

# 1. 实验名称及目的

## 1.1 实验名称

基于Python的Linux系统图像接收与ROS发布实验

## 1.2 实验目的

利用Python接口在Linux远端系统中请求RflySim3D发送指定传感器数据，并接收图像和点云数据，转发到ROS空间，并使用Rviz进行传感器数据的可视化观测。

## 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。通过PX4MavCtrler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。关键代码解析如下：

本例子和其他分布式例子的区别，主要在于通过平台接口自定义更改 frame\_id 接口。

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.sendImuReqCopterSim(StartCopterID, TargetIP) \# 发送imu数据
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) 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) 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 数组, 便于进行计算处理
9
10 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
11
12 mav=MavList[j] \# 从列表中取第i个实例作为当下处理的飞机实例
13
14 MavList[i].endOffboard() \# 对第i个飞机进入endoffboard模式
15
16 MavList[i].stopRun() \# 终止第i个飞机
```

## 2. 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。

## 3. 文件目录

例程目录：

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

文件夹/文件名称	说明
<a href="#">RunRflysim3DSITL.bat</a>	启动仿真配置文件
<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
vision.rviz	Rviz配置文件
<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>

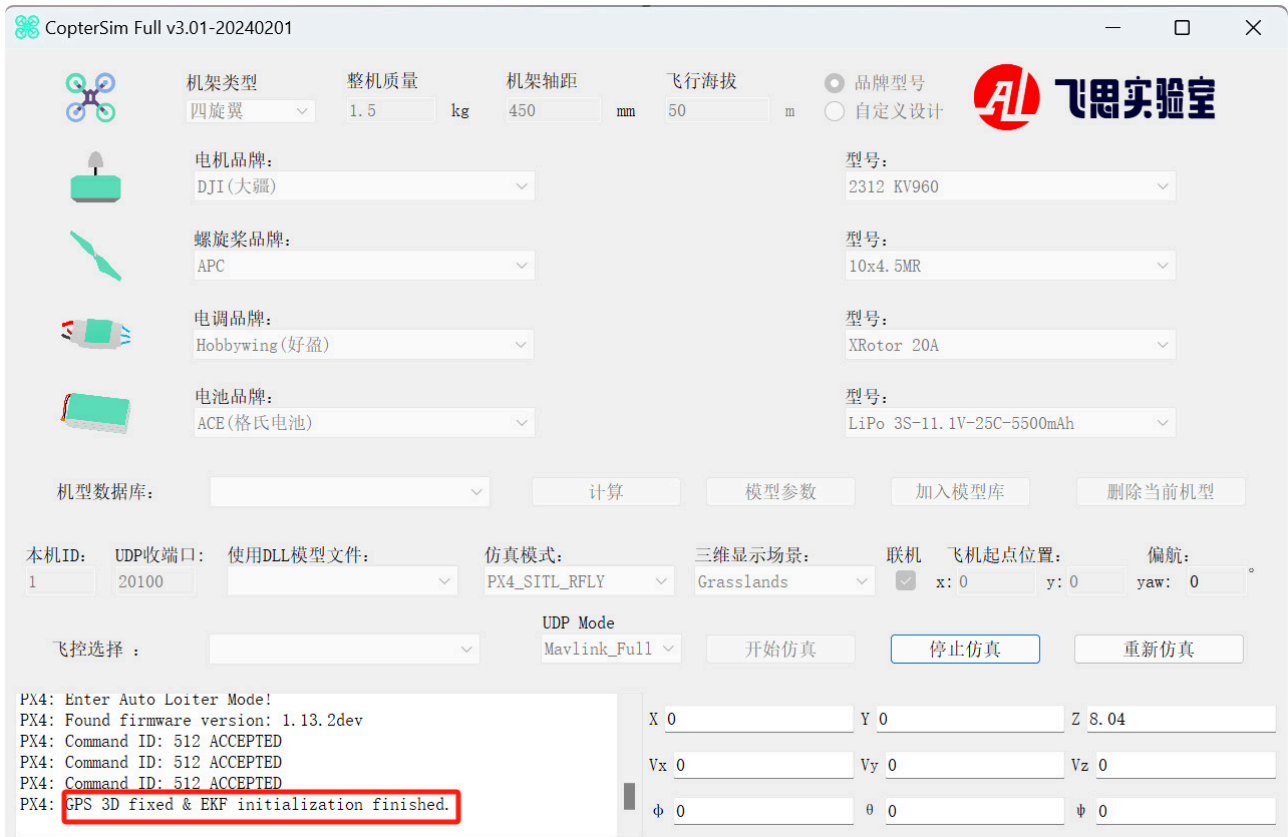
## 5. 实验步骤

### 5.1 必做实验：WinsWSL控制

#### Step 1：开启仿真

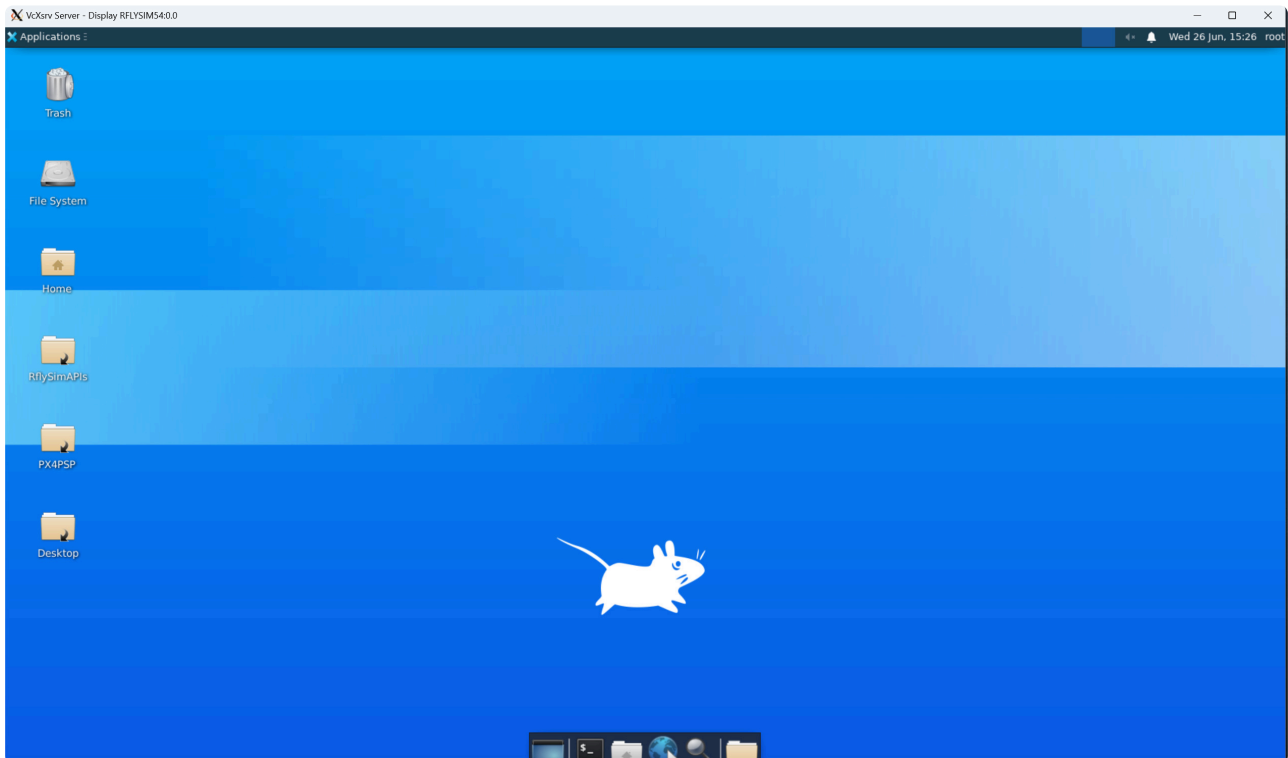
双击运行RunRflysim3DSITL.bat开启一个飞机的软件在环仿真。将会启动1个QGC地面站，1个CopterSim软件且其软件下侧日志栏必须打印出GPS 3D fixed & EKF initialization finished字样代表初始化完成，并且RflySim3D软件内有1架无人机。

并且启动一个已安装了ros的Ubuntu虚拟机。



## Step 2: 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。）

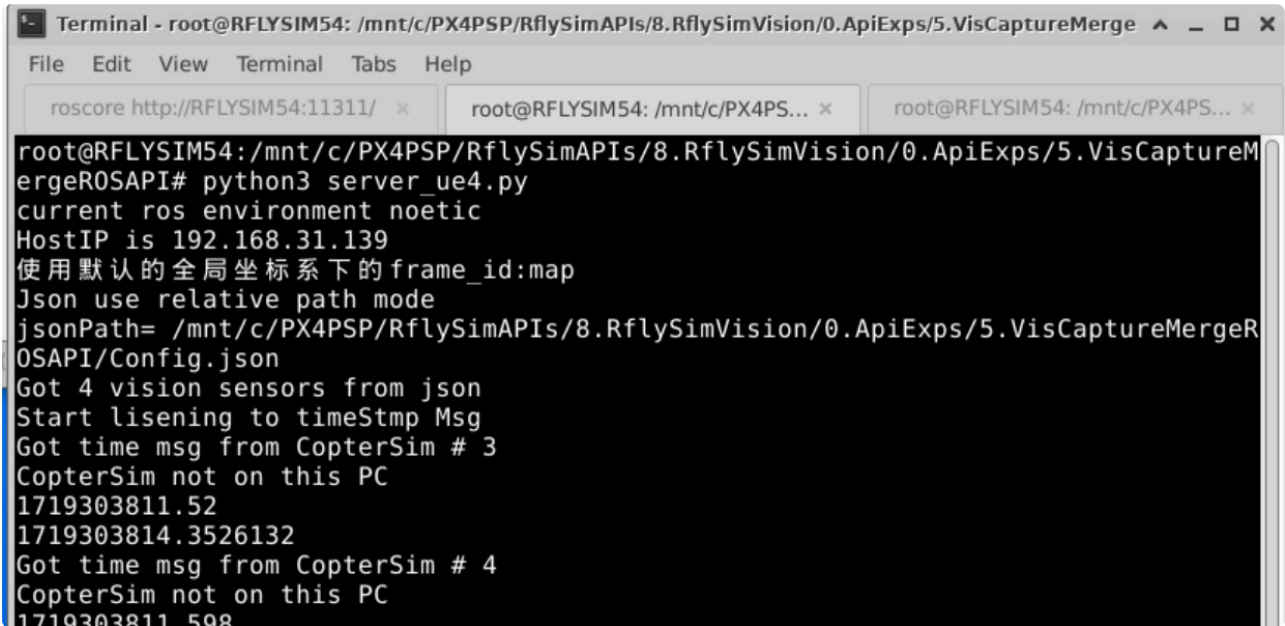


注意：参考[\[安装目录\]\RflySimAPIs\1.RflySimIntro\2.AdvExps\e7\\_WslGUI\Intro.pdf](#)，了解WslGUI的功能与使用。

## Step 3: 运行控制程序

双击打开[WinWSL.bat](#)，在终端中运行 `roscore` 指令。

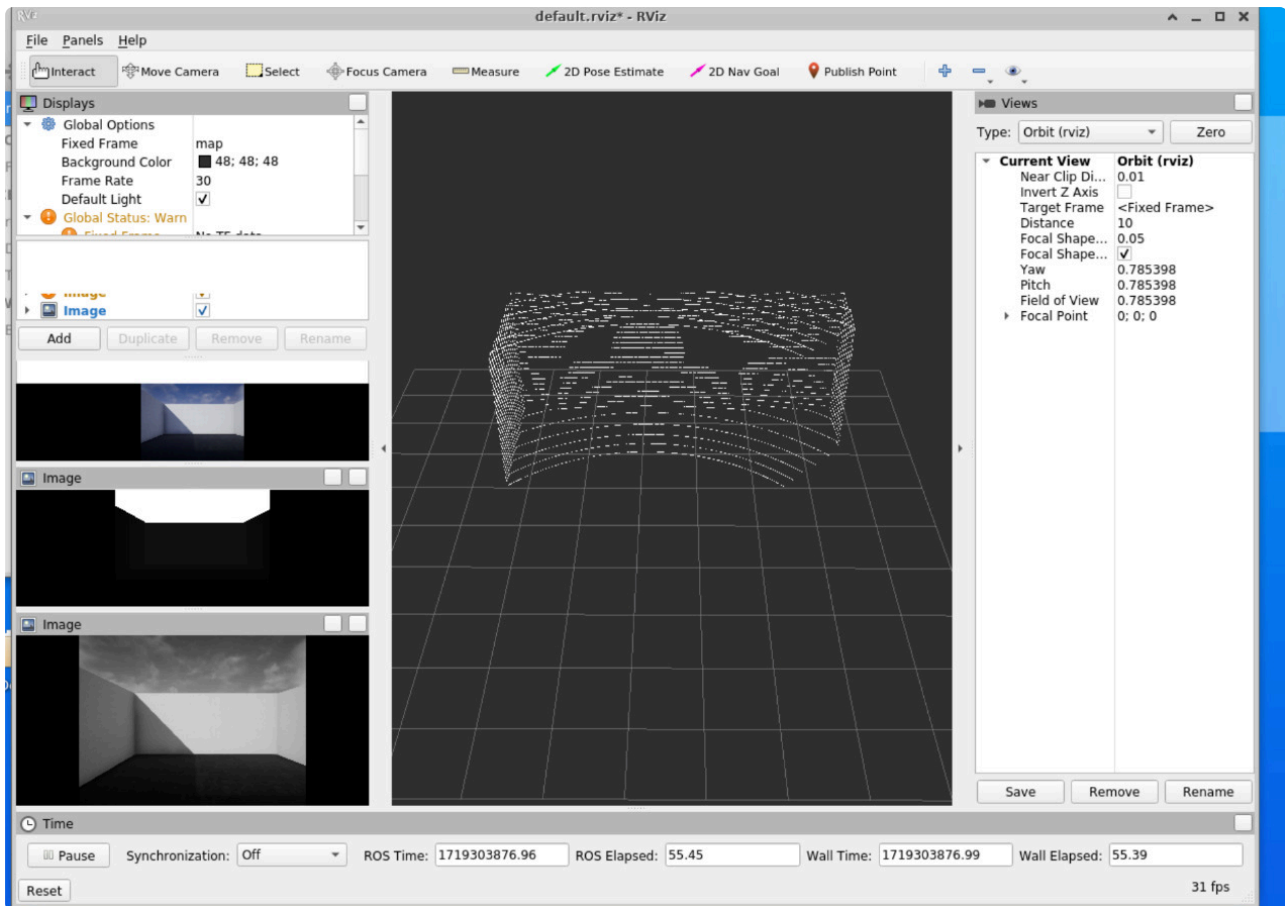
再双击打开[WinWSL.bat](#)，运行命令 `python3 server_ue4.py` 运行脚本 `server_ue4.py`。



```
Terminal - root@RFLYSIM54: /mnt/c/PX4PSP/RflySimAPIs/8.RflySimVision/0.ApiExps/5.VisCaptureMerge ^ _ □ ×
File Edit View Terminal Tabs Help
roscore http://RFLYSIM54:11311/ × root@RFLYSIM54: /mnt/c/PX4PS... × root@RFLYSIM54: /mnt/c/PX4PS... ×
root@RFLYSIM54:/mnt/c/PX4PSP/RflySimAPIs/8.RflySimVision/0.ApiExps/5.VisCaptureMergeROSAPI# 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/5.VisCaptureMergeROSAPI/Config.json
Got 4 vision sensors from json
Start lisening to timeStmp Msg
Got time msg from CopterSim # 3
CopterSim not on this PC
1719303811.52
1719303814.3526132
Got time msg from CopterSim # 4
CopterSim not on this PC
1719303811 598
```

然后再双击打开[WinWSL.bat](#)，另起一个终端运

行 `roslaunch rviz rviz -d ./vision.rviz` 命令，在可视化界面中可以看到打开了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PointCloud2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：



## 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.GeneralUbuntuConfig\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

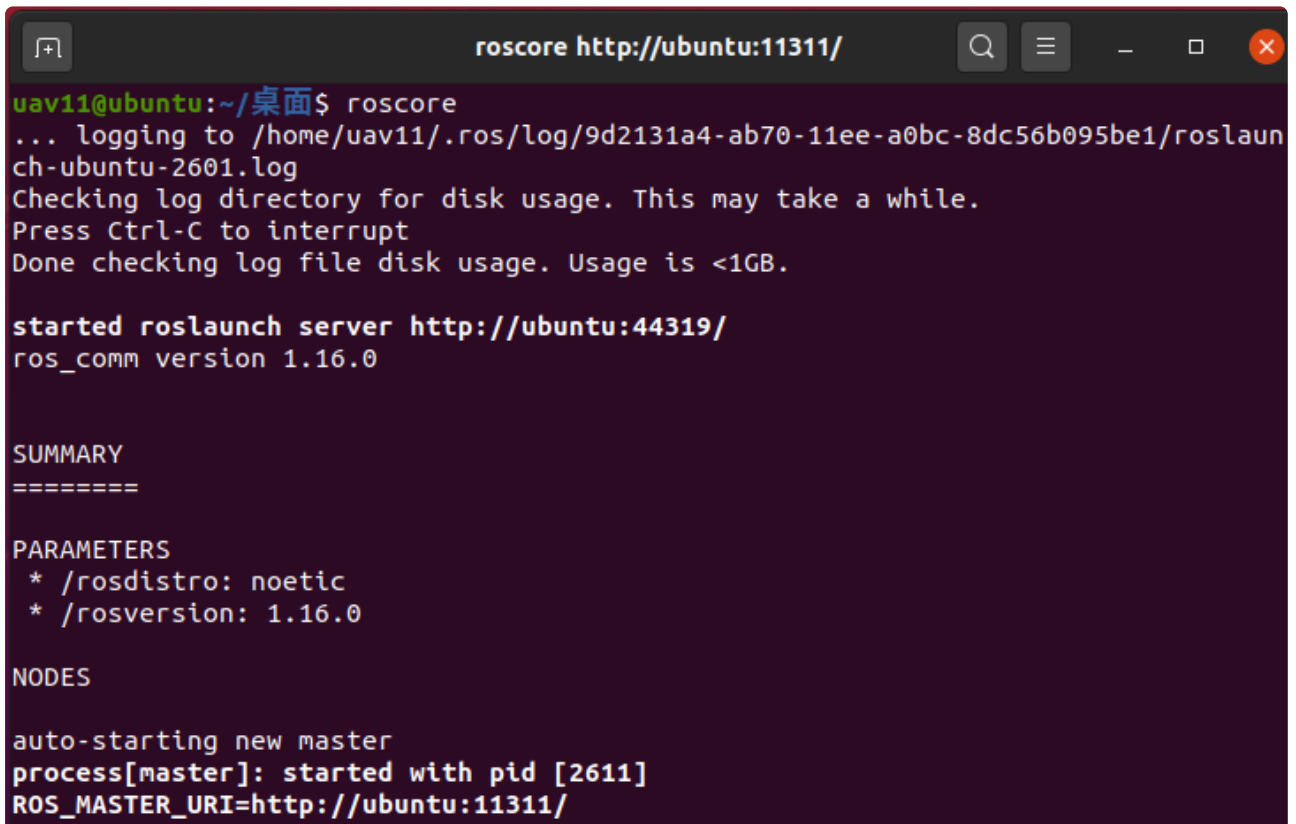
步骤1同上面的Step1步骤。

### Step 2: 运行控制程序

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 PointCloud2 下的话题填入图像数据话题及点云数据话题 `/rflsim/sensor0/img_rgb` (彩色图)、`/rflsim/sensor1/img_depth` (深度图)、`/rflsim/sensor2/img_gray` (灰度图)、`/rflsim/sensor3/vehicle_lidar` 可看到如下效果：



```
roscore http://ubuntu:11311/
uav11@ubuntu:~/桌面$ roscore
... logging to /home/uav11/.ros/log/9d2131a4-ab70-11ee-a0bc-8dc56b095be1/roslaunch-ubuntu-2601.log
Checking log directory for disk usage. This may take a while.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://ubuntu:44319/
ros_comm version 1.16.0

SUMMARY
=====

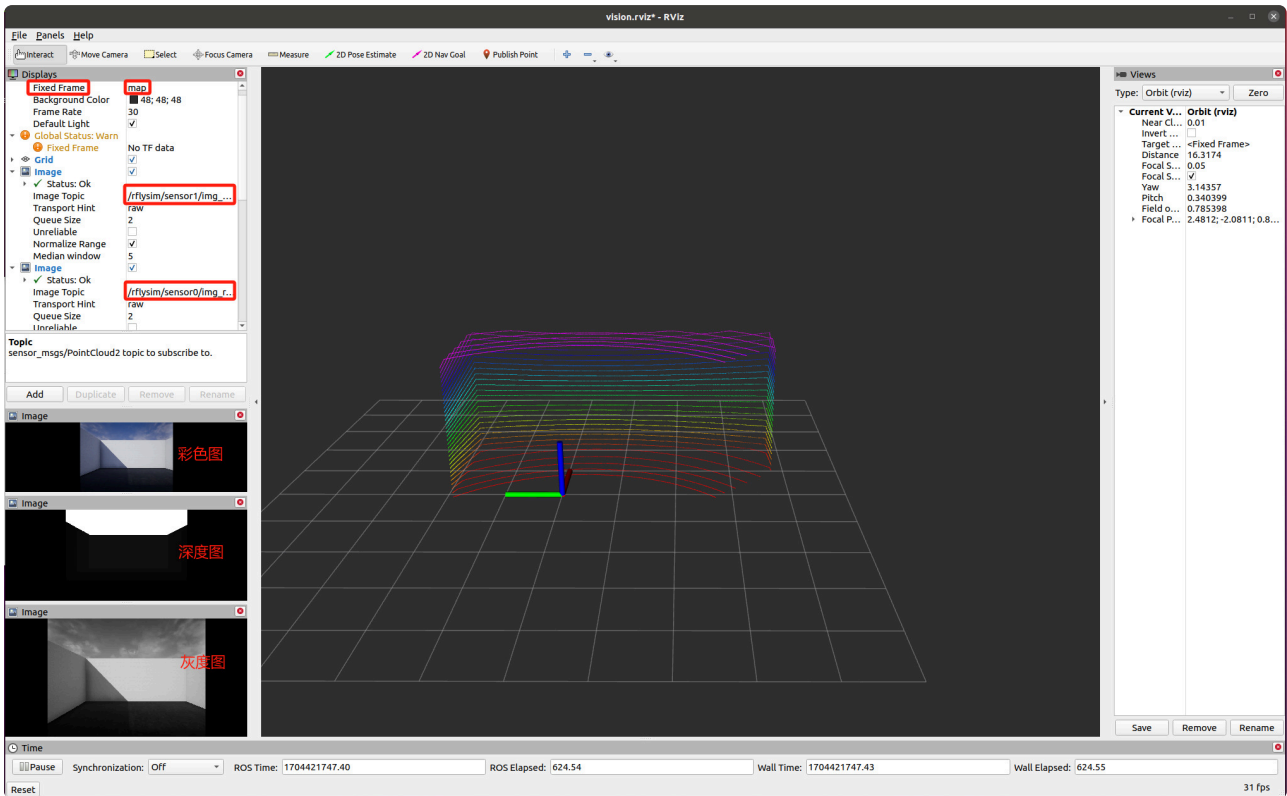
PARAMETERS
* /rostdistro: noetic
* /rosversion: 1.16.0

NODES

auto-starting new master
process[master]: started with pid [2611]
ROS_MASTER_URI=http://ubuntu:11311/
```

```
uav11@ubuntu: ~/桌面/4.VisCaptureMergeROSAPI
uav11@ubuntu:~/桌面/4.VisCaptureMergeROSAPI$ 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/4.VisCaptureMergeROSAPI/Config.json
Got 4 vision sensors from json
Start lisening to timeStmp Msg
Got time msg from CopterSim # 1
CopterSim not on this PC
1704421072.462
1704421072.3900423
Got CopterSim time Data for img
Got start time for SeqID # 0
Got start time for SeqID # 1
Got start time for SeqID # 2
Got start time for SeqID # 3
Start Image Reciver
Start lisening to IMU Msg
Got CopterSim IMU Msg!
Error to listen to IMU Msg!
PX4 Armed!
```

```
uav11@ubuntu: ~/桌面/4.VisCaptureMergeROSAPI
uav11@ubuntu:~/桌面/4.VisCaptureMergeROSAPI$ rviz
[ INFO] [1704421121.142888902]: rviz version 1.14.20
[ INFO] [1704421121.142981777]: compiled against Qt version 5.12.8
[ INFO] [1704421121.143001597]: compiled against OGRE version 1.9.0 (Ghadamon)
[ INFO] [1704421121.211590572]: Forcing OpenGL version 0.
[ INFO] [1704421122.451333891]: Stereo is NOT SUPPORTED
[ INFO] [1704421122.451452528]: OpenGL device: SVGA3D; build: RELEASE; LLVM;
[ INFO] [1704421122.451620693]: OpenGL version: 4.1 (GLSL 4.1) limited to GLSL 1
.4 on Mesa system.
```



## Step 3: 结束仿真

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**

## 6. 参考资料

无

## 7.常见问题

Q1: 无

A1: 无