Digital Image Processing

Image Enhancement in the Spatial Domain

Chang-Su Kim

Image Enhancement

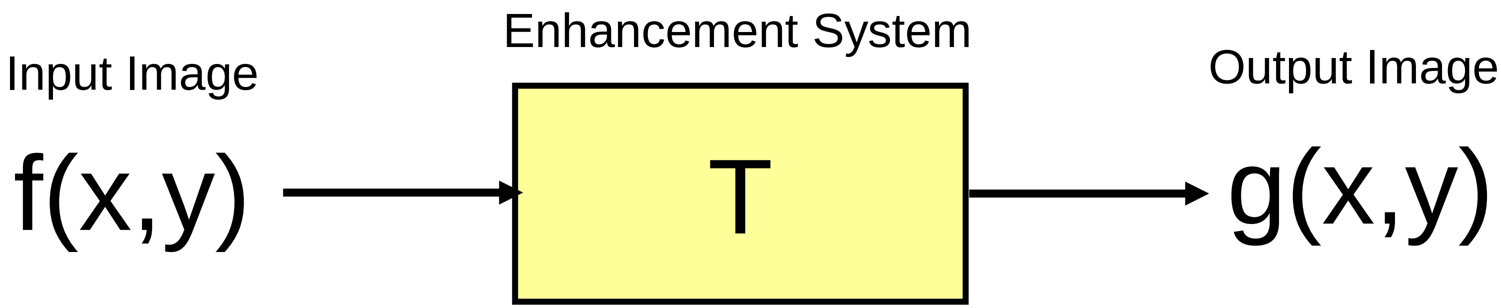
- Objective
  - Make an image more suitable for a specific application
  - Problem oriented
  - Subjective evaluation
    - viewer is the ultimate judge
  - No general theory
    - Many trials and errors are involved to tune several parameters

- Spatial domain approaches
  - Direct manipulation of pixels in an image

- Frequency domain approaches
  - Modification of the Fourier transform of an image

- Combinations of both spatial and frequency approaches

Background



- Point processing

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

  - Output pixel value depends only on the input pixel value at the same location

- Filtering (mask processing)

$$g(x,y) = T[\ \{f(u,v): (u,v) \in N(x,y)\}\ ]$$

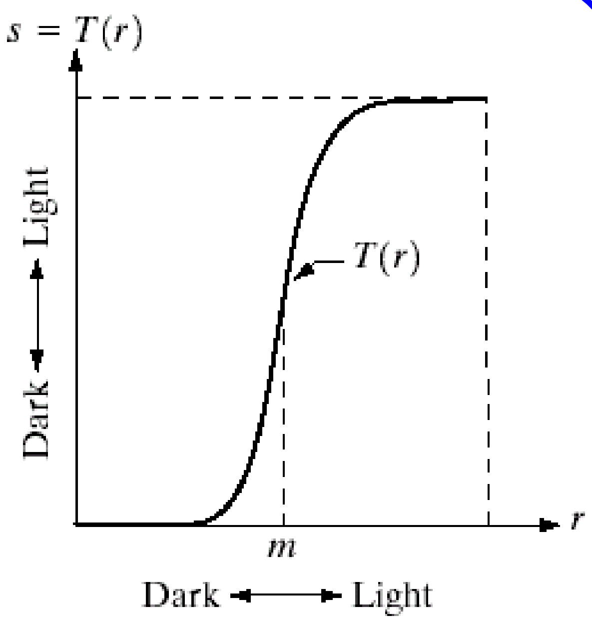
  - Output pixel value at (x,y) is determined by the values of the input pixels within the neighborhood of (x,y)
  - e.g. $g(x,y) = (f(x,y)+f(x+1,y))/2$

Point Processing

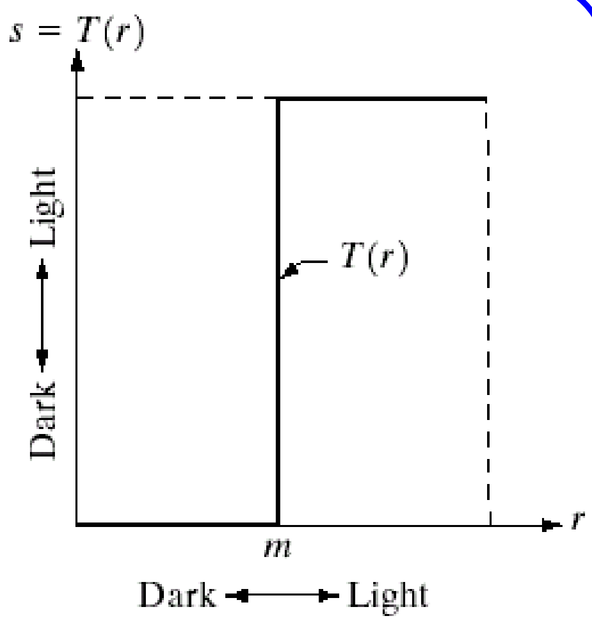
- $g(x,y) = T[\ f(x,y)\ ]$
- The enhancement system is fully described by

$$s = T(r)$$

where $s = g(x,y)$ and $r = f(x,y)$

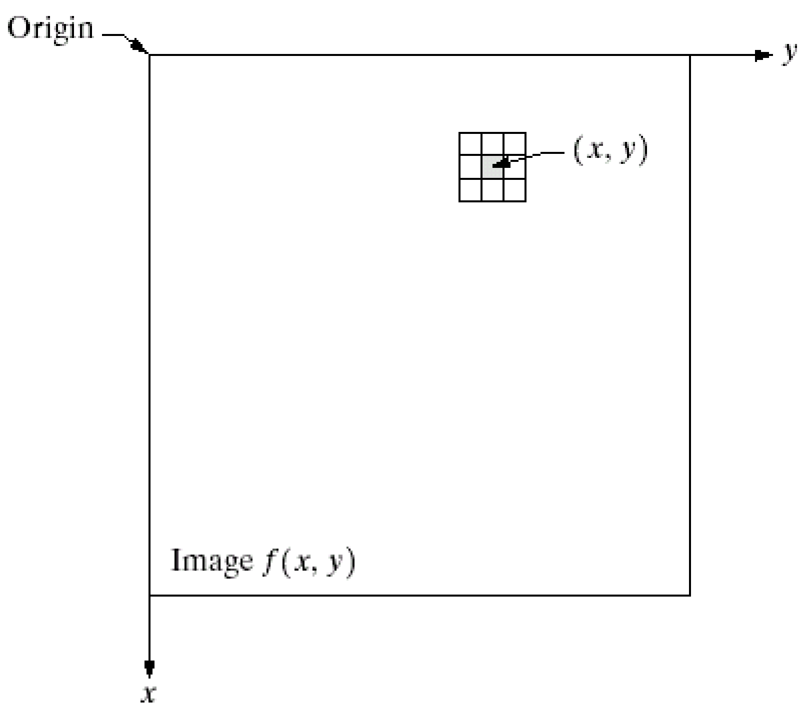


Contrast stretching for medium gray levels



Gray-to-binary image converter

Filtering (Mask Processing)



- Mask is moved from pixel to pixel
- At each location, the mask values are multiplied by the corresponding pixel values, and then summed up
- For example,

0	4	0
2	1	3
0	5	0

g(x,y) = f(x,y)
+ 4f(x-1,y) + 5f(x+1,y)
+ 2f(x,y-1) + 3f(x,y+1)

- This is equivalent to the convolution of f(x,y) with a filter with the impulse response

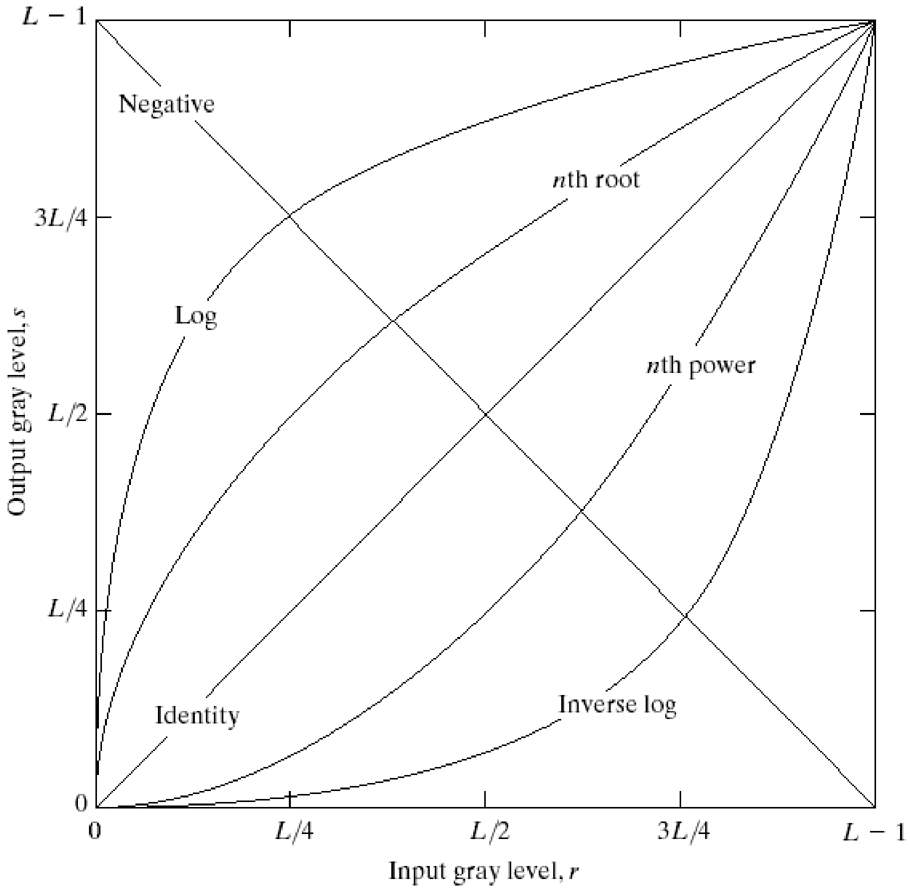
0	5	0
3	1	2
0	4	0

Basic Gray Level Transformations

- s = T(r)
- Lookup table implementation
  - e.g. negative transform

```
unsigned char T[255];
T[0] = 255;
T[1] = 254;
T[2] = 253;
…
T[254] = 1;
T[255] = 0;

for(x=0; x<width; x++)
for(y=0; y<height; y++)
     output.pixel(x,y) =
          T[input.pixel(x,y)];
```



Negative Transform

- $s = L - 1 - r$
  - L: # of gray levels (typically 256)

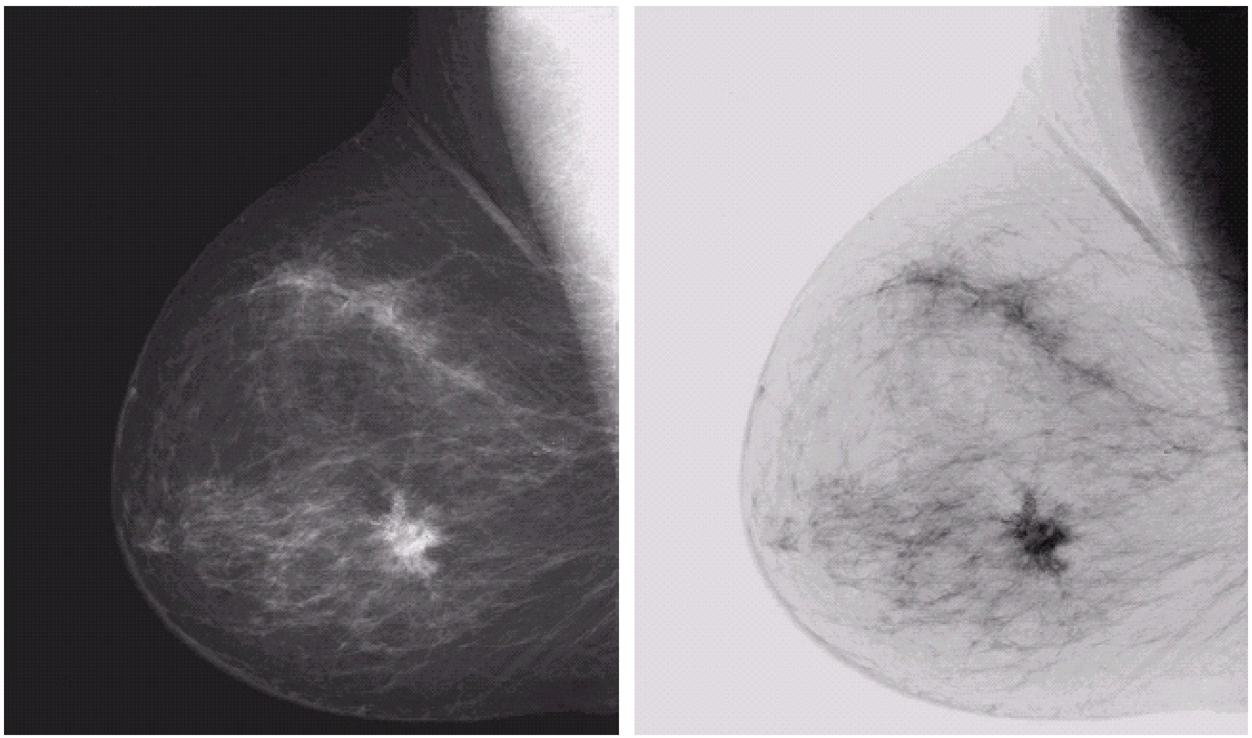
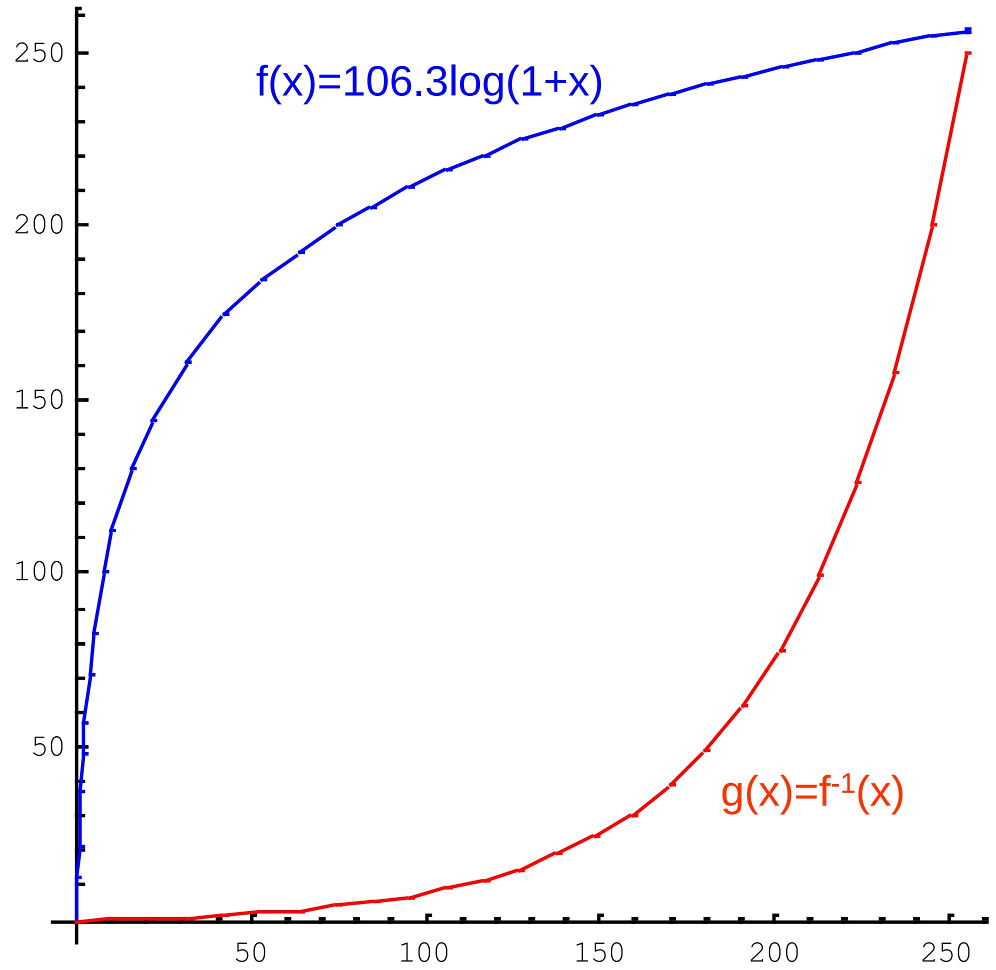


FIGURE 3.4 (a) Original digital mammogram. (b) Negative image obtained using the negative transformation in Eq. (3.2-1). (Courtesy of G.E. Medical Systems.)

Log Transforms

- $s = c \log(1+r)$
  - e.g. c=106.3
    [0,255] →[0,255]
  - enhance contrast on dark regions
    - expand dark pixel values
  - worsen contrast on bright regions
    - compress bright pixel values

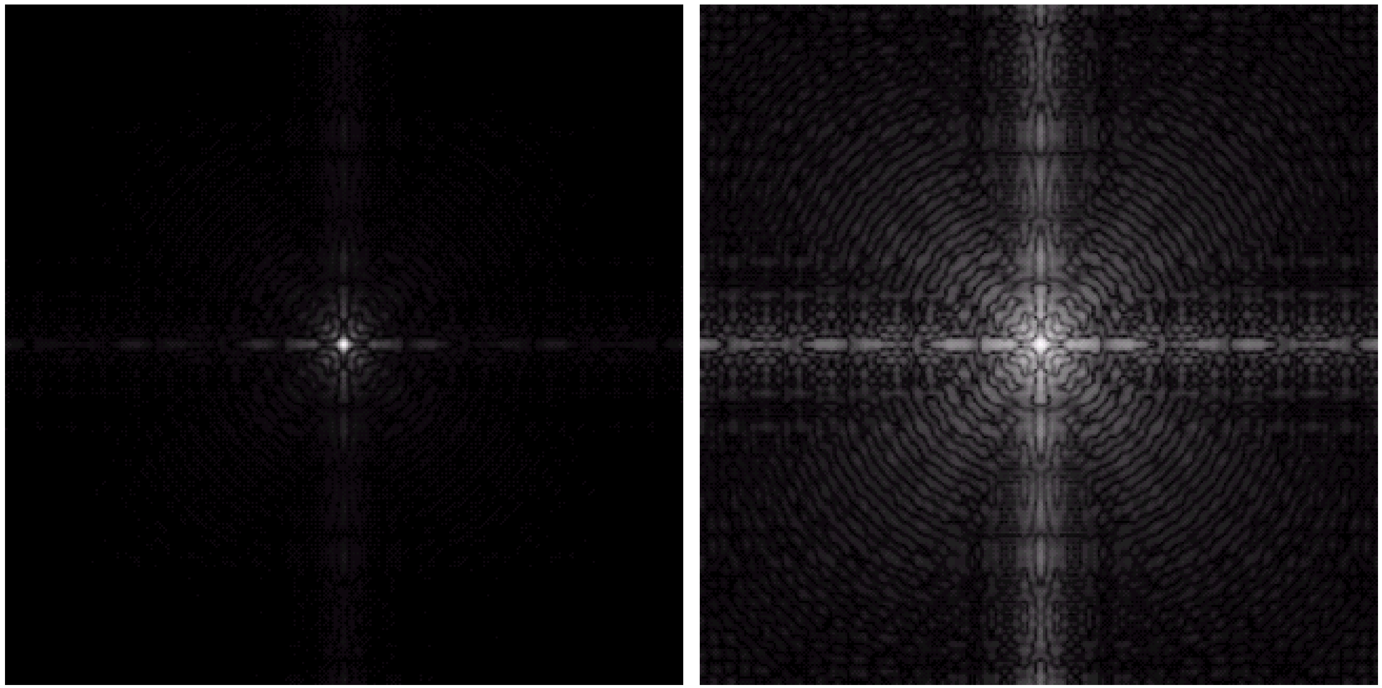
- Inverse log (exponential) transform
  - compress dark pixel values and expand bright pixel values



Log Transform of Fourier Transform

a b

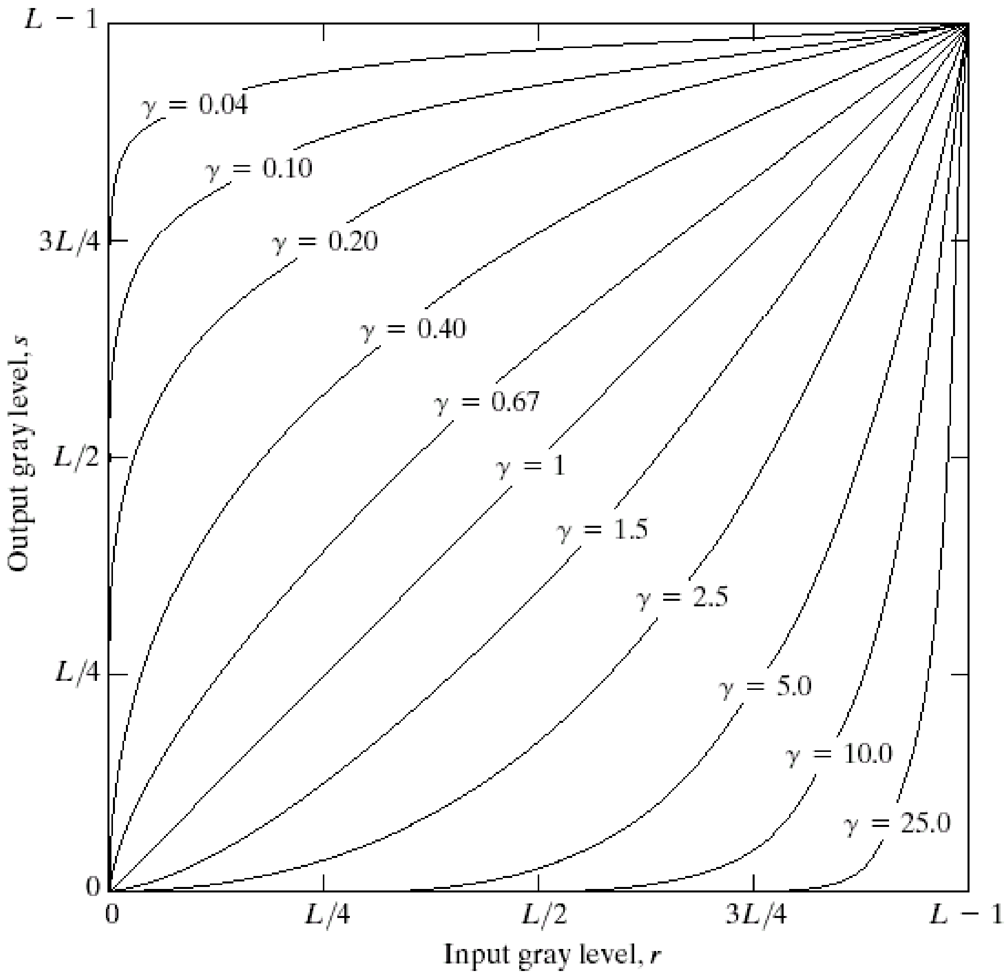
FIGURE 3.5 (a) Fourier spectrum. (b) Result of applying the log transformation given in Eq. (3.2-2) with $c = 1$.



- Fourier spectrum with values $r \in [0, 1.5\text{x}10^6]$
- After log transform $s = \log(1+r)$, $s \in [0, 6.2]$
- After linear scaling $s' = (255/6.2)s$, $s' \in [0, 255]$

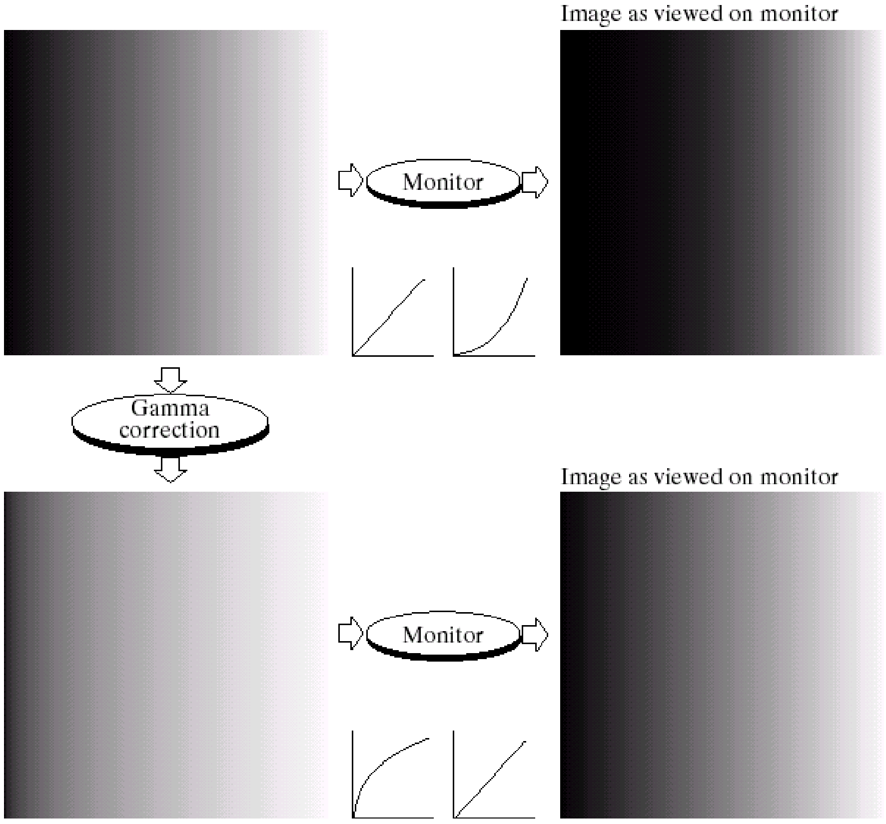
Power-Law Transforms

- $s = c\, r^{\gamma}$
  - $c = 255^{1-\gamma}$:
    [0,255] →[0,255]
- $\gamma<1$:
  - expand dark levels and compress bright levels
- $\gamma>1$:
  - expand bright levels and compress dark levels
- Varying γ controls the amount of expansion and compression



Power-Law Transforms – Gamma Correction

- Non-linear characteristics of capture, printing and display devices
- CRT display
  - (intensity) = (voltage)$^{2.5}$
  - Gamma correction: $s = r^{1/2.5} = r^{0.4}$

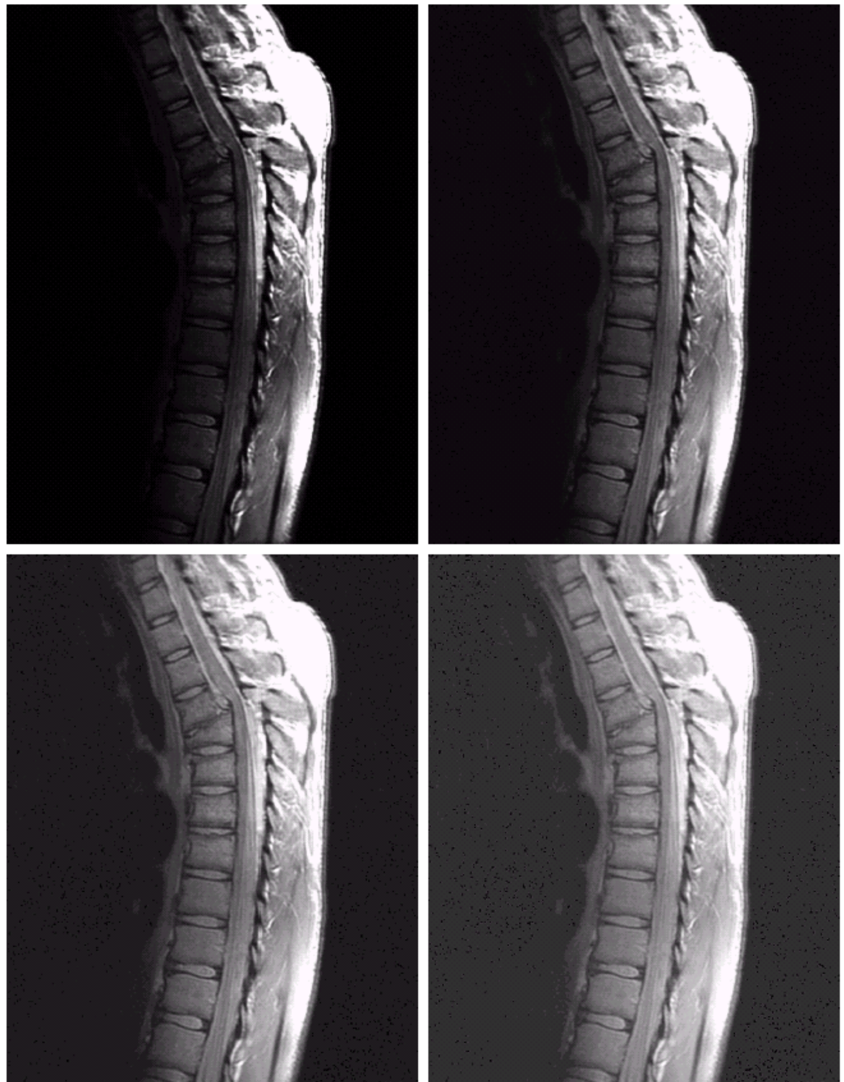


a b
c d

FIGURE 3.7
(a) Linear-wedge gray-scale image.
(b) Response of monitor to linear wedge.
(c) Gamma-corrected wedge.
(d) Output of monitor.

Power-Law Transforms – Contrast Enhancement

- $\gamma<1$: Expand dark gray levels



$\gamma=0.6$

$\gamma=0.4$

$\gamma=0.3$

Power-Law Transforms – Contrast Enhancement

- γ>1: Expand bright gray levels

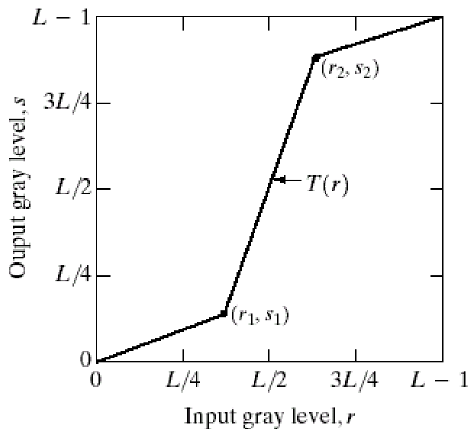
γ=3

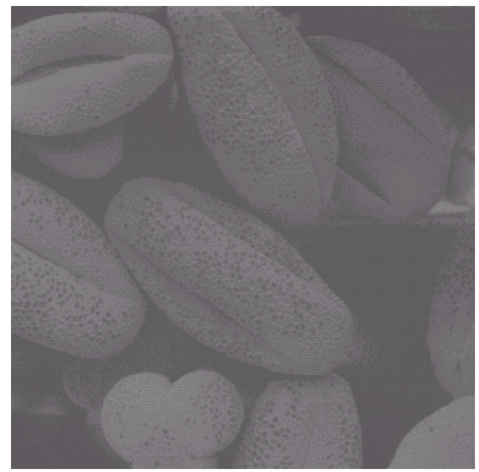
γ=4

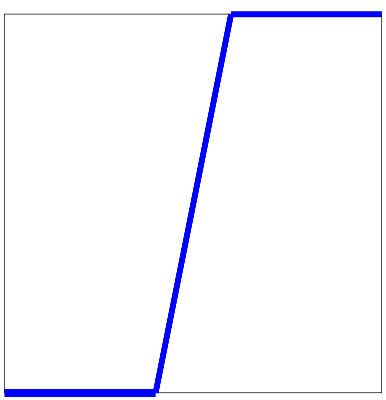
γ=5

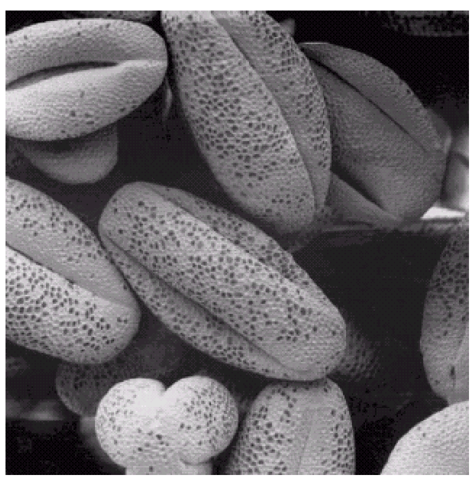
Piecewise-Linear Transformations

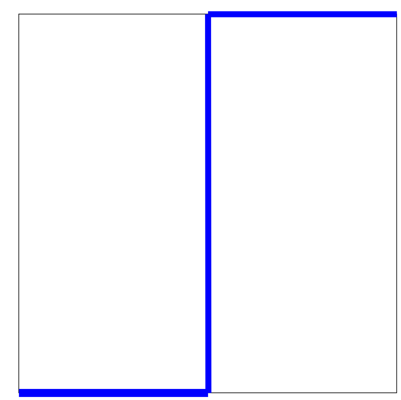
- Piecewise-Linear Functions
  - Freedom of design
  - More user input





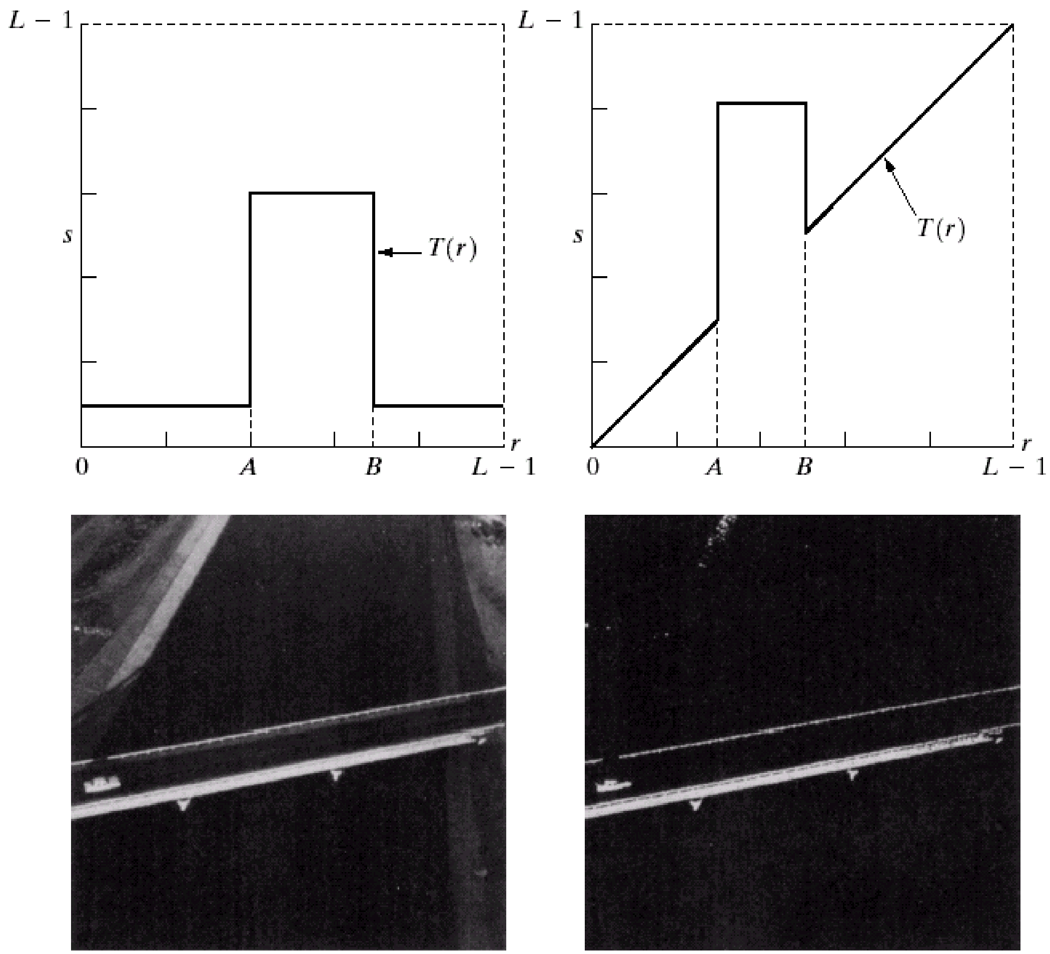






Piecewise-Linear Transformations

- Gray-level 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).

Piecewise-Linear Transformations

- Bit-plane slicing
  - Analysis of bit-plane data
    - Determining the resolution of quantizer
    - Bit-plane coding (JPEG2000)
  - Each bit-plane image is a binary image

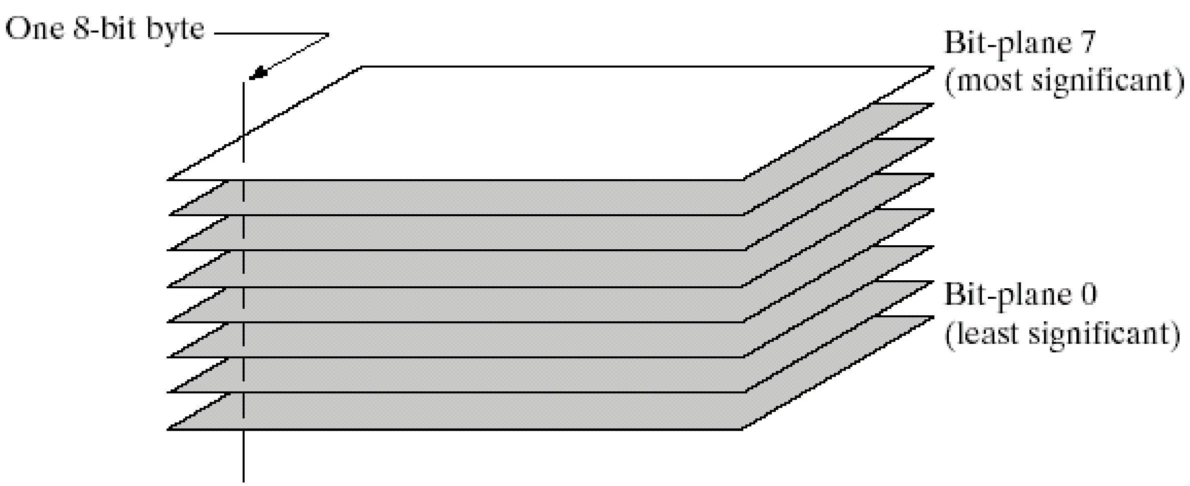
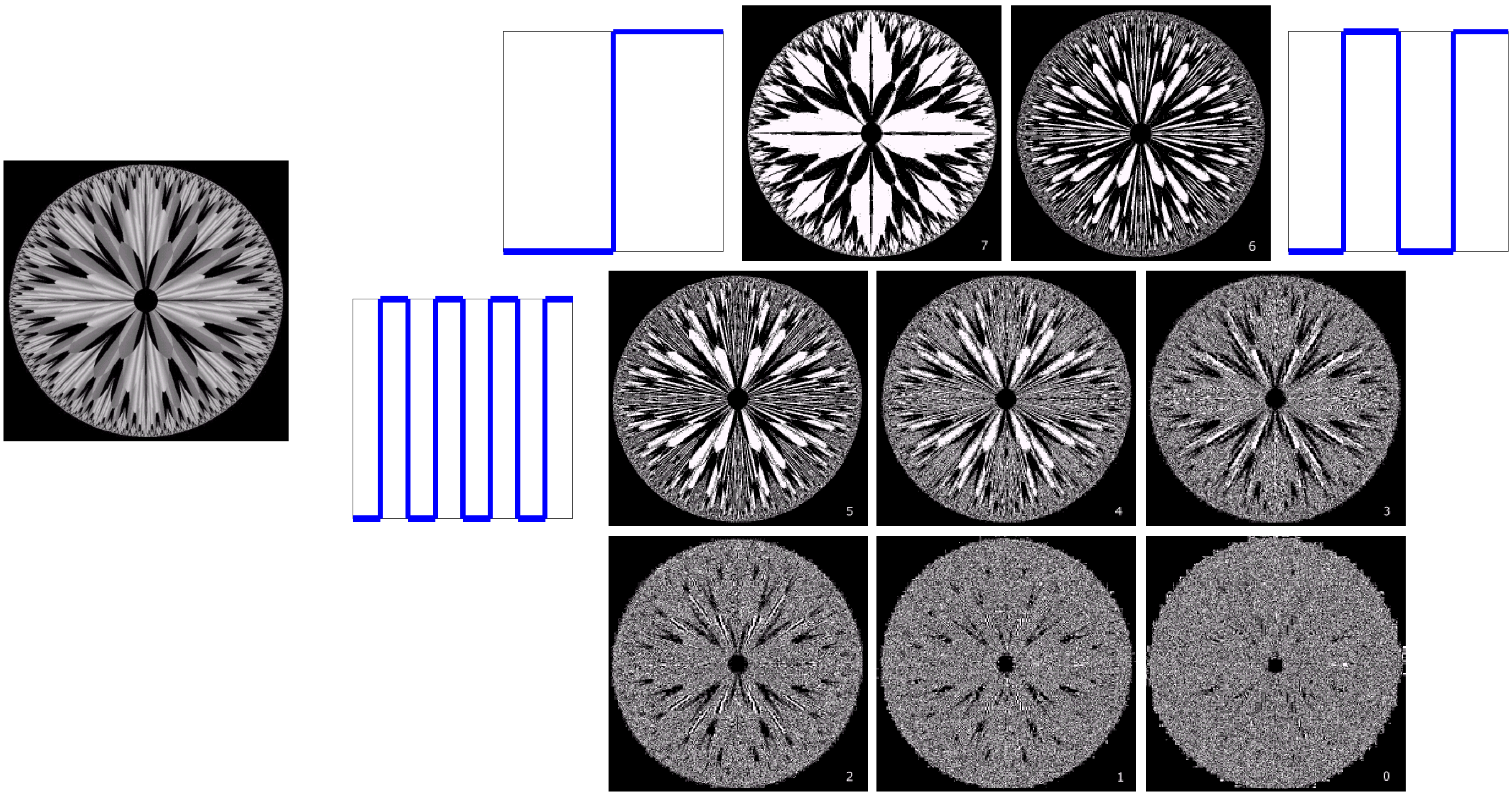


FIGURE 3.12 Bit-plane representation of an 8-bit image.

Piecewise-Linear Transformations

- Bit-plane slicing



Histogram Processing

- Histograms are the basis for numerous spatial domain image processing techniques
  - Rough estimate of probability distribution of gray levels
  - Simple to compute

- Histogram

$$h(r_k) = n_k$$

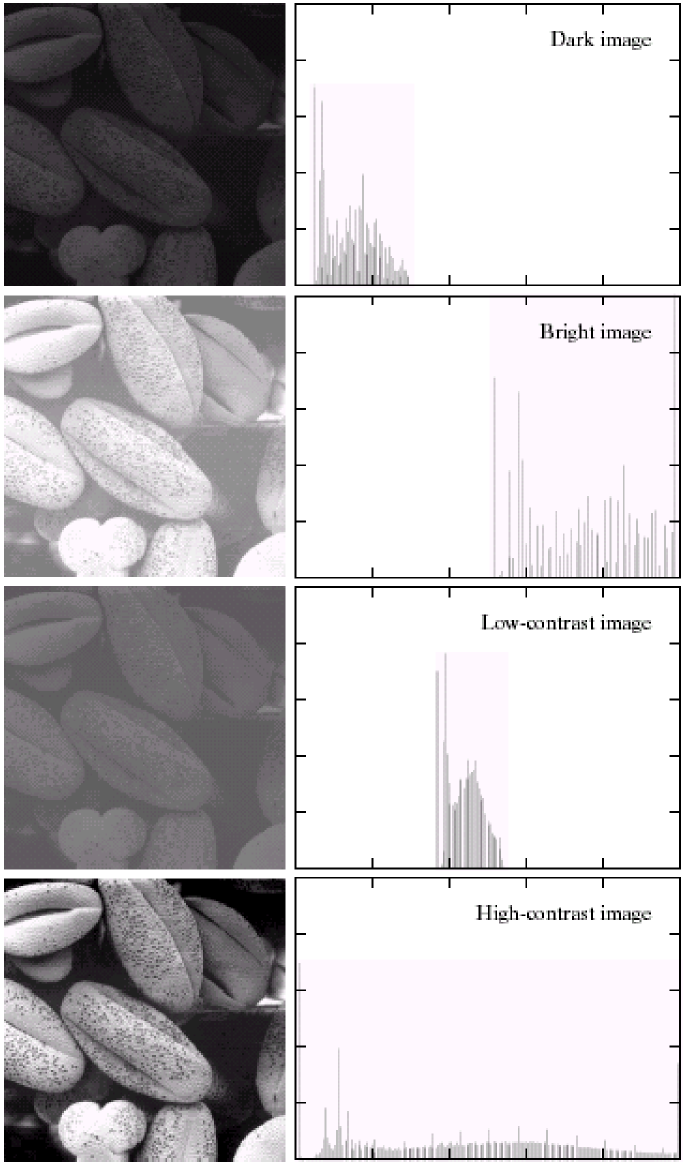
  - r_k: k-th gray level
  - n_k: the number of pixels in the image having gray level r_k

- Normalized histogram

$$p(r_k) = n_k/n$$

  - n: the total number of pixels
  - $\sum_k p(r_k) = 1$

Histogram - Example

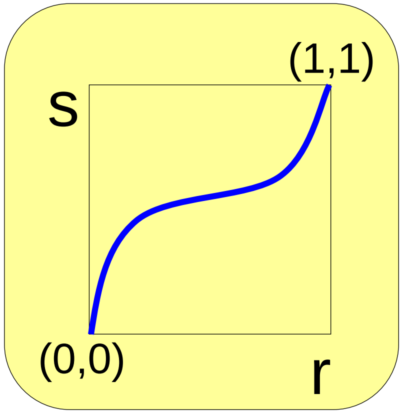


- In general, the uniform distribution of gray levels is desirable
  - high contrast
  - a great deal of details
  - high dynamic range

Histogram Equalization

- Enhance an input mage to have the gray level distribution, which is as uniform as possible

- Approach:
  - Continuous derivation → discrete approximation



- Problem definition
  - r : gray level of input image
    - Normalized to [0, 1]
    - Probability density function: $p_r(r)$
  - Monotonic increasing function: $s = T(r)$, $s \in [0,1]$
  - Goal is to find the function T, such that
    - $p_s(s) = 1$ for all $s \in [0,1]$
    - i.e. s is a uniform random variable

Histogram Equalization

- Function from random variable to random variable

$$s = T(r)$$

- Recall that

$$p_s(s)|ds| = p_r(r)|dr|, \text{ or } p_r(r) = p_s(s)|\frac{ds}{dr}|$$

- Therefore

$$p_r(r) \quad = \quad T'(r)$$

- We have

$$T(r) = \int_{-\infty}^{r} p_r(w)dw + c = \int_{0}^{r} p_r(w)dw$$

Histogram Equalization

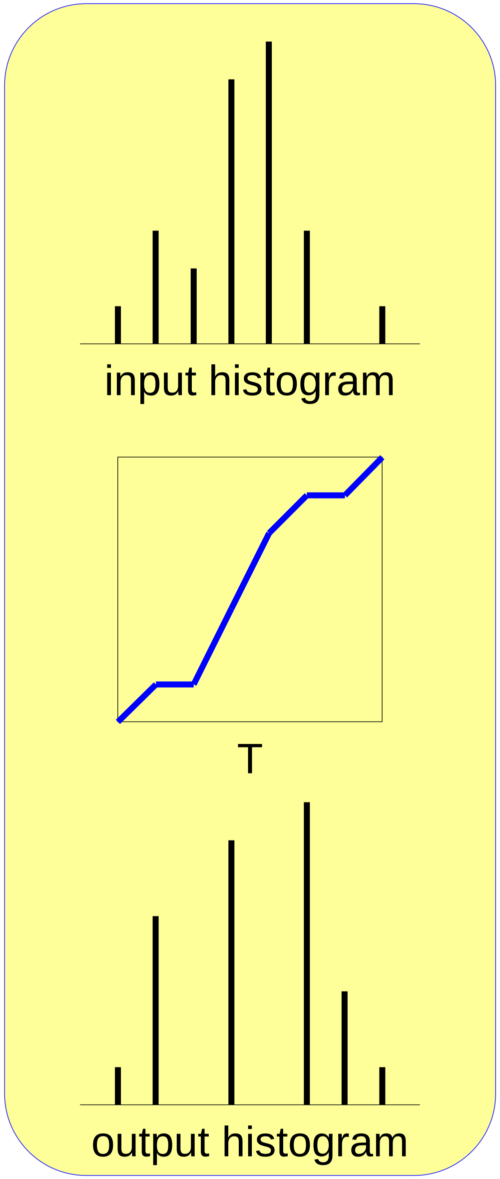
- Continuous case $$s = T(r) = \int_0^r p_r(w)dw$$
- Discrete approximation $$s_k = T(r_k) = \sum_{j=0}^{k} p_r(r_j)$$
- Example

input gray level k	0	1	2	3	4	5	6	7
normalized input r_k	0	1/7	2/7	3/7	4/7	5/7	6/7	1
histogram n_k	1	3	2	7	8	3	0	1
normalized histogram n_k/n	1/25	3/25	2/25	7/25	8/25	3/25	0	1/25
normalized output s_k	1/25	4/25	6/25	13/25	21/25	24/25	24/25	1
denormalized output $o_k = s_k \times 7$	7/25	28/25	42/25	91/25	147/25	168/25	168/25	7
output gray level floor(o_k)	0	1	1	3	5	6	6	7
m	0	1	2	3	4	5	6	7
output histogram n_m	1	5	0	7	0	8	3	1

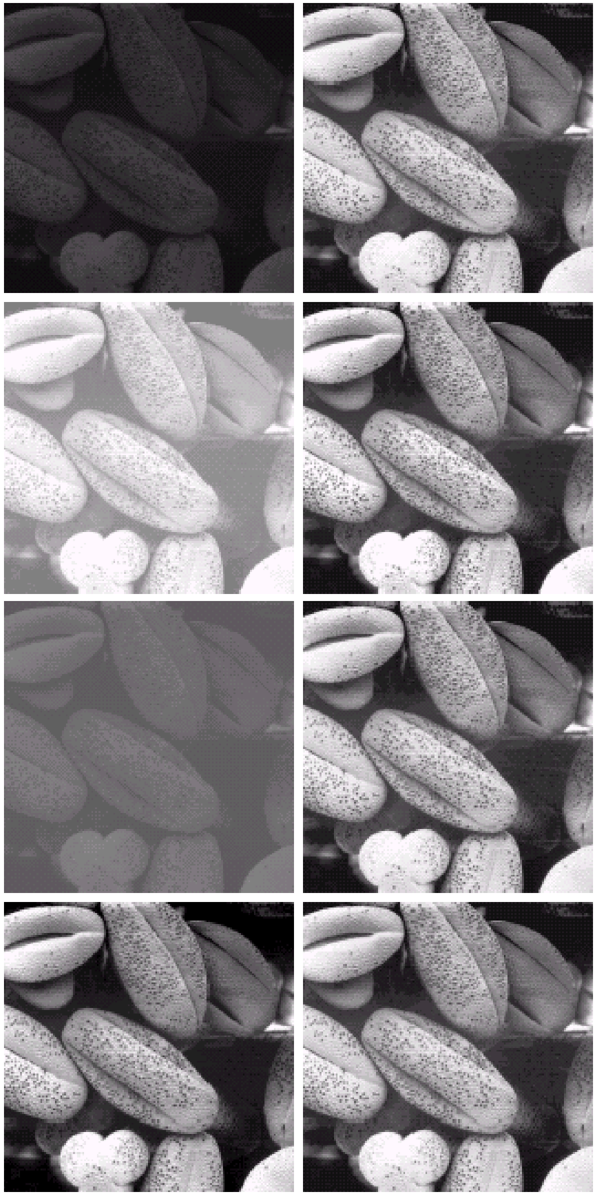
Histogram Equalization

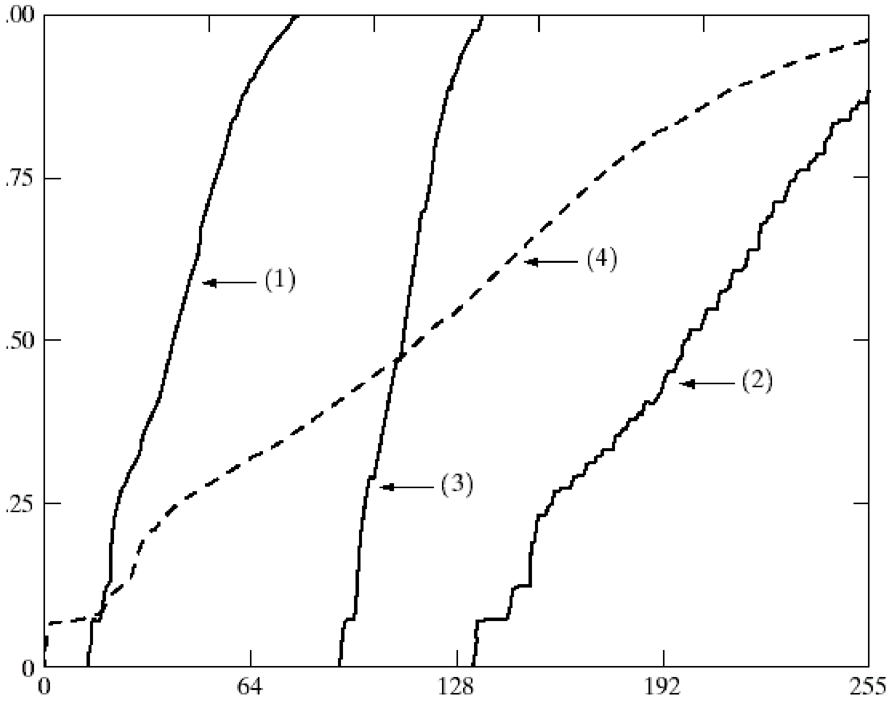
input gray level	0	1	2	3	4	5	6	7
output gray level	0	1	1	3	5	6	6	7
input histogram	1	3	2	7	8	3	0	1
output histogram	1	5	0	7	0	8	3	1

- Does not provide the exactly uniform output
  - Discrete approximation
- But, spread the histogram automatically



Histogram Equalization





Histogram Matching (Specification)

- Specify the shape of output histogram
- Continuous derivation
  - PDF of input: $p_r(r)$, $r \in [0, 1]$
  - Specified PDF of output: $p_z(z)$, $z \in [0, 1]$
  - Temporary variable: s and v

$$
\begin{aligned}
s = T(r) &= \int_0^r p_r(w)dw \\
v = G(z) &= \int_0^z p_z(w)dw
\end{aligned}
$$

  - Both s and v are uniform random variables on $[0, 1]$. Thus we have

$$T(r) = s = v = G(z)$$

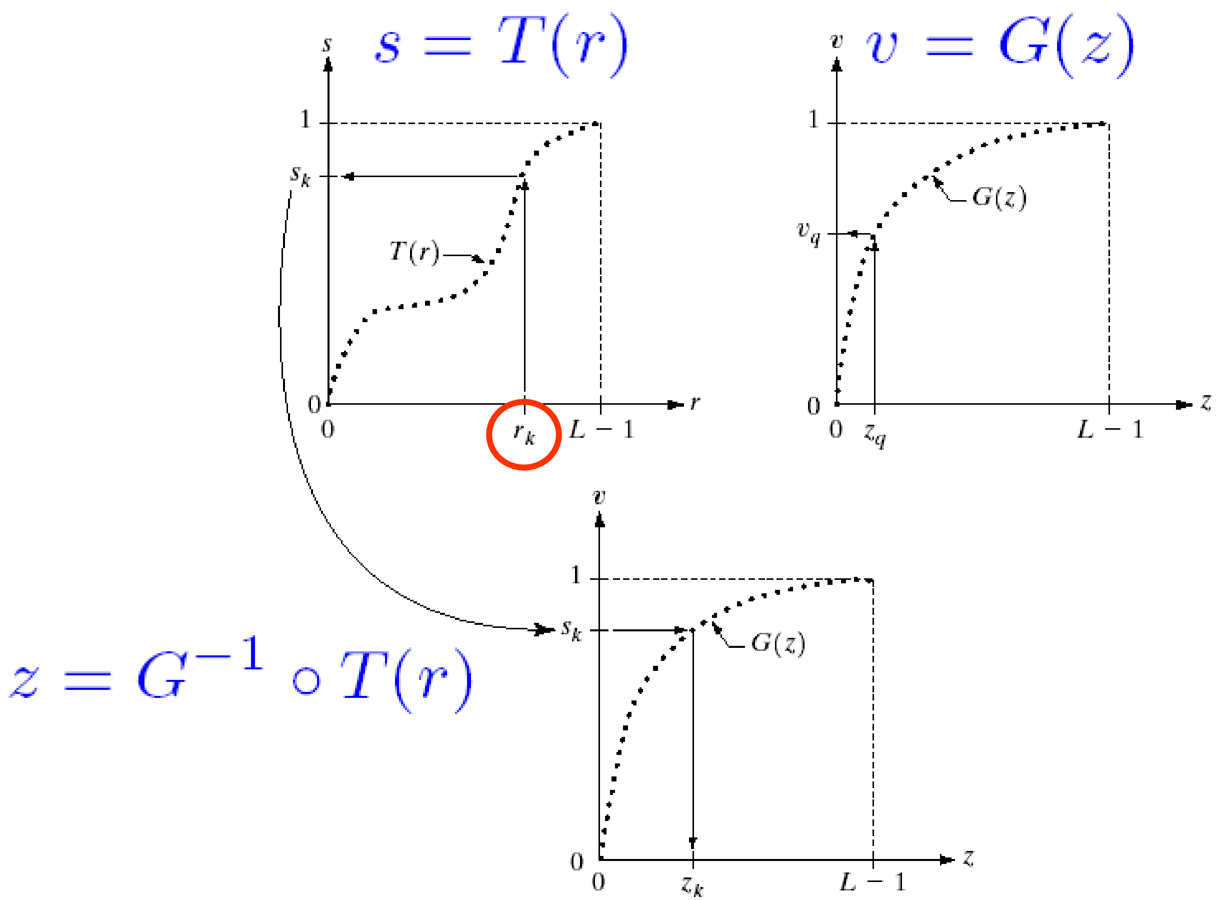
  - Therefore, the desired transform is given by

$$z = G^{-1} \circ T(r)$$

Histogram Matching (Specification)

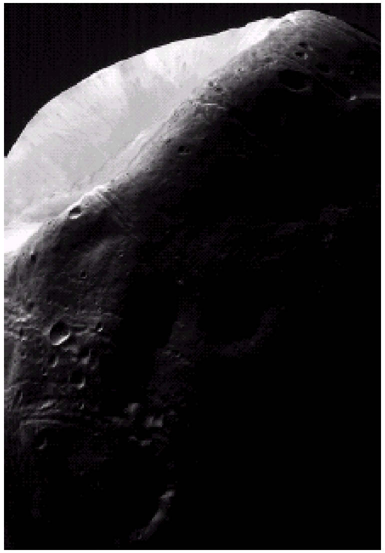
$$z = G^{-1} \circ T(r)$$

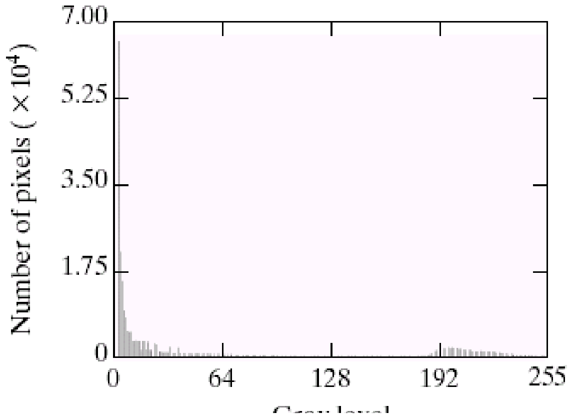
- Discrete Approximation

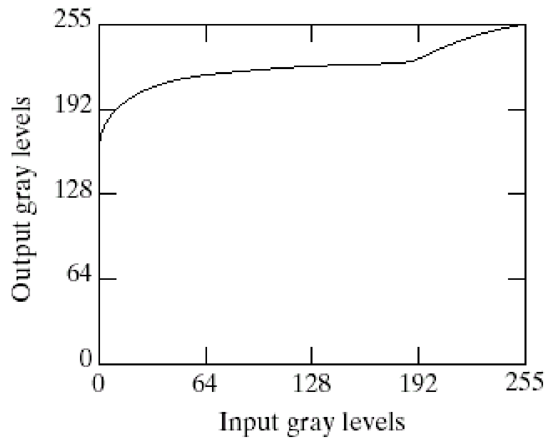


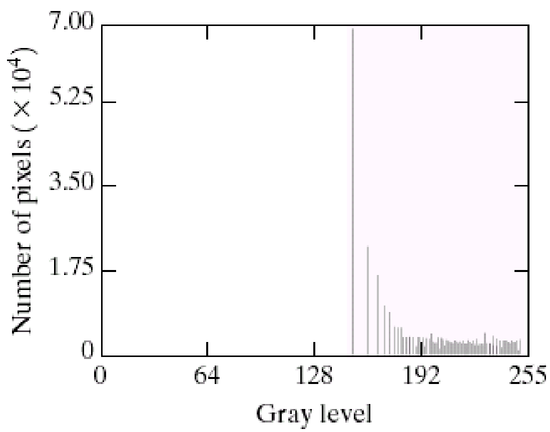
Histogram Matching (Specification)

- Failure of histogram equalization
  - two many zero pixels
  - discrete approximation error
  - Little contrast enhancement

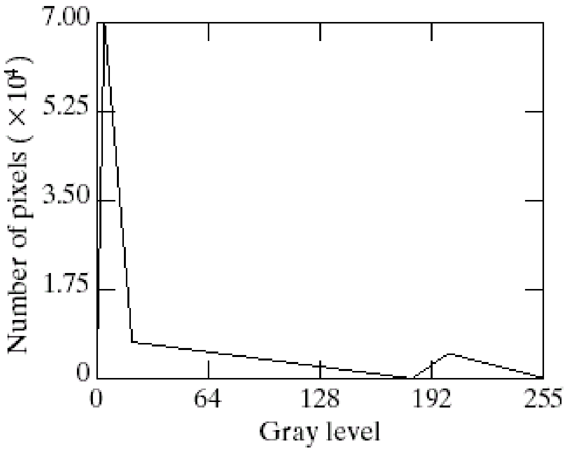


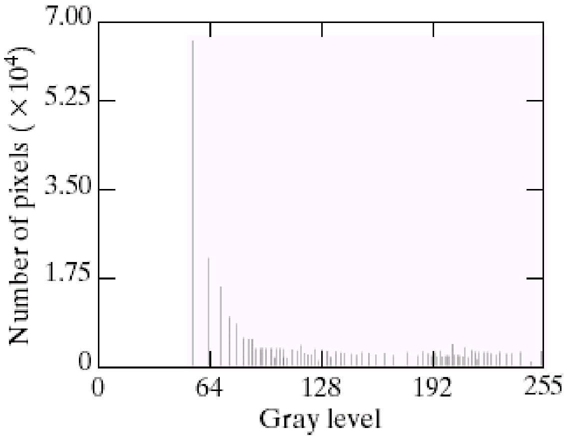






Histogram Matching (Specification)

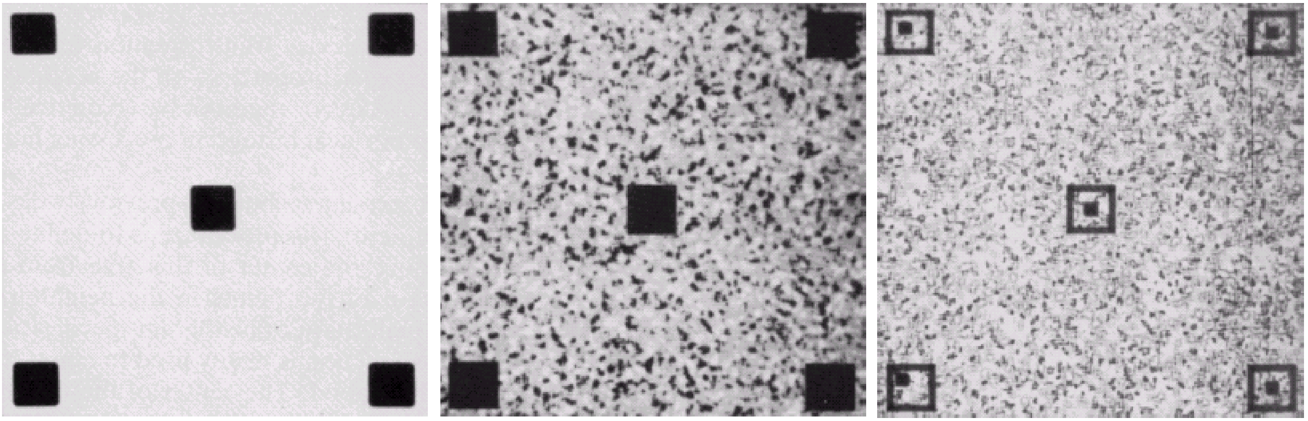




- Better result of histogram matching
  - Specifying target histogram
    - Many trials and errors
- Just illustration
  - Power transform with gamma < 1 may provide a similar or better result with less human inputs
- Useful
  - If we know general characteristics of input images and know what the target histogram should be by experience
  - The same target histogram is reused for many images

Local Enhancement Using Histogram Equalization

- Global equalization may fail to enhance details over small areas in an image
  - The number of pixels in these areas may be too small to have effect on the global histogram
- Local Processing
  1. For each pixel (x,y), define a neighborhood (e.g. 7x7 square)
  2. Compute the local histogram and find the equalizing transform
  3. Change the value of pixel (x,y), and go to step 1



a b c

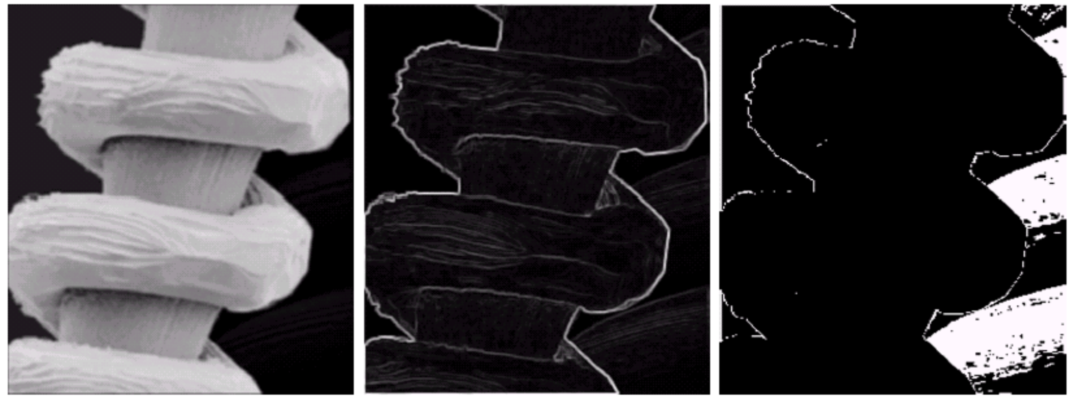
FIGURE 3.23 (a) Original image. (b) Result of global histogram equalization. (c) Result of local histogram equalization using a 7×7 neighborhood about each pixel.

Image Enhancement Using Histogram Statistics

- Global statistics
  - Mean: $m = \sum_{i=0}^{L-1} r_i p(r_i)$
  - Variance: $\sigma^2 = \sum_{i=0}^{L-1} (r_i - m)^2 p(r_i)$
- After defining a neighborhood S_{xy}, we can also define local mean and variance at each pixel (x, y)
  - Local mean: $m_{S_{xy}} = \sum_{(s,t) \in S_{xy}} r_{s,t} p(r_{s,t})$
  - Variance: $\sigma^2_{S_{xy}} = \sum_{(s,t) \in S_{xy}} (r_{s,t} - m_{S_{xy}})^2 p(r_{s,t})$

Image Enhancement Using Histogram Statistics

$m_{S_{xy}}$ $\sigma^2_{S_{xy}}$



3x3 local neighbor is used

$$g(x,y) = \begin{cases} 4 \cdot f(x,y) & \text{if } m_{S_{xy}} \le 0.4m \text{ and } 0.02\sigma^2 \le \sigma^2_{S_{xy}} \le 0.4\sigma^2 \\ f(x,y) & \text{otherwise} \end{cases}$$

Enhancement Using Arithmetic/Logic Operations

- Logic operations for masking

AND: p&255=p,   p&0=0

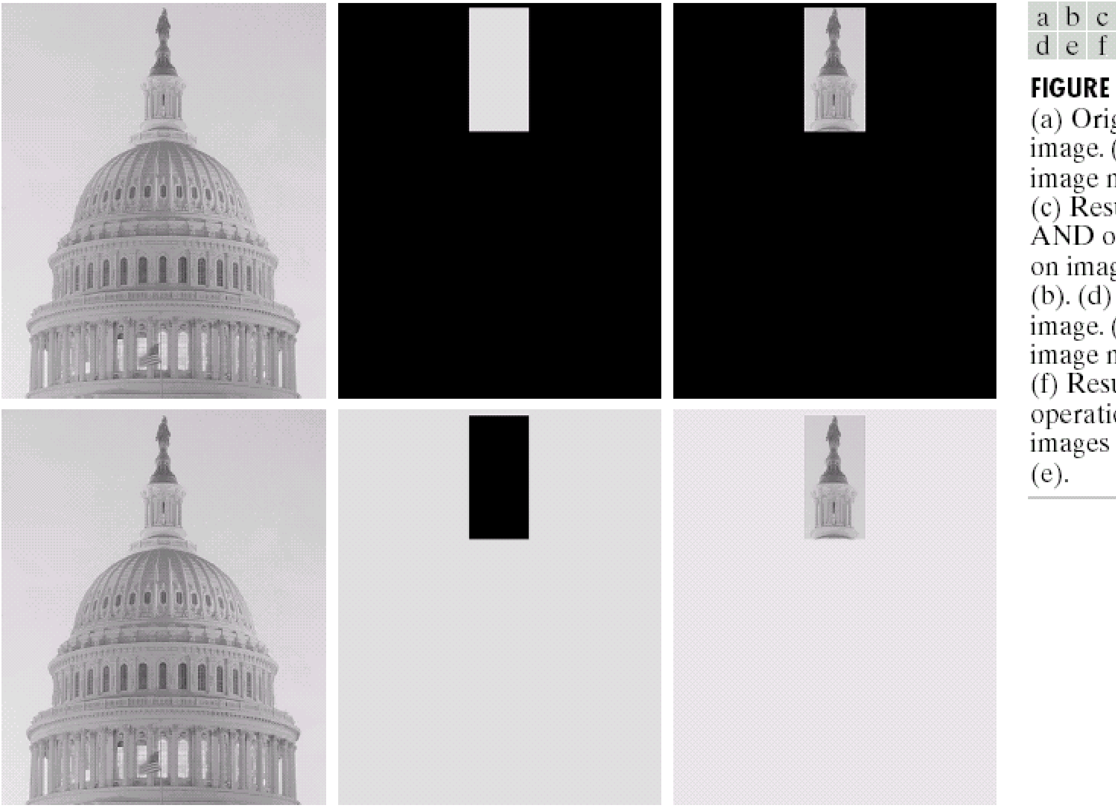


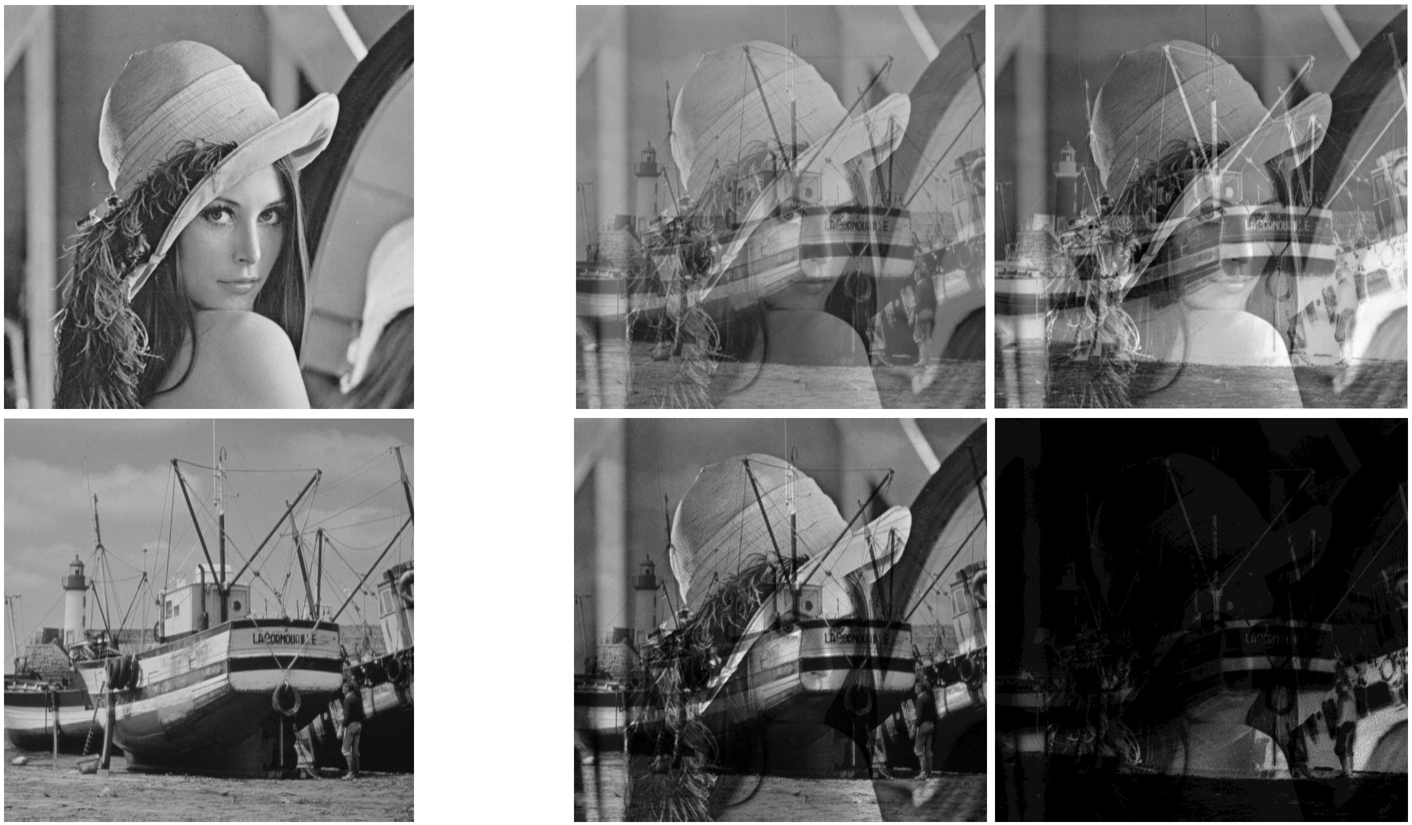
FIGURE 3.27 (a) Original image. (b) AND image mask. (c) Result of the AND operation on images (a) and (b). (d) Original image. (e) OR image mask. (f) Result of operation OR on images (d) and (e).

OR: p|255=255,   p|0=p

Enhancement Using Arithmetic/Logic Operations

- Pixelwise arithmetic operations

A | A+B | A-B



B | A B | A/B

- Normalization: $p' = (p-p_{min})/(p_{max}-p_{min})$ 255

Image Subtraction

- Enhancement of image differences

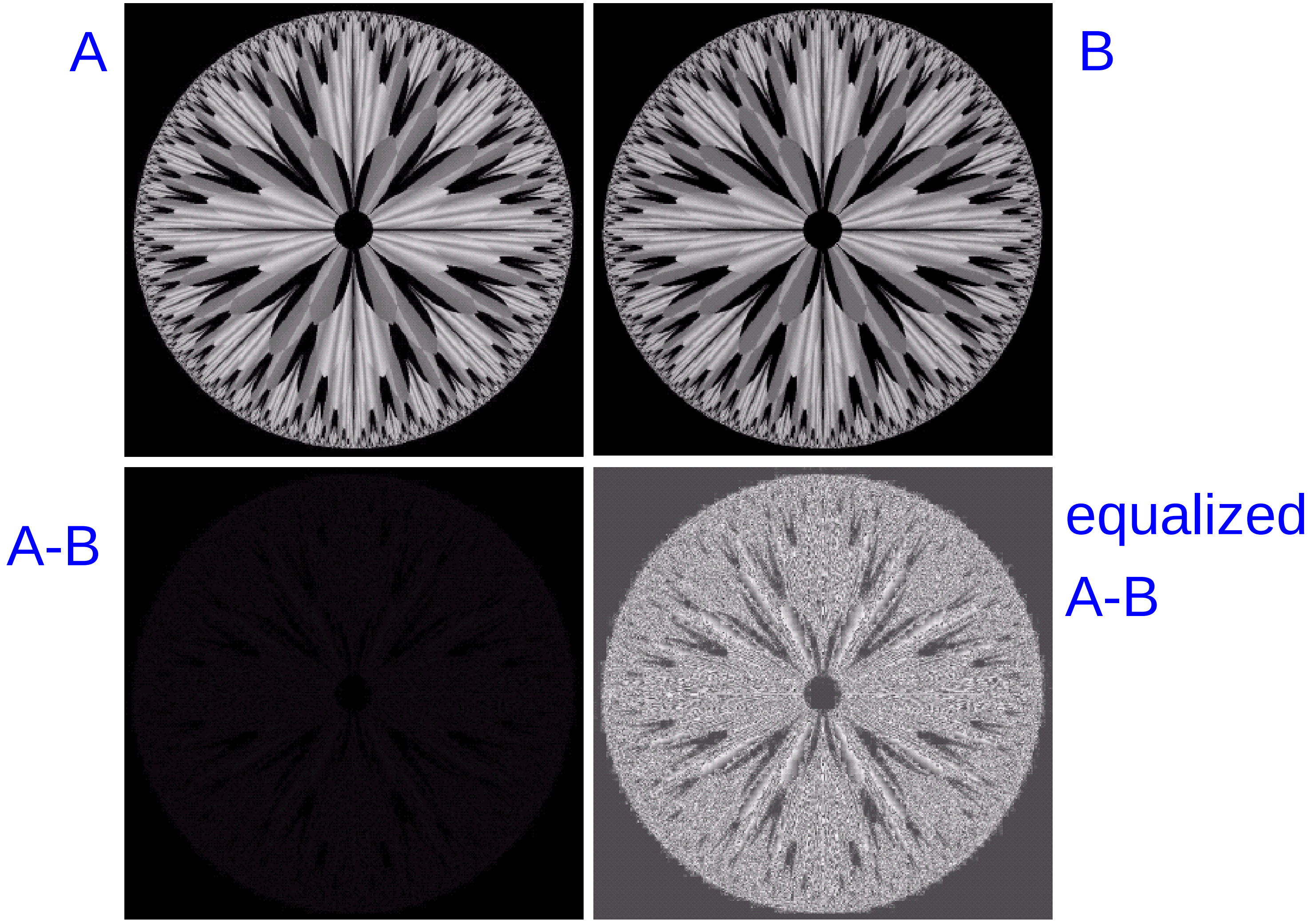


Image Subtraction

- Mask mode radiography

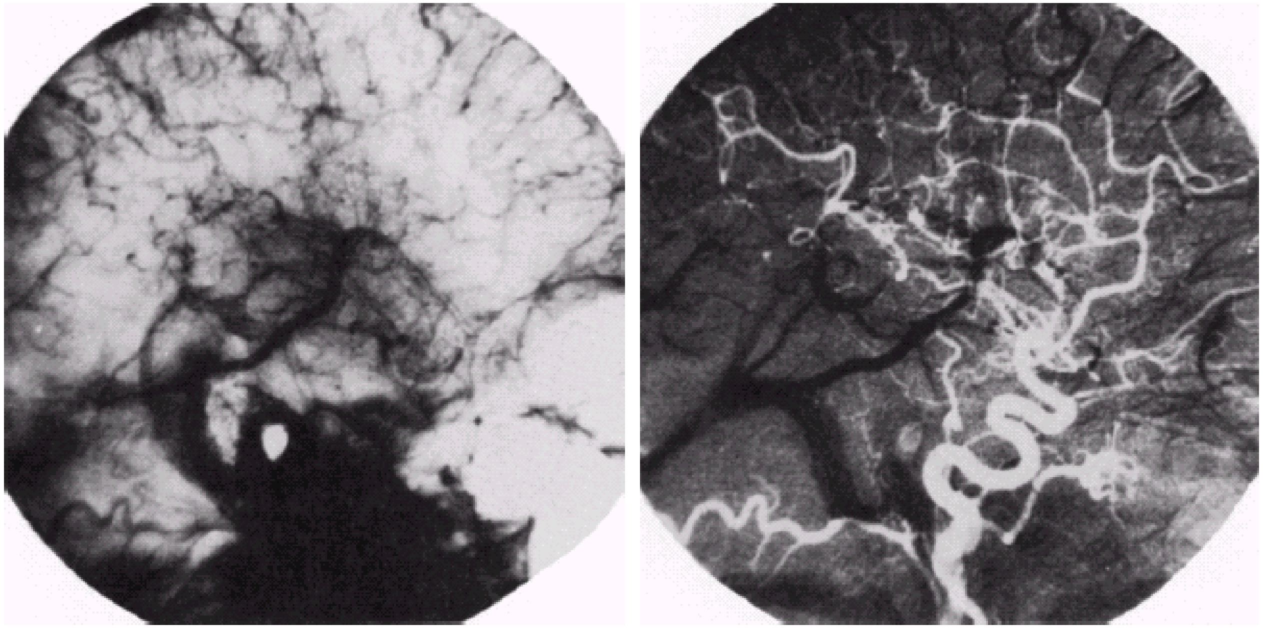


FIGURE 3.29 Enhancement by image subtraction. (a) Mask image. (b) An image (taken after injection of a contrast medium into the bloodstream) with mask subtracted out.

- Tracking moving vehicles
  - output = input – background = vehicles

Image Averaging

- Noise reduction
- Assumption:
  - Several pictures of the same object with different noises are available
  - Noises are uncorrelated with mean 0 and variance σ^2

$$\begin{aligned} g_1(x,y) &= f(x,y) + \eta_1(x,y) \\ &\vdots \\ g_K(x,y) &= f(x,y) + \eta_K(x,y) \end{aligned}$$

- Averaging

$$\begin{aligned} \bar{g}(x,y) &= \frac{1}{K}\sum_{i=1}^{K} g_i(x,y) = f(x,y) + \frac{1}{K}\sum_{i=1}^{K}\eta_i(x,y) = f(x,y) + \bar{\eta}(x,y) \\ \bar{\sigma}^2 &= E[\bar{\eta}^2(x,y)] = \frac{1}{K^2}E[\sum_{i=1}^{K}\sum_{j=1}^{K}\eta_i(x,y)\eta_j(x,y)] = \frac{1}{K}\sigma^2 \end{aligned}$$

- The noise power of the averaged image is K times smaller than each individual image.

Image Averaging

Original

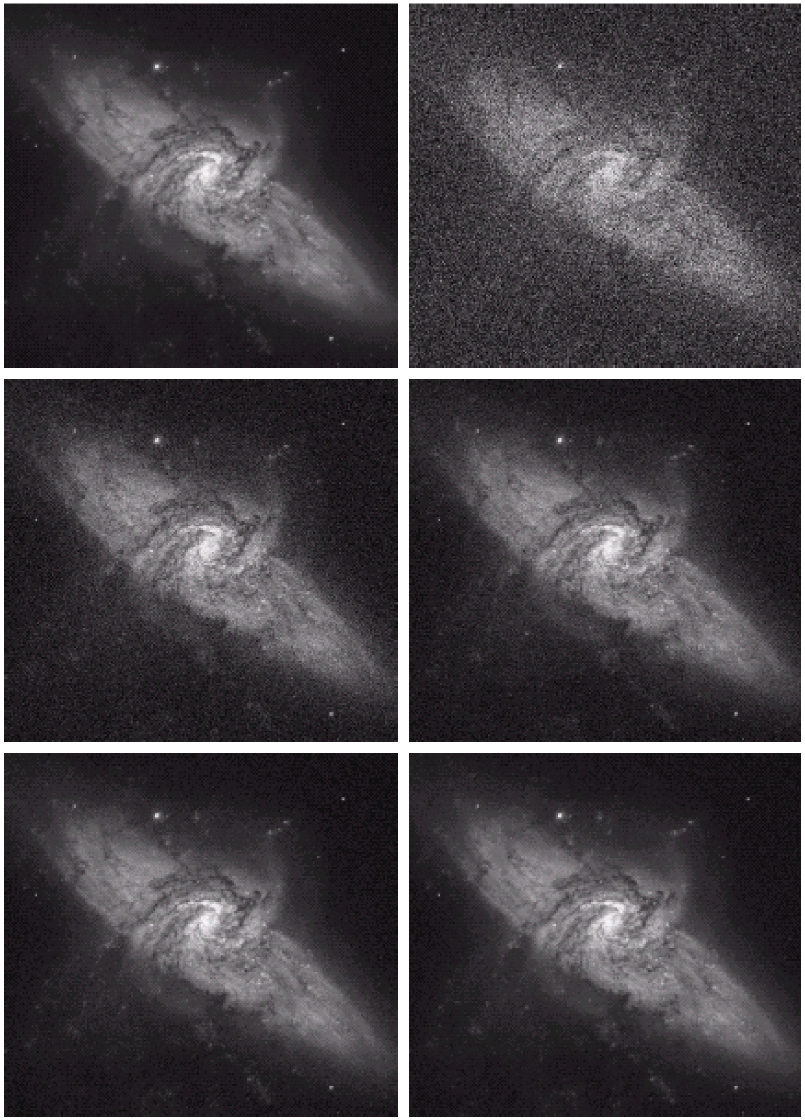
K=1

K=8

K=16

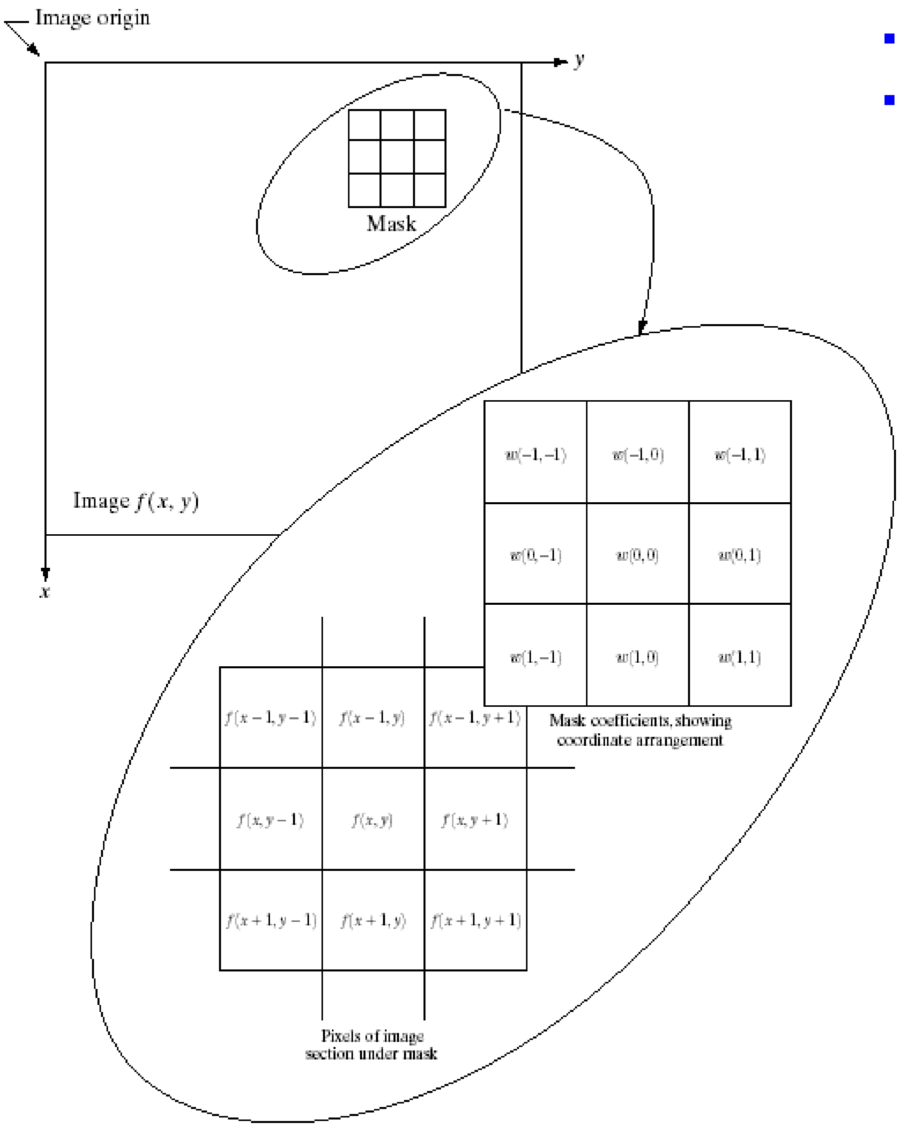
K=32

K=64



- Astronomical observation
- observing the same scene over long periods of time
- continuous integration of signal
- equivalent to summation or averaging

Masking (Spatial Filtering)



- Mask is moved from pixel to pixel
- At each location, the mask coefficients are multiplied by the corresponding pixel values, and then summed up

$$
\begin{aligned}
g(x,y) = & \; w(-1,-1)f(x-1,y-1) \\
& + w(-1,0)f(x-1,y) + \ldots \\
& + w(1,1)f(x+1,y+1)
\end{aligned}
$$

Masking with

a	b	c
d	e	f
g	h	i

=

Convolving with

i	h	g
f	e	d
c	b	a

Masking (Spatial Filtering)

- Masking with a mask w of size $(2a+1)\times(2b+1)$

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

- Convolving with a filter h of size $(2a+1)\times(2b+1)$

$$g'(x,y)=\sum_{s=-a}^{a}\sum_{t=-b}^{b} h(s,t)f(x-s,y-t)$$

- Note that $g(x,y)=g'(x,y)$ if $w(s,t)=h(-s,-t)$
- For masking, we use the following notation also

$$R=\sum_{i=1}^{k} w_i z_i = w_1 z_1 + w_2 z_2 + \ldots + w_k z_k$$

where w_i's are masking coefficients and z_i's are pixel values.

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

Masking (Spatial Filtering)

- Boundary problem
  1. Limit the excursion of the center of the mask, so that the mask is fully contained within the image
     - Output image is smaller than input image

0	0	0	0	0
0	0	0	0	0
0	0	a	b	c
0	0	d	e	f
0	0	g	h	i

  2. Extrapolate the input image sufficiently, so that the mask can be applied near the boundaries also.
     - Zero padding
     - Repetition
     - Mirroring
     - etc

a	a	a	b	c
a	a	a	b	c
a	a	a	b	c
d	d	d	e	f
g	g	g	h	i

a	a	d	e	f
a	a	a	b	c
b	a	a	b	c
e	d	d	e	f
h	g	g	h	i

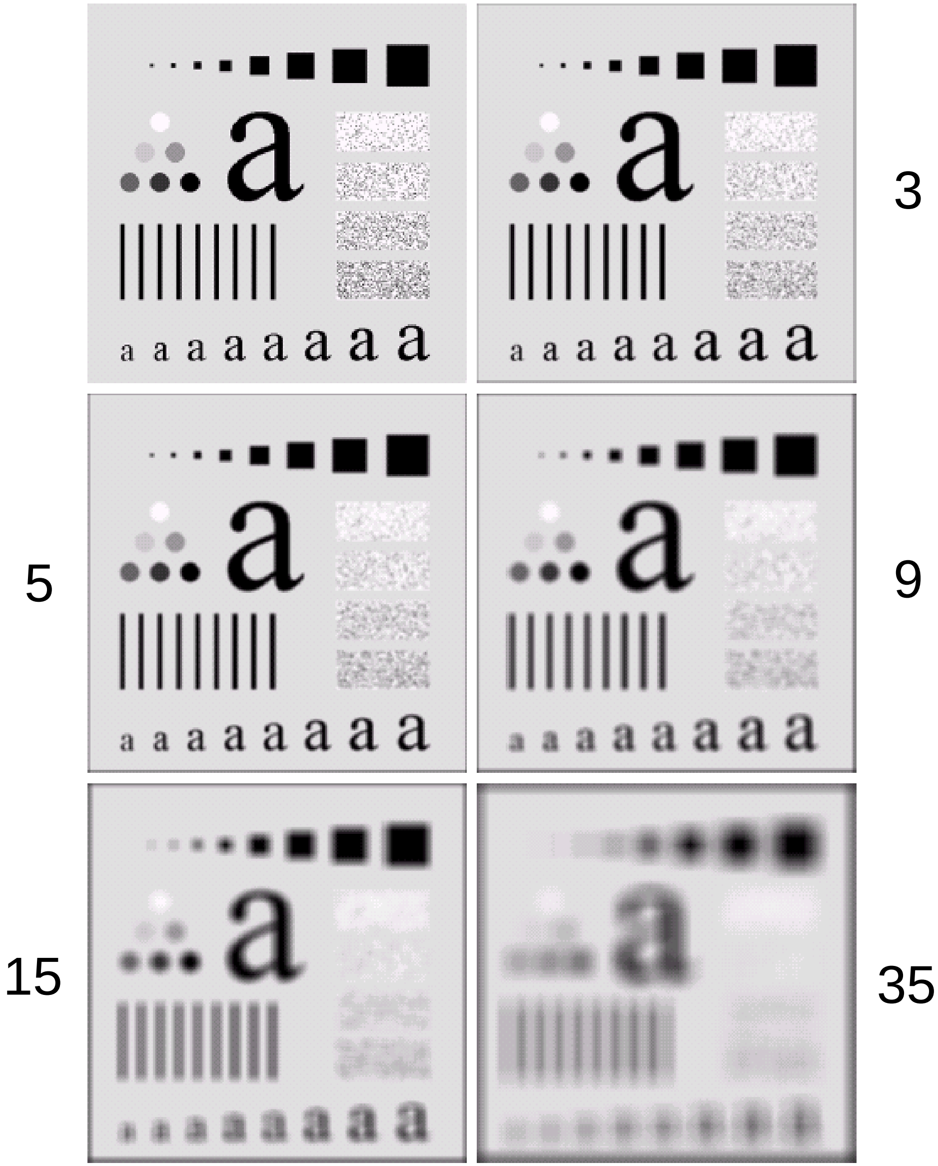
Smoothing Spatial Filters

- Averaging filter and weighted averaging filter

$$\frac{1}{9} \times \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \qquad \frac{1}{16} \times \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

- Blends with adjacent pixel values
- Blurring
  - Removal of small details before large object extraction
  - Bridging of small gaps in lines or curves
  - Reduction of sharp transitions in gray levels
    - Advantage: noise reduction
    - Disadvantage: edge blurring

Smoothing Spatial Filters



- Losing edges
- Reducing noises
- Removing small objects

Smoothing Spatial Filters

- Finding objects of interest

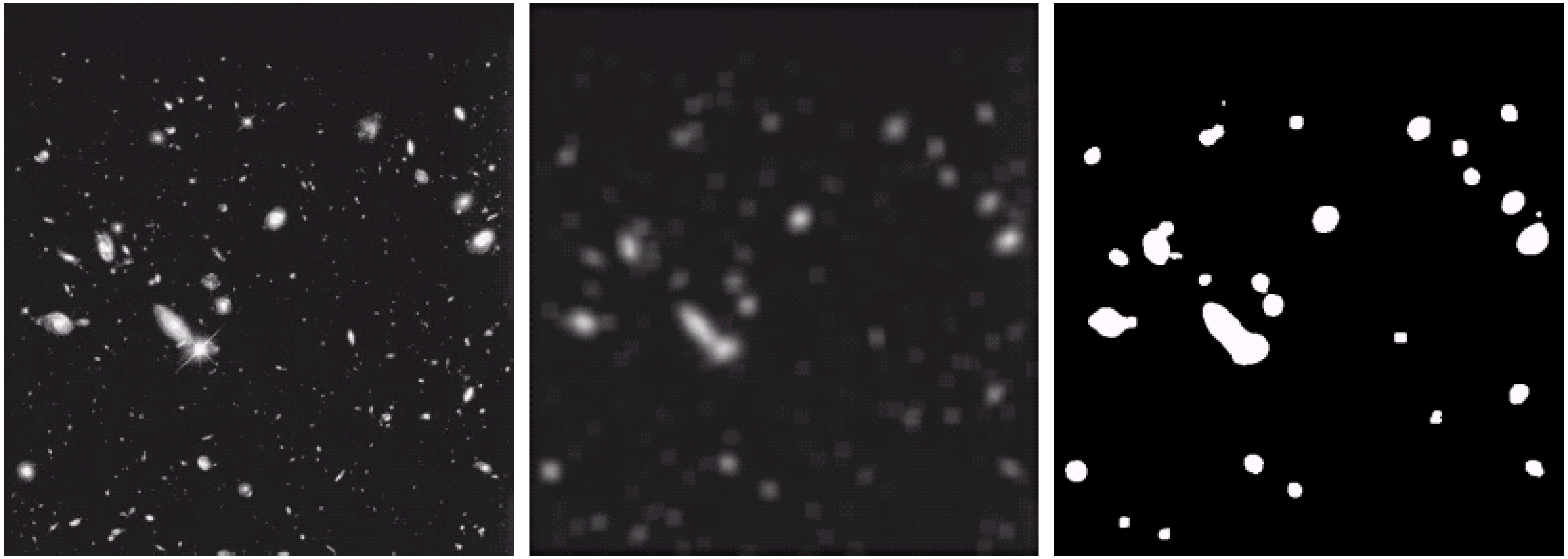


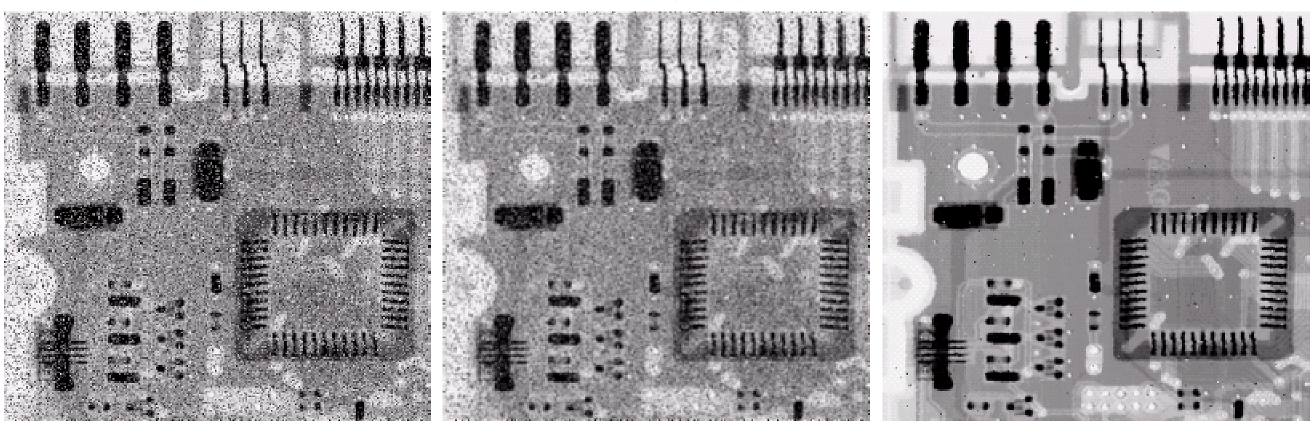
FIGURE 3.36 (a) Image from the Hubble Space Telescope. (b) Image processed by a 15 × 15 averaging mask. (c) Result of thresholding (b). (Original image courtesy of NASA.)

Order-Statistics Filter

- Sort the gray levels of the neighborhood
  - (0, 1, 2, 2, 3, 4, 5, 6, 6)
    min median max
- Min filter
  - Replace the center pixel with the minimum gray level (0)
- Max filter
  - Replace the center pixel with the maximum gray level (6)
- Median filter
  - Replace the center pixel with the median (3)
  - Excellent suppression of salt-and-pepper noises without blurring

6	4	6
2	1	3
2	5	0

3x3 averaging filter   3x3 median filter



Sharpening Spatial Filters

- Highlight fine detail

- Difference operator
  - cf. summation operator for smoothing
  - Derivative in digital domain

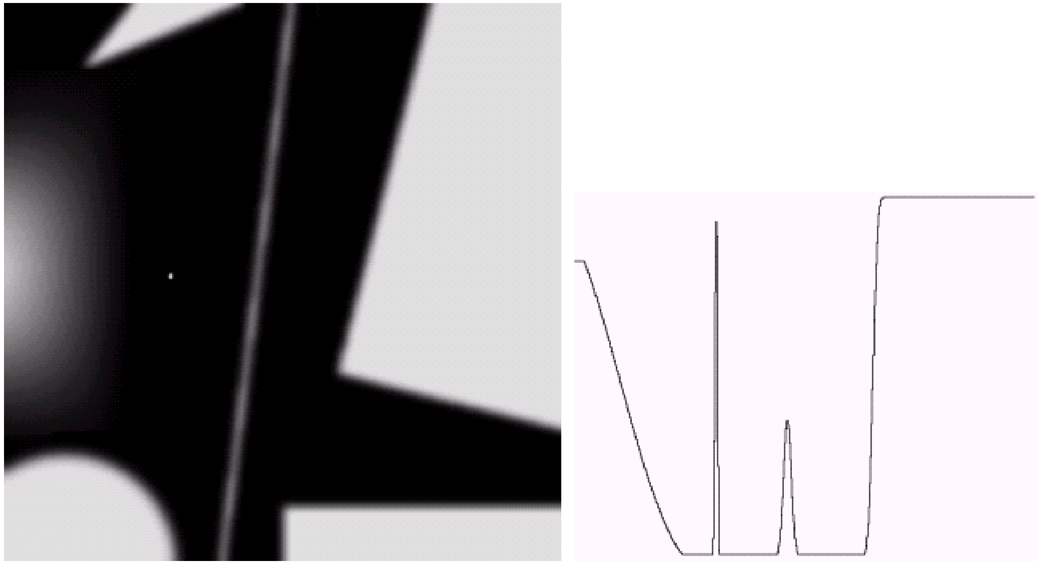
- 1^{st}-order derivative (1D case)

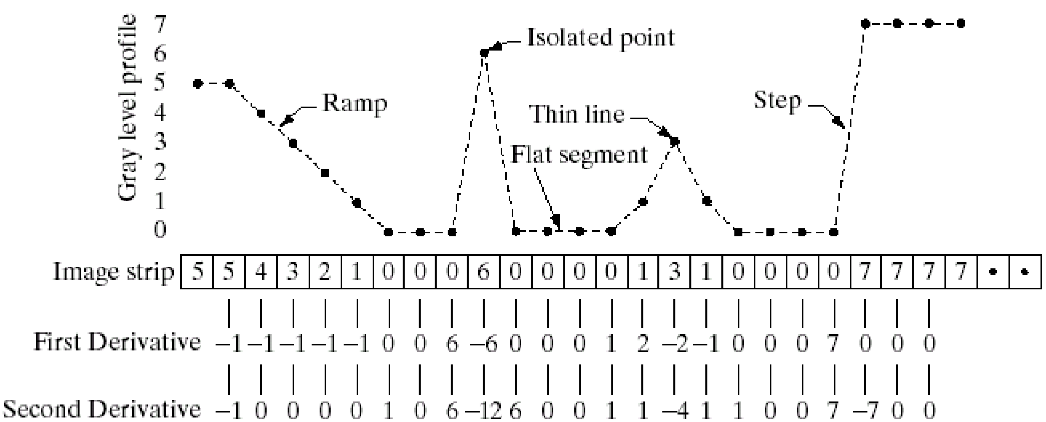
$$\frac{\partial f}{\partial x} = f(x+1) - f(x)$$

- 2^{nd}-order derivative

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

Sharpening Spatial Filters





- 1^{st}-order derivative generates thicker edges
- 2^{nd}-order derivative has a stronger response to fine detail

Laplacian Operator

- Negative definition

$$\begin{aligned}
\nabla^2 f &= \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}, \\
\frac{\partial^2 f}{\partial x^2} &= f(x+1, y) + f(x-1, y) - 2f(x, y), \\
\frac{\partial^2 f}{\partial y^2} &= f(x, y+1) + f(x, y-1) - 2f(x, y), \\
\nabla^2 f &= f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1) - 4f(x, y).
\end{aligned}$$

- Positive definition

$$\nabla^2 f = -[f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1)] + 4f(x, y).$$

- Diagonal derivatives also can be included.
- We will use only the positive definitions.

Laplacian Operator

- Laplacian mask

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

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

- Laplacian operator indicates how brighter the current pixel is than the neighborhood
  - Gray level discontinuity → edge lines
  - Flat background → zero output

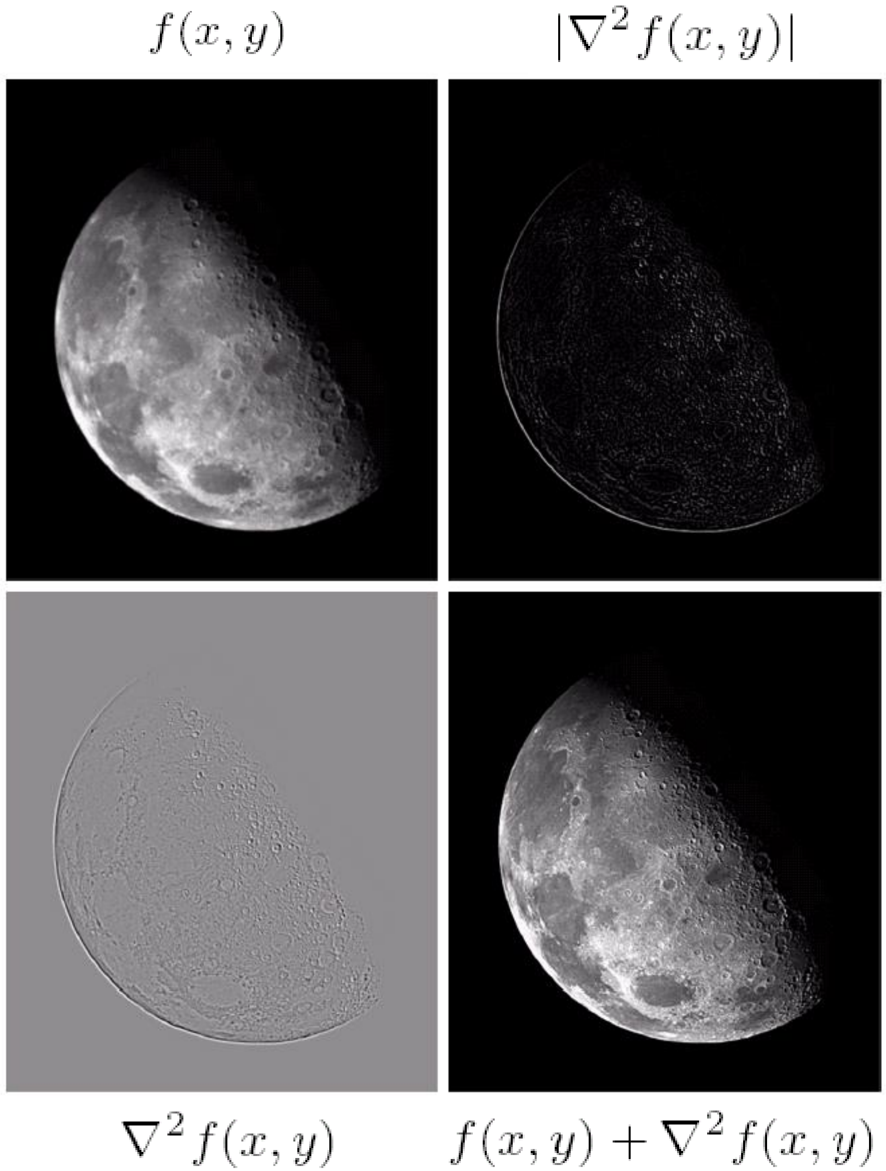
- Background features can be recovered by adding the original image to the Laplacian image

$$g(x,y) = f(x,y) + \nabla^2 f(x,y)$$

0	-1	0
-1	5	-1
0	-1	0

-1	-1	-1
-1	9	-1
-1	-1	-1

Laplacian Operator

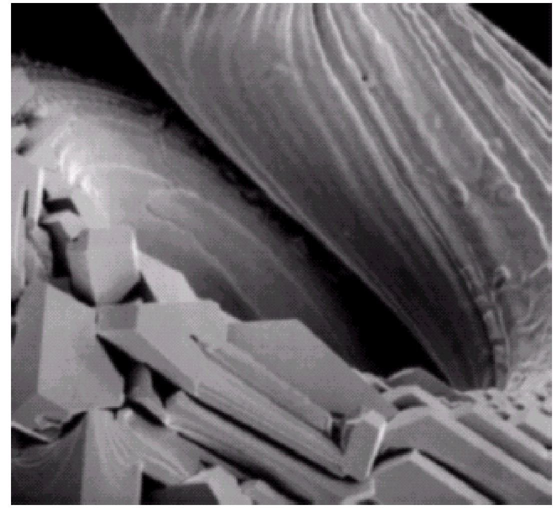


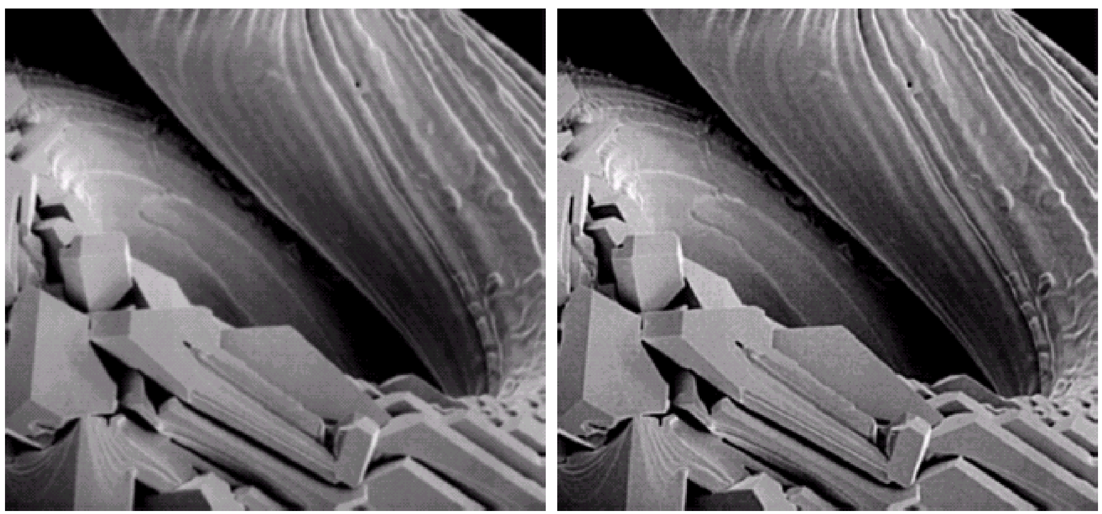
- Enhancing details
- Frequently used sharpening filter

Laplacian Operator

0	−1	0
−1	5	−1
0	−1	0

−1	−1	−1
−1	9	−1
−1	−1	−1





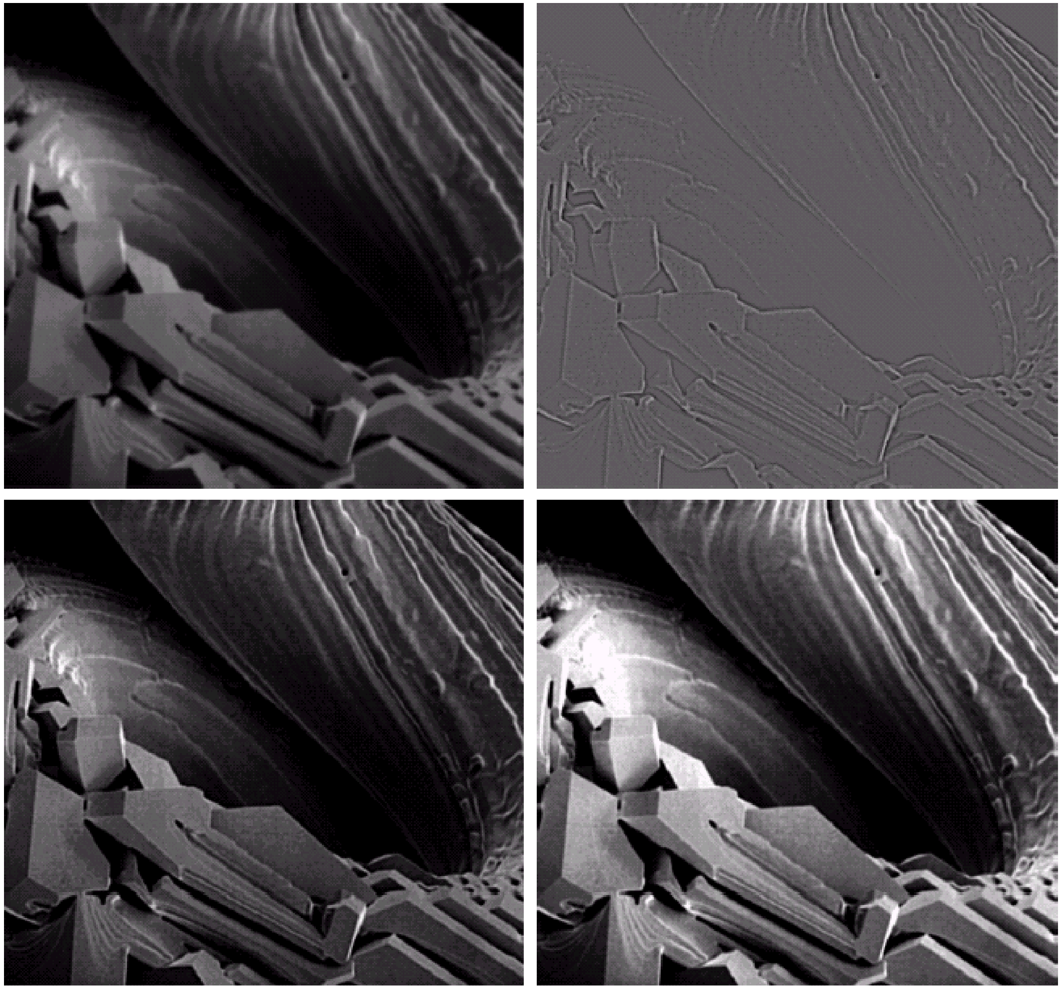
a b c
d e

FIGURE 3.41 (a) Composite Laplacian mask. (b) A second composite mask. (c) Scanning electron microscope image. (d) and (e) Results of filtering with the masks in (a) and (b), respectively. Note how much sharper (e) is than (d). (Original image courtesy of Mr. Michael Shaffer, Department of Geological Sciences, University of Oregon, Eugene.)

High-boost filtering Using Laplacian Operator

- Input image is darker than desired

$$f_{\text{hb}} = Af(x,y) + \nabla^2 f(x,y)$$



a b
c d

FIGURE 3.43 (a) Same as Fig. 3.41(c), but darker. (a) Laplacian of (a) computed with the mask in Fig. 3.42(b) using $A = 0$. (c) Laplacian enhanced image using the mask in Fig. 3.42(b) with $A = 1$. (d) Same as (c), but using $A = 1.7$.

Gradient Operator

- Gradient

$$\nabla \mathbf{f} = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$$

- Magnitude of gradient

$$\nabla f = (G_x^2 + G_y^2)^{1/2} = \left[(\frac{\partial f}{\partial x})^2 + (\frac{\partial f}{\partial y})^2 \right]^{1/2}$$

- Discrete approximation of 1st derivatives – Sobel Masks

$$\frac{\partial f}{\partial x}$$

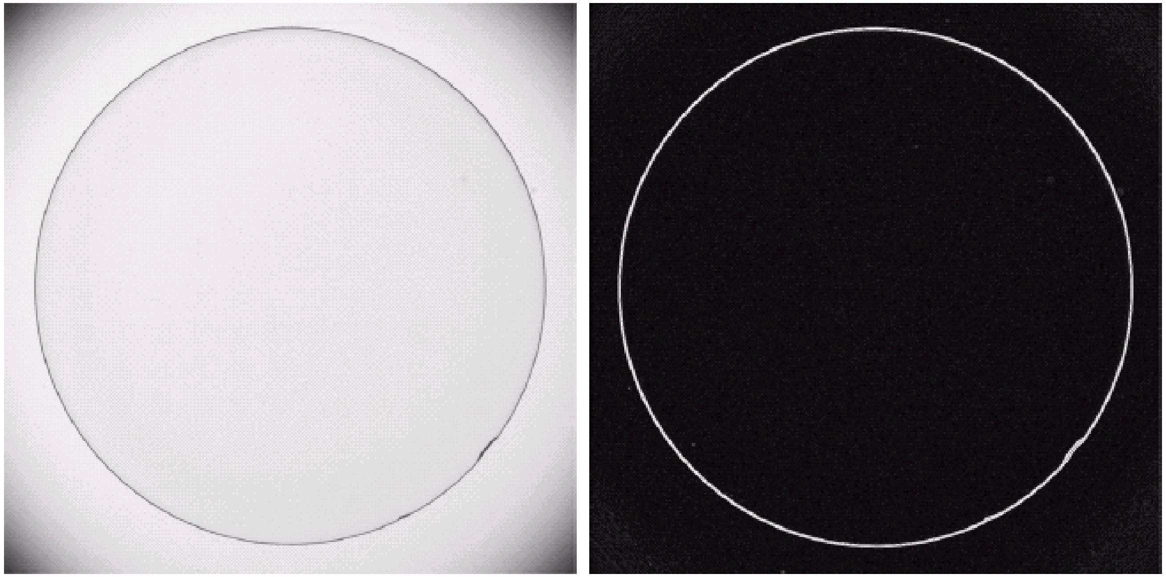
-1	-2	-1
0	0	0
1	2	1

$$\frac{\partial f}{\partial y}$$

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

Gradient Operator

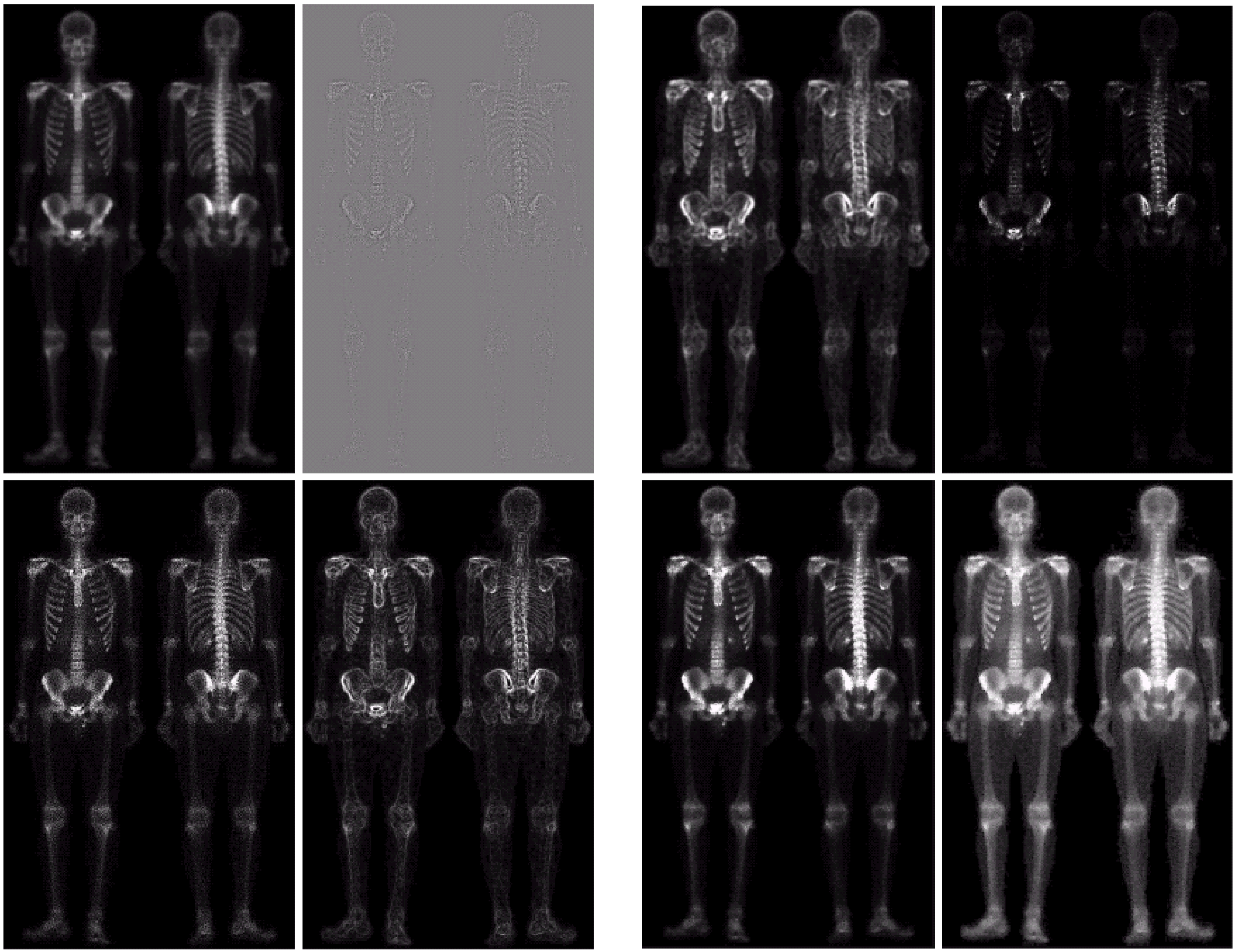
f ∇f



a b

FIGURE 3.45 Optical image of contact lens (note defects on the boundary at 4 and 5 o'clock). (b) Sobel gradient. (Original image courtesy of Mr. Pete Sites, Perceptics Corporation.)

Combining Spatial Enhancement Methods - Art



(a) original
(b) Laplacian of (a)
(e) smoothed (a)
(f) = (c)x(e)
(c) = (a)+(b)
(d) gradient of (a)
(g) = (a)+(f)
(h) power-law transform of (g)