



电子科技大学
University of Electronic Science and Technology of China

实验报告

实验课程： 光电图像处理

姓 名： 李宁

学 号： 2016050201017

实验地点： 211 楼 909

指导老师： 张静

实验时间： 2018 年 09 月 23 日

一. 实验名称：空间图象增强（一）

二. 实验目的

1. 熟悉和掌握利用 matlab 工具进行数字图像的读、写、显示、像素处理等数字图像处理的基本步骤和流程。
2. 熟练掌握各种空间域图像增强的基本原理及方法。
3. 熟悉通过灰度变换方式进行图像增强的基本原理、方法和实现。
4. 熟悉直方图均衡化的基本原理、方法和实现。

三. 实验原理

（一）数字图像的灰度变换

1、线性灰度变换

1)、当图像成像过程曝光不足或过度，或由于成像设备的非线性和图像记录设备动态范围太窄等因素，都会产生对比度不足的弊病，使图像中的细节分辨不清，这时可将灰度范围进行线性扩展/压缩，这种处理过程被称为图像的**线性灰度变换**。

2)、设 $f(x,y)$ 灰度范围为 $[a,b]$ ， $g(x,y)$ 灰度范围为 $[c,d]$ ，则有：

$$g(x,y) = \begin{cases} d & f(x,y) > b \\ \frac{d-c}{b-a}[f(x,y)-a] + c & a \leq f(x,y) \leq b \\ c & f(x,y) < a \end{cases}$$

2、非线性灰度变换——幂率（伽马）变换

1)、 $g(x,y) = cf(x,y)^\gamma$

其中， f 为输入灰度级， g 为变换后的灰度级， c 为一个比例因子。 γ 为一个幂指数，不同的取值范围，将产生不同的变换效果。

2)、幂律（伽马）变换曲线

$$g(x,y) = cf(x,y)^\gamma, \quad \gamma > 1$$

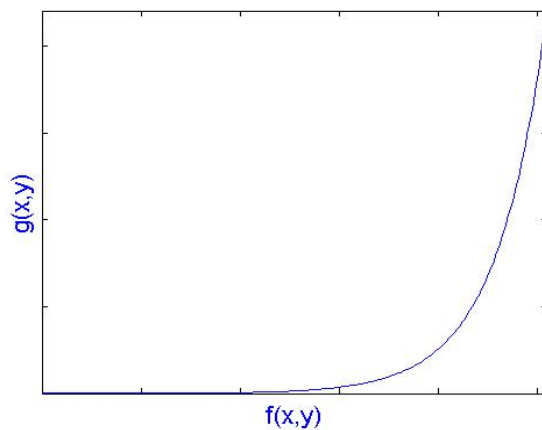


图 1

(二) 直方图处理

1、定义

1)、定义(1)

一幅灰度级范围在 $[0, L-1]$ 的数字图像的直方图定义为以下离散函数，即：

$$h(r_k) = n_k$$

其中， n_k 是图像中灰度级为 r_k 的像素个数。 r_k 是第 k 个灰度级， $k = 0, 1, 2, \dots, L-1$ 。这里，直方图表示图像中不同灰度级像素出现的次数，或者不同灰度级的像素数目。

2)、定义(2)

一幅灰度级范围在 $[0, L-1]$ 的数字图像的直方图定义为以下离散函数，即：

$$p(r_k) = n_k / n$$

其中, n 是图像的像素总数, n_k 是图像中灰度级为 r_k 的像素个数; r_k 是第 k 个灰度级, $k= 0,1,2,\dots,L-1$ 。此处，直方图表示图像中不同灰度级像素出现的概率。

2、直方图均衡化

1)、直方图均衡化是将原图像的直方图通过变换函数修正为均匀的直方图，然后按均衡直方图修正原图像。图像均衡化处理后，图像的直方图是平直的，即各灰度级具有近似相同的出现频数。那么，由于灰度级具有均匀的概率分布，图像看起来就更清晰了。

寻找一种变换: $S=T(r)$,使直方图变平直。为使变换后的灰度仍保持从暗到亮的单一变化顺序，且变换范围与变换前保持一致，以避免整体变亮或变暗。一般规定：

正变换: 在 $0 \leq r \leq 1$ 中， $T(r)$ 是单调递增函数，且 $0 \leq T(r) \leq 1$;

反变换: $r=T^{-1}(s)$, $T^{-1}(s)$ 也为单调递增函数，即 $0 \leq s \leq 1$ 。

考虑到灰度变换不影响像素的位置分布，也不会增减像素数目。所以有：

$$\int_0^r p(r)dr = \int_0^s p(s)ds = \int_0^s 1 \cdot ds = s = T(r)$$

$$T(r) = \int_0^r p(r)dr$$

3)、离散情况

设一幅数字图像的像素总数为 n ，分为 L 个灰度级。若 n_k 为第 k 个灰度级出现的频数，则第 k 个灰度级出现的概率：

$$p(r_k) = n_k / n$$

其中， $0 \leq k \leq L-1$ ， $k=0,1,2,\dots,L-1$

则，灰度级的变换函数为：

$$s_k = T(r_k) = (L-1) \sum_{j=0}^k p(r_j) = (L-1) \sum_{j=0}^k \frac{n_j}{n}$$

(三) 数字图像的空域平滑滤波

1、邻域均值滤波

1)、模板运算

$$h(i,j) = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} * f(x,y) = \begin{bmatrix} \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & 212 & 200 & 198 & \cdot \\ \cdot & 206 & 205 & 201 & \cdot \\ \cdot & 208 & 205 & 207 & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \end{bmatrix}$$

等效：模板 h 与原图像 f 作滤波(相关)运算，即：

$$\bar{g}(x,y) = \sum_{j=-1}^1 \sum_{i=-1}^1 h(i,j) f(x+i,y+j)$$

加权平均模板：

$$h_{avg} = \frac{1}{10} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

高斯模板：

$$H_g = \frac{1}{273} \begin{bmatrix} 1 & 4 & 7 & 4 & 1 \\ 4 & 16 & 26 & 16 & 4 \\ 7 & 26 & 41 & 26 & 7 \\ 4 & 16 & 26 & 16 & 4 \\ 1 & 4 & 7 & 4 & 1 \end{bmatrix}$$

（四）数字图像的空域高通滤波

1、一阶梯度算子

1)、Prewitt operator

$$g_x = \frac{\partial f(x,y)}{\partial x} = (z_7 + z_8 + z_9) - (z_1 + z_2 + z_3)$$

$$g_y = \frac{\partial f(x,y)}{\partial y} = (z_3 + z_6 + z_9) - (z_1 + z_4 + z_7)$$

$$g_x = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}, g_y = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

2)、梯度合成规则

$$M_1(x,y) = \sqrt{(g_x)^2 + (g_y)^2}$$

$$M_2(x,y) = |g_x| + |g_y|$$

$$M_3(x,y) = \max(|g_x|, |g_y|)$$

四. 实验步骤

（一）数字图像的灰度变换

1、线性灰度变换

1)、读取一幅对比度低的灰度图像（如图 2），并显示

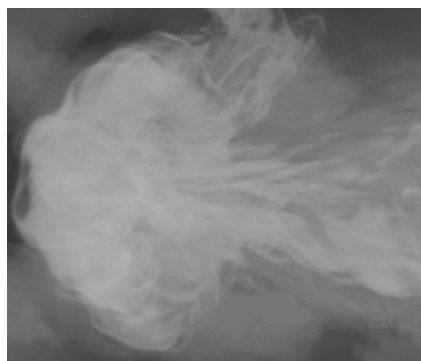


图 2

2) 以 m 文件形式编写 matlab 代码，实现数字图像的灰度范围由[a,b]到[c,d]的线性拉伸，以便于提升原图像的对比度。线性灰度变换公式如下：

$$g(x,y) = \begin{cases} d & f(x,y) > b \\ \frac{d-c}{b-a}[f(x,y)-a]+c & a \leq f(x,y) \leq b \\ c & f(x,y) < a \end{cases}$$

其中，f(x,y)为原始图像，灰度范围为[a,b]，g(x,y)为增强后的数字图像，灰度范围为[c,d]。注：实验中[a,b]可由实际图像来确定，[c,d]可视具体情况人为给定。

3) 显示经过线性灰度变换后的图像，并对变换结果进行必要的分析，画出灰度变换曲线图。（要求：在同一窗口中分别显示原始图像、变换结果及其直方图。）

4) 以另一个文件名形式保存灰度变换后的图像。

2、非线性灰度变换——幂率（伽马）变换

1)、编写 matlab 程序，具体步骤同线性灰度变换，非线性变换灰度公式为：

$$g(x,y) = cf(x,y)^\gamma$$

本实验区 $c=1$ ， $\gamma=0.89$ 。

(二) 直方图处理

1) 读取一幅灰度图像（如图 1-3），并显示原始图像。



图 3

2) 编写 `m` 文件实现对输入图像 2 种定义下的直方图统计统计（即各个灰度级出现的次数及概率分布），并分别画出两种定义下的直方图。

3) 对输入灰度图像进行直方图均衡化处理，分别显示均衡处理前后的图像和对应的直方图。

基本要求：根据直方图均衡化原理自行编写基础代码，不能直接调用 `histeq()` 函数进行处理。

(三) 数字图像的空域平滑滤波

1、邻域均值滤波

1)、读入一幅原始灰度图像。



图 4

2)、在读入的原始图像中加入高斯白噪声。

3)、设计如下空间域的 3×3 均值滤波器 H1 和 5×5 高斯滤波器 H2, 如下式所示

$$H_1 = \frac{1}{10} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$
$$H_g = \frac{1}{273} \begin{bmatrix} 1 & 4 & 7 & 4 & 1 \\ 4 & 16 & 26 & 16 & 4 \\ 7 & 26 & 41 & 26 & 7 \\ 4 & 16 & 26 & 16 & 4 \\ 1 & 4 & 7 & 4 & 1 \end{bmatrix}$$

4)、分别利用 H1 和 H2 对第(2)步中加噪声的图像进行滤波处理, 对处理结果进行必要的分析。

5)、显示原图、加噪声图及处理结果。

(四) 数字图像的空域高通滤波

1、一阶梯度算子

1)、读入一幅原始图像。

2)、设计如下所示的 3×3 一阶梯度算子 (Prewitt 滤波器)。

$$g_x = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}, g_y = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

3)、利用以上式滤波器, 对第(1)步中读入图像分别进行滤波处理。

4)、分别显示原图、x/y 方向梯度图及合成梯度图。



图 5

五. 实验结果及分析

(一)、数字图像灰度变换

1、线性灰度变换

1)、实验结果

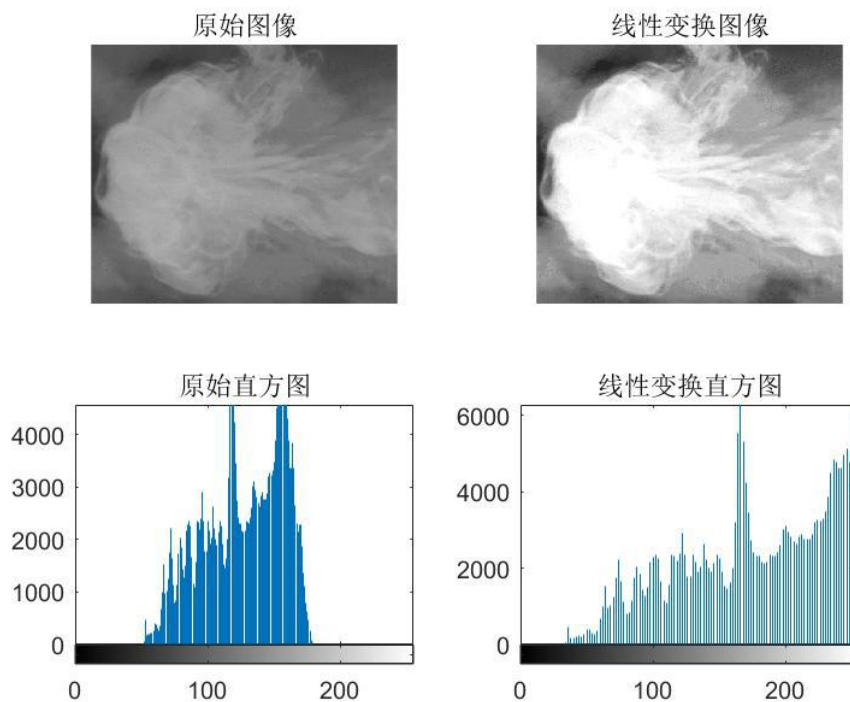


图 6

2)、结果分析

由图像的灰度直方图我们看到，灰度图像明显的被拉伸，根据编写的代码可知灰度的范围向低处扩展 20，向高处扩展 50。

2、非线性变换——幂率变换

1)、实验结果

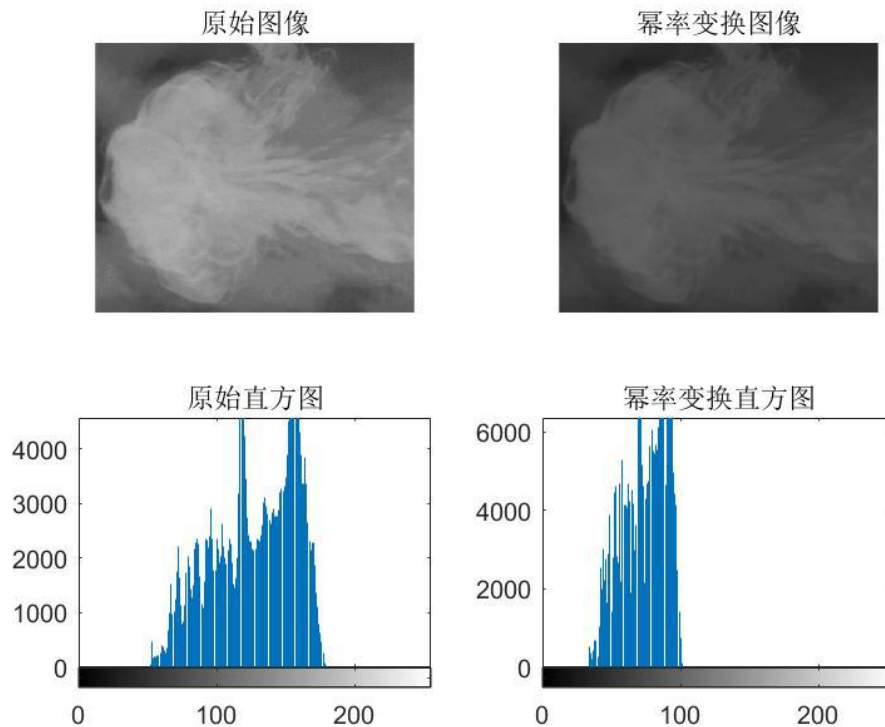


图 7

2)、结果分析

由图像我们发现，图像的灰度被向低处压缩，这和先前参数 $c=1$, $\gamma=0.89$ 一致。

(二)、直方图处理

1、两种定义下（频数和频率）的直方图统计

1)、实验结果

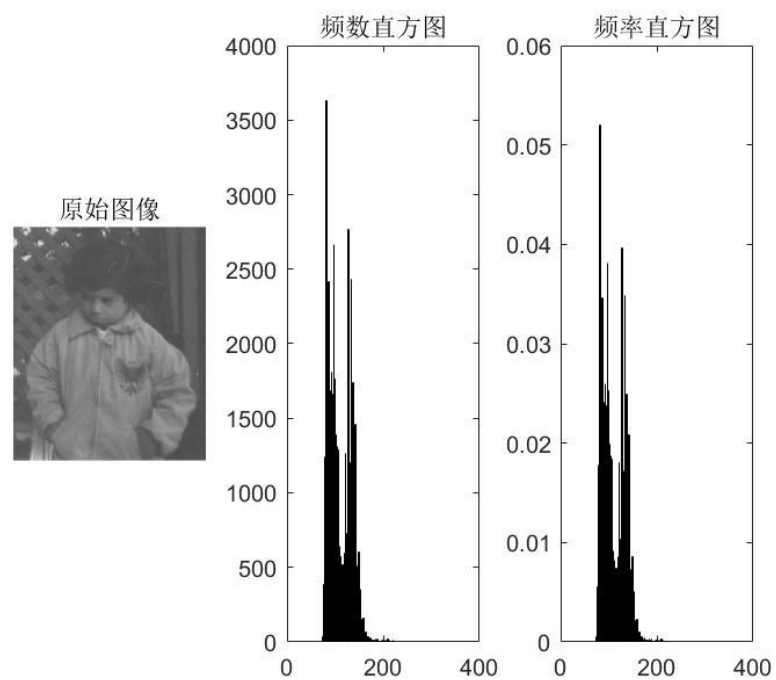


图 8

2)、结果分析

由图像我们可以看出两种方法画出的直方图样式无差别, 仅仅纵坐标不相同。

2、直方图均衡化处理

1)、实验结果

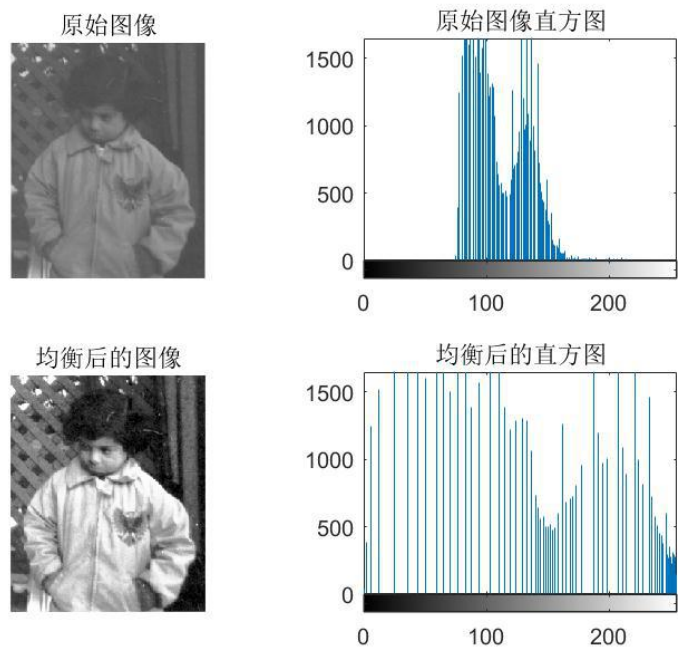


图 9

2)、结果分析

从均衡化后的结果我们可以看出，图画的对比度明显增强，直方图由先前的集中分布于[90,170]之间变为了大致均匀分布于[0,255]之间。

(三)、数字图像的空域平滑滤波

1、邻域均值滤波

1)、实验结果



图 10

2)、结果分析

第三幅和第四幅分别是均值滤波后的图像和高斯滤波后的图像,可以看出去除了一些高斯噪声,代价便是图像变得模糊。

(四)、数字图像的空域高通滤波

1、一阶梯度算子

1)、实验结果



图 11

2)、结果分析

从图中可以看出经过高通滤波后，图像的边缘锐化，人物轮廓更为清晰。

六. 实验心得体会和建议

心得体会：

上机实验使我复习了图像增强的相关知识，也让我了解了书本上算法的具体作用，但是我对算法的原理了解不是很深刻，这导致应用编程的时候不能得心应手，今后应加强这些方面的学习。

图像锐化部分原理和编程实现不清楚。

建议：

把不可以使用的语句代码标注出来，或者提出希望我们能用什么样代码的要求。

七. 程序源代码

（一）数字图像的灰度变换

1、线性灰度变换&2、非线性灰度变换（幂率变化）

```
clc,clear;
close all;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
f0=imread('train1.jpg');%读取图像
f=rgb2gray(f0);%灰度处理
[m,n]=size(f);
a=min(min(f));
b=max(max(f));%灰度范围
c=a-20;
d=b+50;%处理后的灰度范围
g=zeros(m,n);%产生一个等大图像
g=((d-c)/(b-a))*(f-a)+c;
figure,subplot(2,2,1),imshow(f),title('原始图像')
subplot(2,2,2),imshow(g),title('线性变换图像')
subplot(2,2,3),imhist(f),title('原始直方图')
subplot(2,2,4),imhist(g),title('线性变换直方图')%画出四个图像
imwrite(g,'C:\Users\Administrator\Pictures\train1_plus.jpg')%保存修改
的图像
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%幂率变换
f0=double(f);
g0=f0.^0.89;
g1=uint8(g0);
figure,subplot(2,2,1),imshow(f),title('原始图像')
subplot(2,2,2),imshow(g1),title('幂率变换图像')
subplot(2,2,3),imhist(f),title('原始直方图')
subplot(2,2,4),imhist(g1),title('幂率变换直方图')%画出四个图像
```

（二）、直方图处理

1、两种定义下（频数和频率）的直方图统计&2、直方图均衡化处理

```
clc,clear;
close all;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%绘制两种定义下的直方图
f=imread('train2.tif');
```

```

[height,width]=size(f);
a=min(min(f));%图像的最小灰度值
b=max(max(f));%图像的最大灰度值
x=unique(f);%x 为所有不同灰度值的集合
y=zeros(size(x));%y 为存储相应于不同灰度值频数的数组
for i=1:length(x)%相应灰度值的频数
    y(i)=length(find(f==x(i)));
end
frequ=y./(height*width);%相应灰度值的频率
figure,subplot(1,3,1),imshow(f),title('原始图像')%原始图像
subplot(1,3,2),bar(x,y),title('频数直方图')%频数直方图
subplot(1,3,3),bar(x,frequ),title('频率直方图')%频率直方图
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%直方图均衡化
r=mat2gray(x);%归一化的原始灰度
s=zeros(length(r),1);%直方图修正后的灰度
for i=1:length(r)%直方图累加
    if i==1
        s(i)=frequ(i);
    else
        s(i)=s(i-1)+frequ(i);
    end
end
ss=uint8(255.*s);%累积分布取整
g=zeros(height,width);%预先分配内存
for i=1:height%灰度值映射
    for j=1:width
        g(i,j)=ss(x==f(i,j));
    end
end
g=uint8(g);%数据类型转换
sk=unique(ss);
sk_num=zeros(length(sk),1);
for i=1:length(sk)
    sk_num(i)=sum(y(ss==sk(i)));%从修正后的灰度中寻找相等
                                ...的灰度并把相等灰度所对应的
                                ...原始灰度的频数相加，成为
                                ...现有灰度的频数。
end
frequ_sk=sk_num./(height*width);%均衡化后的灰度值频率
figure,subplot(2,2,1),imshow(f),title('原始图像')%原始图像
subplot(2,2,2),imhist(f),title('原始图像直方图')%原始图像直方图
subplot(2,2,3),imshow(g),title('均衡后的图像')%直方图均衡化后的图像
subplot(2,2,4),imhist(g),title('均衡后的直方图')%均衡后直方图频率

```


(三)、数字图像的空域平滑滤波

1、邻域均值滤波

```
clc,clear;
close all;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
f0=imread('cameraman.tif');
[height,width]=size(f0);
noise=wgn(height,width,30);%产生高斯白噪声
f=f0+uint8(noise);%叠加高斯白噪声
figure,subplot(2,2,1),imshow(f0),title('原始图像')
subplot(2,2,2),imshow(f),title('含高斯噪声图像')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
H1=(1/10).*[1 1 1;
           1 2 1;
           1 1 1];%均值滤波器
H2=(1/273).*[1 4 7 4 1;
            4 16 26 16 4;
            7 26 47 26 7;
            4 16 26 16 4;
            1 4 7 4 1];%高斯滤波器
N1=3;
N2=5;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%均值滤波
f1=double(f);
f2=f1;
for i=1:(height-N1+1)%去掉边界
    for j=1:(width-N1+1)
        chip1=f1(i:i+N1-1,j:j+N1-1).*H1;
        s1=sum(sum(chip1));
        f2(i+(N1-1)/2,j+(N1-1)/2)=s1;
    end
end
f_last=uint8(f2);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%高斯滤波
g1=double(f);
g2=g1;
for i=1:(height-N2+1)%去掉边界
    for j=1:(width-N2+1)
        chip2=g1(i:i+N2-1,j:j+N2-1).*H2;
        s2=sum(sum(chip2));
```

```

        g2(i+(N2-1)/2,j+(N2-1)/2)=s2;
    end
end
g_last=uint8(g2);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
subplot(2,2,3),imshow(f_last),title('均值滤波图像')
subplot(2,2,4),imshow(g_last),title('高斯滤波图像')

```

(四)、数字图像的空域高通滤波

1、一阶梯度算子

```

clc,clear;
close all;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
f=imread('cameraman.tif');
[height,width]=size(f);
figure,subplot(2,2,1),imshow(f),title('原始图像')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
gx=[-1 0 1;%x 方向梯度算子
    -1 0 1;
    -1 0 1];
gy=[-1 -1 -1;%y 方向梯度算子
    0 0 0;
    1 1 1];
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
N1=3;
f1=double(f);
g1=zeros(height,width);%x、y、合成方向梯度算子处理后的图像元素
g2=zeros(height,width);
g3=zeros(height,width);
for i=1:height-N1+1%去掉边界
    for j=1:(width-N1+1)
        chip1=f1(i:i+N1-1,j:j+N1-1).*gx;%进行相关的第一步运算，得
            ...到同位置元素相乘后的九宫格
        chip2=f1(i:i+N1-1,j:j+N1-1).*gy;
        s1=sum(sum(chip1));%九宫格的值相加
        s2=sum(sum(chip2));
        s_sy=sqrt(s1^2+s2^2);
        g1(i+(N1-1)/2,j+(N1-1)/2)=s1;
        g2(i+(N1-1)/2,j+(N1-1)/2)=s2;
        g3(i+(N1-1)/2,j+(N1-1)/2)=s_sy;
    end
end
y1=uint8(g1);%转换格式

```

```

y2=uint8(g2);
y3=uint8(g3);
subplot(2,2,2),imshow(y1),title('x 方向梯度图')
subplot(2,2,3),imshow(y2),title('y 方向梯度图')
subplot(2,2,4),imshow(y3),title('合成梯度图')

```

八. 思考题

1、为了扩展一幅图像的灰度，使其最低灰度为 C、最高灰度为 L-1，试给出一个单调的灰度变换函数。

设 $f(x,y)$ 灰度范围为 $[a,b]$

$$g(x,y) = \begin{cases} L-1 & f(x,y) > L-1 \\ \frac{L-1-C}{b-a}[f(x,y)-a]+C & C \leq f(x,y) \leq L-1 \\ C & f(x,y) < C \end{cases}$$

2、数字图像经过对数变换与幂律变换后，变换结果有什么差异？请按不同参数的取值范围进行具体分析。

1)、对数变换：

当底数 $0 < n < 1$ 时，灰度处理结果为负值，图片为黑色。

当底数 $1 < n$ 时，对于同一个灰度照片，当逐渐增加时，图像的灰度会迅速减小，即图像会迅速变黑。

2)、幂率变换

当指数 $0 < r < 1$ 时，随着 r 的增加，图片的被压缩的高灰度值变缓，图片的对比度增强。

当指数 $0 < r$ 时，随着 r 的增加，图片的灰度迅速增大，图片的平均灰度值迅速增加，故图片以很快的速度变亮。

3、从连续函数的概率分布来分析，直方图均衡处理可以使得各个灰度级分布变为均匀分布，即拉平直方图。而对实际数字图像的处理结果却是一种近似平

直，而不是严格意义上的均匀直方图，试说明其理由。

从连续函数的概率分布上来讲，直方图均衡处理可以使连续的灰度级变成为均匀分布，均匀分布产生的公式中，有一项就是，两相邻灰度级之间的差值趋近于 0，但对于离散的灰度级，理论上是不可能使直方图变为均匀分布的。