Filtering in Frequency Domain

Instructor: Yiming Xiao

Email: yiming.xiao@concordia.ca

Department of Computer Science
and Software Engineering
Concordia University

Slides modified from materials provided by Dr. Tien D Bui

Frequency Filtering

F(*u,v*) : Fourier transform of given image

H(*u,v*) : Filter transfer function (i.e. FT of filter *h*(*x,y*))

$$G(u,v) = H(u,v)F(u,v) \quad (1)$$

The filtered image *g*(*x,y*) is obtained by IFT of *G*(*u,v*).

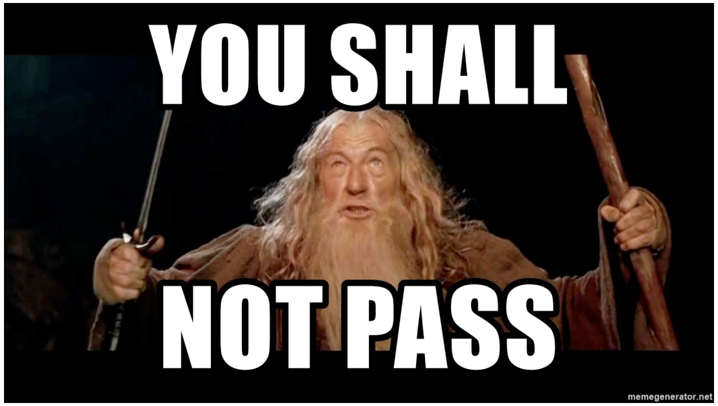
Notes: 1) The slowest varying freq. component (*u*=*v*=0) corresponds to the average intensity. The low freq. corresponds to the slowly varying components.

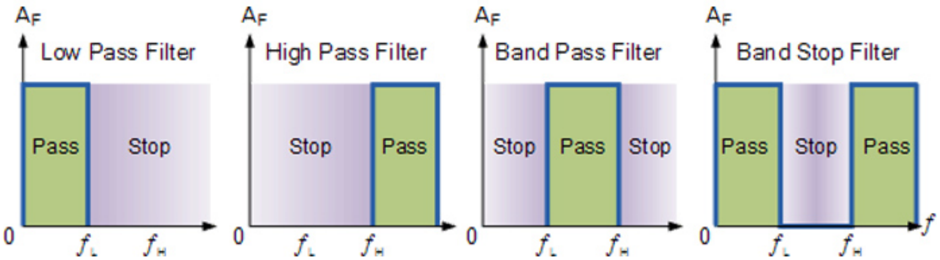
2) Eq. (1) involves the product of two functions in the freq. domain which implies **convolution** in the spatial domain. Therefore the original function must be padded.

3) Filters that affect real and imaginary parts equally have no effects on the phase.

Frequency Filtering

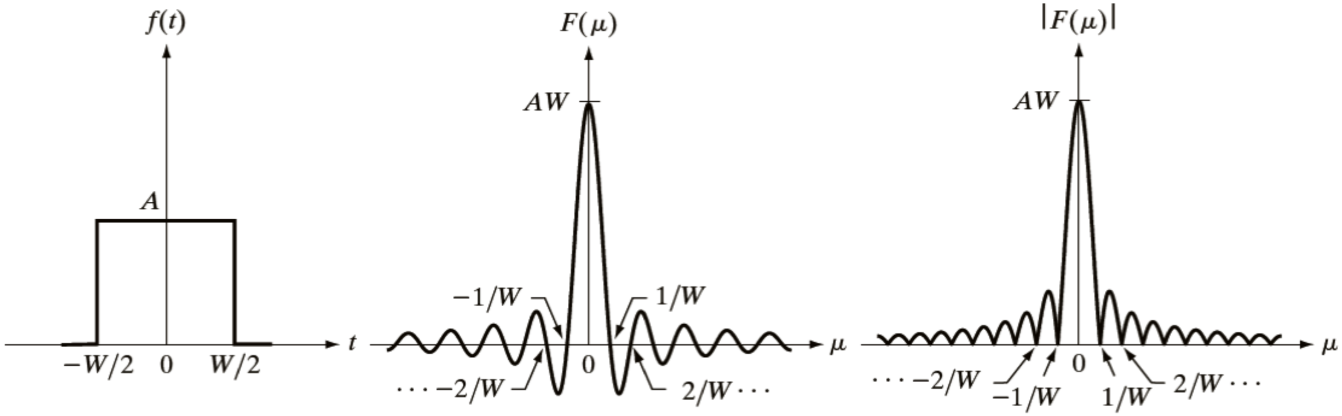
- A **low-pass filter** will allow the low frequency components of the image to pass through, but prevent the high frequency components to go through. Therefore a low-pass filter will blur the image.
- A **high-pass filter** will do the opposite: it allows the high frequency components to pass through while preventing the low frequency components. Therefore a high-pass filter will sharpen the image.
- A **band-pass filter** will allow selected frequency range to pass





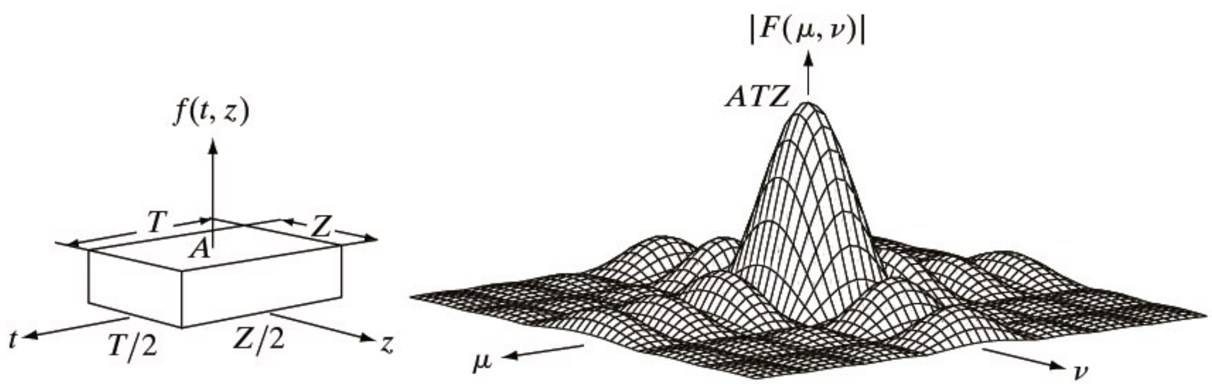
https://www.sunpower-uk.com/glossary/what-is-a-frequency-filter/

Fourier Transform Revisited



a b c

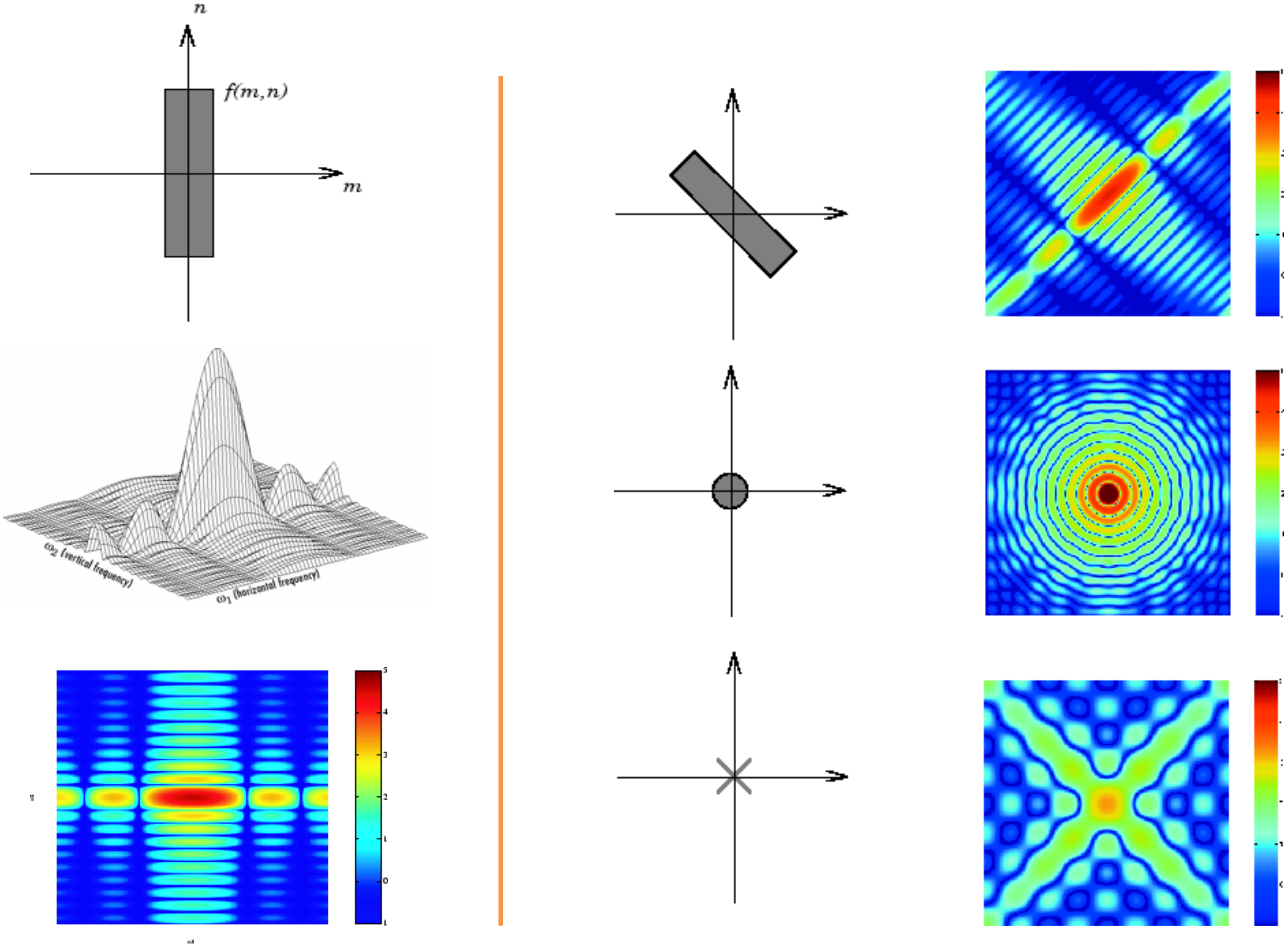
FIGURE 4.4 (a) A simple function; (b) its Fourier transform; and (c) the spectrum. All functions extend to infinity in both directions.



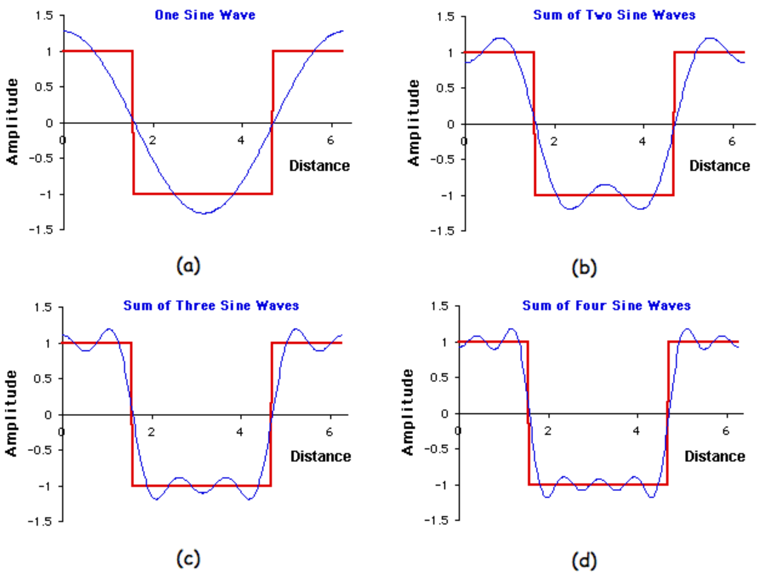
a b

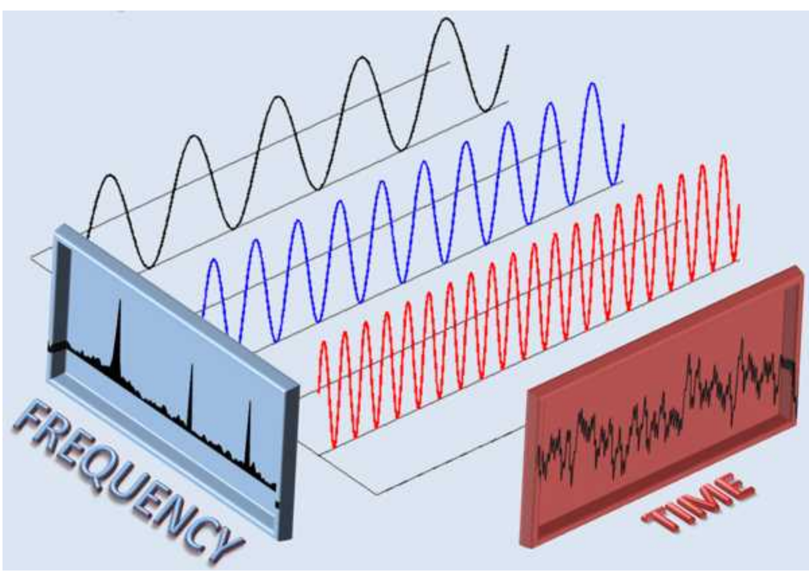
FIGURE 4.13 (a) A 2-D function, and (b) a section of its spectrum (not to scale). The block is longer along the t-axis, so the spectrum is more "contracted" along the μ-axis. Compare with Fig. 4.4.

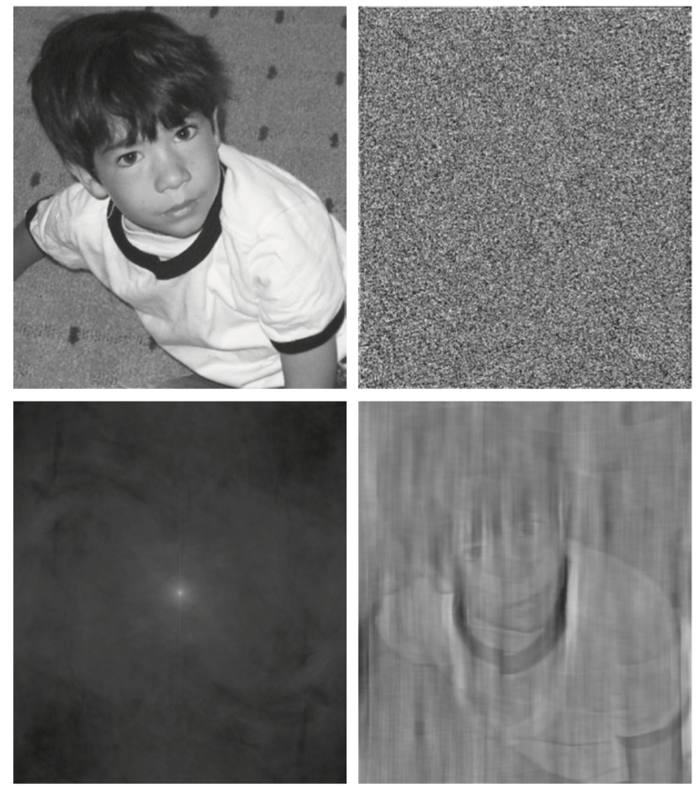
Fourier Transform Revisited

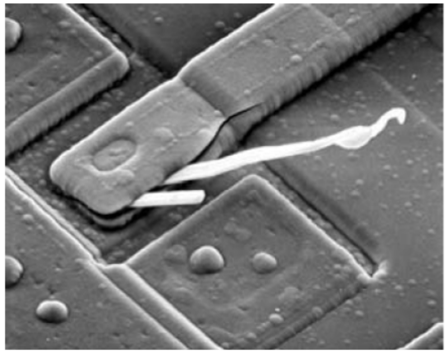


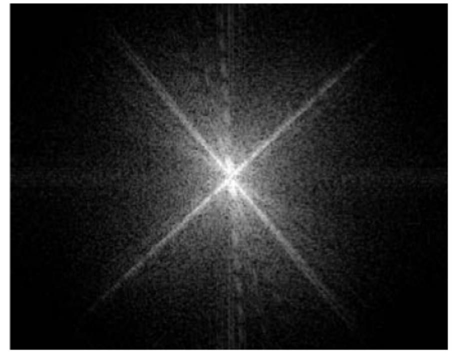
Courtesy of The MathWorks Inc.

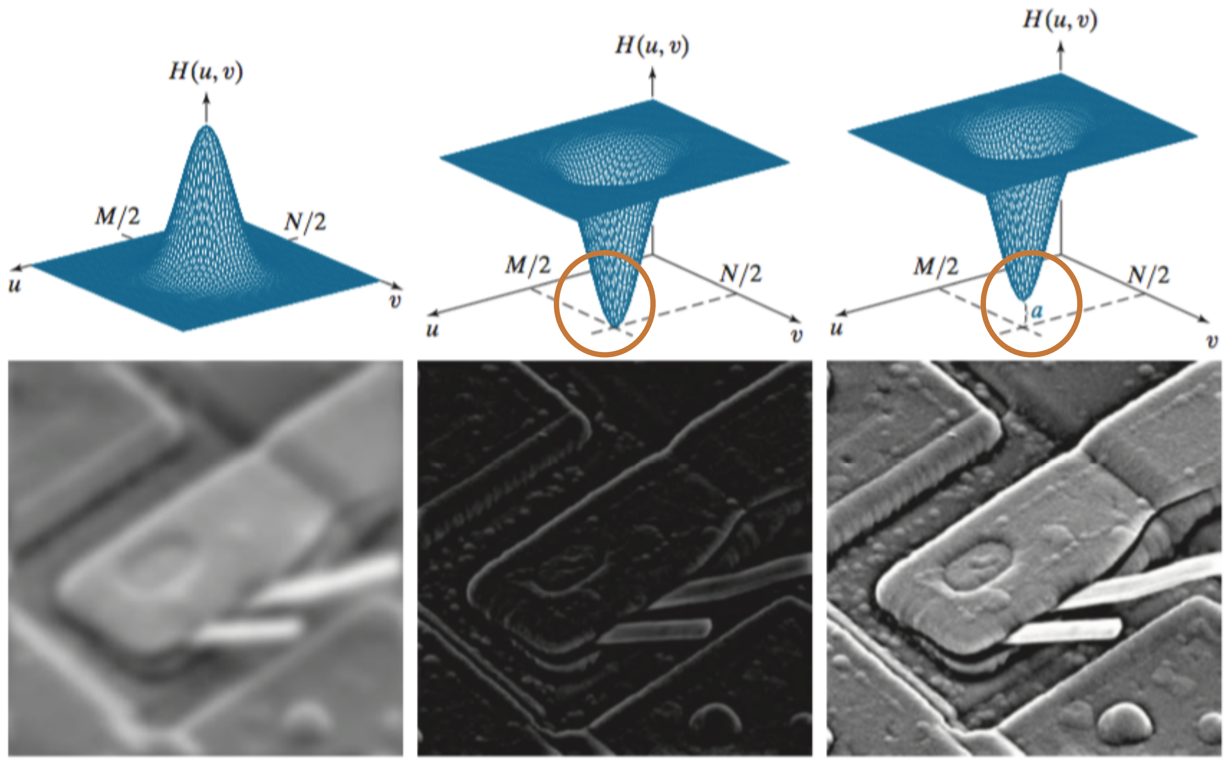

One Sine Wave
Sum of Two Sine Waves
Sum of Three Sine Waves
Sum of Four Sine Waves
Amplitude
Distance
(a)
(b)
(c)
(d)


FREQUENCY
TIME









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).

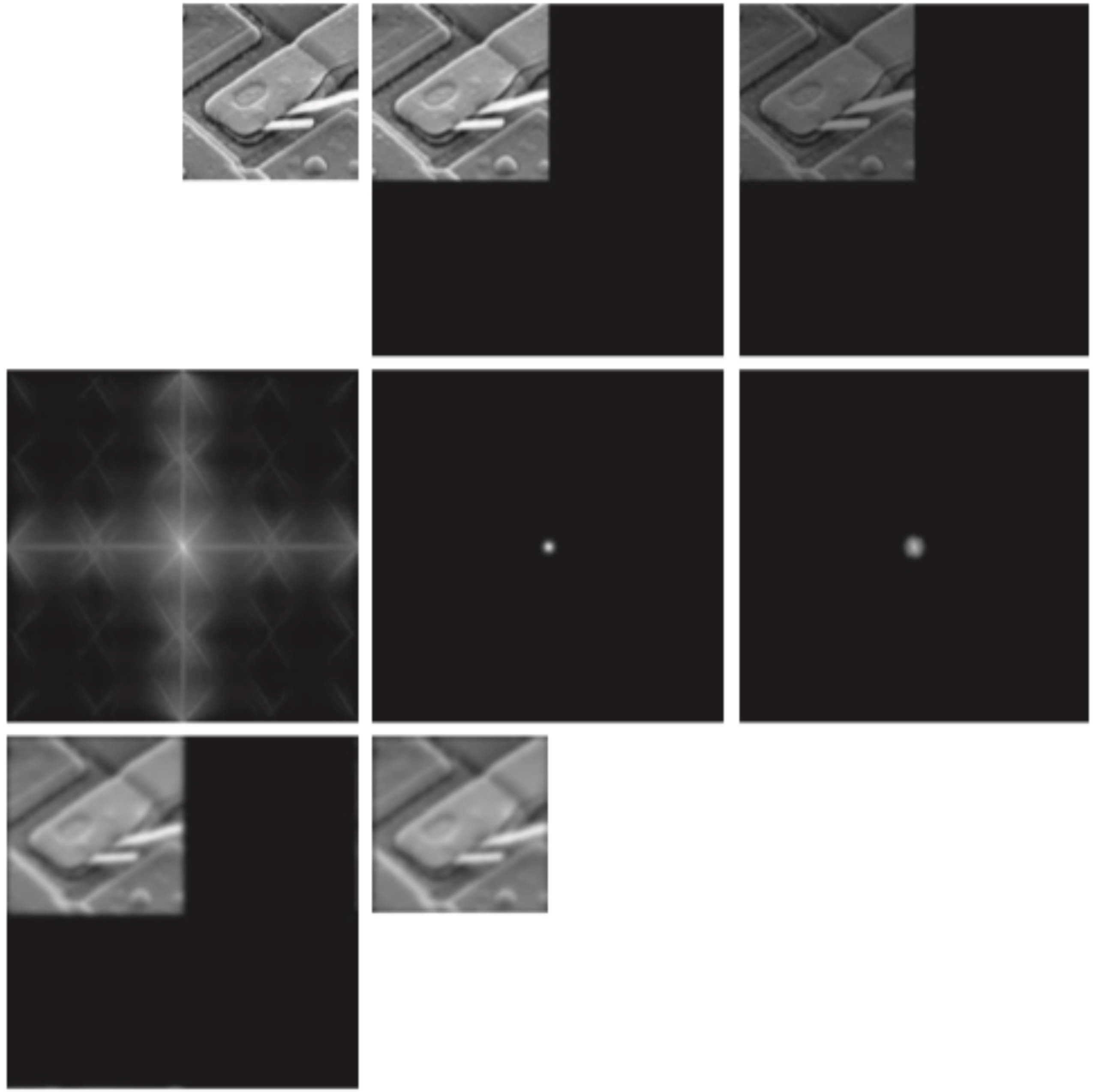
(Practical) Steps for Filtering in the Frequency Domain

1. Given an image $f(x,y)$ of size N x N, obtain the padded image $f_p(x,y)$ of size ($2N$ x $2N$).
2. Multiply $f_p(x,y)$ by $(-1)^{x+y}$ to center its Fourier transform.
3. Compute the DFT, $F(u,v)$, of the image in step 2.
4. Generate a real, symmetric filter function of size ($2N$ x $2N$). Form the product $G(u,v) = H(u,v)F(u,v)$ using array multiplication ($G(i,k)=H(i,k)\ F(i,k)$).
5. Obtain the processed image:

$$g_p(x,y) = \{Real[F^{-1}(G(u,v))]\}(-1)^{x+y}$$

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

Note: The exponential term $e^{j\pi(x+y)}$ which is equal to $(-1)^{x+y}$ because x, y are integers. Hence: $f(x,y)(-1)^{x+y} \Leftrightarrow F(u-N/2, v-N/2)$



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.

Why Filtering in the Frequency Domain?

For an M x N image to be filtered by a m x n filter window

If we perform filter in spatial domain: $MNmn$

*If the spatial filter is **separable** :* $MN(m+n)$

If we perform filter in frequency domain (with FFT): $2MN \log_2(MN)$

For an M x M image to be filtered by a m x m filter window

The ratio between the two:

$$C_n(m) = \frac{M^2 m^2}{2M^2 \log_2 M^2} = \frac{m^2}{4 \log_2 M}$$

*If spatial filter is **separable**:*

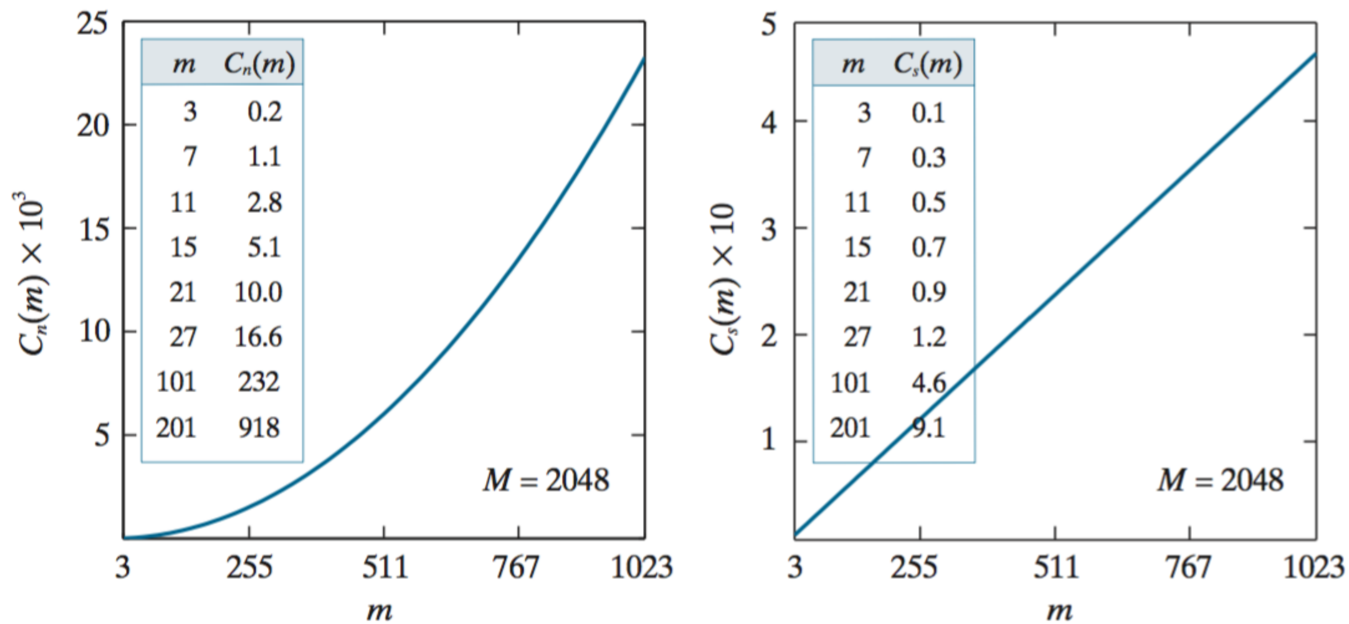
$$C_s(m) = \frac{2M^2 m}{2M^2 \log_2 M^2} = \frac{m}{2 \log_2 M}$$

C>1: Frequency

otherwise: spatial

a b

FIGURE 4.2 (a) Computational advantage of the FFT over non-separable spatial kernels. (b) Advantage over separable kernels. The numbers for $C(m)$ in the inset tables are not to be multiplied by the factors of 10 shown for the curves.



Frequency Filtering: Smoothing

Smoothing can be achieved in the frequency domain by attenuating a specified range of high-frequency components.

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

F : Fourier transform of given image;

H : ideal lowpass filter (ILPF)

Ideal Low-pass Filter

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

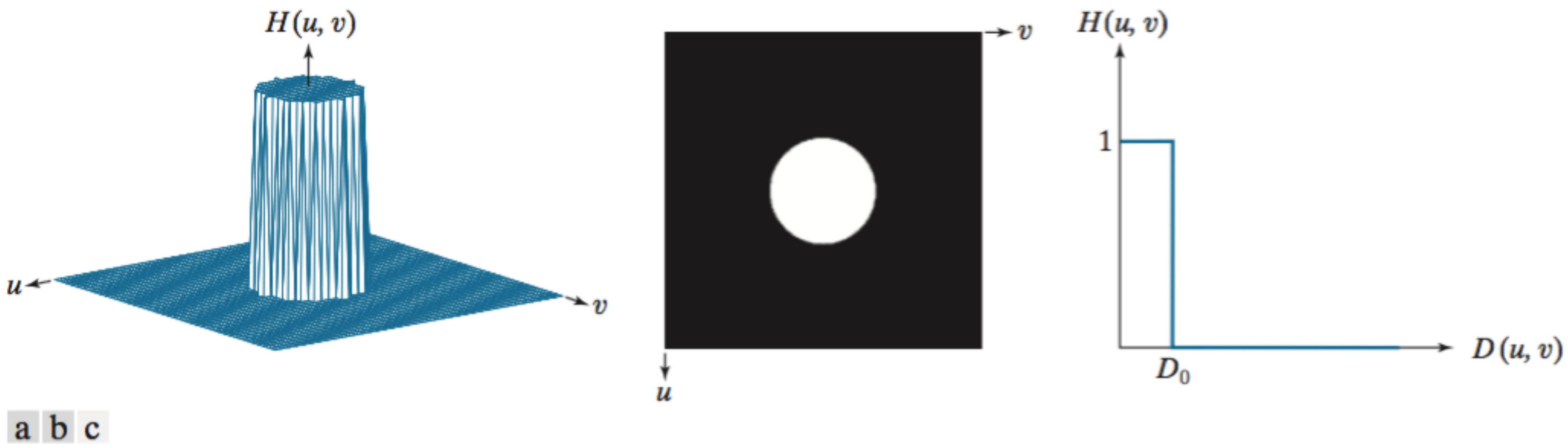


FIGURE 4.39 (a) Perspective plot of an ideal lowpass-filter transfer function. (b) Function displayed as an image. (c) Radial cross section.

Frequency Filtering: Smoothing

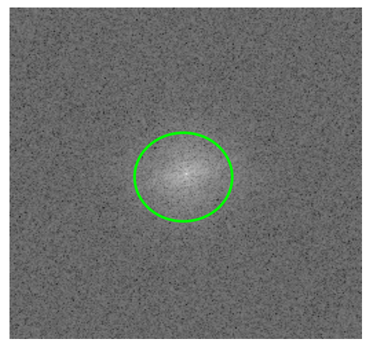
Smoothing can be achieved in the frequency domain by attenuating a specified range of high-frequency components.

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

F : Fourier transform of given image;

H : ideal lowpass filter (ILPF)

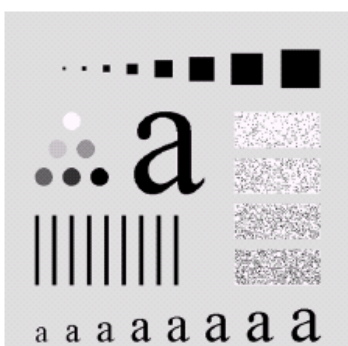
Ideal Low-pass Filter

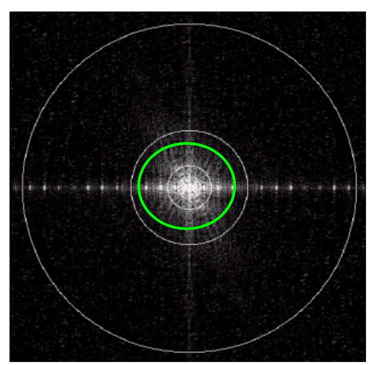


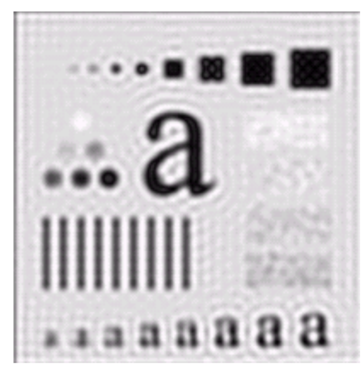
Given image

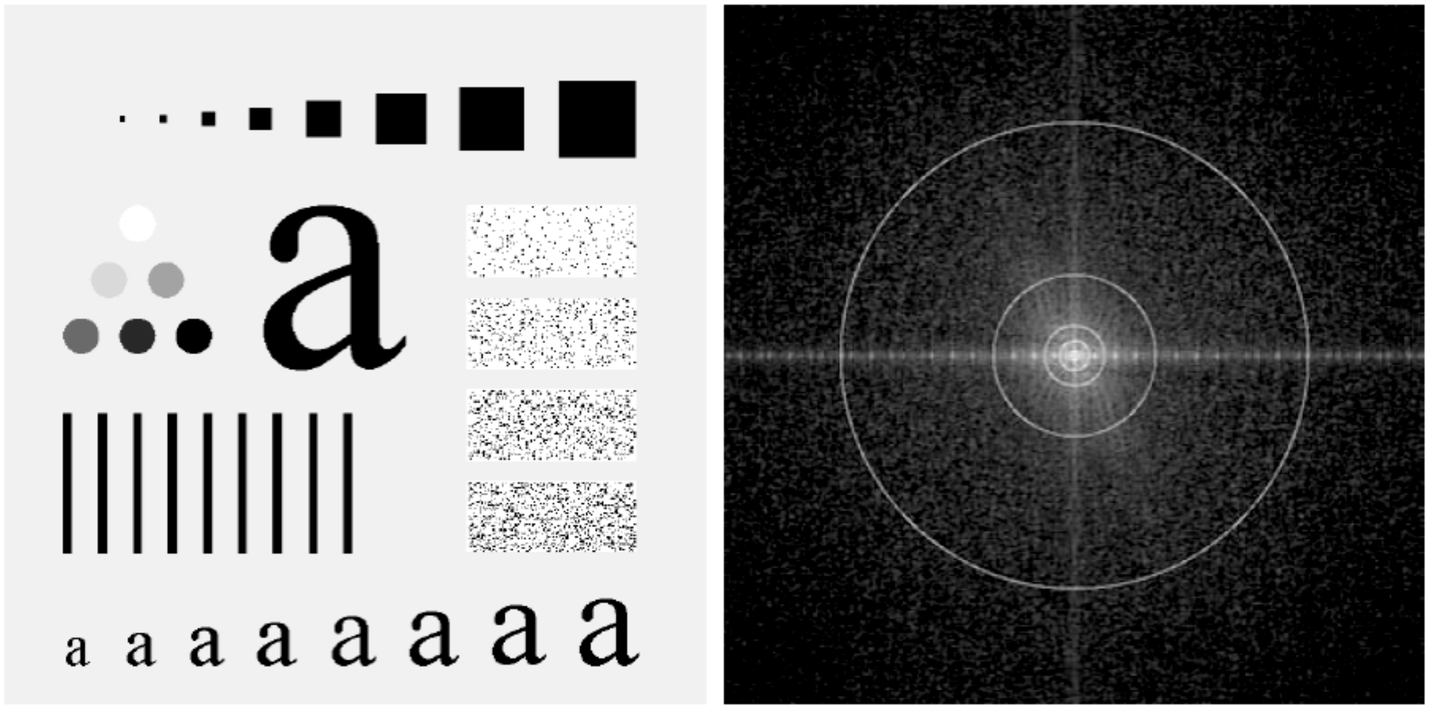
Cut-off of ILPF: green circle

After lowpass filtering









a b

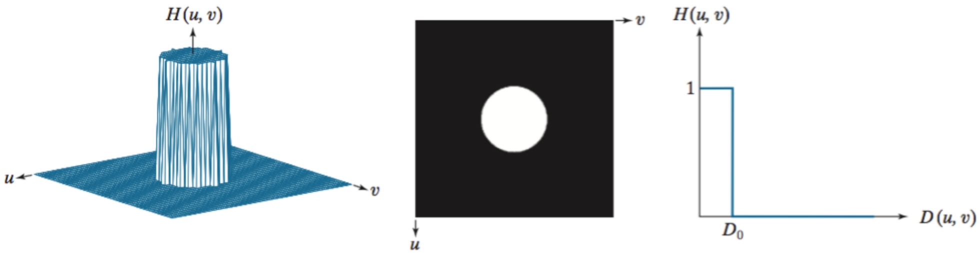
FIGURE 4.41 (a) Test pattern of size 688 × 688 pixels, and (b) its Fourier spectrum. The spectrum is double the image size due to padding but is shown in half size so that it fits in the page. The superimposed circles have radii equal to 10, 30, 60, 160, and 460 with respect to the full-size spectrum image. These radii enclose 87.0, 93.1, 95.7, 97.8, and 99.2% of the padded image power, respectively.

Ideal Lowpass Filters (ILPFs)

$$R = 10, 30, 60, 160, \text{and } 460$$

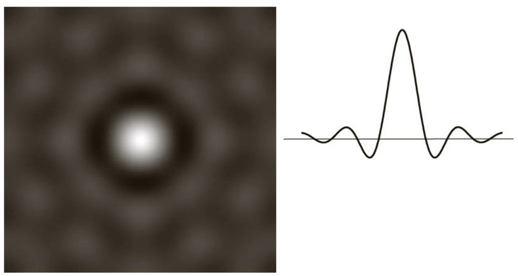
FIGURE 4.42 (a) Original image. (b)–(f) Results of filtering using ILPFs with cutoff frequencies set at radii values 10, 30, 60, 160, and 460, as shown in Fig. 4.41(b). The power removed by these filters was 13, 6.9, 4.3, 2.2, and 0.8% of the total, respectively.

Why ILPFs Have Ringing Effects?



a b c

FIGURE 4.39 (a) Perspective plot of an ideal lowpass-filter transfer function. (b) Function displayed as an image. (c) Radial cross section.



a b

FIGURE 4.43 (a) Representation in the spatial domain of an ILPF of radius 5 and size 1000 × 1000. (b) Intensity profile of a horizontal line passing through the center of the image.

Sinc function

This again comes from the convolution theorem : multiplication in the freq. domain is corresponding to convolution in the spatial domain. The right figure shows the ILPF, $h(x, y)$, in spatial domain. The process of filtering in the freq. domain is equivalent to the convolution of $h(x, y)$ (which is a sinc func.) with the image in spatial domain. Hence riging effects. How to avoid them?

Frequency Filtering: Smoothing

Butterworth Low-pass Filter

TABLE 4.4
Lowpass filters. D_0 is the cutoff frequency and n is the order of the Butterworth filter.

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

Butterworth is the name of a British physicist (1885-1958)

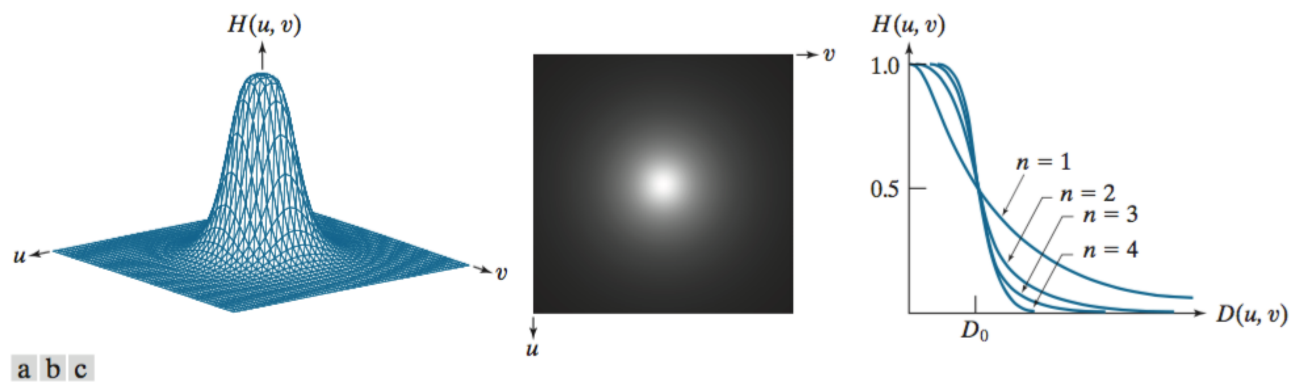


FIGURE 4.45 (a) Perspective plot of a Butterworth lowpass-filter transfer function. (b) Function displayed as an image. (c) Radial cross sections of BLPFs of orders 1 through 4.

Frequency Filtering: Smoothing

Butterworth Low-pass Filter

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

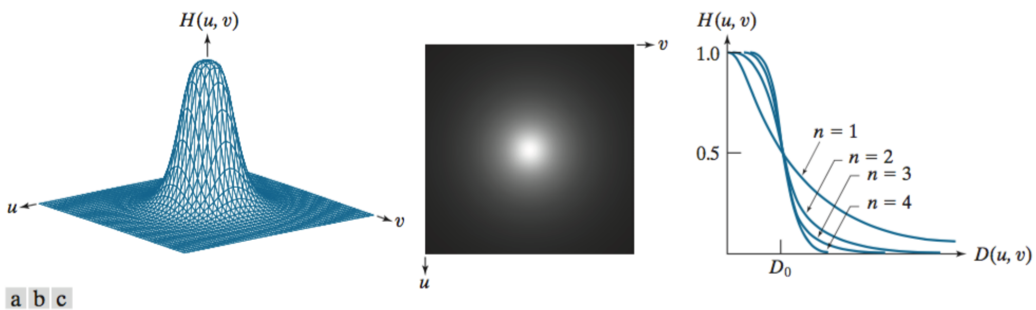


FIGURE 4.45 (a) Perspective plot of a Butterworth lowpass-filter transfer function. (b) Function displayed as an image. (c) Radial cross sections of BLPFs of orders 1 through 4.

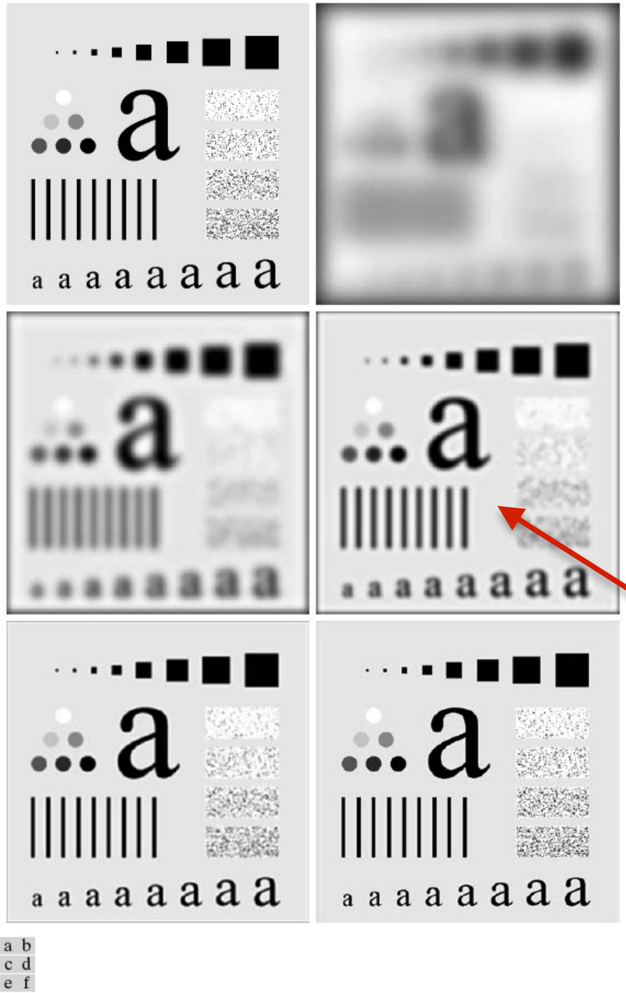
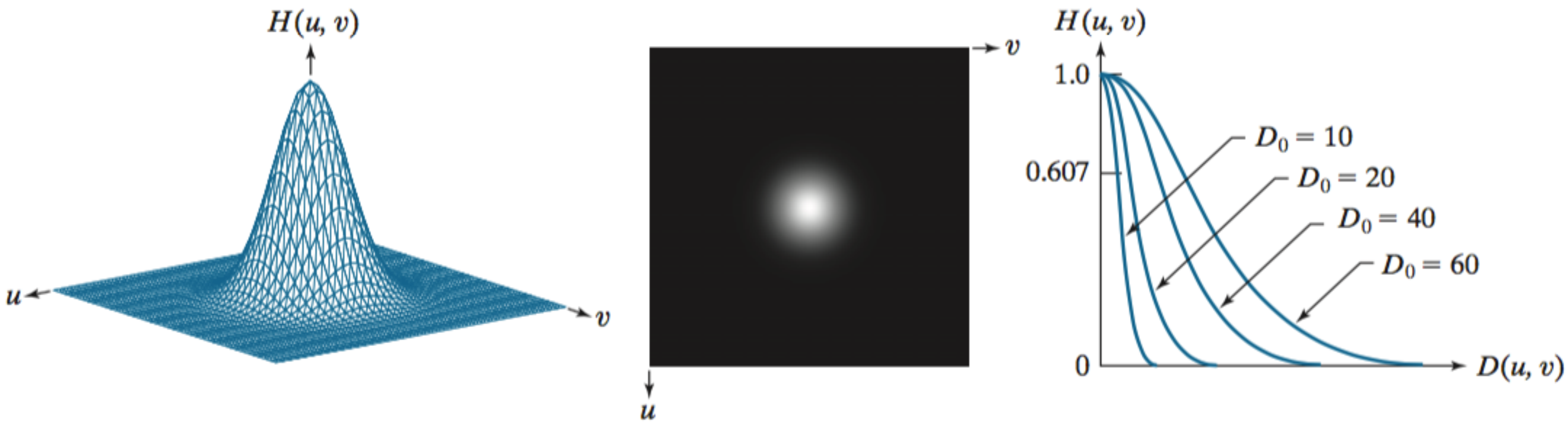


FIGURE 4.45 (a) Original image. (b)–(f) Results of filtering using BLPFs of order 2, with cutoff frequencies at the radii shown in Fig. 4.41. Compare with Fig. 4.42.

Frequency Filtering: Smoothing

Gaussian Low-pass Filter



a b c

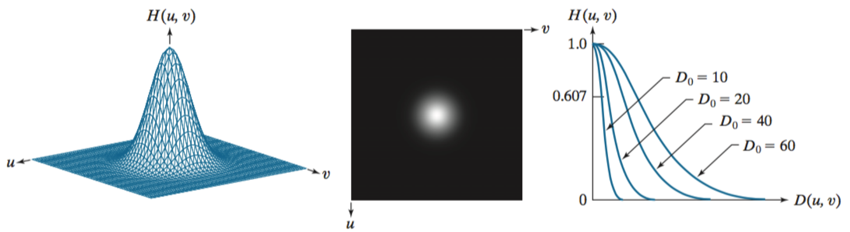
FIGURE 4.43 (a) Perspective plot of a GLPF transfer function. (b) Function displayed as an image. (c) Radial cross sections for various values of D_0.

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

$$H(u) = Ae^{-u^2/2\sigma^2} \Leftrightarrow h(x) = \sqrt{2\pi}\sigma \; Ae^{-2\pi^2\sigma^2x^2}$$

Frequency Filtering: Smoothing

Gaussian Low-pass Filter

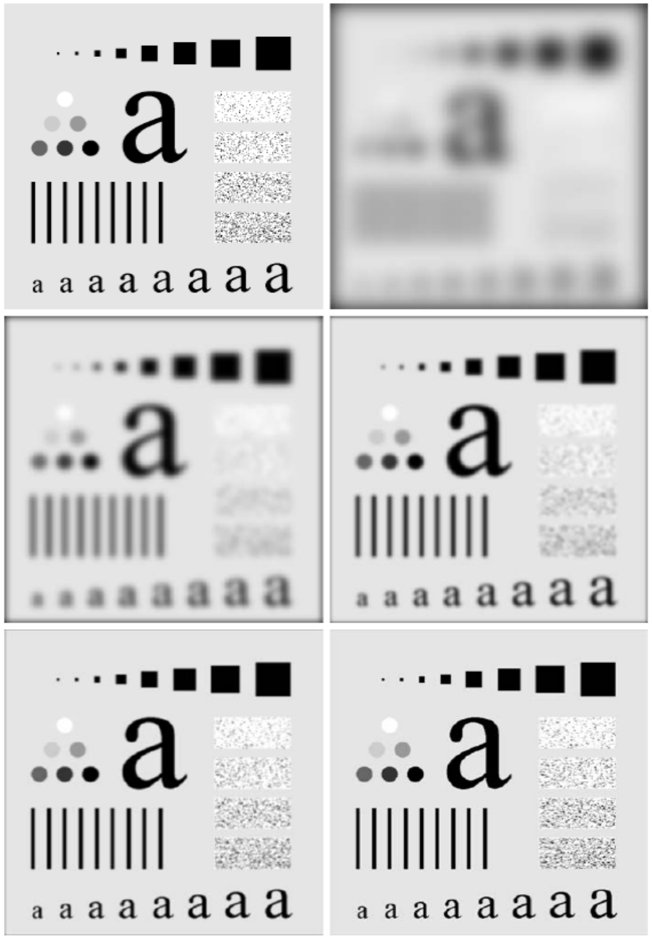


a b c

FIGURE 4.43 (a) Perspective plot of a GLPF transfer function. (b) Function displayed as an image. (c) Radial cross sections for various values of D_0.

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

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

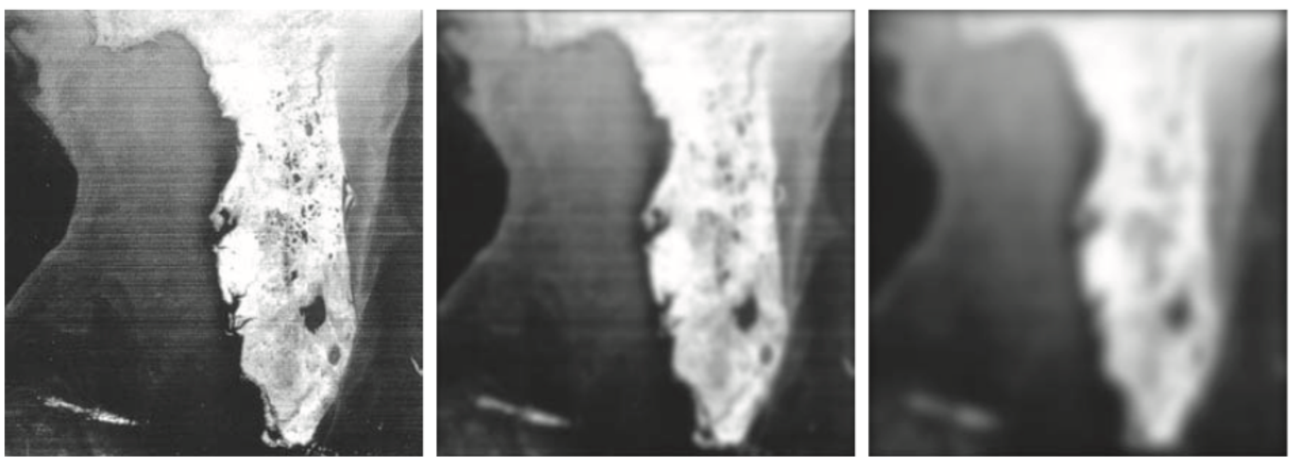


a b
c d
e f

FIGURE 4.48 (a) Original image. (b)–(f) Results of filtering using GLPFs with cutoff frequencies at the radii shown in Fig. 4.41. Compare with Figs. 4.42 and 4.45.

a b c

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).



a b c

FIGURE 4.50 (a) 808×754 satellite 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.)

Frequency Filtering: Smoothing

Bilateral filtering: edge preservation

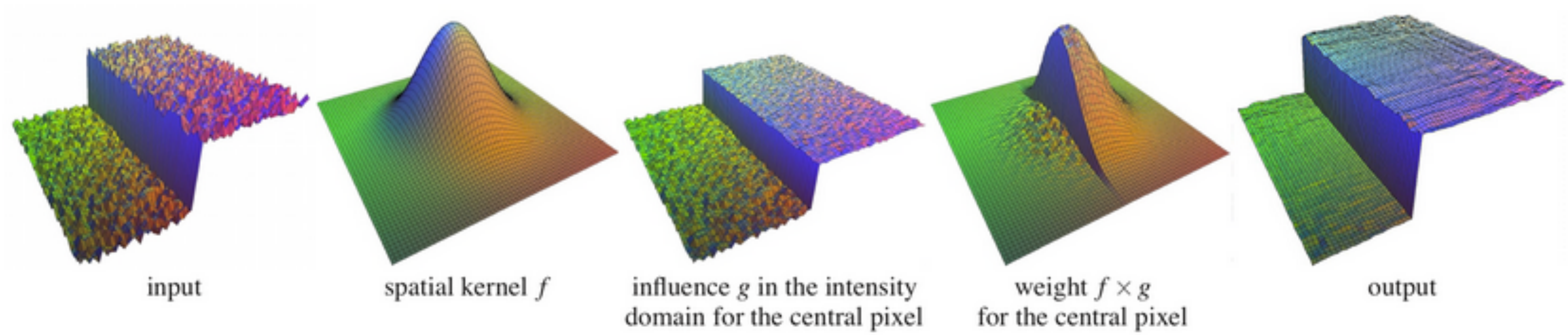


Fig 5. An illustration of how the spatial and color (intensity) weights combine to preserve edges.
Image from *Fast Bilateral Filtering for the Display of High-Dynamic-Range Images*, Durand and Dorsey.

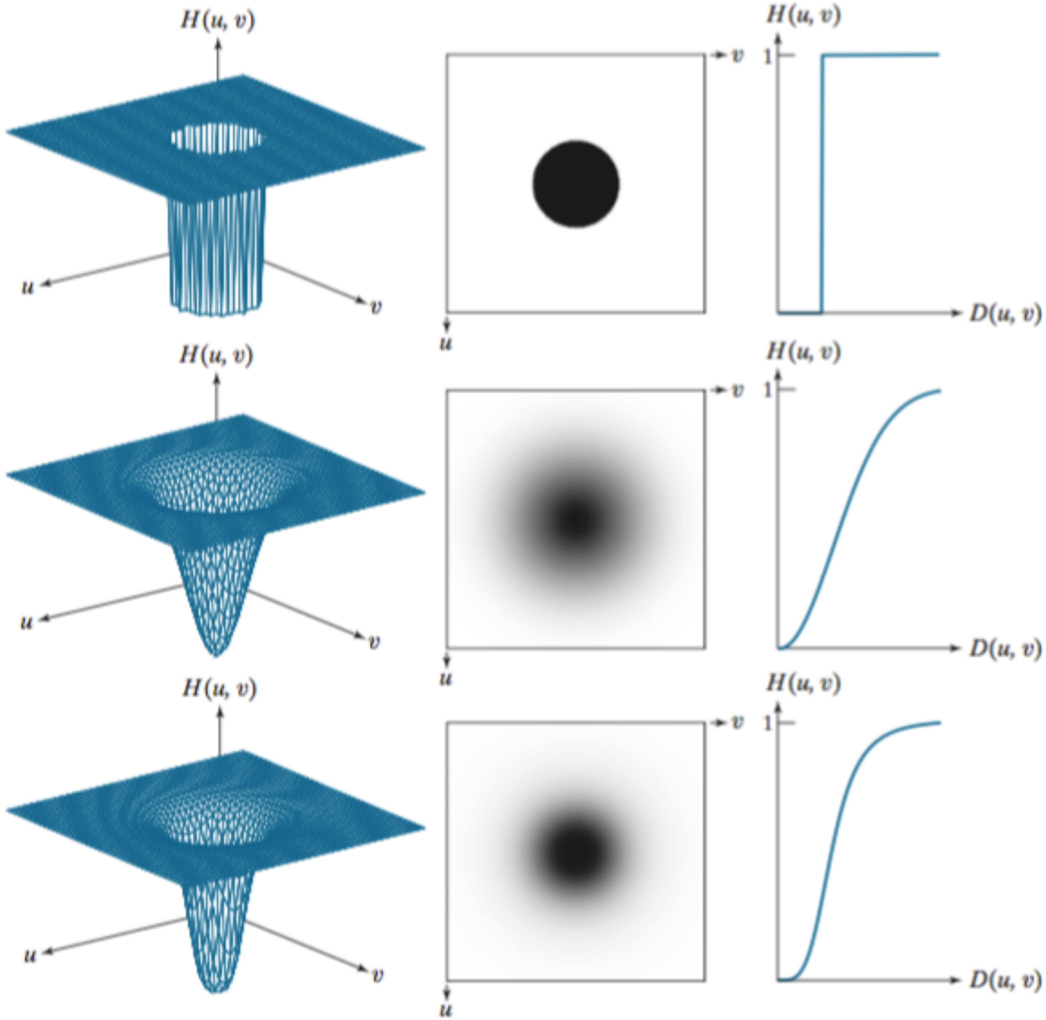
C. Tomasi and R. Manduchi, "Bilateral Filtering for Gray and Color Images", *Proceedings of the 1998 IEEE International Conference on Computer Vision*, Bombay, India.

Frequency Filtering: Sharpening

Ideal High-pass Filter

Butterworth High-pass Filter

Gaussian High-pass Filter



a b c
d e f
g h i

FIGURE 4.51 Top row: Perspective plot, image, and, radial cross section of an IHPF transfer function. Middle and bottom rows: The same sequence for GHPF and BHPF transfer functions. (The thin image borders were added for clarity. They are not part of the data.)

Frequency Filtering: Sharpening

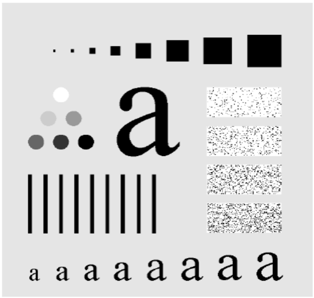
TABLE 4.6
Highpass filter transfer functions. D_0 is the cutoff frequency and n is the order of the Butterworth transfer function.

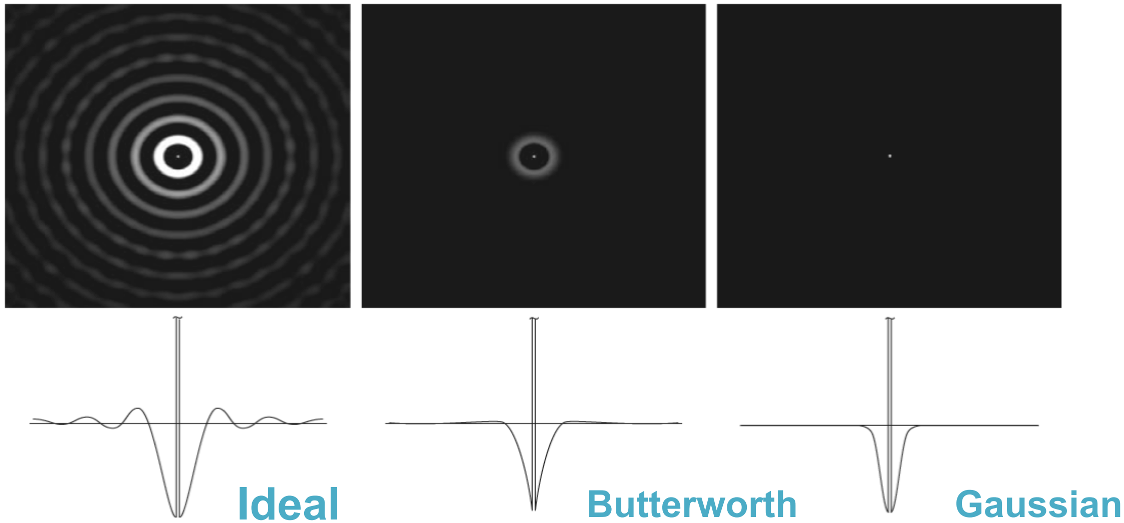
Ideal	Gaussian	Butterworth
$H(u,v) = \begin{cases} 0 & \text{if } D(u,v) \le D_0 \\ 1 & \text{if } D(u,v) > D_0 \end{cases}$	$H(u,v) = 1 - e^{-D^2(u,v)/2D_0^2}$	$H(u,v) = \dfrac{1}{1 + [D_0/D(u,v)]^{2n}}$

High pass filter = 1 - Low-pass filter

$$
\begin{aligned}
h_{\mathrm{HP}}(x,y) &= \Im^{-1}\left[H_{\mathrm{HP}}(u,v)\right] \\
&= \Im^{-1}\left[1 - H_{\mathrm{LP}}(u,v)\right] \\
&= \delta(x,y) - h_{\mathrm{LP}}(x,y)
\end{aligned}
$$

Frequency Filtering: Sharpening





Spatial domain

a b c

FIGURE 4.53 Spatial representation of typical (a) ideal, (b) Butterworth, and (c) Gaussian frequency domain highpass filters, and corresponding intensity profiles through their centers.

Ideal

a b c

FIGURE 4.54 Results of highpass filtering the image in Fig. 4.41(a) using an IHPF with $D_0 = 30, 60$, and 160.

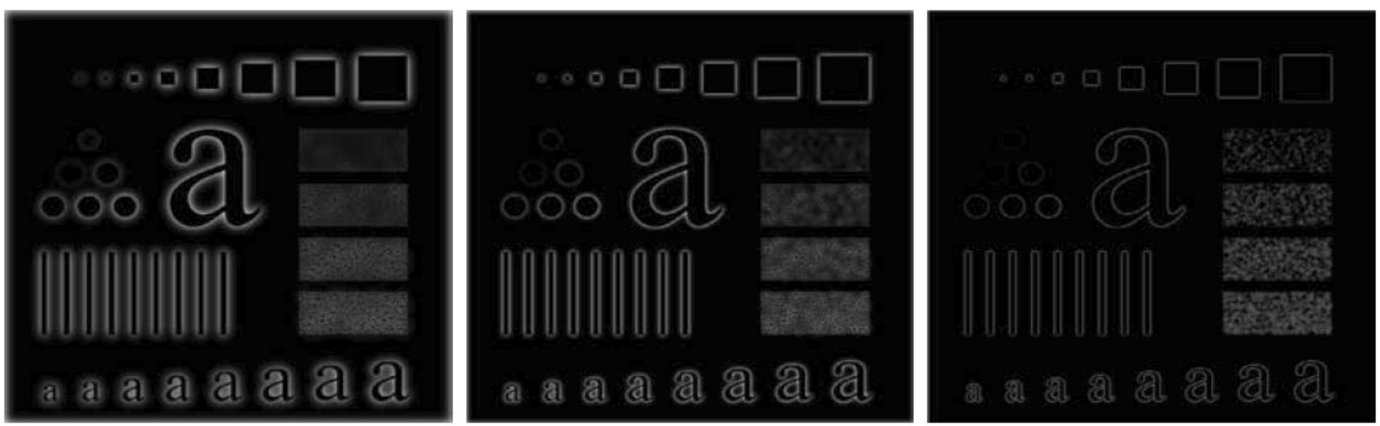
Frequency Filtering: Sharpening

Butterwoth

a b c

FIGURE 4.55 Results of highpass filtering the image in Fig. 4.41(a) using a BHPF of order 2 with $D_0 = 30, 60,$ and 160, corresponding to the circles in Fig. 4.41(b). These results are much smoother than those obtained with an IHPF.

Gaussian



a b c

FIGURE 4.56 Results of highpass filtering the image in Fig. 4.41(a) using a GHPF with $D_0 = 30, 60,$ and 160, corresponding to the circles in Fig. 4.41(b). Compare with Figs. 4.54 and 4.55.

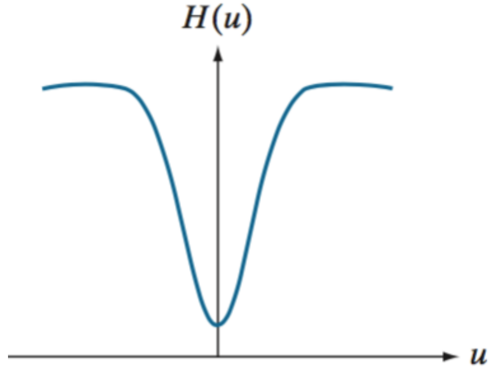
Frequency Filtering: Sharpening

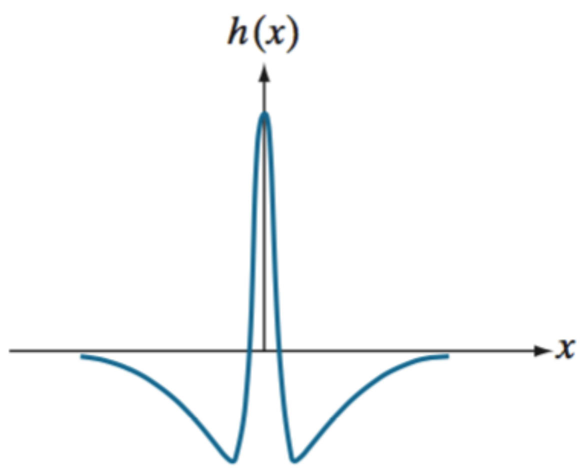
Difference of Gaussian (DoG)

We can construct a high-pass filer by subtracting two Gaussian low-pass filters, and we can get good control of the filter properties:

$$H(u) = Ae^{-u^2/2\sigma_1^2} - Be^{-u^2/2\sigma_2^2} \quad \text{with } A \geq B \text{ and } \sigma_1 > \sigma_2$$

$$h(x) = \sqrt{2\pi}\sigma_1 A e^{-2\pi^2\sigma_1^2 x^2} - \sqrt{2\pi}\sigma_2 B e^{-2\pi^2\sigma_2^2 x^2}$$





Take a break!

Low & High-pass filters in spatial and frequency domains

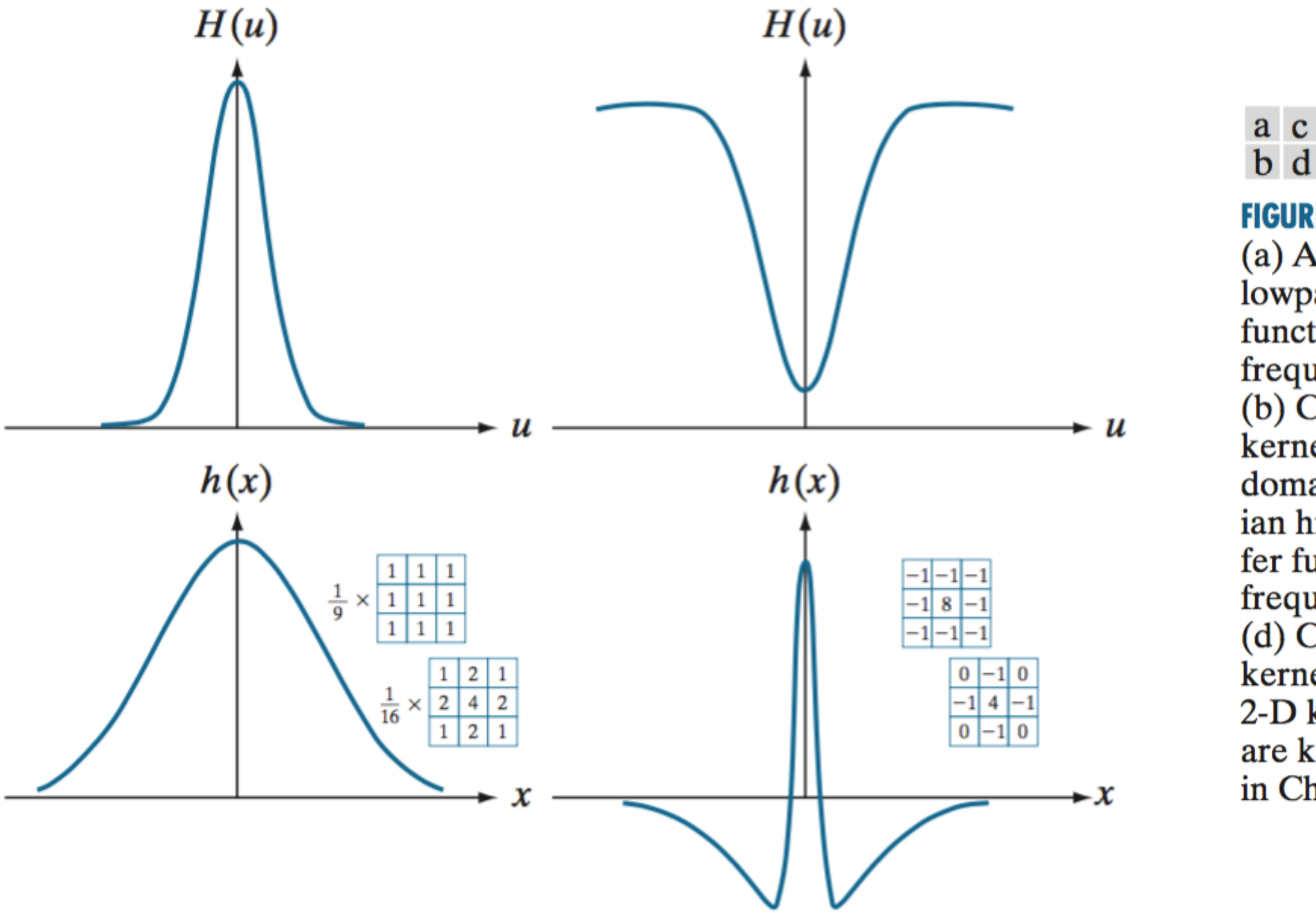
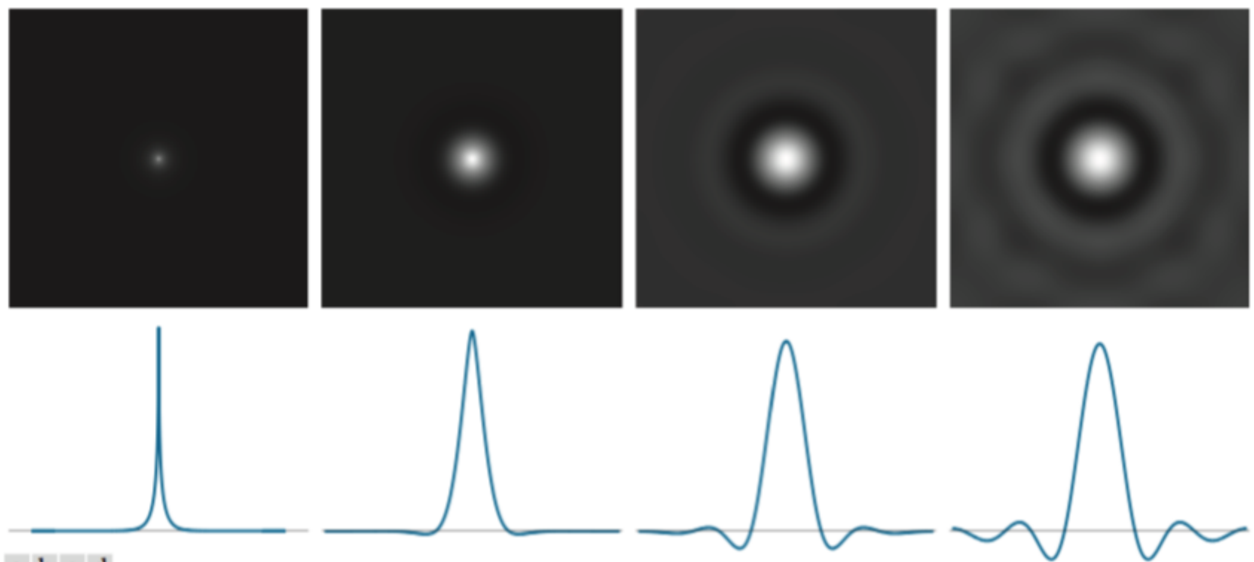


FIGURE 4.36
(a) A 1-D Gaussian lowpass transfer function in the frequency domain. (b) Corresponding kernel in the spatial domain. (c) Gaussian highpass transfer function in the frequency domain. (d) Corresponding kernel. The small 2-D kernels shown are kernels we used in Chapter 3.

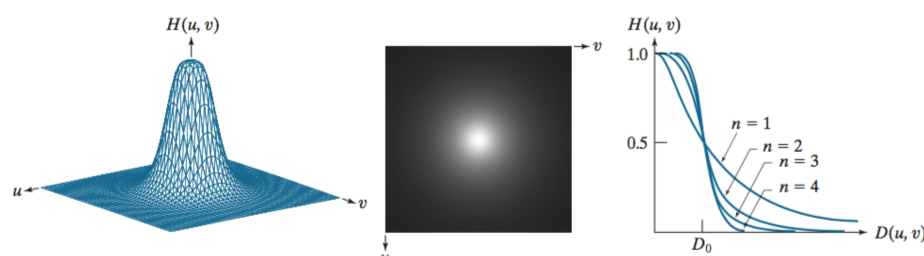
Butterwoth Low pass filters at different orders

n=1 n=2 n=5 n=20



a b c d
e f g h

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



a b c

FIGURE 4.45 (a) Perspective plot of a Butterworth lowpass-filter transfer function. (b) Function displayed as an image. (c) Radial cross sections of BLPFs of orders 1 through 4.

Correspondence Between Filtering in the Spatial and Frequency Domains

The link between filtering in the spatial and frequency domains is the convolution theorem. Filtering in frequency domain is given by: $G(u,v) = H(u,v)F(u,v)$ while filtering in spatial domain is given by the convolution of $h(x,y)$ with $f(x,y)$. Therefore the two filters form a Fourier transform pair:

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

Gaussian filter is of particular interest: both the forward and inverse transforms are real Gaussian functions. This is illustrated for 1D:

$$H(u) = Ae^{-u^2/2\sigma^2} \Leftrightarrow h(x) = \sqrt{2\pi}\sigma \; Ae^{-2\pi^2\sigma^2 x^2}$$

What is a good representation for image analysis?

- Fourier transform domain tells you "what"(textural properties), but not "where".
- Pixel domain representation tells you"where"(pixel location), but not "what".
- Want an image representation that gives you a local description of image events—what is happening where.

Properties of filters in spatial and frequency domains

	Spatial Domain†		Frequency Domain†
1)	$f(x, y)$ real	$\Leftrightarrow$	$F^*(u, v) = F(-u, -v)$
2)	$f(x, y)$ imaginary	$\Leftrightarrow$	$F^*(-u, -v) = -F(u, v)$
3)	$f(x, y)$ real	$\Leftrightarrow$	$R(u, v)$ even; $I(u, v)$ odd
4)	$f(x, y)$ imaginary	$\Leftrightarrow$	$R(u, v)$ odd; $I(u, v)$ even
5)	$f(-x, -y)$ real	$\Leftrightarrow$	$F^*(u, v)$ complex
6)	$f(-x, -y)$ complex	$\Leftrightarrow$	$F(-u, -v)$ complex
7)	$f^*(x, y)$ complex	$\Leftrightarrow$	$F^*(-u - v)$ complex
8)	$f(x, y)$ real and even	$\Leftrightarrow$	$F(u, v)$ real and even
9)	$f(x, y)$ real and odd	$\Leftrightarrow$	$F(u, v)$ imaginary and odd
10)	$f(x, y)$ imaginary and even	$\Leftrightarrow$	$F(u, v)$ imaginary and even
11)	$f(x, y)$ imaginary and odd	$\Leftrightarrow$	$F(u, v)$ real and odd
12)	$f(x, y)$ complex and even	$\Leftrightarrow$	$F(u, v)$ complex and even
13)	$f(x, y)$ complex and odd	$\Leftrightarrow$	$F(u, v)$ complex and odd

The Laplacian in the Frequency Domain

The Laplacian in spatial domain:

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

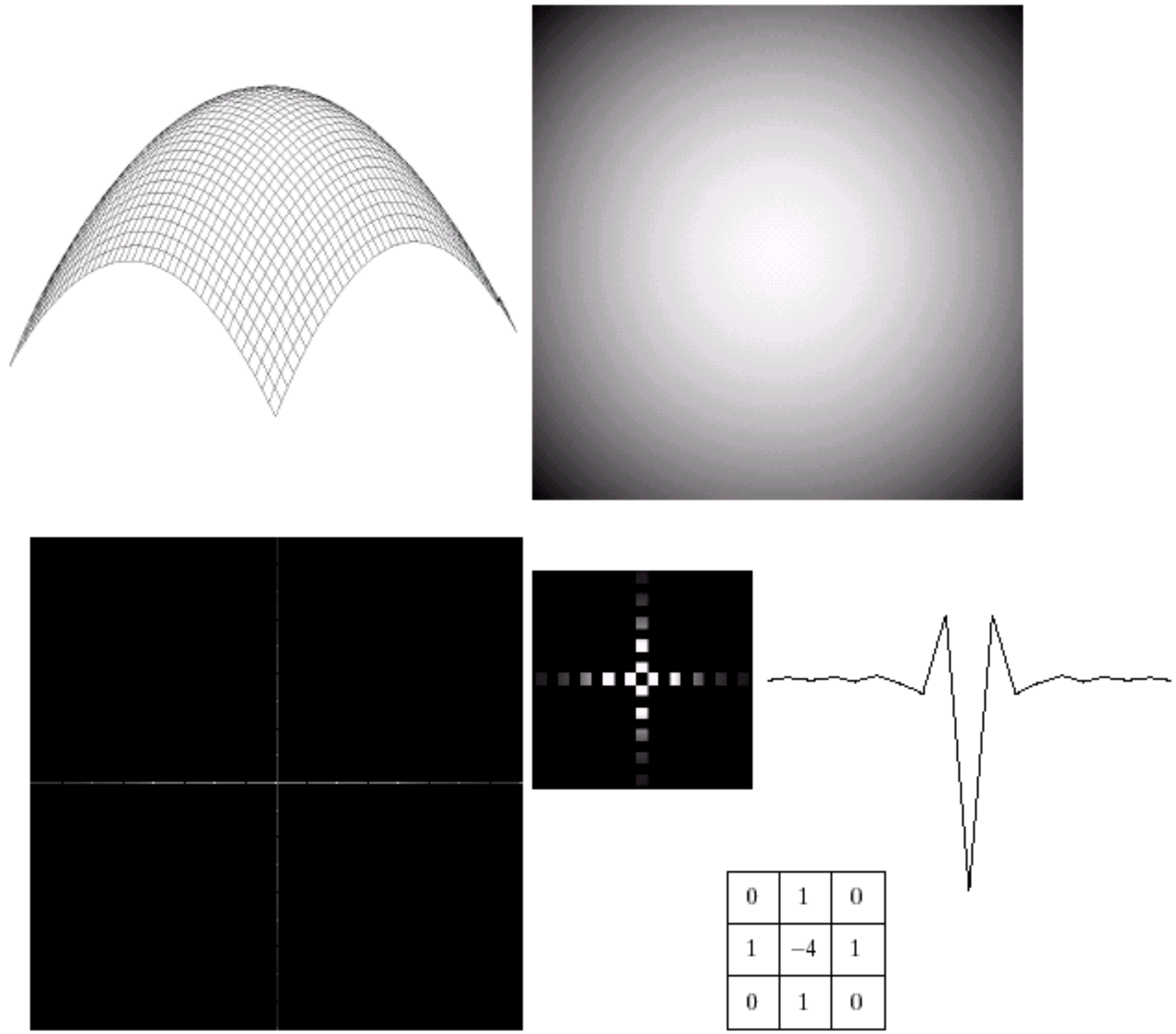
Fourier transform gives:

$$\begin{aligned}
\Im\left(\nabla^2 f(x,y)\right) &= \Im\left(\frac{\partial^2 f(x,y)}{\partial x^2}\right) + \Im\left(\frac{\partial^2 f(x,y)}{\partial y^2}\right) \\
&= (j2\pi\, u)^2 F(u,v) + (j2\pi\, v)^2 F(u,v) \\
&= -4\pi^2(u^2+v^2)F(u,v) \\
&= H(u,v)F(u,v)
\end{aligned}$$

Hence:

$$\nabla^2 f(x,y) \Leftrightarrow -4\pi^2(u^2+v^2)F(u,v)$$

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



a b
c d e
f

FIGURE 4.27 (a) 3-D plot of Laplacian in the frequency domain. (b) Image representation of (a). (c) Laplacian in the spatial domain obtained from the inverse DFT of (b). (d) Zoomed section of the origin of (c). (e) Gray-level profile through the center of (d). (f) Laplacian mask used in Section 3.7.

The Laplacian in the Frequency Domain

With respect to the centre of the frequency rectangle:

$$H(u,v) = -4\pi^2[(u - \boxed{N/2})^2 + (v - \boxed{N/2})^2)]$$

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

The Laplacian image is given by:

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

Enhancement is achieved by:

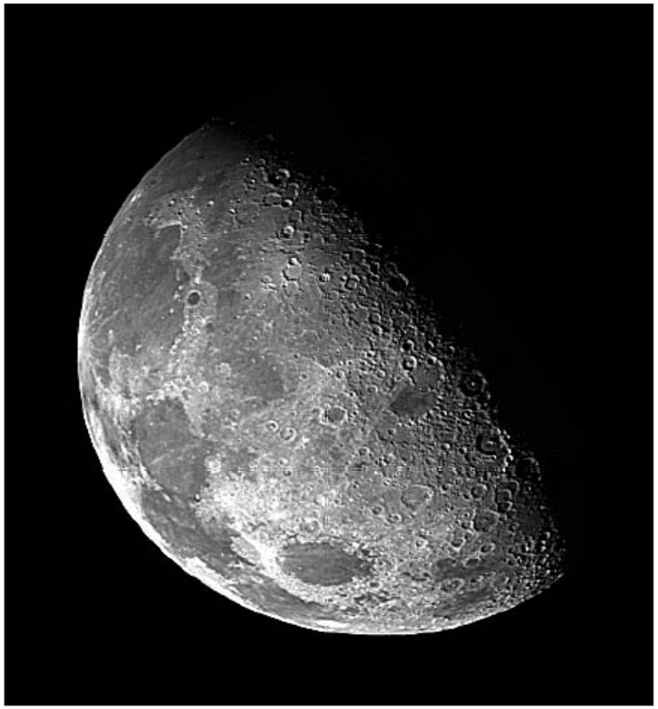
$$\boxed{g(x,y) = f(x,y) + c\nabla^2 f(x,y)}$$

Note: you need to normalize $f(x,y)$ to $[0,1]$ before DFT and divide $\nabla^2 f(x,y)$ by its maximum value.

The Laplacian in the Frequency Domain

In the frequency domain :

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



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).

Corresponding between filtering in spatial and frequency domains

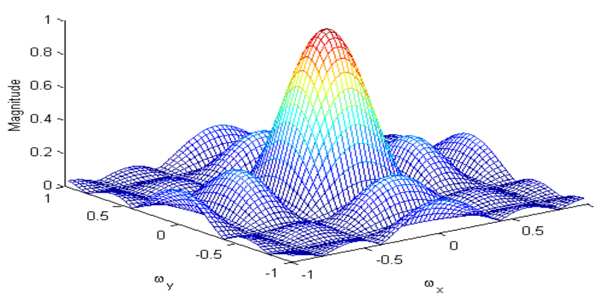
Box filter (smoothing)

$$h(x,y)=\frac{1}{25}\begin{bmatrix}1&1&1&1&1\\1&1&1&1&1\\1&1&[1]&1&1\\1&1&1&1&1\\1&1&1&1&1\end{bmatrix}$$

$$H(u,v)=\sum_{x=-2}^{2}\sum_{y=-2}^{2}h(x,y)e^{-j2\pi(xu+yv)}$$

$$=\frac{1}{25}\sum_{x=-2}^{2}\sum_{y=-2}^{2}e^{-j2\pi(xu+yv)}=\frac{1}{25}\sum_{x=-2}^{2}e^{-j2\pi xu}\sum_{y=-2}^{2}e^{-j2\pi yv}$$

$$=\frac{1}{25}\big(1+2\cos(2\pi u)+2\cos(4\pi u)\big)\big(1+2\cos(2\pi v)+2\cos(4\pi v)\big)$$



Relevant MATLAB function: freqz2

Corresponding between filtering in spatial and frequency domains

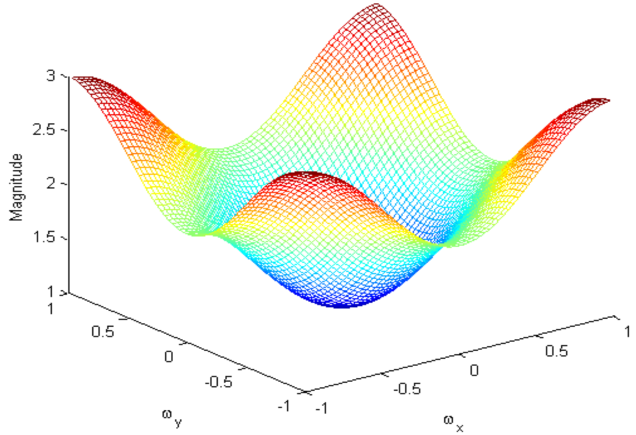
Edge enhancement filter

$$h(x,y) = \frac{1}{4}\begin{bmatrix} 0 & -1 & 0 \\ -1 & 8 & -1 \\ 0 & -1 & 0 \end{bmatrix} \Longrightarrow$$

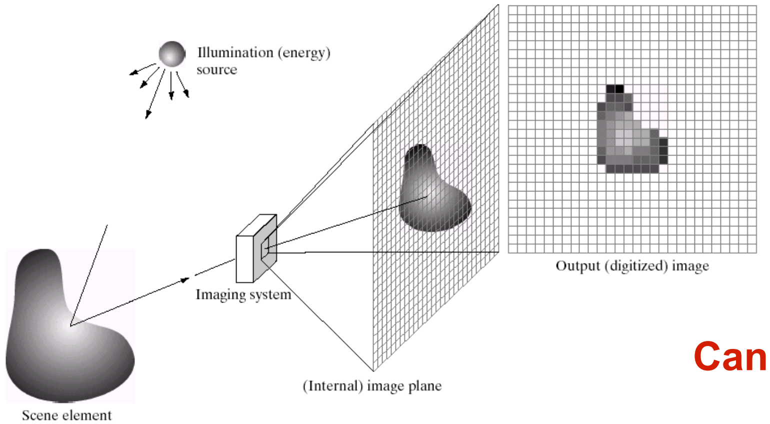
$$H(u,v) = \sum_{x=-1}^{1}\sum_{y=-1}^{1} h(x,y)e^{-j2\pi(xu+yv)}$$

$$= \frac{1}{4}\left(8 - e^{-j2\pi u} - e^{j2\pi u} - e^{-j2\pi v} - e^{j2\pi v}\right)$$

$$= 2 - \frac{1}{2}\cos(2\pi u) - \frac{1}{2}\cos(2\pi v)$$



Homomorphic Filtering



Can be used for multiplicative noise

An image can be expressed as the product of its illumination $i(x, y)$ and reflectance $r(x, y)$: $\boxed{f(x, y) = i(x, y)r(x, y).}$

This equation cannot be transformed directly to the frequency domain. However, we define: $z(x, y) = \ln f(x, y)$, hence:

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

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

Or: $$\boxed{Z(u, v) = F_i(u, v) + F_r(u, v)}$$

Homomorphic Filtering

We can filter $Z(u,v)$ using a filter $H(u,v)$:

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

The filtered image in the spatial domain is:

$$s(x,y) = \Im^{-1}\{S(u,v)\}$$
$$= \Im^{-1}\{H(u,v)F_i(u,v)\} + \Im^{-1}\{H(u,v)F_i(u,v)\}$$
$$\boxed{s(x,y) = i'(x,y) + r'(x,y)}$$

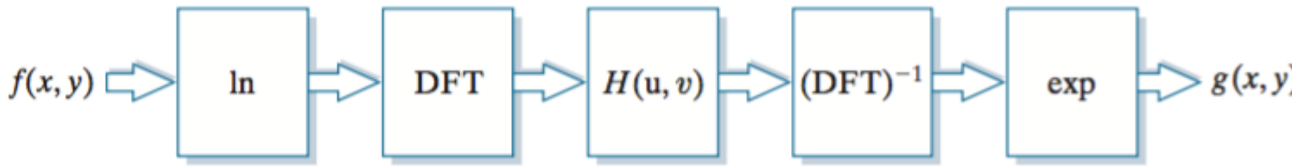
Finally: $g(x,y) = e^{s(x,y)}$

$$= e^{i'(x,y)} . e^{r'(x,y)}$$

It was a natural log

$$= i_0(x,y) . r_0(x,y)$$

FIGURE 4.58 Summary of steps in homomorphic filtering.



- Good controls can be gained over the illumination and reflectance components with a homographic filter.
- Specify H(u,v) that affects the low and high-frequencies differently
- The illumination is the low frequency and reflectance is the high frequency

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

similar to high-frequency-emphasis function

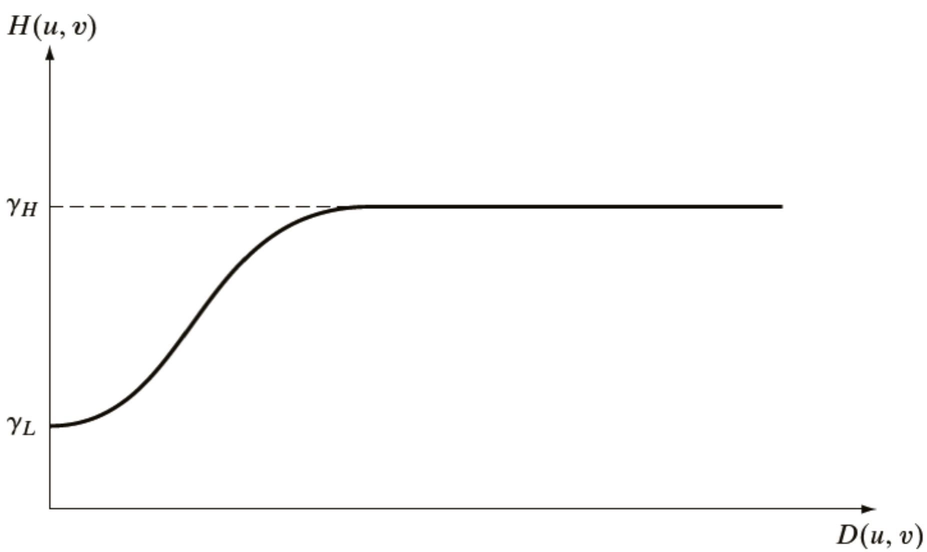
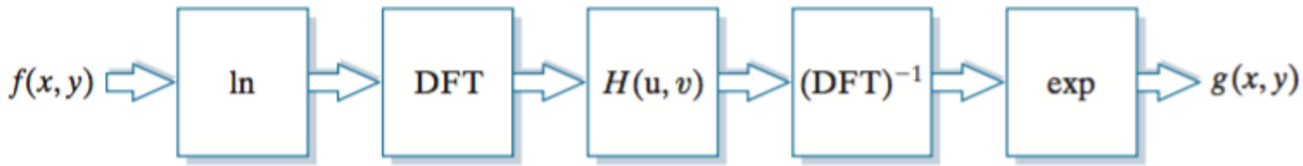
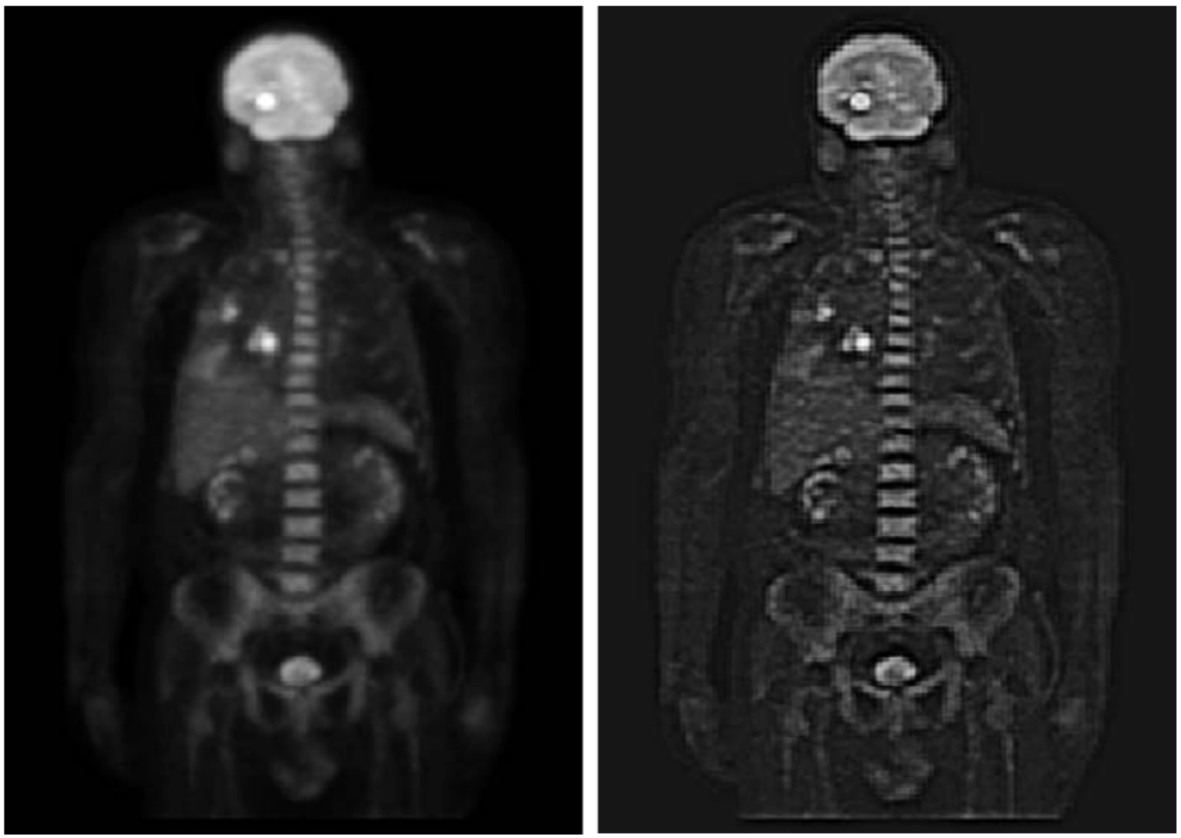


FIGURE 4.61 Radial cross section of a circularly symmetric homomorphic filter function. The vertical axis is at the center of the frequency rectangle and $D(u, v)$ is the distance from the center.

FIGURE 4.58 Summary of steps in homomorphic filtering.





PET: Positron Emission Tomography

a b

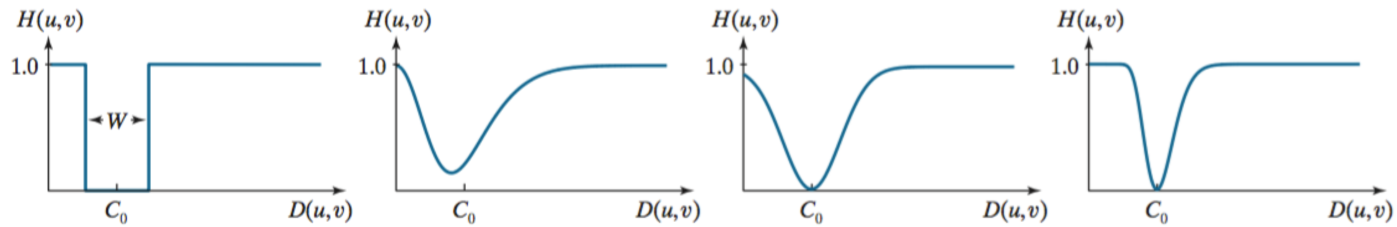
FIGURE 4.62 (a) Full body PET scan. (b) Image enhanced using homomorphic filtering. (Original image courtesy of Dr. Michael E. Casey, CTI PET Systems.)

Band-pass filters

We can use LPFs to form a new filter that only allows information at a frequency band to pass

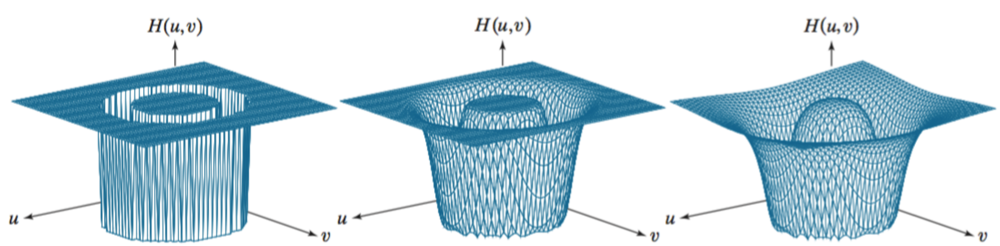
$$H_{\mathrm{BP}}(u,v) = 1 - H_{\mathrm{BR}}(u,v)$$

Band reject functions



a b c d

FIGURE 4.61 Radial cross sections. (a) Ideal bandreject filter transfer function. (b) Bandreject transfer function formed by the sum of Gaussian lowpass and highpass filter functions. (The minimum is not 0 and does not align with C_0.) (c) Radial plot of Eq. (4-149). (The minimum is 0 and is properly aligned with C_0, but the value at the origin is not 1.) (d) Radial plot of Eq. (4-150); this Gaussian-shape plot meets all the requirements of a bandreject filter transfer function.



a b c

FIGURE 4.62 Perspective plots of (a) ideal, (b) modified Gaussian, and (c) modified Butterworth (of order 1) bandreject filter transfer functions from Table 4.7. All transfer functions are of size 512×512 elements, with $C_0 = 128$ and $W = 60$.

Band-pass filters

TABLE 4.7
Bandreject filter transfer functions. C_0 is the center of the band, W is the width of the band, and $D(u,v)$ is the distance from the center of the transfer function to a point (u,v) in the frequency rectangle.

Ideal (IBRF)	Gaussian (GBRF)	Butterworth (BBRF)
$H(u,v) = \begin{cases} 0 & \text{if } C_0 - \frac{W}{2} \le D(u,v) \le C_0 + \frac{W}{2} \\ 1 & \text{otherwise} \end{cases}$	$H(u,v) = 1 - e^{-\left[\frac{D^2(u,v) - C_0^2}{D(u,v)W}\right]^2}$	$H(u,v) = \dfrac{1}{1 + \left[\dfrac{D(u,v)W}{D^2(u,v) - C_0^2}\right]^{2n}}$

a b c

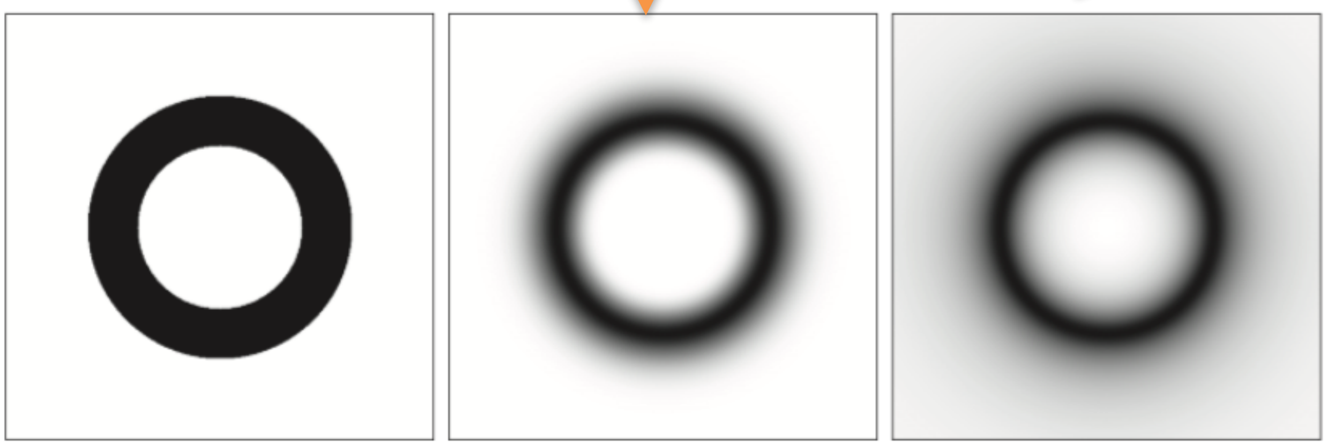


FIGURE 4.63
(a) The ideal, (b) Gaussian, and (c) Butterworth bandpass transfer functions from Fig. 4.62, shown as images. (The thin border lines are not part of the image data.)

Notch filter

- Notch filters are the most useful of the selective filters
- Rejects (or passes) frequencies in a predefined neighbourhood of the frequency rectangle

- We desire that the filters be zero-phase-shift
  - ✓ Must be symmetric about the origin
  - ✓ A notch with center at (u0,v0) must have a corresponding notch at (-u0,-v0)

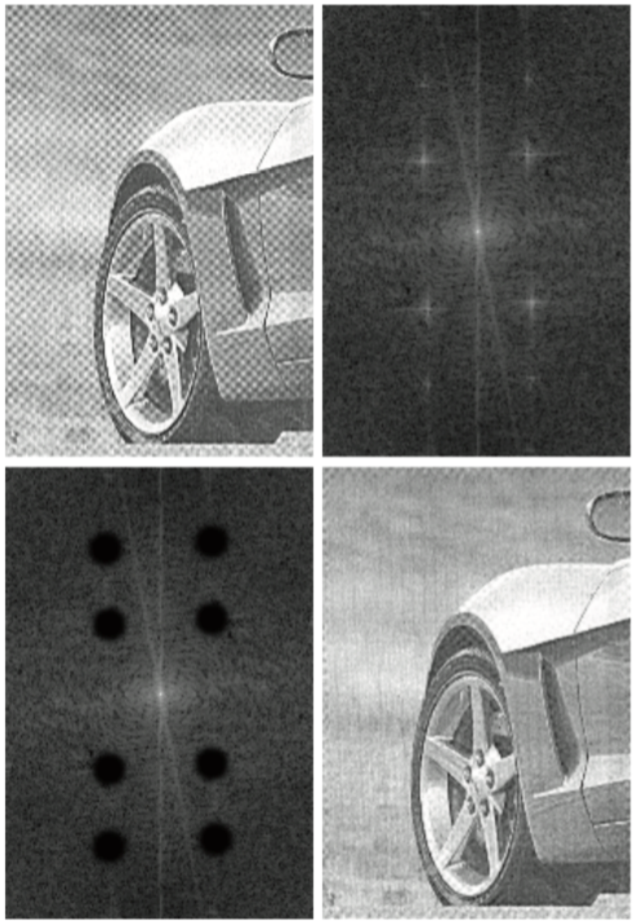
Notch reject filters are formed as $$H_{NR}(u,v) = \prod_{k=1}^{Q} H_k(u,v) H_{-k}(u,v)$$

where $H_k(u,v)$ and $H_{-k}(u,v)$ are highpass filter transfer functions whose centers are at (u_k,v_k) and $(-u_k,-v_k)$, respectively.

Notch filter

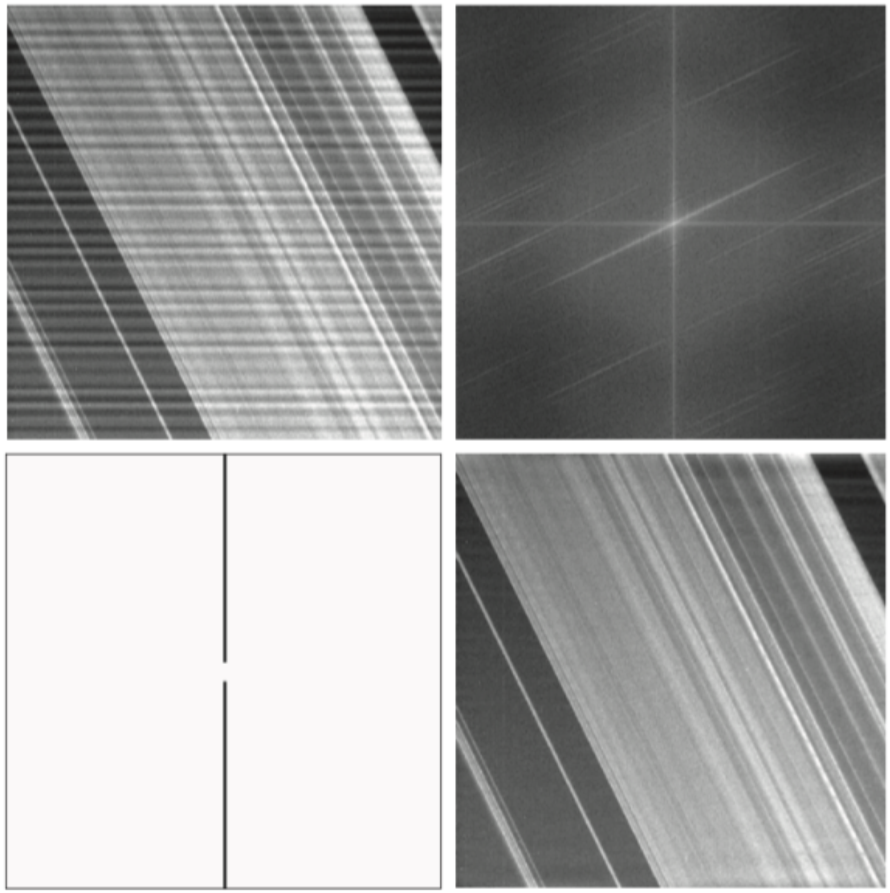
a b
c d

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.



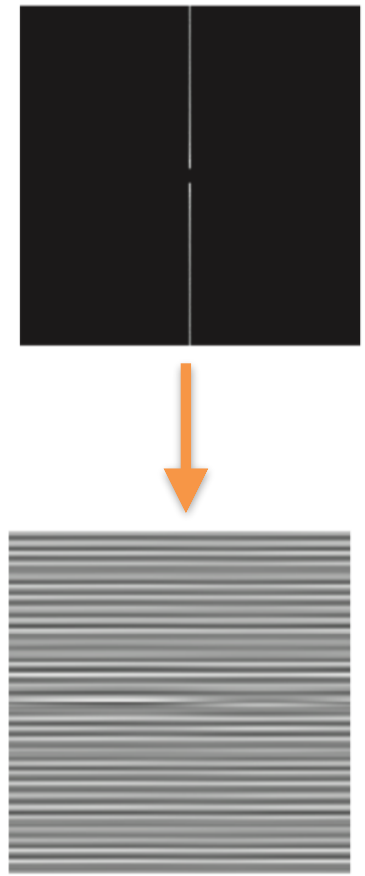
(a) (b)

Notch filter



a b
c d

FIGURE 4.65 (a) Image of 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 transfer function. (d) Result of filtering. (The thin black border in (c) is not part of the data.) (Original image courtesy of Dr. Robert A. West, NASA/ JPL.)



Summary

- Steps for image filtering in frequency domain
- Image smooth and sharpening (ringing artefacts)
- Different filter designs: ideal, butterworth, and Gaussian (low-pass, high-pass, and band-pass filters)
- Correspondence b/w spatial and frequency domains
- Homomorphic filtering
- Notch filtering

Reading materials

- Textbook Chapter 4
  Page 260-303

Questions

- What does a bandpass filter do?
- Why do we want to perform frequency domain filtering?
- Is it always better to perform frequency domain filtering?
- Which filters can cause ringing effects?
- What does a homomorphic filter do?
- What does illumination and reflection components represent in terms of frequency content in an image?
- High-pass filter vs. high-frequency emphasis filter vs. image sharpening
- How do we obtain bandpass filters?
- Notch filter: what does it do? what are the design requirements?