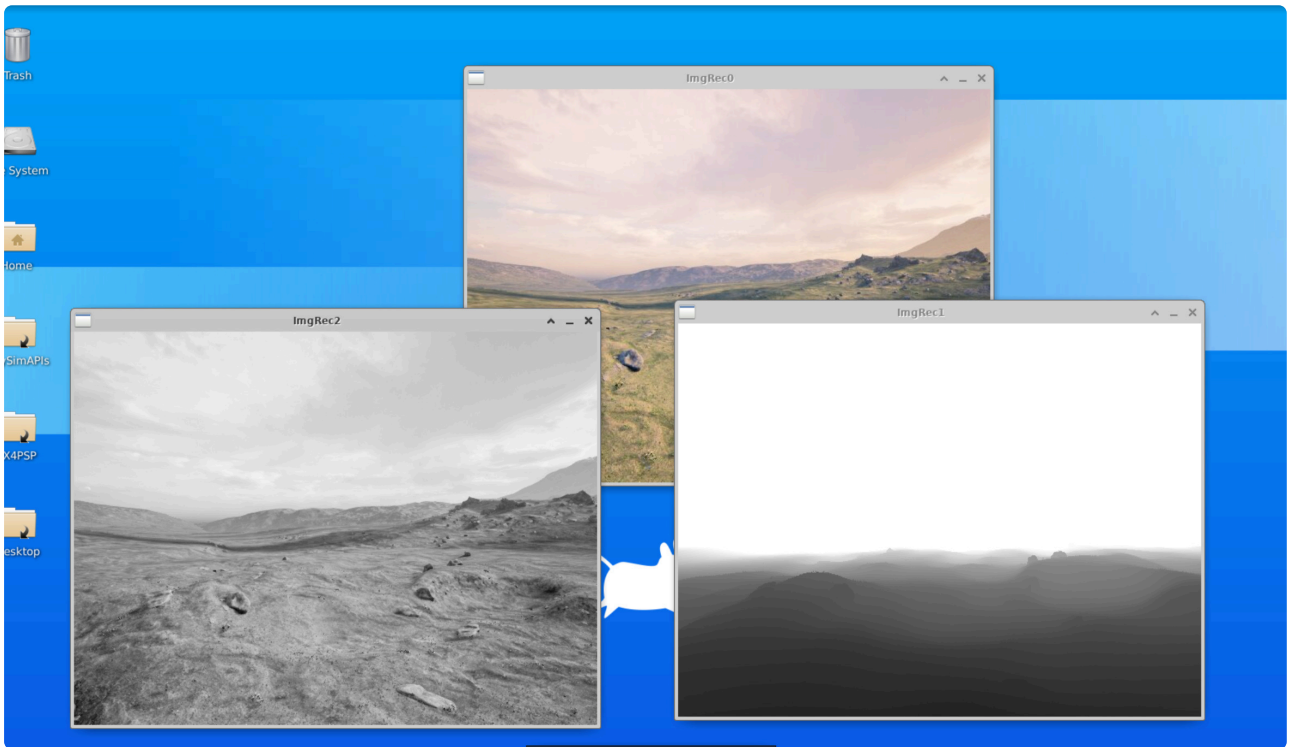
Step 3: 运行控制程序

双击打开 [WinWSL.bat](#) 进入 WSL1/Ubuntu 20.04 环境，运行命

令 `python3 server_ue4.py`，在 WslGUI 界面中可以看到生成如图所示三个窗口。



```
root@RFLYSIM54: /mnt/c/Users/uavcs/Desktop/demo/8.RflySimVision/0.ApiExps/0.Preparation/4.AutoObtainIPRun# python server_ue4.py |
```



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。

```
server_ue4.py X
> Users > uavcs > Desktop > demo > 8.RflySimVision > 0.ApiExps > 0.Preparation > 5.ManModifyIPRun > server_ue4.py > ...
1
2 # import required libraries
3 # pip3 install pymavlink pyserial
4
5 import cv2
6 import numpy as np
7 import time
8 import VisionCaptureApi
9 import PX4MavCtrlV4 as PX4MavCtrl
10 import math
11
12 StartCopterID = 1 # 初始飞机的ID号
13 TargetIP = "192.168.31.141"# 手动修改为电脑主机的IP
14 # 注意:如果是本电脑运行的话,那TargetIP是127.0.0.1的本地地址;如果是远程访问,则是192打头的局域网地址。
15 # 因此本程序能同时在本机运行,也能在其他电脑运行。
16 vis = VisionCaptureApi.VisionCaptureApi(TargetIP)
17
18 # VisionCaptureApi 中的配置函数
19 vis.jsonLoad()
20 isSuss = vis.sendReqToUE4(
21     0, TargetIP
22 )
23 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
24 print('Start Image Receiver')
25 #vis.sendImuReqCopterSim(StartCopterID, TargetIP) # 发送请求,从目标飞机CopterSim读取IMU数据,回传地址为127.0.0.1,默认频率为200Hz
26 # 执行本语句之后,会自动开启数据监听,已经可以通过vis.imu读取到IMU数据了。
27
28 VehilceNum = 1
29 MavList=[]
30 # Create MAV instance
31 for i in range(VehilceNum):
32     CopterID=StartCopterID+i # 当前配置的飞机序号
33
34     time.sleep(1)
35     MavList = MavList+[PX4MavCtrl.PX4MavCtrlIer(CopterID,TargetIP)] # 初始化并建立i号飞机的MAVLink通信连接
36
```

Run Python File
Run Python File in Dedicated Terminal
Python 调试程序: 调试 Python 文件
Python 调试程序: 使用 launch.json 进行调试

选做实验

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.GenenalUbuntuConfig\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:

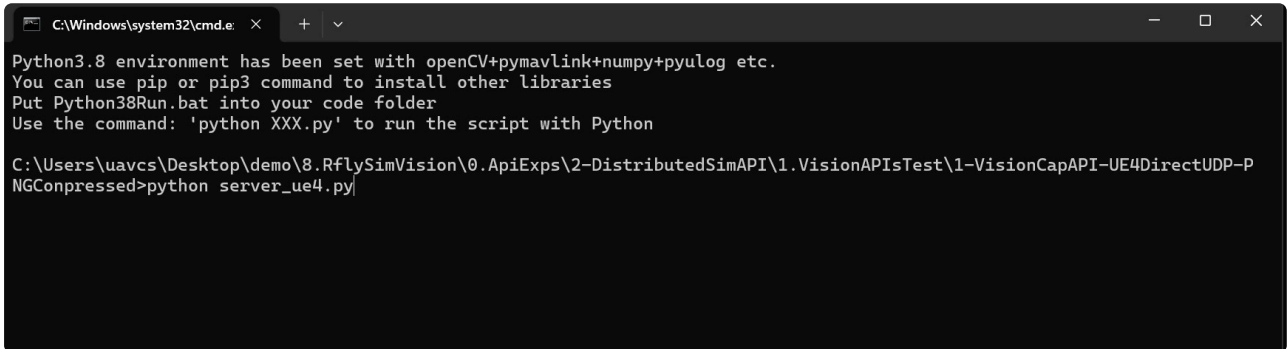
5.2.1 本机Windows接收图像实验

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. x + v
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\2-DistributedSimAPI\1.VisionAPITest\1-VisionCapAPI-UE4DirectUDP-P
NGCompressed>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

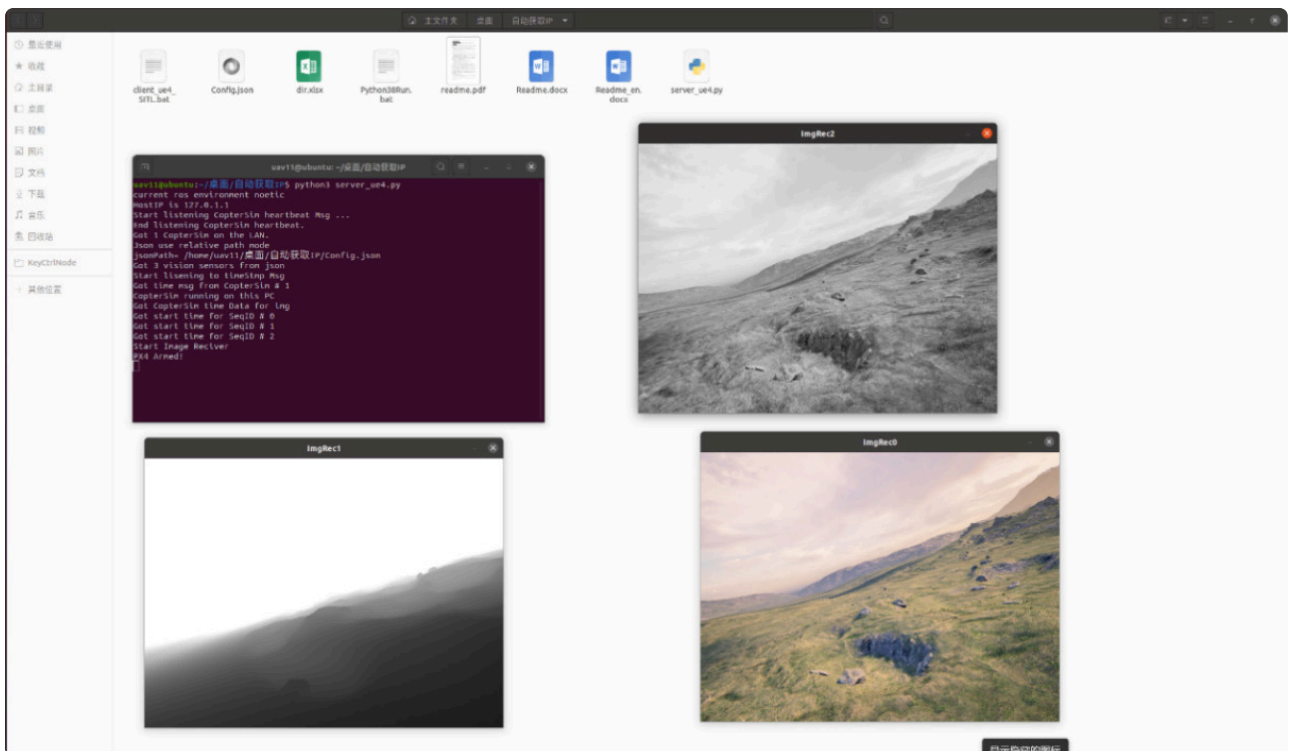
步骤1同上面的Step1步骤。

Step 2: 复制文件夹

将整个文件夹复制到Linux系统文件下。

Step 3: 运行控制程序

在文件夹路径的终端下，运行命令 `python3 server_ue4.py`，可以看到生成如图所示三个窗口。



硬件在环实验（选做）

提示1：硬件准备。首先准备一个飞控，配置好HIL_QuadCopter_X的机架仿真模式，且能正常运行HITLRun.bat并用QGroundControl控制起飞。

提示2：开始仿真。关闭所有仿真相关的程序包括QGroundControl和CopterSim等。将飞控插在电脑上，运行“`client_ue4_HITL.bat`”就能自动连接飞控，并开启一个飞机的硬件在环仿真。

****提示3：测试结果。****实现现象和前面软件在环仿真的现象和结果完全相同。

6.参考资料

无

7.常见问题

Q1: 无

A1: 无