华北电力大学

NORTH CHINA ELECTRIC POWER UNIVERSITY

2 数字图像基本运算

主要内容

- 点运算
  - ✓ 线性变换
  - ✓ 对数变换
  - ✓ 伽马变换
  - ✓ 灰度阈值变换
- 代数运算
  - ✓ 图像加法
  - ✓ 图像减法
  - ✓ 图像乘法
  - ✓ 图像除法

主要内容

- 几何运算
  - ✓ 齐次坐标
  - ✓ 图像平移
  - ✓ 图像镜像
  - ✓ 图像旋转
  - ✓ 图像缩放
- 插值运算
  - ✓ 最近邻插值
  - ✓ 线性（双线性）插值

2.1 点运算

- 图像点运算（Point Operation）指对于一幅输入图像f(x,y)，将产生一幅输出图像g(x,y)，T表示一种数学运算。

$$g(x,y)=T[f(x,y)]$$

- 是一种像素的逐点运算，是灰度到灰度的映射过程，通过映射变换来达到增强或者减弱图像的灰度，T又称为灰度变换函数。
- 点运算包括线性变换、对数变换、伽马变换、灰度阈值变换、直方图均衡化等。

2.1.1 图像灰度化

- 彩色图像通常包括R、G、B三个分量，分别显示出红绿蓝等各种颜色，灰度化就是使彩色图像的R、G、B三个分量相等的过程。
- 将原始RGB(R,G,B)颜色均匀地替换成新颜色RGB(Gray,Gray,Gray)，从而将彩色图片转化为灰度图像。
- 人类的眼睛感官蓝色的敏感度最低，敏感最高的是绿色，因此将RGB按照0.299、0.587、0.114比例加权平均能得到较合理的灰度图像。

$$gray = 0.299 * R + 0.587 * G + 0.114 * B$$

2.1.1 图像灰度化

$$\text{gray} = 0.299 * R + 0.587 * G + 0.114 * \text{B}$$

RGB图

灰度图

2.1.1 图像灰度化

```
I=imread('lena.jpg');
I1=0.299*I(:,:,1)+0.587*I(:,:,2)+0.114*I(:,:,3);
%创建图形窗口
figure(1);
%将多个图画到一个窗口中，参数(m, n, p)，m表示行，n表示列，p表示第几个
subplot(1,2,1);
imshow(I);
title('RGB图');
subplot(1,2,2);
imshow(I1);
title('灰度图');
```

2.1.2 线性变换

- 线性变换函数是一个一维线性函数

$$g(x,y)=k.f(x,y)+\mathrm{b}$$

✓ k>1时输出图像的对比度将会增大

✓ k<1时输出图像的对比度将会降低

✓ k=1, b!=0时，图像更暗或者更亮，不会对对比度造成影响

2.1.2 线性变换

2.1.2 线性变换

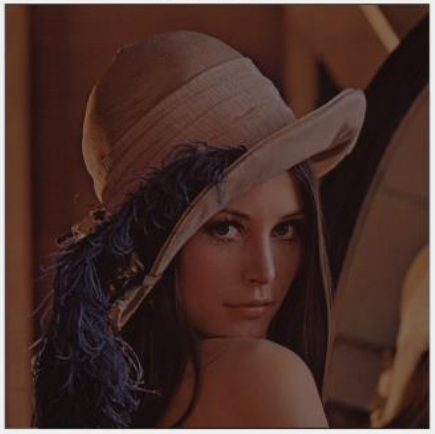
```
I=imread('face.jpg'); %I中是unit8数据，所以b表示灰度
%I = im2double(I);%转换为double，便于计算，此时b要变为b/255
figure(1);
subplot(1,2,1);
imshow(I);
title('原图像');
%对比度增强
k = 1.2;
b = 0;
I1 = k.*I + b; %对于数值与矩阵相乘.*和*相同，但是两个矩阵相乘不同
subplot(1,2,2);
imshow(I1);
title('k=1.2,b=0');
```

2.1.2 线性变换

原图像

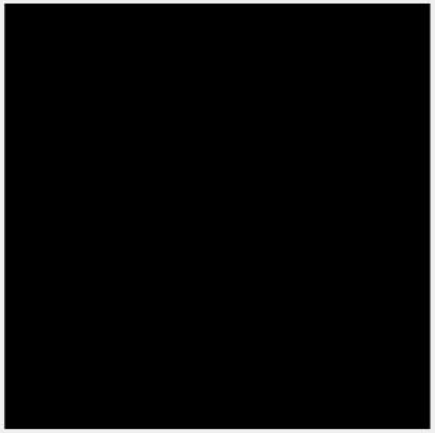
k=1.5,b=0

k=0.5,b=0



k=1,b=0.1

k=-1,b=-0.1



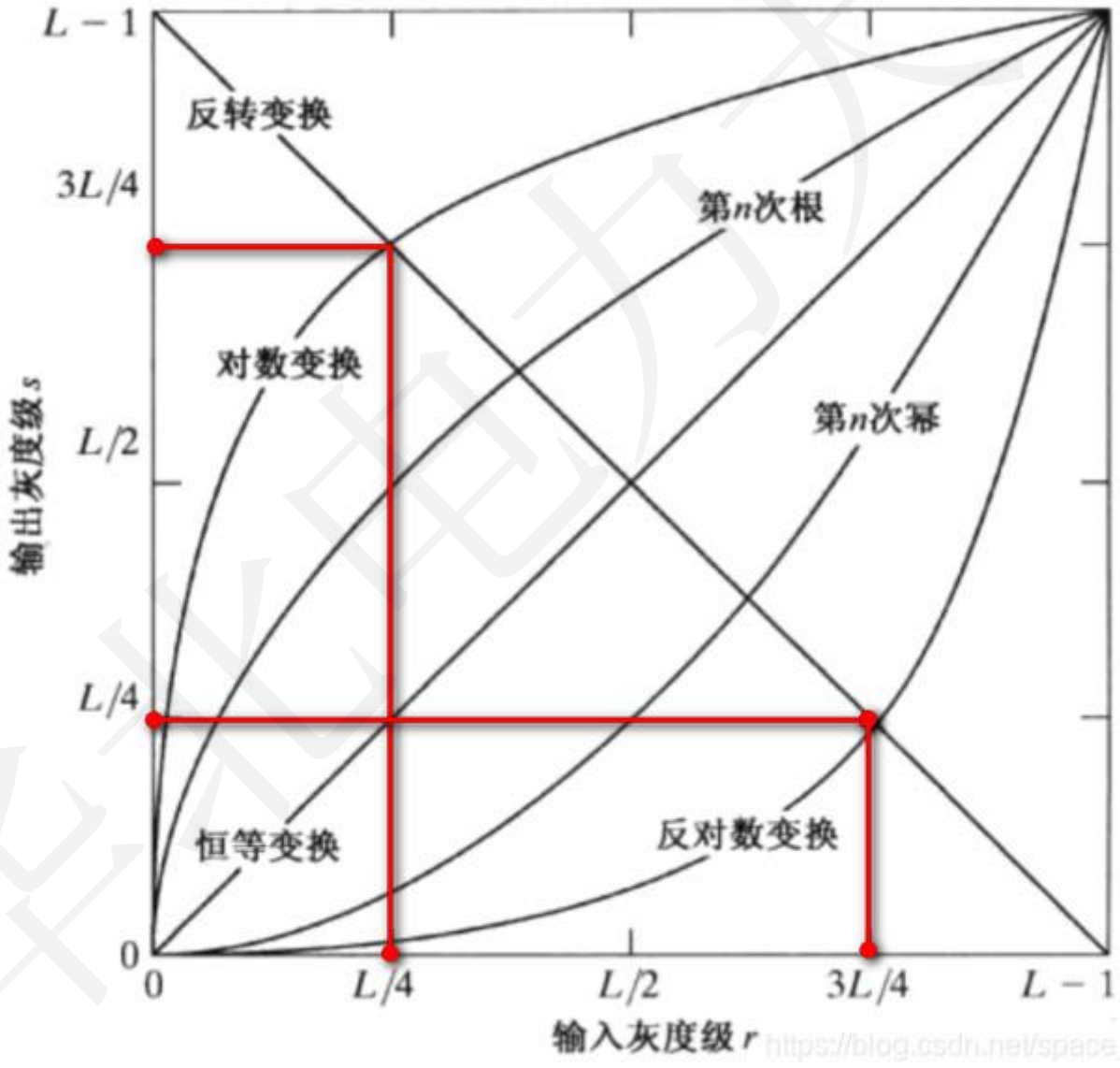
k=-1,b=1

2.1.3 对数变换

- 对数变换将输入中范围较窄的低灰度值映射为输出中范围较宽的灰度值，或将输入中范围较宽的高灰度值映射为输出中范围较窄的灰度值。

$$g(x,y)=c.\log(f(x,y)+1)$$

2.1.3 对数变换

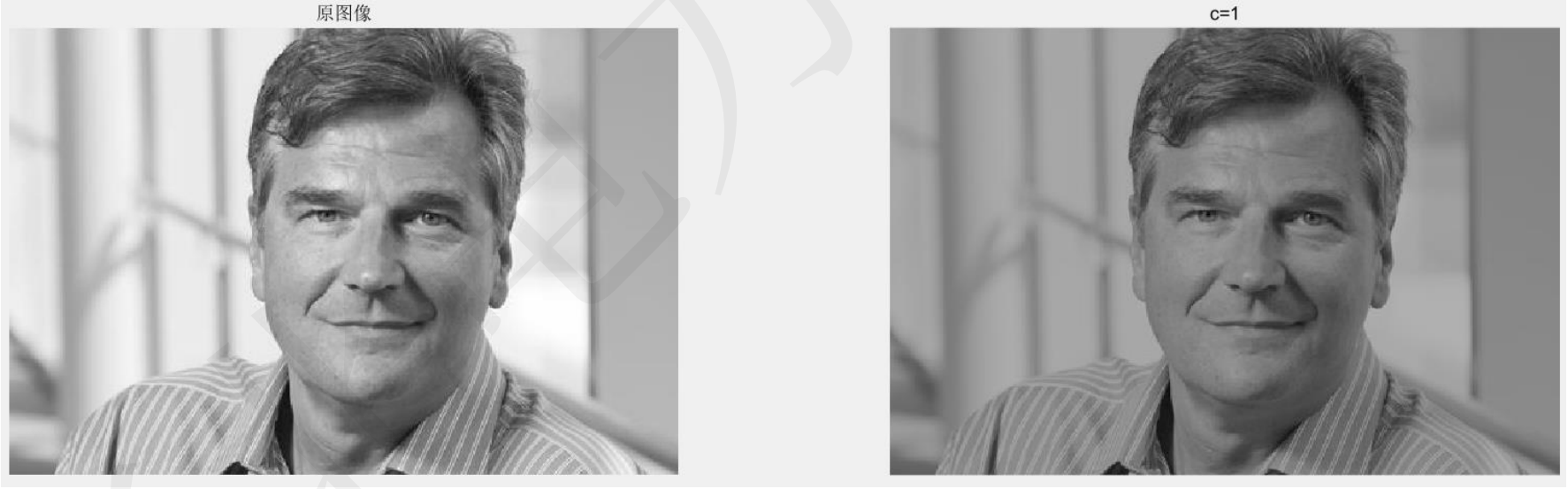

L − 1
3L/4
L/2
L/4
0
反转变换
第n次根
对数变换
第n次幂
恒等变换
反对数变换
输出灰度级 s
0
L/4
L/2
3L/4
L − 1
输入灰度级 r

2.1.3 对数变换

- 对数变换对较暗的部分进行增强。

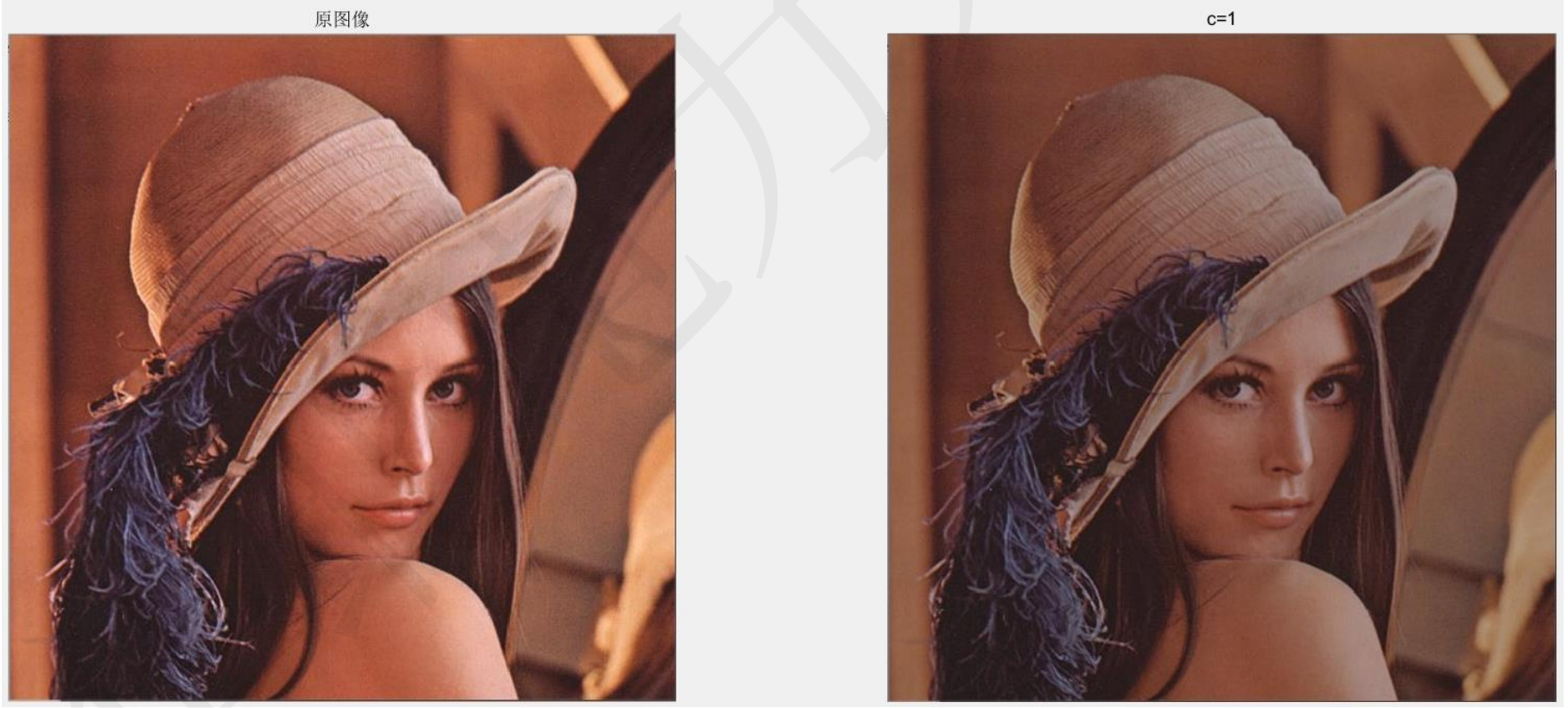
2.1.3 对数变换

- 对数变换对较暗的部分进行增强。



2.1.3 对数变换

- 对数变换对较暗的部分进行增强。



2.1.3 对数变换

```
I=imread('face.jpg'); %I中是unit8数据，所以b表示灰度
I = im2double(I); %log不支持unit8，所以必须转为double
%创建图形窗口
figure(1);
subplot(1,2,1)
imshow(I);
title('原图像');
%对数变换
c = 1;
I1 = c*log(I + 1);
subplot(1,2,2);
imshow(I1);
title('c=1');
```

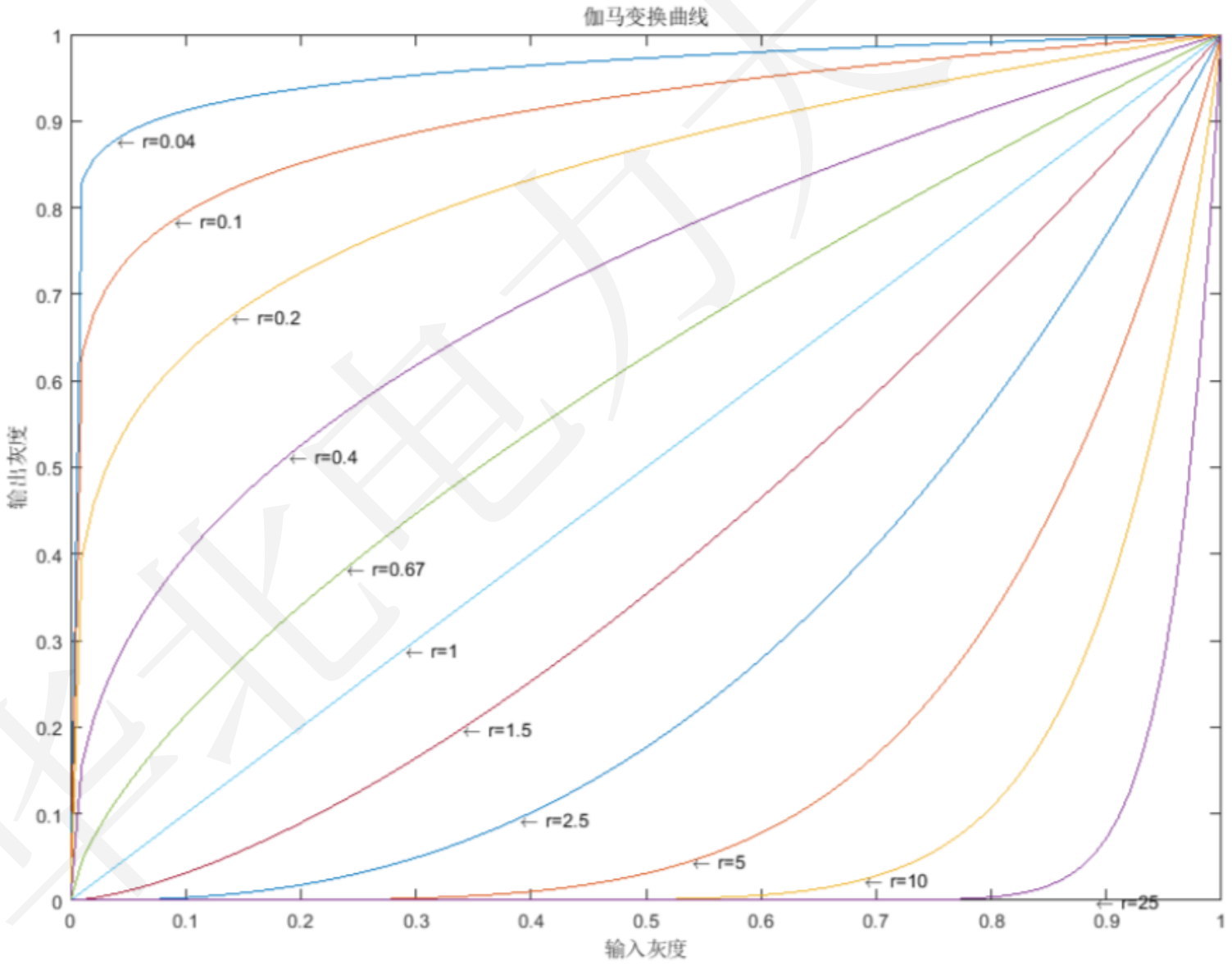
2.1.4 伽马变换

- 伽马变换又称指数变换（幂次变换），是一种非线性变换。

$$g(x,y)=(f(x,y)+\varepsilon)^{\gamma}$$

✓ ε为补偿系数，γ就是伽马系数

2.1.4 伽马变换 $g(x,y)=(f(x,y)+\varepsilon)^{\gamma}$



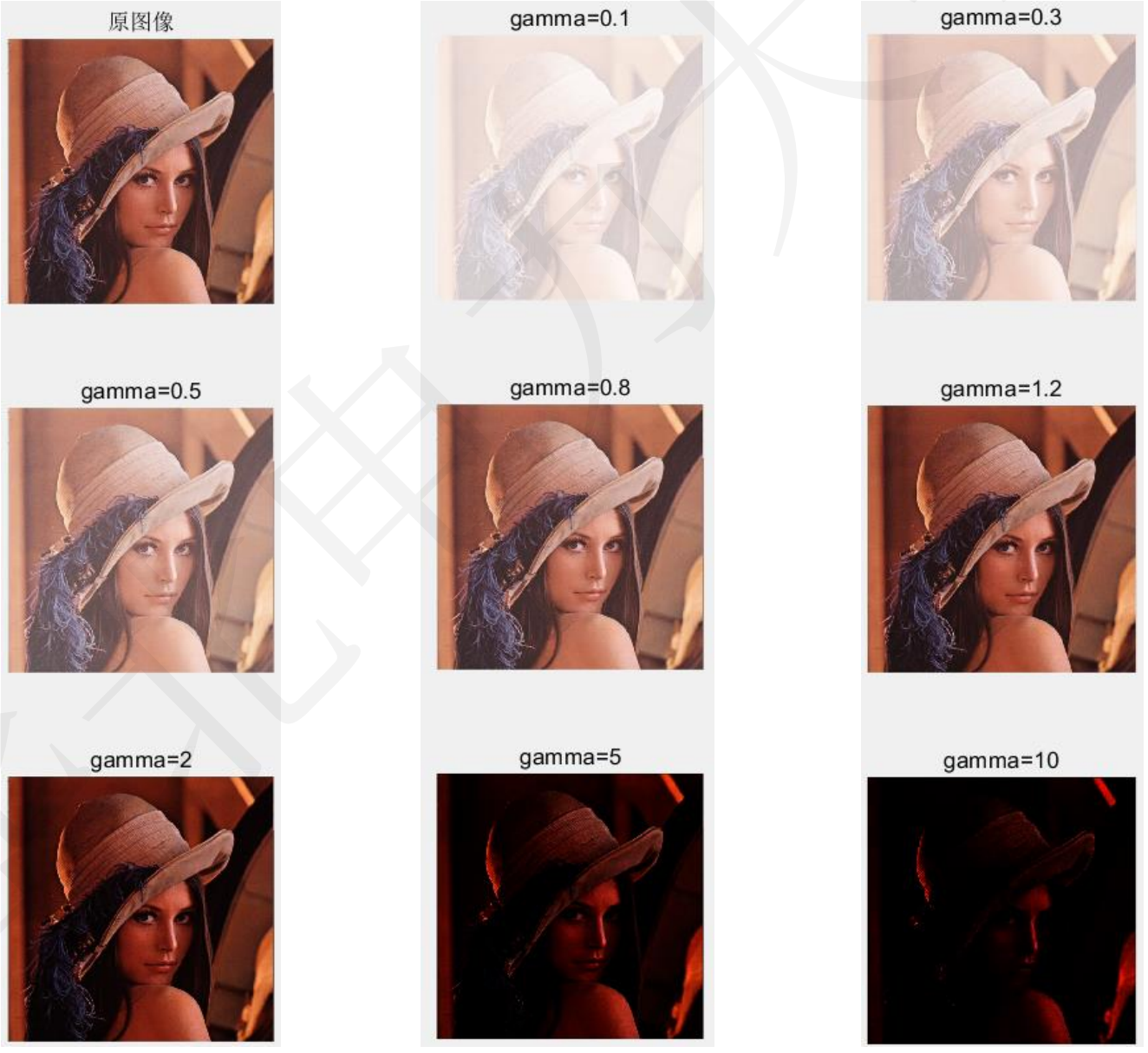
2.1.4 伽马变换 $g(x,y)=(f(x,y)+\varepsilon)^{\gamma}$

✓ γ>1 时，幂函数曲线的斜率是递增的，此时对于灰度值高的区域，伽马变换会增加该区域的灰度。

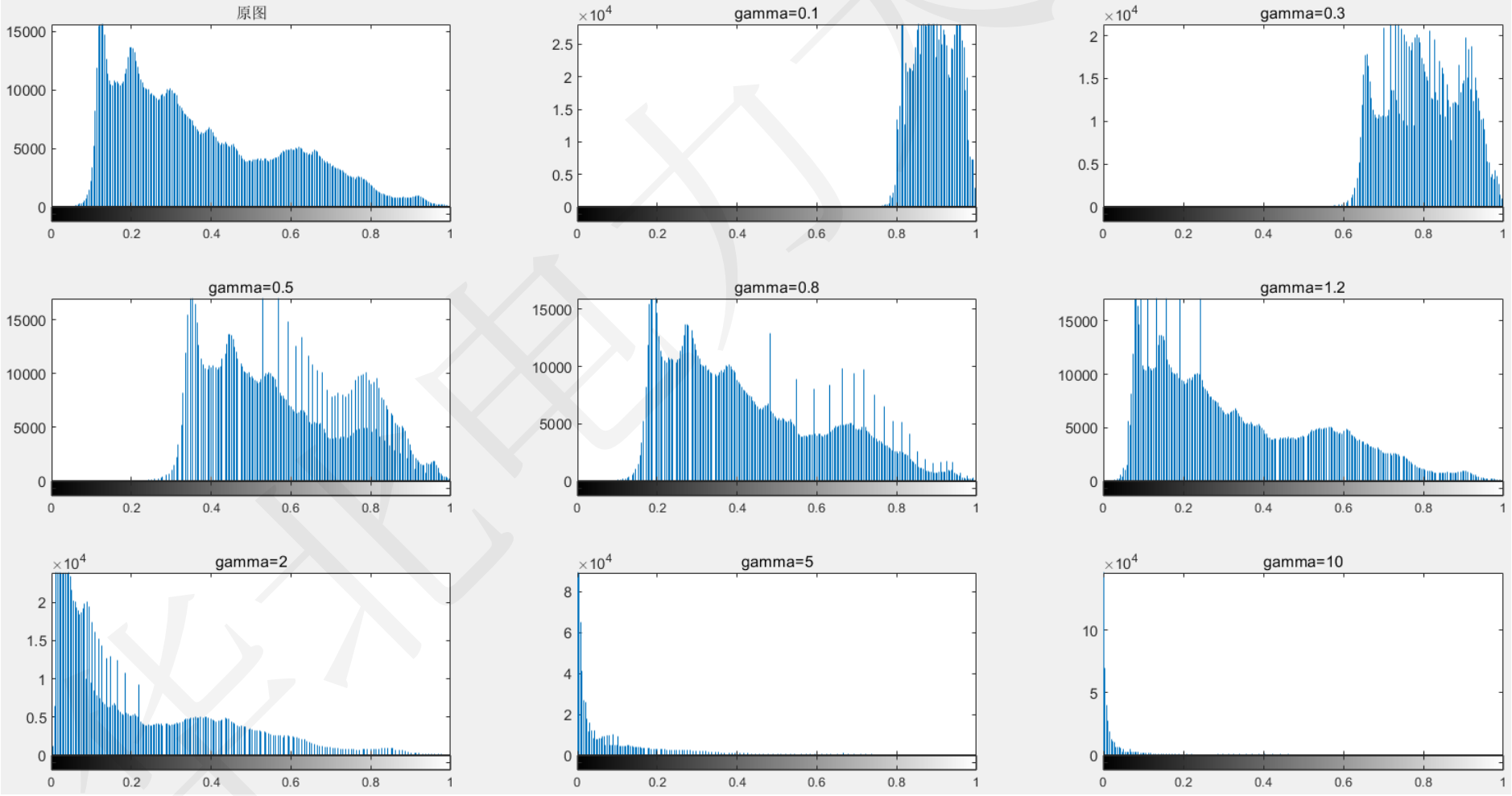
✓ γ＜1时，幂函数曲线的斜率是随灰度增加递减的，灰度值越小，且γ越小，则对于低灰度区域增强效果越大。

✓ 伽马变换可以增强想要增加的灰度区域的灰度，对一些局部区域较暗的图片，可以选择较小的γ。

2.1.4 伽马变换 $g(x,y)=(f(x,y)+\varepsilon)^{\gamma}$



2.1.4 伽马变换 $g(x,y)=(f(x,y)+\varepsilon)^{\gamma}$



2.1.4 伽马变换

```
I=imread('lena.jpg');
I = im2double(I);
%创建图形窗口
figure(1);
subplot(1,2,1)
imshow(I);
title('原图像');
%伽马变换
gamma = 0.1;
I1 = I.^gamma;
subplot(1,2,2);
imshow(I1);
title('gamma=0.1');
```

2.1.5 灰度阈值变换

- 主要应用为二值化，可以有效节省内存，提升计算速度。二值化的最重要的部分就是二值化阈值的选择，这也是称为灰度阈值变换的原因。

$$g(x,y)=\begin{cases}0 & f(x,y)\le \mathrm{T}\\ 255 & f(x,y)>\mathrm{T}\end{cases}$$

✓ T为阈值

2.1.5 灰度阈值变换

$$g(x,y)=\begin{cases}0 & f(x,y)\le \mathrm{T}\\ 255 & f(x,y)>\mathrm{T}\end{cases}$$

2.1.5 灰度阈值变换

```
I=imread('face.jpg');
I = im2double(I);
%创建图形窗口
figure(1);
subplot(1,2,1)
imshow(I);
title('原图像');
%计算阈值
thresh = graythresh(I);
I1 = imbinarize(I,thresh); %得到二值图
subplot(1,3,2);
imshow(I1);
msg = num2str(thresh);
t=strcat('自适应阈值', msg);
title(t);
```

2.1.6 直方图均衡化

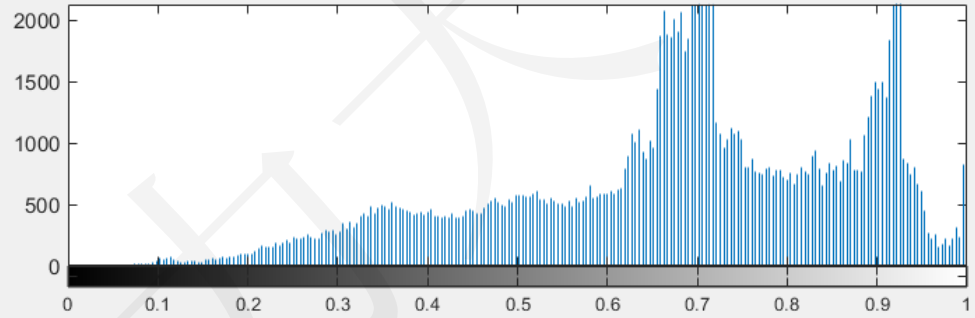
- 把原始图像的灰度直方图从比较集中的某个灰度区间变成在全部灰度范围内的均匀分布。
- 直方图均衡化就是对图像进行非线性拉伸，重新分配图像像素值，使一定灰度范围内的像素数量大致相同。

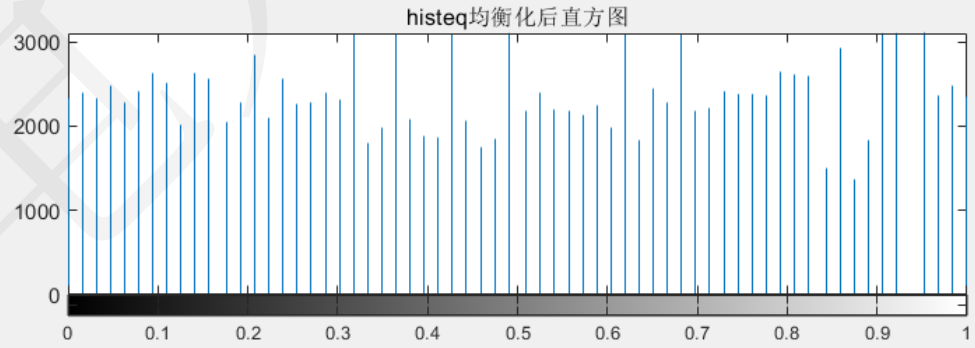
2.1.6 直方图均衡化

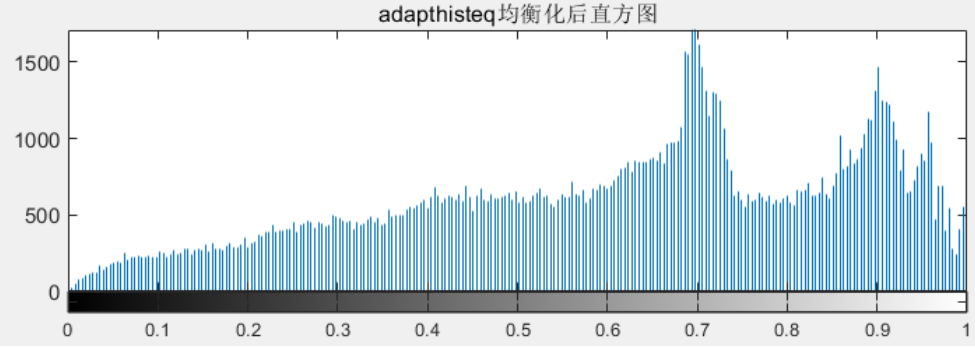
原图像

histeq均衡化

adapthisteq均衡化







2.1.6 直方图均衡化

```
%直方图均衡化
I1 = histeq(I); %使输出图像的直方图近似匹配指定的直方图（均匀分布）
subplot(3,2,3);
imshow(I1);
title('histeq均衡化');
%直方图均衡化
I2 = adapthisteq(I); %它在较小的数据区域（图块）而不是整个图像上运行，增强每个图块的对比度，以使每个输出区域的直方图近似匹配指定的直方图。
subplot(3,2,5);
imshow(I2);
title('adapthisteq均衡化');
```

2.2 代数运算

- 图像代数运算（arithmetic operation）是指对两幅或两幅以上的输入图像的对应像元素逐个地进行加、减、乘、除的四则运算，以产生有增强效果的图像。

$$g(x,y) = f(x,y) \otimes h(x,y)$$

✓$\otimes$ 是代表代数运算

2.2.1 加法

- 图像相加一般用于对同一场景的多幅图像求平均效果，以便有效地降低具有叠加性质的随机噪声。
- 加法运算的两幅图像的大小要一致。

2.2.1 加法

```
I1 = imread('p1.jpg');
I1 = im2double(I1);%转换为double
I2 = imread('p2.jpg');
I2 = im2double(I2);%转换为double
I3 = imadd(I1, I2); %像素做加法，超出范围设定为最大值

%创建图形窗口
figure(1);
subplot(1,3,1);imshow(I1);title('图像1');
subplot(1,3,2);imshow(I2);title('图像2');
%加法
subplot(1,3,3);imshow(I3);title('叠加图像');
```

2.2.1 加法

图像1

图像2

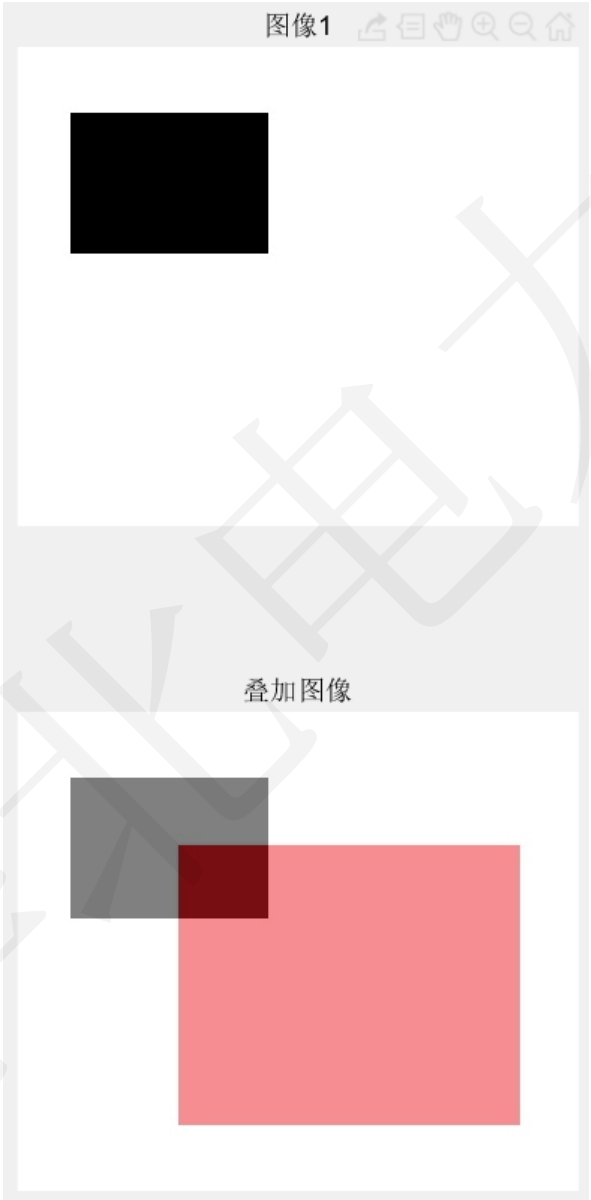
叠加图像

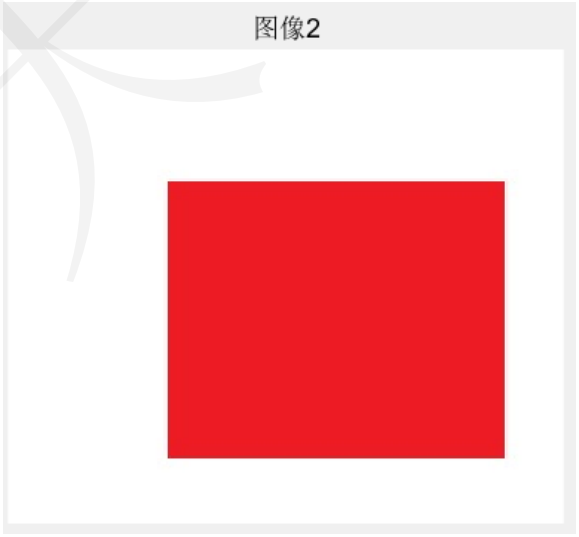
2.2.1 加法

```
I1 = imread('p1.jpg');
I1 = im2double(I1);%转换为double
I2 = imread('p2.jpg');
I2 = im2double(I2);%转换为double
I3 = I1*0.5 + I2*0.5; %融合效果

%创建图形窗口
figure(1);
subplot(1,3,1);imshow(I1);title('图像1');
subplot(1,3,2);imshow(I2);title('图像2');
%加法
subplot(1,3,3);imshow(I3);title('叠加图像');
```

2.2.1 加法





2.2.1 加法

图像1

图像2

I1*0.5+I2*0.5

叠加图像

I1+I2

叠加图像

2.2.2 减法

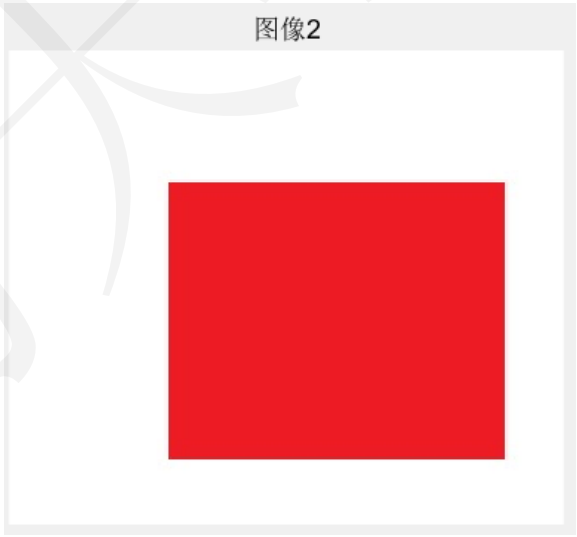
- 图像减法也称为差分方法，是一种常用于检测图像变化及运动物体的图像处理方法。
- 减法运算的两幅图像的大小要一致。

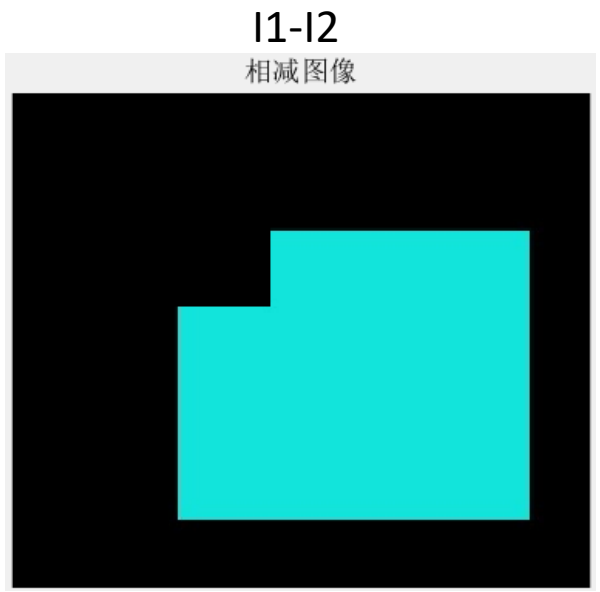
2.2.2 减法

```
I1 = imread('p1.jpg');
I1 = im2double(I1);%转换为double
I2 = imread('p2.jpg');
I2 = im2double(I2);%转换为double
I3 = imsubtract(I2, I1); %像素做减法，超出范围设定为最小值

%创建图形窗口
figure(1);
subplot(1,3,1);imshow(I1);title('图像1');
subplot(1,3,2);imshow(I2);title('图像2');
%减法
subplot(1,3,3);imshow(I3);title('相减图像');
```

2.2.2 减法





2.2.2 减法

图像1

图像2

相减图像

2.2.3 乘法

- 两幅图像进行乘法运算可以实现掩模操作，即屏蔽掉图像的某些部分。
- 一幅图像乘以一个常数通常被称为缩放，这是一种常见的图像处理操作。
- 如果使用的缩放因子大于1，那么将增强图像的亮度，如果因子小于1则会使图像变暗。

2.2.3 乘法

```
I1 = imread('p1.jpg');
I1 = im2double(I1);%转换为double
I2 = 1 - I1; %黑白互换
I3 = imread('p2.jpg');
I3 = im2double(I3);%转换为double
I4 = immultiply(I1,I3);
I5 = immultiply(I2,I3);
%创建图形窗口
figure(1);
subplot(2,2,1);imshow(I1);title('图像1');
subplot(2,2,2);imshow(I3);title('图像2');
%乘法
subplot(2,2,3);imshow(I4);title('相乘图像');
subplot(2,2,4);imshow(I5);title('取反相乘图像');
```

2.2.3 乘法

图像1

图像2

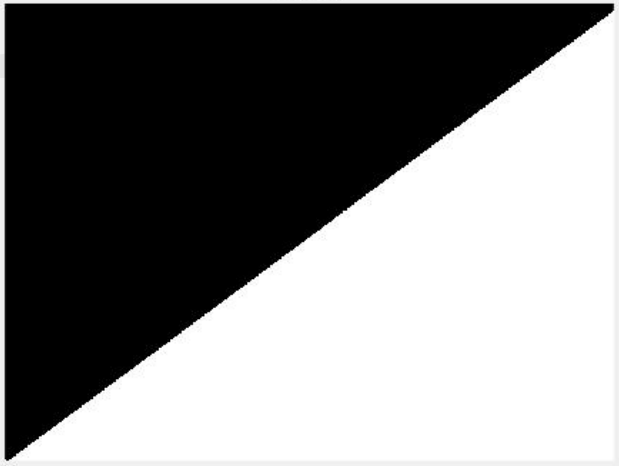
相乘图像

取反相乘图像

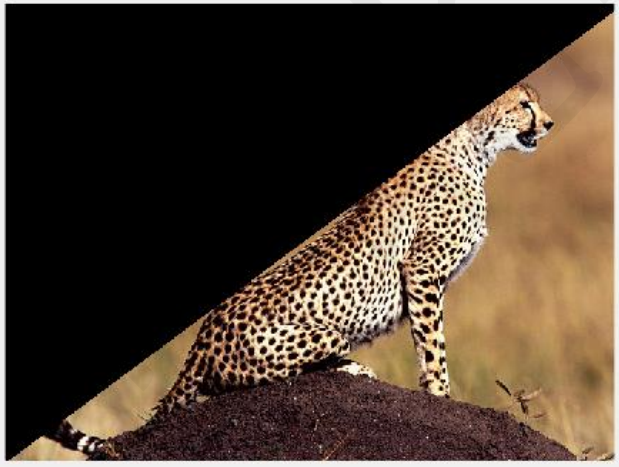
2.2.3 乘法

图像1

图像2



相乘图像



2.2.4 除法

- 除法运算可用于校正成像设备的非线性影响，这在特殊形态的图像（如断层扫描等医学图像）处理中常常用到。
- 图像除法也可以用来检测两幅图像间的区别，但是除法操作给出的是相应像素值的变化比率，而不是每个像素的绝对差异。

2.2.4 除法

```
I1 = imread('p1.jpg');
I1 = im2double(I1);%转换为double
I2 = imread('p2.jpg');
I2 = im2double(I2);%转换为double
I3 = imdivide(I1, I2);
I4 = imdivide(I2, I1);

%创建图形窗口
figure(1);
subplot(2,2,1);imshow(I1);title('图像1');
subplot(2,2,2);imshow(I2);title('图像2');
%除法
subplot(2,2,3);imshow(I3);title('I1/I2相除图像');
subplot(2,2,4);imshow(I4);title('I2/I1相除图像');
```

2.2.4 除法

图像1

图像2

I1/I2相除图像

I2/I1相除图像