
1、实验名称及目的

无 CopterSim 取图实验：不启动 CopterSim 并获取相机数据。

2、实验原理

首先进行 Config.json 文件进行相机传感器的配置，配置参数如下解释：

“SeqID”代表第几个传感器。此处 0、1、2 分别表示第 1、2、3 个传感器（免费版只支持 2 个图）。

“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。

深度相机输出的数据是以 uint16 存储和传输的，它的数据范围是 0~65535。默认情况下，一个单位表示 1mm（由 otherParams[2]控制），也就是说最大范围是 0 到 65.535 米。

但是，数据范围并不代表相机的实际探测距离，还需要 otherParams[0]设置最小探测距离 otherParams[1]设置最大探测距离。otherParams[0]：深度相机的最小识别距离（单位米），如果深度距离小于本值，那么输出 NaN 对应 65535。otherParams[1]：深度相机的最大识别距离（单位米），如果深度距离大于本值，那么输出 NaN 对应 65535。otherParams[2]：深度相机 uint16 输出值的刻度单位（单位米），默认情况下深度值以毫秒为单位，因此需要填 0.001。注，默认值填 0 的话，会被替换为 otherParams[2]=0.001。实际深度值（单位米）= 深度图片值（uint16 范围）* otherParams[2]。

然后通过平台的接口 vis.sendReqToUE4() 向 RflySim3D 发送取图请求，并通过接口 vis.startImgCap()开启取图操作。并进行对图像的获取与操作。

VisionCaptureApi 是本平台的取图接口，包含了 json 加载，图像请求，图像转发等

classVisionSensorReq:#数据结构体，发送给 RflySim3D 的取图数据包

classimuDataCopter:#数据结构体，CopterSim 回传的 IMU 数据包

classSensorReqCopterSim:#数据结构包，发送给 CopterSim 请求传感器数据包

classVisionCaptureApi:#主接口类，实现了取图请求与接收

addVisSensor(vsr=VisionSensorReq()):#类函数，增加一个视觉传感器

sendReqToCopterSim(srcs=SensorReqCopterSim(),copterID=1):#类函数，发送数据包给 CopterSim，可以指定响应请求的 CopterSim 序号

sendImuReqCopterSim(copterID=1,IP='127.0.0.1',port=31000,freq=200):#类函数，发送数据包给 CopterSim 请求发送 IMU 数据（IP 和端口频率），并开始监听数据

sendUpdateUEImage(vs=VisionSensorReq(),windID=0):#发送请求给 RflySim3D，更新指定视觉传感器的参数、位置等，可指定接收的 RflySim3D 窗口号 windID

sendReqToUE4(windID=0):#将存储的视觉传感器列表发送给 RflySim3D，创建传感器，并检测是否创建成功，可指定接收的 RflySim3D 窗口号 windID

startImgCap(isRemoteSend=False):#开始接收图片并存储到 Img 列表，isRemoteSend 可配置共享内存图片是否向外通过 UDP 转发到其他系统中。

jsonLoad(ChangeMode=-1,jsonPath=''): #加载本地的 Json 文件，存储到视觉传感器列表中，ChangeMode 可覆盖 Json 中的 SendProtocol[0]传输模式，jsonPath 可指定 Json 文件地址。

3、实验效果

本实验中，通过 Json 文件定义了两个 RGB 相机，并在不启动 CopterSim 的情况下获取图像数据。

4、文件目录

文件夹/文件名称	说明
VisionCapAPIDemo.bat	启动仿真配置文件
VisionCapAPIDemo.py	python 实验代码
Config.json	视觉传感器配置文件

5、运行环境

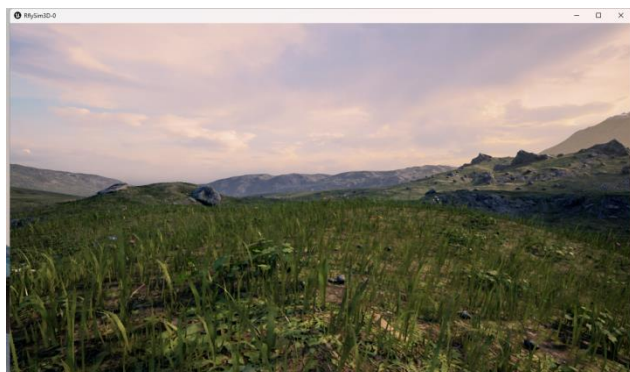
序号	软件要求	硬件要求	
		名称	数量(个)
1	Windows 10 及以上版本	笔记本/台式电脑 ^①	1
2	RflySim 平台免费版及以上		
3	Visual Studio Code		

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

6、实验步骤

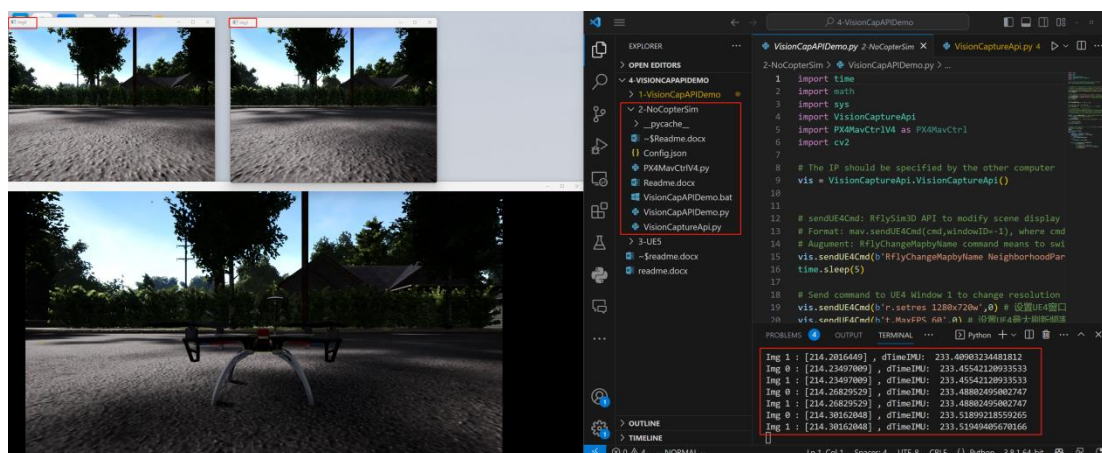
Step 1:

以管理员方式运行 VisionCapAPIDemo.bat，只启动了 1 个 RflySim3D 软件，且没有飞机出现，可以观察到本例程脚本未启动 CopterSim，如下图所示。



Step 2:

用 VScode 打开到本实验路径文件夹，运行 VisionCapAPIDemo.py 文件，启动仿真。并且按 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 文件创建了两个 RGB 摄像头，VScode 中也有数据打印，可看到如下图所示效果。



Step 3:

在下图“VisionCapAPIDemo.bat”脚本开启的命令行提示符 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。



Step 4:

在下图 VScode 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