

## 1、实验名称及目的

**双目视觉人脸识别实验：**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。

## 2、实验原理

首先通过 jsonLoad 函数可以读取传感器配置文件，其参数配置如下

“SeqID”代表第几个传感器。在本例程中，因为是双目相机所以在这里配置了两个传感器 0 和 1 分别代表第一个传感器和第二个传感器。

“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。

“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。

然后开启取图，再通过 ue.sendUE4Cmd()函数创建人并将其动作改为步行，sendUE4Cmd 是一系列使用命令对仿真进行修改的函数有如下命令可用：

1. RflyShowTextTime(String txt, float time)\\ 让 UE 显示 txt，持续 time 秒
2. RflyShowText(String txt)\\ 让 UE 显示 txt，持续 5 秒
3. RflyChangeMapbyID(int id)\\ 根据地图的 ID 切换 RflySim3D 场景地图
4. RflyChangeMapbyName(String txt)\\ 根据地图名切换 RflySim3D 场景地图
5. RflyChangeViewKeyCmd(String key, int num) \\ 与在 RflySim3D 中按一个 key + num 效果一致
6. RflyCameraPosAngAdd(float x, float y, float z,float roll,float pitch,float yaw) \\ 给摄像机的 位置与角度增加一个偏移值
7. RflyCameraPosAng(float x, float y, float z, float roll, float pitch, float yaw) \\ 设置摄像机的 位置与角度(UE 的世界坐标)

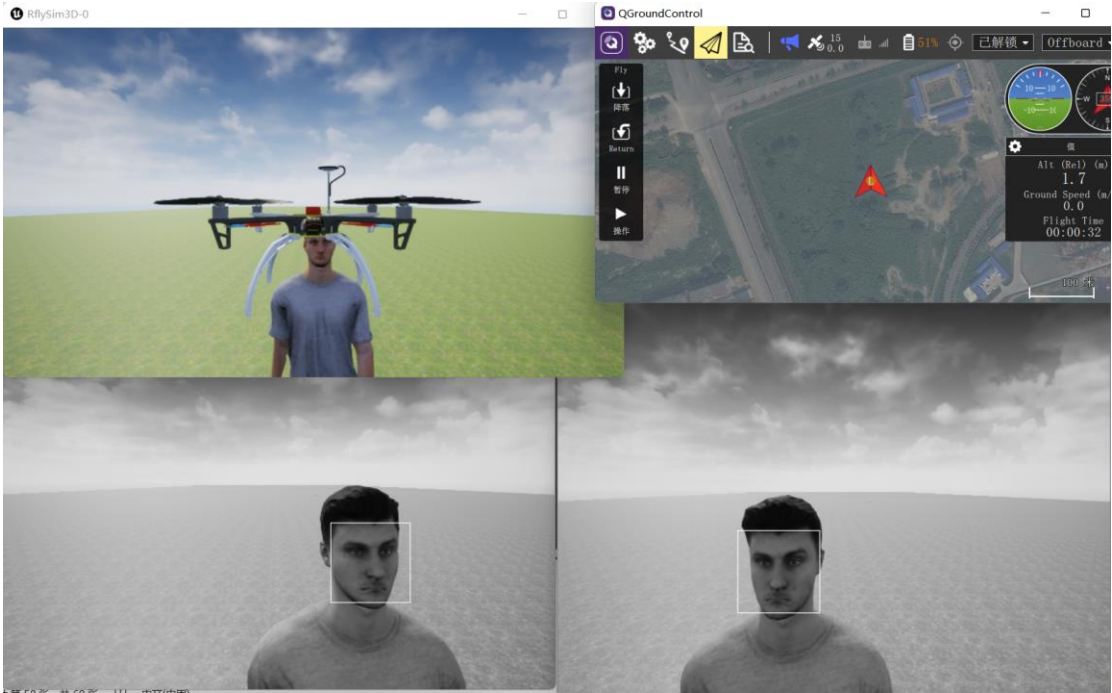
8. RflyCameraFovDegrees(float degrees) \ 设置摄像机的视域体 FOV 角度
9. RflyChange3DModel(int CopterID, int veTypes=0) \ 修改一个无人机的模型样式
10. RflyChangeVehicleSize(int CopterID, float size=0) [\修改一个无人机的缩放大小](#)
11. RflyMoveVehiclePosAng(int CopterID, int isFitGround, float x, float y, float z, float roll, float pitch, float yaw) \ 给无人机的位置与角度设置一个偏移值, isFitGround 设置无人机是否 适应地面
12. RflySetVehiclePosAng(int CopterID, int isFitGround, float x, float y, float z, float roll, float pitch, float yaw) [\设置无人机的位置与角度](#)
13. RflyScanTerrainH(float xLeftBottom(m), float yLeftBottom(m), float xRightTop(m), float yRightTop(m), float scanHeight(m), float scanInterval(m)) \ 扫描地形, 生成一个 png 的高 度图与 txt, CopterSim 程序会需要它才知道 UE 有哪些地形、以及它们的高程
14. RflyCesiumOriPos(double lat, double lon, double Alt) \ 根据经纬度修改 Cesium 的原点位置
15. RflyClearCapture()\ 清空抓取的图像
16. RflySetActuatorPWMs(int CopterID, float pwm1, float pwm2, float pwm3, float pwm4, float pwm5, float pwm6, float pwm7, float pwm8); \传入 8 个值, 并触发目标无人机的蓝图的 接口函数
17. RflySetActuatorPWMsExt(int CopterID, float pwm9, float pwm10, float pwm11, float pwm12, float pwm13, float pwm14, float pwm15, float pwm16, float pwm17, float pwm18, float pwm19, float pwm20, float pwm21, float pwm22, float pwm23, float pwm24);\传入 16 个 值, 并触发目标无人机的蓝图接口函数。该函数需要完整版才能有作用
18. RflyReqVehicleData(FString isEnabled);如果'isEnabled'不为 0, 则 RflySim3D 会开始发送所 有 Copter 的数据 (就是前面介绍的 reqVeCrashData)。
19. RflySetPosScale(float scale);\全局位置的缩放
20. RflyLoad3DFile(FString FileName);\加载并执行路径下的 TXT 脚本文件
21. RflyReqObjData(int opFlag, FString objName, FString colorStr);\请求获取三维场景中物体 的数据
22. RflySetIDLabel(int CopterID, FString Text, FString colorStr, float size);\设置一个 Copter 的头 顶 ID 显示内容 (默认显示 CopterID)
23. RflySetMsgLabel(int CopterID, FString Text, FString colorStr, float size, float time, int flag);\ 设置一个 Copter 头顶的 Message 显示的内容
24. RflyDelVehicles(FString CopterIDList);\删除一些 Copter (逗号是分隔符)
25. RflyDisableVeMove(FString CopterIDList, int disable);\拒接收指定 ID 的 Copter

r 的信息 （逗号是分隔符）

26. 除此之外，还有一些 UE 内置的命令可以使用，例如‘stat fps’可以显示当前帧率，‘t.Maxfps 60’可以设置最大帧率为 60。

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，最后通过通过死循环将图像显示在屏幕上。

3、实验效果



4、文件目录

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

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 <sup>①</sup>	1
2	RflySim 平台免费版及以上		
3	Visual Studio Code		

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

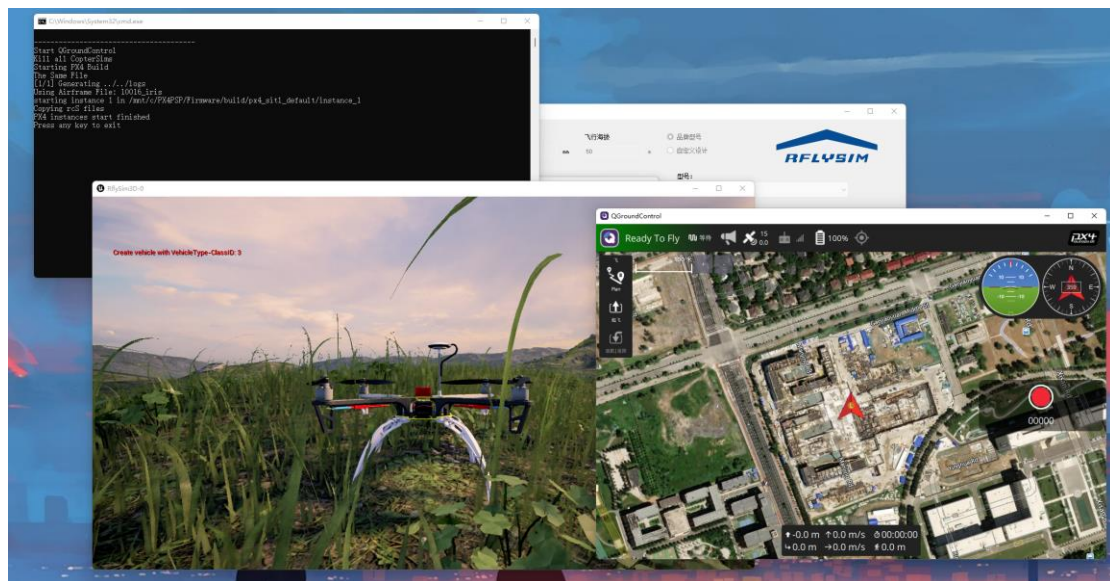
## 6、实验步骤

### Step 1:

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

 ManDetect3HITL.bat	2022/9/20 17:07	Windows 批处理...	5 KB
 ManDetect3SITL.bat	2022/9/20 17:07	Windows 批处理...	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。

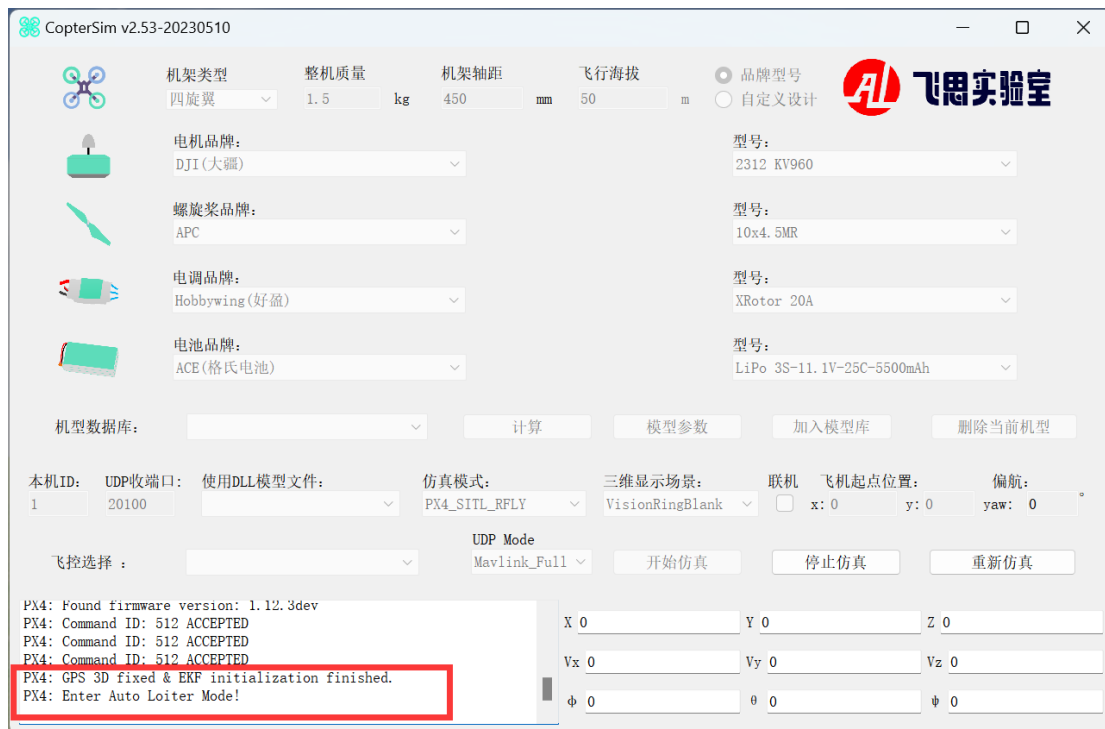


### Step 2:

运行 PX4SPRfySimAPIs\RflySimSDK 目录下的 ReLabPath.py 文件。

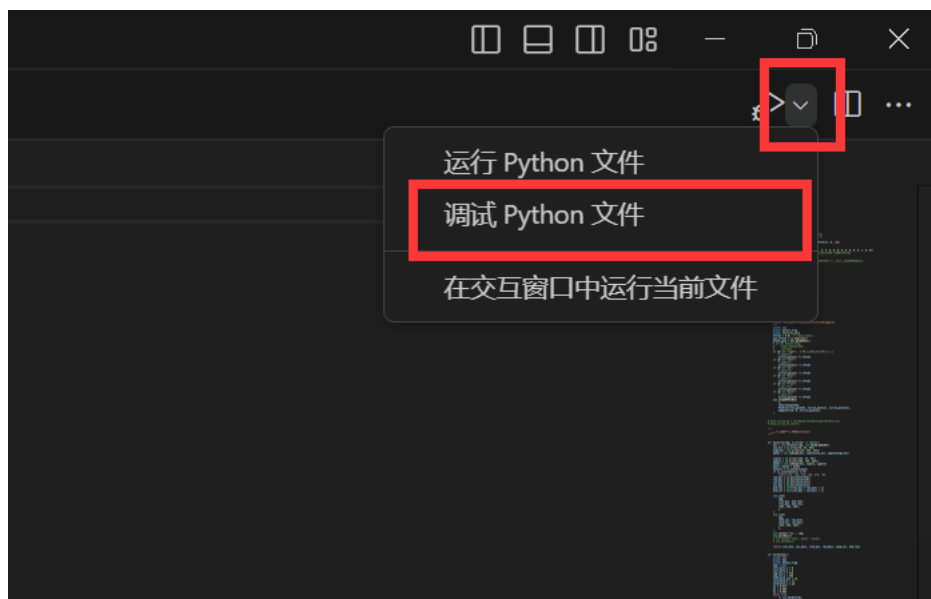
### Step 3:

等待 CopterSim 出现 PX4: GPS 3D fixed & EKF initialization finished.PX4: Enter Auto Loiter Mode!



## Step 4:

用 Visual Studio Code 打开 ManDetect3.py，并点击调试。



即可看到 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:

在下图 VS Code 中，点击“终止终端”，可以彻底退出脚本运行。



## 7、参考文献

[1]. 无

## 8、常见问题

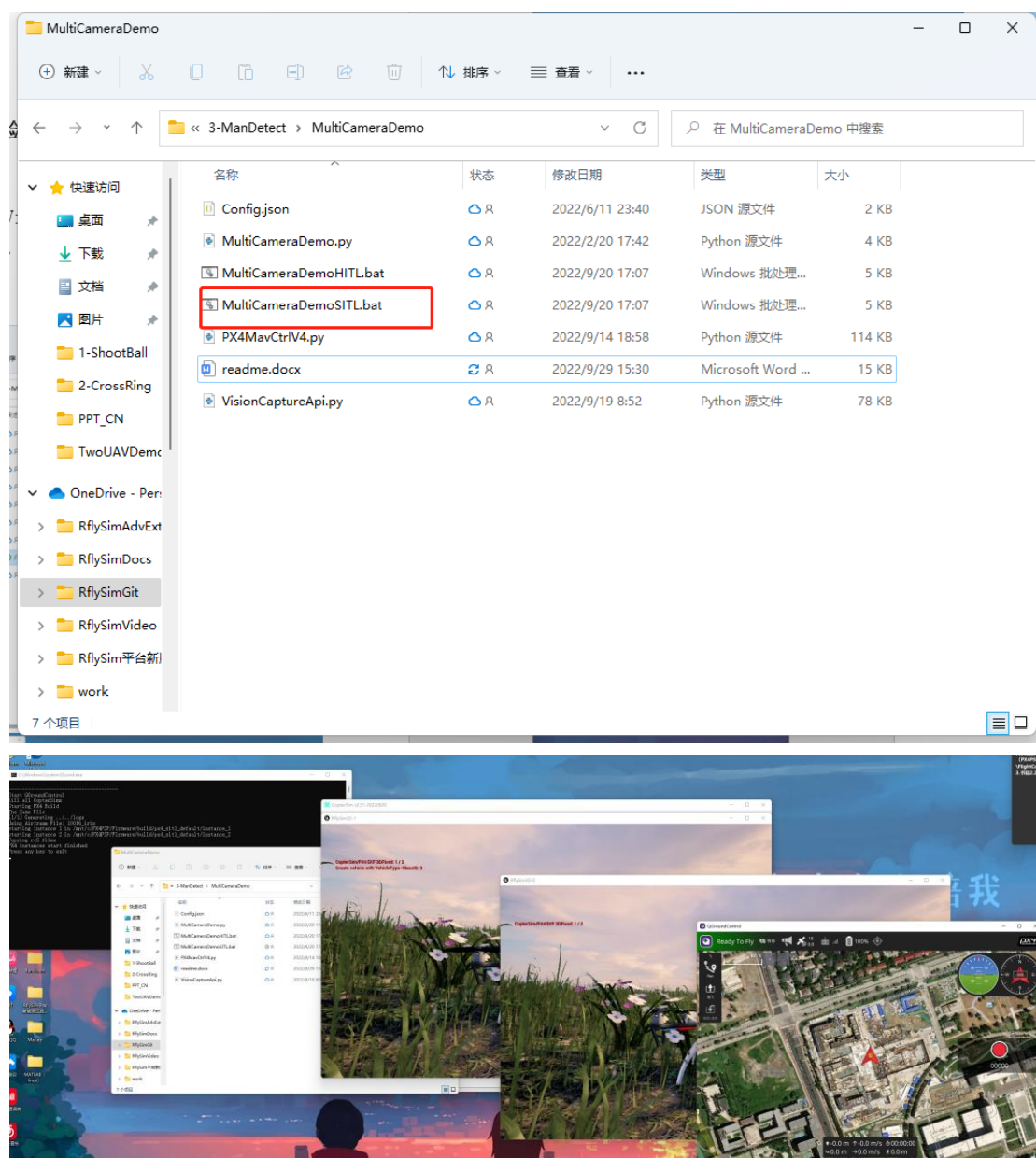
Q1: 无

A1: 无

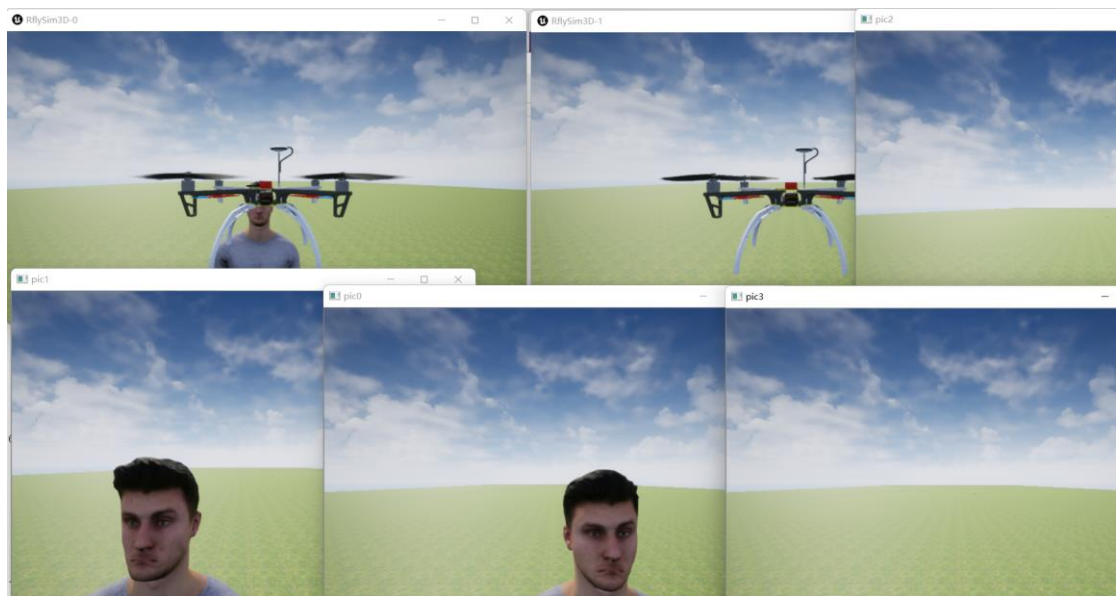


## 双目视觉人脸识别实验（扩展双飞机双目）

1. 在 Windows 资源管理器中，打开并进入 RflySimAPIs\PythonVisionAPI\2-VisionCtrlDemos\3-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 MultiCameraDemoHITL.bat