

1. 实验名称及目的

1.1 实验名称

自动生成 YOLO 数据集实验

1.2 实验目的

运行 `get_dateset.py` 文件即可自动生成 YOLO 格式的数据集。再运行 `maketxt.py`对生成数据分成训练集和测试集。对已有数据集进行划分运行 `maketxt.py`文件即可对已有数据集进行划分，但是需要将其中的一些地址更改为所要划分的数据集地址。

1.3 关键知识点

通过定义的函数获取到目标飞机的位置，并将当前的画面保存到以当前时期和时间戳命名的文件夹中的image文件夹，并在label文件夹中生成txt文件，txt文件保存的5个值为标签号、`x_center`、`y_center`、`width`、`height`，再通过`maketxt.py`文件将其转化为YOLO数据式。

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）。自动生成YOLO 格式的数据集。再运行 `maketxt.py`对生成数据分成训练集和测试集。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 cv2.rectangle(img_, (top_left[0], top_left[1]), (bottom_right[0],bottom_right[1]),
(0, 0, 255), 1) \# 使用 OpenCV 在图像 img_上绘制一个矩形, 左上角坐标为
(top_left[0],top_left[1]), 右下角坐标为(bottom_right[0], bottom_right[1]), 矩形的颜色
(0, 0,255), 矩形边长的粗细为1像素
```

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] : 传输模式为1：UDP网络传输模式（图片使用jpeg压缩，点云直
8 传）。
9
"SensorPosXYZ":[ 0.03,0,0] : 相机分布。
```

3) 飞机控制指令

```
1 mav = PX4MavCtrl.PX4MavCtrler(1) \# 创建飞机控制实例
2
3 mav.InitMavLoop() \# 初始化Mavlink监听程序，读取飞机数据
4
5 mav.initOffboard() \# 进入Offboard模式
6
7 mav.SendMavArm(True) \# 解锁飞控
8
9 mav.SendPosNED(0, 0, -10, 0) \# 发送10米高的位置控制指令
```

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('r.flyChangeMapbyName GrassLands') \#更新地图场景，类型为GrassLands
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，
6 | 取图分辨率在json中配置，本窗口设置越小，资源需求越少。
7 |
8 | 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
9 |
10 | 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
11 | 为[0,0,-8.086-1.5]，欧拉角为[0,0,0]
|
| ue.sendUE4PosScale(100, 152, 0, InitTargePos, [0, 0, 0], [0.1, 0.1, 0.1]) \#与
| ue.sendUE4Pos含义一样，但是多了一个缩放倍数[0.1, 0.1, 0.1]
```

5) 其余代码说明

```
1 | def eul2rot(theta) \#欧拉角旋转矩阵，与世界坐标系方向、欧拉角旋转顺序、旋转正方向的定义有关
2 |
3 | def getExternalMatrix(uavPos, cameraPosForUav, eurTheta) \#获取外参矩阵，转换为齐次坐标
4 | 形式并输出
5 |
6 | def getUav9Point(uavPos, copterCenterHeight, uavAng, UAVh=0.185,
7 | UAVw=0.185,eurTheta=[0, 0, 0]) \# 通过输入UAV的坐标以及相关信息，获取九个空间点的坐标
8 |
9 | def get_image_size(knownWidth, focalLength, distance) \# 图像大小计算函数
10 |
11 | def get_pic_situation(relative_pos, px, py, focalLength) \#计算目标在图像中的位置
12 |
13 | def text_create(name, msg) \# 创建一个txt文件，文件名为name,并向文件写入msg
14 |
15 | camere_pos = PosInit
16 | +np.transpose(np.dot(np.linalg.inv(eul2rot(angCopterE)),np.transpose(cameraPosForUa
17 | v))) \# 到相机在世界坐标系中的精确位置 camere_pos
18 |
19 | timeInterval = 0.1 \# 以10hz的频率进行控制
|
| lastTime = lastTime + timeInterval \# 设置每一帧的处理结束时间
|
| sleepTime = lastTime - time.time() \# 计算休息时间，从而保持按照设定的频率执行代
```

2. 实验效果

运行 get_dateset.py 文件即可自动生成 YOLO 格式的数据集。

3. 文件目录

例程目录：

[安装目录]\RflySimAPIs\8.RflySimVision\0.ApiExps\3-VisionAPI\5.GenVisionDataSet
\assets\bk

文件夹/文件名称	说明
get_dateset.py	自动生成 YOLO 格式的数据集。
maketxt.py	对已有数据集进行划分。
OneCameraCal.bat	一键启动脚本。
Config.json	视觉传感器配置文件
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；Visual Studio 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://rflsim.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://rflsim.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，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，在该环境下运行get_dateset.py文件，输入 `python get_dateset.py`

自动生成YOLO格式的数据集。

| Step 3：观察结果

生成数据后，停止运行。根据自己目标所在位置进行更改，运行maketxt.py文件。

该文件夹内为采集到的数据。

| 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get_dateset.py时，可使用VS Code（或Pycharm等工具）来打开get_dateset.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get_dateset.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。

- 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