

贵州大学实验报告

学院：大数据与信息工程学院

专业：通信工程

班级：通信 152 班

姓名	朱桐	学号	1500890087	实验组	
实验时间	2018/06/06	指导教师	刘洪	成绩	
实验项目名称	实验二 数字图像的空域增强处理				
实验目的	通过本实验的学习使学生熟悉和掌握数字图像中的空域增强的一些典型方法：直方图的均衡化处理、平滑滤波、锐化滤波。				
实验要求	<ol style="list-style-type: none">1. 认真阅读实验手册，严格遵守实验室安全管理规定，规范使用设备；2. 保存实验过程中的输出图像及结果；3. 认真撰写实验报告并及时上交。				
实验原理	<p>直方图均衡化：</p> <p>基本思想：把原始图的直方图变换为均匀分布的形式，这样就增加了灰度值的动态范围从而达到增强图像整体对比度的效果。其优点是能自动地增强整个图像的对比度。</p> <p>均衡化步骤：(1)列出原始灰度级；</p> <p>(2)归一化原始灰度级；</p> <p>(3)统计原始直方图个灰度级像素 n_k ；</p> <p>(4)计算原始直方图；</p> <p>(5)计算累计直方图；</p> <p>(6)确定映射关系（原则是：取最靠近的灰度级别）；</p> <p>(7)统计新直方图各灰度级像素 n_k</p> <p>(8)计算新直方图；</p> <p>也可以用以下方法：</p> <p>(1)计算出原始图像的所有灰度级；</p> <p>(2)统计原始图像各灰度级的像素数；</p>				

	<p>(3)计算原始图像的直方图;</p> <p>(4)计算原始图像的累积直方图</p> <p>(5)取整运算</p> <p>(6)定义映射关系 s_k、t_k;</p> <p>(7)统计新直方图各灰度级的像素数;</p> <p>(8)计算新的直方图;</p> <p>2、邻域平均法</p> <p>对含噪声原始图像 $f(x, y)$ 的每个像素点取一个邻域 S。计算 S 中的所有像素灰度级的平均值, 作为空间域平均处理后图像 $g(x, y)$ 的像素值。</p> <p>3、 中值滤波</p> <p>对含噪声原始图像 $f(x, y)$ 的每个像素点取一个邻域 S (一般选取奇数点模板窗口)。选取 S 中的所有像素灰度级的中间值, 作为空间域平均处理后图像 $g(x, y)$ 的像素值。</p> <p>4、拉普拉斯运算实现图像锐化</p>
实 验 仪 器	<p>1. 硬件环境</p> <p>计算机 一台</p> <p>2. 软件环境</p> <p>MATLAB 2016b</p> <p>Windows10 64bit Professional</p>
实 验 步 骤	<p>1、编写函数 imagehist。</p> <p>其功能为: 求给定图像直方图, 并显示。</p> <p>2、 编写函数 imageeq1。其功能为: 对图像做直方图均衡化处理。</p> <p>3、 调用函数 imagehist, 来求图像 lenna.tif 的直方图。调用 MATLAB 函数 imhist, 来求图像 lenna.tif 的直方图。并比较 imagehist 和 imhist 两者所得结果有无异同。</p> <p>4、编写 m 文件。调用函数 imageeq1 和库函数 histeq, 对图像 lenna.tif 作直方图均衡化处理。同时显示原图像、原始图像的直方图、均衡化的图像、均衡化后的直方图。比较两种增强方式的结果。</p> <p>5、编写均值滤波函数和中值滤波函数。</p> <p>6、分别利用自编函数和库函数对图像进行图像去噪处理。</p>

	7、对图像进行拉普拉斯运算并比较结果。
实 验 内 容	<p>1、对灰度图像进行直方图的均衡化处理，达到图像增强的目的。</p> <p>2、利用平均滤波和中值滤波对图像进行平滑滤波。</p> <p>3、利用拉普拉斯运算对图像进行锐化滤波。</p>
实 验 数 据	<pre> %% Exp2 close all clear, clc %% Q1 im = imread('./data/jingjing.jpg'); % im = imread('peppers.png'); imGray = rgb2gray(im); imGrayHistMat = imageHist(imGray); figure subplot(1,2,1) imshow(imGray) title('Original Image') subplot(1,2,2) bar(imGrayHistMat) axis([0, 255, 0, inf]) title('Gray Histogram') %% Q2 [imEql, relation, imAccumulation] = imageEql(imGray); figure subplot(2,2,1) imshow(imGray) title('Original Image') subplot(2,2,2) bar(imGrayHistMat) </pre>

```

axis([0, 255, 0, inf])
title('Gray Histogram')

subplot(2,2,3)
imshow(imEql)
title('Image after imEql')
subplot(2,2,4)
bar(imageHist(imEql))
axis([0 255 0 inf])
title('Gray Histogram after imEql')

%% Q3
figure

subplot(1,2,1)
bar(imGrayHistMat)
axis([0, 255, 0, inf])
title('Gray Histogram By imageHist')

subplot(1,2,2)
imhist(imGray)
axis([0 255 0 inf])
title('Gray Histogram By Inner Function')

%% Q4
figure
subplot(2,2,1)
imshow(imEql)
title('Image after imEql')

subplot(2,2,2)
bar(imageHist(imEql))
axis([0, 255, 0, inf])
title('Gray Histogram Equalization By imEql')

subplot(2,2,3)
imshow(histeq(imGray))
title('Image after histeq')

subplot(2,2,4)
imeql = histeq(imGray);
bar(imageHist(imeql));
axis([0 255 0 inf])
title('Gray Histogram Equalization By Inner Function')

```

```
%% Q5
```

```
imWithGaussianNoise = imnoise(imGray, 'gaussian', 0, 0.05);  
imFilterGaussianNoise = eightNeighborFilter(imWithGaussianNoise);
```

```
figure  
subplot(1,2,1)  
imshow(imWithGaussianNoise)  
title('Image with Gaussian Noise')  
subplot(1,2,2)  
imshow(imFilterGaussianNoise)  
title('Image after Eight-neighbor Filter')
```

```
figure  
subplot(1,2,1)  
imshow(imWithGaussianNoise)  
title('Image with Gaussian Noise')  
subplot(1,2,2)  
imshow(median(imGray))  
title('Image after Median Filter')
```

```
%% Q6
```

```
[rows, cols] = size(imGray);  
c = zeros(rows, cols);  
imDouble = im2double(imGray);  
  
for row = 2:rows-1  
    for col = 2:cols-1  
        c(row,col) = 4*imDouble(row, col) - imDouble(row-1, col) -  
imDouble(row+1, col) - imDouble(row, col-1) - imDouble(row, col+1);  
    end;  
end;
```

```
figure  
subplot(1,2,1)  
imshow(imGray)  
title('Original Image')  
subplot(1,2,2)  
imshow(imDouble-c)  
title('Image after Laplace Operator')
```

```
figure  
subplot(1,2,1)  
imshow(imGray)
```

```
title('Original Image')
subplot(1,2,2)
w = fspecial('laplacian', 0);
g1 = imfilter(imDouble, w, 'replicate');
imshow(imDouble-g1)
title('Image after imfilter')

function re = eightNeighborFilter(im)

[rows, cols] = size(im);
C = im;

for i = 2:rows-1
    for j = 2:cols-1
        C(i, j) = (sum(sum( C(i-1:i+1, j-1:j+1) )))/9;
    end
end

re = C;
end

function re = medium(im)

[rows, cols] = size(im);
C = im;

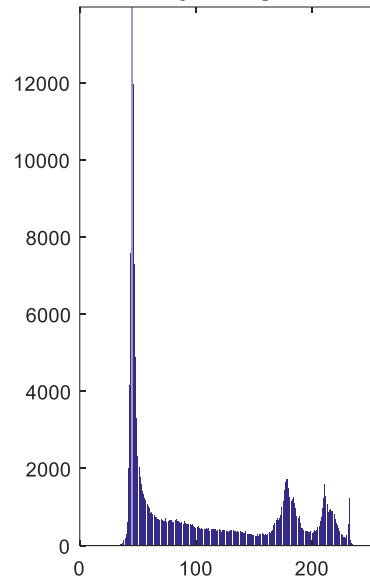
for i = 2:rows-1
    for j = 2:cols-1
        C(i, j) = median( median( C(i-1:i+1, j-1:j+1) ) );
    end
end

re = C;
end
```

Original Image



Gray Histogram



Original Image



Gray Histogram

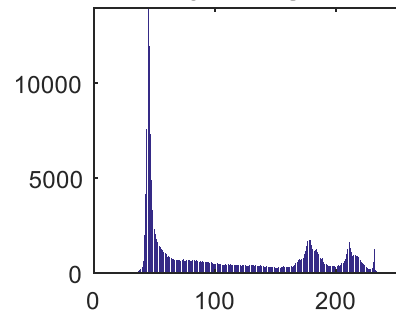
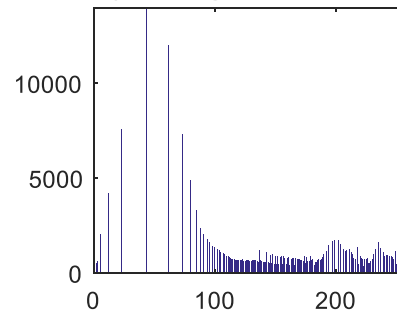


Image after imEql



Gray Histogram after imEql



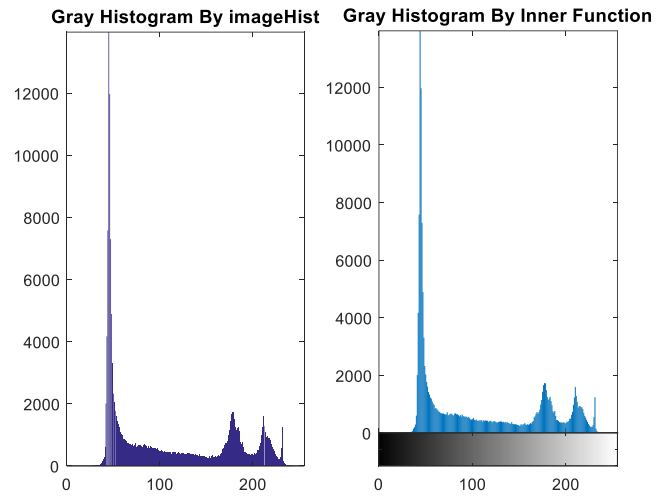


Image after imEql



Gray Histogram Equalization By imEql

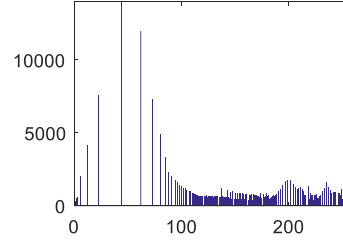


Image after histeq



Gray Histogram Equalization By Inner Function

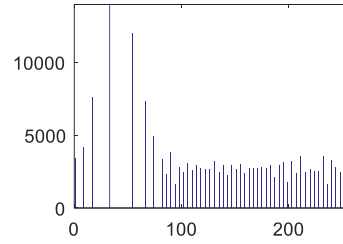


Image with Gaussian Noise



Image after Eight-neighbor Filter



Image with Gaussian Noise



Image after Median Filter



Original Image



Image after Laplace Operator



Original Image



Image after imfilter



实验总结	<p>在本次试验中使用直方图均衡化时应注意不要拿已经被操作过的图像进行再次操作，会导致重复计算，最终导致映射失败。</p> <p>另外，在本次实验中，还运用了拉普拉斯算子进行图像锐化处理。但是值得注意的是，四邻域拉普拉斯算子的模板应为如下表达式：</p> $\nabla^2 f(x, y) = 4f(x, y) - f(x-1, y) - f(x+1, y) - f(x, y-1) - f(x, y+1)$ <p>锐化滤波的本质还是利用模板函数使得对原有图像进行求导处理。故将上述算子模板实例化之后可写成如下矩阵模板：</p> $\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$
指导教师意见	<p>签名： 年 月 日</p>