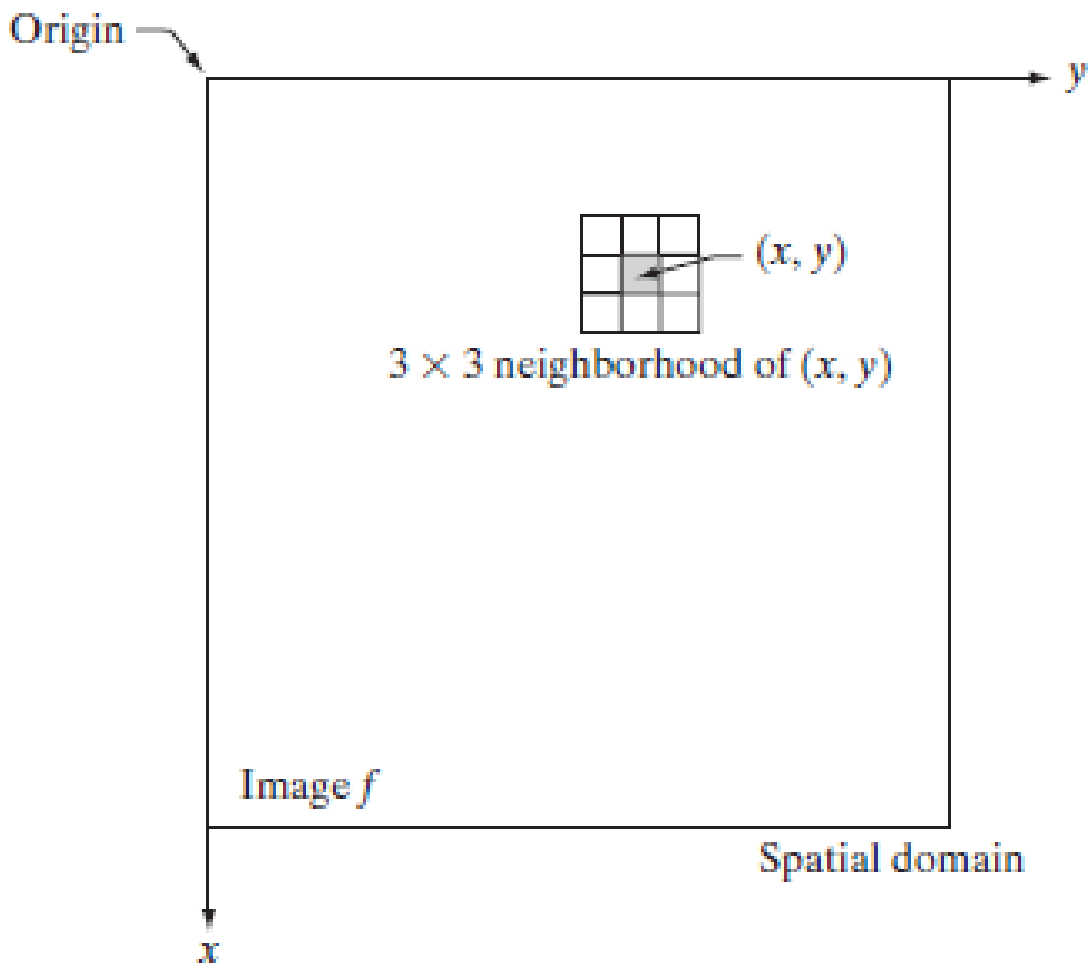
EE610 – Image Processing

Amit Sethi
asethi, 7483

Spatial domain is defined as the domain of pixels in the image grid



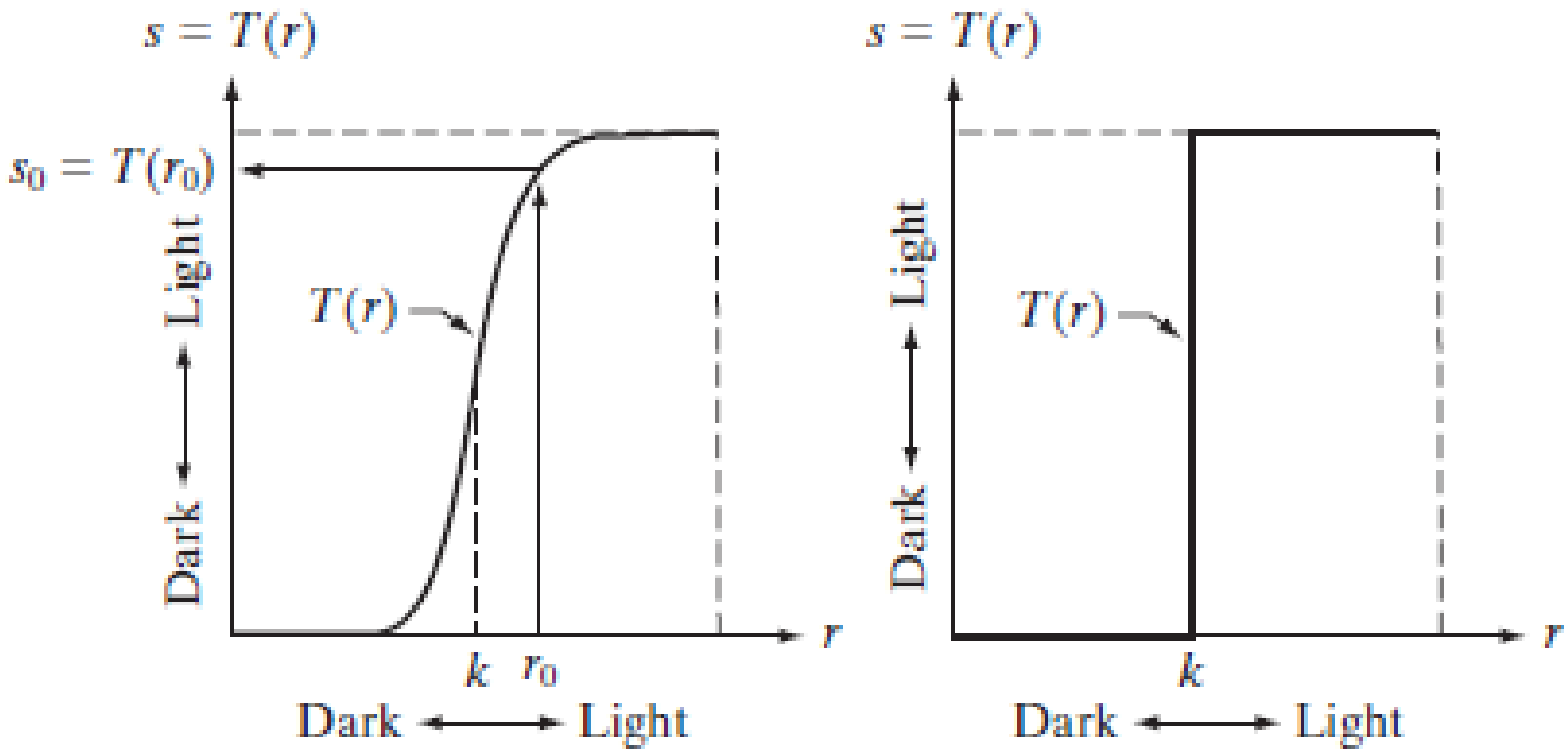
Source: Digital Image Processing, Gonzalez and Woods, Prentice Hall

A special case

- Process each pixel with a W×W window around it
- W = 2M+1

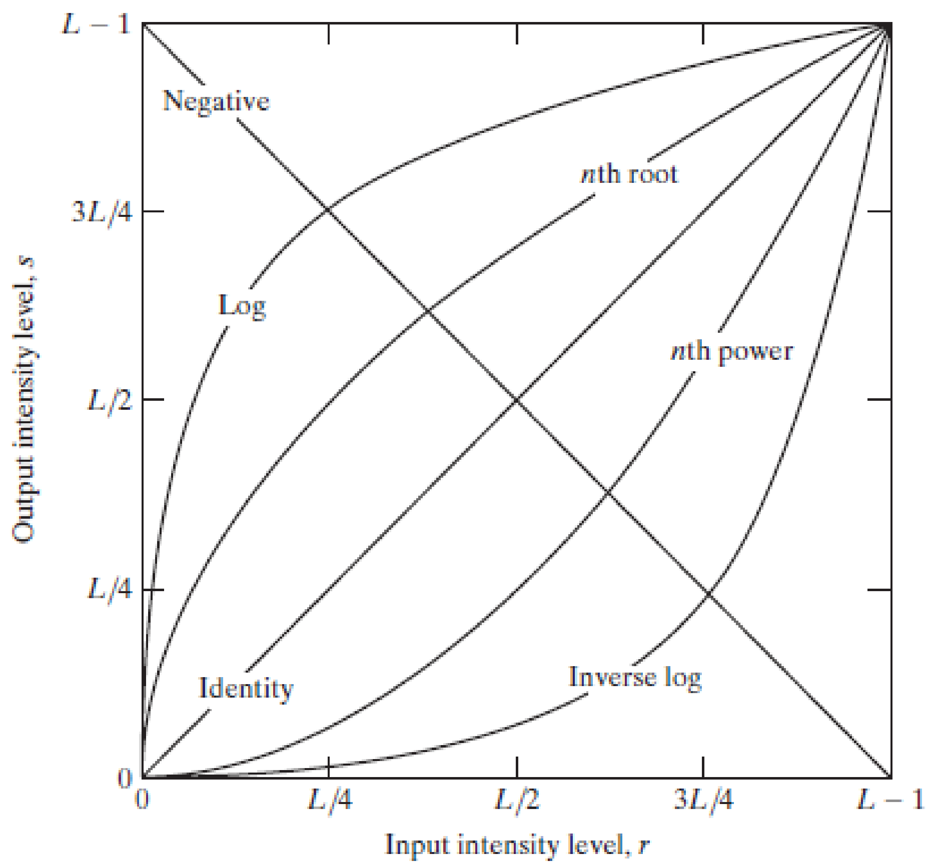
- When M=0,
  - $g(x,y) = T\,[\,f(x,y)\,]$ a.k.a. Intensity Transform
- Else,
  - $g(x,y) = T\,[\,N_f(x,y)\,]$

Contrast stretching and thresholding



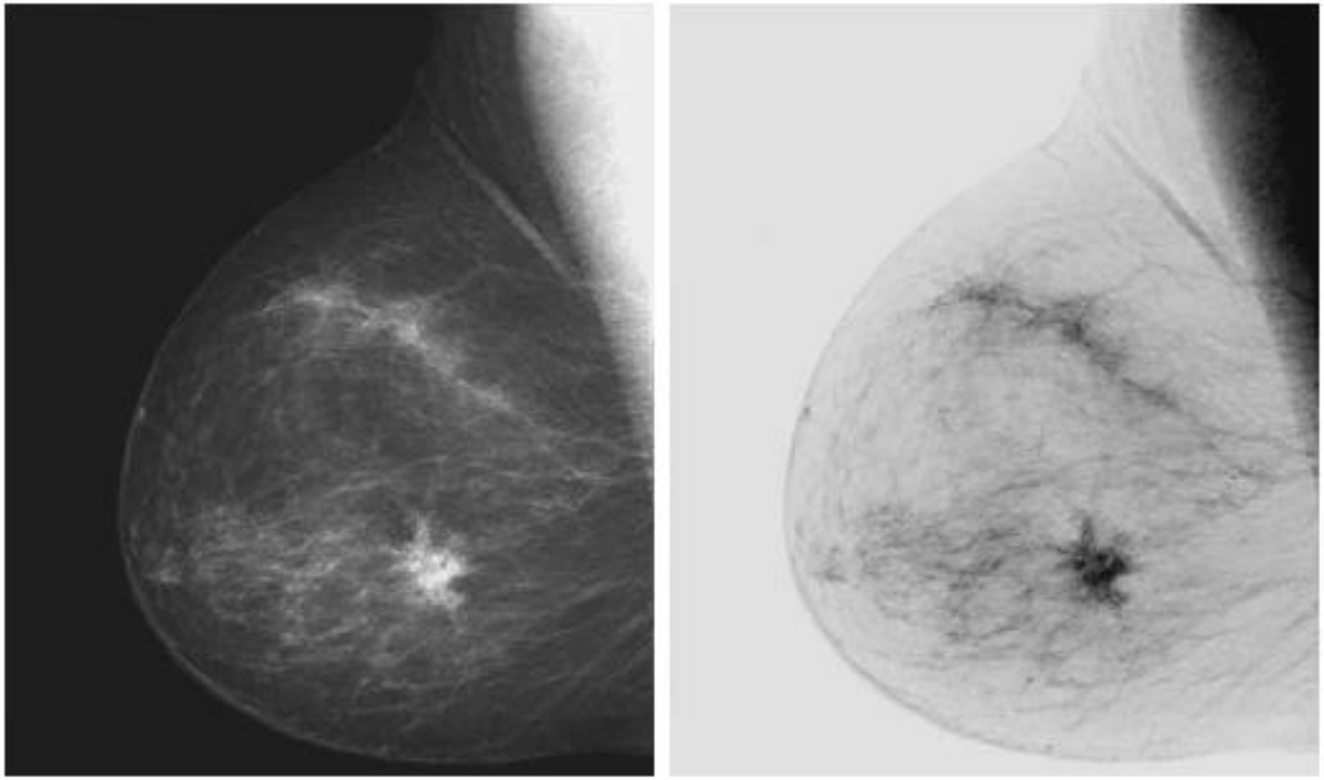
Source: Digital Image Processing, Gonzalez and Woods, Prentice Hall

Some basic intensity transformations



Source: Digital Image Processing, Gonzalez and Woods, Prentice Hall

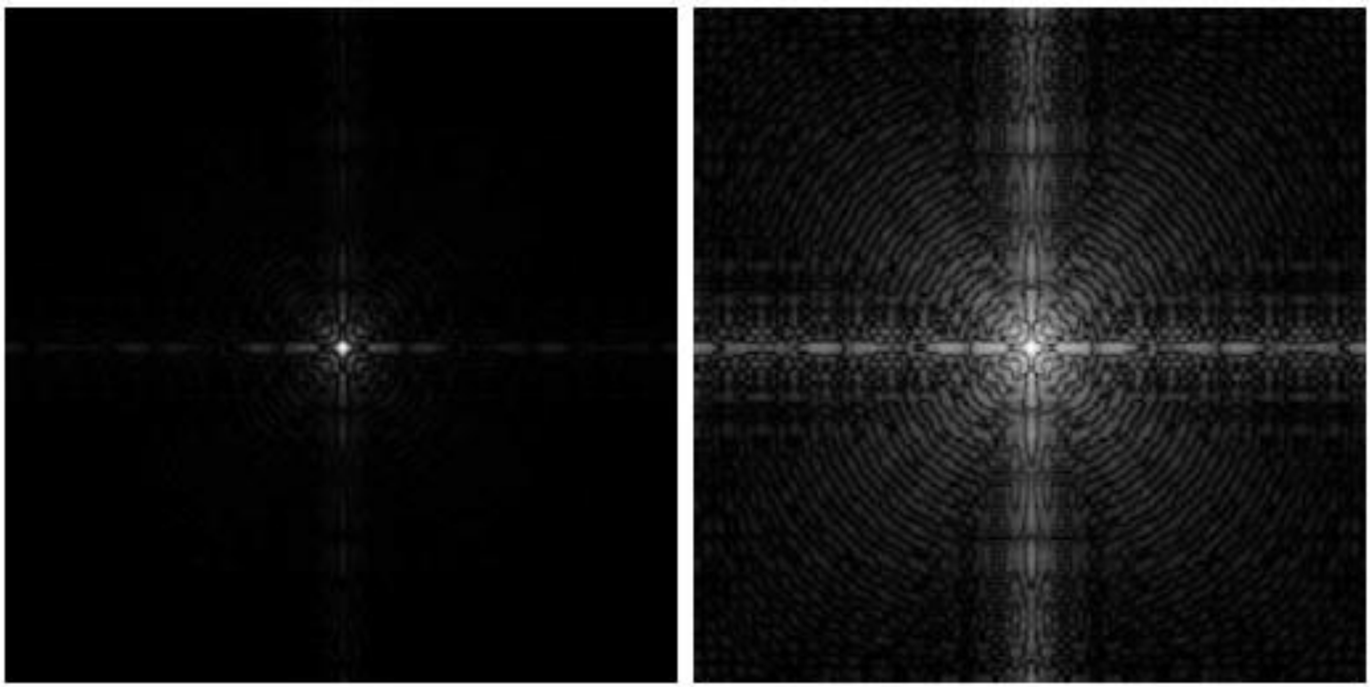
Negative of an image



Source: Digital Image Processing, Gonzalez and Woods, Prentice Hall

Log transformation in log domain

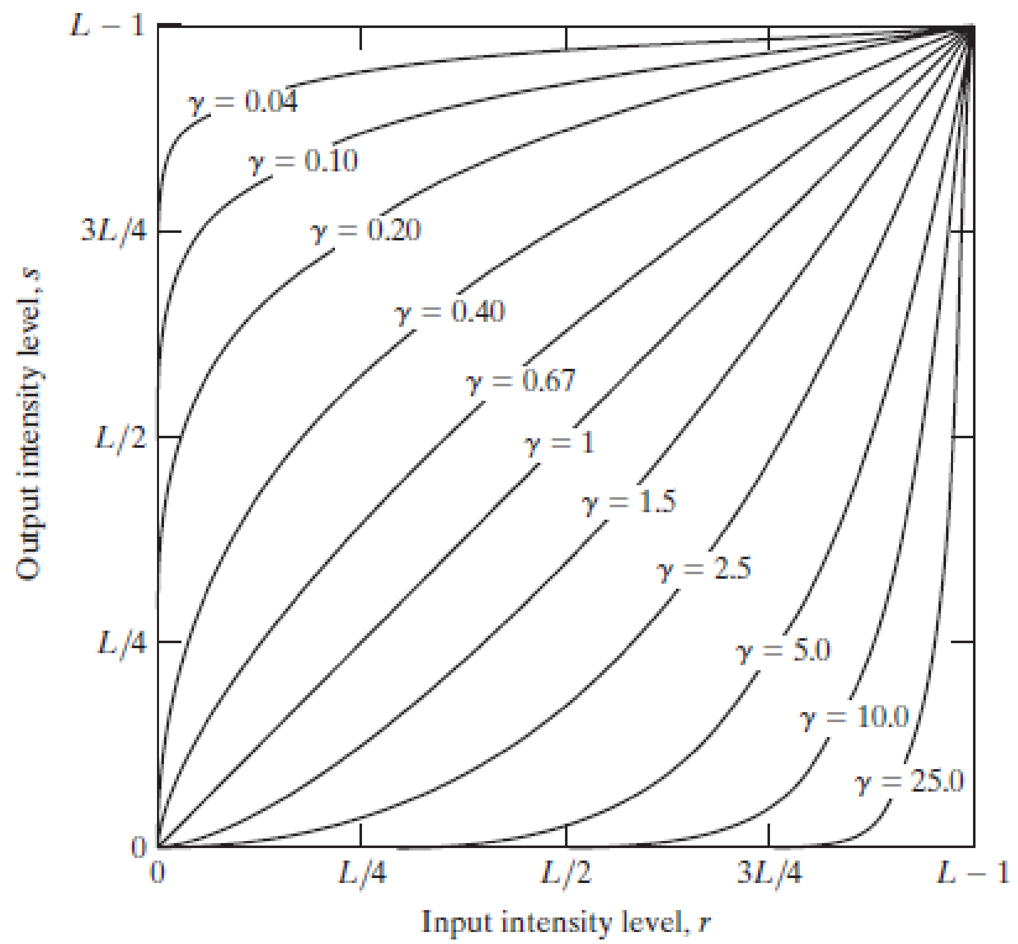
- $s = c \log (1+r)$



Source: Digital Image Processing, Gonzalez and Woods, Prentice Hall

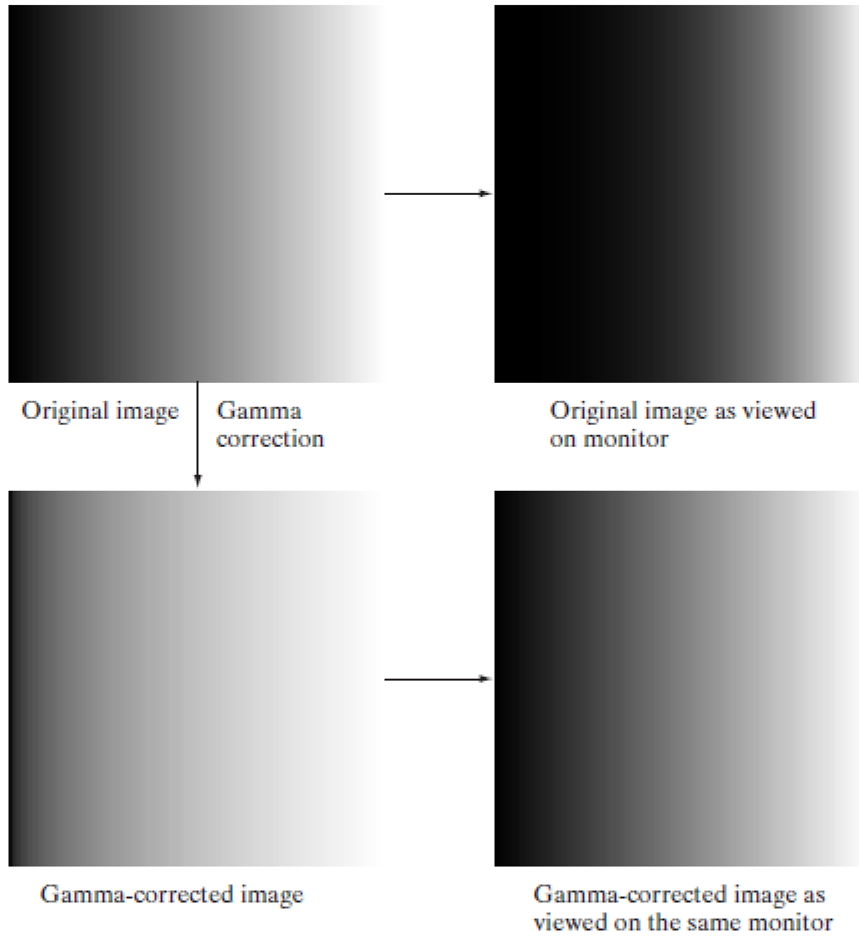
Power-law (gamma transformation)

- $s = c\, r^{\gamma}$



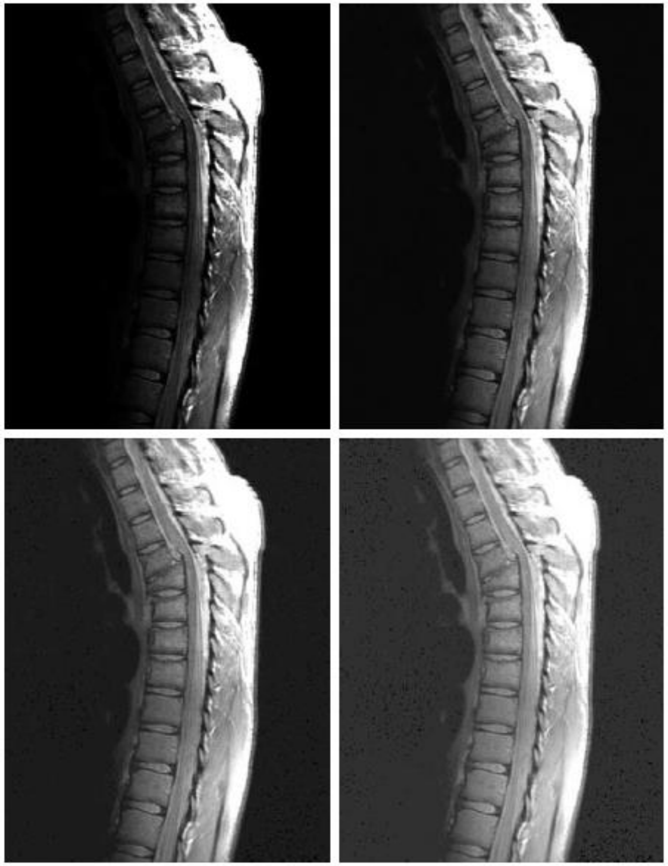
Source: Digital Image Processing, Gonzalez and Woods, Prentice Hall

CRT's needed gamma correction



Source: Digital Image Processing, Gonzalez and Woods, Prentice Hall

Gamma 1, .6, .4, .3

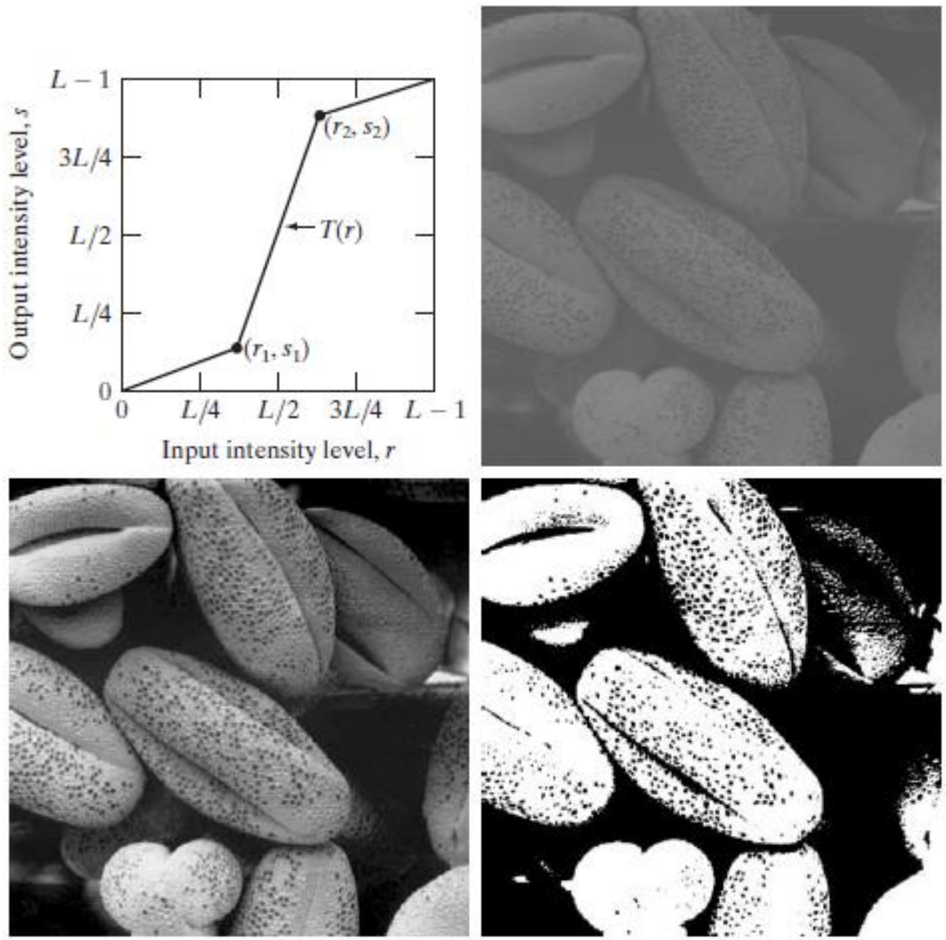


Source: Digital Image Processing, Gonzalez and Woods, Prentice Hall

Gamma 1, 3, 4, 5

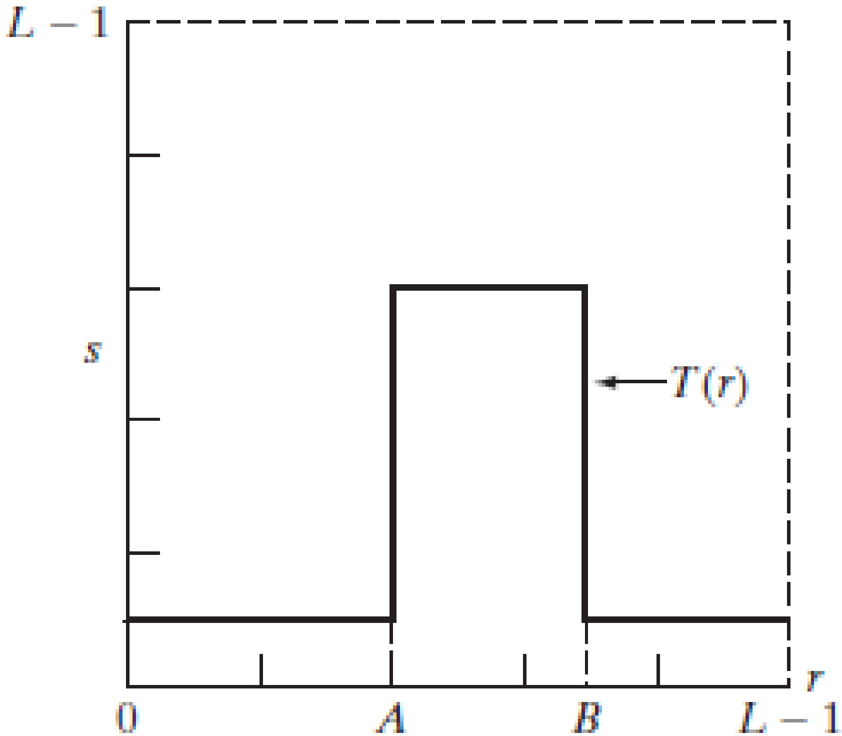
Source: Digital Image Processing, Gonzalez and Woods, Prentice Hall

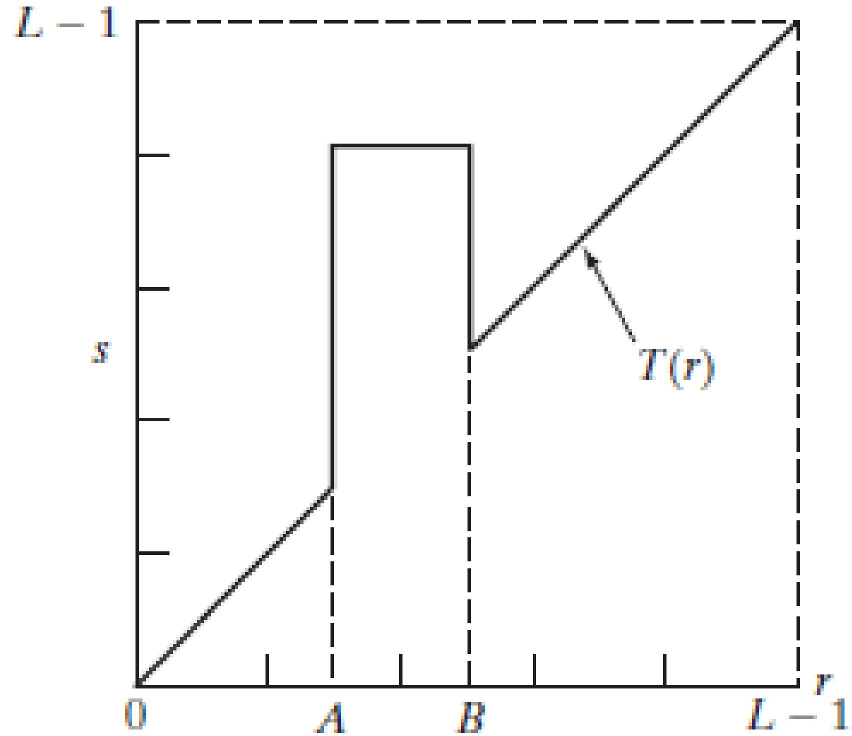
Contrast stretching vs. thresholding



Source: Digital Image Processing, Gonzalez and Woods, Prentice Hall

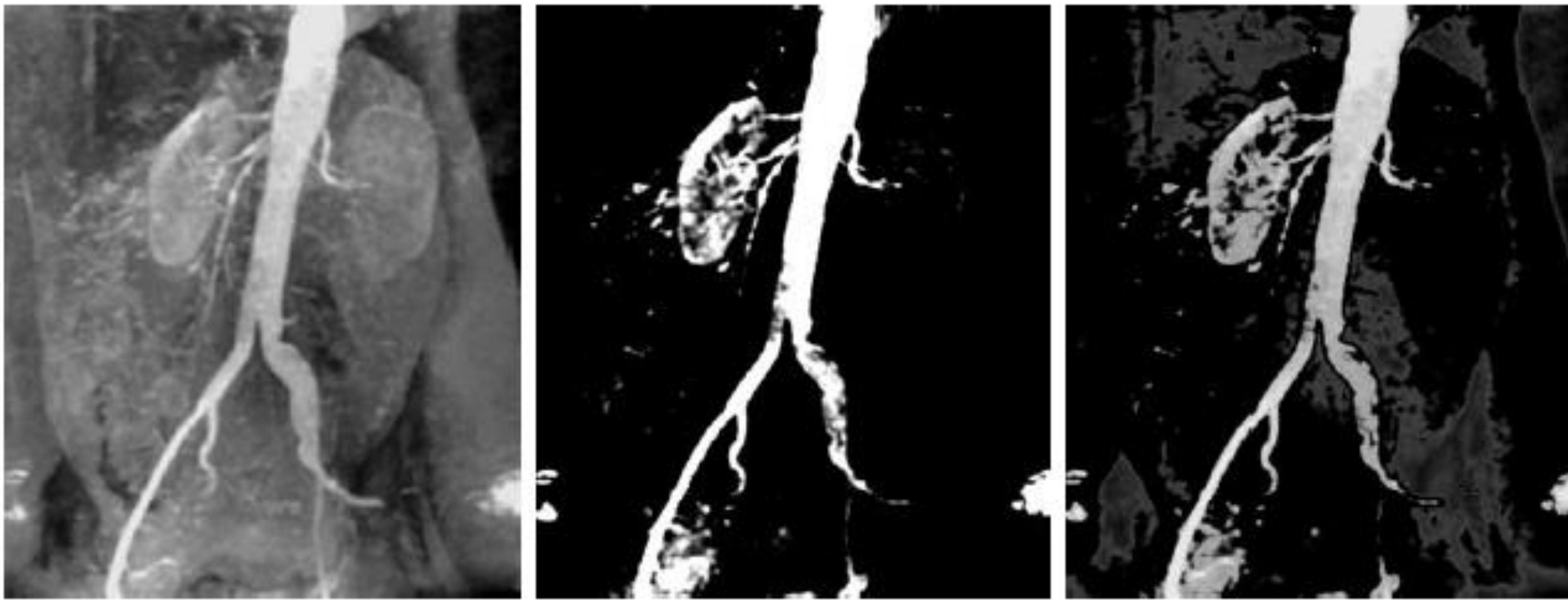
Highlighting a range of intensities





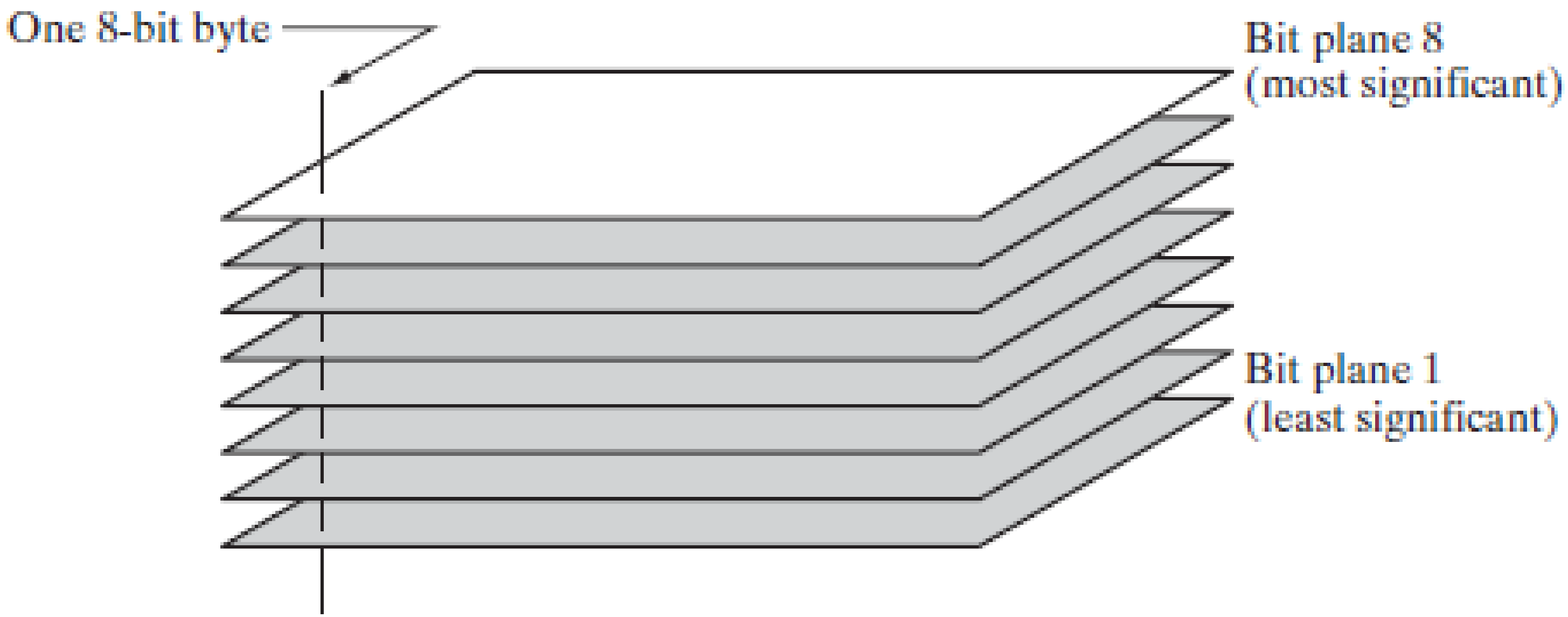
Source: Digital Image Processing, Gonzalez and Woods, Prentice Hall

Some results of intensity highlighting



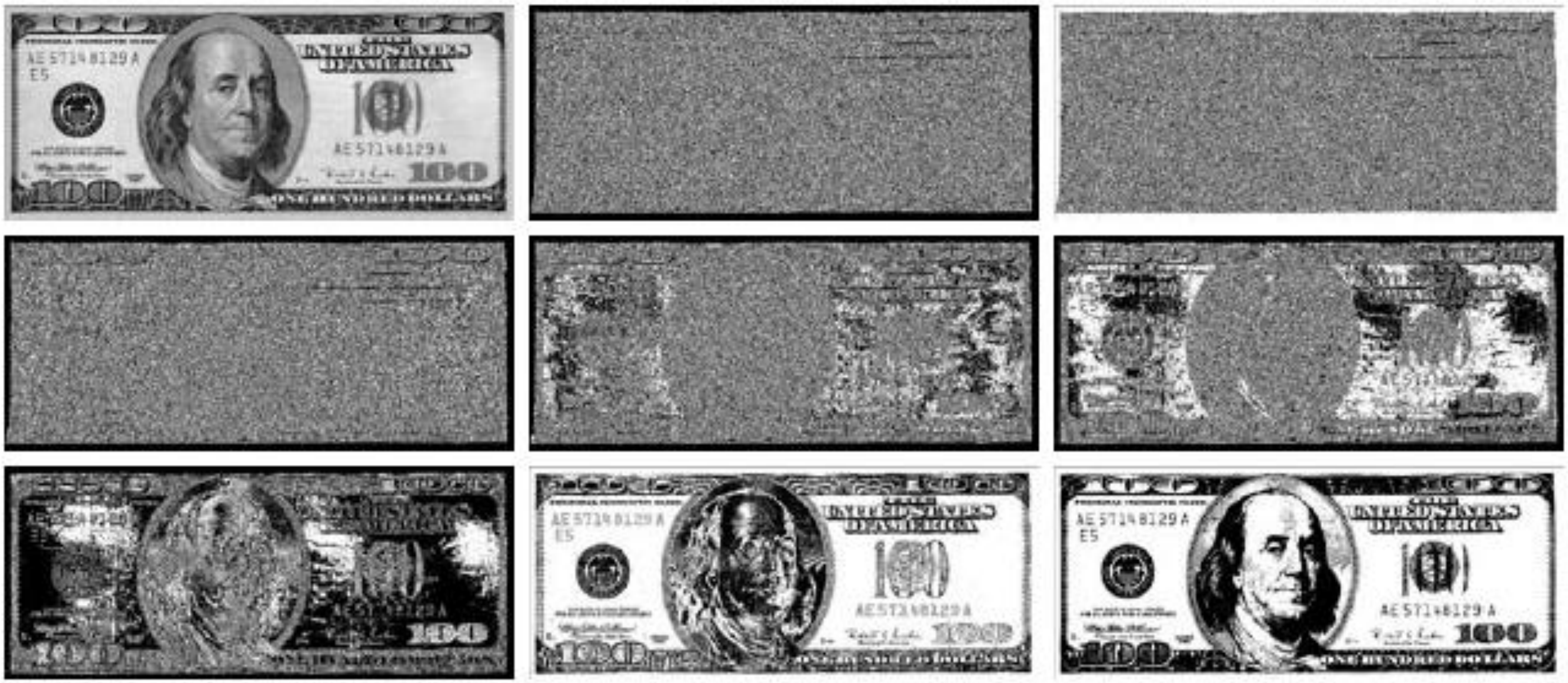
Source: Digital Image Processing, Gonzalez and Woods, Prentice Hall

Bit-plane splicing



Source: Digital Image Processing, Gonzalez and Woods, Prentice Hall

Different bit planes of an image



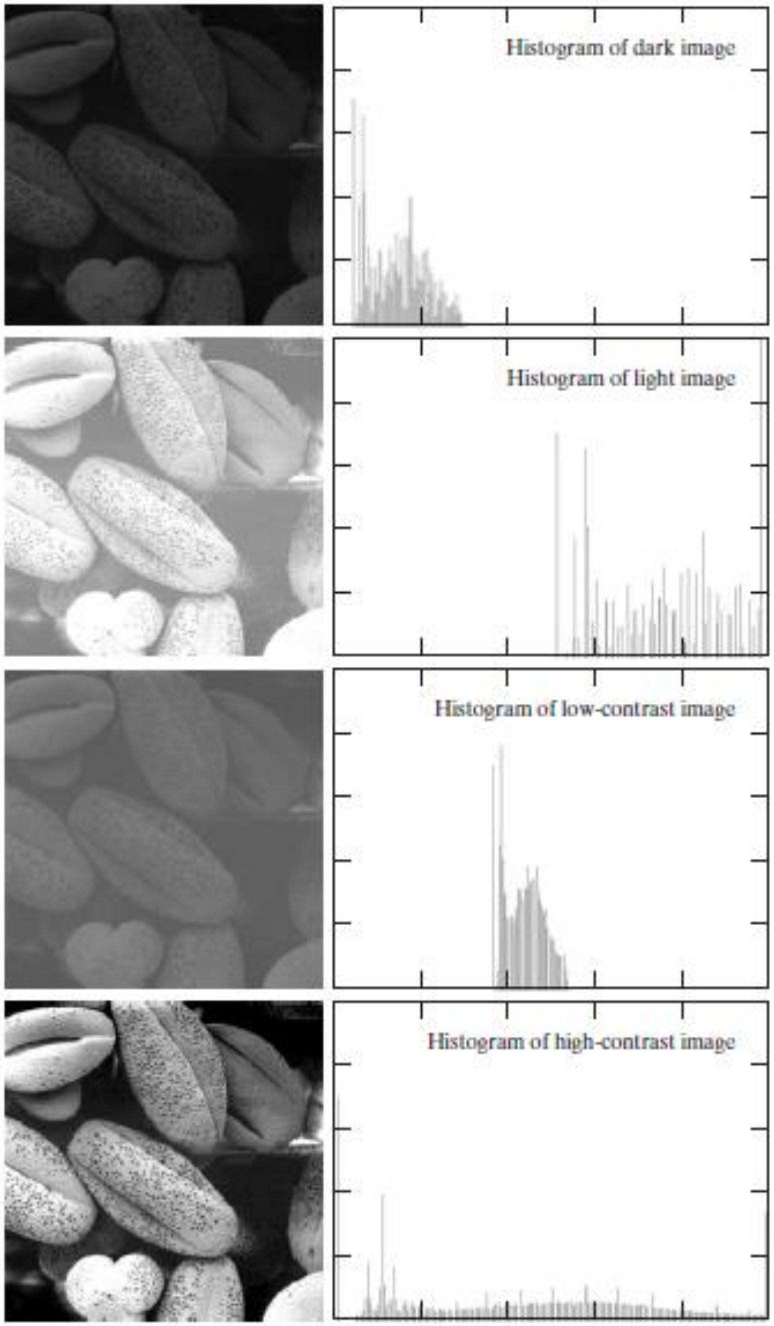
Source: Digital Image Processing, Gonzalez and Woods, Prentice Hall

Higher bit planes contain the most information



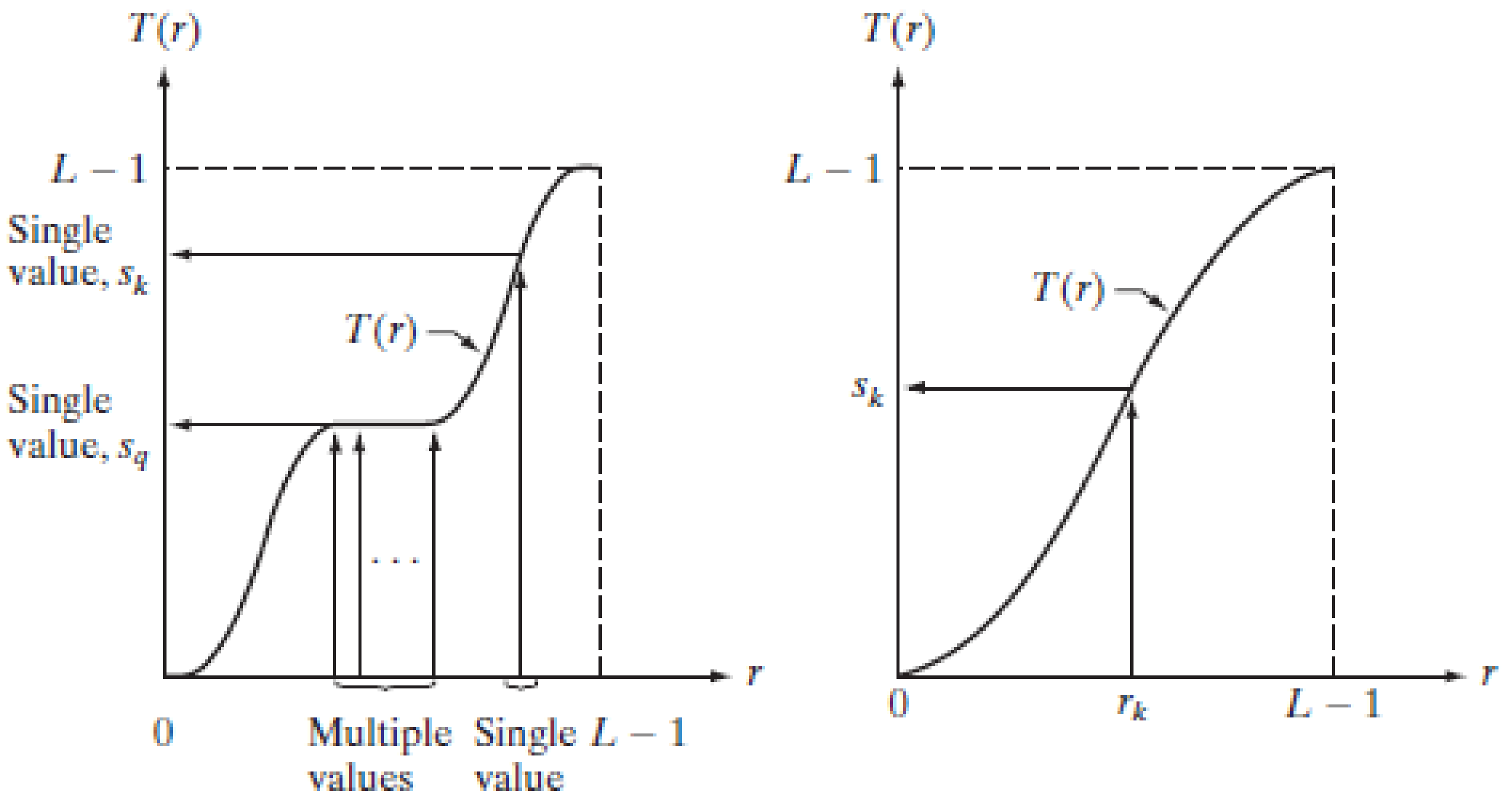
Source: Digital Image Processing, Gonzalez and Woods, Prentice Hall

Image versions and their histograms



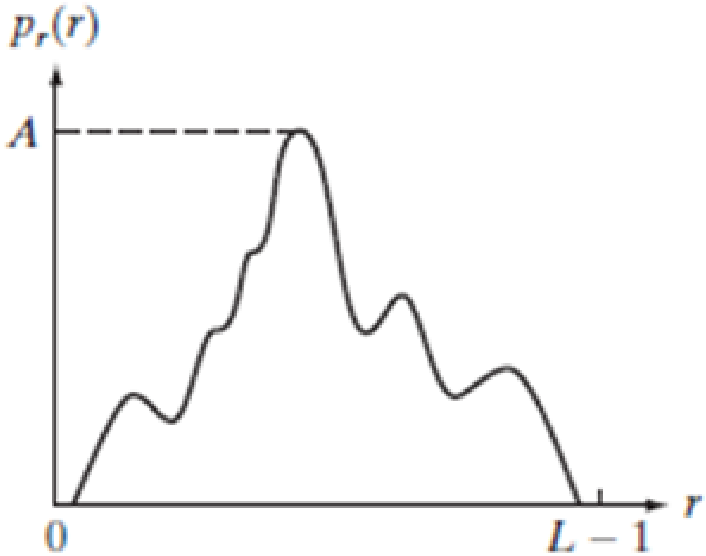
Source: Digital Image Processing, Gonzalez and Woods, Prentice Hall

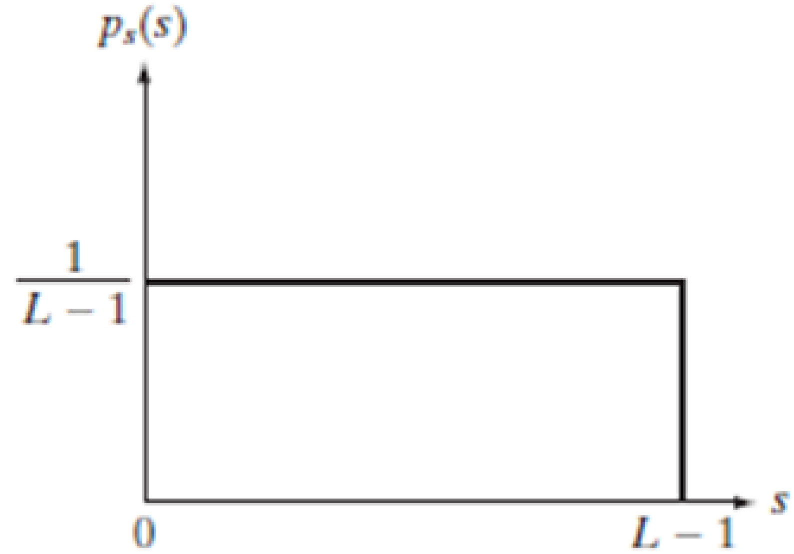
(Strictly) monotonically increasing functions for intensity transformations



Source: Digital Image Processing, Gonzalez and Woods, Prentice Hall

Goal of histogram equalization





Source: Digital Image Processing, Gonzalez and Woods, Prentice Hall

Relation between # pixels and probability of an intensity

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

Source: Digital Image Processing, Gonzalez and Woods, Prentice Hall

Relation between a random variable and its transform

$$s = T(r)$$

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

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

Source: Digital Image Processing, Gonzalez and Woods, Prentice Hall

Transform for equalization

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

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

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

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

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

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

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

Source: Digital Image Processing, Gonzalez and Woods, Prentice Hall

Discrete version of equalizing transformation

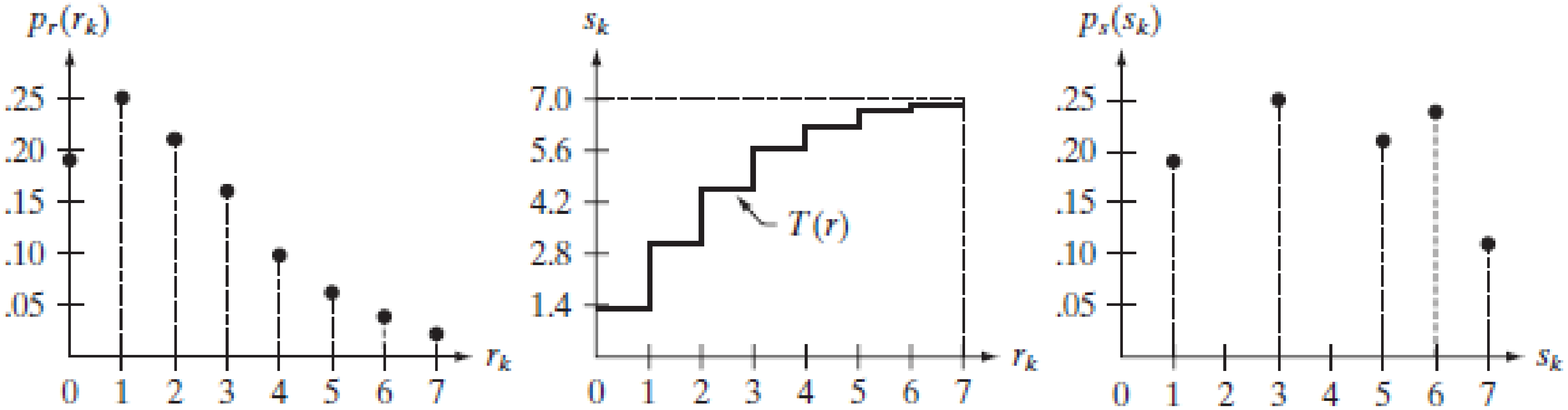
$$s_k = T(r_k) = (L-1)\sum_{j=0}^{k} p_r(r_j)$$

$$= \frac{(L-1)}{MN}\sum_{j=0}^{k} n_j \qquad k = 0, 1, 2, \ldots, L-1$$

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

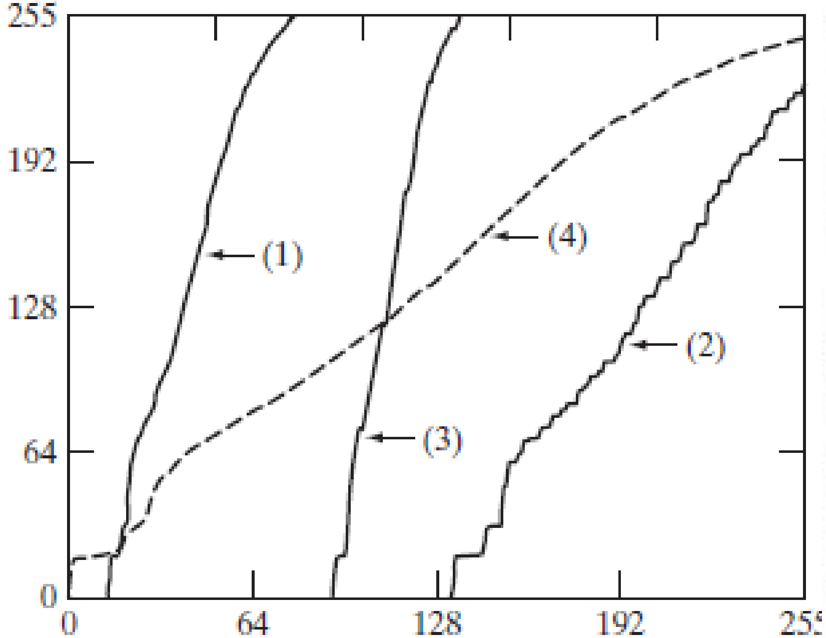
Source: Digital Image Processing, Gonzalez and Woods, Prentice Hall

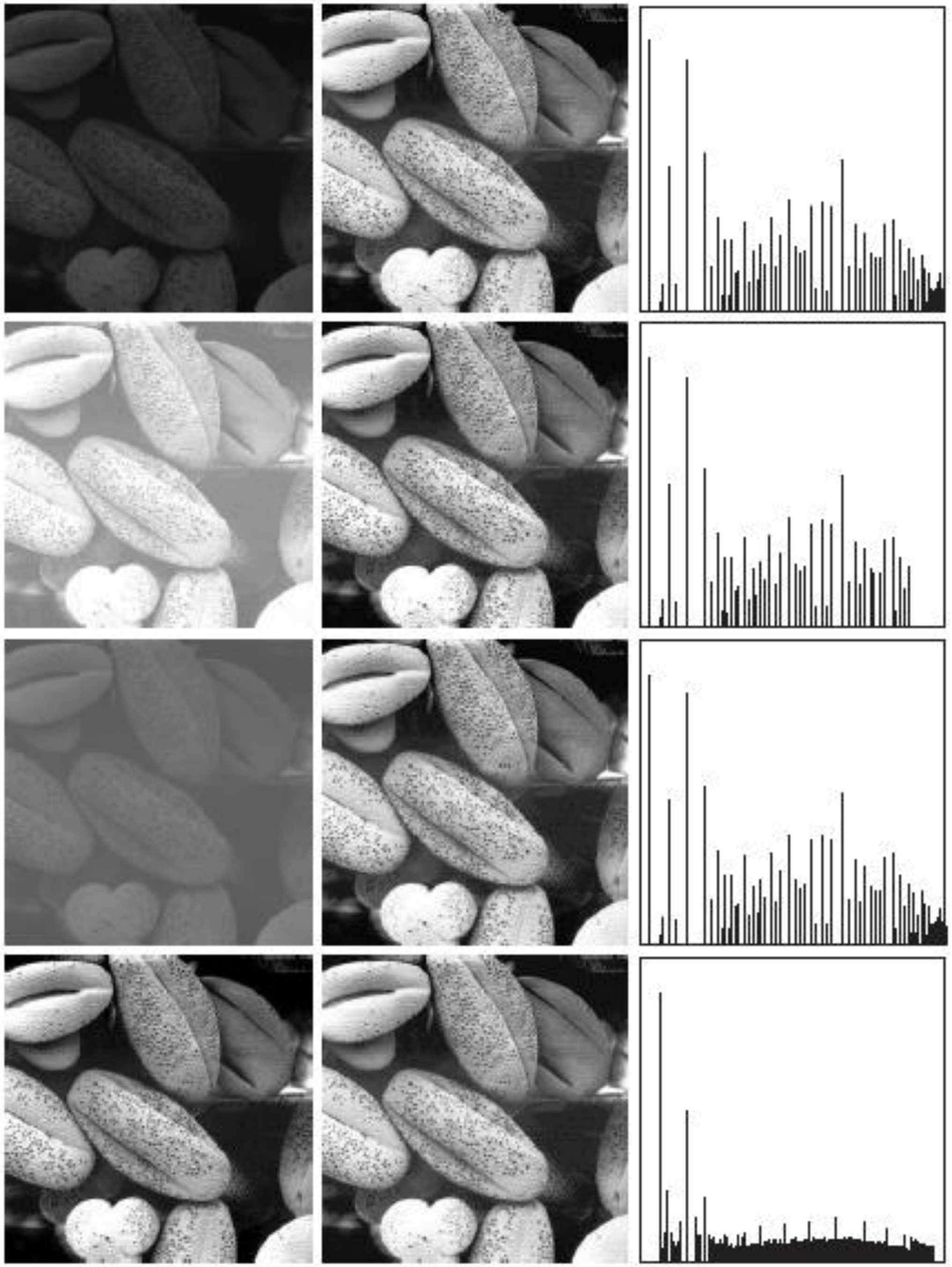
Equalizing a discrete histogram



Source: Digital Image Processing, Gonzalez and Woods, Prentice Hall

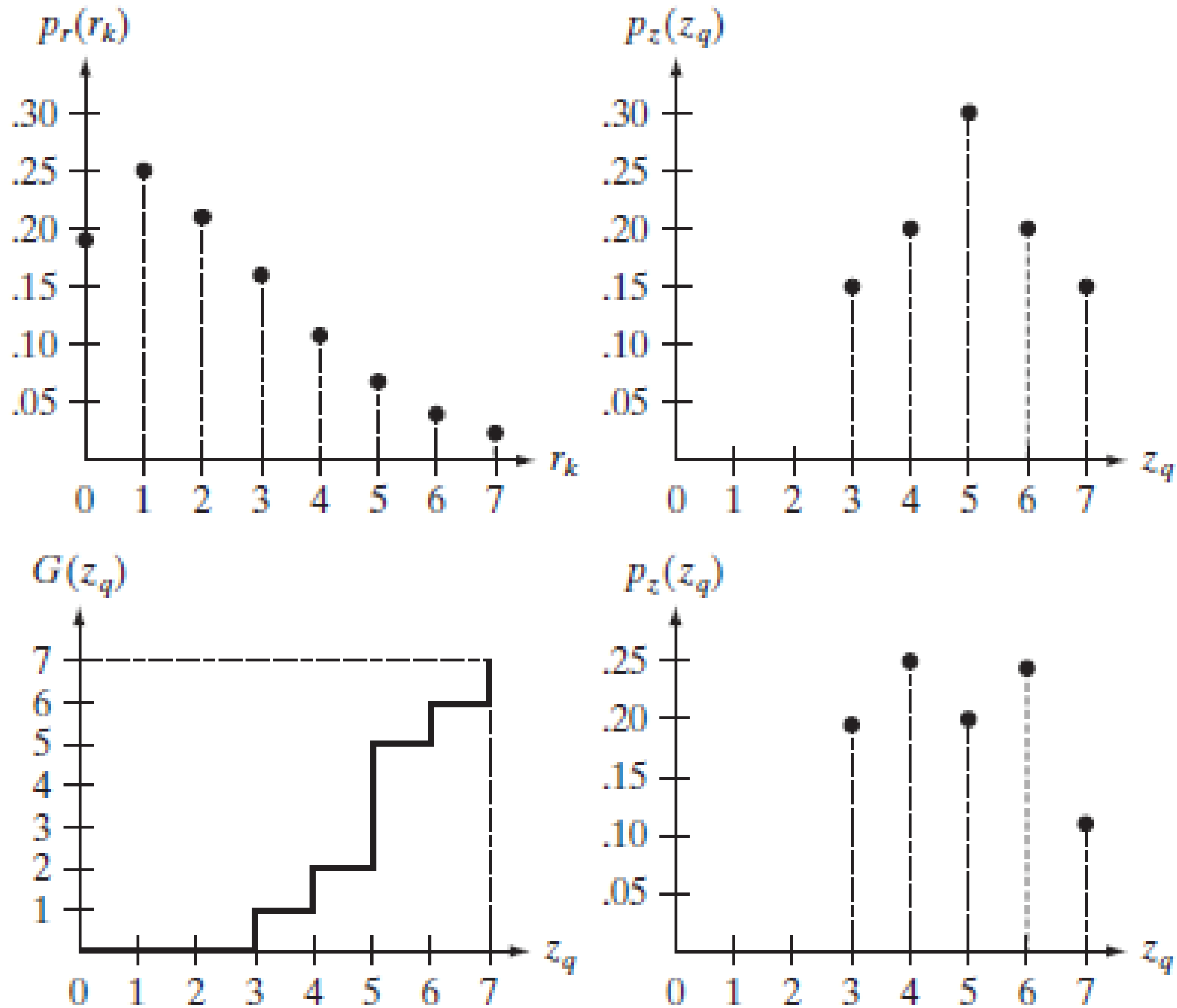
Histogram equalization on different starting images





Source: Digital Image Processing, Gonzalez and Woods, Prentice Hall

One can also specify a histogram



Source: Digital Image Processing, Gonzalez and Woods, Prentice Hall

Now we need two transformations

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

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

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

- Obtain pdf
- Compute *G(z)*
- Compute $G^{-1}(s)$
- First equalize, then apply $G^{-1}(s)$

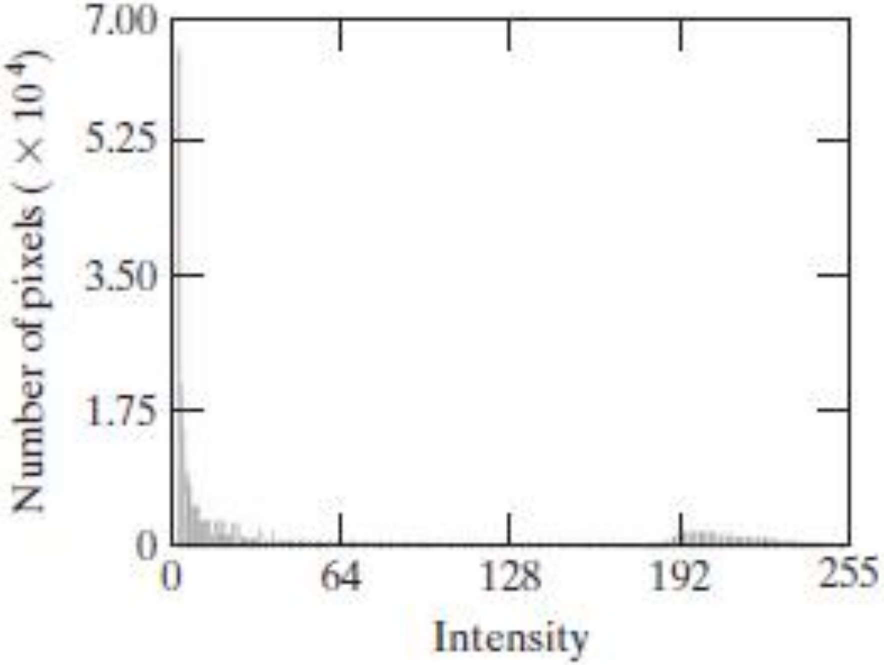
Source: Digital Image Processing, Gonzalez and Woods, Prentice Hall

For discrete values, it cannot be perfect

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

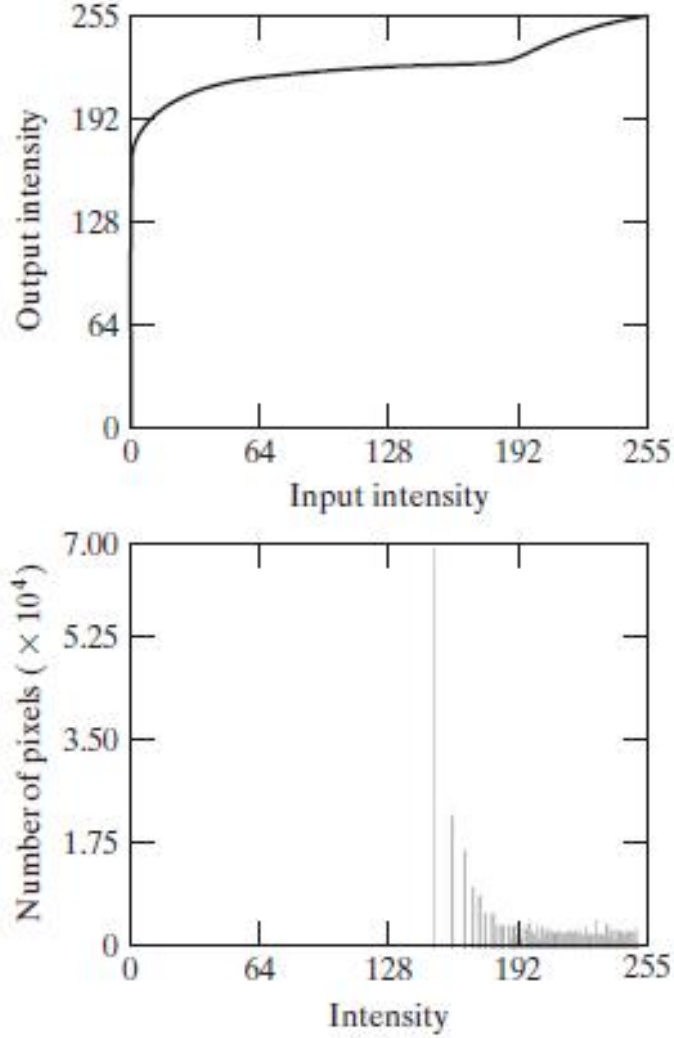
Source: Digital Image Processing, Gonzalez and Woods, Prentice Hall

Original image



Source: Digital Image Processing, Gonzalez and Woods, Prentice Hall

Equalized image



Source: Digital Image Processing, Gonzalez and Woods, Prentice Hall

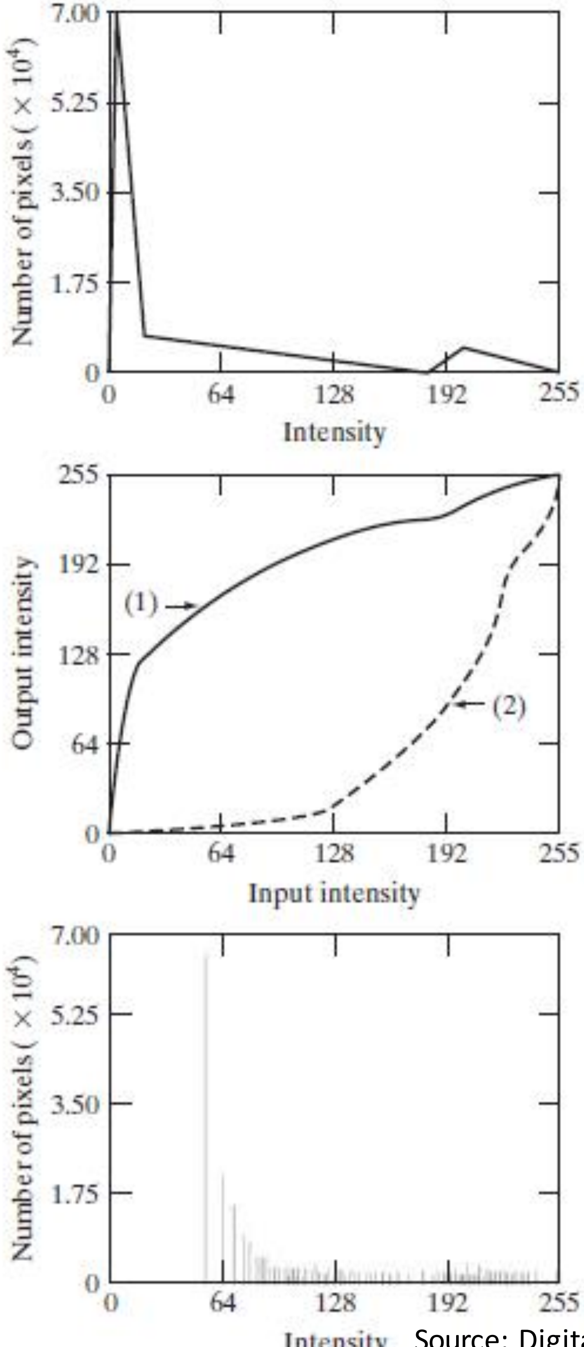
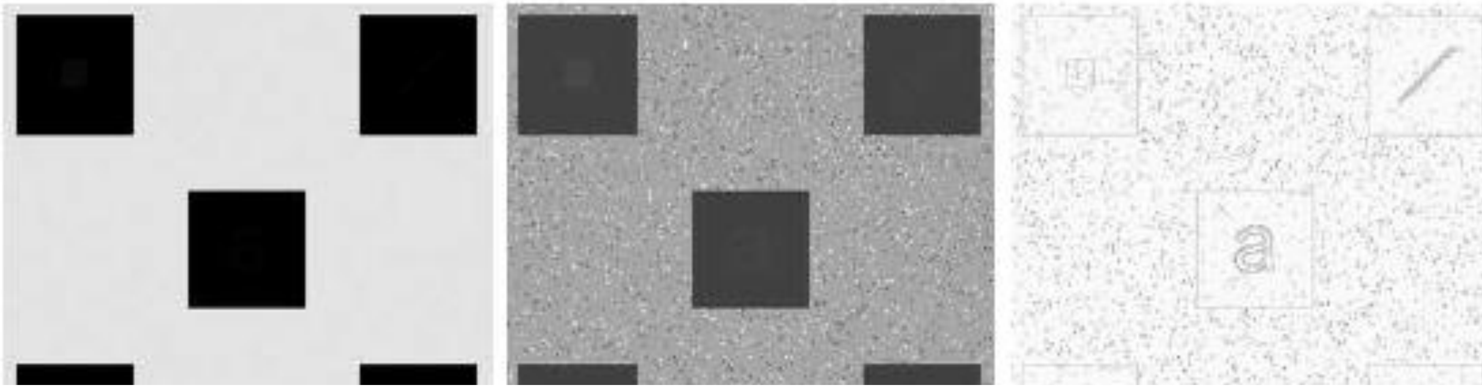
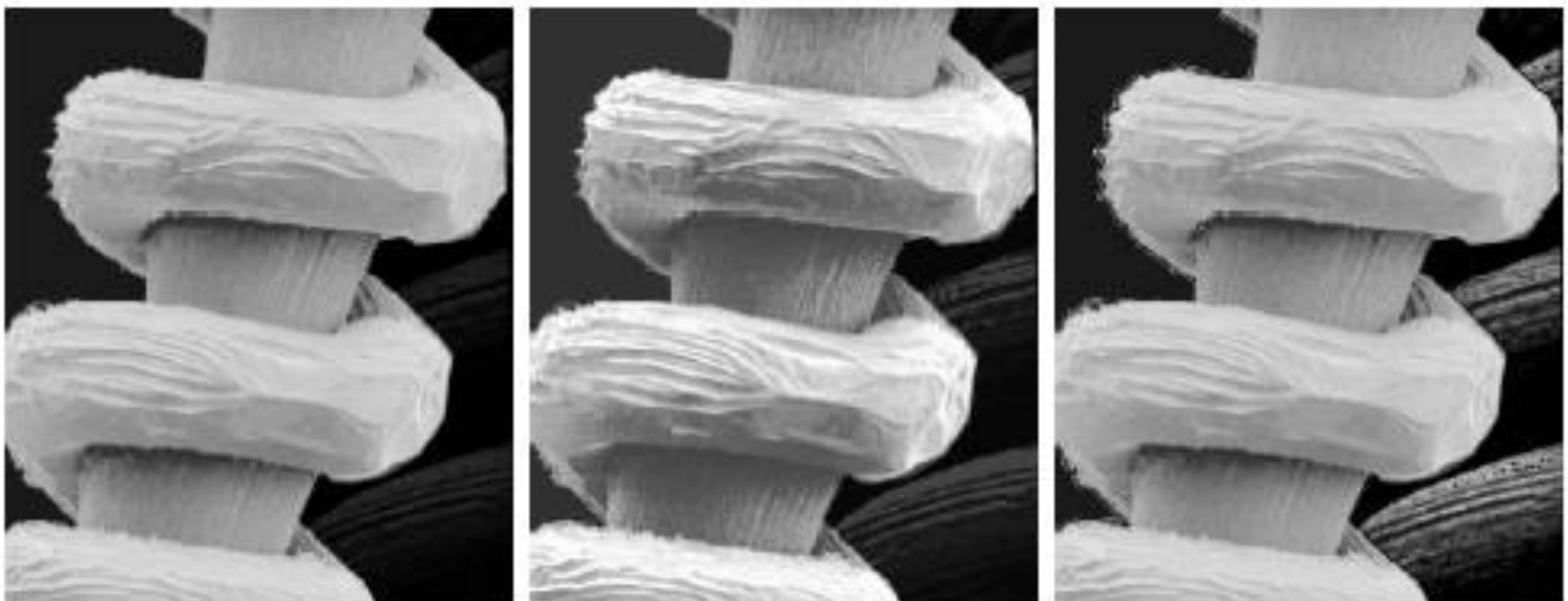


Image with histogram specification

Source: Digital Image Processing, Gonzalez and Woods, Prentice Hall

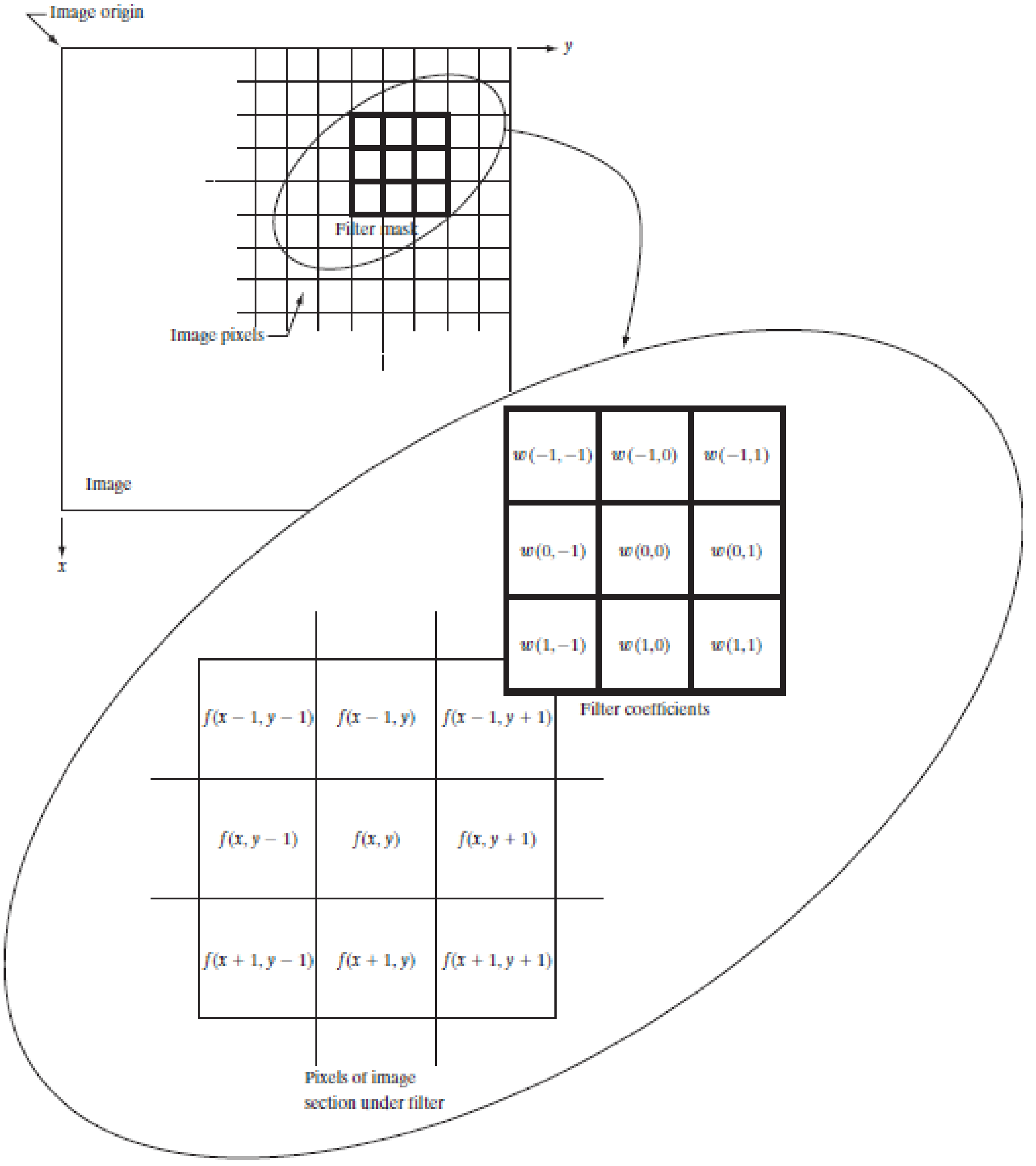
Global vs local histogram equalization





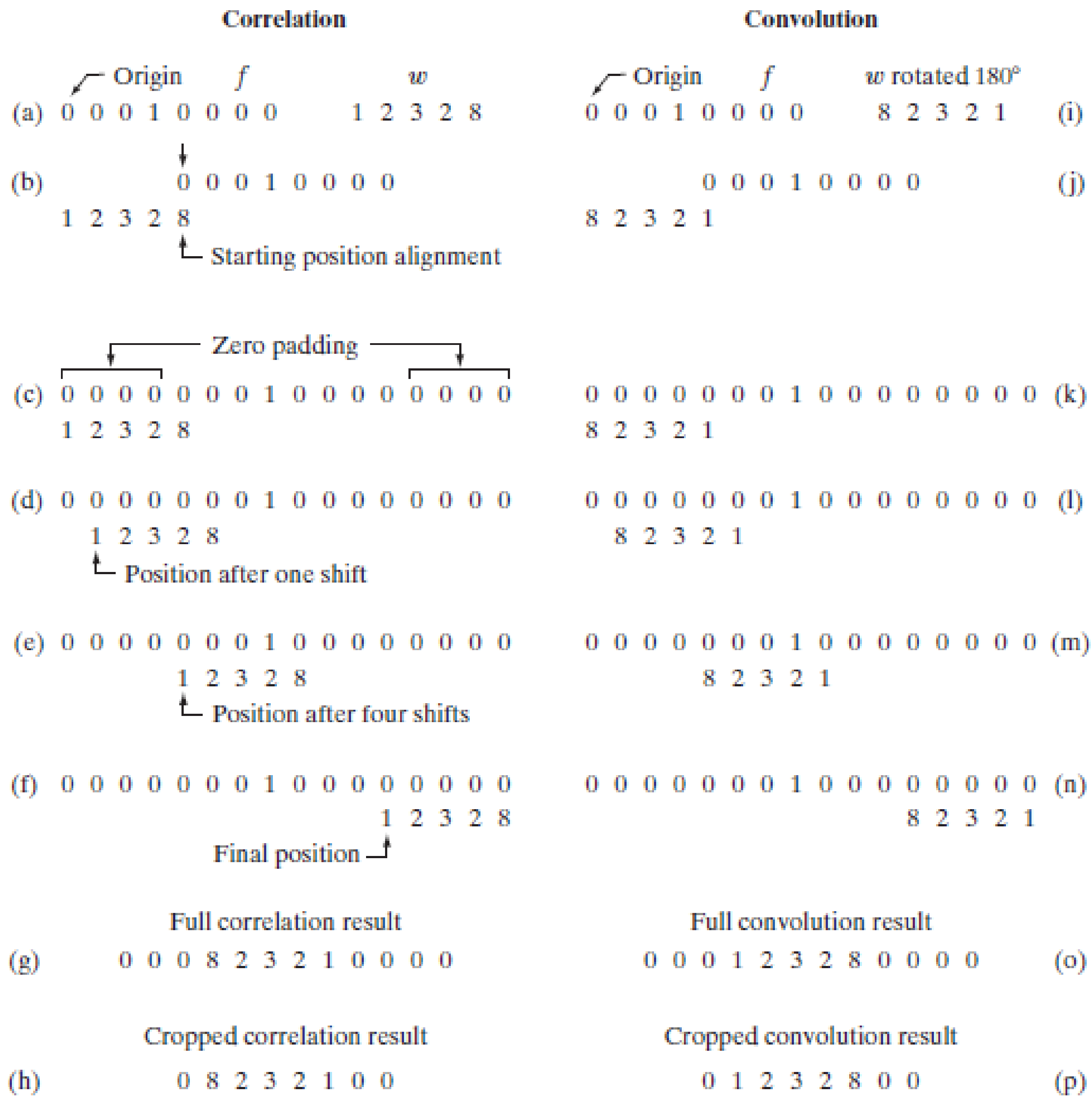
Source: Digital Image Processing, Gonzalez and Woods, Prentice Hall

Basics of spatial filtering



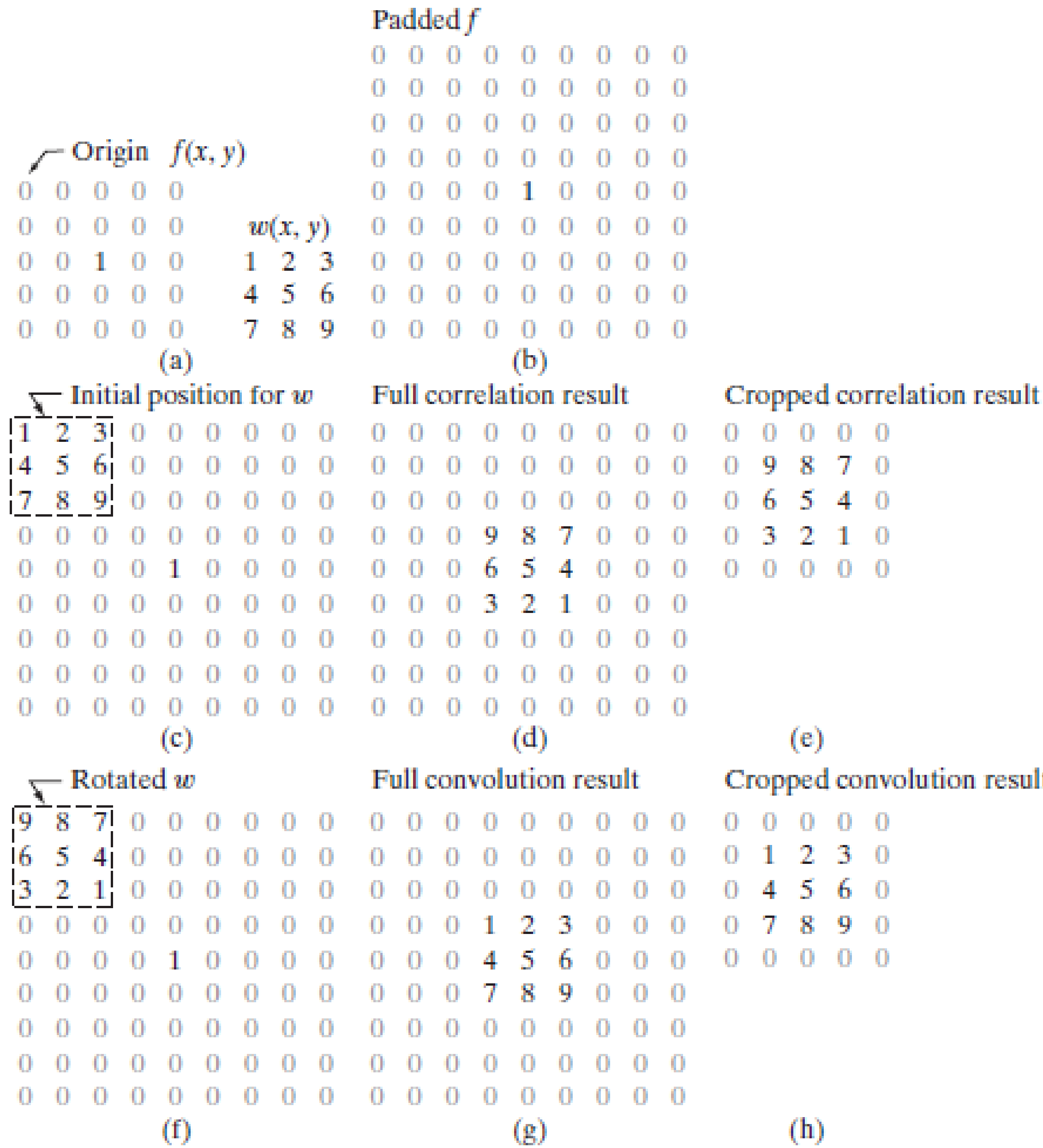
Source: Digital Image Processing, Gonzalez and Woods, Prentice Hall

Correlation vs convolution



Source: Digital Image Processing, Gonzalez and Woods, Prentice Hall

Correlation and convolution in 2-D



Source: Digital Image Processing, Gonzalez and Woods, Prentice Hall

Correlation vs. convolution in 2-D

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

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

Source: Digital Image Processing, Gonzalez and Woods, Prentice Hall

Correlation as a vector operation

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

$$
\begin{aligned}
R &= w_1 z_1 + w_2 z_2 + \ \dots \ + w_{mn} z_{mn} \\
&= \sum_{k=1}^{mn} w_k z_k \\
&= \mathbf{w}^T \mathbf{z}
\end{aligned}
$$

Source: Digital Image Processing, Gonzalez and Woods, Prentice Hall

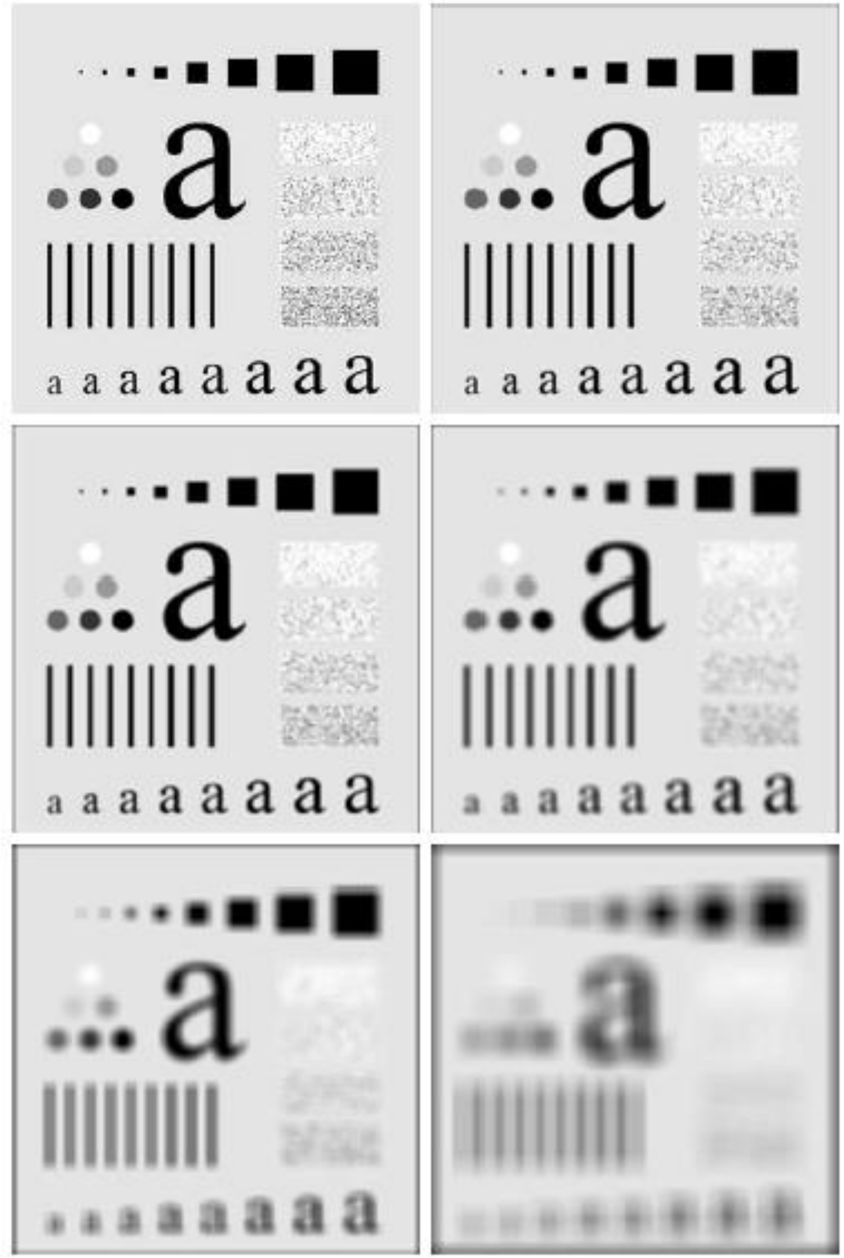
Smoothing (low pass filters)

$$\frac{1}{9} \times \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \qquad \frac{1}{16} \times \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

$$h(x, y) = e^{-\frac{x^2 + y^2}{2\sigma^2}}$$

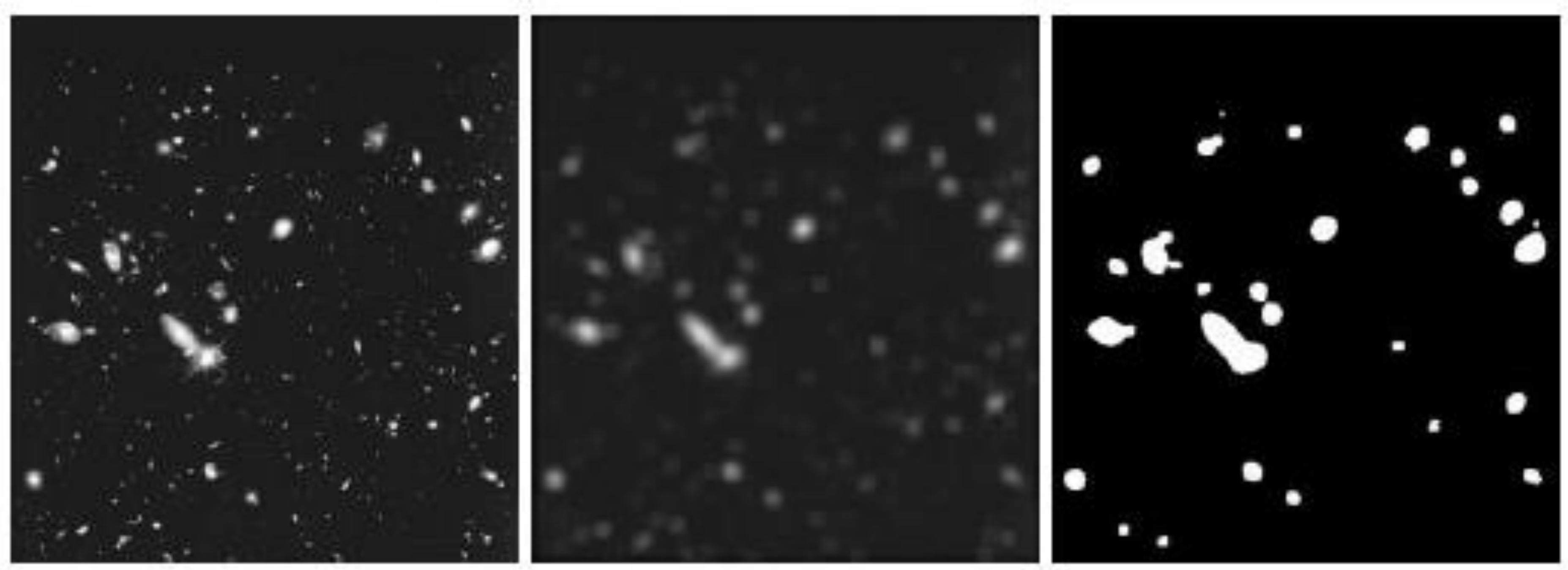
Source: Digital Image Processing, Gonzalez and Woods, Prentice Hall

Averaging with box filters of sizes 1, 3, 5, 9, 15, and 35



Source: Digital Image Processing, Gonzalez and Woods, Prentice Hall

Filtering followed by thresholding



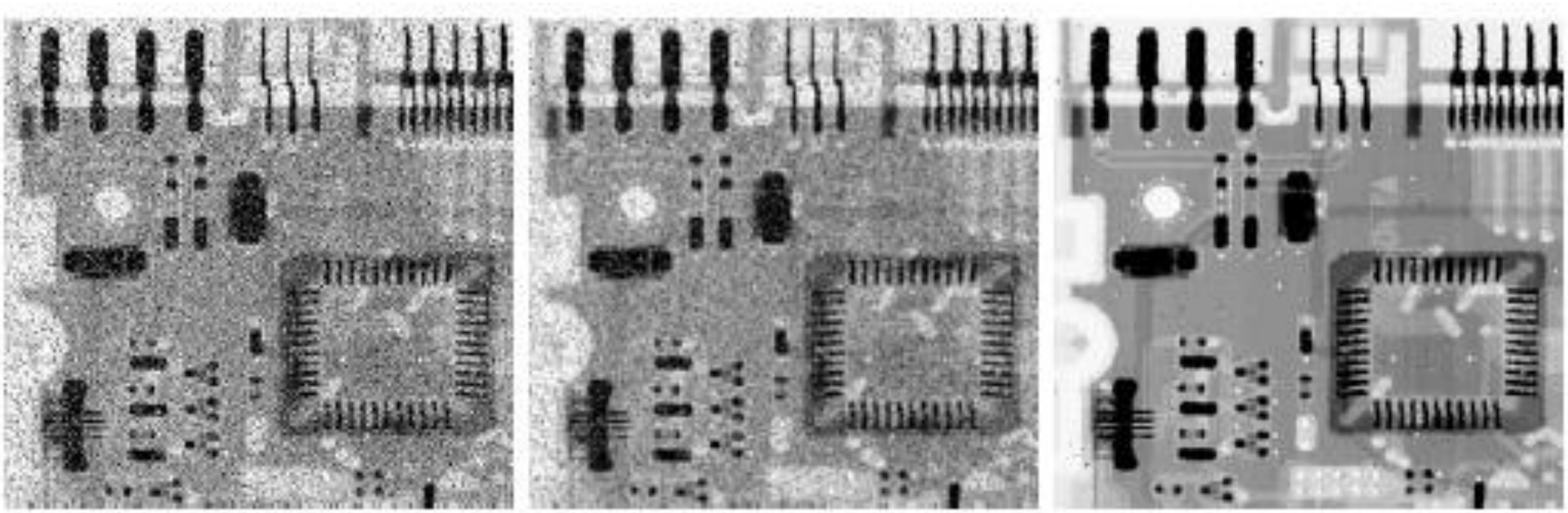
Source: Digital Image Processing, Gonzalez and Woods, Prentice Hall

Filtering as weighted average

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

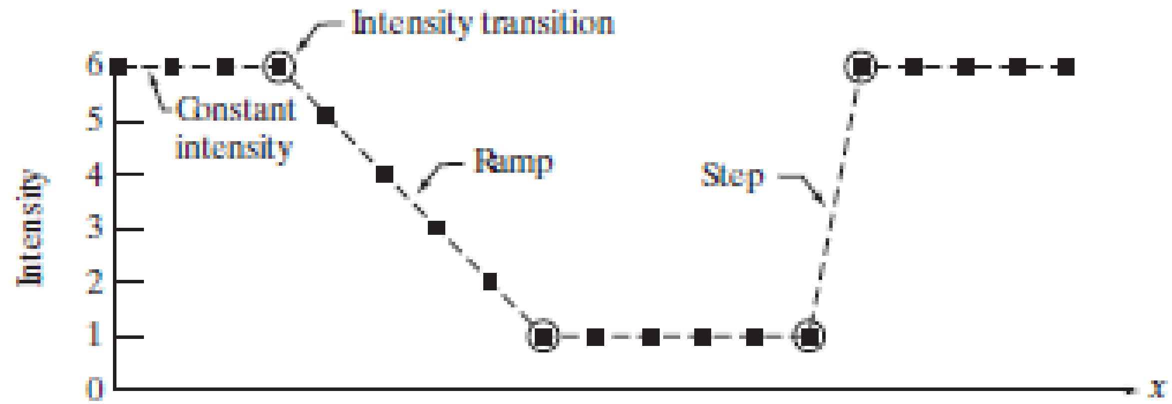
Source: Digital Image Processing, Gonzalez and Woods, Prentice Hall

Nonlinear filters such as median filter (right) are better suited many times



Source: Digital Image Processing, Gonzalez and Woods, Prentice Hall

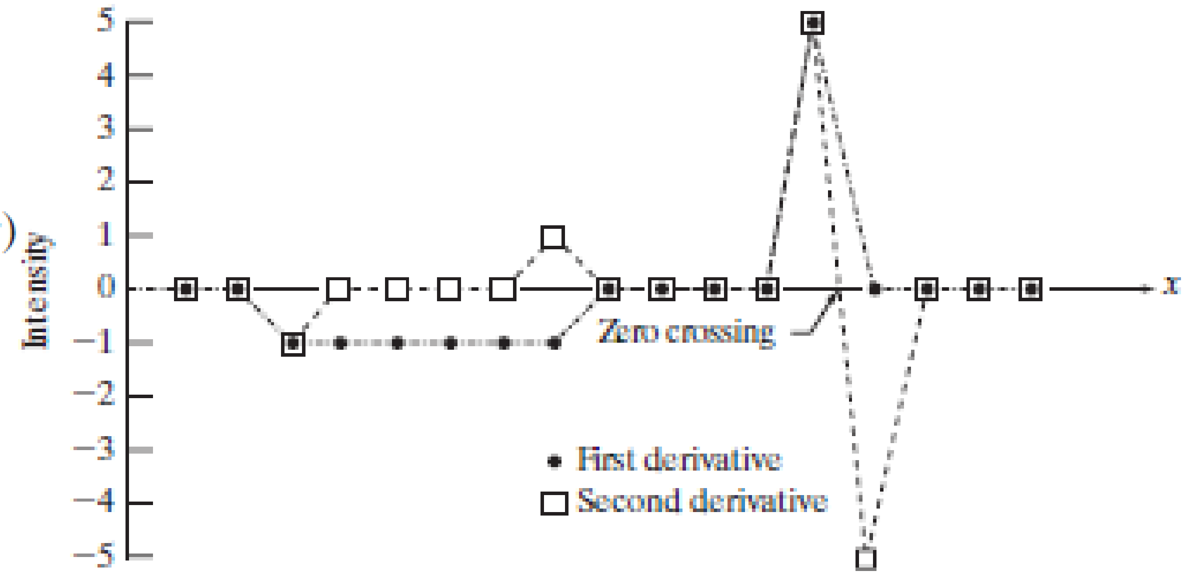
Derivatives of intensity profiles



Scan line	6	6	6	6	5	4	3	2	1	1	1	1	1	1	6	6	6	6	6
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	

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

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



Source: Digital Image Processing, Gonzalez and Woods, Prentice Hall

Sobel operators for directional edges

$$g_x = \frac{\partial f}{\partial x} = (z_7 + 2z_8 + z_9) - (z_1 + 2z_2 + z_3)$$

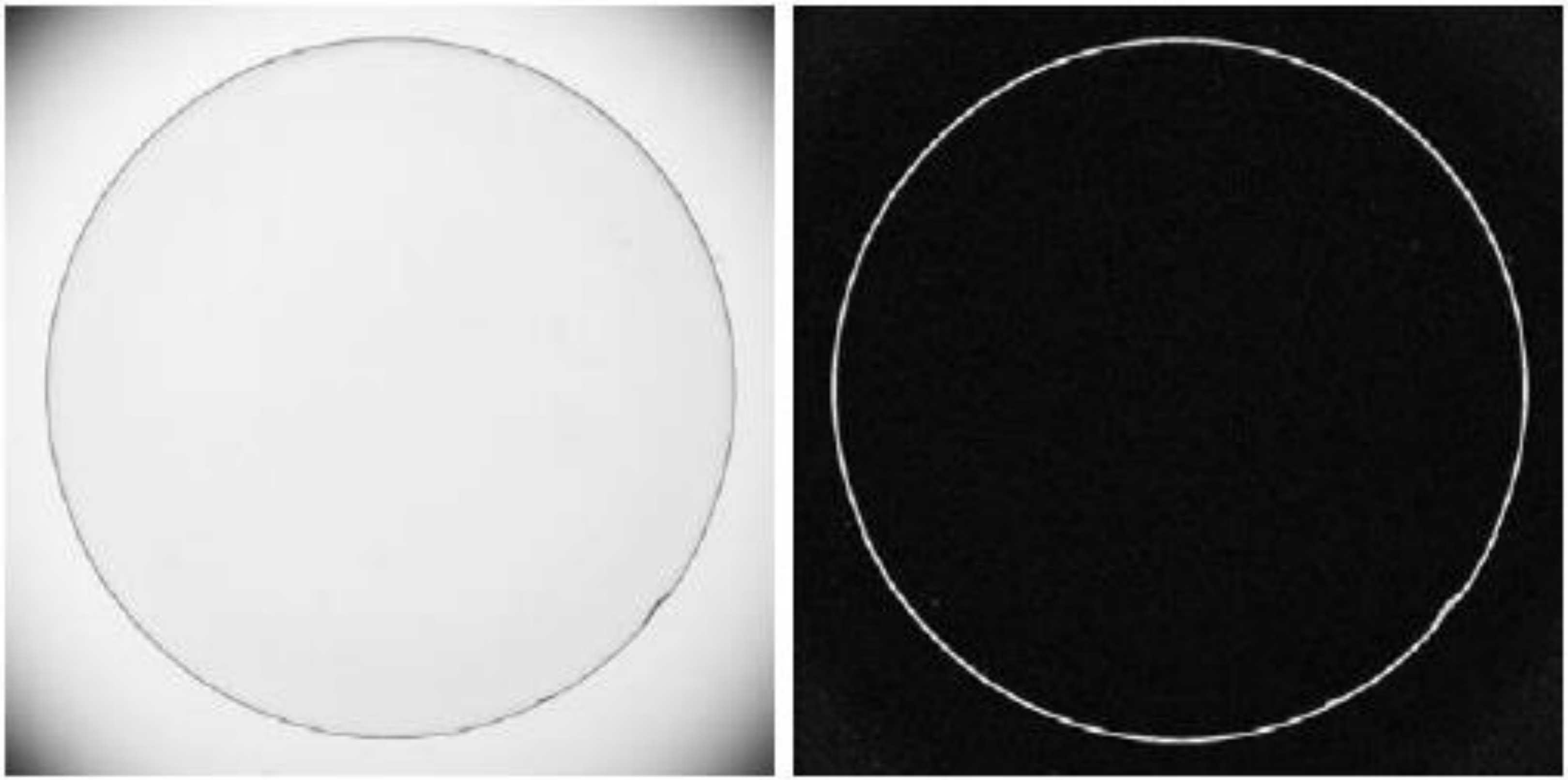
$$g_y = \frac{\partial f}{\partial y} = (z_3 + 2z_6 + z_9) - (z_1 + 2z_4 + z_7)$$

−1	−2	−1
0	0	0
1	2	1

−1	0	1
−2	0	2
−1	0	1

Source: Digital Image Processing, Gonzalez and Woods, Prentice Hall

Original and Sobel gradient



Source: Digital Image Processing, Gonzalez and Woods, Prentice Hall

Laplacian as an isotropic second derivative

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

$$\nabla^2 f(x, y) = f(x + 1, y) + f(x - 1, y) + f(x, y + 1) + f(x, y - 1) - 4f(x, y) \quad (3.6\text{-}6)$$

0	1	0
1	−4	1
0	1	0

1	1	1
1	−8	1
1	1	1

Source: Digital Image Processing, Gonzalez and Woods, Prentice Hall

Practical implementation of Laplacian

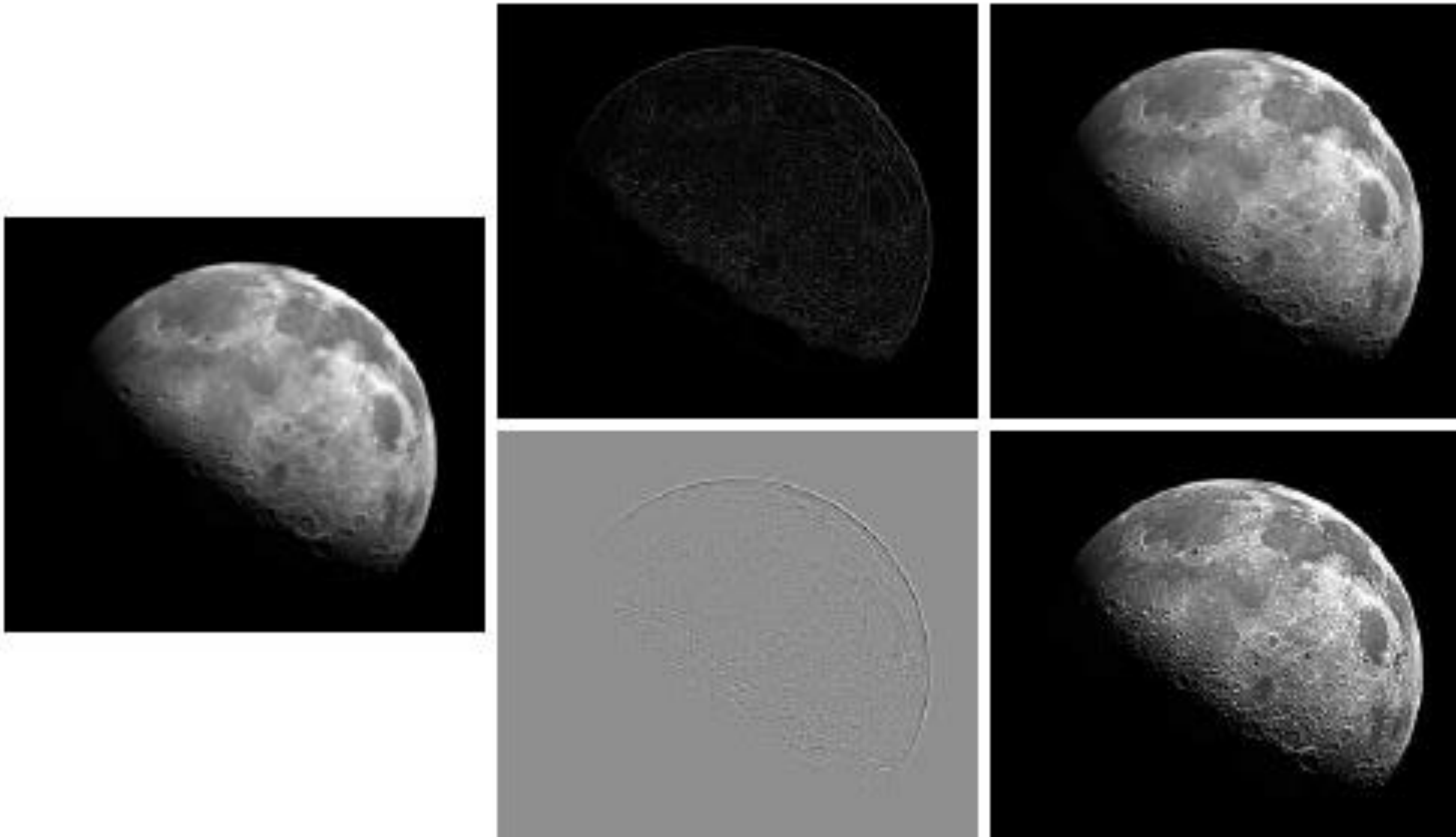
0	−1	0
−1	4	−1
0	−1	0

−1	−1	−1
−1	8	−1
−1	−1	−1

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

Source: Digital Image Processing, Gonzalez and Woods, Prentice Hall

Results of sharpening



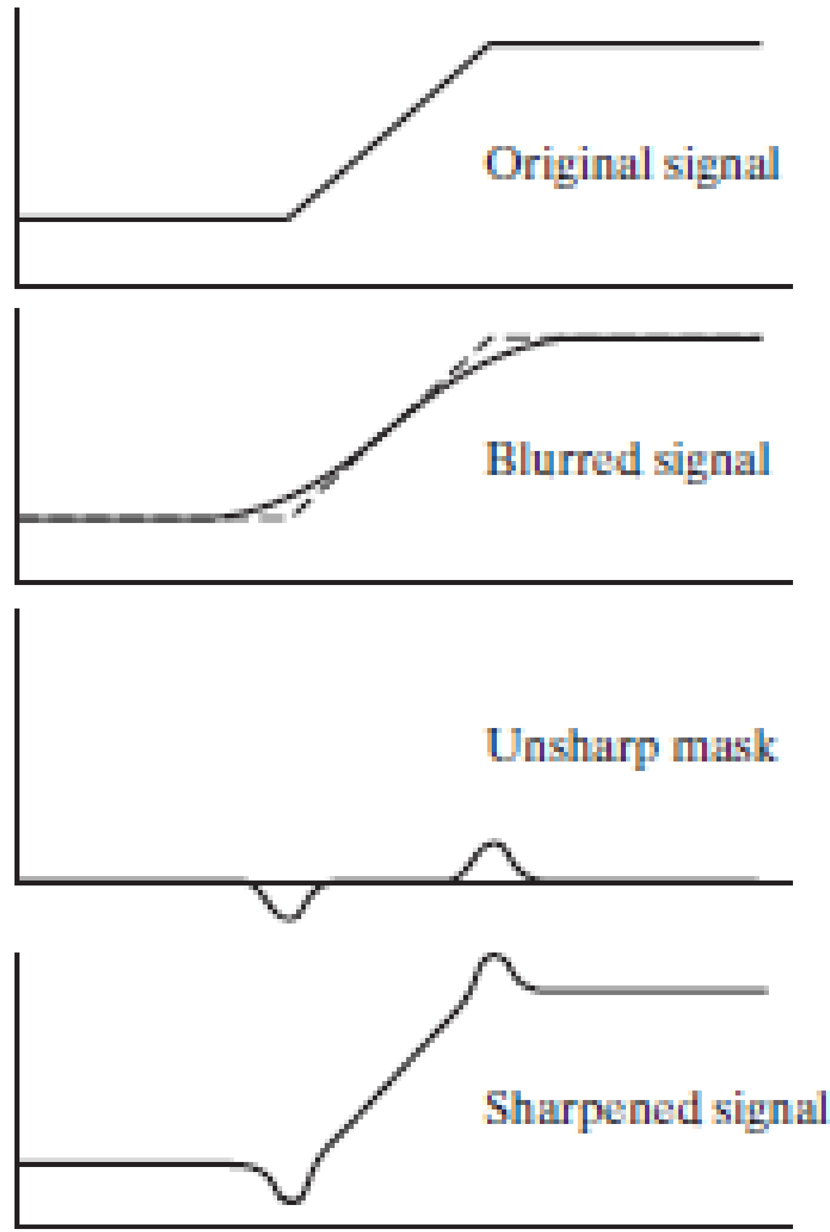
Source: Digital Image Processing, Gonzalez and Woods, Prentice Hall

Unsharp masking

- Blur the original image.
- *mask* = original – blurred
- *output* = original + mask

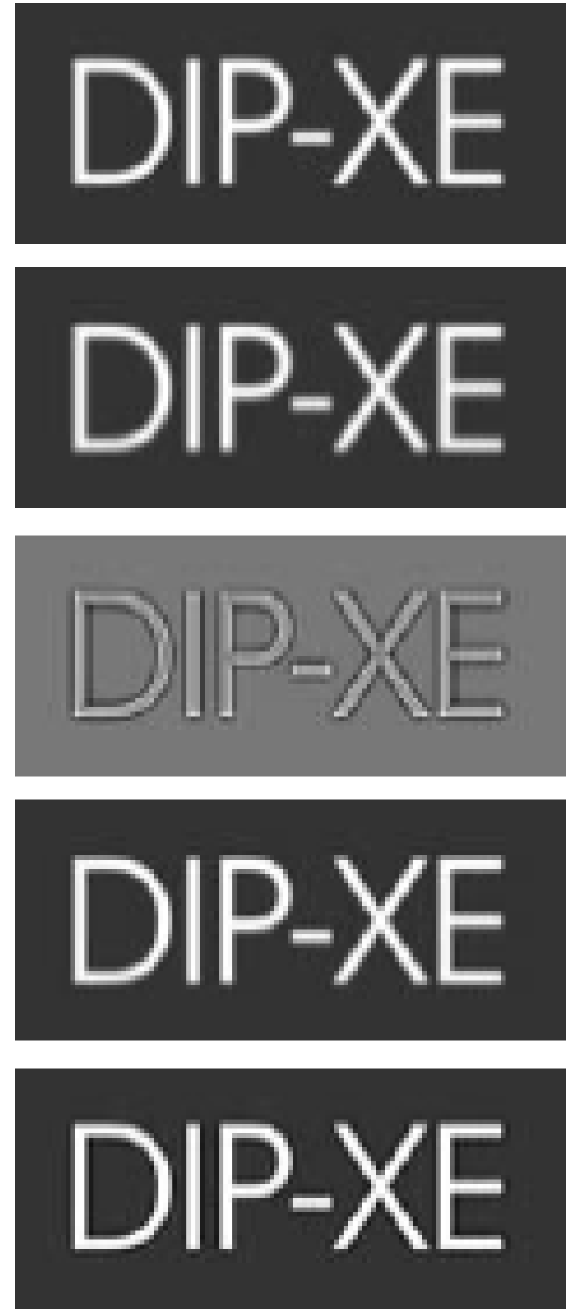
$$g_{\text{mask}}(x, y) = f(x, y) - \overline{f}(x, y)$$

$$g(x, y) = f(x, y) + k * g_{\text{mask}}(x, y)$$



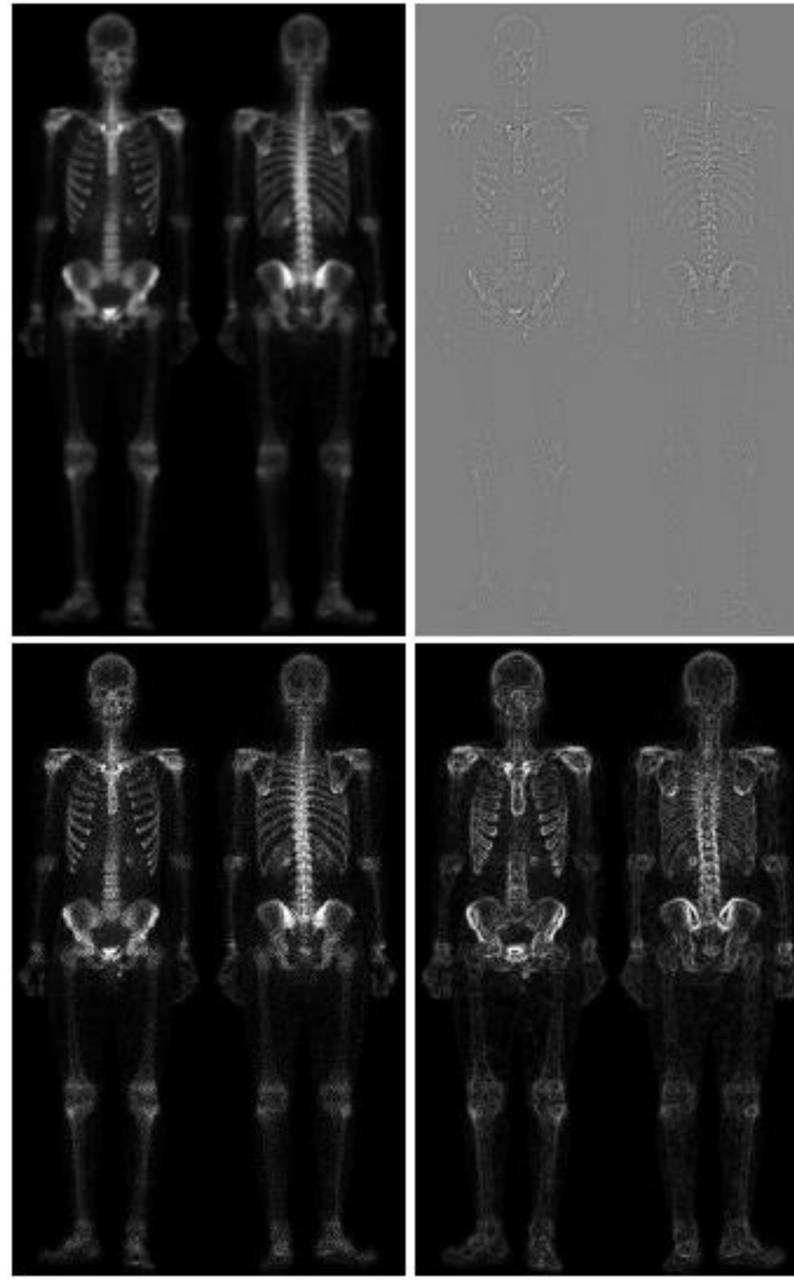
Source: Digital Image Processing, Gonzalez and Woods, Prentice Hall

Sample results of unsharp masking



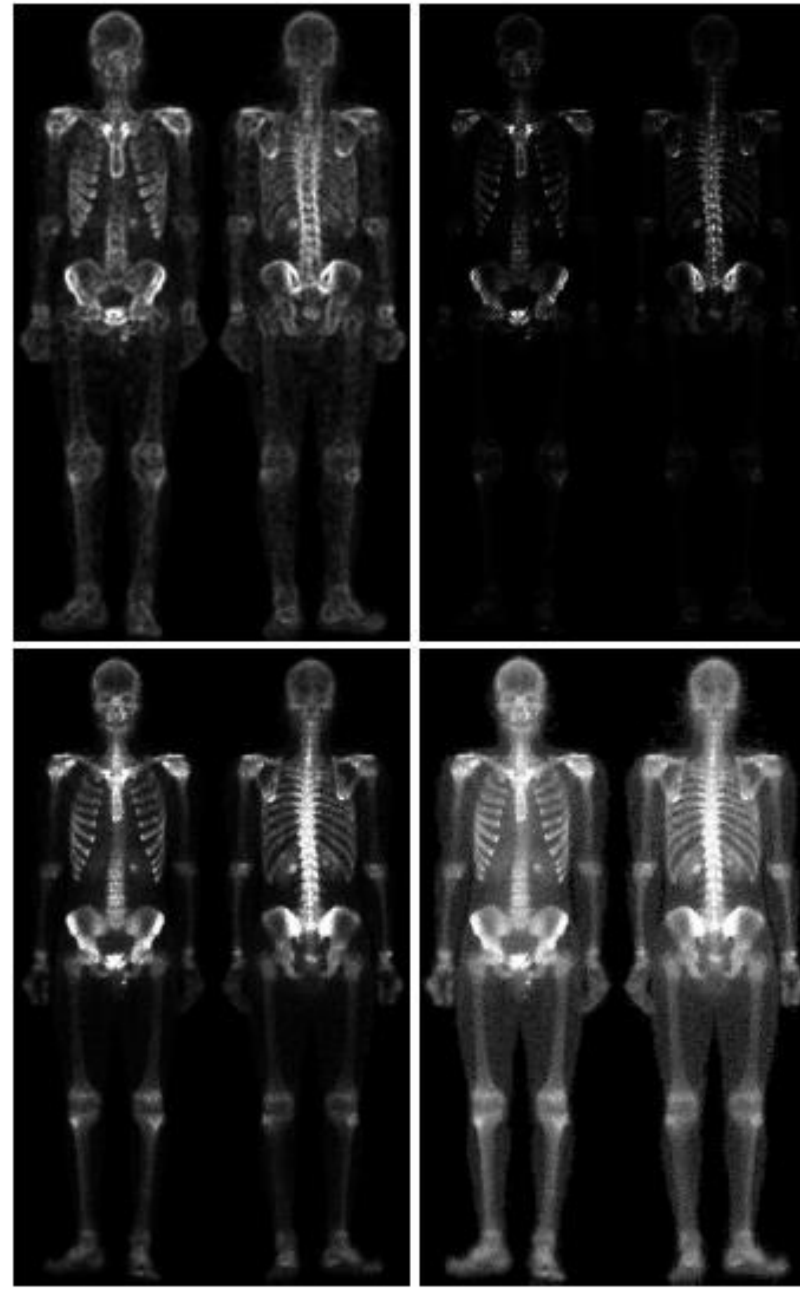
Source: Digital Image Processing, Gonzalez and Woods, Prentice Hall

Original, Laplacian, Sharpened, Sobel gradient



Source: Digital Image Processing, Gonzalez and Woods, Prentice Hall

Smoothened Sobel, c*e, a+f, power law transformation



Source: Digital Image Processing, Gonzalez and Woods, Prentice Hall