

项目编号 Project 03-02 & 03-05 & 03-06

学生姓名：郝裕玮&张闯

截止日期：9.29

上交日期：9.29

摘要

本次实验共完成了3个项目：直方图均衡，拉普拉斯图像锐化，反锐化掩蔽和高提升滤波处理。并在不使用 MATLAB 内置库函数的情况下成功实现了多种功能，如：直方图绘图，直方图均衡化函数，空间滤波处理等等。

小组分工：

郝裕玮：Project 03-02 & Project 03-06

张闯：Project 03-05

Project 03-02

(1) 技术讨论：

代码具体思路为：读取图像，统计各灰度级的值，绘制原图和原图直方图。之后再根据PPT上的步骤（如下所示）即可写出直方图均衡化函数，并绘制出均衡函数转换图，增强图像和均衡化直方图。

数字图像下的直方图均衡化步骤：

1) 概率： $p_r(r_k) = \frac{n_k}{n}$, $k = 0, 1, 2, \dots, L-1$

2) 累积分布函数：

$$P(r_k) = \sum_{j=0}^k p_r(r_j) = \sum_{j=0}^k \frac{n_j}{n}, \quad k = 0, 1, 2, \dots, L-1$$

3) 变换函数：

$$s_k = T(r_k) = (L-1) \cdot \sum_{j=0}^k \frac{n_j}{n}, \quad k = 0, 1, 2, \dots, L-1$$

4) 将 s_k “四舍五入”转换为标准灰度级别，如果有相同的 $\lceil s_k \rceil$ ，则合并。

代码以及具体思路如下所示（详见代码注释）：

```
%读取原图并画出原图
pic=imread('C:\Users\93508\Desktop\38.jpg'); %读取原图
figure(1);
imshow(pic); %画出原图
title('原图');

%自创的绘制图像直方图的程序
b1=zeros(1,256); %用于存储不同灰度值出现的次数,数组维度为 1*256（一维数组）
%for 循环统计不同灰度值出现的次数
for i=0:255
    b1(i+1)=numel(find(pic(:,')==i));
    %find 函数:返回一个数组 a，数组 a 中的元素均为参数传入的数组中满足 find 条件的元素
    %numel 函数:返回数组的元素数目 n
end
figure(2);
bar(b1); %画出原图的直方图
title('原图的直方图');

%调用自创的直方图均衡化函数画出均衡化后的图像
pic_final=balance(pic); %调用自创的直方图均衡化函数
figure(4);
imshow(pic_final); %画出增强图像
title('增强图像');

%自创的绘制图像直方图的程序
b2=zeros(1,256); %用于存储不同灰度值出现的次数,数组维度为 1*256（一维数组）
%for 循环统计不同灰度值出现的次数
for i=0:255
    b2(i+1)=numel(find(pic_final(:,')==i));
    %find 函数:返回一个数组 a，数组 a 中的元素均为参数传入的数组中满足 find 条件的元素
    %numel 函数:返回数组的元素数目 n
end
figure(5);
bar(b2); %画出增强图像的直方图
title('增强图像的直方图');

%均衡化函数
function pic_final=balance(pic)
```

```

[ row,col]=size(pic);
pic_final=zeros(row,col);
a=zeros(1,256);
b=zeros(1,256);

%for 循环统计不同灰度值出现的次数
for i=0:255
    a(i+1)=numel(find(pic(:,')==i))/(row*col);
end

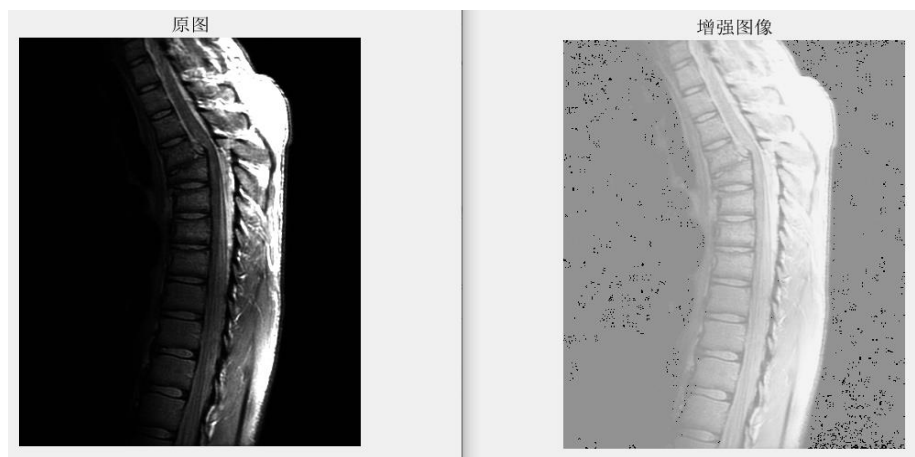
%累积分布函数
b(1)=a(1);
for i=2:256
    b(i)=b(i-1)+a(i);
end
%画出均衡转换函数图
figure(3);
plot(b);
set(gca, 'XTick',0:64:256);

%四舍五入转换为标准灰度级别并合并
for i=1:row
    for j=1:col
        pic_final(i,j)=round(255*b(pic(i,j)+1));
    end
end
pic_final=uint8(pic_final);
end

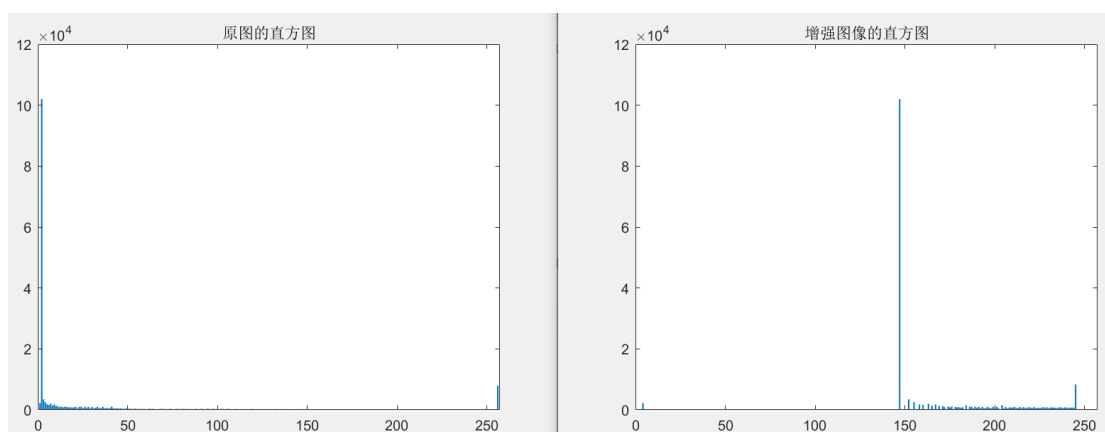
```

(2) 结果讨论:

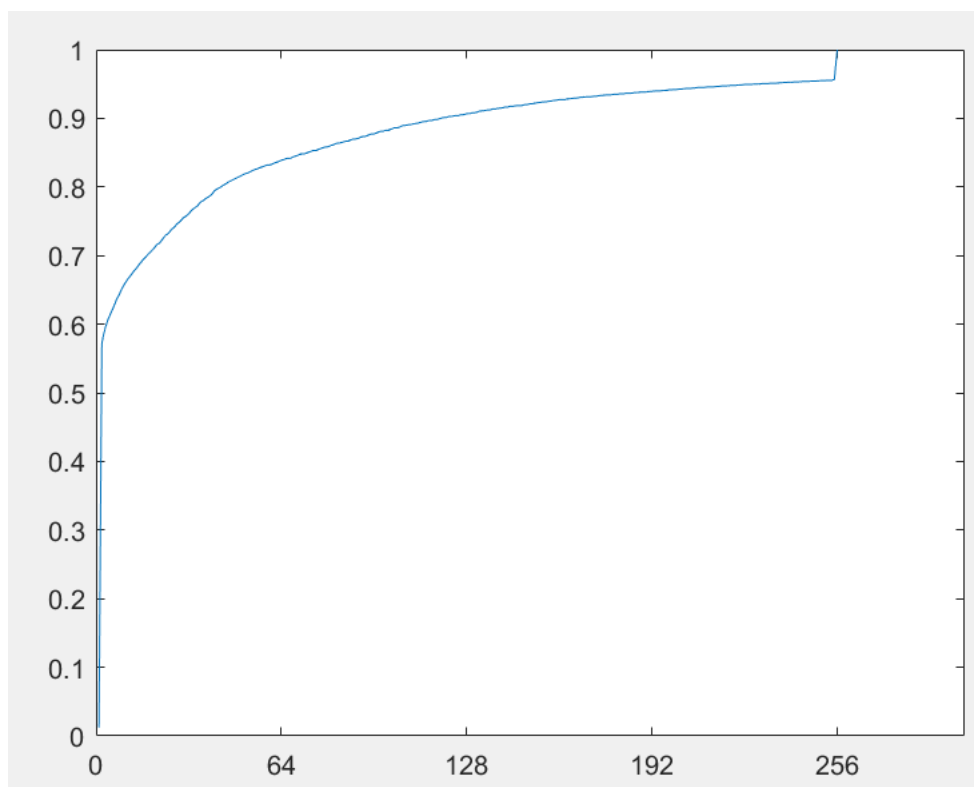
原图和增强图像的对比 (见下页):



原图直方图和增强图像直方图的对比：



直方图均衡转换函数图：



Project 03-05

(1) 技术讨论:

使用二阶微分进行图像锐化——拉普拉斯算子，定义一个基于二阶微分的离散公式，然后构造一个基于该公式的滤波器模板组成的。

使用的滤波器模板为：

0	1	0	1	1	1	0	-1	0	-1	-1	-1	a
1	-4	1	1	-8	1	-1	4	-1	-1	8	-1	b
0	1	0	1	1	1	0	-1	0	-1	-1	-1	c
												d

图 3.37 (a)实现式(3.6-6)所用的滤波器模板；(b)用于实现带有对角项的该公式的扩展的模板；(c)~(d)实践中常用的其他两个拉普拉斯实现

代码以及具体思路如下所示（详见代码注释）：

```
A=imread('pic2.tif'); %读入图片
[m,n]=size(A); %获取图片大小
lap_fil=[-1,-1,-1;-1,8,-1;-1,-1,-1]; %滤波器模板
B=zeros(m+4,n+4); %在矩阵周围补0
for i=1:m
    for j=1:n
        B(i+2,j+2)=A(i,j);
    end
end

res=zeros(m+4,n+4); %进行滤波运算
for i=2:m+3
    for j=2:n+3
        for x=-1:1
            for y=-1:1
                res(i,j)=res(i,j)+B(i+x,j+y)*lap_fil(2+x,2+y);
            end
        end
    end
end

C=zeros(m,n); %将原图像和拉普拉斯图像叠加
for i=1:m
    for j=1:n
        C(i,j)=A(i,j)+res(i+2,j+2);
    end
end
```

```
imshow(C,[0,255]); %生成新的图像
```

(2) 结果讨论:



Project 03-06

(1) 技术讨论:

具体技术和思路实现可详见如下代码 (均已写在代码注释中):

```
%读取原图并画出原图
pic=imread('C:\Users\93508\Desktop\43.jpg'); %读取原图
figure(1);
imshow(pic); %画出原图
title('原图');

%扩展矩阵, 做好卷积准备
mask=1/9*[1 1 1;1 1 1;1 1 1]; %平滑(均值)滤波器掩膜
[row,col]=size(pic);
dx=zeros(2,col+4);
dy=zeros(row,1);
```

```

pic_final=[dx;dy dy pic dy dy;dx]; %这里滤波器模板是 3*3，所以原图矩阵扩展
为[row+2,col+2]
[row_final,col_final]=size(pic_final);
lap=uint8(zeros(row_final,col_final)); %整数只能与同类的整数或双精度标量值
组合使用，不加此行则后续 lap 无法与 pic(uint8 型)相加

%拉普拉斯算子的图像锐化
for i=3:row_final-2
    for j=3:col_final-2
        for s=1:3
            for t=1:3
                lap(i,j)=lap(i,j)+pic_final(i-2+s,j-2+t)*mask(s,t); %与
Project 03-05 的思路一致
            end
        end
    end
end

%原图的非锐化隐蔽和高升压滤波
lap=lap(3:row_final-2,3:col_final-2); %将扩展图片的矩阵裁剪为原图大小
A=1.7;
hb=A*pic+lap; %应用第 2 版书 P105 3.7.11 的公式（该公式等效于题目中要求的
3.7.8）
figure(2);
imshow(hb);
title('高提升滤波处理后的图片');

```

(2) 结果讨论：

原图和高提升滤波处理后的图像对比（A=1.7）：

