

1.实验名称及目的

1.1.实验名称

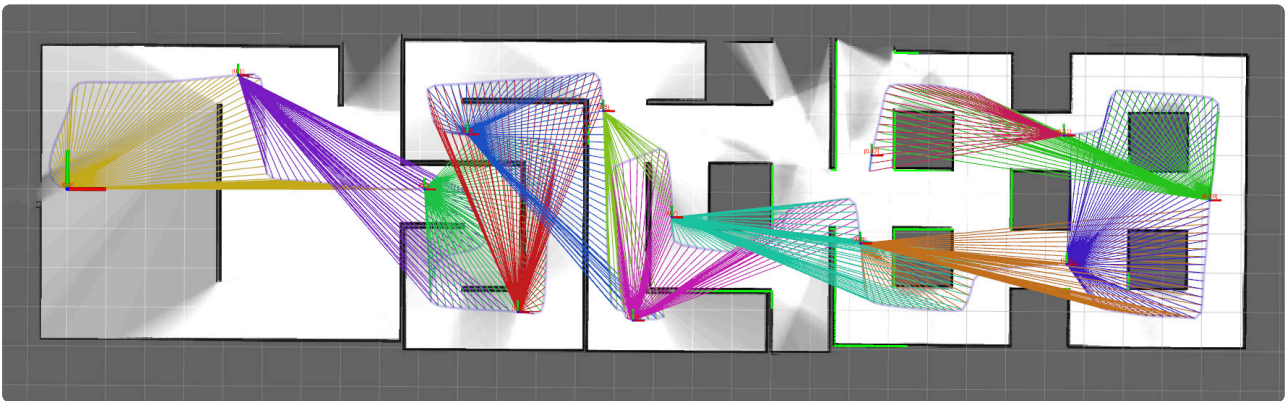
Astar算法规划路径实验

1.2.实验目的

该例程使用A*算法规划路径，在路径搜索中，把传统四邻域搜索改为8邻域搜索。

1.3.关键知识点

首先算法使用场景为MatchScene, 下图为平台单线激光雷达建的地图。

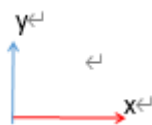


A*算法在平台中的应用

算法输出为图像坐标系的像素位置，那么怎么根据图像坐标系像素点转化为平台里面的NED坐标系的坐标呢？

计算图像坐标系到平台NED坐标系的坐标；

图像坐标系如图：



yaml文件内容：

```

1 | image: competition.pgm
2 | resolution: 0.05
3 | origin: [-8.4, -4.42462, 0.0]
4 | negate: 0
5 | occupied_thresh: 0.65
6 | free_thresh: 0.196

```

分辨率为0.05，原点所在的像素点为: $x = 8.4/0.05$, $y = 4.42462/0.05$ ，即O(168,88)，通常图像坐标系习惯是y朝下的，因此为了方便转换，在图通用图像坐标系下原点坐标为(168,img_height-88)，

在建图时候，原点位置就在飞机起飞的位置，那么只需要把两个坐标系的方向确定，即可转换。假设图标上一点p(x,y)。在平台NED坐标系下为

$$\begin{aligned}
 & x_{ned} \\
 = & (\\
 & x_p \\
 - & \\
 & x_o \\
 &)/0.05, \\
 & y_{ned} \\
 = & -(\\
 & y_p \\
 - & \\
 & y_o
 \end{aligned}$$

)/0.05，这样通过把路径上点按照公式进行转换即可。

本实验主要是实现通过Python接口cv2库对图像进行操作。读取一张图片信息，划分好可行点和不可行点，然后确定起点、终点，使用A*算法规划路径，在路径搜索中，把传统四邻域搜索改为8邻域搜索。关键代码解析如下：

1) cv2接口使用

```
1 | img = cv2.imread(sys.path[0]+"\\\\\\competition.pgm") \# 读取图像
2 |
3 | grid = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY) \# rgb图像转化为灰度图像
4 |
5 | cv2.circle(img, (origin_p[1], origin_p[0]), 4, (0, 0, 255), 8) \# 在图像 img上绘制一个圆
6 |
7 | 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

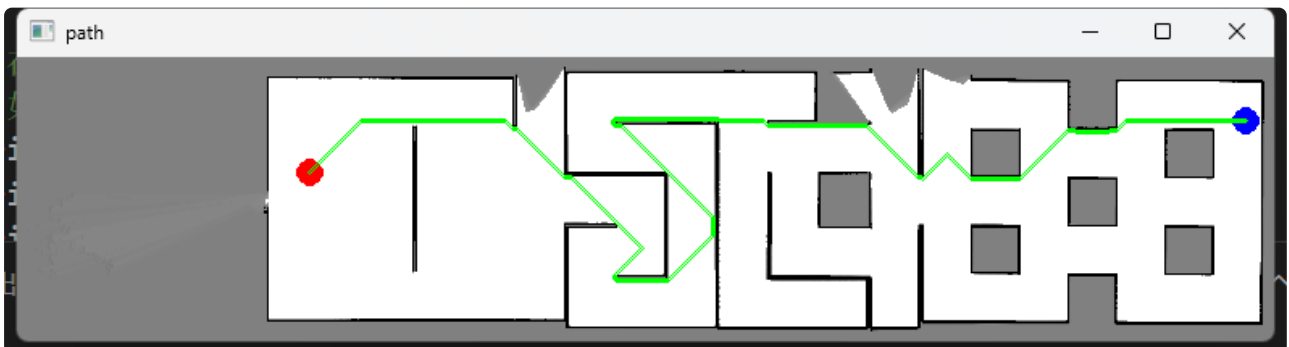
其中，视觉传感器的初始状态由本文件夹下的Config.json决定，主要包含以下配置项：

```
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：传感器类型为分割图像
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，点云直传）
8 |
9 | "SensorPosXYZ":[ 0.3,0,0]：相机分布。
```

3) 其余代码说明

```
1 | def AStar(grid: list, begin_p: list, goal_p: list, cost=2) \#实现了A算法来在一个二维网格中
2 |
3 | assert ((grid[begin_p[0]][begin_p[1]] != 1) and (grid[goal_p[0]][goal_p[1]] !=1)) \#
```

2.实验效果



3. 文件目录

例程目录：[\[安装目录\]](#)\RflySimAPIs\8.RflySimVision\2.AdvExps\e10_Planner\AStar

文件夹/文件名称	说明
Planner.bat	启动软件在环仿真配置文件
AStar.py	Python实验脚本
Config.json	视觉传感器配置文件
slam.png	slam地图图片
competition.yaml	ros配置文件
competition.pgm	地图图片
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；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：运行控制程序

双击运行 [Planner.bat](#) 文件，运行脚本将会启动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。

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 [AStar.py](#) 程序，就能看到路径规划的效果。

Step 2：观察结果



Step 3：结束仿真

在下图“[Planner.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\3.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 [AStar.py](#) 时，可使用VS Code（或Pycharm等工具）来打开 [AStar.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 [AStar.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.Came
8   ue = UE4CtrlAPI.UE4CtrlAPI()
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，;
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

无

7.常见问题

Q1: 无

A1: 无