

# 1. 实验名称及目的

## 1.1 实验名称

双目摄像机系统的标定实验

## 1.2 实验目的

通过运行 [BinocularCameraCalib4.py](#)，演示改变棋盘的位置和姿态，用于双目摄像机系统的标定。可以存储图像和校准相机。

## 1.3 关键知识点

函数解释：向局域网内所有 RflySim3D 发送一个 Copter 的数据，如果不存在该 CopterID 的物体，那么会创建一个这样的物体。其中 vehicleType 表示该 Copter 的样式，PosE 表示该 Copter 的位置（米，北东地），AngEuler 表示该 Copter 的欧拉角（弧度，roll,pitch,yaw），MotorRPMSMean 表示 8 位执行器数据的平均值(8个执行器的值相同)。

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），死循环与 cv 库将获取的图像进行灰度处理并显示在屏幕上。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.sendUE4Cmd('RflyChangeMapbyName GrassLands') \#更新地图场景, 场景类型为GrassLands'
4
5 vis.sendUE4Cmd('r.setres 720x405w',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
6
7 vis.sendUE4Cmd('t.MaxFPS 30',0) \# 设置UE4最大刷新频率, 同时也是取图频率
8
9 vis.jsonLoad() \# 加载Config.json中的传感器配置文件
10
11 isSuss = vis.sendReqToUE4() \# 向RflySim3D发送取图请求
12
13 vis.startImgCap(True) \# 开启取图, 并启用共享内存图像转发
14
15 vis.hasData[i] \# 图片i数据是否更新
16
17 vis.Img[i] \# 图片i数据 (像素矩阵)
18
19 pic1=cv2.cvtColor(img1, cv2.COLOR_BGR2GRAY) \#将名为img1的彩色图像转换为灰度图像pic1
20
21 cv2.imshow('Img'+str(i),vis.Img[i]) \# 显示图片i图像
```

## 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":[0,0,0,0,0,0,0]: 传输模式为0共享内存机制, 因此本例程只能运行在Windows环境下。
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) 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.sendUE4Pos(1,0,0,[0,0,-8.086-1.5],[0,0,0]) \# 创建一个id号为1的物体, 位置为[0,0,-8.086
```

## 2. 实验效果

演示改变棋盘的位置和姿态，用于双目摄像机系统的标定。

## 3. 文件目录

例程目录：

[安装目录]\RflySimAPIs\8.RflySimVision\0.ApiExps\3-VisionAPI\0.BinocularCameraCalib

文件夹/文件名称	说明
<a href="#">BinocularCameraCalib4.bat</a>	双目摄像机系统的标定一键启动脚本
<a href="#">BinocularCameraCalib4.py</a>	Python实验代码
Config.json	视觉传感器配置文件
<a href="#">Python38Run.bat</a>	Python环境启动脚本

## 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\_fm4-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

双击运行 [BinocularCameraCalib4.bat](#) 一键启动脚本，会自动打开RflySim3D仿真平台。



#### Step 2: 运行控制程序

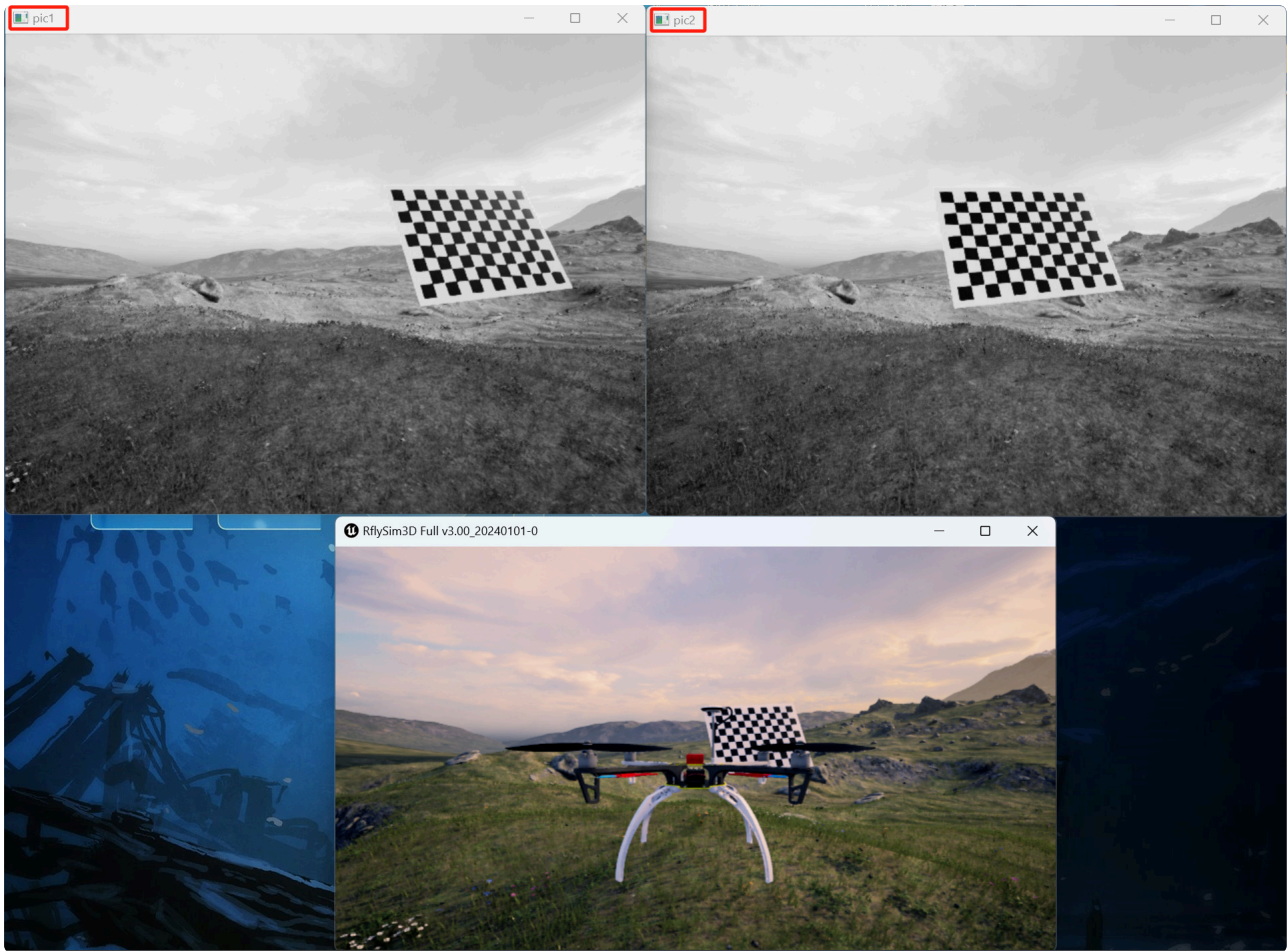
在文件夹下，双击 [Python38Run.bat](#)，打开集成好的python环境，在该环境下运行 [BinocularCameraCalib4.py](#) 文件，输入 `python BinocularCameraCalib4.py`

```
C:\Windows\system32\cmd.e...
Python3.8 environment has been set with openCV+pymavlink+numpy+pylog 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\3-VisionAI\0_BinocularCameraCalib>python BinocularCameraCalib4.py
```

#### Step 3: 观察结果

可对双目摄像机进行标定。



## Step 4: 结束仿真

在下图“[BinocularCameraCalib4.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\e3.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[BinocularCameraCalib4.py](#)时，可使用VS Code（或Pycharm等工具）来打开[BinocularCameraCalib4.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[BinocularCameraCalib4.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。

```
> 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