THE UNIVERSITY of WISCONSIN MADISON

Image Enhancement

Produced by University Communications

Image Enhancement

- Goal: to modify an image so that its utilization on a particular application is enhanced.
- A set of ad hoc tools applicable based on viewer's specific needs.
- No general theory on image enhancement exists.

- Methods:
  - Spatial domain
    - Pixel processing
      - Gray level transformation: Data independent
      - histogram processing: Data-dependent
      - Arithmetic ops
    - Spatial filtering
  - Frequency domain filtering

Image Enhancement Effects

- Resizing, cropping
- Contrasts enhancement: sharpening & softening
- Edge enhancement
- Brightness adjustment, equalization
- Color adjustment, gamma correction
- Noise reduction/unwanted object removal
- Geometric adjustment, lens error correction
- Special enhancement techniques:
  - Red eye removal
  - Hand motion compensation, motion blur reduction

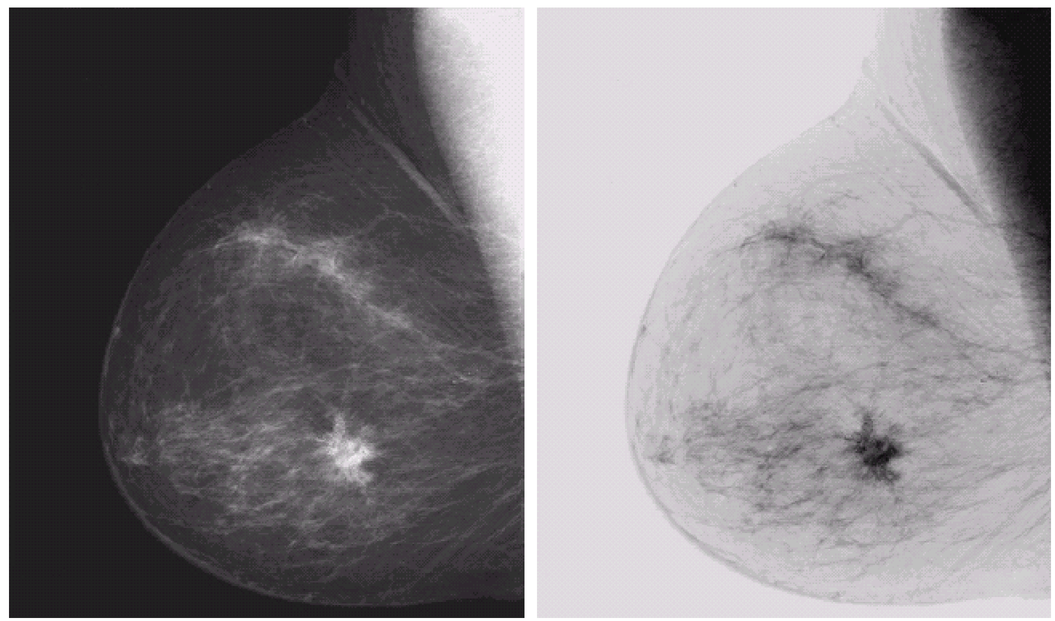
Image Enhancement:
Pixel Gray Scale Processing

Gray-level Transforms

- Operated on individual pixel's intensity values: $s = \mathrm{T}(r)$. r: *original intensity*, s: *new intensity*
- Data independent pixel-based enhancement method.
- Approaches
  - Image negatives
  - Log transform
  - Power law transform
  - Piece-wise linear transform

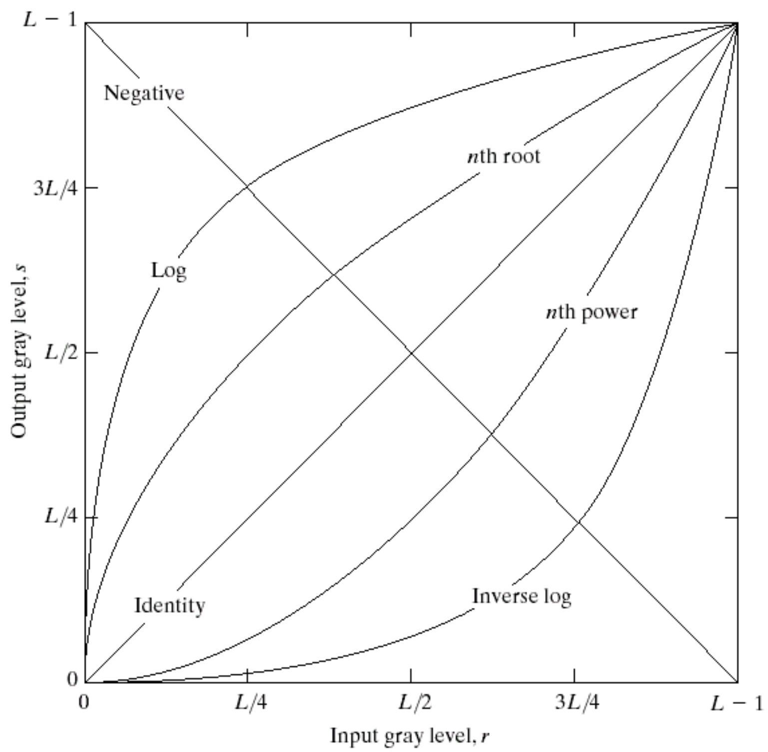
Image Negatives

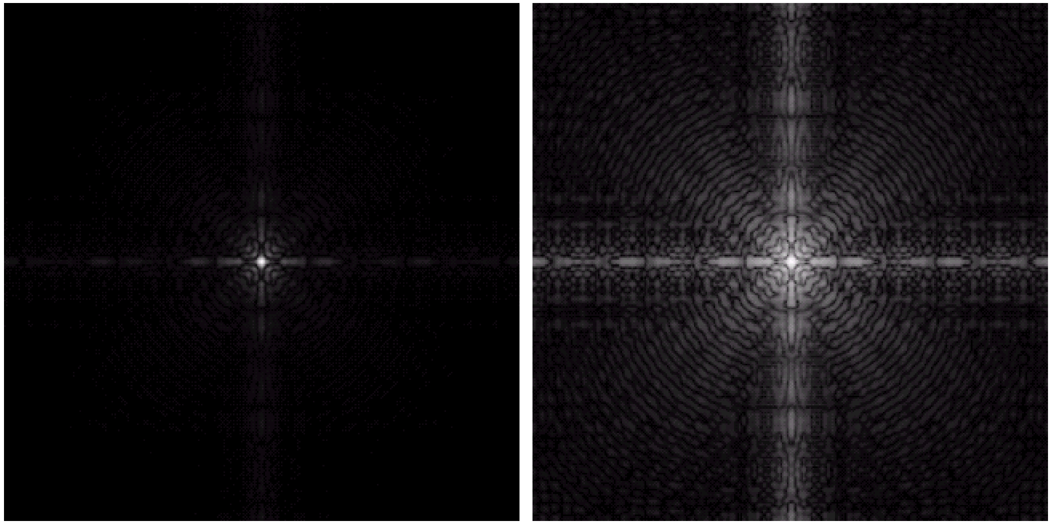
- $s = \mathrm{T}(r) = L\text{-}1\text{-}r$
- Similar to photo negatives.
- Suitable for enhancing white or gray details in dark background.



Log Gray-level Transform

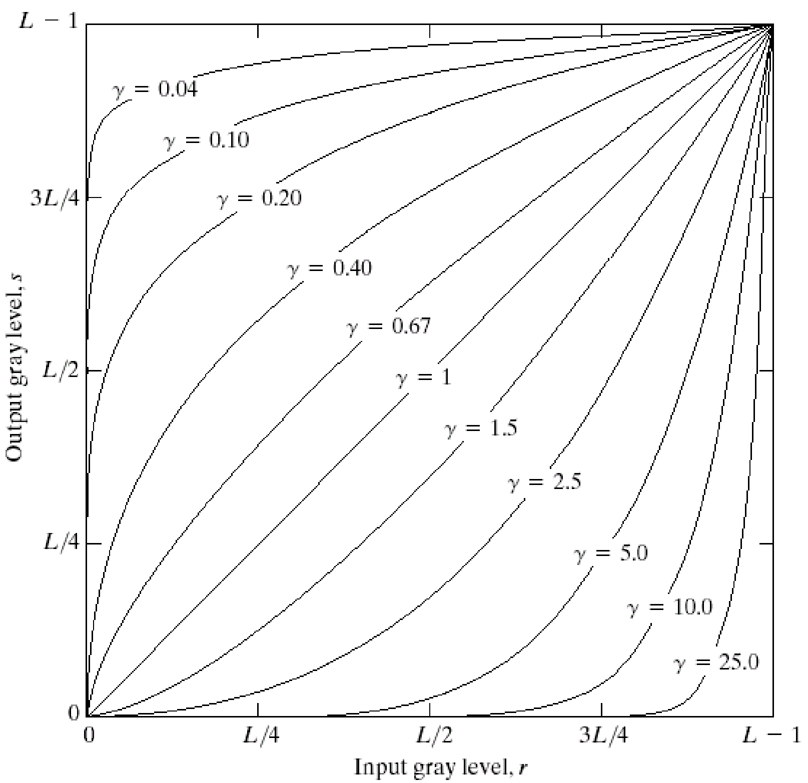
- $s = \mathrm{T}(r) = c \log(1+r)$
- expand dark value to enhance details of dark area



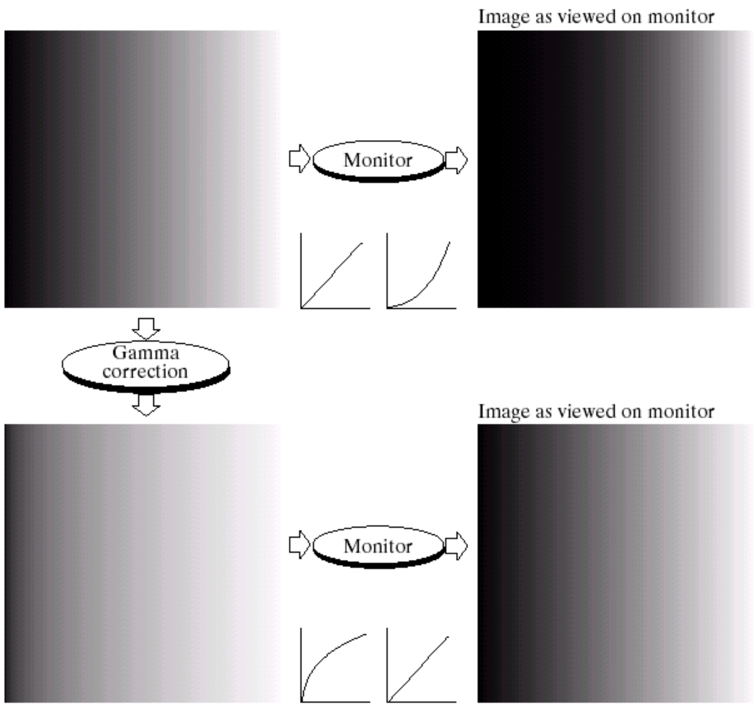


Power Law Gray-level Transform

- $s = \mathrm{T}(r) = c\, r^{g}$



- Gamma correction: to compensate the built-in power law compression due to display characteristics.

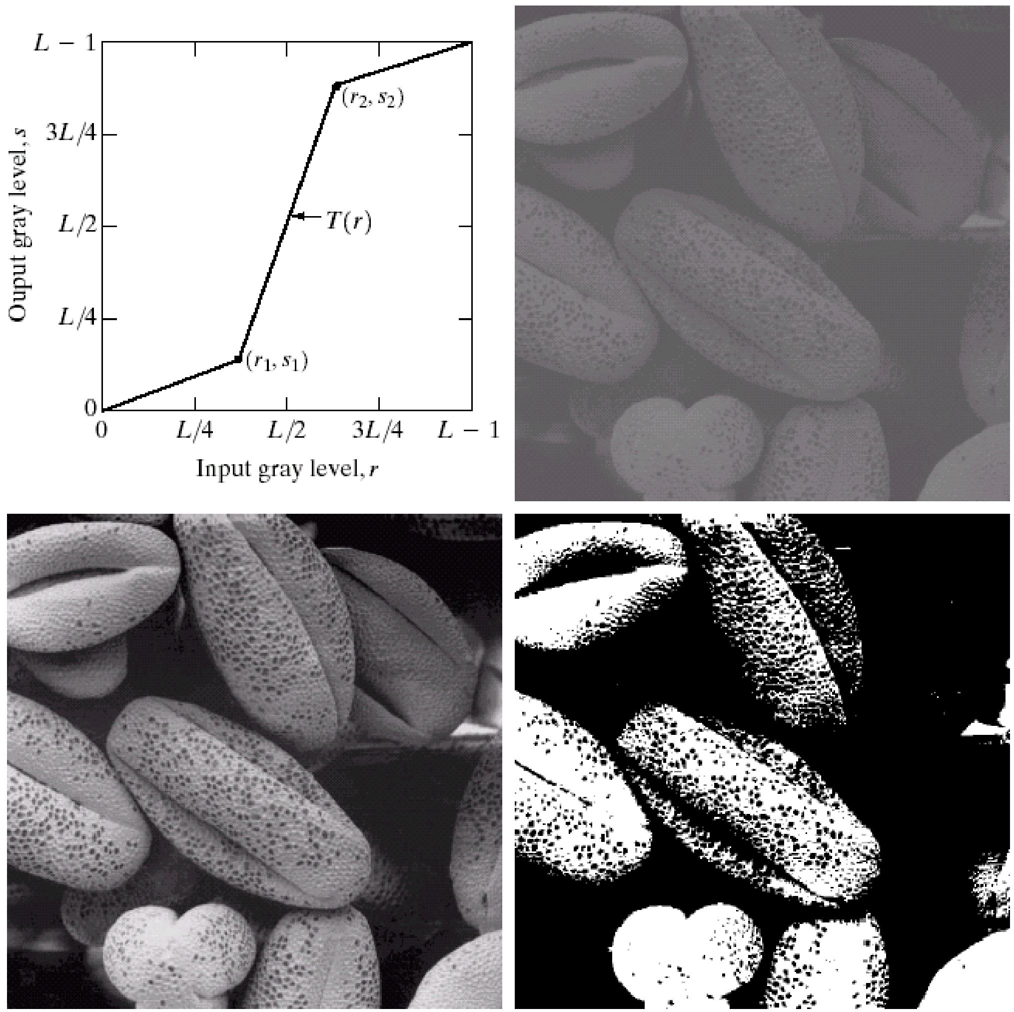


UNIVERSITY OF WISCONSIN–MADISON

Piece-wise Linear Gray-level Transform

- Allow more control on the complexity of T(*r*).
  - Contrast stretching
  - Gray-level slicing
  - Bit-plane slicing

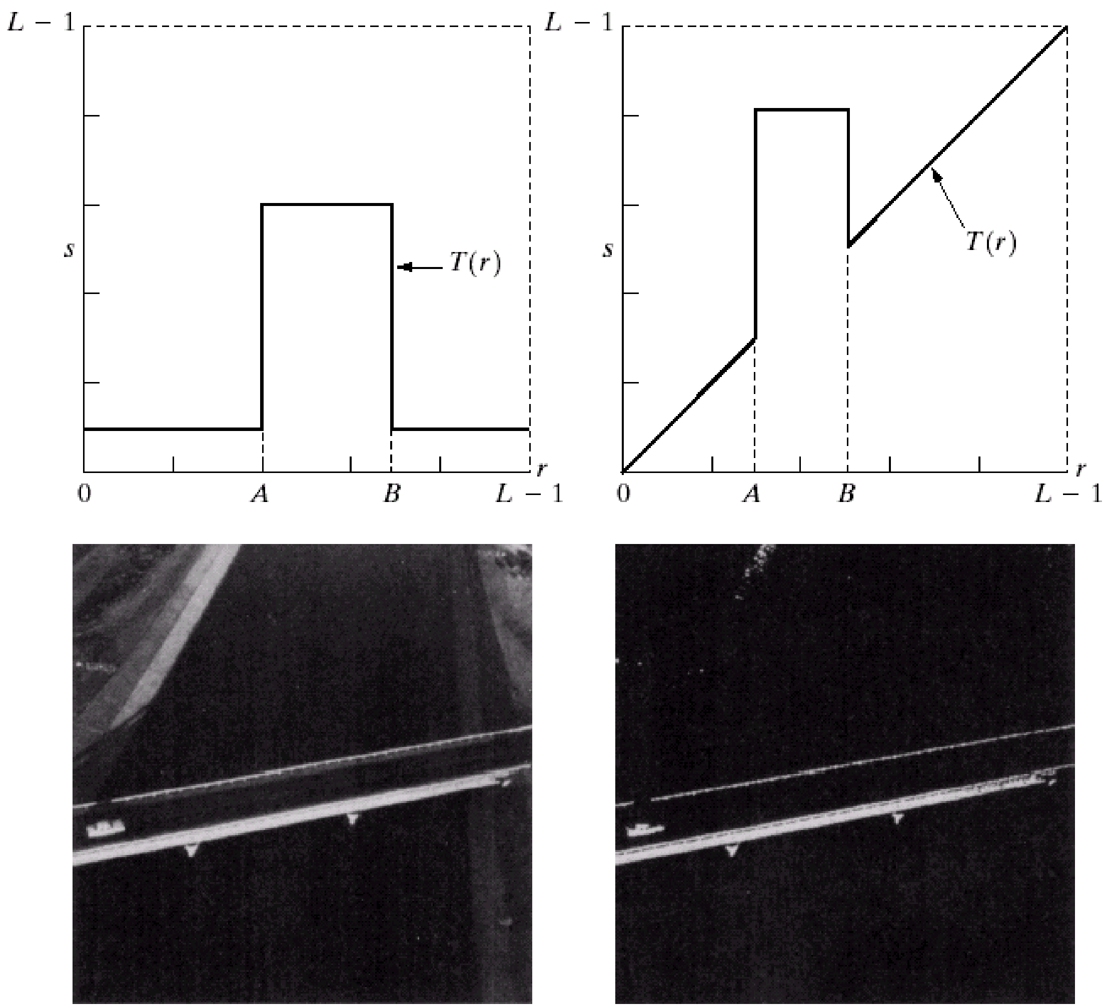
Contrast Stretching



a b
c d

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

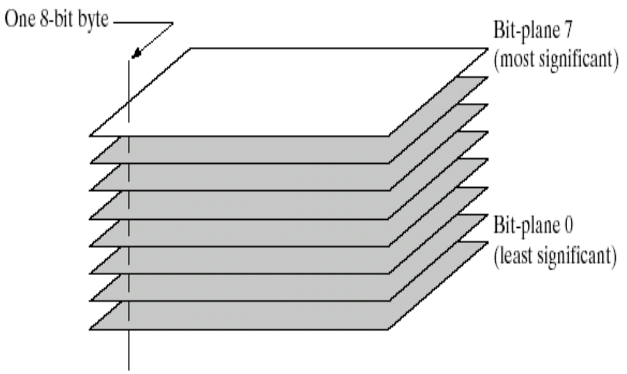
Gray-level Slicing

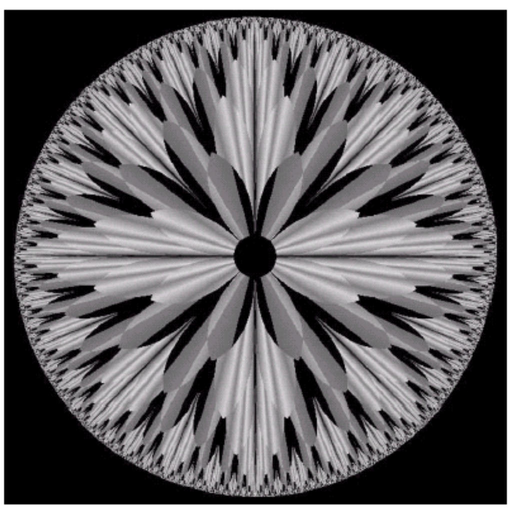


a b
c d

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).

Bit-Slicing





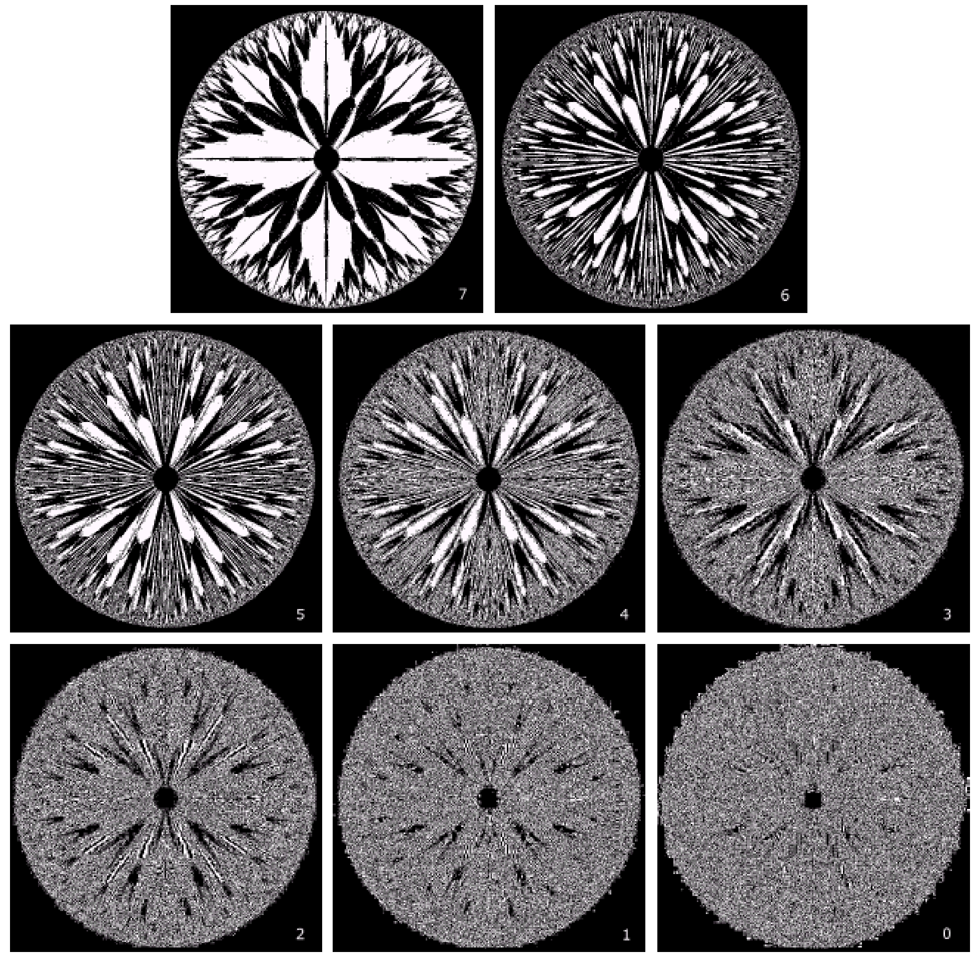
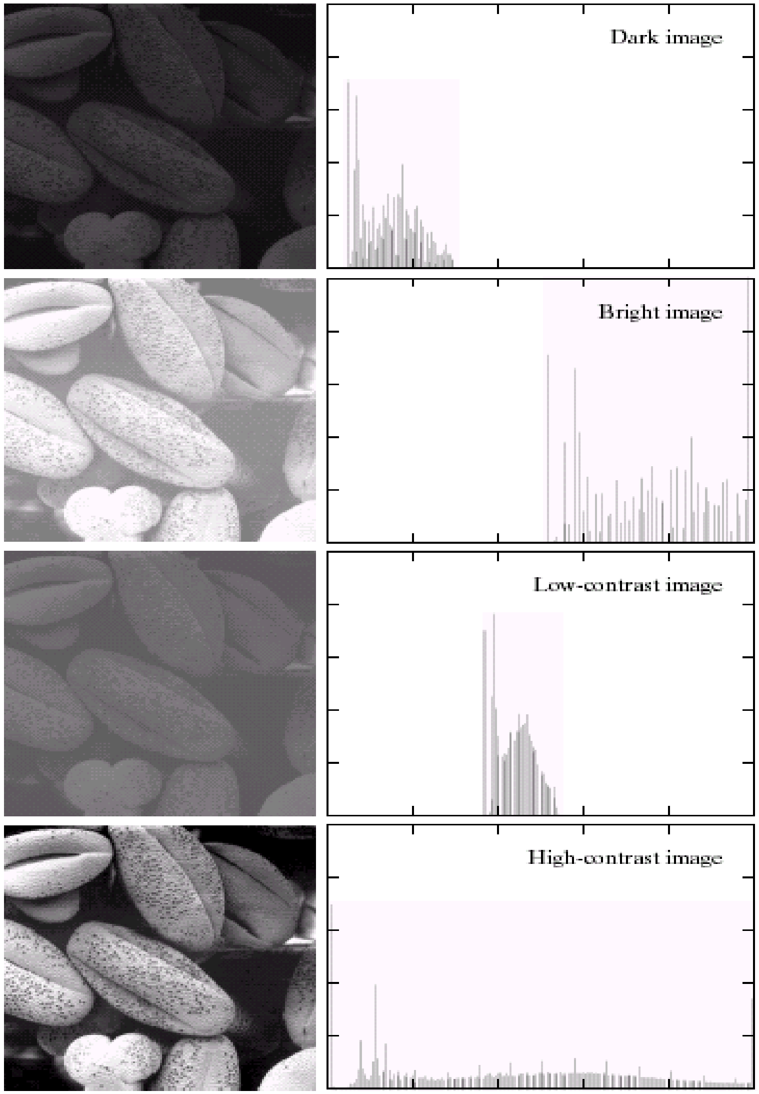


FIGURE 3.14 The eight bit planes of the image in Fig. 3.13. The number at the bottom, right of each image identifies the bit plane.

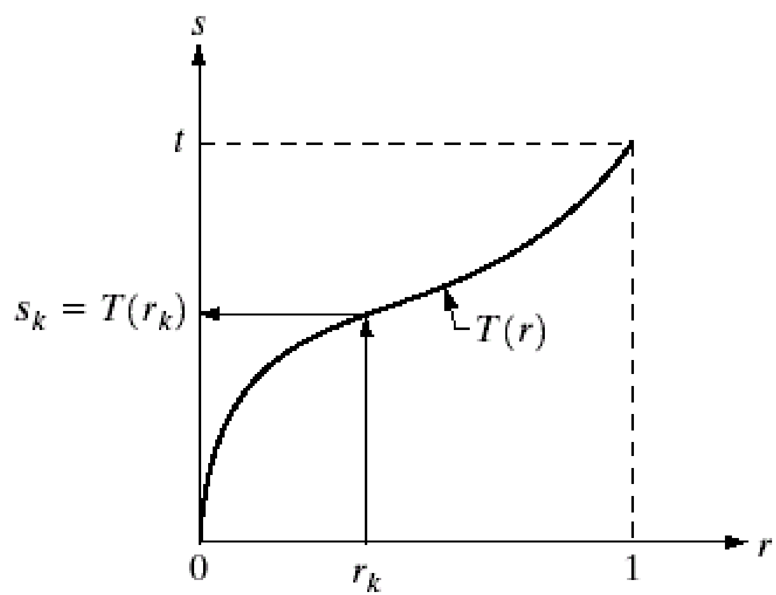
Histogram Processing

- Data-dependent pixel-based image enhancement method.
- Histogram = PDF of image pixels.
  - Assumption: each image pixel is drawn from the same PDF independently (i.i.d.)
  - Several effects of histograms are shown at the right side.



Histogram Equalization

- A gray-level transformation method that forces the transformed gray level to spread over the entire intensity range.
  - Fully automatic,
  - Data dependent,
  - (usually) Contrast enhanced
- Usually, the discrete-valued histogram equalization algorithm does not yield exact uniform distribution of histogram.
- In practice, one may prefer "histogram specification"



$$s = T(r) = \int_{w=0}^{r} p_r(w)dw$$

$$\Rightarrow p_s(s) = 1, \ 0 \le s \le 1.$$

Functions of Random Variables

$Lemma\,1$. Let $F_R(r)$ and $F_S(s)$ be the cdf of original and transformed images respectively. Then for each $s = T(r),\ 0 \le r, s \le 1,\ F_S\left(s\right) = \Pr.\left\{S \le s\right\} = \Pr.\left\{R \le r\right\} = F_R\left(r\right)$

In orther words, fraction of pixels whose value $R \le r$ and fraction of pixels of transfored image whose values $S \le s = T\left(r\right)$ are the same.

Histogram Equalization

An equalized histogram $\Rightarrow p_s(s) = 1,\ 0 \le s \le 1.$

Equivalently, $F_S(s) = \begin{cases} 0 & s < 0; \\ s & 0 \le s \le 1; \\ 1 & s > 1. \end{cases}$

In other words, $F_R(r) = F_S(s) = T(r) = s$

Thus, $s = T(r) = F_R(r) = \int_{w=0}^{r} p_R(w)dw$

WISCONSIN
UNIVERSITY OF WISCONSIN–MADISON

Practical Considerations

- r, s $\in$ {0, 1, …, L-1} instead of [0, 1].

  $\Rightarrow$ integration is replaced by summation
- Assume # pixels in the image is N. # of pixels whose gray scale value is n_r. Then the mapping becomes

$$T(r) = \int_{w=0}^{r} p_R(w)dw$$

$$\Rightarrow s = \left[\frac{L-1}{N} \sum_{w=0}^{r} n_w \right] = \left[(L-1) \cdot cdf(r) \right]$$

the bracket indicates rounding to nearest integer.

Histogram Equalization Example

- Consider a 5×5 image with $L = 4$.

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

r	0	1	2	3
p(r)	6/25	7/25	7/25	5/25
Cdf(r)	6/25	13/25	20/25	25/25
s	1	2	2	3

- Since original image already has an equalized histogram, the effect is not clear in this example.

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

More Practical Considerations

- The number of non-zero bins in the transformed histogram is no larger than that of the original image.
- As such, the equalization process will
  - Move some bins to other locations
  - Combine two or more bins into one at perhaps a different location

Histogram Modification

- One may want to convert the histogram to a target histogram that is not uniformly distributed. Rather with a new pdf $g(s)$.

In this case, one has $F_R(r) = F_S(s) = T(r) = g(s)$

Thus, $T(r) = s = g^{-1}\left(F_R(r)\right) = g^{-1}\left(\int_{w=0}^{r} p_R(w)dw\right)$

Assuming $g^{-1}(\cdot)$ exists over $[0 \quad 1]$.

Histogram Matching

- Transform pdf of r to a desired pdf $p_s(s)$.
- A generalization of histogram equalization.
- Basic idea: *Given* $p_r(r)$ and desired pdf $p_z(z)$, find a transform $z = T(r)$, such that $P(Z \le z) = P(R \le r)$.

Indirect Method

- Indirect approach:
  - First equalize the histogram using transform s = T(r).
  - Equalize the desired histogram v = G(z).
  - Set v = s to obtain the composite transform

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

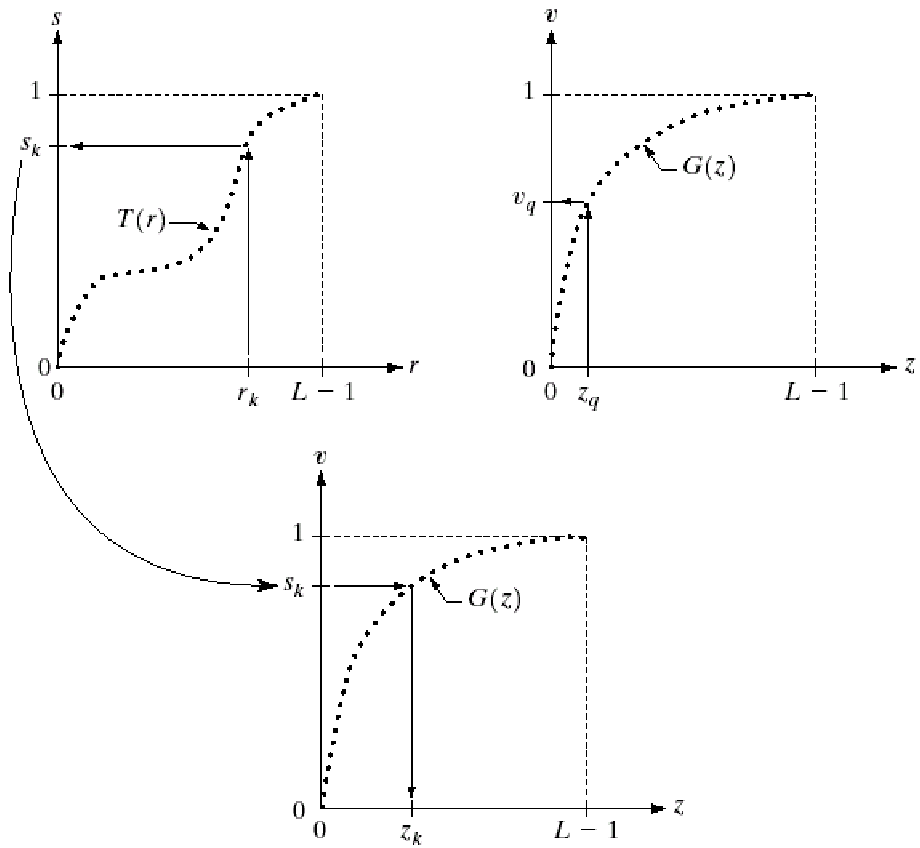
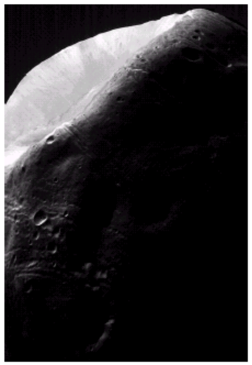
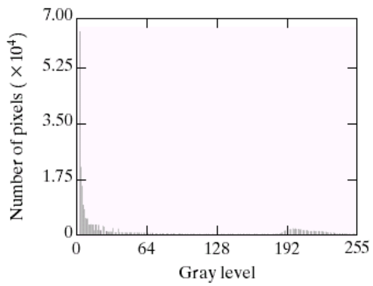
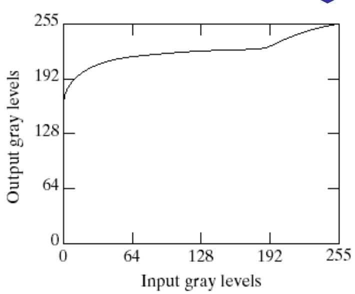


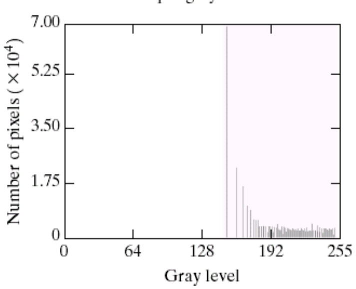
Fig. 3.19

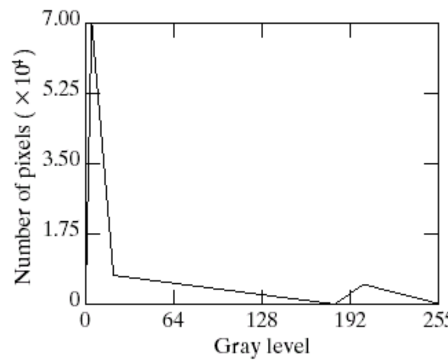
Histogram Matching Example

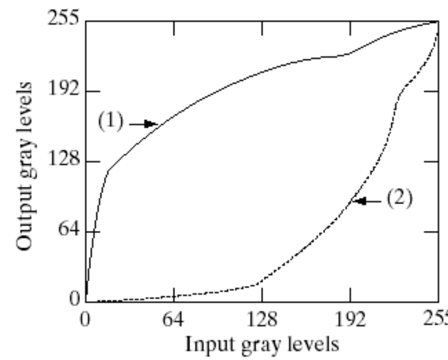


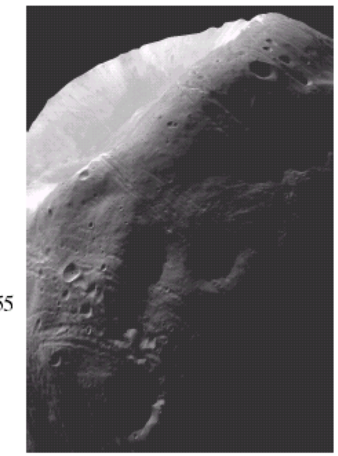


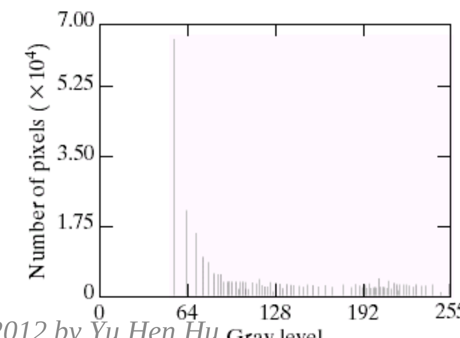




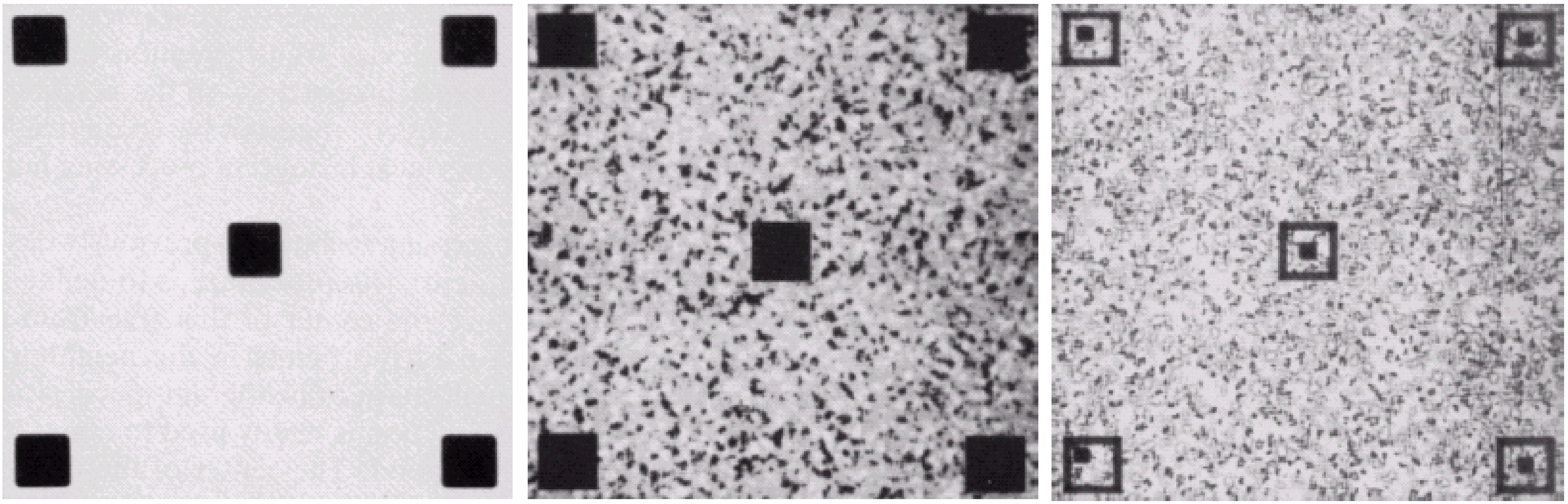








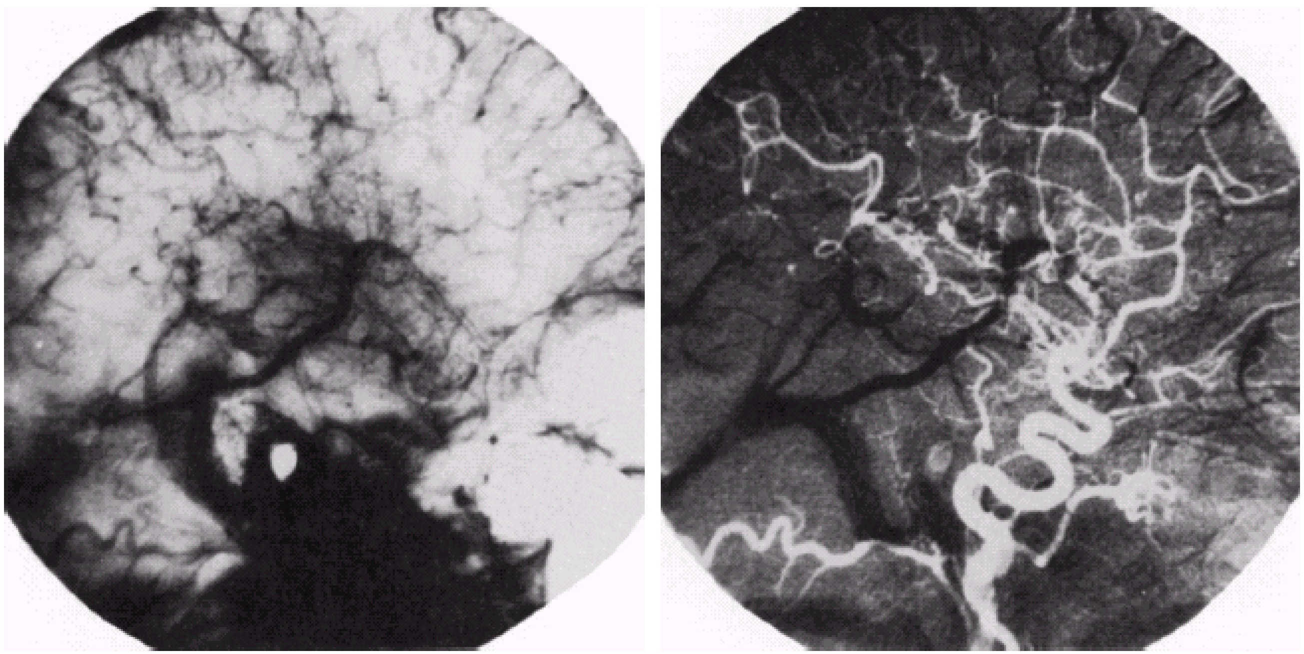
Histogram for Local Enhancement



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 subtraction

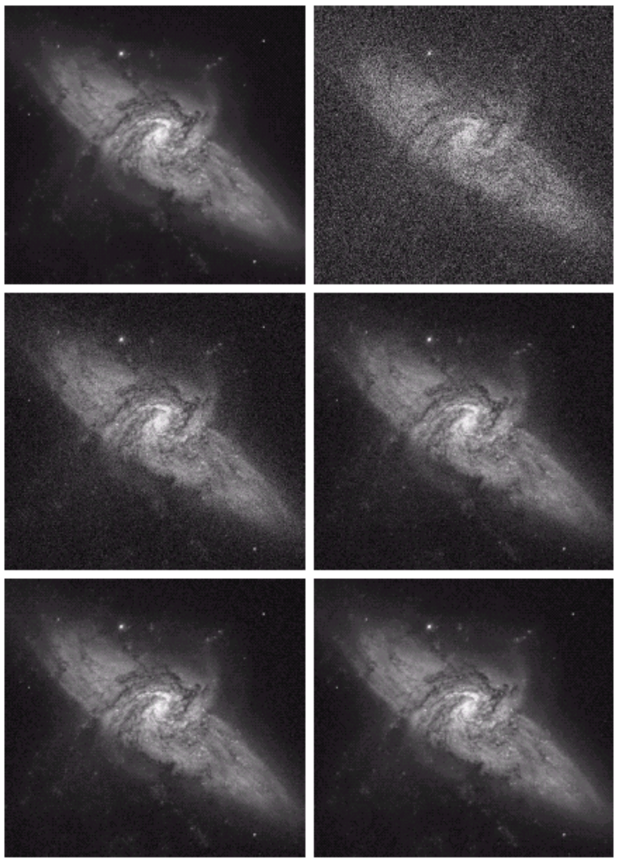


a b

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.

Mask mode radiography

Image Averaging



- Same signal, but different noise realization.
- Averaging of many such images will enhance SNR.




Image Enhancement: Spatial Domain Filtering

UNIVERSITY OF WISCONSIN–MADISON

Spatial Filtering

$$g(m,n) = \sum_{i=-I}^{I} \sum_{j=-J}^{J} w(i,j)\, f(m-i, n-j)$$

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

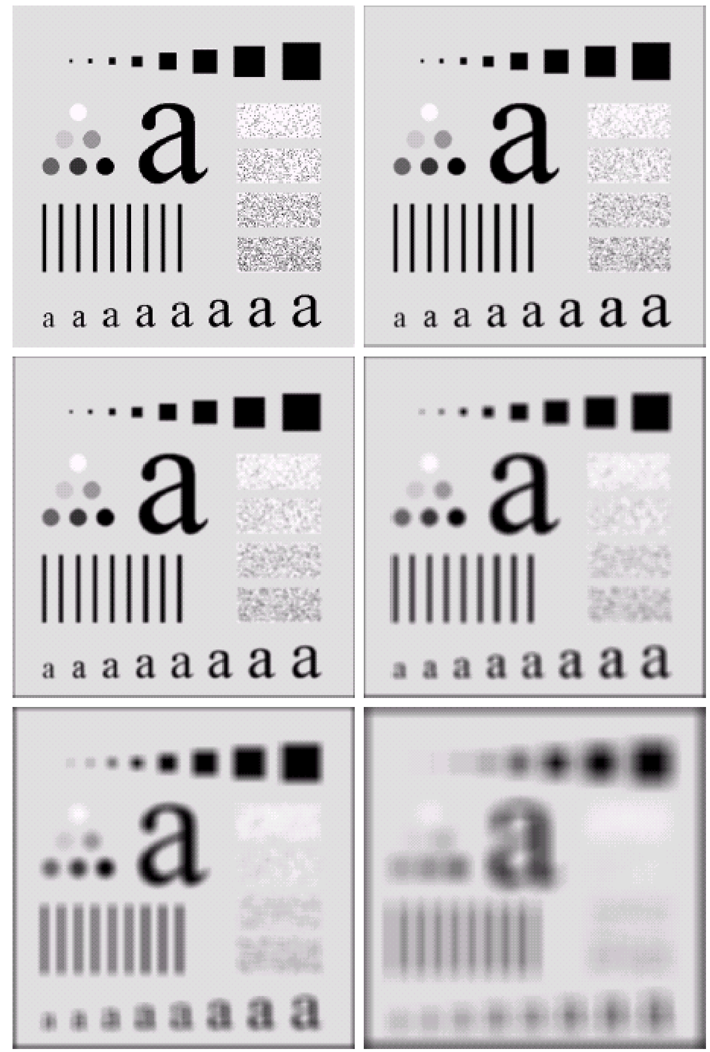
- 2D FIR filtering
  - Mask filtering: convolution of the image with a 2D mask
  - Applications to image enhancement:
    - Smoothing: low pass
    - Sharpening: high pass
- Data-dependent nonlinear filters
  - Local histogram
  - Order statistic filters
    - Medium filter

Smoothing Linear Filters

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

$$g(m,n) = \frac{\sum_{i=-I}^{I} \sum_{j=-J}^{J} w(i,j) f(m-i, n-j)}{\sum_{i=-I}^{I} \sum_{j=-J}^{J} w(i,j)}$$

Normalization of coefficient to ensure $0 \leq g(m,n) \leq L\text{-}1$



a b
c d
e f

FIGURE 3.35 (a) Original image, of size 500 × 500 pixels. (b)–(f) Results of smoothing with square averaging filter masks of sizes $n = 3, 5, 9, 15,$ and 35, respectively. The black squares at the top are of sizes 3, 5, 9, 15, 25, 35, 45, and 55 pixels, respectively; their borders are 25 pixels apart. The letters at the bottom range in size from 10 to 24 points, in increments of 2 points; the large letter at the top is 60 points. The vertical bars are 5 pixels wide and 100 pixels high; their separation is 20 pixels. The diameter of the circles is 25 pixels, and their borders are 15 pixels apart; their gray levels range from 0% to 100% black in increments of 20%. The background of the image is 10% black. The noisy rectangles are of size 50 × 120 pixels.

Sharpening Linear Filters

- High boosting filter:

0	−1	0
−1	$A+4$	−1
0	−1	0

−1	−1	−1
−1	$A+8$	−1
−1	−1	−1

  - A

- Derivative filter:
  - Use derivatives to approximate high pass filters. Usually 2nd derivatives are preferred. The most common one is the Laplacian operator.

- Laplacian operator:

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

$$= f(x+1,y) + f(x-1,y) + f(x,y+1) + f(x,y-1) - 4f(x,y)$$

0	1	0
1	−4	1
0	1	0

1	1	1
1	−8	1
1	1	1

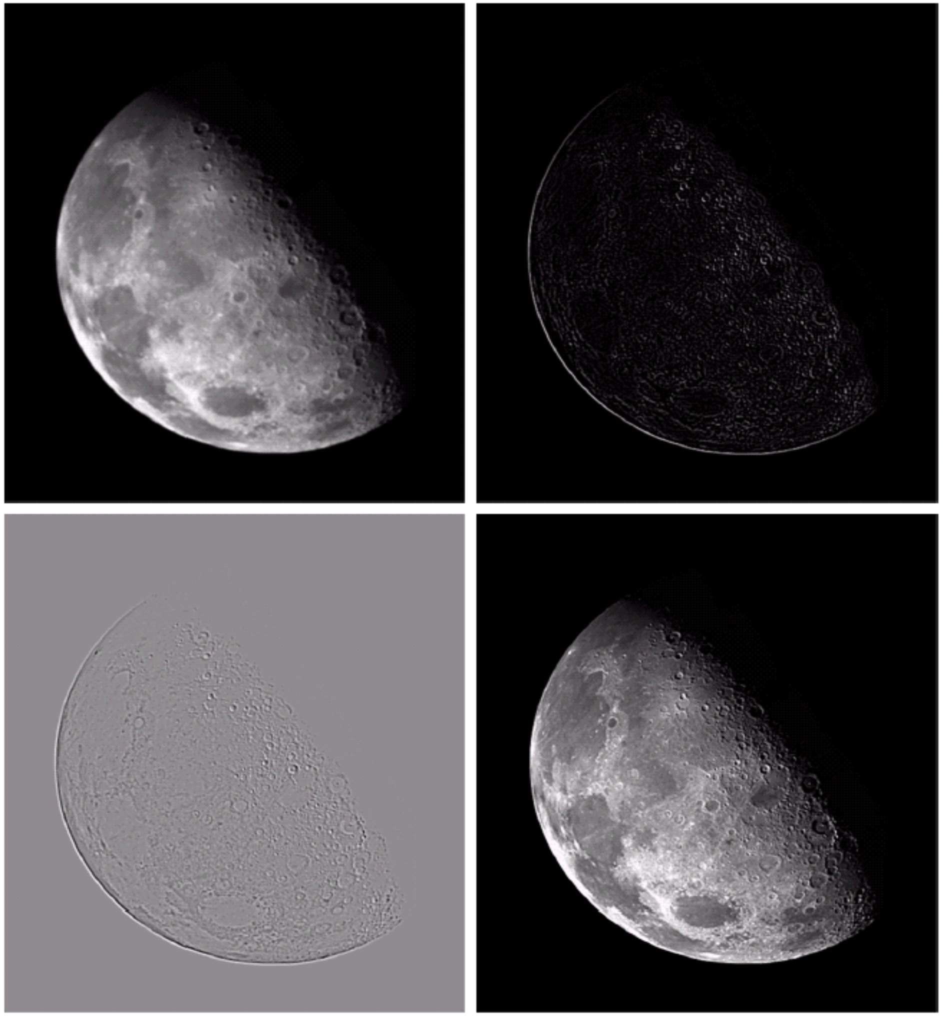
0	−1	0
−1	4	−1
0	−1	0

−1	−1	−1
−1	8	−1
−1	−1	−1

UNIVERSITY OF WISCONSIN–MADISON

Laplacian Filter for Image Enhancement

$$g(x,y)=\begin{cases} f(x,y)-\nabla^2 f(x,y) & \text{center of mask} < 0; \\ f(x,y)+\nabla^2 f(x,y) & \text{center of mask} > 0. \end{cases}$$



a	b
c	d

FIGURE 3.40 (a) Image of the North Pole of the moon. (b) Laplacian-filtered image. (c) Laplacian image scaled for display purposes. (d) Image enhanced by using Eq. (3.7-5). (Original image courtesy of NASA.)

Gradient filters

−1	0
0	1

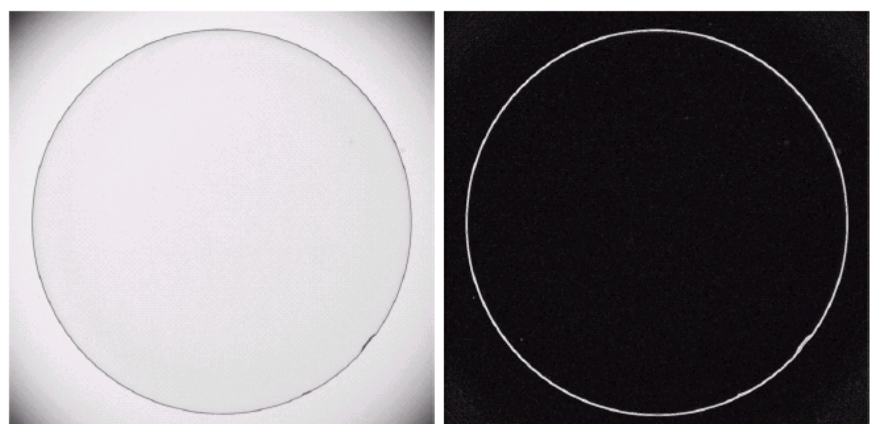
0	−1
1	0

Roberts cross-gradient operator

−1	−2	−1
0	0	0
1	2	1

−1	0	1
−2	0	2
−1	0	1

Sobel operator

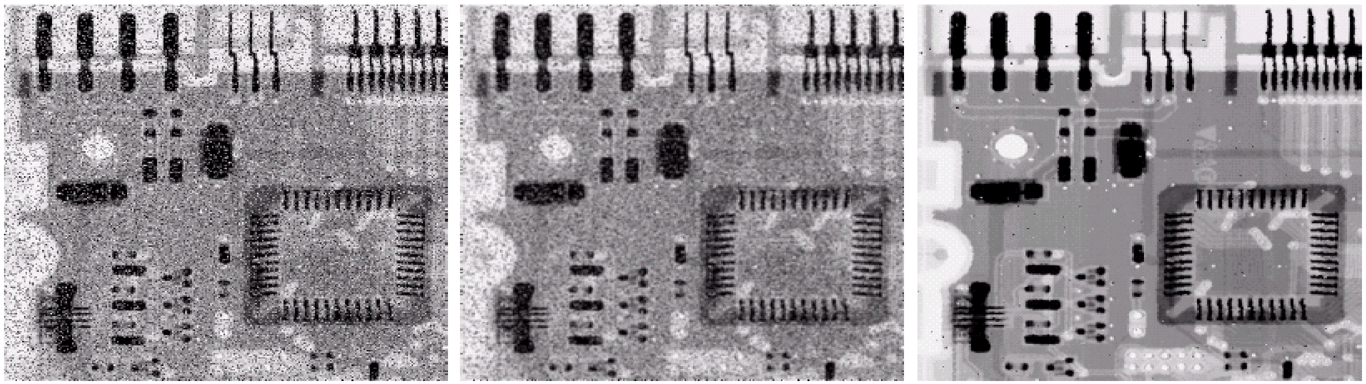


a b

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.)

Local Statistic Filters

- Calculate a local statistics and then replace the center pixel value with the calculated statistics.
- Medium filter
  - Useful in removing impulsive noise (salt-and-pepper noise) without smoothing the rest of the image.



a b c

FIGURE 3.37 (a) X-ray image of circuit board corrupted by salt-and-pepper noise. (b) Noise reduction with a 3 × 3 averaging mask. (c) Noise reduction with a 3 × 3 median filter. (Original image courtesy of Mr. Joseph E. Pascente, Lixi, Inc.)

Image Enhancement: Frequency Domain Processing

Image and Its Fourier Spectrum

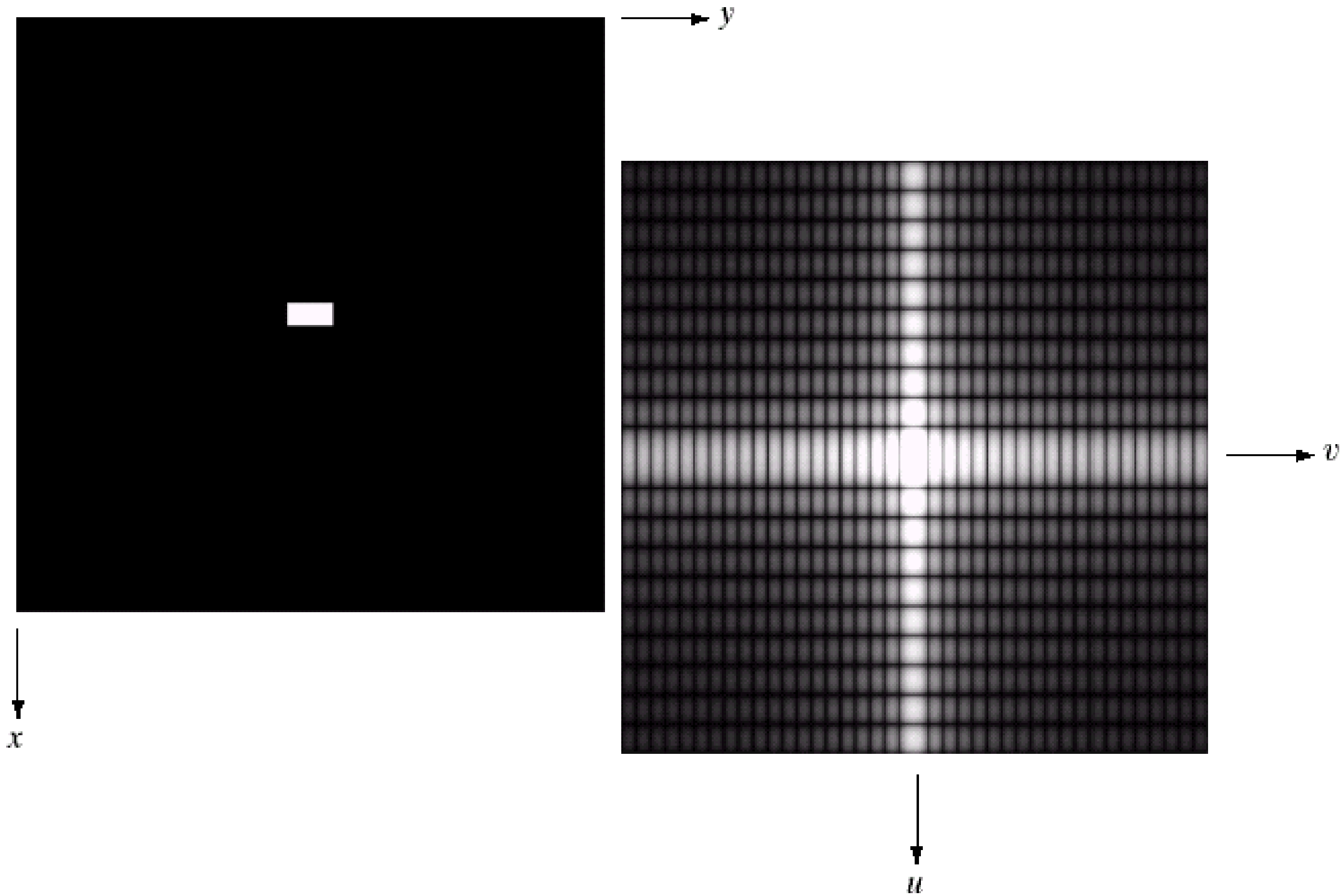


FIGURE 4.3 (a) Image of a 20 × 40 white rectangle on a black background of size 512 × 512 pixels. (b) Centered Fourier spectrum shown after application of the log transformation given in Eq. (3.2-2). Compare with Fig. 4.2.

WISCONSIN
UNIVERSITY OF WISCONSIN–MADISON

Filtering in Frequency Domain: Basic Steps

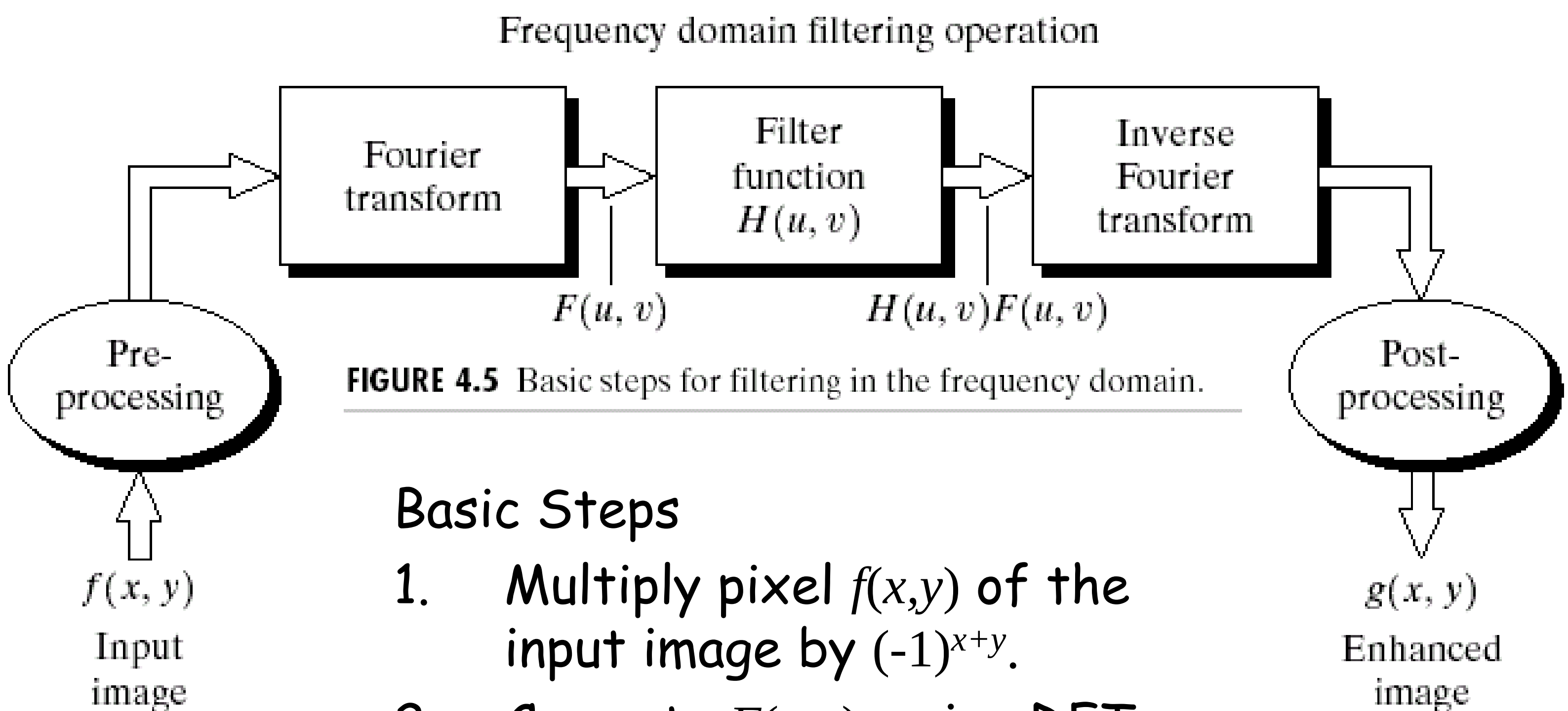


FIGURE 4.5 Basic steps for filtering in the frequency domain.

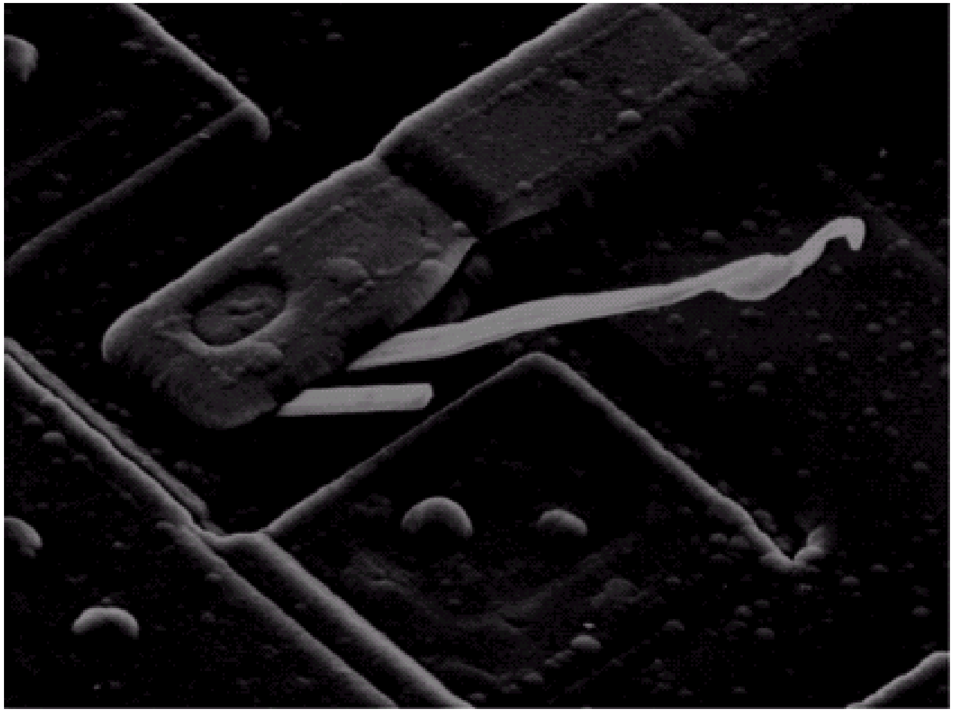
Basic Steps

1. Multiply pixel $f(x,y)$ of the input image by $(-1)^{x+y}$.
2. Compute $F(u,v)$, using DFT
3. $G(u,v)=F(u,v)H(u,v)$
4. $g1(x,y)=F^{-1}\{G(u,v)\}$
5. $g(x,y) = g1(x,y)*(-1)^{x+y}$

Notch Filter

$$H(u,v) = \begin{cases} 0 & u = v = 0 \\ 1 & otherwise. \end{cases}$$

- The frequency response $H(u,v)$ has a notch at origin ($u = v = 0$).
- Effect: reduce mean value.
- After post-processing where gray level is scaled, the mean value of the displayed image is no longer 0.



Low-pass & High-pass Filtering

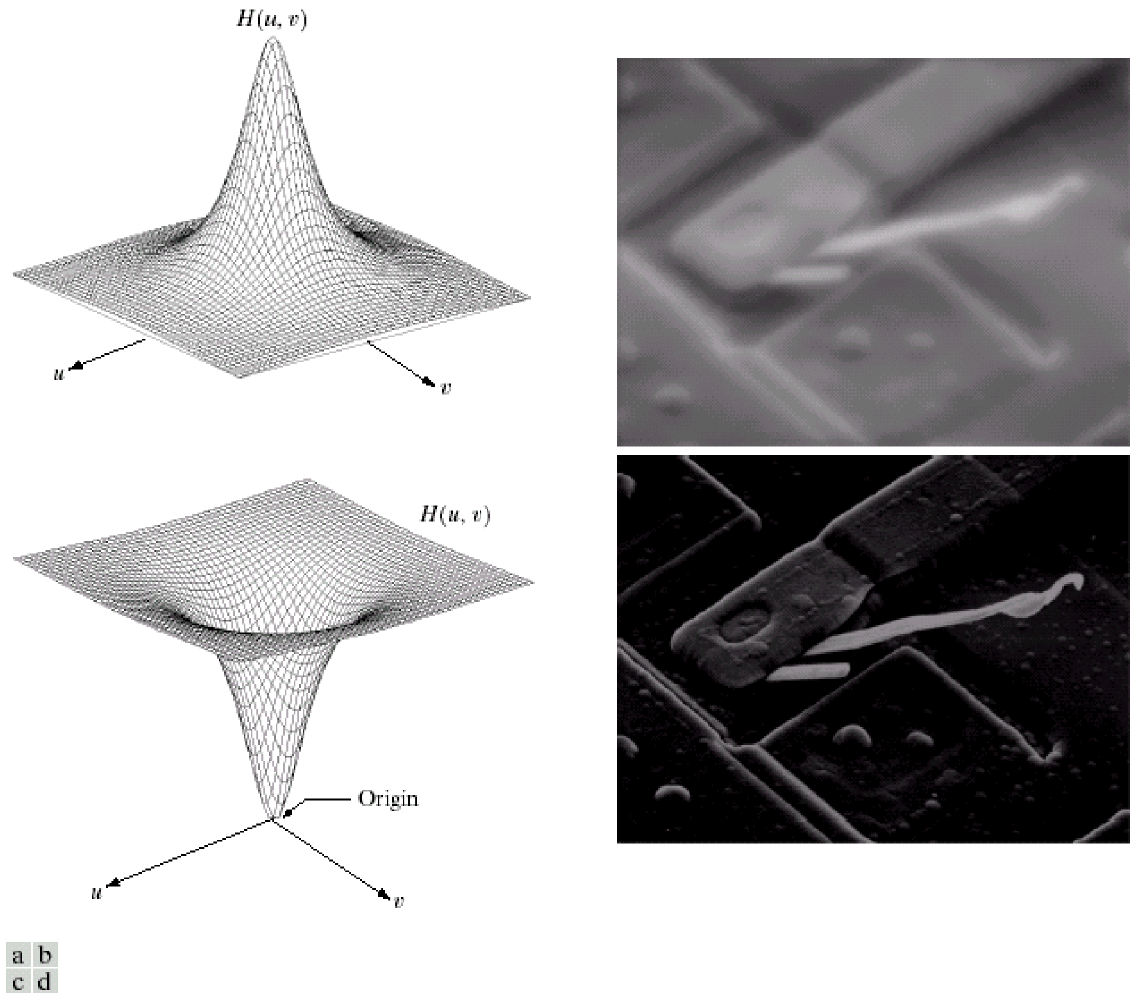


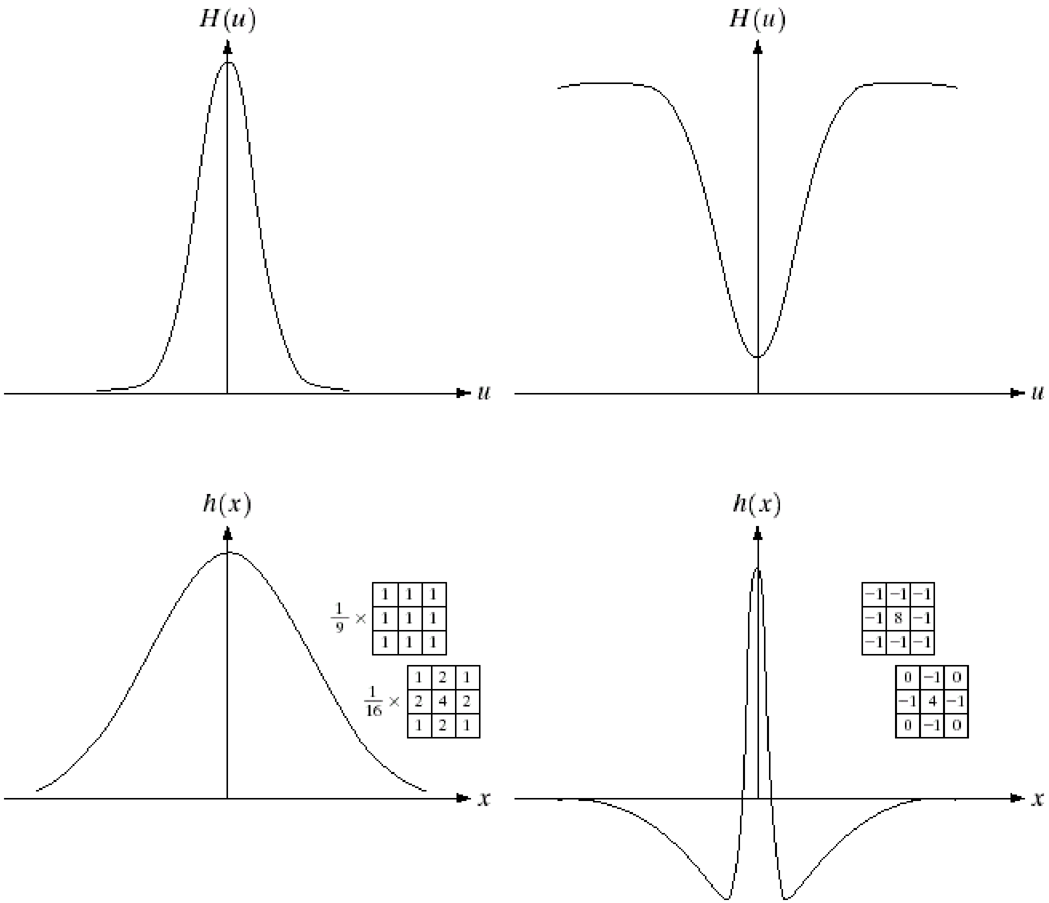
FIGURE 4.7 (a) A two-dimensional lowpass filter function. (b) Result of lowpass filtering the image in Fig. 4.4(a). (c) A two-dimensional highpass filter function. (d) Result of highpass filtering the image in Fig. 4.4(a).

Gaussian Filters

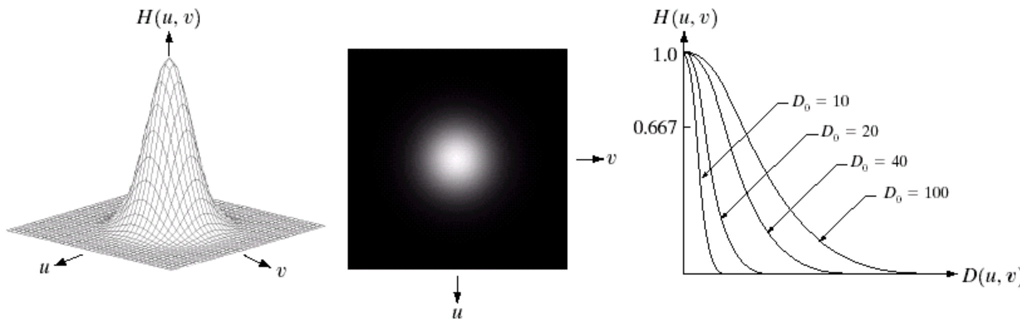
- Fourier Transform pair of Gaussian function

$$\begin{cases} H(u) = Ae^{-u^2/2\sigma^2} \\ h(x) = \sqrt{2\pi}\cdot\sigma\cdot Ae^{-2\pi^2\sigma^2x^2} \end{cases}$$

- Depicted in figures are low-pass and high-pass Gaussian filters, and their spatial response, as well as FIR masking filter approximation.
- High pass Gaussian filter can be constructed from the difference of two Gaussian low pass filters.



Gaussian Low Pass Filters

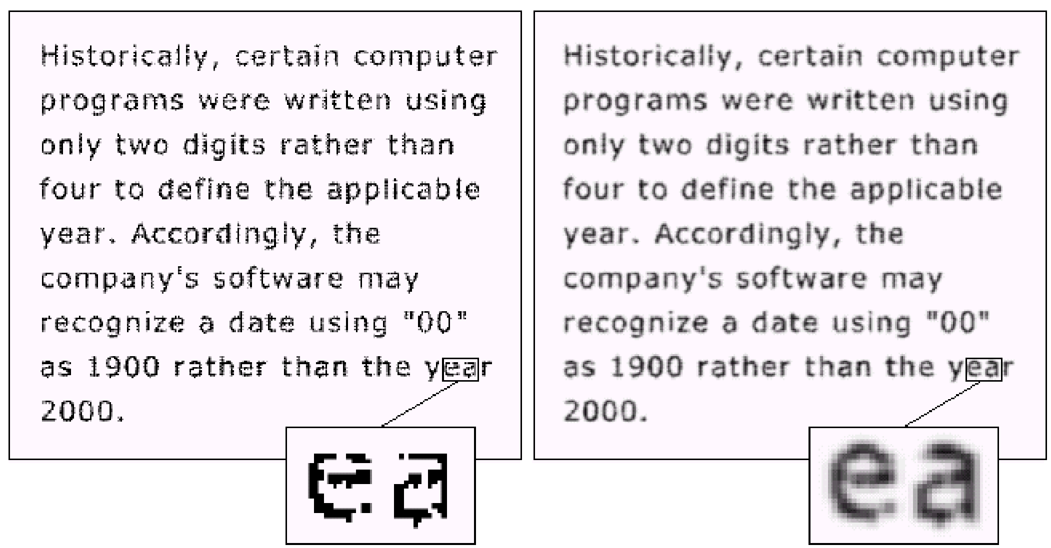


$$H(u,v) = \exp\left(-\frac{\left(D(u,v)\right)^2}{2\sigma^2} \right)$$

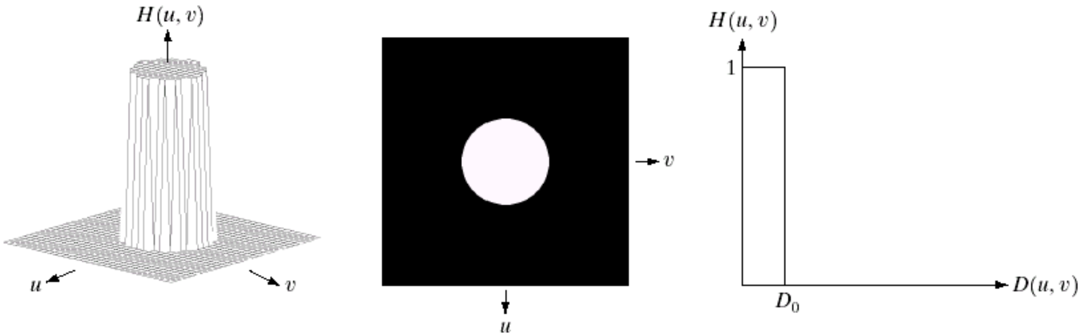
D(*u,v*): distance from the origin of Fourier transform

a b

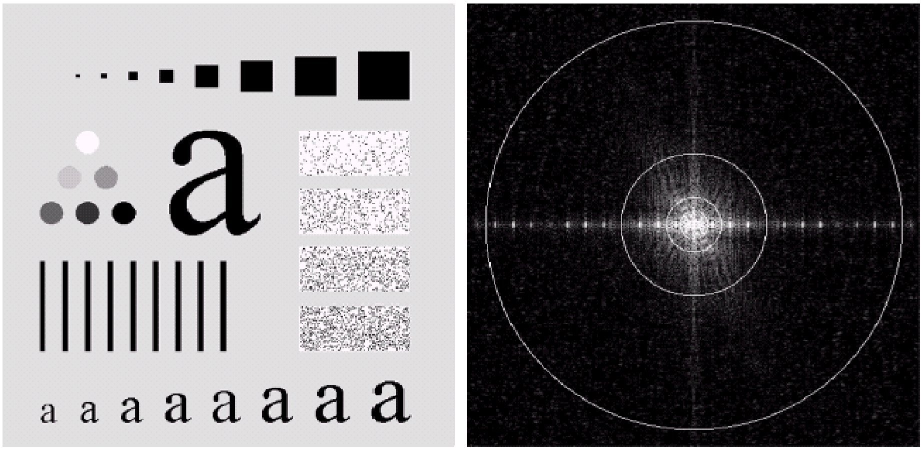
FIGURE 4.19
(a) Sample text of poor resolution (note broken characters in magnified view).
(b) Result of filtering with a GLPF (broken character segments were joined).



Ideal Low Pass Filters

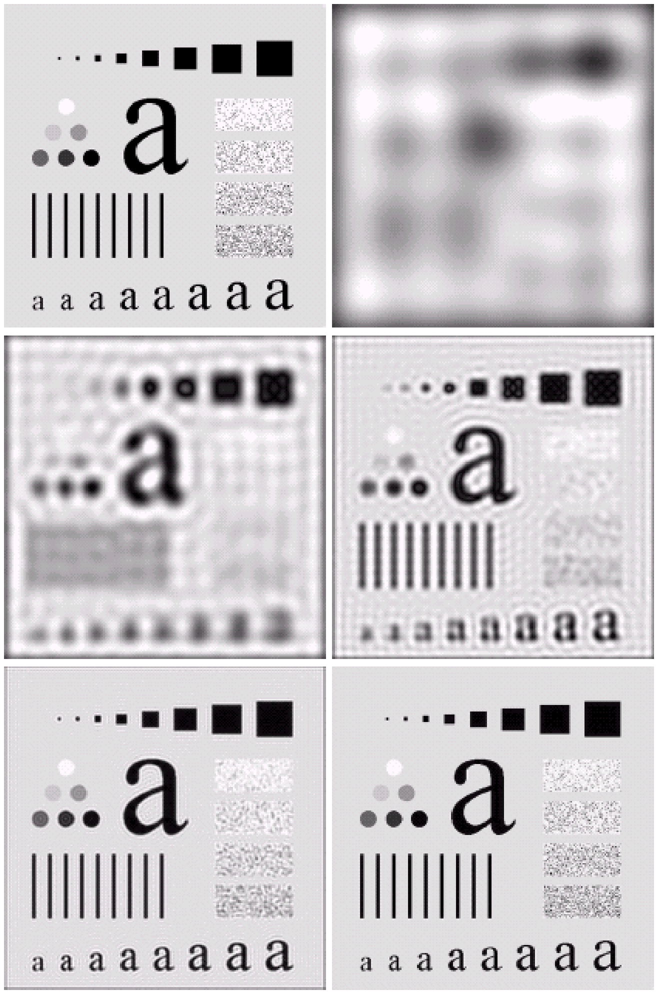


- The cut-off frequency D_o determines % power are filtered out.



- Image power as a function of distance from the origin of DFT (5, 15, 30, 80, 230)

Effects of Ideal Low Pass Filters



- Blurring can be modeled as the convolution of a high resolution (original) image with a low pass filter.

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.

Ringing and Blurring

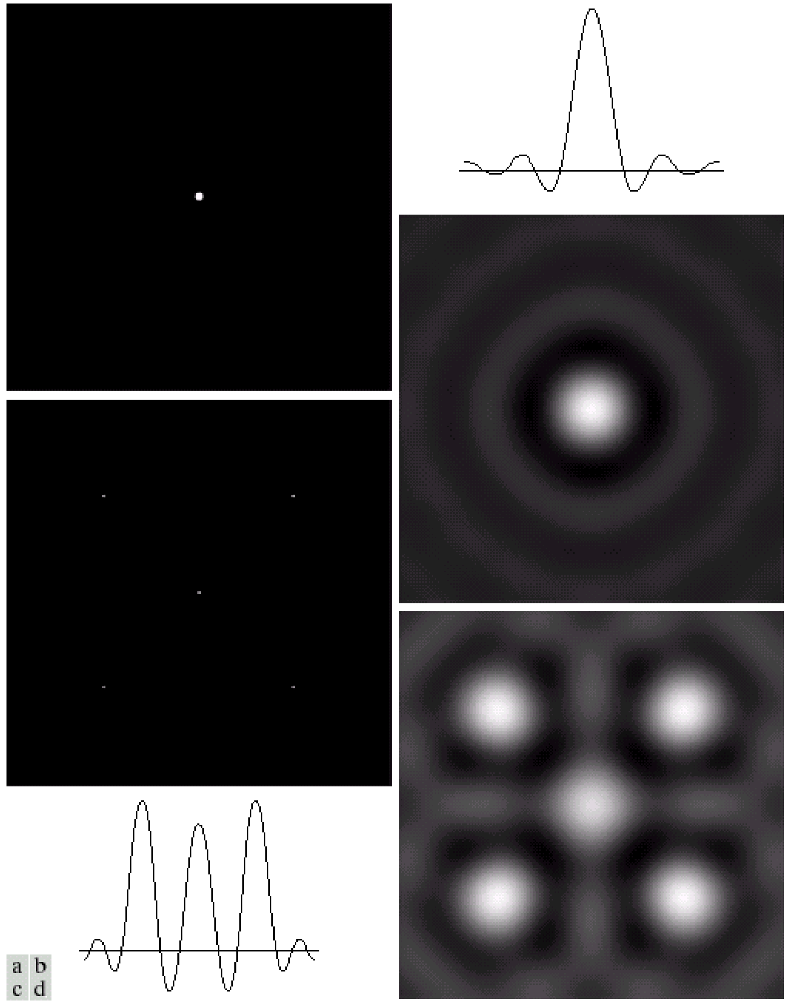
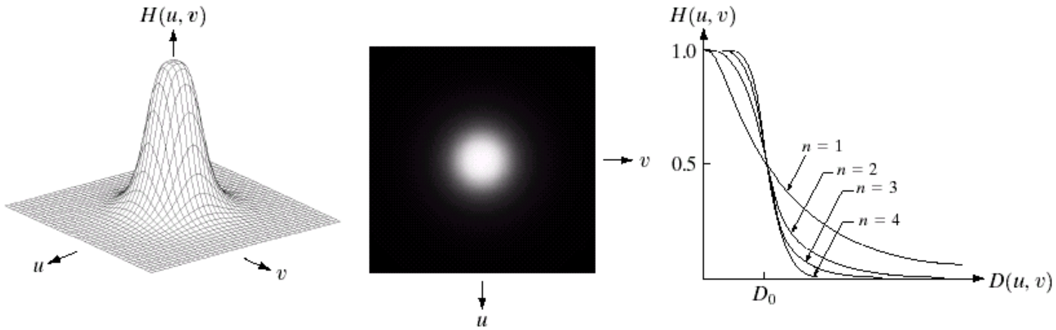
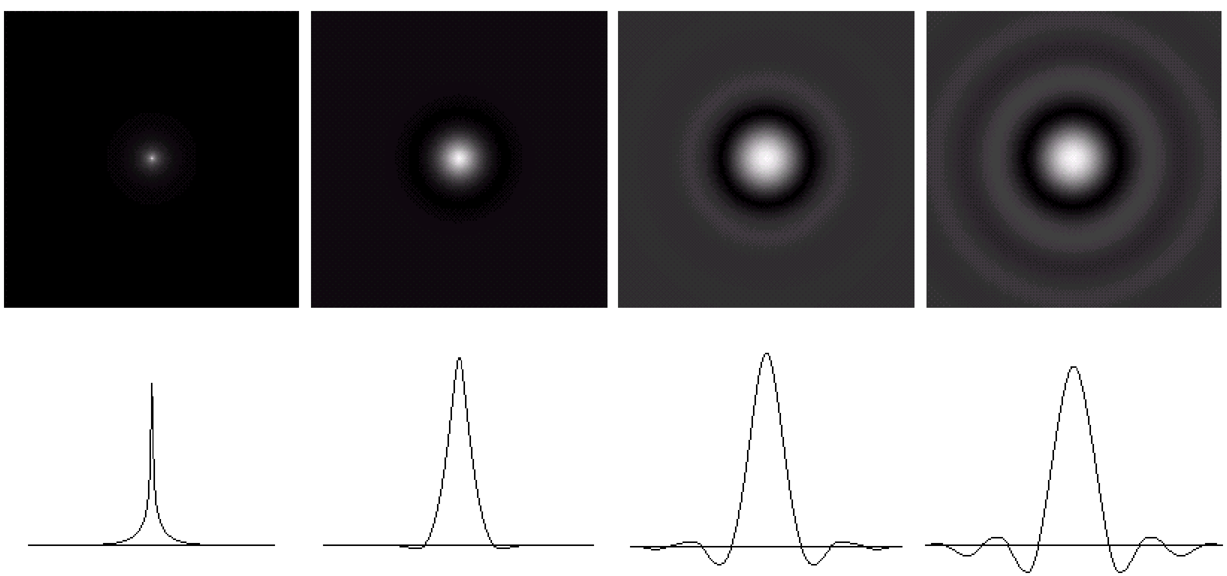


FIGURE 4.13 (a) A frequency-domain ILPF of radius 5. (b) Corresponding spatial filter (note the ringing). (c) Five impulses in the spatial domain, simulating the values of five pixels. (d) Convolution of (b) and (c) in the spatial domain.

Butterworth Low Pass Filters

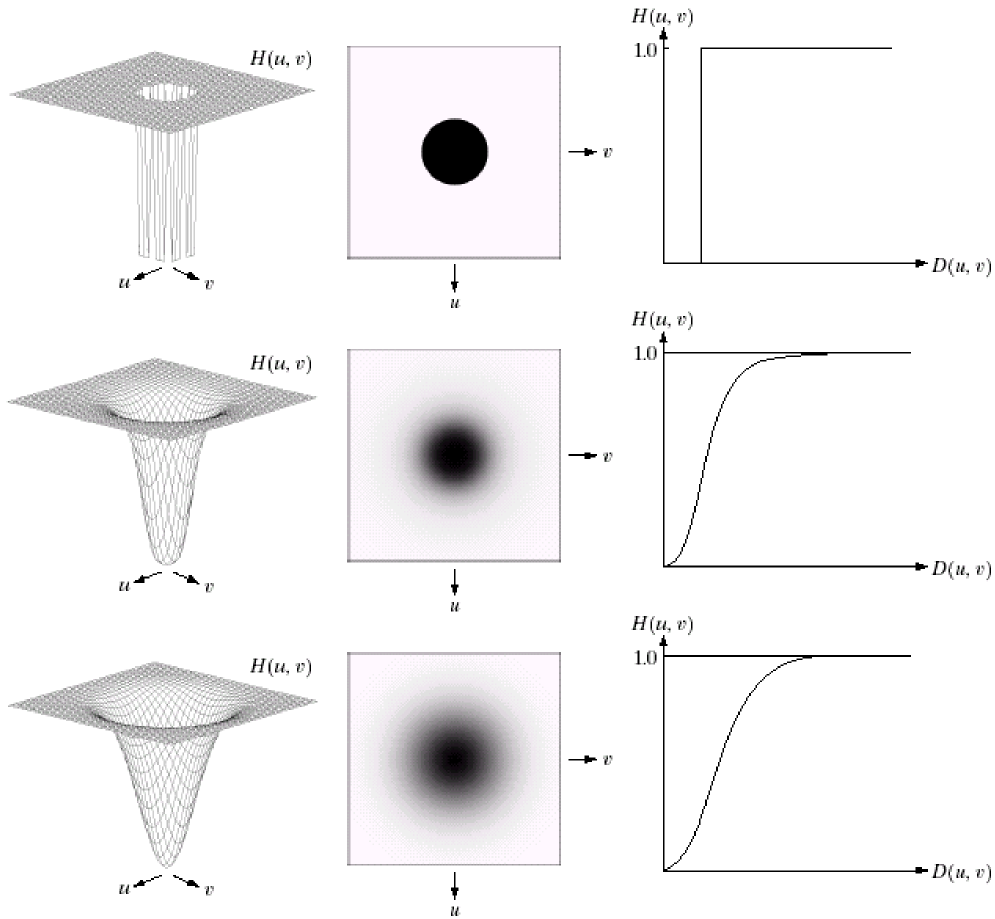


$$H(u,v)=\frac{1}{1+\left[D(u,v)/D_o\right]^{2n}}$$



UNIVERSITY OF WISCONSIN–MADISON

High Pass Filters



- Ideal high pass filter

$$H(u,v) = \begin{cases} 0 & \text{if } D(u,v) \le D_o \\ 1 & \text{otherwise.} \end{cases}$$

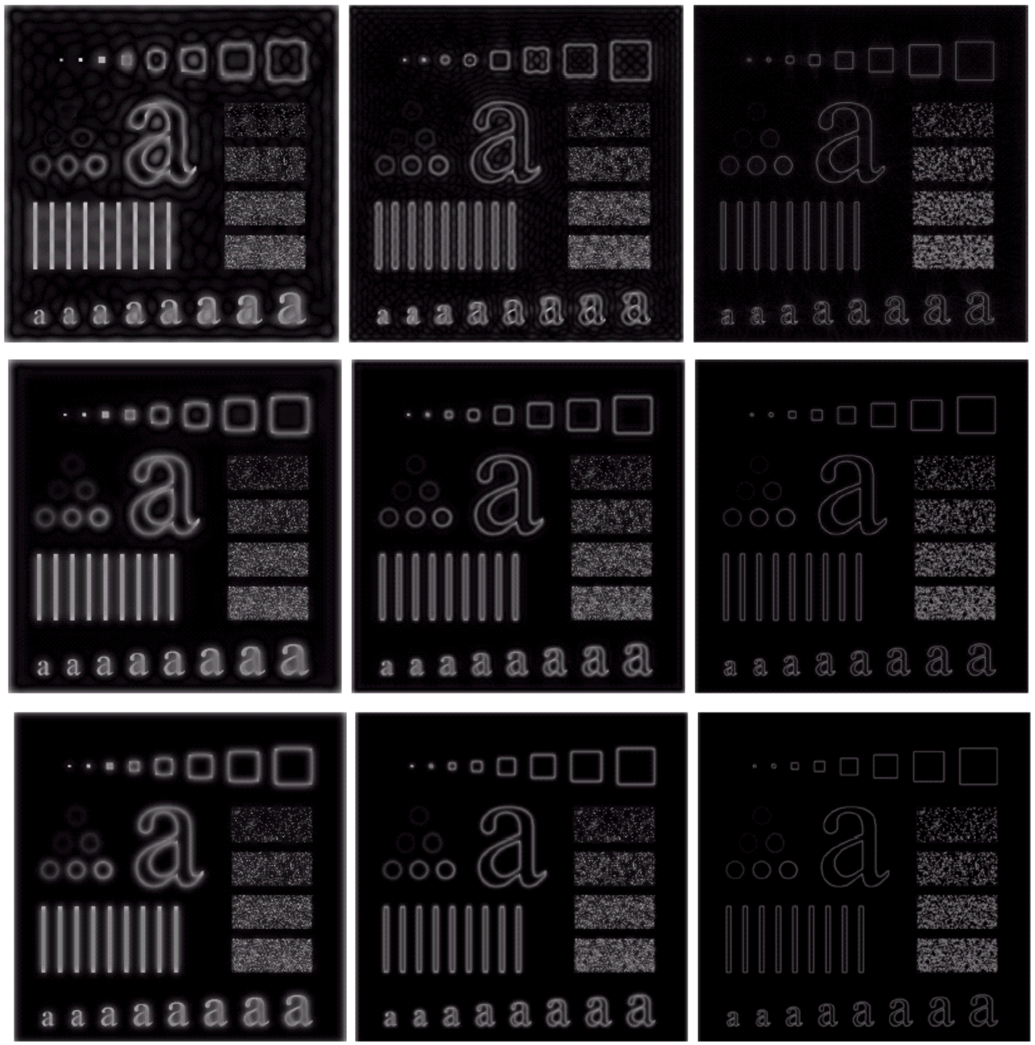
- Butterworth high pass filter

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

- Gaussian high pass filter

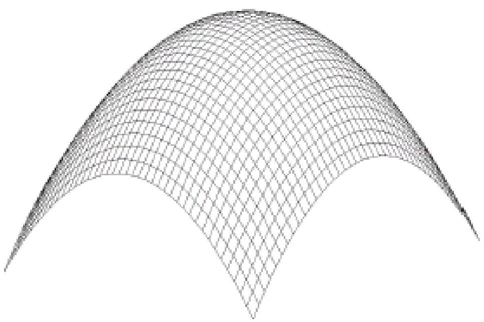
$$H(u,v) = 1 - \exp\left\{ -\frac{\left(D(u,v)\right)^2}{2D_0^2} \right\}$$

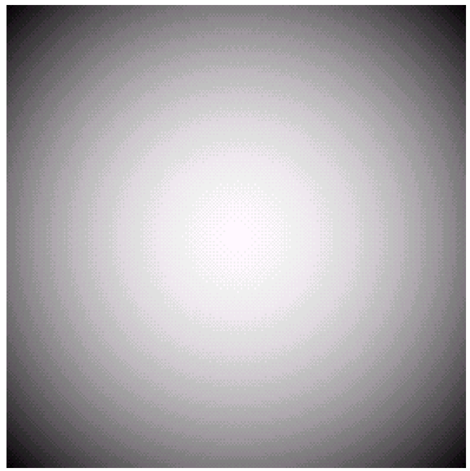
Applications of HPFs

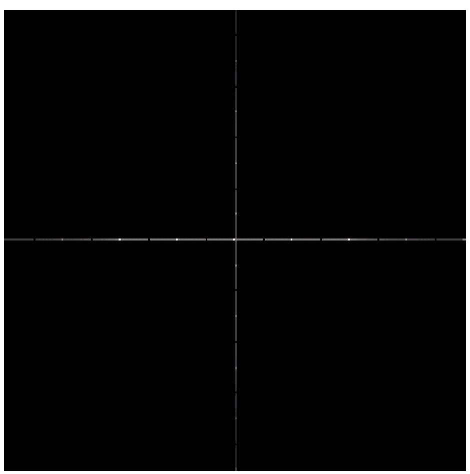


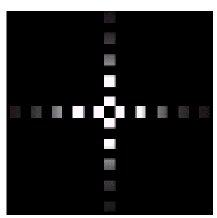
- Ideal HPF
  - D_0 = 15, 30, 80
- Butterworth HPF
  - n = 2,
  - D_0 = 15, 30, 80
- Gaussian HPF
  - D_0 = 15, 30, 80

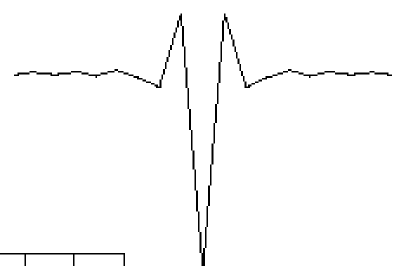
Laplacian HPF











0	1	0
1	−4	1
0	1	0

- 3D plots of the Laplacian operator,
- its 2D images,
- spatial domain response with center magnified, and
- Compared to the FIR mask approximation

$$\nabla^2 f(x,y) \Leftrightarrow -\left[(u-M/2)^2+(v-N/2)^2\right]\cdot F(u,v)$$

Properties of 2D DFT

Property	Expression(s)
Fourier transform	$F(u,v) = \frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)e^{-j2\pi(ux/M+vy/N)}$
Inverse Fourier transform	$f(x,y) = \sum_{u=0}^{M-1}\sum_{v=0}^{N-1} F(u,v)e^{j2\pi(ux/M+vy/N)}$
Polar representation	$F(u,v) = \lvert F(u,v)\rvert e^{-j\phi(u,v)}$
Spectrum	$\lvert F(u,v)\rvert = \left[R^2(u,v) + I^2(u,v)\right]^{1/2}$, $R = \text{Real}(F)$ and $I = \text{Imag}(F)$
Phase angle	$\phi(u,v) = \tan^{-1}\left[\frac{I(u,v)}{R(u,v)}\right]$
Power spectrum	$P(u,v) = \lvert F(u,v)\rvert^2$
Average value	$\bar{f}(x,y) = F(0,0) = \frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)$
Translation	$f(x,y)e^{j2\pi(u_0x/M+v_0y/N)} \Leftrightarrow F(u-u_0, v-v_0)$ $f(x-x_0, y-y_0) \Leftrightarrow F(u,v)e^{-j2\pi(ux_0/M+vy_0/N)}$ When $x_0 = u_0 = M/2$ and $y_0 = v_0 = N/2$, then $f(x,y)(-1)^{x+y} \Leftrightarrow F(u-M/2, v-N/2)$ $f(x-M/2, y-N/2) \Leftrightarrow F(u,v)(-1)^{u+v}$

Properties of 2D DFT

Conjugate symmetry	$F(u, v) = F^*(-u, -v)$ $\lvert F(u, v)\rvert = \lvert F(-u, -v)\rvert$
Differentiation	$\dfrac{\partial^n f(x, y)}{\partial x^n} \Leftrightarrow (ju)^n F(u, v)$ $(-jx)^n f(x, y) \Leftrightarrow \dfrac{\partial^n F(u, v)}{\partial u^n}$
Laplacian	$\nabla^2 f(x, y) \Leftrightarrow -(u^2 + v^2)F(u, v)$
Distributivity	$\Im[f_1(x, y) + f_2(x, y)] = \Im[f_1(x, y)] + \Im[f_2(x, y)]$ $\Im[f_1(x, y) \cdot f_2(x, y)] \neq \Im[f_1(x, y)] \cdot \Im[f_2(x, y)]$
Scaling	$af(x, y) \Leftrightarrow aF(u, v), f(ax, by) \Leftrightarrow \dfrac{1}{\lvert ab\rvert} F(u/a, v/b)$
Rotation	$x = r\cos\theta \quad y = r\sin\theta \quad u = \omega\cos\varphi \quad v = \omega\sin\varphi$ $f(r, \theta + \theta_0) \Leftrightarrow F(\omega, \varphi + \theta_0)$
Periodicity	$F(u, v) = F(u + M, v) = F(u, v + N) = F(u + M, v + N)$ $f(x, y) = f(x + M, y) = f(x, y + N) = f(x + M, y + N)$
Separability	See Eqs. (4.6-14) and (4.6-15). Separability implies that we can compute the 2-D transform of an image by first computing 1-D transforms along each row of the image, and then computing a 1-D transform along each column of this intermediate result. The reverse, columns and then rows, yields the same result.