



第5章 图像增强 Part1

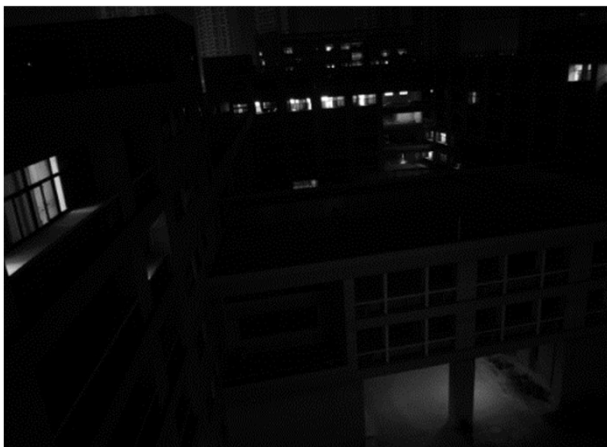
主讲教师 张涛

电子信息与电气工程学院

第5章 图像增强

问题的提出:

- 在实际现场环境，受外在因素干扰程度不同，所获得的视频图像质量好坏不同。
- 对比度较差、视觉效果不好的图像，如何改善视觉质量？





第5章 图像增强

问题的提出:

- 图像增强（Image Enhancement）：指不考虑图像降质原因，利用各种数学方法和变换手段，增强图像中人们感兴趣部分或提高有用的图像特征的清晰度。
- 从图像质量来看，图像增强提高了图像的可懂度，改善了图像的视觉效果。



主要内容

- 5. 1 基于灰度级变换的图像增强
- 5. 2 基于直方图修正的图像增强
- 5. 3 基于照度反射模型的图像增强
- 5. 4 基于模糊技术的图像增强
- 5. 5 基于伪彩色处理的图像增强
- 5. 6 其他图像增强方法



5.1 基于灰度级变换的图像增强

灰度级变换，即借助变换函数 T ，将输入像素灰度值 $f(x,y)$ 映射成新的 $g(x,y)$ ，通过改变像素的亮度来增强图像。

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

通过灵活设计变换函数 T ，实现对比度增强。

由于一般都是将过暗的图像灰度值进行重新映射，扩展灰度级范围，使其分布在整个灰度值区间，因此通常称为扩展（stretching）。



5.1 基于灰度级变换的图像增强

5.1.1 线性灰度级变换

5.1.2 非线性灰度级变换



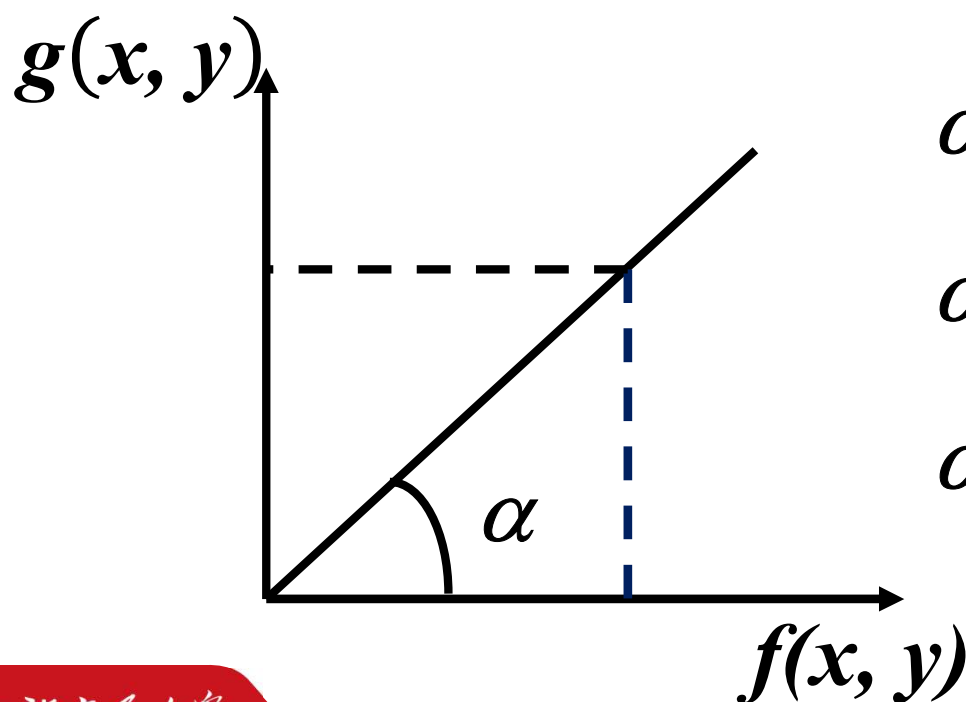
5.1.1 线性灰度级变换

基于灰度级变换的图像增强

(1) 基本线性灰度级变换

- 定义：基本线性变换函数为 $\tan \alpha$ ，有：

$$g(x, y) = f(x, y) \cdot \tan \alpha$$



$\alpha=45^\circ$ ，灰度范围不变；

$\alpha<45^\circ$ ，压缩灰度范围；

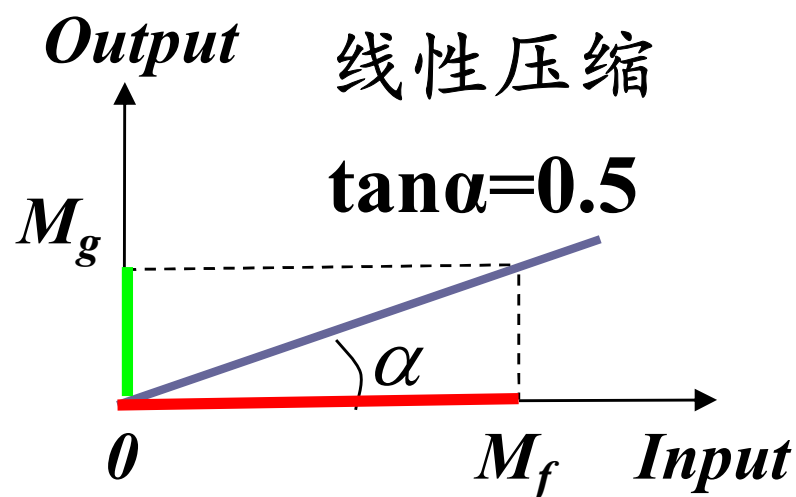
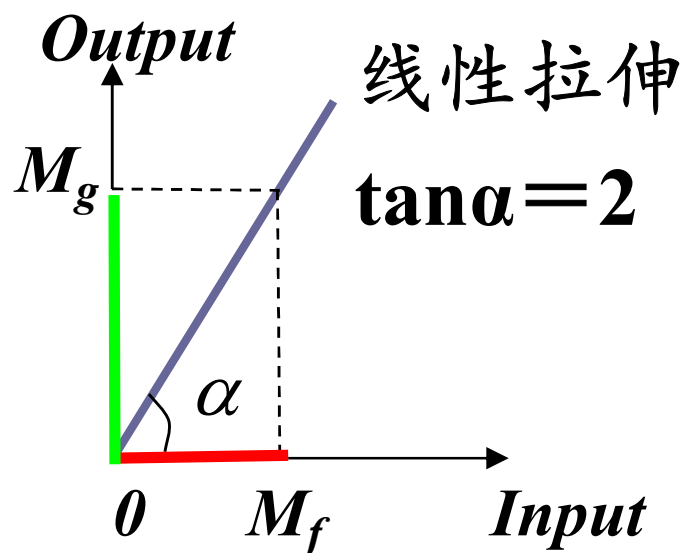
$\alpha>45^\circ$ ，拉伸灰度范围；

5.1.1 线性灰度级变换

基于灰度级变换的图像增强

■ 示例

1	2	1	4	3
1	3	2	3	4
4	6	2	7	4
5	3	4	4	5
2	3	5	3	2



2	4	2	7	6
2	6	4	6	7
7	7	4	7	7
7	6	7	7	7
4	6	7	6	4

1	1	1	2	2
1	2	1	2	2
2	3	1	4	2
3	2	2	2	3
1	2	3	2	1

5.1.1 线性灰度级变换

基于灰度级变换的图像增强

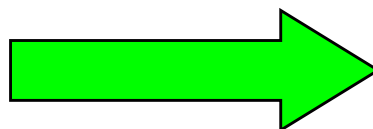
■ 效果

灰度0~255转变到0~127

原始图像



$$\tan\alpha = 0.5$$



$$\tan\alpha = 2$$



线性压缩

灰度0~127转变到0~255

128~255转变为255



线性拉伸



5.1.1 线性灰度级变换

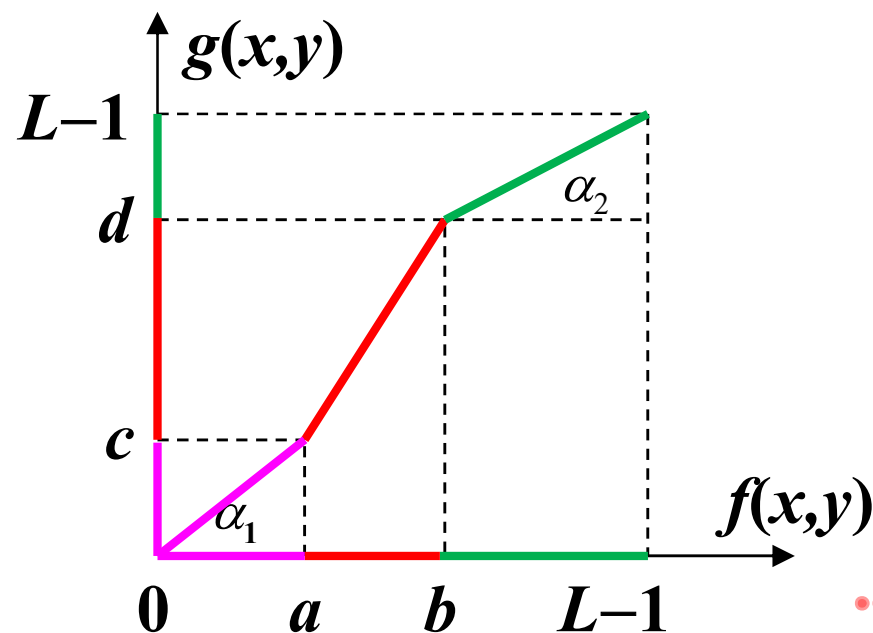
基于灰度级变换的图像增强

(2) 分段线性灰度级变换

■ 定义

$$g(x,y) = \begin{cases} \frac{c}{a}f(x,y) & 0 \leq f(x,y) < a \\ \frac{d-c}{b-a}[f(x,y)-a] + c & a \leq f(x,y) < b \\ \frac{L-1-d}{L-1-b}[f(x,y)-b] + d & b \leq f(x,y) < L-1 \end{cases}$$

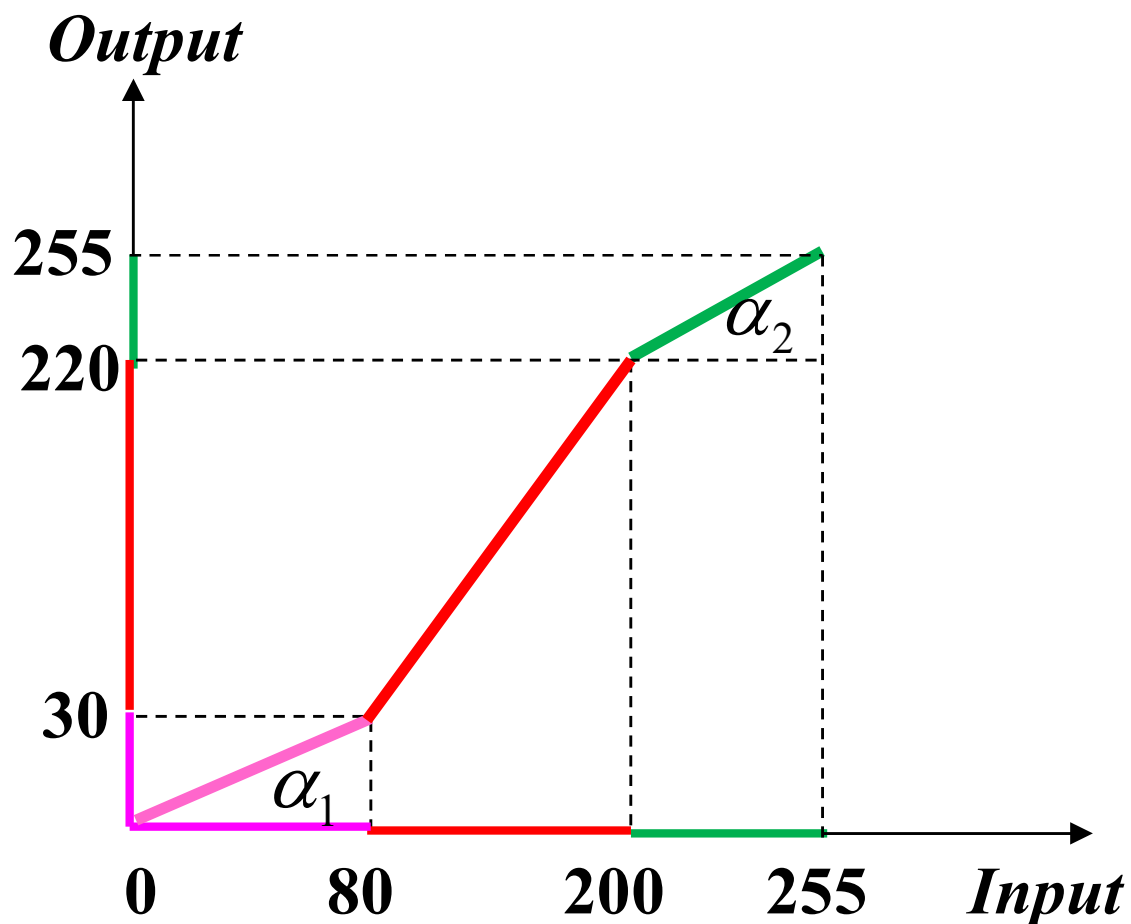
参数 a 、 b 、 c 、 d 为
确定三段线段斜率
的常数



5.1.1 线性灰度级变换

基于灰度级变换的图像增强

■ 示例



低灰度更低
高灰度更高

增强对比度



原图





5.1.1 线性灰度级变换

基于灰度级变换的图像增强

- 若用户仅对感兴趣范围 $[a, b)$ 线性拉伸，则：

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

保持灰度不变

$$g(x, y) = \begin{cases} c & 0 \leq f(x, y) < a \\ \frac{d-c}{b-a} [f(x, y) - a] + c & a \leq f(x, y) < b \\ d & b \leq f(x, y) < L-1 \end{cases}$$

截取式灰度变换



5.1.1 线性灰度级变换

基于灰度级变换的图像增强

□ 截取式灰度变换:

➤ 函数:

1) $J = \text{imadjust}(I, [\text{LOW_IN}; \text{HIGH_IN}], [\text{LOW_OUT}; \text{HIGH_OUT}], \text{GAMMA})$;

2) $\text{NEWMAP} = \text{imadjust}(\text{MAP}, [\text{LOW_IN}; \text{HIGH_IN}], [\text{LOW_OUT}; \text{HIGH_OUT}], \text{GAMMA})$;

3) $\text{RGB2} = \text{imadjust}(\text{RGB1}, \dots)$



5.1.1 线性灰度级变换

基于灰度级变换的图像增强

■ 例程

□ 程序

```
Image=im2double(rgb2gray(imread('lotus.bmp')));  
[h,w]=size(Image);  
NewImage1=zeros(h,w);  
NewImage2=zeros(h,w);  
NewImage3=Image;  
a=30/256; b=100/256; c=75/256; d=200/256;  
for x=1:w  
    for y=1:h
```



5.1.1 线性灰度级变换

基于灰度级变换的图像增强

if Image(y,x)<a

NewImage1(y,x)=Image(y,x)*c/a;

elseif Image(y,x)<b

NewImage1(y,x)=(Image(y,x)-a)*(d-c)/(b-a)+c;

else

NewImage1(y,x)=(Image(y,x)-b)*(1-d)/(1-b)+d;

end

if Image(y,x)>a && Image(y,x)<b

NewImage3(y,x)=(Image(y,x)-a)*(d-c)/(b-a)+c;

end



5.1.1 线性灰度级变换

基于灰度级变换的图像增强

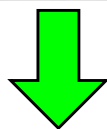
```
end  
  
end  
  
NewImage2=imadjust(Image,[a;b],[c;d]);  
figure;imshow(NewImage1);title('分段线性灰度级  
变换图像');  
  
figure;imshow(NewImage2);title('截取式灰度级变  
换图像');  
  
figure;imshow(NewImage3); title('高低端灰度级  
保持不变图像');
```


5.1.1 线性灰度级变换

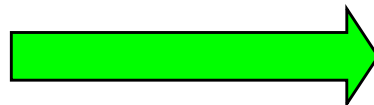
基于灰度级变换的图像增强

□

效果



$$\begin{aligned} a &= 30 & b &= 100 \\ c &= 75 & d &= 200 \end{aligned}$$



高
低
端
不
变



截
取
式
灰
度
变
换

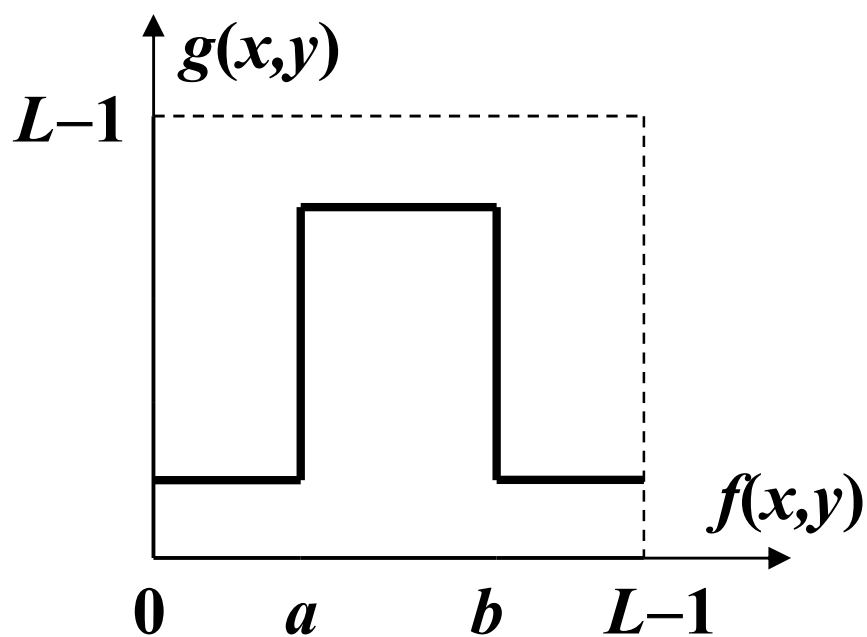
三
段
式
灰
度
变
换



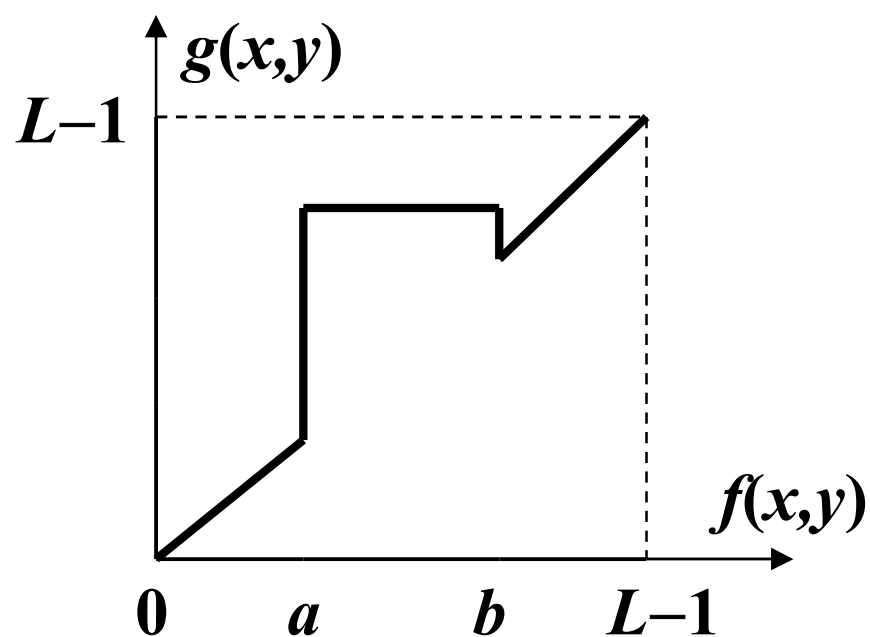
5.1.1 线性灰度级变换

基于灰度级变换的图像增强

■ 特殊的分段线性变换-----窗切片



(a) 区间外指定低灰度值



(b) 区间外灰度保持不变



5.1.1 线性灰度级变换

基于灰度级变换的图像增强

■ 例程

□ 程序

```
Image=im2double(imread('AG.jpg'));  
[h,w]=size(Image);  
imshow(Image);title('ACG图像');  
NewImage1=zeros(h,w);  
NewImage2=Image;  
a=170/256; b=200/256; c=90/256; d=250/256;  
for x=1:w  
    for y=1:h
```



5.1.1 线性灰度级变换

基于灰度级变换的图像增强

```
if Image(y,x)<a
    NewImage1(y,x)=c;
else
    NewImage1(y,x)=d;
end
if Image(y,x)>c && Image(y,x)<a
    NewImage2(y,x)=0;
end
end
end
figure;imshow(NewImage1);
figure;imshow(NewImage2);
```

5.1.1 线性灰度级变换

基于灰度级变换的图像增强

■ 例程

□ 效果



(a) 原图



(b) 区间外指定低灰度值



(c) 区间外灰度保持不变



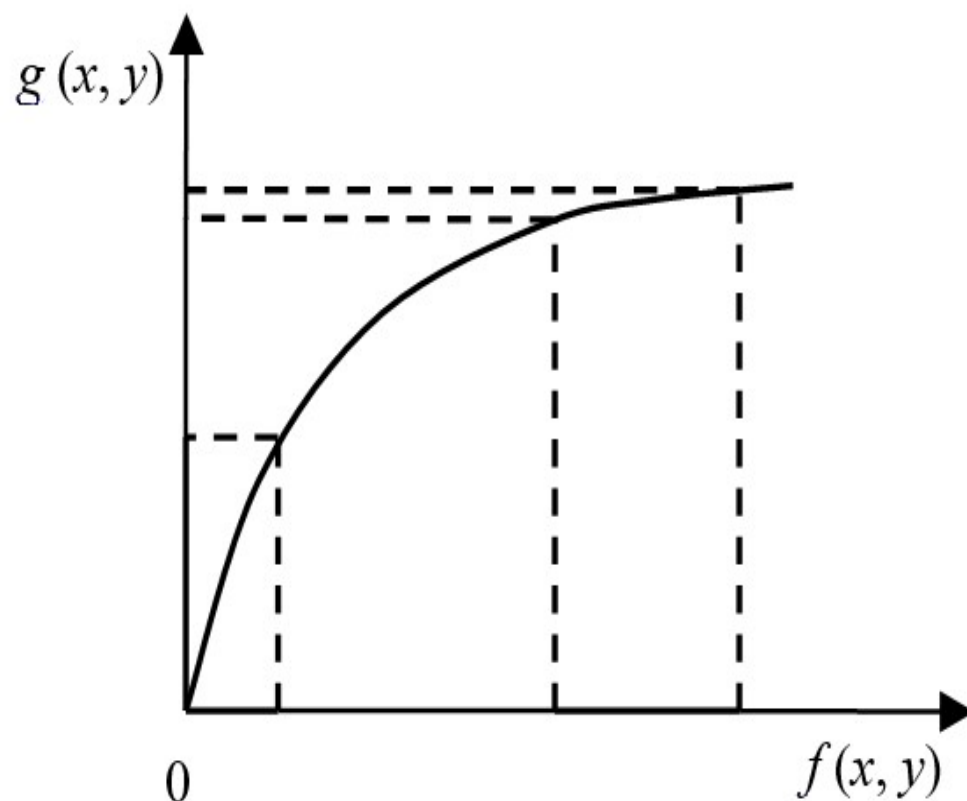
5.1.2非线性灰度级变换

基于灰度级变换的图像增强

(1) 对数变换

■ 定义

$$g(x, y) = c \cdot \log[f(x, y) + 1]$$



- 对数变换压缩图像高灰度区，扩展图像低灰度区。
- 对数变换一般适用于处理过暗图像。



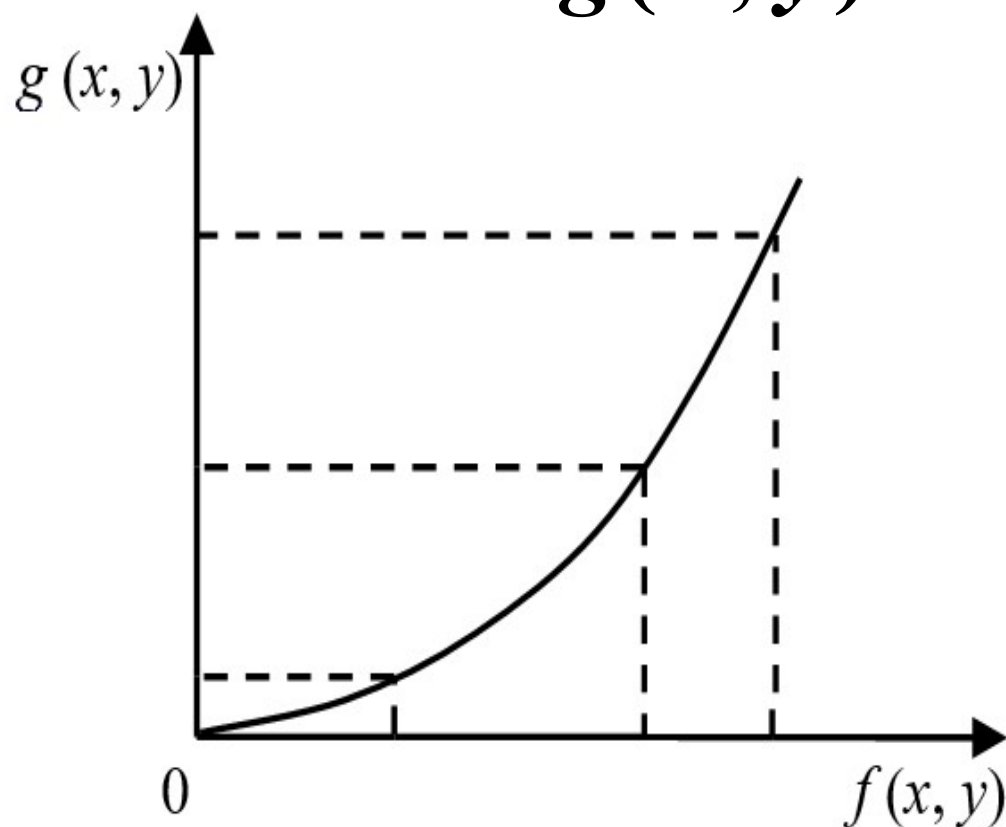
5.1.2非线性灰度级变换

基于灰度级变换的图像增强

(2) 指数变换

■ 定义

$$g(x, y) = b^{c \cdot [f(x, y) - a]} - 1$$



- 指数变换扩展图像高灰度区，压缩图像低灰度区。
- 指数变换一般适用于处理过亮图像。



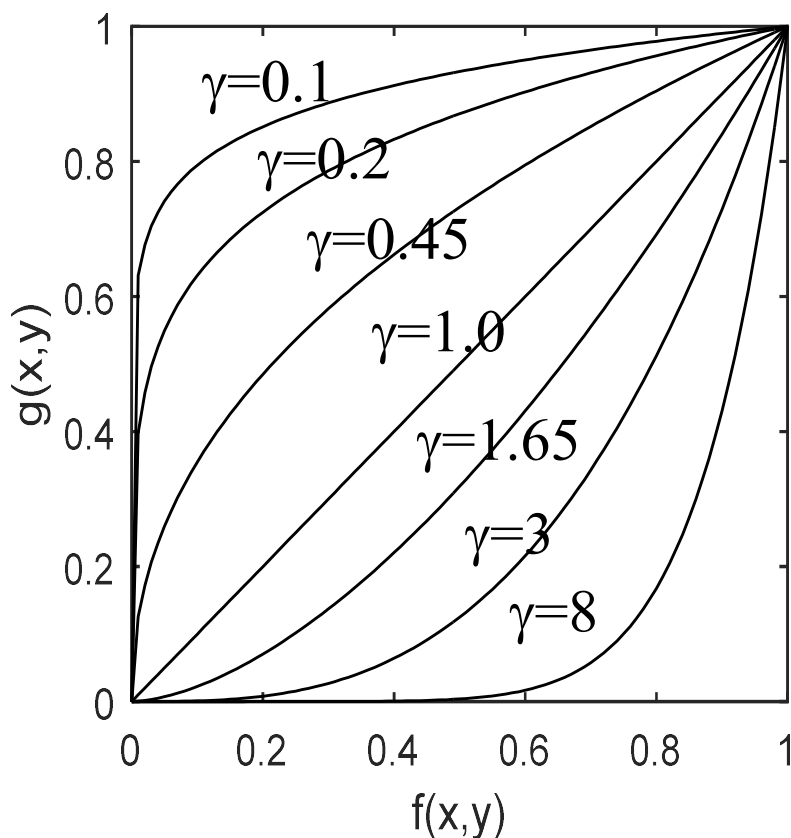
5.1.2非线性灰度级变换

基于灰度级变换的图像增强

(3) 幂次变换

■ 定义

$$g(x, y) = c \cdot [f(x, y)]^\gamma$$



- 当 $c=1$, γ 取不同值时, 可以一簇变换曲线。
- 幂次变换常用于图像获取、打印和显示的各种装置设备的伽马校正, 因此幂次变换也称为伽马变换。



5.1.2非线性灰度级变换

基于灰度级变换的图像增强

(4) 例程

■ 程序

```
Image=(rgb2gray(imread('Goldilocks.bmp')));  
Image=double(Image);  
NewImage1=46*log(Image+1);  
NewImage2=185*exp(0.325*(Image-225)/30)+1;  
a=0.5; c=1.1;  
NewImage3=[(Image/255).^a]*255*c;  
imshow(Image,[]);title('Goldilocks灰度图像');  
figure;imshow(NewImage1,[]);title('对数变换');  
figure;imshow(NewImage2,[]);title('指数变换');  
figure;imshow(NewImage3,[]);title('幂次变换');
```

5.1.2非线性灰度级变换

基于灰度级变换的图像增强

■ 效果



(a) 原图



(b) 对数变换



(c) 指数变换



(d) 幂次变换



5.2 基于直方图修正的图像增强

5.2.1 灰度直方图

5.2.2 直方图修正法理论

5.2.3 直方图均衡化

5.2.4 局部直方图均衡化



5.2.1 灰度直方图

基于直方图修正的图像增强

(1) 灰度直方图定义

■ 原理

灰度直方图表示的是数字图像中每一灰度级与其出现频数（即该灰度上出现像素的数目）间的统计关系。

即：横坐标表示灰度级，纵坐标表示频数或者相对频数（即该灰度级上像素出现的概率）

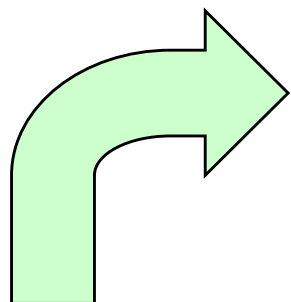
$$p(r_k) = \frac{n_k}{N}$$



5.2.1 灰度直方图

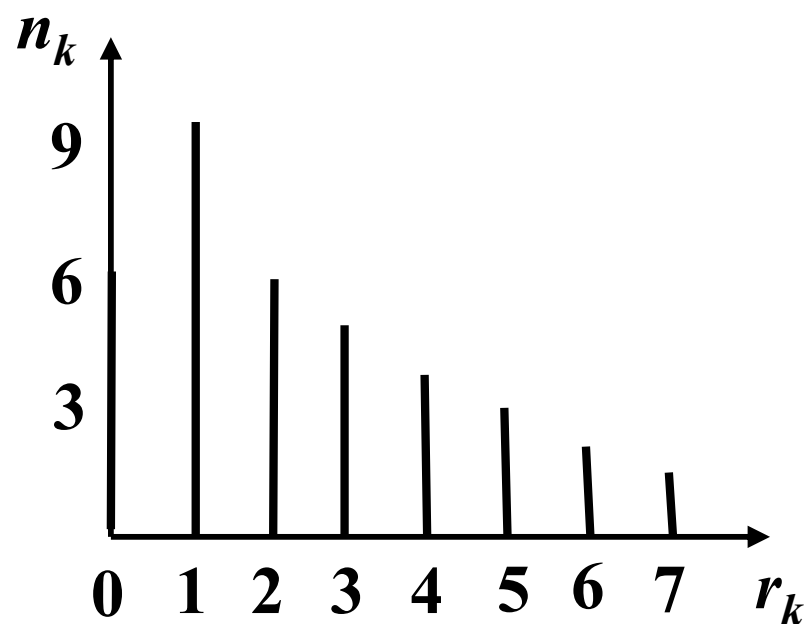
基于直方图修正的图像增强

■ 示例



r_k	0	1	2	3	4	5	6	7
n_k	6	9	6	5	4	3	2	1
$p(r_k)$	6/36	9/36	6/36	5/36	4/36	3/36	2/36	1/36

2	3	4	5	6	7
1	2	3	4	5	6
1	1	2	3	4	5
0	1	1	2	3	4
0	0	1	1	2	3
0	0	0	1	1	2





5.2.1 灰度直方图

基于直方图修正的图像增强

■ 例程(1)

```
Image=rgb2gray(imread('couple.bmp'));  
histgram=zeros(256);    [h w]=size(Image);  
for x=1:w  
    for y=1:h  
        histgram(Image(y,x)+1)=histgram(Image(y,x)+1)+1;  
    end  
end  
imshow(Image);title('couple灰度图像');  
figure;stem(histgram(),'.');  
axis tight;
```



5.2.1 灰度直方图

基于直方图修正的图像增强

■ 例程 (2)

□ 函数 **imhist(I,N):**
imhist(X,MAP):
[COUNTS,X] = imhist(...):

□ 程序

```
Image=rgb2gray(imread('couple.bmp'));  
figure;imhist(Image);  
axis tight;
```

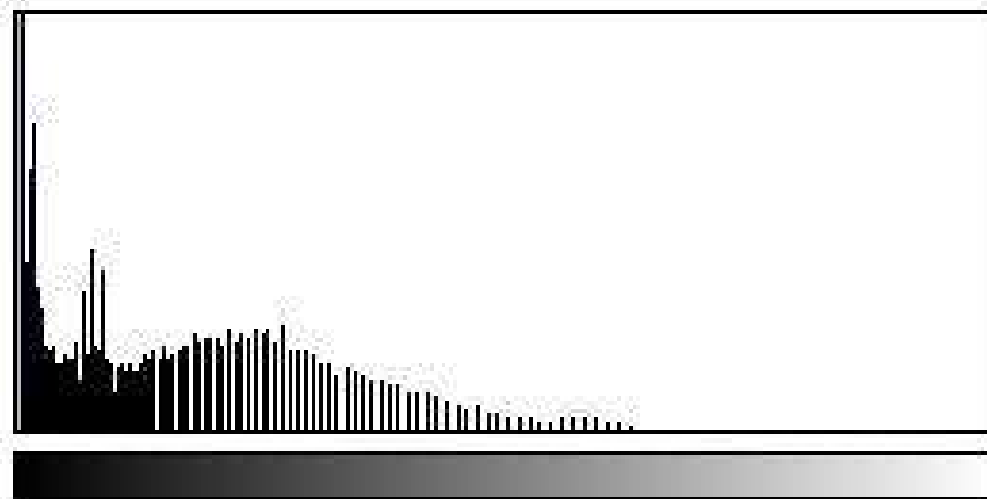
5.2.1 灰度直方图

基于直方图修正的图像增强

□ 效果



(a) 原图



(b) 灰度直方图



5.2.1 灰度直方图

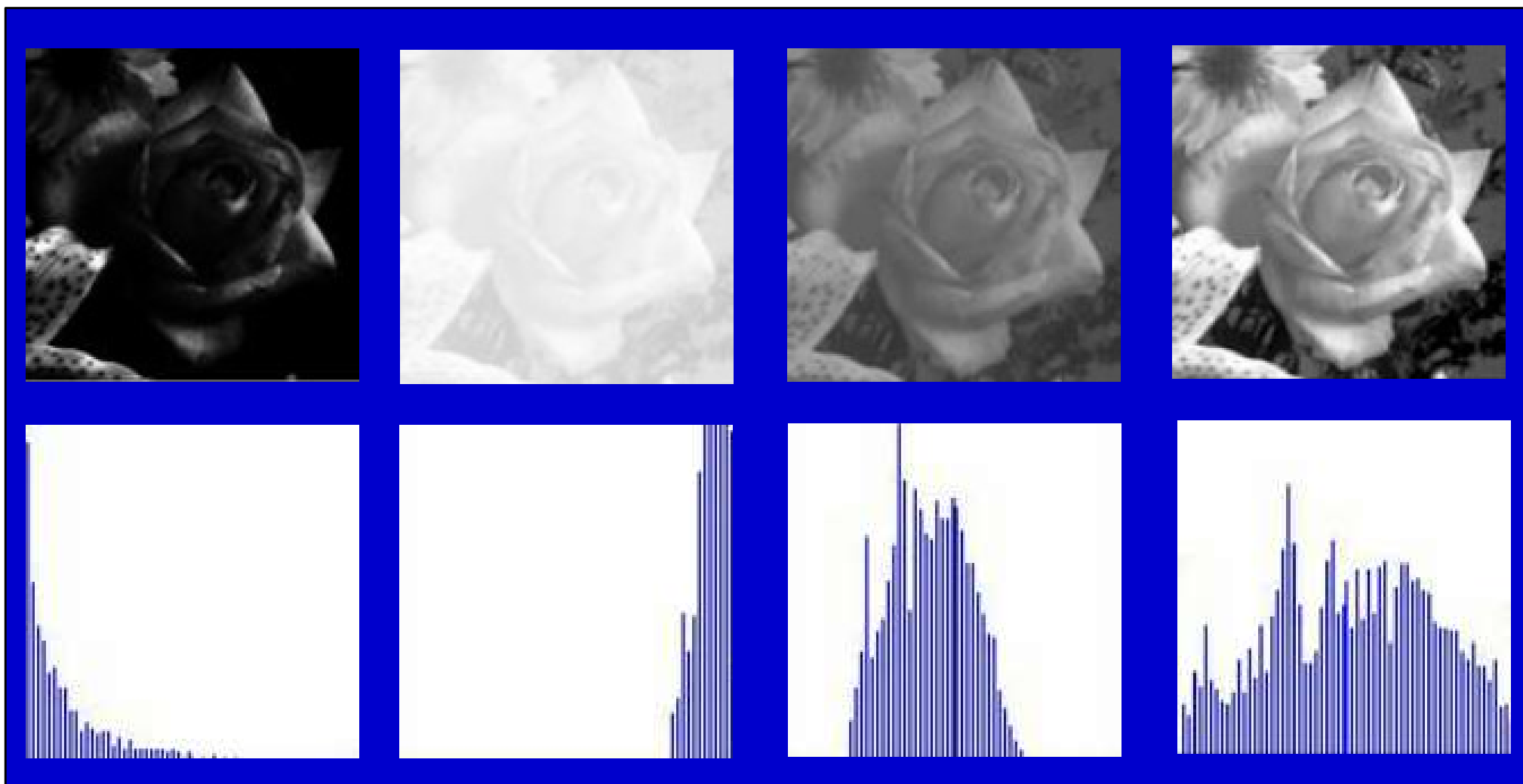
基于直方图修正的图像增强

(2) 灰度直方图性质

- 直方图不具有空间特性。直方图不能反映图像像素空间位置信息。
- 直方图反映图像大致描述。
- 一幅图像唯一对应相应的直方图，而不同的图像可以具有相同的直方图。
- 若一幅图像可分为多个子区，则多个子区直方图之和等于对应的全图直方图。

5.2.1 灰度直方图

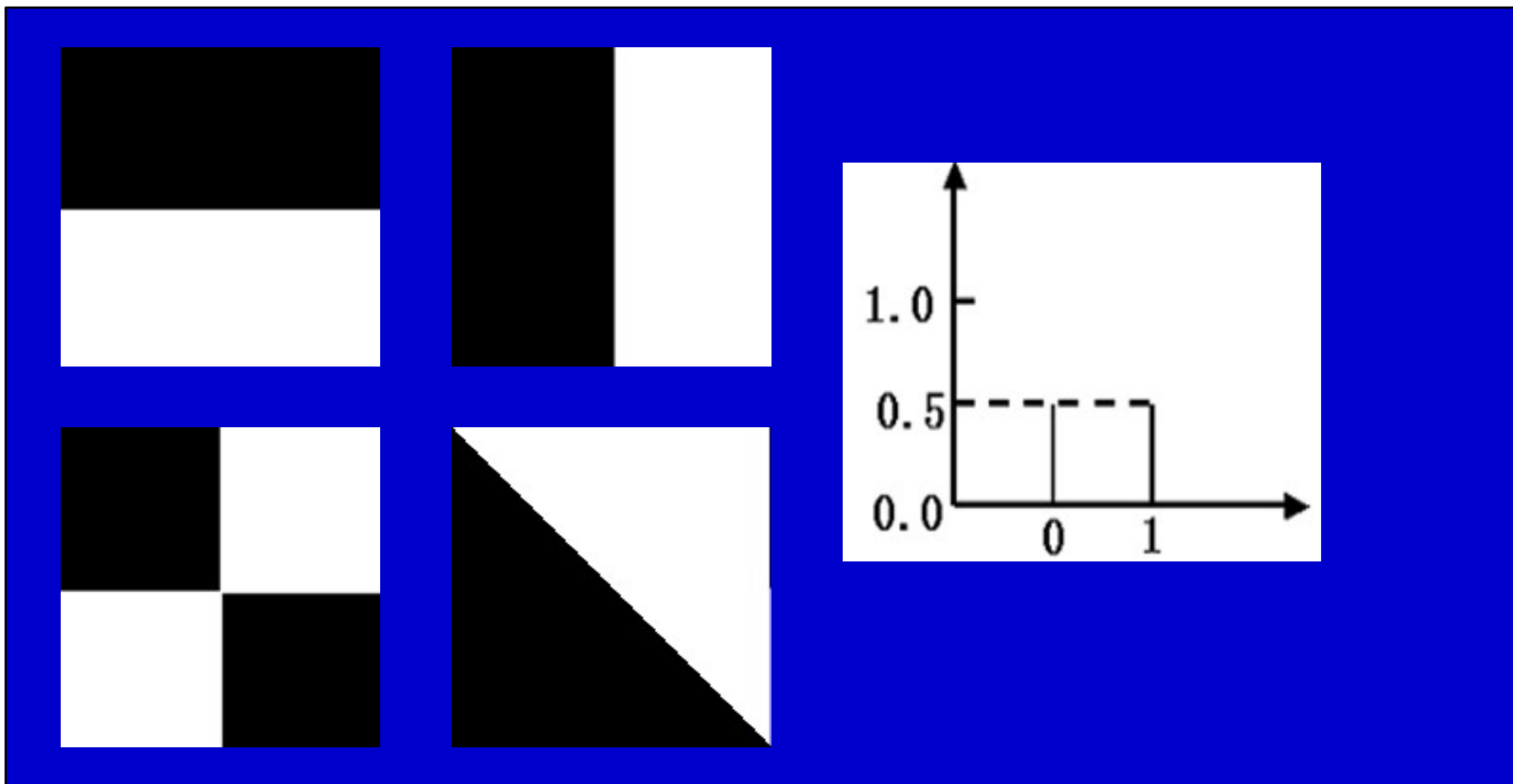
基于直方图修正的图像增强



不同直方图引起不同的视觉效果

5.2.1 灰度直方图

基于直方图修正的图像增强



同一直方图可以对应不同的图像



5.2.2直方图修正法理论

基于直方图修正的图像增强

- 图像的灰度动态范围太小或者说其直方图集中在某一个灰度区间，视觉效果不理想。
- 当图像直方图占满所有灰度级区间，且所有灰度级的概率分布相接近，即，直方图均匀分布的图像其视觉效果会最理想。
- 另一方面，从随机信号及信息量的角度来看，各个灰度级的概率分布等概率时，信息量最大，熵最大。
- 因此，需要寻找这一个变换，使得变换后图像直方图均匀，这就是直方图均衡化。



5.2.2直方图修正法理论

基于直方图修正的图像增强

设 r 为要增强像素灰度级， s 为增强后新灰度级

$$0 \leq r \leq 1 \quad 0 \leq s \leq 1$$

直方图修正公式： $s = T(r)$ 或 $r = T^{-1}(s)$

式中 $T(r)$ 为变换函数，要满足两个条件：

- ✓ $T(r)$ 在 $0 \leq r \leq 1$ 区域内单增，以保证灰度级从黑到白的次序不要出现反转。
- ✓ $T(r)$ 在 $0 \leq r \leq 1$ 区域内满足 $0 \leq s \leq 1$ ，保证变换的像素灰度级仍在允许的灰度级范围内。

关键核心为寻找满足两个条件的变换函数 $T(r)$



5.2.3直方图均衡化

基于直方图修正的图像增强

直方图均衡化，又叫做直方图均匀化。其目的是使所有灰度级出现的相对频数(概率)相同，此时图像所包含的信息量最大。

也就是使得原图像的灰度直方图修正为均匀分布的直方图，实现图像的全局整体均匀化。



5.2.3直方图均衡化

基于直方图修正的图像增强

(1) 直方图均衡化变换函数 $T(r)$ 的求解

$p_r(r)$ 和 $p_s(s)$ 分别表示 r 和 s 的灰度概率密度函数

$$\text{变换前、后} \begin{cases} p_s(s)ds = p_r(r)dr \\ \int_0^s p_s(s)ds = \int_0^r p_r(r)dr \end{cases}$$

因为，均衡化后 $p_s(s) = 1$

直方图均衡化
变换函数

$$s = T(r) = \int_0^r p_r(\omega) d\omega$$



5.2.3直方图均衡化

基于直方图修正的图像增强

(2) 数字图像的直方图均衡化

■ 原理

一幅离散数字图像,共 L 个灰度等级,其中第 k 个灰度级 r_k 出现的像素个数为 n_k , 图像总像素个数为 N 。则第 k 个灰度级出现的概率为:

$$P(r_k) = \frac{n_k}{N} \quad 0 \leq r_k \leq 1, k = 0, 1, \dots, L-1$$

进行均衡化处理的变换函数 $T(r)$ 为:

$$s_k = T(r_k) = \sum_{j=0}^k p_r(r_j) = \sum_{j=0}^k \frac{n_j}{N}$$

$$\text{且 } r_k = T^{-1}(s_k)$$



5.2.3直方图均衡化

基于直方图修正的图像增强

■ 算法步骤

□ 统计原始图像直方图；

□ 计算新的灰度级 $s_k = T(r_k) = \sum_{j=0}^k p_r(r_j) = \sum_{j=0}^k \frac{n_j}{N}$

□ 修正 s_k 为合理灰度级

□ 求新直方图

□ 用处理后的新灰度代替处理前的灰度，生成新图像



5.2.3直方图均衡化

基于直方图修正的图像增强

■ 示例

给定一幅 64×64 的8级灰度图像，其灰度级分布如表所示，对其进行直方图均衡化。

r_k	0	1/7	2/7	3/7	4/7	5/7	6/7	1
n_k	790	1023	850	656	329	245	122	81
$p_r(r_k)$	0.19	0.25	0.21	0.16	0.08	0.06	0.03	0.02



5.2.3直方图均衡化

基于直方图修正的图像增强

解：由原图灰度分布统计可看出，该图像绝大部分像素灰度值集中在低灰度区，图像整体偏暗。

□ 计算新的灰度级

$$s_0 = T(r_0) = \sum_{j=0}^0 p_r(r_j) = P_r(r_0) = 0.19$$

$$s_1 = T(r_1) = \sum_{j=0}^1 P_r(r_j) = P_r(r_0) + P_r(r_1) = 0.19 + 0.25 = 0.44$$

依此类推： $s_2 = 0.19 + 0.25 + 0.21 = 0.65$

$$s_3 = 0.19 + 0.25 + 0.21 + 0.16 = 0.81 \quad s_4 = 0.89$$

$$s_5 = 0.95 \quad s_6 = 0.98 \quad s_7 = 1$$



5.2.3直方图均衡化

基于直方图修正的图像增强

□ 修正 s_k 为合理的灰度级 s'_k

$$s_0 = 0.19 \approx \frac{1}{7} \quad s_1 = 0.44 \approx \frac{3}{7} \quad s_2 = 0.65 \approx \frac{5}{7}$$

$$s_3 = 0.81 \approx \frac{6}{7} \quad s_4 = 0.89 \approx \frac{6}{7}$$

$$s_5 = 0.95 \approx 1 \quad s_6 = 0.98 \approx 1 \quad s_7 = 1$$

则新图像对应只有5个不同灰度级别:

$$s'_0 = \frac{1}{7} \quad s'_1 = \frac{3}{7} \quad s'_2 = \frac{5}{7} \quad s'_3 = \frac{6}{7} \quad s'_4 = 1$$



5.2.3直方图均衡化

基于直方图修正的图像增强

□ 计算新的直方图

$$P_s(s'_0) = P_s\left(\frac{1}{7}\right) = P_r(r_0) = 0.19$$

$$P_s(s'_1) = P_s\left(\frac{3}{7}\right) = P_r(r_1) = 0.25$$

$$P_s(s'_2) = P_s\left(\frac{5}{7}\right) = P_r(r_2) = 0.21$$

$$P_s(s'_3) = P_s\left(\frac{6}{7}\right) = P_r(r_3) + P_r(r_4) = 0.16 + 0.08 = 0.24$$

$$P_s(s'_4) = P_s(1) = P_r(r_5) + P_r(r_6) + P_r(r_7) = 0.06 + 0.03 + 0.02 = 0.11$$

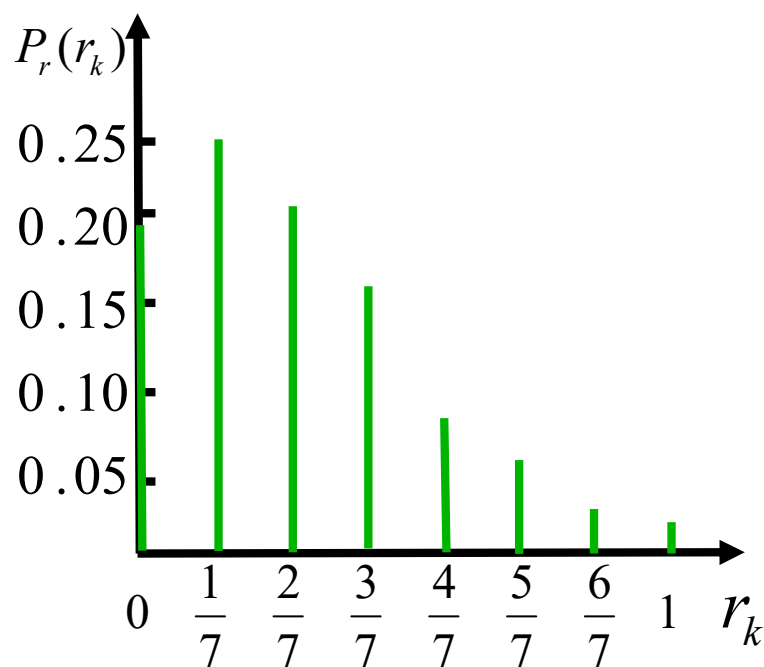


5.2.3直方图均衡化

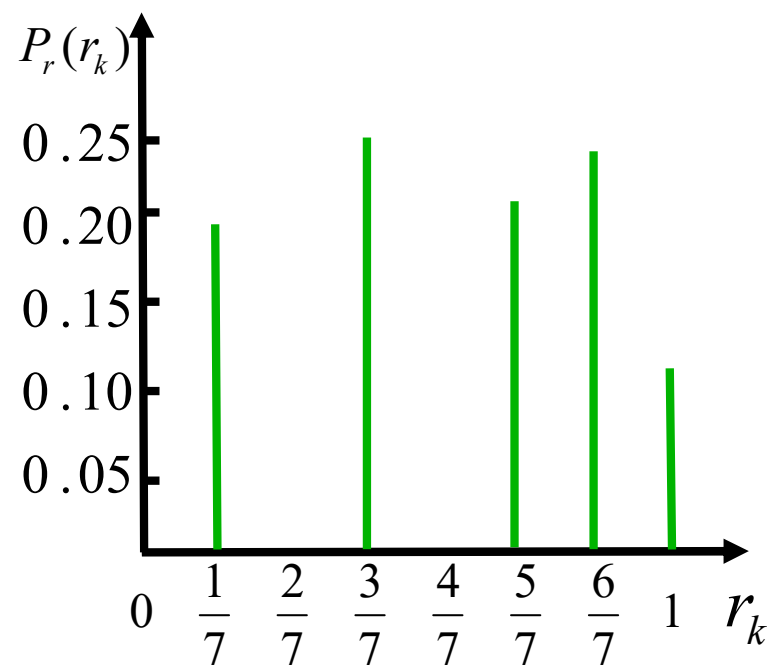
基于直方图修正的图像增强

□ 生成新图像

变换前灰度级	0	1/7	2/7	3/7	4/7	5/7	6/7	1
变换后灰度级	1/7	3/7	5/7	6/7	6/7	1	1	1



原始直方图



新的直方图



5.2.3直方图均衡化

基于直方图修正的图像增强

■ 例程(1)

□ 程序

```
Image=rgb2gray(imread('couple.bmp'));  
histgram =imhist(Image);  
[h w]=size(Image);  
NewImage=zeros(h,w);  
s=zeros(256);    s(1)=histgram(1);  
for t=2:256  
    s(t)=s(t-1)+histgram(t);  
end
```



5.2.3直方图均衡化

基于直方图修正的图像增强

```
for x=1:w
    for y=1:h
        NewImage(y,x)=s(Image(y,x)+1)/(w*h);
    end
end
imshow(Image);title('couple灰度图像');
figure;imhist(Image);title('couple图像的直方图');
axis tight;
figure;imshow(NewImage);title('均衡化后图像');
figure;imhist(NewImage);title('均衡化后图像直方图');
axis tight;
```




5.2.3直方图均衡化

基于直方图修正的图像增强

■ 例程(2)

□ 函数

$J = \text{histeq}(I, \text{HGRAM})$:

$J = \text{histeq}(I, N)$:

□ 程序

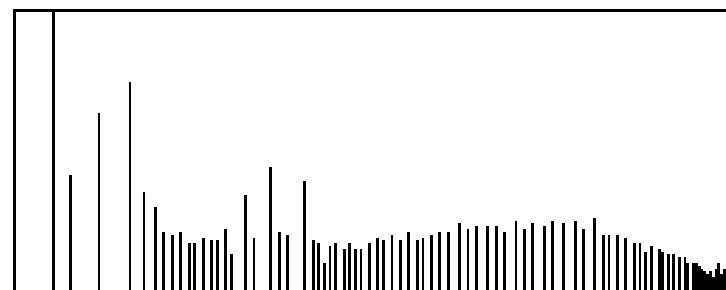
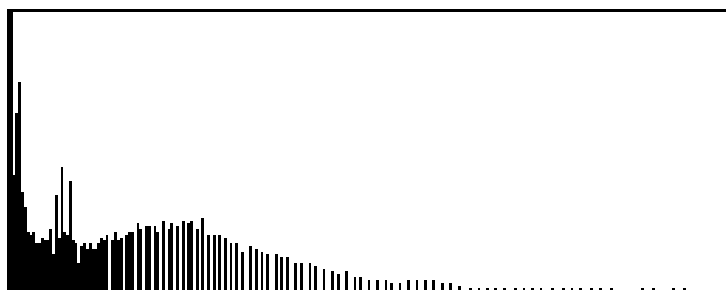
$\text{Image} = \text{rgb2gray}(\text{imread}('couple.bmp'))$;

$\text{NewImage} = \text{histeq}(\text{Image}, 256)$

5.2.3直方图均衡化

基于直方图修正的图像增强

□ 效果



直方图均衡化处理前后的图像以及直方图



5.2.3局部直方图均衡化

基于直方图修正的图像增强

根据区域的局部直方图统计特性来定义灰度级变换函数，进行均衡化处理，这就是局部直方图均衡化。

给出一幅数字图像，选定大小为 $w \times h$ 的矩形子块 S ，子块 S 内进行直方图均衡化处理，有：

$$s_k = T(r_k) = \sum_{j=0}^k p_S(r_j) = \sum_{j=0}^k \frac{n_j}{w \times h}$$

可分为子块不重叠、子块重叠和子块部分重叠的局部直方图均衡化。



5.2.3 局部直方图均衡化

基于直方图修正的图像增强

(1) 子块不重叠的局部直方图均衡化

- 将图像划分为一系列不重叠的相邻矩形子块集合 $\{S_i | i=1,2,...,num\}$ ，逐个独立地对每个子块中所有像素进行直方图均衡化处理并输出。
- 由于划分的各子块的灰度分布统计差异较大，因此增强处理后输出图像有明显的块效应。



5.2.3 局部直方图均衡化

基于直方图修正的图像增强

(2) 子块重叠的局部直方图均衡化

- 利用划分的子块的直方图信息，对子块进行直方图均衡化处理。
- 把均衡化处理后的子块中心像素的值作为该像素的输出值。
- 将子块在图像中逐像素移动，重复上述过程，直至遍历图像中所有像素。
- 该算法效率较低。



5.2.3 局部直方图均衡化

基于直方图修正的图像增强

(3) 子块部分重叠的局部直方图均衡化

- 划分大小为 $w \times h$ 的子块进行直方图均衡化。
- 将子块在图像中按照一定水平步长 $wstep$ 和垂直步长 $hstep$ 移动， $1 < wstep < w$ ， $1 < hstep < h$ 。
- 重复上述过程，直至遍历图像中所有像素。
- 将重叠区域的多次均衡化处理的结果取平均值作为该重叠区域中像素的输出值。
- 该算法的使用受到青睐。



5.2.3 局部直方图均衡化

基于直方图修正的图像增强

(4) 例程

■ 程序

```
Image=rgb2gray(imread('couple.bmp'));  
imshow(Image);title('原始图像');  
result1=blkproc(Image,[32 32],@histeq);  
figure,imshow(result1);title('无重叠局部直方图均衡化');  
[height,width]=size(Image);  
result2=zeros(height,width);  
n=16;  
hh=height+2*n;   ww=width+2*n;  
ff=zeros(hh,ww);
```




5.2.3局部直方图均衡化

基于直方图修正的图像增强

```
ff(n+1:hh-n,n+1:ww-n)=Image;
ff(1:n,n+1:ww-n)=Image(1:n,:);
ff(hh-n+1:hh,n+1:ww-n)=Image(height-n+1:height,:);
ff(:,1:n)=ff(:,n+1:n*2);
ff(:,ww-n+1:ww)=ff(:,ww+1-n*2:ww-n);
ff=uint8(ff);
for i=n+1:hh-n
    for j=n+1:ww-n
        lwc=histeq(ff(i-n:i+n,j-n:j+n),256);
        result2(i-n,j-n)=lwc(n+1,n+1); end
    end
end
figure,imshow(uint8(result2));title('重叠的局部直方图
均衡化图像');
```




5.2.3局部直方图均衡化

基于直方图修正的图像增强

```
sumf=int16(zeros(hh,ww)); num=zeros(hh,ww);  
for i=n+1:8:hh-n  
    for j=n+1:8:ww-n  
        lwc=int16(histeq(ff(i-n:i+n,j-n:j+n),256));  
        sumf(i-n:i+n,j-n:j+n)=sumf(i-n:i+n,j-n:j+n)+lwc;  
        num(i-n:i+n,j-n:j+n)=num(i-n:i+n,j-n:j+n)+1;  
    end  
end  
result3(:,:,:)=double(sumf(n+1:hh-n,n+1:ww-n));  
result3(:,:,:)=result3(:,:,:)./num(n+1:hh-n,n+1:ww-n);  
figure,imshow(uint8(result3(:,:,:)));title('部分重叠的局部  
直方图均衡化图像');
```

5.2.3 局部直方图均衡化

基于直方图修正的图像增强

■ 效果



(a) 原图



(b) 子块不重叠



(c) 子块重叠



(d) 子块部分重叠