1.实验名称及目的

1.1 实验名称

测距传感器实验

1.2 实验目的

通过测距传感器配置文件来获取传感器测距。

1.3 关键知识点

本实验主要是实现通过Python接口创建一个激光测距传感器，并实时获取测距数据。。

1）视觉接口使用

```
vis = VisionCaptureApi.VisionCaptureApi() \# 创建一个视觉传感器实例

vis.jsonLoad() \# 加载Config.json中的传感器配置文件

isSuss = vis.sendReqToUE4() \# 向RflySim3D发送取图请求

vis.startImgCap() \# 开启取图

vis.hasData[i] \# 图片i数据是否更新

vis.Img[i] \# 图片i数据（像素矩阵）

obj_distance = vis.DistanceSensor.Distance \# 获取距离传感器的距离 obj_distance

cv2.line(img, (center_x - crosshair_length, center_y), (center_x + crosshair_length,

cv2.putText(img, distance_text, (center_x - 50, center_y + crosshair_length + 20), cv

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：

```
"SeqID":0：使用自动更新ID的方式，创建了SeqID为0的视觉传感器

"TypeID":1和"TypeID":5：传感器类型为RGB和测距

"TargetCopter":1：相机绑定在1号飞机上

"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）

"SensorPosXYZ":[ 0.3,0,0]和"SensorPosXYZ":[ 0,0,0]：相机的位置。
```

3）飞机控制指令

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

mav.InitMavLoop() \# 初始化Mavlink监听程序，读取飞机数据

mav.initOffboard() \# 进入Offboard模式

mav.SendMavArm(True) \# 解锁飞控

mav.SendPosNED(0, 0, -2) \# 发送2米高的位置控制指令
```

4）UE控制

接口详细使用方法见：UE4CtrlAPI.py

```
ue = UE4CtrlAPI.UE4CtrlAPI() \# 创建UE控制实例

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

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
```

5）测距代码说明

Config.json中，配置TypeID为5，这里1号传感器（从0开始计数）对应测距传感器

```
19          {
20              "SeqID": 0,
21              "TypeID": 5,
22              "TargetCopter": 0,
23              "TargetMountType": 4,
24              "DataWidth": 640,
25              "DataHeight": 480,
26              "DataCheckFreq": 30,
```

在Python程序中，通过vis.hasData[1]确认已收到测距传感器数据，并获取数据指针，赋值给obj_distance=vis.Img[1]。注：vis.hasData[0] 和vis.Img[0]对应0号传感器为RGB图像。

```
77
78  while True:
79      if vis.hasData[0] and vis.hasData[1]: # 是否成功取图和获取测距传感器
80
81          img = vis.Img[0] # 获取0号传感器，图像数据指针，格式为opencv图像格式
82          obj_distance = vis.Img[1] # 获取1号传感器，距离数据指针，格式见Visic
83
```

通过类成员引用，可以打印数据

```
94          cv2.line(img, (center_x, center_y - crosshair_length), (cent
95
96          # 绘制距离信息
97          distance_text = f"Distance: {obj_distance.Distance} m"
98          cv2.putText(img, distance_text, (center_x - 50, center_y + c
99
```

注：测距传感器数据结构体定义如下：

```
class DistanceSensor:

\#\# @brief DistanceSensor的构造函数

\# @param 初始化类属性

def \__init__(self):

\#\# @var DistanceSensor.TimeStamp

\# @brief 这是当前消息的时间戳，初始化为 0

self.TimeStamp = 0

\#\# @var DistanceSensor.Distance

\# @brief 这是距离传感器测量到的距离，初始化为 0

self.Distance = 0

\#\# @var DistanceSensor.CopterID

\# @brief 用于标识直升机的ID，初始化为 0

self.CopterID = 0

\#\# @var DistanceSensor.RayStart

\# @brief 这是射线起点的坐标，初始化为[0,0,0]

self.RayStart = [0,0,0]

\#\# @var DistanceSensor.AngEular

\# @brief 这是传感器的欧拉角（Euler Angles），初始化为[0,0,0]

self.AngEular = [0,0,0]

\#\# @var DistanceSensor.ImpactPoint

\# @brief 这是碰撞点的坐标，初始化为[0,0,0]

self.ImpactPoint = [0,0,0]

\#\# @var DistanceSensor.BoxOri

\# @brief 这是盒子的原点或参考点，初始化为[0,0,0]

self.BoxOri = [0,0,0]
```

2.实验效果

本实验通过测距传感器配置文件来获取传感器测距的距离。

3.文件目录

例程目录：
[安装目录]\RflySimAPIs\8.RflySimVision\0.ApiExps\1-UsageAPI\1.ImgSenorAPI\6.RangingImageDemo

文件夹/文件名称	说明
RangingImage.bat	一键仿真启动脚本
RangingImage.py	Python实验代码
Config.json	视觉传感器配置文件
Python38Run.bat	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。

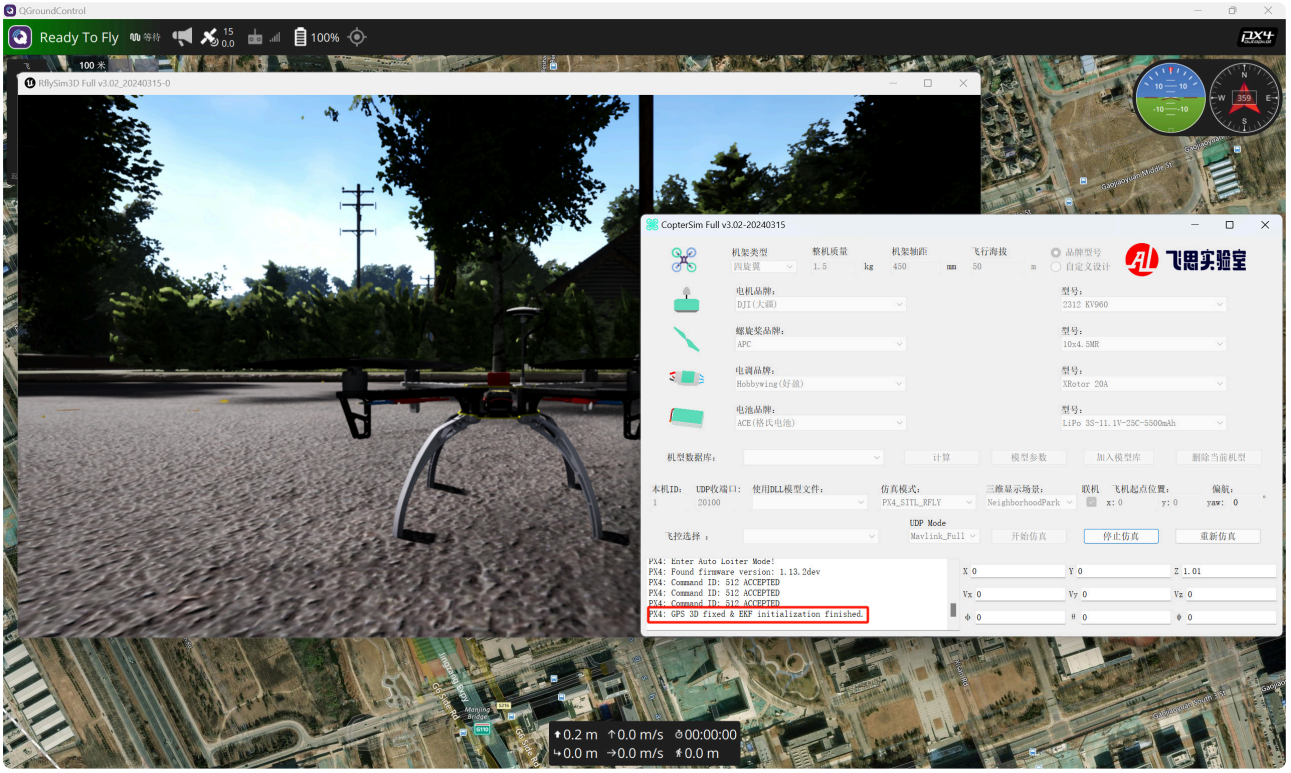
> ①：推荐配置请见：https://rflysim.com/doc/zh/HowToInstall.pdf

5.实验步骤

5.1 必做实验：Windows取图控制

Step 1：开启仿真

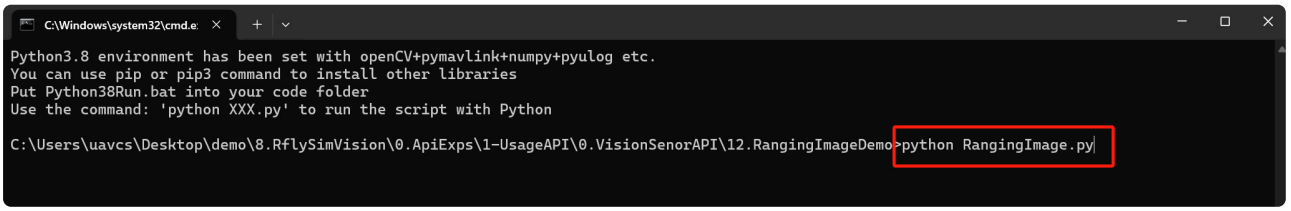
双击运行RangingImage.bat开启一个软件在环仿真。将会启动1个QGC地面站，1个CopterSim软件且其软件下侧日志栏必须打印出GPS
3D fixed & EKF initialization finished字样代表初始化完成，并且RflySim3D软件内有1架无人机。



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RangingImage.py文件，输入

```
python RangingImage.py
```



Step3：观察结果

可以看到一个测距的图像窗口。

Step 4：结束仿真

在下图“RangingImage.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RangingImage.py时，可使用VS Code（或Pycharm等工具）来打开RangingImage.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RangingImage.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.VisionSenorAPI > 1.Came

```
ue = UE4CtrlAPI.UE4CtrlAPI()

#Create a new MAVLink communication instance, UDP sending
mav = PX4MavCtrl.PX4MavCtrler(1)

# The IP should be specified by the other computer
vis = VisionCaptureApi.VisionCaptureApi()

# Send command to UE4 Window 1 to change resolution
ue.sendUE4Cmd('r.setres 1280x720w',0) # 设置UE4窗口分辨率，
ue.sendUE4Cmd('t.MaxFPS 30',0) # 设置UE4最大刷新频率，同时也
time.sleep(2)

# VisionCaptureApi 中的配置函数
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