

1. 实验名称及目的

1.1. 实验名称

yolo 检测气球，控制飞机撞击气球实验

1.2. 实验目的

通过平台接口进行对图像的获取，然后通过yolo算法检测气球，并控制无人机撞击气球。

1.3. 关键知识点

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。

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），通过平台接口进行对图像的获取，然后通过yolo算法检测气球，并控制无人机撞击气球。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
```

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] : 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":[ 0.3,0,0] : RGB相机一左一右分布。
```

3) 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.initOffboard() \# 进入Offboard模式
6
7 mav.SendMavArm(True) \# 解锁飞控
8
9 mav.SendPosNED(0, 0, -10, 0) \# 发送10米高的位置控制指令
```

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.sendUE4Pos(110, 501, 0, [3, 0, -1], [0, 0, 0]) \#创建一个物体, if号为110, 类型为501, 速

```

5) 其余代码说明

```

1 | def Controller(pic_idx) \#根据图像中的目标位置生成无人机的控制指令。通过调整无人机的速度和姿态来
2 |
3 | detect = detect_ball.detectapi(weights=sys.path[0]+"\\best.pt") \#加载yolov5模型
4 |
5 | def DetectBalloon(img) \#用于在图像中检测气球, 并返回气球的位置、大小及检测成功与否
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

轻量级无人机质点模型飞行控制效果与软/硬件在环相近, 但更平稳。

3. 文件目录

例程目录:

[\[安装目录\]\RflySimAPIs\8.RflySimVision\2.AdvExps\e7_ObjDetectYolo\1.ShootBallBaseOnYolo](#)

文件夹/文件名称	说明
CrossRingNoPX4.bat	启动仿真配置文件
CrossRingNoPX4.py	Python实验脚本
Config.json	视觉传感器配置文件
VisionCaptureApi.py	视觉取图接口

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://rflsim.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://rflsim.com/doc/zh/HowToInstall.pdf>

5. 实验步骤

5.1 必做实验：Windows取图控制

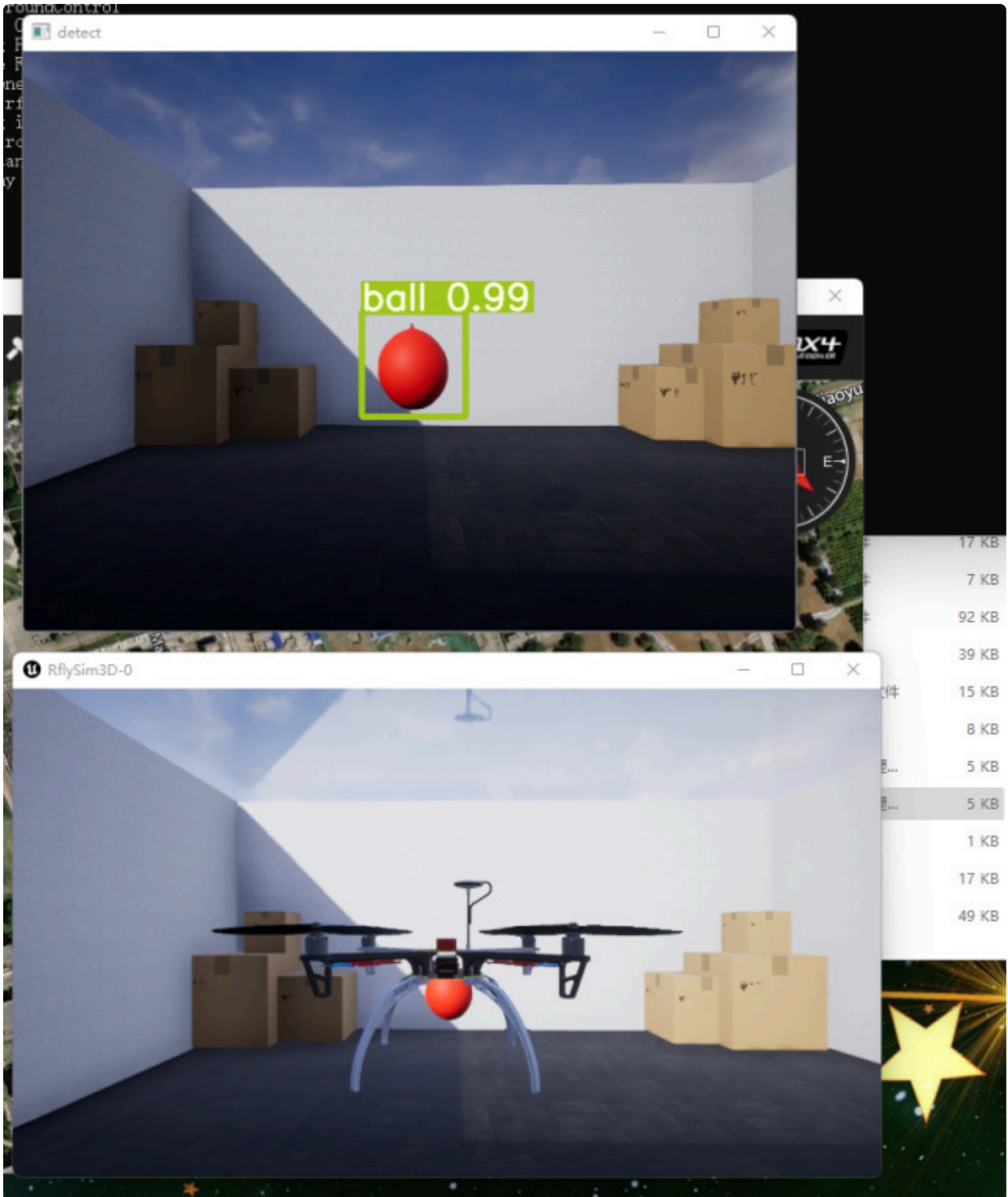
Step 1: 运行控制程序

双击运行[ShootBall3SITL.bat](#)文件，启动仿真环境；

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[ShootBall3.py](#)文件，输入 `python3 ShootBall3.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\2.AdvExps\e7_ObjDetectYolo\ShootBallBaseOnYolo>python ShootBall3.py
```

Step 2: 观察结果



Step 3: 结束仿真

在下图“[ShootBall3SITL.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。

注：若运行程序出现torch相关错误，请确保环境已配置成功。可在桌面右键打开终端(前提是配置好python环境)，打开终端后，输入 `pip list` 命令。

```
requests-toolbelt 0.9.1
rfc3986 1.5.0
rsa 4.8
scipy 1.8.1
seaborn 0.11.2
setuptools 41.2.0
SimpleWebSocketServer 0.1.1
six 1.16.0
stack-data 0.3.0
tensorboard 2.9.1
tensorboard-data-server 0.6.1
tensorboard-plugin-wit 1.8.1
thop 0.1.0.post2206102148
torch 1.11.0
torchvision 0.12.0
tornado 6.2
tqdm 4.60.0
traitlets 5.3.0
twine 3.4.1
typing 3.7.4.3
typing_extensions 4.2.0
urllib3 1.26.4
ushlex 0.99.1
wcwidth 0.2.5
webencodings 0.5.1
websocket-client 1.0.0
Werkzeug 2.1.2
wheel 0.37.1
zipp 3.4.1
```

能看到安装包的列表，我们这个例程依赖与torch,torchvision,如果没有这两个包使用以下命令安装：

```
1 | pip install torch==1.11.0
2 | pip install torchvision==0.12.0
```

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 [ShootBall3.py](#) 时，可使用VS Code（或Pycharm等工具）来打开 [ShootBall3.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 [ShootBall3.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.VisionSenorAPI > 1.Came
8   ue = UE4CTR1API.UE4CTR1API()
9
10  #Create a new MAVLink communication instance, UDP sending
11  mav = PX4MavCtrl.PX4MavCtrler(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，设置
18  ue.sendUE4Cmd('t.MaxFPS 30',0) # 设置UE4最大刷新频率，同时也
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