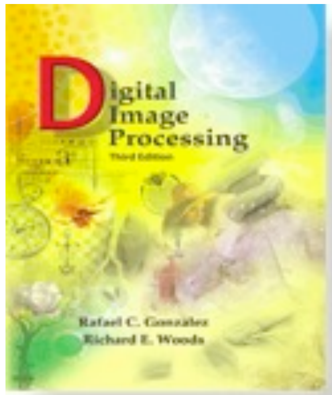


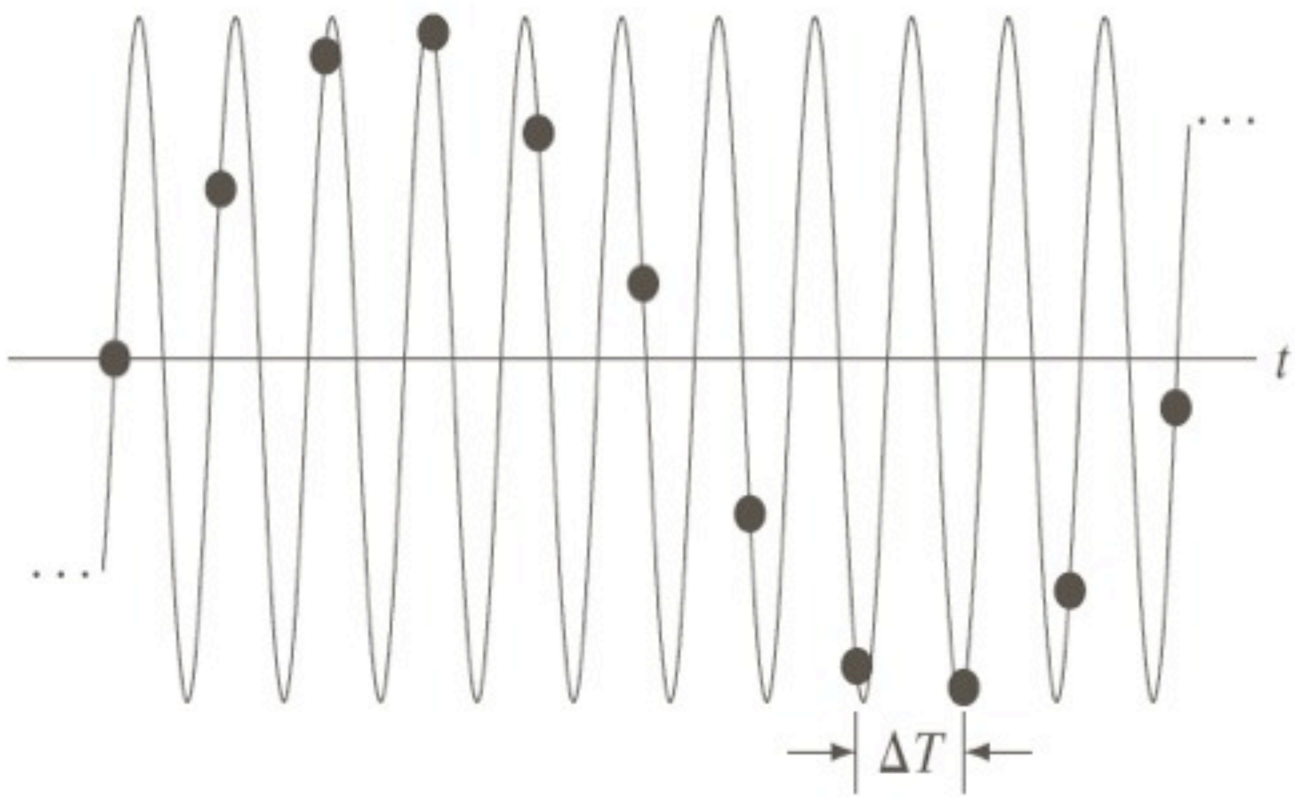
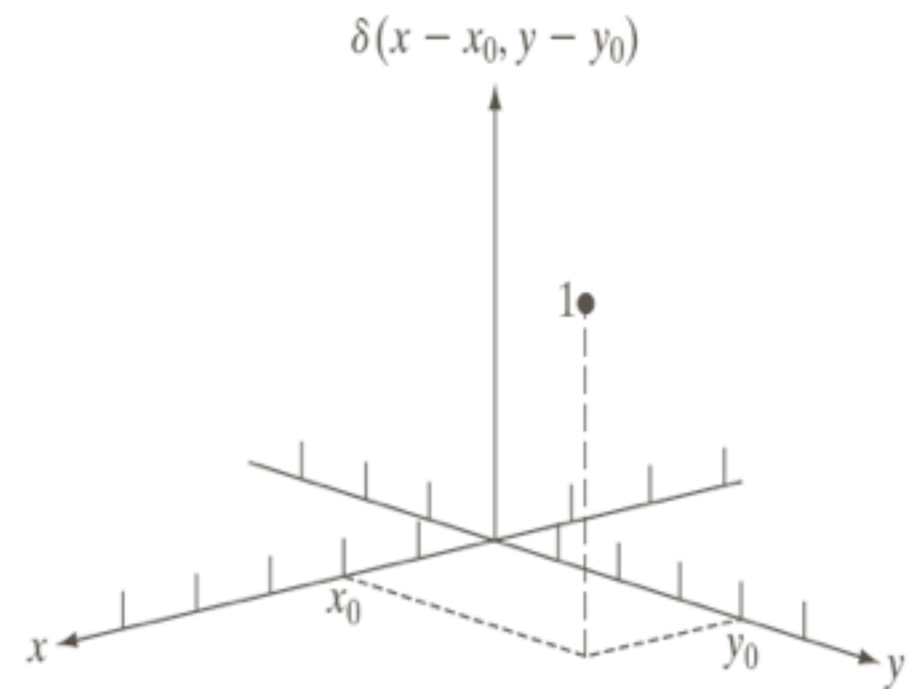


FIGURE 4.10 Illustration of aliasing. The under-sampled function (black dots) looks like a sine wave having a frequency much lower than the frequency of the continuous signal. The period of the sine wave is 2 s, so the zero crossings of the horizontal axis occur every second. ΔT is the separation between samples.

Extension to 2D

$$\delta(t - t_0, z - z_0) = \begin{cases} \infty \text{ for } t = t_0, z = z_0 \\ 0 \text{ otherwise} \end{cases}$$

$$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \delta(t - t_0, z - z_0) dt dz = 1$$



Sifting:

$$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(t, z) \delta(t - t_0, z - z_0) dt dz = f(t_0, z_0)$$

$$F(\mu,\nu)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(t,z)e^{-j2\pi(\mu t+\nu z)}dtdz$$

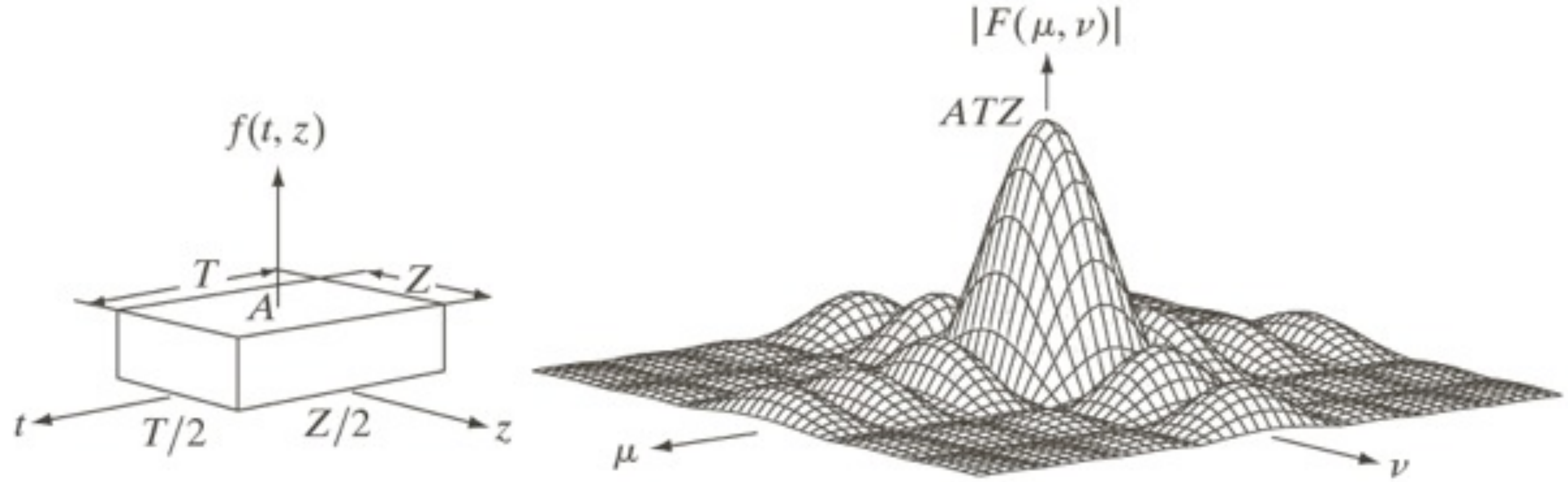
Fourier transform

$$f(x,y)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} F(\mu,\nu)e^{j2\pi(\mu t+\nu z)}d\mu d\nu$$

Inverse Fourier transform

For the rectangular impulse, the transform is a tensor product of two *sinc* functions

$$F(\mu, \nu) = ATZ\,\mathrm{sinc}(\mu T)\,\mathrm{sinc}(\nu Z)$$

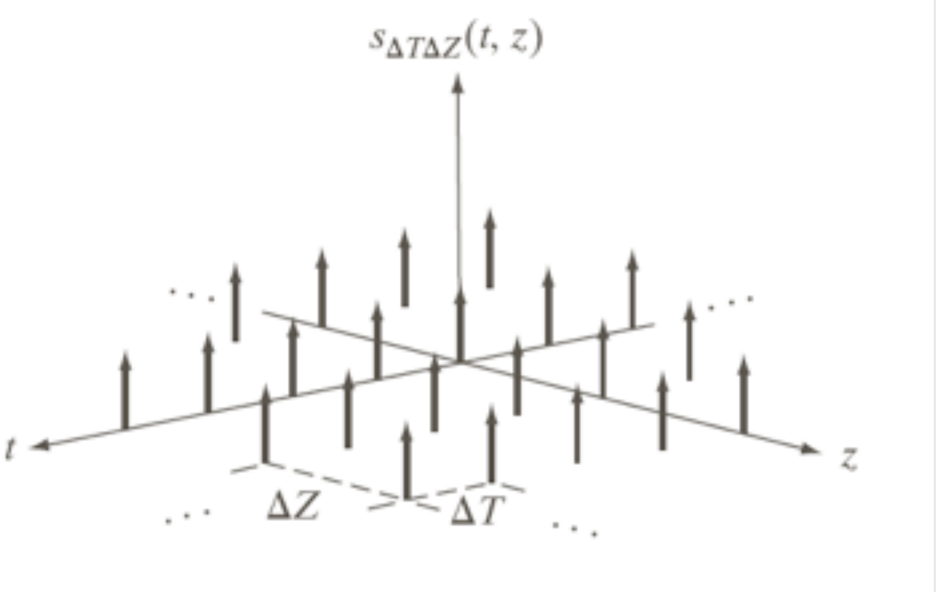


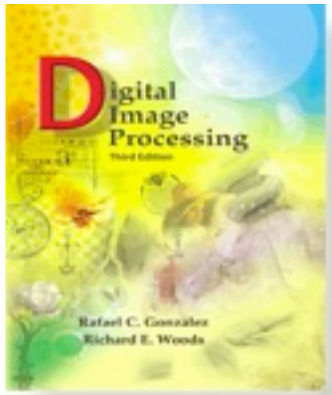
a b

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.

Dirac comb -> spike mat

$$s_{\Delta T \Delta Z}(t, z) = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} \delta(t - m\Delta T, z - n\Delta Z)$$



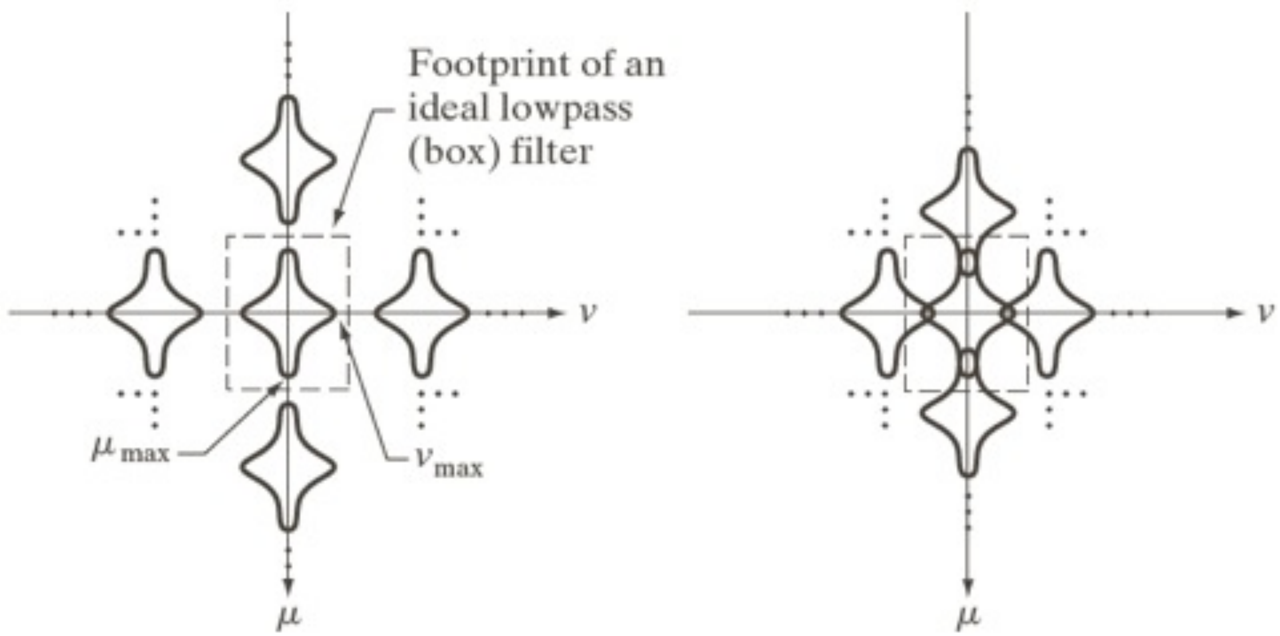


We say that $f(t,z)$ is band limited if the transform $F(\mu,\nu)=0$ outside the rectangle $[-\mu_{max}, \mu_{max}]$ and $[-\nu_{max}, \nu_{max}]$

2D Sampling Theorem: A continuous, band-limited 2D function can be recovered without error if the sampling intervals are

$$\Delta T < \frac{1}{2\mu_{max}}, \qquad \Delta Z < \frac{1}{2\nu_{max}}$$

Aliasing for images



a b

FIGURE 4.15 Two-dimensional Fourier transforms of (a) an over-sampled, and (b) under-sampled band-limited function.

The effect of aliasing in images gives artifacts like spurious highlights and patterns that are not present in the original image.

a b c

FIGURE 4.17 Illustration of aliasing on resampled images. (a) A digital image with negligible visual aliasing. (b) Result of resizing the image to 50% of its original size by pixel deletion. Aliasing is clearly visible. (c) Result of blurring the image in (a) with a 3×3 averaging filter prior to resizing. The image is slightly more blurred than (b), but aliasing is not longer objectionable. (Original image courtesy of the Signal Compression Laboratory, University of California, Santa Barbara.)

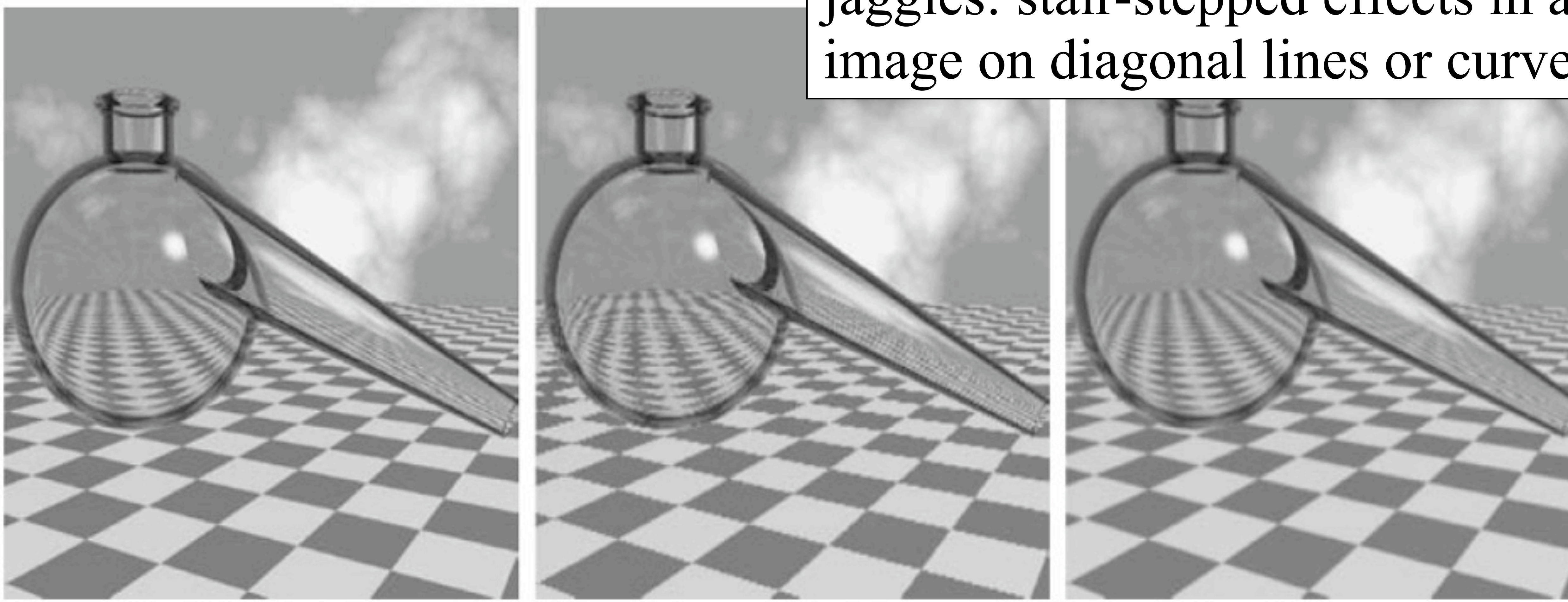
Often these patterns appear when downsampling an image (f.ex. by interpolation).

Zooming <=> oversampling
Shrinking <=> undersampling

The effect of aliasing can be attenuated by a slight defocusing (blurring), **prior** to the downsampling.

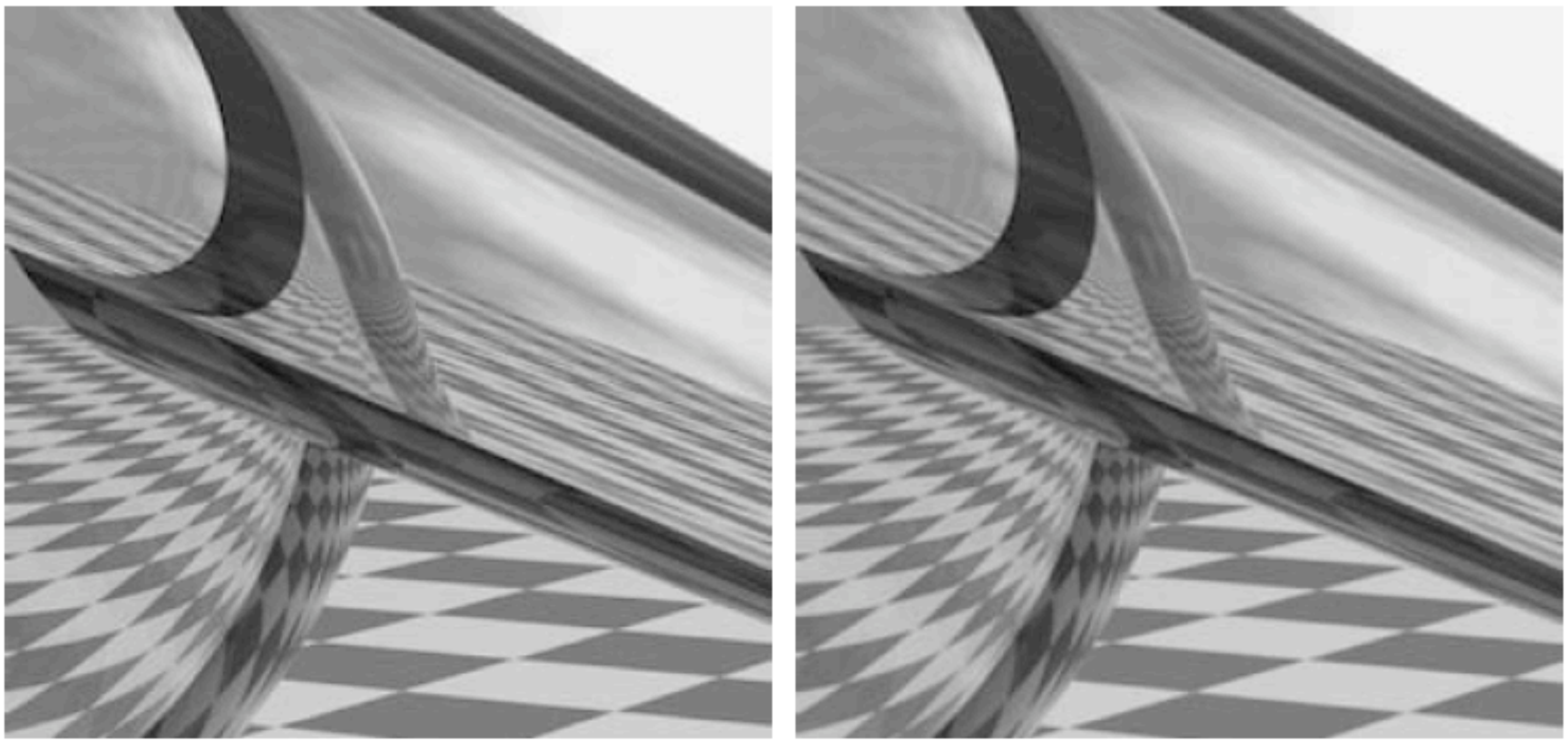
Such defocusing attenuates the high frequencies before the sampling.

jaggies: stair-stepped effects in an image on diagonal lines or curves



a b c

FIGURE 4.18 Illustration of jaggies. (a) A 1024 × 1024 digital image of a computer-generated scene with negligible visible aliasing. (b) Result of reducing (a) to 25% of its original size using bilinear interpolation. (c) Result of blurring the image in (a) with a 5 × 5 averaging filter prior to resizing it to 25% using bilinear interpolation. (Original image courtesy of D. P. Mitchell, Mental Landscape, LLC.)



a b

FIGURE 4.19 Image zooming. (a) A 1024 × 1024 digital image generated by pixel replication from a 256 × 256 image extracted from the middle of Fig. 4.18(a). (b) Image generated using bi-linear interpolation, showing a significant reduction in jaggies.

Another well known artifact: moiré (pron. morei) patterns

These are produced as “interference” patterns of two patterns with approximately the same spacing.

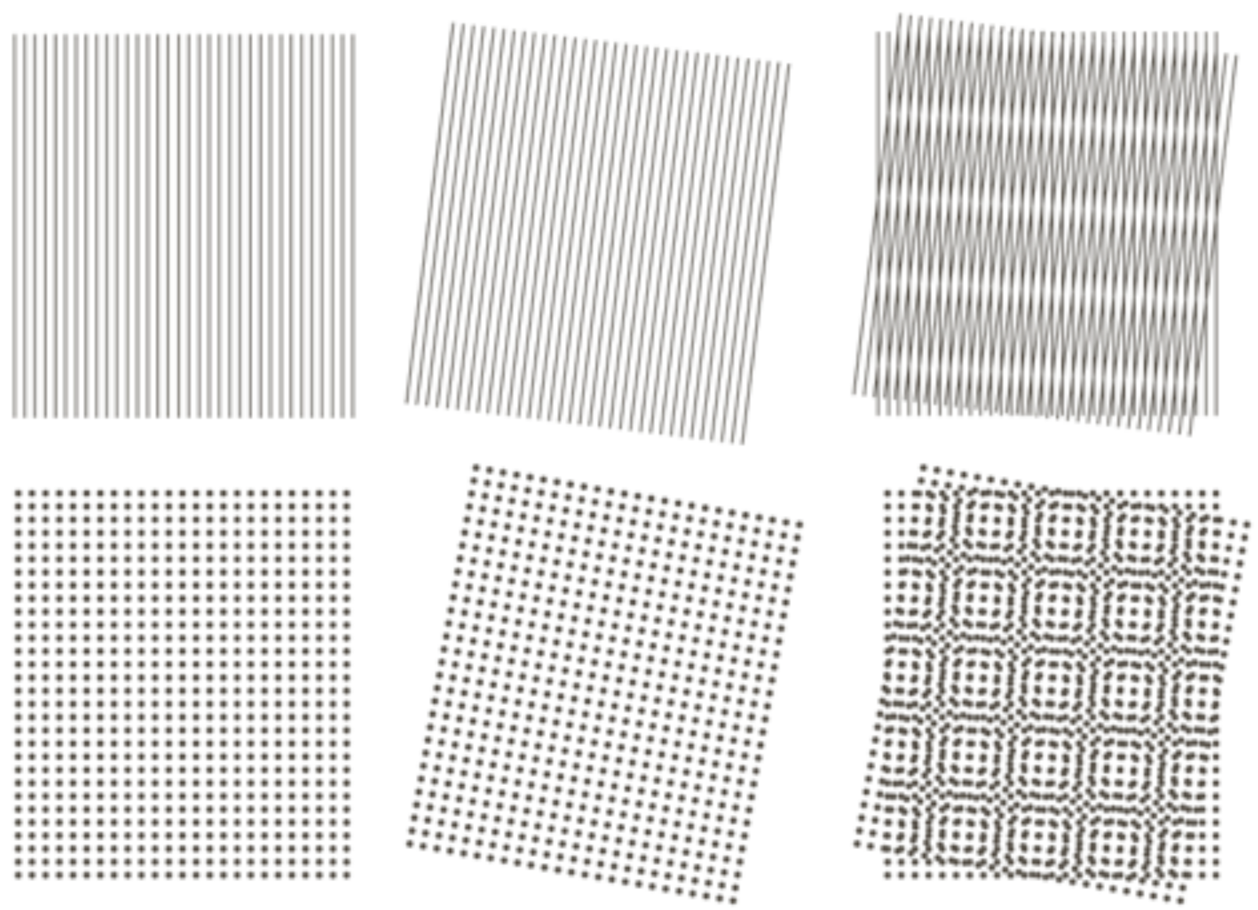
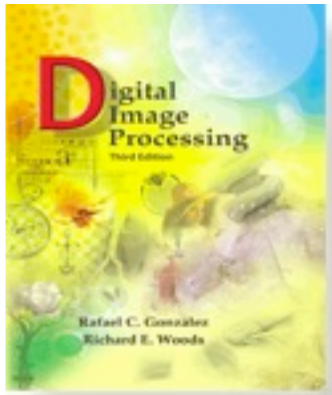


FIGURE 4.20 Examples of the moiré effect. These are ink drawings, not digitized patterns. Superimposing one pattern on the other is equivalent mathematically to multiplying the patterns.



Chapter 4
Filtering in the Frequency Domain

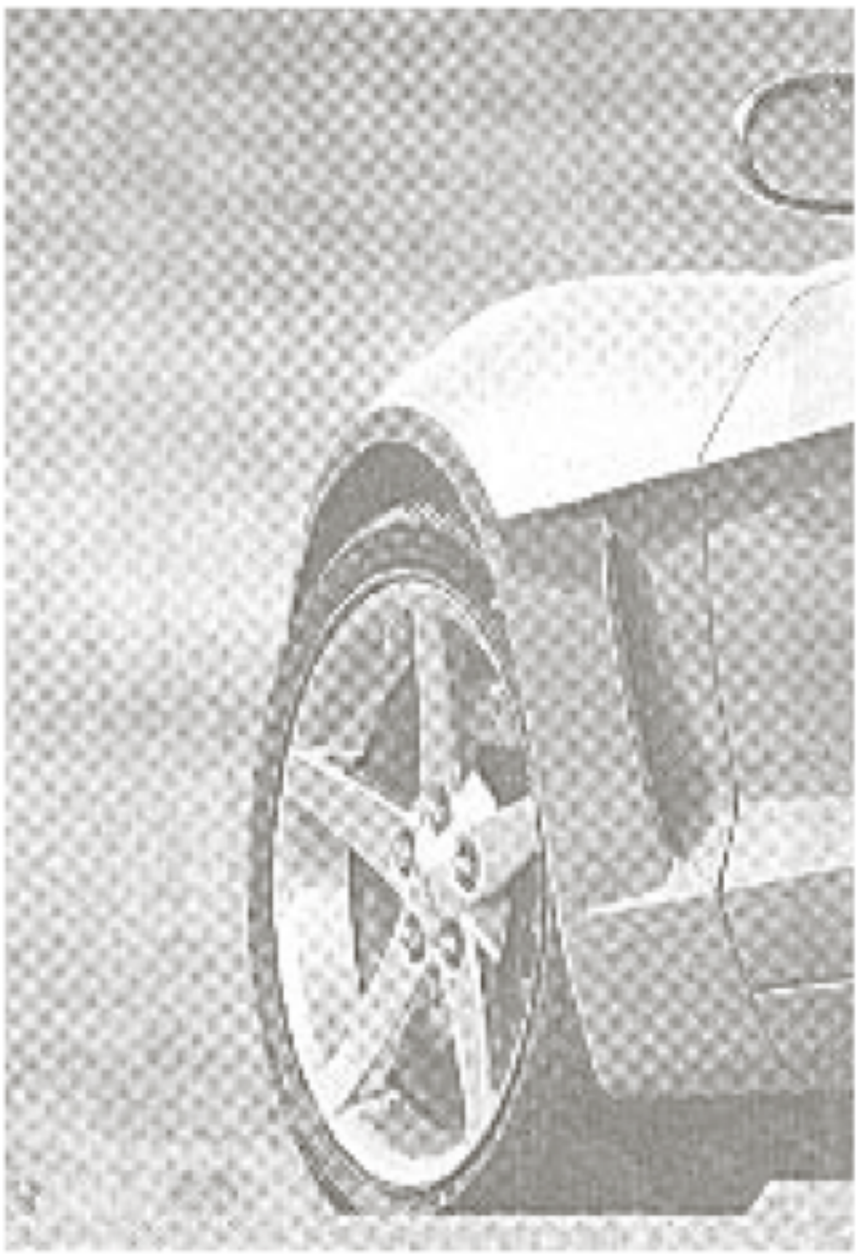


FIGURE 4.21 A newspaper image of size 246 × 168 pixels sampled at 75 dpi showing a moiré pattern. The moiré pattern in this image is the interference pattern created between the ±45° orientation of the halftone dots and the north–south orientation of the sampling grid used to digitize the image.

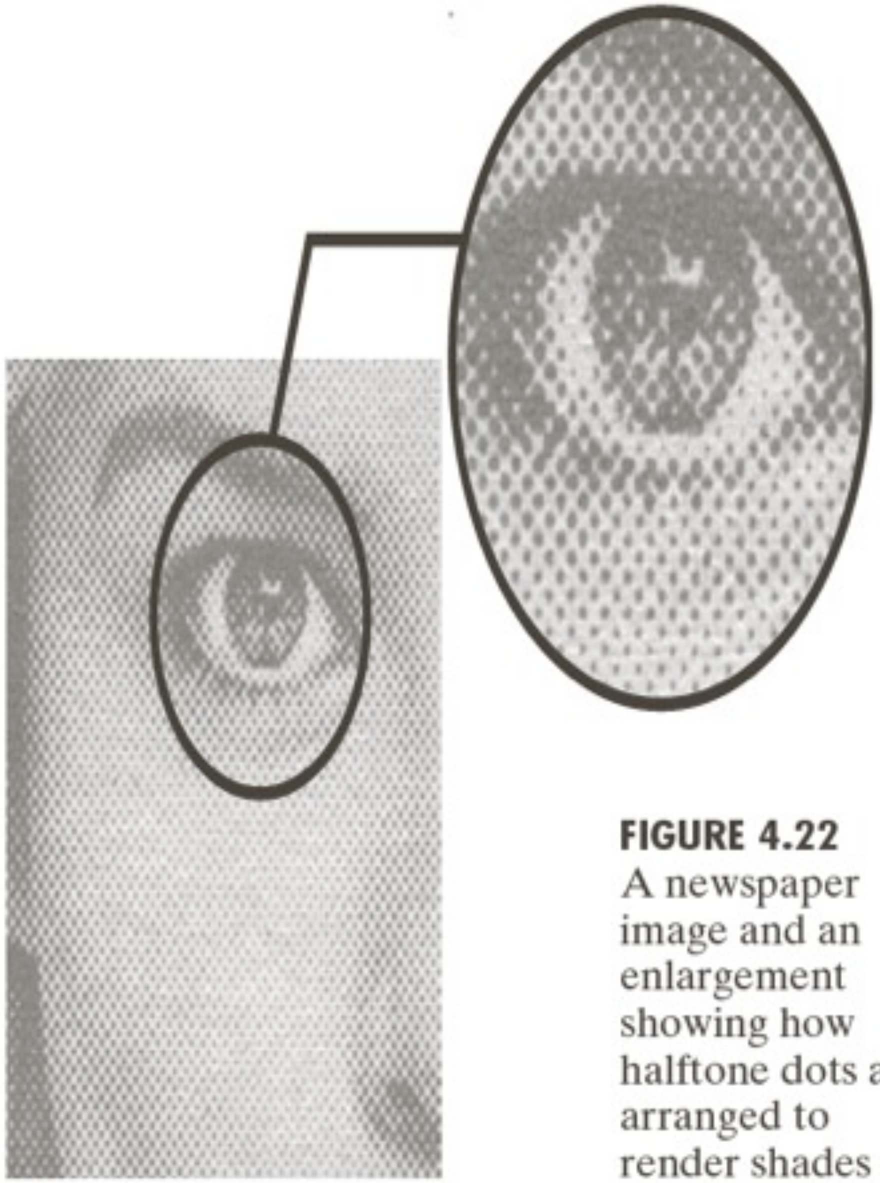
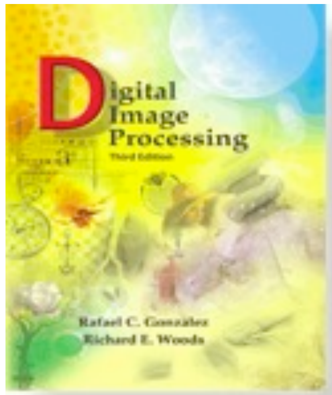


FIGURE 4.22 A newspaper image and an enlargement showing how halftone dots are arranged to render shades of gray.

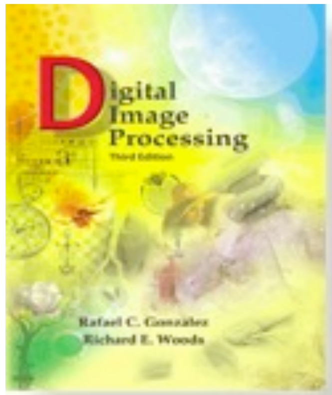


Discrete transforms:

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

Inverse discrete transform:

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

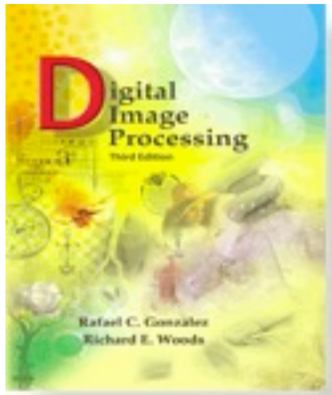


Note that

$$F(0,0)=\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y) = MN\bar{f}(x,y)$$

is the average of the image values multiplied by the size of the image.

This value can be much larger than the other components

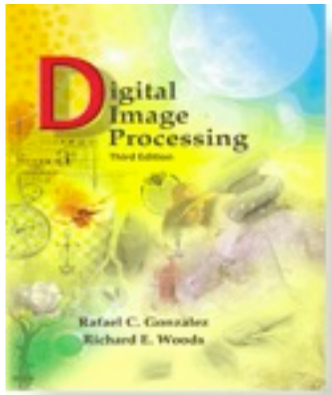


Translation and Rotation properties:

$$f(x-x_0, y-y_0) \Leftrightarrow F(u,v)e^{-j2\pi(x_0u/M+y_0v/N)}$$

$$F(u-u_0, v-v_0) \Leftrightarrow f(x,y)e^{j2\pi(xu_0/M+yv_0/N)}$$

Moreover, rotating f corresponds to rotating F of the same angle (and vice-versa)



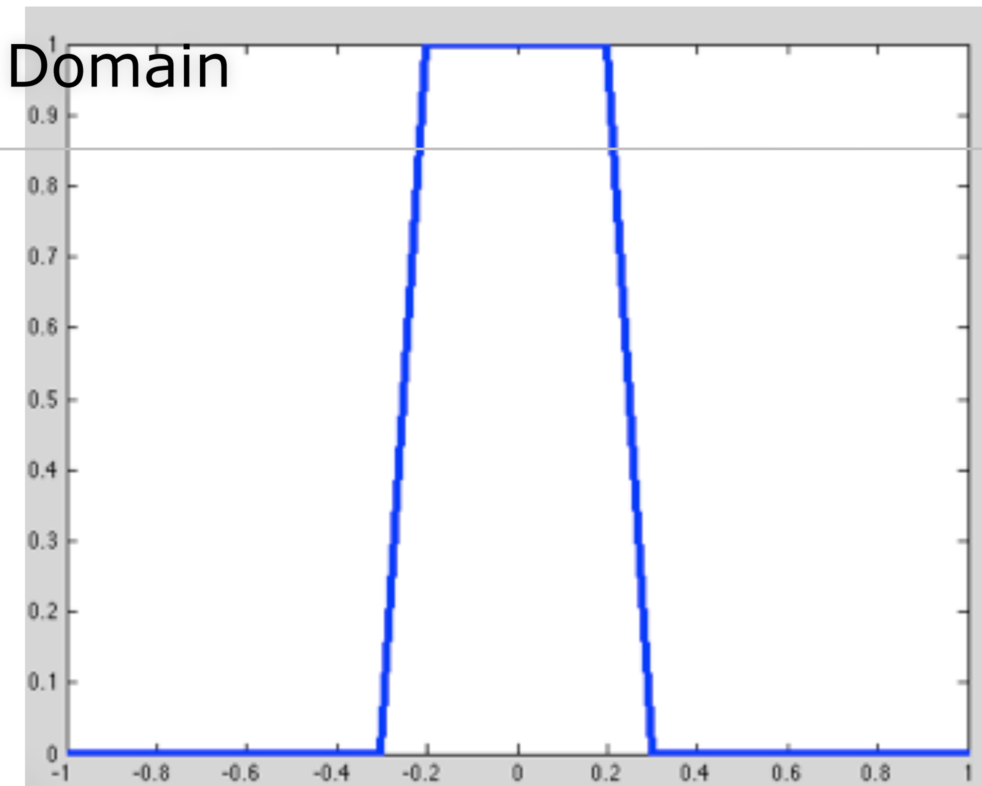
Periodicity:

for any integers k_1 and k_2:

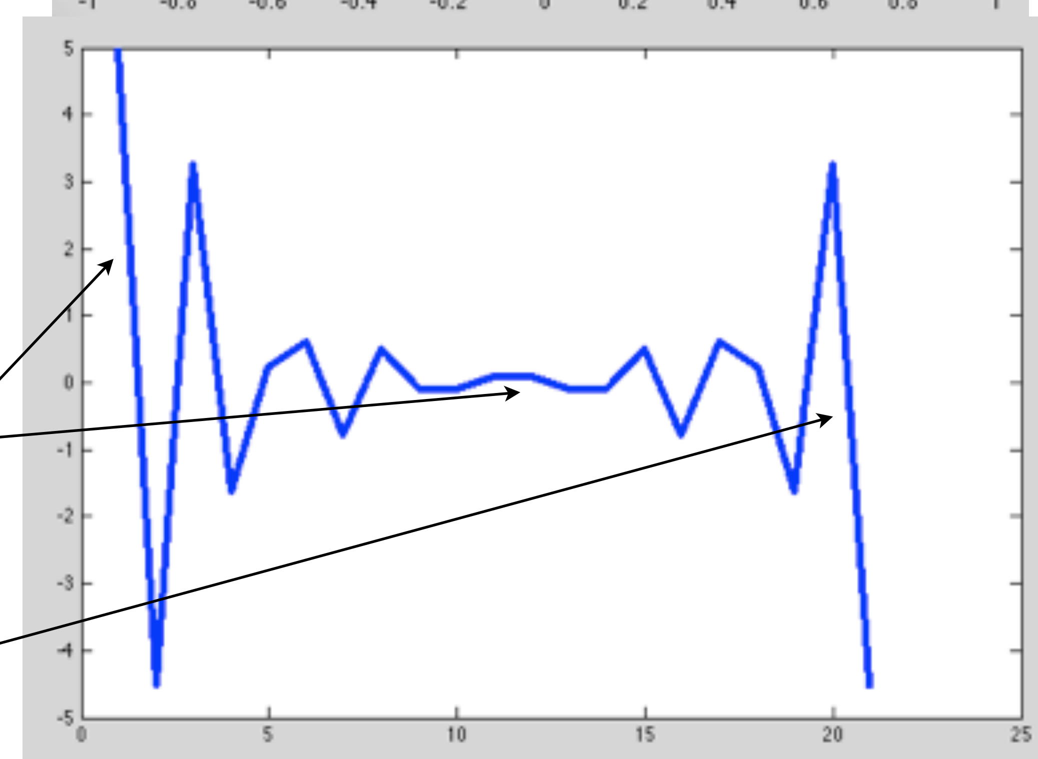
$$F(u + k_1 M, v + k_2 N) = F(u, v)$$
$$f(x + k_1 M, y + k_2 N) = f(x, y)$$

working with fft in matlab:

```
puls = rectpuls(-1:.1:1, .5);
```




```
figure, plot(real(fft(puls)))
```



high frequencies

Low frequencies

rectangular impulse <-> sinc function

To see the sinc function, we have to wrap around in order to move the low frequencies to the middle of the interval

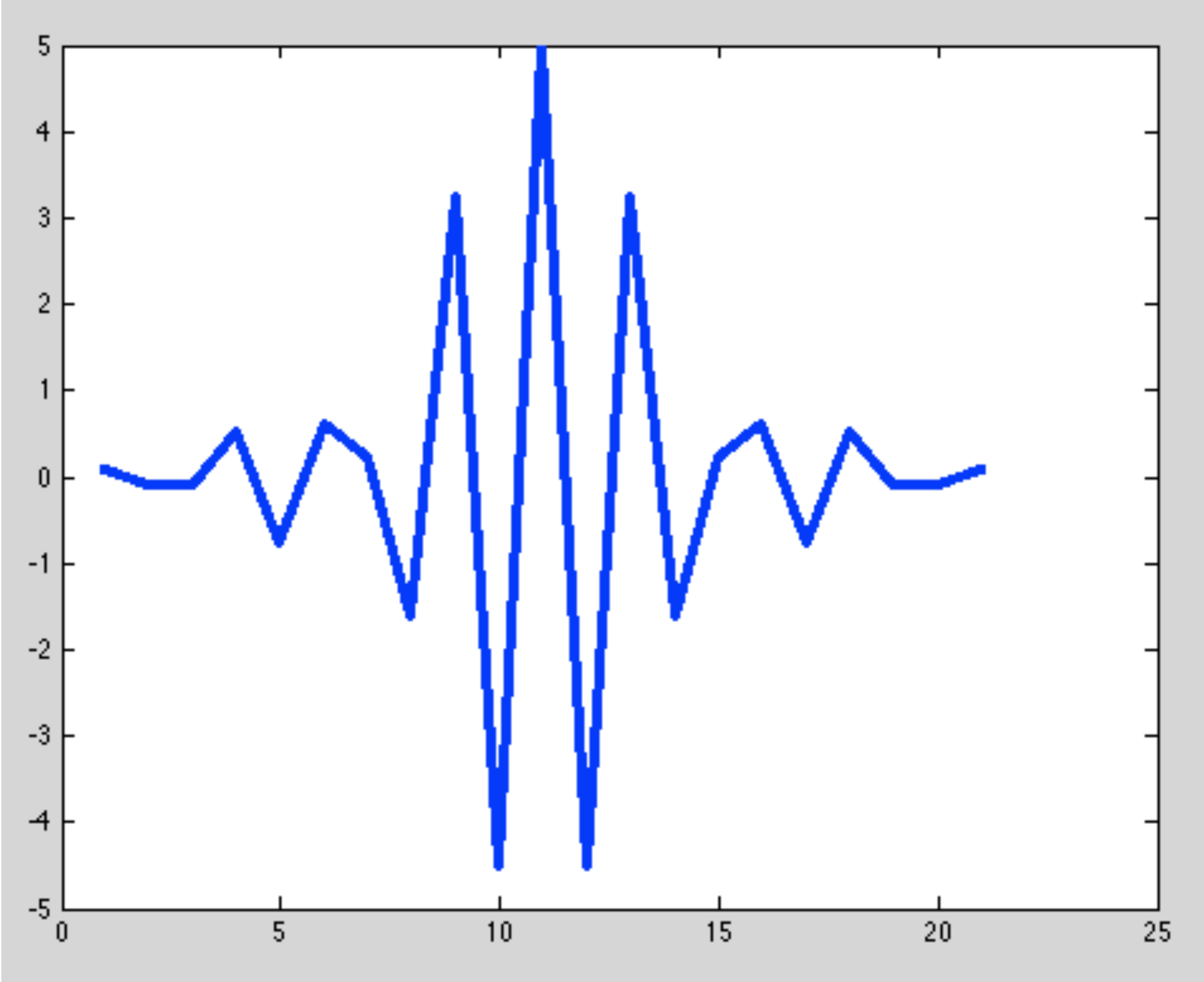
```
fftshift
```

rectangular impulse <-> sinc function

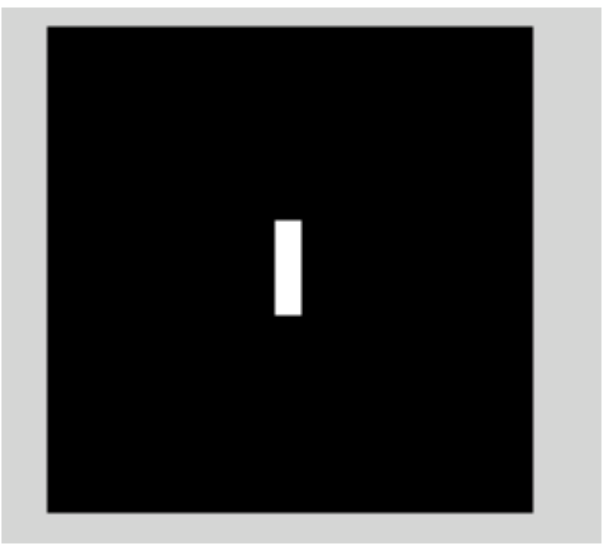
To see the sinc function, we have to wrap around in order to move the low frequencies to the middle of the interval

fftshift

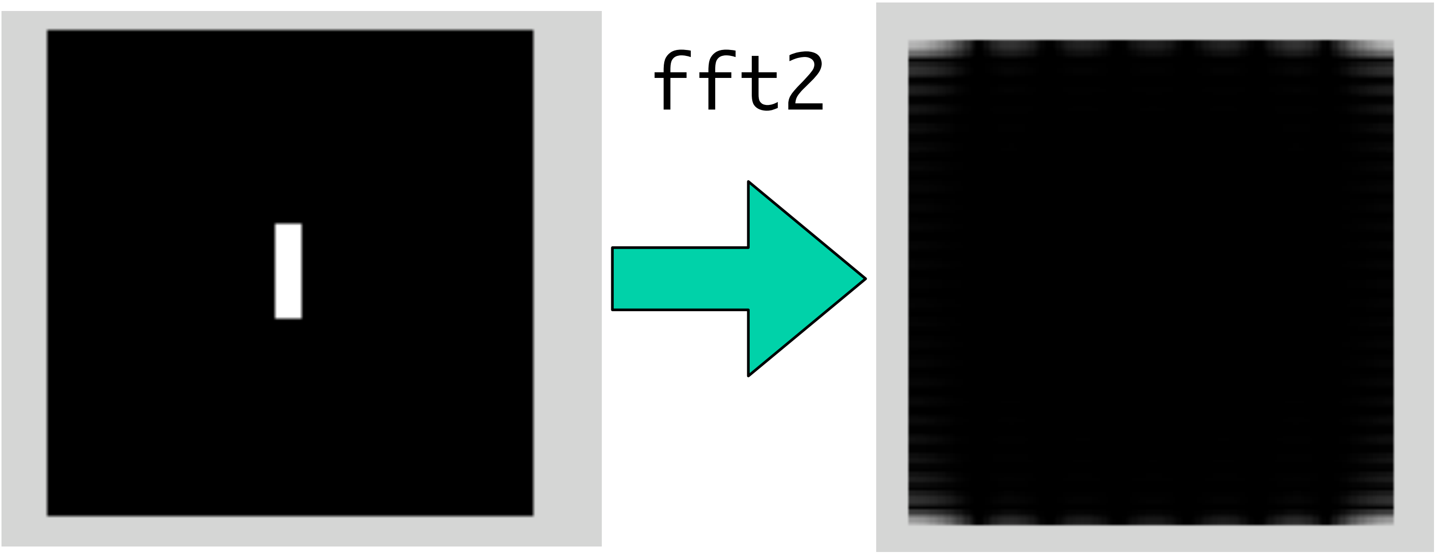
```
>> figure, plot(fftshift(real(fft
(puls))),…
'LineWidth', 4)
```



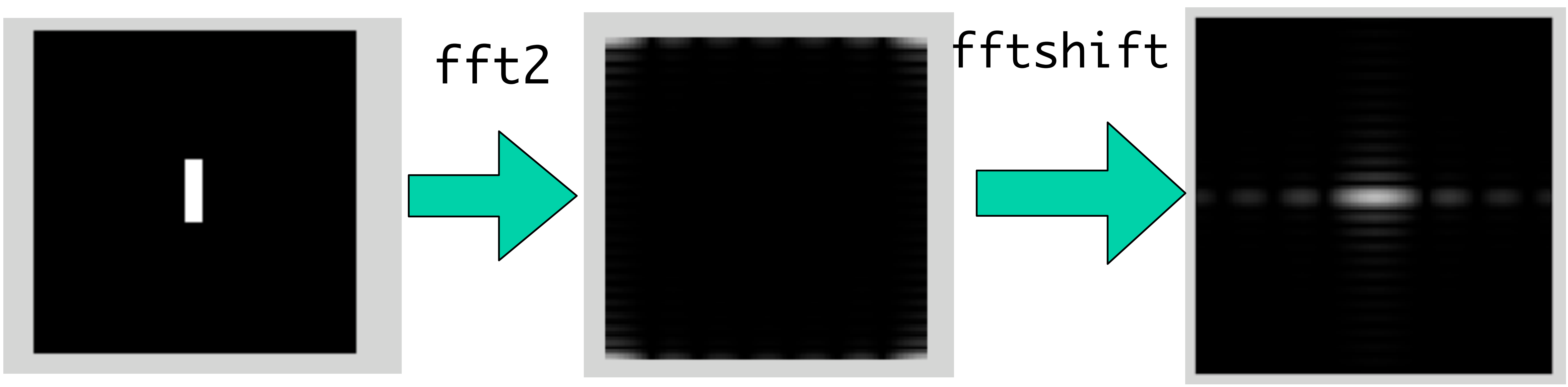
The same happens in 2 dimension:



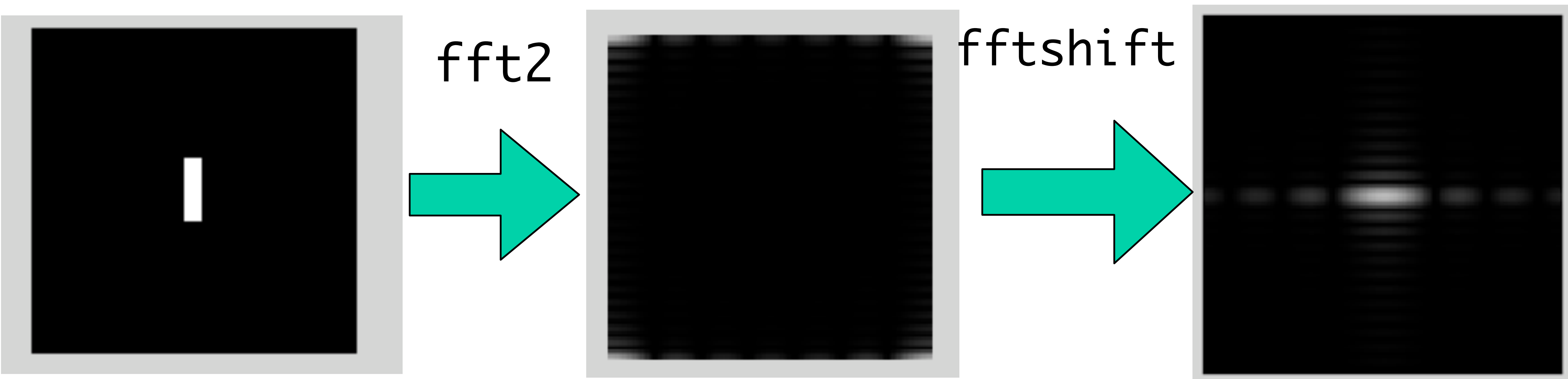
The same happens in 2 dimension:



The same happens in 2 dimension:

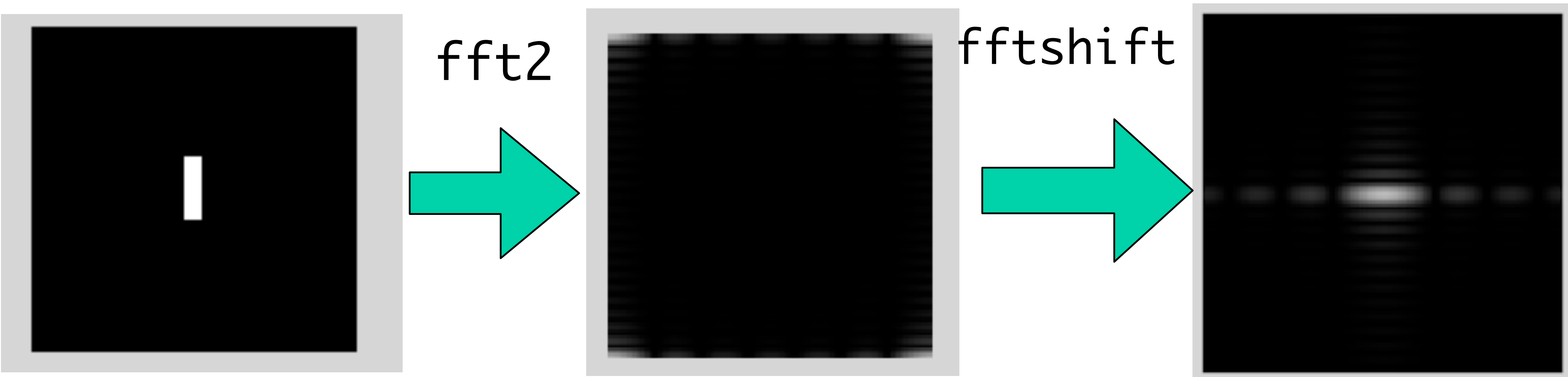


The same happens in 2 dimension:

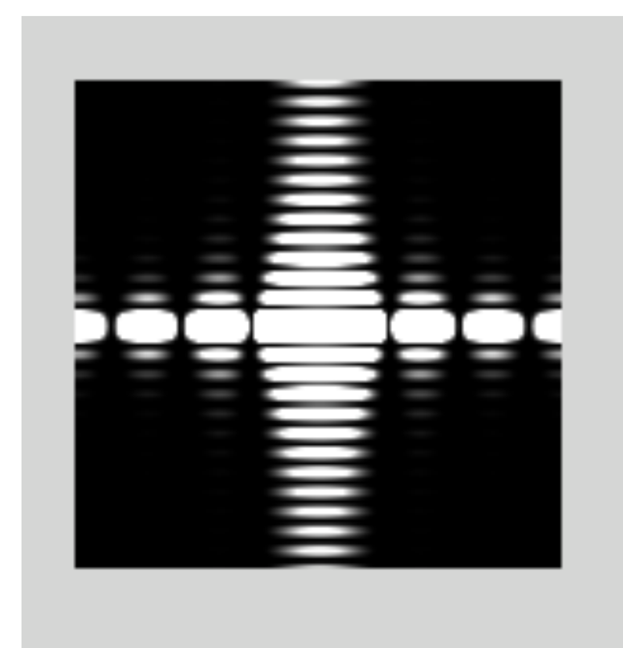


The same effect can be achieved by
f(x,y) -> f(x,y) (-1)^(x+y)

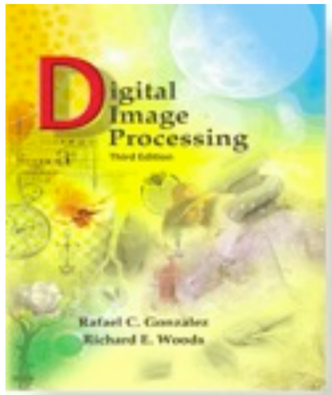
The same happens in 2 dimension:



The same effect can be achieved by
f(x,y) -> f(x,y) (-1)^(x+y)

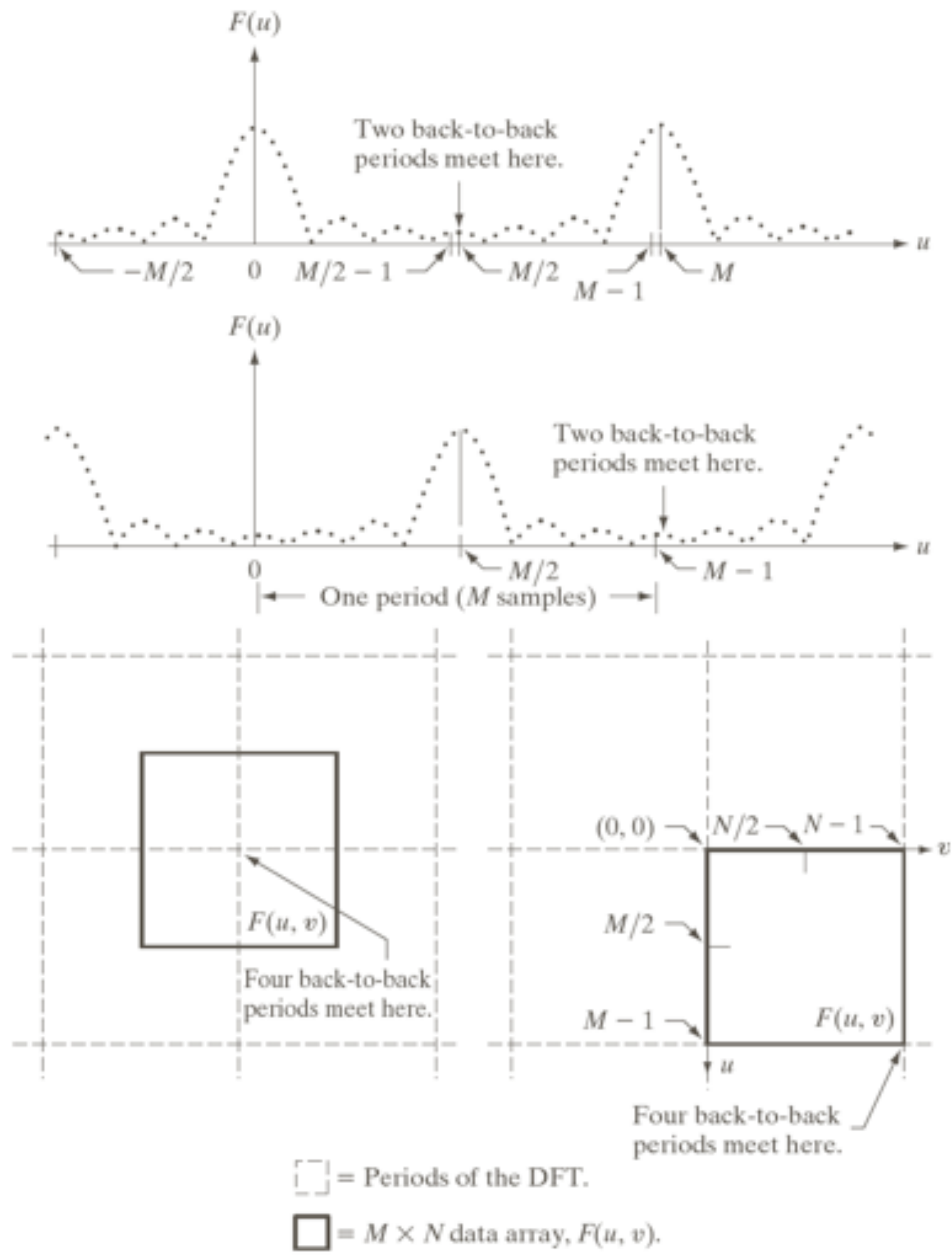


Here we have applied a power tranformation
to highlight the details.



Chapter 4

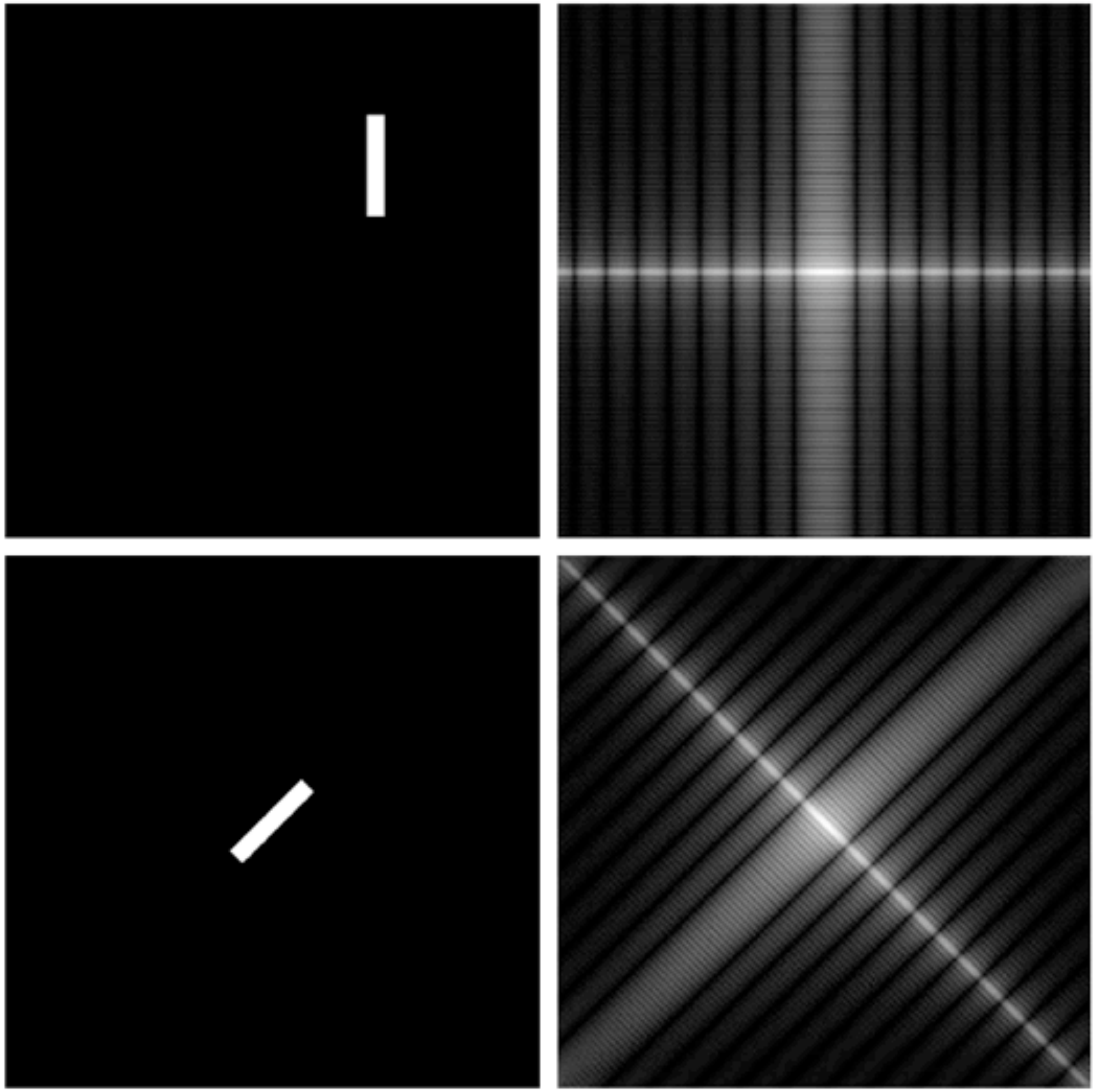
Filtering in the Frequency Domain



a
b
c d

FIGURE 4.23 Centering the Fourier transform. (a) A 1-D DFT showing an infinite number of periods. (b) Shifted DFT obtained by multiplying $f(x)$ by $(-1)^x$ before computing $F(u)$. (c) A 2-D DFT showing an infinite number of periods. The solid area is the $M \times N$ data array, $F(u, v)$, obtained with Eq. (4.5-15). This array consists of four quarter periods. (d) A Shifted DFT obtained by multiplying $f(x, y)$ by $(-1)^{x+y}$ before computing $F(u, v)$. The data now contains one complete, centered period, as in (b).

Chapter 4
Filtering in the Frequency Domain



a b
c d

FIGURE 4.25 (a) The rectangle in Fig. 4.24(a) translated, and (b) the corresponding spectrum. (c) Rotated rectangle, and (d) the corresponding spectrum. The spectrum corresponding to the translated rectangle is identical to the spectrum corresponding to the original image in Fig. 4.24(a).

Chapter 4
Filtering in the Frequency Domain

	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

[†]Recall that x, y, u, and v are *discrete* (integer) variables, with x and u in the range $[0, M - 1]$, and y, and v in the range $[0, N - 1]$. To say that a complex function is *even* means that its real *and* imaginary parts are even, and similarly for an odd complex function.

TABLE 4.1 Some symmetry properties of the 2-D DFT and its inverse. $R(u, v)$ and $I(u, v)$ are the real and imaginary parts of $F(u, v)$, respectively. The term *complex* indicates that a function has nonzero real and imaginary parts.

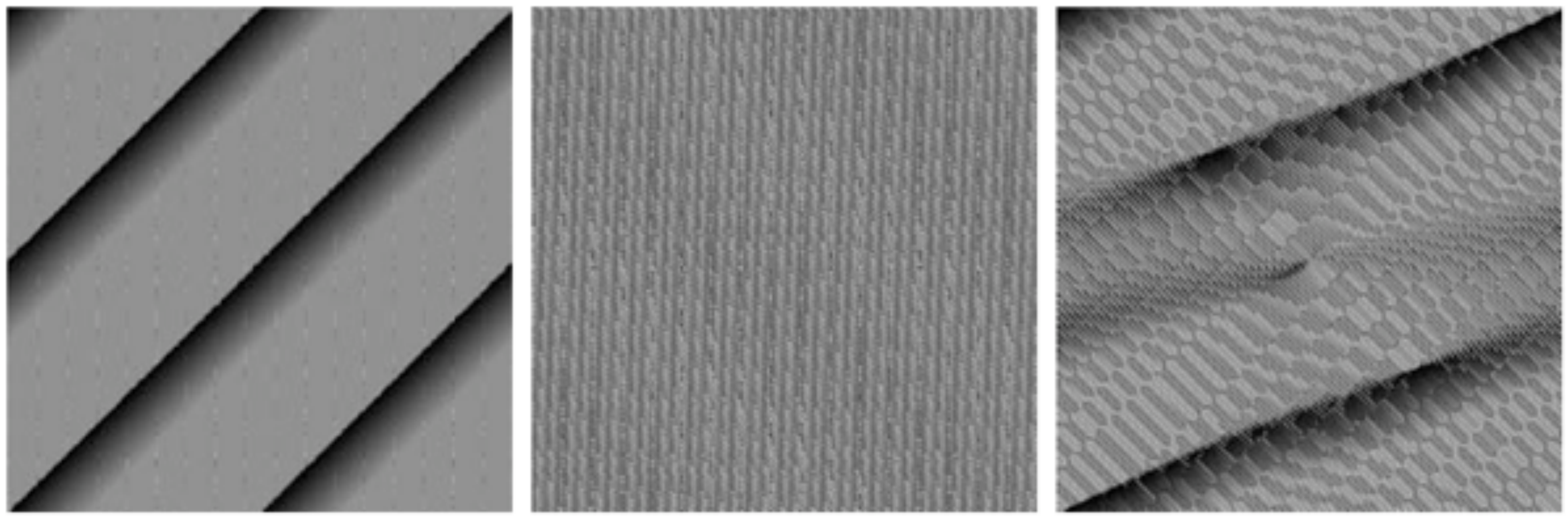
Spectrum and phase angle:

$$F(u,v) = |F(u,v)|e^{j\phi(u,v)}$$

Spectrum

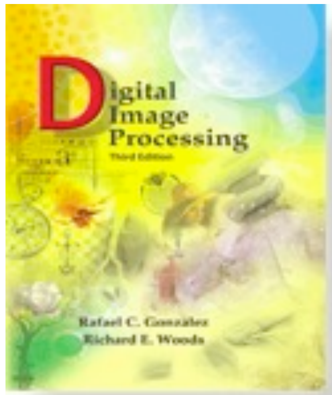
phase angle

Power spectrum = spectrum^2



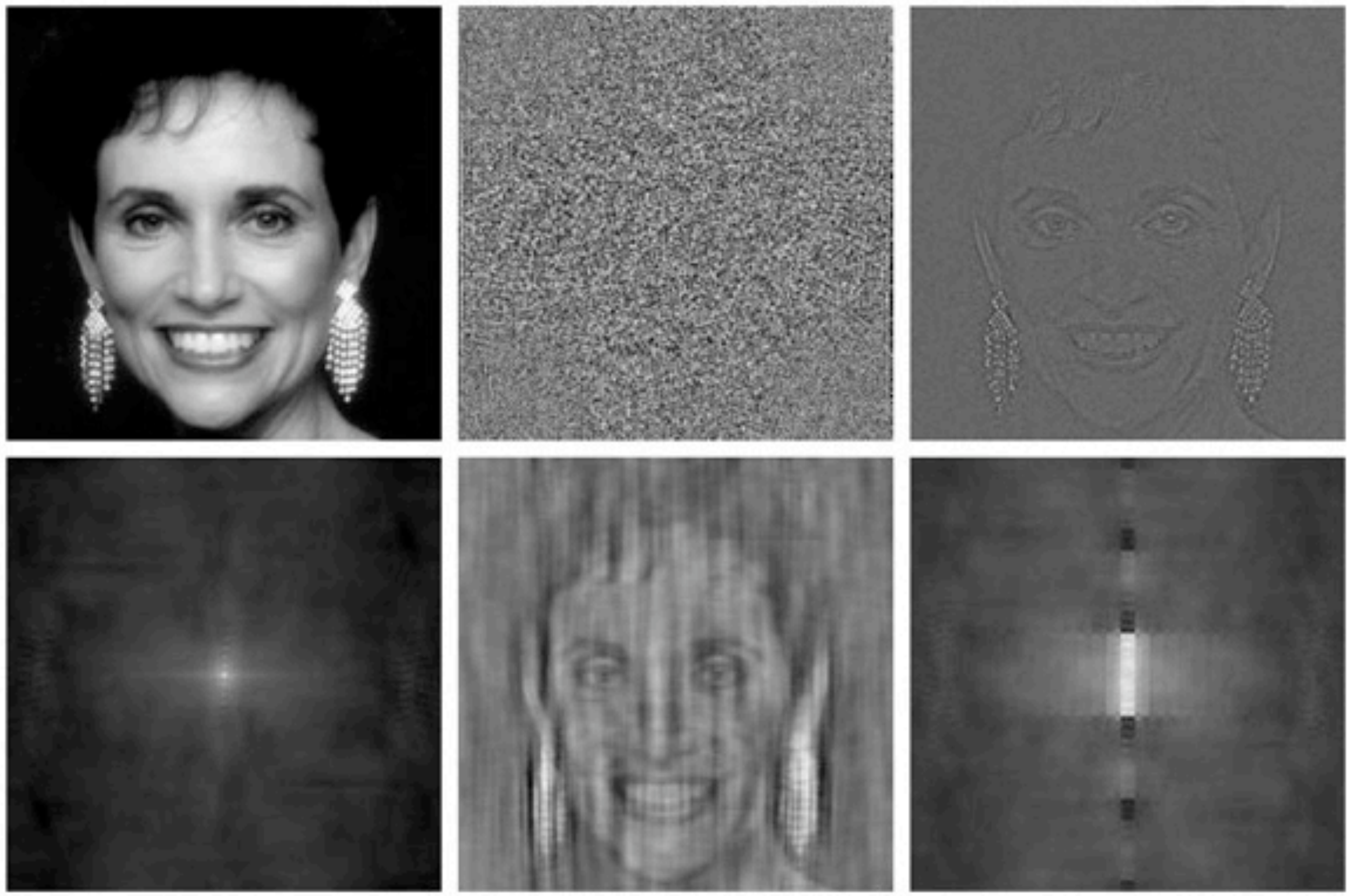
a b c

FIGURE 4.26 Phase angle array corresponding (a) to the image of the centered rectangle in Fig. 4.24(a), (b) to the translated image in Fig. 4.25(a), and (c) to the rotated image in Fig. 4.25(c).



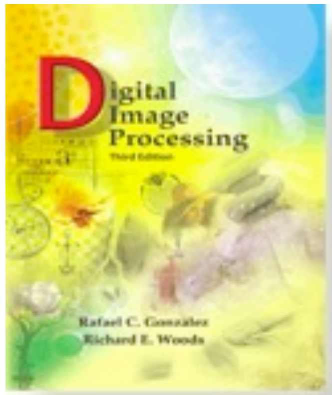
Chapter 4

Filtering in the Frequency Domain



a	b	c
d	e	f

FIGURE 4.27 (a) Woman. (b) Phase angle. (c) Woman reconstructed using only the phase angle. (d) Woman reconstructed using only the spectrum. (e) Reconstruction using the phase angle corresponding to the woman and the spectrum corresponding to the rectangle in Fig. 4.24(a). (f) Reconstruction using the phase of the rectangle and the spectrum of the woman.



Chapter 4

Filtering in the Frequency Domain

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) Polar representation	$F(u, v) = \lvert F(u, v) \rvert e^{j\phi(u,v)}$
4) Spectrum	$\lvert F(u, v) \rvert = \left[R^2(u, v) + I^2(u, v) \right]^{1/2}$ $R = \text{Real}(F); \quad I = \text{Imag}(F)$
5) Phase angle	$\phi(u, v) = \tan^{-1} \left[\frac{I(u, v)}{R(u, v)} \right]$
6) Power spectrum	$P(u, v) = \lvert F(u, v) \rvert^2$
7) Average value	$\bar{f}(x, y) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) = \frac{1}{MN} F(0, 0)$

(Continued)

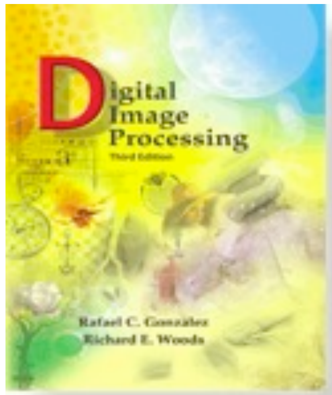


TABLE 4.3
(Continued)

Name	DFT Pairs
7) Correlation theorem[†]	$f(x, y) \star h(x, y) \Leftrightarrow F^*(u, v)\,H(u, v)$ $f^*(x, y)h(x, y) \Leftrightarrow F(u, v) \star H(u, v)$
8) Discrete unit impulse	$\delta(x, y) \Leftrightarrow 1$
9) Rectangle	$\text{rect}[a, b] \Leftrightarrow ab\dfrac{\sin(\pi ua)}{(\pi ua)}\dfrac{\sin(\pi vb)}{(\pi vb)}e^{-j\pi(ua+vb)}$
10) Sine	$\sin(2\pi u_0 x + 2\pi v_0 y) \Leftrightarrow$ $j\dfrac{1}{2}\left[\delta(u + Mu_0, v + Nv_0) - \delta(u - Mu_0, v - Nv_0)\right]$
11) Cosine	$\cos(2\pi u_0 x + 2\pi v_0 y) \Leftrightarrow$ $\dfrac{1}{2}\left[\delta(u + Mu_0, v + Nv_0) + \delta(u - Mu_0, v - Nv_0)\right]$
The following Fourier transform pairs are derivable only for continuous variables, denoted as before by t and z for spatial variables and by μ and ν 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 $f(\pm\infty, \pm\infty) = 0$.)	$\left(\dfrac{\partial}{\partial t}\right)^m\left(\dfrac{\partial}{\partial z}\right)^n f(t, z) \Leftrightarrow (j2\pi\mu)^m(j2\pi\nu)^n F(\mu, \nu)$ $\dfrac{\partial^m f(t, z)}{\partial t^m} \Leftrightarrow (j2\pi\mu)^m F(\mu, \nu); \dfrac{\partial^n f(t, z)}{\partial z^n} \Leftrightarrow (j2\pi\nu)^n F(\mu, \nu)$
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)

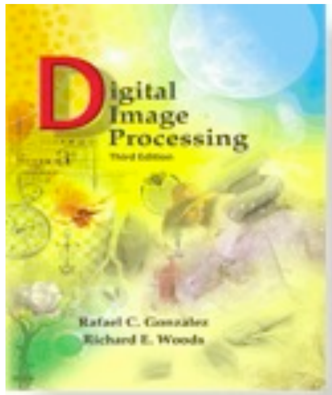
[†] Assumes that the functions have been extended by zero padding. Convolution and correlation are associative, commutative, and distributive.

Chapter 4

Filtering in the Frequency Domain

Name	DFT Pairs
1) Symmetry properties	See Table 4.1
2) Linearity	$af_1(x, y) + bf_2(x, y) \Leftrightarrow aF_1(u, v) + bF_2(u, v)$
3) Translation (general)	$f(x, y)e^{j2\pi(u_0x/M + v_0y/N)} \Leftrightarrow F(u - u_0, v - v_0)$ $f(x - x_0, y - y_0) \Leftrightarrow F(u, v)e^{-j2\pi(ux_0/M + vy_0/N)}$
4) Translation to center of the frequency rectangle, $(M/2, N/2)$	$f(x, y)(-1)^{x+y} \Leftrightarrow F(u - M/2, v - N/2)$ $f(x - M/2, y - N/2) \Leftrightarrow F(u, v)(-1)^{u+v}$
5) Rotation	$f(r, \theta + \theta_0) \Leftrightarrow F(\omega, \varphi + \theta_0)$ $x = r\cos\theta \quad y = r\sin\theta \quad u = \omega\cos\varphi \quad v = \omega\sin\varphi$
6) Convolution theorem[†]	$f(x, y) \star h(x, y) \Leftrightarrow F(u, v)H(u, v)$ $f(x, y)h(x, y) \Leftrightarrow F(u, v) \star H(u, v)$

(Continued)



Chapter 4

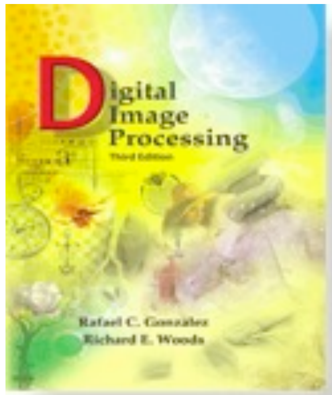
Filtering in the Frequency Domain

Name	Expression(s)
8) Periodicity (k_1 and k_2 are integers)	$F(u, v) = F(u + k_1M, v) = F(u, v + k_2N)$ $= F(u + k_1M, v + k_2N)$ $f(x, y) = f(x + k_1M, y) = f(x, y + k_2N)$ $= f(x + k_1M, y + k_2N)$
9) Convolution	$f(x, y) \bigstar h(x, y) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m, n)h(x - m, y - n)$
10) Correlation	$f(x, y) \star h(x, y) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f^*(m, n)h(x + m, y + n)$
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 4.11.1.
12) Obtaining the inverse Fourier transform using a forward transform algorithm.	$MNf^*(x, y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F^*(u, v)e^{-j2\pi(ux/M + vy/N)}$ 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 Section 4.11.2.

Filtering in the frequency domain

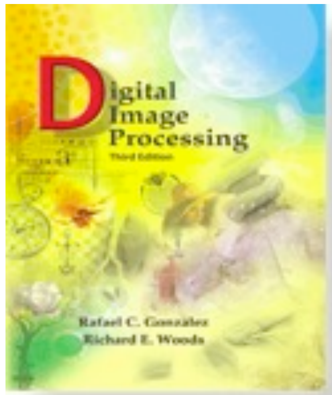
$$g(x, y) = \mathcal{F}^{-1}[H(u, v)F(u, v)]$$

Filter function


Digital
Image
Processing
Rafael C. Gonzalez
Richard E. Woods

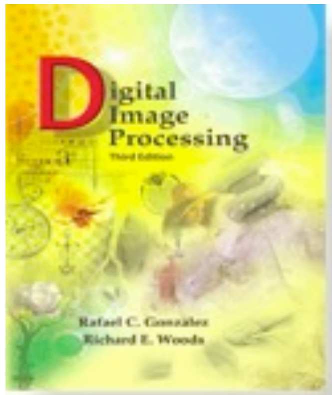
Lowpass filters:

Highpass filters:



Lowpass filters: These are filters that allow only the low frequencies and take out the high frequencies

Highpass filters: These are filters that allow only the high frequencies and take out the low frequencies



Lowpass filters: These are filters that allow only the low frequencies and take out the high frequencies

low frequencies: smoothing

Highpass filters: These are filters that allow only the high frequencies and take out the low frequencies

high frequencies: sharpening

We consider real filters $H(u,v)$

$$
\begin{aligned}
G(x,y) &= H(u,v)\,(R(u,v) + j\,I(u,v)) \\
&= H(u,v)\,R(u,v) + j\,H(u,v)\,I(u,v)
\end{aligned}
$$

The phase angle, $arctan(H(u,v)I(u,v) \,/\, H(u,v)\,F(u,v))$ is unchanged!

(zero phase shift).

Main steps of filtering in the frequency domain:

Assume $f(x,y)$ (image), $h(x,y)$ (filter), in the spatial domain given:
1. Pad $f(x,y)$ to obtain $fp(x,y)$, and similarly for h, hp
2. Compute dft of fp, hp (matlab: fft2)
3. Center the transforms (matlab: fftshift), Fp, Hp
4. Multiply $G(x,y) = Fp(x,y)\,Hp(x,y)$

5. Obtain the processed image $gp(x,y) = Real(ifft2(fftshift(G(u,v)))$ (extract the desired subportion).

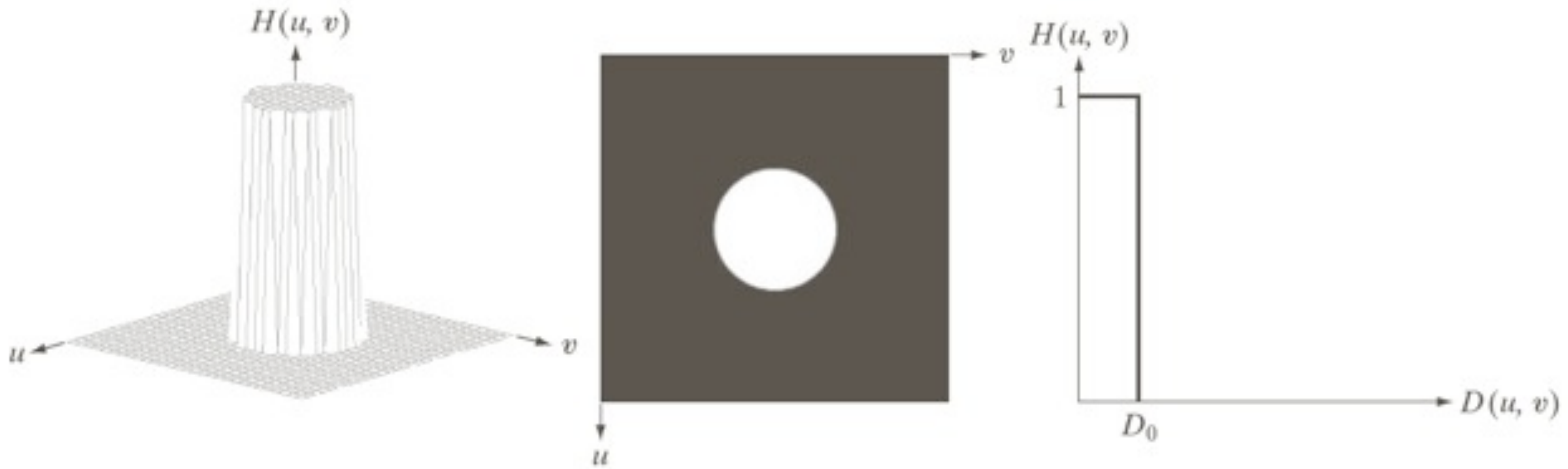
NB. We take the real part to remove spurious imaginary parts that come from our computations (rounding etc.)

Lowpass filters:

- Ideal lowpass filter

- Gaussian lowpass filter

- Butterworth lowpass filter



a b c

FIGURE 4.40 (a) Perspective plot of an ideal lowpass-filter transfer function. (b) Filter displayed as an image. (c) Filter radial cross section.

$$H(u,v) = 1, \qquad \text{if } D(u,v) \le D_0$$
$$H(u,v) = 0 \qquad \text{if } D(u,v) > D_0$$

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

distance from the center of the image