

# 1. 实验名称及目的

## 1.1 实验名称

多目相机实验取图实验

## 1.2 实验目的

通过python接口获取RGB、灰度、深度三个相机图像。

注意：本实验只支持Windows下Python环境运行。不支持在WSL以及虚拟机下运行。

## 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）。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
10
11 vis.Img[i] \# 图片i数据（像素矩阵）
12
13 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, 1和2的三个视觉传感器
2 |
3 | "TypeID":1和"TypeID":2和"TypeID":3 : 传感器类型为RGB彩色图像以及深度图像以及灰度图
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], "SensorPosXYZ":[0.3, 0.15, 0]和"SensorPosXYZ":[0.3, 0, 0] : 三

```

### 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.SendPosNED(0, 0, -10, 0) \# 发送10米高的位置控制指令

```

### 4) 实时修改相机参数 (姿态位置等)

```

1 | vs = vis.VisSensor[0] \#获取第0号相机基本参数# 修改其中的可变部分, 只修改需要改变的部分即可
2 |
3 | vs.TargetCopter=1 \#修改视角绑定的飞机ID
4 |
5 | vs.TargetMountType=0 \# 修改视角绑定类型, 固连飞机还是地面
6 |
7 | vs.CameraFOV=90 \# 修改视角的视场角 (焦距), 可以模拟对焦相机
8 |
9 | vs.SensorPosXYZ=[0.3, -0.15, 0] \# 修改相机的位置, 可以调整相机初始位置
10 |
11 | vs.SensorAngEular=[0,0,0] \# 修改相机的姿态, 可以模拟云台转动
12 |
13 | vis.sendUpdateUEImage(vs) \# 发送更新数据

```

注意：相机宽高、相机类型等数据没法实时修改。

### 5) 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
```

## 2. 实验效果

本实验中，Json定义了RGB、灰度、深度三个相机，并实时显示图像。控制飞机起飞，以及发送相机更新数据。

## 3. 文件目录

例程目录：

[\[安装目录\]\RflySimAPIs\8.RflySimVision\0.ApiExps\1-UsageAPI\1.ImgSensorAPI\1.MutCameraImageGet](#)

文件夹/文件名称	说明
<a href="#">VisionCapAPIDemo.bat</a>	一键仿真启动脚本
<a href="#">VisionCapAPIDemo.py</a>	python实验代码
Config.json	视觉传感器配置文件
<a href="#">Python38Run.bat</a>	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。

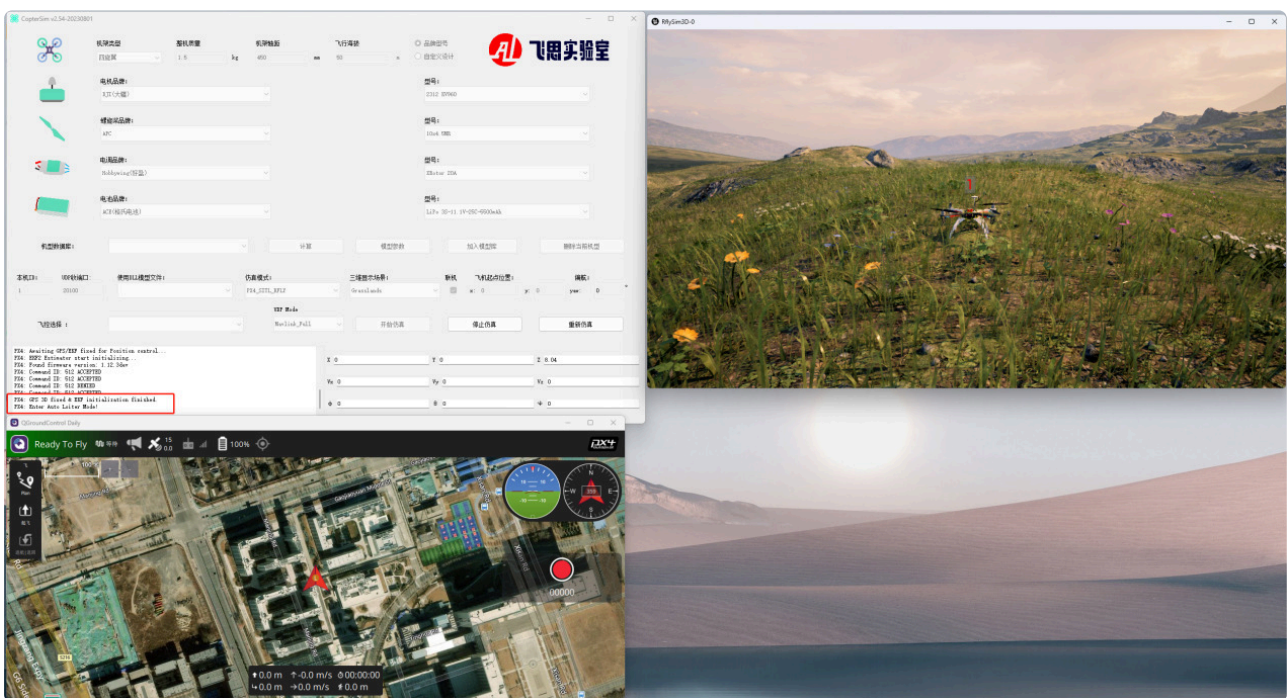
①：推荐配置请见：<https://rflsim.com/doc/zh/HowToInstall.pdf>

## 5. 实验步骤

### 5.1 必做实验：Windows取图控制

#### Step 1：开启仿真

双击运行 `VisionCapAPIDemo.bat`，启动 SITL 软件在环仿真。将会启动 1 个 QGC 地面站，1 个 CopterSim 软件且其软件下侧日志栏必须打印出 `GPS 3D fixed & EKF initialization finished` 字样代表初始化完成，并且 RflySim3D 软件内有 1 架无人机。



#### 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 `VisionCapAPIDemo.py` 文件，输入

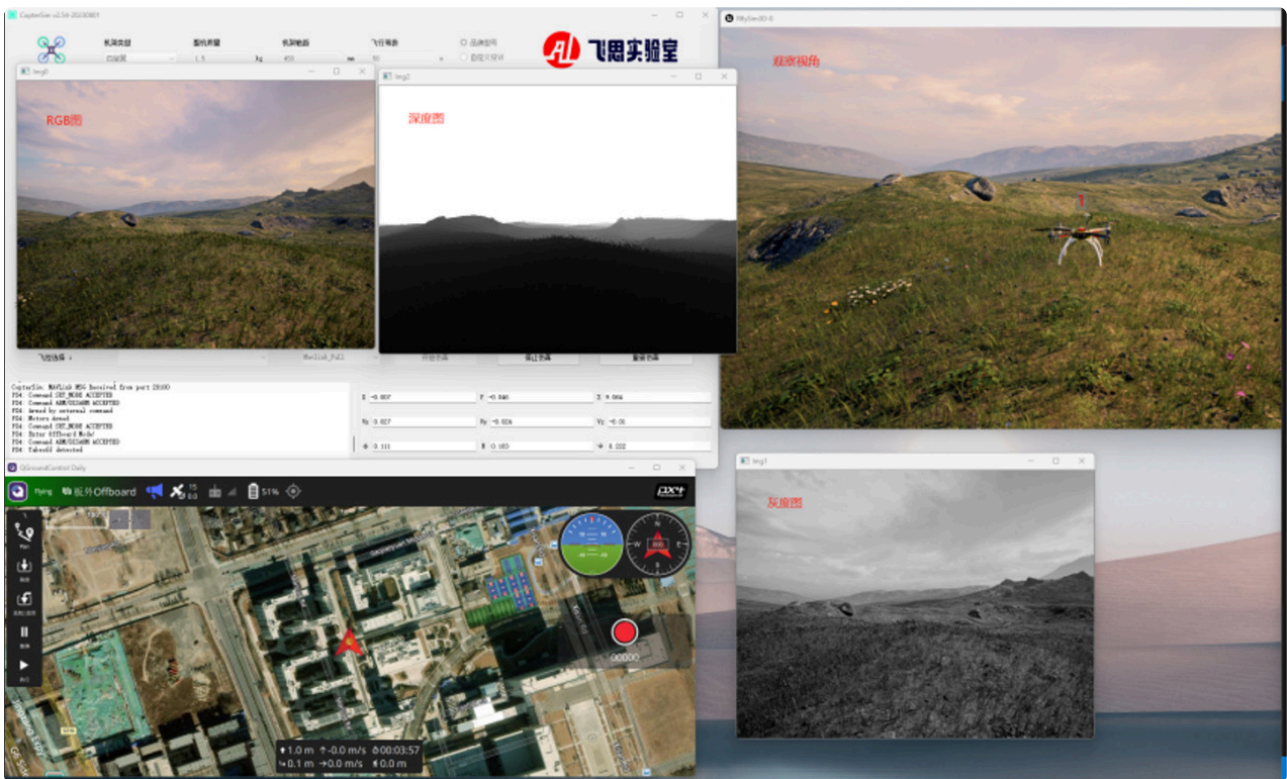
```
python VisionCapAPIDemo.py
```

```
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:\PX4PSP\RflySimAPIs\8.RflySimVision\0.ApiExps\1-UsageAPI\0.VisionSenorAPI\2.MutCameraImageGet>python VisionCapAPIDemo.py
```

### Step3: 观察结果

可以看到飞机成功起飞，并输出三幅图像。注意：在RflySim3D窗口按T键开启或关闭飞机轨迹记录功能，T+数字\*开启/更改轨迹粗细为\*号。Config.json文件创建了三个摄像头，一个RGB图摄像头，一个深度图摄像头，一个灰度图摄像头。可以看到如下图所示效果。



### Step 4: 结束仿真

在下图“[VisionCapAPIDemo.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[VisionCapAPIDemo.py](#)时，可使用VS Code（或Pycharm等工具）来打开[VisionCapAPIDemo.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[VisionCapAPIDemo.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，;
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