

道路行驶规则与路口交叉口数据:

实验原理:

路口可视化需要将提取到的道路规则定义到各个路口的图像上, 为了保障最终成图的整齐美观, 首先对所有路口进行划分, 由于每个路口存在上下左右四个分路口, 需要分别对四个分路口的路口规则进行可视化。考虑到在原地图中, 部分路口存在角度, 为了保证最终成图的精确性, 还设置了路口的角度属性, 最终成图的路口图像将会依据路口角度进行总体偏转。

实验思路:

1.道路路口的生成:

首先读取路口点坐标, 生成数组。数组路径保存在 `tpath` 中, 将读取的数组保存为 `coor`。

依照路口分析结果, 数组最终的构造形式为: `41*4*5`, 及 `41` 类 `4` 行 `5` 列。

数组存在三个维度: 第一维代表路口总数, 经实际计算总数为 `41`。

第二维和第三维: 精确到各个路口来进行路口属性的对应。二三维的数组形式是 `4` 行 `5` 列, 其中, `4` 行分别代表位于该路口的右、上、左、下的分路口的路口属性。`5` 列分别代表了不同的含义, 第一列代表是否存在该方向的路口。路口自身的角度大小, 汽车在此分路口是否可以调头, 是否可以左转, 是否可以直行, 是否可以右转。取值为 `0/1`, 取值 `0` 代表否, 取值 `1` 代表是。数组构造形式如下:

```
basic=[[10,0,1,1,1],[100,0,1,1,1],[190,0,1,1,1],[280,0,1,1,1]]
```

2.单个路口绘图

画图部分由多个函数组成, 方便后续的调试与修改。

单个路口绘图函数 `SinPicture`: 在此函数中, 定义了画布的大小, 背景颜色等基本属性。

分支路口绘制函数 `DrawRoad,Drawline`: 绘制每个路口内的四个分支路口。

四个道路规则绘制函数 `GoBack(a,b),GoStraight(a,b),GoRight(a,b),GoLeft(a,b)`: 采用箭头代表该路口是否存在该道路规则, 该函数的输入参数 `a` 代表不同的分支路口, 输入参数 `b` 代表各个分支路口每个箭头外接矩形的宽度。在四个函数中, 定义了四个箭头的颜色, 大小等属性, 箭头的颜色代表该箭头所处的分支路口具体位置, 颜色由输入参数 `a` 决定。定义规则为: 上方路口为黄色, 左方路口是蓝色, 下方路口是绿色, 右方路口是红色。由于每个分支路口存在的道路规则数量不同, 需要依据规则数量对箭头位置进行调整, 故由外接矩形宽度 `b` 判定箭头位置。如在函数 `GoRight` 中, 有如下代码:

```
t.penup()
t.forward(118)
t.lt(90)
t.fd(b/6)
```

即依据外接矩形的宽度来调整绘图的起始位置。

绘制方法是调用 `turtle` 库, 通过 `t.penup()` 与 `t.pendown()` 来改变绘图位置, 并防止留下笔触。绘图起始时, 笔尖位于该箭头外接矩形的左上角, 绘图结束时, 调整笔尖位置, 使之返回箭头的左上角。之后笔尖向前移动该分支路口路口规则外接矩形的宽度, 继续绘制下一个路口规则。

3.多个路口绘图

循环调用路口绘制函数 `SinPicture`, 绘制所有路口的道路规则。

```
for i in range(len(n)):
    SinPicture(n[i],i)
```

4.绘图结果保存、图片格式转换及图片缩小

将 `turtle` 绘图结果保存为当前文件夹下的 `.eps` 文件, 用 `postscript` 生成图片时, 对于 `.jpg` 格式, 即使生成图片, 也因为 `JPEG` 标识符段长度太短, 导致文件不完整无法打开, 故保存为 `.eps` 格式, 作为中间结果。

```
EpsPic="p"+str(seq)+".eps"
ts=t.getscreen()
ts.getcanvas().postscript(file=EpsPic)
```

将 `.eps` 格式图片转换为 `.png` 格式图片, 由于 `JPEG` 是有损压缩, 会导致生成的图片清晰度较低, 因此不保存为 `.jpg` 格式, 而是 `.png` 格式。

```
JpgPic1 = "p"+str(seq)+".png"
im = Image.open(EpsPic)
im.load(scale=5)
im.save(JpgPic1,"PNG",quality=95) #quality
```

参数: 保存图像的质量, 范围从 `1-95`, 尽量避免设置高于 `95` 的值, 若使用默认参数会导致结果

图片被压缩。

5. 缩小图片到合适尺寸

缩小图片，设置重采样方法为 **ANTIALIAS**，以保证缩小后的图片有较高的分辨率。

```
out.save(JpgPic2, "PNG", quality=100)
```