

# 深圳大学考试答题纸

(以论文、报告等形式考核专用)

二〇二一 ~ 二〇二二 学年度第 2 学期

课程编号 1500760001-01 课程名称 《数字图像处理》 主讲教师 吴惠思 评分

学 号 2020151036 姓名 谢弘烨 专业年级 2020 级 软件工程

教师评语:

项目名称:

图像处理系统

评分标准:

得分点	评分标准	分值	得分
基本功能模块	A. 颜色（灰度）变换（例如：直方图均衡化、点操作、伽马校正、分段函数、像素级代数运算、像素级逻辑运算、位平面分割等）（15 分） B. 图像滤镜增强（例如：邻域操作、线性与非线性滤波、图像平滑、图像锐化等）（20 分） C. 空间几何变换（例如：图像平移、图像缩放、图像旋转自由变换组合等）（15 分） 注：以上 ABC 每组模块至少选择完成 1 项基本功能，与小实验中的功能应有不同。	50	
特色功能模块	结合课程内容及实验的特色扩展模块（自行设计）。	25	
系统界面	系统界面简洁美观，功能交互友好。	5	
报告书写	系统各模块功能、算法和结果描述清晰，格式规范，内容完整。	10	
PPT 答辩 (含程序演示)	PPT 编写美观、有逻辑性，程序演示、口头表达清晰准确。	10	
		总分	

# 1 图像处理系统简介

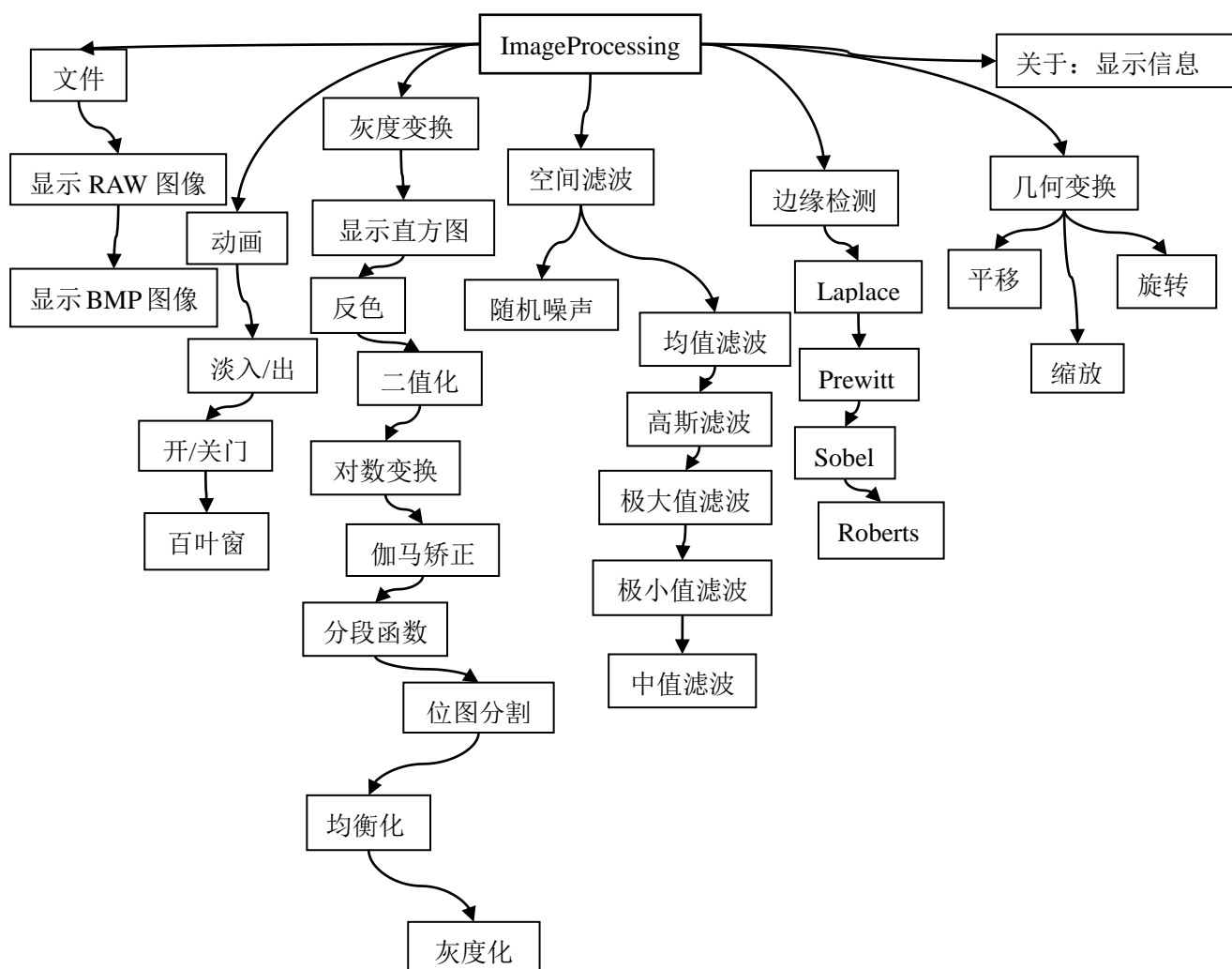
## 1.1 意义及目标

经过一个学期的学习和实验，我对图像处理有了一定的掌握和心得。加之期末临近，大作业也布置下来。将之前的实验结果综合起来稍加改进就能够提交，同时是作为我个人的第一个真正意义上有点作用的带界面程序。

虽然因为时间原因最后呈现出来的还是十分简陋，而且实用价值并不比其他课上的作业高多少，但还是很有成就感。

## 1.2 主要功能模块架构

由于时间关系，没有能够在图形界面上进行美化、设计用户系统等，仅做到了综合之前实验中实现的各种功能。



## 2 项目基本功能模块

### 2.1 文件

在文件模块中，用户可以在本地选择受支持的图片格式（本程序仅支持 .raw 和 .bmp 24 位 两种格式）。

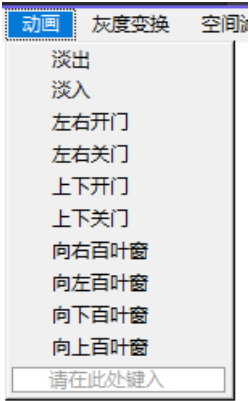


```
case IDM_SHOWRAWIMAGE:
    OpenImageFileDlg("打开图像文件");
    InvalidateRgn(hWnd, NULL, TRUE);    UpdateWindow(hWnd);
    ReadImage(ImgDlgFileName, OrgImage, IMAGEWIDTH, IMAGEHEIGHT);
    ShowImage(OrgImage, IMAGEWIDTH, IMAGEHEIGHT, XPOS, YPOS);
    break;
case IDM_SHOWBMPIMAGE:
    OpenImageFileDlg("打开图像文件");
    InvalidateRgn(hWnd, NULL, TRUE);    UpdateWindow(hWnd);
    ReadBmpImage(ImgDlgFileName, OrgImage);
    ShowBmpImage(OrgImage, IMAGEWIDTH, IMAGEHEIGHT, XPOS, YPOS);
    break;
```

### 2.2 动画

本模块主要实现图片的动画效果，即图片动态显示，包括淡出、淡入、左右开门、左右关门、上下开门、上下关门、从左向右百叶窗、从右向左百叶窗、从上向下百叶窗、从下向上百叶窗。

其中各个功能对于两种图片格式需要分别处理：raw 文件直接处理灰度值，bmp 文件需要将原图数据以 RGB 三个通道分开并分别进行处理。



```
case IDM_GRAYTOLOW:
    if (IMAGEHEIGHT == IMAGEWIDTH)
        ShowGrayToLow(OrgImage, IMAGEWIDTH, IMAGEHEIGHT, XPOS, YPOS);
    else
        ShowBMPToLow(OrgImage, IMAGEWIDTH, IMAGEHEIGHT, XPOS, YPOS);
    break;
case IDM_GRAYTOHIGH:
    if (IMAGEHEIGHT == IMAGEWIDTH)
        ShowGrayToHigh(OrgImage, IMAGEWIDTH, IMAGEHEIGHT, XPOS, YPOS);
    else
        ShowBMPToHigh(OrgImage, IMAGEWIDTH, IMAGEHEIGHT, XPOS, YPOS);
    break;
case IDM_LEFTRIGHTOPEN:
    ShowLeftRightOpen(OrgImage, IMAGEWIDTH, IMAGEHEIGHT, XPOS, YPOS);
    break;
case IDM_LEFTRIGHTCLOSE:
    ShowLeftRightClose(OrgImage, IMAGEWIDTH, IMAGEHEIGHT, XPOS, YPOS);
    break;
case IDM_TOPBOTTOMOPEN:
    ShowTopBottomOpen(OrgImage, IMAGEWIDTH, IMAGEHEIGHT, XPOS, YPOS);
    break;
case IDM_TOPBOTTOMCLOSE:
    ShowTopBottomClose(OrgImage, IMAGEWIDTH, IMAGEHEIGHT, XPOS, YPOS);
    break;
case IDM_TBSHUTTER:
    ShowTopBottomShutter(OrgImage, IMAGEWIDTH, IMAGEHEIGHT, XPOS, YPOS);
    break;
case IDM_BTSHUTTER:
    ShowBottomTopShutter(OrgImage, IMAGEWIDTH, IMAGEHEIGHT, XPOS, YPOS);
    break;
case IDM_LRSHUTTER:
    ShowLeftRightShutter(OrgImage, IMAGEWIDTH, IMAGEHEIGHT, XPOS, YPOS);
    break;
case IDM_RLSHUTTER:
    ShowRightLeftShutter(OrgImage, IMAGEWIDTH, IMAGEHEIGHT, XPOS, YPOS);
    break;
```

其中以左右开门为例。对于两种不同的图片格式需要分开进行处理，包括数据存储在数组中的位置、数据的处理方式、数据的处理顺序等。

```
void ShowLeftRightOpen(char* Image, int wImage, int hImage, int xPos, int yPos)
{
    int i;
    RECT ShowRect;
    InvalidateRgn(hWind, NULL, TRUE); UpdateWindow(hWind); // 清空窗口
    if (wImage == hImage)
    {
        ShowRect.top = 0;
        ShowRect.bottom = hImage;
        for (i = 0; i < wImage / 2; i++) {
            ShowRect.left = wImage / 2 - i;
            ShowRect.right = ShowRect.left + 1; //显示一列
            ShowImageRect(Image, wImage, hImage, xPos, yPos, ShowRect); //左开门
            ShowRect.left = wImage / 2 + i;
            ShowRect.right = ShowRect.left + 1; //显示一列
            ShowBMPImageRect(Image, wImage, hImage, xPos, yPos, ShowRect); //右开门
            Sleep(1);
        }
    }
    else
    {
        ShowRect.top = 0;
        ShowRect.bottom = hImage;
        for (i = 0; i < wImage / 2; i++) {
            ShowRect.left = wImage / 2 - i;
            ShowRect.right = ShowRect.left + 1; //显示一列
            ShowBMPImageRect(Image, wImage, hImage, xPos, yPos, ShowRect); //左开门
            ShowRect.left = wImage / 2 + i;
            ShowRect.right = ShowRect.left + 1; //显示一列
            ShowBMPImageRect(Image, wImage, hImage, xPos, yPos, ShowRect); //右开门
            Sleep(1);
        }
    }
}
```

对于开门和百叶窗效果，当中都包含了将图片分割为小矩形，再分别显示小矩形图片的功能。而 bmp 图片由于数据存储顺序特殊，对于水平和垂直分割还需要进行分别处理。因此对于这两种图片的矩形部分显示功能需要三种实现。

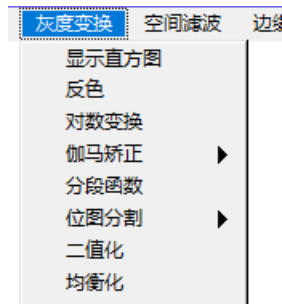
```
void ShowImageRect(char* Image, int wImage, int hImage, int xPos, int yPos, RECT ShowRect)
{
    int i, j;
    int r, g, b;
    for (i = ShowRect.top; i < ShowRect.bottom; i++) {
        for (j = ShowRect.left; j < ShowRect.right; j++) {
            r = g = b = (BYTE)Image[i * wImage + j];
            SetPixel(hWinDC, j + yPos, i + xPos, RGB(r, g, b));
        }
    }
}

void ShowBMPImageRect(char* Image, int wImage, int hImage, int xPos, int yPos, RECT ShowRect)
{
    int i, j;
    int r, g, b;
    char* tmp = Image + wImage * 3 / 2 + (ShowRect.left - (wImage / 2)) * 3 - 3;
    for (i = ShowRect.top; i < ShowRect.bottom; i++) {
        for (j = ShowRect.left; j < ShowRect.right; j++) {
            b = *tmp++;
            g = *tmp++;
            r = *tmp++;
            SetPixel(hWinDC, j + yPos, xPos + hImage - i, RGB(r, g, b));
        }
        tmp += (wImage * 3 - 3);
    }
}

void ShowBMPImageRect1(char* Image, int wImage, int hImage, int xPos, int yPos, RECT ShowRect)
{
    int i, j;
    int r, g, b;
    char* tmp = Image + (hImage / 2 - (ShowRect.top - (hImage / 2))) * wImage * 3;
    for (j = ShowRect.top; j < ShowRect.bottom; j++) {
        for (i = ShowRect.left; i < ShowRect.right; i++) {
            b = *tmp++;
            g = *tmp++;
            r = *tmp++;
            SetPixel(hWinDC, i + yPos, xPos + j, RGB(r, g, b));
        }
    }
}
```

## 2.3 灰度变换

本模块中实现了显示直方图（raw 图片显示灰度直方图、bmp 图片显示三通道颜色直方图）、对数变换、伽马矫正、分段函数、位图分割、反色、二值化、均衡化效果。



以显示直方图为例，同样对于两种图片格式需要分别处理。

```
case IDM_HISTOGRAMSHOW: //直方图
{
    if (IMAGEHEIGHT == IMAGEWIDTH)
    {
        ImageHistogramShowProcessing(OrigImage, NewImage, IMAGEWIDTH, IMAGEHEIGHT);
        ShowImage(NewImage, IMAGEWIDTH, IMAGEHEIGHT, XPOS + 300, YPOS); break;
    }
    else
    {
        rImage = new char[IMAGEHEIGHT * IMAGEWIDTH];
        gImage = new char[IMAGEHEIGHT * IMAGEWIDTH];
        bImage = new char[IMAGEHEIGHT * IMAGEWIDTH];
        NewrImage = new char[256 * 256];
        NewgImage = new char[256 * 256];
        NewbImage = new char[256 * 256];
        DivideEMP(rImage, gImage, bImage, OrigImage, IMAGEWIDTH, IMAGEHEIGHT);
        ImageHistogramShowProcessing(rImage, NewrImage, IMAGEHEIGHT, IMAGEWIDTH);
        ShowImage(NewrImage, 255, 255, XPOS + IMAGEHEIGHT + 50, YPOS);
        ImageHistogramShowProcessing(gImage, NewgImage, IMAGEHEIGHT, IMAGEWIDTH);
        ShowImage(NewgImage, 255, 255, XPOS + IMAGEHEIGHT + 50, YPOS + 300);
        ImageHistogramShowProcessing(bImage, NewbImage, IMAGEHEIGHT, IMAGEWIDTH);
        ShowImage(NewbImage, 255, 255, XPOS + IMAGEHEIGHT + 50, YPOS + 300 * 2);
        break;
    }
}
```

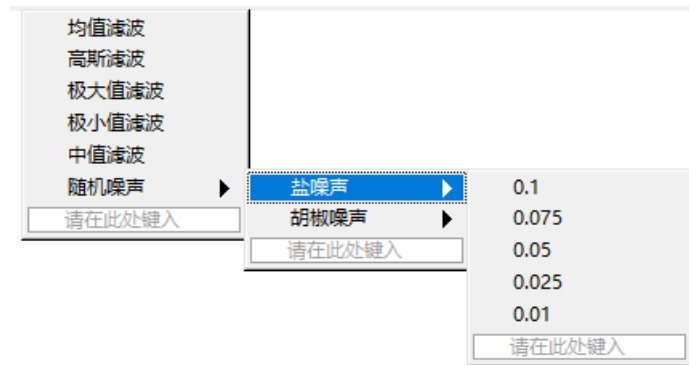
其中，对于 bmp 图片提供了分离通道和合并通道两个函数以便于对图片进行处理。

```
void DivideEMP(char* rImage, char* gImage, char* bImage, char* oImage, int wImage, int hImage)
{
    char* tmp = oImage;
    BYTE r, g, b;
    for (int i = 0; i < hImage; i++) {
        for (int j = 0; j < wImage; j++) {
            b = *tmp++;
            *bImage++ = b;
            g = *tmp++;
            *gImage++ = g;
            r = *tmp++;
            *rImage++ = r;
        }
    }
}

void UnionEMP(char* rImage, char* gImage, char* bImage, char* nImage, int wImage, int hImage)
{
    for (int i = 0; i < hImage; i++) {
        for (int j = 0; j < wImage; j++) {
            *nImage++ = *bImage++;
            *nImage++ = *gImage++;
            *nImage++ = *rImage++;
        }
    }
}
```

## 2.4 空间滤波

本模块分为两大部分：随机噪声生成器和滤波器。噪声生成器中提供了两种不同噪声、每种噪声都有五个噪声比例的选项；滤波器中包含：均值滤波、高斯滤波、极大值滤波、极小值滤波、中值滤波五种滤波操作。



以均值滤波器为例。

```
case IDM_AVERAGEFILTER: //平均滤波器
{
    if (IMAGEHEIGHT == IMAGEWIDTH)
    {
        AverageMaskProcessing(OrgImage, NewImage, IMAGEWIDTH, IMAGEHEIGHT);
        ShowImage(NewImage, IMAGEWIDTH, IMAGEHEIGHT, XPOS, YPOS + 300); break;
    }
    else
    {
        rImage = new char[IMAGEHEIGHT * IMAGEWIDTH];
        gImage = new char[IMAGEHEIGHT * IMAGEWIDTH];
        bImage = new char[IMAGEHEIGHT * IMAGEWIDTH];
        NewrImage = new char[IMAGEHEIGHT * IMAGEWIDTH];
        NewgImage = new char[IMAGEHEIGHT * IMAGEWIDTH];
        NewbImage = new char[IMAGEHEIGHT * IMAGEWIDTH];
        DivideBMP(rImage, gImage, bImage, OrgImage, IMAGEWIDTH, IMAGEHEIGHT);
        AverageMaskProcessing(rImage, NewrImage, IMAGEWIDTH, IMAGEHEIGHT);
        AverageMaskProcessing(gImage, NewgImage, IMAGEWIDTH, IMAGEHEIGHT);
        AverageMaskProcessing(bImage, NewbImage, IMAGEWIDTH, IMAGEHEIGHT);
        UnionBMP(NewrImage, NewgImage, NewbImage, NewImage, IMAGEWIDTH, IMAGEHEIGHT);
        ShowBmpImage(NewImage, IMAGEWIDTH, IMAGEHEIGHT, XPOS, YPOS + IMAGEWIDTH + 300); break;
    }
}
```

## 2.5 边缘检测

本模块中提供了四种边缘检测算法：Laplace、Prewitt、Sobel 和 Roberts。其中 Prewitt 和 Sobel 还提供了两种不同的处理方式。



以 Laplace 为例。

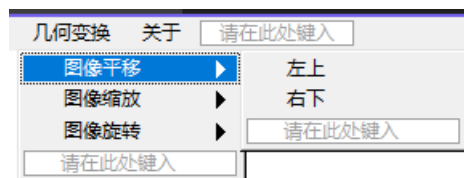
```

case IDM_LAPLACE: //Laplace边缘检测滤波
{
    if (IMAGEHEIGHT == IMAGEWIDTH)
    {
        LaplaceEdgeProcessing(OrgImage, NewImage, IMAGEWIDTH, IMAGEHEIGHT);
        ShowImage(NewImage, IMAGEWIDTH, IMAGEHEIGHT, XPOS, YPOS + 300); break;
    }
    else
    {
        rImage = new char[IMAGEHEIGHT * IMAGEWIDTH];
        gImage = new char[IMAGEHEIGHT * IMAGEWIDTH];
        bImage = new char[IMAGEHEIGHT * IMAGEWIDTH];
        NewrImage = new char[IMAGEHEIGHT * IMAGEWIDTH];
        NewgImage = new char[IMAGEHEIGHT * IMAGEWIDTH];
        NewbImage = new char[IMAGEHEIGHT * IMAGEWIDTH];
        DivideBMP(rImage, gImage, bImage, OrgImage, IMAGEWIDTH, IMAGEHEIGHT);
        LaplaceEdgeProcessing(rImage, NewrImage, IMAGEWIDTH, IMAGEHEIGHT);
        LaplaceEdgeProcessing(gImage, NewgImage, IMAGEWIDTH, IMAGEHEIGHT);
        LaplaceEdgeProcessing(bImage, NewbImage, IMAGEWIDTH, IMAGEHEIGHT);
        UnionBMP(NewrImage, NewgImage, NewbImage, NewImage, IMAGEWIDTH, IMAGEHEIGHT);
        ShowBmpImage(NewImage, IMAGEWIDTH, IMAGEHEIGHT, XPOS, YPOS + IMAGEWIDTH + 300); break;
    }
}

```

## 2.6 几何变换

本模块提供了三种图像的几何移动操作：平移、缩放和旋转。每种操作都有至少两种选项。



以图像旋转为例。由于两种图片数据在数组中存储方式不同，对于同样的旋转操作，处理函数对于不同的图片需要的参数不同。

```

case IDM_ROTATION: //图像旋转
{
    if (IMAGEWIDTH == IMAGEHEIGHT)
    {
        ImageRotation(OrgImage, NewImage, IMAGEWIDTH, IMAGEHEIGHT, 45.0);
        ShowImage(NewImage, IMAGEWIDTH, IMAGEHEIGHT, XPOS, YPOS + 300); break;
    }
    else
    {
        rImage = new char[IMAGEHEIGHT * IMAGEWIDTH];
        gImage = new char[IMAGEHEIGHT * IMAGEWIDTH];
        bImage = new char[IMAGEHEIGHT * IMAGEWIDTH];
        NewrImage = new char[IMAGEHEIGHT * IMAGEWIDTH];
        NewgImage = new char[IMAGEHEIGHT * IMAGEWIDTH];
        NewbImage = new char[IMAGEHEIGHT * IMAGEWIDTH];
        DivideBMP(rImage, gImage, bImage, OrgImage, IMAGEWIDTH, IMAGEHEIGHT);
        ImageRotation(rImage, NewrImage, IMAGEWIDTH, IMAGEHEIGHT, -45.0);
        ImageRotation(gImage, NewgImage, IMAGEWIDTH, IMAGEHEIGHT, -45.0);
        ImageRotation(bImage, NewbImage, IMAGEWIDTH, IMAGEHEIGHT, -45.0);
        UnionBMP(NewrImage, NewgImage, NewbImage, NewImage, IMAGEWIDTH, IMAGEHEIGHT);
        ShowBmpImage(NewImage, IMAGEWIDTH, IMAGEHEIGHT, XPOS, YPOS + IMAGEWIDTH + 300); break;
    }
}

```

## 2.7 关于

本模块显示了版本、作者等信息。

## 3 项目特色功能模块（需贴合老师上课所讲的内容进行延伸）

### 3.1 bmp 图片灰度化

本功能附加在前文灰度变换模块中。本功能实现了将 24 位 bmp 格式彩色图片转换成灰度图的功能。

```

void Change2Grey(char* oImage, char* nImage, int wImage, int hImage)
{
    int i, j;
    char* tmp = oImage;
    int r, g, b;
    for (i = hImage - 1; i >= 0; i--) {
        for (j = 0; j < wImage; j++) {
            b = *tmp++;
            g = *tmp++;
            r = *tmp++;
            nImage[i * wImage + j] = (BYTE)max(r, g, b);
        }
    }
}

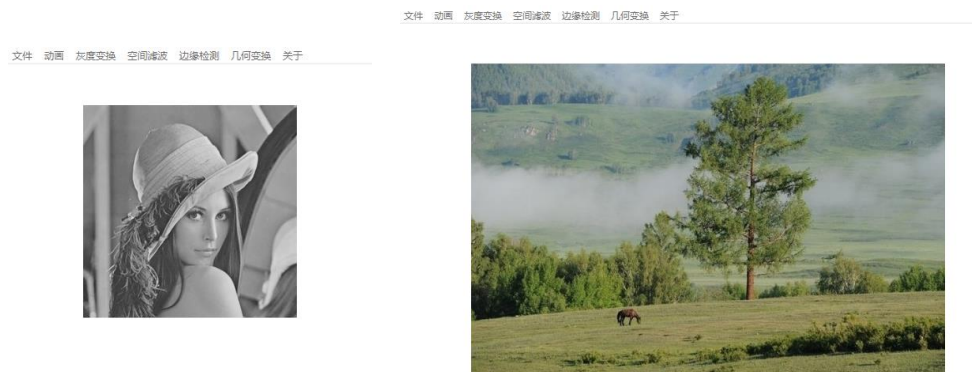
```

## 4 项目效果图

### 4.1 程序界面



### 4.2 图片显示



Raw 图片

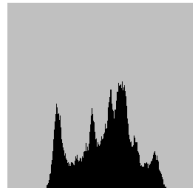
Bmp 图片



## 4.2 部分效果展示



动画-淡出



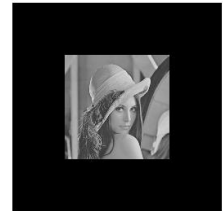
灰度变换  
-显示直方图



空间滤波  
-中值滤波

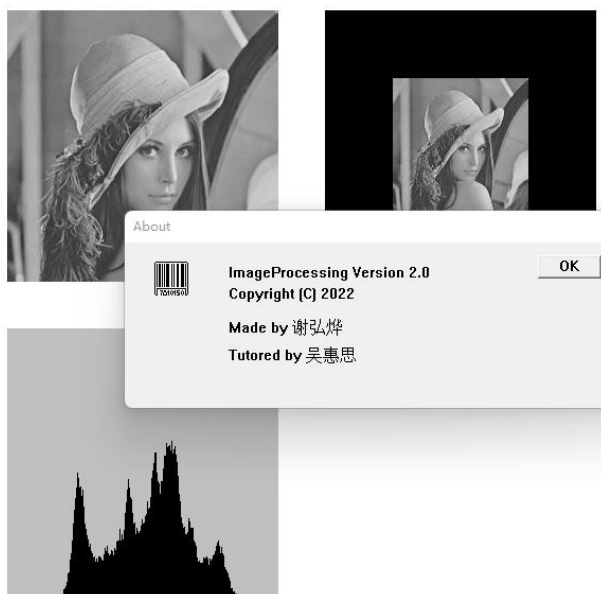


边缘检测-Laplace



几何变换  
-等比缩小

## 4.3 关于



## 5 项目总结与问题分析

本次大作业项目有难度有挑战，虽然效果还是十分简陋，但还是赶在时限前完成。

本项目中各项功能大部分已经在之前的实验中实现，然而当时仅着重在处理 raw 格式图片，没有留意 bmp 图片。本次项目过程中在将之前实现的功能扩展到 bmp 图片上投入较大。

尽管十分努力的完成，本次项目的最终效果还是没能让我自己满意。图形界面不够美观、功能选项不够直观等等。虽然在这一部分投入的时间和精力是在整个项目中最多的，仍然没能有效地解决这一部分问题。这当中最重要的两个原因，一是项目时间较短，没有充足的时间研究 mfc 项目的开发；二是自己的能力不足，不能在有限的时间内完成规定的任务。

## 6 项目心得

这次项目的大部分时间花费在功能拓展和研究 mfc 上。也正是因次，对于彩色图像的数据存储、图像处理有了更深的认识，对于对话框的创建、编辑框的应用以及编辑框中输入内容的处理等 mfc 应用的开发也有了初步的了解。希望这一次项目的经验能够在以后的工作学习生活中给我给予帮助。