

1. 实验名称及目的

1.1 实验名称

获取相机、物体、靶标中心精确三维位置方法实验

1.2 实验目的

通过调用平台接口获取相机、物体、靶标中心精确三维位置。

1.3 关键知识点

使用sendUE4Pos函数生成飞机与球，对于sendUE4Pos函数其有如下参数：

1. copterID：设置的 Copter 的 ID 。
2. vehicleType：设置的 Copter 的样式（在 xml中确定）本例程使用1表示Copter为多旋翼，152表示Copter为红色小球。
3. MotorRPMSMean：表示 8 位执行器数据的平均值(8 个执行器的值相同)。
4. PosE：表示设置的该 Copter 的位置（米，北东地）。
5. AngEuler：表示设置的该 Copter 的欧拉角（弧度，roll,pitch,yaw）。
6. windowID：影响发送目标端口，正常组播时取-1 即可，如果需要指定电脑上的指定的 RflySim3D 进行接收，则可以根据目标电脑IP与RflySim3D程序的窗口标题序号来设置。
（这个序号与上面提到的20010~20029 这20个端口是一一对应的关系，例如 RflySim3D-0 就是指的它在监听 20010，RflySim3D-3就是指的它在监听 20013）。

最后通过计算得到球与飞机的相对位置。

本实验主要是实现通过Python接口UE4CtrlAPI.py（见RflySimAPIs\RflySimSDK\ue目录）通过调用平台接口获取相机、物体、靶标中心精确三维位置。关键代码解析如下：

1) 飞机控制指令

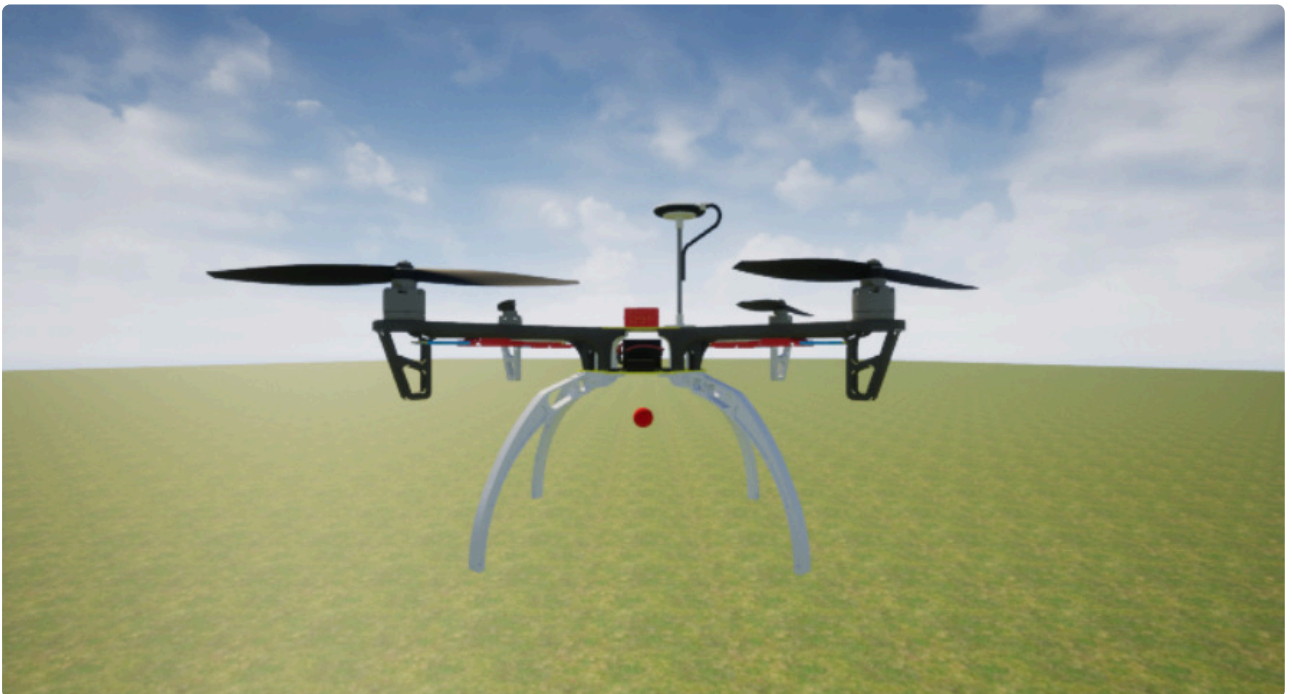
```
1 | mav = PX4MavCtrl.PX4MavCtrler(1) \# 创建飞机控制实例
```

2) 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，类型为VisionRingBlank
4 |
5 | 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
6 |
7 | 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
8 |
9 | ue.sendUE4Pos(1,0,0,[0,0,-8.086-1.5],[0,0,0]) \#创建物体，id号为1，类型为0，速度为0，位置为
10 |
11 | ue.sendUE4PosScale(100, 152, 0, InitTargePos, [0, 0, 0], [0.1, 0.1, 0.1]) \#与ue.send
```

2.实验效果



3. 文件目录

例程目录：

[安装目录]\RflySimAPIs\8.RflySimVision\0.ApiExps\3-VisionAPI\4.GetRelativePosDemo

文件夹/文件名称	说明
GetRelativePosDemo.bat	一键启动脚本。
GetRelativePosDemo.py	十小球图像例程。
Python38Run.bat	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://rflysim.com/doc/zh/1/Hardware.html>

4.2 硬件要求

笔记本/台式电脑① 1台。

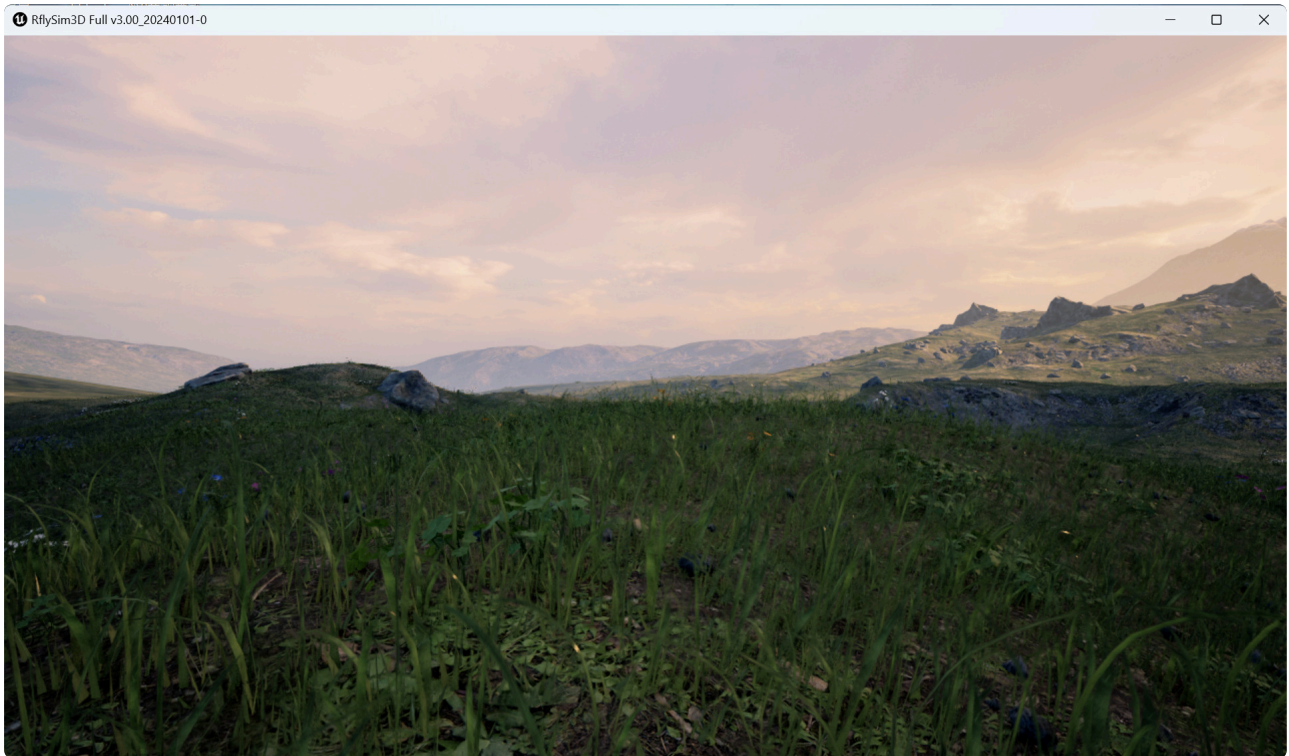
①：推荐配置请见：<https://rflysim.com/doc/zh/HowToInstall.pdf>

5. 实验步骤

5.1. 必做实验：Windows取图控制

Step 1: 开启仿真

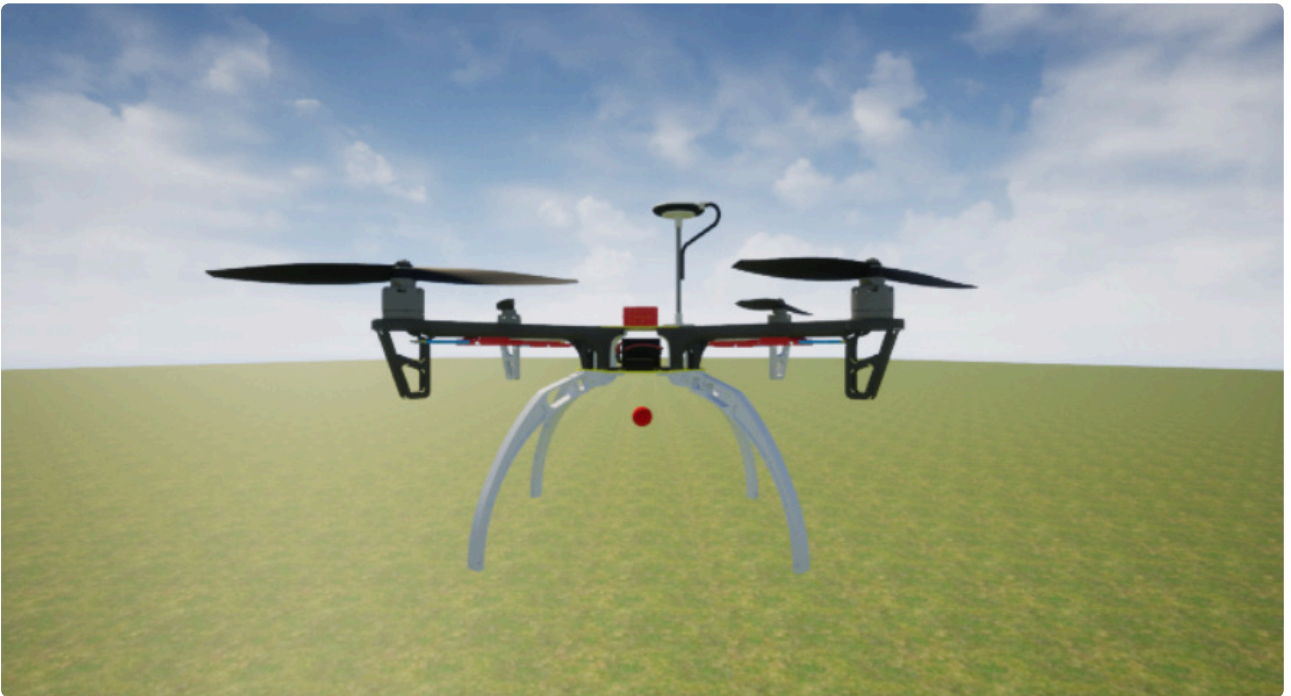
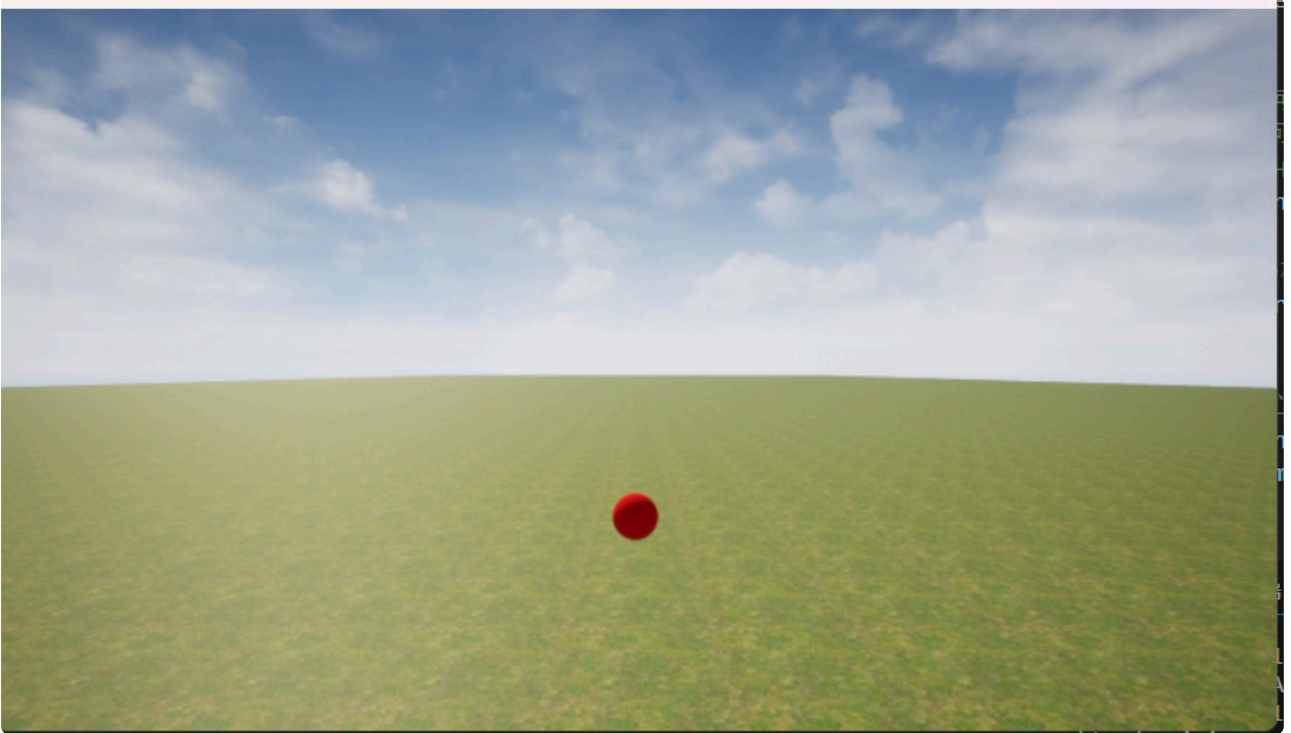
双击运行 [GetRelativePosDemo.bat](#)，打开RflySim3D仿真平台。



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，在该环境下输入 `python GetRelativePosDemo.py` 运行 [GetRelativePosDemo.py](#) 文件。

Step 3: 观察结果



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\3.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 [GetRelativePosDemo.py](#) 时，可使用VS Code（或Pycharm等工具）来打开 [GetRelativePosDemo.py](#) 文件，并阅读代码，修改代码，调试执行等。

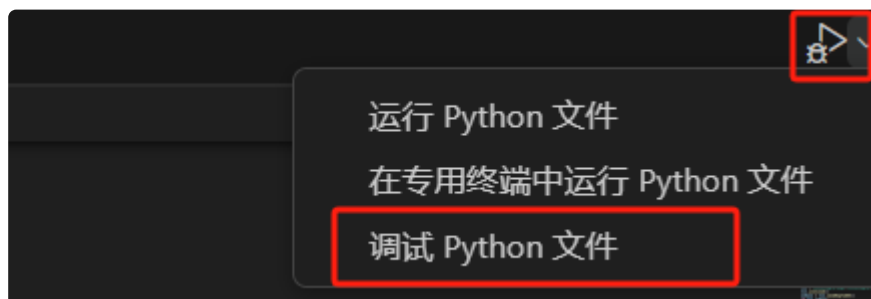
扩展实验：

- 请自行使用VS Code阅读 [GetRelativePosDemo.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，并
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
--
```

通过Visual Studio Code打开 [GetRelativePosDemo.py](#) 文件。点击运行此文件。进入调试模式，逐句执行语句。



- 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

[1] 无

| 7.常见问题

Q1: 无

A1: 无