

1. 实验名称及目的

1.1 实验名称

外部传感器之光流定位（UE计算）（仅限完整版及以上版本）

1.2 实验目的

了解UE中基于真值计算光流和基于OpenCV计算光流两种方式的使用方法

1.3 关键知识点

本实验需要电脑中部署Visual Studio 2022环境，部署方式见：[\[安装目录\]\RflySimAPIs\1.RflySimIntro\2.AdvExps\6.VisualStudioInstall](#)

例程功能API

传感器配置接口：

UE控制接口：python库文件 [UE4CtrlAPI.py](#)：[RflySimSDK/html/UE4CtrlAPI_8py.html](#)

UE控制命令发送接口sendUE4Cmd：

[classRflySimSDK_1_1ue_1_1UE4CtrlAPI_1_1UE4CtrlAPI.html#sendUE4Cmd](#)

载具控制接口：python库文件 [PX4MavCtrlV4.py](#)：

[RflySimSDK/html/PX4MavCtrlV4_8py.html](#)

MAVLink消息侦听启用接口InitMavLoop：

[RflySimSDK/html/PX4MavCtrlV4_8py.html](#)

Offboard模式启用接口initOffboard：

[RflySimSDK/html/PX4MavCtrlV4_8py.html#initOffboard](#)

定点位置控制接口SendPosNED:

RflySimSDK/html/PX4MavCtrlV4_8py.html#SendPosNED

降落控制接口sendMavLand:

RflySimSDK/html/PX4MavCtrlV4_8py.html#sendMavLand

关键知识点1: 光流传感器的实现原理

光流传感器使用一个朝下的摄像头（通常是低分辨率、低帧率的摄像头）来持续拍摄地面的图像，以及一个处理单元来分析相邻图像之间的像素移动，从而计算出“光流”——即图像在二维平面上的移动速度。通过对光流信息的处理（例如扩展卡尔曼滤波器EKF2），可以估算出无人机相对地面的平移速度和方向。

注意，光流传感器通常用于室内（GPS不可用）近地面低速飞行（如悬停）。光流传感器通常需要有纹理丰富的地面作为参照，光滑或单调的地面会降低其性能（低光照或反光环境也会）。

本例中根据真值直接计算的方式：通过传感器配置的Width和Height以及FOV和已知的运动真值计算当前光流在像素上的偏移量；OpenCV函数计算光流的方式：将相机捕获的画面传递给OpenCV，获取图像的角点数量，通过识别的角点数量/最大角点数量来计算光流质量，然后通过稀疏光流计算函数来获取有效数据，计算平均值。

注意，上述两种方式的计算过程均在UE中实现，本例使用的python脚本仅用于输出计算结果；两种计算方式的核心区别在于基于真值计算时固定光流质量为255（假设真值数据准确且无噪声），而基于OpenCV计算时识别的角点越多，光流质量越高。

关键知识点2: [Opticalflow.py](#) 关键代码解析

2. 实验效果

3. 文件目录

例程目录:

[\[安装目录\]\RflySimAPIs\4.RflySimModel\3.CustExps\e7_ExternalSensors\e3_Opticalflow_UE](#)

文件夹/文件名称	说明
Config.json	传感器配置文件
Python38Run.bat	python环境启动脚本
SITLPosStr.bat	软件在环启动脚本
Opticalflow.py	例程功能主程序
VisionCaptureApi.py	传感器接口库文件

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/>

5. 实验步骤

直接计算光流实验（必做）

Step 1: 修改传感器配置

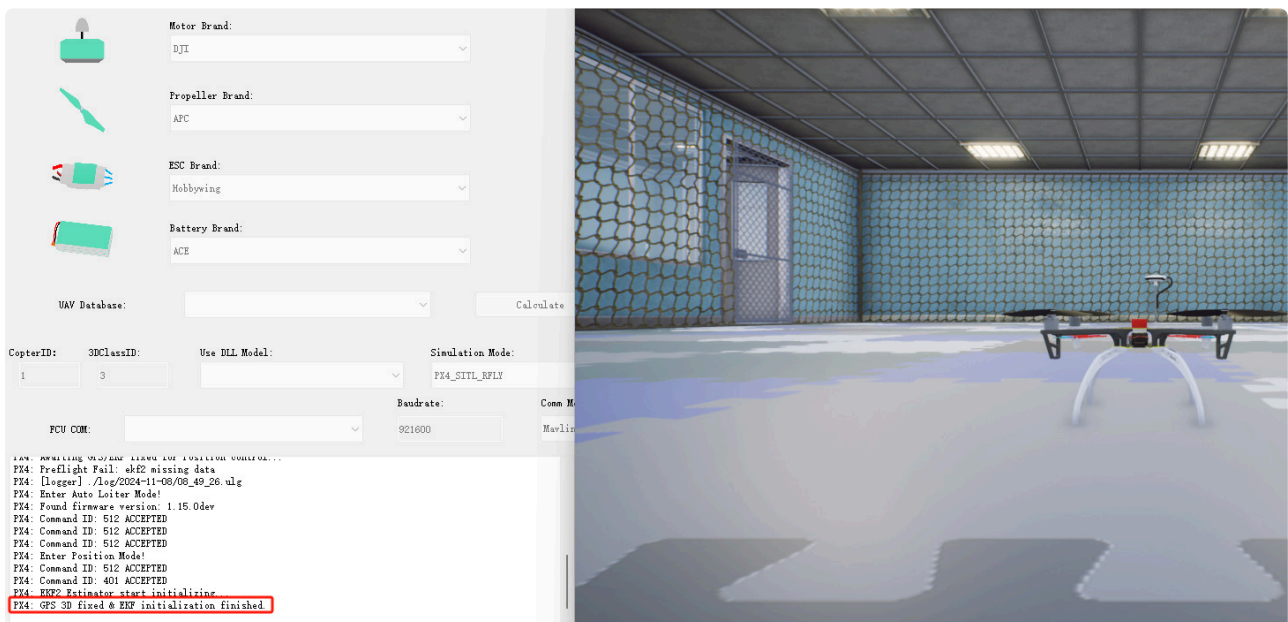
打开本目录的Config.json将otherParams[1]置为0

```
Config.json
文件 编辑 查看

{
  "VisionSensors": [
    {
      "SeqID": 1,
      "TypeID": 10,
      "TargetCopter": 1,
      "TargetMountType": 0,
      "DataWidth": 640,
      "DataHeight": 480,
      "DataCheckFreq": 1,
      "SendProtocol": [1, 127, 0, 0, 1, 9999, 0, 0],
      "CameraFOV": 90,
      "SensorPosXYZ": [0, 0, 0],
      "SensorAngEuler": [0, 90, 0],
      "otherParams": [5, 0, 0, 0, 0, 0, 0, 0]
    }
  ]
}
```

Step 2: 启动软件在环仿真

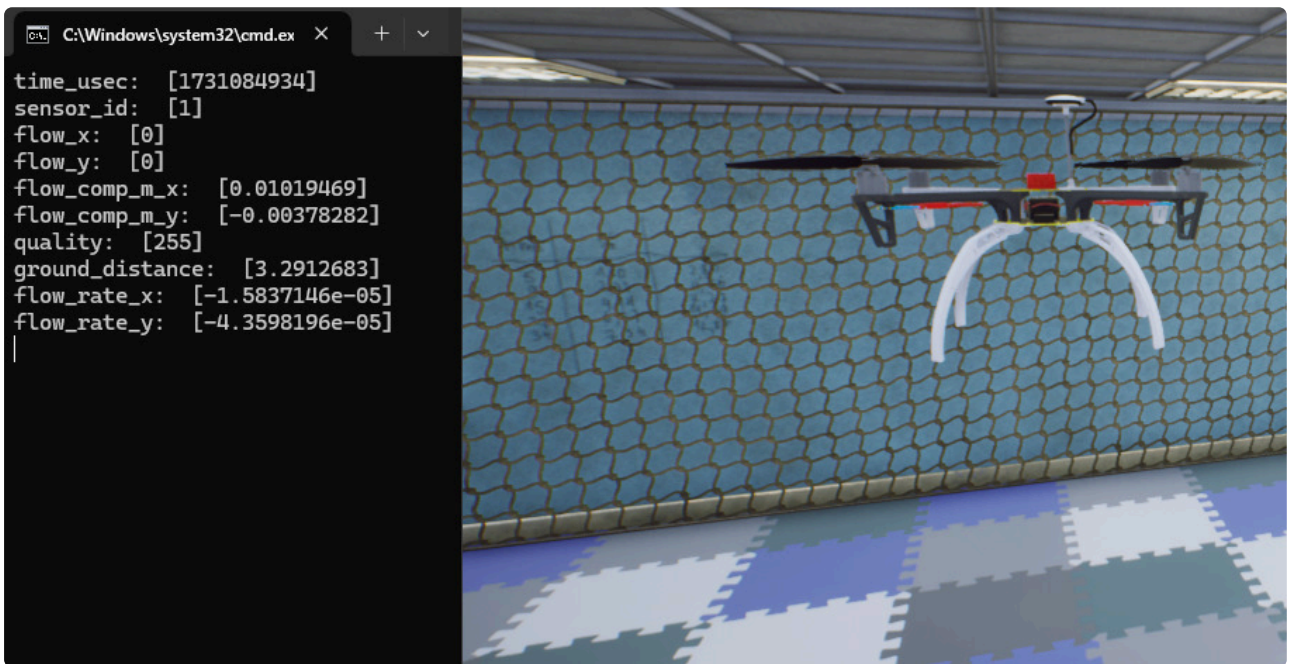
双击 `SITLPosStr.bat` 启动软件在环仿真，等待初始化完成



Step 3: 运行python脚本加载光流计算结果

```
C:\Windows\system32\cmd.exe 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
F:\d2\4.RflySimModel\3.CustExps\e7_ExternalSensors\e3_Opticalflow_UE>python Opticalflow.py
```

在文件夹下，双击38Run.bat，打开集成好的环境，输入python Opticalflow.py，回车运行。



注：运行本目录的 [comp.py](#) 可以对比光流传感器的计算结果与飞机的运动状态真值，飞机运动状态真值日志的获取方法见 [..\..\0.ApiExps\8.OutCopterData](#)

OpenCV计算光流实验（必做）

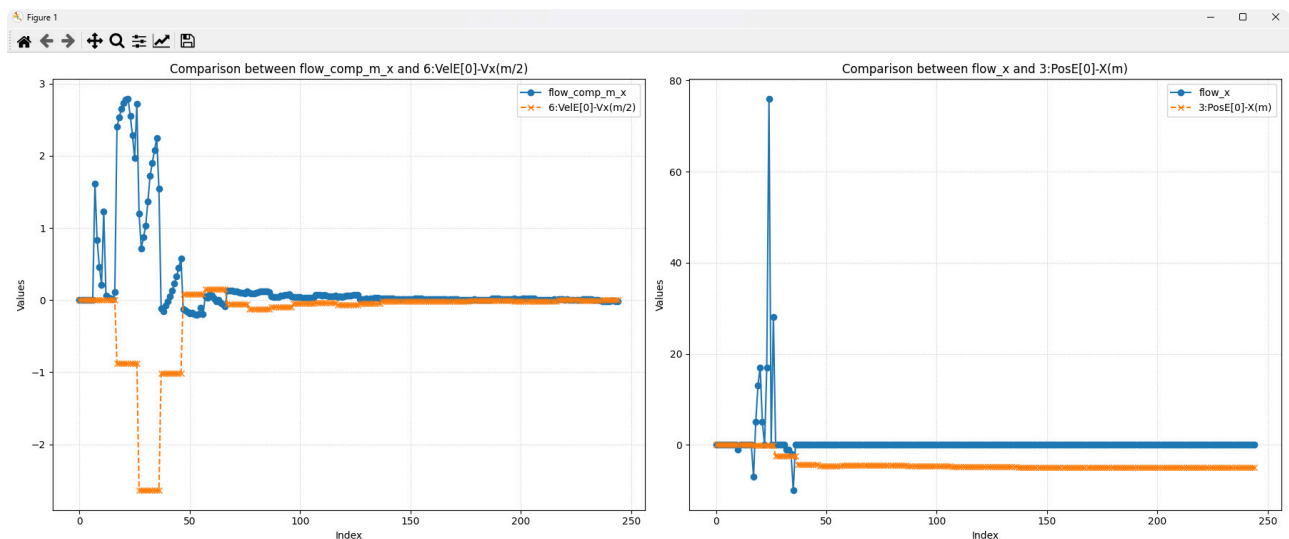
Step 1: 修改传感器配置

打开本目录的Config.json将otherParams[1]置为1

```
Config.json
文件 编辑 查看

{
  "VisionSensors":[
    {
      "SeqID":1,
      "TypeID":10,
      "TargetCopter":1,
      "TargetMountType":0,
      "DataWidth":640,
      "DataHeight":480,
      "DataCheckFreq":1,
      "SendProtocol":[1,127,0,0,1,9999,0,0],
      "CameraFOV":90,
      "SensorPosXYZ":[0,0,0],
      "SensorAngEular":[0,-90,0],
      "otherParams":[5,1,0,0,0,0,0,0]
    }
  ]
}
```

之后按照 [直接计算光流实验（必做）](#) 中的step2~3完成光流计算结果的输出



1. Vscode调试运行实验（选做）

准备工作:

- 先确保已经按 [RflySimAPIs\1.RflySimIntro\2.AdvExps\e3_Config\Readme.pdf](#) 步骤，正确配置VS Code环境。或者配置了自己的Pycharm等自定义环境。
- 其他步骤与上文相同，在运行文件时，可使用VS Code（或Pycharm等工具）来打开文件文件，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阅读例程中的源码，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。
- 修改Config.json，测试不同频率时的传感器精度

6.参考资料

[1]. [\[安装目录\]\RflySimAPIs\3.RflySim3DUE\API.pdf](#)

UE4性能调试分析常用方法 -

[2]. 知乎 : <https://zhuoanlan.zhihu.com/p/273608458>

[3].

7.常见问题

Q1: ***

A1: ***