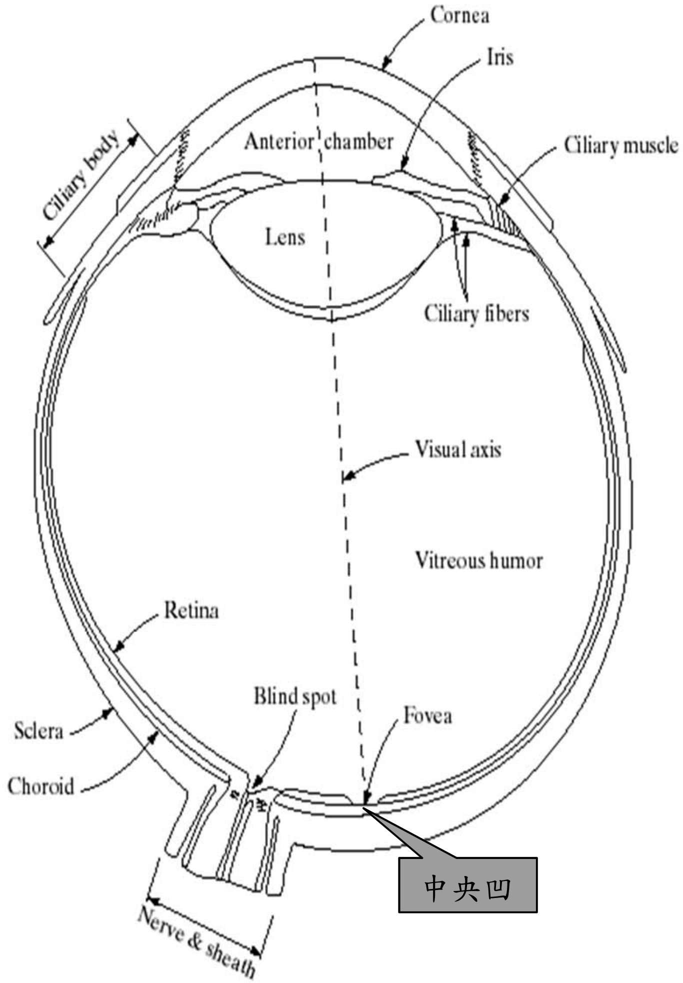
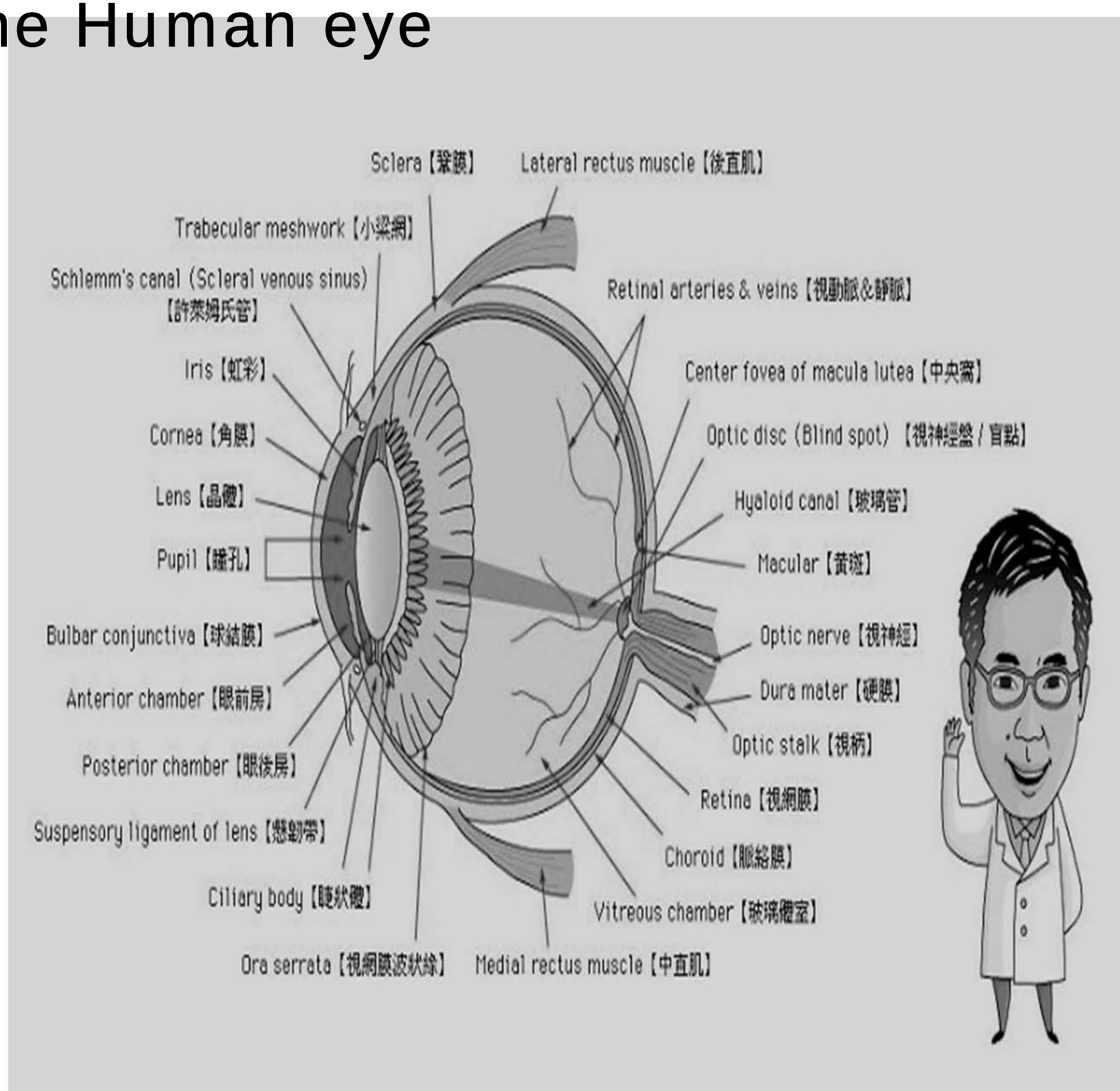
Chapter 2
Digital Image Fundamentals

Outlines

- Elements of Visual Perception
- Light and the Electromagnetic Spectrum
- Image Sensing and Acquisition
- Image Sampling and Quantization
- Some Basic Relationships Between Pixels
- Linear and Nonlinear Operations

Structure of the Human eye





Distribution of rods/cones in the retina

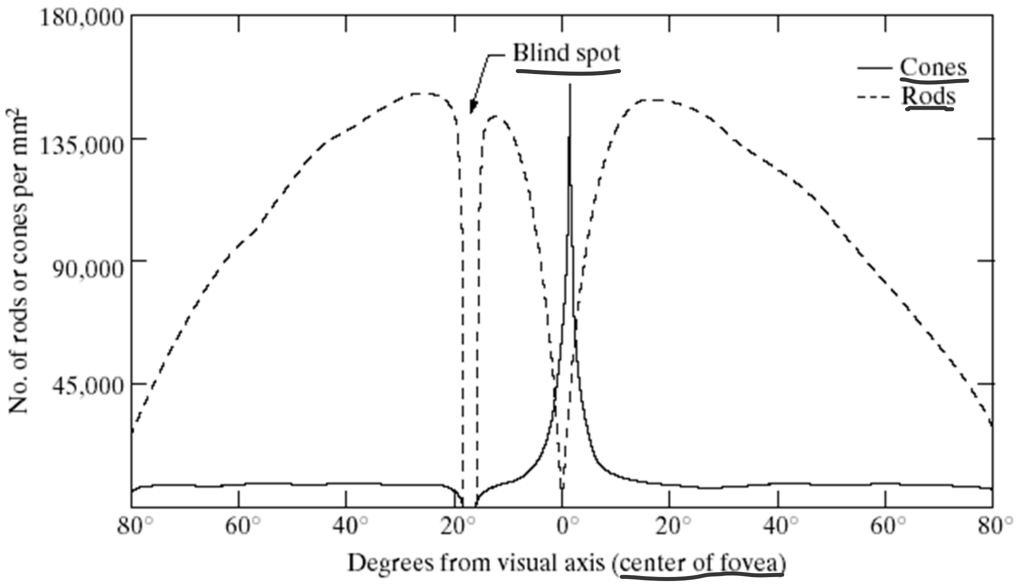
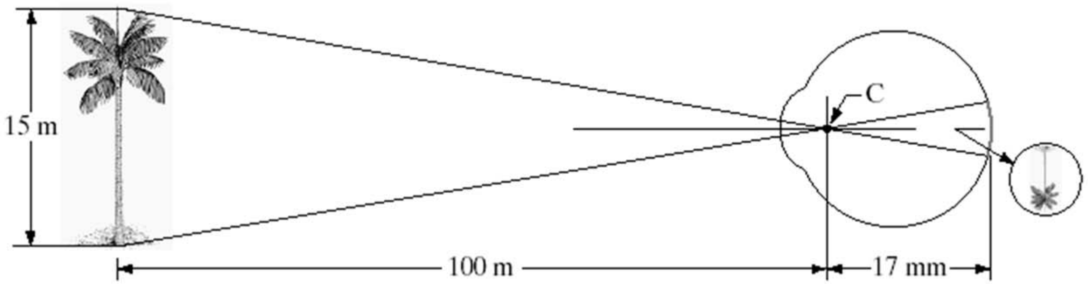


FIGURE 2.2 Distribution of rods and cones in the retina.

- Cones are mostly concentrated at fovea area.
- Cone vision is called photopic or bright-light vision.
- While rods are distributed all over the retina except blind points and fovea.
- Rods are not involved in color vision and are sensitive to low levels of illumination.

Image Formation in the Eye

FIGURE 2.3 Graphical representation of the eye looking at a palm tree. Point C is the optical center of the lens.

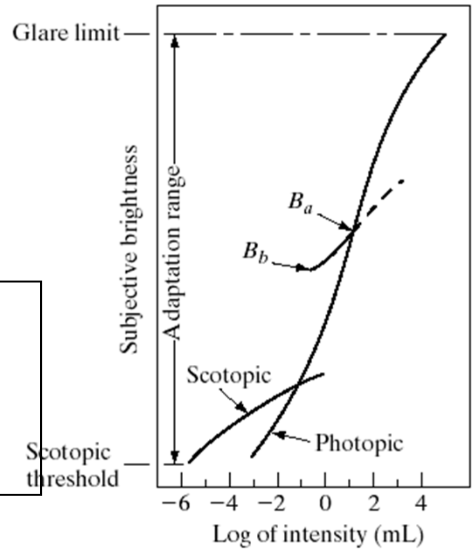


$$15/100 = h/17 \quad \Rightarrow \quad h = 2.55 \text{ mm}$$

Brightness Adaption

- Subjective brightness:
  intensity perceived by the human visual system
- The subjective brightness is a logarithmic function of the light intensity incident in the eye

FIGURE 2.4 Range of subjective brightness sensations showing a particular adaptation level.



Sensing at low illumination is by **rods**, while sensing at high illumination is by **cones**.

Brightness Discrimination

- Weiber Ratio: $\Delta I_c / I$
  ΔI_c is the increment of illumination discriminable 50% of time with backgroumd illumination *I*.

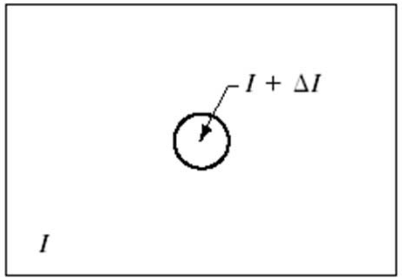
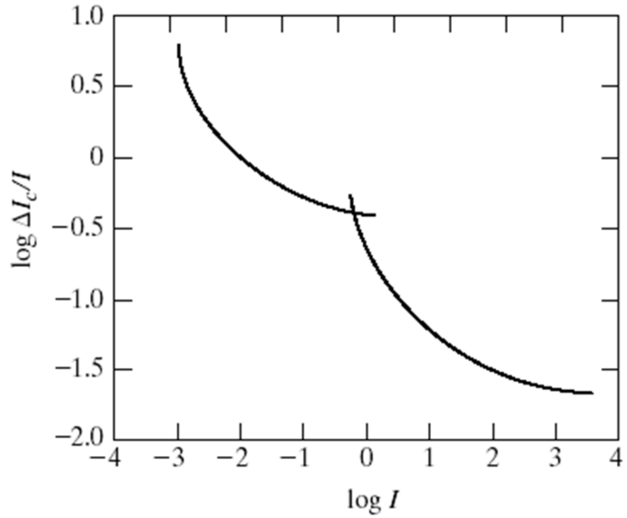


FIGURE 2.5 Basic experimental setup used to characterize brightness discrimination.

Weber ratio as a function of intensity

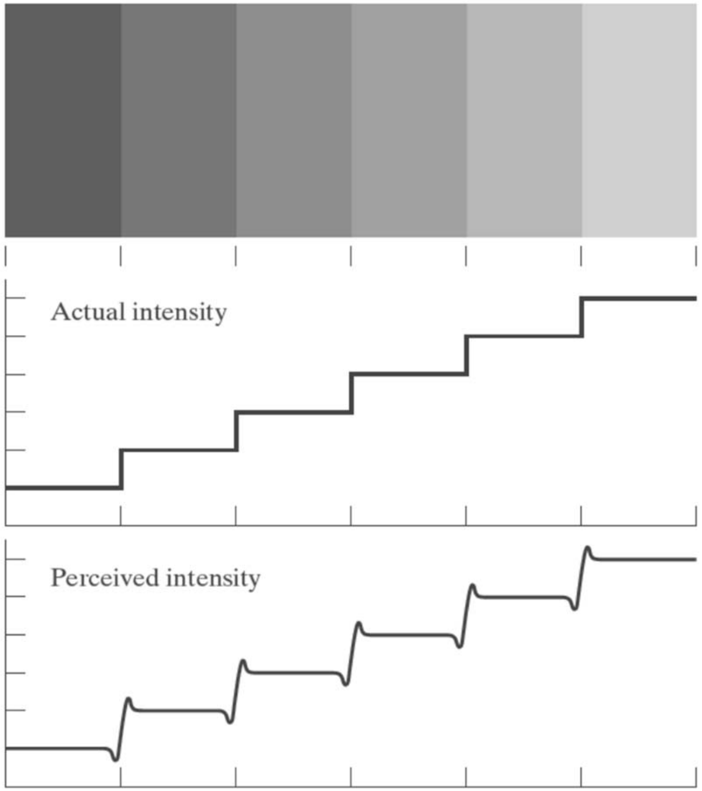
FIGURE 2.6 Typical Weber ratio as a function of intensity.



- **The brightness discrimination is poor at low illumination, and is much improved at higher illumination.**

Perceived brightness is not a simple function of intensity

1. The visual system tends to undershoot or overshoot around the boundary

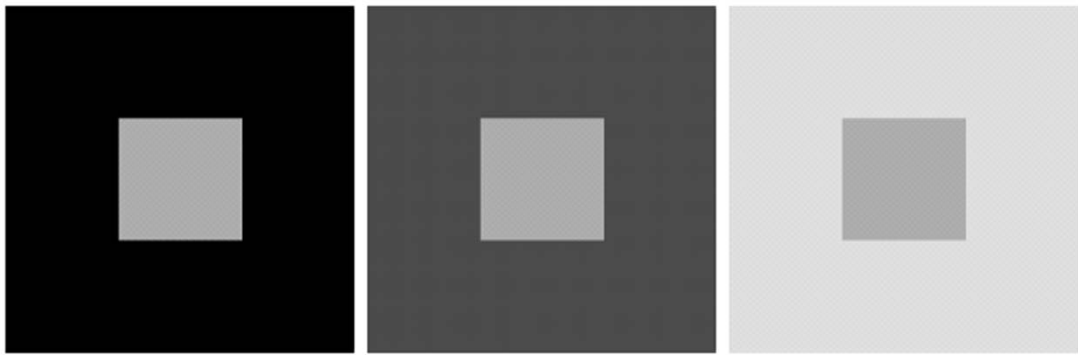


a
b
c

FIGURE 2.7 Illustration of the Mach band effect. Perceived intensity is not a simple function of actual intensity.

Perceived brightness is not a simple function of intensity

2. Simultaneous contrast: the perceived brightness does not simply only depend on the intensity.

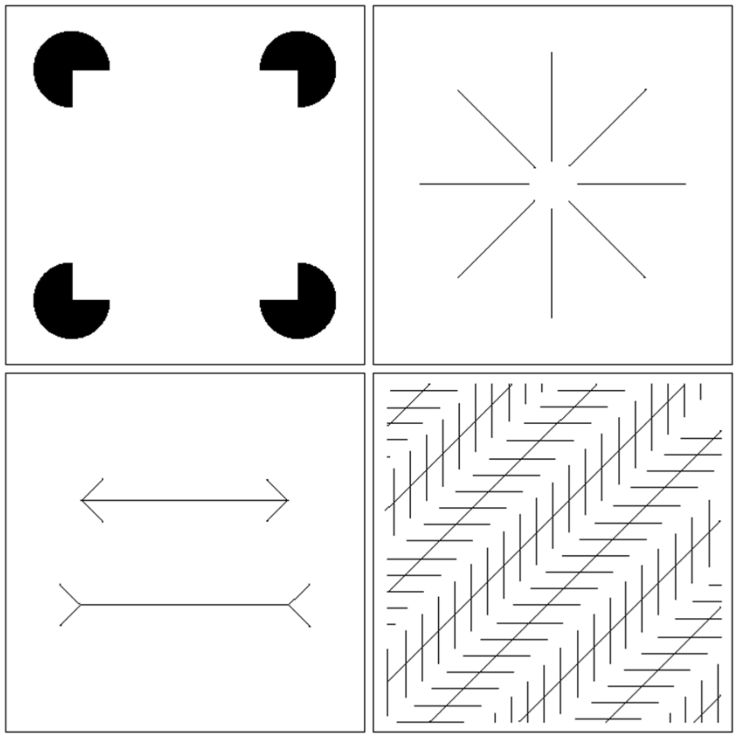


a b c

FIGURE 2.8 Examples of simultaneous contrast. All the inner squares have the same intensity, but they appear progressively darker as the background becomes lighter.

Optical **illusions**

- Optical illusions: the eye fills in non-existing information or wrongly perceives geometrical properties of objects.



a b
c d

FIGURE 2.9 Some well-known optical illusions.

The electromagnetic spectrum

The visible spectrum is a very small portion of the EM spectrum

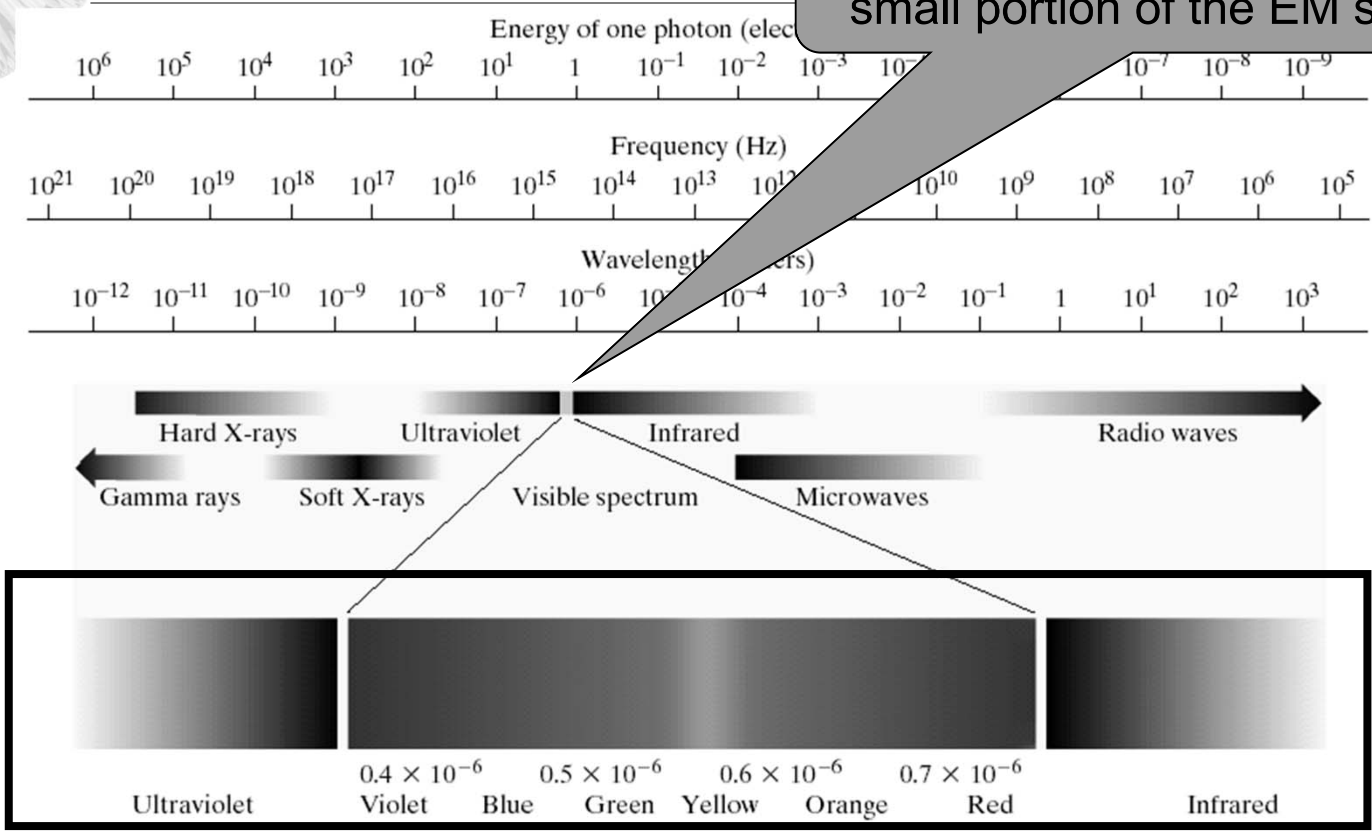


FIGURE 2.10 The electromagnetic spectrum. The visible spectrum is shown zoomed to facilitate explanation, but note that the visible spectrum is a rather narrow portion of the EM spectrum.

The EM expression

Wavelength
(micron μm=10^{-6}m
nanometers 10^{-9}m)

$$\lambda = \frac{c}{\nu} \qquad (2.2\text{-}1)$$

Speed of light
($2.998*10^{8}$ m/s)

Frequency
(Hertz HZ)

Energy
(electron-volt)

$$E = h\nu \qquad (2.2\text{-}2)$$

Plank’s constant

FIGURE 2.11
Graphical representation of one wavelength.

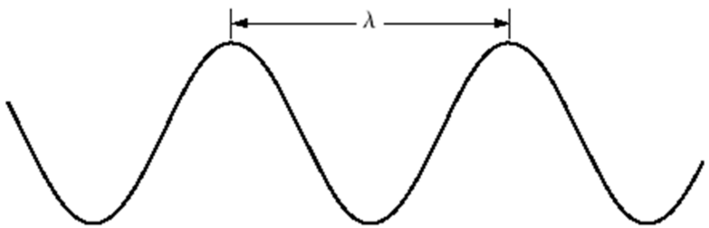


Image sensing and acquiaition

- Purpose: transform illumination energy into digital images
- Three principal sensor arrangements:

1. Single sensor
2. Sensor Stripes (Line sensor)
3. Sensor Arrays (Array sensor)

Three principal sensor arrangements

a
b
c

FIGURE 2.12
(a) Single imaging sensor.
(b) Line sensor.
(c) Array sensor.

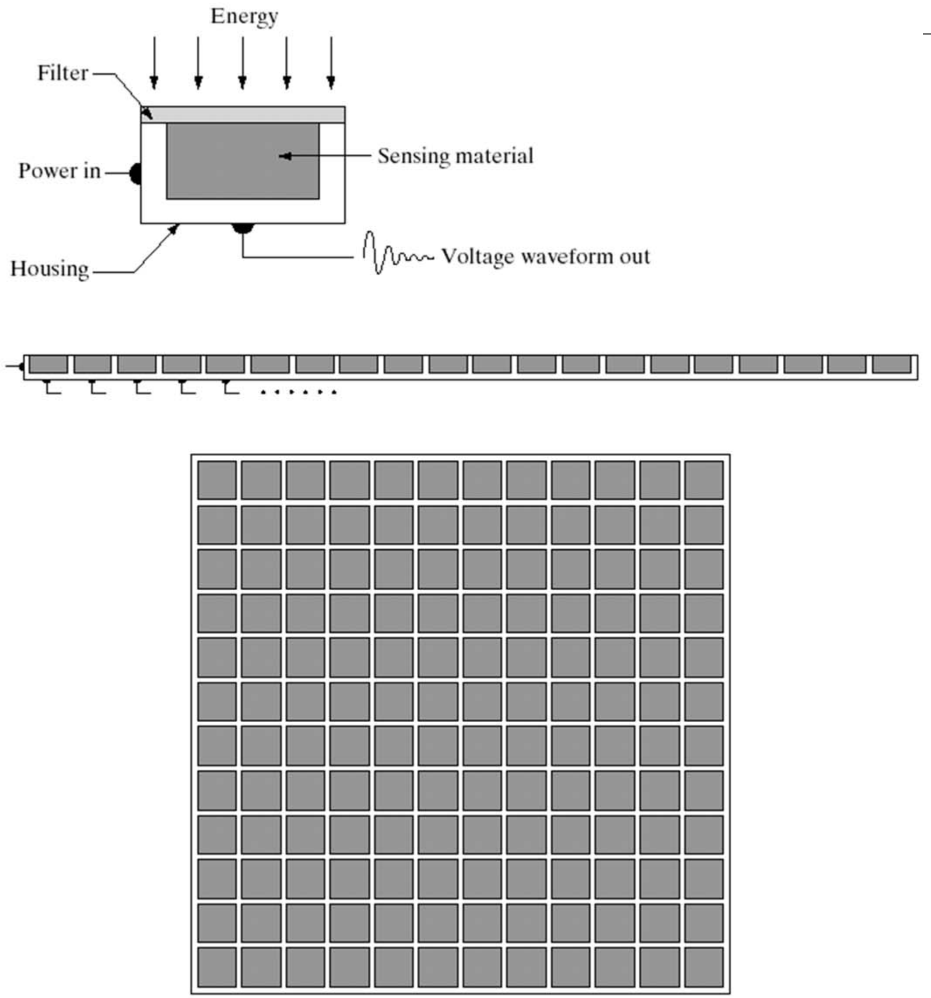


Image acquisition using a single sensor

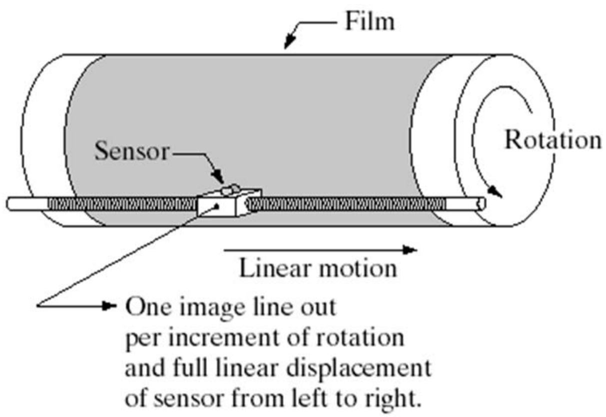
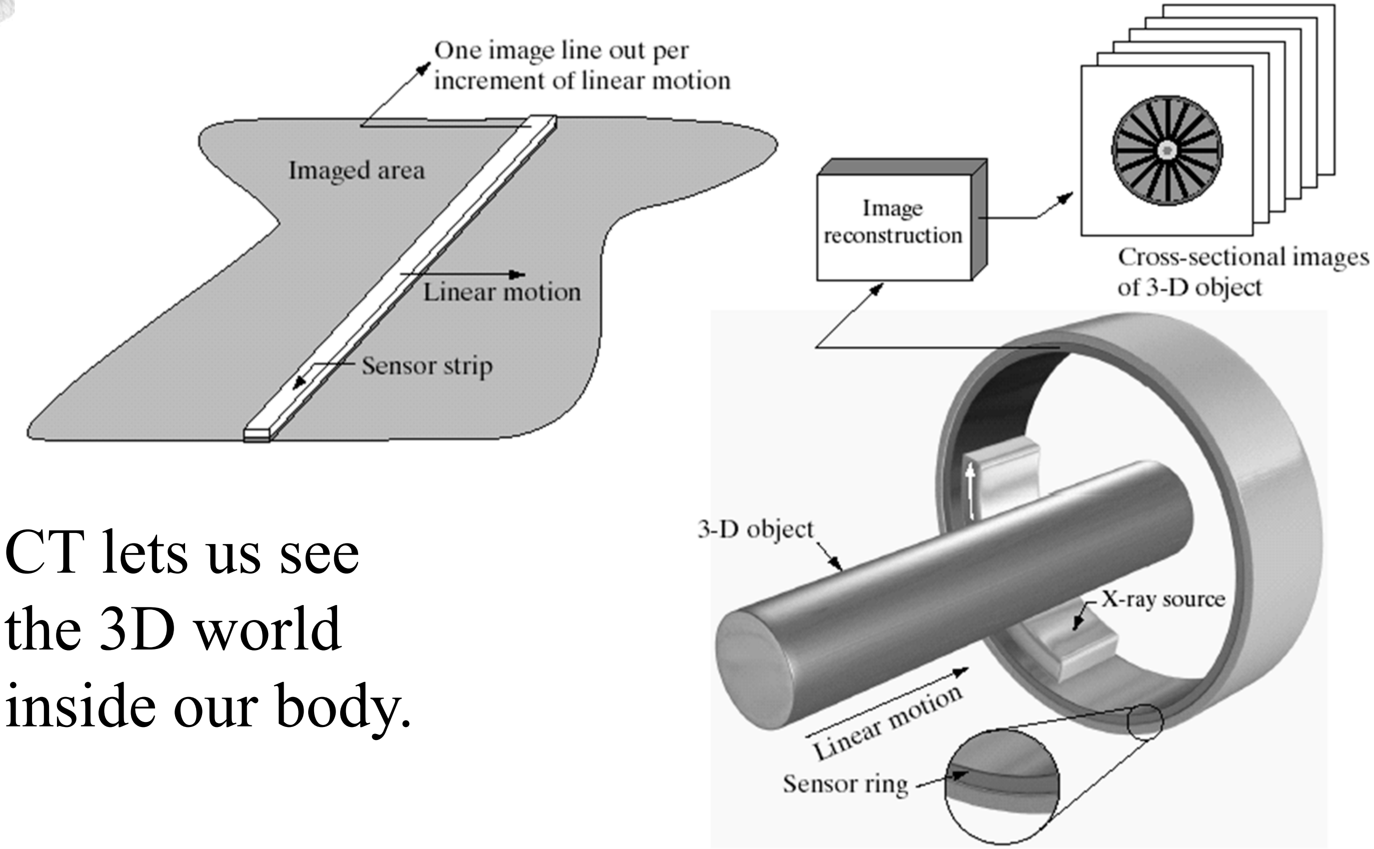


FIGURE 2.13 Combining a single sensor with motion to generate a 2-D image.

It once was the treasure of our lab. in Pittsburgh.

Image acquisition using a sensor stripes



CT lets us see
the 3D world
inside our body.

a b

FIGURE 2.14 (a) Image acquisition using a linear sensor strip. (b) Image acquisition using a circular sensor strip.

Image acquisition using a single arrays

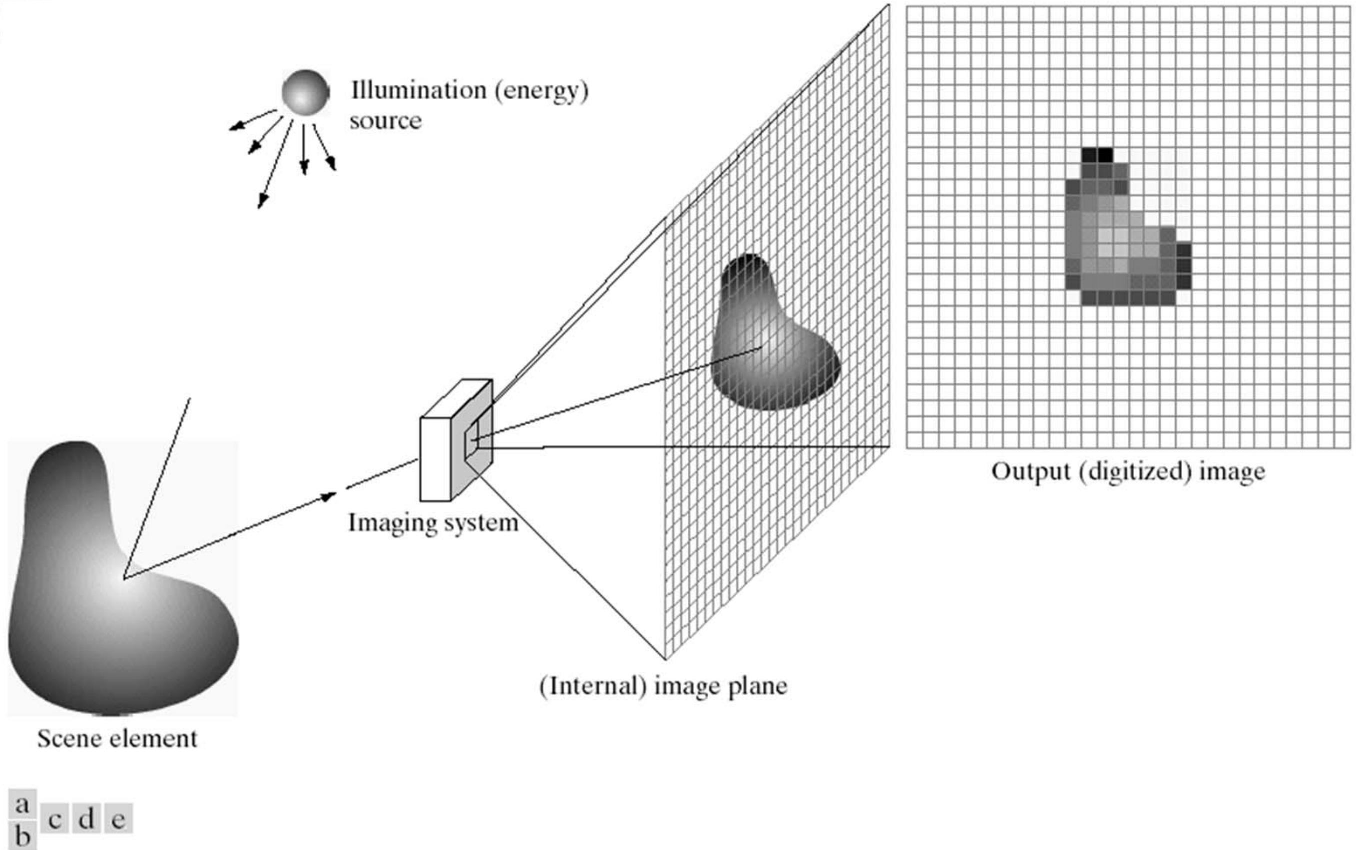


FIGURE 2.15 An example of the digital image acquisition process. (a) Energy ("illumination") source. (b) An element of a scene. (c) Imaging system. (d) Projection of the scene onto the image plane. (e) Digitized image.

A simple image formation model

- Image: a two-dimensional function *f(x,y)*

$$0 < f(x, y) < \infty. \quad (2.3\text{-}1)$$

$$f(x, y) = i(x, y)r(x, y) \quad (2.3\text{-}2)$$

i(x,y): illumination

r(x,y): reflectance

$$0 < i(x, y) < \infty \quad (2.3\text{-}3)$$

$$0 < r(x, y) < 1. \quad (2.3\text{-}4)$$

Gray level and Gray scale

- Gray level: the **intensity** of a monochrome image at (x_0,y_0)

$$\ell = f(x_0, y_0) \qquad (2.3\text{-}5)$$

$$L_{\min} \le \ell \le L_{\max} \qquad (2.3\text{-}6)$$

- Gray scale: the interval $[L_{min}, L_{max}]$
  => common practice [0,L-1]

  where

  $l=0$: black

  $l=L-1$: white

  intermediate values: shades of gray varying from ***black to white***

Generating a digital image: Image sampling and quantization

- Sampling : digitizing the coordinate values
- Quantization : digitizing the amplitude values

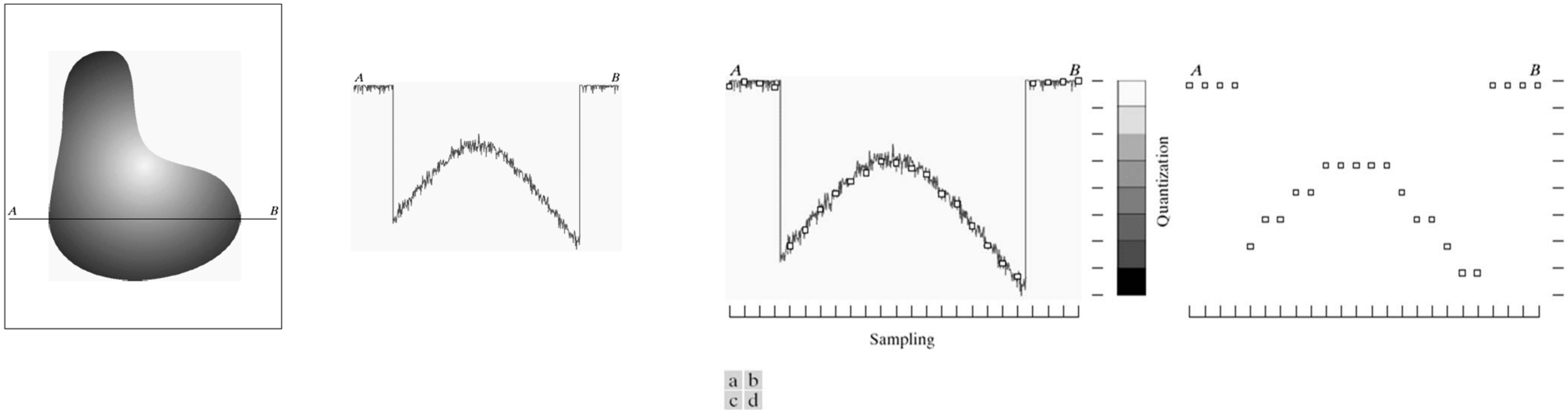
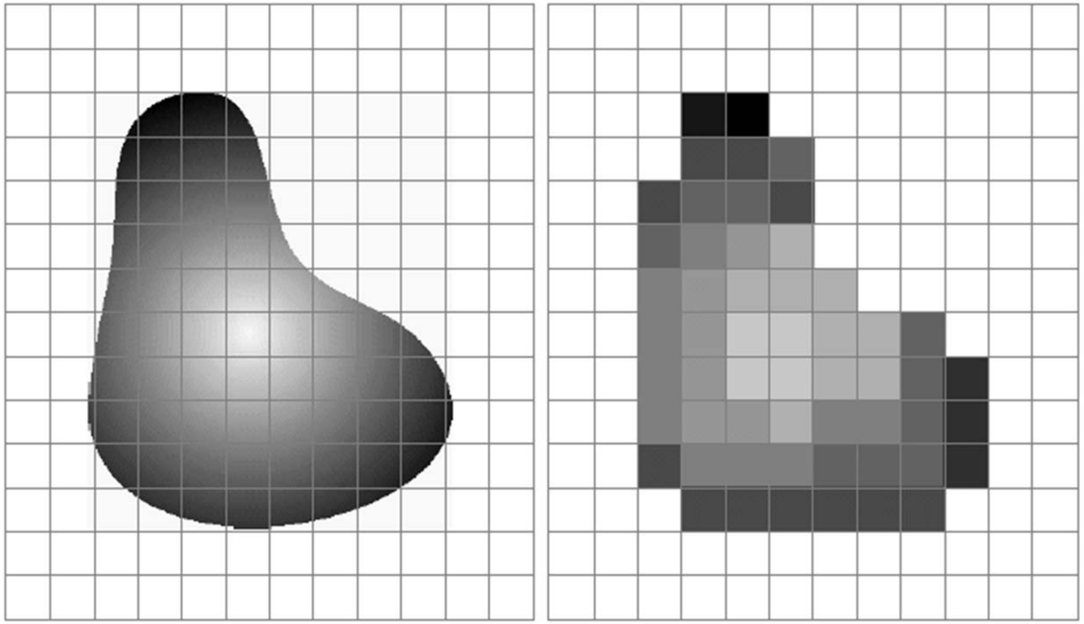


FIGURE 2.16 Generating a digital image. (a) Continuous image. (b) A scan line from A to B in the continuous image, used to illustrate the concepts of sampling and quantization. (c) Sampling and quantization. (d) Digital scan line.

Image sampling and quantization : Sensor array



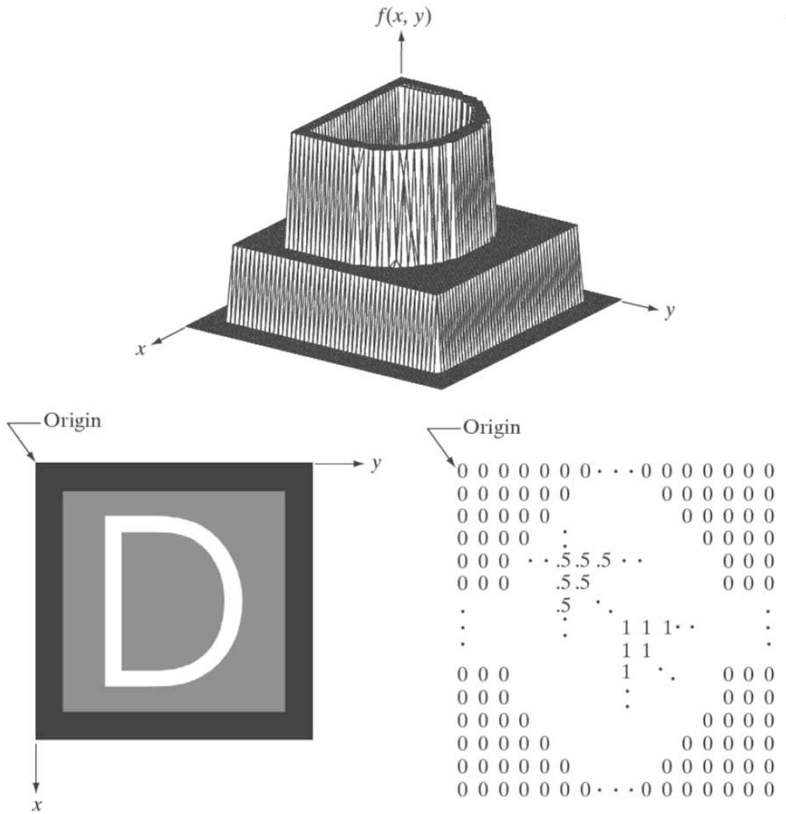
a b

FIGURE 2.17 (a) Continuos image projected onto a sensor array. (b) Result of image sampling and quantization.

The differences between the analog and digital world:

- resolution, manipulation, dynamic range,…… etc.

Representing digital images



a
b c

FIGURE 2.18 (a) Image plotted as a surface. (b) Image displayed as a visual intensity array. (c) Image shown as a 2-D numerical array (0, .5, and 1 represent black, gray, and white, respectively).

$$f(x, y) = \begin{bmatrix} f(0,0) & f(0,1) & \cdots & f(0, N-1) \\ f(1,0) & f(1,1) & \cdots & f(1, N-1) \\ \vdots & \vdots & & \vdots \\ f(M-1,0) & f(M-1,1) & \cdots & f(M-1, N-1) \end{bmatrix}. \quad (2.4\text{-}1)$$

$$\mathbf{A} = \begin{bmatrix} a_{0,0} & a_{0,1} & \cdots & a_{0,N-1} \\ a_{1,0} & a_{1,1} & \cdots & a_{1,N-1} \\ \vdots & \vdots & & \vdots \\ a_{M-1,0} & a_{M-1,1} & \cdots & a_{M-1,N-1} \end{bmatrix}. \quad (2.4\text{-}2)$$

Dynamic range and image storage size

- The number of gray level typically is an integer power of two

$$L = 2^k. \qquad (2.4\text{-}3)$$

- Dynamic range: the range of value spanned by the gray levels

Dynamic range

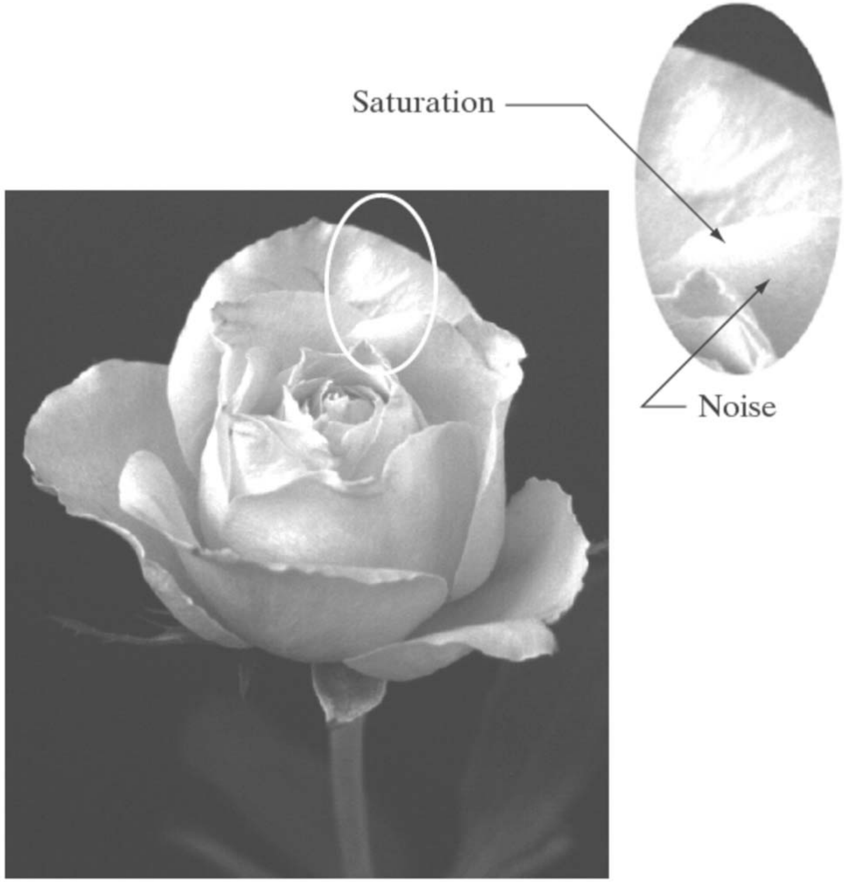


FIGURE 2.19 An image exhibiting saturation and noise. Saturation is the highest value beyond which all intensity levels are clipped (note how the entire saturated area has a high, *constant* intensity level). Noise in this case appears as a grainy texture pattern. Noise, especially in the darker regions of an image (e.g., the stem of the rose) masks the lowest detectable true intensity level.

Image storage size

- Bits required to store a digitized image is

$$b = M \times N \times k. \qquad (2.4\text{-}4)$$

- When M=N, the equation becomes

$$b = N^2k. \qquad (2.4\text{-}5)$$

TABLE 2.1
Number of storage bits for various values of N and k.

N/k	1 (L = 2)	2 (L = 4)	3 (L = 8)	4 (L = 16)	5 (L = 32)	6 (L = 64)	7 (L = 128)	8 (L = 256)
32	1,024	2,048	3,072	4,096	5,120	6,144	7,168	8,192
64	4,096	8,192	12,288	16,384	20,480	24,576	28,672	32,768
128	16,384	32,768	49,152	65,536	81,920	98,304	114,688	131,072
256	65,536	131,072	196,608	262,144	327,680	393,216	458,752	524,288
512	262,144	524,288	786,432	1,048,576	1,310,720	1,572,864	1,835,008	2,097,152
1024	1,048,576	2,097,152	3,145,728	4,194,304	5,242,880	6,291,456	7,340,032	8,388,608
2048	4,194,304	8,388,608	12,582,912	16,777,216	20,971,520	25,165,824	29,369,128	33,554,432
4096	16,777,216	33,554,432	50,331,648	67,108,864	83,886,080	100,663,296	117,440,512	134,217,728
8192	67,108,864	134,217,728	201,326,592	268,435,456	335,544,320	402,653,184	469,762,048	536,870,912

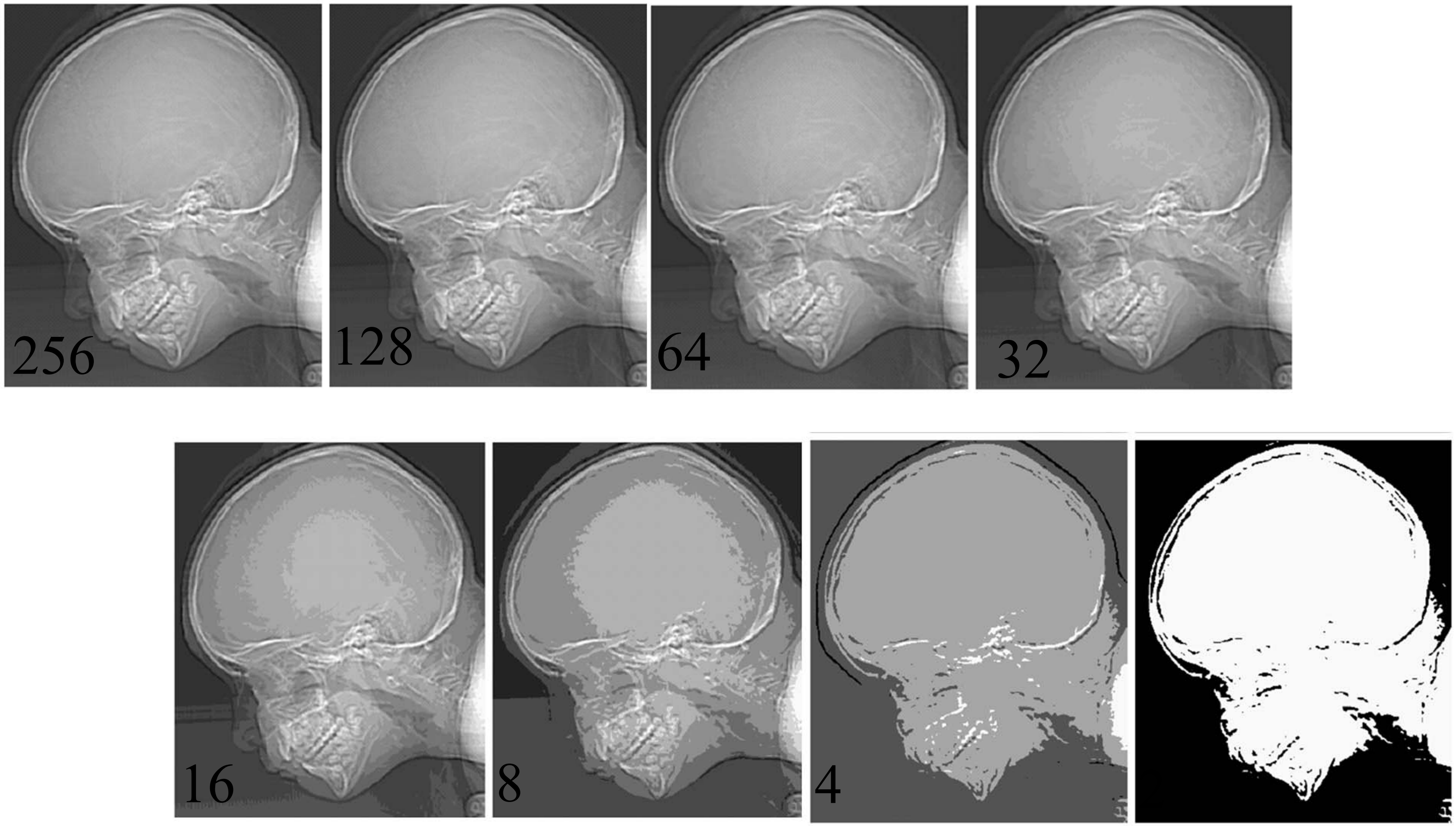
Spatial resolution

a	b
c	d

FIGURE 2.20 Typical effects of reducing spatial resolution. Images shown at: (a) 1250 dpi, (b) 300 dpi, (c) 150 dpi, and (d) 72 dpi. The thin black borders were added for clarity. They are not part of the data.

Intensity resolution

- 452*374 256-level



Level of Details

a b c

FIGURE 2.22 (a) Image with a low level of detail. (b) Image with a medium level of detail. (c) Image with a relatively large amount of detail. (Image (b) courtesy of the Massachusetts Institute of Technology.)

Iso-preference curves

- Each point in the *Nk*-plane represents an image having values of *N* and *k* equal to the coordinate of that point.
- Isopreference curve:
  points lying on an isopreference curve correspond to images of equal subject quality.

FIGURE 2.23 Representative isopreference curves for the three types of images in Fig. 2.22.

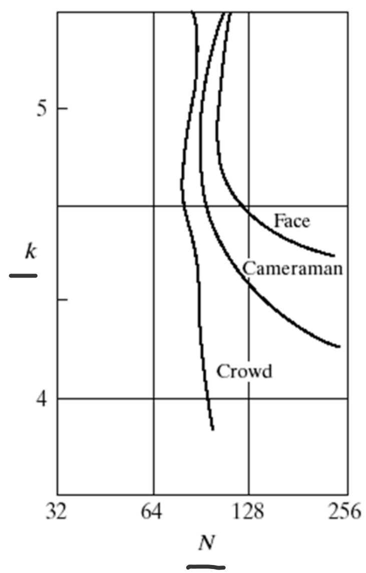


Image Interpolation

- Interpolation is the process of using known data to estimate values at unknown location.
- Suppose that an image of size 500 x 500 pixels has to be enlarged 1.5 times to 750 x 750 pixels.
- Shrink the 750x750 image to 500x500.
- The pixel spacing in the shrunken 750 x 750 grid will be less than the pixel spacing in the original image.
- To perform intensity-level assignment for any point in the overlay, we look for its closest pixel in the original image and assign the intensity of that pixel to the new pixel in the 750 x 750 grid that is called nearest neighbor interpolation.

Image Interpolation

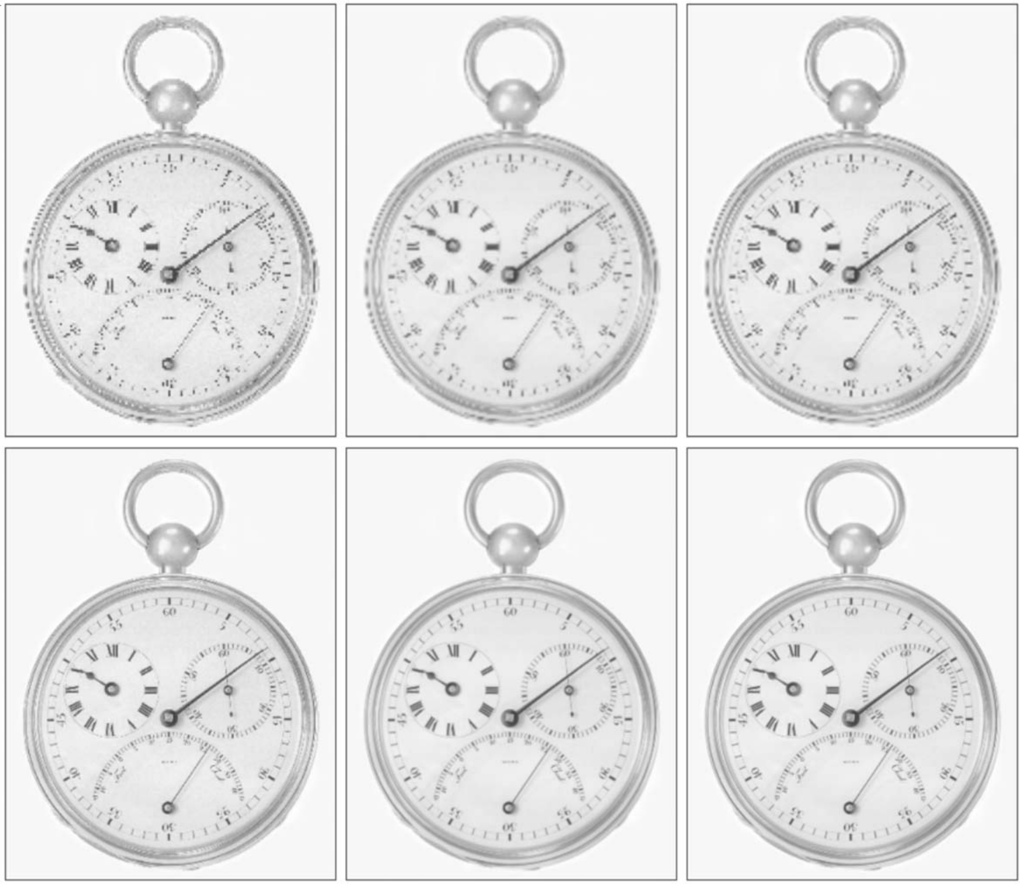
- Bilinear interpolation

$$v(x,y) = ax + by + cxy + d \quad (2.4\text{-}6)$$

- Bicubic interpolation

$$v(x,y) = \sum_{i=0}^{3}\sum_{j=0}^{3} a_{ij}x^i y^j \quad (2.4\text{-}7)$$

Image Interpolation



a	b	c
d	e	f

FIGURE 2.24 (a) Image reduced to 72 dpi and zoomed back to its original size (3692 × 2812 pixels) using nearest neighbor interpolation. This figure is the same as Fig. 2.20(d). (b) Image shrunk and zoomed using bilinear interpolation. (c) Same as (b) but using bicubic interpolation. (d)–(f) Same sequence, but shrinking down to 150 dpi instead of 72 dpi [Fig. 2.24(d) is the same as Fig. 2.20(c)]. Compare Figs. 2.24(e) and (f), especially the latter, with the original image in Fig. 2.20(a).

Some basic relationships between pixels

- Neighbors of a pixel
- Adjacency, connectivity, regions, and boundaries
- Distance measures
- Image operation on a pixel basis

Neighbors of a pixel

- $N_4(p)$: 4-neighbors of $p(x,y)$
  point set of the *horizontal* and *vertical* pixels

$$(x+1, y), (x-1, y), (x, y+1), (x, y-1)$$

- $N_D(p)$: *diagonal* neighbors of $p(x,y)$

$$(x+1, y+1), (x+1, y-1), (x-1, y+1), (x-1, y-1)$$

- $N_8(p) = N_4(p) \cup N_D(p)$: 8-neighbors of $p(x,y)$

Adjacency

(a) *4-adjacency*. Two pixels p and q with values from V are 4-adjacent if q is in the set $N_4(p)$.

(b) *8-adjacency*. Two pixels p and q with values from V are 8-adjacent if q is in the set $N_8(p)$.

(c) *m-adjacency* (mixed adjacency). Two pixels p and q with values from V are m-adjacent if

  (i) q is in $N_4(p)$, *or*

  (ii) q is in $N_D(p)$ *and* the set $N_4(p) \cap N_4(q)$ has no pixels whose values are from V.

a b c

FIGURE 2.25 (a) An arrangement of pixels. (b) Pixels that are 8-adjacent (adjacency is shown by dashed lines; note the ambiguity). (c) m-adjacency.

Connectivity

- *A Digital path* (or *curve*) from pixel p with coordinates (x,y) to pixel q with coordinates (s,t) is a sequence of distinct pixels with coordinates

  $$(x_0, y_0), (x_1, y_1), \ldots, (x_n, y_n)$$

  where $(x_0,y_0)=(x,y)$, $(x_n,y_n)=(s,t)$, and pixels (x_i,y_i) and (x_{i-1},y_{i-1}) are adjacent for $1 \leq i \leq n$.
- n: the length of the path
- If $(x_0,y_0)= (x_n,y_n)$, the path is a *closed path*

Connectivity (cont.)

- *S:* a sub set of pixels in an image
- *Connected*: p and q are said to be connected if there exists a path between them consisting entirely of pixels in S.
- *Connected component*: for any pixel p in S, the set of pixels that are connected to it in S is called a connected component of S.
- *Connected set*: If it only has one connected component, then set S is called a connected set.

Regions

- *R:* a subset of pixels in an image
- *Region*: *R* is a region of the image if *R* is a connected set.
- *Adjacent*: union of two regions forms a connected set.
- *Disjoint*: Region that are not adjacent

$$\begin{matrix} \left.\begin{matrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{matrix}\right\} R_i \\ \left.\begin{matrix} 0 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{matrix}\right\} R_j \end{matrix}$$

d

FIGURE 2.25 ((d) Two regions that are adjacent if 8-adjecency is used.

Boundaries

- *Boundary*: (also called *border* or *contour)* The boundary of a region R is the set of pixels in the region that have one or more neighbors that are not in R. (be referred to as the *inner border*)

- *Outer border*: corresponding border in the background.

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

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

e f **FIGURE 2.25** (e) The circled point is part of the boundary of the 1-valued pixels only if 8-adjacency between the region and background is used. (f) The inner boundary of the 1-valued region does not form a closed path, but its outer boundary does.

Distance measures

- Distance function (metric)
- Euclidean distance
- D_4 distance (city-block distance)
- D_8 distance (chessboard distance)
- D_m distance

Distance function and Euclidean distance

- Distance function (metric)

For pixels p, q, and z, with coordinates (x, y), (s, t), and (v, w), respectively, D is a *distance function* or *metric* if

(a) $D(p, q) \geq 0 \quad \big(D(p, q) = 0 \quad \text{iff} \quad p = q\big)$,
(b) $D(p, q) = D(q, p)$, and
(c) $D(p, z) \leq D(p, q) + D(q, z)$.

- Euclidean distance

The *Euclidean distance* between p and q is defined as

$$D_e(p, q) = \left[(x - s)^2 + (y - t)^2\right]^{\frac{1}{2}}. \qquad (2.5\text{-}1)$$

D_4 distance

- D_4 distance (city-block distance)

The D_4 *distance* (also called *city-block distance*) between p and q is defined as

$$D_4(p, q) = |x - s| + |y - t|. \qquad (2.5\text{-}2)$$

In this case, the pixels having a D_4 distance from (x, y) less than or equal to some value r form a diamond centered at (x, y). For example, the pixels with D_4 distance ≤ 2 from (x, y) (the center point) form the following contours of constant distance:

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

The pixels with $D_4 = 1$ are the 4-neighbors of (x, y).

D_8 distance

- D_8 distance (chessboard distance)

The D_8 *distance* (also called *chessboard distance*) between p and q is defined as

$$D_8(p, q) = \max(|x - s|, |y - t|). \tag{2.5-3}$$

In this case, the pixels with D_8 distance from (x, y) less than or equal to some value r form a square centered at (x, y). For example, the pixels with D_8 distance ≤ 2 from (x, y) (the center point) form the following contours of constant distance:

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

The pixels with $D_8 = 1$ are the 8-neighbors of (x, y).

D_m distance

- D_m distance : If we elect to consider the *m*-adjacency, the D_m *distance* between two points is defined as the shortest *m*-path between the points.

P_3 P_4

P_1 P_2

P

An Introduction to the Mathematical Tools Used in Digital Image Processing

- Array versus Matrix Operation
- Linear versus Nonlinear Operations
- Arithmetic Operations
- Set and Logical Operations
- Spatial Operations
- Vector and Matrix Operations
- Image Transforms
- Probabilistic Methods

Array versus Matrix Operation

- Consider the follow 2x2 image:

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \qquad \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}$$

- The array product of these two image is

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}\begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} = \begin{bmatrix} a_{11}b_{11} & a_{12}b_{12} \\ a_{21}b_{21} & a_{22}b_{22} \end{bmatrix}$$

- On the other hand, the matrix product is given by

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}\begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} = \begin{bmatrix} a_{11}b_{11} + a_{12}b_{21} & a_{11}b_{12} + a_{12}b_{22} \\ a_{21}b_{11} + a_{22}b_{21} & a_{21}b_{12} + a_{22}b_{22} \end{bmatrix}$$

Linear versus Nonlinear Operation

- Consider a general operator, H, that produces an output image, $g(x, y)$, for a given input image, $f(x, y)$:

$$H\left[f\left(x,y\right)\right]=g\left(x,y\right)$$

- H is said to be a linear operator if

$$\begin{aligned}H\left[a_i f_i\left(x,y\right)+a_j f_j\left(x,y\right)\right]&=a_i H\left[f_i\left(x,y\right)\right]+a_j H\left[f_j\left(x,y\right)\right]\\&=a_i g_i\left(x,y\right)+a_j g_j\left(x,y\right)\end{aligned}$$

- As a simple example ,suppose that H is the sum operator, Σ.

$$\begin{aligned}\sum\left[a_i f_i\left(x,y\right)+a_j f_j\left(x,y\right)\right]&=\sum a_i f_i\left(x,y\right)+\sum a_j f_j\left(x,y\right)\\&=a_i\sum f_i\left(x,y\right)+a_j\sum f_j\left(x,y\right)\\&=a_i g_i\left(x,y\right)+a_j g_j\left(x,y\right)\end{aligned}$$

Linear versus Nonlinear Operation

- On the other hand, consider the max operation.
- Consider the following two images

$$f_1 = \begin{bmatrix} 0 & 2 \\ 2 & 3 \end{bmatrix} \qquad f_2 = \begin{bmatrix} 6 & 5 \\ 4 & 7 \end{bmatrix}$$

  and suppose that we let $a_1 = 1$ and $a_2 = -1$.

- To test for linearity, we start with the left side

$$\max\left\{(1)\begin{bmatrix} 0 & 2 \\ 2 & 3 \end{bmatrix} + (-1)\begin{bmatrix} 6 & 5 \\ 4 & 7 \end{bmatrix}\right\} = \max\left\{\begin{bmatrix} -6 & -3 \\ -2 & -4 \end{bmatrix}\right\}$$
$$= -2$$

- Working next with the right side, we obtain

$$(1)\max\left\{\begin{bmatrix} 0 & 2 \\ 2 & 3 \end{bmatrix}\right\} + (-1)\max\left\{\begin{bmatrix} 6 & 5 \\ 4 & 7 \end{bmatrix}\right\} = 3 + (-1)7$$
$$= -4$$

Arithmetic Operations

- Arithmetic operations between images are array operations which means that arithmetic operations are carried out between corresponding pixel pairs.

- The four arithmetic operations are denoted as

$$s(x,y) = f(x,y) + g(x,y)$$
$$d(x,y) = f(x,y) - g(x,y)$$
$$p(x,y) = f(x,y) \times g(x,y)$$
$$v(x,y) = f(x,y) \div g(x,y)$$

Addition (averaging) of noisy images for noise reduction

- Let $g(x, y)$ denote a corrupted image formed by the addition of noise, $\eta(x, y)$, to a noiseless image $f(x, y)$.

$$g(x,y) = f(x,y) + \eta(x,y) \quad (2.6\text{-}4)$$

- If the noise satisfies the constraints just stated, it can be shown that if an image $\bar{g}(x,y)$ is formed by averaging K different noisy images,

$$\bar{g}(x,y) = \frac{1}{K}\sum_{1}^{K} g_i(x,y) \quad (2\text{-}6.5)$$

then if follows that

expected value of $\bar{g}$

$$E\{\bar{g}(x,y)\} = f(x,y) \quad (2\text{-}6.6)$$

variance of $\bar{g}$

variance of η

$$\sigma^2_{\bar{g}(x,y)} = \frac{1}{K}\sigma^2_{\eta(x,y)} \quad (2\text{-}6.7)$$

Addition (averaging) of noisy images for noise reduction

- The standard deviation at any point in the average image is

$$\sigma_{\bar{g}(x,y)} = \frac{1}{\sqrt{K}} \sigma_{\eta(x,y)} \quad (2\text{-}6.8)$$

- As K increase, Eqs. (2.6-7) and (2.6-8) indicate that the variability of the pixel values at each location (x, y) decrease.

- The images must be registered in order to avoid the introduction of blurring and other artifacts in the output image.

Addition (averaging) of noisy images for noise reduction

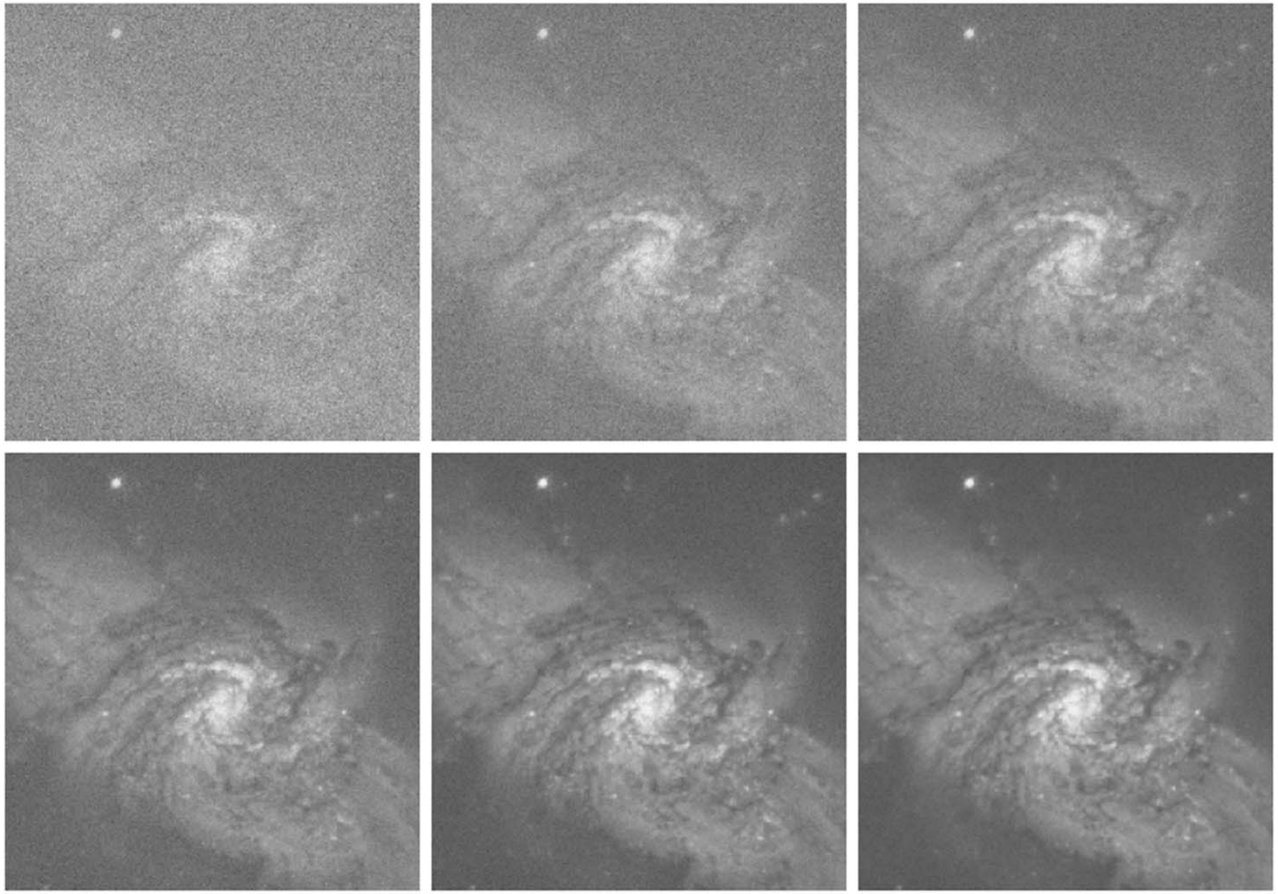
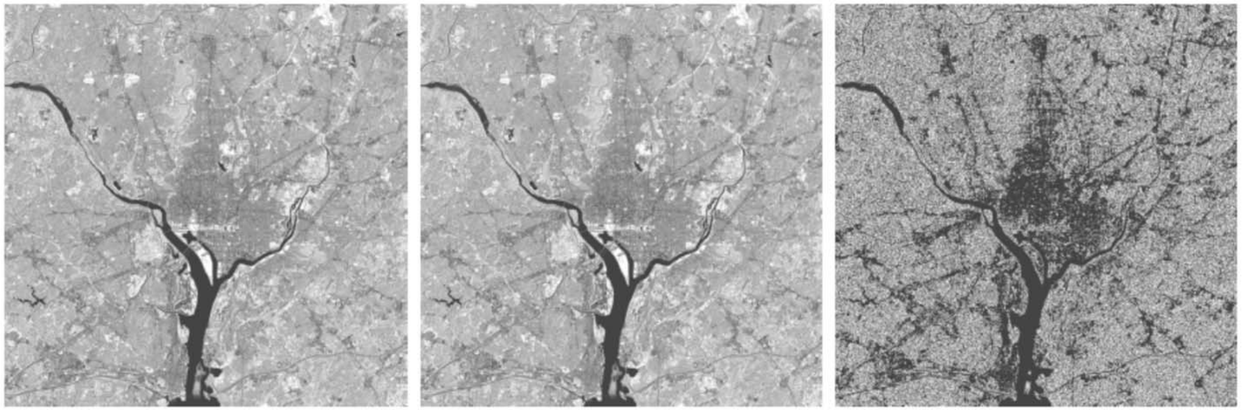


FIGURE 2.26 (a) Image of Galaxy Pair NGC 3314 corrupted by additive Gaussian noise. (b)–(f) Results of averaging 5, 10, 20, 50, and 100 noisy images, respectively. (Original image courtesy of NASA.)

Image subtraction for enhancing differences

- Consider image differences of the form

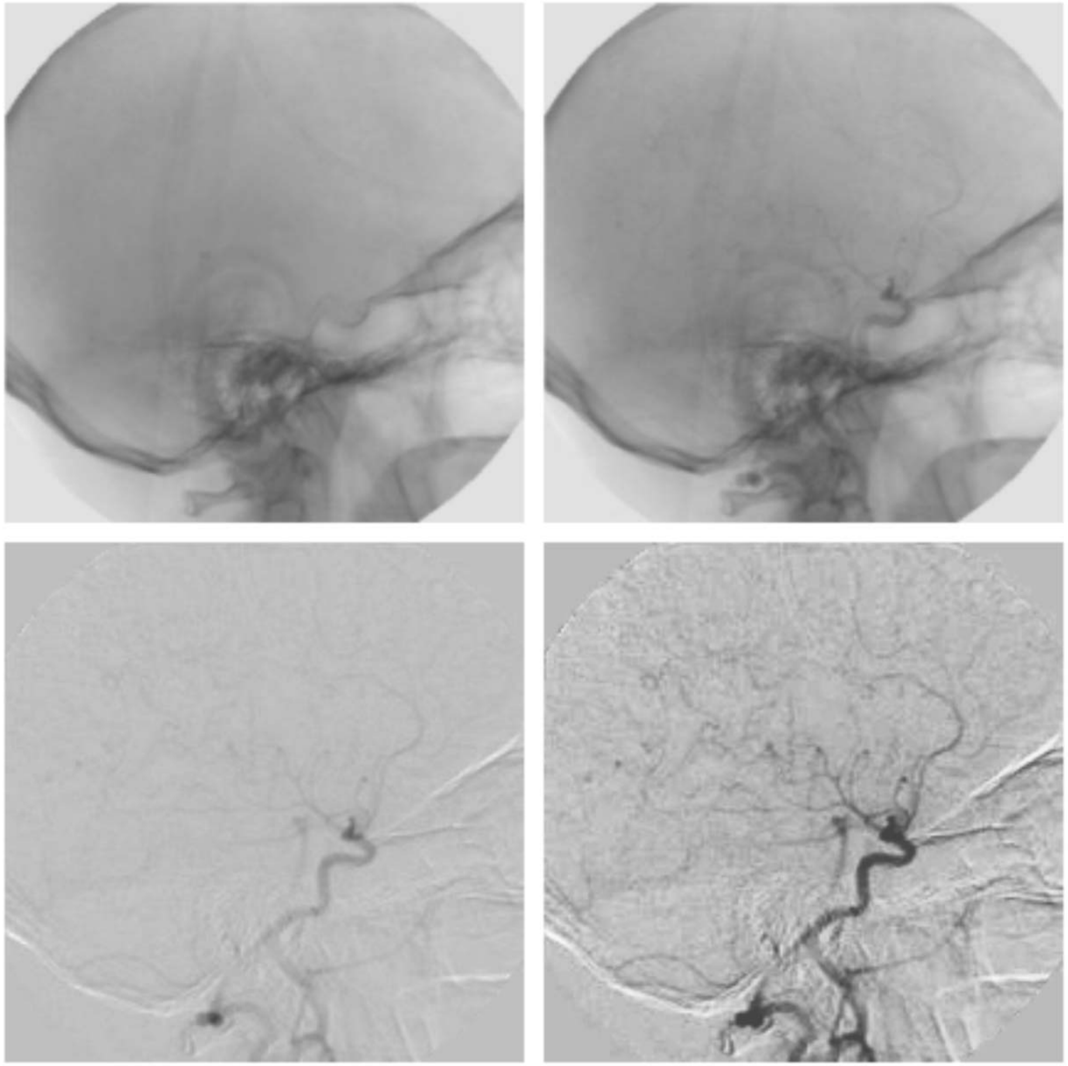
$$g(x,y) = f(x,y) - h(x,y) \quad (2\text{-}6.9)$$



a b c

FIGURE 2.27 (a) Infrared image of the Washington, D.C. area. (b) Image obtained by setting to zero the least significant bit of every pixel in (a). (c) Difference of the two images, scaled to the range [0, 255] for clarity.

Image subtraction for enhancing differences



a b
c d

FIGURE 2.28 Digital subtraction angiography. (a) Mask image. (b) A live image. (c) Difference between (a) and (b). (d) Enhanced difference image. (Figures (a) and (b) courtesy of The Image Sciences Institute, University Medical Center, Utrecht, The Netherlands.)

Using image multiplication and division for shading correction

a b c

FIGURE 2.29 Shading correction. (a) Shaded SEM image of a tungsten filament and support, magnified approximately 130 times. (b) The shading pattern. (c) Product of (a) by the reciprocal of (b). (Original image courtesy of Mr. Michael Shaffer, Department of Geological Sciences, University of Oregon, Eugene.)

a b c

FIGURE 2.30 (a) Digital dental X-ray image. (b) ROI mask for isolating teeth with fillings (white corresponds to 1 and black corresponds to 0). (c) Product of (a) and (b).

Full range of an arithmetic operation

- The values in the difference of two 8-bit image can range from a minimum of -255 to a maximum of 255.
- Given an image f, an approach that guarantees that the full range of an arithmetic operation between images is "captured" into a fixed number of bits is as follow
  - First, we perform the operation

$$f_m = f - \min(f) \quad (2\text{-}6.10)$$

  - Then we perform the operation

$$f_s = K\left[f_m / \max(f_m) \right] \quad (2\text{-}6.11)$$

which create a scaled image, f_s, whose values are in the range [0, K].

Basic set operations

- If $a = (a_1, a_2)$ is an element of A

$$a \in A \quad (2\text{-}6.12)$$

- If a is not an element of A

$$a \notin A \quad (2\text{-}6.13)$$

- Null or empty set: $\varnothing$
- A set is specified by the contents of two braces: $\{\cdot\}$

Basic set operations

- If every element of a set A is also an element of a set B, then a is said to be a *subset* of B

$$A \subseteq B \quad (2\text{-}6.14)$$

- The *union* of two sets A and B

$$C = A \bigcup B \quad (2\text{-}6.15)$$

- *Intersection* of two sets A and B

$$D = A \bigcap B \quad (2\text{-}6.16)$$

- Two sets A and B are said to be disjoint or mutually exclusive if they have no common elements

$$A \bigcap B = \varnothing \quad (2\text{-}6.17)$$

Basic set operations

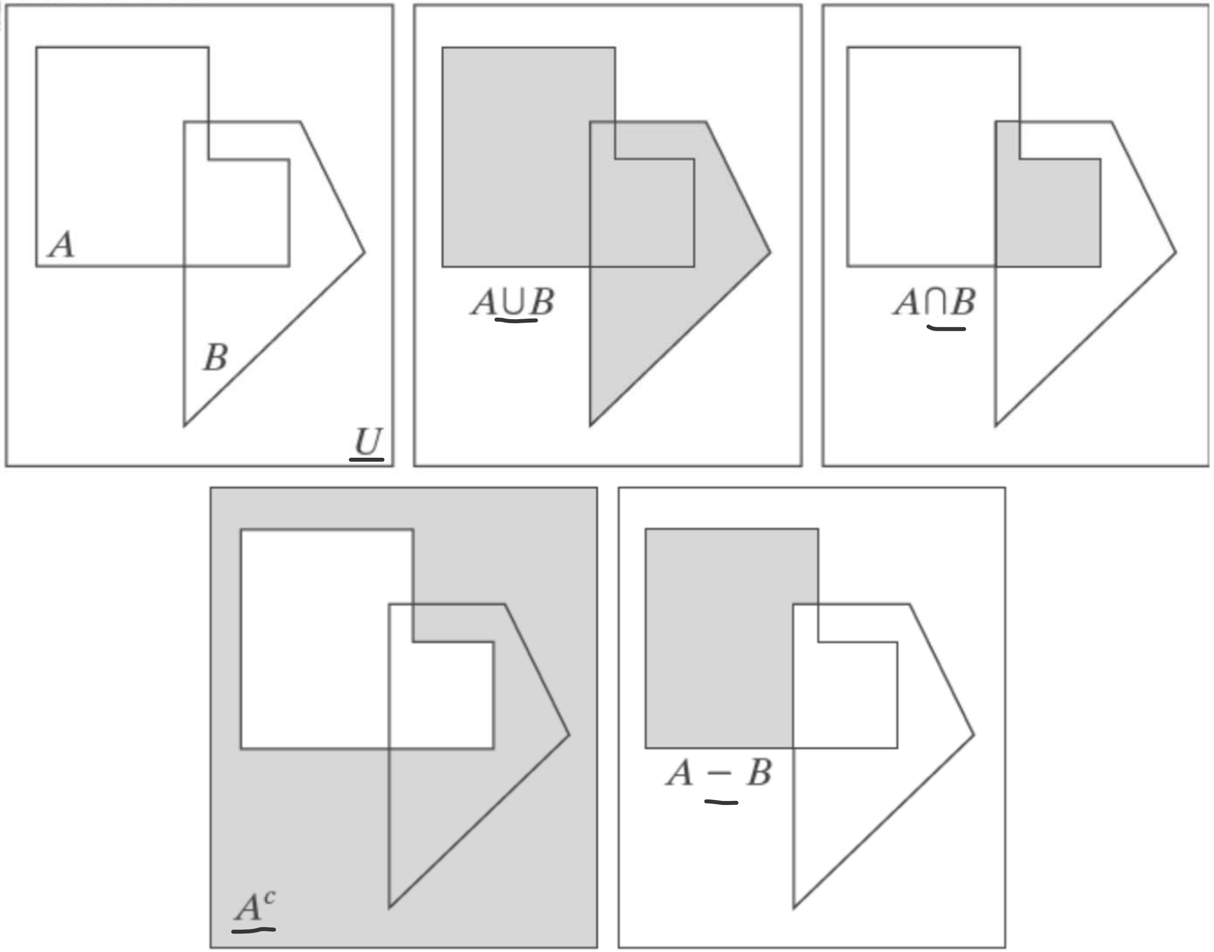
- The set *universe*, U, is the set of all elements in a given application
- The *complement* of a set A is the set of elements that are not in A

$$A^c = \{w \mid w \notin A\} \quad (2\text{-}6.18)$$

- The difference of two sets A and B

$$A - B = \{w \mid w \in A, w \notin B\} = A \cap B^c \quad (2\text{-}6.19)$$

Basic set operations



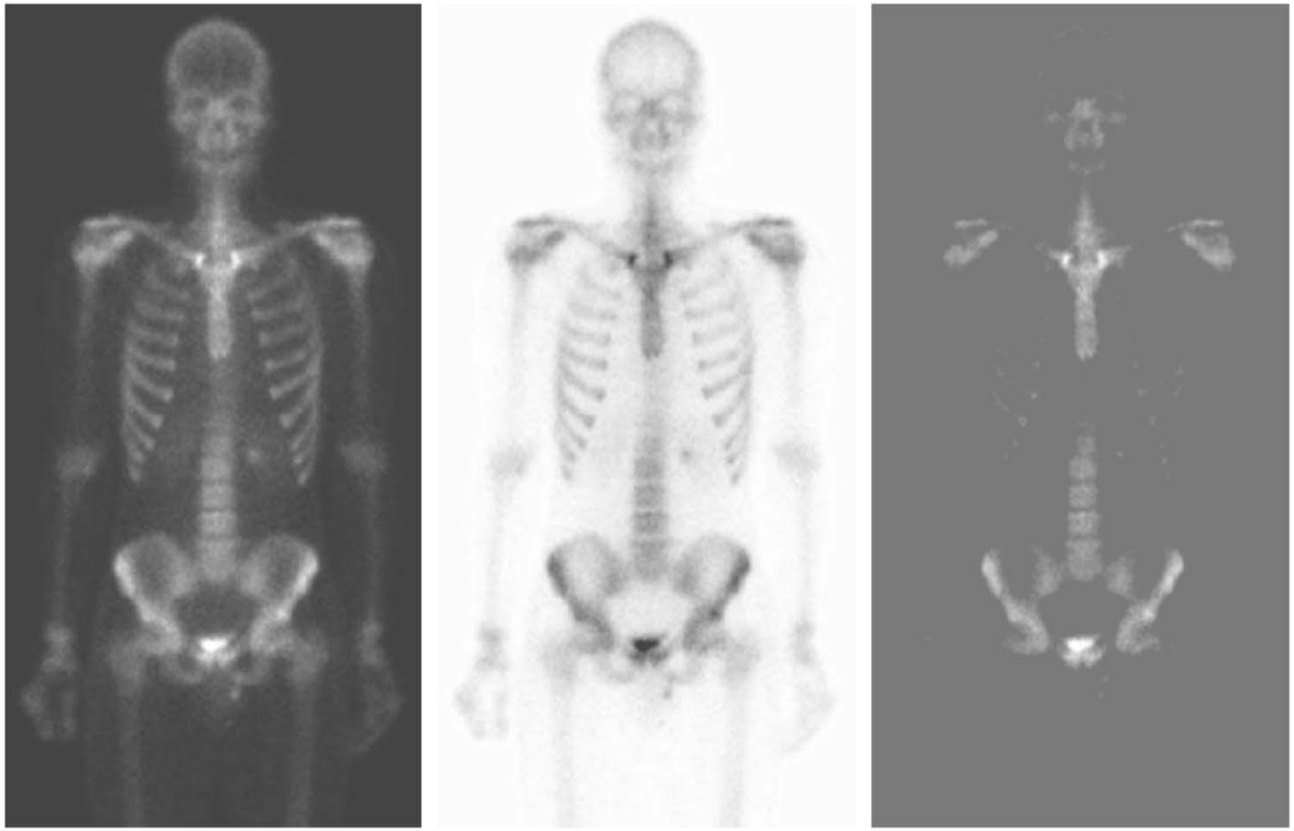
a b c
d e

FIGURE 2.31
(a) Two sets of coordinates, A and B, in 2-D space. (b) The union of A and B. (c) The intersection of A and B. (d) The complement of A. (e) The difference between A and B. In (b)–(e) the shaded areas represent the member of the set operation indicated.

Set operations involving image intensities

$$An=Ac=\{(x,y,255-z) \mid (x,y,z) \in A\}$$

$$A \cup B = \left\{ \max_z (a,b) \mid a \in A, b \in B \right\}$$



a b c

FIGURE 2.32 Set operations involving gray-scale images. (a) Original image. (b) Image negative obtained using set complementation. (c) The union of (a) and a constant image. (Original image courtesy of G.E. Medical Systems.)

Logical operation

- *Logical operator*: NOT, AND, OR, and XOR

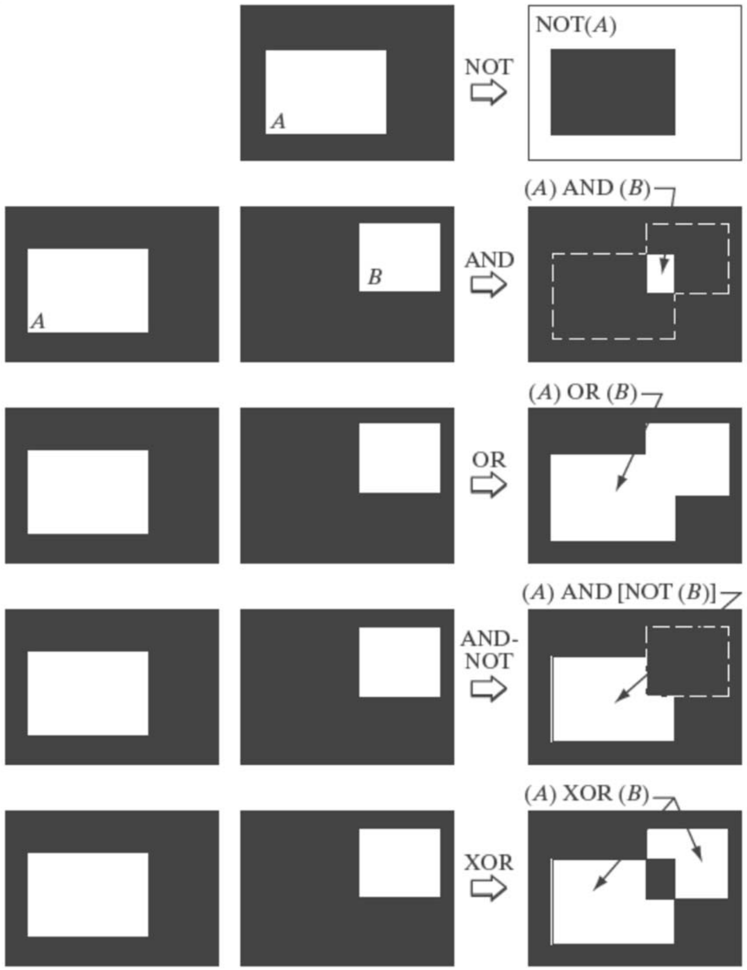


FIGURE 2.33 Illustration of logical operations involving foreground (white) pixels. Black represents binary 0s and white binary 1s. The dashed lines are shown for reference only. They are not part of the result.

Fuzzy sets

- Using crisp sets, let *U* denote the set of all people and let A be a subset of *U*, which we call the *set of young people*.
- Suppose that we define as young any person of age 20 or younger. A person whose age is 20 years and 1 sec would not be a member of the set of young people.
- The theory of fuzzy sets implements this concept by utilizing <u>membership function that are gradual between the limit values of 1 (definitely young) to 0 (definitely not young).</u>
- Using fuzzy sets, we can make a statement such as a person <u>being 50% young</u> (in the middle of the transition between young and not young).

Single-pixel operations

- This type of process may be expressed as a transformation function, T, of the form:

$$s = T(z) \quad (2\text{-}6.20)$$

where z is the intensity of the original image and s is the (mapped) intensity of the corresponding pixel in the processed image.

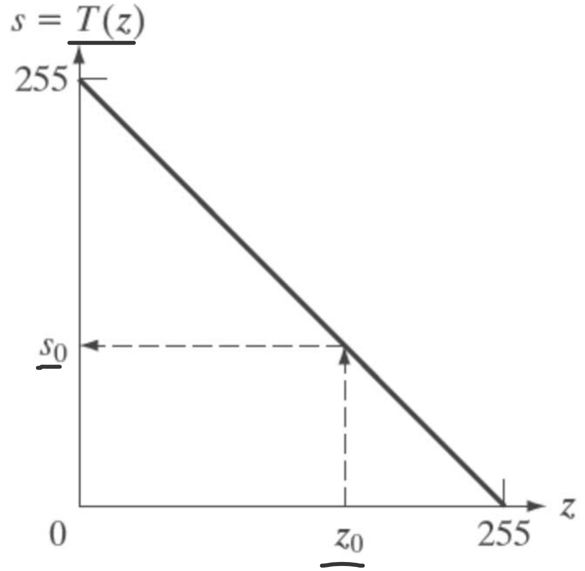
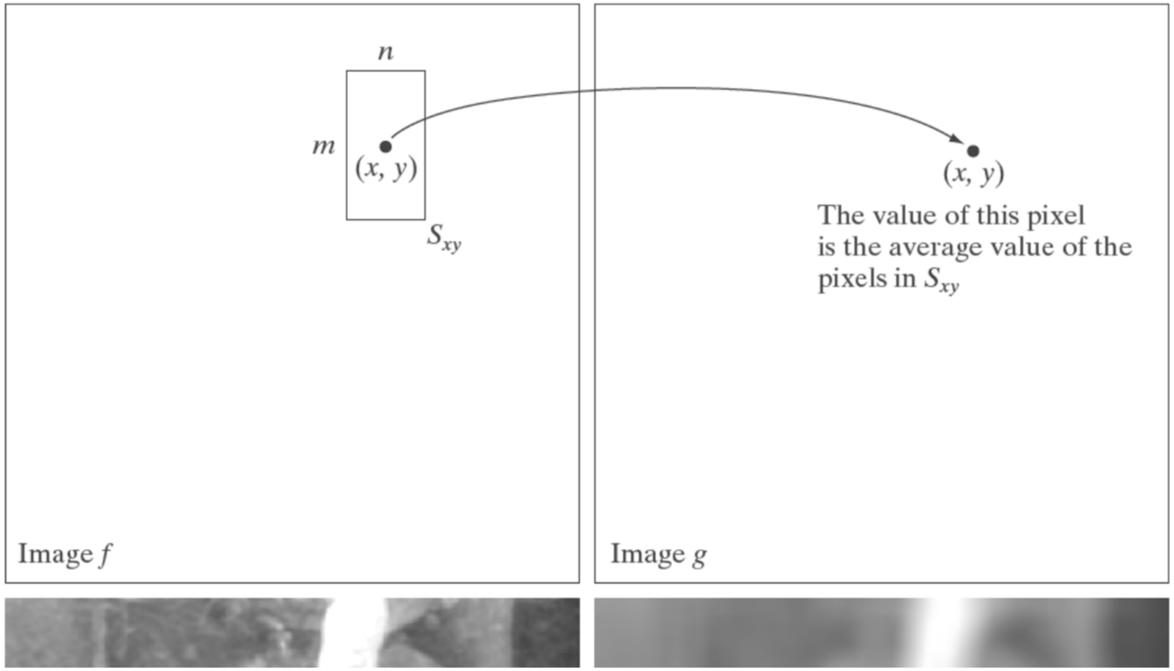


FIGURE 2.34 Intensity transformation function used to obtain the negative of an 8-bit image. The dashed arrows show transformation of an arbitrary input intensity value z_0 into its corresponding output value s_0.

Neighborhood operation

- Figures 2.35(a) and (b) illustrate the process. We can express this operation in equation form as

$$g(x,y) = \frac{1}{mn} \sum_{(r,c)\in S_{xy}} f(r,c) \quad (2\text{-}6.21)$$



a b
c d

FIGURE 2.35 Local averaging using neighborhood processing. The procedure is illustrated in (a) and (b) for a rectangular neighborhood. (c) The aortic angiogram discussed in Section 1.3.2. (d) The result of using Eq. (2.6-21) with $m = n = 41$. The images are of size 790 × 686 pixels.

Geometric spatial transformations and image registration

- In term of digital image processing, a geometric transformation consists two basic operations:

  1) A spatial transformation of coordinates
  2) Intensity interpolation that assigns intensity value to the spatially transformed pixels.

- The transformation of coordinates may be expressed as

$$(x, y) = T\{(v, w)\} \quad (2\text{-}6.22)$$

where (v, w) are pixel coordinates in the original image and (x, y) are the corresponding pixel coordinates in the transformed image.

Geometric spatial transformations and image registration

- One of the most commonly used spatial coordinate transformations is the affine transform

$$[x\ y\ 1] = [v\ w\ 1]\mathbf{T} = [v\ w\ 1]\begin{bmatrix} t_{11} & t_{12} & 0 \\ t_{21} & t_{22} & 0 \\ t_{31} & t_{32} & 1 \end{bmatrix} \quad (2\text{-}6.23)$$

- This transformation can scale, rotate, translate, or sheer a set of coordinate points, depending on the value chosen for the elements of matrix $\mathbf{T}$.

Geometric spatial transformations and image registration

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

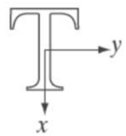
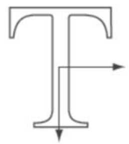
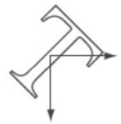
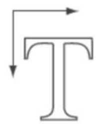
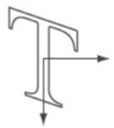
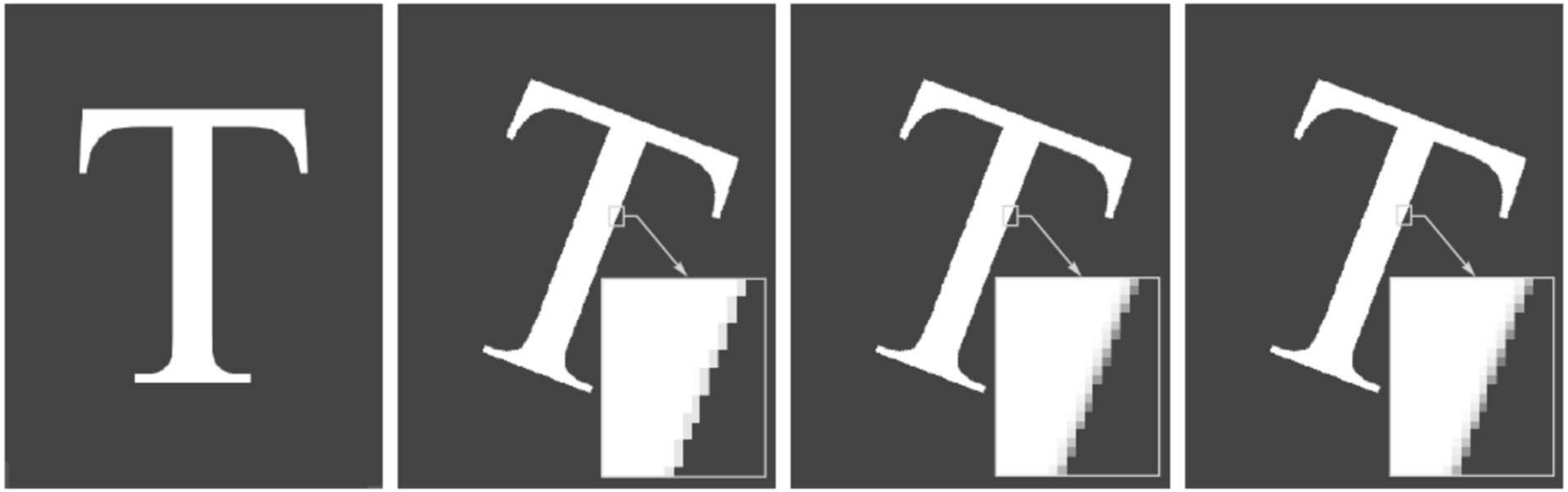
Transformation Name	Affine Matrix, T	Coordinate Equations	Example
Identity	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = v$ $y = w$	
Scaling	$\begin{bmatrix} c_x & 0 & 0 \\ 0 & c_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = c_x v$ $y = c_y w$	
Rotation	$\begin{bmatrix} \cos\theta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

Image rotation and intensity interpolation



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.

Image registration

- The problem is to estimate the transformation function and then use it to register the two images.
- To clarify terminology, the input image is the image that we wish to transform, and what we call the *reference* image is the image against which we want to register the input.
- One of the principal approaches for solving the problem just discussed is to use *tie points* (also called *control points*).

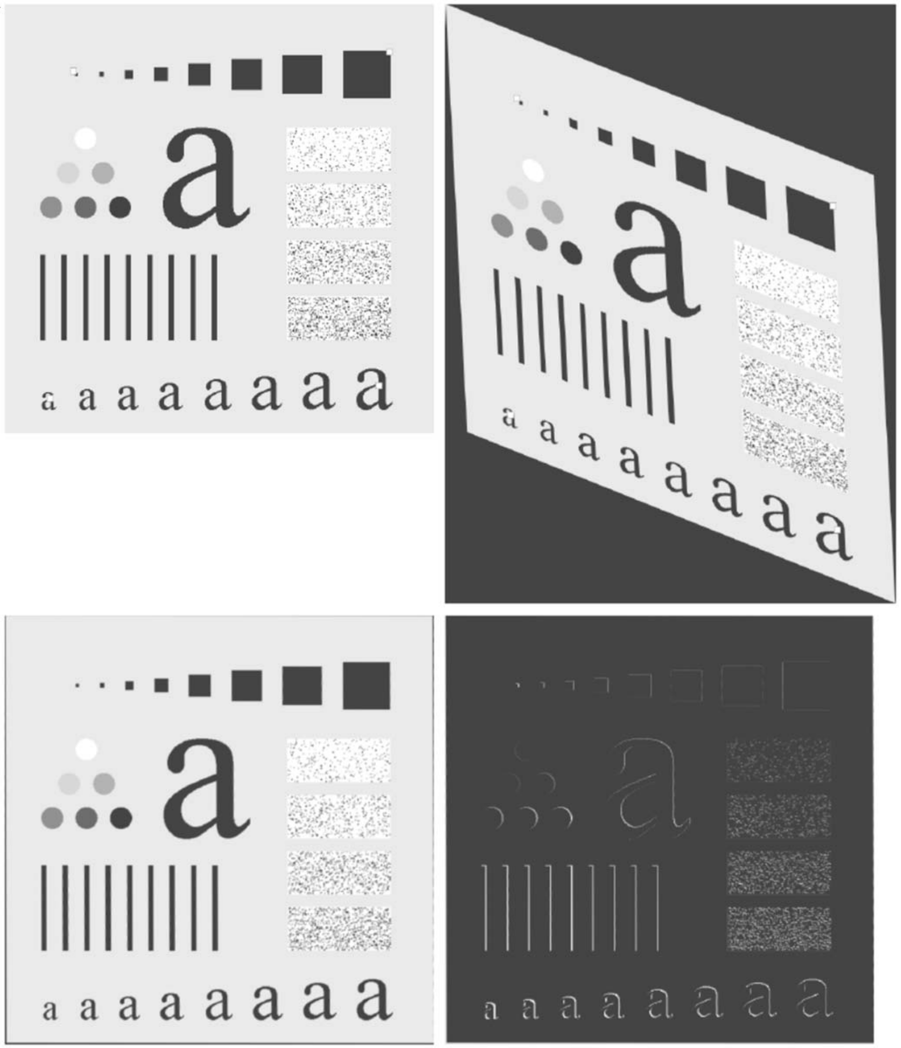
Image registration

- The problem of estimating the transformation function is one of modeling.
- For example, suppose that we have a set of four tie points each in an input and a reference image.
- A simple model based on a bilinear approximation is given by

$$x = c_1 v + c_2 w + c_3 vw + c_4 \quad (2.6\text{-}24)$$
$$y = c_5 v + c_6 w + c_7 vw + c_8 \quad (2.6\text{-}25)$$

- If we have four pairs of corresponding tie points in both image, we can write eight equations using Eqs. (2.6-24) and Eqs. (2.6-25) and use them to solve for the eight unknown coefficients, $c_1, c_2, \ldots, c_8$.

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.

Vector and Matrix Operations

- We see that *each* pixel of an RGB image has three components which can be organized in the form of a *column vector*

$$\mathbf{z} = \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} \begin{array}{l} \text{R} \\ \text{G} \\ \text{B} \end{array} \qquad (2.6\text{-}26)$$

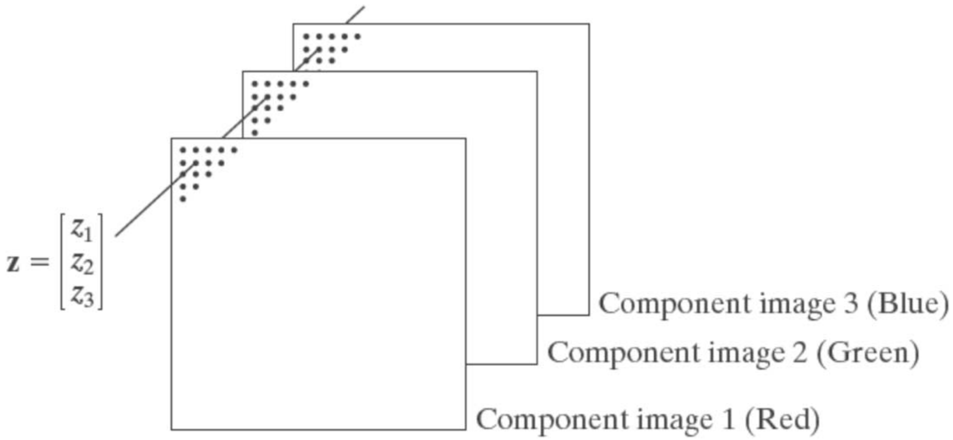


FIGURE 2.38 Formation of a vector from corresponding pixel values in three RGB component images.

Vector and Matrix Operations

- Once pixels have been represented as vectors we have at our disposal the tools of vector-matrix theory.

- For example, the *Euclidean distance*, D, between a pixel vector **z** and an arbitrary point **a** in n-dimensional space is defined as the vector product

$$D(\mathbf{z},\mathbf{a}) = \left[(\mathbf{z}-\mathbf{a})^T(\mathbf{z}-\mathbf{a})\right]^{\frac{1}{2}}$$

$$= \left[(z_1-a_1)^2+(z_2-a_2)^2+\cdots+(z_n-a_n)^2\right]^{\frac{1}{2}} \quad (2.6\text{-}27)$$

- Equation (2.6-27) sometimes is referred to as a *vector norm*, denoted by $||\mathbf{z}-\mathbf{a}||$.

Vector and Matrix Operations

- Another important advantage of pixel vectors is in linear transformations

$$\mathbf{w} = \mathbf{A}(\mathbf{z} - \mathbf{a}) \quad (2.6\text{-}28)$$

- We can express an image of size M x N as a vector of dimension MN x 1 by letting the first row of the image be the first N elements of the vector, the second row the next N elements, and so on.

- With image formed in this manner, we can express a broad range of linear processes applied to an image by using the notation

Noise

$$\underline{\mathbf{g}} = \underline{\mathbf{H}}\underline{\mathbf{f}} + \underline{\mathbf{n}} \quad (2.6\text{-}29)$$

Linear process

Input image

Image Transforms

- In some case, image processing task are best formulated by transforming the input images, carrying the specified task in a *transform domain*, and applying the inverse transform to return to the spatial domain.

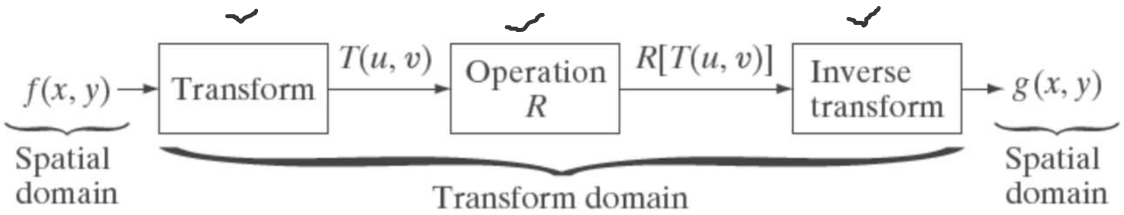


FIGURE 2.39 General approach for operating in the linear transform domain.

Image Transforms

- A particularly important class of 2-D linear transforms, denoted $T(u, v)$, can be expressed in the general form

$$T(u,v) = \sum_{x=0}^{M-1} \sum_{y=1}^{N-1} f(x,y)\, r(x,y,u,v) \quad (2.6\text{-}30)$$

Input Image

Forward transformation kernel

- We can recover $f(x, y)$ using the *inverse transform* of $T(u, v)$

$$f(x,y) = \sum_{u=0}^{M-1} \sum_{v=1}^{N-1} T(u,v)\, s(x,y,u,v) \quad (2.6\text{-}30)$$

Inverse transformation kernel

Image processing in the transforms domain

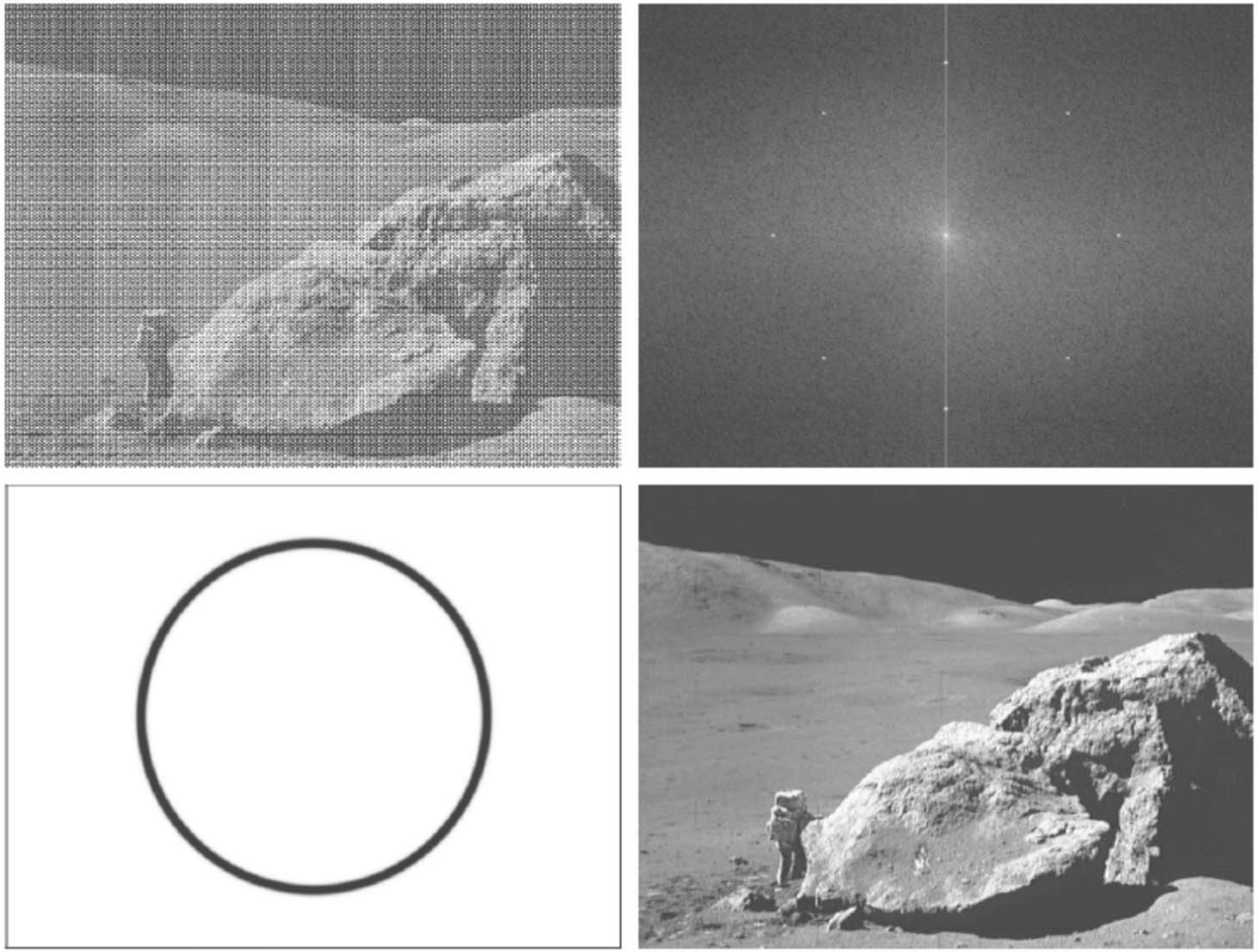


FIGURE 2.40 (a) Image corrupted by sinusoidal interference. (b) Magnitude of the Fourier transform showing the bursts of energy responsible for the interference. (c) Mask used to eliminate the energy bursts. (d) Result of computing the inverse of the modified Fourier transform. (Original image courtesy of NASA.)

The circle is regarded as a filter.

Image Transforms

- The kernel r is *separable*

$$r(x,y,u,v) = r_1(x,u)\,r_2(y,v) \quad (2.6\text{-}32)$$

- The kernel r is *Symmetric* *(r_1=r_2)*

$$r(x,y,u,v) = r_1(x,u)\,r_1(y,v) \quad (2.6\text{-}33)$$

- 2-D Fourier transform has the following forward and inverse kernels

$$r(x,y,u,v) = e^{-j2\pi(ux/M+vy/N)} \quad (2.6\text{-}34)$$

$$s(x,y,u,v) = \frac{1}{MN}e^{j2\pi(ux/M+vy/N)} \quad (2.6\text{-}35)$$

where

$$j = \sqrt{-1}$$

Image Transforms

- *Discrete Fourier transformation pair*

$$T(u,v)=\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)\,e^{-j2\pi(ux/M+vy/N)} \quad (2.6\text{-}36)$$

$$f(x,y)=\frac{1}{MN}\sum_{u=0}^{M-1}\sum_{v=0}^{N-1} T(u,v)\,e^{j2\pi(ux/M+vy/N)} \quad (2.6\text{-}37)$$

- When the forward and inverse kernels of a transform pair satisfy separable and symmetric conditions, and $f(x,y)$ is a square image of size $M\mathrm{x}M$, Eqs(2.6-30) and Eqs (2.6-31) can be expressed in matrix form

$$\mathbf{T}=\mathbf{AFA} \quad (2.6\text{-}38)$$

Image Transforms

- We pre- and post-multiply Eq. (2.6-38) by an inverse transformation matrix $\mathbf{B}$:

$$\mathbf{BTB} = \mathbf{BAFAB} \quad (2.6\text{-}39)$$

- If $\mathbf{B} = \mathbf{A}^{-1}$

$$\mathbf{F} = \mathbf{BTB} \quad (2.6\text{-}40)$$

indicating that $\mathbf{F}$ can be recovered completely from its forward transform.

- If $\mathbf{B}$ is not equal to $\mathbf{A}^{-1}$, then use of Eq. (2.6-40) yields an approximation:

$$\hat{\mathbf{F}} = \mathbf{BAFAB} \quad (2.6\text{-}41)$$

Probabilistic Method

- The probability, $p(z_k)$, of intensity level z_k occurring in a given image is estimated as

$$p(z_k) = \frac{n_k}{MN} \quad (2.6\text{-}42)$$

Number of time that intensity z_k occur

Total number of pixel

- Clearly,

$$\sum_{k=0}^{L-1} p(z_k) = 1 \quad (2.6\text{-}43)$$

- Once we have $p(z_k)$, we can determine a number of important image characteristics.

Probabilistic Method

- Mean (average) intensity

$$m = \sum_{k=0}^{L-1} z_k p(z_k) \quad (2.6\text{-}44)$$

- Variance of the intensities

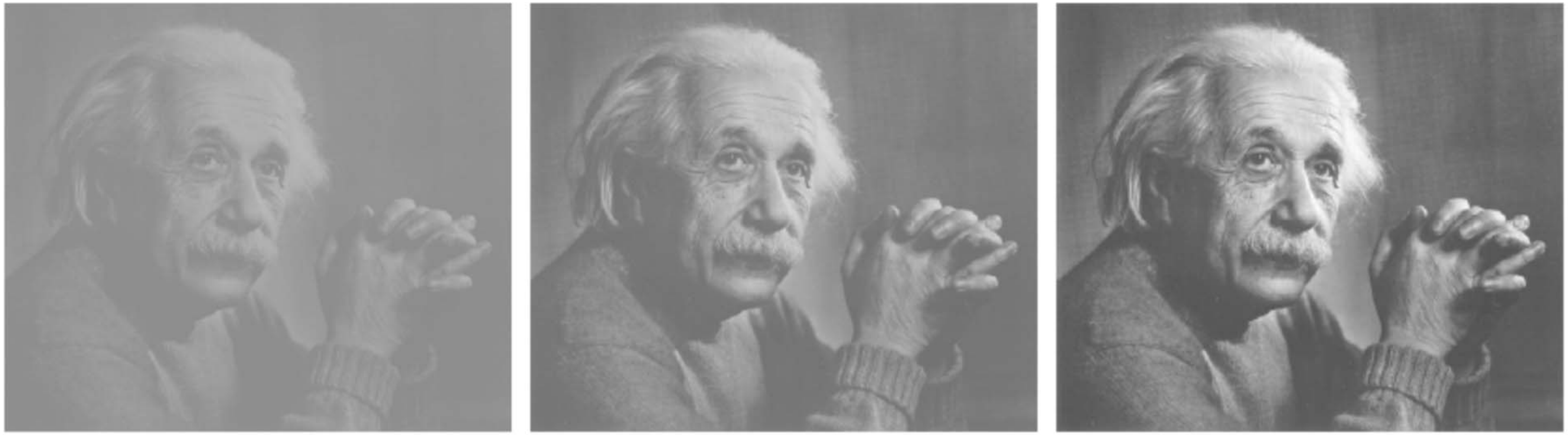
$$\sigma^2 = \sum_{k=0}^{L-1} (z_k - m)^2 p(z_k) \quad (2.6\text{-}45)$$

- nth moment of random variable z about the mean

$$\mu_n(z) = \sum_{k=0}^{L-1} (z_k - m)^n p(z_k) \quad (2.6\text{-}46)$$

we see that $\mu_0(z) = 1$, $\mu_1(z) = 0$, $\mu_2(z) = \sigma^2$,

Comparison of standard deviation values as measures of image intensity contrast



$\sigma = 14.3$ $\sigma = 31.6$ $\sigma = 49.2$

a b c

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