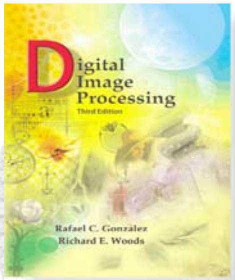


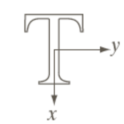
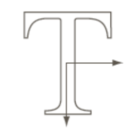
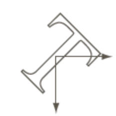
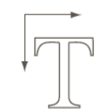
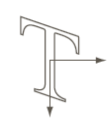
Lecture #4

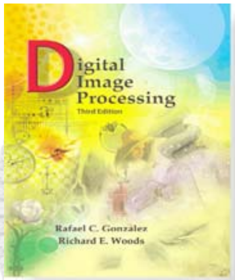
- Image Warping
- Spatial & gray level interpolation
- Intensity transformations

Geometric Transformations

$$\begin{bmatrix} x & y & 1 \end{bmatrix} = \begin{bmatrix} v & w & 1 \end{bmatrix} T = \begin{bmatrix} v & w & 1 \end{bmatrix} \begin{bmatrix} t_{11} & t_{12} & 0 \\ t_{21} & t_{22} & 0 \\ t_{31} & t_{32} & 1 \end{bmatrix}$$

TABLE 2.2
Affine transformations based on Eq. (2.6.–23).

Transformation Name	Affine Matrix, T	Coordinate Equations	Example
Identity	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = v$ $y = w$	
Scaling	$\begin{bmatrix} c_x & 0 & 0 \\ 0 & c_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = c_x v$ $y = c_y w$	
Rotation	$\begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = v\cos\theta - w\sin\theta$ $y = v\cos\theta + w\sin\theta$	
Translation	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ t_x & t_y & 1 \end{bmatrix}$	$x = v + t_x$ $y = w + t_y$	
Shear (vertical)	$\begin{bmatrix} 1 & 0 & 0 \\ s_v & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = v + s_v w$ $y = w$	
Shear (horizontal)	$\begin{bmatrix} 1 & s_h & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = v$ $y = s_h v + w$	


Digital Image Processing
Rafael C. Gonzalez
Richard E. Woods

Rotation

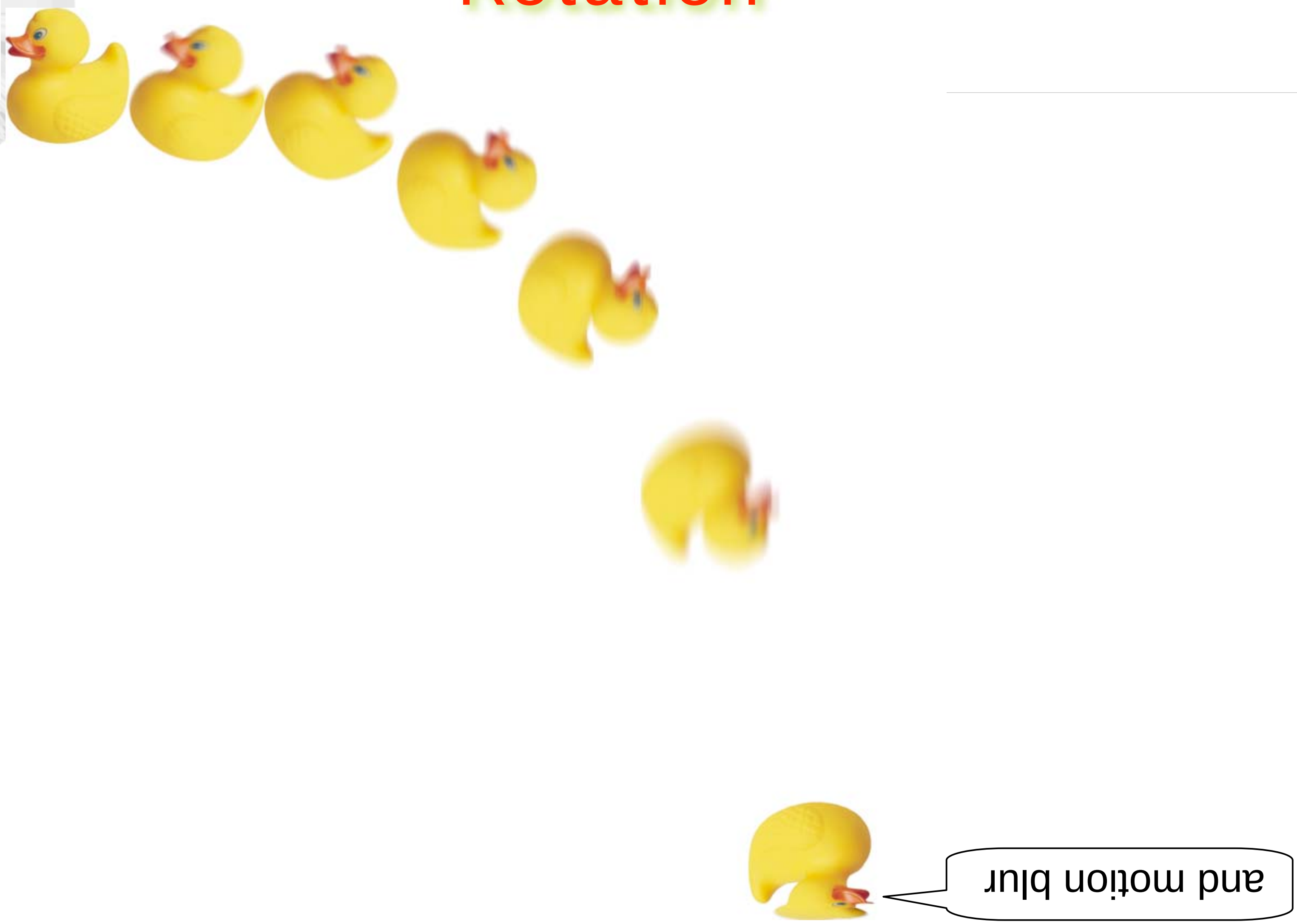
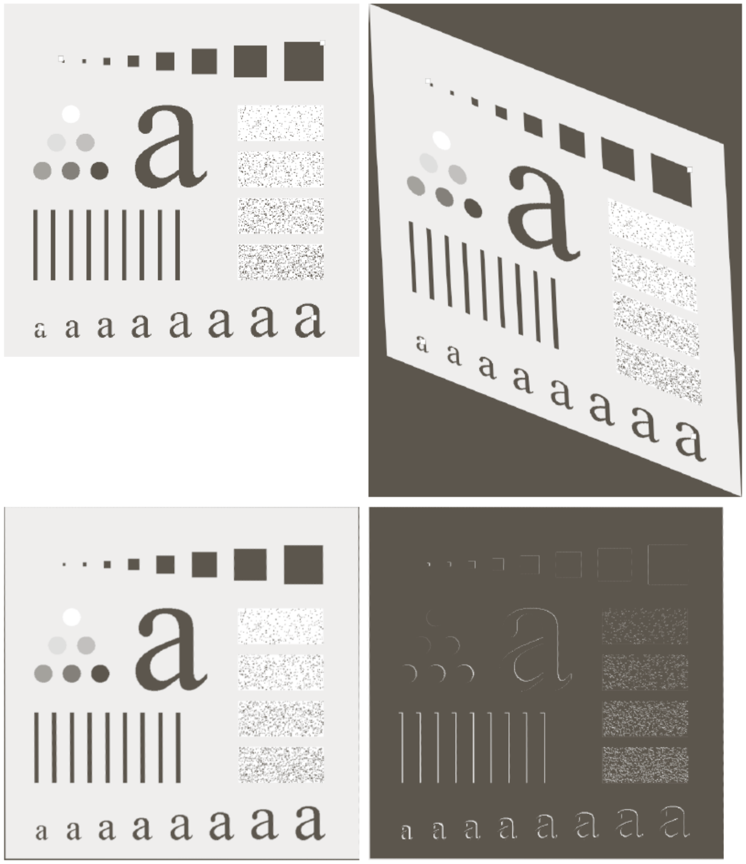

and motion blur

Image Registration

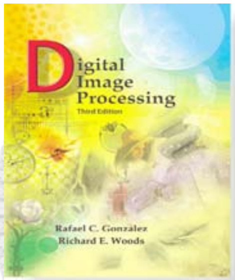


a b
c d

FIGURE 2.37 Image registration. (a) Reference image. (b) Input (geometrically distorted image). Corresponding tie points are shown as small white squares near the corners. (c) Registered image (note the errors in the borders). (d) Difference between (a) and (c), showing more registration errors.

Image Warping

1999-2007 by Richard Alan Peters II



Geometric Transformations (Warping)

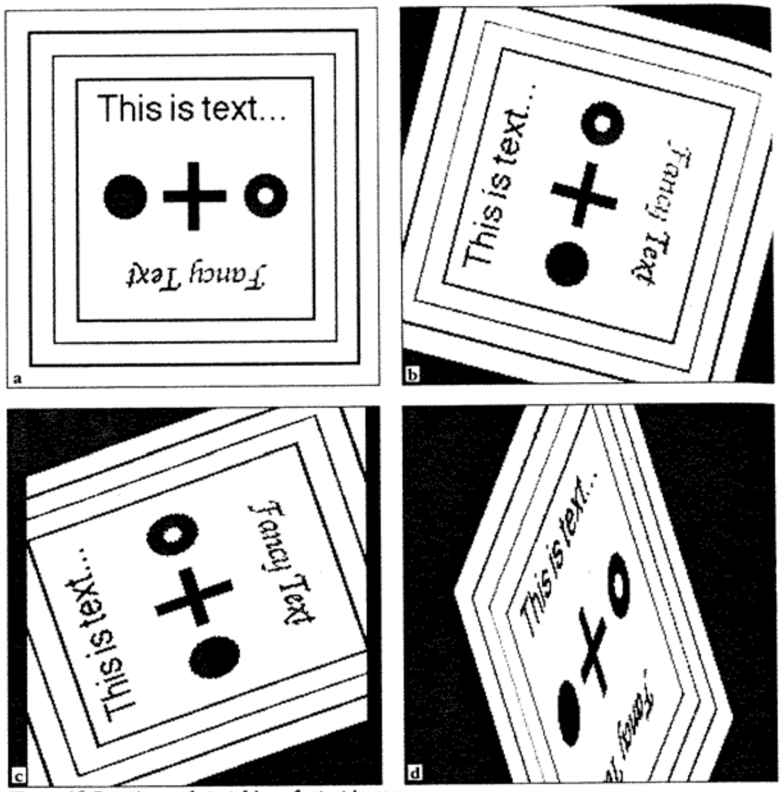
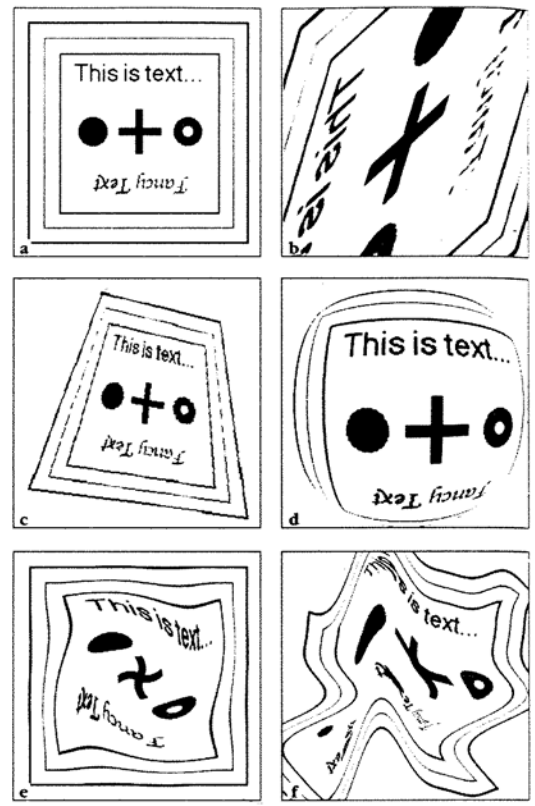
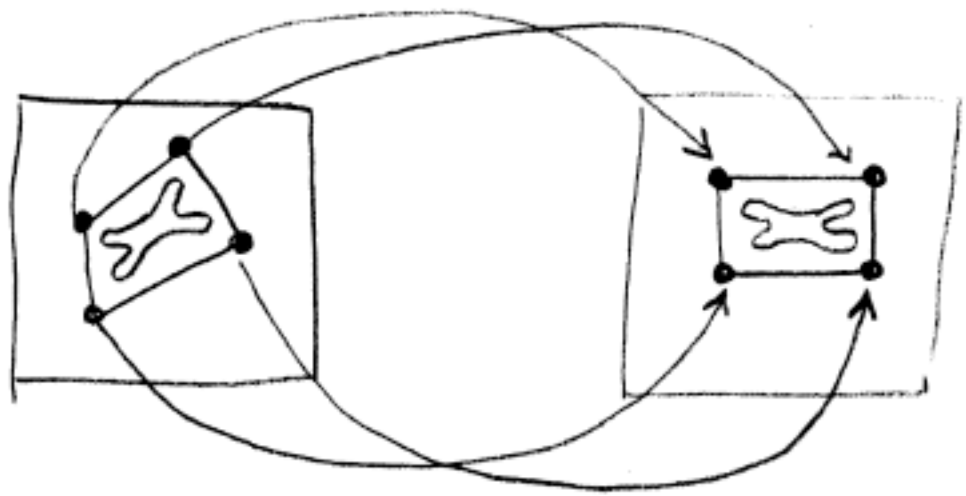


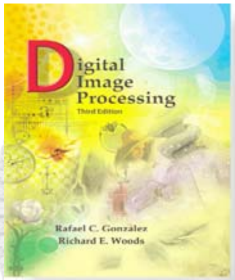
Figure 46. Rotation and stretching of a test image:
a) *original;*
b) *rotation only, no change in scale;*
c) *rotation and uniform stretching while maintaining angles;*
d) *general rotation and stretching.*

Figure 49. Some additional examples of image warping:
a) *original test image;*
b) *linear warping with reversal;*
c) *quadratic warping showing trapezoidal foreshortening (no interpolation);*
d) *cubic warping in which lines are curved (approximation here is to a spherical surface);*
e) *twisting the center of the field while holding the edges fixed (also cubic warping);*
f) *arbitrary warping in which higher order and trigonometric terms are required.*



Spatial Warping





Polynomial Spatial Warp

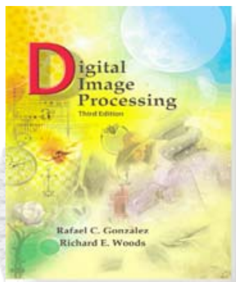
- Represent the coordinate transformations as polynomials in x and y

$$G(x,y) = F(x',y') = F(ax+by+cxy+d, ex+fy+gxy+h)$$

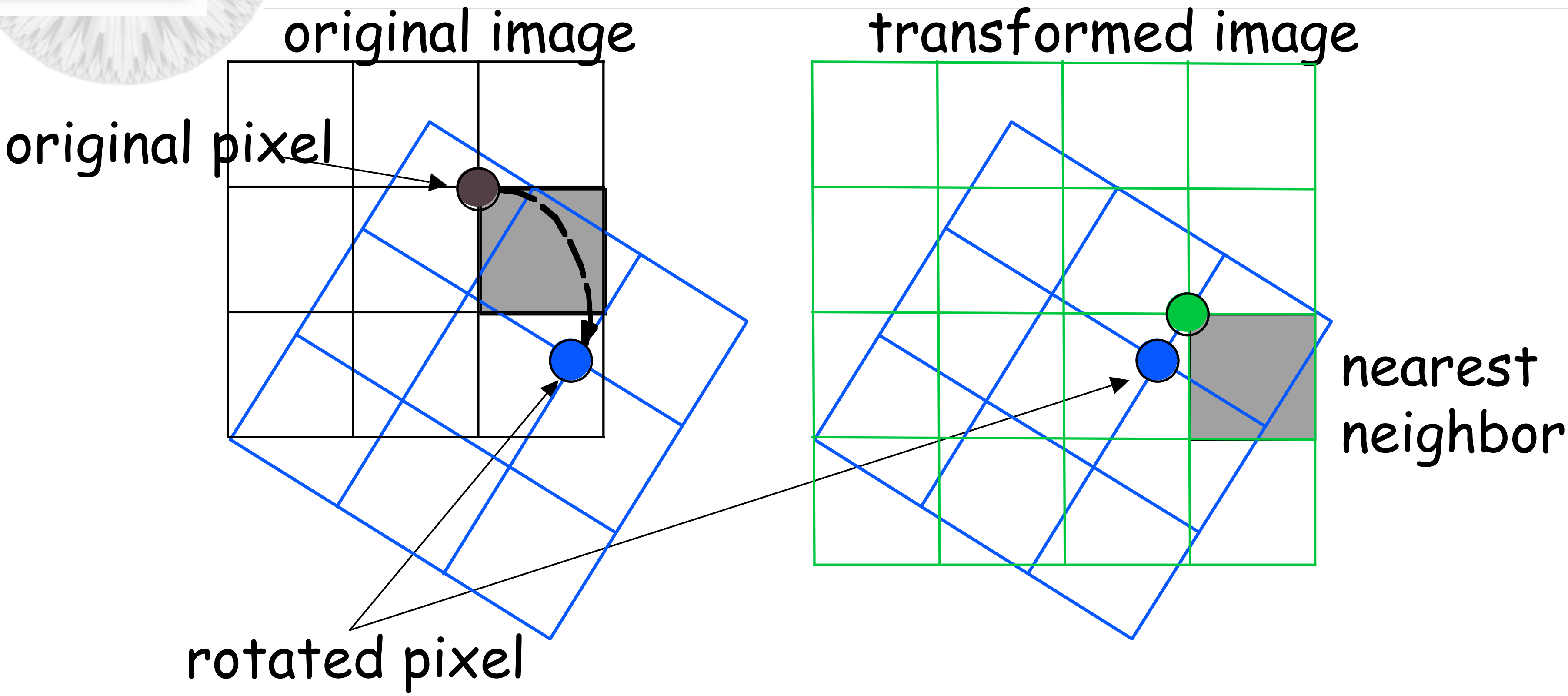
- Using corresponding point pairs write the coordinate transformations as systems of linear equations and put them in matrix form

$$\begin{aligned} x_1' &= ax_1 + by_1 + cx_1y_1 + d \\ x_2' &= ax_2 + by_2 + cx_2y_2 + d \\ x_3' &= ax_3 + by_3 + cx_3y_3 + d \\ x_4' &= ax_4 + by_4 + cx_4y_4 + d \end{aligned} \qquad \begin{bmatrix} x_1' \\ x_2' \\ x_3' \\ x_4' \end{bmatrix} = \begin{bmatrix} x_1 & y_1 & x_1y_1 & 1 \\ x_2 & y_2 & x_2y_2 & 1 \\ x_3 & y_3 & x_3y_3 & 1 \\ x_4 & y_4 & x_4y_4 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix}$$

- solve for the unknown coefficients $\begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix}$



Simple Rotation Example

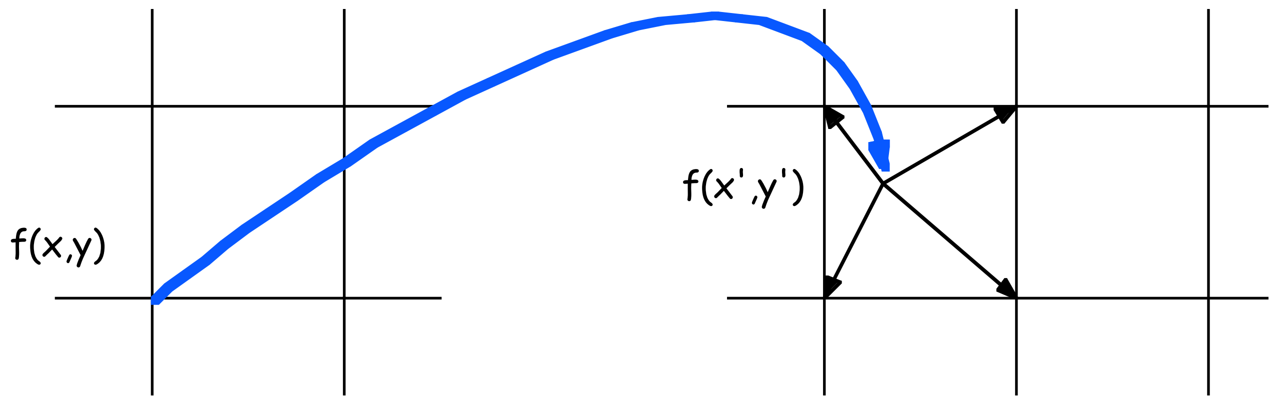


Forward mapping with nearest neighbor assignment of intensity value

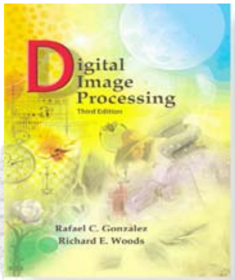
a b c d

FIGURE 2.36 (a) A 300 dpi image of the letter T. (b) Image rotated 21° clockwise using nearest neighbor interpolation to assign intensity values to the spatially transformed pixels. (c) Image rotated 21° using bilinear interpolation. (d) Image rotated 21° using bicubic interpolation. The enlarged sections show edge detail for the three interpolation approaches.

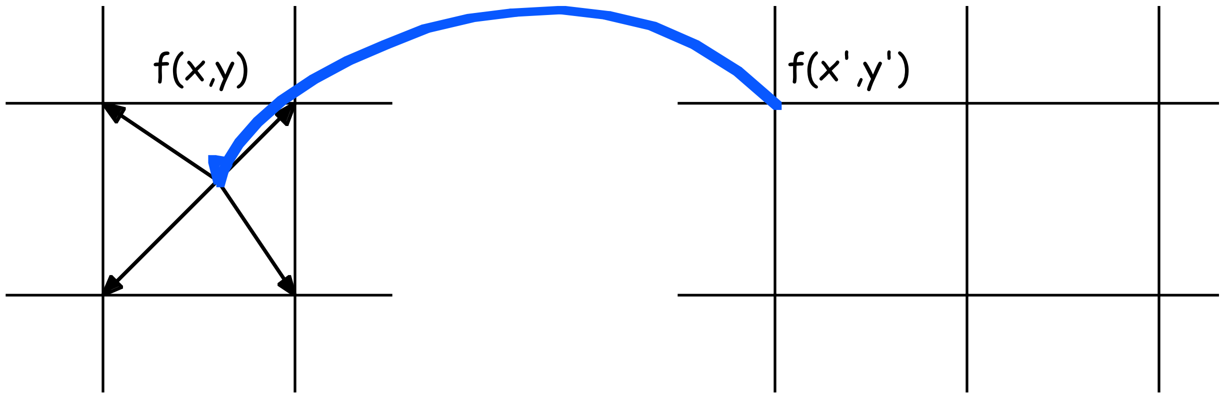
Gray Level Interpolation forward mapping



1. Can result in pixels mapping to pixels outside the image
2. Complex transforms can map several input pixels to the same output pixel
3. Does not guarantee that all output pixels will have a value



Gray Level Interpolation backward mapping (pixel filling)



1. for EACH output pixel (x',y') determine corresponding location (x,y) in input image
2. Use gray level interpolation* for pixels surrounding (x,y) to assign a pixel value f(x',y') to selected output pixel
3. Guarantees that all pixels in output image will have a value

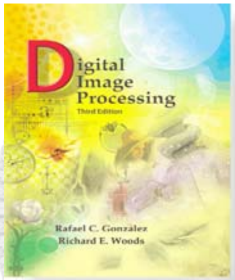
*can be nearest neighbor, bilinear, etc.

Gray Level Interpolation

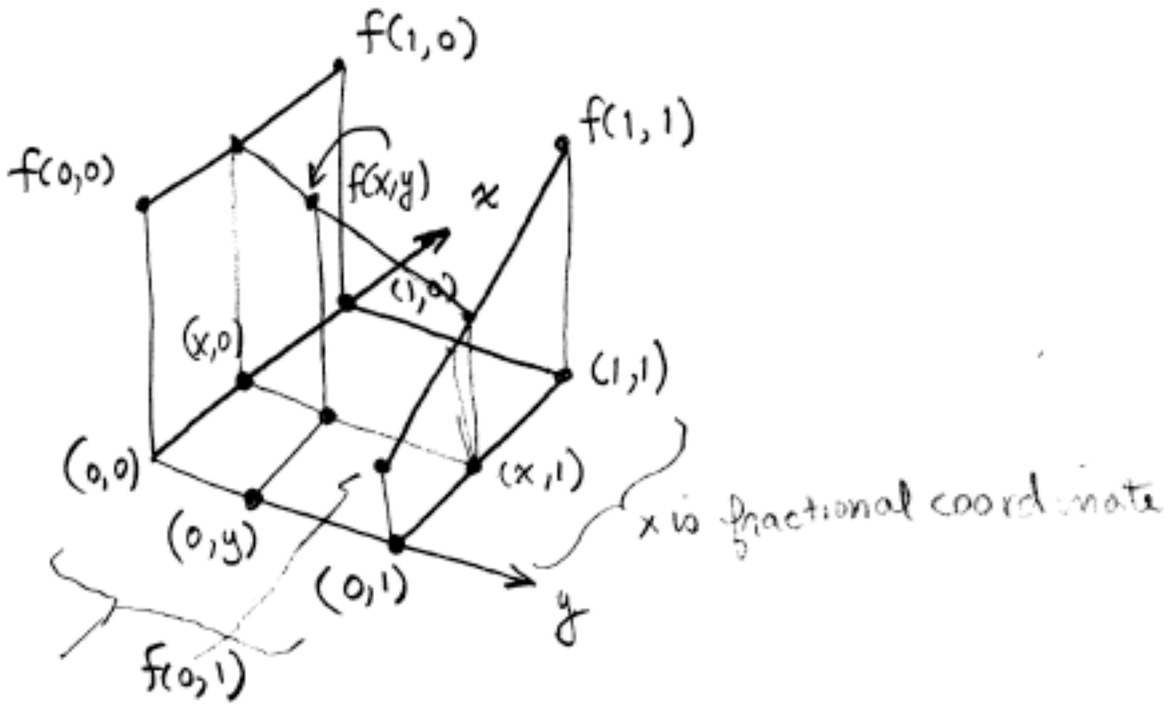
<u>Nearest Neighbor</u>

Assigns output f(x’,y’) the value of the closest pixel

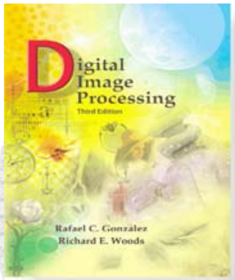
Can produce artifacts when input f(x,y) changes rapidly



Gray level interpolation bilinear



fit hyperbolic paraboloid $f(x,y) = ax + by + cxy + d$



Bilinear Intensity Interpolation

- Linearly interpolate along y=0

$$f(x,0) = f(0,0) + x\left[f(1,0) - f(0,0)\right]$$

- Linearly interpolate along y=1

$$f(x,1) = f(0,1) + x\left[f(1,1) - f(1,0)\right]$$

- Interpolate along x

$$f(x,y) = f(x,0) + y\left[f(x,1) - f(x,0)\right]$$

- Combine equations

$$f(x,y) = \left[f(1,0) - f(0,0)\right]x + \left[f(0,1) - f(0,0)\right]y + \left[f(1,1) + f(0,0) - f(0,1) - f(1,0)\right]xy + f(0,0)$$

Results of interpolation methods

FIGURE 2.36 (a) A 300 dpi image of the letter T. (b) Image rotated 21° clockwise using nearest neighbor interpolation to assign intensity values to the spatially transformed pixels. (c) Image rotated 21° using bilinear interpolation. (d) Image rotated 21° using bicubic interpolation. The enlarged sections show edge detail for the three interpolation approaches.

Interpolation can be applied to vectors

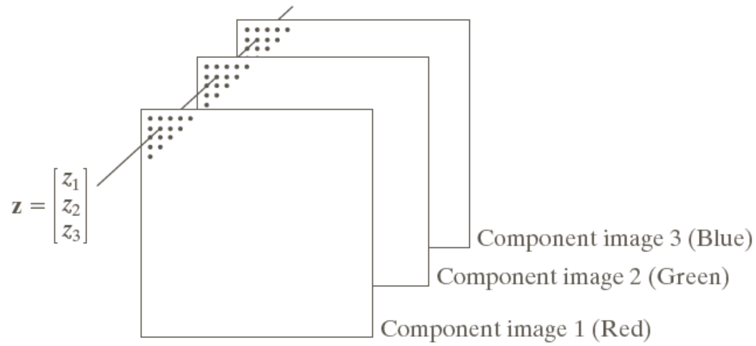
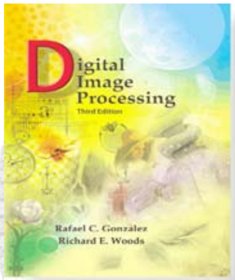


FIGURE 2.38 Formation of a vector from corresponding pixel values in three RGB component images.

Point Processing of Images

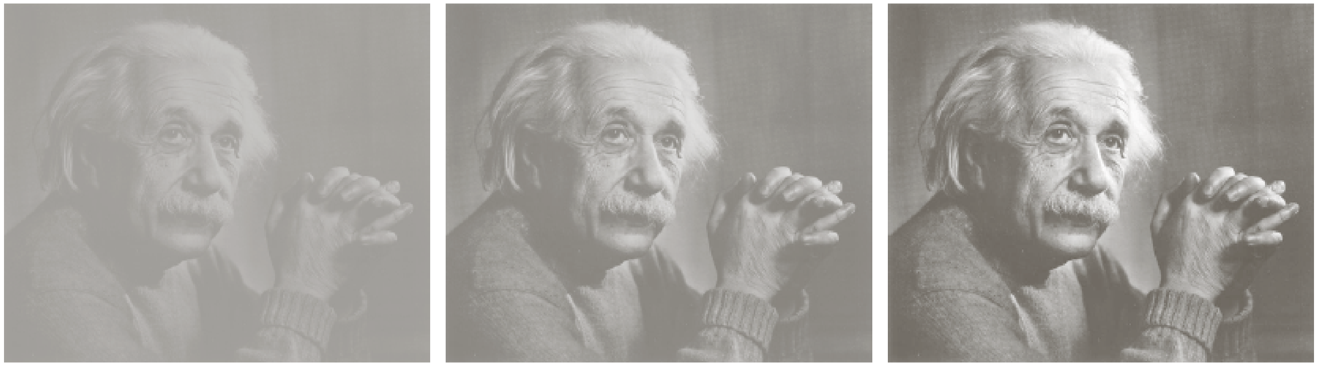
- In a digital image, point = pixel.
- Point processing transforms a pixel's value as a function of its value alone;
- pixel's value does not depend on the values of the pixel's neighbors.



Point Processing of Images

- Brightness and contrast adjustment
- Gamma correction
- Histogram equalization
- Histogram matching
- Color correction.

Image Contrast



a b c

FIGURE 2.41 Images exhibiting (a) low contrast, (b) medium contrast, and (c) high contrast.

Negative Transformation

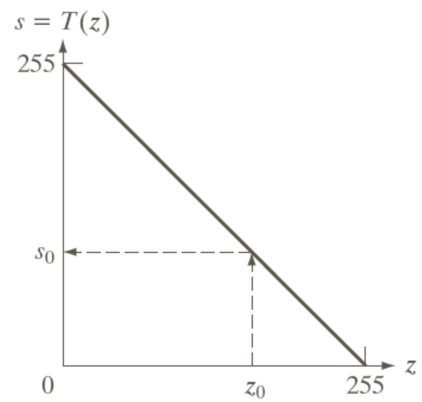
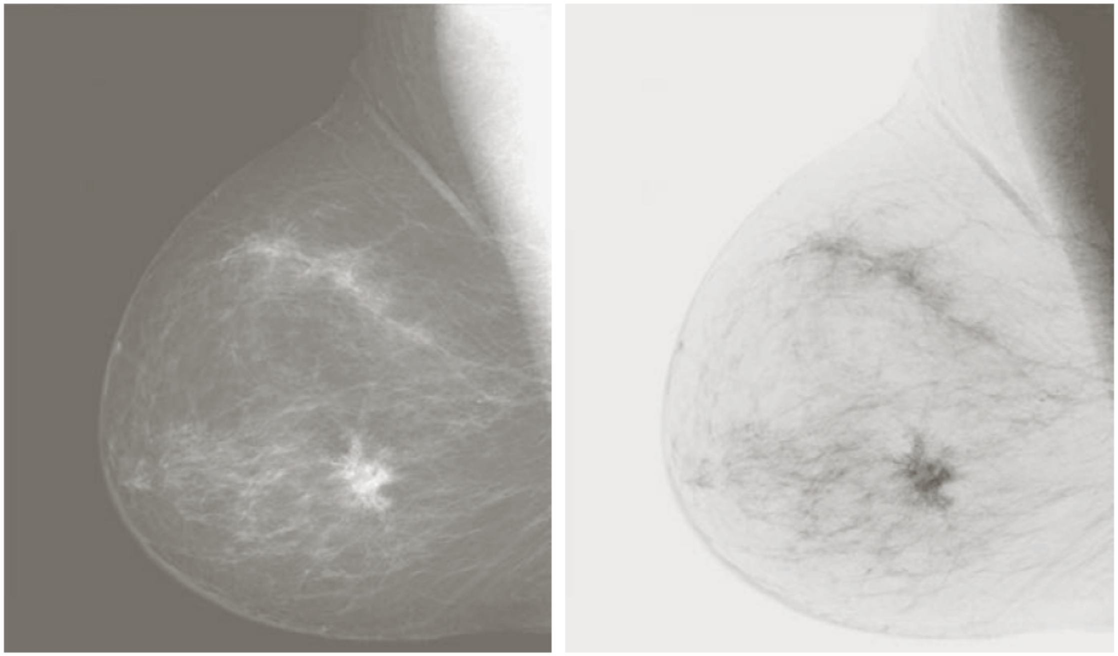


FIGURE 2.34 Intensity transformation function used to obtain the negative of an 8-bit image. The dashed arrows show transformation of an arbitrary input intensity value z_0 into its corresponding output value s_0.

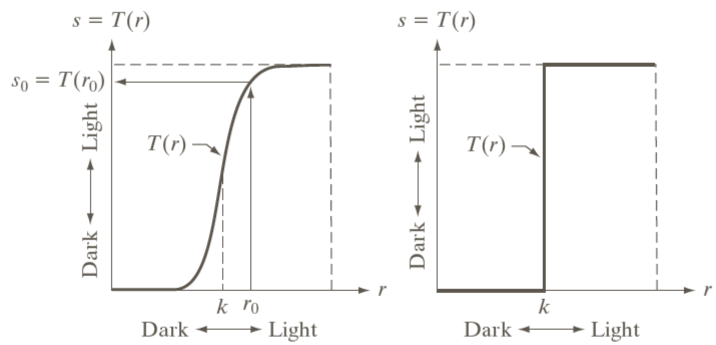
Negative Transformation



a b

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

More Intensity Transformations



a b

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

Log Transformation

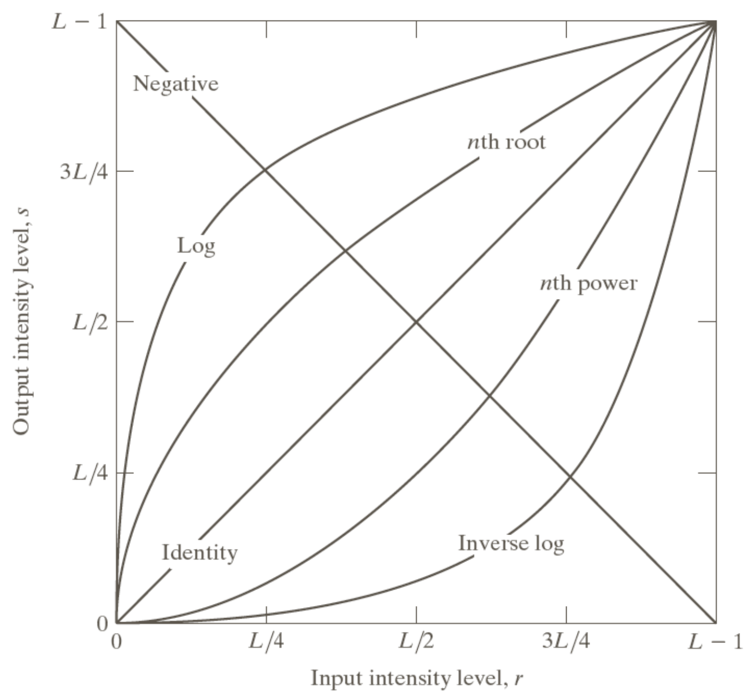
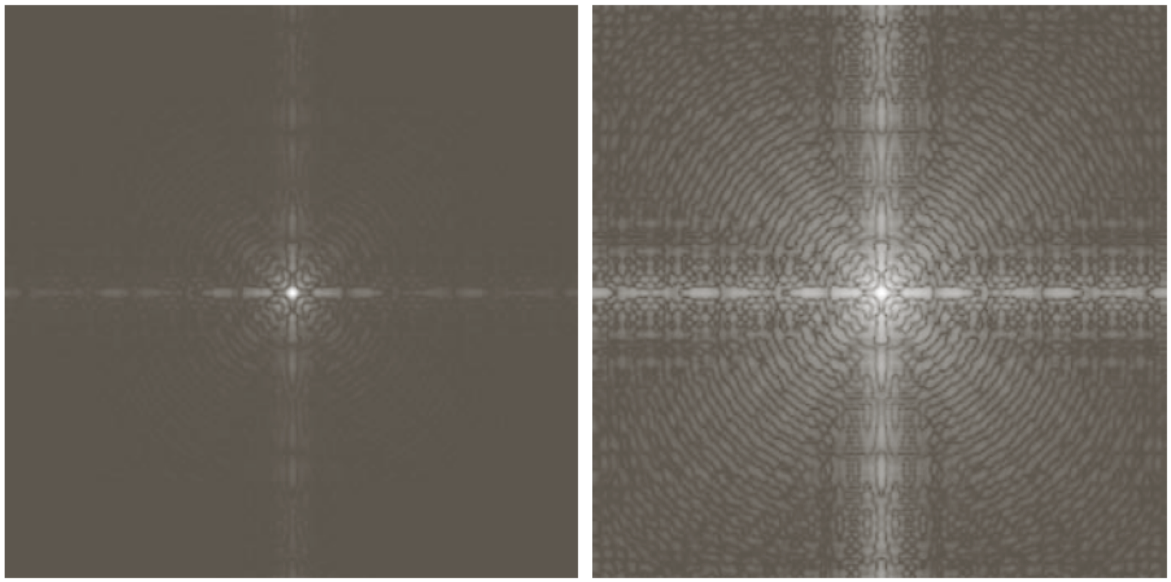


FIGURE 3.3 Some basic intensity transformation functions. All curves were scaled to fit in the range shown.

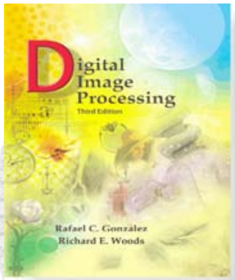
$$s = c\log(1+r)$$

Log Transformation



a b

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



Gamma Transformation

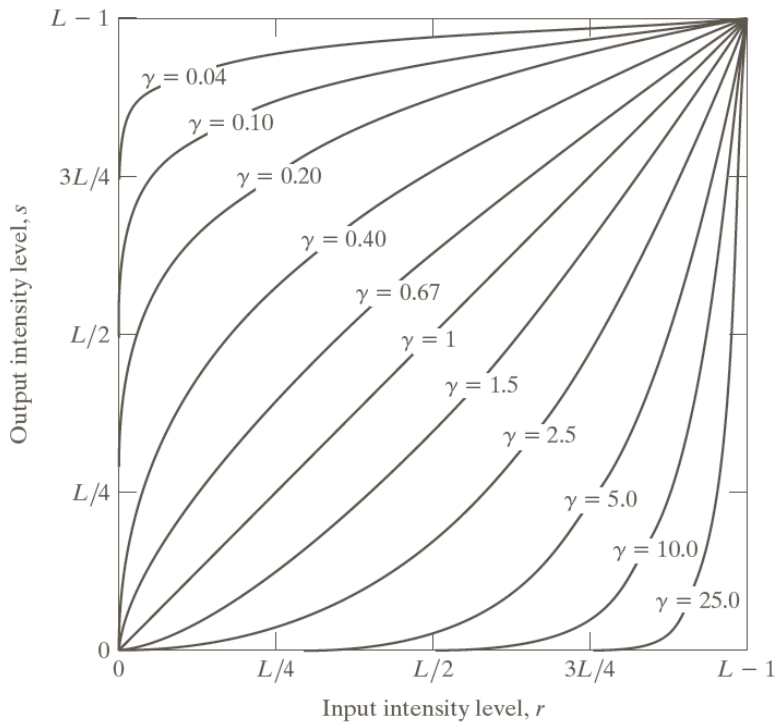
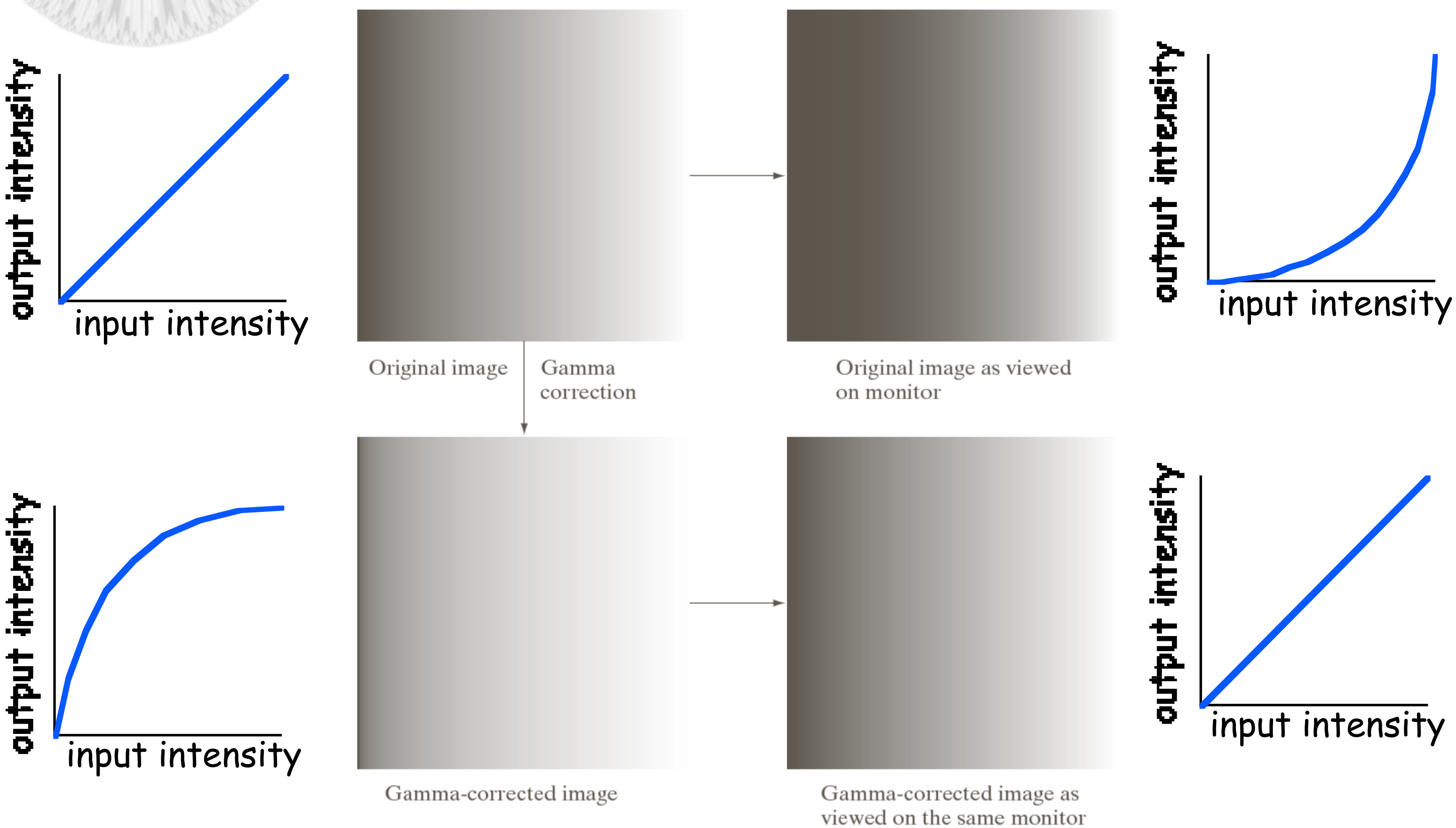


FIGURE 3.6 Plots of the equation $s = cr^{\gamma}$ for various values of γ ($c = 1$ in all cases). All curves were scaled to fit in the range shown.

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

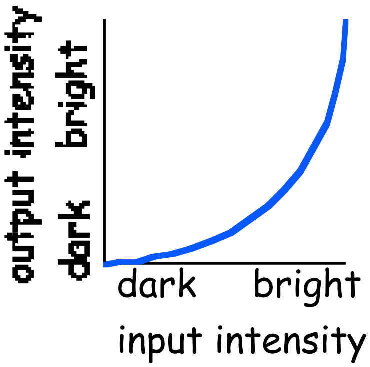
Gamma Transformation

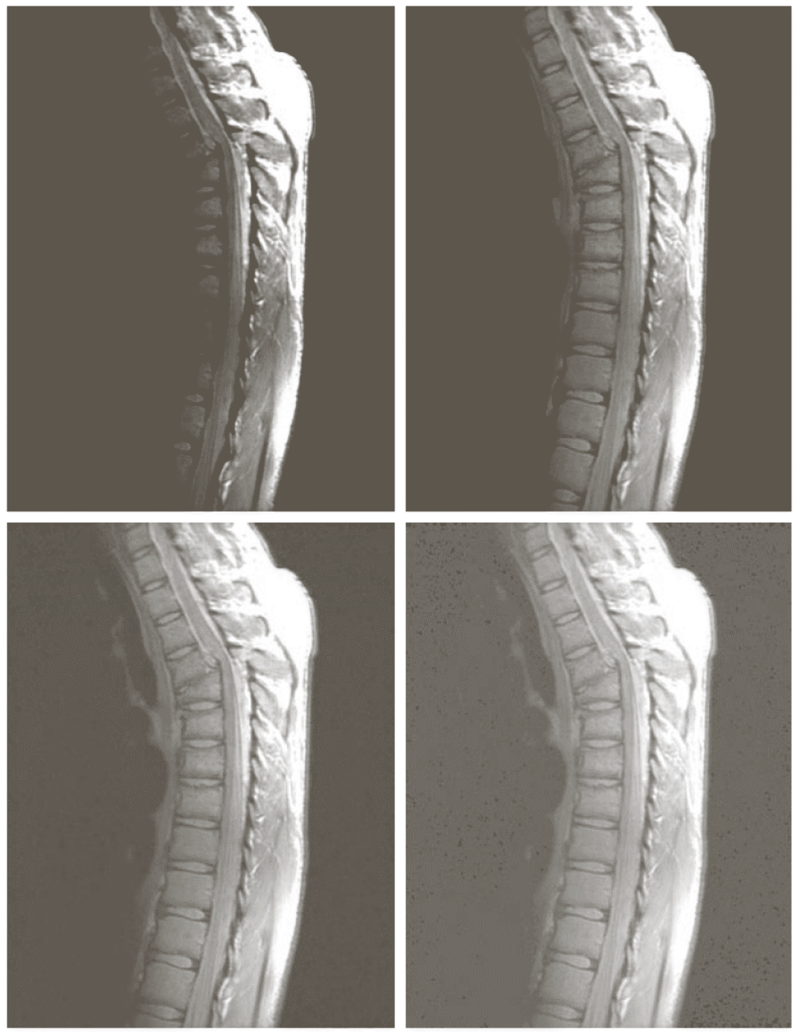


a b
c d

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).

Gamma Transformation

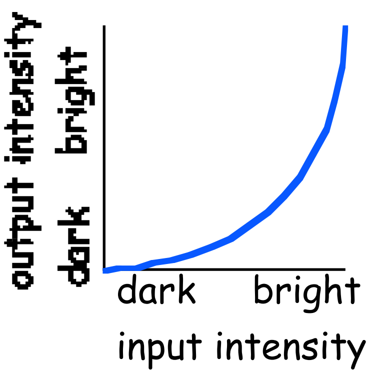




a b
c d

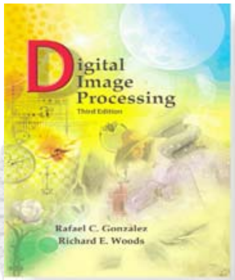
FIGURE 3.8 (a) Magnetic resonance image (MRI) of a fractured human spine. (b)–(d) Results of applying the transformation in Eq. (3.2-3) with $c = 1$ and $\gamma = 0.6, 0.4,$ and 0.3, respectively. (Original image courtesy of Dr. David R. Pickens, Department of Radiology and Radiological Sciences, Vanderbilt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

Gamma Transformation

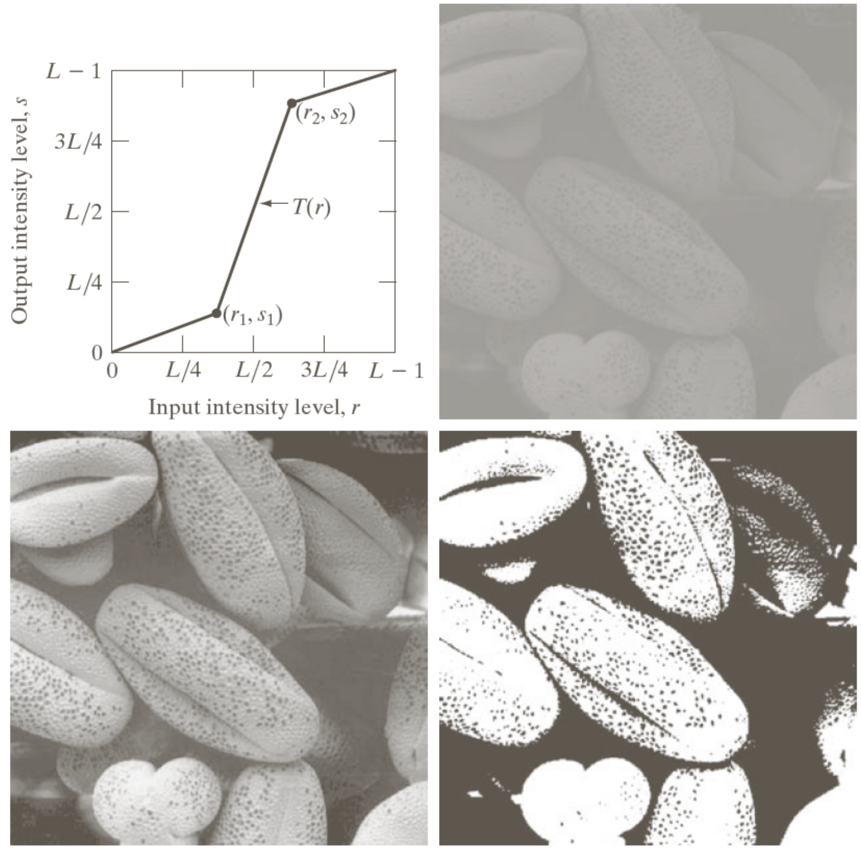


a b
c d

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



Piece-wise Linear Transformations



a b
c d

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

Piece-wise Linear Transformations

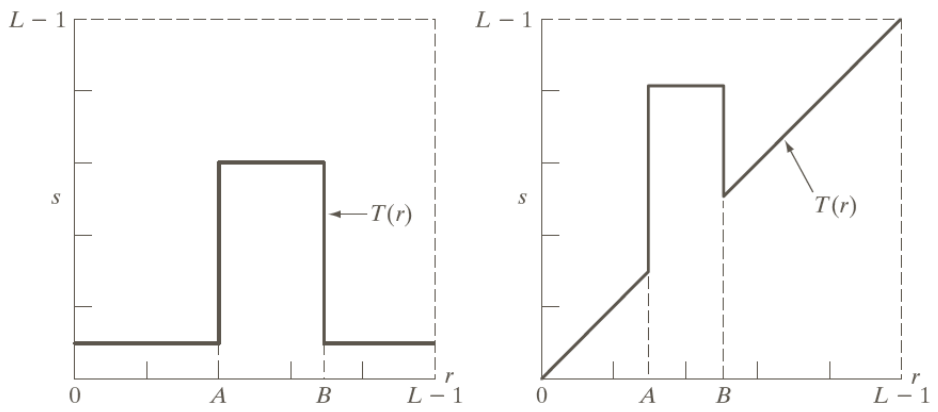
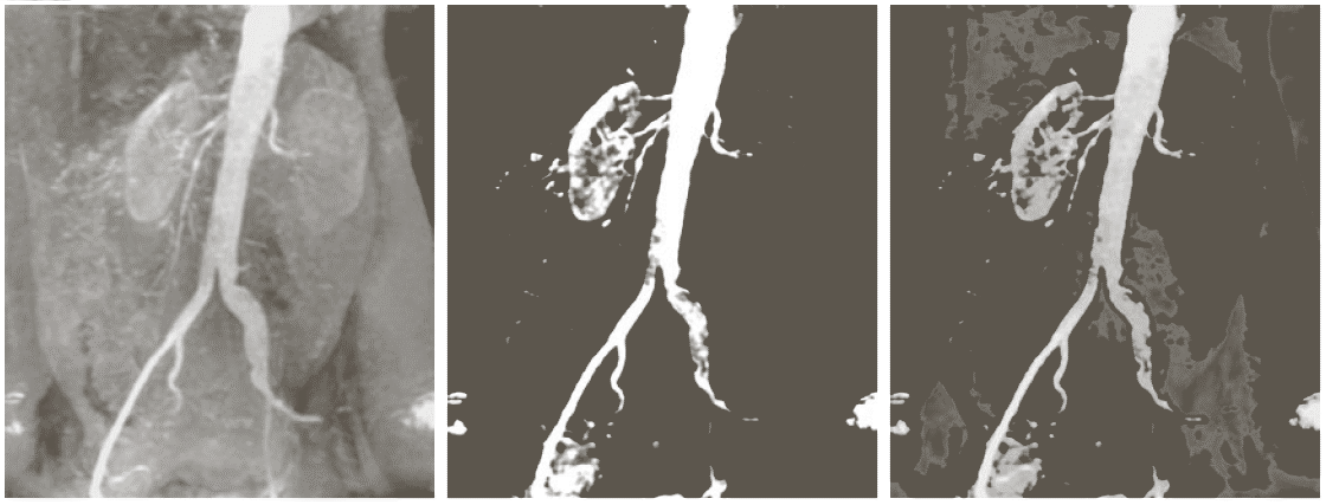


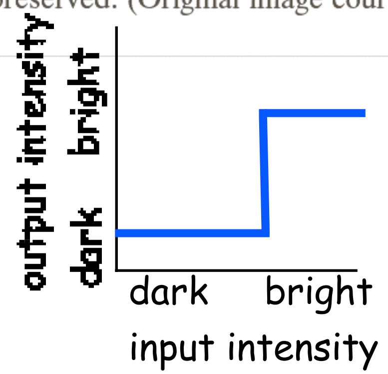
FIGURE 3.11 (a) This transformation highlights intensity range [A, B] and reduces all other intensities to a lower level. (b) This transformation highlights range [A, B] and preserves all other intensity levels.

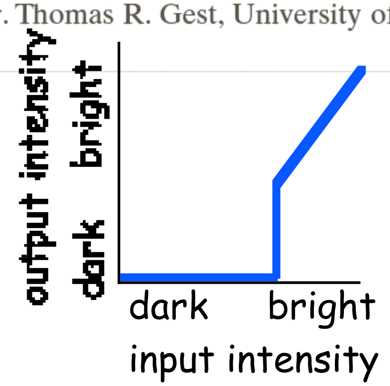
Piece-wise Linear Transformations



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.)





Piece-wise Linear Transformations

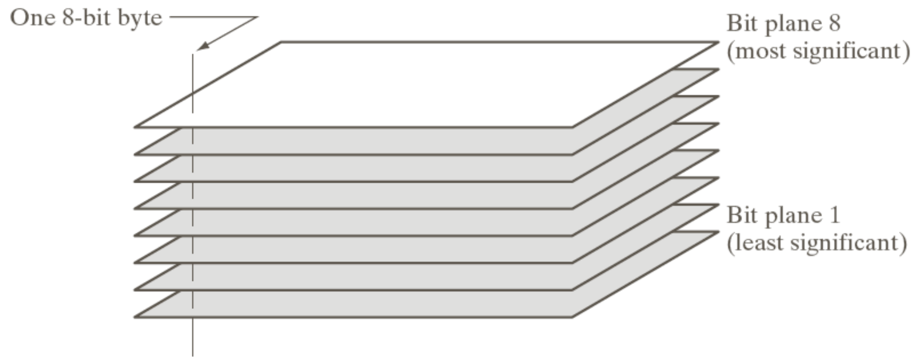


FIGURE 3.13 Bit-plane representation of an 8-bit image.

a	b	c
d	e	f
g	h	i

FIGURE 3.14 (a) An 8-bit gray-scale image of size 500 × 1192 pixels. (b) through (i) Bit planes 1 through 8, with bit plane 1 corresponding to the least significant bit. Each bit plane is a binary image.

a b c

FIGURE 3.15 Images reconstructed using (a) bit planes 8 and 7; (b) bit planes 8, 7, and 6; and (c) bit planes 8, 7, 6, and 5. Compare (c) with Fig. 3.14(a).

Brightness & Contrast

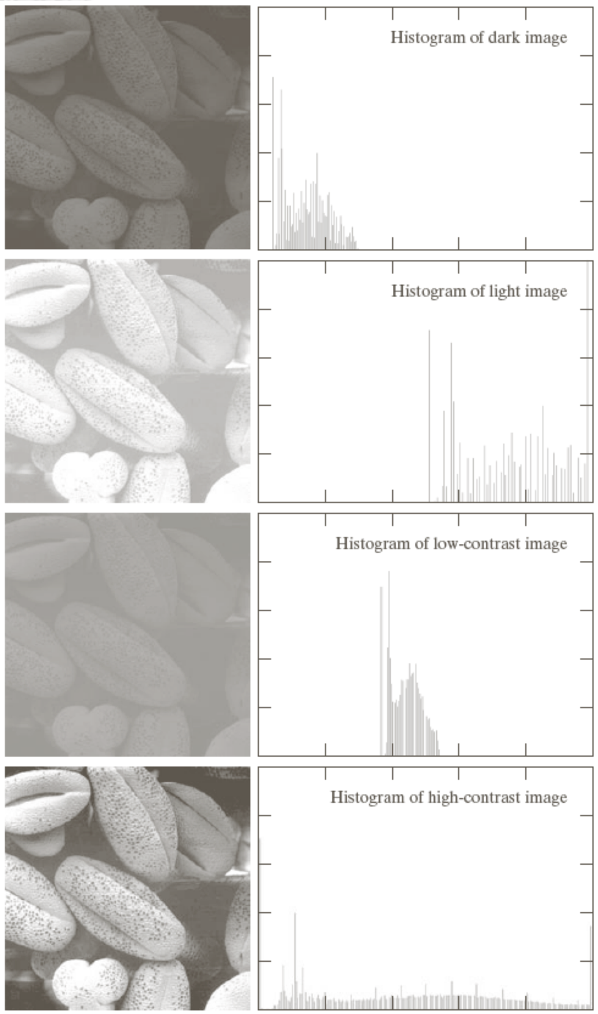


FIGURE 3.16 Four basic image types: dark, light, low contrast, high contrast, and their corresponding histograms.