

1. 实验名称及目的

1.1 实验名称

UDP激光雷达坐标系点云数据传输实验

1.2 实验目的

在虚拟机中运行程序发送取图请求，通过UDP直传方式接收处理点云数据

1.3 关键知识点

本实验主要是实现通过Python接口VisionCaptureApi.py使用（见RflySimAPIs\RflySimSDK\vision目录）发送取图请求。通过PX4MavCtrlr.py（见RflySimAPIs\RflySimSDK\ctrl目录）使用创建控制接口，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。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 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）
```

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":20 : 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 mav = PX4MavCtrl.PX4MavCtrler(1) \# 创建飞机控制实例
2
3 mav.InitMavLoop() \# 初始化Mavlink监听程序，读取飞机数据
4
5 mav.InitTrueDataLoop() \# 初始化来自 CopterSim 的 UDP True 数据监听循环
6
7 mav.initOffboard() \# 进入Offboard模式
8
9 mav.SendMavArm(True) \# 解锁飞控
10
11 mav.SendPosNED(0, 0, -8, 0) \# 发送8米高的位置控制指令
12
13 mav.SendCopterSpeed(3) \# 发送速度为3m/s控制指令
14
15 mav.endOffboard() \# 结束offboard模式
16
17 mav.stopRun() \# 结束mav飞机实例运行
```

6) 其余代码说明

t

```
1 imeInterval = 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

本实验通过平台python接口UDP直传模式取图并转化获取为激光雷达坐标系点云数据动态显示。

3. 文件目录

例程目录：[\[安装目录\]\RflySimAPIs\8.RflySimVision\0.ApiExps\8.LidarAPIDemo\2.UDPDirectClientServer](#)

| 文件夹/文件名称 | 说明 |
|-------------------------------------|------------------------------|
| client_ue4_SITL.bat | Windows客户端启动仿真配置文件 |
| server_ue4.py | 取图控制程序（支持Windows或Linux分布式运行） |
| server_ue4ROS.py | Linux服务端点云数据处理发布程序 |
| Config.json | 视觉传感器配置文件 |
| lidar.rviz | Rviz配置文件 |
| 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。

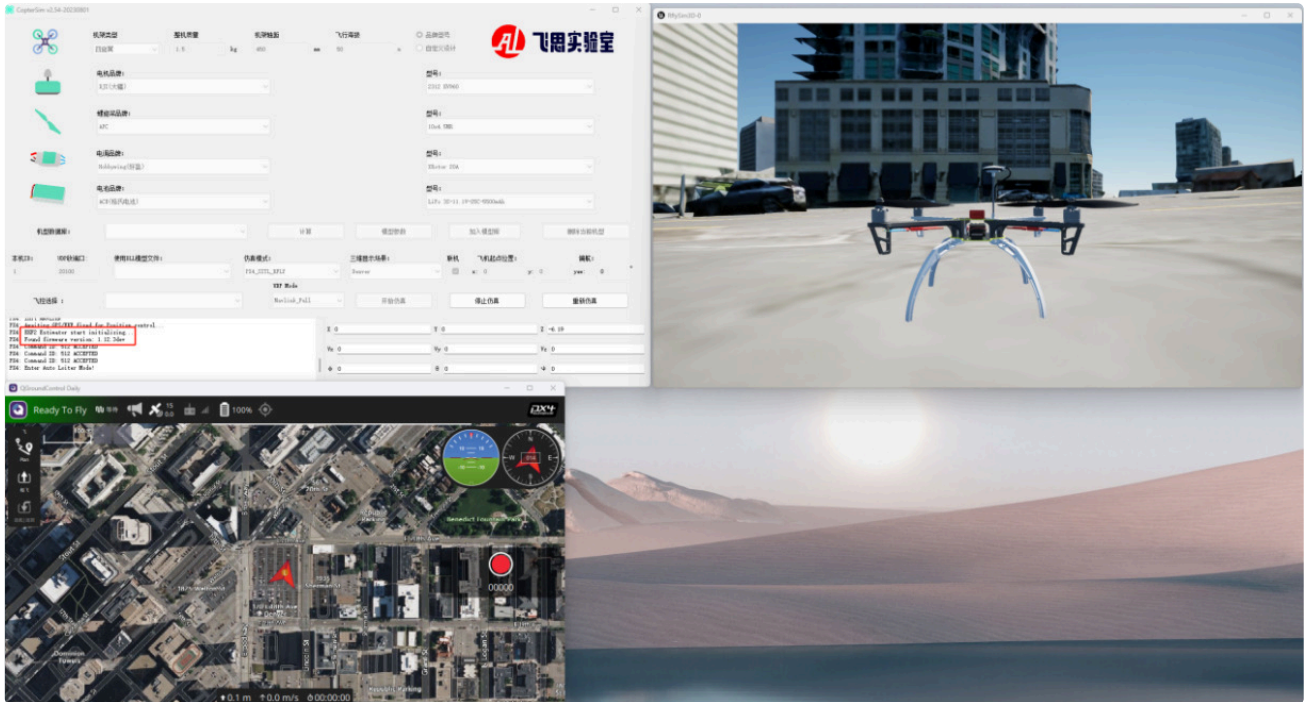
①：推荐配置请见：<https://rflysim.com/doc/zh/HowToInstall.pdf>

5. 实验步骤

5.1 必做实验：WinsWSL控制

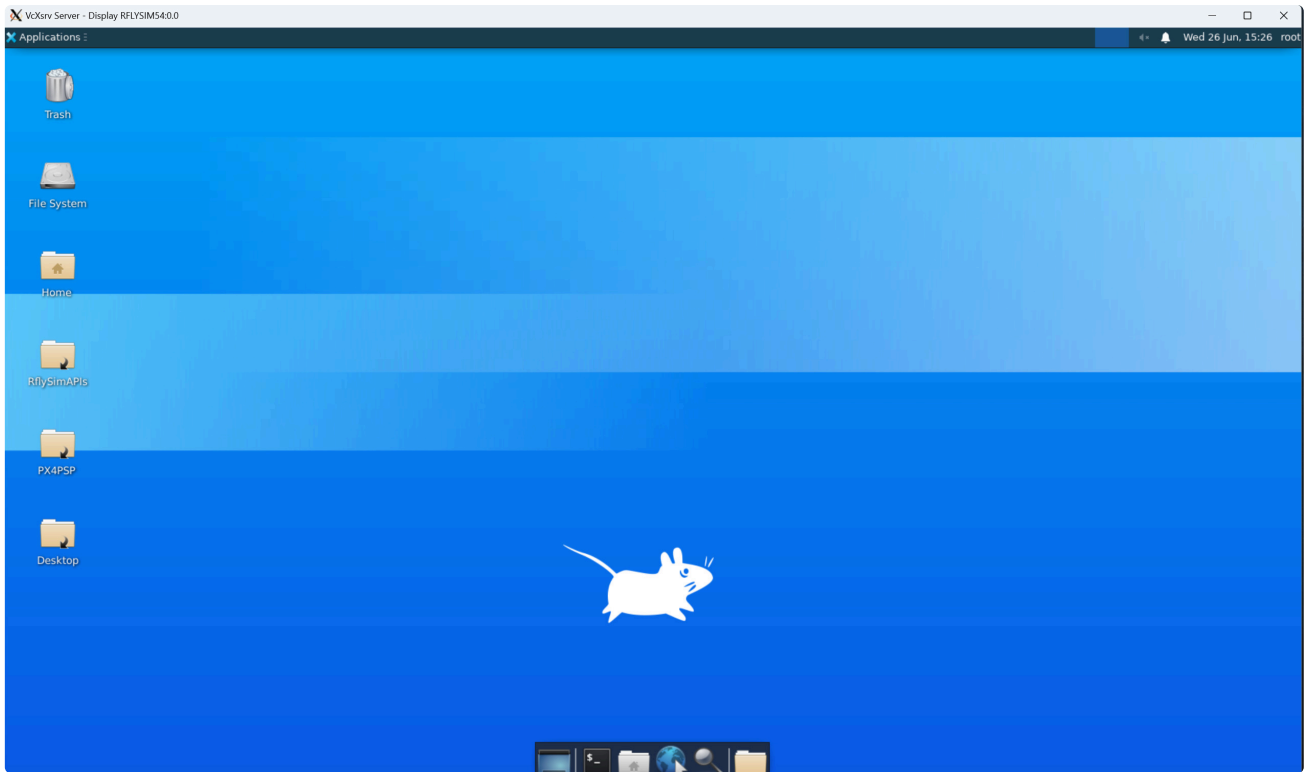
Step 1：开启仿真

双击运行 `client_ue4_SITL.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。）



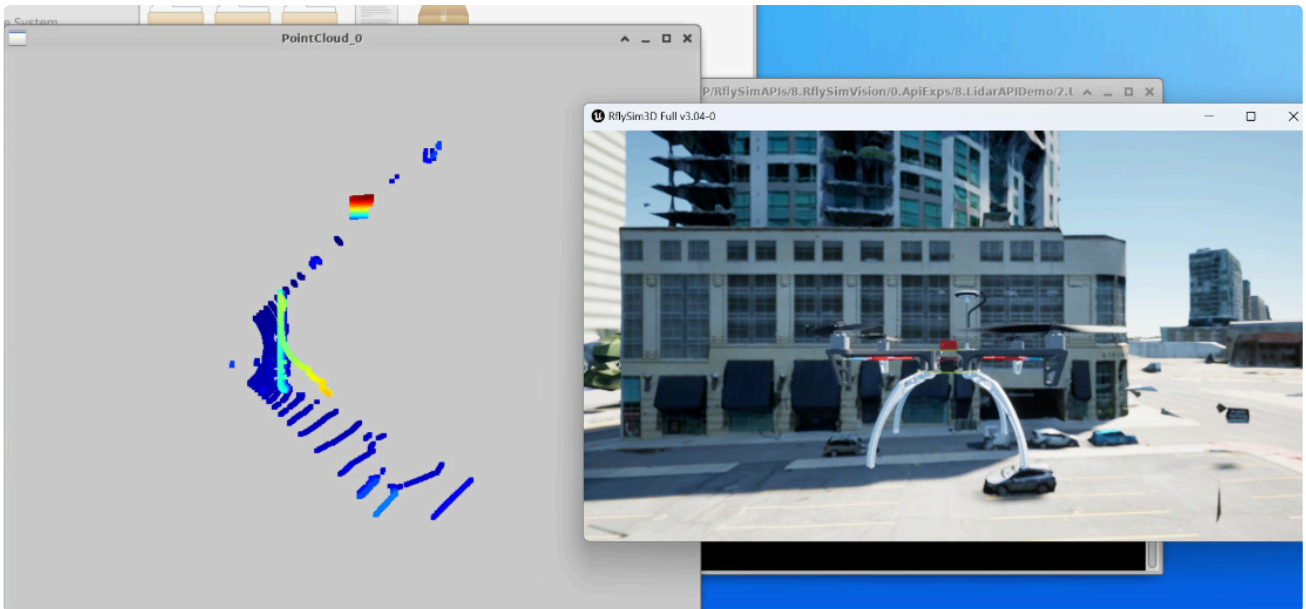
注意：参考 `[安装目录]\\RflySimAPIs\\1.RflySimIntro\\2.AdvExps\\e7_WslGUI\\Intro.pdf`，来了解WslGUI的功能与使用。

Step 3: 运行控制程序

在文件夹路径下，双击打开 `WinWSL.bat`，输入命令 `roscore` 启动rosmaster。

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

```
Terminal - root@RFLYSIM54: /mnt/c/PX4PSP/RflySimAPIs/8.RflySimVision/0.ApiExps/8.LidarAPIDemo/2.U...
File Edit View Terminal Tabs Help
root@RFLYSIM54: /mnt/c/PX4P... x root@RFLYSIM54: /mnt/c/PX4PS... x root@RFLYSIM54: /mnt/c/PX4PS... x
root@RFLYSIM54:/mnt/c/PX4PSP/RflySimAPIs/8.RflySimVision/0.ApiExps/8.LidarAPIDemo/2.U...
o/2.UDPDirectClientServer# python3 server_ue4.py
current ros environment noetic
HostIP is 192.168.31.139
Json use relative path mode
jsonPath= /mnt/c/PX4PSP/RflySimAPIs/8.RflySimVision/0.ApiExps/8.LidarAPIDemo/2.U...
DPDClientServer/Config.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
^CTraceback (most recent call last):
  File "server_ue4.py", line 82, in <module>
```



备注：可以参考 [安装目录]\RflySimAPIs\1.RflySimIntro\2.AdvExps\8.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[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
37

```

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.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：

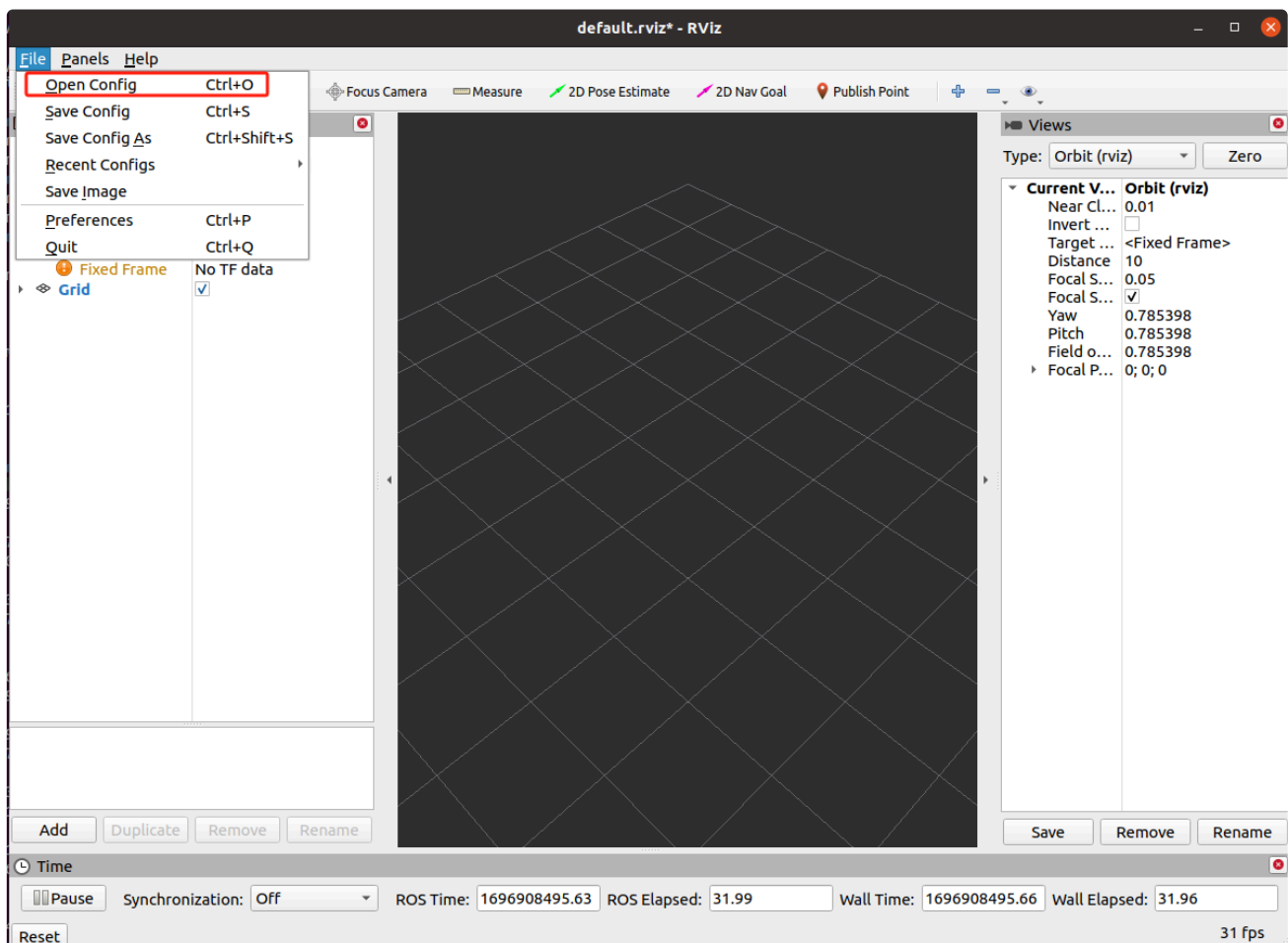
5.2.1在虚拟机/视觉板卡/另一台Ubuntu上接收图像实验

Step 1：开启仿真

步骤1同上面的Step1步骤。

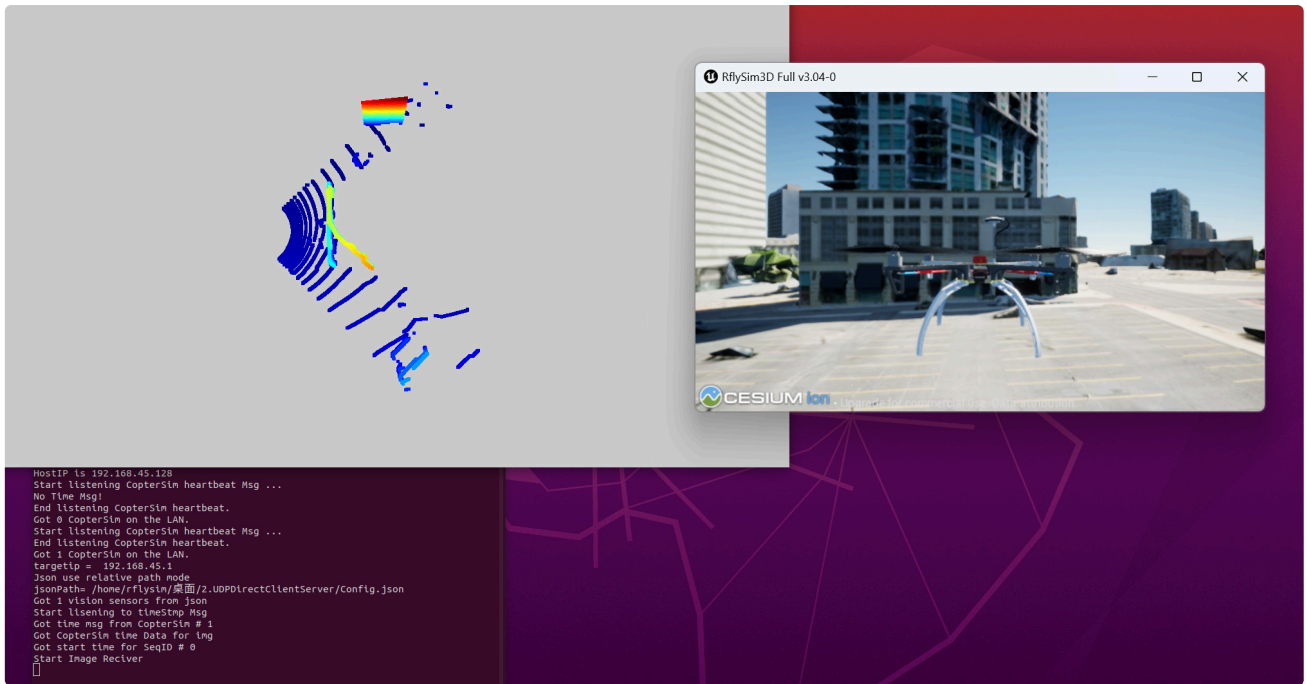
Step 2：配置rviz显示

将实验文件夹全部拷贝到虚拟机中，然后分别打开两个新的终端，输入命令 `roscore` 启动rosmaster,在另一个终端输入rviz命令启动Rviz工具，然后在Rviz工具中加载文件夹中的lidar.rviz配置文件。



Step 3：运行控制程序

在实验文件夹下打开终端输入 `python3 server_ue4.py` 命令，启动程序进行图片解析，以及激光点云数据的发布。



6.参考资料

无

7.常见问题

Q1: 运行 `server_ue4.py` 时报错 `ValueError: Unknown projection '3d'`

A1: 这是因为matplotlib库的版本的版本的原因,可通过添加如下语句解决

```
from mpl_toolkits.mplot3d import Axes3D
```