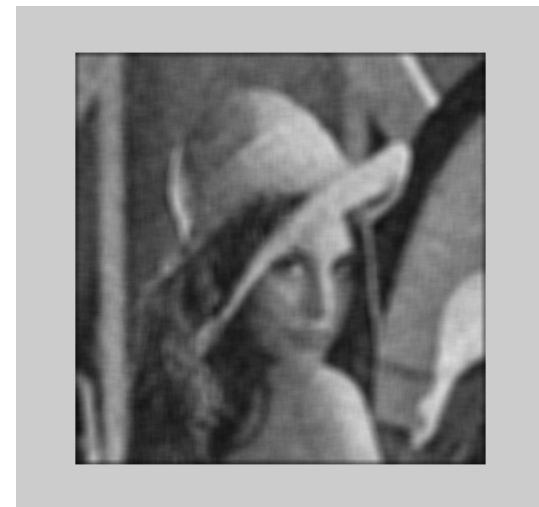
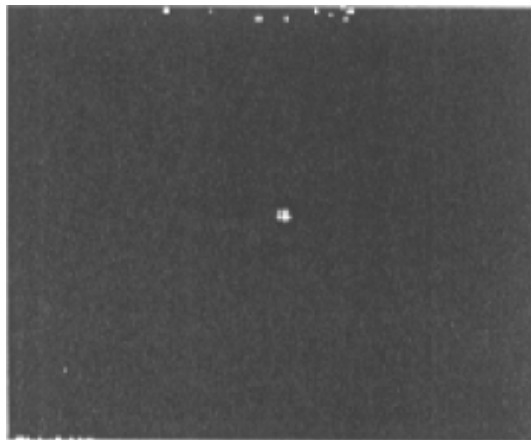


## 第三章 图像增强

- ◆ 背景知识
- ◆ 基本灰度变换
- ◆ 直方图处理
- ◆ 用算术/逻辑操作增强
- ◆ 空间滤波基础
- ◆ 平滑空间滤波器
- ◆ 锐化空间滤波器

## 一、背景知识



图像增强的首要目标是处理图像，使其比原始图像更适用于**特定**应用。

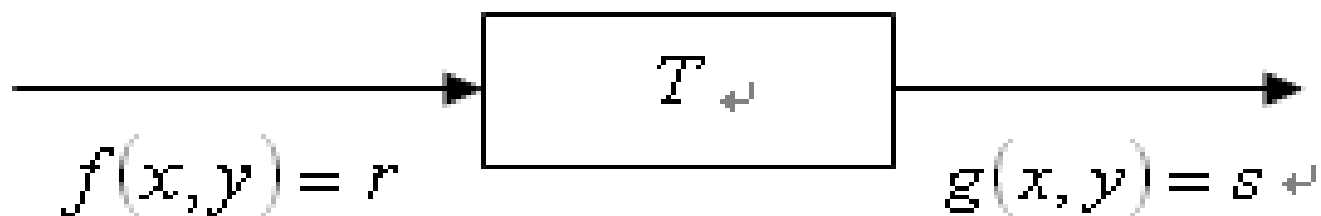
图像增强的方法分为：**空间域方法**和频域方法。  
空间域方法以对图像的**像素直接处理**为基础，频域处理技术以**修改图像的傅立叶变换**为基础。

# 空间域图像增强技术

- 在空间域中，通过线性或非线性变换来增强构成图像的像素。
- 增强的方法主要分为点处理和模板处理两大类
- **点处理**是作用于单个像素的空间域处理方法，包括图像灰度变换、直方图处理、伪彩色处理等技术；
- **模板处理**是作用于像素邻域的处理方法，包括空域平滑、空域锐化等技术。

“空间域增强”是指增强构成图像的像素。空间域方法是直接对这些像素操作的过程。

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



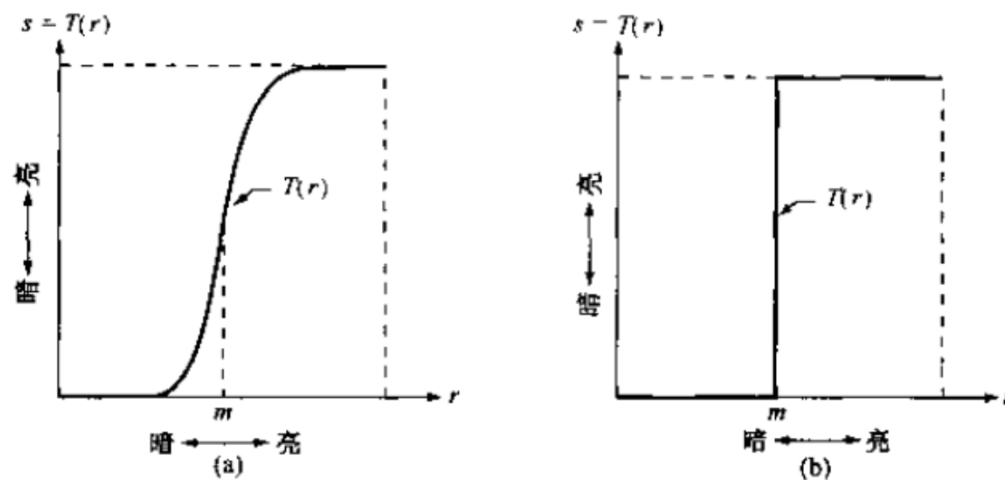
$f(x, y)$ 是待增强图像， $g(x, y)$ 是增强后的图像， $T$ 是对 $f$ 的一种操作，其定义在 $(x, y)$ 的邻域。

定义一个点 $(x, y)$ 邻域的主要方法是利用中心在 $(x, y)$ 点的正方形或矩形子图像。

$T$ 操作最简单的形式是邻域为 $1 \times 1$ 的尺度（单个像素）。在这种情况下， $g$ 仅依赖于 $f$ 在 $(x,y)$ 点的值， $T$ 操作成为灰度级变换函数（强度映射，点处理）。

$$s = T(r)$$

$r$ 和 $s$ 所定义的变量，分别是 $f(x,y)$ 和 $g(x,y)$ 在任意点 $(x,y)$ 的灰度级。



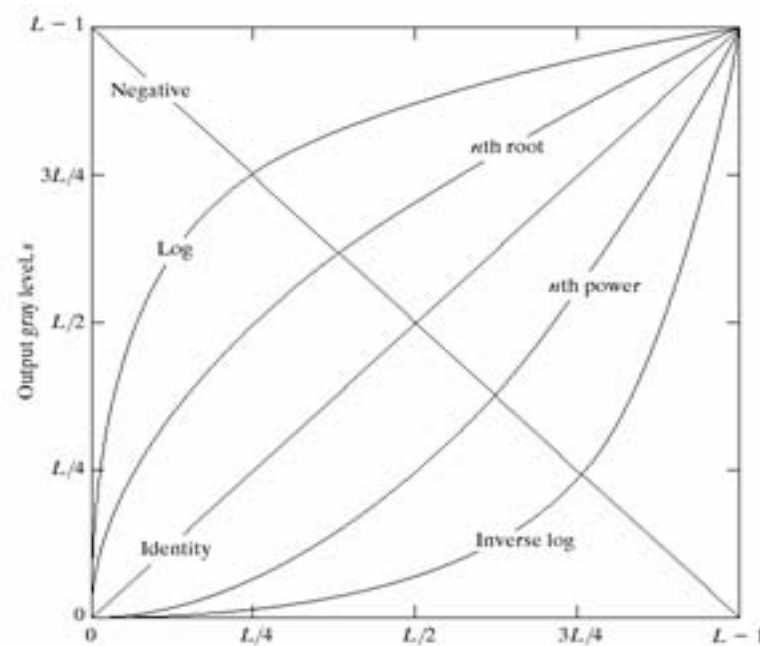
更大的邻域会有更多的灵活性。一般方法是，利用点 $(x,y)$ 事先定义的邻域里的一个 $f$ 值的函数来决定 $g$ 在 $(x,y)$ 的值，其公式化的一个主要方法是以利用所谓的模板(滤波器、核、掩膜或窗口)为基础的。

模板是一个小的(即 $3 \times 3$ )二维阵列，模板的系数值决定了处理的性能。

## 二、某些基本灰度变换

图像增强的三个基本类型函数：

**线性的**（正比和反比）、**对数的**（对数和反对数变换）、**幂次的**（**n**次幂和**n**次方根变换）

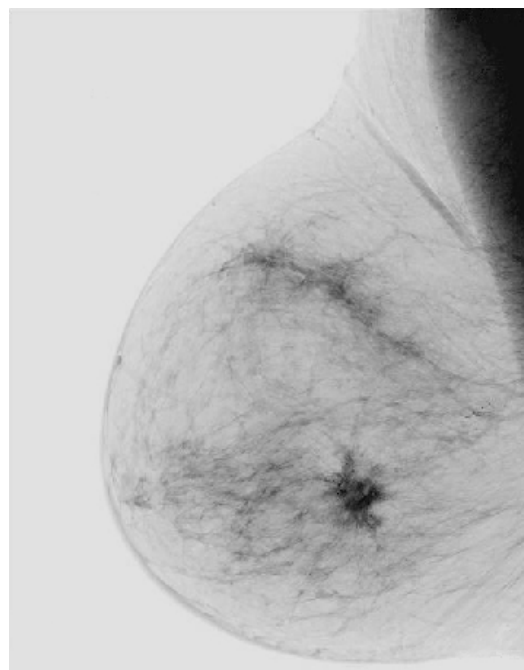
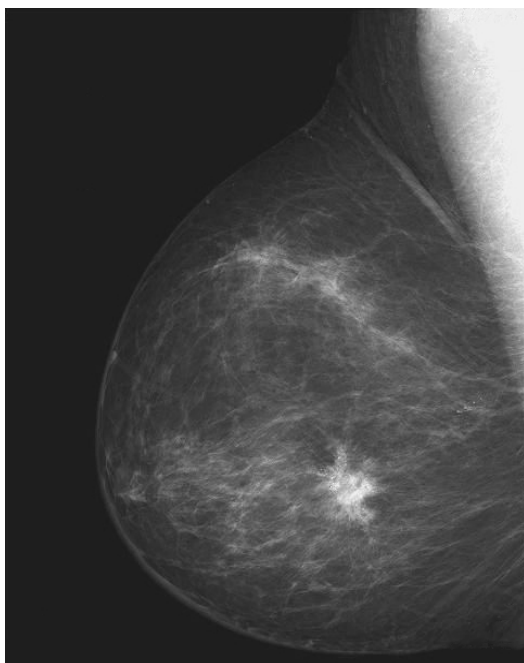




## 图像反转

灰度级范围为 $[0, L-1]$ 的图像反转可定义为:

$$s = L - 1 - r$$



- 用这种方式倒转图像的强度，可以产生图像反转的对等图像。
- 反转变换适用于增强嵌入于图像暗色区域的白色或灰色细节，特别是当黑色面积占主导地位时。

# 对数变换

$$s=c \log(1+r)$$

$$g(i,j) = a + \frac{\ln[f(i,j) + 1]}{b \cdot \ln c}$$

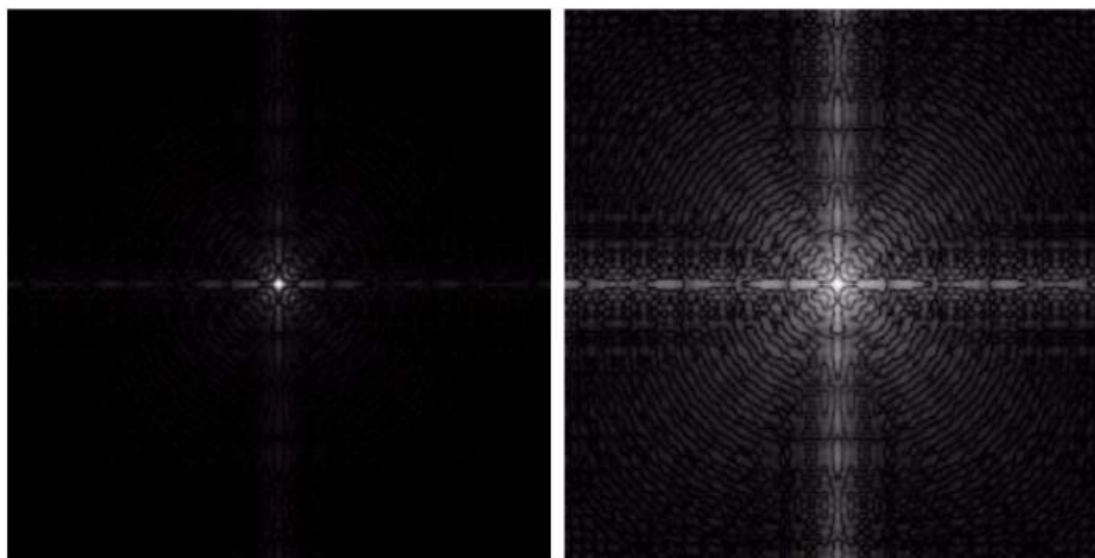
$a, b, c$ 是为了调整曲线的形状和位置而引入的参数。

对数变换使一窄带低灰度输入图像值映射为一宽带输出值。当希望对图像的低灰度区进行较大的拉伸而对高灰度区压缩时，可以采用这种变换，它能使图像灰度分布与人的视觉特性相匹配。

一般对数函数的所有曲线都能完成图像灰度的扩散/压缩。

傅立叶频率的像素值有很大的动态范围，当计算机处理像这样的无误差数字时，图像显示系统通常不能够如实再现如此大范围的强度值。

0-1.5\*10<sup>6</sup>的傅里叶频谱



对数变换的图像（显示在一个**8bit**的系统中）

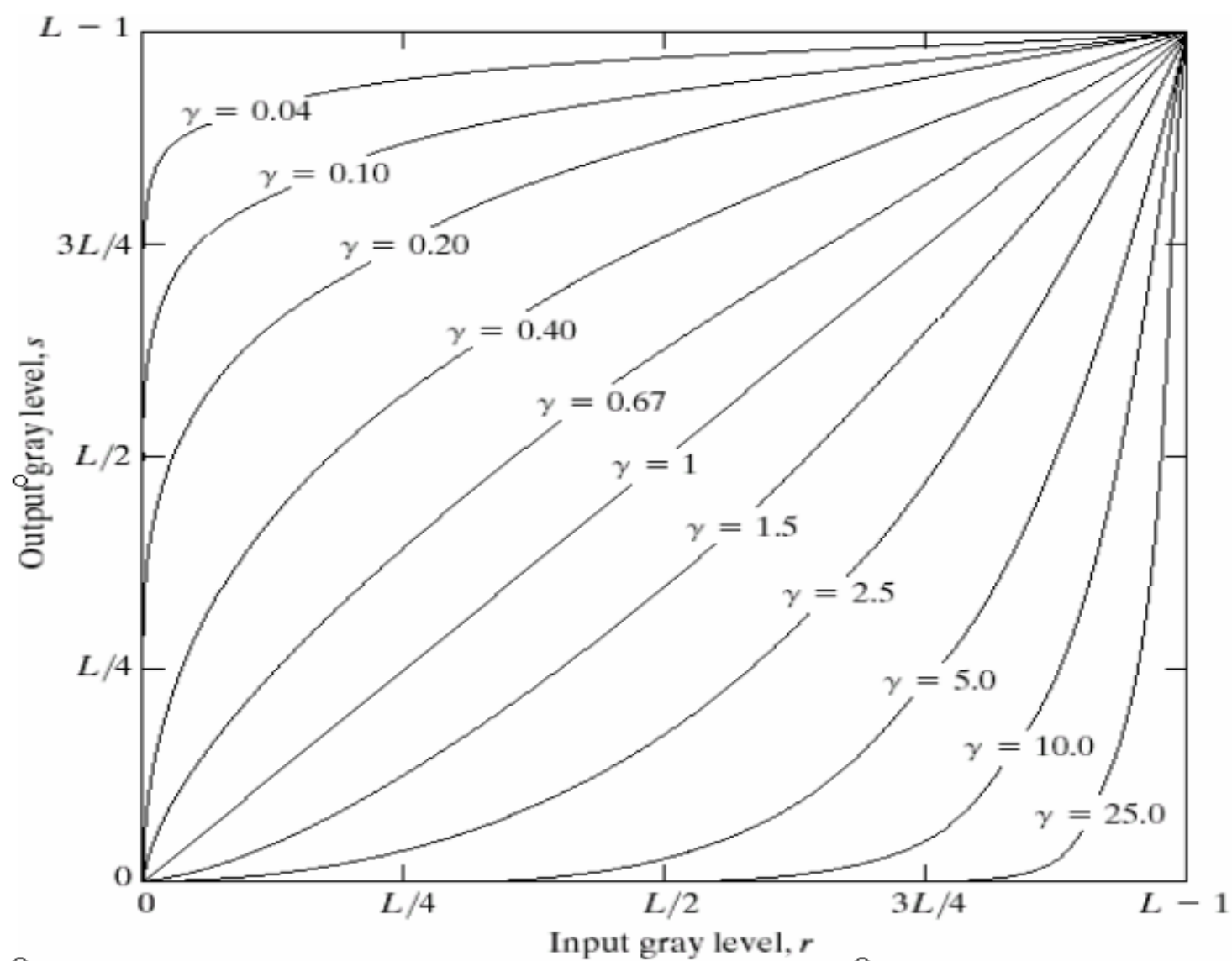
## 幂次变换

$$s = cr^Y$$

$Y$ 的部分值把输入窄带暗值映射到宽带输出值，相反，输入高值时也成立。

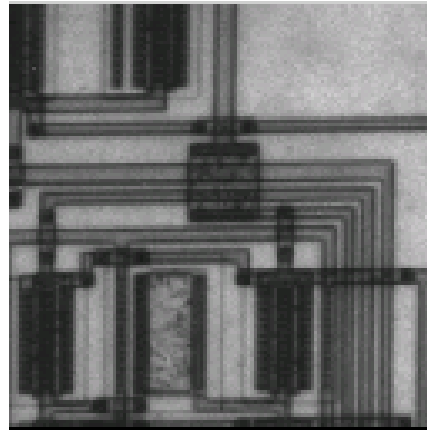
当  $\gamma > 1$  时，把输入的窄带暗值映射到宽带输出亮值；

当  $\gamma < 1$  时，把输入高值映射为宽带。





$\gamma=1.5$



$\gamma=0.66$

## 幂次变换的应用 ~~伽~~伽马校正

- 为什么要进行伽马校正？

几乎所有的CRT显示设备、摄像胶片、许多电子照相机的光电转换特性都是非线性的。所以，如果不进行校正处理的话，将无法得到好的图像效果。



- 实际中  $\gamma$  值的确定方法

通常CCD的  $\gamma$  值在0.4 ~ 0.8之间， $\gamma$  值越小，画面的效果越差。根据画面对比度的观察与分析，可以大致得到该设备的  $\gamma$  值（或依据设备的参考  $\gamma$  值）。

# 分段线性变换

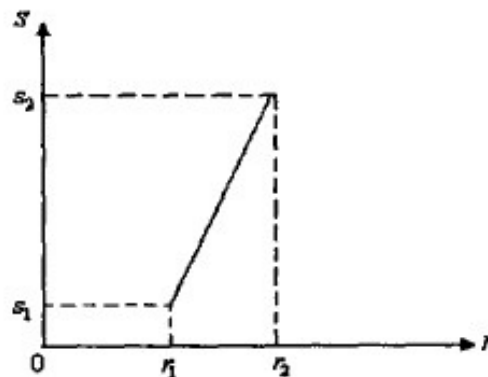
## 1. 对比拉伸

最简单的分段线性函数之一是对比拉伸变换。低对比度图像可由照明不足、成像传感器动态范围大小，甚至在图像获取过程中透镜光圈设置错误引起。对比拉伸的思想是提高图像处理时灰度级的动态范围。

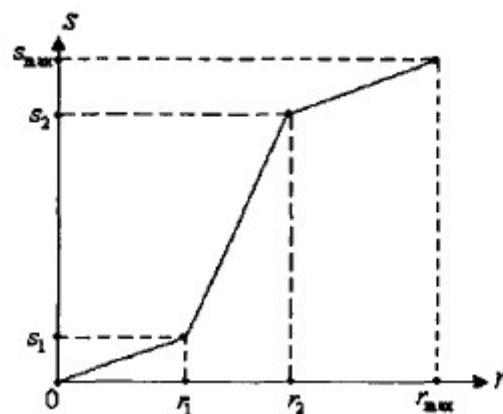
## 分段线性变换

令 $r$ 和 $s$ 分别为 $f(x,y)$ 和 $g(x,y)$ 在任意点 $(x,y)$ 的灰度级。假设原图像 $f(x,y)$ 的灰度范围为 $[r_1, r_2]$ ，如果希望变换后的图像 $g(x,y)$ 的灰度范围拓展至 $[s_1, s_2]$ ，则线性变换表示式为：

$$s = \frac{s_2 - s_1}{r_2 - r_1} \times (r - r_1) + s_1$$



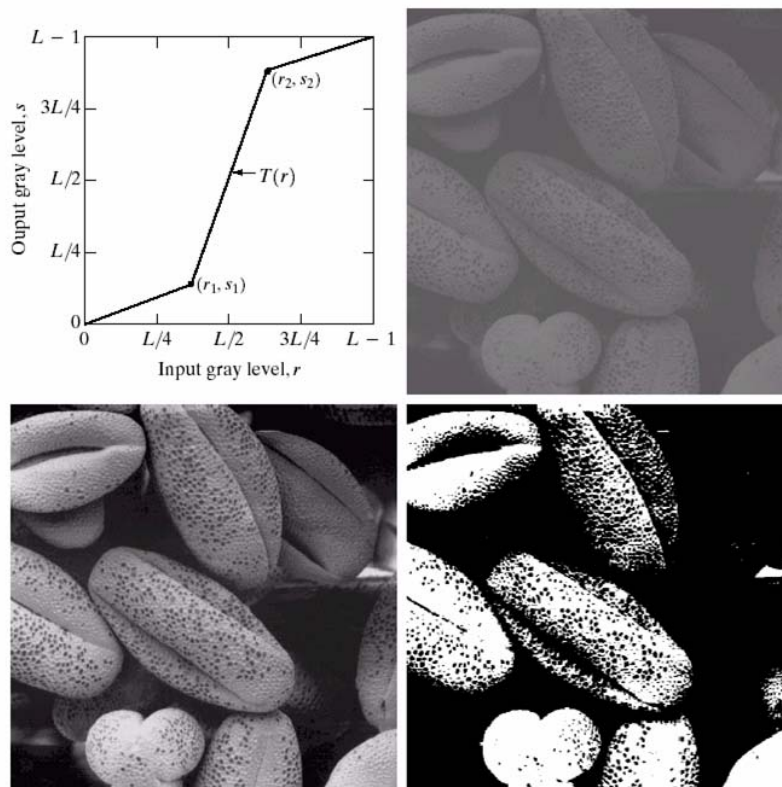
为了突出图像中感兴趣的目标或者灰度区间，相对抑制那些不感兴趣的灰度区间（如背景区域），而不惜牺牲其它灰度级上的细节信息，我们可以采用分段线性变换法，将需要的图像细节灰度级拉伸，增强其对比度；而将不需要的细节灰度级压缩。最常用的是三段线性变换法。通过调整折线拐点的位置及控制分段直线的斜率，可对任意灰度区间进行拉伸或压缩。



$$s = \frac{s_1}{r_1} \cdot r \quad 0 \leq r \leq r_1$$

$$s = \frac{s_2 - s_1}{r_2 - r_1} \cdot (r - r_1) + s_1 \quad r_1 \leq r \leq r_2$$

$$s = \frac{s_{\max} - s_2}{r_{\max} - r_2} \cdot (r - r_2) + s_2 \quad r_2 \leq r \leq r_{\max}$$



(a) 变换函数的形式

(b) 低对比度图像

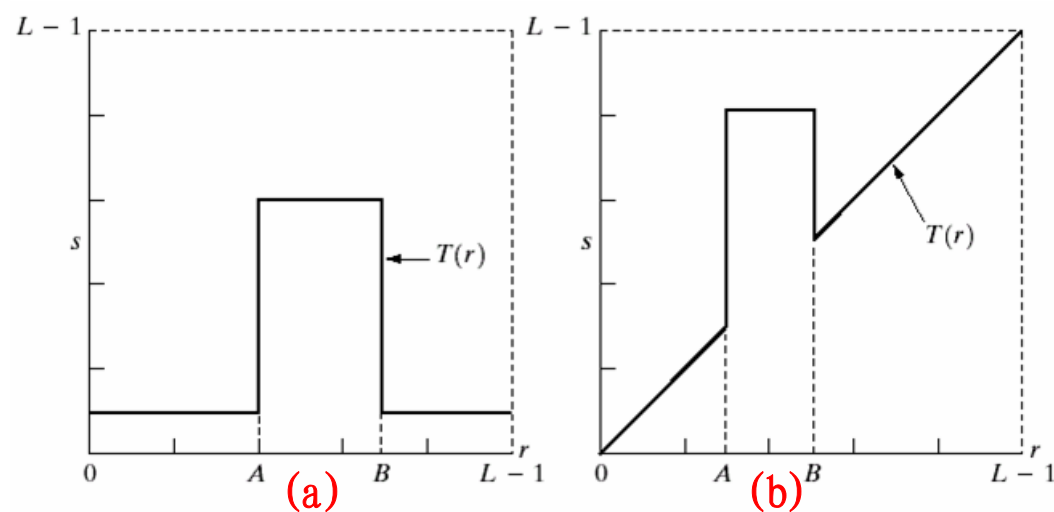
(c) 对比度拉伸的结果

(d) 门限化的结果

$(r_1, s_1) = (r_{\min}, 0)$  且  $(r_2, s_2) = (r_{\max}, L-1)$ ;

如果 $r_1=s_1$ 且 $r_2=s_2$ , 变换函数为一线性函数, 它产生一个没有变化的灰度级。若 $r_1=r_2, s_1=0, s_2=L-1$ , 变换为阈值函数, 并产生二值图像。

## 2、灰度切割：提高特定灰度范围的亮度

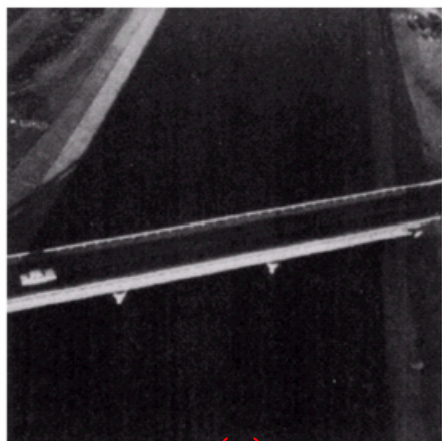


(a)加亮 $[A, B]$ 范围,其他  
灰度减小为一恒定值

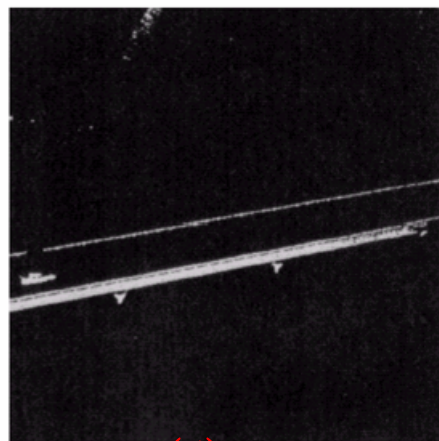
(b)加亮 $[A, B]$ 范围,其他  
灰度级不变

(c)原图像

(d)使用(a)变换的结果



(c)

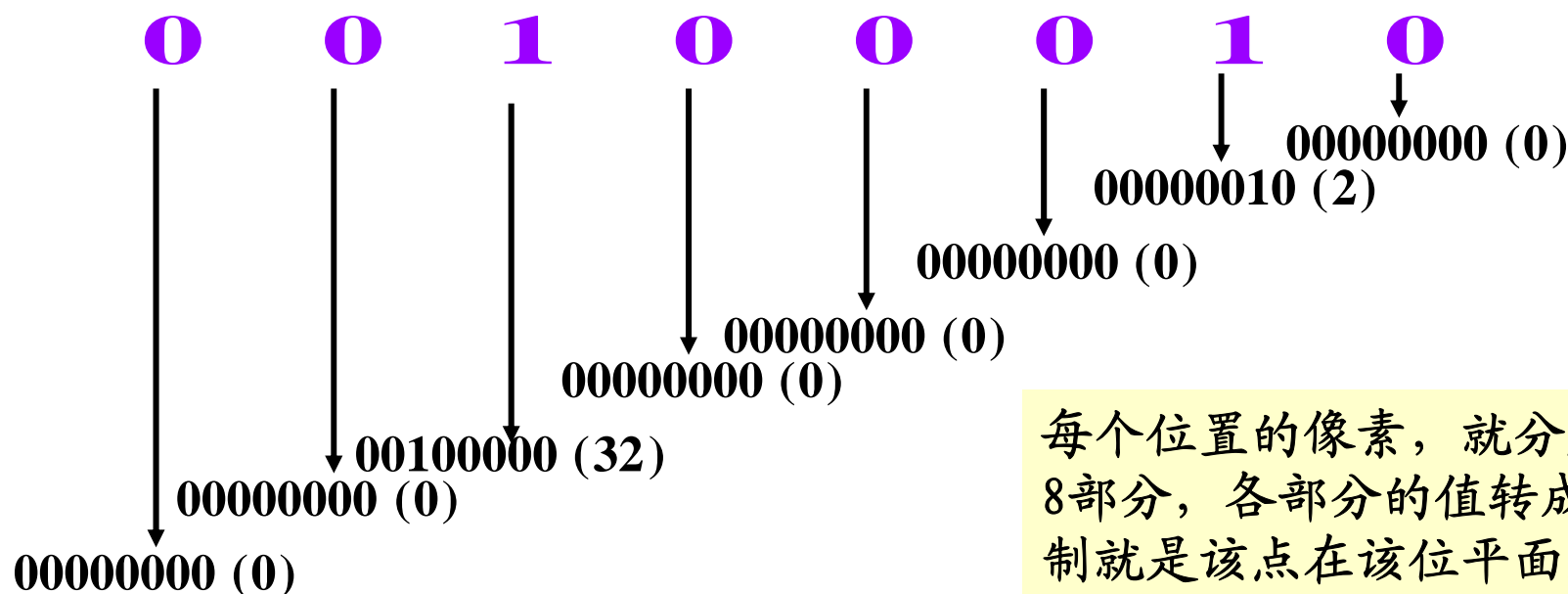


(d)

## 3、位图切割

把数字图像分解成为位平面，（**每一个位平面可以处理为一幅二值图像**）对于分析每一位在图像中的相对重要性是有用的。（高阶位如前4位包含视觉上很重要的大多数数据；其它位对图像中的更多微小细节有作用）

例如每个象素点的灰度值用8bit表示，假如某像素点的灰度值为00100010，分解处理如下：

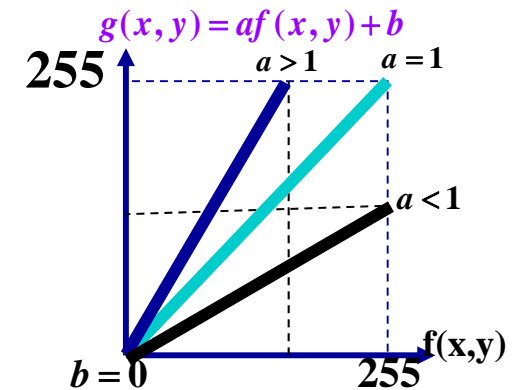


每个位置的像素，就分解成了8部分，各部分的值转成十进制就是该点在该位平面上的灰度值。



```
I=imread('pout.tif');  
pout=double(I);  
A=0.5; B=50;  
pout2=pout*A+B;  
A=1.5; B=50;  
pout3=pout*A+B;
```

```
J1=uint8(pout2);  
J2=uint8(pout3);  
subplot(1,3,1),imshow(I);  
subplot(1,3,2),imshow(J1);  
subplot(1,3,3),imshow(J2);
```



$f(x, y)$

$g(x, y) = 0.5f(x, y) + 50$

$g(x, y) = 1.5f(x, y) + 50$



## Matlab灰度变换函数

功能：通过灰度变换调整对比度

格式： **J=imadjust(I,[low high],[bottom top],gamma)**

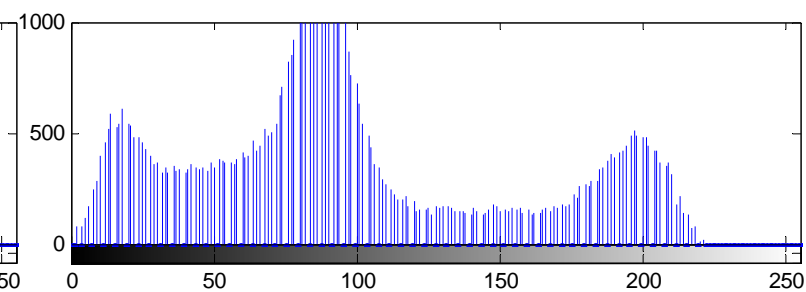
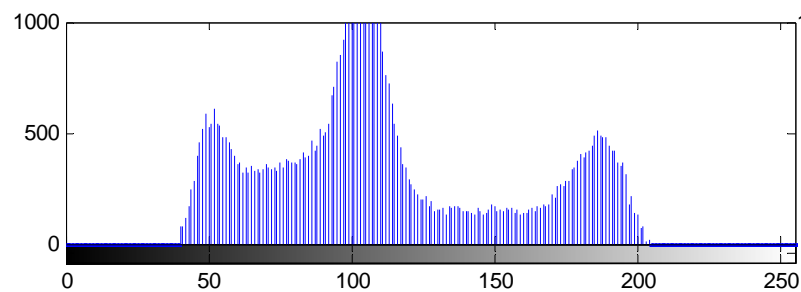
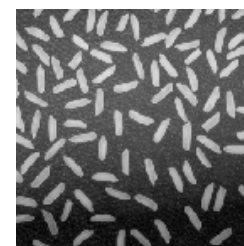
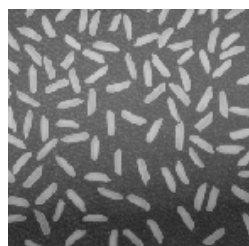
将图像I中的灰度值映射到J中的新值，即将灰度在[low high]之间的值映射到[bottom top]之间。

- ✓ **gamma** 为校正量 $r$ ,默认为1(线性变换)
- ✓ [low high] 为原图像中要变换的灰度范围，取值范围在[0, 1]（归一化后的灰度值），
- ✓ [bottom top]指定了变换后的灰度范围，取值范围在[0, 1]

```
out = adjustArray(img, lIn, hIn, lOut, hOut, g, expansionFactor);
```

```
img(:) = max(lIn(d,:), min(hIn(d,:),img));  
out = ( (img - lIn(d,:)) ./ (hIn(d,:) - lIn(d,:)) ) .^ (g(d,:));  
out(:) = out .* (hOut(d,:) - lOut(d,:)) + lOut(d,:);
```

```
Im=imread('rice.png');  
Jm=imadjust(Im,[0.15,0.9],[0,1]);  
figure(1);subplot(211);imshow(Im);subplot(212);imhist(Im);  
figure(2);subplot(211);imshow(Jm);subplot(212);imhist(Jm);
```



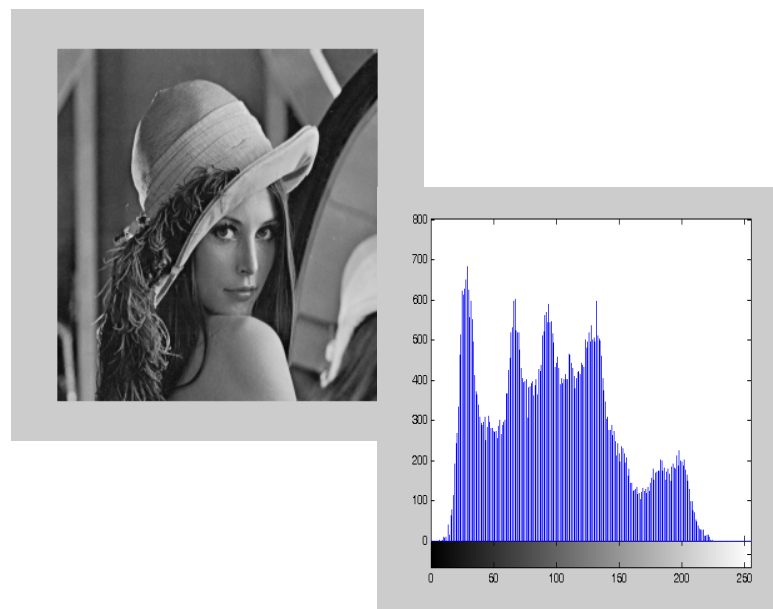
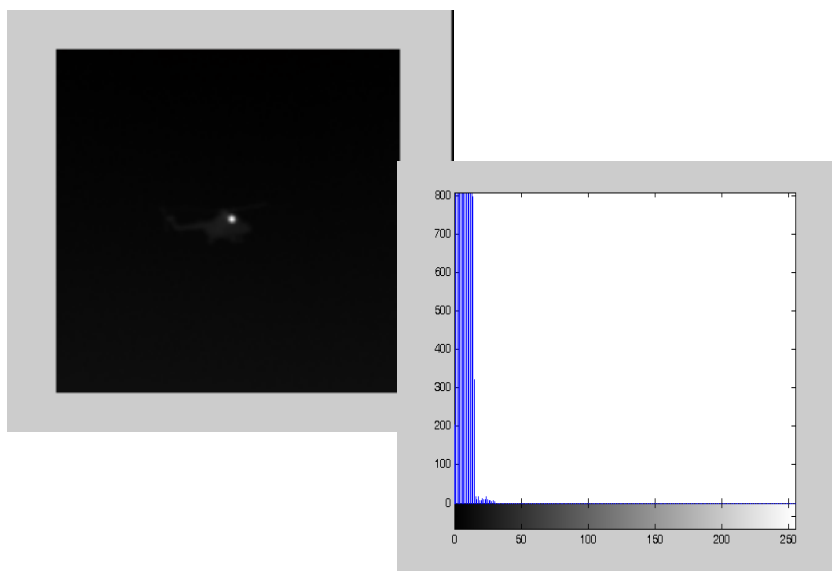
- (1) 观察图像的**直方图**，判断灰度范围
- (2) 将灰度范围转换为0.0 ~ 1.0之间的分数，使得灰度范围可以通过向量[low,high]传递给imadjust函数。
- (3) 可以利用stretchlim函数以分数向量形式返回灰度范围，直接传递给imadjust()。

```
Im=imread('rice.png');  
Jm=imadjust(Im,stretchlim(Im),[0,1]);  
figure(1);subplot(211);imshow(Im);subplot(212);imhist(Im);  
figure(2);subplot(211);imshow(Jm);subplot(212);imhist(Jm);
```

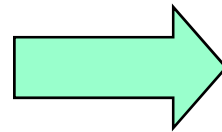
## 三、直方图处理

直方图是图像的灰度统计:  $p(s_k) = n_k / n$

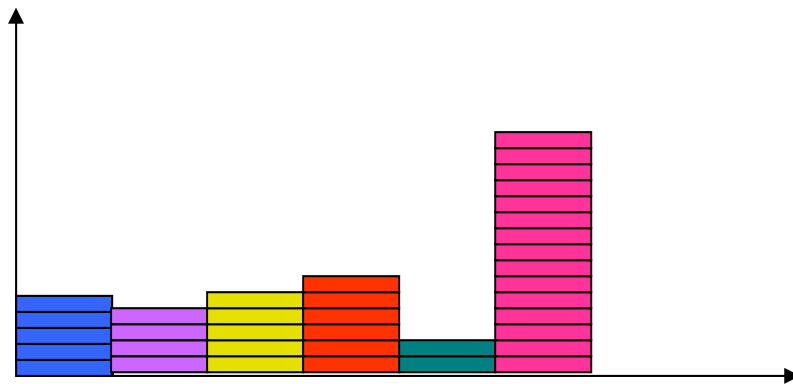
其中 $S_k$ 是图像 $f(x,y)$ 的第 $k$ 级灰度值,  $n_k$ 是 $f(x,y)$ 中具有灰度值 $S_k$ 的像素的个数,  $n$ 是图像像素总数。

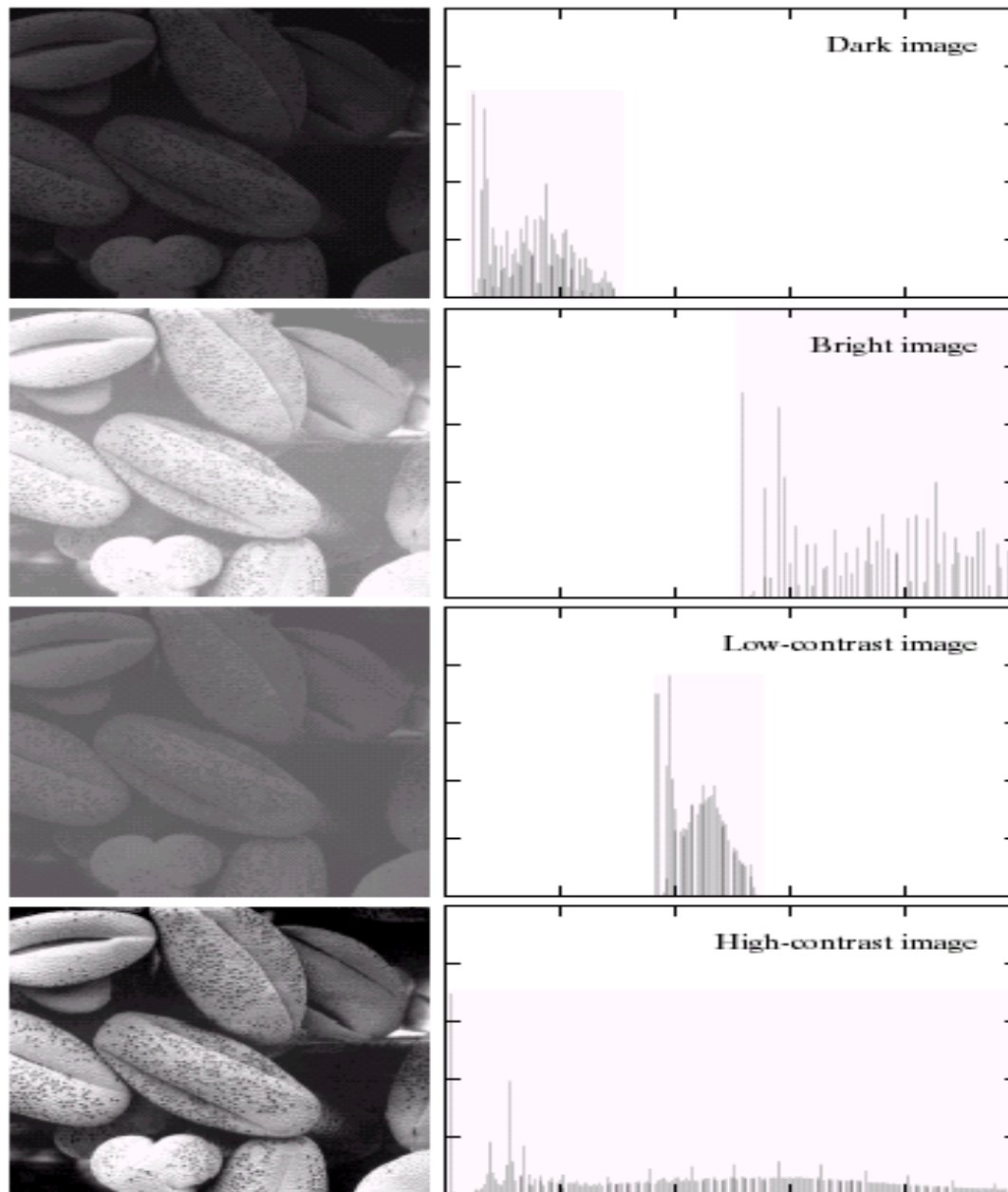


1	2	3	4	5	6
6	4	3	2	2	1
1	6	6	4	6	6
3	4	5	6	6	6
1	4	6	6	2	3
1	3	6	4	6	6



1	2	3	4	5	6
5	4	5	6	2	14

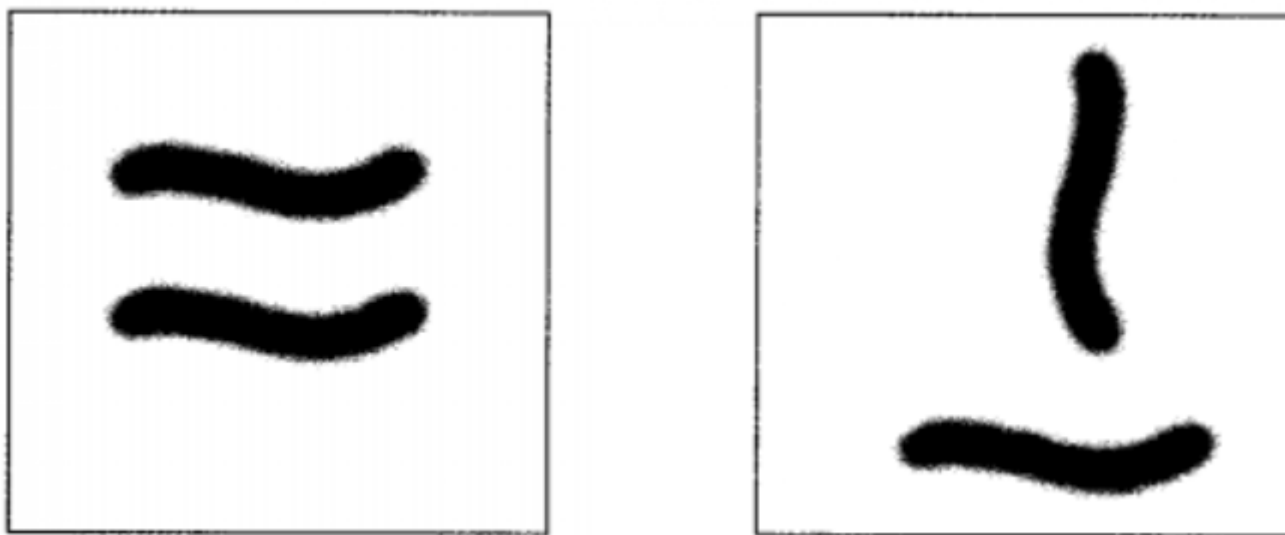






## 直方图的性质

- ◆ 直方图只反应图像的灰度分布情况，不能反应图像像素的位置。
- ◆ 一幅图像对应唯一的灰度直方图，反之不成立。



不同的图像具有相同的直方图

## 直方图均衡化处理

假设原图的灰度值变量为 $r$ , 变换后新图的灰度值变量为 $s$ , 我们希望寻找一个灰度变换函数 $T$ :  $s=T(r)$ ,

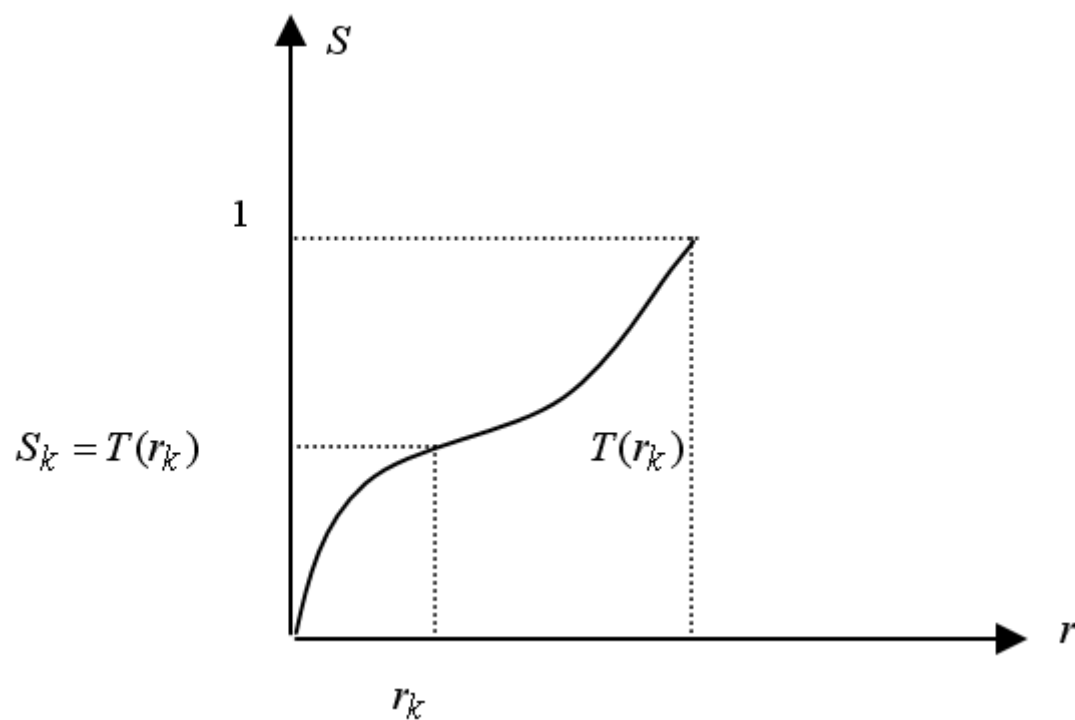
使得概率密度函数 $p_r(r)$ 变换成希望的概率密度函数 $p_s(s)$

从人眼视觉特性来考虑, 一幅图像的直方图如果是均匀分布的, 信息量最大。因此要求将原图像进行直方图均衡化, 以满足人眼视觉要求的目的。

灰度变换函数 $T(r)$ 应该满足:

- (1)  $T(r)$ 在区间 $[0, 1]$ 中单调递增且单值;
- (2)  $\forall r \in [0, 1]$ , 有  $T(r) \in [0, 1]$ ;

满足这两个条件的变换函数的一个例子。



灰度变换函数

**从  $s$  到  $r$  的反变换可用式下表示**

$$r = T^{-1}(s) \qquad 0 \leq s \leq 1$$

把原始图像的直方图变换为均匀分布的形式，这样就增加了像素灰度值的动态范围，从而达到增强图像整体对比度的效果。

(1) 计算原始图像直方图  $P_f(f_j) = \frac{n_j}{n}, \quad j = 0, 1, \dots, k, \dots, L-1$

(2) 计算累积分布函数  $c(f) = \sum_{j=0}^k P_f(f_j), \quad j = 0, 1, \dots, k, \dots, L-1$

(3) 计算映射后输出的灰度级  $g_i = \text{INT}\{(L-1)c(f) + 0.5\}$

(4) 统计映射后各灰度级的像素数目

(5) 计算输出图像直方图  $P_g(g_i) = \frac{n_i}{n}, \quad i = 0, 1, \dots, p-1$

(6) 用 $f_i$ 和 $g_i$ 的映射关系，修改原始图像的灰度级，获得直方图近似均匀分布的输出图像

对 $8 \times 8$ 的图像,灰度级 $L=8$ , 对其进行直方图均衡

4	4	4	4	4	4	4	0
4	5	5	5	5	5	4	0
4	5	6	6	6	5	4	0
4	5	6	7	6	5	4	0
4	5	6	6	6	5	4	0
4	5	5	5	5	5	4	0
4	4	4	4	4	4	4	0
4	4	4	4	4	4	4	0

$$n = 8 \times 8 = 64$$

## Matlab进行直方图均衡化的函数

功能：用柱状图均衡化增强对比

语法： **J = histeq(I)**

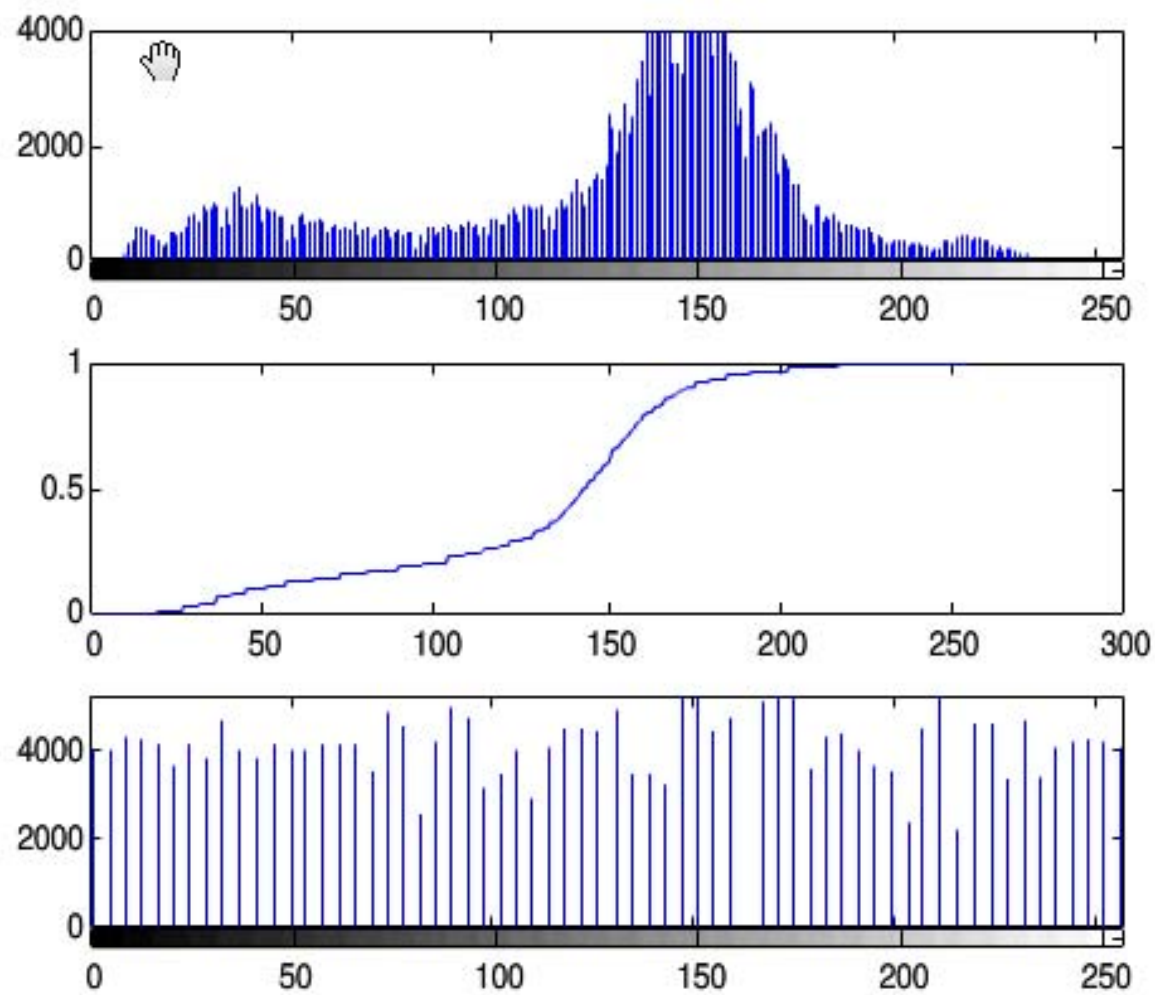
```
I = imread('pout.tif');  
figure(1); subplot(211);imshow(I);subplot(212);imhist(I);
```

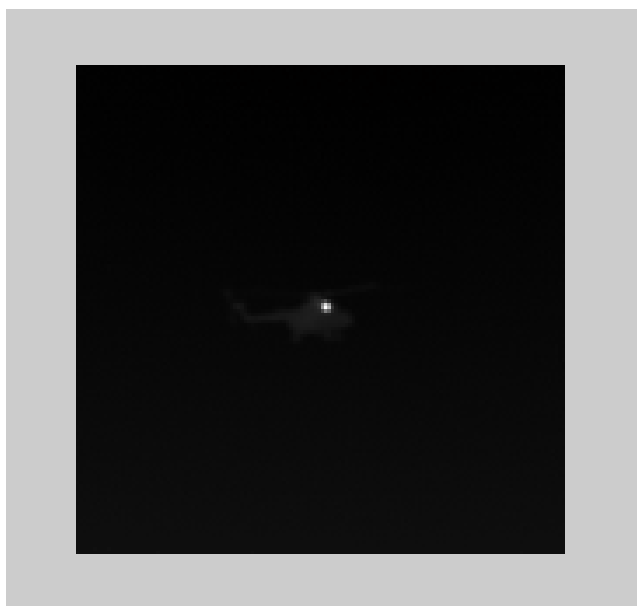
```
[J,T] = histeq(I);  
figure(2);subplot(211);imshow(J);subplot(212);imhist(J);  
figure(3);plot([0:255]/255,T);
```



```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
function [nn,cum] = computeCumulativeHistogram(img,nbins)  
  
    nn = imhist(img,nbins)';  
    cum = cumsum(nn);  
  
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%  
function I = createTransformationToIntensityImage(a,hgram,m,n,nn,cum)  
  
    cumd = cumsum(hgram*numel(a)/sum(hgram));  
  
    % Create transformation to an intensity image by minimizing the error  
    % between desired and actual cumulative histogram.  
    tol = ones(m,1)*min([nn(1:n-1),0;0,nn(2:n)])/2;  
    err = (cumd(:)*ones(1,n)-ones(m,1)*cum(:)') + tol;  
    d = find(err < -numel(a)*sqrt(eps));  
    if ~isempty(d)  
        err(d) = numel(a)*ones(size(d));  
    end  
    [dum,I] = min(err); %#ok  
    I = (I-1)/(m-1);
```

```
[nn,cum] = computeCumulativeHistogram(a,n);  
T = createTransformationToIntensityImage(a,hgram,m,n,nn,cum);  
b = grayxform(a, T);  
if nargout == 0  
    imshow(b);  
return;  
elseif classChanged  
    out = im2int16(b);  
else  
    out = b;  
end
```





## 直方图规定化

将原始图象的直方图转换为期望的直方图的形状

- 直方图规定化：运用均衡化原理的基础

算法思想：

设： $\{r_k\}$  是原图象的灰度级，

$\{z_k\}$  是符合指定直方图结果图象的灰度级

目标：找到一个灰度级变换函数  $T$ ，使：

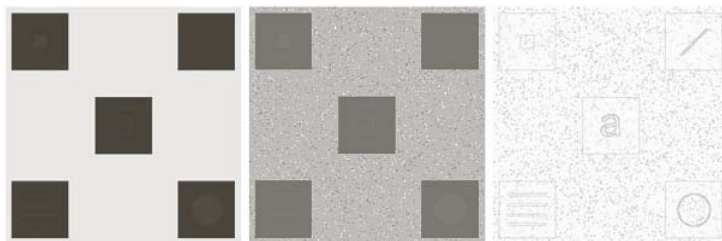
$$z_k = T(r_k) \quad r \rightarrow z$$

## 局部增强

有时需要对图像小区域细节的局部增强. 解决的办法就是在图像每一个像素的邻域中, 根据灰度级分布设计变换函数. 然后利用前面介绍的技术来进行局部增强。

步骤:

- ①定义一个方形或矩形的区域(邻域), 该区域的中心位置在某个像素点
- ②计算该邻域的直方图, 利用前面介绍的技术来得到变换函数.
- ③使用该变换函数来映射该区域的中心像素的灰度;
- ④把该区域的中心从一个像素移动至另一像素. 重复②~④



(a)原图 (b)全局均衡化的结果 (c)对每一个像素用 $7 \times 7$ 邻域局部增强均衡化的结果

## 在图像增强中使用直方图统计学

直方图除可用于对图像进行增强外，还可以使用直接从直方图获得的统计参数。

令 $r$ 表示在区间 $[0, L-1]$ 上代表离散灰度的离散随机变量，并且令 $p(r_i)$ 代表对应于 $r$ 的第 $i$ 个值得归一化直方图分量。

全局灰度均值:  $m = \sum_{i=0}^{L-1} r_i \cdot p(r_i) \quad (3.3-18)$

全局灰度方差:  $\mu_2(r) = \sum_{i=0}^{L-1} (r_i - m)^2 \cdot p(r_i) \quad (3.3-19)$

局部灰度均值:  $m_{S_{xy}} = \sum_{i=0}^{L-1} r_i \cdot p_{S_{xy}}(r_i) \quad (3.3-22)$

局部灰度标准差:  $\sigma_{S_{xy}} = \sqrt{\sum_{i=0}^{L-1} (r_i - m_{S_{xy}})^2 \cdot p_{S_{xy}}(r_i)} \quad (3.3-23)$

◆ 灰度均值: 对平均灰度的一种度量;

◆ 灰度方差 (标准差): 对平均对比度的一种度量;



## 增强暗区域，尽可能保持亮区域不变。

①判断一个区域  $S_{xy}$  是暗区域还是亮区域；

*if  $m_{S_{xy}} \leq k_0 M_G$  ( $0 < k_0 < 1.0$ ) then  $s_{xy}$  is a dark area*

②判断一个区域  $S_{xy}$  的对比度是否需要处理（增强）；

*if  $\sigma_{S_{xy}} \leq k_2 \sigma_G$  ( $0 < k_2 < 1.0$ ) then it is necessary to enhance  $s_{xy}$*

③限定能接受的最低对比度值，避免增强标准差为0的恒定区域；

*if  $\sigma_{S_{xy}} < k_1 \sigma_G$  ( $k_1 < k_2$ ) then it isn't necessary to enhance  $s_{xy}$*

$m_G$ ——全局平均值；  $\sigma_G$ ——全局标准差；

$$g(x, y) = \begin{cases} E \cdot f(x, y) & \text{if } m_{s_{xy}} \leq k_0 \sigma_G \text{ \& } k_1 \sigma_G \leq \sigma_{s_{xy}} \leq k_2 \sigma_G \\ f(x, y) & \text{other} \end{cases}$$

其中:  $f(x, y)$ —— $(x, y)$ 处的原灰度值;

$g(x, y)$ —— $(x, y)$ 处的新灰度值;

$$0 < k_0 < 1, \quad 0 < k_1 < k_2$$

$$\begin{cases} k_2 > 1 & \text{增亮} \\ 0 < k_2 < 1 & \text{减暗} \end{cases} \quad \begin{cases} E > 1 & \text{增亮} \\ 0 < E < 1 & \text{减暗} \end{cases}$$