

# 1. 实验名称及目的

## 1.1. 实验名称

屏幕截图接口、撞击小球实验

## 1.2. 实验目的

双击 `ShootBall3SITL.bat`，后会打开一个 CopterSim 飞机的仿真闭环，同时打开两个 RflySim3D窗口，通过平台接口进行图像的获取，一个用于显示前置摄像头，一个用于全局观察。运行“`ShootBall3.py`”，开始视觉处理，然后控制无人机撞向小球。

## 1.3. 关键知识点

本实验与以往的撞击小球实验不同点在于没有使用取图接口选择使用屏幕截图的方式与UE4内部共享内存传图不同的是，屏幕截图取图方式一个窗口只能一路图像，图像为最终渲染效果，在例程中首先通过 `sca.getHwndInfo()` 获取句柄的窗口信息，然后通过 `sca.moveWd()` 将窗口始终保持在最顶层，再通过 `sca.getCVImg()` 从 RflySim3D窗口的工作区获取图像，再通过处理图像得到无人机速度控制命令，从而实现撞击小球。关键代码解析如下：

### 1) PX4MavCtrlV4接口说明

```
1 | img_bgr=sca.getCVImg(ImgInfo1) \# 获取图像
2 |
3 | window_hwnds = sca.getWndHandles() \# 获取所有 UE4/RflySim3D 窗口的句柄
4 |
5 | xx,yy=sca.moveWd(window_hwnds[0],0,0,True) \# 将 UE4 窗口 0移至所需位置并保持顶部
6 |
7 | ImgInfo1 = sca.getHwndInfo(window_hwnds[0]) \# 获取第一个窗口(索引=0)的信息
8 |
9 | cv2.moveWindow("dilated",0,yy) \#移动窗口到屏幕上的指定位置,(0,yy)窗口左上角的坐标
10 |
11 | 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":[0,0,0,0,0,0,0,0] : 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) 飞机控制指令

```

1 | mav = PX4MavCtrl.PX4MavCtrl(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.SendVelFRD(ctrl[0], ctrl[1], ctrl[2], ctrl[3]) \#设置飞机的速度, 分别是x、y、z以及yaw角

```

### 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('RflyChangeMapbyName VisionRingBlank') \#更新地图场景, 场景类型为VisionRing
4 |
5 | ue.sendUE4Cmd('RflyChangeViewKeyCmd B 1', 0) \# 将目标车辆改为copterID=1的车辆
6 |
7 | ue.sendUE4Cmd('RflyCameraPosAng 0.3 0 0.05 0 0 0', 0) \#相机移动到与身体相关的位置[0.3, 0, 0]
8 |
9 | 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
10 |
11 | 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
12 |
13 | ue.sendUE4Pos(100, 152, 0, [3, 0, -2], [0, 0, 0]) \#创建一个物体, id号为100, 类型为152, 速度为0, 位置

```

## 5) 其余代码说明

```
1 def calc_centroid(img) \# 用于计算图像中红色区域的质心和面积
2
3 def controller(p_i) \# 根据检测到的红色球的位置生成控制Pixhawk的速度命令
4
5 def procssImage() \# 处理图像以生成车辆的速度控制命令
6
7 def sat(inPwm,thres=1) \# 用于限制速度命令的最大值
8
9 timeInterval = 1/30.0 \# 以30hz的频率进行控制
10
11 lastTime = lastTime + timeInterval \# 设置每一帧的处理结束时间
12
13 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

运行“[ShootBall3.py](#)”，开始视觉处理，然后控制无人机撞向小球。

## 3.文件目录

例程目录：

[\[安装目录\]\RflySimAPIs\8.RflySimVision\1.BasicExps\1-VisionCtrlDemos\5\\_ScreenCapAPI\1-ShootBall](#)

文件夹/文件名称	说明
<a href="#">ShootBall3HITL.bat</a>	硬件在环一键启动脚本
<a href="#">ShootBall3SITL.bat</a>	软件在换一键启动脚本
<a href="#">ShootBall3.py</a>	撞击小球例程
<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\_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控制

### Step 1：开启仿真

双击运行“ShootBall3SITL.bat”文件开启软件在环仿真系统。也可插入飞控，并运行硬件在环仿真脚本“ShootBall3HITL.bat”，输入串口号来开启HITL仿真。

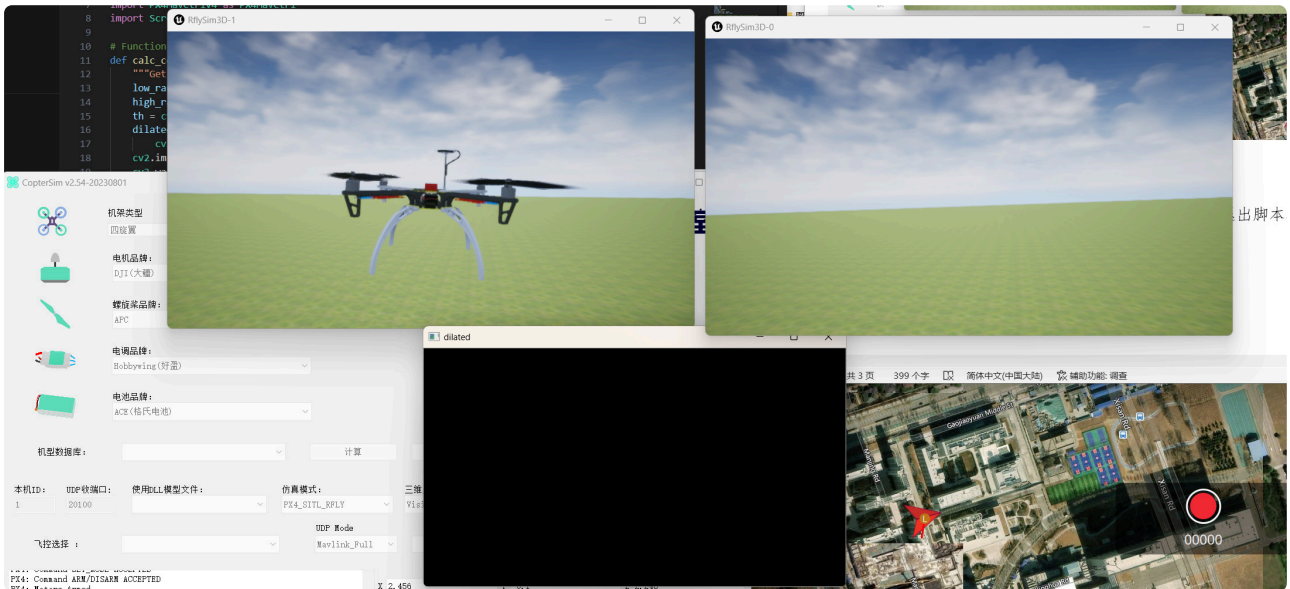


## Step 2: 运行控制程序

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 `python ShootBall3.py`

## Step 3: 观察结果

可以看到场景切换到草地场景，观察到下面现象。



## Step 4: 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。

## 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ShootBall3SITL.py时，可使用VS Code（或Pycharm等工具）来打开ShootBall3SITL.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ShootBall3SITL.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.Camera
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，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