

1. 实验名称及目的

1.1. 实验名称

双目视觉人脸识别实验

1.2. 实验目的

通过平台 Config.json 配置文件配置好双目视觉灰度相机传感器，然后通过平台接口进行图像的获取，并在飞机起飞后开启人脸识别算法，双目框选出人脸。

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）。然后开启取图，再通过 ue.sendUE4Cmd() 函数创建人并将其动作改为步行。再通过对不同的传感器使用 cv2 自带的人脸检测识别器检测人脸位置并通过方形绘制函数框选出来，此处 vis.hasData[0] 代表第一个相机，vis.hasData[1] 代表第二个相机，最后通过死循环将图像显示在屏幕上。关键代码解析如下：

1) 视觉接口使用

```
vis = VisionCaptureApi.VisionCaptureApi() # 创建一个视觉传感器实例
vis.jsonLoad() # 加载 Config.json 中的传感器配置文件
isSuss = vis.sendReqToUE4() # 向 RflySim3D 发送取图请求
vis.startImgCap() # 开启取图
vis.hasData[i] # 图片 i 数据是否更新
vis.Img[i] # 图片 i 数据（像素矩阵）
cv2.imshow('Img'+str(i),vis.Img[i]) # 显示图片 i 图像
face_cascade=cv2.CascadeClassifier(sys.path[0]+'\\cascades\\haarcascade_frontalface_default.xml') #加载 Haar 级联分类器来检测人脸
faces1=face_cascade.detectMultiScale(pic1,1.3,5) # 第一个摄像头的脸部识别
```

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：

"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
"TypeID":1: 传感器类型为 RGB 彩色图像
"TargetCopter":1: 相机绑定在 1 号飞机上

"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）

"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

```
mav = PX4MavCtrl.PX4MavCtrler(1) # 创建飞机控制实例
mav.InitMavLoop() # 初始化 Mavlink 监听程序，读取飞机数据
mav.initOffboard() # 进入 Offboard 模式
mav.SendMavArm(True) # 解锁飞控
mav.SendPosNED(0, 0, -10, 0) # 发送 10 米高的位置控制指令
```

4) UE 控制

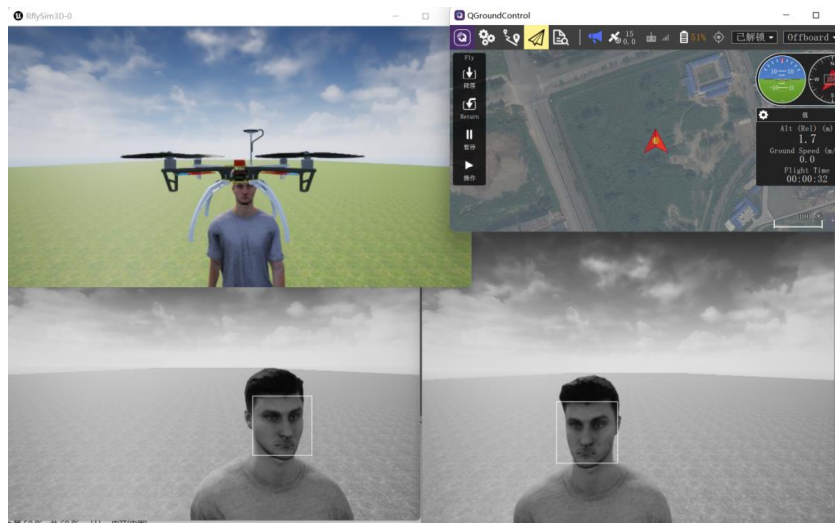
接口详细使用方法见: [UE4CtrlAPI.py](#)

```
ue = UE4CtrlAPI.UE4CtrlAPI() # 创建 UE 控制实例
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) # 发送指令，设置 UE4 最大刷新频率 30Hz，
同时也是取图频率
ue.sendUE4Cmd('RflyChange3DModel 100 16') # 发送命令将 copterID=100 的
对象（刚刚创建的人）更改为行走风格
ue.sendUE4PosNew(3, vehicleType=809, PosE=[2, 0, -2]) # 创建一个物
体，id 号为 3，类型为 809，位置为[2, 0, -2]
ue.sendUE4ExtAct(3, ActExt=[1, 0.8, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]) # 外部动作信息，用于指定飞行器执行的外部动作。这是一个包含 16 个元素的列表，每个元素表示一个外部动作的值。
```

5) 其余代码说明

```
timeInterval = 1/30.0 # 以 30hz 的频率进行控制
lastTime = lastTime + timeInterval # 设置每一帧的处理结束时间
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



3. 文件目录

例程目录：[\[安装目录\]\RflySimAPIs\8.RflySimVision\1.BasicExps\1-VisionCtrlDemos\7-ManDetect\](#)

文件夹/文件名称	说明
ManDetect3HITL.bat	硬件在环一键启动脚本
ManDetect3SITL.bat	软件在环一键启动脚本
ManDetect3.py	双目视觉人脸识别例程
Config.json	视觉传感器配置文件
Python38Run.bat	Python 环境启动脚本

4. 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 工具链		
3	VS Code		可选

①：推荐配置请见：<https://rflysim.com>

5. 实验步骤

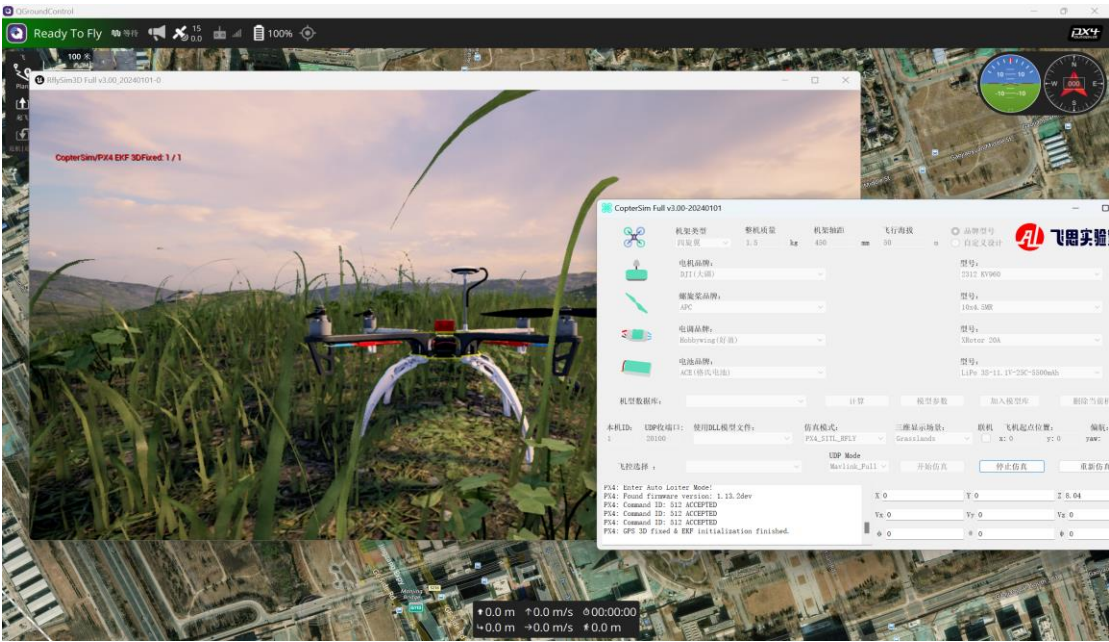
5.1 必做实验：Windows 取图控制

Step 1：开启仿真

双击运行“ManDetect3SITL.bat”或“ManDetect3HITL.bat”开启软/硬件在环仿真。

Config.json	2023/10/24 10:19	JSON 源文件	1 KB
ManDetect3.py	2023/10/27 10:53	Python 源文件	4 KB
ManDetect3HITL.bat	2024/1/22 11:44	Windows 批处理文件	5 KB
ManDetect3SITL.bat	2024/1/22 11:44	Windows 批处理文件	5 KB

注意：硬件仿真跟软件仿真步骤相同，只需把运行的脚本 ManDetect3SITL.bat 改成 ManDetect3HITL.bat 即可。



Step 2：设置搜索路径

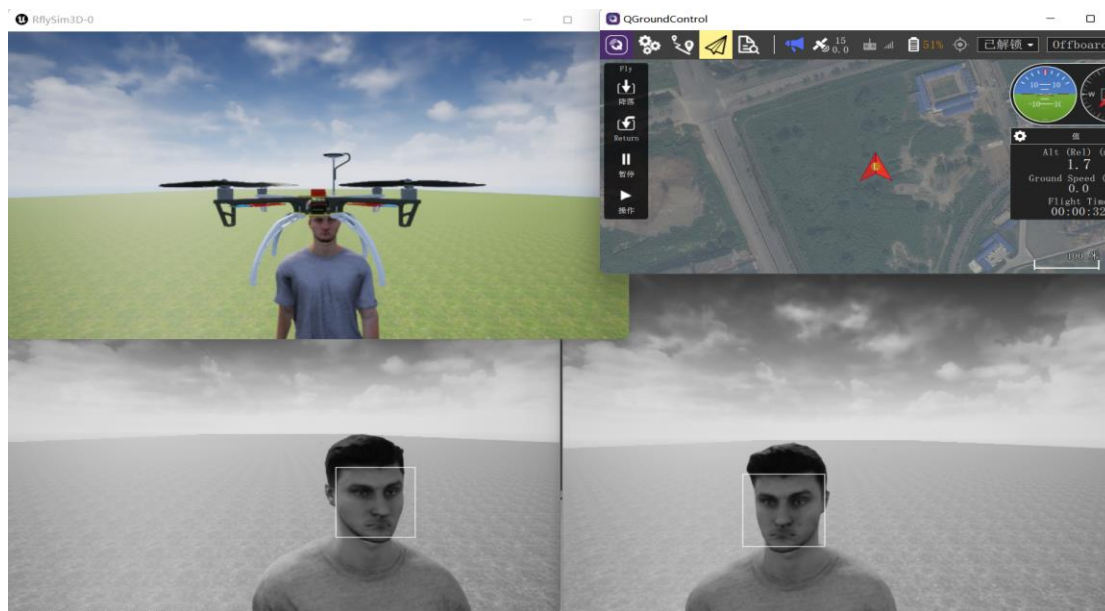
运行 PX4PSRfySimAPIs\RflySimSDK 目录下的 ReLabPath.py 文件。

Step 3：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 ManDetect3.py，输入 python ManDetect3.py 并点击调试。

Step 4: 观察结果

可看到 RflySim3D 中生成走动的人，并设置面对飞机，飞机起飞后开启人脸识别算法，双目框选出人脸。



Step 5: 结束仿真

在下图“ManDetect3SITL.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 ManDetect3.py 时，可使用 VS Code（或 Pycharm 等工具）来打开 ManDetect3.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 ManDetect3.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 X
> RflySimAPIs > 8.RflySimVision > 0.ApiExps > 1-UsageAPI > 0.VisionSenorAPI > 1.Came
8   ue = UE4CtrlAPI.UE4CtrlAPI()
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, 2
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

[1]. 无

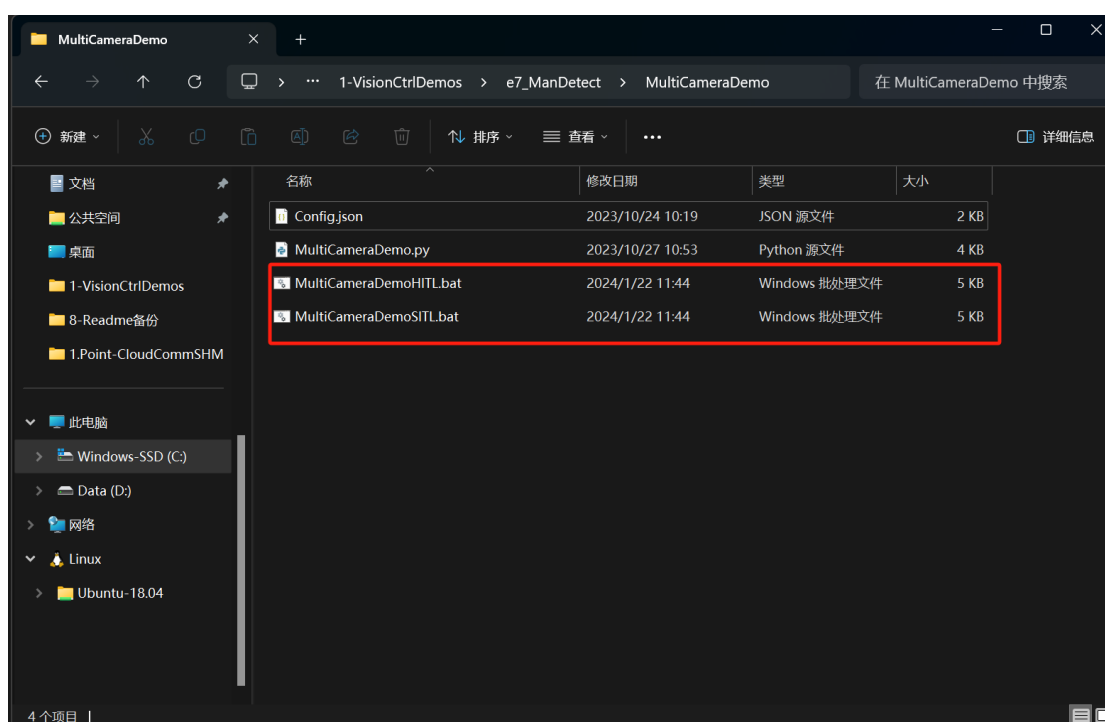
7. 常见问题

Q1: 无

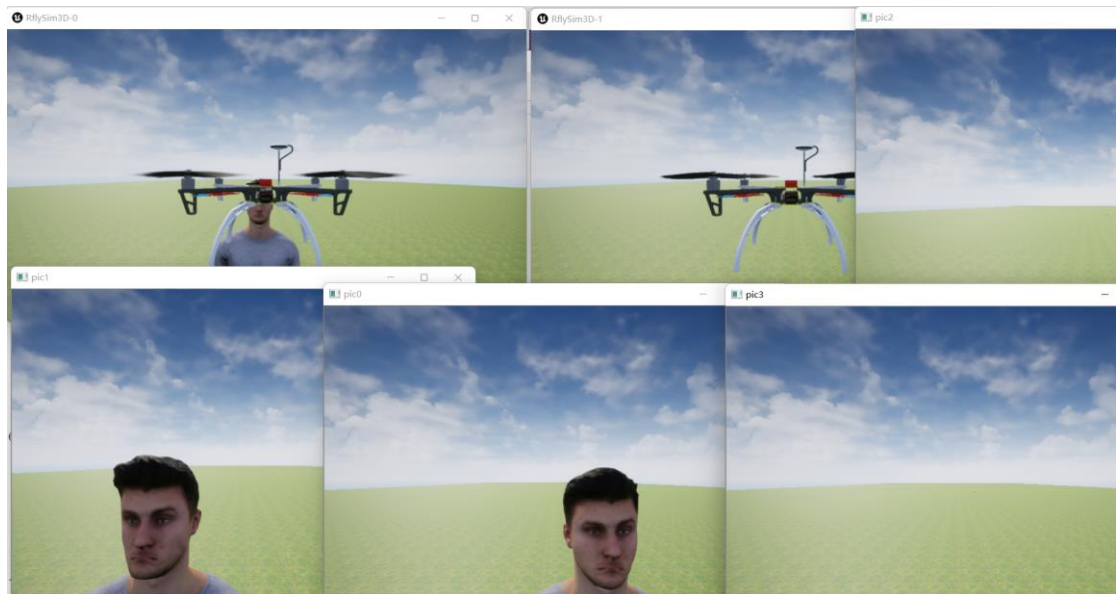
A1: 无

双目视觉人脸识别实验（扩展双飞机双目）

1. 在 Windows 资源管理器中，打开并进入 RflySimAPIs\8. RflySimVision\1. BasicExps\1-VisionCtrlDemos\e7_ManDetect\MultiCameraDemo 文件夹，运行“MultiCameraDemoSITL.bat”或“MultiCameraDemoHITL.bat”开启软/硬件在环仿真。



2. 然后运行“MultiCameraDemo.py”可以看到飞机起飞后，得到四个视角的图像。



注意：使用硬件在环仿真步骤一样，只需把打开的脚本文件 MultiCameraDemoSITL.bat 换成 MultiCameraDemoHITL.bat