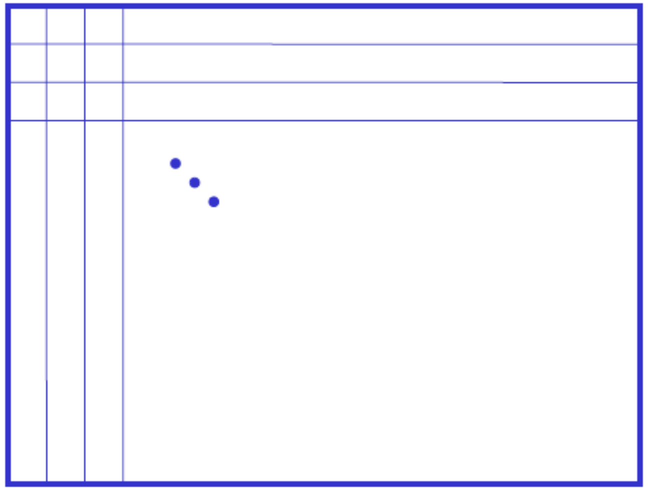
Spatial Processing

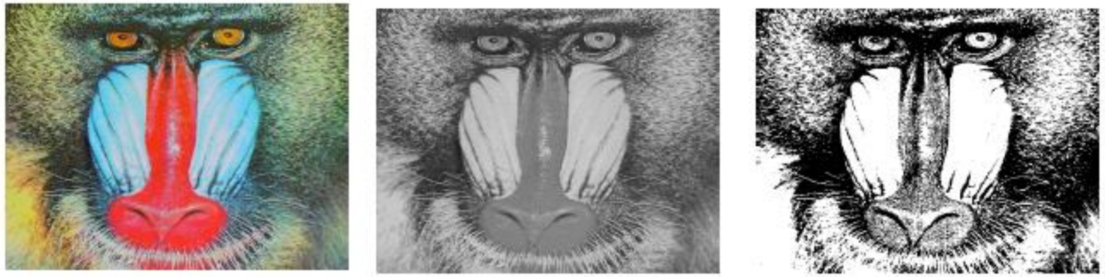
ENEE 6581
Digital Image Processing

Slide Credits: Digital Image Processing, Gonzalez & Woods

Images

- Spatial functions: (x,y) image plane.
  - x represents rows of function
  - y represents columns
  - Origin is (1,1); located in upper left corner
    - Rows increase downward
  - f(x,y) = pixel values.
- Pixel value related to image color:
  - BW: 1 bit
  - Gray level: 1 byte
  - Full color: 3 bytes





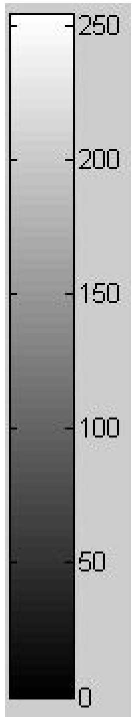
RGB Greyscale Binary

BW Images

- Pixel values = 0,1
- No color = black = 0
- Full color = white = 1
- Useful in processing, e.g.: edge detection

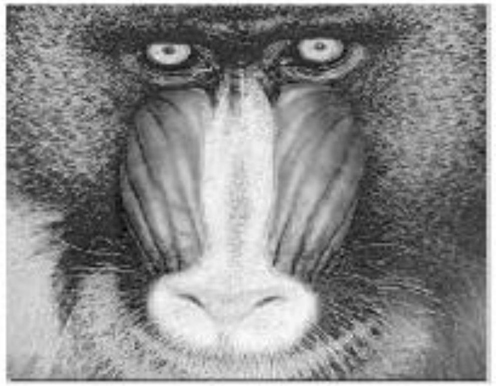
Graylevel

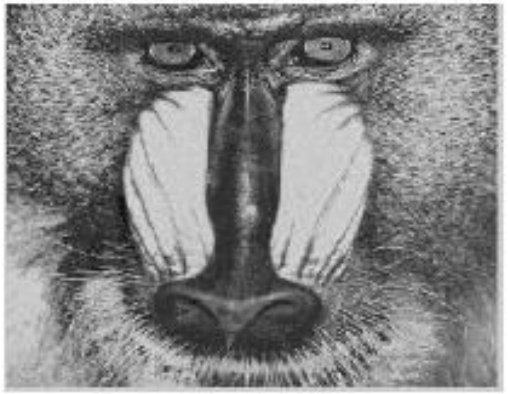
- Pixel values = 0…255
  - 8 bit (uint8)
- Shades of gray:
  - black = 0
  - white = 255

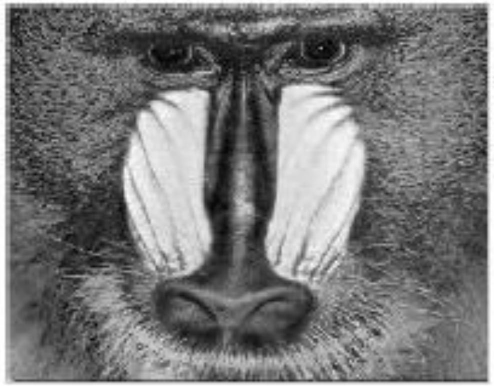


Full Color

- f(x,y,c)
- Color schemes vary. Schemes use 3 measurable quantities to determine color
- Basic color schemes: RGB
  - c = r,g,b
    - R = f(x,y,1), G = f(x,y,2), B = f(x,y,3)
  - Each color is a 256 (uint8) level matrix
    - 24bit color = ?? colors
    - r,g,b = 0…255, 0…255, 0…255







Distance Measures

- Given pixels *p*, *q*, and *z* at (x_1,y_1), (x_2,y_2) and (x_3,y_3)
- *D* is a *distance function* (or *metric*) if:
  - $D(p,q) \geq 0$; $(D(p,q)=0$ iff $p=q)$,
  - $D(p,q) = D(q,p)$,
  - $D(p,z) \leq D(p,q) + D(q,z)$.
- The *Euclidean distance* between *p* and *q* is given by:

$$D_e(p,q) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

  - The pixels having distance less than or equal to some value *r* from (x,y) are the points contained in a disk of radius *r* centered at (x,y)

D4 Distance

- The D_4 distance (also called the *city block distance*) between *p* and *q* is given by:

$$D_4(p,q) = |x_1 - x_2| + |y_1 - y_2|$$

  - The pixels having a D_4 distance less than some *r* from (x,y) form a diamond centered at (x,y)
  - Example: pixels where $D_4 \leq 2$

$$\begin{array}{ccccc} & & 2 & & \\ & 2 & 1 & 2 & \\ 2 & 1 & 0 & 1 & 2 \\ & 2 & 1 & 2 & \\ & & 2 & & \end{array}$$

D8 Distace

- The D_8 distance (also called the *chessboard distance*) between *p* and *q* is given by:

$$D_8(p,q) = \max\{|x_1 - x_2|, |y_1 - y_2|\}$$

  - The pixels having a D_8 distance less than some *r* from (x,y) form a square centered at (x,y)
  - Example: pixels where $D_8 \leq 2$

$$\begin{matrix} 2 & 2 & 2 & 2 & 2 \\ 2 & 1 & 1 & 1 & 2 \\ 2 & 1 & 0 & 1 & 2 \\ 2 & 1 & 1 & 1 & 2 \\ 2 & 2 & 2 & 2 & 2 \end{matrix}$$

Image Enhancement

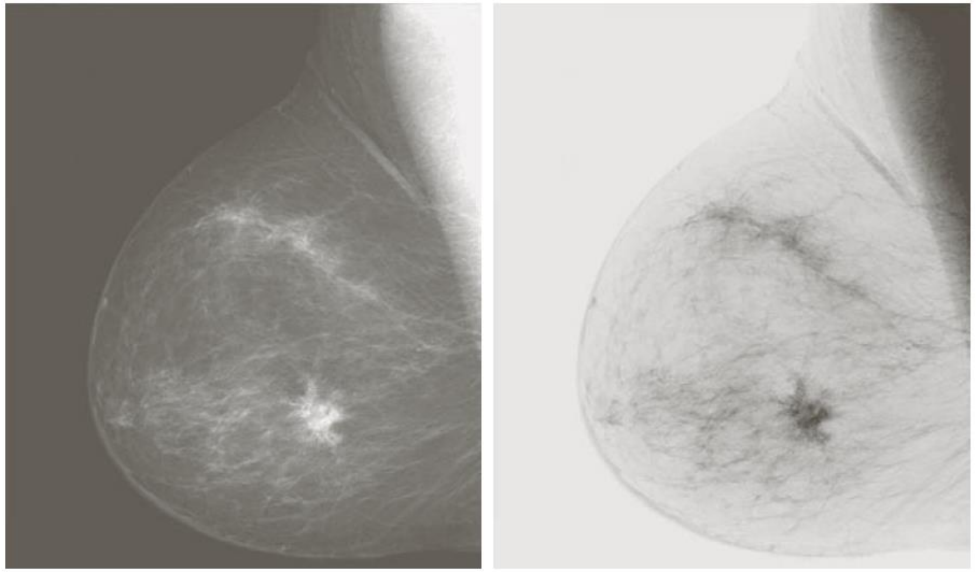
- Purposes
  - Process an image so that the result will be more suitable for a specific application
- Applications
  - Human Visual: subjective, no standard
  - Machine Perception: objective, better recognition results
- Domains
  - Spatial: manipulation of pixel values in an image
  - Spectral: modifying of Fourier transform of an image

Spatial Enhancement

- Pixel Processing
  - Linear (Piece-wise) transformation, lookup table
  - Nonlinear transform: Logarithm and Power-law
  - Gamma Correction
- Histogram Processing
  - Histogram normalization, Equalization
  - Histogram Matching (Specification)
- Area Processing
  - Linear Filters: smoothing and sharpening
  - Nonlinear Filters: max, min, median
- Thresholding
  - Binary map

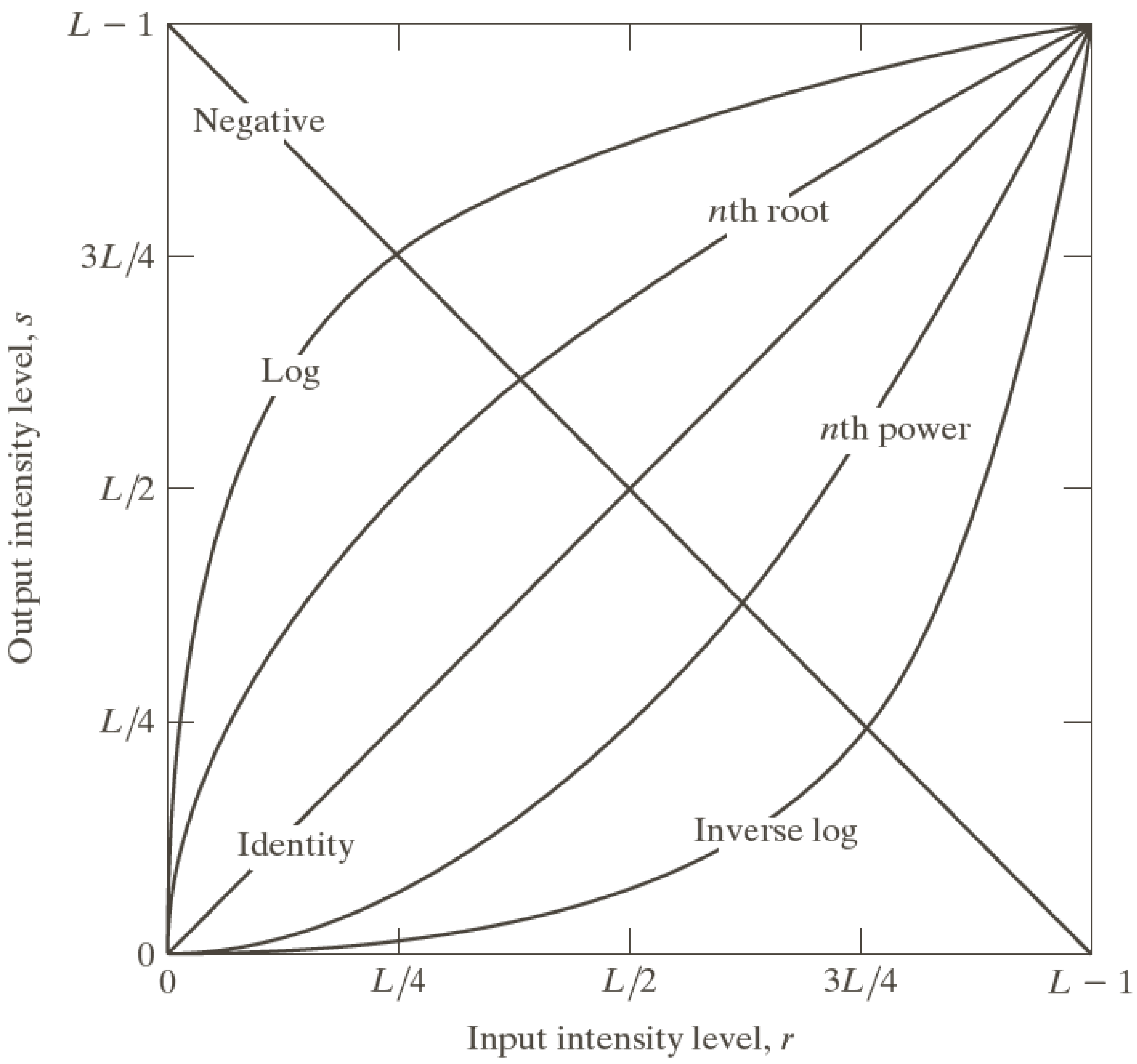
Pixel Processing: Transformation

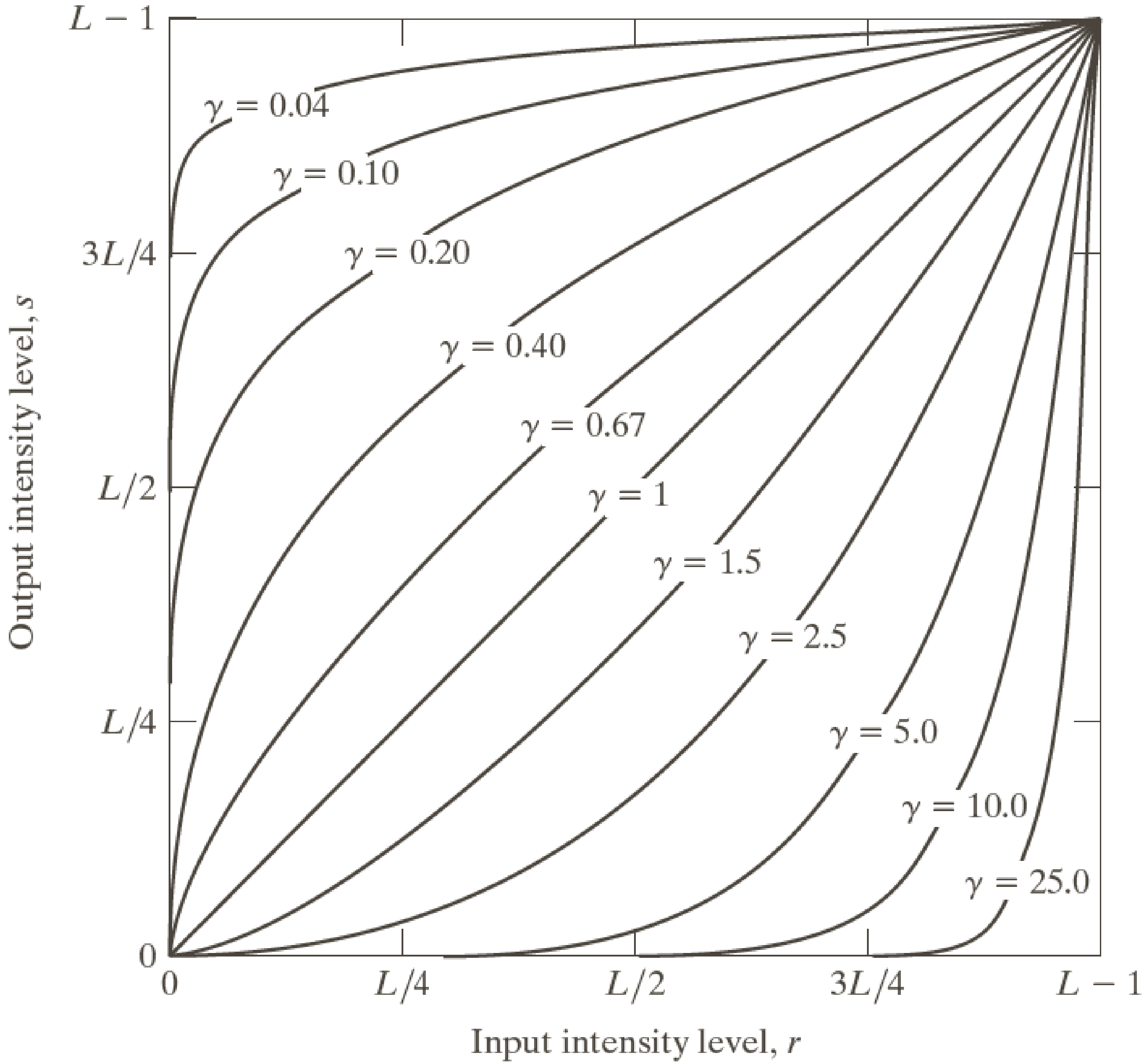
- Linear function
  - Identity: out = in
  - Negative: out = (L–1) – in
    - L = levels
    - in is distributed between 0 and L-1
    - Usefull for enhancing white/gray detail embedded in dark regions, e.g. mamograms
    - Human perception



Pixel Processing: Transformation

- Logarithm function
  - Log: out = c log (1+in);
    - c is a constant. Why is 1 added?
    - Linearizes power => enhance dark, reduce light
    - MATLAB: `out = round(log10(1+in))./log10(255);`
  - inverse-log: out = c 10^{in}
    - Not too common. Used to inverse effects of log.
- Power-law function
  - nth power: out = c in^{γ}
    - Compresses high values. Suitable for bright images.
    - MATLAB: `out = 255*(in/255).^power;`
  - nth root: out = c $in^{1/\gamma}$
    - Expands low values. Suitable for dark images

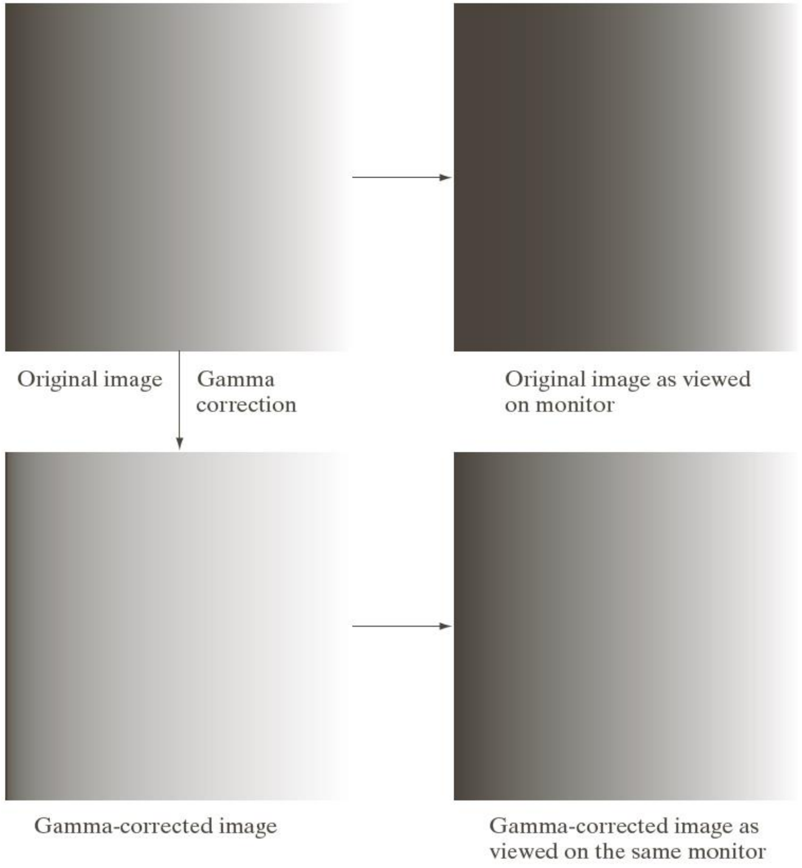

L − 1
3L/4
L/2
L/4
0
Output intensity level, s
Negative
Log
nth root
Identity
nth power
Inverse log
0
L/4
L/2
3L/4
L − 1
Input intensity level, r



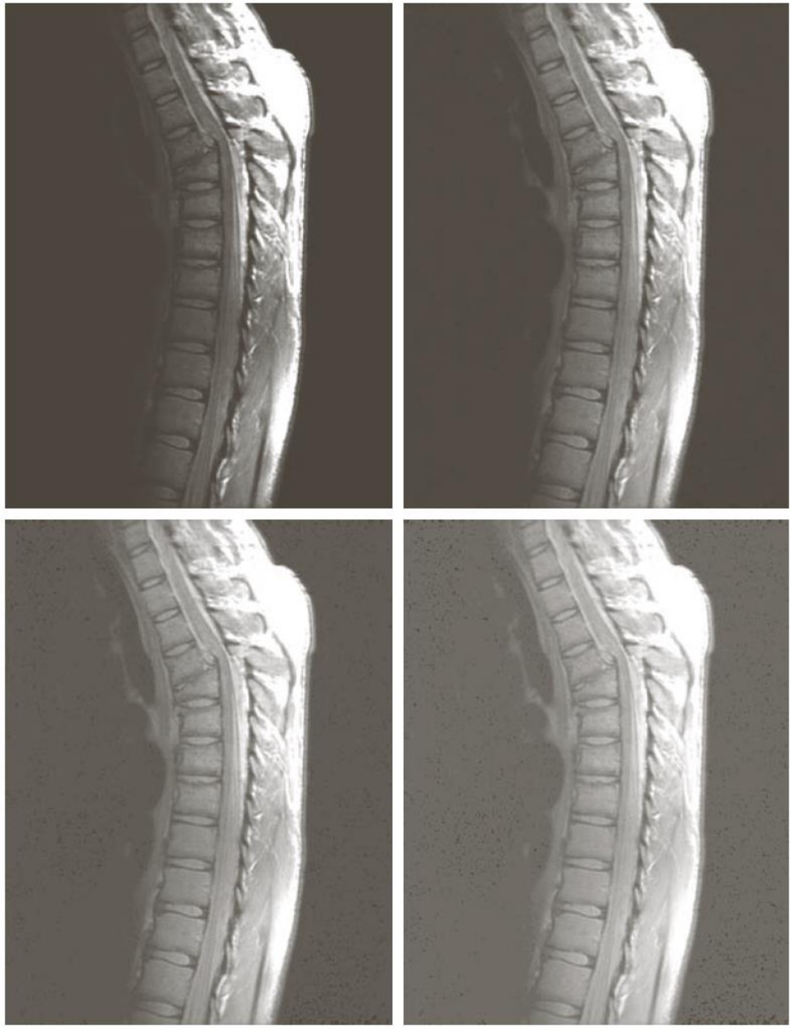
L − 1
3L/4
L/2
L/4
0
γ = 0.04
γ = 0.10
γ = 0.20
γ = 0.40
γ = 0.67
γ = 1
γ = 1.5
γ = 2.5
γ = 5.0
γ = 10.0
γ = 25.0
0
L/4
L/2
3L/4
L − 1
Input intensity level, r
Output intensity level, s

Examples: CRT Gamma Correction

- Cathode ray tube (CRT), DLP display devices have an intensity-to-voltage response that is a power function, with γ varying from 1.8 to 2.5
  - $\alpha\ r^{\gamma}$
- The picture will become darker.
- Gamma correction is done by preprocessing the image before inputting it to the display with $s = c\ r^{1/\gamma}$



Examples: power transformation



Histogram Processing

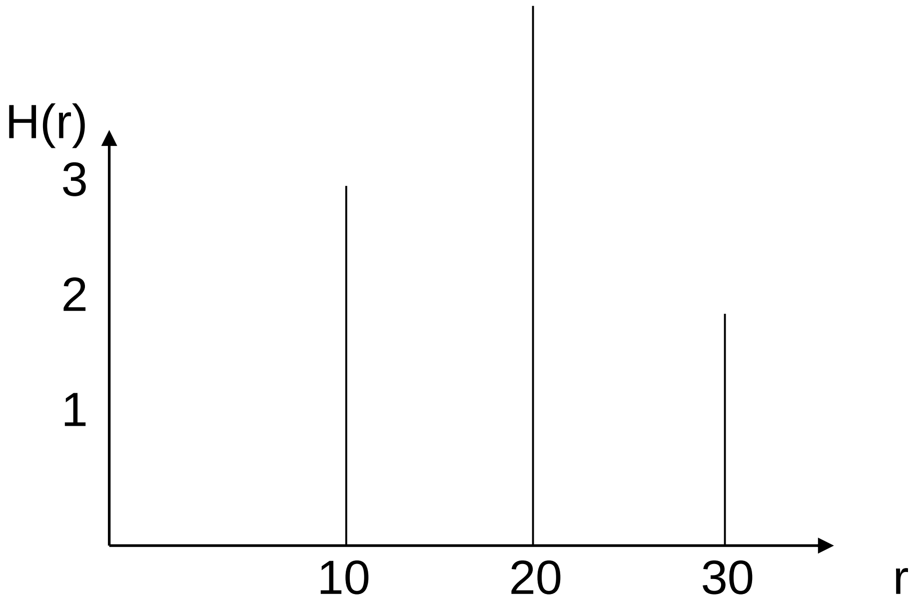
- Used effectively for image enhancement
  - useful in image compression and segmentation
- Histogram: distribution of pixels values
  - E.g. f = [20 10 20; 30 10 20; 30 20 10].

    Values = r = 10, 20, 30

    Occurrence = 3, 4, 2
  - Probability distribution:

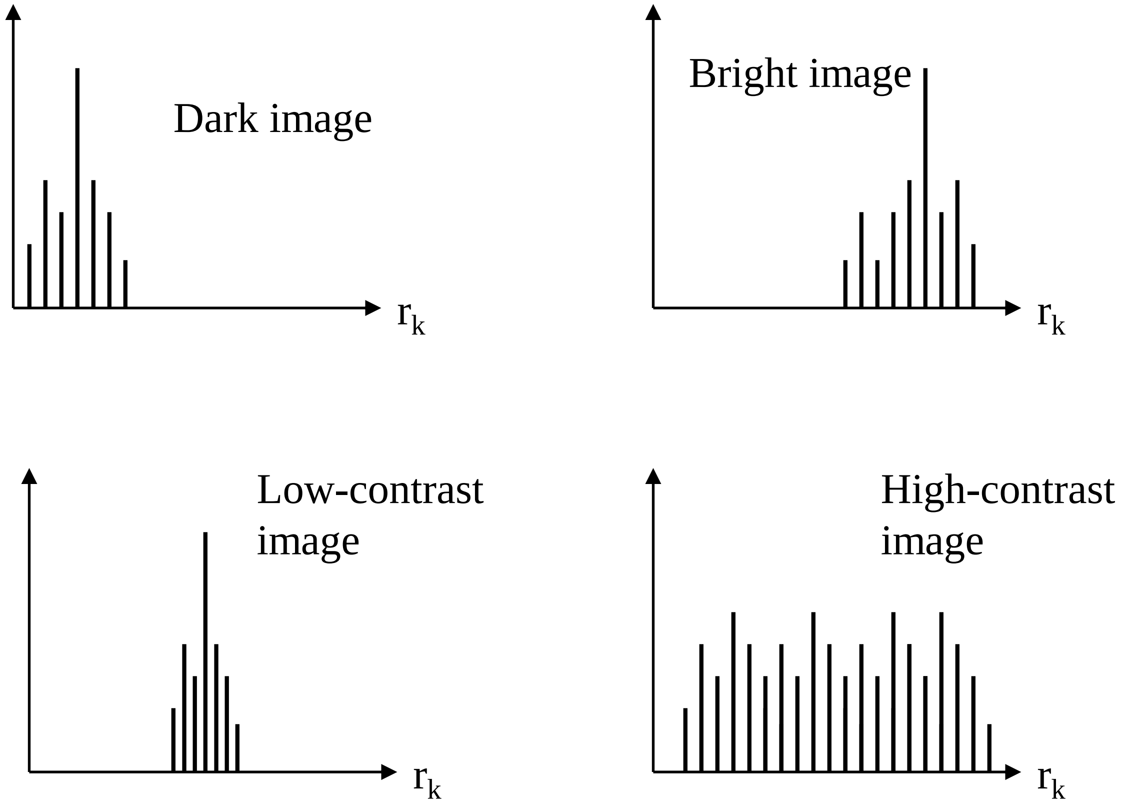
    $P_r(r) = H(r)/MN$

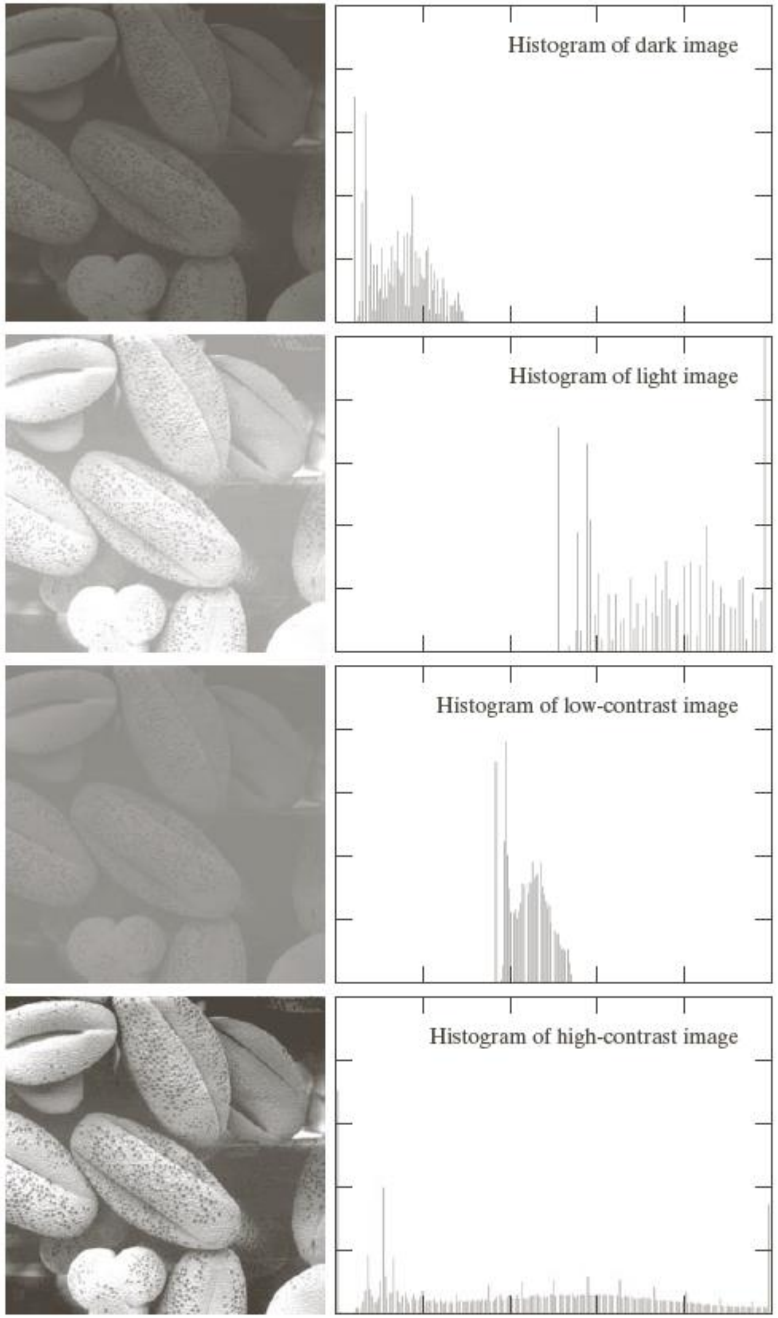
    MN = number of pixels



Histogram Processing

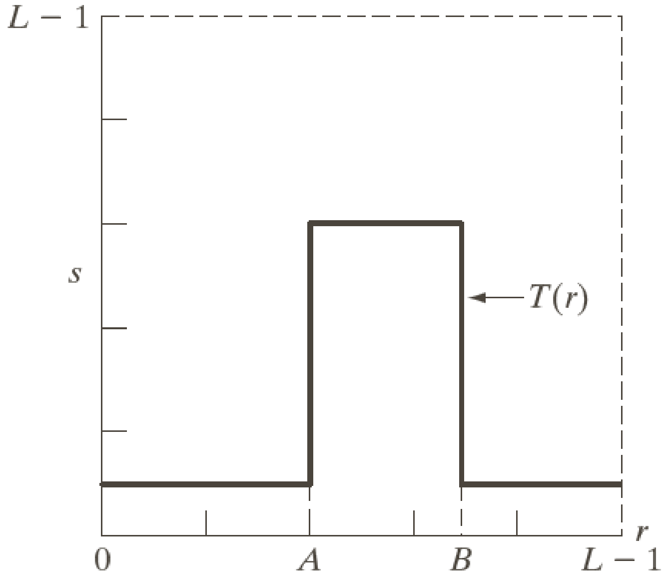
- Dark images:
  - Components of histogram are concentrated on the low side of the gray scale.
- Bright images:
  - Components of histogram are concentrated on the
  - high side of the gray scale.
- Low contrast
  - histogram is narrow and centered toward the middle of the gray scale
- High-contrast image
  - histogram covers broad range of the gray scale and the distribution of pixels is not too far from uniform, with very few vertical lines being much higher than the others

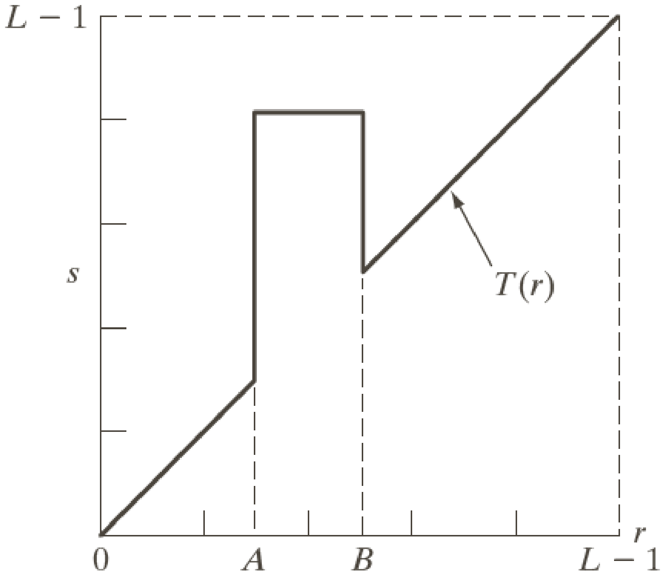

Dark image
r_k
Bright image
r_k
Low-contrast image
r_k
High-contrast image
r_k

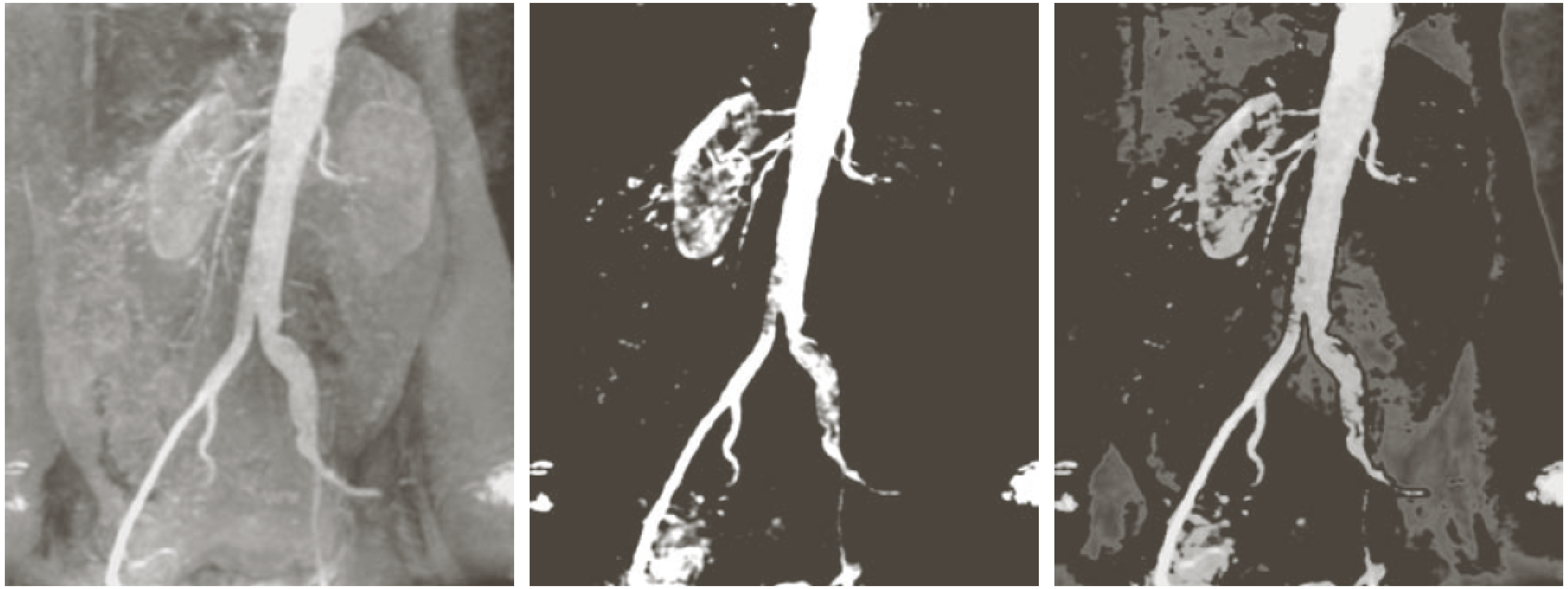

Histogram of dark image
Histogram of light image
Histogram of low-contrast image
Histogram of high-contrast image

Intensity Level Slicing: Threshold

- Highlight a a specific range of intensities
- Certain intensities belong to objects of interest
  - E.g. water bodies in satellite images or bones in a x-ray image, depth information in range data
- Two approaches:
  - Binary slicing.
  - Non-binary slices







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.)

Area Processing

- Purposes
  - Smoothing: spatial integration, reduce discontinuities, remove noise
  - Sharpening: spatial differential, enhance discontinuities, such as edges
- Linear Spatial Filtering
  - FIR/IIR filters
  - Average (integration), 1st 2nd Derivative, unsharp, high-boosting
- Nonlinear Spatial Filtering
  - Max, min, median
  - Self-defined function

Mask Operations

- Formulated in the context of mask operations (aka template, kernel, window or filter operations)
- Basic concept: let the value of a pixel be a function of its (current) gray level and the gray level of its neighbors (in some sense)
- Average operation:
  - Replace the value of a pixel with the average value of the pixels in a mxm region centered around it.

	Z_1	Z_2	Z_3	
	Z_4	Z_5	Z_6	
	Z_7	Z_8	Z_9	

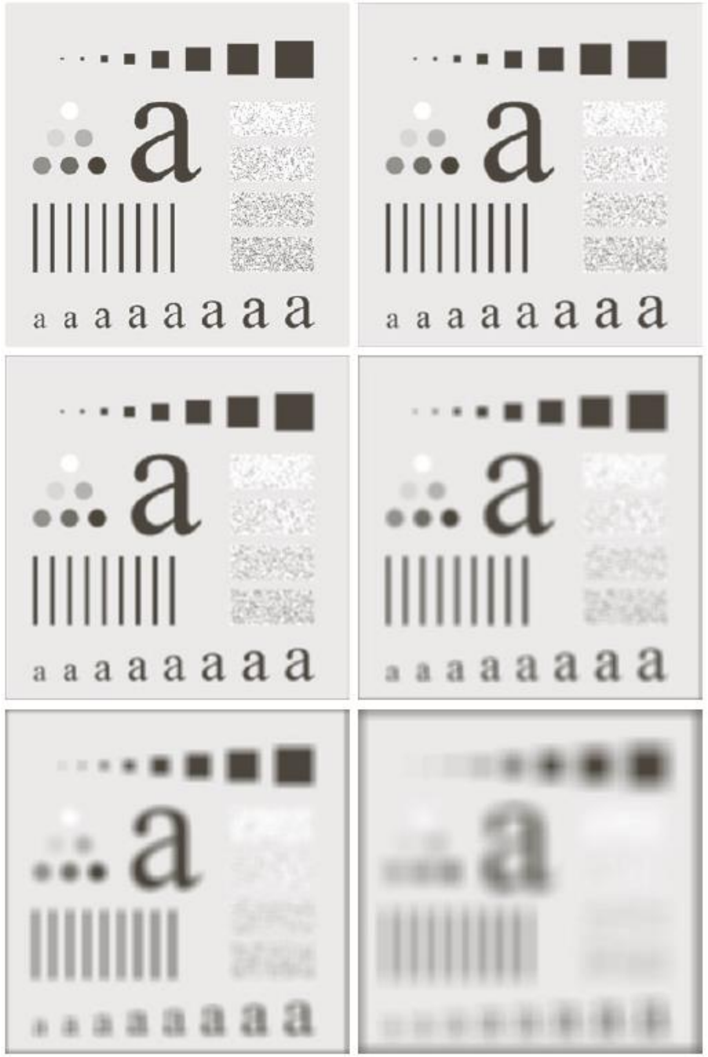
$$z = \frac{1}{m^2}(z_1 + z_2 + \ldots + z_{m\times m}) = \frac{1}{m^2}\sum_{i=1}^{m^2} z_i$$

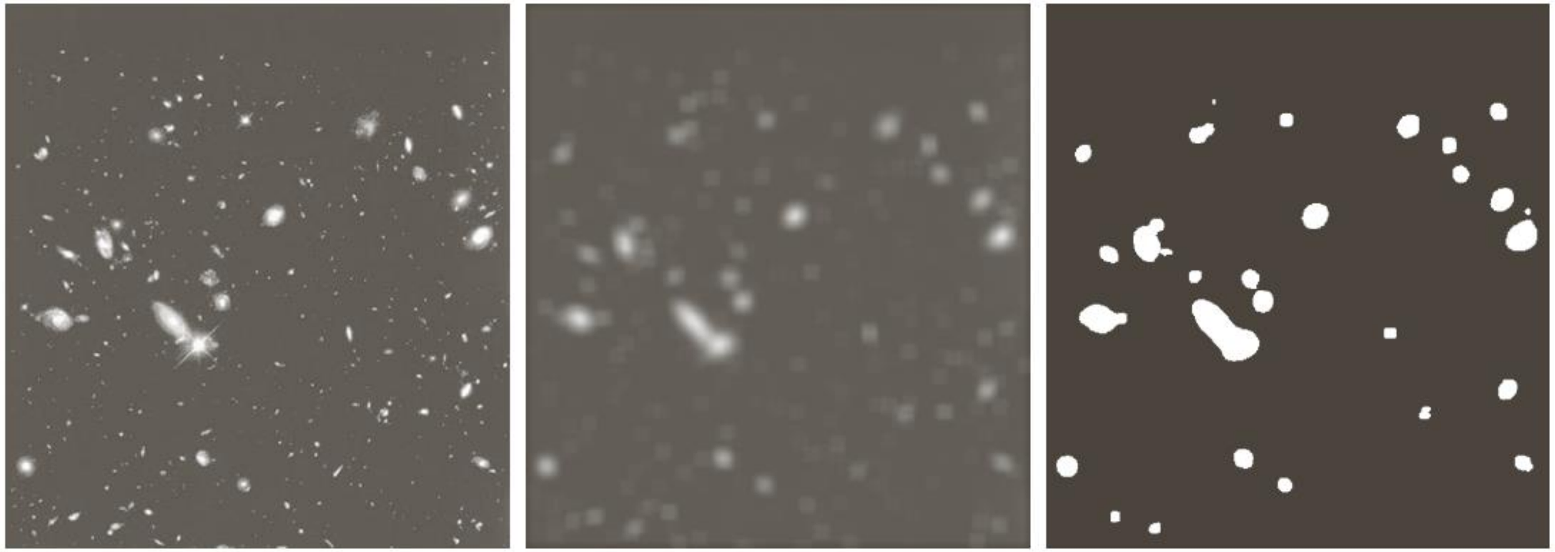
  - m should be odd.
  - Example: 3x3 window, $z_5 = 1/9(z_1+z_2+\ldots+z_9)$

- 3x3, 5x5 average mask: $\frac{1}{9}\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{bmatrix}$ $\frac{1}{25}\begin{bmatrix}1 & 1 & 1 & 1 & 1\\ 1 & 1 & 1 & 1 & 1\\ 1 & 1 & 1 & 1 & 1\\ 1 & 1 & 1 & 1 & 1\\ 1 & 1 & 1 & 1 & 1\end{bmatrix}$
- Remove the noise (pro), blur the edges (con)
- Effect of averaging/mask size?
  - border

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





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.)

Gaussian Operators

- Gaussian mask:

$$G(x,y)=\frac{1}{2\pi\sigma^2}e^{-\frac{x^2+y^2}{\sigma^2}}$$

  - $z=\frac{\sum_{i=1}^{m^2} w_i z_i}{\sum_{i=1}^{m^2} w_i}$, where each weight is: $w_i = G(x,y);$

	w_1	w_2	w_3	
	w_4	w_5	w_6	
	w_7	w_8	w_9	

	G(-1,-1)	G(-1,0)	G(-1,1)	
	G(0,-1)	G(0,0)	G(0,1)	
	G(1,-1)	G(1,0)	G(1,1)	

$$\frac{1}{273}\begin{bmatrix} 1 & 4 & 7 & 4 & 1 \\ 4 & 16 & 26 & 16 & 4 \\ 7 & 26 & 41 & 26 & 7 \\ 4 & 16 & 26 & 16 & 4 \\ 1 & 4 & 7 & 4 & 1 \end{bmatrix}$$

Normal Distribution

- In theory, the Gaussian distribution is non-zero everywhere=> infinitely large mask.
- In practice it is effectively zero more than about $\pm 3\sigma$ from the mean, and so we can truncate the kernel at this point.
- To create a Gaussian mask that is n-pixels in diameter, centered around 0:
  - $n = 6\sigma$,
  - $\mu=0$,
  - Place Gaussian in image center: x-n/2, y-n/2.
  - good practice pick odd n

Operators

- Weighted average operation
- Proper selection of coefficients (weights) allows for operations such as
  - noise reduction
  - region thinning
  - edge detection
- Example: 3x3 weighted avg mask:

$$\frac{1}{9}\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

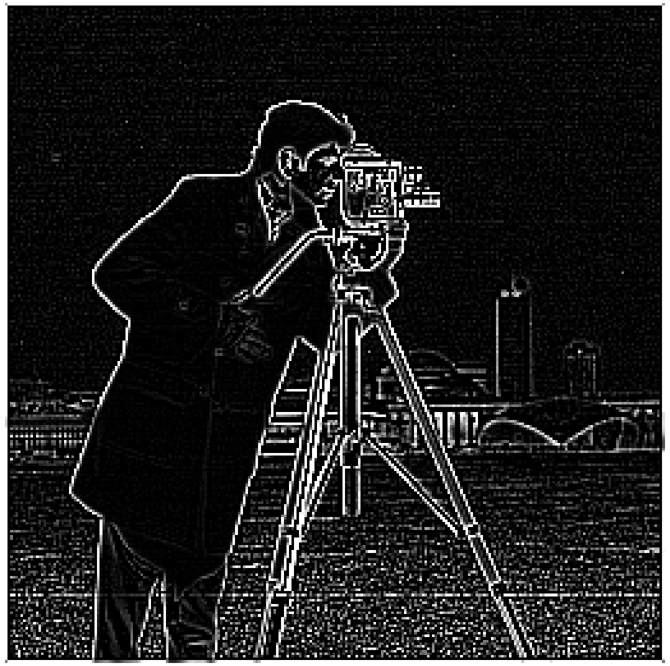
Laplacian Operators

0	1	0
1	−4	1
0	1	0

1	1	1
1	−8	1
1	1	1

0	−1	0
−1	4	−1
0	−1	0

−1	−1	−1
−1	8	−1
−1	−1	−1



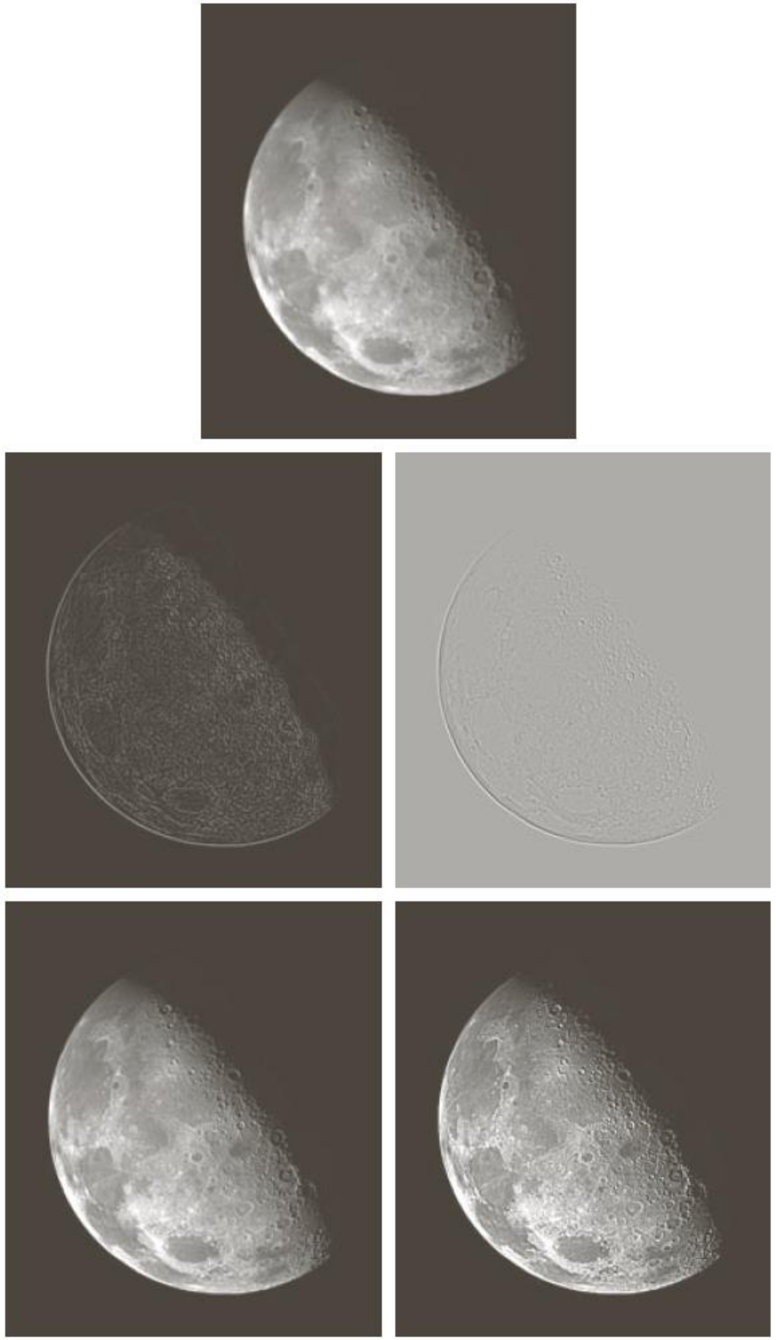


FIGURE 3.38
(a) Blurred image of the North Pole of the moon. (b) Laplacian without scaling. (c) Laplacian with scaling. (d) Image sharpened using the mask in Fig. 3.37(a). (e) Result of using the mask in Fig. 3.37(b). (Original image courtesy of NASA.)

Ordered Filters

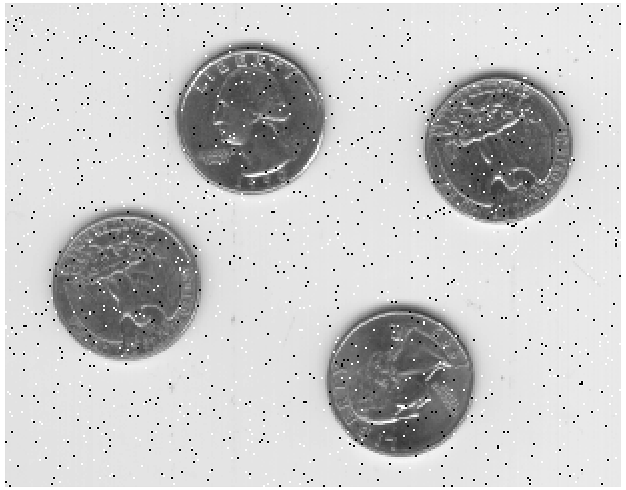
- Response is based on ordering (ranking) the pixels contained in the image area encompassed by the filter
- Replace the center of the mask with a nonlinear value
- Median mask:
  - Sort values and choose the one in the center
  - Removes salt-pepper noise, less blurring removing small clusters
  - Examples:

$$\begin{bmatrix} 10 & 20 & 10 \\ 25 & 10 & 75 \\ 90 & 85 & 100 \end{bmatrix}$$ Sort:(10,10,10,20,**25,**75,85,90,100) => $$\begin{bmatrix} 10 & 20 & 10 \\ 25 & 25 & 75 \\ 90 & 85 & 100 \end{bmatrix}$$

- Max/Min
  -

$$\begin{bmatrix} 10 & 20 & 10 \\ 25 & 10 & 75 \\ 90 & 85 & 100 \end{bmatrix} \qquad \begin{bmatrix} 10 & 20 & 10 \\ 25 & 100 & 75 \\ 90 & 85 & 100 \end{bmatrix}$$

Original



Salt&Pepper
noise added

3x3
averaging

3x3
median