

# 1. 实验名称及目的

## 1.1 实验名称

分布式UDP压缩传图实验（手动获取ip）

## 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VisionCaptureApi.py（见RflySimAPIs\RflySimSDK\vision目录）获取RflySim3D三种图像。不同于8.RflySimVision\0.ApiExps\0.Preparation\4.AutoObtainIPRun例程，本例程没有采用ReqCopterSim接口来自动获取IP，需要手动设置ip。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压缩传图模式以及手动设置ip地址。

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：本例子主要强调的是如何手动设置ip。Windows终端输入ipconfig，linux终端输入ifconfig来查询电脑终端。

## 1) 视觉接口使用

```
1 vis = VisionCaptureApi.VisionCaptureApi(TargetIP) \#创建一个视觉传感器实例，这个实例对应的i
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) 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":[1,0,0,0,0,0,0]：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":[0.3,0,0]：相机分布位置。
```

## 3) 飞机控制指令

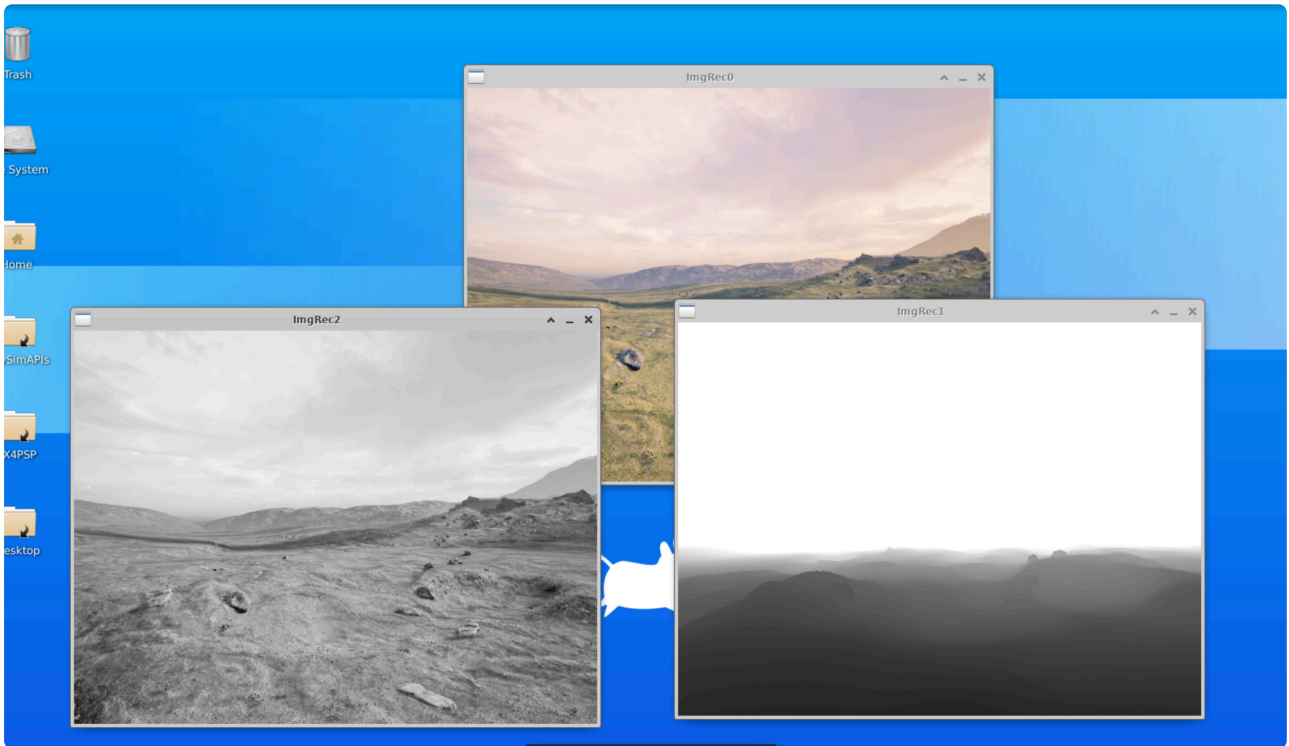
```
1 MavList = MavList+[PX4MavCtrl.PX4MavCtrl(CopterID,TargetIP)] \#初始化并建立i号飞机的MA
2
3 MavList[i].InitMavLoop() \# 初始化Mavlink监听程序，读取第i个飞机数据
4
5 MavList[i].initOffboard() \# 第i个飞机进入Offboard模式
6
7 Error2UE4Map = Error2UE4Map+[-np.array([mav.uavGlobalPos[0]-mav.uavPosNED[0],mav.uavG
8
9 MavList[i].SendPosNED(0, 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
```

## 4) 其余代码说明

```
1 TargetIP = "192.168.31.141" \# 手动修改为电脑主机的IP
2
3 timeInterval = 1/30.0 \# 以30hz的频率进行控制
4
5 lastTime = lastTime + timeInterval \# 设置每一帧的处理结束时间
6
7 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
8
9 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，并生成三个窗口如图。



## 3.文件目录

例程目录：

[\[安装目录\]\RflySimAPIs\8.RflySimVision\0.ApiExps\0.Preparation\5.ManModifyIPRun](#)

文件夹/文件名称	说明
<a href="#">server_ue4.py</a>	取图控制程序（支持Windows或Linux分布式运行）

文件夹/文件名称	说明
Config.json	视觉传感器配置文件
<a href="#">client_ue4_SITL.bat</a>	软件在环仿真一键脚本（MAVLinkFull模式）
Python38Run.bat	Windows下Python程序运行脚本
<a href="#">WinWSL.bat</a>	WSL1/Ubuntu 20.04环境程序运行脚本
<a href="#">WslGUI.bat</a>	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: 手动设置ip

在windows的cmd命令终端输入ipconfig指令，找到本机的ip地址。打开文件夹下的[server\\_ue4.py](#)文件，改为Windows主机的地址。

```
C:\Windows\system32\cmd.e. x + v
Microsoft Windows [版本 10.0.22631.3737]
(c) Microsoft Corporation。保留所有权利。
C:\Users\uavcs>ipconfig

Windows IP 配置

以太网适配器 以太网:

    连接特定的 DNS 后缀 . . . . . :
    IPv6 地址 . . . . . : fd00:6868:6868::37f
    IPv6 地址 . . . . . : fd00:6868:6868:0:a036:9018:9f10:a709
    临时 IPv6 地址 . . . . . : fd00:6868:6868:0:645d:7594:ecca:fb3
    临时 IPv6 地址 . . . . . : fd00:6868:6868:0:9d19:cc2d:9698:b8db
    临时 IPv6 地址 . . . . . : fd00:6868:6868:0:acb5:e923:642f:e1bf
    临时 IPv6 地址 . . . . . : fd00:6868:6868:0:b8d0:24f7:2933:6e29
    本地链接 IPv6 地址 . . . . . : fe80::64e4:bdd7:740e:23dc%17
    IPv4 地址 . . . . . : 192.168.31.139
    子网掩码 . . . . . : 255.255.255.0
    默认网关 . . . . . : fe80::d6da:21ff:fee6:33a1%17
                        192.168.31.1

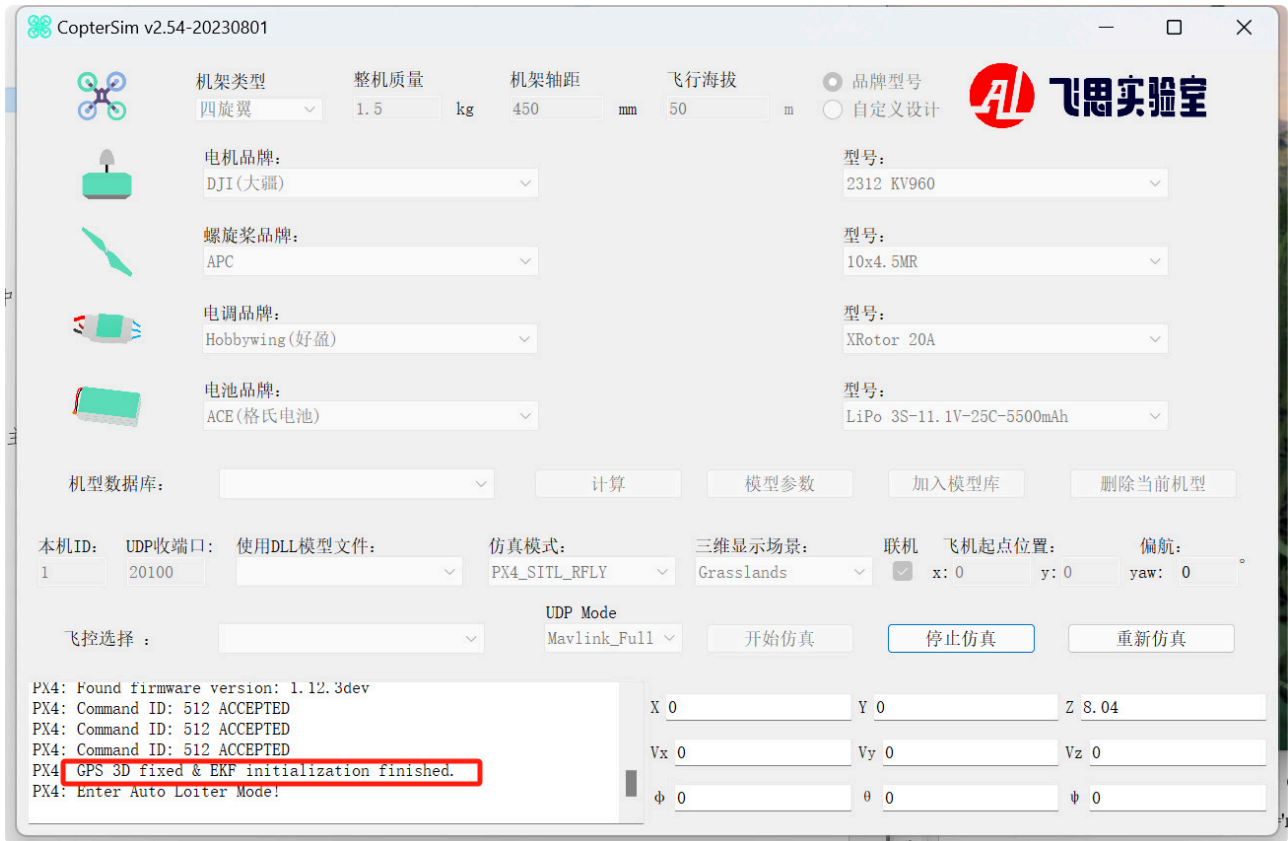
无线局域网适配器 本地连接* 1:

    媒体状态 . . . . . : 媒体已断开连接
    连接特定的 DNS 后缀 . . . . . :
```

```
C:\PX4PSP > RflySimAPIs > 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 = "192.168.31.139"# 手动修改为电脑主机的IP
14
15 # 注意: 如果是本电脑运行的话, 那TargetIP是127.0.0.1的本机地址; 如果是远程访问, 则是192打头的局域网地址。
16 # 因此本程序能同时在本机运行, 也能在其他电脑运行。
17 vis = VisionCaptureApi.VisionCaptureApi(TargetIP)
18
19 # VisionCaptureApi 中的配置函数
20 vis.jsonLoad()
21 isSuss = vis.sendReqToUE4(
22     0, TargetIP
23 )
24 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
25 print('Start Image Receiver')
26 #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.
27 # 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。
28
29 VehilceNum = 1
```

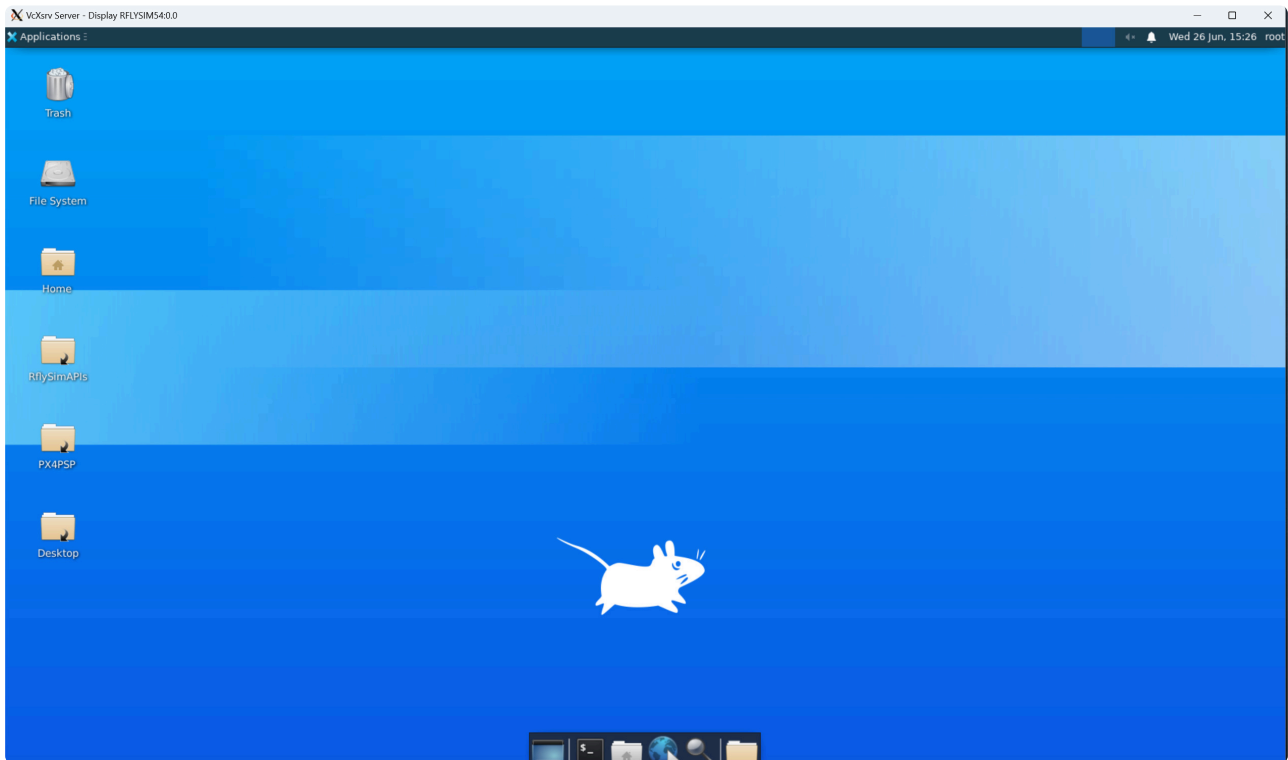
## Step 2: 开启仿真

在Windows主机中，双击运行 `client_ue4_SITL.bat` 运行启动仿真界面，等待CopterSim界面中返回如下语句即可。



### Step 3: 开启WSL可视化界面

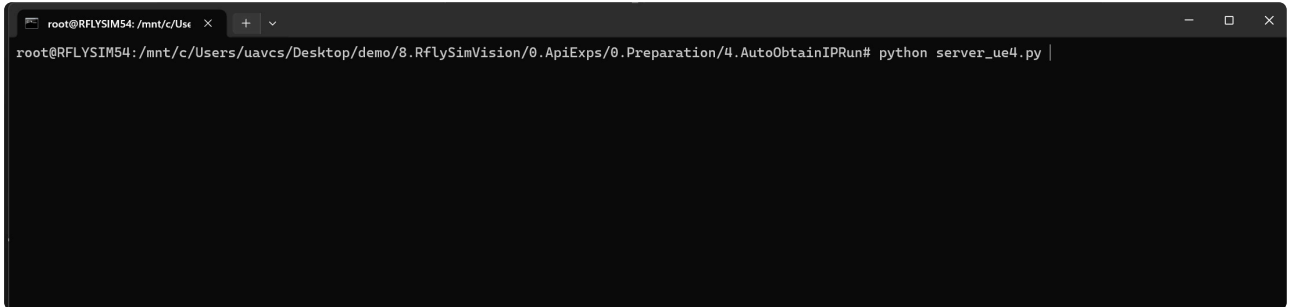
双击打开 `WslGUI.bat`，启动WSL可视化界面。（注：如果打开发现窗口白屏，没有桌面，则关了重开一两次。）



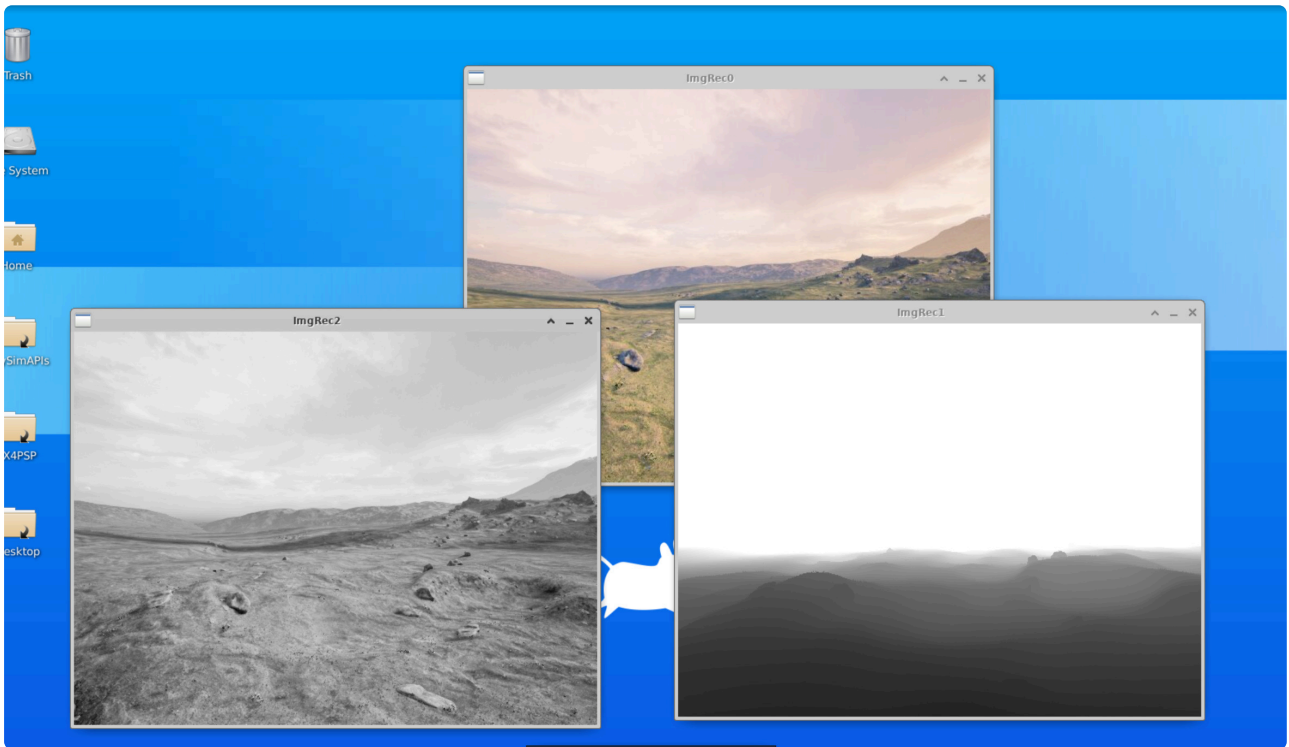
注意：参考 `[安装目录]\RflySimAPIs\1.RflySimIntro\2.AdvExps\e7_WslGUI\Intro.pdf`，来了解WslGUI的功能与使用。

## 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.Auto0btainIPRun# 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.PX4MavCtrl(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

## 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.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](#)

o

扩展实验:

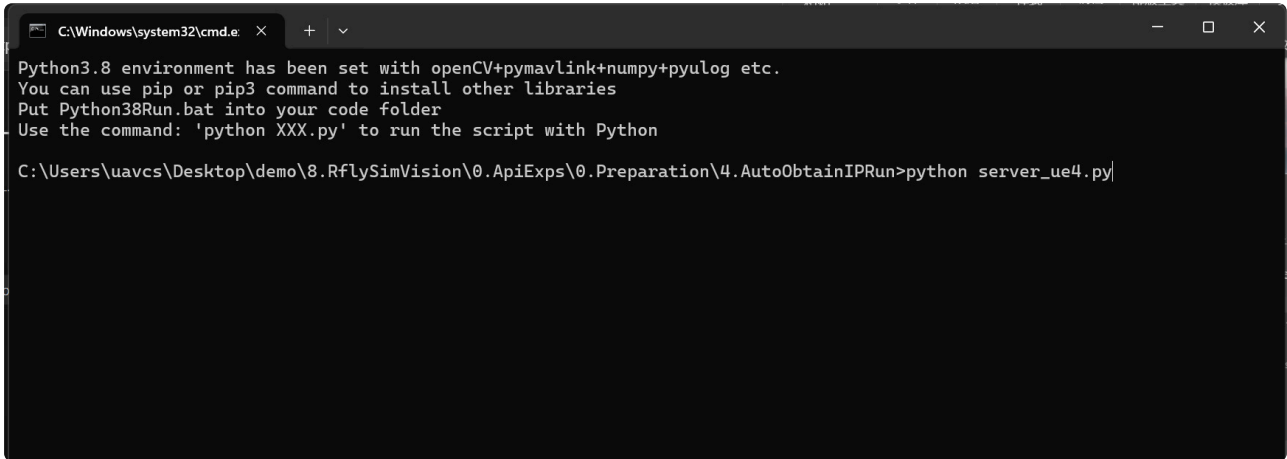
## 5.2.1 本机Windows接收图像实验

### Step 1: 设置ip和开启仿真

步骤1同6、实验步骤的Step1和Step2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\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: 设置ip和开启仿真

步骤1同6、实验步骤的Step1和Step2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: 查看ip

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：

```
uav11@ubuntu: ~/桌面
uav11@ubuntu:~/桌面$ ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.31.7 netmask 255.255.255.0 broadcast 192.168.31.255
    inet6 fe80::e18c:2b99:f1fd:f232 prefixlen 64 scopeid 0x20<link>
    inet6 fd00:6868:6868:0:1d6:4d0c:bcc7:b533 prefixlen 64 scopeid 0x0<glo
bal>
    inet6 fd00:6868:6868::596 prefixlen 128 scopeid 0x0<global>
    inet6 fd00:6868:6868:0:35bf:e91:25e1:184e prefixlen 64 scopeid 0x0<glo
bal>
    ether 00:0c:29:38:07:6e txqueuelen 1000 (以太网)
    RX packets 4872 bytes 897759 (897.7 KB)
    RX errors 0 dropped 49 overruns 0 frame 0
    TX packets 479 bytes 46548 (46.5 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (本地环回)
    RX packets 199 bytes 18029 (18.0 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 199 bytes 18029 (18.0 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

## Step 2: 修改ip

修改 `client_ue4_SITL.bat` 文件中的 `IS_BROADCAST` 的值为Linux主机的IP地址。

```
client_ue4_SITL.bat
文件 编辑 查看
REM Set use DLL model name or not, use number index or name string
REM This option is useful for simulation with other types of vehicles instead of multicopters
set DLLModel=0

REM Set the simulation mode on CopterSim, use number index or name string
REM e.g., SimMode=2 equals to SimMode=PX4 SITL RFLY
set SimMode=2

REM Set the vehicle-model (airframe) of PX4 SITL simulation, the default airframe is a quadcopter: iris
REM Check folder Firmware\ROMFS\px4fmu common\init.d-posix for supported airframes (Note: You can also create your airframe file here)
REM E.g., fixed-wing aircraft: PX4SITLFrame=plane; small cars: PX4SITLFrame=rover
set PX4SITLFrame=iris

REM Set the map, use index or name of the map on CopterSim
REM e.g., UE4 MAP=1 equals to UE4 MAP=Grasslands
SET UE4 MAP=Grasslands

REM Set the origin x,y position (m) and yaw angle (degree) at the map
SET /a ORIGIN POS X=0
SET /a ORIGIN POS Y=0
SET /a ORIGIN YAW=0

REM Set the interval between two vehicle, unit:m
SET /a VEHICLE INTERVAL=2

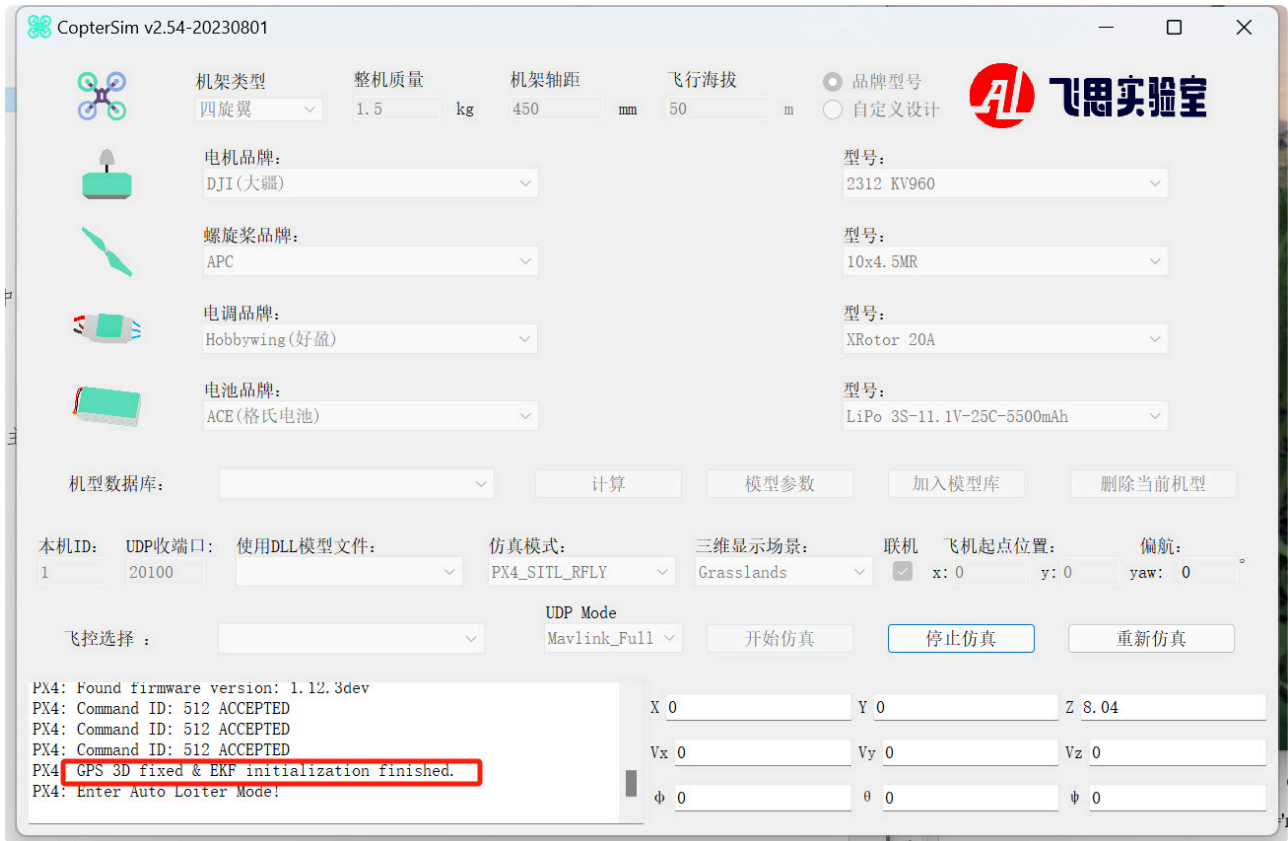
REM Set broadcast to other computer; 0: only this computer, 1: broadcast; or use IP address to increase speed
REM e.g., IS BROADCAST=0 equals to IS BROADCAST=127.0.0.1, IS BROADCAST=1 equals to IS BROADCAST=255.255.255.255
SET IS BROADCAST=192.168.31.7

REM Set UDP data mode; 0: UDP FULL, 1:UDP Simple, 2: Mavlink Full, 3: Mavlink simple. input number or string
REM e.g., UDPSIMMODE=1 equals to UDPSIMMODE=UDP Simple
SET UDPSIMMODE=2

REM Set the path of the RflySim tools
SET PSP PATH=C:\PX4PSP
SET PSP PATH LINUX=/mnt/c/PX4PSP
C:
```

### Step 3: 开启仿真

在Windows主机中，双击运行 `client_ue4_SITL.bat` 运行启动仿真界面，等待CopterSim界面中返回如下语句即可。



## Step 4: 修改ip

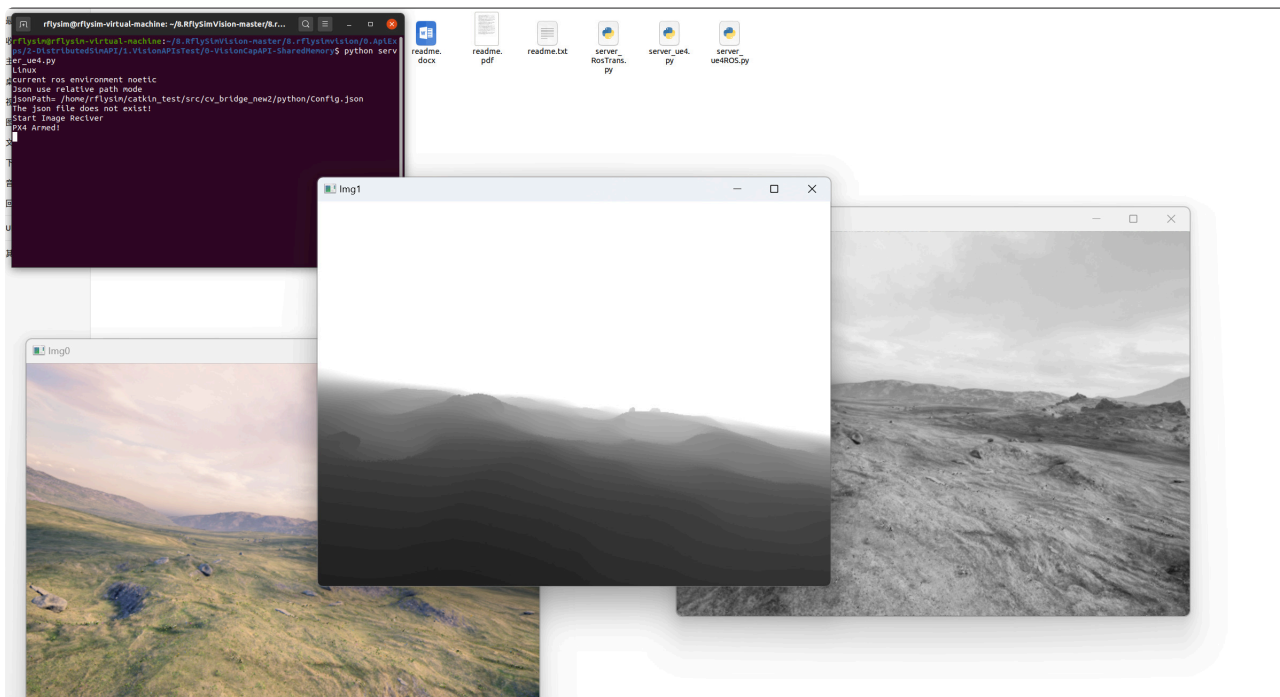
将整个文件夹复制到Linux系统中，打开 [server\\_ue4.py](#) 文件，改为Windows主机的地址。

```
server_ue4.py
~/桌面/5.Manually modify the IP address
保存(S)

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
```

## Step 5: 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。



## | 6.参考资料

1. 无

## | 7.常见问题

Q1: \*\*\*

A1: \*\*\*