DIGITAL IMAGE PROCESSING

Unit-2: Image Enhancement in Spatial Domain

Dr. Piyush Joshi

Indian Institute of Information Technology
Sri City, Andhra Pradesh

Spatial Domain Vs. Transform Domain

- **Spatial domain:** image plane itself, directly process the intensity values of the image plane
- **Transform domain:** process the transform coefficients, not directly process the intensity values of the image plane

Spatial Domain Process

$$g(x, y) = T[f(x, y)]$$

$f(x, y)$: input image

$g(x, y)$: output image

T: an operator of f defined over a neighborhood of point (x, y).

Spatial Domain Process

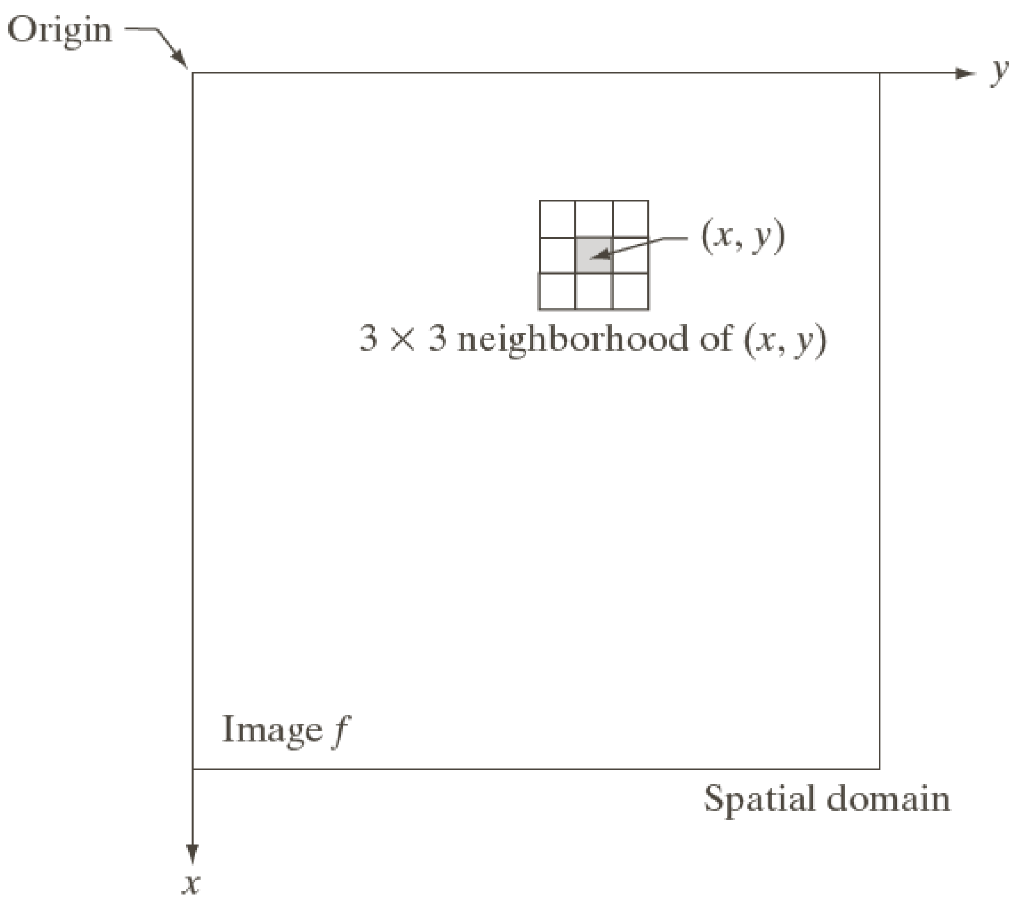
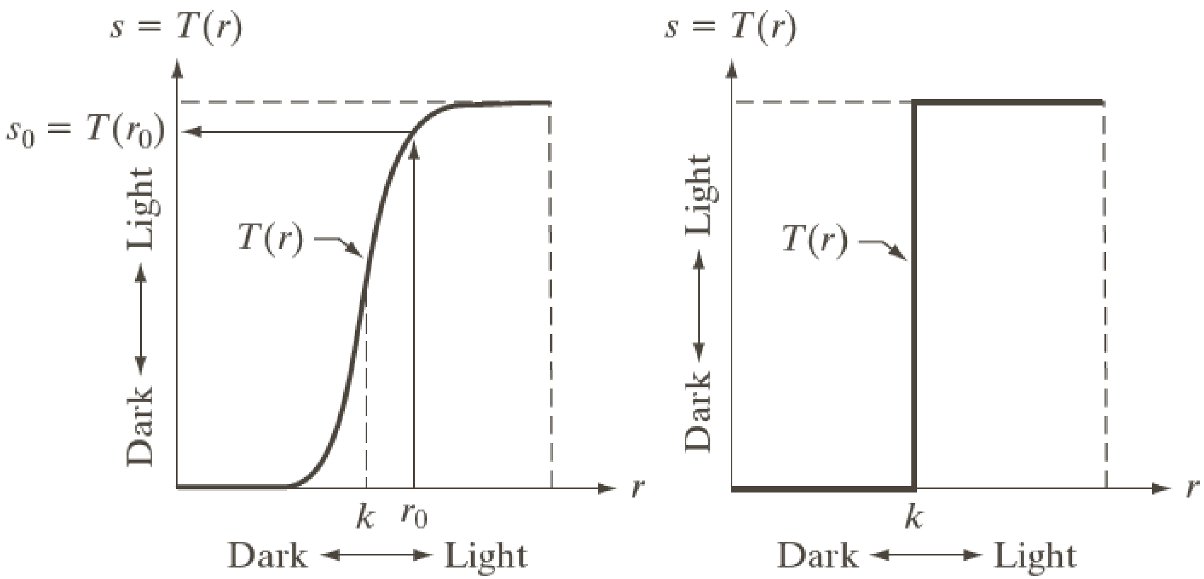


FIGURE 3.1 A 3 × 3 neighborhood about a point (x, y) in an image in the spatial domain. The neighborhood is moved from pixel to pixel in the image to generate an output image.

Spatial Domain Process

Intensity Transformation Function: $s = T(r)$



a b

FIGURE 3.2 Intensity transformation functions. (a) Contrast-stretching function. (b) Thresholding function.

Basic Intensity Transformation Functions

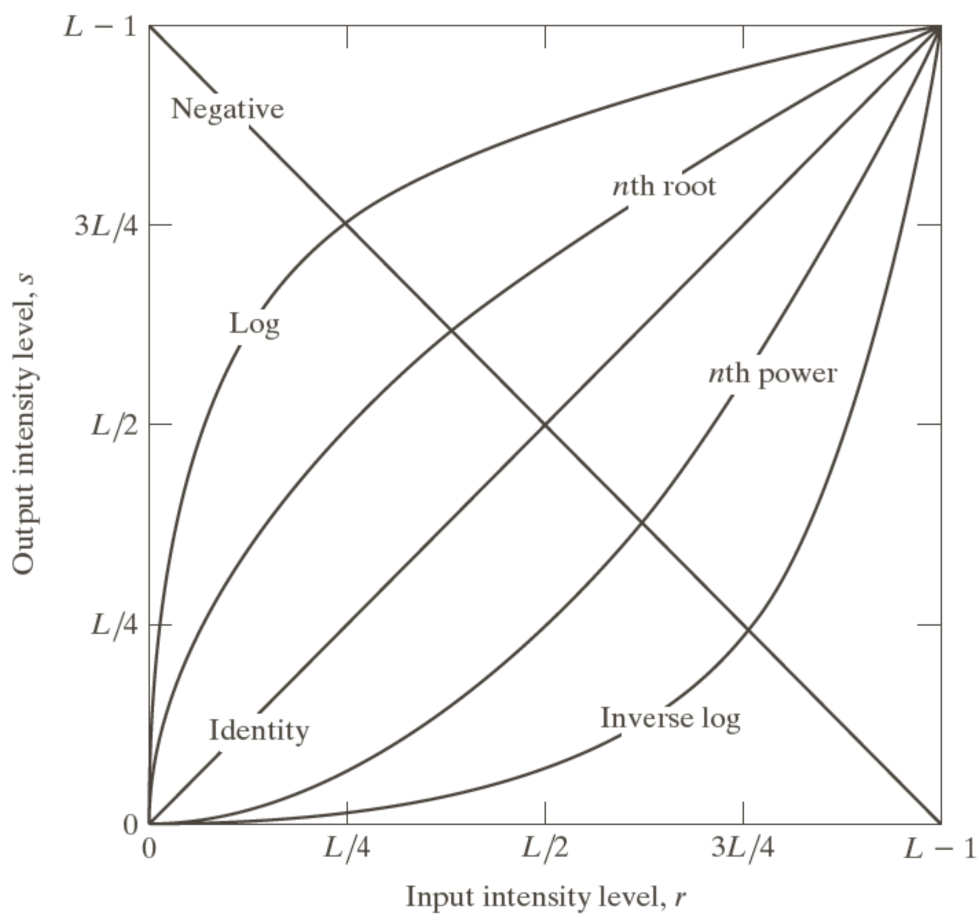


FIGURE 3.3 Some basic intensity transformation functions. All curves were scaled to fit in the range shown.

Basic Intensity Transformation Functions

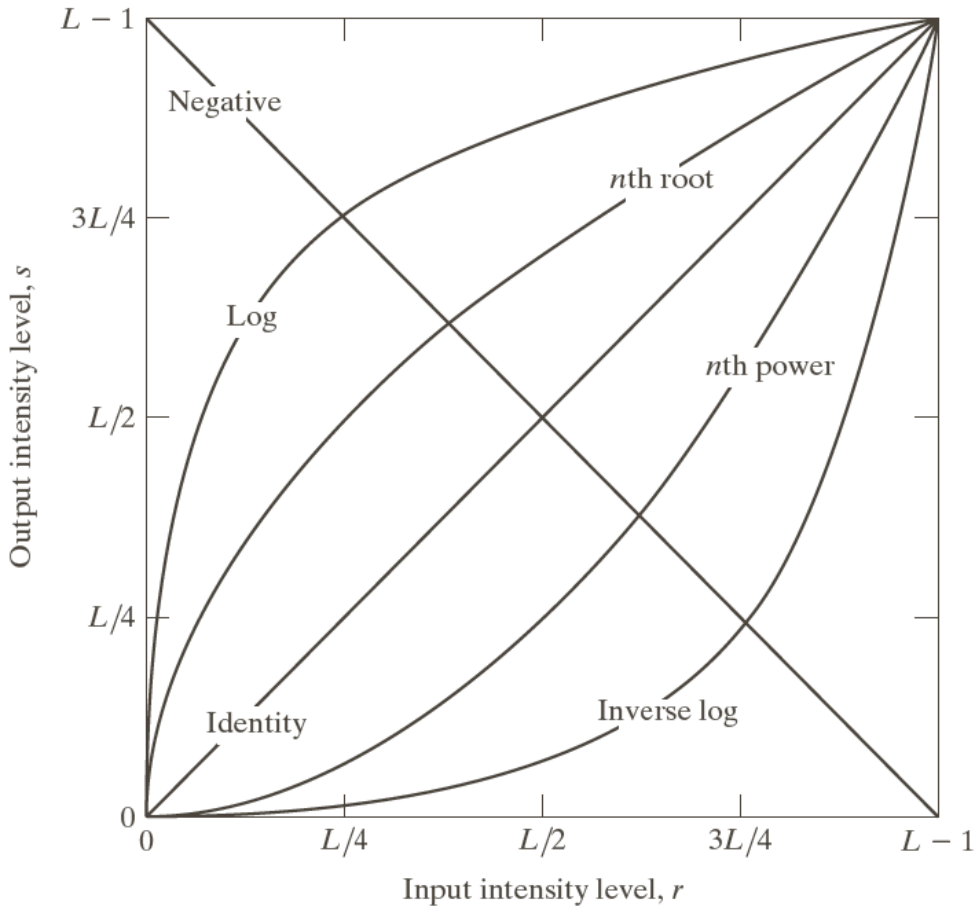


Image negatives

$$s = L - 1 - r$$

Basic Intensity Transformation Functions

Example: Image Negative

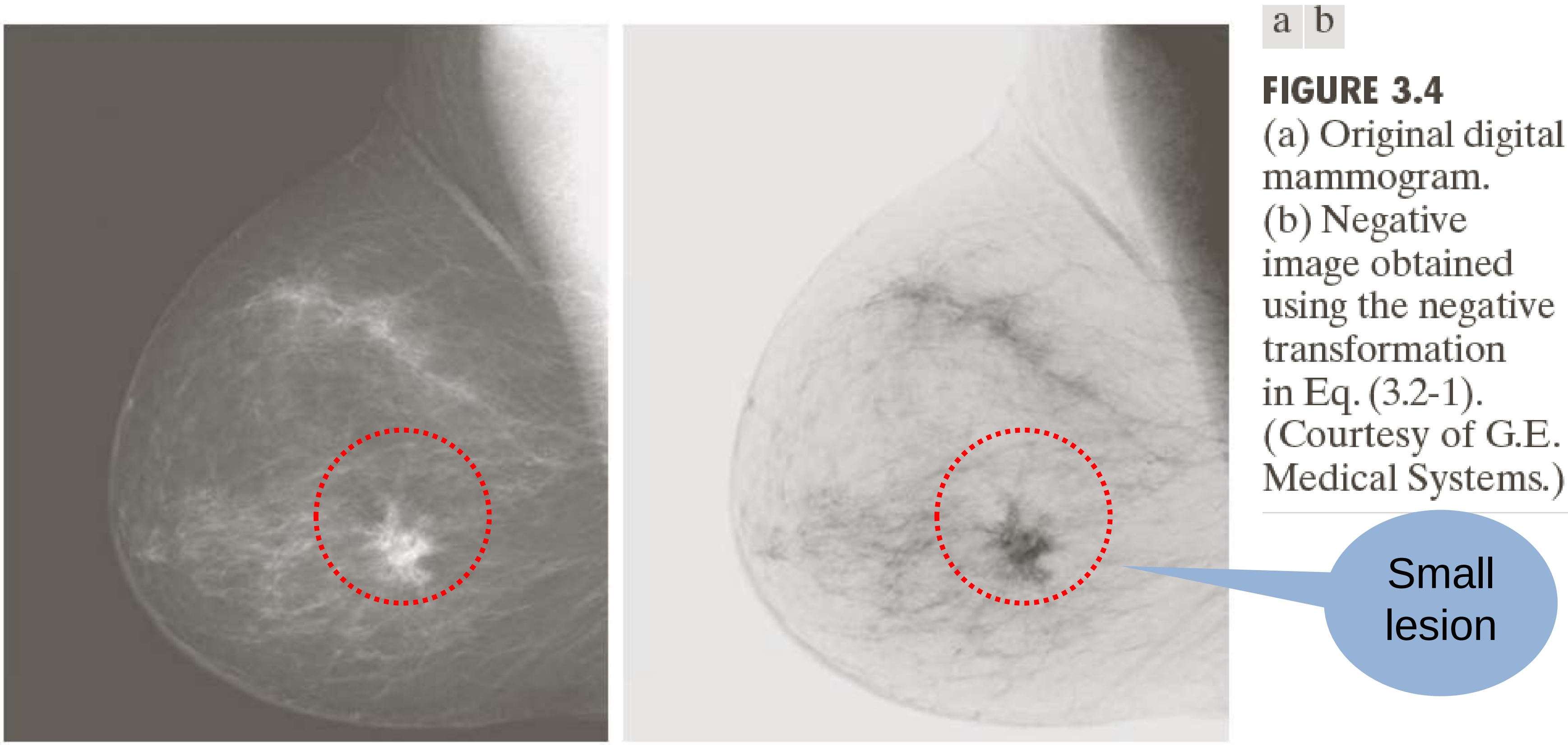
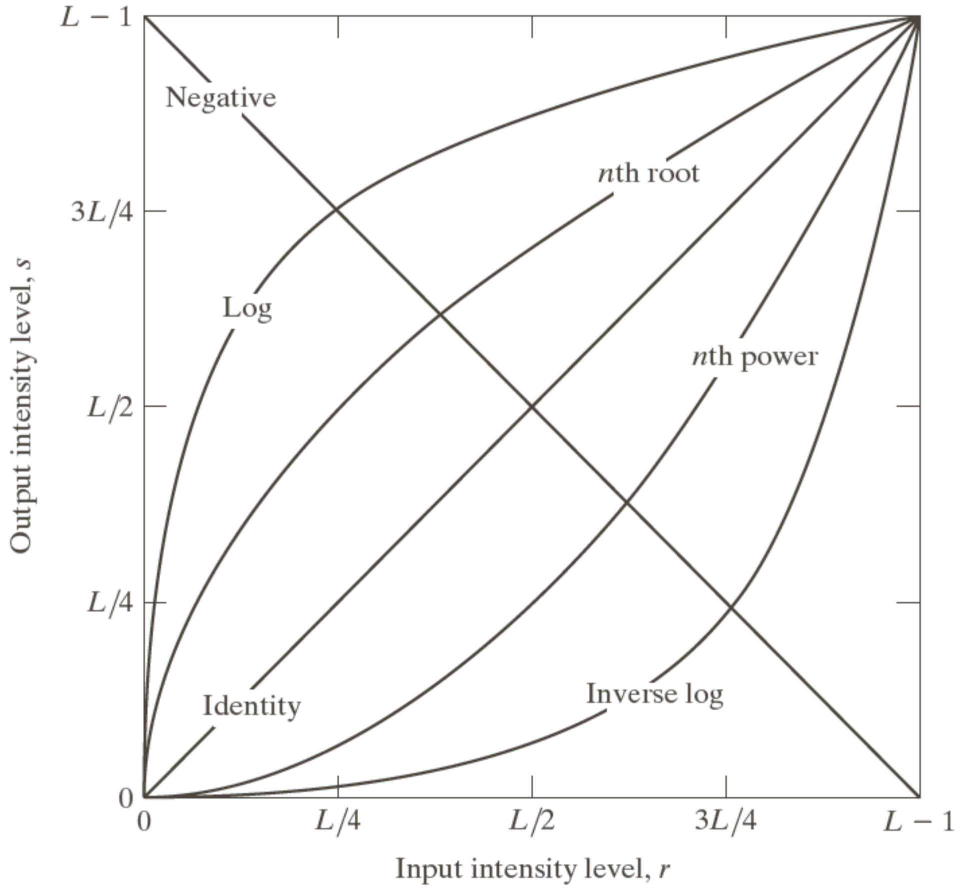


FIGURE 3.4 (a) Original digital mammogram. (b) Negative image obtained using the negative transformation in Eq. (3.2-1). (Courtesy of G.E. Medical Systems.)

Basic Intensity Transformation Functions

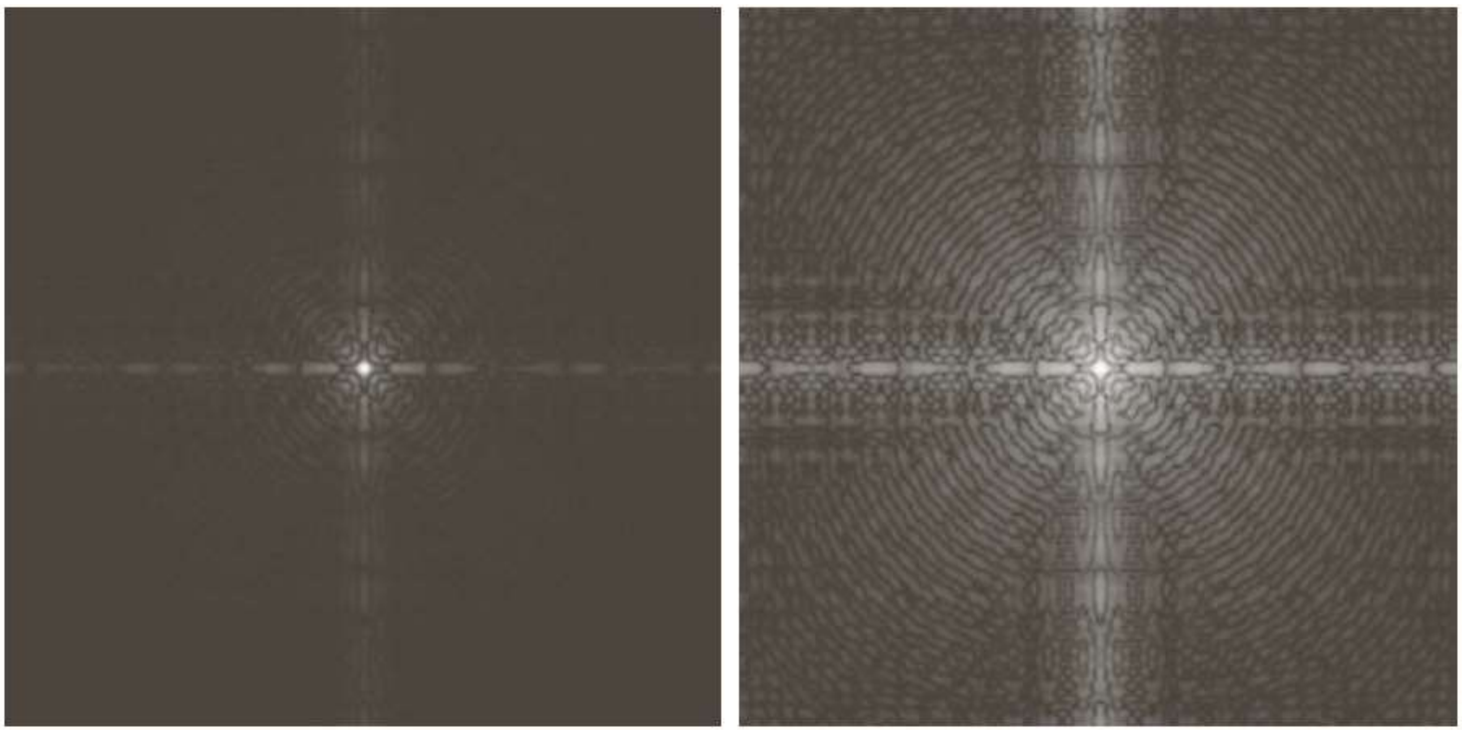


Log Transformations

$$s = c\log(1 + r)$$

Basic Intensity Transformation Functions

Example: Log Transformations



a b

FIGURE 3.5 (a) Fourier spectrum. (b) Result of applying the log transformation in Eq. (3.2-2) with $c = 1$.

Basic Intensity Transformation Functions

Power-Law (Gamma) Transformations

$$s = cr^{\gamma}$$

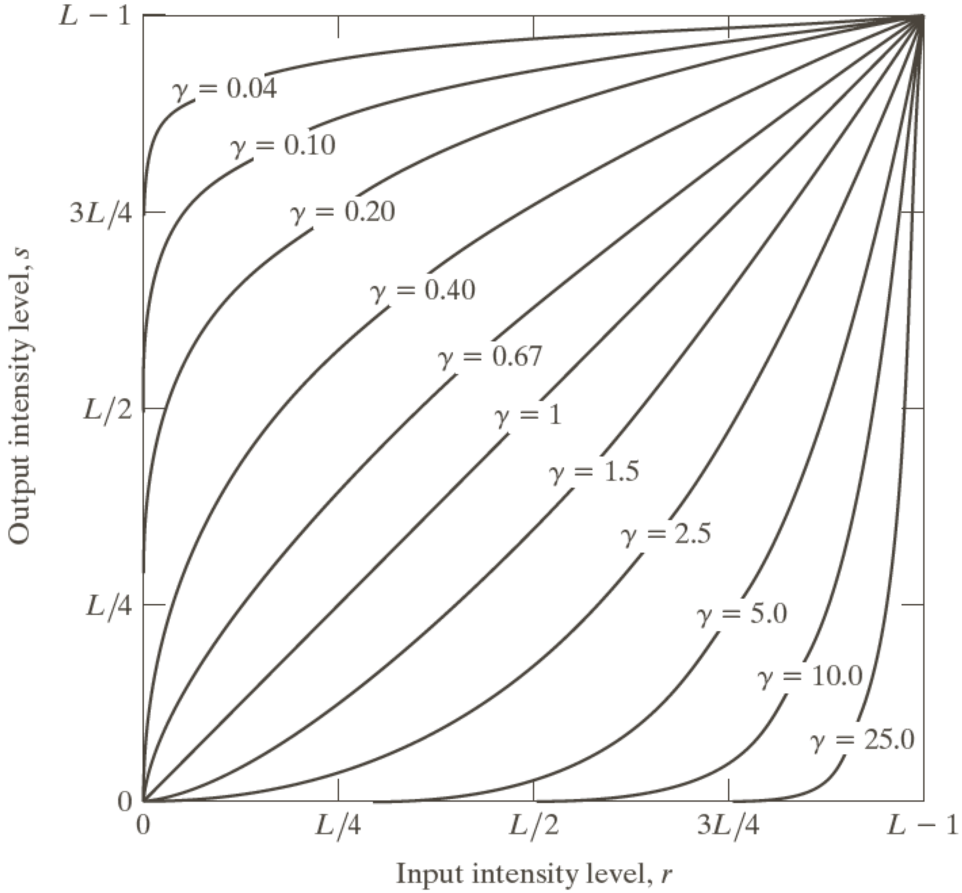


FIGURE 3.6 Plots of the equation $s = cr^{\gamma}$ for various values of γ ($c = 1$ in all cases). All curves were scaled to fit in the range shown.

Basic Intensity Transformation Functions

Example: Gamma Transformations

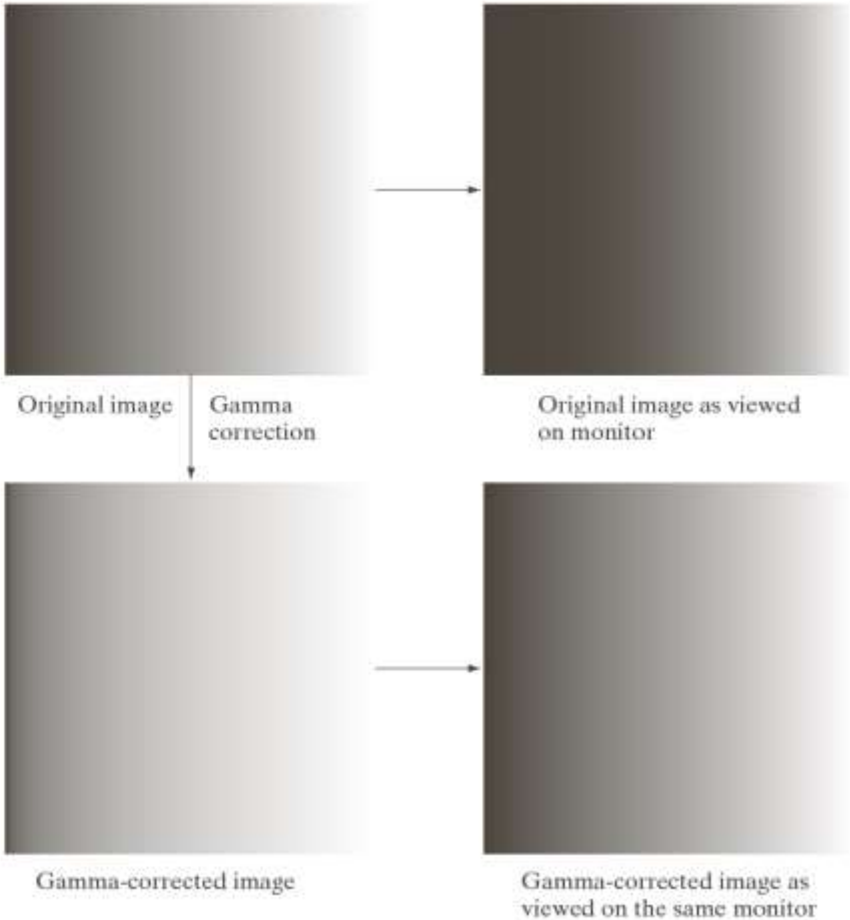
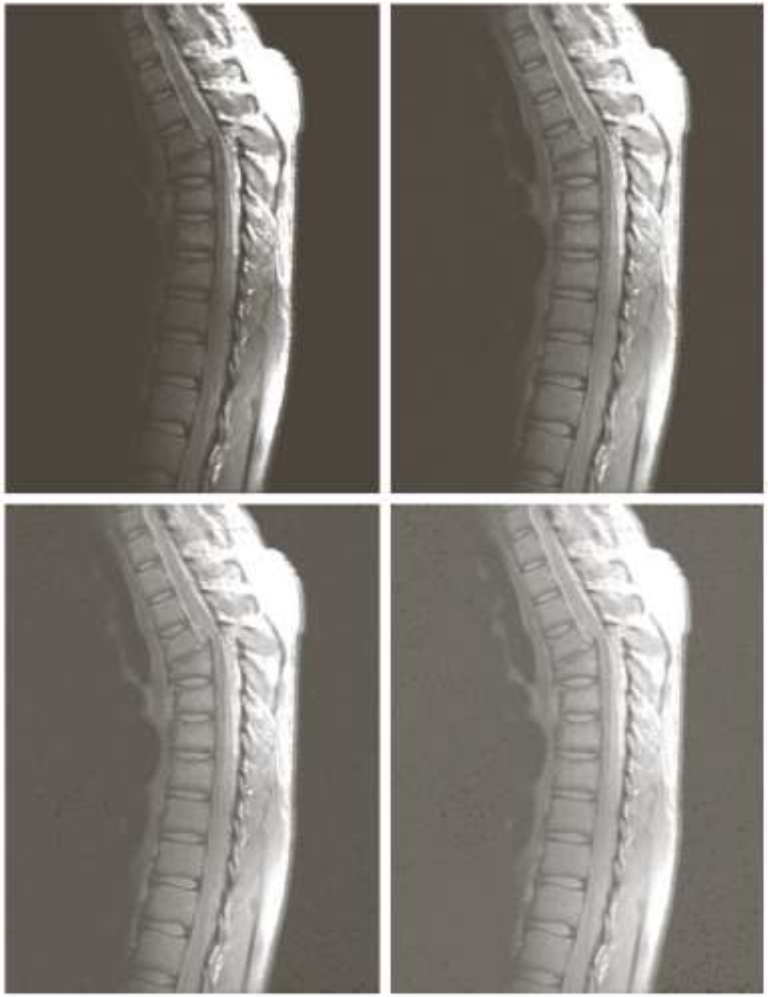


FIGURE 3.7
(a) Intensity ramp image. (b) Image as viewed on a simulated monitor with a gamma of 2.5. (c) Gamma-corrected image. (d) Corrected image as viewed on the same monitor. Compare (d) and (a).

Basic Intensity Transformation Functions

More Example: Gamma Transformations



a	b
c	d

FIGURE 3.8 (a) Magnetic resonance image (MRI) of a fractured human spine. (b)–(d) Results of applying the transformation in Eq. (3.2-3) with $c = 1$ and $\gamma = 0.6, 0.4,$ and 0.3, respectively. (Original image courtesy of Dr. David R. Pickens, Department of Radiology and Radiological Sciences, Vanderbilt University Medical Center.)

Basic Intensity Transformation Functions

More Example: Gamma Transformations

a	b
c	d

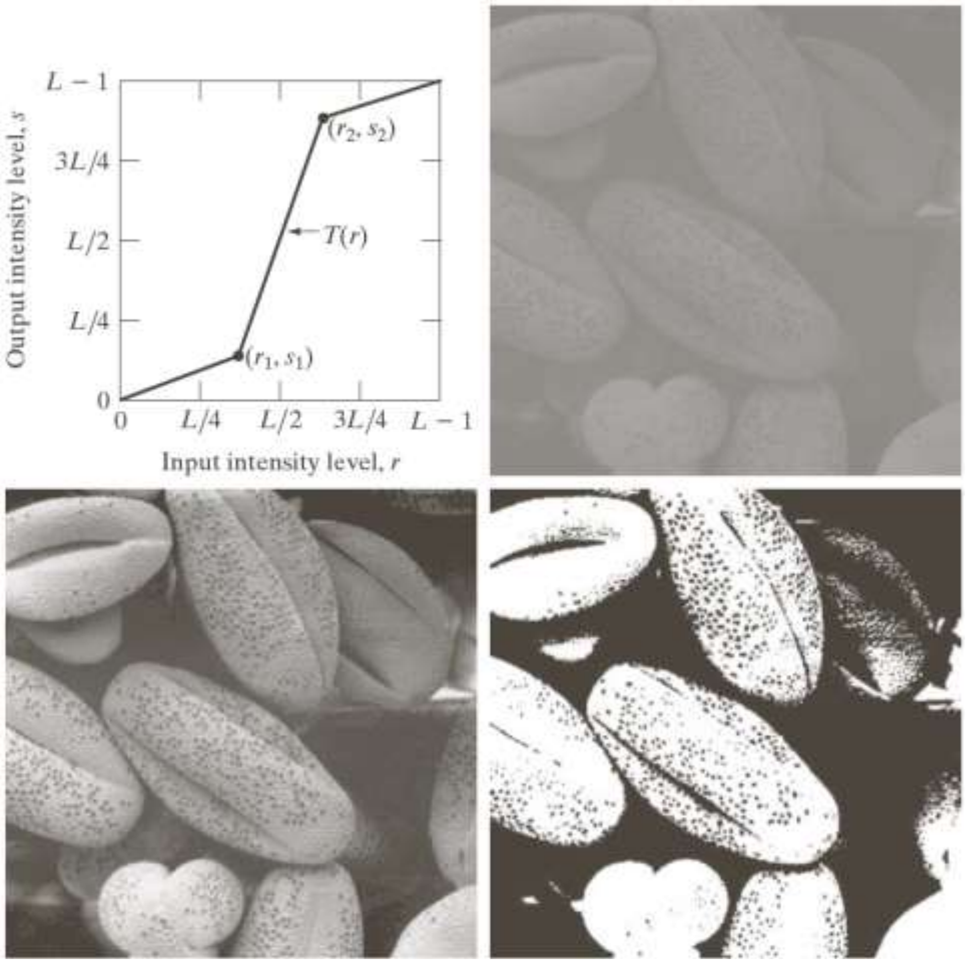
FIGURE 3.9 (a) Aerial image. (b)–(d) Results of applying the transformation in Eq. (3.2-3) with $c = 1$ and $\gamma = 3.0, 4.0,$ and 5.0, respectively. (Original image for this example courtesy of NASA.)

Piecewise-Linear Transformations

- **Contrast Stretching**
  Expands the range of intensity levels in an image so that it spans the full intensity range of the recording medium or display device.

- **Intensity-level Slicing**
  Highlighting a specific range of intensities in an image often is of interest.

Piecewise-Linear Transformations

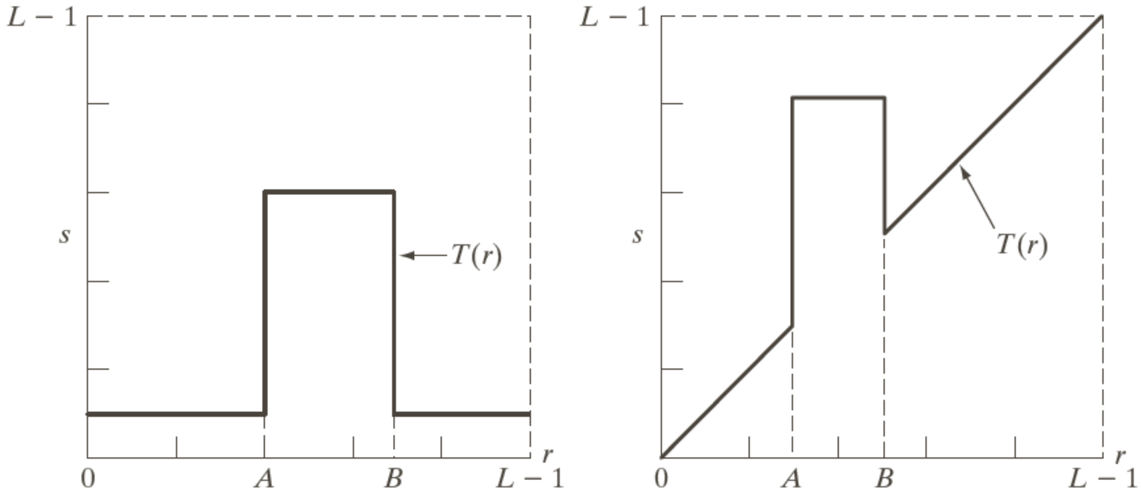


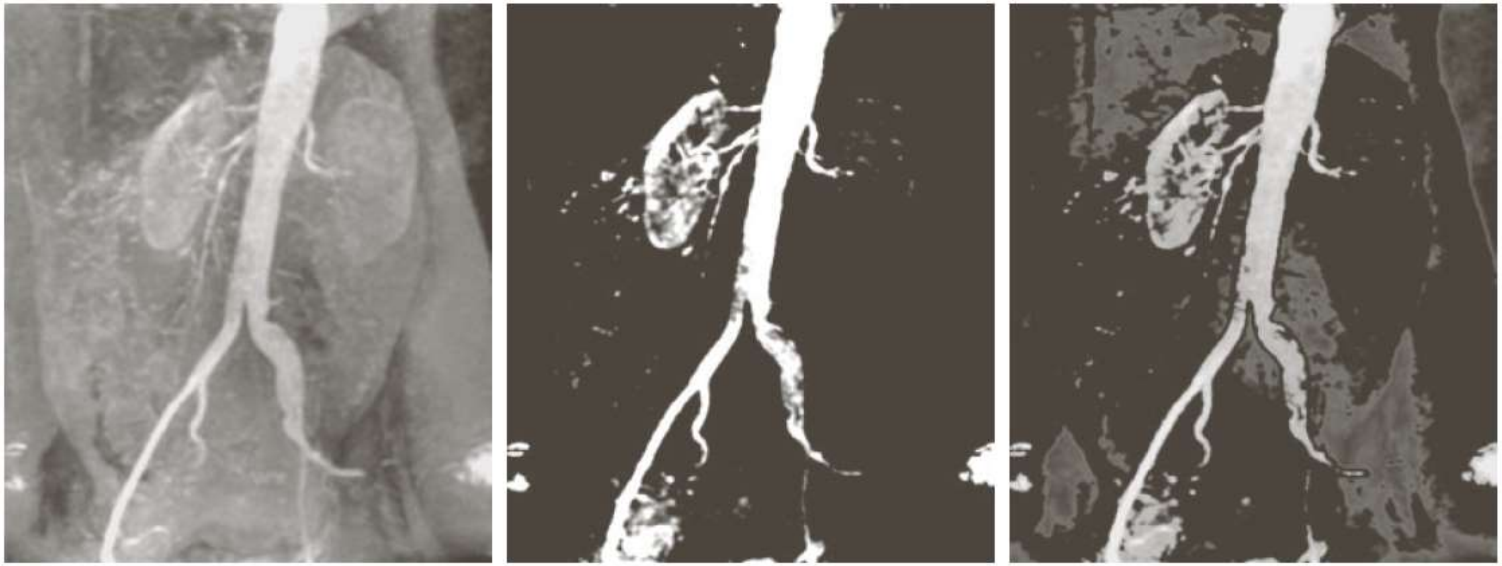
a b
c d

FIGURE 3.10 Contrast stretching. (a) Form of transformation function. (b) A low-contrast image. (c) Result of contrast stretching. (d) Result of thresholding. (Original image courtesy of Dr. Roger Heady, Research School of Biological Sciences, Australian National University, Canberra, Australia.)

a b

FIGURE 3.11 (a) This transformation highlights intensity range [A, B] and reduces all other intensities to a lower level. (b) This transformation highlights range [A, B] and preserves all other intensity levels.





a b c

FIGURE 3.12 (a) Aortic angiogram. (b) Result of using a slicing transformation of the type illustrated in Fig. 3.11(a), with the range of intensities of interest selected in the upper end of the gray scale. (c) Result of using the transformation in Fig. 3.11(b), with the selected area set to black, so that grays in the area of the blood vessels and kidneys were preserved. (Original image courtesy of Dr. Thomas R. Gest, University of Michigan Medical School.)

Image Enhancement in Spatial Domain

Histogram Processing

- Histogram Equalization
- Histogram Matching
- Local Histogram Processing
- Using Histogram Statistics for Image Enhancement

Image Enhancement in Spatial Domain

Histogram Processing

Histogram $h(r_k) = n_k$

r_k is the k^{th} intensity value

n_k is the number of pixels in the image with intensity r_k

Normalized histogram $p(r_k) = \dfrac{n_k}{MN}$

n_k: the number of pixels in the image of size $\mathrm{M} \times \mathrm{N}$ with intensity r_k

Image Enhancement in Spatial Domain

Histogram Processing

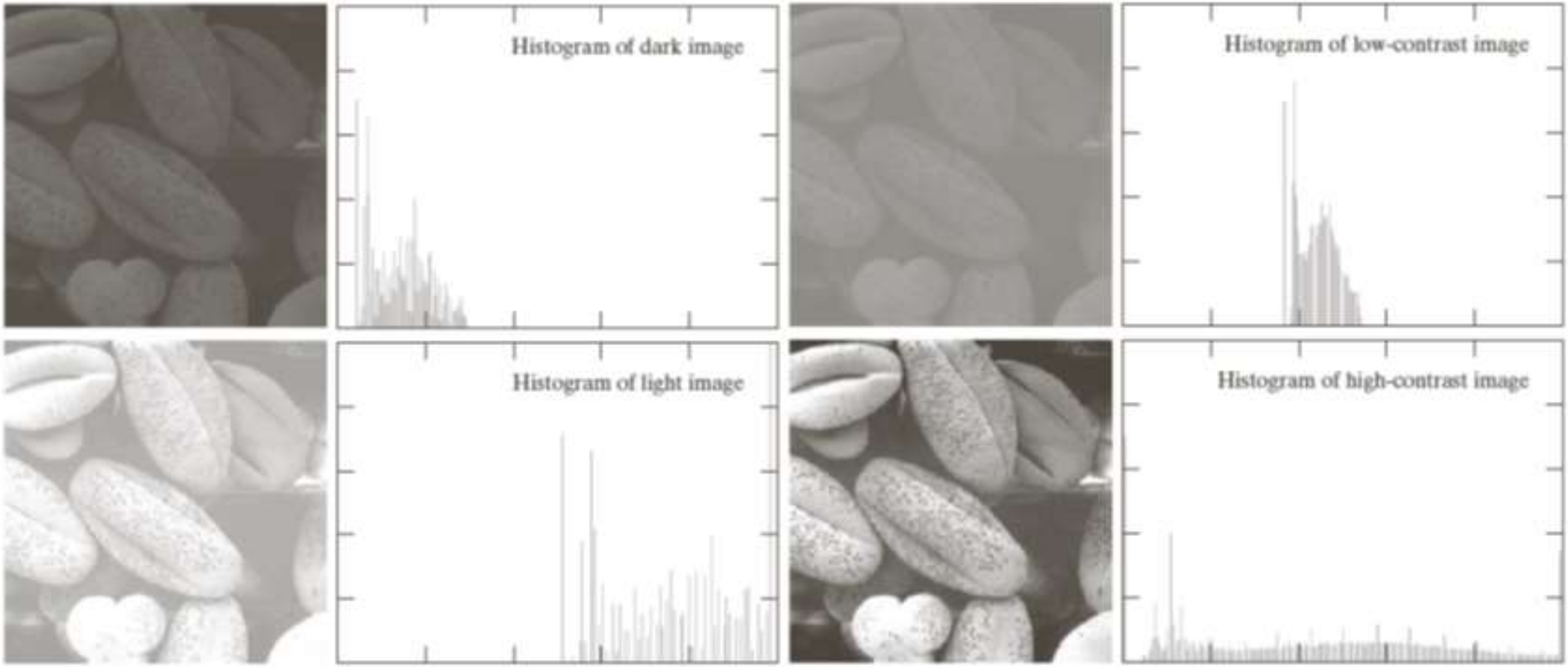


Image Enhancement in Spatial Domain

Histogram Equalization

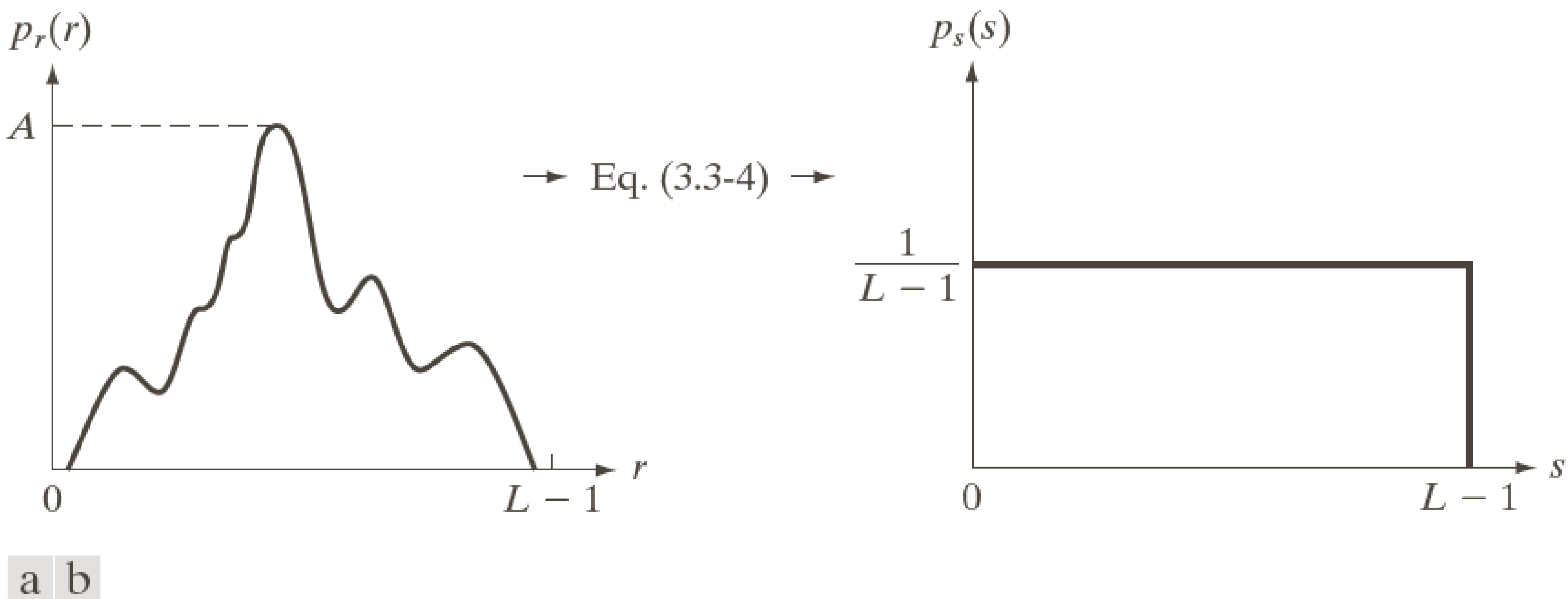


FIGURE 3.18 (a) An arbitrary PDF. (b) Result of applying the transformation in Eq. (3.3-4) to all intensity levels, r. The resulting intensities, s, have a uniform PDF, independently of the form of the PDF of the r's.

$$s = T(r)$$

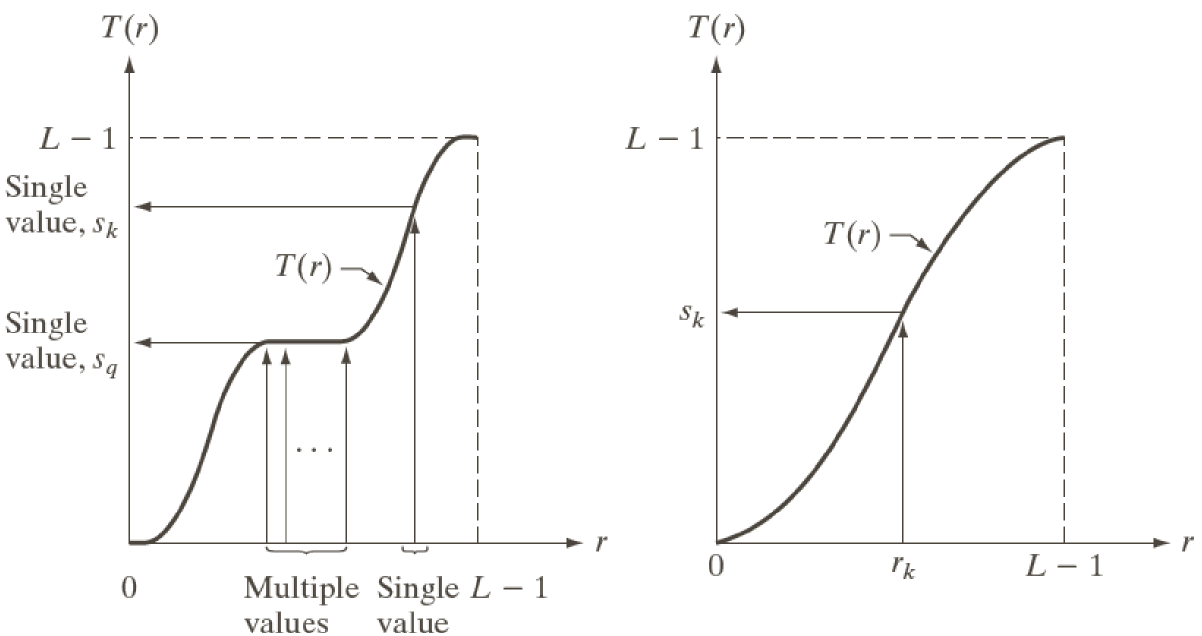
Image Enhancement in Spatial Domain

Histogram Equalization

$$s = T(r) \quad 0 \leq r \leq L-1$$

a) $T(r)$ is single-valued and strictly monotonically increasing function in the interval $\mathbf{0 \leq r \leq L-1}$.

b) $0 \leq T(r) \leq L-1$ for $0 \leq r \leq L-1$.



a b

FIGURE 3.17 (a) Monotonically increasing function, showing how multiple values can map to a single value. (b) Strictly monotonically increasing function. This is a one-to-one mapping, both ways.

Image Enhancement in Spatial Domain

Histogram Equalization

- The objective is to get a uniform histogram of the resultant image $T(r)$.
- The intensity level in an image may be viewed as random variables in the interval $[0\ L-1]$.
- Let $p_r(r)$ and $p_s(s)$ denote the probability density function (PDF) of random variables r and s.
- **$T(r)$ is continuous and differentiable**
- **If $p_r(r)$ and $T(r)$ are known and $T^{-1}(s)$ satisfies condition (a)** then the PDF $p_s(s)$ of the transformed random variable s can be obtained by

$$\boldsymbol{p_s(s) = p_r(r)\frac{dr}{ds}}$$

Image Enhancement in Spatial Domain

Histogram Equalization

$$s = T(r) = (L-1)\int_0^r p_r(w)\,dw$$

Cumulative Distribution Function
Satisfies condition (a) and (b)

$$\frac{ds}{dr} = \frac{dT(r)}{dr} = (L-1)\frac{d}{dr}\left[\int_0^r p_r(w)\,dw\right]$$

$$= (L-1)\,p_r(r)$$

$$p_s(s) = \frac{p_r(r)dr}{ds} = p_r(r)\Big/\left(\frac{ds}{dr}\right) = p_r(r)\Big/\left((L-1)\,p_r(r)\right) = \frac{1}{L-1}$$

Image Enhancement in Spatial Domain

Histogram Equalization

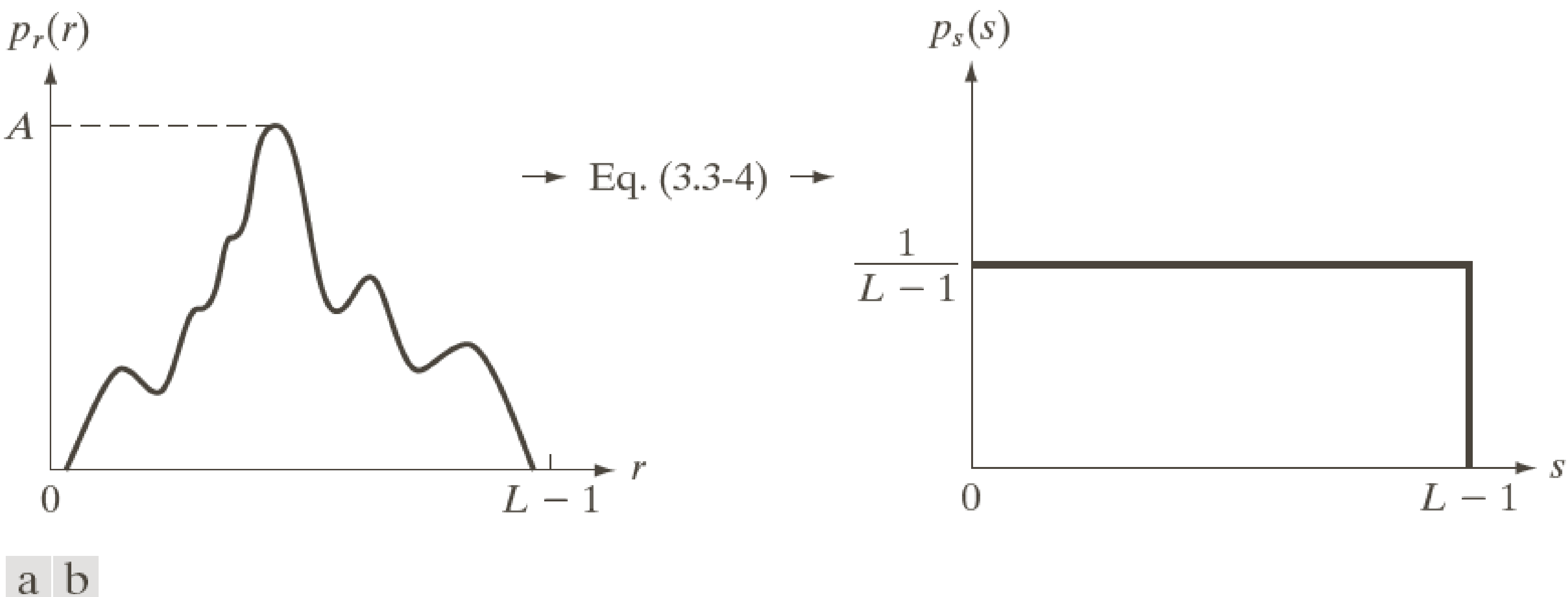


FIGURE 3.18 (a) An arbitrary PDF. (b) Result of applying the transformation in Eq. (3.3-4) to all intensity levels, r. The resulting intensities, s, have a uniform PDF, independently of the form of the PDF of the r's.

Image Enhancement in Spatial Domain

Histogram Equalization

Continuous case:

$$s = T(r) = (L-1)\int_0^r p_r(w)dw$$

Discrete values:

$$s_k = T(r_k) = (L-1)\sum_{j=0}^{k} p_r(r_j)$$

$p_r(r_j)$ is the probability of occurrence of gray level r_j in an image

$$= (L-1)\sum_{j=0}^{k}\frac{n_j}{MN} = \frac{L-1}{MN}\sum_{j=0}^{k} n_j \qquad k=0,1,\ldots, L-1$$

histogram equalization

Image Enhancement in Spatial Domain

Histogram Equalization: Example

Suppose that a 3-bit image ($L = 8$) of size 64×64 pixels ($MN = 4096$) has the intensity distribution shown in following table. Get the histogram equalization transformation function and give the $p_s(s_k)$ for each s_k.

r_k	n_k	$p_r(r_k) = n_k/MN$
$r_0 = 0$	790	0.19
$r_1 = 1$	1023	0.25
$r_2 = 2$	850	0.21
$r_3 = 3$	656	0.16
$r_4 = 4$	329	0.08
$r_5 = 5$	245	0.06
$r_6 = 6$	122	0.03
$r_7 = 7$	81	0.02

Image Enhancement in Spatial Domain

Histogram Equalization: Example

$$s_0 = T(r_0) = 7\sum_{j=0}^{0} p_r(r_j) = 7 \times 0.19 = 1.33 \quad \rightarrow 1$$

$$s_1 = T(r_1) = 7\sum_{j=0}^{1} p_r(r_j) = 7 \times (0.19 + 0.25) = 3.08 \quad \rightarrow 3$$

$$s_2 = 4.55 \quad \rightarrow 5 \qquad s_3 = 5.67 \quad \rightarrow 6$$

$$s_4 = 6.23 \quad \rightarrow 6 \qquad s_5 = 6.65 \quad \rightarrow 7$$

$$s_6 = 6.86 \quad \rightarrow 7 \qquad s_7 = 7.00 \quad \rightarrow 7$$

Image Enhancement in Spatial Domain

Histogram Equalization: Example

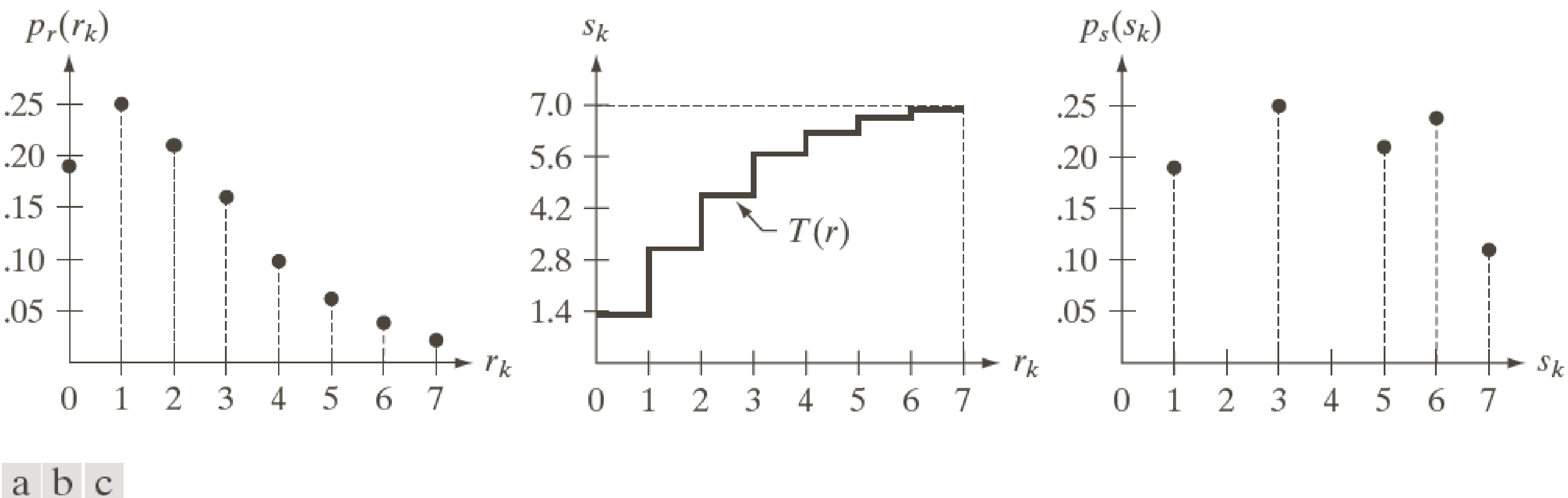


FIGURE 3.19 Illustration of histogram equalization of a 3-bit (8 intensity levels) image. (a) Original histogram. (b) Transformation function. (c) Equalized histogram.

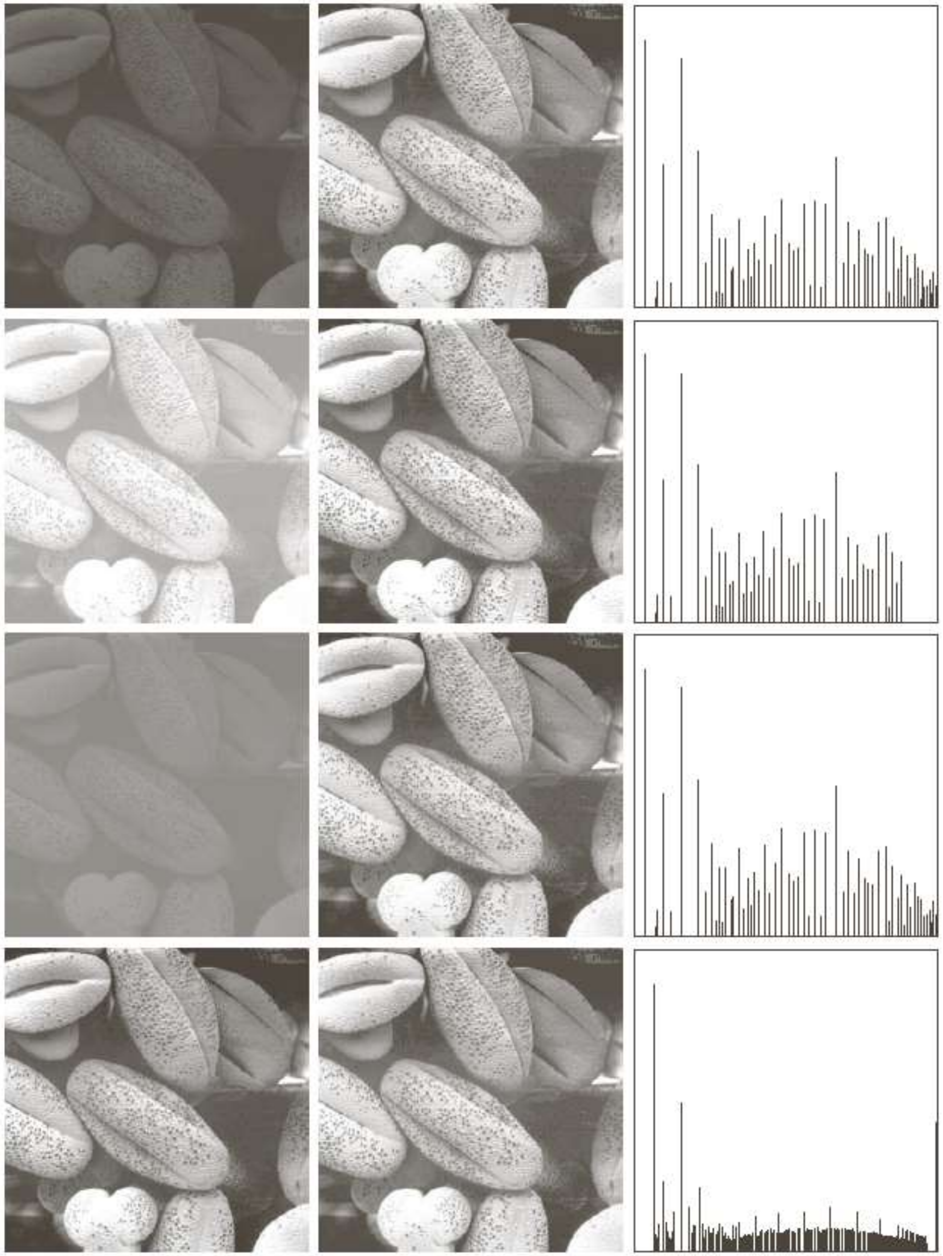


FIGURE 3.20 Left column: images from Fig. 3.16. Center column: corresponding histogram-equalized images. Right column: histograms of the images in the center column.

DIGITAL IMAGE PROCESSING

Unit-2: Image Enhancement in Spatial Domain

Dr. Piyush Joshi

Indian Institute of Information Technology
Sri City, Andhra Pradesh

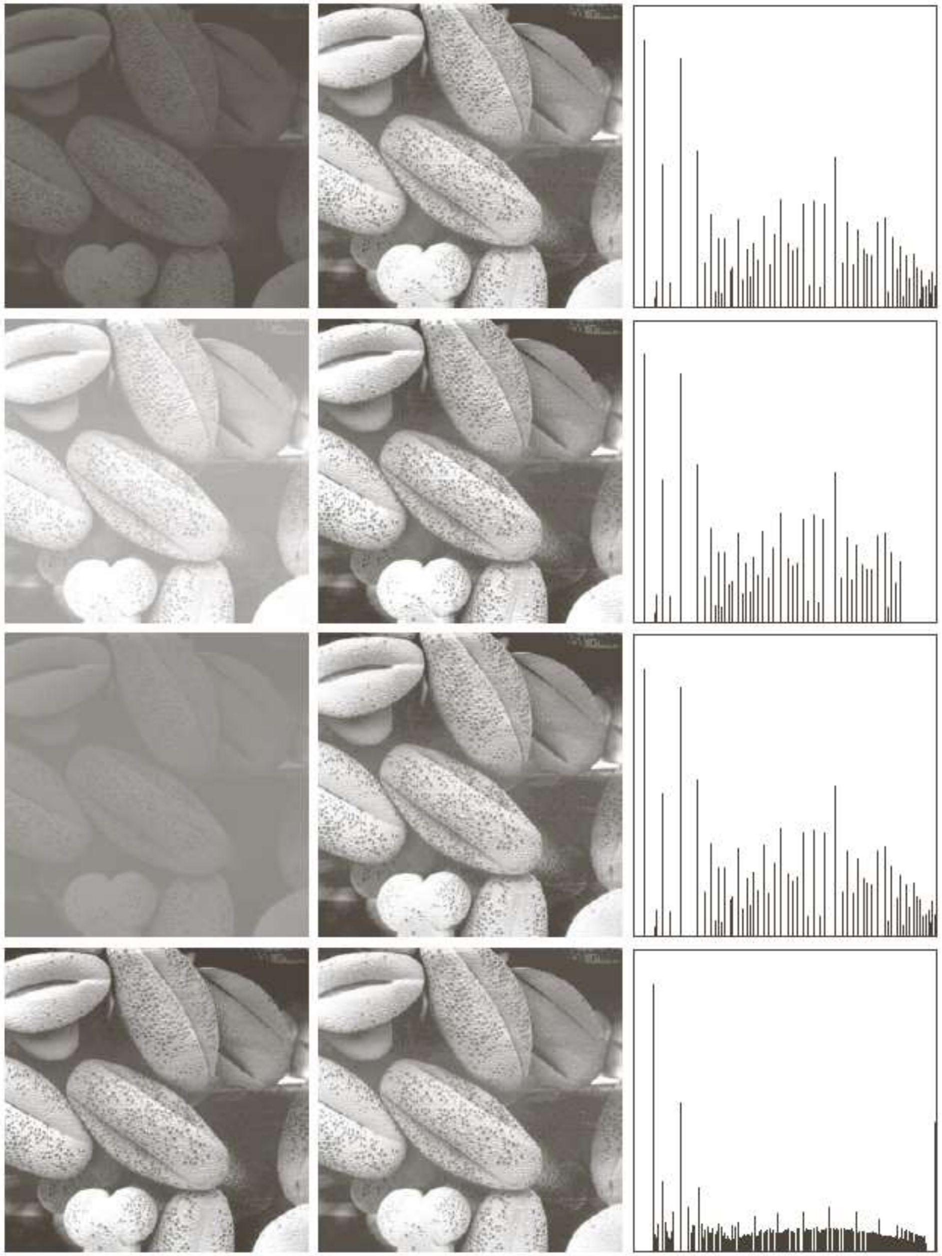
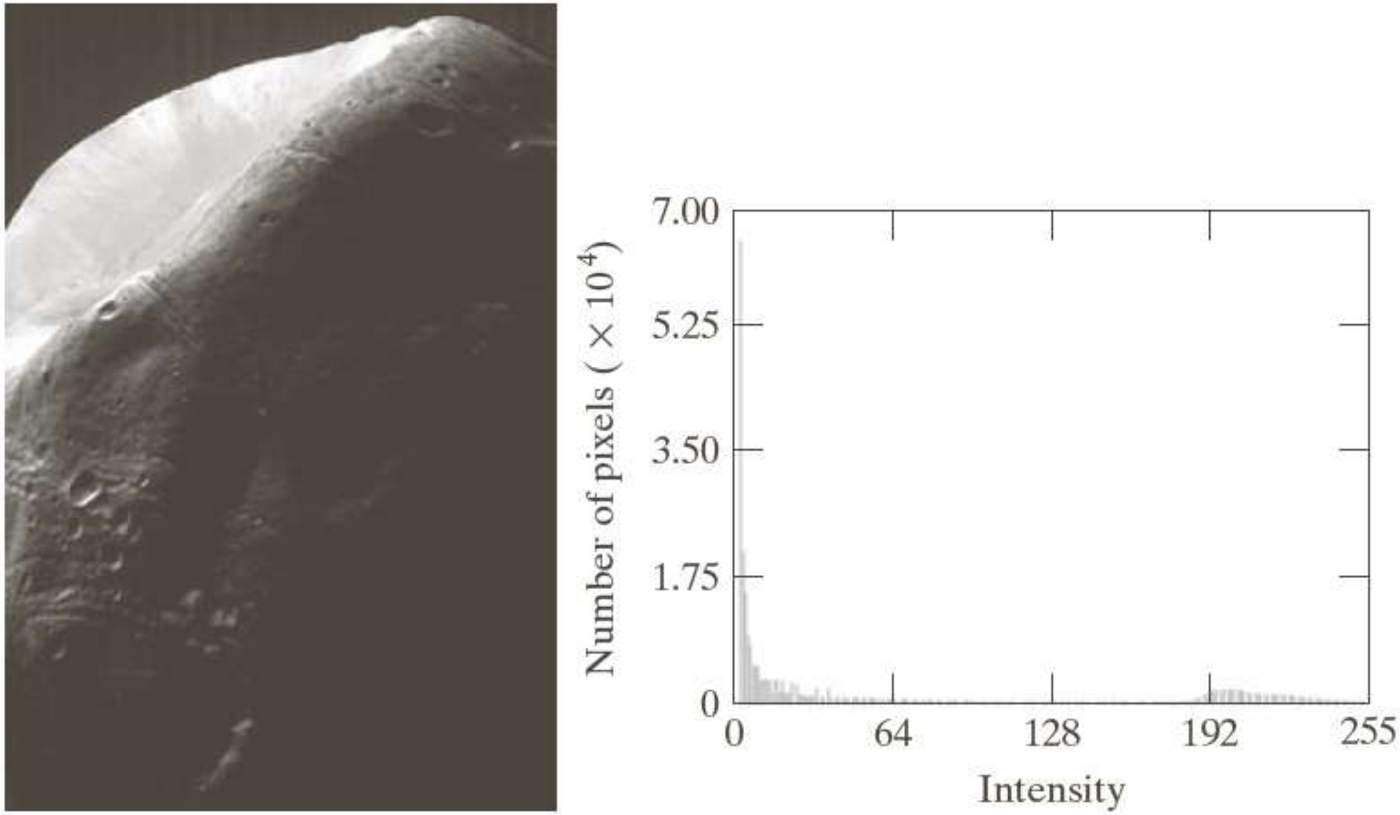


FIGURE 3.20 Left column: images from Fig. 3.16. Center column: corresponding histogram-equalized images. Right column: histograms of the images in the center column.

Image Enhancement in Spatial Domain

Histogram Matching: Example



a b

FIGURE 3.23
(a) Image of the Mars moon Phobos taken by NASA's *Mars Global Surveyor*. (b) Histogram. (Original image courtesy of NASA.)

Image Enhancement in Spatial Domain

Histogram Matching: Example

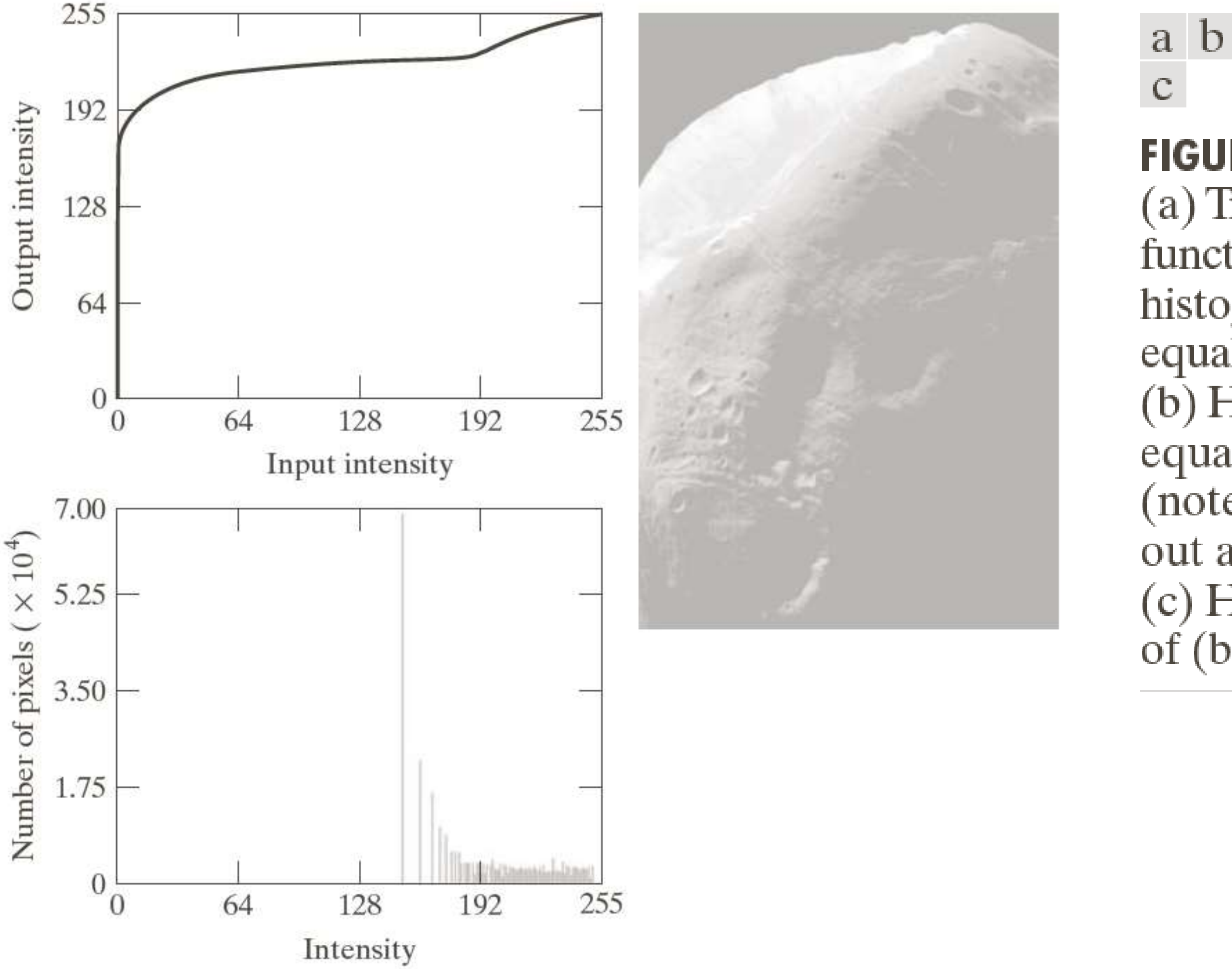


FIGURE 3.24 (a) Transformation function for histogram equalization. (b) Histogram-equalized image (note the washed-out appearance). (c) Histogram of (b).

Histogram Matching (Histogram Specification)

The goal of histogram equalization is to produce an output image that has a flattened histogram

The goal of histogram matching is to take an input image and generate an output image that is based upon the shape of a specific (or reference) histogram.

Image Enhancement in Spatial Domain

Histogram Matching: Going as close as possible to target histogram

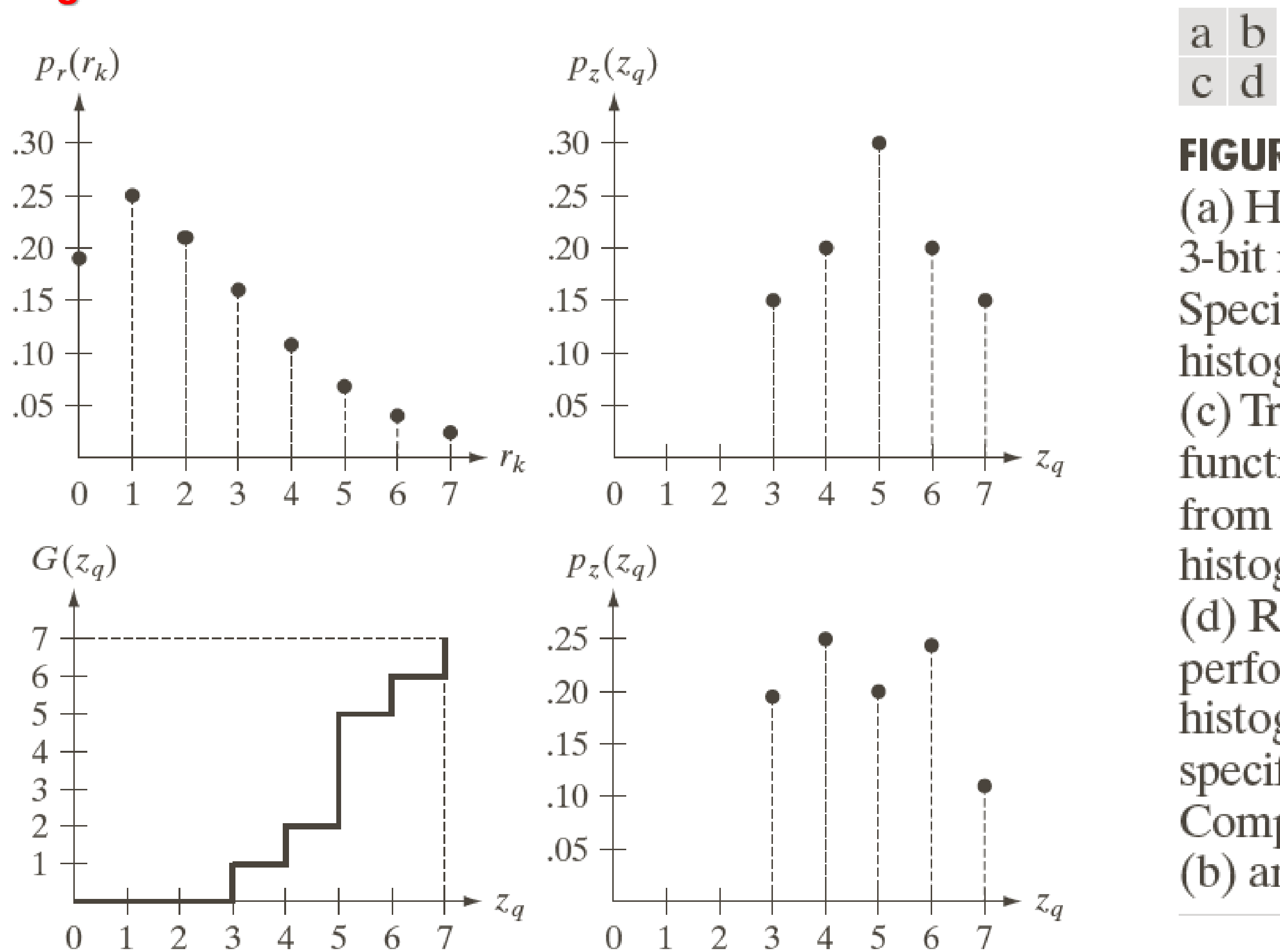
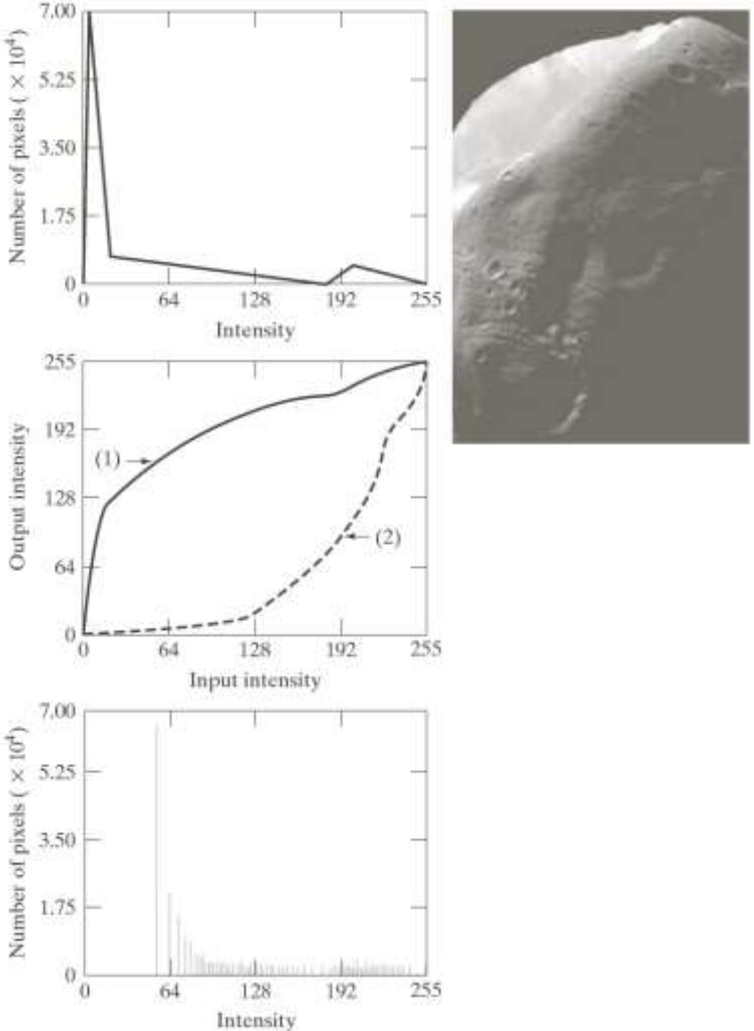


FIGURE 3.22
(a) Histogram of a 3-bit image. (b) Specified histogram. (c) Transformation function obtained from the specified histogram. (d) Result of performing histogram specification. Compare (b) and (d).

Image Enhancement in Spatial Domain

Histogram Matching: Example



a	c
b	
d	

FIGURE 3.25
(a) Specified histogram.
(b) Transformations.
(c) Enhanced image using mappings from curve (2).
(d) Histogram of (c).

Image Enhancement in Spatial Domain

Histogram Matching (Histogram Specification)

- **Generate a processed image that has a specified histogram**

Let $p_r(r)$ and $p_z(z)$ denote the continous probability density functions of the variables r and z. $p_z(z)$ is the specified probability density function.

Let s be the random variable with the probability

$$s = T(r) = (L-1)\int_0^r p_r(w)dw$$

Define a random variable z with the probability

$$G(z) = (L-1)\int_0^z p_z(t)dt = s$$

Image Enhancement in Spatial Domain

Histogram Matching (Histogram Specification)

$$s = T(r) = (L-1)\int_0^r p_r(w)dw$$

$$G(z) = (L-1)\int_0^z p_z(t)dt = s$$

$$z = G^{-1}(s) = G^{-1}\left[T(r)\right]$$

Note: $T(r)$ can be obtained once $p_r(r)$ has been estimated from the input image. $G(z)$ can be obtained since $p_z(z)$ is given.

Image Enhancement in Spatial Domain

Histogram Matching: Procedure

- Obtain $p_r(r)$ from the input image and then obtain the values of s

$$s = (L-1)\int_0^r p_r(w)dw$$

- Use the specified PDF and obtain the transformation function $G(z)$

$$G(z) = (L-1)\int_0^z p_z(t)dt = s$$

- Mapping from s to z

$$z = G^{-1}(s)$$

Image Enhancement in Spatial Domain

Histogram Matching: Discrete Case

- Obtain $p_r(r_j)$ from the input image and then obtain the values of s_k, round the value to the integer range $[0\ L-1]$.

$$s_k = T(r_k) = (L-1)\sum_{j=0}^{k} p_r(r_j) = \frac{(L-1)}{MN}\sum_{j=0}^{k} n_j$$

- Use the specified PDF and obtain the transformation function $G(z_q)$, round the value to the integer range $[0\ L-1]$.

$$G(z_q) = (L-1)\sum_{i=0}^{q} p_z(z_i) = s_k$$

- Mapping from s_k to z_q

$$z_q = G^{-1}(s_k)$$

Example

(a)

Gray level.	0	1	2	3	4	5	6	7
No. of pixels	790	1023	850	656	329	245	122	81

(b)

Gray level.	0	1	2	3	4	5	6	7
No. of pixels	0	0	0	614	819	1230	819	614

Example: HE1

Gray level	nk	PDF	CDF	Sk x 7	Round off	New nk.
0	790	0.19	0.19	1.33	1	790
1	1023	0.25	0.44	3.08	3	1023
2	850	0.21	0.65	4.55	5	850
3	656	0.16	0.81	5.67	6	985
4	329	0.08	0.89	6.23	6 656+329	
5	245	0.06	0.95	6.65	7	
6	122	0.03	0.98	6.86	7 245 +122+81	448
7	81	0.02	1	7	7	
	N=4096					

Example: HE2

Gray level	nk	PDF	CDF	Sk x 7	Round off
0	0	0	0	0	0
1	0	0	0	0	0
2	0	0	0	0	0
3	614	0.149	0.149	1.05	1
4	819	0.20	0.35	2.50	3
5	1230	0.30	0.65	4.55	5
6	819	0.20	0.85	5.97	6
7	614	0.15	1	7	7
	N=4096				

Image Enhancement in Spatial Domain

Histogram Matching: Example (Discrete Case)

Suppose that a 3-bit image ($L = 8$) of size 64×64 pixels ($MN = 4096$) has the intensity distribution shown in following table(on the left). Get the histogram transformation function and make the output image with the specified histogram, listed in the table on the right.

r_k	n_k	$p_r(r_k) = n_k/MN$
$r_0 = 0$	790	0.19
$r_1 = 1$	1023	0.25
$r_2 = 2$	850	0.21
$r_3 = 3$	656	0.16
$r_4 = 4$	329	0.08
$r_5 = 5$	245	0.06
$r_6 = 6$	122	0.03
$r_7 = 7$	81	0.02

z_q	Specified $p_z(z_q)$	Actual $p_z(z_k)$
$z_0 = 0$	0.00	0.00
$z_1 = 1$	0.00	0.00
$z_2 = 2$	0.00	0.00
$z_3 = 3$	0.15	0.19
$z_4 = 4$	0.20	0.25
$z_5 = 5$	0.30	0.21
$z_6 = 6$	0.20	0.24
$z_7 = 7$	0.15	0.11

Image Enhancement in Spatial Domain

Histogram Matching: Example (Discrete Case)

Obtain the scaled histogram-equalized values,

$$s_0 = 1, s_1 = 3, s_2 = 5, s_3 = 6, s_4 = 7,$$
$$s_5 = 7, s_6 = 7, s_7 = 7.$$

Compute all the values of the transformation function G,

$$G(z_0) = 7\sum_{j=0}^{0} p_z(z_j) = 0.00 \rightarrow 0$$

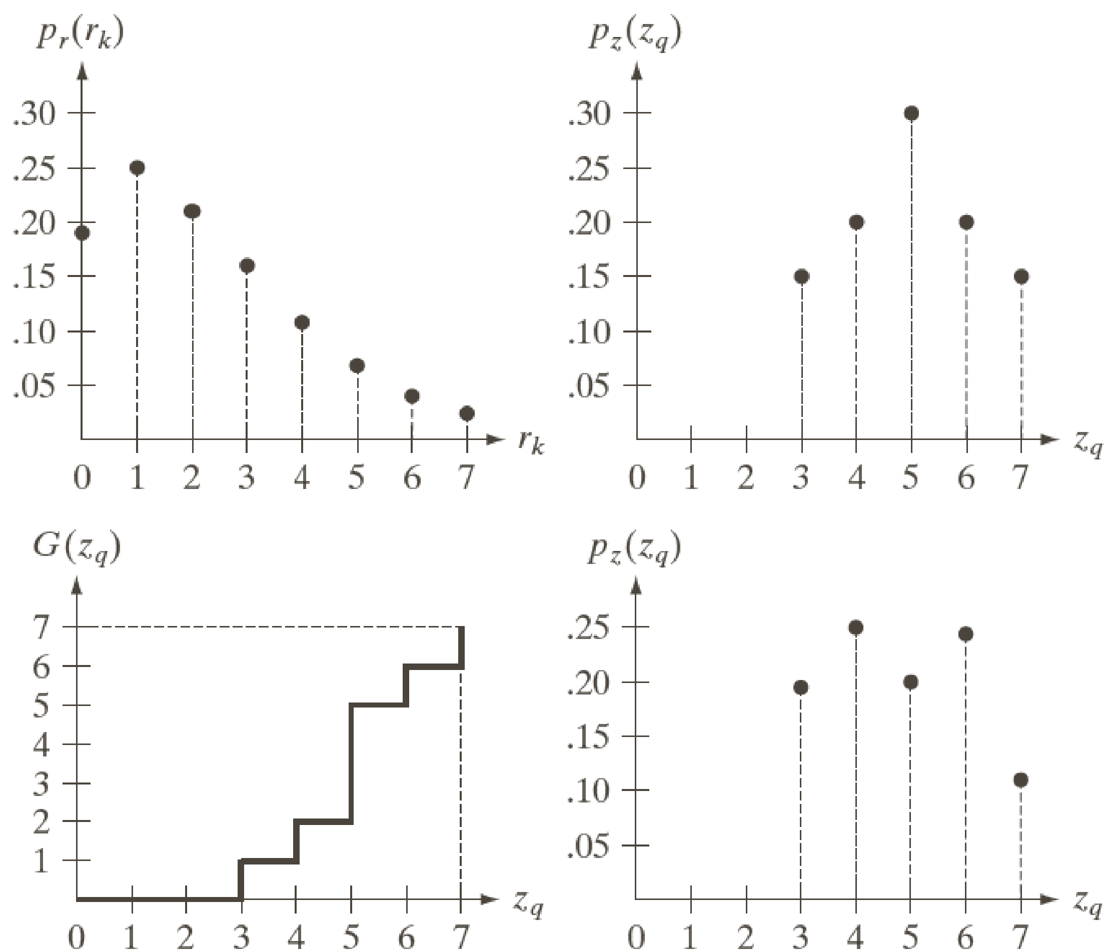
$$G(z_1) = 0.00 \rightarrow 0 \qquad G(z_2) = 0.00 \rightarrow 0$$
$$G(z_3) = 1.05 \rightarrow 1 \qquad G(z_4) = 2.45 \rightarrow 2$$
$$G(z_5) = 4.55 \rightarrow 5 \qquad G(z_6) = 5.95 \rightarrow 6$$
$$G(z_7) = 7.00 \rightarrow 7$$

r_k	n_k	$p_r(r_k) = n_k/MN$
$r_0 = 0$	790	0.19
$r_1 = 1$	1023	0.25
$r_2 = 2$	850	0.21
$r_3 = 3$	656	0.16
$r_4 = 4$	329	0.08
$r_5 = 5$	245	0.06
$r_6 = 6$	122	0.03
$r_7 = 7$	81	0.02

z_q	Specified $p_z(z_q)$	Actual $p_z(z_k)$
$z_0 = 0$	0.00	0.00
$z_1 = 1$	0.00	0.00
$z_2 = 2$	0.00	0.00
$z_3 = 3$	0.15	0.19
$z_4 = 4$	0.20	0.25
$z_5 = 5$	0.30	0.21
$z_6 = 6$	0.20	0.24
$z_7 = 7$	0.15	0.11

Image Enhancement in Spatial Domain

Histogram Matching: Example (Discrete Case)



a	b
c	d

FIGURE 3.22
(a) Histogram of a 3-bit image. (b) Specified histogram. (c) Transformation function obtained from the specified histogram. (d) Result of performing histogram specification. Compare (b) and (d).

Image Enhancement in Spatial Domain

Histogram Matching: Example (Discrete Case)

- Obtain the scaled histogram-equalized values,

$$s_0 = 1, s_1 = 3, s_2 = 5, s_3 = 6, s_4 = 7,$$
$$s_5 = 7, s_6 = 7, s_7 = 7.$$

- Compute all the values of the transformation function G,

$$G(z_0) = 7\sum_{j=0}^{0} p_z(z_j) = 0.00 \rightarrow 0$$

$G(z_1) = 0.00 \rightarrow 0$ $\qquad$ $G(z_2) = 0.00 \rightarrow 0$

$G(z_3) = 1.05 \rightarrow 1$ **s_0** $\qquad$ $G(z_4) = 2.45 \rightarrow 2$ **s_1**

$G(z_5) = 4.55 \rightarrow 5$ **s_2** $\qquad$ $G(z_6) = 5.95 \rightarrow 6$ **s_3**

$G(z_7) = 7.00 \rightarrow 7$ **s_4 s_5 s_6 s_7**

Image Enhancement in Spatial Domain

Histogram Matching: Example (Discrete Case)

$$s_0 = 1, s_1 = 3, s_2 = 5, s_3 = 6, s_4 = 7,$$

$$s_5 = 7, s_6 = 7, s_7 = 7.$$

r_k

0

1

2

3

4

5

6

7

s_k	$\rightarrow$	z_q
1	$\rightarrow$	3
3	$\rightarrow$	4
5	$\rightarrow$	5
6	$\rightarrow$	6
7	$\rightarrow$	7

Image Enhancement in Spatial Domain

Histogram Matching: Example (Discrete Case)

$$r_k \rightarrow z_q$$

$$0 \rightarrow 3$$

$$1 \rightarrow 4$$

$$2 \rightarrow 5$$

$$3 \rightarrow 6$$

$$4 \rightarrow 7$$

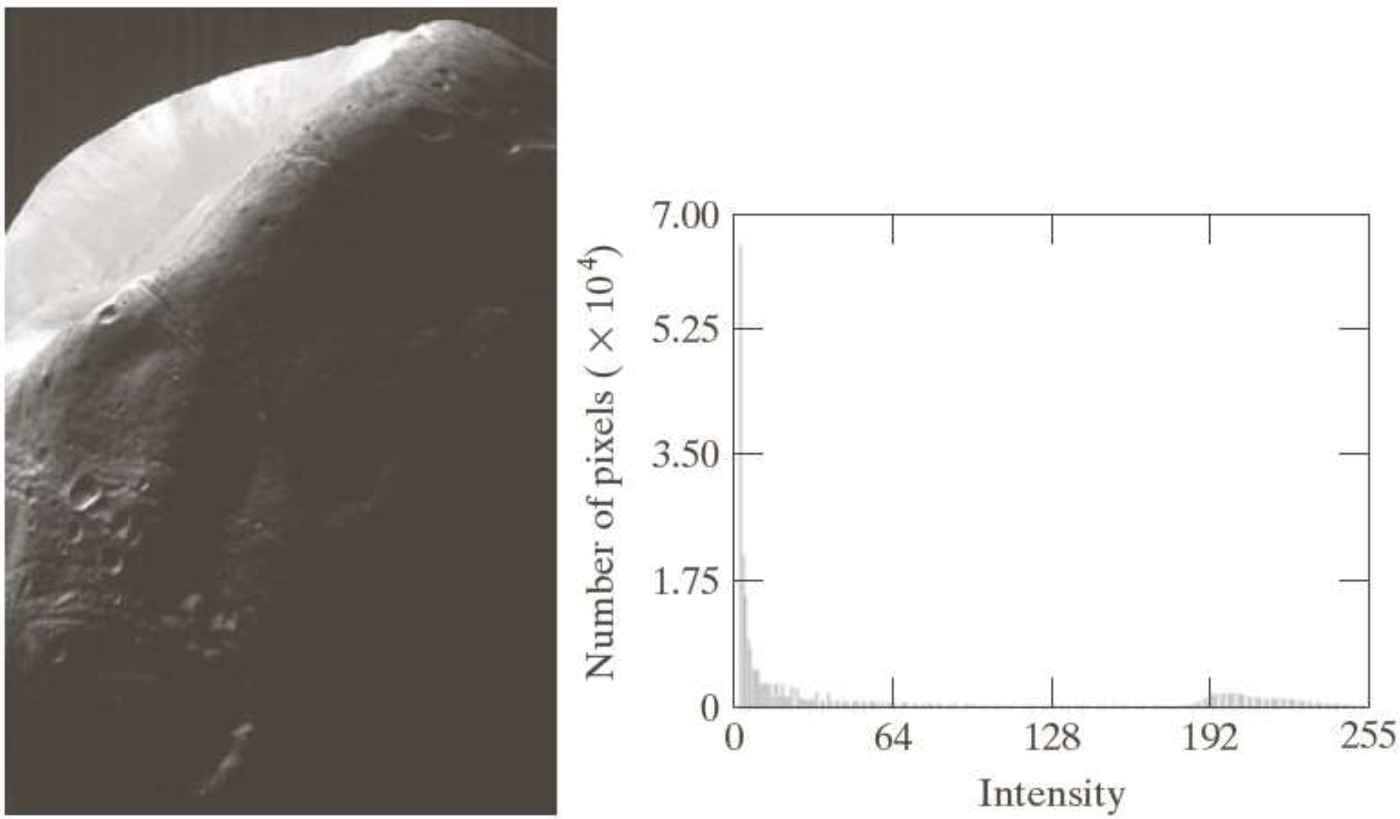
$$5 \rightarrow 7$$

$$6 \rightarrow 7$$

$$7 \rightarrow 7$$

Image Enhancement in Spatial Domain

Histogram Matching: Example

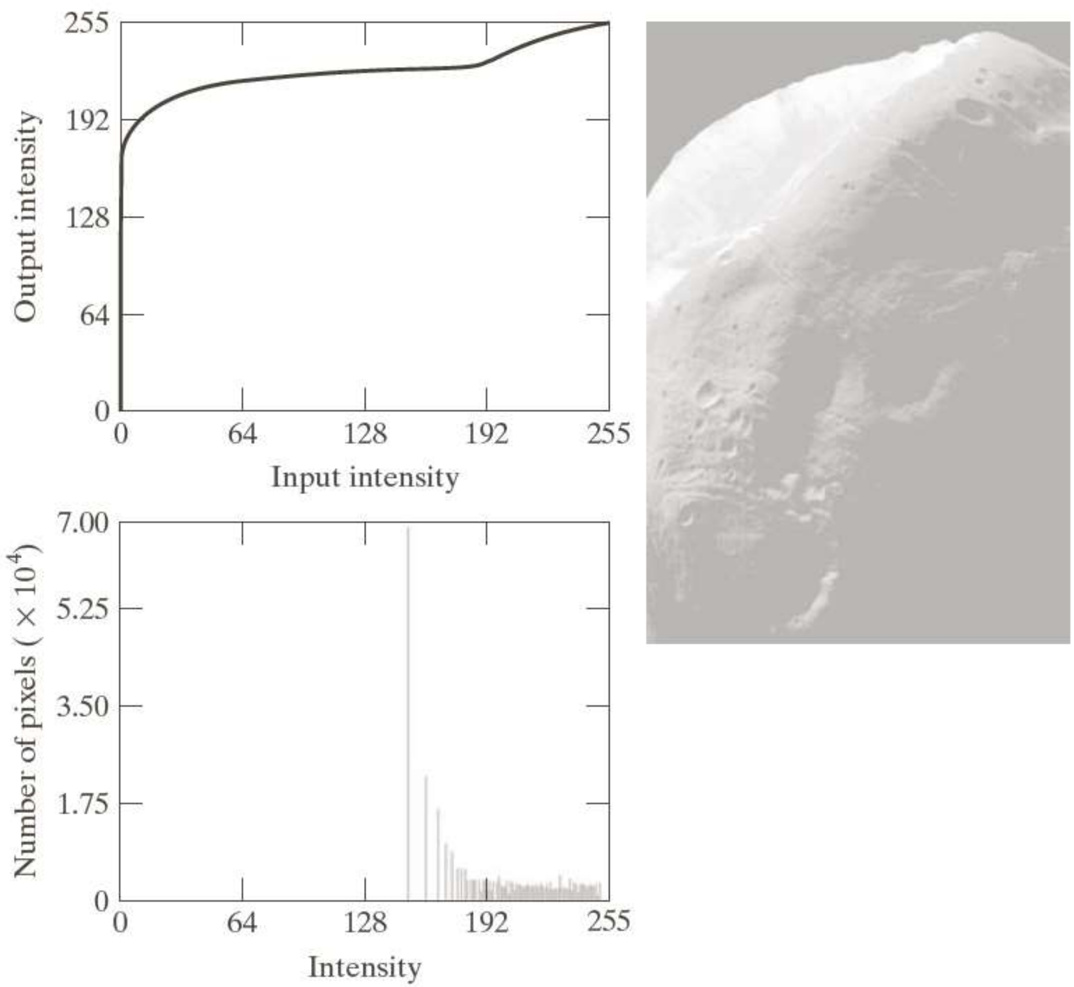


a b

FIGURE 3.23 (a) Image of the Mars moon Phobos taken by NASA's *Mars Global Surveyor*. (b) Histogram. (Original image courtesy of NASA.)

Image Enhancement in Spatial Domain

Histogram Matching: Example

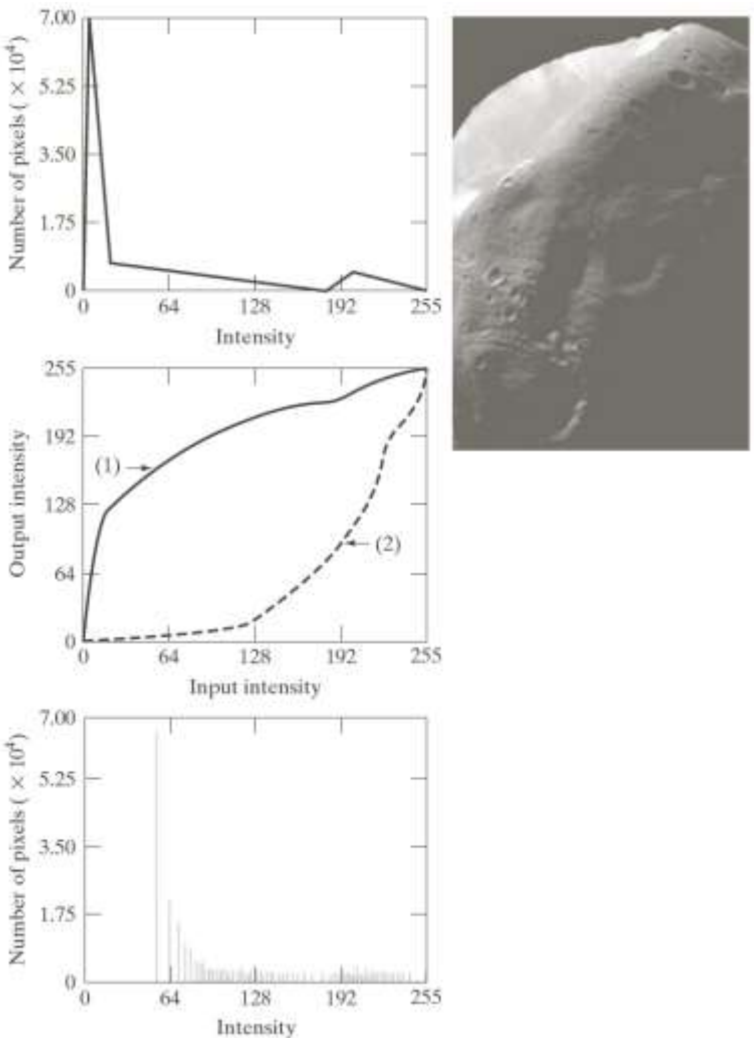


a b
c

FIGURE 3.24
(a) Transformation function for histogram equalization.
(b) Histogram-equalized image (note the washed-out appearance).
(c) Histogram of (b).

Image Enhancement in Spatial Domain

Histogram Matching: Example



a	c
b	
d	

FIGURE 3.25
(a) Specified histogram.
(b) Transformations.
(c) Enhanced image using mappings from curve (2).
(d) Histogram of (c).

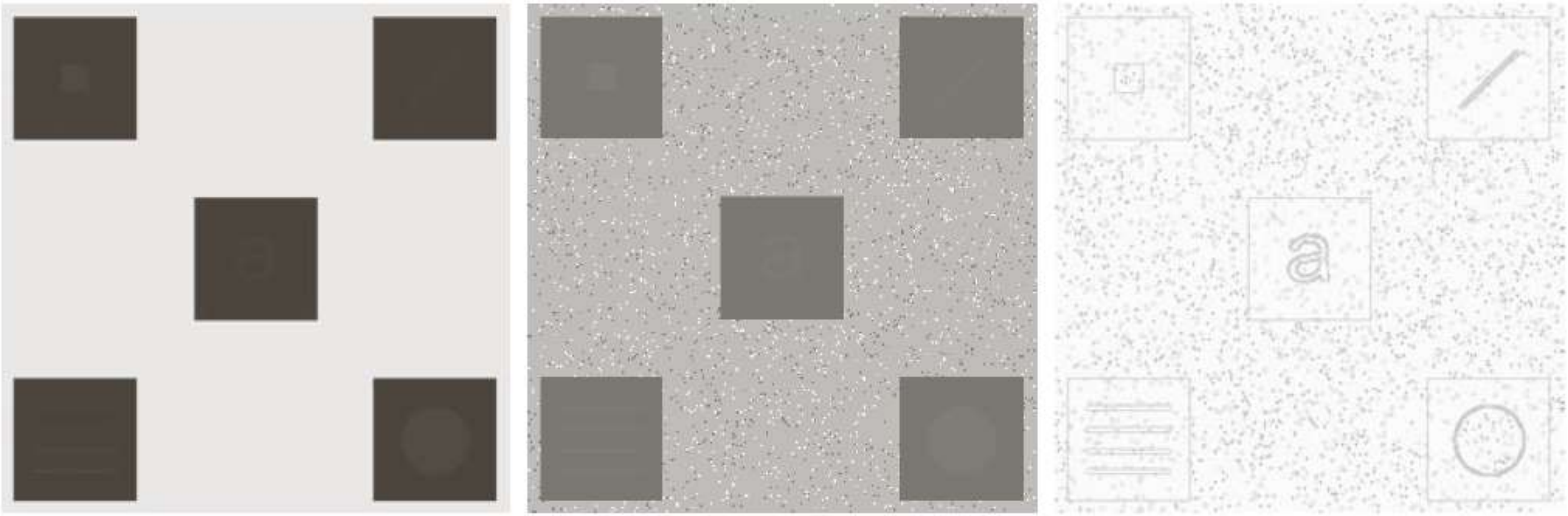
Image Enhancement in Spatial Domain

Local Histogram Processing

- Define a neighborhood and move its center from pixel to pixel.
- At each location, the histogram of the points in the neighborhood is computed. Either histogram equalization or histogram specification transformation function is obtained.
- Map the intensity of the pixel centered in the neighborhood.
- Move to the next location and repeat the procedure.

Image Enhancement in Spatial Domain

Local Histogram Processing



a b c

FIGURE 3.26 (a) Original image. (b) Result of global histogram equalization. (c) Result of local histogram equalization applied to (a), using a neighborhood of size 3 × 3.

Image Enhancement in Spatial Domain

Histogram Statistics for Image Enhancement

Average Intensity

$$m = \sum_{i=0}^{L-1} r_i p(r_i) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y)$$

$$u_n(r) = \sum_{i=0}^{L-1} (r_i - m)^n p(r_i)$$

Nth moment of mean

Variance

$$\sigma^2 = u_2(r) = \sum_{i=0}^{L-1} (r_i - m)^2 p(r_i) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [f(x, y) - m]^2$$

Image Enhancement in Spatial Domain

Histogram Statistics for Image Enhancement

Local average intensity

$$m_{s_{xy}} = \sum_{i=0}^{L-1} r_i p_{s_{xy}}(r_i)$$

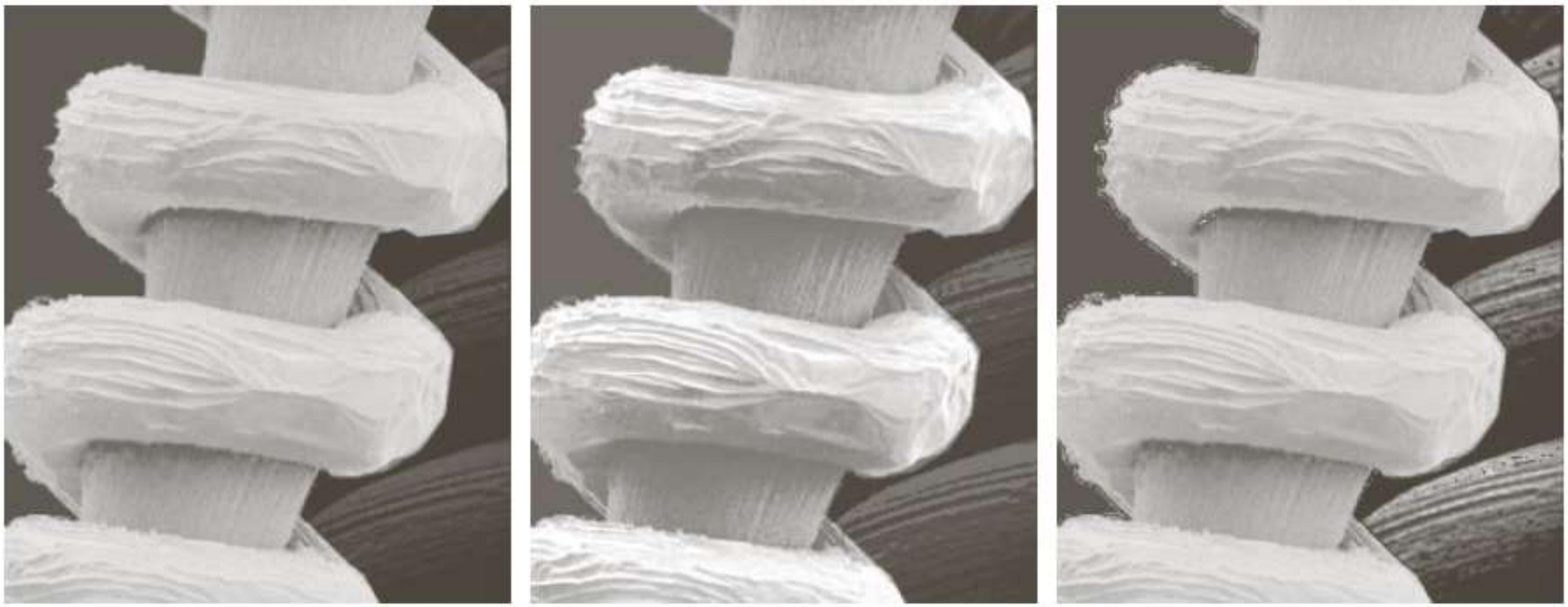
s_{xy} denotes a neighborhood

Local variance

$$\sigma_{s_{xy}}^2 = \sum_{i=0}^{L-1} (r_i - m_{s_{xy}})^2 p_{s_{xy}}(r_i)$$

Image Enhancement in Spatial Domain

Histogram Statistics for Image Enhancement



a b c

FIGURE 3.27 (a) SEM image of a tungsten filament magnified approximately 130×. (b) Result of global histogram equalization. (c) Image enhanced using local histogram statistics. (Original image courtesy of Mr. Michael Shaffer, Department of Geological Sciences, University of Oregon, Eugene.)

Image Enhancement in Spatial Domain

Spatial Filtering

- The process of linear filtering is called convolution.
- The filter masks are sometimes called convolution masks or convolution kernel.

Image Enhancement in Spatial Domain

Smoothing Spatial Filters

- Smoothing filters are used for blurring and for noise reduction.
- Blurring is used in removal of small details and bridging of small gaps in lines or curves.
- Smoothing spatial filters include linear filters and nonlinear filters.
- These filters sometimes are called ***averaging filters.***
- They are also referred to a ***lowpass filters*** in frequency domain.

Spatial Filtering

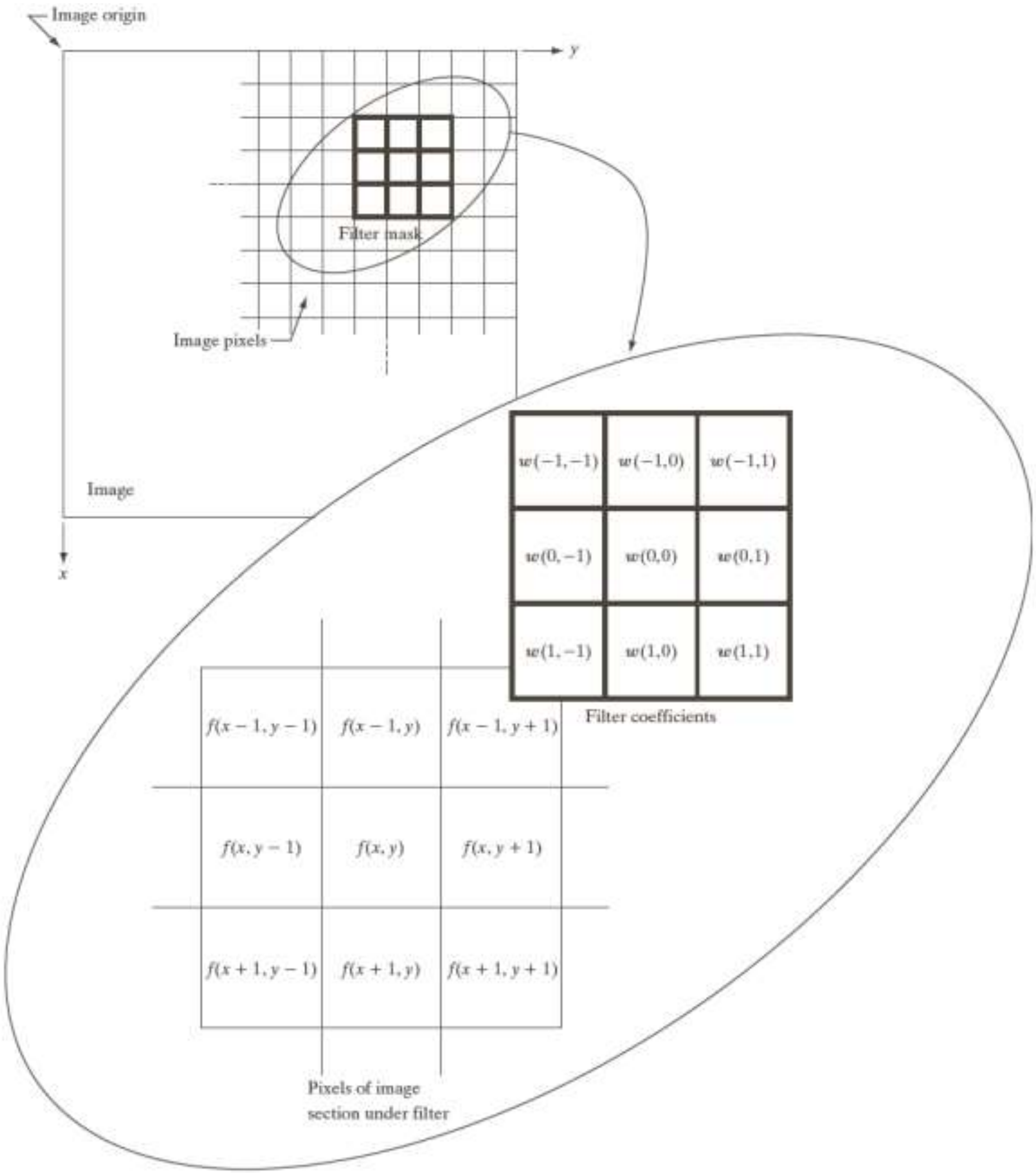


Image Enhancement in Spatial Domain

Smoothing Linear Spatial Filters

The general implementation for filtering an M × N image with a weighted averaging filter of size m × n is given

$$g(x,y)=\frac{\sum_{s=-a}^{a}\sum_{t=-b}^{b} w(s,t)\,f(x+s,y+t)}{\sum_{s=-a}^{a}\sum_{t=-b}^{b} w(s,t)}$$

where $m=2a+1,\quad n=2b+1.$

Image Enhancement in Spatial Domain

Two Smoothing Averaging Filter Masks

a b

$\frac{1}{9} \times$

1	1	1
1	1	1
1	1	1

average

$\frac{1}{16} \times$

1	2	1
2	4	2
1	2	1

weighted average

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.

Two Smoothing Averaging Filter Masks

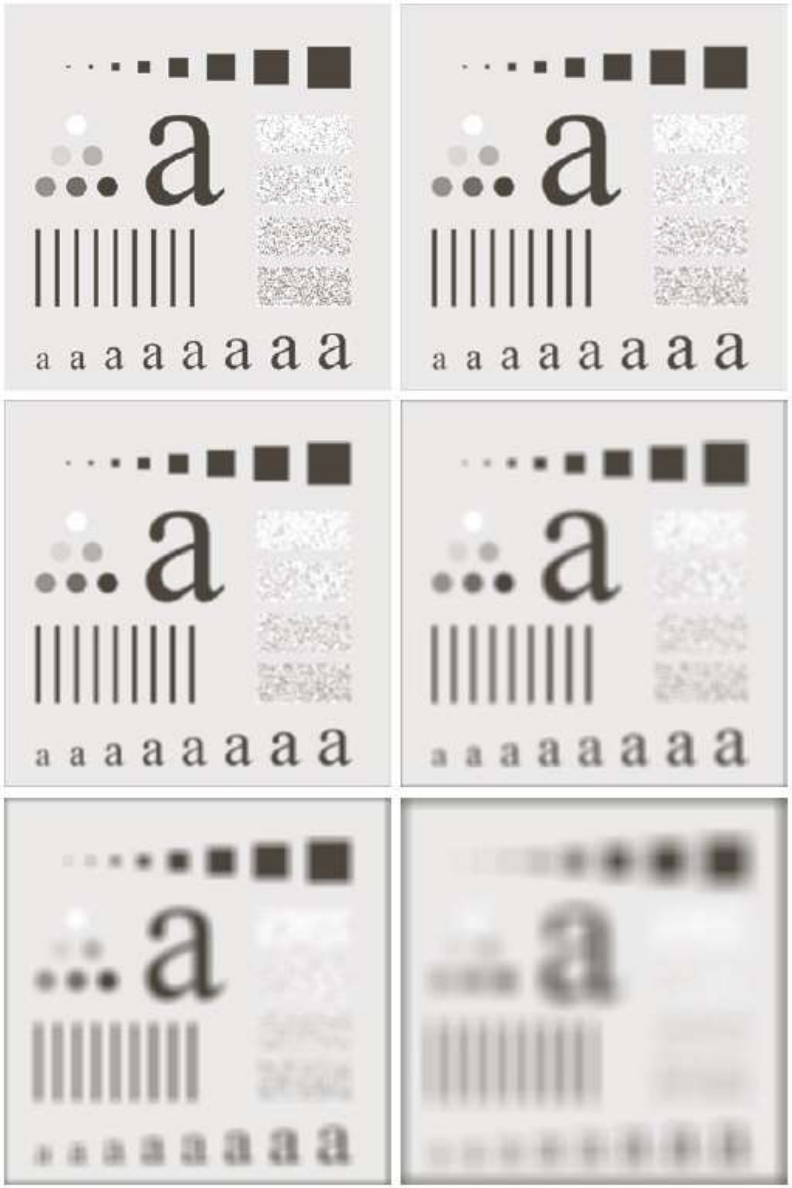
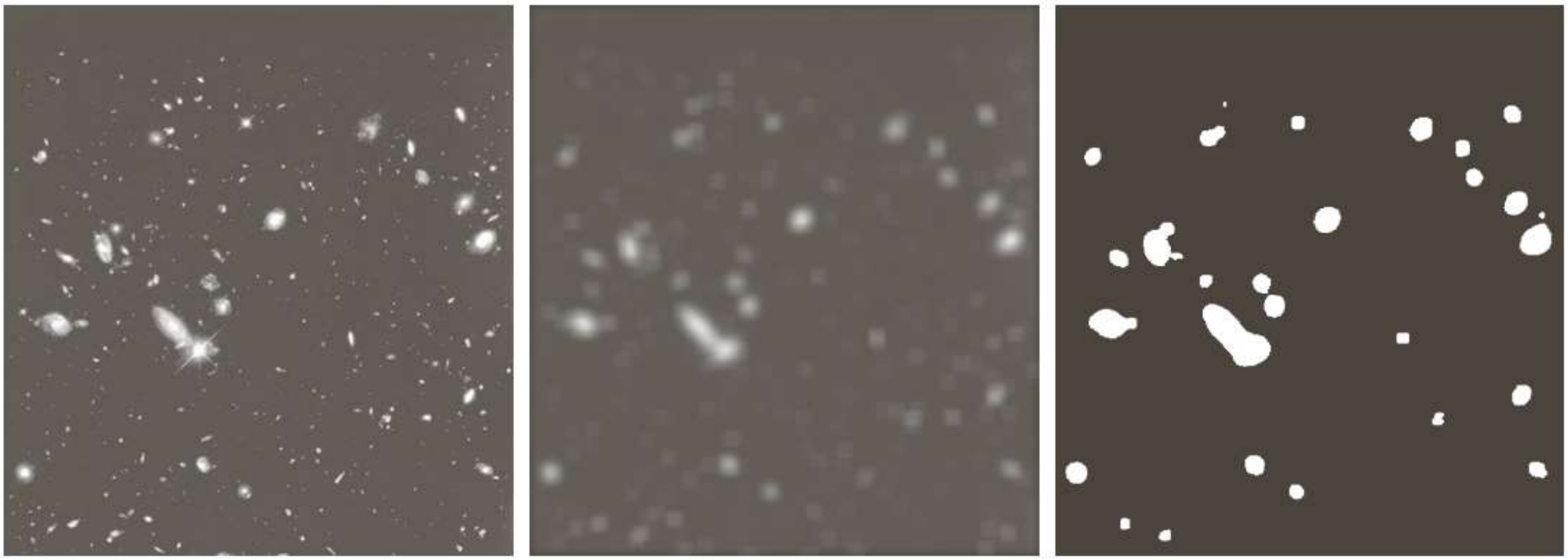


FIGURE 3.33 (a) Original image, of size 500 × 500 pixels. (b)–(f) Results of smoothing with square averaging filter masks of sizes $m = 3, 5, 9, 15$, and 35, respectively. The black squares at the top are of sizes 3, 5, 9, 15, 25, 35, 45, and 55 pixels, respectively; their borders are 25 pixels apart. The letters at the bottom range in size from 10 to 24 points, in increments of 2 points; the large letter at the top is 60 points. The vertical bars are 5 pixels wide and 100 pixels high; their separation is 20 pixels. The diameter of the circles is 25 pixels, and their borders are 15 pixels apart; their intensity levels range from 0% to 100% black in increments of 20%. The background of the image is 10% black. The noisy rectangles are of size 50 × 120 pixels.

a b
c d
e f

Image Enhancement in Spatial Domain

Example: Gross Representation of Objects



a b c

FIGURE 3.34 (a) Image of size 528 × 485 pixels from the Hubble Space Telescope. (b) Image filtered with a 15 × 15 averaging mask. (c) Result of thresholding (b). (Original image courtesy of NASA.)

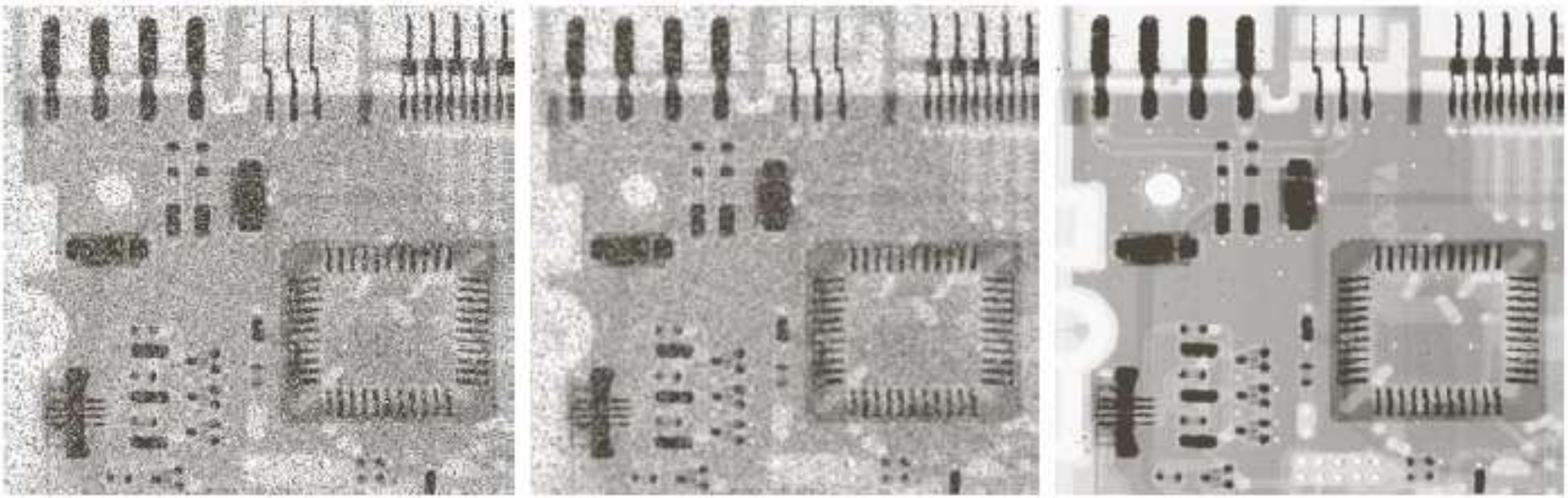
Image Enhancement in Spatial Domain

Order-Statistic (Nonlinear) Filters

- Nonlinear
- Based on ordering (ranking) the pixels contained in the filter mask
- Replacing the value of the center pixel with the value determined by the ranking result
- E.g., median filter, max filter, min filter

Image Enhancement in Spatial Domain

Example: Use of Median Filtering for Noise Reduction



a b c

FIGURE 3.35 (a) X-ray image of circuit board corrupted by salt-and-pepper noise. (b) Noise reduction with a 3 × 3 averaging mask. (c) Noise reduction with a 3 × 3 median filter. (Original image courtesy of Mr. Joseph E. Pascente, Lixi, Inc.)

Image Enhancement in Spatial Domain

Sharpening Spatial Filters

- Foundation
- Laplacian Operator
- Unsharp Masking and Highboost Filtering
- Using First-Order Derivatives for Nonlinear Image Sharpening — The Gradient

Image Enhancement in Spatial Domain

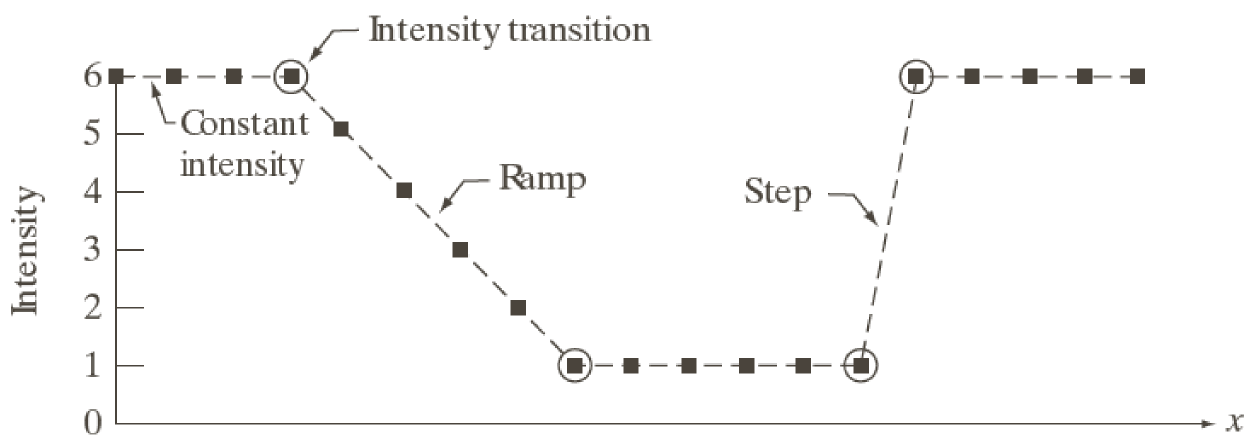
Sharpening Spatial Filters: **Foundation**

- The first-order derivative of a one-dimensional function f(x) is the difference

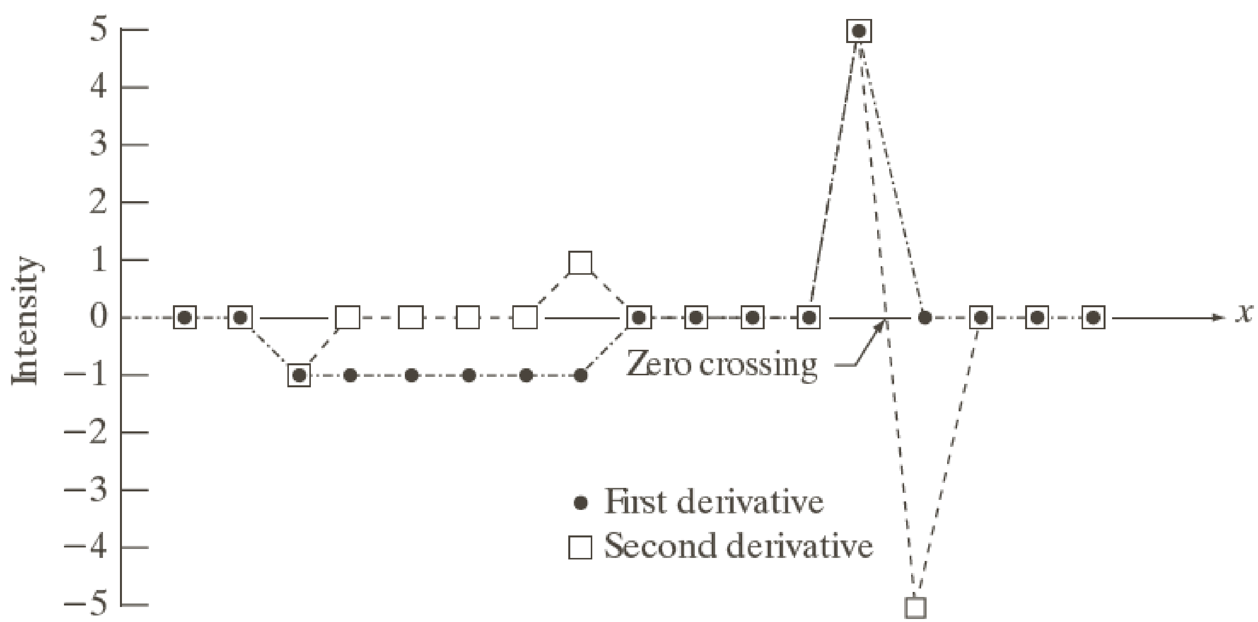
$$\frac{\partial f}{\partial x} = f(x+1) - f(x)$$

- The second-order derivative of f(x) as the difference

$$\frac{\partial^2 f}{\partial x^2} = f(x+1) + f(x-1) - 2f(x)$$



Scan line	6	6	6	6	5	4	3	2	1	1	1	1	1	1	6	6	6	6	6 → x
1st derivative		0	0	−1	−1	−1	−1	−1	0	0	0	0	0	5	0	0	0	0	
2nd derivative		0	0	−1	0	0	0	0	1	0	0	0	0	5	−5	0	0	0	



a
b
c

FIGURE 3.36 Illustration of the first and second derivatives of a 1-D digital function representing a section of a horizontal intensity profile from an image. In (a) and (c) data points are joined by dashed lines as a visualization aid.

Image Enhancement in Spatial Domain

Sharpening Spatial Filters: **Laplacian Operator**

The second-order isotropic derivative operator is the Laplacian for a function (image) $\boldsymbol{f(x, y)}$

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

$$\frac{\partial^2 f}{\partial x^2} = f(x+1, y) + f(x-1, y) - 2f(x, y)$$

$$\frac{\partial^2 f}{\partial y^2} = f(x, y+1) + f(x, y-1) - 2f(x, y)$$

$$\nabla^2 f = f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1)$$
$$-4f(x, y)$$

Image Enhancement in Spatial Domain

Sharpening Spatial Filters: Laplacian Operator

(a)

0	1	0
1	−4	1
0	1	0

(b)

1	1	1
1	−8	1
1	1	1

(c)

0	−1	0
−1	4	−1
0	−1	0

(d)

−1	−1	−1
−1	8	−1
−1	−1	−1

a	b
c	d

FIGURE 3.37
(a) Filter mask used to implement Eq. (3.6-6). (b) Mask used to implement an extension of this equation that includes the diagonal terms. (c) and (d) Two other implementations of the Laplacian found frequently in practice.

Image Enhancement in Spatial Domain

Sharpening Spatial Filters: **Laplacian Operator**

Image sharpening in the way of using the Laplacian:

$$g(x,y) = f(x,y) + c\left[\nabla^2 f(x,y)\right]$$

where,

$f(x,y)$ is input image,

$g(x,y)$ is sharpenend images,

$c = -1$ if $\nabla^2 f(x,y)$ corresponding to Fig. 3.37(a) or (b)

and $c = 1$ if either of the other two filters is used.

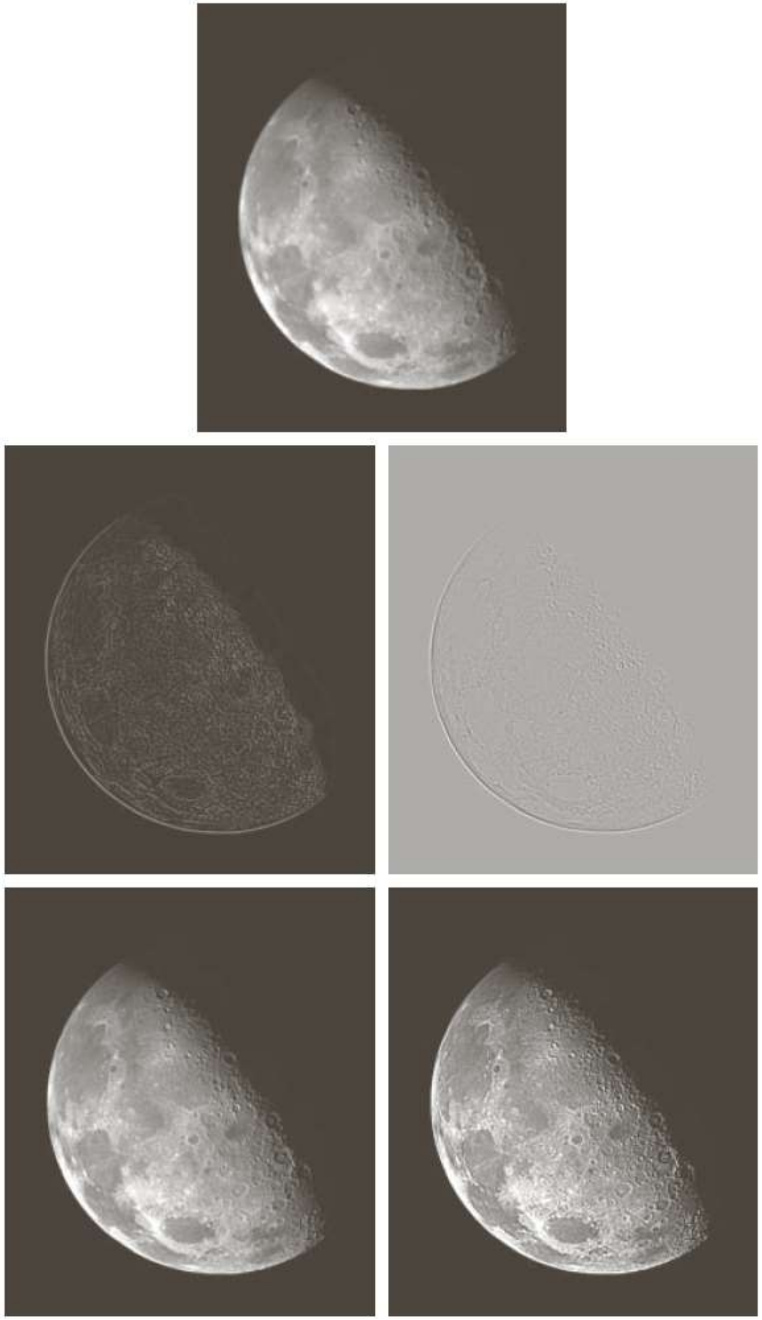


FIGURE 3.38 (a) Blurred image of the North Pole of the moon. (b) Laplacian without scaling. (c) Laplacian with scaling. (d) Image sharpened using the mask in Fig. 3.37(a). (e) Result of using the mask in Fig. 3.37(b). (Original image courtesy of NASA.)

Image Enhancement in Spatial Domain

Image Sharpening based on First-Order Derivatives

For function $f(x, y)$, the gradient of f at coordinates (x, y) is defined as

$$\nabla f \equiv \operatorname{grad}(f) \equiv \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \dfrac{\partial f}{\partial x} \\ \dfrac{\partial f}{\partial y} \end{bmatrix}$$

The *magnitude* of vector ∇f, denoted as $M(x, y)$

Gradient Image $M(x, y) = \operatorname{mag}(\nabla f) = \sqrt{g_x{}^2 + g_y{}^2}$

Image Enhancement in Spatial Domain

Image Sharpening based on First-Order Derivatives

The *magnitude* of vector ∇f, denoted as $M(x, y)$

$$M(x, y) = \text{mag}(\nabla f) = \sqrt{g_x{}^2 + g_y{}^2}$$

$$M(x, y) \approx |g_x| + |g_y|$$

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

$$M(x, y) = |z_8 - z_5| + |z_6 - z_5|$$

Image Sharpening based on First-Order Derivatives

(a)

z_1	z_2	z_3
z_4	z_5	z_6
z_7	z_8	z_9

(b)

−1	0
0	1

(c)

0	−1
1	0

(d)

−1	−2	−1
0	0	0
1	2	1

(e)

−1	0	1
−2	0	2
−1	0	1

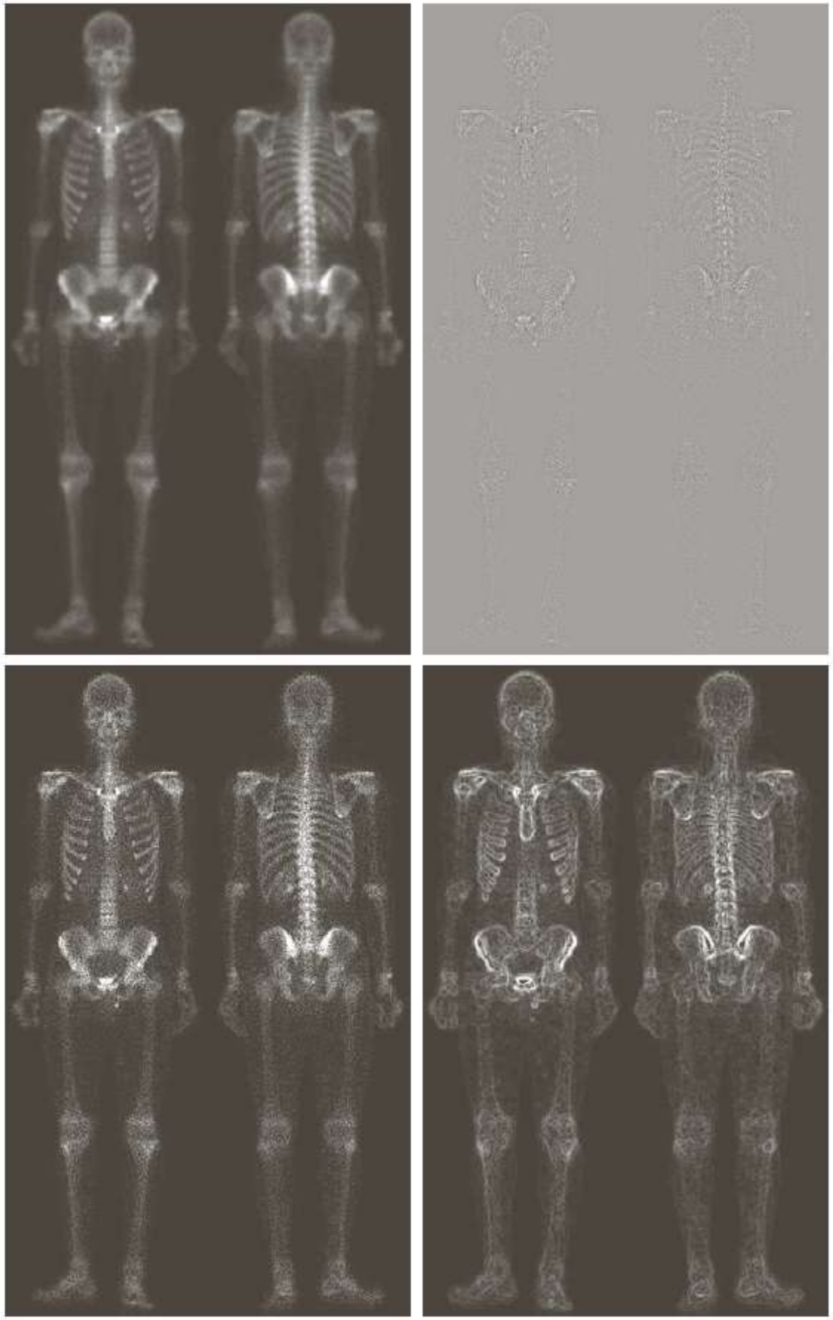
a
b c
d e

FIGURE 3.41
A 3×3 region of an image (the zs are intensity values).
(b)–(c) Roberts cross gradient operators.
(d)–(e) Sobel operators. All the mask coefficients sum to zero, as expected of a derivative operator.

Example

Combining Spatial Enhancement Methods

Goal:
Enhance the image by sharpening it and by bringing out more of the skeletal detail



a b
c d

FIGURE 3.43
(a) Image of whole body bone scan. (b) Laplacian of (a). (c) Sharpened image obtained by adding (a) and (b). (d) Sobel gradient of (a).

DIGITAL IMAGE PROCESSING

Unit-3: Image Enhancement in Frequency Domain

Dr. Piyush Joshi

Indian Institute of Information Technology
Sri City, Andhra Pradesh

Image Enhancement in Frequency Domain

Convert time domain signal into frequency domain signal

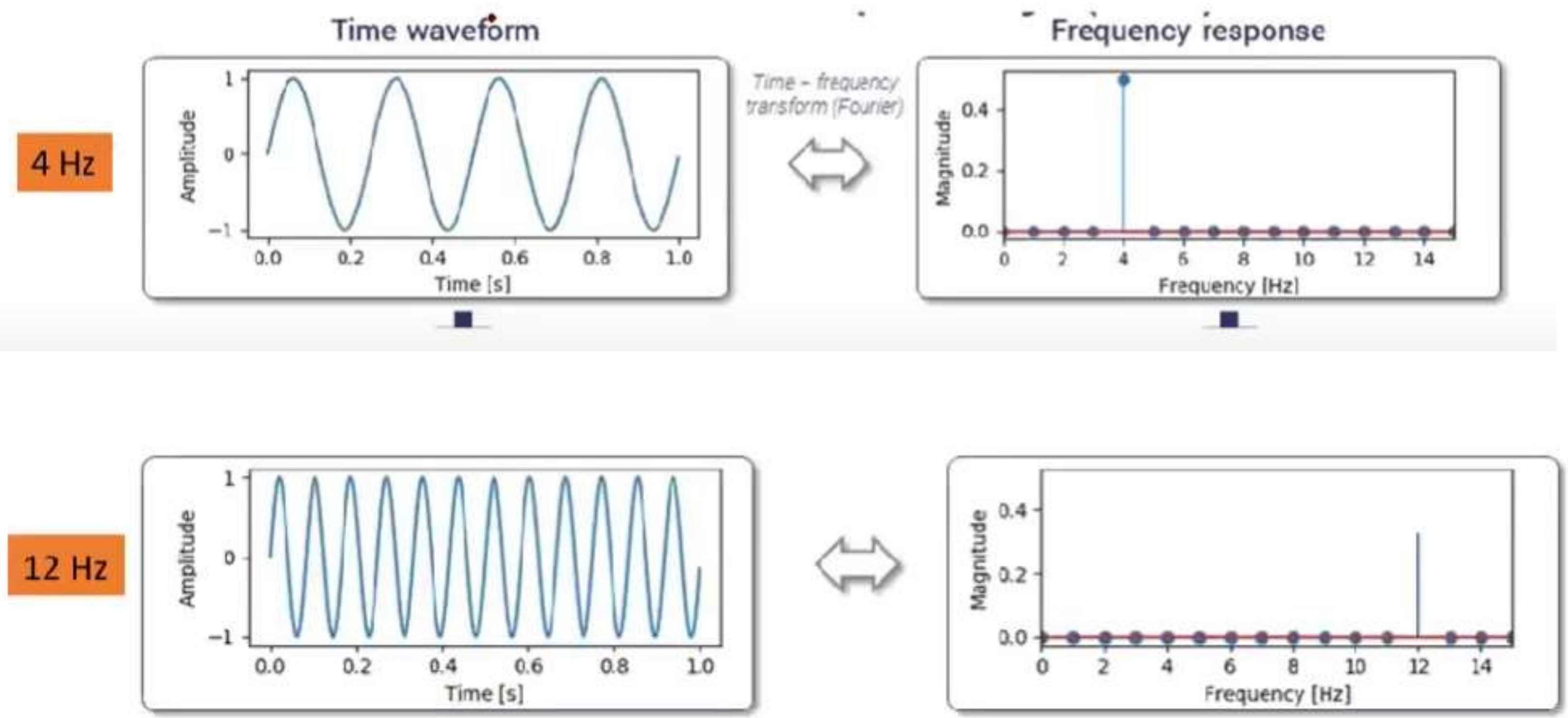
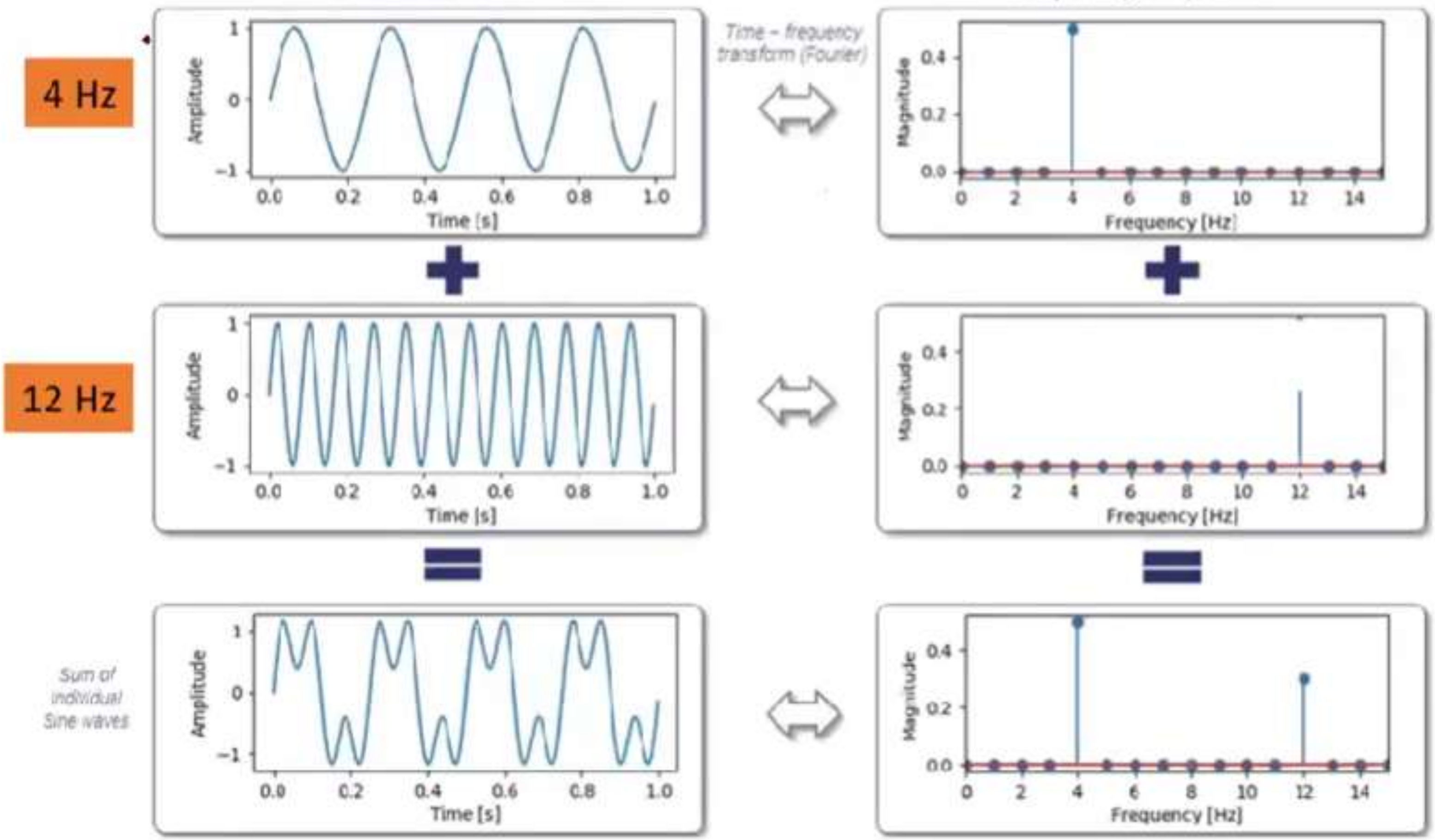


Image Enhancement in Frequency Domain



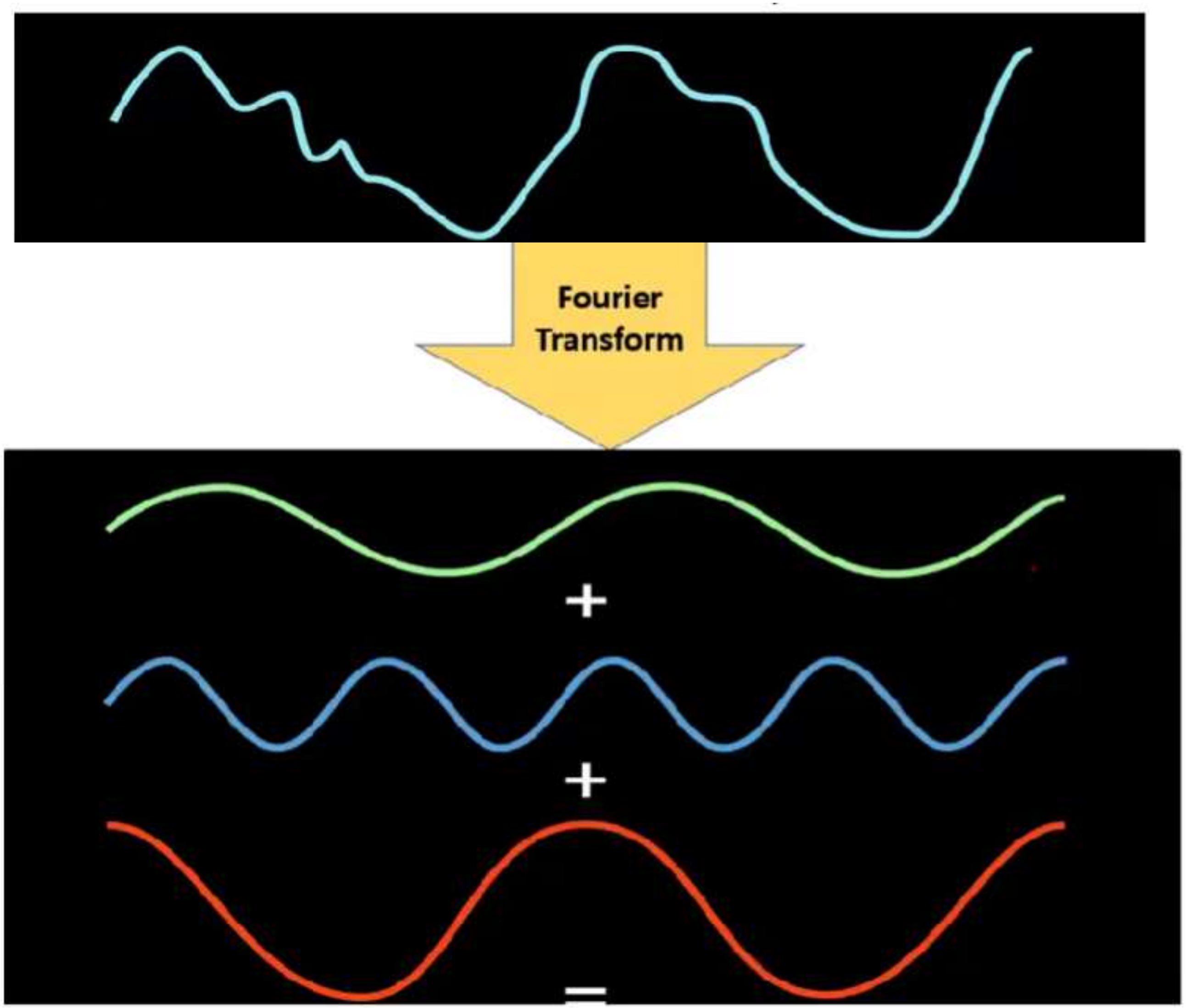

Fourier
Transform
+
+
=

Image Enhancement in Frequency Domain

Introduction

- Fourier Series

  Any periodic function can be expressed as the sum of sines and /or cosines of different frequencies, each multiplied by a different coefficients.

- Fourier Transform

  Any function that is not periodic can be expressed as the integral of sines and /or cosines multiplied by a weighing function.

Jean Baptiste Joseph Fourier, French mathematician and physicist (03/21/1768-05/16/1830)

Image Enhancement in Frequency Domain

Fourier Series

A function $f(t)$ of a continuous variable t that is periodic with period, T, can be expressed as the sum of sines and cosines multiplied by appropriate coefficients

$$f(t) = \sum_{n=-\infty}^{\infty} c_n e^{j\frac{2\pi n}{T}t}$$

$$e^{j\theta} = cos\theta + j\ sin\theta$$

$$e^{-j\theta} = cos\theta - j\ sin\theta$$

where

$$c_n = \frac{1}{T}\int_{-T/2}^{T/2} f(t)e^{-j\frac{2\pi n}{T}t}dt \qquad \text{for } n = 0, \pm 1, \pm 2, \ldots$$

Image Enhancement in Frequency Domain

1-D Fourier Transform: **Continuous Variable**

The *Fourier Transform* of a continous function $f(t)$

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

The *Inverse Fourier Transform* of $F(\mu)$

$$f(t) = \Im^{-1}\{F(\mu)\} = \int_{-\infty}^{\infty} F(\mu) e^{j2\pi\mu t} d\mu$$

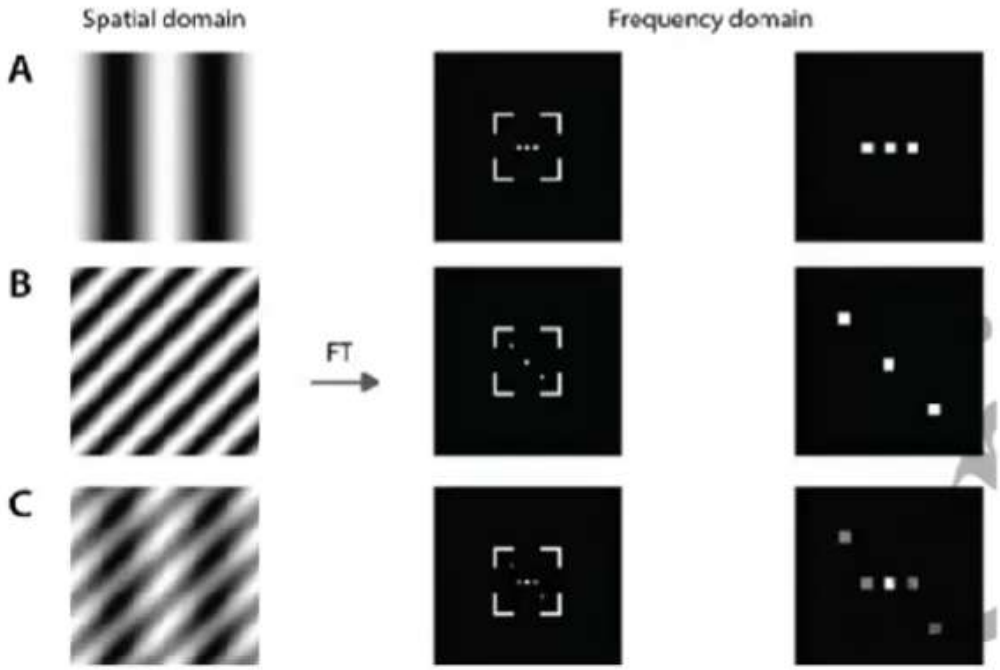
Image Enhancement in Frequency Domain

Fourier Transform of an Image

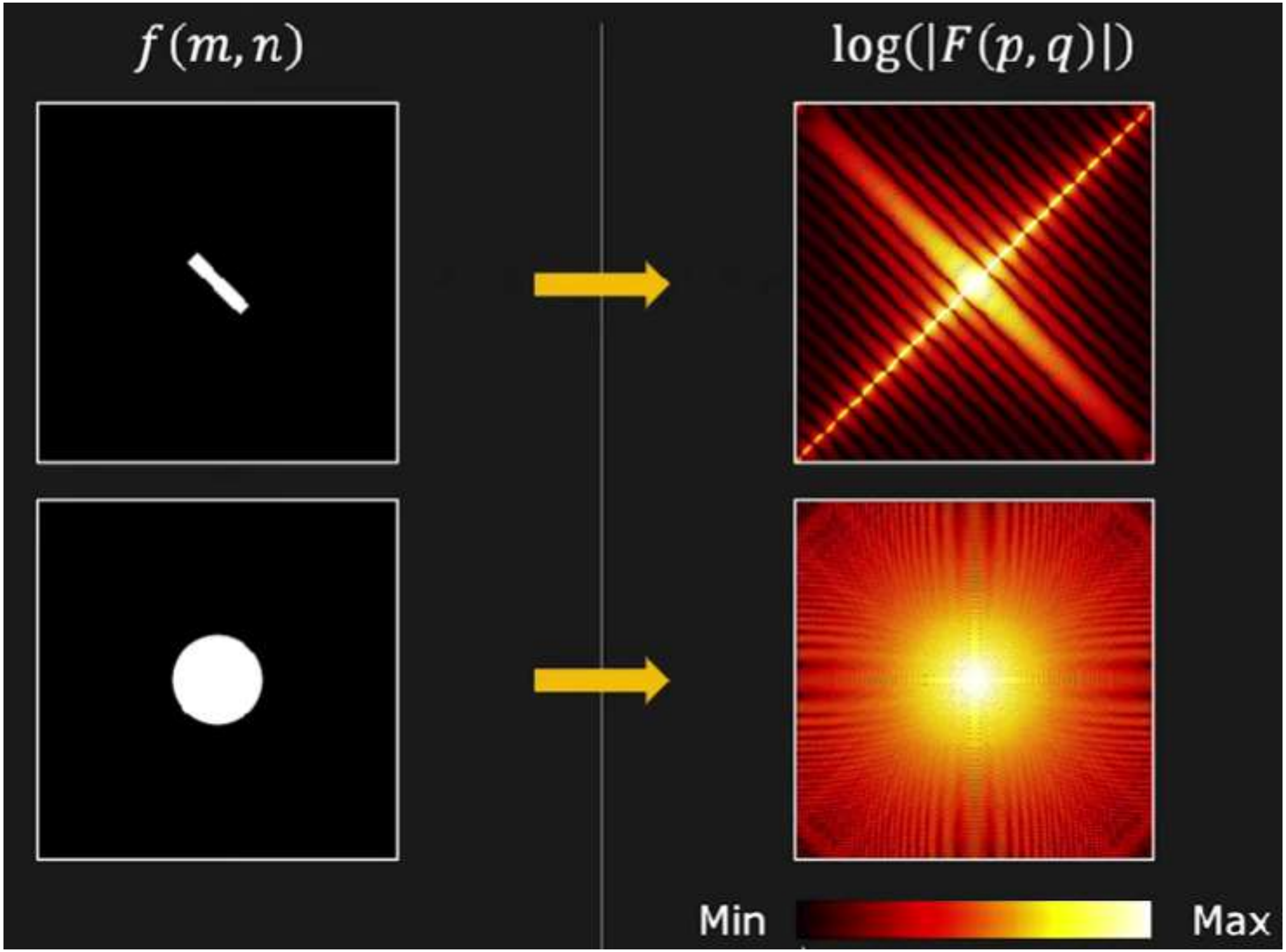
- The Fourier Transform is an important image processing tool which is used to decompose an image into its sine and cosine components.
- As we are only concerned with digital images, we will restrict this discussion to the Discrete Fourier Transform (DFT).
- For a square image of size N×N, the two-dimensional DFT is given by:

Fourier Transform of an image

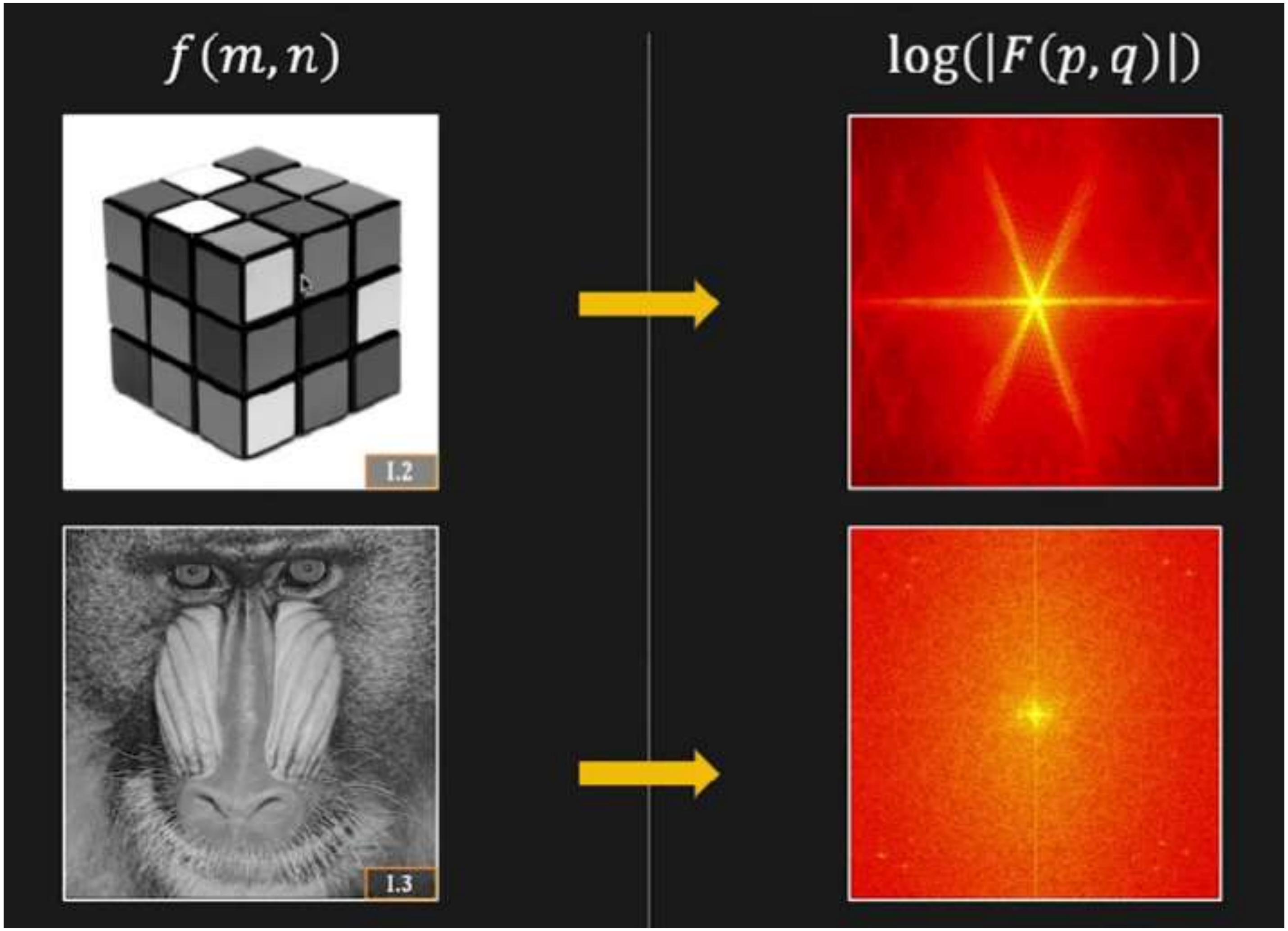
The response of the Fourier Transform to periodic patterns in the spatial domain images can be seen very easily in the following artificial images



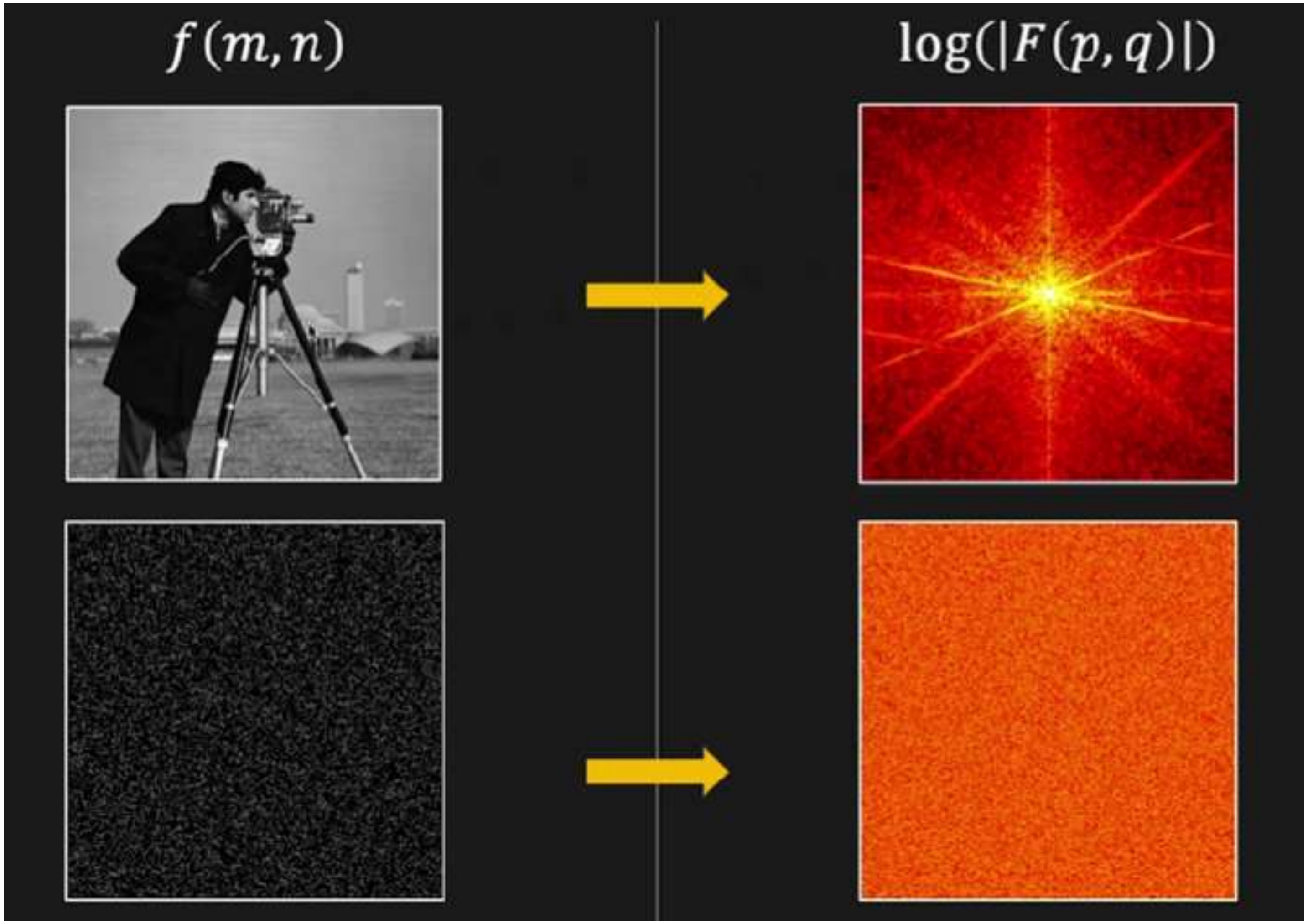
Fourier Transform of an image



$f(m,n)$
$\log(|F(p,q)|)$
Min
Max

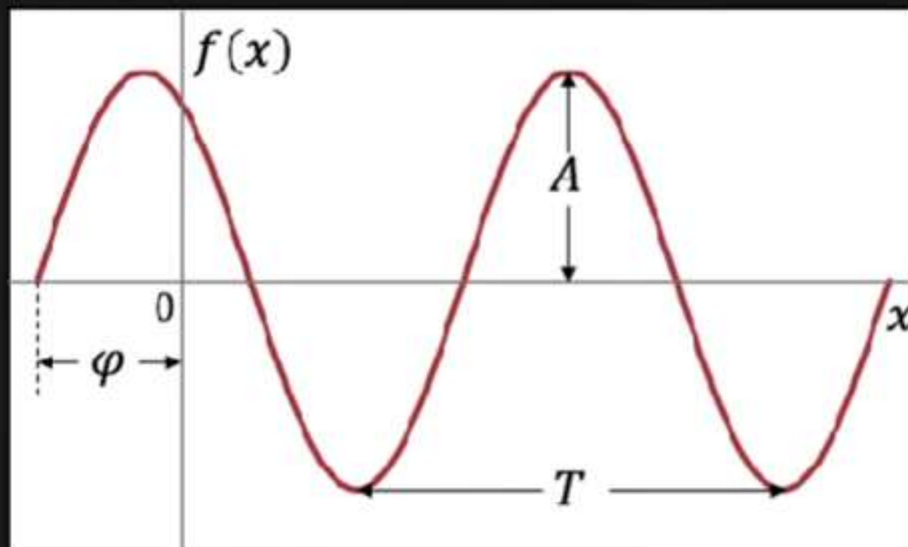


$f(m,n)$
$\log(|F(p,q)|)$
1.2
1.3


$f(m,n)$
$\log(|F(p,q)|)$

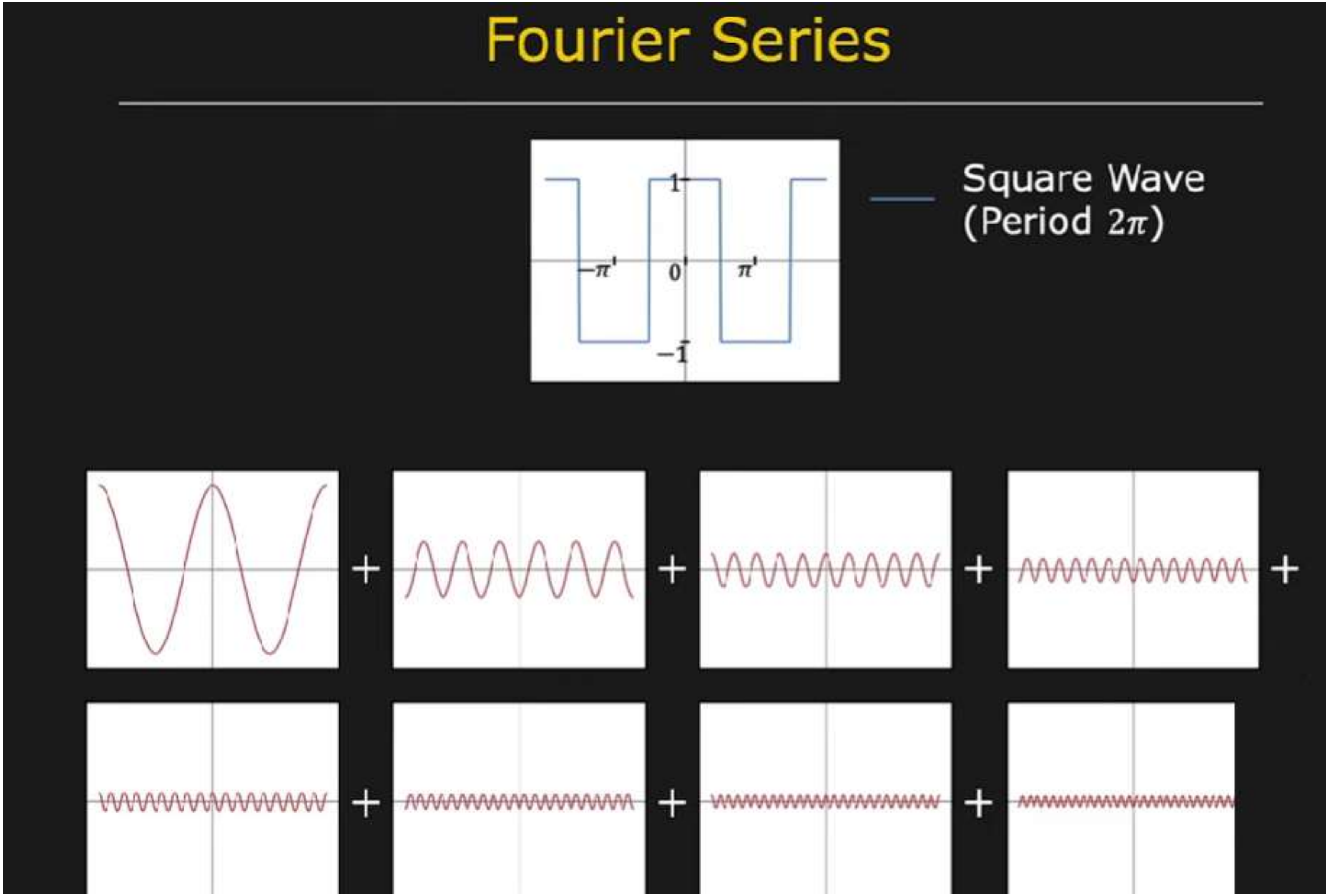
Sinusoid

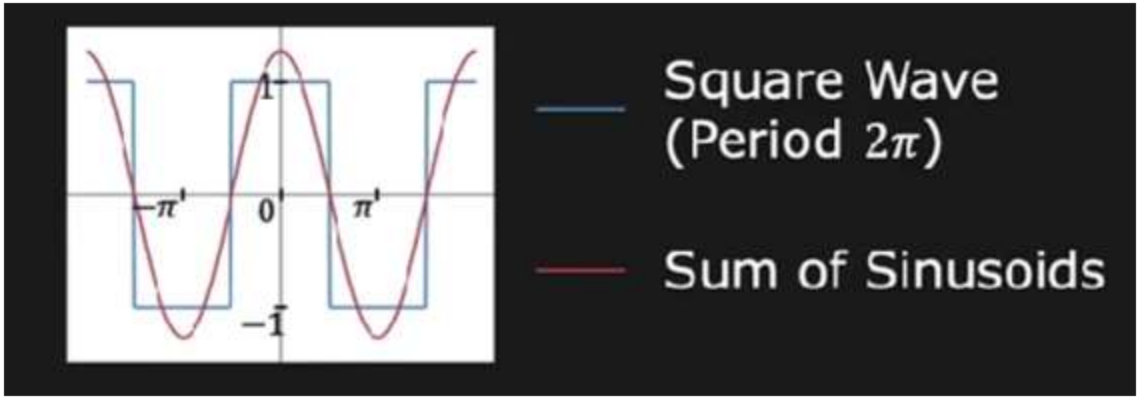
$$f(x) = A\sin(2\pi ux + \varphi)$$



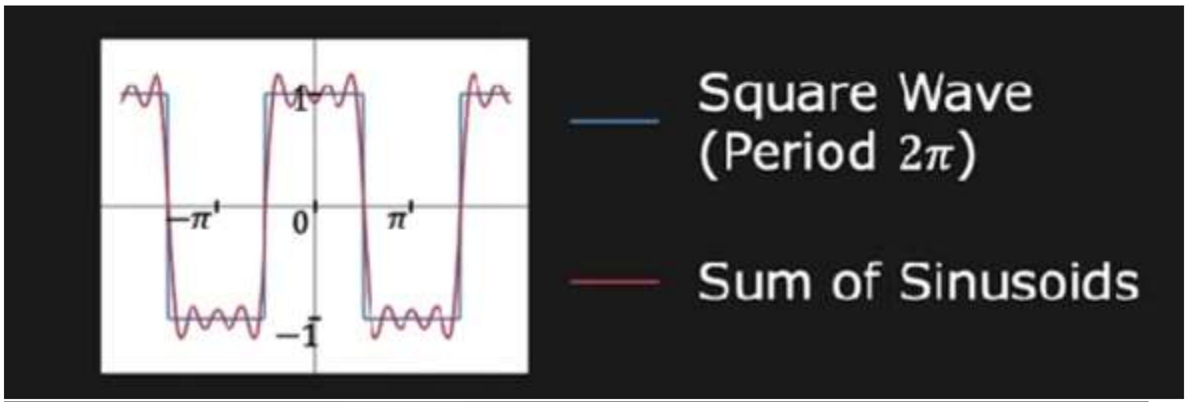
A: Amplitude $\quad$ T: Period

φ: Phase $\quad$ u: Frequency $(1/T)$

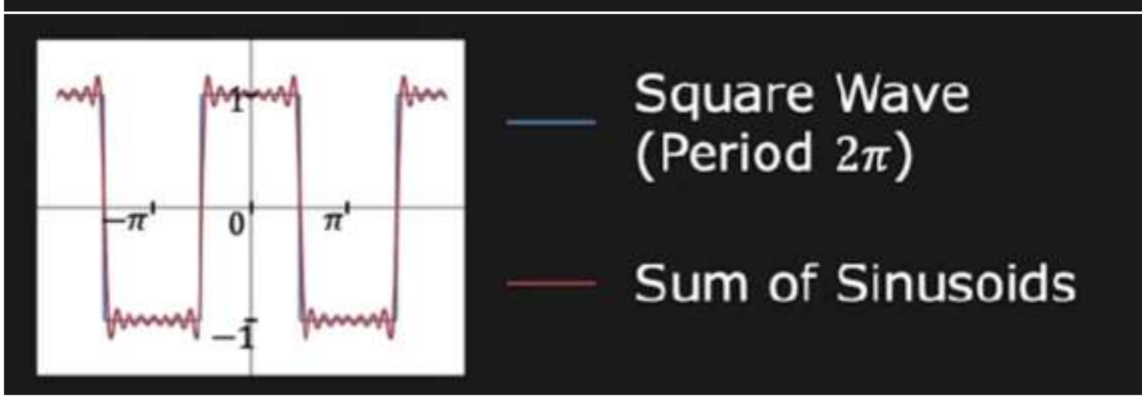

Fourier Series
1
−π
0
π
−1
Square Wave
(Period 2π)
+
+
+
+
+
+
+



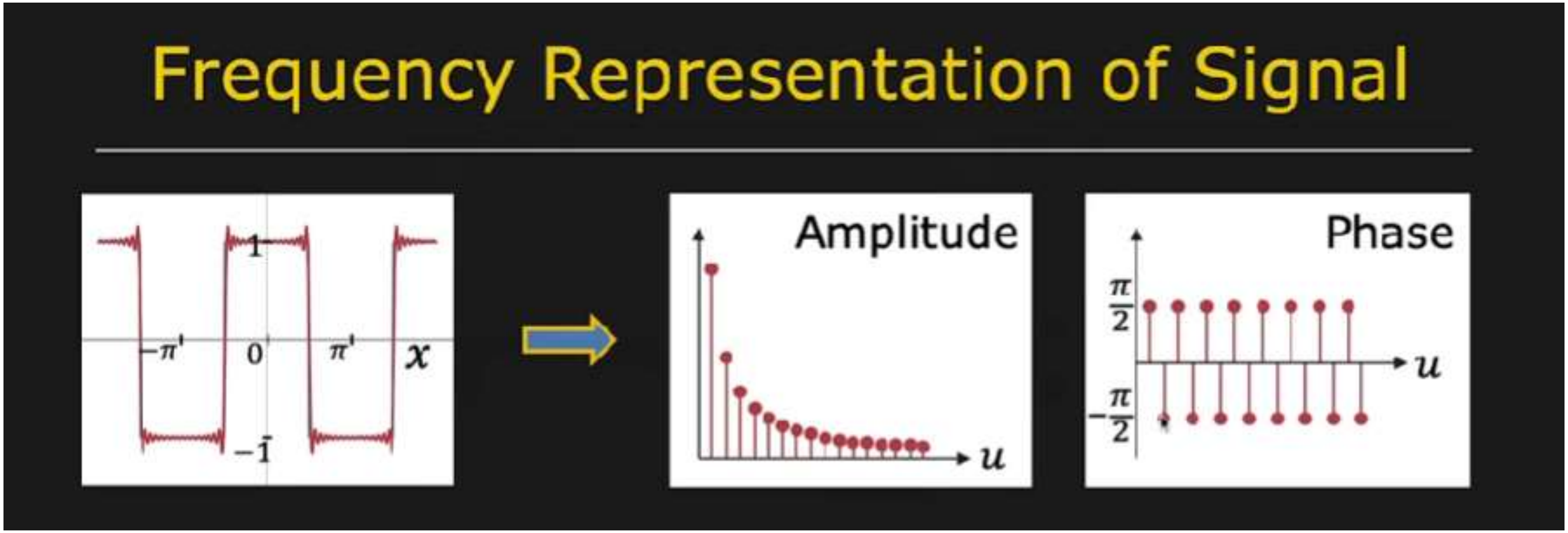
One sinusoids



Sum of first four sinusoids



Sum of eight sinusoids


Frequency Representation of Signal
1
$-\pi$
0
π
x
-1
Amplitude
u
Phase
$\frac{\pi}{2}$
$-\frac{\pi}{2}$
u

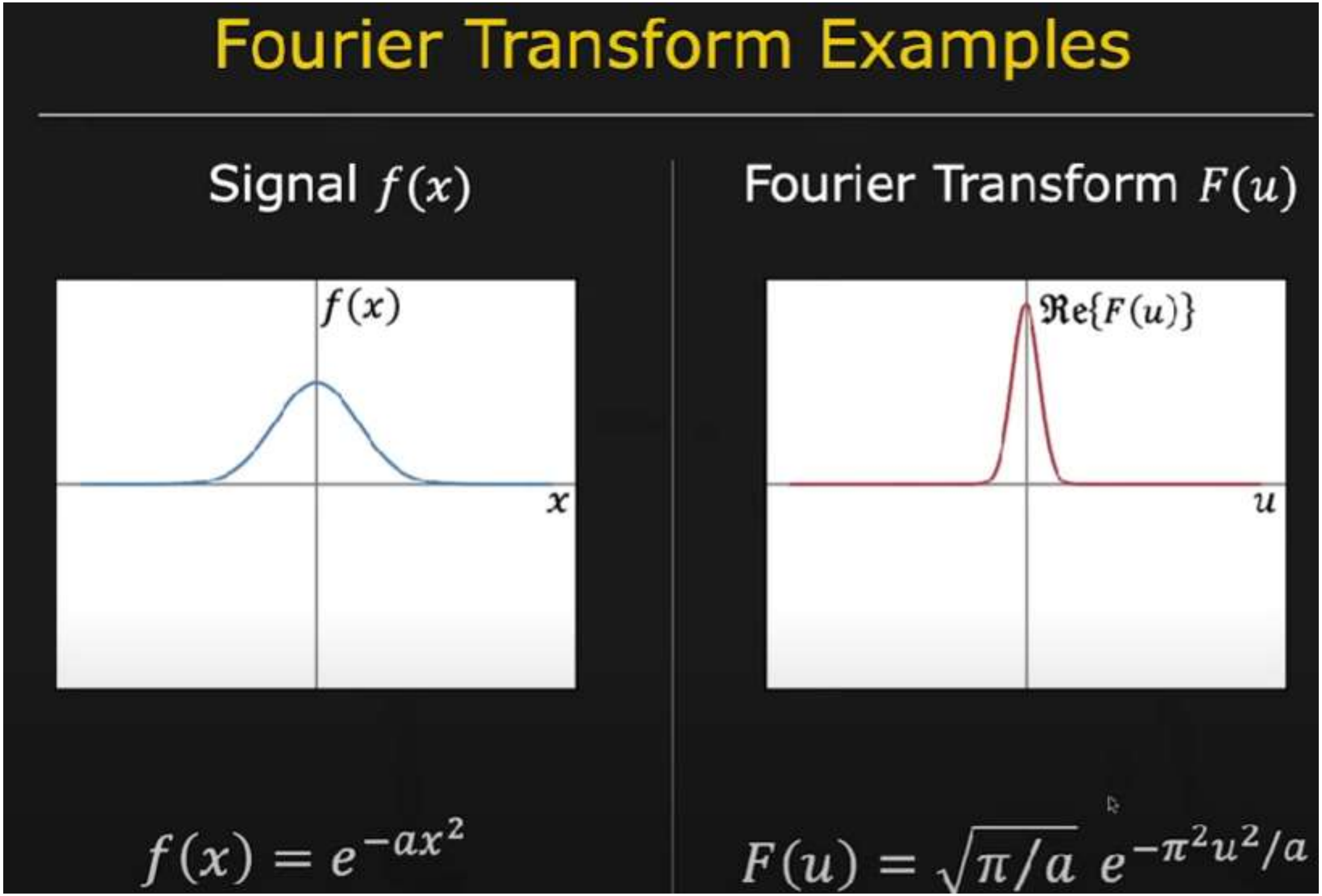

Fourier Transform Examples
Signal $f(x)$
Fourier Transform $F(u)$
$f(x)$
x
$\Re e\{F(u)\}$
u
$f(x) = e^{-ax^2}$
$F(u) = \sqrt{\pi/a}\, e^{-\pi^2 u^2/a}$

Image Enhancement in Frequency Domain

1-D Discrete Fourier Transform

$$F(\mu) = \sum_{x=0}^{M-1} f(x) e^{-j2\pi\mu x/M}, \qquad \mu = 0, 1, \ldots, M-1$$

$$f(x) = \frac{1}{M} \sum_{\mu=0}^{M-1} F(\mu) e^{j2\pi\mu x/M}, \qquad x = 0, 1, 2, \ldots, M-1$$

1. The domain (values of μ) over which the value of $F(\mu)$ range is called the Frequency Domain.
2. Each of the *M* terms of $F(\mu)$ is called a Frequency Component of the transform.

Image Enhancement in Frequency Domain

2-D Fourier Transform: **Continuous Variable**

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

and

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

Image Enhancement in Frequency Domain

2-D Discrete Fourier Transform and Its Inverse

DFT:

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

$$\mu=0,1,2,\ldots,M-1;\nu=0,1,2,\ldots,N-1;$$

$f(x,y)$ is a digital image of size $\mathrm{M}\times\mathrm{N}$.

IDFT:

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

Fourier Transform is Complex!

$F(u)$ holds the Amplitude and Phase of the sinusoid of frequency u.

$$\boxed{F(u) = \int_{-\infty}^{\infty} f(x) e^{-i2\pi ux} dx}$$

$$F(u) = \Re e\{F(u)\} + i\,\Im m\{F(u)\}$$

Amplitude: $A(u) = \sqrt{\Re e\{F(u)\}^2 + \Im m\{F(u)\}^2}$

Phase: $\varphi(u) = \operatorname{atan2}(\Im m\{F(u)\}, \Re e\{F(u)\})$

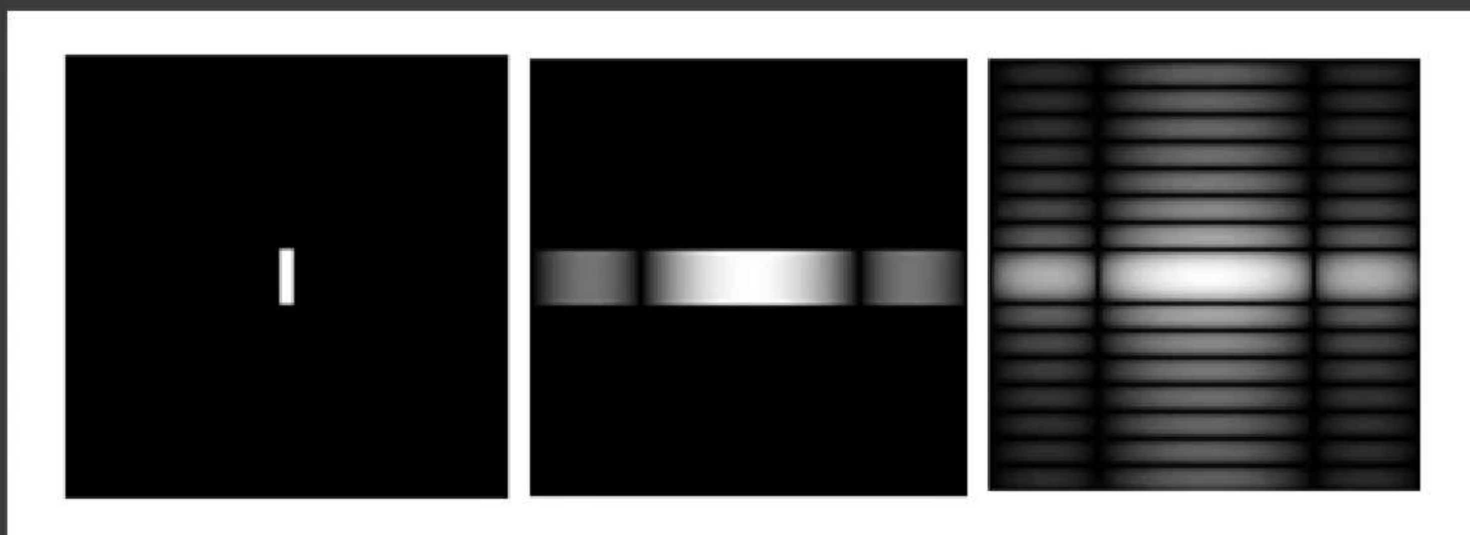
Properties of Fourier Transform

Property	Spatial Domain	Frequency Domain
Linearity	$\alpha f_1(x) + \beta f_2(x)$	$\alpha F_1(u) + \beta F_2(u)$
Scaling	$f(ax)$	$\frac{1}{\lvert a \rvert} F\left(\frac{u}{a}\right)$
Shifting	$f(x-a)$	$e^{-i2\pi u a} F(u)$

Properties of 2-D DFT: **Separability**

The 2D DFT $F(u,v)$ can be obtained by

1. taking the 1D DFT of every row of image $f(x,y)$-> $F(x,v)$,
2. taking the 1D DFT of every column of $F(x,v)$ -> F(u,v)



(a) $f(x,y)$ (b) $F(x,v)$ (c) $F(u,v)$

Image Enhancement in Frequency Domain

Properties of 2-D DFT: **Shift**

$$f(x,y)e^{j2\pi(\mu_0 x/M+\nu_0 y/N)} \Leftrightarrow F(\mu-\mu_0,\nu-\nu_0)$$

and

$$f(x-x_0,y-y_0) \Leftrightarrow F(\mu,\nu)e^{-j2\pi(\mu x_0/M+\nu y_0/N)}$$

Image Enhancement in Frequency Domain

Properties of 2-D DFT: **Convolution**

- Let $F(u,v)$ and $H(u,v)$ denote the Fourier transforms of $f(x,y)$ and $h(x,y)$, then

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

Convolution

Convolution of two functions $f(x)$ and $h(x)$

$$\boxed{g(x) = f(x) * h(x) = \int_{-\infty}^{\infty} f(\tau)h(x-\tau)\,d\tau}$$

Convolution and Fourier Transform

Convolution: $g(x) = f(x) * h(x) = \int_{-\infty}^{\infty} f(\tau)h(x-\tau)\,d\tau$

Fourier Transform of $g(x)$:

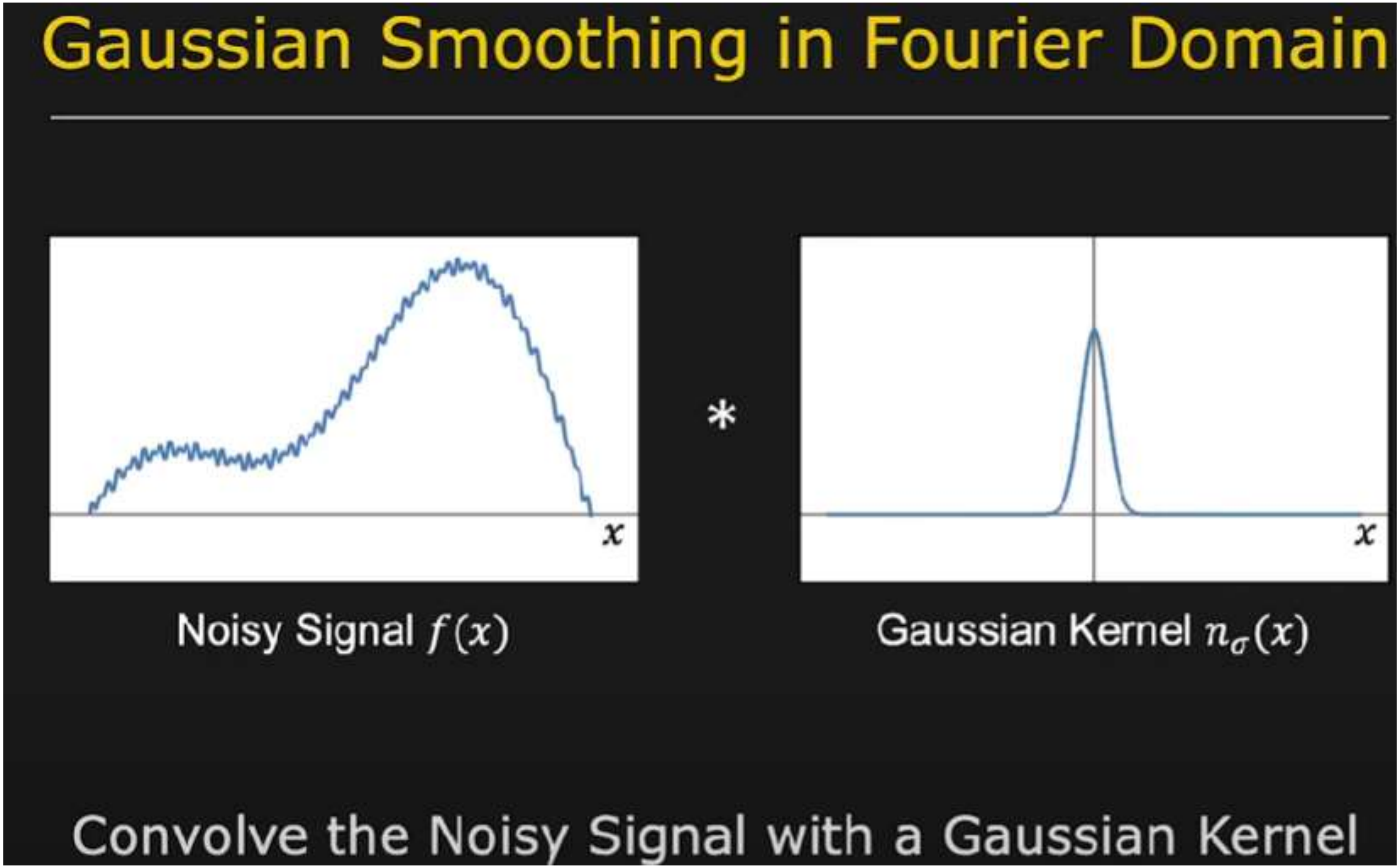
$$G(u) = \int_{-\infty}^{\infty} g(x)e^{-i2\pi ux}dx$$

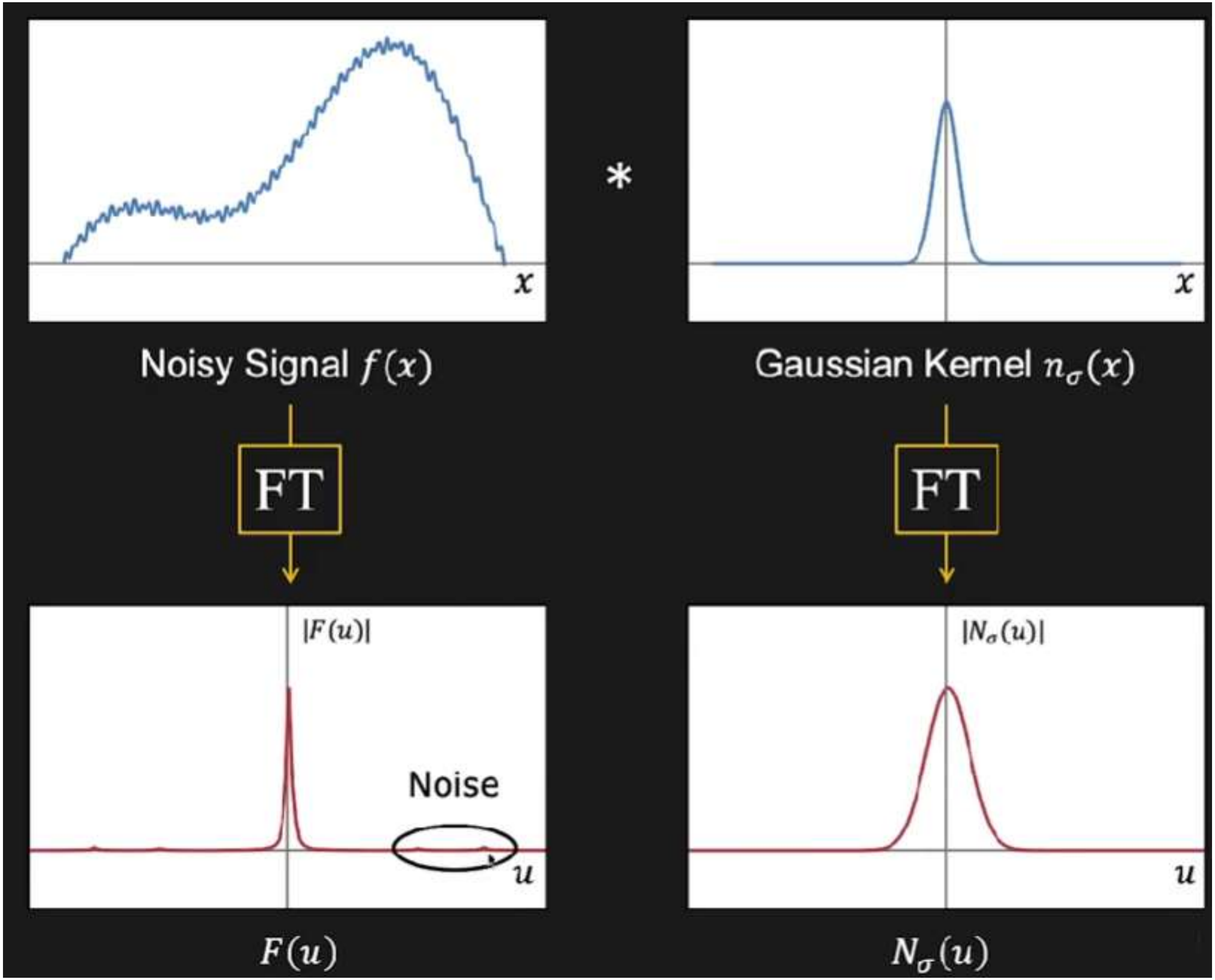
$$G(u) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(\tau)h(x-\tau)e^{-i2\pi ux}d\tau dx$$

$$G(u) = \int_{-\infty}^{\infty} f(\tau)e^{-i2\pi u\tau}d\tau \int_{-\infty}^{\infty} h(x-\tau)e^{-i2\pi u(x-\tau)}\,dx$$

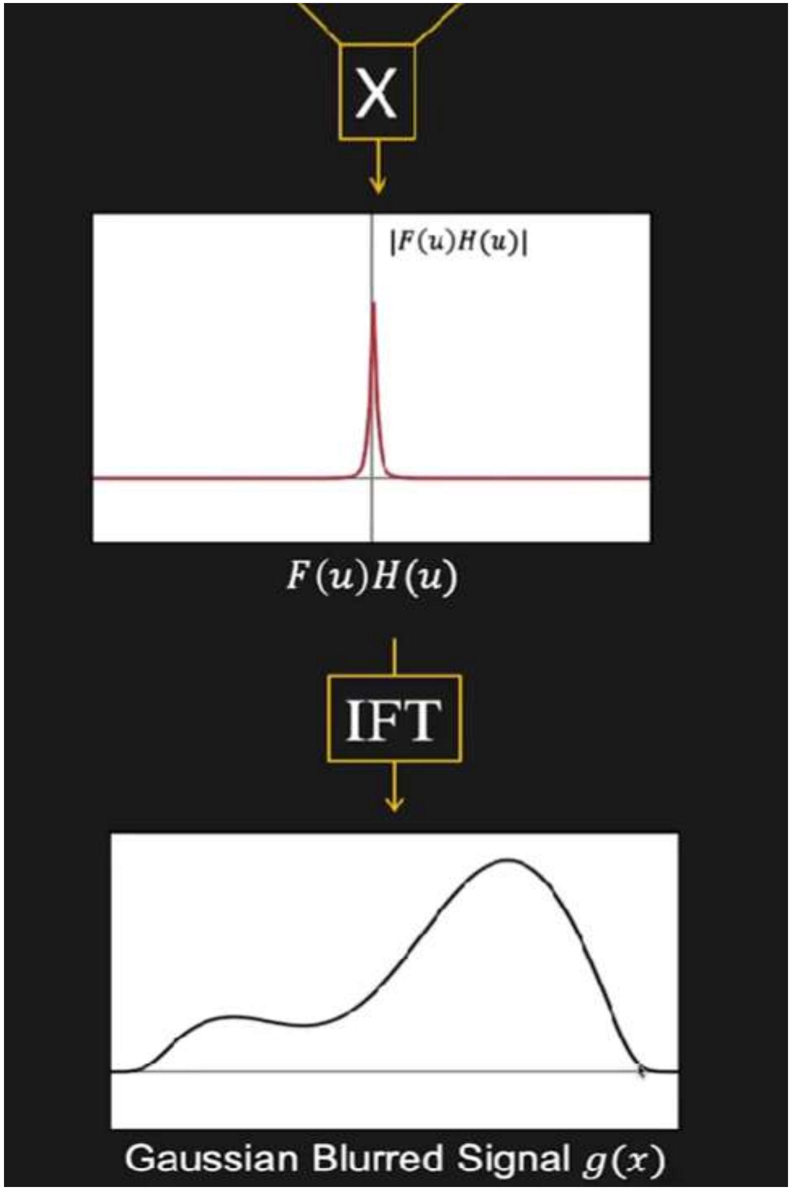
Convolution and Fourier Transform

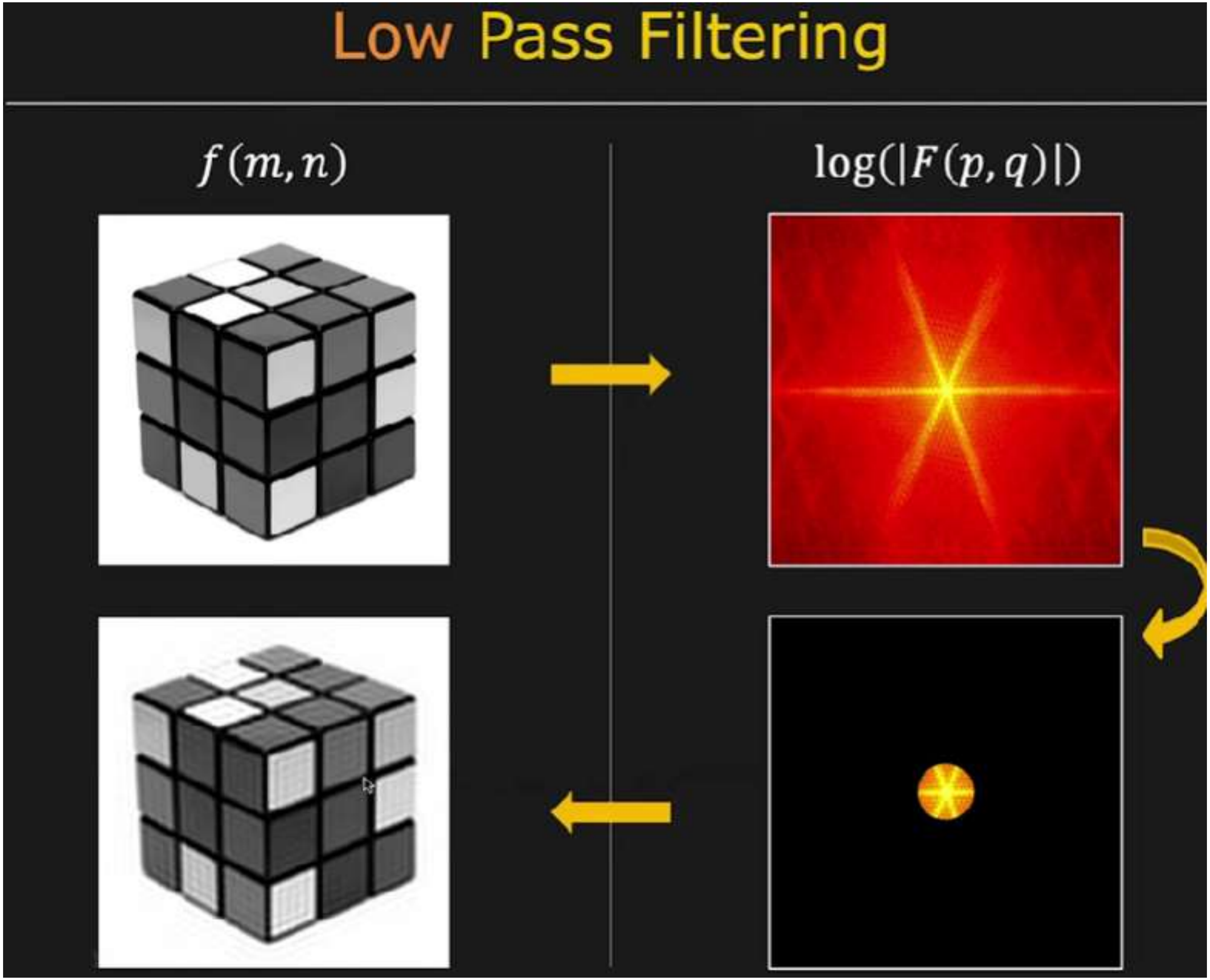
Spatial Domain		Frequency Domain
$g(x) = f(x) * h(x)$ Convolution	$\longleftrightarrow$	$G(u) = F(u)\,H(u)$ Multiplication
$g(x) = f(x)\,h(x)$ Multiplication	$\longleftrightarrow$	$G(u) = F(u) * H(u)$ Convolution

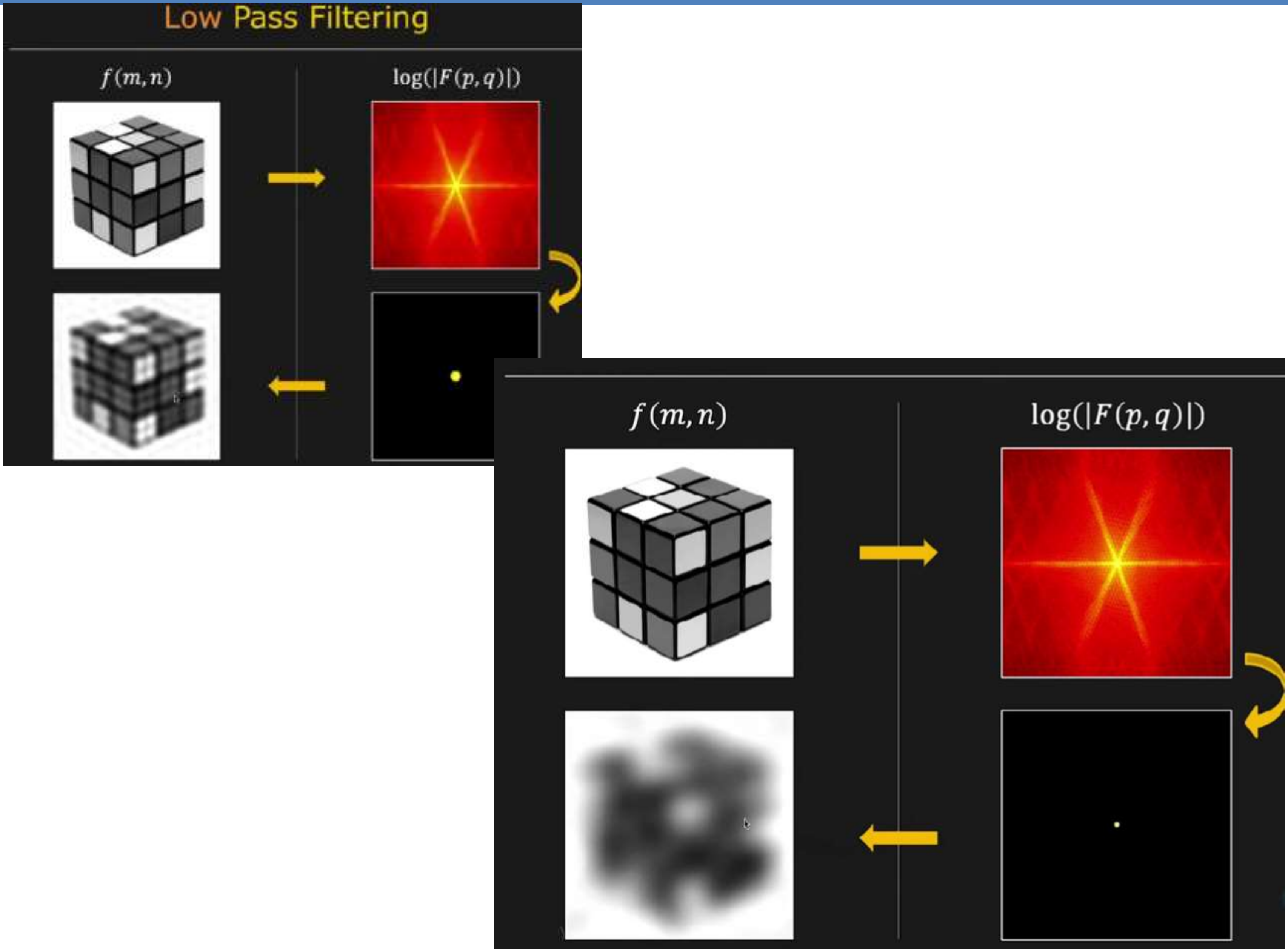

Gaussian Smoothing in Fourier Domain
x
*
x
Noisy Signal $f(x)$
Gaussian Kernel $n_\sigma(x)$
Convolve the Noisy Signal with a Gaussian Kernel

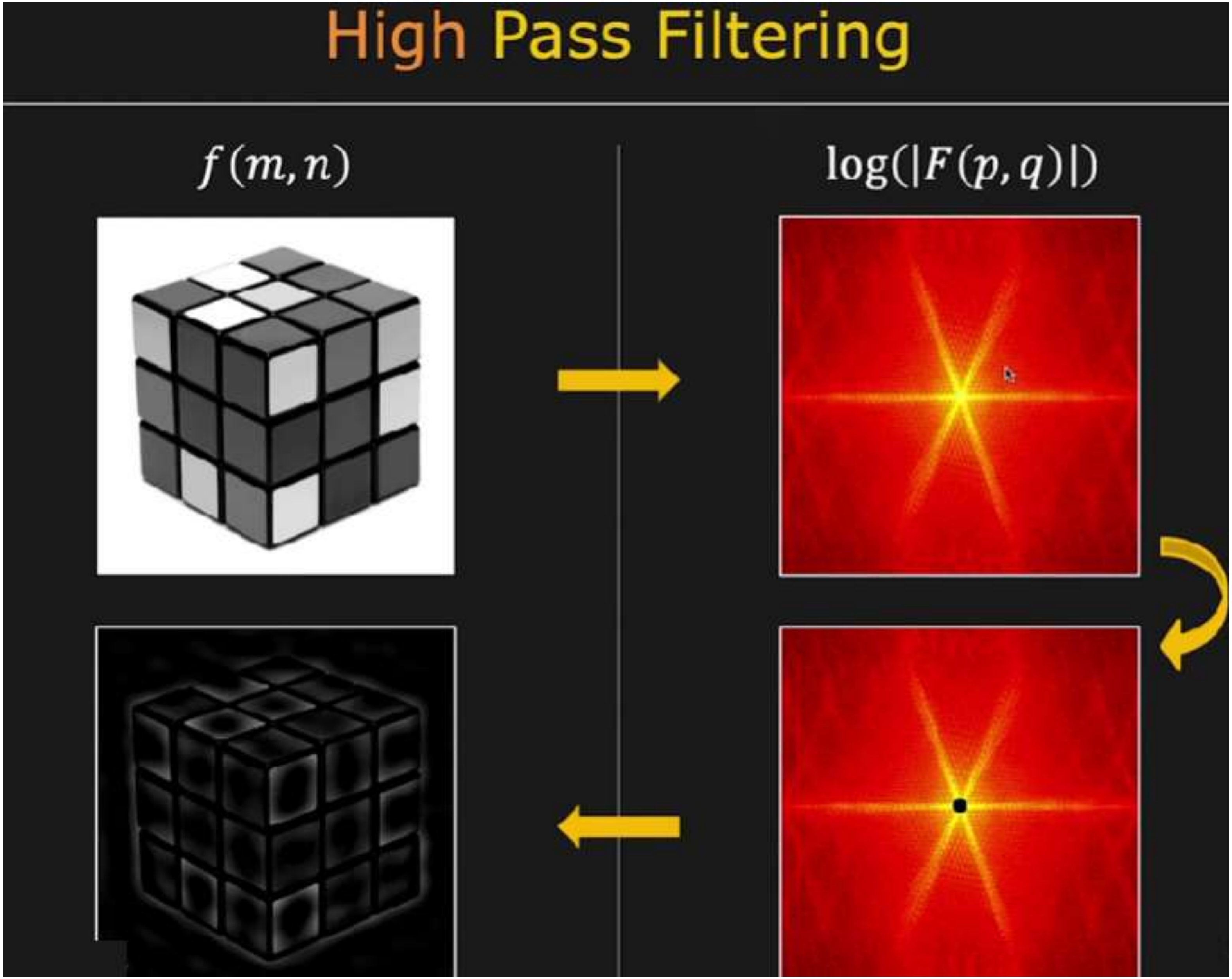


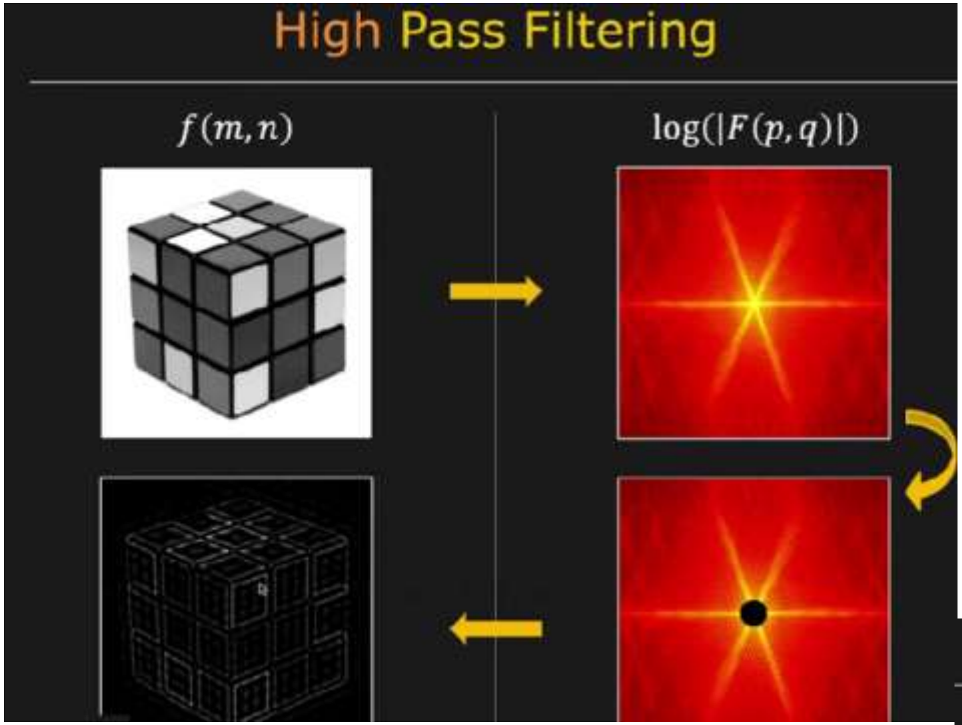
x
Noisy Signal $f(x)$
*
x
Gaussian Kernel $n_\sigma(x)$
FT
FT
$|F(u)|$
Noise
u
$F(u)$
$|N_\sigma(u)|$
u
$N_\sigma(u)$

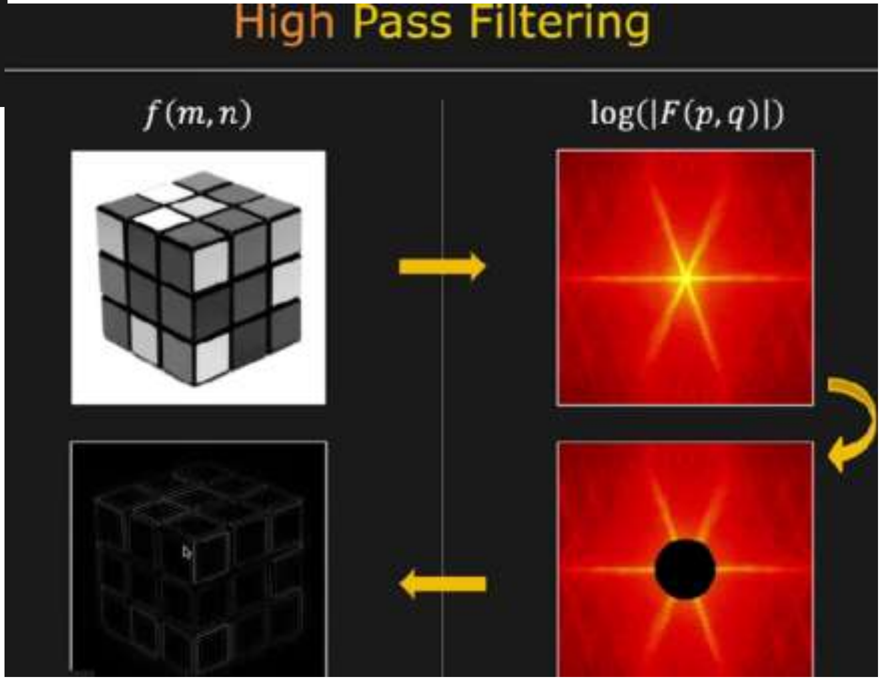

X
|F(u)H(u)|
F(u)H(u)
IFT
Gaussian Blurred Signal g(x)

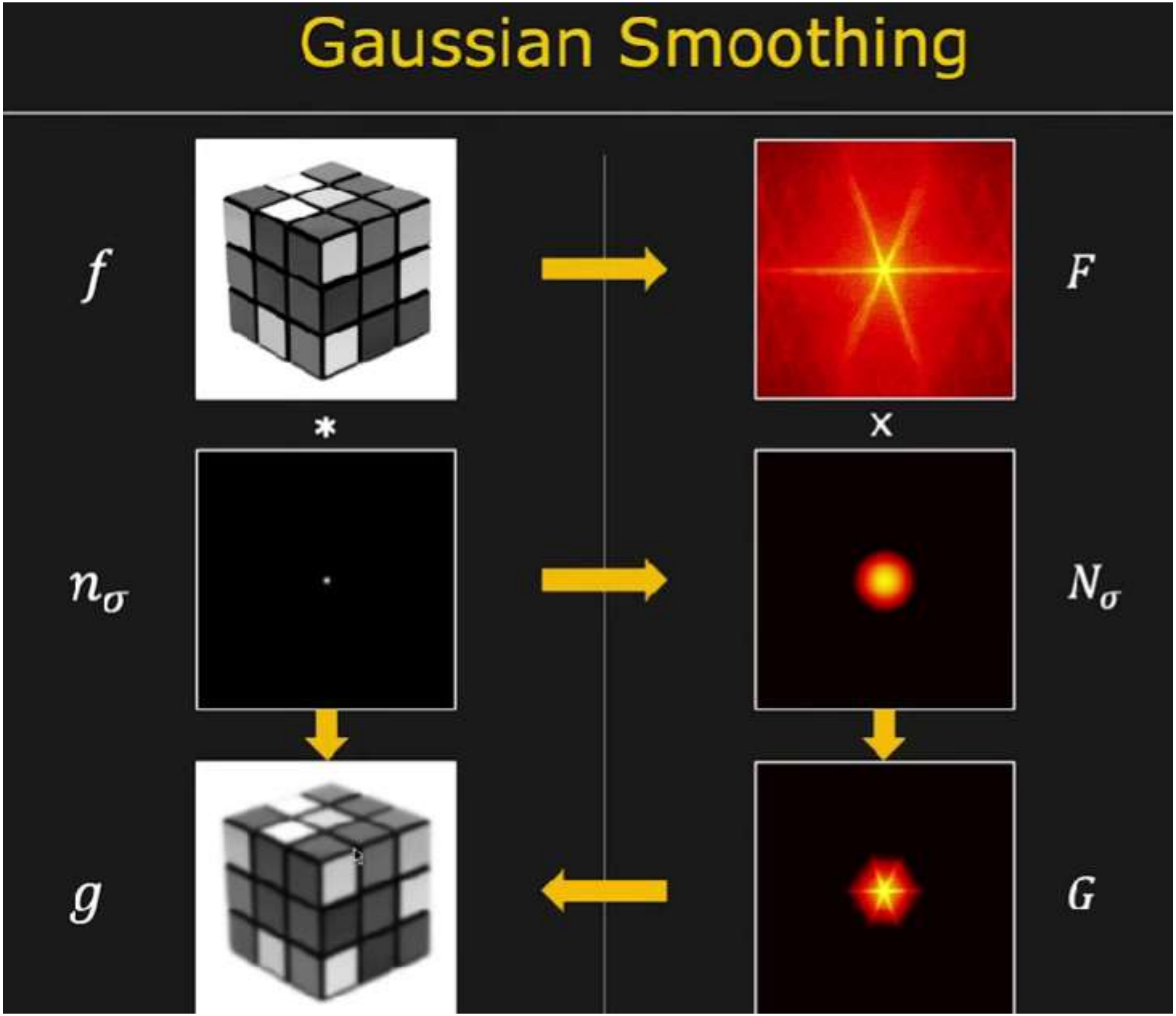

Low Pass Filtering
$f(m,n)$
$\log(|F(p,q)|)$

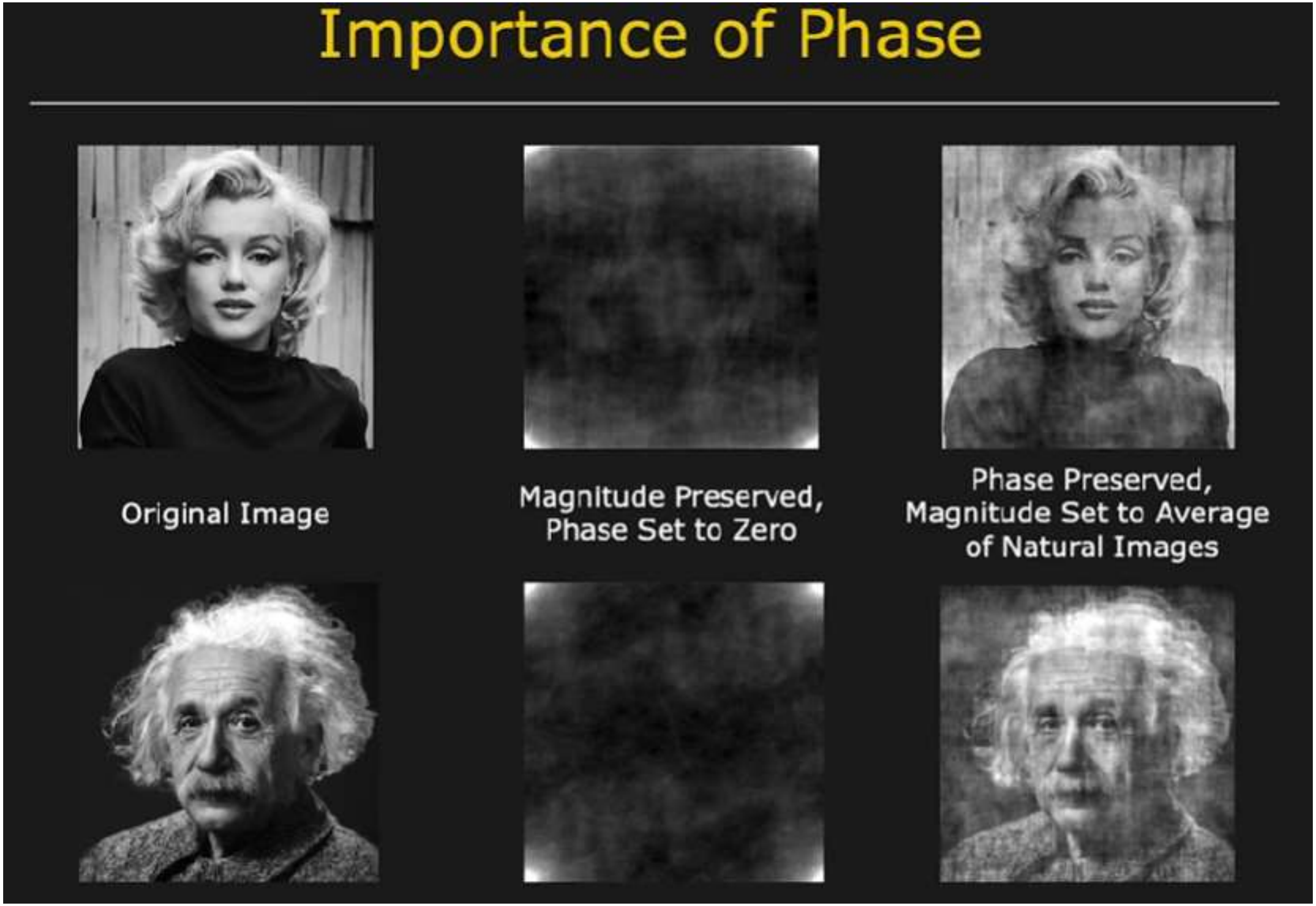

Low Pass Filtering
$f(m,n)$
$\log(|F(p,q)|)$
$f(m,n)$
$\log(|F(p,q)|)$


High Pass Filtering
$f(m,n)$
$\log(|F(p,q)|)$


High Pass Filtering
$f(m,n)$
$\log(|F(p,q)|)$

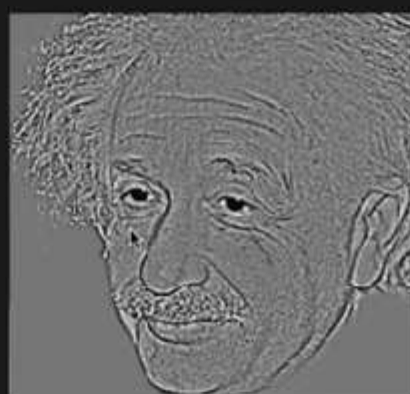

High Pass Filtering
$f(m,n)$
$\log(|F(p,q)|)$


Gaussian Smoothing
f
F
$*$
x
n_σ
N_σ
g
G

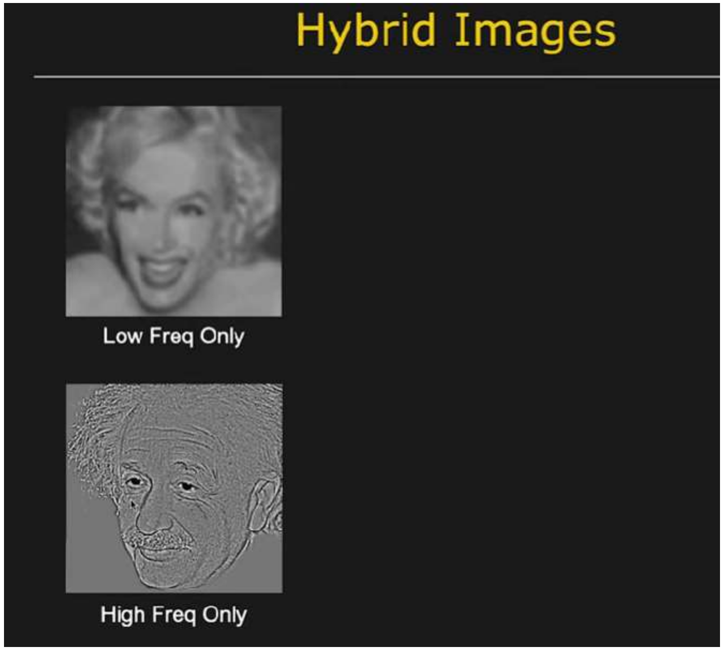

Importance of Phase
Original Image
Magnitude Preserved, Phase Set to Zero
Phase Preserved, Magnitude Set to Average of Natural Images

Hybrid Images

Low Freq Only



High Freq Only


Hybrid Images
Low Freq Only
High Freq Only

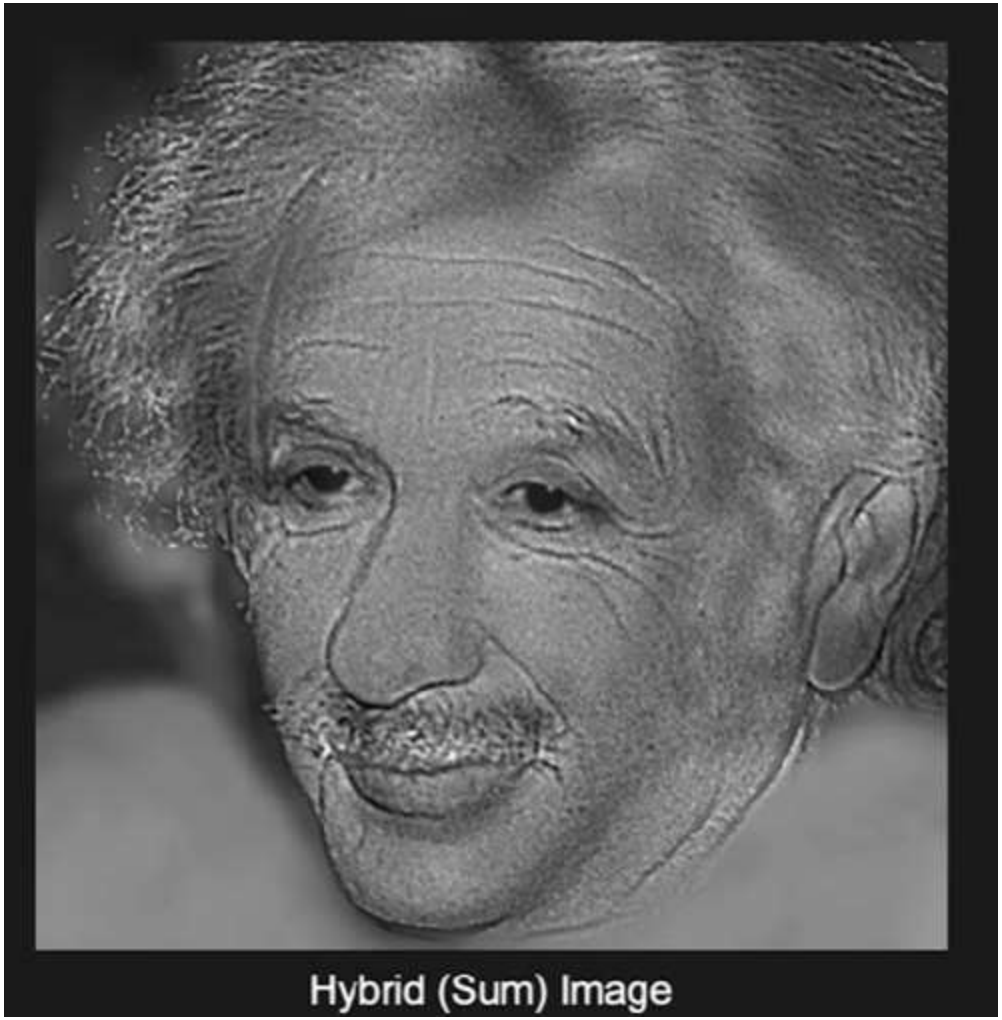

Hybrid (Sum) Image

Image Enhancement in Frequency Domain

Filtering in Frequency Domain: **Properties**

- Low frequencies in the Fourier transform are responsible for the general gray-level appearance of an image over smooth regions.
- High frequencies are responsible for detail, such as edges and noise.
- A filter that attenuates high frequencies while "passing" the low frequencies is called *lowpass filter.*
- A filter that attenuates low frequencies while "passing" the high frequencies is called *highpass filter.*

Image Enhancement in Frequency Domain

Smoothing Filters

- Edges and other sharp transitions (such as noise) contribute significantly to the high frequency content of Fourier transform.
- Smoothing (blurring) is achieved in frequency domain by attenuating a specified range of high frequency components.
- Three types of lowpass filters: Ideal, Butterworth and Gaussian filters
- These filters cover the range from very sharp (ideal) to very smooth (Gaussian) filter functions.

Image Enhancement in Frequency Domain

Smoothing Filters: **Ideal Lowpass Filters (ILPF)**

Ideal Lowpass Filters (ILPF)

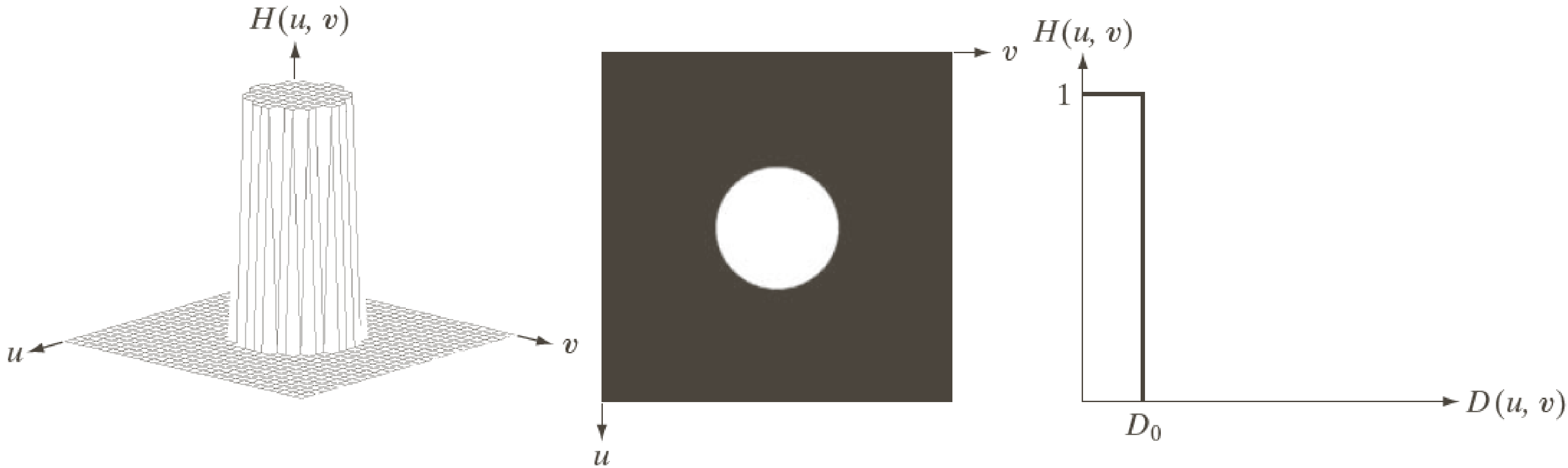
$$H(u,v) = \begin{cases} 1 & \text{if } D(u,v) \le D_0 \\ 0 & \text{if } D(u,v) > D_0 \end{cases}$$

D_0 is a positive constant and $D(u,v)$ is the distance between a point (u,v) in the frequency domain and the center of the frequency rectangle

$$D(u,v) = \left[(u - P/2)^2 + (v - Q/2)^2 \right]^{1/2}$$

Image Enhancement in Frequency Domain

Smoothing Filters: ILPF

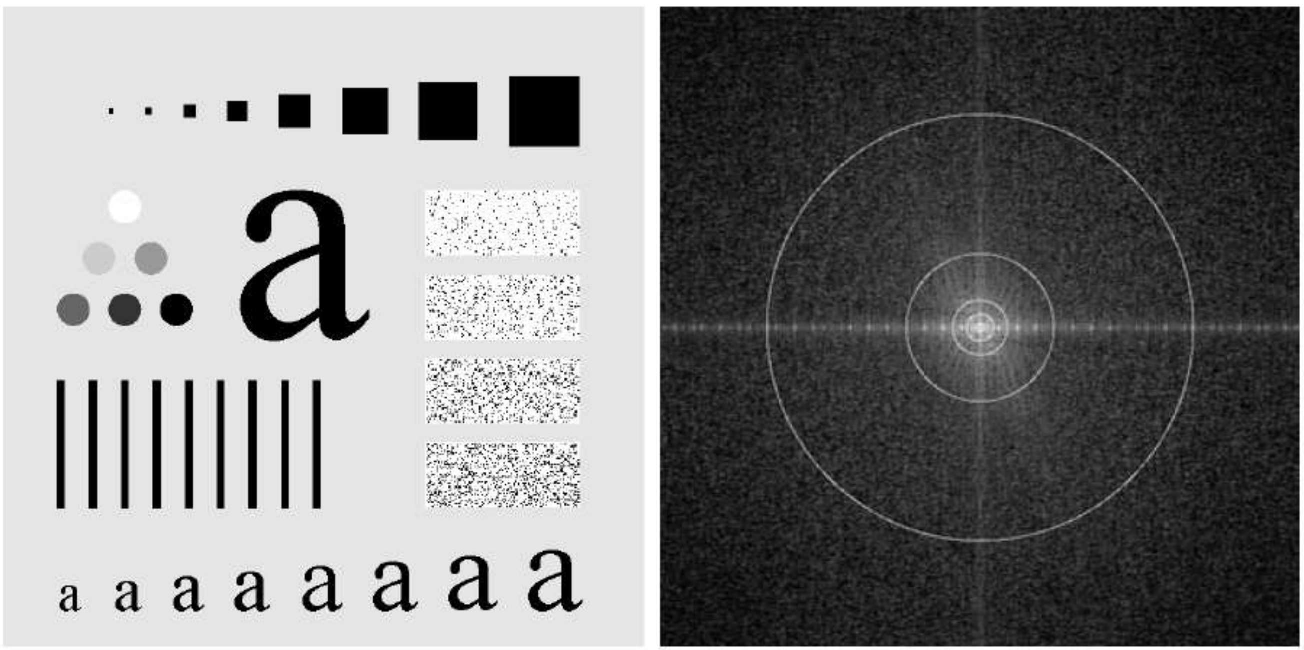


a b c

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

Image Enhancement in Frequency Domain

Smoothing Filters: **ILPF**



a b

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

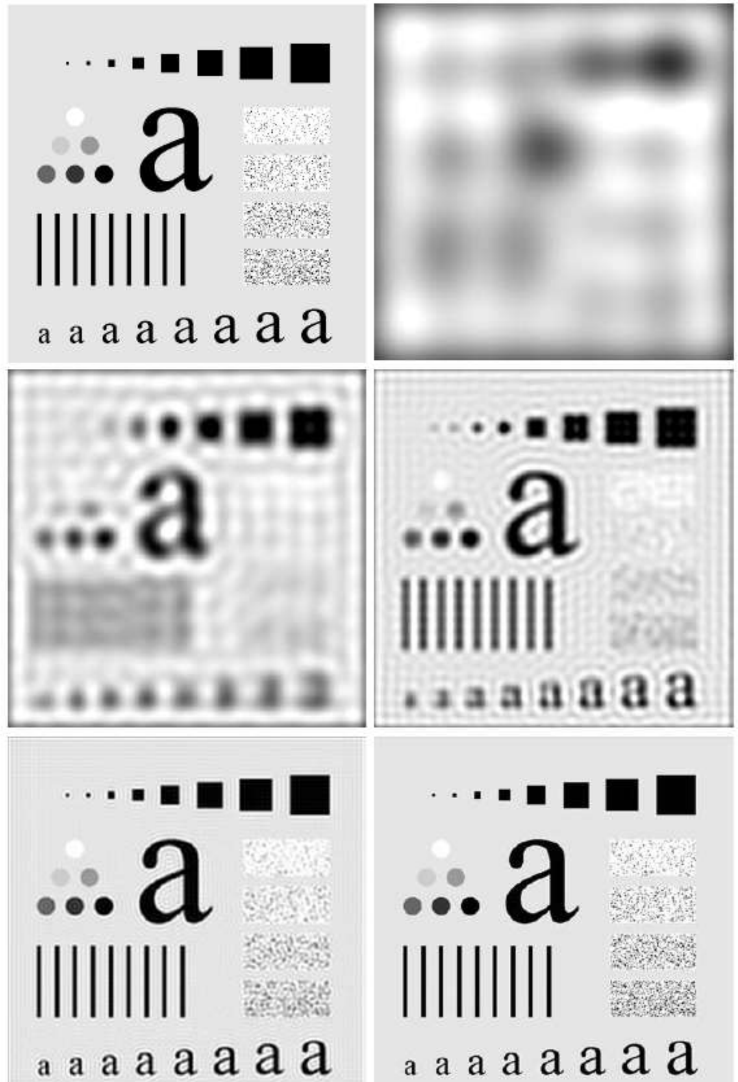


Image Enhancement in Frequency Domain

Smoothing Filters: Butterworth Lowpass Filters (BLPF)

Butterworth Lowpass Filters (BLPF) of order n and with cutoff frequency D_0

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

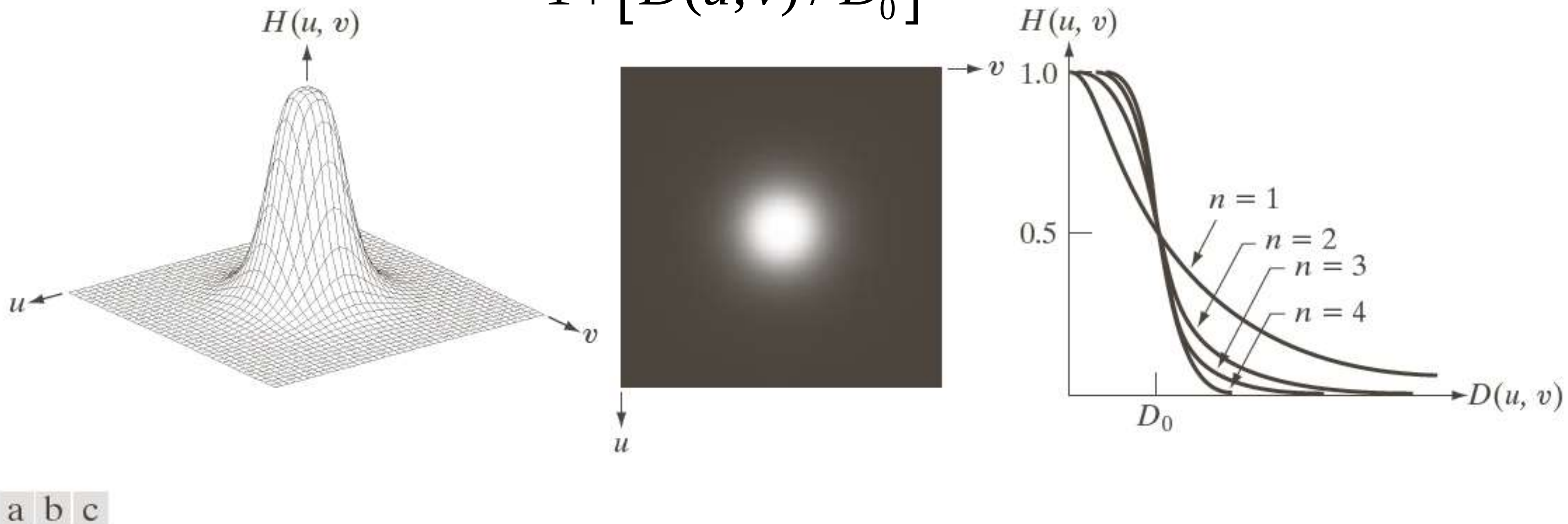
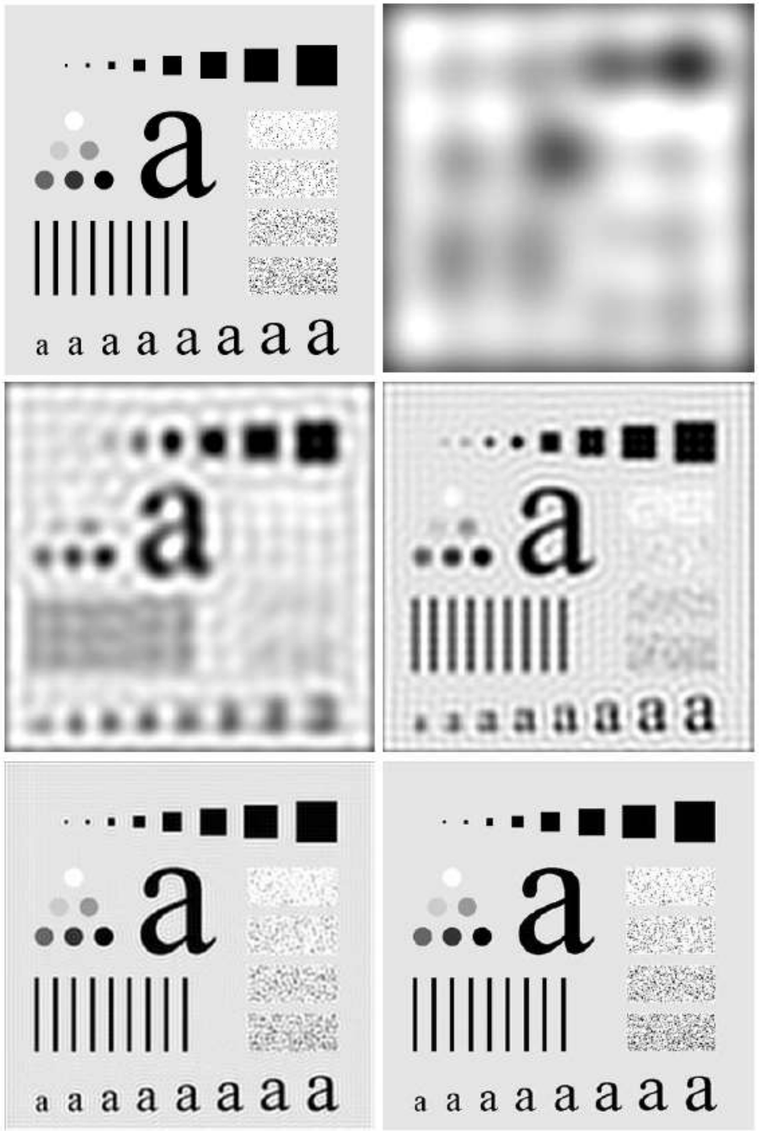
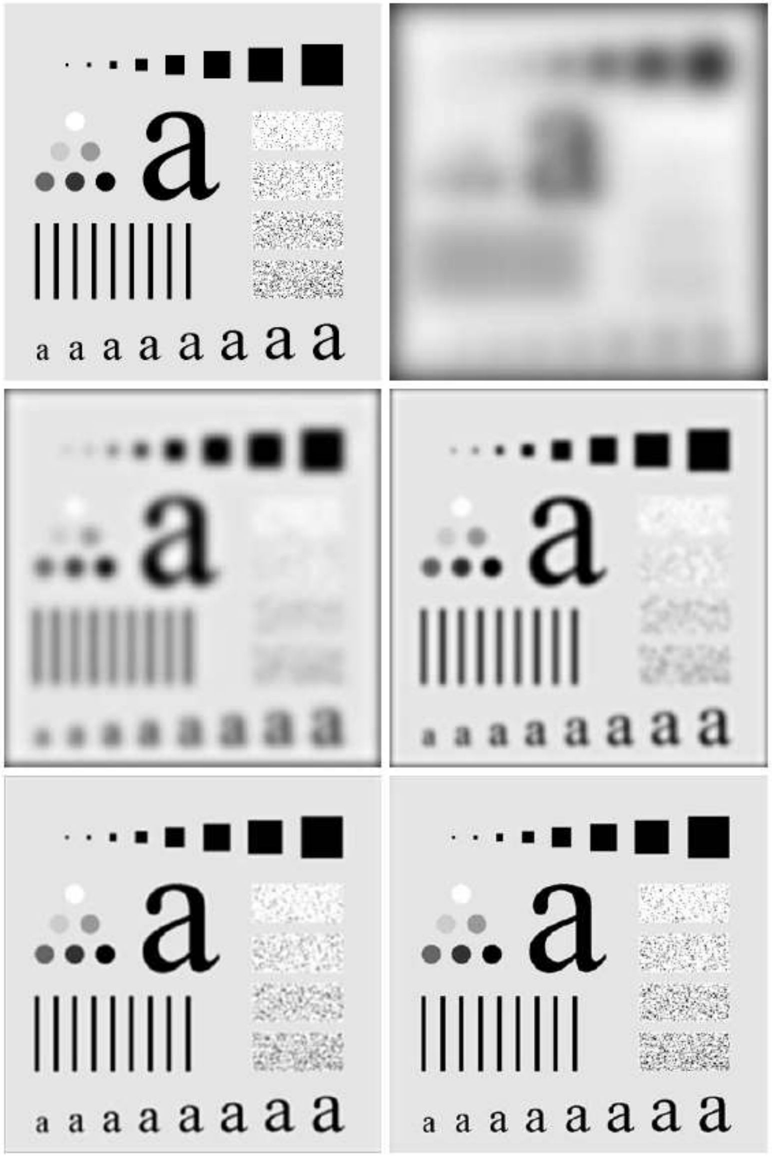


FIGURE 4.44 (a) Perspective plot of a Butterworth lowpass-filter transfer function. (b) Filter displayed as an image. (c) Filter radial cross sections of orders 1 through 4.



a b
c d
e f

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



a b
c d
e f

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

Image Enhancement in Frequency Domain

Smoothing Filters: **Gaussian Lowpass Filters (GLPF)**

Gaussian Lowpass Filters (GLPF) in two dimensions is given

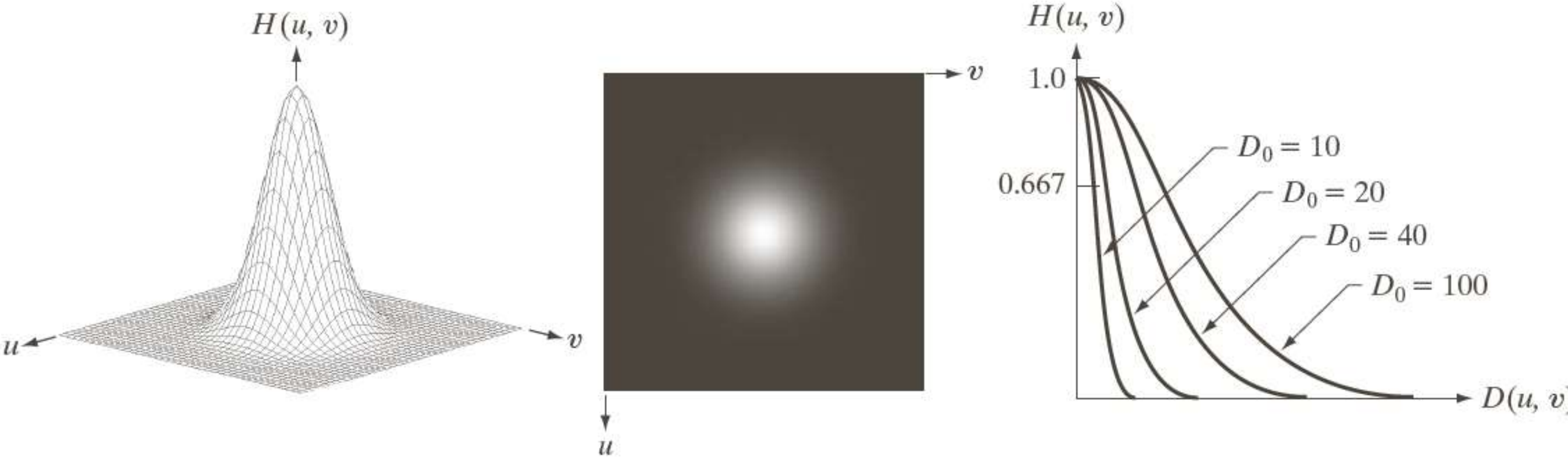
$$H(u,v) = e^{-D^2(u,v)/2\sigma^2}$$

By letting $\sigma = D_0$

$$H(u,v) = e^{-D^2(u,v)/2D_0^2}$$

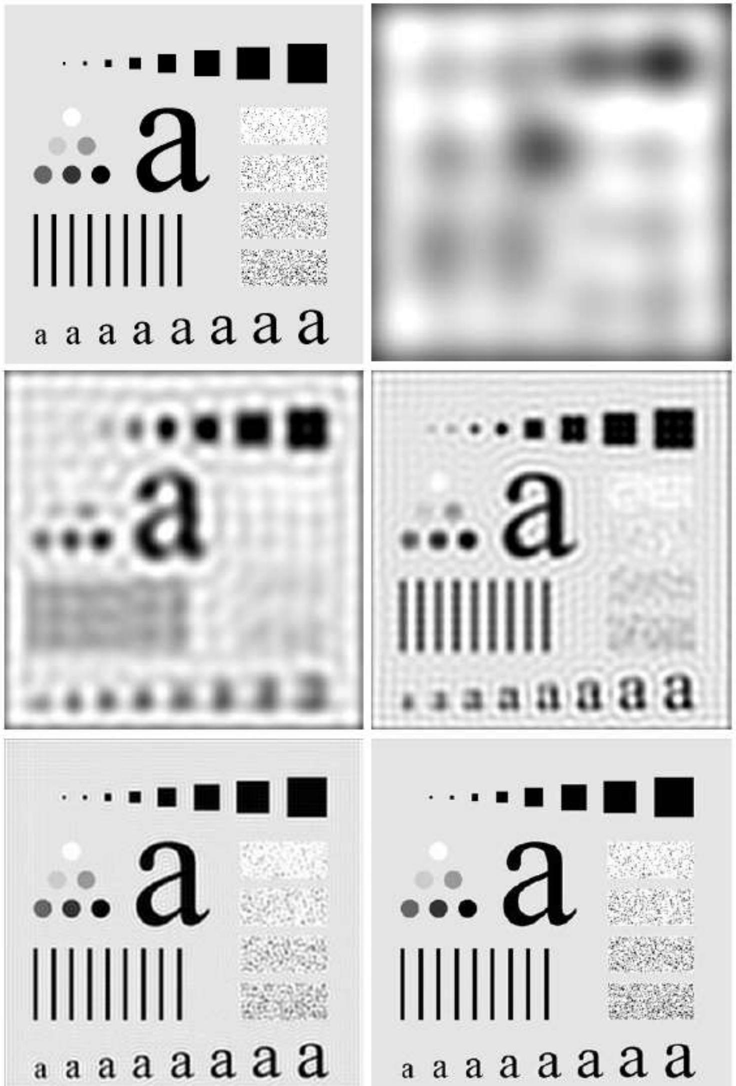
Image Enhancement in Frequency Domain

Smoothing Filters: GLPF



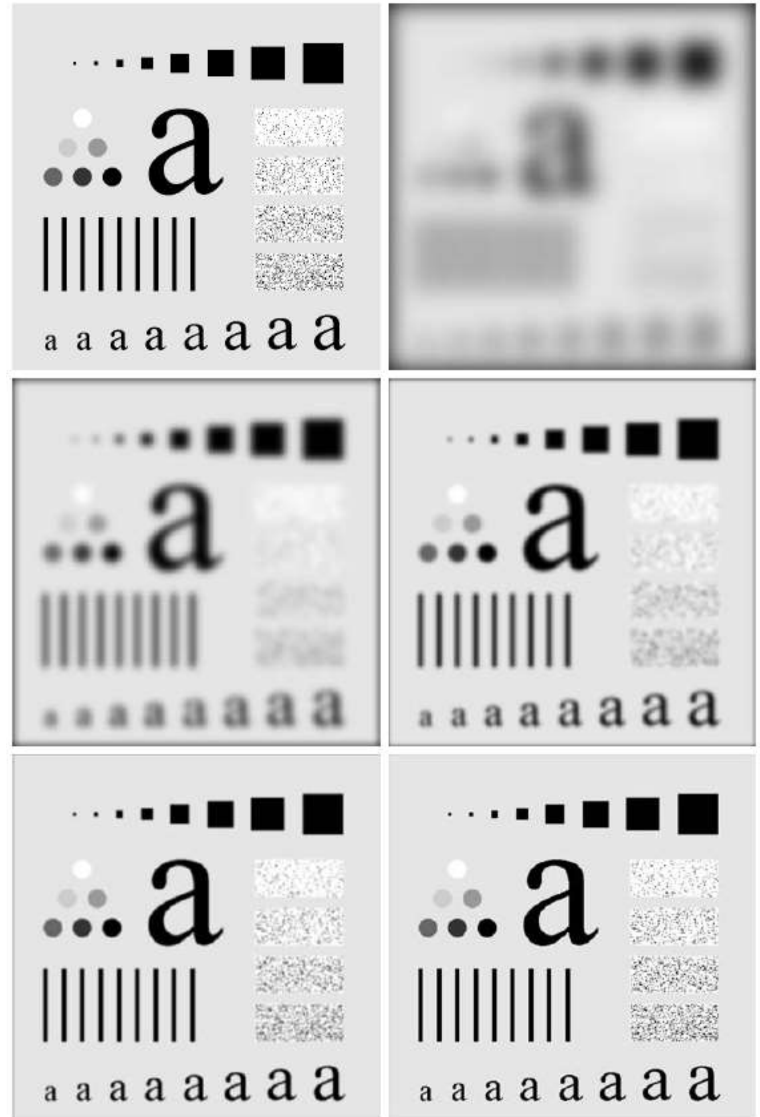
a b c

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.



a b
c d
e f

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

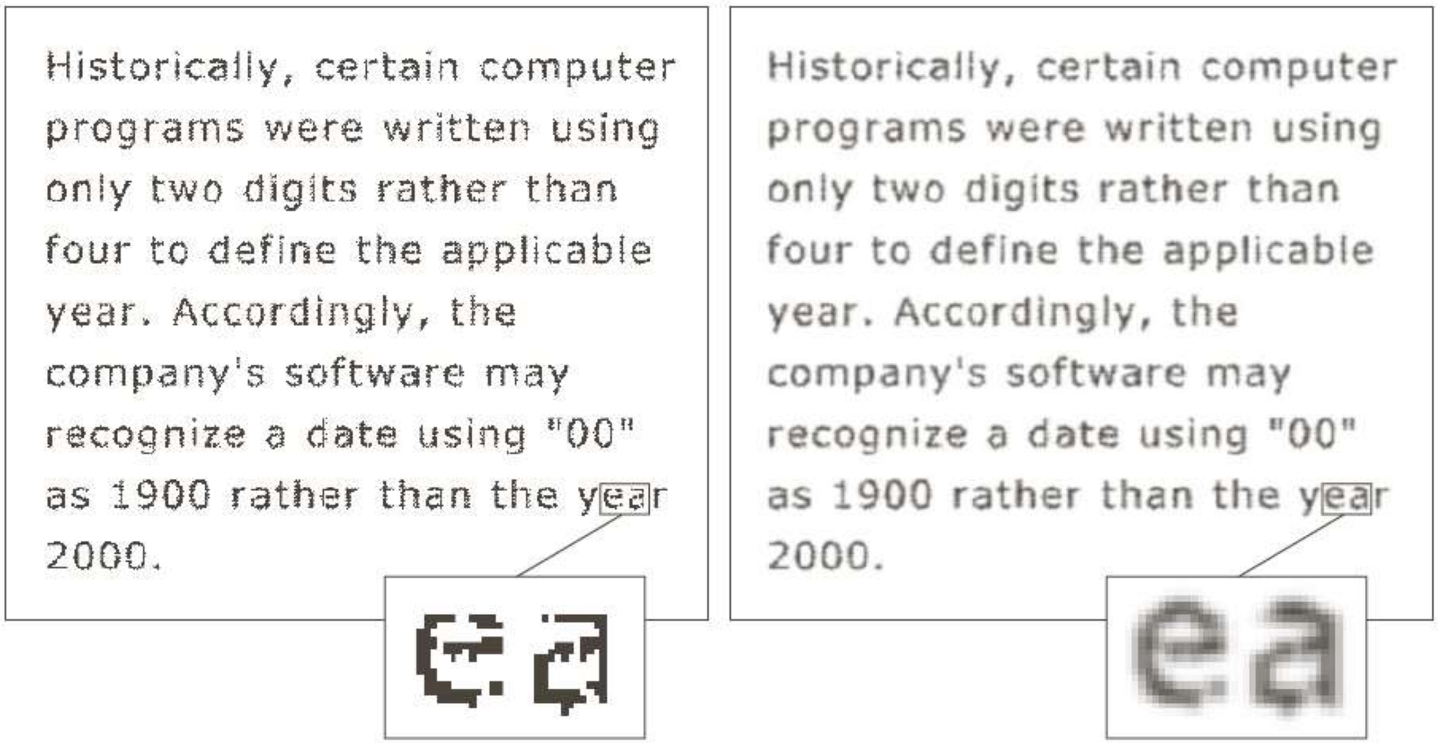


a b
c d
e f

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

Image Enhancement in Frequency Domain

Smoothing Filters: Example of smoothing by GLPF



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).

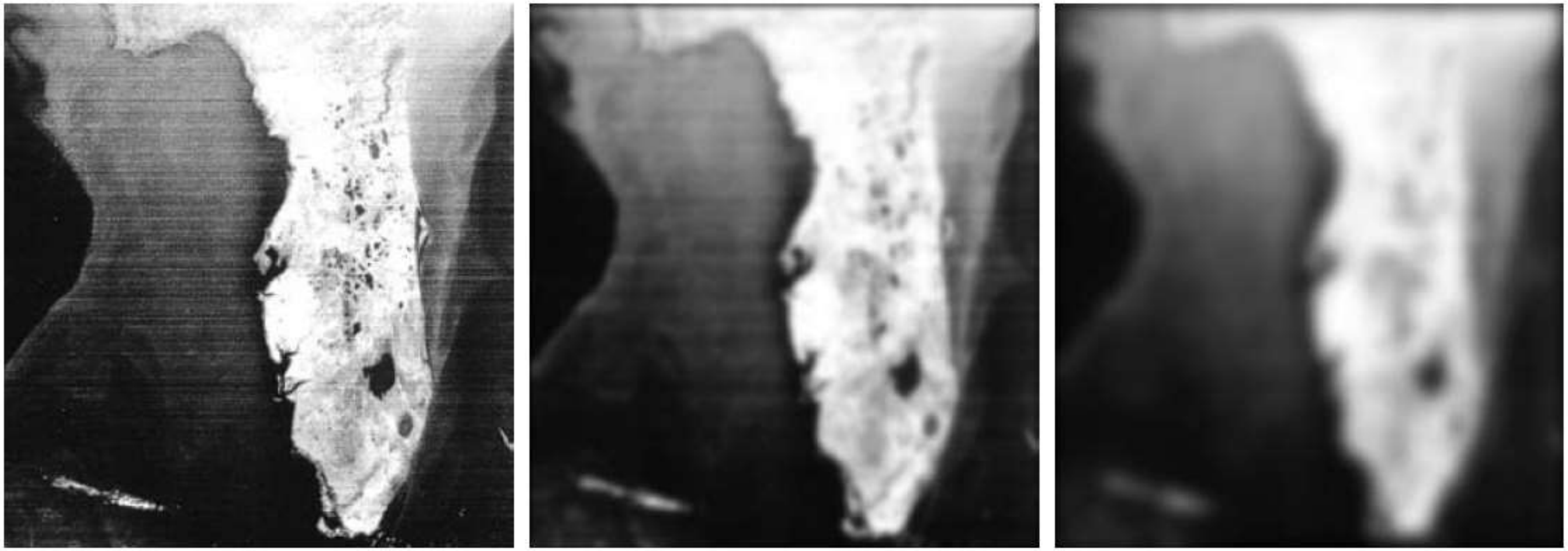
Image Enhancement in Frequency Domain

Smoothing Filters: Example of smoothing by GLPF

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).

Image Enhancement in Frequency Domain

Smoothing Filters: **Example of smoothing by GLPF**



a b c

FIGURE 4.51 (a) Image showing prominent horizontal scan lines. (b) Result of filtering using a GLPF with $D_0 = 50$. (c) Result of using a GLPF with $D_0 = 20$. (Original image courtesy of NOAA.)

Image Enhancement in Frequency Domain

Sharpening Filters

- Image sharpening can be achieved in the frequency domain by highpass filtering.
- It attenuates low-frequency components without disturbing high-frequency information in the Fourier transform.

Image Enhancement in Frequency Domain

Sharpening Filters

A highpass filter is obtained from a given lowpass filter using

$$H_{HP}(u,v) = 1 - H_{LP}(u,v)$$

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

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

Image Enhancement in Frequency Domain

Sharpening Filters

A 2-D Butterworth highpass filter (BHPL) is defined as

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

A 2-D Gaussian highpass filter (GHPL) is defined as

$$H(u,v) = 1 - e^{-D^2(u,v)/2D_0^2}$$

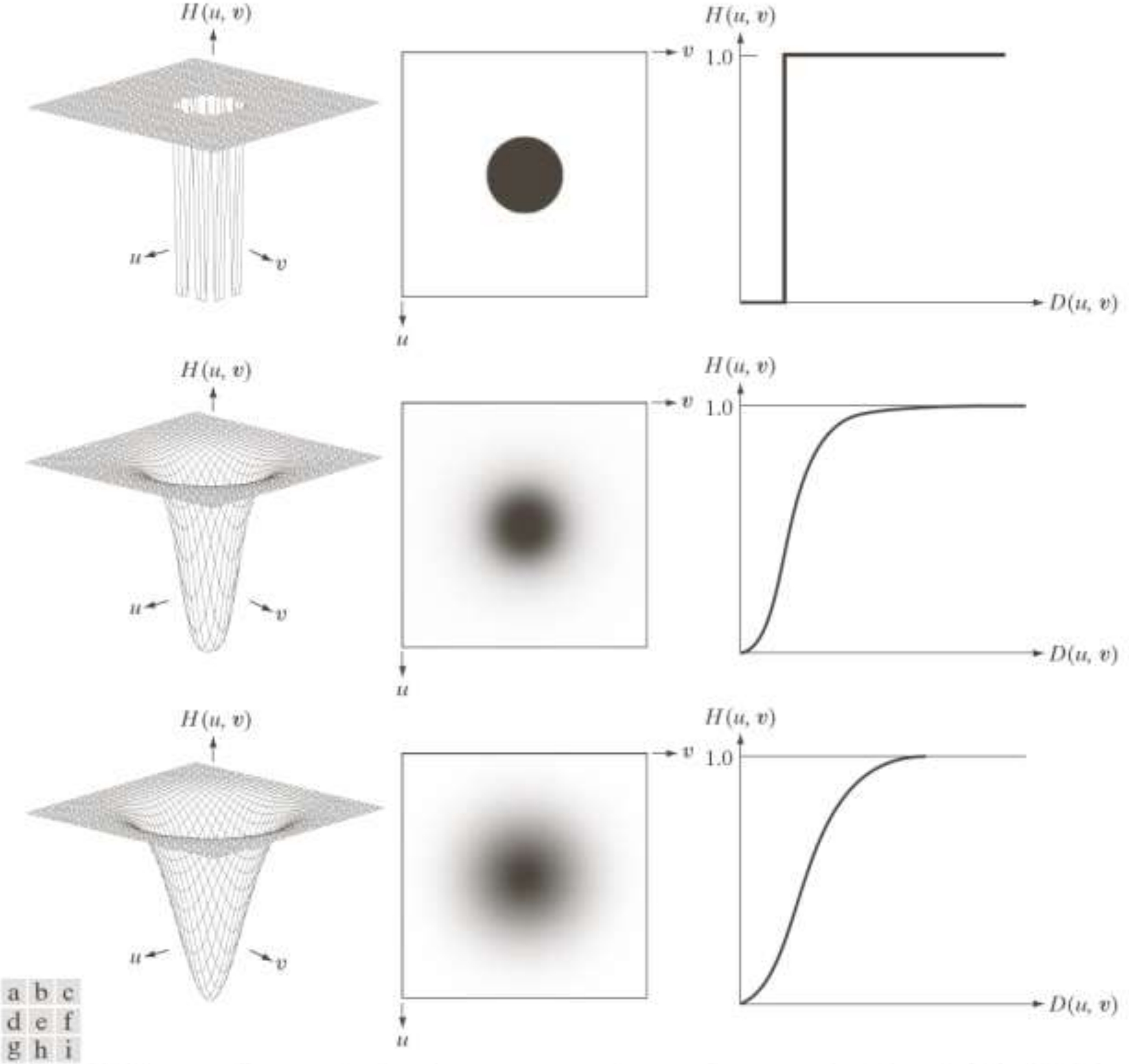


FIGURE 4.52 Top row: Perspective plot, image representation, and cross section of a typical ideal highpass filter. Middle and bottom rows: The same sequence for typical Butterworth and Gaussian highpass filters.

Image Enhancement in Frequency Domain

Sharpening Filters: Result of IHPF

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.

Image Enhancement in Frequency Domain

Sharpening Filters: Result of BHPF

a b c

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

Image Enhancement in Frequency Domain

Sharpening Filters: Result of GHPF

a b c

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

DIGITAL IMAGE PROCESSING

Unit-5: Image Compression

Dr. Piyush Joshi

Indian Institute of Information Technology
Sri City, Andhra Pradesh

Image Compression

Goal:

- Reduce the amount of data required to represent a digital image.
- The underlying basis of reduction process is the removal of redundant data.

Usages:

- Data storage
- Data transmission

Applications:

- Video conferencing, broadcast television, remote sensing, document and medical imaging etc.

Example : A 2-hr SD movie

- Array size: 720 x 480 x 24
- 30 fps
- Data rate: 30 x 720 x 480 x 3 bytes per sec = 31,104,000
- For 2-hr: 31,104,000 x (60^2) sec/hr x 2 hrs = 2.24×10^{11} bytes or 224 GB
- 1 DVD = 4.7GB
- Total DVDs for a movie = 224/4.7 = 47

Image Compression

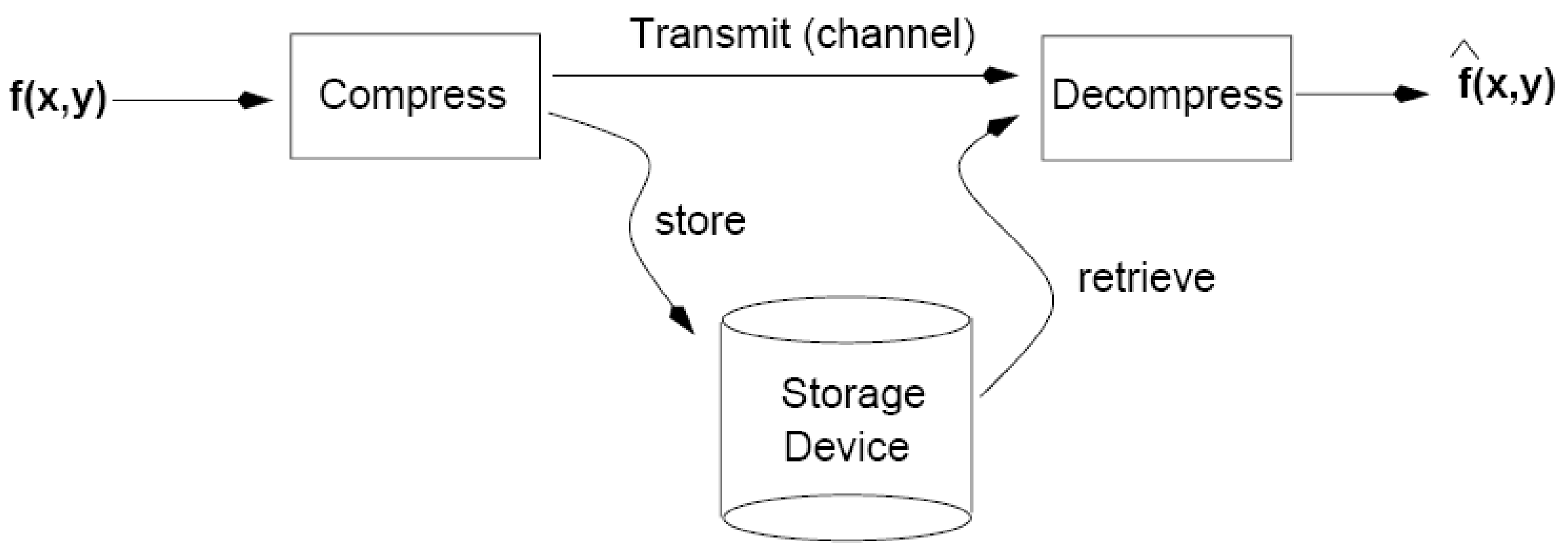


Image Compression

Data ≠ Information

- Data and information are not synonymous terms!
- Data is the means by which information is conveyed.
- The same information can be represented by different amount of data.

Ex1: *Your wife, Helen, will meet you at Logan Airport in Boston at 5 minutes past 6:00 pm tomorrow night.*
Ex2: *Your wife will meet you at Logan Airport at 5 minutes past 6:00 pm tomorrow night.*
Ex3: *Helen will meet you at Logan at 6:00 pm tomorrow night*

- Goal: Reduce the amount of data while preserving as much information as possible!

Image Compression

Relative Data Redundancy

- Let n_1 and n_2 denote the number of bits in two representations of the same information, the relative data redundancy R of the first dataset is

$$R = 1 - 1/C$$

C is called the compression ratio, defined as

$$C = \frac{n_1}{n_2}$$

e.g., C = 10, the corresponding relative data redundancy of the larger representation is 0.9, indicating that 90% of its data is redundant.

Image Compression

- **Lossless**
  -- Information preserving
  -- Low compression ratios

- **Lossy**
  -- Information loss
  -- High compression ratios

Trade-off: information loss **vs** compression ratio

Image Compression

Types of Data Redundancy

- Coding Redundancy
- Interpixel Redundancy
- Psychovisual Redundancy

Image Compression

Coding Redundancy

To reduce coding redundancy, we need efficient coding schemes.

- **Code:** a list of symbols (letters, numbers, bits etc.)
- **Code word:** a sequence of symbols used to represent some information (e.g., gray levels).
- **Code word length:** number of symbols in a code word.
  -- Could be fixed or variable

Example: (binary code, symbols: 0,1, length: 3)

0: 000	4: 100
1: 001	5: 101
2: 010	6: 110
3: 011	7: 111

Image Compression

Coding Redundancy

To compare the efficiency of different coding schemes, we need to compute the average number of symbols L_{avg} per code word.

Example: **N x M** image

r_k: k-th gray level

$l(r_k)$: # of bits for r_k

$P(r_k)$: probability of r_k

$$\text{Average \# of bits: } L_{avg} = E(l(r_k)) = \sum_{k=0}^{L-1} l(r_k)P(r_k)$$

Average Image Size: NML_{avg}

Image Compression

Coding Redundancy

Case 1: $l(r_k)$ = fixed length (code 1)

r_k	$p_r(r_k)$	Code 1	$l_1(r_k)$
$r_0 = 0$	0.19	000	3
$r_1 = 1/7$	0.25	001	3
$r_2 = 2/7$	0.21	010	3
$r_3 = 3/7$	0.16	011	3
$r_4 = 4/7$	0.08	100	3
$r_5 = 5/7$	0.06	101	3
$r_6 = 6/7$	0.03	110	3
$r_7 = 1$	0.02	111	3

Assume an image with $L = 8$

Assume $l(r_k) = 3,\ L_{avg} = \sum_{k=0}^{7} 3P(r_k) = 3\sum_{k=0}^{7} P(r_k) = 3$ bits

Total number of bits: $3NM$

Image Compression

Coding Redundancy

Case 2: $l(r_k)$ = variable length (code 2 – Huffman code)

r_k	$p_r(r_k)$	Code 1	$l_1(r_k)$	Code 2	$l_2(r_k)$
$r_0 = 0$	0.19	000	3	11	2
$r_1 = 1/7$	0.25	001	3	01	2
$r_2 = 2/7$	0.21	010	3	10	2
$r_3 = 3/7$	0.16	011	3	001	3
$r_4 = 4/7$	0.08	100	3	0001	4
$r_5 = 5/7$	0.06	101	3	00001	5
$r_6 = 6/7$	0.03	110	3	000001	6
$r_7 = 1$	0.02	111	3	000000	6

TABLE 8.1 Example of variable-length coding.

$$L_{avg} = \sum_{k=0}^{7} l(r_k)P(r_k) = 2.7 \text{ bits}$$

$$C_R = \frac{3}{2.7} = 1.11 \text{ (about 10\%)}$$

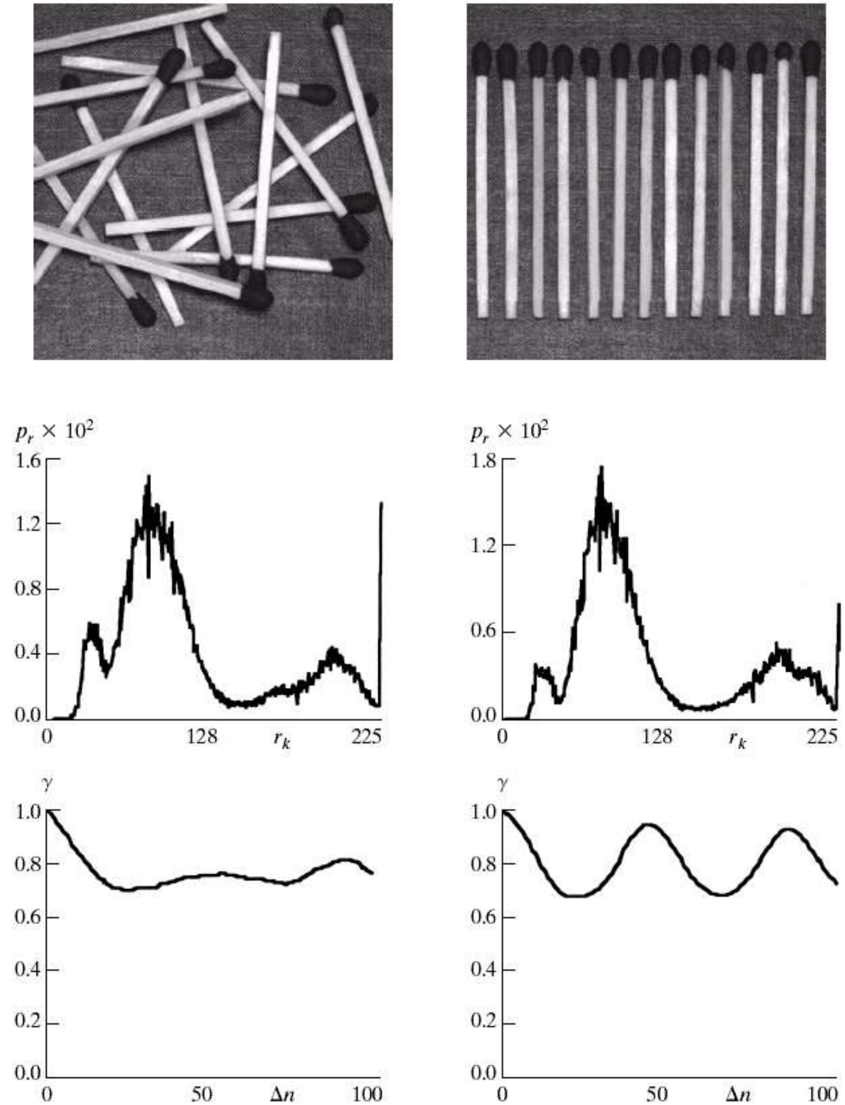
Total number of bits: 2.7NM

$$R_D = 1 - \frac{1}{1.11} = 0.099$$

Image Compression

Interpixel Redundancy

Interpixel redundancy implies that pixel values are correlated (i.e., a pixel value can be reasonably predicted by its neighbors).



a b
c d
e f

FIGURE 8.2 Two images and their gray-level histograms and normalized autocorrelation coefficients along one line.

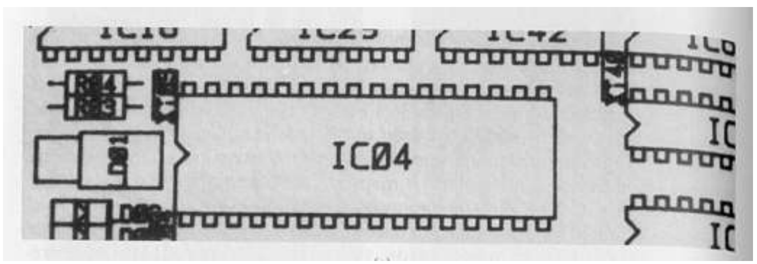
Image Compression

Interpixel Redundancy

- To reduce interpixel redundancy, some kind of a transformation must be applied on the data.

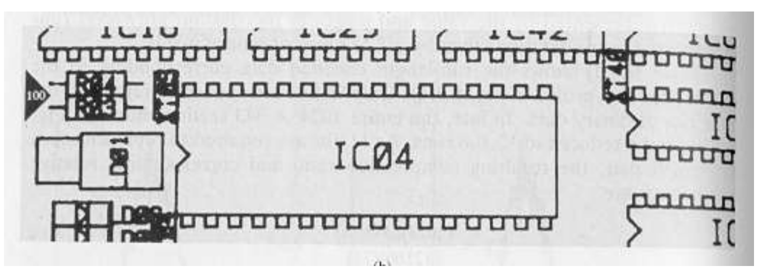
Example

Original

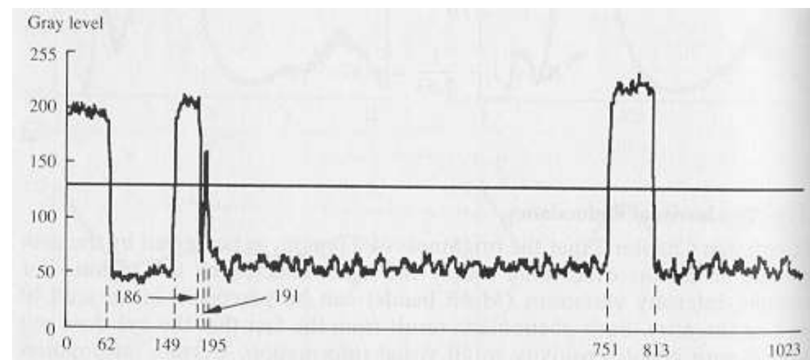


threshold

Binary image



Additional savings using run-length coding



11 …………….0000……………………….11…..000….

Run-length encoding:

(1,63) (0,87) (1,37) (0,5) (1,4) (0, 556) (1,62) (0,210)

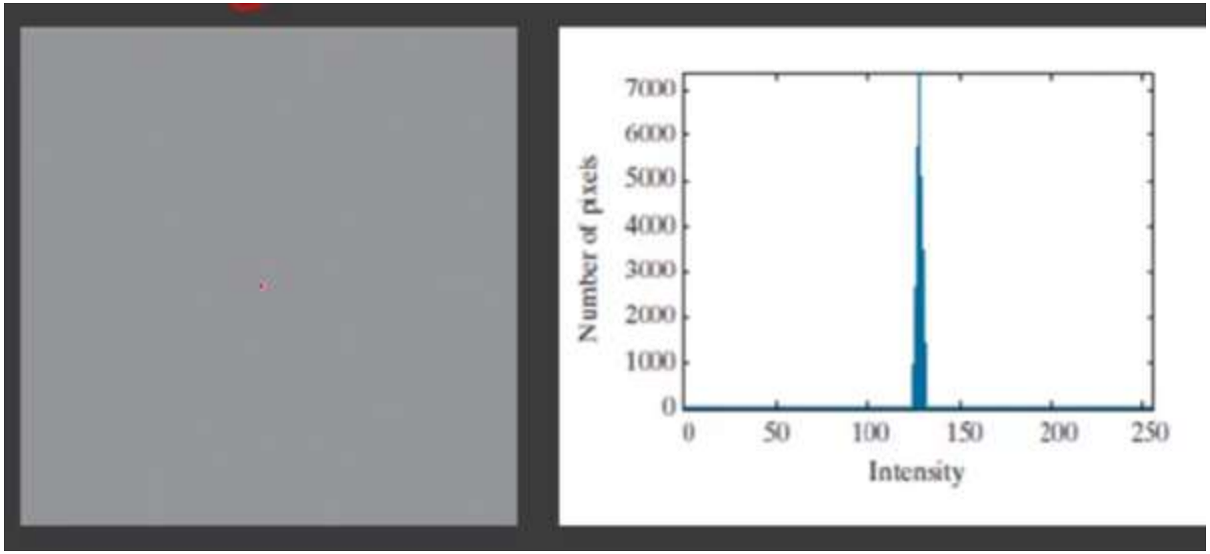
Using 11 bits/pair:

88 bits are required (compared to 1024 !!)

Image Compression

Psychovisual Redundancy

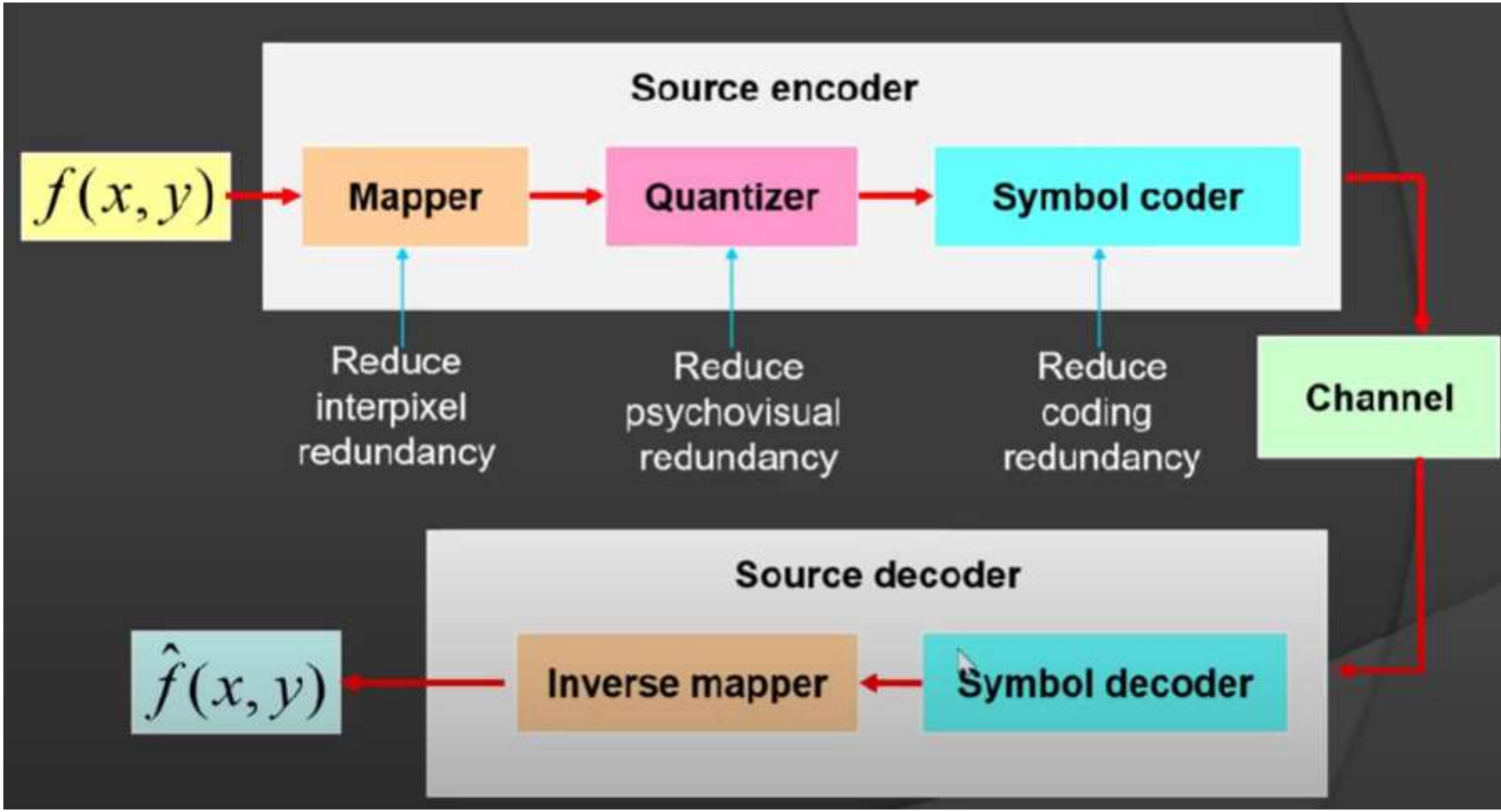
- The human eye is more sensitive to the lower frequencies than to the higher frequencies in the visual spectrum.
- Idea: discard data that is perceptually insignificant.



C=8/4 = 2:1

Image Compression

Image Compression Model



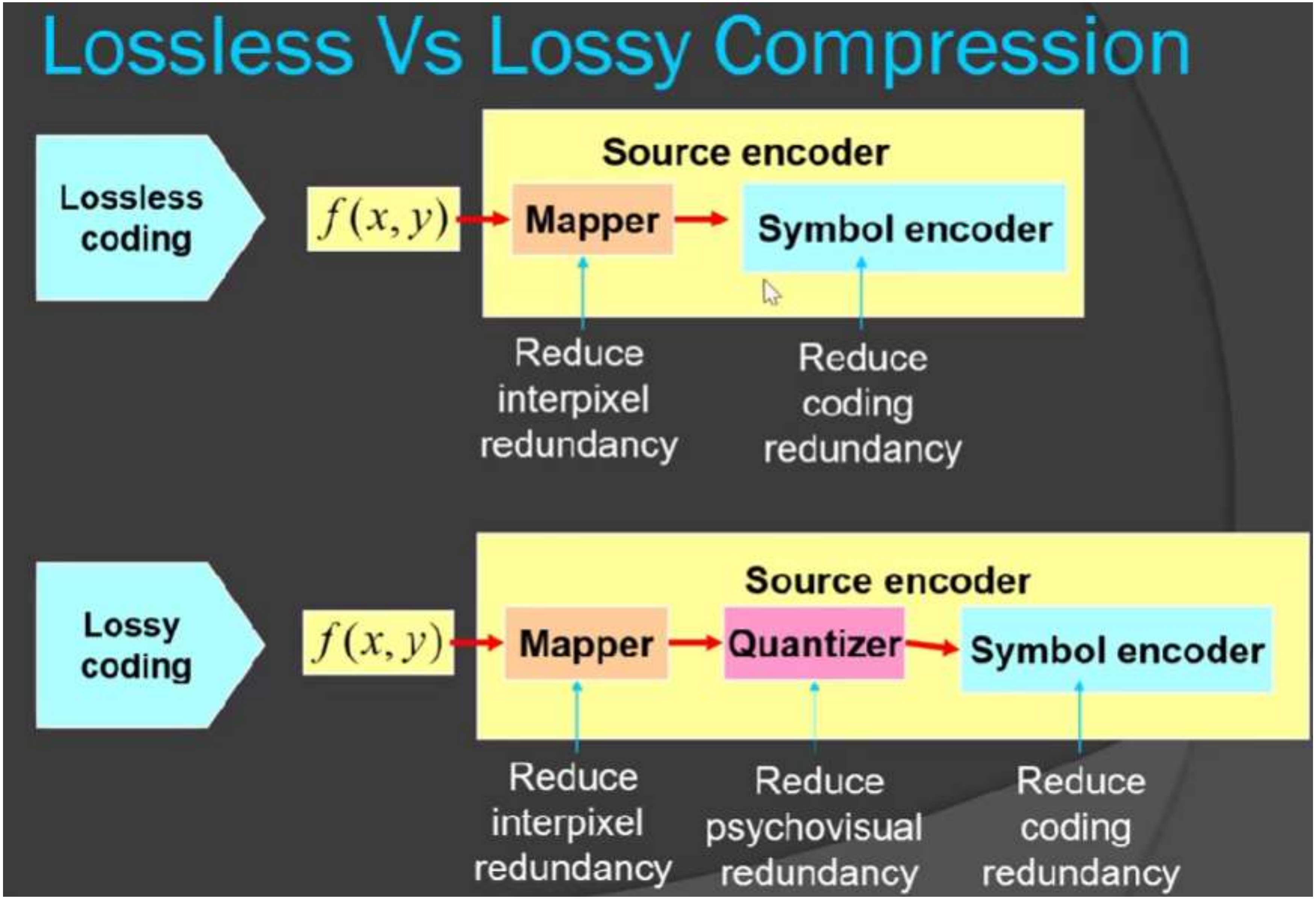

Lossless Vs Lossy Compression
Lossless coding
$f(x,y)$
Source encoder
Mapper
Symbol encoder
Reduce interpixel redundancy
Reduce coding redundancy
Lossy coding
$f(x,y)$
Source encoder
Mapper
Quantizer
Symbol encoder
Reduce interpixel redundancy
Reduce psychovisual redundancy
Reduce coding redundancy

Image Compression

Fidelity Criteria

- How close is $f(x,y)$ to $\hat{f}(x,y)$?

- Criteria
  - Subjective: based on human observers
  - Objective: mathematically defined criteria

Image Compression

Lossless Coding Techniques

- Statistical Encoding
  - Huffman Coding
  - Arithmetic Coding
  - Lempel-Ziv-Welch (LZW) Coding
- Repetitive Sequence Encoding
  - Run-Length Encoding (RLE)

Image Compression

Huffman Coding

- Addresses **coding redundancy**
- A variable-length coding technique
- Source symbols are encoded one at a time!
- There is a one-to-one correspondence between source symbols and code words.
- Optimal code - minimizes code word length per source symbol.

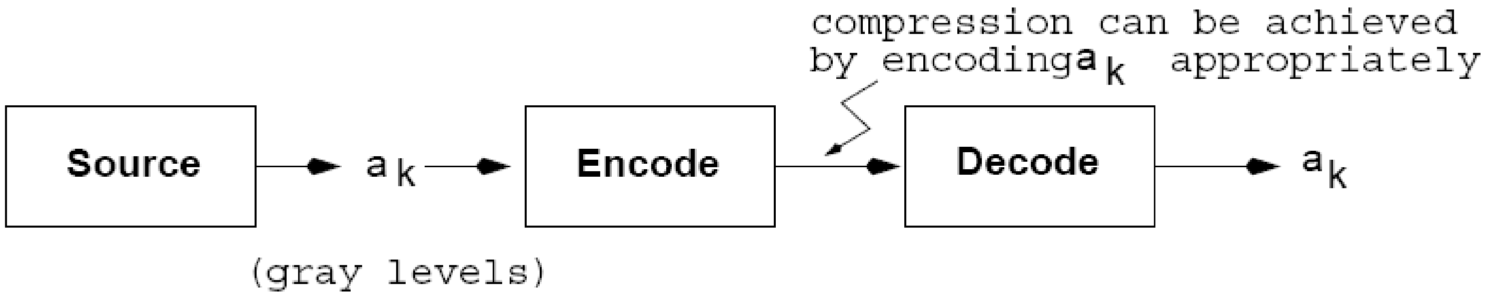


Image Compression

Huffman Coding: Forward Pass

Steps:

1. Sort probabilities per symbol
2. Combine the lowest two probabilities
3. Repeat **Step2** until only two probabilities remain.

Original source		Source reduction			
Symbol	Probability	1	2	3	4
a_2	0.4	0.4	0.4	0.4	0.6
a_6	0.3	0.3	0.3	0.3	0.4
a_1	0.1	0.1	0.2	0.3	
a_4	0.1	0.1	0.1		
a_3	0.06	0.1			
a_5	0.04				

FIGURE 8.7 Huffman source reductions.

Image Compression

Huffman Coding: Backward Pass

- Assign code symbols going backwards

Original source			Source reduction							
Symbol	Probability	Code	1		2		3		4	
a_2	0.4	1	0.4	1	0.4	1	0.4	1	0.6	0
a_6	0.3	00	0.3	00	0.3	00	0.3	00	0.4	1
a_1	0.1	011	0.1	011	0.2	010	0.3	01		
a_4	0.1	0100	0.1	0100	0.1	011				
a_3	0.06	01010	0.1	0101						
a_5	0.04	01011								

FIGURE 8.8
Huffman code assignment procedure.

Image Compression

Huffman Coding

- L_{avg} assuming binary coding:

6 symbols, we need a 3-bit code

$(a_1: 000,\ a_2: 001,\ a_3: 010,\ a_4: 011,\ a_5: 100,\ a_6: 101)$

$$L_{avg} = \sum_{k=1}^{6} l(a_k)P(a_k) = \sum_{k=1}^{6} 3P(a_k) = 3\sum_{k=1}^{6} P(a_k) = 3 \text{ bits/symbol}$$

- L_{avg} assuming Huffman coding:

$$L_{avg} = E(l(a_k)) = \sum_{k=1}^{6} l(a_k)P(a_k) =$$

3x0.1 + 1x0.4 + 5x0.06 + 4x0.1 + 5x0.04 + 2x0.3=2.2 bits/symbol

Image Compression

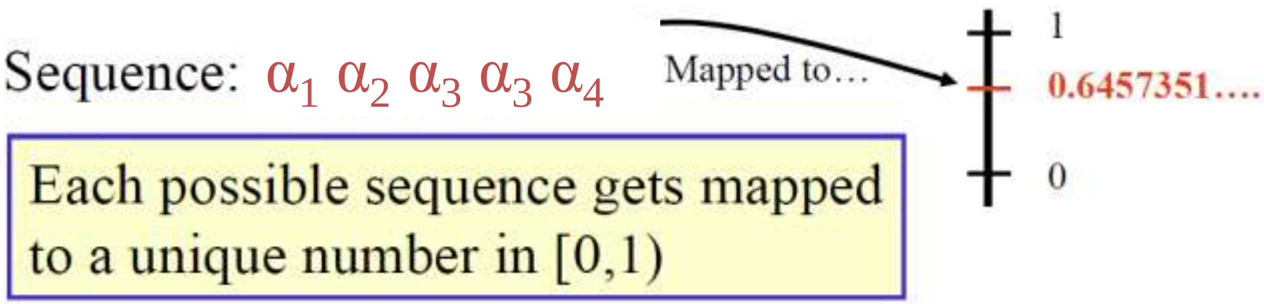
Arithmetic Coding

- Addresses **coding redundancy**
- Huffman coding encodes source symbols one at a time which might not be efficient.
- Arithmetic coding encodes sequences of source symbols to variable length code words.
  - There is no one-to-one correspondence between source symbols and code words.
  - Slower than Huffman coding but can achieve better compression.

Image Compression

Arithmetic Coding

- Map a sequence of symbols to a number (arithmetic code) in the interval [0, 1).
- Encoding the arithmetic code is more efficient.



- The mapping depends on the probabilities of the symbols.
- The mapping is built as each symbol arrives.

Image Compression

Arithmetic Coding

- Start with the interval [0, 1)
- A sub-interval of [0,1) is chosen to represent the first symbol (based on its probability of occurrence).
- As more symbols are encoded, the sub-interval gets smaller and smaller.
- At the end, the symbol sequence is encoded by a number within the final interval.

sub-interval   arithmetic code

$\alpha_1\ \alpha_2\ \alpha_3\ \alpha_3\ \alpha_4 \rightarrow [0.06752, 0.0688) \rightarrow 0.068$

Image Compression

Arithmetic Coding: Example

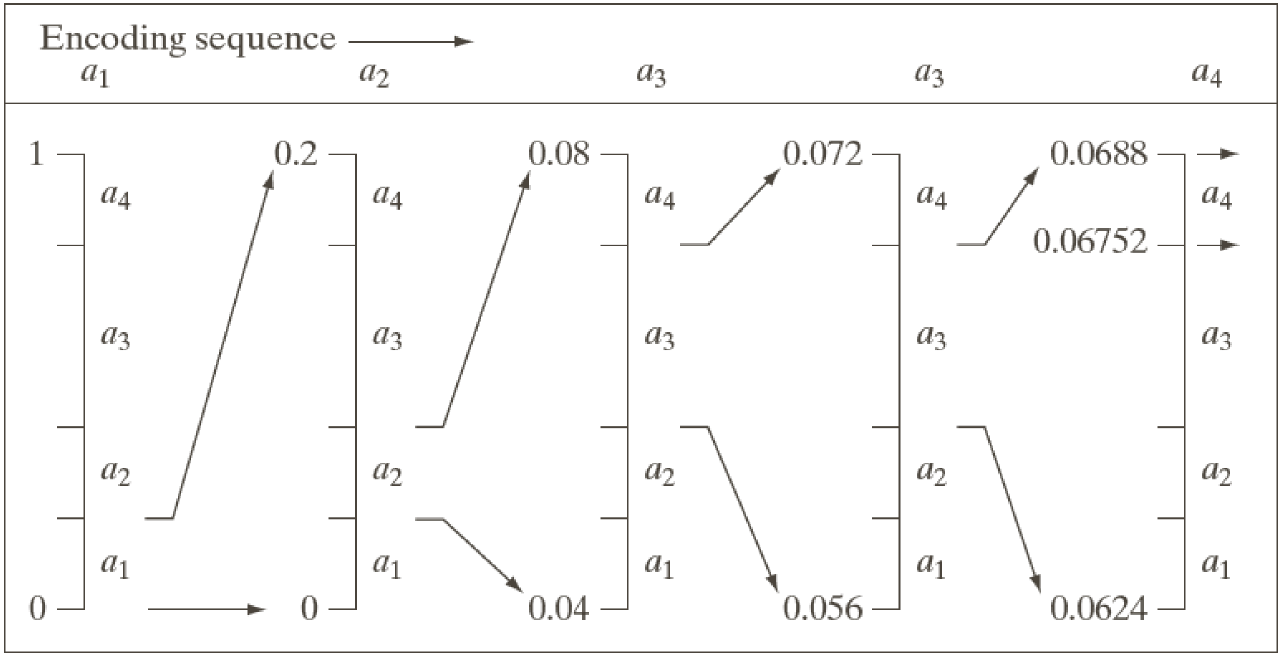


FIGURE 8.12 Arithmetic coding procedure.

Source Symbol	Probability	Initial Subinterval
a_1	0.2	[0.0, 0.2)
a_2	0.2	[0.2, 0.4)
a_3	0.4	[0.4, 0.8)
a_4	0.2	[0.8, 1.0)

TABLE 8.6 Arithmetic coding example.

Image Compression

Arithmetic Coding: Example

- The arithmetic code 0.068 can be encoded using Binary Fraction:

  $0.0068 \approx 0.000100011$ (9 bits) (subject to conversion error; exact value is 0.068359375)

- Huffman Code:

  0100011001 (10 bits) $\alpha_1\ \alpha_2\ \alpha_3\ \alpha_3\ \alpha_4$

- Fixed Binary Code:

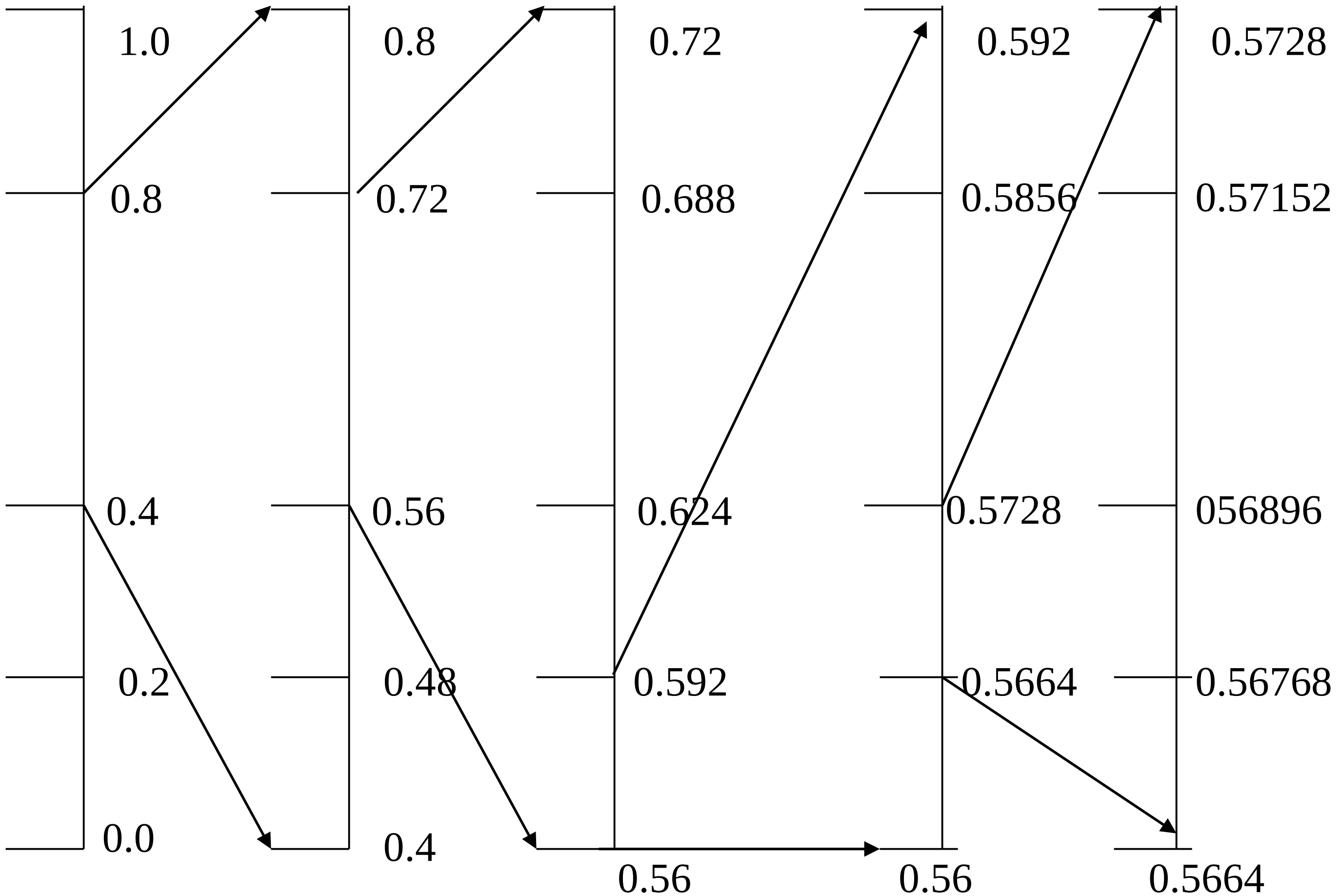
  5 x 8 bits/symbol = 40 bits

Image Compression

Arithmetic Coding: Example

Decode 0.572. The length of the message is 5.

Since 0.8>code word > 0.4, the first symbol should be a_3.



Therefore, the message is $a_3a_3a_1a_2a_4$

Image Compression

Run-Length Coding (RLC)

- Addresses interpixel redundancy.
- Run-length Encoding, or RLE is a technique used to reduce the size of a repeating string of characters.
- This repeating string is called a *run*, typically RLE encodes a run of symbols into two bytes, a count and a symbol.
- RLE can compress any type of data.

Run-length Coding

Example: Given a sample binary image. Apply Run-length Coding.

0	0	0	0	0
0	0	0	1	1
1	1	1	1	1
1	1	1	1	1
1	1	1	1	1

Image Compression

Run-Length Coding (RLC)

- RLE cannot achieve high compression ratios compared to other compression methods.
- It is easy to implement and is quick to execute.
- Run-length encoding is supported by most bitmap file formats such as TIFF, BMP and PCX

1 1 1 1 1 0 0 0 0 0 0 1 → (1,5) (0, 6) (1, 1)
a a a b b b b b b c c → (a,3) (b, 6) (c, 2)