《图像处理设计》
课程设计

题目：简单图像处理系统

数字媒体 学 院 数字媒体技术 专 业

学　　号 1030515409

班　　级 数媒 1504

学生姓名 金卓群

指导教师 狄　岚

二〇一八年六月

目录

第一章　绪论

1.1 图像处理设计的目的

21 世纪是一个充满信息的时代，图像作为人类感知世界的视觉基础，是人类获取信息、表达信息和传递信息的重要手段。数字图像处理，即用计算机对图像进行处理，其发展历史并不长。数字图像处理技术源于 20 世纪 20 年代，当时通过海底电缆从英国伦敦到美国纽约传输了一幅照片，采用了数字压缩技术。首先数字图像处理技术可以帮助人们更客观、准确地认识世界，人的视觉系统可以帮助人类从外界获取 3/4 以上的信息，而图像、图形又是所有视觉信息的载体，尽管人眼的鉴别力很高，可以识别上千种颜色，但很多情况下，图像对于人眼来说是模糊的甚至是不可见的，通过图象增强技术，可以使模糊甚至不可见的图像变得清晰明亮。

在计算机中，按照颜色和灰度的多少可以将图像分为二值图像、灰度图像、索引图像和真彩色 RGB 图像四种基本类型。大多数图像处理软件都支持这四种类型的图像。

常用方法包括：图像变换，图像编码压缩，图像增强和复原，图像分割，图像描述，图像分类等。

1.2 图像处理设计的任务和要求

1.2.1、 对图像文件（bmp、jpg、等）进行打开、保存、退出等功能操作；

1.2.2、 几何变换

（1）图像的平移、缩放、旋转、镜像、转置等；

（2）特殊的几何变换，如：百叶窗效果、分块显示效果等。

1.2.3、图像增强

（1）空域中的各种增强方法

a) 空域中的点运算：对比度的增加、减少，亮度的增加、减少，图像反色
b) 图像的灰度拉伸、直方图的均衡化
c) 空间域平滑算法（如均值滤波、中值滤波等）

（2）频域的各种增强方法：低通滤波、高通滤波等。

（3）色彩增强：

伪彩色增强:将一幅灰度图像转换成一幅真彩色图像；
假彩色增强: 将三幅灰度图像合成一幅真彩色图像；
真彩色增强：

a) 将一副真彩色图像进行 RGB 分离；
b) 将某一分离后的单色图像增加、减少；
c) 产生新的真彩色图像。

1.2.4、图像分割：

（1）边缘检测

分别用 Roberts 算子、Sobel 算子、Laplacian 算子等作边缘检测；

（2）阈值分割

a) 交互式阈值分割；
b) 迭代阈值分割。

1.2.5、二值图像处理：膨胀、腐蚀、开运算与比运算。

第二章　总体设计内容

2.1 系统基本功能

简易图像处理系统应具有的基本功能包括打开一幅或多幅图像（用户自主选择），保存当前变换的图像（可以选择保存的位置和名称），以及退出系统。

2.2 图像的算术运算

图像运算指以图像为单位进行的揉作（该操作对图像中的所有像素同样进行），运算的结果是 一幅其灰度分布与原来参与运算图像灰度分布不同的新图像。具体的运算主要包括算术和逻辑运算，它们通过改变像素的值来得到图像增强的效果。 [1]

算术和逻辑运算中每次只涉及一个空间像素的位置，所以可以“原地"完成，即在(x，y)位置做一个算术运算或逻辑运算的结果可以存在其中 一个图像的相应位置，因为那个位置在其后的运算中不会再使用。换句话说，设对两幅图像 f(x，y) 和 h(x，y)的算术或逻辑运算的结果是 g(x，y),则可直接将 g(x，y)覆盖 f(x，y)或 h(x，y)，即从原存放 输入图像的空间直接得到输出图像。

本系统中图像的算术运算包括多幅图像的叠加，图像与二值图像之间的点乘，两幅图像之间的相减。

2.3 图像变换

图像基本变换应包括平移、旋转、缩放、镜像，并能够实现傅里叶变换，灰度拉伸及特殊的百叶窗效果处理。其中：

平移：在 x,y 轴上进行任意长度的平移

旋转：围绕任意点做任意角度的旋转

缩放：在 x,y 轴上做任意比例的缩放

镜像：包括水平镜像，垂直镜像

2.4 图像增强

图像增强的目的是为了提高图像的质量，如去除噪声，提高图像的清晰度等。图像增强不考虑图像降质的原因，突出图像中所感兴趣的部分。如强化图像高频分量，可使图像中物体轮廓清晰，细节明显；如强化低频分量可减少图像中噪声影响。图像复原要求对图像降质的原因有一定的了解，一般讲应根据降质过程建立“降质模型”，再采用某种滤波方法，恢复或重建原来的图像。

图像增强的基本操作应包括有空域滤波，频域滤波，彩色图像增强，和滤波器的实现。

其中彩色图像增强，包括针对灰度图像的伪彩色增强，针对真彩色图像的假彩色图像增强和真彩色图像增强。而滤波器中包括 Sobel 算子及自定义算子。

2.5 图像编码

图像编码压缩技术可减少描述图像的数据量（即比特数），以便节省图

像传输、处理时间和减少所占用的存储器容量。压缩可以在不失真的前提下获得，也可以在允许的失真条件下进行。编码是压缩技术中最重要的方法，它在图像处理技术中是发展最早且比较成熟的技术。

本系统中的图像编码中主要是行程编码。针对灰度图和真彩色图有不同的编码解码方式，并能将编码数据存储在 txt 文本中。

2.6 图像分割

图像分割是数字图像处理中的关键技术之一。图像分割是将图像中有意义的特征部分提取出来，其有意义的特征有图像中的边缘、区域等，这是进一步进行图像识别、分析和理解的基础。虽然目前已研究出不少边缘提取、区域分割的方法，但还没有一种普遍适用于各种图像的有效方法。因此，对图像分割的研究还在不断深入之中，是目前图像处理中研究的热点之一。

本系统中的图像分割包活边缘检测和阈值分割。边缘检测中包括 roberts 算子，sobel 算子，Laplacian 算子；阈值分割中包括交互式阈值分割和迭代式阈值分割。

2.7 形态学处理

形态学一词通常指生物学的一个分支，它用于处理动物和植物的形状和结构。在数学形态学的语境中也使用该词来作为提取图像分量的一种工具，这些分量在表示和描述区域形状（如边界，骨骼和凸壳）时是很有用的。此外，我们还很关注用于预处理和后处理的形态学技术，如形态学滤波、细化和裁剪。

数学形态学的基本运算有 4 个：腐蚀、膨胀、开启和闭合。数学形态学方法利用一个称作结构元素的”探针”收集图像的信息，当探针在图像中不断移动时，便可考察图像各个部分之间的相互关系，从而了解图像的结构特征。在连续空间中，灰度图像的腐蚀、膨胀、开运算和闭运算分别表述如下：

腐蚀：“收缩”或“细化”二值图像中的对象

膨胀：是在二值图像中“加长”或“变粗”的操作

开运算：A 被 B 的形态学开运算是 A 被 B 腐蚀后再用 B 来膨胀腐蚀结果

闭运算：A 被 B 形态学闭运算是先膨胀后腐蚀的结果。

第三章　方案设计

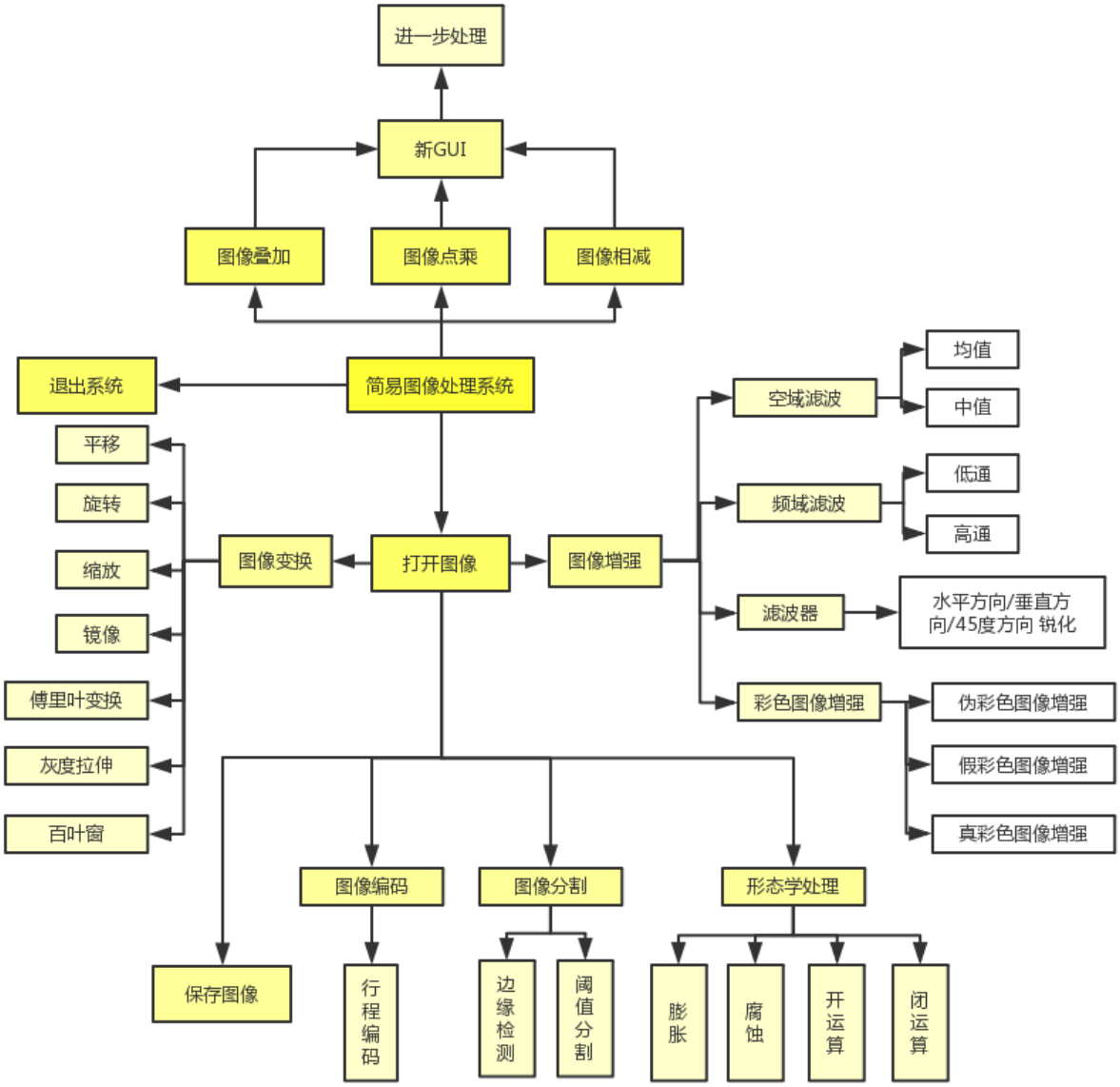
3.1 系统介绍

利用 Matlab GUI 组件,设计实现了能够与用户交互、图像处理过程可视化的图像处理平台; 充分利用了 Matlab GUI 特性,采用底层代码设计,实现图像处理平台的交互式、可视化,较好的实现了图像处理算法一体化集成的功能。

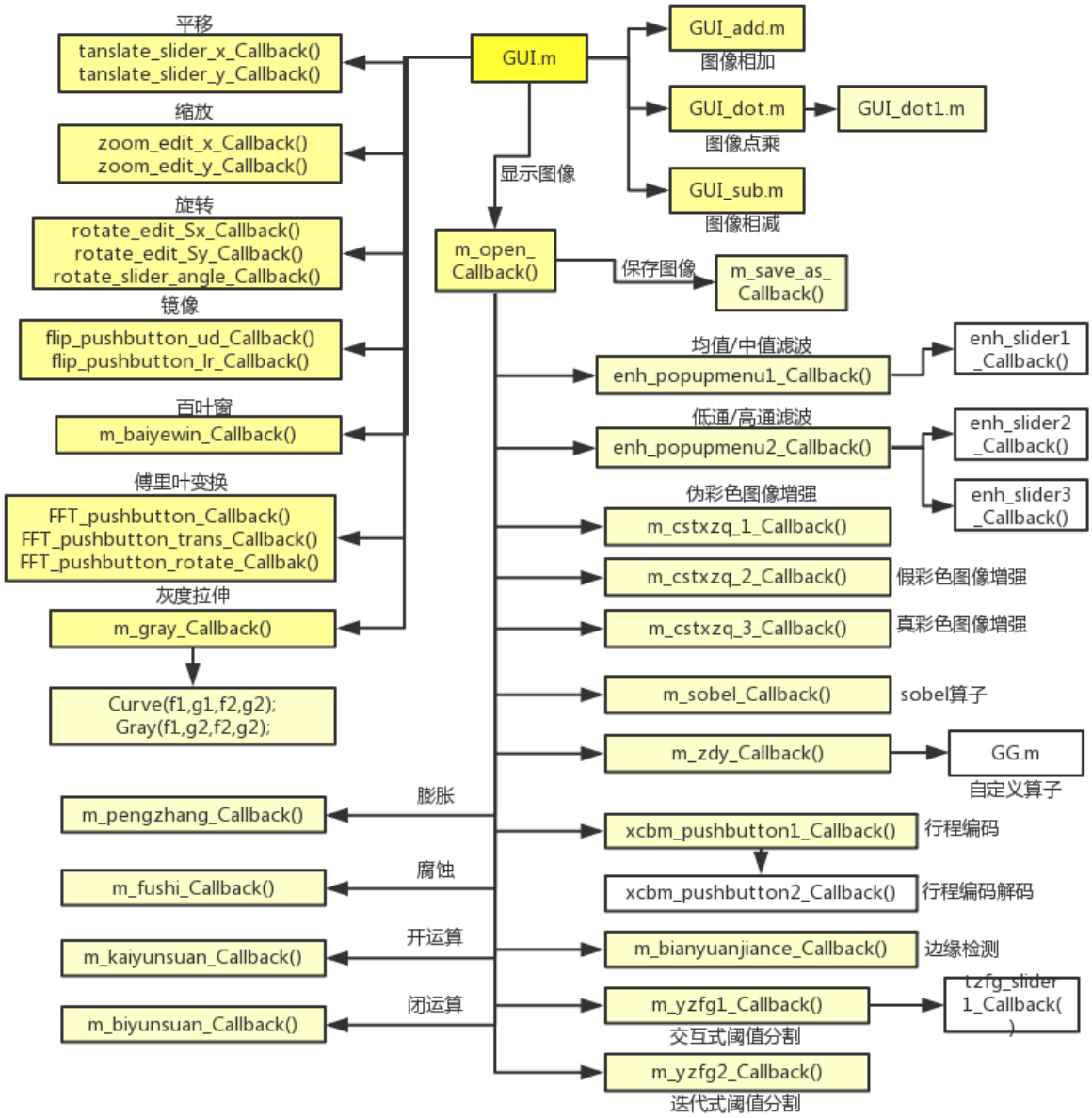
将多种图像处理算法融合到一个系统,不仅丰富了图像处理平台的功能,而且还可以为用户二次开发提供便利.使用算法集成的方法,设计实现了包含图像变换、图像编码、图像增强等已有图像处理技术等算法的一体化图像处理平台。

开发环境：Win10　　Matlab R2016b

3.2 功能模块框架图


进一步处理
新GUI
图像叠加
图像点乘
图像相减
退出系统
简易图像处理系统
平移
旋转
缩放
镜像
傅里叶变换
灰度拉伸
百叶窗
图像变换
打开图像
图像增强
空域滤波
均值
中值
频域滤波
低通
高通
滤波器
水平方向/垂直方向/45度方向 锐化
彩色图像增强
伪彩色图像增强
假彩色图像增强
真彩色图像增强
保存图像
图像编码
行程编码
图像分割
边缘检测
阈值分割
形态学处理
膨胀
腐蚀
开运算
闭运算

3.3 系统流程图



第四章　系统实现

4.1 图像的算术运算

a) **叠加：将两幅不同大小的图像缩放为相同大小，再将其所有像素值减半相加，否则会超出 255 的范围。见[图 4-1-1]**

```
[m,n,~]=size(B);
A = imresize(A,[m n]);
C=0.5*A+0.5*B;
```

b) **点乘：先构造一个二值图像，二值图像的大小要与原图像一致，再进行点乘。见[图 4-1-2]**

```
[m,n,o]=size(A);
```

```
A1=zeros(m,n,o);
A1(0.2*m:0.8*m,0.2*n:0.8*n,:)=1;
A=im2double(A);A1=im2double(A1);
A2=A1.*A;
```

c) 相减:先将两幅图像缩放成相同大小，再利用 imabsdiff 函数进行相减，得出图像矩阵。见[图 4-1-3]

```
[m,n,~]=size(B);
A = imresize(A,[m n]);
C1=imabsdiff(A,B);   %相减的绝对值函数
C2=imabsdiff(B,A);
```

[图 4-1-1]

[图 4-1-2]

[图 4-1-3]

4.2 基本图像变换

a) 平移：利用 GUI 控件中的 Slider 获取平移量，利用函数进行平移，超出部分裁掉，移出部分用黑色填充。见[图 4-2-1]

```
se = translate(strel(1), [tanslate_slider_y tanslate_slider_x]);
trans_pic= imdilate(trans_pic,se);
```

b) 缩放：利用 imresize 函数进行图像缩放，但图像的显示会拉伸到整个坐标系，若按相同比例缩放，只会改变分辨率，不同比例的效果较明显。见[图 4-2-2]

```
trans_pic= imresize(trans_pic, [m*Sy n*Sx], 'nearest');
```

c) 旋转：自定义输入旋转中心，对灰度图像进行处理；用 silider 获取旋转角度，并进行弧度制转换，进入旋转矩阵，与图像矩阵相乘。见[图 4-2-3]

```
alfa = -angle * 3.1415926 / 180.0;
tras = [cos(alfa) -sin(alfa) 0; sin(alfa) cos(alfa) 0; 0 0 1]; %
tx=-rotate_Sy;ty=-rotate_Sx;
disp(tx);disp(ty);
MM=[1 0 0;0 1 0;tx ty 1];
NN=[1 0 0;0 1 0;-tx -ty 1];
tras1=MM*tras*NN;
tras1=tras1';
for i = 1 : R
    for j = 1 : C
        temp = [i; j; 1];
        temp = tras1 * temp;
        x = uint16(temp(1, 1));
        y = uint16(temp(2, 1));

        if (x <= R) && (y <= C) && (x >= 1) && (y >= 1)
            res(i, j) = trans_pic(x, y);
        end
    end
end;
```

[图 4-2-1]

[图 4-2-2]

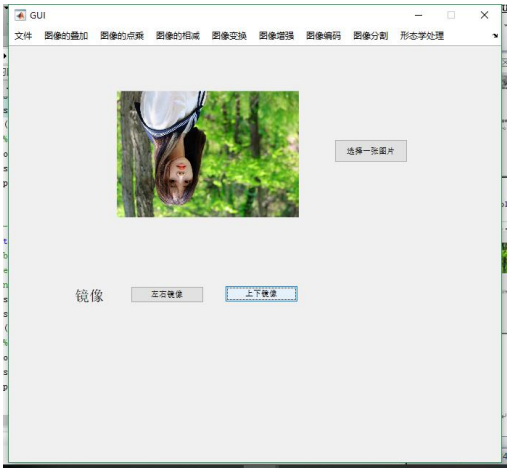
[图 4-2-3]

d) 镜像：利用函数 fliplr 和 flipud 进行水平和垂直镜像。
见[图 4-2-4] [图 4-2-5] [图 4-2-6]

```
trans_pic=fliplr(trans_pic);
trans_pic=flipud(trans_pic);
```

[图 4-2-4]

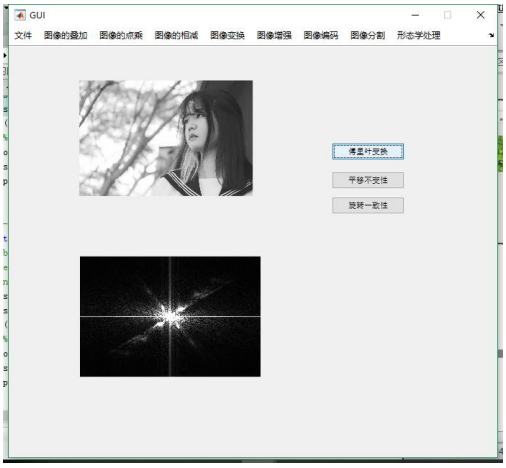
[图 4-2-5]



[图 4-2-6]

e) 傅里叶变换：利用函数 **fftshift**，对图像矩阵进行傅里叶变换。见**[图 4-2-7]**

FFT_B = fftshift(FFT_B);



[图 4-2-7]

f) 灰度拉伸

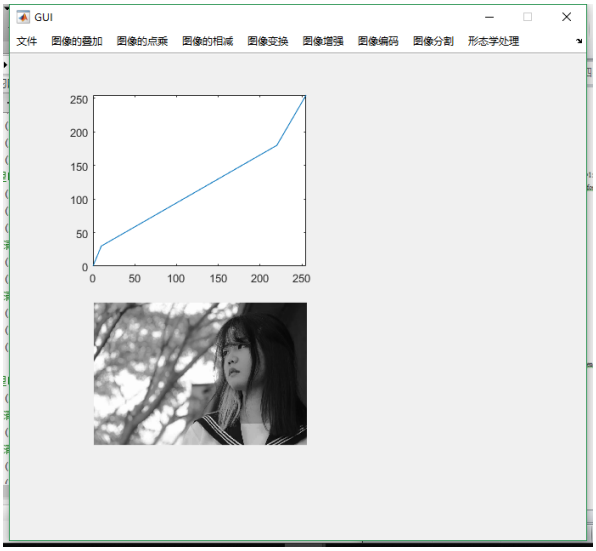
画曲线的部分：见[图 4-2-8]

```
f0=0;g0=0;
f1=x1;g1=y1;
f2=x2;g2=y2;
f3=255;g3=255;
r1=(g1-g0)/(f1-f0);
b1=-r1*f0+g0;
r2=(g2-g1)/(f2-f1);
b2=-r2*f1+g1;
r3=(g3-g2)/(f3-f2);
b3=-r3*f2+g2;
```

对曲线进行分析，对应到图像各个灰度值区域进行改变：见**[图 4-2-8]**

```
for j=1:n
    t=h(i,j);
    g(i,j)=0;
    if((t>=f0)&&(t<=f1))
        g(i,j)=r1*t+b1;
   else
       if((t>=f1)&&(t<=f2))
           g(i,j)=r2*t+b2;
      else
          if((t>=f2)&&(t<=f3))
                g(i,j)=r3*t+b3;
          end
       end
   end
end
```

```
end
```

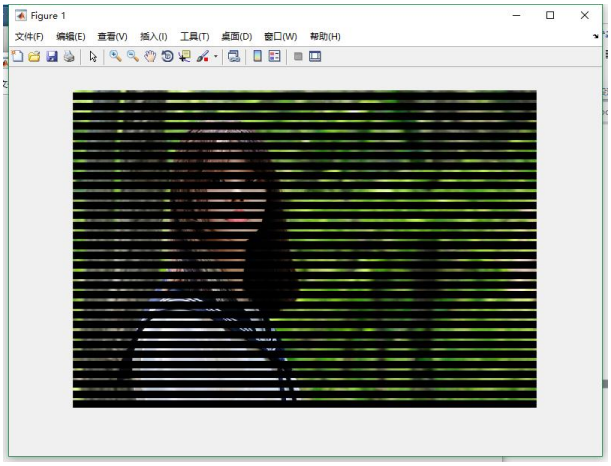


[图 4-2-8]

-

g) 百叶窗：对图像进行分析，先构造一个相同大小的黑色图像，将其按高度分为 20 块，遍历每一行，每一次循环中将原本的黑色像素替换成图像相应位置的像素值，并在每次循环中用 pause 函数暂定一毫秒，即可得到动态的百叶窗效果。见[图 4-2-9]

```
for row_i=1:height
    Map(row_i, :,:)=1-row_i/height;
end
H_shade=20;
figure;imshow(Map);
for j=1:H_shade
    for i=1:H_shade:height
    Map(i+j-1,:,:)=Image(i+j-1,:,:);
    end;
    pause(0.01);
    imshow(uint8(Map));
end;
```



[图 4-2-9]

4.3 图像增强

a) 空域滤波

均值滤波：

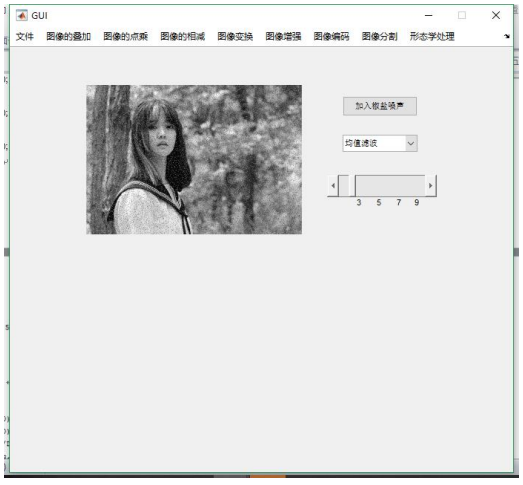
构造一个 3*3 的矩阵，矩阵值均为 1，利用函数 imfilter 进行滤波变换，即用这个矩阵去遍历图像中的每一个 3*3 的像素块，将像素块与均值矩阵相应像素相乘结果的均值，来代替像素块中心像素的值。见[图 4-3-1]

```
h1 = ones(3,3)/9;
B1=imfilter(A,h1);
```

中值滤波：

构造一个 5*5 的像素块，再像素块中对所有像素值重新排序，取其中值来代替中心像素值。用 medfilt2 函数来实现。见[图 4-2-11]

```
B2 = medfilt2(A,[5,5]);
```



[图 4-3-1]

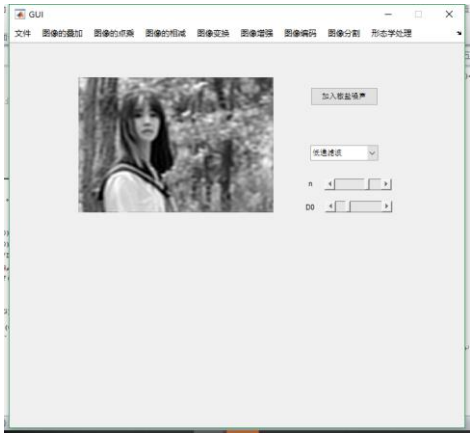
[图 4-3-2]

b) 频域滤波

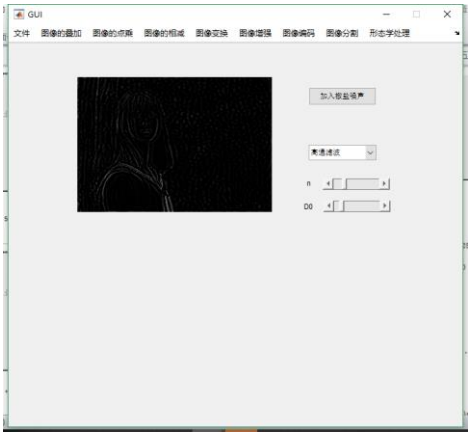
低通滤波和高通滤波：先进行傅里叶变换，再仿照切比雪夫滤波器所提供的公式进行算法运算，遍历每一个像素，低通滤波使低频信号通过，高通滤波使高

频信号通过，再进行傅里叶反变换还原图像。见[图 4-2-12][图 4-2-13]

```
for u=1:M
    for v=1:N
        D=sqrt((u-u0)^2+(v-v0)^2);
        H=1/(1+(D/D0)^(2*n));
        H1=1/(1+(D0/D)^(2*n));
        G(u,v)=H*F(u,v);
        G1(u,v)=H1*F(u,v);
    end;
end;
g=ifft2(ifftshift(G));
g=im2uint8(real(g));
g1=ifft2(ifftshift(G1));
g1=im2uint8(real(g1));
```



[图 4-3-3]



[图 4-3-4]

c) 彩色图像增强

伪彩色图像增强：对一副灰度图进行操作，将灰度值小于 128 的像素颜色改为蓝色，大于 128 的像素颜色改为蓝色。[图 4-3-5]

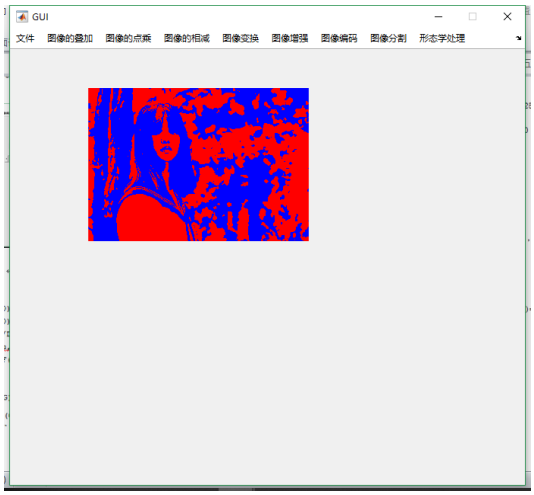
```
for i= 1:M
    for j=1:N
        if X(i,j)<128
            Y(i,j,1:3)=[0 0 255];
        else
            Y(i,j,1:3)=[255 0 0];
        end
    end
end
```

假彩色图像增强：对一副真彩色图像进行操作，将 R、G、B 三个通道交换。见[图 4-3-6]

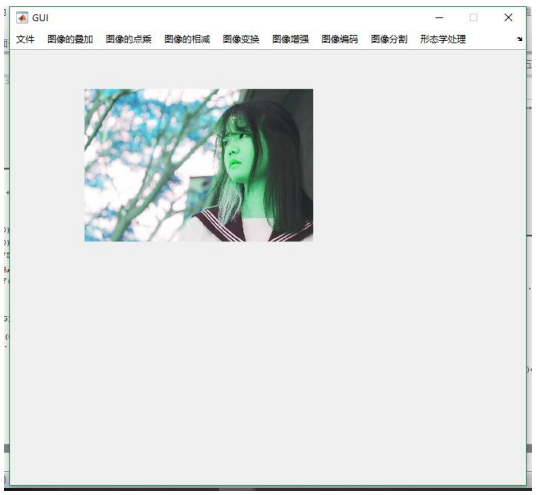
```
X1=R;X2=G;X3=B;
R=X3;G=X1;B=X2;
```

g(:,:,1)=R;g(:,:,2)=G;g(:,:,3)=B;
g=uint8(g);

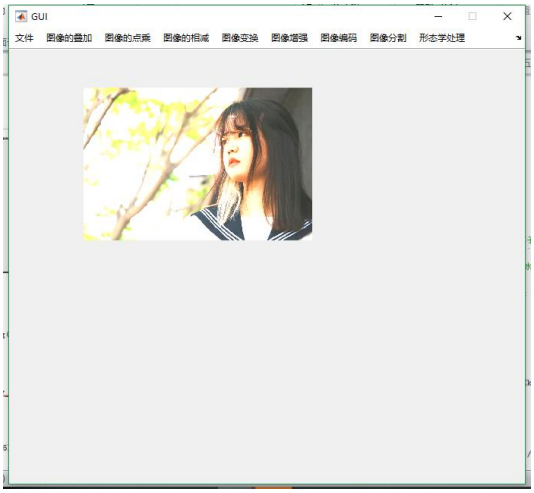
真彩色图像增强：分别增强了 R、G、B 通道的值。见[图 4-3-7]
R=imlincomb(1.5,R);
G=imlincomb(1.5,G);
B=imlincomb(1.5,B);
g(:,:,1)=R;g(:,:,2)=G;g(:,:,3)=B;



[图 4-3-5]



[图 4-3-6]



[图 4-3-7]

d) 滤波器：
利用 Mtalab 的 GUI 组件可以获取用户自定义输入的滤波器的各个值，利用 imfilter 函数进行滤波变换。见[图 4-3-8]
A=imfilter(pic,zdy_h);

[图 4-3-8]

4.4 图像编码

a) 行程编码

对一副灰度图像：转为二值图像后，获得所有图像数据，长度为所有像素和，依此遍历，若具有相同颜色值的像素是相邻的，则只记录第一个像素的值以及这个像素块的长度。并将编码信息输出到了 txt 文件。

对一副真彩色图像：将 RGB 三通道分离处理，分离后的矩阵类似灰度图像进行处理。见[图 4-4-1]

```
I2=im2bw(I,0.5);
I3=I2(:)';
I3len=length(I3);
j=1;n=1;
for z=1:(I3len-1)
    if I3(z)==I3(z+1)
        n=n+1;
    else
        pixel(j)=I3(z);
        numpixel(j)=n;
        j=j+1;
        n=1;
    end
end
pixel(j)=I3(z+1);
numpixel(j)=n;
pixel_len=length(pixel);
CR=I3len/pixel_len;
```

```
压缩比：
    39.0383

原图像数据长度：
        592640

压缩后的图像数据长度：
         15181
```

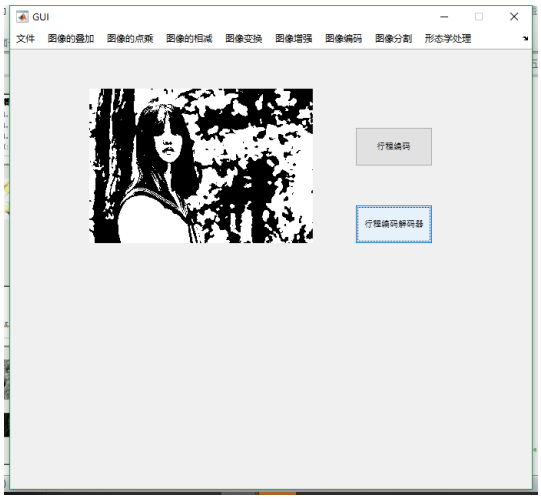
[图 4-4-1]

b) 行程编码解码：

分析编码结果，还原像素值。见[图 4-4-2]

```
 for a=1:pixel_len
        for b=1:numpixel(a)
              I4(n)=pixel(a);
              n=n+1;
        end;
end;
```

```
解码后的图像数据长度：
        592641
```



[图 4-4-2]

4.5 图像分割

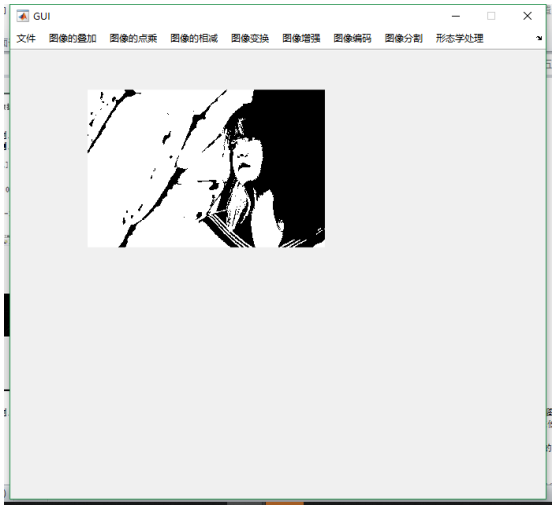
a) 边缘检测：定义了 roberts 算子，sobel 算子，laplacian 算子。见[图 4-5-1]

```
h1=[-1 0;0 1];
h2=[1 2 1; 0 0 0;-1 -2 -1];
h3=[0 -1 0; -1 4 -1; 0 -1 0];
```

[图 4-5-1]

b) 阈值分割：按照灰度级，对像素集合进行一个划分。见[图 4-5-2]

```
t=[];t(1)=128;MAX=500;
for i=1:MAX
    s1=0;s2=0;s3=0;s4=0;
    for k=1:t(i)
        s1=s1+h(k)*k; s2=s2+h(k);
    end
    for k=t(i)+1:256
        s3=s3+h(k)*k;
        s4=s4+h(k);
    end
    t(i+1)=floor((s1/s2+s3/s4)/2);
    if(abs(t(i+1)-t(i))<10^-7)
        break;
    end
end
tt=t(i+1);
B=A;
B(B<tt)=0;
B(B>=tt)=255;
```



[图 4-5-2]

4.6 形态学处理

a) 膨胀：

用 3x3 的结构元素，扫描图像的每一个像素
用结构元素与其覆盖的二值图像做“与”操作
如果都为 1，结果图像的该像素为 1。否则为 0。
结果：使二值图像减小一圈。见[图 4-6-1]

sel=strel('square',3);

B=imdilate(A,sel);

[图 4-6-1]

b) 腐蚀：

用 3x3 的结构元素，扫描图像的每一个像素
用结构元素与其覆盖的二值图像做“与”操作
如果都为 0，结果图像的该像素为 0。否则为 1
结果：使二值图像扩大一圈。见[图 4-6-2]

sel=strel('square',3);

B=imerode(A,sel);

[图 4-6-2]

c)　开运算：先腐蚀后膨胀的过程. 见[图 4-6-3]

```
sel=strel('square',2);
B=imopen(A,sel);
```

[图 4-6-3]

d)　闭运算：先膨胀后腐蚀的过程. 见[图 4-6-4]

```
sel=strel('square',7);
B=imclose(A,sel);
```

[图 4-6-4]

第五章　总结与体会

通过一学期图像处理的学习，以及这段时间的课程设计，算是给图像处理入了个门吧。基本的变换，像是平移旋转缩放镜像、多幅图像之间的处理、傅里叶变换以及空域滤波和频域滤波、图像增强中的各种方法、以及膨胀腐蚀开运算闭运算等，都基本了解应用。在后来学习到数字信号和数字视音频的时候也用到了这方面知识，比如傅里叶变换，图像编码到视频编码，不得不说学院的培养方案还是有一些道理，一环扣一环，存在即合理。

学这门课的时候总觉得，matlab 现在都过时了，图像处理也已经烂大街了，但现在回过头来看，我们只是站在了巨人的肩膀上，相对轻松的学习了高度以下的东西，上限是很容易达到的，上限以上的东西，则需要靠着自己的兴趣，不断学习领悟，才能有所成就。

第六章　参考文献

[1] 兰红.　多阈值优化的交互式医学图像分割方法[J]. 计算机科学. 2013(09)
[2] 郭依正,焦蓬蓬.　Matlab GUI 在低质量指纹图像增强中的应用[J]. 计算机技术与发展. 2013(07)
[3] 孙祥,黄晓鸣.　基于 MATLAB 的集成化图像处理系统[J]. 科学技术与工程. 2007(20)
[4] 赵书兰,主编.MATLAB R2008 数字图像处理与分析实例教程[M]. 化学工业出版社, 2009
[5] 史孝文,杨晓京,傅中裕.　基于 MATLAB 的 GUI 设计伺服驱动系统仿真软件[J]. 机械工程与自动化. 2005(02)
[6] 左飞,万晋森,刘航,著.数字图像处理原理与实践[M]. 电子工业出版社, 2010
[7] 赵书兰,主编.MATLAB R2008 数字图像处理与分析实例教程[M]. 化学工业出版社, 2009