

# 计算机图像处理

COMPUTER IMAGE PROCESSING



# 第三章 图像增强

# 目的

视觉效果更“好”，更“有用”的图像

对 图 像 进 行 加 工





*F.BMP*  $F(X,Y)$ 增强  
前

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

*G.BMP*  $G(X,Y)$ 增强后





*F.BMP F(X,Y)增强前*

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

HXLBMPFILE G;

```
G.IMAGEW = F.IMAGEW;
G.IMAGEH = F.IMAGEH;
G.IYGRBNUM = F.IYGRBNUM;
```

```
IF (!G.ALLOCATEMEM())
RETURN 1;
```

.....

```
G.SAVEBMPFILE( "G.BMP" );
RETURN 0;
```



*G.BMP G(X,Y)增强后*

HXLBMPFILE f;

```
if (!f.LoadBMPFILE( "f.bmp" ))
return 1;
```



*F.BMP F(X,Y)*增强  
前

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

HXLBMPFILE G;

```
G.IMAGEW = F.IMAGEW;
G.IMAGEH = F.IMAGEH;
G.IYGRBNUM = F.IYGRBNUM;
```

```
IF (!G.ALLOCATEMEM())
RETURN 1;
```

```
FOR ( I =0; I<F.IMAGEH; I++)
```

```
FOR (J =0; J<F.IMAGEW; J++)
```

```
G.PDATAAT(I)[J] = T
[ F.PDATAAT(I)[J] ];
```

```
G.SAVEBMPFILE( "G.BMP" );
RETURN 0;
```

HXLBMPFILE f;

```
if (!f.LoadBMPFILE( "f.bmp" ))
return 1;
```

BYTE

T[256]={T0,T1,T2,T3,T4,T5,.....,T255}

输入

输出

*X,Y)*增强后

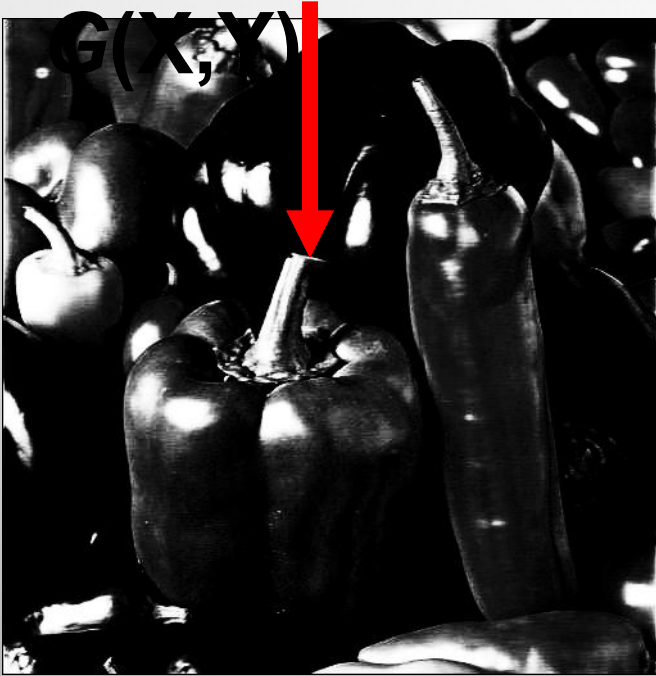


如果用  $s$  和  $t$  分别代表  $f$  和  $g$  在  $(x,y)$  处的灰度值:

$$t = T(s)$$

$T=$

$G(X,Y)$

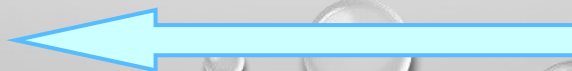
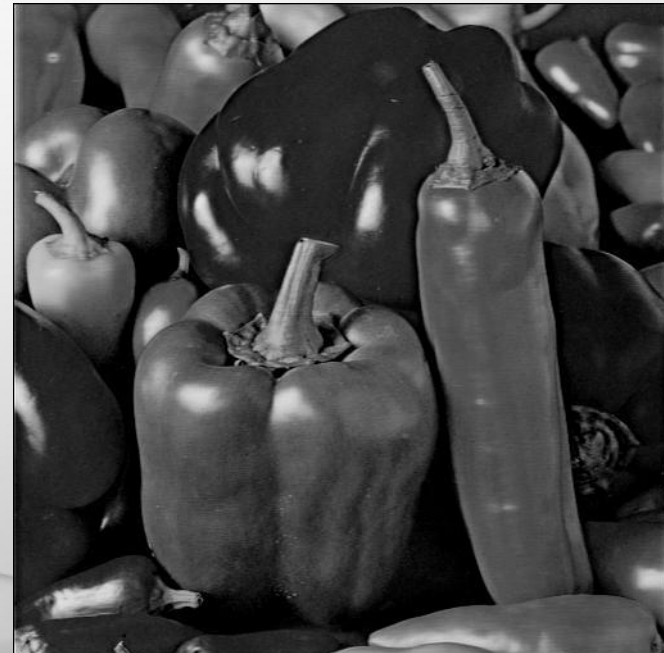
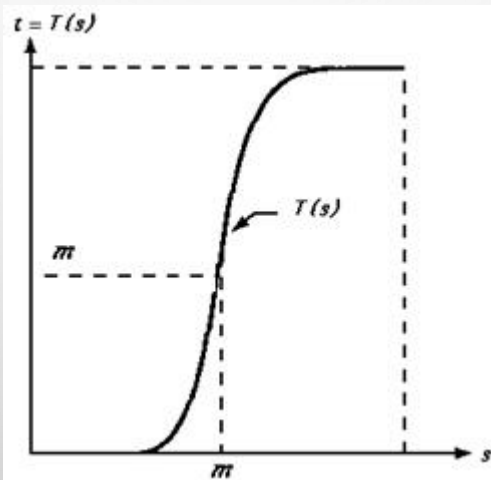


$S=$

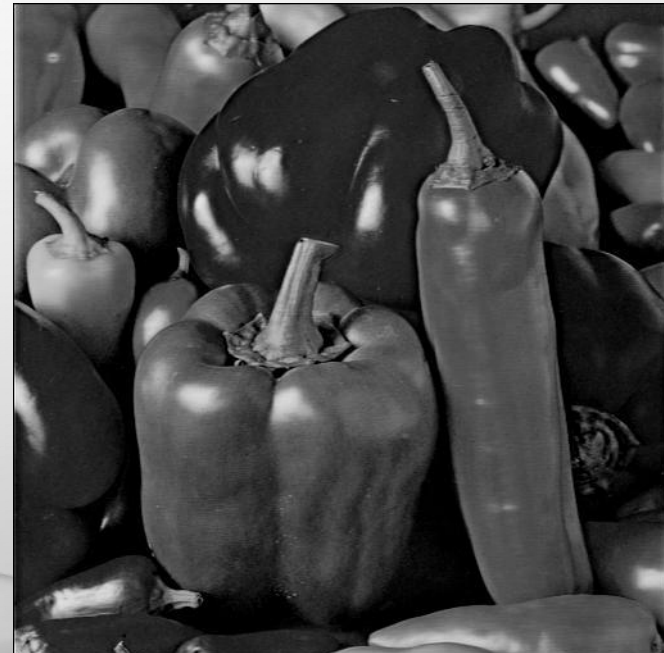
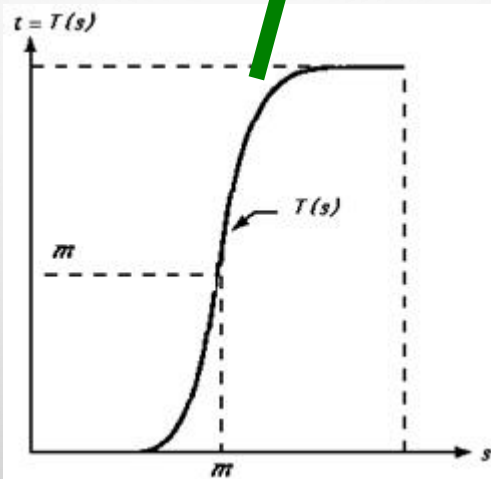
$F(X,Y)$







BYTE  $T[256]=\{0,0,0,0,0,0,0,0,0,0,0,0,1,4,8,10,16,\dots,255,255,255\}$



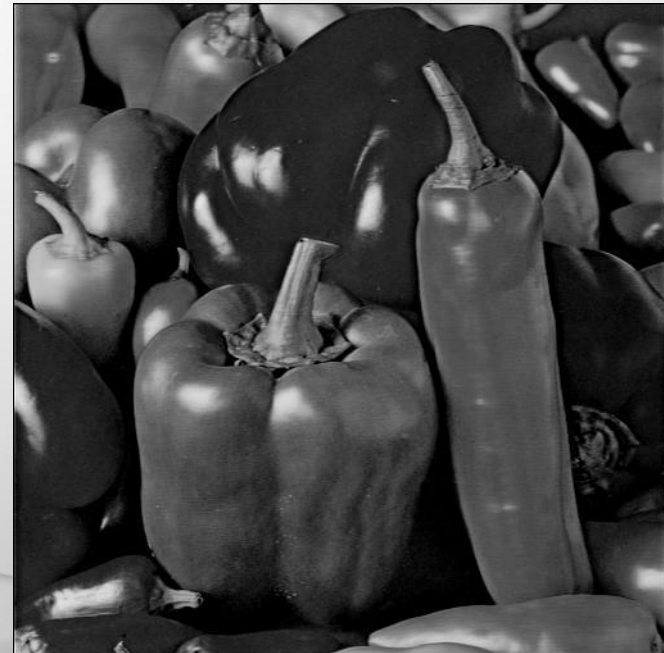
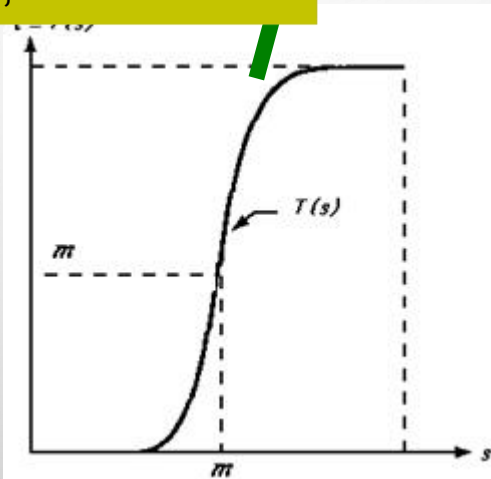
```

BYTE TI[256]={0,0,0,0,0,0,0,0,0,0,0,1,4,8,10,16,.....,255,255,255}
HXLBMPFILE F,G;
IF (!F.LOADBMPFILE( "F.BMP")) RETURN
1;
G.IMAGEW = F.IMAGEW; G.IMAGEH =
F.IMAGEH;
G.IYGRBNUM = F.IYGRBNUM;
IF (!G.ALLOCATEMEM()) RETURN 1;
FOR ( I = 0; I < F.IMAGEH; I++)
FOR ( J = 0; J < F.IMAGEW; J++)

G.PDATAAT(I)[J] = T
[ F.PDATAAT(I)[J] ];

G.SAVEBMPFILE( "G.BMP" );
RETURN 0;

```



BYTE T[256]={0,0,0,0,0,0,0,0,0,0,0,1,4,8,10,16,.....,255,255,255}

```
FILE *F;
IF (!F.LOADBMPFILE( "F.BMP")) RETURN
1;
```

```
G.IMAGEW = F.IMAGEW; G.IMAGEH =
F.IMAGEH;
```

```
G.IYGRBNUM = F.IYGRBNUM;
```

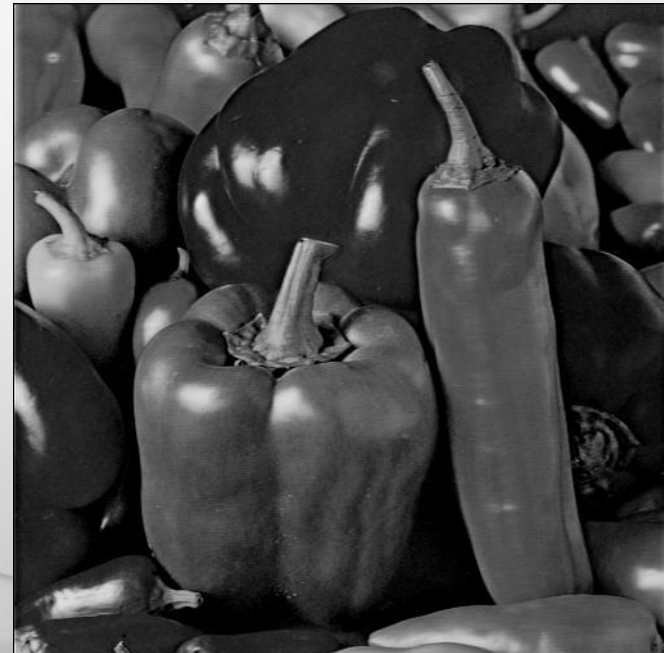
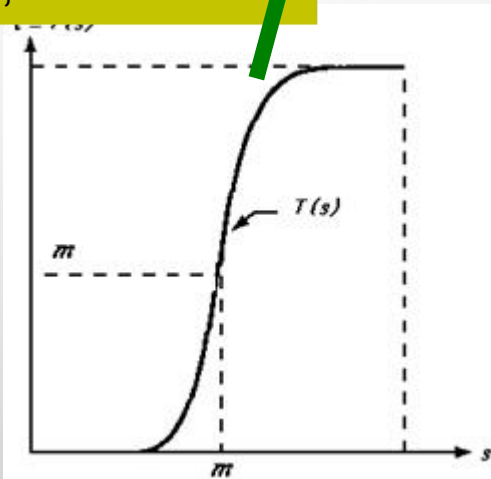
```
IF (!G.ALLOCATEMEM()) RETURN 1;
```

```
FOR ( I = 0; I < F.IMAGEH; I++)
FOR ( J = 0; J < F.IMAGEW; J++)
```

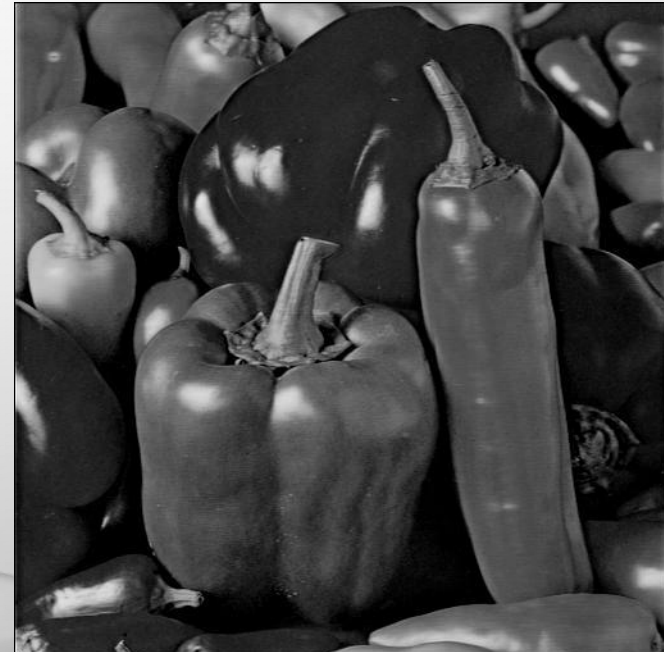
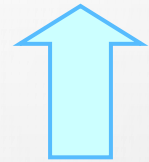
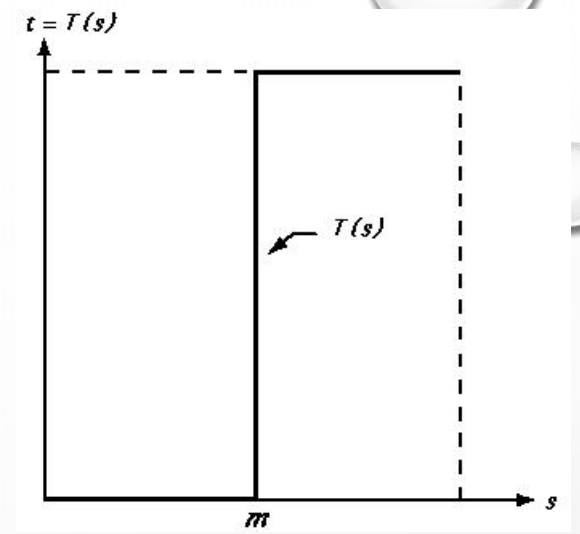
```
G.PDATAAT(I)[J] = T
[ F.PDATAAT(I)[J] ];
```

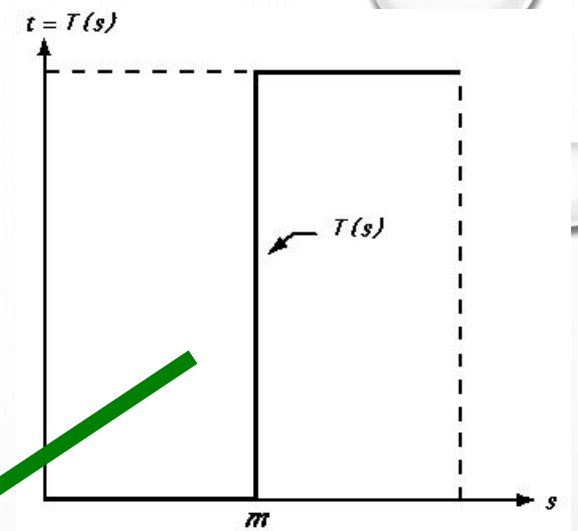
```
G.SAVEBMPFILE( "G.BMP" );
```

```
RETURN
```









BYTE  $T[256] = \{0, 0, 0, \dots, 0, 0, 0, 255, 255, 255, \dots, 255, 255, 255\}$



# 直接灰度变换

## 1、图像求反

假设对灰度级范围是 $[0, L-1]$ 的图像求反，就是通过变换将 $[0, L-1]$ 变换到 $[L-1, 0]$ ，变换公式如下：

$$t = L - 1 - s$$

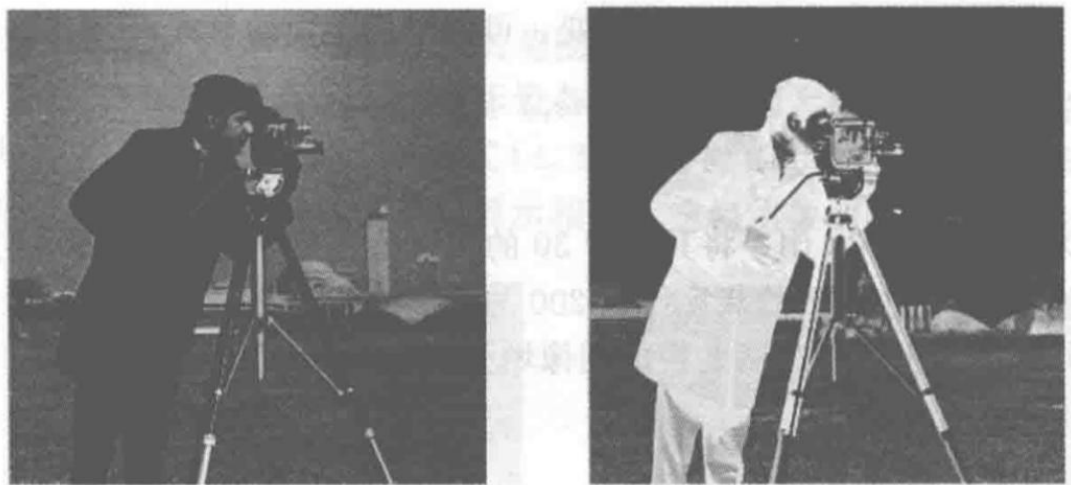
# 直接灰度变换

## 1、图像求反

假设对灰度级范围是 $[0, L-1]$ 的图像求反，就是通过变换将 $[0, L-1]$ 变换到 $[L-1, 0]$ ，变换公式如下：

$$t = L - 1 - s$$

此方法适用于增强嵌入于图像暗色区域的白色或灰色细节。





# 直接灰度变换

## 1、图像求反

假设对灰度级范围是 $[0, L-1]$ 的图像求反，就是通过变换将 $[0, L-1]$ 变换到 $[L-1, 0]$ ，变换公式如下：

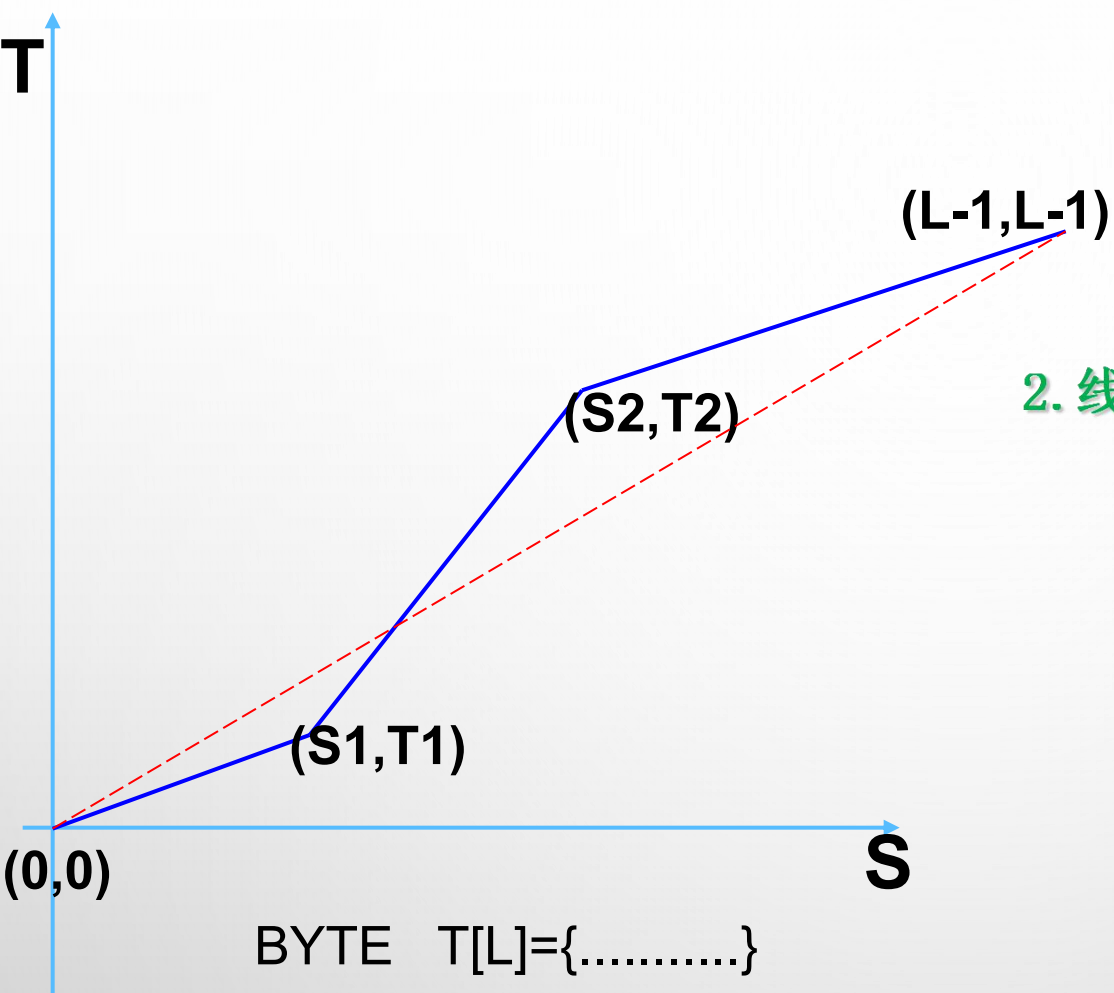
$$t = L - 1 - s$$

此方法适用于增强嵌入于图像暗色区域的白色或灰色细节。

BYTE  $T[256]=\{\dots\dots\dots\}$

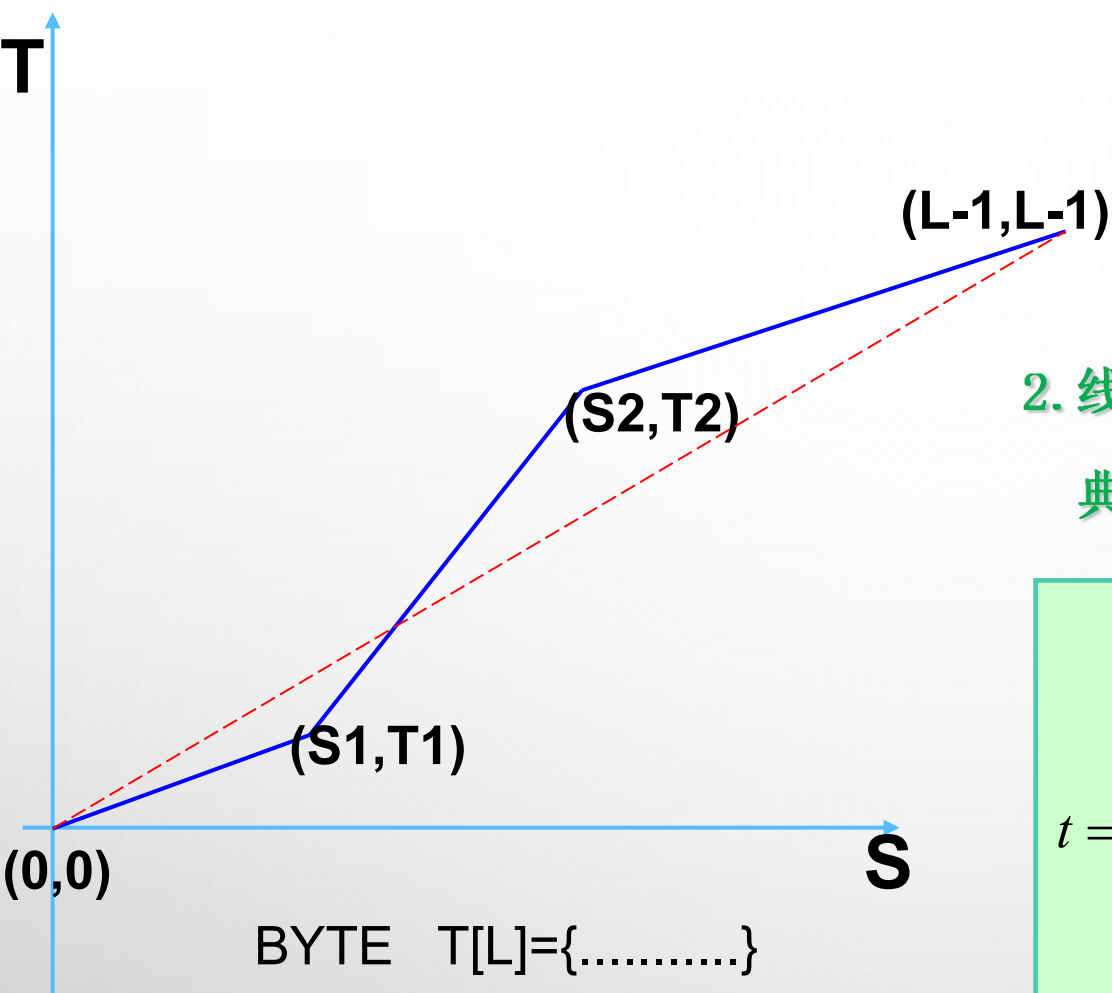
T的变换曲线？

# 直接灰度变换



用分段线性法，将需要的图像细节灰度级拉伸，增强对比度，不需要的细节灰度级压缩

# 直接灰度变换



## 2. 线性灰度变换

典型的分段线性变换如下：

$$t = \begin{cases} \frac{t_1}{s_1} s & 0 \leq s \leq s_1 \\ \frac{t_2 - t_1}{s_2 - s_1} [s - s_1] + t_1 & s_1 < s \leq s_2 \\ \frac{L-1-t_2}{L-1-s_2} [s - s_2] + t_2 & s_2 < s \leq L-1 \end{cases}$$

用分段线性法，将需要的图像细节灰度级拉伸，增强对比度，不需要的细节灰度级压缩



对于左侧原图将其小于30的灰度值不变，将30~150的灰度值拉伸到30~200，同时压缩150~255的灰度值到200与255之间，得到右侧图片。

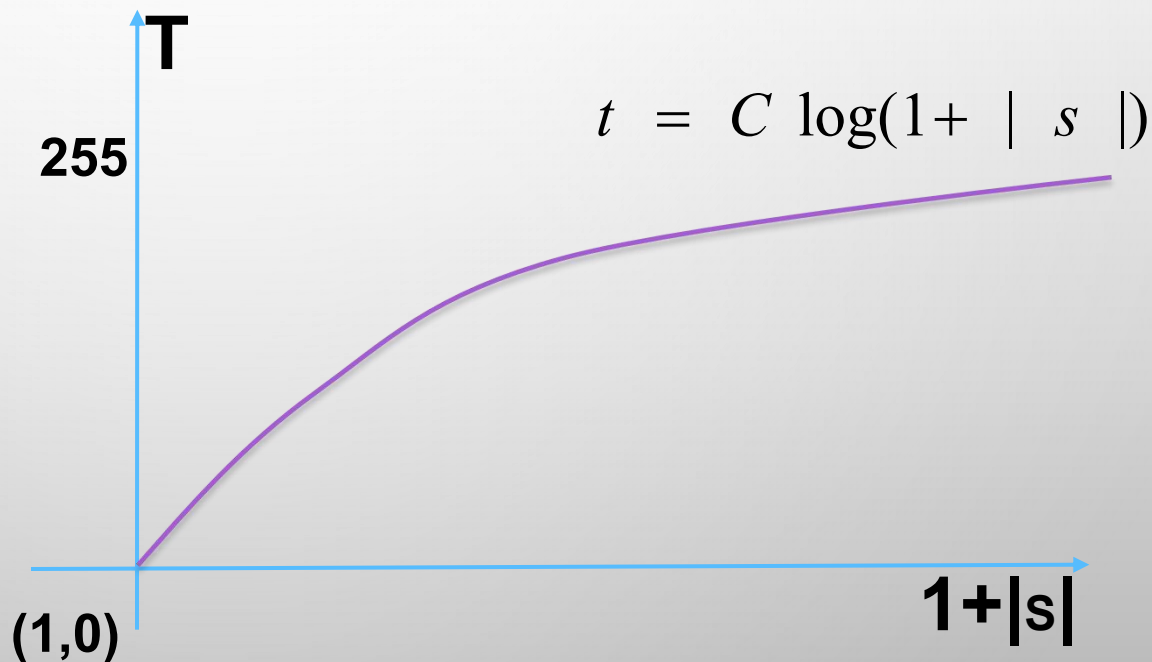


# 直接灰度变换

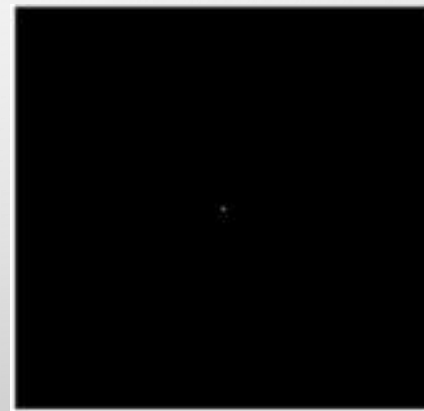
## 3. 对数变换

要消除因动态范围太大而引起的失真，一种有效的方法是对原图像的动态范围进行压缩，最常用的是借助对数形式对动态范围进行调整：

$$t = C \log(1 + |s|)$$

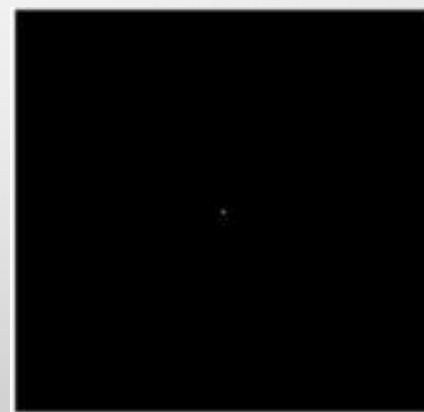
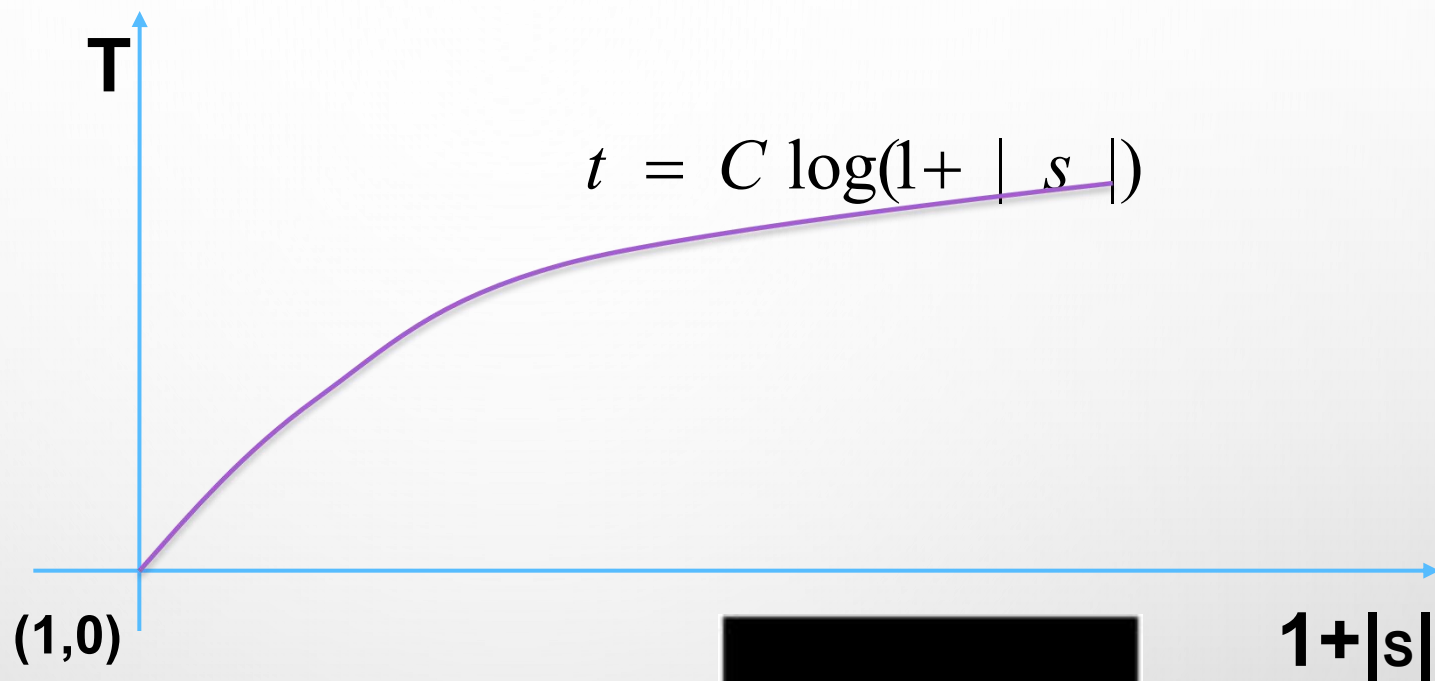


- 例：傅里叶谱的范围  $|S| \in [0, 1.6 \times 10^6]$ ，显示时使用对数变换。



原图

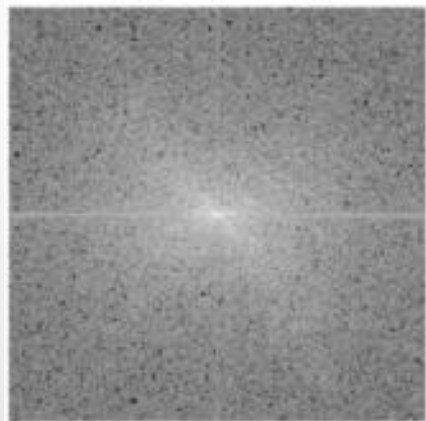
- 例：傅里叶谱的范围  $|S| \in [0, 1.6 \times 10^6]$ ，显示时使用对数变换。



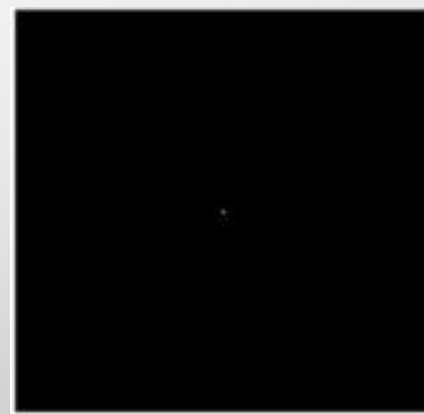
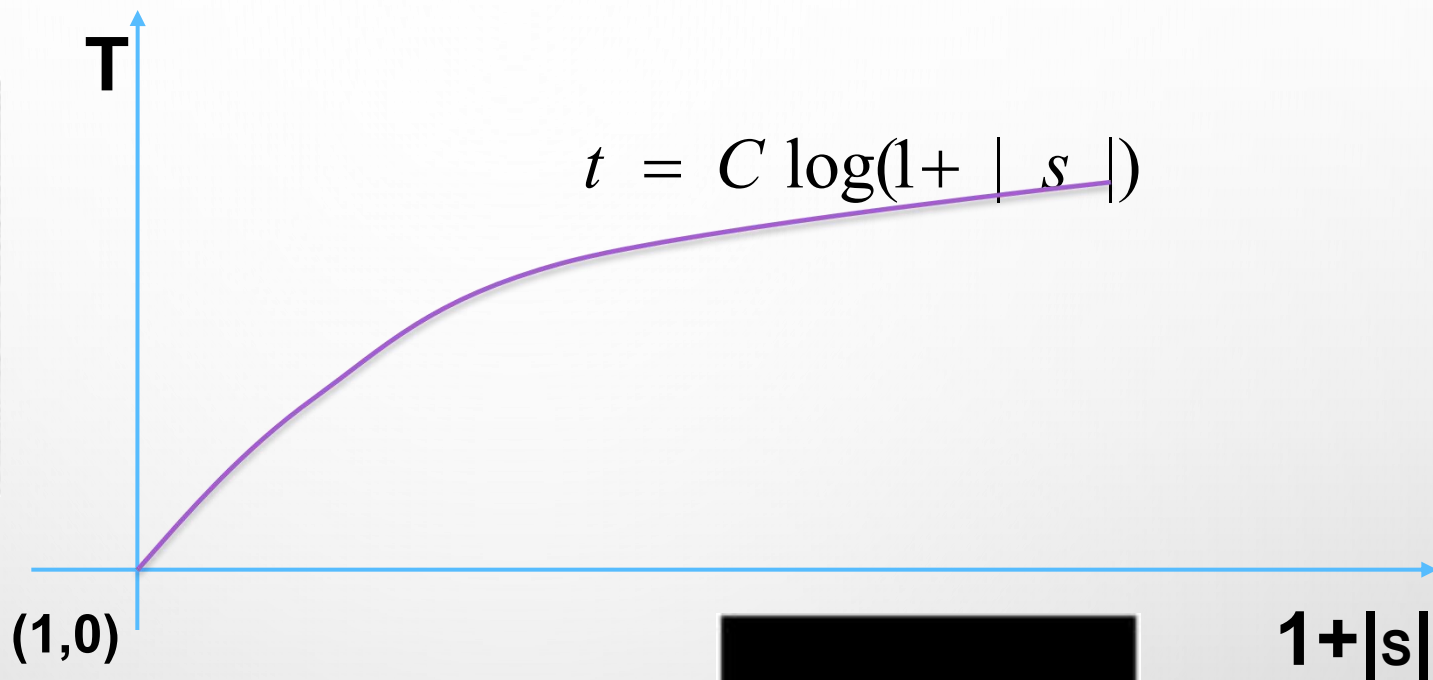
原图

$$C = 255 / \log(1 + 1.6 \times 10^6)$$

- 例：傅里叶谱的范围  $|S| \in [0, 1.6 \times 10^6]$ ，显示时使用对数变换。



对数变换得到的图



原图

$$C = 255 / \log(1 + 1.6 \times 10^6)$$



# 直接灰度变换

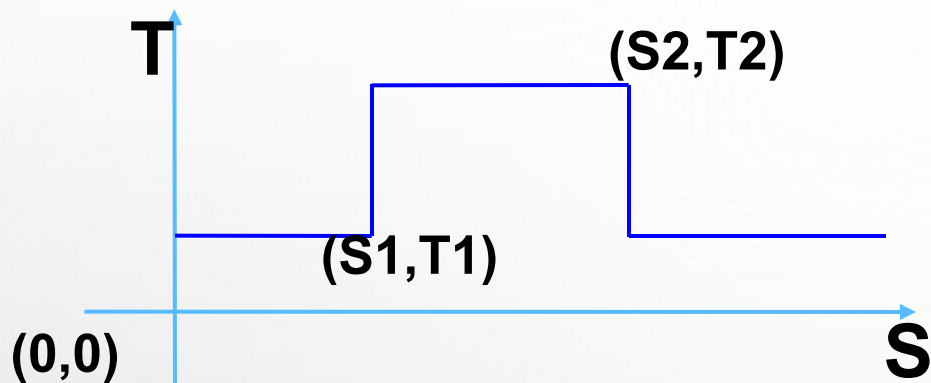
## 4、灰度切割

灰度切割的目的是增强特定范围的对比度，用来突出图像中特定灰度范围的亮度。

# 直接灰度变换

## 4、灰度切割

一种方法：是对感兴趣的灰度级以较大的灰度值  $T_2$  显示而对另外的灰度级则以较小的灰度值  $T_1$  来显示。



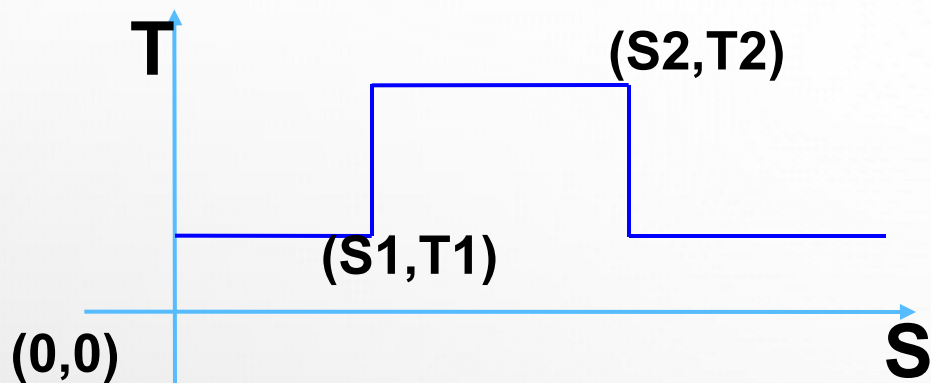
$$t = \begin{cases} t_2 & s_1 \leq s \leq s_2 \\ t_1 & \text{其它} \end{cases}$$

灰度切割的目的是增强特定范围的对比度，用来突出图像中特定灰度范围的亮度。

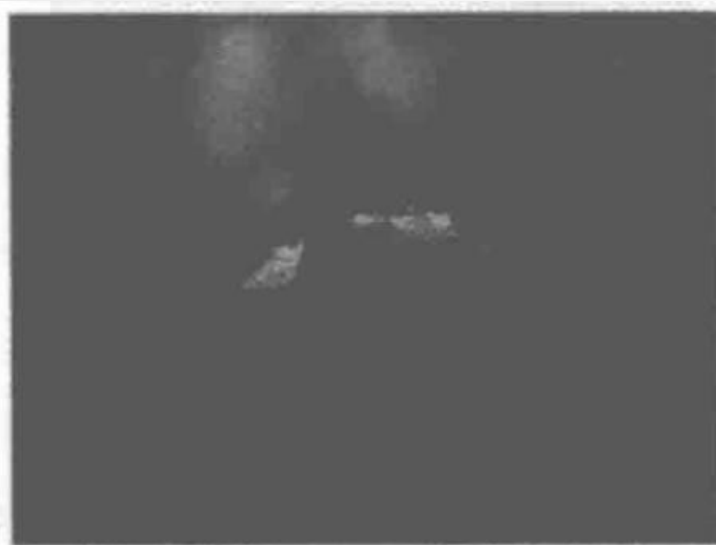
# 直接灰度变换

## 4、灰度切割

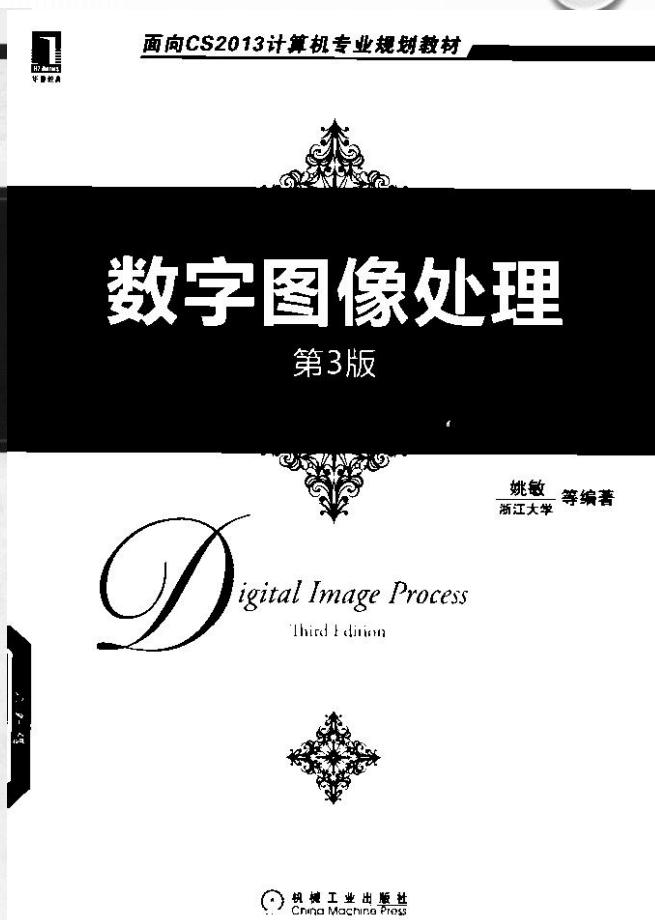
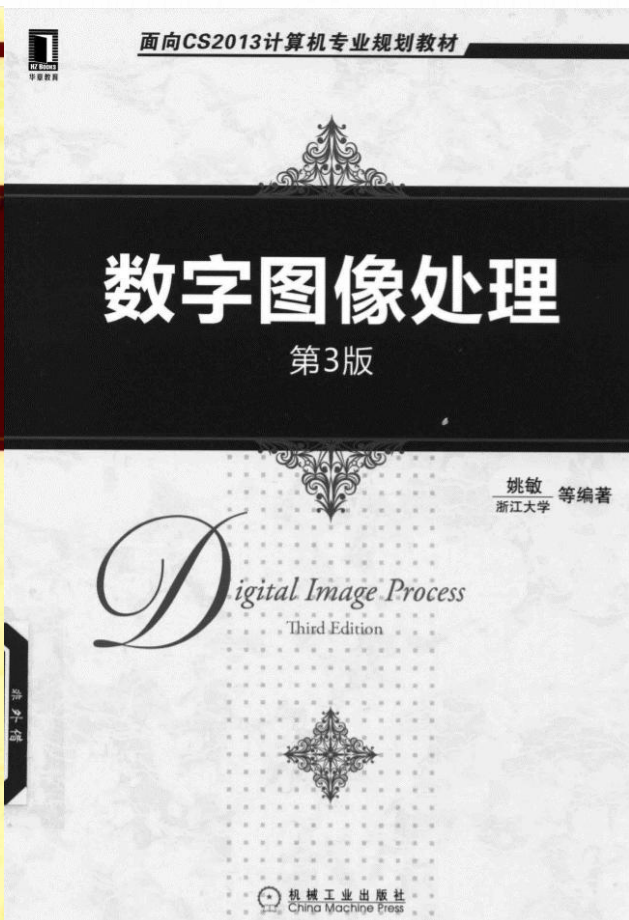
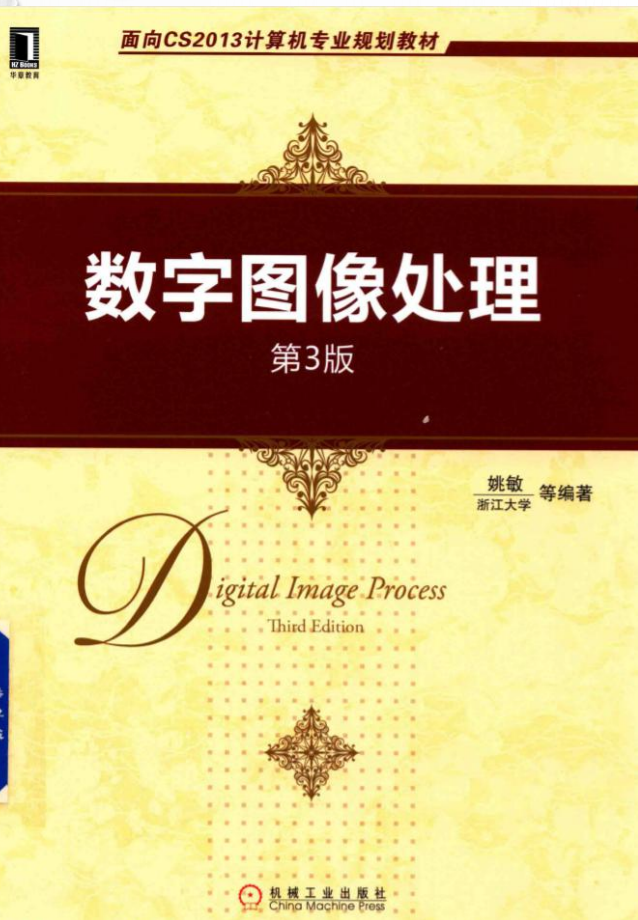
一种方法：是对感兴趣的灰度级以较大的灰度值  $T_2$  显示而对另外的灰度级则以较小的灰度值  $T_1$  来显示。



$$t = \begin{cases} t_2 & s_1 \leq s \leq s_2 \\ t_1 & \text{其它} \end{cases}$$



灰度切割的目的是增强特定范围的对比度，用来突出图像中特定灰度范围的亮度。

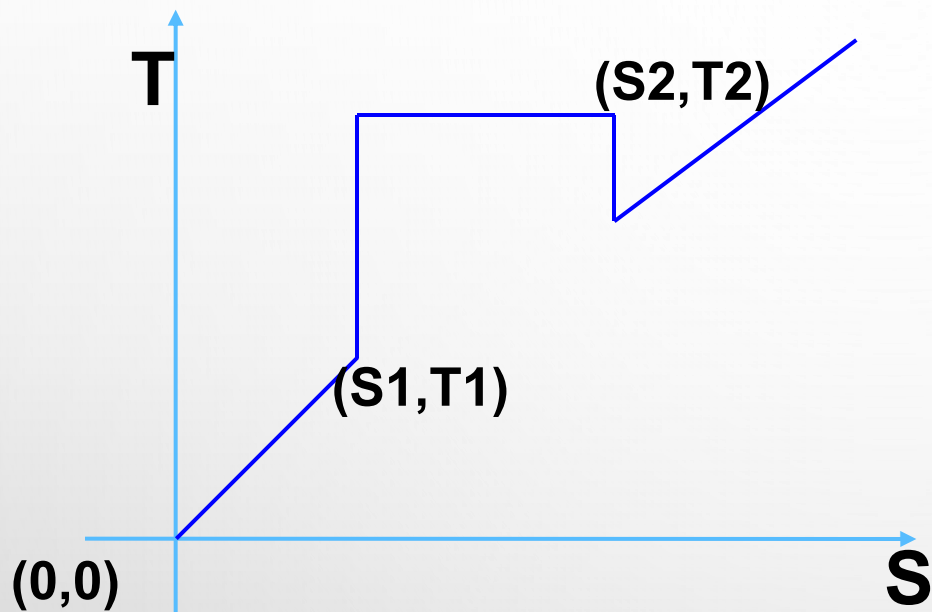


对于教材封面图像先将其进行256级灰度变换，然后将小于150的灰度值以0来显示，将大于等于150的灰度值以255来显示，得到右侧图片。



# 直接灰度变换

## 4、灰度切割



另一种方法：对感兴趣的灰度级以较大的灰度值进行显示而其他的灰度级则保持不变。

$$t = \begin{cases} t_2 & s_1 \leq s \leq s_2 \\ s & \text{其它} \end{cases}$$

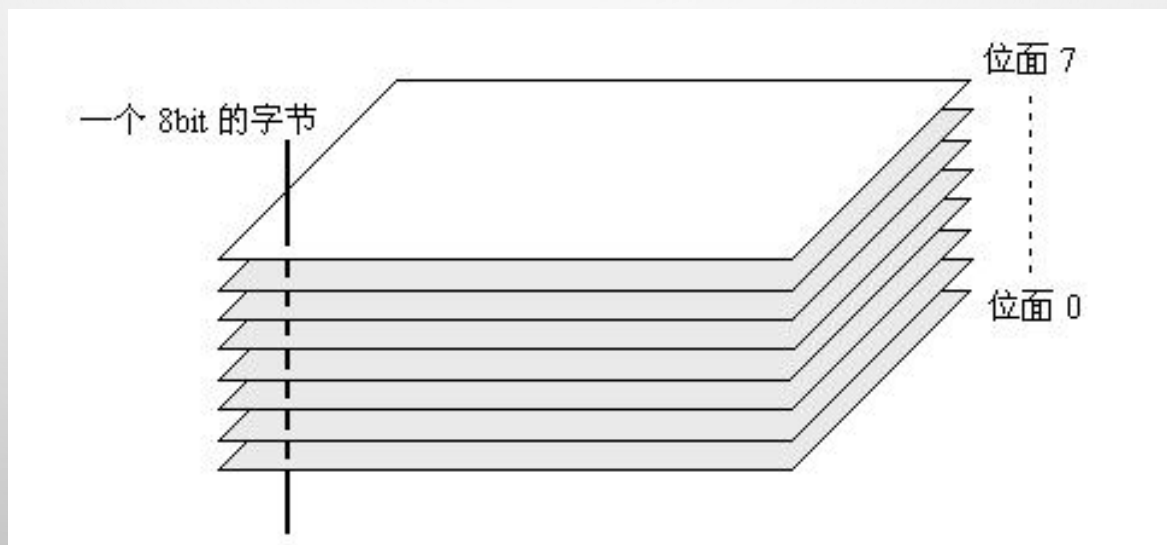
灰度切割的目的是增强特定范围的对比度，用来突出图像中特定灰度范围的亮度。



# 直接灰度变换

## 5、位图切割

8BIT图像有8个位面，一般用位面0表示最低位面，位面7表示最高位面，如图所示。借助图像的位面表示形式可通过对图像特定位面的操作来达到对图像的增强效果。



# 数字图像处理

第3版

姚敏 等编著  
浙江大学

*D*igital Image Process  
Third Edition

机械工业出版社  
China Machine Press

# 数字图像处理

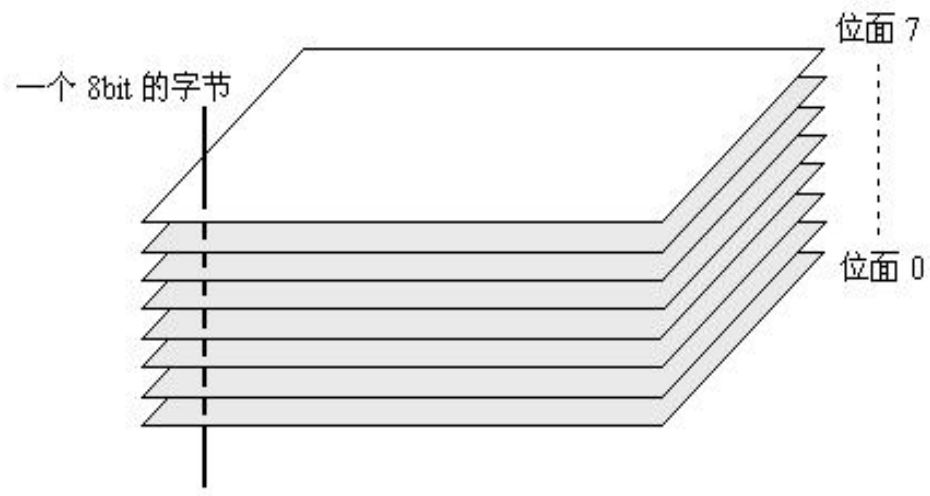
第3版

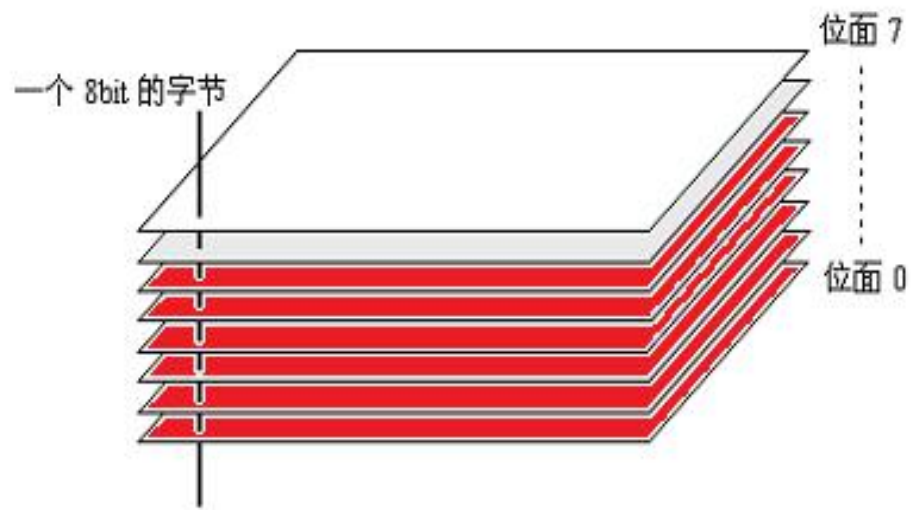
姚敏 等编著  
浙江大学

*D*igital Image Process  
Third Edition

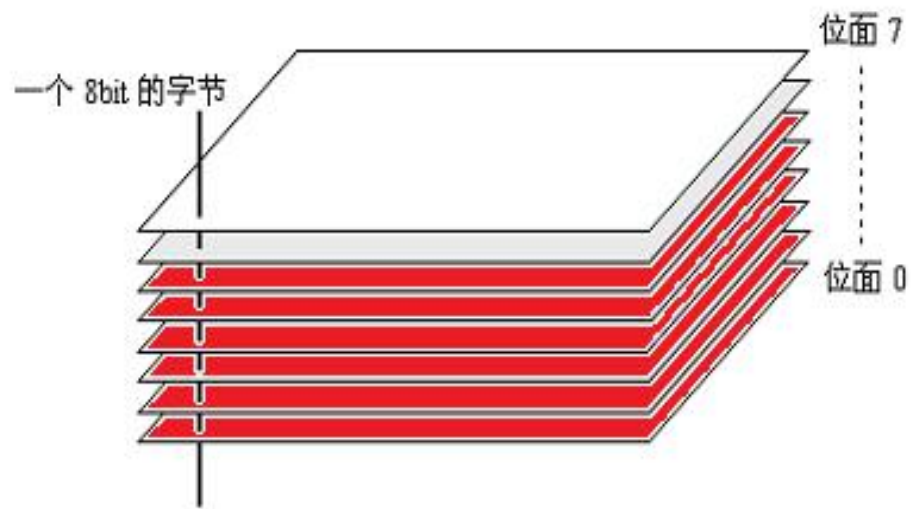
机械工业出版社  
China Machine Press

左侧为教材封面图像的256级灰度变换图，右侧为该灰度变换图位面7的切割图。

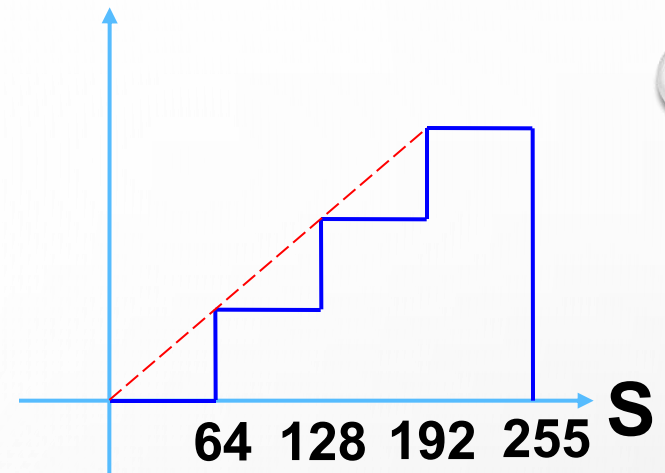




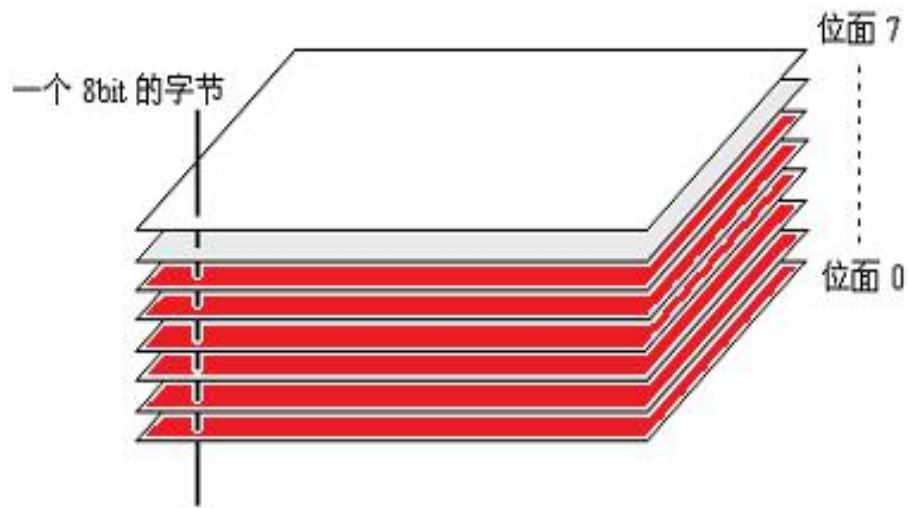
去掉位面0、1、2、3、4、5



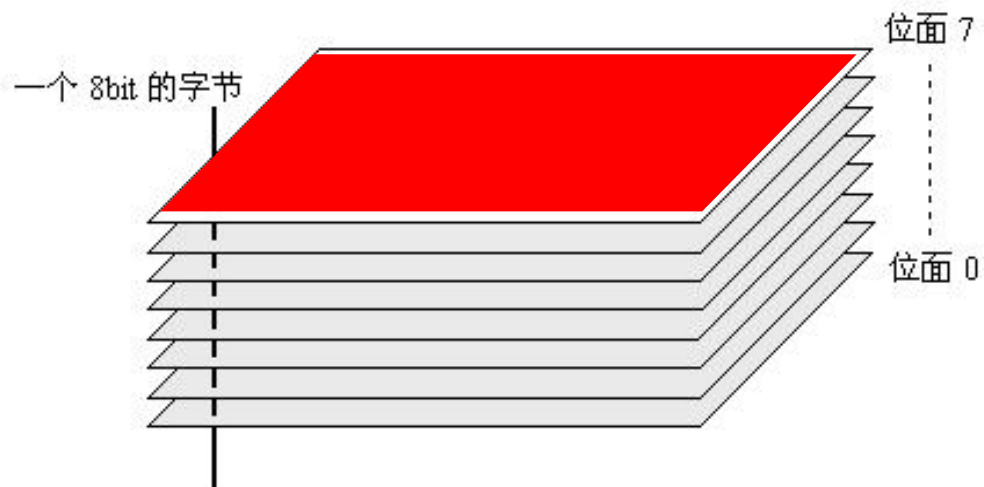
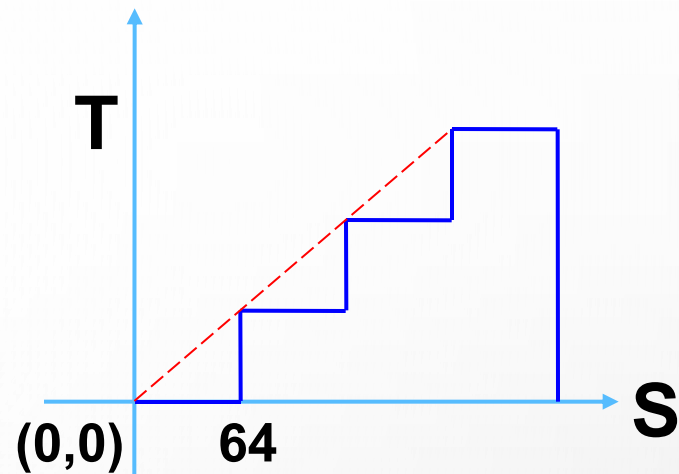
去掉位面 0、1、2、3、4、5



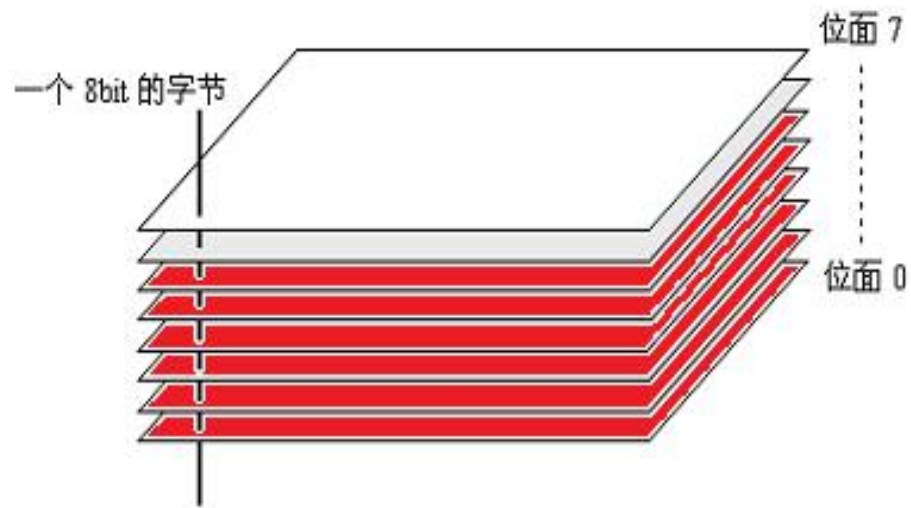




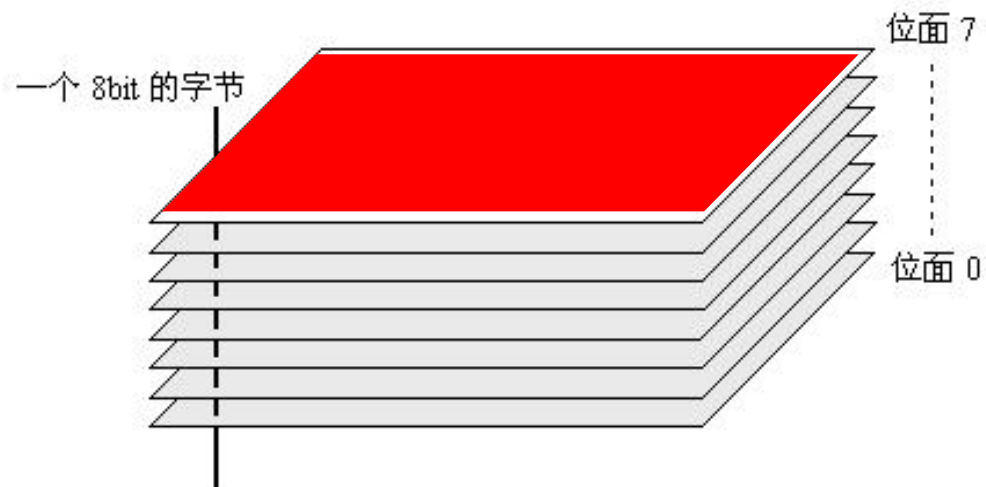
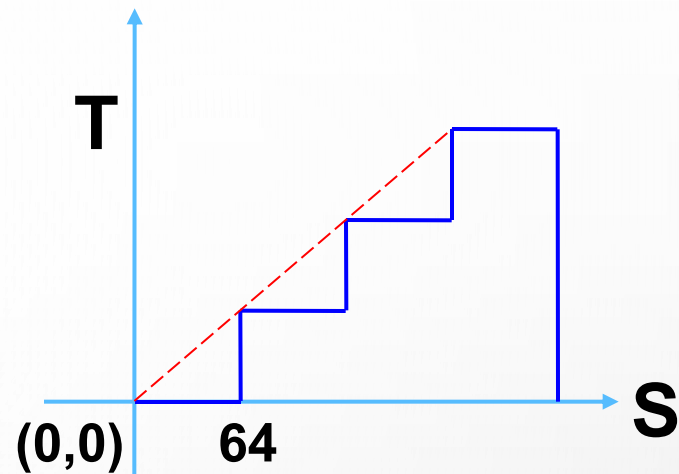
去掉位面 0、1、2、3、4、5



去掉位面 7



去掉位面 0、1、2、3、4、5



去掉位面 7

