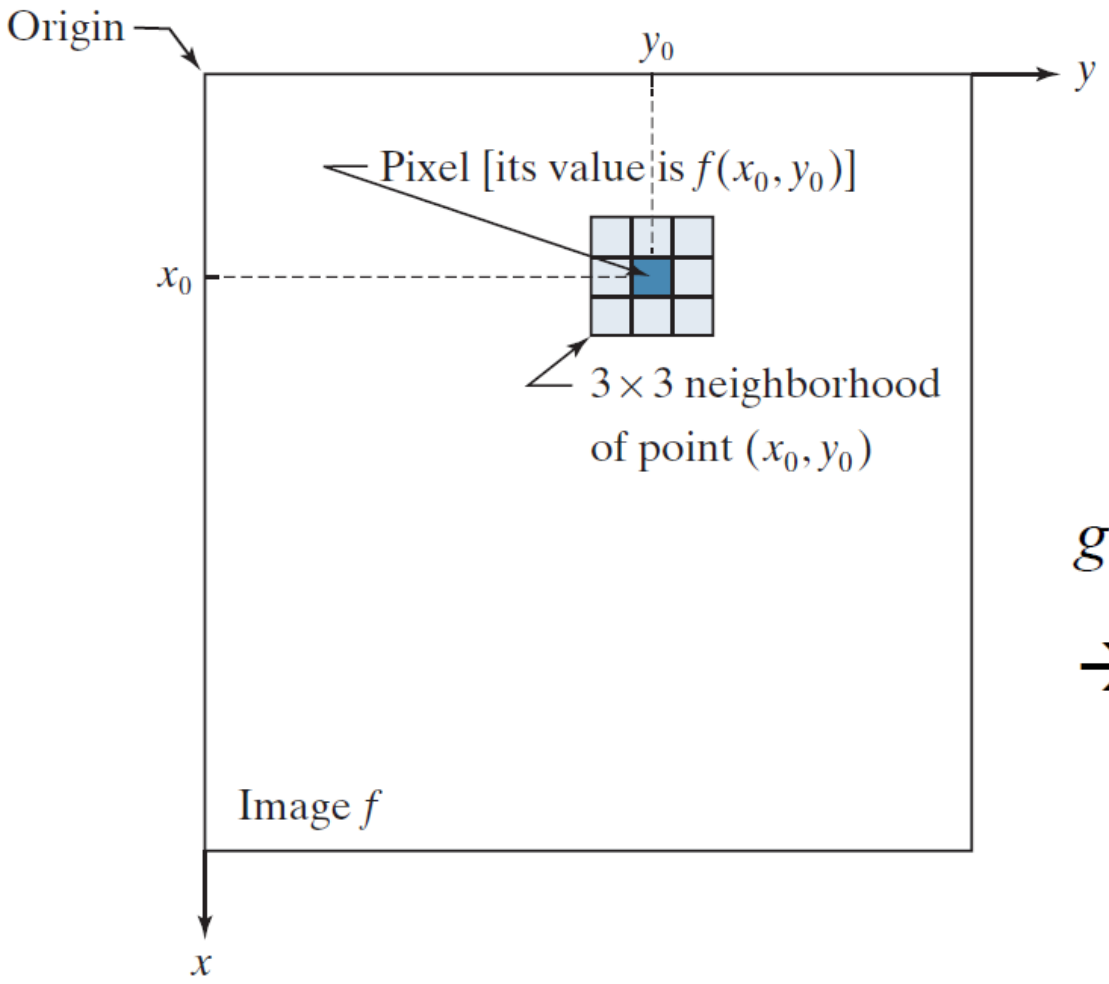
CSL7320: Digital Image Analysis

Image Enhancements

Intensity Transformation

Spatial Domain

FIGURE 3.1
A 3×3 neighborhood about a point (x_0, y_0) in an image. The neighborhood is moved from pixel to pixel in the image to generate an output image. Recall from Chapter 2 that the value of a pixel at location (x_0, y_0) is $f(x_0, y_0)$, the value of the image at that location.



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

→ spatial filter

Slide credit: Yan Tong

Point Processing

Point processing methods (image enhancement techniques) are based only on the transformation of single pixel intensity.

$$s=f(r)$$

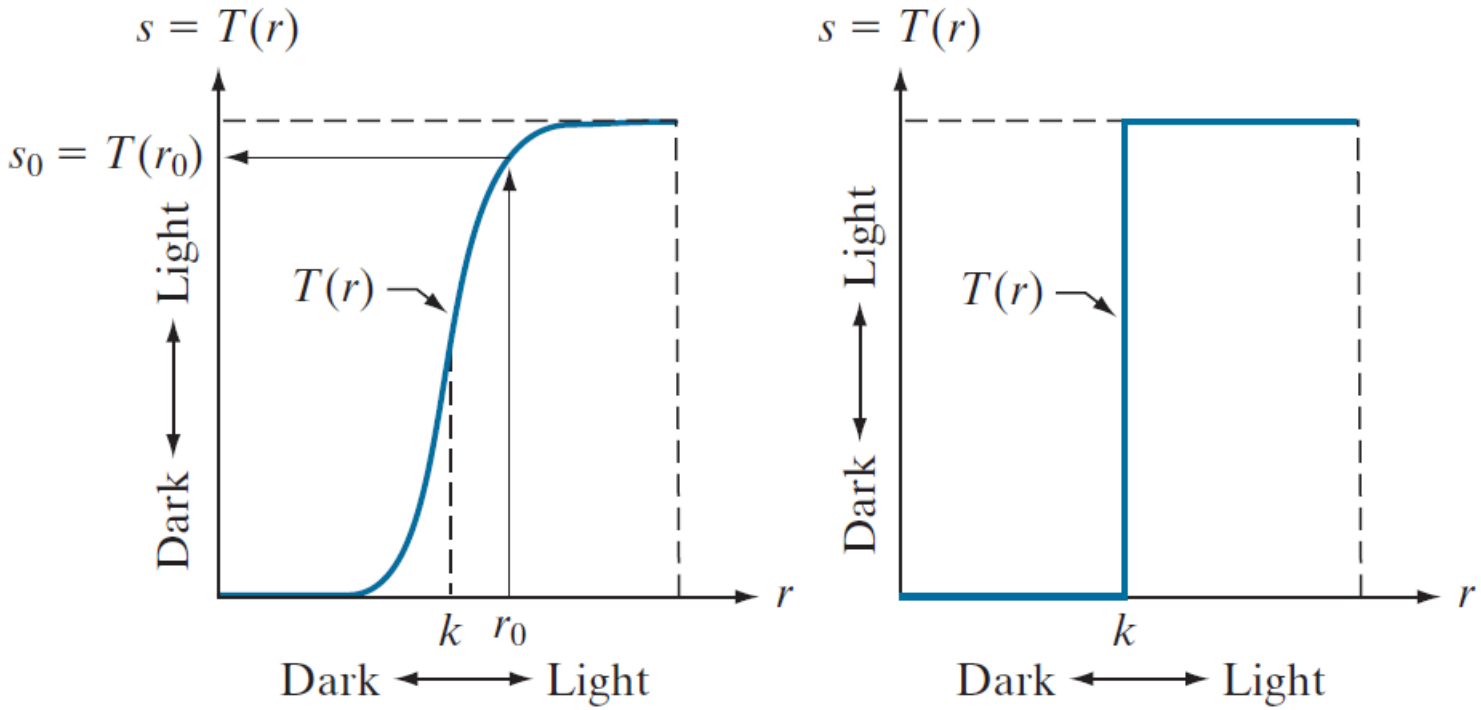
where *r* is the pixel value before processing;
s is the pixel value after processing;
f is a transformation that maps pixel value *r* into pixel value *s*.

Slide credit: Ashish Ghosh

1x1 Neighborhood → Intensity Transformation → Image Enhancement

a b **Contrast stretch**

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



Soft thresholding (logistic function)

Hard thresholding (step function)

$$\sim s = \frac{1}{1 + e^{r}}$$

Slide credit: Yan Tong

Some Basic Intensity Transformation Functions

- **Thresholding – Logistic function**
- **Log transformation**
- **Power-law (Gamma correction)**
- **Piecewise-linear transformation**
- **Histogram processing**

Slide credit: Yan Tong

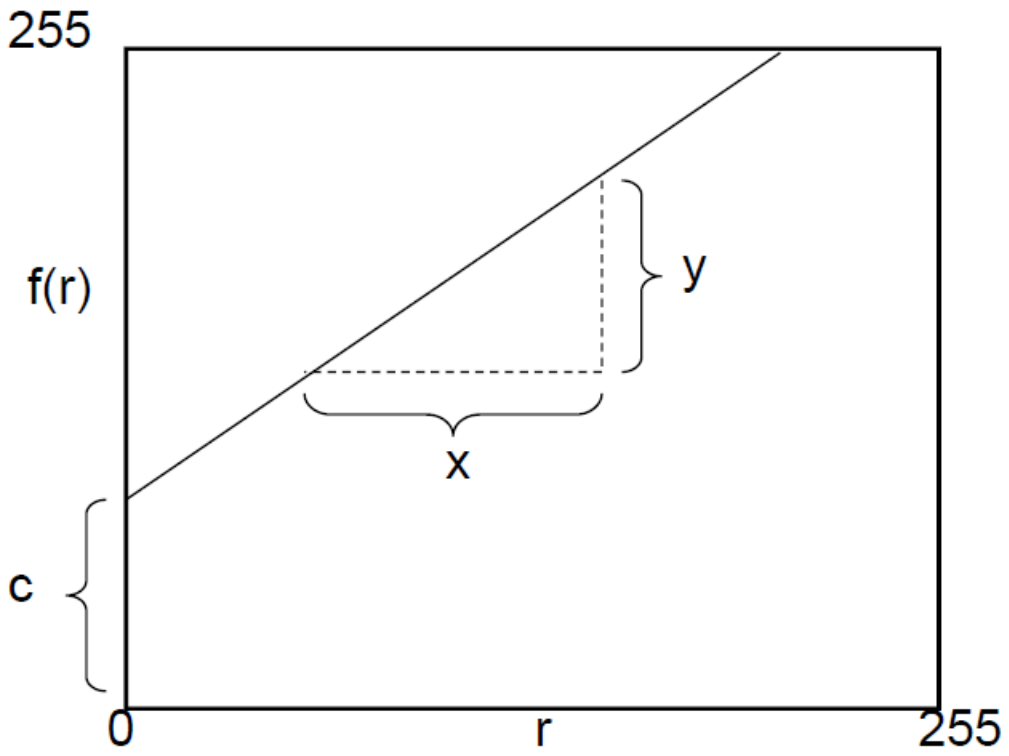
Linear Transformation

It is of the form: $f(r) = m \cdot r + c$

where m = slope = y/x

c = y – intercept

(supplied by the user)

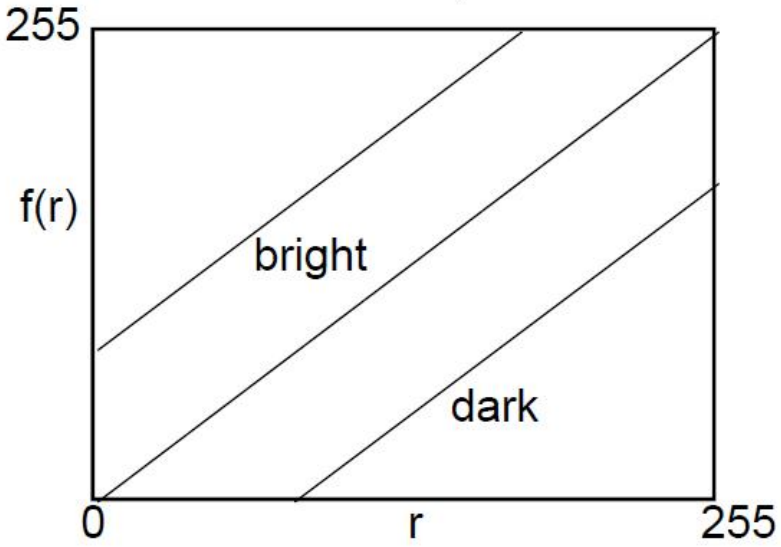


Slide credit: Ashish Ghosh

Linear Transformation: effect on brightness

In order to make the image darker we choose ‘c’ < 0.

In order to make the Image brighter we choose ‘c’ > 0.

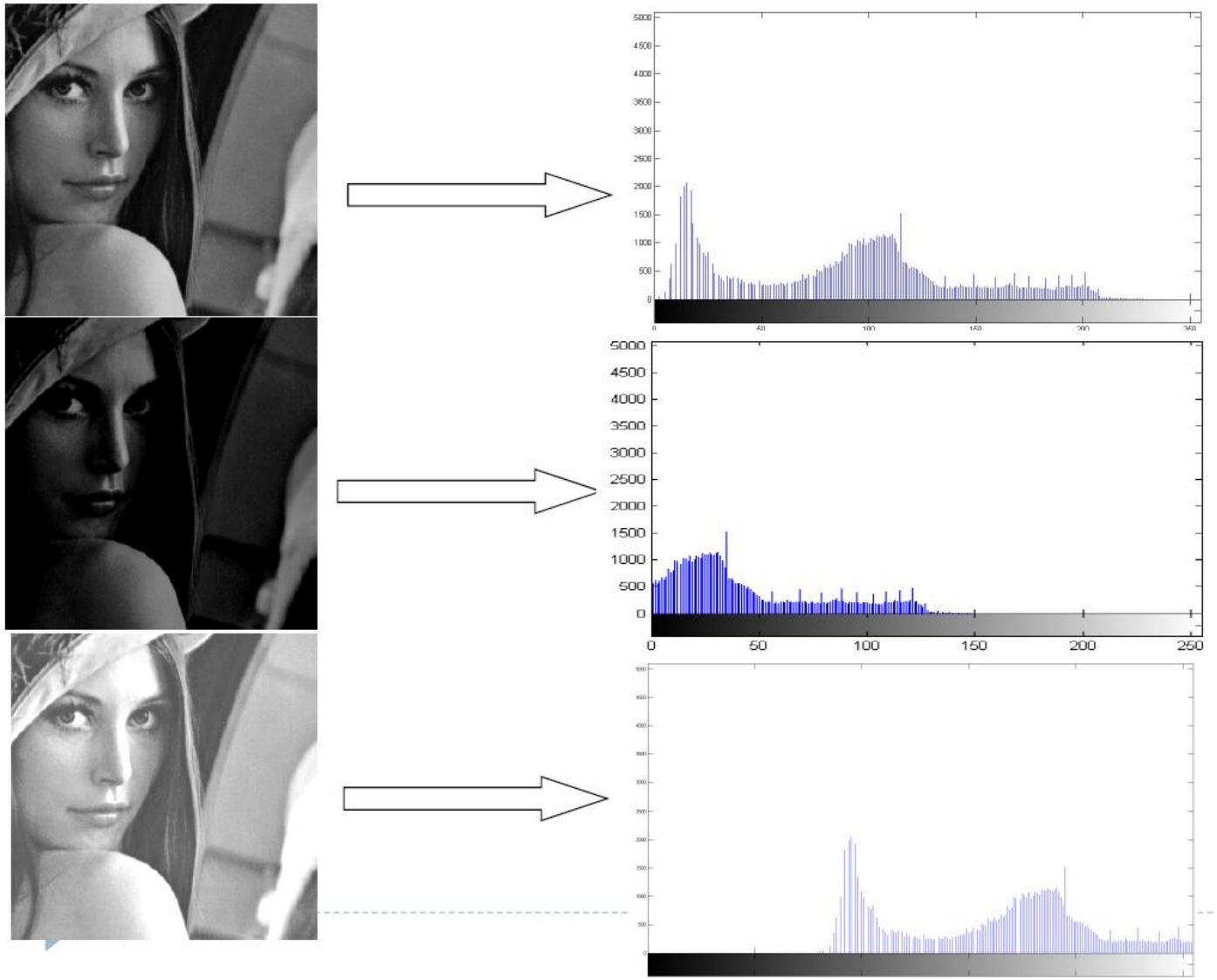


C=-80

C=80

Slide credit: Ashish Ghosh

Linear Transformation: effect on brightness

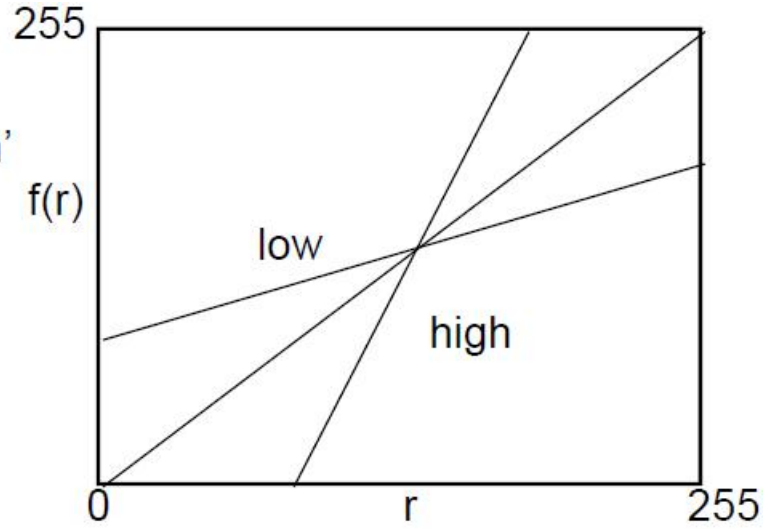


Slide credit: Ashish Ghosh

Linear Transformation: effect on brightness

To increase contrast increase the slope ‘m’

Similarly, decrease the slope in order to decrease the contrast.

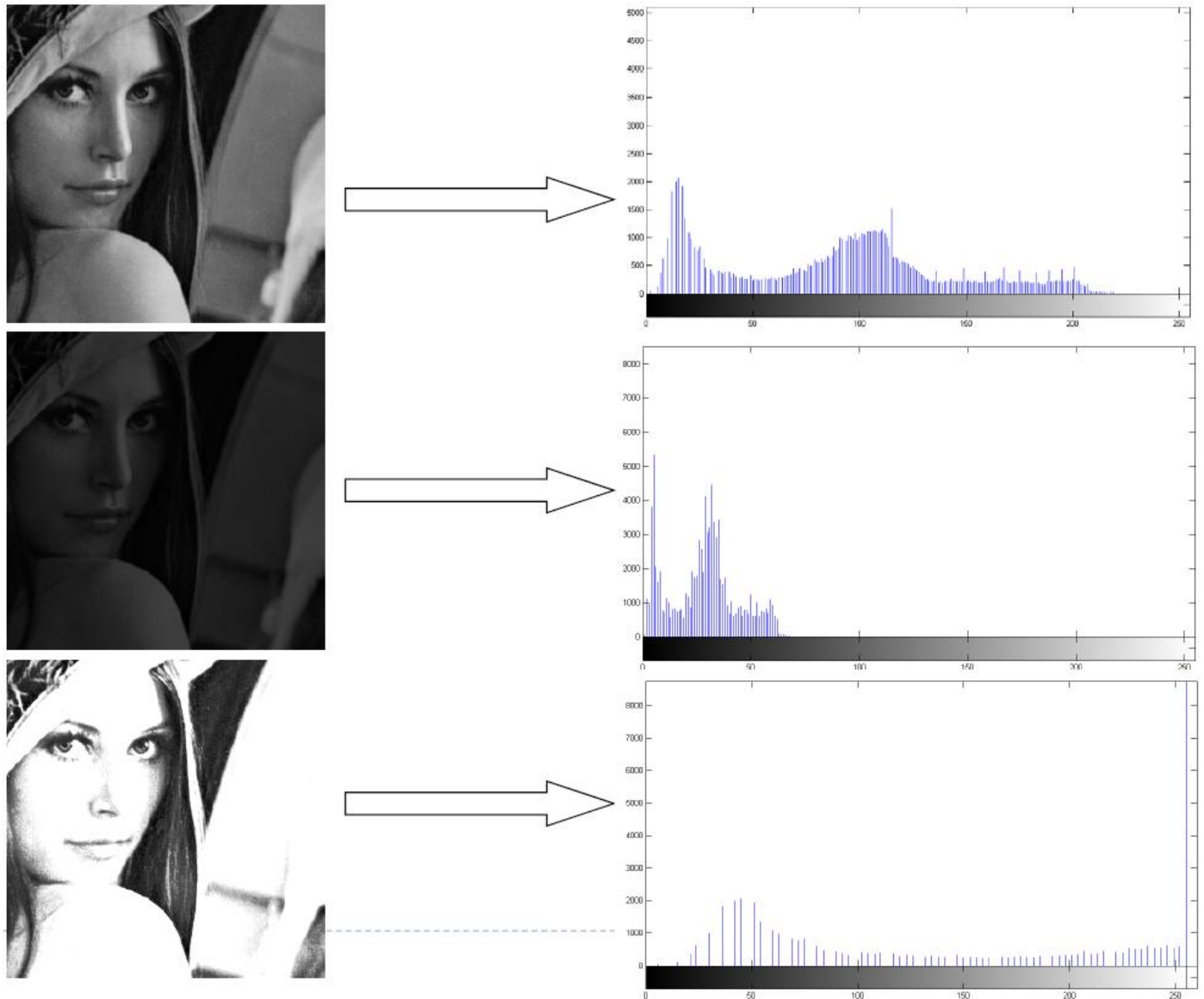


m = 0.3

m = 3.0

Slide credit: Ashish Ghosh

Linear Transformation: effect on brightness



Slide credit: Ashish Ghosh

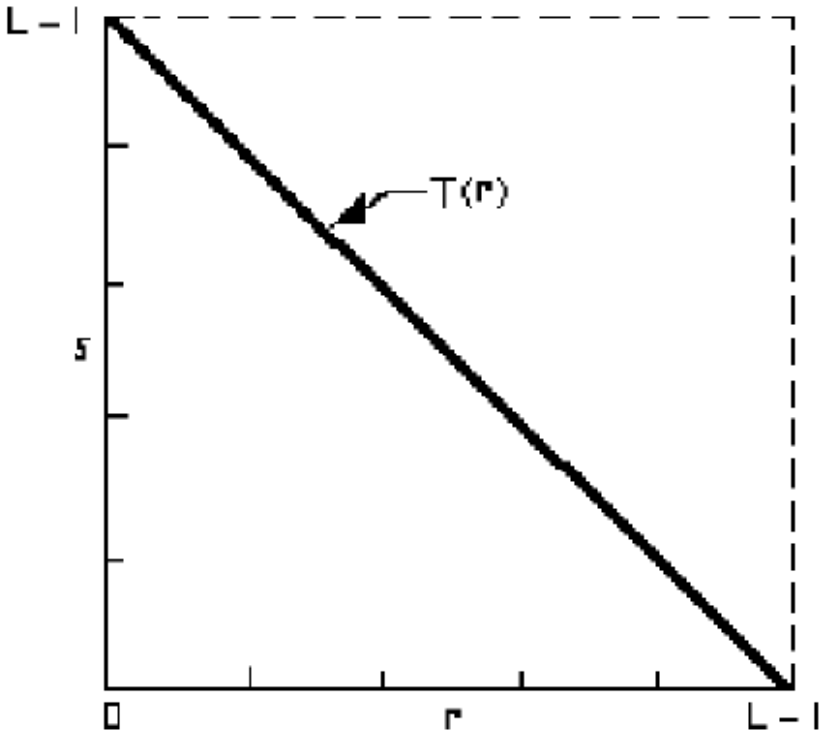
Image Negative

Image negative is obtained by using the transformation function $s = f(r) = (L-1) - r$.

The range of gray levels is $[0, L-1]$.

This function reverses the order from black to white so that the intensity of the output image decreases as the intensity of the input image increases.

It is used in medical images and to produce slides of the screen.



Slide credit: Ashish Ghosh

Image Negative

a b

FIGURE 3.4
(a) A digital mammogram. (b) Negative image obtained using Eq. (3-3). (Image (a) Courtesy of General Electric Medical Systems.)

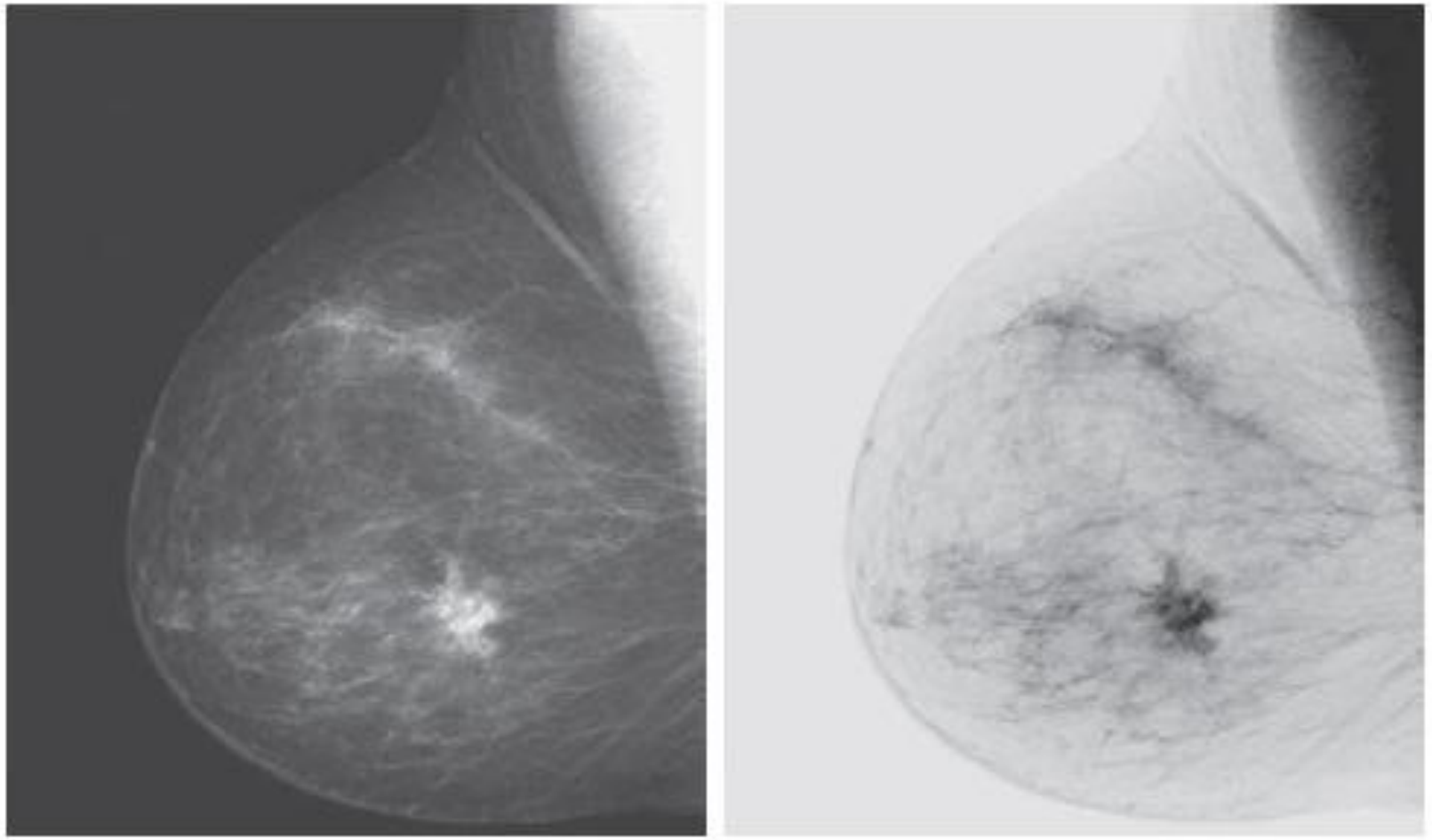
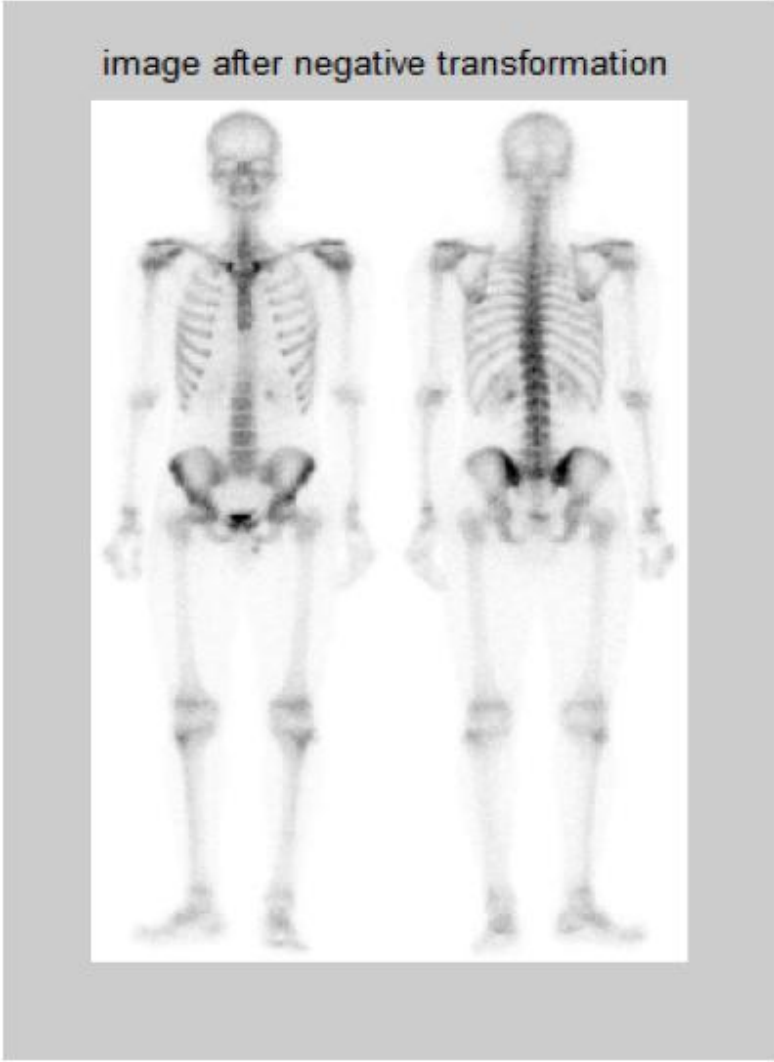
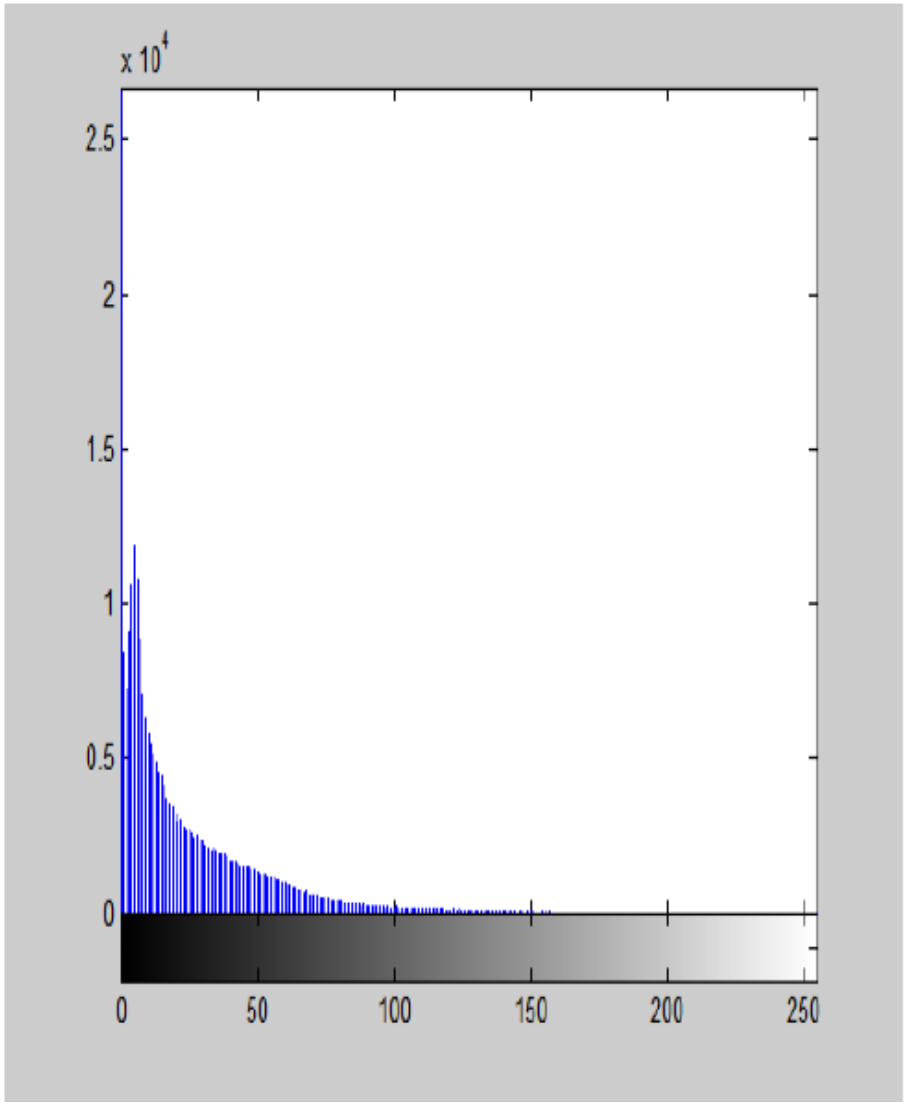


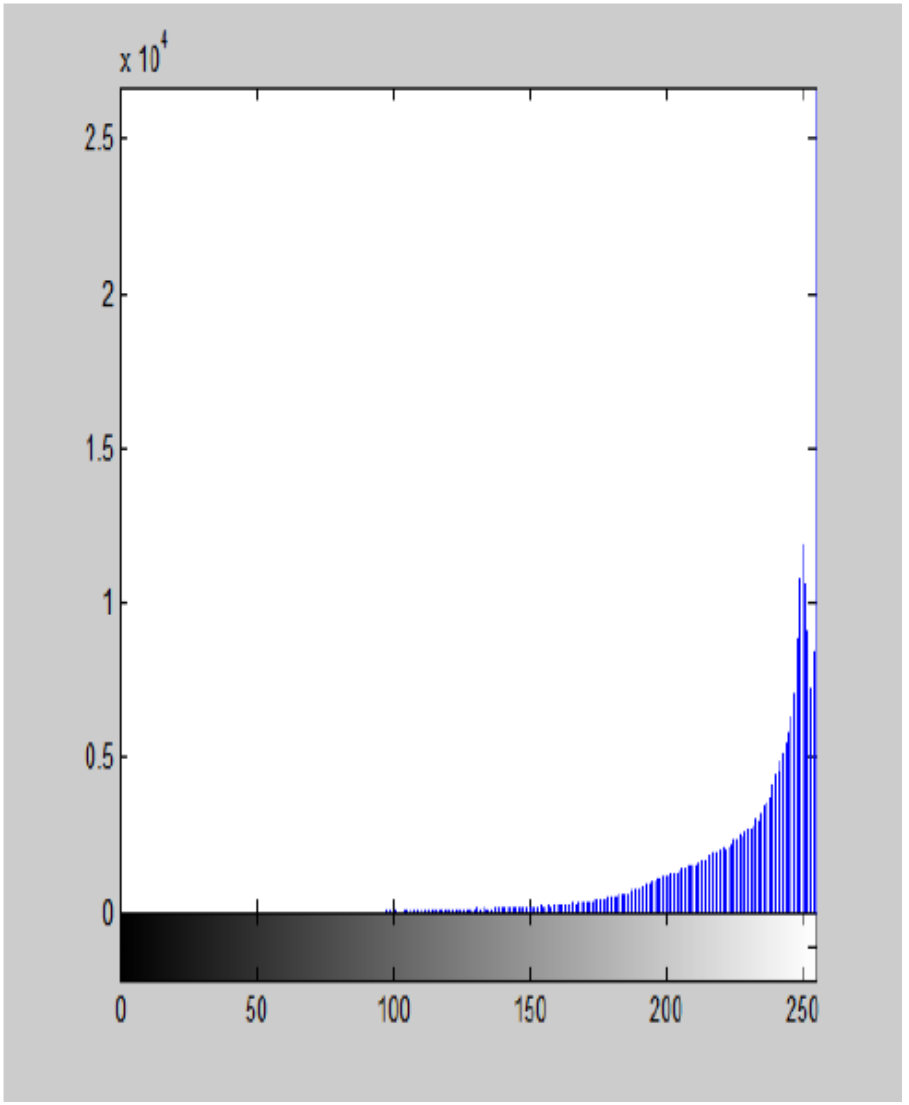
Image Negative



Slide credit: Ashish Ghosh

Image Negative





Slide credit: Ashish Ghosh

Image Negative

Slide credit: Ashish Ghosh

Basic Intensity Transformation Functions

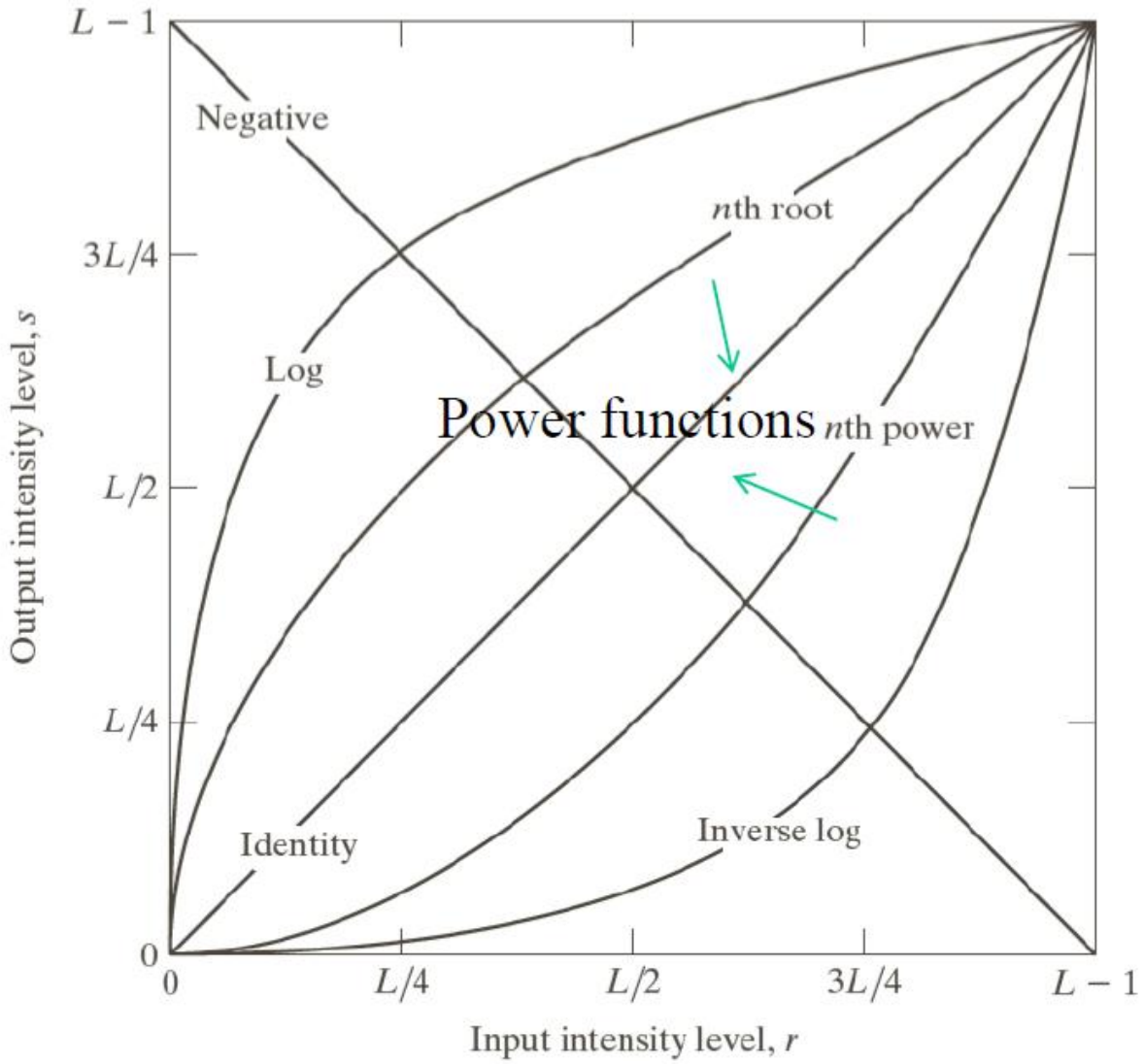


FIGURE 3.3 Some basic intensity transformation functions. All curves were scaled to fit in the range shown.

Slide credit: Yan Tong

Log/Inverse Log Transformation Functions

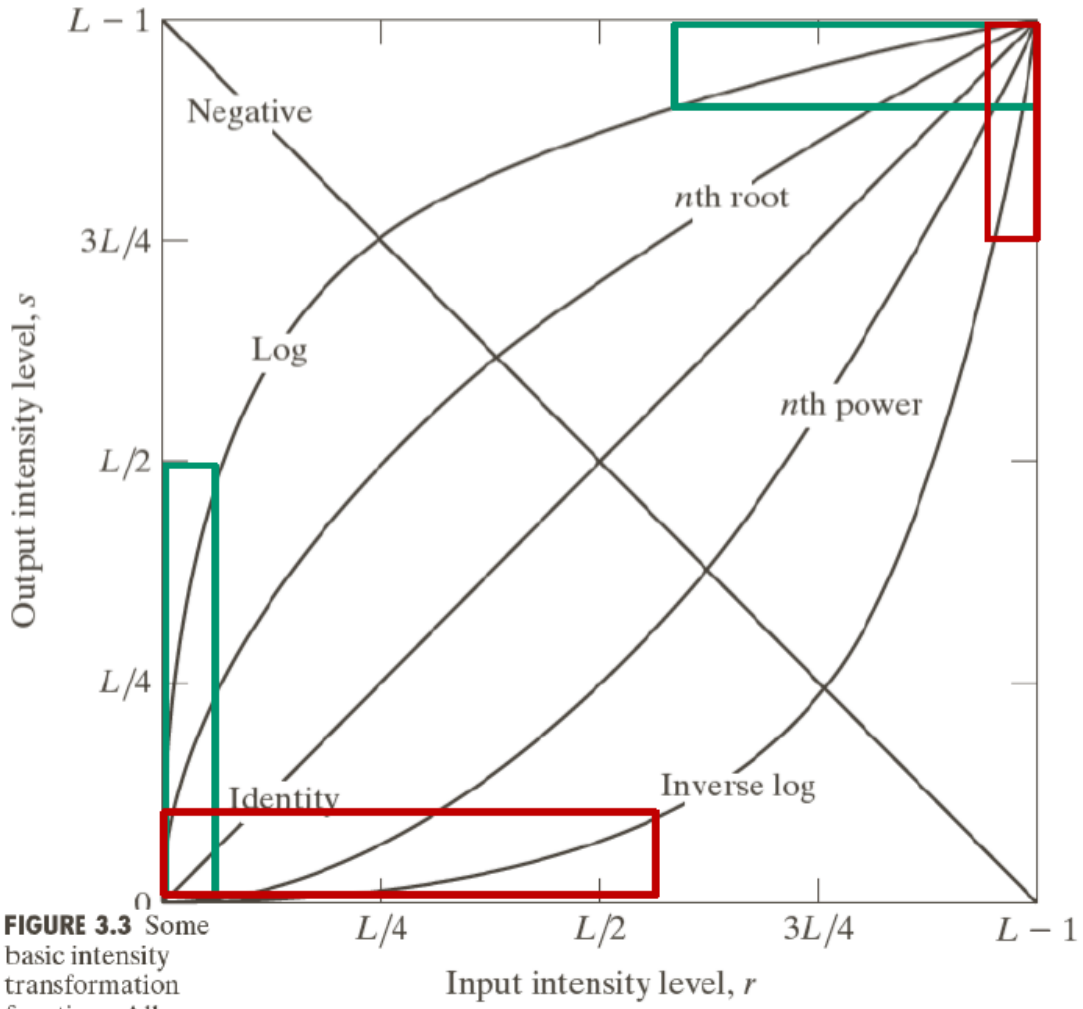


FIGURE 3.3 Some basic intensity transformation functions. All curves were scaled to fit in the range shown.

Log function:

$$s = c\log(1 + r) \quad r \geq 0$$

Stretch low intensity levels
Compress high intensity levels

Inverse log function:

$$s = c\log^{-1}(r)$$

Stretch high intensity levels
Compress low intensity levels

Slide credit: Yan Tong

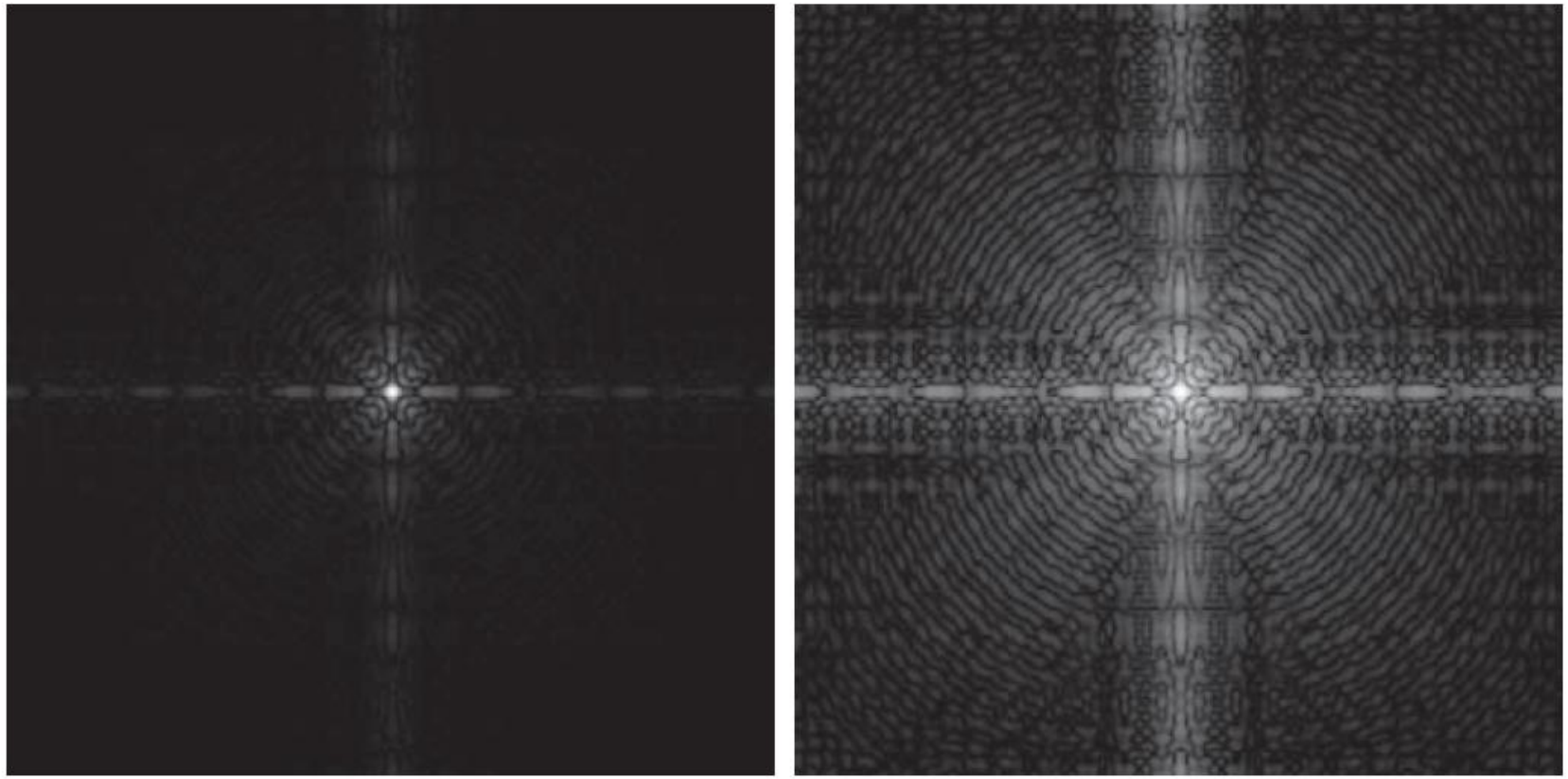
Log Transformations

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

where c is a constant and it is assumed that $r >= 0$

a b

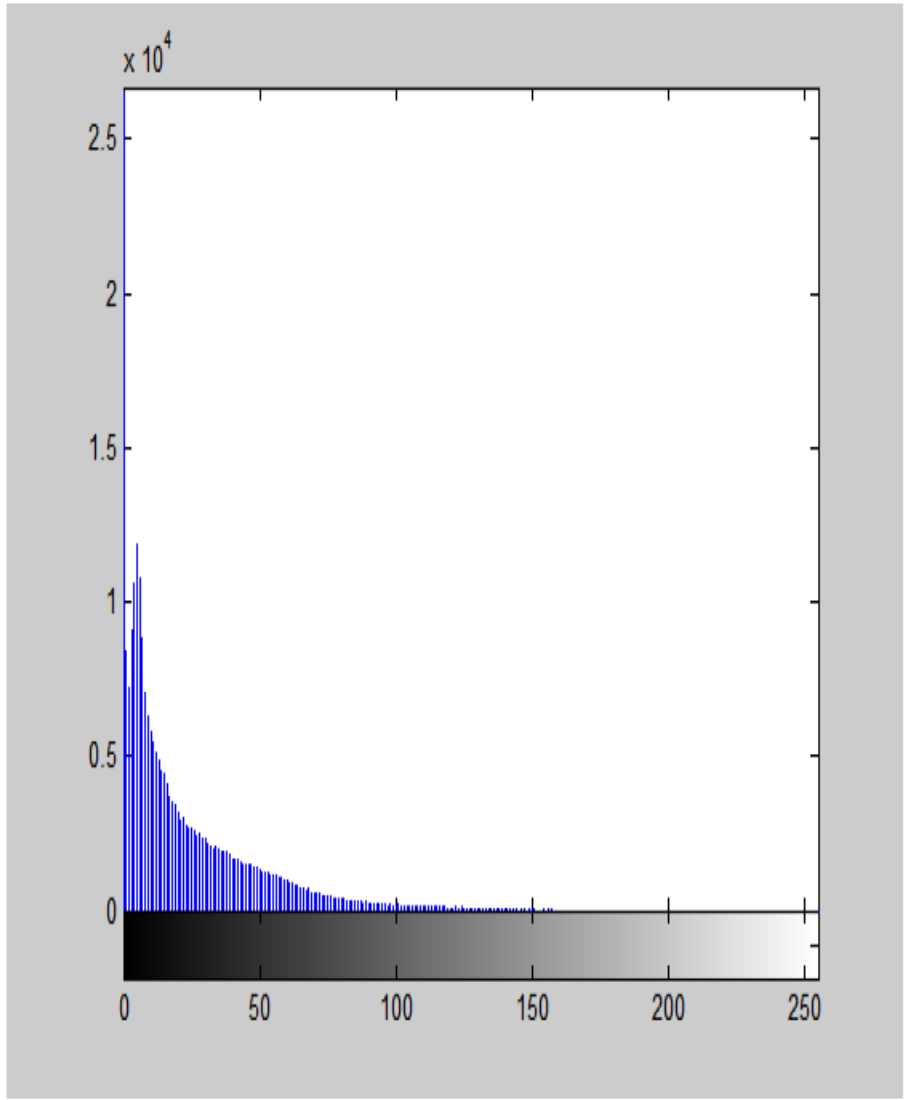
FIGURE 3.5
(a) Fourier spectrum displayed as a grayscale image. (b) Result of applying the log transformation in Eq. (3-4) with $c = 1$. Both images are scaled to the range [0, 255].

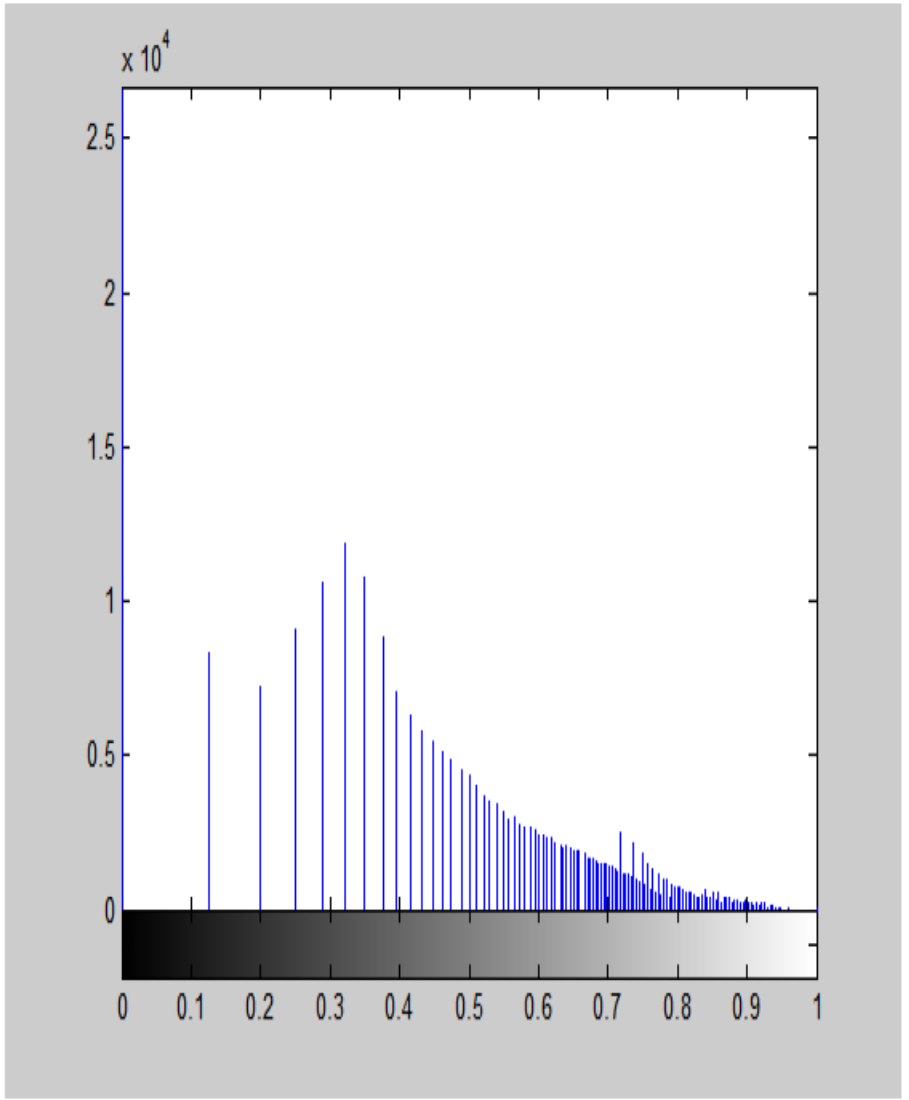


Log Transformations

Slide credit: Ashish Ghosh

Log Transformations

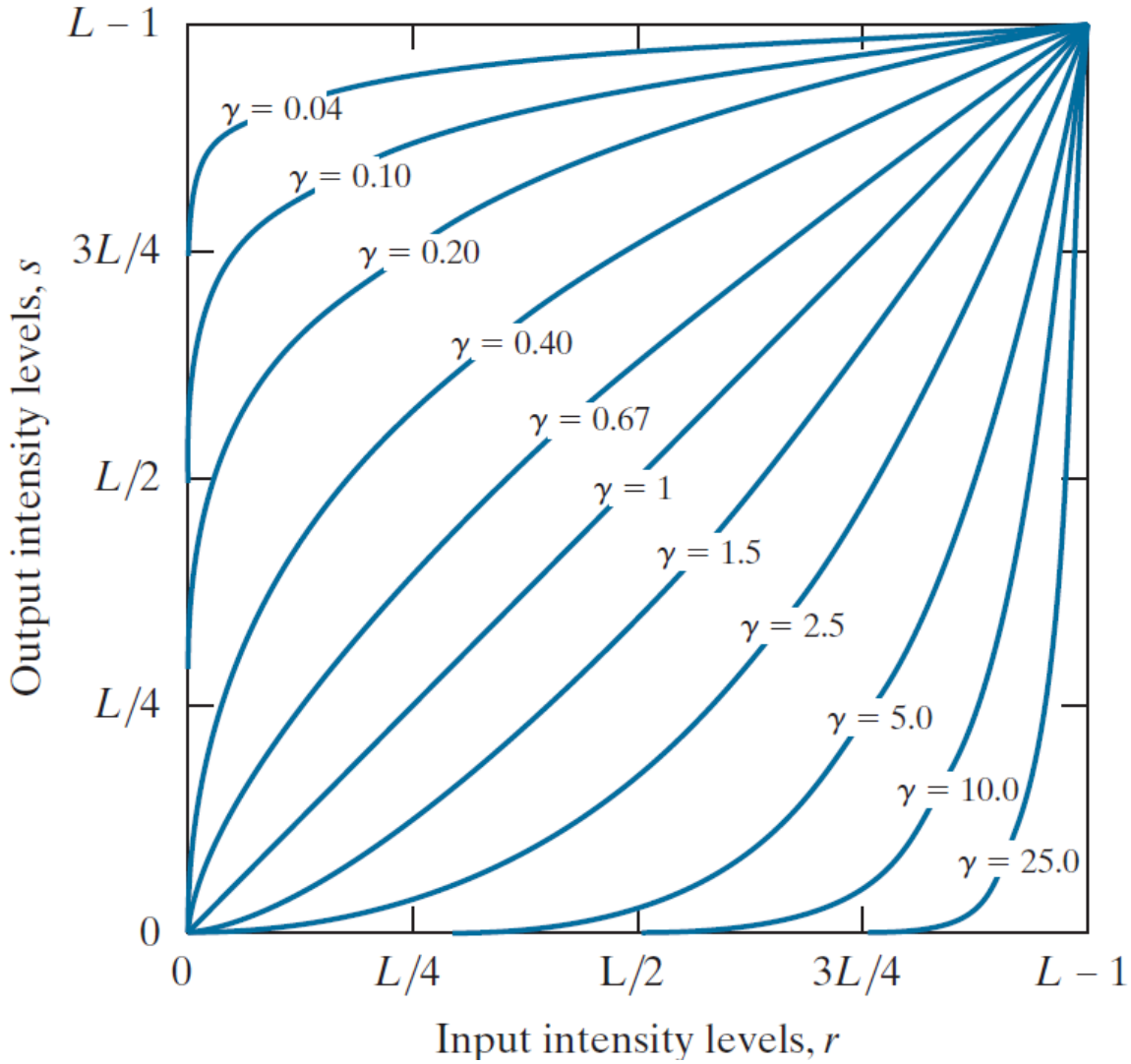




Slide credit: Ashish Ghosh

Power-Law (Gamma) Transformations

FIGURE 3.6
Plots of the gamma equation $s = cr^{\gamma}$ for various values of γ ($c = 1$ in all cases). Each curve was scaled *independently* so that all curves would fit in the same graph. Our interest here is on the *shapes* of the curves, not on their relative values.



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

- More versatile than log transformation
- Performed by a lookup table

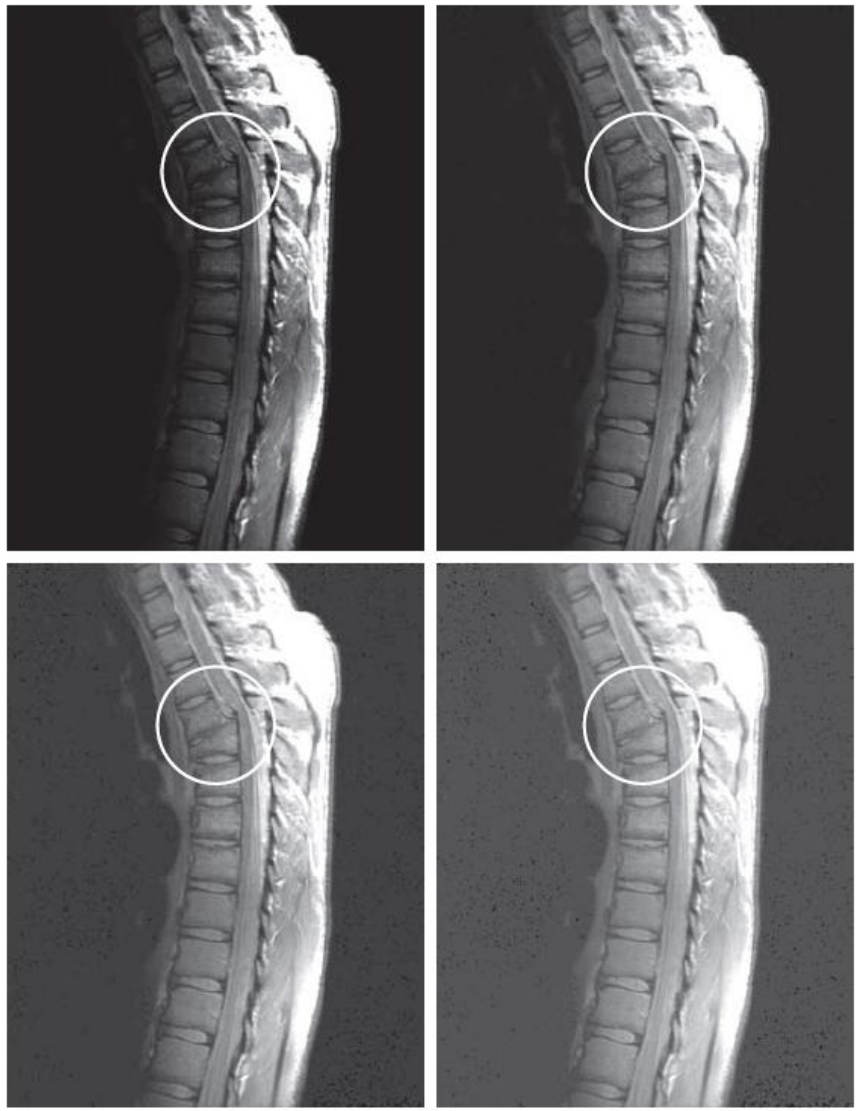
Power-Law (Gamma) Transformations

Power-law curves with fractional values of \gamma map a narrow range of dark input values into a wider range of output values, with the opposite being true for higher values of input levels.

Power-Law (Gamma) Transformations

a b
c d

FIGURE 3.8
(a) Magnetic resonance image (MRI) of a fractured human spine (the region of the fracture is enclosed by the circle).
(b)–(d) Results of applying the transformation in Eq. (3-5) 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.)



The fracture is visible in the region highlighted by the circle. Because the image is predominantly dark, an expansion of intensity levels is desirable. This can be accomplished using a power-law transformation with a fractional exponent.

Power-Law (Gamma) Transformations

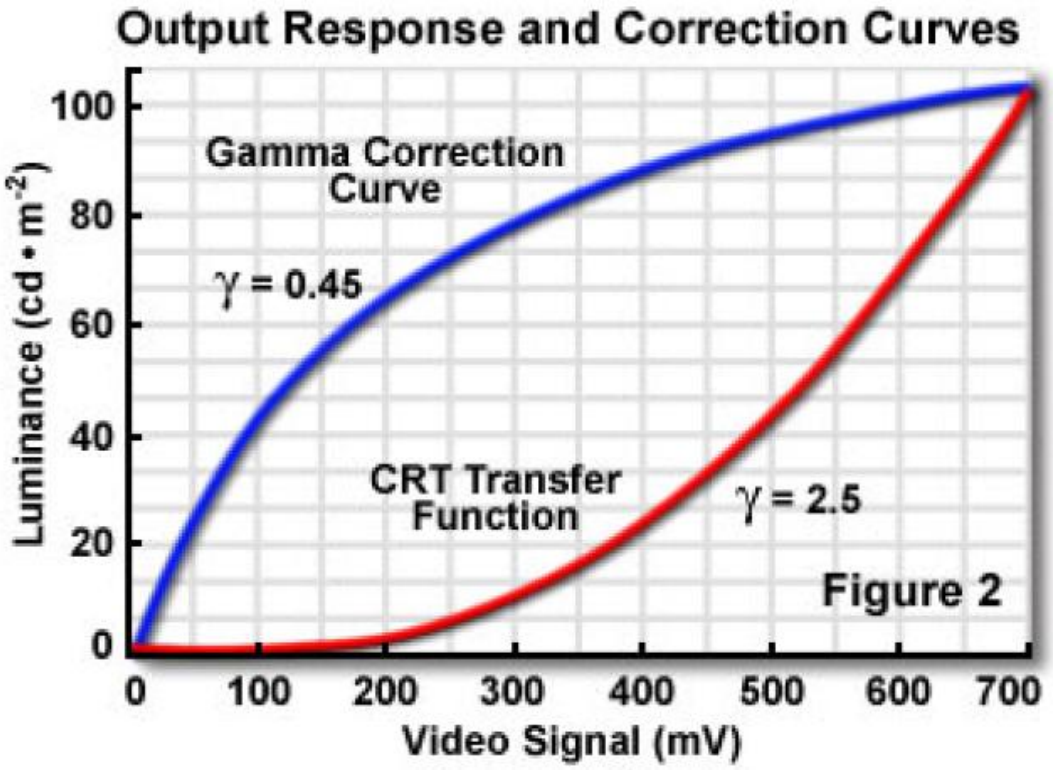
a b
c d

FIGURE 3.9
(a) Aerial image. (b)–(d) Results of applying the transformation in Eq. (3-5) with $\gamma = 3.0$, 4.0, and 5.0, respectively. ($c = 1$ in all cases.) (Original image courtesy of NASA.)

The image to be processed now has a washed-out appearance, indicating that a compression of intensity levels is desirable.

Gamma Correction

- If we take the image file and turn each pixel value into a ***voltage*** and feed it into a CRT, it will be observed that the CRT does not give an amount of light proportional to the voltage.



Slide credit: Ashish Ghosh

Gamma Correction

- The amount of light coming from the phosphor in the screen depends on the voltage according to the relation:

```
Light_output = Voltage ^ CRT_gamma
```

- So if the image is fed to the CRT as it is, it will look much too dark. To fix this we need to "gamma correct" the image first.

- Do the opposite of what the CRT will do to the image, so that things cancel out. So we must feed this image:

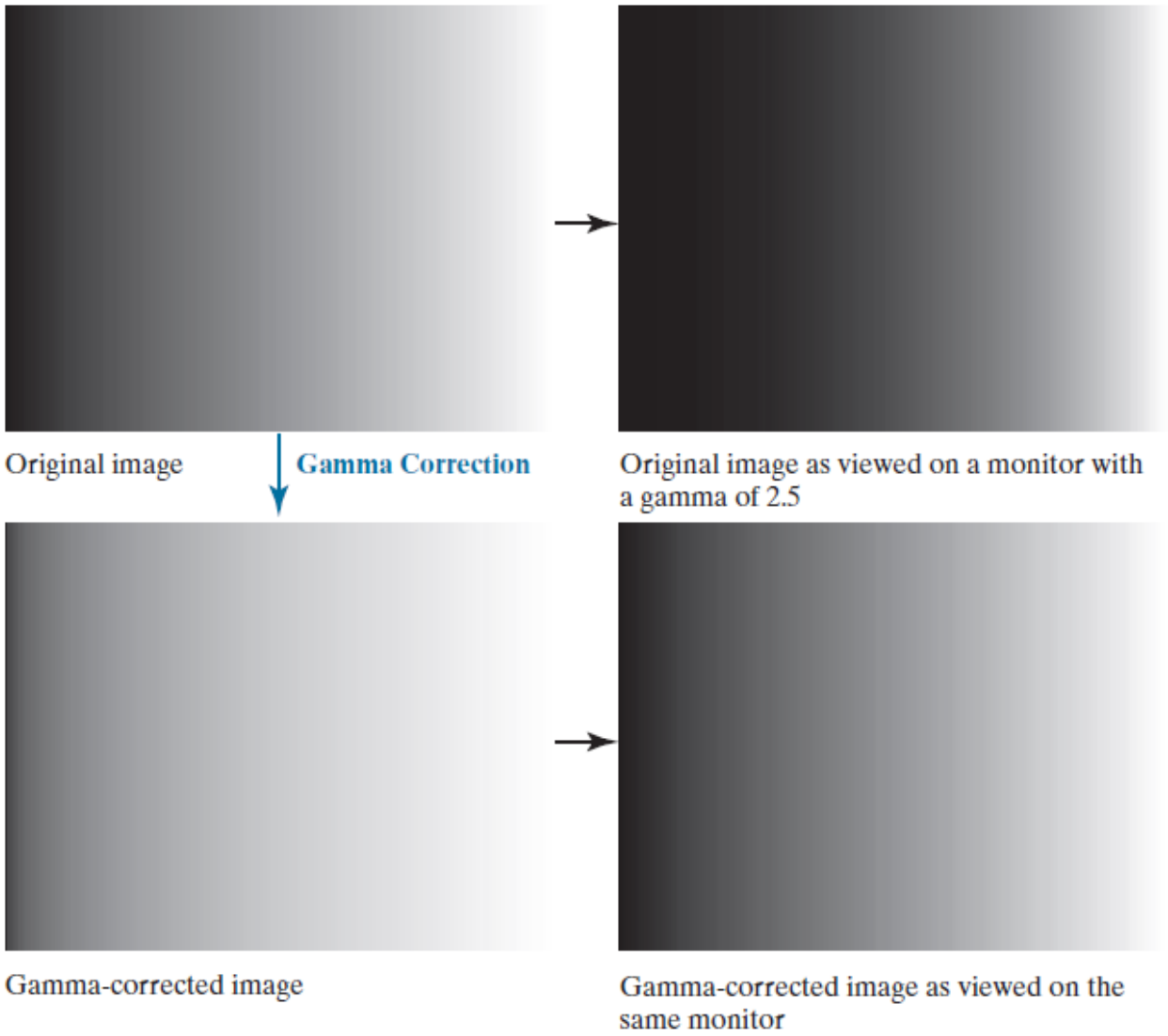
```
Gamma_Corrected_Image = Image ^ (1/CRT_gamma)
```

Slide credit: Ashish Ghosh

Power-Law (Gamma) Transformations

a b
c d

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).



Monitors have an intensity-to-voltage response with a power function

$$s = r^{1/2.5}$$

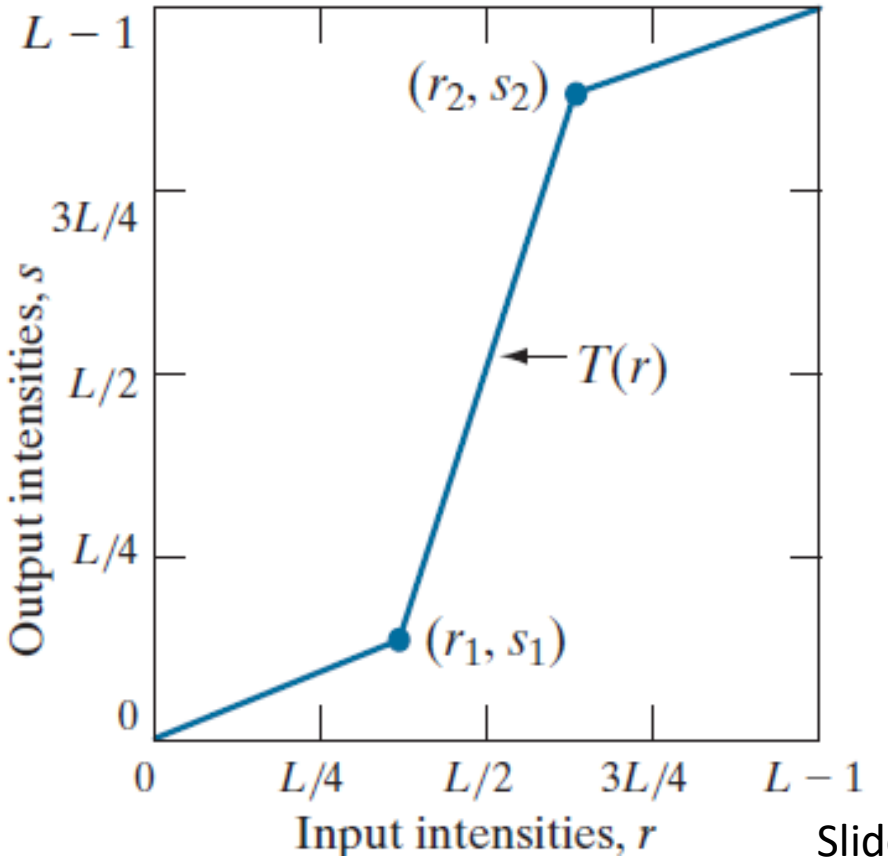
Piecewise-Linear Transformation Functions: Contrast Stretching

Contrast Stretching is used to increase the dynamic range of the gray levels in the image being processed.

The locations of (r_1,s_1) and (r_2,s_2) control the shape of the transformation function.

It is generally assumed that $r_1 \leq r_2$ and $s_1 \leq s_2$.

Let us consider some examples.

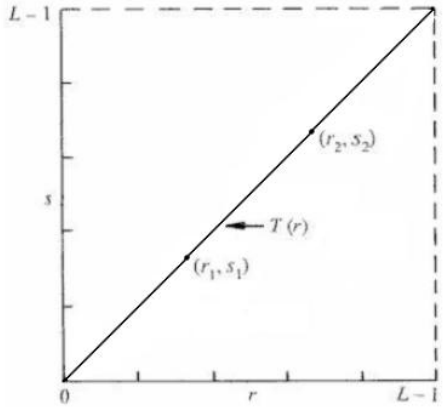


Slide credit: Ashish Ghosh

Piecewise-Linear Transformation Functions: Contrast Stretching

$r_1 = s_1$ $r_2 = s_2$

Transformation is a linear function and produces no changes.

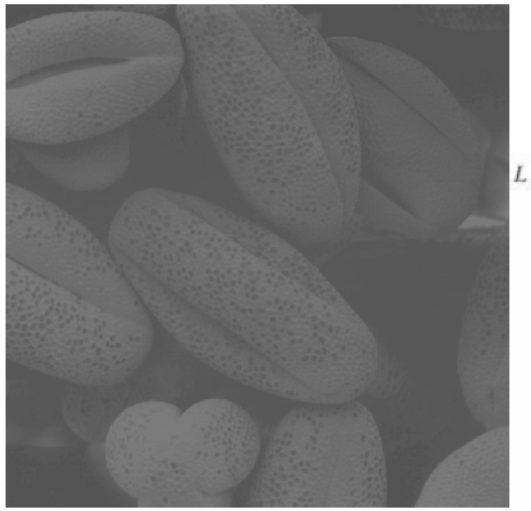


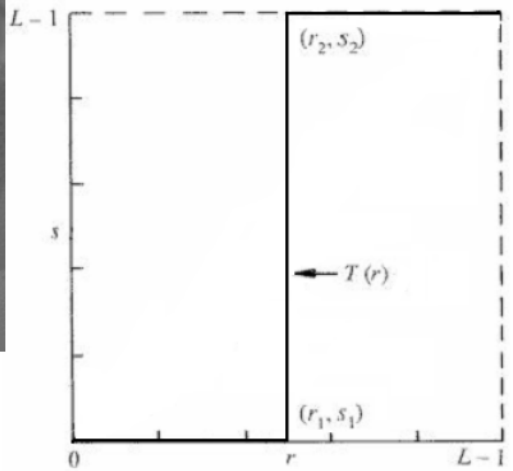
Slide credit: Ashish Ghosh

Piecewise-Linear Transformation Functions: Contrast Stretching

$r_1 = r_2$	$s_1 = 0$	$s_2 = L\text{-}1$

Transformation is a thresholding function creating a binary image.

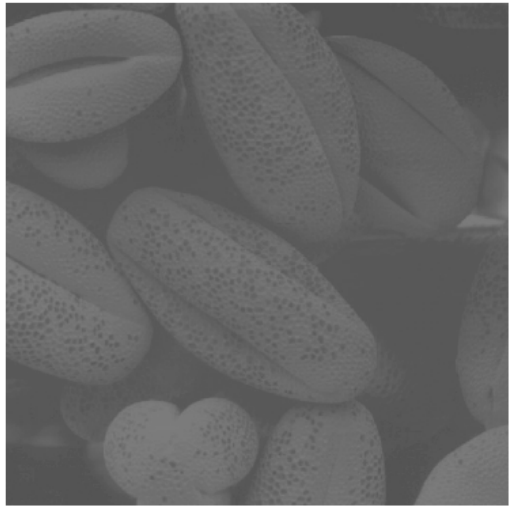


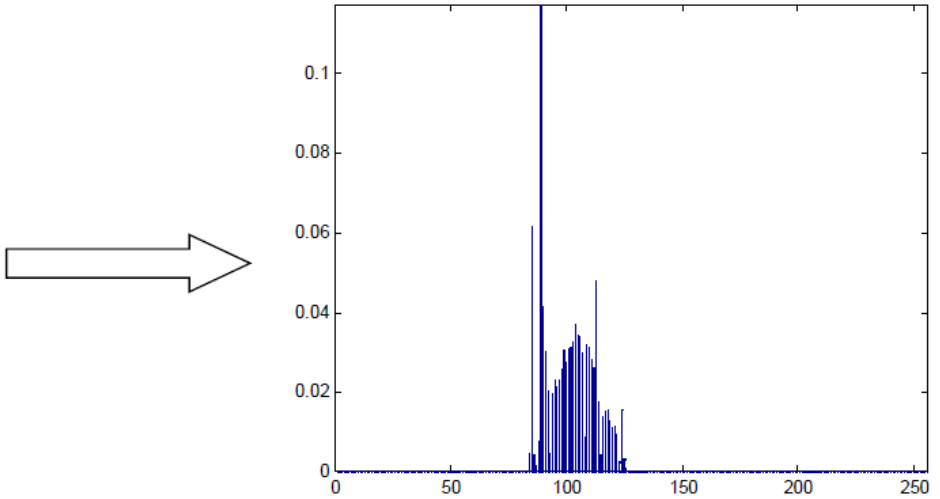


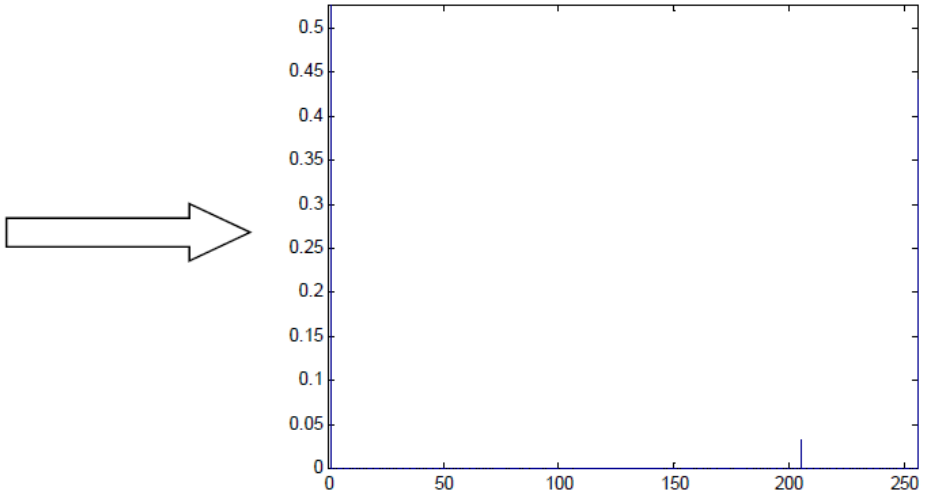
$r_1 = r_2 = 102$

Slide credit: Ashish Ghosh

Piecewise-Linear Transformation Functions: Contrast Stretching



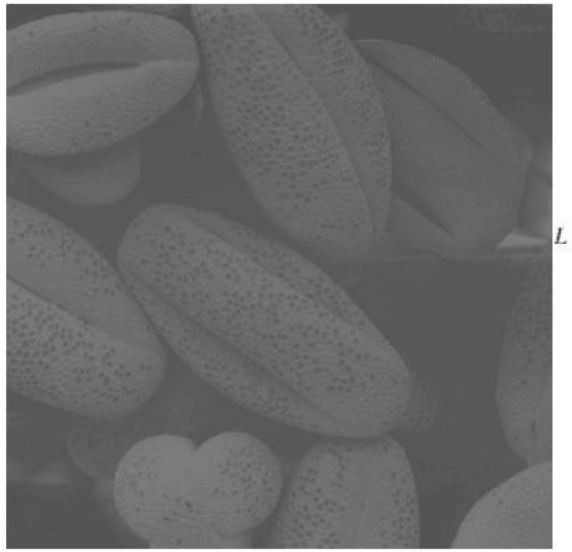


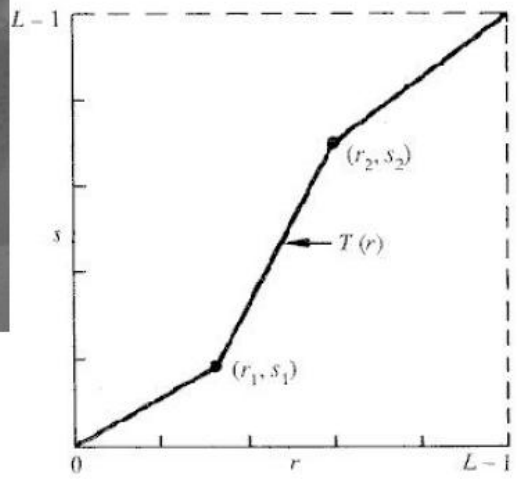


Slide credit: Ashish Ghosh

Piecewise-Linear Transformation Functions: Contrast Stretching

Intermediate values of (r_1,s_1) and (r_2,s_2) produce various degrees of spread in the gray levels of the output image, thus affecting its contrast.

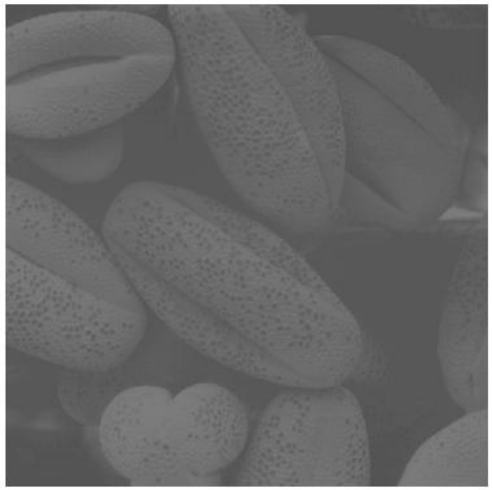


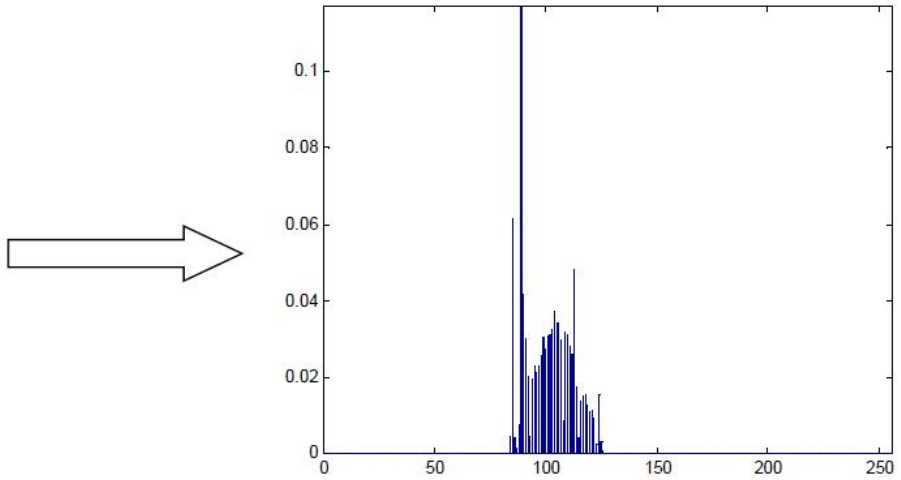


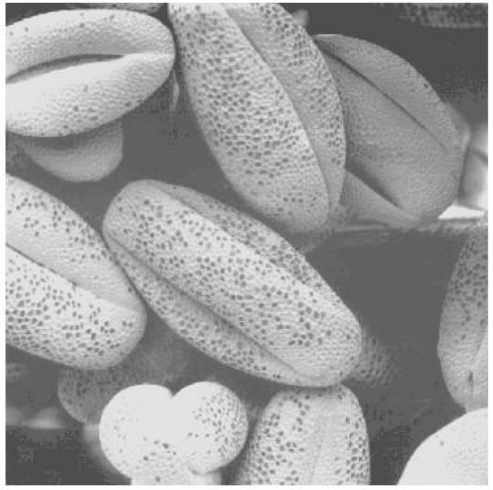
$r_1 = 71$ $r_2 = 115$ $s_1 = 17$ $s_2 = 226$

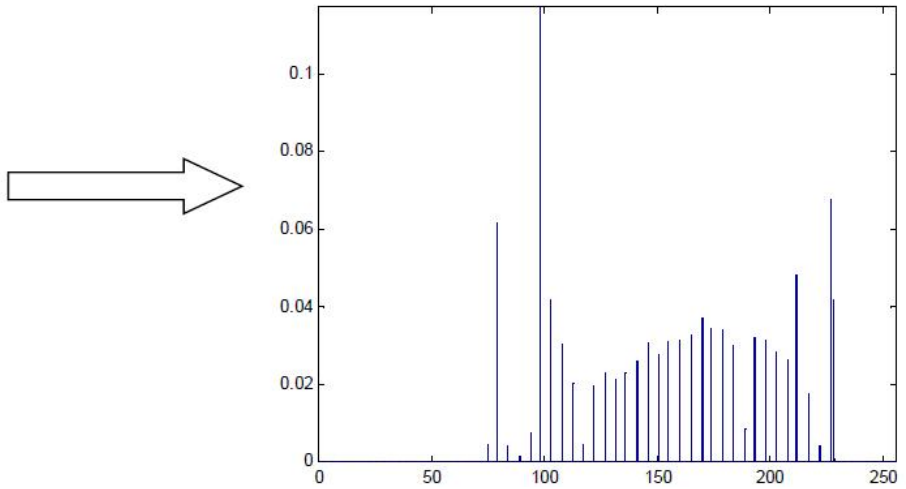
Slide credit: Ashish Ghosh

Piecewise-Linear Transformation Functions: Contrast Stretching







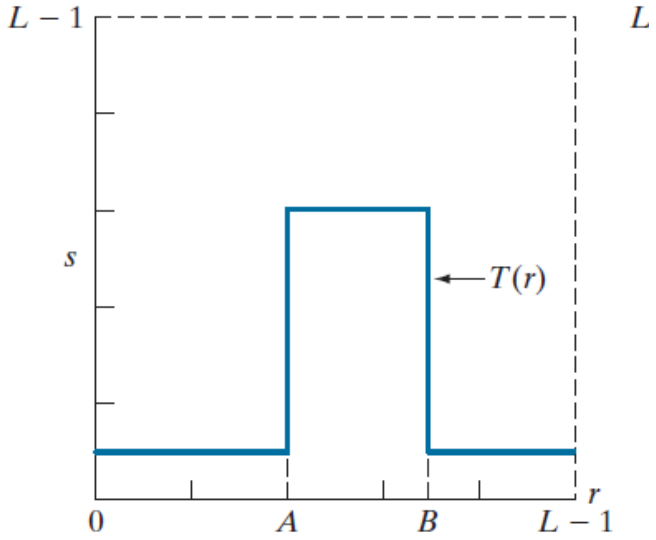


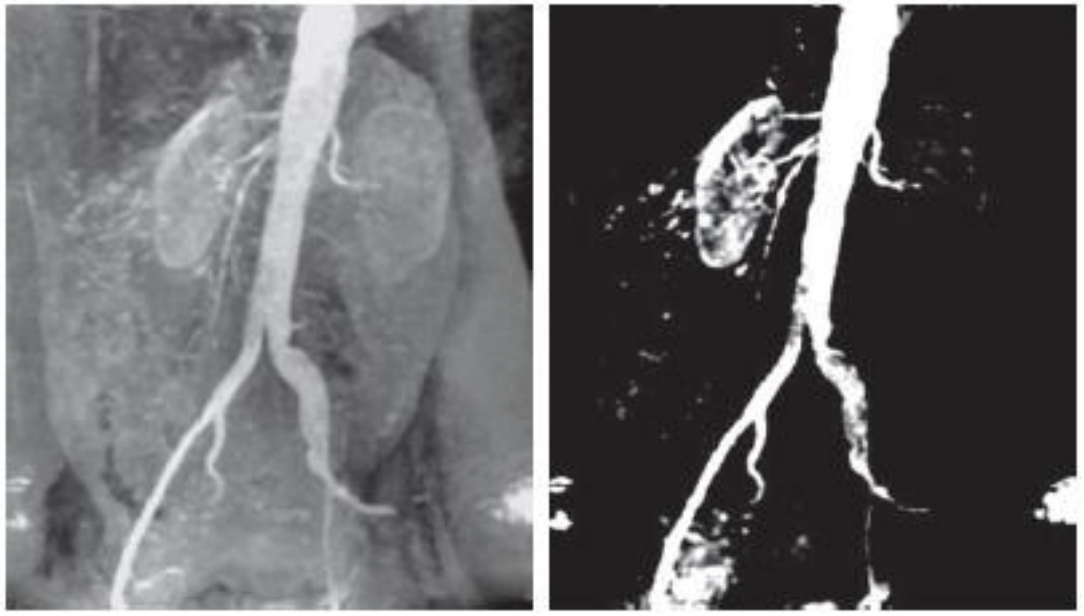
Slide credit: Ashish Ghosh

Piecewise-Linear Transformation Functions: Gray-Level / Intensity-Level Slicing

Gray-level slicing is carried out to highlight a specific range of gray levels in an image to enhance certain features.

- One way is to display a high value for all gray levels in the range of interest and a low value for all other gray levels (binary image).



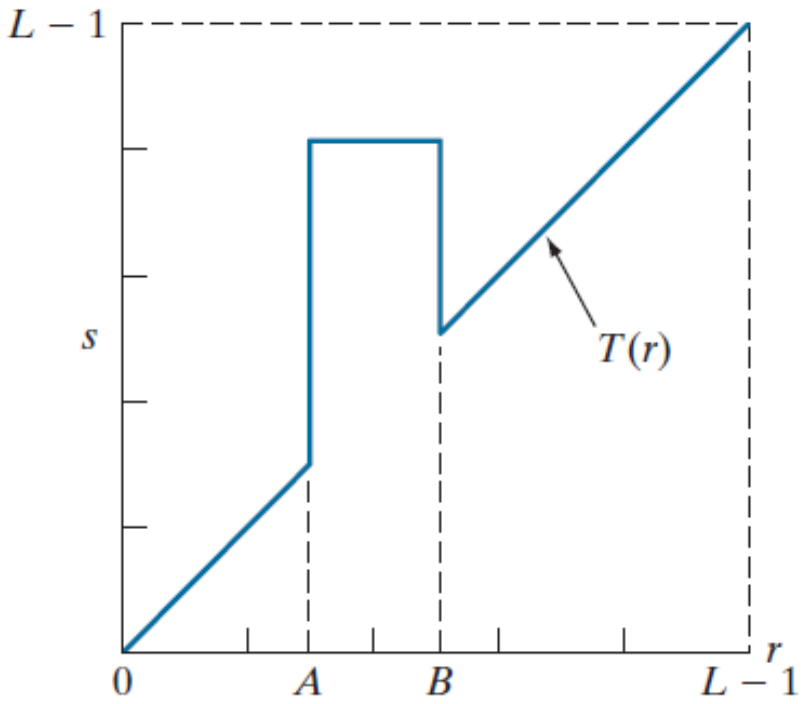


with the range of intensities of interest selected in the upper end of the gray scale

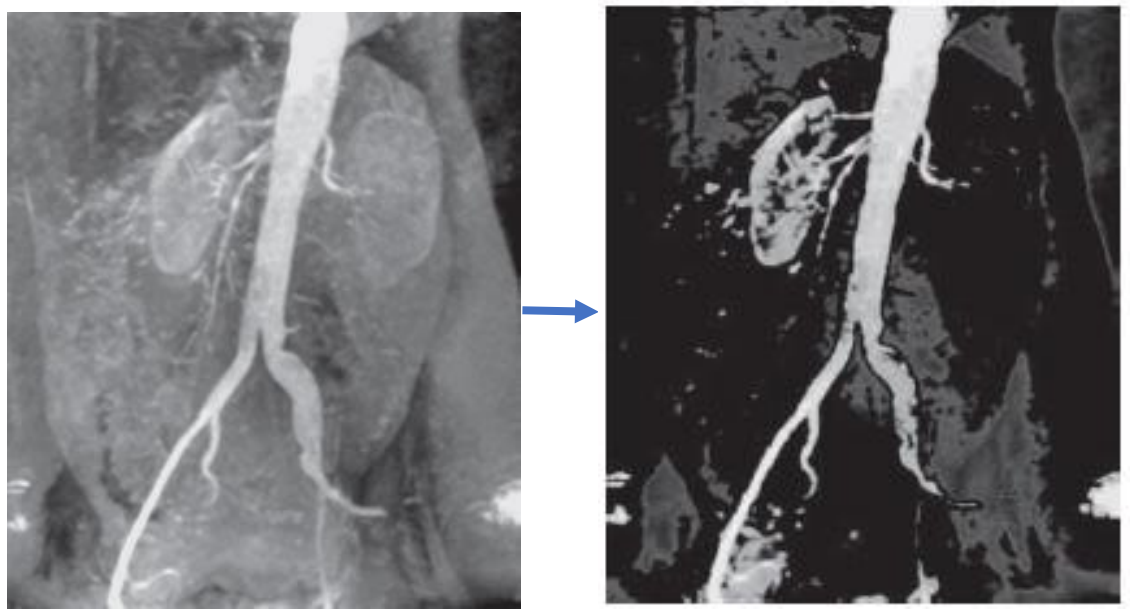
Slide credit: Ashish Ghosh

Piecewise-Linear Transformation Functions: Gray-Level / Intensity-Level Slicing

The second approach is to brighten the desired range of gray levels but preserve the background and gray-level tonalities in the image.



Aortic angiogram



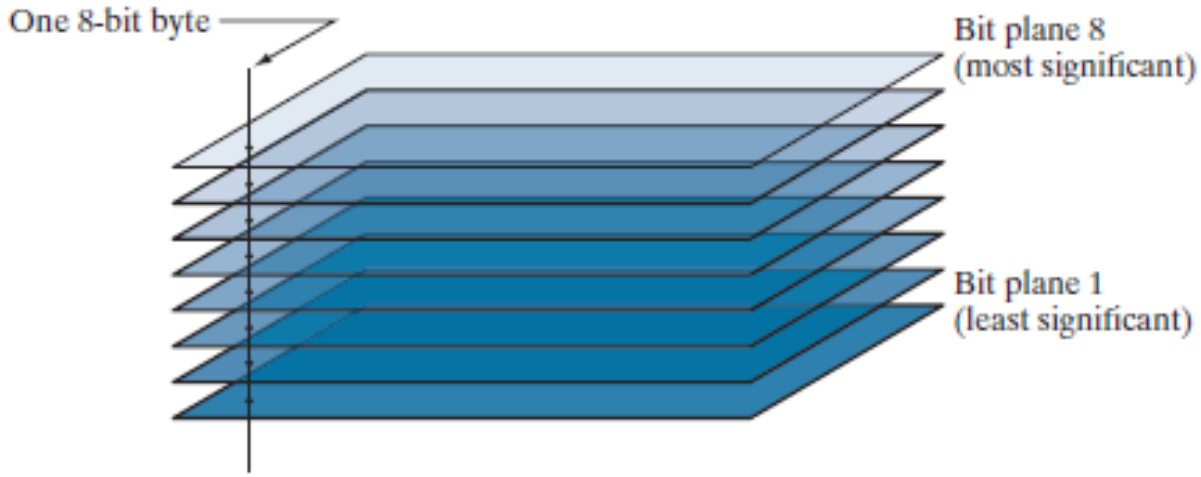
with the selected range set near black, so that the grays in the area of the blood vessels and kidneys were preserved

Piecewise-Linear Transformation Functions: Bit-Plane Slicing

Bit-plane slicing is carried out to highlight the contribution made to the total image appearance by specific bits.

Assuming that each pixel is represented by 8 bits, the image is composed of eight 1-bit planes.

FIGURE 3.13
Bit-planes of an 8-bit image.

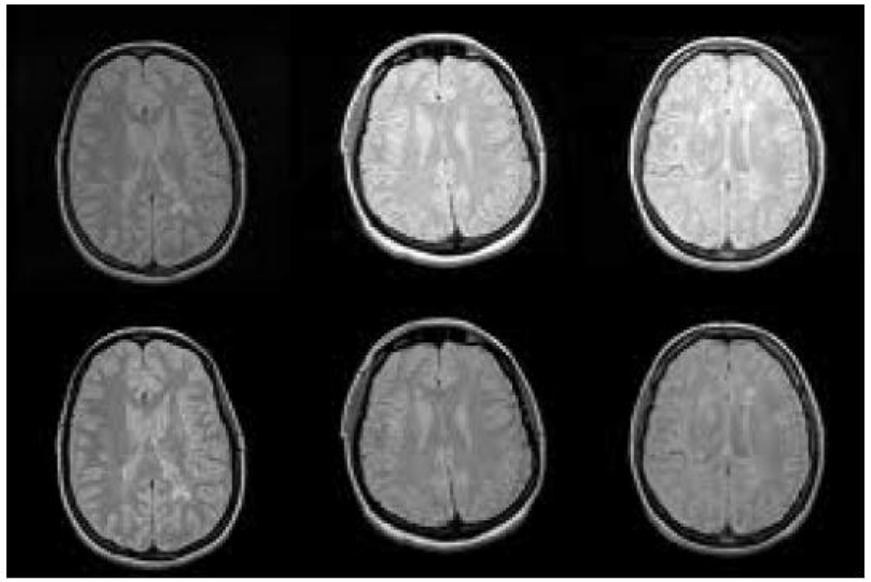


Piecewise-Linear Transformation Functions: Bit-Plane Slicing

Plane 0 contains the ***least*** significant bit and plane 7 contains the ***most*** significant bit.

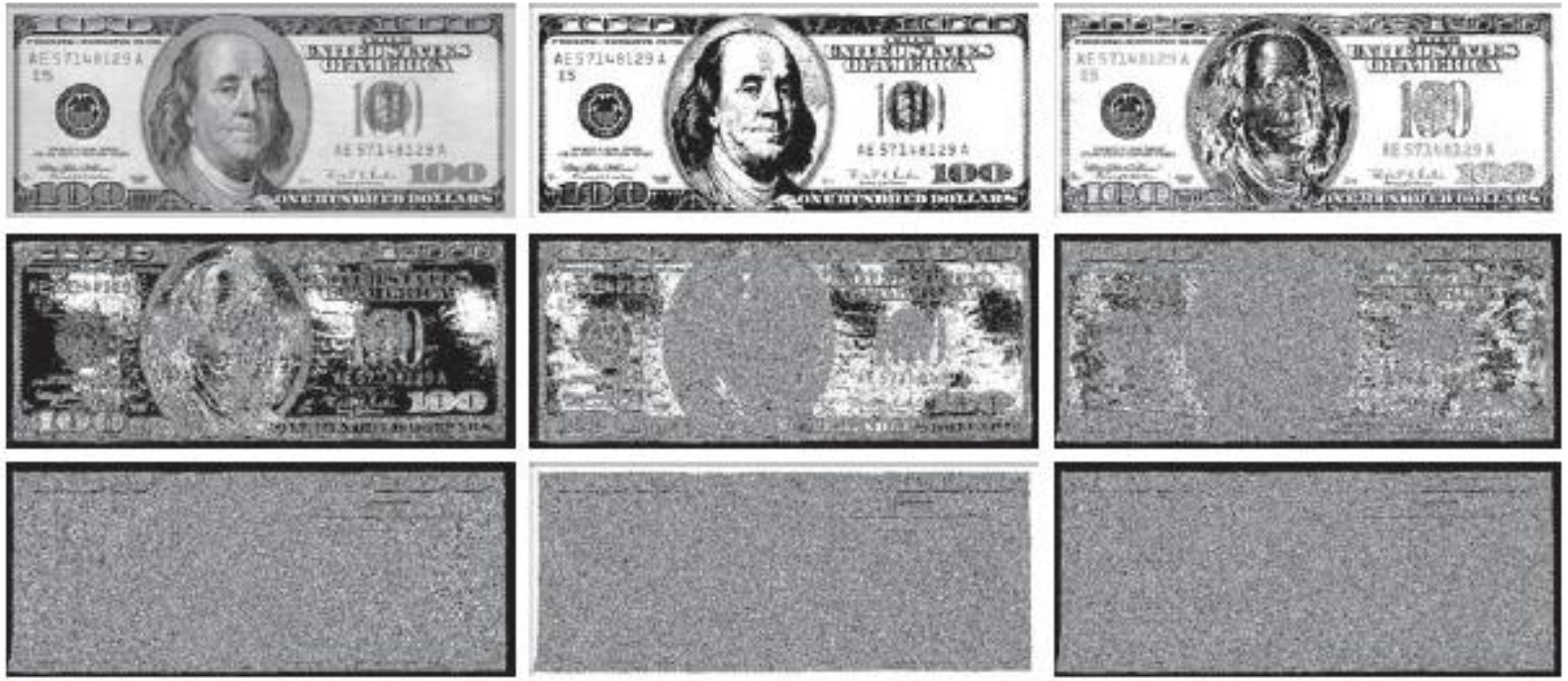
Only the higher order bits (top four) contain visually significant data. The other bit planes contribute the more subtle details.

Plane 7 corresponds exactly with an image thresholded at gray level 128.



Slide credit: Ashish Ghosh

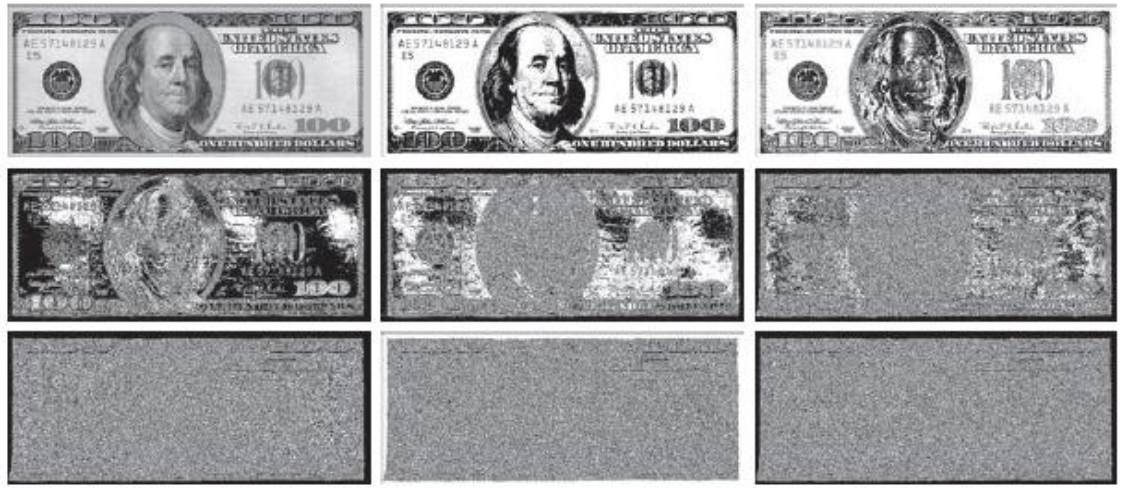
Piecewise-Linear Transformation Functions: Bit-Plane Slicing



a b c
d e f
g h i

FIGURE 3.14 (a) An 8-bit gray-scale image of size 550×1192 pixels. (b) through (i) Bit planes 8 through 1, with bit plane 1 corresponding to the least significant bit. Each bit plane is a binary image..

Piecewise-Linear Transformation Functions: Bit-Plane Slicing



a b c
d e f
g h i

FIGURE 3.14 (a) An 8-bit gray-scale image of size 550×1192 pixels. (b) through (i) Bit planes 8 through 1, with bit plane 1 corresponding to the least significant bit. Each bit plane is a binary image..

Less bit planes are sufficient to obtain an acceptable details, while require half of the storage

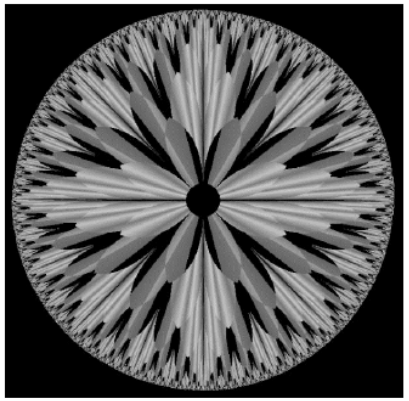
a b c **FIGURE 3.15** Image reconstructed from bit planes: (a) 8 and 7; (b) 8, 7, and 6; (c) 8, 7, 6, and 5.

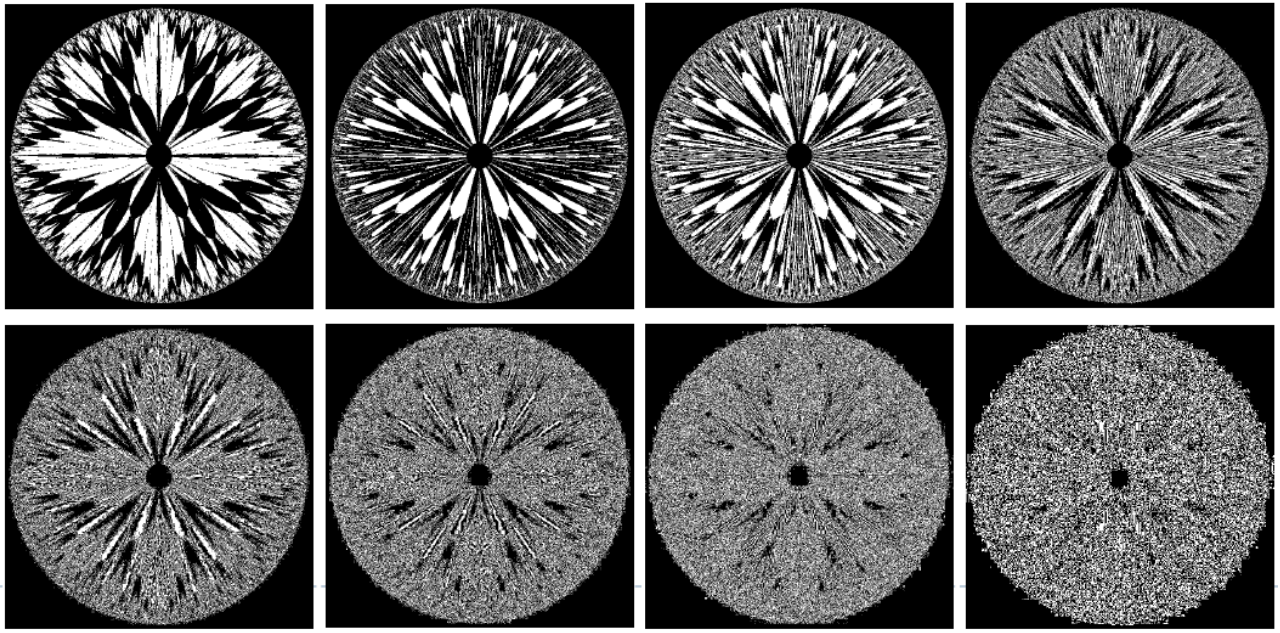
Piecewise-Linear Transformation Functions: Bit-Plane Slicing

Here is a fractal image.

Let us see it's constituent bit-planes.

The MSB contains the most information while the LSB contains the least amount of information.



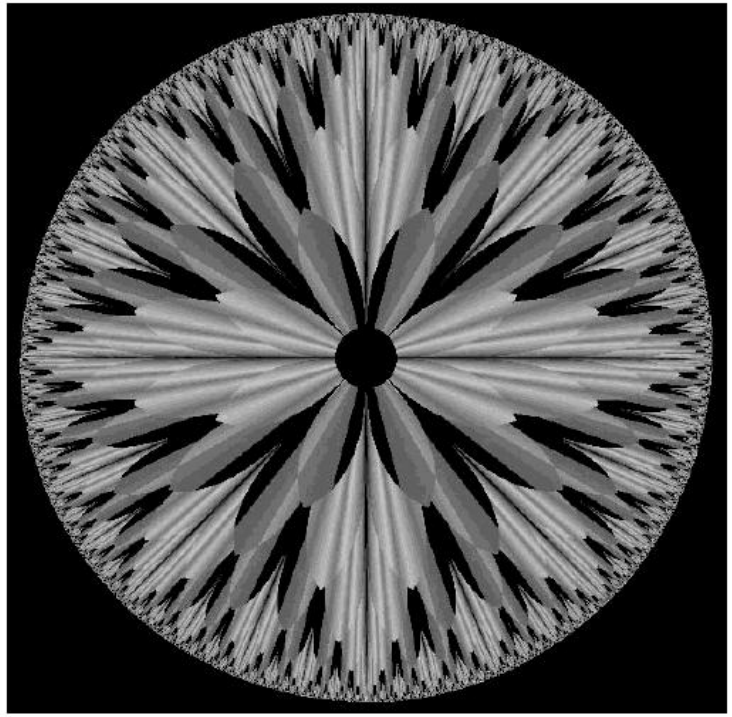


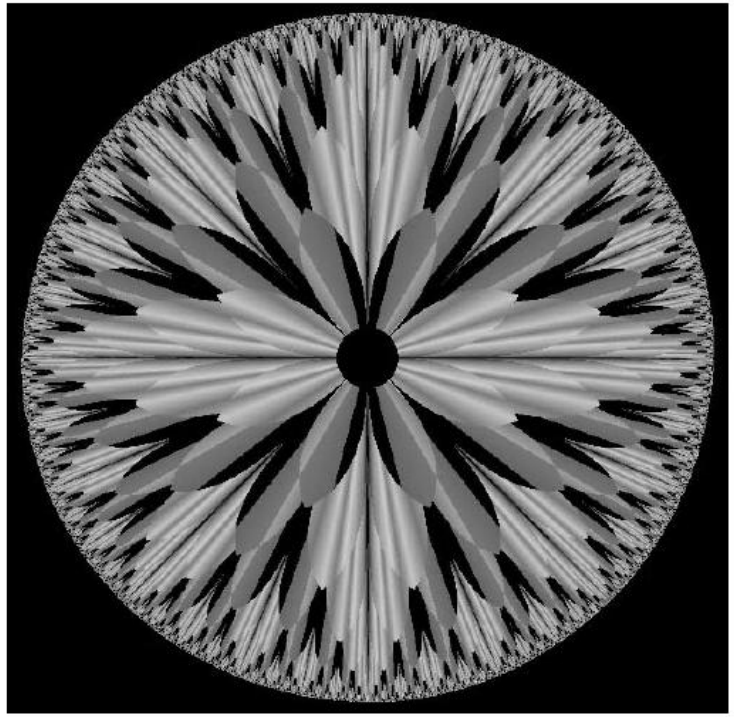
Slide credit: Ashish Ghosh

Piecewise-Linear Transformation Functions: Bit-Plane Slicing

Let us take the look at the effect of removing the last 4 bits .

Now compare this with the original image.





Slide credit: Ashish Ghosh

Histogram Processing

The histogram of a digital image with gray levels from 0 to L-1 is a discrete function $h(r_k)=n_k$, where:

- r_k is the k^{th} gray level
- n_k is the number of pixels with that gray level, and
- n is the total number of pixels in the image

Here, k = 0, 1, 2, …, L-1

The normalized histogram is obtained by dividing the n_k by n. Thus $p(r_k) = n_k/n$

The plot of $p_r(r_k)$ versus r_k is called a histogram.

Histograms are the basis for numerous spatial domain techniques.

Slide credit: Ashish Ghosh

Histogram Processing

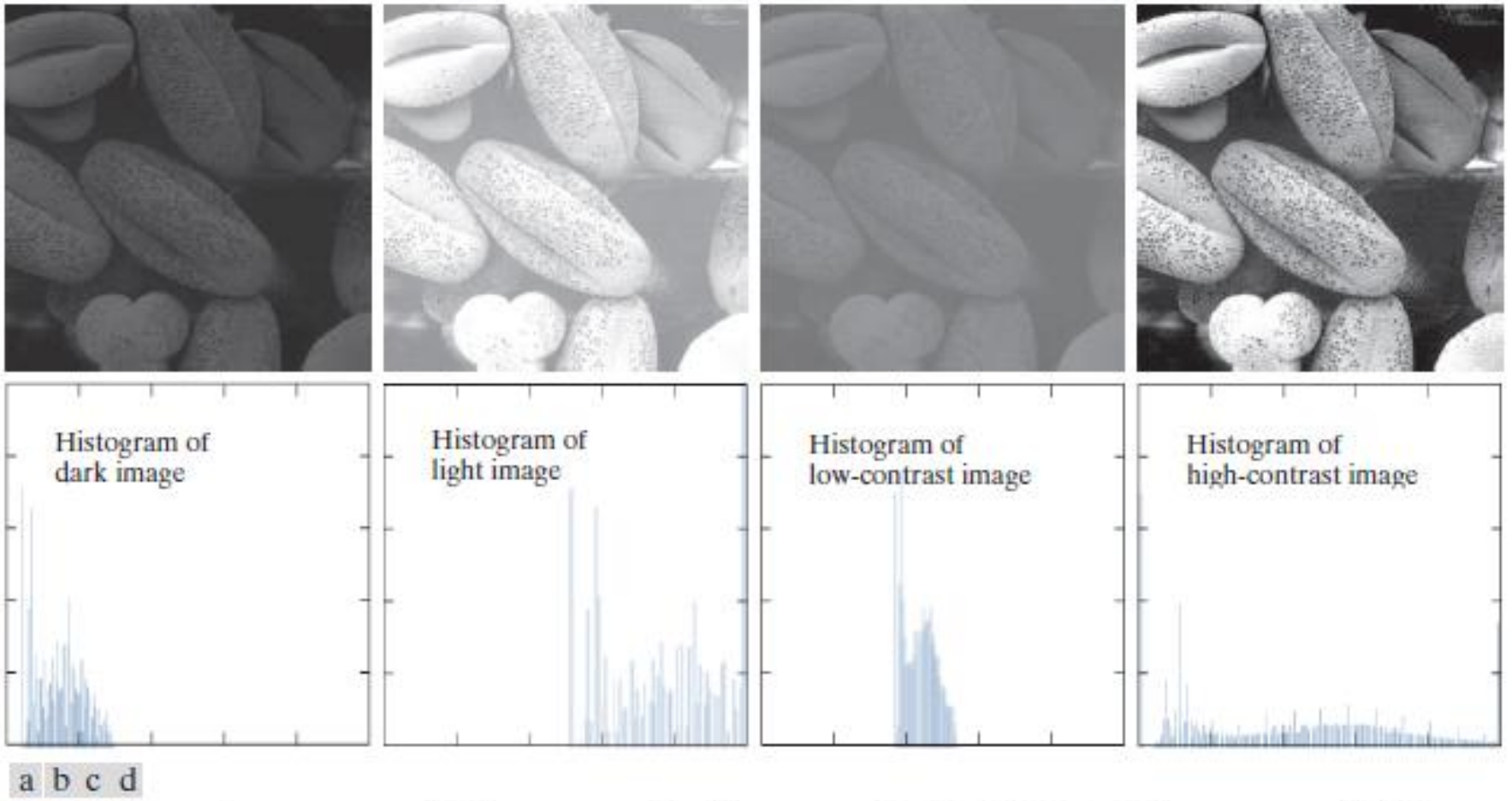


FIGURE 3.16 Four image types and their corresponding histograms. (a) dark; (b) light; (c) low contrast; (d) high contrast. The horizontal axis of the histograms are values of r_k and the vertical axis are values of $p(r_k)$.

Histogram

$$h(r_k) = n_k$$

Normalized histogram

$$p(r_k) = n_k / MN$$

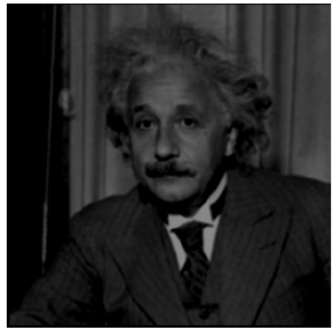
$$\sum_{k=0}^{255} p(r_k) = 1$$

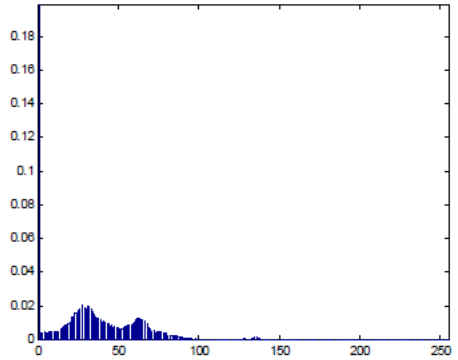
Histogram Processing

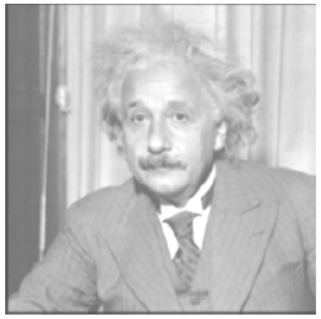
The shape of the histogram provides useful information

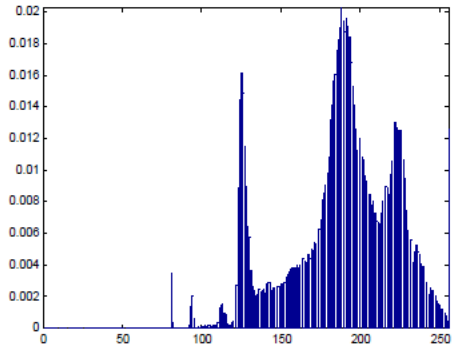
In the case of a dark image, the components of the histogram are concentrated on the lower side of the gray scale.

Similarly the components of the histogram of a bright image are concentrated on the higher side of the gray scale.









Slide credit: Ashish Ghosh

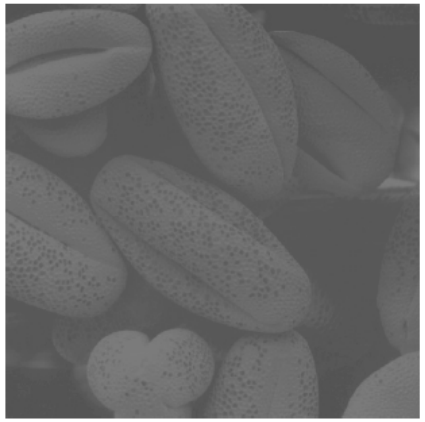
Histogram Processing

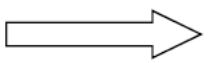
Slide credit: Ashish Ghosh

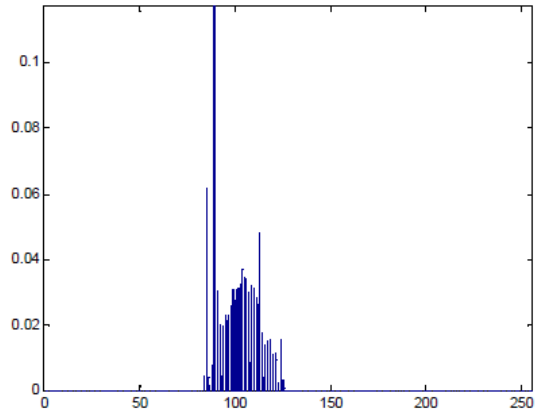
Histogram Processing

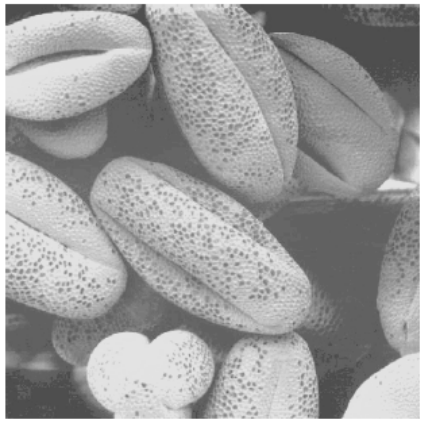
In the case of a low contrast image the histogram is narrow and centered towards the middle of the gray scale.

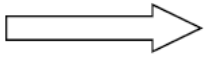
It is spread out in the case of a high contrast image.

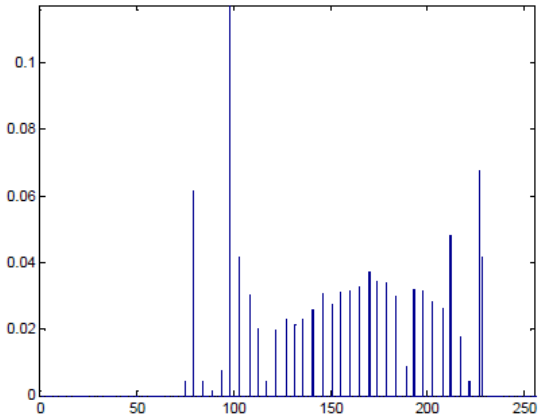












Slide credit: Ashish Ghosh

Histogram Processing

Major histogram processing types are:

Histogram equalization

Histogram matching (specification)

Local enhancement

Slide credit: Ashish Ghosh

Histogram Equalization

- Histogram equalization:
  - To improve the contrast of an image.
  - To transform an image in such a way that the transformed image. has a nearly uniform distribution of pixel values.

Slide credit: Ashish Ghosh

Histogram Equalization: Transformation Function

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

A valid transformation function must satisfy two conditions:

(a) $T(r)$ is monotonically increasing, i.e., $T(r_1) \ge T(r_2)$ if $r_1 > r_2$

(b) $0 \le T(r) \le L-1$ The same range as input

(a') $T(r)$ is strictly monotonic : one - to - one mapping $r = T'(s)$

Slide credit: Yan Tong

Histogram Equalization: Transformation Function

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

A valid transformation function must satisfy two conditions:

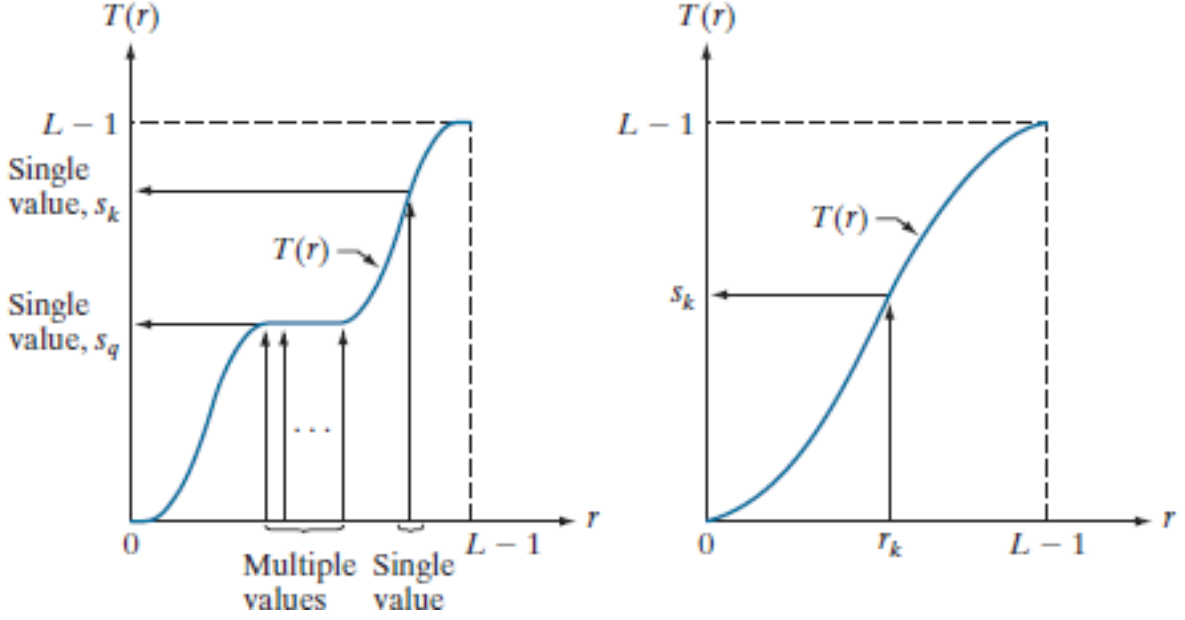
(a) $T(r)$ is monotonically increasing, i.e., $T(r_1) \ge T(r_2)$ if $r_1 > r_2$

(b) $0 \le T(r) \le L-1$

(a') $T(r)$ is strictly monotonic : one - to - one mapping $r = T'(s)$

a b

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



Slide credit: Yan Tong

Histogram Equalization: Transformation Function

If $T(r)$ is continuous and differentiable over the range of r, then

$$p_s(s) = p_r(r)\left|\frac{dr}{ds}\right|$$

Probability density function of intensity value

Slide credit: Yan Tong

Histogram Equalization: Transformation Function

A special transformation function

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

Cumulative distribution function of r

Is it a valid transformation function?

Yes.

(a) $T(r)$ is monotonically increasing, i.e., $T(r_1) \geq T(r_2)$ if $r_1 > r_2$

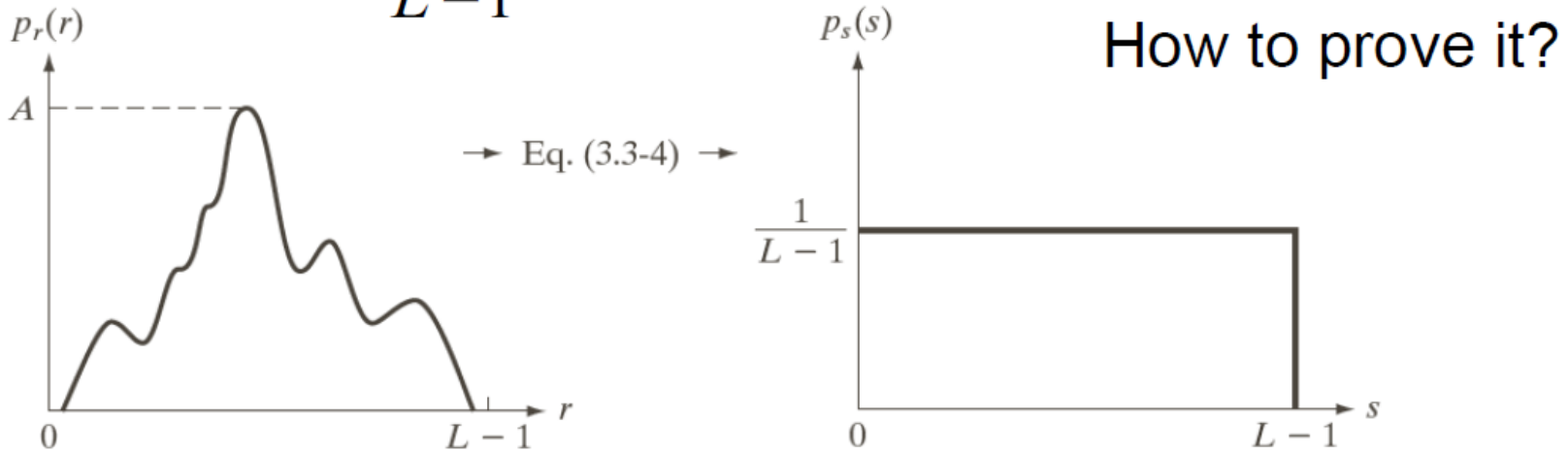
(b) $0 \leq T(r) \leq L-1$

Slide credit: Yan Tong

Histogram Equalization: Transformation Function

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

$$\Rightarrow \quad p_s(s) = \frac{1}{L-1}$$



How to prove it?

a b

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.

Slide credit: Yan Tong

Histogram Equalization: Transformation Function

Proof:

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

$$\frac{ds}{dr} = \frac{dT(r)}{dr}$$

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

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

$$p_s(s) = p_r(r)\left|\frac{dr}{ds}\right|$$

$$= p_r(r)\left|\frac{1}{(L-1)p_r(r)}\right|$$

$$= \frac{1}{L-1} \qquad 0 \le s \le L-1$$

Slide credit: Yan Tong

Histogram Equalization: Transformation Function

Example:

$$p_r(r) = \begin{cases} \dfrac{2r}{(L-1)^2} & \text{for } 0 \le r \le L-1 \\ 0 & \text{otherwise} \end{cases}$$

$$s = T(r) = (L-1)\int_0^r p_r(w)\,dw = \frac{2}{L-1}\int_0^r w\,dw = \frac{r^2}{L-1}$$

$$p_s(s) = p_r(r)\left|\frac{dr}{ds}\right| = \frac{2r}{(L-1)^2}\left|\left[\frac{ds}{dr}\right]^{-1}\right|$$

$$= \frac{2r}{(L-1)^2}\left|\left[\frac{d}{dr}\frac{r^2}{L-1}\right]^{-1}\right| = \frac{2r}{(L-1)^2}\left|\frac{(L-1)}{2r}\right| = \frac{1}{L-1}$$

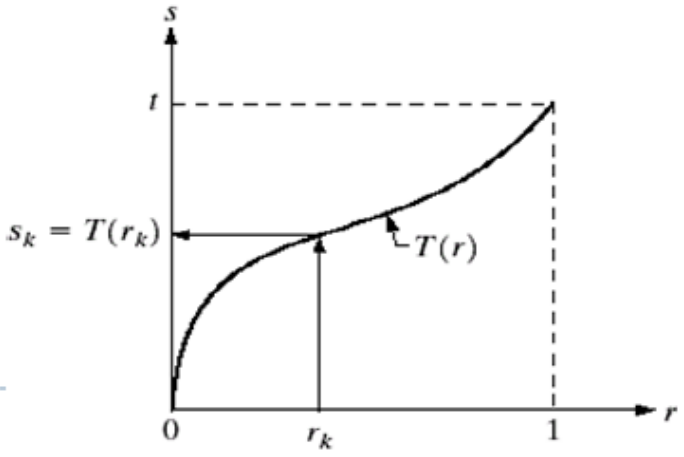
Slide credit: Yan Tong

Histogram Equalization: Transformation Function

- Transformation:
  - Assume r has been normalized to the interval [0, 1], with $r = 0$ representing black and $r = 1$ representing white.

$$s = T(r) \qquad 0 \le r \le 1.$$

  - The transformation function satisfies the following conditions:
    - $T(r)$ is single-valued and monotonically increasing in the interval $0 \le r \le 1$
    - $0 \le T(r) \le 1 \quad \text{for} \quad 0 \le r \le 1.$



Slide credit: Ashish Ghosh

Histogram Equalization

- Histogram equalization is based on a transformation of the probability density function of a random variable.
- Let $p_r(r)$ and $p_s(s)$ denote the probability density function of random variable r and s, respectively.
- If $p_r(r)$ and $T(r)$ are known, then the probability density function $p_s(s)$ of the transformed variable s can be obtained

$$p_s(s) = p_r(r)\left|\frac{dr}{ds}\right|$$

- Define a transformation function $s = T(r) = \int_0^r p_r(w)\,dw$

where w is a dummy variable of integration and the right side of this equation is the cumulative distribution function of random variable r.

Slide credit: Ashish Ghosh

Histogram Equalization

- Given transformation function $T(r)$,

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

$$p_s(s) = p_r(r)\left|\frac{dr}{ds}\right| = p_r(r)\left|\frac{1}{p_r(r)}\right| = 1 \qquad 0 \le s \le 1$$

$p_s(s)$ now is a uniform probability density function.

- $T(r)$ depends on $p_r(r)$, but the resulting $p_s(s)$ always is uniform.

Slide credit: Ashish Ghosh

Histogram Equalization: Discrete Case

$$p_r(r_k) = \frac{n_k}{MN}$$

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

$$s_k = T(r_k) = (L-1)\sum_{j=0}^{k} p_r(r_j) \qquad k = 0, 1, 2, \ldots, L-1$$

Thus, an output image is obtained by mapping each pixel with level r_k in the input image into a corresponding pixel with level s_k.

Histogram Equalization

TABLE 3.1
Intensity distribution and histogram values for a 3-bit, 64×64 digital image.

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

Histogram Equalization

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

$$s_0 = T(r_0) = 7\sum_{j=0}^{0} p_r(r_j) = 7p_r(r_0) = 1.33$$

Similarly, $s_1 = T(r_1) = 3.08$, $s_2 = 4.55$, $s_3 = 5.67$, $s_4 = 6.23$, $s_5 = 6.65$, $s_6 = 6.86$, and $s_7 = 7.00$.

At this point, the s values are fractional because they were generated by summing probability values, so we round them to their nearest integer values in the range $[0, 7]$:

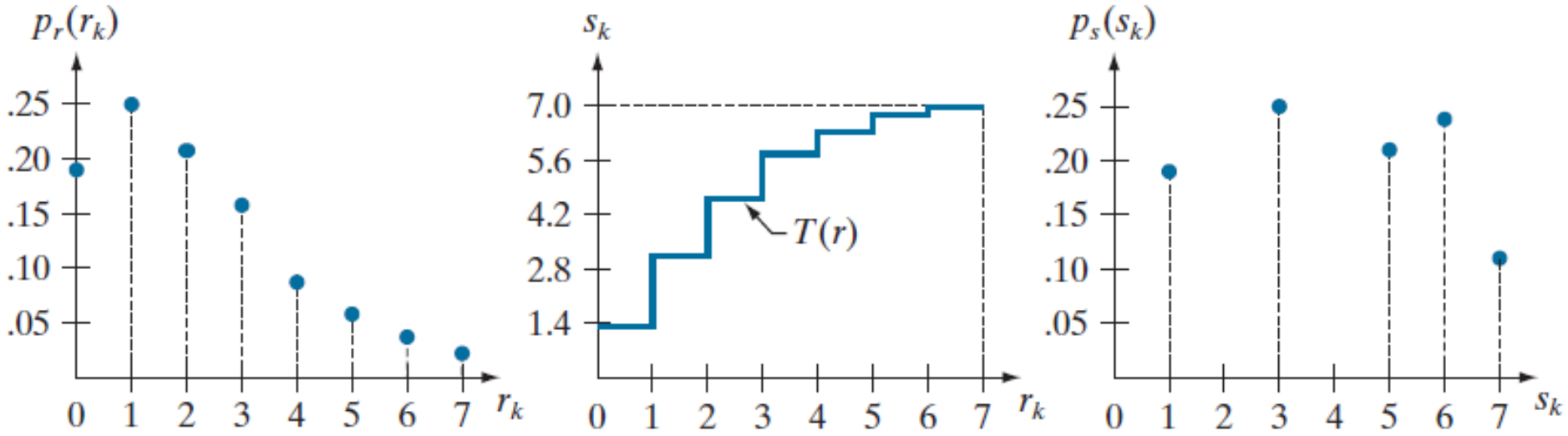
$$s_0 = 1.33 \rightarrow 1 \qquad s_2 = 4.55 \rightarrow 5 \qquad s_4 = 6.23 \rightarrow 6 \qquad s_6 = 6.86 \rightarrow 7$$
$$s_1 = 3.08 \rightarrow 3 \qquad s_3 = 5.67 \rightarrow 6 \qquad s_5 = 6.65 \rightarrow 7 \qquad s_7 = 7.00 \rightarrow 7$$

Histogram Equalization

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

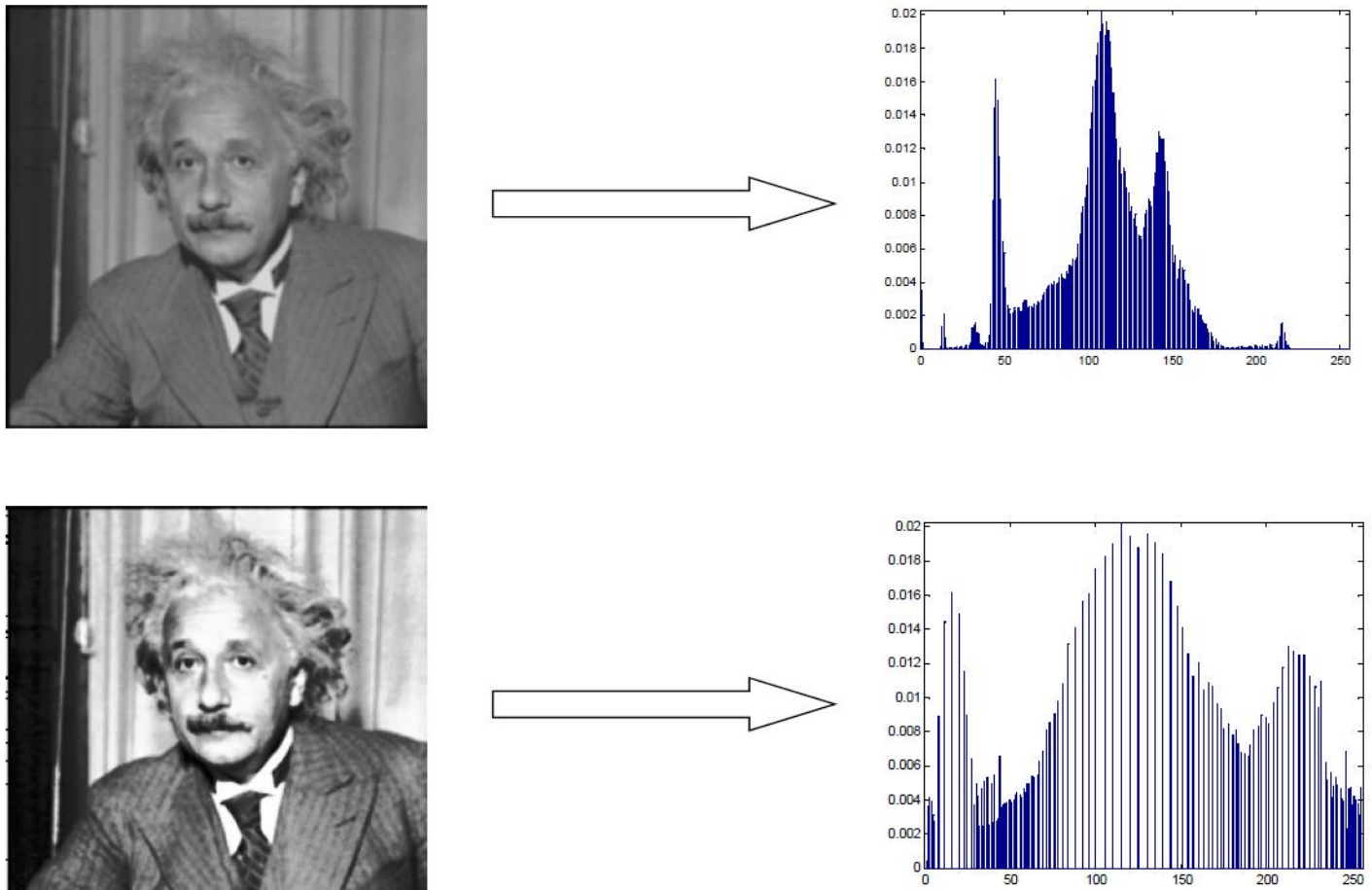
a b c

FIGURE 3.19 Histogram equalization. (a) Original histogram. (b) Transformation function. (c) Equalized histogram.



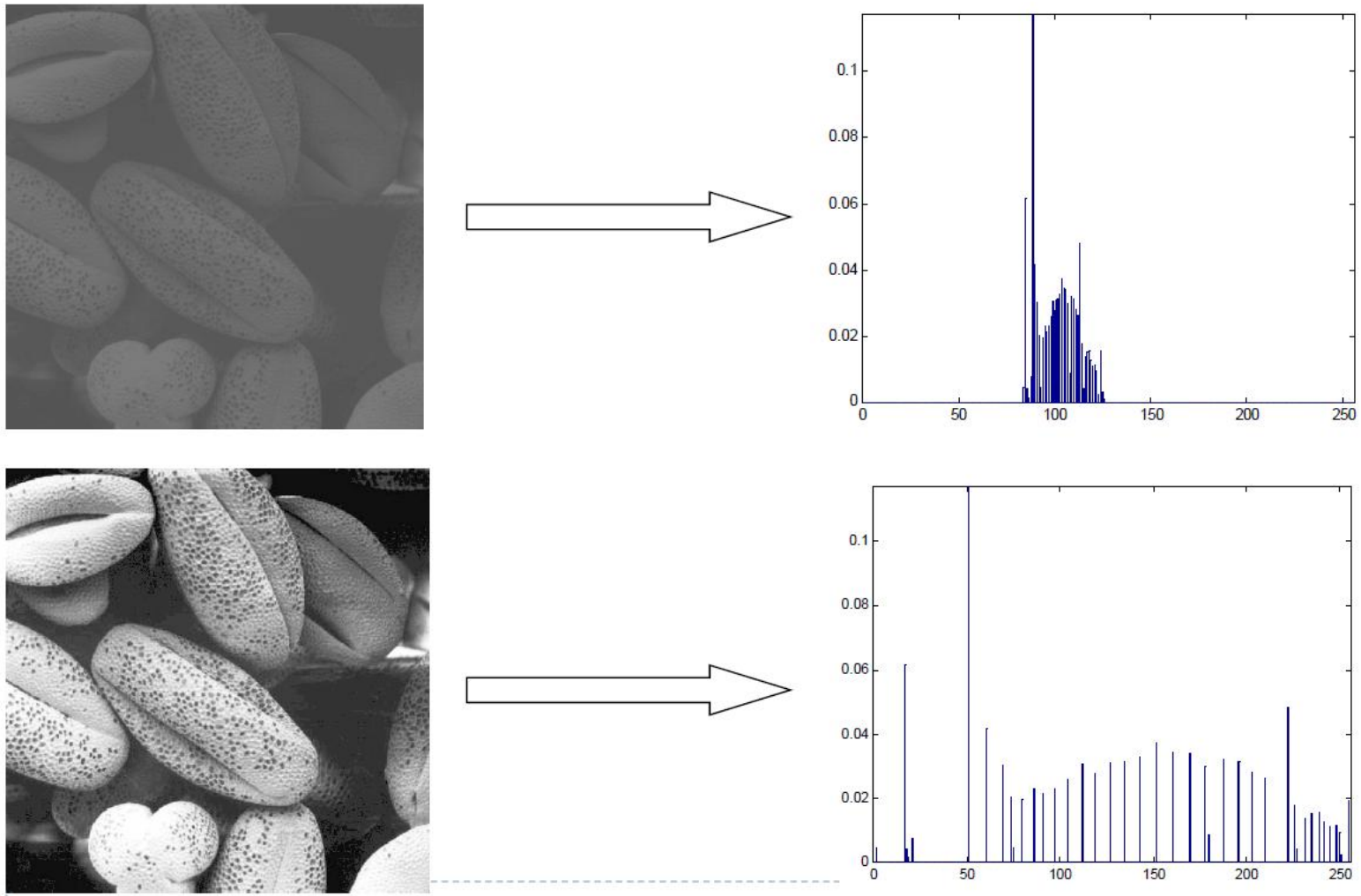
Histogram Equalization

Effect of applying the Histogram Equalization.



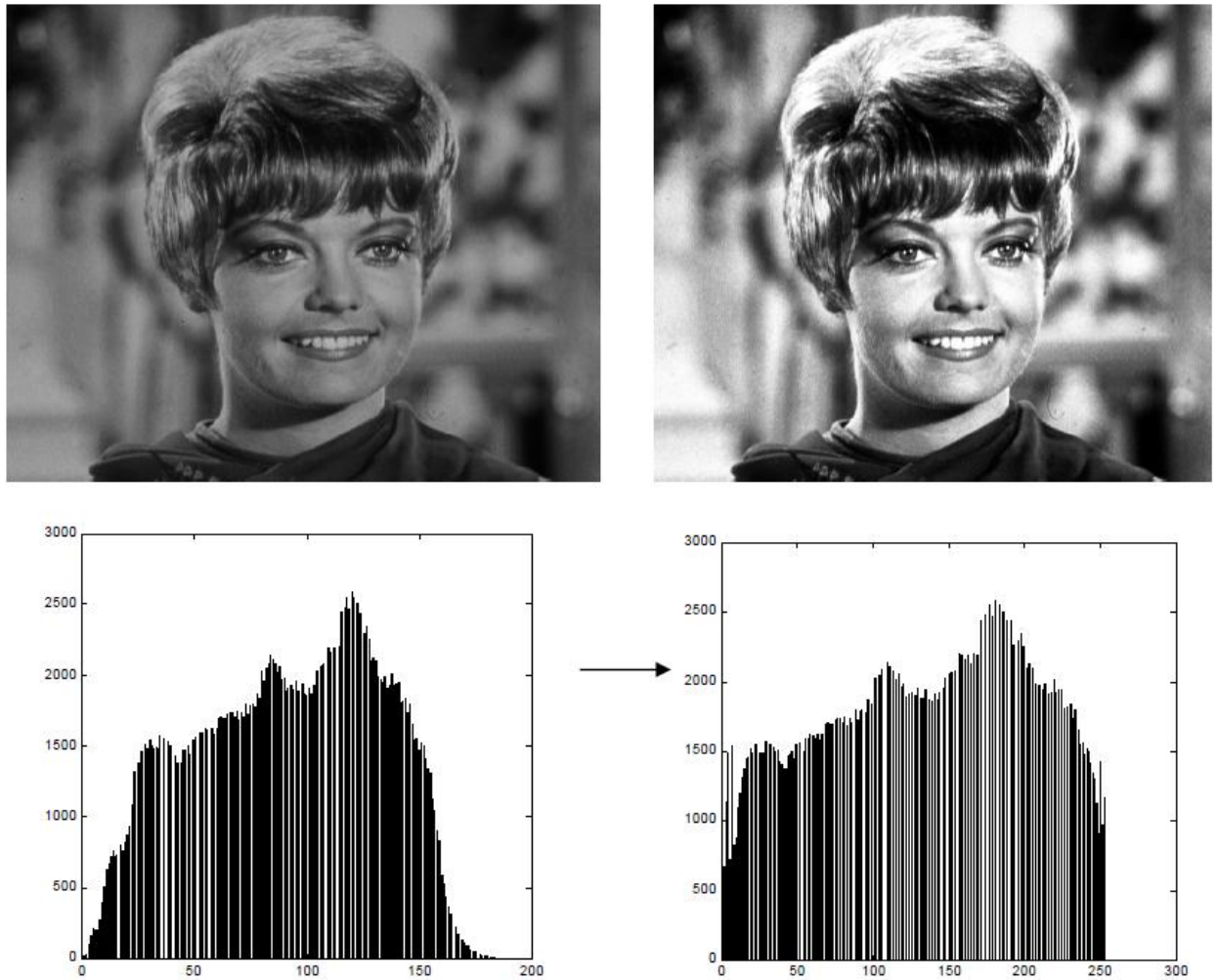
Histogram Equalization

Another example highlighting the effectiveness of this technique



Histogram Equalization

Another example highlighting the effectiveness of this technique



Histogram Matching

Histogram equalization *does not allow interactive image enhancement* and generates only one result that is an approximation to a uniform histogram.

When we need to specify a particular histogram shape capable of highlighting certain gray-level ranges, the technique is called ***histogram matching***.

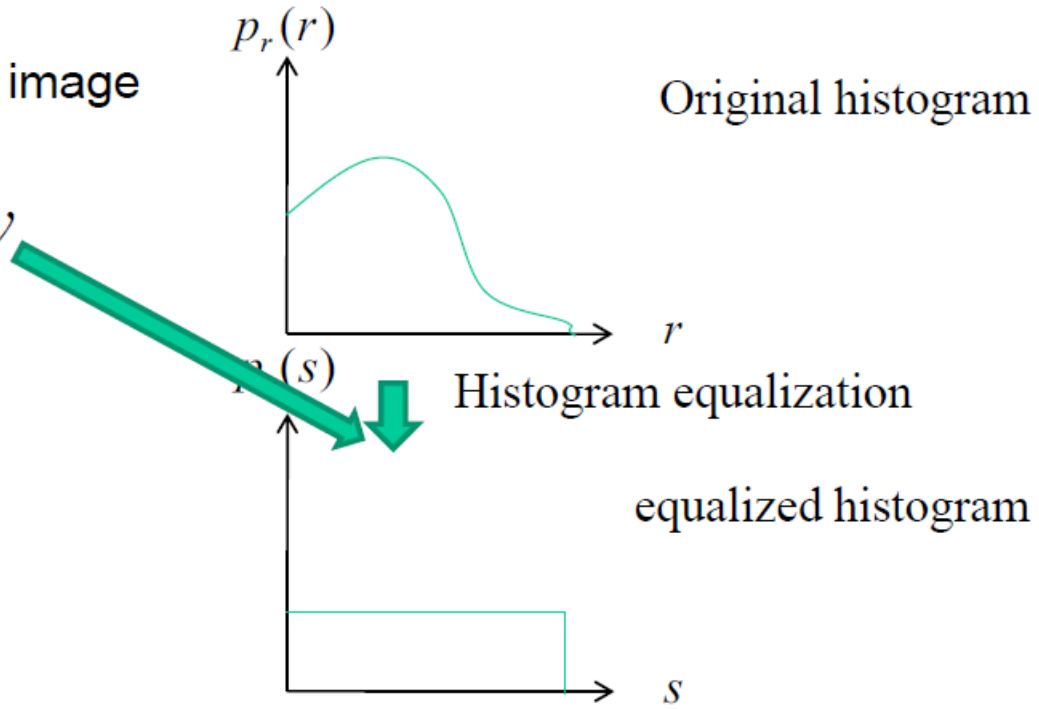
It is desired here that the output image have a specified probability density function $p_z(z)$.

Slide credit: Ashish Ghosh

Histogram Matching

Estimated pdf in the input image

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

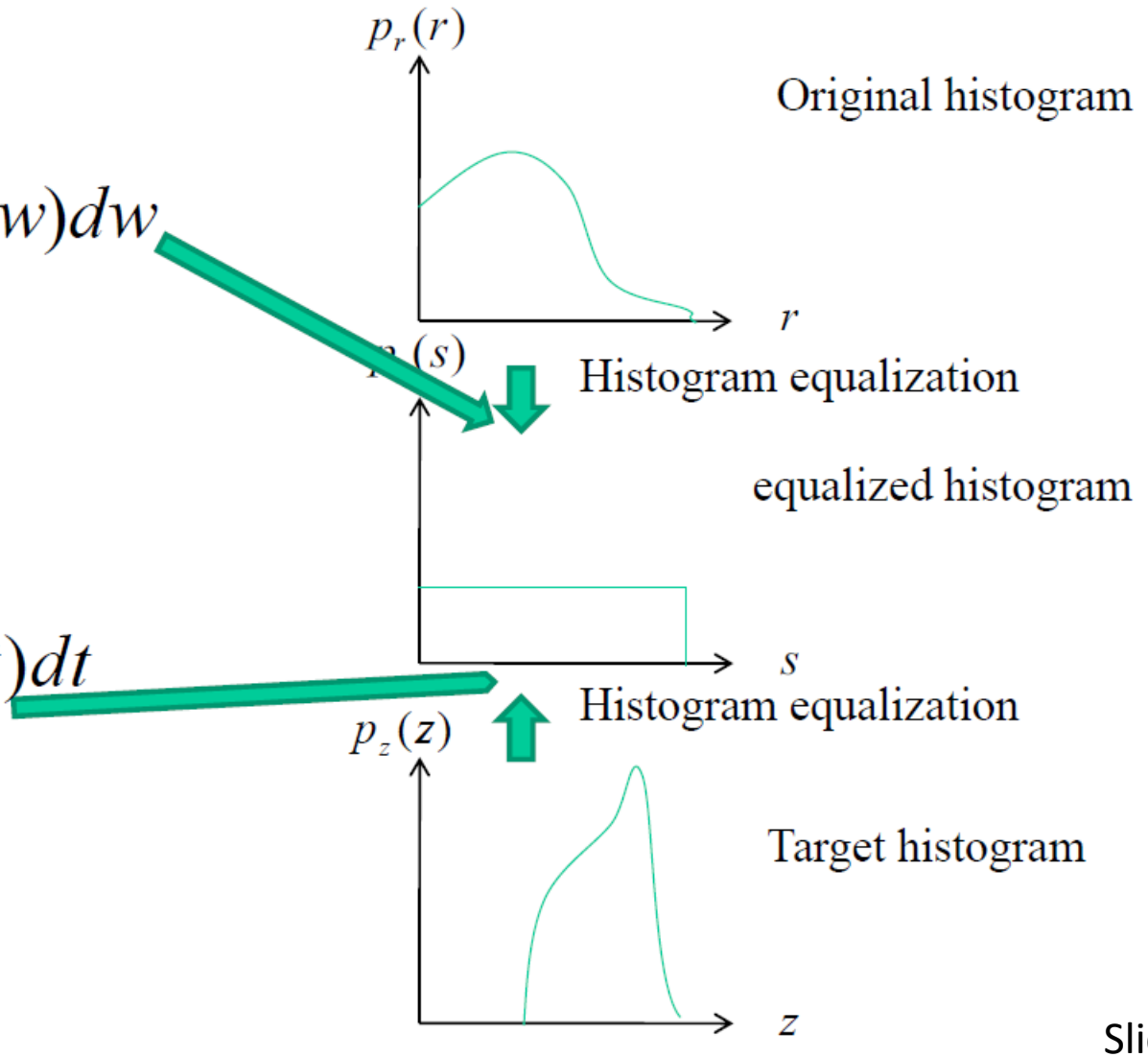


Slide credit: Yan Tong

Histogram Matching

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

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



Slide credit: Yan Tong

Histogram Matching

Estimated pdf in the input image

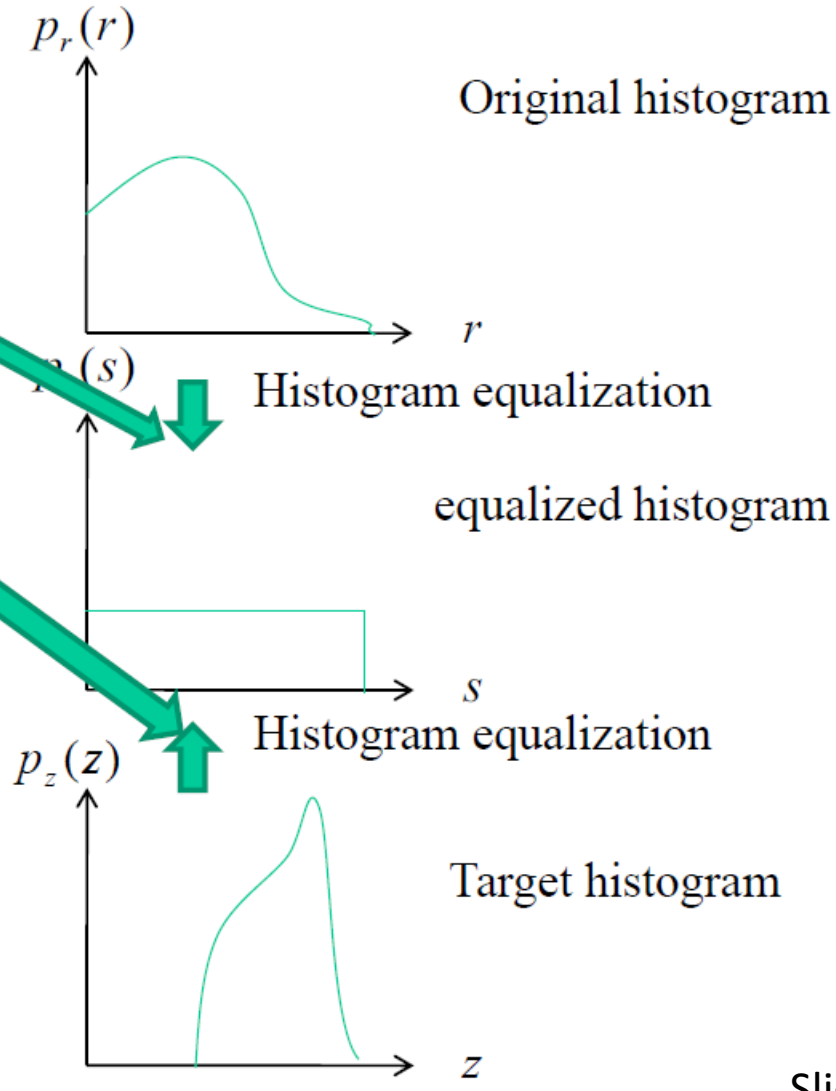
$$s = T(r) = (L-1)\int_0^r p_r(w)dw$$

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

Desired pdf in the output image

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

Green circles represent known
Red circles represent unknown



Slide credit: Yan Tong

Histogram Matching

Discrete histogram require a discretization of the output intensity values

Step1: Compute histogram of the input image $p_r(r)$ and the histogram equalized image $s = T(r)$

Step2: Compute $G(z)$ given the desired histogram $p_z(z)$

Ideally, $G(z) = s$. In practice, $G(z) \approx s$

Step3: Given the s_k value, find the value of z_q so that $G(z_q)$ is closest to s_k

Step4: form the histogram-specified image using the mapping r-z found above

Slide credit: Yan Tong

Histogram Matching

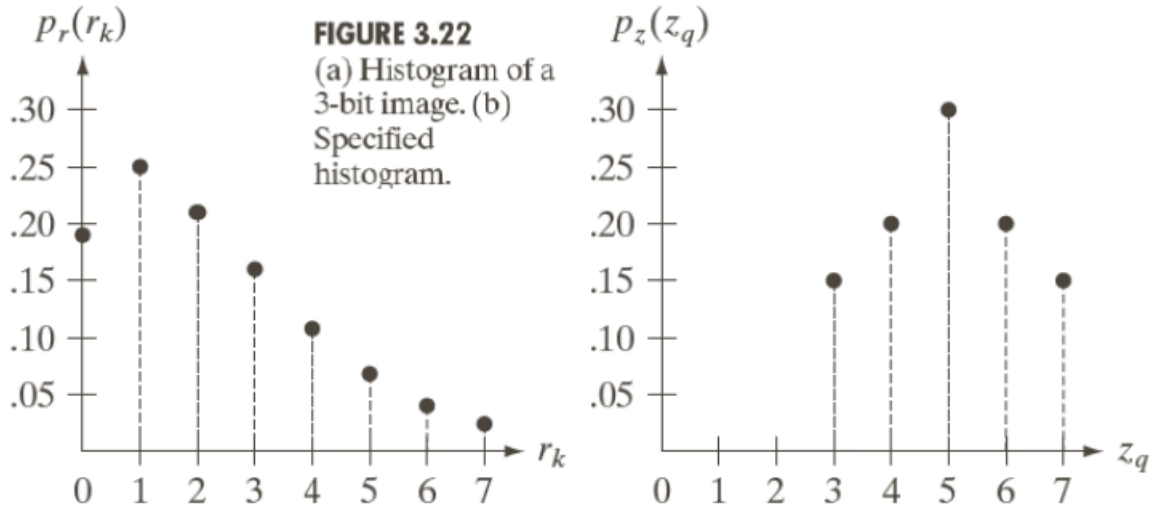


FIGURE 3.22 (a) Histogram of a 3-bit image. (b) Specified histogram.

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 $p_z(z_q)$
$z_0 = 0$	0.00
$z_1 = 1$	0.00
$z_2 = 2$	0.00
$z_3 = 3$	0.15
$z_4 = 4$	0.20
$z_5 = 5$	0.30
$z_6 = 6$	0.20
$z_7 = 7$	0.15

$$s_k = T(r_k) = (L-1)\sum_{j=0}^{k} p_r(r_j) \qquad k = 0, 1, 2, \ldots, L-1$$

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

$$G(z_q) = s_k$$

Slide credit: Yan Tong

Histogram Matching

$$s_0 = 1.33 \rightarrow 1 \quad s_2 = 4.55 \rightarrow 5 \quad s_4 = 6.23 \rightarrow 6 \quad s_6 = 6.86 \rightarrow 7$$
$$s_1 = 3.08 \rightarrow 3 \quad s_3 = 5.67 \rightarrow 6 \quad s_5 = 6.65 \rightarrow 7 \quad s_7 = 7.00 \rightarrow 7$$

$$s_0 = 1;\ s_1 = 3;\ s_2 = 5;\ s_3 = 6;\ s_4 = 6;\ s_5 = 7;\ s_6 = 7;\ s_7 = 7$$

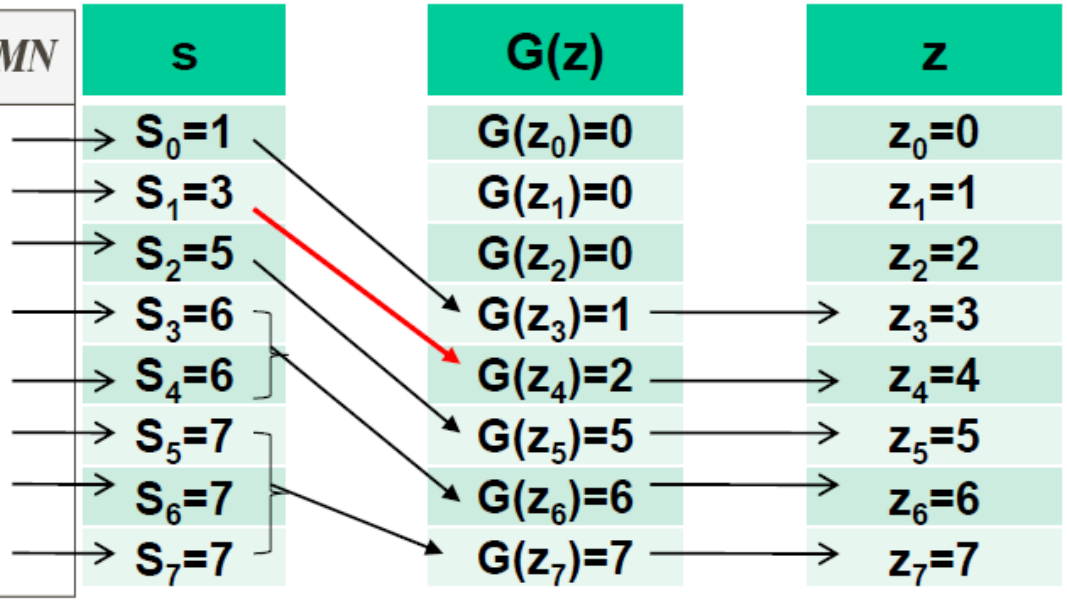
$$G(z_0) = 0.00 \quad G(z_2) = 0.00 \quad G(z_4) = 2.45 \quad G(z_6) = 5.95$$
$$G(z_1) = 0.00 \quad G(z_3) = 1.05 \quad G(z_5) = 4.55 \quad G(z_7) = 7.00$$

$$G(z_0) = 0.00 \rightarrow 0 \qquad G(z_4) = 2.45 \rightarrow 2$$
$$G(z_1) = 0.00 \rightarrow 0 \qquad G(z_5) = 4.55 \rightarrow 5$$
$$G(z_2) = 0.00 \rightarrow 0 \qquad G(z_6) = 5.95 \rightarrow 6$$
$$G(z_3) = 1.05 \rightarrow 1 \qquad G(z_7) = 7.00 \rightarrow 7$$

Histogram Matching

$$s_k = T(r_k) = (L-1)\sum_{j=0}^{k} p_r(r_j) \qquad k = 0, 1, 2, \ldots, L-1$$

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



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

$$G(z_q) = s_k$$

$r_0 \rightarrow z_3$

$r_1 \rightarrow z_4$

$r_2 \rightarrow z_5$

$r_3, r_4 \rightarrow z_6$

$r_5, r_6, r_7 \rightarrow z_7$

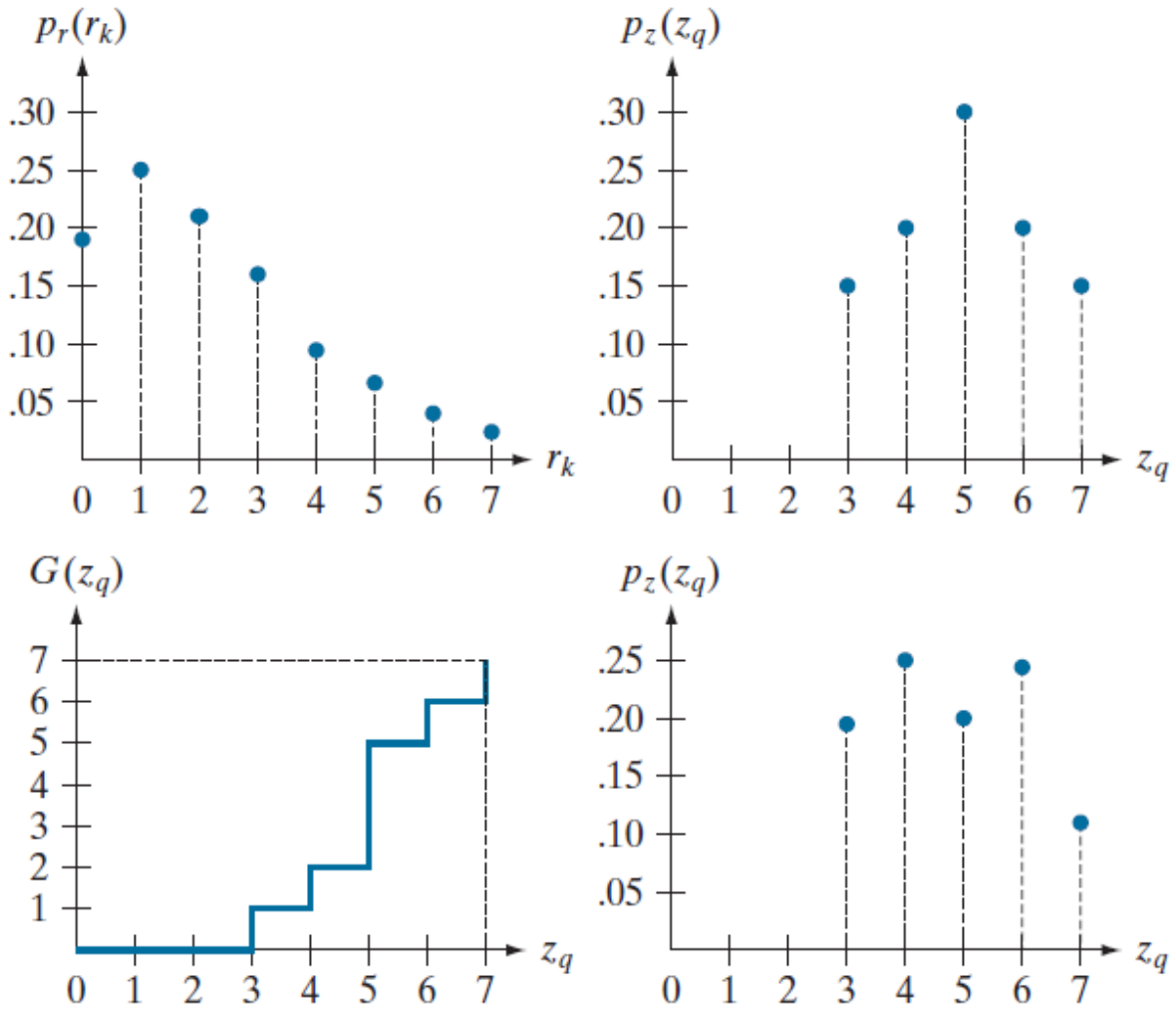
z_q	Specified $p_z(z_q)$	Actual $p_z(z_k)$
$z_0 = 0$	0.00	0.00
$z_1 = 1$	0.00	0.00
$z_2 = 2$	0.00	0.00
$z_3 = 3$	0.15	0.19
$z_4 = 4$	0.20	0.25
$z_5 = 5$	0.30	0.21
$z_6 = 6$	0.20	0.24
$z_7 = 7$	0.15	0.11

Slide credit: Yan Tong

Histogram Matching

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 histogram specification. Compare the histograms in (b) and (d).

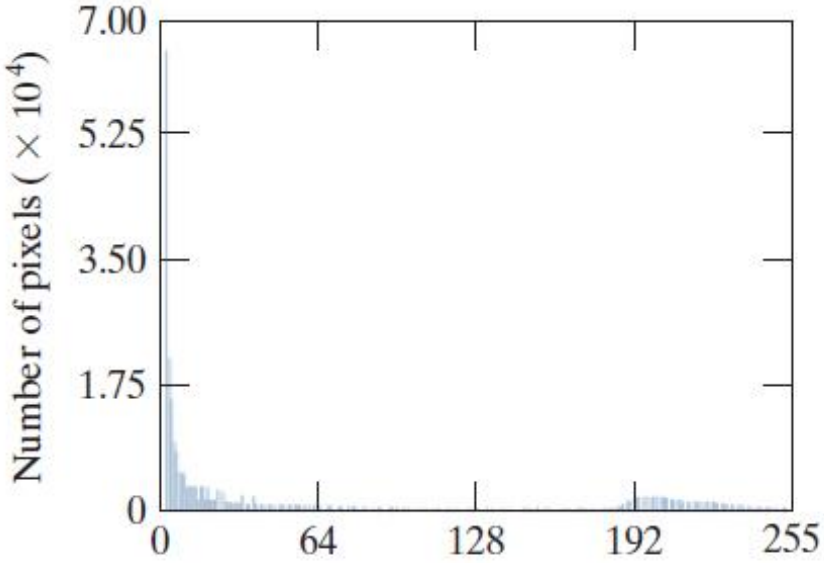


Slide credit: Yan Tong

Histogram Matching: Example

a b

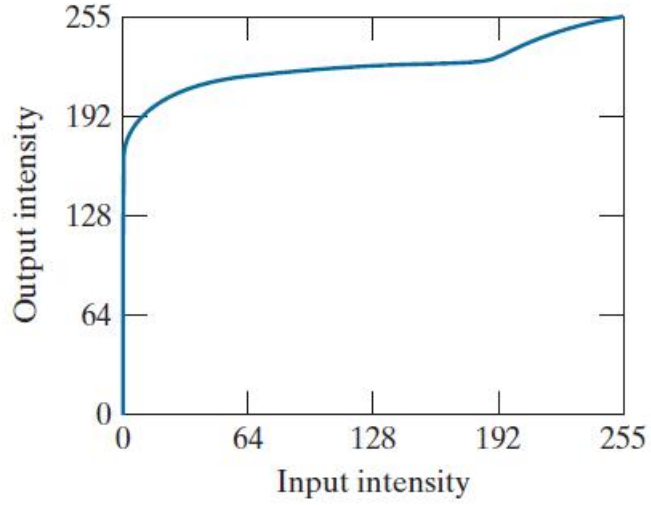
FIGURE 3.23
(a) An image, and
(b) its histogram.

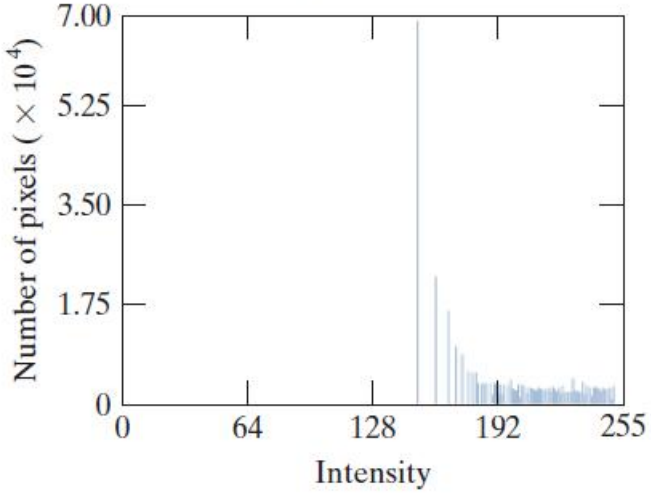


Histogram Matching: Example

a b
c

FIGURE 3.24
(a) Histogram equalization transformation obtained using the histogram in Fig. 3.23(b). (b) Histogram equalized image. (c) Histogram of equalized image.

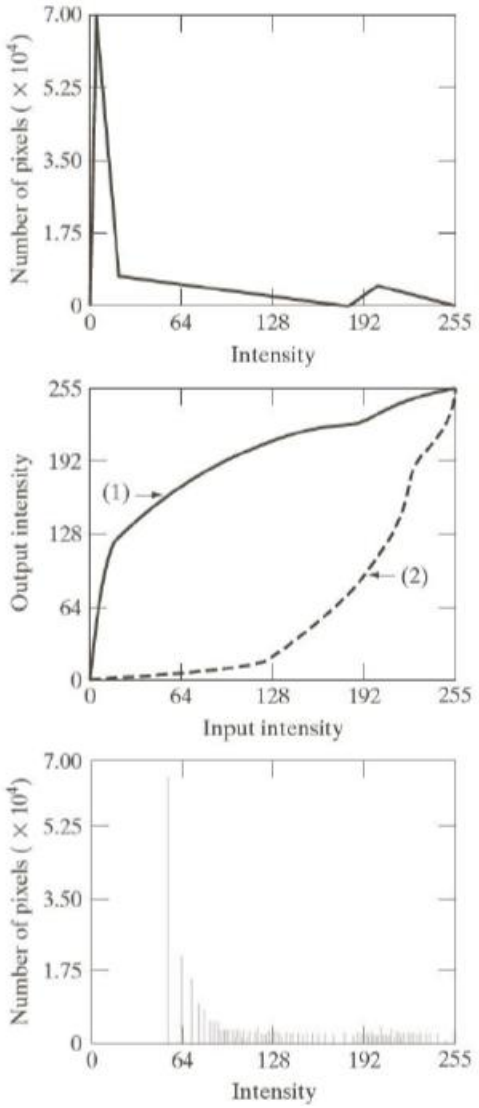




Histogram Matching: Example

Original

HE result



a c
b
d

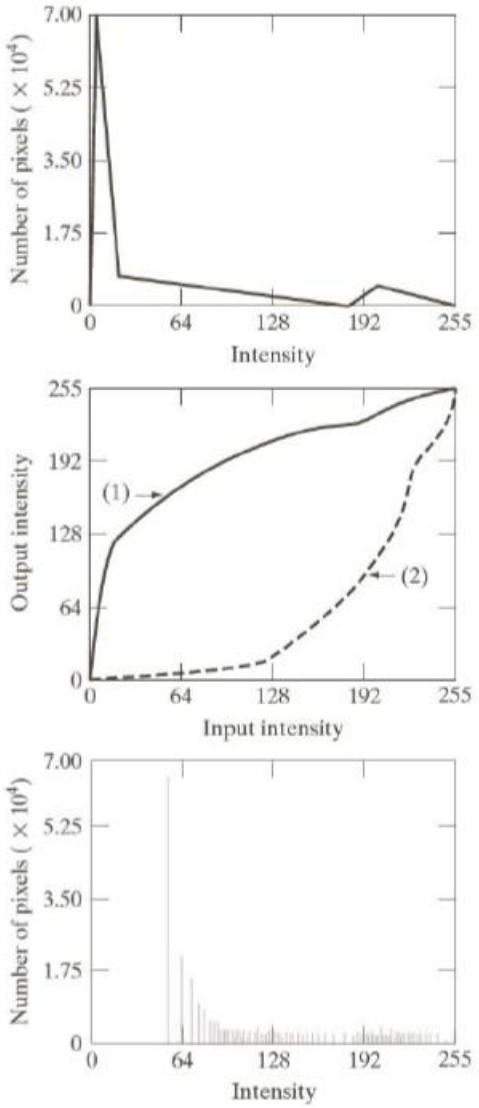
FIGURE 3.25
(a) Specified histogram.
(b) Transformations.
(c) Enhanced image using mappings from curve (2).
(d) Histogram of (c).

Slide credit: Yan Tong

Histogram Matching: Example

Original

HE result



a c
b
d

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

Slide credit: Yan Tong

Local Histogram Processing

Local Enhancement is used when it is necessary to enhance details over smaller areas.

The transformation functions are based on the gray-level distribution in the neighborhood of every pixel.

The **procedure** is:

Define a square/rectangular neighborhood.

Compute the histogram of the points in the neighborhood.

Obtain a histogram equalization/specification transformation.

Use this function to map the gray level of the center pixel.

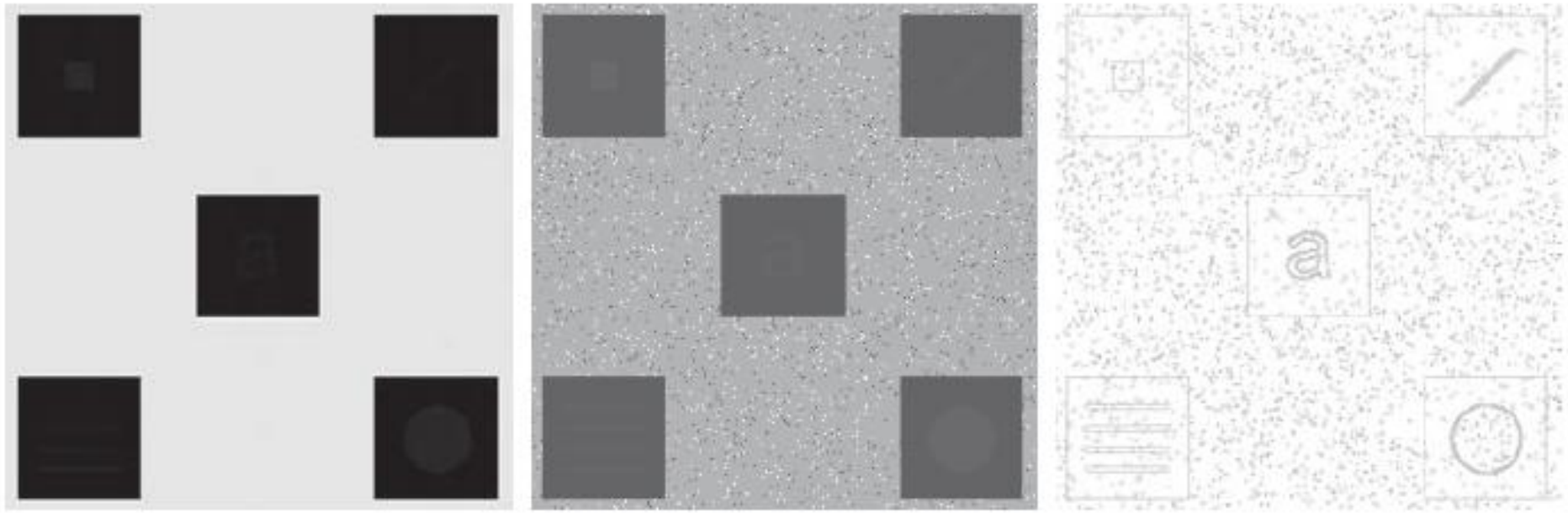
Move the center to the adjacent pixel location and repeat.

Slide credit: Ashish Ghosh

Local Histogram Processing

a b c

FIGURE 3.26
(a) Original image. (b) Result of global histogram equalization. (c) Result of local histogram equalization.

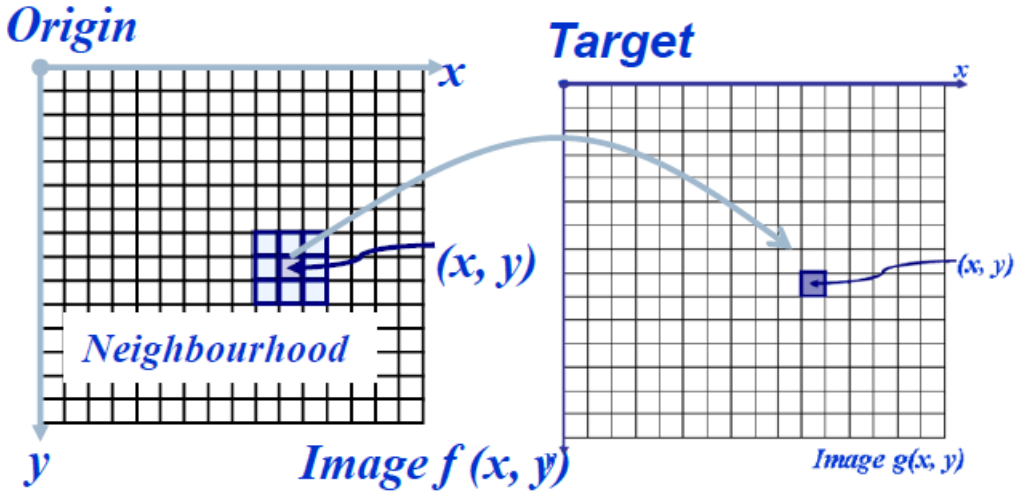


Slide credit: Yan Tong

Spatial Filtering

Neighbourhood Operations

- Neighbourhood operations simply operate on a larger neighbourhood of pixels than point operations
- Neighbourhoods are mostly rectangular
- Filters of any shape and size are possible
- For each pixel in the original image, the outcome is written on the same location at the target image.



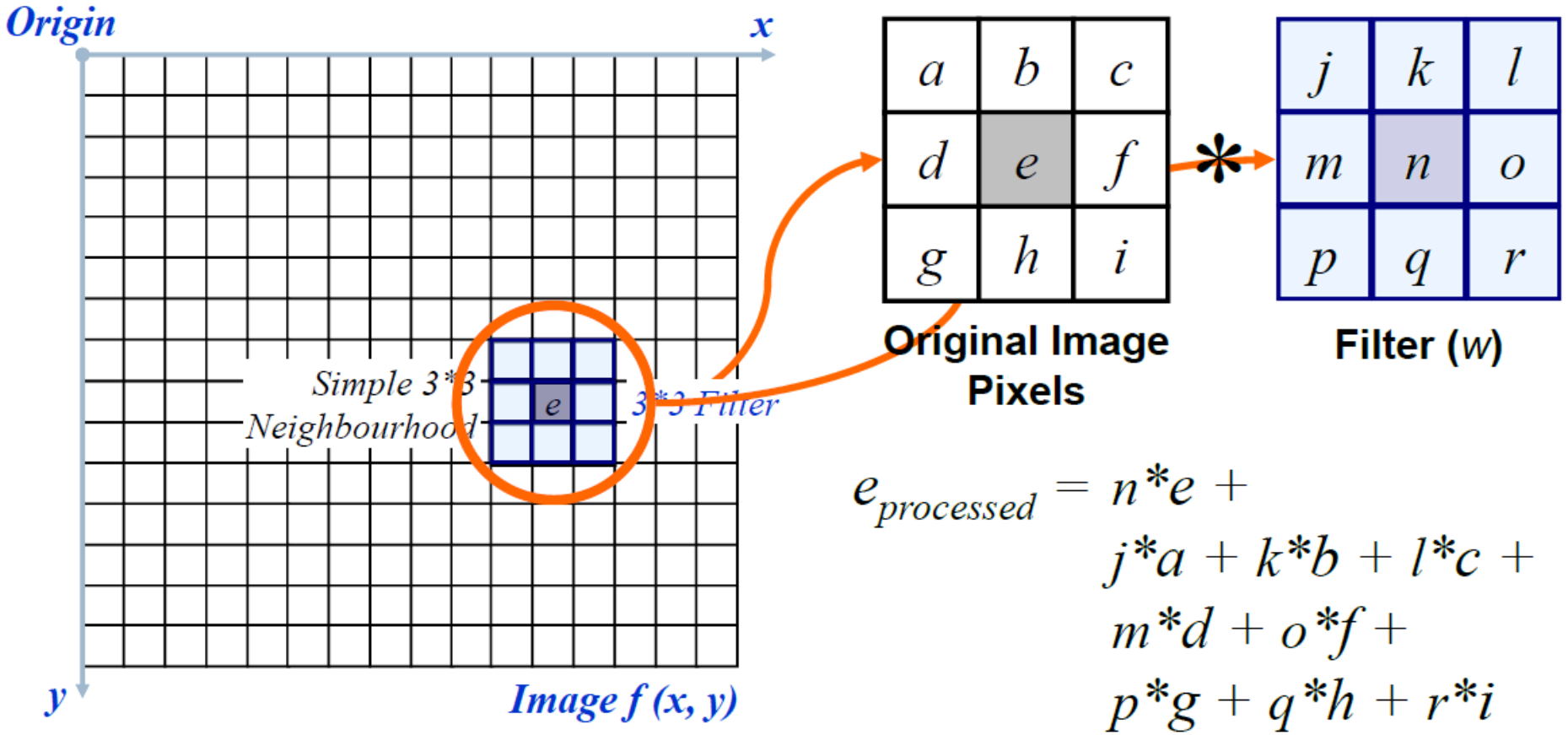
Slide credit: Ashish Ghosh

Neighbourhood Operations

- Simple neighbourhood operations example:
  - **Min:** Set the pixel value to the minimum in the neighbourhood
  - **Max:** Set the pixel value to the maximum in the neighbourhood

Slide credit: Ashish Ghosh

Spatial Filtering Process

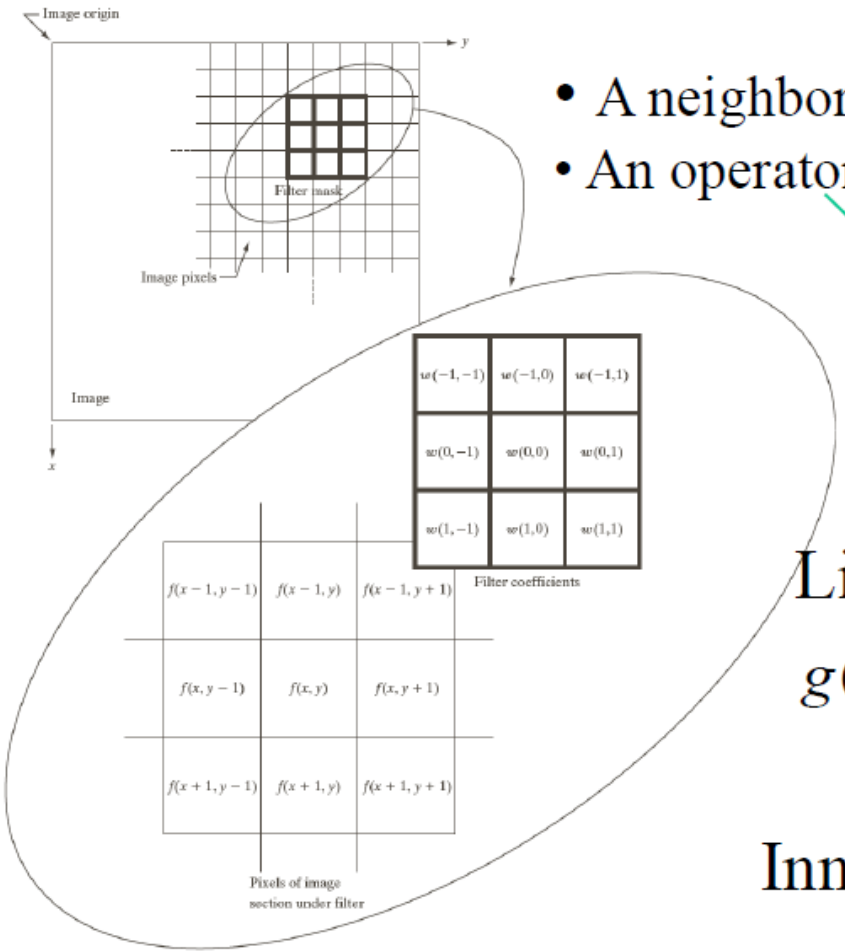


$$e_{processed} = n*e + j*a + k*b + l*c + m*d + o*f + p*g + q*h + r*i$$

- The above is repeated for every pixel in the original image to generate the filtered image

Slide credit: Ashish Ghosh

Fundamentals of Spatial Filtering



- A neighborhood
- An operator with the same size: linear/nonlinear

Note: Each element in w will visit every pixel in the image just once.

Linear spatial filtering:

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

Inner product $g(x,y) = \mathbf{w} \bullet \mathbf{f} = \mathbf{w}^T \mathbf{f}$

FIGURE 3.28 The mechanics of linear spatial filtering using a 3×3 filter mask. The form chosen to denote the coordinates of the filter mask coefficients simplifies writing expressions for linear filtering.

Slide credit: Yan Tong

Fundamentals of Spatial Filtering

✓Low-pass filters eliminate or attenuate high frequency components in the frequency domain, and result in image blurring.

✓High-pass filters attenuate or eliminate low-frequency components resulting in sharpening edges and other sharp details.

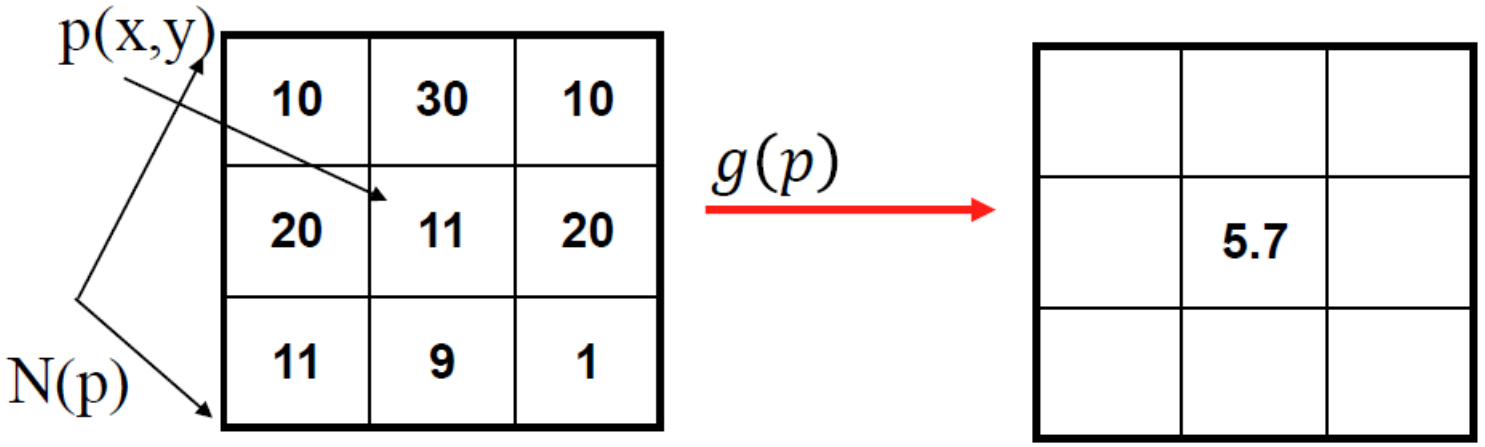
$$g(x,y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t) f(x+s, y+t)$$

- Here $w(s,t)$ is the mask of size m x n. $a = (m-1)/2$ and $b = (n-1)/2$, m, n are ODD numbers.
- $g(x,y)$ is calculated for x=0,1,…,M-1 and y=0,1,…,N-1 to get the final filtered image.
- Non-linear filters also use pixel neighborhoods but do not explicitly use coefficients.

Slide credit: Ashish Ghosh

Fundamentals of Spatial Filtering

Modifying the pixels in an image based on some function of a local neighborhood of the pixels

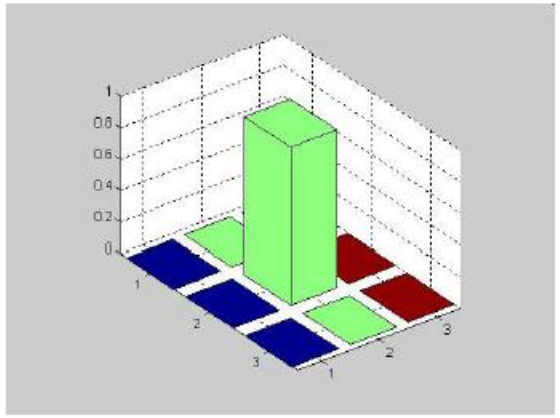


g(p):

- Linear function
  - Correlation
  - Convolution
- Nonlinear function
  - Order statistic (median)

Slide credit: Yan Tong

Linear Filtering



*

0	0	0
0	1	0
0	0	0

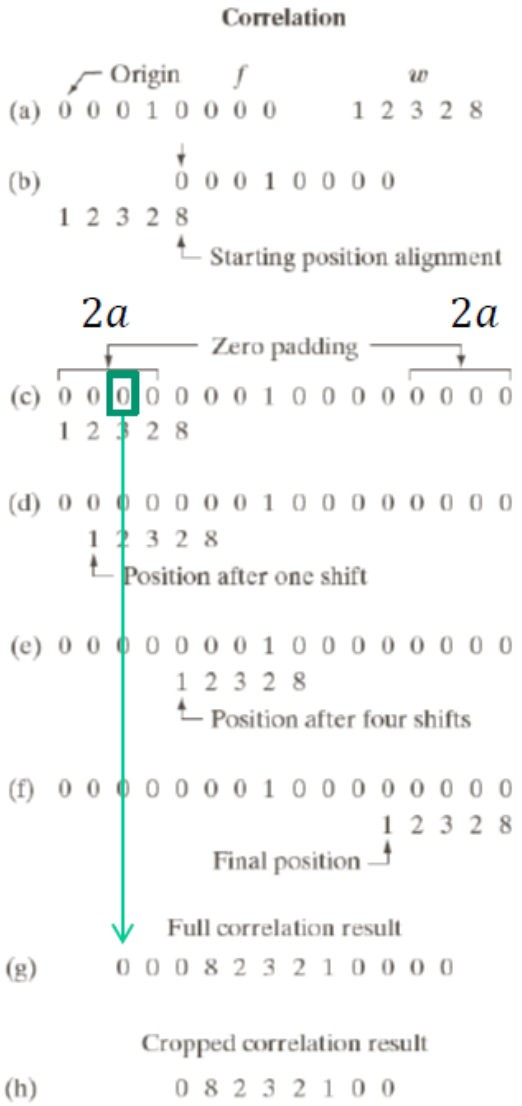
=

Slide credit: Yan Tong

Spatial Correlation: 1D Signal

$$\text{1D correlation} \sum_{s=-a}^{a} w(s) f(x+s)$$

Zero-padding: add zeros on the left and right margin, respectively

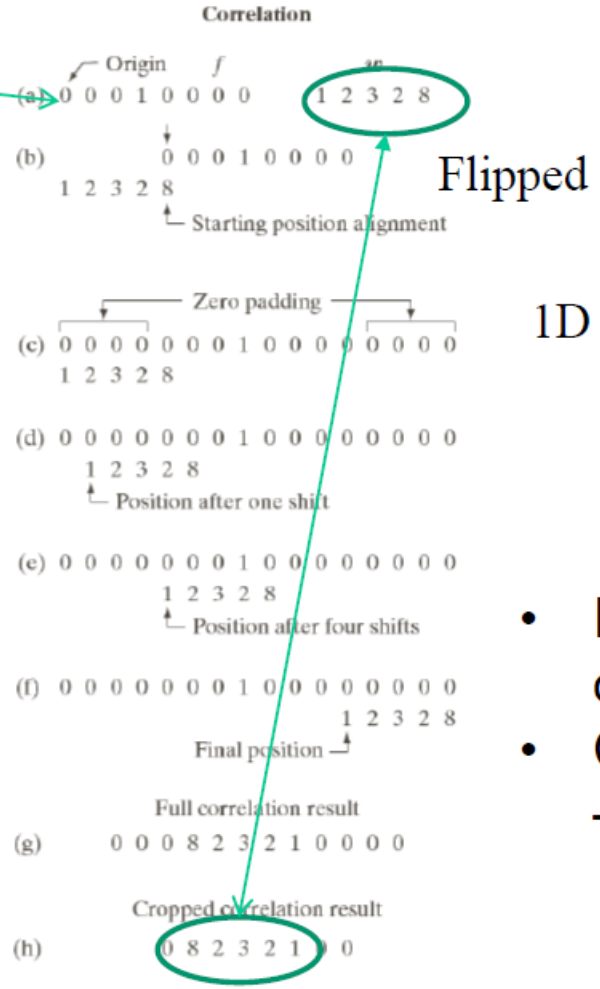


- **Full correlation result** has the size of $M + 2a$
- **Cropped result** has the size of M – the size of the original signal

Slide credit: Yan Tong

Spatial Correlation: 1D Signal

Discrete impulse



Flipped

$$\text{1D correlation} \sum_{s=-a}^{a} w(s) f(x+s)$$

- Full correlation result has the size of $M + 2a$
- Cropped result has the size of M – the size of the original signal

The impulse response is a rotation of the filter by 180 degree

Slide credit: Yan Tong

Spatial Convolution: 1D Signal

Discrete impulse

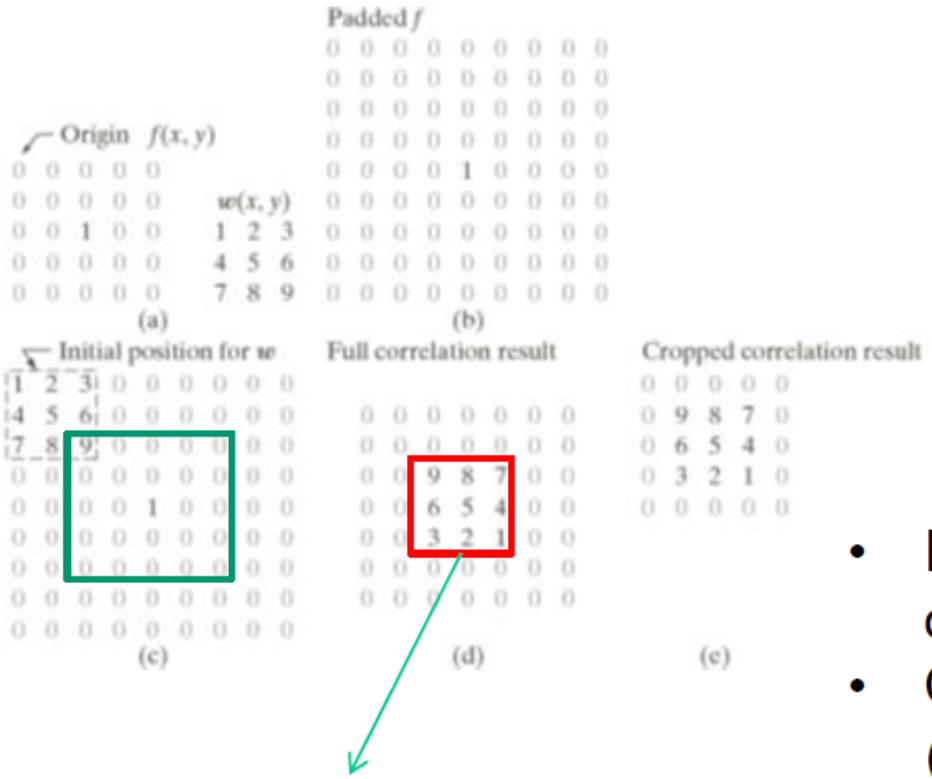
Flipped filter

1D convolution $\sum_{s=-a}^{a} w(s) f(x-s)$

The impulse response is the same as the filter

Slide credit: Yan Tong

Extend to 2D Image: 2D Image Correlation



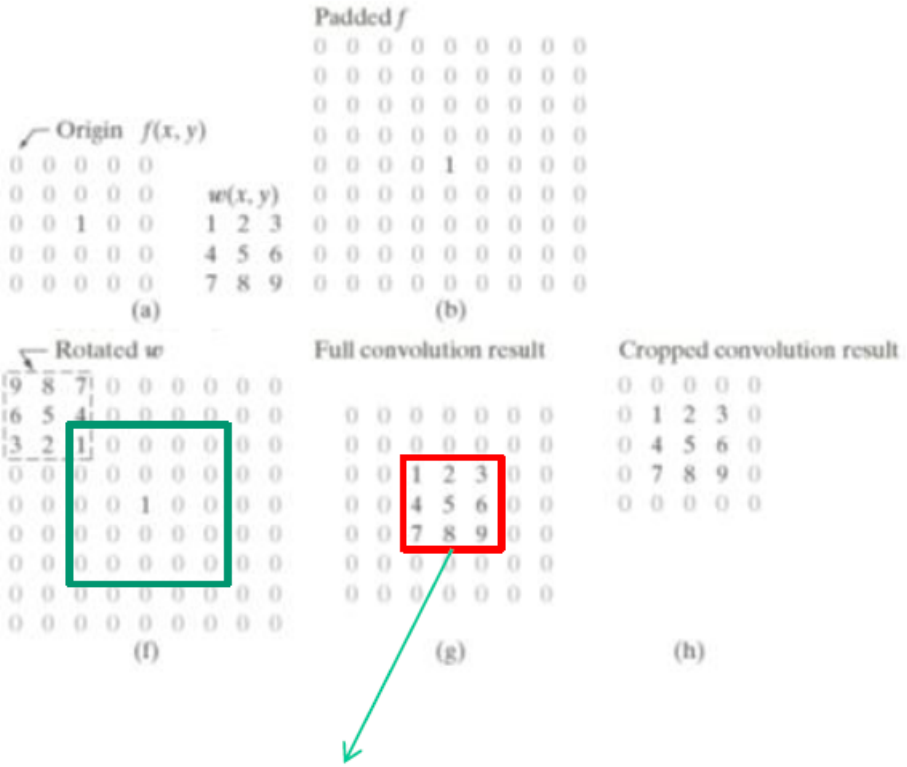
$$\sum_{s=-a}^{a}\sum_{t=-b}^{b} w(s,t)\, f(x+s, y+t)$$

- Full correlation result has the size of $(M + 2a, N + 2b)$
- Cropped result has the size of (M, N) – the size of the original image

The 2D impulse response of image correlation is a rotation of the filter by 180 degree

Slide credit: Yan Tong

Extend to 2D Image: 2D Image Convolution



The 2D impulse response of image convolution is the same as the filter

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

- Flip in both horizontal and vertical directions (rotate 180 degree) -> same if the filter is symmetric
- Convolution filter/mask/kernel
- Full convolution result has the size of $(M + 2a, N + 2b)$
- Cropped result has the size of (M, N) – the size of the original image

Slide credit: Yan Tong

Linear Filters

General process:

- Form new image whose pixels are a weighted sum of original pixel values, using the same set of weights at each point.

Properties

- Output is a linear function of the input
- Output is a shift-invariant function of the input (i.e. shift the input image two pixels to the left, the output is shifted two pixels to the left)

Example: smoothing by averaging

- form the average of pixels in a neighborhood

Example: smoothing with a Gaussian

- form a weighted average of pixels in a neighborhood

Example: finding an edge

Slide credit: Yan Tong

Smoothing Spatial Filter – Low Pass Filters

Weighted average

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

1	1	1
1	1	1
1	1	1

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

1	2	1
2	4	2
1	2	1

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.

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

- Noise deduction
- reduction of "irrelevant details"
- edge blurred

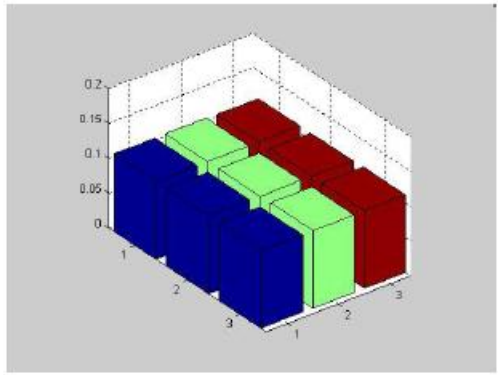
Normalization factor

Slide credit: Yan Tong

Smoothing Spatial Filter

Image averaging $R = \frac{1}{9}\sum_{i=1}^{9} z_i$

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9



$* \frac{1}{9}$

1	1	1
1	1	1
1	1	1

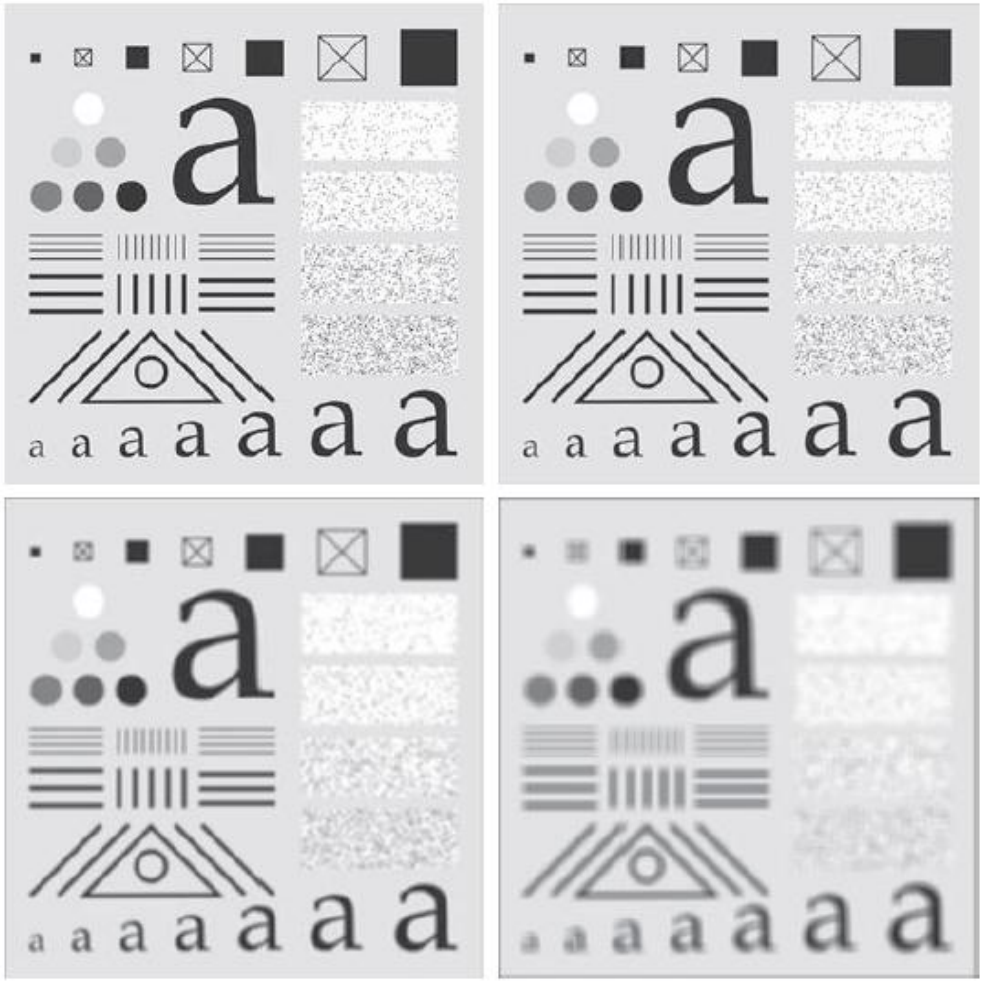
$=$

Slide credit: Yan Tong

Comparison using Different Smoothing Filters – Different Kernels

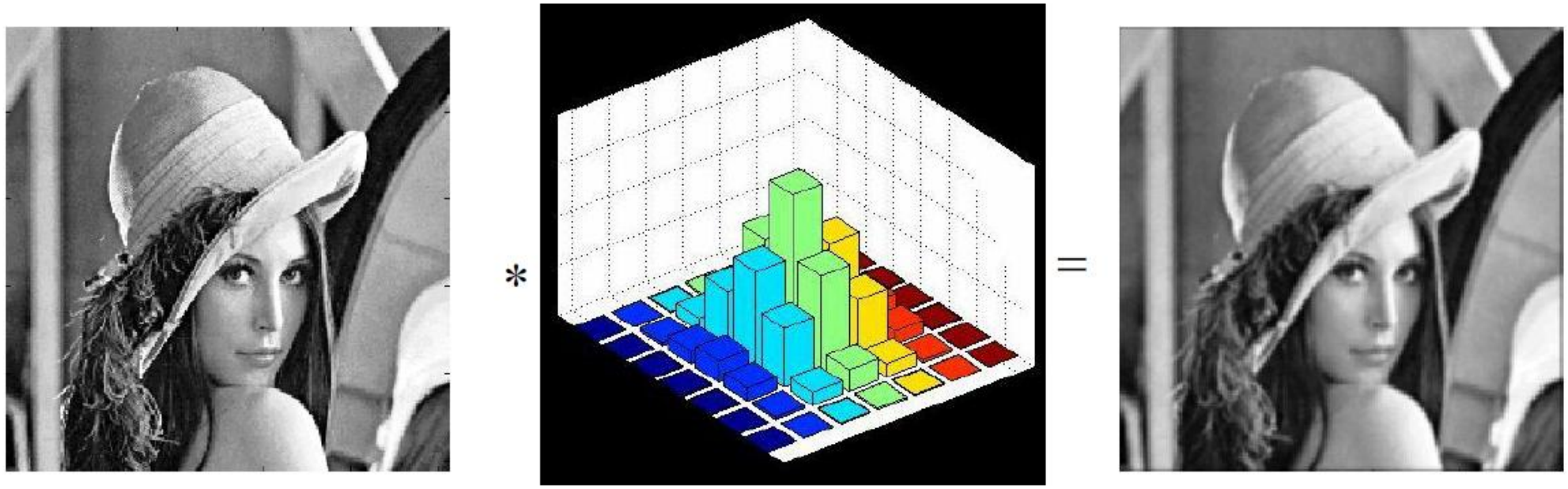
a b
c d

FIGURE 3.33
(a) Test pattern of size 1024×1024 pixels.
(b)-(d) Results of lowpass filtering with box kernels of sizes 3×3, 11×11, and 21×21, respectively.



Smoothing Spatial Filter

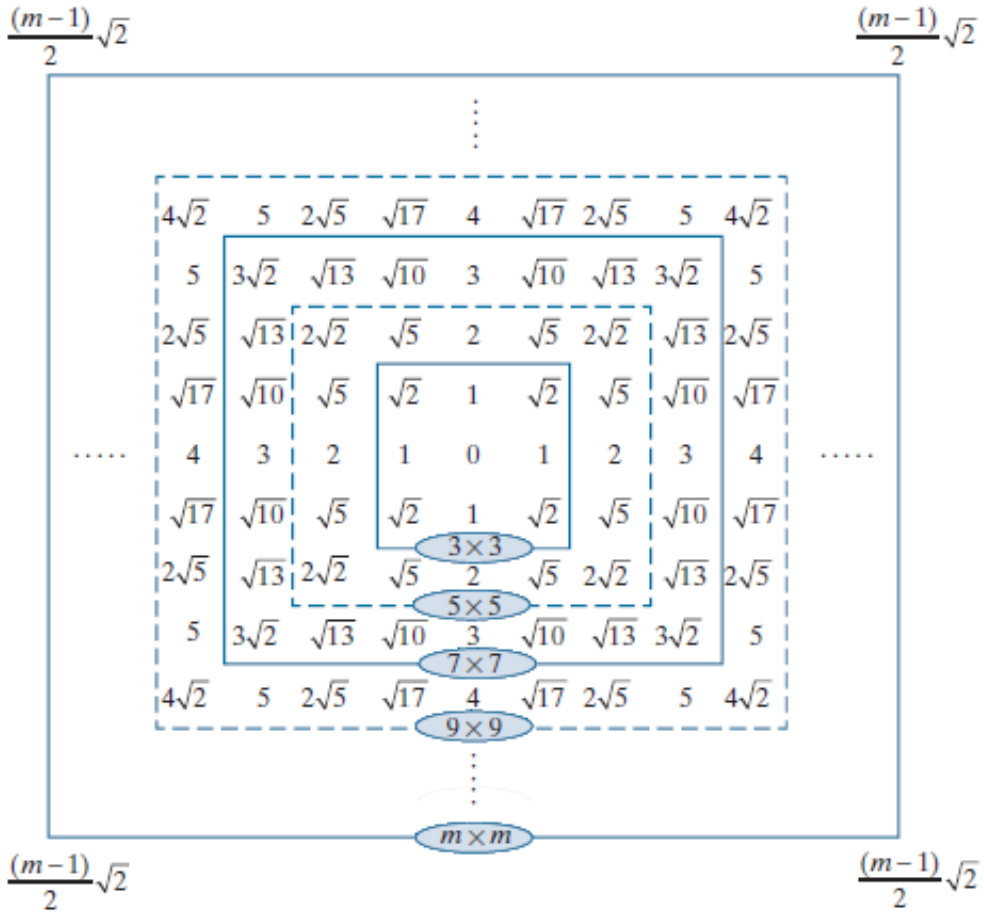
2D Gaussian filter $$h(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$



Slide credit: Yan Tong

Smoothing Spatial Filter

FIGURE 3.34
Distances from the center for various sizes of square kernels.



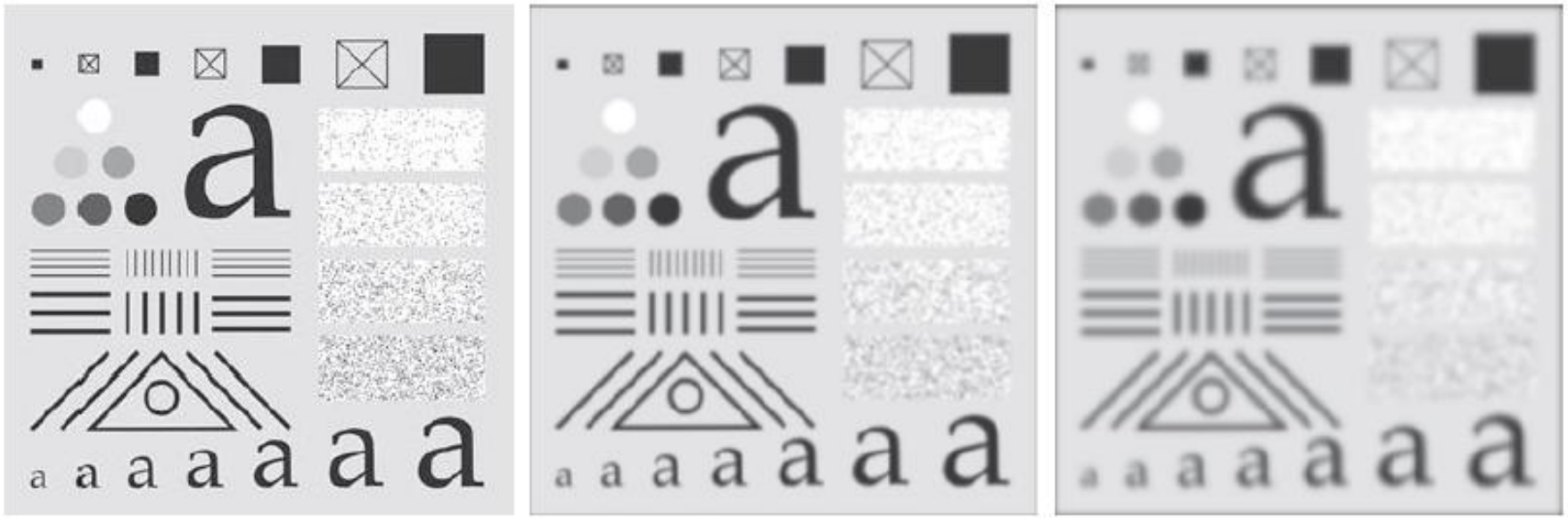
Comparison using Different Smoothing Filters – Different Kernels

Average

Gaussian

Slide credit: Yan Tong

Comparison using Different Smoothing Filters – Different Kernels

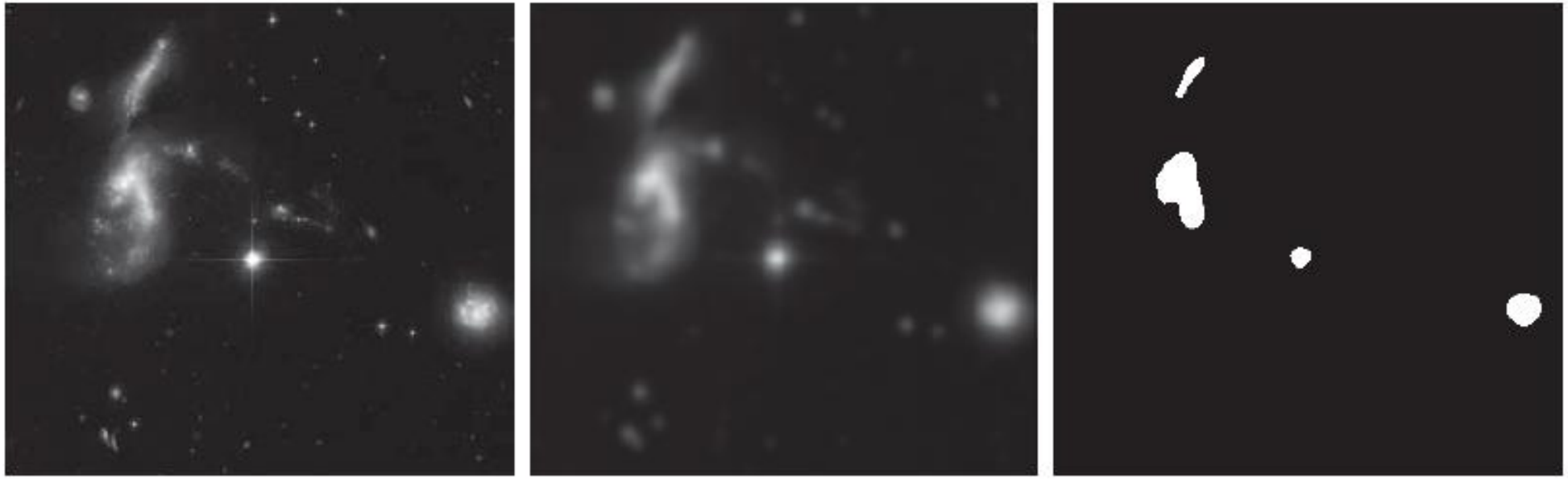


a b c

FIGURE 3.36 (a)A test pattern of size 1024×1024. (b) Result of lowpass filtering the pattern with a Gaussian kernel of size 21×21, with standard deviations $\sigma = 3.5$. (c) Result of using a kernel of size 43×43, with $\sigma = 7$. This result is comparable to Fig. 3.33(d). We used $K = 1$ in all cases.

Image Smoothing and Thresholding

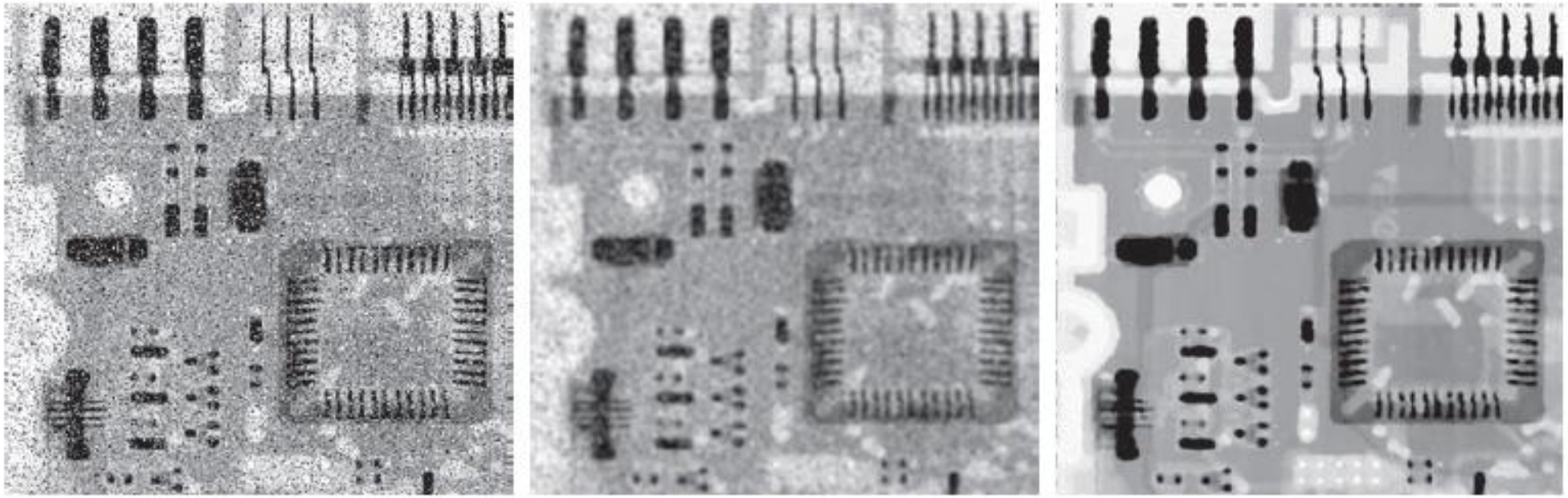
Figure 3.41(b) is the result of filtering the original image with a Gaussian kernel of size 151 x 151 (approximately 6% of the image width) and standard deviation s = 25.



a b c

FIGURE 3.41 (a) A 2566 × 2758 Hubble Telescope image of the *Hickson Compact Group*. (b) Result of lowpass filtering with a Gaussian kernel. (c) Result of thresholding the filtered image (intensities were scaled to the range [0, 1]). The Hickson Compact Group contains dwarf galaxies that have come together, setting off thousands of new star clusters. (Original image courtesy of NASA.)

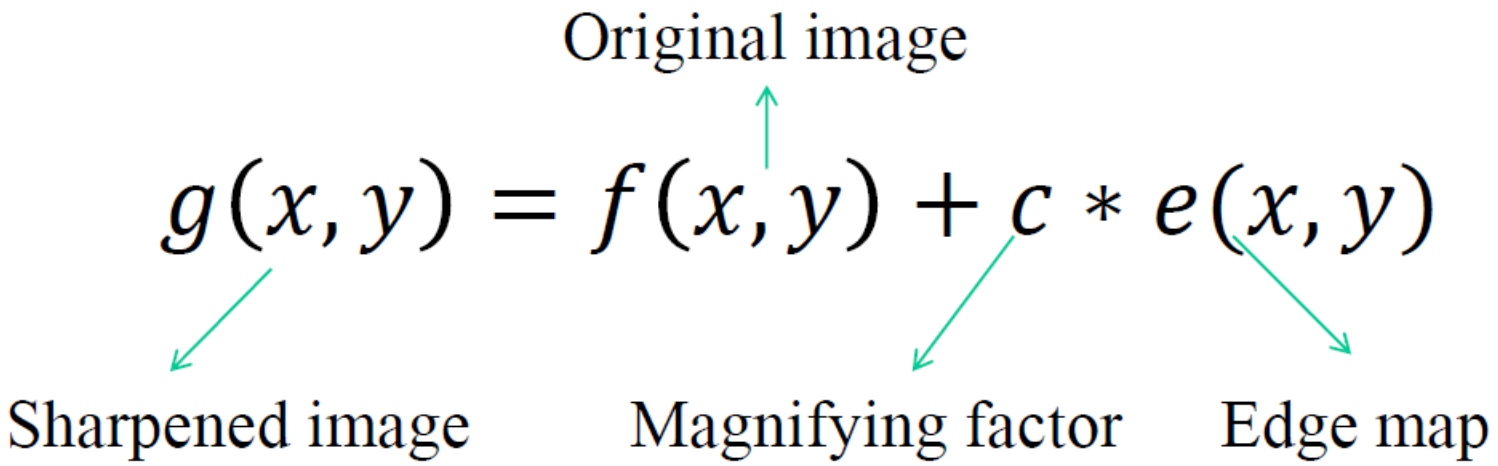
Order-Statistic (Non-linear) Filters – Median Filter



a b c

FIGURE 3.43 (a) X-ray image of a circuit board, corrupted by salt-and-pepper noise. (b) Noise reduction using a 19×19 Gaussian lowpass filter kernel with $\sigma = 3$. (c) Noise reduction using a 7×7 median filter. (Original image courtesy of Mr. Joseph E. Pascente, Lixi, Inc.)

Sharpening Spatial Filters



We will briefly introduce edge detection here and will have a more comprehensive discussion when we discuss image segmentation.

Slide credit: Yan Tong

Spatial Filters for Edge Detection

first derivative

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

zero in areas of constant intensity

Nonzero at

- onset of ramp and step
- along ramp or step

second derivative

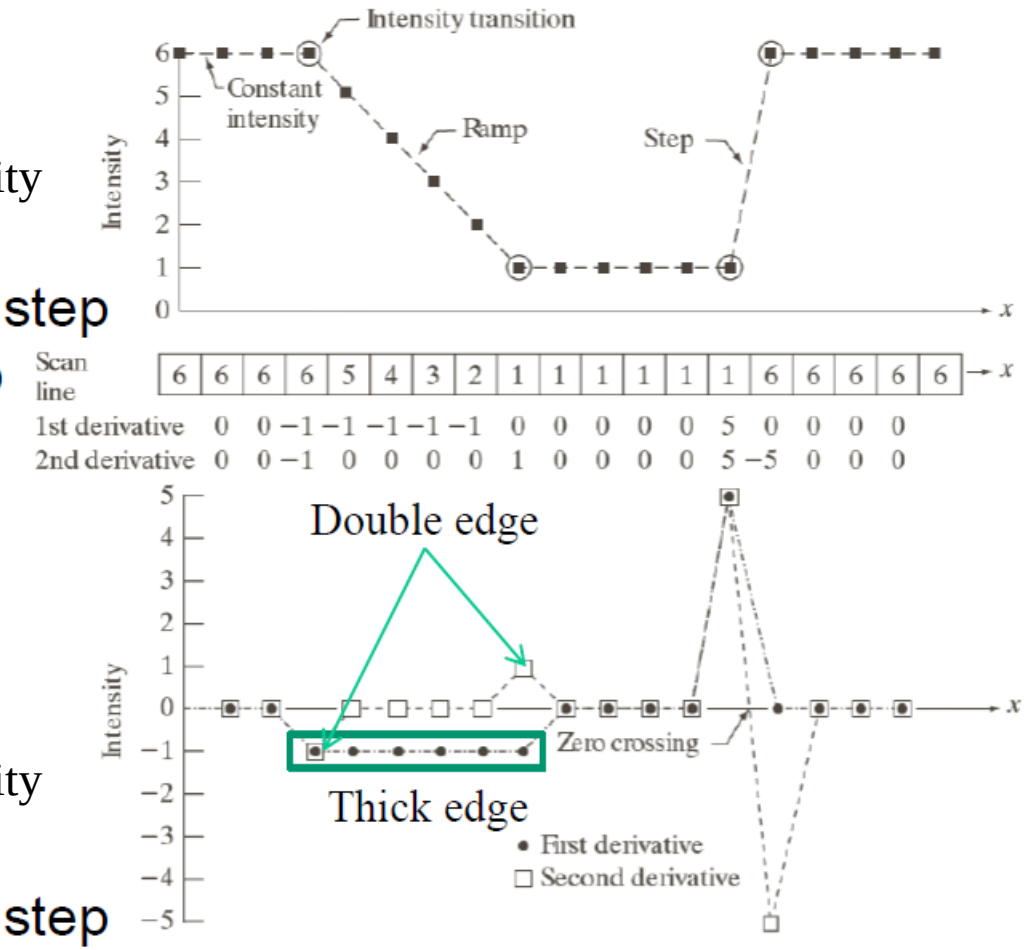
$$\frac{\partial^2 f}{\partial x^2} = f(x+1) + f(x-1) - 2f(x)$$

zero in areas of constant intensity

Nonzero at

- onset of ramp and step
- end of ramp and step

zero along intensity ramps



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.

Slide credit: Yan Tong

First-order VS Second-order Derivative for Edge Detection

- First-order derivative produces thick edge along the direction of the transition
- Second-order derivative produces thinner edges

Slide credit: Yan Tong

Gradient for Image Sharpening

Direction of change

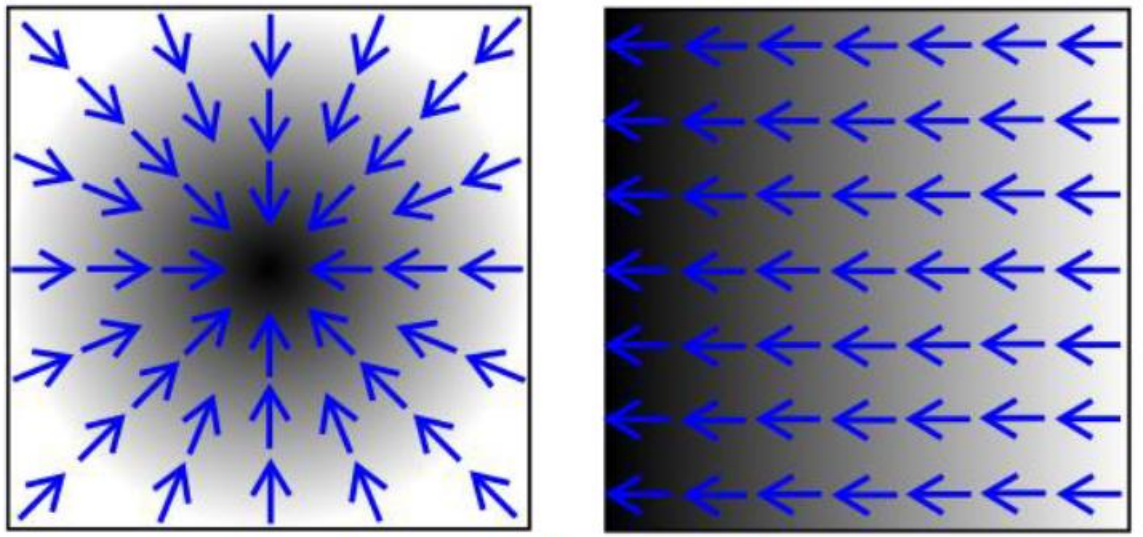
$$\nabla f = \text{grad}(f) = \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix}$$

Magnitude of change (gradient image)

$$M(x,y) = mag(\nabla f)$$

$$= \sqrt{g_x^2 + g_y^2}$$

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



http://en.wikipedia.org/wiki/Image_gradient

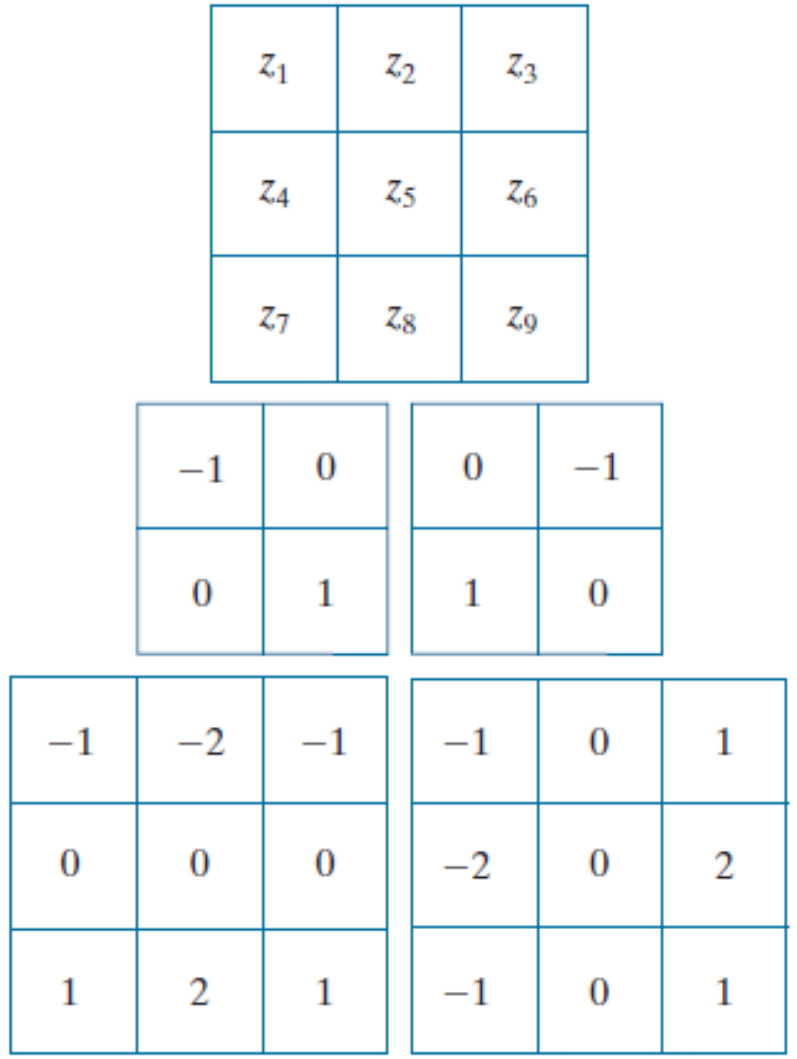
Slide credit: Yan Tong

Gradient for Image Sharpening

Sum of the coefficients is 0 – the response of a constant region is 0

Edge detectors:

- Roberts cross – fast while sensitive to noise
- Sobel - smooth



a
b c
d e

FIGURE 3.50
(a) A 3×3 region of an image, where the zs are intensity values. (b)–(c) Roberts cross-gradient operators. (d)–(e) Sobel operators. All the kernel coefficients sum to zero, as expected of a derivative operator.

Slide credit: Yan Tong

Gradient Operators

- Robert's Operator:
  - the operator consists of a pair of 2×2 convolution masks

+1	0
0	-1

G_x

0	+1
-1	0

G_y

- Perwitt's Operatior:
  - the operator uses two 3×3 convolution mask.

-1	0	+1
-1	0	+1
-1	0	+1

G_x

-1	-1	-1
0	0	0
+1	+1	+1

G_y

Slide credit: Ashish Ghosh

Examples: Gradient Operators

Original Image

Roberts operator

Perwitt operator

Slide credit: Ashish Ghosh

Sobel Operator

- The operator uses two 3×3 convolution mask.
- Doesn't make a difference for edge detection

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

In the X-direction

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

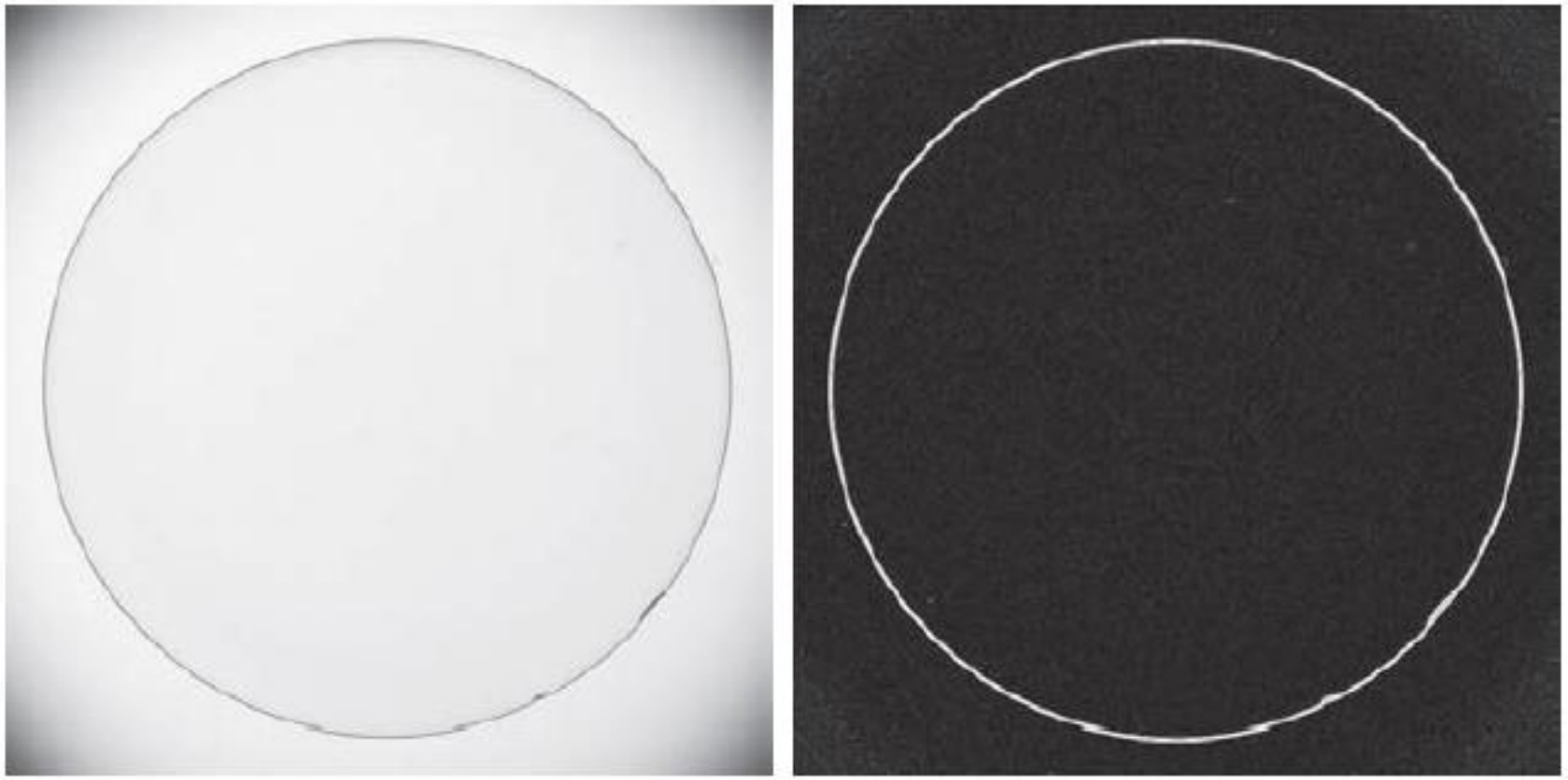
In the Y-direction

Slide credit: Ashish Ghosh

Effect of applying the Sobel Operator: Edge Enhancement

a b

FIGURE 3.51
(a) Image of a contact lens (note defects on the boundary at 4 and 5 o'clock).
(b) Sobel gradient.
(Original image courtesy of Perceptics Corporation.)



Gradient for Image Sharpening

2D Isotropic filters – rotation invariant

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

Laplacian operator

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

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

Image sharpening with Laplacian

0	1	0	1	1	1
1	−4	1	1	−8	1
0	1	0	1	1	1
0	−1	0	−1	−1	−1
−1	4	−1	−1	8	−1
0	−1	0	−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.

Slide credit: Yan Tong

Laplacian Operator

- The simplest isotropic (rotation invariant) derivative operator is the Laplacian.

  It is defined by:

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

- The partial second-order derivatives in the x and y-direction are:

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

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

- The digital implementation of the 2-D Laplacian operator is obtained by summing these two components to get:

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

Slide credit: Ashish Ghosh

Laplacian Mask

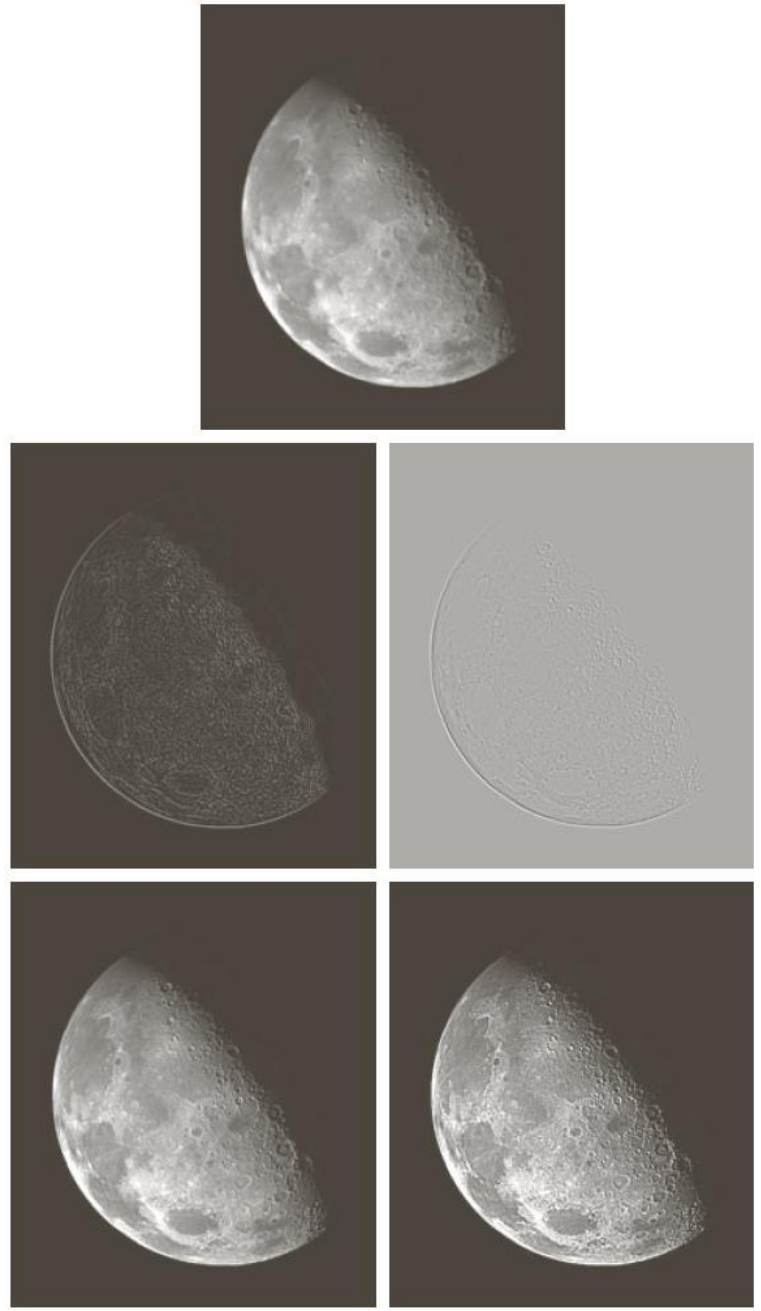
-1	-1	-1
-1	8	-1
-1	-1	-1

- The filter has positive coefficients near the center and negative in the outer periphery.
- The sum of the coefficients is 0, indicating that when the filter is passing over regions of almost uniform gray levels, the output of the mask is very small.
- Some scaling must be done to the final result to compensate for possible negative gray levels after filtering.

Slide credit: Ashish Ghosh

Gradient for Image Sharpening

Scale the Laplacian by shifting the intensity range to [0, L-1]

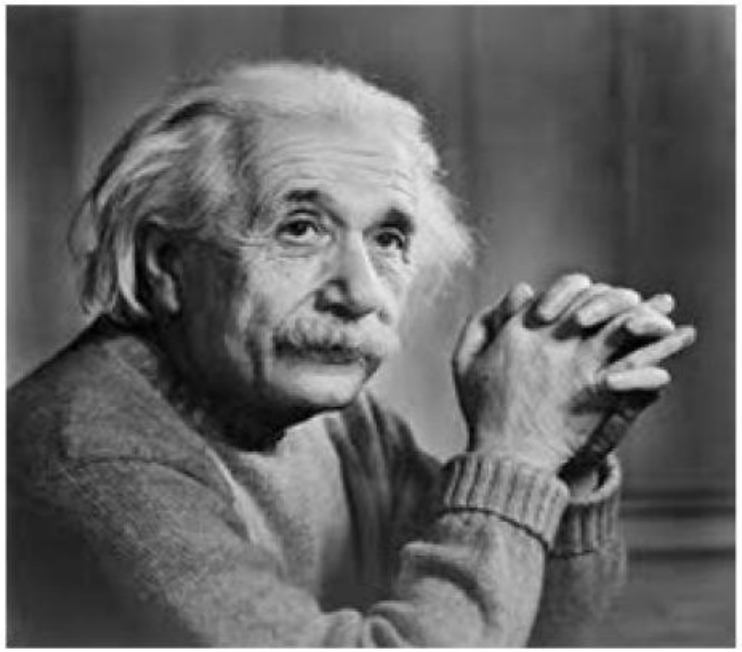


a
b c
d e

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.)

Slide credit: Yan Tong

The Laplacian



Original Image

Laplacian Image

Slide credit: Ashish Ghosh

The Laplacian

- To recover background features the image must be added back to the Laplacian filtered image. Thus we get:

$$g(x, y) = f(x, y) + \nabla^2 f(x, y)$$

- Thus the composite Laplacian mask would be:

-1	-1	-1
-1	8	-1
-1	-1	-1

+

0	0	0
0	1	0
0	0	0

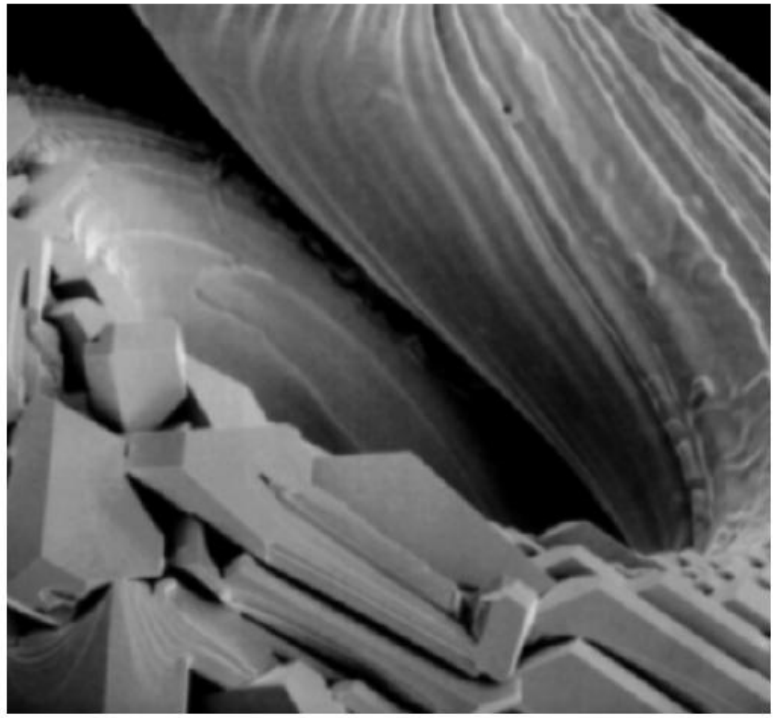
=

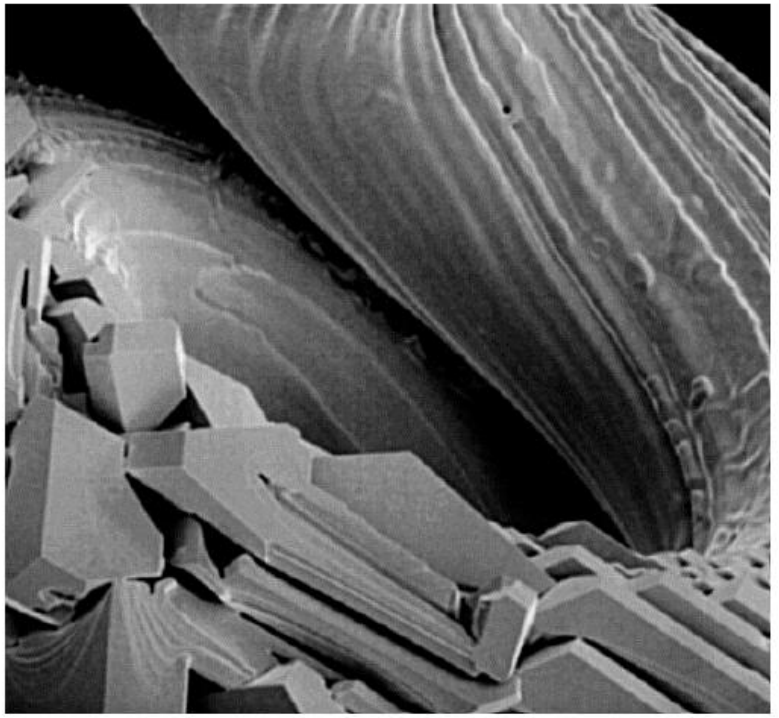
-1	-1	-1
-1	9	-1
-1	-1	-1

Slide credit: Ashish Ghosh

The Laplacian

Effect of applying the composite Laplacian filter.



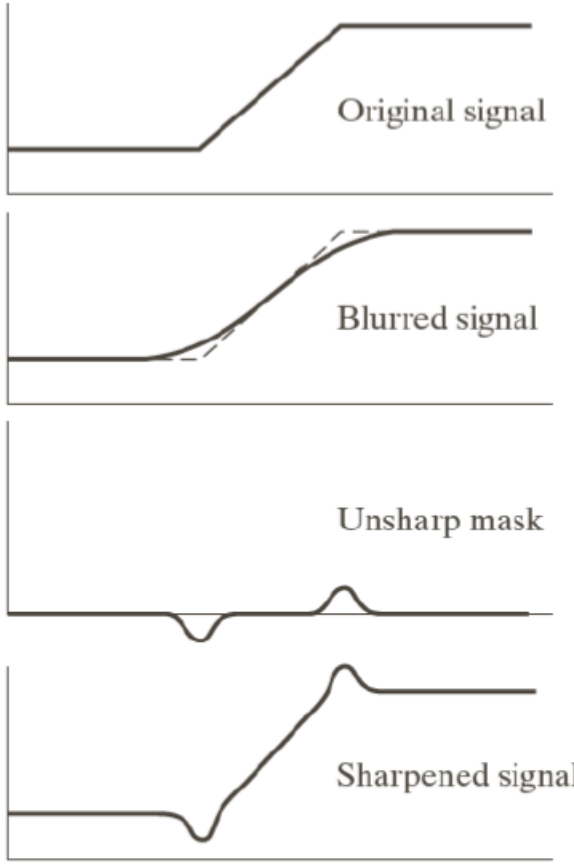


Observe the sharpened result.

Slide credit: Ashish Ghosh

Image Sharpening by Unsharp Masking and Highboost Filtering

1. **Blur the original image**
2. **Subtract the blurred image from the original to get the mask**
3. **Add the mask to the original**



a
b
c
d

FIGURE 3.39 1-D illustration of the mechanics of unsharp masking. (a) Original signal. (b) Blurred signal with original shown dashed for reference. (c) Unsharp mask. (d) Sharpened signal, obtained by adding (c) to (a).

Slide credit: Yan Tong

Image Sharpening by Unsharp Masking and Highboost Filtering

$$g(x,y) = f(x,y) + k*\left(f(x,y) - \bar{f}(x,y)\right) \quad k \geq 0$$

When $k > 1$, it becomes a highboost filtering.

Example: when $k = 1$

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \left(\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} - \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \right) = \frac{1}{9}\begin{bmatrix} -1 & -1 & -1 \\ -1 & 17 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

Sum of the coefficients is 1

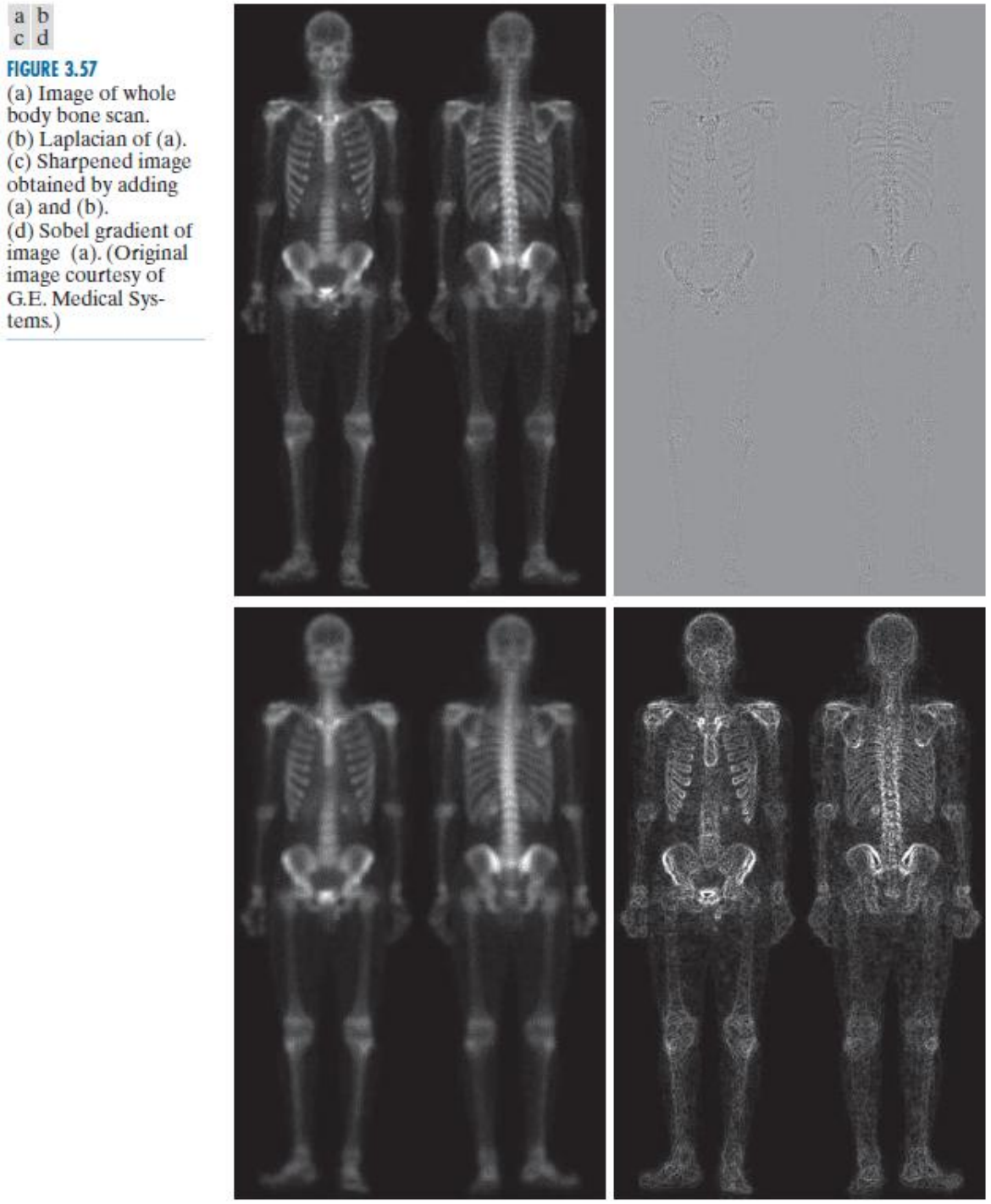
Slide credit: Yan Tong

Example: Image Sharpening by Unsharp Masking and Highboost Filtering

a b c
d e

FIGURE 3.49 (a) Original image of size 600×259 pixels. (b) Image blurred using a 31×31 Gaussian lowpass filter with $\sigma = 5$. (c) Mask. (d) Result of unsharp masking using Eq. (3-56) with $k = 1$. (e) Result of highboost filtering with $k = 4.5$.

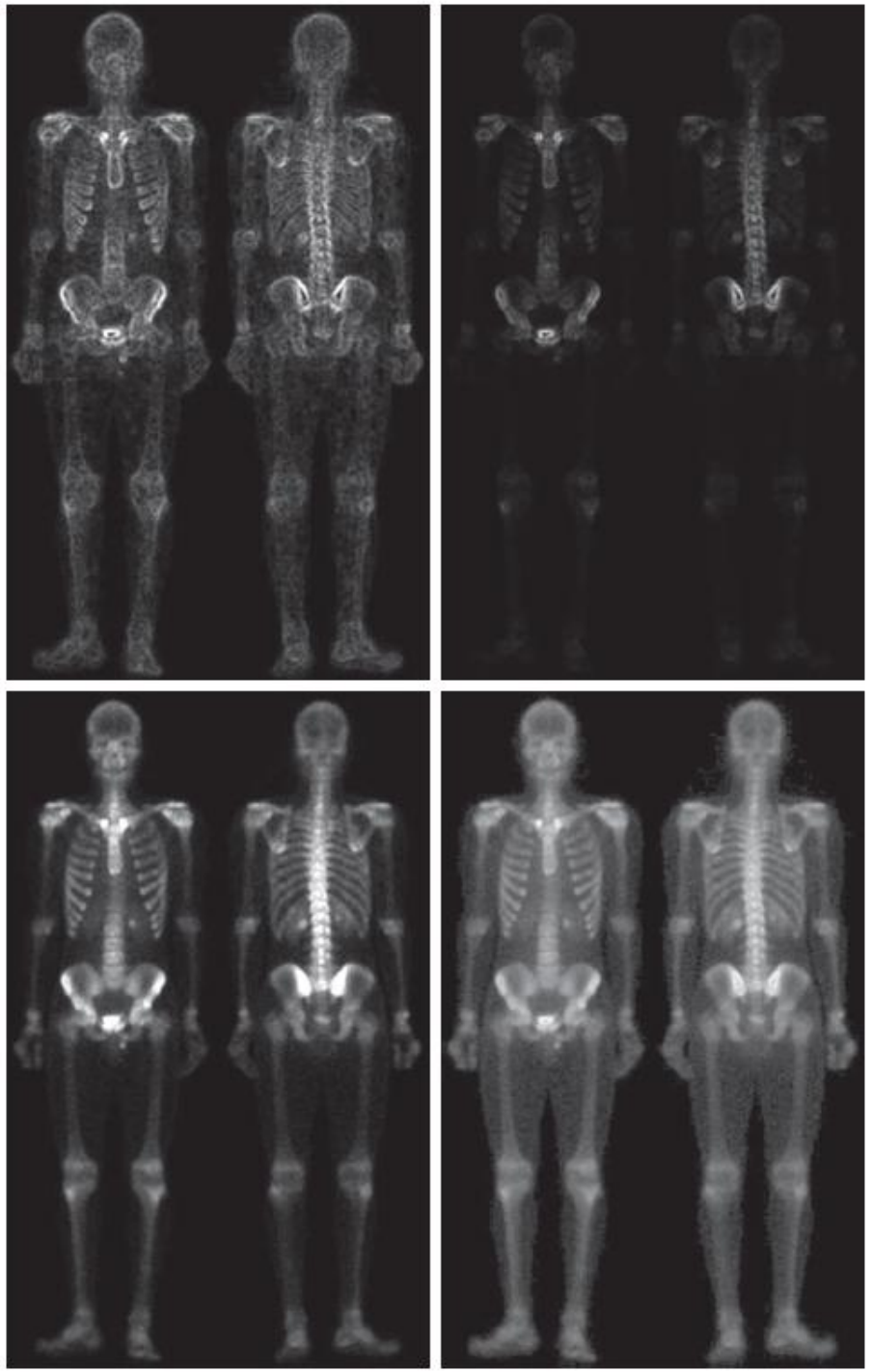
Combining Spatial Enhancement Methods



a b
c d

FIGURE 3.57
(a) Image of whole body bone scan.
(b) Laplacian of (a).
(c) Sharpened image obtained by adding (a) and (b).
(d) Sobel gradient of image (a). (Original image courtesy of G.E. Medical Systems.)

Combining Spatial Enhancement Methods



e f
g h

FIGURE 3.57
(Continued)
(e) Sobel image smoothed with a 5×5 box filter. (f) Mask image formed by the product of (b) and (e). (g) Sharpened image obtained by the adding images (a) and (f). (h) Final result obtained by applying a power-law transformation to (g). Compare images (g) and (h) with (a). (Original image courtesy of G.E. Medical Systems.)

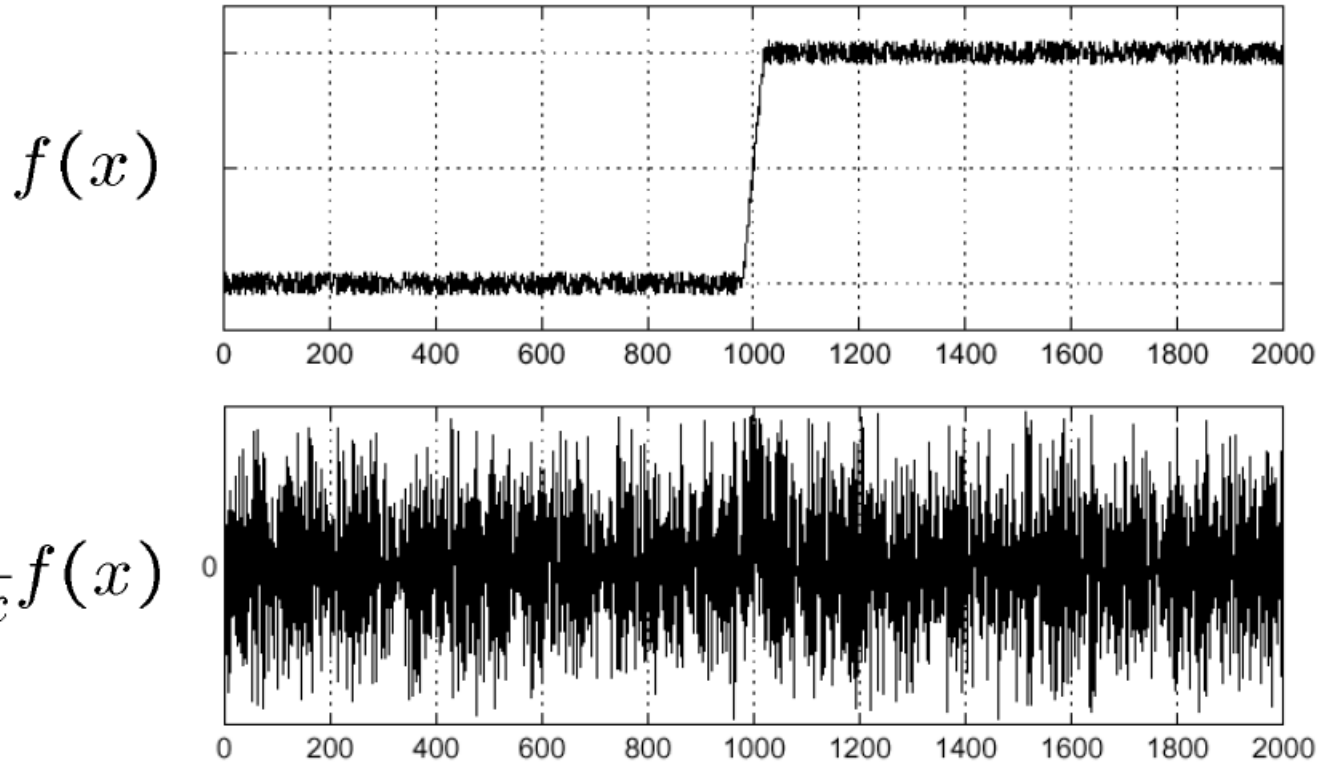
Laplacian of Gaussian

- Laplace operator may detect edges as well as noise (isolated, out-of-range).
- It may be desirable to smooth the image first by a convolution with a Gaussian kernel and then detect the edges.

Slide credit: Ashish Ghosh

Effects of Noise

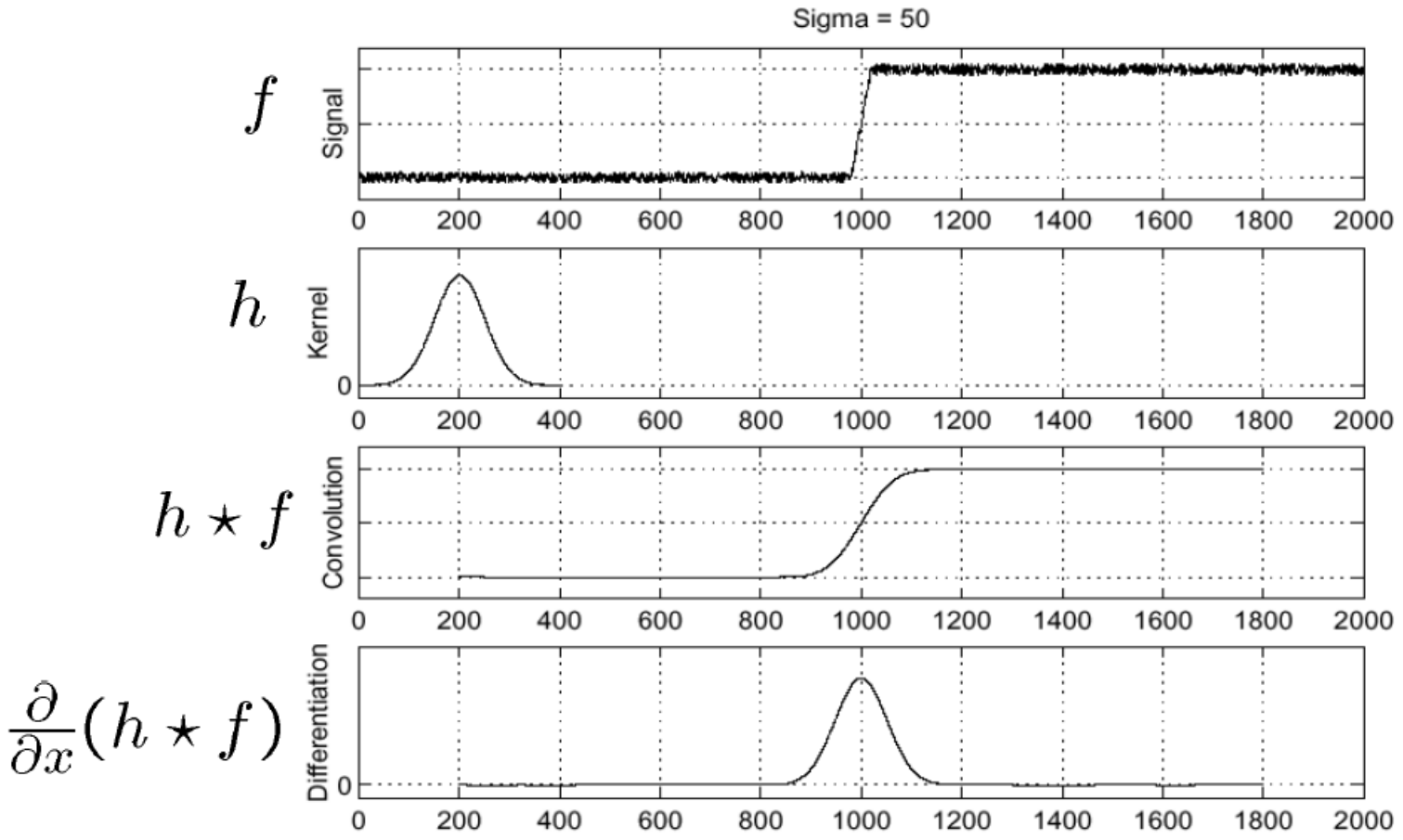
- Consider a single row or column of the image
  - Plotting intensity as a function of position gives a signal



Where is the edge??

Slide credit: Ashish Ghosh

Solution: Smooth First



Where is the edge?

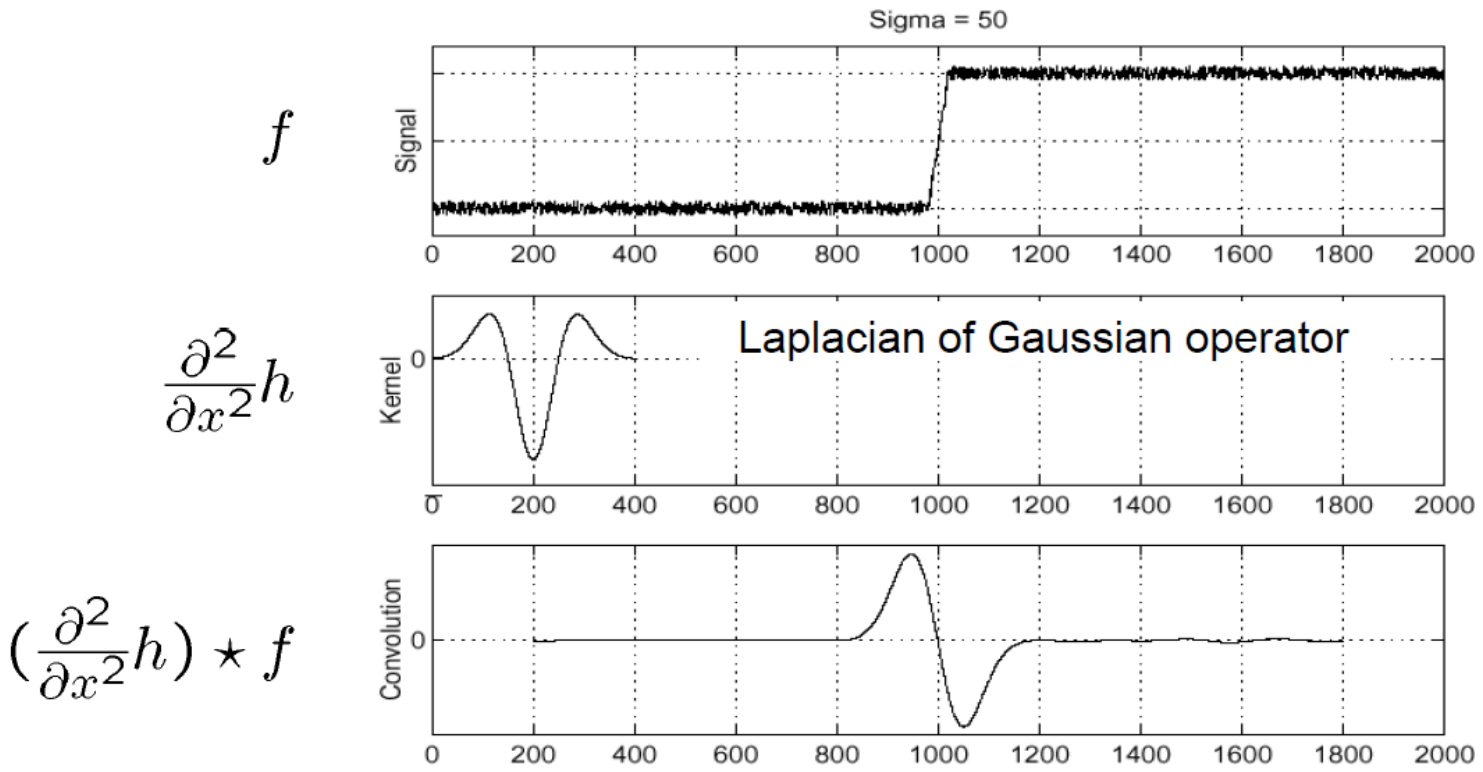
Look for peaks in $\frac{\partial}{\partial x}(h \star f)$

Slide credit: Ashish Ghosh

Laplacian of Gaussian (LoG)

$$\frac{\partial^2}{\partial x^2}(h * f) = \left(\frac{\partial^2}{\partial x^2} h\right) * f$$

Laplacian of Gaussian

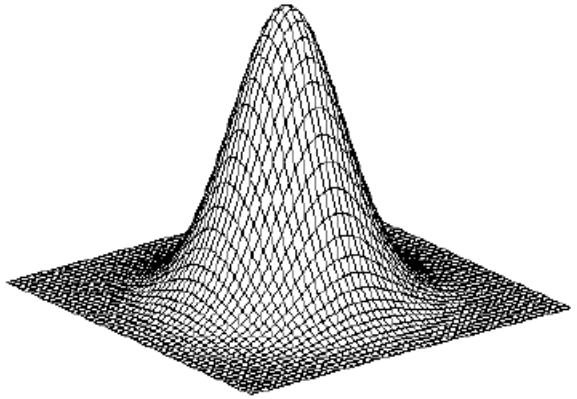


Where is the edge?

Zero-crossings of bottom graph !

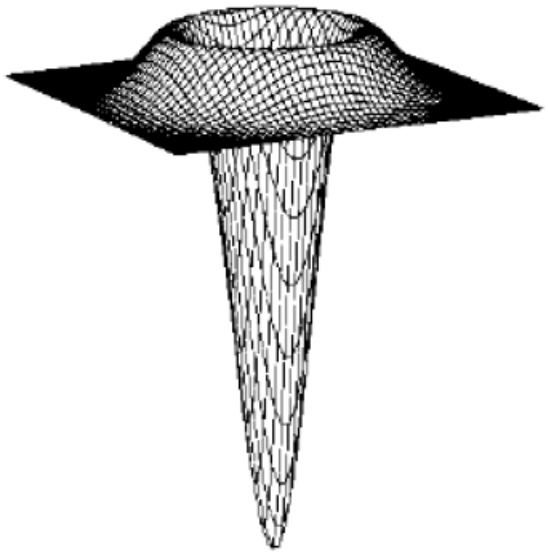
Slide credit: Ashish Ghosh

Laplacian of Gaussian (LoG)



$$h(r) = -e^{-\frac{r^2}{2\sigma^2}}$$

Gaussian



$$\nabla^2 h_\sigma(r)$$

Laplacian of Gaussian

$$\nabla^2 h_\sigma(r) = -\left[\frac{r^2 - \sigma^2}{\sigma^4}\right] e^{-\frac{r^2}{2\sigma^2}}; where\ r^2 = u^2 + v^2$$

Slide credit: Ashish Ghosh

Laplacian of Gaussian (LoG)

0	1	1	2	2	2	1	1	0
1	2	4	5	5	5	4	2	1
1	4	5	3	0	3	5	4	1
2	5	3	-12	-24	-12	3	5	2
2	5	0	-24	-40	-24	0	5	2
2	5	3	-12	-24	-12	3	5	2
1	4	5	3	0	3	5	4	1
1	2	4	5	5	5	4	2	1
0	1	1	2	2	2	1	1	0

LoG convolution mask with σ =1.4

Slide credit: Ashish Ghosh