

南京理工大学电子工程与光电技术学院

C++课程设计实验报告

班 级_____9181040G08_____

学生姓名_____黄海浪_____

学 号_____9181040G0818_____

任课教师_____赵琦_____

时 间_____2019-03-23_____

所选题目_____7-4 特殊图像处理_____

难易程度_____A_____

报告成绩_____

1. 实验内容:

1.1 程序功能介绍:

特殊图像处理, 可以对任意大小 256 色 bmp 格式图像进行柔化、锐化、浮雕、扩散、马赛克效果的处理。

1.2 设计原理及思想:

BMP (全称 Bitmap) 是 Windows 操作系统中的标准图像文件格式, 使用非常广泛。BMP 文件存储数据时, 图像的扫描方式是按从左到右、从下到上的顺序。由于 BMP 文件格式是 Windows 环境中交换与图有关的数据的一种标准, 因此在 Windows 环境中运行的图形图像软件都支持 BMP 图像格式。

通过 C++ 头文件 Windows.h 中提供的结构体 RGBQUAD、BITMAPFILEHEADER、BITMAPINFOHEADER 能够实现对 bmp 图像进行处理。

柔化算法: 取图像周围块颜色取平均值作为当前像素点颜色。

锐化算法: $\text{new_color} = \text{original_color} + X * \text{difference}$, X 为锐化系数, difference 为相邻像素差值。

浮雕效果: $\text{new_color} = \text{difference} + 128$

扩散效果: 在像素点周围 $x * x$ 的像素块中随机一点作为当前像素值。

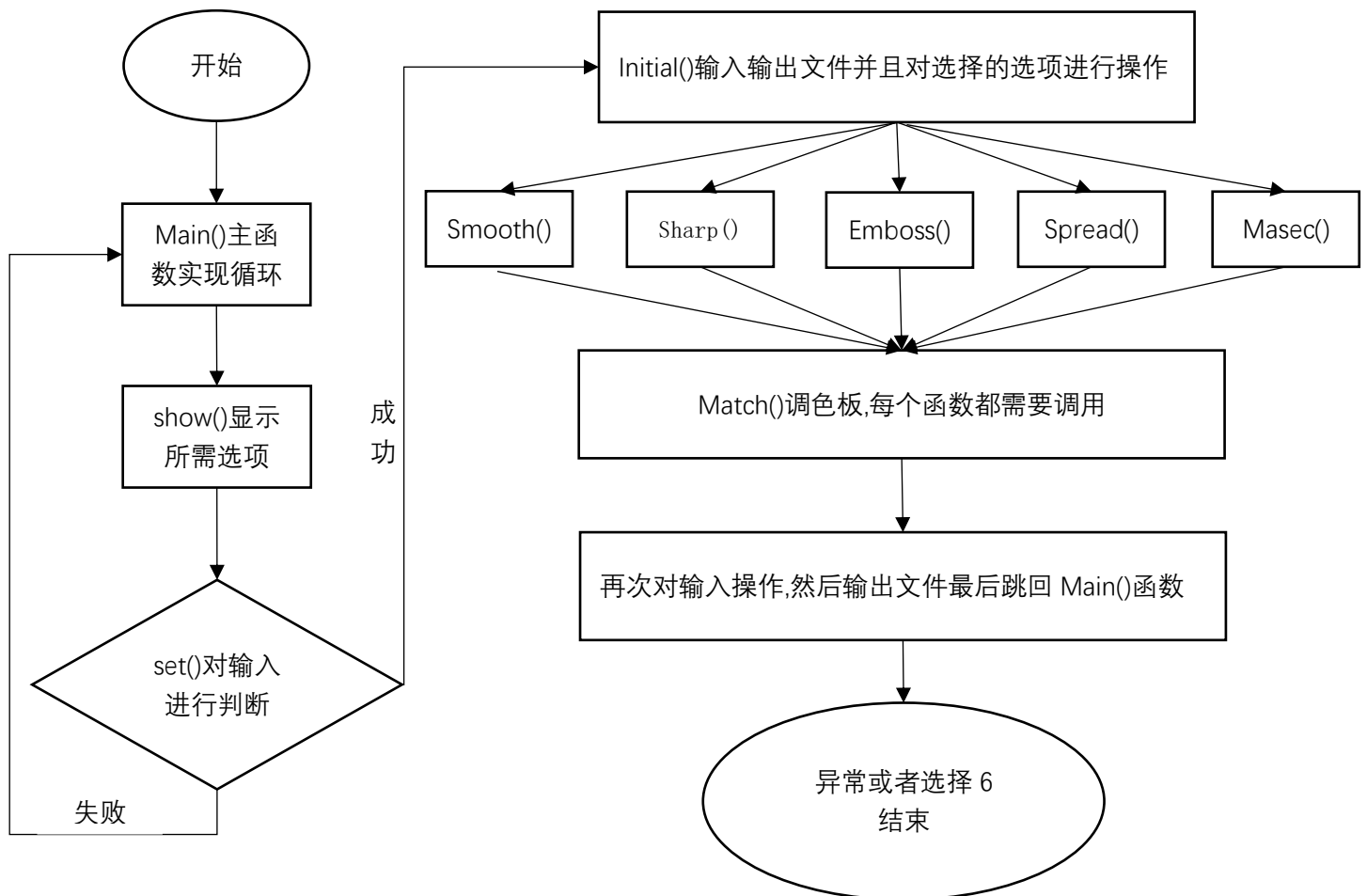
马赛克效果: 找到一个点 (x, y) , 向右向上延伸 (m, n) , 把区域内的点分为 x 块, 块内每个像素颜色取其块颜色平均值。

边缘处理: 均是完美的边缘处理 (从测试图暂时没发现 bug)

1.3 程序设计要求

- (1) 扩展程序功能, 可以根据所输入的图像而对任意大小的位图进行处理。
已完成
- (2) 对图像进行任意位置及大小的马赛克处理。
已完成
- (3) 对图像进行加强柔化及锐化效果的处理。
已完成
- (4) 对图像进行扩散效果的处理 (即将像素周围的某点值随机赋给该像素)。
已完成

2. 程序流程图分析与说明：



注意：部分错误以及不按照引导操作会跳回主函数，操作失败等错误会退出程序

3. 函数以及部分算法：

1. 通过 windows.h 的结构体 RGBQUAD、BITMAPFILEHEADER、BITMAPINFOHEADER 实现对图像的处理。
2. 实现了动态建立数组处理任意大小图片

```
infile.read((char*)& strHead, sizeof(strHead));
infile.read((char*)& strInfo, sizeof(strInfo));
//获取图片文件信息，动态建立二维数组
nShare = strInfo.biHeight;
nWidth = strInfo.biWidth;
char** naImage = new char* [nShare];
for (unsigned int i = 0; i < nShare; i++)
    naImage[i] = new char[nWidth];

//释放动态创建的数组
for (unsigned int i = 0; i < nShare; i++)
    delete[] naImage[i];
delete[] naImage;
```

3. #include<time.h>，加入计时器，获取处理用时。
4. 各个函数以及其作用

```
//函数的声明
int Match(int nRed, int nGreen, int nBlue);           //寻找与像素匹配的调色板的序号
int Smooth(ifstream& infile, ofstream& outfile);    //柔化效果
int Sharp(ifstream& infile, ofstream& outfile);      //锐化处理
int Emboss(ifstream& infile, ofstream& outfile);     //浮雕效果
int Spread(ifstream& infile, ofstream& outfile);     //扩散效果
int Masec(ifstream& infile, ofstream& outfile);      //马赛克效果
int Initial(char nChoice);                           //初始化图像文件
```

5. 边缘处理（一个例子：柔化效果，其他均同理计算）
处理共四步：
 1. 从左下开始到所输入点的阈值
 2. 处理上方的点，横着继续使用阈值，竖着就按照剩余点进行
 3. 处理右方的点，竖着继续使用阈值，横着就按照剩余点进行
 4. 处理剩下方块，均按照剩余点进行计算
6. 部分小创新避免 bug
 1. 对输入进行数组操作，判断是否多输入，避免输入多次后还能运行的 bug，导致程序不正常运行。
 2. 加入自动退出/强制退出时间等待，避免测试时的“闪退”，当然也可以使用 system("pause")

4. 实验数据（软件界面以及结果）：

例子：扩散处理

运行结果：

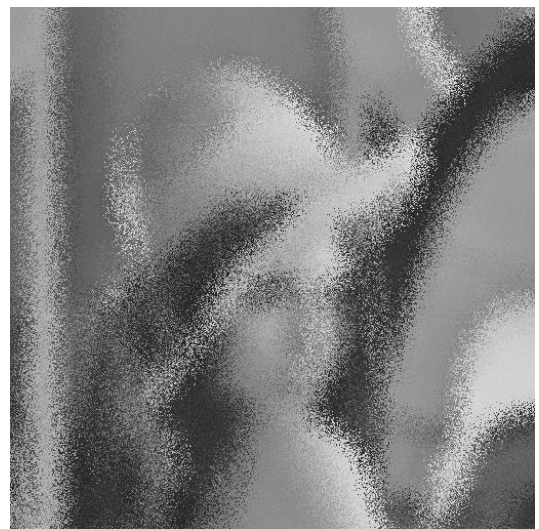
```
F:\0000000000000000files\大学相关\大一下\作业及考试\C++\课程设计\9181040G0818\数据文件\VKCSJ_Test.exe
处理完成!
请选择:
    1. 柔化
    2. 锐化
    3. 浮雕
    4. 扩散
    5. 马赛克
    6. 退出
4
输入的bmp文件名字(如123.bmp): 原图.bmp
输出的bmp文件名字(如321.bmp): 扩散.bmp
图片长度: 512 宽度: 512
请输入扩散系数X, 越大越扩散 X范围:
理论大小: 0<X<=512 建议大小: 2<X<=20
16
处理中...
处理用时: 0.038s
处理完成!
请选择:
    1. 柔化
    2. 锐化
    3. 浮雕
    4. 扩散
    5. 马赛克
    6. 退出
```

处理结果：

原图



扩散（系数 30）



例子：浮雕处理

运行结果：

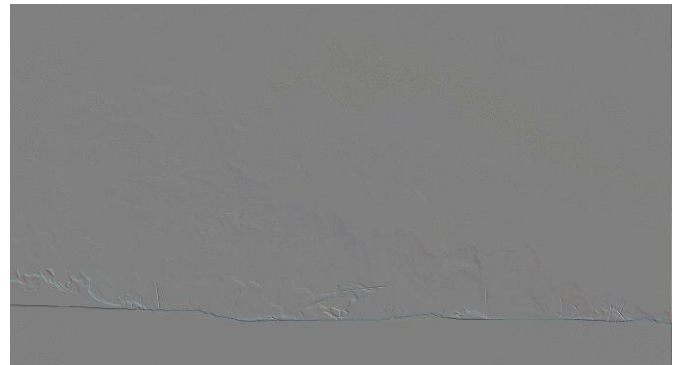
```
F:\0000000000000000files\大学相关\大一下\作业及考试\C++\课程设计\9181040G0818\数据文件\VCKCSJ_Test.exe
4
处理中...
处理用时: 0.184s
处理完成!
请选择:
    1. 柔化
    2. 锐化
    3. 浮雕
    4. 扩散
    5. 马赛克
    6. 退出
3
输入的bmp文件名字(如123. bmp): 原图. bmp
输出的bmp文件名字(如1321. bmp): 浮雕. bmp
图片长度: 512 宽度: 512
处理中...
处理用时: 0.196s
处理完成!
请选择:
    1. 柔化
    2. 锐化
    3. 浮雕
    4. 扩散
    5. 马赛克
    6. 退出
```

处理结果：

原图



浮雕



注：更多输入见附带文件

5. 实验心得:

问题:

1. 兼容问题: 首先更改源程序时有一个 vc6.0 与 vs2010 以上版本不兼容问题, 需要更改地方较多, 虽然很复杂, 不过这样再次加深对程序的理解与印象。
2. 算法问题: 部分算法有时候写起来比较困难, 尤其是边缘处理, 出现多次数组下标越界和无法完成边缘处理的任务, 经过多次调试才正确并达到效果。
3. 任意图像问题: 由源程序的结构体查资料找到了 windows.h 里面自带的结构体后, 通过查找资料完成对输入的任意图像大小进行信息的获取。再使用动态创建二维数组, 最后带入算法即可。

任意图像不足处: 没有进行对申请空间的判断可能导致程序出错, 由于此程序所需内存空间极小且目前基本上所有电脑配置均满足此软件, 故舍去对申请空间是否成功的判断。

4. 由于编辑器问题或者编译器问题, 编码解码可能不同, 可能出现源程序编码无法正常显示中文, 建议全部改为英文后再在其他地方运行。

感想:

程序基本完美, 看见自己的程序能够跑起来, 很开心。处理错误以及查找资料让自己对 C++ 有了更多认识, 让我更加熟悉, 奠定了基础。

有时候觉得很苦, 但是最后能够让它基本完美的跑起来, 心中成就感爆棚哈哈。