Intensity Transformation and Spatial Filtering

Spatial Domain vs. Transform Domain

- Spatial domain

  image plane itself, directly process the intensity values of the image plane

- Transform domain

  process the transform coefficients, not directly process the intensity values of the image plane

Spatial Domain Process

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

$f(x,y)$: input image

$g(x,y)$: output image

T : an operator on f defined over a neighborhood of point (x,y)

Spatial Domain Process

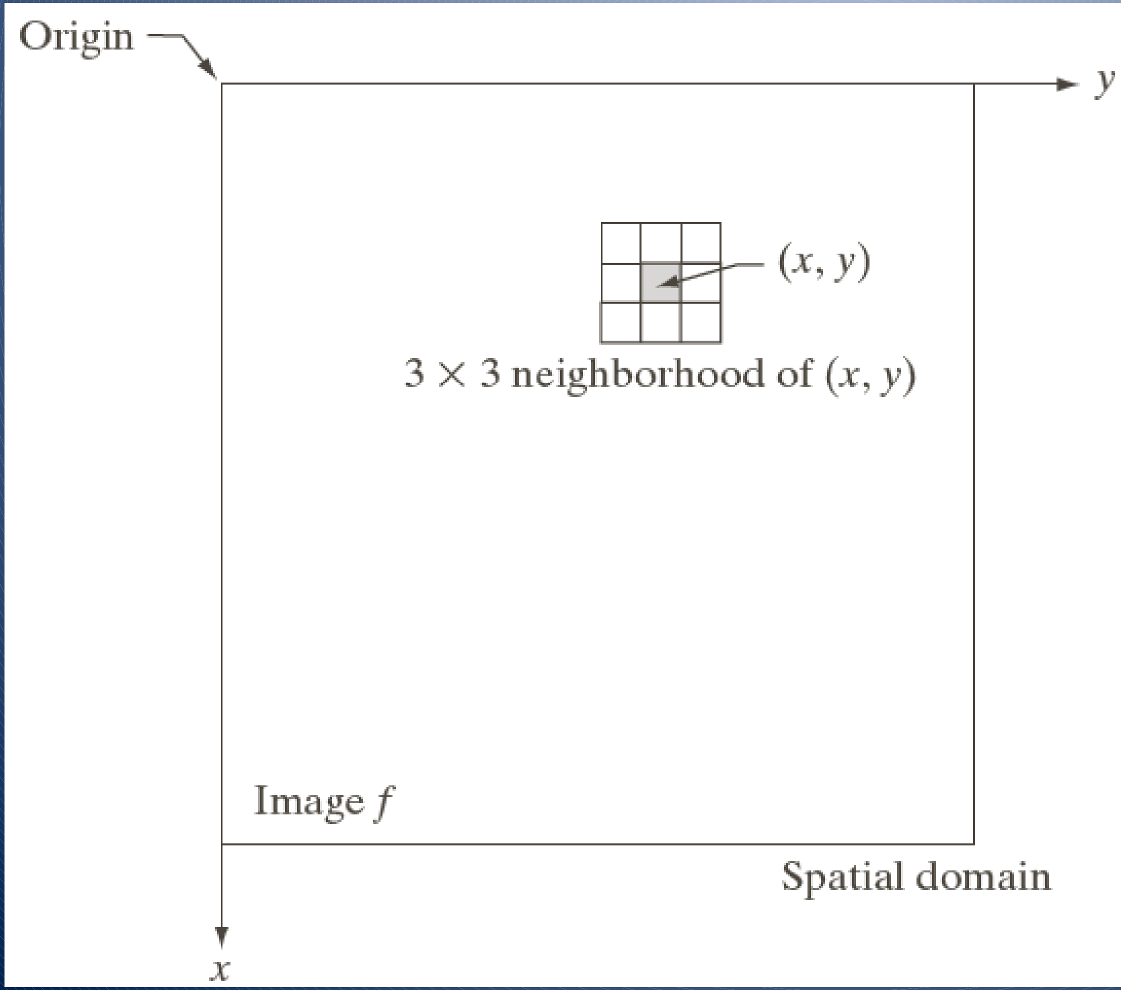
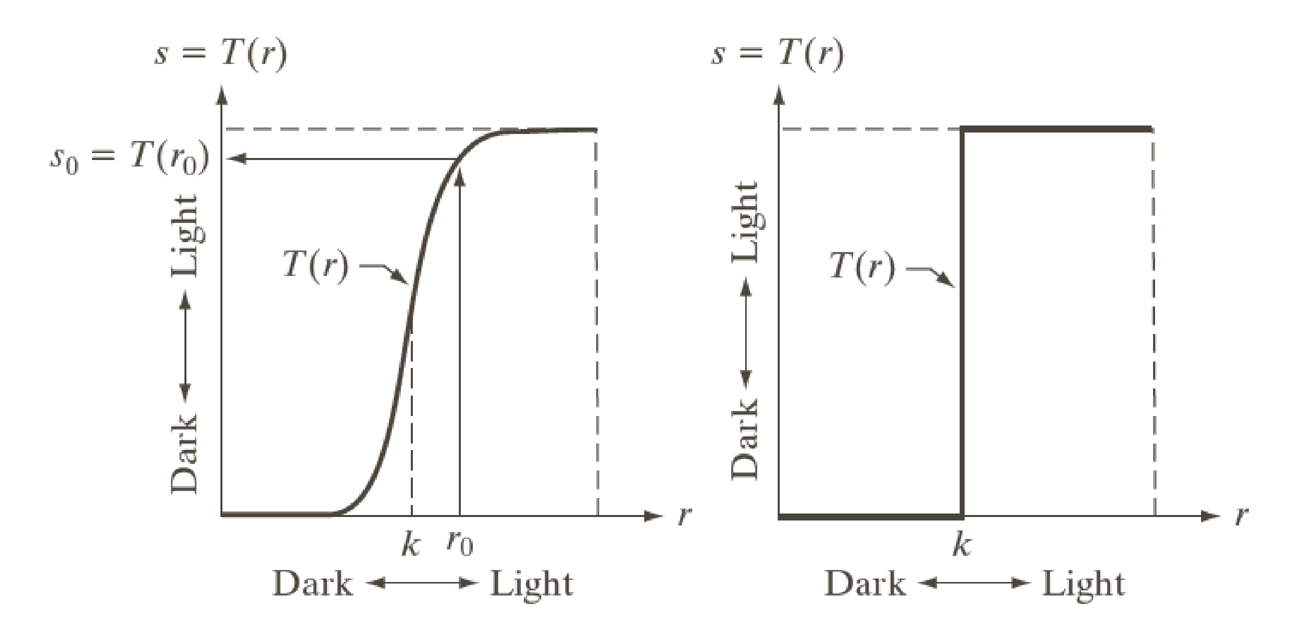


FIGURE 3.1
A 3 × 3 neighborhood about a point (x, y) in an image in the spatial domain. The neighborhood is moved from pixel to pixel in the image to generate an output image.

Spatial Domain Process

Intensity transformation function

$$s = T(r)$$



a b

FIGURE 3.2 Intensity transformation functions. (a) Contrast-stretching function. (b) Thresholding function.

Some Basic Intensity Transformation Functions

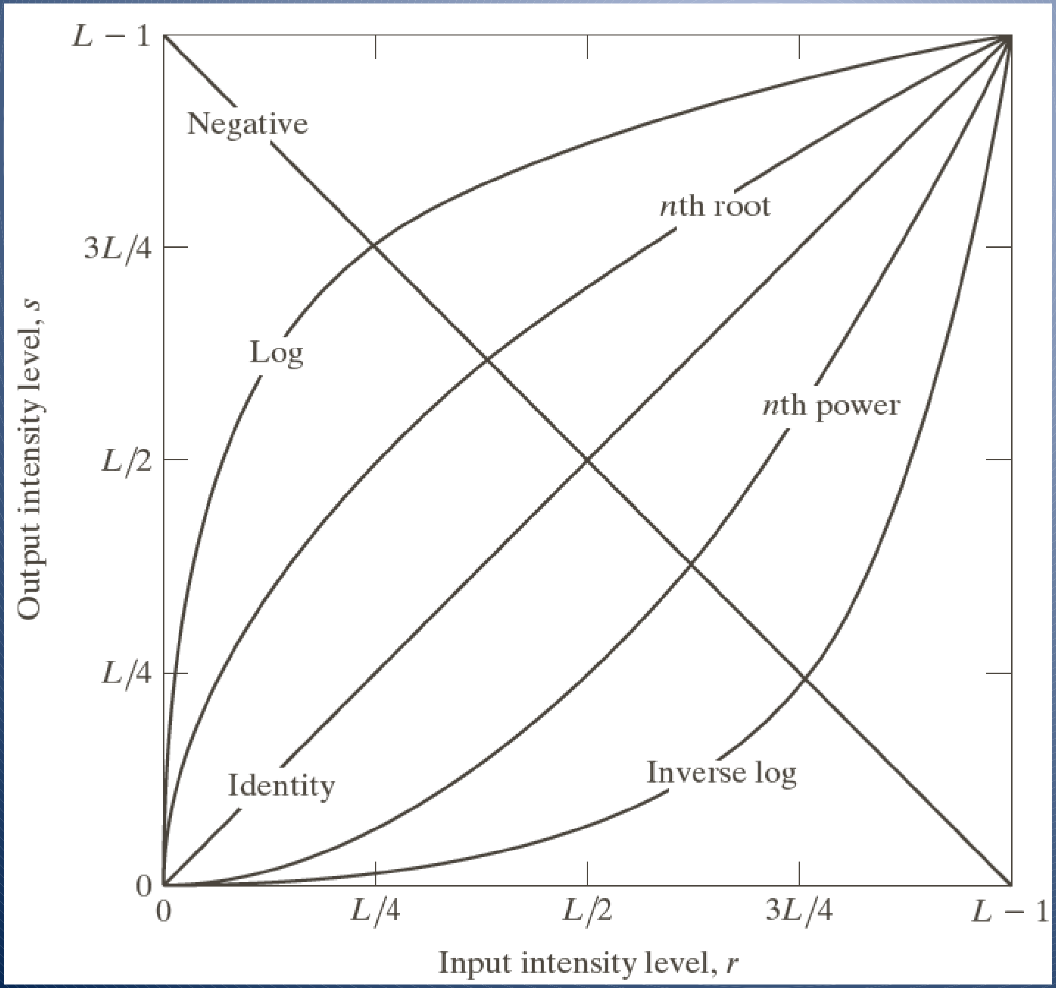


FIGURE 3.3 Some basic intensity transformation functions. All curves were scaled to fit in the range shown.

Image Negatives

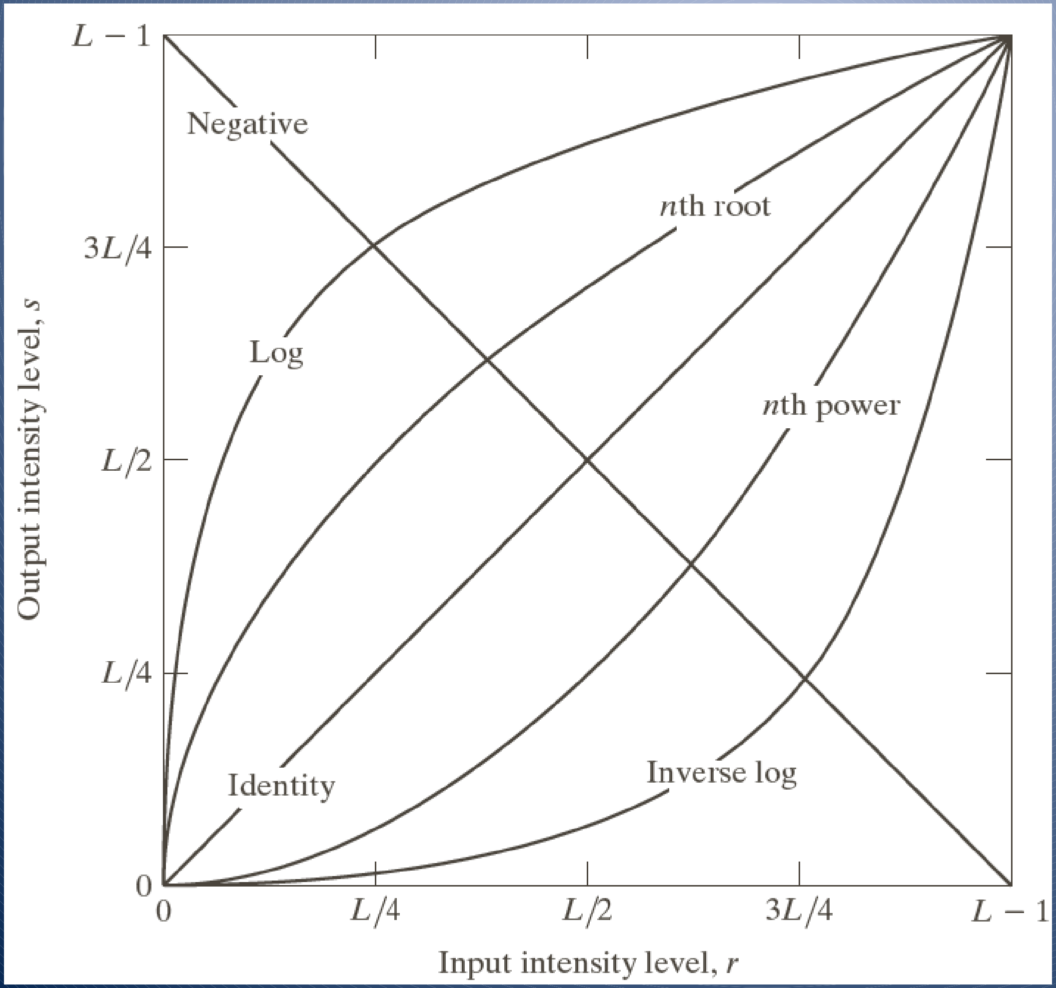


Image negatives

$$s = L - 1 - r$$

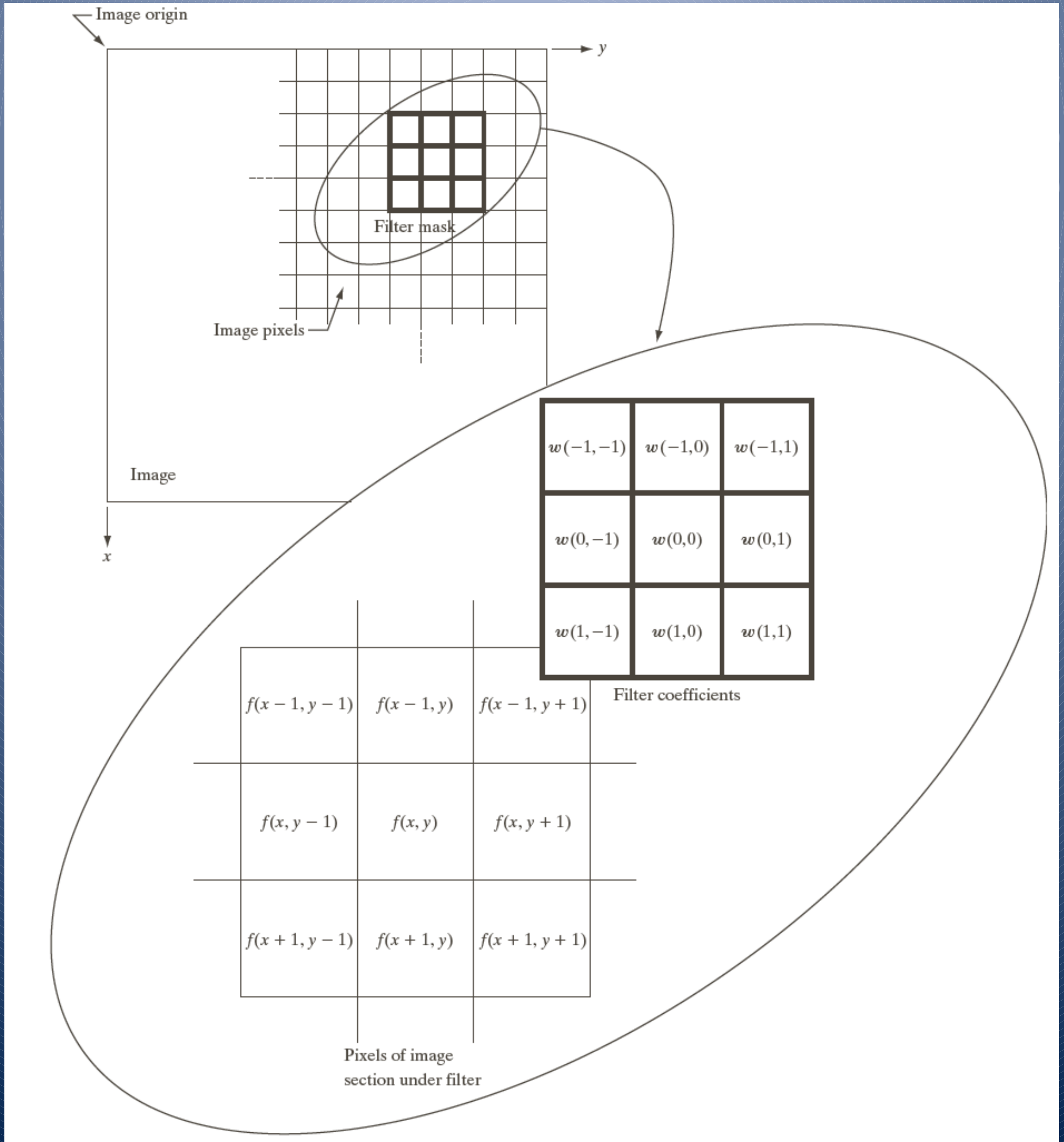
Spatial Filtering

A spatial filter consists of (a) **a neighborhood**, and (b) **a predefined operation**

Linear spatial filtering of an image of size MxN with a filter of size mxn is given by the expression

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

Spatial Filtering



Spatial Correlation

The correlation of a filter $w(x, y)$ of size $m \times n$ with an image $f(x, y)$, denoted as $w(x, y) \star f(x, y)$

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

Spatial Convolution

The convolution of a filter $w(x, y)$ of size $m \times n$ with an image $f(x, y)$, denoted as $w(x, y) \star f(x, y)$

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

Origin $f(x, y)$

$$\begin{array}{ccccc} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{array}$$

$w(x, y)$

$$\begin{array}{ccc} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{array}$$

(a)

[redacted]

FIGURE 3.30 Correlation (middle row) and convolution (last row) of a 2-D filter with a 2-D discrete, unit impulse. The 0s are shown in gray to simplify visual analysis.

Smoothing Spatial Filters

Smoothing filters are used for blurring and for noise reduction

Blurring is used in removal of small details and bridging of small gaps in lines or curves

Smoothing spatial filters include linear filters and nonlinear filters.

Spatial Smoothing Linear Filters

The general implementation for filtering an M $\times$N image with a weighted averaging filter of size m $\times$n is given

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

where $m=2a+1, \quad n=2b+1.$

Two Smoothing Averaging Filter Masks

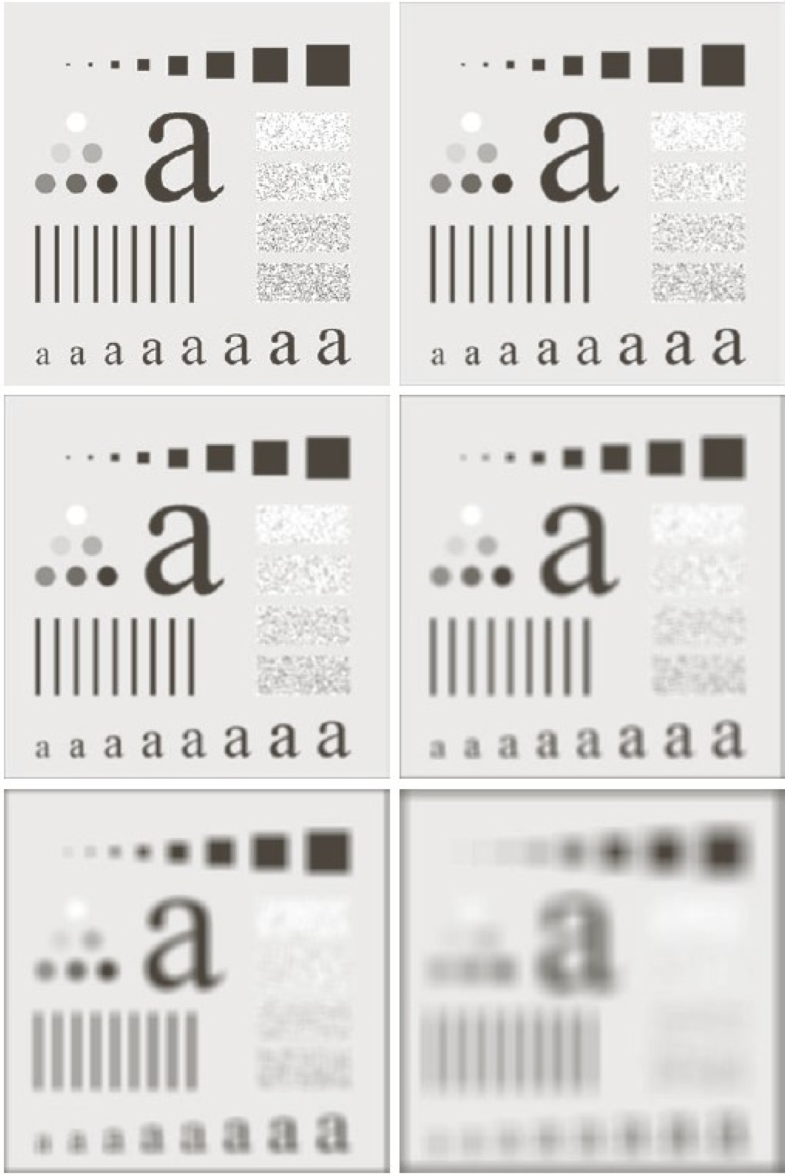
$$\frac{1}{9} \times \begin{array}{|c|c|c|}\hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array} \qquad \frac{1}{16} \times \begin{array}{|c|c|c|}\hline 1 & 2 & 1 \\ \hline 2 & 4 & 2 \\ \hline 1 & 2 & 1 \\ \hline \end{array}$$

a b

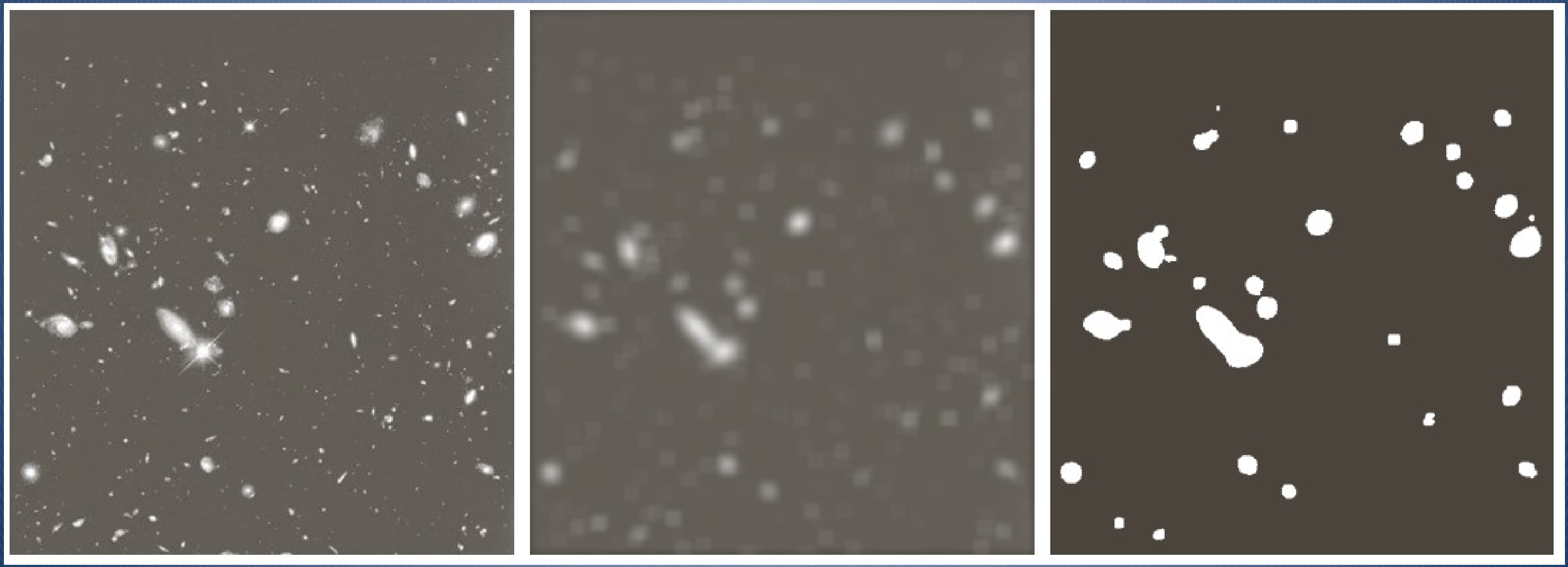
FIGURE 3.32 Two 3×3 smoothing (averaging) filter masks. The constant multiplier in front of each mask is equal to 1 divided by the sum of the values of its coefficients, as is required to compute an average.

FIGURE 3.33 (a) Original image, of size 500 × 500 pixels. (b)–(f) Results of smoothing with square averaging filter masks of sizes $m = 3, 5, 9, 15$, 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 intensity 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.

a b
c d
e f



Example: Gross Representation of Objects



a b c

FIGURE 3.34 (a) Image of size 528 × 485 pixels from the Hubble Space Telescope. (b) Image filtered with a 15 × 15 averaging mask. (c) Result of thresholding (b). (Original image courtesy of NASA.)

Order-statistic (Nonlinear) Filters

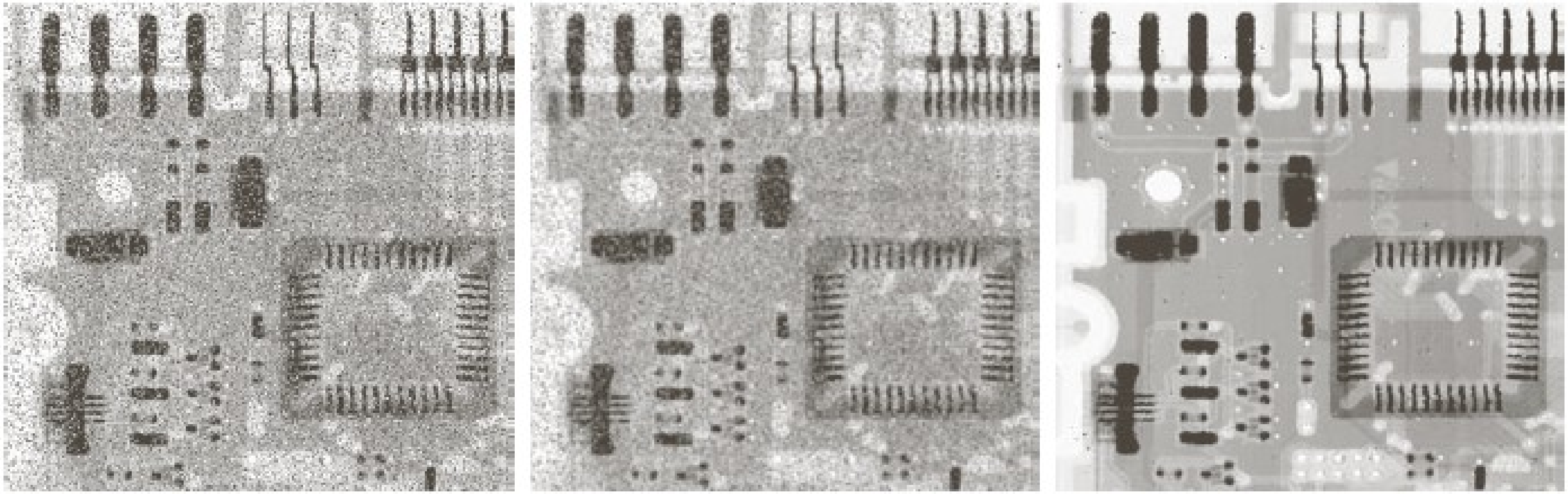
— Nonlinear

— Based on ordering (ranking) the pixels contained in the filter mask

— Replacing the value of the center pixel with the value determined by the ranking result

E.g., median filter, max filter, min filter

Example: Use of Median Filtering for Noise Reduction



a b c

FIGURE 3.35 (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.)

Sharpening Spatial Filters

- Foundation
- Laplacian Operator
- Unsharp Masking and Highboost Filtering
- Using First-Order Derivatives for Nonlinear Image Sharpening
  — The Gradient

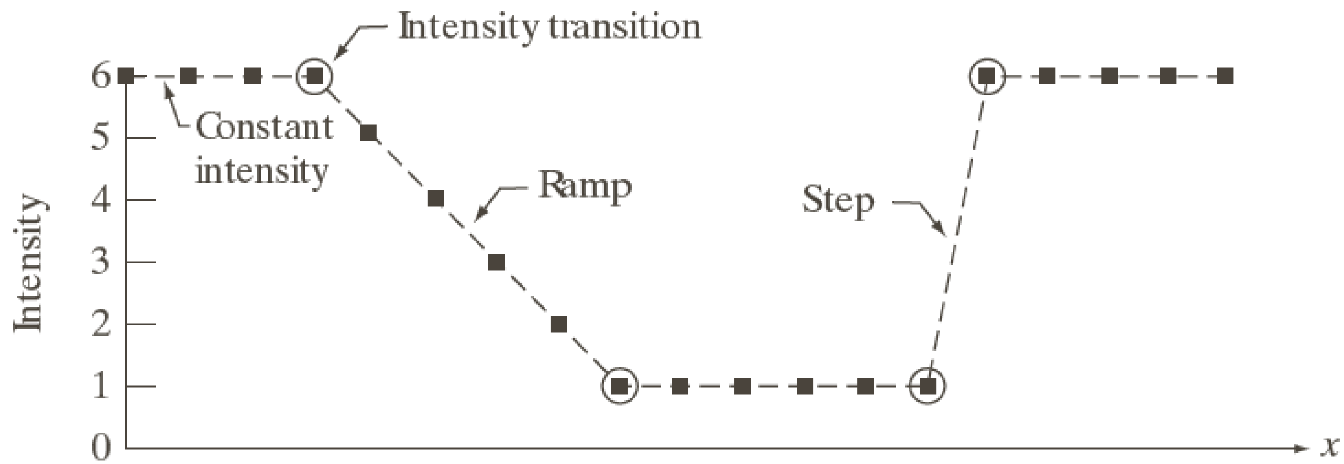
Sharpening Spatial Filters: Foundation

- The first-order derivative of a one-dimensional function f(x) is the difference

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

- The second-order derivative of f(x) as the difference

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



a
b
c

FIGURE 3.36 Illustration of the first and second derivatives of a 1-D digital function representing a section of a horizontal intensity profile from an image. In (a) and (c) data points are joined by dashed lines as a visualization aid.

Sharpening Spatial Filters: Laplace Operator

The second-order isotropic derivative operator is the Laplacian for a function (image) f(x,y)

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

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

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

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

Sharpening Spatial Filters: Laplace Operator

(a)

0	1	0
1	−4	1
0	1	0

(b)

1	1	1
1	−8	1
1	1	1

(c)

0	−1	0
−1	4	−1
0	−1	0

(d)

−1	−1	−1
−1	8	−1
−1	−1	−1

a	b
c	d

FIGURE 3.37
(a) Filter mask used to implement Eq. (3.6-6). (b) Mask used to implement an extension of this equation that includes the diagonal terms. (c) and (d) Two other implementations of the Laplacian found frequently in practice.

Sharpening Spatial Filters: Laplace Operator

Image sharpening in the way of using the Laplacian:

$$g(x,y) = f(x,y) + c\left[\nabla^2 f(x,y)\right]$$

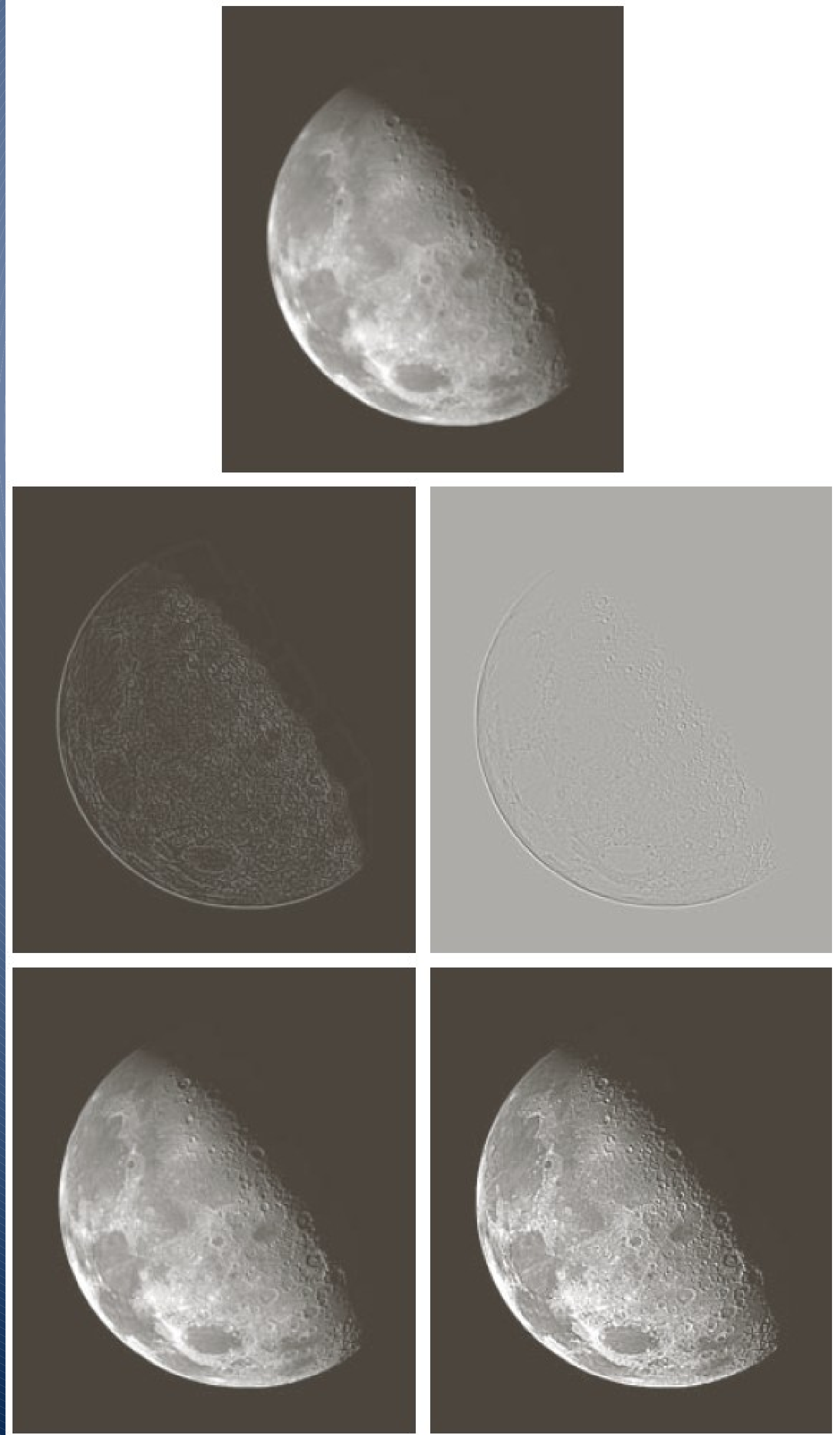
where,

$f(x,y)$ is input image,

$g(x,y)$ is sharpenend images,

c =-1 if $\nabla^2 f(x,y)$ corresponding to Fig. 3.37(a) or (b)

and c =1 if either of the other two filters is used.



a
b c
d e

FIGURE 3.38
(a) Blurred image of the North Pole of the moon. (b) Laplacian without scaling. (c) Laplacian with scaling. (d) Image sharpened using the mask in Fig. 3.37(a). (e) Result of using the mask in Fig. 3.37(b). (Original image courtesy of NASA.)

Unsharp Masking and Highboost Filtering

- Unsharp masking

  Sharpen images consists of subtracting an unsharp (smoothed) version of an image from the original image

  e.g., printing and publishing industry

- Steps

  1. Blur the original image

  2. Subtract the blurred image from the original

  3. Add the mask to the original

Unsharp Masking and Highboost Filtering

Let $\overline{f}(x, y)$ denote the blurred image, unsharp masking is

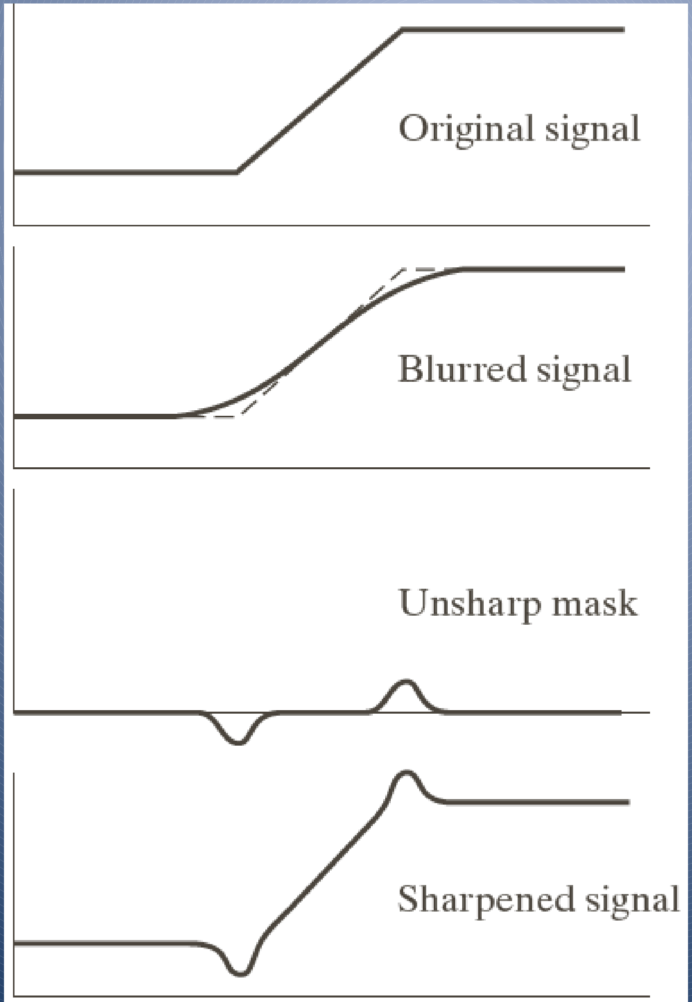
$$g_{mask}(x, y) = f(x, y) - \overline{f}(x, y)$$

Then add a weighted portion of the mask back to the original

$$g(x, y) = f(x, y) + k * g_{mask}(x, y) \qquad k \geq 0$$

when $k > 1$, the process is referred to as highboost filtering.

Unsharp Masking: Demo



a
b
c
d

FIGURE 3.39 1-D illustration of the mechanics of unsharp masking. (a) Original signal. (b) Blurred signal with original shown dashed for reference. (c) Unsharp mask. (d) Sharpened signal, obtained by adding (c) to (a).

Unsharp Masking and Highboost Filtering: Example

a
b
c
d
e

FIGURE 3.40 (a) Original image. (b) Result of blurring with a Gaussian filter. (c) Unsharp mask. (d) Result of using unsharp masking. (e) Result of using highboost filtering.

Image Sharpening based on First-Order Derivatives

For function $f(x, y)$, the gradient of f at coordinates (x, y) is defined as

$$\nabla f \equiv \text{grad}(f) \equiv \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \dfrac{\partial f}{\partial x} \\ \dfrac{\partial f}{\partial y} \end{bmatrix}$$

The *magnitude* of vector ∇f, denoted as $M(x, y)$

Gradient Image $$M(x, y) = \text{mag}(\nabla f) = \sqrt{g_x^{\ 2} + g_y^{\ 2}}$$

Image Sharpening based on First-Order Derivatives

The *magnitude* of vector ∇f, denoted as $M(x, y)$

$$M(x, y) = \text{mag}(\nabla f) = \sqrt{g_x^2 + g_y^2}$$

$$M(x, y) \approx |g_x| + |g_y|$$

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

$$M(x, y) = |z_8 - z_5| + |z_6 - z_5|$$

Image Sharpening based on First-Order Derivatives

Roberts Cross-gradient Operators

$$M(x,y) \approx |z_9 - z_5| + |z_8 - z_6|$$

Sobel Operators

$$M(x,y) \approx |(z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)|$$
$$+ |(z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)|$$

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

Image Sharpening based on First-Order Derivatives

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

−1	0
0	1

0	−1
1	0

−1	−2	−1
0	0	0
1	2	1

−1	0	1
−2	0	2
−1	0	1

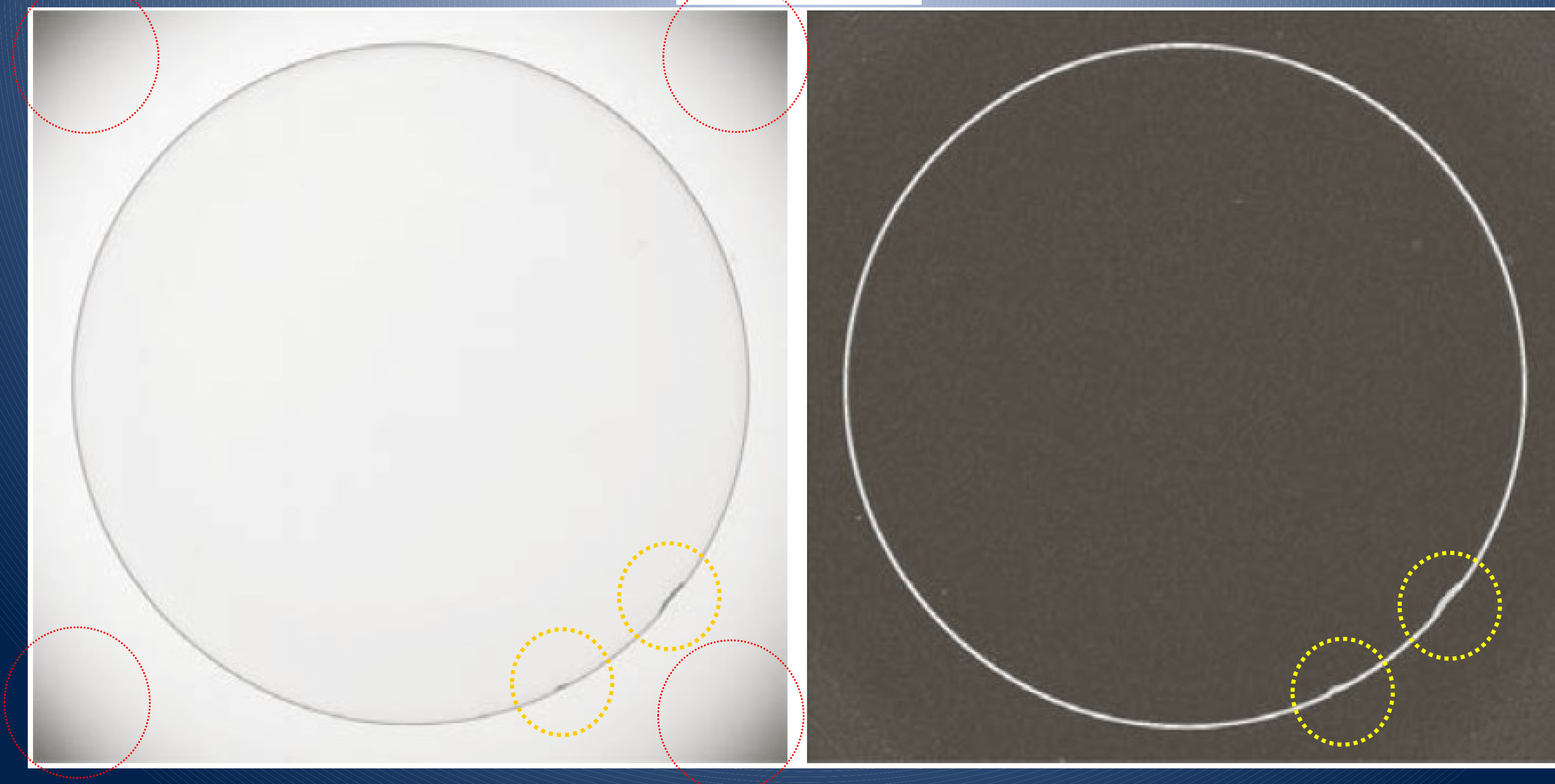
a
b c
d e

FIGURE 3.41 A 3×3 region of an image (the zs are intensity values). (b)–(c) Roberts cross gradient operators. (d)–(e) Sobel operators. All the mask coefficients sum to zero, as expected of a derivative operator.

Example

a b

FIGURE 3.42 (a) Optical image of contact lens (note defects on the boundary at 4 and 5 o'clock). (b) Sobel gradient. (Original image courtesy of Pete Sites, Perceptics Corporation.)



Example:

Combining Spatial Enhancement Methods

Goal:

Enhance the image by sharpening it and by bringing out more of the skeletal detail

a b
c d

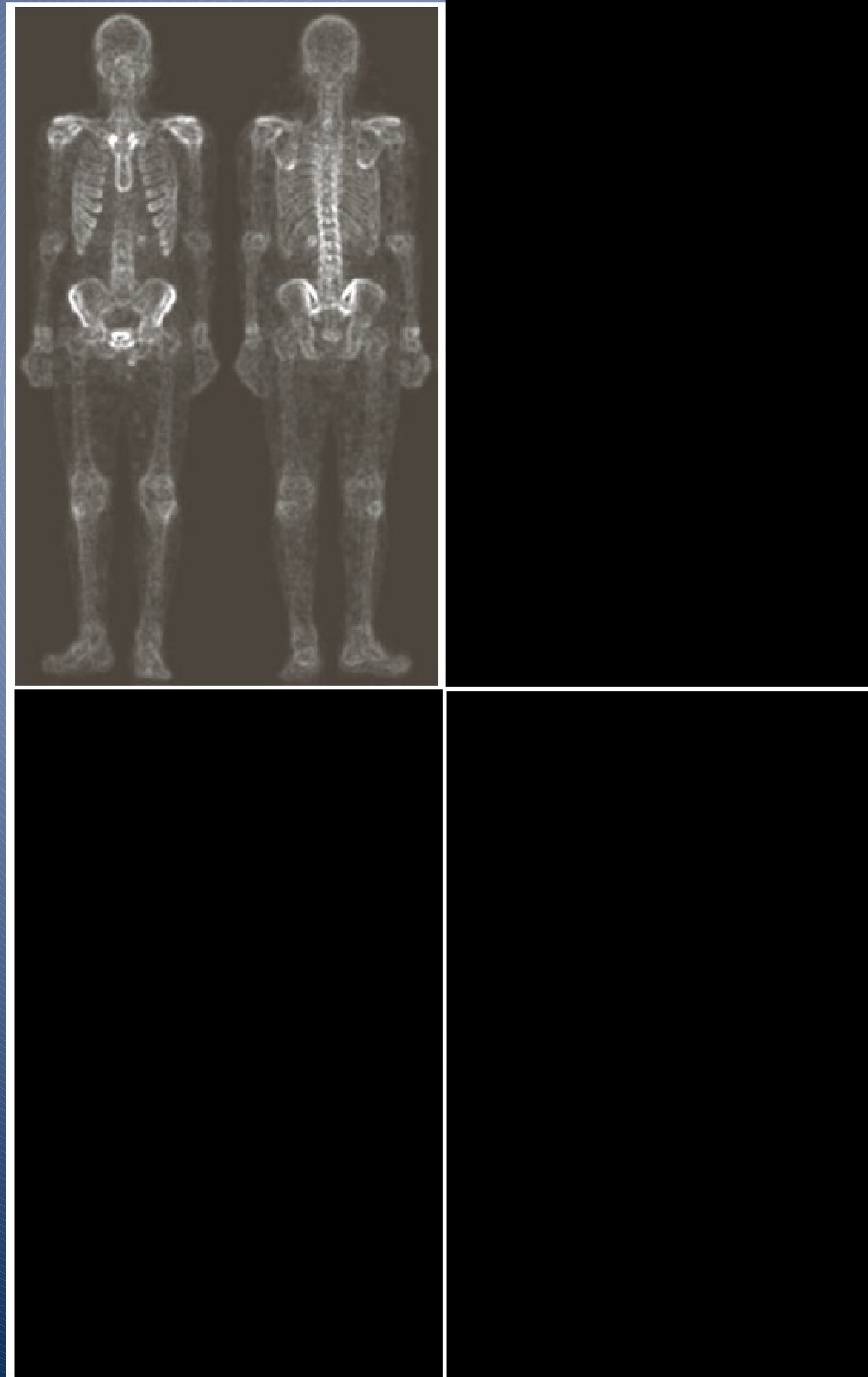
FIGURE 3.43 (a) Image of whole body bone scan. (b) Laplacian of (a). (c) Sharpened image obtained by adding (a) and (b). (d) Sobel gradient of (a).

Example:

Combining Spatial Enhancement Methods

Goal:

Enhance the image by sharpening it and by bringing out more of the skeletal detail



e f
g h

FIGURE 3.43 *(Continued)* (e) Sobel image smoothed with a 5×5 averaging filter. (f) Mask image formed by the product of (c) and (e). (g) Sharpened image obtained by the sum of (a) and (f). (h) Final result obtained by applying a power-law transformation to (g). Compare (g) and (h) with (a). (Original image courtesy of G.E. Medical Systems.)

Image Enhancement - Frequency Domain

- Low Pass Filter
- High Pass Filter
- Band pass Filter
- Blurring
- Sharpening

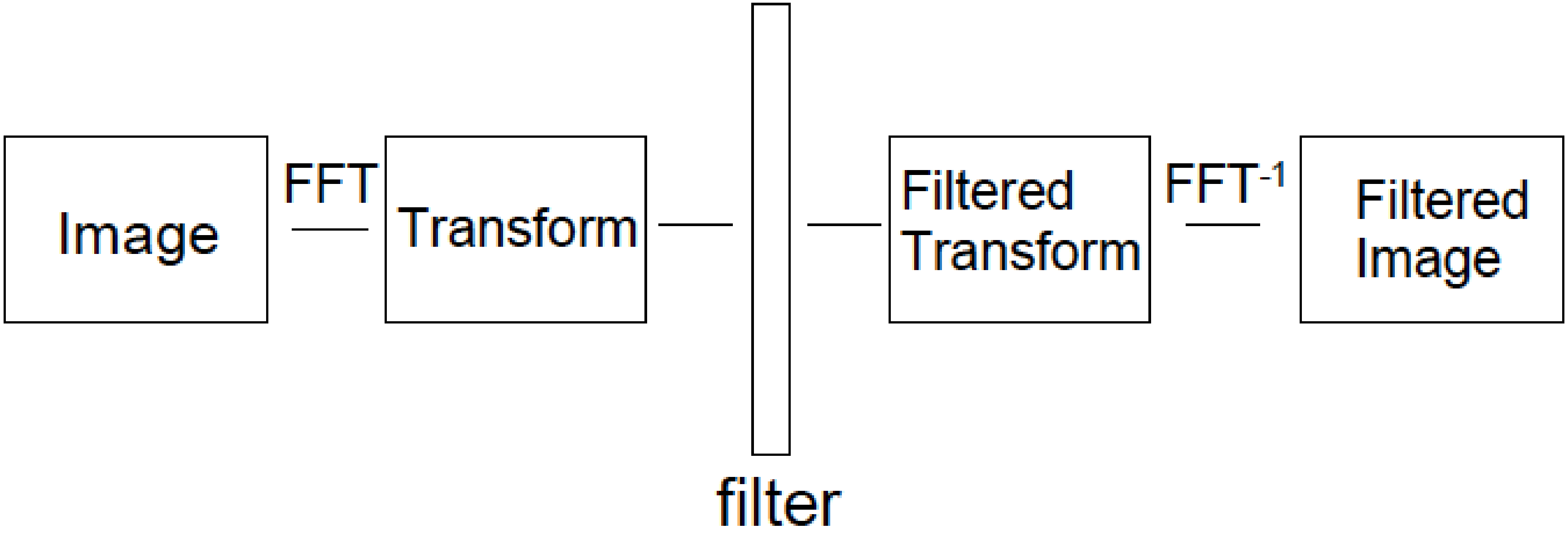
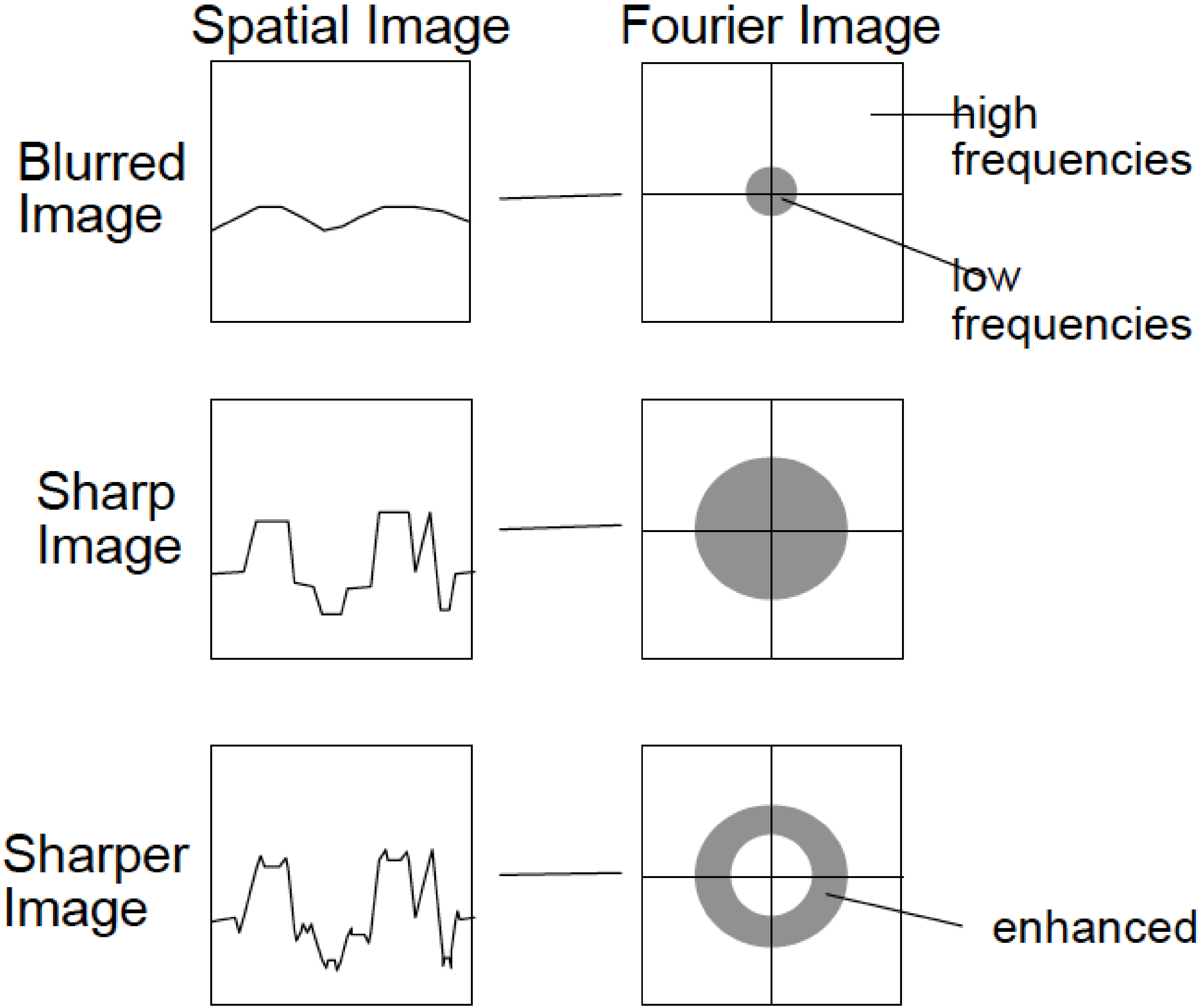


Image
FFT
Transform
filter
Filtered Transform
FFT-1
Filtered Image

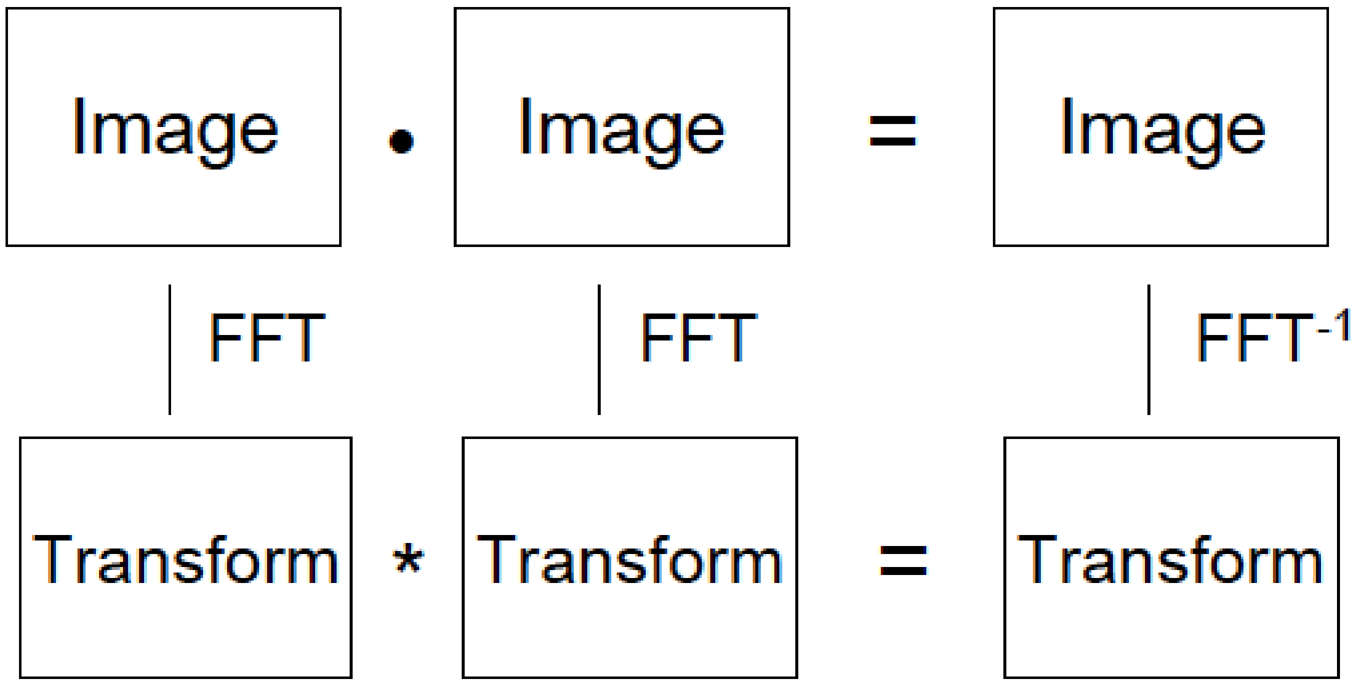

Spatial Image
Fourier Image
Blurred Image
high frequencies
low frequencies
Sharp Image
Sharper Image
enhanced

Convolution Theorem

$$\mathit{F}(\mathrm{I1}) * \mathit{F}(\mathrm{I2}) = \mathit{F}(\mathrm{I1} \bullet \mathrm{I2})$$

$$\begin{tikzcd}
\boxed{\text{Image}} \arrow[d, dash, "\text{FFT}"] & * & \boxed{\text{Image}} \arrow[d, dash, "\text{FFT}"] & = & \boxed{\text{Image}} \arrow[d, dash, "\text{FFT}^{-1}"] \\
\boxed{\text{Transform}} & \bullet & \boxed{\text{Transform}} & = & \boxed{\text{Transform}}
\end{tikzcd}$$

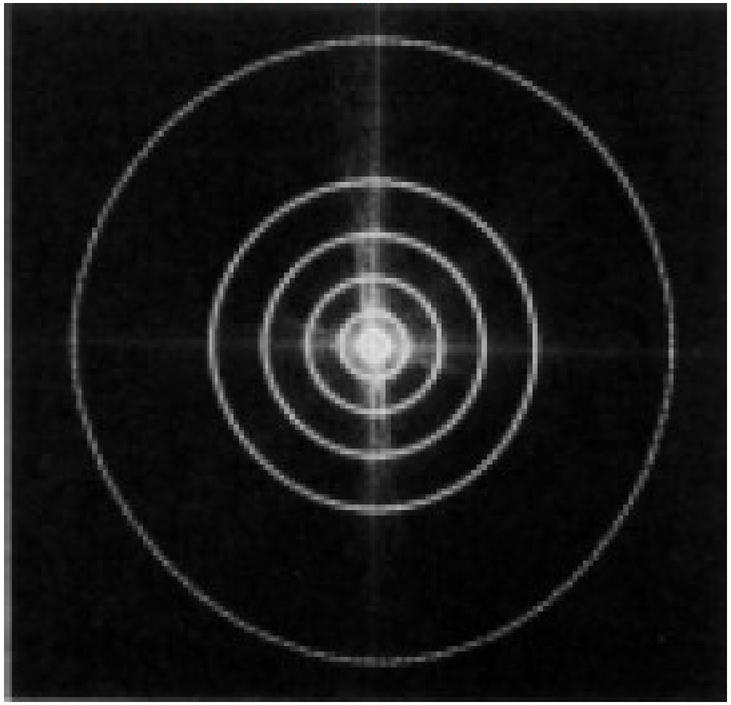
$$\boxed{F(I1) \bullet F(I2) = F(I1 * I2)}$$



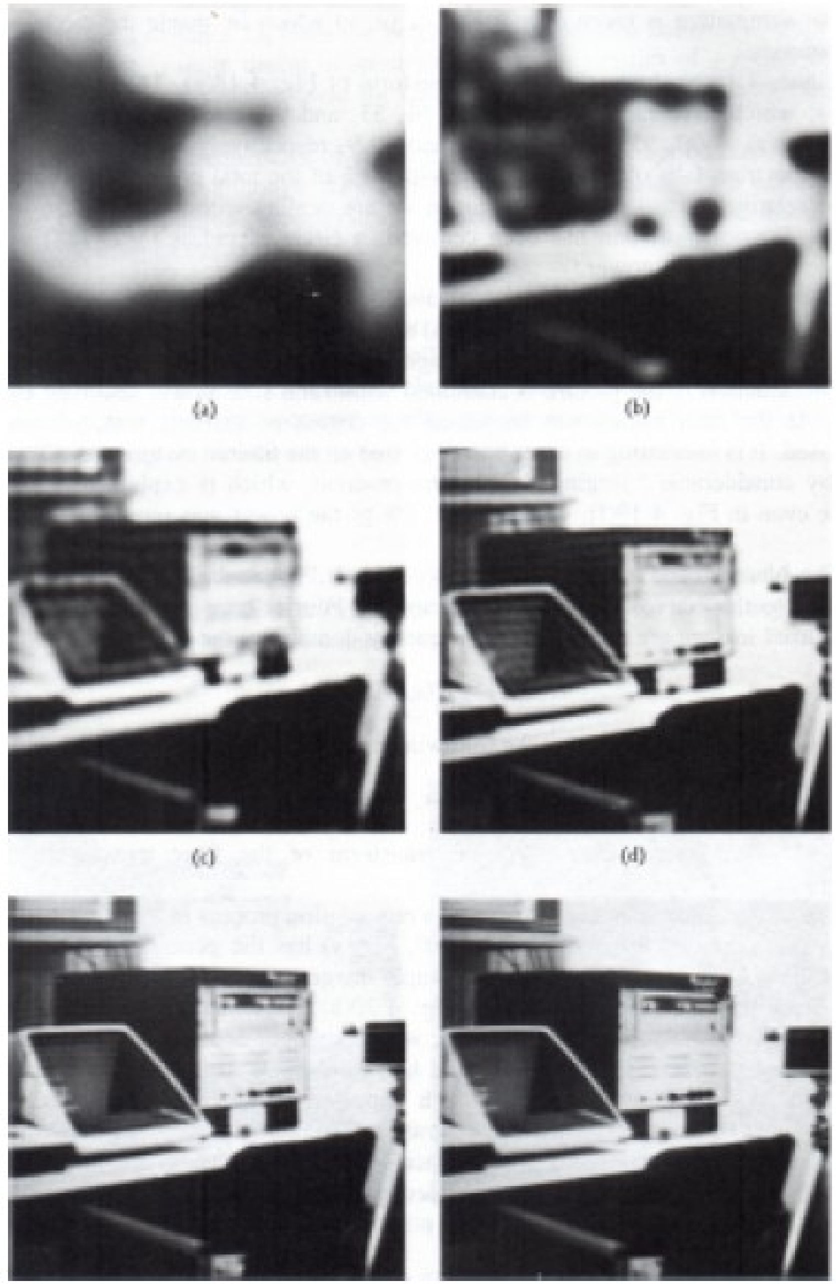
Frequency Bands

Image

Fourier Spectrum



Blurring - Ideal Low pass Filter



Low pass Filter

spatial domain — frequency domain

$f(x,y)$ —— $F(u,v)$

filter

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

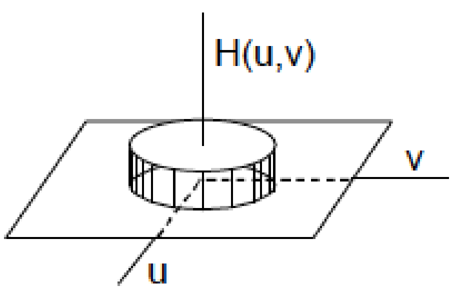
$g(x,y)$ —— $G(u,v)$

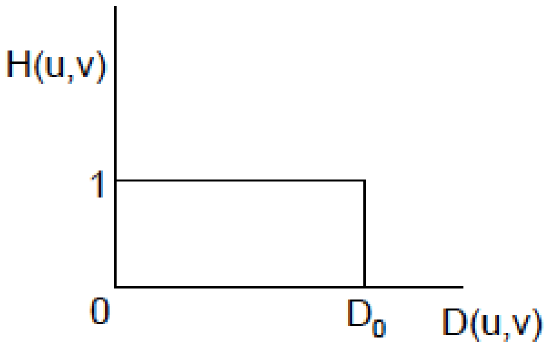
H(u,v) - Ideal Filter

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

$$D(u,v) = \sqrt{u^2 + v^2}$$

D_0 = cut-off frequency



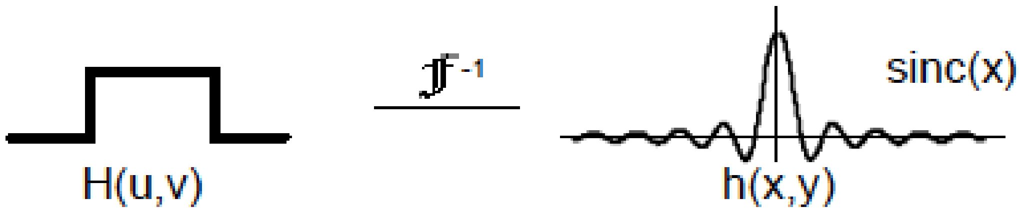


The Ringing Problem

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

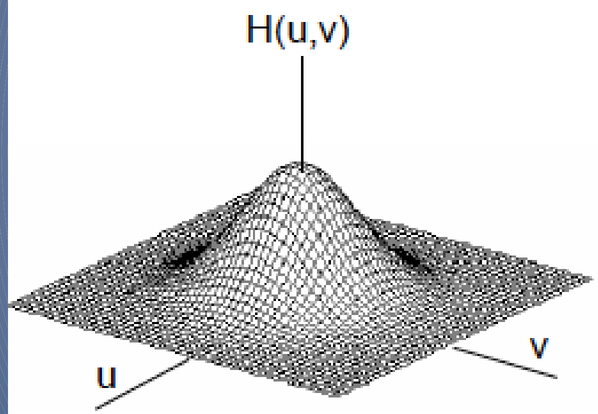
Convolution Theorm

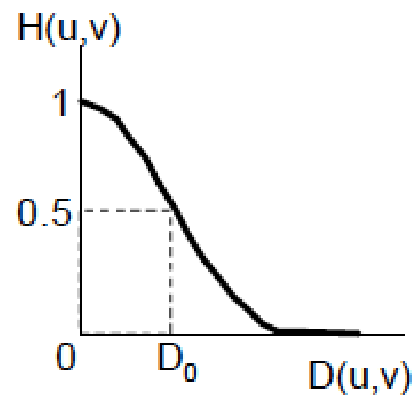
$$g(x,y) = f(x,y) * h(x,y)$$



$\uparrow D_0$ —— $\downarrow$ Ringing radius + blur

H(u,v) - Butterworth Filter





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

$$D(u,v) = \sqrt{u_2 + v_2}$$

Softer Blurring + no Ringing

Blurring - Butterworth Lowpass Filter

(a)

(b)

(c)

(d)

Low Pass Filtering - Image Smoothing

Original - 4 level Quantized Image

Smoothed Image

Original Noisy Image

Smoothed Image

Blurring in the Spatial Domain:

Averaging = convolution with $\frac{1}{4}\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$

= point multiplication of the transform with sinc

Gaussian Averaging = convolution with $\frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$

= point multiplication of the transform with a gaussian.

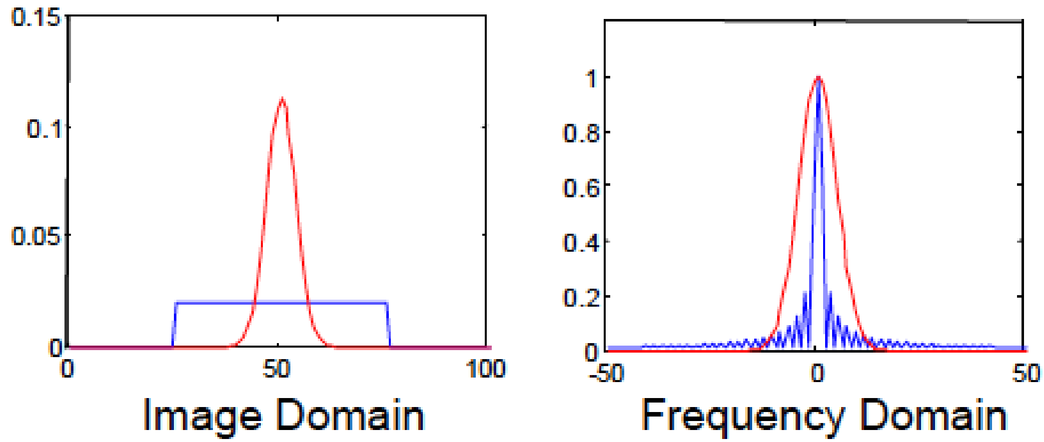


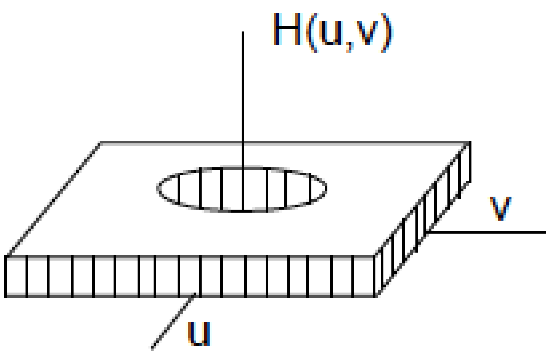
Image Sharpening - High Pass Filter

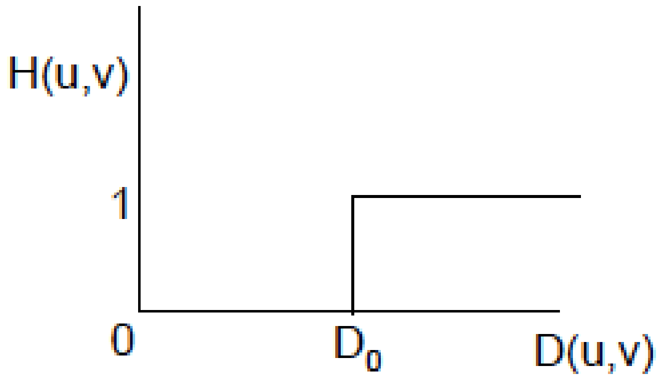
H(u,v) - Ideal Filter

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

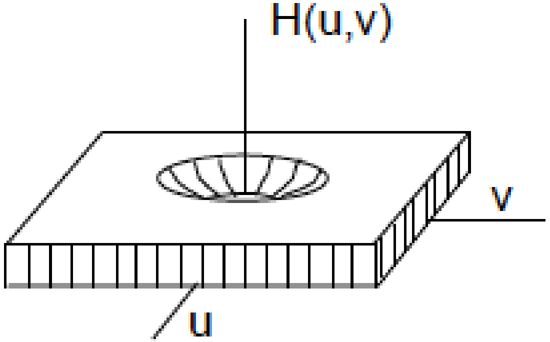
$$D(u,v) = \sqrt{u^2 + v^2}$$

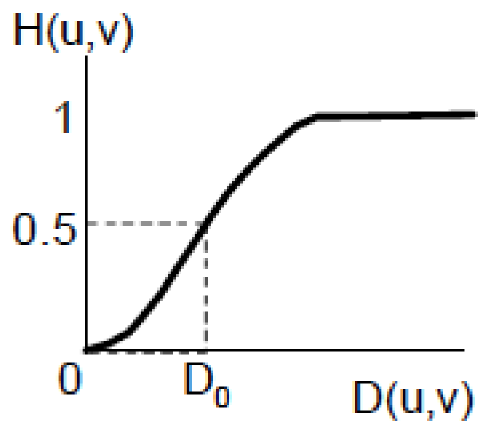
D_0 = cut-off frequency





H(u,v) - Butterworth Filter



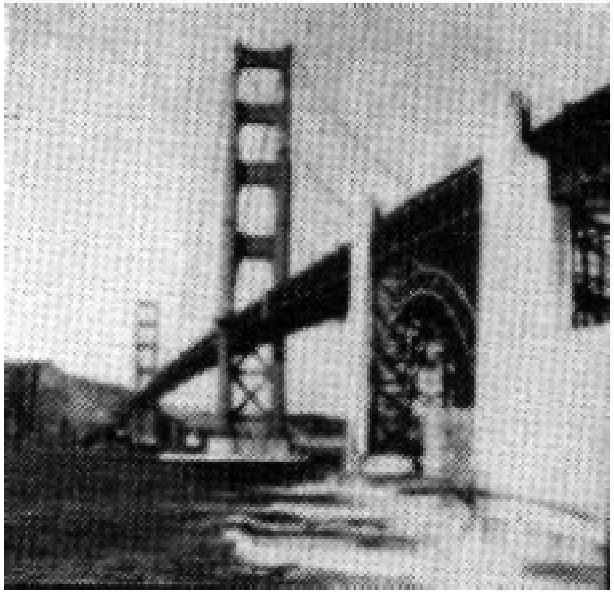


$$H(u,v) = \frac{1}{1 + (D_0/D(u,v))^{2n}}$$

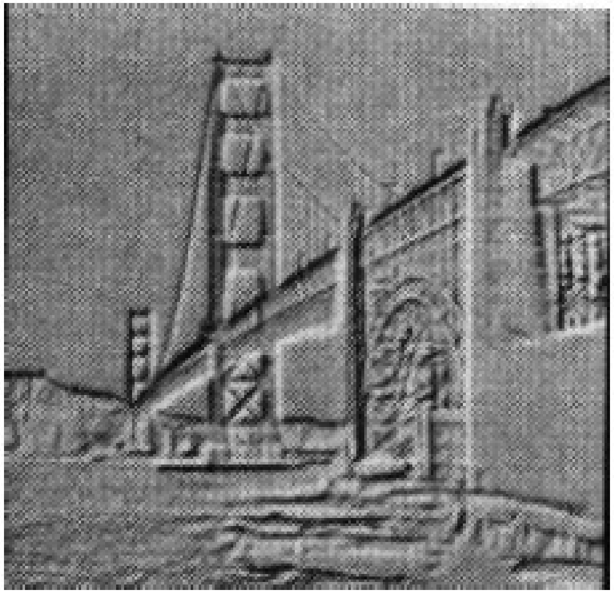
$$D(u,v) = \sqrt{u_2 + v_2}$$

High Pass Filtering

Original



High Pass Filtered



High Frequency Emphasis

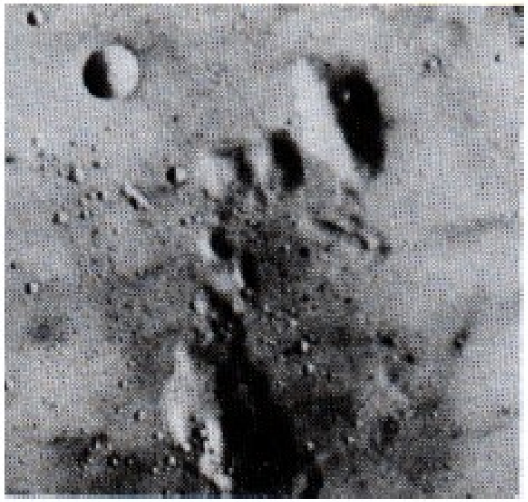
Emphasize High Frequency.
Maintain Low frequencies and Mean.

$$\boxed{H'(u,v) = K_0 + H(u,v)}$$

(Typically $K_0 = 1$)

High Frequency Emphasis - Example

Original

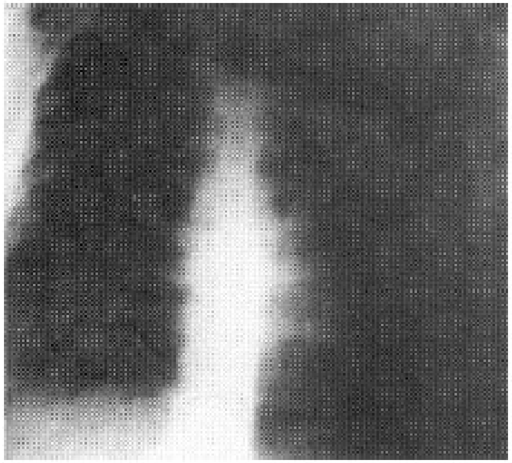


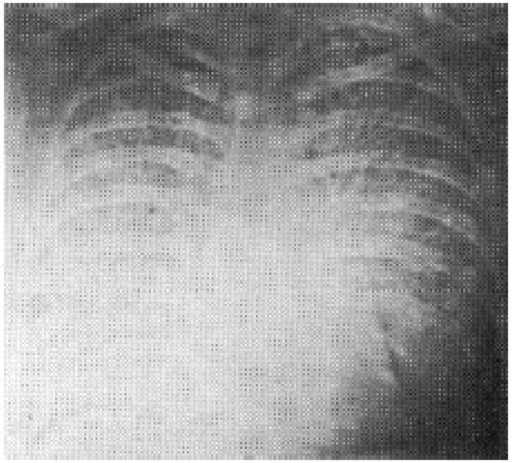
High Frequency Emphasis

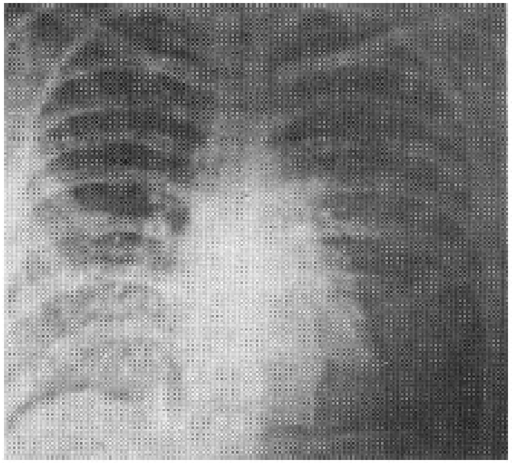
Original

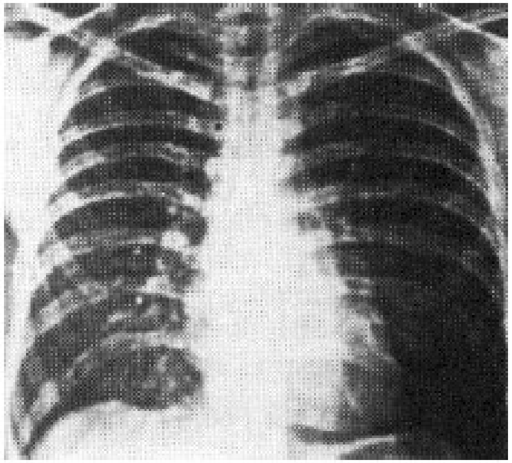
High Frequency Emphasis

High Pass Filtering - Examples


Original


High pass Butterworth Filter


High Frequency Emphasis


High Frequency Emphasis + Histogram Equalization

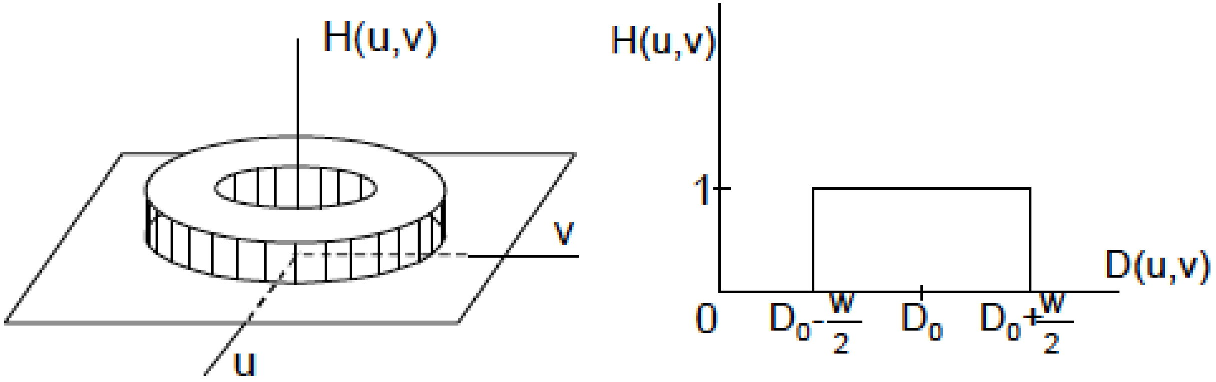
Band Pass Filtering

$$H(u,v) = \begin{cases} 0 & D(u,v) \le D_0 - \frac{w}{2} \\ 1 & D_0 - \frac{w}{2} \le D(u,v) \le D_0 + \frac{w}{2} \\ 0 & D(u,v) > D_0 + \frac{w}{2} \end{cases}$$

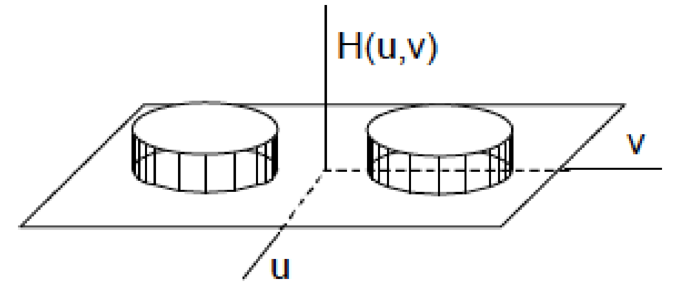
$D(u,v) = \sqrt{u^2 + v^2}$

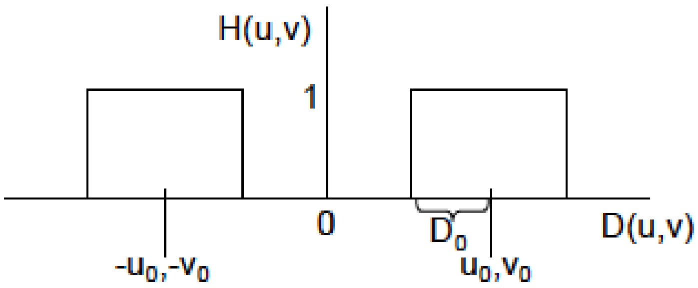
D_0 = cut-off frequency

w = band width



Local Frequency Filtering





$$H(u,v) = \begin{cases} 1 & D_1(u,v) \leq D_0 \ \textbf{ or } \ D_2(u,v) \leq D_0 \\ 0 & \textbf{otherwise} \end{cases}$$

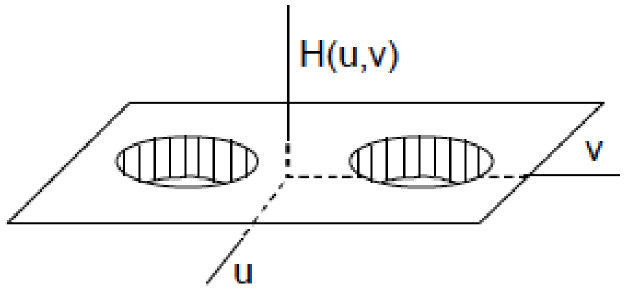
$$D_1(u,v) = \sqrt{(u-u_0)^2 + (v-v_0)^2}$$

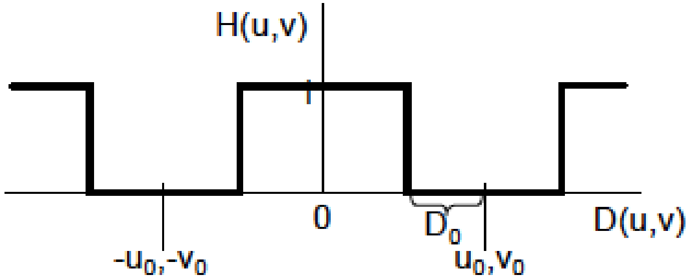
$$D_2(u,v) = \sqrt{(u+u_0)^2 + (v+v_0)^2}$$

D_0 = local frequency radius

u_0,v_0 = local frequency coordinates

Band Rejection Filtering





$$H(u,v) = \begin{cases} 0 & D_1(u,v) \leq D_0 \ \text{ or } \ D_2(u,v) \leq D_0 \\ 1 & \text{otherwise} \end{cases}$$

$$D_1(u,v) = \sqrt{(u-u_0)^2 + (v-v_0)^2}$$

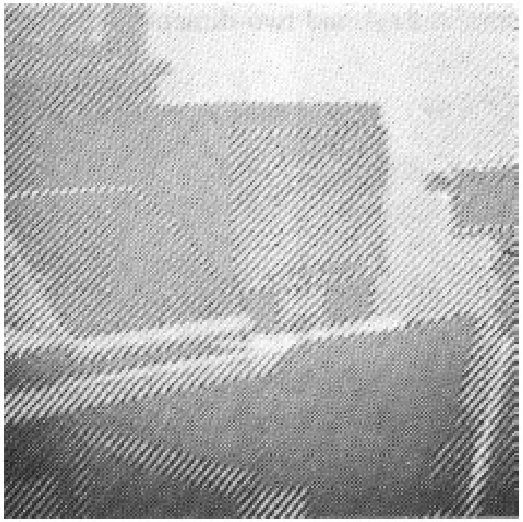
$$D_2(u,v) = \sqrt{(u+u_0)^2 + (v+v_0)^2}$$

D_0 = local frequency radius

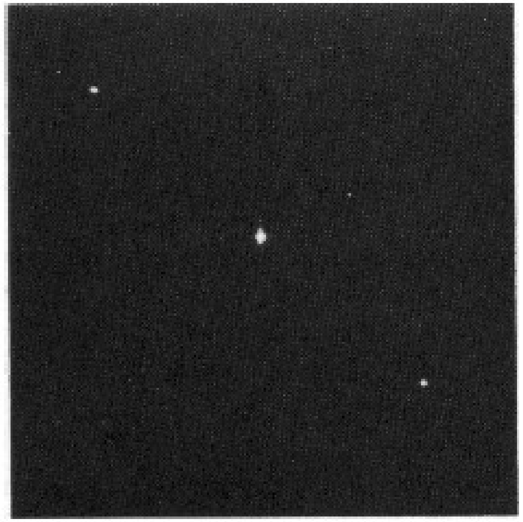
u_0,v_0 = local frequency coordinates

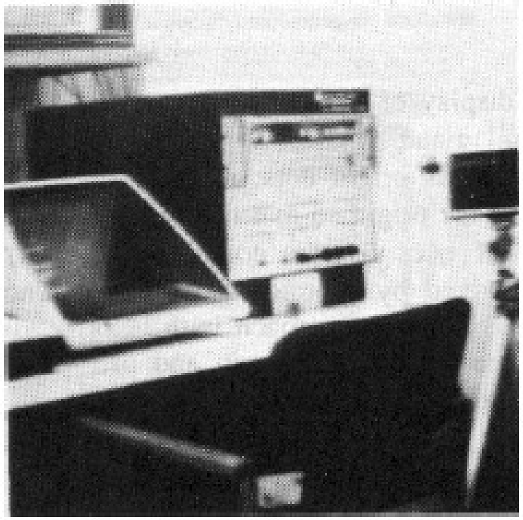
Band Reject Filter - Example

Original Noisy image



Fourier Spectrum

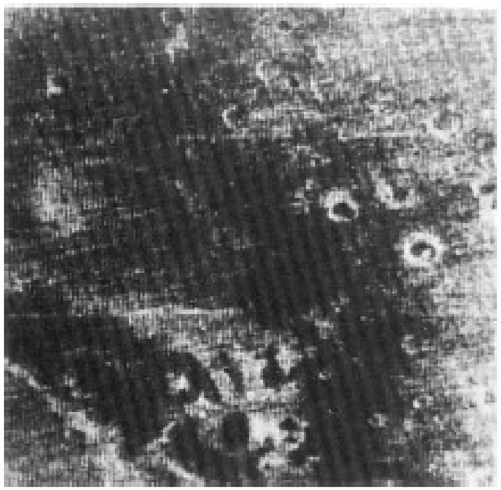




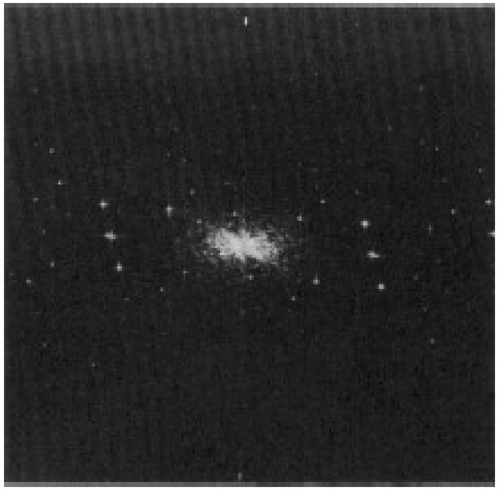
Band Reject Filter

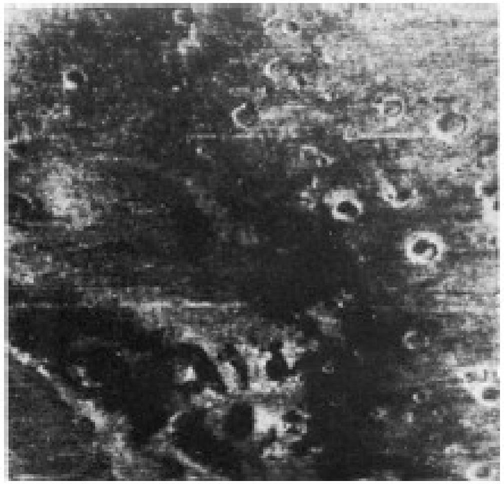
Local Reject Filter - Example

Original Noisy image



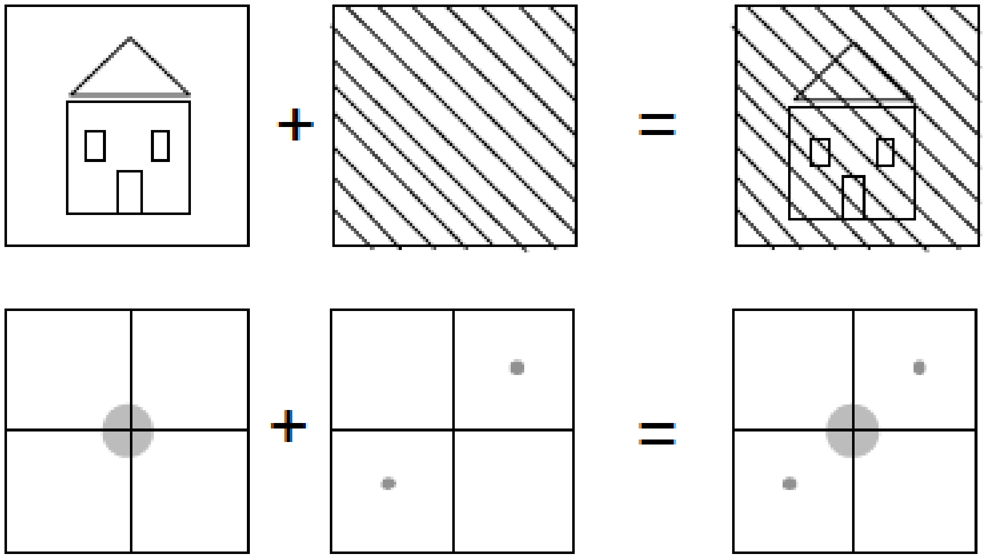
Fourier Spectrum





Local Reject Filter

Image Enhancement



Homomorphic Filtering

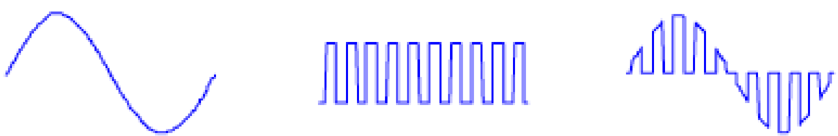
Reflectance Model:

Illumination	i(x,y)
Surface Reflectance	r(x,y)
Brightness	$\boxed{f(x,y) = i(x,y) \cdot r(x,y)}$

Assumptions:

Illumination changes "slowly" across scene
$\Longrightarrow$ Illumination $\approx$ low frequencies.

Surface reflections change "sharply" across scene
$\Longrightarrow$ reflectance $\approx$ high frequencies.



Illumination   Reflectance   Brightness

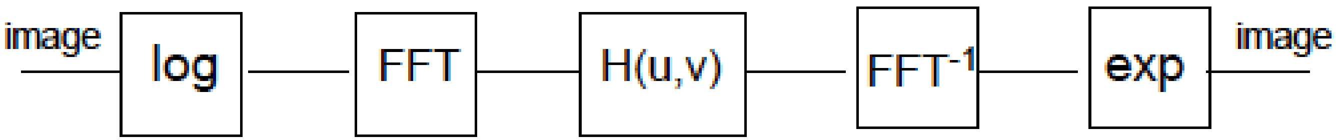
Goal: repress the low frequencies associated with I(x,y).
However:

$$\mathfrak{F}(i(x,y) \cdot r(x,y)) \neq \mathfrak{F}(i(x,y)) \cdot \mathfrak{F}(r(x,y))$$

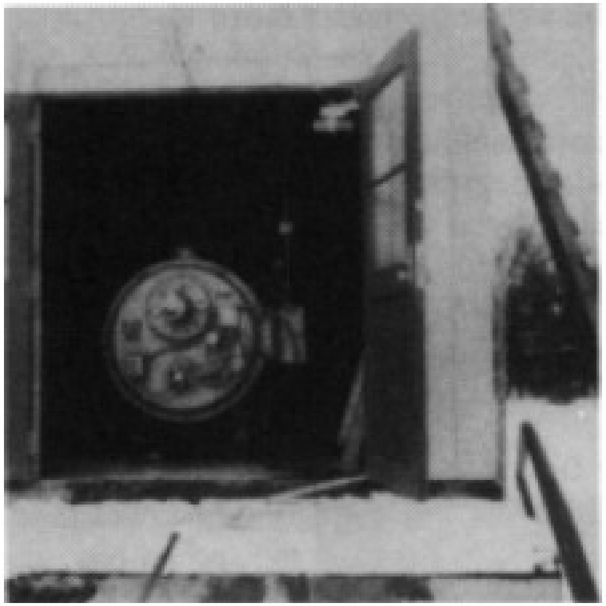
Perform:

$$\boxed{\begin{aligned} \mathbf{z}(x,y) &= \mathbf{log}(f(x,y)) \\ &= \mathbf{log}(i(x,y) \bullet (r(x,y)) = \mathbf{log}(i(x,y)) + \mathbf{log}(r(x,y)) \end{aligned}}$$

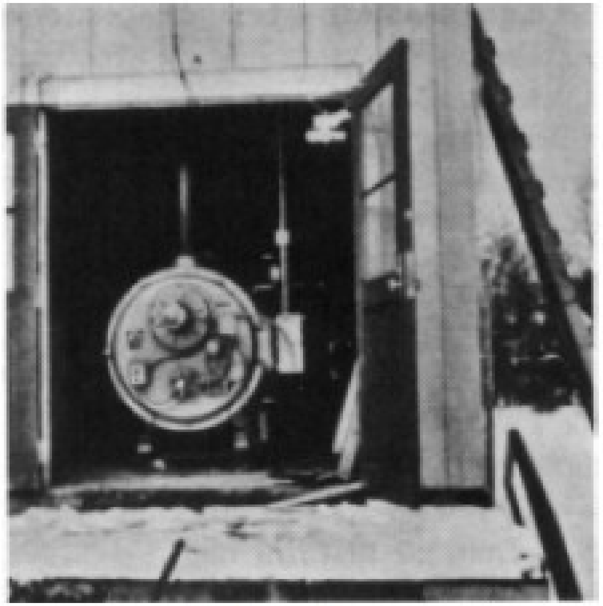
Homomorphic Filtering:



Homomorphic Filtering



Original



Filtered

Acknowledgement

- New Mexico Tech University

THANK YOU ALL