

1. 实验名称及目的

1.1. 实验名称

目标跟踪算法实验

1.2. 实验目的

通过平台接口进行对图像的获取，然后通过目标跟踪算法控制无人机对目标物体的跟踪。

1.3. 关键知识点

本实验主要是实现通过Python接口VisionCaptureApi.py（见RflySimAPIs\RflySimSDK\vision目录）获取RflySim3D图像并实时更新相机参数（姿态、位置、FOV等），对图像进行图像算法的运算处理以及目标跟踪算法运算，控制跟随无人机跟踪另一架无人机运动。。关键代码解析如下：

1) 视觉接口使用

```
1 vis = VisionCaptureApi.VisionCaptureApi() \# 创建一个视觉传感器实例
2
3 vis.jsonLoad() \# 加载Config.json中的传感器配置文件
4
5 isSuss = vis.sendReqToUE4() \# 向RflySim3D发送取图请求
6
7 vis.startImgCap() \# 开启取图
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.rectangle(img, p1, p2, (255, 0, 0)) \#画出一个长方形，在img图像里，左上角右下角的坐标分别
14
15 cv2.imshow('Img'+str(i),vis.Img[i]) \# 显示图片i图像
16
17 tracker = cv2.TrackerTLD_create() \# 使用 OpenCV 中的 TLD跟踪器创建一个跟踪对象tracker
```

2) 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和1的两个视觉传感器
2 |
3 | "TypeID":1 : 传感器类型为RGB彩色图像
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":[0.3,-0.15,0]和"SensorPosXYZ":[0.3,0.15,0] : 两个RGB相机一左一右分布。
```

3) 飞机控制指令

```
1 | MavList = MavList+[PX4MavCtrl.PX4MavCtrl(1+i)] \#初始化并建立i号飞机的MAVLink通信连接，送
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 | MavList[i].InitTrueDataLoop() \# 通过 30100 系列端口初始化来自 CopterSim 的 UDP True 数据
8 |
9 | MavList[i].SendPosNED(0, 0, -10, 0) \# 对第i个飞机发送10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
```

4) UE控制

接口详细使用方法见：[UE4CtrlAPI.py](#)

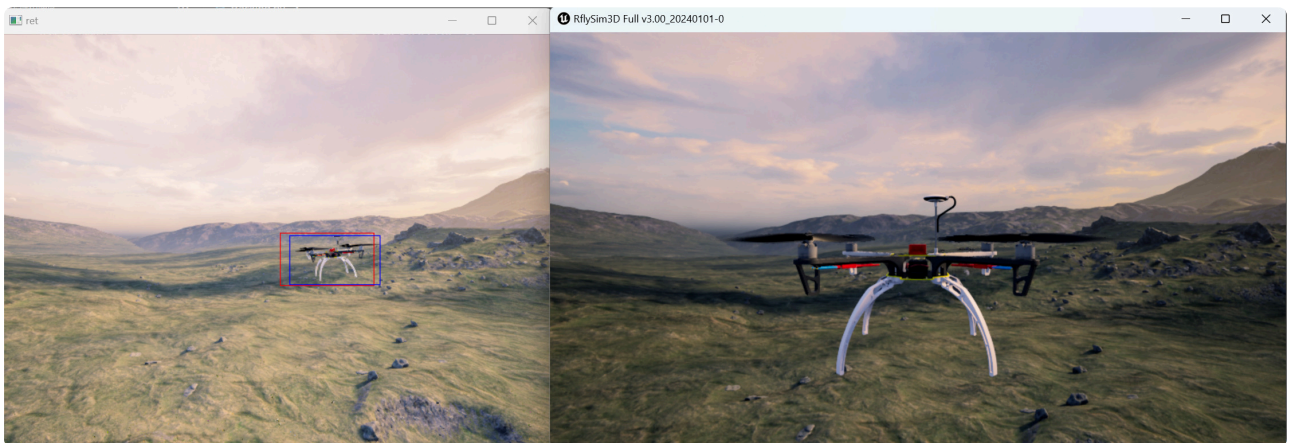
```
1 | ue = UE4CtrlAPI.UE4CtrlAPI() \# 创建UE控制实例
2 |
3 | ue.sendUE4Cmd('r.setres 1280x720w',0) \#发送指令，设置UE4窗口分辨率，注意本窗口仅限于显示，取
4 |
5 | ue.sendUE4Cmd('t.MaxFPS 30',0) \#发送指令，设置UE4最大刷新频率30Hz，同时也是取图频率
6 |
7 | ue.reqCamCoptObj(1, [1, 2]) \# 创建带copter_id的目标，对应相机的seqID为[1,2]。设置一个实例
```

5) 其余代码说明

```
1 def CtrlTargetDrone(mav: PX4MavCtrl.PX4MavCtrl, max_vel) \#用于控制无人机 (目标无人机) ↓
2
3 def GetFrame(target_drone: UE4CtrlAPI.CoptReqData, camera:UE4CtrlAPI.CameraData) \#用
4
5 def Tracking(ue: UE4CtrlAPI.UE4CtrlAPI) \#通过不断更新和处理图像数据以及飞机信息, 实现了目标的
6
7 timeInterval = 1/30.0 \# 以30hz的频率进行控制
8
9 lastTime = lastTime + timeInterval \# 设置每一帧的处理结束时间
10
11 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
```

2.实验效果

会看到在飞机右侧又生成了一架飞机, 两架飞机起飞后右侧飞机会飞行到前方并进行运动, 同时生成一个窗口, 显示目标飞机识别检测框与跟踪检测框重合效果。



3.文件目录

例程目录: [\[安装目录\]\RflySimAPIs\8.RflySimVision\2.AdvExps\e8_SingleObjTracking](#)

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

文件夹/文件名称	说明
Python38Run.bat	Python程序运行脚本

4. 运行环境

4.1 软件要求

Windows 10及以上版本；RflySim工具链；VS Code。

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。

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：Windows取图控制

Step 1: 开启仿真

双击打开[Tracking.bat](#)脚本，启动仿真环境。

Step 2: 运行控制程序

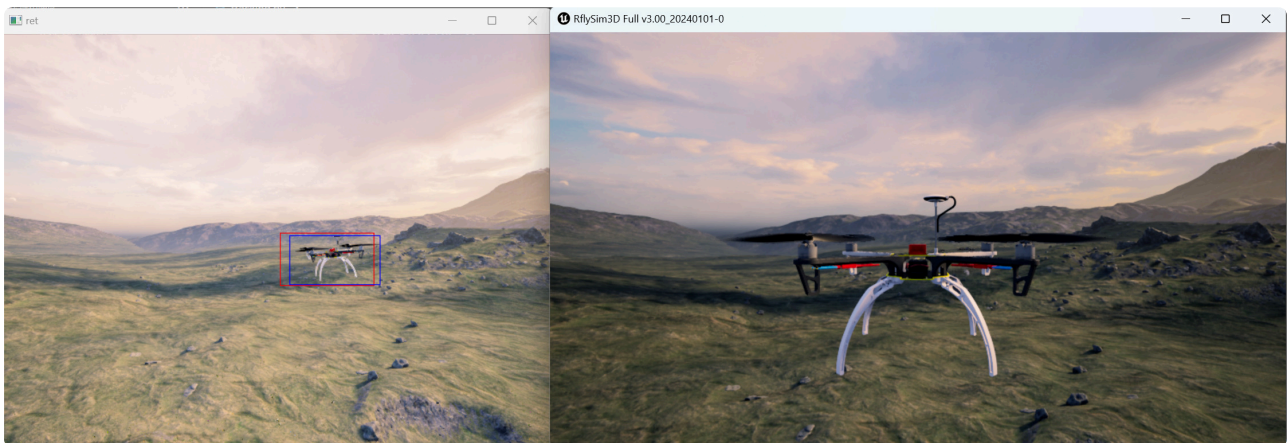
在文件夹下，双击[Python38Run.bat](#)，打开集成好的python环境，在该环境下运行[tracking.py](#)文件，输入 `python tracking.py`。

```
C:\Windows\system32\cmd.e. 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 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\0.Preparation\4.AutoObtainIPRun>python server_ue4.py
```

Step3: 观察结果

可以看到在飞机右侧又生成了一架飞机，两架飞机起飞后右侧飞机会飞行到前方并进行运动，同时生成一个窗口，显示目标飞机识别检测框与跟踪检测框重合效果。



Step 4: 结束仿真

在下图“Tracking.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（VS Code调试运行）

准备工作:

- 先确保已经按 [RflySimAPIs\1.RflySimIntro\2.AdvExps\3.PythonConfig\Readme.pdf](#) 步骤，正确配置VS Code环境。或者配置了自己的Pycharm等自定义Python环境。
- 其他步骤与上文相同，在Step2运行 [tracking.py](#) 时，可使用VS Code（或Pycharm等工具）来打开 [tracking.py](#) 文件，并阅读代码，修改代码，调试执行等。

扩展实验：

- 请自行使用VS Code阅读 [tracking.py](#) 源码，通过程序跳转，了解每条代码的执行原理；再通过调试工具，验证每条指令的执行效果。

🔗 VisionCapAPIDemo.py ×

```
> RflySimAPIs > 8.RflySimVision > 0.ApiExps > 1-UsageAPI > 0.VisionSensorAPI > 1.CameraAPI
8   ue = UE4CtrlAPI.UE4CtrlAPI()
9
10  #Create a new MAVLink communication instance, UDP sending
11  mav = PX4MavCtrl.PX4MavCtrl(1)
12
13  # The IP should be specified by the other computer
14  vis = VisionCaptureApi.VisionCaptureApi()
15
16  # Send command to UE4 Window 1 to change resolution
17  ue.sendUE4Cmd('r.setres 1280x720w',0) # 设置UE4窗口分辨率, 同时设置UE4窗口大小
18  ue.sendUE4Cmd('t.MaxFPS 30',0) # 设置UE4最大刷新频率, 同时也设置UE4帧率
19  time.sleep(2)
20
21  # VisionCaptureApi 中的配置函数
22  vis.jsonLoad() # 加载Config.json中的传感器配置文件
--
```

- 请尝试修改代码，实现飞机位置改变、相机姿态角改变、相机参数改变等功能。

6.参考资料

无

7.常见问题

Q1: 无

A1: 无