
1、实验名称及目的

Windows 平台图像发送与 Linux 环境接收图片撞击小球实验：通过在 Windows 平台下调用接口进行图像数据的请求转发，然后在 Linux 环境下进行图像数据的接收，并通过视觉算法进行对小球的撞击。

2、实验原理

通信模式

1.在此演示中，采用了广播 UDP，其中

1) 在 clientue4SITL.bat 和 clientue4HITL.bat 中设置 “SET ISBROADCAST=1” 或 (SET ISBROADCAST=255.255.255.255)；

2) 在 clientue4.py 中设置 “TargetIP='255.255.255.255'”；

3) 在 serverue4.py 和 serverue4ROS.py 中设置 “mav = PX4MavCtrl.PX4MavCtrlr(20100,'255.255.255.255')”。

2.使用 IP 模式可以提高通信性能。假设主控计算机的 IP 是 192.168.1.20（运行 clientue4.py），目标嵌入式计算机的 IP 是 192.168.1.25（运行 serverue4.py）。以下更改将启用 IP 通信模式。

1) 在 clientue4SITL.bat 和 clientue4HITL.bat 中设置 “SET ISBROADCAST=192.168.1.25”（目标嵌入式计算机的 IP）；

2) 在 clientue4.py 中设置 “TargetIP='192.168.1.25'”（目标嵌入式计算机的 IP）；

3) 在 serverue4.py 和 serverue4_ROS.py 中设置 “mav = PX4MavCtrl.PX4MavCtrlr(20100,'192.168.1.20')”（主控计算机的 IP）。

在本例程中，会使用到两种连接方式连接飞控与主机，由于连接方式与主机的系统的原因对于创建的控制接口有如下情况：

当为 Windows 主机与飞控之间的连接为 USB 连接时，接口函数应为 PX4MavCtrlr(1,'127.0.0.1','COM3',57600)，其中 57600 为数据传输的波特率；

当为 Windows 主机与飞控之间的连接方式为串口连接时，接口函数应为 PX4MavCtrlr(1,'127.0.0.1','COM4',57600)，其中 57600 为数据传输的波特率；

当为 Linux 主机与飞控之间的连接为 USB 连接时，接口函数应为 PX4MavCtrlr(1,'127.0.0.1','/dev/ttyS0',57600)，其中 57600 为数据传输的波特率；

当为 Linux 主机与飞控之间的连接为串口连接时，接口函数应为PX4MavCtrlr(1,'127.0.0.1','/dev/ttyAMA0',57600)，其中 57600 为数据传输的波特率；

若不连接飞控，可直接使用 IP 地址建立通信

通过上述控制接口的使用方法正确建立主机与飞控之间的通信，即可对无人机进行仿真控制。

API 文件（见 PX4PSPRfySimAPIs\RflySimSDK 目录）

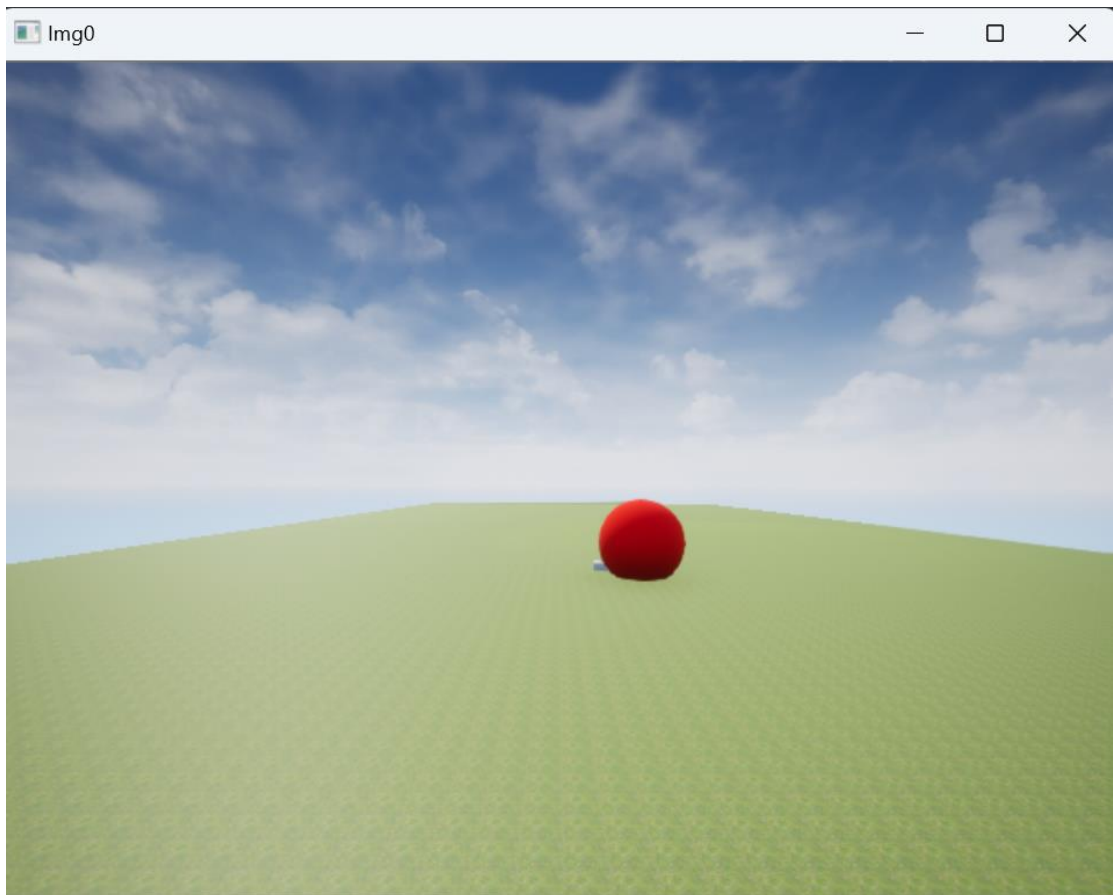
1.PX4MavCtrlV4.py 是通过 Mavlink 与 Pixhawk 进行通信的 API（并通过 UDP 与 UE4 进行通信）。PX4MavCtrlV4ROS.py 是 mavros 版本的 API。

2.ScreenCapApiV4.py 是屏幕捕获 API。值"isNewUE=False"将启用旧的 API 用于 RflySim3D 屏幕捕获，其速度更快，但不兼容 UE4.23+；值"isNewUE=True"将启用新的 API 用于 RflySim3D 屏幕捕获，其速度稍慢，但兼容所有 UE4 版本。

3.RflyVisionAPI.py 是图像传输 API。

3、实验效果

运行 server_ue4.py 文件后飞机起飞并向后倒退一定距离开始对其红球进行跟踪撞击。



4、文件目录

文件夹/文件名称	说明
client_ue4_SITL.bat	启动仿真配置文件
clinent_ue4.py	Python 实验脚本
Config.json	视觉传感器配置文件

server_ue4.py	仿真 python 实验脚本
server_ue4_Serial.py	串口连接的仿真实验脚本
server_ue4ROS.py	ROS 下的仿真实验脚本
server_ue4ROS_Serial.py	ROS 下通过仿真实验脚本

5、运行环境

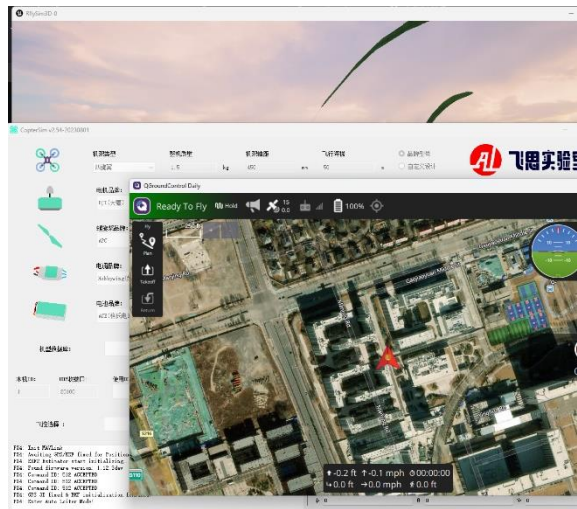
序号	软件要求	硬件要求	
		名称	数量(个)
1	Windows 10 及以上版本	笔记本/台式电脑 ^①	1
2	RflySim 平台高级版及以上		
3	Visual Studio Code		

- ①：推荐配置请见：<https://doc.rflysim.com/1.1InstallMethod.html>
- ②：须保证平台安装时的编译命令为：droneyee_zyfc-h7_default，固件版本为：1.12.1。其他配套飞控请见：<http://doc.rflysim.com/hardware.html>

6、实验步骤

Step 1:

运行 clientue4SITL.bat 来启动 SITL 模拟（或者运行 clientue4HITL.bat 来启动 HIL 模拟，其中 Pixhawk 应该配置好）。



Step 2:

运行 PX4PSPRfySimAPIs\RflySimSDK 目录下的 ReLabPath.py 文件。

Step 3:

运行 Python38Run.bat 并输入 “python client_ue4.py” 来运行客户端程序，该程序捕获 RflySim3Ds 的屏幕并通过 UDP 发送到网络。

```
C:\Windows\system32\cmd.exe
Python3.8 environment has been set with openCV+pymavlink+numpy+pyulog etc.
You can use pip or pip3 command to install other libraries
Put your python scripts 'XXX.py' into the folder 'F:\8.RflySimVision\1.BasicExps\2-DistributedSimDemos\e1_OneVehilceCtrl
s\2-ShootBall'
Use the command: 'python XXX.py' to run the script with Python

F:\8.RflySimVision\1.BasicExps\2-DistributedSimDemos\e1_OneVehilceCtrls\2-ShootBall>python client_ue4.py
```

```
问题 输出 终端 调试控制台 端口 Python Debug
Error: No sensor is obtained.
PS F:\8.RflySimVision\1.BasicExps\2-DistributedSimDemos\e1
all> ^C
PS F:\8.RflySimVision\1.BasicExps\2-DistributedSimDemos\e1
all> ^C
PS F:\8.RflySimVision\1.BasicExps\2-DistributedSimDemos\e1
在新窗口中打开文件夹 (Ctrl + 单击) ion\1.BasicExps\2-Distributed
ls\2-ShootBall'; & 'C:\PX4PSP\Python38\python.exe' 'c:\Use
ns\ms-python.python-2023.18.0\pythonFiles\lib\python\debug
launcher' '61512' '--' 'F:\8.RflySimVision\1.BasicExps\2-D
VehilceCtrls\2-ShootBall\client_ue4.py'
jsonPath= F:\RflySimSDK\vision\shootball\Config.json
Got 1 vision sensors from json
Sensor req success from UE4.
Start Transfer Img
█
```



Step 4:

使用 VS Code 打开 `server_ue4.py` 并在此计算机上运行它，以接收来自本地网络的图像并使用计算机视觉算法进行处理。

```
Got 1 vision sensors from json
MainThreadFPS: 29.92919909802187
5s, Arm the drone
Arm the drone!, and fly to NED 0,0,-5
PX4 Armed!
MainThreadFPS: 24.077105445945246
MainThreadFPS: 30.03655113148095
MainThreadFPS: 29.968820823805608
15s, fly to pos: -30,-5, -5
```

Step 5:

您可以将此文件夹中的所有文件复制到另一台计算机（树莓派、TX2 或任何具有 Linux 和 ROS 环境的计算机），使用命令“`python3 server_ue4.py`”通过 UDP 从前一台计算机接收图像并使用人脸识别算法进行处理。

`serverue4ROS.py` 是“`server_ue4.py`”的 ROS 版本。它们之间唯一的区别是句子“`impo`

rt PX4MavCtrlV4 as PX4MavCtrl” 和 “import PX4MavCtrlV4ROS as PX4MavCtrl”，其中 PX4MavCtrlV4.py 是通过 pymavlink 进行的 Mavlink API，而 PX4MavCtrlV4ROS.py 是通过 mavros 进行的 Mavlink API。

7、参考文献

[1]. 无

8、常见问题

Q1: 仿真启动后一直卡在初始地图且无飞机出现。

A1: 仿真 bat 脚本使用了广播通信方式，在 CopterSim 中等待如下语句出现后，切换到仿真界面按下按键 i