《数字图像处理》
第4讲 空域图像增强

冯建江

清华大学 自动化系

2017.10.19

内　容

- 图像增强基础
- 灰度变换（Intensity transformation）
  - 基本的灰度变换函数
  - 直方图处理（Histogram processing）
- 空域滤波（Spatial filtering）
  - 平滑（Smoothing）
  - 锐化（Sharpening）

内　容

- 图像增强基础
- 灰度变换（Intensity transformation）
  - 基本的灰度变换函数
  - 直方图处理（Histogram processing）
- 空域滤波（Spatial filtering）
  - 平滑（Smoothing）
  - 锐化（Sharpening）

图像增强

输入图像　　输出图像

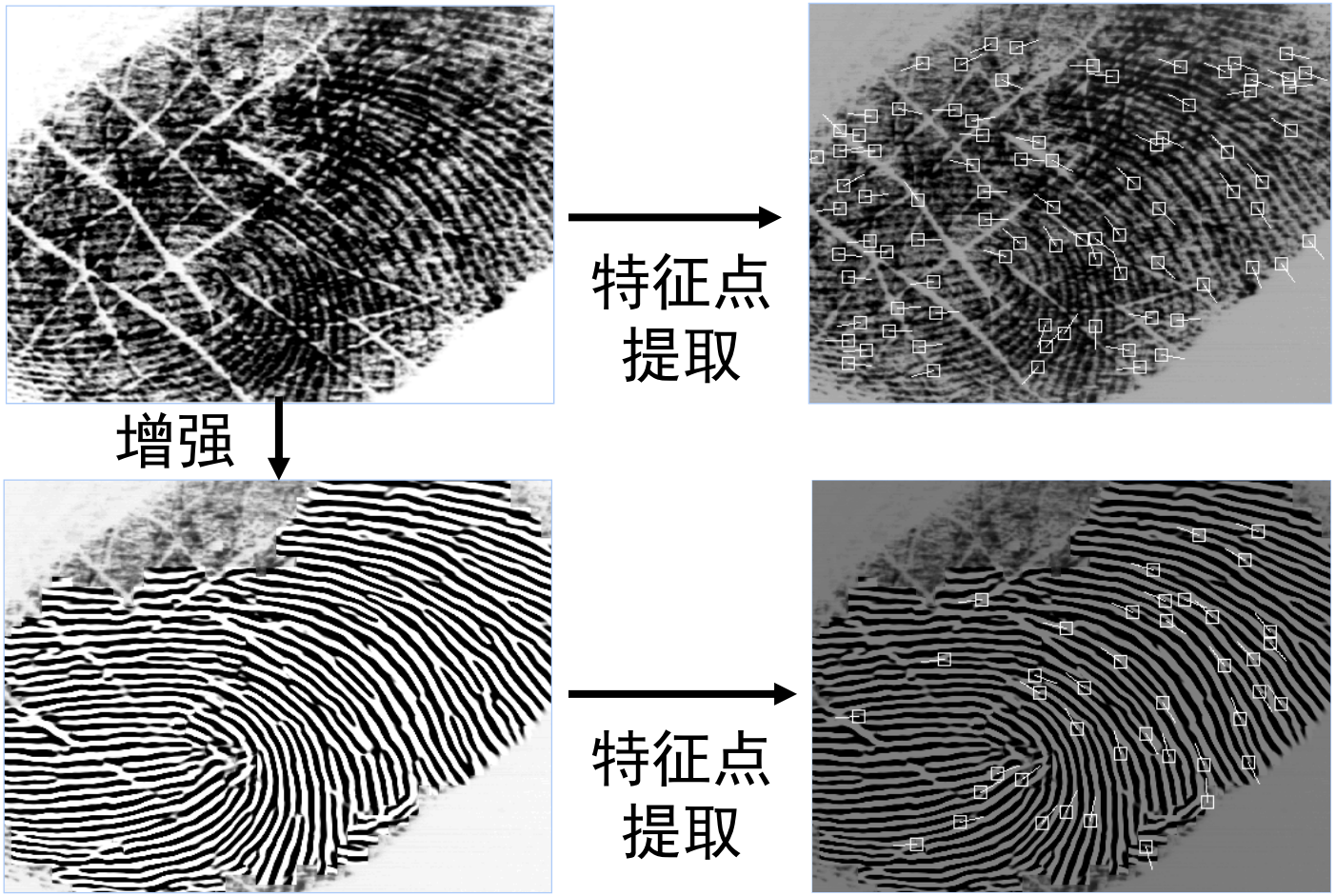
- 图像增强是一种图像处理技术，目的是改善图像，以便于人的观看或自动的图像分析与识别
- 评价标准：人的感觉或后续算法的性能

例1：增强图像便于观看

增强

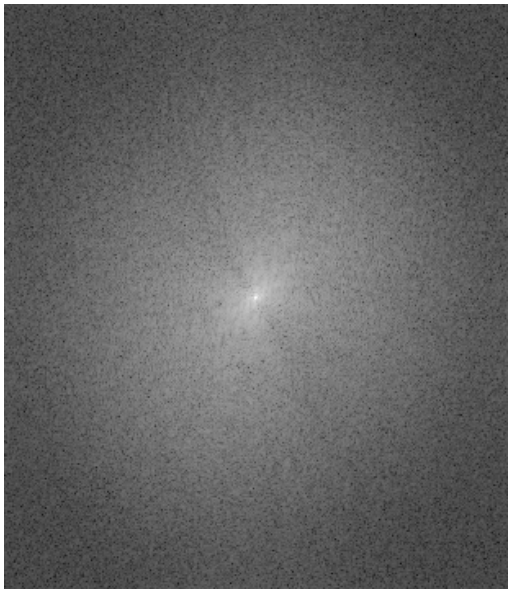
例2：增强图像便于识别

- 指纹识别系统包含2个主要模块：特征点提取和特征点匹配。
- 特征点的质量对识别性能的影响很大。



空域和频域的图像增强

月球图像



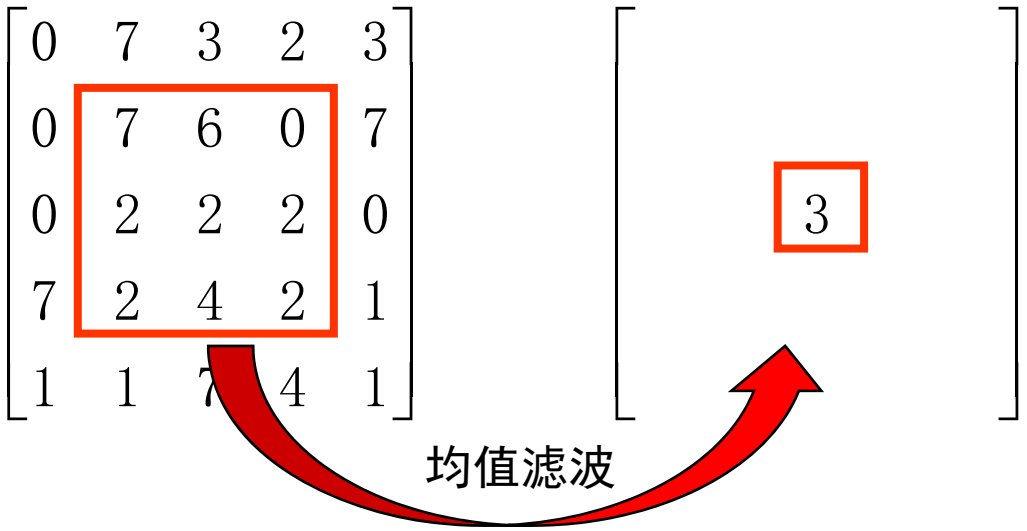
月球图像的傅里叶变换的幅度谱
(magnitude spectrum)

- 图像增强可在空域（spatial domain）或频域（frequency domain）进行
- 空域图像增强：直接修改像素值。
- 频域图像增强：先对输入图像做傅里叶变换（ Fourier transform），然后修改频谱，最后做傅里叶反变换，得到增强的图像。

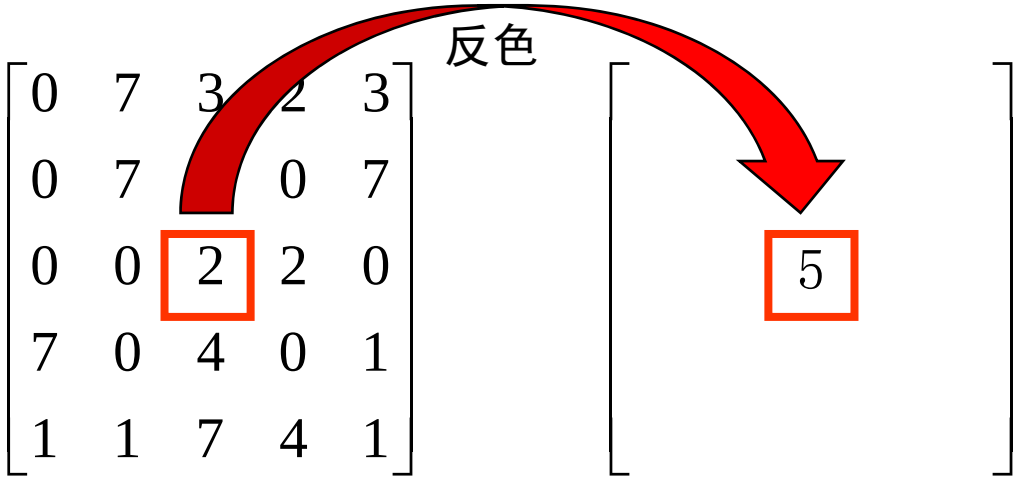
空域图像增强

输入图像$f(x,y)$

输出图像$g(x,y)$



- $g(x,y)$是对$f(x,y)$邻域进行某种运算的结果
- 邻域通常是以(x,y)为中心的矩形窗口
- 运算有很多，例如平均
- 称为空域滤波（spatial filtering），也叫邻域运算（neighborhood processing）



- 邻域大小为1×1时，运算与位置(x,y)无关
- 称为灰度变换（intensity transformation）也叫点运算（point processing）

内　容

- 图像增强基础
- 灰度变换（Intensity transformation）
  - 基本的灰度变换函数
  - 直方图处理（Histogram processing）
- 空域滤波（Spatial filtering）
  - 平滑（Smoothing）
  - 锐化（Sharpening）

灰度变换

定义：$s = T(r)$

r：输入灰度

s：输出灰度

T：变换函数

常见的变换

- 反色（image negative）
- 幂函数（power function）
- 分段线性变换函数（pairwise linear transformation function）

反　色

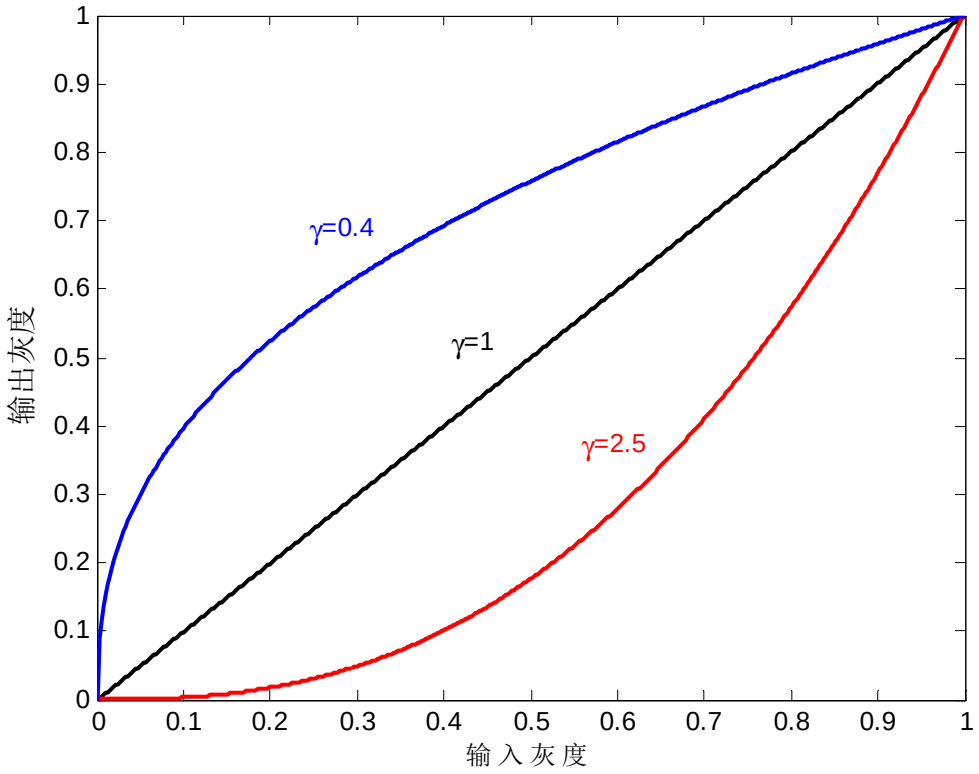
$$s = 1 - r, \quad r \in [0,1]$$

r为归一化的灰度

接触式指纹　　直接拍照（非接触式）　反色使图像一致

幂函数（power function）

$$s = r^{\gamma}, \quad r \in [0,1], \quad \gamma > 0$$



$\gamma < 1$, 图像变亮

$\gamma = 1$, 图像不变

$\gamma > 1$, 图像变暗

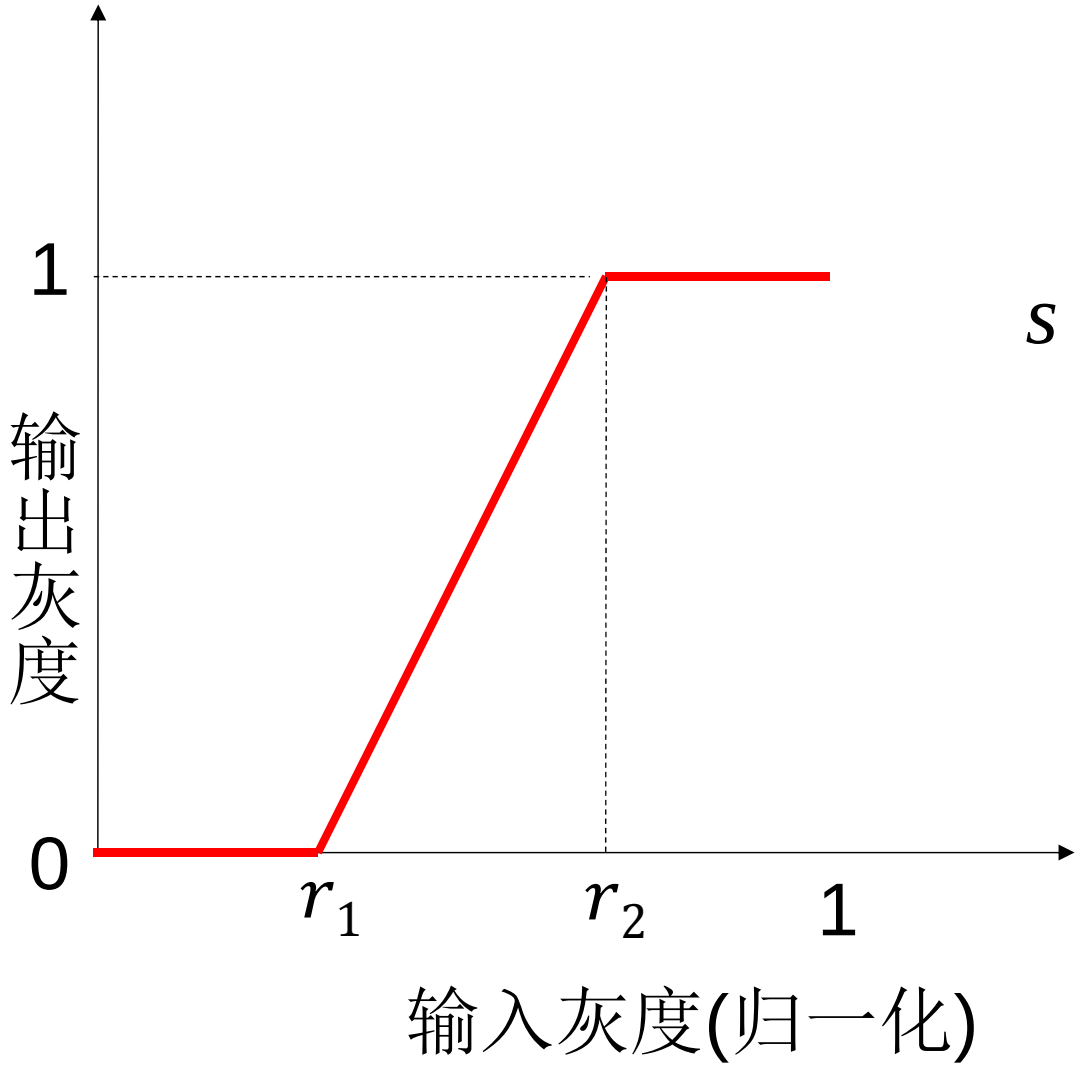
幂函数：提高对比度

$\gamma = 3$

$\gamma = 0.4$

```
I =
imread('..\data\Fig0309(a)(washed_out_aerial_im
age).tif');
% I =
imread('..\data\Fig0308(a)(fractured_spine).tif
');
I = im2double(I);
gamma = 5;
I2 = I.^gamma;
figure(1),imshow(I);
figure(2),imshow(I2);
```

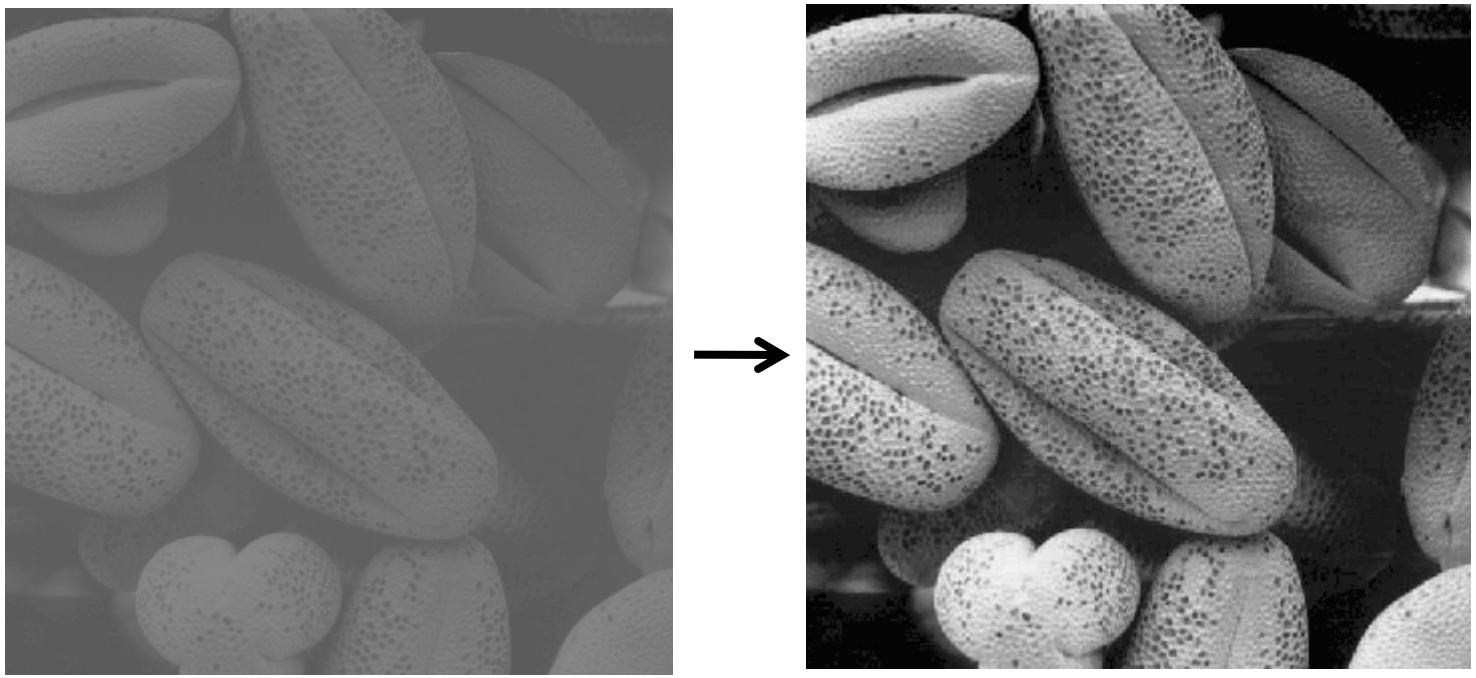
分段线性变换函数
（pairwise linear transformation）



$$s=\begin{cases}0, & r\le r_1\\ \dfrac{r-r_1}{r_2-r_1}, & r_1<r<r_2\\ 1, & r\ge r_2\end{cases}$$

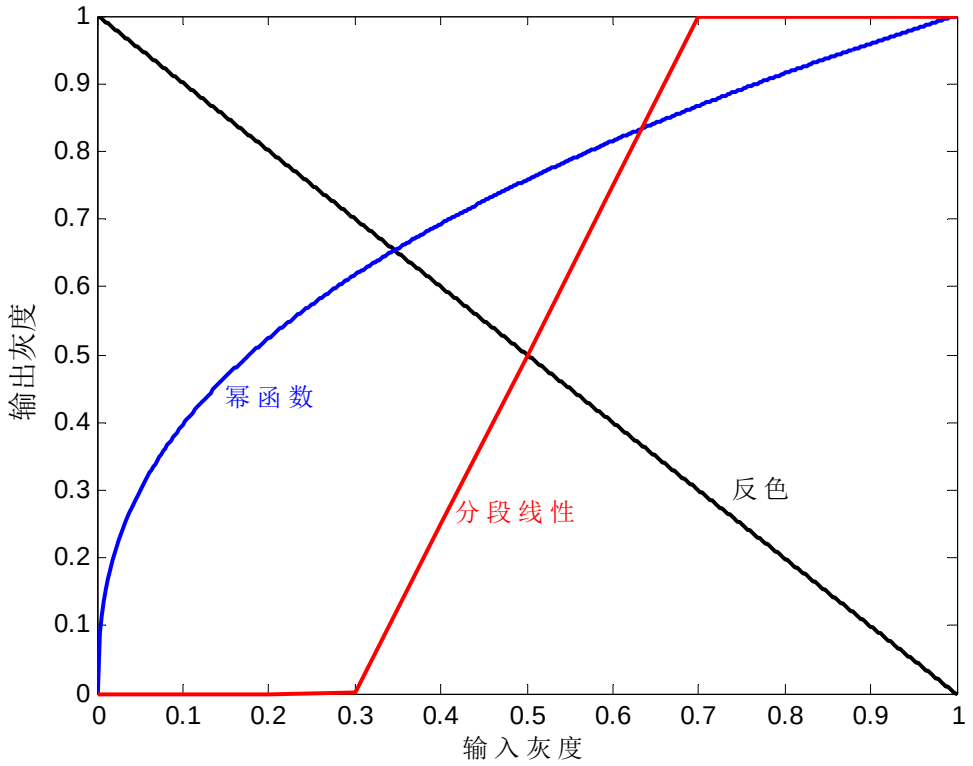
如何选择r_1和r_2 ？

分段线性变换函数



r_1=输入图像的最小灰度(0.3)
r_2=输入图像的最大灰度(0.5)

如何选择具体的灰度变换函数及参数？



- 一般需要经验和多次尝试
- 下面介绍的灰度变换方法——直方图均衡，能自动确定变换函数

内　容

- 图像增强基础
- 灰度变换（Intensity transformation）
  - 基本的灰度变换函数
  - 直方图处理（Histogram processing）
- 空域滤波（Spatial filtering）
  - 平滑（Smoothing）
  - 锐化（Sharpening）

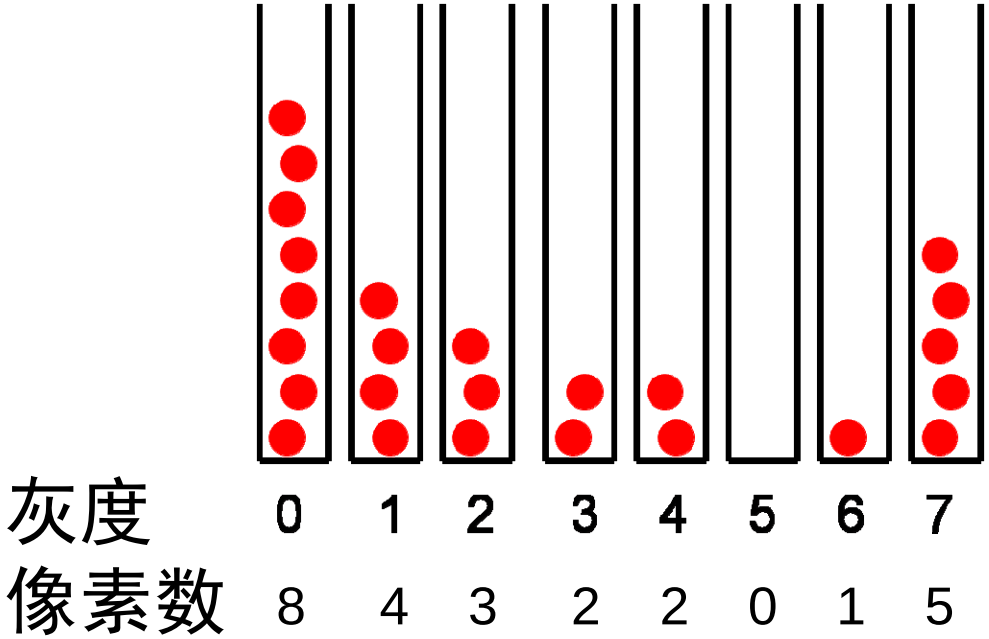
直方图（Histogram）

一幅图像的直方图统计了各灰度级在图像中的出现次数

$h(r_k) = n_k$，n_k为第k个灰度级r_k的像素数

例子：5 × 5图像，每像素占3 bits（即灰度范围[0,7]）

$$\begin{bmatrix} 0 & 7 & 3 & 2 & 3 \\ 0 & 7 & 6 & 0 & 7 \\ 0 & 0 & 2 & 2 & 0 \\ 7 & 0 & 4 & 0 & 1 \\ 1 & 1 & 7 & 4 & 1 \end{bmatrix}$$

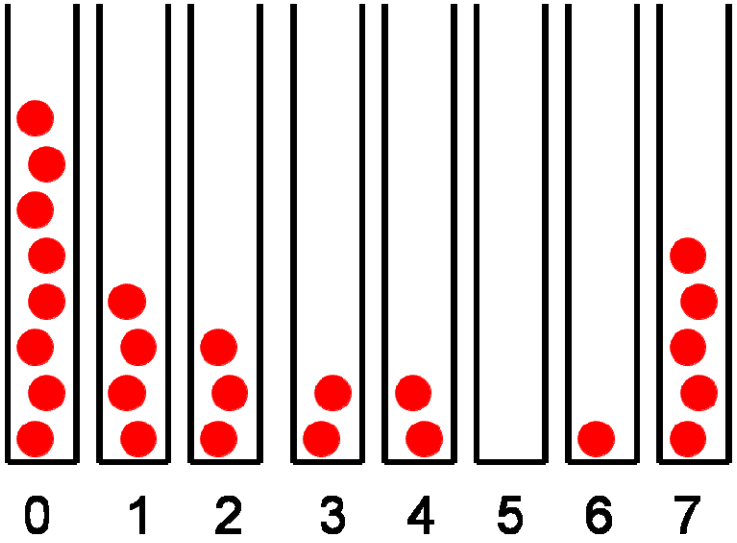


归一化的直方图（Normalized histogram）

- 归一化的直方图$p(r_k) = n_k/n$，n:图像的总像素数
- 将像素灰度看作随机变量。归一化的直方图是对灰度概率分布的估计。

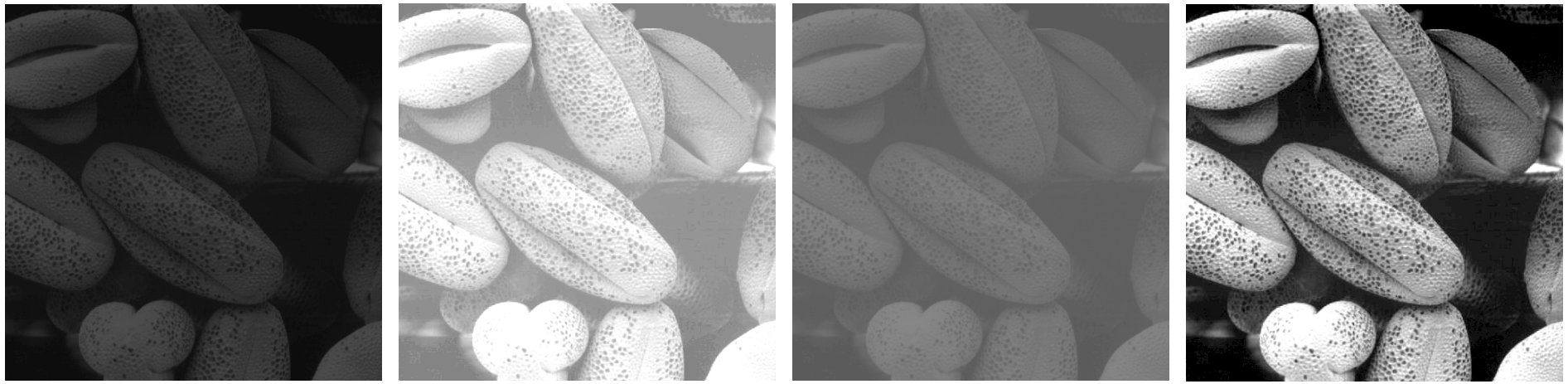
$$\begin{bmatrix} 0 & 7 & 3 & 2 & 3 \\ 0 & 7 & 6 & 0 & 7 \\ 0 & 0 & 2 & 2 & 0 \\ 7 & 0 & 4 & 0 & 1 \\ 1 & 1 & 7 & 4 & 1 \end{bmatrix}$$

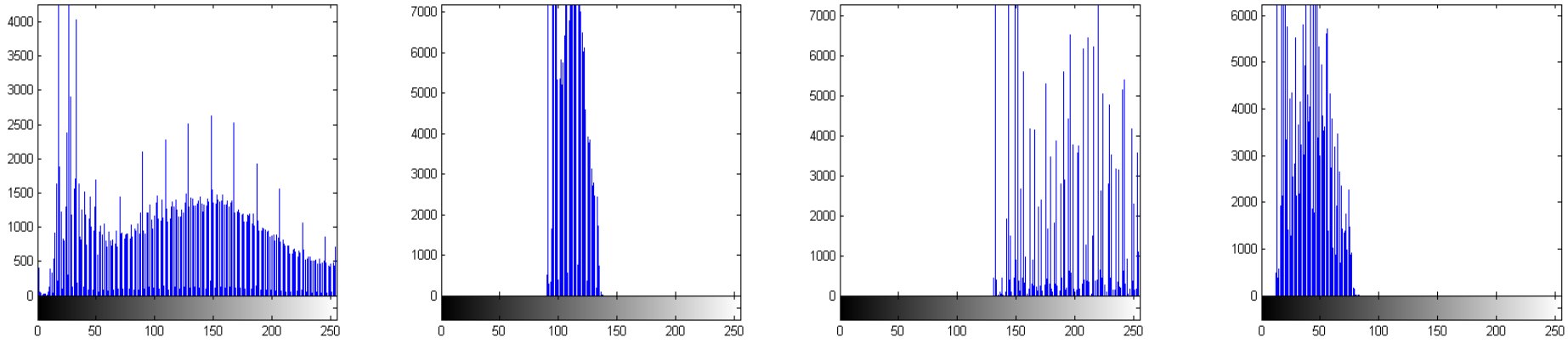
5x5图像



归一化直方图 $\frac{8}{25}, \frac{4}{25}, \frac{3}{25}, \frac{2}{25}, \frac{2}{25}, 0, \frac{1}{25}, \frac{1}{5}$

直方图的例子





(a) (b) (c) (d)

如果能对灰度进行某种变换，将直方图调整为均匀分布，那么质量就提高了。

直方图均衡：连续随机变量

已知连续随机变量r（输入图像的像素值，$0 \le r \le L-1$）的概率密度函数（PDF）$p_r(r)$，寻找函数$s = \boldsymbol{T}(r)$，使得随机变量s（输出图像的像素值）服从均匀分布（uniform distribution），即概率密度$p_s(s) = 1/(L-1)$，$0 \le s \le L-1$.

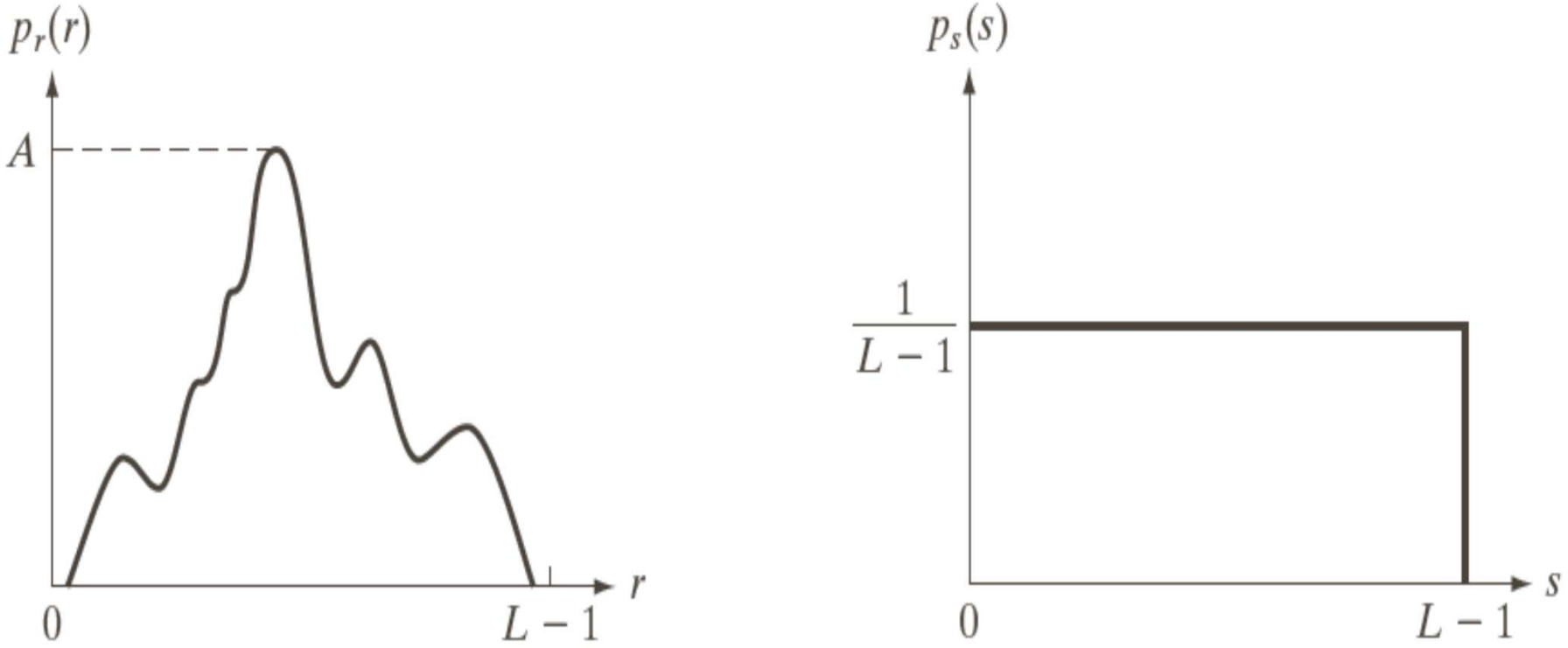
$$p_s(s) = p_r(r)\frac{dr}{ds} = \frac{1}{L-1}$$

$$\Rightarrow \frac{ds}{dr} = p_r(r)$$

r的累积分布函数
（cumulative distribution function）

$$\Rightarrow s = (L-1)\int_0^r p_r(w)dw$$

直方图均衡：连续随机变量



直观解释：将$p_r(r)$曲线下的面积均匀划分为K份，第k份对应的区间映射为$s = \frac{k(L-1)}{K}$，s服从均匀分布

直方图均衡：连续随机变量例子

输入图像灰度的概率密度函数如下：

$$p_r(r)=\begin{cases}\dfrac{2r}{(L-1)^2} & \text{for } 0\le r\le L-1\\ 0 & \text{otherwise}\end{cases}$$

计算使其直方图均衡的灰度变换函数$s=T(r)$，并验证输出图像灰度为均匀分布。

$$s=(L-1)\int_0^r \frac{2w}{(L-1)^2}dw=\frac{r^2}{L-1}$$

验证s是否服从均匀分布

$$p_s(s)=p_r(r)\left|\frac{dr}{ds}\right|=\frac{2r}{(L-1)^2}\left|\left[\frac{ds}{dr}\right]^{-1}\right|=\frac{2r}{(L-1)^2}\frac{L-1}{2r}=\frac{1}{L-1}$$

直方图均衡：离散随机变量

数字图像的灰度值是离散随机变量

连续随机变量 $s = (L-1)\int_0^r p_r(w)dw$

离散随机变量 $s_k = (L-1)\sum_{j=0}^{k} p_r(r_j)$

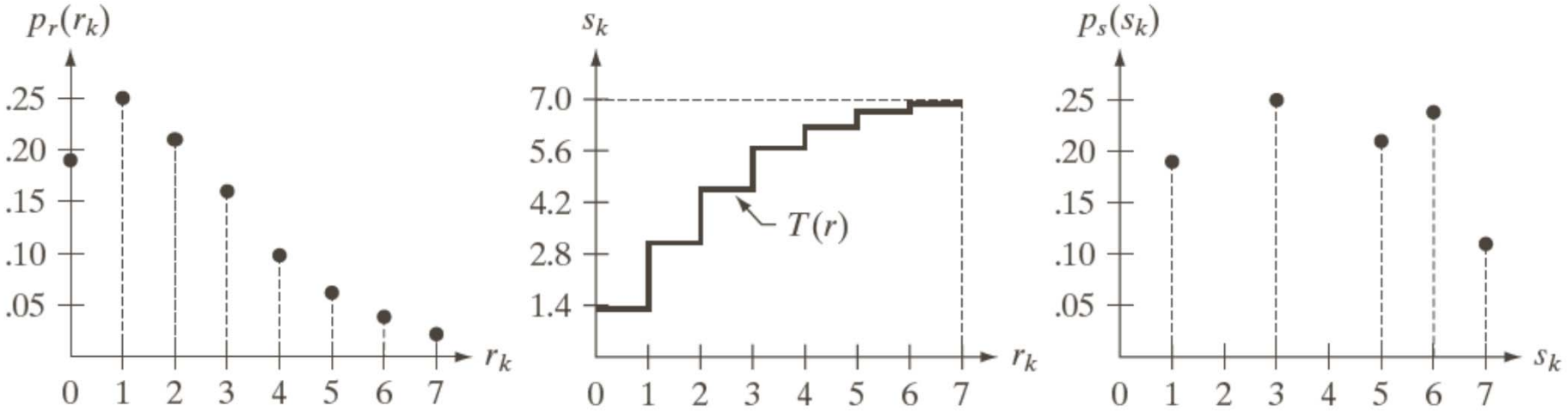
$k = 0,1,2,\ldots,L-1$

直方图均衡：离散随机变量例子

输入图像大小64 × 64像素，像素深度为3比特。灰度分布如下表。对其进行直方图均衡。

r_k	n_k	$p_r(r_k) = n_k/MN$
$r_0 = 0$	790	0.19
$r_1 = 1$	1023	0.25
$r_2 = 2$	850	0.21
$r_3 = 3$	656	0.16
$r_4 = 4$	329	0.08
$r_5 = 5$	245	0.06
$r_6 = 6$	122	0.03
$r_7 = 7$	81	0.02

直方图均衡：离散随机变量例子

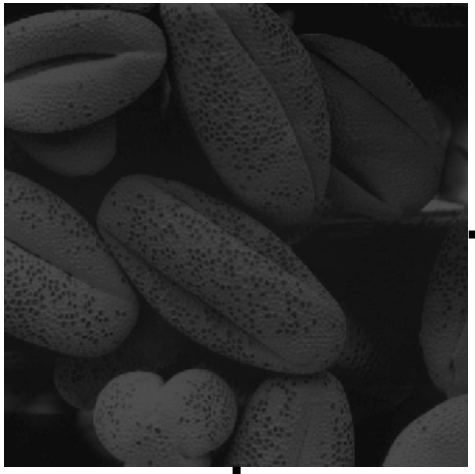


a b c

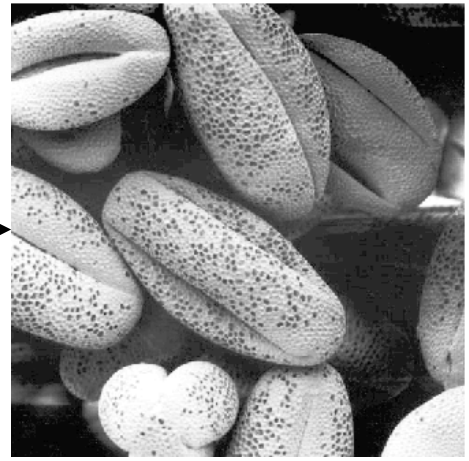
FIGURE 3.19 Illustration of histogram equalization of a 3-bit (8 intensity levels) image. (a) Original histogram. (b) Transformation function. (c) Equalized histogram.

- 对连续随机变量，直方图均衡使输出服从均匀分布
- 而对离散随机变量，直方图均衡不能保证输出图像的直方图为均匀分布（通常都不是）

直方图均衡的流程图

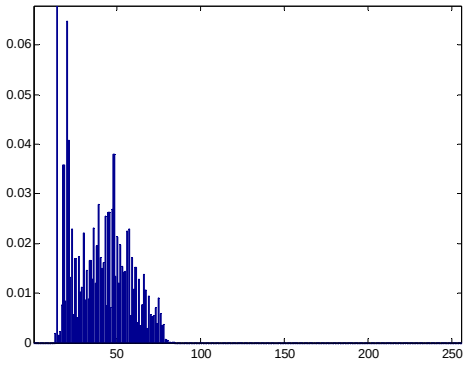


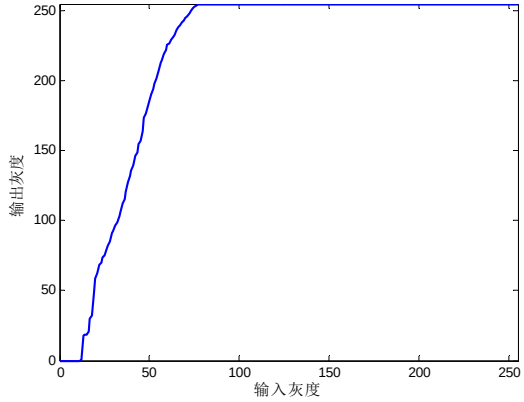
灰度变换



输入图像

输出图像





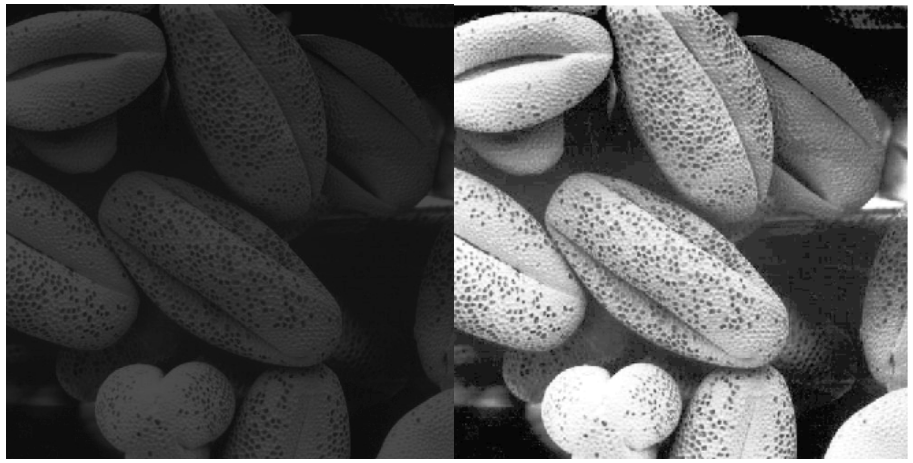
直方图

累积直方图（变换函数）

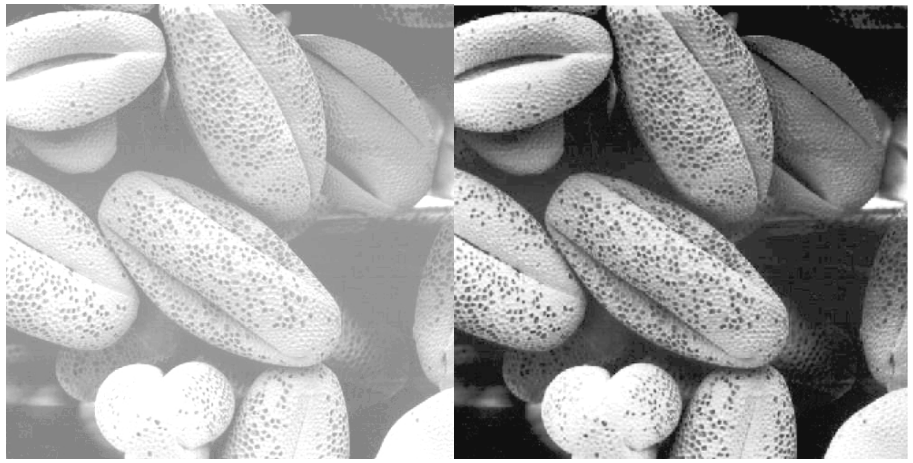
histEqStep.m

```
close all
I=imread('..\data\Fig0320(1)(top_left).tif');
h = imhist(I,256);
h = h/sum(h);
ch = cumsum(h);
figure(1),imshow(I);
figure(2),bar(h);
axis tight
figure(3),plot([0:255],ch*255),axis
tight,xlabel('ÊäÈë»Ò¶È'),ylabel('Êä³ö»Ò¶È')

J = histeq(I);
figure(4),imshow(J);
h = imhist(J,256);
h = h/sum(h);
figure(5),bar(h);
```

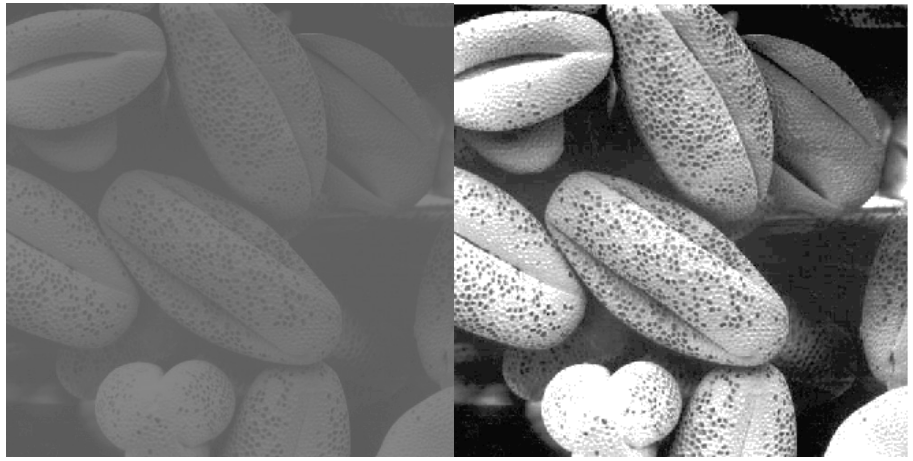
直方图均衡：例子



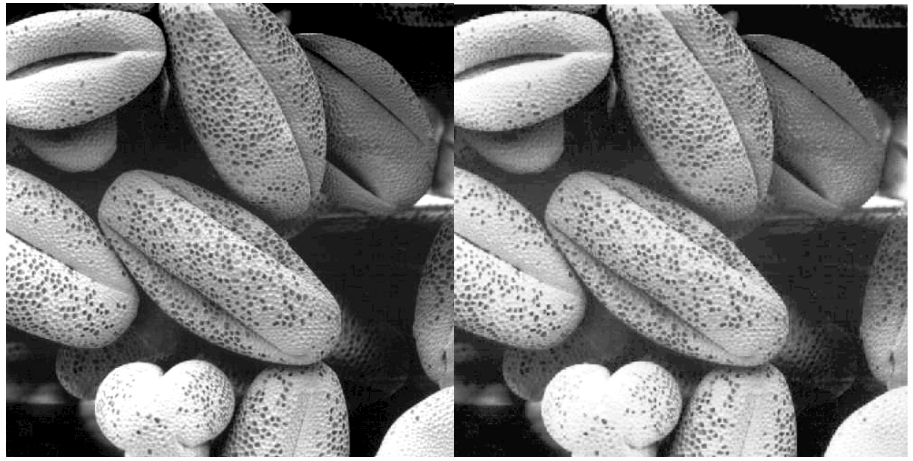
对比度低，暗



对比度低，亮



对比度很低，灰

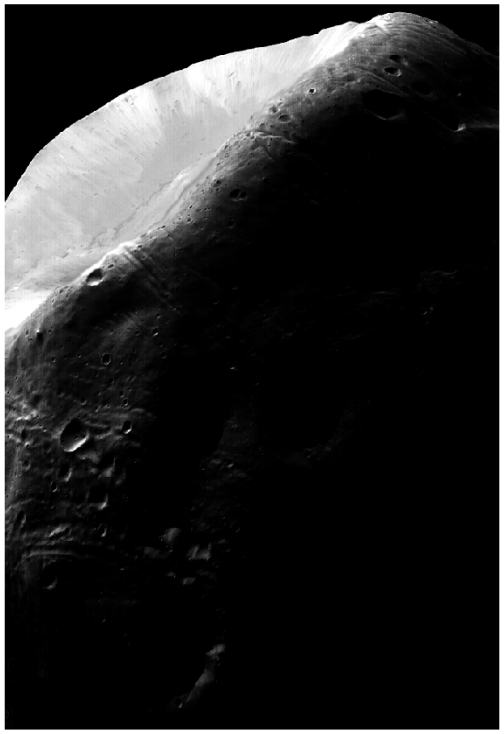


对比度高

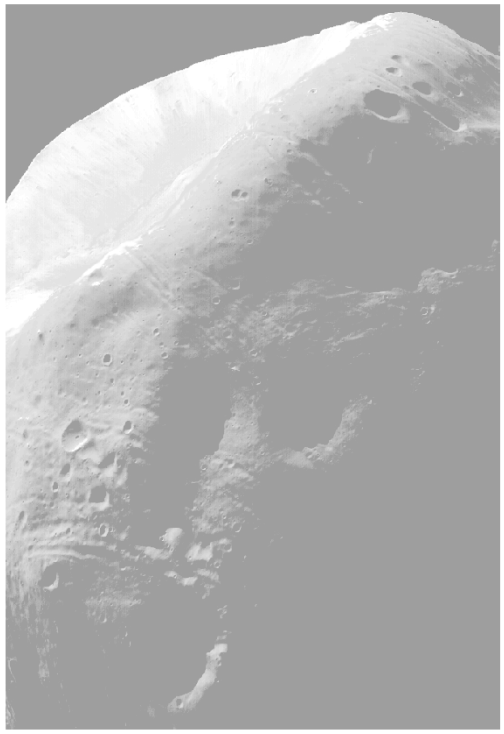
直方图匹配：连续随机变量

- 直方图匹配（histogram matching），也叫直方图定制（histogram specification），使输出图像的直方图与指定的直方图$p_z(z)$一致
- 方法：
  - 先将输入图像做直方图均衡，得到其变换$s = T(r)$
  - 再对规定直方图做直方图均衡，得到其变换$s = G(z)$
  - 计算其反变换$z = G^{-1}(s)$

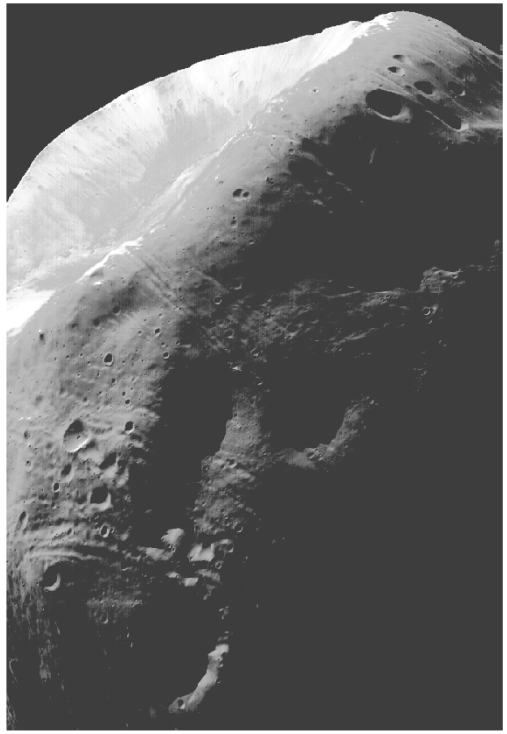
直方图匹配：例子



火卫一



直方图均衡

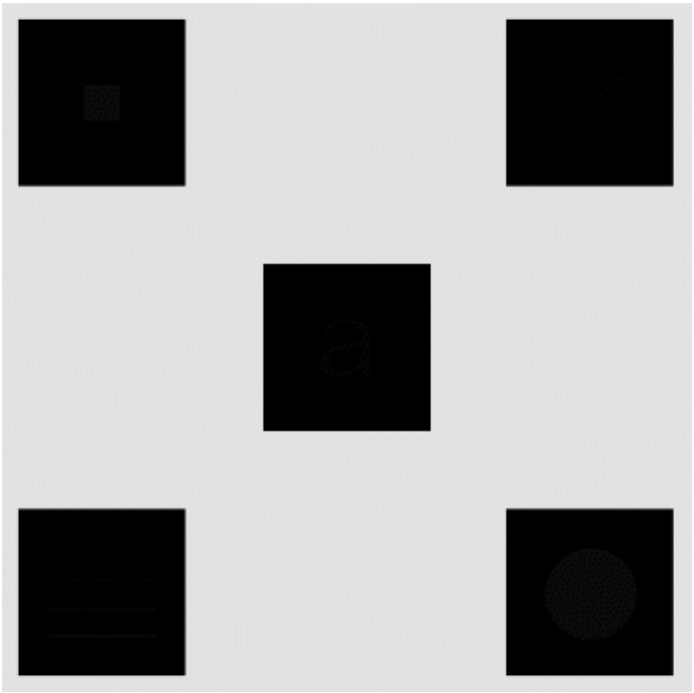


直方图匹配

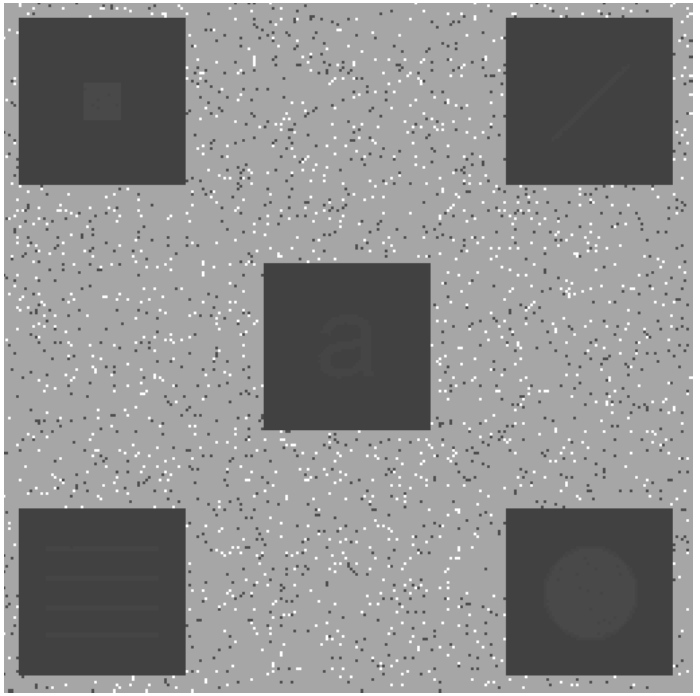
```
close all
I = imread('..\data\Fig0323(a)(mars_moon_phobos).tif');
[J1, T1] = histeq(I);

p = manualhist;
%the following arguments were typed at the prompt
% 0.15 0.05 0.75 0.05 1 0.1 0.002  - input is looped don't forget
to end with x
[J2, T2] = histeq(I, p);

figure(1),
ax(1)=subplot(1,3,1); imshow(I);
ax(2)=subplot(1,3,2); imshow(J1);
ax(3)=subplot(1,3,3); imshow(J2);
linkaxes(ax);

figure(2), subplot(2,2,1), imhist(I);
subplot(2,2,2), imhist(J1);
subplot(2,2,3), stem(0:255,p, 'Marker', 'none');
subplot(2,2,4), imhist(J2);

figure(3), subplot(1,2,1), plot(0:255, 255*T1), xlim([0 255]),
ylim([0 255]);
subplot(1,2,2), plot(0:255, 255*T2), xlim([0 255]), ylim([0 255]);
```

局部直方图

- 前面的直方图处理考虑整幅图像的灰度分布
- 图像的某些局部的灰度分布未必和全局一致
- 全局直方图处理对于某些局部未必合适



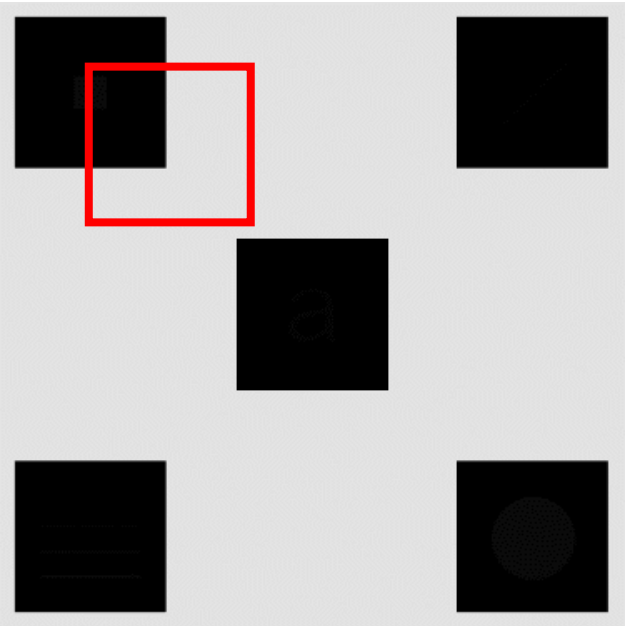
原始图像

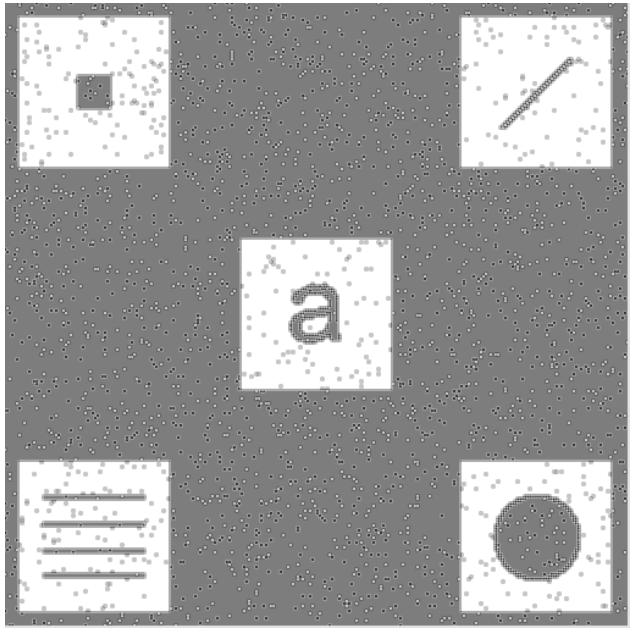


全局直方图均衡

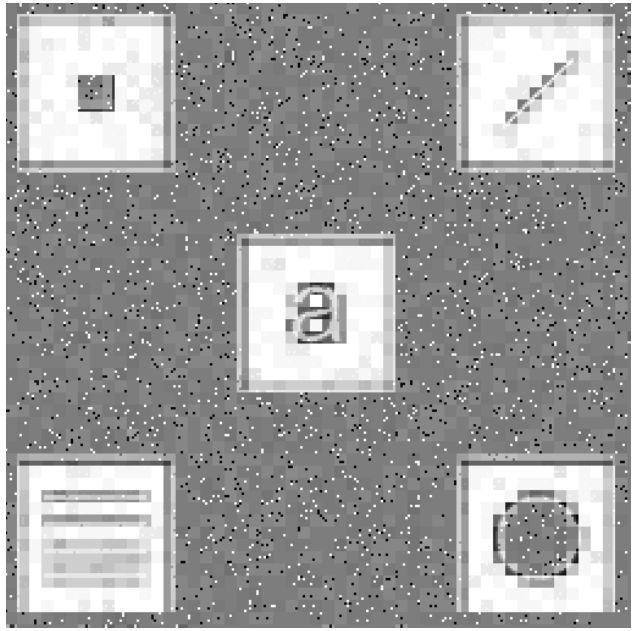
局部直方图处理

- 对于每个像素，在其邻域内计算直方图，求取灰度变换函数，对该像素进行变换
- 加速策略：分块进行（缺点是块效应）





逐像素局部直方图均衡



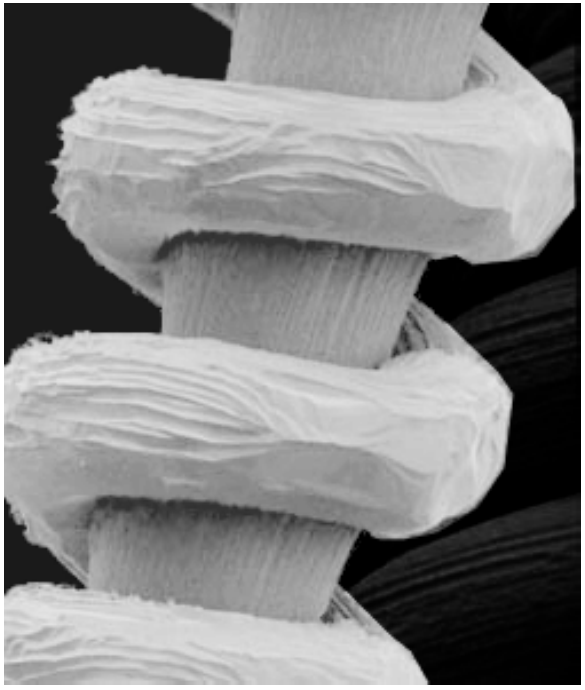
分块局部直方图均衡

```
I = imread('..\data\Fig0326(a)(embedded_square_noisy_512).tif');
figure(1),imshow(I),set(gcf,'name','Original');
%% Global histogram equalization
tic
J1 = histeq(I);
time1 = toc
figure(2),imshow(J1),set(gcf,'name','Global histogram equalization');
%% Local histogram equalization (pixelwise)
if 0
    tic
    n = 1;% neighborhood size (2*n+1)*(2*n+1)
    J2 = I;
    [H, W] = size(I);
    for r = 1+n:H-n
        for c = 1+n:W-n
            local_image = I(r-n:r+n,c-n:c+n);
            new_local_image = histeq(local_image);
            J2(r,c) = new_local_image(n+1,n+1);
        end
end
    time2 = toc
    figure(3),imshow(J2),set(gcf,'name','Pixelwise local histogram equalization');
end
%% Local histogram equalization (blockwise)
if 1
    tic
    h = @(block_struct) histeq(block_struct.data);
    J3 = blockproc(I,[10 10],h);
    time3 = toc
    figure(4),imshow(J3),set(gcf,'name','Blockwise local histogram equalization');
end
```

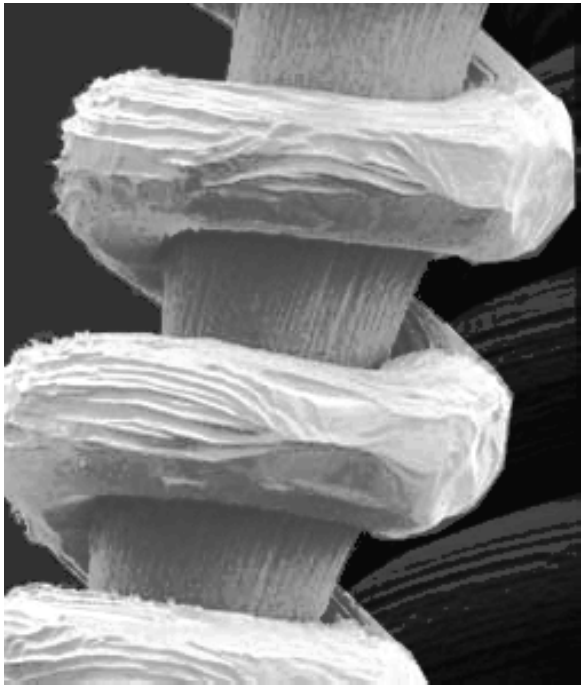
利用直方图统计量进行图像增强

- 图像的归一化直方图可视为对其像素值概率分布的估计
- 利用归一化直方图，可得到均值、方差（二阶矩）
- 如果算法只需要均值和方差，则无需计算直方图，可直接从图像计算均值和方差

利用局部统计量进行图像增强



钨丝的电子显微照片



全局直方图均衡

利用局部统计量增强

```
I = imread('..\data\Fig0327(a)(tungsten_original).tif');
figure(1),imshow(I),set(gcf,'name','Original');

%% Global histogram equalization
J1 = histeq(I);
figure(2),imshow(J1),set(gcf,'name','Global histogram equalization');

%% Enhancement using local histogram statistics
I = double(I);
mean_global = mean2(I);
std_global = std2(I);
fun1 = @(x) mean2(x);
mean_local = nlfilter(I,[3 3],fun1);% slow
fun2 = @(x) std2(x);
std_local = nlfilter(I,[3 3],fun2);% slow
figure(3),imshow(mean_local,[]),set(gcf,'name','Local mean');
figure(4),imshow(std_local,[]),set(gcf,'name','Local standard deviation');

k0 = 0.4; k1 = 0.02; k2 = 0.4; E = 4;
mask = (mean_local<=k0*mean_global) & (std_local>=k1*std_global) &
(std_local<=k2*std_global);
J2 = I;
J2(mask) = I(mask)*E;
figure(5),imshow(uint8(J2)),set(gcf,'name','Enhancement by local
statistics');
```

内　容

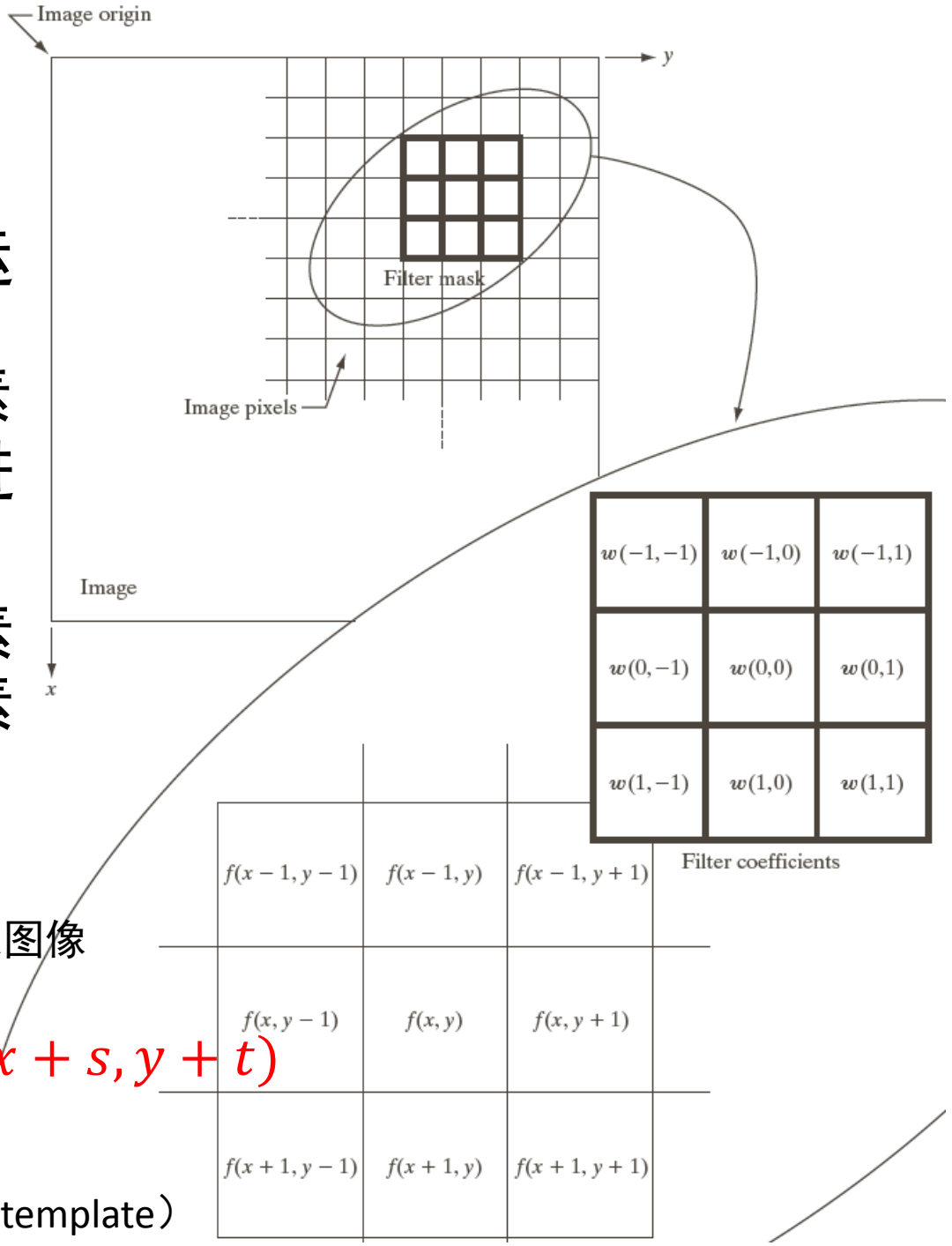
- 图像增强基础
- 灰度变换（Intensity transformation）
  - 基本的灰度变换函数
  - 直方图处理（Histogram processing）
- 空域滤波（Spatial filtering）
  - 平滑（Smoothing）
  - 锐化（Sharpening）

滤波（filtering）

- Wikipedia:
  - “A filter is a device or process that removes some unwanted components or features from a signal.”
  - “Most often, this means removing some frequencies and not others in order to suppress interfering signals and reduce background noise.”
- 滤波原指频域信号处理
- 随着计算机、DSP发展，越来越多的信号（图像）处理是在时间（空间）域进行

空域滤波

- 空域滤波是一种邻域运算（neighborhood operation），输出像素值是对输入像素邻域进行某种运算的结果
- 线性滤波器：输出像素值是输入像素邻域像素值的线性组合
- 反之，为非线性滤波



$$g(x,y) = \sum_{s=-1}^{1}\sum_{t=-1}^{1} w(s,t) f(x+s, y+t)$$

输入图像

输出图像

滤波器

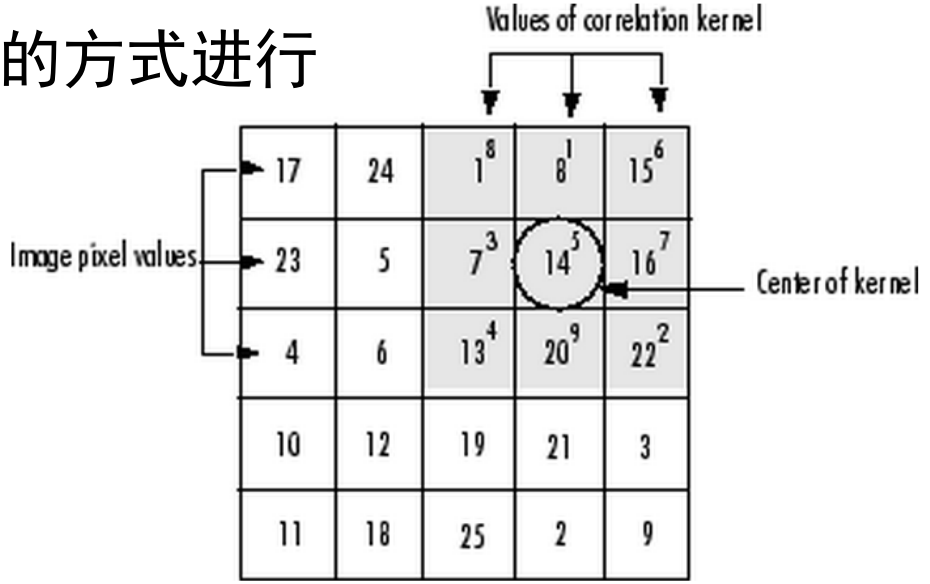
（filter, kernel, mask, window, template）

相关（Correlation）与卷积（Convolution）

空域滤波可通过相关或者卷积的方式进行

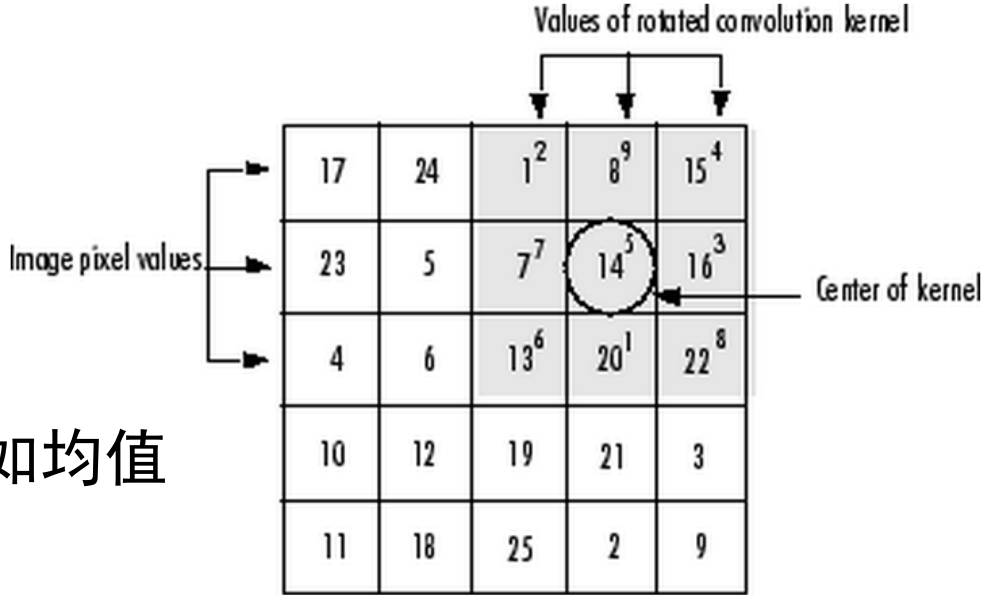
输入图像

```
A = [17   24    1    8   15
     23    5    7   14   16
      4    6   13   20   22
     10   12   19   21    3
     11   18   25    2    9]
```



滤波器

```
h = [8    1    6
     3    5    7
     4    9    2]
```



如滤波器为180度对称（例如均值滤波器），二者结果一样

内　容

- 图像增强基础
- 灰度变换（Intensity transformation）
  - 基本的灰度变换函数
  - 直方图处理（Histogram processing）
- 空域滤波（Spatial filtering）
  - 平滑（Smoothing）
  - 锐化（Sharpening）

平滑滤波器

- 平滑滤波器的用途：
  - 模糊（blurring）：去除小细节
  - 去噪
- 平滑滤波器分为线性和非线性

均值滤波器

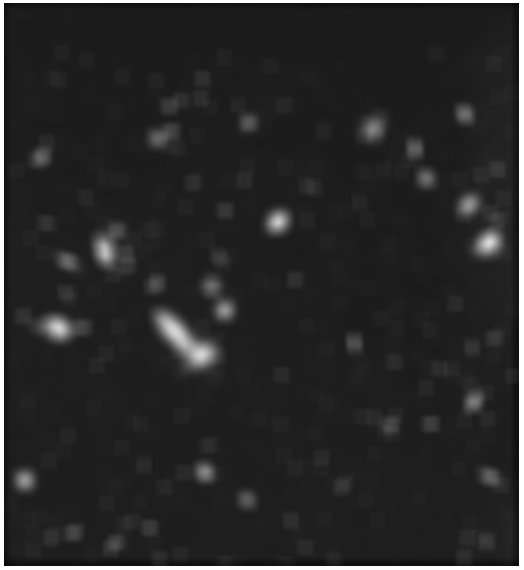
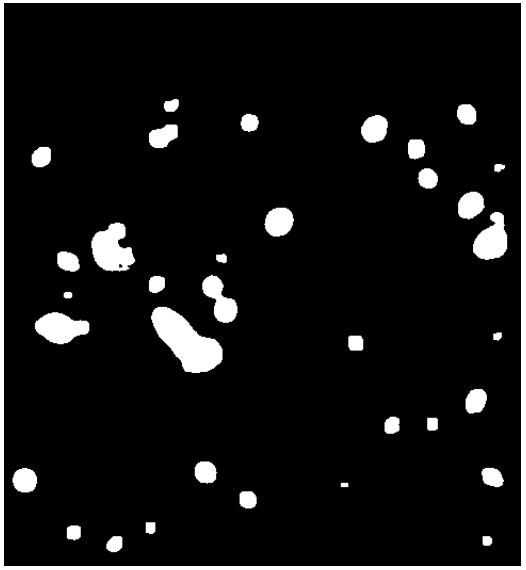
均值滤波器
（averaging filter）

$$\frac{1}{9}\times\begin{bmatrix}1&1&1\\1&1&1\\1&1&1\end{bmatrix}$$

加权均值滤波器
（weighted averaging filter）

$$\frac{1}{16}\times\begin{bmatrix}1&2&1\\2&4&2\\1&2&1\end{bmatrix}$$

均值滤波例子

哈勃太空望远镜所拍图像

```
close all
I = imread('..\data\Fig0334(a)(hubble-original).tif');
J1 = imfilter(I,ones(15,15)/(15*15));
threshold = 0.25*max(J1(:));
J2 = J1>threshold;
figure(1),clf
ax(1)=subplot(1,3,1); imshow(I);
ax(2)=subplot(1,3,2); imshow(J1);
ax(3)=subplot(1,3,3); imshow(J2);
linkaxes(ax);
```

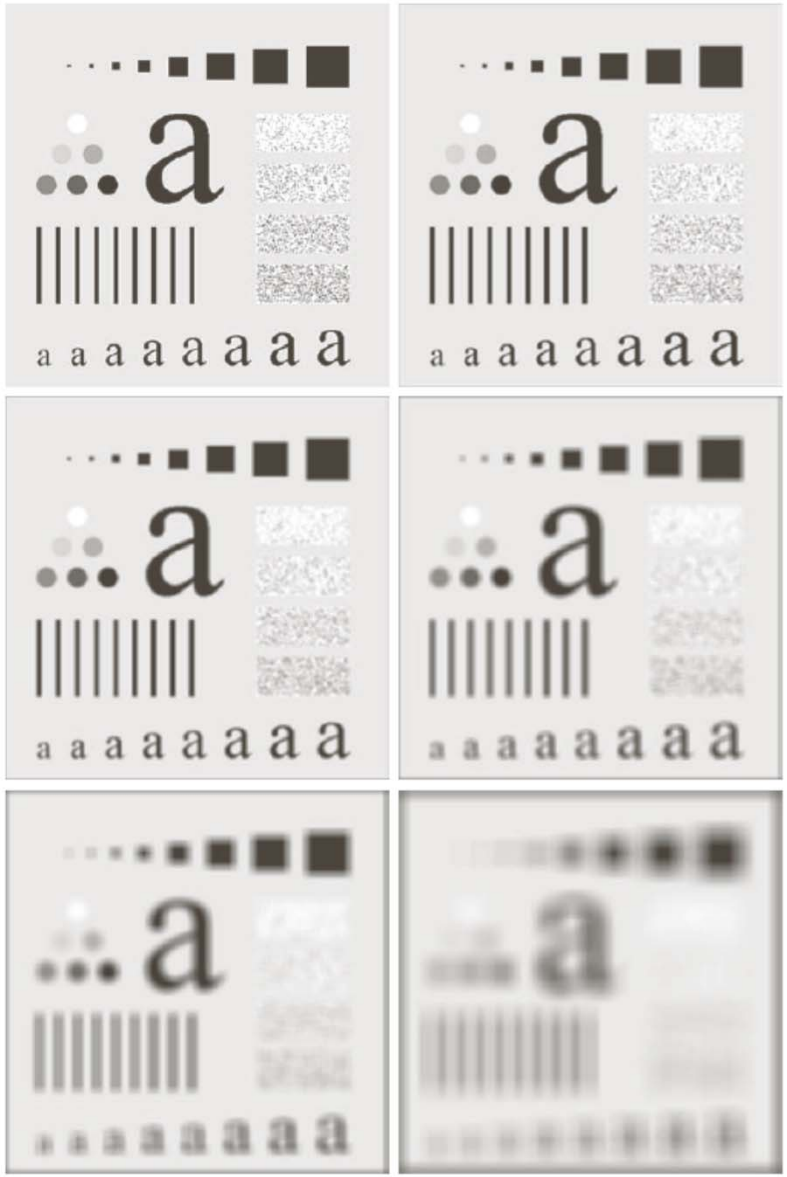
均值滤波器的尺寸

不同尺寸均值滤波器的滤波结果；

原始图像，3×3

$5 \times 5, 9 \times 9$

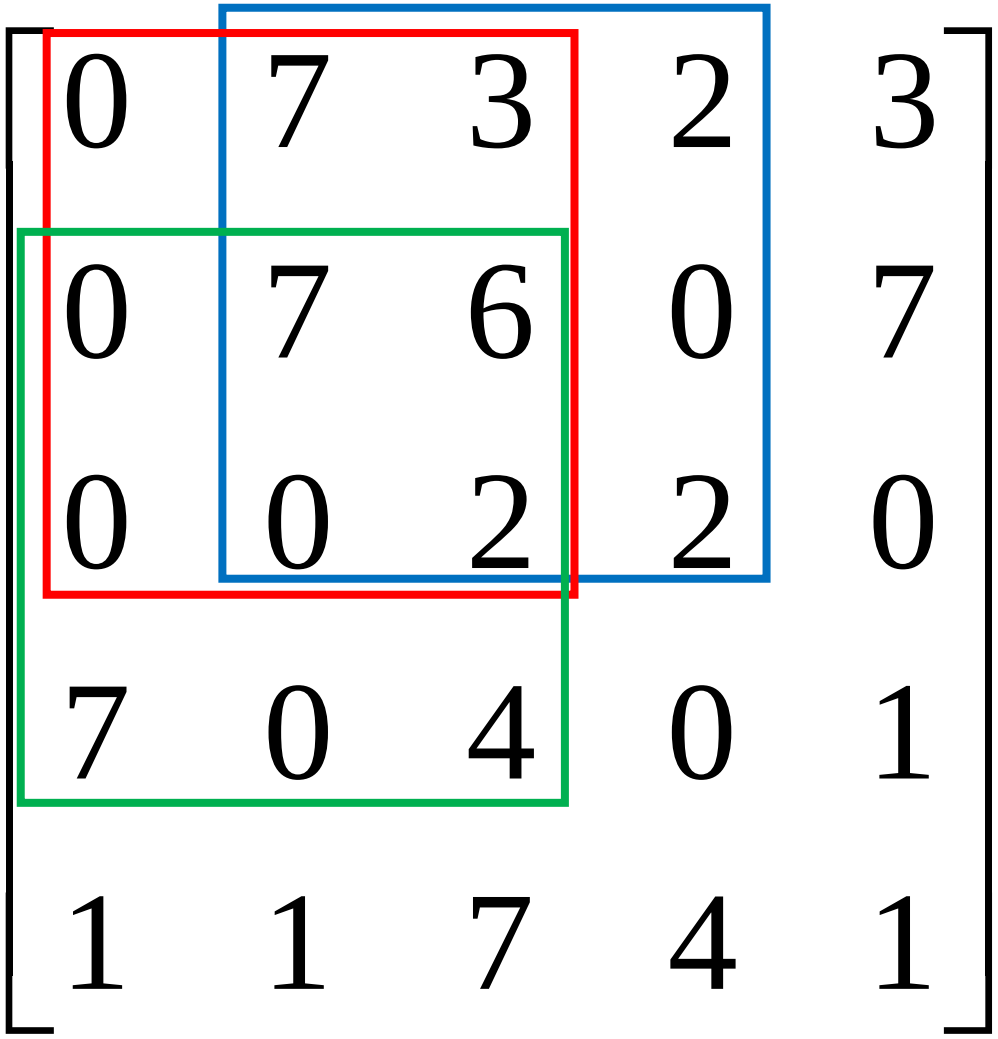
$15 \times 15, 35 \times 35$



```
close all
I = cell(1,6);
I{1} =
imread('..\data\Fig0333(a)(test_pattern_blurring_orig).tif');
sizes = [3 5 9 15 35];

for i = 1:5
tic
I{i+1} =
imfilter(I{1},ones(sizes(i),sizes(i))/(sizes(i)*sizes(i)));
toc
end

figure(1),clf
ax = zeros(1,6);
for i = 1:6
    ax(i)=subplot(2,3,i);imshow(I{i});
end
linkaxes(ax);
```

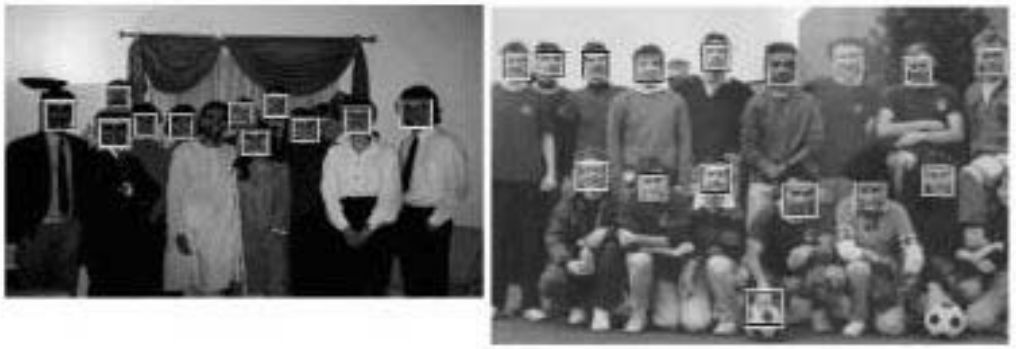
均值滤波能否快速计算？

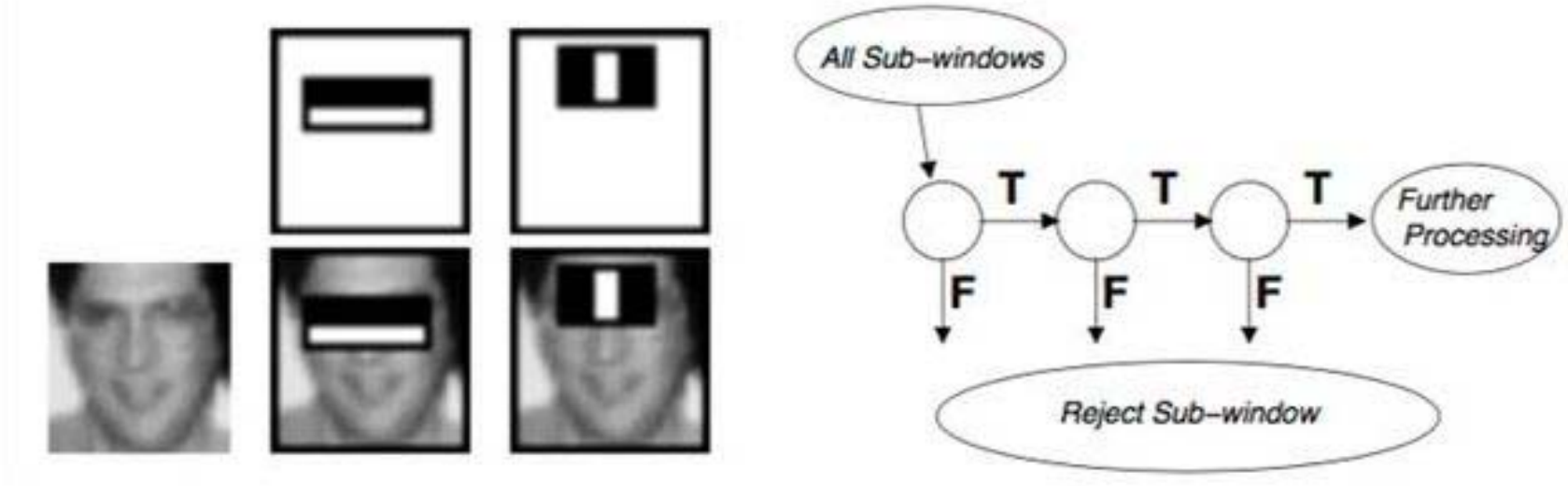
- 实际应用中，算法的速度很重要
- 通常，线性滤波的计算复杂度 n^2MN
- 滤波器尺寸$n \times n$
- 图像尺寸$M \times N$



均值滤波的快速算法

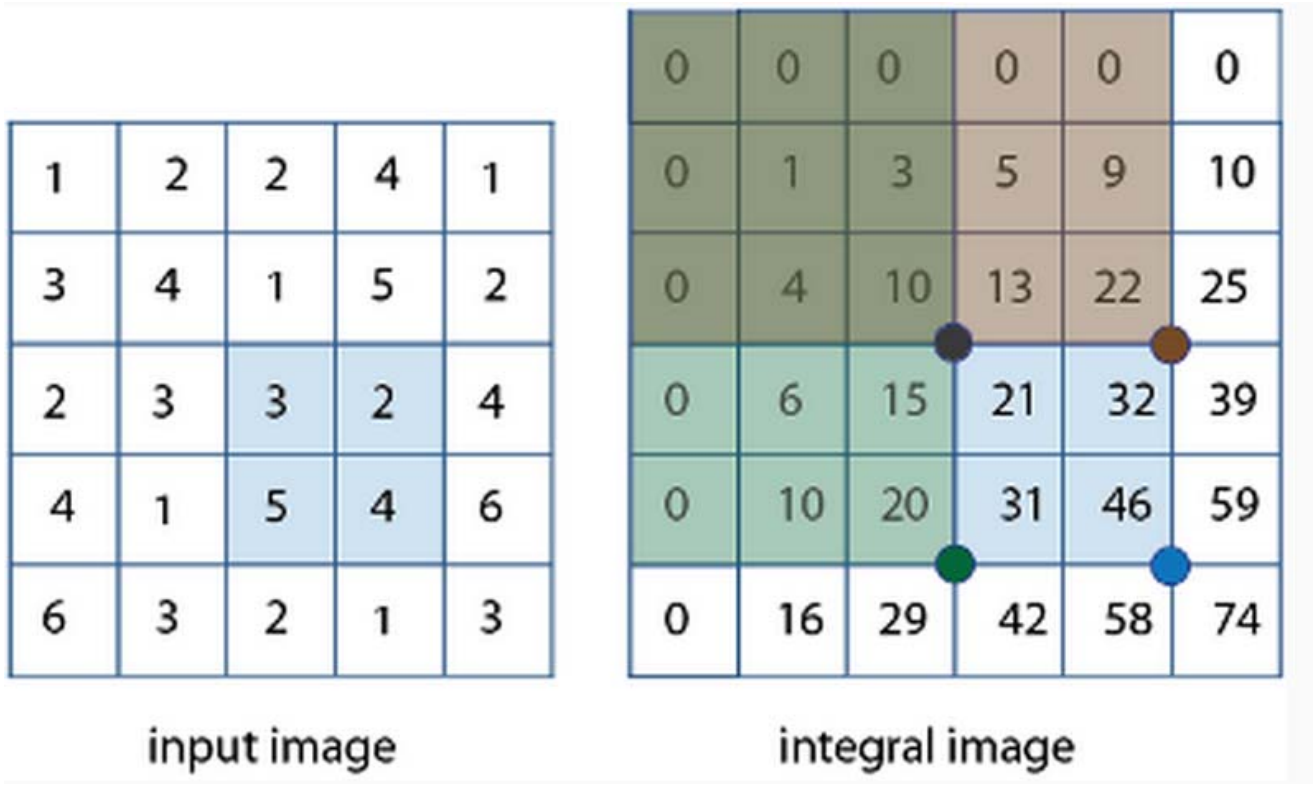
利用积分图像（Integral image）可加速均值滤波





Paul Viola and Michael J. Jones, “Rapid Object Detection using a Boosted Cascade of Simple Features”, *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR),* pp.511–518. Google Scholar引用次数>13000

均值滤波的快速算法



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

input image

0	0	0	0	0	0
0	1	3	5	9	10
0	4	10	13	22	25
0	6	15	21	32	39
0	10	20	31	46	59
0	16	29	42	58	74

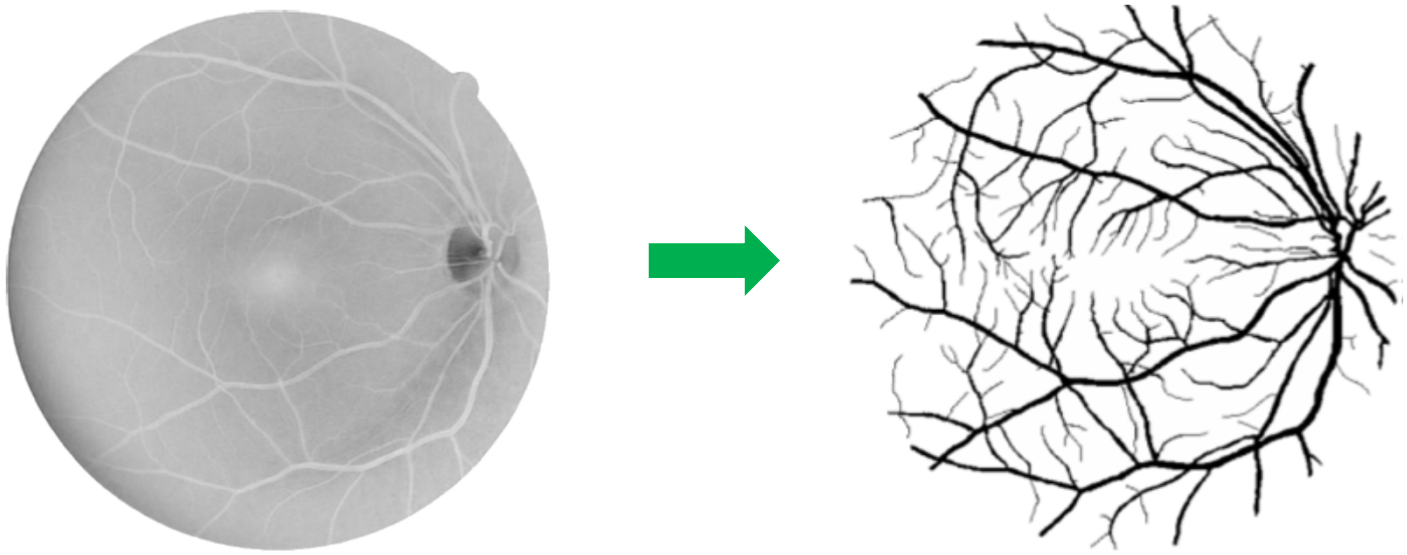
integral image

$3 + 2 + 5 + 4 = 14$ $46 - 22 - 20 + 10 = 14$

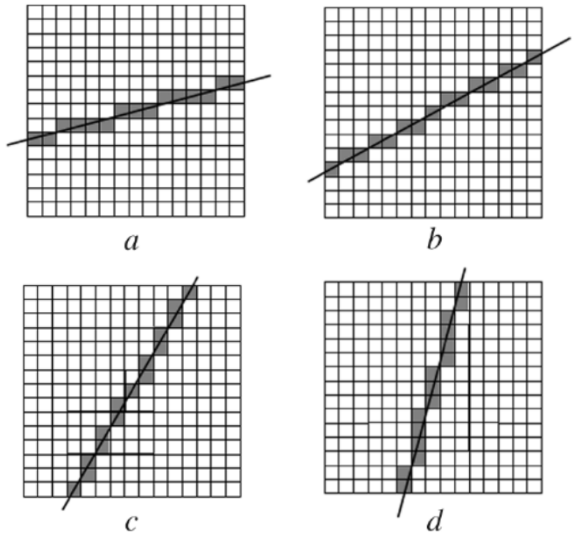
Matlab, Computer Vision System Toolbox

```
close all
I = cell(1,6);
I{1} = imread('..\data\Fig0333(a)(test_pattern_blurring_orig).tif');
sizes = [3 5 9 15 35];

intImage = integralImage(I{1});
for i = 1:5
tic
    avgH = integralKernel([1 1 sizes(i) sizes(i)], 1/(sizes(i)*sizes(i)));
    I{i+1} = integralFilter(intImage, avgH);
    I{i+1} = uint8(I{i+1});
toc
end

figure(1),clf
ax = zeros(1,6);
for i = 1:6
    ax(i)=subplot(2,3,i);imshow(I{i});
end
linkaxes(ax);
```

增强线条的均值滤波器



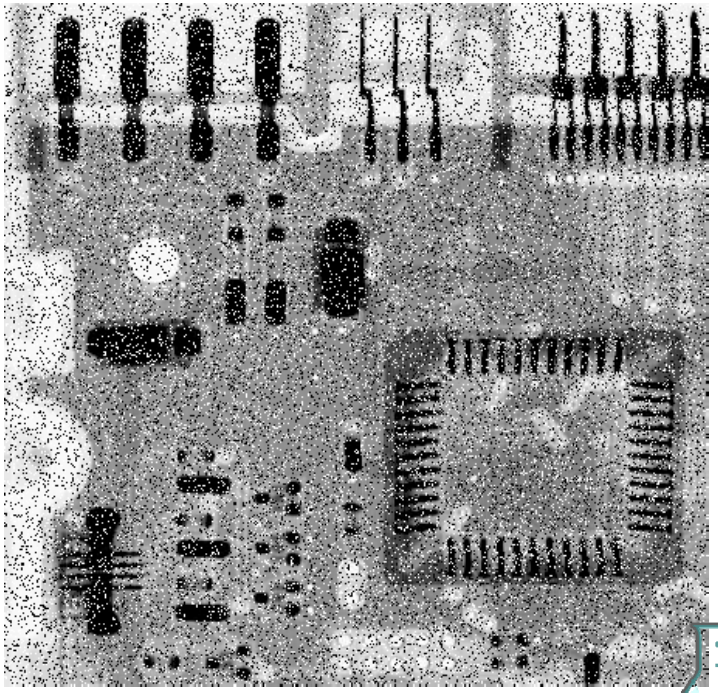
从眼底图像提取血管



方向均值滤波器

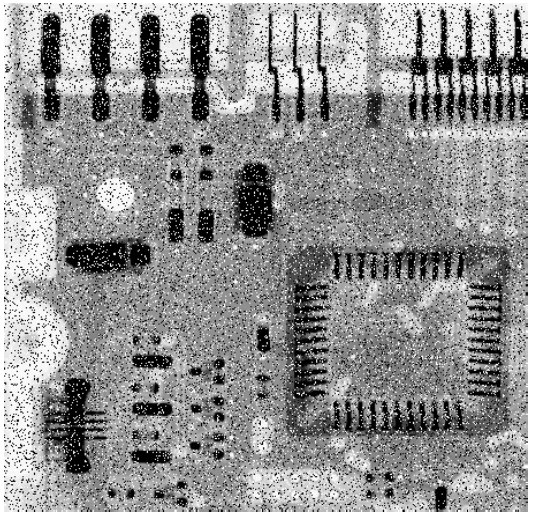
中值滤波

- 中值滤波是一种基于顺序统计的滤波器，对于椒盐噪声（salt-and-pepper noise）很有效
- 对当前像素的邻域像素按灰度值排序，取中间的灰度值作为输出值
- 它是线性滤波器吗？
- 顺序统计滤波器都是非线性的

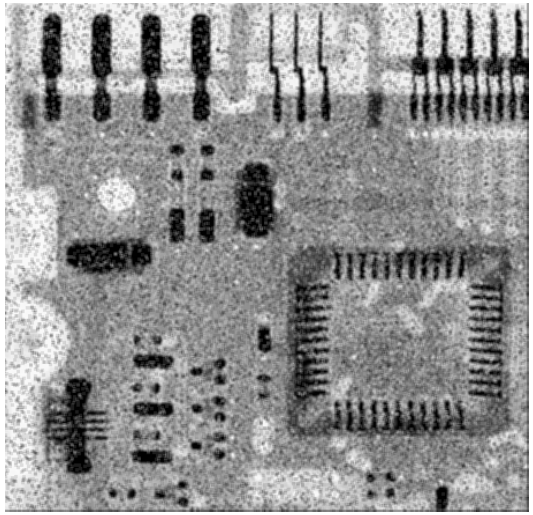


```
close all
I =
imread('..\data\Fig0335(a)(ckt_board_saltpep_p
rob_pt05).tif');
J1 = imfilter(I,ones(3,3)/9);
J2 = medfilt2(I,[3 3]);
figure(1),clf
ax(1)=subplot(1,3,1); imshow(I);
ax(2)=subplot(1,3,2); imshow(J1);
ax(3)=subplot(1,3,3); imshow(J2);
linkaxes(ax);
```

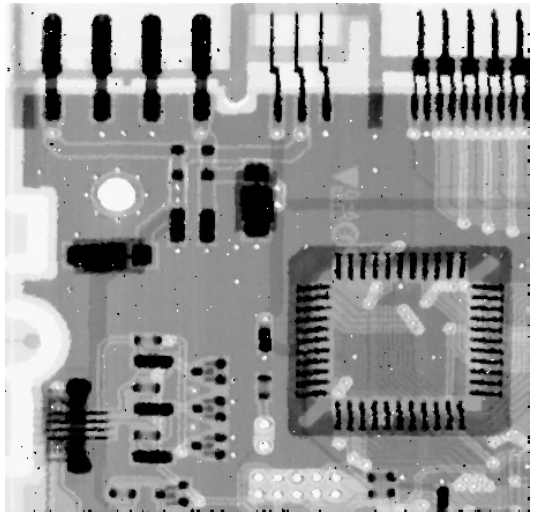
均值滤波 vs 中值滤波



椒盐噪声图像



3×3均值滤波结果

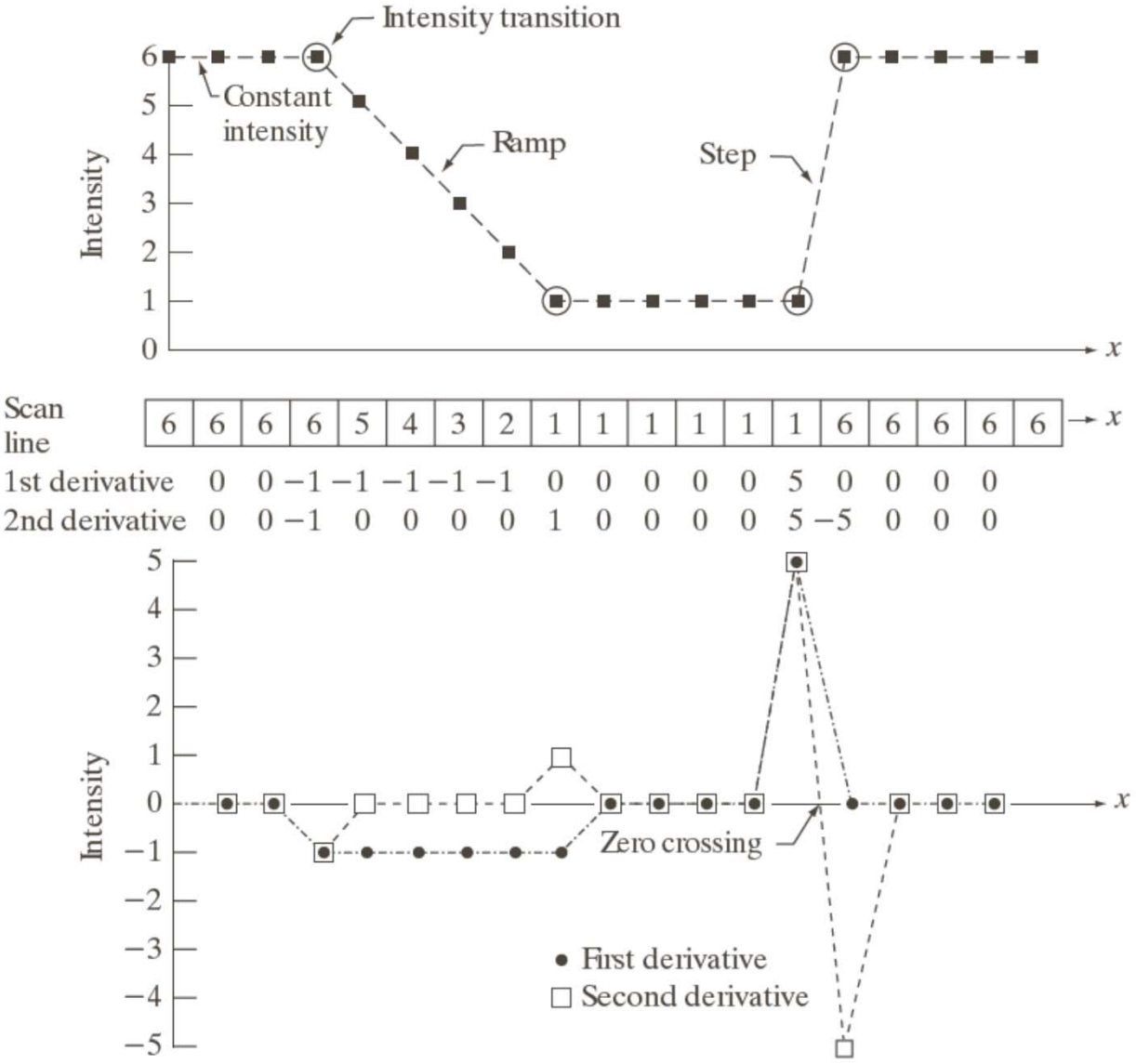


中值滤波结果

内　容

- 图像增强基础
- 灰度变换（Intensity transformation）
  - 基本的灰度变换函数
  - 直方图处理（Histogram processing）
- 空域滤波（Spatial filtering）
  - 平滑（Smoothing）
  - 锐化（Sharpening）

- 一维数字信号的一阶导$\frac{\partial f}{\partial x} = f(x+1) - f(x)$
- 二阶导$\frac{\partial^2 f}{\partial x^2} = f(x+1) + f(x-1) - 2f(x)$



二阶导锐化

拉普拉斯算子$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$，对连续函数各向同性（旋转无关）

$$\frac{\partial^2 f}{\partial x^2} = f(x+1, y) + f(x-1, y) - 2f(x, y)$$

$$\frac{\partial^2 f}{\partial y^2} = f(x, y+1) + f(x, y-1) - 2f(x, y)$$

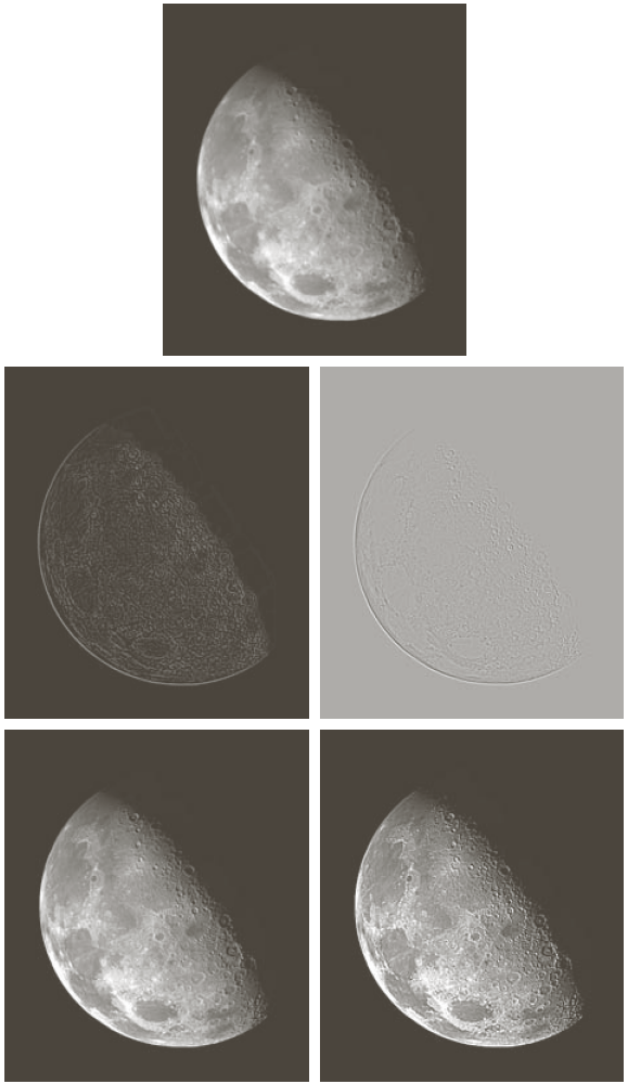
90度不变

0	1	0
1	−4	1
0	1	0

1	1	1
1	−8	1
1	1	1

45度不变

二阶导锐化例子



```
I = imread('..\data\Fig0338(a)(blurry_moon).tif');
I = im2double(I);

h1 = [0 1 0; 1 -4 1; 0 1 0];
h2 = [1 1 1; 1 -8 1; 1 1 1];

J1 = imfilter(I,h1);
J2 = imfilter(I,h2);

figure(1),imshow(I);
figure(2),imshow(J1,[]);
figure(3),imshow(J2,[]);
figure(4),imshow(I-J1,[]);
figure(5),imshow(I-J2,[]);
```

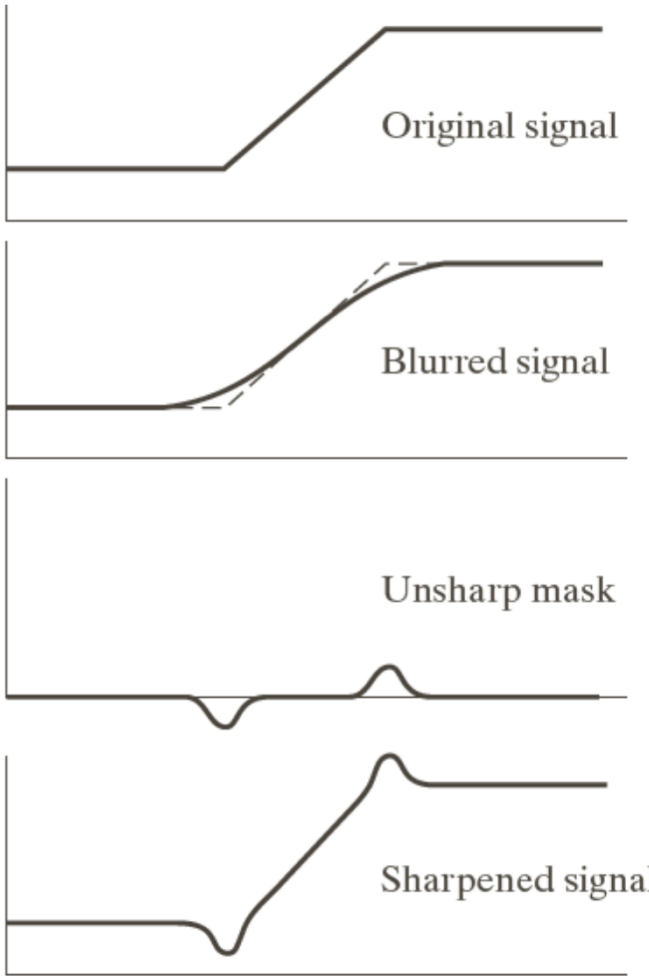
非锐化掩膜（Unsharp masking）

1. 平滑原图像$f(x,y)$
2. 从原图像中减去模糊图像，得到掩膜图像$g_{\text{mask}}(x,y)$
3. 将掩膜加到原图像中

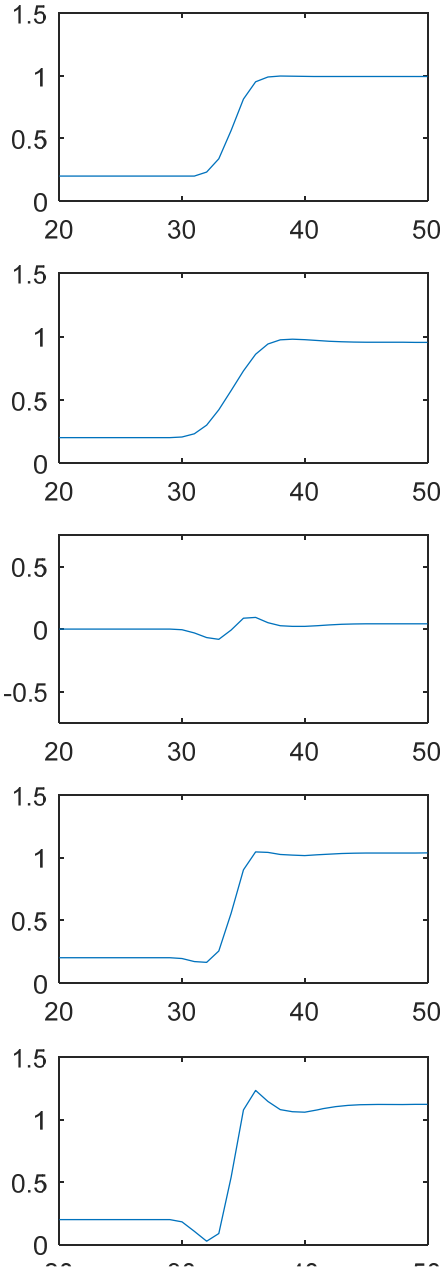
$$g(x,y) = f(x,y) + k \times g_{\text{mask}}(x,y)$$

$k > 1$时，称为高提升滤波（highboost filtering）

非锐化掩膜（Unsharp masking）



$k = 3$



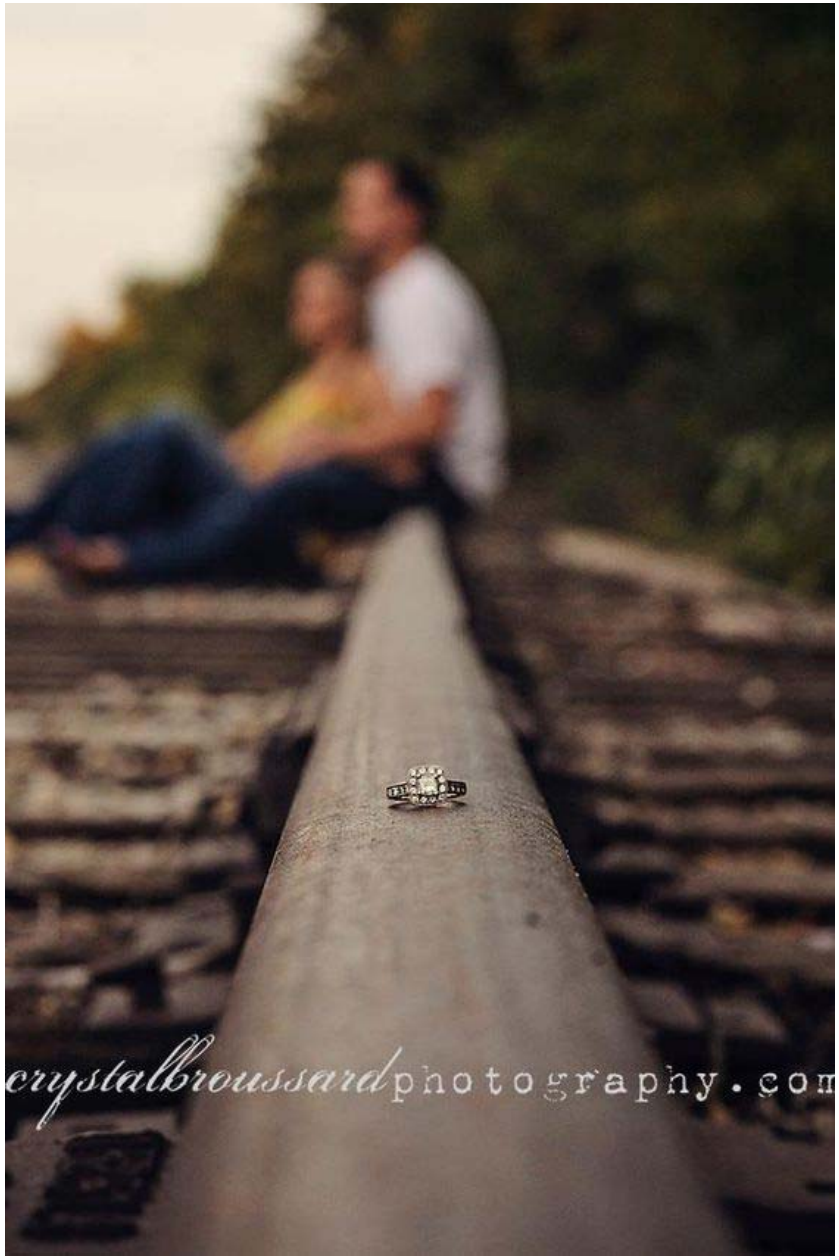
```
close all, clear all
I = imread('..\data\Fig0340(a)(dipxe_text).tif');
I = im2double(I);

h = fspecial('gaussian',5,3);
I_blur = imfilter(I,h);
I_mask = I-I_blur;
J1 = I+1*I_mask;
J2 = I+3*I_mask;

figure(1),
ax(1)=subplot(5,1,1); imshow(I);
ax(2)=subplot(5,1,2); imshow(I_blur);
ax(3)=subplot(5,1,3); imshow(I_mask,[]);
ax(4)=subplot(5,1,4); imshow(J1);
ax(5)=subplot(5,1,5); imshow(J2);
linkaxes(ax);

y=100;x=20:50;
idx = sub2ind(size(I),y*ones(size(x)),x);
figure(6),
subplot(5,1,1),plot(x,I(idx));ylim([0 1.5])
subplot(5,1,2),plot(x,I_blur(idx));ylim([0 1.5])
subplot(5,1,3),plot(x,I_mask(idx));ylim([-0.75 0.75])
subplot(5,1,4),plot(x,J1(idx));ylim([0 1.5])
subplot(5,1,5),plot(x,J2(idx));ylim([0 1.5])
```

小　结

- 图像增强是一种图像处理技术，目的是改善图像，以便于人的观看或自动的图像分析与识别
- 图像增强可在空域或者频域进行
- 2种空域增强方法：灰度变换和空域滤波
- 灰度变换：几种基本的灰度变换函数和直方图处理
- 空域滤波：平滑和锐化

手机厂商喜欢吹自己手机摄像头媲美单反

单反的出色功能之一是背景虚化

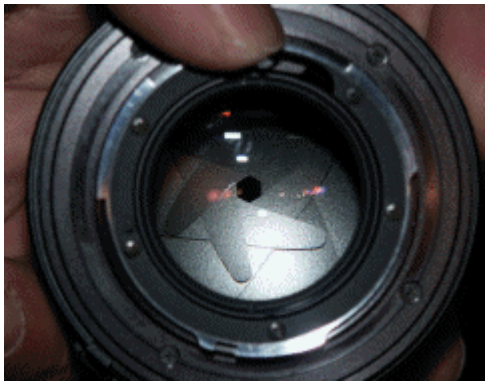


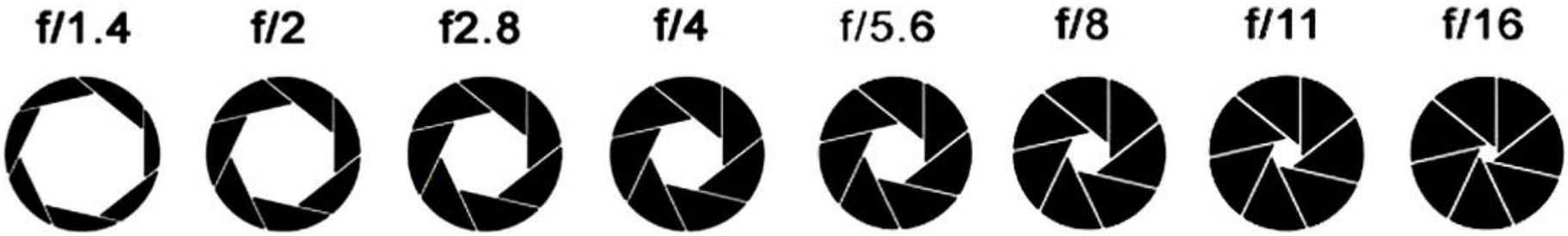
单反相机镜头的光圈

光圈（aperture）：装在镜头内用来调节进光量的装置

f后面的数字越小，光圈越大

数字的比例为半径的比例





http://www.canon.com.cn/specialsite/ds_abcbook/beginner02.html

光圈对景深的影响

光圈在调节进光量的同时，还能调节景深。

大光圈F2.8

小光圈F22

http://www.canon.com.cn/specialsite/ds_abcbook/intermediate03.html

光圈对景深的影响