Chapter4 Image Enhancement

- Preview
- 4.1 General introduction and Classification
- 4.2 Enhancement by Spatial Transforming(contrast enhancement)
- 4.3 Enhancement by Spatial Filtering (image smoothing)
- 4.4 Enhancement by Frequency Filtering (image sharpening)
- 4.5 Color Enhancement
- Summary

Preview

Image Negatives

Preview

Histogram Equalization

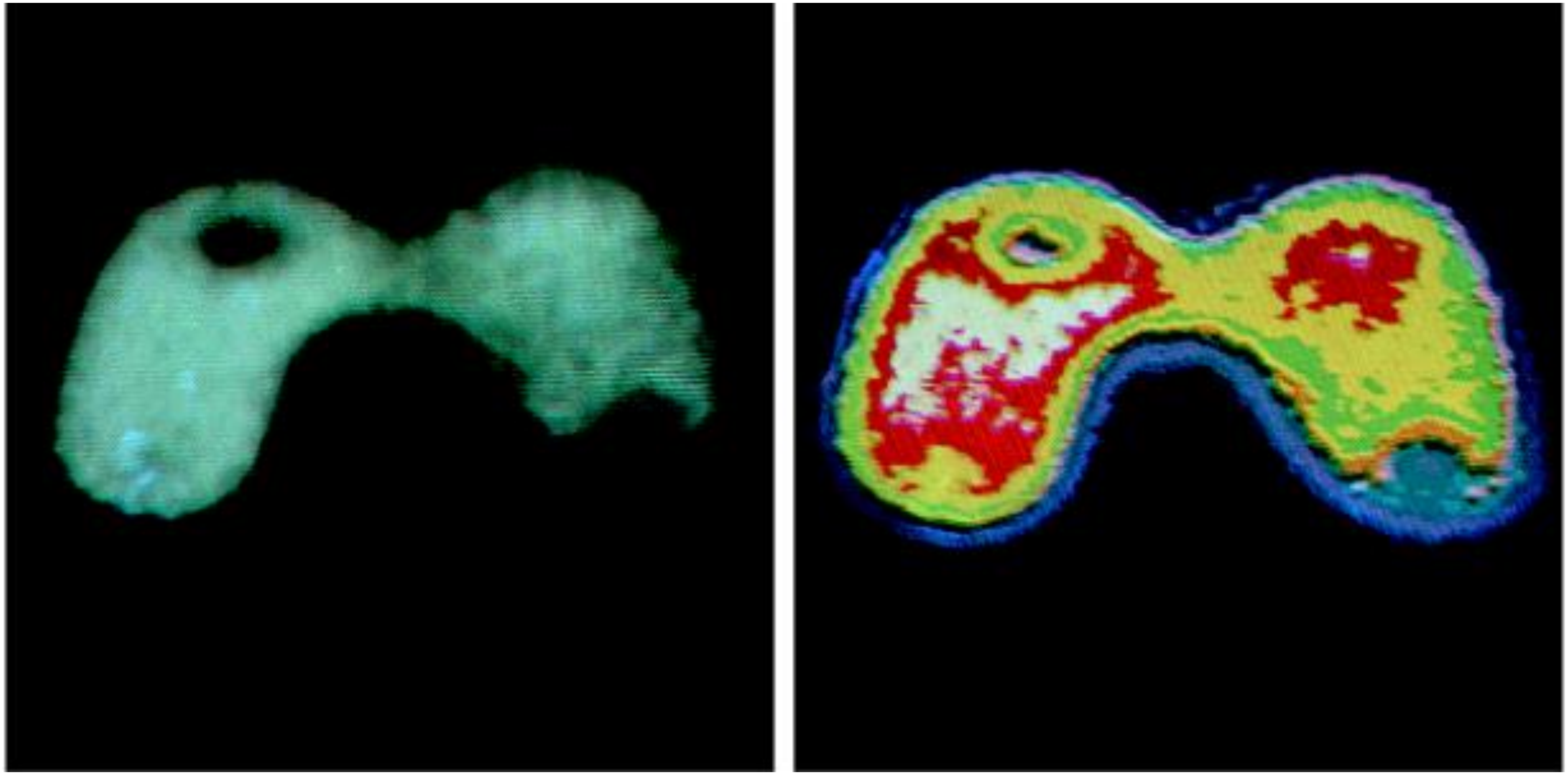
Preview

Image smoothing

Preview

Image sharpening

Preview



Pseudo color processing

Preview

Full color processing

4.1 General Introduction and Classification

4.1.1 Purposes

- improve the visual effects
- easy to edge extracting

4.1.2 Methods

- spatial domain: point operations, local operations
- frequency domain: DFT $\Rightarrow$ Filter $\Rightarrow$ IDFT

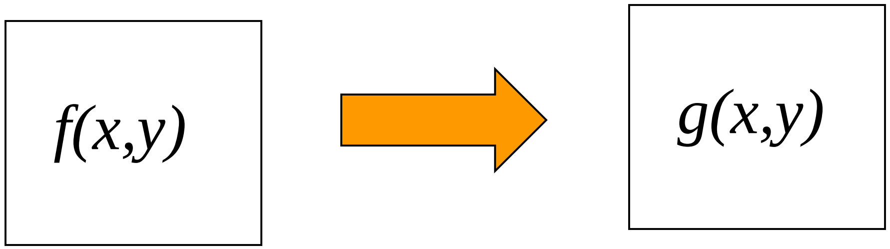
4.1 General Introduction and Classification

4.1.3 contents

- contrast enhancement: linear transform
  non-linear transform,
  histogram equalization
  histogram matching
  local enhancement
- image smoothing: averaging mask,
  order-statistics filter
  lowpass filter.
- image sharpening: derivatives,
  highpass filter
- color image enhancement: pseudo color processing,
  full color processing

4.2 Contrast Enhancement

4.2.1. Introduction: General expression



Local operator

$$g(x,y) = T[\,f(x \pm i, y \pm j)] \qquad i, j = 0, \pm 1, \pm 2 \cdots$$

where *T* is a operator on *f*, defined over some neighborhood of (*x,y*)

when the neighborhood is of size 1*1,(a single pixel), we define

$$r = f(x,y) \qquad\qquad s = g(x,y)$$

Point operator

and $$s = T(r)$$

4.2 Contrast Enhancement

4.2.1. Introduction: Histogram

Histogram gives an estimate of the probability of the occurrence of gray levels

$$p(s_k) = n_k / n \qquad k = 0, 1, \cdots, L-1$$

Where s_k is the the kth gray level and the n_k is the number of Pixels in the image having gray level s_k

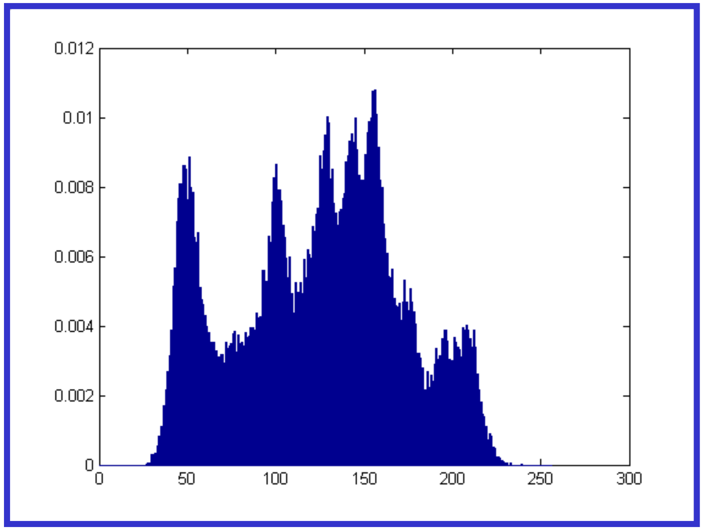
apparently

$$\sum_{k=0}^{L-1} p(s_k) = 1$$

4.2 Contrast Enhancement

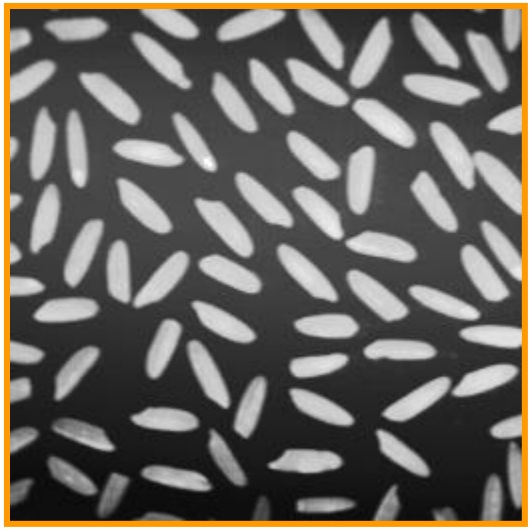
4.2.1. Introduction: Histogram

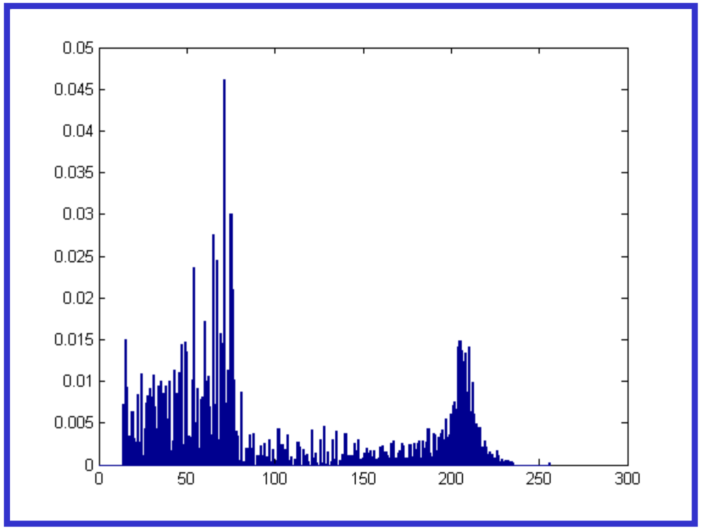
Horizontal axis: gray level values
Vertical axis: probability of gray level values

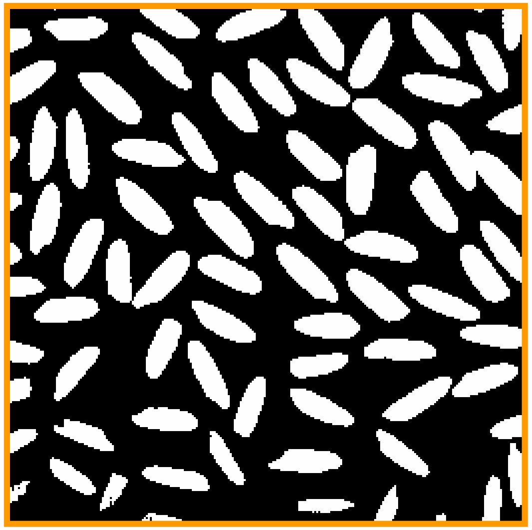


4.2 Contrast Enhancement

4.2.1. Introduction: Histogram



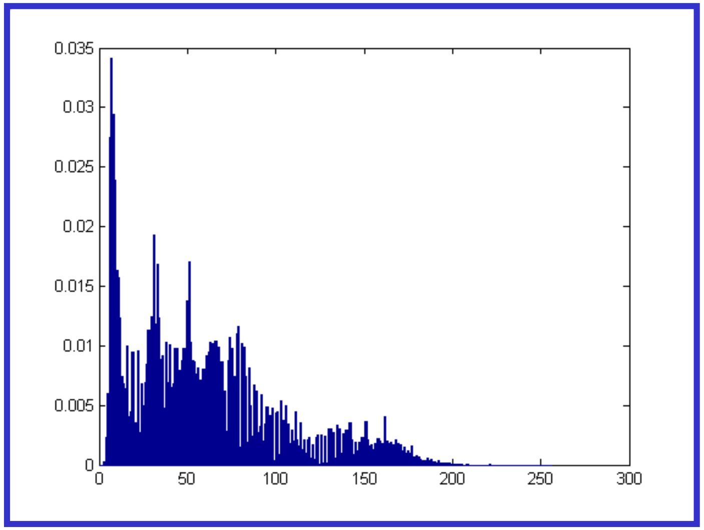




4.2 Contrast Enhancement

4.2.1. Introduction: Histogram

Histograms and images quality

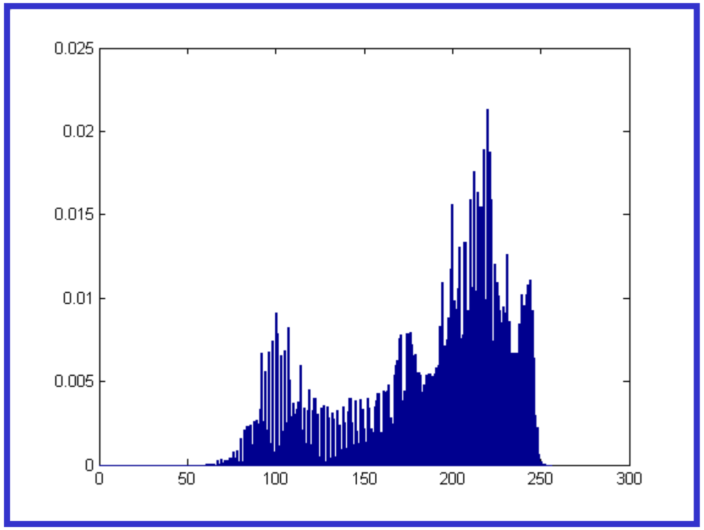


Dark image

4.2 Contrast Enhancement

4.2.1. Introduction: Histogram

Histograms and images quality

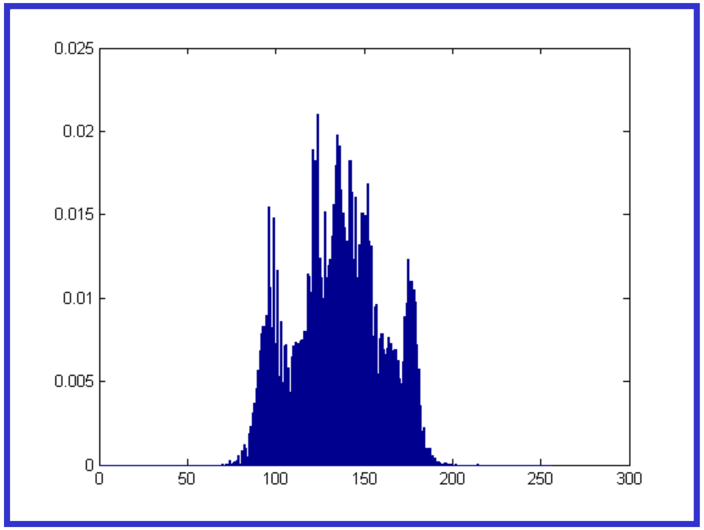


Bright image

4.2 Contrast Enhancement

4.2.1. Introduction: Histogram

Histograms and images quality

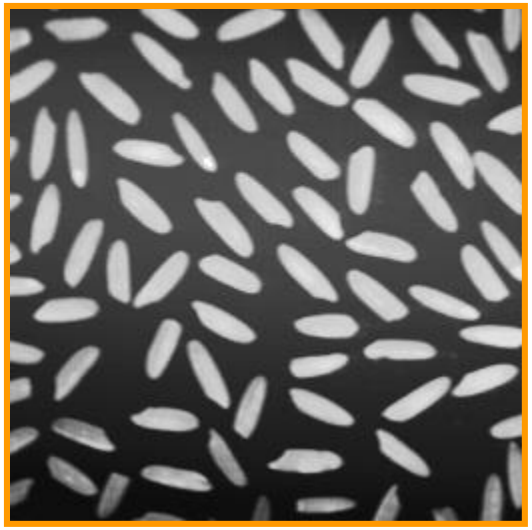


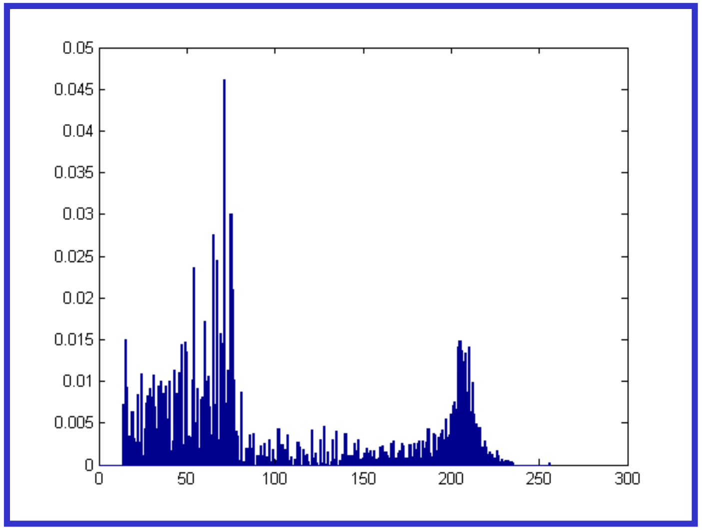
Low-contrast image

4.2 Contrast Enhancement

4.2.1. Introduction: Histogram

Histograms and images quality



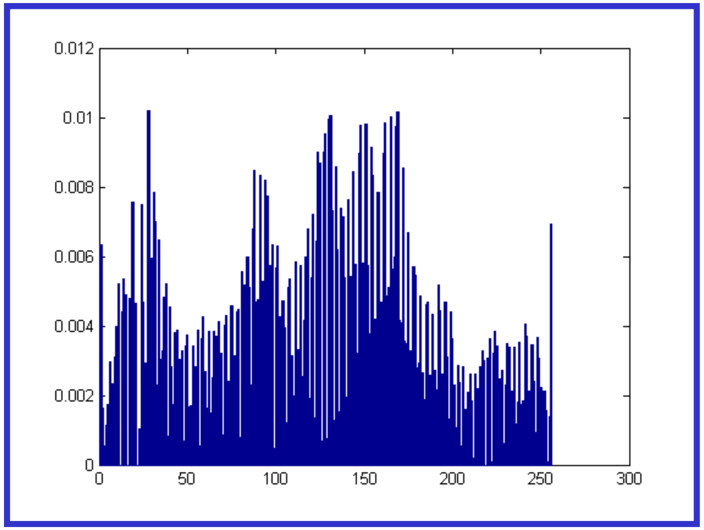


Double-peaks image

4.2 Contrast Enhancement

4.2.1. Introduction: Histogram

Histograms and images quality



Equalized image

4.2 Contrast Enhancement

4.2.1. Introduction: Classification

- Direct gray level transformations
- Histogram processing
- Operations among images

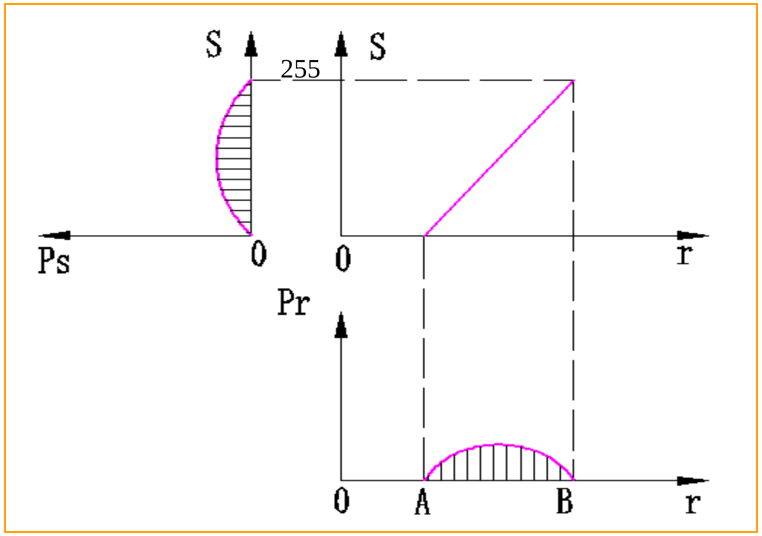
4.2 Contrast Enhancement

4.2.2 Direct gray level transformations: Linear transformations

Expression: $s = T(r) = ar + b \quad r \in [A, B] \quad s \in [C, D]$

Formulation: $$s = \frac{D-C}{B-A} r + \frac{BC-AD}{B-A}$$

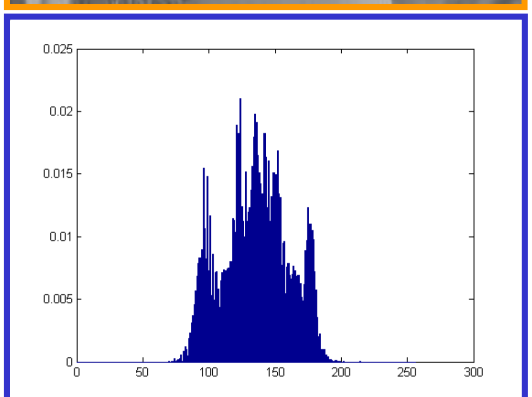
Example:
C=0
D=255

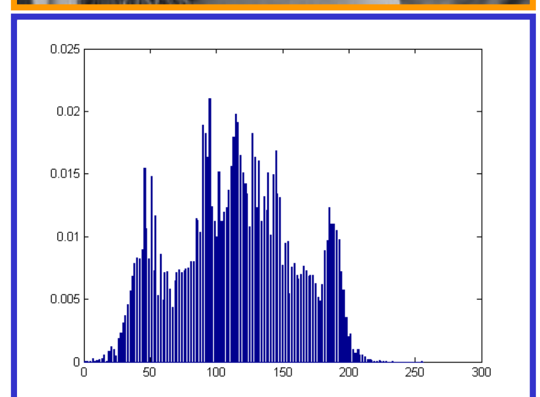


4.2 Contrast Enhancement

4.2.2 Direct gray level transformations : Linear transformations

example: $A= 69, B=213, C=0, D=255 \longrightarrow s = 1.7r - 122.2$



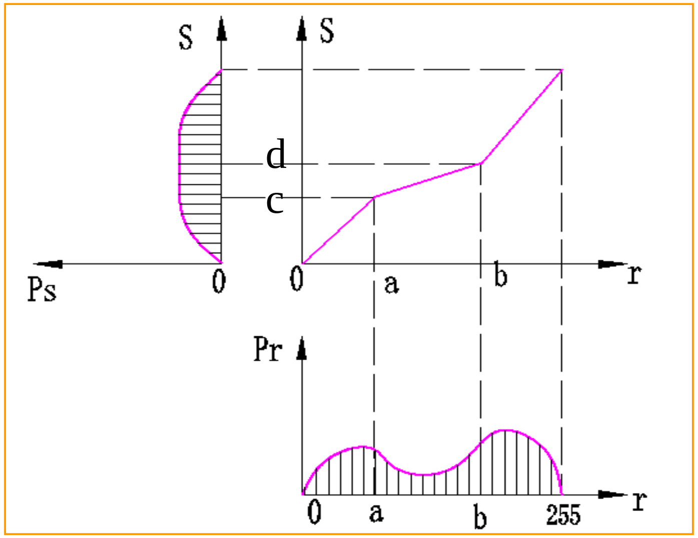


4.2 Contrast Enhancement

4.2.2 Direct gray level transformations : Piecewise-linear transformations

Formulation:

$$s=\begin{cases}\dfrac{c}{a}r & r\in[0,a)\\[2ex] \dfrac{d-c}{b-a}r+c & r\in[a,b)\\[2ex] \dfrac{255-d}{255-b}(r-b)+d & r\in[b,255)\end{cases}$$



4.2 Contrast Enhancement

4.2.2 Direct gray level transformations : Piecewise-linear transformations

Original image

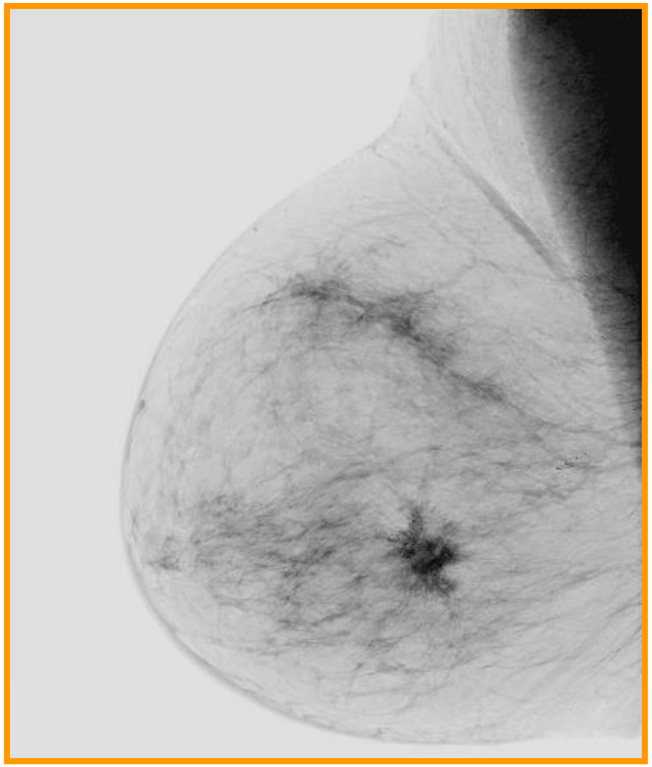
$(a,c)=(10,50),(b,d)=(210,150)$

$(a,c)=(50,10),(b,d)=(150,210)$

4.2 Contrast Enhancement

4.2.2 Direct gray level transformations : image negatives

Formulation: $$s = L - 1 - r$$

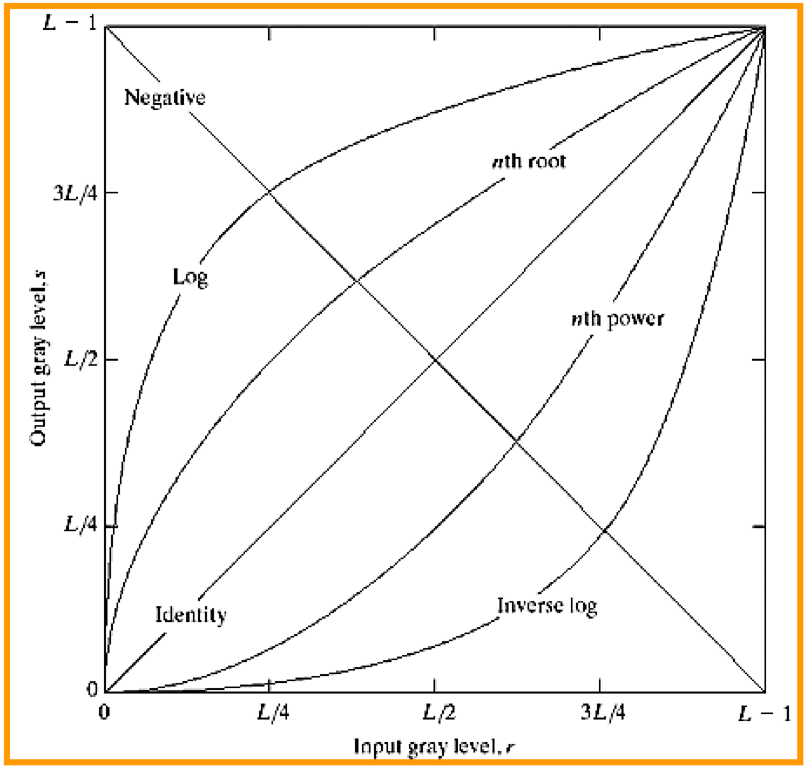


4.2 Contrast Enhancement

4.2.2 Direct gray level transformations : Log transformations

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

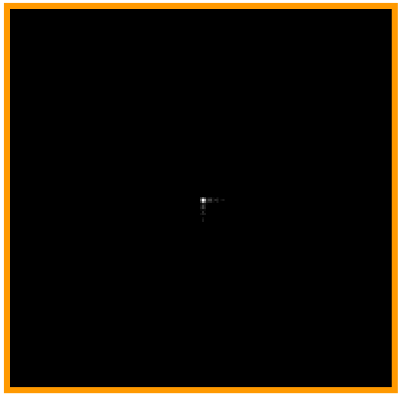
c is a constant and it is assumed that $r \geq 0$



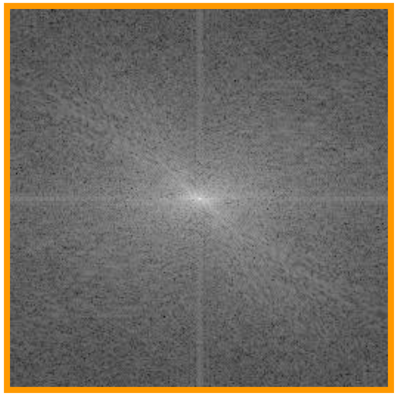
4.2 Contrast Enhancement

4.2.2 Direct gray level transformations : Log transformations

Example: display of DFT spectrum



Direct display



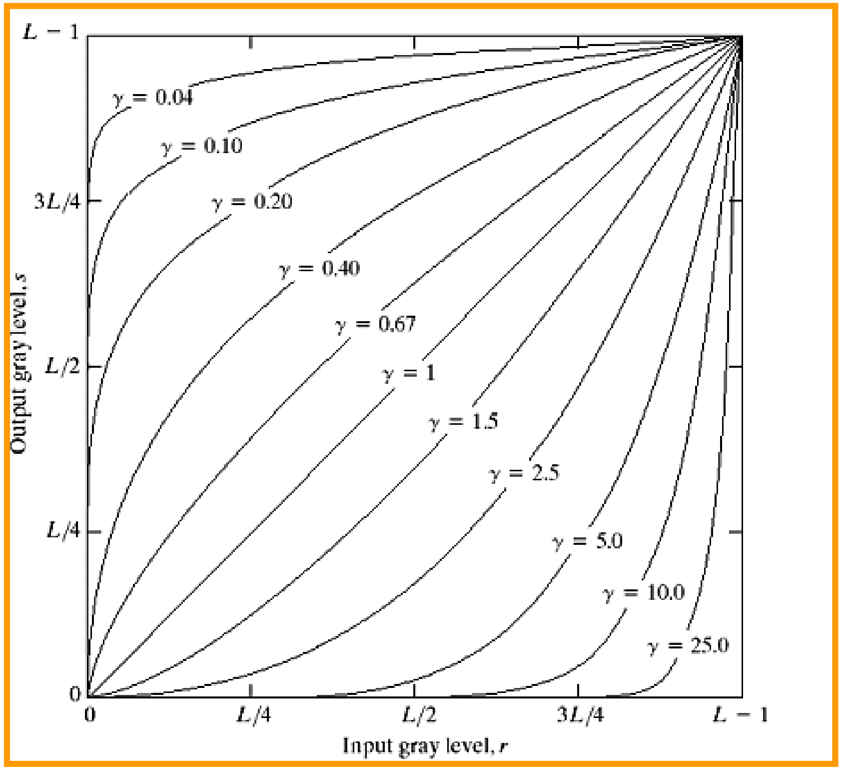
After log operation

4.2 Contrast Enhancement

4.2.2 Direct gray level transformations : Power transformations

Formulation: $$s = cr^{\gamma}$$

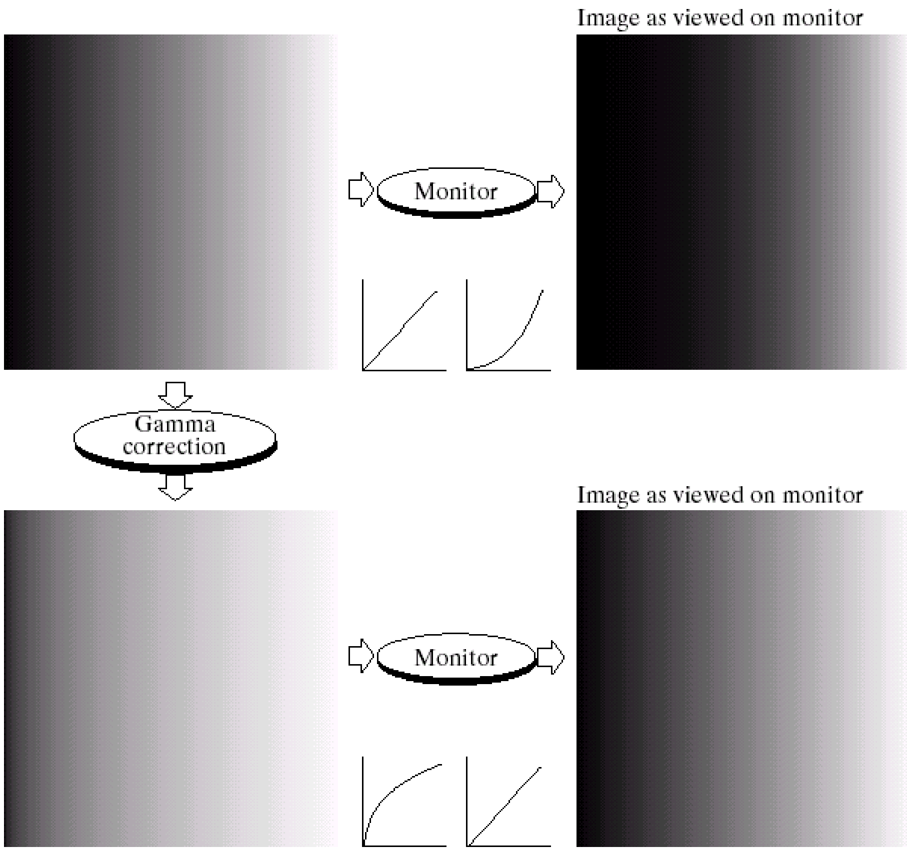
Where c and γ are positive constants.



4.2 Contrast Enhancement

4.2.2 Direct gray level transformations : Power transformations

Gamma correction



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.

4.2 Contrast Enhancement

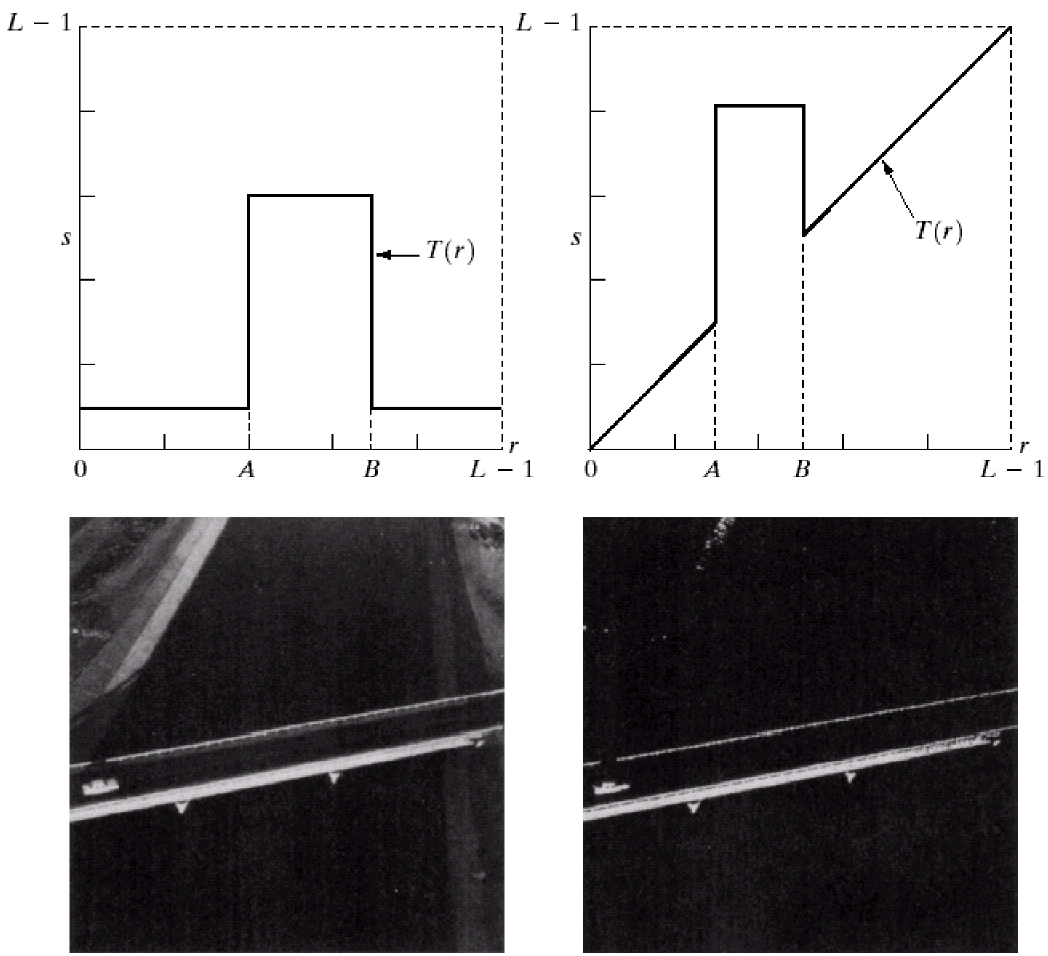
4.2.2 Direct gray level transformations : Power transformations

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.)

4.2 Contrast Enhancement

4.2.2 Direct gray level transformations : Gray-lever slicing



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).

4.2 Contrast Enhancement

4.2.3 Histogram processing : Histogram equalization

If a transform has the form:

$$s = T(r) \qquad 0 \le r \le 1$$

We hope the Probability Density Function (PDF) of s is

$$p_s(s) = 1$$

Inverse transform exist
and, if $r_1 < r_2$ then $s_1 < s_2$

$T(r)$ satisfies the following conditions:

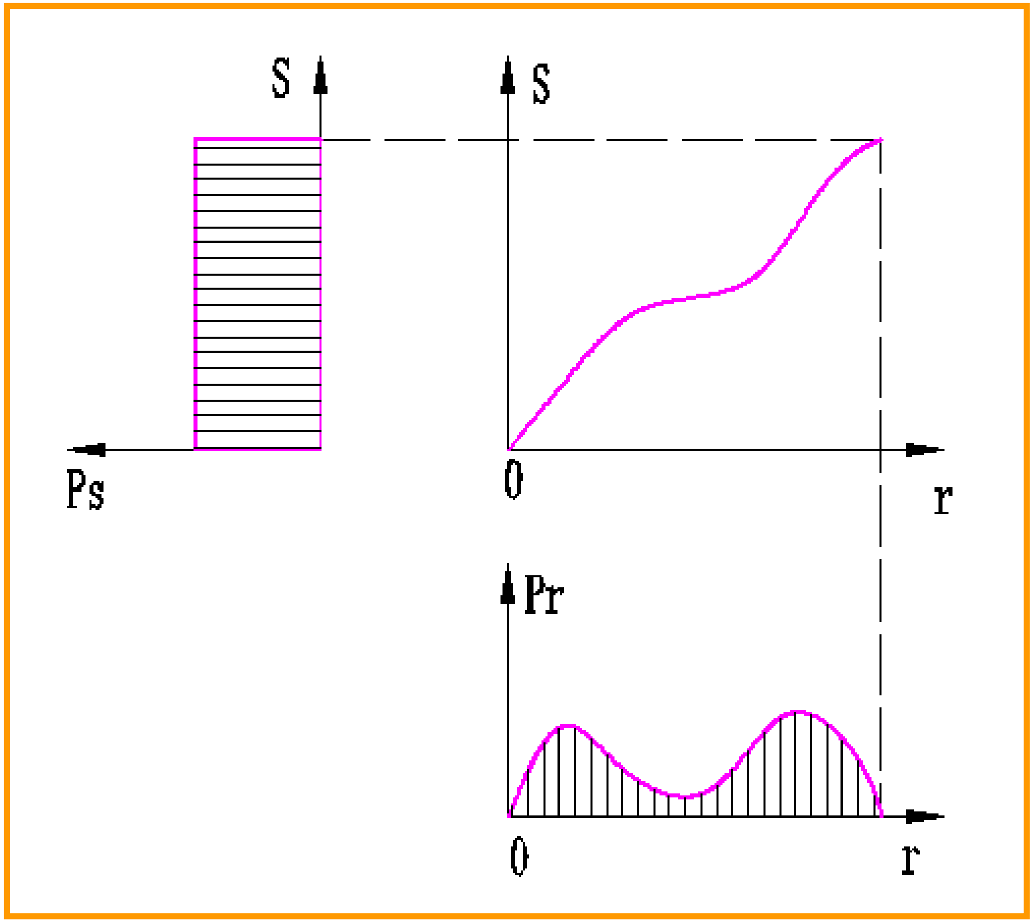
(a) is single-valued and monotonically in the interval $0 \le r \le 1$

(b) $0 \le s \le 1$ for $0 \le r \le 1$

s has the same range as the r

4.2 Contrast Enhancement

4.2.3 Histogram processing : Histogram equalization



4.2 Contrast Enhancement

4.2.3 Histogram processing : Histogram equalization

Continuous condition

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

$$\left.\begin{array}{l} \dfrac{ds}{dr} = \dfrac{p_r(r)}{p_s(s)} \\ p_s(s) = 1 \end{array}\right\} \Rightarrow s = \int_0^r p_r(x)dx$$

namely $T(r) = \int_0^r p_r(x)dx$

4.2 Contrast Enhancement

4.2.3 Histogram processing : Histogram equalization

Discrete condition

$$s_k = T(r_k) = \sum_{j=0}^{k} p_r(r_j) = \sum_{j=0}^{k} \frac{n_j}{n} \qquad k = 0, 1, 2, \ldots, L-1$$

In practice

$$s_k = T(r_k) = (L-1)\sum_{j=0}^{k} p_r(r_j)$$

4.2 Contrast Enhancement

4.2.3 Histogram processing : Histogram equalization

Example:page 78

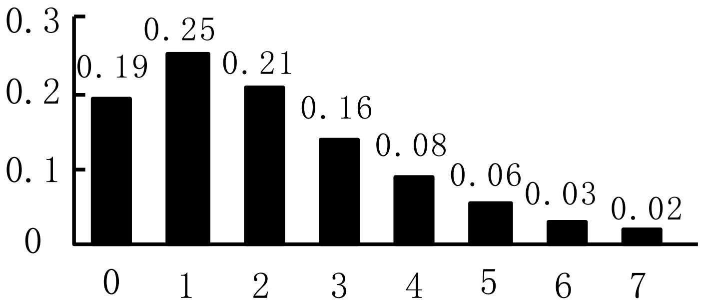
Suppose that a 64*64, 3bits image has the gray-level distribution as show in the first row, the calculate steps are:

序号	运　算	步骤和结果							
1	列出原始图灰度级 $s_k, k=0,1,...,7$	0	1	2	3	4	5	6	7
2	统计原始直方图各灰度级象素 n_k	790	1023	850	656	329	245	122	81
3	计算原始直方图	0.19	0.25	0.21	0.16	0.08	0.06	0.03	0.02
4	计算累计直方图	0.19	0.44	0.65	0.81	0.89	0.95	0.98	1.00
5	取整 $t_k = \text{int}[(N-1)t_k + 0.5]$	1	3	5	6	6	7	7	7
6	确定映射对应关系 $(s_k \rightarrow t_k)$	$0 \rightarrow 1$	$1 \rightarrow 3$	$2 \rightarrow 5$	$3,4 \rightarrow 6$		$5,6,7 \rightarrow 7$		
7	统计新直方图各灰度级象素 n_k		790		1023		850	985	448
8	用计算新直方图		0.19		0.25		0.21	0.24	0.11

4.2 Contrast Enhancement

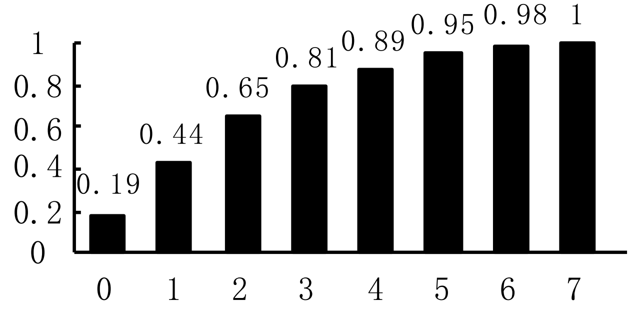
4.2.3 Histogram processing : Histogram equalization

Histograms



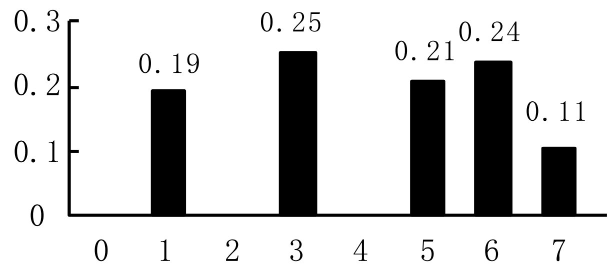
(a)

original histogram



(b)

transformation function



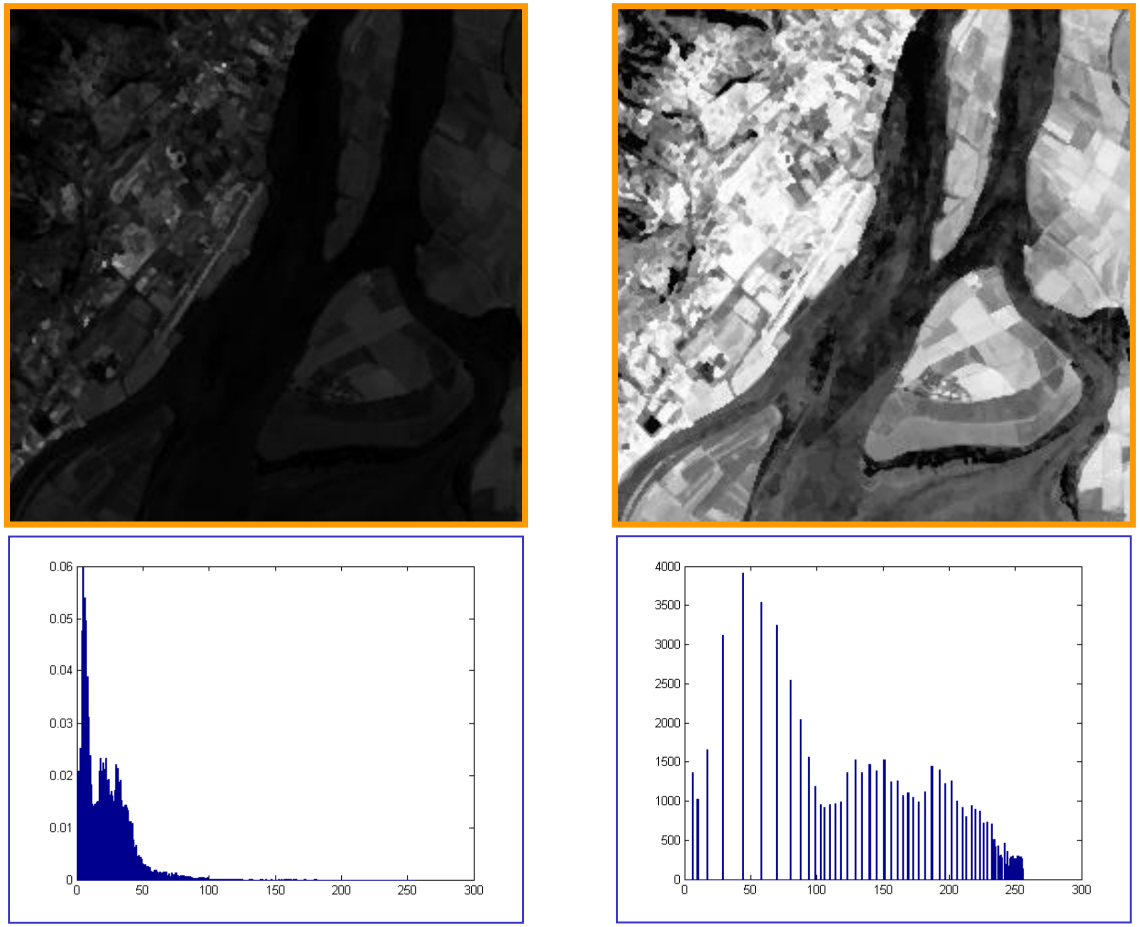
(c)

equalized histogram

4.2 Contrast Enhancement

4.2.3 Histogram processing : Histogram equalization

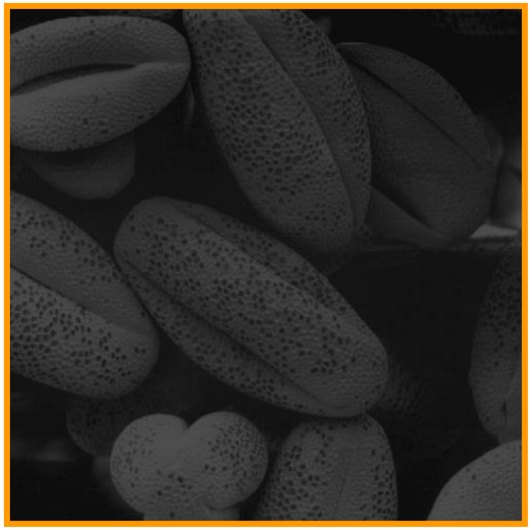
Experimental results

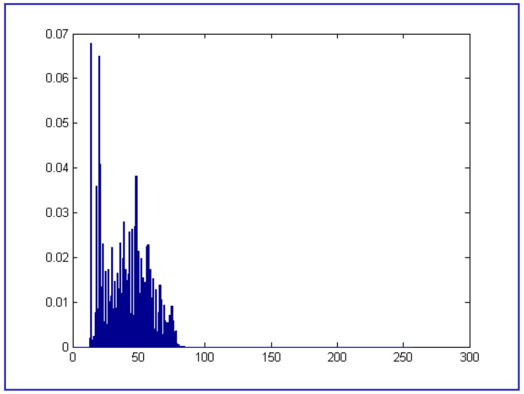


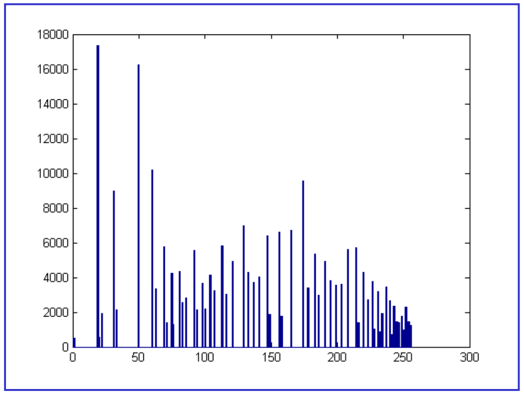
4.2 Contrast Enhancement

4.2.3 Histogram processing : Histogram equalization

Experimental results



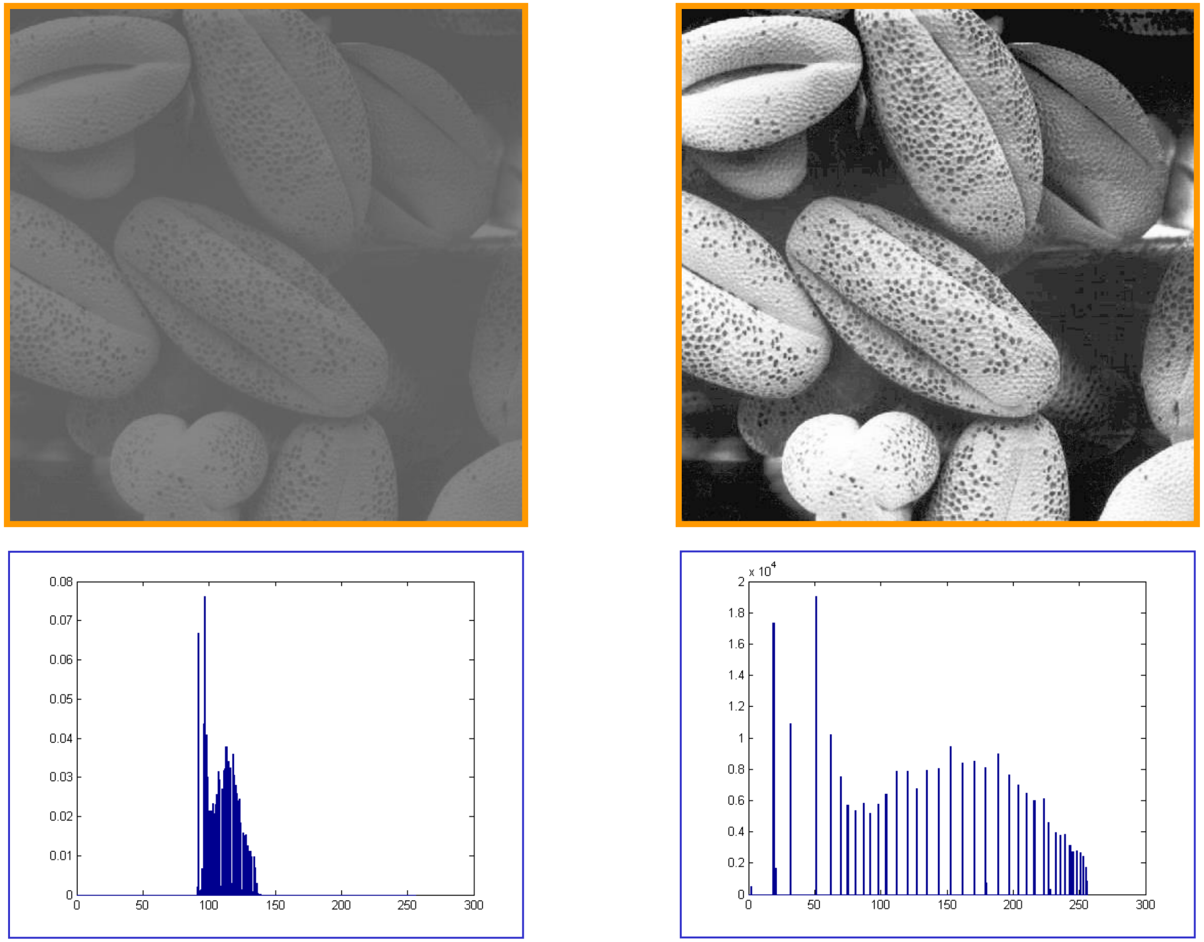




4.2 Contrast Enhancement

4.2.3 Histogram processing : Histogram equalization

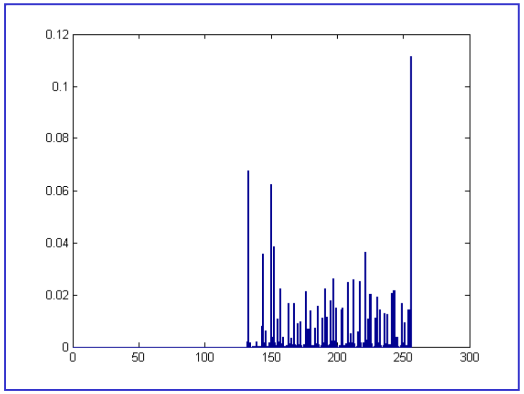
Experimental results

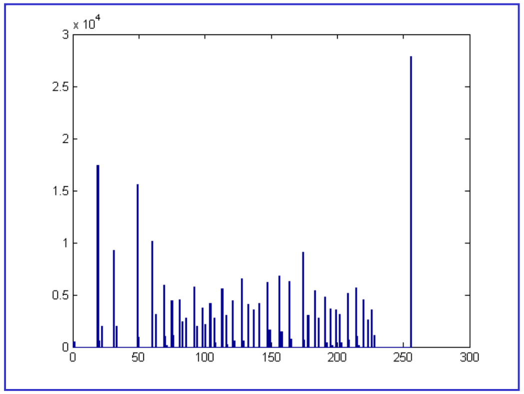


4.2 Contrast Enhancement

4.2.3 Histogram processing : Histogram equalization

Experimental results

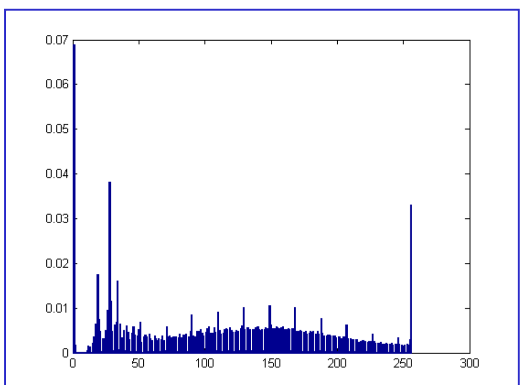


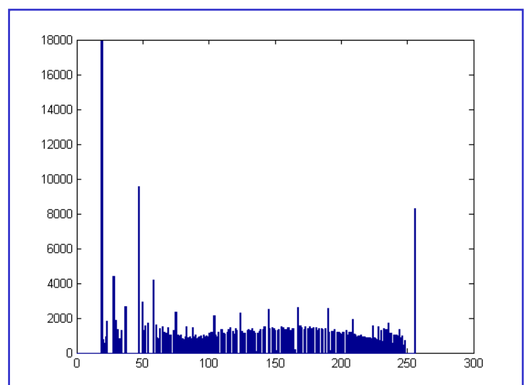


4.2 Contrast Enhancement

4.2.3 Histogram processing : Histogram equalization

Experimental results





4.2 Contrast Enhancement

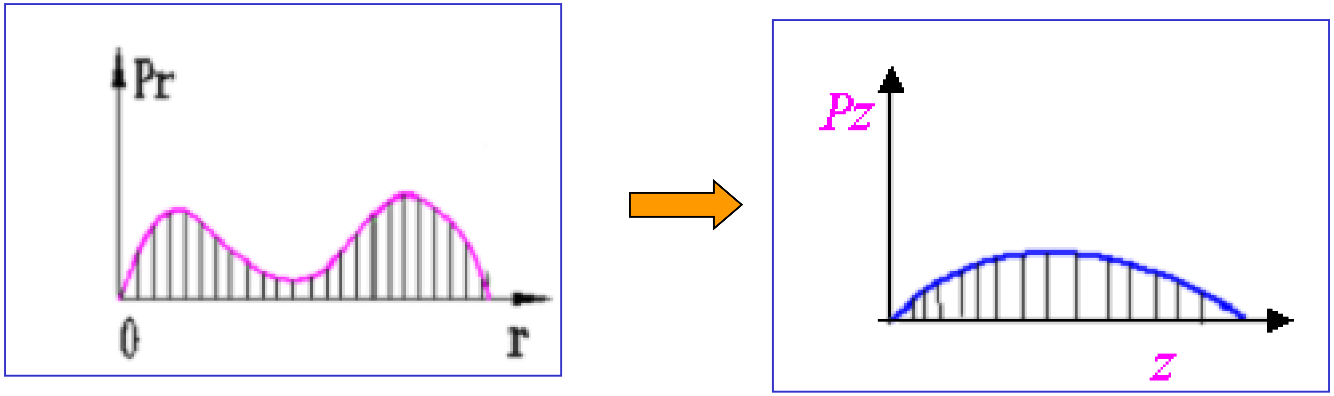
4.2.3 Histogram processing : Histogram equalization

question: page 99, 4.2

Why the discrete histogram equalization technique does not yield a flat histogram?

4.2 Contrast Enhancement

4.2.3 Histogram processing : Histogram matching (specification)



4.2 Contrast Enhancement

4.2.3 Histogram processing : Histogram matching (specification)

Histogram equalization: $r \longrightarrow s$

$$s_k = T(r_k) = \sum_{j=0}^{k} p_r(r_j) \qquad k = 0,1,2 \cdots L-1$$

Histogram equalization: $z \longrightarrow v$

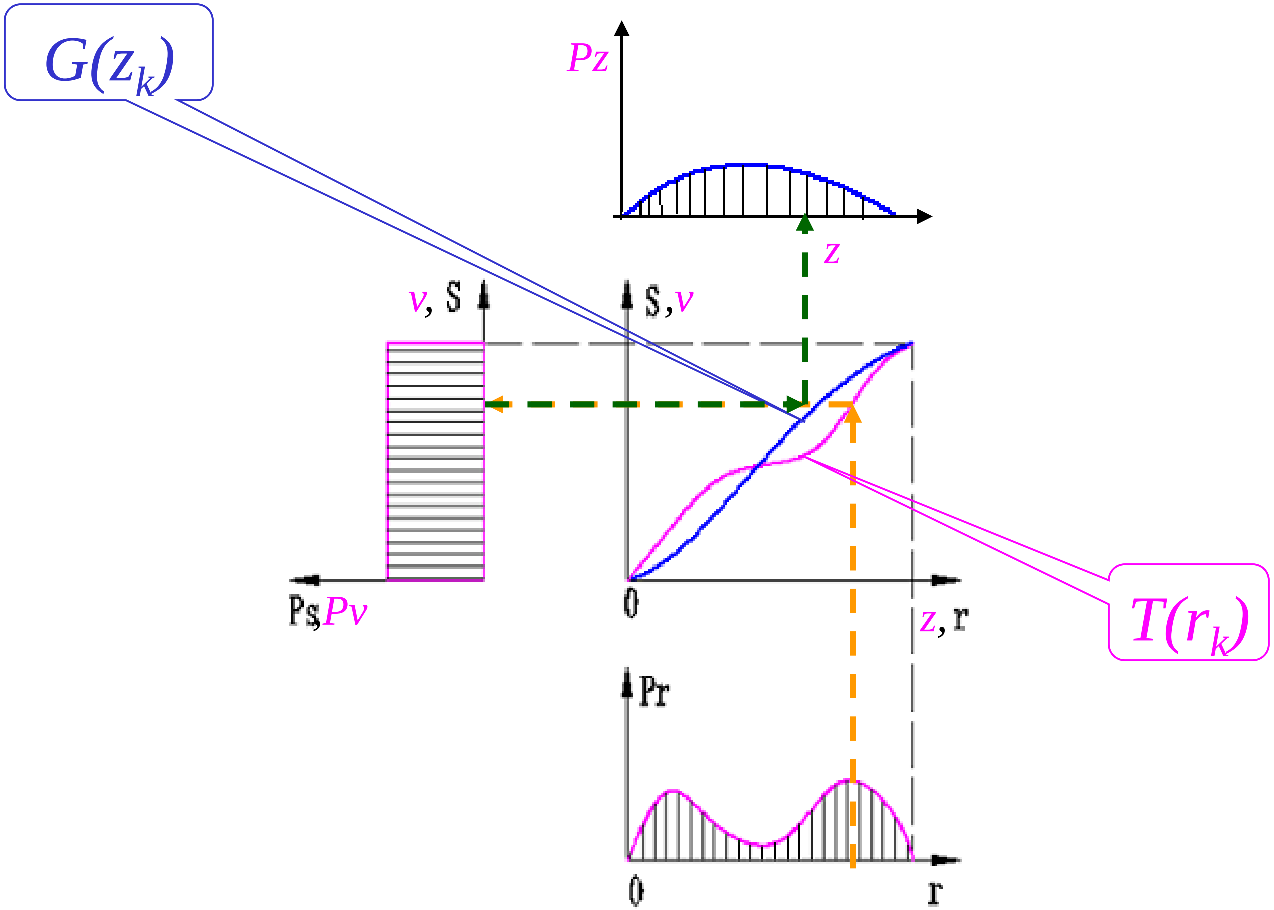
$$v_k = G(z_k) = \sum_{i=0}^{k} p_z(z_i) \qquad k = 0,1,2 \cdots L-1$$

let $\quad v_k = s_k \quad$ then $\quad r \longrightarrow z$

$$z_k = G^{-1}(v_k) = G^{-1}(s_k) = G^{-1}(T(r_k))$$

4.2 Contrast Enhancement

4.2.3 Histogram processing : Histogram matching (specification)



4.2 Contrast Enhancement

4.2.3 Histogram processing : Histogram matching (specification)

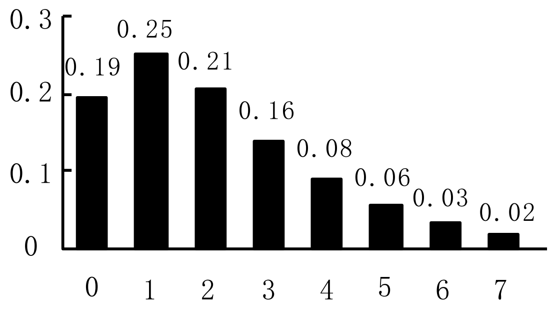
Example:page 80

Suppose that a 64*64, 3bits image has the gray-level distribution as show in the first row, and the 5th row is the specified gray-level distribution. The calculate steps are:

序号	运 算	步骤和结果							
1	列出原始图灰度级 $s_k, k = 0,1,\ldots,7$	0	1	2	3	4	5	6	7
2	统计原始直方图各灰度级象素 n_k	790	1023	850	656	329	245	122	81
3	计算原始直方图	0.19	0.25	0.21	0.16	0.08	0.06	0.03	0.02
4	计算原始累计直方图	0.19	0.44	0.65	0.81	0.89	0.95	0.98	1.00
5	规定直方图 $p_u(u_k) = n_k / n, n = 4096$	0	0	0	0.2	0	0.6	0	0.2
6	计算规定累计直方图	0	0	0	0.2	0.2	0.8	0.8	1.0
7s	SML映射	3	3	5	5	5	7	7	7
8s	确定映射对应关系	$0,1 \rightarrow 3$		$2,3,4 \rightarrow 5$			$5,6,7 \rightarrow 7$		
9s	变换后直方图	0	0	0	0.44	0	0.45	0	0.11

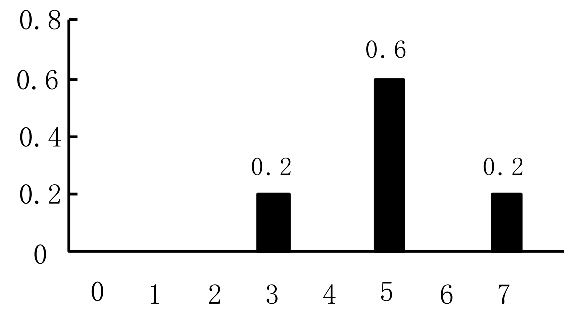
4.2 Contrast Enhancement

4.2.3 Histogram processing : Histogram matching (specification)



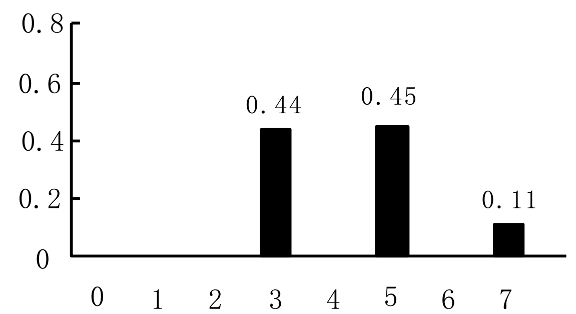
(a)

original histogram



(b)

specified histogram

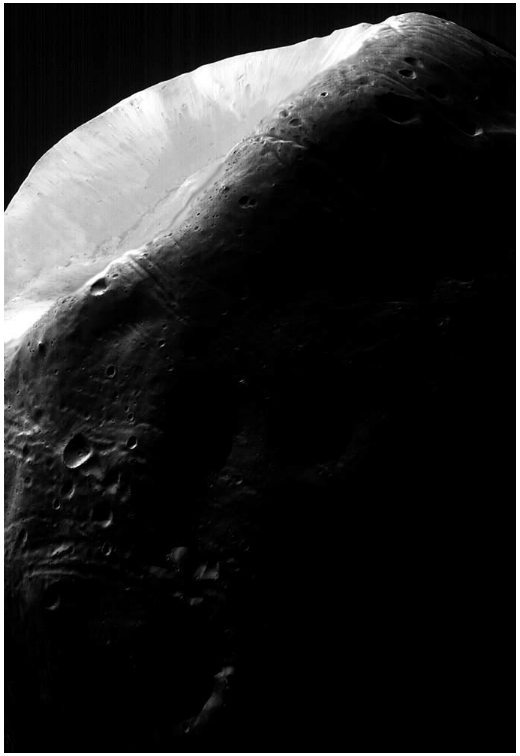


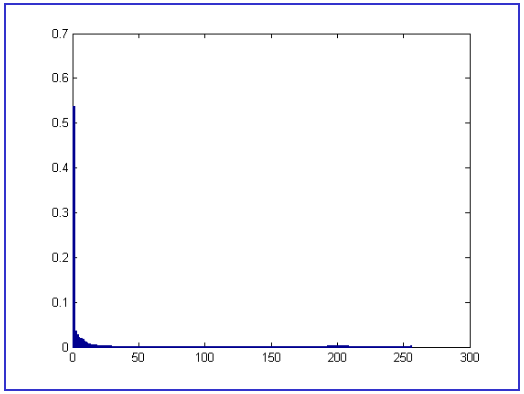
(c)

resulting histogram

4.2 Contrast Enhancement

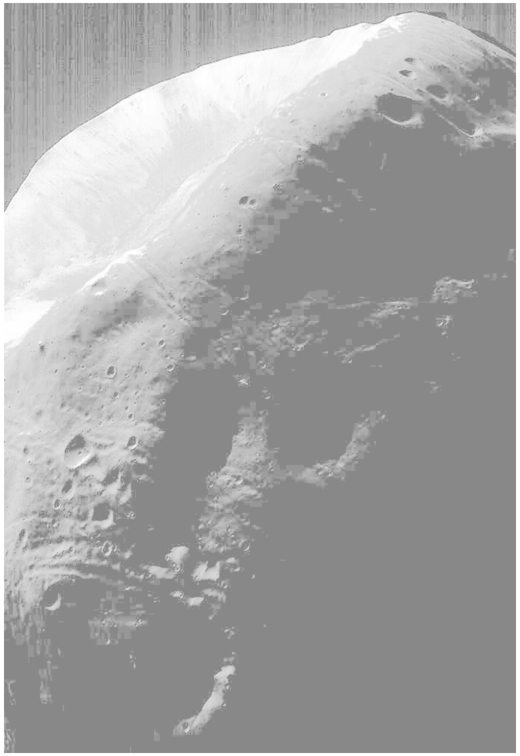
4.2.3 Histogram processing : Histogram matching (specification)

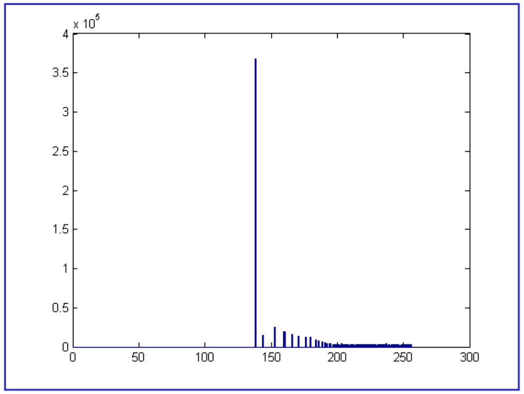




4.2 Contrast Enhancement

4.2.3 Histogram processing : Histogram matching (specification)





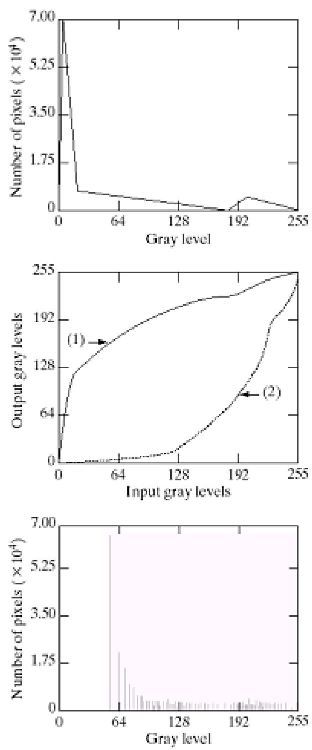
4.2 Contrast Enhancement

4.2.3 Histogram processing : Histogram matching (specification)

Example

a c
b
d

FIGURE 3.22
(a) Specified histogram.
(b) Curve (1) is from Eq. (3.3-14), using the histogram in (a); curve (2) was obtained using the iterative procedure in Eq. (3.3-17).
(c) Enhanced image using mappings from curve (2).
(d) Histogram of (c).



original

equalized

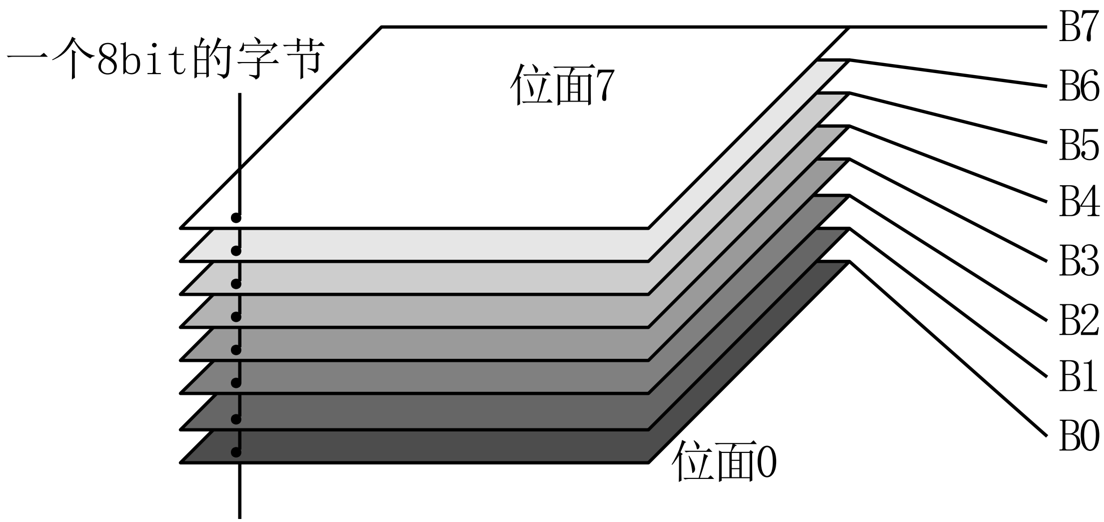
matched

4.2 Contrast Enhancement

4.2.2 Direct gray level transformations : Bit-plane slicing

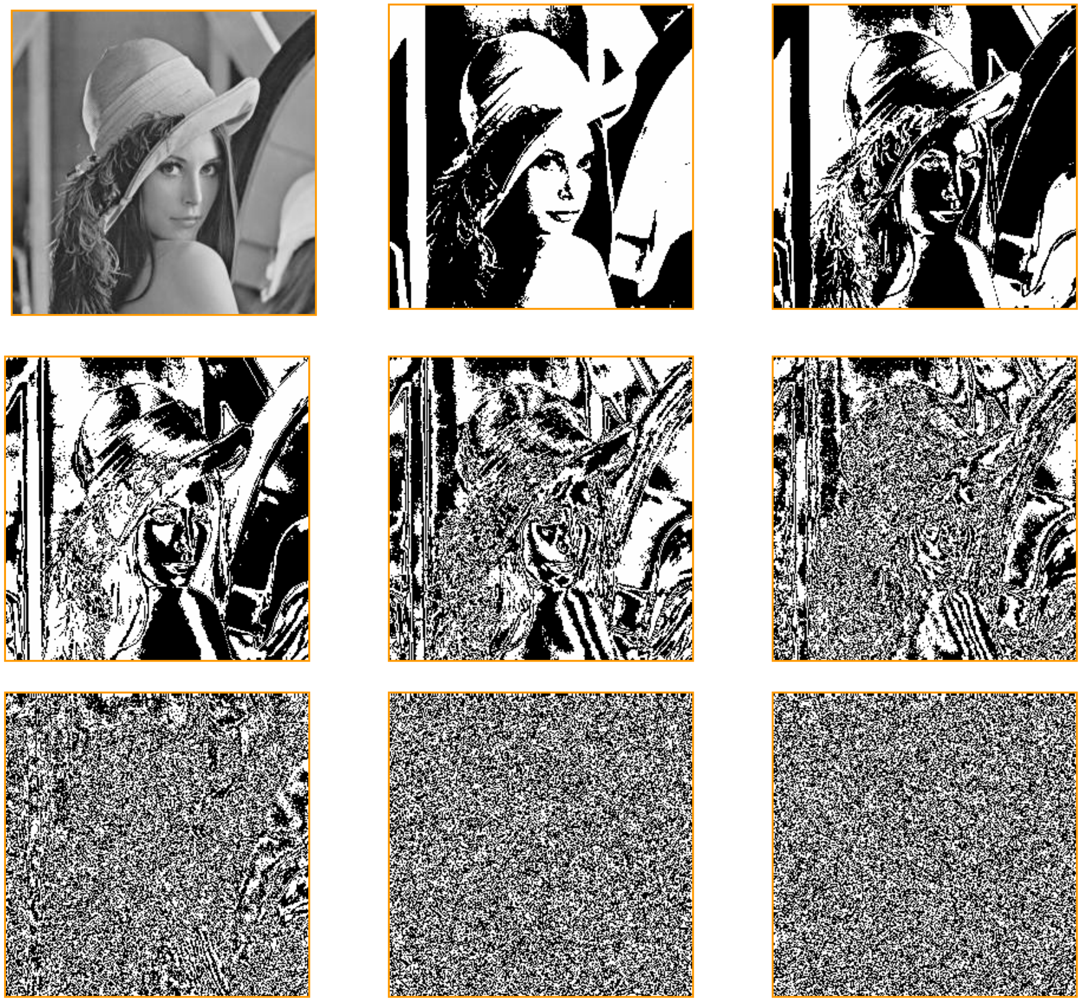
eg:

b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0	
1	1	1	1	1	1	1	1	255
1	1	1	1	1	1	1	0	254
1	0	0	0	0	0	0	1	129



4.2 Contrast Enhancement

4.2.2 Direct gray level transformations : Bit-plane slicing



	b_7	b_6
b_5	b_4	b_3
b_2	b_1	b_0

4.3 Image Smoothing

- Introduction
- Smoothing linear filters
- Order- statistic filters
- Low-pass filter in frequency domain

4.3 Image Smoothing

4.3.1 Introduction

Purposes: removing noise

Request: keeping edges and details

Noise Edges Details

Gray levels change greatly in neighborhood

Having the same texture in a region

Having the similar grads directions in a segment

Random in gray levels and spatial location

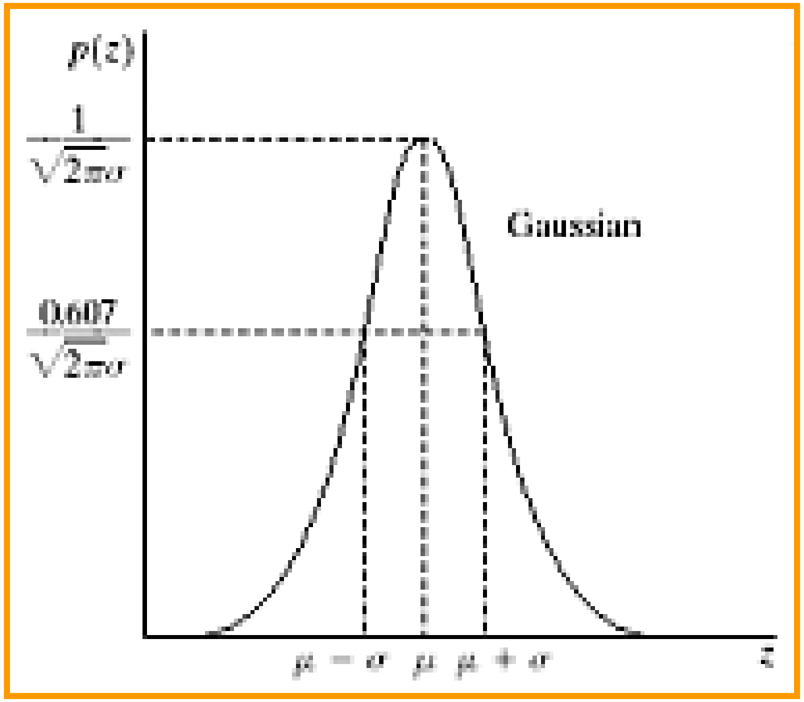
Salt-pepper noise

Gaussian noise

4.3 Image Smoothing

4.3.1 Introduction: Noise models

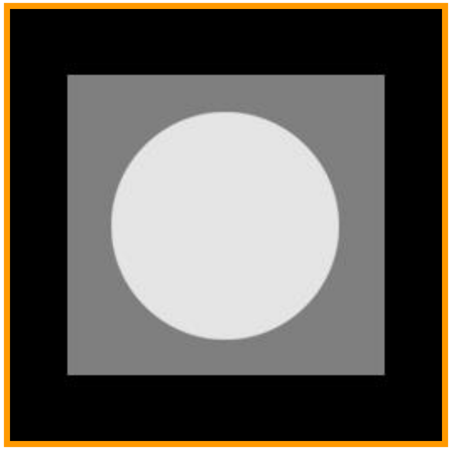
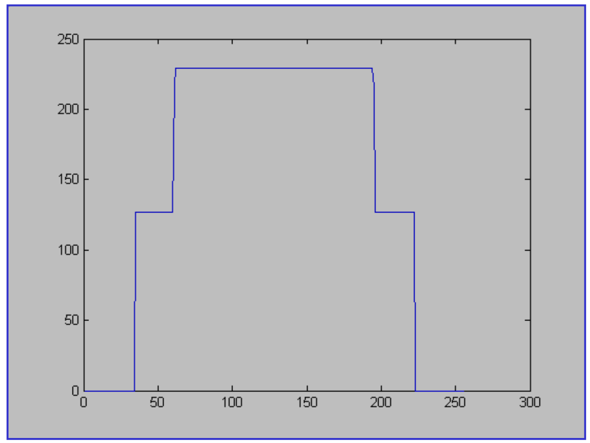
PDF of Gaussian noise: $p(z)=\frac{1}{\sqrt{2\pi}\sigma}e^{-(z-\mu)^2/2\sigma^2}$

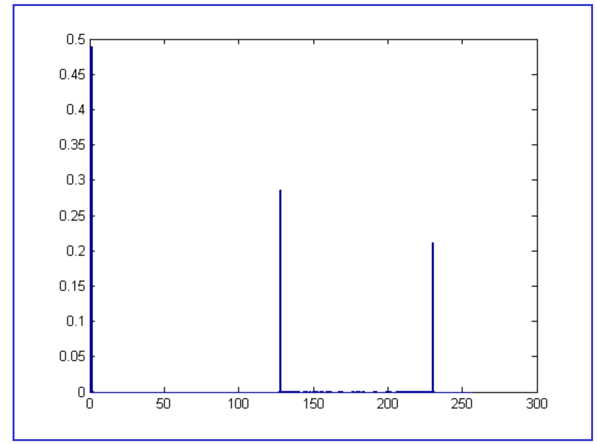


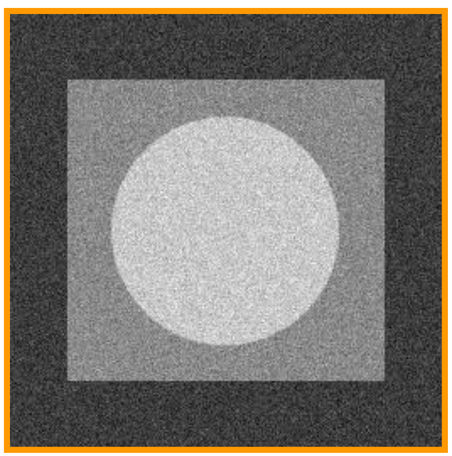
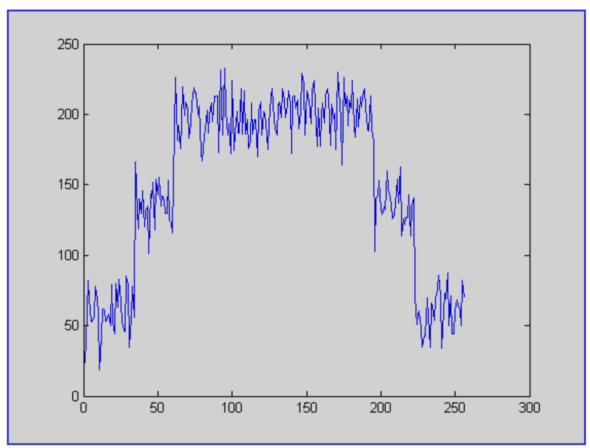
4.3 Image Smoothing

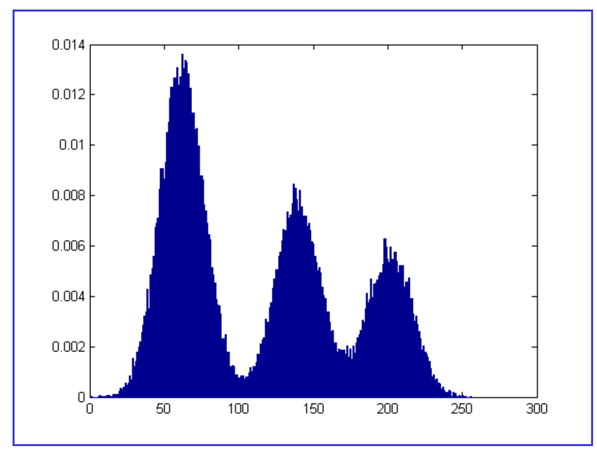
4.3.1 Introduction: Noise models

Gaussian noise $$g(x,y) = f(x,y) + \eta(x,y)$$



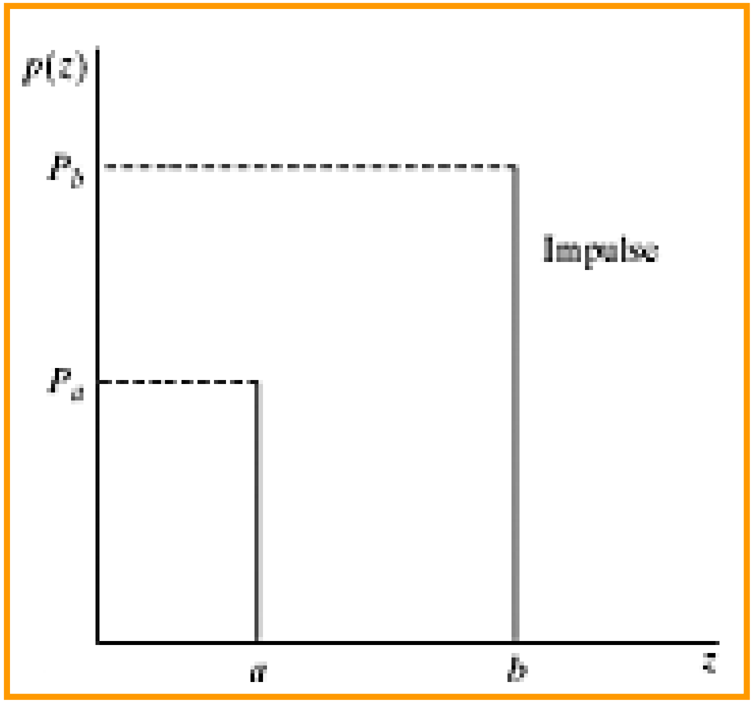


4.3 Image Smoothing

4.3.1 Introduction: Noise models

PDF of salt-pepper noise:

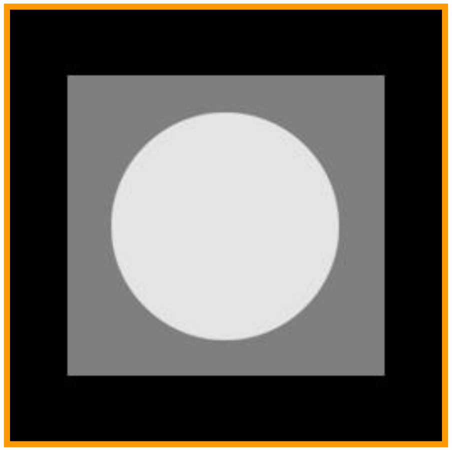
$$p(z) = \begin{cases} P_a & z = a \\ P_b & z = b \\ 0 & otherwise \end{cases}$$

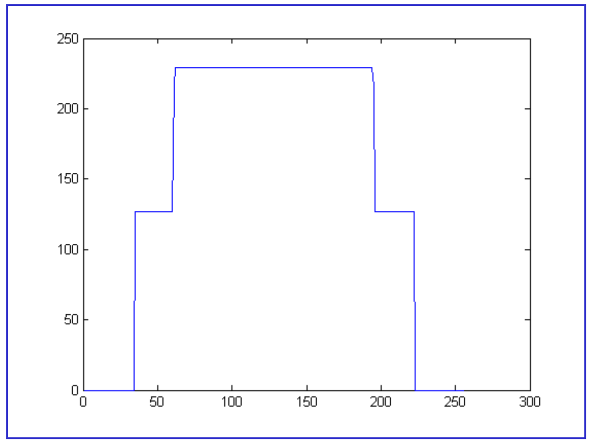


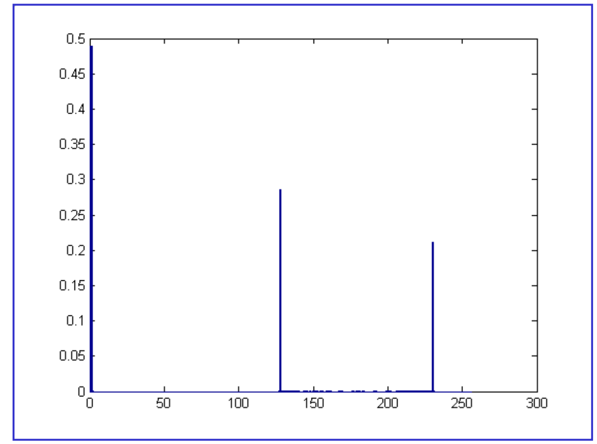
4.3 Image Smoothing

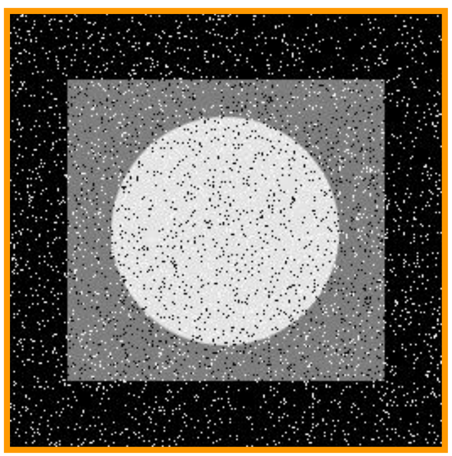
4.3.1 Introduction: Noise models

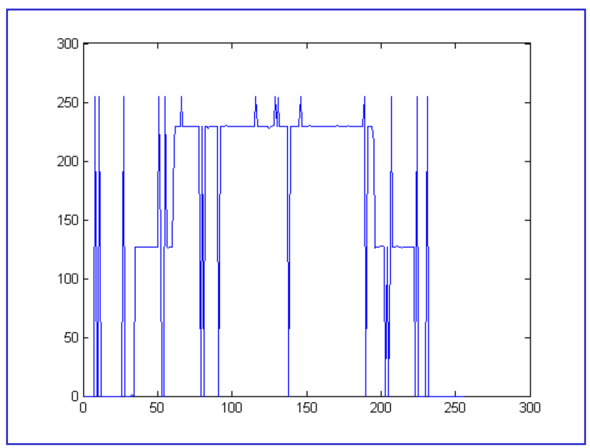
salt-pepper noise:

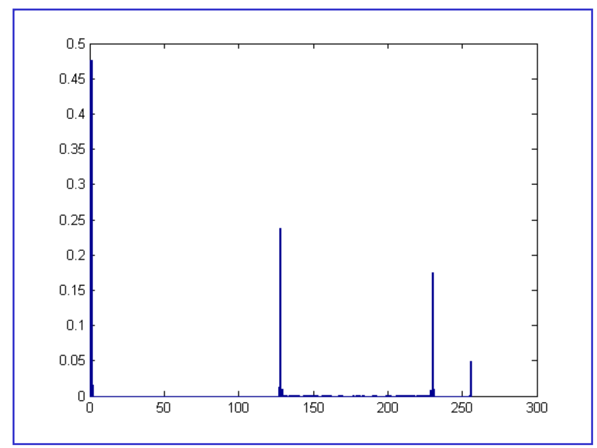












4.3 Image Smoothing

4.3.1 Introduction: Smoothing filters

(1) Smoothing linear filters: neighbor averaging
out range pixel smoothing
Maximum homogeneity smoothing

(2)Order- statistic filters: Max filters
Min filters
Midpoint filters
Median filters
Alpha-trimmed mean filter

(3)Low-pass filters: Idea low-pass filter (ILPF)
Butterworth low-pass filter (BLPF)
Gaussian low-pass filter (GLPF)

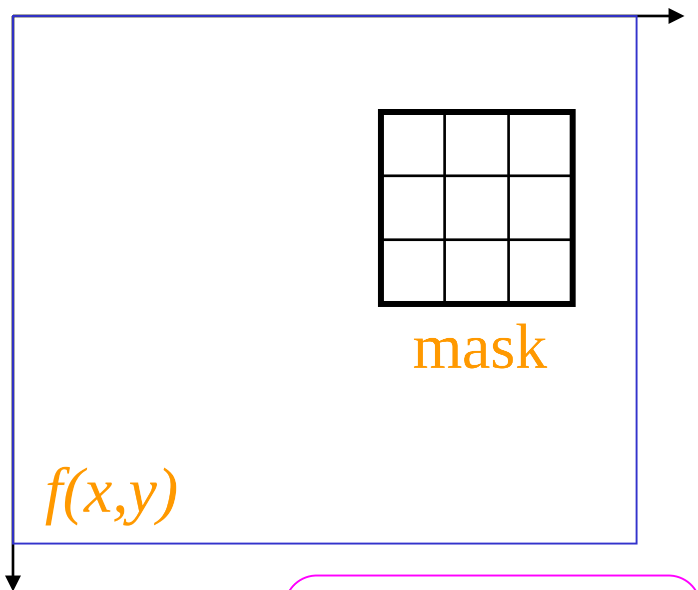
4.3 Image Smoothing

4.3.2 Smoothing linear filters: neighbor averaging

Spatial filter: mask, kernel, template, window

Mask coefficients

w(-1,-1)	*w(-1,0)*	*w(-1,1)*
w(0,-1)	*w(0,0)*	*w(0,-1)*
w(1,-1)	*w(1,0)*	*w(1,1)*



Pixels under mask

f(x-1,y-1)	*f(z-1,y)*	*f(x-1,y+1)*
f(x,y-1)	*f(x,y)*	*f(x,y+1)*
f(x+1,y-1)	*f(x+1,y)*	*f(x+1,y+1)*

4.3 Image Smoothing

4.3.2 Smoothing linear filters: neighbor averaging

Spatial filter: operating steps: (page 83)

(1) Moving the filter mask from point to point in an image

(2) Multiplying the filter coefficient and the corresponding image pixels

(3) Adding all the products

(4) The sum is the response of the filter at a given point

$$g(x,y)=\sum_{i=-a}^{a}\sum_{j=-b}^{b} w(i,j)\,f(x+i,y+j)$$

convolution

The convolution of two functions *f(x)* and *g(x)*, denoted by *f(x)*g(x)*, is defined by the integral:

$$f(x) * g(x) = \int_{-\infty}^{\infty} f(z)g(x-z)dz$$

Where *z* is a dummy variable of integration

convolution

Example1: graphic illustration of convolution *f(x)*g(x)*

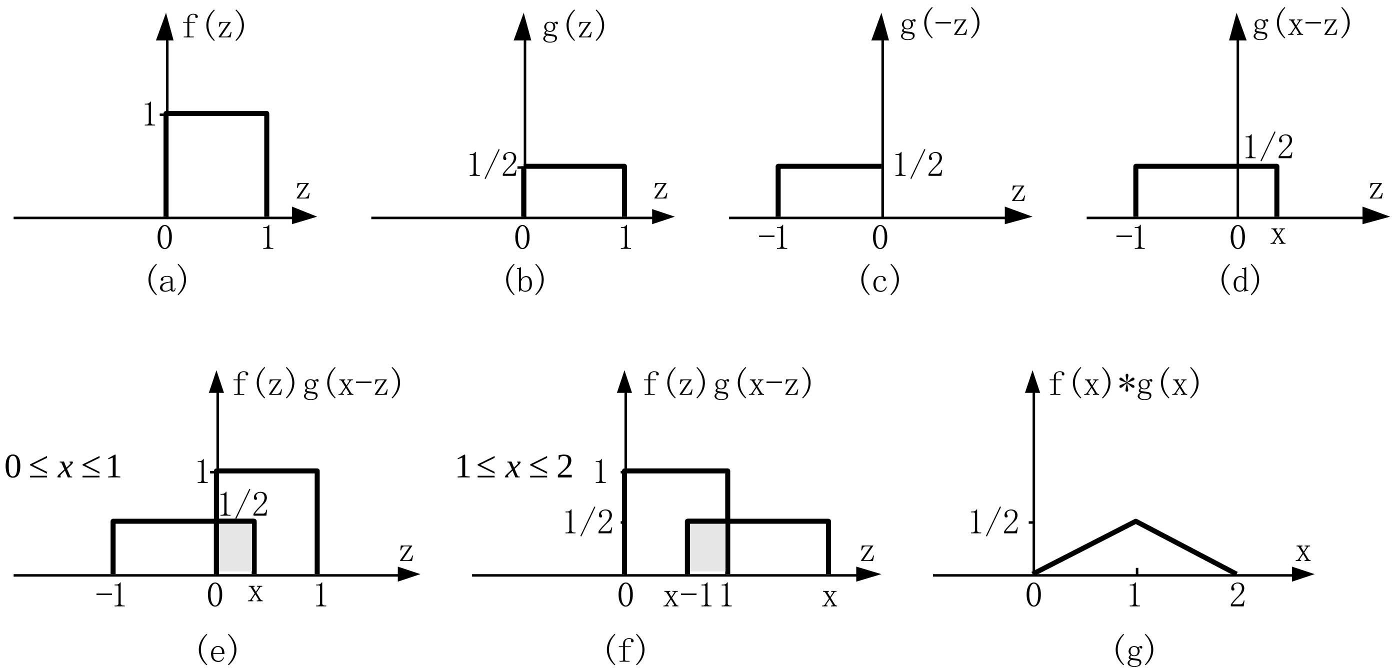
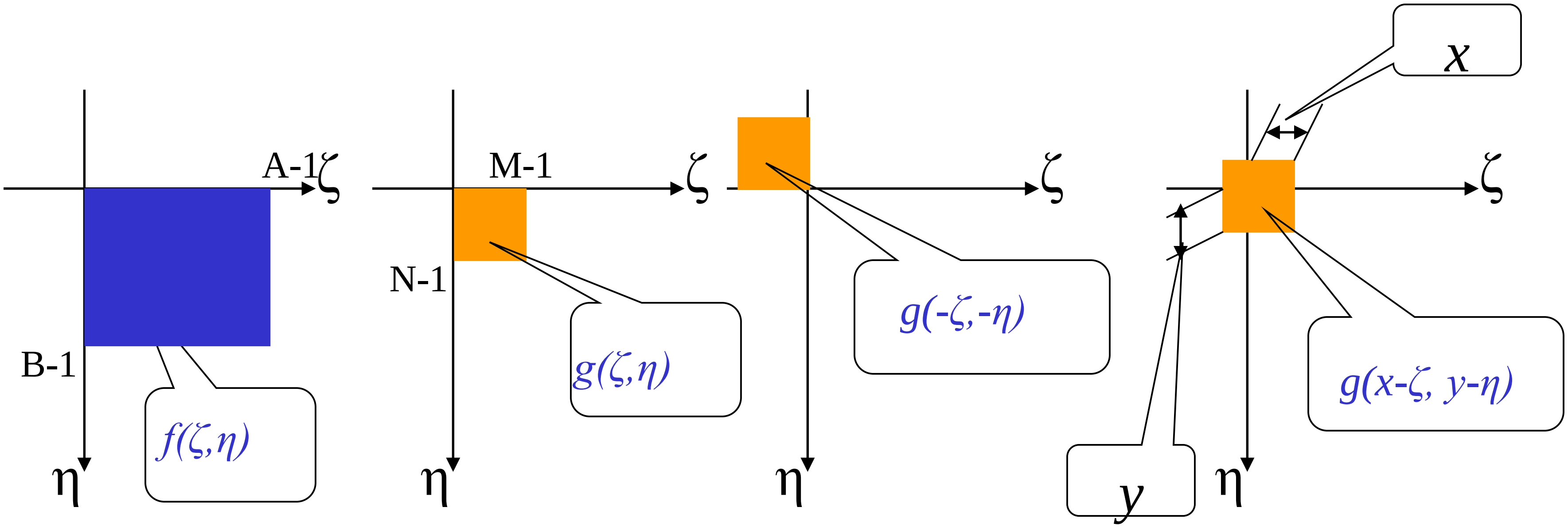


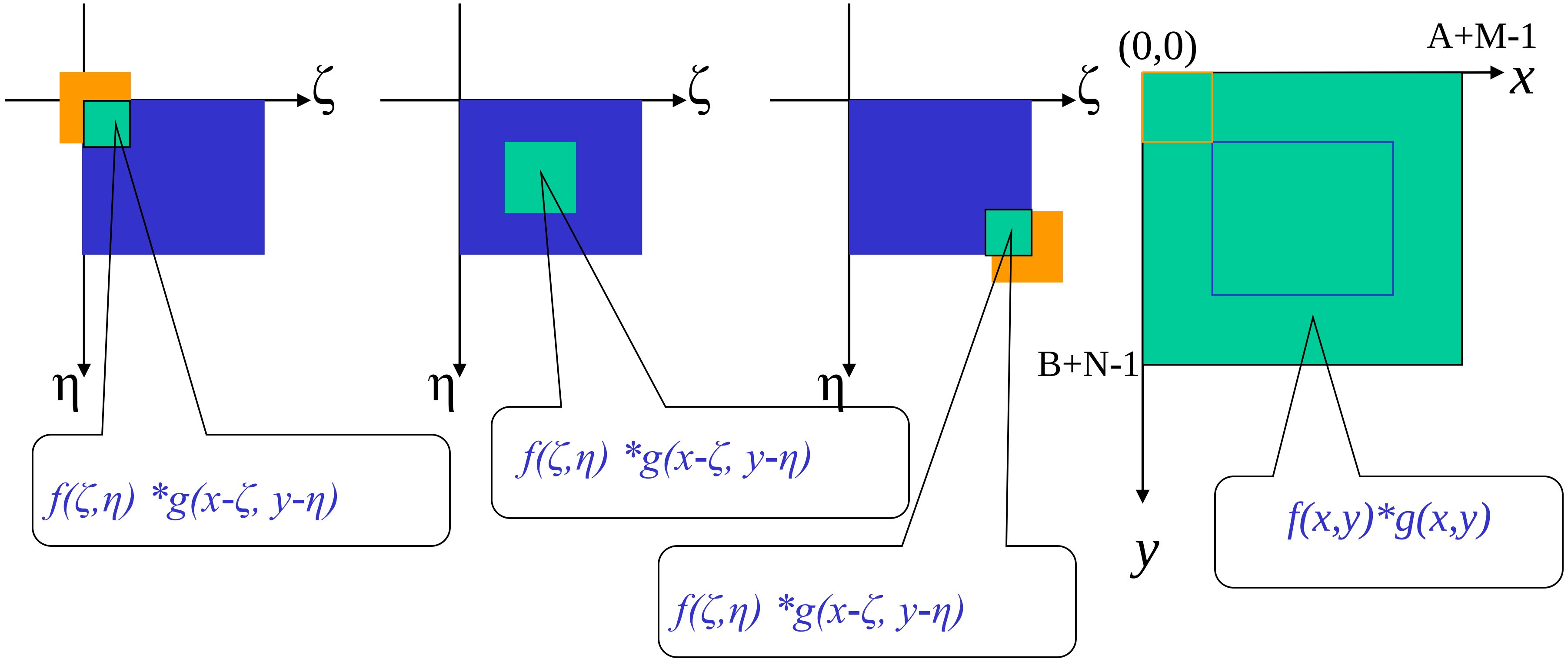
图3.2.3　1-D函数卷积示例

: convolution

Example2: graphic illustration of convolution $f(x,y)*g(x,y)$



convolution



4.3 Image Smoothing

4.3.2 Smoothing linear filters: neighbor averaging

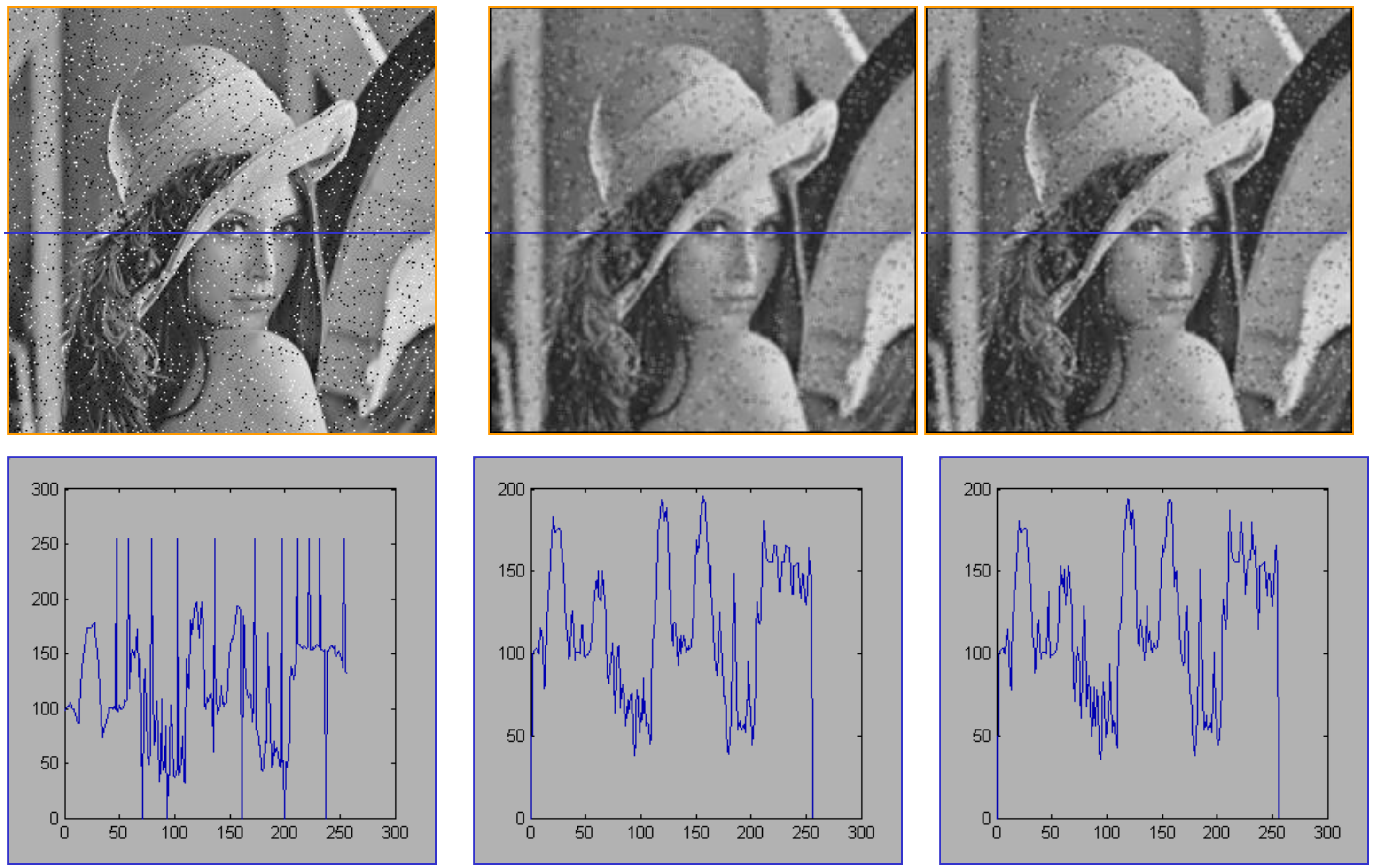
Typical mask:

$$\frac{1}{9}\times\begin{array}{|c|c|c|}\hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline\end{array}\qquad \frac{1}{16}\times\begin{array}{|c|c|c|}\hline 1 & 2 & 1 \\ \hline 2 & 4 & 2 \\ \hline 1 & 2 & 1 \\ \hline\end{array}$$

4.3 Image Smoothing

4.3.2 Smoothing linear filters: neighbor averaging

Experimental results: different masks

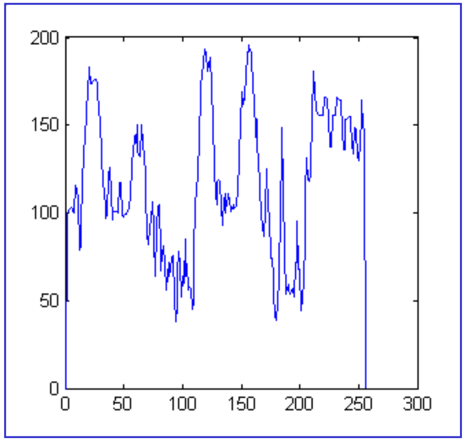


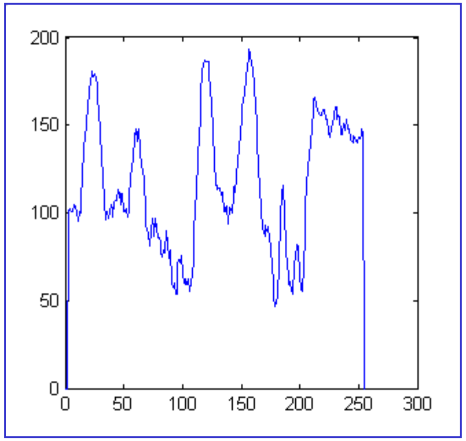
4.3 Image Smoothing

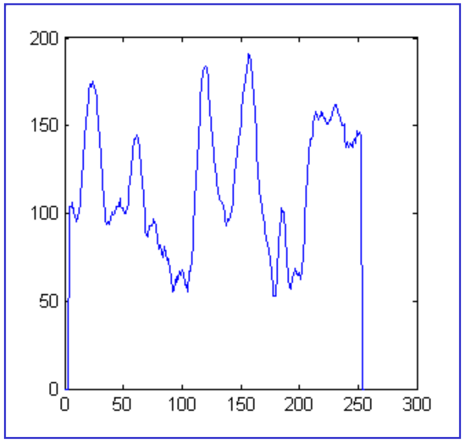
4.3.2 Smoothing linear filters: neighbor averaging

Experimental results:different sizes









4.3 Image Smoothing

4.3.2 Smoothing linear filters: out range pixel smoothing

formulate

$$g(x,y)=\sum_{i=-a}^{a}\sum_{j=-b}^{b}w(i,j)f(x+i,y+j)$$

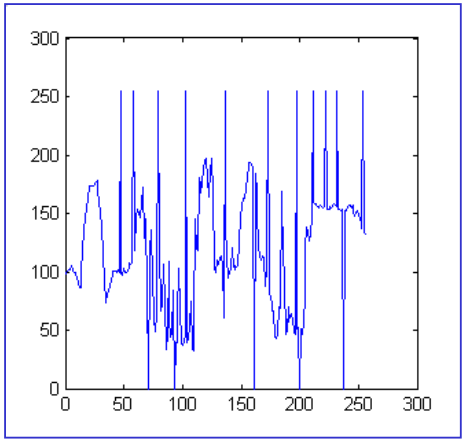
$$\hat{g}(x,y)=\begin{cases} g(x,y) & \left|g(x,y)-f(x,y)\right|>T \\ f(x,y) & \text{others} \end{cases}$$

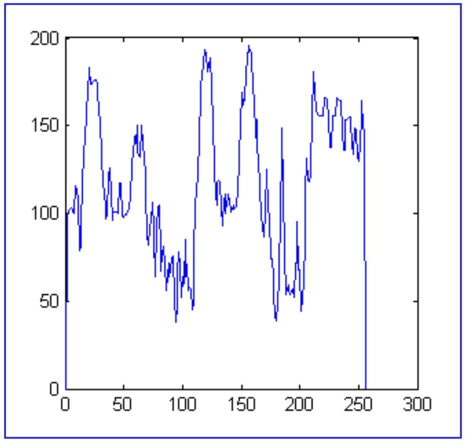
Advantage: keep details in the image with salt-pepper noise

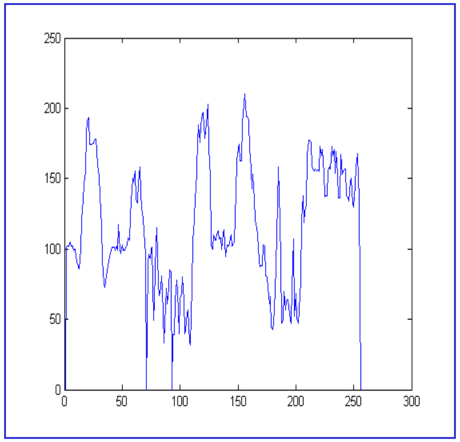
4.3 Image Smoothing

4.3.2 Smoothing linear filters: out range pixel smoothing

Experimental results

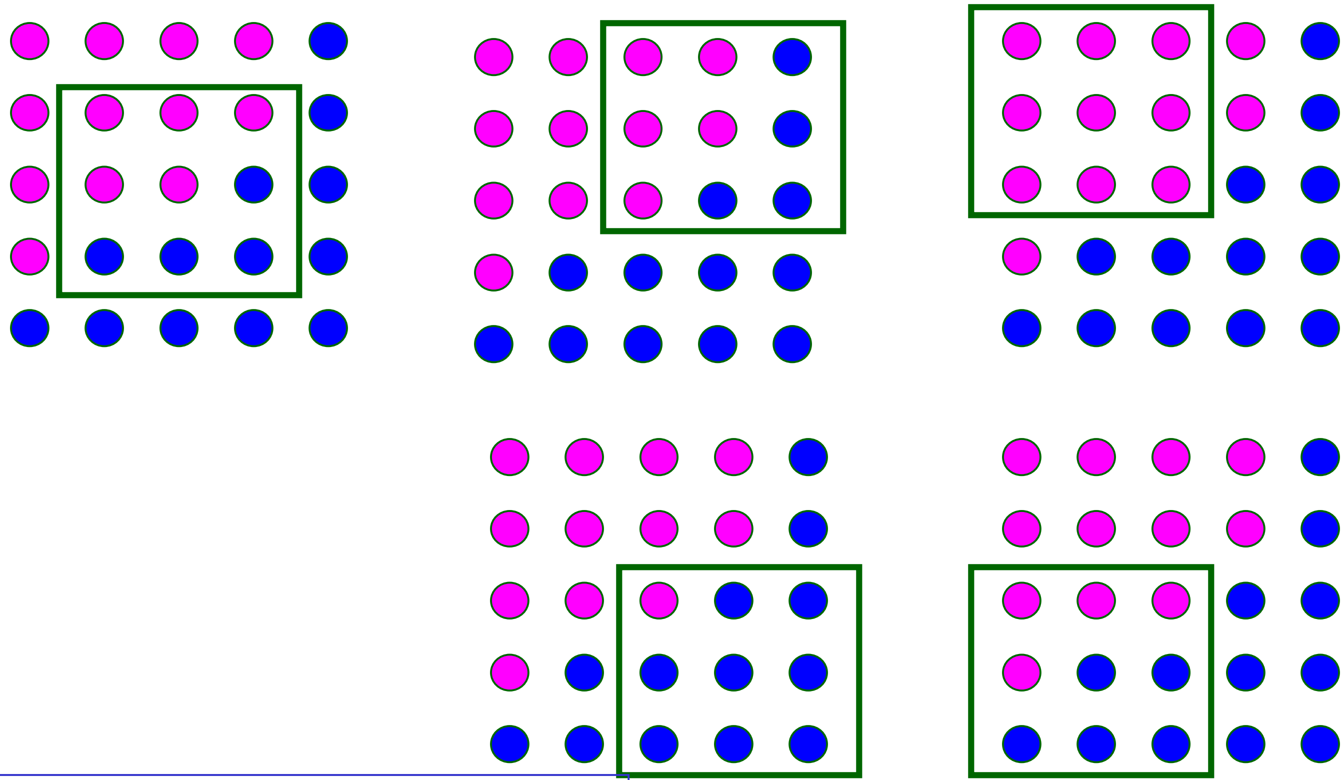






4.3 Image Smoothing

4.3.2 Smoothing linear filters: Maximum homogeneity smoothing

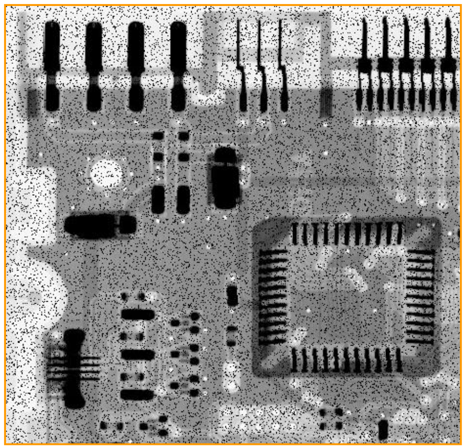


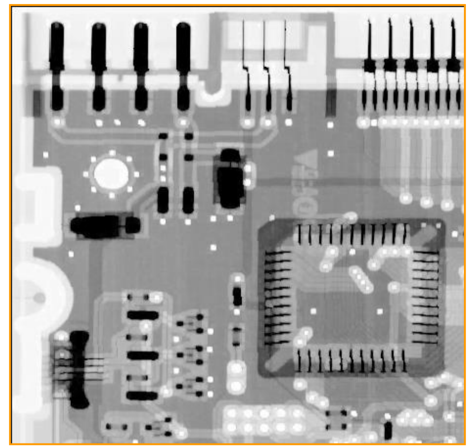
Advantage: keep edges

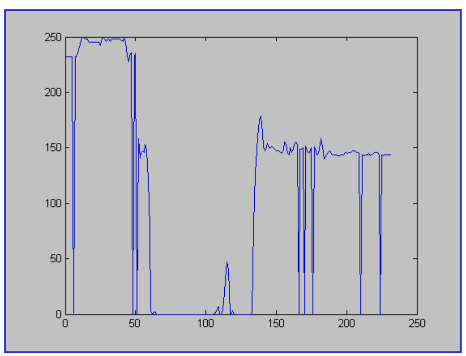
4.3 Image Smoothing

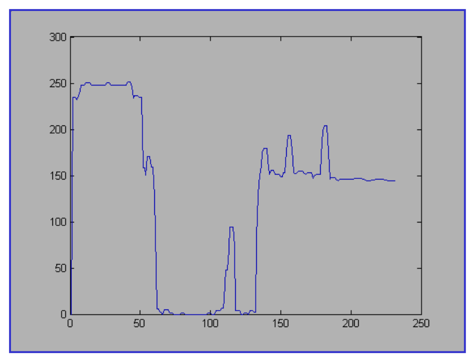
4.3.3 Order- statistic filters: Max filters

formulate $$g(x,y) = \max_{(i,j=0,\pm 1\cdots)} \{ f(x+i, y+j) \}$$







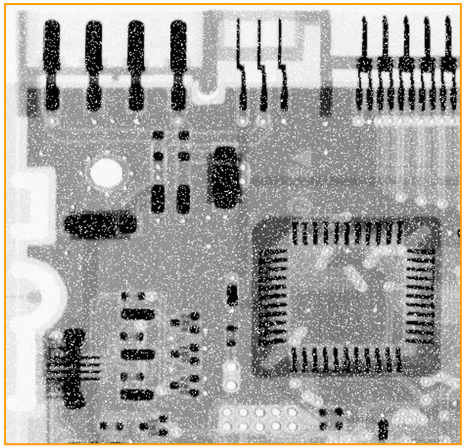


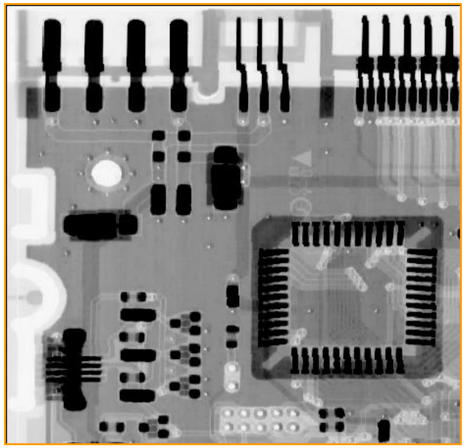
Advantage: removes pepper noise

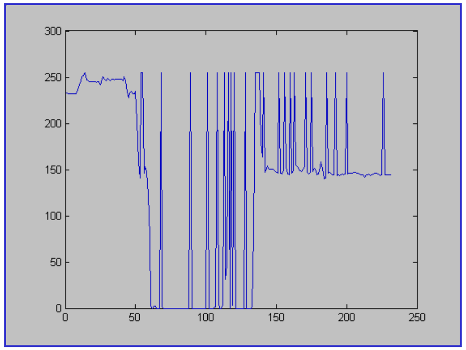
4.3 Image Smoothing

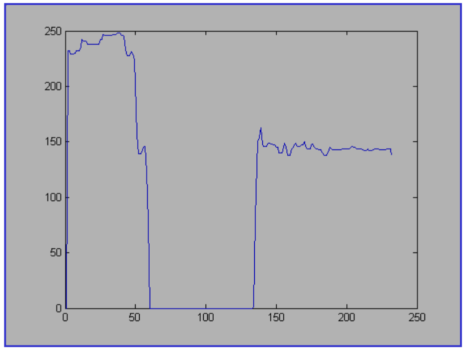
4.3.3 Order- statistic filters: Min filters

formulate $g(x,y) = \min\limits_{(i,j=0,\pm1\cdots)} \{f(x+i, y+j)\}$







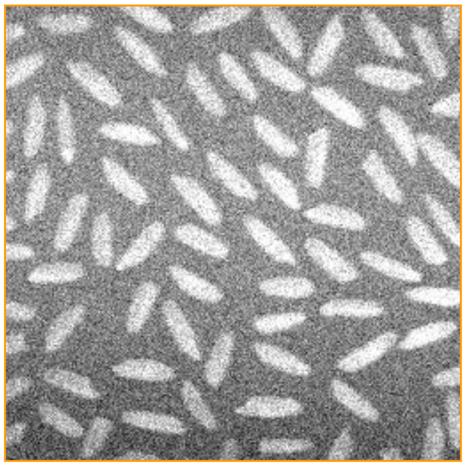


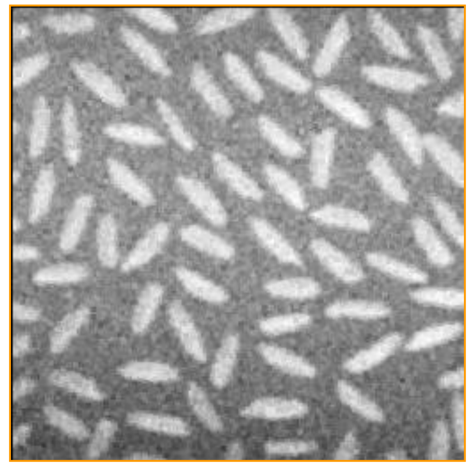
Advantage: removes salt noise

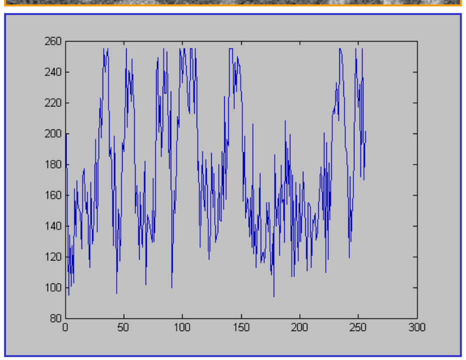
4.3 Image Smoothing

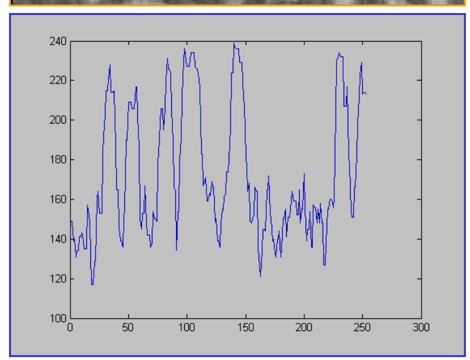
4.3.3 Order- statistic filters: Midpoint filter

formulate $$g(x,y)=\frac{1}{2}[\max_{(i,j=0,\pm1\cdots)}\{f(x+i,y+j)\}+\min_{(i,j=0,\pm1\cdots)}\{f(x+i,y+j)\}]$$









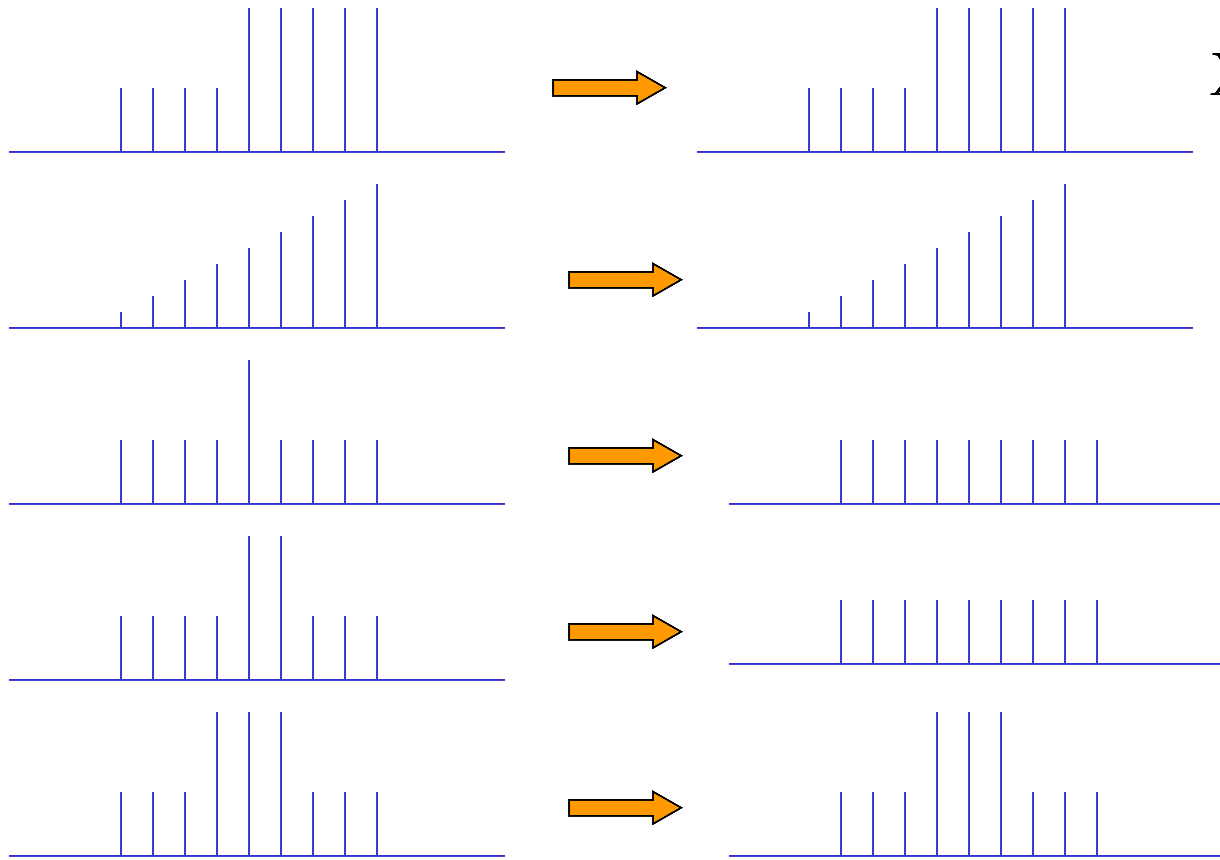
Advantage: removes Gausian noise

4.3 Image Smoothing

4.3.3 Order- statistic filters: Median filter

1-D: replace the signal value by the median value of neighborhood

L=5



X=[4 5 9 5 4]

[4 4 5 5 9]

[4 5 5 5 4]

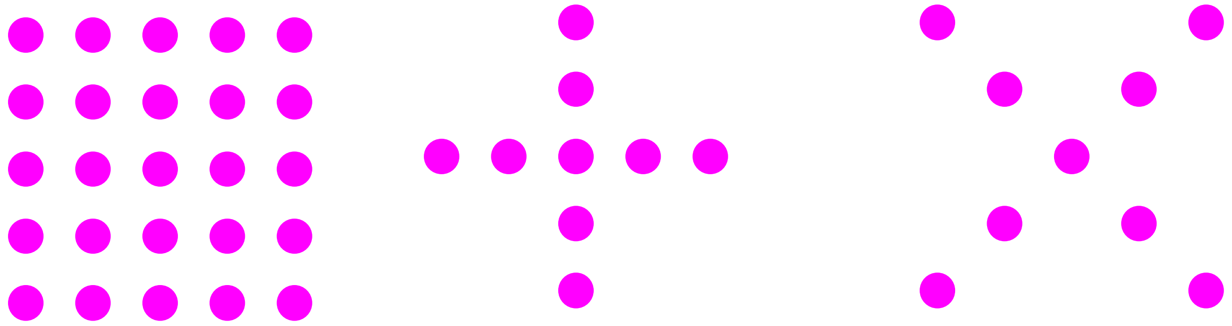
Advantage: keep edges

4.3 Image Smoothing

4.3.3 Order- statistic filters: Median filter

Replaces the value of a pixel by the median of the gray levels in the neighborhood of that pixel

$$g(x,y) = \underset{(i,j=0,\pm 1\cdots)}{median}\{f(x+i, y+j)\}$$



Shapes of 2-D filter

4.3 Image Smoothing

4.3.3 Order- statistic filters: Median filter

Experiment result

Gaussian noised

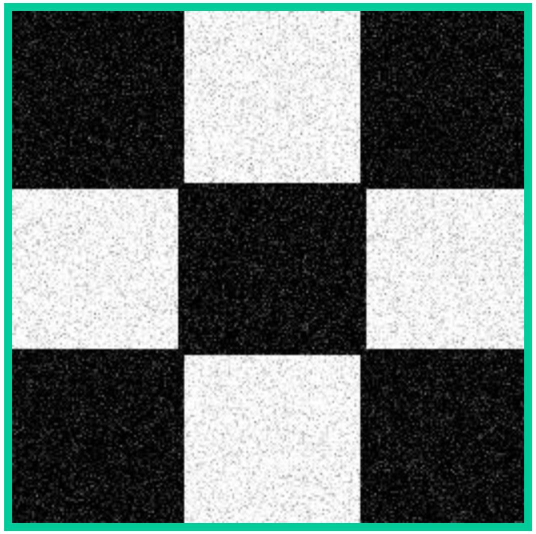
neighbor averaging

median filter

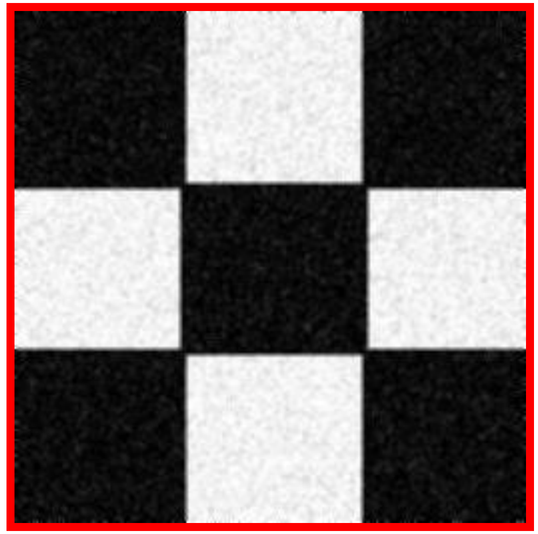
salt-pepper noised

neighbor averaging

median filter

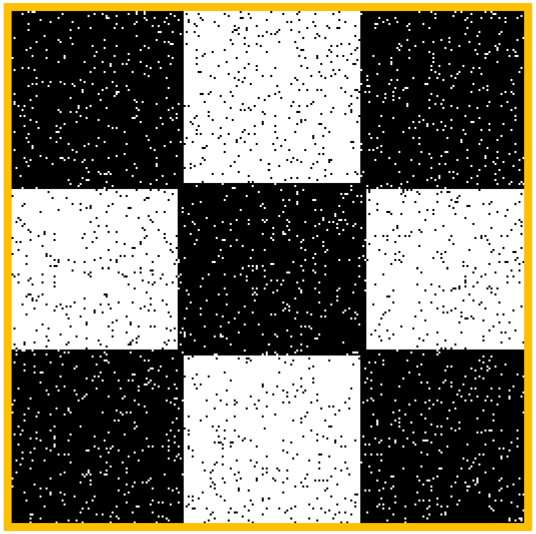


Gaussian noised

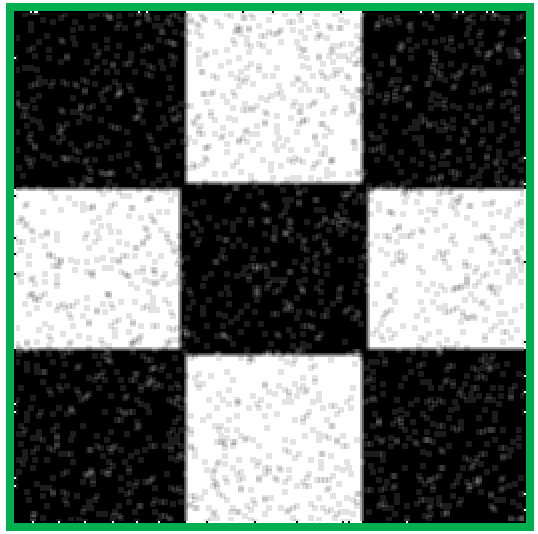


neighbor averaging

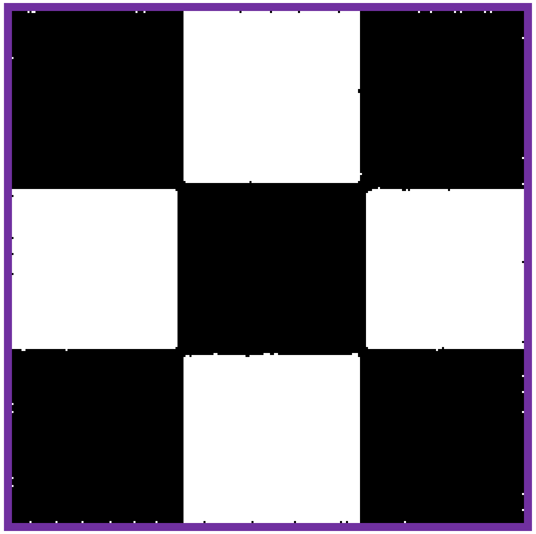
median filter



salt-pepper noised



neighbor averaging



median filter

4.3 Image Smoothing

4.3.3 Order- statistic filters: alpha-trimmed mean filter

Delete the α lowest and α highest gray-level values of a sub-image, averaging the remaining pixels as output

Let the pixels in a sub-image: $A_0 \le A_1 \cdots \le A_{N-1}$

Then:

$$g(x, y) = \frac{1}{N - 2\alpha} \sum_{i=\alpha}^{N-\alpha} A_i$$

4.3 Image Smoothing

4.3.4 Low-pass filters: Idea low-pass filter (ILPF)

formulate

$$g(x, y) = h(x, y) * f(x, y)$$

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

D_0: cutoff frequency

where

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

$$D(u,v) = [(u - M/2)^2 + (v - N/2)^2]^{1/2}$$

4.3 Image Smoothing

4.3.4 Low-pass filters: Idea low-pass filter(ILPF)

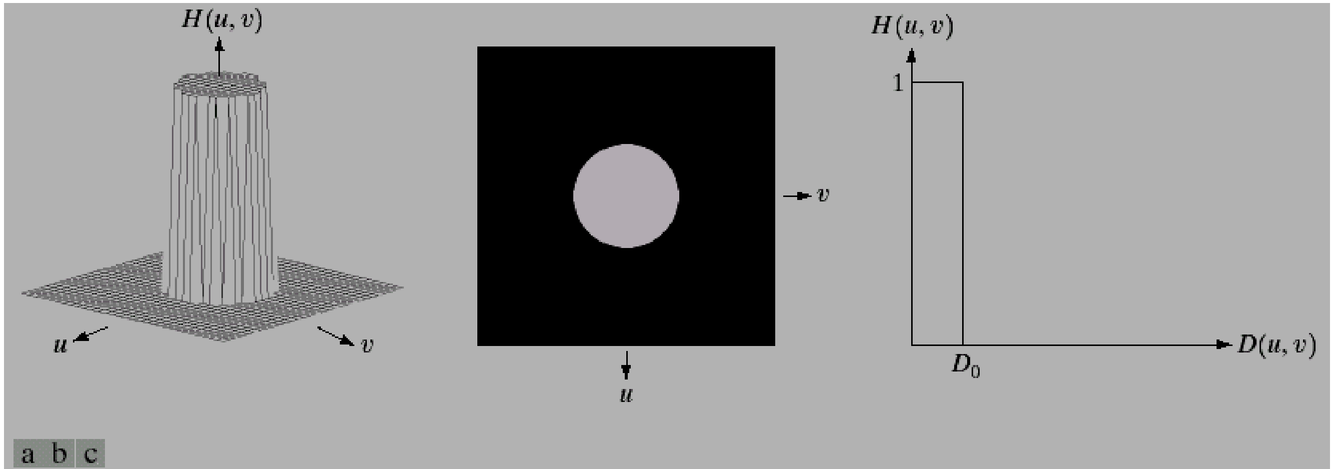


FIGURE 4.10 (a) Perspective plot of an ideal lowpass filter transfer function. (b) Filter displayed as an image. (c) Filter radial cross section.

4.3 Image Smoothing

4.3.4 Low-pass filters: Idea low-pass filter (ILPF)

Properties: total image power

$$P_T = \sum_{u=0}^{M-1}\sum_{v=0}^{N-1} P(u,v) = \sum_{u=0}^{M-1}\sum_{v=0}^{N-1} \left|F(u,v)\right|^2$$

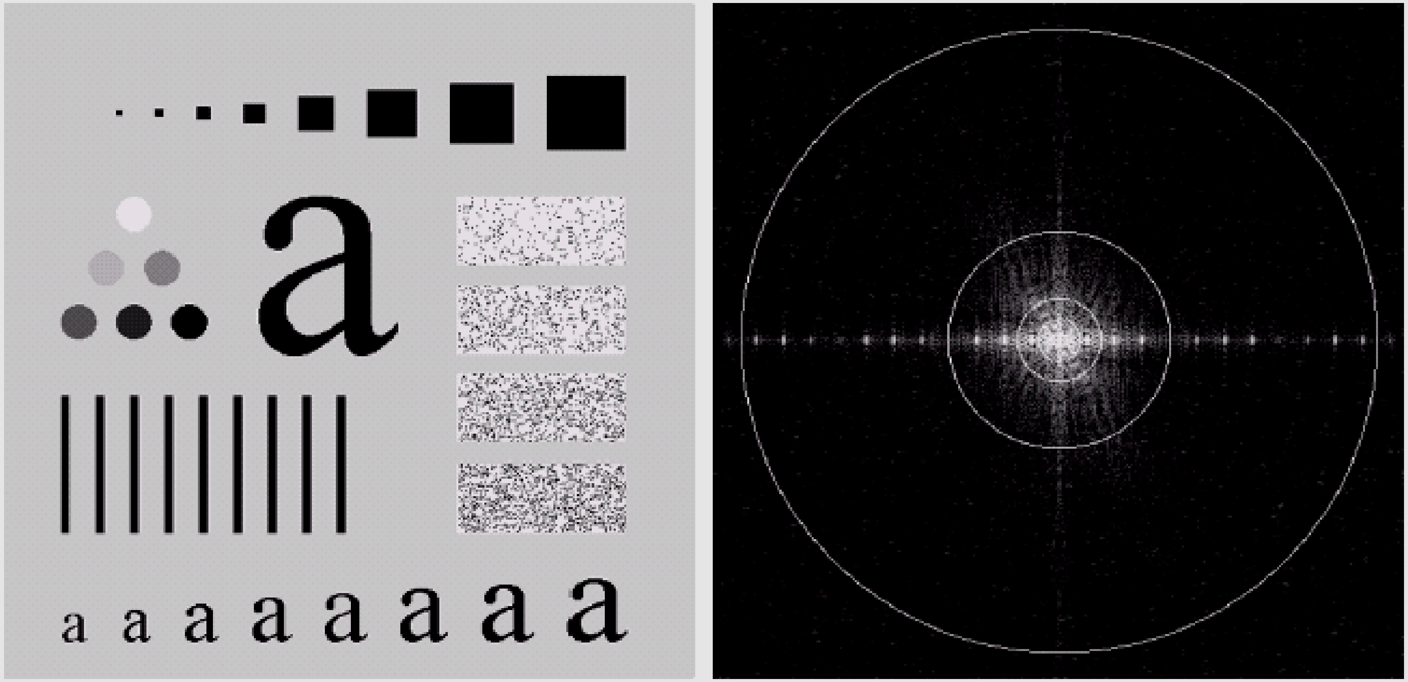
Power percent in a circle

$$\alpha = 100 \sum_{u=0}^{M-1}\sum_{v=0}^{N-1} P(u,v) / P_T$$

4.3 Image Smoothing

4.3.4 Low-pass filters: Idea low-pass filter (ILPF)

Properties: total image power



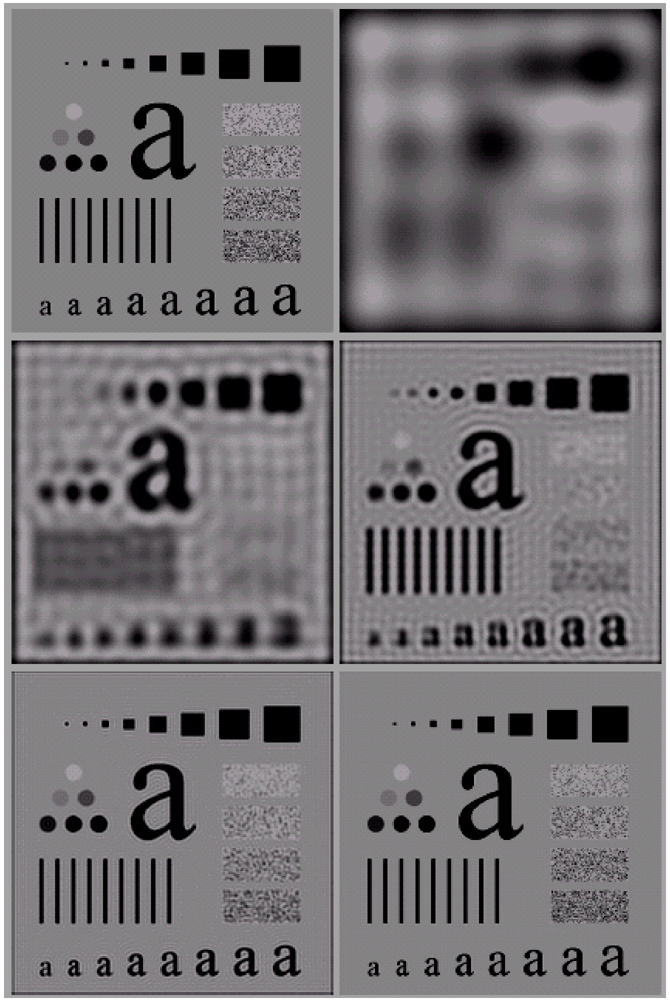
a b

FIGURE 4.11 (a) An image of size 500 × 500 pixels and (b) its Fourier spectrum. The superimposed circles have radii values of 5, 15, 30, 80, and 230, which enclose 92.0, 94.6, 96.4, 98.0, and 99.5% of the image power, respectively.

4.3 Image Smoothing

4.3.4 Low-pass filters: Idea low-pass filter (ILPF)

Properties: total image power



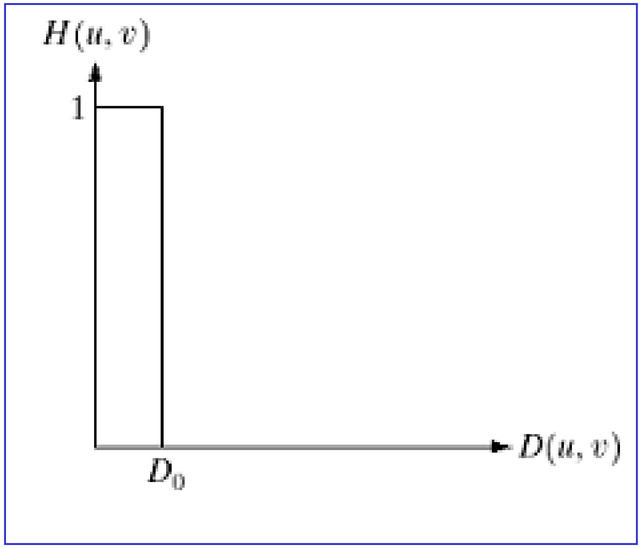
a b
c d
e f

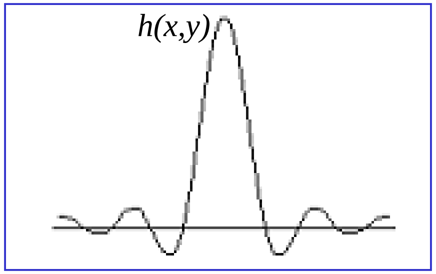
FIGURE 4.12 (a) Original image. (b)–(f) Results of ideal lowpass filtering with cutoff frequencies set at radii values of 5, 15, 30, 80, and 230, as shown in Fig. 4.11(b). The power removed by these filters was 8, 5.4, 3.6, 2, and 0.5% of the total, respectively.

4.3 Image Smoothing

4.3.4 Low-pass filters: Idea low-pass filter (ILPF)

Properties: blurring and ringing

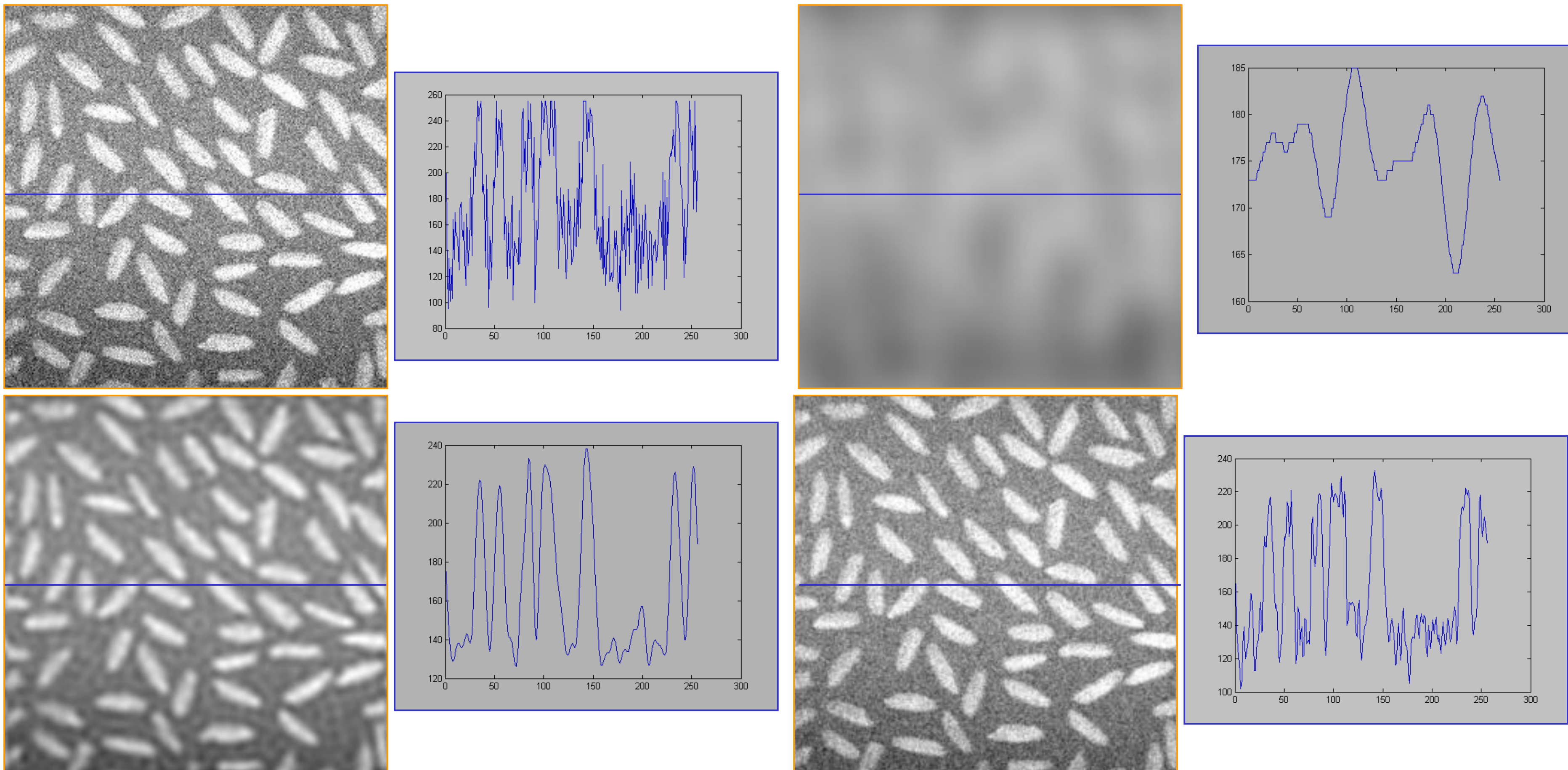




4.3 Image Smoothing

4.3.4 Low-pass filters: Idea low-pass filter (ILPF)

Experiment result cutoff frequencies set at radii values of 5、30、80



4.3 Image Smoothing

4.3.4 Low-pass filters: Butterworth low-pass filter

formulate

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

where

$$D(u,v) = [(u - M/2)^2 + (v - N/2)^2]^{1/2}$$

4.3 Image Smoothing

4.3.4 Low-pass filters: Butterworth low-pass filter

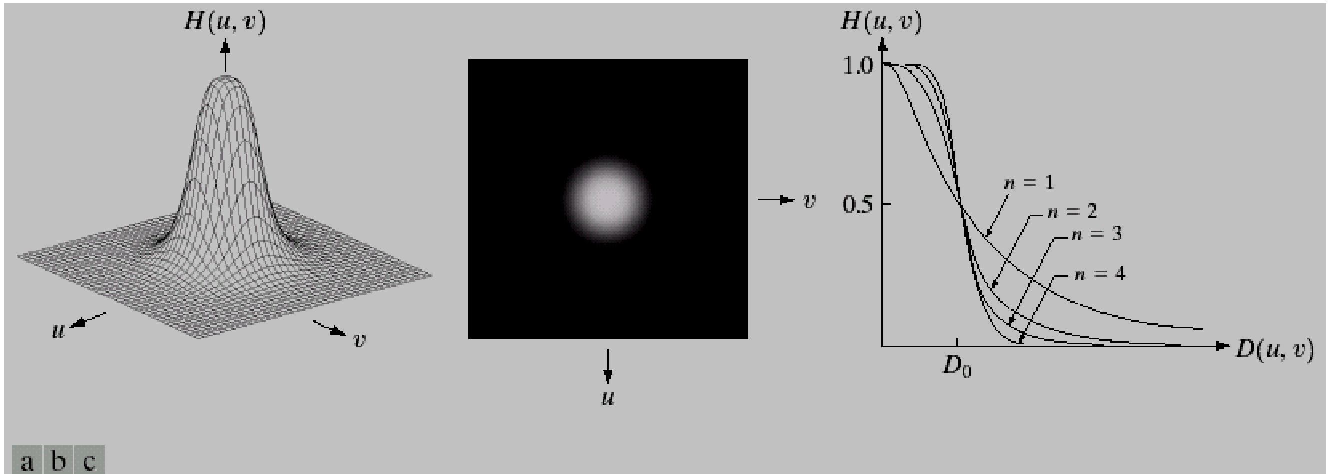


FIGURE 4.14 (a) Perspective plot of a Butterworth lowpass filter transfer function. (b) Filter displayed as an image. (c) Filter radial cross sections of orders 1 through 4.

$H(u,v)=0.5$ when $D(u,v)=D_0$

4.3 Image Smoothing

4.3.4 Low-pass filters: Butterworth low-pass filter

Properties: blurring and ringing

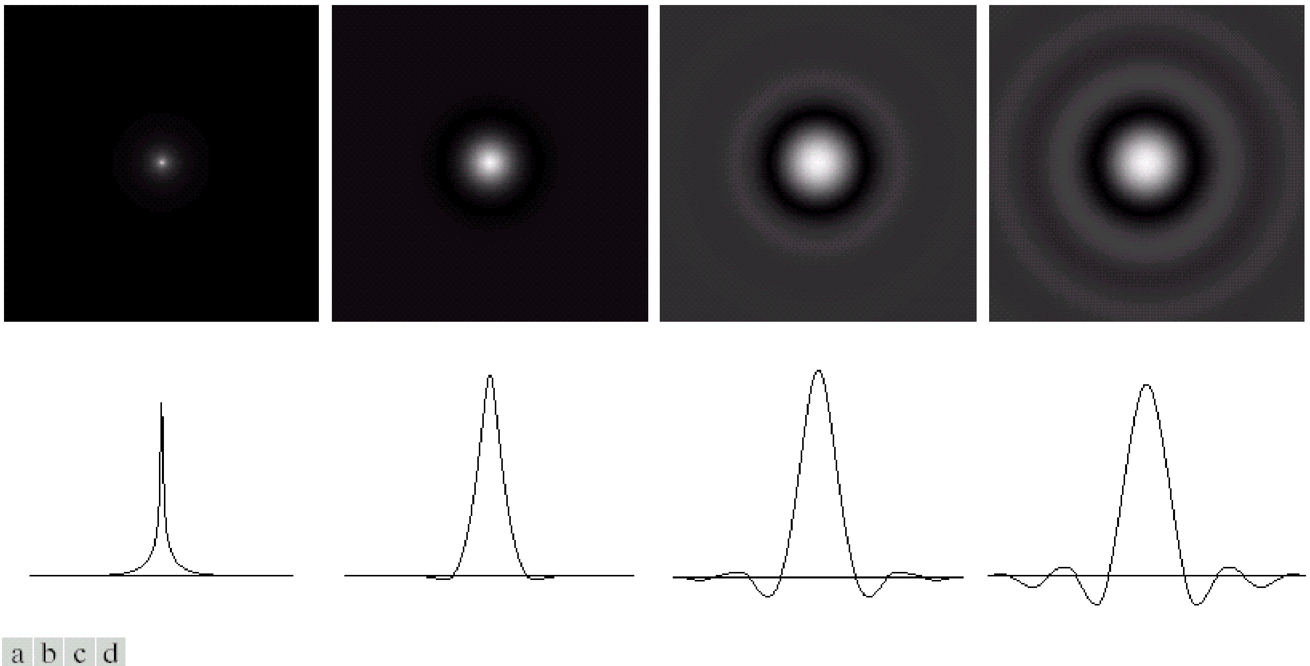
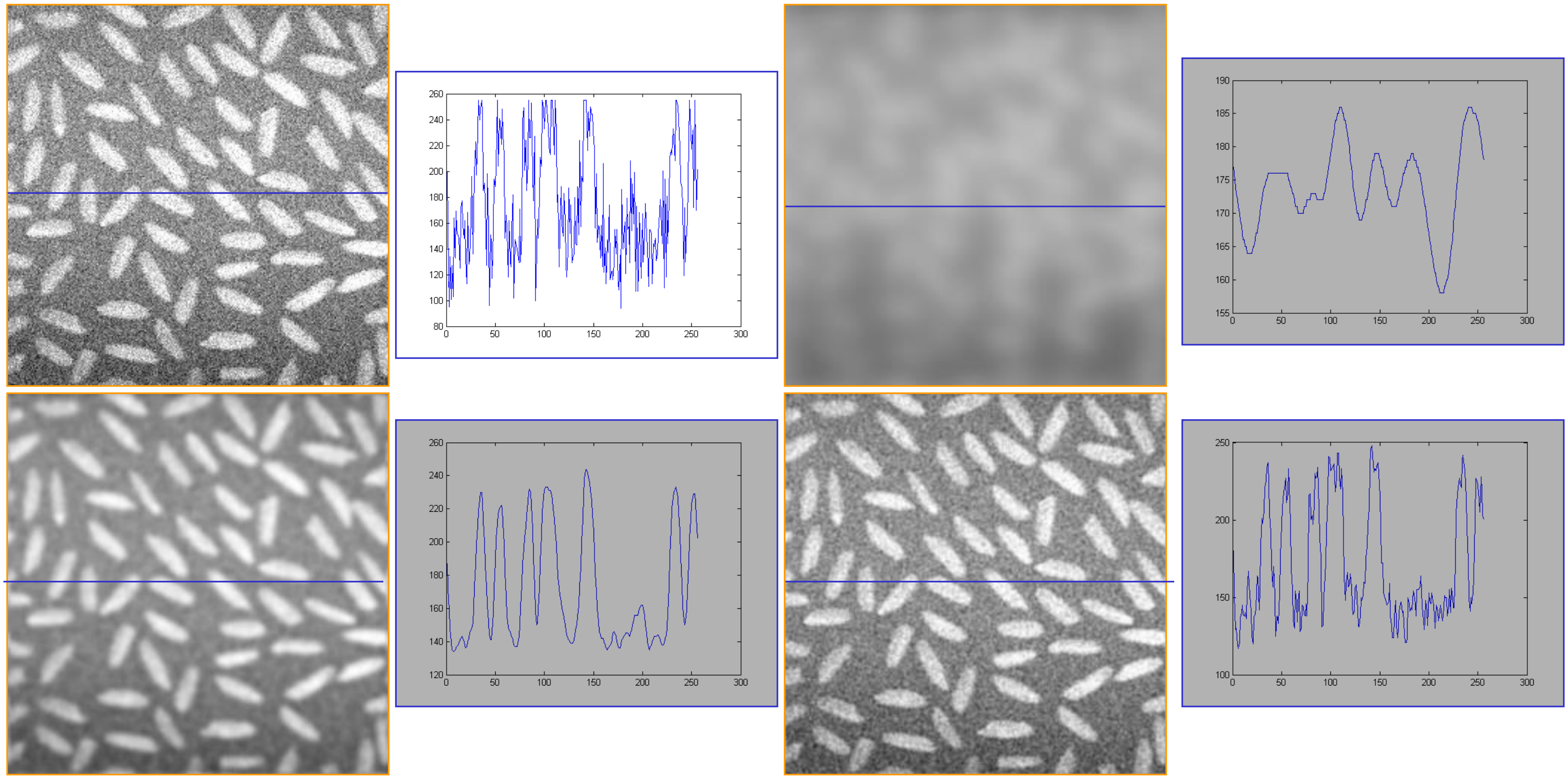


FIGURE 4.16 (a)–(d) Spatial representation of BLPFs of order 1, 2, 5, and 20, and corresponding gray-level profiles through the center of the filters (all filters have a cutoff frequency of 5). Note that ringing increases as a function of filter order.

4.3 Image Smoothing

4.3.4 Low-pass filters: Butterworth low-pass filter

Experiment result cutoff frequencies set at radii values of 5、30、80



4.3 Image Smoothing

4.3.4 Low-pass filters: Gaussian low-pass filter

formulate

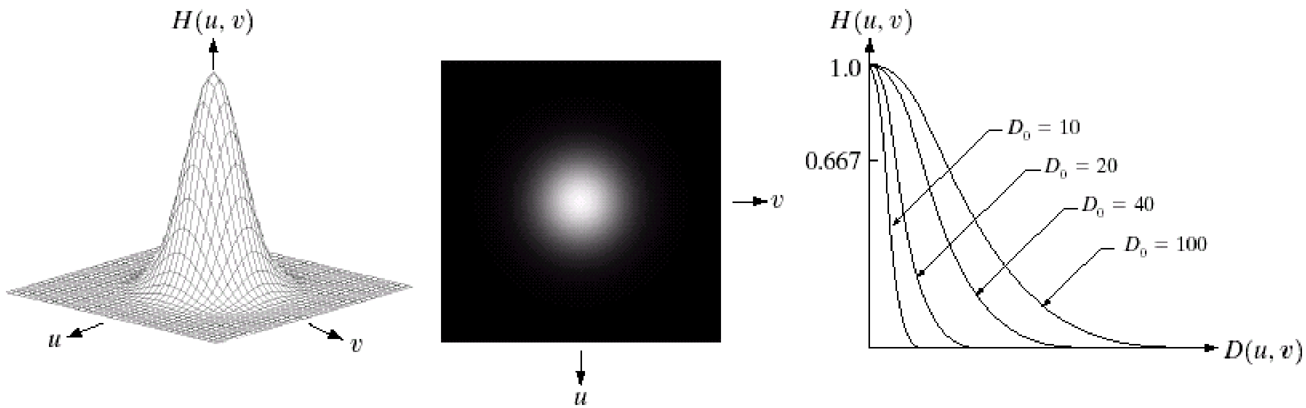
$$H(u,v) = e^{-D^2(u,v)/2D_0{}^2}$$

where

$$D(u,v) = [(u - M/2)^2 + (v - N/2)^2]^{1/2}$$

4.3 Image Smoothing

4.3.4 Low-pass filters: Gaussian low-pass filter



a b c

FIGURE 4.17 (a) Perspective plot of a GLPF transfer function. (b) Filter displayed as an image. (c) Filter radial cross sections for various values of D_0.

Inverse Fourier transform of the Gaussian lowpass filter also is Gaussian

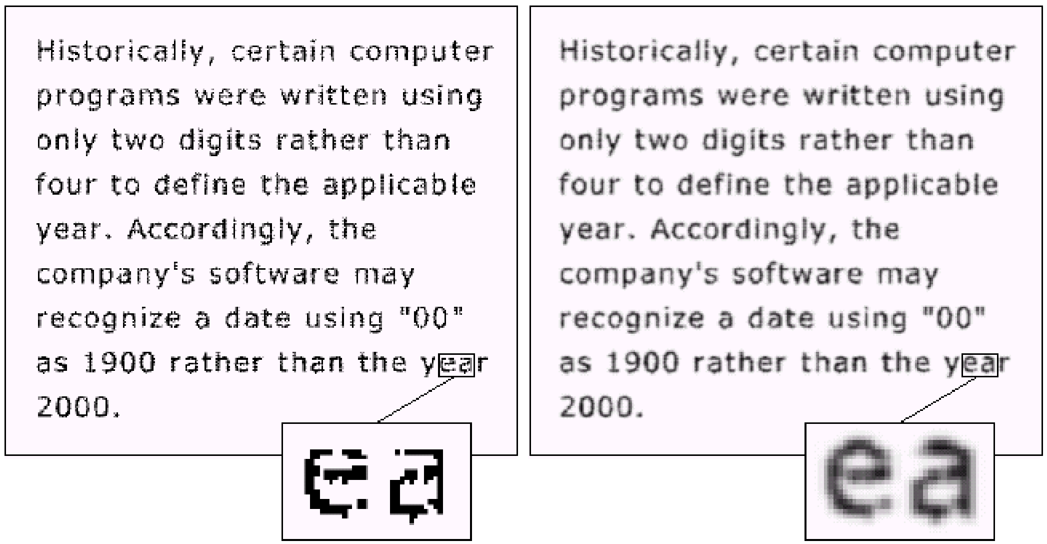
4.3 Image Smoothing

4.3.4 Low-pass filters: Gaussian low-pass filter

Applications: machine perception

a b

FIGURE 4.19 (a) Sample text of poor resolution (note broken characters in magnified view). (b) Result of filtering with a GLPF (broken character segments were joined).



4.3 Image Smoothing

4.3.4 Low-pass filters: Gaussian low-pass filter

Applications: printing and publishing industry

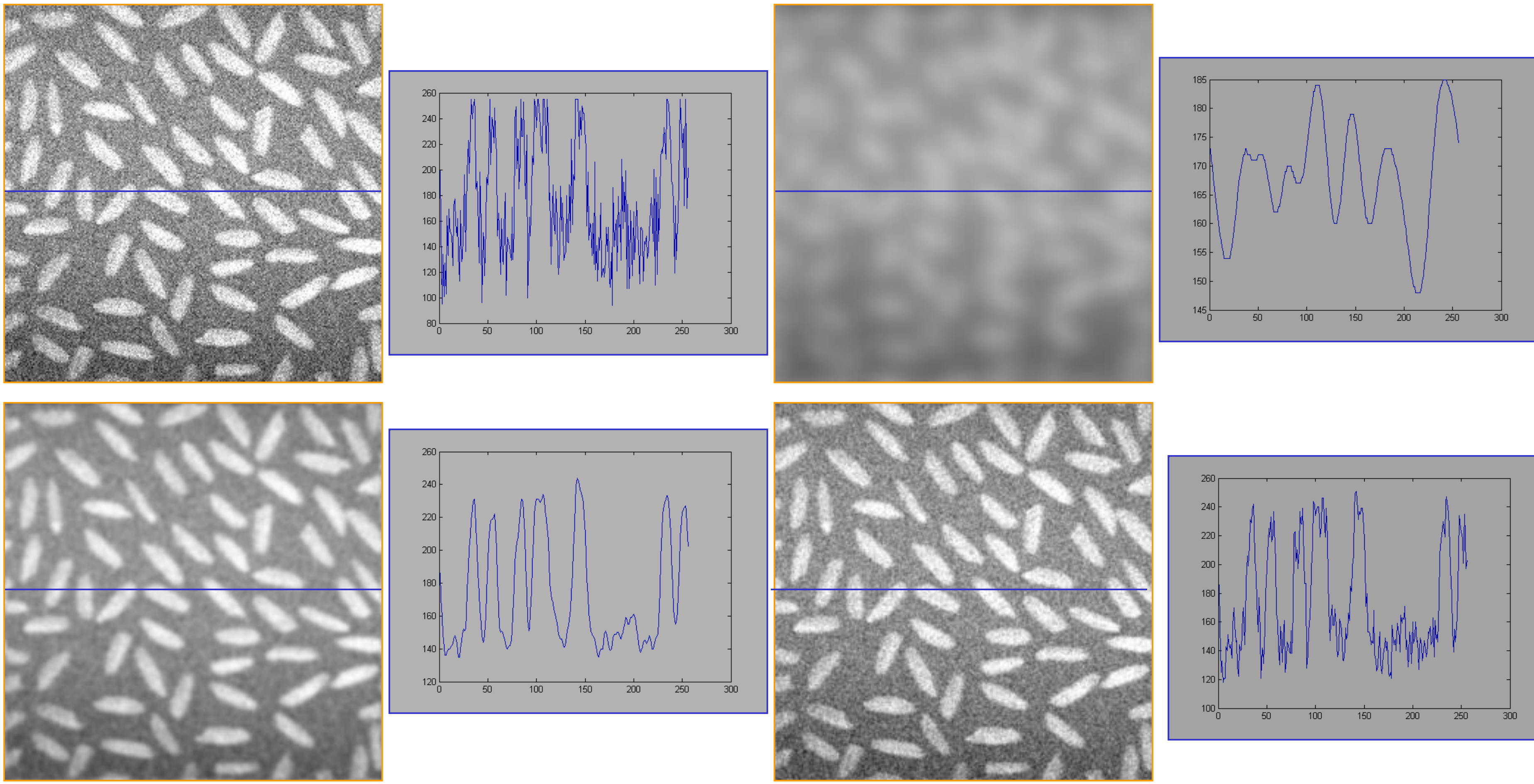
a b c

FIGURE 4.20 (a) Original image (1028 × 732 pixels). (b) Result of filtering with a GLPF with $D_0 = 100$. (c) Result of filtering with a GLPF with $D_0 = 80$. Note reduction in skin fine lines in the magnified sections of (b) and (c).

4.3 Image Smoothing

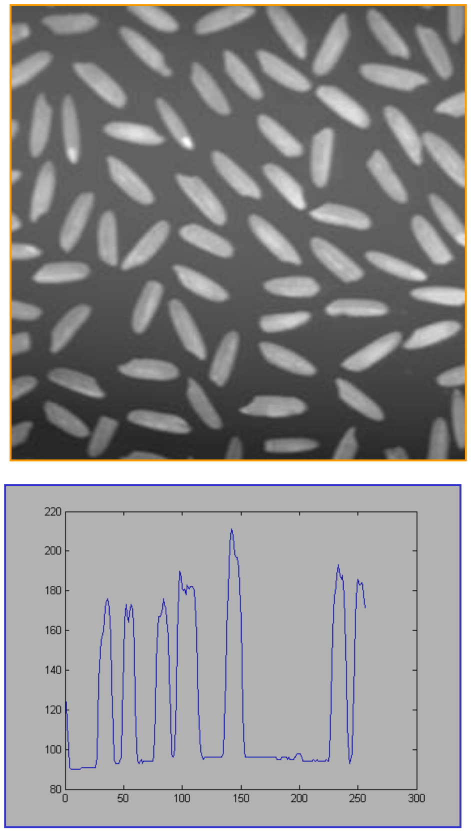
4.3.4 Low-pass filters: Gaussian low-pass filter

Experiment result cutoff frequencies set at radii values of 5、30、80

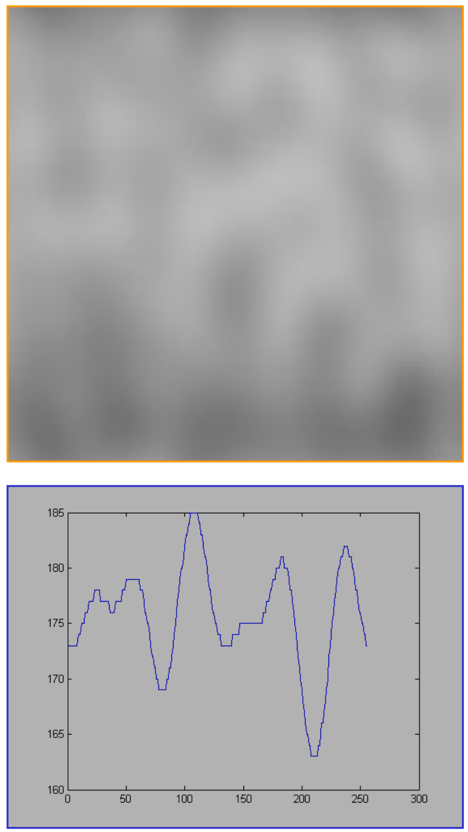


4.3 Image Smoothing

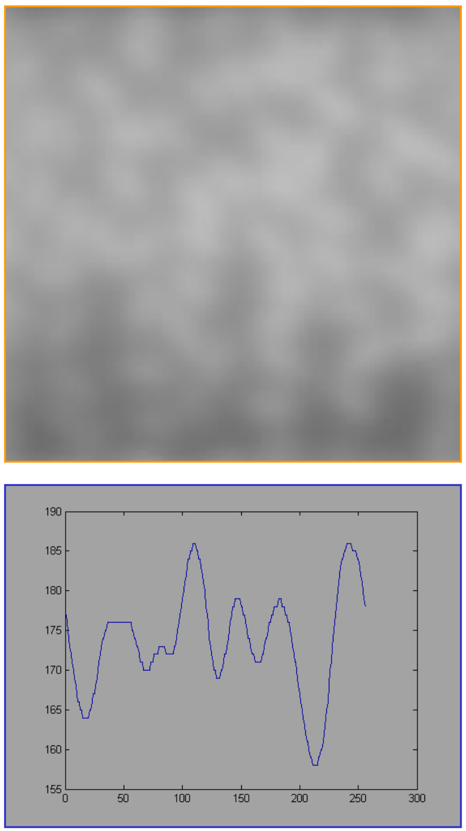
4.3.4 Low-pass filters: Comparisons (cutoff 5)



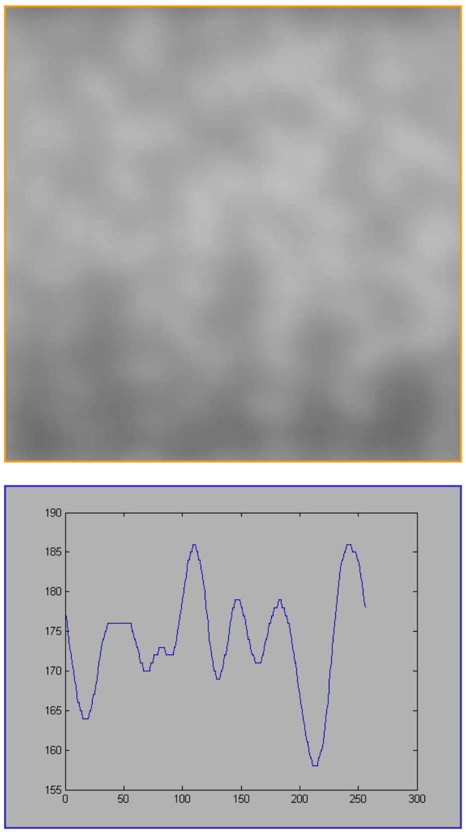
Origin



Ideal filter



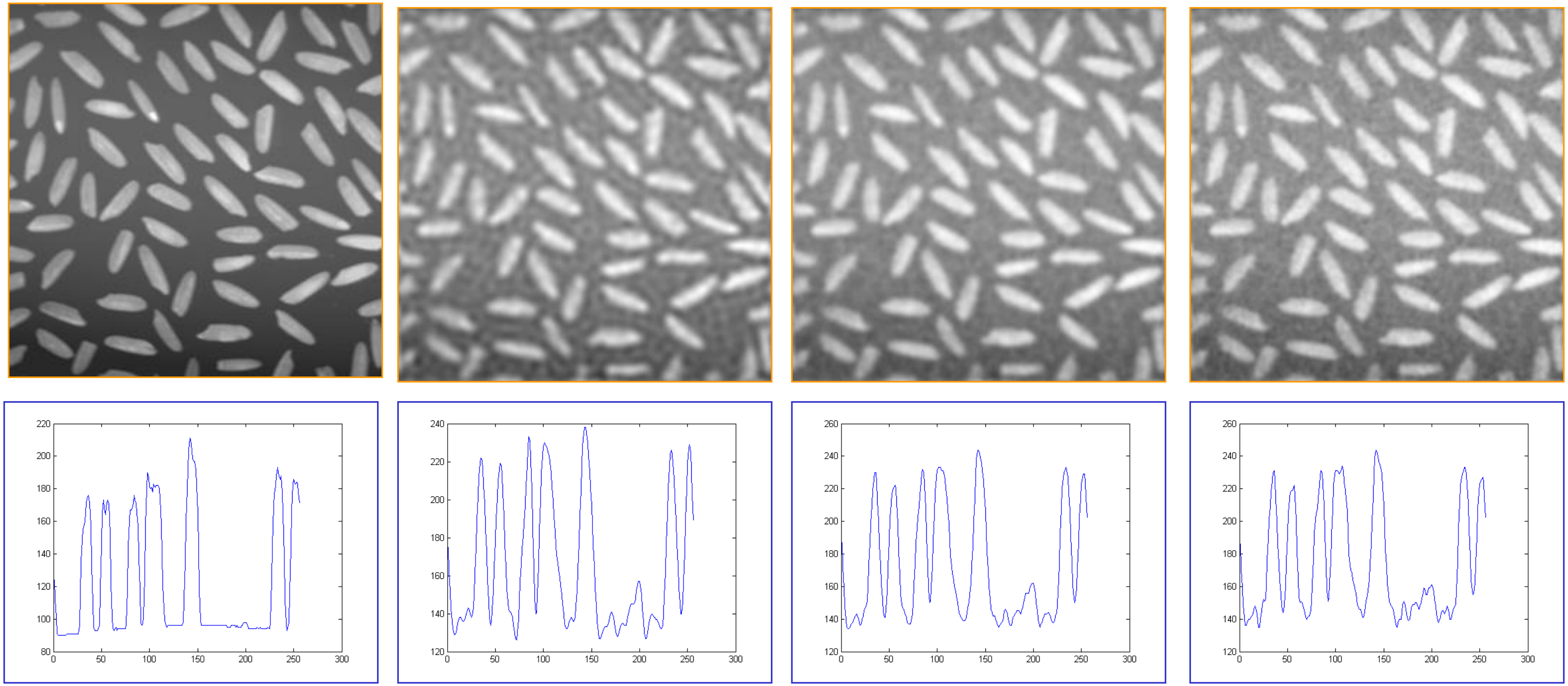
Butterworth filter



Gaussian filter

4.3 Image Smoothing

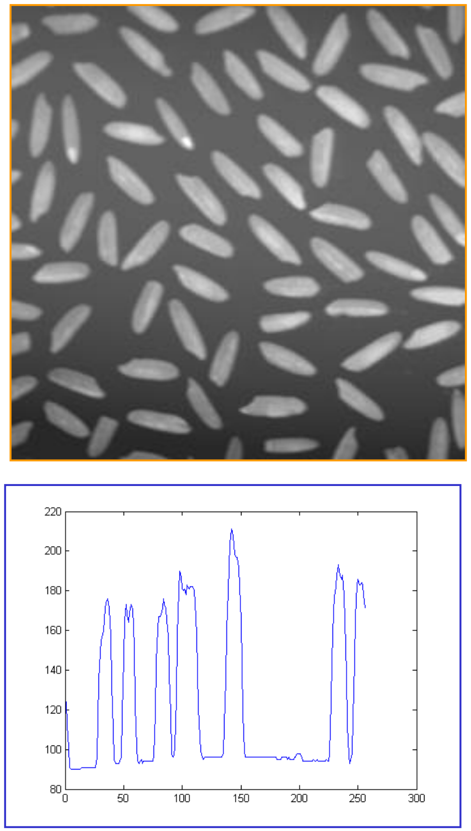
4.3.4 Low-pass filters: Comparison (cutoff 30)



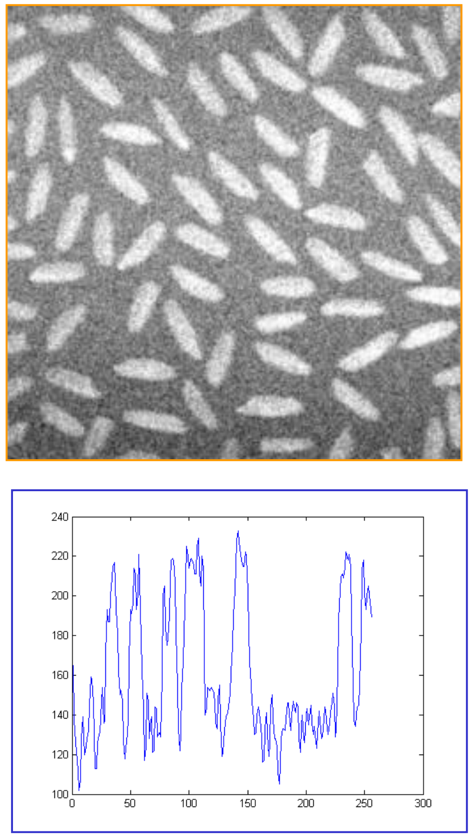
Origin | Ideal filter | Butterworth filter | Gaussian filter

4.3 Image Smoothing

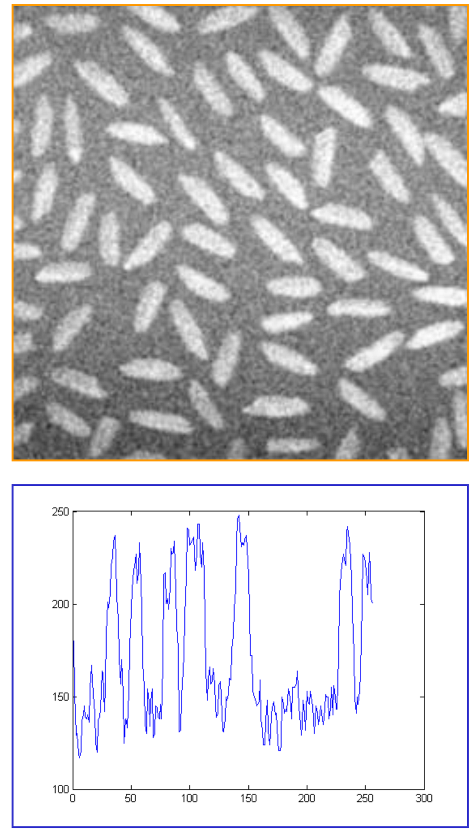
4.3.4 Low-pass filters: Comparison (cutoff 80)



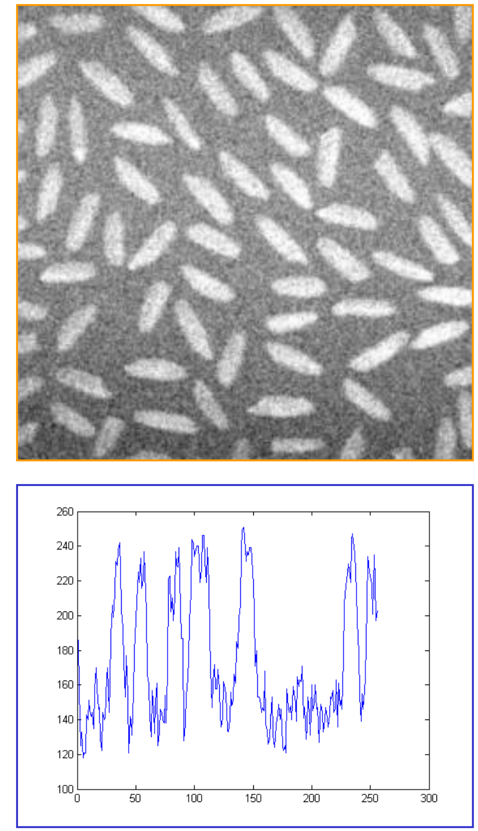
Origin



Ideal filter



Butterworth filter



Gaussian filter

The End