Lecture 6
Frequency Domain Filtering (2)

Dr. Xiran Cai

Email: caixr@shanghaitech.edu.cn

Office: 3-438 SIST

Tel: 20684431

ShanghaiTech University

上海科技大学
ShanghaiTech University

Outline

- Discrete Convolution Theorem (wraparound problem)
- Frequency domain filtering fundamentals
- Frequency domain filtering procedure
- Typical Lowpass filtering
- Typical Highpass filtering
- Other filtering (Homomorphic filtering, Bandreject/Bandpass filtering)

Discrete Convolution Theorem

❑ Convolution theorem

$$f(x,y) \star h(x,y) \Leftrightarrow F(u,v)H(u,v)$$

$$f(x,y)\, h(x,y) \Leftrightarrow \frac{1}{MN} F(u,v) \star H(u,v)$$

Basis of Frequency Domain Filtering!

So, What we can do with frequency domain filtering?

1D case:

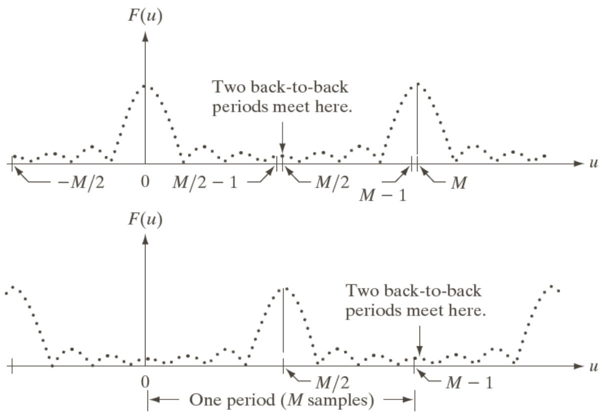
$$h[n] \otimes x[n]$$
$$= \mathcal{IDFT}\{\mathcal{DFT}\{h[n]\} \cdot \mathcal{DFT}\{x[n]\}\}$$

Is that correct?

Periodicity

- $f(x,y) = f(x+k_1M,y) = f(x,y+k_2N) = f(x+k_1M,y+k_2N)$
- $F(u,v) = F(u+k_1M,v) = F(u,v+k_2N) = F(u+k_1M,v+k_2N)$

Where k_1 and k_2 are integers



$$f(x,y) \star h(x,y) \Leftrightarrow F(u,v)H(u,v)$$

What is the truly inverse Fourier Transform of $F(u,v)$?

上海科技大学 ShanghaiTech University

Wraparound problem

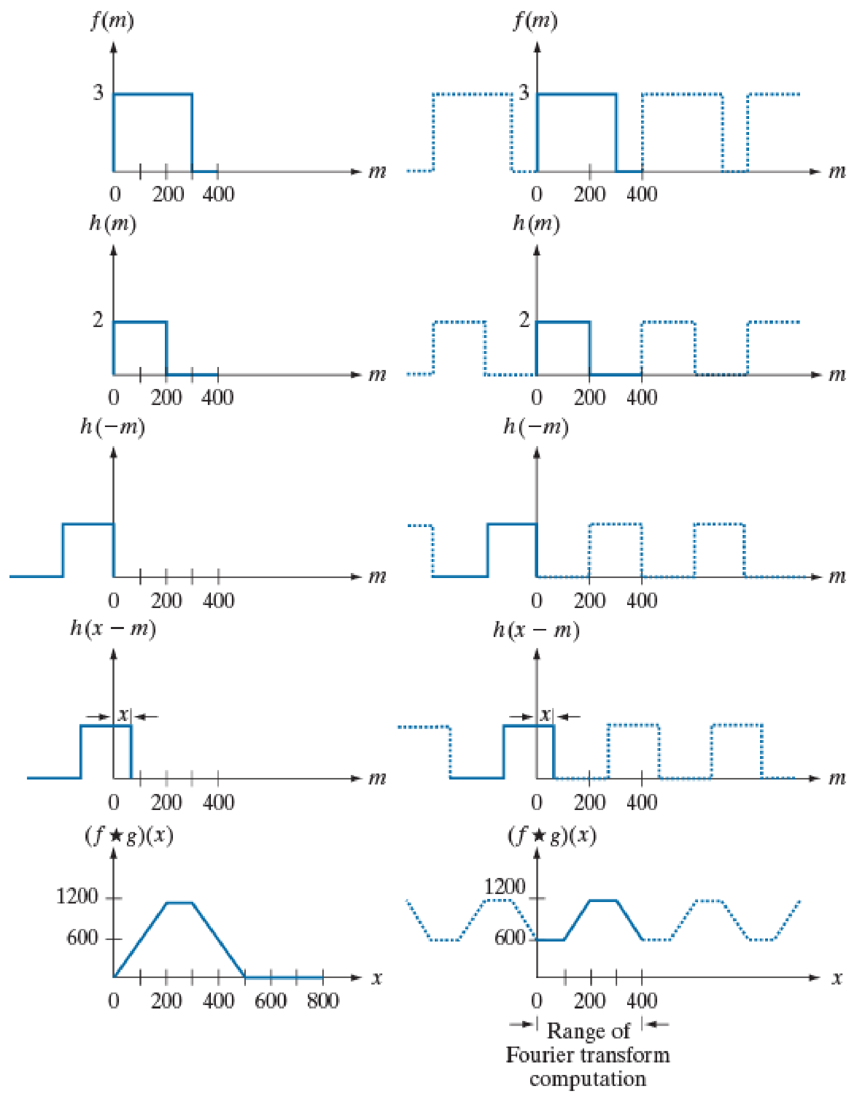


FIGURE 4.27 Left column: Spatial convolution computed with Eq. (3-44), using the approach discussed in Section 3.4. Right column: Circular convolution. The solid line in (j) is the result we would obtain using the DFT, or, equivalently, Eq. (4-48). This erroneous result can be remedied by using zero padding.

$f[n] \otimes h[n]$ $\qquad$ $f[n] \, Ⓝ \, h[n]$

One important DFT property

- Circular Convolution: Let $x_1[n]$ and $x_2[n]$ be length N with DFT $X_1[k]$ and $X_2[k]$

$$x_1[n] \circledN x_2[n] \leftrightarrow X_1[k] \cdot X_2[k]$$

  - Very useful!!! (for linear convolutions with DFT)

- Multiplication (Modulation): Let $x_1[n]$ and $x_2[n]$ be length N with DFT $X_1[k]$ and $X_2[k]$

$$x_1[n] \cdot x_2[n] \leftrightarrow \frac{1}{N} X_1[k] \circledN X_2[k]$$

Zero padding

- Zero padding is necessary for applying the convolution theorem

  - **Zero padding**

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

$$h_p(x,y)=\begin{cases} h(x,y), & 0\le x\le C-1, 0\le y\le D-1 \\ 0, & C\le x\le P, D\le y\le Q \end{cases}$$

Where $f(x,y)$**:** $A\times B$ image**;** $h(x,y)$**:** $C\times D$ image; $P\ge A+C-1$; $Q\ge B+D-1$

Linear Convolution using DFT

- In practice we can implement a circular convolution using the DFT property:

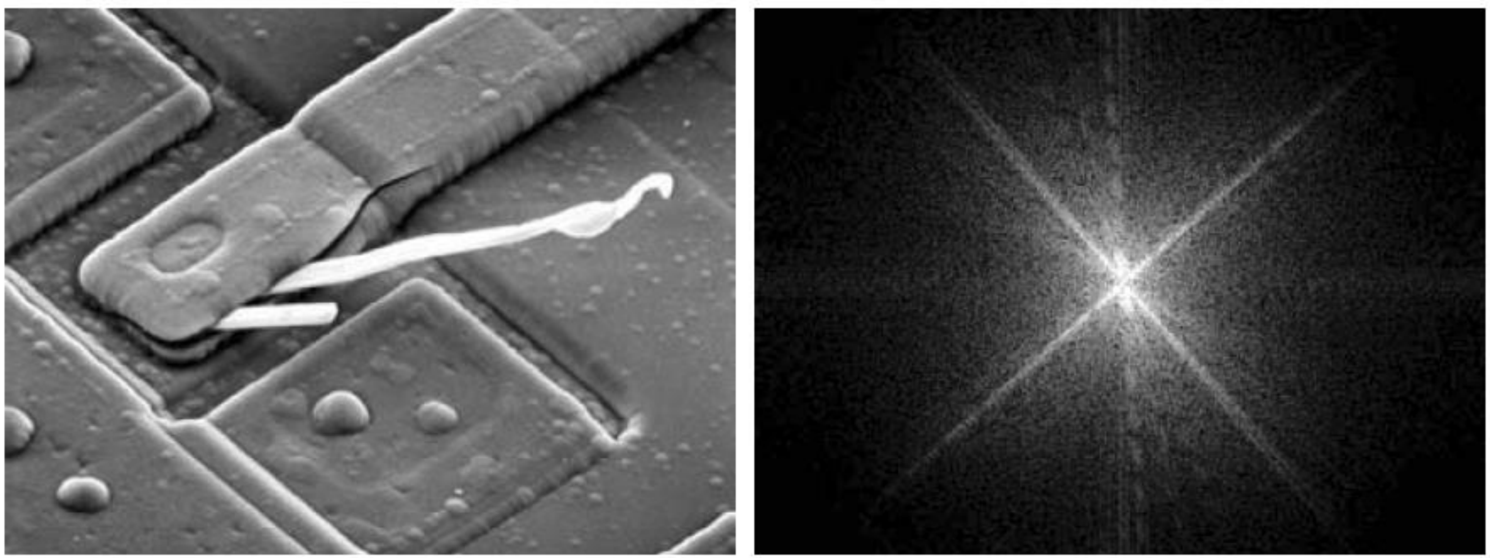
$$h[n] \circledast x[n] = x_{zp}[n] \; \textcircled{\scriptsize N} \; h_{zp}[n]$$

$$= \mathcal{IDFT}\{\mathcal{DFT}\{x_{zp}[n]\} \cdot \mathcal{DFT}\{h_{zp}[n]\}\}$$

- Advantage: DFT can be computed with $N\log_2 N$ complexity (FFT algorithm!)
- Drawback: Must wait for all the samples -- huge delay -- incompatible with real-time implementation (but **not a problem for an image!**)

Frequency domain filtering fundamentals

- How shall we read the Fourier Spectrum?
  - 45 degree?
  - White Components?



a b

FIGURE 4.29 (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.)

Simplest frequency domain filtering

- Set the DC component to 0

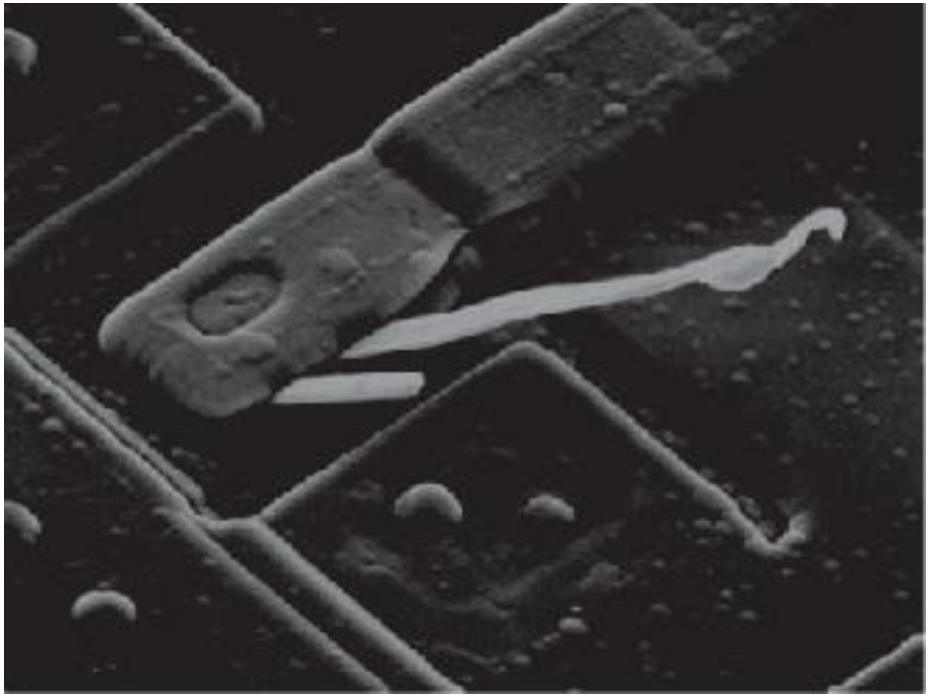
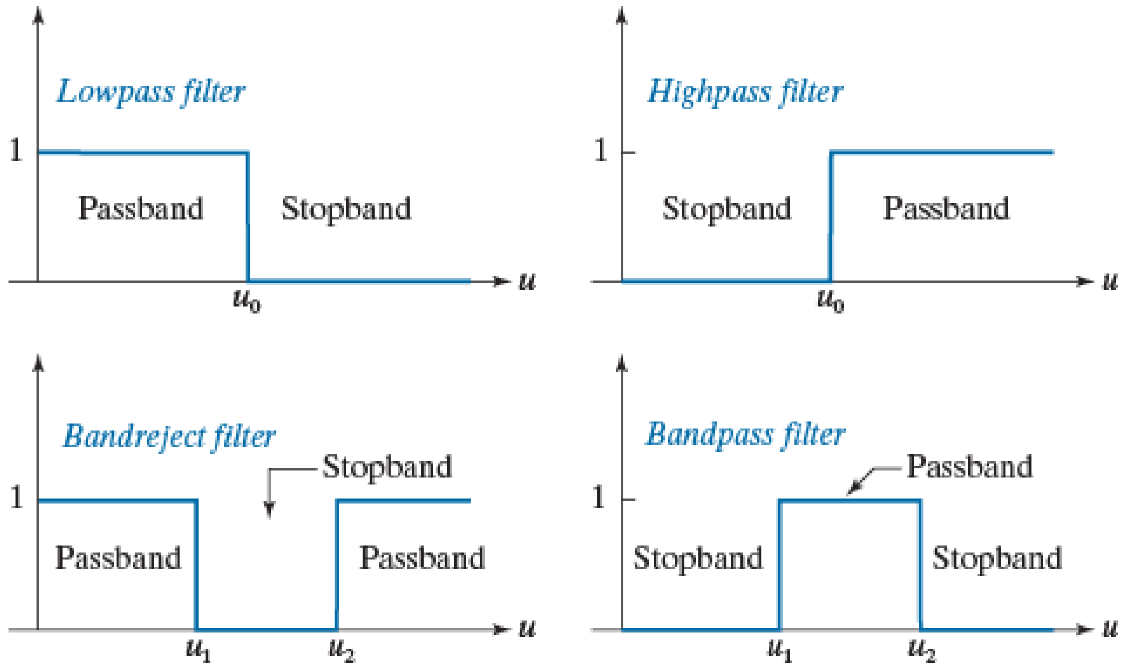


FIGURE 4.29 Result of filtering the image in Fig. 4.28(a) with a filter transfer function that sets to 0 the dc term, $F(P/2, Q/2)$, in the centered Fourier transform, while leaving all other transform terms unchanged.

Filtering in frequency domain

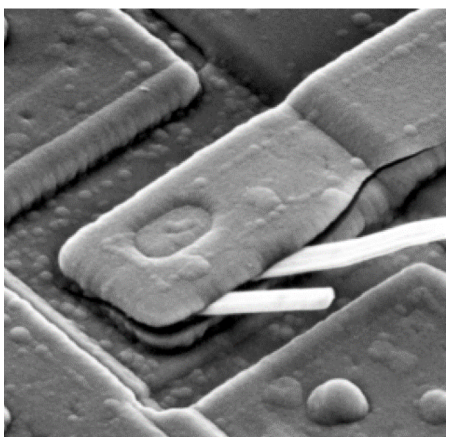
- 4 types of filters

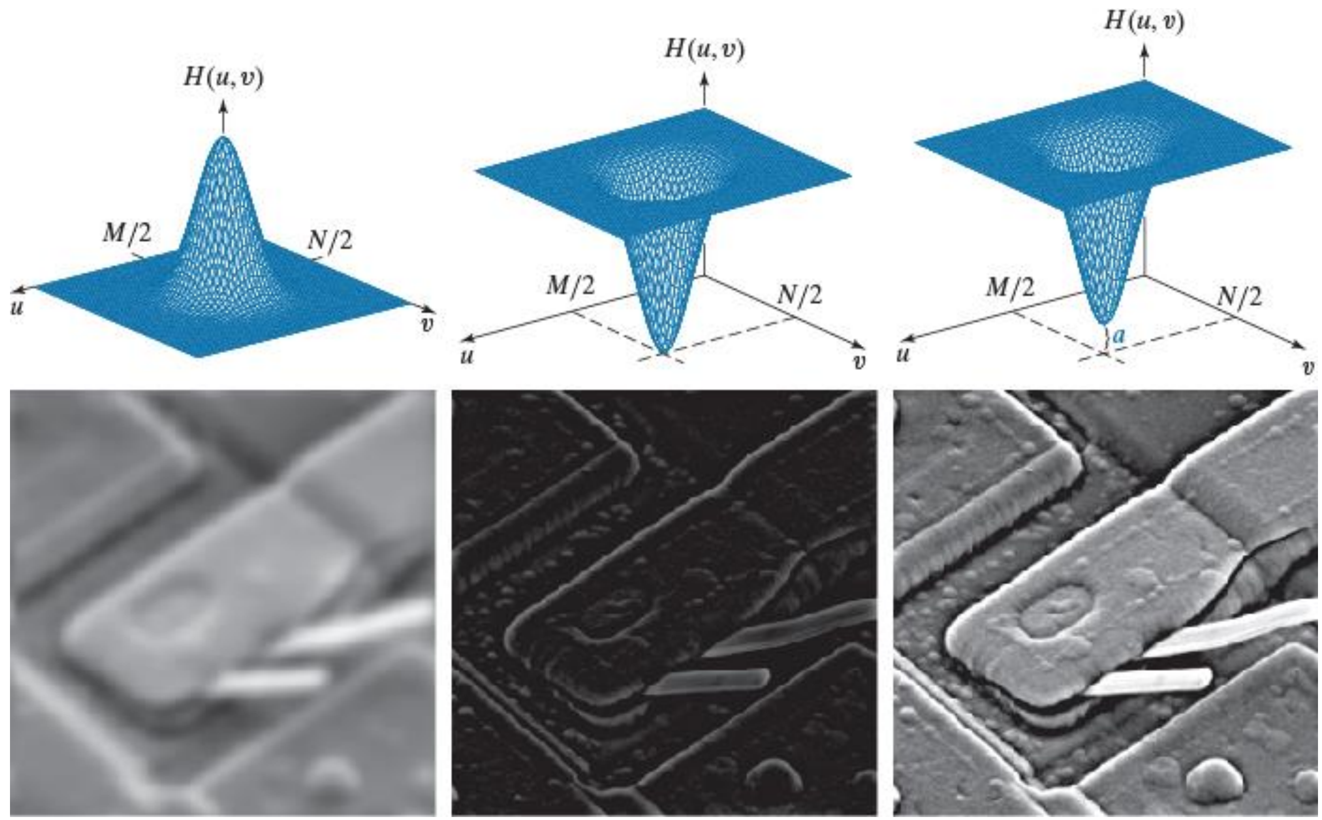


Frequency Domain Filtering

Basic Filtering form: $g(x, y) = \mathcal{F}^{-1}[H(u, v)F(u, v)]$

Original image





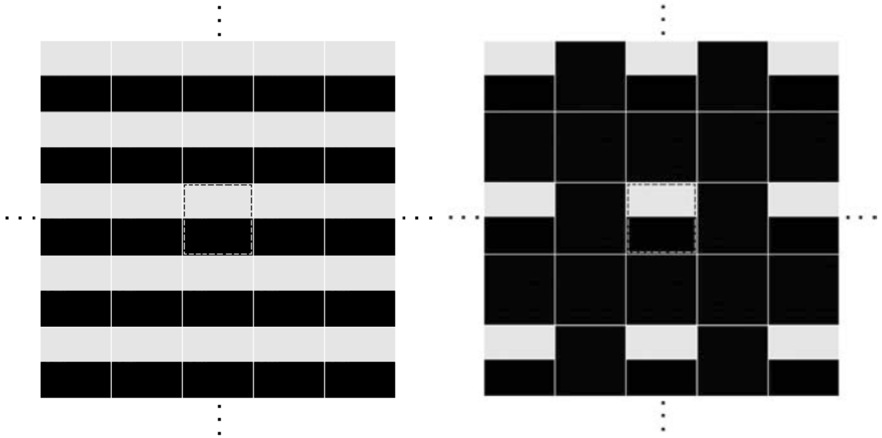
a b c
d e f

FIGURE 4.30 Top row: Frequency domain filter transfer functions of (a) a lowpass filter, (b) a highpass filter, and (c) an offset highpass filter. Bottom row: Corresponding filtered images obtained using Eq. (4-104). The offset in (c) is $a = 0.85$, and the height of $H(u,v)$ is 1. Compare (f) with Fig. 4.28(a).

Padding or not?

a b c

FIGURE 4.31 (a) A simple image. (b) Result of blurring with a Gaussian lowpass filter without padding. (c) Result of lowpass filtering with zero padding. Compare the vertical edges in (b) and (c).



a b

FIGURE 4.32 (a) Image periodicity without image padding. (b) Periodicity after padding with 0's (black). The dashed areas in the center correspond to the image in Fig. 4.31(a). Periodicity is inherent when using the DFT. (The thin white lines in both images are superimposed for clarity; they are not part of the data.)

Steps for frequency domain filtering

1. Given an input image $f(x,y)$ of size MxN, obtain the padding parameters P and Q. Typically, $P = 2M$ and $Q = 2N$.

2. Form a padded image, $f_p(x,y)$ of size PxQ by appending the necessary number of zeros to $f(x,y)$

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

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

5. Generate a real, symmetric filter function*, $H(u,v)$, of size PxQ with center at coordinates $(P/2, Q/2)$

*generate from a given spatial filter, we pad the spatial filter, multiply the expadded array by $(-1)^{x+y}$, and compute the DFT of the result to obtain a centered $H(u,v)$.

Steps for frequency domain filtering

6. Form the product $G(u,v) = H(u,v)F(u,v)$ using array multiplication

7. Obtain the processed image

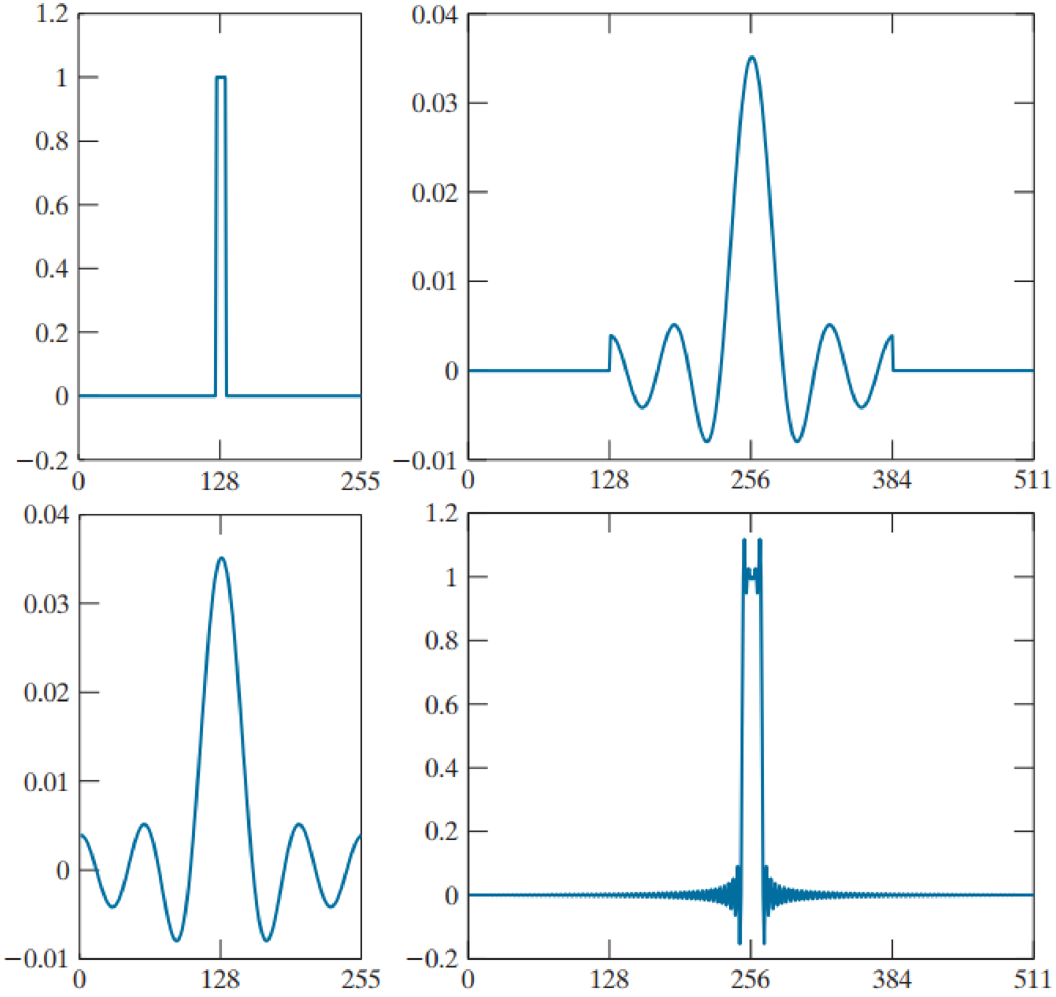
$$g_p(x,y) = \left\{ real\left[\Im^{-1}\left[G(u,v) \right] \right] \right\} (-1)^{x+y}$$

8. Obtain the final processed result, $g(x,y)$, by extracting the MxN region from the top, left quadrant of $g_p(x,y)$

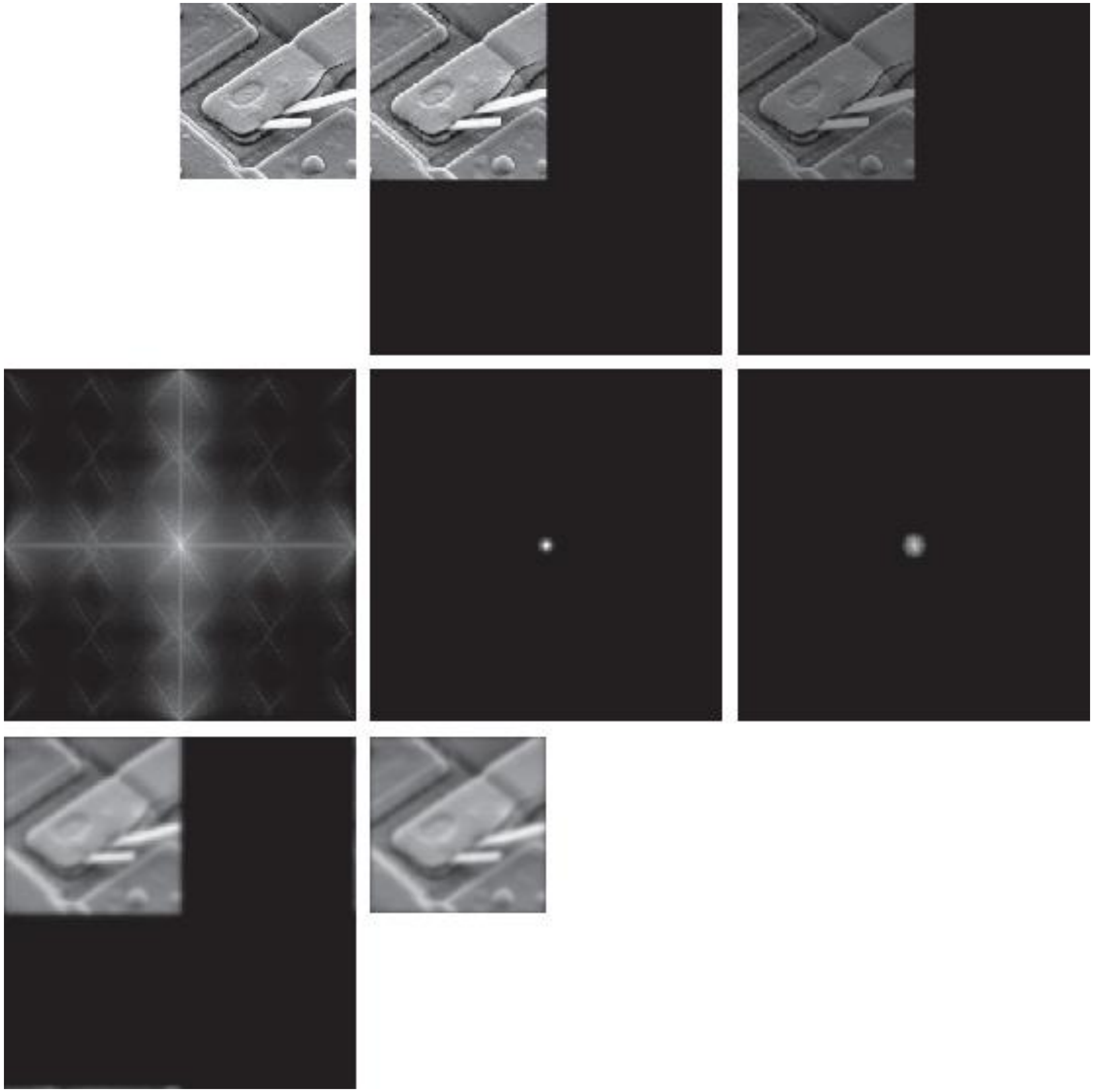
Note

a c
b d

FIGURE 4.33
(a) Filter transfer function specified in the (centered) frequency domain. (b) Spatial representation (filter kernel) obtained by computing the IDFT of (a). (c) Result of padding (b) to twice its length (note the discontinuities). (d) Corresponding filter in the frequency domain obtained by computing the DFT of (c). Note the ringing caused by the discontinuities in (c). Part (b) of the figure is below (a), and (d) is below (c).



Steps for frequency domain filtering



a b c
d e f
g h

FIGURE 4.35
(a) An $M \times N$ image, f.
(b) Padded image, f_p of size $P \times Q$.
(c) Result of multiplying f_p by $(-1)^{x+y}$.
(d) Spectrum of F. (e) Centered Gaussian lowpass filter transfer function, H, of size $P \times Q$.
(f) Spectrum of the product HF.
(g) Image g_p, the real part of the IDFT of HF, multiplied by $(-1)^{x+y}$.
(h) Final result, g, obtained by extracting the first M rows and N columns of g_p.

Filtering in Spatial and Frequency Domains

❑ Frequency filters ⇒ Spatial filter $H(u,v) \Rightarrow h(x,y)$

Design filter in frequency domain → Compute IDFT → Full size spatial filter → Smaller spatial masks

❑ Gaussian Filter

$$H(u) = Ae^{-\frac{u^2}{2\sigma^2}} \Leftrightarrow h(x) = \sqrt{2\pi}\sigma A e^{-2\pi^2\sigma^2x^2}$$

$$H(u) = Ae^{-\frac{u^2}{2\sigma_1^2}} - Be^{-\frac{u^2}{2\sigma_2^2}} \Leftrightarrow h(x) = \sqrt{2\pi}\sigma_1 A e^{-2\pi^2\sigma_1^2x^2} - \sqrt{2\pi}\sigma_2 B e^{-2\pi^2\sigma_2^2x^2}$$

$$H(u,v) = Ae^{-\frac{u^2+v^2}{2\sigma^2}} \Leftrightarrow h(x,y) = A2\pi\sigma^2 e^{-2\pi^2\sigma^2(x^2+y^2)}$$

Filtering in Spatial and Frequency Domains

- **Frequency filters ⇒ Spatial filter** $\boldsymbol{H(u,v) \Rightarrow h(x,y)}$

$$H(u) = Ae^{-\frac{u^2}{2\sigma^2}} \Leftrightarrow h(x) = \sqrt{2\pi}\sigma A e^{-2\pi^2\sigma^2x^2}$$

$$H(u) = Ae^{-\frac{u^2}{2\sigma_1^2}} - Be^{-\frac{u^2}{2\sigma_2^2}} \Leftrightarrow h(x) = \sqrt{2\pi}\sigma_1 A e^{-2\pi^2\sigma_1^2x^2} - \sqrt{2\pi}\sigma_2 B e^{-2\pi^2\sigma_2^2x^2}$$

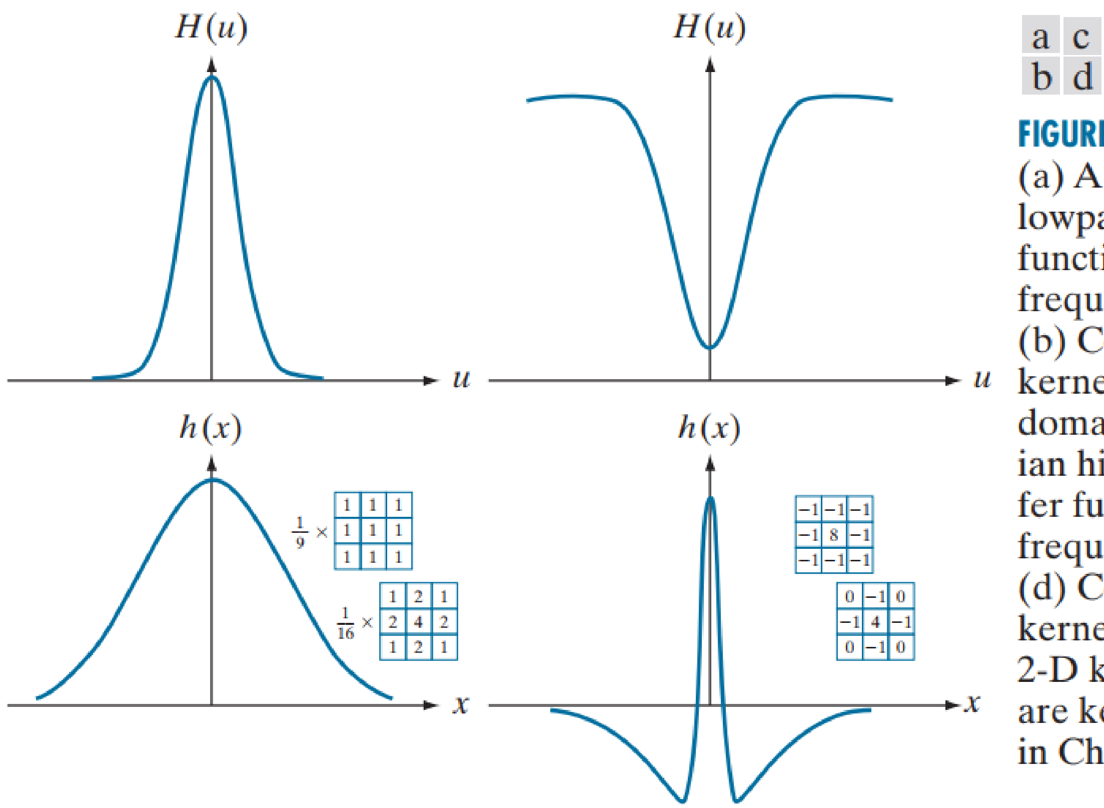


FIGURE 4.36 (a) A 1-D Gaussian lowpass transfer function in the frequency domain. (b) Corresponding kernel in the spatial domain. (c) Gaussian highpass transfer function in the frequency domain. (d) Corresponding kernel. The small 2-D kernels shown are kernels we used in Chapter 3.

Even and Odd Functions

- **Even function**

$$w_e(x,y) = w_e(M-x, N-y)$$

$$f = \{f(0), f(1), f(2), f(3)\}$$
$$= \{2,1,1,1\}$$

$$f(0) = f(4), f(1) = f(3),$$
$$f(2) = f(2), f(3) = f(1)$$

M is even, then $\{\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}, \boldsymbol{b}\}$

M is odd, then $\{\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}, \boldsymbol{c}, \boldsymbol{b}\}$

- **Odd function**

$$w_o(x,y) = -w_o(M-x, N-y)$$

$$g = \{g(0), g(1), g(2), g(3)\}$$
$$= \{0, -1, 0, 1\}$$

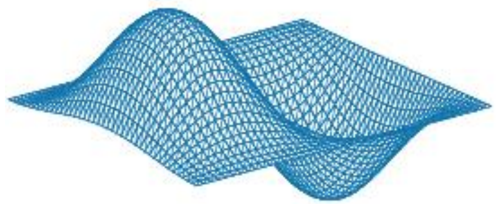
$$g(0) = -g(4) = 0, g(1) = -g(3),$$
$$g(2) = -g(2), g(3) = -g(1)$$

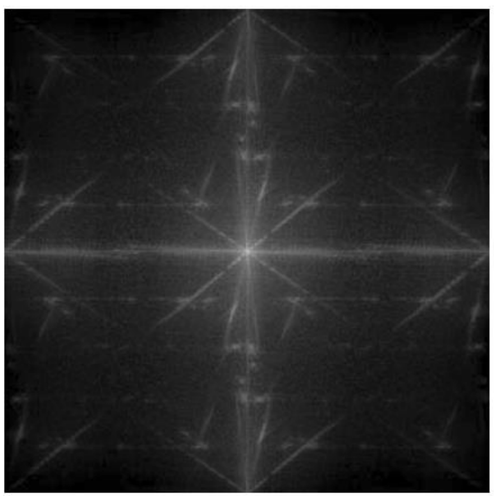
M is even, then $\{\mathbf{0}, \boldsymbol{b}, \mathbf{0}, -\boldsymbol{b}\}$

M is odd, then $\{\mathbf{0}, \boldsymbol{b}, \boldsymbol{c}, -\boldsymbol{c}, -\boldsymbol{b}\}$

Spatial and Frequency Filtering

−1	0	1
−2	0	2
−1	0	1





- **Zero Padding**

$$
\begin{array}{cccccc}
 & & & \vdots & & \\
0 & 0 & 0 & 0 & 0 & 0 \\
\cdots\cdots\; 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -1 & 0 & 1 & 0 \\
0 & 0 & -2 & \mathbf{0} & 2 & 0 \;\cdots\cdots \\
0 & 0 & -1 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 \\
 & & & \vdots & &
\end{array}
$$

- **Odd Function (奇函数)**

$$w_o(x,y) = -w_o(-x,-y)$$

$$w_o(x,y) = -w_o(M-x, N-y)$$

$f(x,y)$ real and odd $\Leftrightarrow$ $F(u,v)$ imaginary and odd

Procedure

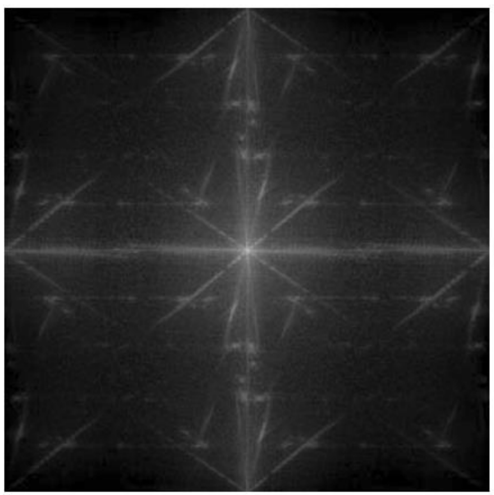
The procedure used to generate $H(u,v)$ is: (1) multiply $h_p(x,y)$ by $(-1)^{x+y}$ to center the frequency domain filter; (2) compute the forward DFT of the result in (1) to generate $H(u,v)$; (3) set the real part of $H(u,v)$ to 0 to account for parasitic real parts (we know that H has to be purely imaginary because h_p is real and odd); and (4) multiply the result by $(-1)^{u+v}$. This last step reverses the multiplication of $H(u,v)$ by $(-1)^{u+v}$, which is implicit when $h(x,y)$ was manually placed in the center of $h_p(x,y)$. Figure 4.38(a) shows a perspective plot of $H(u,v)$, and Fig. 4.38(b) shows $H(u,v)$ as an image. Note the antisymmetry in this image about its center, a result of $H(u,v)$ being odd. Function $H(u,v)$ is used as any other frequency domain filter transfer function. Figure 4.38(c) is the result of using the filter transfer function just obtained to filter the image in Fig. 4.37(a) in the frequency domain, using the step-by-step filtering procedure outlined earlier. As expected from a derivative filter, edges were enhanced and all the constant intensity areas were reduced to zero (the grayish tone is due to scaling for display). Figure 4.38(d) shows the result of filtering the same image in the spatial domain with the Sobel kernel $h(x,y)$, using the procedure discussed in Section 3.6. The results are identical.

4) Translation to center of the frequency rectangle, $(M/2, N/2)$

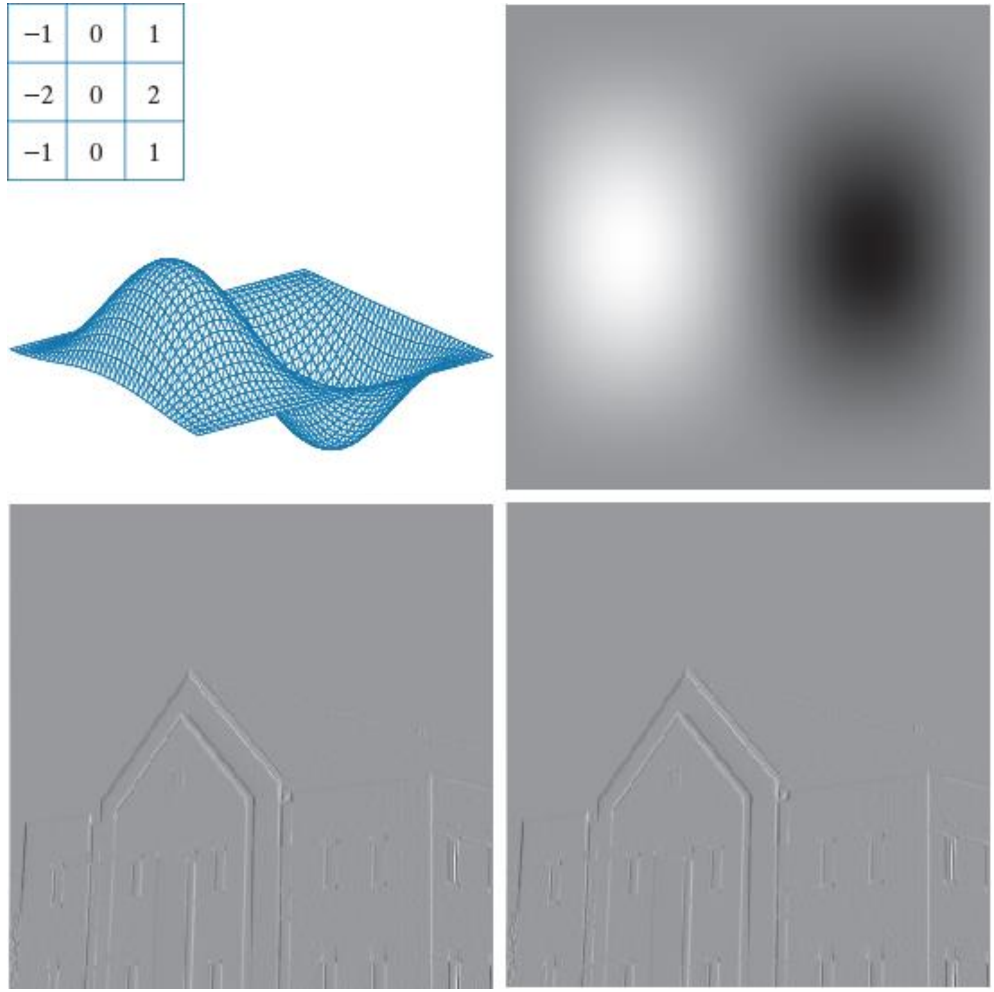
$$f(x,y)(-1)^{x+y} \Leftrightarrow F(u-M/2, v-N/2)$$

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

Spatial and Frequency Filtering



−1	0	1
−2	0	2
−1	0	1



a b
c d

FIGURE 4.38 (a) A spatial kernel and perspective plot of its corresponding frequency domain filter transfer function. (b) Transfer function shown as an image. (c) Result of filtering Fig. 4.37(a) in the frequency domain with the transfer function in (b). (d) Result of filtering the same image in the spatial domain with the kernel in (a). The results are identical.

Lowpass filtering

- Ideal Lowpass filter
- Butterworth Lowpass filter
- Gaussian Lowpass filter

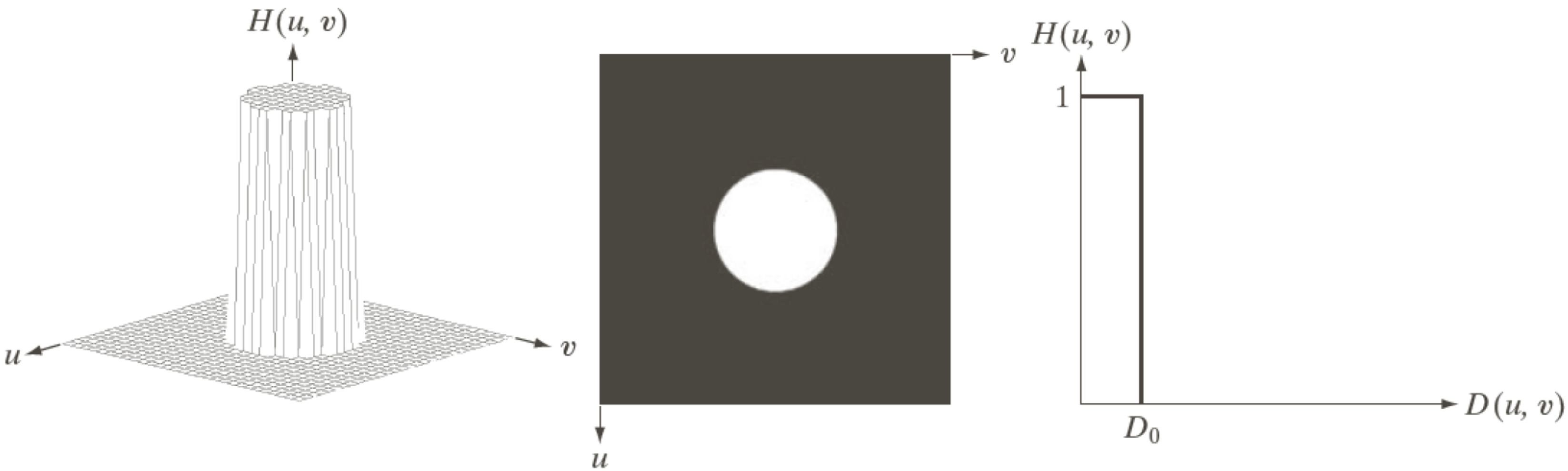
TABLE 4.5
Lowpass filter transfer functions. D_0 is the cutoff frequency, and n is the order of the Butterworth filter.

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

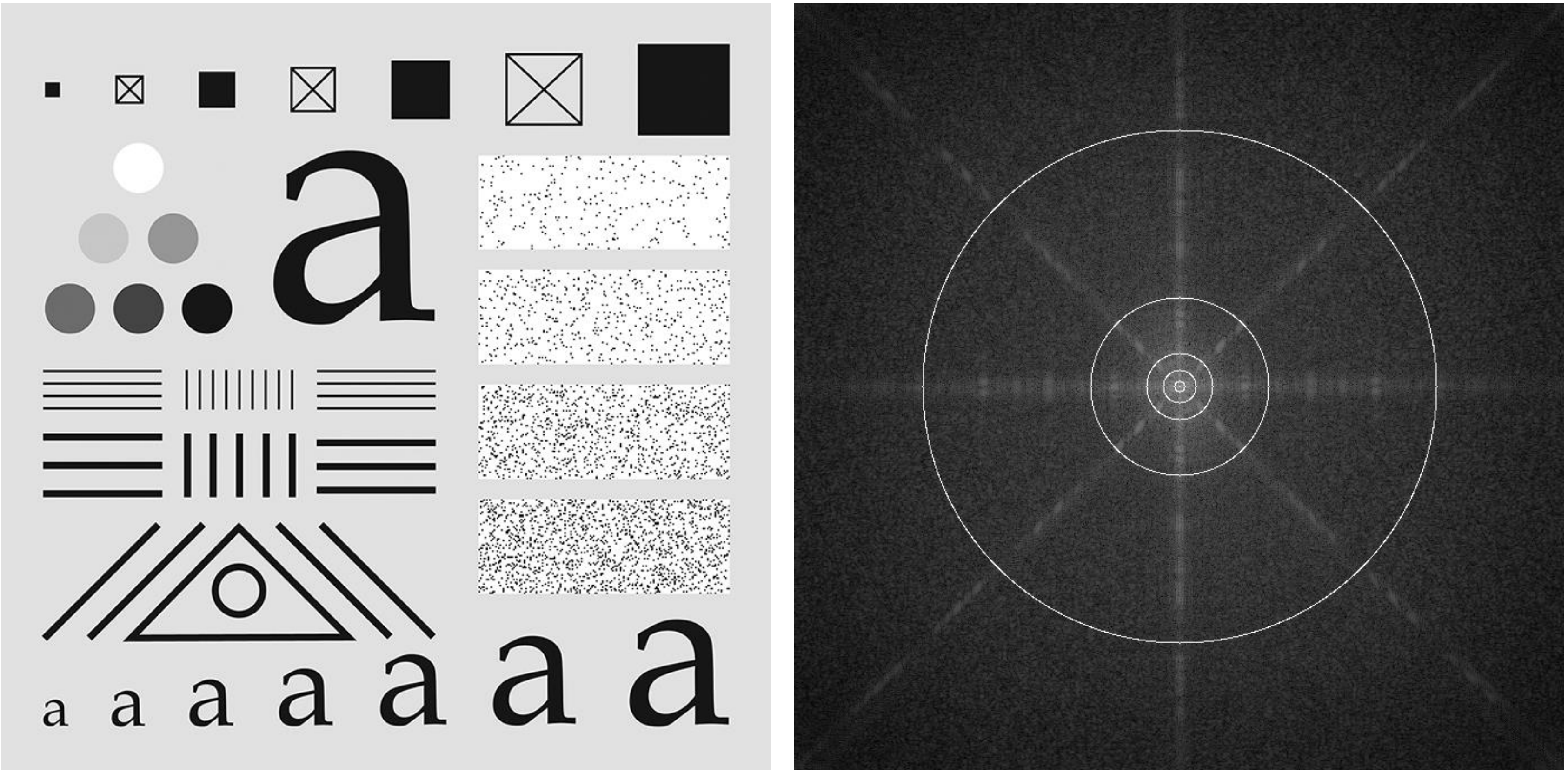
Ideal Lowpass filter

Ideal Lowpass Filter (ILPF):

$$H(u,v) = \begin{cases} 1, & D(u,v) \le D_0 \\ 0, & D(u,v) > D_0 \end{cases} \qquad D(u,v) = \left[(u - \frac{P}{2})^2 + (v - \frac{Q}{2})^2 \right]^{1/2}$$



Ideal Lowpass filter (cutoff frequency)



a b

FIGURE 4.40 (a) Test pattern of size 688 × 688 pixels, and (b) its spectrum. The spectrum is double the image size as a result of padding, but is shown half size to fit. The circles have radii of 10, 30, 60, 160, and 460 pixels with respect to the full-size spectrum. The radii enclose 86.9, 92.8, 95.1, 97.6, and 99.4% of the padded image power, respectively.

上海科技大学
ShanghaiTech University

Ideal Lowpass filter (cutoff frequency)

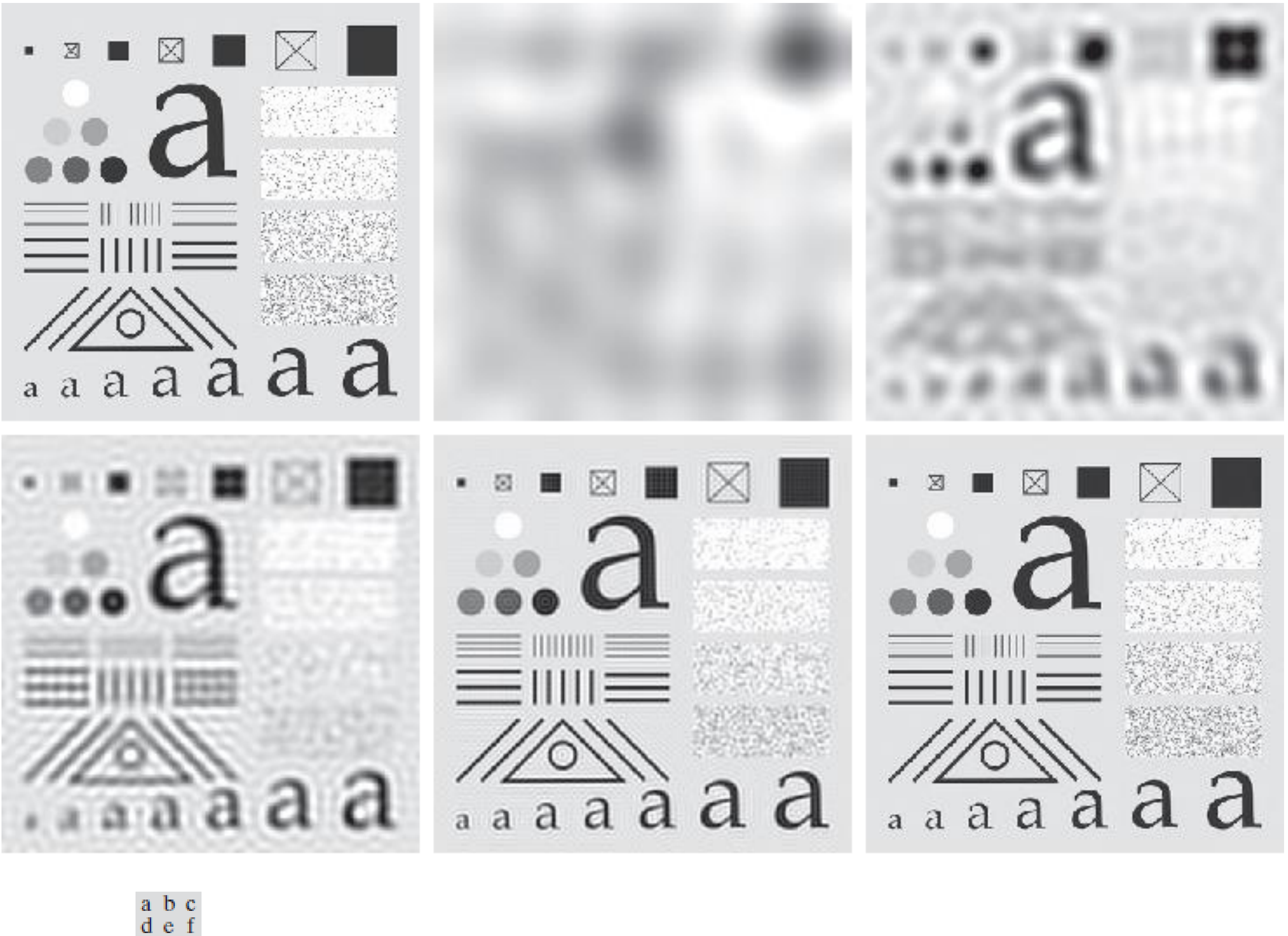
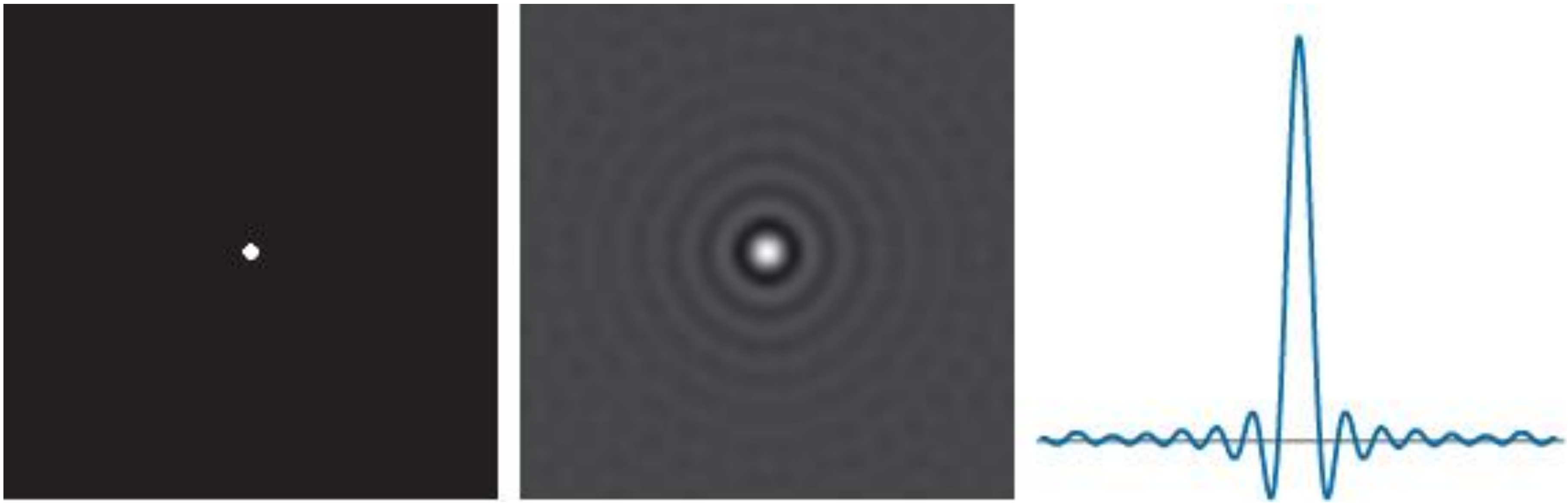


FIGURE 4.41 (a) Original image of size 688 × 688 pixels. (b)–(f) Results of filtering using ILPFs with cutoff frequencies set at radii values 10, 30, 60, 160, and 460, as shown in Fig. 4.40(b). The power removed by these filters was 13.1, 7.2, 4.9, 2.4, and 0.6% of the total, respectively. We used mirror padding to avoid the black borders characteristic of zero padding, as illustrated in Fig. 4.31(c).

Ideal Lowpass filter (cutoff frequency)



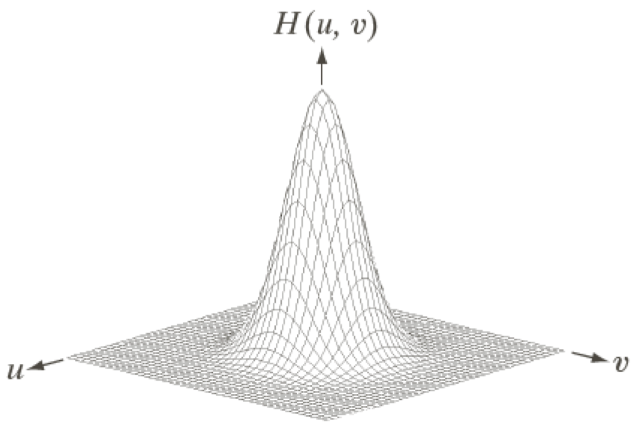
a b c

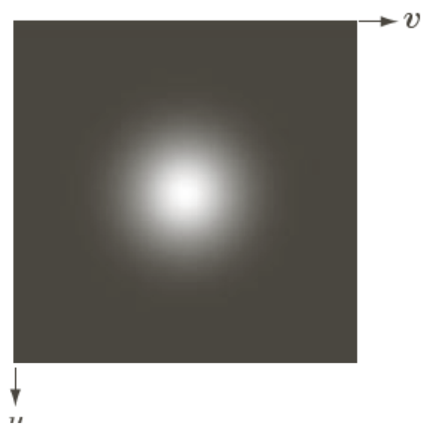
FIGURE 4.42
(a) Frequency domain ILPF transfer function. (b) Corresponding spatial domain kernel function. (c) Intensity profile of a horizontal line through the center of (b).

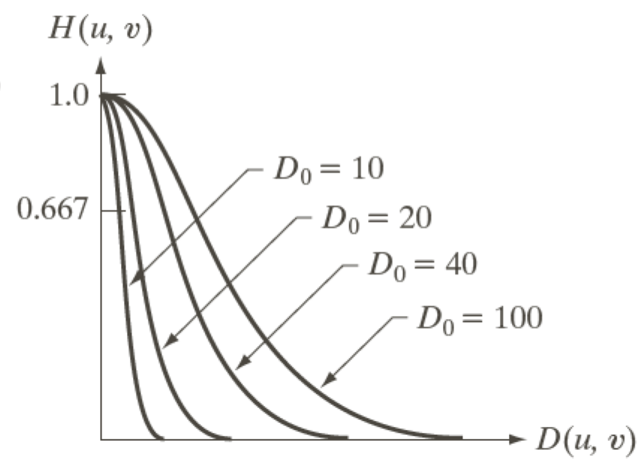
Gaussian Lowpass filter

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

Where $H(u,v) = 0.607$ when $D(u,v) = D_0$





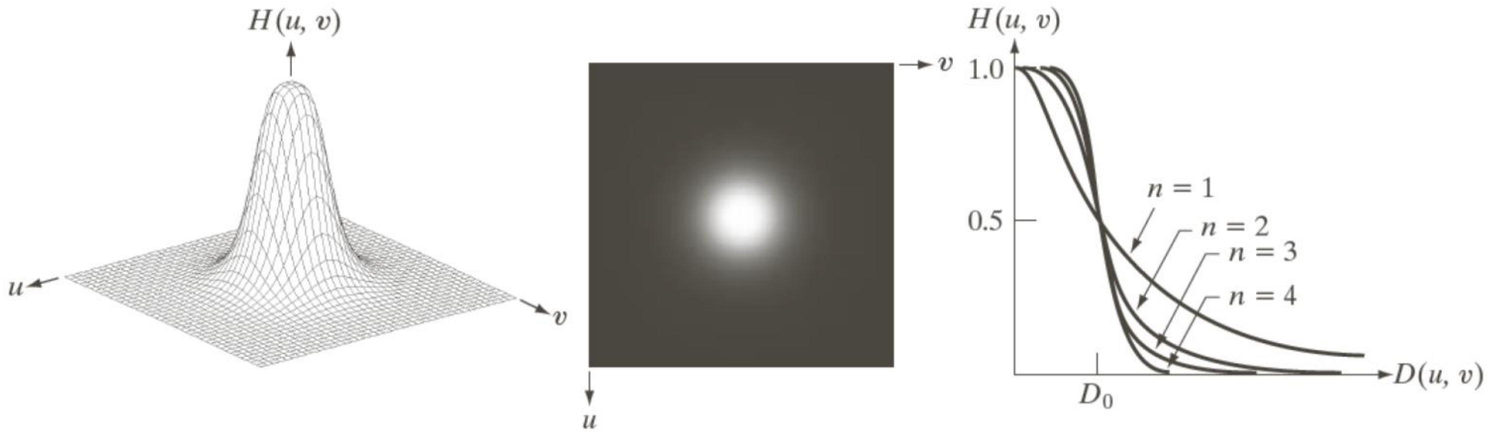


Gaussian Lowpass filter

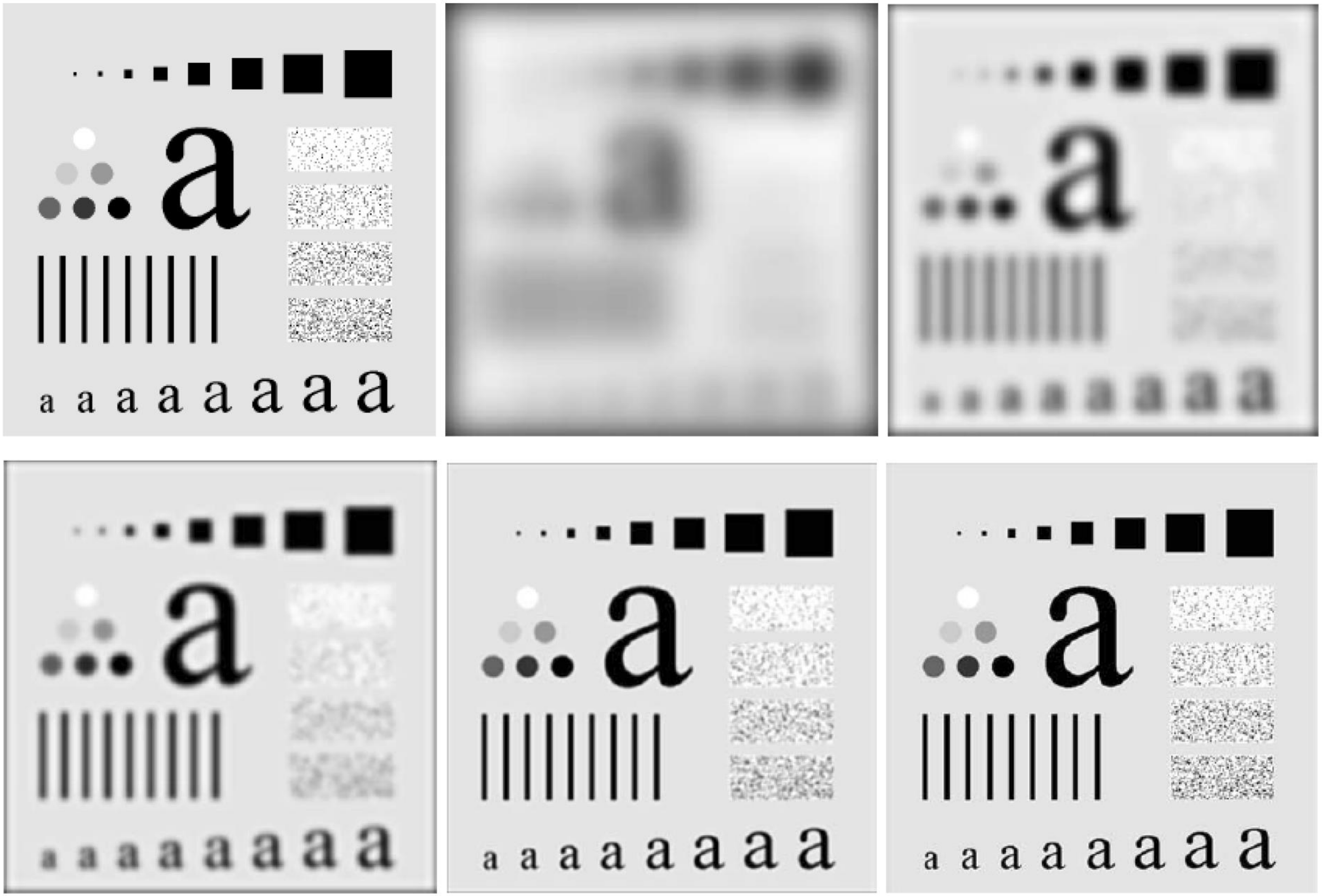
Butterworth Lowpass filter

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

Where $D(u,v) = \left[(u - \frac{P}{2})^2 + (v - \frac{Q}{2})^2\right]^{1/2}$, and $H(u,v) = 0.5$ when $D(u,v) = D_0$



Butterworth Lowpass filter



Butterworth Lowpass filter

- Order of BLPF

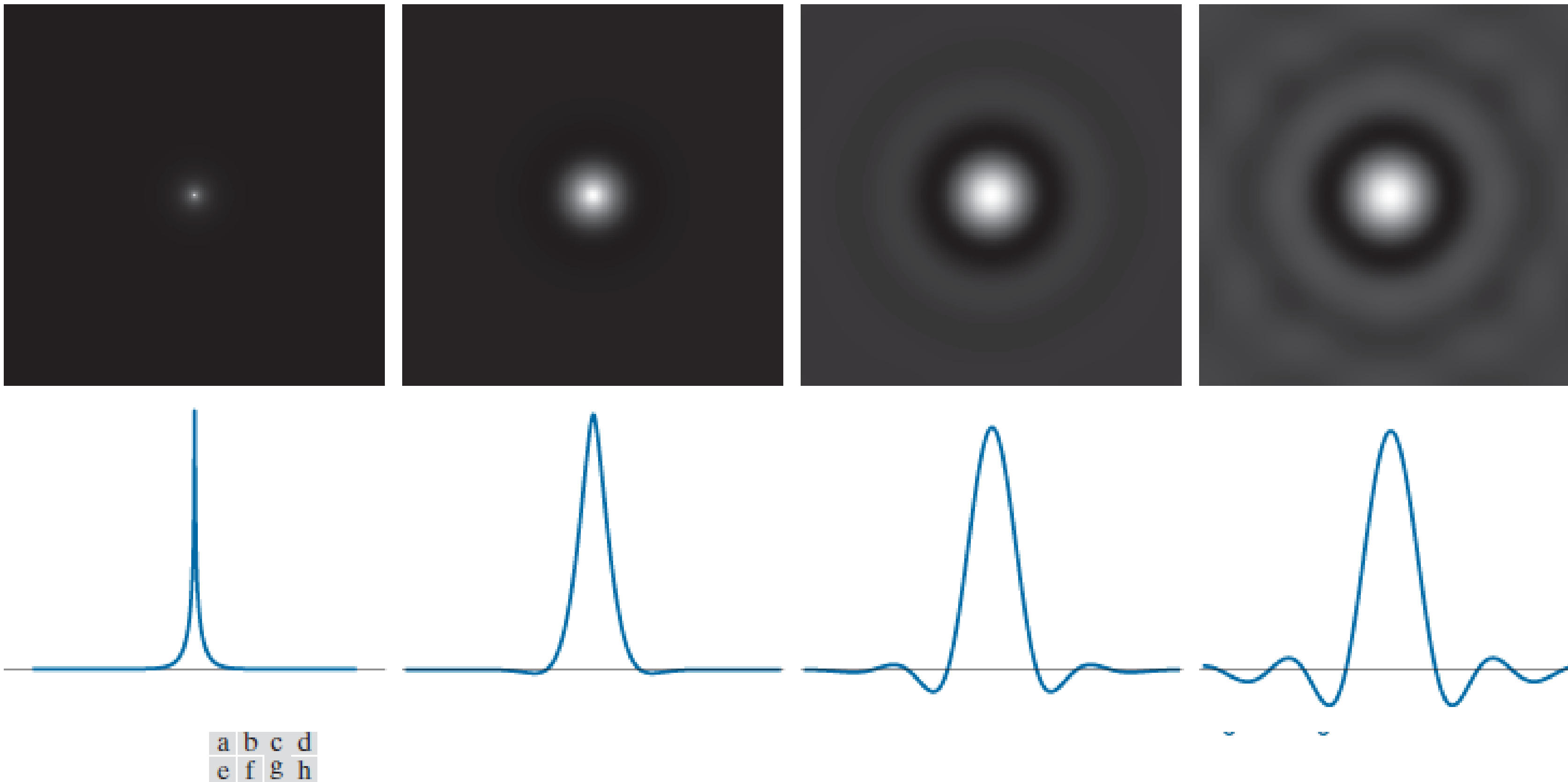
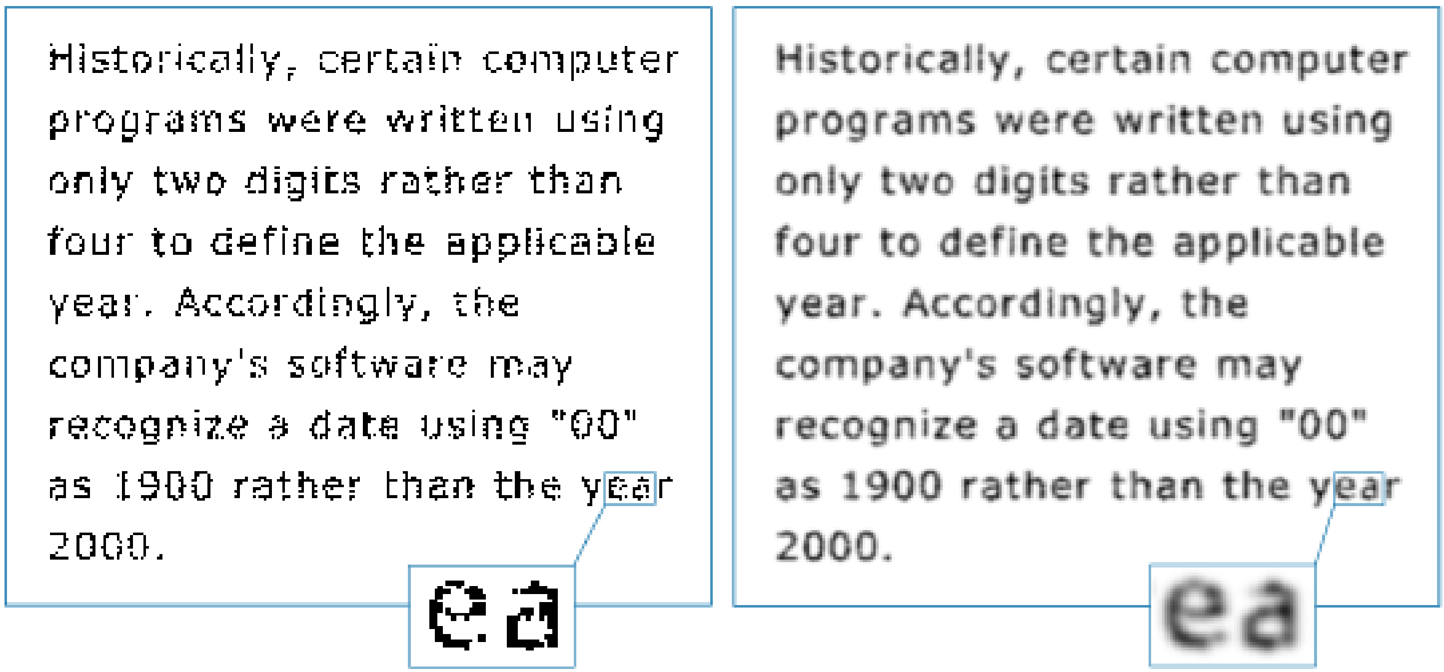


FIGURE 4.47 (a)–(d) Spatial representations (i.e., spatial kernels) corresponding to BLPF transfer functions of
1000×1000 pixels, cut-off frequency of 5, and order 1, 2, 5, and 20, respectively. (e)–(h) Corresponding inten
profiles through the center of the filter functions.

Application of GLPF

a b

FIGURE 4.48
(a) Sample text of low resolution (note the broken characters in the magnified view). (b) Result of filtering with a GLPF, showing that gaps in the broken characters were joined.



Application of GLPF

FIGURE 4.49 (a) Original 785×732 image. (b) Result of filtering using a GLPF with $D_0 = 150$. (c) Result of filtering using a GLPF with $D_0 = 130$. Note the reduction in fine skin lines in the magnified sections in (b) and (c).

Highpass filtering

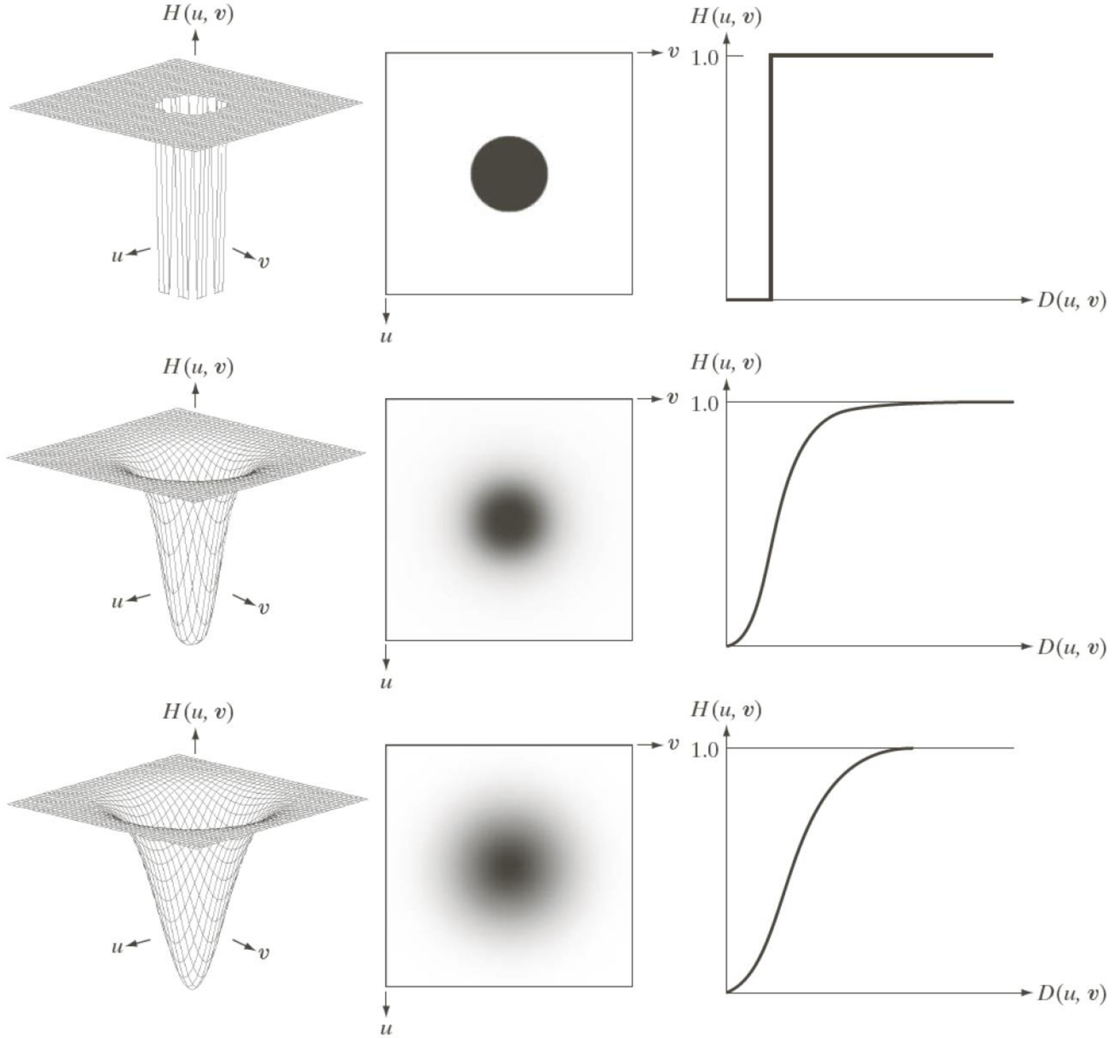
- Ideal Highpass filter
- Butterworth Highpass filter
- Gaussian Highpass filter

$$H_{\mathrm{HP}}(u,v) = 1 - H_{\mathrm{LP}}(u,v)$$

TABLE 4.6
Highpass filter transfer functions. D_0 is the cutoff frequency and n is the order of the Butterworth transfer function.

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

Highpass filtering

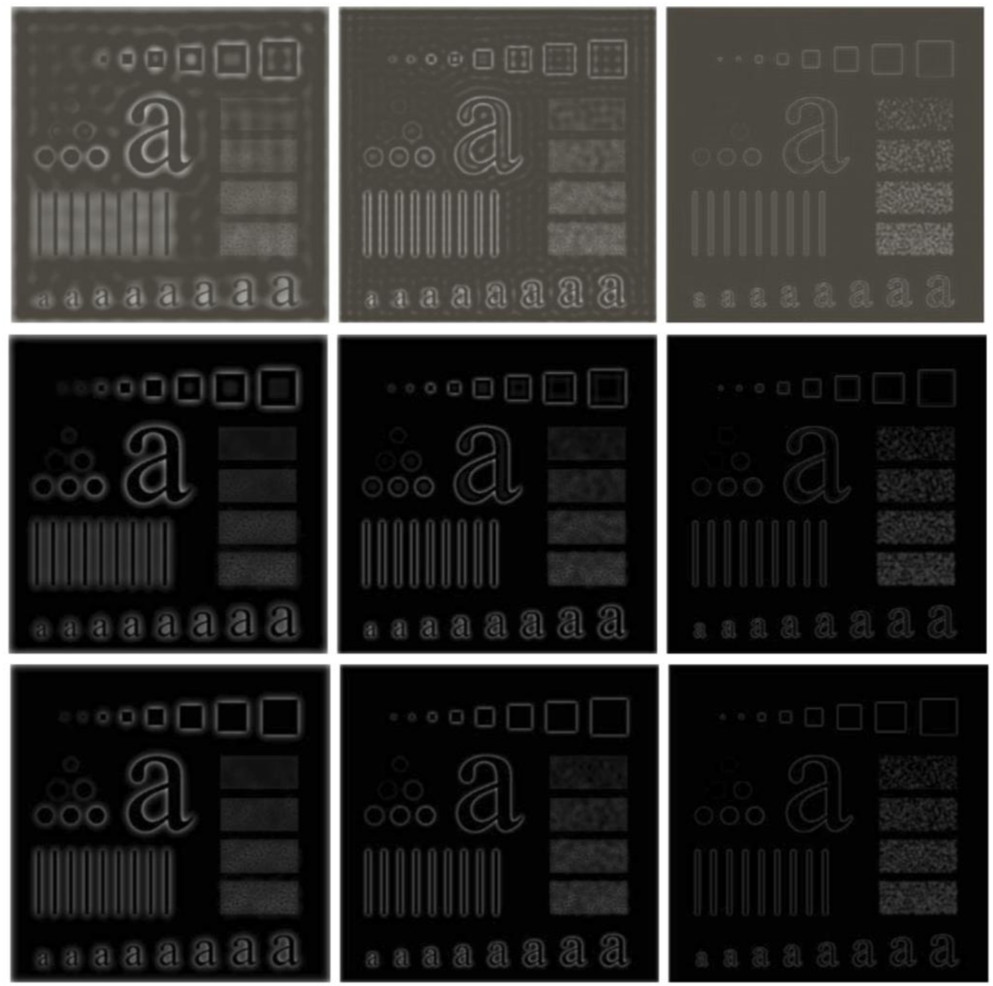


Highpass filtering

IHPF

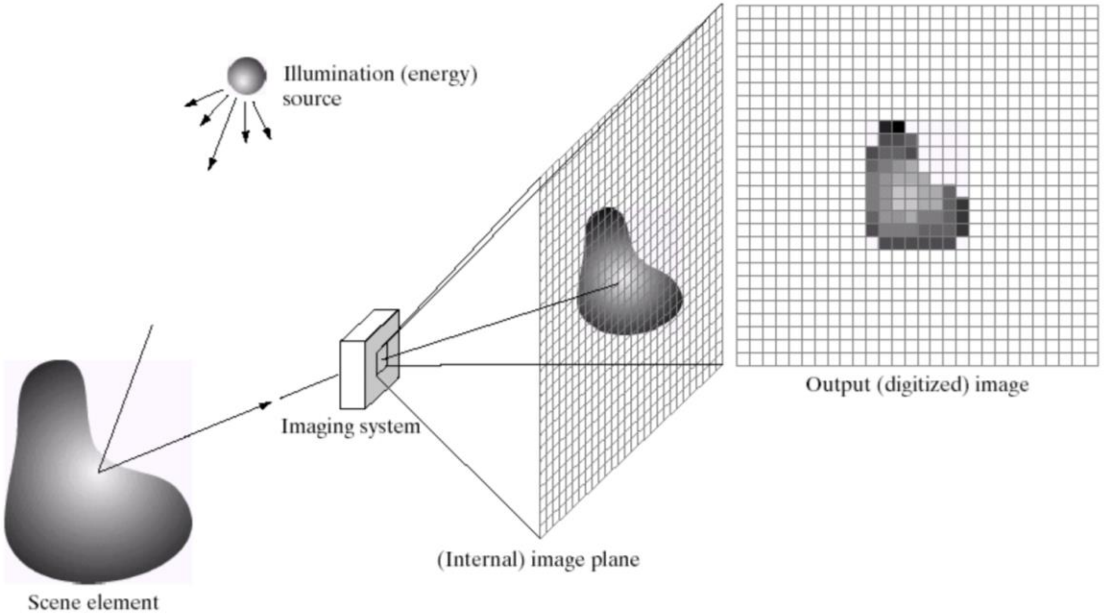
BHPF

GHPF



Highpass Filtering-Homomorphic Filtering

Homomorphic Filtering （同态滤波）



$$f(x,y) = i(x,y)r(x,y) \qquad 0 < i(x,y) < \infty, 0 \leq r(x,y) < 1$$

Homomorphic Filtering

- We first transform the multiplicative components to additive components by moving to the log domain.

$$f(x,y) = i(x,y)r(x,y) \qquad 0 < i(x,y) < \infty, 0 \le r(x,y) < 1$$

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

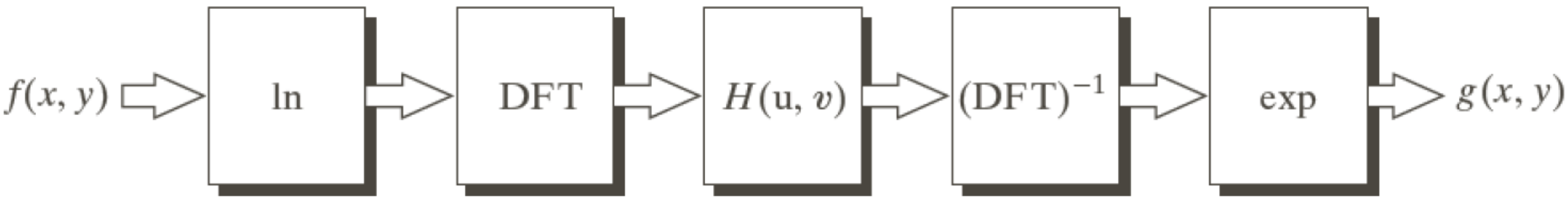
$$\boldsymbol{Z(u,v) = F_i(u,v) + F_r(u,v)}$$

Slow variation   Rapid variation

$$\begin{aligned} s(x,y) &= \mathcal{F}^{-1}[H(u,v)Z(u,v)] \\ &= \mathcal{F}^{-1}[H(u,v)F_i(u,v)] + \mathcal{F}^{-1}[H(u,v)F_r(u,v)] \end{aligned}$$

$$f(x,y) \Rightarrow \boxed{\ln} \Rightarrow \boxed{\text{DFT}} \Rightarrow \boxed{H(u,v)} \Rightarrow \boxed{(\text{DFT})^{-1}} \Rightarrow \boxed{\exp} \Rightarrow g(x,y)$$

Homomorphic Filtering

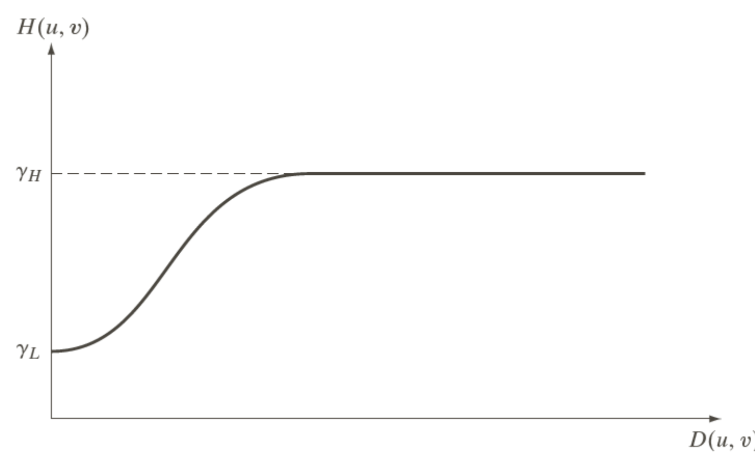


- After filtering the image is reconstructed by a inverted DFT and exponential computation.

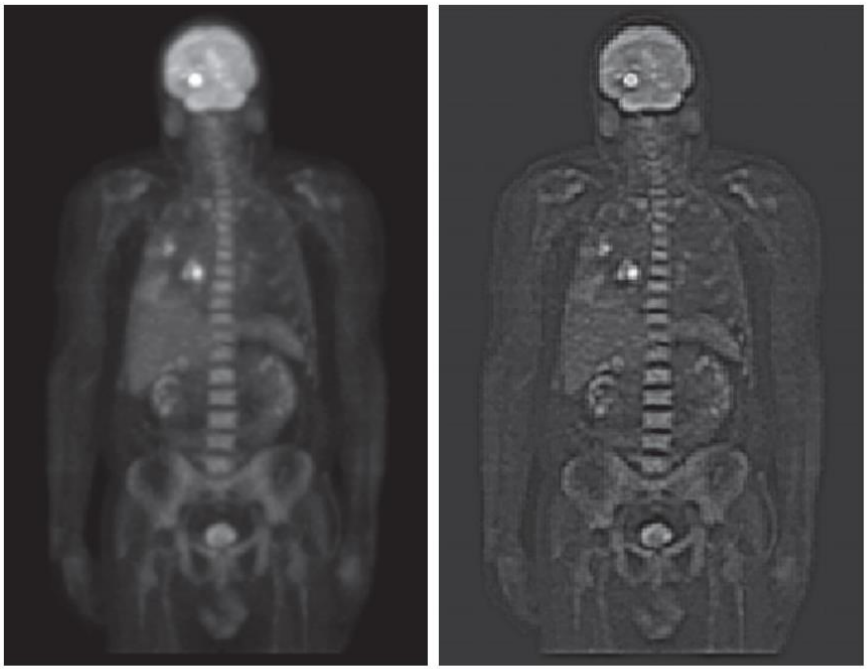
$$\boldsymbol{g(x,y) = e^{s(x,y)} = i_0(x,y)r_0(x,y)}$$

- How to design the H?

$$H(u,v) = (\gamma_H - \gamma_L)\left[1 - e^{-c\left[\frac{D(u,v)}{D_0}\right]^2} + \gamma_L\right]$$



Homomorphic Filtering



a b

FIGURE 4.60
(a) Full body PET scan. (b) Image enhanced using homomorphic filtering. (Original image courtesy of Dr. Michael E. Casey, CTI Pet Systems.)

Homomorphic Filtering

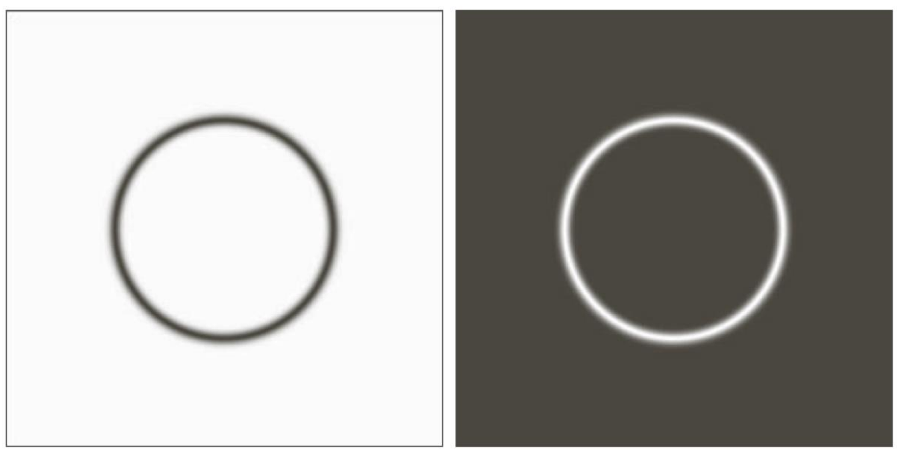
- Homomorphic filtering is most commonly used for correcting non-uniform illumination in images.
- Illumination typically varies slowly across the image as compared to reflectance which can change quite abruptly at object edges.
- We use a high-pass filter in the log domain to remove the low-frequency illumination component while preserving the high-frequency reflectance component.

Selective Filtering

❑ Bandreject and Bandpass filtering

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

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



Selective Filtering

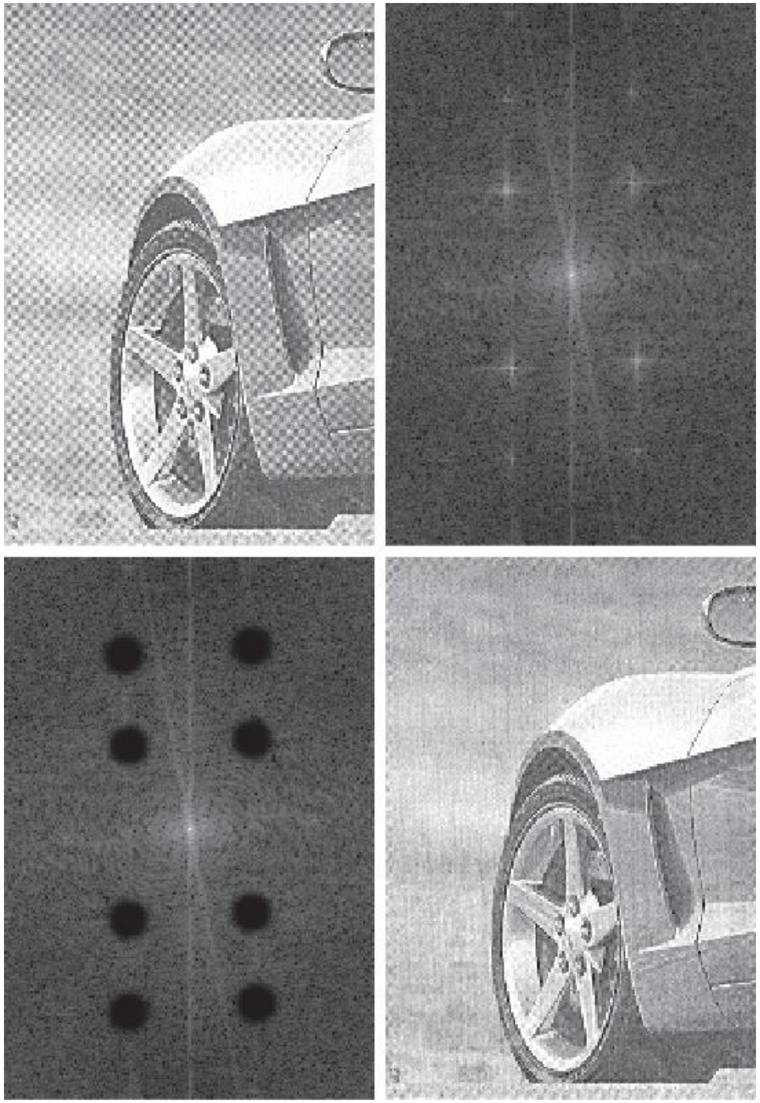
- **Notch Filter** (陷波滤波器)
  - Reject or pass frequencies in predefined neighborhood
  - Symmetric about the origin for a zero-phase shift filters
  - Selectively modify local regions of the DFT

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

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

Where $H_k(u,v)$, $H_{-k}(u,v)$ are Highpass filters with center at (u_k, v_k) and (u_{-k}, v_{-k})

Notch Filter (陷波滤波器)



a	b
c	d

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

Take home message

- Zero-padding in spatial domain is necessary for frequency domain filtering
- Know the procedure of frequency domain filtering
- Frequency domain filtering and spatial domain filtering are related
- Know typical LP/HP filters
- 'Ideal filters' are not ideal
- Homomorphic Filtering is able to reduce abnormal illumination effect in image.