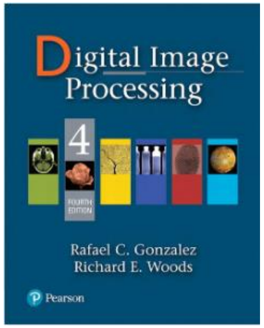


Digital Image Processing

Filtering in the Frequency Domain (I)

Dr. Tun-Wen Pai

1) **Concept of frequency domain filtering**
2) **Sampling/reconstruction/aliasing**
3) **Convolution in frequency domain**

Oct. 10, 2022

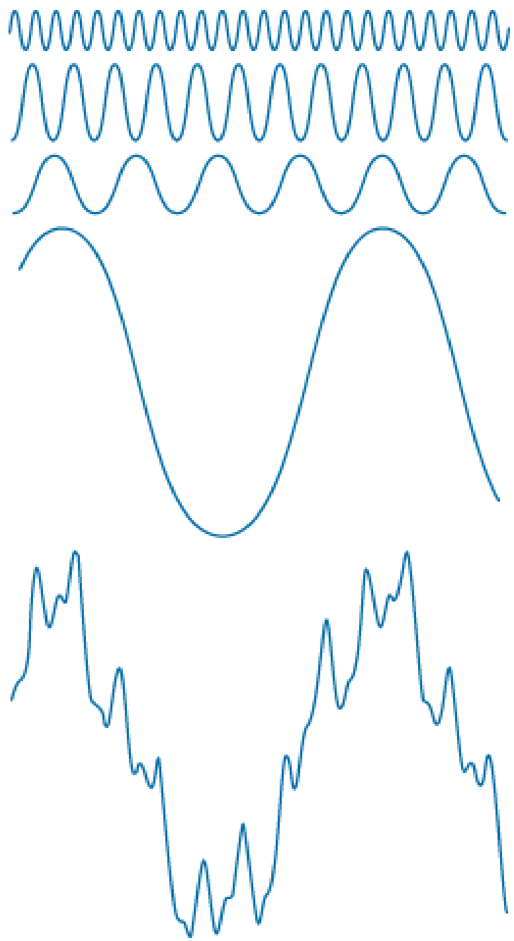
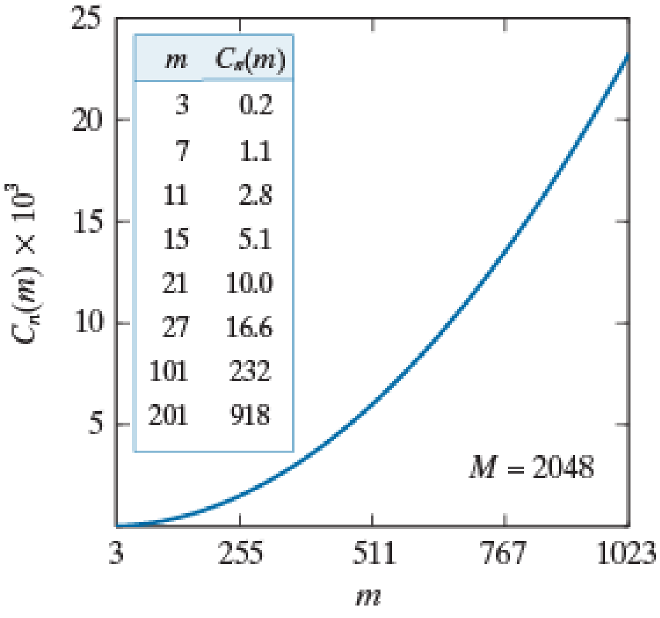
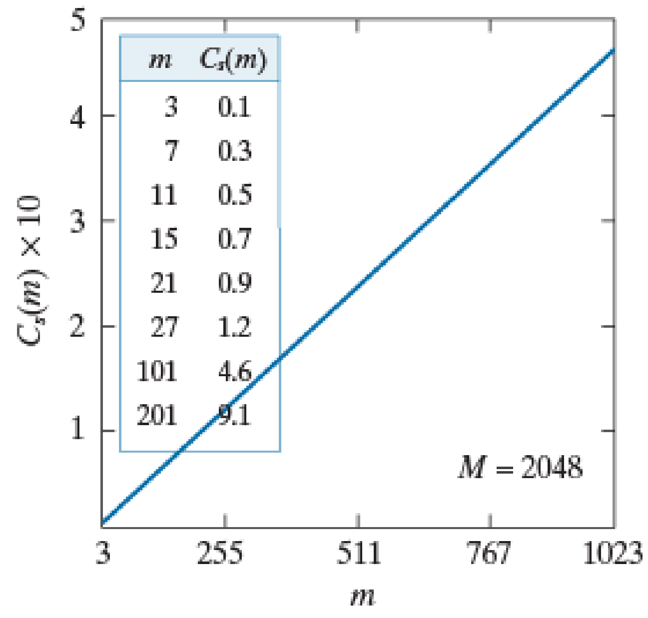


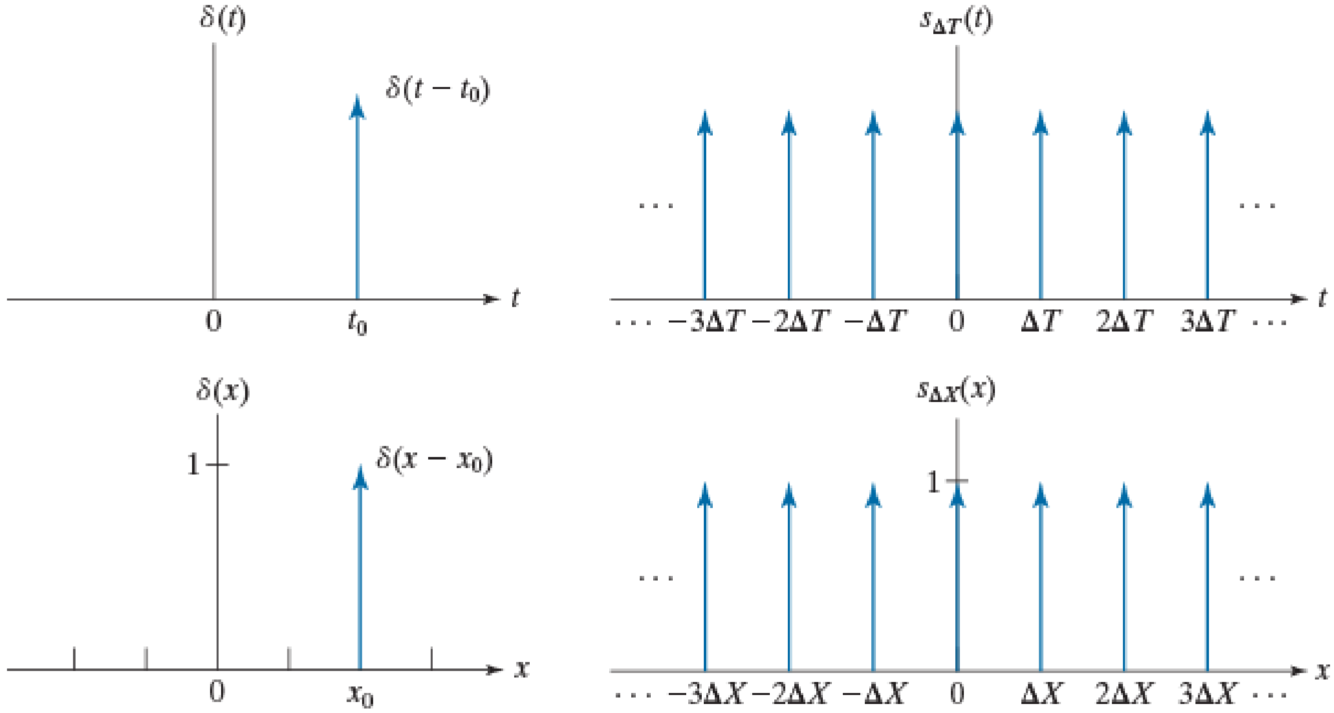
FIGURE 4.1 The function at the bottom is the sum of the four functions above it. Fourier's idea in 1807 that periodic functions could be represented as a weighted sum of sines and cosines was met with skepticism.

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.

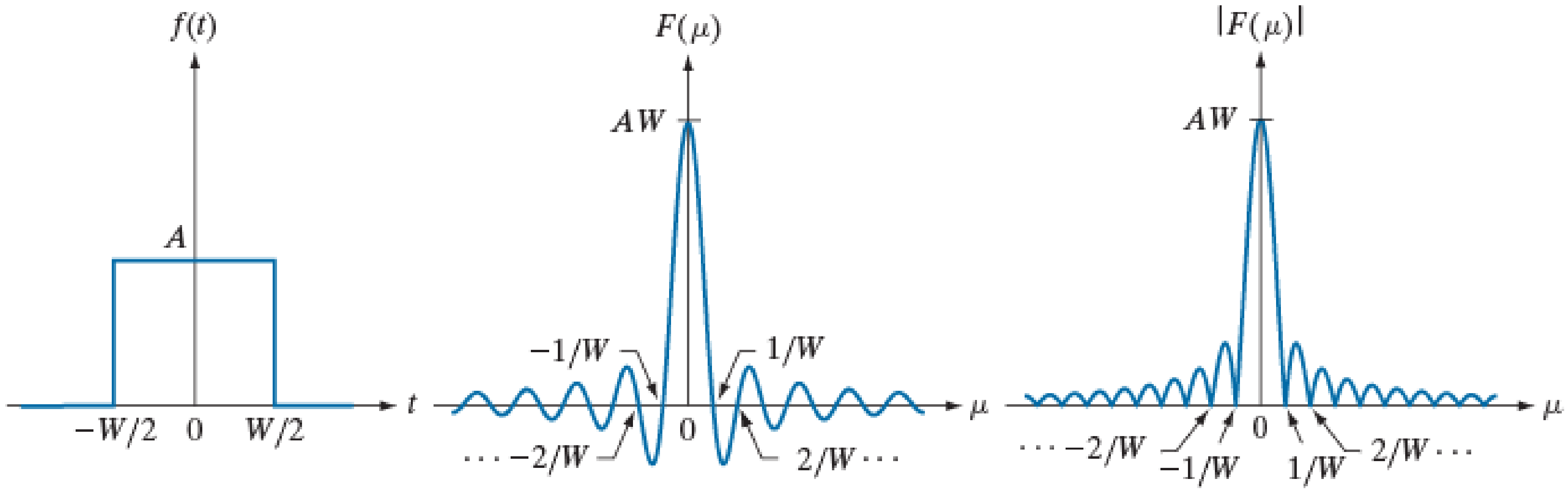






a	b
c	d

FIGURE 4.3
(a) Continuous impulse located at $t = t_0$. (b) An impulse train consisting of continuous impulses. (c) Unit discrete impulse located at $x = x_0$. (d) An impulse train consisting of discrete unit impulses.

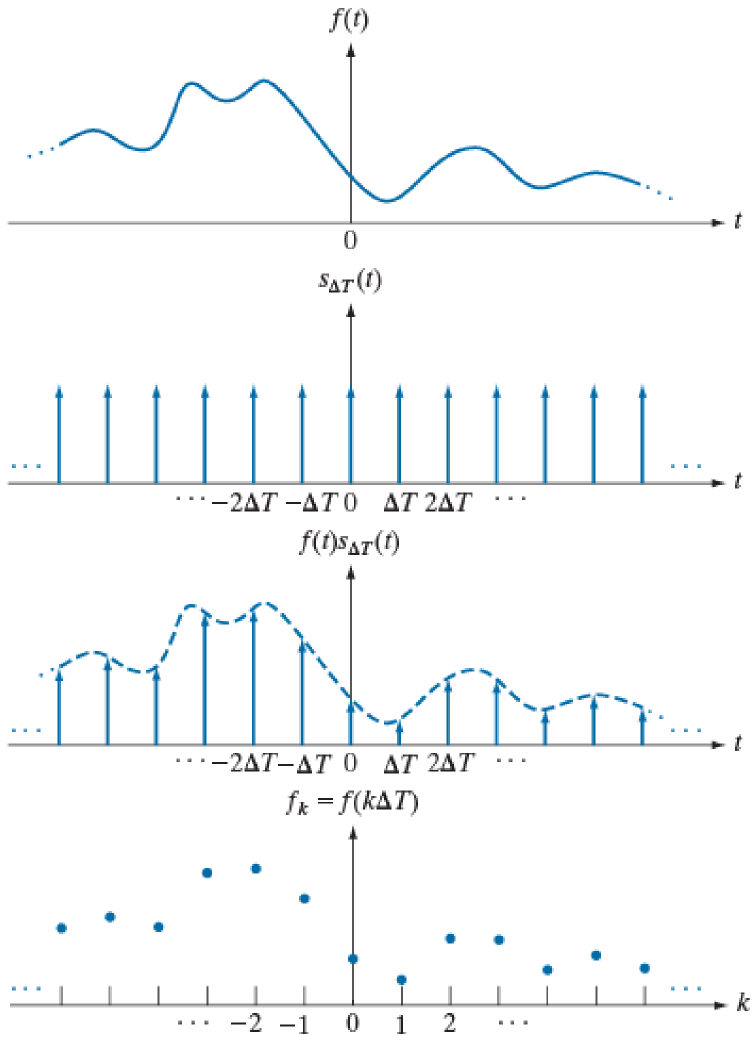


a b c

FIGURE 4.4 (a) A box function, (b) its Fourier transform, and (c) its spectrum. All functions extend to infinity in both directions. Note the inverse relationship between the width, W, of the function and the zeros of the transform.

a
b
c
d

FIGURE 4.5
(a) A continuous function. (b) Train of impulses used to model sampling. (c) Sampled function formed as the product of (a) and (b). (d) Sample values obtained by integration and using the sifting property of impulses. (The dashed line in (c) is shown for reference. It is not part of the data.)



a
b
c
d

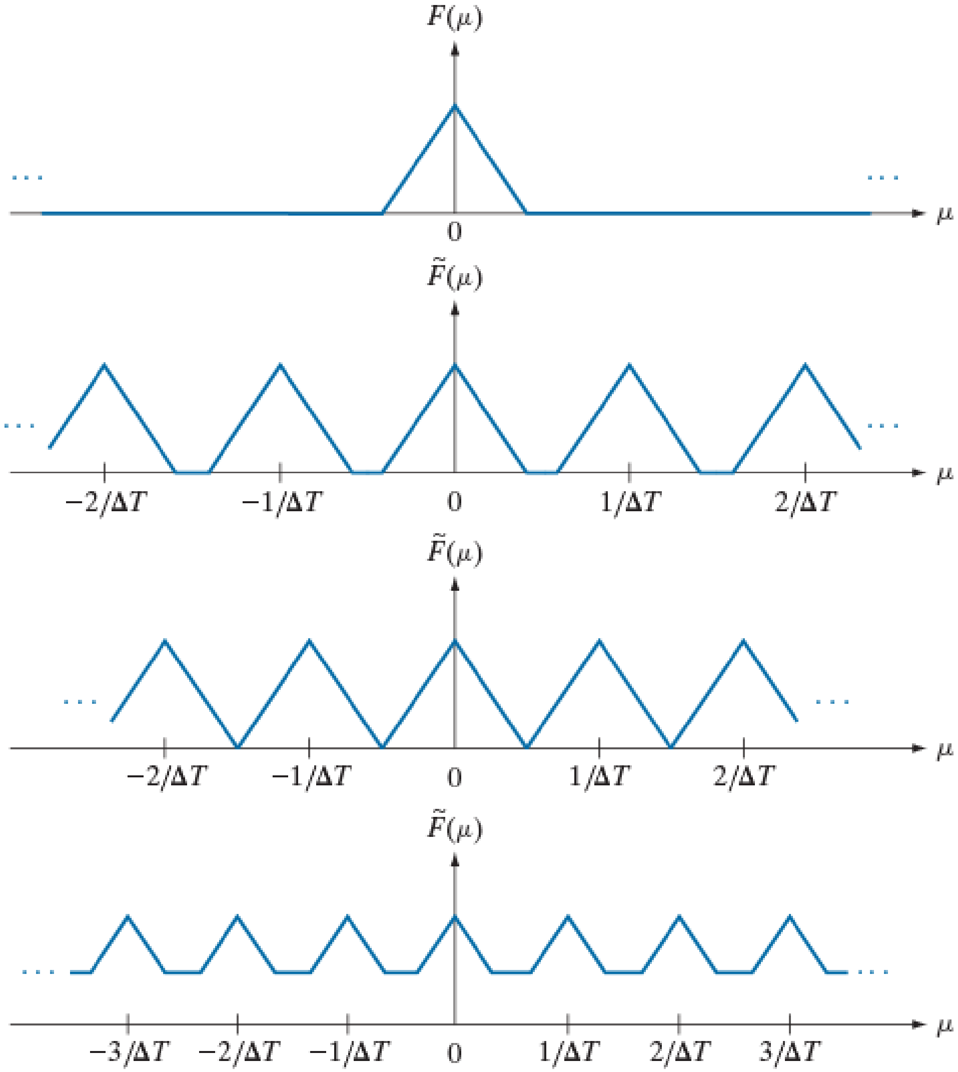
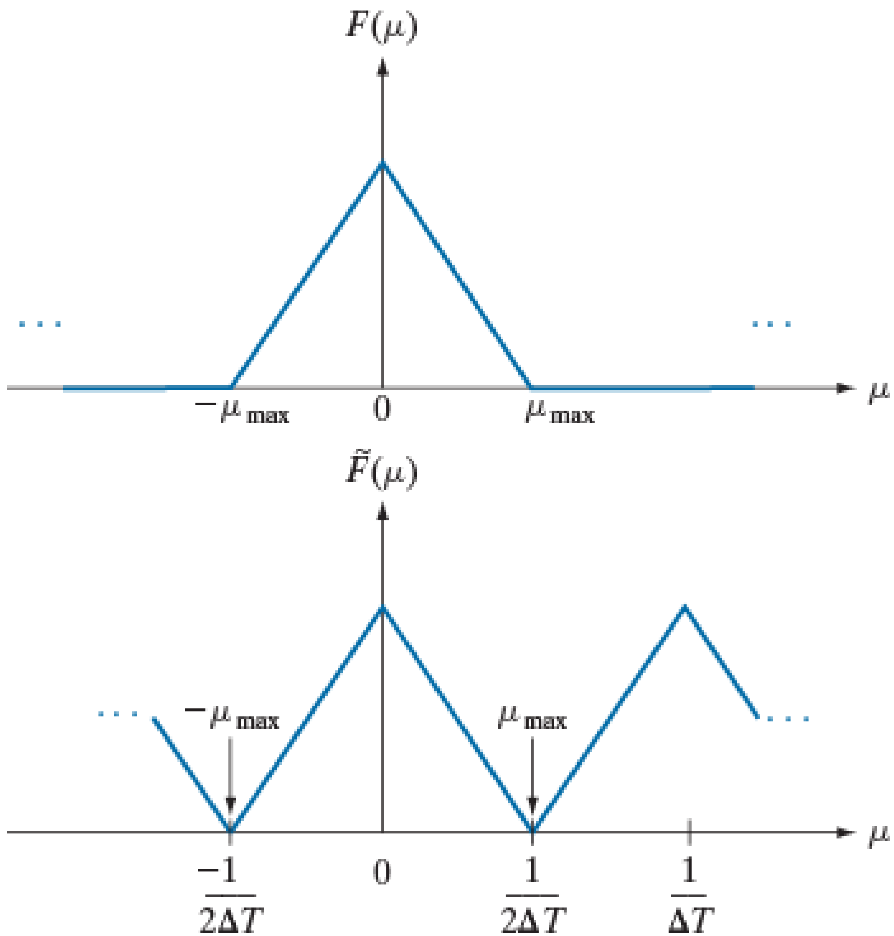
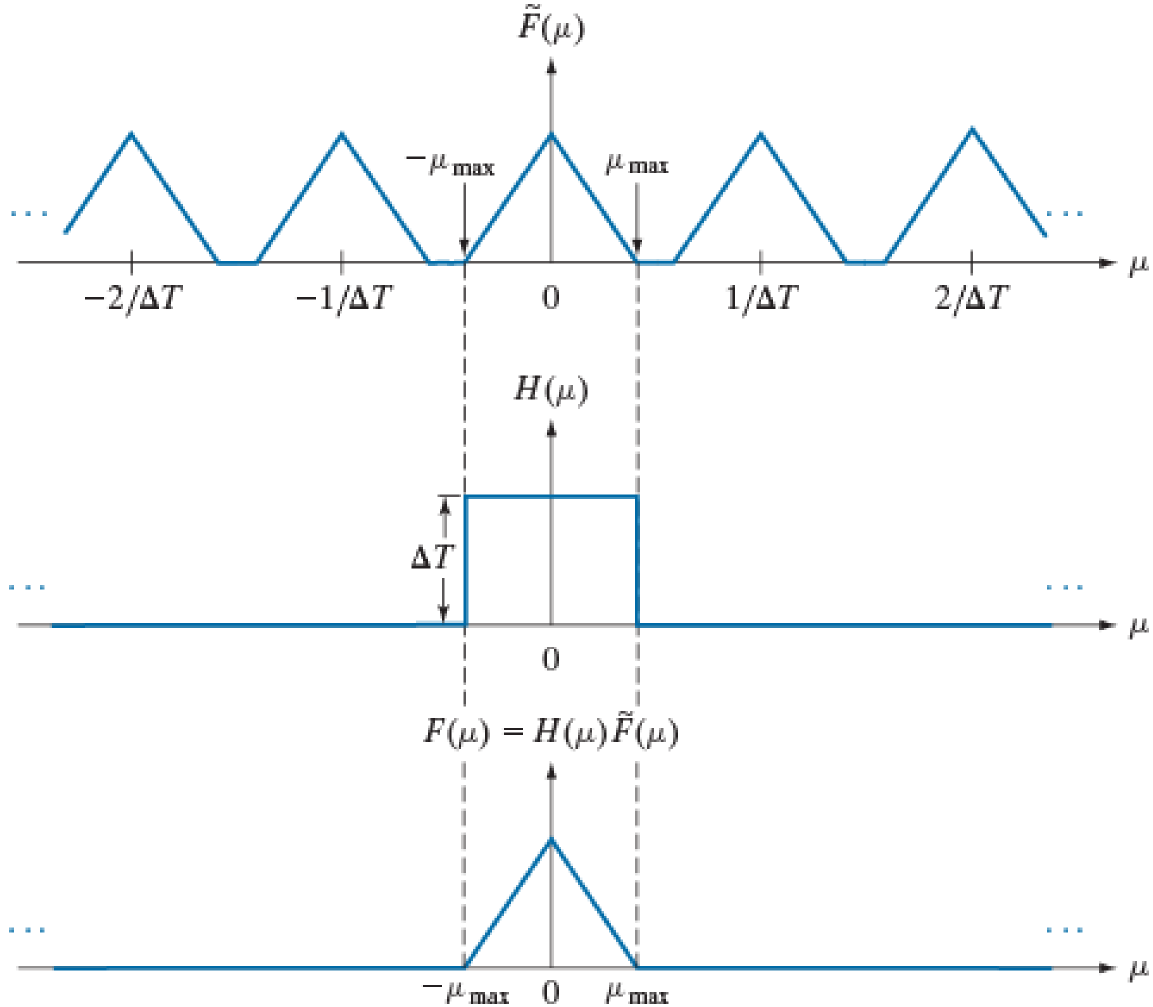


FIGURE 4.6
(a) Illustrative sketch of the Fourier transform of a band-limited function. (b)–(d) Transforms of the corresponding sampled functions under the conditions of over-sampling, critically sampling, and under-sampling, respectively.

a
b

FIGURE 4.7 (a) Illustrative sketch of the Fourier transform of a band-limited function. (b) Transform resulting from critically sampling that band-limited function.





a
b
c

FIGURE 4.8
(a) Fourier transform of a sampled, band-limited function.
(b) Ideal lowpass filter transfer function.
(c) The product of (b) and (a), used to extract one period of the infinitely periodic sequence in (a).

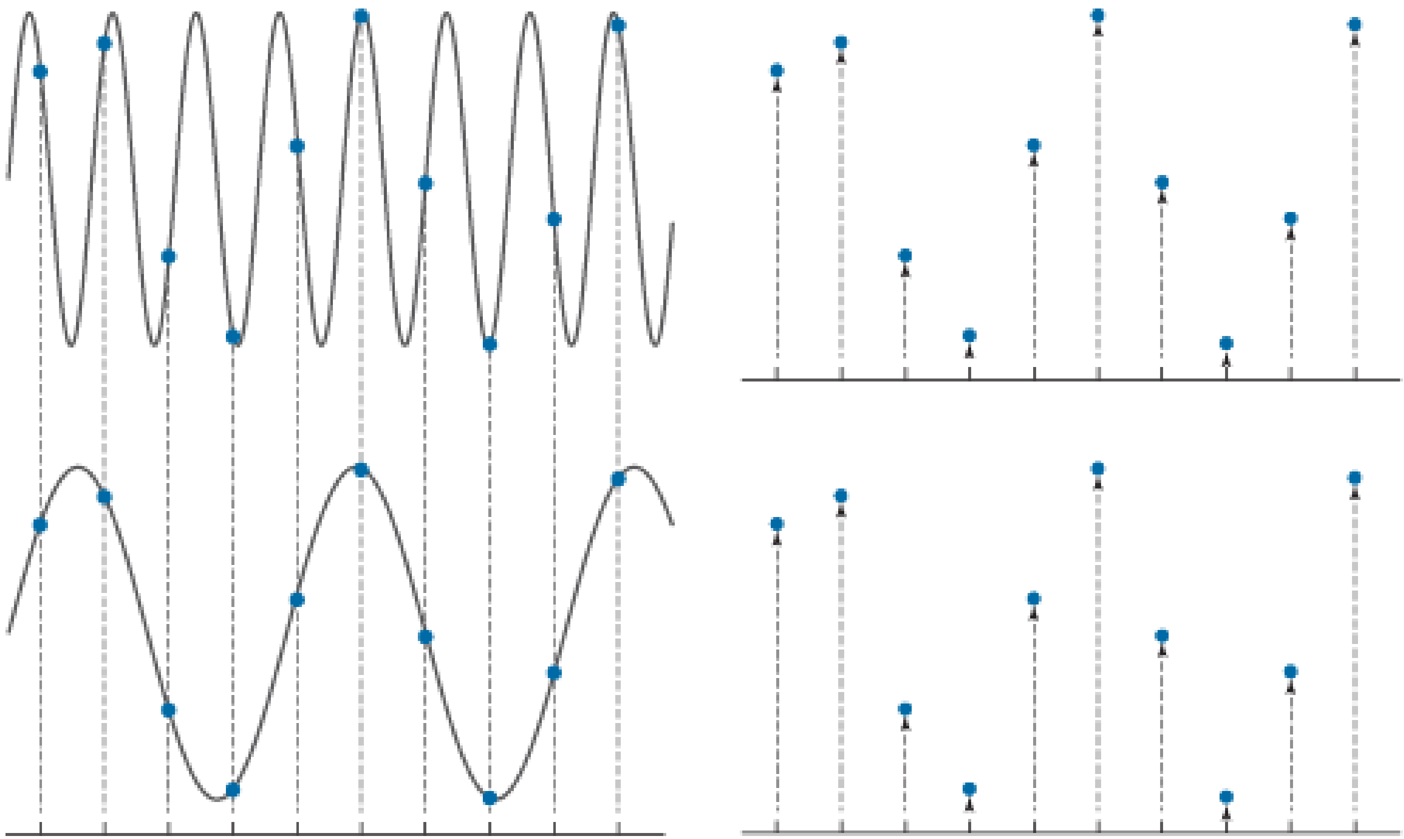


FIGURE 4.9 The functions in (a) and (c) are totally different, but their digitized versions in (b) and (d) are identical. Aliasing occurs when the samples of two or more functions coincide, but the functions are different elsewhere.

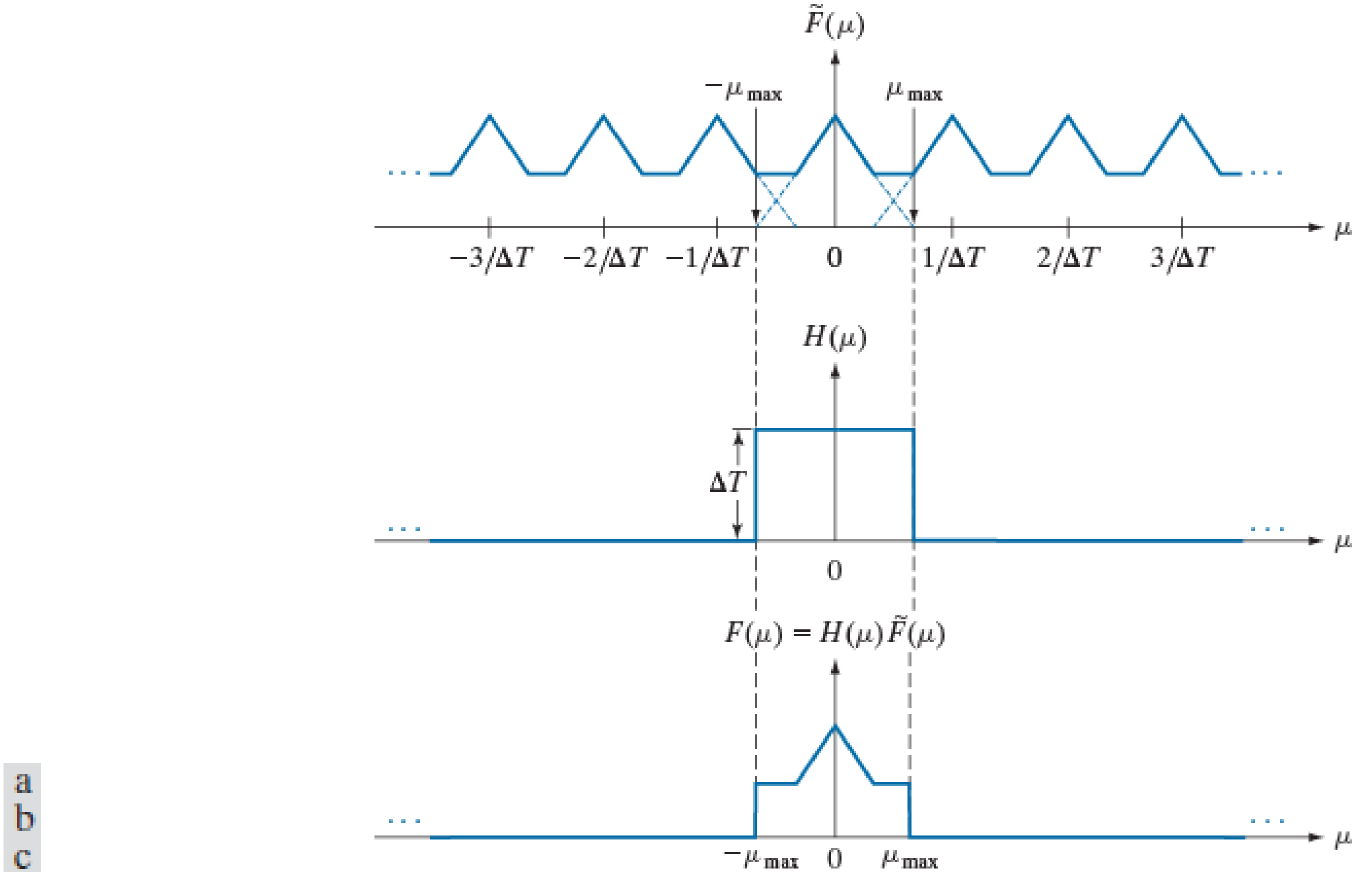


FIGURE 4.10 (a) Fourier transform of an under-sampled, band-limited function. (Interference between adjacent periods is shown dashed). (b) The same ideal lowpass filter used in Fig. 4.8. (c) The product of (a) and (b).The interference from adjacent periods results in aliasing that prevents perfect recovery of $F(\mu)$ and, consequently, of $f(t)$.

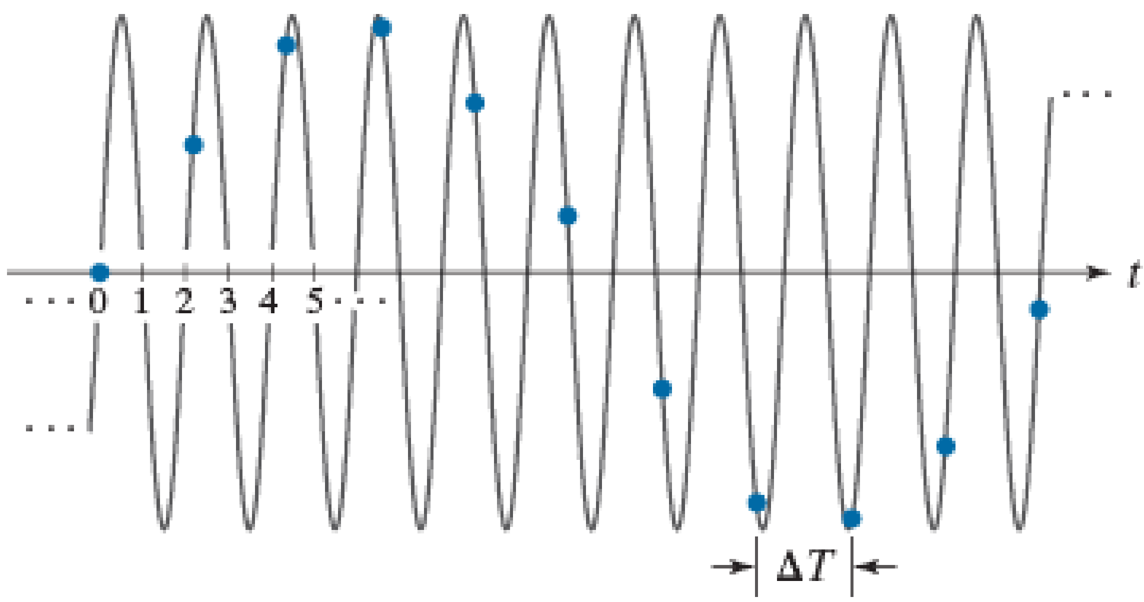
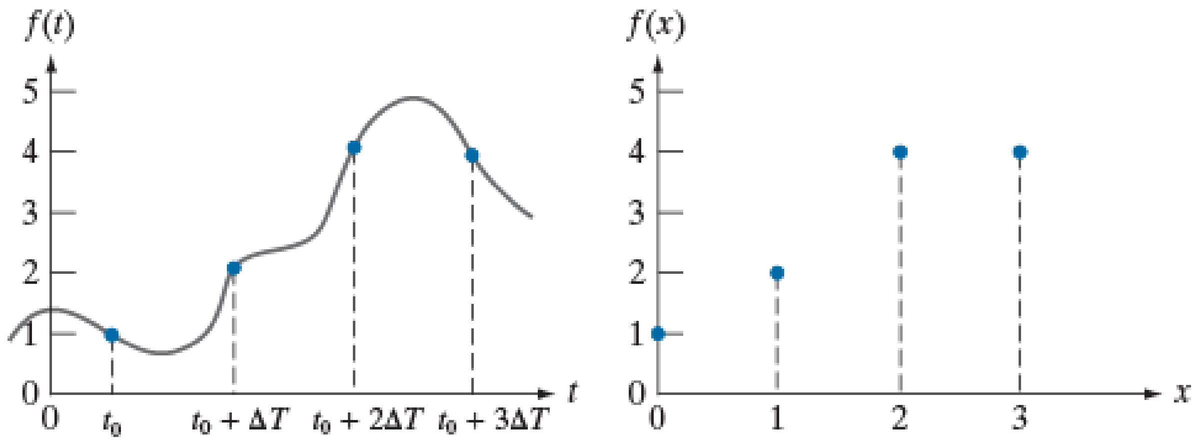


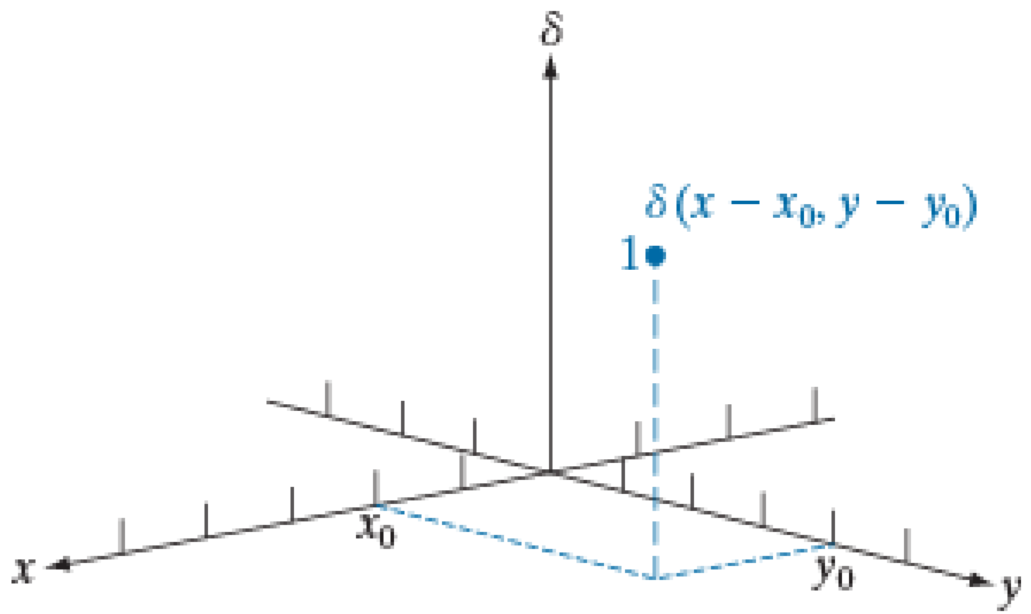
FIGURE 4.11 Illustration of aliasing. The under-sampled function (dots) looks like a sine wave having a frequency much lower than the frequency of the continuous signal. The period of the sine wave is 2 s, so the zero crossings of the horizontal axis occur every second. ΔT is the separation between samples.

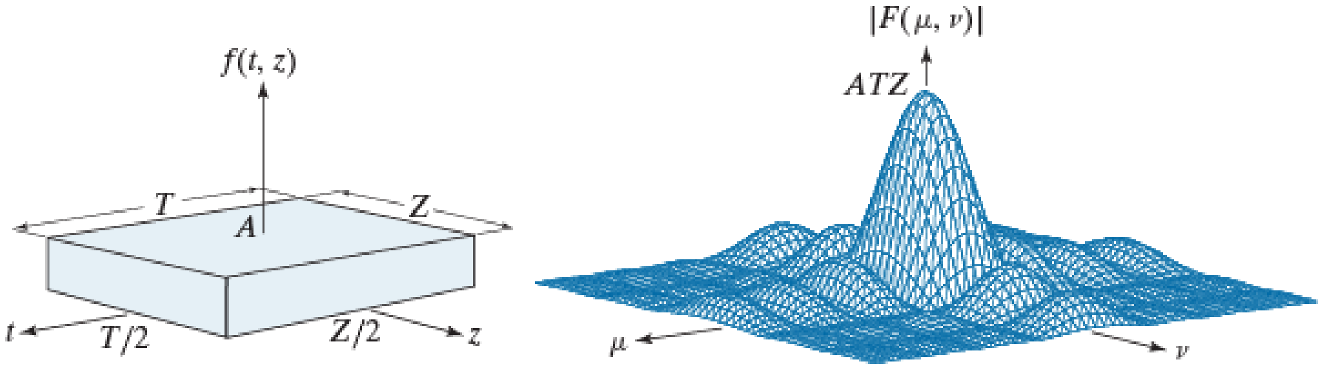


a b

FIGURE 4.12 (a) A continuous function sampled ΔT units apart. (b) Samples in the x-domain. Variable t is continuous, while x is discrete.

FIGURE 4.13 2-D unit discrete impulse. Variables x and y are discrete, and δ is zero everywhere except at coordinates (x_0, y_0), where its value is 1.





a b

FIGURE 4.14 (a) A 2-D function and (b) a section of its spectrum. The box is longer along the t-axis, so the spectrum is more contracted along the μ-axis.

a b

FIGURE 4.3 (a) Image of a 20 x 40 white rectangle on a black background of size 512 x 512 pixels. (b) Centered Fourier spectrum shown after application of the log transformation given in Eq.(3.2-2). Compare with Fig.4.2.

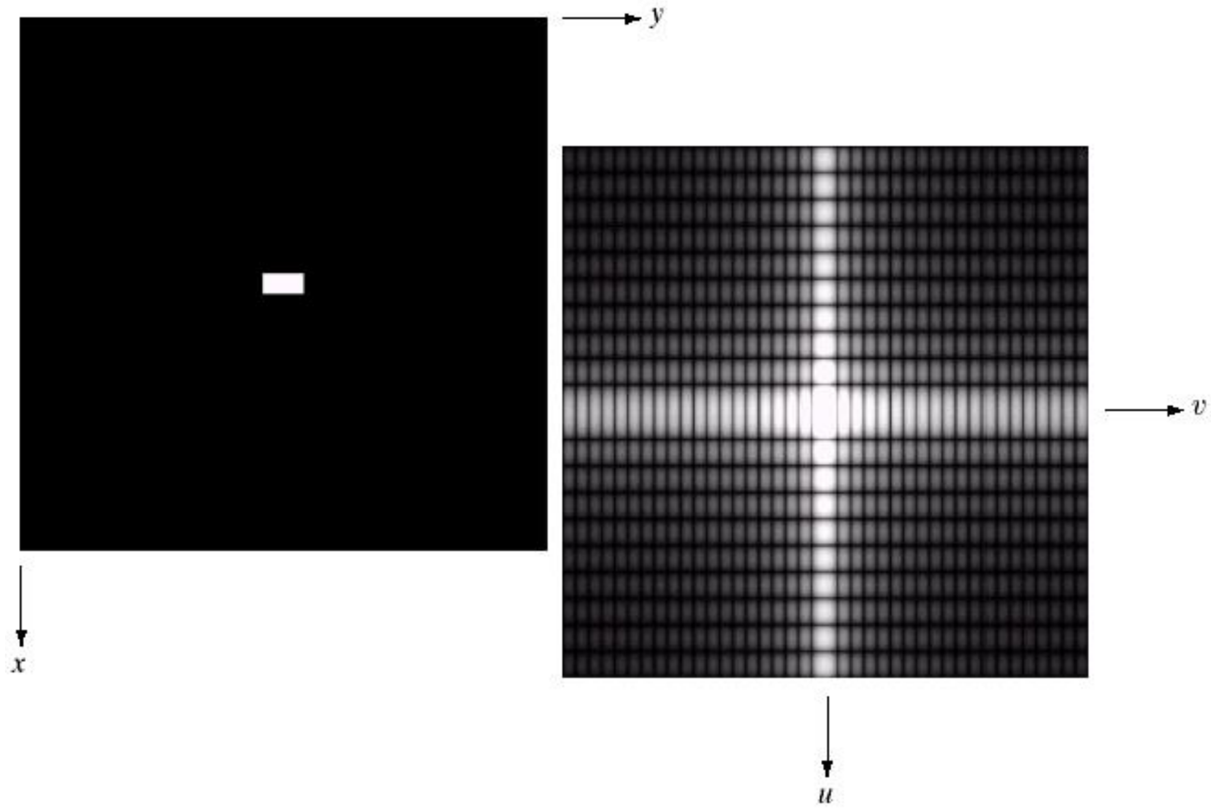
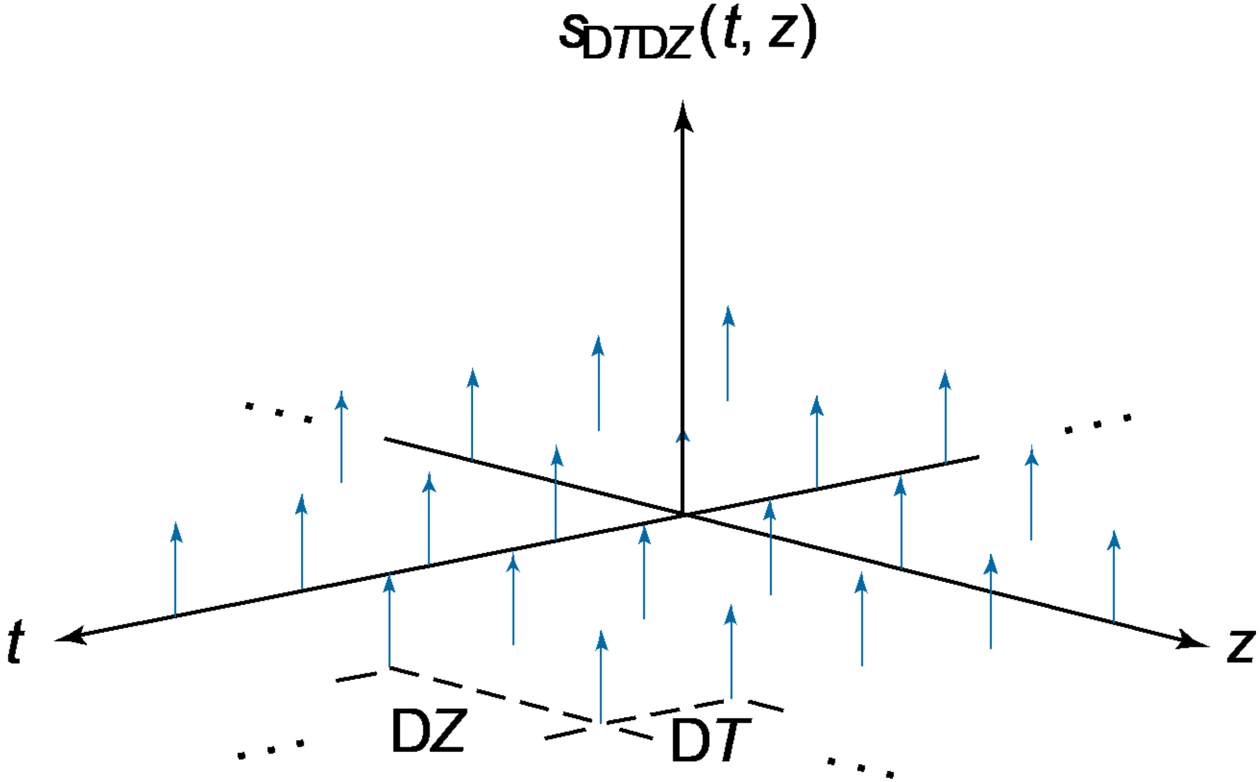


FIGURE 4.15
2-D impulse train.



a b

FIGURE 4.16 Two-dimensional Fourier transforms of (a) an over-sampled, and (b) an under-sampled, band-limited function.

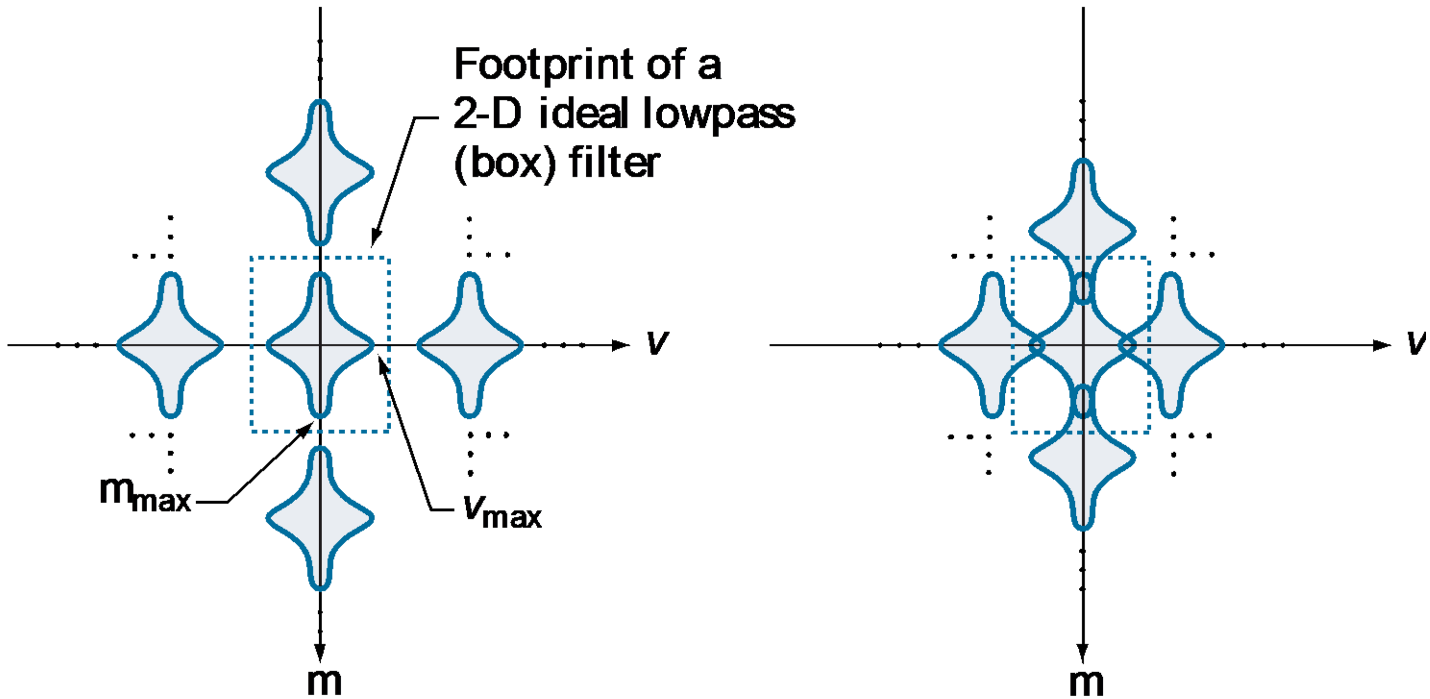
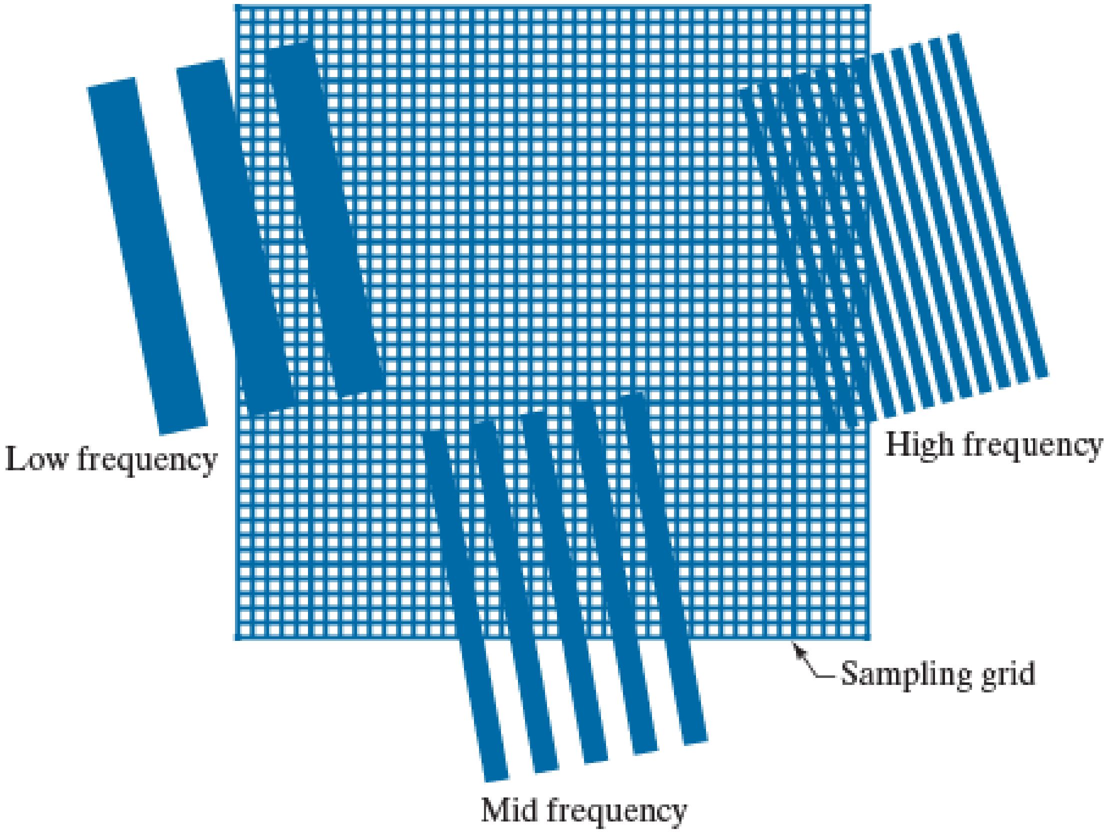


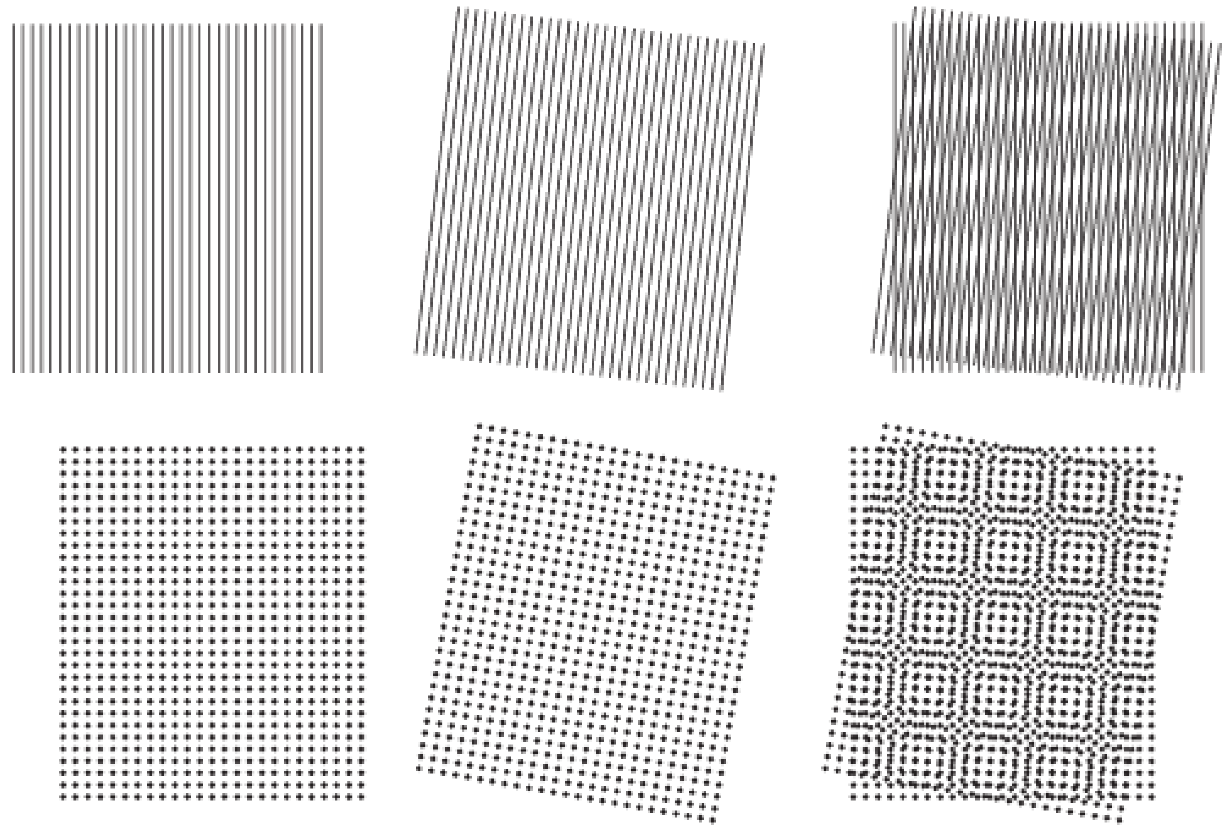
FIGURE 4.17 Various aliasing effects resulting from the interaction between the frequency of 2-D signals and the sampling rate used to digitize them. The regions outside the sampling grid are continuous and free of aliasing.



wagon wheel effect
https://www.youtube.com/watch?v=QOwzkND_ooU

a b c

FIGURE 4.19 Illustration of aliasing on resampled natural images. (a) A digital image of size 772 × 548 pixels with visually negligible aliasing. (b) Result of resizing the image to 33% of its original size by pixel deletion and then restoring it to its original size by pixel replication. Aliasing is clearly visible. (c) Result of blurring the image in (a) with an 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.)

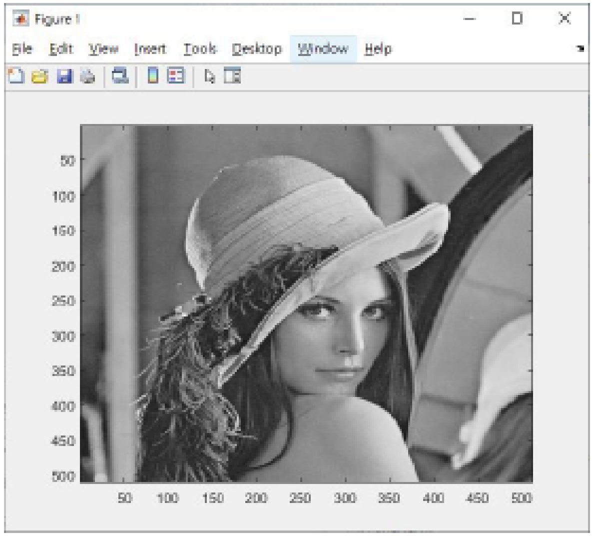


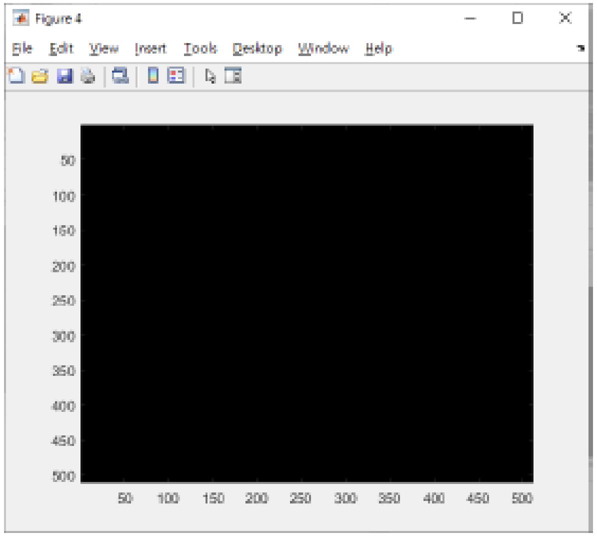
a b c
d e f

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

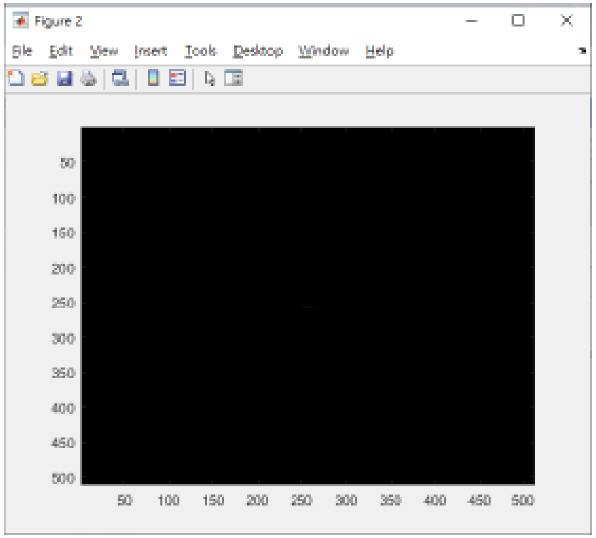
FIGURE 4.21
A newspaper image digitized at 75 dpi. Note the moiré-like pattern resulting from the interaction between the ±45° orientation of the half-tone dots and the north-south orientation of the sampling elements used to digitized the image.

```
% 2. display fft image in absolute value
[x,map]=imread('lena.bmp');
xf=fftshift(fft2(x));
fabs=abs(xf);
fm=max(fabs(:));
figure,image(x);
colormap(map);
figure,image(fabs*255/fm);
colormap(map)
```



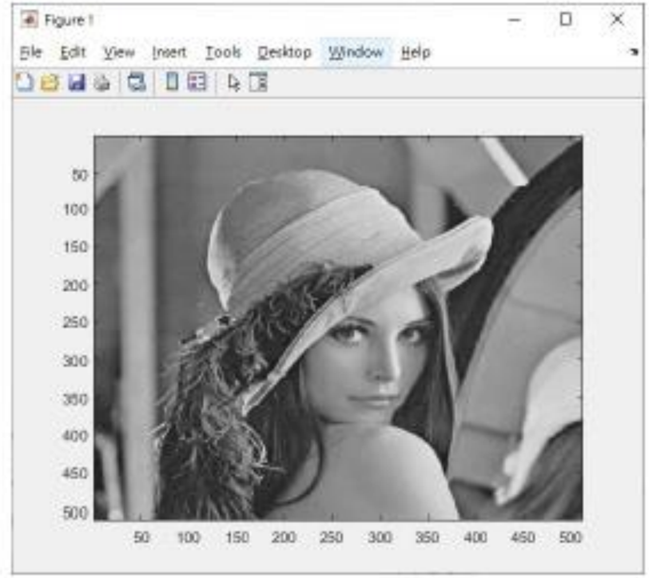


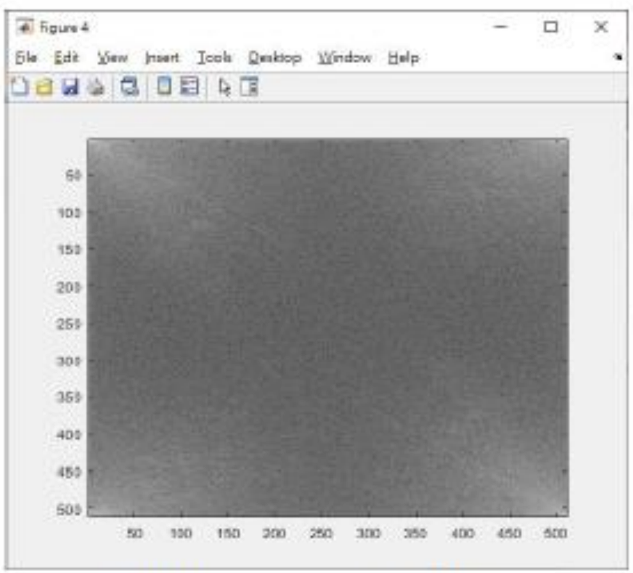
Without using fftshift()



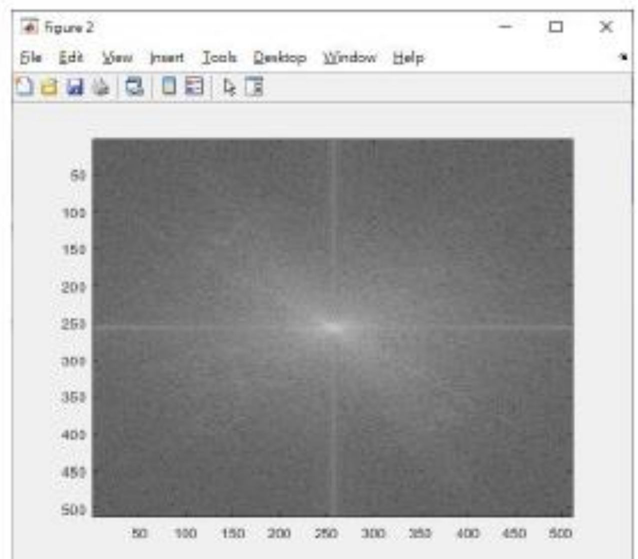
Using fftshift()

```
%1. display fft image in log
[x,map]=imread('lena.bmp');
xf=fftshift(fft2(x));
flog = log(1+abs(xf));
fm = max(flog(:));
figure,image(x);
colormap(map)
figure,image(flog*255/fm);
colormap(map)
```



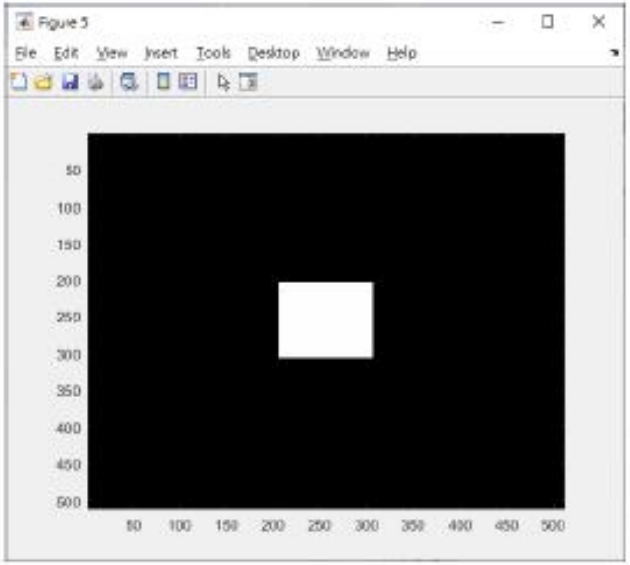


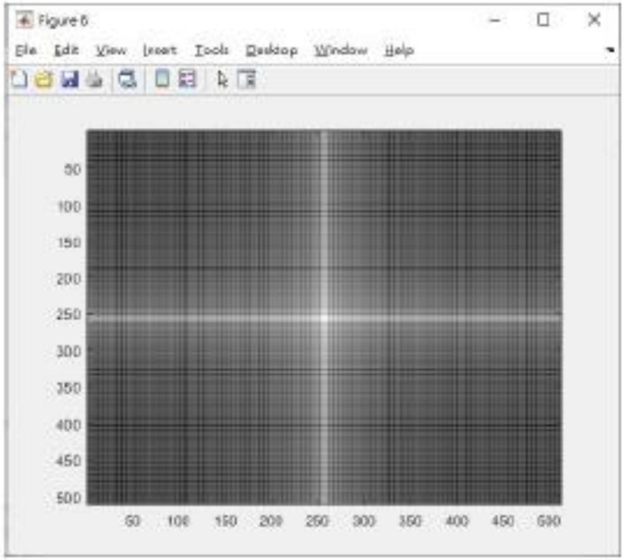
Without using fftshift()

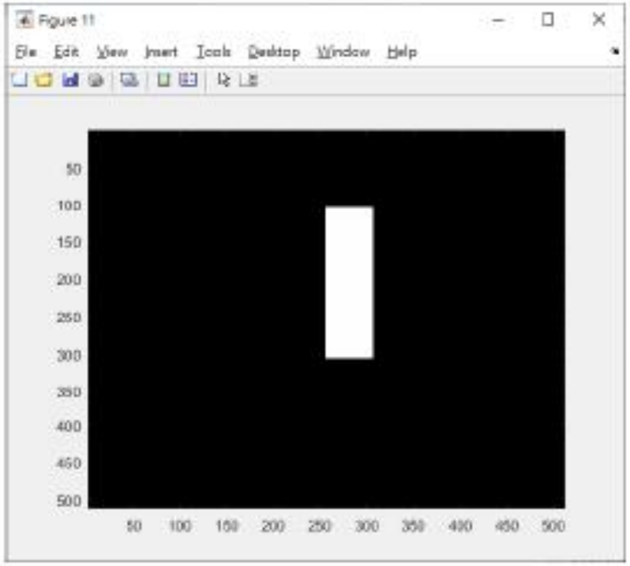


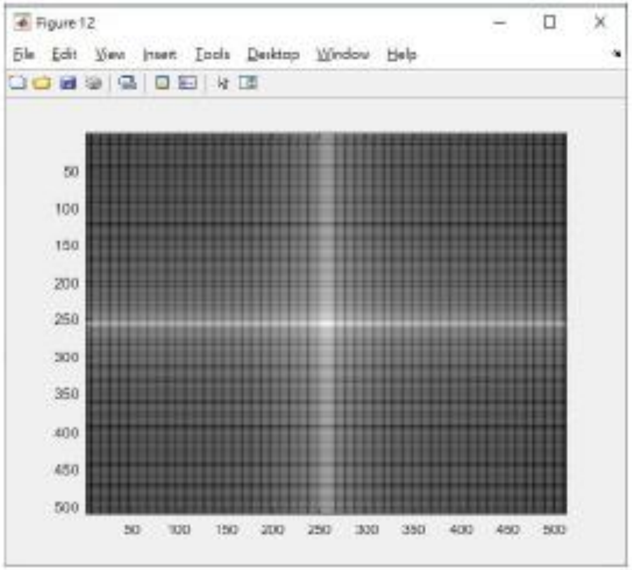
Using fftshift()

```
%3.    square
a=zeros(512,512);
a(206:306,206:306)=255;
af=fftshift(fft2(a));
flog = log(1+abs(af));
fm = max(flog(:));
image(flog/fm);
figure,image(a);
colormap(map)
figure,image(flog*255/fm);
colormap(map)
```

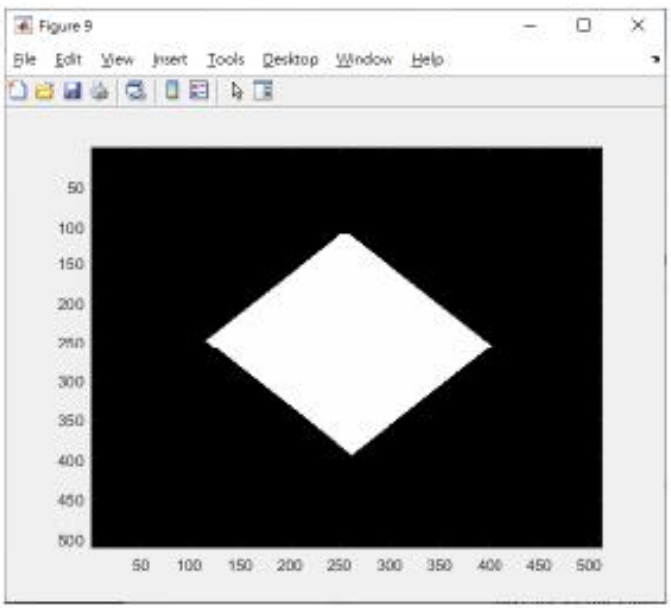


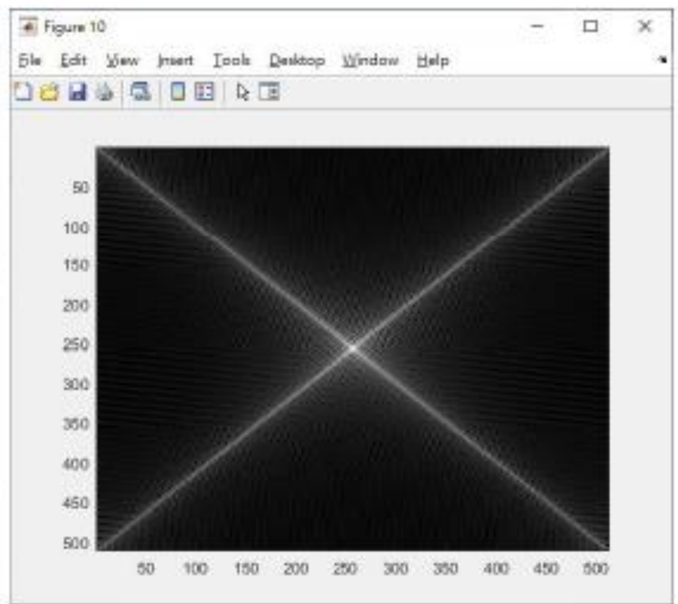




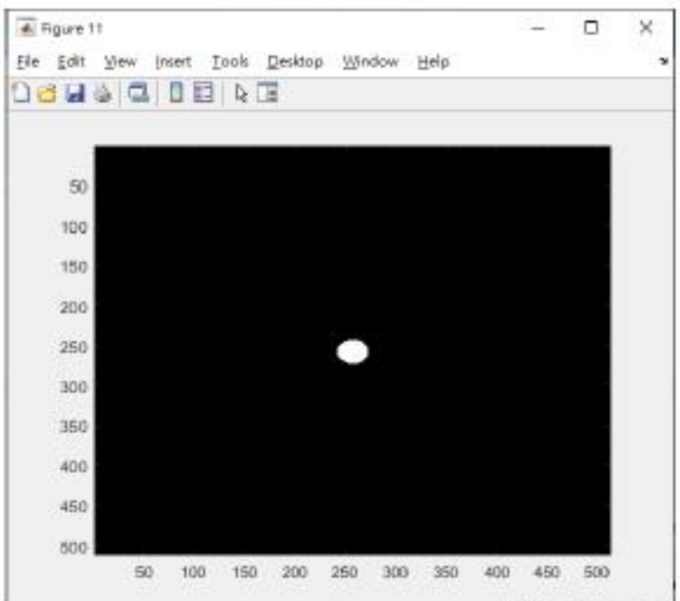


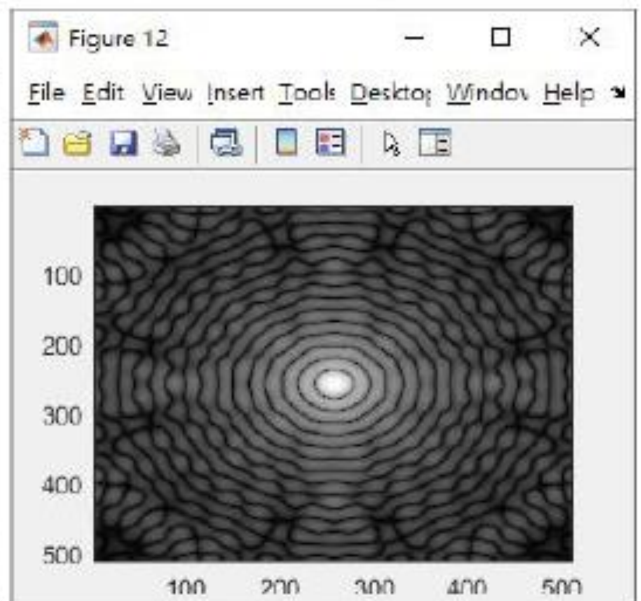
```
% 4. rotate square
[x,y]=meshgrid(1:512,1:512);
b= ((x+y)<658)&((x+y)>364)&((x-y)>-134)&((x-y)<146);
bf=fftshift(fft2(b));
flog = log(1+abs(bf));
fm = max(flog(:));
image(flog/fm);
figure,image(255*b);
colormap(map)
figure,image(flog*255/fm);
colormap(map)
```





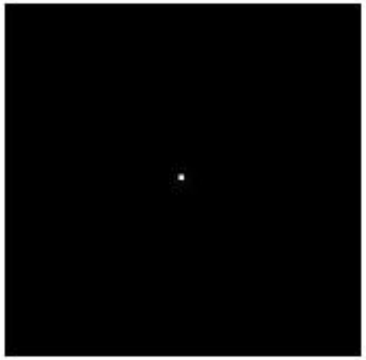
```
% 5. small circle
[x,y]=meshgrid(-256:255,-256:255);
z=sqrt(x.^2+y.^2);
c=(z<15);
cf=fftshift(fft2(c));
flog = log(1+abs(cf));
fm = max(flog(:));
image(flog/fm);
figure,image(255*c);
colormap(map)
figure,image(flog*255/fm);
colormap(map)
```



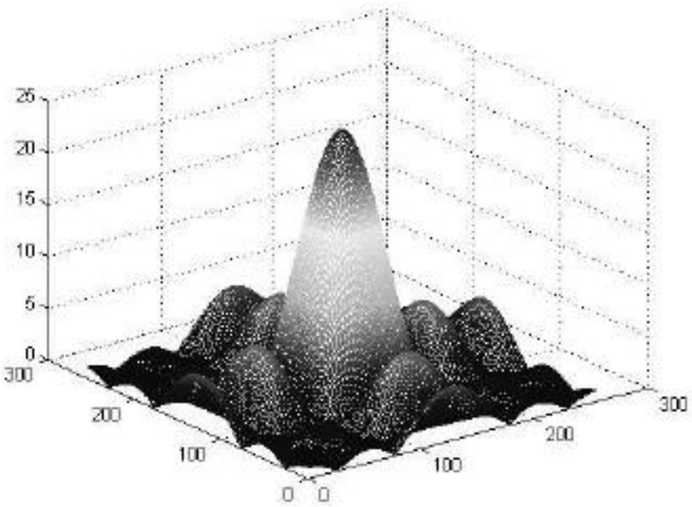


Fourier Transform Example

$f(x,y)$ $\quad$ $|F(u,v)|$ displayed as image

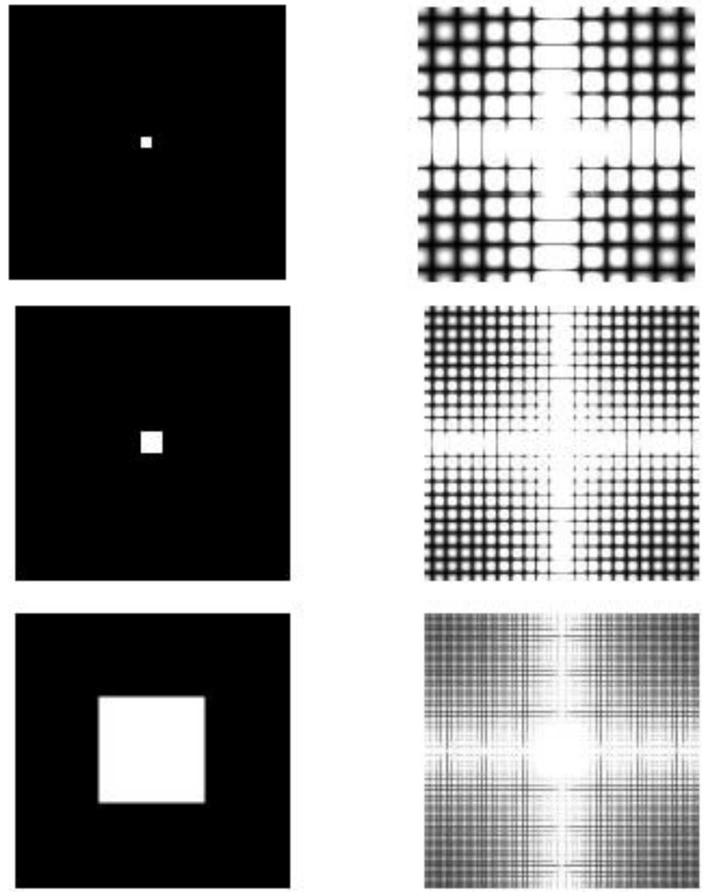


$|F(u,v)|$ displayed in 3-D



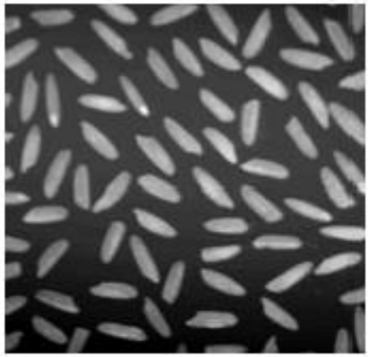
Fourier Transform Example

Image Magnitude Spectrum

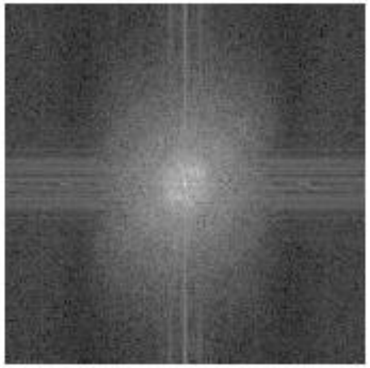


•As the size of the box increases in spatial domain, the corresponding “size” in the frequency domain decreases.

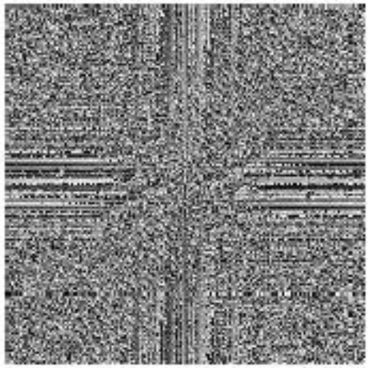
Fourier Transform of “Rice” Image



$f(x,y)$



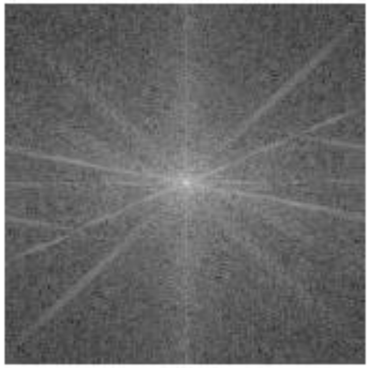
$|F(u,v)|$



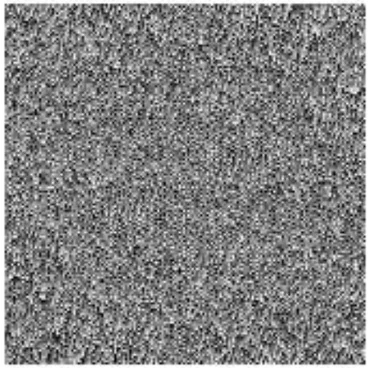
$\angle F(u,v)$

Fourier Transform of “Camera Man”

$$g(x, y)$$



$$|G(u,v)|$$



$$\angle G(u,v)$$

Importance of Phase Information in Images

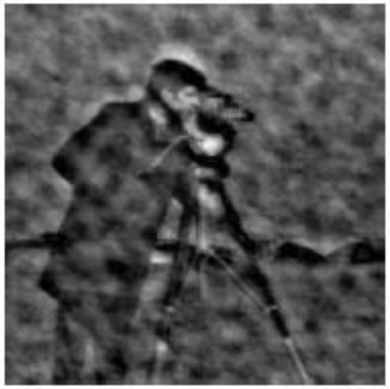


Image formed from magnitude spectrum of Rice and phase spectrum of Camera man

$\mathcal{F}^{-1}(| F(u, v) |) * \exp(i \angle G(u, v)))$

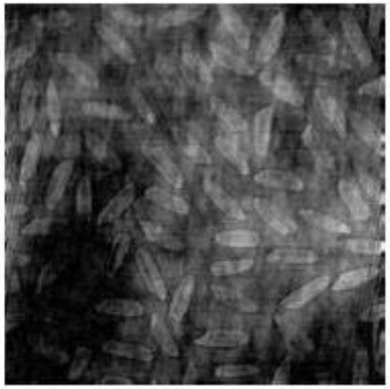


Image formed from magnitude spectrum of Camera man and phase spectrum of Rice

$\mathcal{F}^{-1}(| G(u, v) |) * \exp(i \angle F(u, v)))$

1-D Discrete Fourier Transform (DFT)

- For discrete images of **finite extent**, the analogous Fourier transform is the DFT.

- We will first study this for the 1-D case

- Suppose{ f(0), f(1),…, f(N-1) } is a sequence/vector/1-D image of length N. Its N-point DFT is defined as

$$F(u) = \sum_{n=0}^{N-1} f(n) e^{-j2\pi \frac{nu}{N}}, \quad \mathrm{u} = 0,1,\ldots \mathrm{N}-1$$

- Inverse DFT (note the normalization):

$$f(n) = \frac{1}{N} \sum_{u=0}^{N-1} F(u) e^{-j2\pi \frac{nu}{N}}, \quad \mathrm{u} = 0,1,\ldots \mathrm{N}-1$$

- Example: Let f(n) = {1, -1, 2, 3}. (Note that N=4.)

$$F(0) = \sum_{n=0}^{3} f(n) e^{0} = f(0) + f(1) + f(2) + f(3) = 5$$

$$F(1) = \sum_{n=0}^{3} f(n) e^{-j2\pi \frac{n}{4}} = 1e^{0} + (-1)e^{-j\frac{\pi}{2}} + 2e^{-j\pi} + 3e^{-j\frac{3\pi}{2}}$$

$$= 1 + (0 + j) + 2(-1) + 3(0 + j) = -1 + 4j$$

$$F(2) = \sum_{n=0}^{3} f(n) e^{-j2\pi\frac{2n}{4}} = 1e^0 + (-1)e^{-j\pi} + 2e^{-j2\pi} + 3e^{-j3\pi}$$
$$= 1 + 1 + 2 + (-3) = 1$$

$$F(3) = \sum_{n=0}^{3} f(n) e^{-j2\pi\frac{3n}{4}} = 1e^0 + (-1)e^{-j\frac{3\pi}{2}} + 2e^{-j3\pi} + 3e^{-j\frac{9\pi}{2}}$$
$$= 1 + (0 - j) + 2(-1) + 3(0 - j) = -1 - 4j$$

- F(u) is complex even though f(n) is real. This is typical.

- Implementing the DFT directly requires $O(N^2)$ computations, where N is the length of the sequence.

- There is a much more efficient implementation of the DFT using the **Fast Fourier Transform** (FFT) algorithm. This is not a new transform (as the name suggests) but just an efficient algorithm to compute the DFT.

- The FFT works best when $N=2^m$ (or is the power of some integer base/radix). The radix-2 algorithm is most commonly used.

- The computational complexity of the radix-2 FFT algorithm is Nlog(N) multiplies.
So it is an O(Nlog(N)) algorithm.

- In matlab, the command **fft** implements this algorithm (for 1-D case). See section 3.4 of the text for more details about the FFT algorithm.

2-D Discrete Fourier Transform (DFT)

- The Fourier transform is suitable for continuous-domain images, which maybe of infinite extent.
- For discrete image of **finite extent**, the analogous Fourier transform is the 2-D DFT.
- Suppose f(m, n), m = 0, 1,…, M-1, n = 0, 1,…, N-1, is a discrete M x N image. Its 2-D DFT F(u, v) is defined as:

$$F(u,v) = \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} f(m,n) e^{-j2\pi(\frac{mu}{M} + \frac{nv}{N})}$$

$$u = 0,1,\ldots M-1, \ v = 0,1,\ldots N-1$$

- Inverse DFT is defined as:

$$f(m,n) = \frac{1}{MN} \sum_{v=0}^{N-1} \sum_{u=0}^{M-1} F(u,v) e^{j2\pi(\frac{mu}{M} + \frac{nv}{N})}$$

$$m = 0,1,\ldots M-1, \ \text{n} = 0,1,\ldots N-1$$

- For discrete image of **finite extent**, the analogous Fourier transform is the 2-D DFT.
- <u>Note about normalization</u>: The normalization by MN is different than that in text. We will use the one above since it is more widely used. The matlab function **fft2** implements the DFT as defined above.

- Most often we have M=N (square image) and in that case, we define a unitary DFT as follows:

$$F(u,v)=\frac{1}{N}\sum_{n=0}^{N-1}\sum_{m=0}^{N-1} f(m,n)e^{-j2\pi(\frac{mu+nv}{N})},$$

$$f(m,n)=\frac{1}{N}\sum_{u=0}^{N-1}\sum_{v=0}^{N-1} F(u,v)e^{j2\pi(\frac{mu+nv}{N})}$$

- We will refer to the above as just DFT (drop unitary) for simplicity.

- In matlab, **fft2** implement 2-D FFT, using 1-D FFT.

Properties of DFT

- **Linearity** (Distributivity and Scaling): This holds in both discrete and continuous-domains.
  - DFT of the sum of two images is the sum of their individual DFTs.

$$F\left[f_1(m,n)+f_2(m,n)\right]=F\left[f_1(m,n)\right]+F\left[f_2(m,n)\right]$$

  - DFT of a scaled image is the DFT of the original image scaled by the same factor.

$$F\left[af(m,n)\right]=aF\left[f(m,n)\right]$$

- **Spatial scaling** (only for continuous-domain):

$$F\left[f(x,y)\right]=F(u,v)\Rightarrow F\left[f(ax,by)\right]=\frac{1}{|ab|}F(\frac{u}{a},\frac{v}{b})$$

  - If a, b > 1, image "shrinks" and the spectrum "expands".

- **Periodicity** (only for discrete case): The DFT and its inverse are periodic (in both the dimensions), with period N.

$$F(u,v)=F(u+N,v)=F(u,v+N)=F(u+N,v+N)$$

□ Similarly

$$f(m,n) = \frac{1}{N}\sum_{u=0}^{N-1}\sum_{v=0}^{N-1} F(u,v) e^{j2\pi(\frac{mu+nv}{N})}$$

is also N-periodic in m and n.

- **Separability** (both continuous and discrete):
  Decomposition of 2D DFT into 1D DFTs

$$F(u,v) = \frac{1}{N}\sum_{n=0}^{N-1}\sum_{m=0}^{N-1} f(m,n) e^{-j2\pi(\frac{mu+nv}{N})}$$

$$= \frac{1}{N}\sum_{m=0}^{N-1} e^{-j2\pi(\frac{mu}{N})} \quad \text{x} \quad \underbrace{\sum_{n=0}^{N-1} f(m,n) e^{-j2\pi(\frac{nv}{N})}}_{\tilde{f}(m,v)\text{: 1D DFT applied to each row of } f(m,n)}$$

$$= \frac{1}{N} \underbrace{\sum_{m=0}^{N-1} \tilde{f}(m,v) e^{-j2\pi(\frac{mu}{N})}}_{\text{1D DFT applied to each column of } \tilde{f}(m,n)}$$

□ Similarly

$$f(m,n) = \underbrace{\frac{1}{N}\sum_{u=0}^{N-1} e^{j2\pi(\frac{mu}{N})} \qquad \underbrace{\sum_{v=0}^{N-1} F(u,v) e^{j2\pi(\frac{nv}{N})}}_{\text{1D DFT applied to each row of } F(u,v)}}_{\text{1D DFT applied to each column of previous result}}$$

- **Translation** (discrete and continuous case):

$$\mathcal{F}\{f(m-m_o,n-n_o)\}=F(u,v)e^{\frac{-j2\pi}{N}(um_0+vn_0)}$$

- □ Note that

$$\left|F(u,v)e^{-j\frac{2\pi}{N}(um_0+vn_0)}\right|=|F(u,v)|, \text{so } f(m,n)\ and$$

$$f(m-m_0,n-n_0) \text{ have the same magnitude spectrum}$$

$$\text{but different phase spectrum.}$$

- □ Similarly, $f(m,n)e^{\frac{-j2\pi}{N}(u_0m+v_0n)} = \mathcal{F}^{-1}\{F(u-u_0,v-v_0)\}$

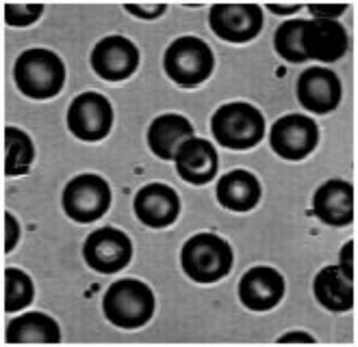
- **Conjugate Symmetry:** If f(m, n) is real, then F(u, v) is conjugate symmetric, i.e.

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

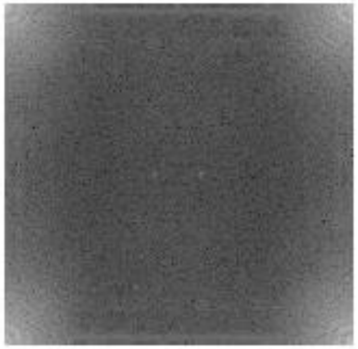
$$\Leftrightarrow |F(-u,-v)|=|F(u,v)| \text{ and } \angle F(-u,-v)=-\angle F(u,v)$$

- □ Therefore, we usually display $F(u-\frac{N}{2},v-\frac{N}{2})$, instead of F(u, v), since it is easier to visualize the symmetry of the spectrum in this case.
- □ This is done in Matlab using the ***fftshift*** command.

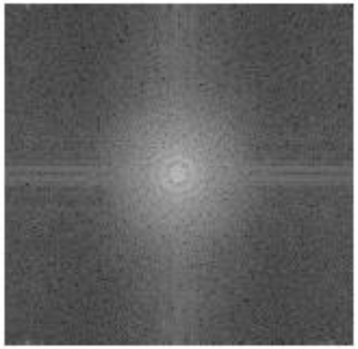
"Center" Magnitude Spectrum



$f(m,n)$



$|F(u,v)|$



$|F(u-N/2, V-N/2)|$

- **Convolution:** In continuous-space, Fourier transform of the convolution is the product of the Four transforms.

$$\mathcal{F}[f(x,y)*h(x,y)] = F(u,v)H(u,v)$$

  - So if

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

    is the output of an LT1 transformation with PSF h(x, y) to an input image f(x, y), then

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

  - In other words, output spectrum G(u, v) is the product of the input spectrum F(u, v) and the transfer function H(u, v).
  - So the FT can be used as a computational tool to simplify the convolution operation.

$$\begin{aligned} g(x,y) &= f(x,y)*h(x,y) \\ &= \mathcal{F}^{-1}\{\mathcal{F}[f(x,y)*h(x,y)]\} \\ &= \mathcal{F}^{-1}\{F(u,v)H(u,v)\} \\ &= \mathcal{F}^{-1}\{\mathcal{F}[f(x,y)]\mathcal{F}[h(x,y)]\} \end{aligned}$$

- **Correlation:** In continuous-space, correlation between two images f(x, y) and h(x, y) is defined as: $r_{fh}(x,y) = f(x,y) \circ h(x,y) = f^*(-x,-y) * h(x,y)$
- $$= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f^*(\alpha,\beta) h(x+\alpha, y+\beta) d\alpha d\beta$$

  - Therefore,

$$\mathcal{F}\left|r_{fh}(x,y)\right| = \mathcal{F}\left[f(x,y) \circ h(x,y)\right]$$
$$= \mathcal{F}\left[f^*(-x,-y) * h(x,y)\right]$$
$$= \mathcal{F}\left[f^*(-x,-y)\right]\mathcal{F}\left[h(x,y)\right]$$
$$= \mathcal{F}^*(u,v)H(u,v)$$

  - $r_{ff}(x,y)$ is usually called the **auto-correlation** of image f(x, y) (with itself). $r_{fh}(x,y)$ is called the **cross-correlation** between f(x, y) and h(x, y).

  - Roughly speaking, $r_{fh}(x,y)$ measures the degree of similarity between images f(x, y) and h(x, y). Large values of $r_{fh}(x,y)$ would indicate that the images are very similar.

  - This is usually used in template matching, where h(x, y) is a template shape whose presence we want to detect in the image f(x, y).

  - Location where $r_{fh}(x,y)$ is high (peaks of the cross-correlation function) are most likely to be the location of shape h(x, y) in image f(x, y).

f(m, n), m = 0, 1,…, M-1, n = 0, 1,…,N-1 is an M x N image and

h(m, n), m = 0, 1,…, K-1, n = 0, 1,…,L-1 is an K x L mask. Then

$$g(m, n) = f(m, n) * h(m, n)$$

is a (M + K -1) x (N + L -1) image.

- So if we want a convolution property for discrete images --- something like

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

  we need to have G(u, v) to be of size (M + K -1) x (N + L-1) (since g(m, n) has that dimension).

- Therefore, we should require that F(u, v) and H(u, v) also have the same dimension, i.e. (M + K-1) x (N + L -1).

- So we **zero-pad** the images f(m, n), h(m, n), so that they are of size (M + K-1) x (N + L-1). Let $f_e(m, n)$ and $h_e(m, n)$ be the zero-padded (or extended images). Take their 2D-DFTs to obtain F(u, v) and H(u, v), each of size (M + K-1) x (N + L-1). Then

$$\begin{aligned} g(m,n) &= \mathcal{F}^{-1}\{G(u,v)\} = \mathcal{F}^{-1}\{\mathrm{F}(u,v)H(u,v)\} \\ &= \mathcal{F}^{-1}\{\mathcal{F}[f_e(m,n)]\mathcal{F}[h_e(m,n)]\} \end{aligned}$$

- Similar comments hold for correlation of discrete images as well.

- **Multiplication:** (In continuous-domain) This is the dual of the convolution property. Multiplication of two images corresponds to convolving their spectra.

$$\mathcal{F}\{f(x,y)h(x,y)\} = F(u,v) * G(u,v)$$

- Average value: The average pixel value in an image:

$$\bar{f} = \frac{1}{N^2}\sum_{n=0}^{N-1}\sum_{m=0}^{N-1} f(m,n)$$

Notice that (substitute u = v = 0 in the definition):

$$F(0,0) = \sum_{n=0}^{N-1}\sum_{m=0}^{N-1} f(m,n) = N^2\bar{f} \Rightarrow \bar{f} = \frac{1}{N^2}F(0,0)$$

- Differentiation: (Only in continuous-domain):
Derivatives are normally used for detecting edges in an image. An edge is the boundary of an object and denotes an abrupt change in gray-value. Hence it is a region with high value of derivative.

$$\mathcal{F}\left\{\frac{\partial f(x,y)}{\partial x}\right\} = (-j2\pi u)F(u,v)$$

$$\mathcal{F}\left\{\frac{\partial f(x,y)}{\partial y}\right\} = (-j2\pi v)F(u,v)$$

$$\mathcal{F}\left\{\frac{\partial^2 f(x,y)}{\partial x^2}\right\} = (-4\pi^2 u^2)F(u,v)$$

$$\mathcal{F}\left\{\frac{\partial^2 f(x,y)}{\partial y^2}\right\} = (-4\pi^2 v^2)F(u,v)$$

$$\nabla^2 f(x,y) = \frac{\partial^2 f(x,y)}{\partial x^2} + \frac{\partial^2 f(x,y)}{\partial y^2} = -4\pi^2(u^2 + v^2)F(u,v)$$

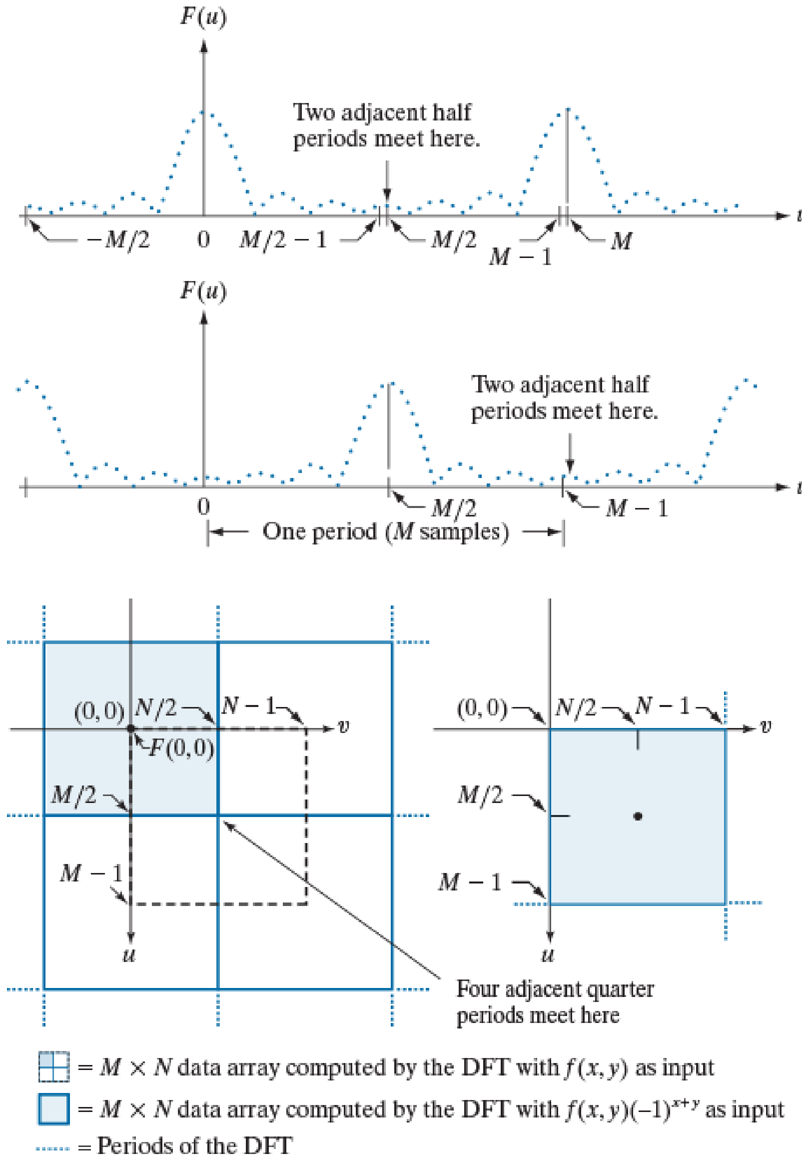


FIGURE 4.22 Centering the Fourier transform. (a) A 1-D DFT showing an infinite number of periods. (b) Shifted DFT obtained by multiplying $f(x)$ by $(-1)^x$ before computing $F(u)$. (c) A 2-D DFT showing an infinite number of periods. The area within the dashed rectangle is the data array, $F(u, v)$, obtained with Eq. (4-67) with an image $f(x, y)$ as the input. This array consists of four quarter periods. (d) Shifted array 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).

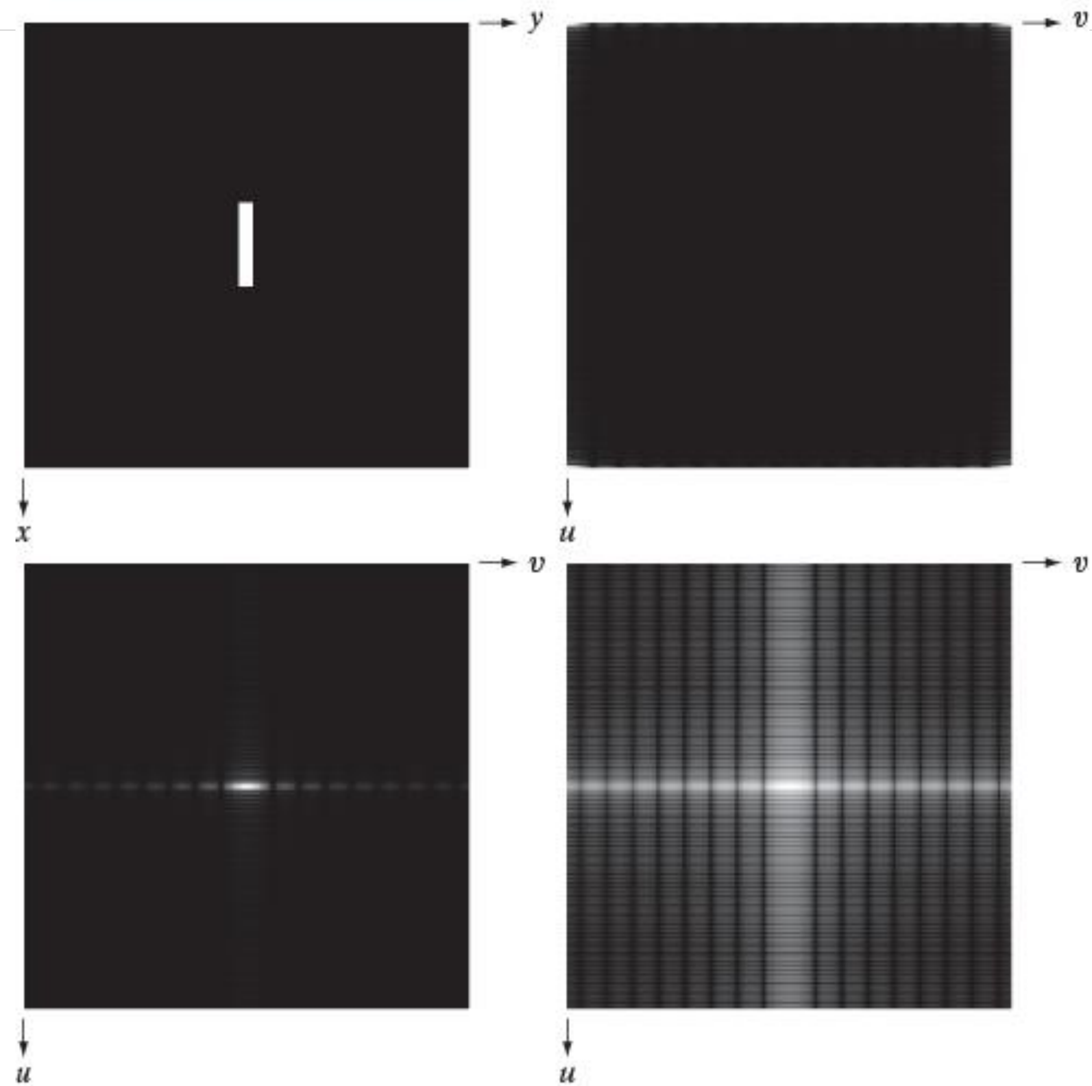
TABLE 4.1
Some symmetry properties of the 2-D DFT and its inverse. $R(u,v)$ and $I(u,v)$ are the real and imaginary parts of $F(u,v)$, respectively. Use of the word *complex* indicates that a function has nonzero real and imaginary parts.

	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. As before, "$\Leftrightarrow$" indicates a Fourier transform pair.

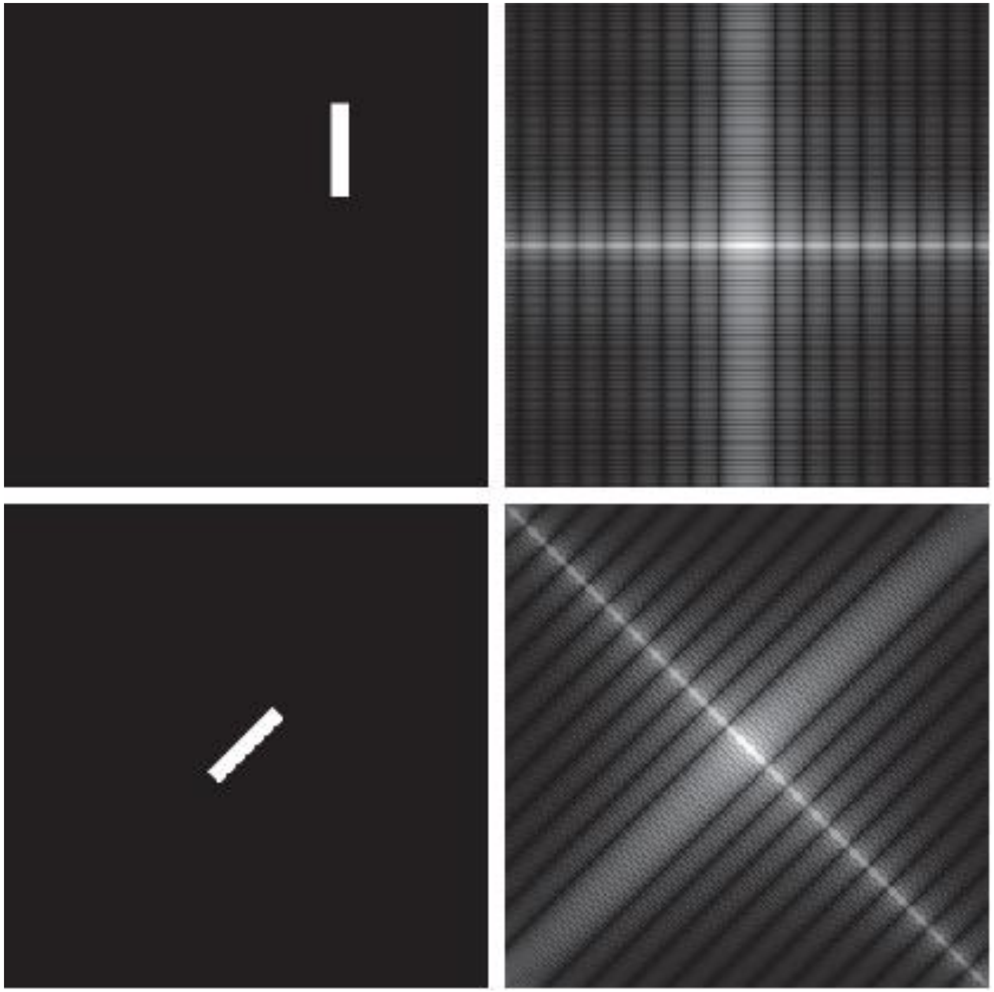
TABLE 4.2
1-D examples of some of the properties in Table 4.1.

Property	$f(x)$		$F(u)$
3	$\{1, 2, 3, 4\}$	$\Leftrightarrow$	$\{(10+0j),(-2+2j),(-2+0j),(-2-2j)\}$
4	$\{1j, 2j, 3j, 4j\}$	$\Leftrightarrow$	$\{(0+2.5j),(.5-.5j),(0-.5j),(-.5-.5j)\}$
8	$\{2, 1, 1, 1\}$	$\Leftrightarrow$	$\{5, 1, 1, 1\}$
9	$\{0, -1, 0, 1\}$	$\Leftrightarrow$	$\{(0+0j),(0+2j),(0+0j),(0-2j)\}$
10	$\{2j, 1j, 1j, 1j\}$	$\Leftrightarrow$	$\{5j, j, j, j\}$
11	$\{0j, -1j, 0j, 1j\}$	$\Leftrightarrow$	$\{0, -2, 0, 2\}$
12	$\{(4+4j), (3+2j), (0+2j), (3+2j)\}$	$\Leftrightarrow$	$\{(10+10j),(4+2j),(-2+2j),(4+2j)\}$
13	$\{(0+0j),(1+1j),(0+0j),(-1-j)\}$	$\Leftrightarrow$	$\{(0+0j),(2-2j),(0+0j),(-2+2j)\}$



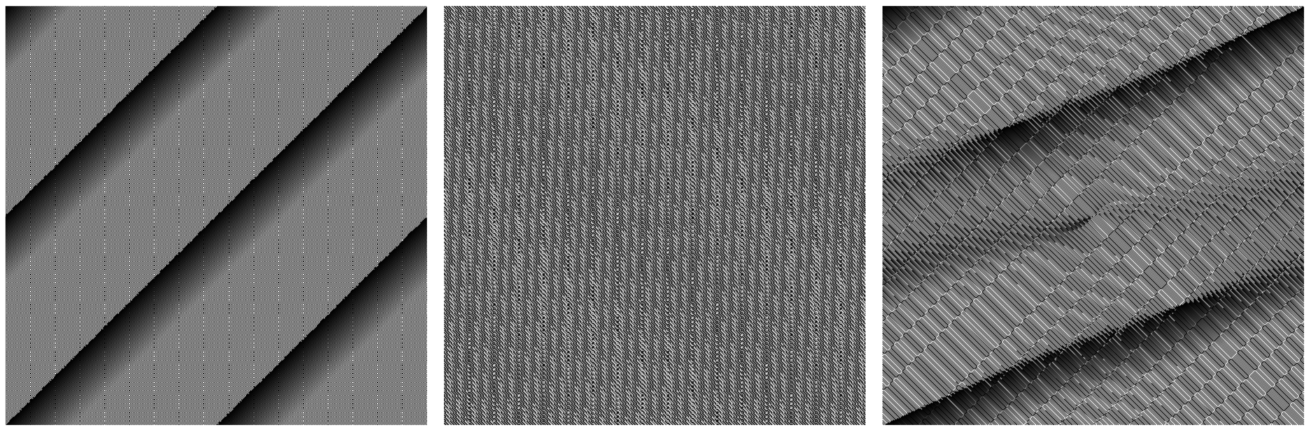
a b
c d

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).



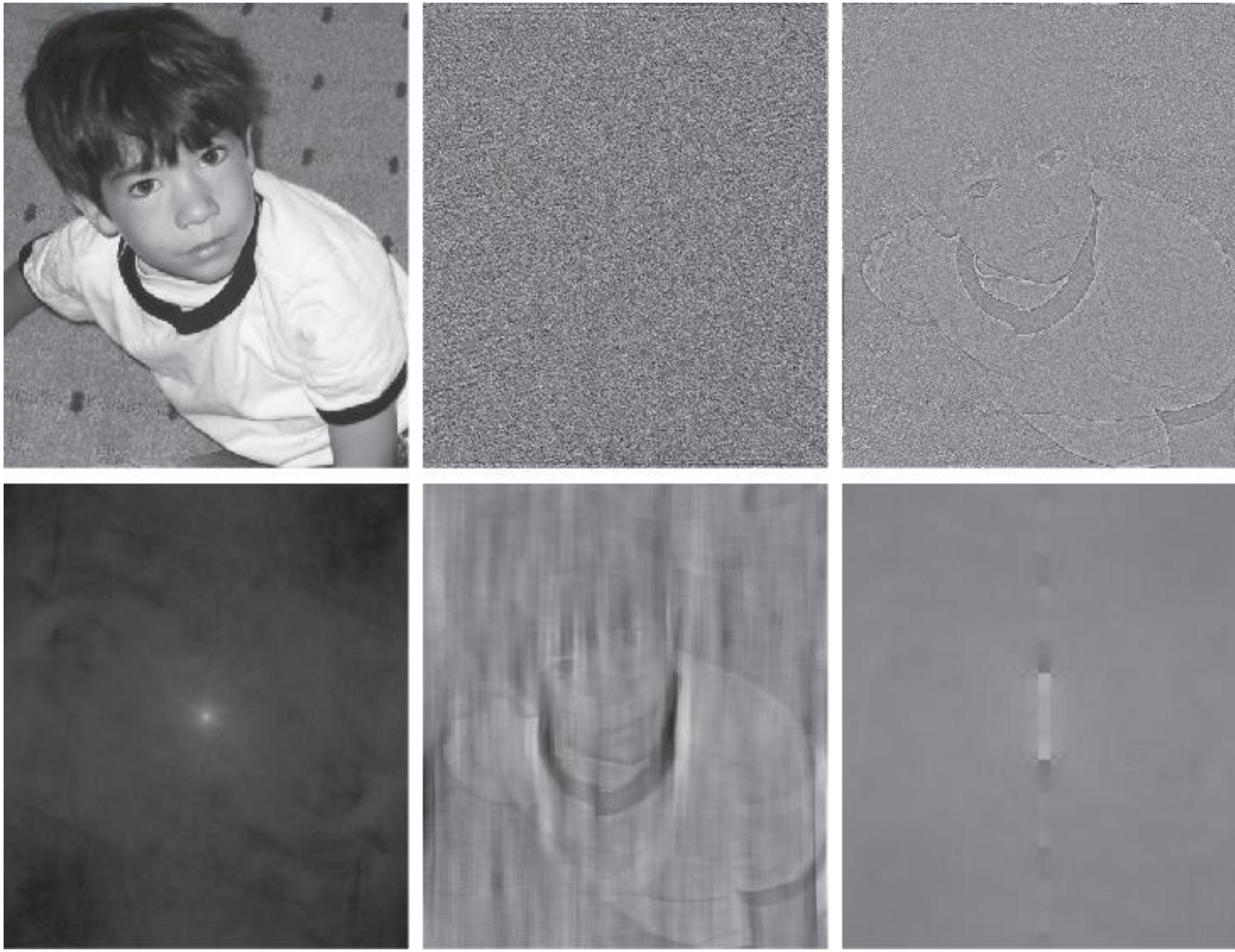
a b
c d

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).



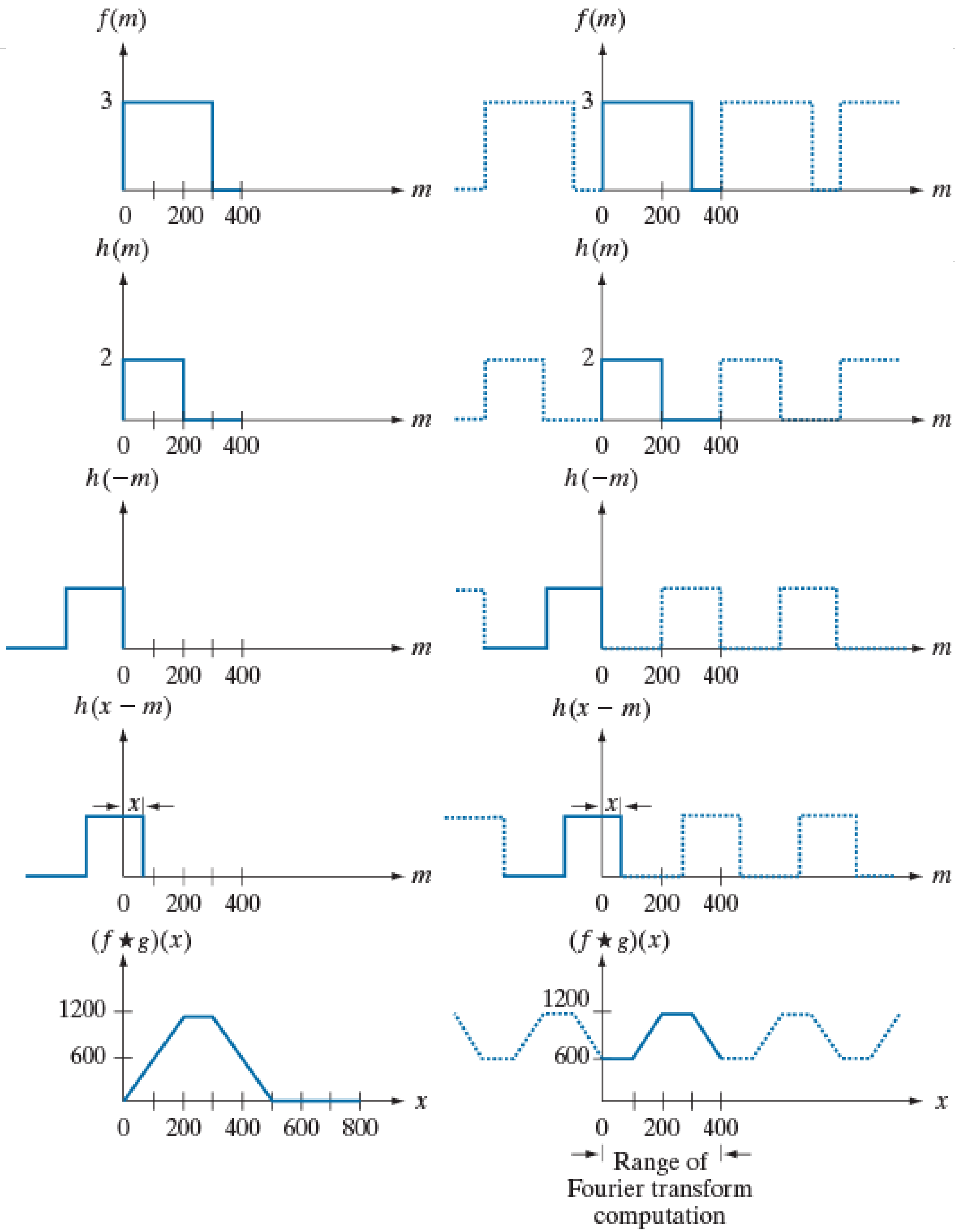
a b c

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



a b c
d e f

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.



a f
b g
c h
d i
e j

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.

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	$\lvert F(u,v)\rvert=\left[R^2(u,v)+I^2(u,v)\right]^{1/2}$ $R=\text{Real}(F);\ I=\text{Imag}(F)$
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)	$F(u,v)=F(u+k_1M,v)=F(u,v+k_2N)=F(u+k_1,v+k_2N)$ $f(x,y)=f(x+k_1M,y)=f(x,y+k_2N)=f(x+k_1M,y+k_2N)$
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	$(f\,☆\,h)(x,y)=\sum_{m=0}^{M-1}\sum_{n=0}^{N-1} f^*(m,n)h(x+m,y+n)$
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	$MNf^*(x,y)=\sum_{u=0}^{M-1}\sum_{v=0}^{N-1} F^*(u,v)e^{-j2\pi(ux/M+vy/N)}$ 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.

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	$af_1(x,y) + bf_2(x,y) \Leftrightarrow aF_1(u,v) + bF_2(u,v)$
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	$f(r, \theta+\theta_0) \Leftrightarrow F(\omega, \varphi+\theta_0)$ $r=\sqrt{x^2+y^2} \quad \theta=\tan^{-1}(y/x) \quad \omega=\sqrt{u^2+v^2} \quad \varphi=\tan^{-1}(v/u)$
6)	Convolution theorem†	$f \star h)(x,y) \Leftrightarrow (F \bullet H)(u,v)$ $(f \bullet h)(x,y) \Leftrightarrow (1/MN)[(F \star H)(u,v)]$
7)	Correlation theorem†	$(f \star h)(x,y) \Leftrightarrow (F^* \bullet H)(u,v)$ $(f^* \bullet h)(x,y) \Leftrightarrow (1/MN)[(F \star H)(u,v)]$
8)	Discrete unit impulse	$\delta(x,y) \Leftrightarrow 1$ $1 \Leftrightarrow MN\delta(u,v)$
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 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 $f(\pm\infty, \pm\infty) = 0$.	$\left(\dfrac{\partial}{\partial t}\right)^m \left(\dfrac{\partial}{\partial z}\right)^n f(t,z) \Leftrightarrow (j2\pi\mu)^m (j2\pi\nu)^n F(\mu,\nu)$ $\dfrac{\partial^m f(t,z)}{\partial t^m} \Leftrightarrow (j2\pi\mu)^m F(\mu,\nu); \; \dfrac{\partial^n f(t,z)}{\partial z^m} \Leftrightarrow (j2\pi\nu)^n F(\mu,\nu)$
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)

† Assumes that $f(x,y)$ and $h(x,y)$ have been properly padded. Convolution is associative, commutative, and distributive. Correlation is distributive (see Table 3.5). The products are elementwise products (see Section 2.6).