

《程序设计与算法综合实习 I》 课程设计报告

题 目： 图像的边缘识别和处理问题

班 级： 19 级软件工程四班

学生学号： 201925220402

学生姓名： 陈浩铭

指导老师： 梁 云

提交时间： 2020 年 4 月 15 日

成 绩：

华南农业大学 数学与信息学院

一、需求分析

1、系统概述：

对于图像的处理在我们日常生活中占有十分重要的地位，而边缘检测更是图像处理中十分重要的一个部分。对于机器学习和深度学习来说，图像中物体的边缘就是一个重要的特征点，没有这样一个重要的特征点，物体识别，人脸识别，数字识别等许许多多的功能都无法实现，而人工智能的能力也会大打折扣。其次，图像边缘检测在分类算法中也有许许多多的应用。除了边缘检测，空间滤波也是十分重要的一个基础领域，在计算机视觉中许多对图像的操作都需要进行滤波等预处理，试图达到最佳状态，才能继续进行操作。

本系统一共有四大主要功能：

1. 边缘检测，可以自定义阈值，可以选择输出最长的前 n 条边，不同边以不同颜色输出，可以以坐标形式查看边缘
2. 空间滤波：可选择三个滤波算子：高斯滤波（平滑图像，去除噪音），拉普拉斯滤波（锐化图像，突出细节，同时保留背景），中值滤波（选择 3×3 卷积矩阵中的中值）
同时，本操作支持撤回（设计只可撤回一步）
3. 颜色填充：根据边缘检测得到的封闭边缘，将图像分区，可以选择填充边缘外，边缘和边缘内的区块颜色，输出图像直接显示填色后的图像（彩色）
4. 颜色反转：使灰度图像的黑白进行颠倒，相应灰度也进行反转

2、系统运行环境：

- (1) 硬件环境： CPU：无限制
内存：1G 以上
显卡：无限制
硬盘：25M 以上
- (2) 软件环境： Windows 8/10
系统支持中文输出

3、功能需求描述：

(1) **数据读取、储存模块：**数据是以 txt 文件读取，以 txt 文件储存。读取文件和储存文件都可以以绝对路径储存或者当前目录下储存

(2) **主操控台模块：**数据读取完毕后，便会进入此模块，在本模块中可以进行本系统的所有功能，包括程序使用说明，更改界面颜色等辅助功能

(3) **边缘检测模块：**是本程序的主要功能之一，用户可以选择默认阈值或者自定义阈值。检测完

边缘后，会输出全部边缘图像，之后用户可以选择输出前 n 条边缘（以长度排序），还可以选择以坐标形式输出

(4) 空间滤波模板： 用户在本模块可以选择高斯滤波，拉普拉斯滤波、中值滤波，操作完后可直接看到滤波结果，同时支持撤回操作，最后用户可以选择是否保存滤波图像（该保存会覆盖原图像）

(5) 颜色反转模板： 用户在本模块可以使灰度图像的黑白进行颠倒，相应灰度也进行反转

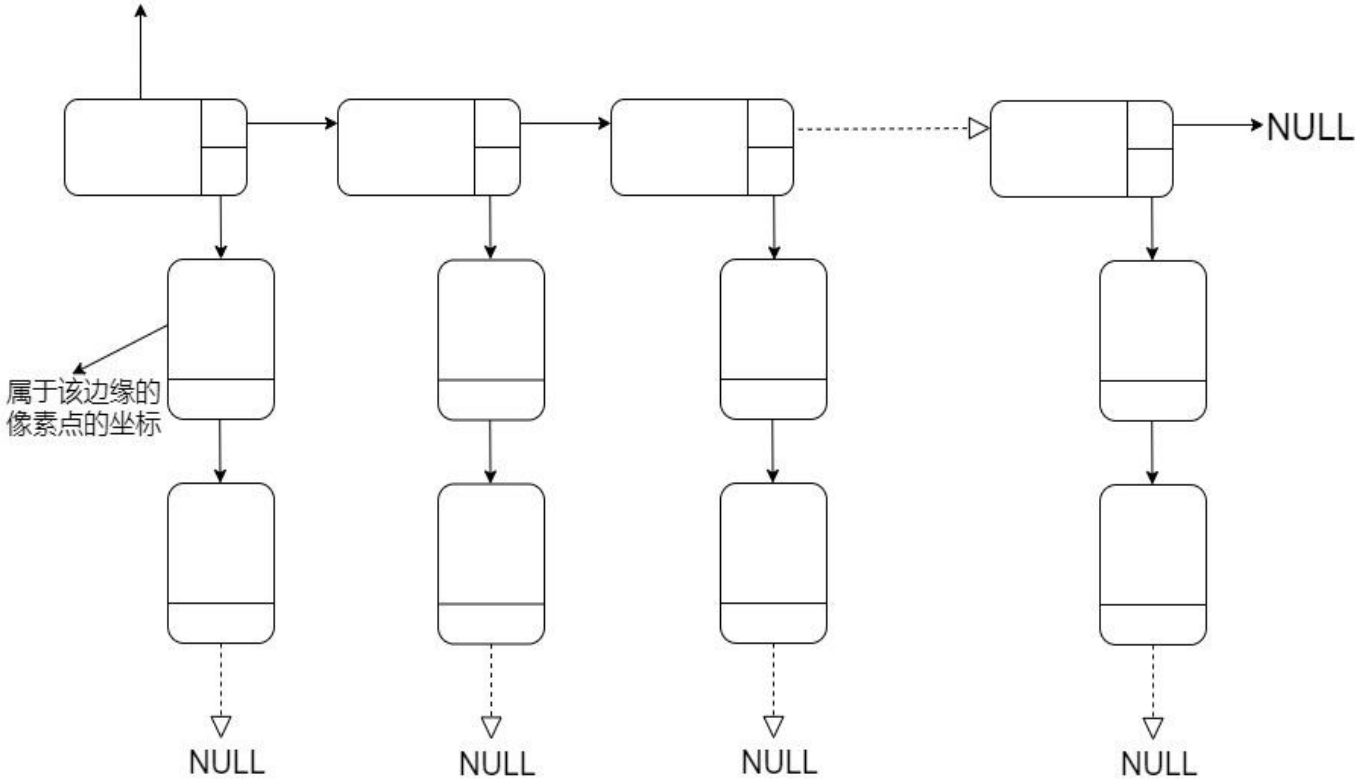
(6) 颜色填充模板： 此模块需要进行边缘检测后才可进行，在本模块会以边缘为界限将图像分为边缘外部，边缘内部，边缘三个区域，用户可选择颜色对三个区域进行填色（理论上可以填 `rgb` 图像中的所有颜色，在此便于观察以 `R`，`G`，`B` 三种颜色代替），操作完后可直接看到填色结果（彩色图像）

二、总体设计

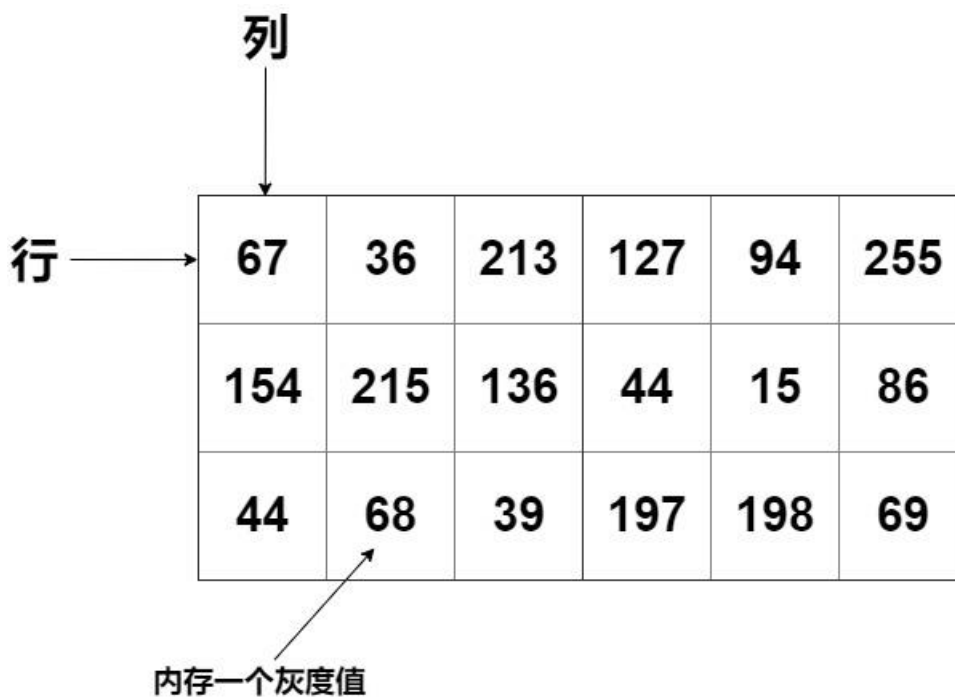
1、总体开发思想：

(1) 主要的数据结构：

一条边缘的头结点，储存该条边缘的长度，当前该节点扩展空间，已使用的空间，储存边缘像素信息



二维链表储存结构



二维矩阵储存结构

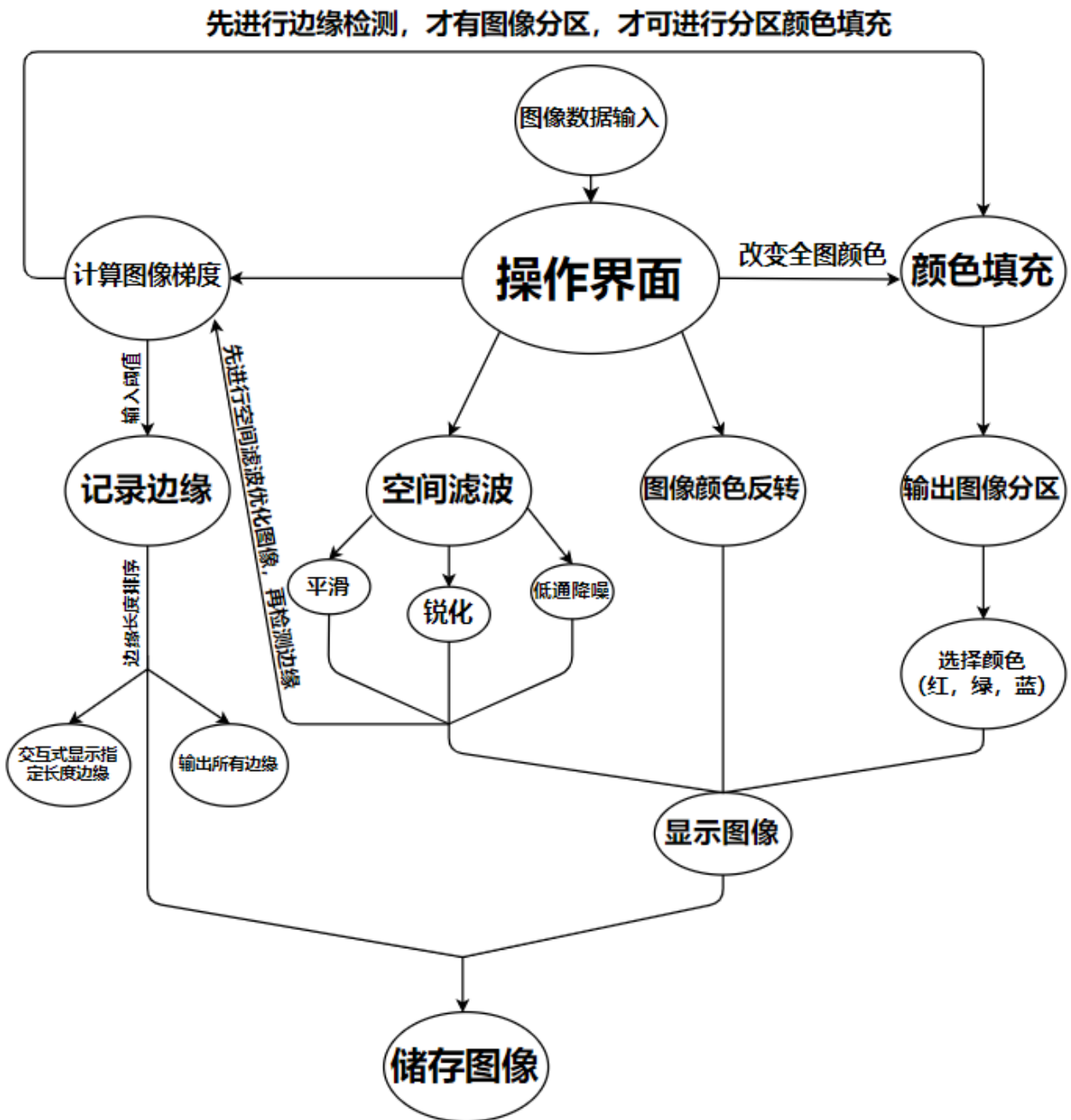
(2) 数据存贮方式:

对于图像使用的二维矩阵进行储存，而对于边缘则使用两个维度都是动态的二维链表进行储存。对于一维的链表结点则是每条边缘的头结点，储存这条边的长度，已扩展的空间，已使用的空间，已经当前已经使用的空间的位置。对于边缘像素每次动态扩展会扩展十个像素点的存储空间，对于单条边每次动态拓展会拓展一条边的储存空间（初始拓展十条边的空间）

(3) 使用的语言:

C/C++

2、系统模块结构图：



3、模块设计： 以下为各个功能模块的流程图

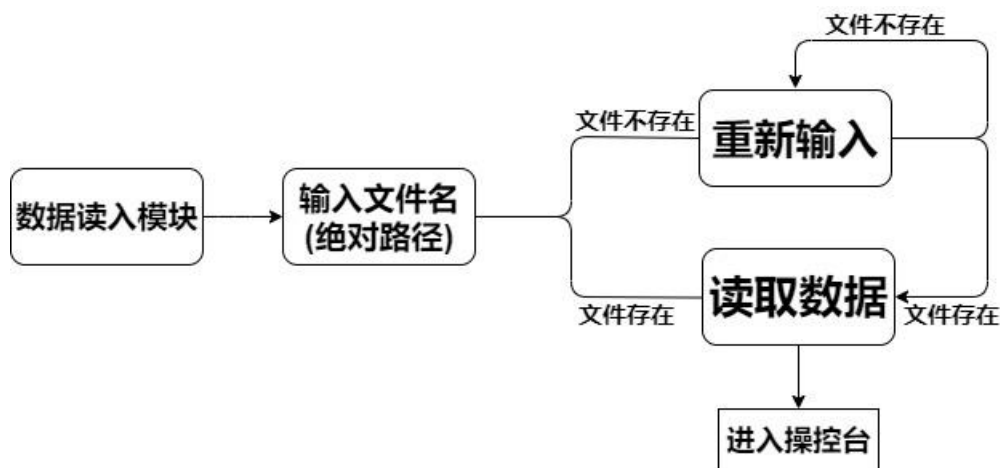


图 1 图像数据输入模块

整个方法执行从文件输入开始，功能流程图如图 1 所示。首先系统进入“数据读入模块”启动数据读入功能，然后给出提示要求用户“输入文件名”，接下来若输入文件名满足“文件存在”则执行读取文件，否则会提示“文件不存在”，将提示用户“重新输入”，直到文件数据正确读入为止。图像数据的输入为后续图像处理提供必要的数据来源。

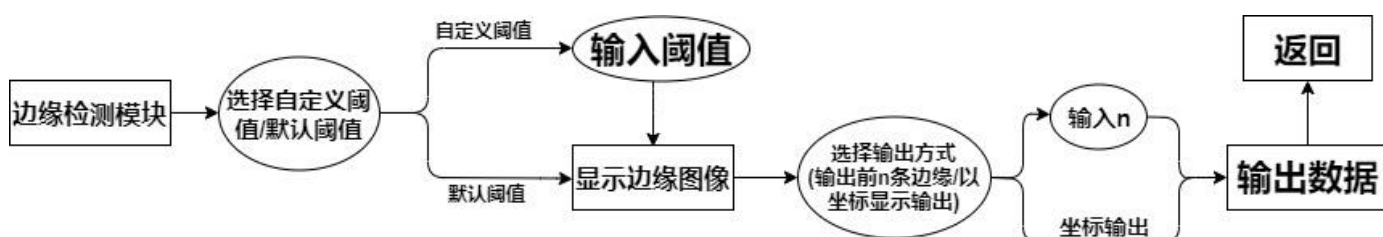


图 2 边缘检测模块

该模块中，首先从选择阈值开始，边缘检测流程图如图 2 所示。首先系统进入“边缘检测模块”开始启动边缘检测功能，然后系统提示用户输入“自定义阈值”或“默认阈值”，接下来会显示出边缘图像，然后系统会提示“选择边缘输出的方式”，若选择“前 n 条边输出”，则继续输入边数 n，或者选择“坐标输出”，之后输出数据，返回主操控台。边缘检测是该系统的主要功能之一，同时也为之后的颜色填充功能提供条件。

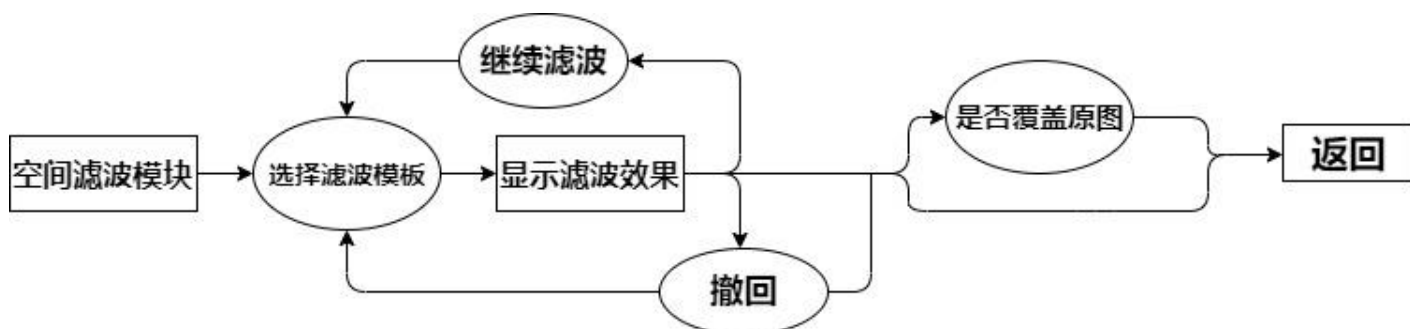


图 3 空间滤波模块

该模块中，首先从选择滤波模板开始。空间滤波流程图如图 3 所示。首先系统进入“空间滤波模块”开始启动空间滤波功能，然后提示用户输入滤波模板，接下来会显示滤波效果，用户可选择继续进行滤波操作，

也可以选择撤回上一步操作，之后用户可以决定是否覆盖原图，最后返回至主操控台。空间滤波可以给图像进行优化，增强边缘检测的准确程度。

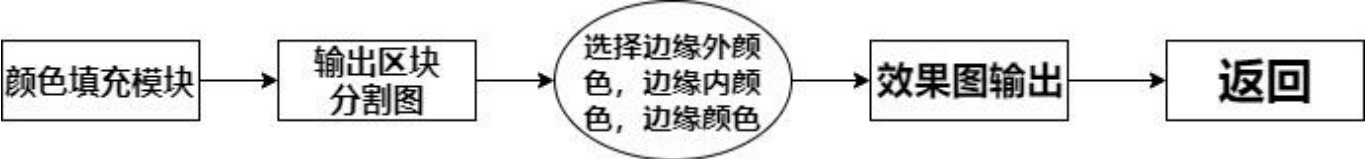


图 4 颜色填充模块

该模块中，首先从输出分割图像开始。颜色填充流程图如图 4 所示。首先系统进入“颜色填充模块”开始启动颜色填充功能，然后显示出“区块分割图”，用户根据提示选择“边缘外部”“边缘内部”“边缘”三个区域的颜色，然后输出效果图，最后返回主操控台

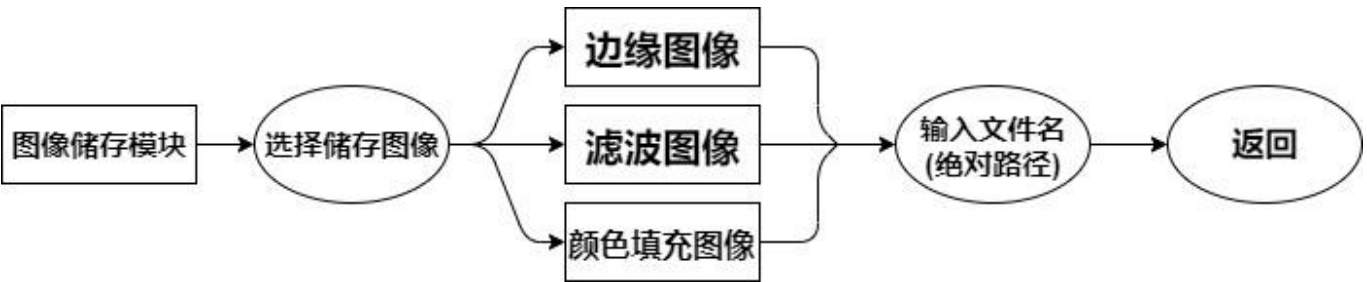


图 5 图像储存模块

在该模块中，首先从选择储存哪一幅图像开始。图像储存流程图如图 2 所示。首先系统进入“图像储存模块”开始启动图像储存功能，然后用户选择储存图像，之后输入文件名保存文件，最后返回至主操控台，图像储存模块是将系统对图像中的操作以文件形式进行保存，是整个功能的意义所在

三、功能实现

1、主要函数：函数名及参数含义；

（1）链表模块主要函数：

`void CreateDataBase(LinkList &L)` //创建储存每条边的空间，初始化开辟储存 20 个点的空间，初始化已开辟空间(Tol)为 20，长度(length)为 0，目前已经使用空间(Loc)为 0

`int CreateLink_L(LinkList &L)` //创建每条边的头结点，初始化创建 10 个头结点

`void Insert_Point(LinkList &L,int i,int j)` //插入储存每条边相连接的像素点坐标，若使用空间已满，动态扩建 10 个像素点的空间，并修改长度(length)，已开辟空间(Tol)，已使用空间(Loc)

`void Insert_Node(LinkList &L)` //若边的头结点已用完，动态拓展结点，每次只拓展一个

`int Sort(LinkList &pHead)` //对检测完的边缘进行排序，使用的是插入排序

（2）主要功能模块：

```
void Color(const unsigned short textColor) //更改界面颜色

void PrintImage() //输出图像

void ReadData() //读取图像文件数据

void Print_Edge(LinkList L, int n) //以各种形式输出边缘

void Save_Edge() //储存边缘图像

void Edge_Detection() //检测边缘

int cmp(const void *a, const void *b) //快排比较函数

int Media_filter(int ch[]) //中值滤波

void Spatial_Filtering() //空间滤波功能

void Color_Reversal() //颜色反转功能

void Color_Fill() //颜色填充功能

void Change_Color() //重新更改界面颜色函数

void SaveFile_Edge() //以 txt 文件储存边缘图像

void SaveFile_Filter() //以 txt 文件储存空间滤波图像

void SaveFile_ColorFill() //以 txt 文件储存颜色填充函数

void Instructions() //程序说明文档

void MainControl() //主操控台
```

2、使用界面

(1) 欢迎/退出画面:
欢迎:

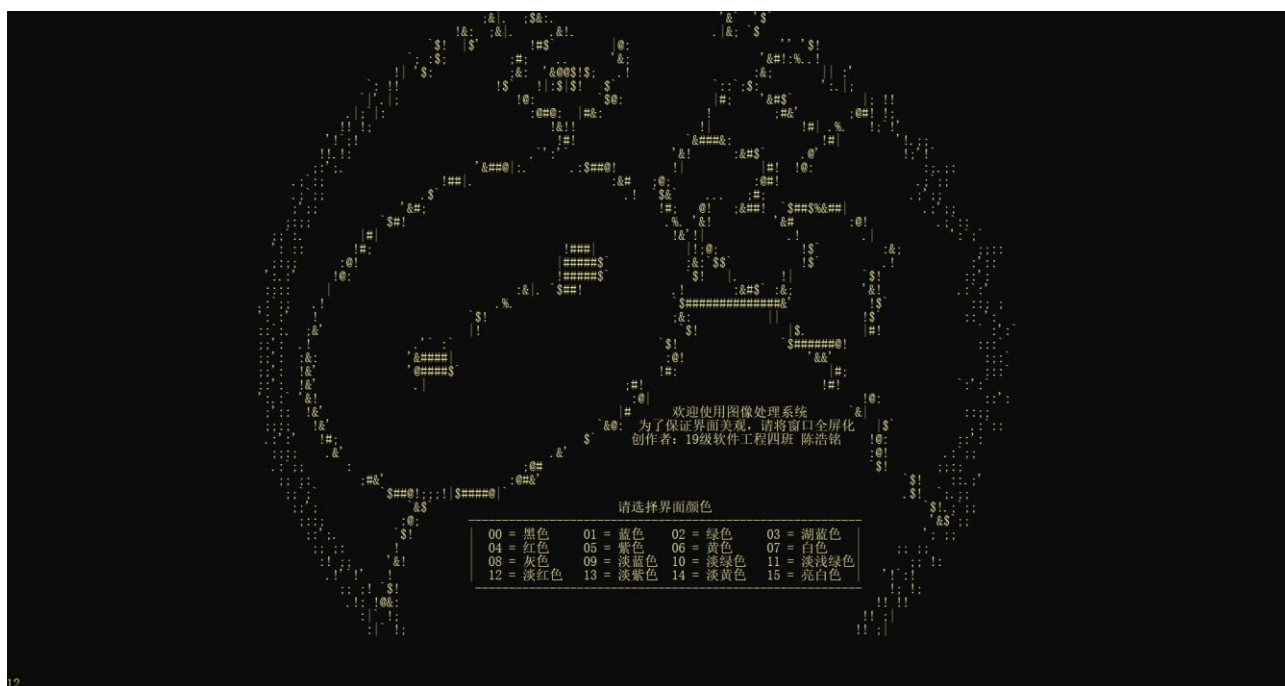


图 6 欢迎界面

根据图 6 界面的提示, 选择界面的颜色



图 7 退出界面

(2) 图像数据读入界面 (成功/失败):



图 8 数据读入界面

根据图 8 的界面提示, 用户需输入待处理图像文件的名称, 图像名称输入有 2 种, 输入绝对路径输入 (即输入图像的完整路径输入) 和输入相对路径 (代码目录下的文件名)。每个文件名必须以.txt 结束。系统提供的自带文件包括 image1.txt (50x70), image2.txt (80x70), image3.txt (200x300)。操作成功后, 系统会直接显示图像的所有数据 (如图 9), 每个数据值在 0~255 内。

[illegible]

根据界面提示，按回车键继续，之后会跳转至主操控台

该文件不存在，请重新输入

输入格式可为绝对路径(例如:E:/photo/image.txt)或本目录中已有文件(例如:image.txt)
程序自带测试文件(image1.txt/image2.txt/image3.txt)

根据界面提示, 用户需要重新输入正确的相对路径或者绝对路径, 若文件依旧不存在, 则仍会显示图 5 界面, 直至文件输入成功, 跳转至输出图像界面 (图 9)

(3) 主操控台：

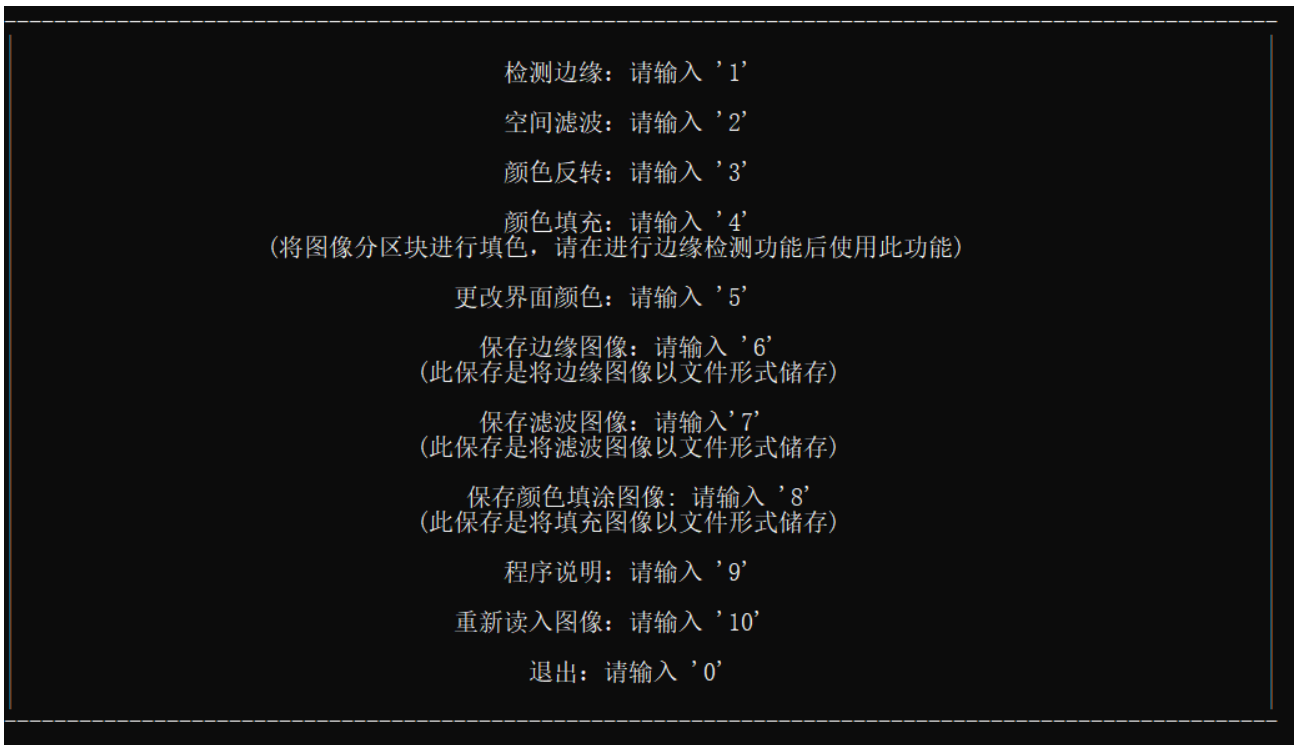


图 11 主操控台界面

根据界面提示，用户在本界面可以选择相应的功能，输入相应的数字后就会进入相应功能的界面

(4) 图像边缘检测：

选择自定义阈值/默认阈值：

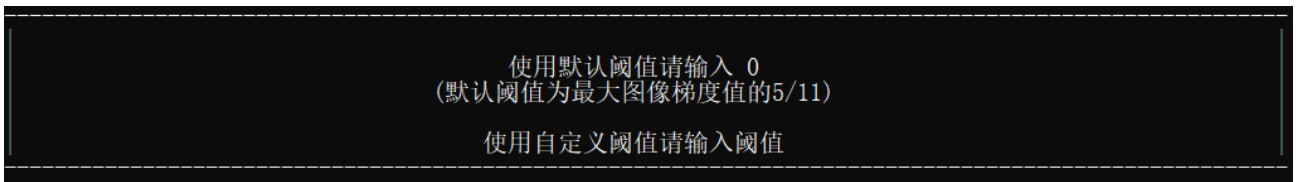


图 12 边缘检测界面

根据界面提示，用户在本界面可以选择默认阈值（图像最大梯度值得五分之一），或者选择自定义阈值，之后会跳转至图 13 显示边缘图像

[illegible]

图 13 检测边缘效果

如图 13 所示，显示出边缘检测的效果，并且输出使用的 sobel 算子，对于边缘图像，以“0”作为背景，“#”作为边缘，并且标注背景为白色，边缘为蓝色，按回车继续后跳转至选择输出边缘方式界面（图 14）

选择输出形式:

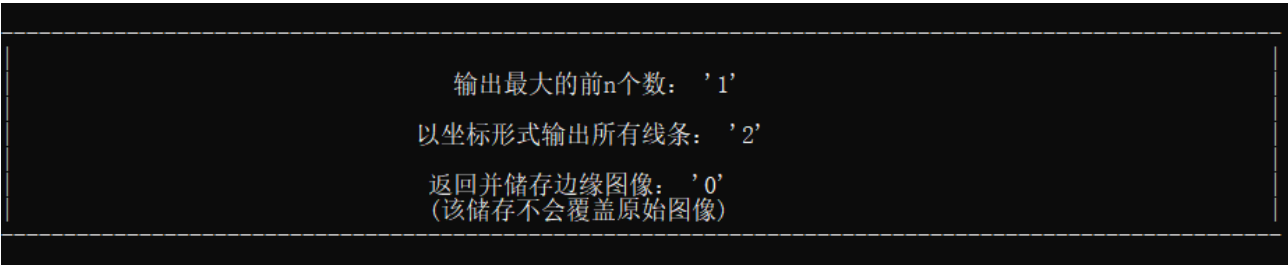


图 14 选择图像输出方式界面

如图 14 所示，用户根据提示选择图像输出方式，若选择“输出前 n 条边”，就会跳转至“输入边缘数”界面（图 15），若选择“坐标形式输出”，就会跳转至“坐标数据输出图像”（图 17），若返回，便会保存边缘图像并显示保存的边缘图像（图 18）

前 n 条边:

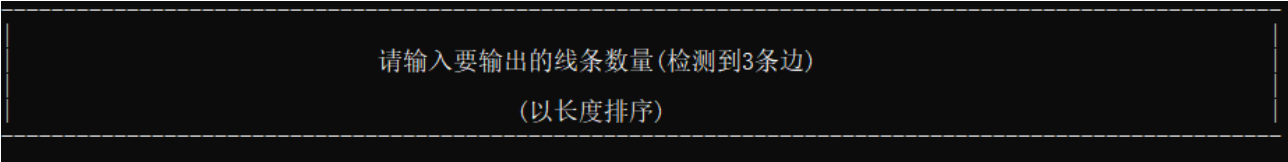


图 15 输入边缘数量界面

如图 15 所示，用户根据提示输入要输出的线条数量（边缘是以长度进行排序），输入的数字应该小于或等于所检测到的边缘数，之后会显示输出的边缘图像（图 16）

(5) 空间滤波功能:

操作界面:

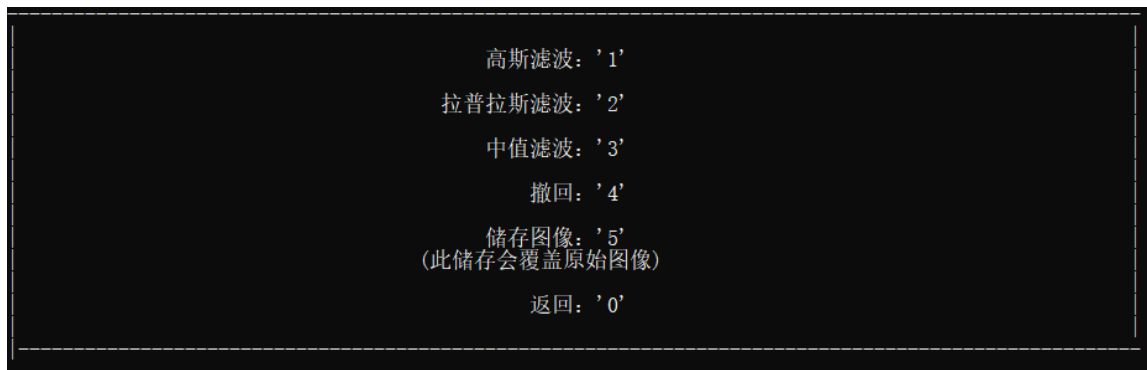


图 19 空间滤波界面

如图 19 所示, 用户可选择使用的滤波算子, 同时还以可以选择撤回操作和覆盖原图操作, 返回会回到主操控台界面 (图 11)

高斯滤波:

```
使用的算子为:
0.011, 0.084, 0.011,
0.084, 0.619, 0.084,
0.011, 0.084, 0.011
高斯滤波的效果图像是:
254 254 254 254 254 254 254 254 254 254 254 254 254 253 253 254 254 254 254 254 254 254 254 254 254 254
254 254 254 254 254 254 254 254 254 254 253 253 254 254 254 253 253 254 254 254 254 254 254 254 254 254
254 254 254 254 254 254 254 254 254 254 253 253 253 253 253 253 254 254 253 253 254 254 254 254 254 254
254 254 254 254 254 254 254 254 254 254 253 253 253 253 253 253 253 253 253 253 254 254 254 254 254 254
254 254 254 254 254 254 254 254 254 253 253 253 252 252 253 253 253 253 253 253 254 254 254 254 254 254
254 254 254 254 254 254 254 254 253 253 253 253 253 253 253 253 253 253 253 254 254 254 254 254 254 254
254 254 254 254 254 254 254 253 254 253 253 254 252 253 250 253 250 252 253 253 253 252 253 251 251 252 254 254 254 254
254 254 254 254 254 254 254 254 253 252 253 250 253 250 252 253 253 253 252 253 251 251 252 254 254 254 254 254
254 254 254 254 254 254 254 254 250 253 254 250 253 253 253 254 254 254 254 250 253 250 253 253 250 253 254 254 254
254 254 254 254 254 254 254 254 254 253 252 254 253 252 254 253 253 253 254 250 249 253 253 248 252 254 254 254 254
254 254 254 254 254 254 254 254 252 250 229 222 226 227 227 227 227 227 226 227 247 253 254 253 254 254 254 254
254 254 254 254 254 254 254 254 248 206 51 55 53 53 53 53 54 51 50 206 251 253 254 254 254 254 254
253 253 253 253 253 253 253 230 50 48 205 222 226 226 223 226 206 48 50 227 251 254 254 254 254 254
253 253 253 253 253 253 251 209 69 208 247 253 253 248 253 253 249 250 208 69 209 250 252 253 254 254 254 254
253 254 253 254 253 248 211 92 211 249 253 253 252 252 253 253 253 246 211 93 208 249 250 253 252 254 253
253 254 250 252 250 209 94 209 250 254 254 253 250 249 253 253 254 250 249 250 210 91 211 250 252 253 253 251
252 254 252 250 211 90 212 250 247 253 253 253 253 252 253 253 252 253 253 247 208 90 208 248 253 254 252
253 253 243 208 69 183 223 225 226 227 227 226 225 227 227 226 227 227 224 225 226 224 184 69 207 248 254 253
253 251 210 67 12 50 54 55 54 54 57 54 57 54 53 57 54 54 57 54 54 48 12 69 211 249 250
254 231 93 184 50 201 224 227 226 223 221 227 227 226 223 227 227 224 227 227 203 49 185 94 232 253
253 250 230 224 53 222 252 250 253 250 252 252 251 253 252 252 252 254 253 253 252 252 226 55 224 232 249 254
253 252 253 227 54 226 250 253 251 250 253 250 251 254 254 254 254 253 251 254 253 249 225 55 227 252 253 253
254 254 253 225 53 227 253 253 250 251 254 251 251 253 252 253 253 253 253 253 253 226 54 227 253 254 253
254 254 252 225 53 227 250 226 226 227 227 226 224 229 250 253 253 253 253 253 253 226 53 227 253 254 253
254 254 252 226 54 227 230 56 55 51 33 53 54 54 230 253 253 253 253 253 253 253 226 53 227 253 254 253
253 254 252 226 54 227 227 51 201 203 53 203 200 54 226 253 253 253 253 253 253 226 53 227 253 254 253
253 254 253 226 54 227 227 54 203 203 48 200 203 52 223 253 253 253 253 253 253 226 53 227 253 254 253
253 254 253 226 54 227 227 34 49 50 12 49 52 33 226 253 253 253 253 253 253 226 53 227 253 254 254
253 254 253 226 54 227 226 51 203 200 49 198 203 51 224 253 253 253 253 253 253 226 53 227 253 254 254
253 254 253 227 54 226 225 50 202 201 48 202 203 54 227 253 253 253 253 253 253 226 54 227 253 254 254
253 254 253 227 54 227 230 53 51 51 34 52 54 54 229 253 253 253 254 254 254 227 54 227 253 254 254
253 254 253 227 54 227 251 228 227 227 224 226 225 227 251 253 253 253 254 254 254 227 54 227 253 254 254
253 254 253 227 54 227 254 252 250 253 253 250 248 252 252 253 253 253 254 254 254 227 54 227 253 254 254
253 254 253 226 54 227 254 253 253 252 253 253 252 248 253 253 253 253 253 253 227 54 227 253 254 254
253 254 253 226 54 227 254 251 253 253 250 249 251 253 254 253 253 253 253 253 227 54 227 253 254 254
253 254 253 226 53 227 254 253 253 251 253 253 252 253 252 253 253 253 253 253 226 54 227 253 254 254
253 253 253 226 53 227 254 252 254 254 252 249 253 249 253 253 253 253 253 253 226 54 227 253 254 254
253 253 253 226 53 227 253 254 252 254 254 252 254 254 252 254 253 251 254 253 253 253 227 54 227 253 254 254
252 253 254 225 51 203 226 226 226 227 227 226 227 226 226 227 227 227 227 227 227 203 52 224 252 253 252
253 251 252 226 53 51 54 54 54 54 54 54 55 55 55 55 55 55 52 55 229 253 251 253
253 253 249 250 230 227 227 227 227 227 227 227 226 226 226 226 226 226 226 226 228 247 252 253 253
249 253 253 253 253 250 254 254 254 254 254 254 254 254 254 254 254 254 254 253 253 254 253 254
253 253 252 253 251 253 254 254 254 254 254 254 254 254 254 254 254 254 254 253 253 253 253
```

图 20 高斯滤波效果图

如图 20 所示, 输出的是高斯滤波后的效果图, 按回车键继续, 到“空间滤波界面” (图 19)

拉普拉斯滤波:

[illegible]

图 21 拉普拉斯滤波效果图

如图 21 示, 输出的是拉普拉斯后的效果图, 按回车键继续跳转到“空间滤波界面”(图 19)

中值滤波:

中值滤波的效果图像是:

[illegible]

图 22 中值滤波效果图

如图 22 示, 输出的是拉普拉斯后的效果图, 按回车键继续, 会跳转到“空间滤波界面”(图 19)

覆盖原图：



图 23 覆盖原图的图像

如图 23，输出覆盖原图的图像数据，按回车键继续，会跳转到“空间滤波界面”（图 19）

(6) 颜色反转：

输入边缘外，边缘内，边缘颜色：

请选择边界外部颜色、边界内部颜色、边界颜色：R(红色)/G(绿色)/B(蓝色)

输入格式：依次输入边界外部、边界内部、边界颜色，中间空格隔开
(例如输入：R G B)

图 26 选择区域颜色界面

如图 26 所示，根据页面提示选择区域颜色，以 R，G，B 代替红色，绿色，蓝色，输入完成后跳至输出效果图（图 27）

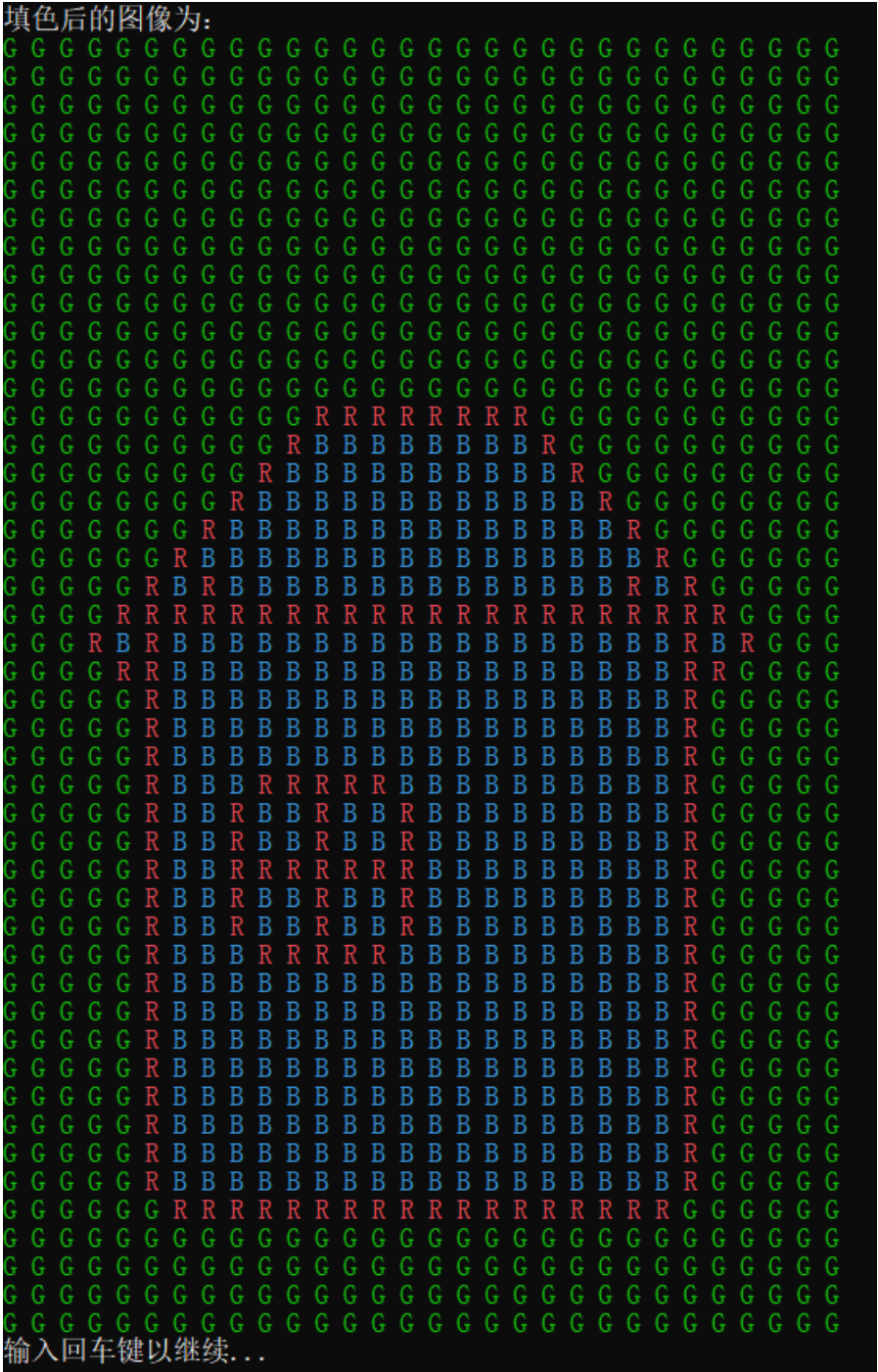


图 27 输出效果图

如图 27 所示，输出填色后的效果图，不同颜色的区域以不同颜色来标注，同时也以 R G B 来显示各区域的图像，按回车继续，跳至保存界面（图 27）

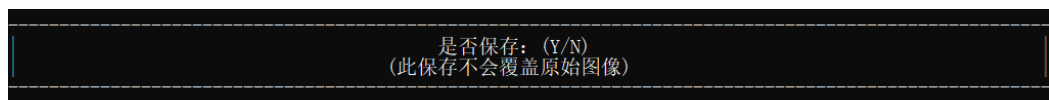


图 28 填充图像保存界面

如图 28 所示，选择是否保存图像，之后跳转至主操控台（图 11）

（8）储存图像：



图 29 图像储存界面

如图 29 所示，用户需输入待保存文件的名字，图像名字输入有 2 种，输入绝对路径输入（即输入图像的完整路径输入）和输入相对路径（代码目录下的文件名）。每个文件名必须以.txt 结束。之后会返回至主操控台（图 11）

四、总结

在三月末便拿到了题目，一开始其实脑子里是有一些乱的，功能有一些繁琐，一开始也不知该如何储存边缘，并且还要对检测到的边缘进行排序，和用户交互式输出。等等一系列的功能让我的脑袋乱如麻。而这次的编程和以往是不一样的，以前的编程都是只有几十行最后一两百行的小程序，完全不需要思考代码规范，模块化问题，甚至将所有代码都写在 main 函数中也不会出错。但是这次不一样了，这次是设计一个文件读取储存的实实在在的可使用的程序，代码少说也有一千多行，如果全部都写在 main 函数，后期的调试，修改，优化将会完全无从下手。思考了许久之后，我决定不从代码下手，而从流程图下手，先把流程，功能之间的关系理清楚了，那么将流程转换成代码也就不难了，而代码之间的关系也就没有那么乱了。

果然，磨刀不费砍柴功，虽然先花了一些时间再制作流程图上，但是通过流程图再来看这道题目，思路一下子就清晰了许多，应该先写主操控台的函数，再由主操控台像网状一样蔓延开来。准备开始写边缘检测的时候，一开始准备使用静态链表，开辟一个足够大的空间用来储存边缘线条，虽然不需要担心动态开辟空间的问题，并且在边缘长度排序方面可以使用快排等算法，但是，一幅图像的边缘数量并不是一个可以大概估计的值，边缘长度也不是一个可以估计值，它取决于读入图像的大小以及图像里的内容，所以对于静态的二维链表，很有可能空间浪费或者空间溢出，所以最后我还是选择了动态的二维链表，虽然需要考虑拓展空间，并且 qsort 等算法用不了，但是空间得到了最大化的利用并且也不需要担心空间溢出的问题，再者，一幅图像的边缘即使全部像素各自为一条边，那么一幅图像的边缘数也只是十万级别，qsort 算法优化的时间并不多，所以权衡利弊，我果断选择动态二维链表。

到后面，在边缘检测上又出现了问题，一开始对于边缘检测的方法是模拟点在边缘上“行走”，若该像素点的八个方向上其中一个也是边缘像素点的话，便将其储存，并且走到该像素点上，可是这样的问题是，如果该像素点的八个方向上有不只一个属于边缘的像素点的话，程序就会选择其中一个方向“走”，这样就是从中间“截断”了边缘，这样一条边缘有可能就被截断很多次，那么这个边缘的长度就没有意义了。而且，在颜色填充上，使用这样的“行走”的方法来确定区域的话，如果以四个方向搜索，在斜的边缘上边会终止，如果以八个方向搜索的话，在有角的地方就会把边缘内部和外部都视为一个区域。最后，学习了深度优先搜索和广度优先搜索后，我发现这个算法不就是可以用于我的边缘检测了吗。于是这样一个

回溯的检测方法，使每一个边缘都不会被截断，并且可以达到单像素的精确检测。在边缘填充上，只需要四个方向就可以实现点的“行走”，并且在斜的线上不会终止（因为会回溯）。就这样，遇到问题，先不慌，寻找解法，终究会得以解决。

当所有功能都实现完了后，我又遇到了问题，当边缘检测完了后，背景设为任意字符都很难和边缘区分开来，看得眼睛十分难受，同时颜色填充也是，单纯的以 **R, G, B** 来代替红，绿，蓝并不能让人能区分开每个区块。于是我想到，我们看到图片，之所以能辨认这是边缘，不就是因为边上的颜色和背景颜色不一样吗（或者灰度值相差很大），那在这个程序中，如果我能改变输出时边的颜色，那不自然就能很轻松地辨认处背景和边缘了吗，如果可以改变多种颜色，那么不属于同一条边地像素点也就可以区分开来了吗。果然，在网上我找到了在 `windows.h` 这个库里面，有可以改变输出颜色的函数。

最后，这个完整的程序终于写完了，收获十分多。首先，在编程的时候应该先规划好编程的流程，将代码分为几个模块进行编写，这样十分有条理性，并且在出错是也能很快就找到是哪个函数出现错误。其次，遇到问题不要慌，先仔细想好问题出在哪里，逻辑矛盾又出现哪里，再根据这个问题和矛盾去寻找最优的解法。最后，设计好程序后，应该换位思考，如果是使用者，有可能在哪里不懂该程序的操作，根据这个思路去设计提示，并且该程序的界面是否友好？是否又可操作性？是不是应该重复读取图像？结果是否便于观察（如一开始我的边缘图像不易观察）？解决完以上的问题后，程序基本上就可以使用了。

评分表	
题目：图像的边缘识别和处理问题	
学号：201925220402	姓名：陈浩铭
评分项目	分数
1. 功能是否满足要求（50 分）	
2. 界面是否美观（10 分）	
3. 所学知识在设计中的运用是否得当（10 分）	
4. 对效率要求较高的部分是否有更深的思考与设计（10 分）	
5. 答辩情况（20 分）	
总成绩：	
评语：	
系统代码实现：完成题目要求的主要功能，如:边缘计算统计和图像平滑；	
界面设计：简单、美观、直接；	
知识运用：运用链表、结构体、矩阵；	
效率方面：使用了排序搜索，矩阵相乘，代码清晰明了，注释得当准确；	
报告撰写：符合规范，功能设计合理，算法流程恰当，结果描述准确；	
答辩情况：	
其他：系统有一定复杂度和代码量，探索了图像区域选择和翻转，实现了多种滤波的平滑，实现了边缘长度的计算和边缘点的记录，结果显示比较丰富。	
。	
评阅老师：	评阅时间： 年 月 日