

项目介绍

本项目采用c++编写，适用系统为ubuntu20.04TLS，使用make和gcc进行编译

包含内容

eigen3: c.cpp中使用的矩阵运算库

input.jpg: 为选定的卡通图像

edges文件夹: 内含组成input.jpg的12条曲线，分别放在1~12.txt文件中

cordinates文件夹: 内含1~24两个文件夹，每两个文件夹对应edges中的一条曲线，文件夹中存放数个坐标文件

splines文件夹: c.cpp拟合出的曲线的输出位置，内含res1~12.txt共12个文件，是原曲线的拟合结果

process.py: python文件，用于处理图片，抽取图形边缘曲线，将坐标转化成可视化的txt文件

c.cpp: 该文件利用cordinates中的坐标文件，利用三次样条拟合出曲线放在splines文件夹中

d.cpp: 该文件利用splines中的曲线坐标判断曲线之间的偏序关系，生成偏序集

makefile: 内含编译命令

使用方法

按照重置，编译，运行的顺序使用本项目

重置指令

```
make reset
```

编译指令

```
make all
```

运行指令

```
./c
```

```
./d
```

原理介绍

曲线提取方法:

利用广度优先搜索，从一白点开始，搜索完周围联通的所有白点，得到的和黑点相邻的白点就是一条曲线的边缘，

从黑点开始搜索同理。

曲线拟合方法:

由于需要计算dx和dy，在计算机中不允许除零的操作，也不允许导数无穷大，所以将一条曲线拆分成多个部分，有的按x-y坐标系进行拟合，有的按y-x坐标系进行拟合，避免了除零操作。所以之前cordinates文件夹中两个文件夹对应一条曲线的x-y部分和y-x部分。

三次样条的生成主要是利用数学原理推导出计算方法，将 dx 等信息放入矩阵 H ，将 dy/dx 等信息放入 Y 向量，然后利用eigen3提供的 $H.colpivhouseholder().solve(Y)$ 方法解方程，计算出各部分曲线的四个系数，然后在选定坐标之间插入拟合曲线的坐标点，生成一条完整的曲线。由于最终坐标是整数，需要对拟合结果进行舍入操作。

偏序关系判断方法：

可以通过判断曲线之间内点是否存在子集关系来判断包含关系即偏序关系，但需要记录每条曲线的一个内点进行宽度优先搜索。在本项目中反其道而行之，因为坐标 $0,0$ 是所有曲线的外点，利用 $0,0$ 找出每条曲线的所有外点，然后根据外点集合的子集关系推断出曲线之间的偏序关系。