

# 1. 实验名称及目的

## 1.1 实验名称

大疆 Livox 激光雷达点云数据UDP直传模式实验

## 1.2 实验目的

通过平台接口python发送取图请求给RflySim3D,后者通过大疆 Livox 激光雷达扫描直接通过UDP直传10hz频率传出点云数据。

## 1.3 关键知识点

### 1.3.1 Windows版本本地回环地址通信测试

首先运行 [LidarAPIDemo.bat](#) 开启一个飞机的软件在环仿真，然后运行 [LidarAPIDemo.py](#) 文件

然后向RflySim3D发送取图请求，通过共享内存的方式得到点云数据，通过死循环将不断得到的点云数据输入到自定义的绘制点云图函数从而绘制点云图，使得点云图一直显示在屏幕上。

### 1.3.2 Windows运行客户端与Ubuntu远程通信

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,启动Rviz, 添加话题，使得Rviz能够获取到 [server\\_ue4.py](#) 发布的话题消息，同时添加lidar.rviz配置文件，使得Rviz生成自定义tf坐标的结果。

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。通过平台接口python发送取图请求给RflySim3D,后者通过Python接口Open3DShow.py使用（见RflySimAPIs\RflySimSDK\vision目录）采用UDP直传10hz频率传出点云数据。关键代码解析如下：

本例子和其他分布式例子的区别，主要在于直接通过UDP直传传出点云数据。

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 VisionCaptureApi.isEnableRosTrans = True \# 启用ROS发布模式
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.sendImuReqCopterSim(StartCopterID, TargetIP) \# 发送imu数据
10
11 vis.hasData[i] \# 图片i数据是否更新
12
13 vis.Img[i] \# 图片i数据 (像素矩阵)
14
15 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号程序所在电脑的IP, 作为目标IP. 这里获取Cop
4
5 vis = VisionCaptureApi.VisionCaptureApi(TargetIP) \#创建一个视觉传感器实例, 这个实例对应的ip号为TargetIP
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的视觉传感器
2
3 "TypeID":22 : 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。
```

## 4) Open3DShow接口使用

```
1 show3d=Open3DShow.Open3DShow() \# 创建3D点云显示实例
2
3 show3d.CreatShow(0) \# 创建点云显示窗口
4
5 show3d.UpdateShow(vis.Img[0]) \# 更新点云
```

## 5) 飞机控制指令

```
1 | MavList = MavList+[PX4MavCtrl.PX4MavCtrler(CopterID,TargetIP)] \#初始化并建立i号飞机的MAVLink通信连接, 连接上远端的电
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.uavGlobalPos[1]-mav.uavPosNED
8 |
9 | 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
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
```

## 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.sendUE4PosScale(1, 3, 0, PosE=[2, 0, -3], AngEuler=[0, 0, -1.57]) \#创建一个障碍物, id为1, 类型为2, 速度为0, 位置为
```

## 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\7.LidarLivoxDemo](#)

文件夹/文件名称	说明
<a href="#">LidarAPIDemo.bat</a>	启动仿真配置文件
<a href="#">LidarAPIDemo.py</a>	Python实验脚本
<a href="#">server_ue4.py</a>	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
Livox_theory.pdf	Livox 激光雷达原理详解
<a href="#">Python38Run.bat</a>	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://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

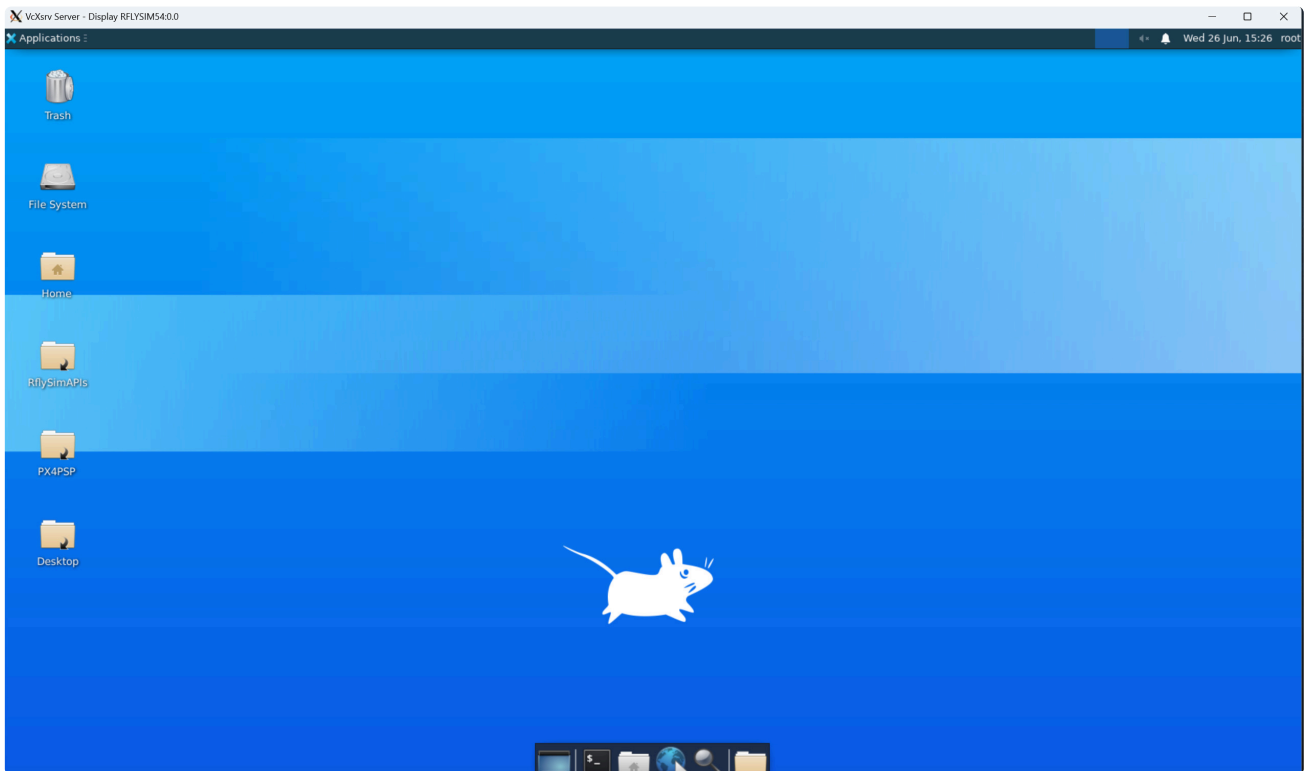
### 5.1 必做实验：WinsWSL控制

#### Step 1：开启仿真

双击运行 `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。

## 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。

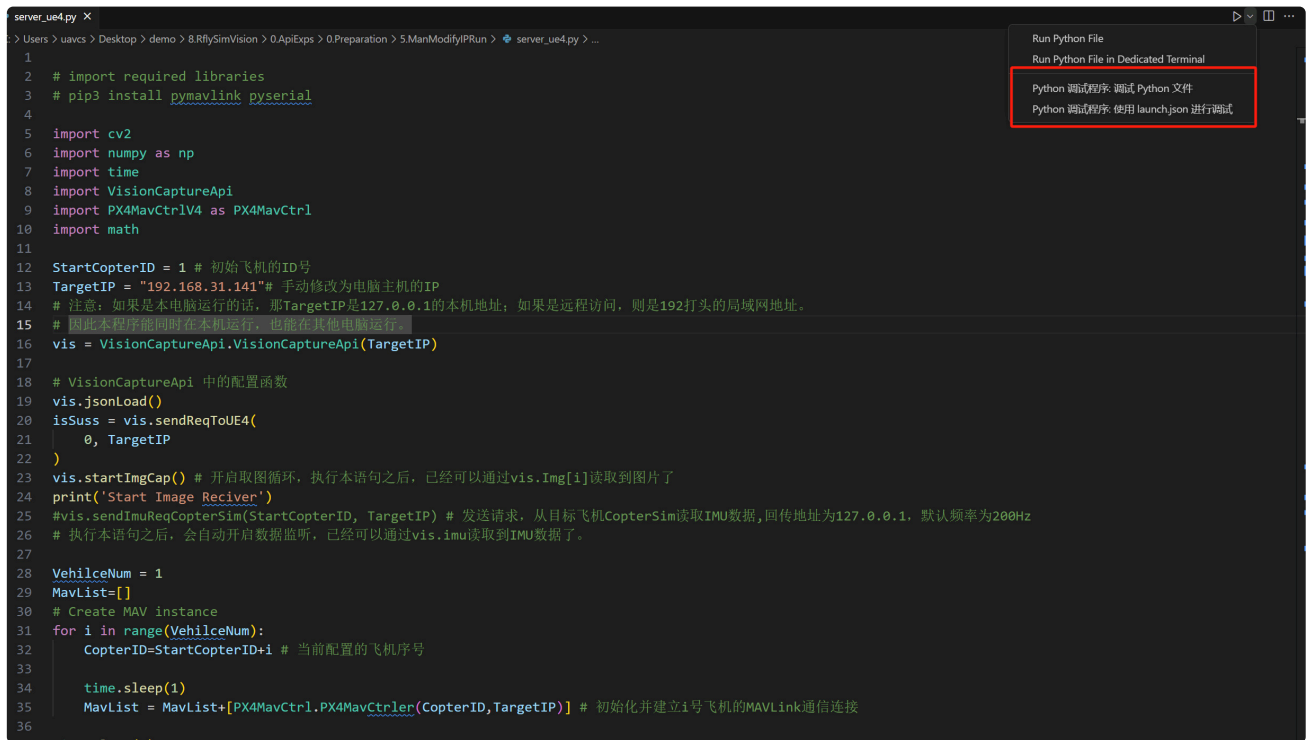
## Step 3: 运行控制程序

在文件夹路径的终端下，双击打开 `WinWSL.bat`，运行 `roscore` 指令，

双击打开 `WinWSL.bat`，运行命令 `python3 server_ue4.py` 运行脚本 `server_ue4.py`。

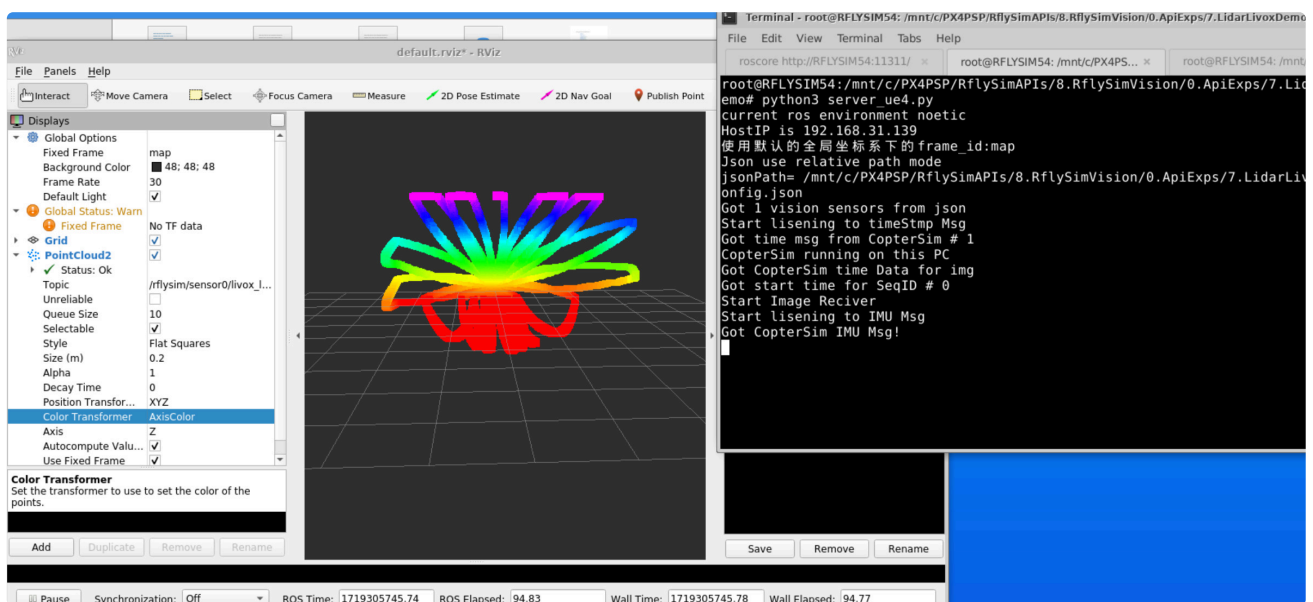
```
Terminal - root@RFLYSIM54: /mnt/c/PX4PSP/RflySimAPIs/8.RflySimVision/0.ApiExps/7.LidarLivoxDemo
File Edit View Terminal Tabs Help
roscore http://RFLYSIM54:11311/ x root@RFLYSIM54: /mnt/c/PX4PS... x root@RFLYSIM54: /mnt/c/PX4PS... x
root@RFLYSIM54: /mnt/c/PX4PSP/RflySimAPIs/8.RflySimVision/0.ApiExps/7.LidarLivoxDemo# python3 server_ue4.py
current ros environment noetic
HostIP is 192.168.31.139
使用默认的全局坐标系下的 frame_id:map
Json use relative path mode
jsonPath= /mnt/c/PX4PSP/RflySimAPIs/8.RflySimVision/0.ApiExps/7.LidarLivoxDemo/C
onfig.json
Got 1 vision sensors from json
Start lisening to timeStmp Msg
Got time msg from CopterSim # 1
CopterSim running on this PC
Got CopterSim time Data for img
Got start time for SeqID # 0
Start Image Reciver
Start lisening to IMU Msg
Got CopterSim IMU Msg!
```

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的本机地址；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[1]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 VehicNum = 1
29 MavList=[]
30 # Create MAV instance
31 for i in range(VehicNum):
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
```

然后双击打开 WinWSL.bat，运行rviz命令，打开rviz可视化工具，点击Add键添加PointCloud2，然后在Topic选择话题 /rflsim/sensor0/livox\_lidar 数据，然后选择文件夹下的lidar.rviz配置文件可看到如下效果：



## Step 4: 结束仿真

在下图 “ LidarAPIDemo.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。

## 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.GenenralUbuntuConfig\\Readme.pdf

；NX板卡的配置方法，先看

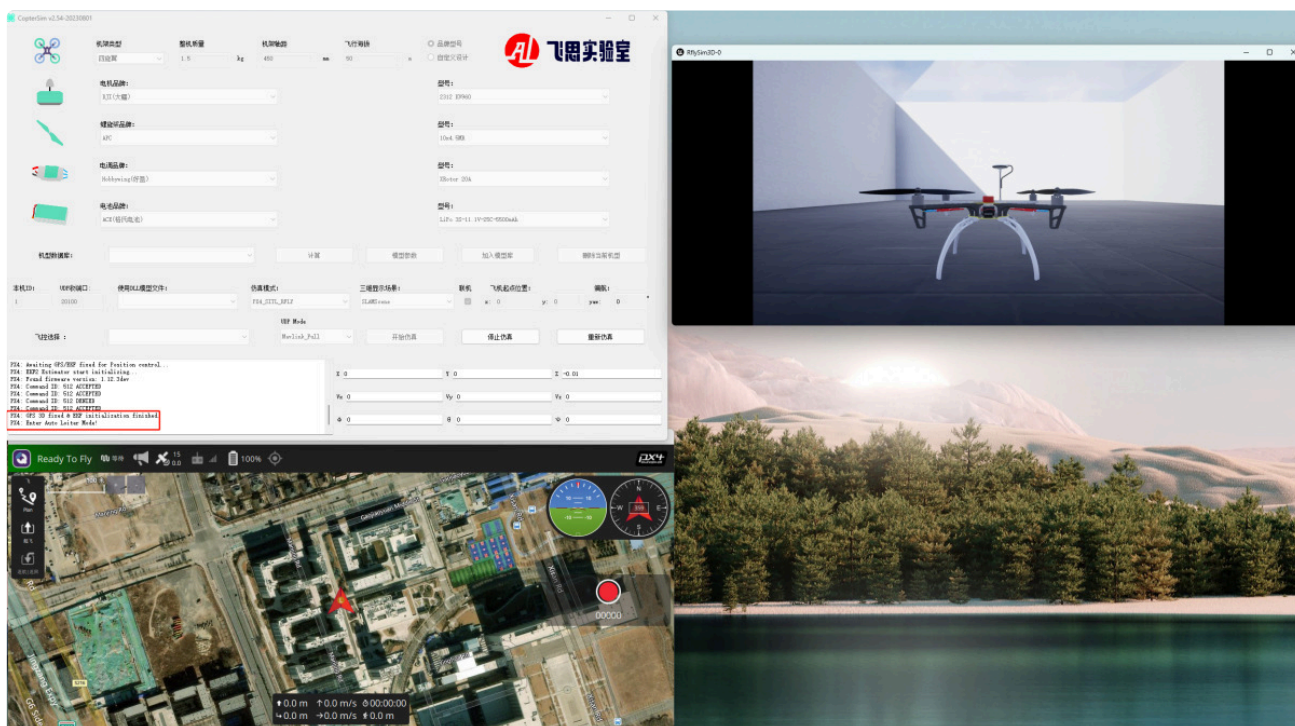
[安装目录]\\RflySimAPIs\\8.RflySimVision\\0.ApiExps\\0.Preparation\\3.NXwithPX4Config\\Readme.pdf。

扩展实验：

## 5.2.1 本机Windows接收图像实验

### Step 1：开启仿真

双击运行 `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。



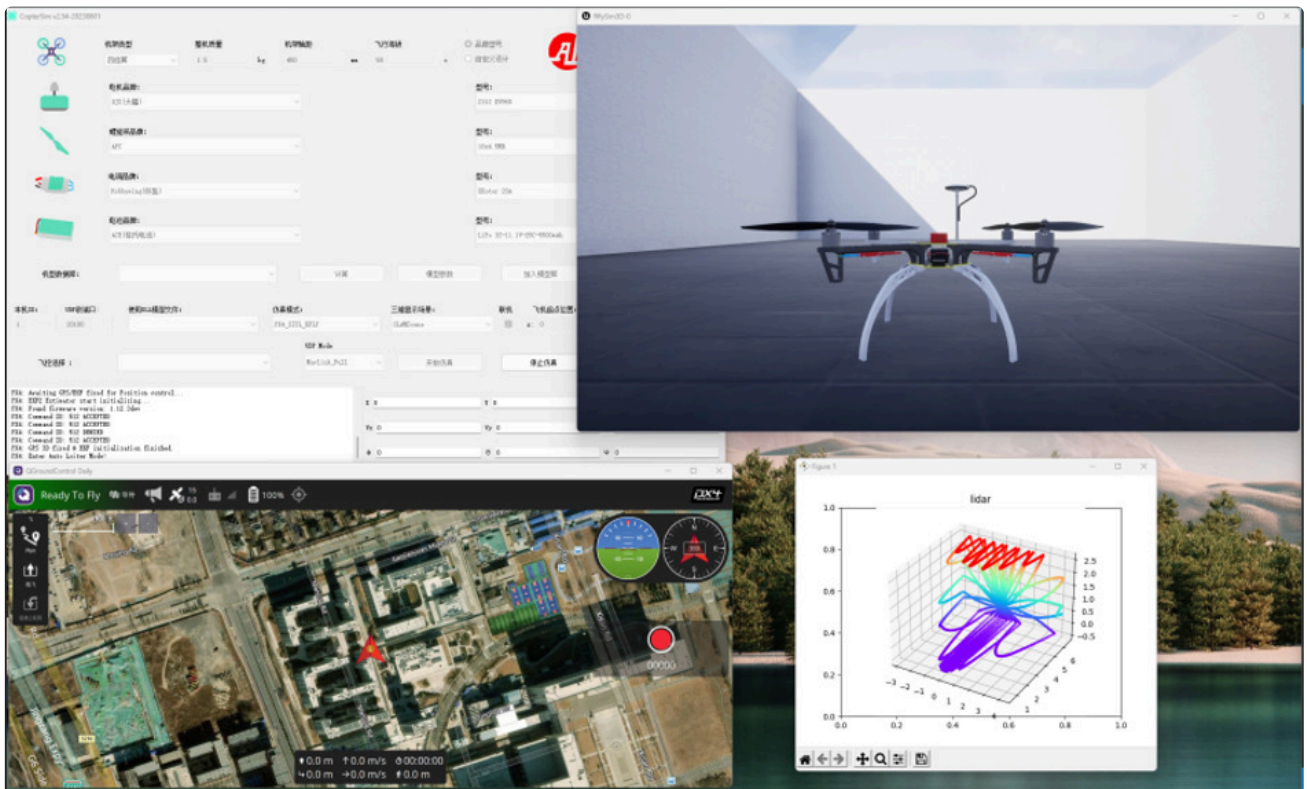
### Step 2：运行控制程序

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 `LidarAPIDemo.py` 文件，输入 `python LidarAPIDemo.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\7.LidarLivoxDemo>python LidarAPIDemo.py
```

注：也可以使用VSCode打开运行（如果安装好了VSCode），执行如下：

运行 LidarAPIDemo.py 文件，可以看到一个三维点云窗口界面以及终端出现的提示。



## 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

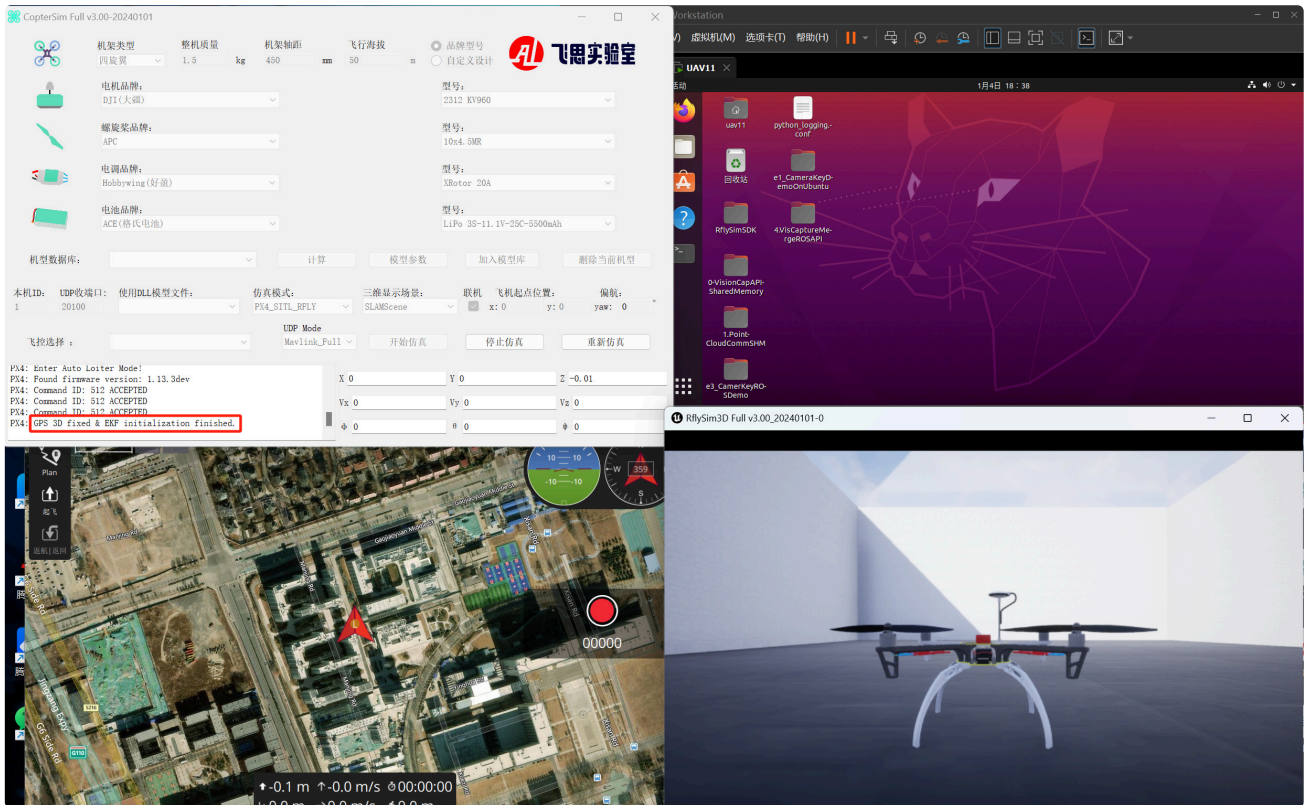
在另一台Windows电脑上双击 Python38Run.bat，输入 `python LidarAPIDemo.py` 再运行 LidarAPIDemo.py。

## 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。

并且启动一个已安装了ros的Ubuntu虚拟机。



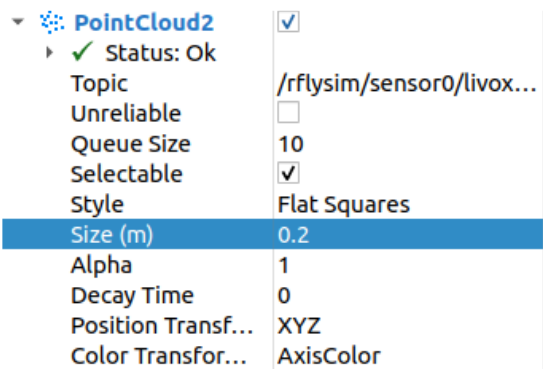
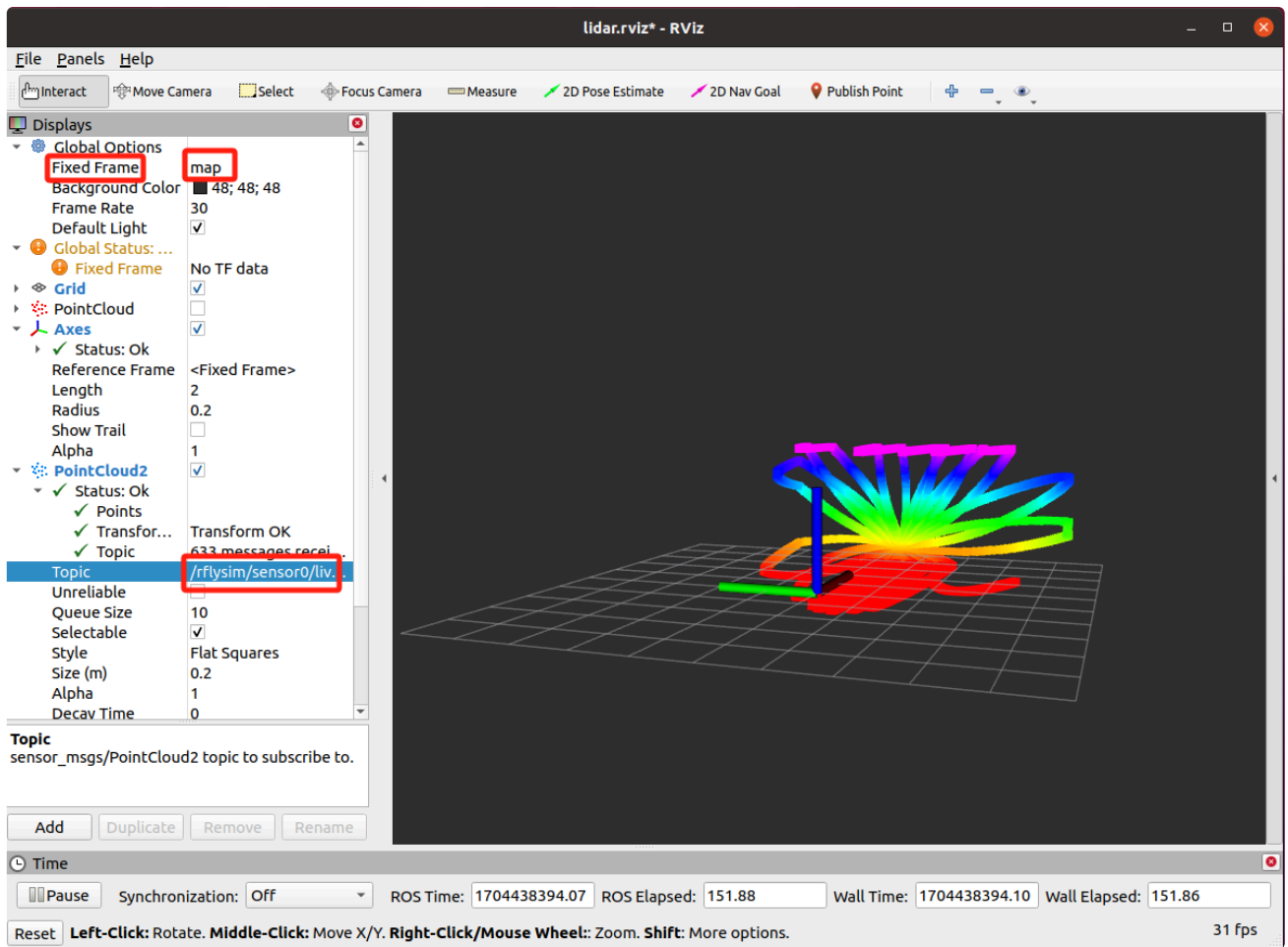
## Step 2: 运行控制程序

在虚拟机中新建一个文件夹，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，点击Add键添加PointCloud2，然后在Topic选择话题/rflysim/sensor0/livox\_lidar数据，然后选择文件夹下的lidar.rviz配置文件可看到如下效果：

```

uav11@ubuntu: ~/桌面/2.UDPDirect10Hz
uav11@ubuntu:~/桌面/2.UDPDirect10Hz$ python3 server_ue4.py
current ros environment noetic
使用默认的全局坐标系下的frame_id:map
Json use relative path mode
jsonPath= /home/uav11/桌面/2.UDPDirect10Hz/Config.json
Got 1 vision sensors from json
Start lisening to timeStmp Msg
Got time msg from CopterSim # 1
CopterSim not on this PC
1704437624.369
1704437623.9894388
Got CopterSim time Data for img
Got start time for SeqID # 0
Start Image Reciver
Start lisening to IMU Msg
Got CopterSim IMU Msg!
Error to listen to IMU Msg!
PX4 Armed!
PX4 DisArmed!

```



## 6.参考资料

无

## 7.常见问题

Q1: 无

A1: 无