

# 1. 实验名称及目的

## 1.1 实验名称

理论上推导 UE4 相机的理想模型实验

## 1.2 实验目的

在指定分辨率和视场角的情况下，可以快速计算焦距和内参矩阵、以及根据相机位置解算外参矩阵。

## 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）。然后通过自定义的目标找寻函数找到球的轮廓，并通过定义的一系列函数获取到球的位置、距离等一系列信息，通过这些信息计算出理想相机模型。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
14
15 gray = cv.cvtColor(image, cv.COLOR_BGR2GRAY) # 将输入的彩色图像转换为灰度图像
16
17 gray = cv.GaussianBlur(gray, (5, 5), 0) #对灰度图像进行高斯模糊处理, 以平滑图像并减少噪声
18
19 edged = cv.Canny(gray, 35, 125) # 使用 Canny 边缘检测算法检测图像边缘
20
21 (cnts, \_) = cv.findContours(edged.copy(), cv.RETR_LIST, cv.CHAIN_APPROX_SIMPLE)# 在
22
23 cv.drawContours(img1, [box1], -1, (0, 255, 0), 2) # 在图像 img1 上绘制 box1所表示的矩形轮廓
```

## 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的视觉传感器
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":[1,0,0,0,0,0,0,0] : 传输模式为1 : UDP网络传输模式 (图片使用jpeg压缩, 点云直传)。
8
9 "SensorPosXYZ":[ 0.3,0,0] : 相机分布。
```

### 3) 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=1 \# 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, 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。

### 4) UE控制

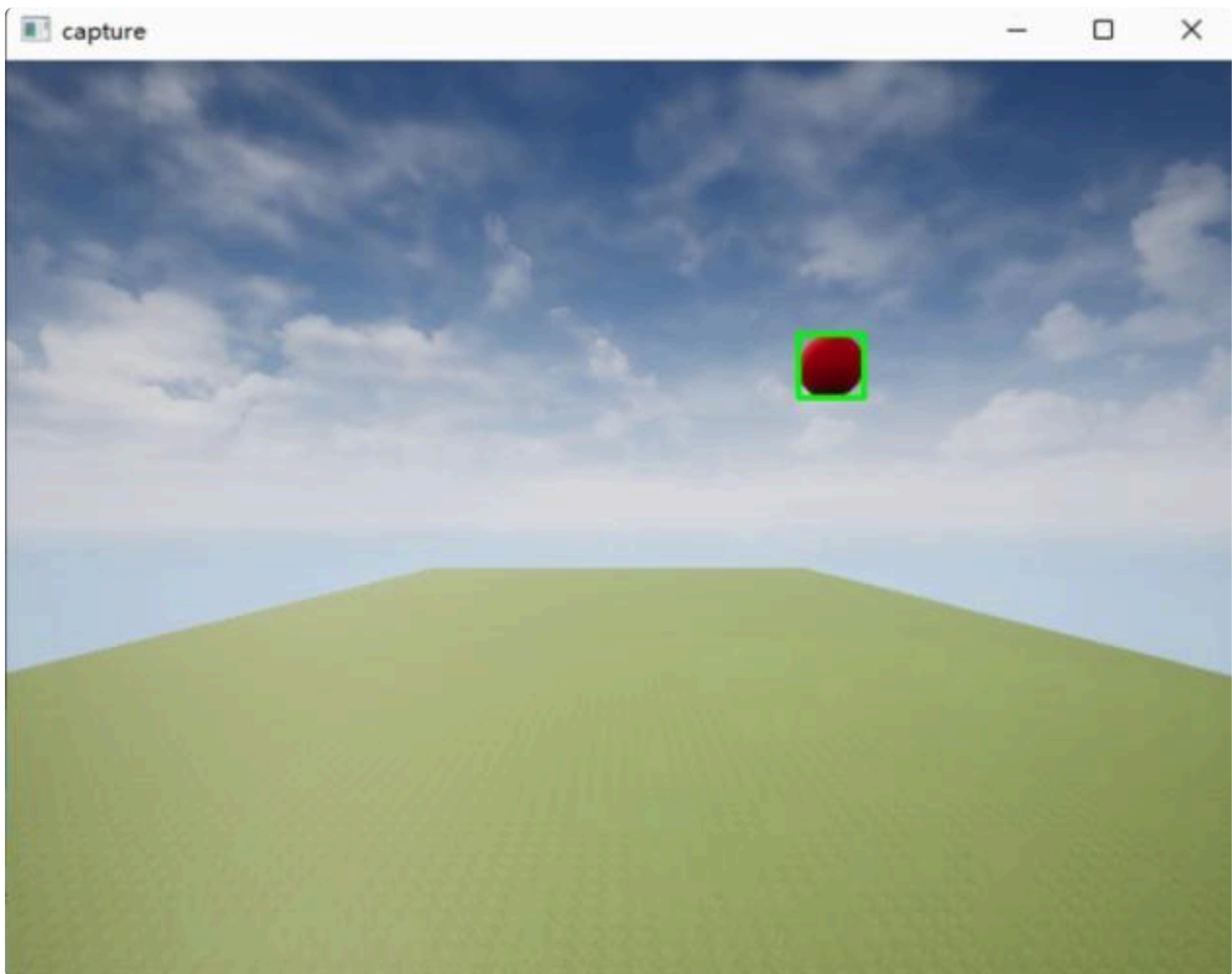
接口详细使用方法见：UE4CtrlAPI.py

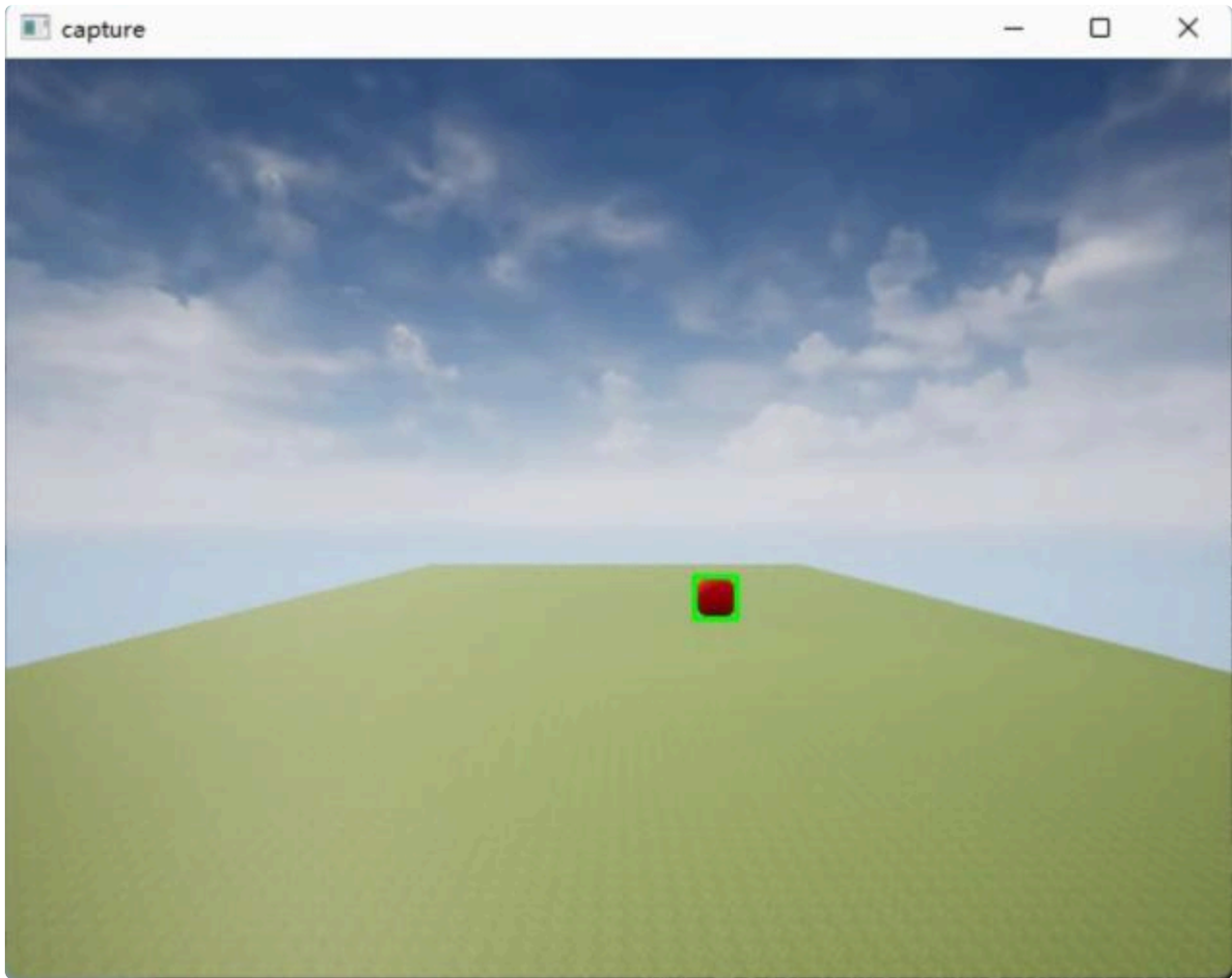
```
1 ue = UE4CtrlAPI.UE4CtrlAPI() \# 创建UE控制实例
2
3 ue.sendUE4Cmd('RflyChangeMapbyName VisionRingBlank') \#更新地图场景，类型为VisionRingBlank
4
5 ue.sendUE4Cmd('r.setres 1280x720w',0) \#发送指令，设置UE4窗口分辨率，注意本窗口仅限于显示，取
6
7 ue.sendUE4Cmd('t.MaxFPS 30',0) \#发送指令，设置UE4最大刷新频率30Hz，同时也是取图频率
8
9 ue.sendUE4Pos(1,0,0,[0,0,-8.086-1.5],[0,0,0]) \#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
```

## 5) 其余代码说明

```
1 def get_image_size(knownWidth, focalLength, distance) \# 图像大小计算函数
2
3 def procssImage() \# 处理图像并返回目标位置的标记
4
5 def rotate_matrix(roll, pitch, yaw) \#计算并返回绕指定的滚转角、俯仰角和偏航角的旋转矩阵
6
7 def get_pic_situation(relative_pos, px, py, focalLength) \#计算目标在图像中的位置
8
9 def change_cam( width, height, FOV) \# 改变相机配置
10
11 timeInterval = 1/30.0 \# 以30hz的频率进行控制
12
13 lastTime = lastTime + timeInterval \# 设置每一帧的处理结束时间
14
15 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





## 3. 文件目录

例程目录：

[\[安装目录\]\RflySimAPIs\8.RflySimVision\0.ApiExps\3-VisionAPI\3.CameraCalcDemo2](#)

文件夹/文件名称	说明
<a href="#">CameraCalcDemo.bat</a>	一键启动脚本
<a href="#">Calibration-pictest.py</a>	十小球图像例程
<a href="#">OneCameraCal.bat</a>	一键启动脚本
<a href="#">ChangeFOV.py</a>	FOV改变例程
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\_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[OneCameraCal.bat](#)一键启动脚本，会自动打开RflySim3D仿真平台。



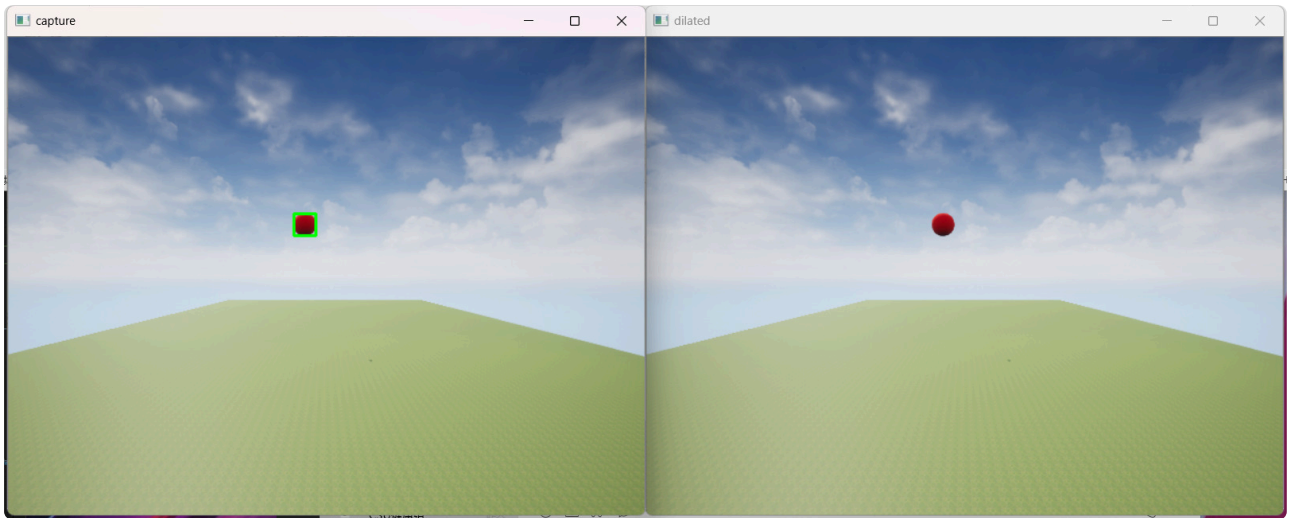
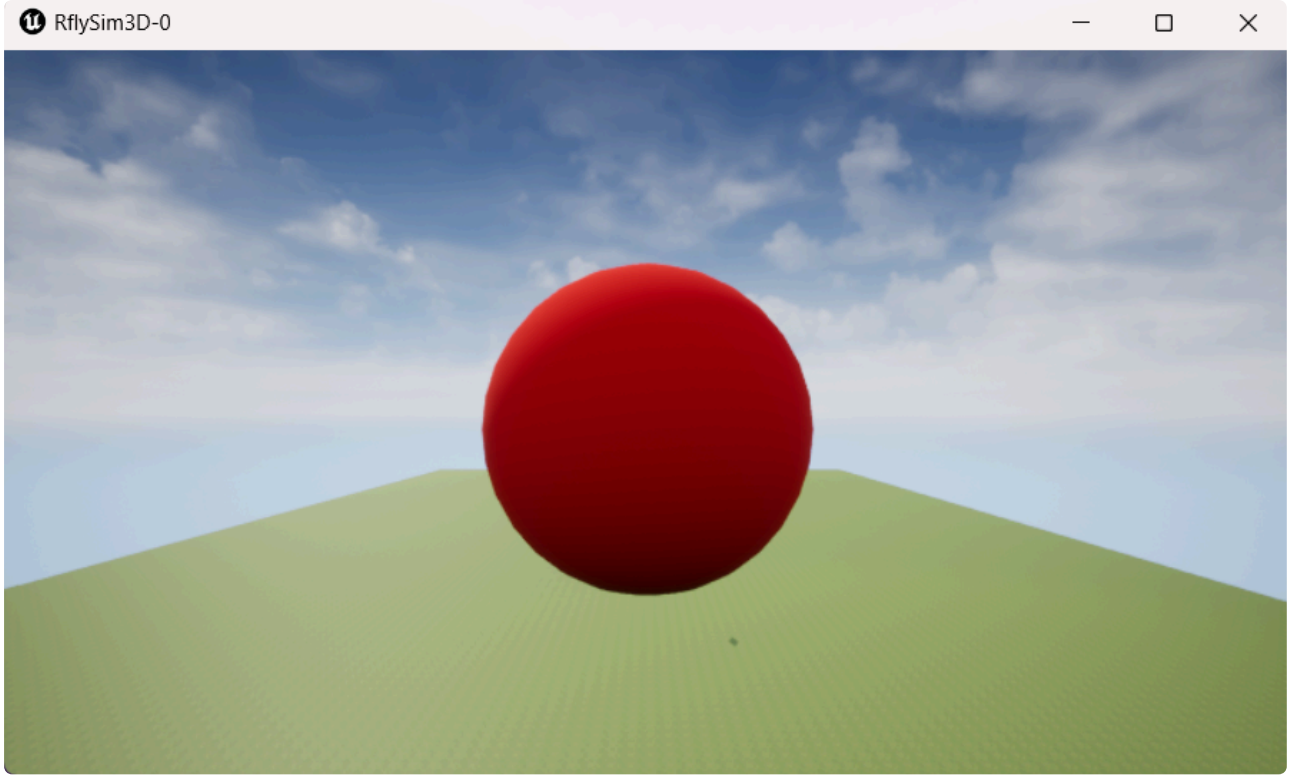
## Step 2: 运行控制程序

在文件夹下，双击 `Python38Run.bat`，打开集成好的python环境，在该环境下输入 `python Calibration-picctest.py` 运行 `Calibration-picctest.py` 文件。

```
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:\Users\uavcs\Desktop\demo\8.RflySimVision\0.ApiExps\3-VisionAI\3.CameraCalcDemo2>python Calibration-picctest.py
```

## Step 3: 观察结果

输出图像如下：



```
通过公式计算焦距预测的图像坐标为: [414.2760942760943, 176.43097643097644]
图像中的真实图像坐标为: [418.5, 173.0]
MATLAB标定结果预测的图像坐标误差为: 3.7709098337246734
通过公式计算焦距预测的图像坐标误差为: 5.441780851372033
MATLAB标定结果预测的小球图像直径为: 25.756434672496447
通过公式计算焦距预测的小球图像直径为: 25.381322520631375
图像中的实际小球直径为: 28.0
MATLAB标定结果预测的图像坐标为: [339.9338778217822, 210.4206974257425]
通过公式计算焦距预测的图像坐标为: [339.64356435643566, 210.8514851485148]
图像中的真实图像坐标为: [340.5, 208.5]
MATLAB标定结果预测的图像坐标误差为: 2.00239180030382
通过公式计算焦距预测的图像坐标误差为: 2.502591539834881
MATLAB标定结果预测的小球图像直径为: 31.959192361960525
通过公式计算焦距预测的小球图像直径为: 31.49374434591325
图像中的实际小球直径为: 33.0
MATLAB标定结果预测的图像坐标为: [289.2361715789474, 244.27275394736841]
通过公式计算焦距预测的图像坐标为: [289.6842105263158, 244.21052631578945]
图像中的真实图像坐标为: [288.0, 244.0]
MATLAB标定结果预测的图像坐标误差为: 1.2659047706689237
通过公式计算焦距预测的图像坐标误差为: 1.69731742069442
MATLAB标定结果预测的小球图像直径为: 21.26671512447537
通过公式计算焦距预测的小球图像直径为: 20.956990452762092
图像中的实际小球直径为: 20.0
MATLAB标定结果预测的图像坐标为: [335.29879680851064, 262.701440425532]
通过公式计算焦距预测的图像坐标为: [335.0759878419453, 262.37082066869306]
图像中的真实图像坐标为: [335.0, 263.0]
MATLAB标定结果预测的图像坐标误差为: 0.42239478249934526
通过公式计算焦距预测的图像坐标误差为: 0.6337513574481343
MATLAB标定结果预测的小球图像直径为: 24.588261156913315
通过公式计算焦距预测的小球图像直径为: 24.2301620771894
图像中的实际小球直径为: 24.0
MATLAB标定结果预测的图像坐标为: [289.0092369294606, 234.61030207468883]
通过公式计算焦距预测的图像坐标为: [289.46058091286307, 234.68879668049797]
图像中的真实图像坐标为: [288.0, 234.0]
MATLAB标定结果预测的图像坐标误差为: 1.1794184169142645
通过公式计算焦距预测的图像坐标误差为: 1.6148490548917973
MATLAB标定结果预测的小球图像直径为: 22.352448046729204
通过公式计算焦距预测的小球图像直径为: 22.02691095307182
图像中的实际小球直径为: 22.0
MATLAB标定结果预测的图像坐标为: [403.19510991735535, 276.6774140495868]
通过公式计算焦距预测的图像坐标为: [401.9834710743802, 276.1432506887052]
图像中的真实图像坐标为: [404.5, 277.0]
MATLAB标定结果预测的图像坐标误差为: 1.3441725421940256
通过公式计算焦距预测的图像坐标误差为: 2.6583711584136003
MATLAB标定结果预测的小球图像直径为: 21.536046199166318
通过公式计算焦距预测的小球图像直径为: 21.222399037392748
图像中的实际小球直径为: 22.0
```

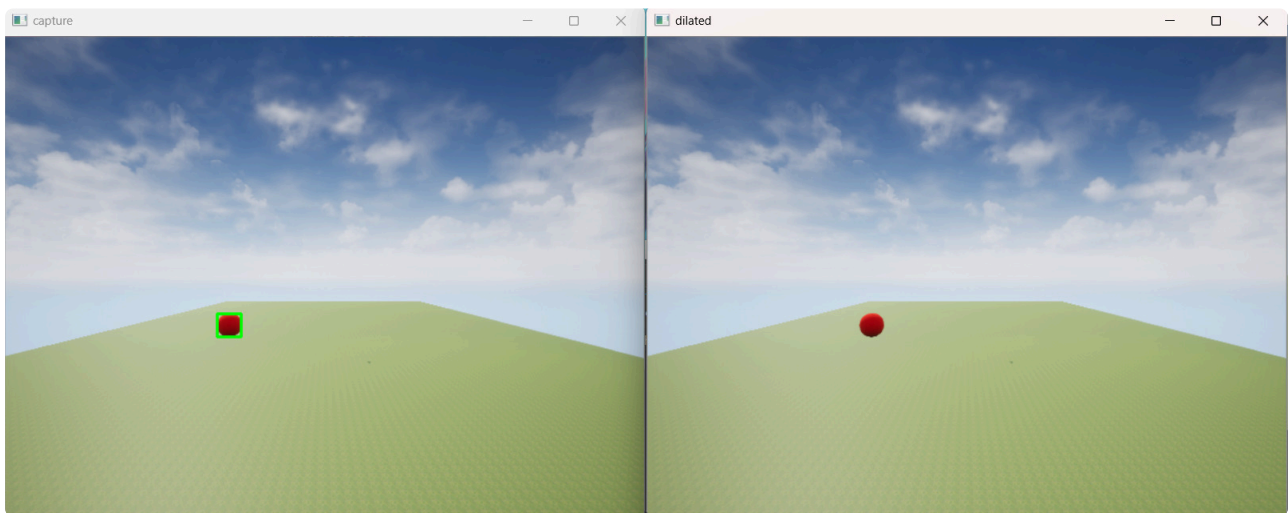
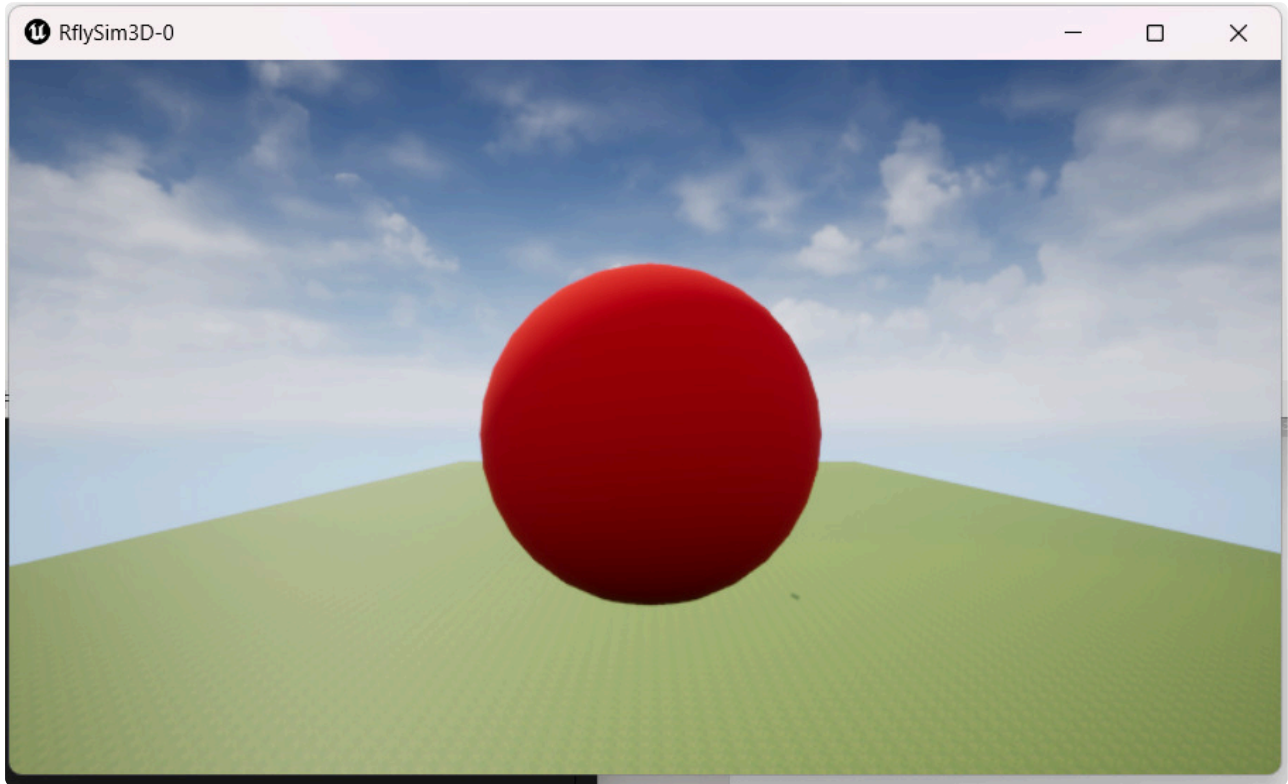
关闭所有程序!!!

## Step 4: 运行控制程序

双击运行 [CameraCalcDemo.bat](#)，然后在文件夹下，双击 [Python38Run.bat](#)，打开集成好的python环境，在该环境下输入 `python ChangeFOV.py` 运行 [ChangeFOV.py](#) 文件。

```
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:\Users\uavcs\Desktop\demo\8.RflySimVision\0.ApiExps\3-VisionAIAPI\3.CameraCalcDemo2>python ChangeFOV.py
```

## Step 5: 观察结果



检测按键来转换视场角的代码如下：

```

if keyboard.is_pressed('Q' or 'q'):
    px, py, CACULATION_FOCALLENGTH, MATLAB_FOCALLENGTH = change_cam(640, 480, 60)
if keyboard.is_pressed('E' or 'e'):
    px, py, CACULATION_FOCALLENGTH, MATLAB_FOCALLENGTH = change_cam(640, 480, 120)
if keyboard.is_pressed('W' or 'w'):
    px, py, CACULATION_FOCALLENGTH, MATLAB_FOCALLENGTH = change_cam(640, 480, 90)

```

在以上三个界面按“W”或“w”键，切换相机视场角为90°、由MATLAB标定图像焦距（MATLAB标定图像坐标括号后的数据）和公式计算的焦距（公式计算焦距预测图像坐标括号后的数据）可得焦距在320左右。

```

MATLAB标定结果预测的图像坐标为: [357.3202762219287, 250.29524861294584] 324.7293
通过公式计算焦距预测的图像坐标为: [356.776750330251, 250.14531043593132] 320
图像中的真实图像坐标为: [357.5, 250.0]
MATLAB标定结果预测的图像坐标误差为: 0.34564776847344475
通过公式计算焦距预测的图像坐标误差为: 0.7377026552633531
MATLAB标定结果预测的小球图像直径为: 21.297612431818123
通过公式计算焦距预测的小球图像直径为: 20.98743777719411
图像中的实际小球直径为: 20.0
MATLAB标定结果预测的图像坐标为: [334.73881156661787, 330.3346515373353] 324.7293
通过公式计算焦距预测的图像坐标为: [334.5241581259151, 329.0190336749634] 320
图像中的真实图像坐标为: [334.5, 332.0]
MATLAB标定结果预测的图像坐标误差为: 1.6823841613764203
通过公式计算焦距预测的图像坐标误差为: 2.9810642136743666
MATLAB标定结果预测的小球图像直径为: 22.880746027330556
通过公式计算焦距预测的小球图像直径为: 22.547514895470712
图像中的实际小球直径为: 24.0
MATLAB标定结果预测的图像坐标为: [343.1301596368715, 161.53887025139665] 324.7293
通过公式计算焦距预测的图像坐标为: [342.7932960893855, 162.68156424581002] 320
图像中的真实图像坐标为: [343.0, 159.5]
MATLAB标定结果预测的图像坐标误差为: 2.0430206638946995
通过公式计算焦距预测的图像坐标误差为: 3.1882718762489555
MATLAB标定结果预测的小球图像直径为: 21.989691331367368
通过公式计算焦距预测的小球图像直径为: 21.669437362250825
图像中的实际小球直径为: 23.0
MATLAB标定结果预测的图像坐标为: [399.641638540146, 196.86077912408757] 324.7293
通过公式计算焦距预测的图像坐标为: [398.48175182481754, 197.4890510948905] 320
图像中的真实图像坐标为: [401.0, 195.0]
MATLAB标定结果预测的图像坐标误差为: 2.3038326337337933
通过公式计算焦距预测的图像坐标误差为: 3.5407554596138664
MATLAB标定结果预测的小球图像直径为: 22.83137789045533
通过公式计算焦距预测的小球图像直径为: 22.49886574739546
图像中的实际小球直径为: 24.0

```

在以上三个界面按“E”或“e”键，切换相机视场角为120°，焦距在185左右。

```
MATLAB标定结果预测的图像坐标为: [307.658711775726, 253.91677182737283] 187.48254876875882
通过公式计算焦距预测的图像坐标为: [307.8384481111877, 253.7140904278096] 184.7520861406803
图像中的真实图像坐标为: [311.0, 254.0]
MATLAB标定结果预测的图像坐标误差为: 3.3423246291153585
通过公式计算焦距预测的图像坐标误差为: 3.174453437857106
MATLAB标定结果预测的小球图像直径为: 13.06488763652377
通过公式计算焦距预测的小球图像直径为: 12.874612927406321
图像中的实际小球直径为: 12.0
MATLAB标定结果预测的图像坐标为: [317.47328101389814, 244.04275037776299] 187.48254876875882
通过公式计算焦距预测的图像坐标为: [317.5100797016081, 243.98387247742707] 184.7520861406803
图像中的真实图像坐标为: [321.0, 244.0]
MATLAB标定结果预测的图像坐标误差为: 3.526978083534143
通过公式计算焦距预测的图像坐标误差为: 3.4899575622222265
MATLAB标定结果预测的小球图像直径为: 12.629512408000005
通过公式计算焦距预测的小球图像直径为: 12.445578426584854
图像中的实际小球直径为: 12.0
MATLAB标定结果预测的图像坐标为: [271.1731592522097, 232.48817834649378] 187.48254876875882
通过公式计算焦距预测的图像坐标为: [271.88426471127525, 232.59757918635003] 184.7520861406803
图像中的真实图像坐标为: [274.0, 231.5]
MATLAB标定结果预测的图像坐标误差为: 2.994582618303727
通过公式计算焦距预测的图像坐标误差为: 2.383488175398401
MATLAB标定结果预测的小球图像直径为: 15.133089366989928
通过公式计算焦距预测的小球图像直径为: 14.912693734248116
图像中的实际小球直径为: 15.0
MATLAB标定结果预测的图像坐标为: [328.4299707180197, 264.95271332533844] 187.48254876875882
通过公式计算焦距预测的图像坐标为: [328.3071981178363, 264.58930642879557] 184.7520861406803
图像中的真实图像坐标为: [333.5, 265.5]
MATLAB标定结果预测的图像坐标误差为: 5.099482289840765
通过公式计算焦距预测的图像坐标误差为: 5.2720540748398745
MATLAB标定结果预测的小球图像直径为: 16.695993247589637
通过公式计算焦距预测的小球图像直径为: 16.452835759596326
图像中的实际小球直径为: 17.0
```

在以上三个界面按“Q”或“q”键，切换相机视场角为60°，焦距在560左右。

```
MATLAB标定结果预测的图像坐标为: [430.9209501679708, 345.3188819776692] 562.4476463062763
通过公式计算焦距预测的图像坐标为: [429.3055170991674, 343.78503643759325] 554.2562584220408
图像中的真实图像坐标为: [452.5, 349.5]
MATLAB标定结果预测的图像坐标误差为: 21.980380787644695
通过公式计算焦距预测的图像坐标误差为: 23.888173759344877
MATLAB标定结果预测的小球图像直径为: 54.05741988032504
通过公式计算焦距预测的小球图像直径为: 53.27013719336079
图像中的实际小球直径为: 59.0
MATLAB标定结果预测的图像坐标为: [357.95061973346844, 276.97752691978985] 562.4476463062763
通过公式计算焦距预测的图像坐标为: [357.3979136305529, 276.4389927682311] 554.2562584220408
图像中的真实图像坐标为: [374.0, 277.5]
MATLAB标定结果预测的图像坐标误差为: 16.057882334208234
通过公式计算焦距预测的图像坐标误差为: 16.635955282593372
MATLAB标定结果预测的小球图像直径为: 48.440161963179236
通过公式计算焦距预测的小球图像直径为: 47.73468802543336
图像中的实际小球直径为: 49.0
MATLAB标定结果预测的图像坐标为: [275.9916917966463, 276.49469460765914] 562.4476463062763
通过公式计算焦距预测的图像坐标为: [276.63262100132886, 275.9631923403614] 554.2562584220408
图像中的真实图像坐标为: [290.5, 277.5]
MATLAB标定结果预测的图像坐标误差为: 14.54309615781212
通过公式计算焦距预测的图像坐标误差为: 13.952275014330436
MATLAB标定结果预测的小球图像直径为: 53.39353852242215
通过公式计算焦距预测的小球图像直径为: 52.61592448594902
图像中的实际小球直径为: 55.0
MATLAB标定结果预测的图像坐标为: [449.3942637142454, 413.2213530368124] 562.4476463062763
通过公式计算焦距预测的图像坐标为: [447.5097885794676, 410.6985879370293] 554.2562584220408
图像中的真实图像坐标为: [470.5, 420.0]
MATLAB标定结果预测的图像坐标误差为: 22.167592535396125
通过公式计算焦距预测的图像坐标误差为: 24.800525952607423
MATLAB标定结果预测的小球图像直径为: 48.70065401703677
通过公式计算焦距预测的小球图像直径为: 47.9913863191642
图像中的实际小球直径为: 56.0
```

fov=90°时，图像坐标平均误差为4.016776；fov=60°时，图像坐标平均误差为18.000508；fov=120°时，图像坐标平均误差为4.538749。误差与目标生成的区域相关，目标越大，离中心越远，误差就越大。

## 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 [ChangeFOV.py](#) 时，可使用VS Code（或Pycharm等工具）来打开 [ChangeFOV.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 [ChangeFOV.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 = UE4CTF1API.UE4CTF1API()
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
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

[1] 无

## 7.常见问题

Q1: 无

A1: 无