

1. 实验名称及目的

1.1 实验名称

自动生成 AI 训练书数据集实验

1.2 实验目的

文件夹内有两个例程 python 脚本，分别生成图像数据集以及点云数据集，图像数据集是以 VOC 格式输出，因此 VOC 转到具体训练框架也很方便，点云数据集以 kitti 数据集格式，图像数据集生成例程([ExampleImg.py](#))与点云数据生成例程([ExamplePointCloud.py](#))都是以静态目标位置，具体到目标怎么运动，由用户规划其运动轨迹以及控制姿态。

1.3 关键知识点

本实验主要是实现通过Python接口VisionCaptureApi.py使用（见 RflySimAPIs\RflySimSDK\vision目录），去建立自动生成 AI 训练书数据集。关键代码解析如下：

本例子和其他例子的区别，主要在于：

ExampleImg.py利用从rfly.GetData方法获取的数据存储到xml文件里，通过遍历所有物体的数据，将全部图像数据集存储起来。而txt文件则是记录哪一部分是训练集，哪一部分是验证集。Jpg文件则是图像文件。

ExamplePointCloud.py利用从rfly.GetData方法获取的数据通过计算和转化，分别将点云数据以二进制的形式存储到bin文件里，将标注信息存储到txt文件中。

关键知识点1: SendProtocol[0]决定了图像的传出模式。SendProtocol[0]=0: 共享内存（仅限Windows下获取图像），1: UDP直传png压缩，2: UDP直传图片不压缩（只适用图片类传感器），3: UDP直传jpg压缩（只适用图片类传感器）。如果是激光雷达数据只有0或1（共享内存和UDP网络传输）。

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(0, TargetIP) #向RflySim3D发送取图请求, 发给ip为TargetIP的地址
6
7 vis.startImgCap()# 开启取图
8
9 cv2.rectangle(img_, (top_left[0], top_left[1]), (bottom_right[0],bottom_right[1]), (0
10 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
11
12 cv2.imwrite(path_img + '/' + "{}.jpg".format(count), img) \# 将当前图像 img保存到指定路径
13
14 cv2.imshow("ret", img_) \# 显示图片img_图像, 名字为ret
```

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和"TypeID":20 : 传感器类型为RGB彩色图像和激光雷达
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": [0,0,-0.5] : 相机分布位置。
```

3) UE控制

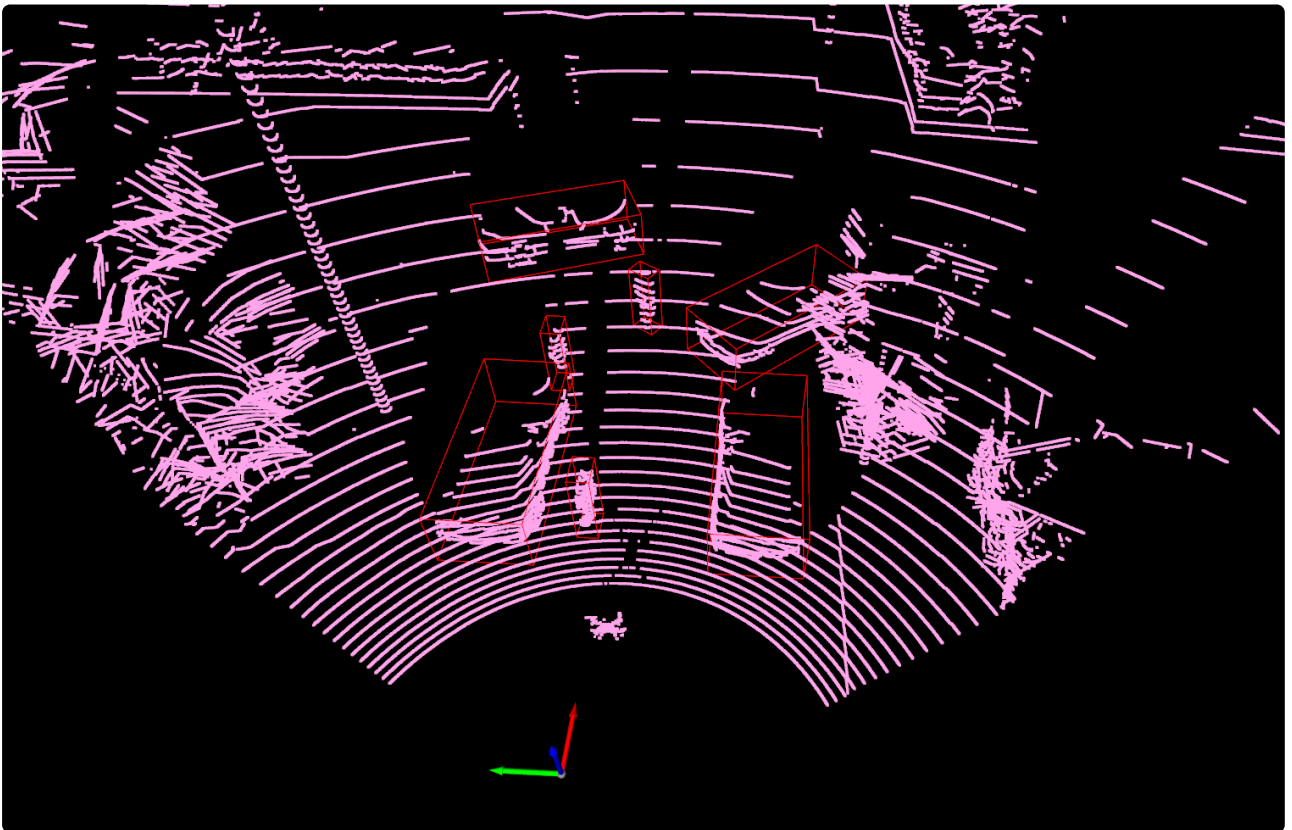
接口详细使用方法见：[UE4CtrlAPI.py](#)

```
1 ue = UE4CtrlAPI.UE4CtrlAPI() # 创建UE控制实例
2
3 ue.sendUE4Cmd('RflyChangeMapbyName NeighborhoodPark', 0) #更新地图场景, 类型为Neighborhood
4
5 ue.sendUE4PosScale(1, 3, 0, PosE=[2, 0, -3], AngEuler=[0, 0, -1.57]) #创建一个障碍物, ic
```

4) 其余代码说明

```
1 rfly = RflySimObj.RflySimObj(ue, vis) \# 创建RflySimObj实例, 用于获取数据
2
3 rfly.ReqRfySim() \# 与RflySim 建立连接用
4
5 th = threading.Thread(target=KeyEvents) \# 设置键盘打断线程, 用于停止生成数据集
6
7 os.makedirs(path_dir) \# 创建一个目录, 该目录的位置、名字信息都在path_dir里
8
9 timeInterval = 0.1 \# 以10hz的频率进行控制
10
11 lastTime = lastTime + timeInterval \# 设置每一帧的处理结束时间
12
13 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
14
15 data, obj = rfly.GetData(0, target) \# 调用 rfly.GetData 方法获取数据
16
17 img = copy.deepcopy(data) \# 深拷贝data, 以便后续对图像数据的处理不会影响原始数据
18
19 xml_full_path = xml_floder + '/' + '{}.xml'.format(count) \# 创建 XML文件的完整路径, 文件
20
21 with open(xml_full_path, 'w', encoding='utf-8') as f: doc.writexml(f, addindent='\t',
22
23 rfly.sensors[str(sensor_id)].min_points = 200 \# 设置目标最少点云数
24
25 rfly.sensors[str(sensor_id)].max_dis = 50 \#设置目标最远有效距离, 即能识别轮廓的最远距离, 这个
26
27 points.tofile(bin_path) \# 将点云数据保存为二进制文件, 路径为bin_path
28
29 file = open(full_path, 'w') \# 在 path_label 目录下创建一个新的文本文件count.txt 用于保存标
30
31 d_l = np.linalg.norm(np.array(val[1]) - np.array(val[5])) \#计算目标物体的长度, np.linalg
32
33 alpha = ori[2] - 1.57 - rfly.objs[str(1)].orientation[2] \#计算目标物体与传感器之间的相对方
34
35 b=np.dot(eul2rot(rfly.objs[str(1)].orientation), np.transpose(relative_position(obj[st
36
37 file.writelines([object[str(id)], ' ', str(0), ' ', '0', ' ', str(alpha), ' ', str(0),
```

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\1.GenObjectDataSet

文件夹/文件名称	说明
20221218_122838_img	存放生成的图像数据集目录
20221219_120849_pc	存放点云数据集目录
RflySimObj.py	类接口代码脚本
ExampleImg.py	图像数据集生成例程
ExamplePointCloud.py	点云数据生成例程
Python38Run.bat	Windows下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_fm4-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GenRflySimSets.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，在该环境下运行 ExampleImg.py文件，输入 `python ExampleImg.py`

```
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\1.GenObjectDataSet>python ExampleImg.py|
```

Step 3: 观察结果



观察到结果后，关闭全部窗口!!!

Step 4: 运行点云控制程序

重新执行Step1步骤。

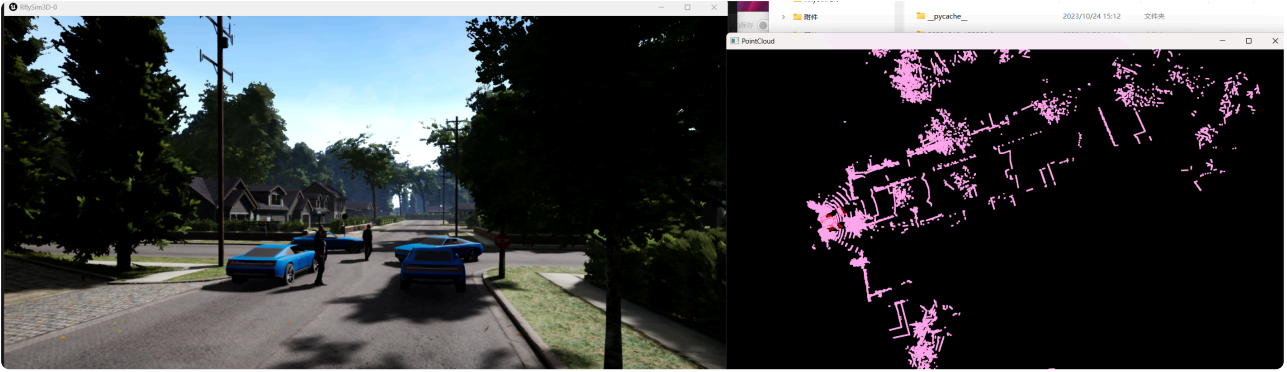
再在文件夹下，双击Python38Run.bat，打开集成好的python环境，在该环境下输入 `python ExamplePointCloud.py` 运行ExamplePointCloud.py文件文件，在采集完数据后，在代码中找到相应的位置，取消本行代码的注释（在154行左右）。

```
lastTime = time.time()
data, obj, orientation = rfly.GetData(1, target)
# rfly.Visualization3D(data, obj) # 该接口为阻塞模式可视化，主要是为通过鼠标缩放以及调整最佳视角姿态
points = np.copy(data)
bin_path = path_bin_point + "/" + "{}.bin".format(count)
```

（注：实际采集数据的时候，这个接口的调用得注释掉，因为它是阻塞模式显示的。）

再次运行本代码。

Step 5: 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\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ExampleImg.py时，可使用VS Code（或Pycharm等工具）来打开ExampleImg.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ExampleImg.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。

```
> RflySimAPIs > 8.RflySimVision > 0.ApiExps > 1-UsageAPI > 0.VisionSenorAPI > 1.Came
8   ue = UE4CtrlAPI.UE4CtrlAPI()
9
10  #Create a new MAVLink communication instance, UDP sending
11  mav = PX4MavCtrl.PX4MavCtrler(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: 运行过程中RflySim仿真画面较卡甚至出现RflySim仿真界面出现卡死的现象并显示 Fatal error!

A1: 可能是由于在Config.json文件中设置的点云数据传输的数据量设置过大（发送的点云数据量由“DataWidth”和“DataHeight”共同决定），需要调整“DataWidth”和“DataHeight”的值，从而减小电脑的压力。...