Filtering in Frequency Domain

Basics of Filtering in Frequency Domain

2-Dimensional Discrete Fourier Transform

For an image of size MxN pixels

2-D DFT

$$F(u,v) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) e^{-j2\pi(ux/M + vy/N)}$$

u = frequency in x direction, $u = 0 ,\ldots, M-1$
v = frequency in y direction, $v = 0 ,\ldots, N-1$

2-D IDFT

$$f(x,y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u,v) e^{j2\pi(ux/M + vy/N)}$$

$x = 0 ,\ldots, M-1$
$y = 0 ,\ldots, N-1$

2-Dimensional Discrete Fourier Transform (cont.)

F(*u,v*) can be written as

$$F(u,v) = R(u,v) + jI(u,v) \quad \text{or} \quad F(u,v) = |F(u,v)|e^{j\varphi(u,v)}$$

where

$$|F(u,v)| = \sqrt{R(u,v)^2 + I(u,v)^2} \qquad \phi(u,v) = \tan^{-1}\left(\frac{I(u,v)}{R(u,v)}\right)$$

For the purpose of viewing, we usually display only the Magnitude part of *F*(*u,v*)

Relation Between Spatial and Frequency Resolutions

$$\Delta u = \frac{1}{M\Delta x} \qquad \Delta v = \frac{1}{N\Delta y}$$

where

Δx = spatial resolution in x direction

Δy = spatial resolution in y direction

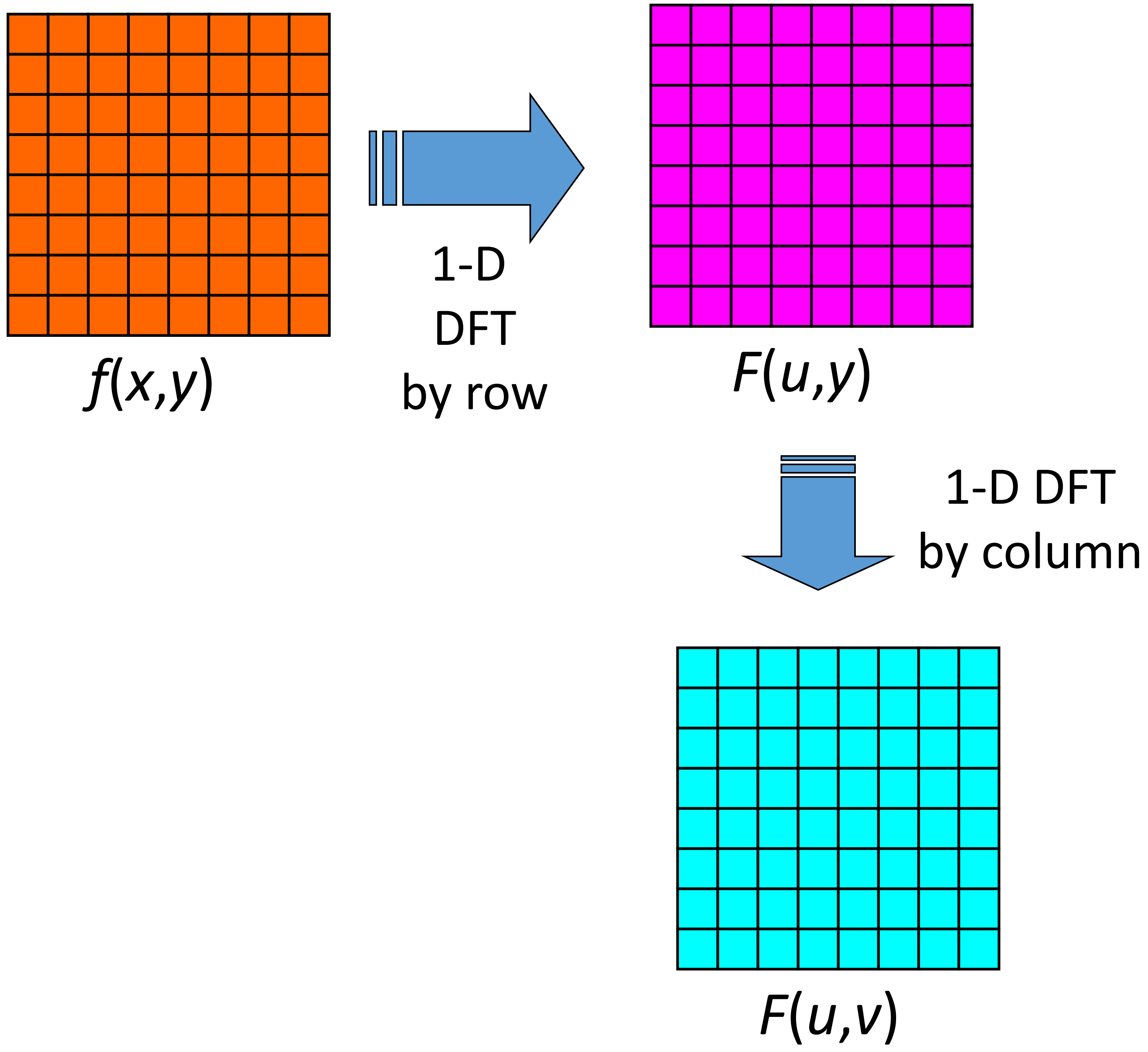
(Δx and Δy are pixel width and height.)

Δu = frequency resolution in x direction

Δv = frequency resolution in y direction

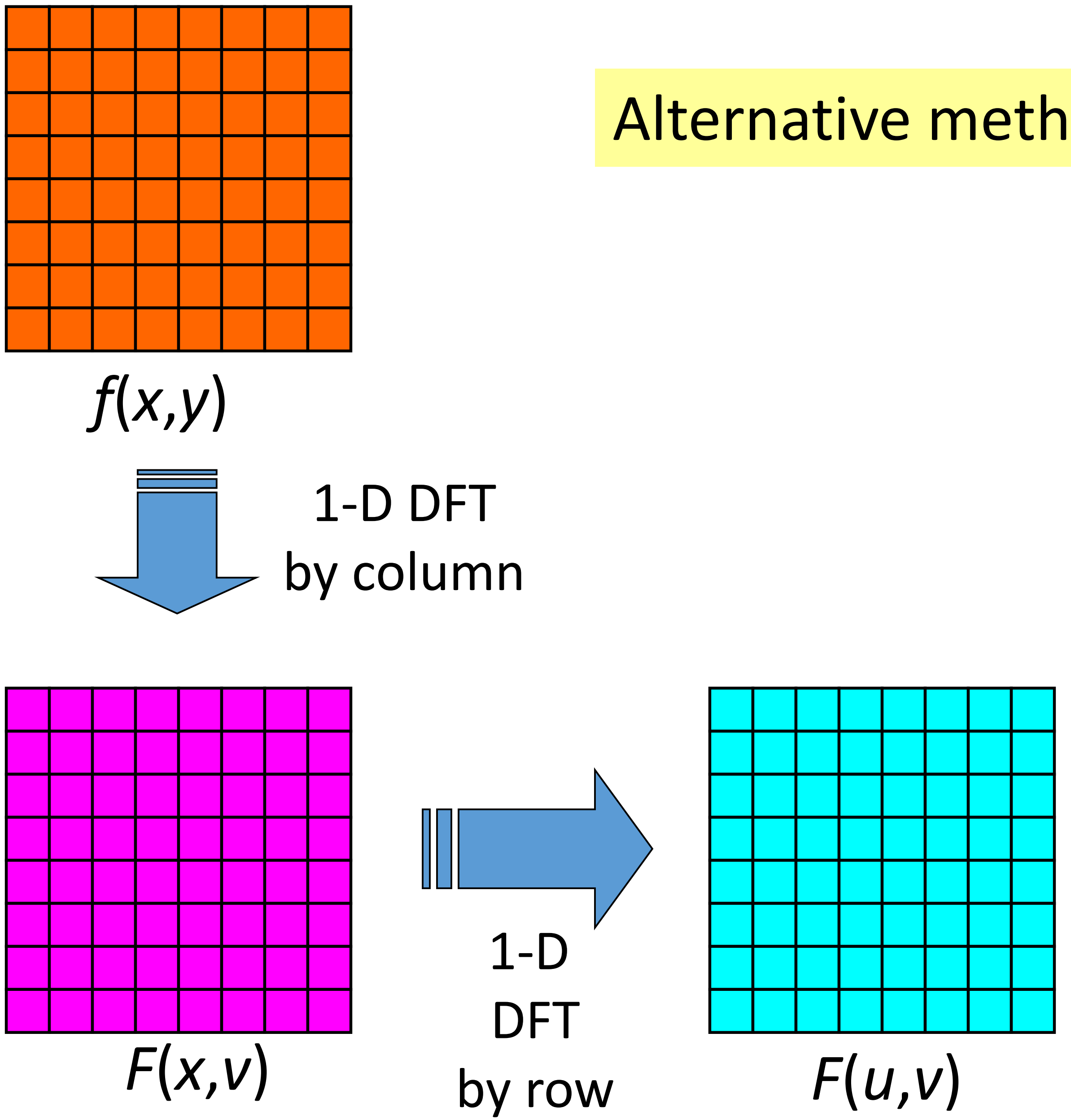
N,M = image width and height

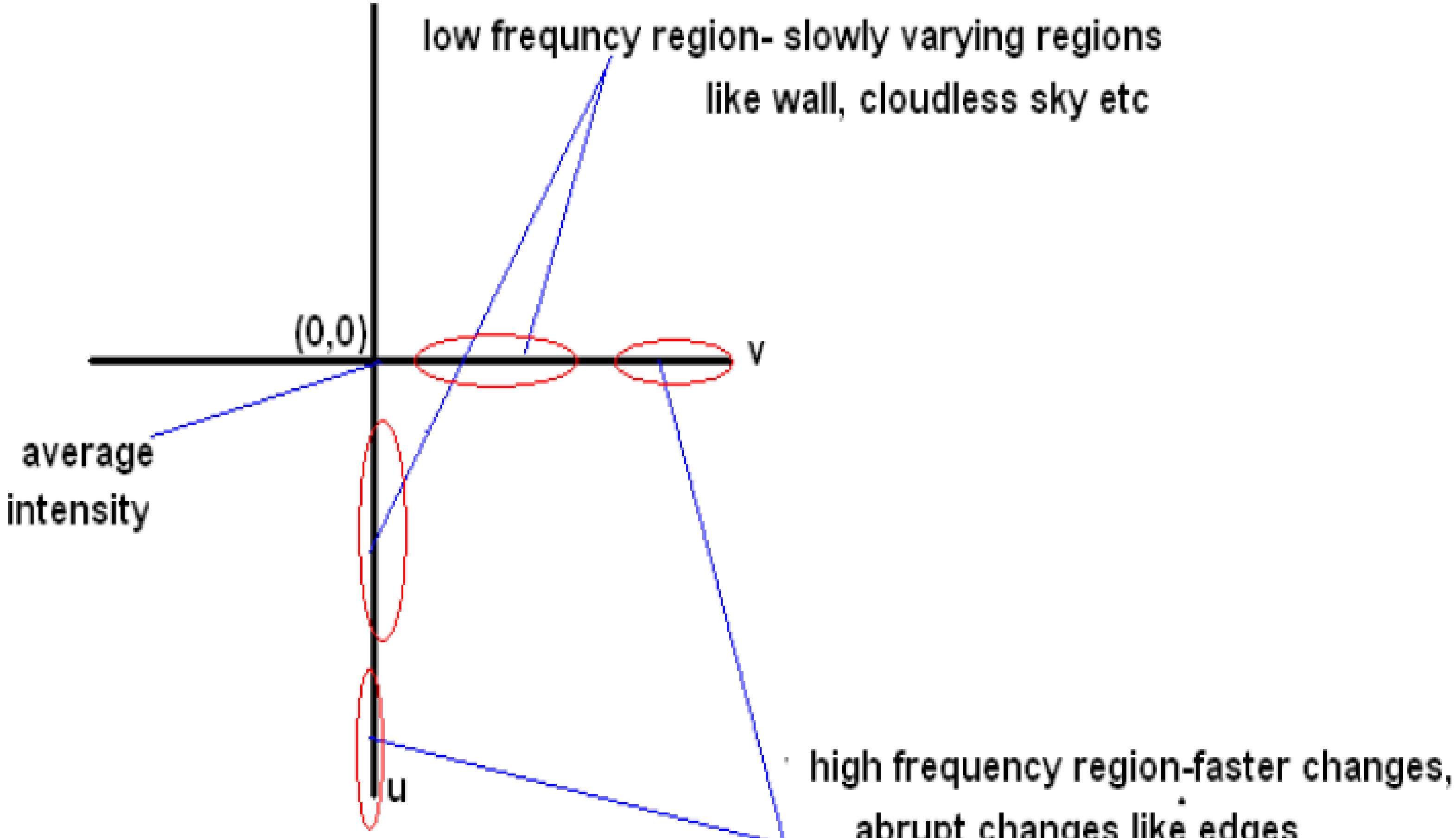
How to Perform 2-D DFT by Using 1-D DFT



How to Perform 2-D DFT by Using 1-D DFT (cont.)

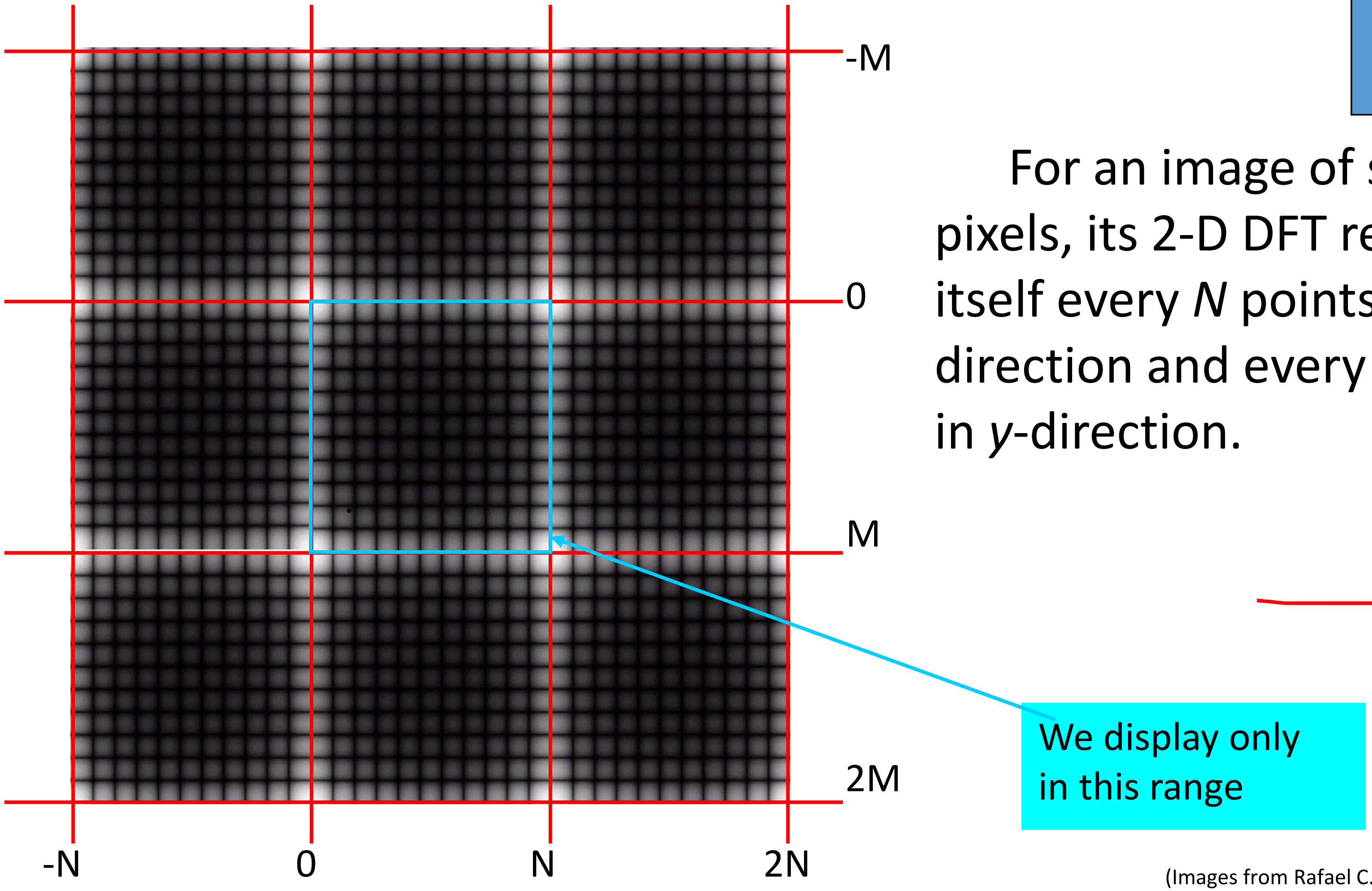
Alternative method

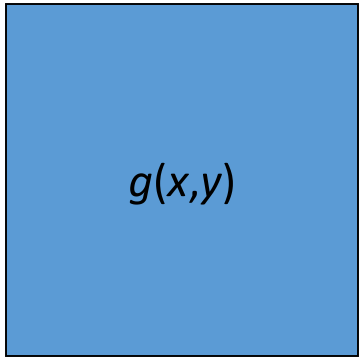



low frequncy region- slowly varying regions
like wall, cloudless sky etc
(0,0)
v
average
intensity
u
high frequency region-faster changes,
abrupt changes like edges

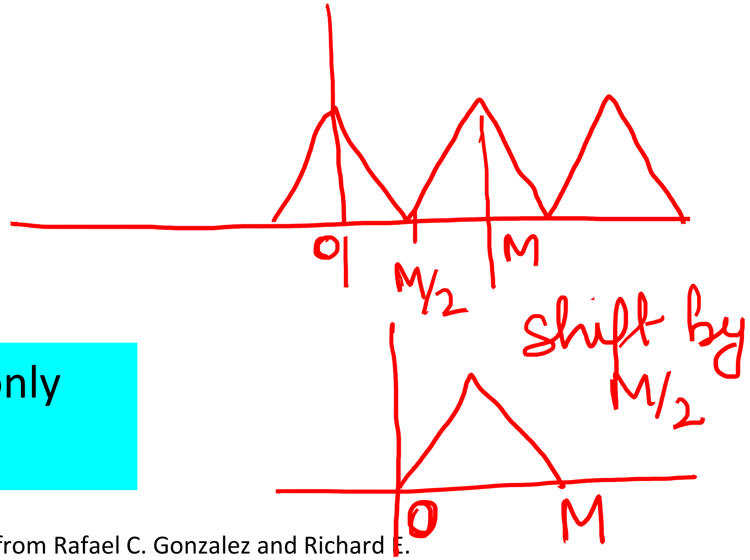
Periodicity of 2-D DFT

2-D DFT: $$F(u,v)=\frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)e^{-j2\pi(ux/M+vy/N)}$$





For an image of size *NxM* pixels, its 2-D DFT repeats itself every *N* points in *x*-direction and every *M* points in *y*-direction.



(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

→ How to get shift in Freq. domain?

$$f(x) \cdot e^{j2\pi u_0 x/M} \Leftrightarrow F(u-u_0)$$

$u_0 = M/2$

$$f(x) \cdot e^{(j\cancel{2}\pi \cancel{M}/\cancel{2}\, x)/\cancel{M}} \Leftrightarrow F(u-M/2)$$

$$f(x) \cdot e^{j\pi x} \Leftrightarrow F(u-M/2)$$

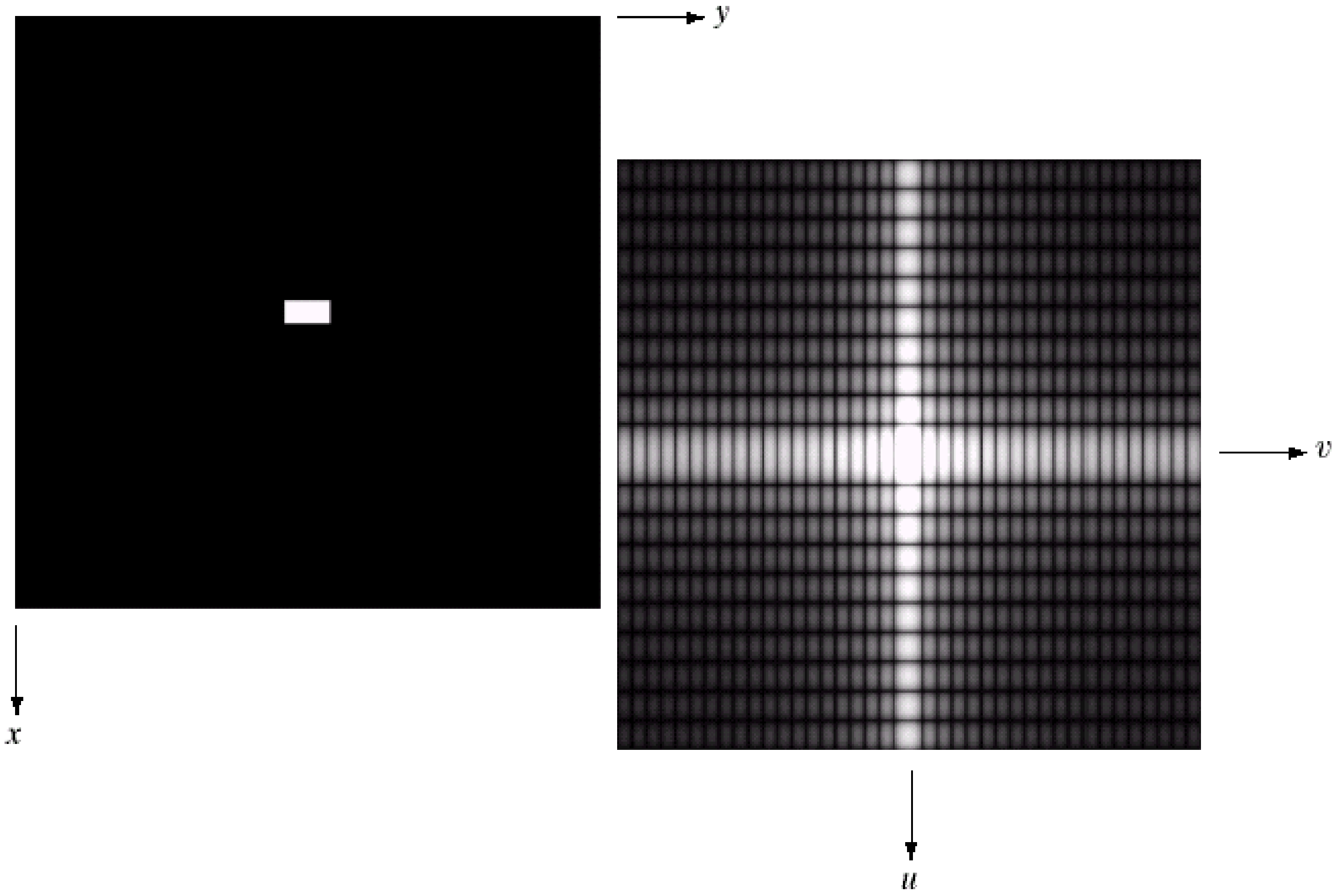
$$f(x) \cdot (-1)^{x} \Leftrightarrow F(u-M/2)$$

for 2D

$$\underbrace{f(x,y)\,(-1)^{x+y}} \Leftrightarrow \underbrace{F(u-M/2,\, v-N/2)}$$

centralizing the spectrum

Example of 2-D DFT

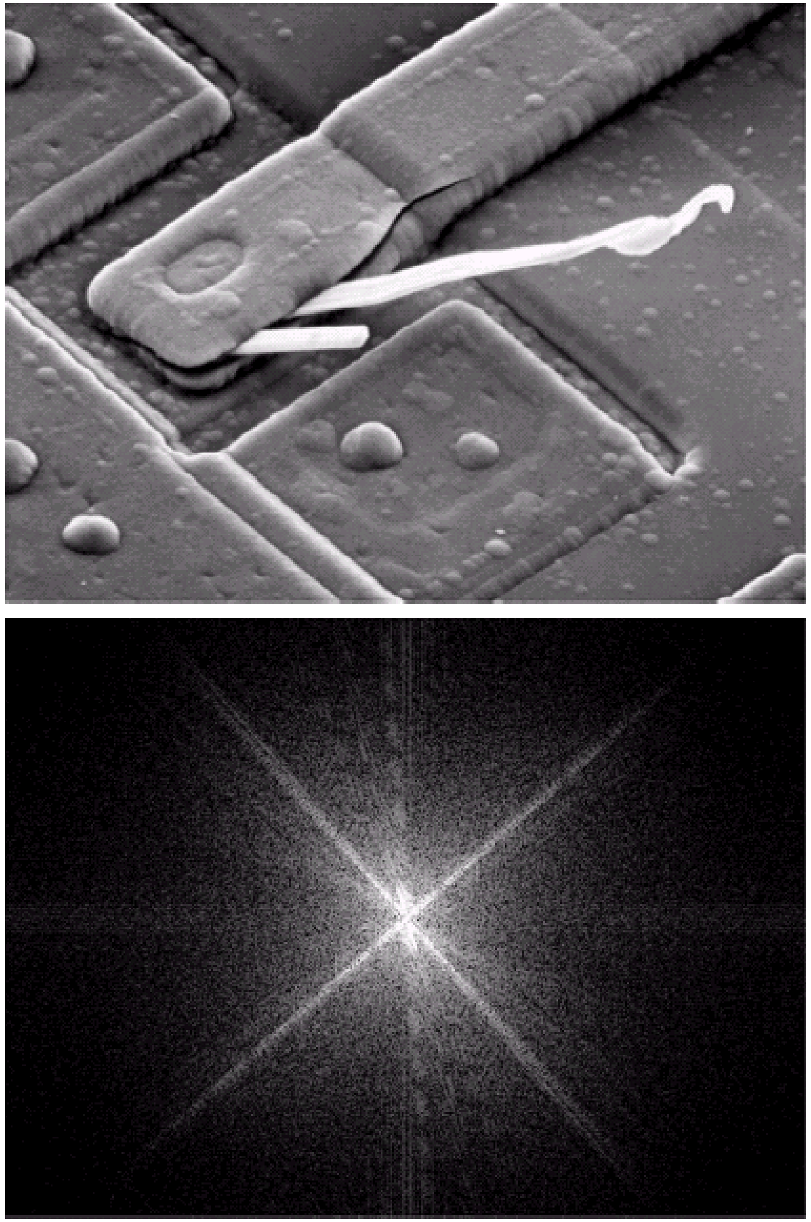


a b

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.

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Example of 2-D DFT



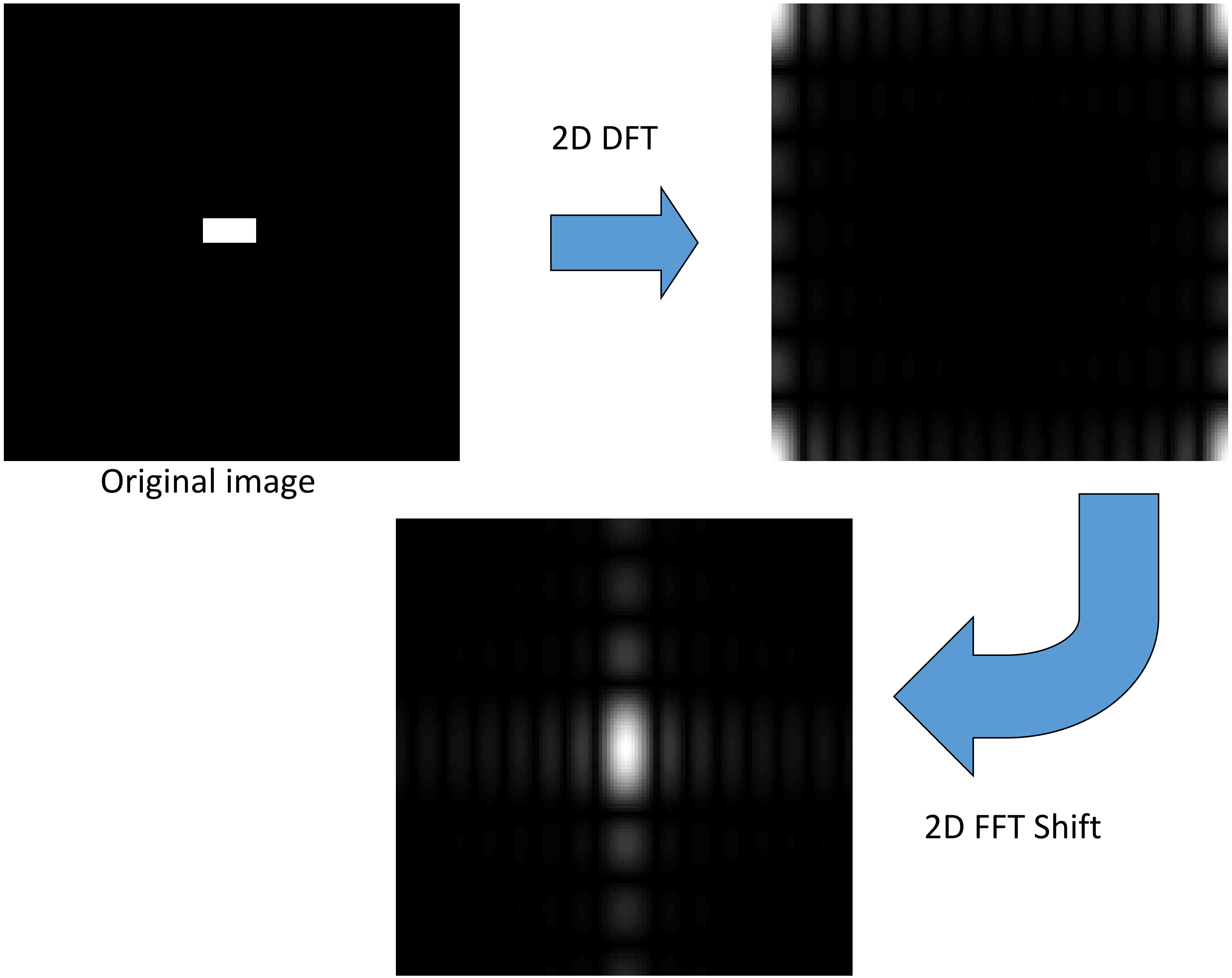
a
b

FIGURE 4.4 (a) SEM image of a damaged integrated circuit. (b) Fourier spectrum of (a). (Original image courtesy of Dr. J. M. Hudak, Brockhouse Institute for Materials Research, McMaster University, Hamilton, Ontario, Canada.)

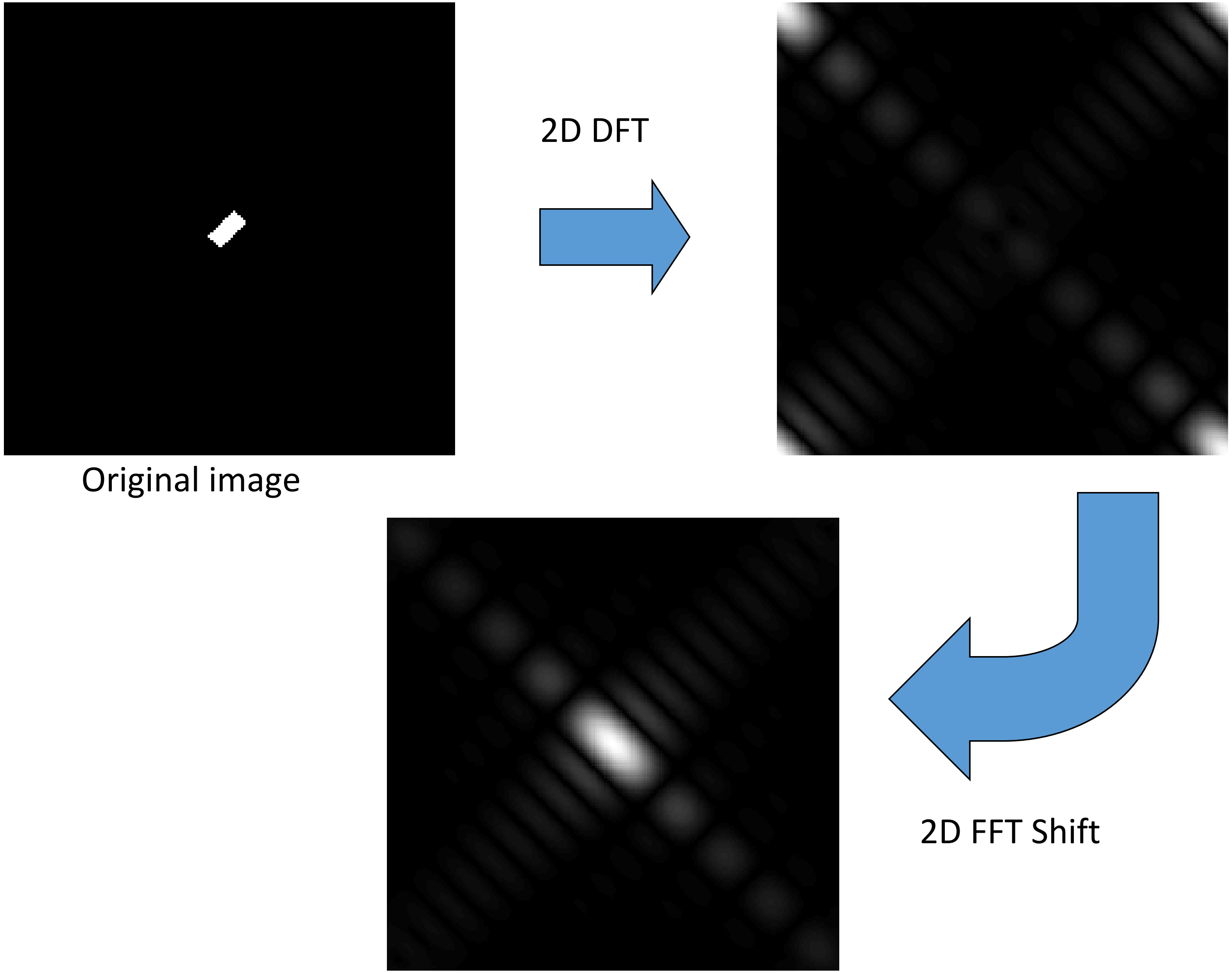
Notice that direction of an object in spatial image and Its Fourier transform are orthogonal to each other.

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Example of 2-D DFT



Example of 2-D DFT



Frequency Domain Methods

Spatial Domain

$$f(x,y) \longrightarrow \boxed{h(x,y)} \longrightarrow g(x,y)$$

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

Frequency Domain

$$F(u,v) \longrightarrow \boxed{H(u,v)} \longrightarrow G(u,v)$$

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

$$(\, g(x,y) = F^{-1}(F(u,v)\, H(u,v))\,)$$

element-wise complex multiplication

Major filter categories

- Filters are classified based on their properties in the frequency domain:

  (1) Low-pass

  (2) High-pass

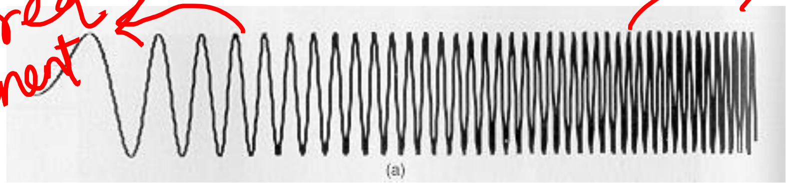
  (3) Band-pass

  (4) Band-stop

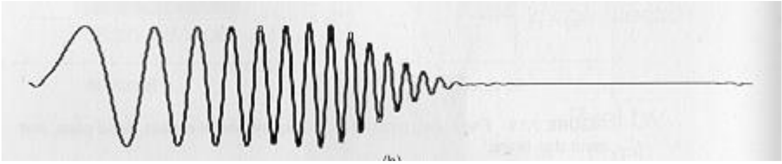
Example

low freq component

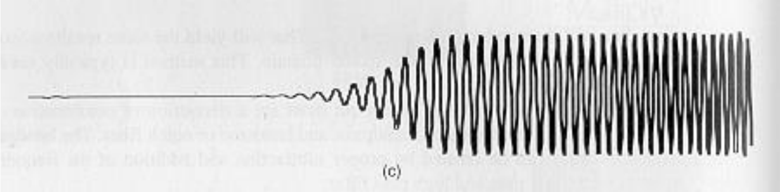
high freq component



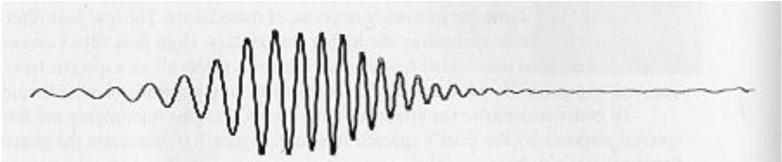
Original signal



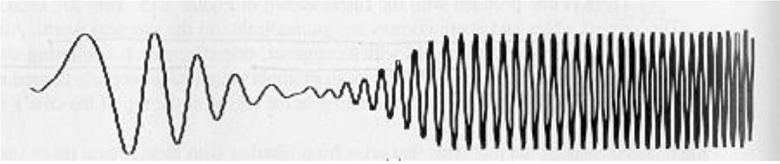
Low-pass filtered



High-pass filtered



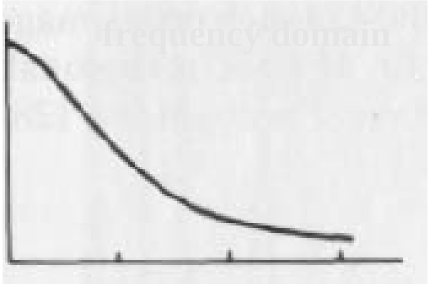
Band-pass filtered



Band-stop filtered

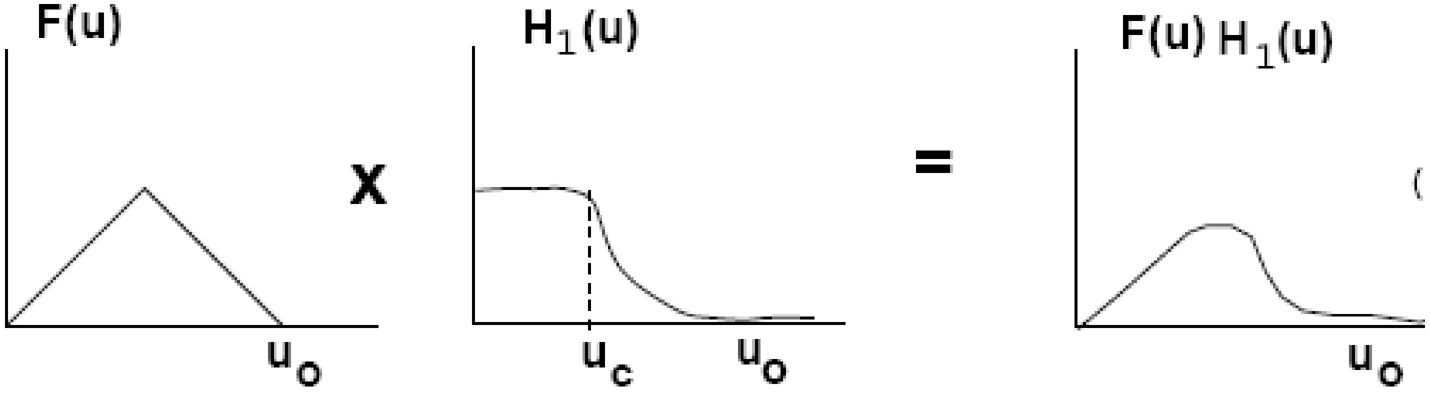
Low-pass filters (i.e., smoothing filters)

- Preserve low frequencies.
  - Useful for removing details and noise suppression.



representative example

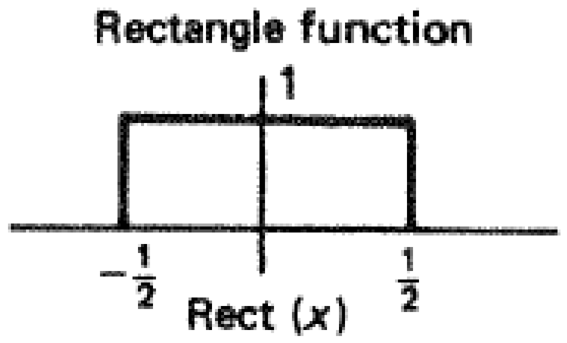
Example:



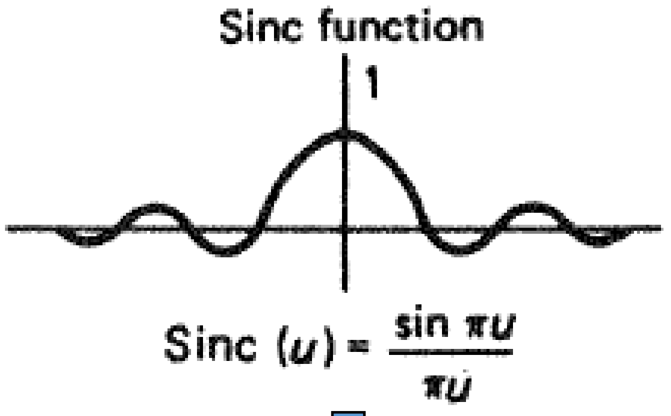
Low-pass filter - Example

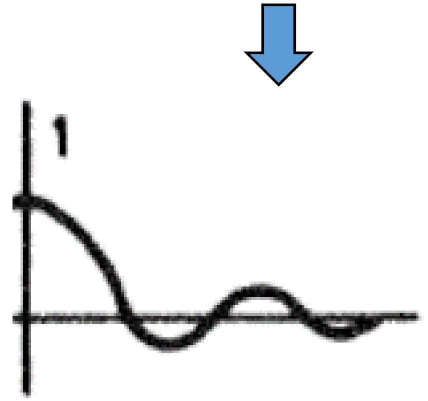
- The rectangle function is a low pass filter!

spatial domain



frequency domain

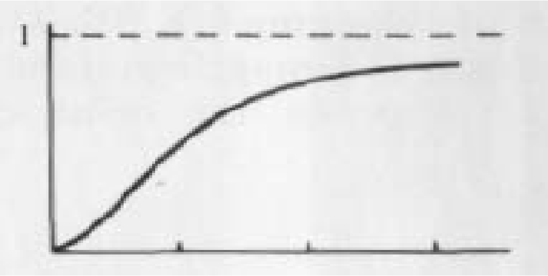




Note: when discussing filters in the frequency domain, we'll illustrate their properties using positive frequencies only; however, keep in mind that there is a "symmetric" component on the negative axis!

High-pass filters (i.e., sharpening filters)

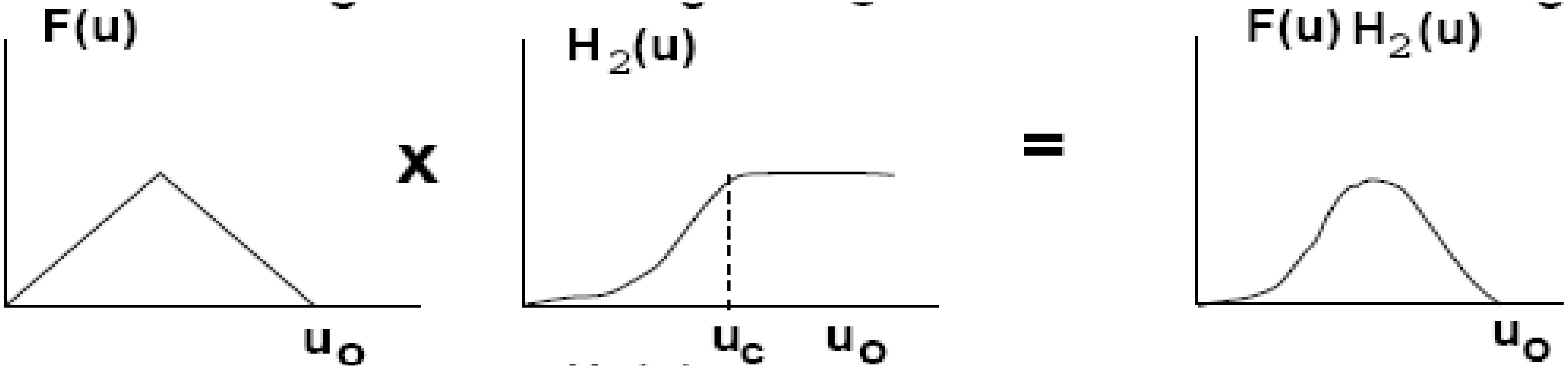
- Preserve high frequencies.
  - Useful for highlighting details.



representative example

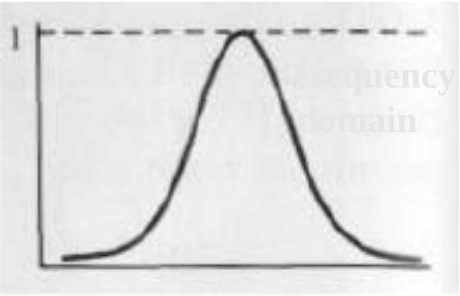
frequency domain

Example:



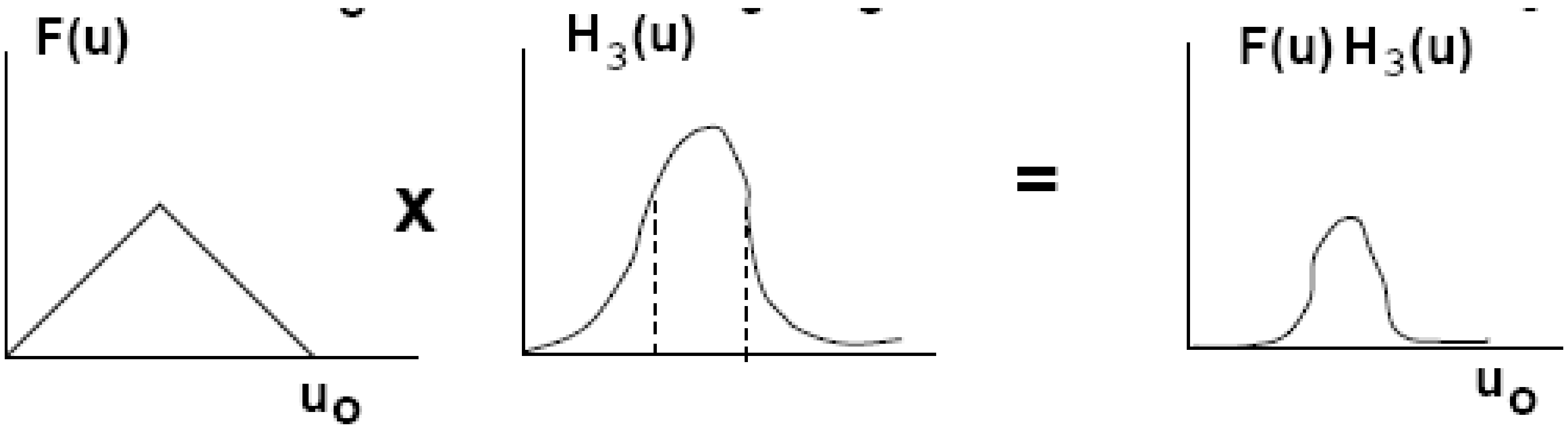
Band-pass filters

- Preserve frequencies within a specific band.
  - Useful in image restoration.



representative
example

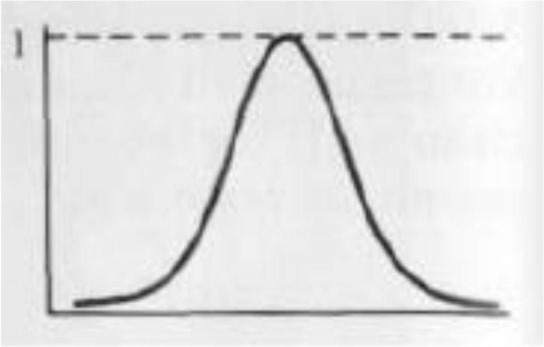
Example:



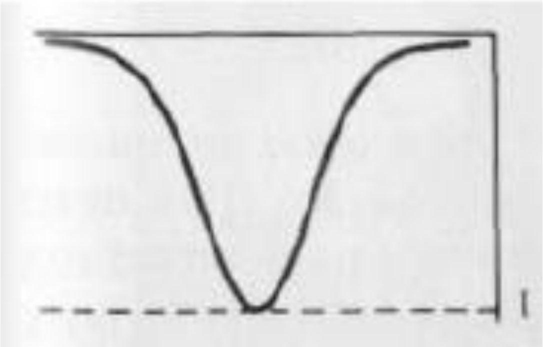
Band-stop filters

- Removes frequencies within a specific band.
  - Useful in image restoration.

Band-pass



Band-stop



representative
example

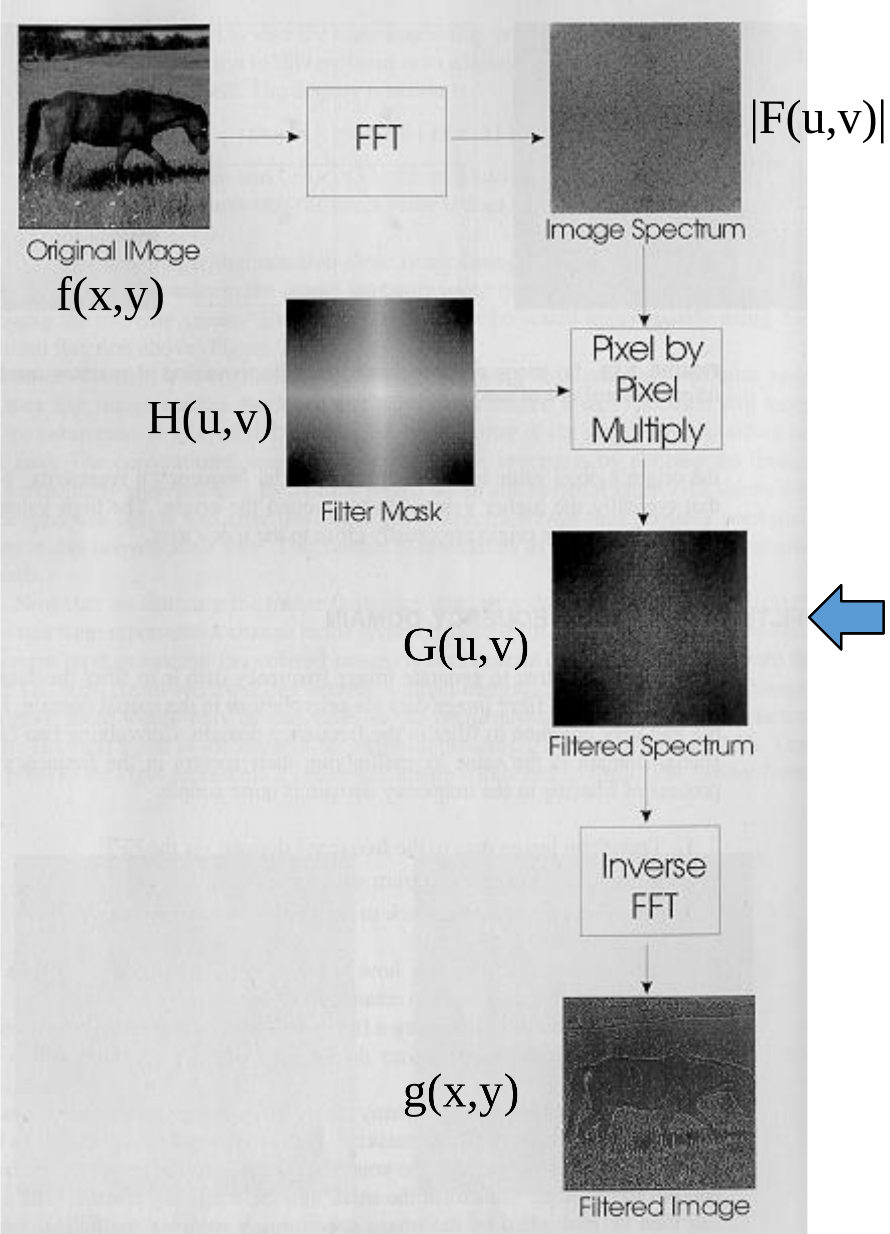
Frequency Domain Methods

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

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

Case 1: H(u,v) is specified directly in the frequency domain.

Case 2: H(u,v) is specified indirectly in the frequency domain by specifying h(x,y) in the spatial domain.



complex multiplication

Very important: the centers of F(u,v) and H(u,v) must coincide!

Padding (for removing wrap around effect)

- 2-D :Let $f(x, y)$ and $h(x, y)$ be two image arrays of sizes A x B and C x D pixels, respectively.

$$P \geq A + C - 1 \qquad\qquad Q \geq B + D - 1$$

- If both arrays are of the same size, M x N then we require that

$$P \geq 2M - 1$$

$$Q \geq 2N - 1$$

- As rule, DFT algorithms tend to execute faster with arrays of even size, so it is good practice to select P and Q as the smallest even integers that satisfy the preceding equations.
- If the two arrays are of the same size, this means that P and Q are selected as twice the array size.

Suppose $f(x,y)$ is an image of size 5×5 ∴ $A \times B = 5 \times 5$

$h(x,y)$ is a filter of size 5×5 ∴ $C \times D = 5 \times 5$

∴ $P \geq A + C - 1$ $\quad Q \geq B + D - 1$

usually if f & h are of same size $M \times N$

$P = 2M \quad Q = 2N$

$$f_p(x,y) = \begin{cases} f(x,y) & 0 \leq x \leq A - 1 \text{ and } 0 \leq y \leq B - 1 \\ 0 & A \leq x \leq P \text{ and } B \leq x \leq Q \end{cases}$$

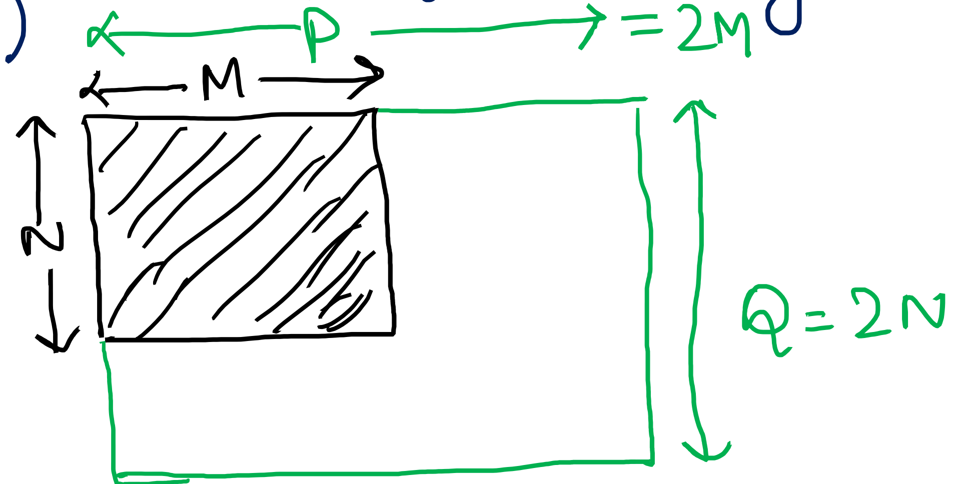
M N M N

fp(x,y) is padded image

same for $h_p(x,y)$

∴ 5×5 image will now be = ?

10×10



Frequency domain filtering:

1. Given an input image $f(x, y)$ of size M x N, set the padding sizes P and Q, e.g.,
$P=2M$ and $Q=2N$ (note: P and Q must also be a power of 2 to use FFT)

2. Form a padded image $f_p(x, y)$ of size P x Q

3. Multiply $f_p(x, y)$ by $(-1)^{x+y}$ to center its spectrum.

4. Compute the DFT, $F(u, v)$, of the image from Step 3.

$$F(u,v) = R(u,v) + jI(u,v)$$

Frequency domain filtering: (cont'd)

5. Create H (u, v) of size P x Q with center at (P/2, Q/2). This is typically done by specifying and sampling the desired filter function.

Example: $$H(u,v) = \frac{1}{1 + [\sqrt{u^2 + v^2}/D_0]^{2n}}$$ Butterworth lowpass filter

6. Compute the product G (u, v) = H $(u$, v) F $(u$, v) using elementwise "complex" multiplication. If H (u, v) is real, this is:

G(u,v)= F(u,v)H(u,v) = H(u,v) R(u,v) + jH(u,v)I(u,v)

where u=0,1,...,P-1, v=0,1,…,Q-1

where F(u,v) = R(u,v) + jI(u,v)

Frequency domain filtering:

7. Obtain the filtered image $g_p\,(x, y)$ by computing the inverse DFT of $G\,(u, v)$:

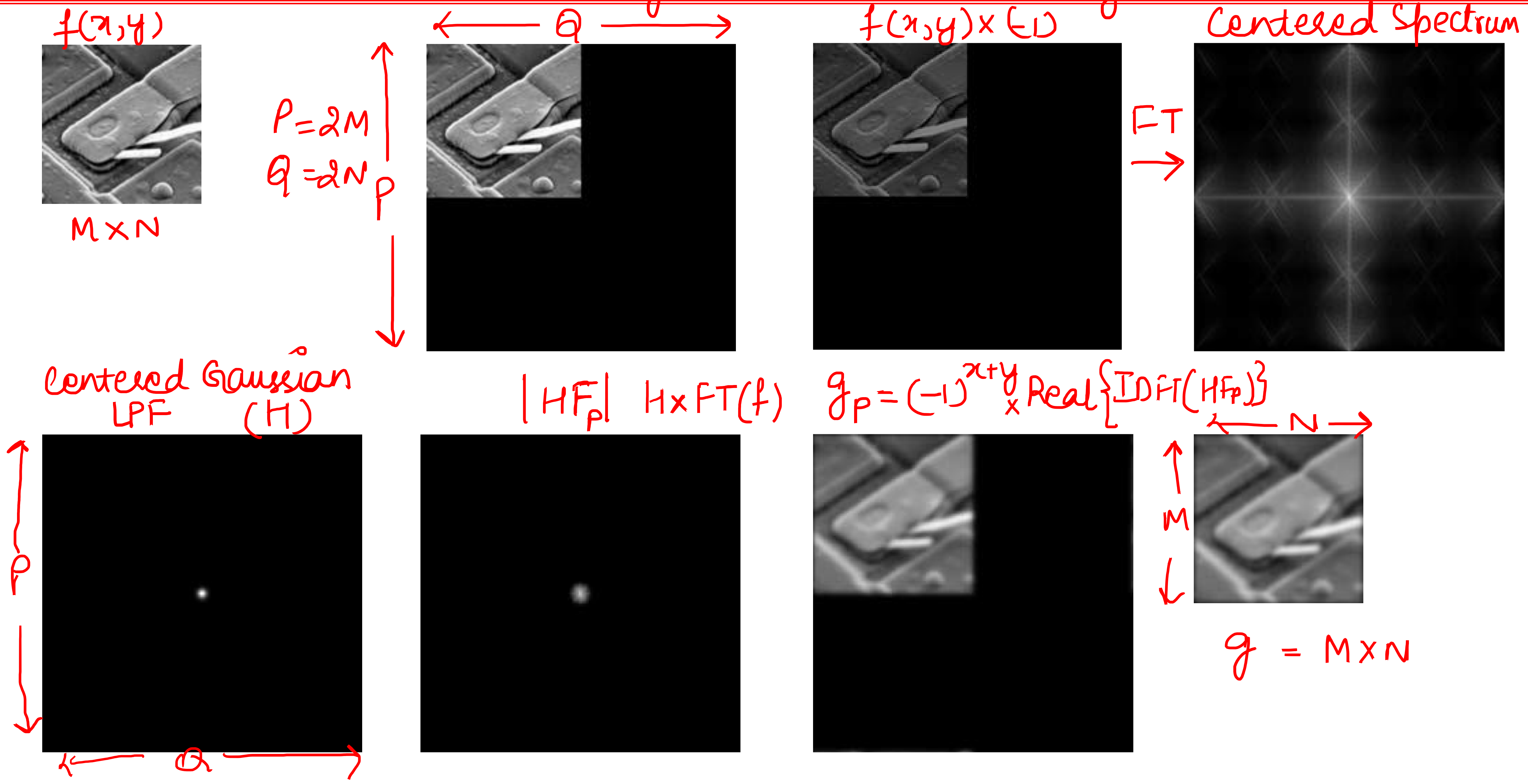
$$g_p(x, y) = \left\{\text{real}\left[\Im^{-1}[G(u, v)]\right]\right\}(-1)^{x+y}$$

explicitly disregard the imaginary part

"undo" the centering transformation

8. Obtain the final filtered result, $g\,(x, y)$, of size M x N, by extracting the M x N region from the top, left quadrant of $g_p\,(x, y)$ (i.e., "undo" padding).

Padded image

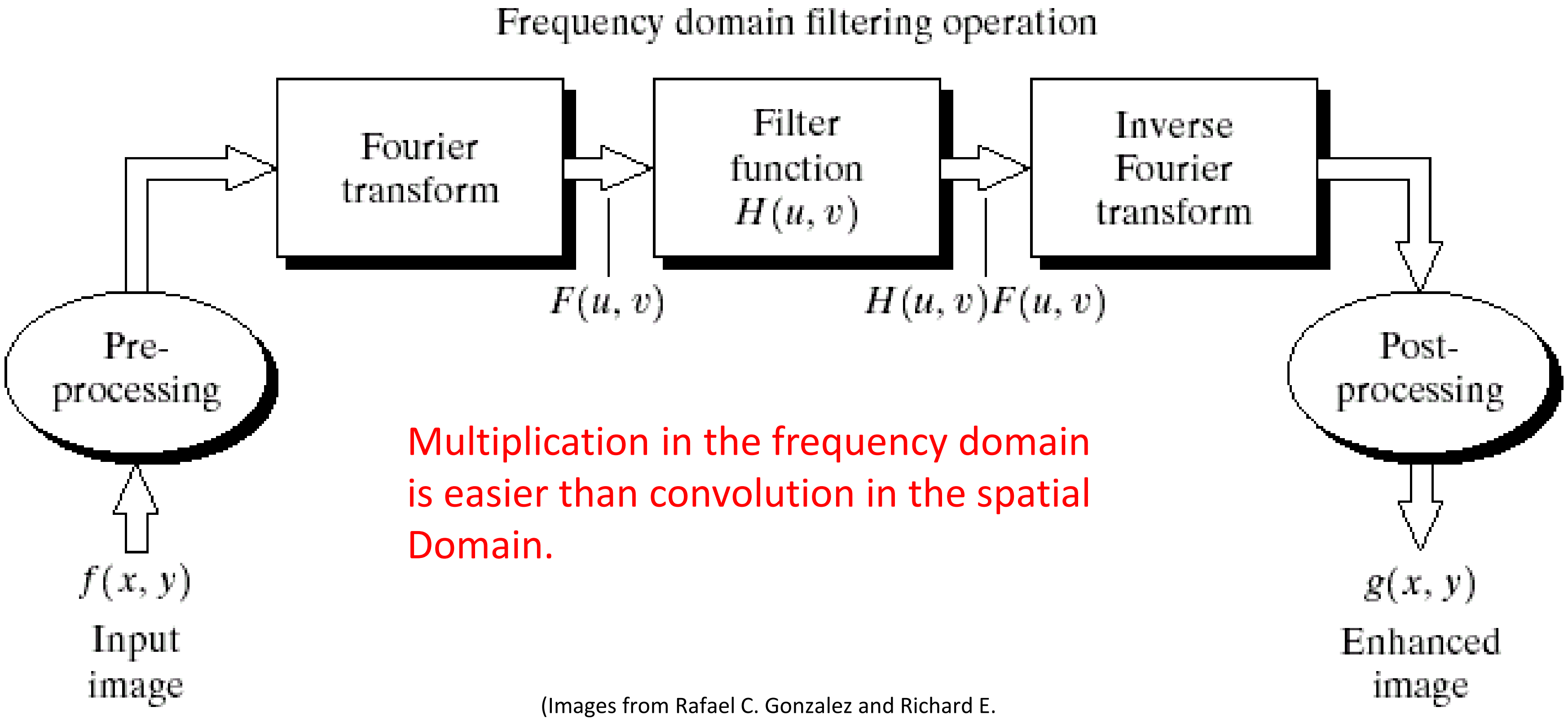


Basic Concept of Filtering in the Frequency Domain

From Fourier Transform Property:

$$g(x,y) = f(x,y) * h(x,y) \Leftrightarrow F(u,v) \cdot H(u,v) = G(u,v)$$

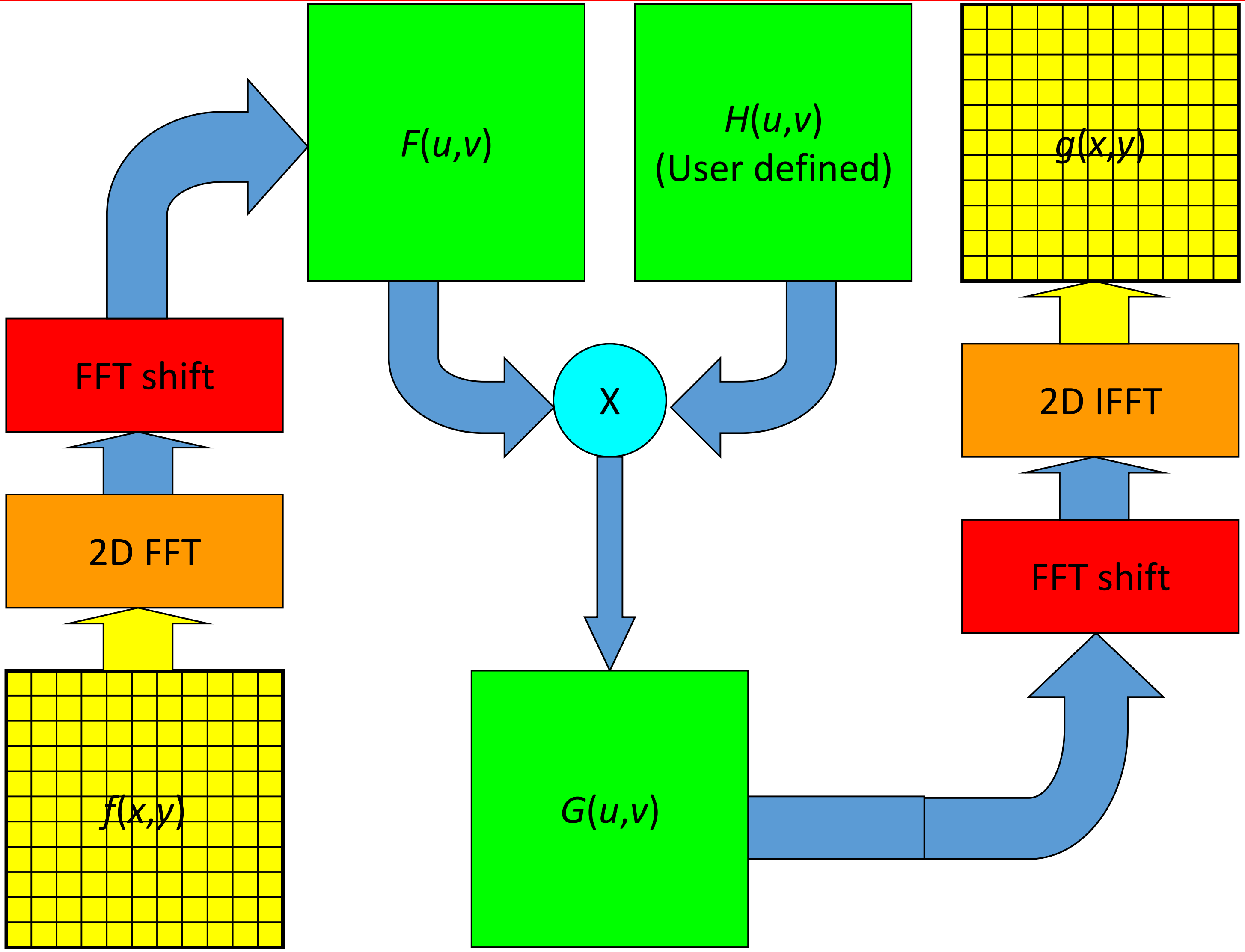
We cam perform filtering process by using



Multiplication in the frequency domain is easier than convolution in the spatial Domain.

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Filtering in the Frequency Domain with FFT shift



In this case, $F(u,v)$ and $H(u,v)$ must have the same size and have the zero frequency at the center.

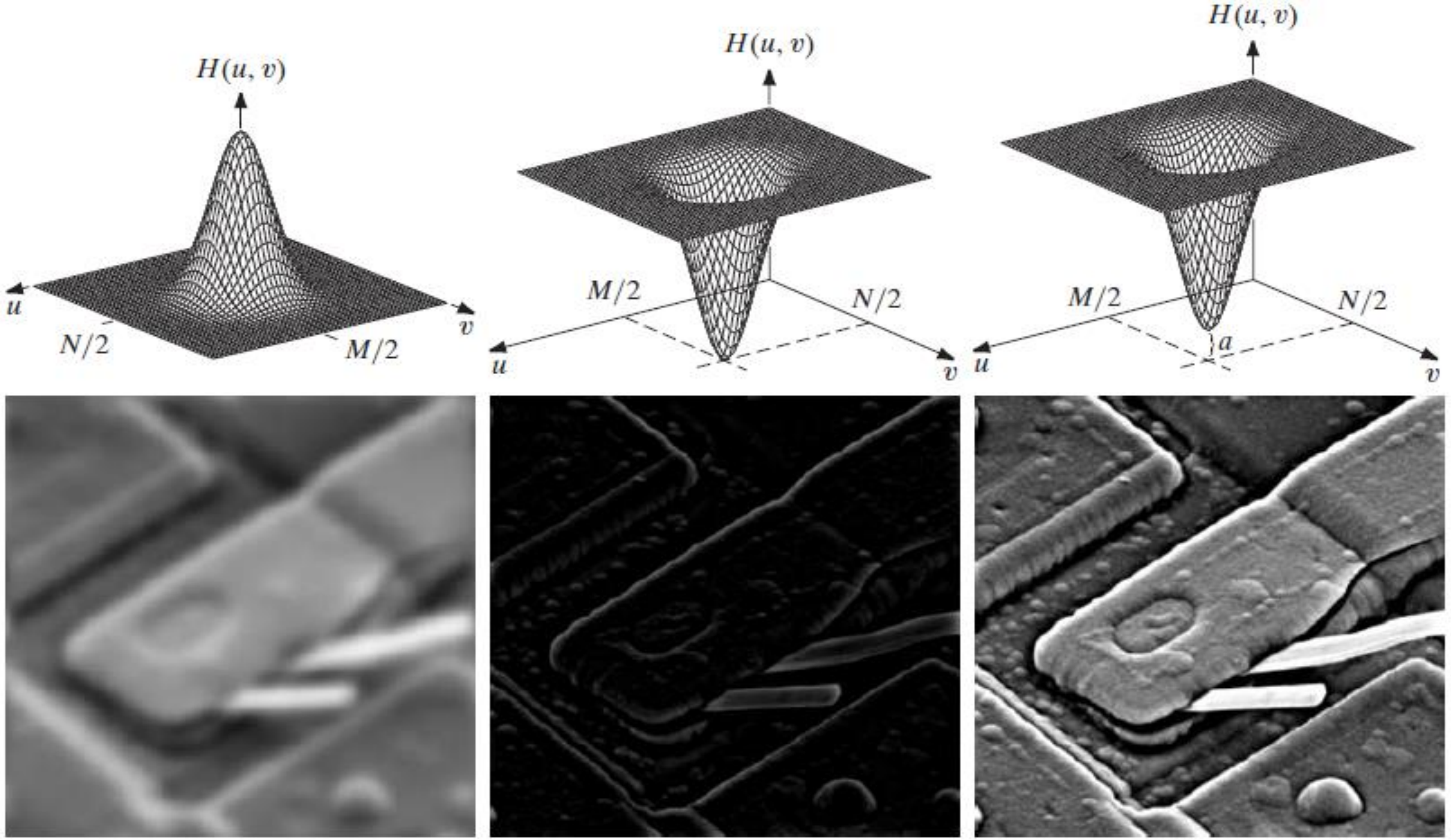
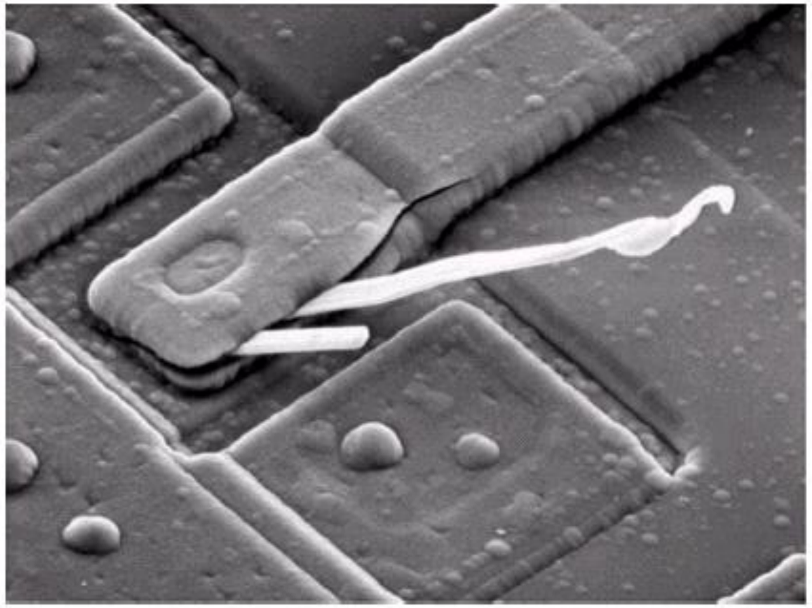
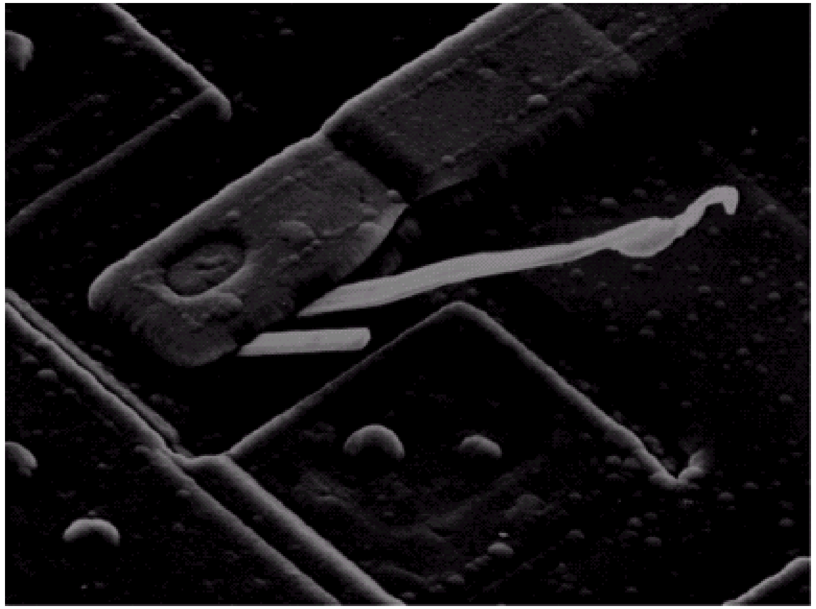


FIGURE 4.31 Top row: frequency domain filters. Bottom row: corresponding filtered images obtained using Eq. (4.7-1). We used $a = 0.85$ in (c) to obtain (f) (the height of the filter itself is 1). Compare (f) with Fig. 4.29(a).

Filtering in the Frequency Domain : Example



In this example, we set F(0,0) to zero which means that the zero frequency component is removed.



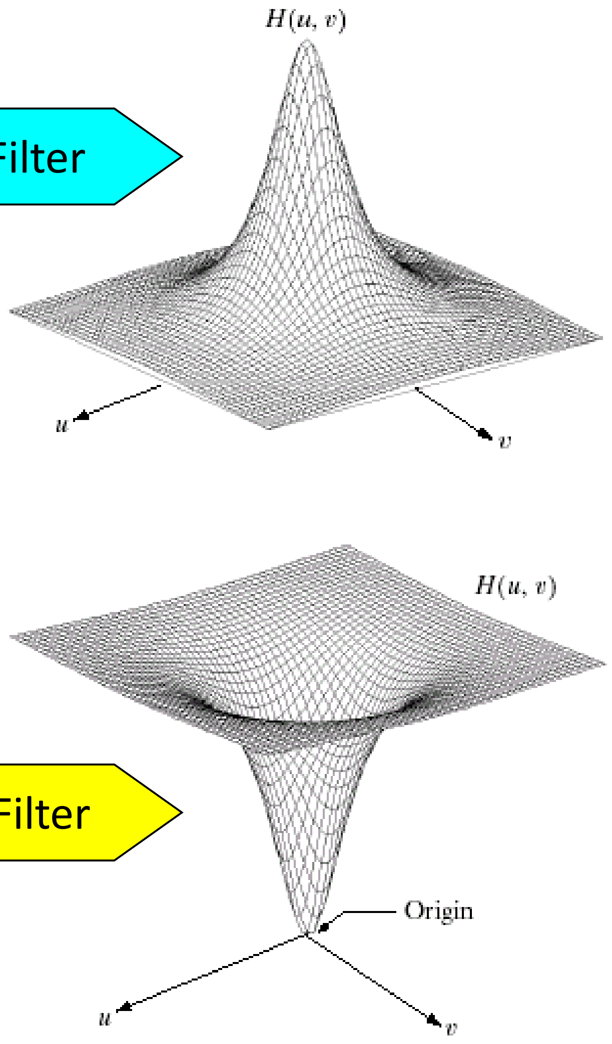
Note: Zero frequency = average intensity of an image

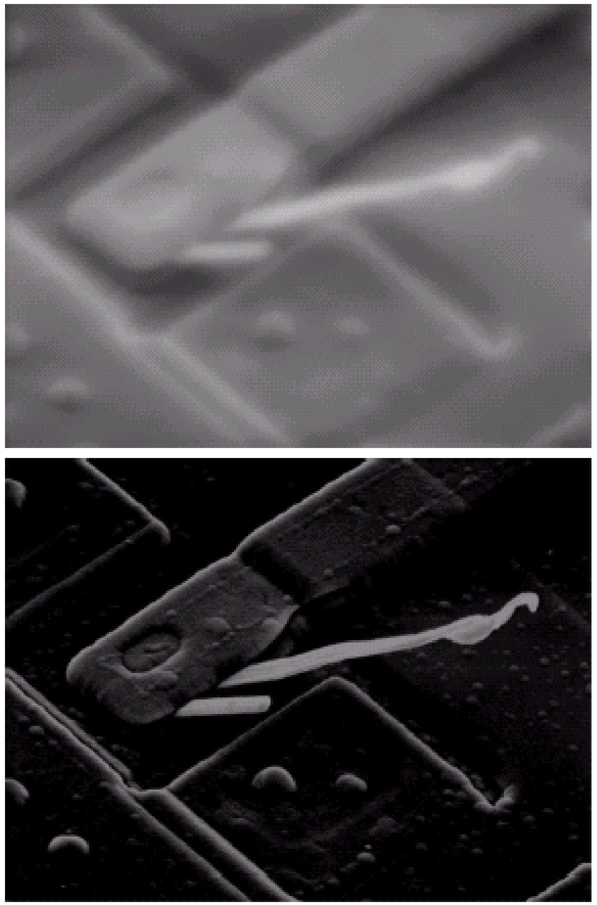
(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Filtering in the Frequency Domain : Example

Lowpass Filter

Highpass Filter





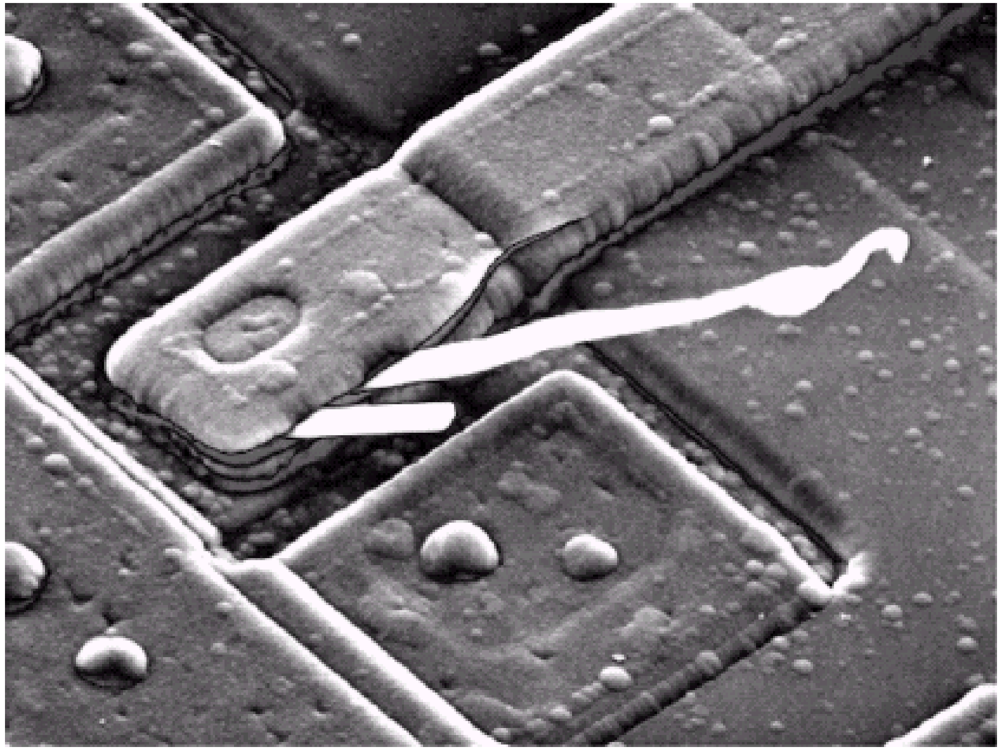
(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

a b
c d

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).

Filtering in the Frequency Domain : Example (cont.)

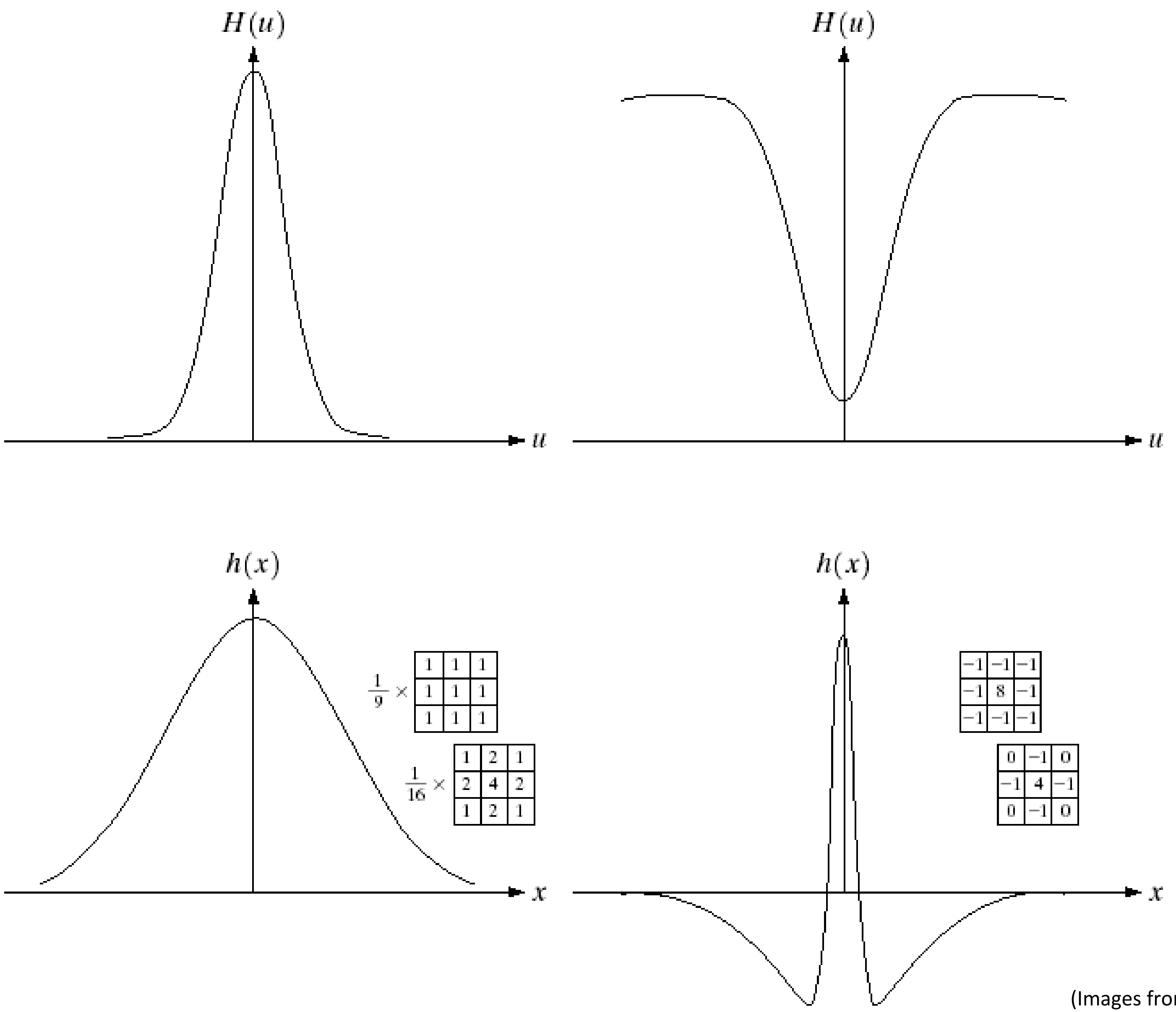
FIGURE 4.8
Result of highpass filtering the image in Fig. 4.4(a) with the filter in Fig. 4.7(c), modified by adding a constant of one-half the filter height to the filter function. Compare with Fig. 4.4(a).



Result of Sharpening Filter

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Filter Masks and Their Fourier Transforms



a b
c d

FIGURE 4.9
(a) Gaussian frequency domain lowpass filter. (b) Gaussian frequency domain highpass filter. (c) Corresponding lowpass spatial filter. (d) Corresponding highpass spatial filter. The masks shown are used in Chapter 3 for lowpass and highpass filtering.

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Ideal Lowpass Filter

Ideal LPF Filter Transfer function

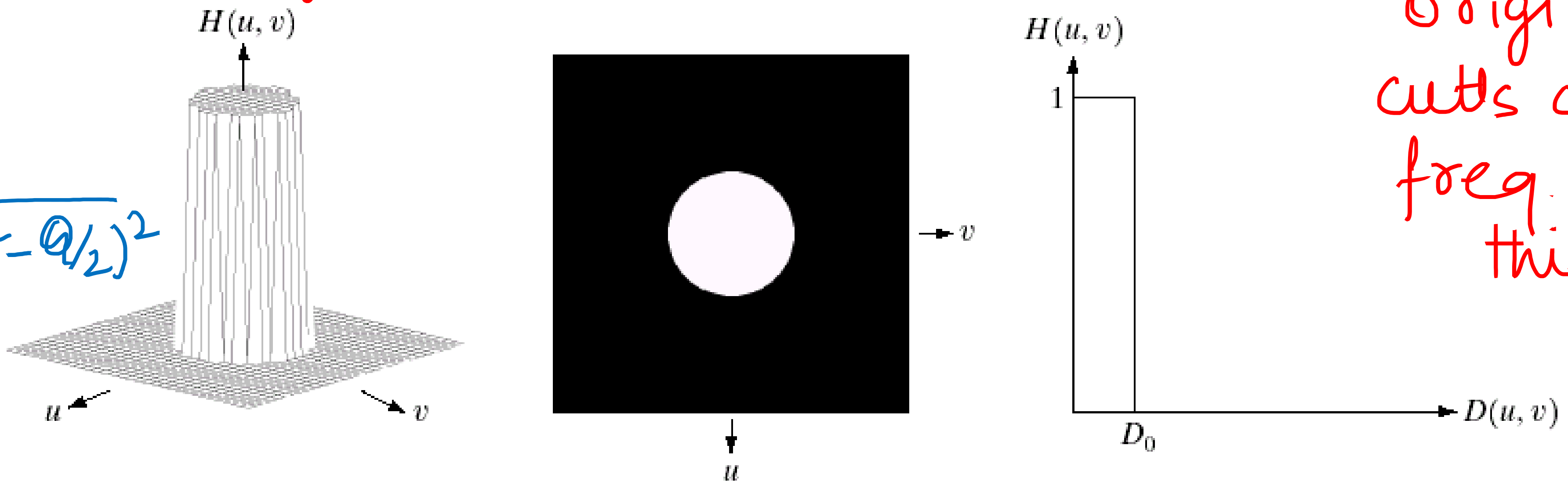
D(u,v) is distance b/w pt. (u,v) in freq domain & center of frequency rect

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

passes without attenuation all freq. within a circle of radius D_0 from the origin & cuts off all freq. outside this circle.

where $D(u,v)$ = Distance from (u,v) to the center of the mask.

$$D(u,v) = \sqrt{(u-P/2)^2+(v-Q/2)^2}$$



(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

a b c

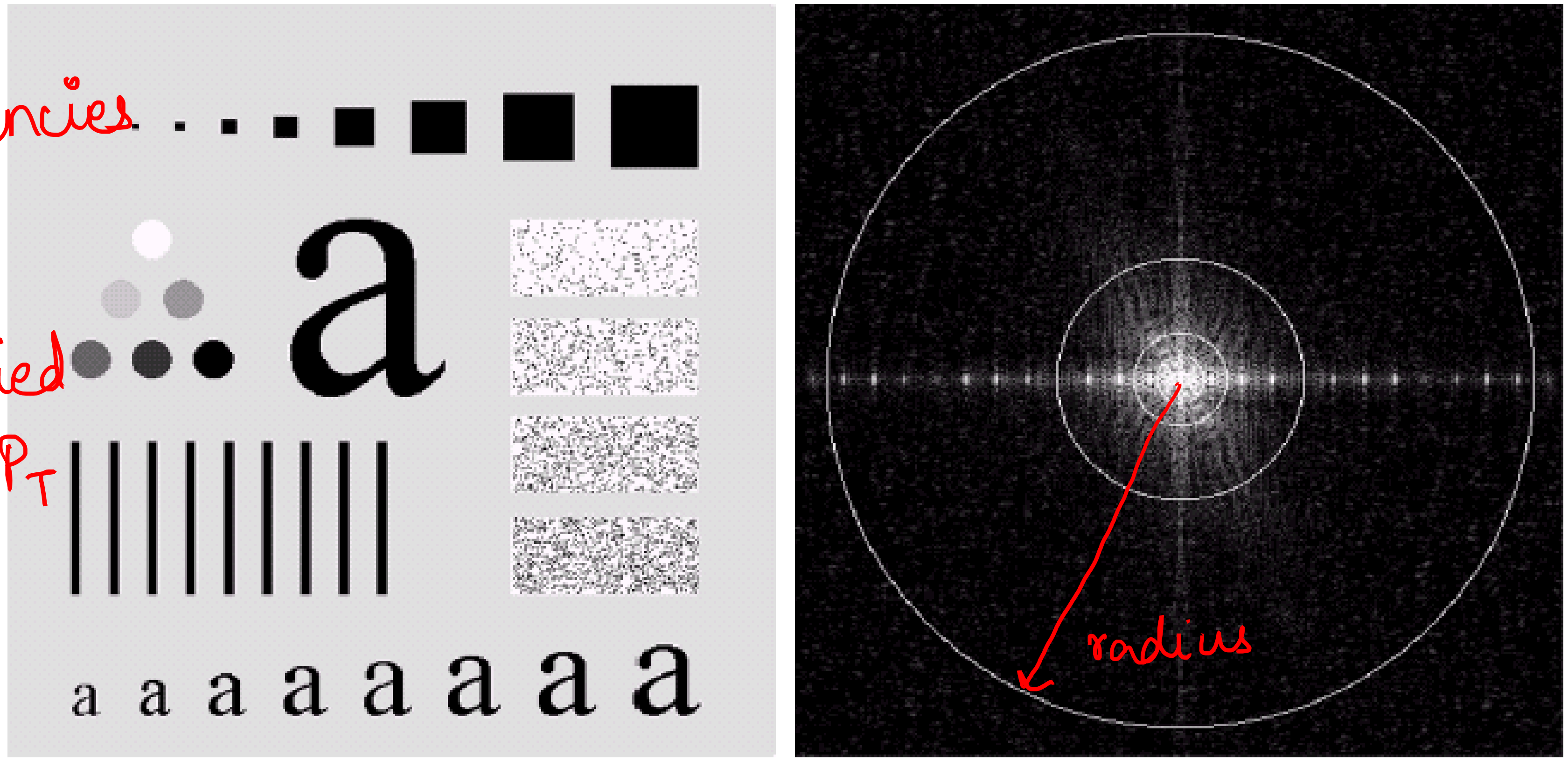
FIGURE 4.10 (a) Perspective plot of an ideal lowpass filter transfer function. (b) Filter displayed as an image. (c) Filter radial cross section.

Examples of Ideal Lowpass Filters

→ set of std cutoff frequencies

→ compute circles that enclose specified amt. of power P_T

Power spec $|F(u,v)|^2$



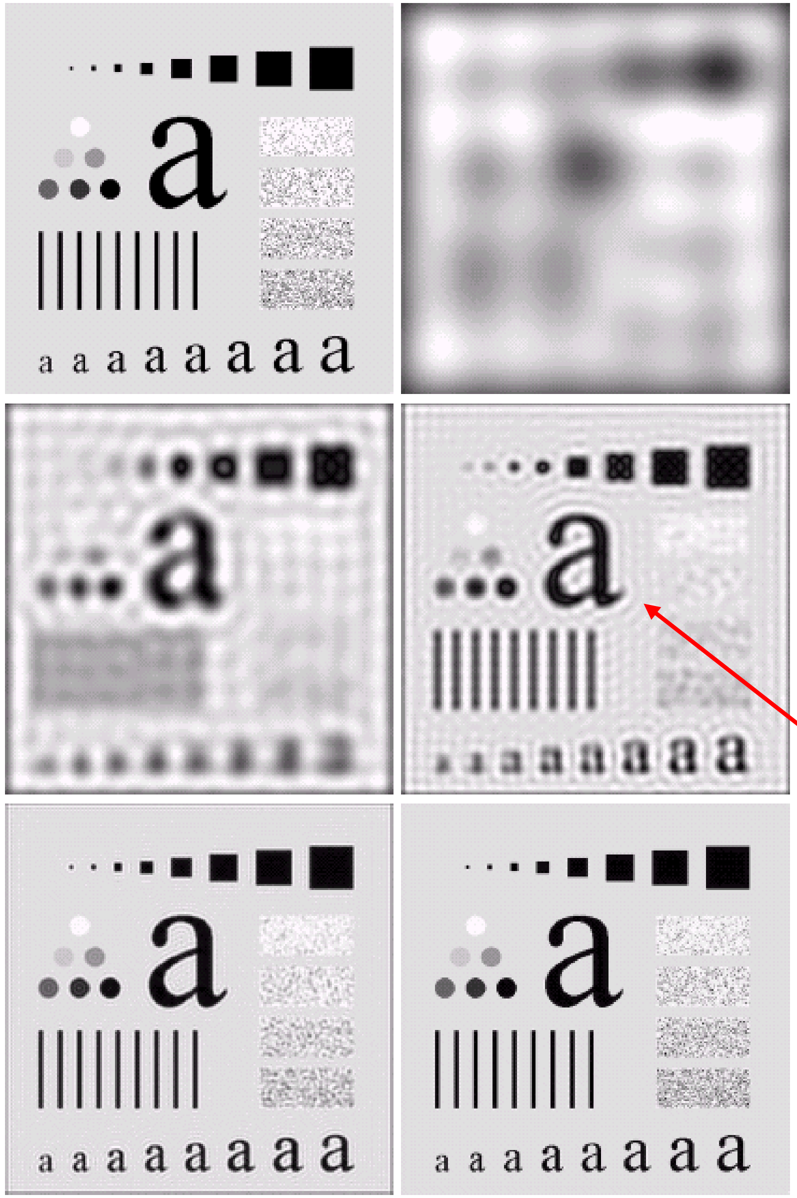
a b

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

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.

The smaller D_0, the more high frequency components are removed.

Results of Ideal Lowpass Filters



$D_0 = 10$

$D_0 = 30$

$D_0 = 60$

$D_0 = 160$

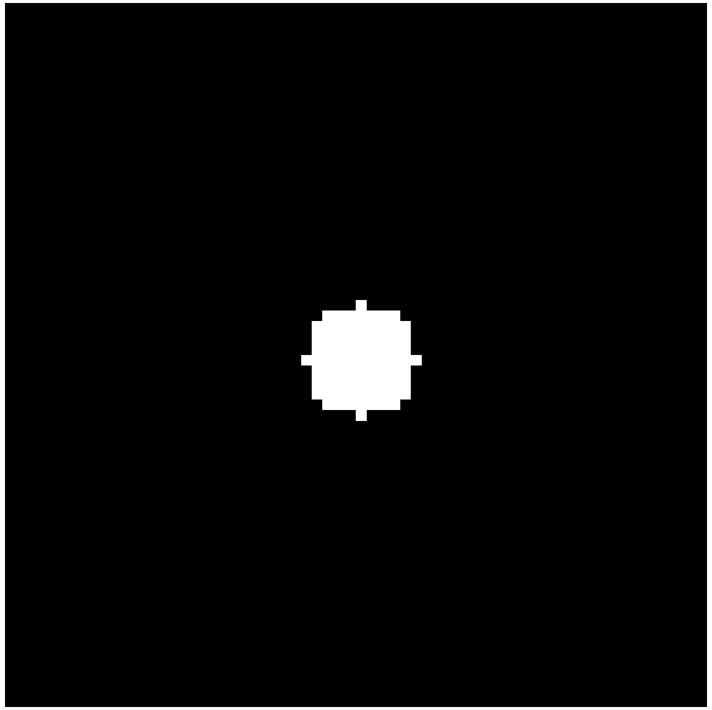
$D_0 = 460$

As the filter radius increases less & less power is removed, resulting in less blurring

Ringing effect can be obviously seen!

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.)

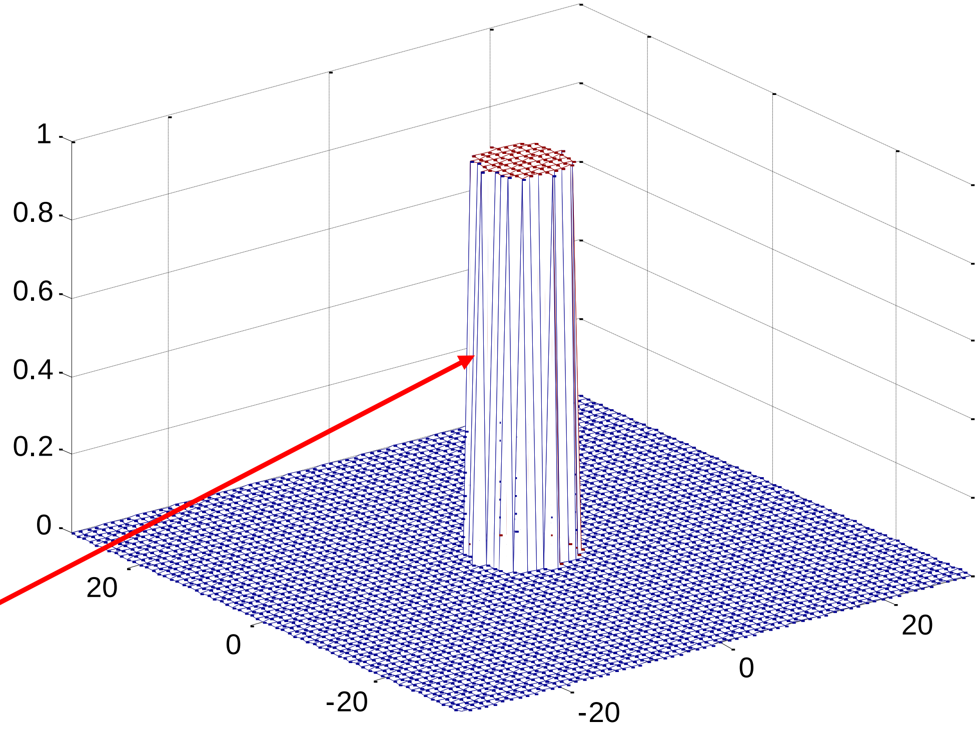
How ringing effect happens



Ideal Lowpass Filter
with D_0 = 5

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

Surface Plot

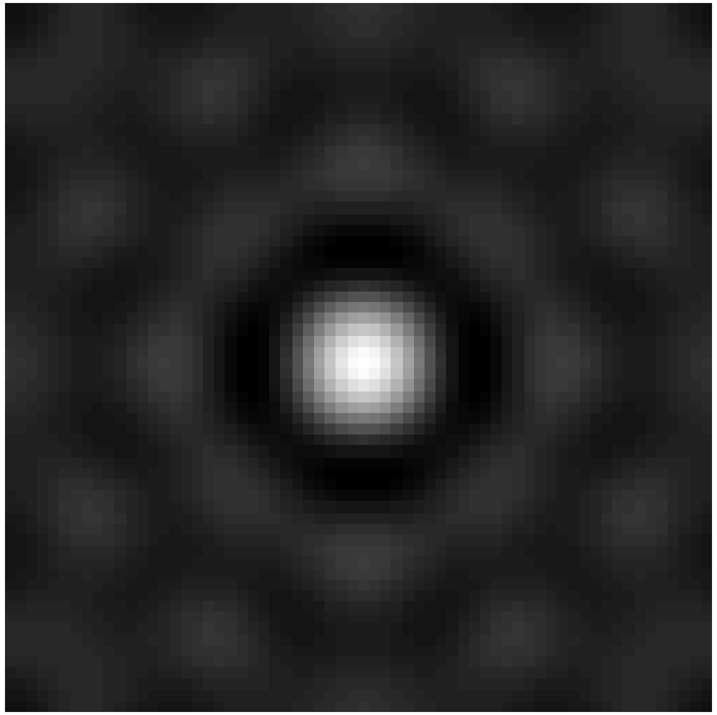


Abrupt change in the amplitude

- The blurring and ringing properties of ILPFs can be explained using the convolution theorem.
- Figure (a) on the next slide shows the spatial representation, $h(x,y)$, of an ILPF of radius 5, and Figure (b) shows the surface plot of a line passing through the center of the image.
- Because a cross section of the ILPF in the frequency domain looks like a box filter, it is not unexpected that a cross section of the corresponding spatial filter has the shape of a sinc function.
- Filtering in the spatial domain is done by convolving $h(x, y)$ with the image. Imagine each pixel in the image being a discrete impulse whose strength is proportional to the intensity of the image at that location. Convolving a sinc with an impulse copies the sinc at the location of the impulse.
- The center lobe of the sinc is the principal cause of blurring, while the outer, smaller lobes are mainly responsible for ringing.

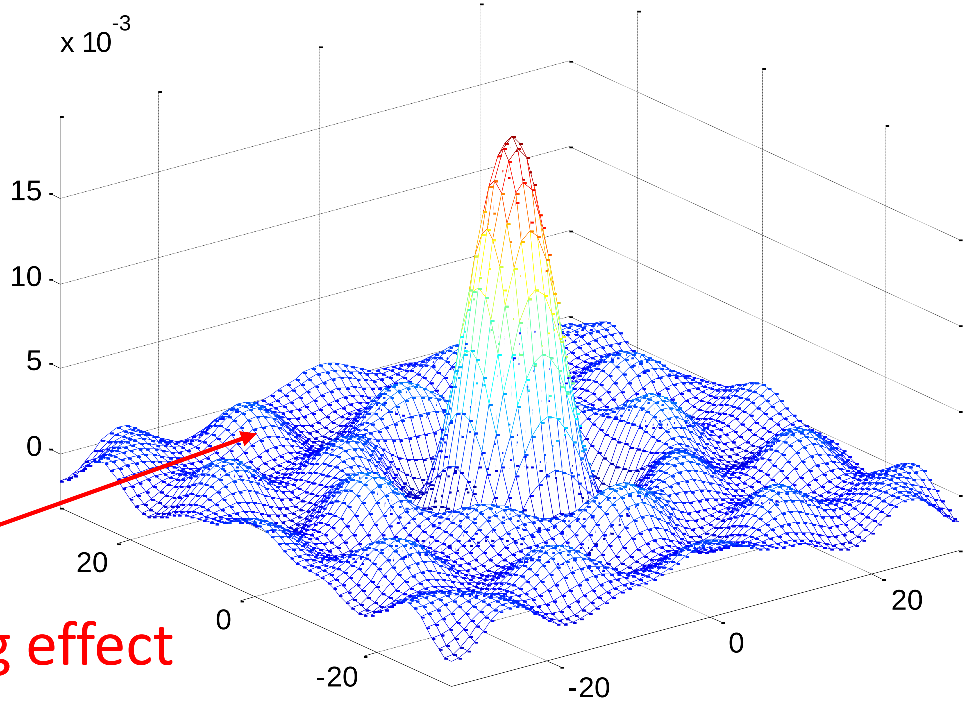
How ringing effect happens (cont.)

Larger the D_0 less ringing effect



Spatial Response of Ideal Lowpass Filter with $D_0 = 5$

Surface Plot

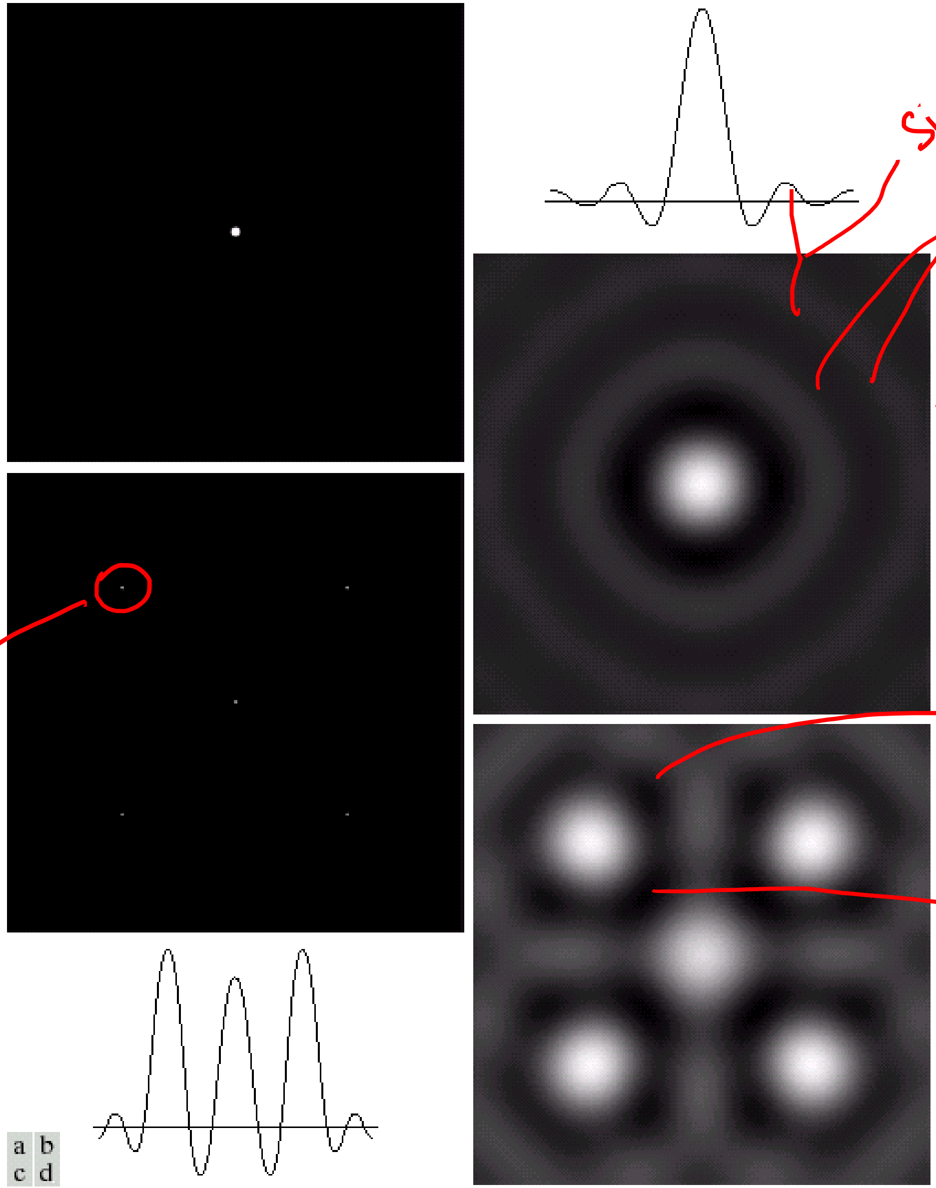


Ripples that cause ringing effect

How ringing effect happens (cont.)

$H(u,v)$
ILPF of radius 5

$f(x,y)$
impulse in spatial domain



sinc fn ripples
ringing

← corresponding spatial filter $h(x,y)$

$f(x,y) \ast h(x,y)$
↳ convolution

ringing effect

a b
c d

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.

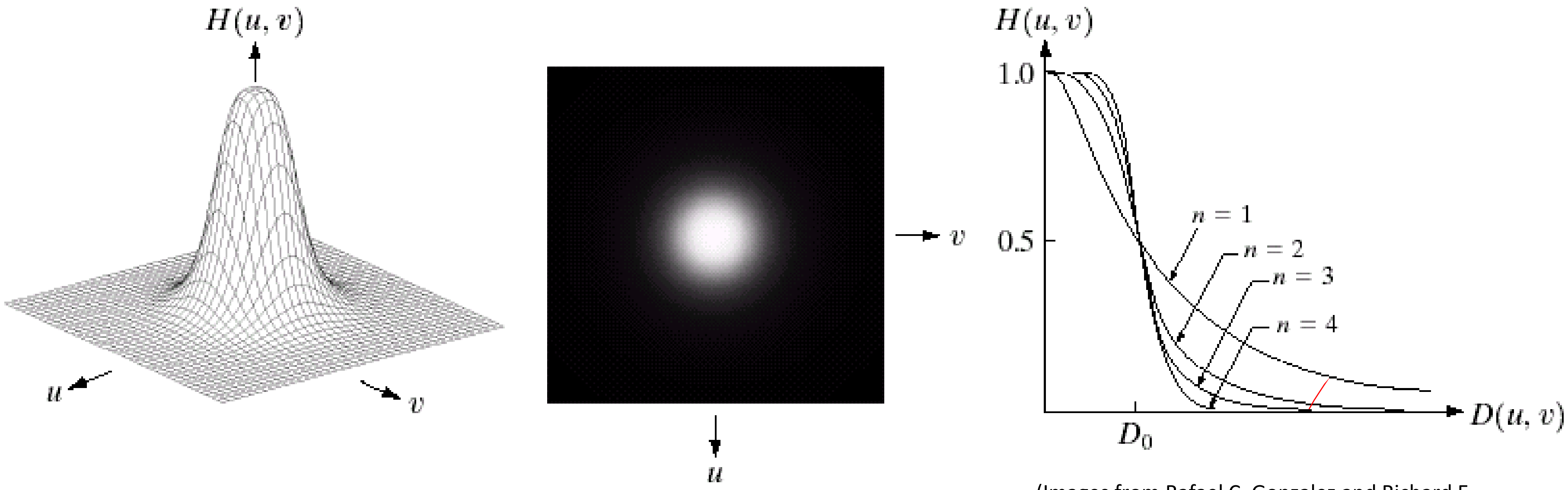
(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.)

Butterworth Lowpass Filter

Transfer function

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

Where D_0 = Cut off frequency, N = filter order.

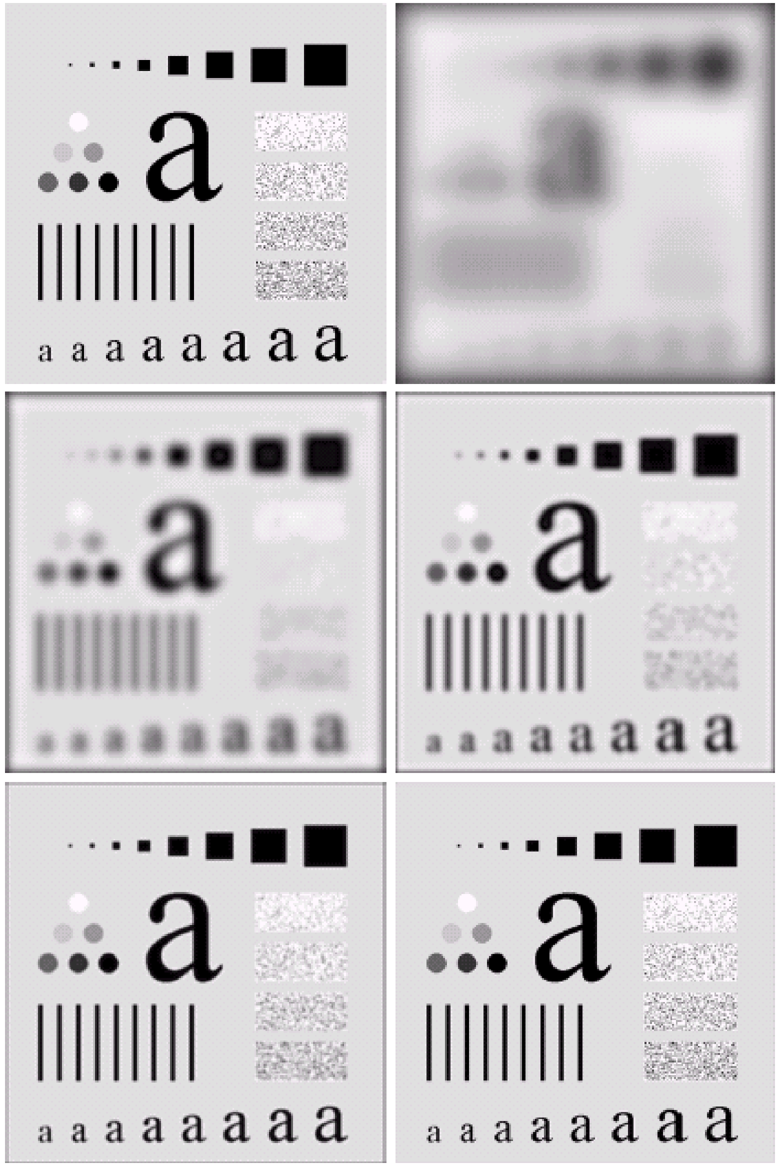


(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

a b c

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.

Results of Butterworth Lowpass Filters



$n = 2$, $D_0 = 10$

$n = 2$, $D_0 = 30$

$n = 2$, $D_0 = 60$

$n = 2$, $D_0 = 160$

$n = 2$, $D_0 = 460$

a b c d e f **FIGURE 4.15** (a) Original image. (b)–(f) Results of filtering with BLPFs of order 2, with cutoff frequencies at radii of 5, 15, 30, 80, and 230, as shown in Fig. 4.11(b). Compare with Fig. 4.12.

There is less ringing effect compared to those of ideal lowpass filters!

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Spatial Masks of the Butterworth Lowpass Filters

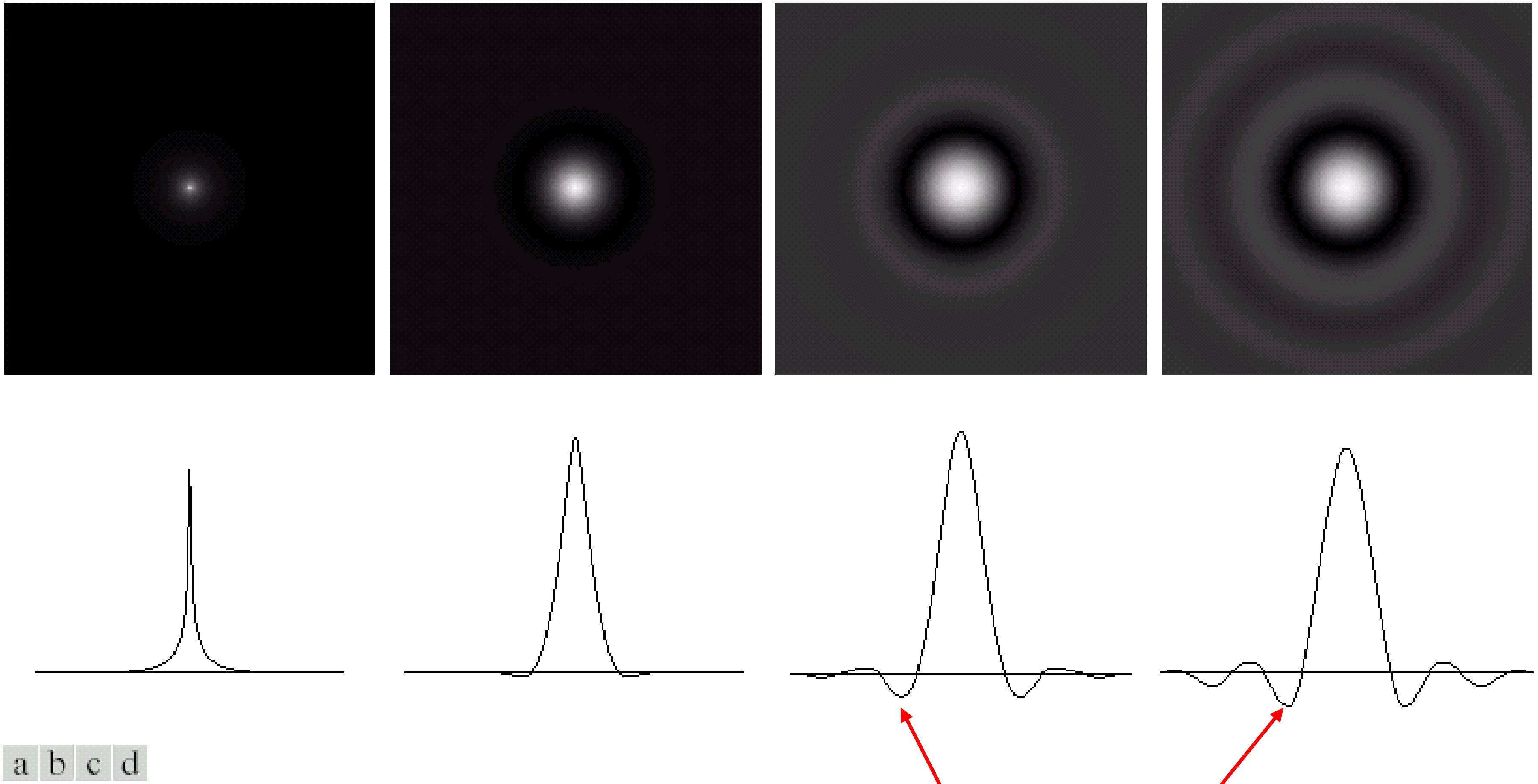


FIGURE 4.16 (a)–(d) Spatial representation of BLPFs of order 1, 2, 5, and 20, and corresponding gray-level profiles through the center of the filters (all filters have a cutoff frequency of 5). Note that ringing increases as a function of filter order.

Some ripples can be seen.

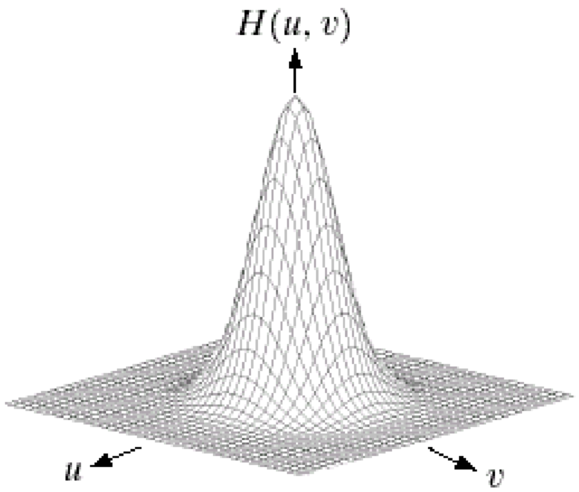
(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

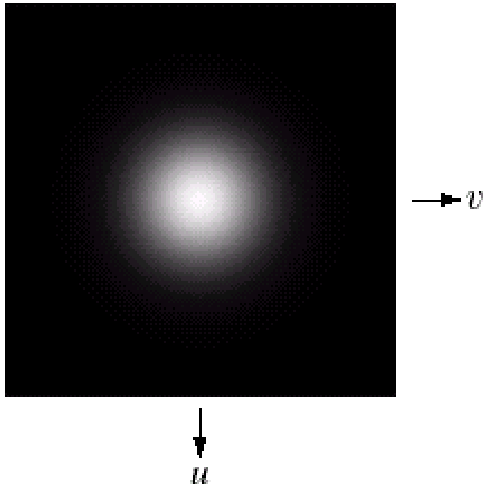
Gaussian Lowpass Filter

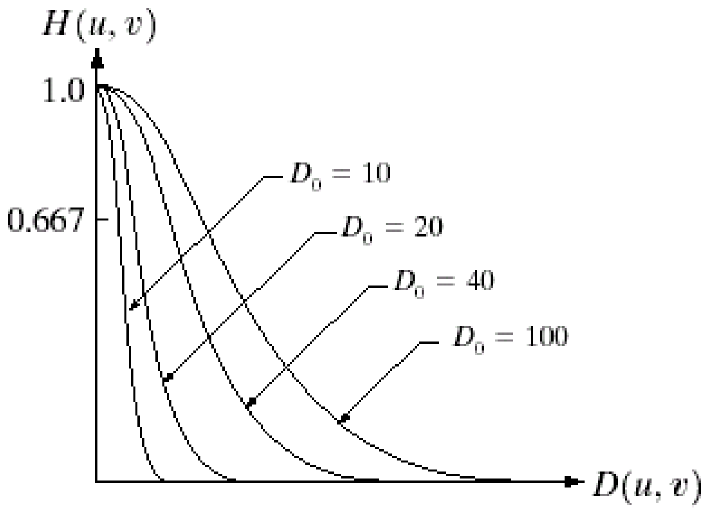
Transfer function

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

Where D_0 = spread factor.







(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

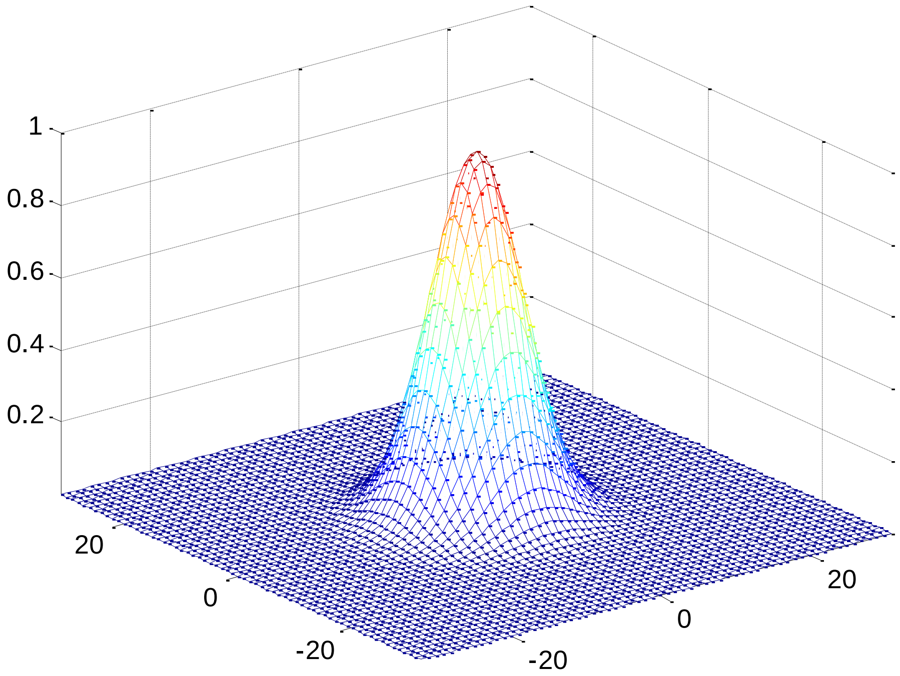
a b c

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.

Note: the Gaussian filter is the only filter that has no ripple and hence no ringing effect.

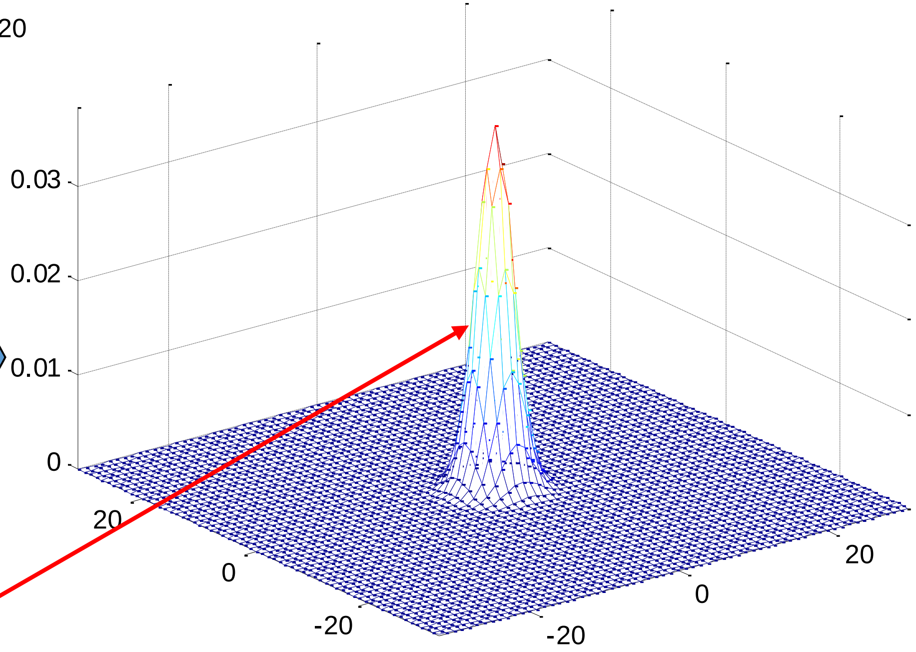
Gaussian Lowpass Filter (cont.)

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



Gaussian lowpass filter with $D_0 = 5$

Spatial respones of the Gaussian lowpass filter with $D_0 = 5$



Gaussian shape

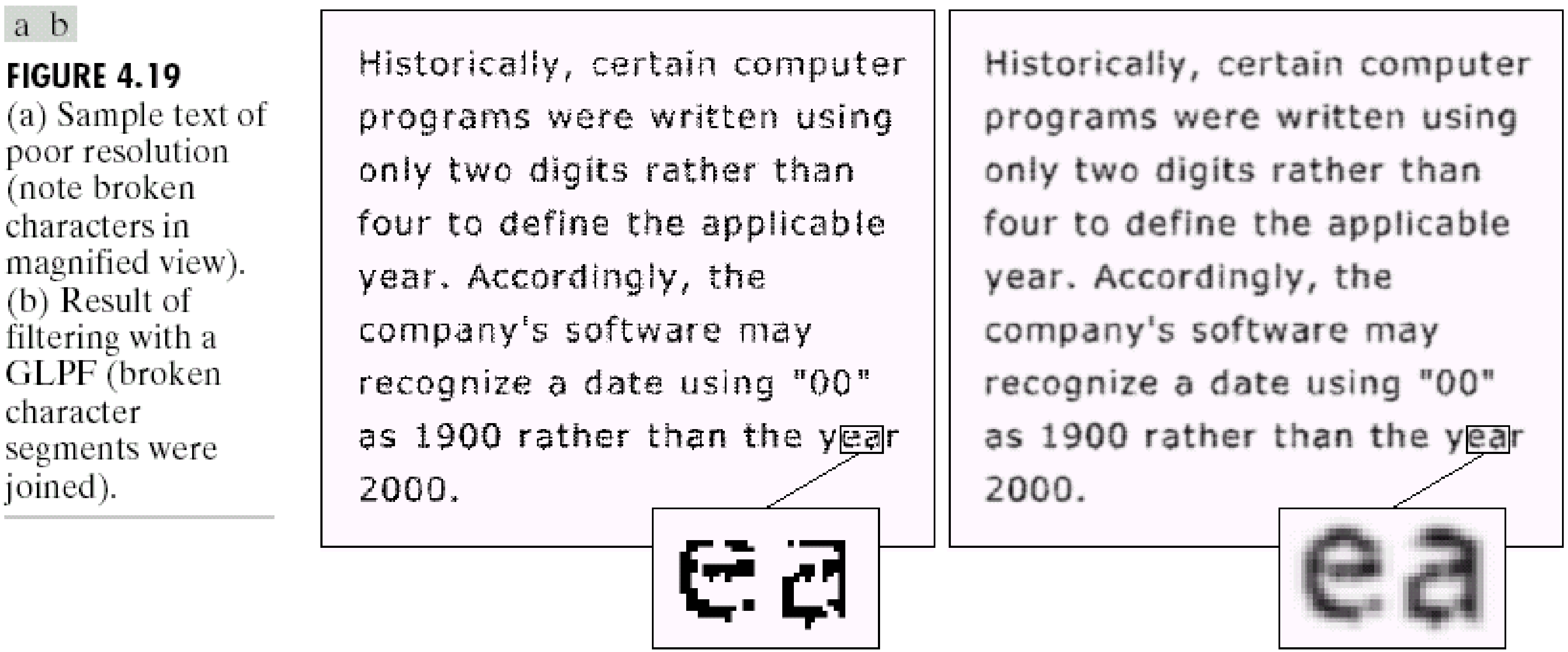
Results of Gaussian Lowpass Filters

FIGURE 4.18 (a) Original image. (b)–(f) Results of filtering with Gaussian lowpass filters with cutoff frequencies set at radii values of 5, 15, 30, 80, and 230, as shown in Fig. 4.11(b). Compare with Figs. 4.12 and 4.15.

No ringing effect!

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Application of Gaussian Lowpass Filters



Original image

Better Looking

The GLPF can be used to remove jagged edges and “repair” broken characters.

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Application of Gaussian Lowpass Filters (cont.)

Remove wrinkles

Original image

Softer-Looking

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).

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Application of Gaussian Lowpass Filters (cont.)

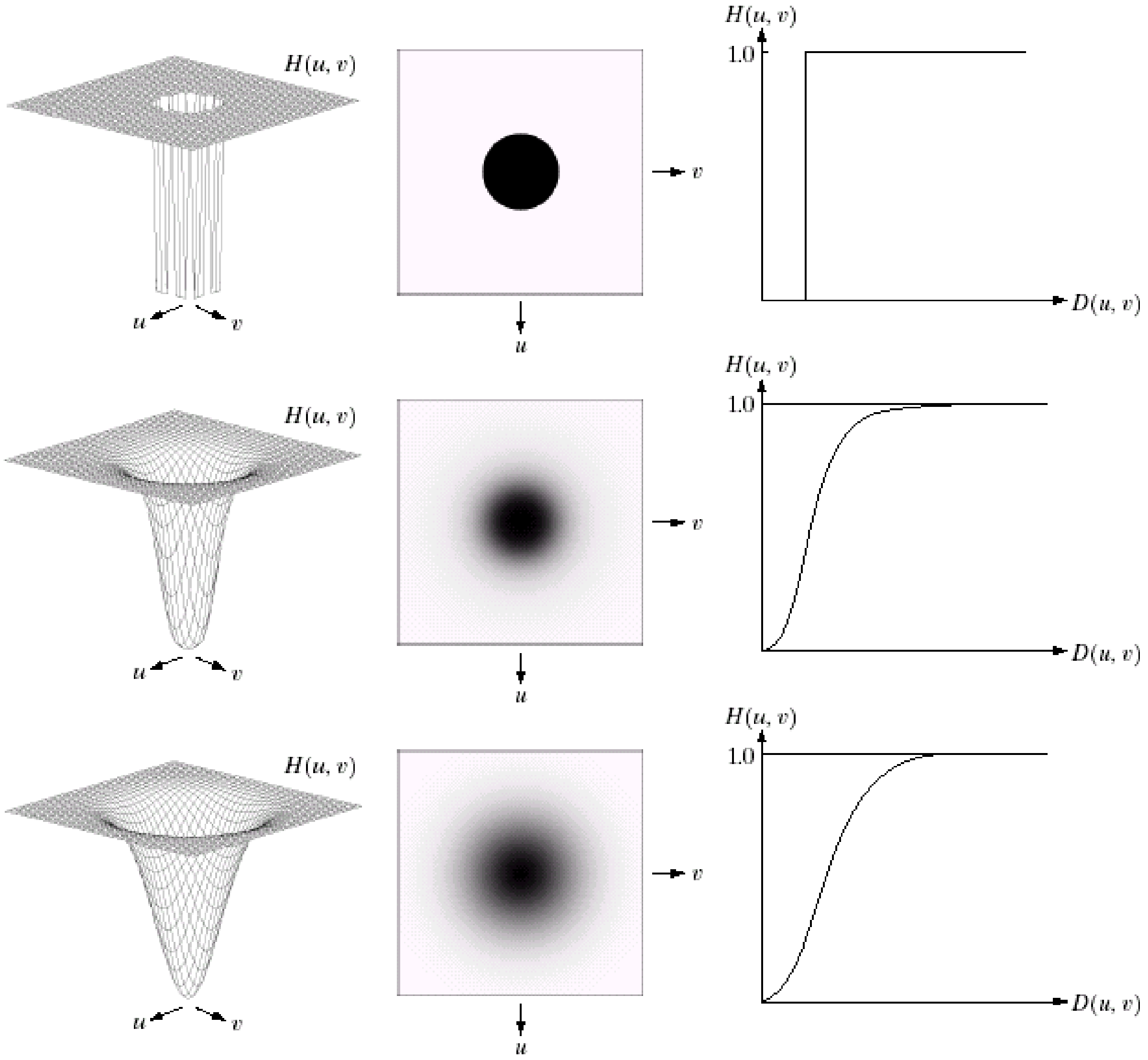
Original image : The gulf of Mexico and Florida from NOAA satellite.

a b c

FIGURE 4.51 (a) Image showing prominent horizontal scan lines. (b) Result of filtering using a GLPF with $D_0 = 50$. (c) Result of using a GLPF with $D_0 = 20$. (Original image courtesy of NOAA.)

Remove artifact lines: this is a simple but crude way to do it!

Highpass Filters



a b c
d e f
g h i

FIGURE 4.22 Top row: Perspective plot, image representation, and cross section of a typical ideal highpass filter. Middle and bottom rows: The same sequence for typical Butterworth and Gaussian highpass filters.

$$H_{hp} = 1 - H_{lp}$$

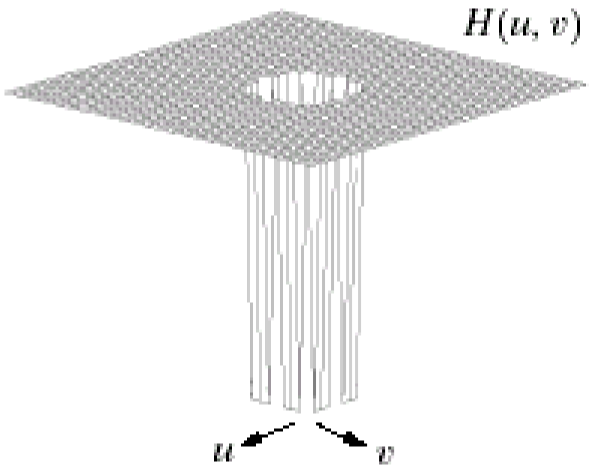
(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

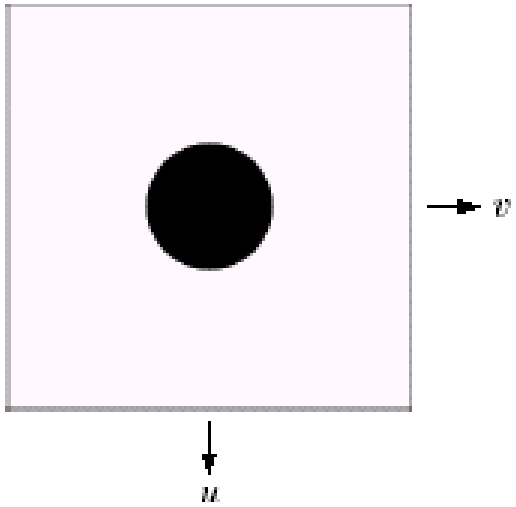
Ideal Highpass Filters

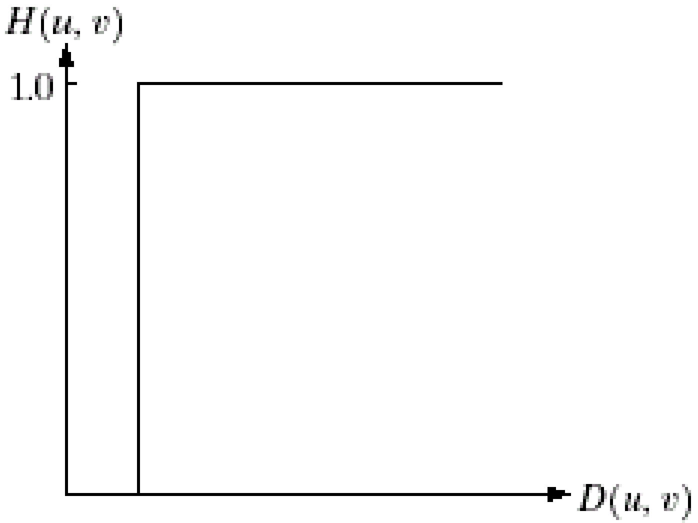
Ideal HPF Filter Transfer function

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

where $D(u,v)$ = Distance from (u,v) to the center of the mask.







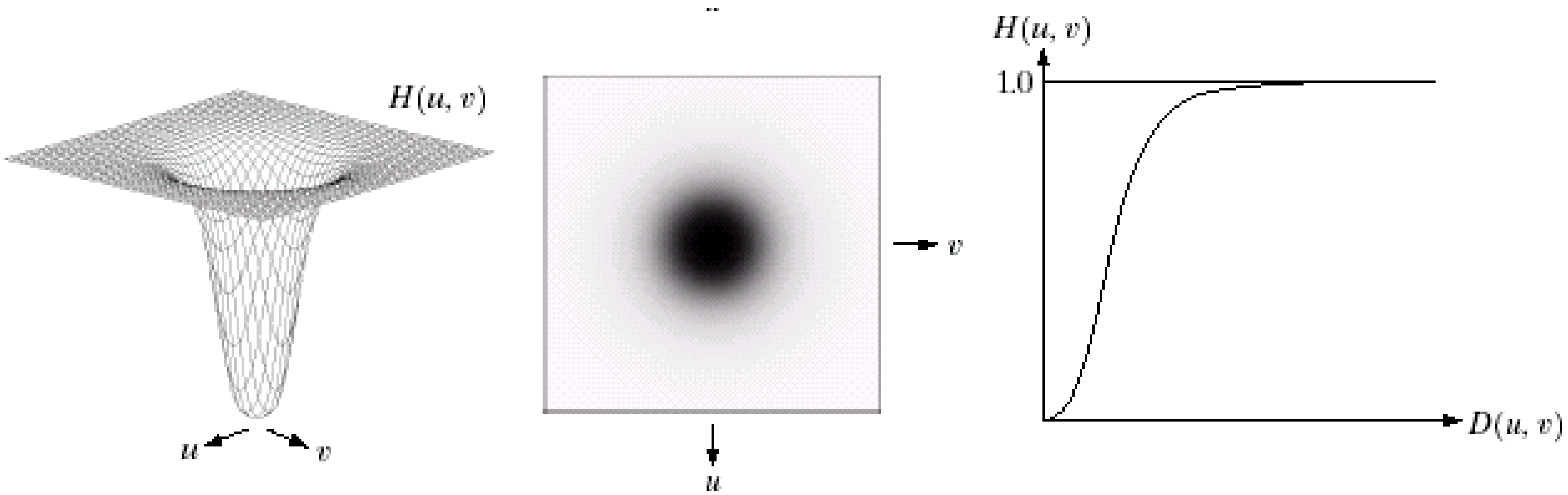
(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Butterworth Highpass Filters

Transfer function

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

Where D_0 = Cut off frequency, N = filter order.



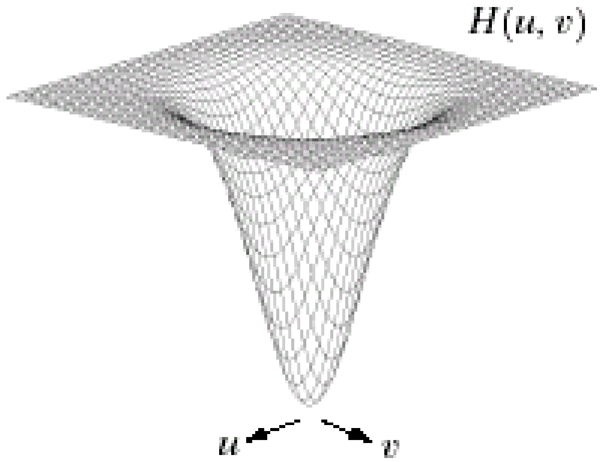
(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

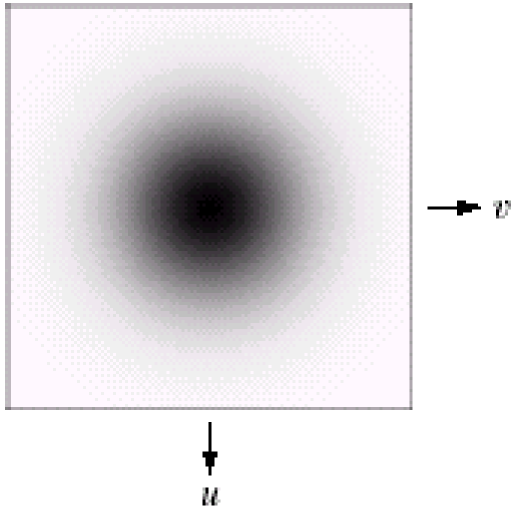
Gaussian Highpass Filters

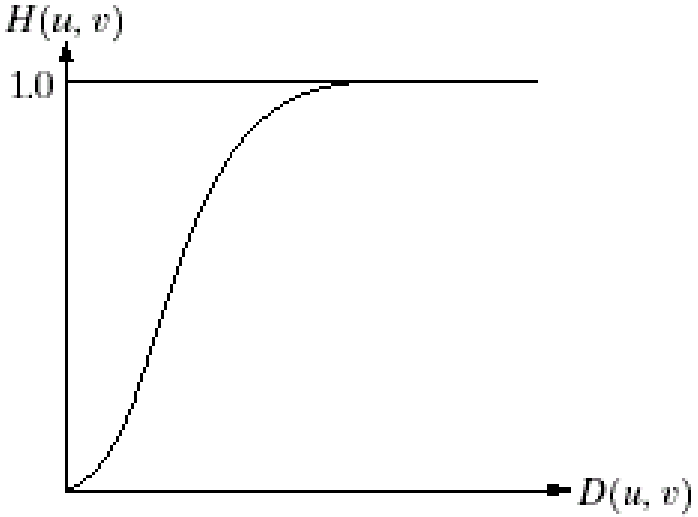
Transfer function

$$H(u,v) = 1 - e^{-D^2(u,v)/2D_0^2}$$

Where D_0 = spread factor.

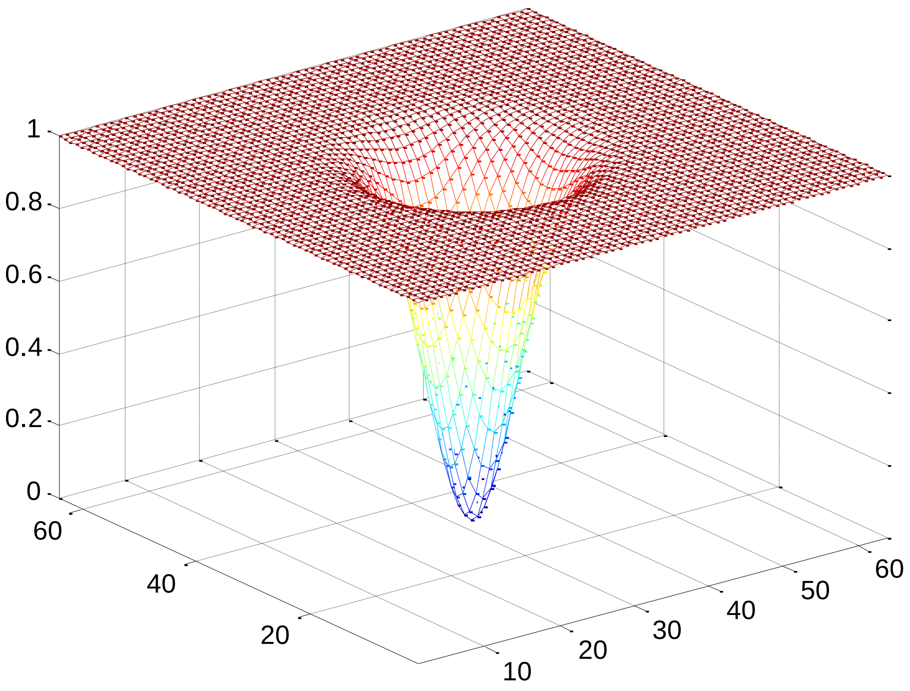






(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

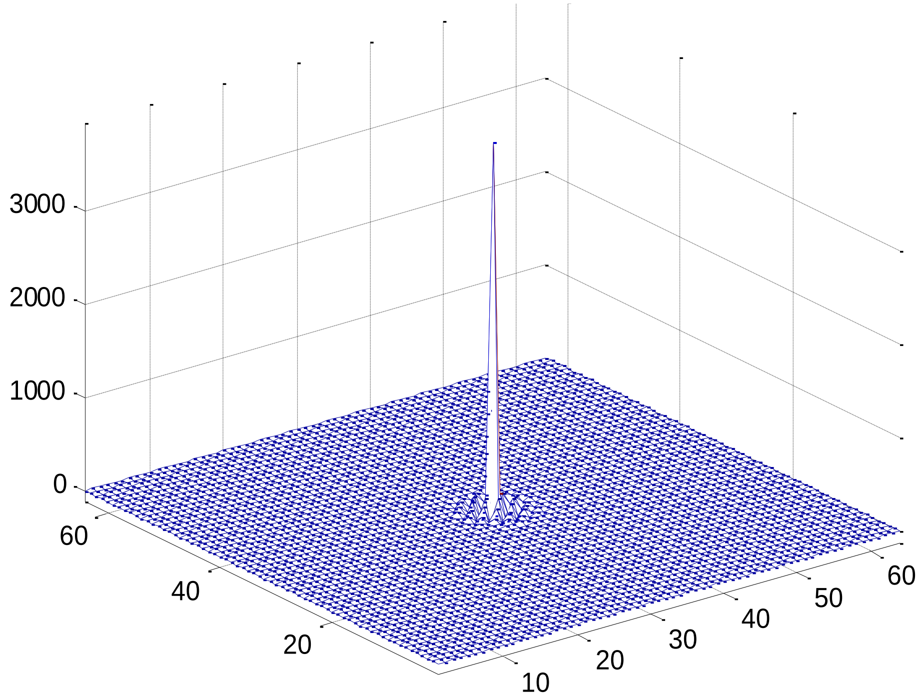
Gaussian Highpass Filters (cont.)



$$H(u,v) = 1 - e^{-D^2(u,v)/2D_0^2}$$

Gaussian highpass filter with D_0 = 5

Spatial respones of the Gaussian highpass filter with D_0 = 5



Spatial Responses of Highpass Filters

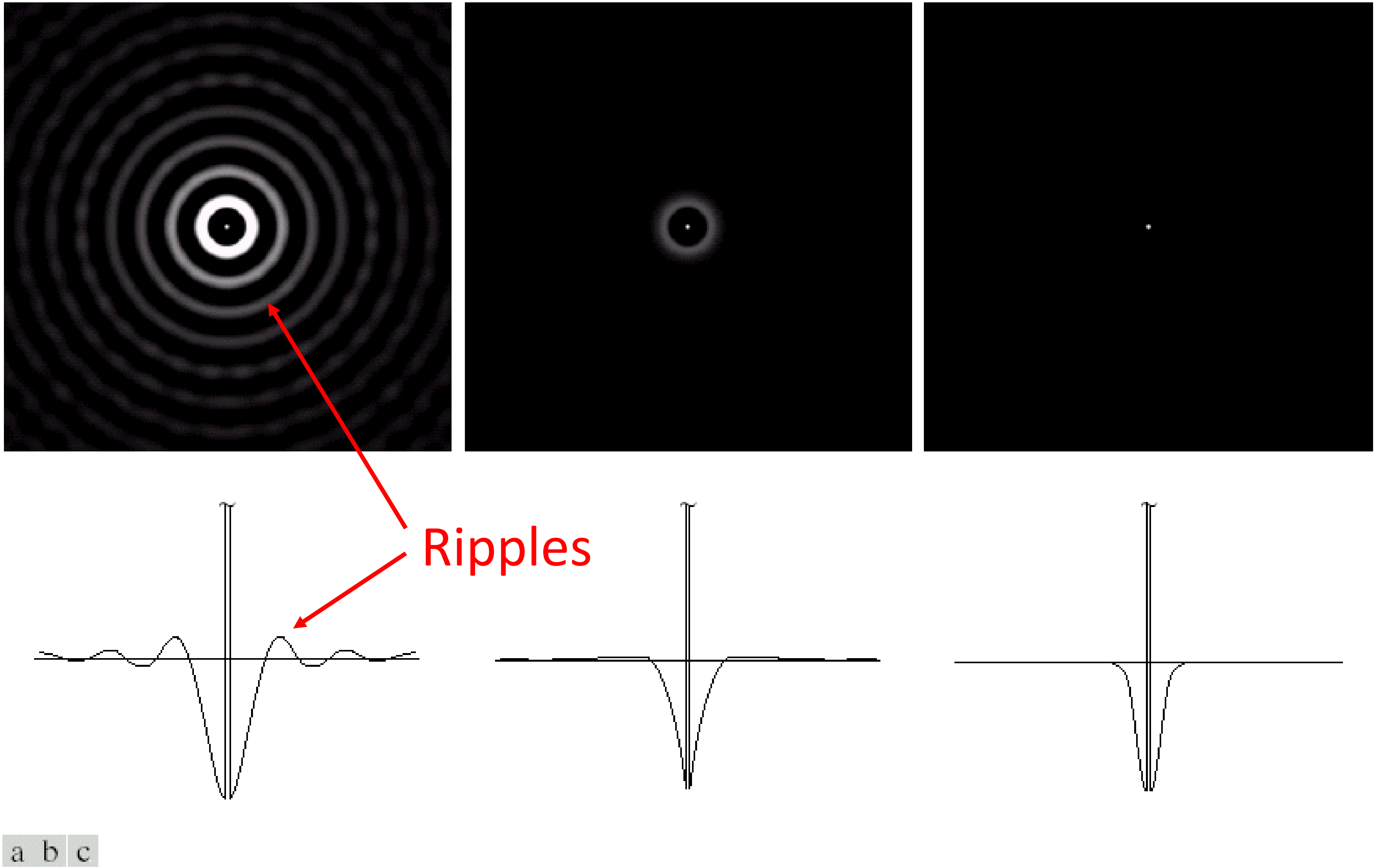
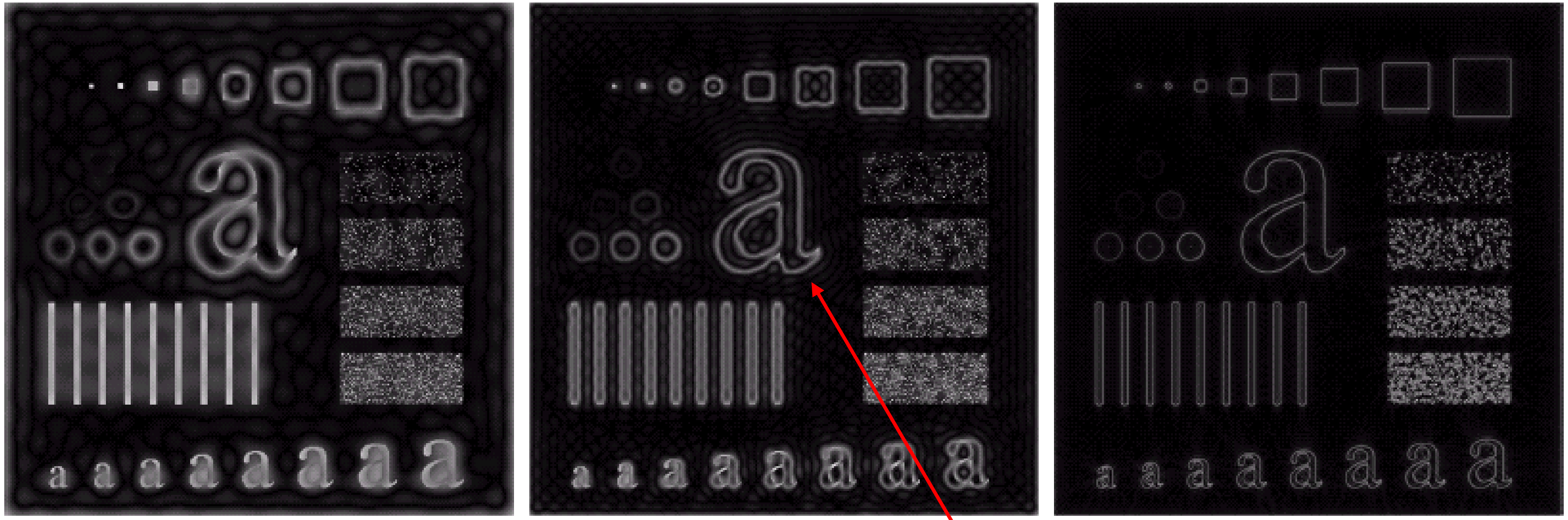


FIGURE 4.23 Spatial representations of typical (a) ideal, (b) Butterworth, and (c) Gaussian frequency domain highpass filters, and corresponding gray-level profiles.

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Results of Ideal Highpass Filters



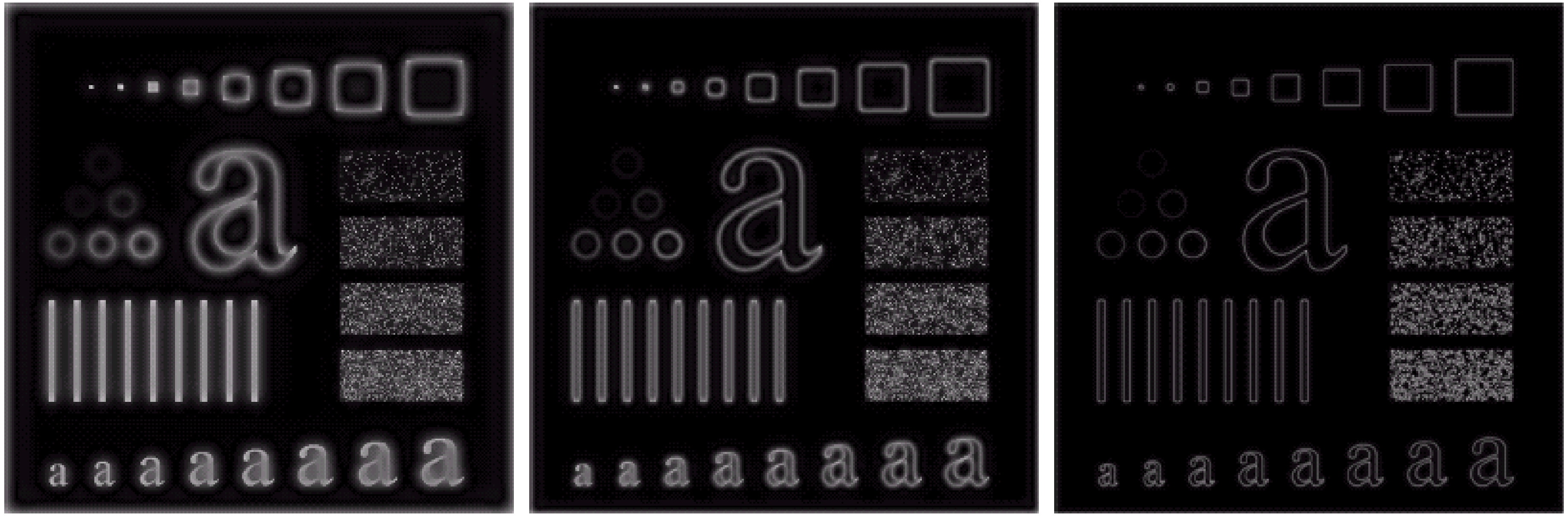
a b c

FIGURE 4.24 Results of ideal highpass filtering the image in Fig. 4.11(a) with $D_0 = 15$, 30, and 80, respectively. Problems with ringing are quite evident in (a) and (b).

Ringing effect can be obviously seen!

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Results of Butterworth Highpass Filters

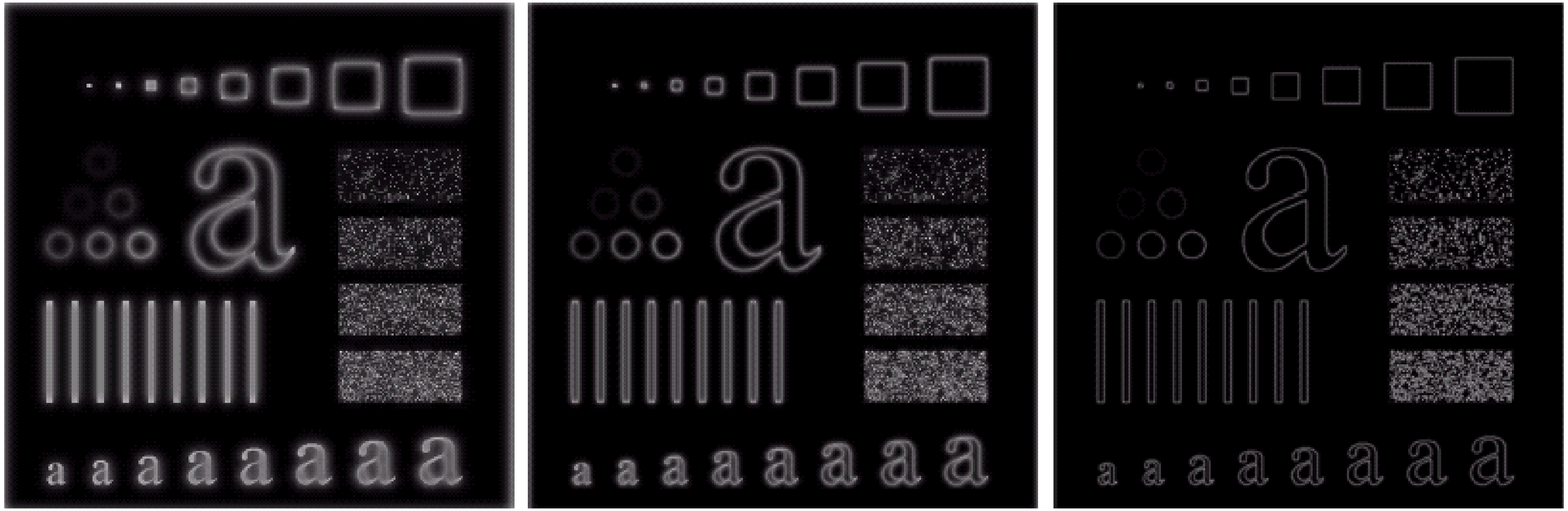


a b c

FIGURE 4.25 Results of highpass filtering the image in Fig. 4.11(a) using a BHPF of order 2 with $D_0 = 15$, 30, and 80, respectively. These results are much smoother than those obtained with an ILPF.

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Results of Gaussian Highpass Filters



a b c

FIGURE 4.26 Results of highpass filtering the image of Fig. 4.11(a) using a GHPF of order 2 with $D_0 = 15$, 30, and 80, respectively. Compare with Figs. 4.24 and 4.25.

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

TABLE 4.5
Highpass filters. D_0 is the cutoff frequency and n is the order of the Butterworth filter.

Ideal	Butterworth	Gaussian
$H(u,v) = \begin{cases} 1 & \text{if } D(u,v) \le D_0 \\ 0 & \text{if } D(u,v) > D_0 \end{cases}$	$H(u,v) = \dfrac{1}{1 + [D_0/D(u,v)]^{2n}}$	$H(u,v) = 1 - e^{-D^2(u,v)/2D_0^2}$

Additional examples of HPF

a b c

FIGURE 4.57 (a) Thumb print. (b) Result of highpass filtering (a). (c) Result of thresholding (b). (Original image courtesy of the U.S. National Institute of Standards and Technology.)

Laplacian in frequency domain:

Using the second derivative for image sharpening – The Laplacian (Revisit)

- Isotropic filters: rotation invariant
- Simplest isotropic second-order derivative operator: Laplacian

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

2-D Laplacian operation

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

x-direction

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

y-direction

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

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

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.

Using the second derivative for image sharpening – The Laplacian (revisit)

- Image enhancement (sharpening) by Laplacian operation

$$g(x,y)=\begin{cases} f(x,y)-\nabla^2 f(x,y) & \text{if the center coefficient of the Laplacian mask is negative} \\ f(x,y)+\nabla^2 f(x,y) & \text{if the center coefficient of the Laplacian mask is positive} \end{cases}$$

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

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.

Laplacian in frequency domain:

- Laplacian can be implemented in the frequency domain using the filter

$$H(u, v) = -4\pi^2(u^2 + v^2)$$

- or, with respect to the center of the frequency rectangle, using the filter

$$H(u, v) = -4\pi^2\left[(u - P/2)^2 + (v - Q/2)^2\right]$$

$$= -4\pi^2 D^2(u, v)$$

- The Laplacian image is obtained by:

$$\nabla^2 f(x, y) = \Im^{-1}\left\{H(u, v)F(u, v)\right\}$$

- Enhancement of the image is achieved using (H (u, v) negative):

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

Scaling in laplacian

- Introduction of large scaling factors.
- Practical solution:
- Normalize $f(x, y)$ to the range [0,1] before computing the DFT
- Divide $\nabla^2 f(x, y)$ by its maximum value to bring it in range [-1,1]

- In the frequency domain

$$
\begin{aligned}
g(x, y) &= \Im^{-1}\{F(u, v) - H(u, v)F(u, v)\} \\
&= \Im^{-1}\{[1 - H(u, v)]F(u, v)\} \\
&= \Im^{-1}\{[1 + 4\pi^2 D^2(u, v)]F(u, v)\}
\end{aligned}
$$

a b

FIGURE 4.58
(a) Original, blurry image.
(b) Image enhanced using the Laplacian in the frequency domain. Compare with Fig. 3.38(e).

Unsharp Masking, Highboost Filtering and High-Frequency-Emphasis Filtering:

Unsharp Masking and Highboost Filtering in spatial domain (revisit)

- 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

Perv Sems jevuj che
This process is called
Highboost Filtering if
multiplied
by constant >1

Unsharp Masking and Highboost Filtering in spatial domain (revisit)

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.

a
b
c
d

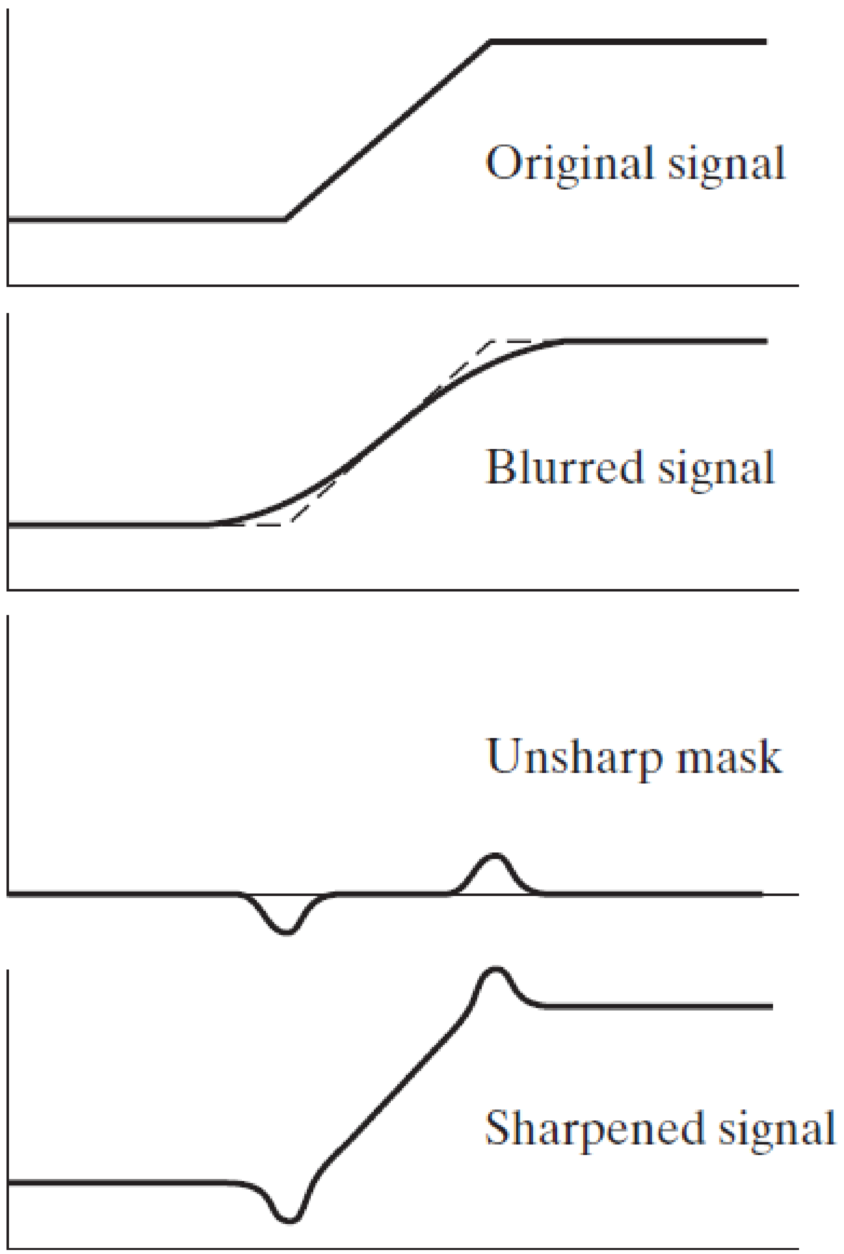


FIGURE 3.39 1-D illustration of the mechanics of unsharp masking. (a) Original signal. (b) Blurred signal with original shown dashed for reference. (c) Unsharp mask. (d) Sharpened signal, obtained by adding (c) to (a).

Unsharp Masking, Highboost Filtering and High-Frequency-Emphasis Filtering:

- In this section, we discuss frequency domain formulations of the unsharp masking and high-boost filtering image sharpening techniques

$$g_{\text{mask}}(x, y) = f(x, y) - f_{\text{LP}}(x, y)$$

$$f_{\text{LP}}(x, y) = \Im^{-1}\left[H_{\text{LP}}(u, v)F(u, v)\right]$$

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

- This expression defines the unsharp masking when $k = 1$ and highboost filtering when $k > 1$.
- we can express $g\,(x,\ y)$ entirely in terms of frequency domain computations involving a lowpass filter:

$$\mathcal{F}(g_{mask}(x,y)) = \mathcal{F}(f(x,y)) - \mathcal{F}(f_{LP}(x,y))$$

$$= F(u,v) - H_{LP}(u,v)F(u,v)$$

Transformed (sharpened) image in spatial domain

$$\mathcal{F}(g(x,y)) = \mathcal{F}(f(x,y)) + \mathcal{F}(K * g_{mask}(x,y))$$

$$= F(u,v) + K * (F(u,v) - H_{LP}(u,v)F(u,v))$$

$$g(x,y) = \mathcal{F}^{-1}\left\{\left[1 + K * \underbrace{\left[1 - H_{LP}(u,v)\right]}_{H_{HP}(u,v)}\right] F(u,v)\right\}$$

$$g(x,y) = \mathcal{F}^{-1}\left\{\left[K_1 + K_2 * H_{HP}(u,v)\right] F(u,v)\right\}$$

$$g(x, y) = \Im^{-1}\left\{[1 + k * H_{\mathrm{HP}}(u, v)]F(u, v)\right\}$$

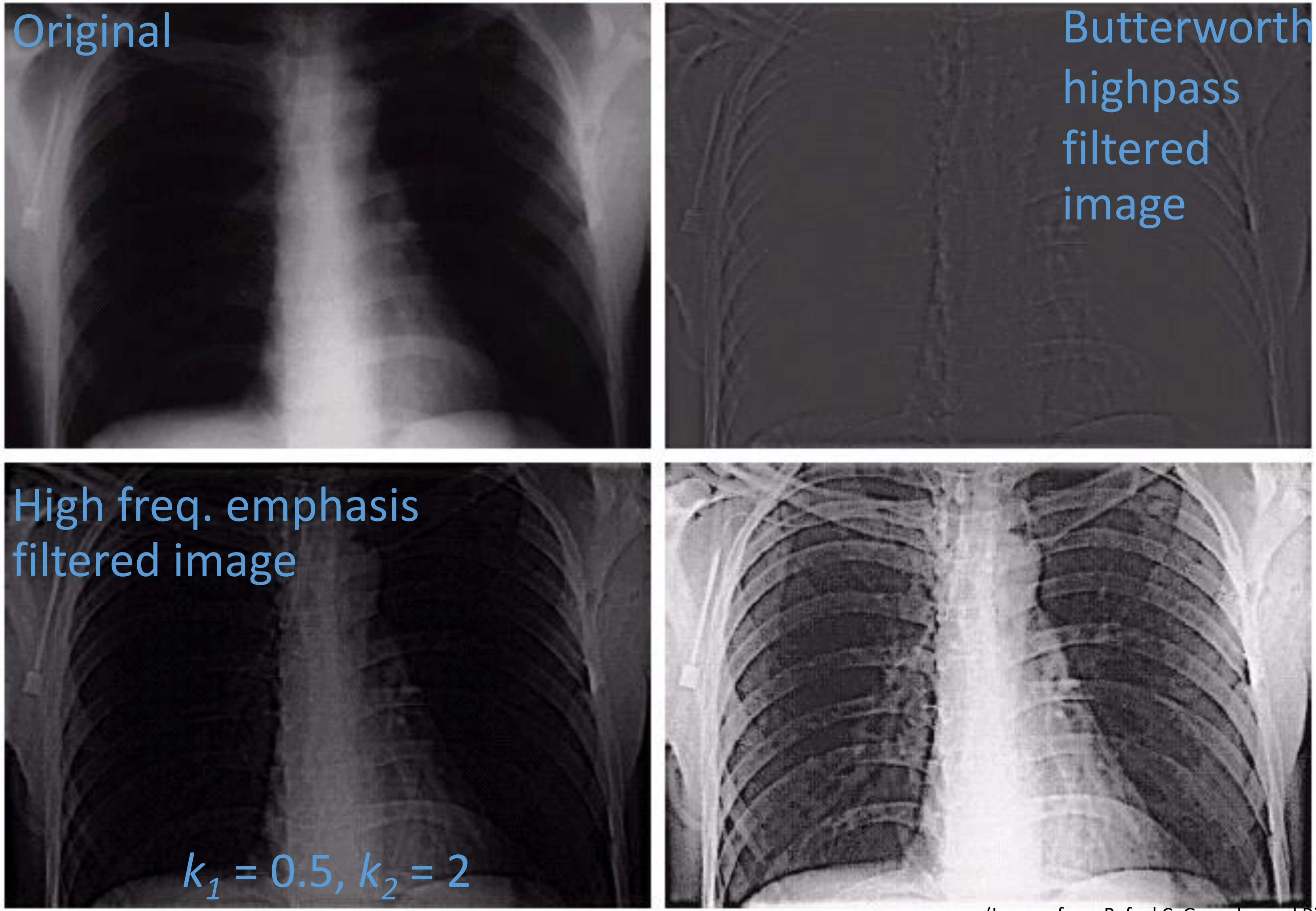
The expression contained within the square brackets is called a *high-frequency emphasis filter*

$$g(x, y) = \Im^{-1}\left\{[k_1 + k_2 * H_{\mathrm{HP}}(u, v)]F(u, v)\right\}$$

where $k_1 \geq 0$ gives controls of the offset from the origin and $k_2 \geq 0$ controls the contribution of high frequencies.

High Frequency Emphasis Filtering

$$H_{hfe} = [k_1 + k_2 * H_{HP}(u,v)]$$



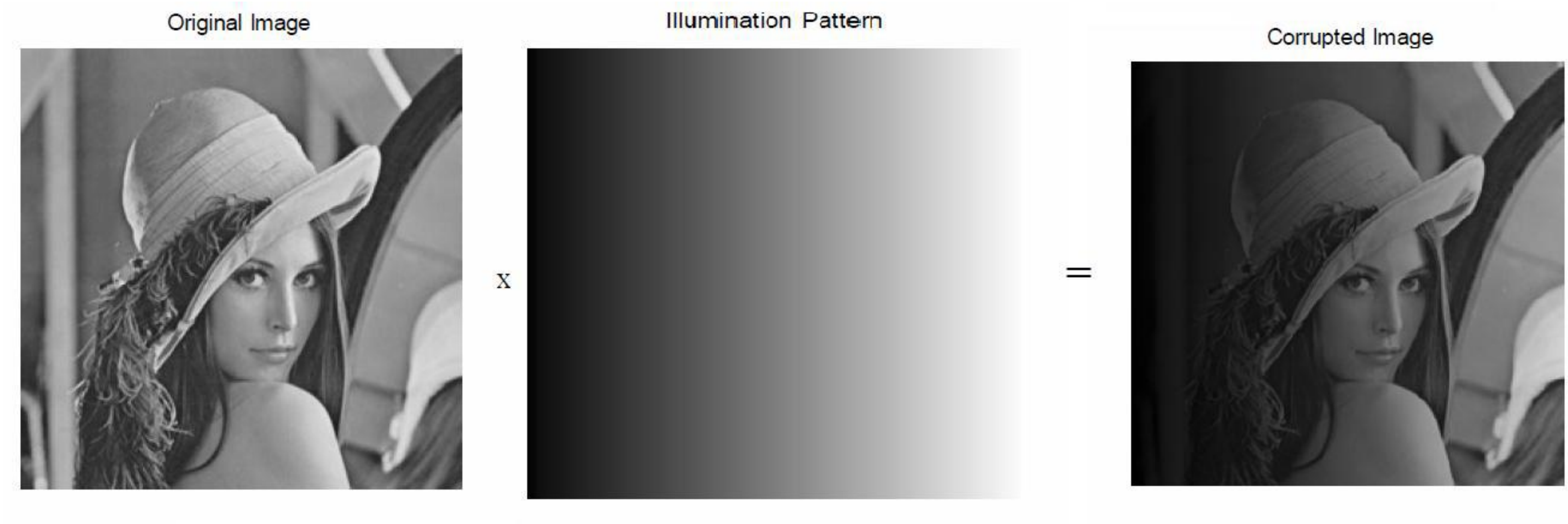
After Hist Eq.

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

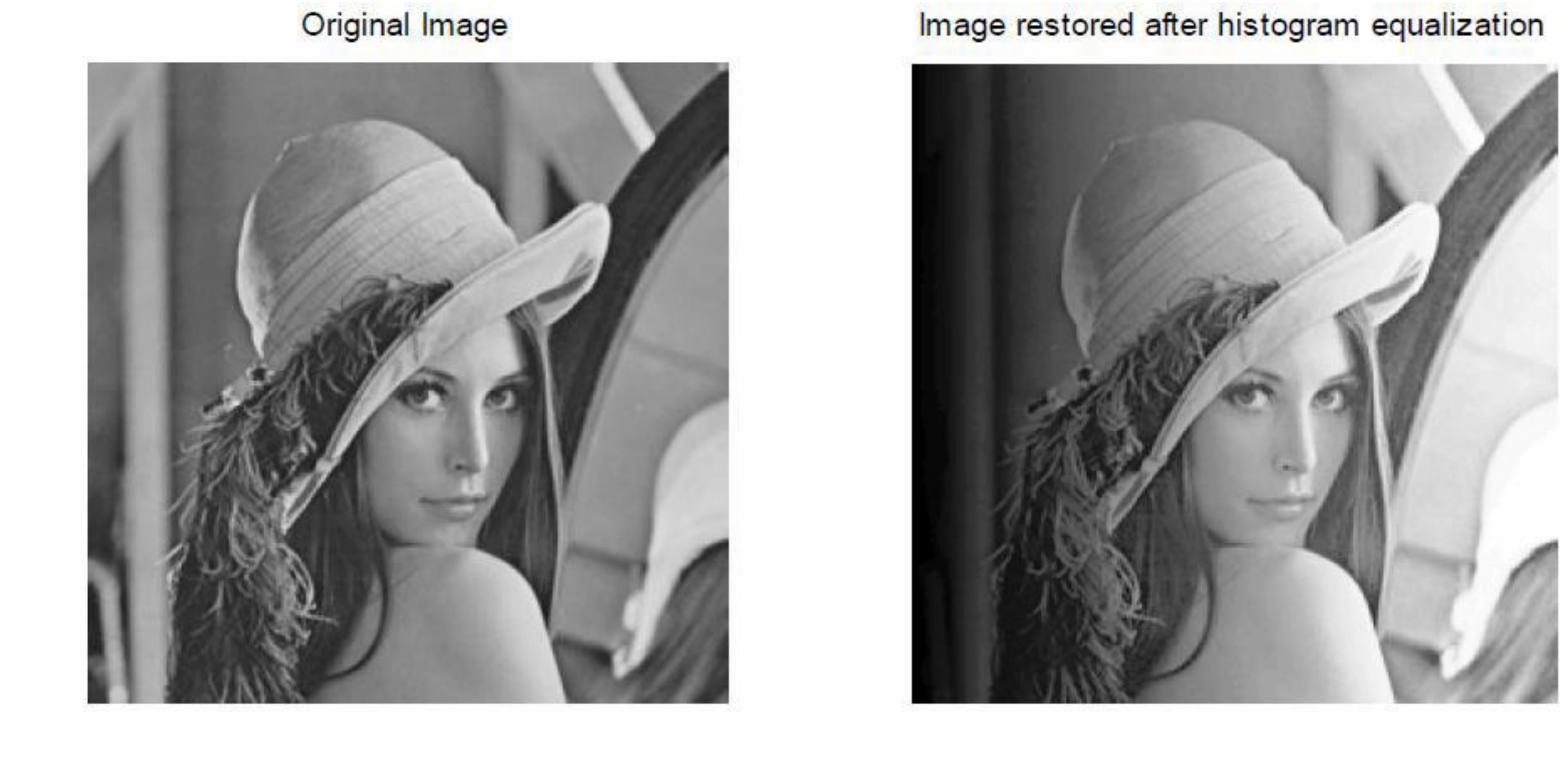
- The illumination-reflectance model (Chapter 2) can be used to develop a frequency domain procedure for improving the appearance of an image by simultaneous intensity range compression and contrast enhancement.

Original Image

Corrupted Image



In real world when images are poorly illuminated or corrupted by multiplicative noise –how to enhance?



Still non uniform illumination is present. solution?

Homomorphic Filtering

- an image $f(x, y)$ can be expressed as the product of its illumination, $i(x, y)$, and reflectance, $r(x, y)$, components:

$$f(x, y) = i(x, y)r(x, y)$$

$$\Im[f(x, y)] \neq \Im[i(x, y)]\Im[r(x, y)]$$

- However, suppose that we define

$$z(x, y) = \ln f(x, y)$$

$$= \ln i(x, y) + \ln r(x, y)$$

- then $\Im\{z(x, y)\} = \Im\{\ln f(x, y)\}$

$$= \Im\{\ln i(x, y)\} + \Im\{\ln r(x, y)\}$$

$$Z(u, v) = F_i(u, v) + F_r(u, v)$$

- where $F_i\ (u, v)$ and $F_r\ (u, v)$ are the Fourier transforms of $ln\ i\ (x, y)$ and $ln\ r\ (x, y)$ respectively.
- We can filter $Z\ (u, v\ )$ using a filter $H\ (u, v\ )$ so that

$$\begin{aligned} S(u, v) &= H(u, v)Z(u, v) \\ &= H(u, v)F_i(u, v) + H(u, v)F_r(u, v) \end{aligned}$$

- The filtered image in the spatial domain is

$$\begin{aligned} s(x, y) &= \Im^{-1}\left\{S(u, v)\right\} \\ &= \Im^{-1}\left\{H(u, v)F_i(u, v)\right\} + \Im^{-1}\left\{H(u, v)F_r(u, v)\right\} \end{aligned}$$

- By defining

$$i'(x, y) = \Im^{-1}\{H(u, v)F_i(u, v)\} \qquad r'(x, y) = \Im^{-1}\{H(u, v)F_r(u, v)\}$$

- We can express equation as

$$s(x, y) = i'(x, y) + r'(x, y)$$

- Finally, because $z(x, y)$ was formed by taking the natural logarithm of the input image, we reverse the process by taking the exponential of the filtered result to form the output image:

$$\begin{aligned} g(x, y) &= e^{s(x,y)} \\ &= e^{i'(x,y)}e^{r'(x,y)} \\ &= i_0(x, y)\,r_0(x, y) \end{aligned}$$

$$g(x, y) = i_0(x, y)\, r_0(x, y)$$

- Where

$$i_0(x, y) = e^{i'(x,y)} \qquad\qquad r_0(x, y) = e^{r'(x,y)}$$

- are the illumination and reflectance components of the output (processed) image.

Homomorphic Filtering

An image can be expressed as

$$f(x,y) = i(x,y)r(x,y)$$

$i(x,y)$ = illumination component
$r(x,y)$ = reflectance component

We need to suppress effect of illumination that cause image Intensity to change slowly.

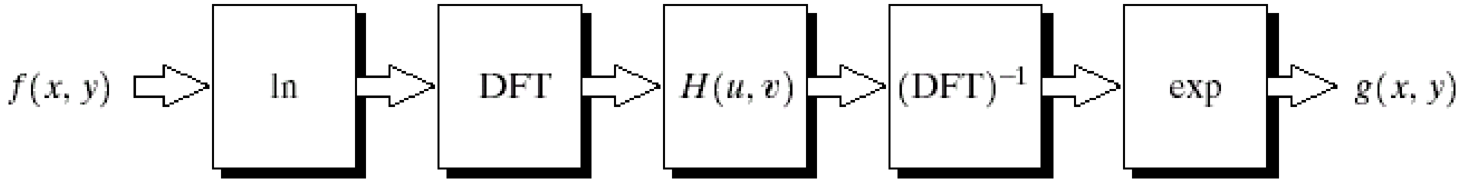
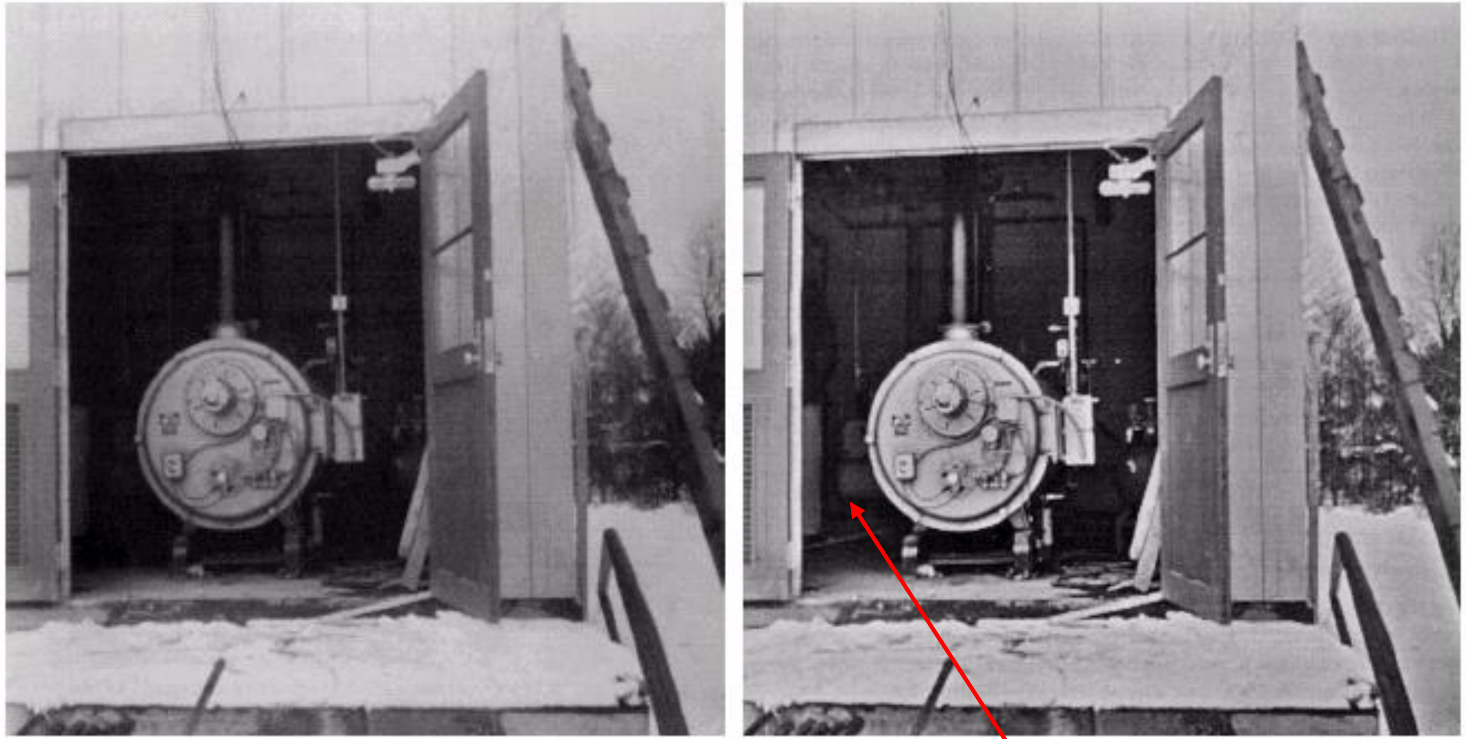


FIGURE 4.31 Homomorphic filtering approach for image enhancement.

Homomorphic Filtering

a b

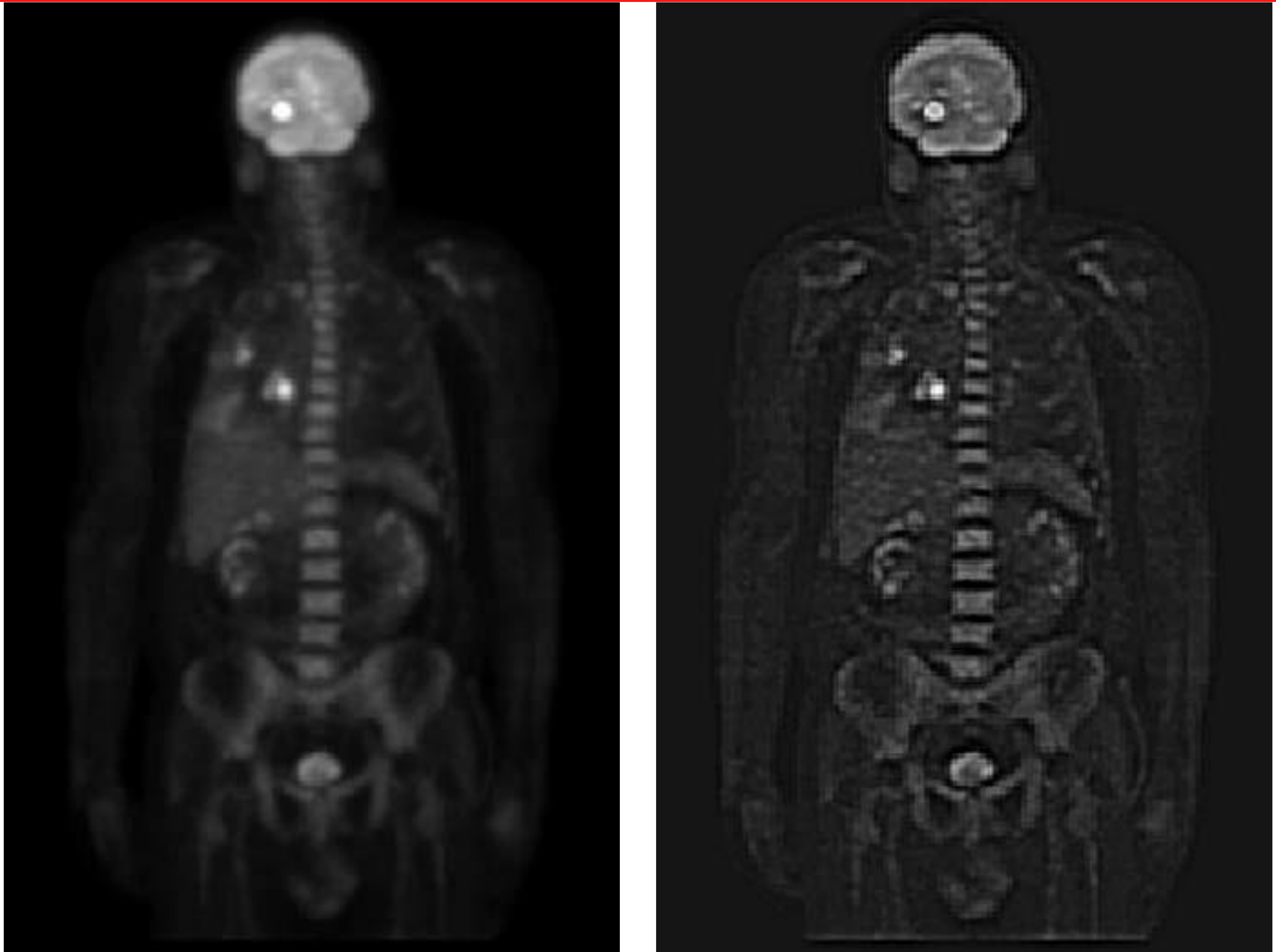
FIGURE 4.33 (a) Original image. (b) Image processed by homomorphic filtering (note details inside shelter). (Stockham.)



More details in the room can be seen!

(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Homomorphic Filtering



a b

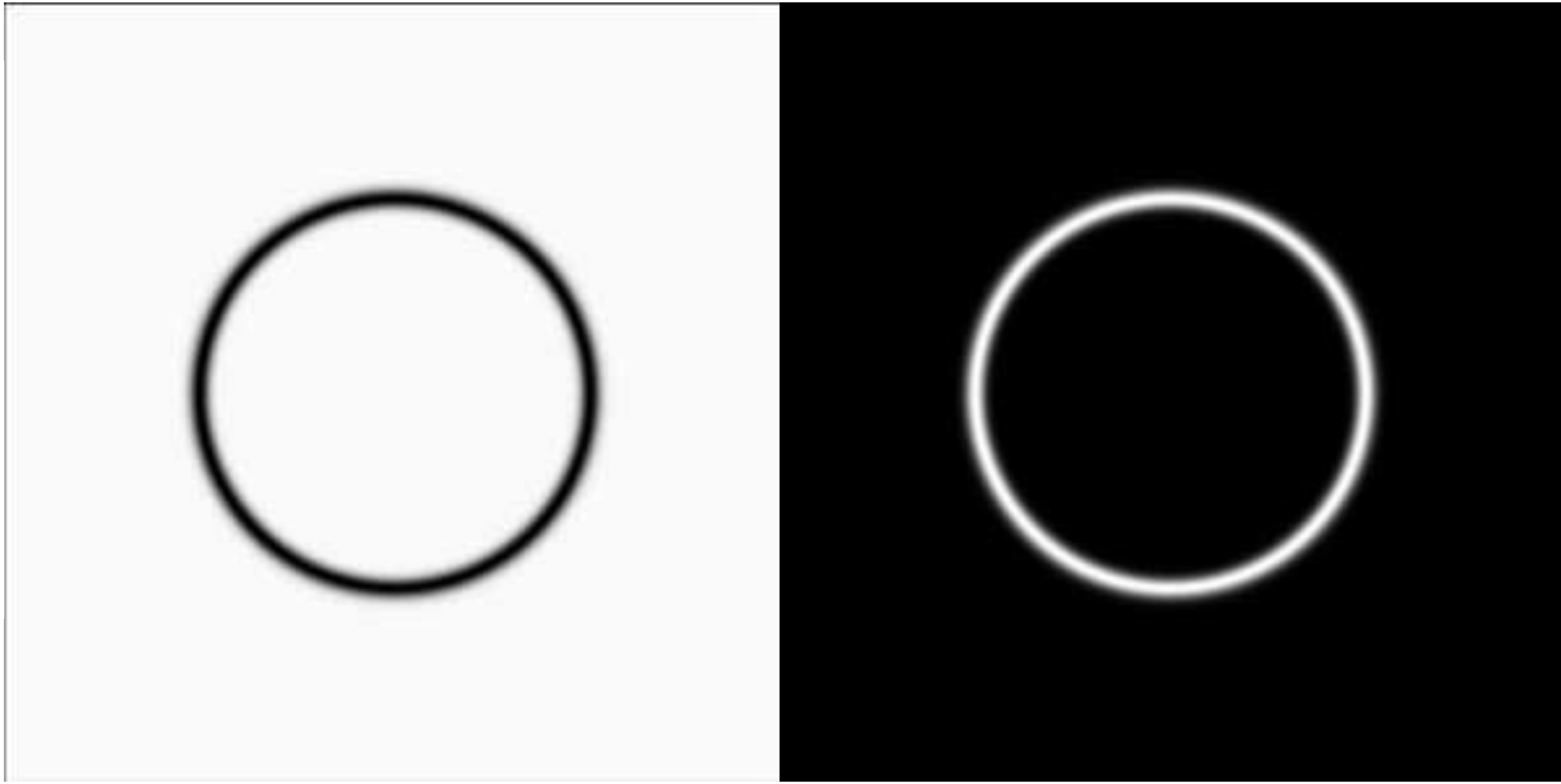
FIGURE 4.62
(a) Full body PET scan. (b) Image enhanced using homomorphic filtering. (Original image courtesy of Dr. Michael E. Casey, CTI PET Systems.)

Selective Filtering (Band pass and Band reject Filters)

TABLE 4.6
Bandreject filters. W is the width of the band, D is the distance $D(u, v)$ from the center of the filter, D_0 is the cutoff frequency, and n is the order of the Butterworth filter. We show D instead of $D(u, v)$ to simplify the notation in the table.

Ideal	Butterworth	Gaussian
$H(u,v) = \begin{cases} 0 & \text{if } D_0 - \frac{W}{2} \leq D \leq D_0 + \frac{W}{2} \\ 1 & \text{otherwise} \end{cases}$	$H(u,v) = \dfrac{1}{1 + \left[\dfrac{DW}{D^2 - D_0^2}\right]^{2n}}$	$H(u,v) = 1 - e^{-\left[\frac{D^2 - D_0^2}{DW}\right]^2}$

$$H_{\mathrm{BP}}(u,v) = 1 - H_{\mathrm{BR}}(u,v)$$



a b

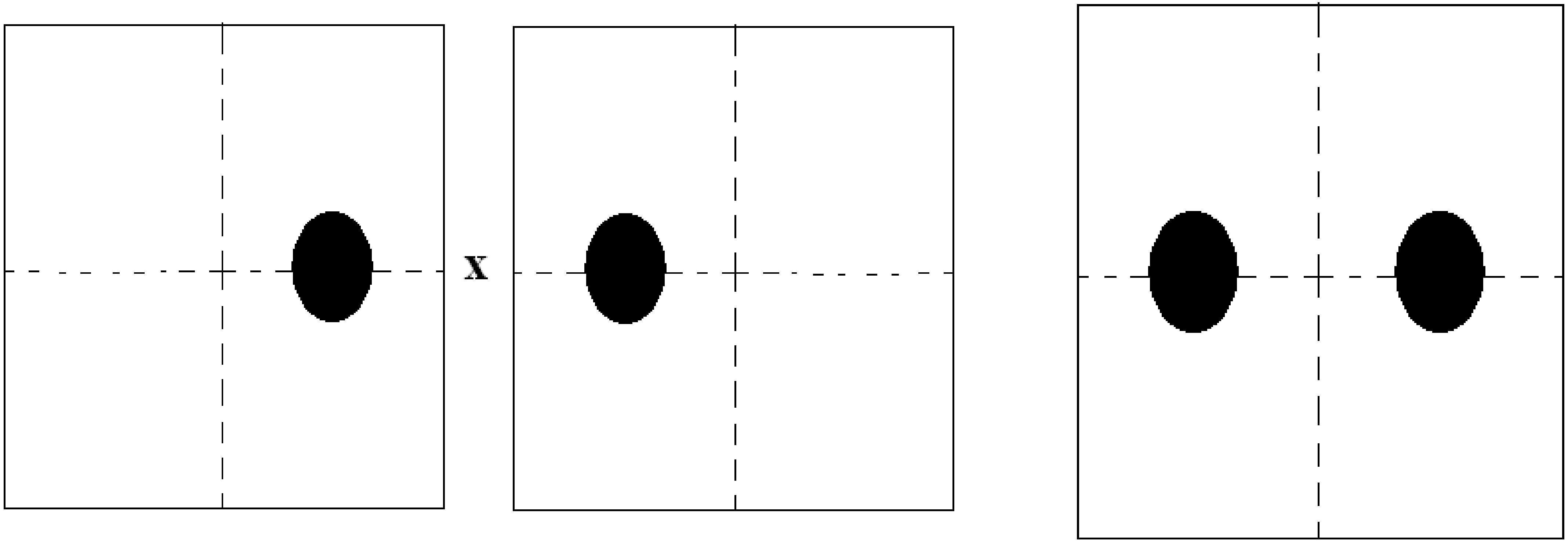
FIGURE 4.63
(a) Bandreject Gaussian filter. (b) Corresponding bandpass filter. The thin black border in (a) was added for clarity; it is not part of the data.

Notch Filter

- A notch filter rejects (or passes) frequencies in a predefined neighborhood about the center of the frequency rectangle.
- Zero-phase-shift filters must be symmetric about the origin, so a notch with center at (u_0, v_0) must have a corresponding notch at location $(-u_0, -v_0)$
- Notch reject filters are constructed as products of highpass filters whose centers have been translated to the centers of the notches.
- The general form is:

$$H_{\mathrm{NR}}(u, v) = \prod_{k=1}^{Q} H_k(u, v) H_{-k}(u, v)$$

where $H_k(u, v)$ and $H_{-k}(u, v)$ are highpass filters whose centers are at (u_k, v_k) and $(-u_k, -v_k)$, respectively. These centers are specified with respect to the



$$H_{\mathrm{NR}}(u, v) = \prod_{k=1}^{Q} H_k(u, v) H_{-k}(u, v)$$

- For example, the following is a Butterworth notch reject filter of order n, containing three notch pairs:

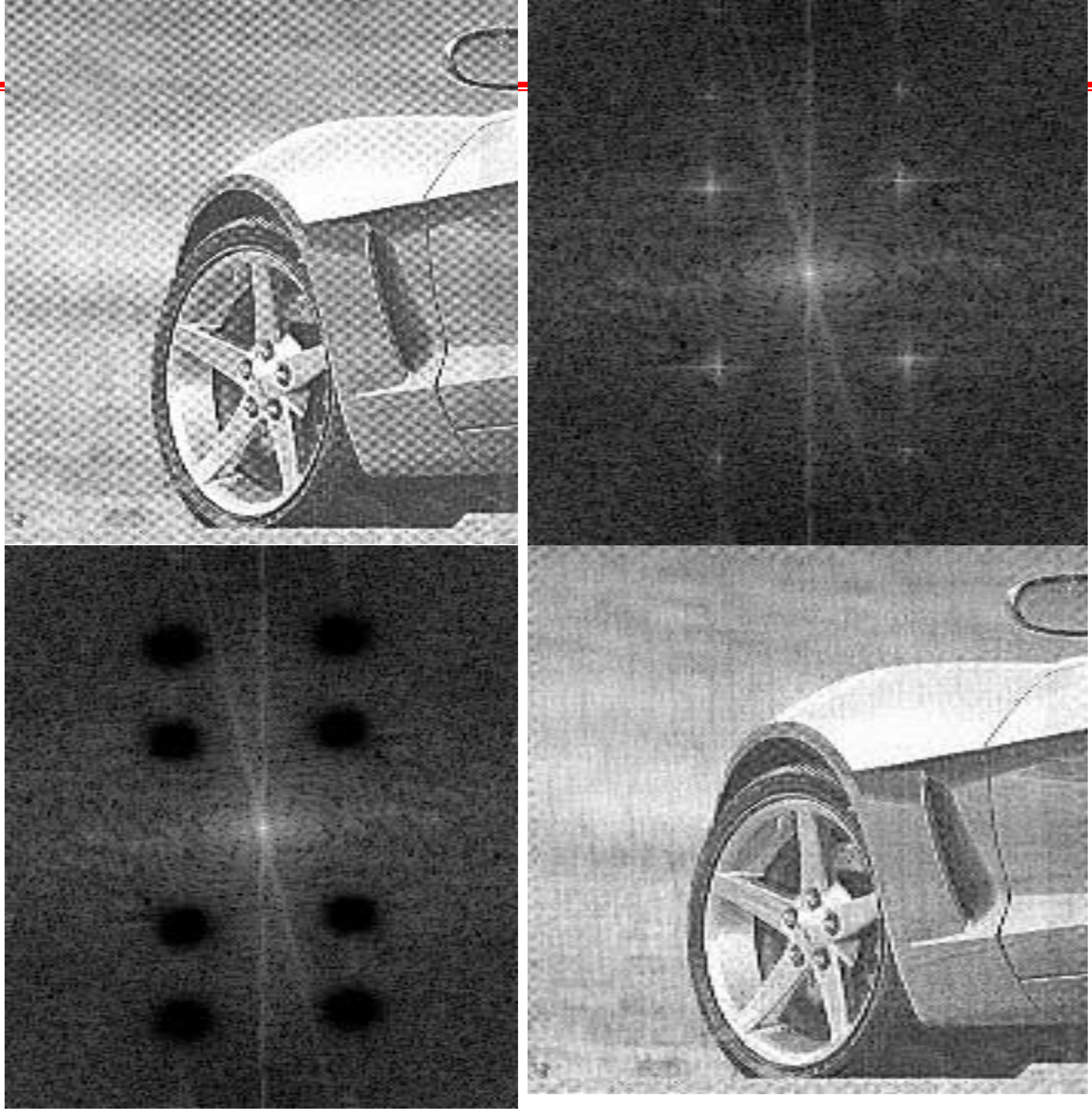
$$H_{\mathrm{NR}}(u, v) = \prod_{k=1}^{3} \left[\frac{1}{1 + [D_{0k}/D_k(u, v)]^{2n}} \right] \left[\frac{1}{1 + [D_{0k}/D_{-k}(u, v)]^{2n}} \right]$$

- The distance computations for each filter are thus carried out using the expressions

$$D_k(u, v) = \left[(u - M/2 - u_k)^2 + (v - N/2 - v_k)^2 \right]^{1/2}$$

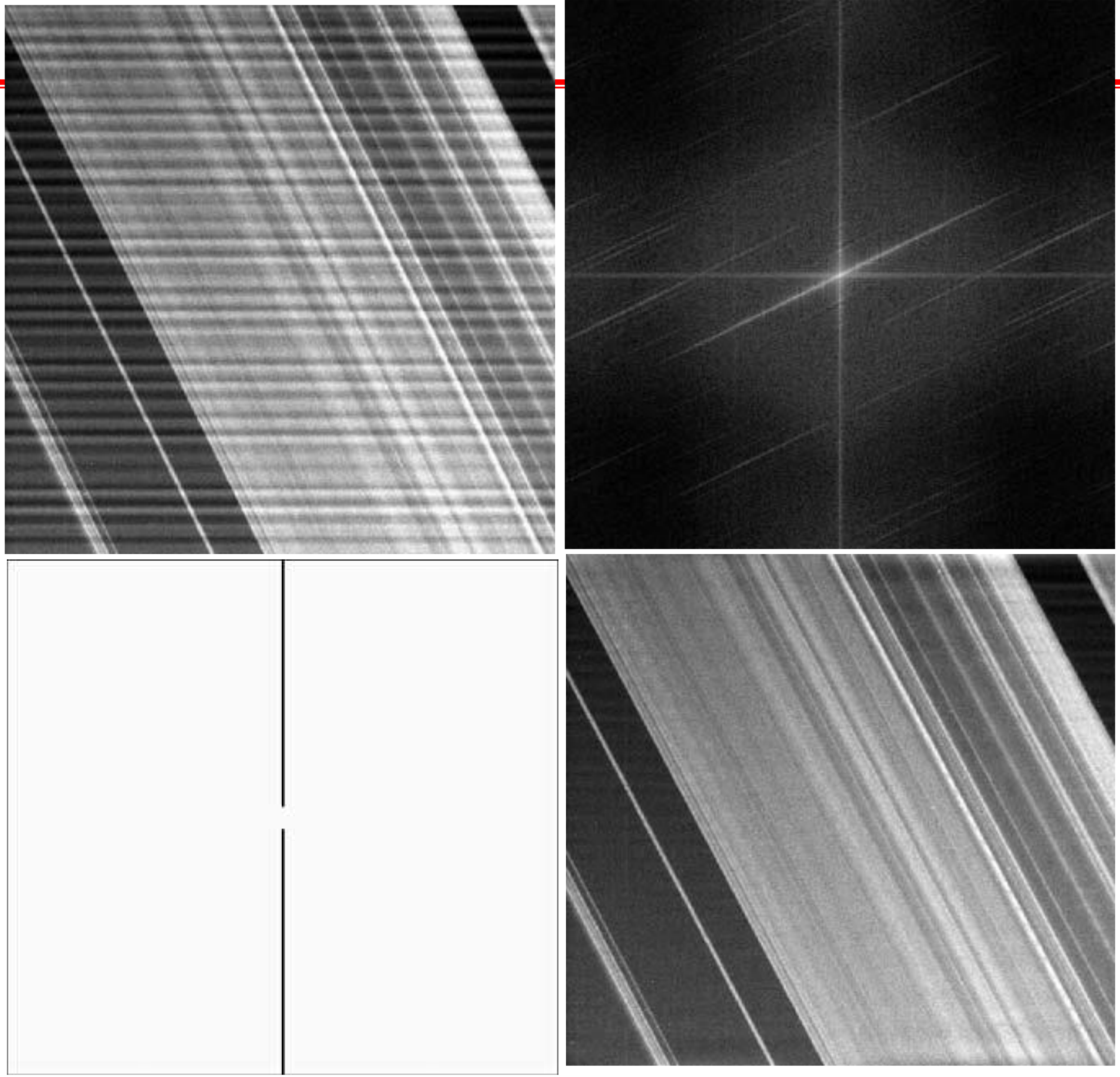
$$D_{-k}(u, v) = \left[(u - M/2 + u_k)^2 + (v - N/2 + v_k)^2 \right]^{1/2}$$

$$H_{\mathrm{NP}}(u, v) = 1 - H_{\mathrm{NR}}(u, v)$$



a b
c d

FIGURE 4.64
(a) Sampled newspaper image showing a moiré pattern. (b) Spectrum. (c) Butterworth notch reject filter multiplied by the Fourier transform. (d) Filtered image.



a	b
c	d

FIGURE 4.65 (a) 674 × 674 image of the Saturn rings showing nearly periodic interference. (b) Spectrum: The bursts of energy in the vertical axis near the origin correspond to the interference pattern. (c) A vertical notch reject filter. (d) Result of filtering. The thin black border in (c) was added for clarity; it is not part of the data. (Original image courtesy of Dr. Robert A. West, NASA/JPL.)

a b

FIGURE 4.66
(a) Result (spectrum) of applying a notch pass filter to the DFT of Fig. 4.65(a). (b) Spatial pattern obtained by computing the IDFT of (a).