
1、实验名称及目的

数据 UDP 直传不压缩实验：尝试使用 UDP 直传不压缩的传输的方式传图。

2、实验原理

在客服端程序中使用 `vis.jsonLoad()` 函数导入 `config.json` 文件中的相机配置，其参数配置如下

“SeqID”代表第几个传感器。此处 0 表示第 1 个传感器（免费版只支持 2 个图）。

“TypeID”代表传感器类型 ID，1:RGB 图（免费版只支持 RGB 图），2:深度图，3:灰度图，本实验共采用了三种不同的传感器类型。

“TargetCopter”传感器装载的目标飞机的 ID，可改变。

“TargetMountType”代表坐标类型，0：固定飞机上（相对几何中心），1：固定飞机上（相对底部中心），2：固定地面上（监控）也可变。

“DataWidth”为数据或图像宽度此处为 640，“DataHeight”为数据或图像高度此处为 480。

“DataCheckFreq”检查数据更新频率此处为 30HZ。

“SendProtocol[8]”为传输方式与地址，SendProtocol[0]取值 0：共享内存（免费版只支持共享内存），1：UDP 直传 png 压缩，2：UDP 直传图片不压缩，3：UDP 直传 jpg 压缩；SendProtocol[1-4]：IP 地址；SendProtocol[5]端口号。

“CameraFOV”为相机视场角（仅限视觉类传感器），单位度也可改变。

“SensorPosXYZ[3]”为传感器安装位置，单位米也可改变。

“SensorAngEular[3]”为传感器安装角度，单位度°也可改变。

然后修改 `vis.isUE4DirectUDP` 的值为 True 将会强制将传输方式改为 UDP 直传。

在服务端程序中，首先通过 `PX4MavCtrl.PX4MavCtrlr()` 创建一个通信接口实例命名为 `mav`，其中在硬件在环仿真中，使用的主机不同需要在 `PX4MavCtrl.PX4MavCtrlr()` 中设定不同的端口号与通信波特率，其输入格式也有所不同具体如下

ID: 如果 $ID \leq 10000$ 则表示飞机的 CopterID 号。如果 $ID > 10000$ ，例如 20100 这种，则表示通信端口号 port。按平台规则， $port = 20100 + CopterID * 2 - 2$ （为了兼容旧接口的过渡定义，将来 ID 只表示 CopterID）。

ip: 数据向外发送的 IP 地址。默认是发往本机的 127.0.0.1 的 IP；在分布式仿真时，也可以指定 192.168 打头的局域网电脑 IP；也可以使用 255.255.255.255 的广播地址（会干扰网络其他电脑）

Com: 与 Pixhawk 的连接模式。

Com='udp'，表示使用默认的 udp 模式接收数据，这种模式下，是接收 CopterSim 转发的 PX4 的 MAVLink 消息（或 UDP_full,simple）消息包

使用 port+1 端口收和 port 端口发（例如，1 号飞机是 20101 端口收，20100 端口

发，与 CopterSim 对应）。

Com='COM3'（Windows 下）或 Com='/dev/ttyUSB0'（Linux 系统，也可能是 ttyS0、ttyAMA0 等），表示通过 USB 线（或者数传）连接飞控，使用默认 57600 的波特率。注意：波特率使用 port 口设置，默认 port=0，会重映射为 57600

Com='Direct'，表示 UDP 直连模式（对应旧版接口的真机模式），这种模式下使用同一端口收发（端口号有 port 设置），例如 Com='Direct'，port=15551，表示通过 15551 这一个端口来收发数据

注意：COM 模式和 Direct 模式下，ID 只表示飞机的 ID 号，而不表示端口号

Com='redis':使用 Redis 模式通信，服务器地址为 ip，服务器端口为 port

port: UDP 模式下默认情况下设为 0，会自动根据 IP 填充，按平台规则，port=20100+CopterID*2-2。如果这里赋值大于 0，则会强制使用 port 定义的端口。

COM 模式下，Port 默认表示波特率 self.baud=port。如果 port=0，则会设置 self.baud=57600

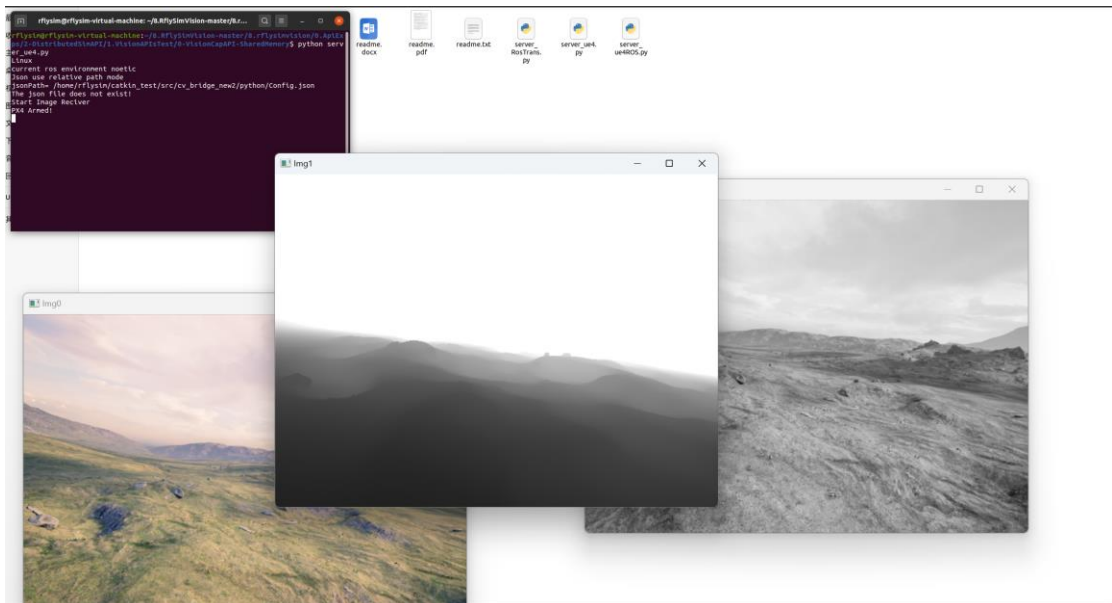
Direct 模式下，Port 默认表示收发端口号（使用相同端口）

redis 模式下，Port 对应服务器端口号 self.redisPort = port。如果 port=0，则 self.redisPort=6379 为平台默认值。

然后根据 Config.json 配置相机，并使用 vis.startImgCap 函数开启取图，然后通过 SendPosNED 函数发送位置指令，飞机即可完成飞行，通过一个死循环将 UDP 直接接收到的图像实时显示在屏幕上。

3、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



4、文件目录

文件夹/文件名称	说明
client_ue4.py	客户端程序
server_ue4.py	服务端程序
server_ue4Ros.py	Ros 环境下的服务端程序
Config.json	相机配置文件
client_ue4_HITL.bat	硬件在环仿真脚本
client_ue4_SITL.bat	软件在环仿真脚本
Python38Run.bat	Python 环境启动脚本

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		
4	Linux	虚拟机/NX	1

①：推荐配置请见：<https://doc.rflysim.com/1.1InstallMethod.html>

6、实验步骤

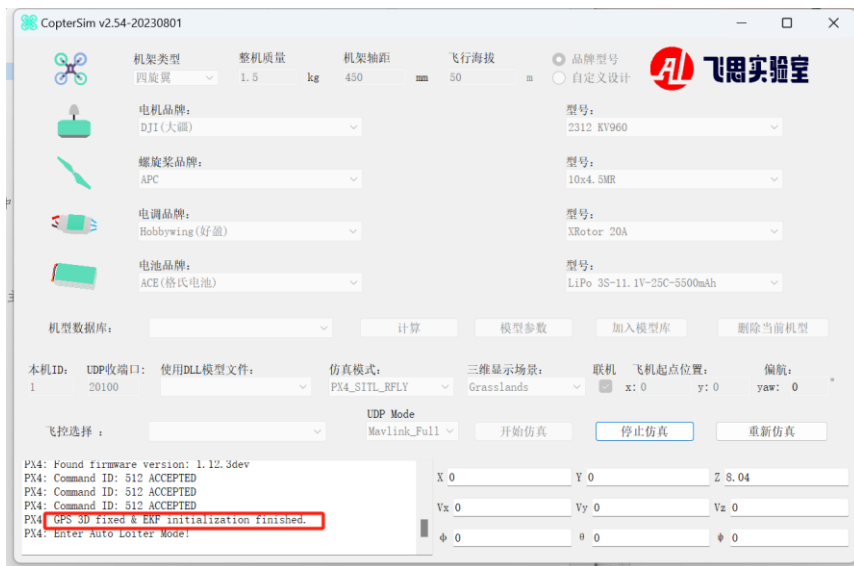
Windows 下

Step 1:

选中 client_ue4_SITL.bat,右键选择以管理员身份运行，启动仿真脚本

名称	修改日期	类型	大小
client_ue4.py	2023/11/1 18:02	Python 源文件	2 KB
client_ue4_HITL.bat	2023/10/31 15:28	Windows 批处理文件	5 KB
client_ue4_SITL.bat	2023/11/2 15:38	Windows 批处理文件	5 KB
Config.json	2023/10/31 15:28	JSON 源文件	2 KB
Python38Run.bat	2023/10/31 15:28	Windows 批处理文件	1 KB
readme.docx	2023/10/31 15:28	Microsoft Word 文档	14 KB
readme.pdf	2023/10/31 15:28	QQBrowser pdf Do...	86 KB
readme.txt	2023/10/31 15:28	文本文档	3 KB
Readme模板.docx	2023/11/7 10:28	Microsoft Word 文档	3,355 KB

等待 CopterSim 界面中返回如下语句即可



Step 2:

在 Windows 主机中打开 client_ue4.py 文件并运行

```
# -*- coding: utf-8 -*-
import cv2
import sys
import time
import VisionCaptureApi
import os
import UE4CtrlAPI
ue = UE4CtrlAPI.UE4CtrlAPI()

# The IP should be specified by the other computer
vis = VisionCaptureApi.VisionCaptureApi()

# Send command to UE4 Window 1 to change resolution
ue.sendUE4Cmd('r.setres 720x485w',0) # 设置UE4窗口分辨率, 注意本窗口仅限于显示, 取图分辨率在json中配置, 本窗口设置越小, 资源需求越少。
ue.sendUE4Cmd('t.MaxFPS 30',0) # 设置UE4最大刷新频率, 同时也是取图频率
time.sleep(2)

# VisionCaptureApi 中的配置函数
vis.jsonLoad() # 加载Config.json中的传感器配置文件

# vis.RemotSendIP = '192.168.3.80'
# 注意, 手动修改RemotSendIP的值, 可以将图片发送到本地地址
# 如果不修改这个值, 那么发送的IP地址为json文件中SendProtocol[1:4]定义的IP
# 图片的发送端口, 为json中SendProtocol[5]定义好的。

isSuss = vis.sendReqToUE4() # 向RflySim3D发送取图请求, 并验证
if not isSuss: # 如果请求取图失败, 则退出
    sys.exit(0)
vis.startImgCap(True) # 开启取图, 并启用共享内存图像转发, 转发到填写的目录
print('Start Transfer Img')

# 下面的程序非必需, 仅用于观察图像用, 在电脑性能不足时, 请删除
lastTime = time.time()
num=0
lastClock=time.time()
while True:
    lastTime = lastTime + 1/30.0
    sleepTime = lastTime - time.time()
    if sleepTime > 0:
        time.sleep(sleepTime)
    else:
        lastTime = time.time()

    for i in range(len(vis.hasData)):
        if vis.hasData[0]:
            cv2.imshow('Img'+str(i),vis.Img[i])
            cv2.waitKey(1)
```

Step 3:

回到 文件夹中双击 Python38Run.bat 启动 python3.8 脚本, 输入 python server_ue4.py, 运行 server_ue4.py 文件。

```
C:\Windows\system32\cmd.exe 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 your python scripts 'XXX.py' into the folder 'C:\PX4PSP\RflySimAPIs\8.RflySimVision-master\8.rflysimvision\0.ApiExps\2-DistributedSimAPI\1.VisionAPITest\0-VisionCapAPI-SharedMemory'
Use the command: 'python XXX.py' to run the script with Python

C:\PX4PSP\RflySimAPIs\8.RflySimVision-master\8.rflysimvision\0.ApiExps\2-DistributedSimAPI\1.VisionAPITest\0-VisionCapAPI-SharedMemory>python server_ue4.py
Failed to load Redis labs
Json use relative path mode
jsonPath= C:\PX4PSP\RflySimAPIs\8.RflySimVision-master\8.rflysimvision\0.ApiExps\2-DistributedSimAPI\1.VisionAPITest\0-VisionCapAPI-SharedMemory\Config.json
Got 3 vision sensors from json
Start Image Reciver
```

Step 4:

回到 Windows 主机上观察 RflySim3D 查看飞机

Linux 下

Step 1:

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

```
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=192.168.70.128

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:

:Top
ECHO.
ECHO -----
REM Max vehicle number 50
SET /a MAX_VEHICLE=50
```

Step 2:

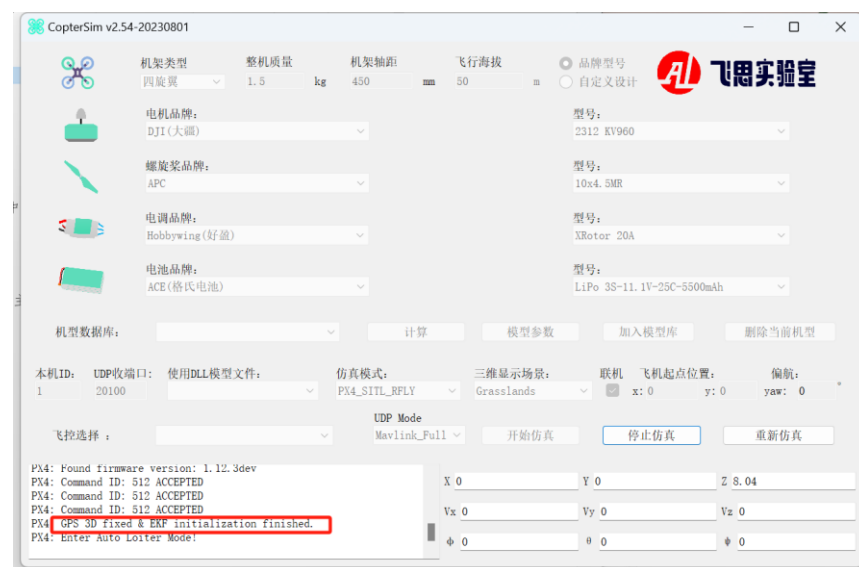
将文件夹下的 server_ue4ROS.py 文件复制到 Linux 系统文件下

Step 3:

在 Windows 主机中，右键点击 client_ue4_SITL.bat 选择以管理员身份运行启动仿真界面

名称	修改日期	类型	大小
client_ue4.py	2023/11/1 18:02	Python 源文件	2 KB
client_ue4_HITL.bat	2023/10/31 15:28	Windows 批处理文件	5 KB
client_ue4_SITL.bat	2023/11/2 15:38	Windows 批处理文件	5 KB
Config.json	2023/10/31 15:28	JSON 源文件	2 KB
Python38Run.bat	2023/10/31 15:28	Windows 批处理文件	1 KB
readme.docx	2023/10/31 15:28	Microsoft Word 文档	14 KB
readme.pdf	2023/10/31 15:28	QQBrowser pdf Do...	86 KB
readme.txt	2023/10/31 15:28	文本文档	3 KB
Readme模板.docx	2023/11/7 10:28	Microsoft Word 文档	3,355 KB

等待 CopterSim 界面中返回如下语句即可



Step 4:

在 Windows 主机中打开 client_ue4.py 文件将 vis.RemotSendIP = '192.168.3.80' 语句的注释取消并设置为 Linux 主机的 IP 地址，并运行

```
time.sleep(2)

# VisionCaptureApi 中的配置函数
vis.jsonLoad() # 加载Config.json中的传感器配置文件

vis.RemotSendIP = '192.168.3.80'
# 注意，手动修改RemotSendIP的值，可以将图片发送到本地址
# 如果不修改这个值，那么发送的IP地址为json文件中SendProtocol[1:4]定义的IP
# 图片的发送端口，为json中SendProtocol[5]定义好的。

isSuss = vis.sendReqToUE4() # 向RflySim3D发送取图请求，并验证
if not isSuss: # 如果请求取图失败，则退出
    sys.exit(0)
vis.startImgCap(True) # 开启取图，并启用共享内存图像转发，转发到填写的目录
print('Start Transfer Img')
```

Step 5:

在 Linux 系统中打开 server_ue4ROS.py 文件，将 `MavList=MavList+[PX4MavCtrl.PX4MavCtrl(1+i,'255.255.255.255')]` 中的 '255.255.255.255' 改为 Windows 主机的地址，并运行。

```
VehilceNum = 3
MavList=[]
# Create MAV instance
for i in range(VehilceNum):
    MavList=MavList+[PX4MavCtrl.PX4MavCtrl(1+i,'255.255.255.255')]
time.sleep(2)
# Start MAV loop with UDP mode: MAVLINK_FULL
for i in range(VehilceNum):
    MavList[i].InitMavLoop()

# Enter Offboard mode to start vehicle control
time.sleep(2)
for i in range(VehilceNum):
```

Step 6:

回到 Windows 主机上观察 RflySim3D 查看飞机

硬件在环仿真

硬件在环仿真的方法与软件在环仿真类似，不同点在于启动仿真脚本使用的是 client_ue4_HITL.bat 文件，此外，在 server_ue4.py 等服务端例程程序中，需要将 `MavList=MavList+[PX4MavCtrl.PX4MavCtrl(1+i,'255.255.255.255')]` 中的 `1+i,'255.255.255.255'` 值进行修改，具体修改方式参考实验原理。

7、参考文献

[1]. 无

8、常见问题

Q1: 无

A1: 无