# Ψηφιακή Επεξεργασία Εικόνας (ΨΕΕ) – MYE037

Εαρινό εξάμηνο 2023-2024

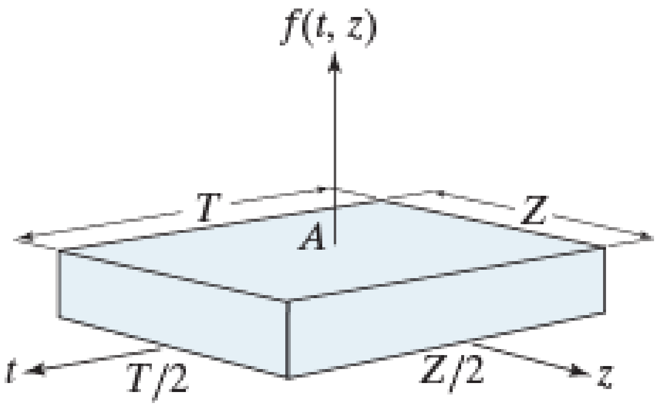
# Filtering in the Frequency Domain (Application)

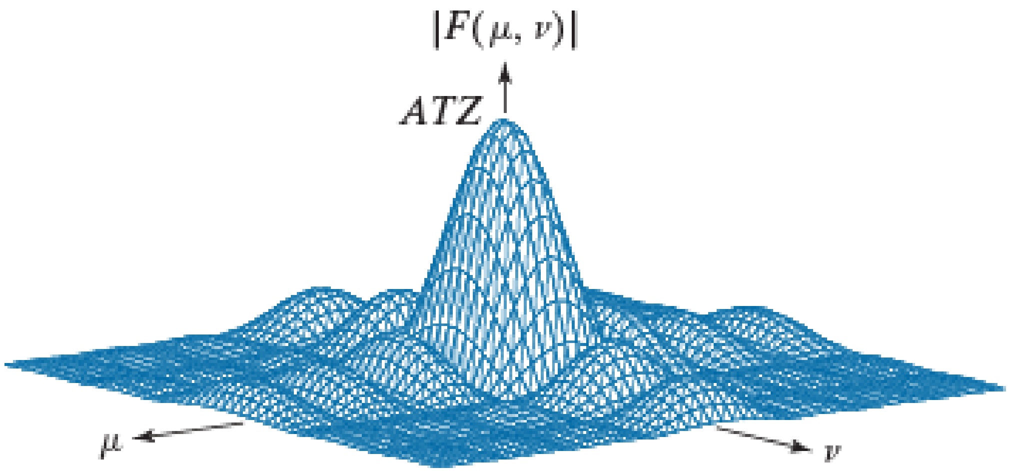
Άγγελος Γιώτης

a.giotis@uoi.gr

# Filtering in the frequency domain (Previously in 2D…)

- We talked about representing a signal in the frequency domain




$$f(x,y) = \mathrm{A}\,P_{W/2,W/2}(x,y) \leftrightarrow F(\mu,\nu) = \mathrm{A}W^2 \frac{\sin(\pi\mu W)}{(\pi\mu W)} \frac{\sin(\pi\nu W)}{(\pi\nu W)}$$

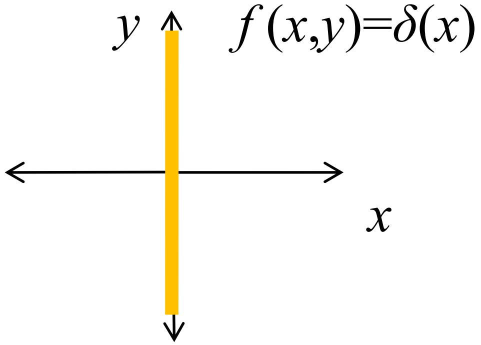
# Filtering in the frequency domain (Previously in 2D…)

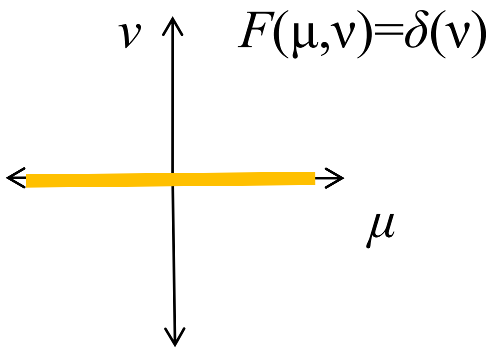
- Example: FT of $f(x,y)=\delta(x)$

$$F(\mu,\nu)=\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}\delta(x)e^{-j2\pi(\mu x+\nu y)}dydx$$

$$=\int_{-\infty}^{+\infty}\delta(x)e^{-j2\pi\mu x}dx\int_{-\infty}^{+\infty}e^{-j2\pi\nu y}dy$$

$$=\int_{-\infty}^{+\infty}e^{-j2\pi\nu y}dy \quad =\delta(\nu)$$



# Filtering in the frequency domain (Previously in 2D…)

- Reminder

$$\Im\{\delta(t)\} = F(\mu) = \int_{-\infty}^{\infty} \delta(t) e^{-j2\pi\mu t} dt = \int_{-\infty}^{\infty} e^{-j2\pi\mu t} \delta(t)\, dt = e^{-j2\pi\mu 0}$$

$$\int_{-\infty}^{\infty} f(t)\delta(t)\, dt = f(0)$$

- Hence, the term that follows is ‘1’.

$$\int_{-\infty}^{+\infty} \delta(x) e^{-j2\pi\mu x} dx$$

# Filtering in the frequency domain (Previously in 2D…)

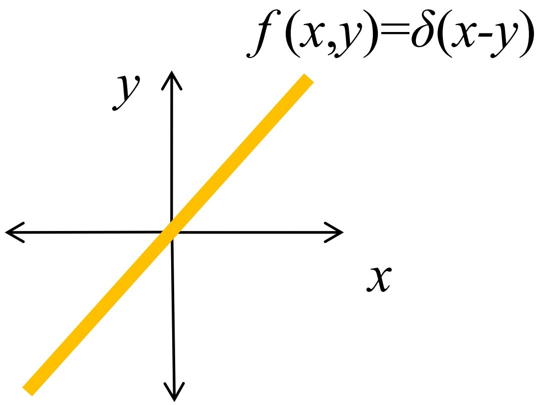
- Example: FT of $f(x,y)=\delta(x-y)$

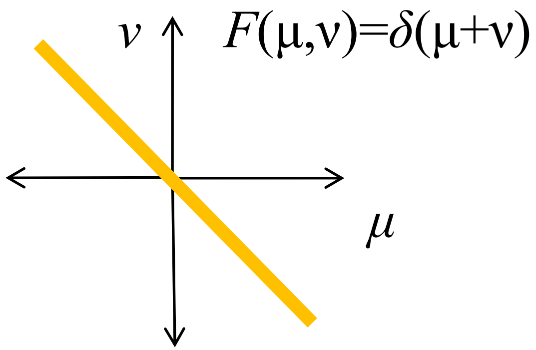
$$F(\mu,\nu)=\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}\delta(x-y)e^{-j2\pi(\mu x+\nu y)}dydx$$

$$=\int_{-\infty}^{+\infty}\left[\int_{-\infty}^{+\infty}\delta(x-y)e^{-j2\pi\mu x}dx\right]e^{-j2\pi\nu y}dy$$

$$=\int_{-\infty}^{+\infty}e^{-j2\pi\mu y}e^{-j2\pi\nu y}dy=\int_{-\infty}^{+\infty}e^{-j2\pi(\mu+\nu)y}dy$$

$$=\delta(\mu+\nu)$$



# Filtering in the frequency domain (Previously in 2D…)

- Reminder

$$\Im\{\delta(t-t_0)\} = F(\mu) = \int_{-\infty}^{\infty} \delta(t-t_0)e^{-j2\pi\mu t}dt = \int_{-\infty}^{\infty} e^{-j2\pi\mu t}\,\delta(t-t_0)dt = e^{-j2\pi\mu t_0}$$

$$\int_{-\infty}^{\infty} f(t)\delta(t-t_0)dt = f(t_0)$$

- Hence, the term that follows is:

$$\left[\int_{-\infty}^{+\infty} \delta(x-y)e^{-j2\pi\mu x}dx\right] = e^{-j2\pi\mu y}$$

# Filtering in the frequency domain (Previously in 2D…)

- 2D continuous convolution

$$f(x,y)*h(x,y)=\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f(x-\alpha,y-\beta)h(\alpha,\beta)\,d\alpha\, d\beta$$

- We will examine the discrete convolution in more detail.
- Convolution property

$$f(x,y)*h(x,y) \leftrightarrow F(\mu,\nu)H(\mu,\nu)$$

# Filtering in the frequency domain (Previously in 2D…)

- 2D sampling is accomplished by

$$\boldsymbol{s}_{\Delta X \Delta Y}(x,y) = \boldsymbol{s}_{\Delta X}(x)\boldsymbol{s}_{\Delta Y}(y) = \sum_{m=-\infty}^{\infty}\sum_{n=-\infty}^{\infty} \delta(x - m\Delta X, y - nDY)$$

- The FT of the sampled 2D signal consists of repetitions of the spectrum of the 1D continuous signal.

$$\tilde{F}(\mu,\nu) = \frac{1}{\Delta X}\frac{1}{\Delta Y}\sum_{m=-\infty}^{\infty}\sum_{n=-\infty}^{\infty} \mathrm{F}(\mu - \frac{m}{\Delta X}, \nu - \frac{n}{\Delta X})$$

# 2D Discrete Fourier Transform (2D DFT)

- 2D DFT pair of image $f[m,n]$ of size $M$x$N$.

$$F[k,l]=\sum_{m=0}^{M-1}\sum_{n=0}^{N-1} f[m,n]e^{-j2\pi\left(\frac{km}{M}+\frac{ln}{N}\right)}$$

$$f[m,n]=\frac{1}{MN}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1} F[k,l]e^{j2\pi\left(\frac{km}{M}+\frac{ln}{N}\right)}$$

$$\begin{cases} 0\le k\le M-1 \\ 0\le l\le N-1 \end{cases}, \quad \begin{cases} 0\le m\le M-1 \\ 0\le n\le N-1 \end{cases}$$

# 2D Discrete Fourier Transform (2D DFT)

Separability of the 2D DFT:

- We can express the 2D DFT as two 1D DFTs:
- First, perform a 1D DFT along the columns and then along the rows (or vice versa).

# From 1D to 2D DFT in matrix form

2D DFT can be represented in matrix form:

- Reminder for 1D: $\mathbf{F} = \mathbf{A}\mathbf{f}$ $\quad w_N^{nk} = e^{-j\frac{2\pi kn}{N}}$

$$\mathbf{A} = \begin{bmatrix} \left(w_N^0\right)^0 & \left(w_N^0\right)^1 & \left(w_N^0\right)^2 & \dots & \left(w_N^0\right)^{N-1} \\ \left(w_N^1\right)^0 & \left(w_N^1\right)^1 & \left(w_N^1\right)^2 & \dots & \left(w_N^1\right)^{N-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \left(w_N^{N-1}\right)^0 & \left(w_N^{N-1}\right)^1 & \left(w_N^{N-1}\right)^2 & \dots & \left(w_N^{N-1}\right)^{N-1} \end{bmatrix}$$

# From 1D to 2D DFT in matrix form

2D DFT can be represented in matrix form:

- Reminder 1D: $\mathbf{F} = \mathbf{A}\mathbf{f}$ $\quad w_N^{nk} = e^{-j\frac{2\pi kn}{N}}$

  $\mathbf{f} = \mathbf{A}^{-1}\mathbf{F}$

$$\mathbf{A}^{-1} = \frac{1}{N}\left(\mathbf{A}^*\right)^T = \frac{1}{N}\left(\begin{bmatrix} \left(w_N^0\right)^0 & \left(w_N^0\right)^1 & \left(w_N^0\right)^2 & \cdots & \left(w_N^0\right)^{N-1} \\ \left(w_N^1\right)^0 & \left(w_N^1\right)^1 & \left(w_N^1\right)^2 & \cdots & \left(w_N^1\right)^{N-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \left(w_N^{N-1}\right)^0 & \left(w_N^{N-1}\right)^1 & \left(w_N^{N-1}\right)^2 & \cdots & \left(w_N^{N-1}\right)^{N-1} \end{bmatrix}^*\right)^T$$

# 2D Discrete Fourier Transform (2D DFT)

2D DFT can be represented in matrix form:

- In a similar fashion for 2D we employ the same matrix A:

$$F = \mathrm{A} f \mathrm{A}^{\mathrm{T}}$$

- Where now *F*, *f* are now NxN matrices:
  - Equivalent: $F = AfA$, since $A = A^T$

# From 1D to 2D DFT in matrix form

- General case: for NxM εικόνες/signals we have:

$$F = \mathrm{A_N} f \mathrm{A_M}^{\mathrm{T}}$$

- Where now *F, f* are now NxM matrices:
  - Equivalent: $F = A_N f A_M$, since $A = A^T$
  - caution: when N ≠ M we multiply with different matrix on the left and right, respectively.

# DFT symmetric properties

- From functional analysis any real or complex function, w(x, y), can be expressed as:

$$w(x,y) = w_e(x,y) + w_o(x,y)$$

- Where the even and odd parts are defined as:

$$w_e(x,y) \triangleq \frac{w(x,y) + w(-x,-y)}{2} \qquad w_o(x,y) \triangleq \frac{w(x,y) - w(-x,-y)}{2}$$

- Substituting gives the identity w(x, y) ≡ w(x, y) and then:

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

- *Even* functions are said to be *symmetric* and *odd* functions *antisymmetric*.

# DFT symmetry property

- Symmetry Property – FT of real function *f(x,y)* is conjugate symmetric:

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

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

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

$$= F(-u,-v)$$

- For which we can infer: $F(u,v) = | F(-u,-v) |$

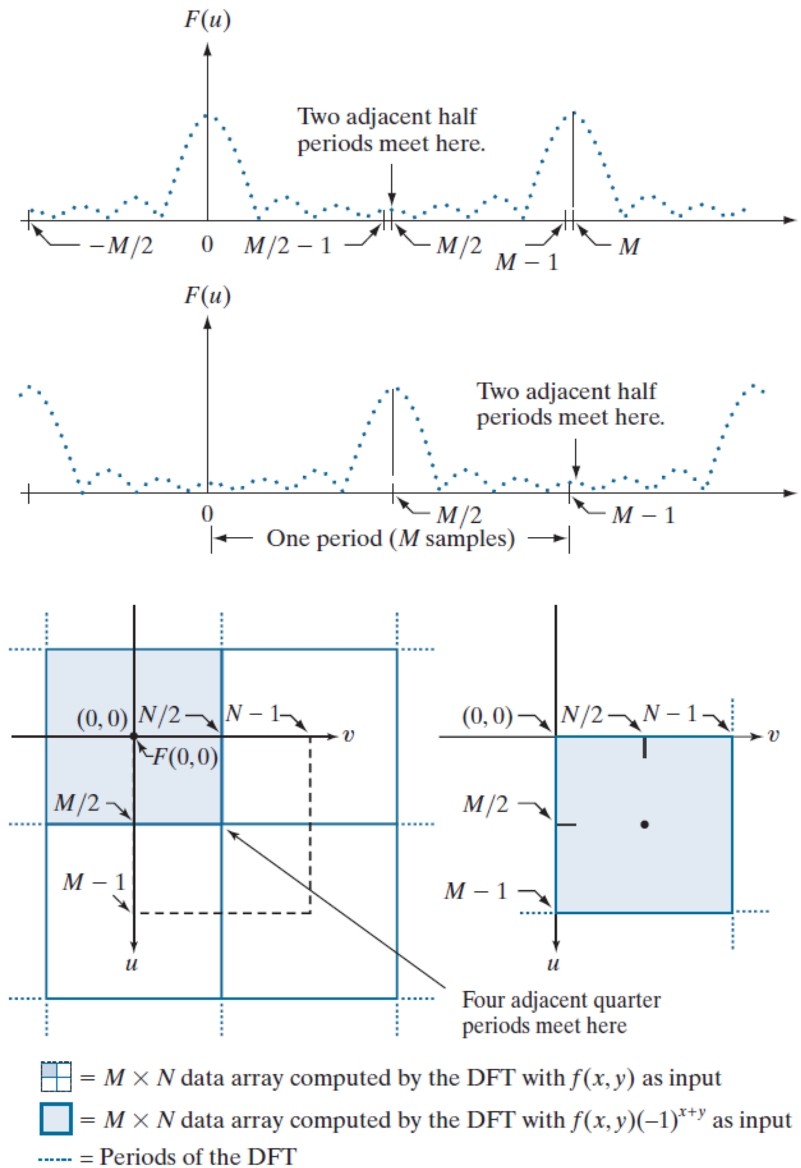
# DFT symmetry property

- Similarly, FT of conjugate function f is conjugate anti-symmetric:

$$F^{*}(-u,-v) = -F(u,v)$$

# Periodicity of the DFT

- The range of frequencies of the signal is between [*-M/2, M/2*].
- The DFT covers two back-to-back half periods of the signal as it covers [*0, M-1*].
- For display and computation purposes it is convenient to shift the DFT and have a complete period in [*0, M-1*].




(b) Shifted DFT obtained by multiplying $f(x)$ by $(-1)^x$ before computing $F(u)$

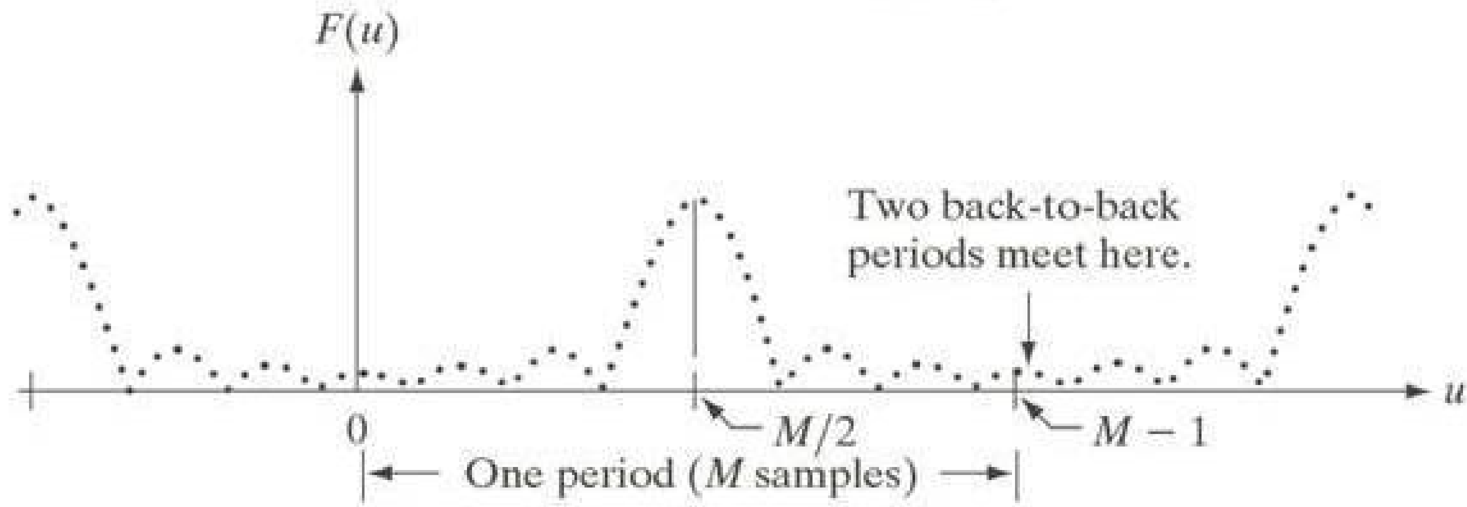
(d) Shifted DFT obtained by multiplying $f(x,y)$ by $(-1)^{x+y}$ before computing $F(u,v)$

# Periodicity of the DFT (cont...)

- From DFT properties: $f[n]e^{j2\pi(N_0 n/M)} \Leftrightarrow F(k-N_0)$

Letting $N_0=M/2$: $f[n](-1)^n \Leftrightarrow F(k-M/2)$

And $F(0)$ is now located at $M/2$.




- Similarly in 2D we shift F(0,0) to (M/2,N/2) using: $f(x,y)(-1)^{x+y}$

# DFT properties (synopsis)

**TABLE 4.3**
Summary of DFT definitions and corresponding expressions.

| Name | Expression(s) |
|---|---|
| 1) Discrete Fourier transform (DFT) of $f(x,y)$ | $F(u,v) = \sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)e^{-j2\pi(ux/M+vy/N)}$ |
| 2) Inverse discrete Fourier transform (IDFT) of $F(u,v)$ | $f(x,y) = \frac{1}{MN}\sum_{u=0}^{M-1}\sum_{v=0}^{N-1} F(u,v)e^{j2\pi(ux/M+vy/N)}$ |
| 3) Spectrum | $\lvert F(u,v)\rvert = \left[R^2(u,v) + I^2(u,v)\right]^{1/2}$ $\quad R = \text{Real}(F);\ I = \text{Imag}(F)$ |
| 4) Phase angle | $\phi(u,v) = \tan^{-1}\left[\frac{I(u,v)}{R(u,v)}\right]$ |
| 5) Polar representation | $F(u,v) = \lvert F(u,v)\rvert e^{j\phi(u,v)}$ |
| 6) Power spectrum | $P(u,v) = \lvert F(u,v)\rvert^2$ |
| 7) Average value | $\bar{f} = \frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y) = \frac{1}{MN}F(0,0)$ |
| 8) Periodicity ($k_1$ and $k_2$ are integers) | $F(u,v) = F(u+k_1M,v) = F(u,v+k_2N)$ $= F(u+k_1,v+k_2N)$ <br> $f(x,y) = f(x+k_1M,y) = f(x,y+k_2N)$ $= f(x+k_1M,y+k_2N)$ |
| 9) Convolution | $(f \bigstar h)(x,y) = \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} f(m,n)h(x-m,y-n)$ |
| 10) Correlation | $(f \star h)(x,y) = \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} f^*(m,n)h(x+m,y+n)$ |
| 11) Separability | The 2-D DFT can be computed by computing 1-D DFT transforms along the rows (columns) of the image, followed by 1-D transforms along the columns (rows) of the result. See Section 4.11. |
| 12) Obtaining the IDFT using a DFT algorithm | $MNf^*(x,y) = \sum_{u=0}^{M-1}\sum_{v=0}^{N-1} F^*(u,v)e^{-j2\pi(ux/M+vy/N)}$ <br> This equation indicates that inputting $F^*(u,v)$ into an algorithm that computes the forward transform (right side of above equation) yields $MNf^*(x,y)$. Taking the complex conjugate and dividing by $MN$ gives the desired inverse. See |

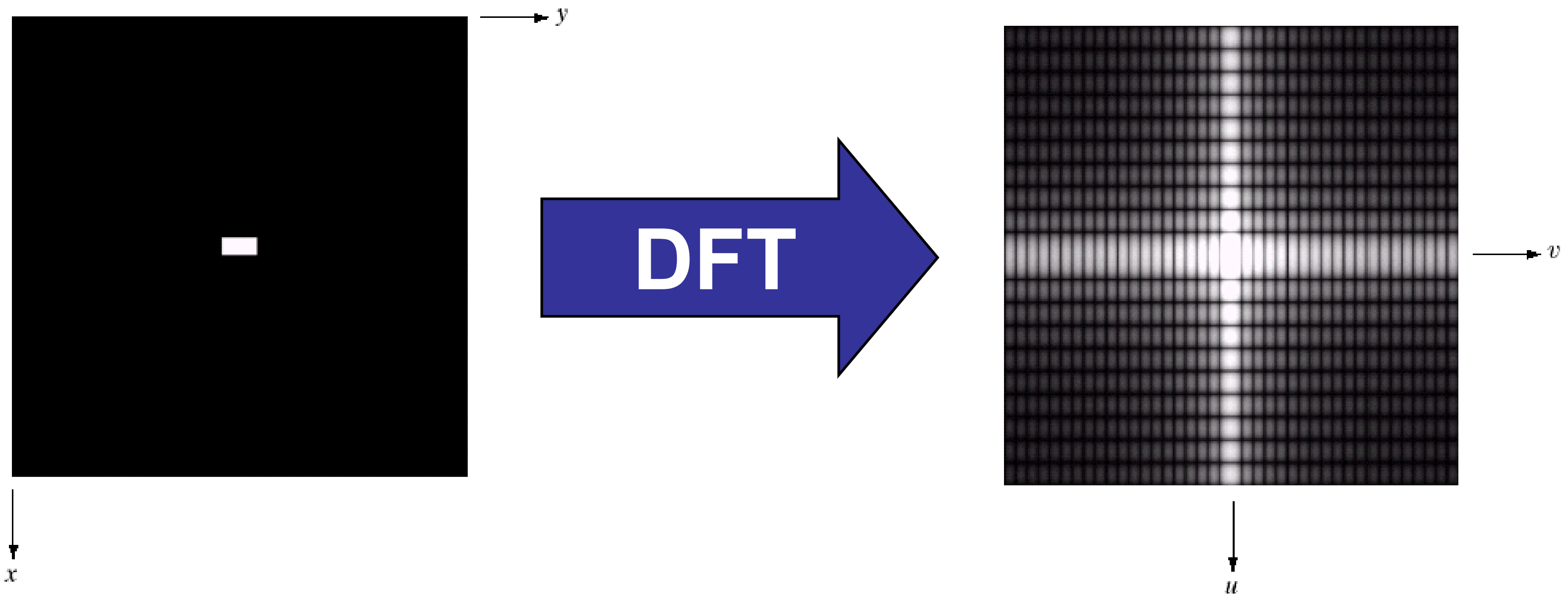
# DFT properties (synopsis, cont..)

**TABLE 4.4** Summmary of DFT pairs. The closed-form expressions in 12 and 13 are valid only for continuous variables. They can be used with discrete variables by sampling the continuous expressions.

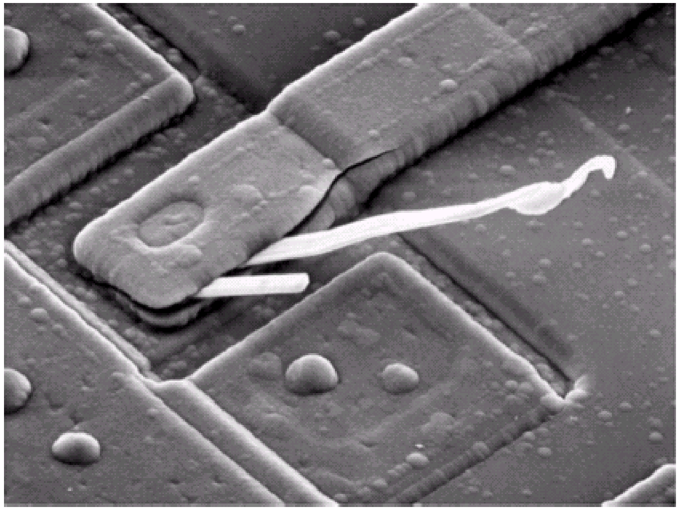
| | Name | DFT Pairs |
|---|---|---|
| 1) | Symmetry properties | See Table 4.1 |
| 2) | Linearity | $a f_1(x,y) + b f_2(x,y) \Leftrightarrow aF_1(u,v) + bF_2(u,v)$ |
| 3) | Translation (general) | $f(x,y)e^{j2\pi(u_0x/M + v_0y/N)} \Leftrightarrow F(u-u_0, v-v_0)$<br>$f(x-x_0, y-y_0) \Leftrightarrow F(u,v)e^{-j2\pi(ux_0/M + vy_0/N)}$ |
| 4) | Translation to center of the frequency rectangle, $(M/2, N/2)$ | $f(x,y)(-1)^{x+y} \Leftrightarrow F(u-M/2, v-N/2)$<br>$f(x-M/2, y-N/2) \Leftrightarrow F(u,v)(-1)^{u+v}$ |
| 5) | Rotation | $f(r,\theta+\theta_0) \Leftrightarrow F(\omega,\varphi+\theta_0)$ $\quad x = r\cos\theta \quad y = r\sin\theta \quad u = \omega\cos\varphi \quad v = \omega\sin\varphi$<br>$r = \sqrt{x^2+y^2} \quad \theta = \tan^{-1}(y/x) \quad \omega = \sqrt{u^2+v^2} \quad \varphi = \tan^{-1}(v/u)$ |
| 6) | Convolution theorem† | $f \star h)(x,y) \Leftrightarrow (F \bullet H)(u,v)$<br>$(f \bullet h)(x,y) \Leftrightarrow (1/MN)[(F \star H)(u,v)]$ |
| 7) | Correlation theorem† | $(f \mathbin{☆} h)(x,y) \Leftrightarrow (F^* \bullet H)(u,v)$<br>$(f^* \bullet h)(x,y) \Leftrightarrow (1/MN)[(F \mathbin{☆} H)(u,v)]$ |
| 8) | Discrete unit impulse | $\delta(x,y) \Leftrightarrow 1$<br>$1 \Leftrightarrow MN\delta(u,v)$ |
| 9) | Rectangle | $\text{rec}[a,b] \Leftrightarrow ab\dfrac{\sin(\pi ua)}{(\pi ua)}\dfrac{\sin(\pi vb)}{(\pi vb)}e^{-j\pi(ua+vb)}$ |
| 10) | Sine | $\sin(2\pi u_0x/M + 2\pi v_0y/N) \Leftrightarrow \dfrac{jMN}{2}[\delta(u+u_0, v+v_0) - \delta(u-u_0, v-v_0)]$ |
| 11) | Cosine | $\cos(2\pi u_0x/M + 2\pi v_0y/N) \Leftrightarrow \dfrac{1}{2}[\delta(u+u_0, v+v_0) + \delta(u-u_0, v-v_0)]$ |
| | The following Fourier transform pairs are derivable only for continuous variables, denoted as before by $t$ and $z$ for spatial variables and by $\mu$ and $\nu$ for frequency variables. These results can be used for DFT work by sampling the continuous forms. | |
| 12) | Differentiation (the expressions on the right assume that $f(\pm\infty, \pm\infty) = 0$. | $\left(\dfrac{\partial}{\partial t}\right)^m \left(\dfrac{\partial}{\partial z}\right)^n f(t,z) \Leftrightarrow (j2\pi\mu)^m (j2\pi\nu)^n F(\mu,\nu)$<br>$\dfrac{\partial^m f(t,z)}{\partial t^m} \Leftrightarrow (j2\pi\mu)^m F(\mu,\nu); \quad \dfrac{\partial^n f(t,z)}{\partial z^m} \Leftrightarrow (j2\pi\nu)^n F(\mu,\nu)$ |
| 13) | Gaussian | $A2\pi\sigma^2 e^{-2\pi^2\sigma^2(t^2+z^2)} \Leftrightarrow Ae^{-(\mu^2+\nu^2)/2\sigma^2}$ ($A$ is a constant) |

# DFT & Images

The DFT of a two dimensional image can be visualised by showing the spectrum of the image component frequencies

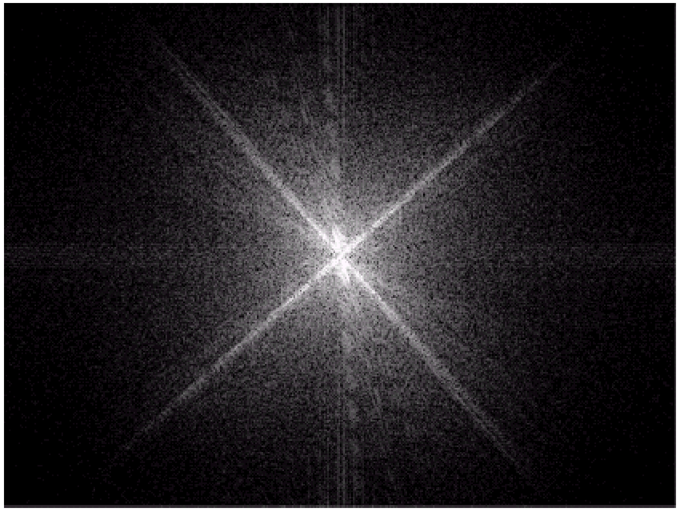


# DFT & Images (cont…)



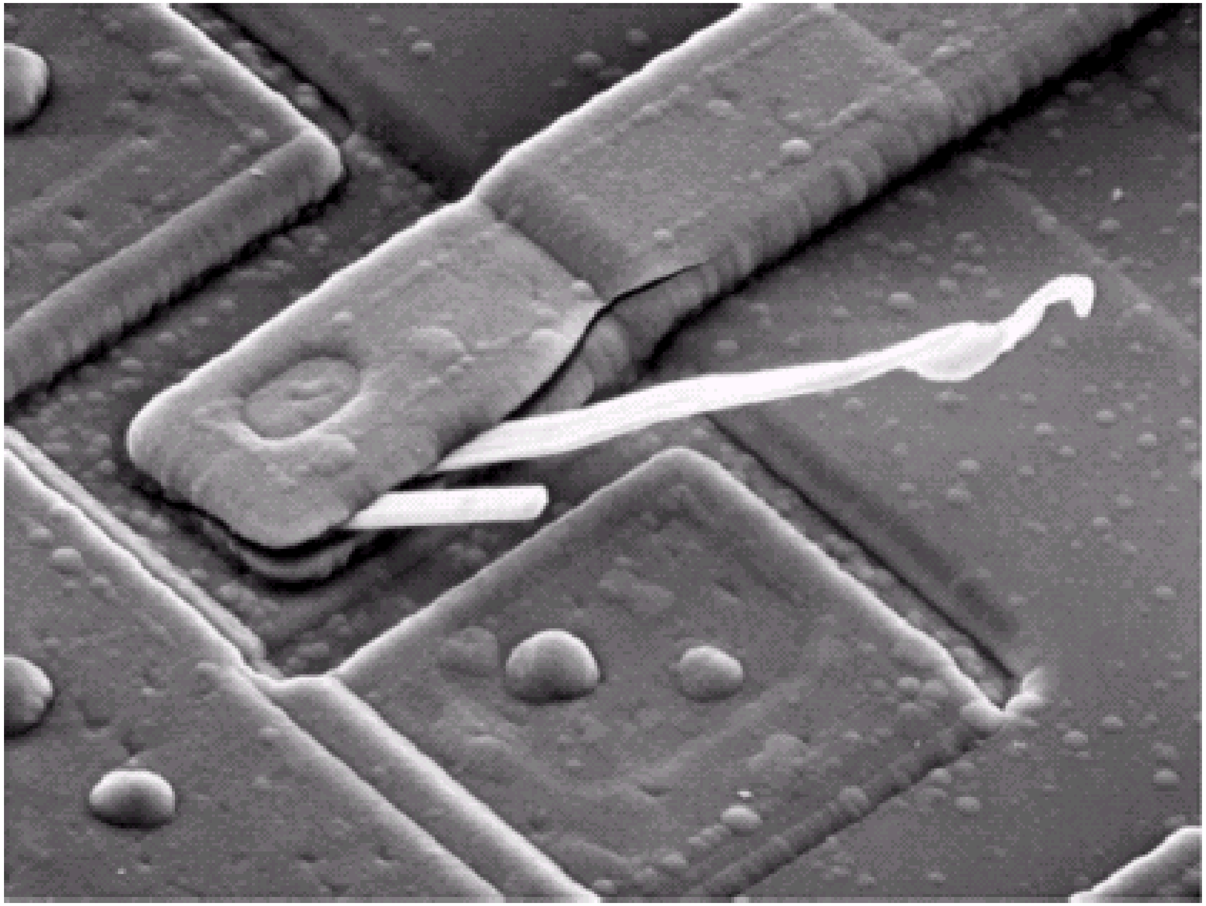
Scanning electron microscope image of an integrated circuit magnified ~2500 times

DFT

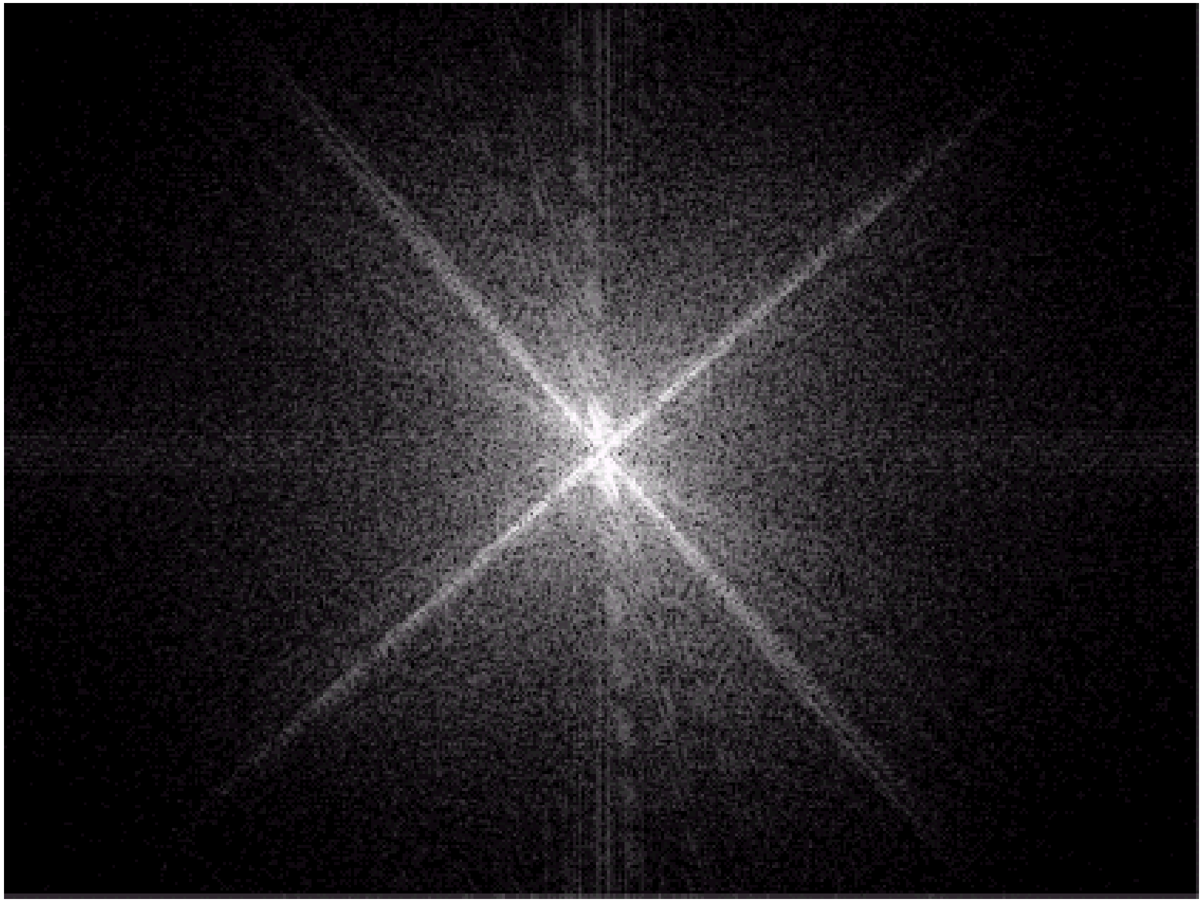


Fourier spectrum of the image

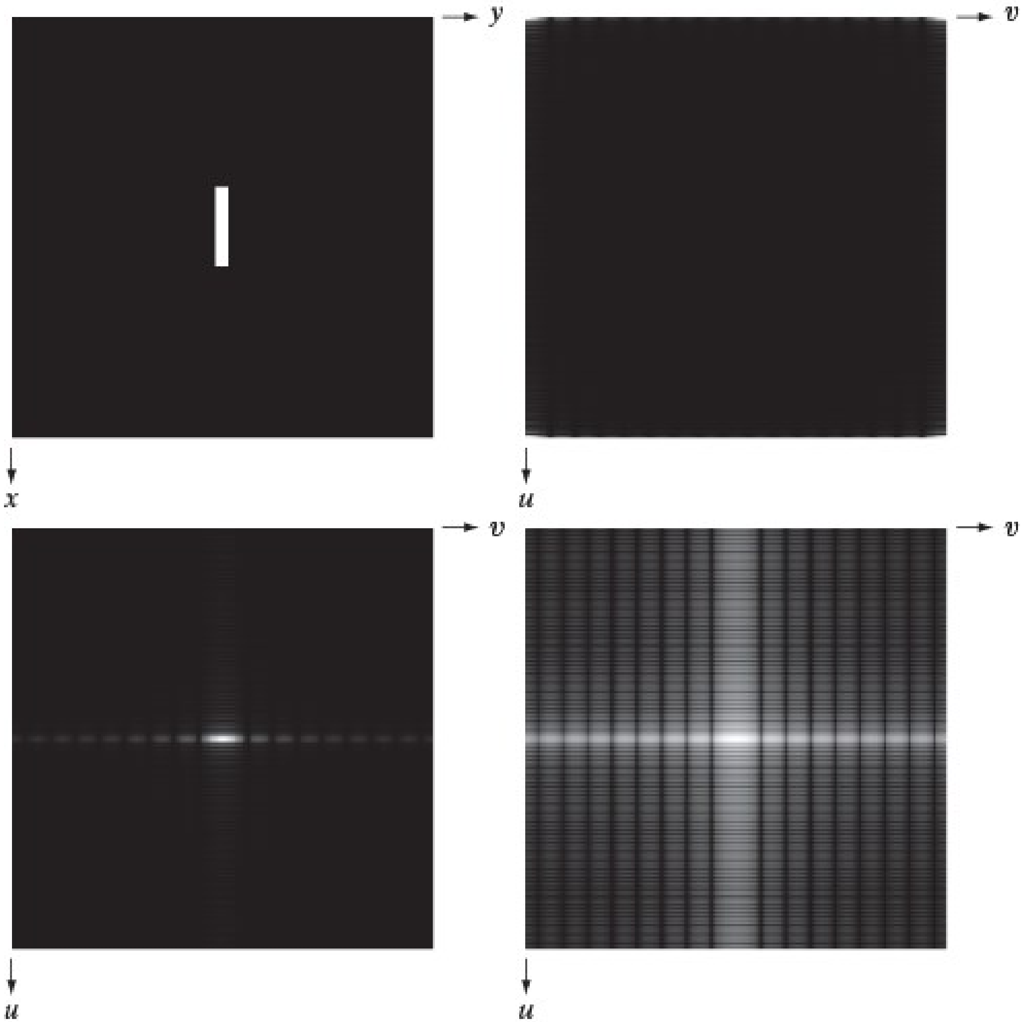
# DFT & Images (cont…)



# DFT & Images (cont…)



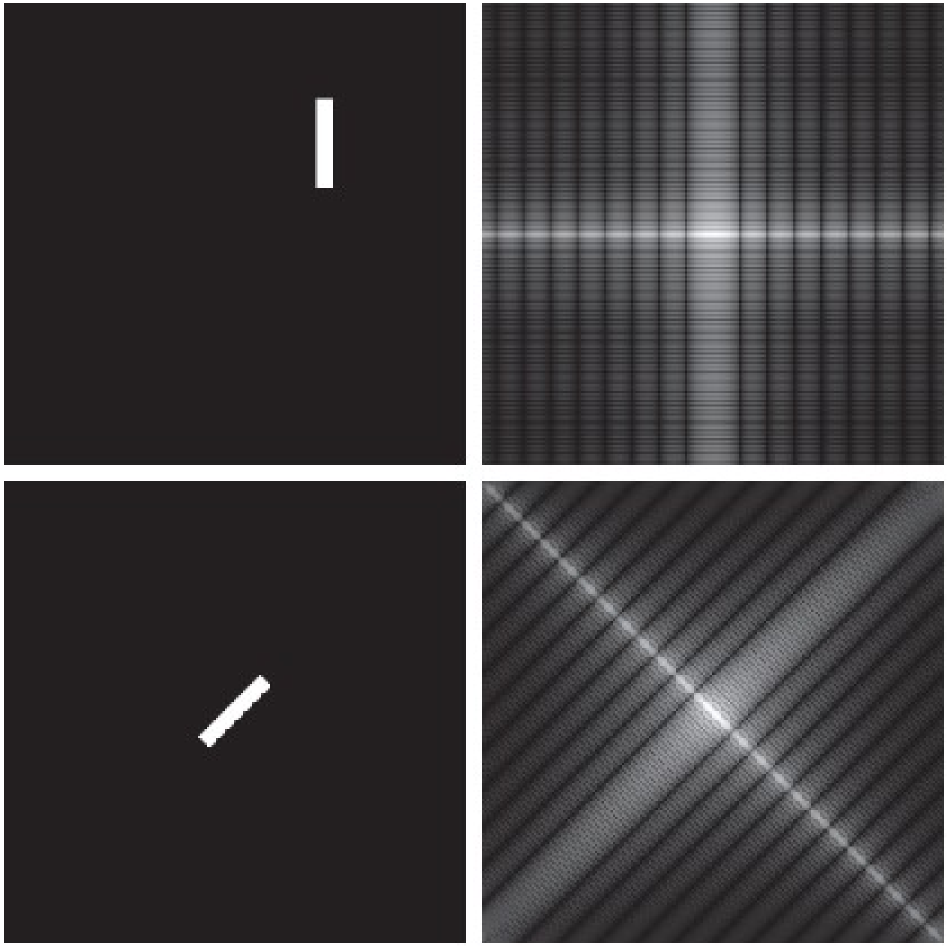
# DFT & Images (cont…)




a b
c d

**FIGURE 4.23**
(a) Image. (b) Spectrum, showing small, bright areas in the four corners (you have to look carefully to see them). (c) Centered spectrum. (d) Result after a log transformation. The zero crossings of the spectrum are closer in the vertical direction because the rectangle in (a) is longer in that direction. The right-handed coordinate convention used in the book places the origin of the spatial and frequency domains at the top left (see Fig. 2.19).

# DFT & Images (cont…)



a b
c d

**FIGURE 4.24**
(a) The rectangle in Fig. 4.23(a) translated.
(b) Corresponding spectrum.
(c) Rotated rectangle.
(d) Corresponding spectrum.
The spectrum of the translated rectangle is identical to the spectrum of the original image in Fig. 4.23(a).

# DFT & Images (cont…)

a b c

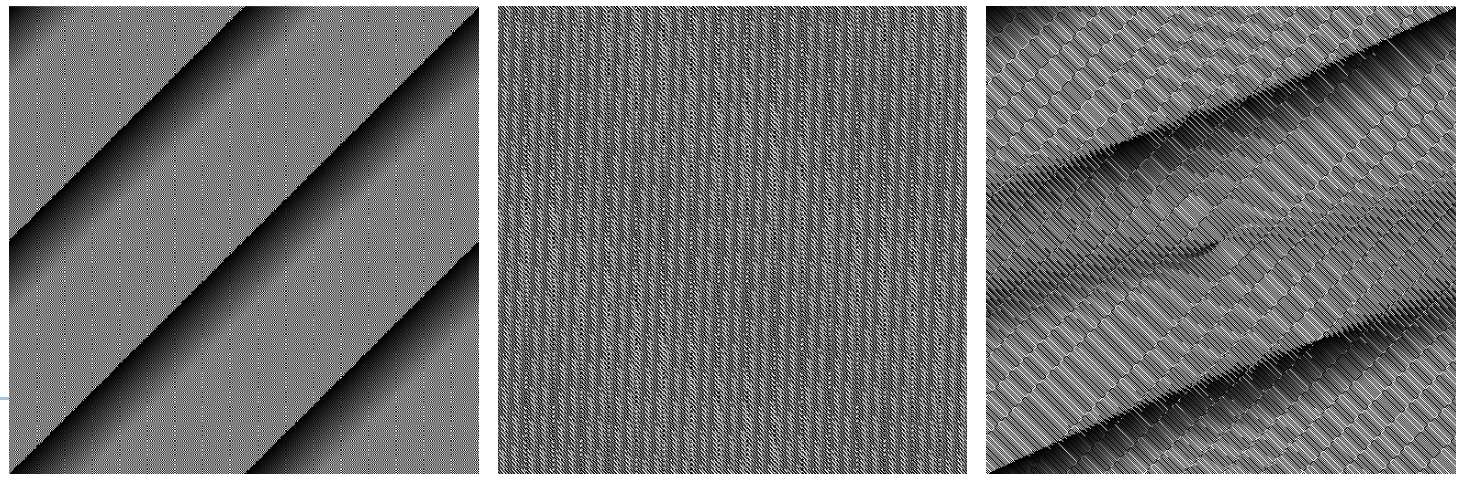


**FIGURE 4.25** Phase angle images of (a) centered, (b) translated, and (c) rotated rectangles.

Although the images differ by a simple geometric transformation no intuitive information may be extracted from their phases regarding their relation.

# DFT & Images (cont…)

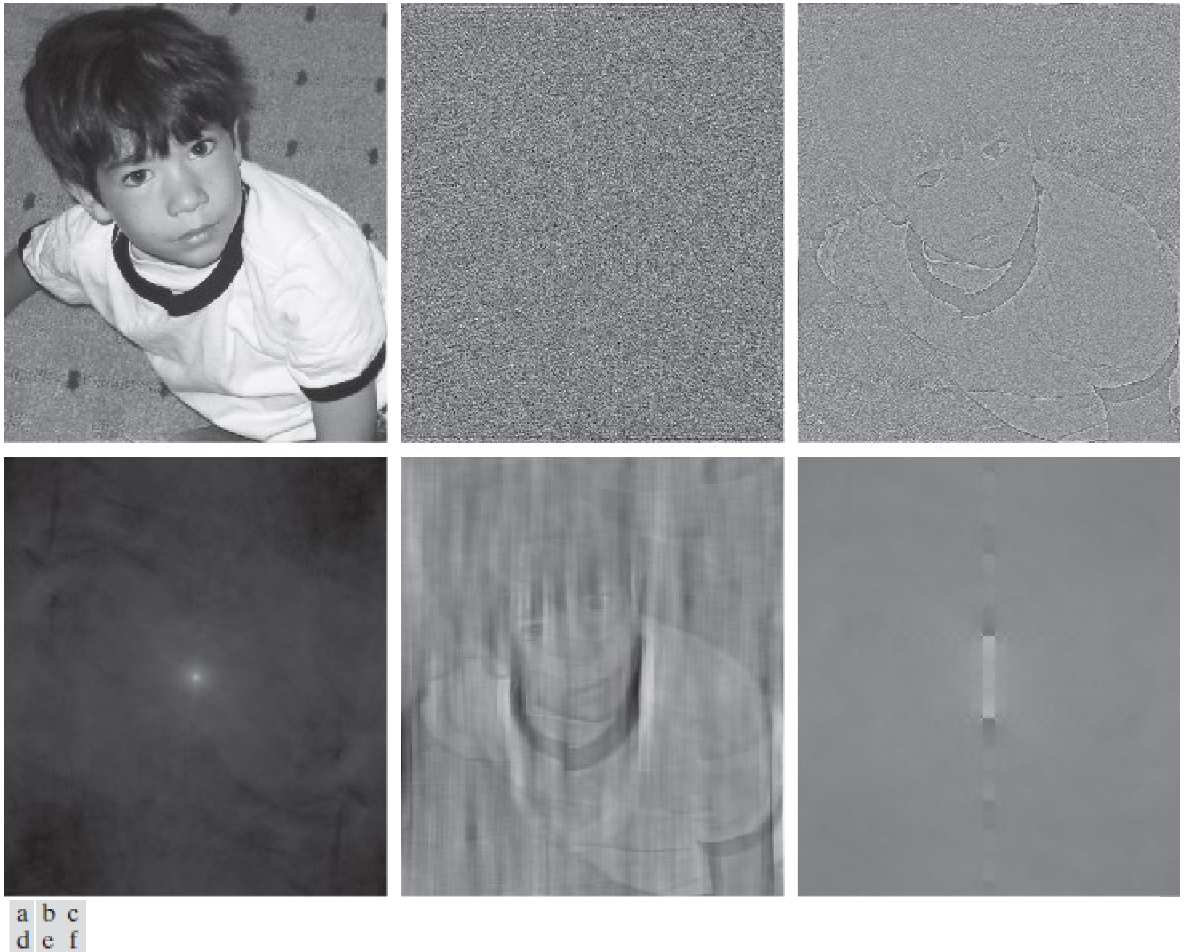
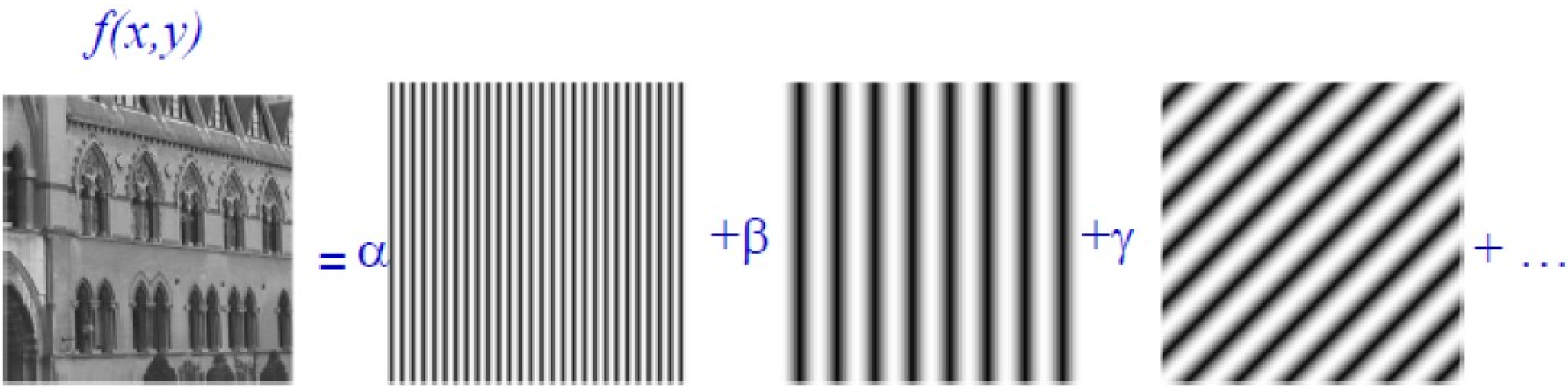



**FIGURE 4.26** (a) Boy image. (b) Phase angle. (c) Boy image reconstructed using only its phase angle (all shape features are there, but the intensity information is missing because the spectrum was not used in the reconstruction). (d) Boy image reconstructed using only its spectrum. (e) Boy image reconstructed using its phase angle and the spectrum of the rectangle in Fig. 4.23(a). (f) Rectangle image reconstructed using its phase and the spectrum of the boy's image.

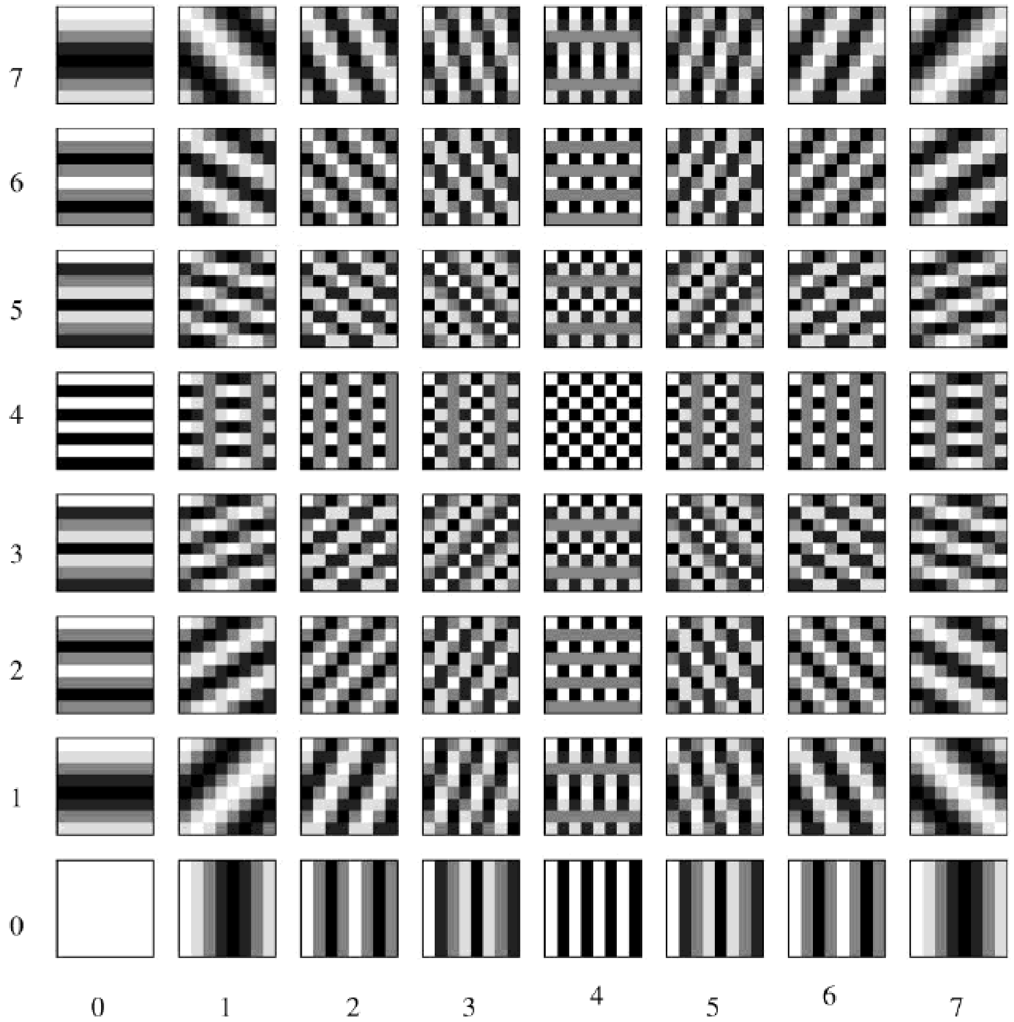
# DFT in image domain (interpretation example)

- Basis functions example of 2D continuous FT (real part)



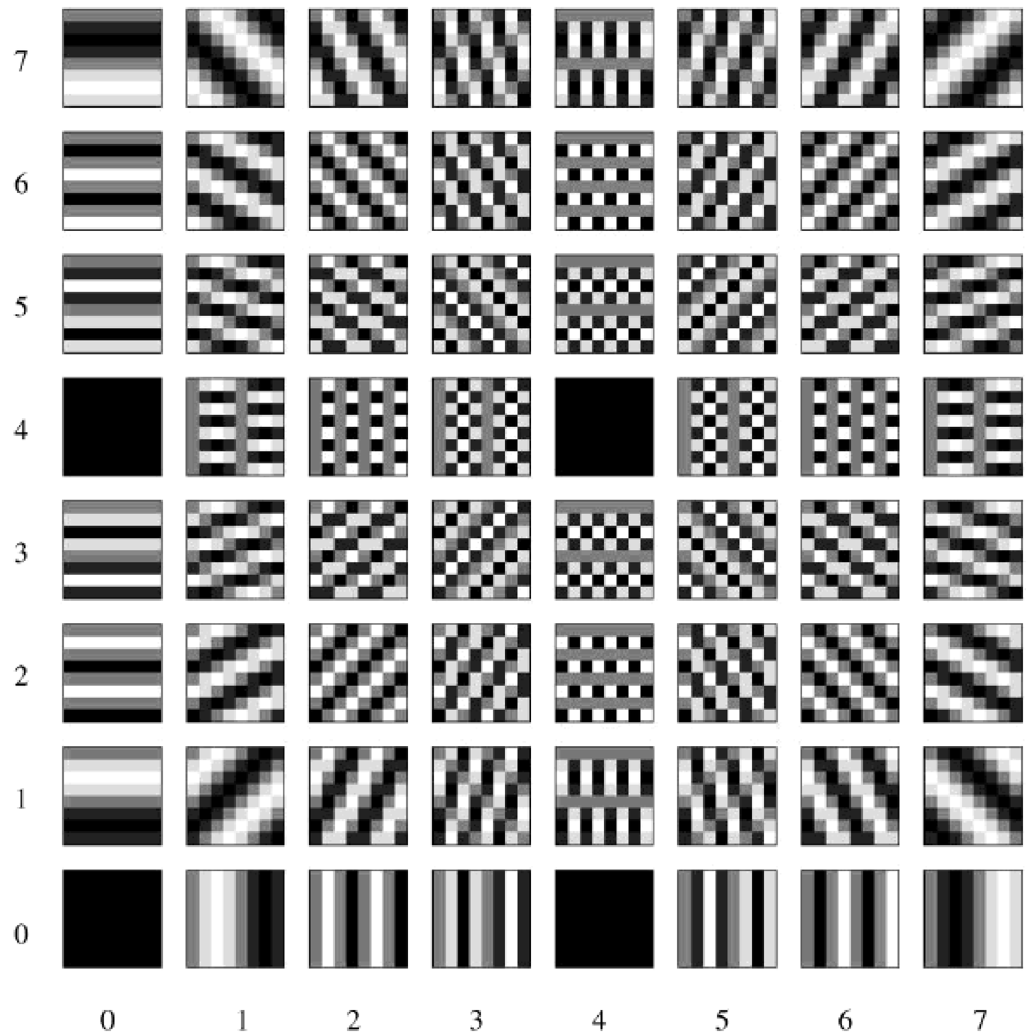
# DFT in image domain (interpretation example)

- Basis functions example of 2D DFT (real part – 8x8))



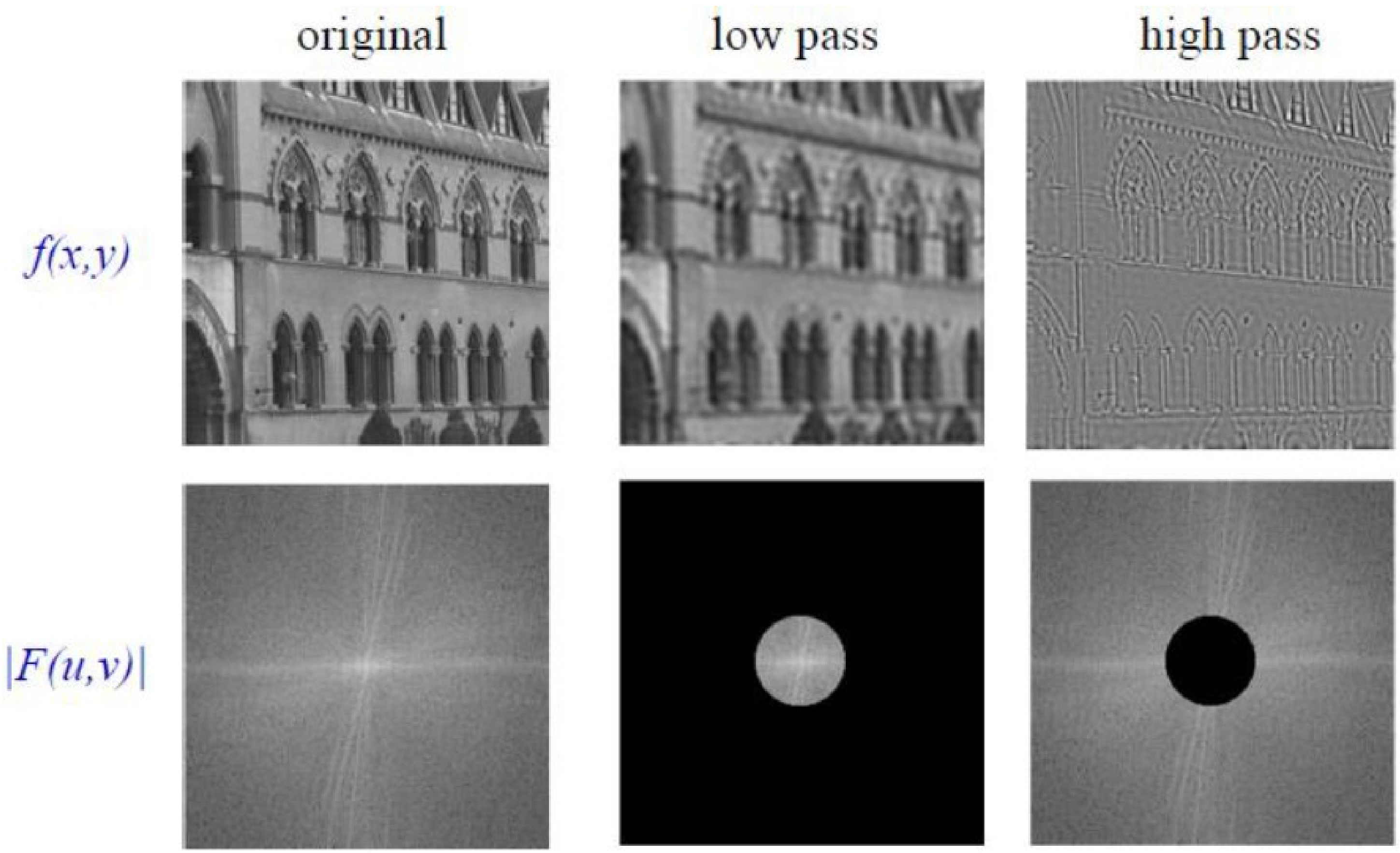
# DFT in image domain (interpretation example)

- Basis functions example of 2D DFT (imaginary part – 8x8))



# DFT in image domain (interpretation example)

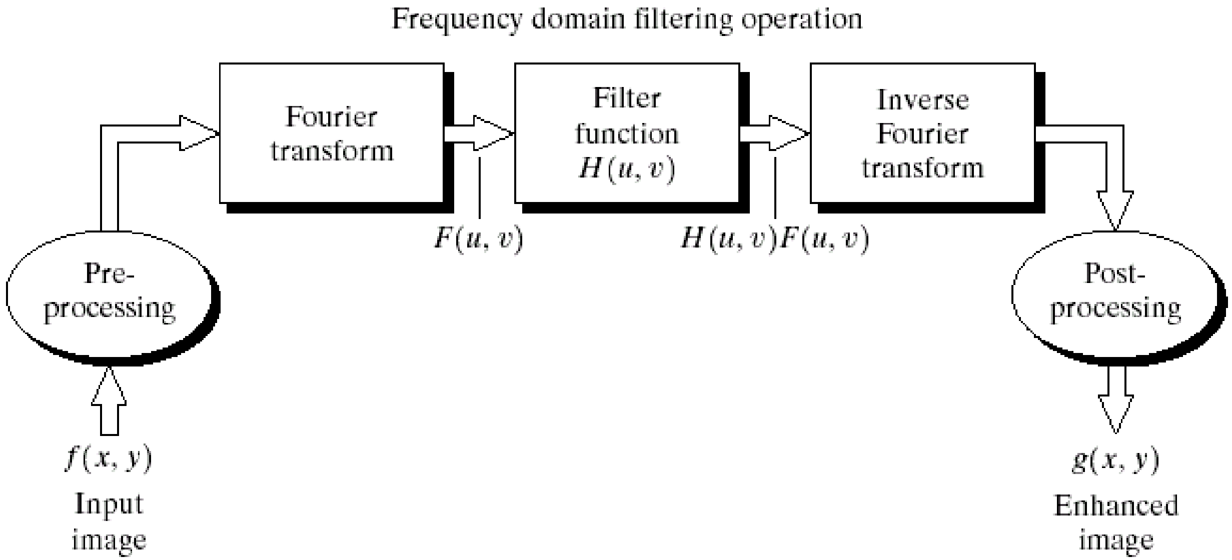
- Keeping part of all the terms is one way to construct filters in the frequency domain.



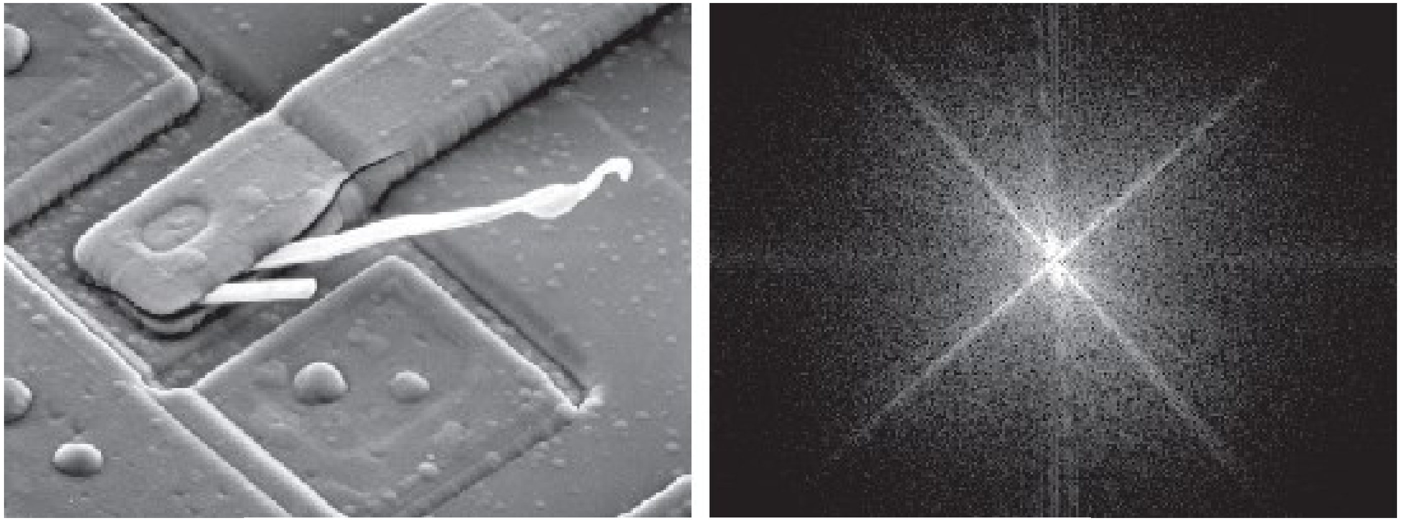
# The DFT and Image Processing

To filter an image in the frequency domain:

1. Compute $F(u,v)$ the DFT of the image
2. Multiply $F(u,v)$ by a filter function $H(u,v)$
3. Compute the inverse DFT of the result



# Some Basic Frequency Domain Filters

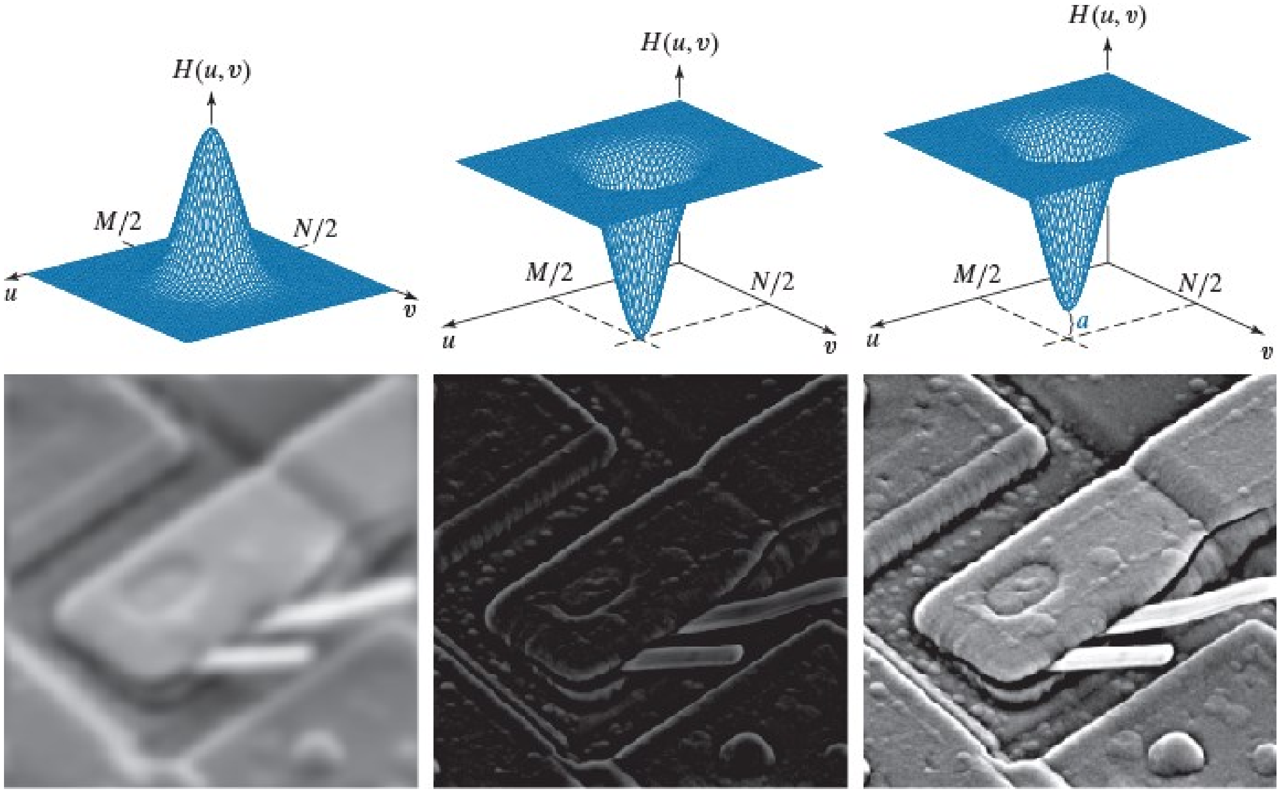


a b

**FIGURE 4.28** (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.)

The DFT is centered after multiplication of the image by $(-1)^{m+n}$

# Some Basic Frequency Domain Filters (cont.)



# The importance of zero padding (cont.)

The DFT considers that the signal is periodic and produces *wraparound* errors.

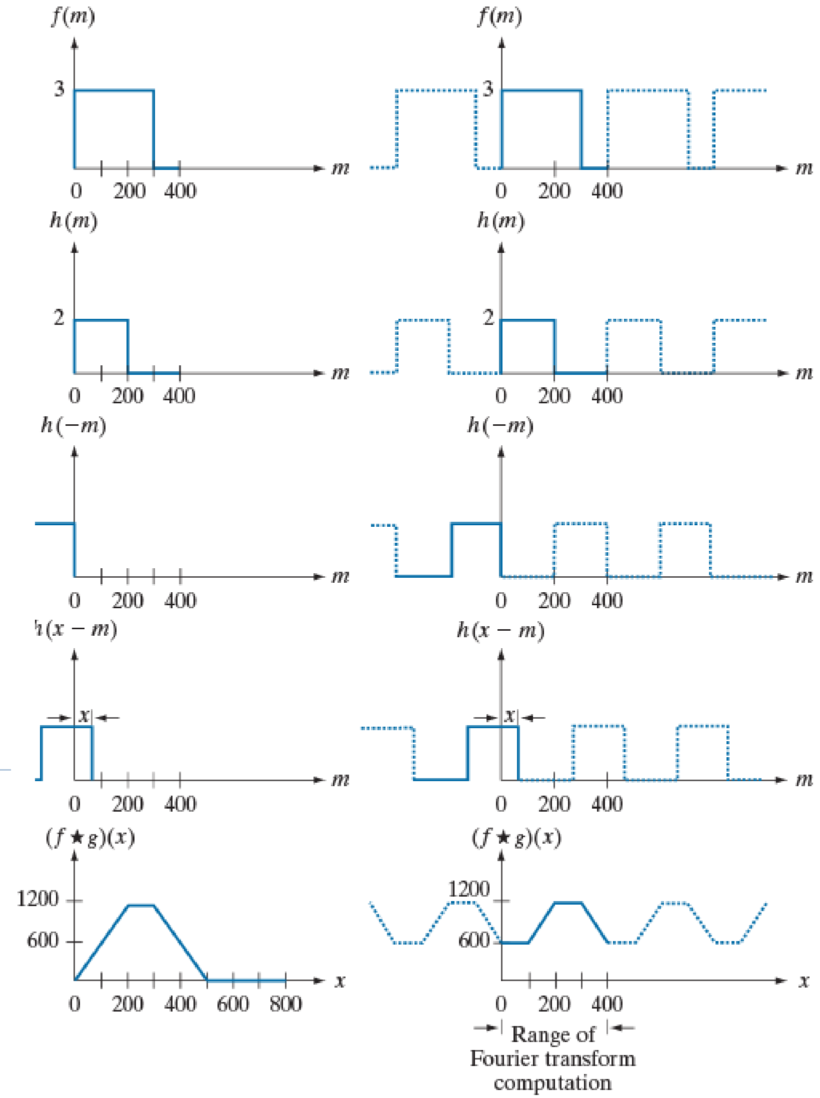



**FIGURE 4.27** Left column: Spatial convolution computed with Eq. (3-44), using the approach discussed in Section 3.4. Right column: Circular convolution. The solid line in (j) is the result we would obtain using the DFT, or, equivalently, Eq. (4-48). This erroneous result can be remedied by using zero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

# The importance of zero padding (cont.)

- All of the properties of 1D DFT hold.
- Particularly:
  - Let $f[m,n]$ be of size $M_1 \mathrm{x} N_1$ and $h[m,n]$ of size $M_2 \mathrm{x} N_2$ .
  - If the signals are zero-padded to size $(M_1+M_2-1)\mathrm{x}(N_1+N_2-1)$ then their circular convolution will be the same as their linear convolution and:

$$\tilde{g}[m,n] = \tilde{f}[m,n] * \tilde{h}[m,n] \leftrightarrow \tilde{G}[k,l] = \tilde{F}[k,l]\tilde{H}[k,l]$$

# The importance of zero padding (cont.)

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).

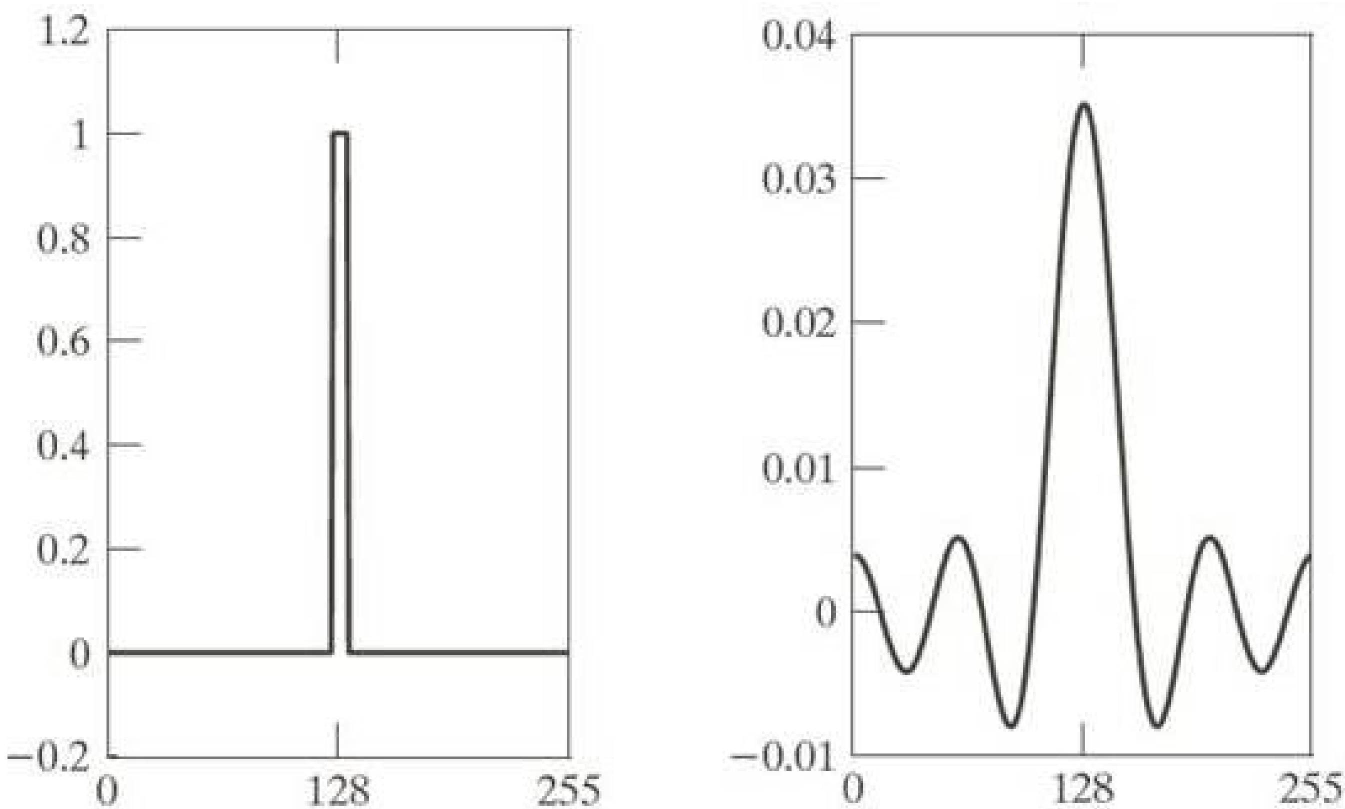
- The image and the DFT are considered to be periodic.
- The vertical edges of the middle image are not blurred if no padding is applied. Why?

# The importance of zero padding

- In reality, we perform padding in the spatial domain, while the filter is defined in the frequency domain, which poses a problem. One solution is (naïve approach):
  - Compute the inverse DFT of the filter.
  - Pad the filter in the spatial domain to have the same size as the image.
  - Compute its DFT to return to the frequency domain.

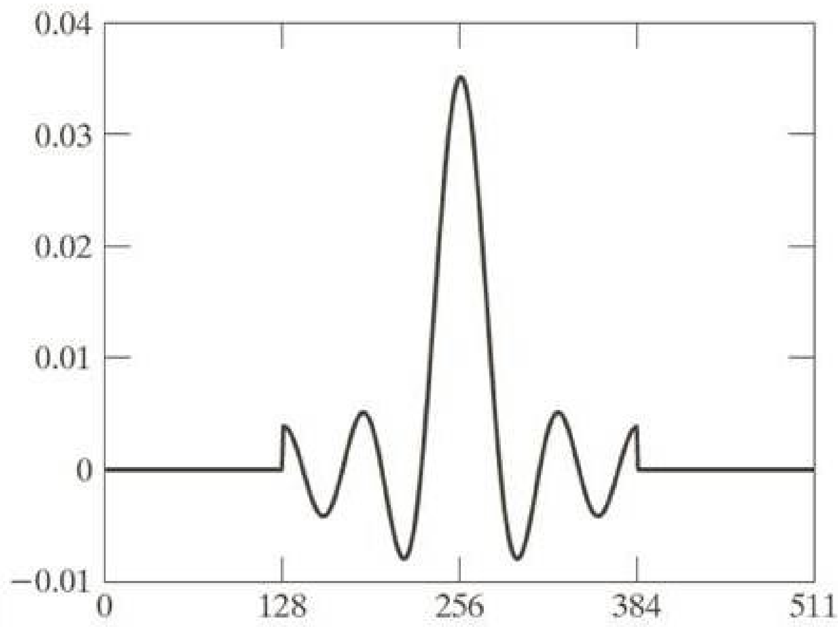
# Spatial zero-padding and filters (cont.)

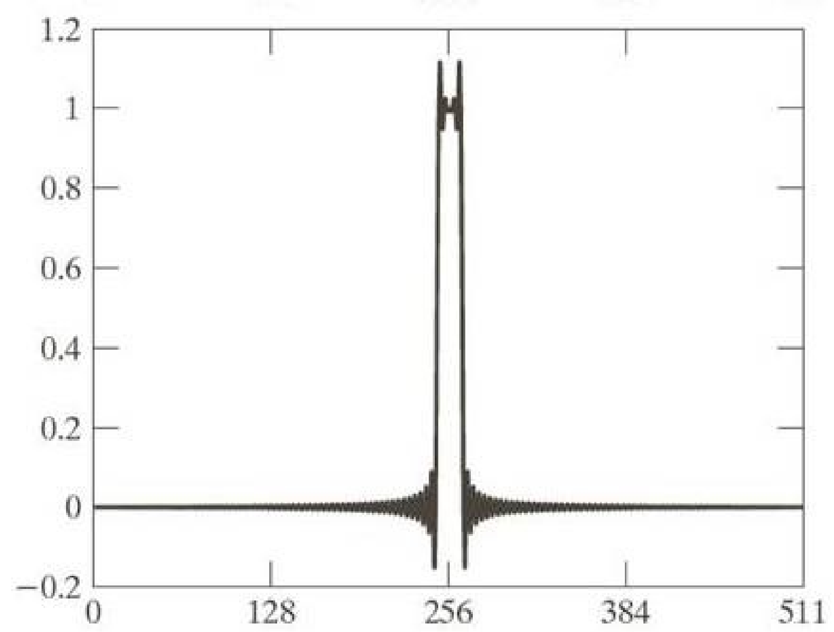
- The filter and its inverse DFT of length 256 (continuous line)



# Spatial zero-padding and filters (cont.)

- Zero-padded filter and its DFT




- Spatial truncation of the filter results in ringing effects.

# Spatial zero-padding and filters (cont.)

- We cannot work with an infinite number of filter components and simultaneously perform zero-padding to avoid aliasing.
- A decision on which limitation to accept is required.
- One solution to zero-pad the image and then use a filter of the same size with no zero-padding
  - Small errors due to aliasing but it is generally preferable than ringing.
- Another solution is to choose filters attenuating gradually instead of ideal filters.

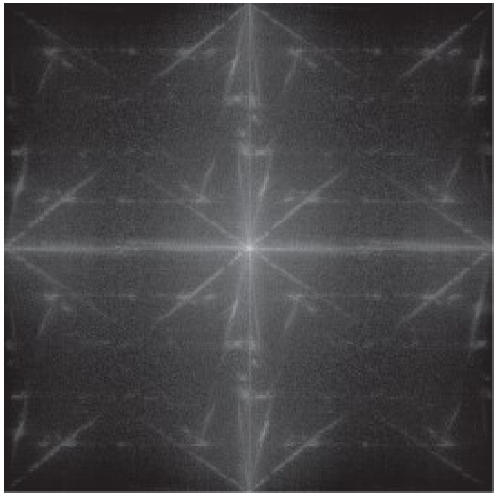
# Spatial zero-padding and filters (cont.)

- Smooth regions in terms of intensity are related to low frequencies.

- *Low-pass* filters allow low frequencies to pass through while attenuating high frequencies.

# Steps of filtering in the DFT domain

- What if the filter is known in the spatial domain?



| | | |
|---|---|---|
| -1 | 0 | 1 |
| -2 | 0 | 2 |
| -1 | 0 | 1 |

- Apply a 3x3 Sobel filter to the 600x600 image in the frequency domain.

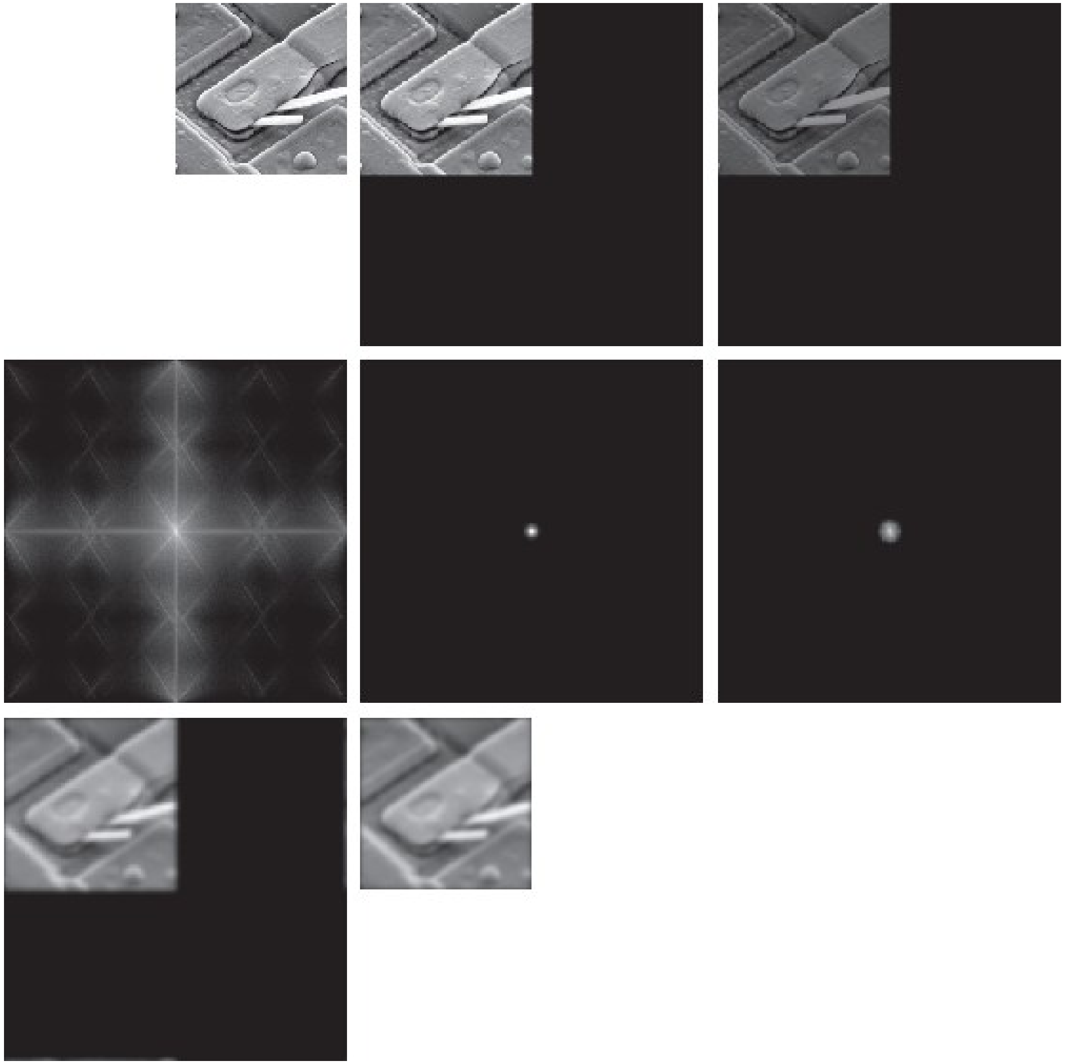
# Steps of filtering in the DFT domain (cont.)

1. Pad the image and the filter to 602x602.
2. Place the filter to the center of the 602x602 padded array.
3. Multiply the filter by $(-1)^{m+n}$ to place the center of the filter to the top left corner (0,0) of the array.
4. Compute the DFT of the filter.
5. Compute the DFT of the image.
6. Multiply the DFTs and invert.

# Steps of filtering in the DFT domain (cont.)

a b c
d e f
g h

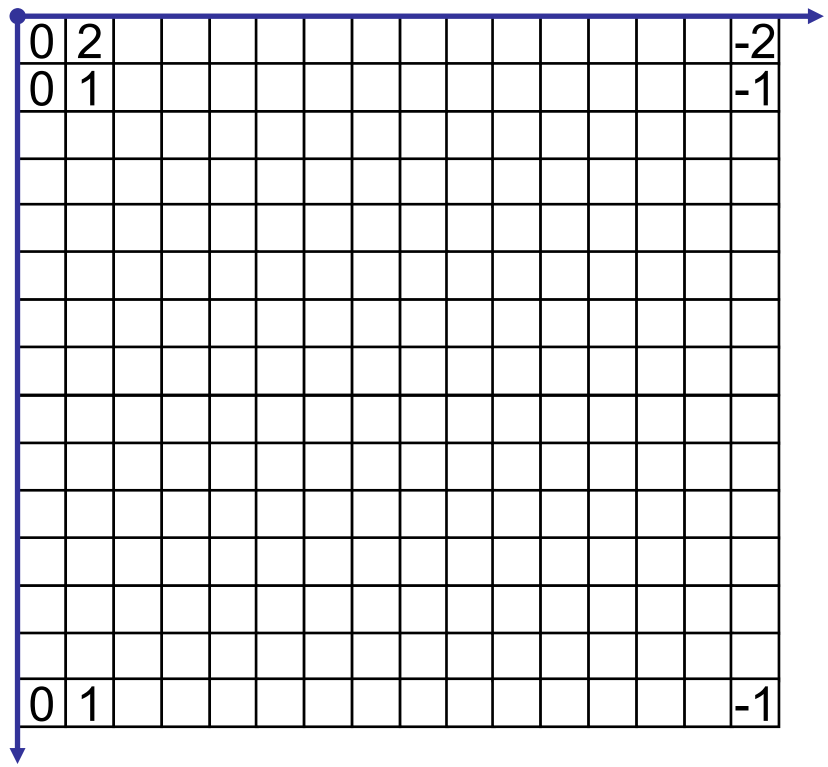
**FIGURE 4.35**
(a) An $M \times N$ image, $f$.
(b) Padded image, $f_p$ of size $P \times Q$.
(c) Result of multiplying $f_p$ by $(-1)^{x+y}$.
(d) Spectrum of $F$. (e) Centered Gaussian lowpass filter transfer function, $H$, of size $P \times Q$.
(f) Spectrum of the product $HF$.
(g) Image $g_p$, the 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 $g_p$.



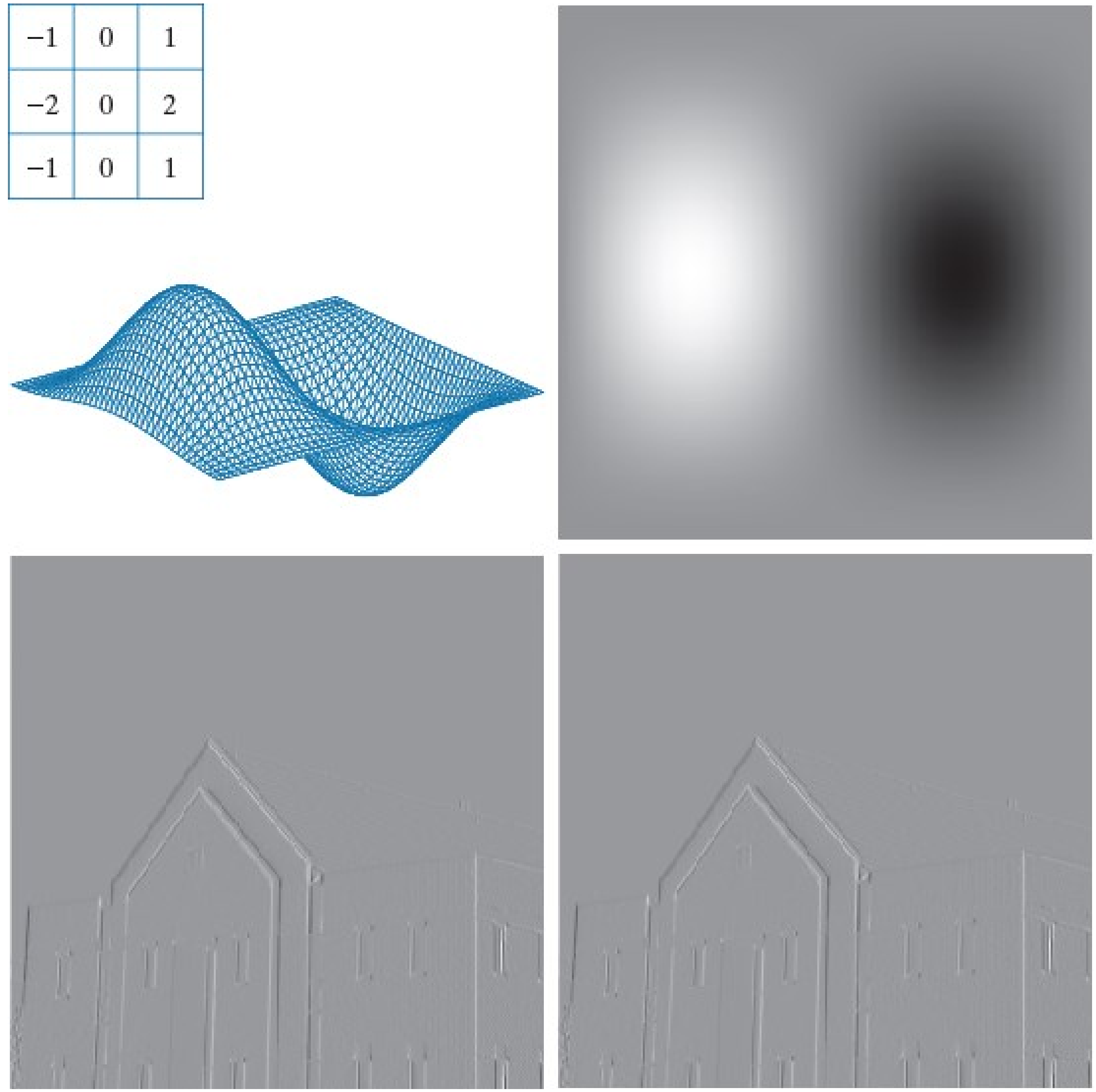
# Steps of filtering in the DFT domain (cont.)

- Alternatively, pad to 602x602, repeat the filter periodically and compute the DFT.

| | | |
|---|---|---|
| -1 | 0 | 1 |
| -2 | 0 | 2 |
| -1 | 0 | 1 |



# The importance of zero padding (cont...)



| | | |
|---|---|---|
| −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.

# Smoothing Frequency Domain Filters

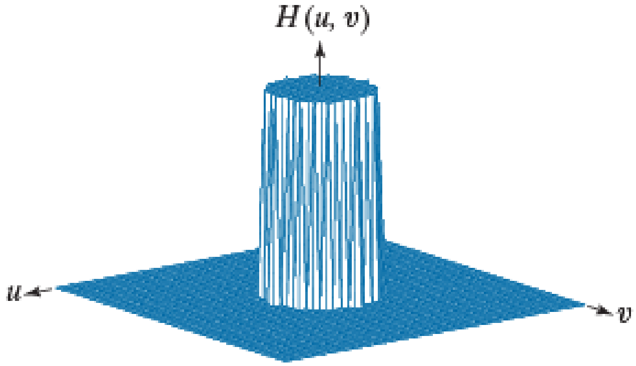
- Smoothing is achieved in the frequency domain by dropping out the high frequency components
- The basic model for filtering is:

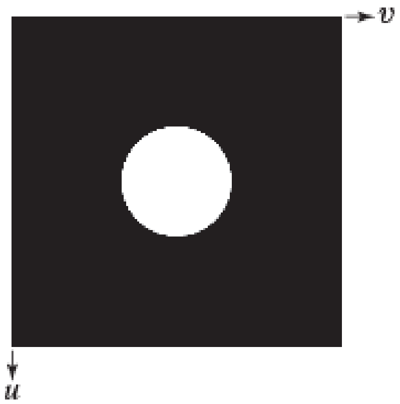
$$G(u,v) = H(u,v)F(u,v)$$

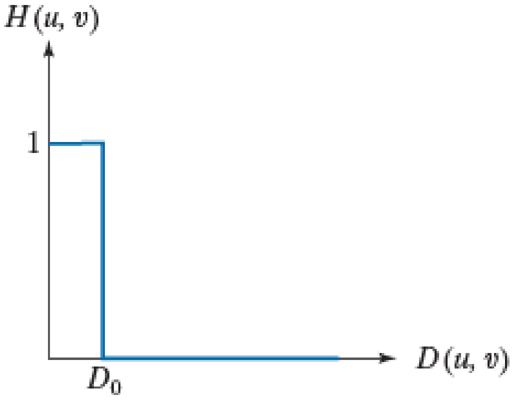
- where $F(u,v)$ is the Fourier transform of the image being filtered and $H(u,v)$ is the filter transform function
- *Low pass filters* – only pass the low frequencies, drop the high ones.

# Ideal Low Pass Filter

Simply cut off all high frequency components that are a specified distance $D_0$ from the origin of the transform.

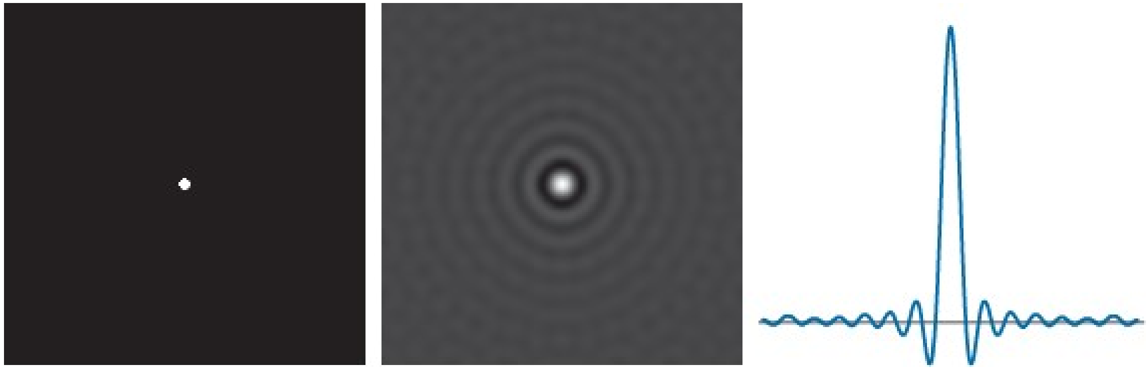



Changing the distance changes the behaviour of the filter.

# Ideal Lowpass Filters (cont...)



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).

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

# Ideal Low Pass Filter (cont…)

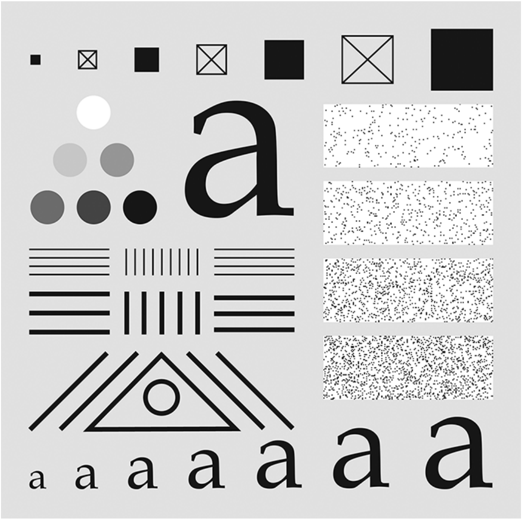
The transfer function for the ideal low pass filter can be given as:

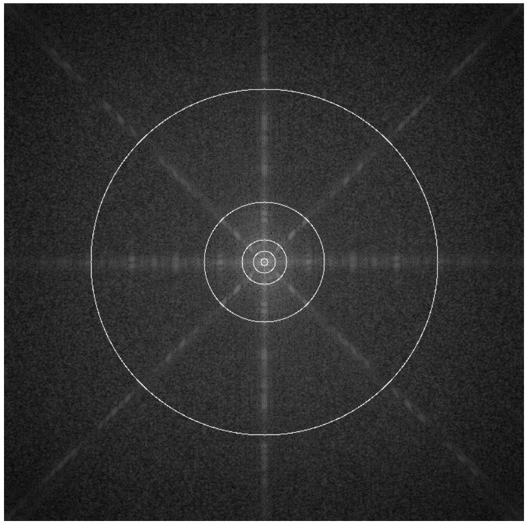
$$H(u,v) = \begin{cases} 1 & \text{if } D(u,v) \leq D_0 \\ 0 & \text{if } D(u,v) > D_0 \end{cases}$$

where $D(u,v)$ is given as:

$$D(u,v) = [(u - M/2)^2 + (v - N/2)^2]^{1/2}$$

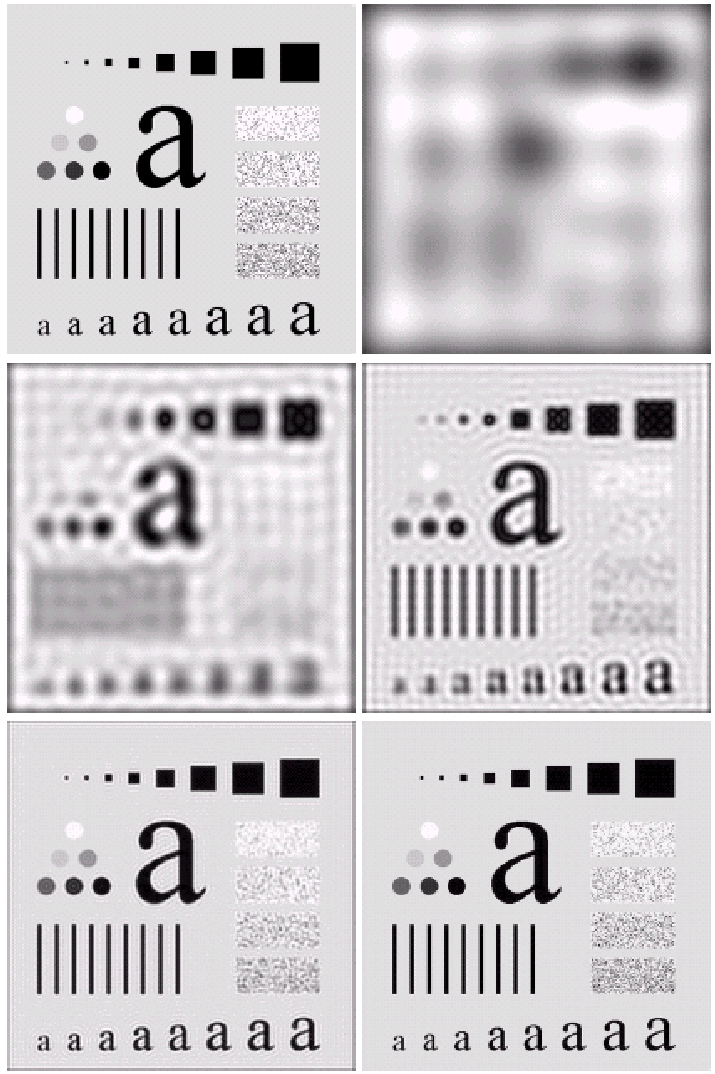
# Ideal Low Pass Filter (cont…)





An image, its Fourier spectrum and a series of ideal low pass filters of radius 5, 15, 30, 80 and 230 superimposed on top of it.

# Ideal Low Pass Filter (cont…)



Original image

ILPF of radius $D_0 = 5$

ILPF of radius $D_0 = 15$

ILPF of radius $D_0 = 30$

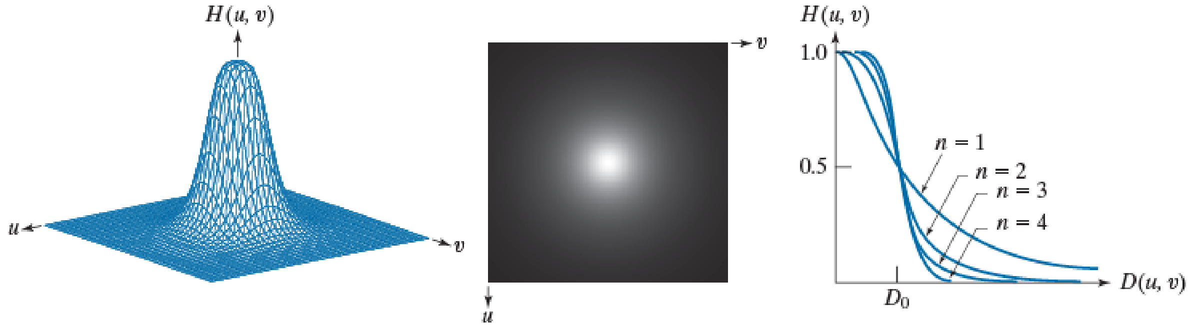
ILPF of radius $D_0 = 80$

ILPF of radius $D_0 = 230$

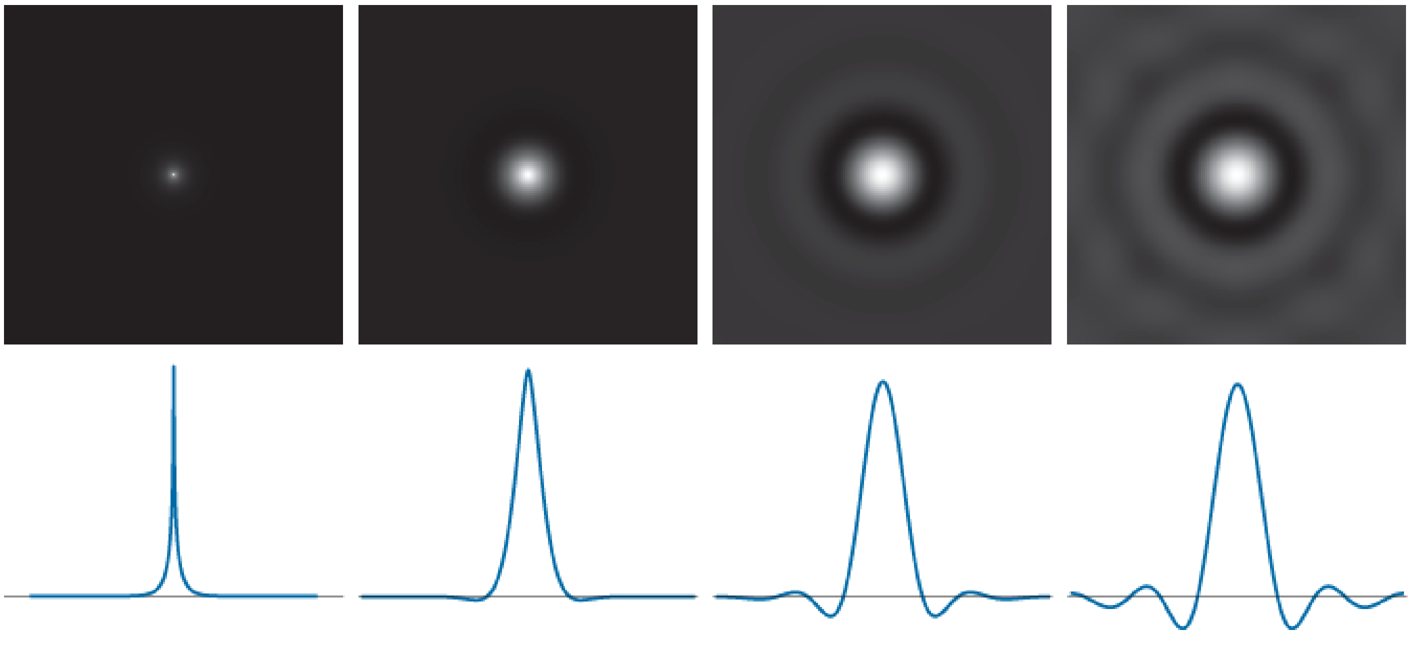
# Butterworth Lowpass Filters

- The transfer function of a Butterworth lowpass filter of order $n$ with cutoff frequency at distance $D_0$ from the origin is defined as:

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



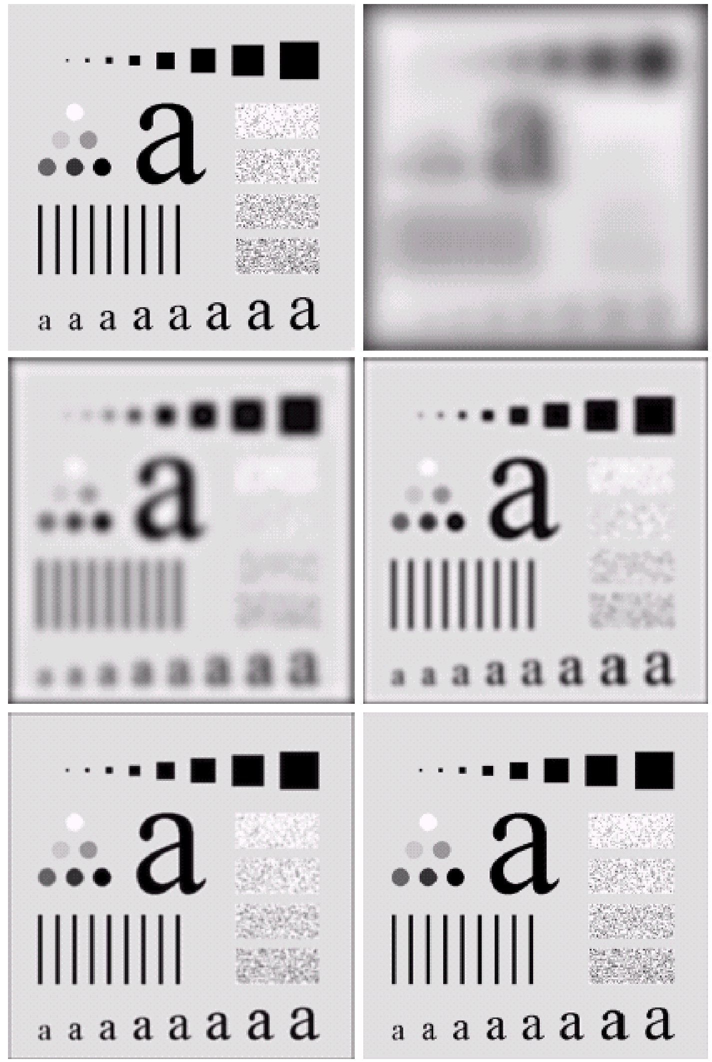
# Butterworth Lowpass Filters (cont...)



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
$1000 \times 1000$ 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.

# Butterworth Lowpass Filter (cont…)



Original image

BLPF $n$=2, $D_0$=5

BLPF $n$=2, $D_0$=15

BLPF $n$=2, $D_0$=30

BLPF $n$=2, $D_0$=80

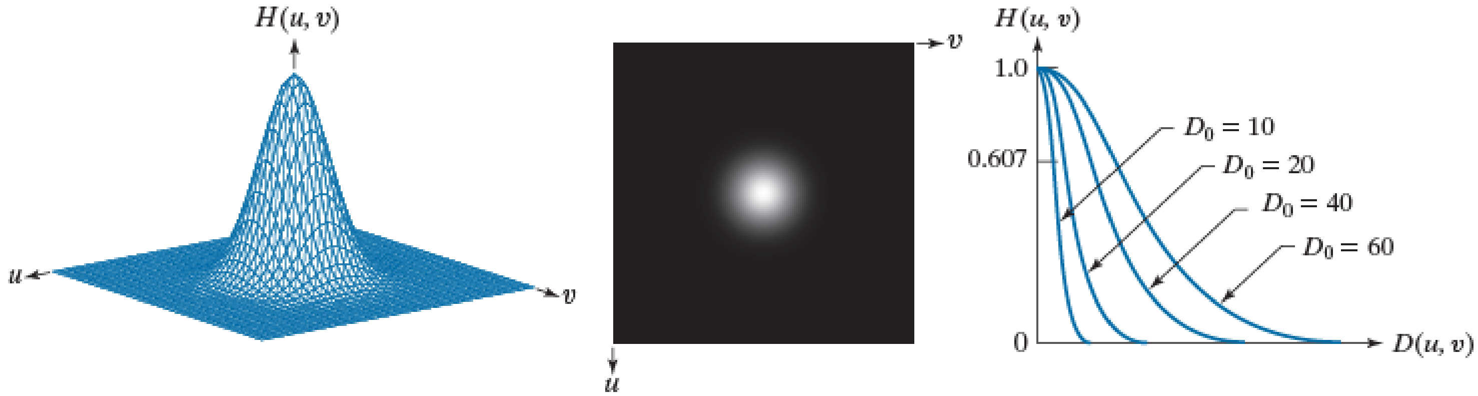
BLPF $n$=2, $D_0$=230

Less ringing than ILPF and more smoothing than the Gaussian

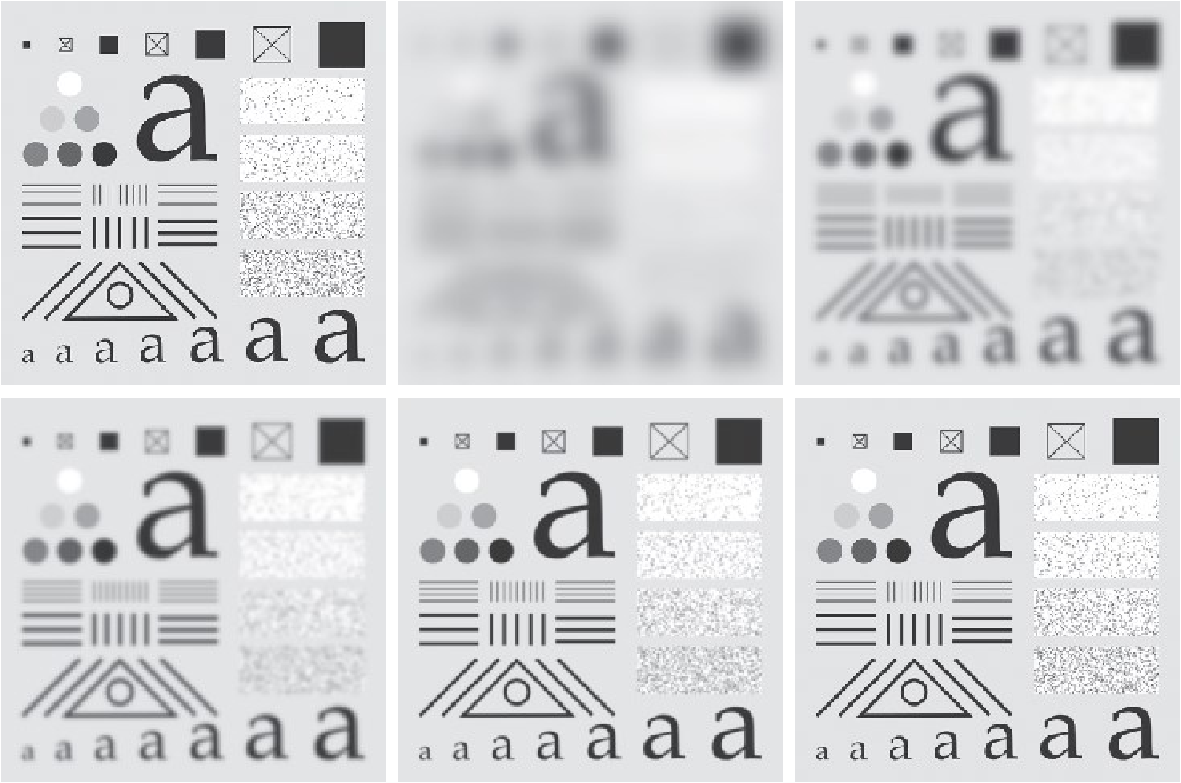
# Gaussian Lowpass Filters

- The transfer function of a Gaussian lowpass filter is defined as:

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



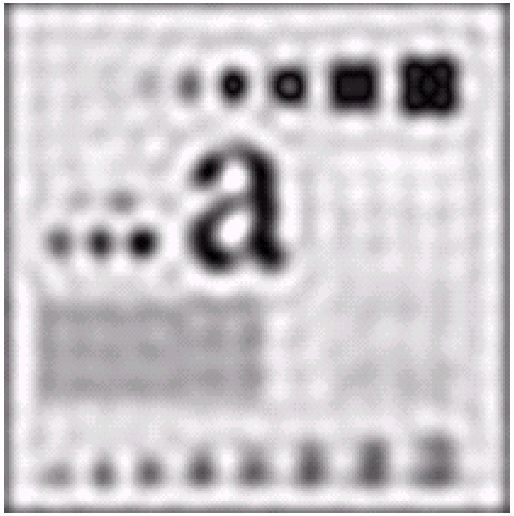
# Gaussian Lowpass Filters (cont…)

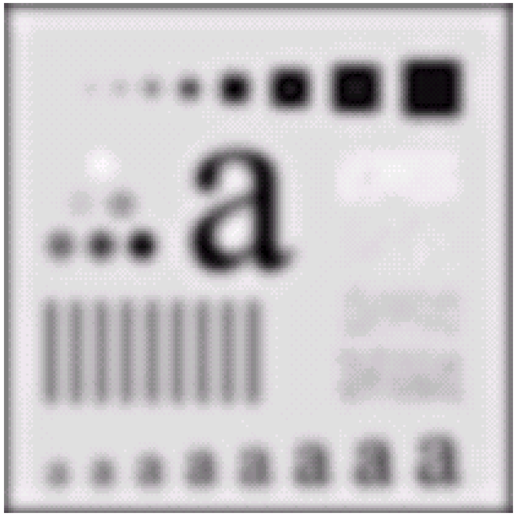


Less ringing than ILPF due to smoother transition

# Lowpass Filters Compared

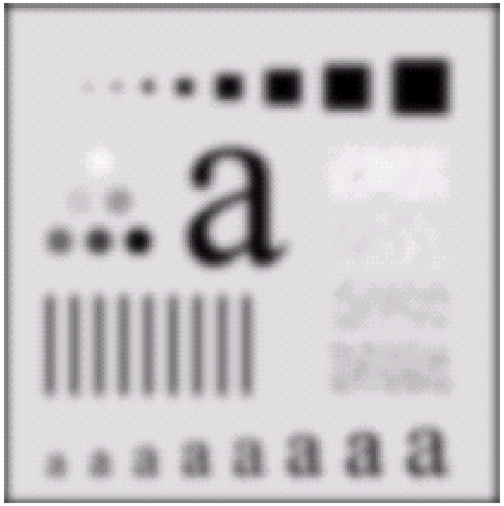
ILPF $D_0$=15





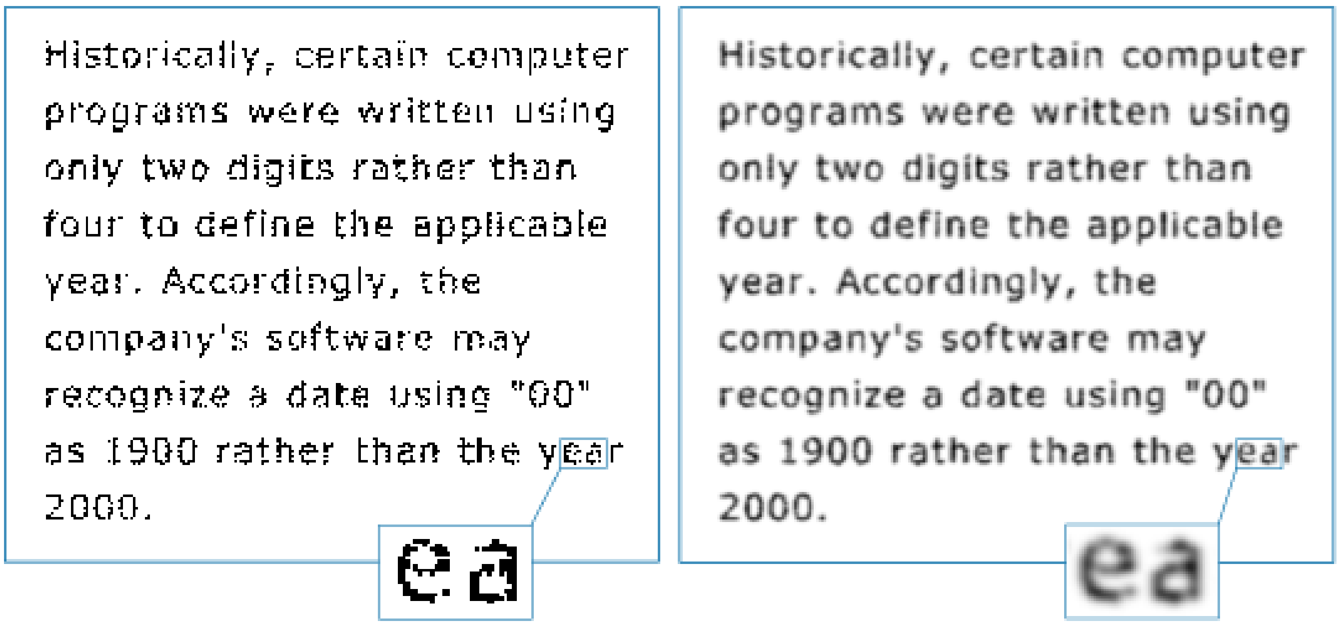
BLPF $n$=2, $D_0$=15

Gaussian $D_0$=15



# Lowpass Filtering Examples

A low pass Gaussian filter is used to connect broken text



# Lowpass Filtering Examples (cont…)

- Different lowpass Gaussian filters used to remove blemishes in a photograph.

# Sharpening in the Frequency Domain

- Edges and fine detail in images are associated with high frequency components
- *High pass filters* – only pass the high frequencies, drop the low ones
- High pass frequencies are precisely the reverse of low pass filters, so:

$$H_{hp}(u, v) = 1 - H_{lp}(u, v)$$

# High pass filters

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

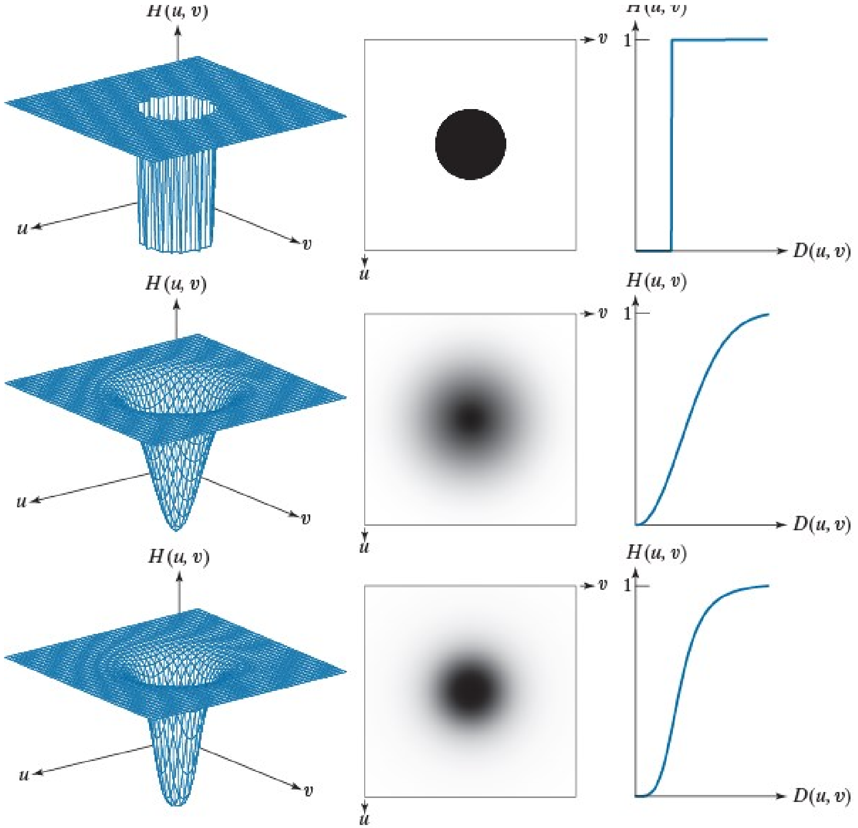



**FIGURE 4.51** Top row: Perspective plot, image, and, radial cross section of an IHPF transfer function. Middle and bottom rows: The same sequence for GHPF and BHPF transfer functions. (The thin image borders were added for clarity. They are not part of the data.)

# High pass filters (spatial domain)

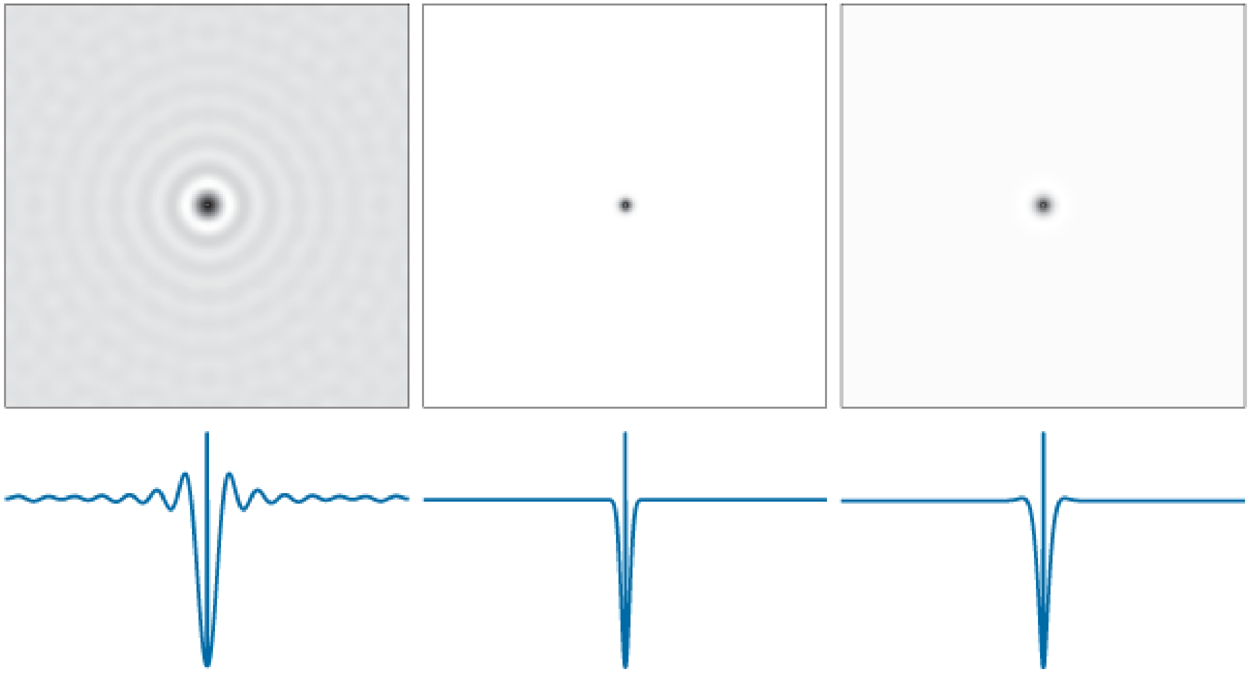
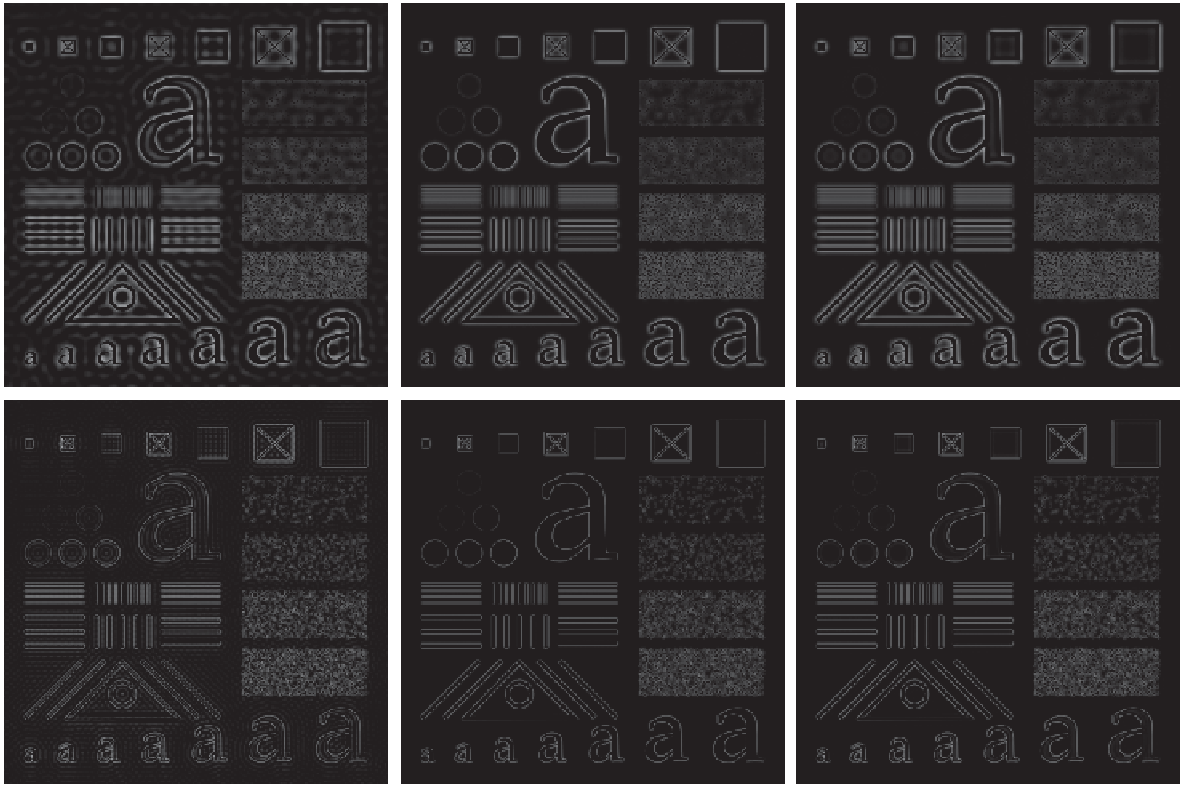



FIGURE 4.52 (a)–(c): Ideal, Gaussian, and Butterworth highpass spatial kernels obtained from IHPF, GHPF, and BHPF frequency-domain transfer functions. (The thin image borders are not part of the data.) (d)–(f): Horizontal intensity profiles through the centers of the kernels.

# High pass filtering



a b c
d e f

**FIGURE 4.53** Top row: The image from Fig. 4.40(a) filtered with IHPF, GHPF, and BHPF transfer functions using $D_0 = 60$ in all cases ($n = 2$ for the BHPF). Second row: Same sequence, but using $D_0 = 160$.

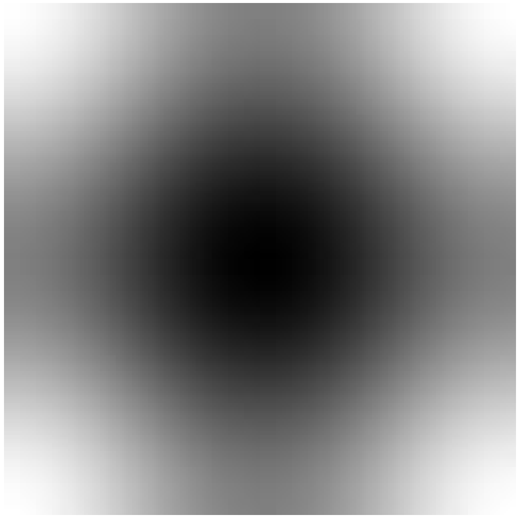
# High pass filtering

High-pass filtering followed by thresholding to highlight the details

# The Laplacian in the Frequency Domain

- Image enhancement operations (e.g. unsharp masking, high boost filtering) may be alternatively implemented in the frequency domain.
- Laplacian in the DFT domain:

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

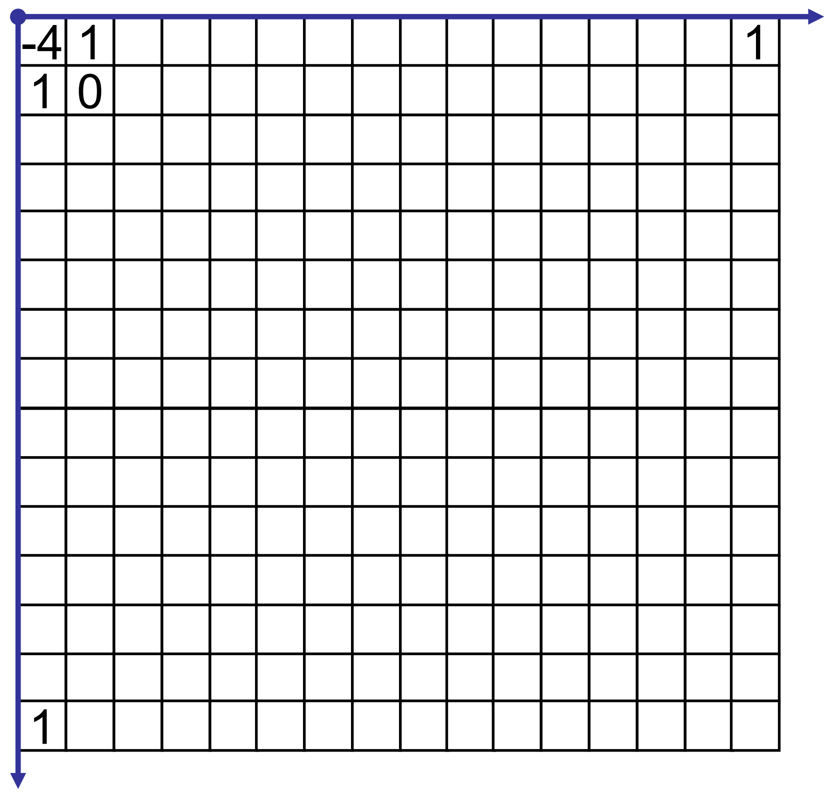


# Steps of filtering in the DFT domain (cont.)

- To obtain it in the frequency domain, we zero-pad the image and repeat the content periodically.

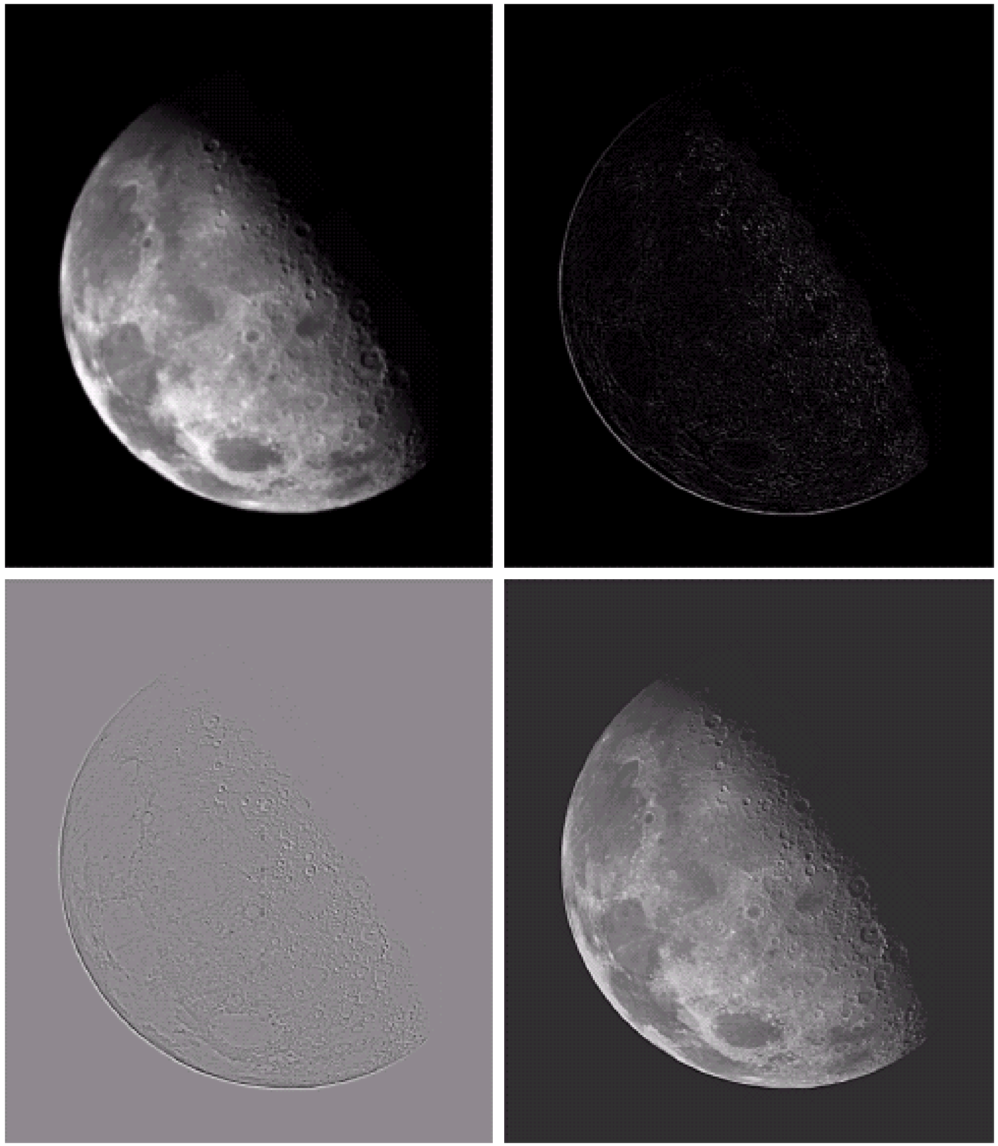
| | | |
|---|---|---|
| 0 | 1 | 0 |
| 1 | -4 | 1 |
| 0 | 1 | 0 |

- The center of the mask is at (0,0).



# Frequency Domain Laplacian Example

Original image



Laplacian filtered image

Laplacian image scaled

Enhanced image

# Application – periodic noise removal

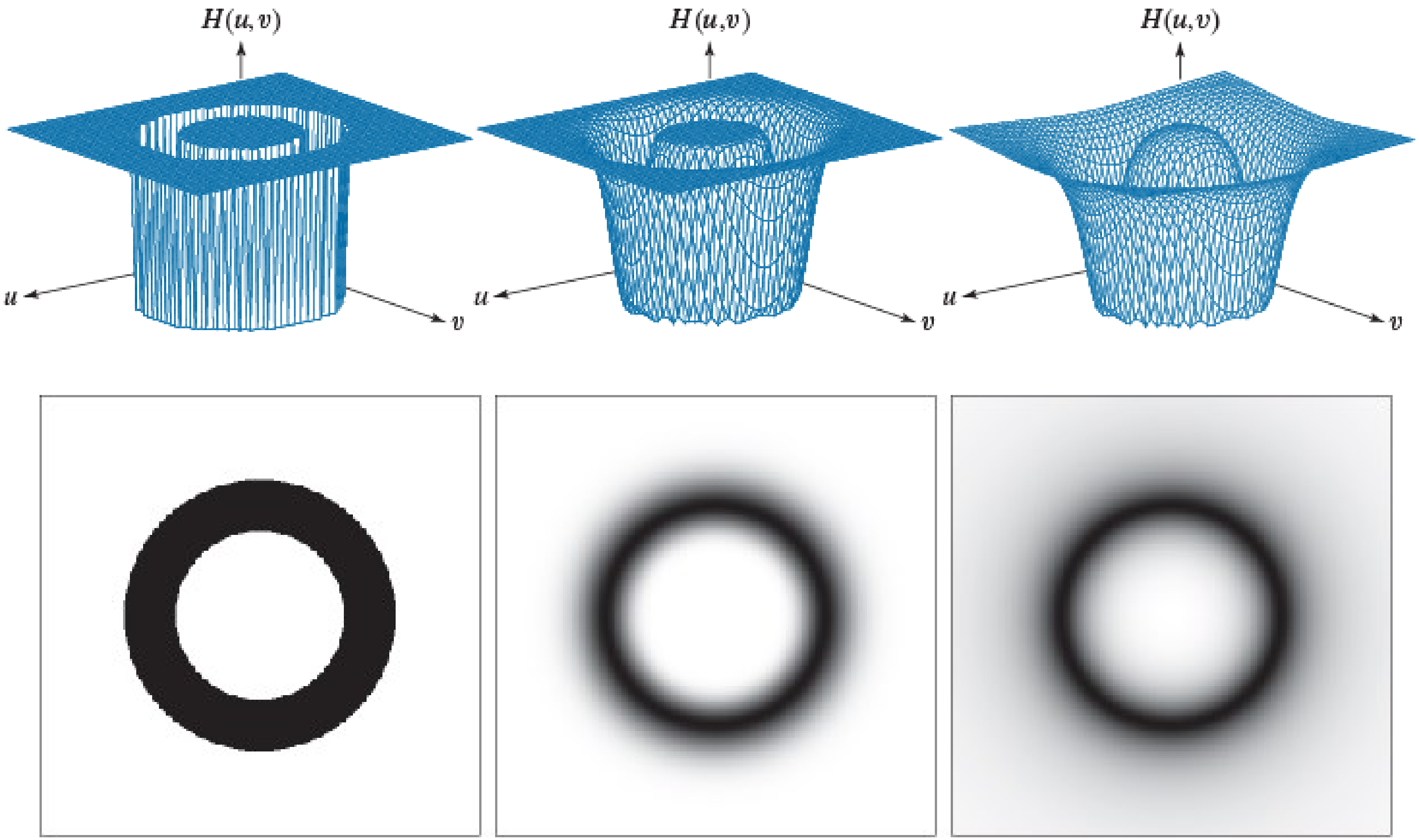
- Periodic noise or patterns often appear as a result of electrical or electromagnetic interference, Moiré patterns, etc.
- Frequency domain techniques are the most effective for addressing such types of noise.
- Some synonymous/similar techniques include:
  - Bandpass filters (ζωνοπερατά φίλτρα)
  - Notch filters (φίλτρα εγκοπής)
  - Bandreject/stop filters (ζωνοφρακτικά φίλτρα)

# Band-Stop Filters

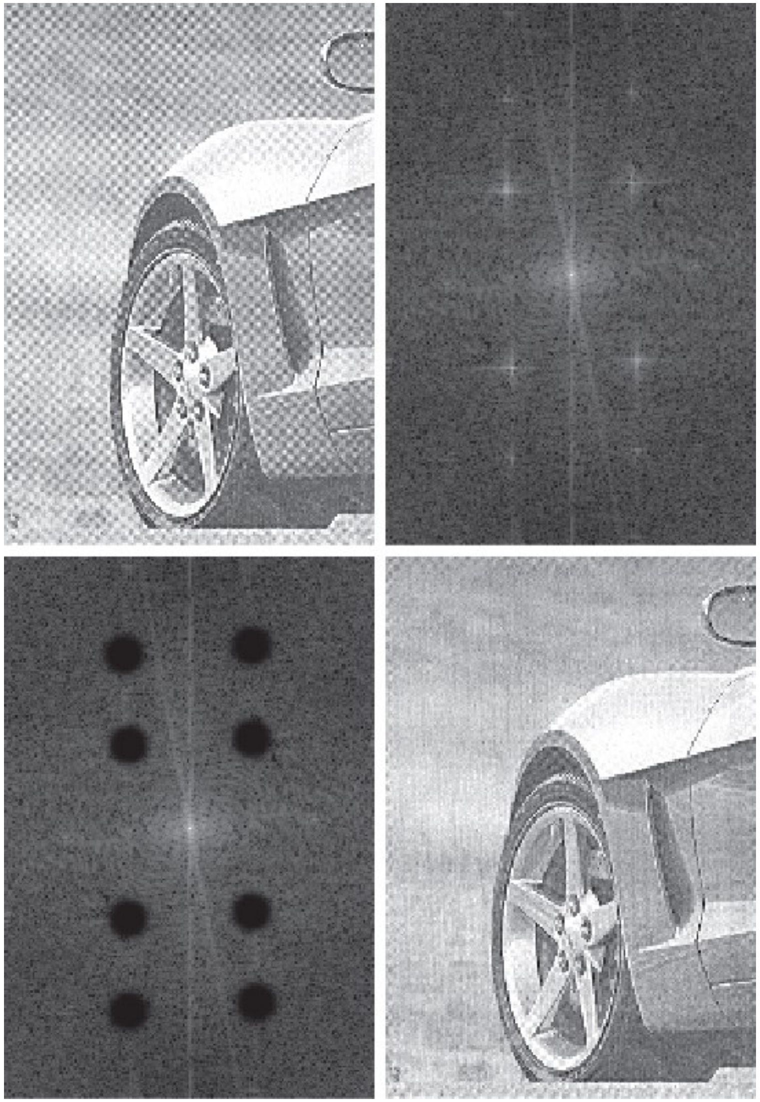
- Removal of periodic noise involves eliminating a specific range of frequencies from the image.
- This is typically achieved using bandstop filters. Conversely, bandpass filters are used to isolate specific frequency ranges.
- An ideal band reject filter is an example of a bandstop filter.

$$H(u,v)=\begin{cases}1 & \text{if } D(u,v)<D_0-\dfrac{W}{2}\\ 0 & \text{if } D_0-\dfrac{W}{2}\le D(u,v)\le D_0+\dfrac{W}{2}\\ 1 & \text{if } D(u,v)>D_0+\dfrac{W}{2}\end{cases}$$

# Band-Stop Filters



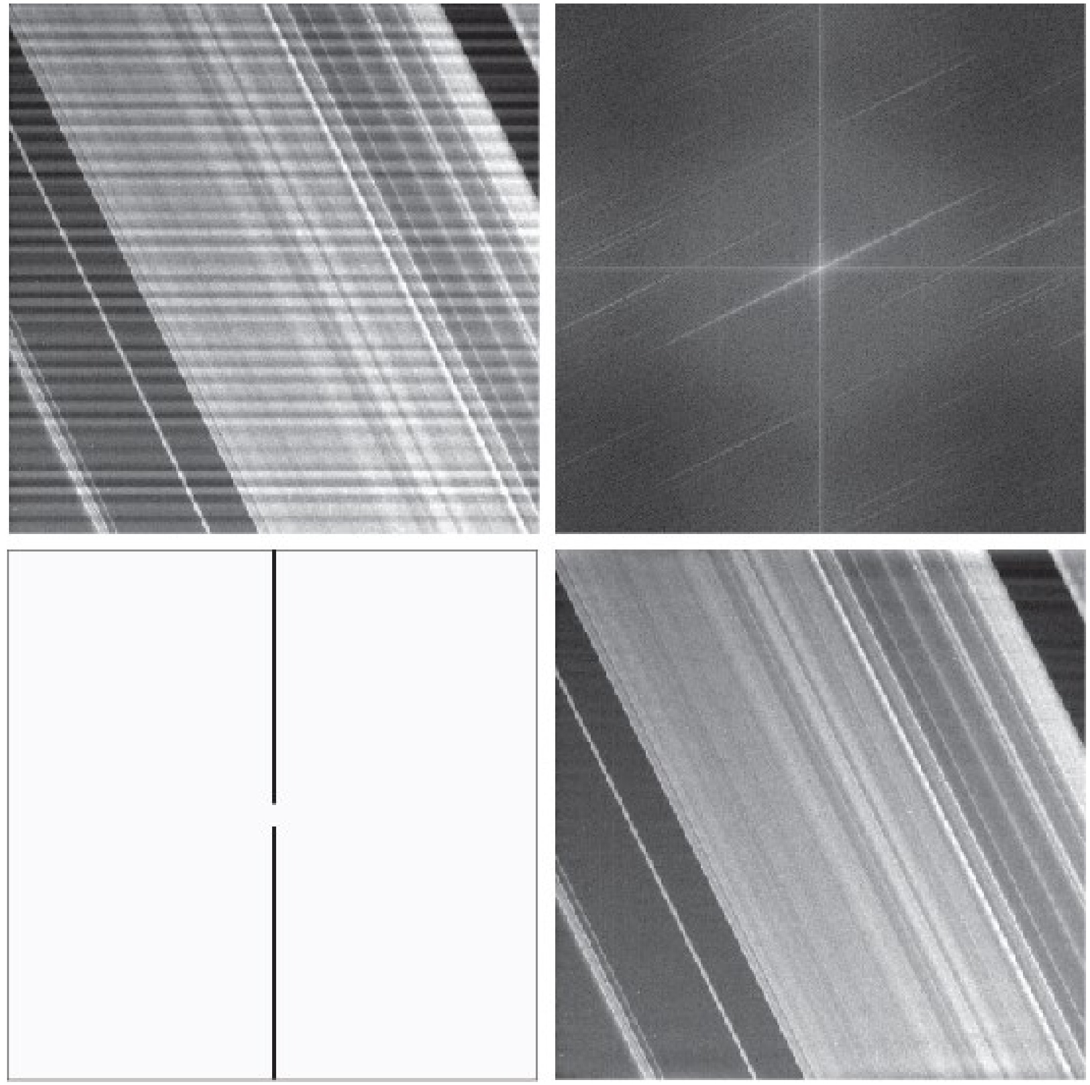
# Band-Stop Filters (cont...)



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.

# Band-Pass Filters (cont...)



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.)

# Fast Fourier Transform

- The reason that Fourier based techniques have become so popular is the development of the **Fast Fourier Transform (FFT)** algorithm.
- It allows the Fourier transform to be carried out in a reasonable amount of time.
- Reduces the complexity from $\mathrm{O}(N^4)$ to $\mathrm{O}(N^2 \log N^2)$.

# Frequency Domain Filtering & Spatial Domain Filtering

- Similar jobs can be done in the spatial and frequency domains.
- Filtering in the spatial domain can be easier to understand.
- Filtering in the frequency domain can be much faster – especially for large images.