

**计算机与信息学院实验报告**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 实验课程： | 数字图像处理实验 | | | | |
| 实验编号： | 实验3 | | | | |
| 实验名称： | 图像的线性和非线性变换 | | | | |
| 实验人员： | 学号 | 18111207248 | | | |
| 姓名 | 吴钰 | | | |
| 班级 | 计算机科学与技术（18创新班） | | | |
| 实验日期： | 2021.5.27 | | | | |
| 实验室： | 2060404 | | | | |
|  |  | | | | |
| 实验评价： |  | | | | |
| 实验成绩： | |  | 评价日期： |  |
|  | 指导教师： | |  | | |

# 图像的线性和非线性变换

一、实验目的

1．掌握像素点运算的各种图像处理方法；

2．理解各类变换函数实现图像对比度增强的原理；

3．能正确选用合适的变换方法对图像进行对比度增强。

二、实验要求

1．在MATLAB环境下实现图像的线性和非线性变换算法；

2．应用设计的算法实现对图像的对比度增强。

三、实验内容

1．应用对数变换、伽马变换、分段线性变换等灰度变换函数实现灰度图像的对比度增强；

2．针对具体应用场景，设计一个线性对比度拉伸算法，实现对直方图分布居中的灰度图像的对比度增强。

四、实验结果及分析

实验素材见实验三附件材料。

1．灰度变换-线性变换



（1）实验代码

%线性变换

A = imread('D:\大三（下）专业课\数字图像处理\实验三\image\beauty.jpg');

A1 = rgb2gray(A);%彩色图像转为灰度图像

figure;

subplot(2,2,1);

imshow(A1);

title('原始图像');

figure;

a1=1;b1=0;

a2=1;b2=50;

a3=1.2;b3=0;

a4=0.6;b4=0;

A21=a1\*A1+b1;

A22=a2\*A1+b2;

A23=a3\*A1+b3;

A24=a4\*A1+b4;

subplot(221),imshow(A21);

title('a1=1;b1=0;');

subplot(222),imshow(A22);

title('a2=1;b2=50;');

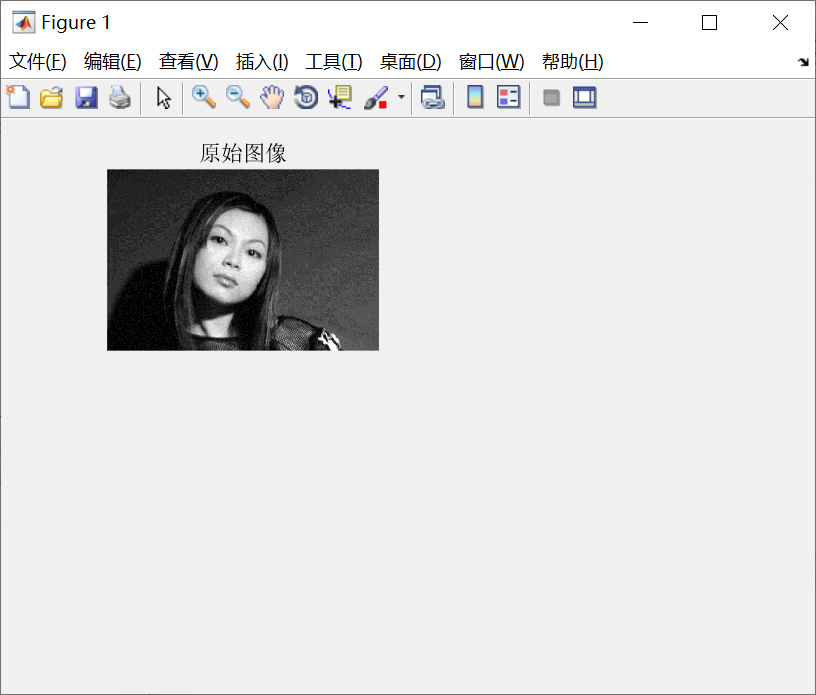
subplot(223),imshow(A23);

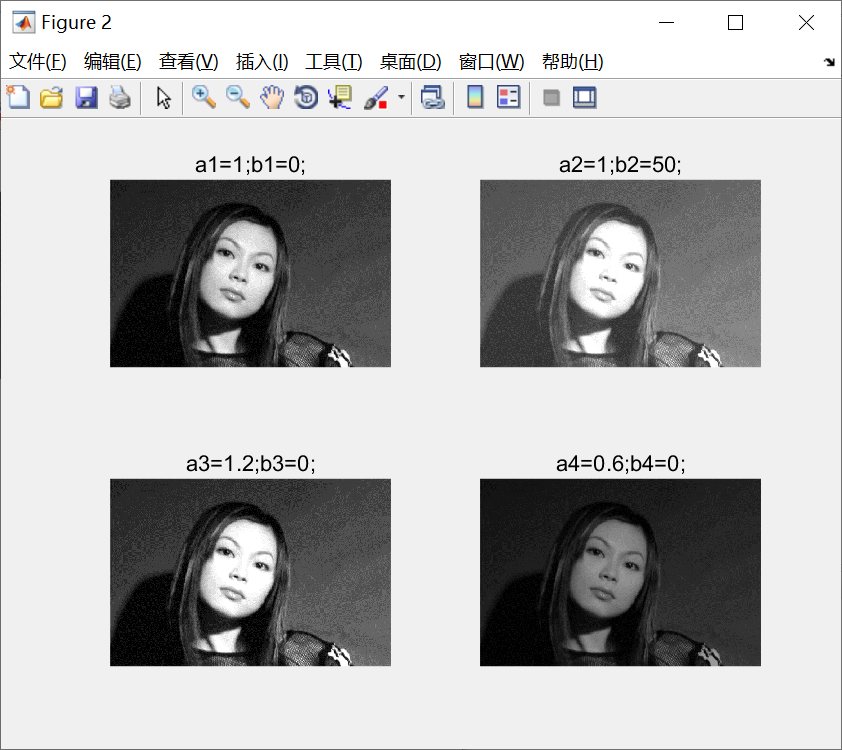
title('a3=1.2;b3=0;');

subplot(224),imshow(A24);

title('a4=0.6;b4=0;');

（2）实验结果（图示）





（3）实验结果分析

显然，当α = 1，β =0时，原图像不发生变化；

当α =1，β ≠ 0时，图像灰度值增加或降低；

当α>1时，输出图像对比度增大；

当0<α<1时，输出图像对比度减小；

当α<0时，图像亮区域变暗，暗区域变亮，即图像求补。

2．图像的对数变换



（1）实验代码

%非线性对数变换

A = imread('D:\大三（下）专业课\数字图像处理\实验三\image\DFT\_no\_log.tif');

A1 = mat2gray(log(1+double(A)));%彩色图像转为灰度图像

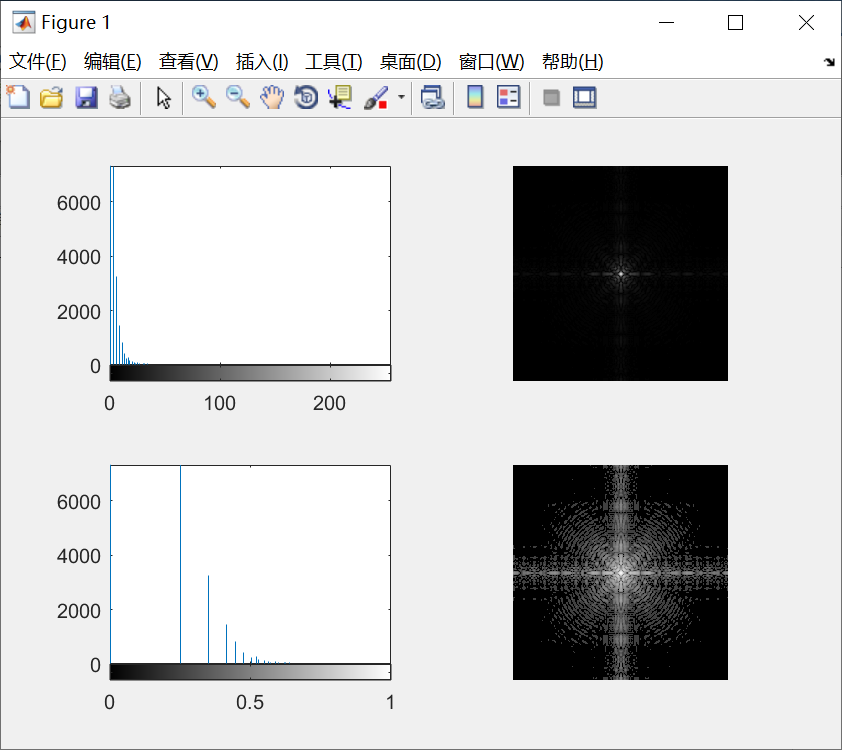
subplot(221),imhist(A)%查看灰度直方图

subplot(222),imshow(A);

subplot(223),imhist(A1)

subplot(224),imshow(A1);

（2）实验结果（图示）



（3）实验结果分析

通过灰度变换函数，调整输入低质图像的灰度值范围。将图像的低灰度值部分扩展，高灰度值部分压缩。借此强调图像低灰度部分，达到增强图像的目的。

显示器无法显示大范围灰度值时，许多灰度细节会被丢失掉，

对数变换可将其动态范围变换到一个合适的区间，就可以显示更多细节。

3．图像的幂次变换



（1）实验代码

clc;

clear;

img1=imread('D:\大三（下）专业课\数字图像处理\实验三\image\washed\_out\_aerial\_image.tif');

img3 = imread('D:\大三（下）专业课\数字图像处理\实验三\image\fractured\_spine.tif');

figure;

subplot(2,2,1);

imshow(img1);

title('原始图像');

C = 1;

X = 3;

img2 = myImageMiLv(img1,C,X);

subplot(2,2,2);

imshow(img2);

title(['C:',num2str(C),' X:',num2str(X)]);

C = 1;

X = 4;

img2 = myImageMiLv(img1,C,X);

subplot(2,2,3);

imshow(img2);

title(['C:',num2str(C),' X:',num2str(X)]);

C = 1;

X = 5;

img2 = myImageMiLv(img1,C,X);

subplot(2,2,4);

imshow(img2);

title(['C:',num2str(C),' X:',num2str(X)]);

figure;

subplot(2,2,1);

imshow(img3);

title('原始图像');

C = 1;

X = 0.6;

%img2 = myImageMiLv(img1,C,X);

img4 = myImageMiLv(img3,C,X);

subplot(2,2,2);

imshow(img4);

title(['C:',num2str(C),' X:',num2str(X)]);

C = 1;

X = 0.4;

%img2 = myImageMiLv(img1,C,X);

img4 = myImageMiLv(img3,C,X);

subplot(2,2,3);

imshow(img4);

title(['C:',num2str(C),' X:',num2str(X)]);

C = 1;

X = 0.3;

%img2 = myImageMiLv(img1,C,X);

img4 = myImageMiLv(img3,C,X);

subplot(2,2,4);

imshow(img4);

title(['C:',num2str(C),' X:',num2str(X)]);

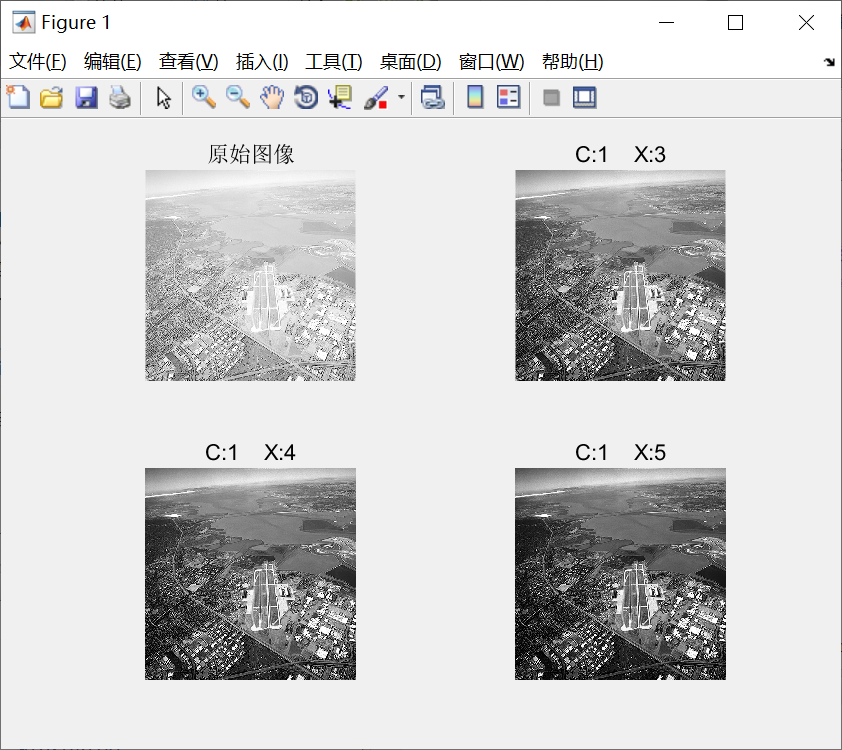
function img2 = myImageMiLv(img1,C,X)

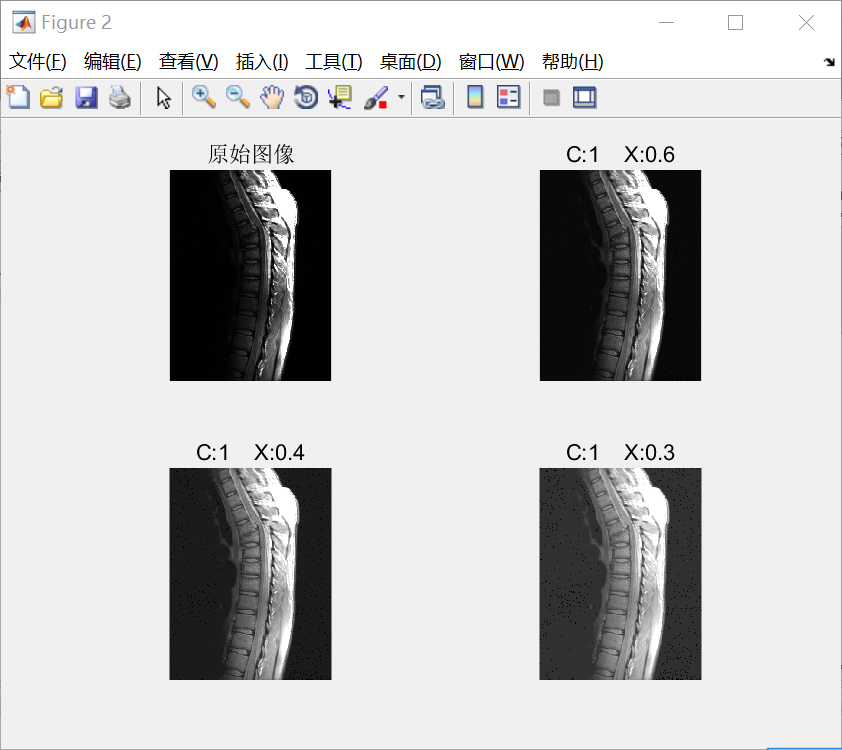
img = im2double(img1);

img2 = C\*(img .^ X);%进行幂律变换

end

（2）实验结果（图示）





（3）实验结果分析

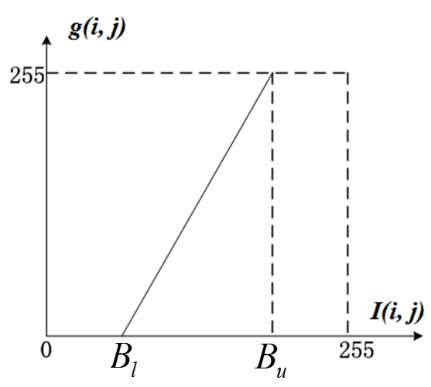
C,X（伽马）是正常数

X<1 提高灰度级，使图像变亮

X=0.4 增强效果最好

X>1降低灰度级，使图像变暗c=1,X=3,4,5

4．图像的线性对比度拉伸

（1）实验代码

LCSMain.m

clc,clear

close all

image = imread('D:\大三（下）专业课\数字图像处理\实验三\image\finger.bmp');

pi = countP(image);

para = [0.02,0.03];

img2 = LCSfun (image, para, pi);

para = [0.07,0.08];

img3 = LCSfun (image, para, pi);

para = [0.2,0.3];

img4 = LCSfun (image, para, pi);

subplot(2,4,1);

imshow(image);

title('原图像');

count1 = countP(image);

subplot(2,4,5);

bar(count1);

subplot(2,4,2);

imshow(mat2gray(img2));

title('0.02~0.03');

count1 = countP(img2);

subplot(2,4,6);

bar(count1);

subplot(2,4,3);

imshow(mat2gray(img3));

title('0.07~0.08');

count1 = countP(img3);

subplot(2,4,7);

bar(count1);

subplot(2,4,4);

imshow(mat2gray(img4));

title('0.2~0.3');

count1 = countP(img4);

subplot(2,4,8);

bar(count1);

LCSfun.m

function res = LCSfun (GrayImage, para, P)

a = para(1);

b = para(2);

Sl=0;

Sk=0;

for i = 1 : 256

if(Sl>=a)

Bl=i-1;

break;

end

Sl=Sl+P(i);

end

for i = 1 : 256

if(Sk>=b)

BB=256-i;

break;

end

Sk=Sk+P(257-i);

end

img = double(GrayImage);

[x,y] = size(img);

for i = 1 : x

for j = 1 : y

if img(i,j)<Bl

img(i,j)=0;

elseif img(i,j)<=BB

img(i,j)=(img(i,j)-Bl)\*255/(BB-Bl);

else

img(i,j)=255;

end

end

end

res = uint8(img);

end

countP.m

function res = countP(GrayImage)

%得到图像矩阵的行、列数

[rows , cols] = size(GrayImage);

res = zeros(1 , 256);

%用来存储灰度图像中每个像素出现的频率

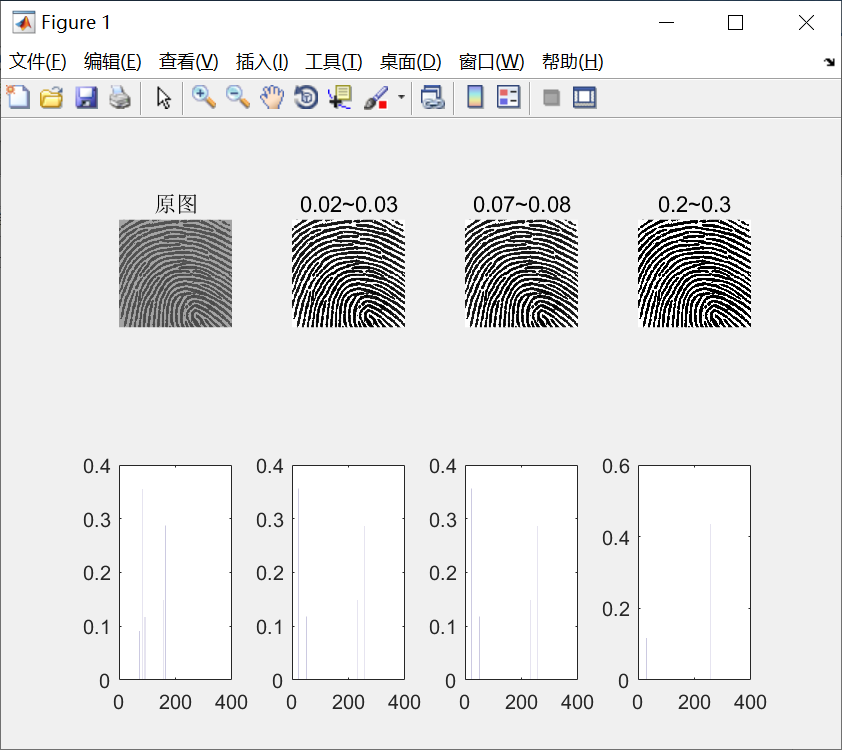
for i = 0:255

res(1 , i+1) = length(find(GrayImage == i)) / (rows \* cols);

%计算矩阵GrayImage中值与i相等的元素的个数

end

（2）实验结果（图示）



（3）实验结果分析

1.定义函数：

  定义的函数有两个参数：图像img1和灰度级区间最小值、最大值的二维数组parameter。首先对parameter参数进行判断，若灰度级的最大值和最小值不符合要求则报错。若输入的参数没有问题，就开始对图像进行处理：先使用im2double函数将uint8类型的图像数据转换为double类型的，分别取图像矩阵的最大值和最小值，使用公式利用公式把数据范围压缩到0~1之间的数值，归一化以便计算。再还原图片像素的大小为parameter区间内的数值，最后使用函数uint8将图片数据转换为uint8的格式，以便输出。

2.测试函数：

使用imread函数读取图像'finger.bmp'；

分别定义灰度级parameter = [0.02,0.03]、 parameter = [0.07,0.08]、parameter= [0.2,0.3]，函数得到对比度拉伸后的图像；

三段线性拉伸的公式：假定要将原图 r1 - r2 的区域拉伸至 s1 - s2 区域。令oldl为原图灰度级，newl为新灰度级。

oldl < r1 :  newl = oldl / r1 \* s1 ;

r1 < oldl < r2 : newl = (oldl - r1) / (r2 - r1) \* (s2 - s1) + s1 ；

oldl > r2 ：newl = (oldl - r2) / (255 - r2) \* (255 - s2) + s2 ;