Last Week
Math. of 2D Convolution/Correlation

Convolution

$$g(x,y) = h * f(x,y) = \sum_{j=-n}^{n} \sum_{i=-m}^{m} h(i,j) f(x-i, y-j)$$

Correlation

$$g(x,y) = h \circ f(x,y) = \sum_{j=-n}^{n} \sum_{i=-m}^{m} h(i,j) f(x+i, y+j)$$

Note: When the filter is symmetric: correlation = convolution!

1	1	1
1	1	1
1	1	1

1

2	3	2
-1	0	-1
2	3	2

Smoothing Spatial Filters

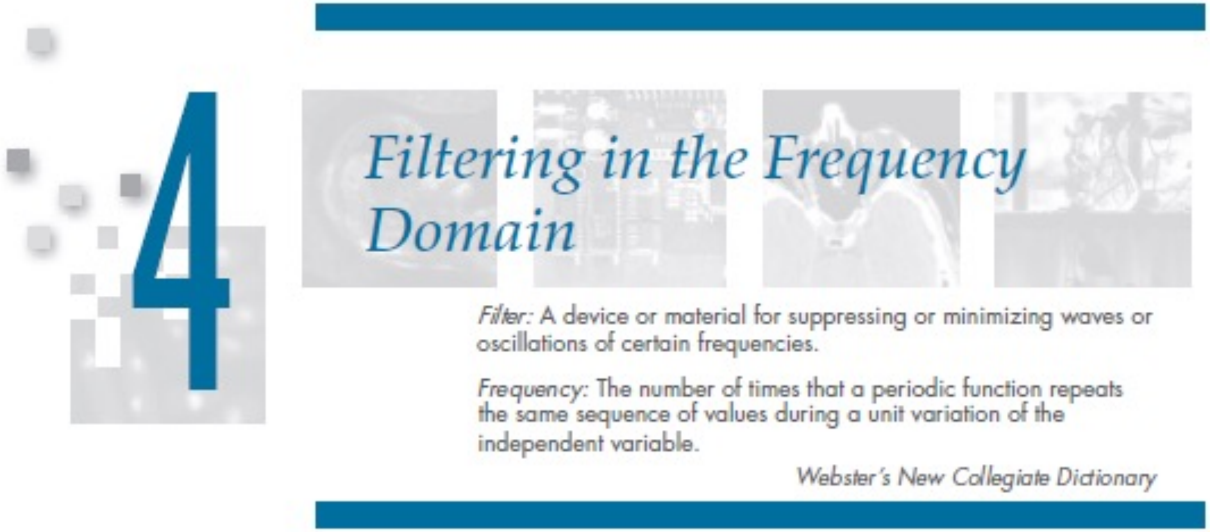
$$\frac{1}{9} \times \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array} \qquad \frac{1}{16} \times \begin{array}{|c|c|c|} \hline 1 & 2 & 1 \\ \hline 2 & 4 & 2 \\ \hline 1 & 2 & 1 \\ \hline \end{array}$$

a b

FIGURE 3.32 Two 3×3 smoothing (averaging) filter masks. The constant multiplier in front of each mask is equal to 1 divided by the sum of the values of its coefficients, as is required to compute an average.

Laplacian Image Enhancement

Images taken from Gonzalez & Woods, Digital Image Processing (2002)



4

Filtering in the Frequency Domain

Filter: A device or material for suppressing or minimizing waves or oscillations of certain frequencies.

Frequency: The number of times that a periodic function repeats the same sequence of values during a unit variation of the independent variable.

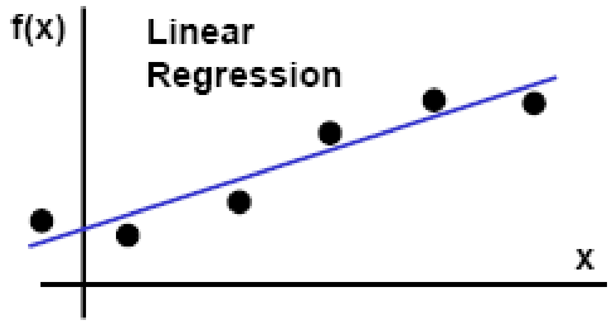
Webster's New Collegiate Dictionary

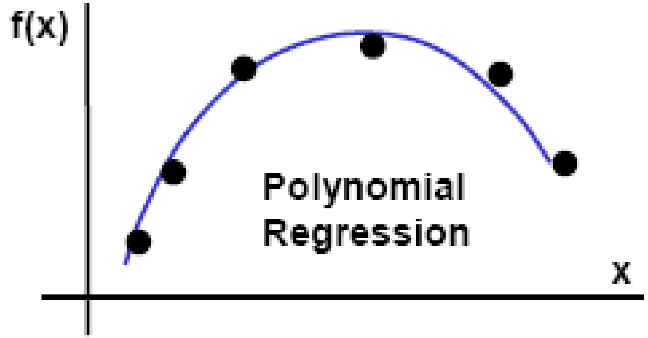
Upon completion of this chapter, readers should:

- Understand the meaning of frequency domain filtering, and how it differs from filtering in the spatial domain. Be familiar with the concepts of sampling, function reconstruction, and aliasing
- Understand convolution in the frequency domain, and how it is related to filtering.
- Know how to obtain frequency domain filter functions from spatial kernels, and vice versa
- Be able to construct filter transfer functions directly in the frequency domain.
- Understand why image padding is important. Know the steps required to perform filtering in the frequency domain.
- Understand when frequency domain filtering is superior to filtering in the spatial domain. Be familiar with other filtering techniques in the frequency domain, such as unsharp masking and homomorphic filtering.
- Understand the origin and mechanics of the fast Fourier transform, and how to use it effectively in image processing.

CURVE FITTING

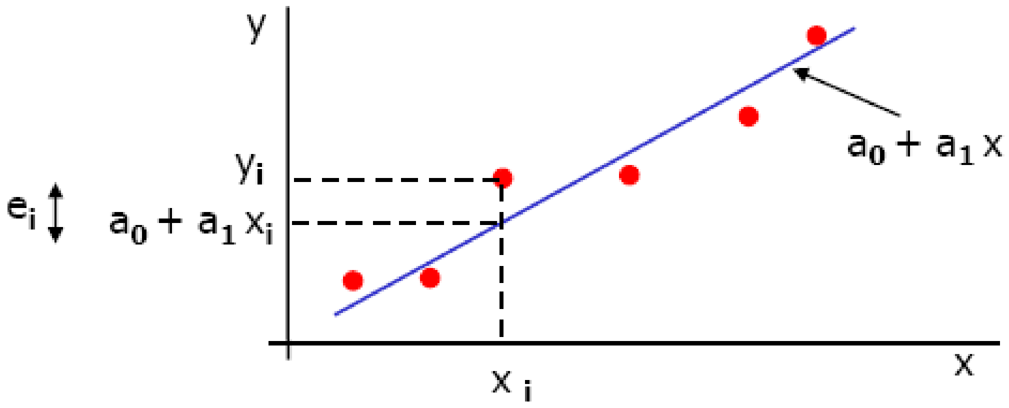
- Curve fitting is expresssing a discrete set of data points as a continuous function.
- Regression: Mainly used with experimental data which might have significant amount of error (noise). No need to find a function that passes through the data points.





LEAST SQUARES REGRESSION

- Fitting a straight line to a set of data set (paired data points).(x_1, y_1), (x_2, y_2), (x_3, y_3), . . . , (x_n, y_n)



a_0 : y-intercept (unknown)

a_1 : slope (unknown)

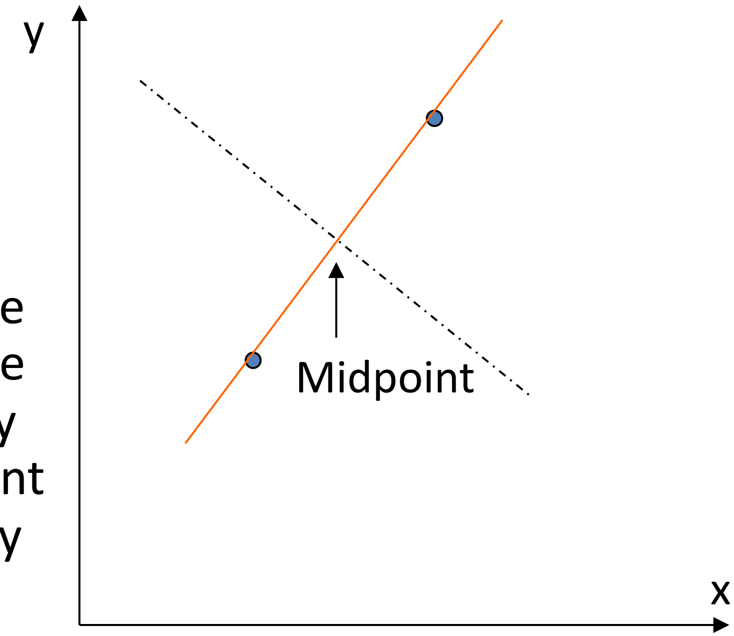
$e_i = y_i - a_0 - a_1 x_i$ Error (deviation) for the i^{th} data point

CRITERIA FOR A "BEST FIT"

1. Minimize the sum of the residual errors

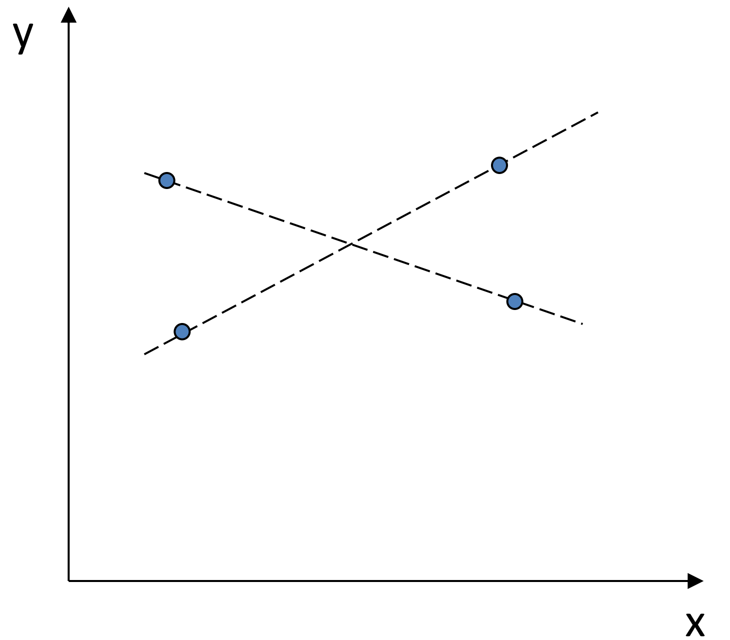
$$\sum_{i=1}^{n} e_i = \sum_{i=1}^{n} (y_i - a_0 - a_1 x_i)$$

This is an inadequate criterion. For example in Figure the fit of a straight line to two points is shown. The best fit is the line connecting the points. However, any straight line passing through the midpoint of the connecting line (except a perfectly vertical line) results in a minimum value of zero because the errors cancel.



2) Another logical criterion might be to minimize the sum of the absolute values of the discrepancies

$$\sum_{i=1}^{n} |e_i| = \sum_{i=1}^{n} |y_i - a_0 - a_1 x_i|$$



The figure shows why this criterion is inadequate. For the four points shown, any straight line falling within the dashed lines will minimize the sum of the absolute values. Thus, this criterion is also does not yield a unique best fit.

LEAST SQUARES FIT

$$S_r = \sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n} (y_i - a_0 - a_1 x_i)^2$$ Sum of squares of the residuals

- Determine the unknowns a_0 and a_1 by minimizing S_r.
- To do this set the derivatives of S_r wrt a_0 and a_1 to zero.

$$\frac{\partial S_r}{\partial a_0} = -2\sum_{i=1}^{n}(y_i - a_0 - a_1 x_i) = 0 \quad \rightarrow \quad n\, a_0 + \left(\sum x_i\right) a_1 = \sum y_i$$

$$\frac{\partial S_r}{\partial a_1} = -2\sum_{i=1}^{n}[(y_i - a_0 - a_1 x_i) x_i] \quad \rightarrow \quad \left(\sum x_i\right) a_0 + \left(\sum x_i^2\right) a_1 = \sum x_i y_i$$

or $$\begin{bmatrix} n & \sum x_i \\ \sum x_i & \sum x_i^2 \end{bmatrix} \begin{Bmatrix} a_0 \\ a_1 \end{Bmatrix} = \begin{Bmatrix} \sum y_i \\ \sum x_i y_i \end{Bmatrix}$$ These are called the normal equations.

- Solve these for a_0 and a_1. The results are

$$\boxed{a_1 = \frac{n\sum (x_i y_i) - \sum x_i \sum y_i}{n\sum x_i^2 - (\sum x_i)^2} \qquad a_0 = \bar{y} - a_1 \bar{x}}$$

where y-bar and x-bar are the means of y and x, respectively.

Polynomial Regression

- Some engineering data is poorly represented by a straight line. A curve (polynomial) may be better suited to fit the data. The least squares method can be extended to fit the data to higher order polynomials.
- As an example let us consider a second order polynomial to fit the data points:

$$y = a_0 + a_1 x + a_2 x^2$$

$$\text{Minimize error:} \quad S_r = \sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n} (y_i - a_0 - a_1 x_i - a_2 x_i^2)^2$$

$$\frac{\partial S_r}{\partial a_o} = -2\sum (y_i - a_o - a_1 x_i - a_2 x_i^2) = 0$$

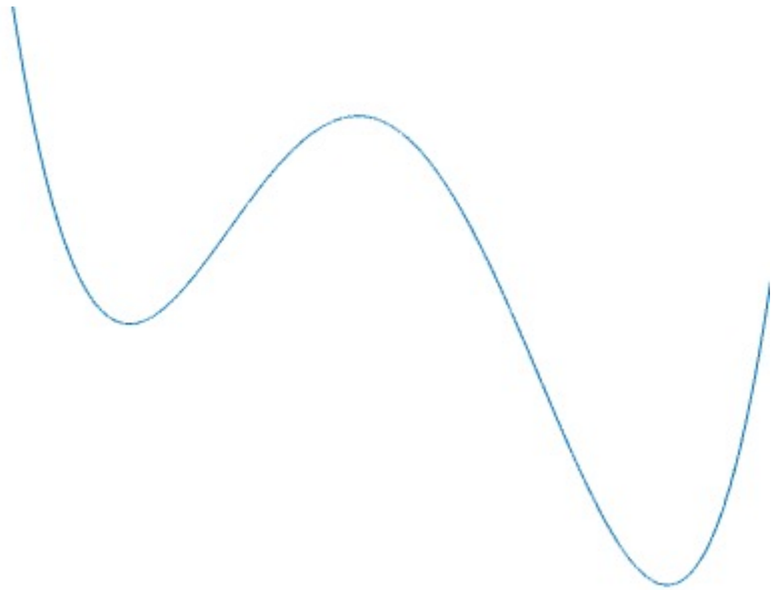
$$\frac{\partial S_r}{\partial a_1} = -2\sum x_i (y_i - a_o - a_1 x_i - a_2 x_i^2) = 0$$

$$\frac{\partial S_r}{\partial a_2} = -2\sum x_i^2 (y_i - a_o - a_1 x_i - a_2 x_i^2) = 0$$

$$n a_0 + \left(\sum x_i\right) a_1 + \left(\sum x_i^2\right) a_2 = \sum y_i$$

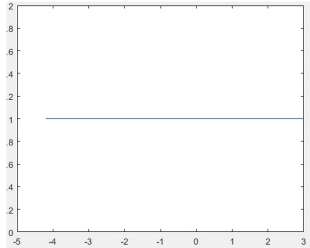
$$\left(\sum x_i\right) a_0 + \left(\sum x_i^2\right) a_1 + \left(\sum x_i^3\right) a_2 = \sum x_i y_i$$

$$\left(\sum x_i^2\right) a_0 + \left(\sum x_i^3\right) a_1 + \left(\sum x_i^4\right) a_2 = \sum x_i^2 y_i$$

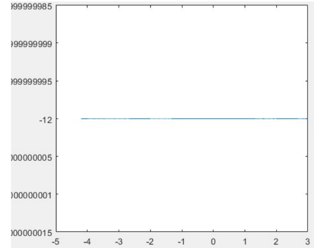


$$y = x^4 + 3x^3 - 9x^2 - 23x - 12$$

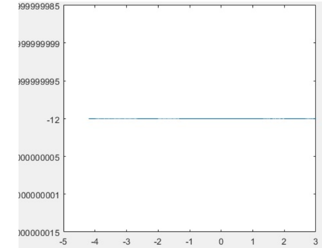
y=x^0



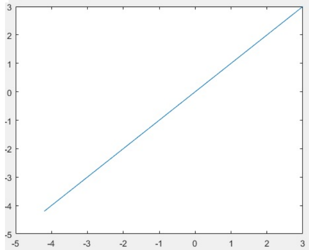
y=-12*x^0



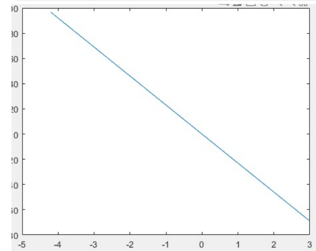
y=-12*x^0



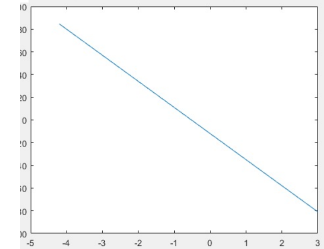
y=x^1



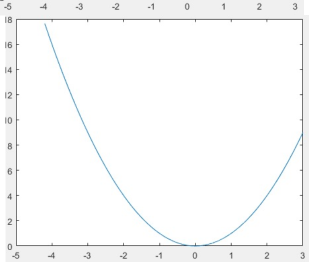
y=-23*x^1



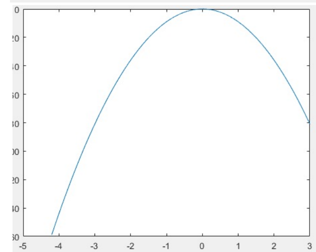
y=-12*x^0-23*x^1



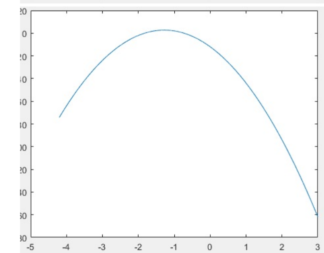
y=x^2



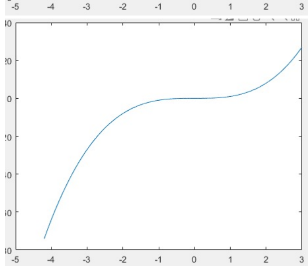
y=-9*x^2



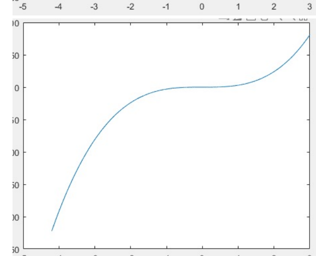
y=-12*x^0-23*x^1-9*x^2



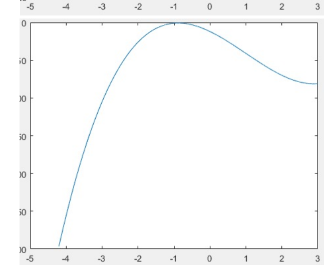
y=x^3



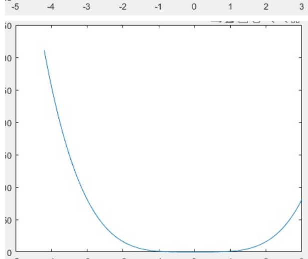
y=3*x^3



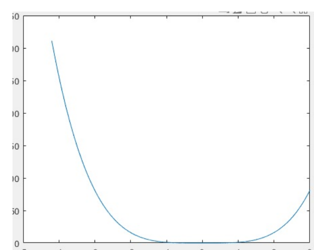
y=-12*x^0-23*x^1-9*x^2+3*x^3



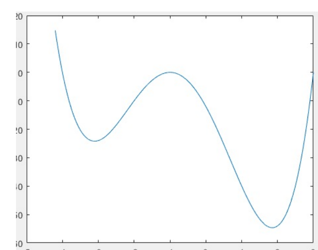
y=x^4



y=1*x^4



y=-12*x^0-23*x^1-9*x^2+3*x^3+1*x^4



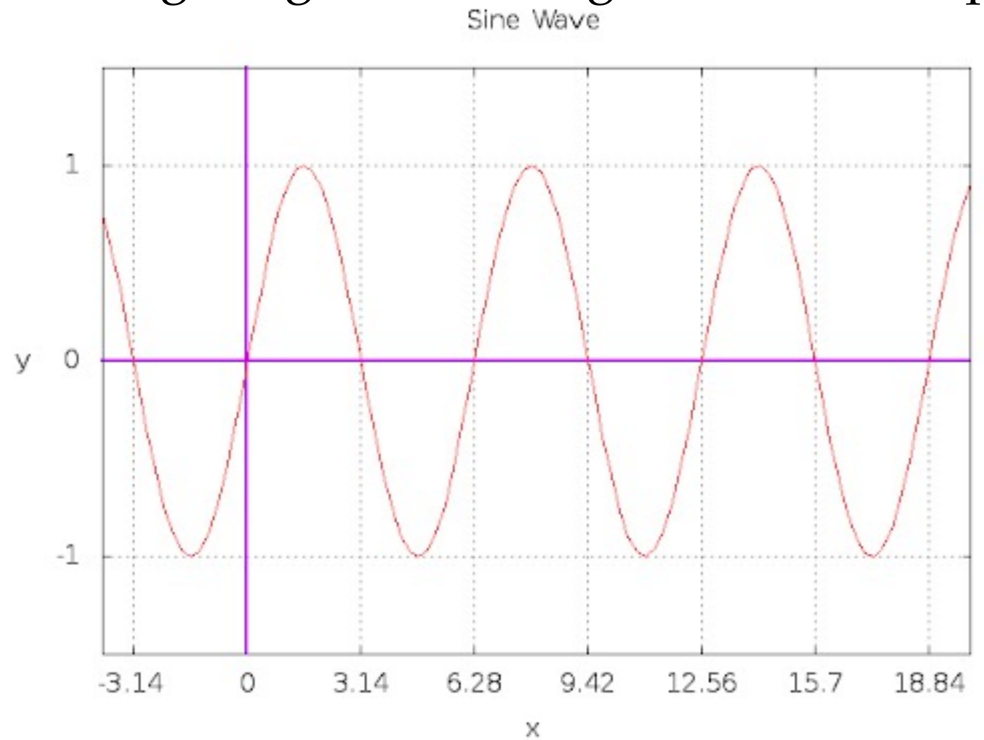
Any signal / function / curve / data can be represented as a **linear combination** of a set of **basic components**

$$F(x) = a_n X^n + a_{n-1} X^{n-1} + a_{n-2} X^{n-1} \ldots\ldots + a_1 X + a_0$$

$$F(X) = \sum_{i=0}^{n} a_n x^n$$

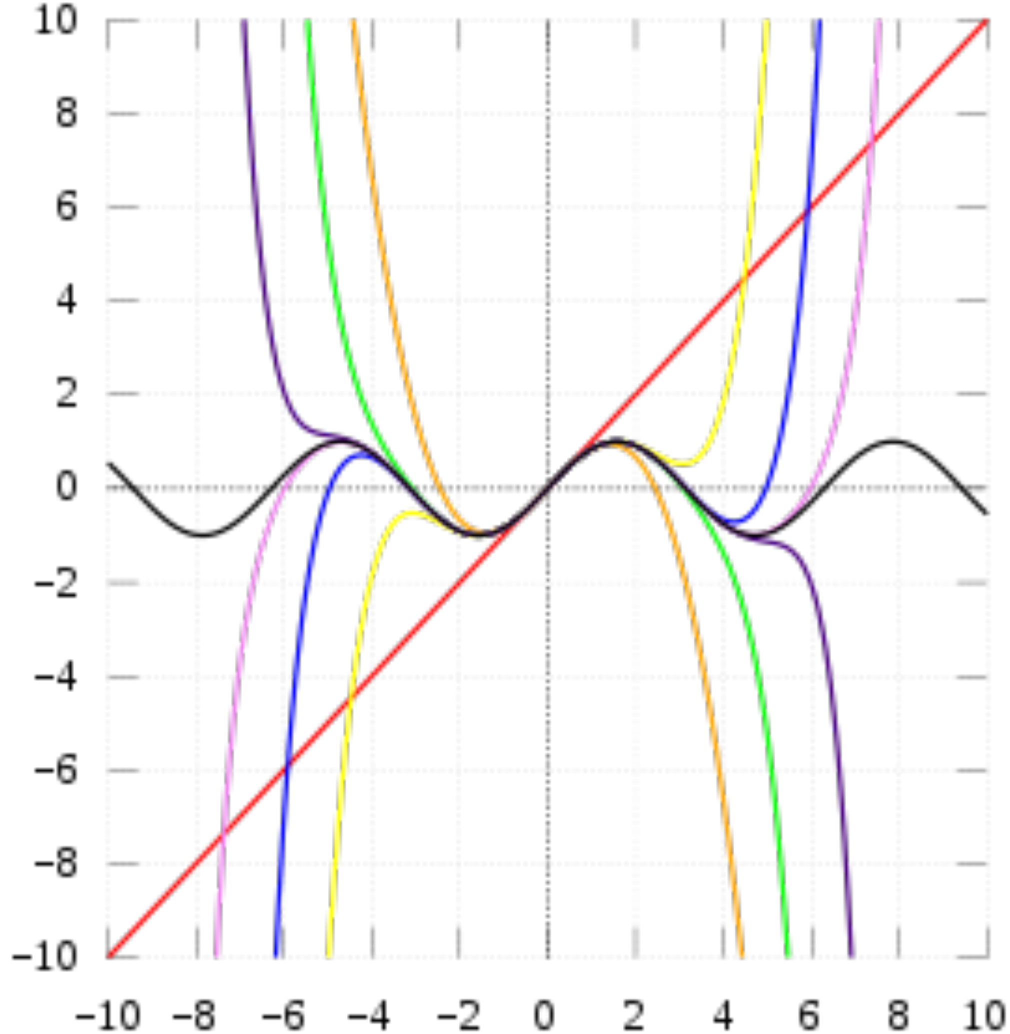
–**components**: polynomial patterns

–**coefficients**: weighting factors assigned to the components

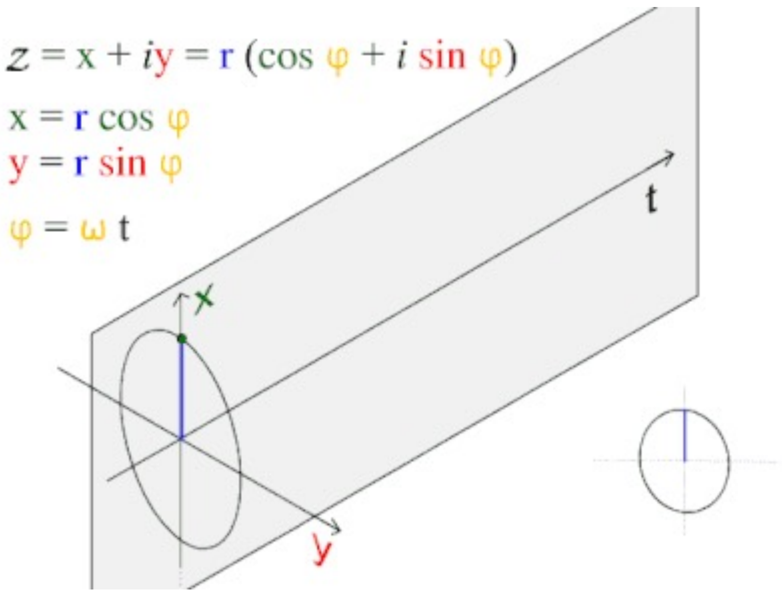


how many parameters are required to represent the sinus curve

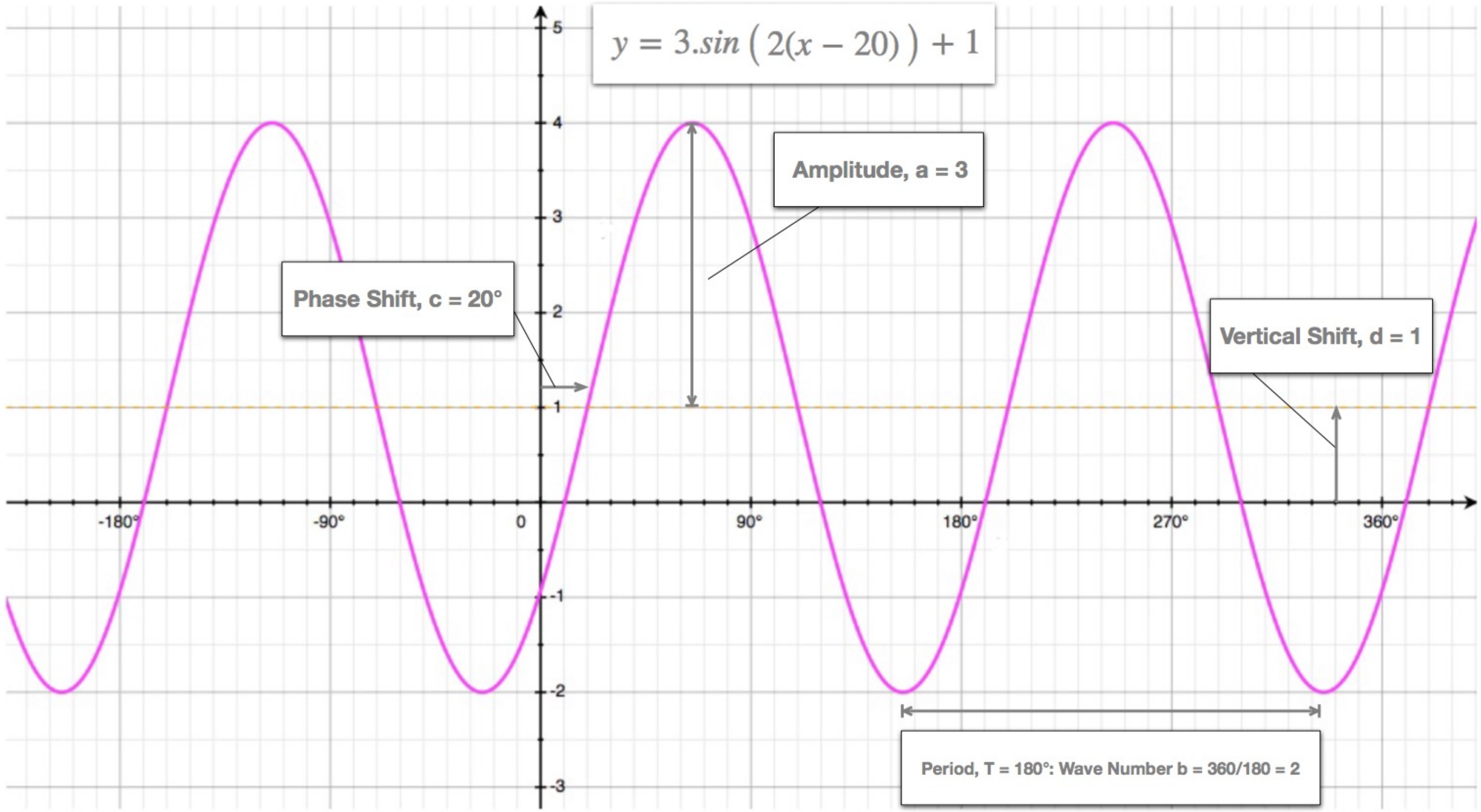
$$\sin(x) = 0 + 1x + 0x^2 + \frac{-1}{3!}x^3 + 0x^4 + \cdots$$
$$= x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \cdots$$



As the degree of the Taylor polynomial rises, it approaches the correct function. This image shows *sin x* and its Taylor approximations, polynomials of degree **1**, **3**, **5**, **7**, **9**, **11**, and **13**.



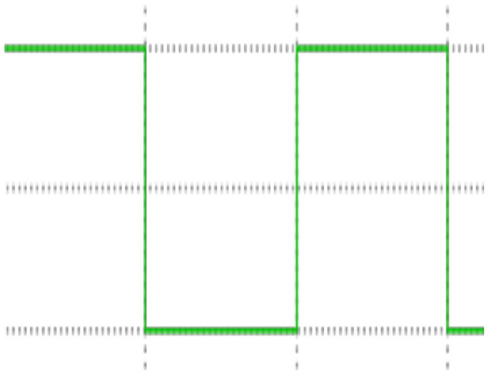
Illustrating the cosine wave's fundamental relationship to the circle.



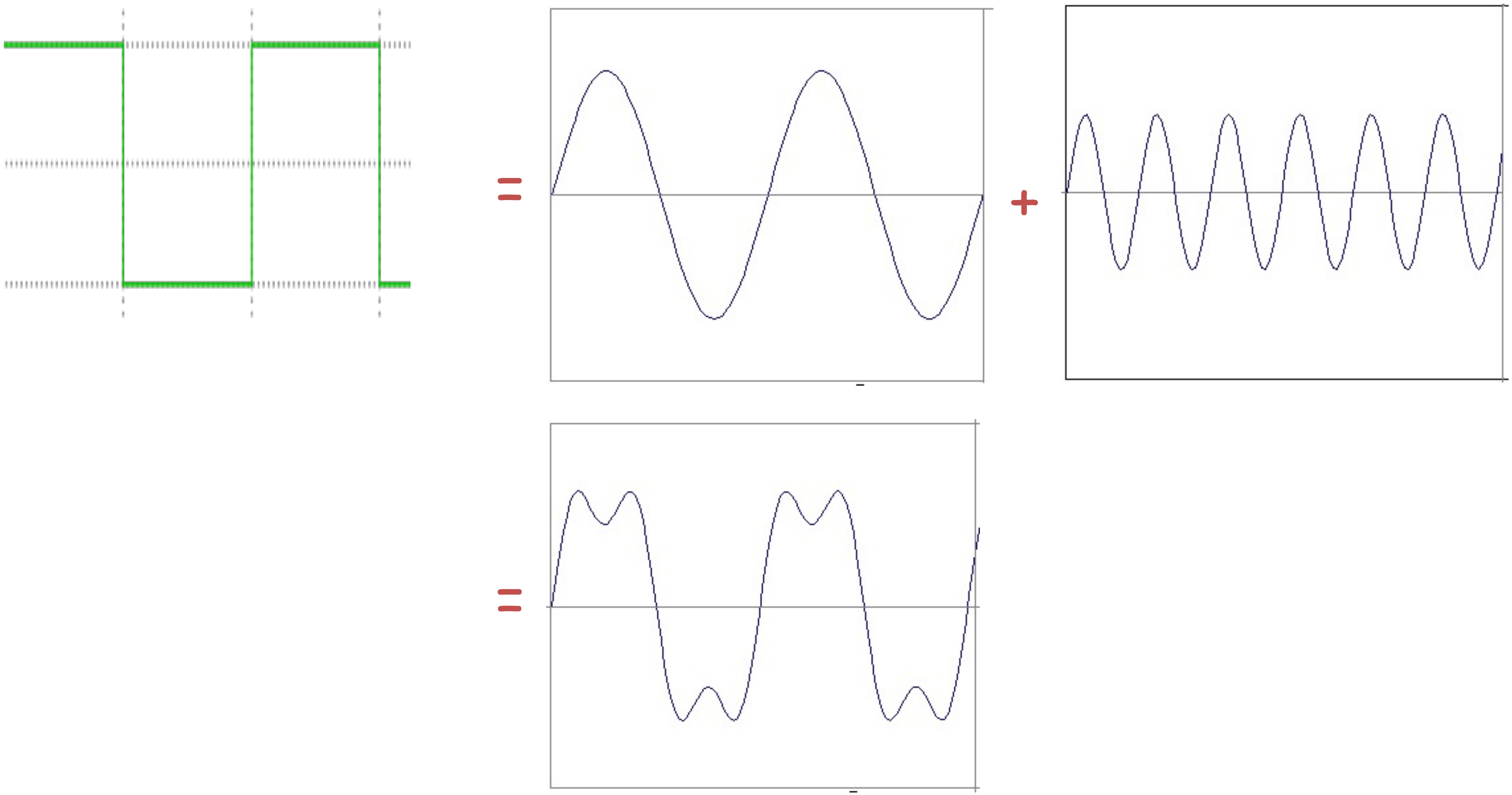
$$y(t) = A\sin(2\pi ft + \varphi) = A\sin(\omega t + \varphi)$$

- *A*, *amplitude*, the peak deviation of the function from zero.
- *f*, *ordinary frequency*, the number of oscillations (cycles) that occur each second of time.
- $\omega = 2\pi f$, *angular frequency*, the rate of change of the function argument in units of radians per second
- φ, *phase*, specifies (in radians) where in its cycle the oscillation is at $t = 0$.

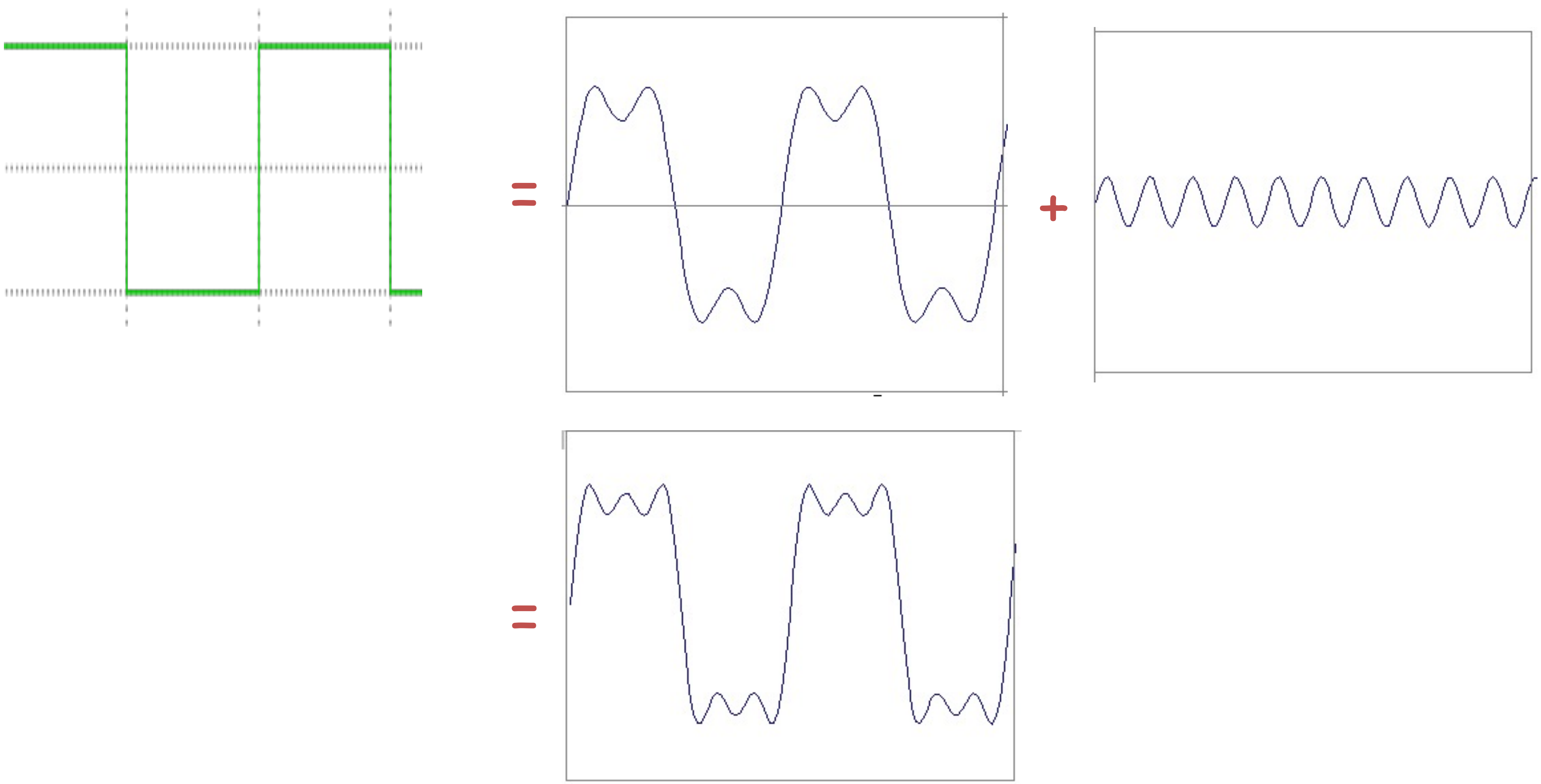
Frequency Spectra



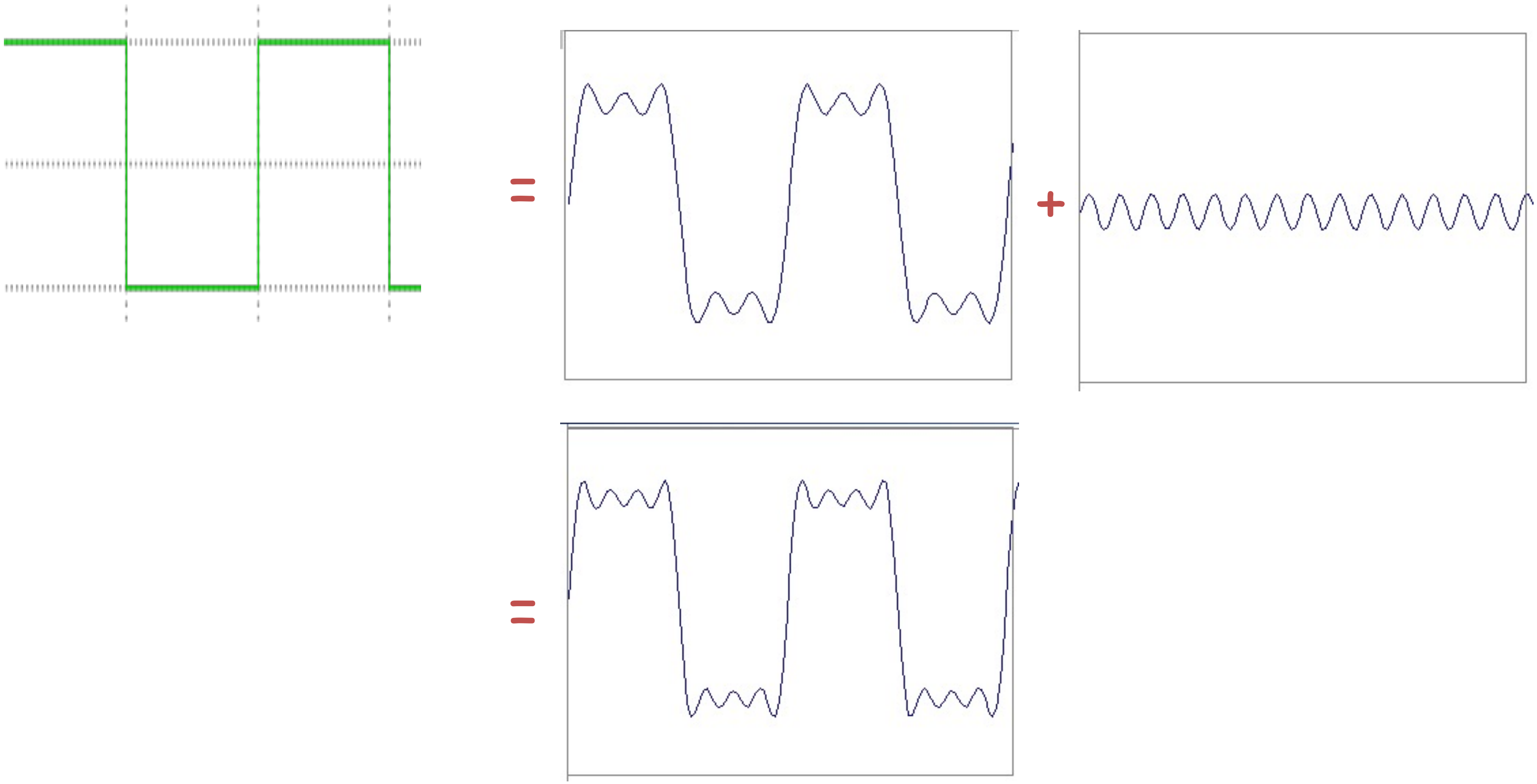
Frequency Spectra



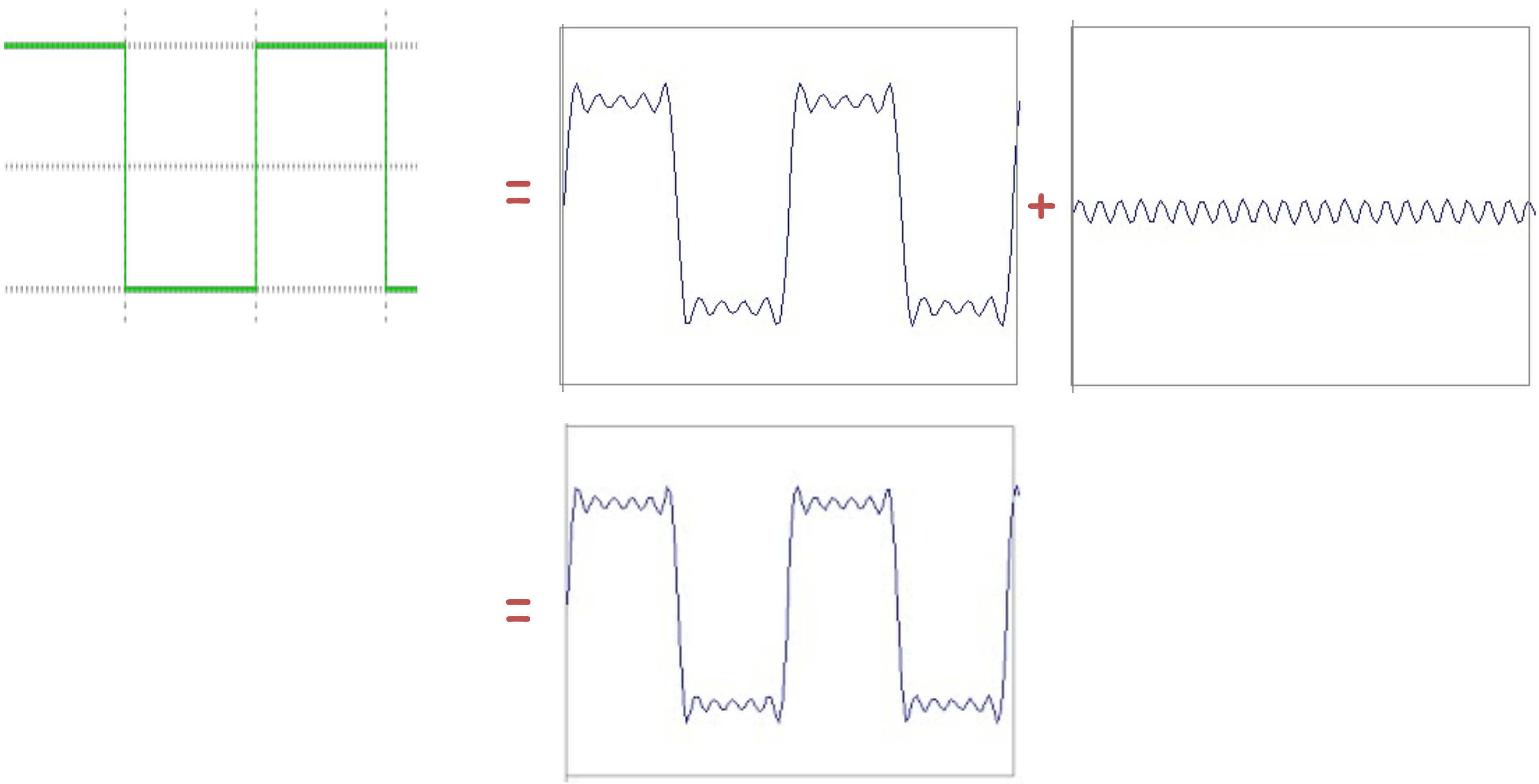
Frequency Spectra



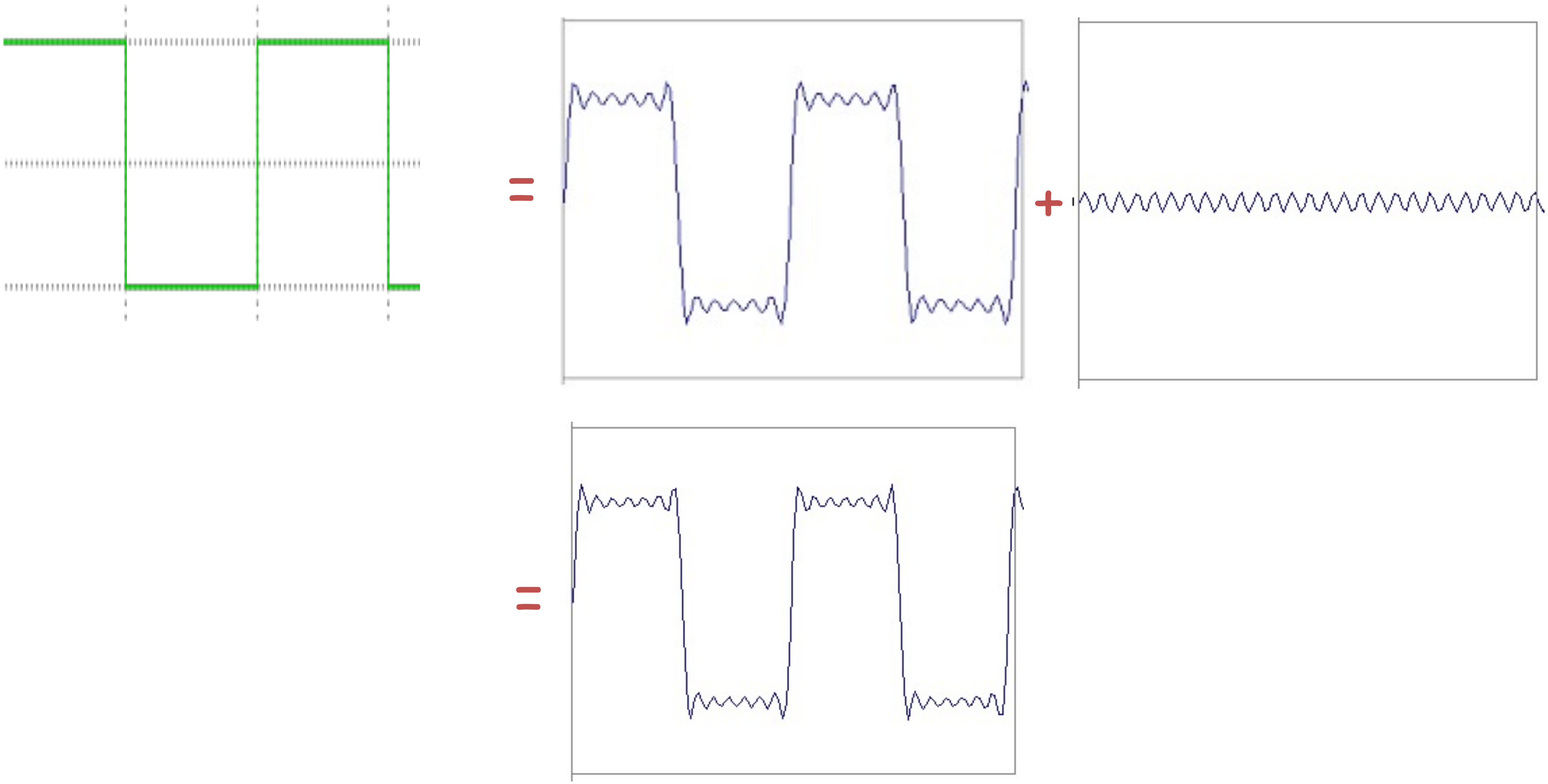
Frequency Spectra



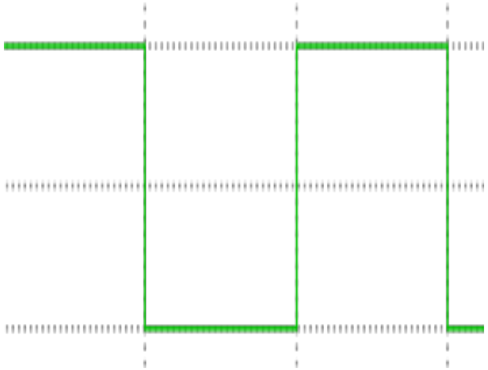
Frequency Spectra



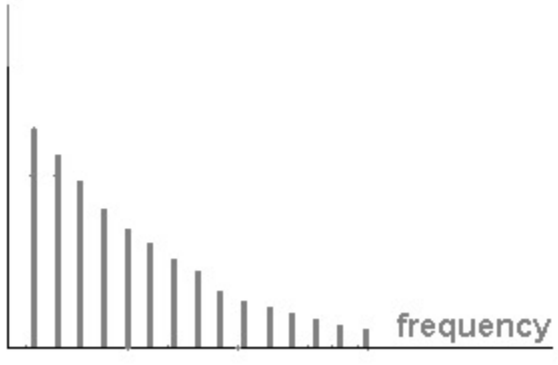
Frequency Spectra



Frequency Spectra



$$= \boxed{A\sum_{k=1}^{\infty}\frac{1}{k}\sin(2\pi kt)}$$



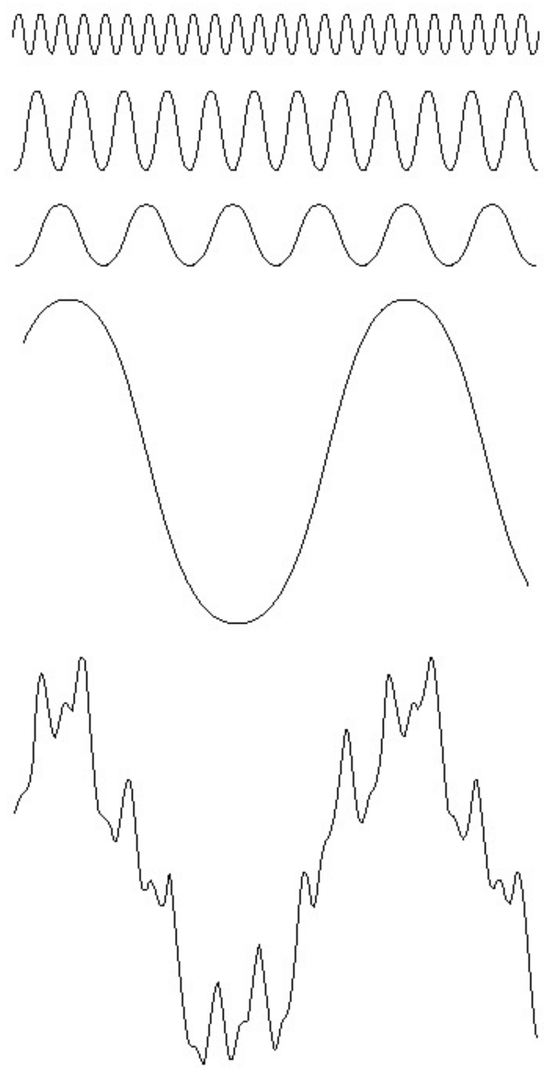


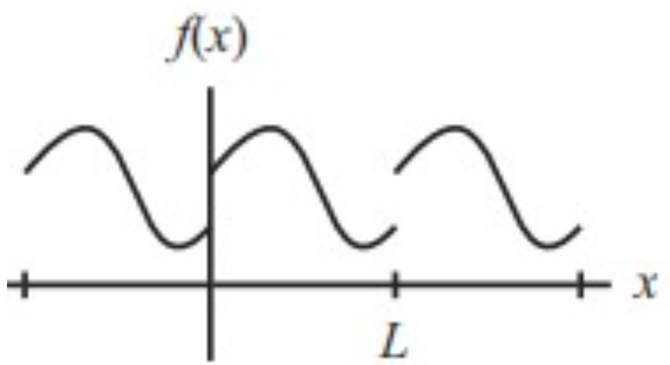
FIGURE 4.1 The function at the bottom is the sum of the four functions above it. Fourier's idea in 1807 that periodic functions could be represented as a weighted sum of sines and cosines was met with skepticism.

Any signal / function / curve / data can be represented as a **linear combination** of a set of **basic components**

$$f(x) = a_0 + \sum_{n=1}^{\infty} \left[a_n \cos\left(\frac{2\pi n x}{L}\right) + b_n \sin\left(\frac{2\pi n x}{L}\right) \right]$$

– **Fourier components**: sinusoidal patterns

– **Fourier coefficients**: weighting factors assigned to the Fourier components



Orthogonal functions

$$\int_0^L \sin\left(\frac{2\pi nx}{L}\right)\cos\left(\frac{2\pi mx}{L}\right)dx. \Rightarrow \frac{1}{2}\int_0^L \left[\sin\left((n+m)\frac{2\pi x}{L}\right)+\sin\left((n-m)\frac{2\pi x}{L}\right)\right]dx.$$

$$\int_0^L \cos\left(\frac{2\pi nx}{L}\right)\cos\left(\frac{2\pi mx}{L}\right)dx \Rightarrow \frac{1}{2}\int_0^L \left[\cos\left((n+m)\frac{2\pi x}{L}\right)+\cos\left((n-m)\frac{2\pi x}{L}\right)\right]dx,$$

$$\int_0^L \sin\left(\frac{2\pi nx}{L}\right)\sin\left(\frac{2\pi mx}{L}\right)dx \Rightarrow \frac{1}{2}\int_0^L \left[-\cos\left((n+m)\frac{2\pi x}{L}\right)+\cos\left((n-m)\frac{2\pi x}{L}\right)\right]dx,$$

$$\begin{aligned}
\int_0^L \sin\left(\frac{2\pi nx}{L}\right)\cos\left(\frac{2\pi mx}{L}\right)dx &= 0,\\
\int_0^L \cos\left(\frac{2\pi nx}{L}\right)\cos\left(\frac{2\pi mx}{L}\right)dx &= \frac{L}{2}\\
\int_0^L \sin\left(\frac{2\pi nx}{L}\right)\sin\left(\frac{2\pi mx}{L}\right)dx &= \frac{L}{2}
\end{aligned}$$

$$f(x) = a_0 + \sum_{n=1}^{\infty} \left[a_n \cos\left(\frac{2\pi n x}{L}\right) + b_n \sin\left(\frac{2\pi n x}{L}\right) \right]$$

Calculating a0

$$\int_0^L f(x)\,dx = a_0 L \quad \Longrightarrow \quad \boxed{a_0 = \frac{1}{L}\int_0^L f(x)\,dx}$$

Calculating an

$$\int_0^L f(x)\cos\left(\frac{2\pi m x}{L}\right)dx = a_m \frac{L}{2} \quad \Longrightarrow \quad \boxed{a_n = \frac{2}{L}\int_0^L f(x)\cos\left(\frac{2\pi n x}{L}\right)dx}$$

Calculating bn

$$\int_0^L f(x)\sin\left(\frac{2\pi m x}{L}\right)dx = b_m \frac{L}{2} \quad \Longrightarrow \quad \boxed{b_n = \frac{2}{L}\int_0^L f(x)\sin\left(\frac{2\pi n x}{L}\right)dx}$$

Fourier Series Theorem

- Any periodic signal can be expressed as a weighted sum (infinite) of sine and cosine functions of varying frequency

$$f(t) = a_0 + \sum_{n=1}^{\infty} a_n cos(nf_0 t) + \sum_{n=1}^{\infty} b_n sin(nf_0 t)$$

f_0 is called the "fundamental frequency"

a_n and b_n are the weights of the expansion

The Continuous Fourier Transform

1D Continuous Fourier Transform:

$$f(x) = \int_{u=-\infty}^{\infty} \mathrm{F}(\mathrm{u})\mathrm{e}^{+2\pi iux} du = \Im^{-1}(F(u))$$

The <u>Inverse</u> Fourier Transform

$$F(u) = \int_{x=-\infty}^{\infty} \mathrm{f}(\mathrm{x})\mathrm{e}^{-2\pi iux} dx = \Im(f(x))$$

The Fourier Transform

2D Continuous Fourier Transform:

$$f(x,y) = \int_{u=-\infty}^{\infty} \int_{v=-\infty}^{\infty} \mathrm{F}(\mathrm{u},\mathrm{v})\mathrm{e}^{+2\pi i(ux+\mathrm{vy})} dudv$$

The Inverse Transform

$$F(u,v) = \int_{x=-\infty}^{\infty} \int_{y=-\infty}^{\infty} \mathrm{f}(\mathrm{x},\mathrm{y})\mathrm{e}^{-2\pi i(ux+\mathrm{vy})} dxdy$$

The Transform

The Discrete Fourier Transform

1D Discrete Fourier Transform:

$$F(u) = \frac{1}{N} \sum_{x=0}^{N-1} f(x) e^{\frac{-2\pi i u x}{N}}$$

(u = 0,..., N-1)

$$f(x) = \sum_{u=0}^{N-1} F(u) e^{\frac{+2\pi i u x}{N}}$$

(x = 0,..., N-1)

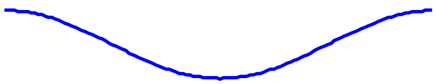
2D Discrete Fourier Transform:

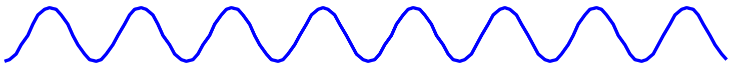
$$F(u,v) = \frac{1}{N} \frac{1}{M} \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} f(x,y) e^{-2\pi i \left(\frac{ux}{N} + \frac{vy}{M}\right)}$$

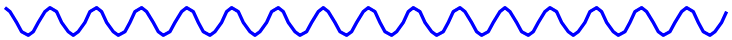
(u = 0,..., N-1; v = 0,…,M-1)

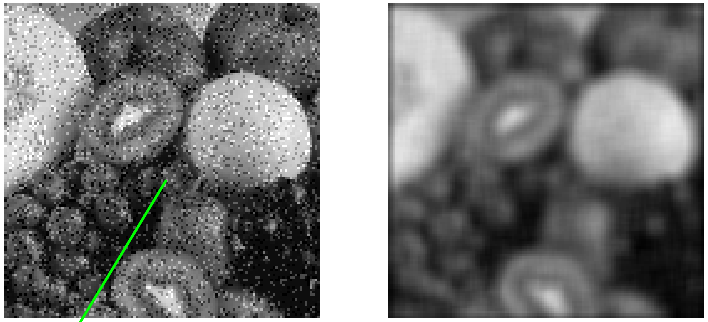
$$f(x,y) = \sum_{u=0}^{N-1} \sum_{v=0}^{M-1} F(u,v) e^{+2\pi i \left(\frac{ux}{N} + \frac{vy}{M}\right)}$$

(x = 0,..., N-1; y = 0,…,M-1)

Low frequency  plain wall

High frequency   picket fence

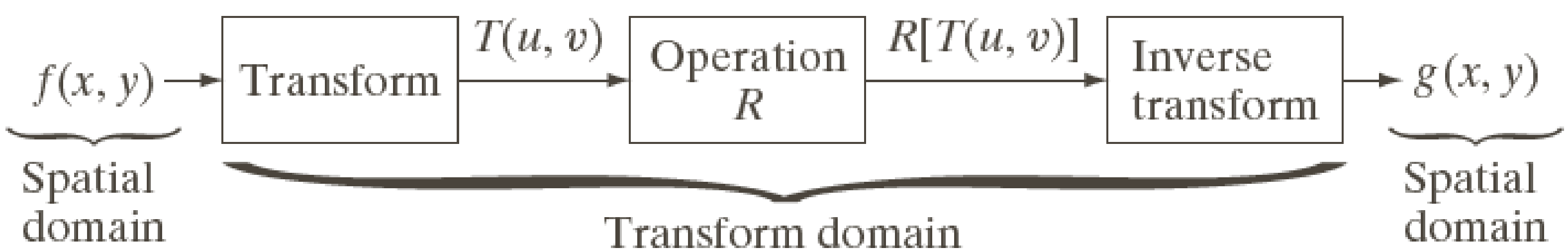
Very High frequency   checkerboard



Higher frequencies due to sharp image variations (e.g., edges, noise, etc.)

Image Transforms

- Many times, image processing tasks are best performed in a domain other than the *spatial domain.*
- Main steps
  – Transform the image
  – Carry the task(s) in the *transformed domain*.
  – Apply *inverse transform* to return to the spatial



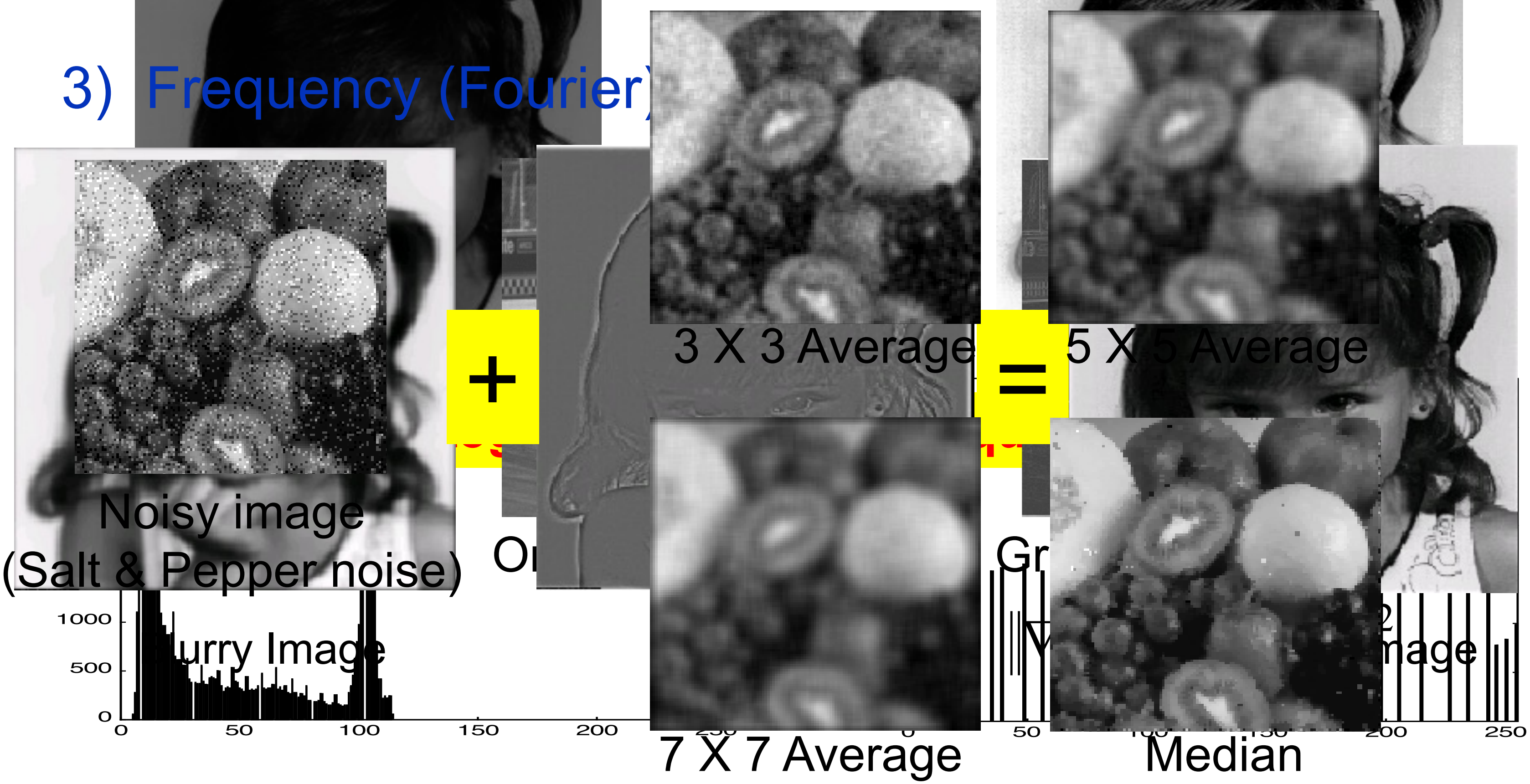
Why Do Transforms?

- Fast computation
  - E.g., convolution vs. multiplication for filter with wide support
- Conceptual insights for various image processing
  - E.g., spatial frequency info. (smooth, moderate change, fast change, etc.)
- Obtain transformed data as measurement
  - E.g., blurred images, radiology images (medical and astrophysics)
  - Often need inverse transform
  - May need to get assistance from other transforms

- <u>For efficient storage and transmission</u>
  - Pick a few "representatives" (basis)
  - Just store/send the "contribution" from each basis

Image Operations in Different Domains

1) Gray value (histogram) domain
   - Histogram stretching, equalization, specification, etc...
2) Spatial (image) domain
   - Average filter, median filter, gradient, laplacian, etc…
3) Frequency (Fourier)



The Continuous Fourier Transform

1D Continuous Fourier Transform:

$$f(x) = \int_{u=-\infty}^{\infty} \mathrm{F(u)e}^{+2\pi iux} du = \Im^{-1}(F(u))$$

The Inverse Fourier Transform

$$F(u) = \int_{x=-\infty}^{\infty} \mathrm{f(x)e}^{-2\pi iux} dx = \Im(f(x))$$

The Fourier Transform

$$\mathbf{e}^{2\pi iux} = \cos(2\pi ux) + i\sin(2\pi ux)$$

2D Discrete Fourier Transform:

$$F(u,v)=\frac{1}{N}\frac{1}{M}\sum_{x=0}^{N-1}\sum_{y=0}^{M-1} f(x,y)e^{-2\pi i\left(\frac{ux}{N}+\frac{vy}{M}\right)}$$

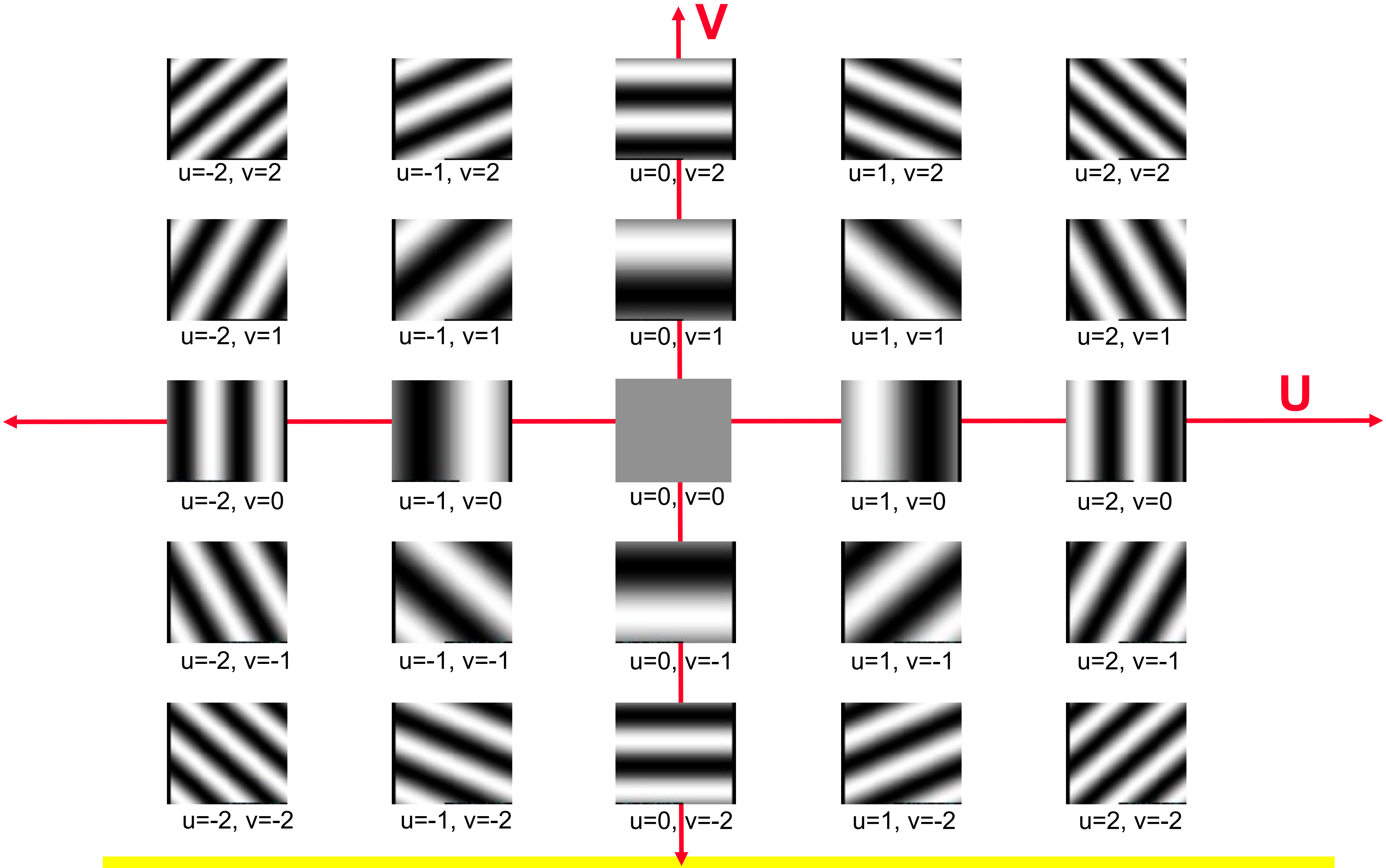
(u = 0,..., N-1; v = 0,…,M-1)

$$f(x,y)=\sum_{u=0}^{N-1}\sum_{v=0}^{M-1} F(u,v)e^{+2\pi i\left(\frac{ux}{N}+\frac{vy}{M}\right)}$$

(x = 0,..., N-1; y = 0,…,M-1)

$$\mathbf{e}^{2\pi iux}=\cos(2\pi ux)+i\sin(2\pi ux)$$

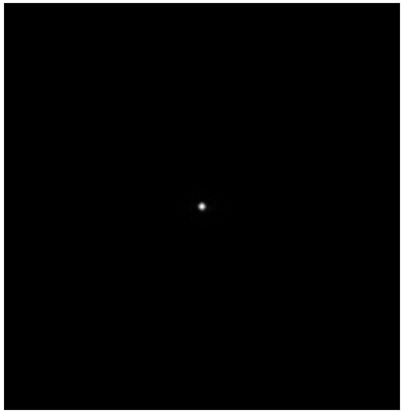
The 2D Basis Functions $e^{2\pi i(ux+vy)}$



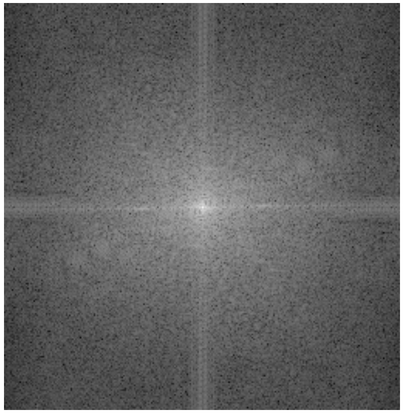
The wavelength is $1/\sqrt{u^2+v^2}$. The direction is u/v .

– Example : Fourier Transform

Original image



Fourier transform



Logarithmic operator applied

- The image contains components of all frequencies, but their magnitude gets smaller for higher frequencies.
- Low frequencies contain more image information than the higher ones.
- Two dominating directions in the Fourier image, vertical and horizontal. These originate from the regular patterns in the background.

Amplitude and phase spectrum of the Fourier transform of images

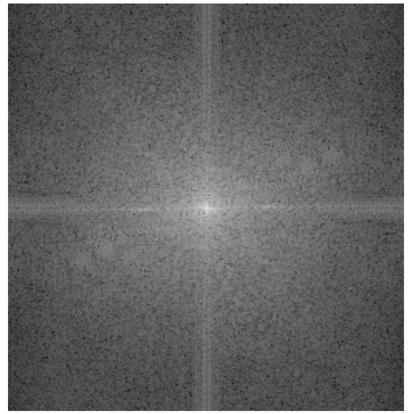
$$F(u,v) = |F(u,v)|\, e^{-j\arg[F(u,v)]}$$

$$|F(u,v)| = \sqrt{\mathrm{Re}(F(u,v))^2 + \mathrm{Im}(F(u,v))^2}$$

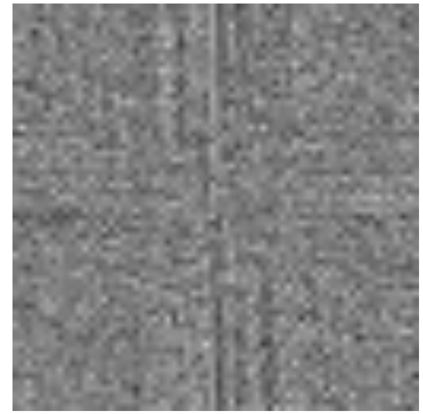
$$\arg(F(u,v)) = \arctan\frac{\mathrm{Im}(F(u,v))}{\mathrm{Re}(F(u,v))}$$

Fourier Transform

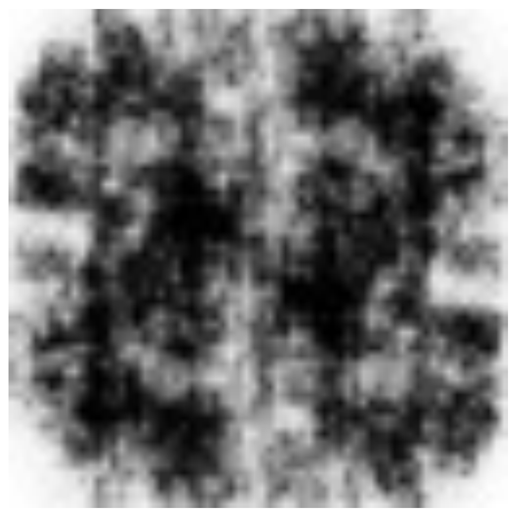
Original image



Magnitude spectrum

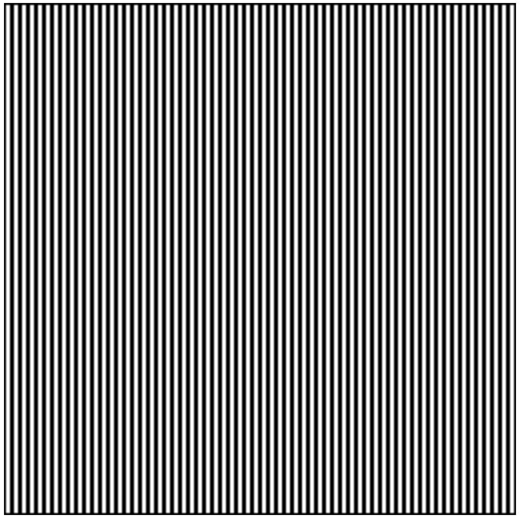


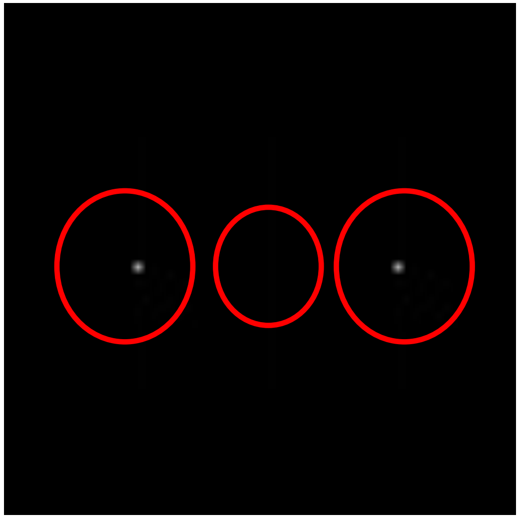
Phase spectrum



Inverse FT applied only Magnitude spectrum

Example : Fourier Transform

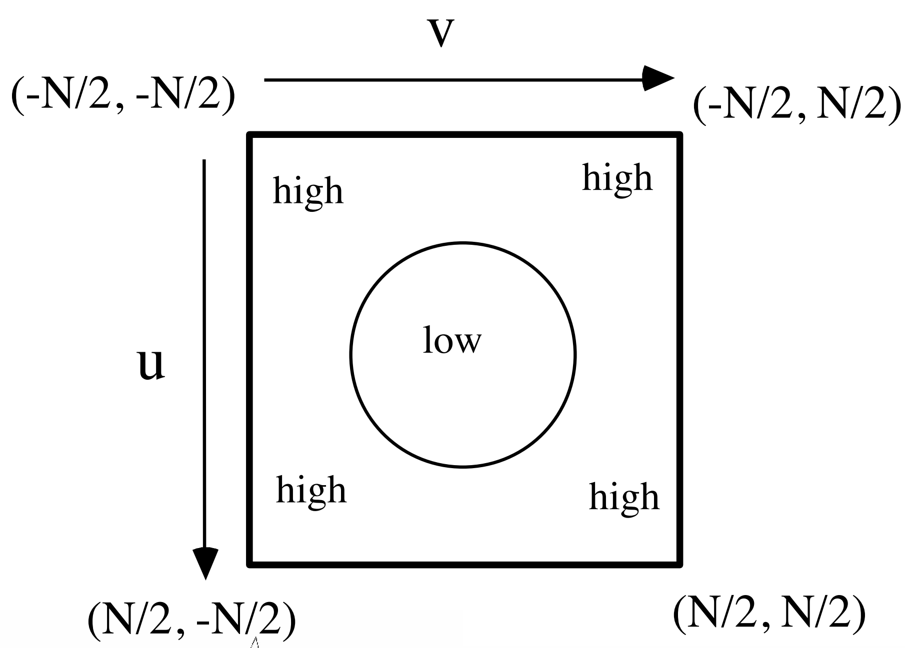




$$f_{max} = \frac{1}{2\ \text{pixels}}$$

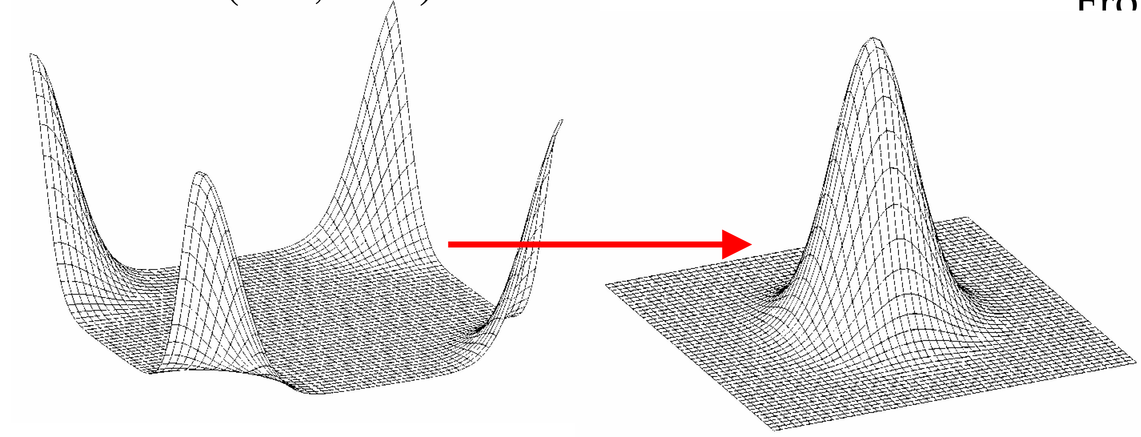
$$f = \frac{1}{4\ \text{pixels}} = \frac{f_{max}}{2}$$

Centered Representation



From Prof. Al Bovik

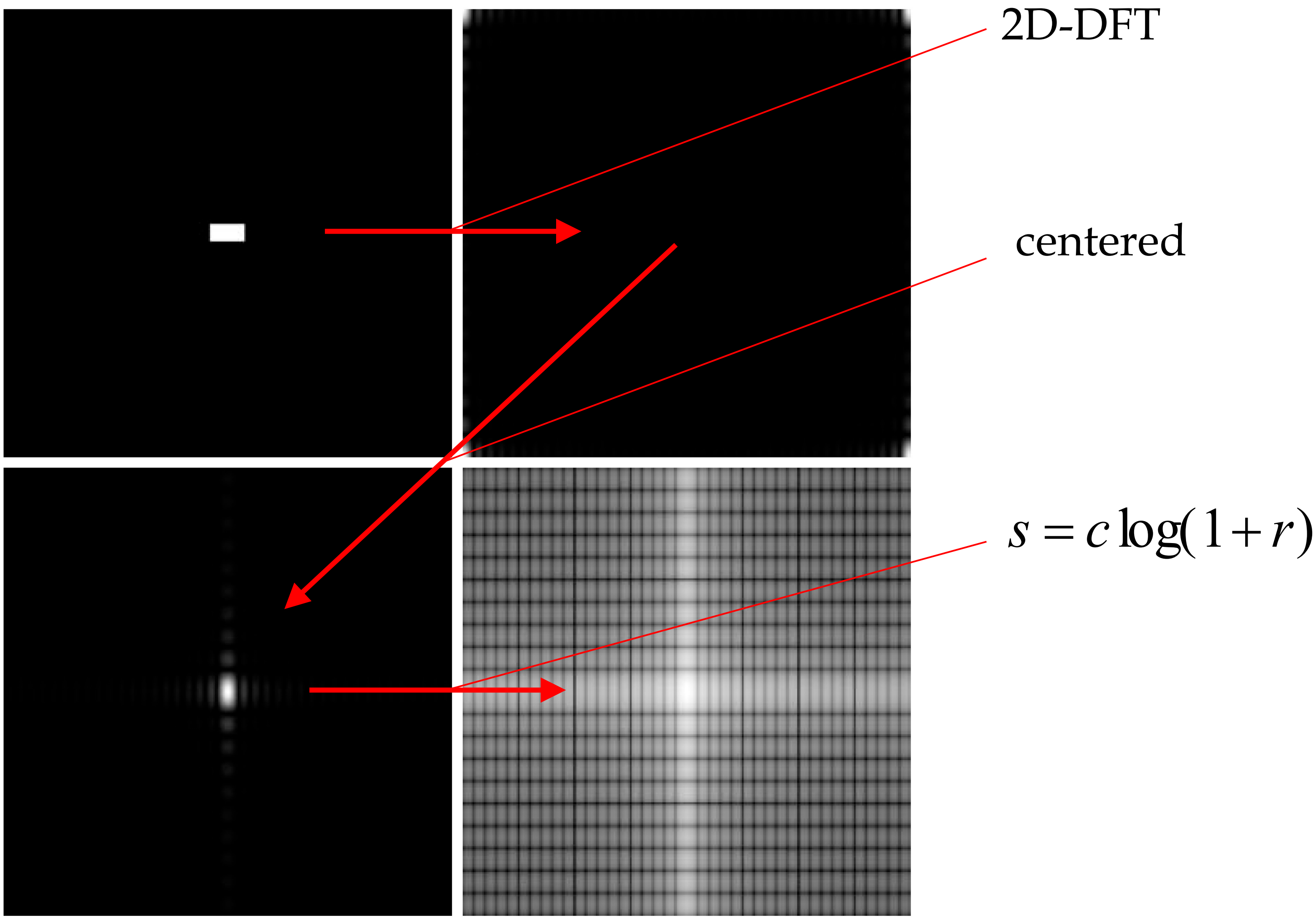
Example:



From [Gonzalez & Woods]

Multiply input image by $(-1)^{x+y}$ we get the same effect.

Log-Magnitude Visualization



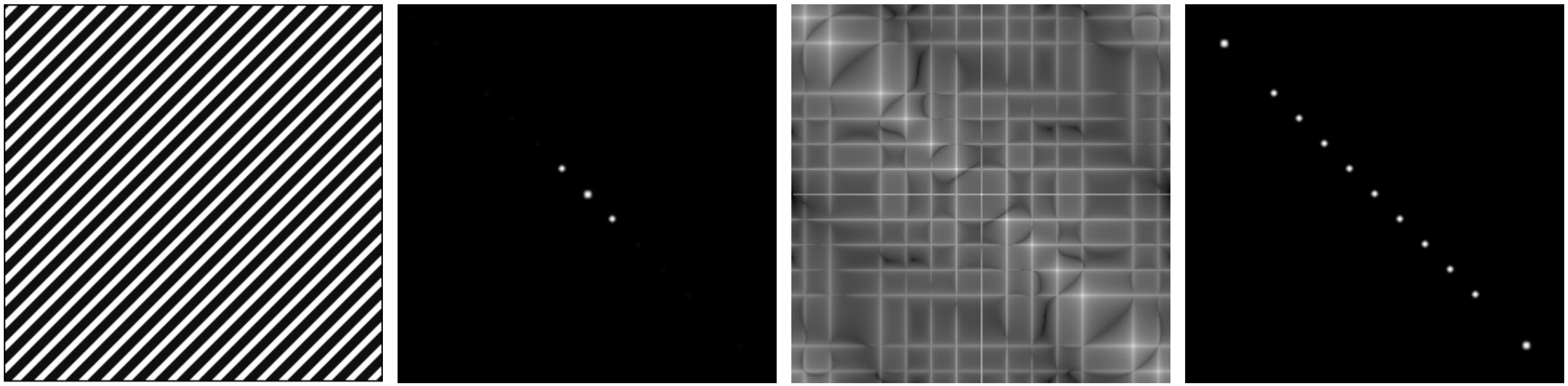
From [Gonzalez & Woods]

Fourier Transform: Properties

Property	Signal		Transform
superposition	$f_1(x) + f_2(x)$		$F_1(\omega) + F_2(\omega)$
shift	$f(x - x_0)$		$F(\omega)e^{-j\omega x_0}$
reversal	$f(-x)$		$F^*(\omega)$
convolution	$f(x) * h(x)$		$F(\omega)H(\omega)$
correlation	$f(x) \otimes h(x)$		$F(\omega)H^*(\omega)$
multiplication	$f(x)h(x)$		$F(\omega) * H(\omega)$
differentiation	$f'(x)$		$j\omega F(\omega)$
domain scaling	$f(ax)$		$1/aF(\omega/a)$
real images	$f(x) = f^*(x)$	$\Leftrightarrow$	$F(\omega) = F(-\omega)$
Parseval's Theorem	$\sum_x [f(x)]^2$	$=$	$\sum_\omega [F(\omega)]^2$

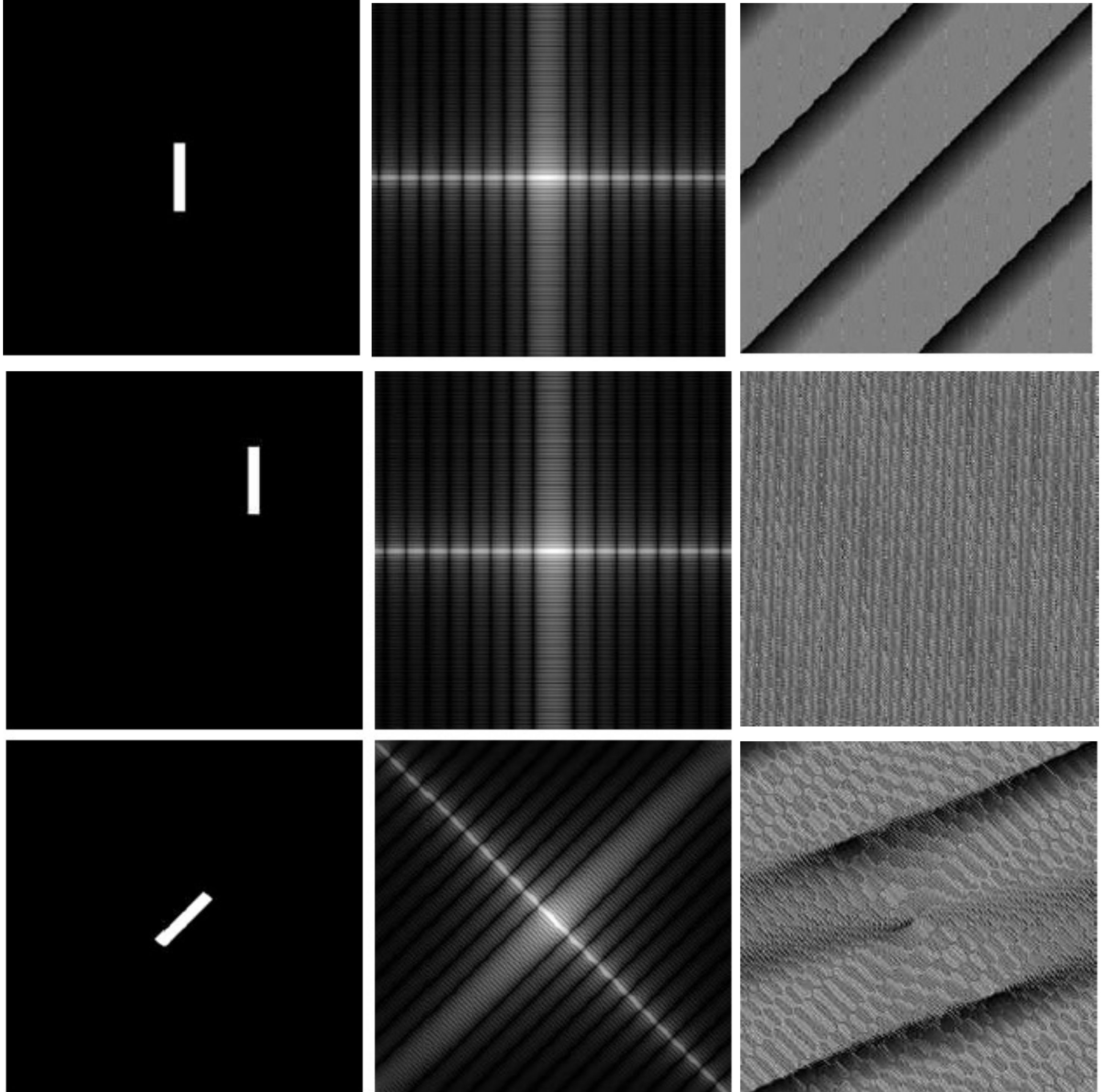
Fourier Transform

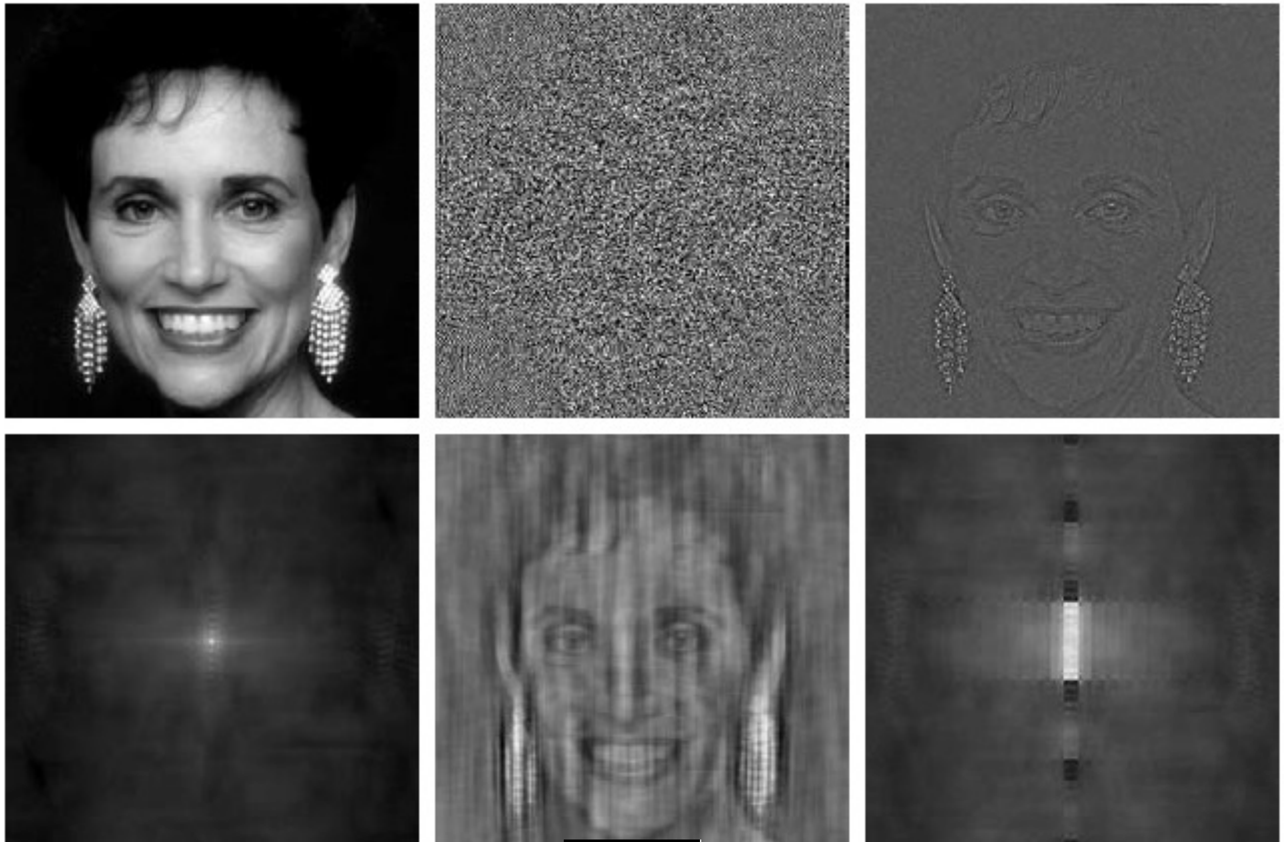
– Example : Simple image



(a) Diagonal stripes (b) Fourier transform (c) Logarithm scaling (d) Thresholding

- (c) Minor frequencies appear by approximating the diagonal as the square pixels of the image.
- (d) To find the most important frequencies, threshold all the frequencies whose magnitude is at least 5% of the main peak.
- The represented frequencies are all multiples of the basic frequency of the stripes in the spatial domain image.





a b c
d e f

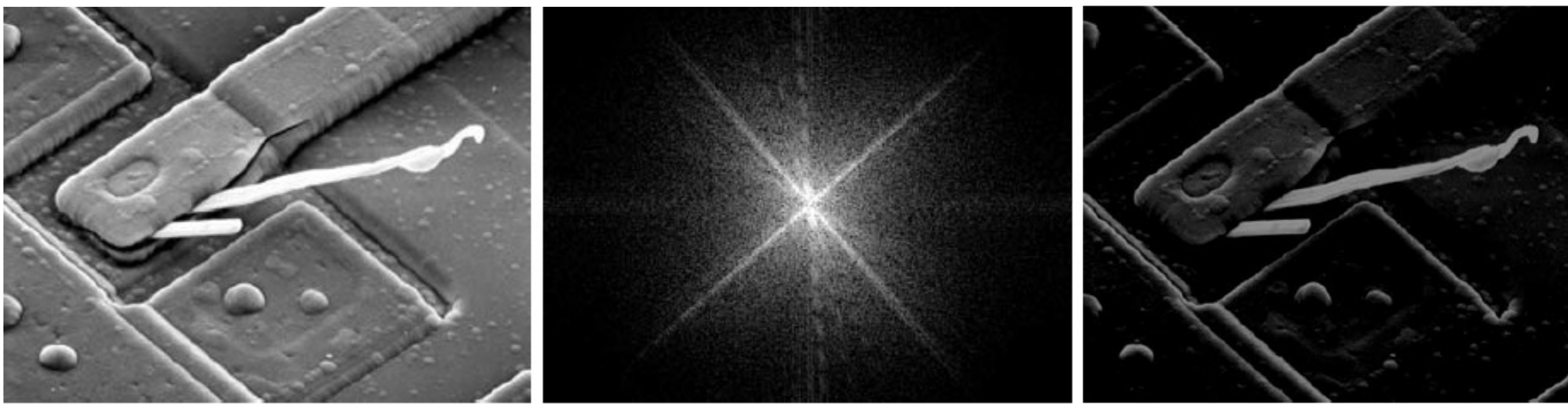
FIGURE 4.27 (a) Woman. (b) Phase angle. (c) Woman reconstructed using only the phase angle. (d) Woman reconstructed using only the spectrum. (e) Reconstruction using the phase angle corresponding to the woman and the spectrum corresponding to the rectangle in Fig. 4.24(a). (f) Reconstruction using the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

Remember: Fourier Transform: Properties

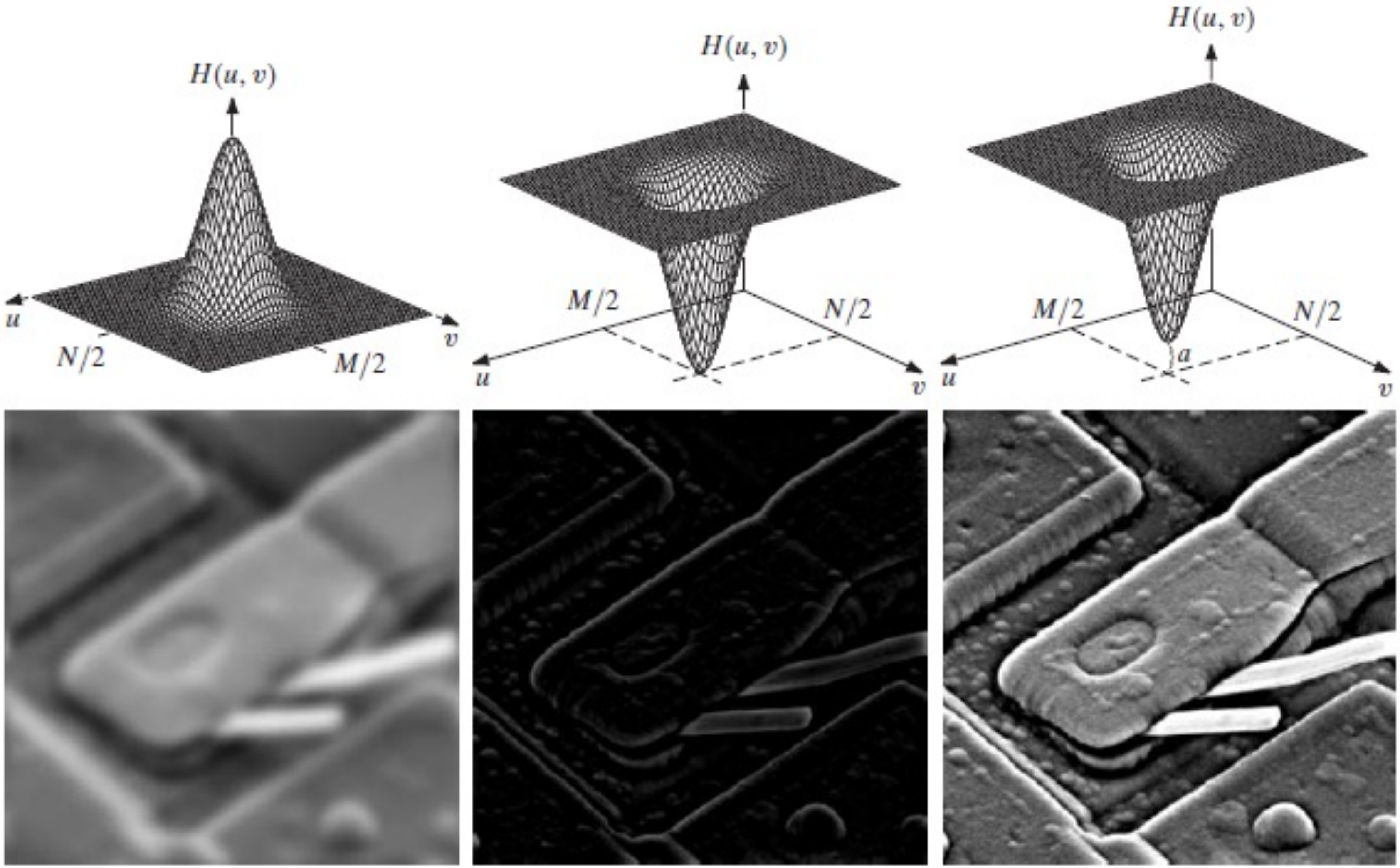
convolution $f(x) * h(x)$ $F(\omega)H(\omega)$

Frequency Domain Filtering Fundamentals

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



(a) SEM image of a damaged integrated circuit. (b) Fourier spectrum of (a).
(c) Result of filtering the image in (a) by setting to 0 the term (*M* 2, *N* 2) in the Fourier transform.
(Original image courtesy of Dr. J. M. Hudak, Brockhouse Institute for Materials Research, McMaster University, Hamilton, Ontario, Canada.)

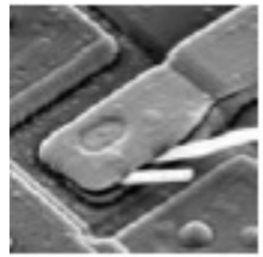


a b c
d e f

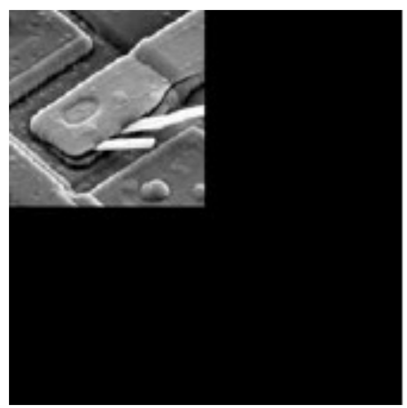
FIGURE 4.31 Top row: frequency domain filters. Bottom row: corresponding filtered images obtained using Eq. (4.7-1). We used $a = 0.85$ in (c) to obtain (f) (the height of the filter itself is 1). Compare (f) with Fig. 4.29(a).

Summary of Steps for Filtering in the Frequency Domain

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

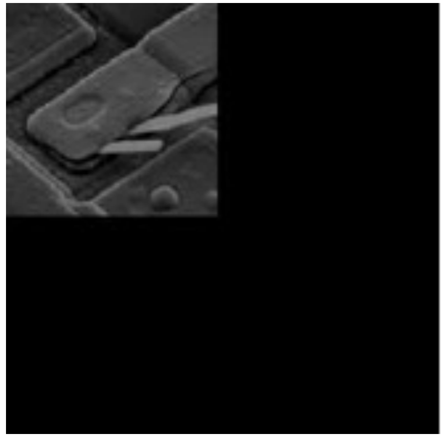


Form a padded image $f_p(x,y)$, of size PxQ by appending the necessary number of zeros to f(x, y).



.

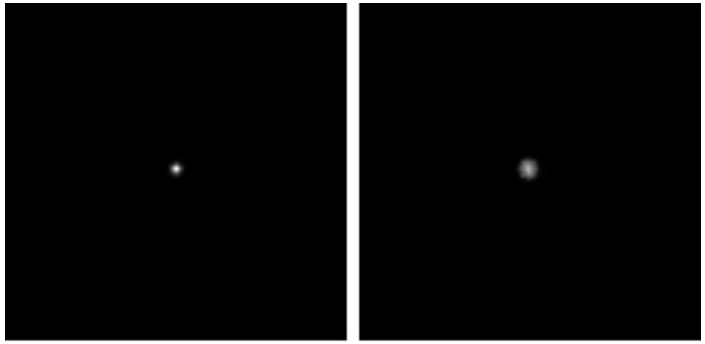
Multiply $f_p(x,y)$ by $(-1)^{x+y}$ to center its transform



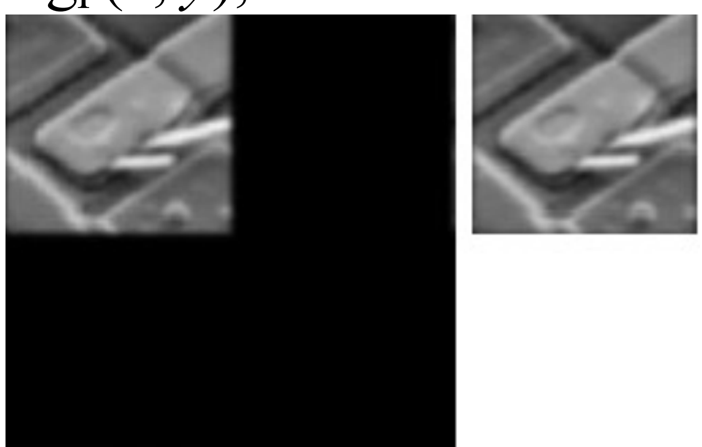
As noted earlier, centering helps in visualizing the filtering process and in generating the filter functions themselves, but centering is not a fundamental requirement.

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

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



Obtain the processed image $g_p(x, y) = \left\{\text{real}\left[\Im^{-1}[G(u, v)]\right]\right\}(-1)^{x+y}$ where the real part is selected in order to ignore parasitic complex components resulting from computational inaccuracies, and the subscript p indicates that we are dealing with padded arrays. Obtain the final processed result, $g(x, y)$, by extracting the region from the top, left quadrant of $g_p(x, y)$,



a b

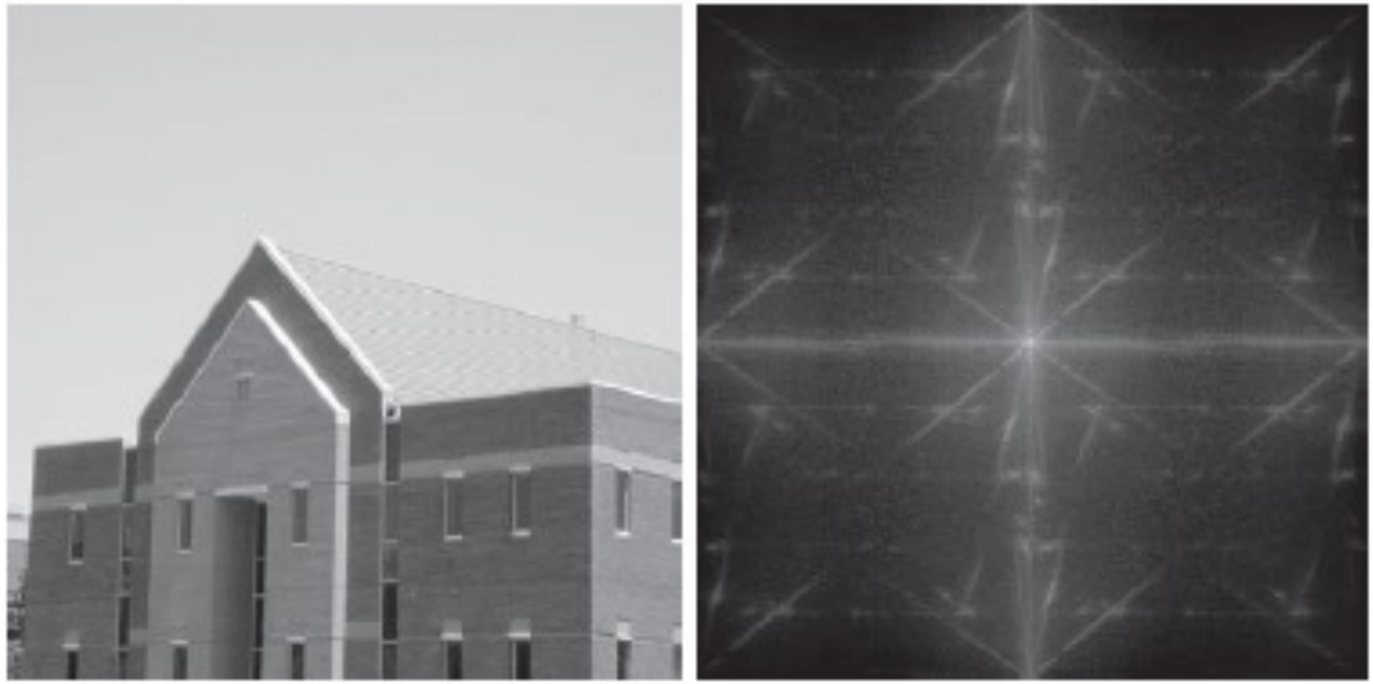
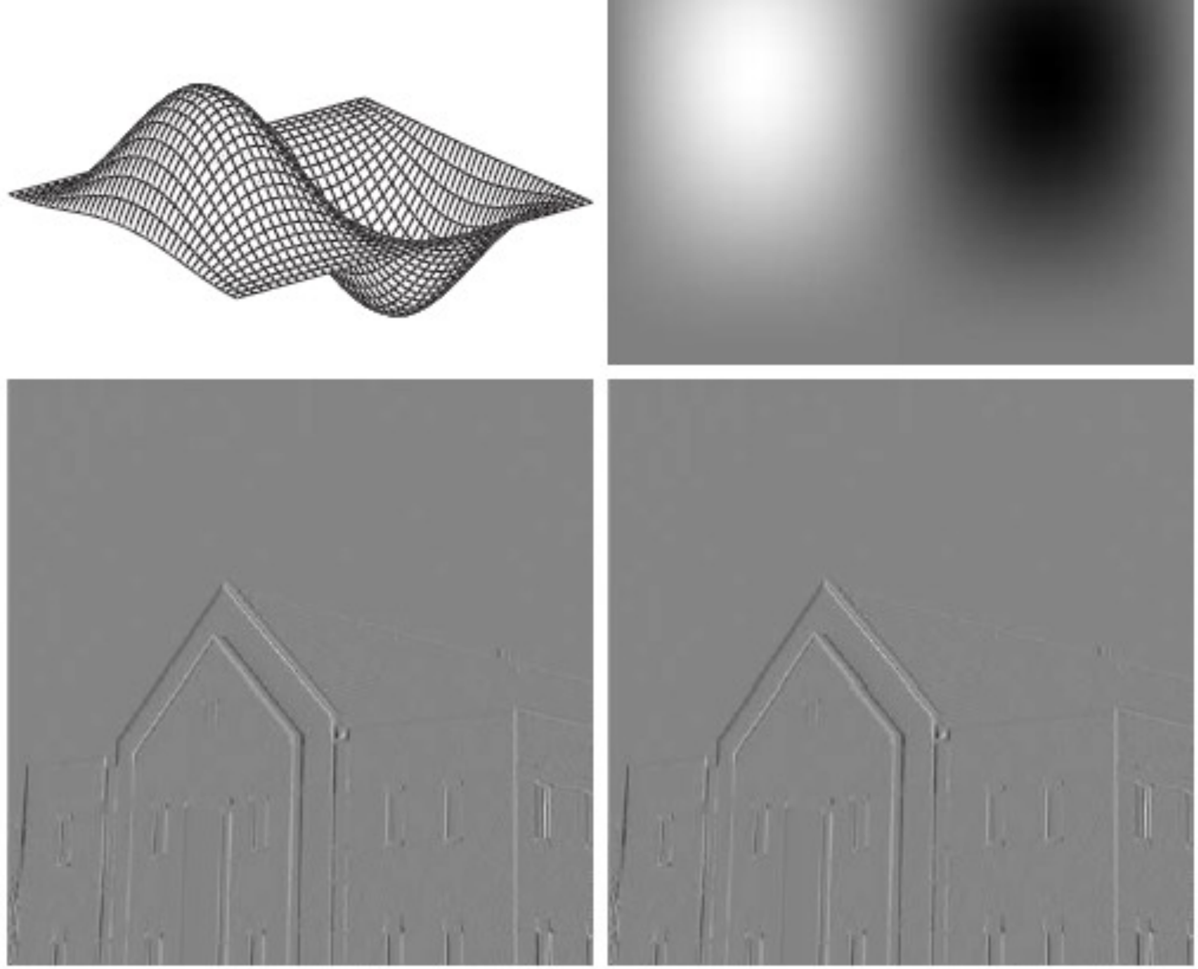


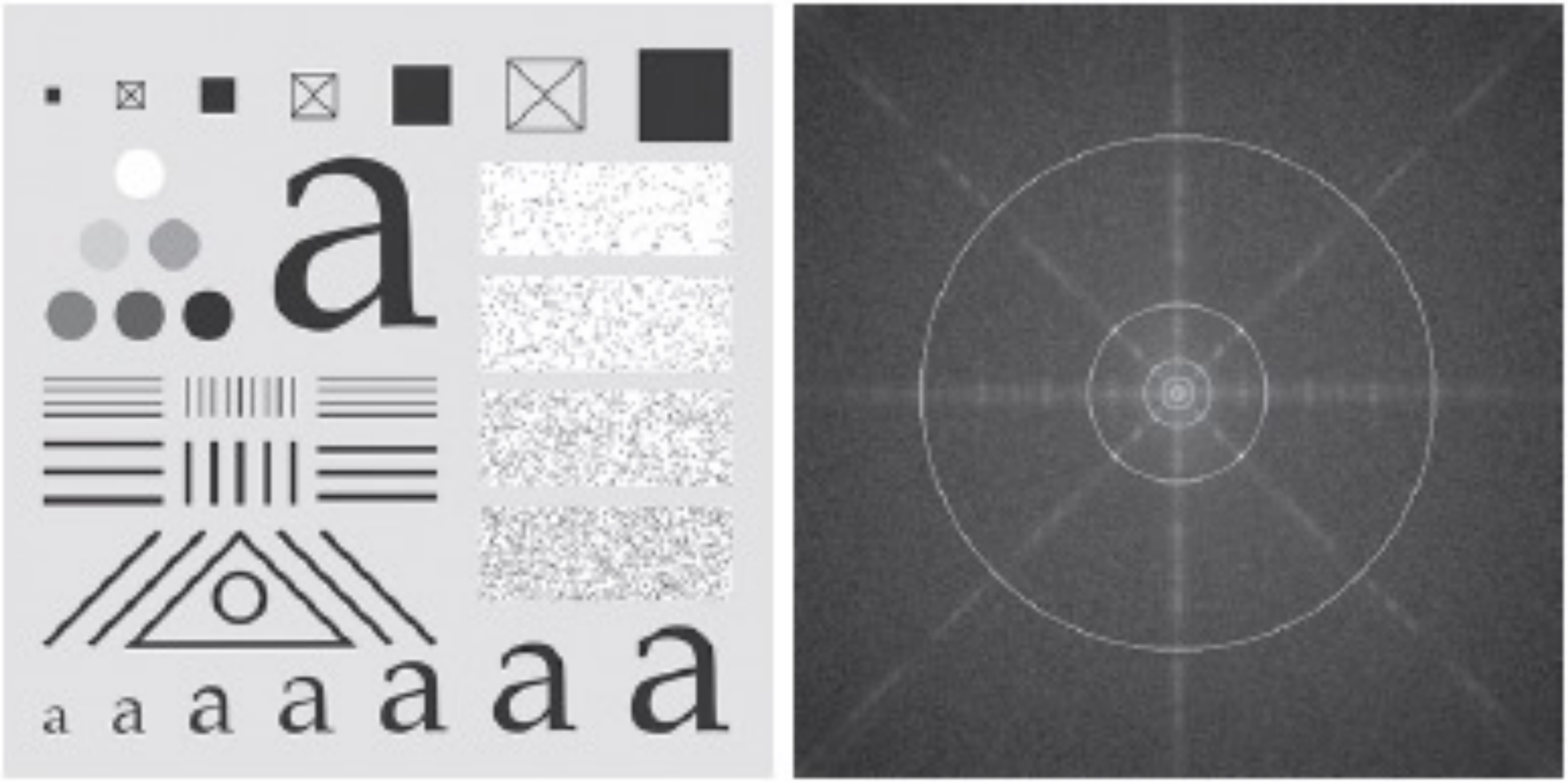
FIGURE 4.37 (a) Image of a building, and (b) its Fourier spectrum.

−1	0	1
−2	0	2
−1	0	1



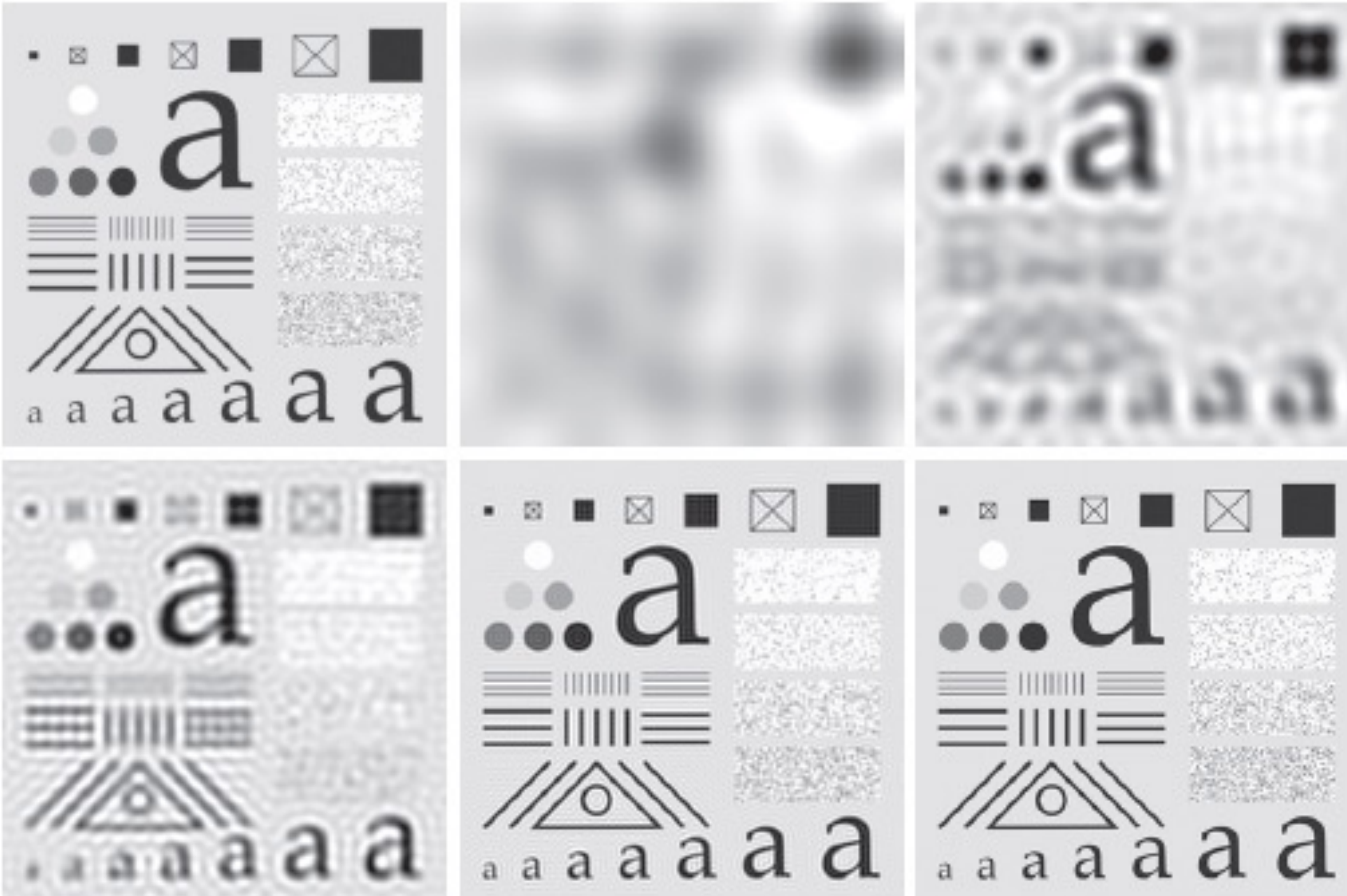
a b
c d

FIGURE 4.39 (a) A spatial mask and perspective plot of its corresponding frequency domain filter. (b) Filter shown as an image. (c) Result of filtering Fig. 4.38(a) in the frequency domain with the filter in (b). (d) Result of filtering the same image with the spatial filter in (a). The results are identical.



a b

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



a b c
d e f

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

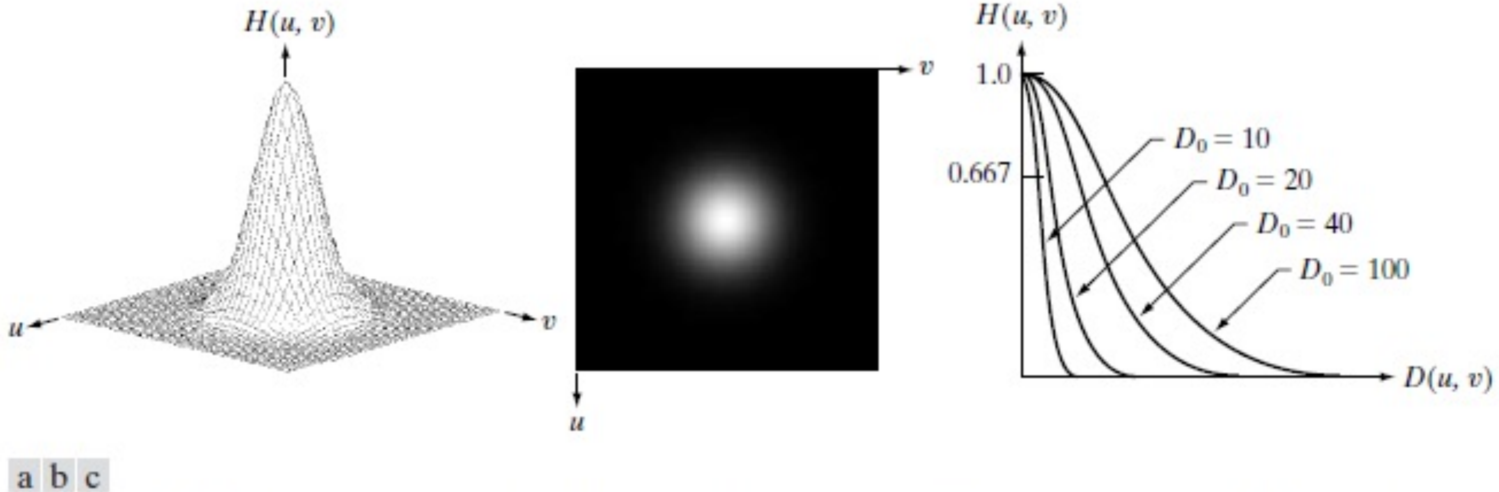
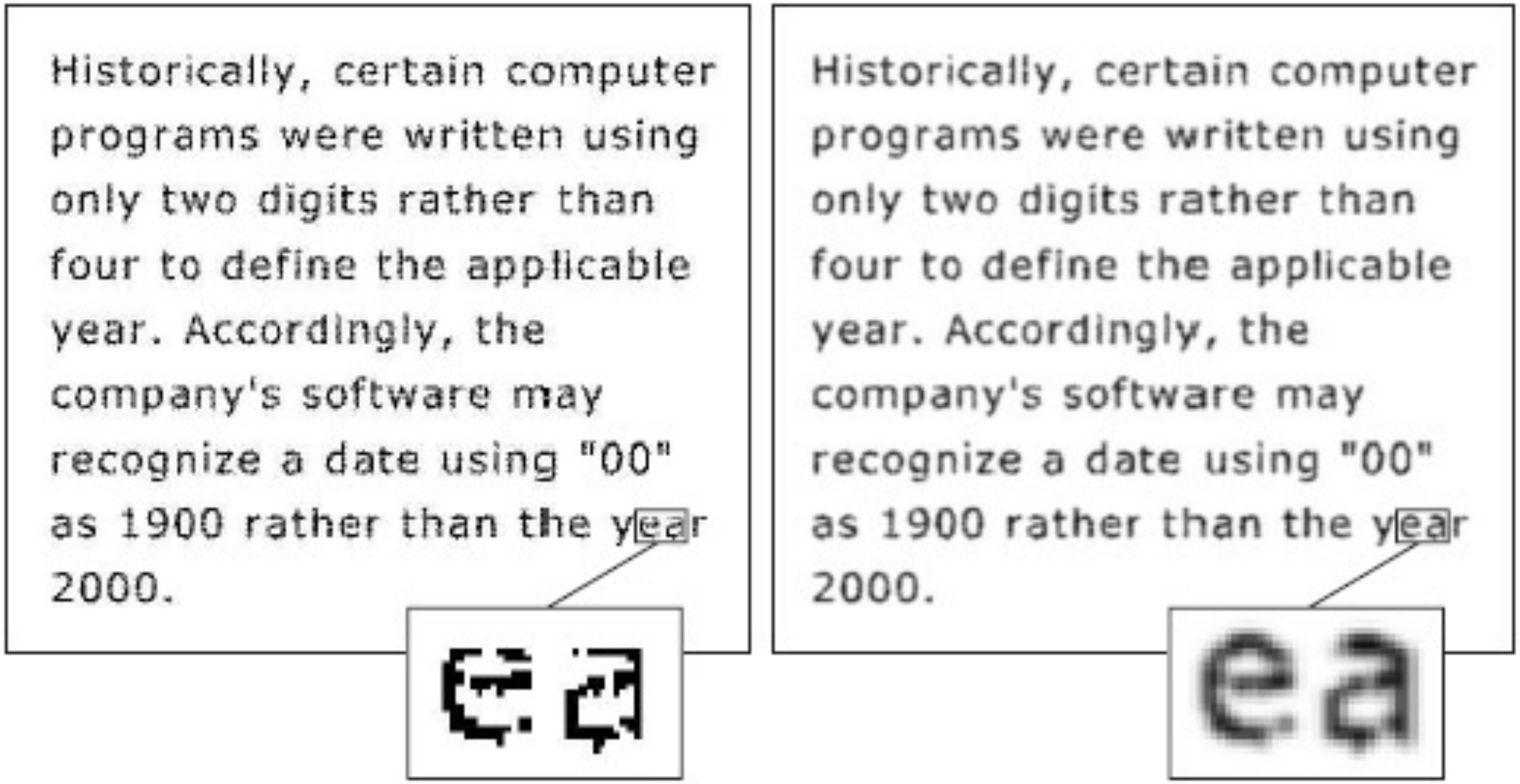


FIGURE 4.47 (a) Perspective plot of a GLPF transfer function. (b) Filter displayed as an image. (c) Filter radial cross sections for various values of D_0.

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

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



a b

FIGURE 4.49
(a) Sample text of low resolution (note broken characters in magnified view). (b) Result of filtering with a GLPF (broken character segments were joined).

a b c

FIGURE 4.50 (a) Original image (784 × 732 pixels). (b) Result of filtering using a GLPF with $D_0 = 100$. (c) Result of filtering using a GLPF with $D_0 = 80$. Note the reduction in fine skin lines in the magnified sections in (b) and (c).

A 2-D *ideal highpass filter* (IHPF) is defined as

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

a b c

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

NEXT COURSE

Some restoration techniques are best formulated in the spatial domain, while others are better suited for the frequency domain. Next cources will be introduces a linear model of the image degradation/restoration And also deals with various noise models encountered frequently in practice. We develop several spatial filtering techniques for reducing the noise content of an image, a process often referred to as *image denoising and restoration*.