



# 第2.5章 图像增强

图象增强是一种重要的图像处理的方法，它力求提高图象的质量，使之适合于人眼的观察判断或进一步的分析处理。

## 2.5.0 某些基本灰度变换

### 2.5.1 直方图修正

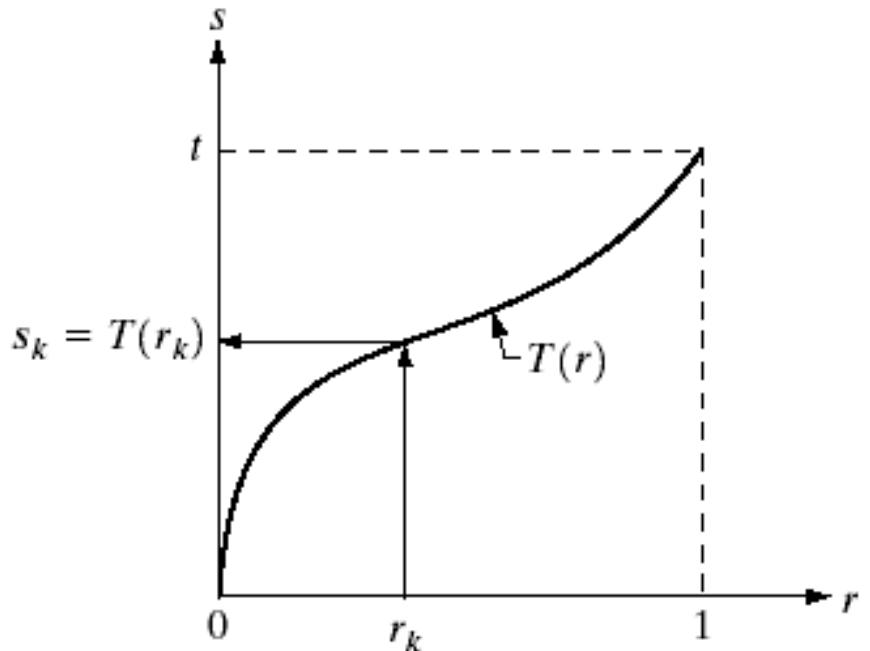
### 2.5.2 锐化

### 2.5.3 平滑

### 2.5.4 伪彩色

### 2.5.5 颜色迁移

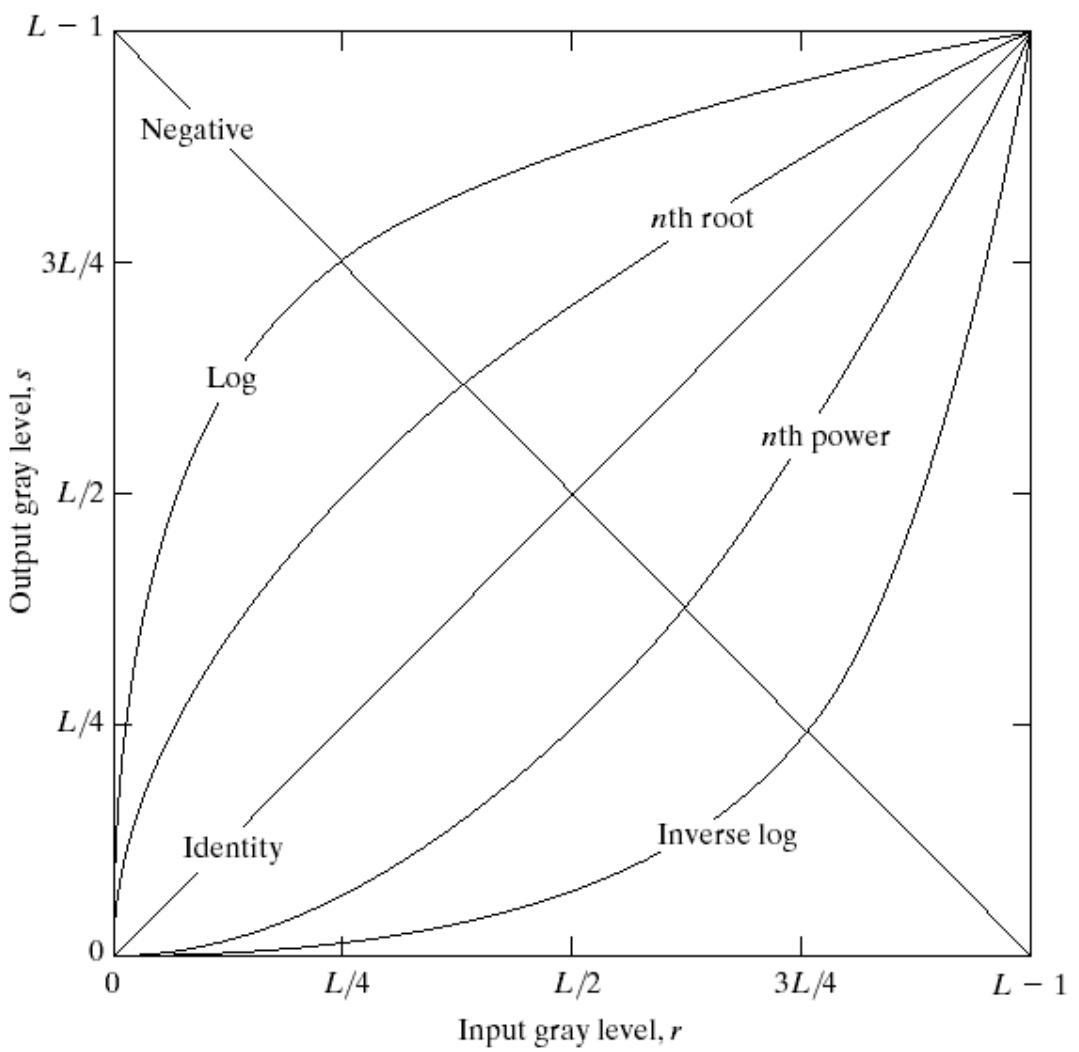
# 直接灰度映射原理



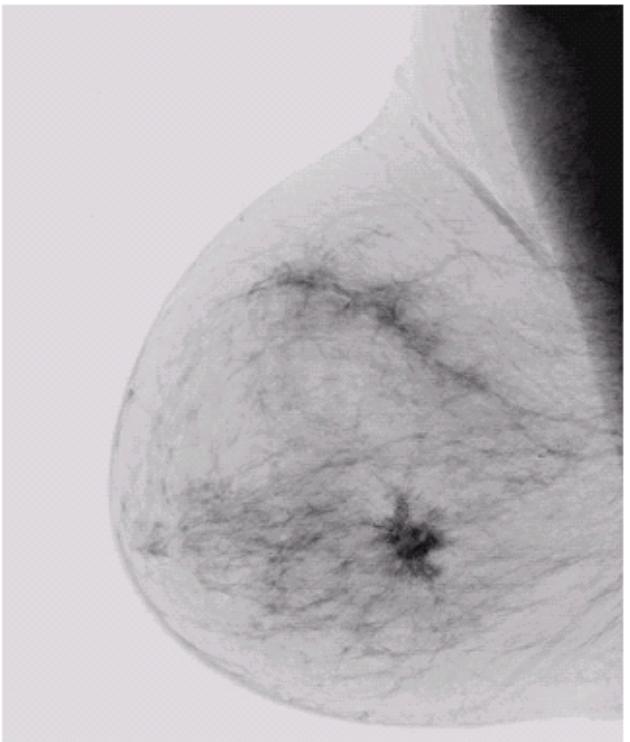
**FIGURE 3.16** A gray-level transformation function that is both single valued and monotonically increasing.

# 常见灰度变换

**FIGURE 3.3** Some basic gray-level transformation functions used for image enhancement.



# 图象求反



a b

**FIGURE 3.4**

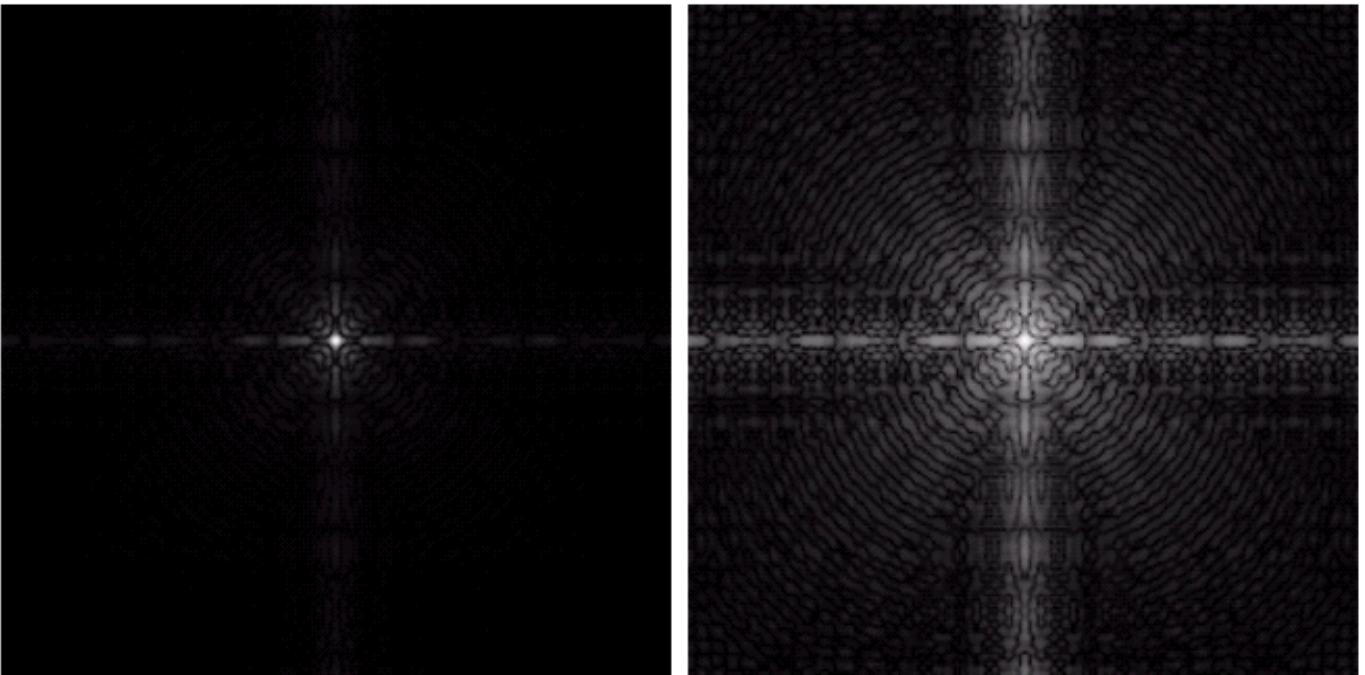
(a) Original digital mammogram.  
(b) Negative image obtained using the negative transformation in Eq. (3.2-1).  
(Courtesy of G.E. Medical Systems.)

# 对数变换

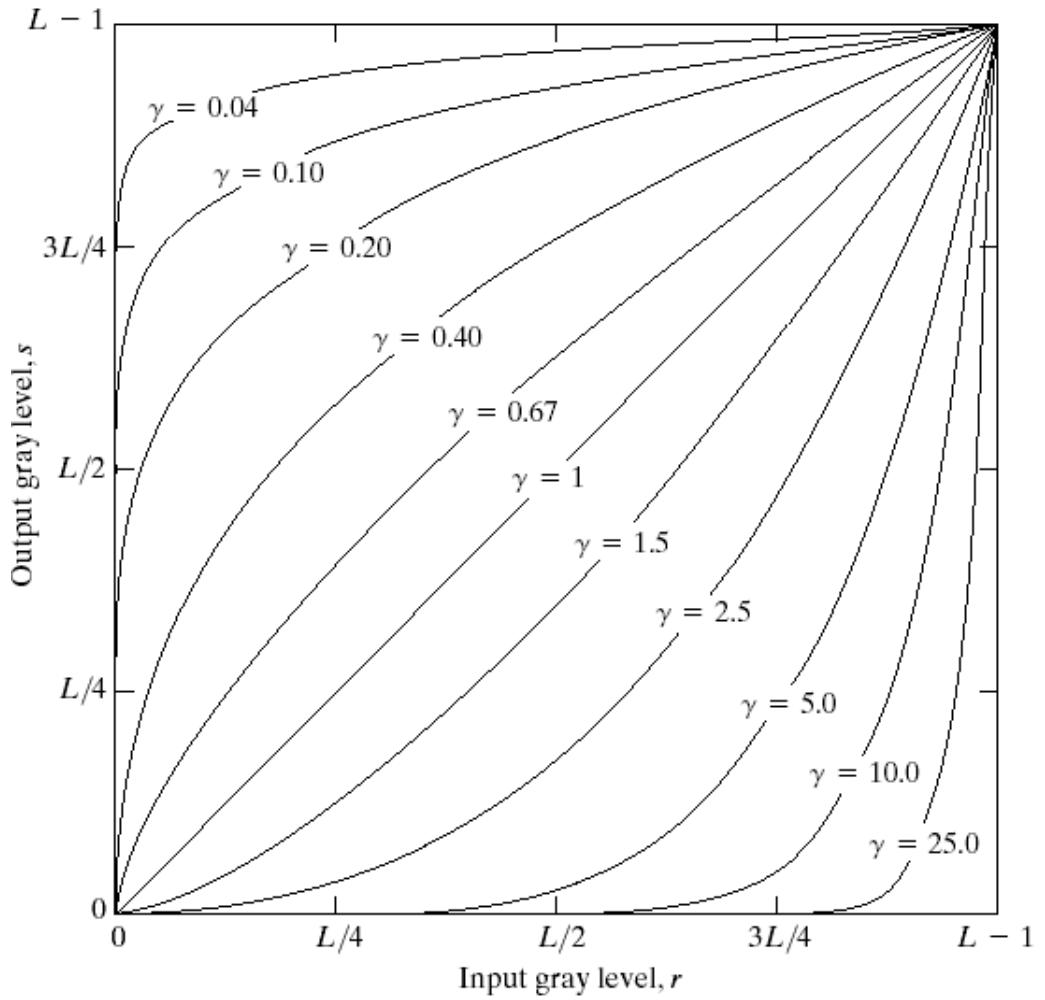
a b

**FIGURE 3.5**

(a) Fourier spectrum.  
(b) Result of applying the log transformation given in Eq. (3.2-2) with  $c = 1$ .



# 常见灰度变换—指数变换



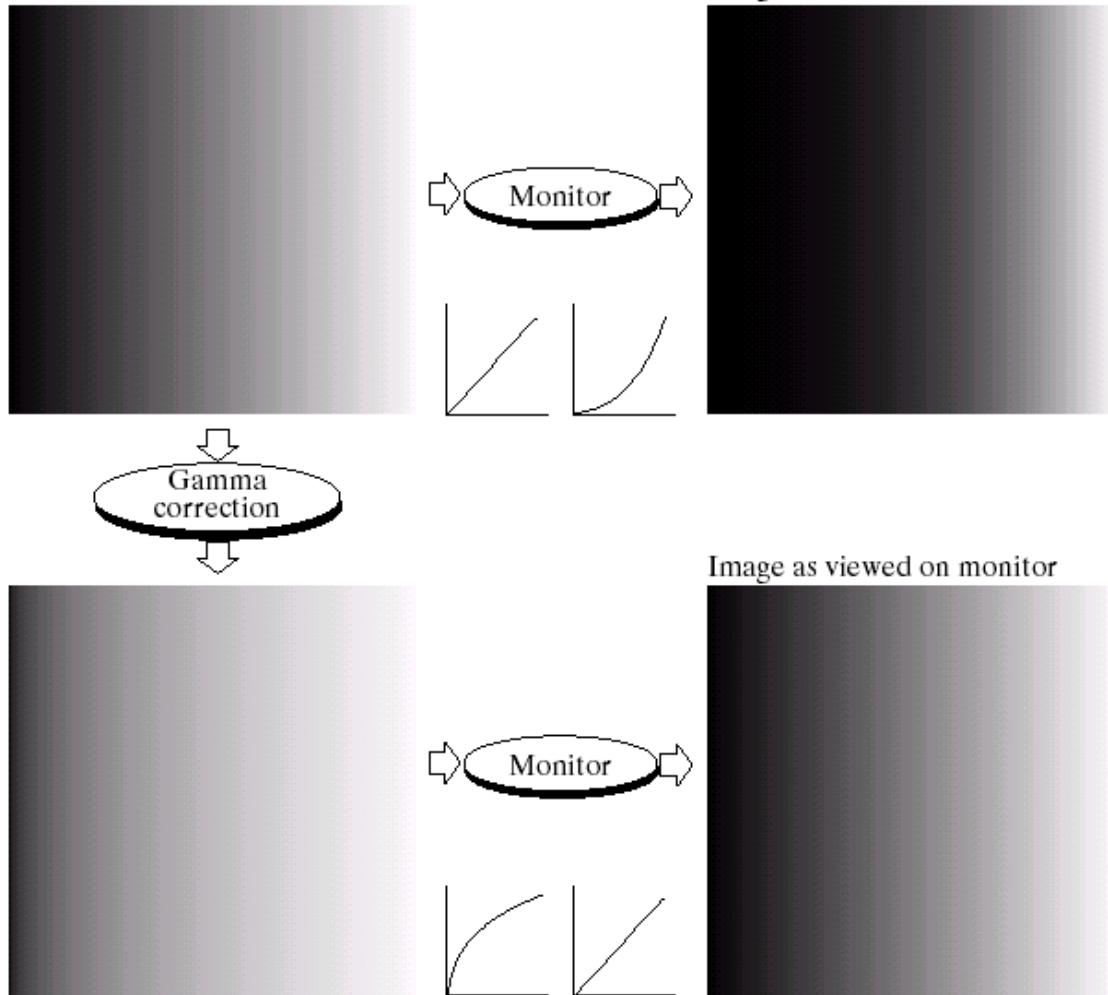
**FIGURE 3.6** Plots of the equation  $s = cr^\gamma$  for various values of  $\gamma$  ( $c = 1$  in all cases).

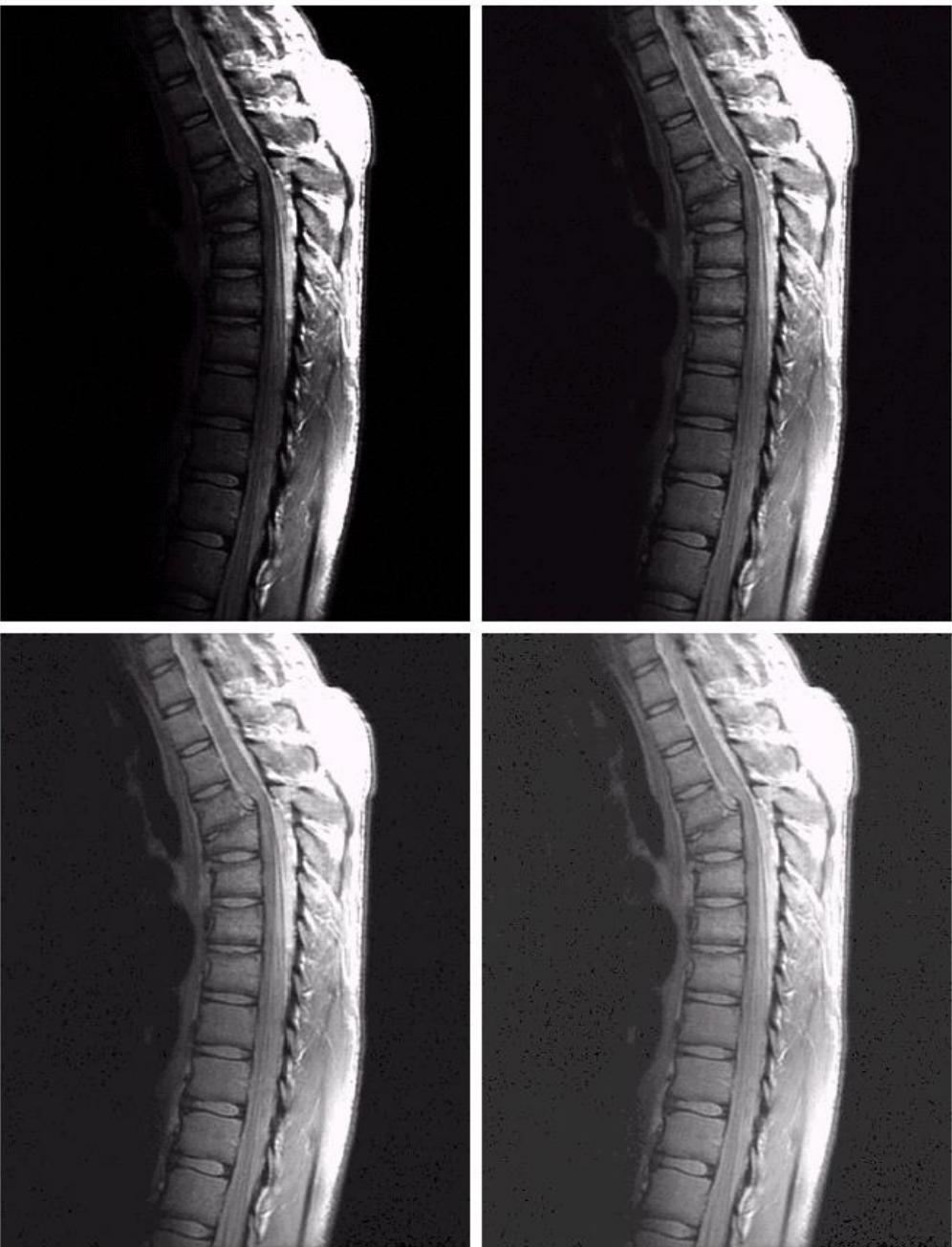
# 伽马校正

a b  
c d

**FIGURE 3.7**

- (a) Linear-wedge gray-scale image.
- (b) Response of monitor to linear wedge.
- (c) Gamma-corrected wedge.
- (d) Output of monitor.





a b  
c d

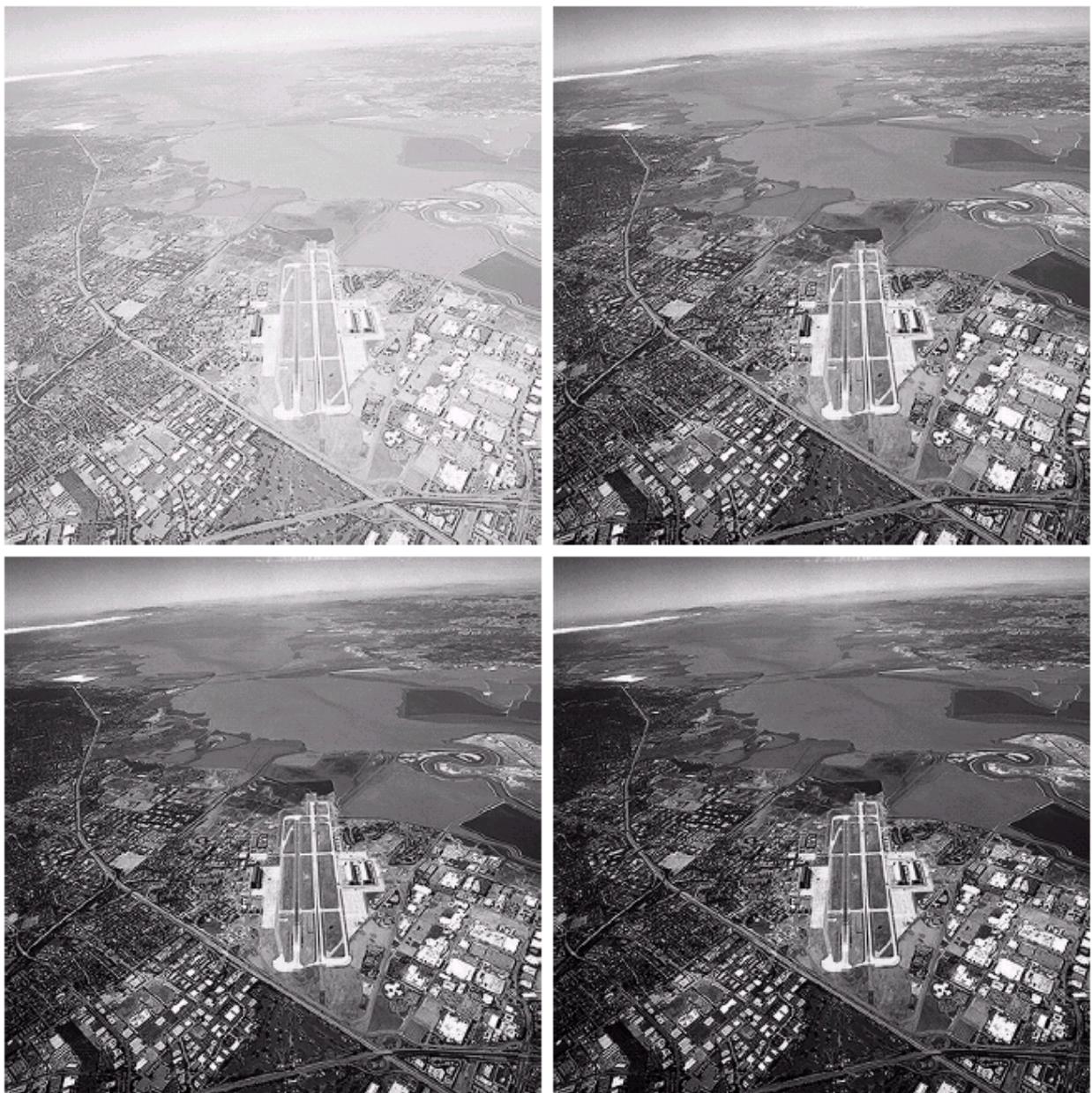
**FIGURE 3.8**

(a) Magnetic resonance (MR) image of a fractured human spine.  
(b)–(d) Results of applying the transformation in Eq. (3.2-3) with  $c = 1$  and  $\gamma = 0.6, 0.4$ , and  $0.3$ , respectively.  
(Original image for this example courtesy of Dr. David R. Pickens, Department of Radiology and Radiological Sciences, Vanderbilt University Medical Center.)

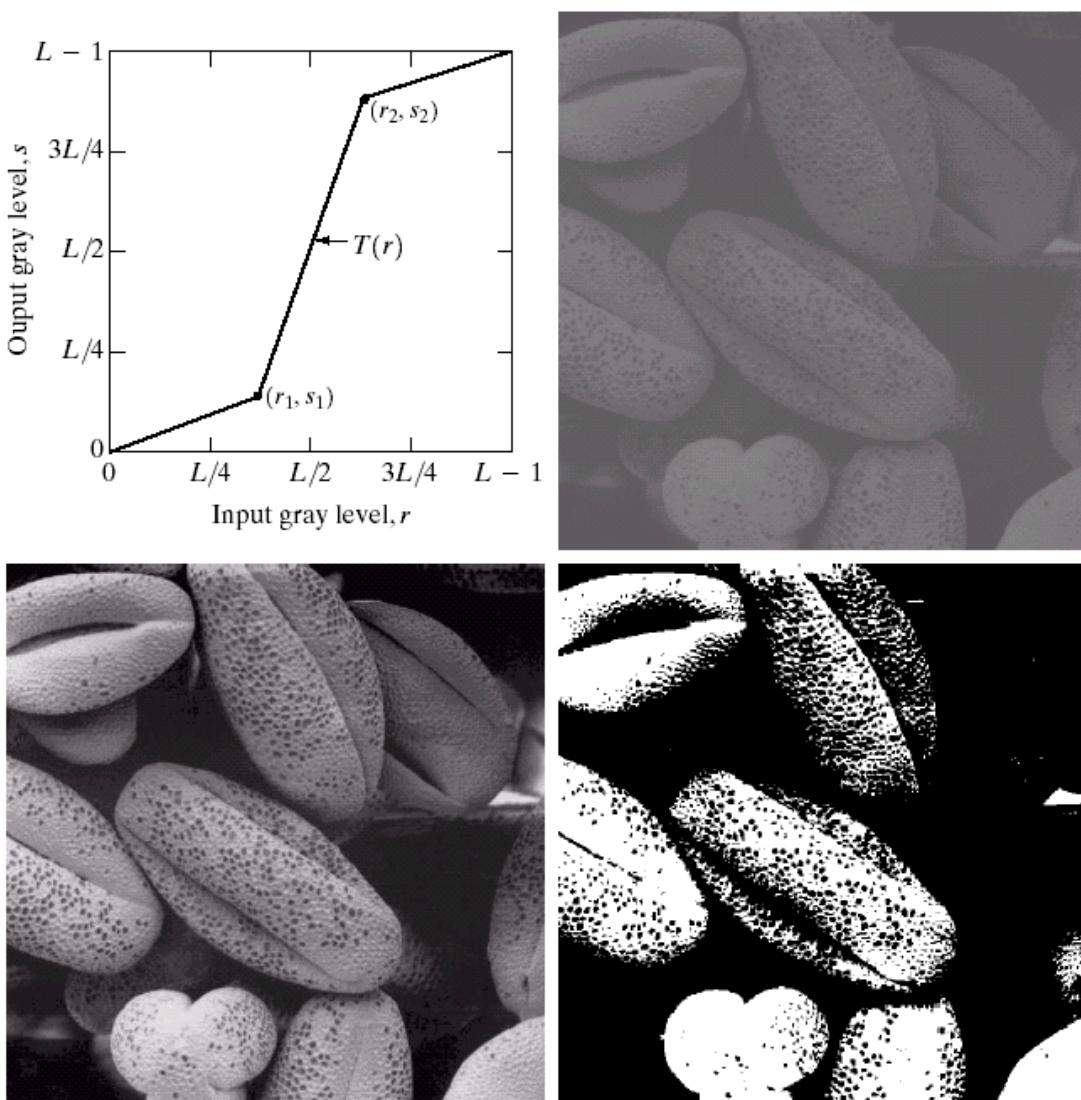
a b  
c d

**FIGURE 3.9**

(a) Aerial image.  
(b)–(d) Results of applying the transformation in Eq. (3.2-3) with  $c = 1$  and  $\gamma = 3.0, 4.0$ , and  $5.0$ , respectively. (Original image for this example courtesy of NASA.)



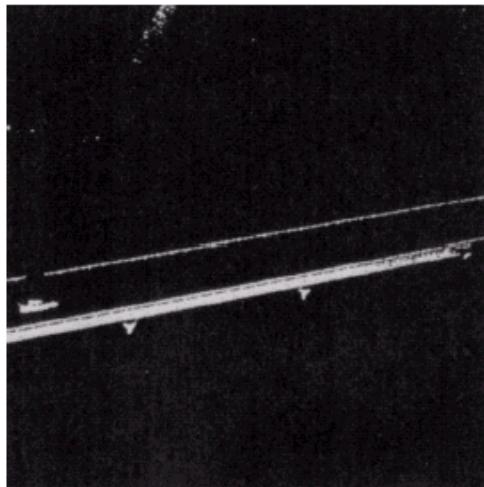
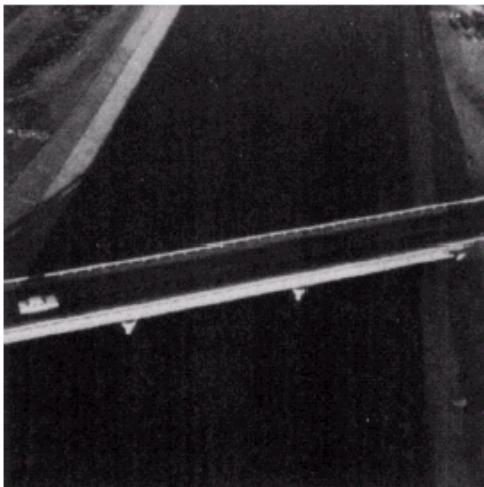
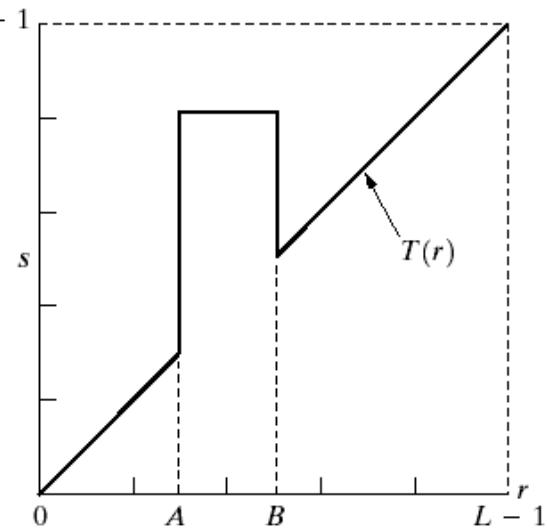
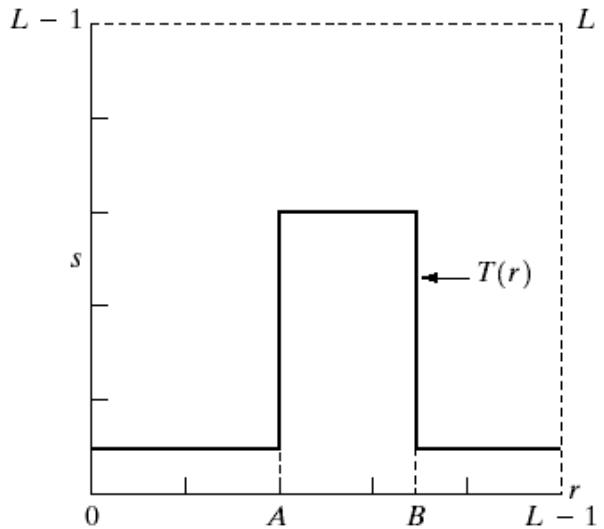
# 分段线性变换函数—对比度拉伸



a | b  
c | d

**FIGURE 3.10**  
Contrast stretching.  
(a) Form of transformation function.  
(b) A low-contrast image.  
(c) Result of contrast stretching.  
(d) Result of thresholding.  
(Original image courtesy of Dr. Roger Heady, Research School of Biological Sciences, Australian National University, Canberra, Australia.)

# 分段线性变换函数—灰度级切分

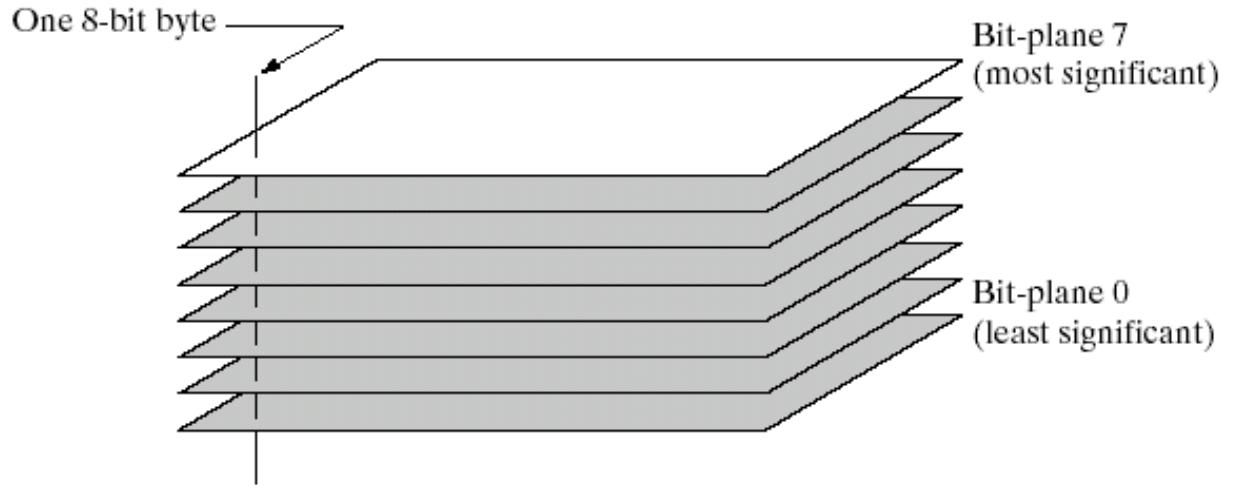


a	b
c	d

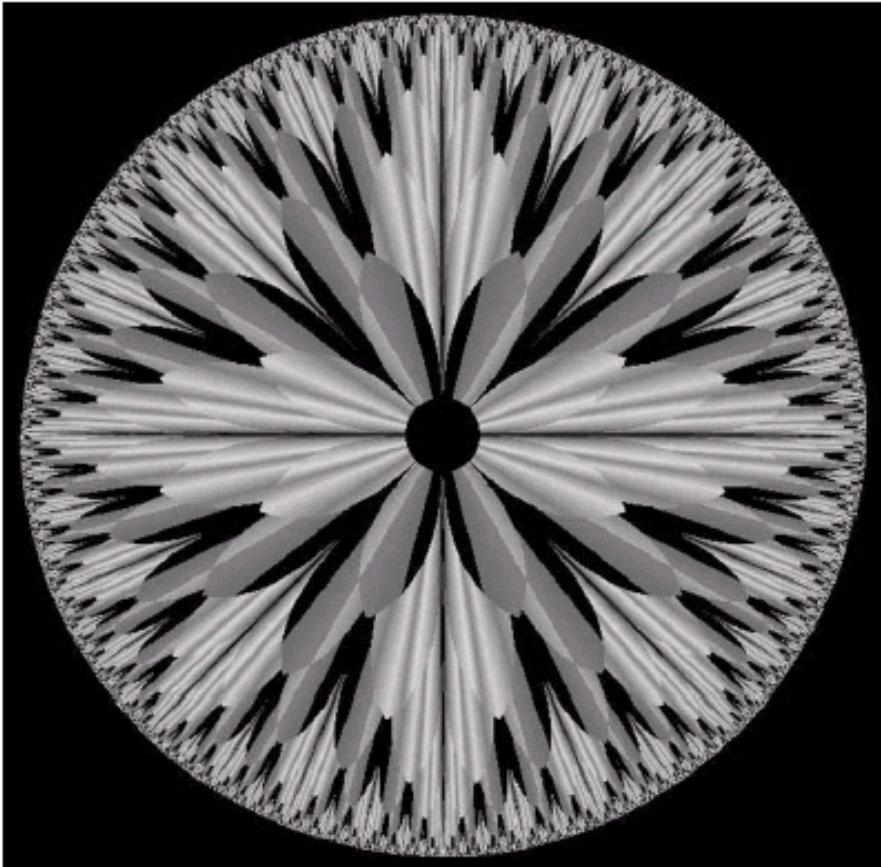
**FIGURE 3.11**

- (a) This transformation highlights range  $[A, B]$  of gray levels and reduces all others to a constant level.
- (b) This transformation highlights range  $[A, B]$  but preserves all other levels.
- (c) An image.
- (d) Result of using the transformation in (a).

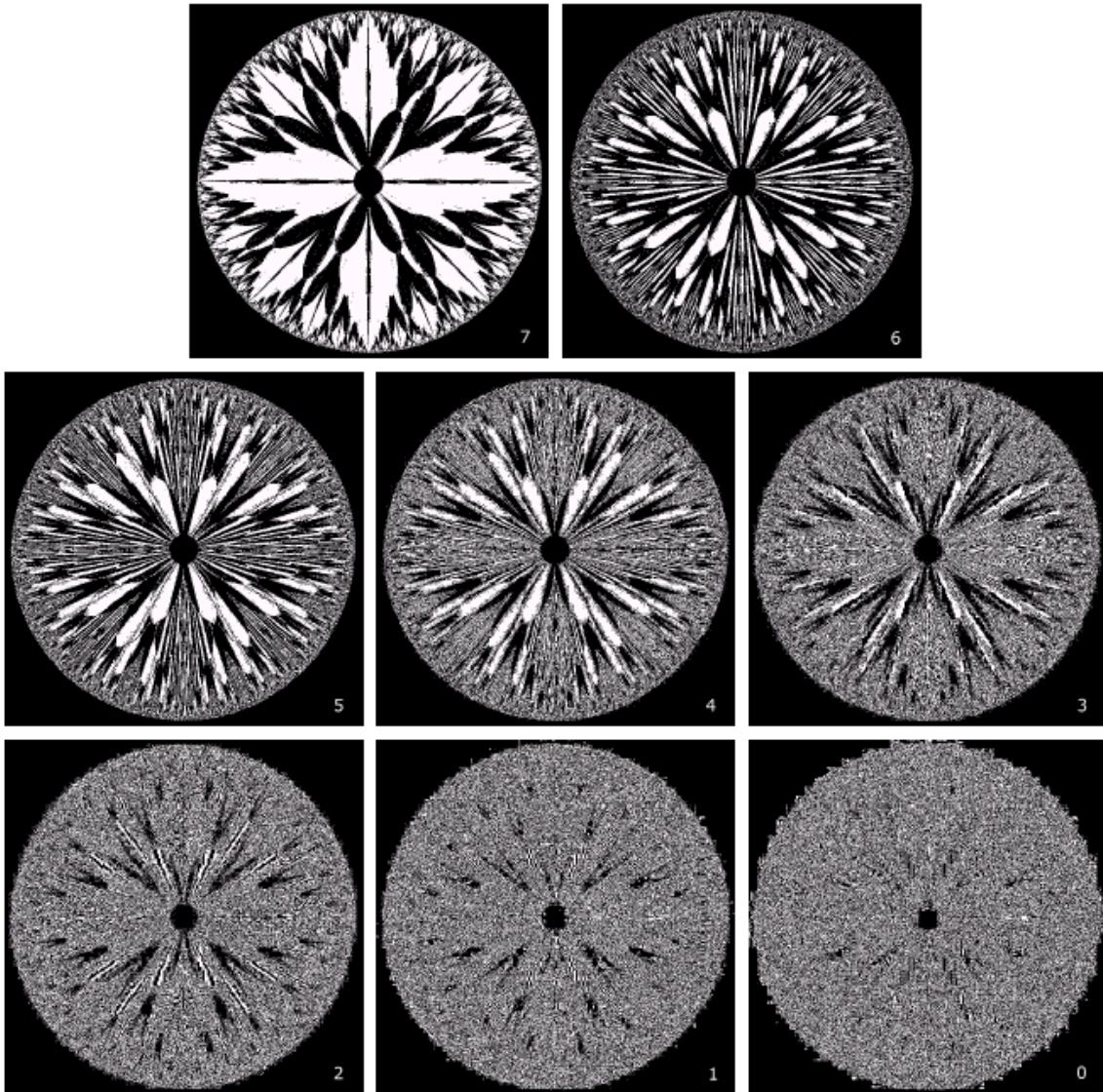
# 比特面分割



**FIGURE 3.12**  
Bit-plane  
representation of  
an 8-bit image.



**FIGURE 3.13** An 8-bit fractal image. (A fractal is an image generated from mathematical expressions). (Courtesy of Ms. Melissa D. Binde, Swarthmore College, Swarthmore, PA.)



**FIGURE 3.14** The eight bit planes of the image in Fig. 3.13. The number at the bottom, right of each image identifies the bit plane.



# 灰度直方图

## □ 灰度直方图

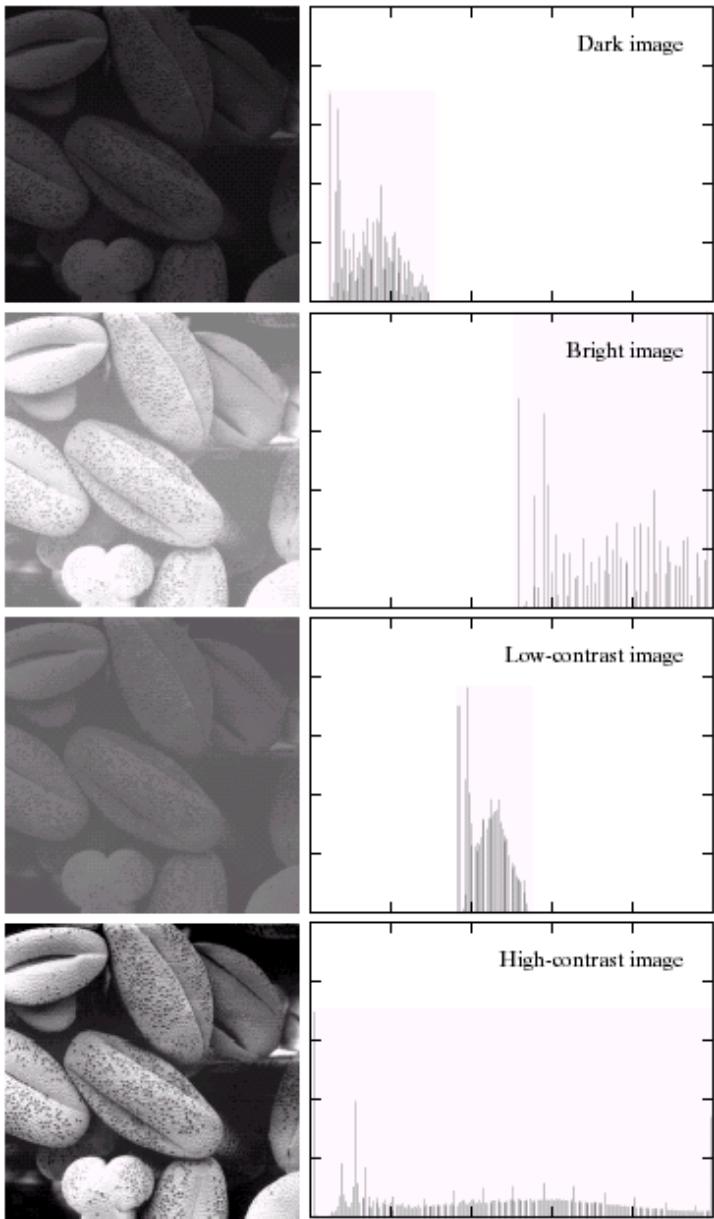
- 是一种灰度级的函数，它表示数字图象中每一灰度级与该灰度出现的频数之间的对应关系。
- 一幅特定的图有唯一的直方图，但反之不成立。
- 直方图在一定程度上可以反映图像的状况，例如：有时可以根据直方图确定分割物体和背景的边界。

## □ 基于直方图的灰度变换

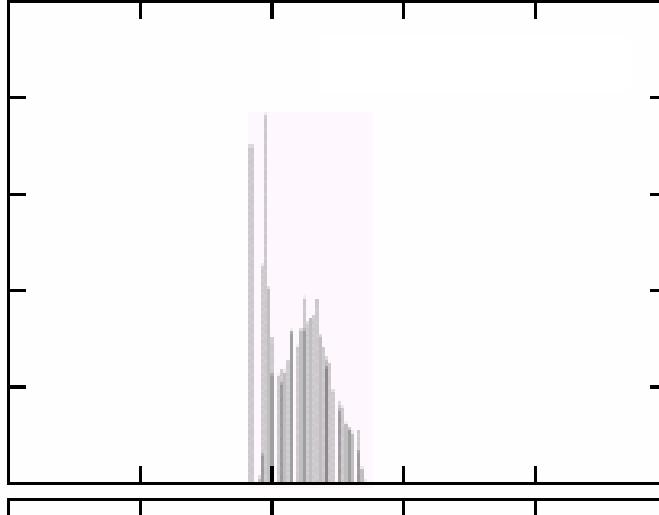
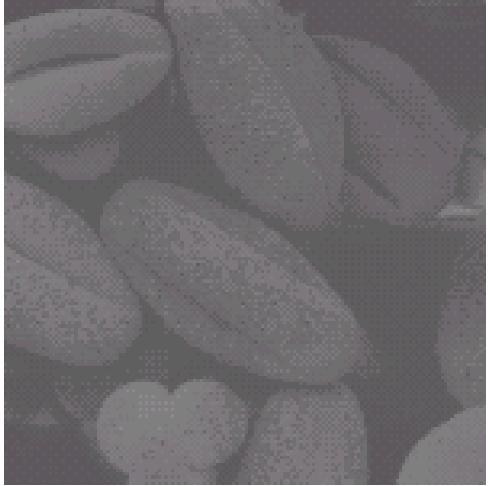
- 不改变像素的位置，只改变像素的灰度值。

## □ 通过修改直方图可以达到增强图象的对比度、使感兴趣的部分看得更清楚。

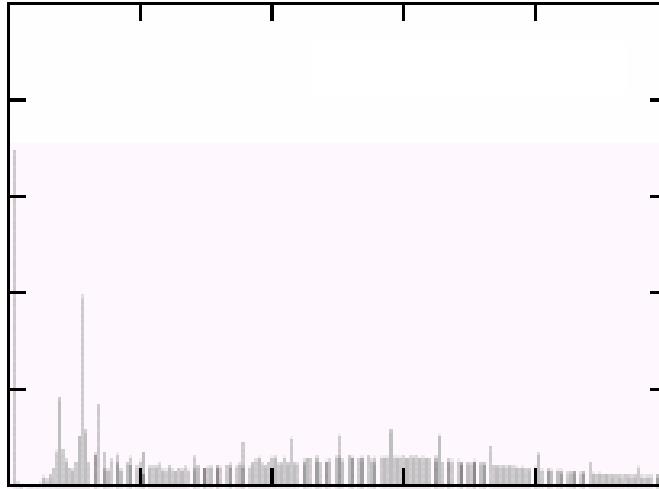
## □ 常用的有一维灰度特征直方图，也有其他特征直方图，还有多维直方图。



## 2.5.1 直方图修正



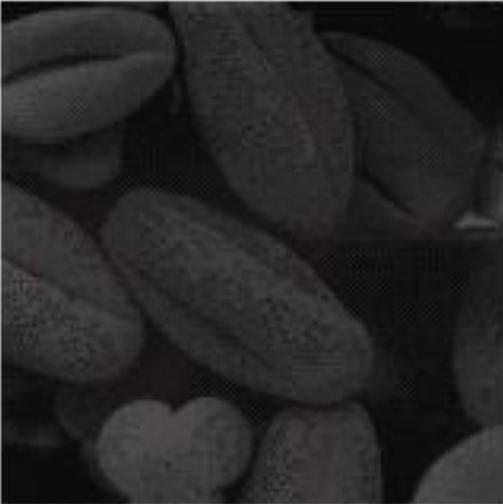
低对比度



高对比度

# 1. 直方图均衡

- 是基于直方图的灰度变换中常用的一种处理方法。
- 通常，用直方图拉伸来执行直方图均衡处理。





## 2. 直方图匹配（直方图规定化）

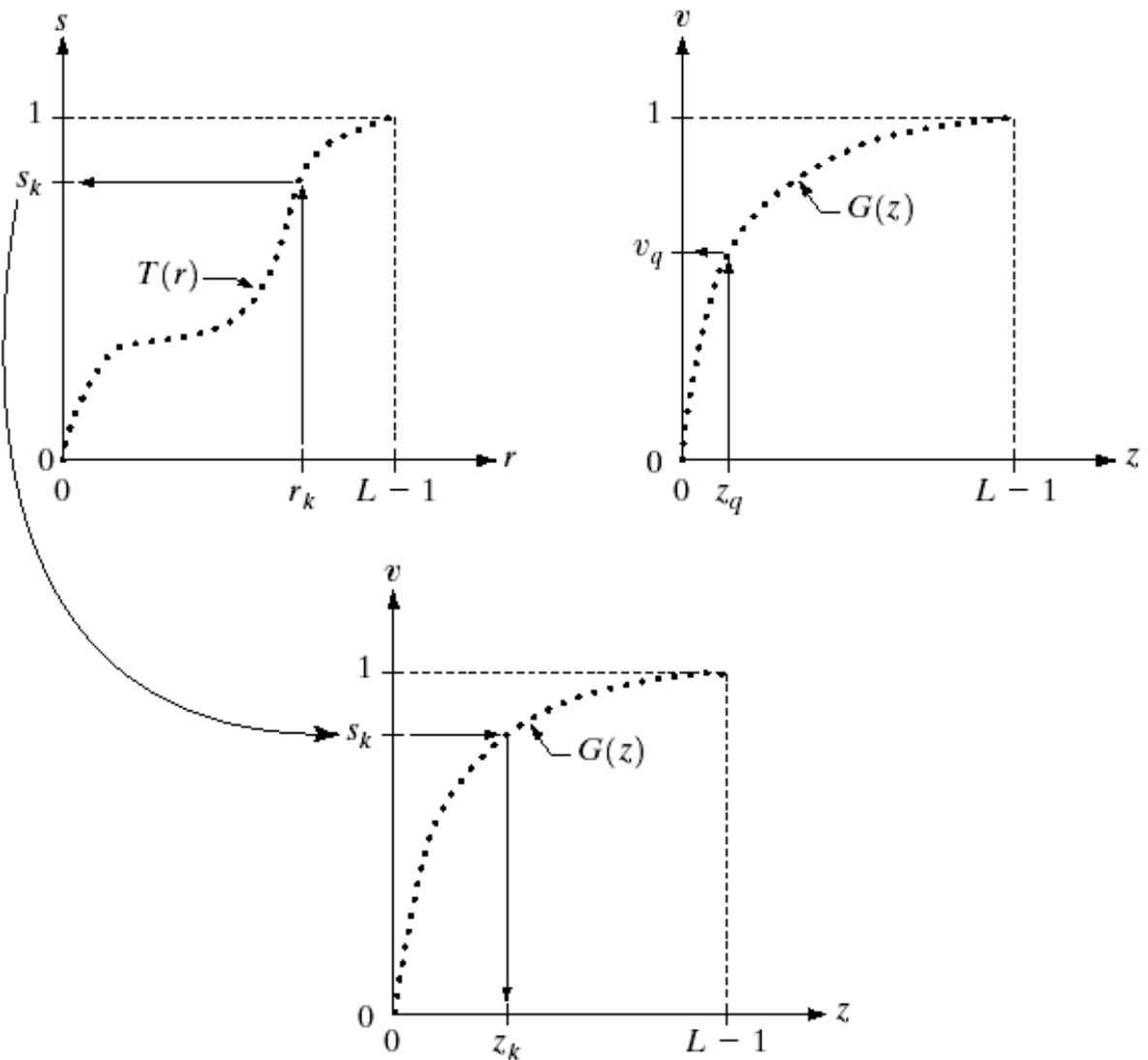
- 借助直方图变换实现规定/特定的灰度映射
- 三个步骤
  1. 对原始直方图进行灰度均衡化
  2. 规定需要的直方图，计算能使规定直方图均衡化的变换
  3. 将原始直方图对应映射到规定直方图

## 2. 直方图匹配（直方图规范化）

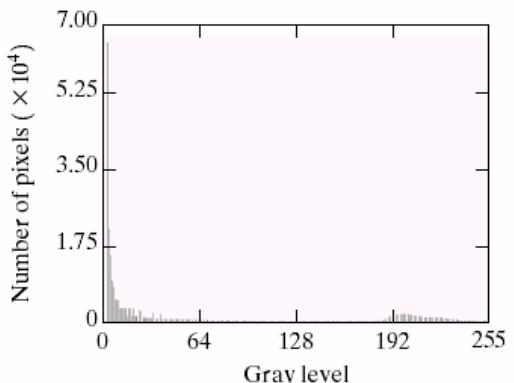
a b  
c

**FIGURE 3.19**

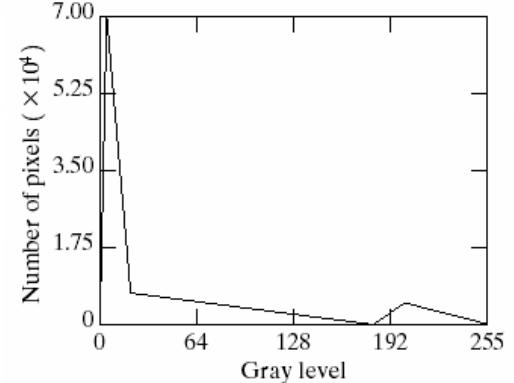
- (a) Graphical interpretation of mapping from  $r_k$  to  $s_k$  via  $T(r)$ .
- (b) Mapping of  $z_q$  to its corresponding value  $v_q$  via  $G(z)$ .
- (c) Inverse mapping from  $s_k$  to its corresponding value of  $z_k$ .



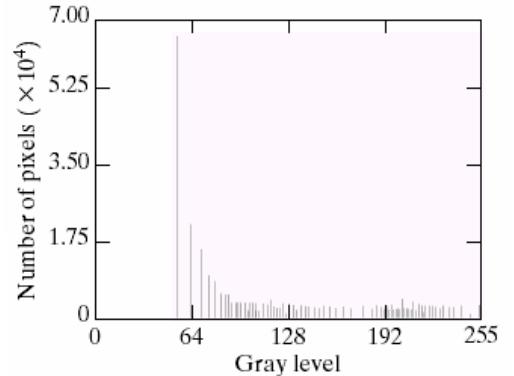
# 直方图匹配



原图



处理后  
图像



原图直  
方图

匹配目  
标直方  
图

处理后  
直方图



# 直方图均衡 vs. 直方图匹配

## □ 直方图均衡：

- 自动增强
- 效果不易控制
- 总得到全图增强的结果

## □ 直方图匹配：

- 有选择地增强
- 须给定需要的直方图
- 可特定增强的结果

## □ 直方图均衡可以看作是匹配直方图为均匀分布的直方图匹配



## 2.5.2 锐化

- 图象锐化针对常见的图像模糊、边缘不清晰所采取的处理方法，它能加强图象的轮廓，使图象看起来比较清楚。
  
- 图象锐化的方法，总体上可分为两大类：
  - 频率域处理
  - 空间域处理



# 1、空间频率域高通滤波法

- 图象的轮廓是灰度陡然变化的部分，包含着丰富 的空间高频分量，高通滤波增强高频分量，可使 轮廓突出。

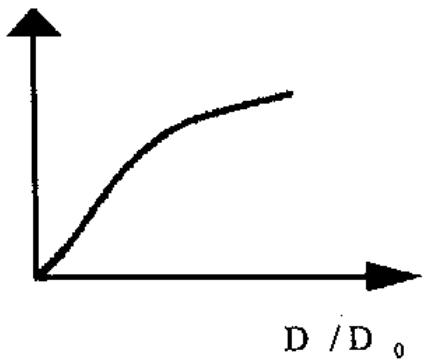
$$G(u, v) = H(u, v)F(u, v)$$

# 常用的高通滤波器

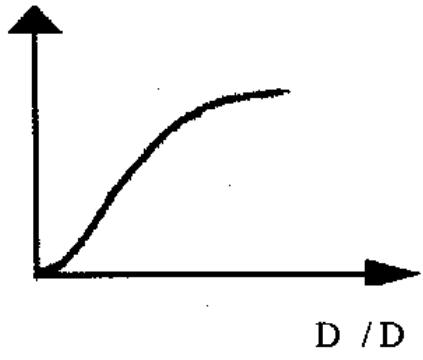
- 巴特沃斯(Butterworth)高通滤波器
- 指数形高通滤波器
- 梯形高通滤波器

常用的高通滤波器一维图例：

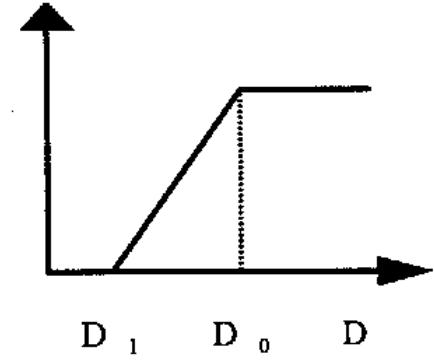
$H(u, v)$



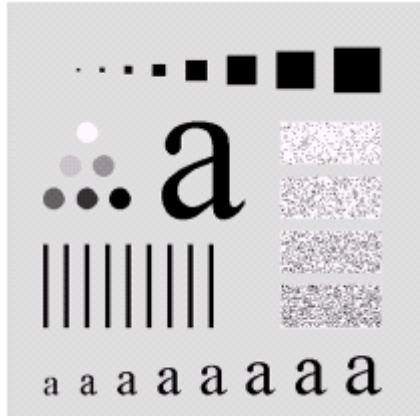
$H(u, v)$



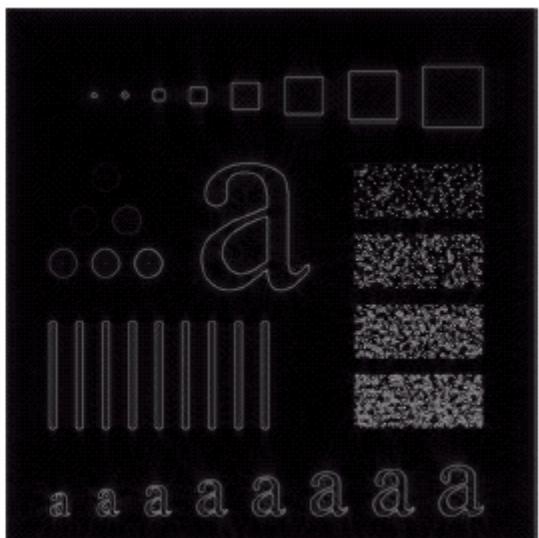
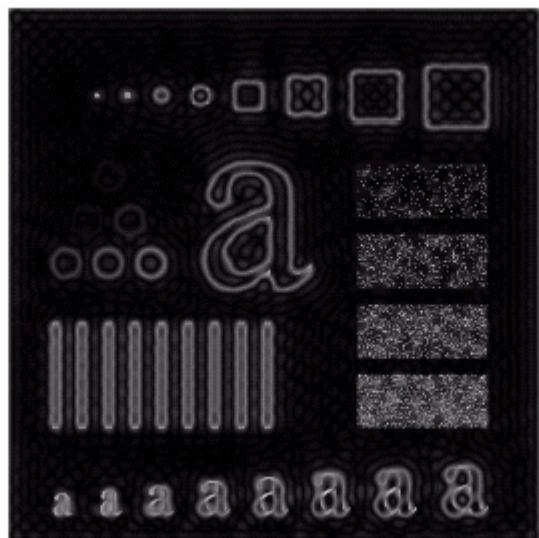
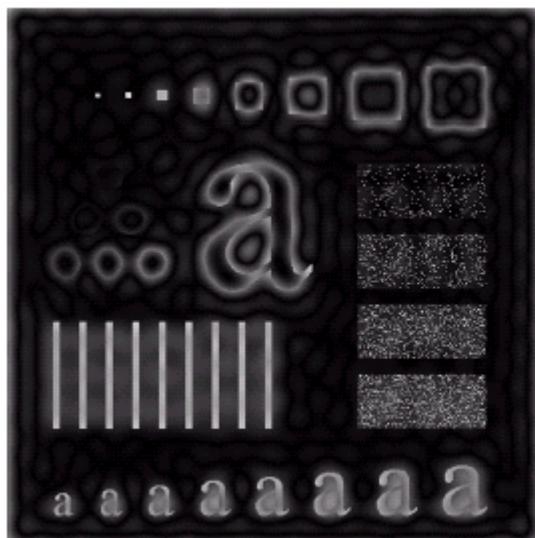
$H(u, v)$



# 理想高通滤波器

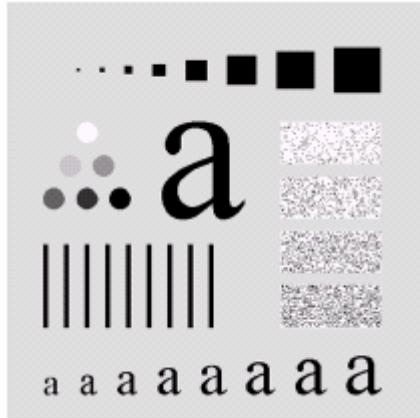


$$H(u, v) = \begin{cases} 0 & D(u, v) \leq D_0 \\ 1 & D(u, v) > D_0 \end{cases}$$

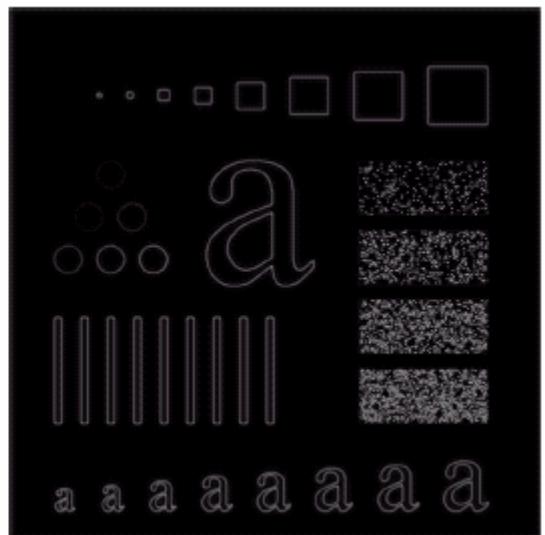
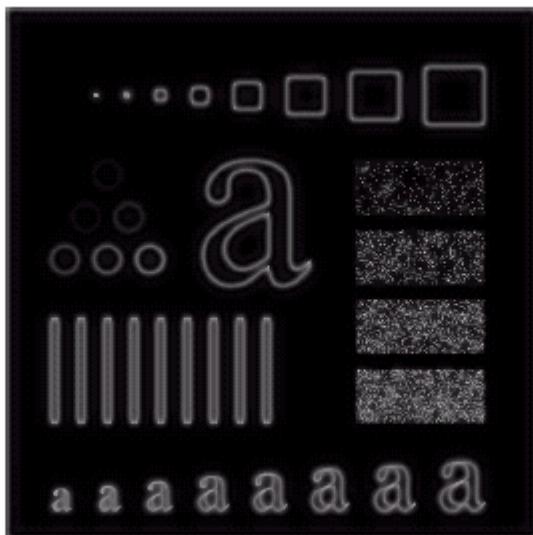
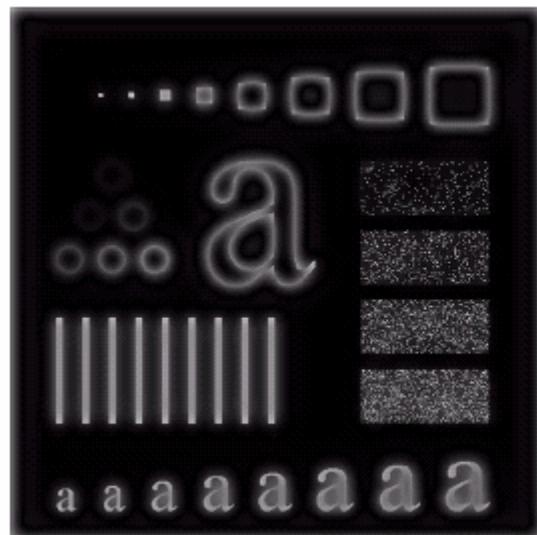


从左至右 $D_0$ 分别等于15, 30和80

# 巴特沃思高通滤波器

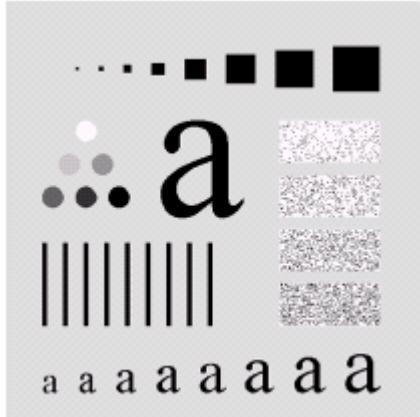


$$H(u, v) = \frac{1}{1 + [D_0 / D(u, v)]^{2n}}$$

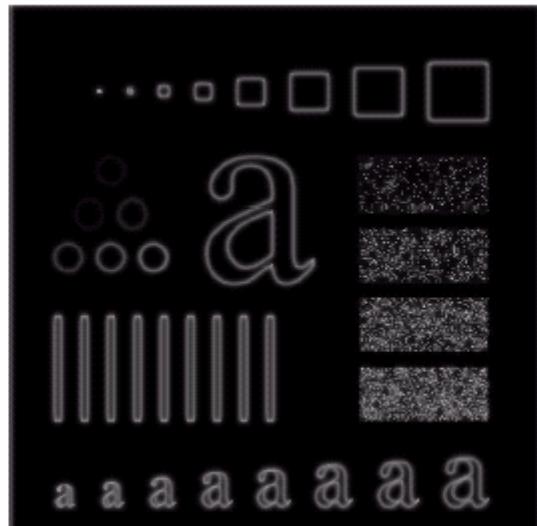
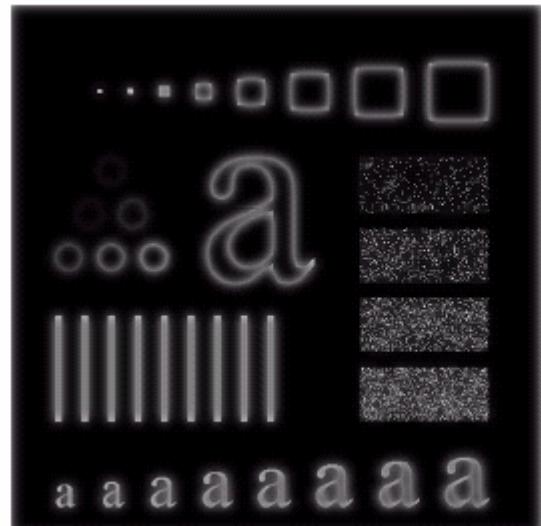


均为二阶，从左至右D<sub>0</sub>分别等于15, 30和80

# 高斯高通滤波器



$$H(u, v) = 1 - e^{-D^2(u, v)/2D_0^2}$$



从左至右 $D_0$ 分别等于15, 30和80



## 2、空间域卷积法

- 分析表明微分可以突出高频成分，从而使图象轮廓清晰。因此在空间域可用微分法锐化图象。
- 图象处理中常用的微分方法是“梯度法”，即用梯度作为锐化图象的度量。对于数字图象，则用差分来表示梯度。
- 实际应用中，为了获得各向同性的度量，用梯度的模代替梯度。



# 掩模

- 掩模是在空间域内处理图像象素值的矩阵。反映了处理后图像的象素值与原图像相同位置及其周围的象素值之间的关系。
- 掩模又被称为模板、滤波器、核或窗口。

-1	-2	-1
0	0	0
1	2	1



# 掩模

□ 典型的差分运算，可表示为：

$$G[f(x, y)] \approx |f(x, y) - f(x-1, y)| + |f(x, y) - f(x, y-1)|$$

0	0
-1	1

0	-1
0	1

直接差分算子



# 梯度算子

- Roberts算子
- Sobel算子
- Prewitt算子
- Kirsch算子

空间频率的滤波  
也可以在空间域  
来实现

-1	0
0	1

0	-1
1	0

-1	-2	-1
0	0	0
1	2	1

-1	0	1
-2	0	2
-1	0	1

-1	-1	-1
0	0	0
1	1	1

-1	0	1
-1	0	1
-1	0	1

5	5	5
-3	0	-3
-3	-3	-3

-3	5	5
-3	0	5
-3	-3	-3

# 梯度算子

原图



$|G_x|$



$|G_y|$



$|G_x|$

+

$|G_y|$



# 拉普拉斯算子

一个二元图像函数 $f(x, y)$ 的拉普拉斯变换定义为

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

几种常用的拉普拉斯算子：

0	1	0
1	-4	1
0	1	0

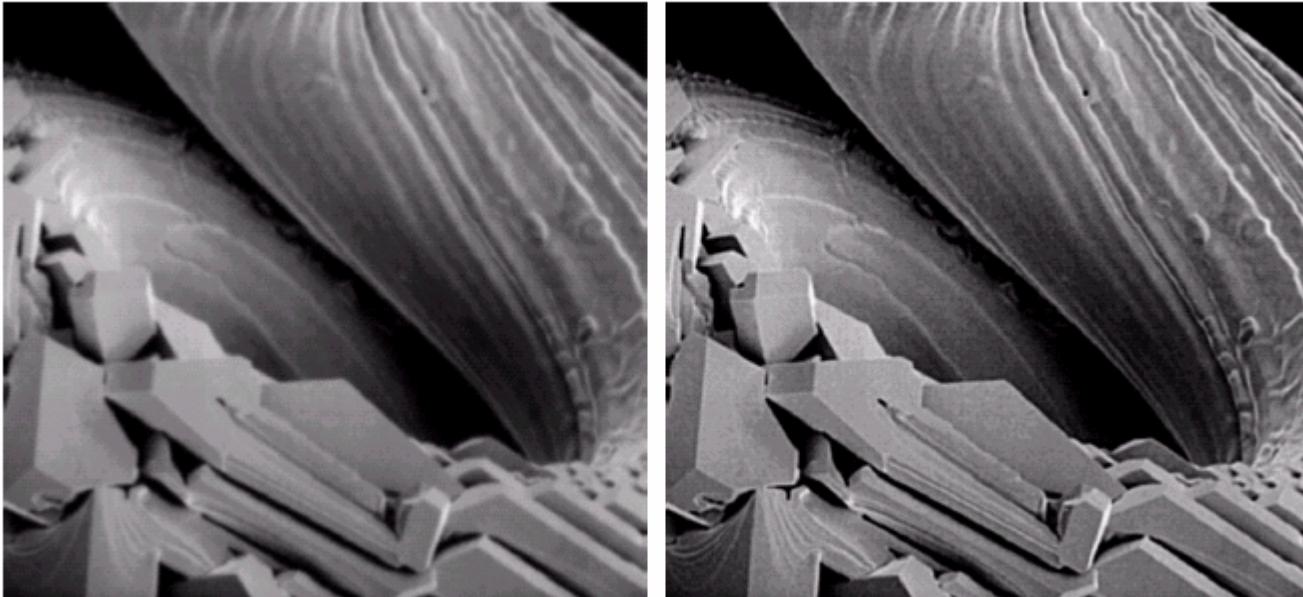
1	1	1
1	-8	1
1	1	1

0	-1	0
-1	4	-1
0	-1	0

-1	-1	-1
-1	8	-1
-1	-1	-1

# 使用拉普拉斯算子进行图形增强

$$g(x, y) = \begin{cases} f(x, y) - \nabla^2 f(x, y) & \text{如果拉普拉斯掩模中心系数为负} \\ f(x, y) + \nabla^2 f(x, y) & \text{如果拉普拉斯掩模中心系数为正} \end{cases}$$





## 2.5.3 平滑

### □ 空间频率域低通滤波法

- 巴特沃斯(Butterworth)低通滤波器
- 指数形低通滤波器
- 梯形低通滤波器

### □ 空间域法

- (1)局部平均法
- (2)中值滤波
- (3)多帧平均



# 1、空间频率域低通滤波法

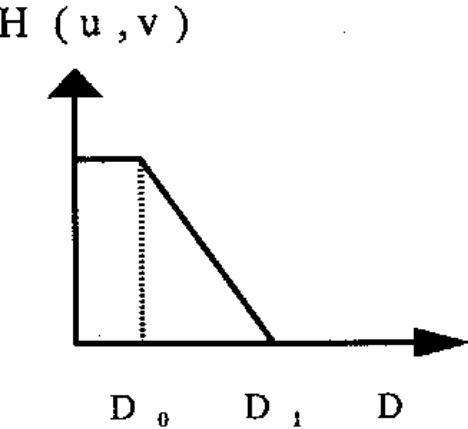
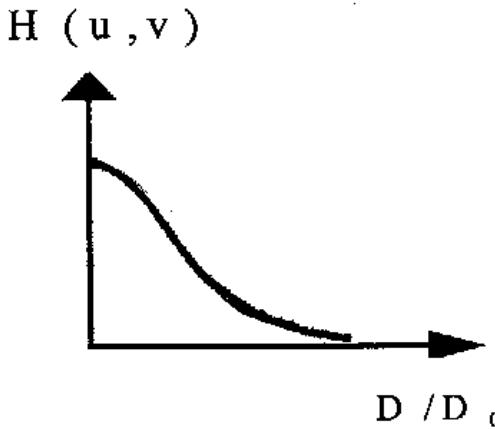
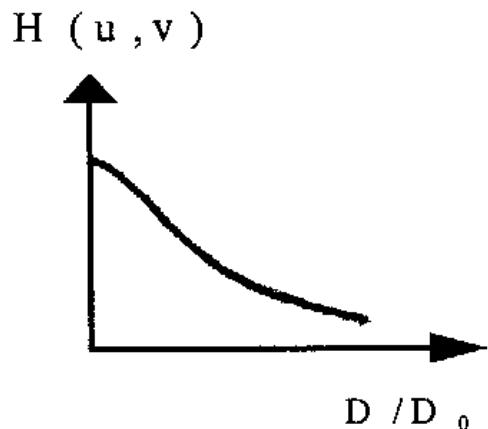
- 平滑的频率域方法是低通滤波法，这是一维低通滤波概念在二维的推广。
- 图象经过二维傅立叶变换后，噪声通常含在空间高频分量中，因此，对高频成份加以衰减可在频率域中实现平滑处理。

# 1、空间频率域低通滤波法

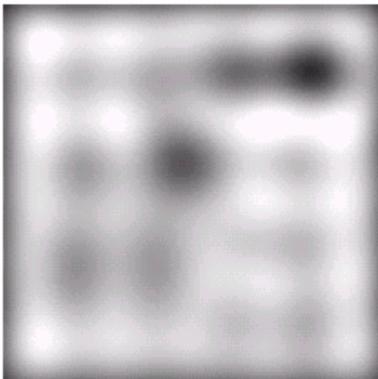
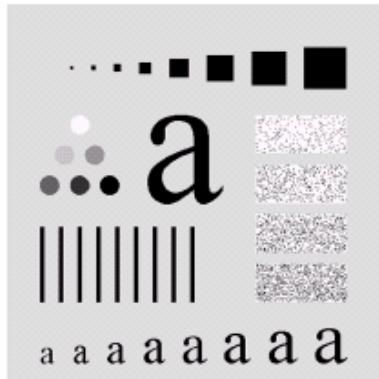
## □ 常用的低通滤波器

- 巴特沃斯(Butterworth)低通滤波器
- 指数形低通滤波器
- 梯形低通滤波器

## □ 常用的低通滤波器一维图例：



# 理想低通滤波器

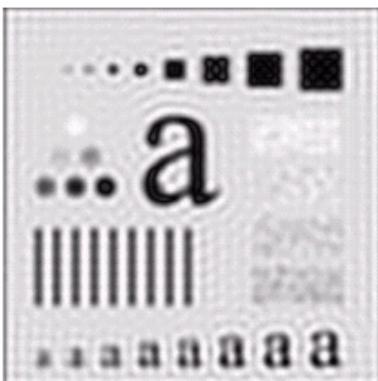
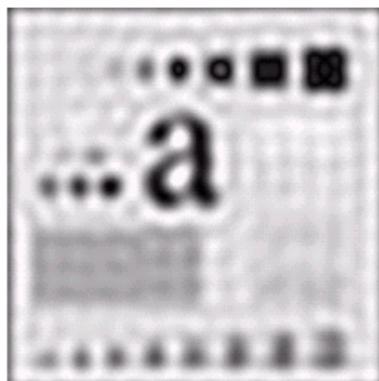


$$H(u, v) = \begin{cases} 1 & D(u, v) \leq D_0 \\ 0 & D(u, v) > D_0 \end{cases}$$

原图

滤除功率8%

$$D_0=5$$

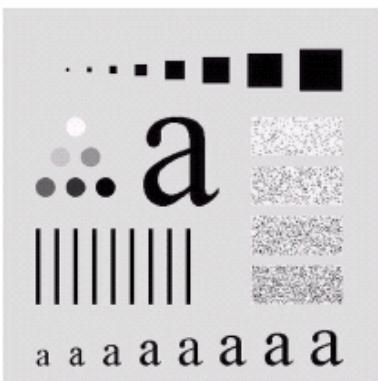
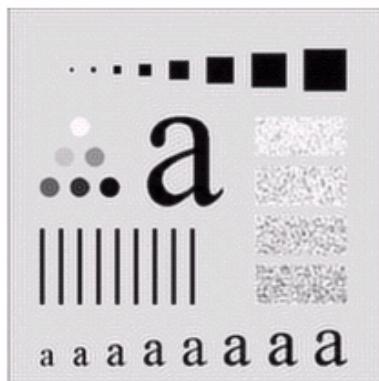


滤除功率5.4%

$$D_0=15$$

滤除功率3.6%

$$D_0=30$$



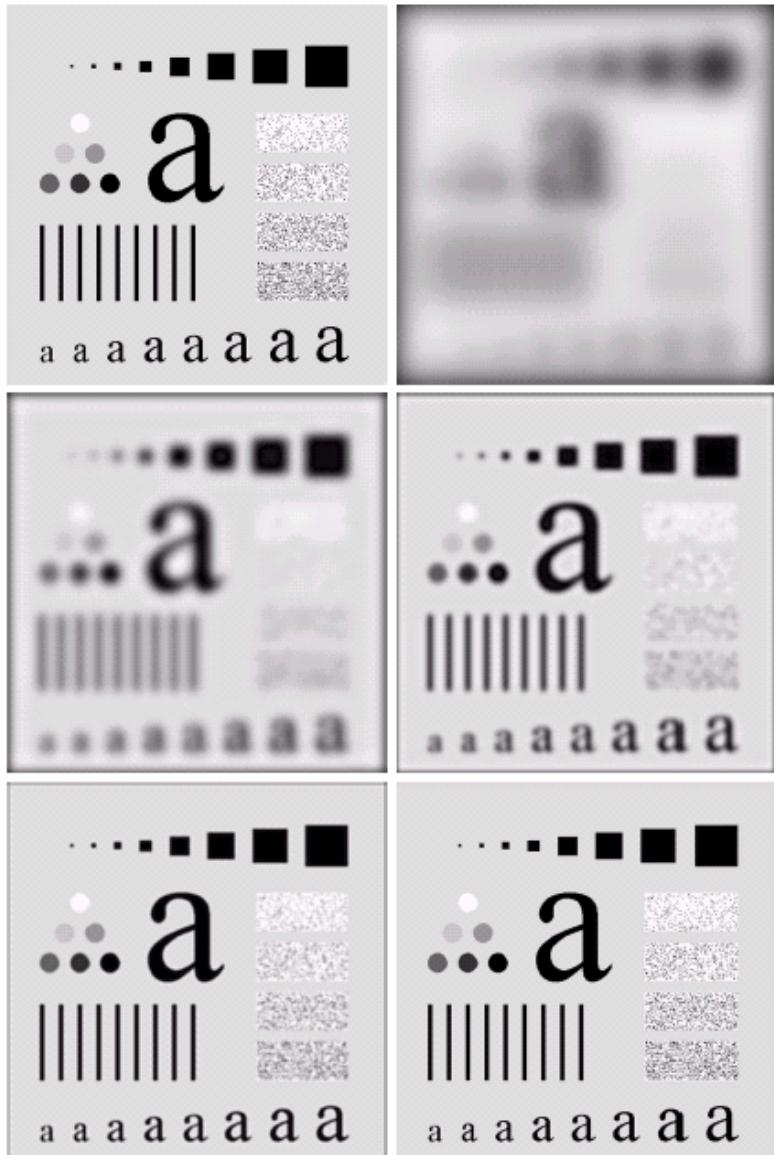
滤除功率2%

$$D_0=80$$

滤除功率0.5%

$$D_0=230$$

# 巴特沃思低通滤波器



$$H(u, v) = \frac{1}{1 + [D(u, v)/D_0]^{2n}}$$

原图

二阶,  $D_0=5$

二阶,  
 $D_0=15$

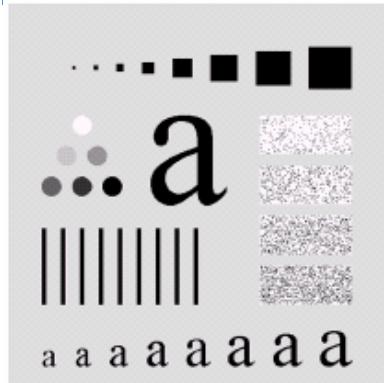
二阶,  
 $D_0=30$

二阶,  
 $D_0=80$

二阶,  
 $D_0=230$



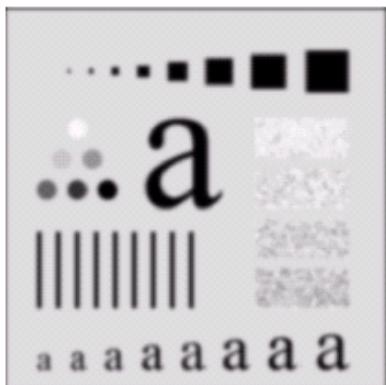
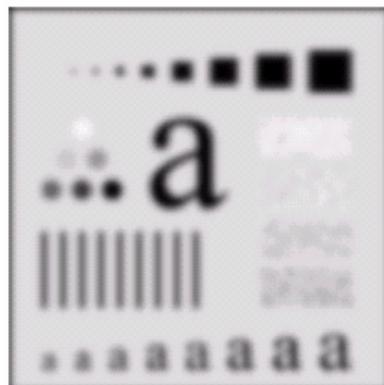
# 高斯低通濾波器



$$H(u,v) = e^{-D^2(u,v)/2D_0^2}$$

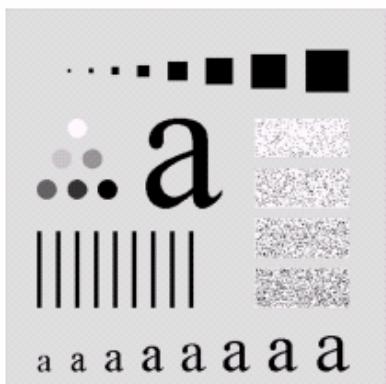
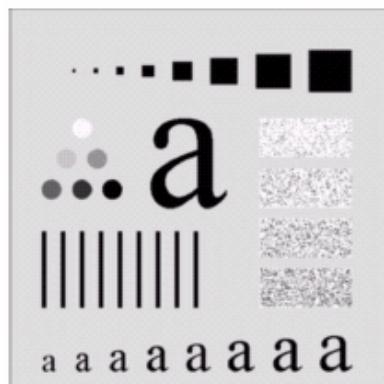
原图

$D_0=5$



$D_0=15$

$D_0=30$



$D_0=80$

$D_0=230$



## 2、空间域法

### (1) 局部平均法

局部平均法是用某象素邻域内的各点灰度的平均值来代替该象素原来的灰度值。

1/9

1	1	1
1	1	1
1	1	1

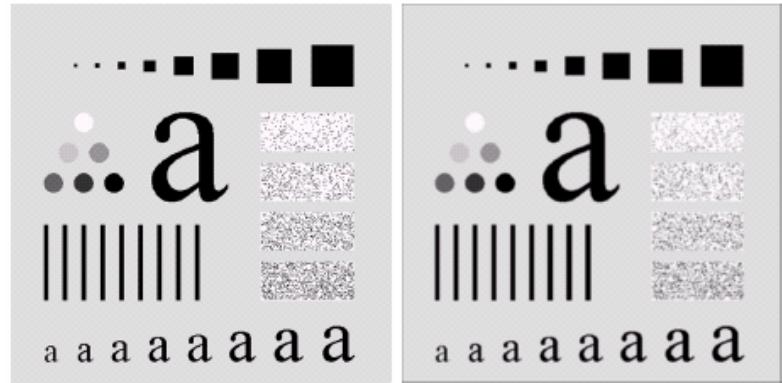
1/10

1	1	1
1	2	1
1	1	1

1/16

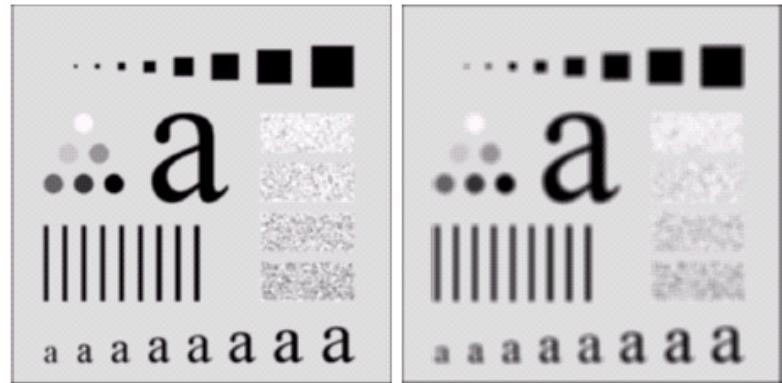
1	2	1
2	4	2
1	2	1

# 例



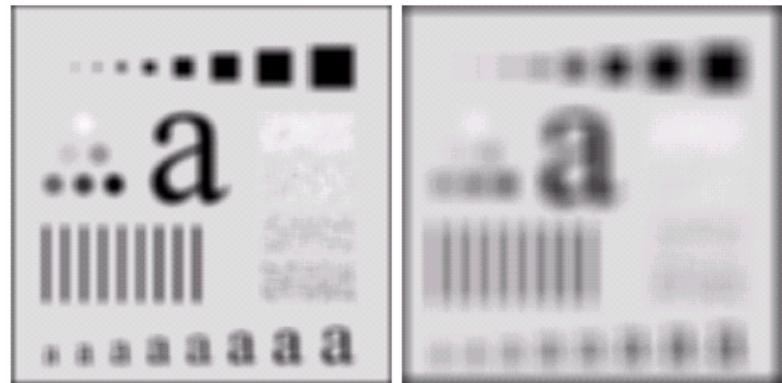
原图

$n=3$ 方形均  
值滤波



$n=5$ 方形均  
值滤波

$n=9$ 方形均  
值滤波



$n=15$ 方形均  
值滤波

$n=35$ 方形均  
值滤波



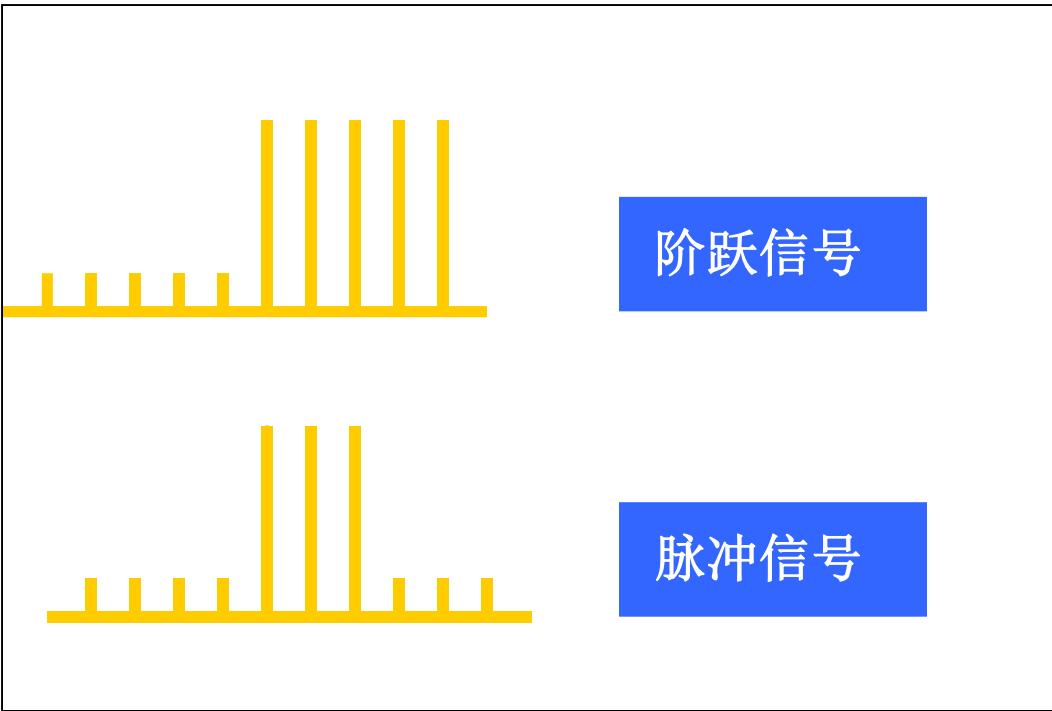
## 2、空间域法

### (2) 中值滤波

- 中值滤波法用局部邻域（窗口）里的中值来代替上述局部平均法中的局部平均值。也就是将以该点为中心的某个窗口框住的各象素的中间值作为处理后图象中该点象素的值。
- 二维中值滤波的概念是一维的推广。
- 以一维为例：

奇数窗口中五点的值为88、95、230、110、120，按大小重排这五个值得88、95、110、120、230，其位于中间位置上的“中值”是110。

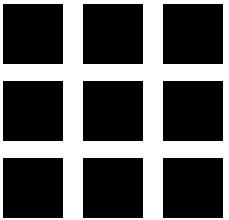
# 中值滤波保边缘的性能



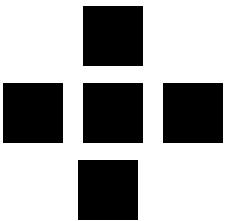
# 二维中值滤波窗口

大小、形状  
均可调节

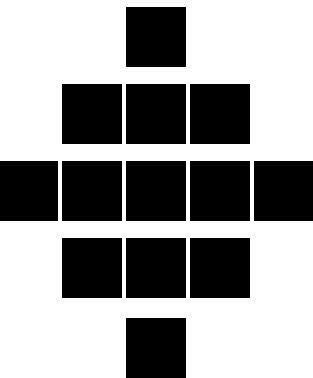
方形



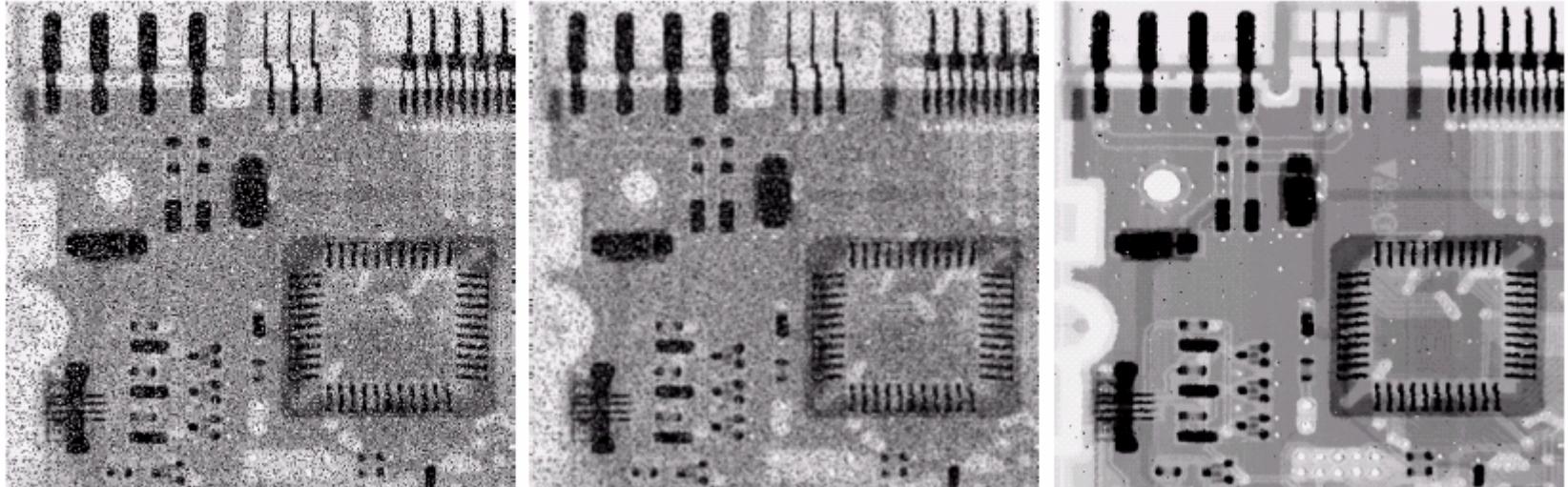
十字形



菱形



# 例



从左到右依次分别为  
椒盐噪声污染的电路板X光图像  
用 $3 \times 3$ 均值掩模去除噪声  
用 $3 \times 3$ 中值滤波器去除噪声

去除椒盐噪声使用自适  
应中值滤波器的效果会  
更好一些，详情参见  
《数字图像处理》（冈  
萨雷斯）



## 2、空间域法

### (3)多帧平均

- 多帧平均是较好的方法。
  - ✓ 可以提高信噪比
  - ✓ 基本不引起模糊
- 多帧平均要求能多次得到同一图象。

# 例



星团  
图像

加性高斯噪  
声污染



K=8带噪图  
像取平均

K=16带噪  
图像取平均



K=24带噪  
图像取平均

K=128带噪  
图像取平均



## 2.5.4 伪彩色 (Pseudo Color)

- 在图象的分析中，彩色是一种能帮助人们简化目标提取和分类的重要参量。
- 人眼只能区分几十种不同等级的灰度，但却可以分辨几千种不同颜色与不同亮度的彩色。
- 伪彩色处理就是把黑白图象的各个灰度按照一种线性或非线性函数关系映射成相应的彩色。这种映射是输入与输出象素间一对一的运算，即不涉及图象空间位置的变动。

# 彩色空间

## □ 彩色规范

### ■ RGB格式

### ■ HSI格式

✓ Intensity

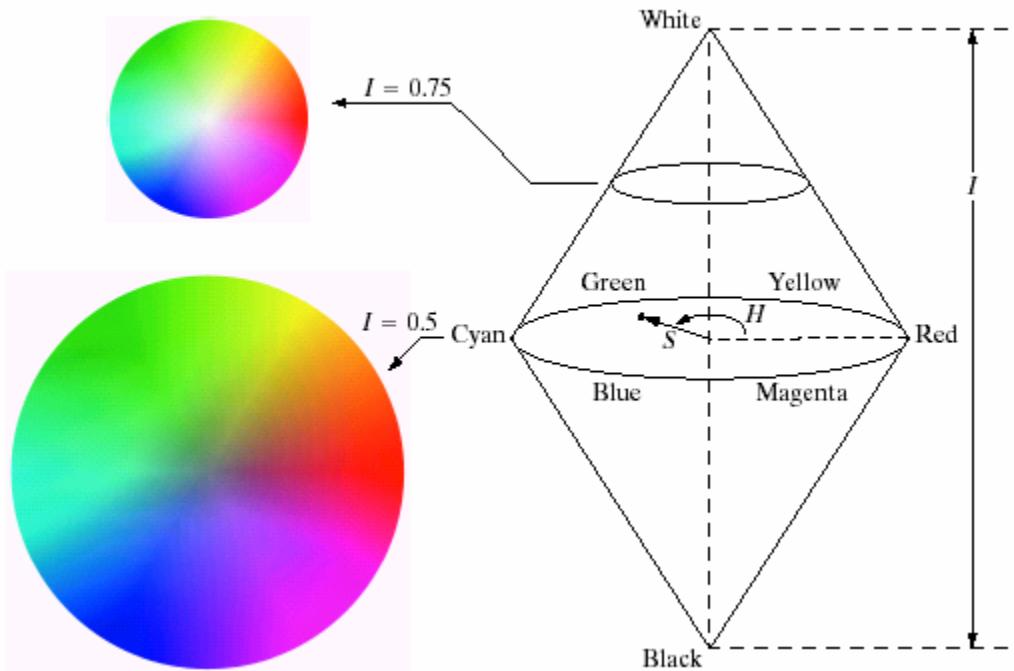
强度, 亮度

✓ Hue

色度

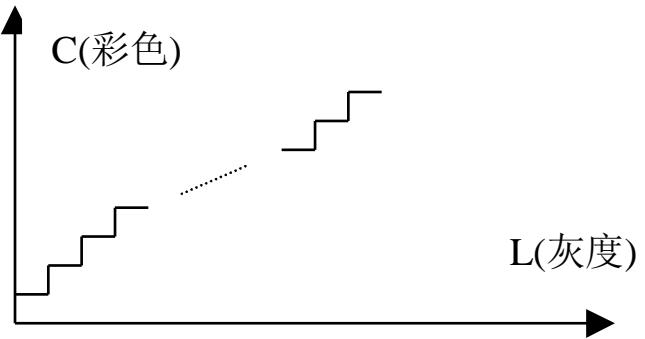
✓ Saturation

饱和度



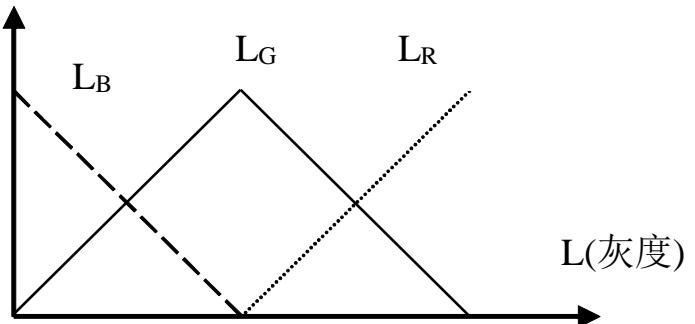
# 伪彩色方法

## □ 灰度分层法



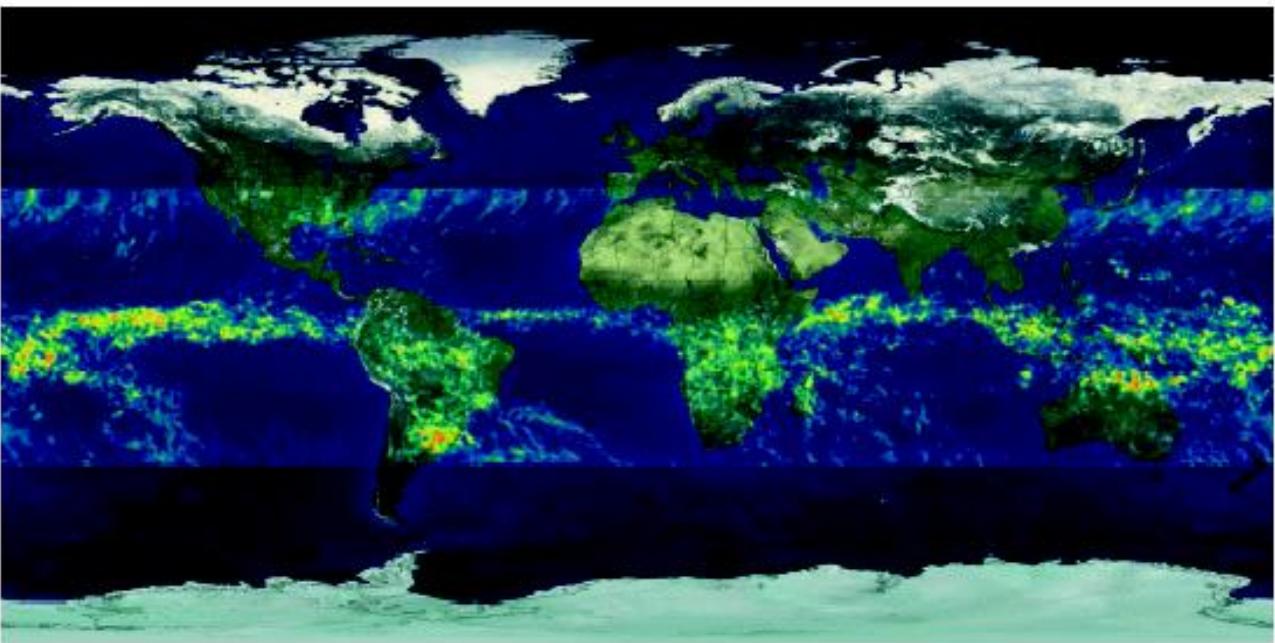
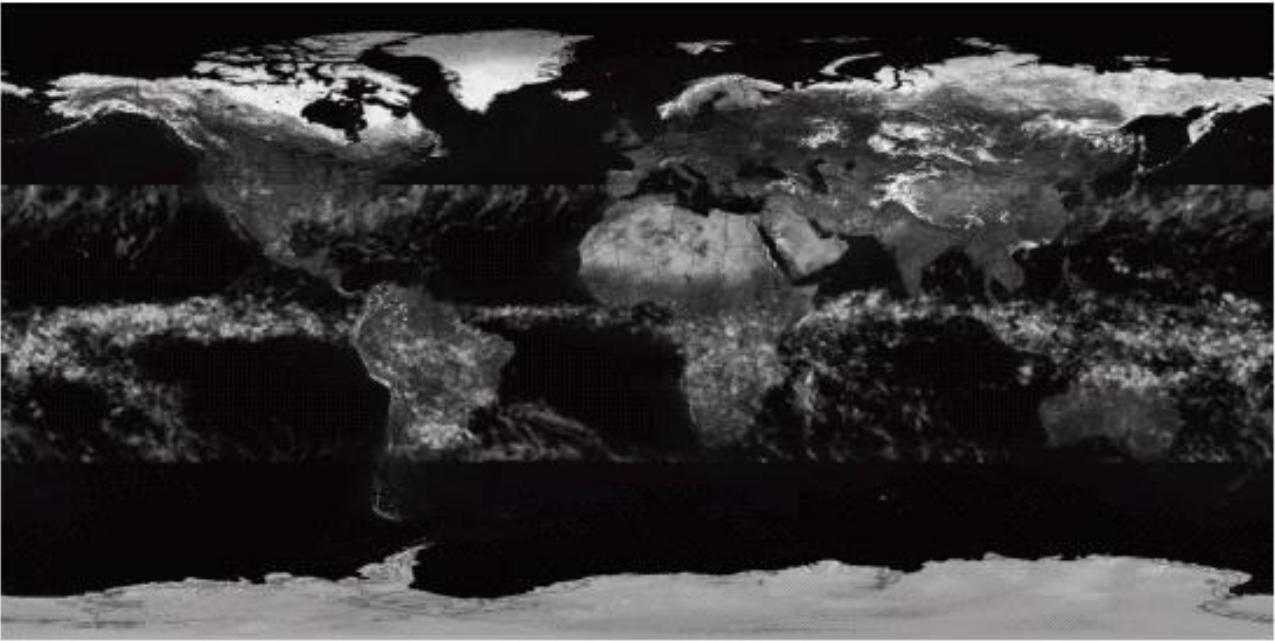
(a) 灰度分层

## □ 变换法

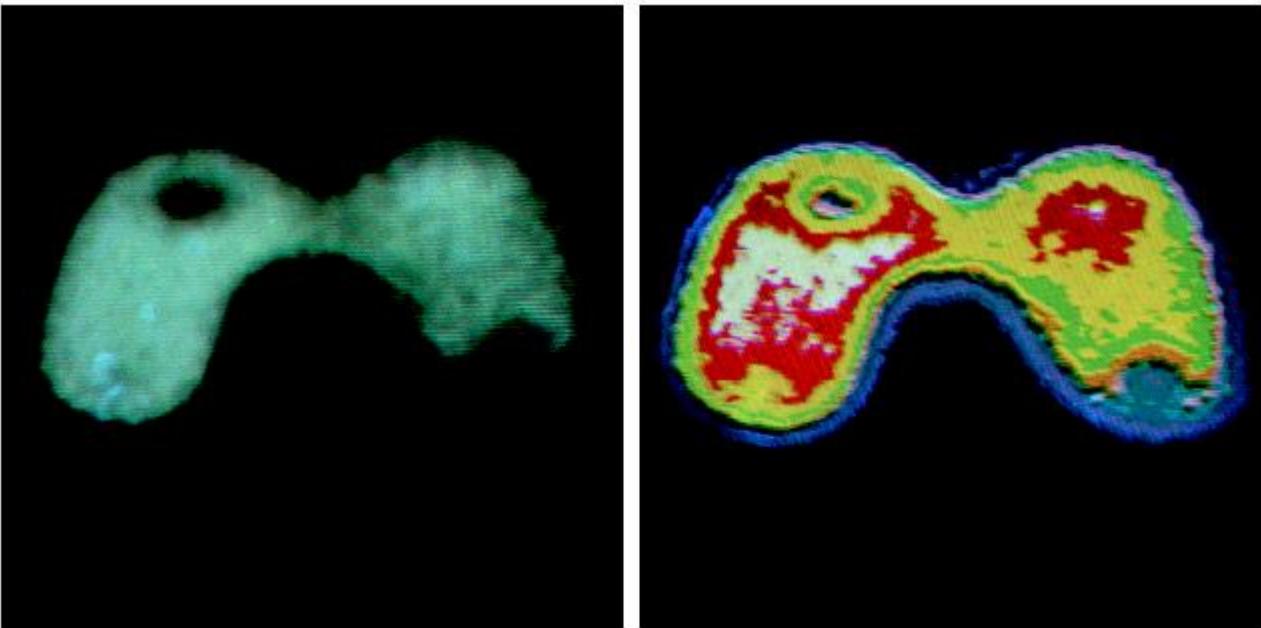


(b) 灰度映射函数

# 伪彩色例



# 伪彩色例





## 2.5.5 同态滤波

- 一幅图像 $f(x,y)$ 可以表示为照射分量和反射分量的乘积。

$$f(x,y) = i(x,y)r(x,y)$$

然而上式不能用来直接对两部分分量分别进行操作，因为两个函数乘积的傅立叶变换是不可分的。

$$\mathcal{F}\{f(x,y)\} \neq \mathcal{F}\{i(x,y)\}\mathcal{F}\{r(x,y)\}$$

我们对图像函数两边取对数，则可以将两个分量分开。

$$z(x,y) = \ln f(x,y) = \ln i(x,y) + \ln r(x,y)$$

$$\mathcal{F}\{\ln f(x,y)\} = \mathcal{F}\{\ln i(x,y)\} + \mathcal{F}\{\ln r(x,y)\}$$



# 同态滤波的过程如下：

1. 两边取对数： $f(x,y) = \ln i(x,y) + \ln r(x,y)$
2. 两边取付氏变换：

$$F(u,v) = I(u,v) + R(u,v)$$

3. 用一频域函数  $H(u,v)$  处理  $F(u,v)$ ：

$$H(u,v)F(u,v) = H(u,v)I(u,v) + H(u,v)R(u,v)$$

4. 反变换到空域：

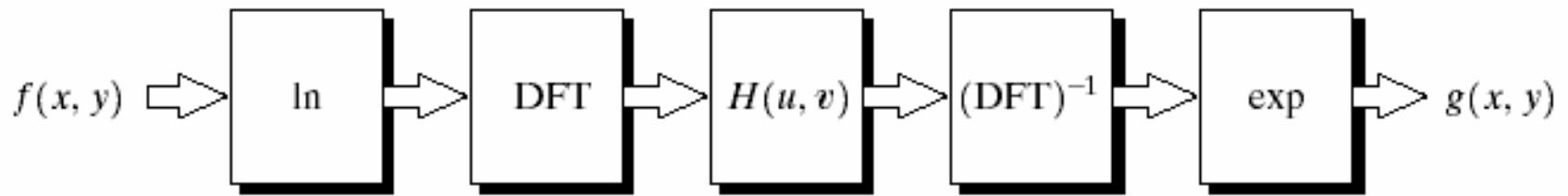
$$s(x,y) = i'(x,y) + r'(x,y)$$

5. 两边取指数：

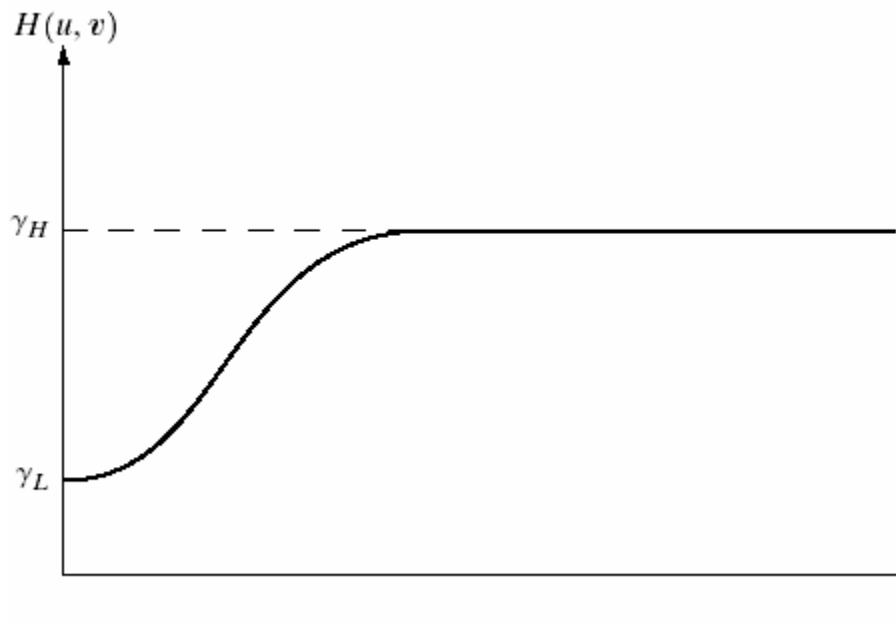
$$g(x,y) = \exp\{i'(x,y)\} + \exp\{r'(x,y)\}$$

# 同态滤波

## □ 流程图



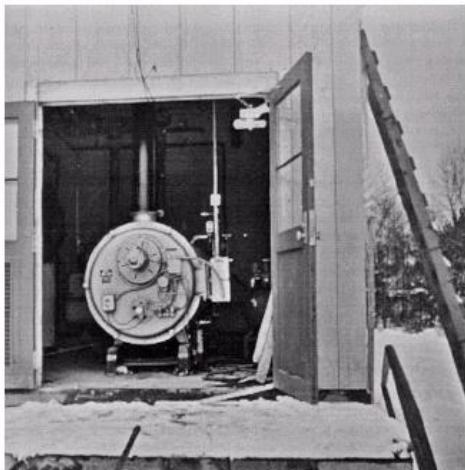
## □ 典型曲线



# 同态滤波

- **特点：**能消除乘性噪声，能同时压缩图象的整体动态范围和增加图象中相邻区域间的对比度

- **示例：**

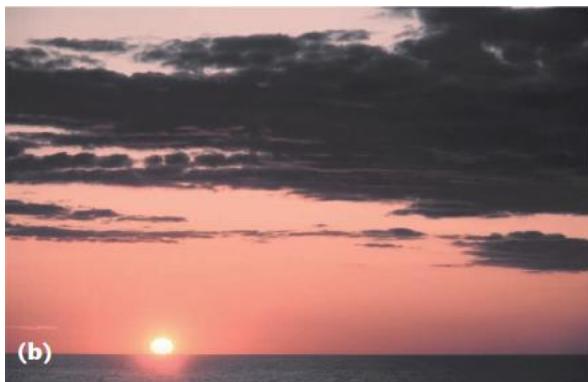


## 2.5.5 颜色迁移 (Color Transfer)

- 给定一张参考图像(b)，根据它的颜色对目标图像(a)进行颜色变换，使得变换后的结果(c)有和参考图像相似的颜色分布。



(a) 目标图像



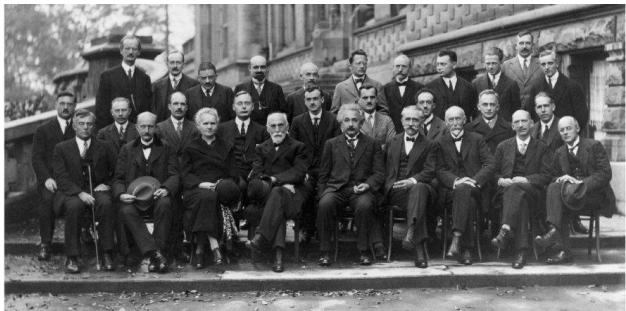
(b) 参考图像



(c) 迁移结果

- 灰度图像 → 彩色图像

1927年第五次  
索尔维会议物  
理学家合影





# 颜色迁移的应用

- 对彩色图像进行光照、色彩的变换和加工，获得不同的艺术效果
- 通过一张场景图像，获得同一场景在不同的时间、光照条件下的效果
- 为灰度图像添加颜色，提高图像视觉效果
- 对褪色的平面文物（建筑彩绘、壁画等）进行修复



# 经典算法：Reinhard算法[1]

- 将目标图像和参考图像变换到 $l\alpha\beta$ 颜色空间
- 分别计算图像在 $l, \alpha, \beta$ 通道的均值和方差
- 将目标图像各像素点在各通道的取值减去均值，弱化其整体的彩色信息，将剩余部分按照标准差比值缩放
- 加入参考图像的均值得到合成图像

$$I_k = \frac{\delta_t^k}{\delta_s^k} \cdot (S^k - \text{mean}(S^k)) + \text{mean}(T^k), \quad k = (l, a, b)$$

- 转换回RGB空间

[1] Reinhard E, Ashikhmin M, Gooch B, et al. Color transfer between images[J]. *IEEE Computer graphics and applications*, 2001, 21(5): 34-41.

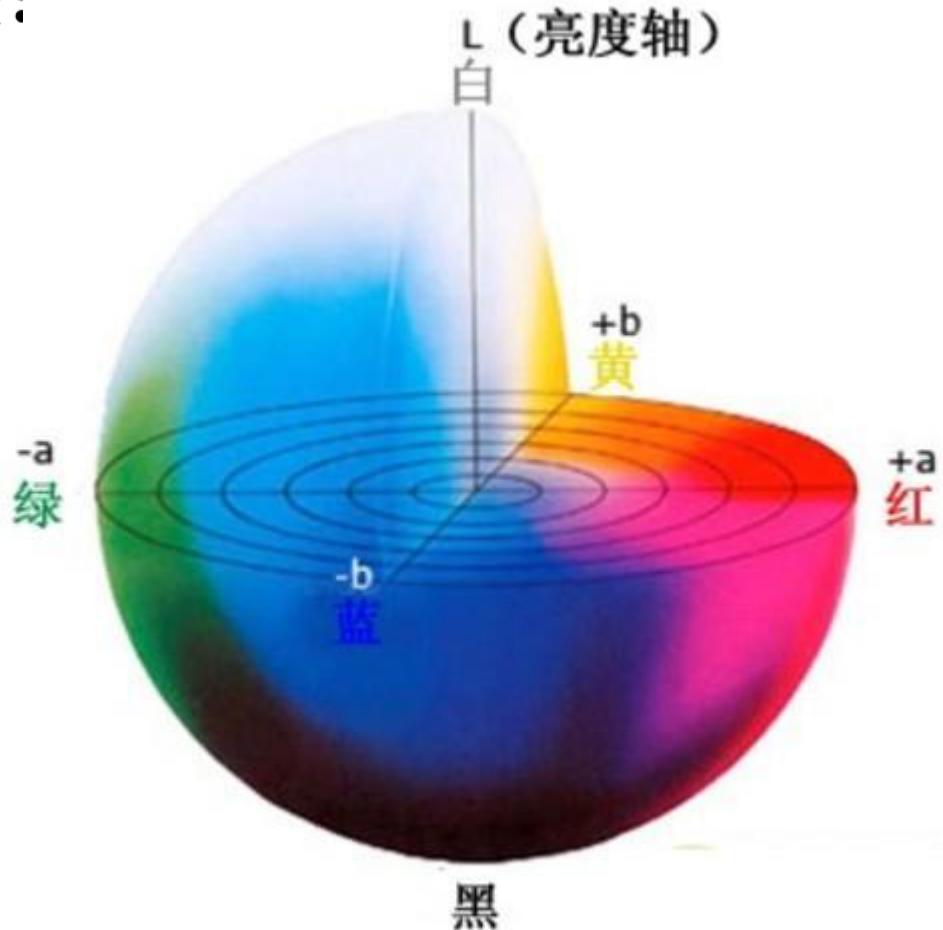


# 为什么选择 $l\alpha\beta$ 颜色空间？

- $l$ : 亮度分量
- $\alpha$ : 黄-蓝色度分量
- $\beta$ : 红-绿色度分量
  
- 颜色迁移是改变图像颜色基调的过程，通常希望在改变图像的一个颜色属性时，不改变其它属性
- RGB空间三通道有很强的相关性，而 $l\alpha\beta$ 空间各通道有最小的相关性

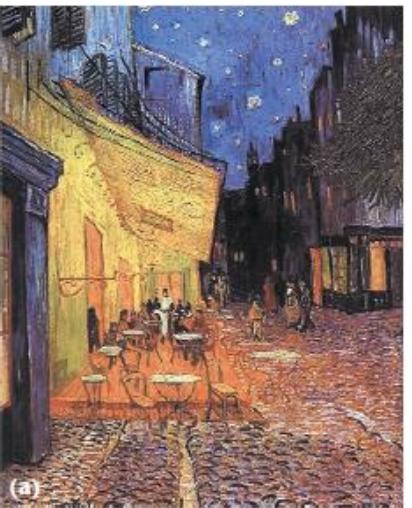
# $l\alpha\beta$ 颜色空间

- RGB- $l\alpha\beta$  空间转换:  
RGB-XYZ-LMS-  $l\alpha\beta$
- 其中LMS-  $l\alpha\beta$ 为  
非线性变换
- 变换方法参见[1]



[1] Reinhard E, Ashikhmin M, Gooch B, et al. Color transfer between images[J]. *IEEE Computer graphics and applications*, 2001, 21(5): 34-41.

# 迁移效果



左：  
一天中的不同时刻

上：  
使用艺术图象做参考

右下：  
Gamma校正



# 经典算法：Welsh算法[2]

- 应用于灰度图像颜色迁移
  - 将参考图像和目标灰度图像转换到 $l\alpha\beta$ 颜色空间
  - 根据灰度图像的亮度和标准差，对参考图像进行亮度重映射

$$l_{ref}^* = \frac{\sigma_{l_{dst}}}{\sigma_{l_{ref}}} (l_{ref} - \bar{l}_{ref}) + \bar{l}_{dst}$$

- 从参考图像随机选取一批样本点，将像素点的亮度和邻域内亮度的线性组合值作为权值  $w = \frac{l}{2} + \frac{\sigma}{2}$
- 对灰度图像每个像素计算权值，在参考图像中寻找最接近的样本点，获得样本点的 $\alpha, \beta$ 分量
- 变换回RGB空间

[2] Welsh, Tomihisa, Michael Ashikhmin, and Klaus Mueller. "Transferring color to greyscale images." ACM Transactions on Graphics (TOG). Vol. 21. No. 3. ACM, 2002.

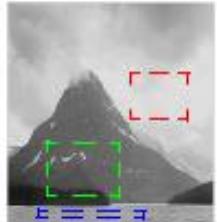
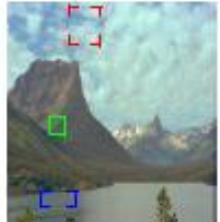
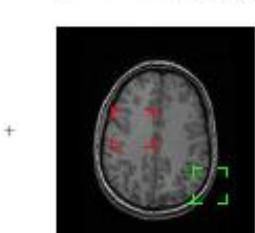
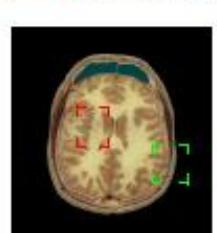
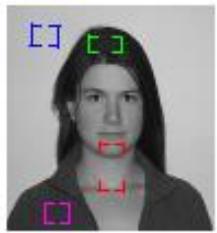
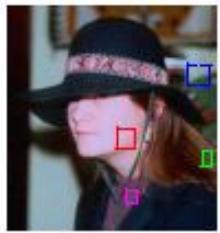
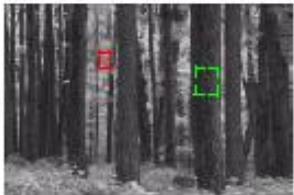
# 迁移效果



+

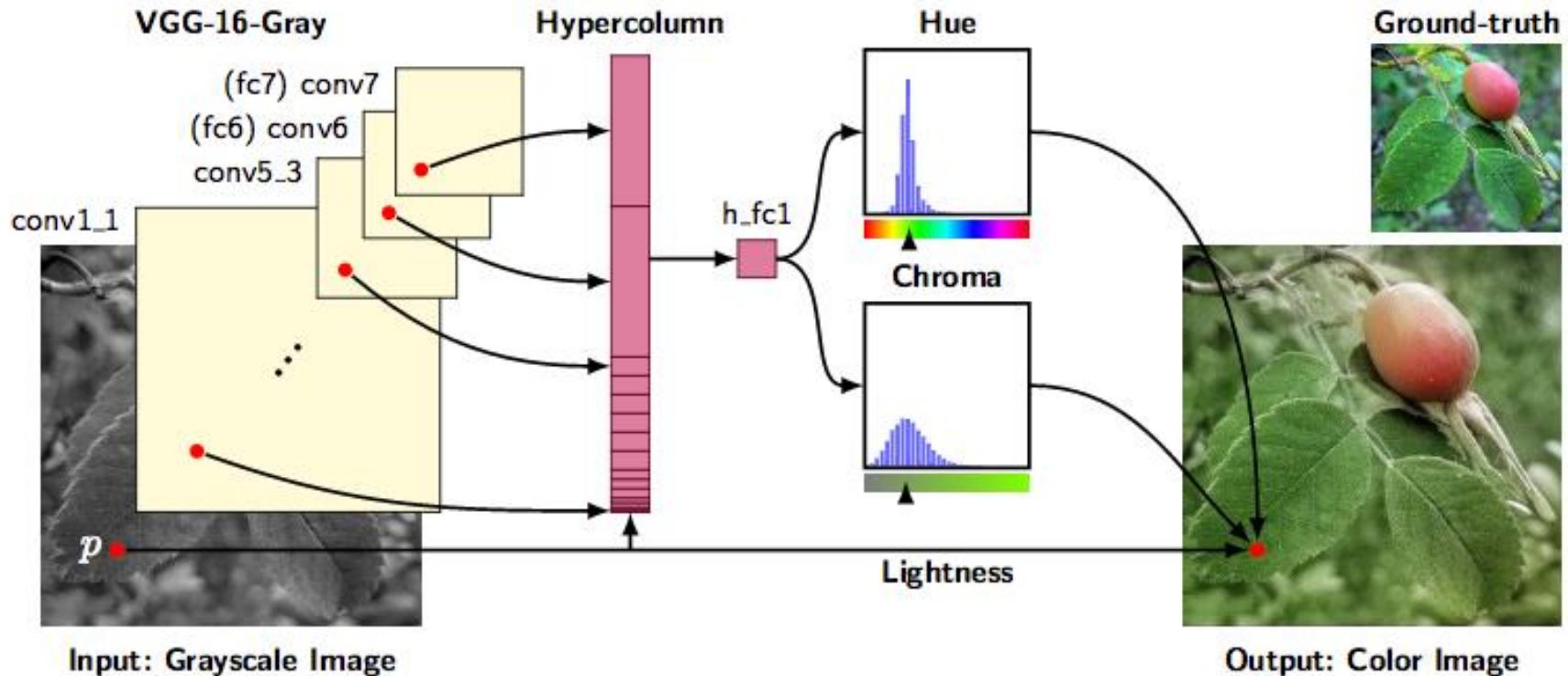


=

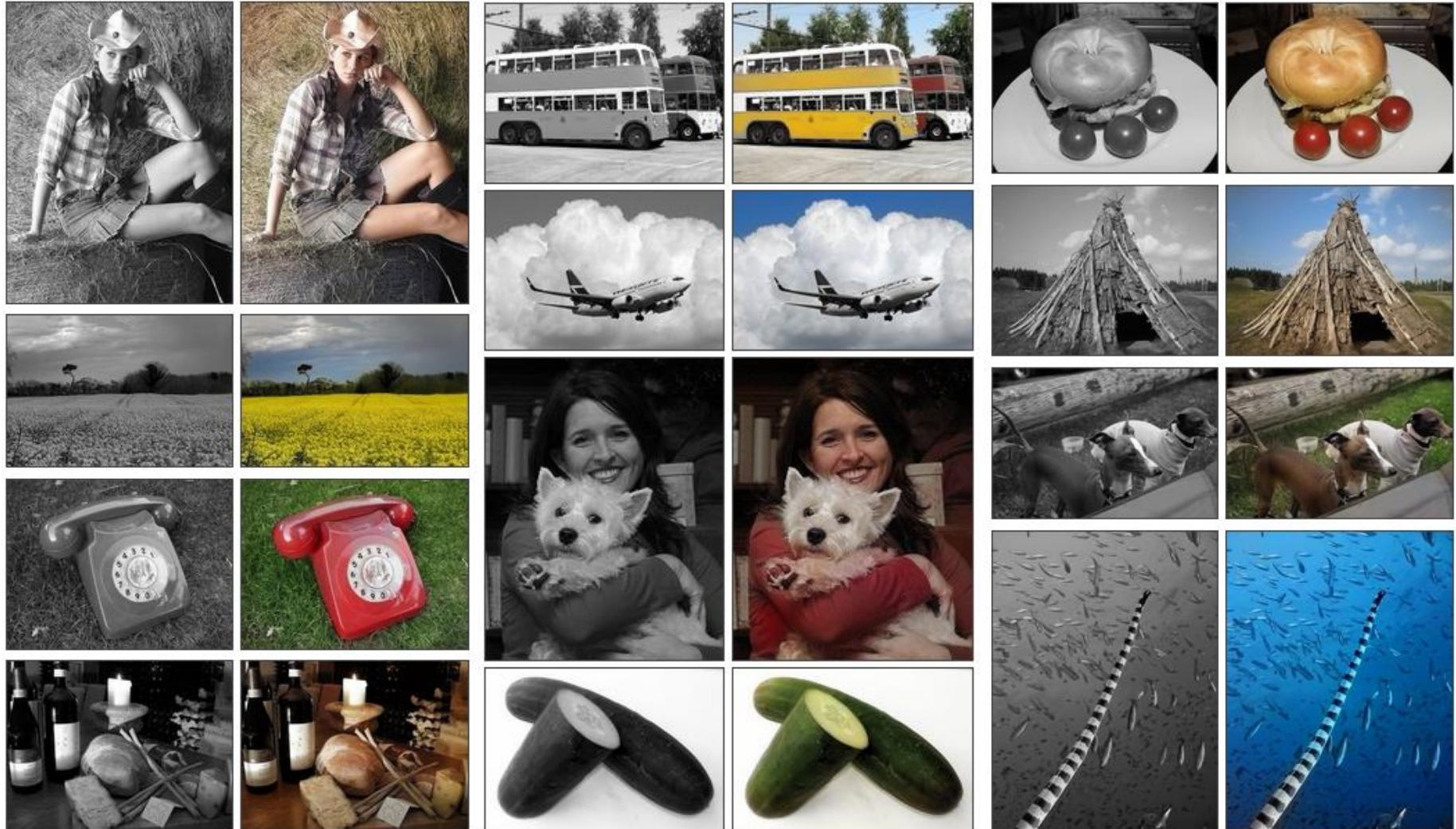


# 基于深度学习的自动图像配色

## □ Automatic Colorization



# 实验结果





# 经典算法：Hertzmann算法

- 图像类比算法
  - 已知源图像对  $A, A'$ , 和一张未处理的目标图像  $B$ , 按照以下规则合成处理后的目标图像  $B'$ :

$$A: A' :: B: B'$$

即学习源图像对  $A, A'$  之间的关系，并将其作用于  $B$ , 使得视觉效果上,  $B$  和  $B'$  之间的关系接近于  $A$  和  $A'$  之间的关系

效果：将图像风格、特征传递到新图像中

[4] Hertzmann, Aaron, et al. "Image analogies." Proceedings of the 28th annual conference on Computer graphics and interactive techniques. ACM, 2001.

# 迁移效果



*A*



*A'*



*B*



*B'*



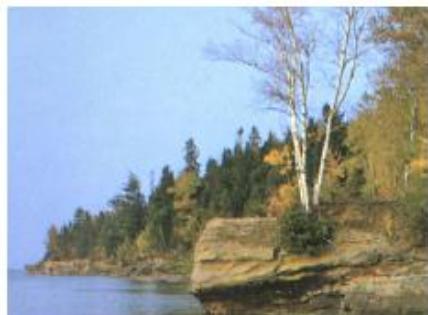
Unfiltered source (*A*)



Filtered source (*A'*)



Training pairs (*A, A'*)



Unfiltered target (*B*)



Filtered target (*B'*)



Unfiltered target (*B*)

Filtered target (*B'*)

上：  
油画效果

左：  
模糊效果

右：  
Super  
resolution

# 迁移效果

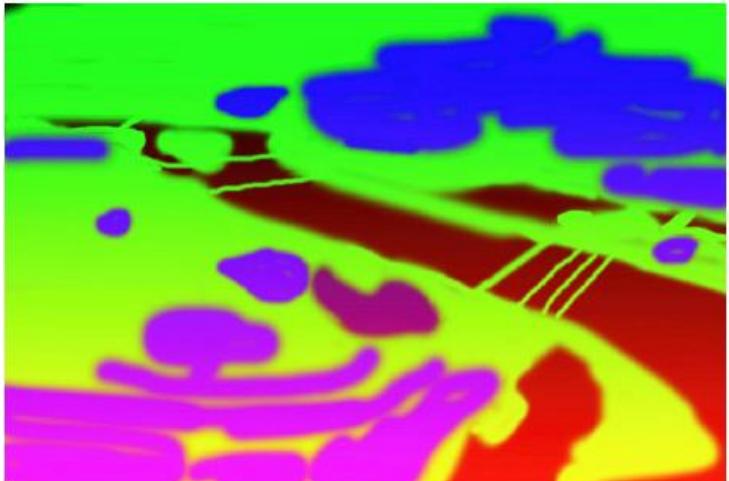


(a) 目标图像

(a) 梵高风格

(c) 水彩风格

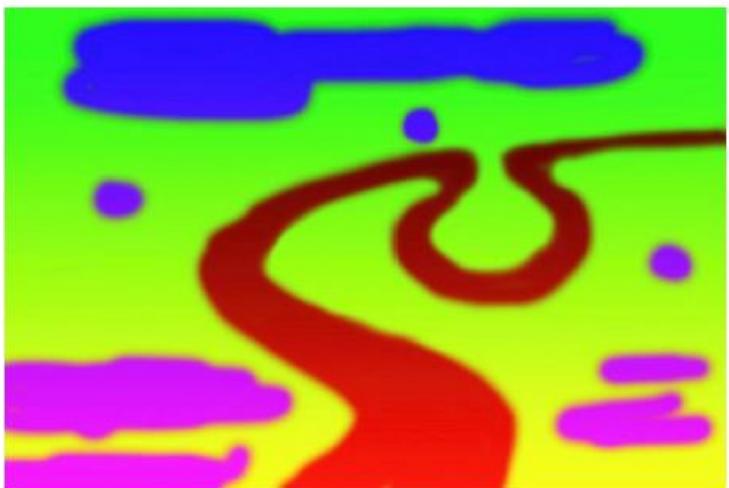
# 迁移效果（纹理拼贴）



Unfiltered source ( $A$ )



Filtered source ( $A'$ )



Unfiltered ( $B$ )



Filtered ( $B'$ )

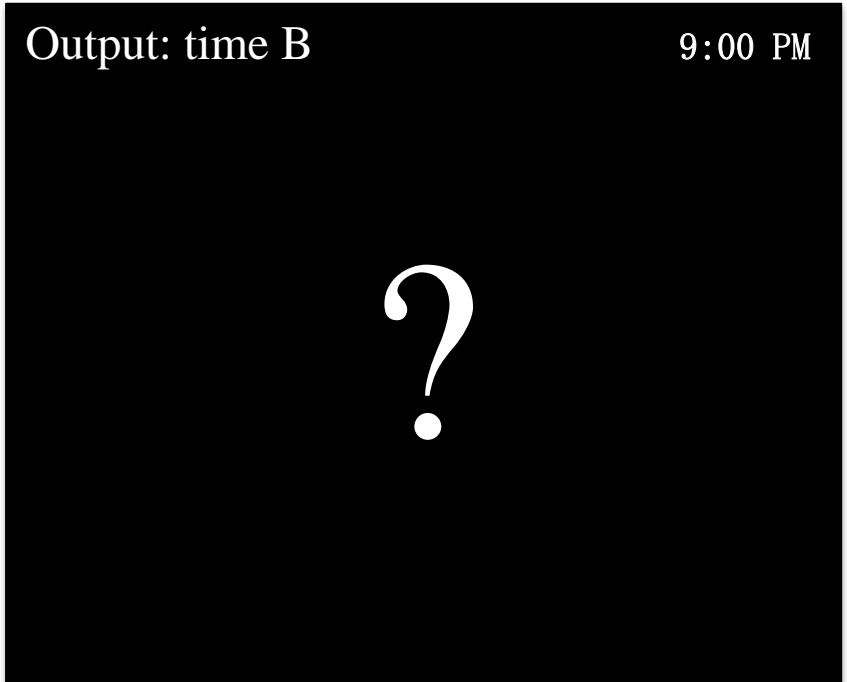
# 延迟摄影色彩迁移

- 通过给定的一张户外照片，预测 同一个地点和视角下 其他时间的场景表现

Input: time A



Output: time B



[5] Shih, Yichang, et al. "Data-driven hallucination of different times of day from a single outdoor photo." ACM Transactions on Graphics (TOG) 32.6 (2013)

# 延迟摄影色彩迁移

- 通过给定的一张户外照片，预测 同一个地点和视角下 其他时间的场景表现

Input: time A



Output: time B

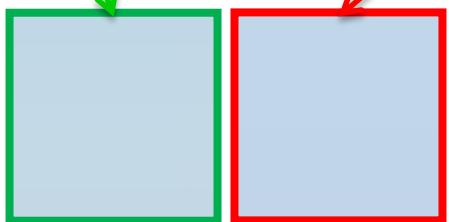


[5] Shih, Yichang, et al. "Data-driven hallucination of different times of day from a single outdoor photo." ACM Transactions on Graphics (TOG) 32.6 (2013): 200.

# 问题挑战性

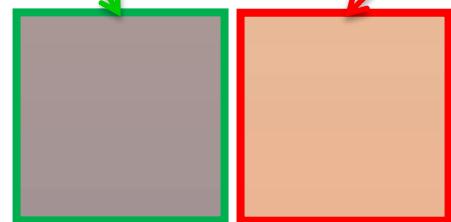
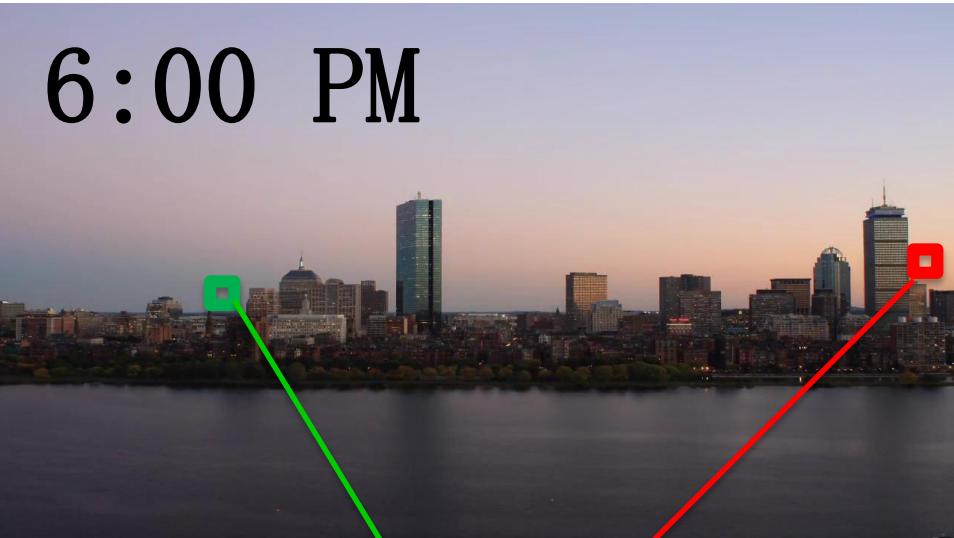
- 颜色变换和空间位置相关

4:00 PM



白天时颜色相近

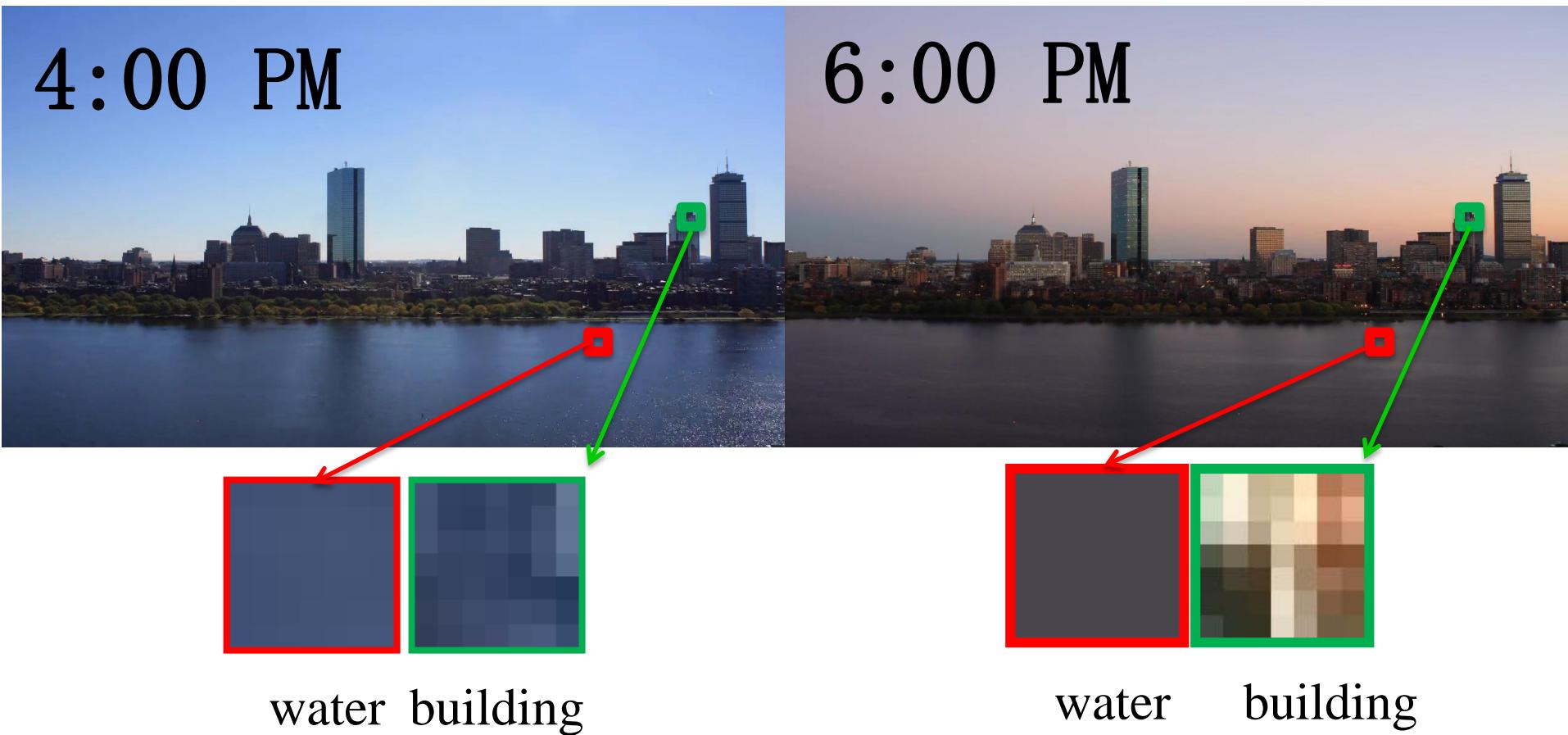
6:00 PM



但黄昏时变得不同

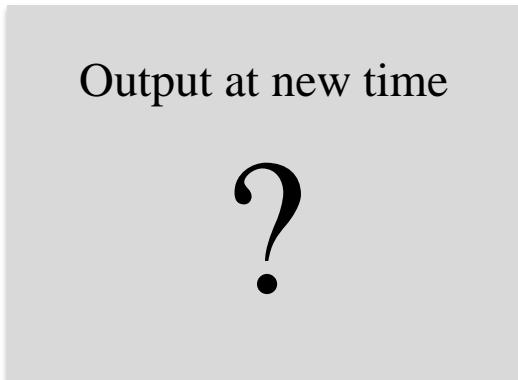
# 问题挑战性

- 颜色变换和物体相关
  - 水面和建筑的颜色变化不同



# 问题描述

- 输入：单张图像 + 目标时间
- 输出：目标时间下的同一场景



- 需求：自动处理，不需要用户交互



# 算法框架

Input



Target time: 9pm

# 算法框架

## 1. 从数据库中检索和输入匹配的延迟摄影视频



**Target time: 9pm**



# 算法框架

## 1. 从数据库中检索和输入匹配的延迟摄影视频



Target time: 9pm

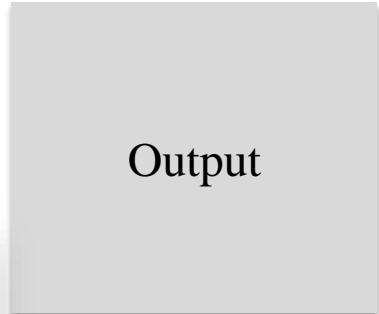
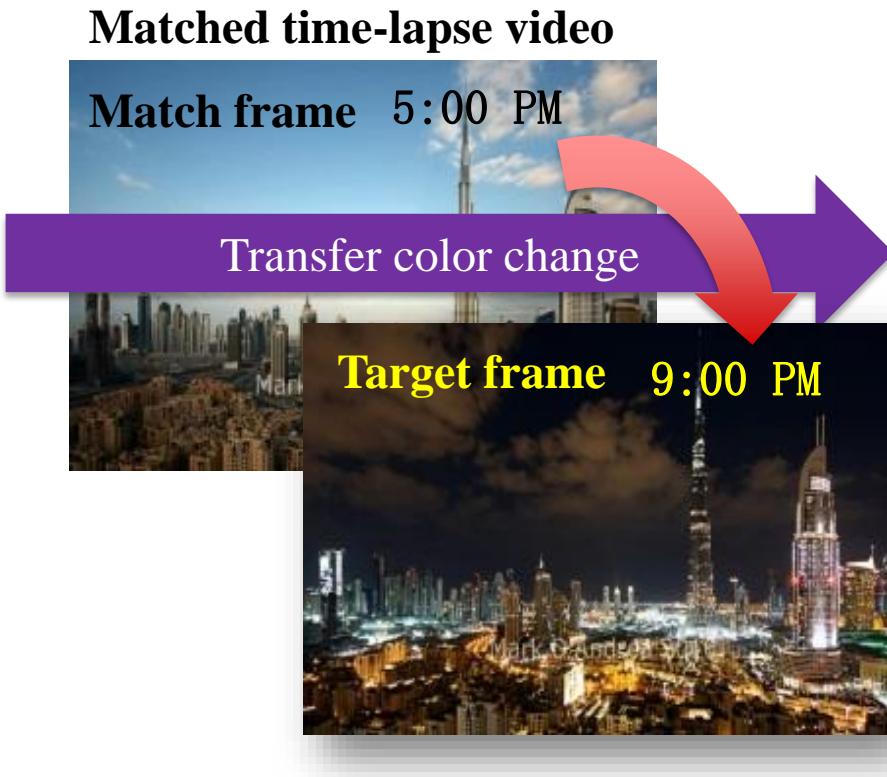


# 算法框架

1. 从数据库中检索和输入匹配的延迟摄影视频
2. 进行颜色迁移



Target time: 9pm

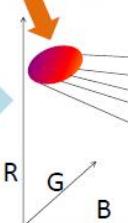
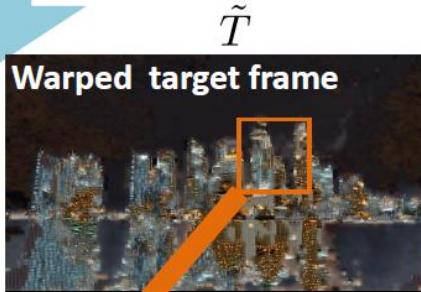
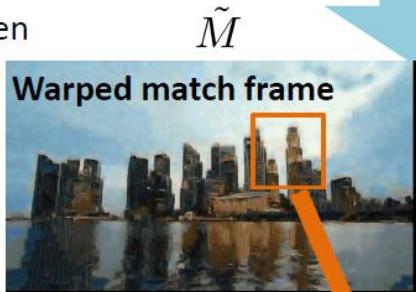


# 算法框架

(1) Retrieve from database. Time-lapse videos similar to input image (Sec 5.1)



(2) Compute a dense correspondence across the input image and the time-lapse , and then warp the time-lapse (Sec. 5.2)



Affine color mapping learned  
from the time-lapse

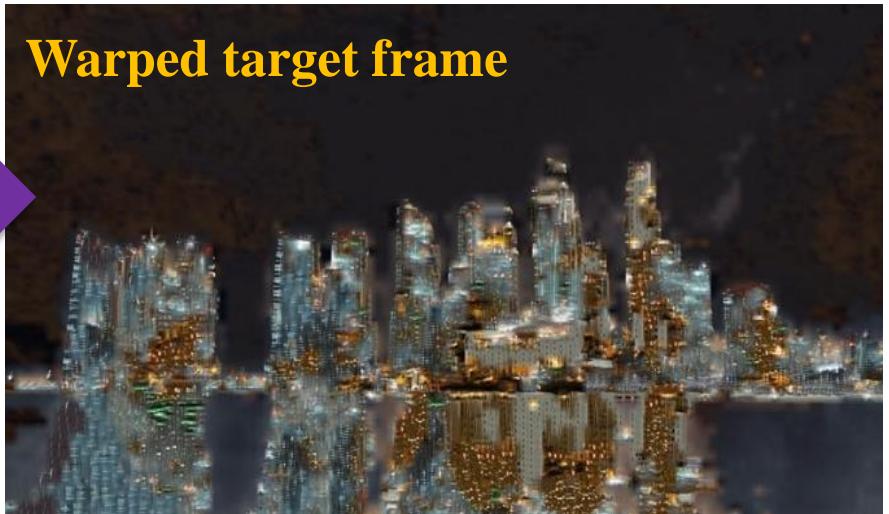
(3) Locally affine transfer from time-lapse to the input image (Sec. 6).

# Naïve 迁移方法: 像素替换

- 替换图像的纹理错乱
- 应该充分利用输入图像的纹理信息

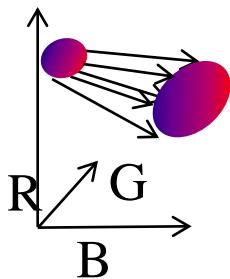
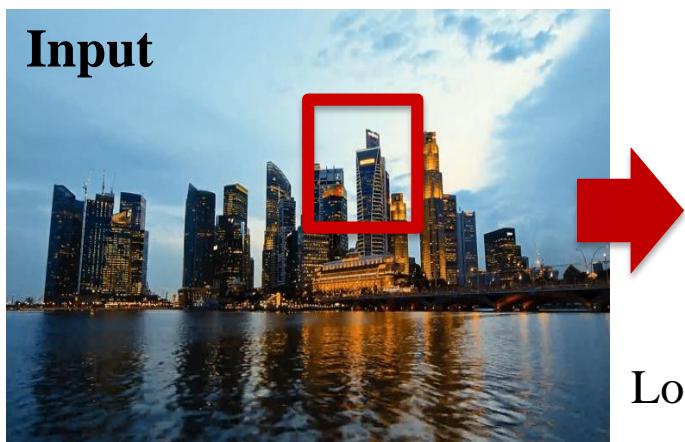


warp



# Locally affine color transfer

- *Local* : 可以处理复杂场景
- *Affine* 在每个图像块中进行颜色迁移
  - 保留输入图像的局部结构
  - 获取了延时视频中matched frame到target frame的颜色变化信息



Locally affine model



# Input at sunset



# Input at sunset



# Result at night



# 迁移效果

□ 同一输入，四个不同时间

Input



Day



Before sunset (golden hour)



After sunset (blue hour)

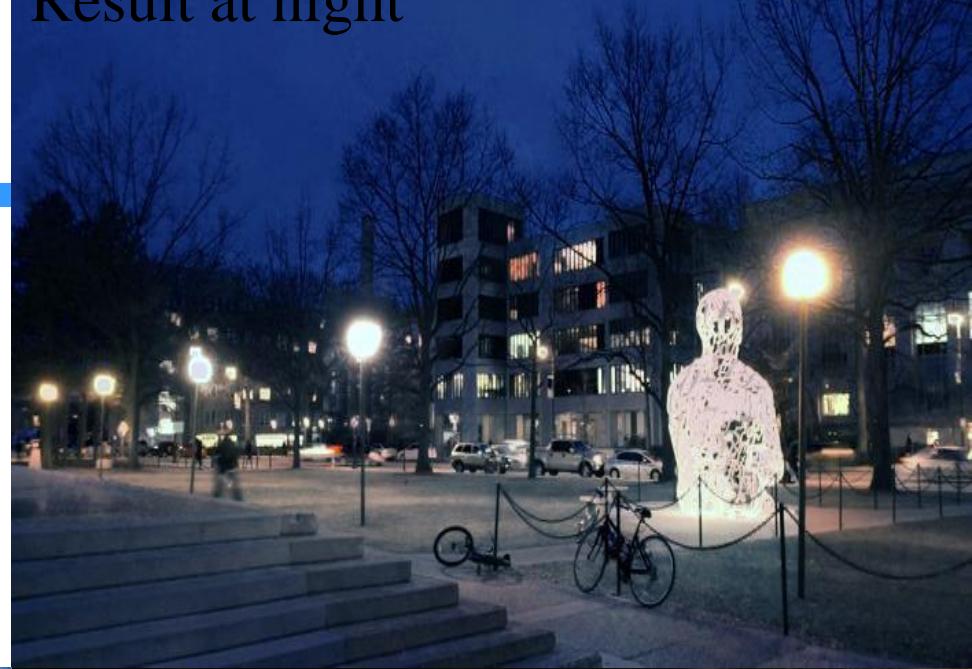


Night

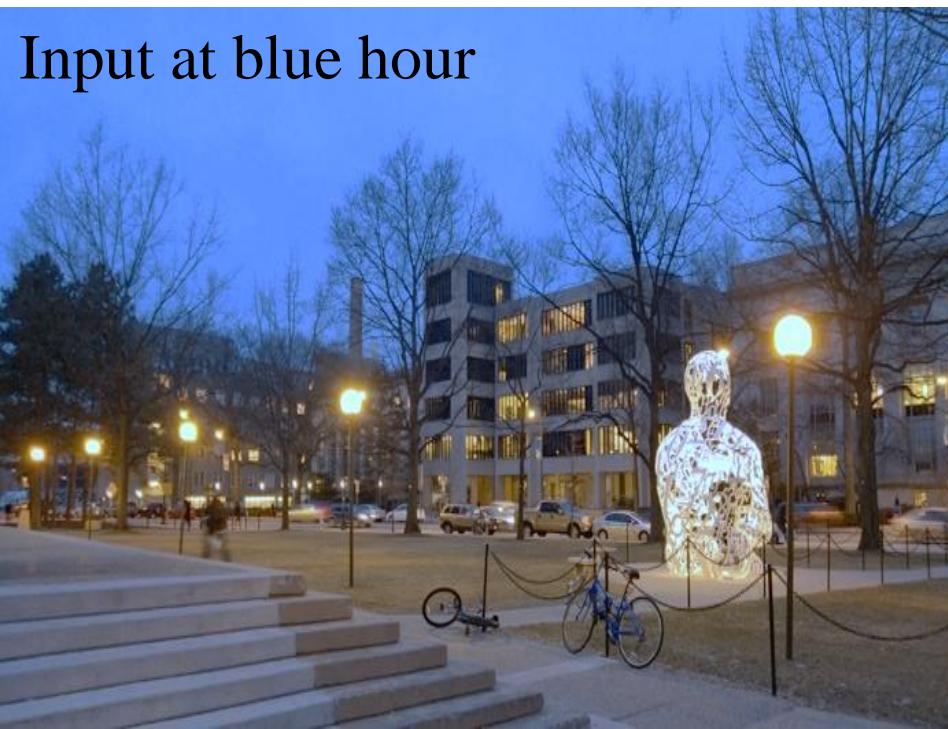


# 效果验证

Result at night



Input at blue hour

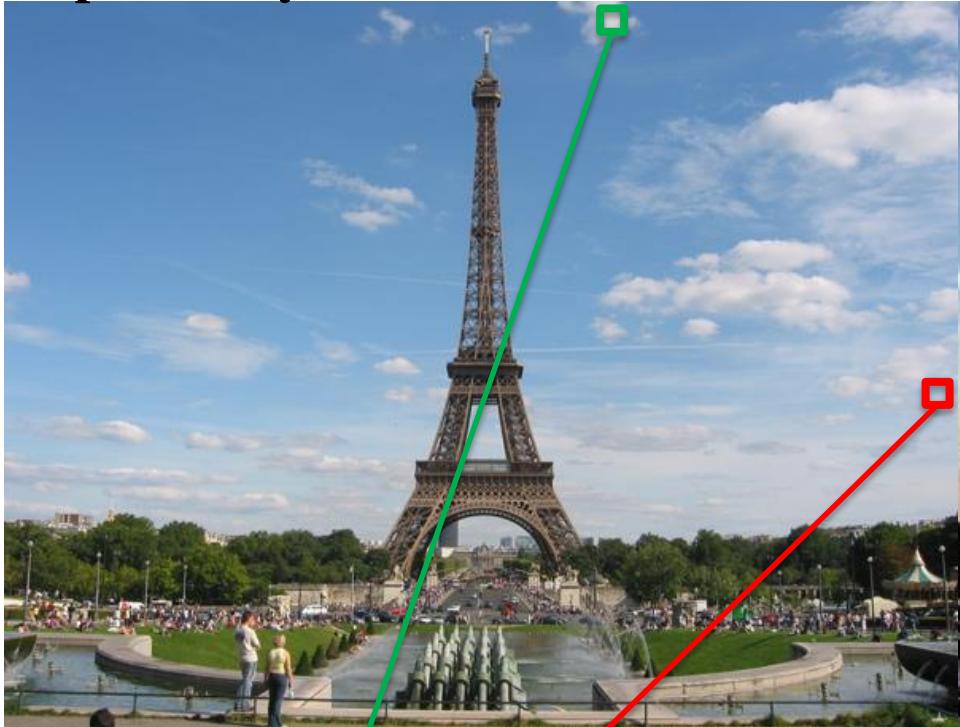


Ground truth at night

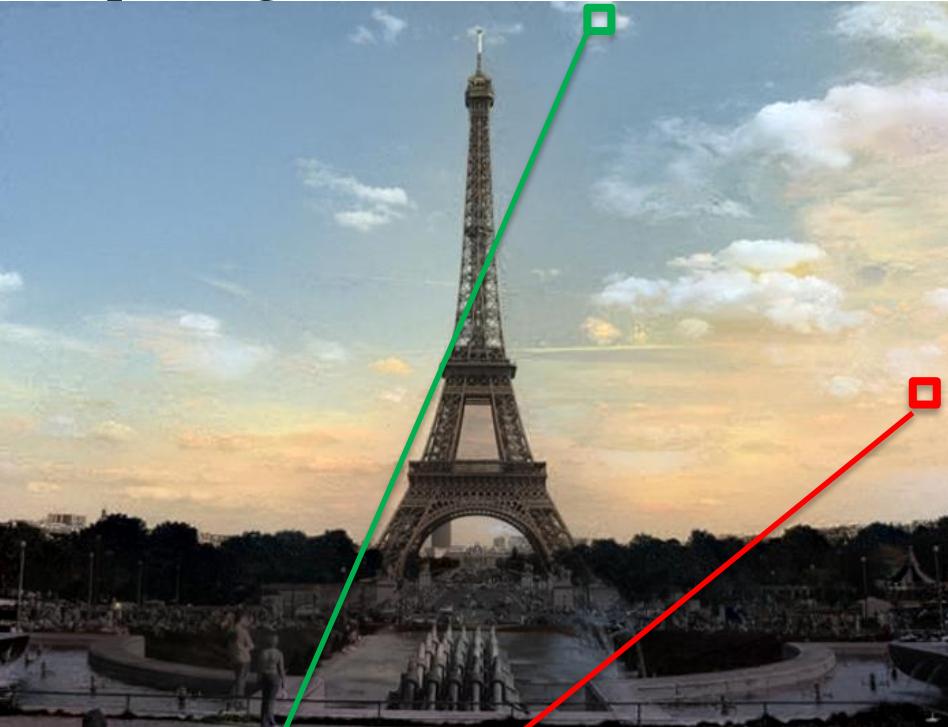


# 克服空间不一致性

Input at day

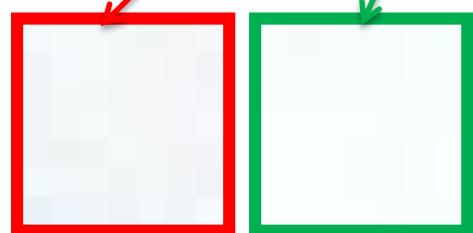


Output at golden hour



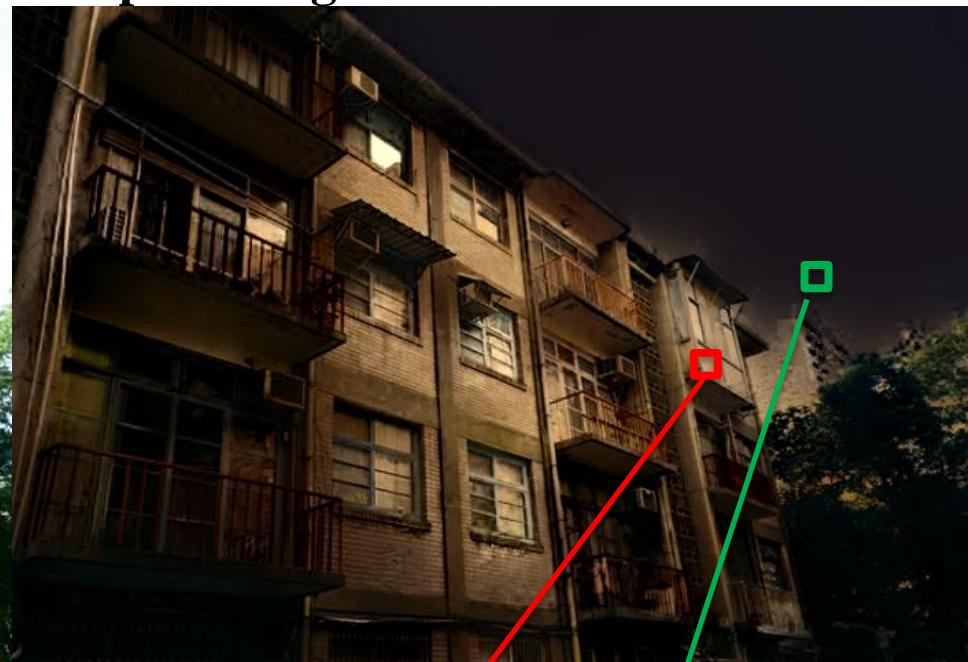
# 克服不同物体的差异

Input at day



building      sky

Output at night



building      sky

# 应用：绘画的颜色迁移



Input at day

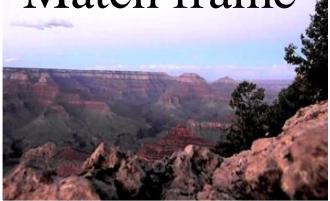


Output at blue hour

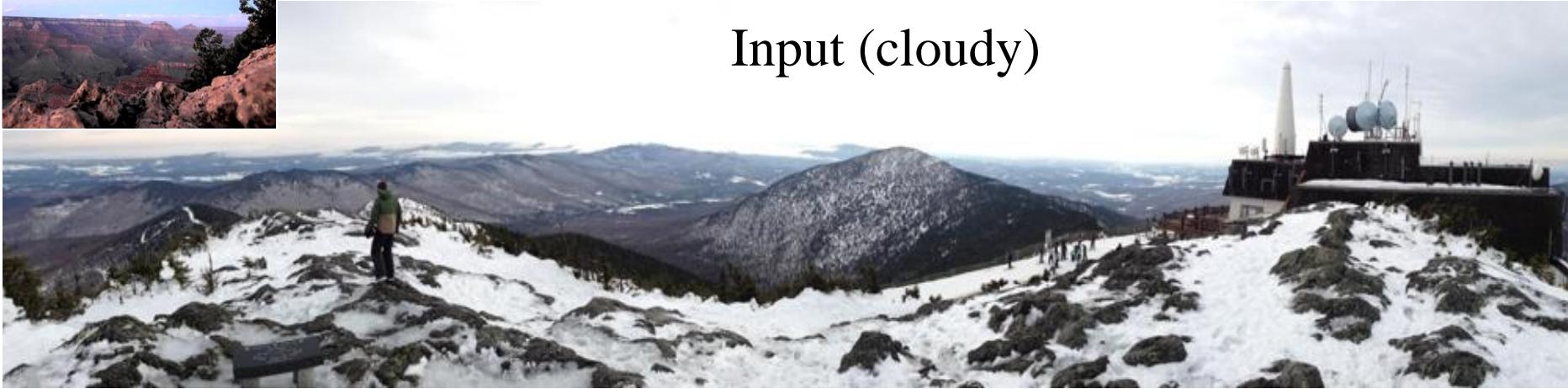
“In the Auvergne”, Jean-Francois Millet

# 应用：亮度迁移

Match frame



Input (cloudy)



Hand-picked target frame



Light transfer (more sunshine)

