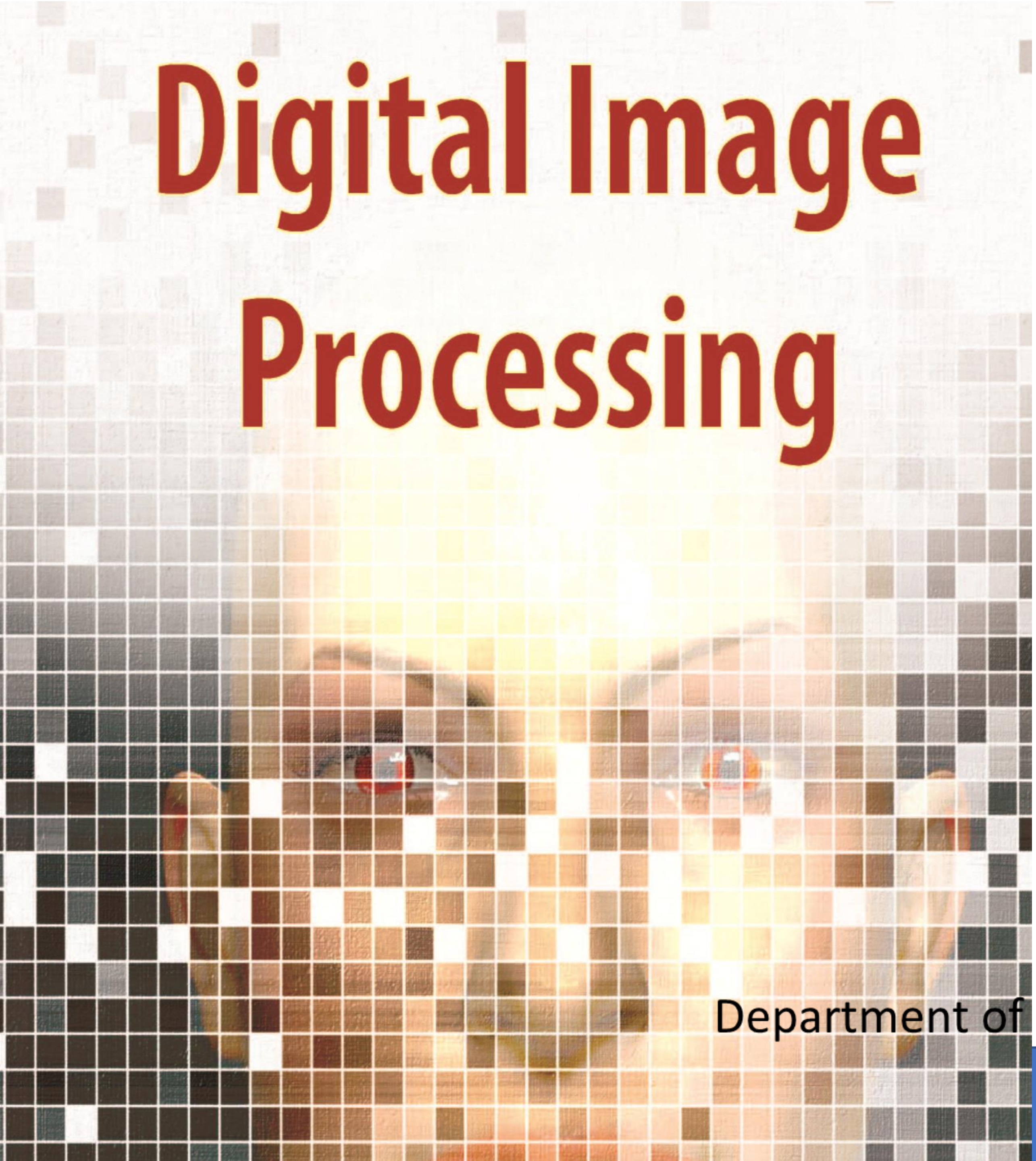


Image Data Handling

Prof. Sheli Sinha Chaudhuri
Department of Electronics and Telecommunication Engineering
Jadavpur University

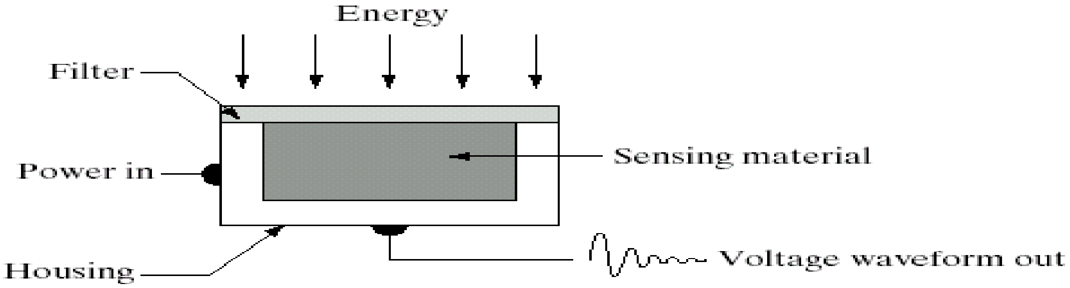
Chapter 2

Lecture 2: Image Data Handling-I

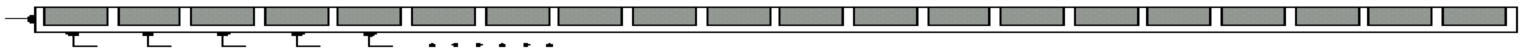
Contents

- Image Acquisition
- Image File Formats
- Interpolation

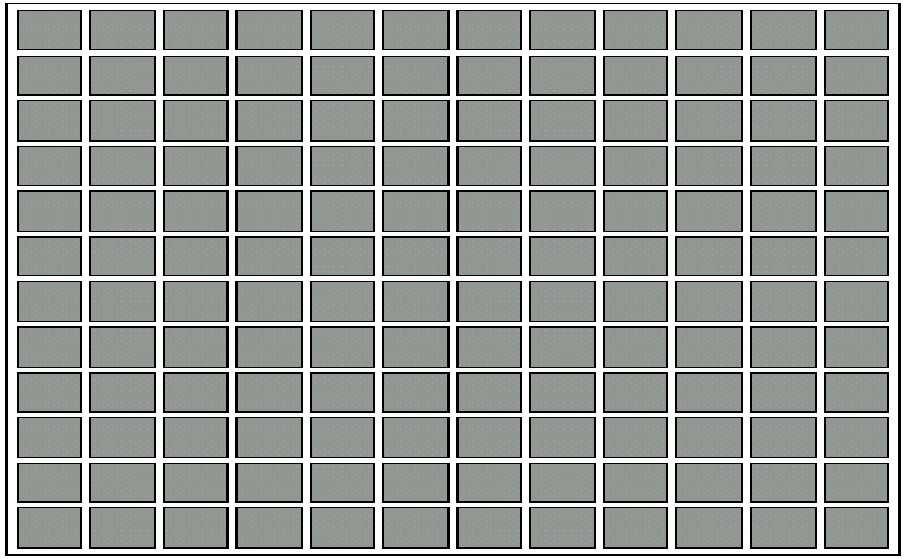
Image Sensing and Acquisition



Single sensor



Line sensor



Array sensor

(Images from Rafael C. Gonzalez and Richard Wood, Digital Image Processing, 2nd Edition.

Image Sensors: Single Sensors

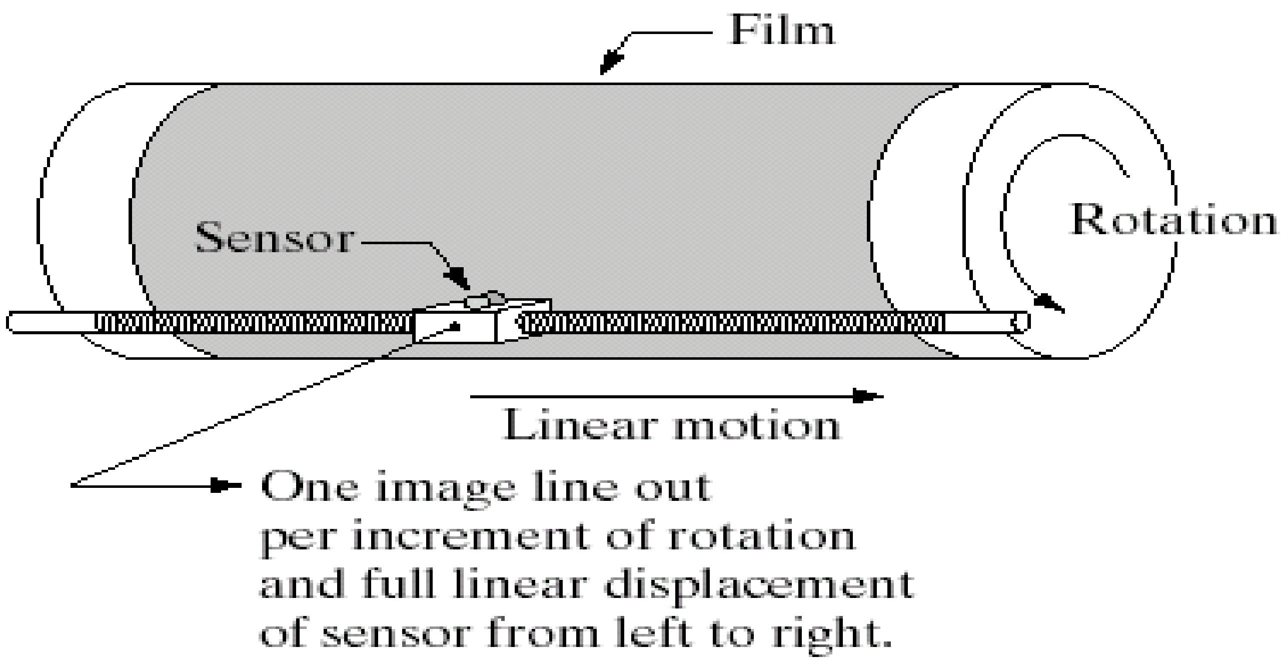
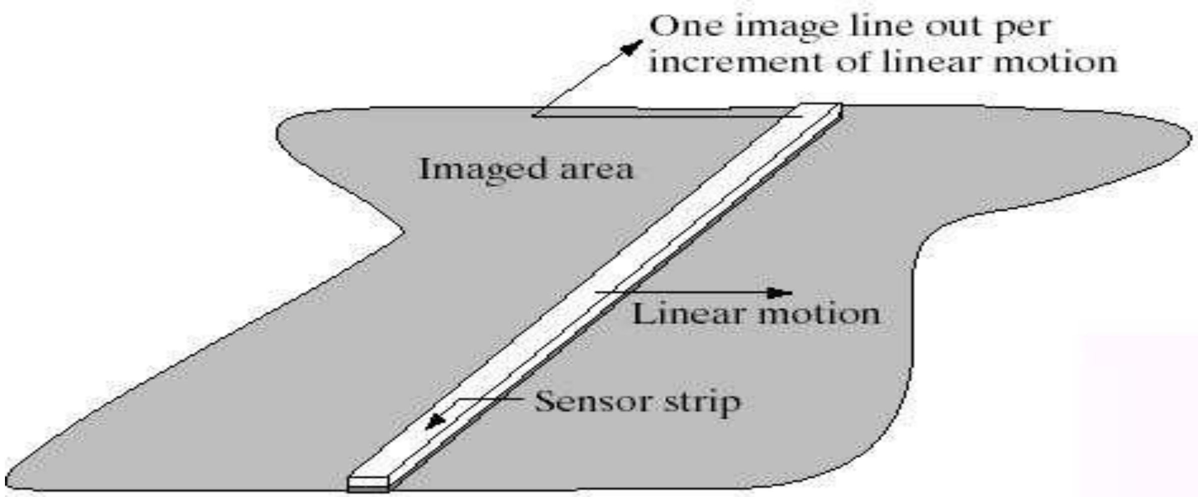
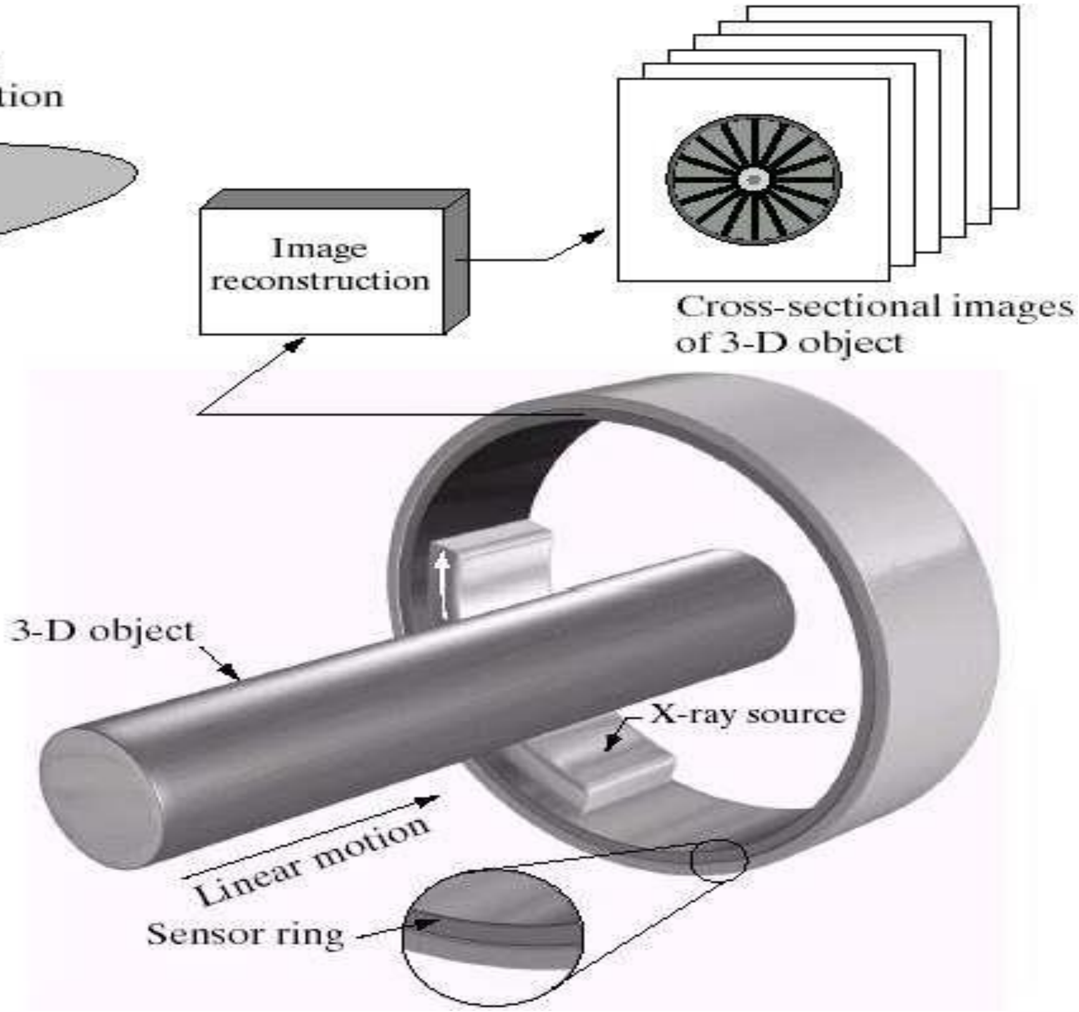


FIGURE 2.13 Combining a single sensor with motion to generate a 2-D image.

(Images from Rafael C. Gonzalez and Richard Wood, Digital Image Processing, 2nd Edition.

Image Sensors: Line Sensor



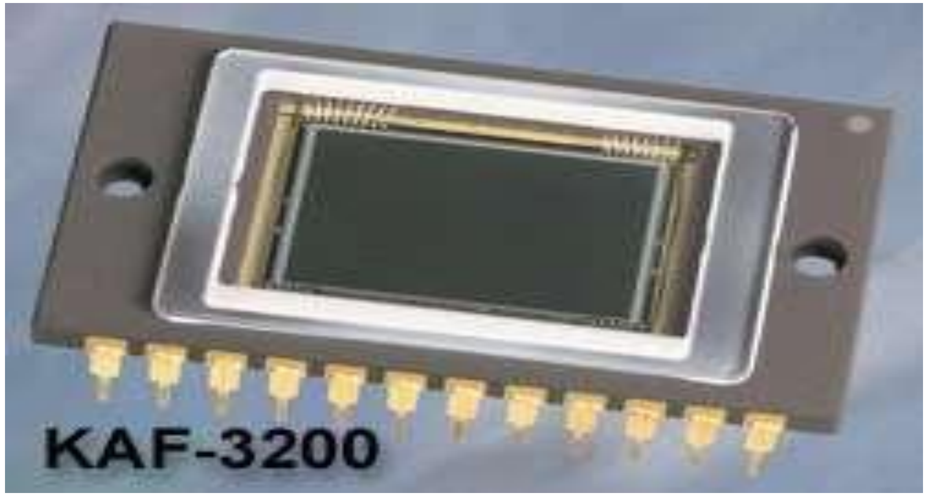


Fingerprint sweep sensor

Computerized Axial Tomography

(Images from Rafael C. Gonzalez and Richard Wood, Digital Image Processing, 2nd Edition.

Image Sensors : Array Sensor Charge-Coupled Device (CCD)



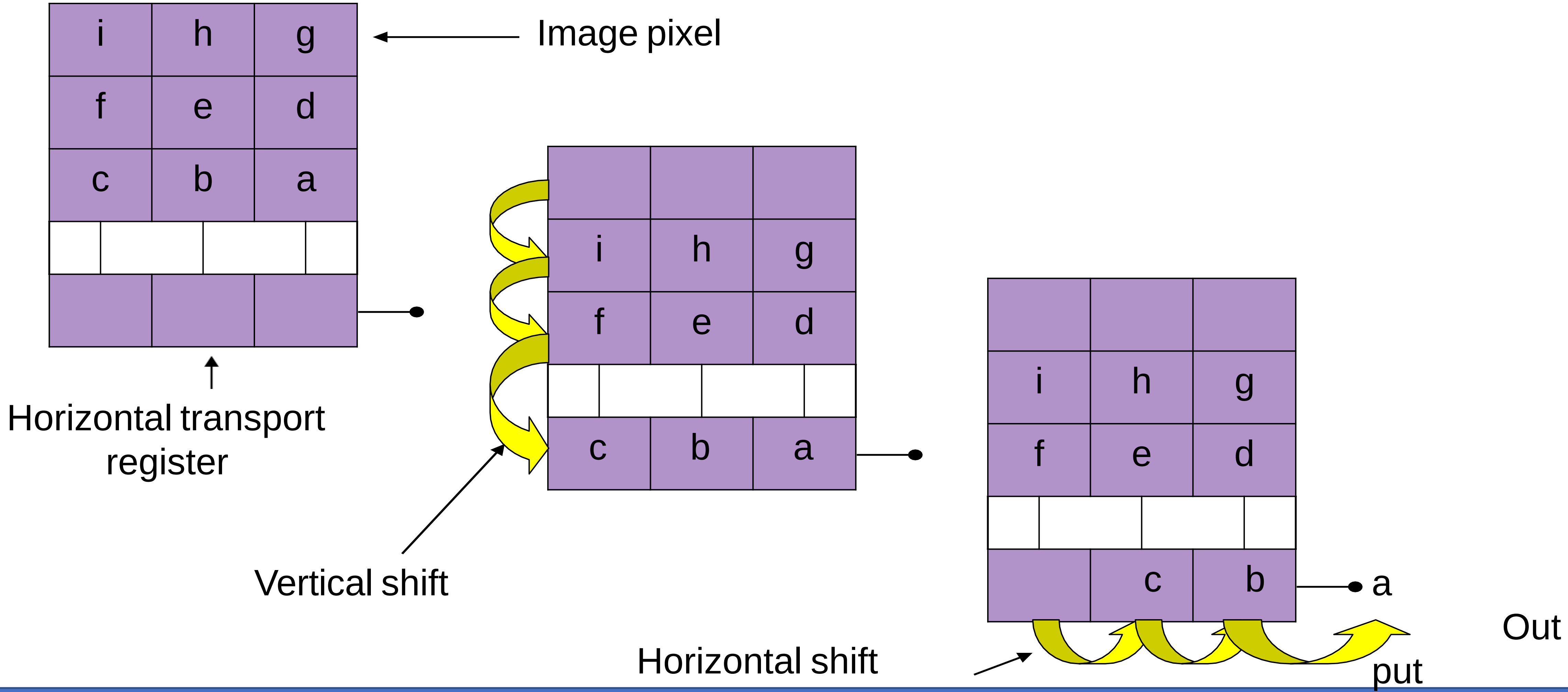
Used for convert a continuous image into a digital image

Contains an array of light sensors

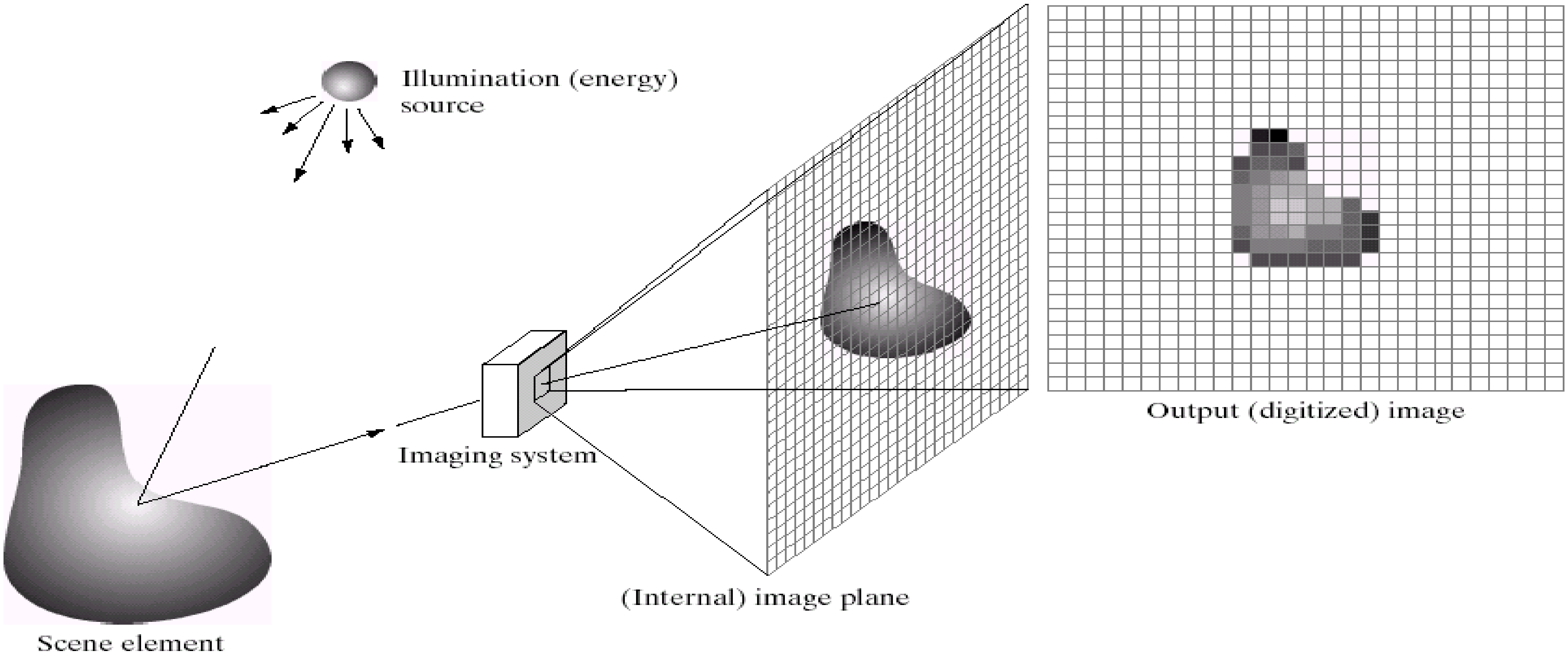
Converts photon into electric charges accumulated in each sensor unit

CCD KAF-3200E from Kodak.
(2184 x 1472 pixels, Pixel size 6.8 microns^2)

Image Sensor: How CCD works



Digital Image Acquisition Process



(Images from Rafael C. Gonzalez and Richard Wood, Digital Image Processing, 2nd Edition.

Generating a Digital Image

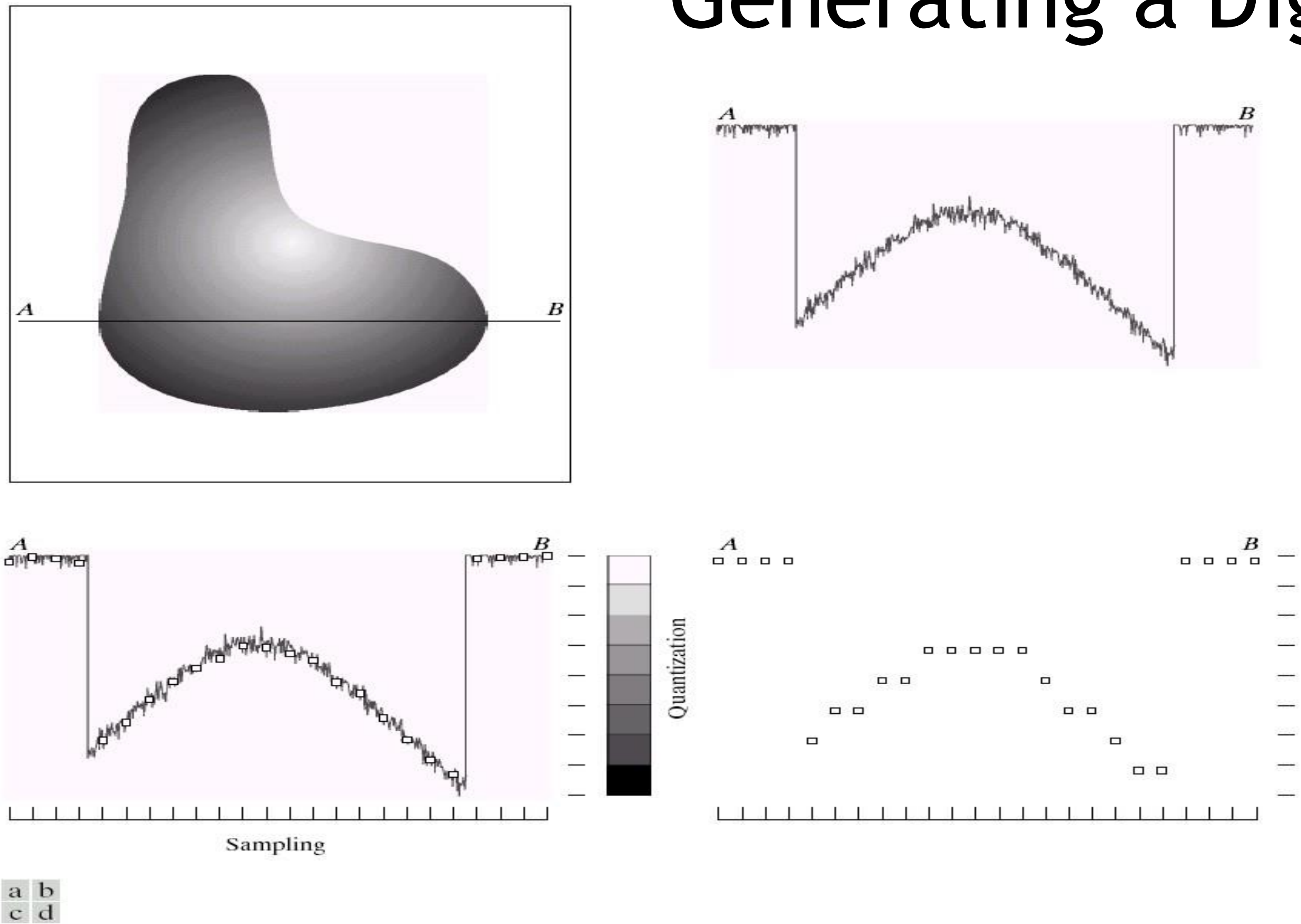


FIGURE 2.16 Generating a digital image. (a) Continuous image. (b) A scan line from *A* to *B* in the continuous image, used to illustrate the concepts of sampling and quantization. (c) Sampling and quantization. (d) Digital scan line.

Image Sampling and Quantization

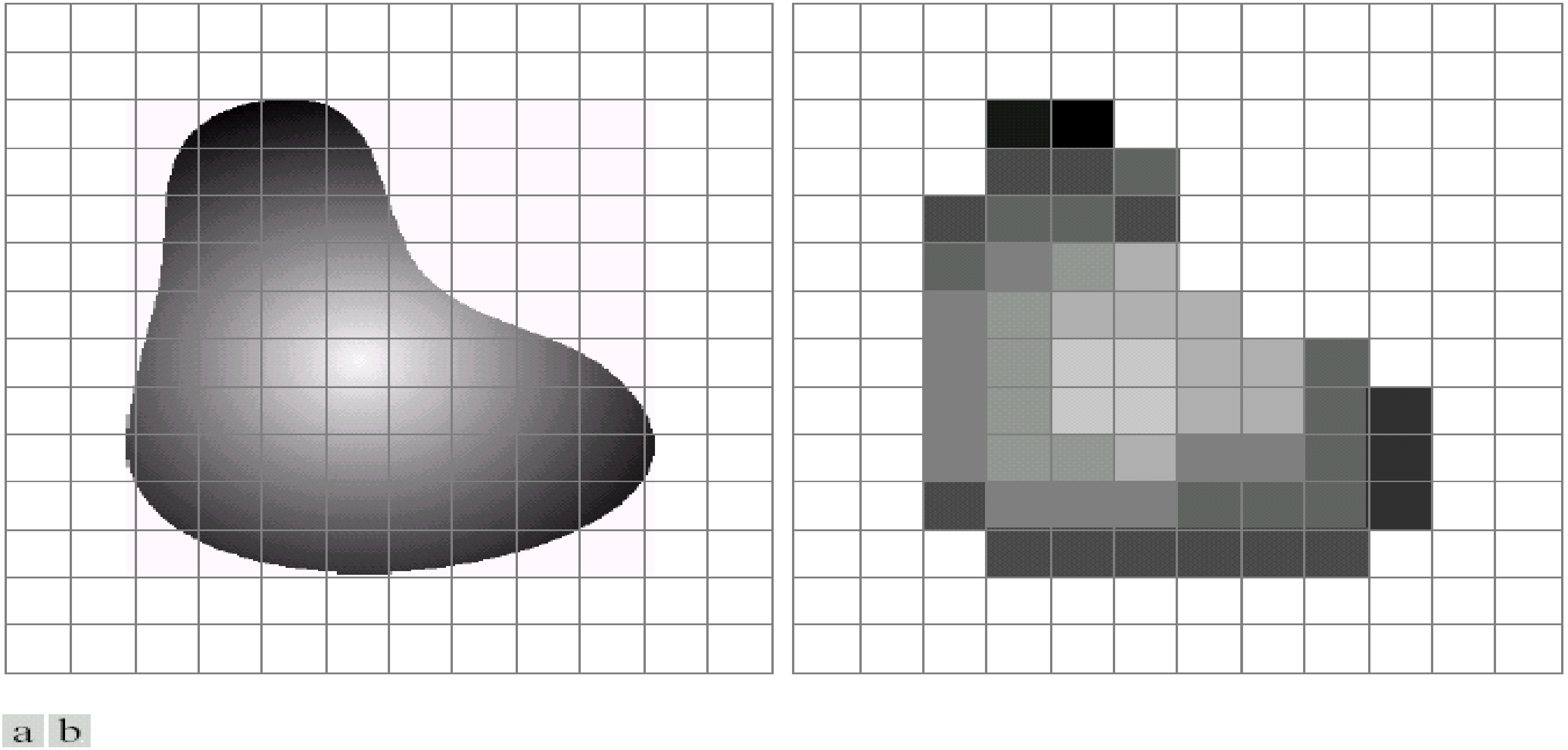


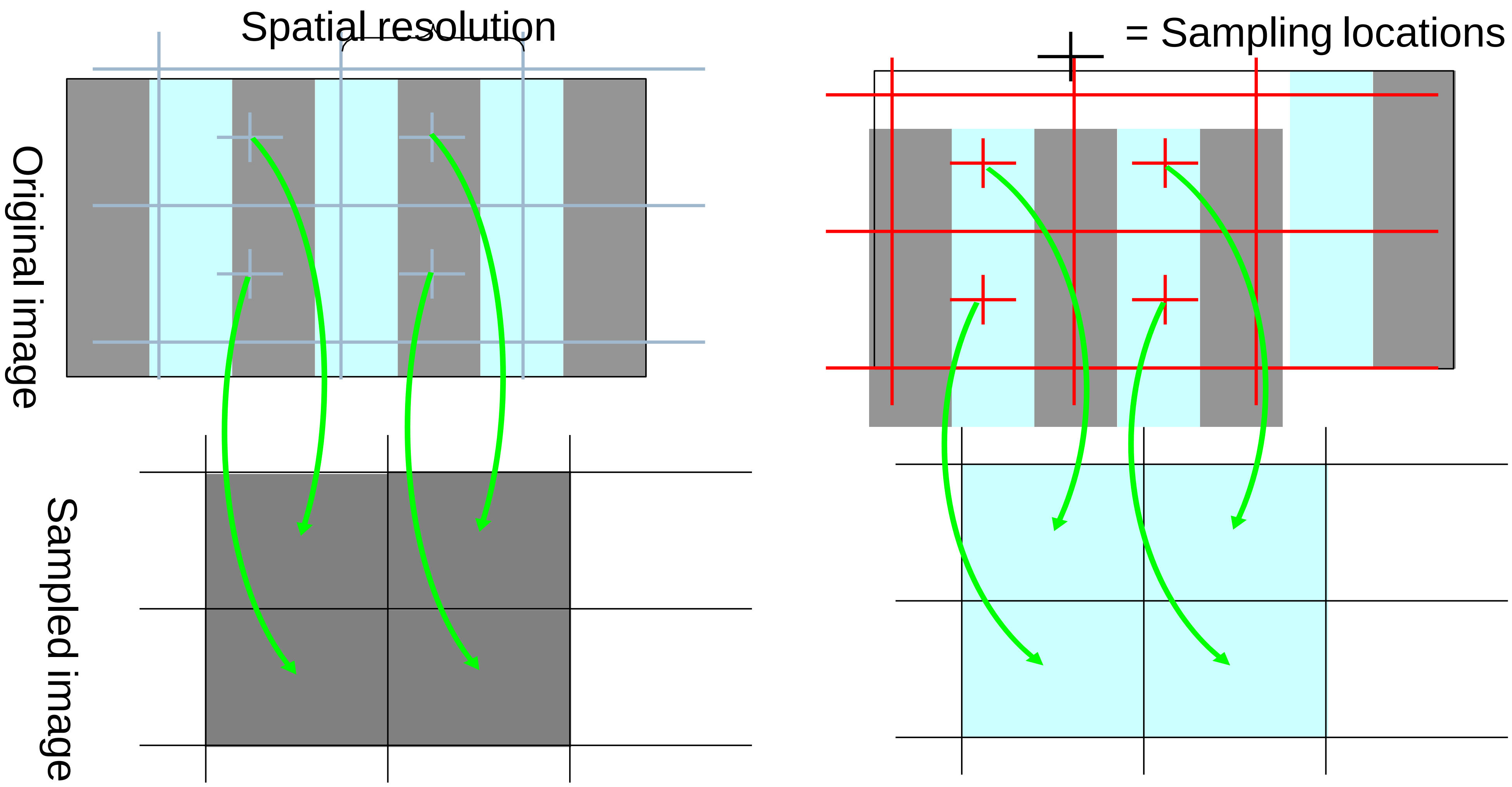
FIGURE 2.17 (a) Continuos image projected onto a sensor array. (b) Result of image sampling and quantization.

Image sampling: discretize an image in the spatial domain

Spatial resolution / image resolution: pixel size or number of pixels

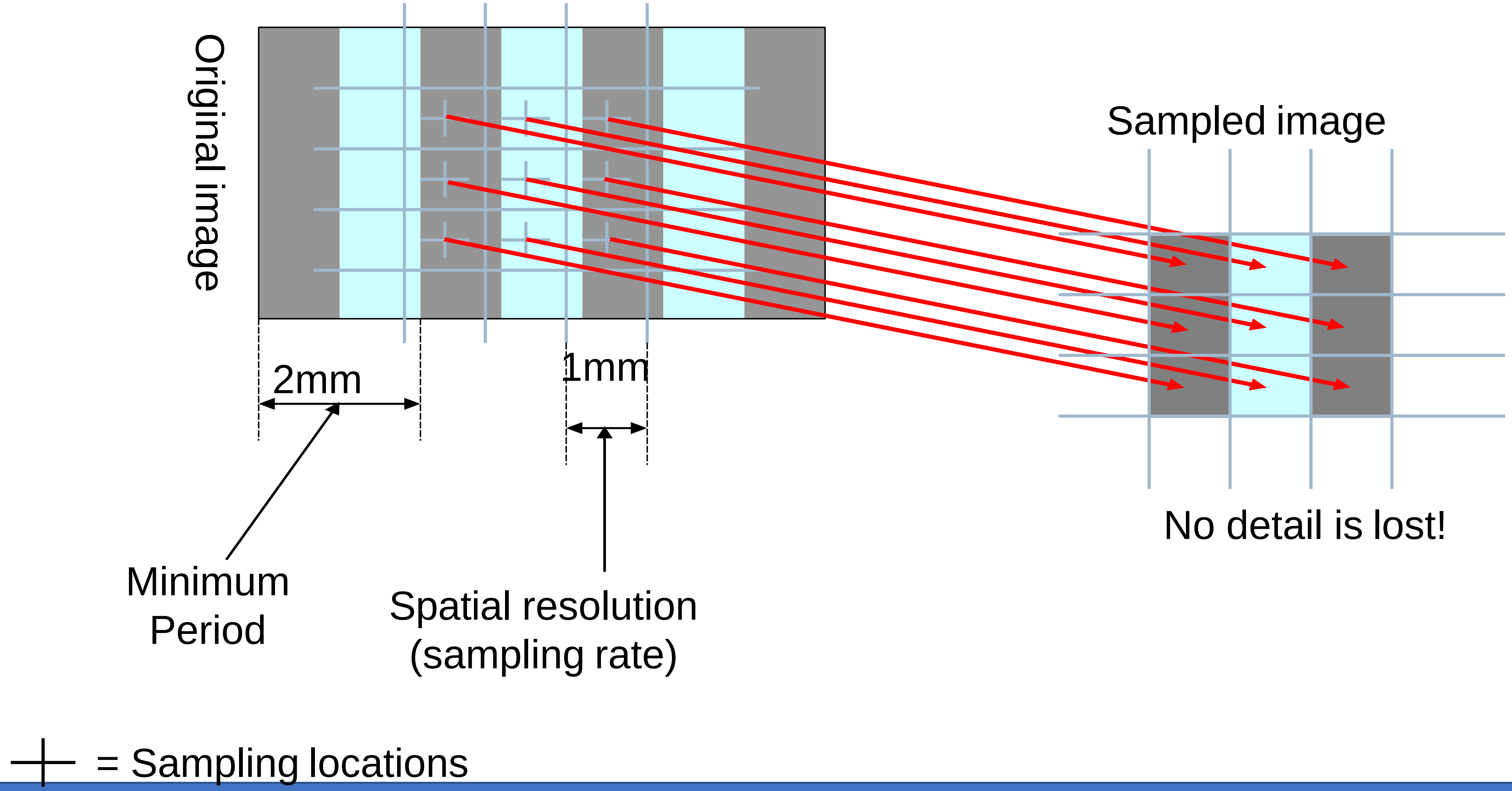
(Images from Rafael C. Gonzalez and Richard Wood, Digital Image Processing, 2nd Edition.

Choice of Spatial Resolution



Under sampling, we lost some image details!

Choice of spatial resolution based on Nyquist Rate

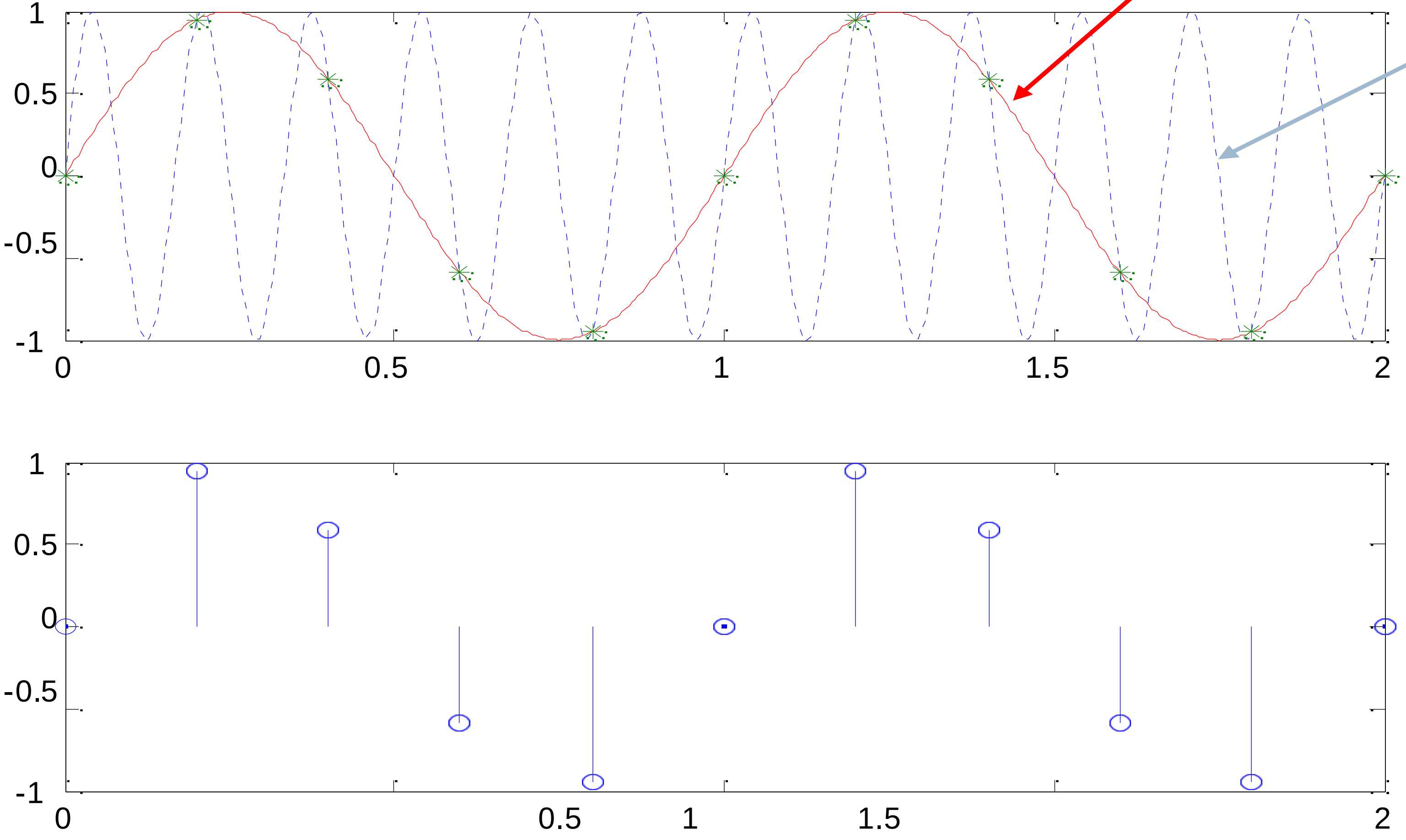


Nyquist Rate:
Spatial resolution must be less or equal to half of the minimum period of the image or sampling frequency must be greater or equal to twice the maximum frequency

Aliased Frequency

$x_1(t) = \sin(2\pi t), \quad f = 1$

$x_2(t) = \sin(12\pi t), \quad f = 6$



Sampling rate:
5 samples/sec

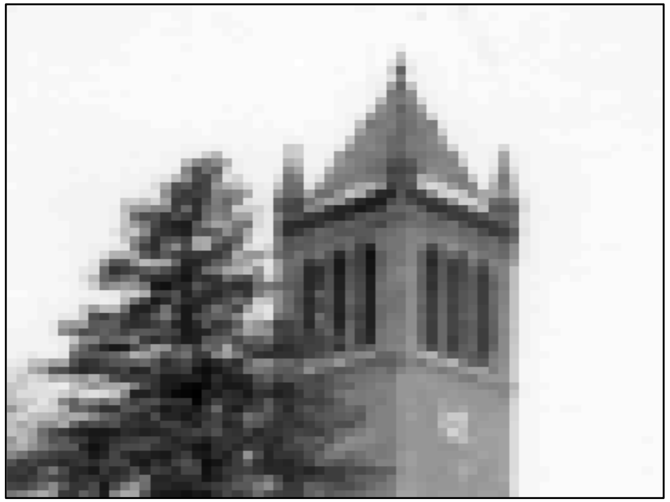
Two different frequencies but the same results !

Effect of Spatial Resolution

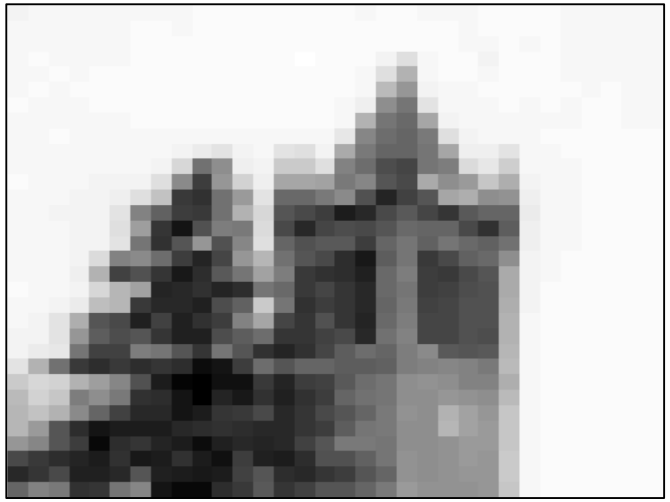
256x256 pixels

128x128 pixels

64x64 pixels



32x32 pixels



Spatial Resolution

- It is a measure of the smallest discernible detail in an image
- Can be stated in *line pairs per unit distance,* and *dots(pixels) per unit distance*
- *Dots per unit distance* commonly used in printing and publishing industry (dots per inch)
- Newspaper are printed with a resolution of 75 dpi, magazines at 133 dpi, and glossy brochures at 175 dpi
- examples

Effect of Spatial Resolution

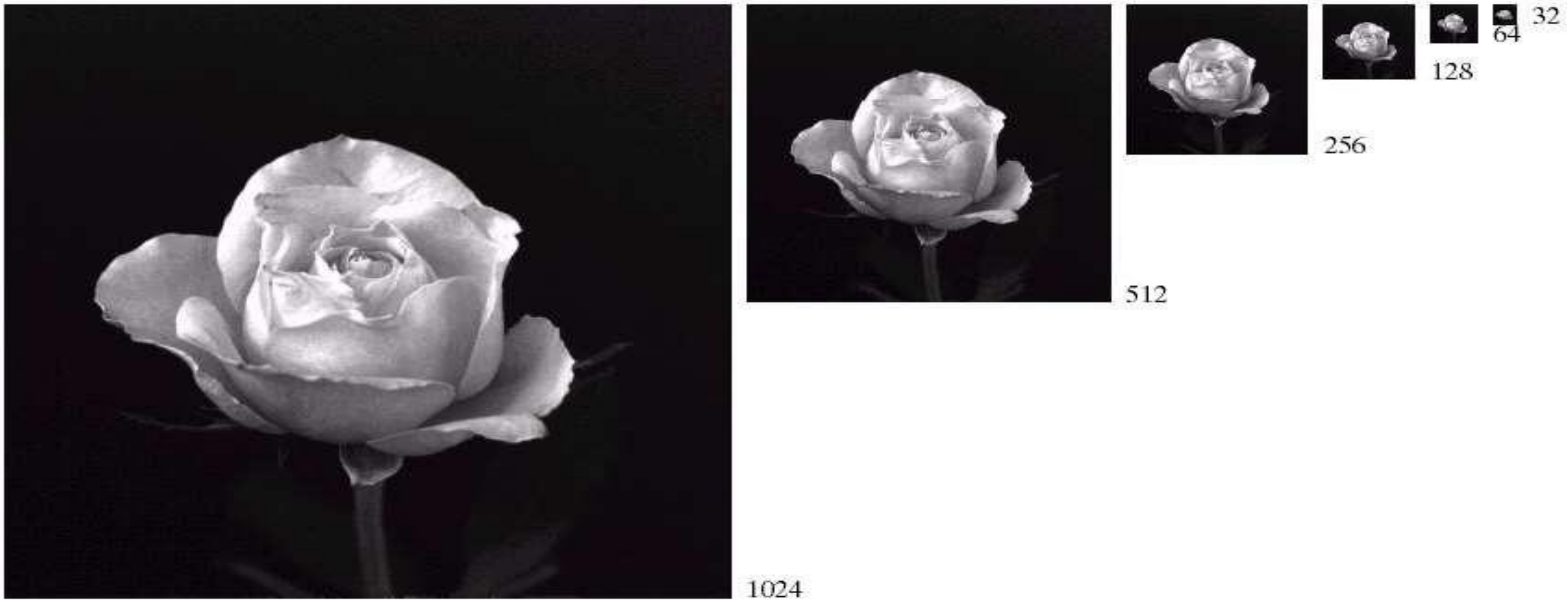
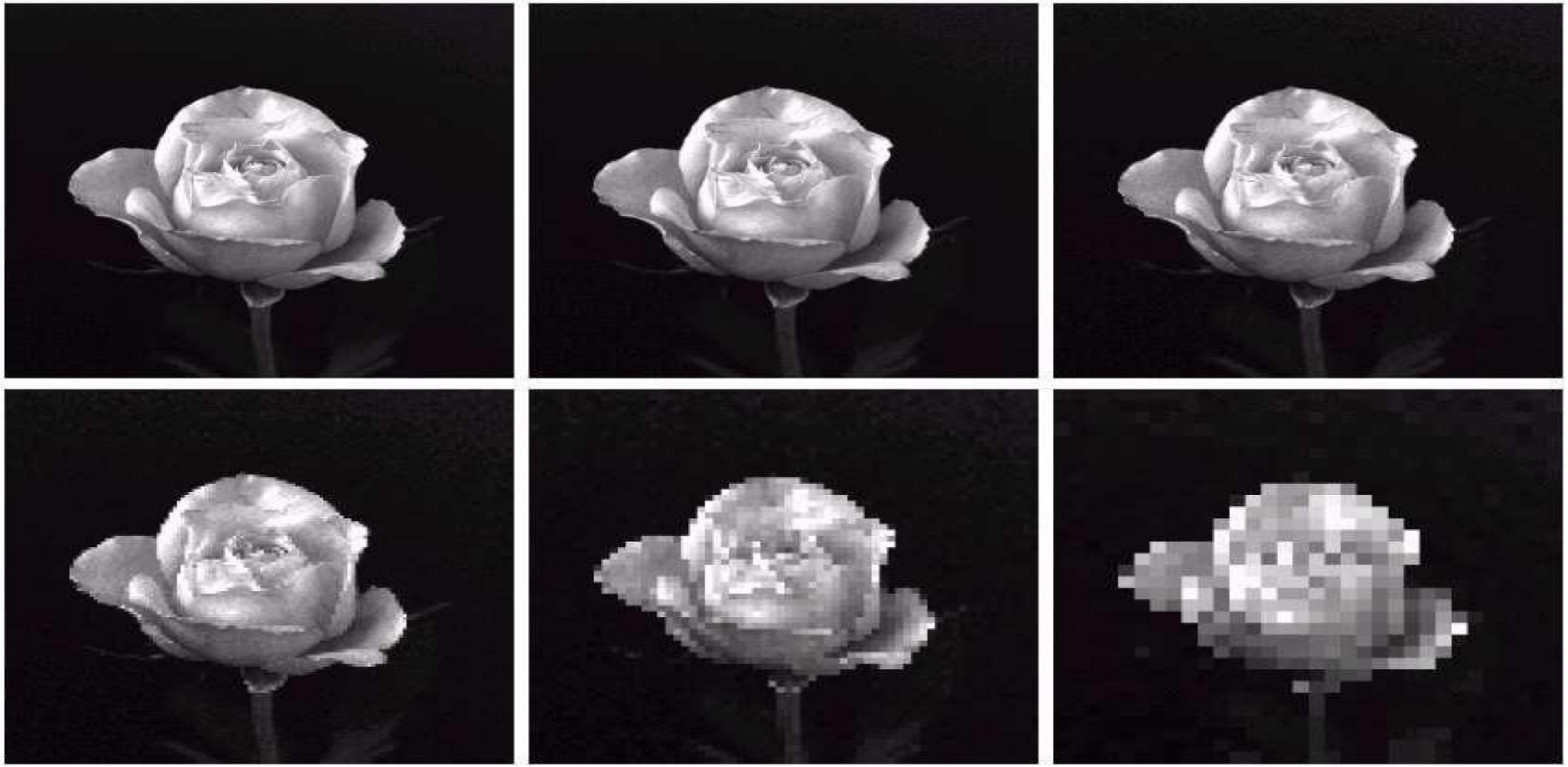


FIGURE 2.19 A 1024 × 1024, 8-bit image subsampled down to size 32 × 32 pixels. The number of allowable gray levels was kept at 256.

Effect of Spatial Resolution



a b c
d e f

FIGURE 2.20 (a) 1024 × 1024, 8-bit image. (b) 512 × 512 image resampled into 1024 × 1024 pixels by row and column duplication. (c) through (f) 256 × 256, 128 × 128, 64 × 64, and 32 × 32 images resampled into 1024 × 1024 pixels.

Effect of Interpolation on Spatial Resolution

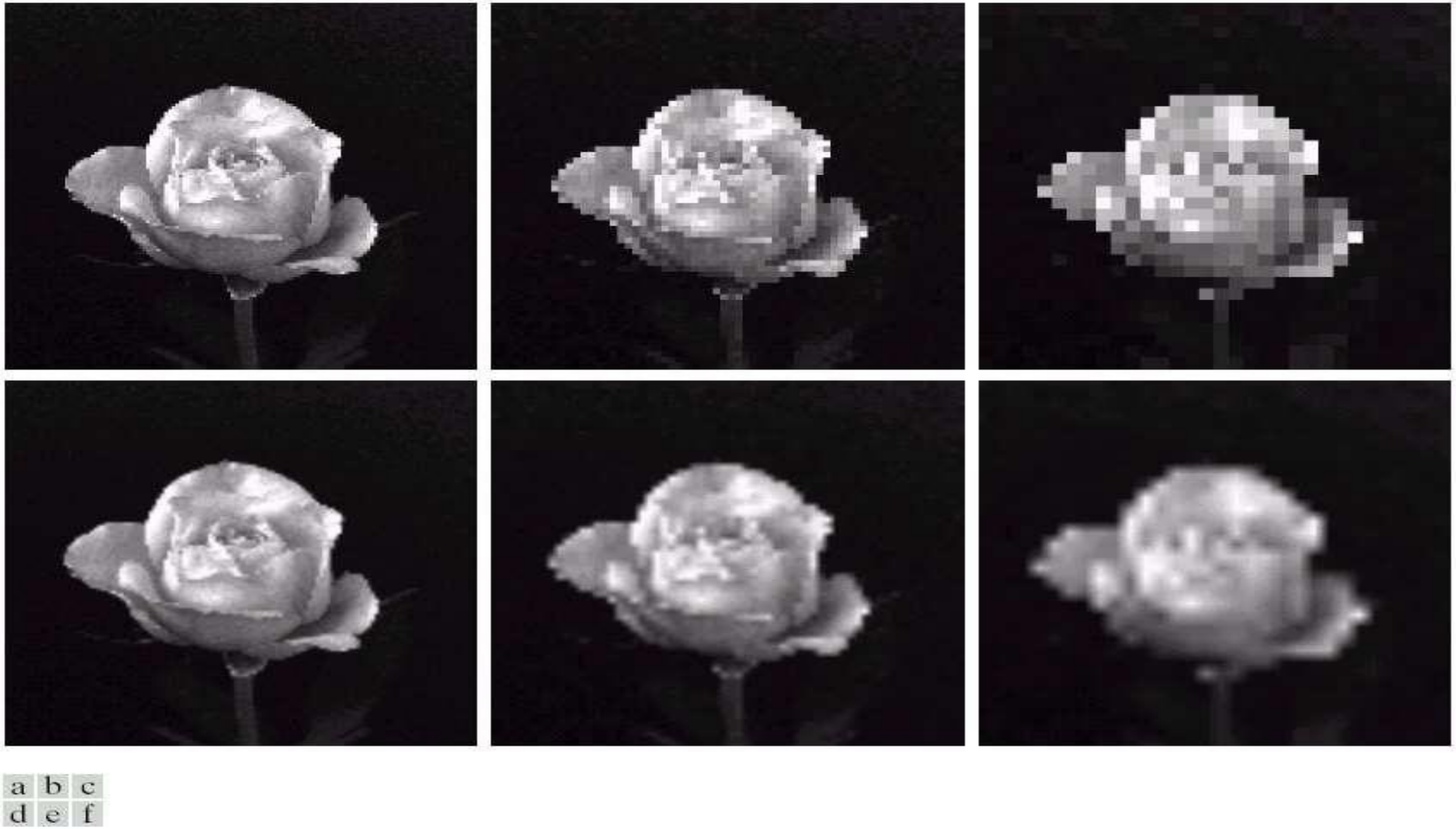


FIGURE 2.25 Top row: images zoomed from 128 × 128, 64 × 64, and 32 × 32 pixels to 1024 × 1024 pixels, using nearest neighbor gray-level interpolation. Bottom row: same sequence, but using bilinear interpolation.

Down sampling is an irreversible process.

Image Quantization

Image quantization:

discretize continuous pixel values into discrete numbers

Color resolution/ color depth/ levels:

- No. of colors or gray levels or
- No. of bits representing each pixel value
- No. of colors or gray levels N_c is given by

$$N_c = 2^b$$

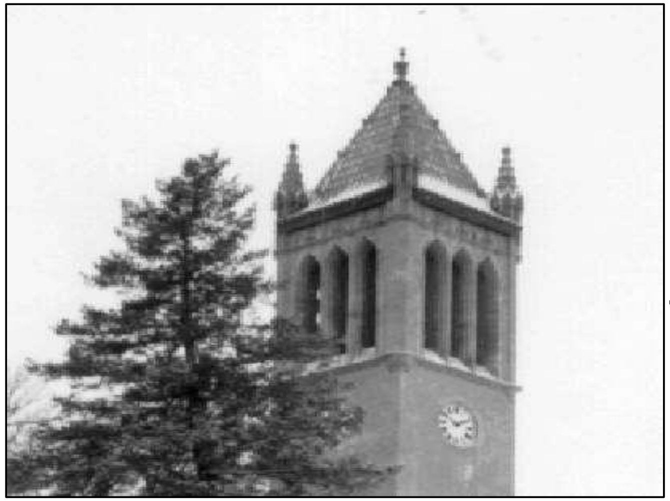
where b = no. of bits

Intensity Resolution

- It refers to the smallest discernible change in intensity level
- Number of intensity levels usually is an integer power of two
- Also refers to number of bits used to quantize intensity as the intensity resolution
- Which intensity resolution is good for human perception 8 bit, 16 bit, or 32 bit

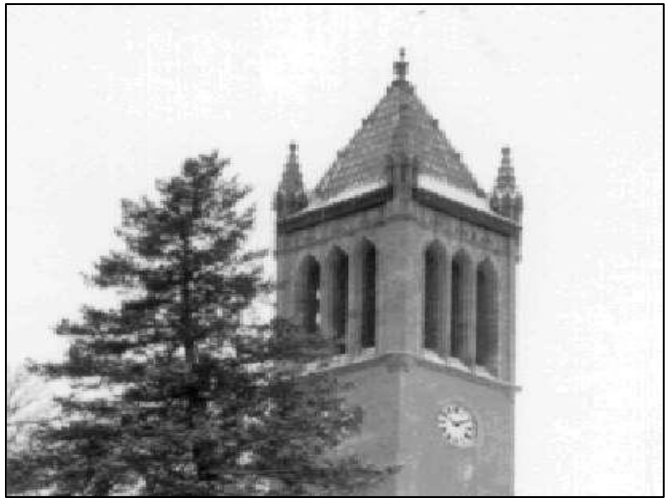
Effect of Quantization Levels or Intensity Resolution

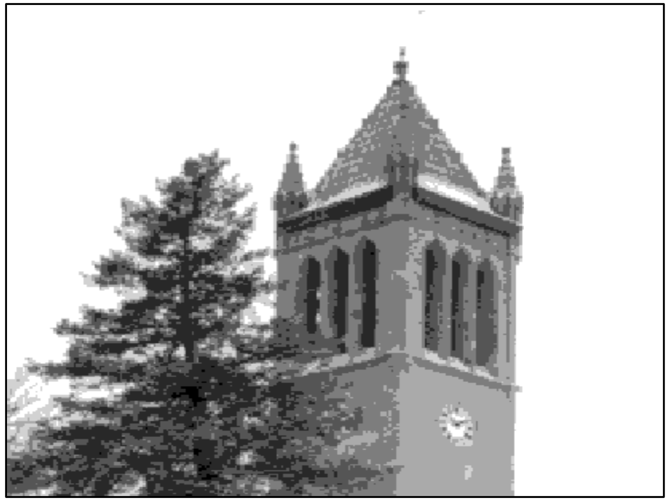
256 levels



128 levels

64 levels

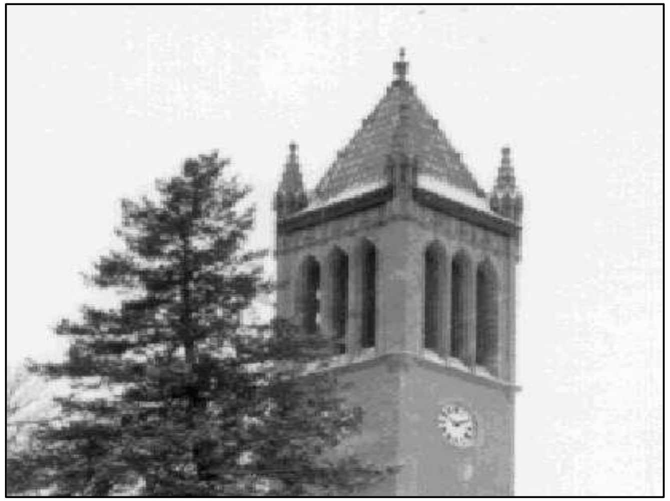


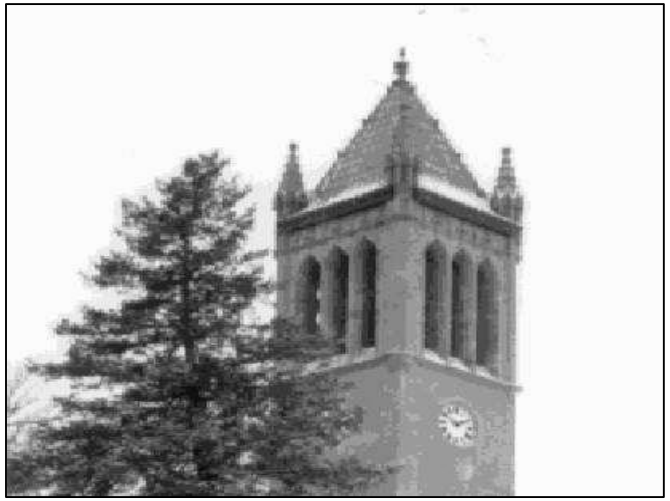


32 levels

Effect of Quantization Levels or Intensity Resolution

16 levels

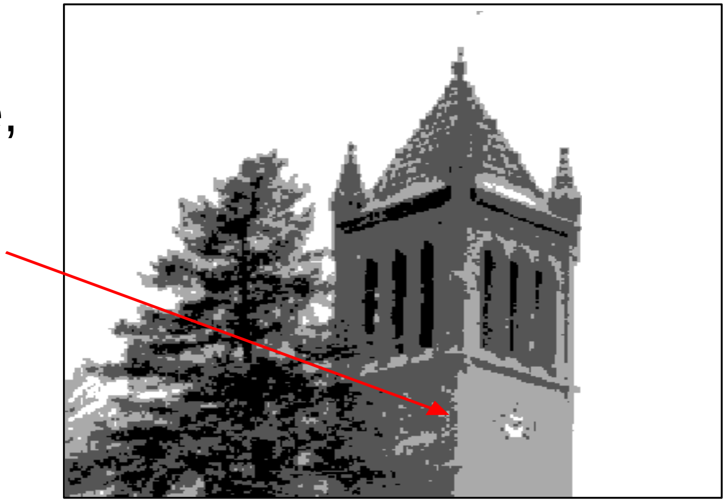




8 levels

In this image, it is easy to see false contour.

4 levels



2 levels

Selection of Suitable Size and Pixel Depth of Images

The word “suitable” is subjective: depending on “subject”.

Low detail image

Lena image

Medium detail image

Cameraman image

High detail image

Crowd image

To satisfy human mind

1. For images of the same size, the low detail image may need more pixel de
2. As an image size increase, fewer gray levels may be needed.

Isopreference Curve

- Curves tend to become more vertical as the detail in the image increases
- Image with a large amount of detail only a few intensity levels may be needed

These curves are:
Graphic representation of quantified values of image quality whose points all refer to images that are of a constant subjective quality.

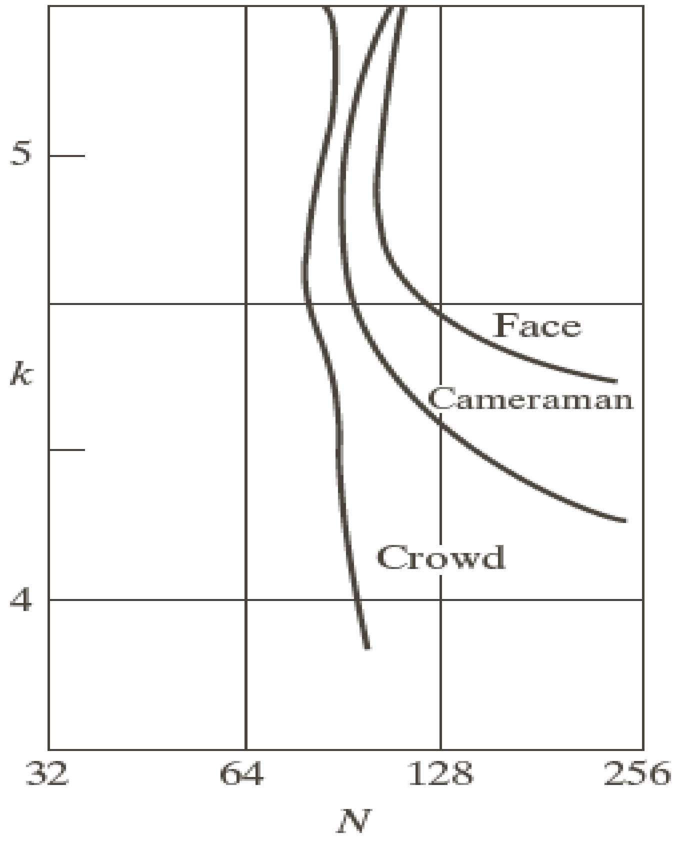
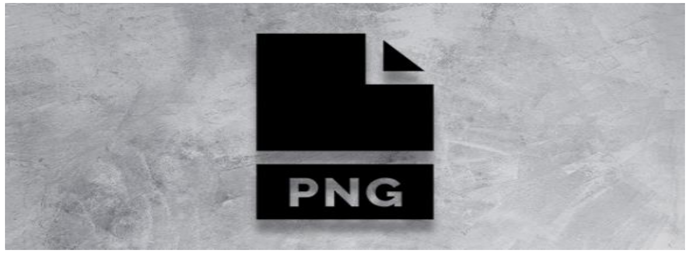
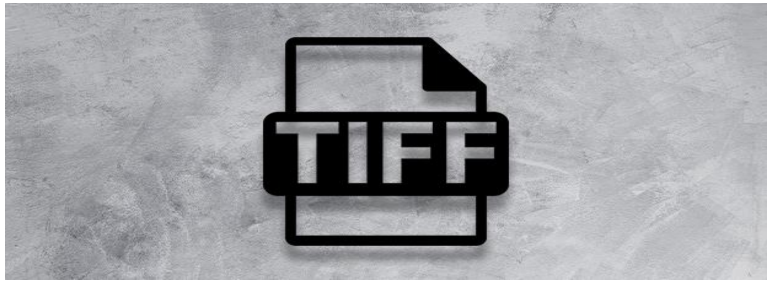


FIGURE 2.23 Typical isopreference curves for the three types of images in Fig. 2.22.

Image Formats





RAW Image Files (.raw, .cr2, .nef, .orf, .sr2, and more)

RAW images are images that are unprocessed that have been created by a camera or scanner. Many digital SLR cameras can shoot in RAW, whether it be a .raw, .cr2, or .nef. These RAW images are the equivalent of a digital negative, meaning that they hold a lot of image information, but still need to be processed in an editor such as Adobe Photoshop or Lightroom.

Compression: None

Best For: Photography

Special Attributes: Saves metadata, unprocessed, lots of information

TIFF (.tif, .tiff)

TIFF or **Tagged Image File Format** are lossless image files.
They do not need to compress or lose any image quality or information
Thus they allow very high-quality images but of large size.
Hence are the most used file format by photographers and designers.
Images stored as TIFF files are best for post-processing.
TIFF files can be used with any photo editing software .

Compression: Lossless-no compression. Very high quality images
Best For: High quality prints, professional publication, archival copies
Special Attributes: Can save transparencies
When to Use TIFF Image Format:
For images to be stored without losing details.
For images to be reproduced or printed.

Bitmap (.bmp)

Bitmap (.bmp)
BMP or Bitmap Image File is a format developed by Microsoft for Windows. There is no compression or information loss with BMP files which allow images to have very high quality, but also very large file sizes. Due to BMP being a proprietary format, it is generally recommended to use TIFF files.
Compression: None
Best For: High quality scans, archival copies

JPEG (.jpeg or .jpg)

JPEG stands for Joint Photographic Experts Group, and it's extension is widely written as **.jpg**.
This most used image file format is used to store photos all over the world, and is generally a default file format for saving images.
Most of the images you find online will download as .jpg files.
JPEG files come in different quality levels like low, medium and high. Low quality JPEGs are more compressed than high quality versions. So, for a high quality image, a less compressed JPEG option is required.
Most commonly, JPEG images are great for sharing on social media, via email and on websites. These files are relatively small, so they take up less space on your memory cards and computer storage.
Saving images as JPEGs can compromise the quality of that image.

When to Use JPEG Image Format:

Complex images with a lot of different colors, like photographs
To compress highly detailed images
For print

GIF (.gif)

GIF or Graphics Interchange Format files are widely used for web graphics, because they are limited to only 256 colors, can allow for transparency, and can be animated. GIF files are typically small is size and are very portable.

Compression: Lossless - compression without loss of quality

Best For: Web Images

Special Attributes: Can be Animated, Can Save Transparency

GIF stands for Graphics Interchange Format, and it’s quite similar to PNG in terms of its image quality preservation. With GIF image files, you can also create short animations for web.

This image format has a smaller color range, so it’s not suitable for all photos. Like PNG, GIF images can be transparent. But, unlike PNG, GIF doesn’t support partial transparency, which means you can’t use them to preserve shadow effects in your photos.

When to Use GIF Image Format:

For simple images with few colors, like icons

For animated images

PNG (.png)

PNG or Portable Network Graphics files are a lossless image format originally designed to improve upon and replace the GIF format. PNG files are able to handle up to 16 million colors, unlike the 256 colors supported by GIF.

Compression: Lossless - compression without loss of quality

Best For: Web Images

Special Attributes: Save Transparency

PNG stands for Portable Network Graphics. It's an important file format that helps a lot in photo editing. You can use PNGs for completely transparent backgrounds or drop shadows (partial transparency) for the great effects.

This image format will not sacrifice the quality and details of the photos, but that means that they are typically larger in size than JPEGs. Thus, PNGs are best for small images like logos.

When to Use PNG Image Format:

Images with transparency

Small images, like logos

Online images

To retain the quality of a detailed image, provided that you have storage for a larger file size

PSD (.psd)

PSD stands for Photoshop Document. When an image is saved from Adobe Photoshop, the program's default is to save that photo as a PSD file. That file can be used to edit the individual layers created in Photoshop at a later time.

This file format is not suitable for web, nor is it suitable for clients because it isn't versatile.

It is one of the best image formats for maintaining the quality of the image over a long period of time.

The layers should not be merged before the PSD file is saved, or some of the editing capabilities will be lost.

The size of PSD files is quite large. But, the trade-off is access to the highest quality version of the images for printing.

When to Use PSD Image Format:

To save images that may need to be edited again in the future

To retain the quality and detail of images to be printed.

Interpolation

Image interpolation occurs when you resize or distort your image from one pixel grid to another. Image resizing is necessary when you need to increase or decrease the total number of pixels, whereas remapping can occur when you are correcting for lens distortion or rotating an image. Zooming refers to increase the quantity of pixels, so that when you zoom an image, you will see more detail.

Interpolation

Interpolation works by using known data to estimate values at unknown points. Image interpolation works in two directions, and tries to achieve a best approximation of a pixel's intensity based on the values at surrounding pixels. Common interpolation algorithms can be grouped into two categories: adaptive and non-adaptive. Adaptive methods change depending on what they are interpolating, whereas non-adaptive methods treat all pixels equally. Non-adaptive algorithms include: nearest neighbour, bilinear, bicubic, spline, sinc, lanczos and others. Adaptive algorithms include many proprietary algorithms in licensed software such as: Qimage, PhotoZoom Pro and Genuine Fractals.

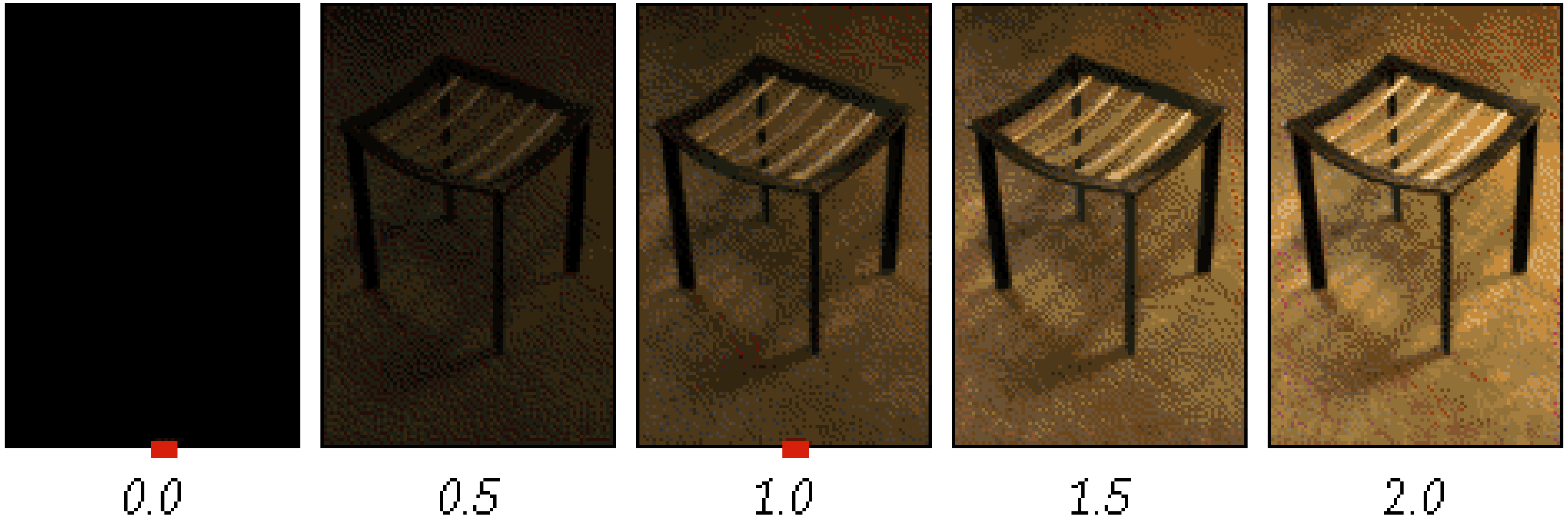


Image Interpolation

- Used in image resizing (zooming and shrinking), rotating, and geometric corrections
- Interpolation is the process of using known data to estimate values at unknown locations
- Nearest Neighbor interpolation
  - It assigns to each new location the intensity of its nearest neighbor in the original image
  - Produce undesirable artifacts, such as severe distortion of straight edges
- Bilinear Interpolation
  - We use the four nearest neighbors to estimate the intensity
  - $V(x, y) = ax + by + cxy + d$

Bilinear Image Interpolation

10	20
30	40

2x2

2x →

10	12	17	20
15	17	22	25
25	27	32	35
30	32	37	40

4x4

10 2/3of10+1/3of20 1/3of10+2/3of20 20

Image Interpolation

- Need to solve four equations
- Better results than nearest neighbor interpolation, with a modest increase in computational burden

- Bicubic Interpolation
  - Involves sixteen neighbors to estimate intensity

  - $V(x, y) = \sum\sum a_{ij}\, x^i\, y^j$ (i, j = 0 to 3)
  - Need to solve sixteen equations
  - Gives better results than other methods
  - More complex
  - Used in Adobe Photoshop, and Corel Photopaint

Bicubic Image Interpolation

10	20
30	40

2x2

2x →

7	10	16	19
13	17	22	26
24	28	33	37
31	34	40	43

4x4

Bilinear Interpolation method uses 4 nearest neighbours to determine the output while Bicubic method uses 16 neighbours. Weight distribution in Bicubic method is done differently.

$$p(x,y) = \sum_{i=0}^{3} \sum_{j=0}^{3} a_{ij} x^i y^j .$$

(0,0), (1,0), (0,1), (1,1) are the four corners of the unit square and can be represented as:

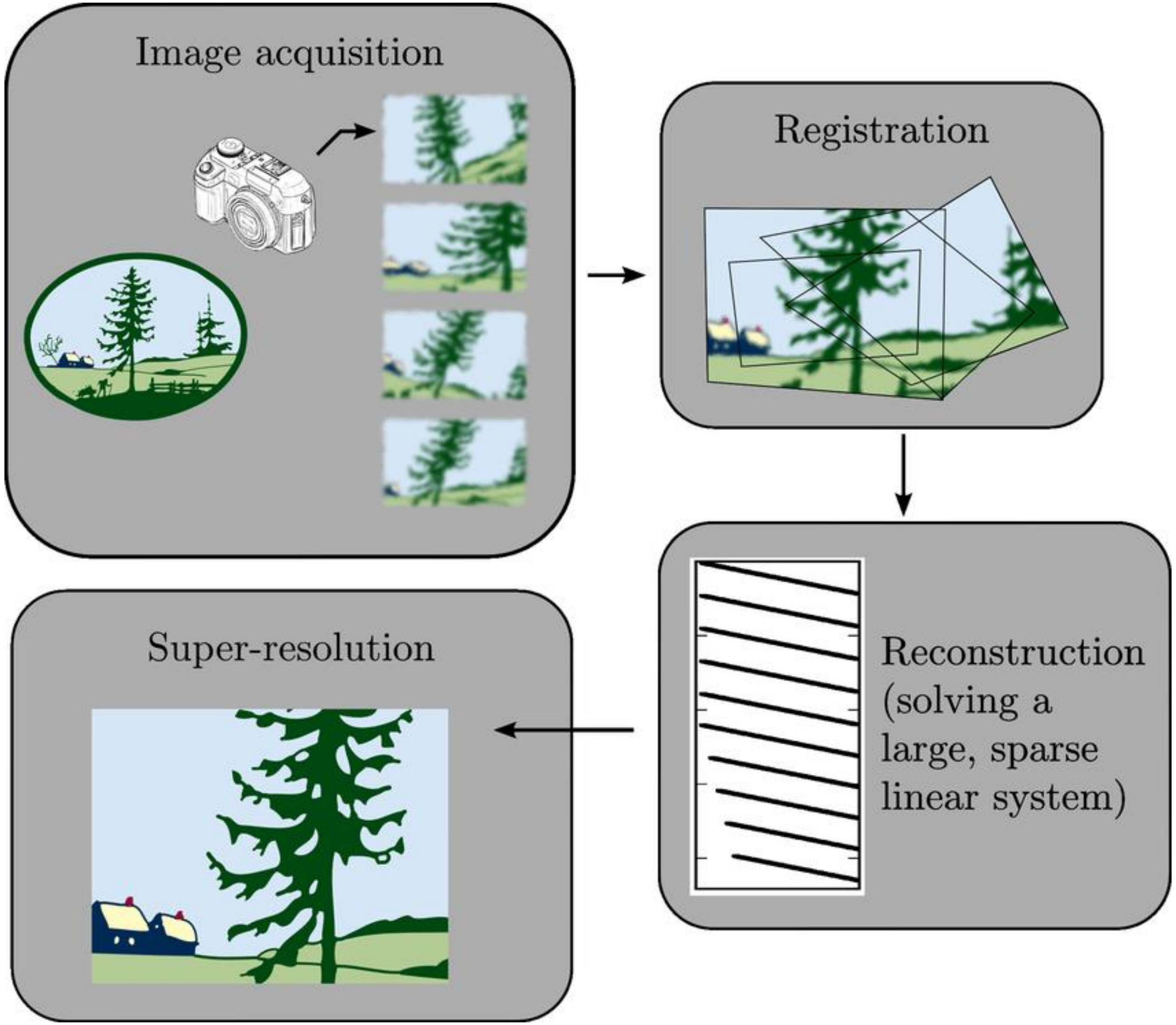
1. $f(0,0) = p(0,0) = a_{00},$
2. $f(1,0) = p(1,0) = a_{00} + a_{10} + a_{20} + a_{30},$
3. $f(0,1) = p(0,1) = a_{00} + a_{01} + a_{02} + a_{03},$
4. $f(1,1) = p(1,1) = \sum_{i=0}^{3} \sum_{j=0}^{3} a_{ij}.$

This requires the determination of 16 coefficients of p(x,y) leading to the creation of interpolated surface of 2D image.

Bicubic interpolation works well at high resolution, but not beyond 24 MP super high resolution. This is because addition of more pixels requires the preservation of more details. Too much upscaling can be go wrong with bicubic interpolation. There are now super resolution techniques like **SRCNN (Super Resolution Convolutional Neural Networks)** or **SRGAN (Super Resolution Generative Adversarial Networks)** that do a better job at preserving details with sharpness.

Super Resolution

- Creation of a high resolution image from a sequence of low resolution images

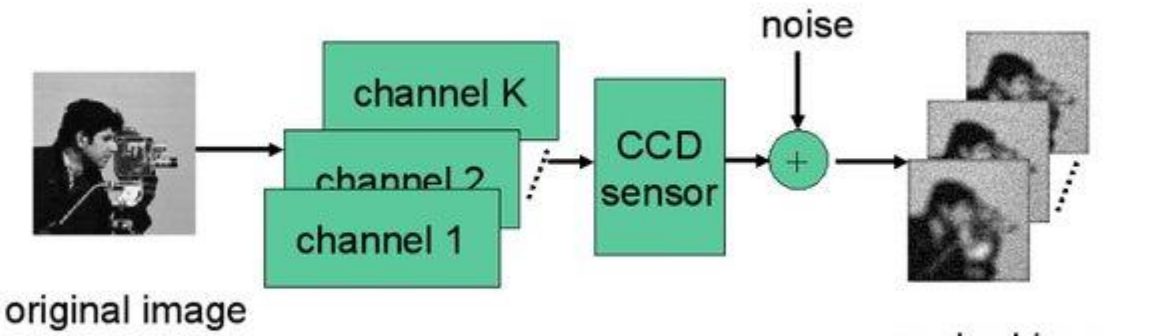


Super Resolution

Super-resolution imaging

Super-resolution (SR) is the process of combining a sequence of low resolution images in order to produce a higher resolution image or sequence.

We assume a similar acquisition model as in the case of multichannel blind deconvolution. However for SR, the original image undergoes three degradations during the measurement: blurring, resolution decimation, and corruption by noise.



$$D([u * h_k](x, y)) + n_k(x, y) = z_k(x, y)$$

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).

Chapter 2

End of Lecture 2

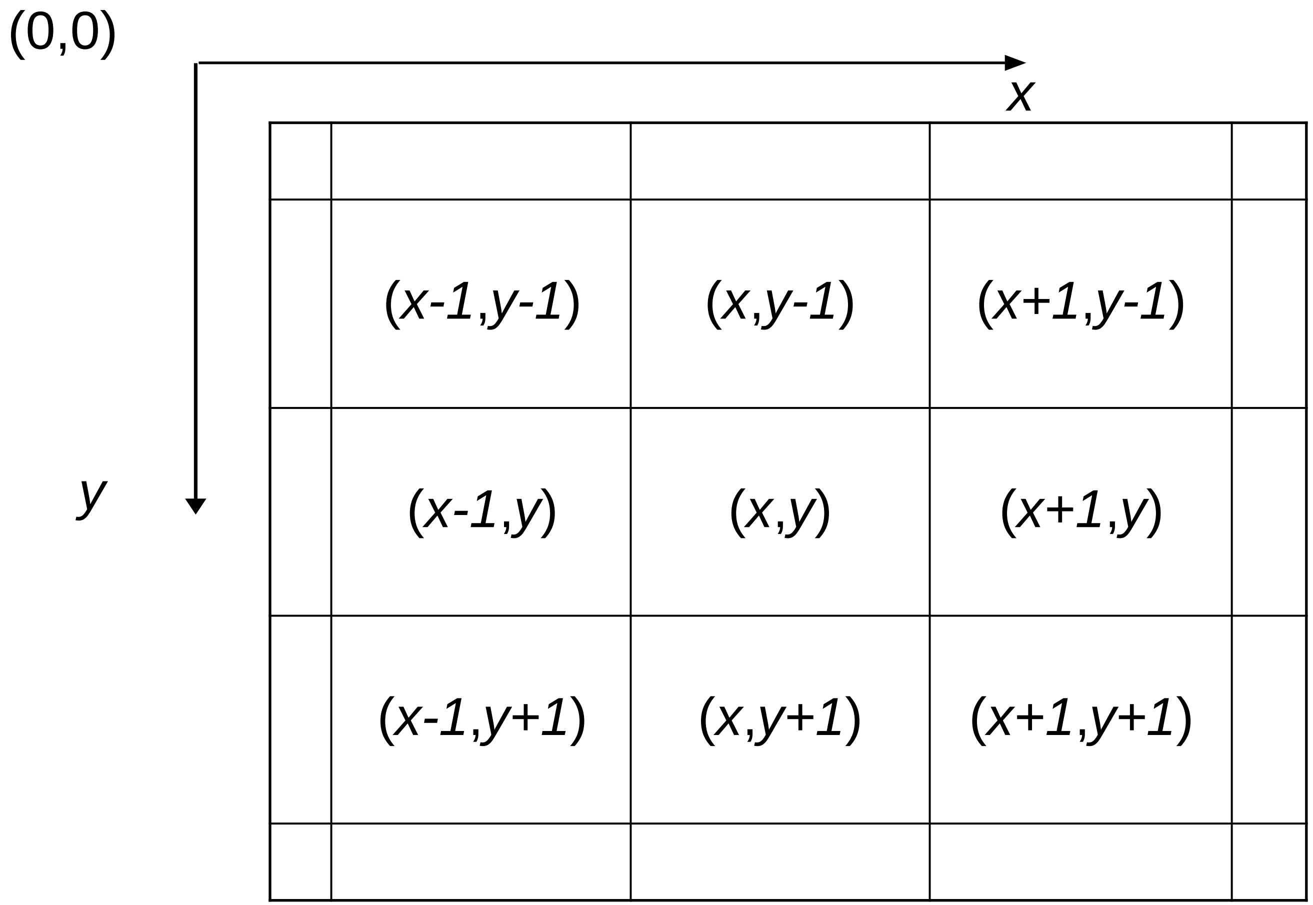
Chapter 2

Lecture 3: Image Data Handling-II

Contents

- Relationships between pixels
- Introduction to Mathematical Tools used in Digital Image Processing

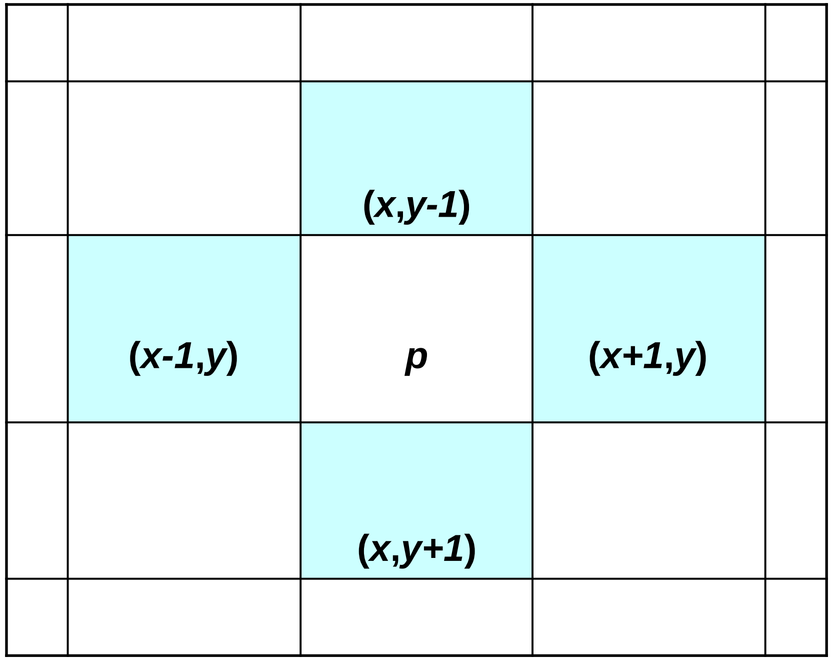
Basic Relationship of Pixels



Conventional indexing method

Neighbors of a Pixel

Neighborhood relation is used to tell adjacent pixels. It is useful for analyzing regions.



4-neighbors of *p*:

$$N_4(p) = \left\{ \begin{array}{c} (x-1,y) \\ (x+1,y) \\ (x,y-1) \\ (x,y+1) \end{array} \right\}$$

4-neighborhood relation considers only vertical and horizontal neighbors.

Note: $q \in N_4(p)$ implies $p \in N_4(q)$

Neighbors of a Pixel

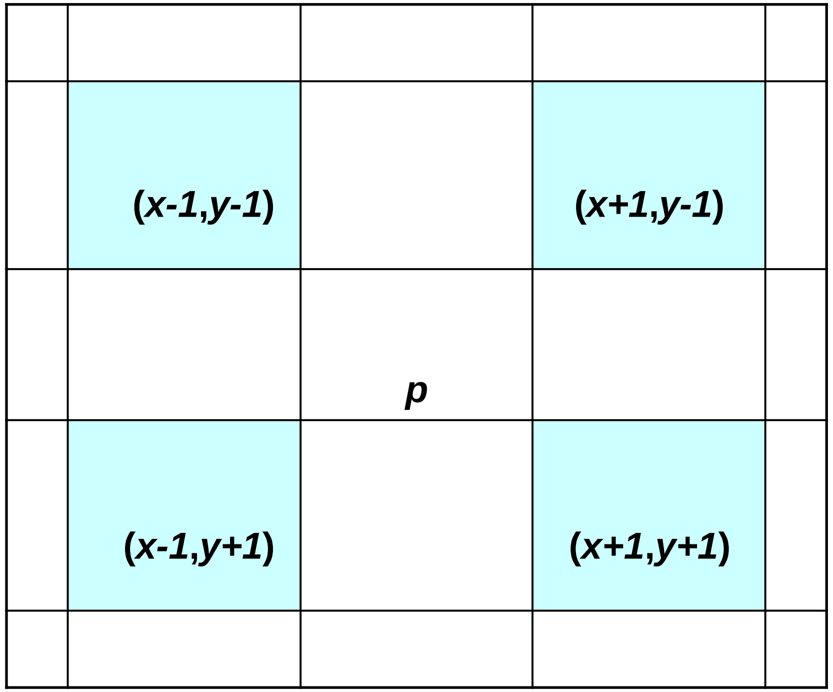
	(x-1,y-1)	*(x,y-1)*	*(x+1,y-1)*	
	(x-1,y)	*p*	*(x+1,y)*	
	(x-1,y+1)	*(x,y+1)*	*(x+1,y+1)*	

8-neighbors of *p*:

$$N_8(p) = \left\{ \begin{array}{c} (x-1,y-1) \\ (x,y-1) \\ (x+1,y-1) \\ (x-1,y) \\ (x+1,y) \\ (x-1,y+1) \\ (x,y+1) \\ (x+1,y+1) \end{array} \right\}$$

8-neighborhood relation considers all neighbor pixels.

Neighbors of a Pixel



Diagonal neighbors of *p*:

$$N_D(p) = \left\{ \begin{array}{c} (x-1,y-1) \\ (x+1,y-1) \\ (x-1,y+1) \\ (x+1,y+1) \end{array} \right\}$$

Diagonal -neighborhood relation considers only diagonal neighbor pixels.

Connectivity

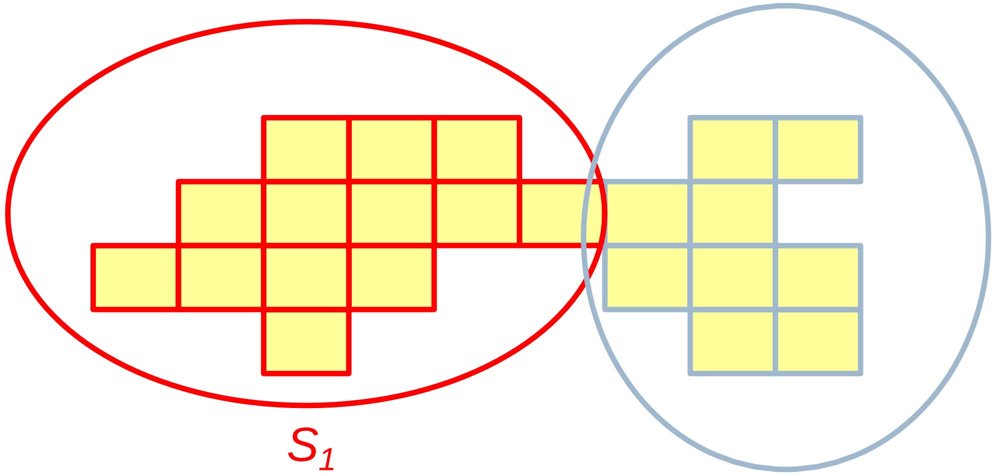
Connectivity is adapted from neighborhood relation. Two pixels are connected if they are in the same class (i.e. the same color or the same range of intensity) and they are neighbors of one another.

For p and q from the same class

- 4-connectivity: p and q are 4-connected if $q \in N_4(p)$
- 8-connectivity: p and q are 8-connected if $q \in N_8(p)$
- mixed-connectivity (m-connectivity):
  p and q are m-connected if $q \in N_4(p)$ or
  $q \in N_D(p)$ and $N_4(p) \cap N_4(q) = \emptyset$

Adjacency

A pixel *p* is *adjacent* to pixel *q* is they are connected. Two image subsets S_1 and S_2 are adjacent if some pixel in S_1 is adjacent to some pixel in S_2



We can define type of adjacency: 4-adjacency, 8-adjacency or m-adjacency depending on type of connectivity.

Adjacency and Connectivity

- *Adjacency*- Two pixels p and q are adjacent if q is in N(p) where N(p) is the neighborhood of p and they have closely related pixel values. Three common definitions of neighborhood are

(1) 4-adjacency. If p=(x,y), values are similar, but q is

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

(2) 8-adjacency. It is possible for q to be one of the diagonal points (x-1,y-1),(x-1,y+1),(x+1,y-1),(x+1,x+1).

(3) m-adjacency. Either q is 4-adjacent to p, or q is a diagonal point and the intersection of the four neighborhood of p and that of q have no similar pixel values.

Path

A ***path*** from pixel p at (x,y) to pixel q at (s,t) is a sequence of distinct pixels:

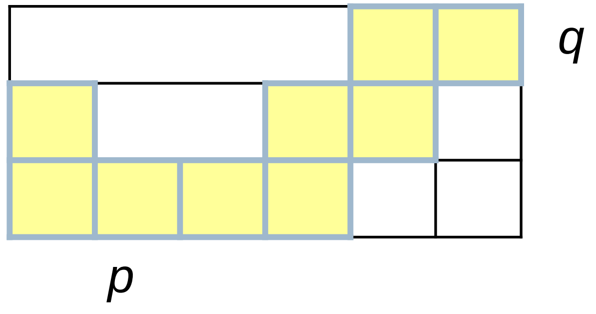
$$(x_0,y_0), (x_1,y_1), (x_2,y_2),\ldots, (x_n,y_n)$$

such that

$$(x_0,y_0) = (x,y) \text{ and } (x_n,y_n) = (s,t)$$

and

$$(x_i,y_i) \text{ is adjacent to } (x_{i-1},y_{i-1}), \quad i = 1,\ldots,n$$

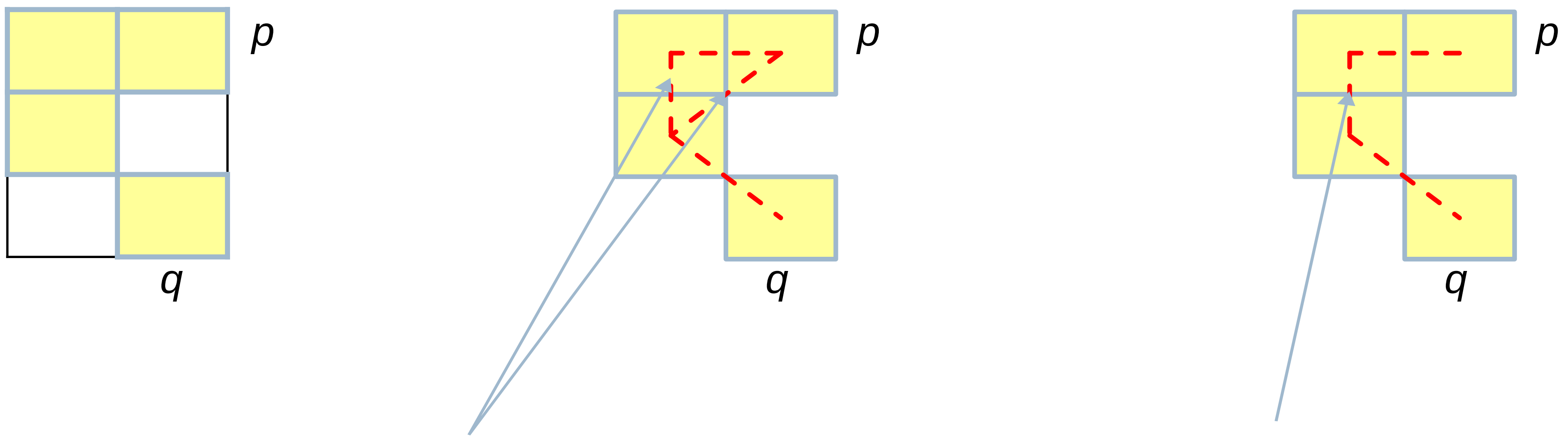


We can define type of path: 4-path, 8-path or m-path depending on type of adjacency.

Path

8-path

m-path



8-path from *p* to *q*
results in some ambiguity

m-path from *p* to *q*
solves this ambiguity

Distance

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

- $D(p,q) \geq 0$ $\quad$ ($D(p,q) = 0$ if and only if $p = q$)
- $D(p,q) = D(q,p)$
- $D(p,z) \leq D(p,q) + D(q,z)$

Example: Euclidean distance

$$D_e(p,q) = \sqrt{(x-s)^2 + (y-t)^2}$$

Distance

D_4-distance (*city-block distance*) is defined as

$$D_4(p,q) = |x-s| + |y-t|$$

		2		
	2	1	2	
2	1	0	1	2
	2	1	2	
		2		

Pixels with $D_4(p) = 1$ is 4-neighbors of *p*.

Introduction to Mathematical Operations in DIP

- **Array vs. Matrix Operation**

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

Array product operator

$$A \;.*\; B = \begin{bmatrix} a_{11}b_{11} & a_{12}b_{12} \\ a_{21}b_{21} & a_{22}b_{22} \end{bmatrix}$$

Array product

Matrix product operator

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

Matrix product

Introduction to Mathematical Operations in DIP

- **Linear vs. Nonlinear Operation**

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

$$H\left[a_i f_i(x,y)+a_j f_j(x,y)\right]$$

$$=H\left[a_i f_i(x,y)\right]+H\left[a_j f_j(x,y)\right]$$

Additivity

$$=a_i H\left[f_i(x,y)\right]+a_j H\left[f_j(x,y)\right]$$

Homogeneity

$$=a_i g_i(x,y)+a_j g_j(x,y)$$

H is said to be a **linear operator;**

H is said to be a **nonlinear operator** if it does not meet the above qualification.

Arithmetic Operations

- Arithmetic operations between images are array operations. The four arithmetic operations are denoted as

$$s(x,y) = f(x,y) + g(x,y)$$

$$d(x,y) = f(x,y) - g(x,y)$$

$$p(x,y) = f(x,y) \times g(x,y)$$

$$v(x,y) = f(x,y) \div g(x,y)$$

Example: Addition of Noisy Images for Noise Reduction

Noiseless image: f(x,y)

Noise: n(x,y) (at every pair of coordinates (x,y), the noise is uncorrelated and has zero average value)

Corrupted image: g(x,y)

$$g(x,y) = f(x,y) + n(x,y)$$

Reducing the noise by adding a set of noisy images, $\{g_i(x,y)\}$

$$\bar{g}(x,y) = \frac{1}{K}\sum_{i=1}^{K} g_i(x,y)$$

Example: Addition of Noisy Images for Noise Reduction

$$\bar{g}(x,y) = \frac{1}{K}\sum_{i=1}^{K} g_i(x,y)$$

$$E\{\bar{g}(x,y)\} = E\left\{\frac{1}{K}\sum_{i=1}^{K} g_i(x,y)\right\}$$

$$= E\left\{\frac{1}{K}\sum_{i=1}^{K}[f(x,y)+n_i(x,y)]\right\}$$

$$= f(x,y) + E\left\{\frac{1}{K}\sum_{i=1}^{K} n_i(x,y)\right\}$$

$$= f(x,y)$$

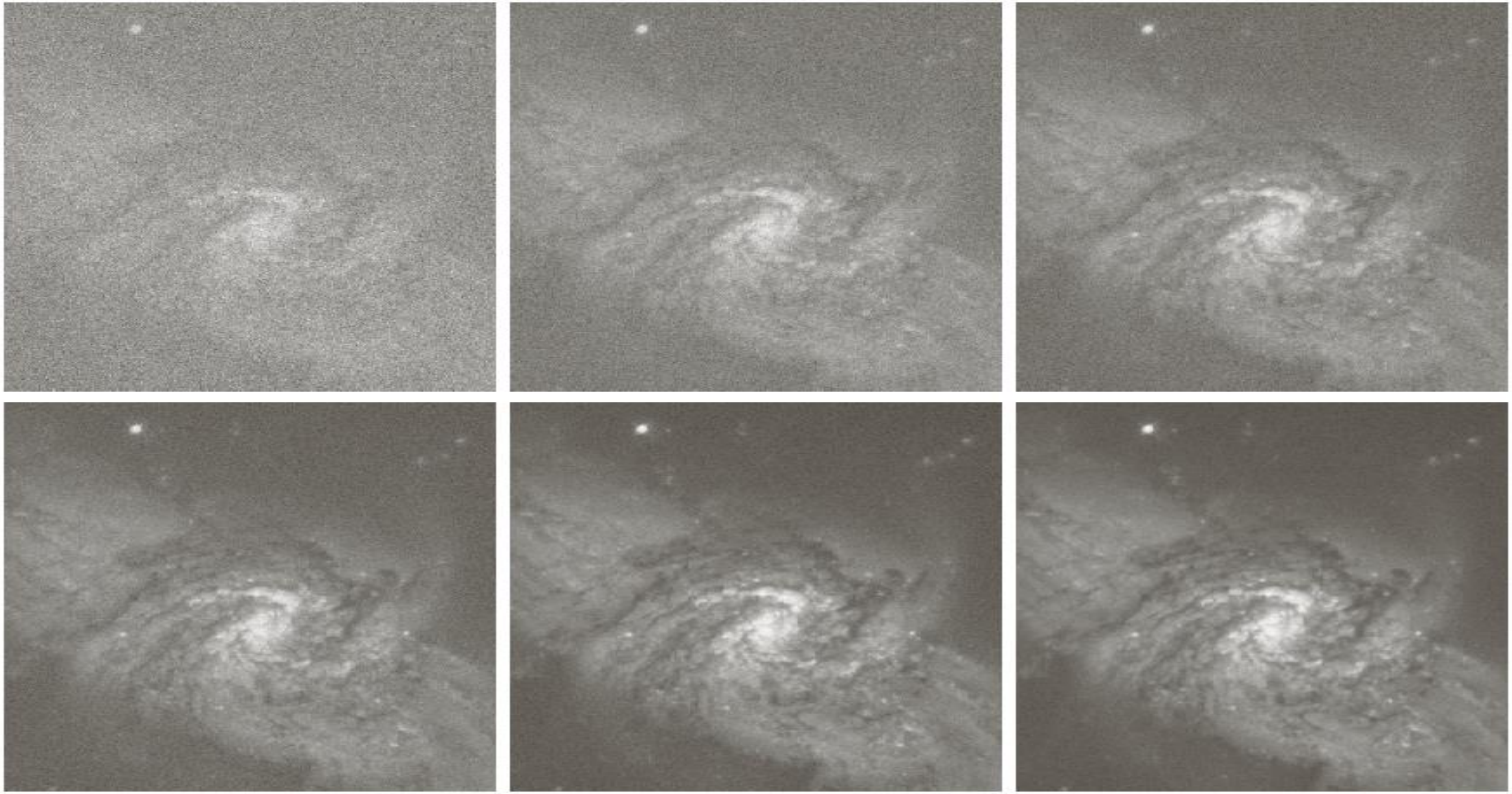
$$\sigma^2_{\bar{g}(x,y)} = \sigma^2_{\frac{1}{K}\sum_{i=1}^{K} g_i(x,y)}$$

$$= \sigma^2_{\frac{1}{K}\sum_{i=1}^{K} n_i(x,y)} = \frac{1}{K}\sigma^2_{n(x,y)}$$

Example: Addition of Noisy Images for Noise Reduction

- In astronomy, imaging under very low light levels frequently causes sensor noise to render single images virtually useless for analysis.

- In astronomical observations, similar sensors for noise reduction by observing the same scene over long periods of time. Image averaging is then used to reduce the noise.



a	b	c
d	e	f

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.)

An Example of Image Subtraction: Mask Mode Radiography

Mask h(x,y): an X-ray image of a region of a patient's body

Live images f(x,y): X-ray images captured at TV rates after injection of the contrast medium

Enhanced detail g(x,y)

$$g(x,y) = f(x,y) - h(x,y)$$

The procedure gives a movie showing how the contrast medium propagates through the various arteries in the area being observed.

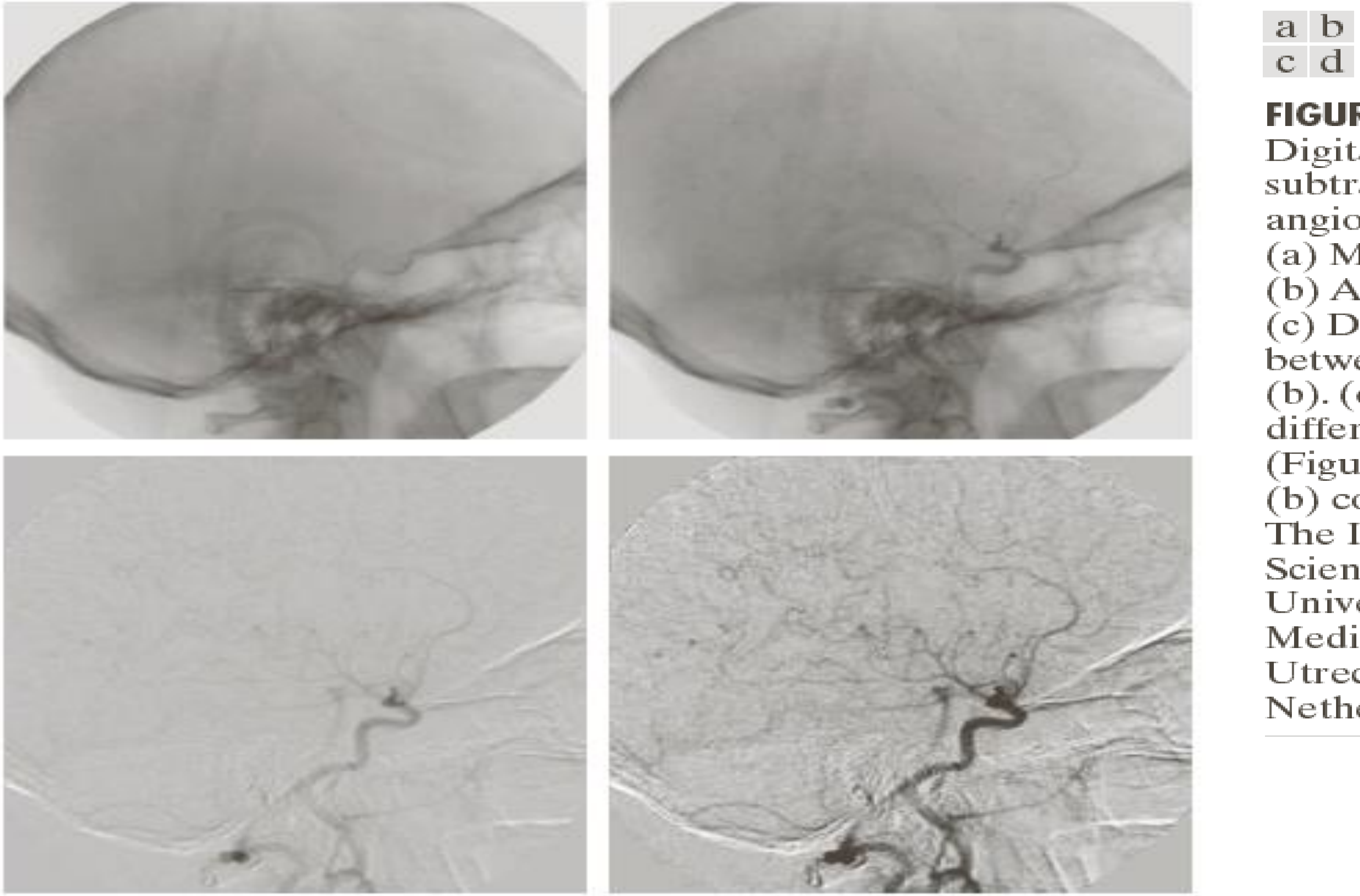
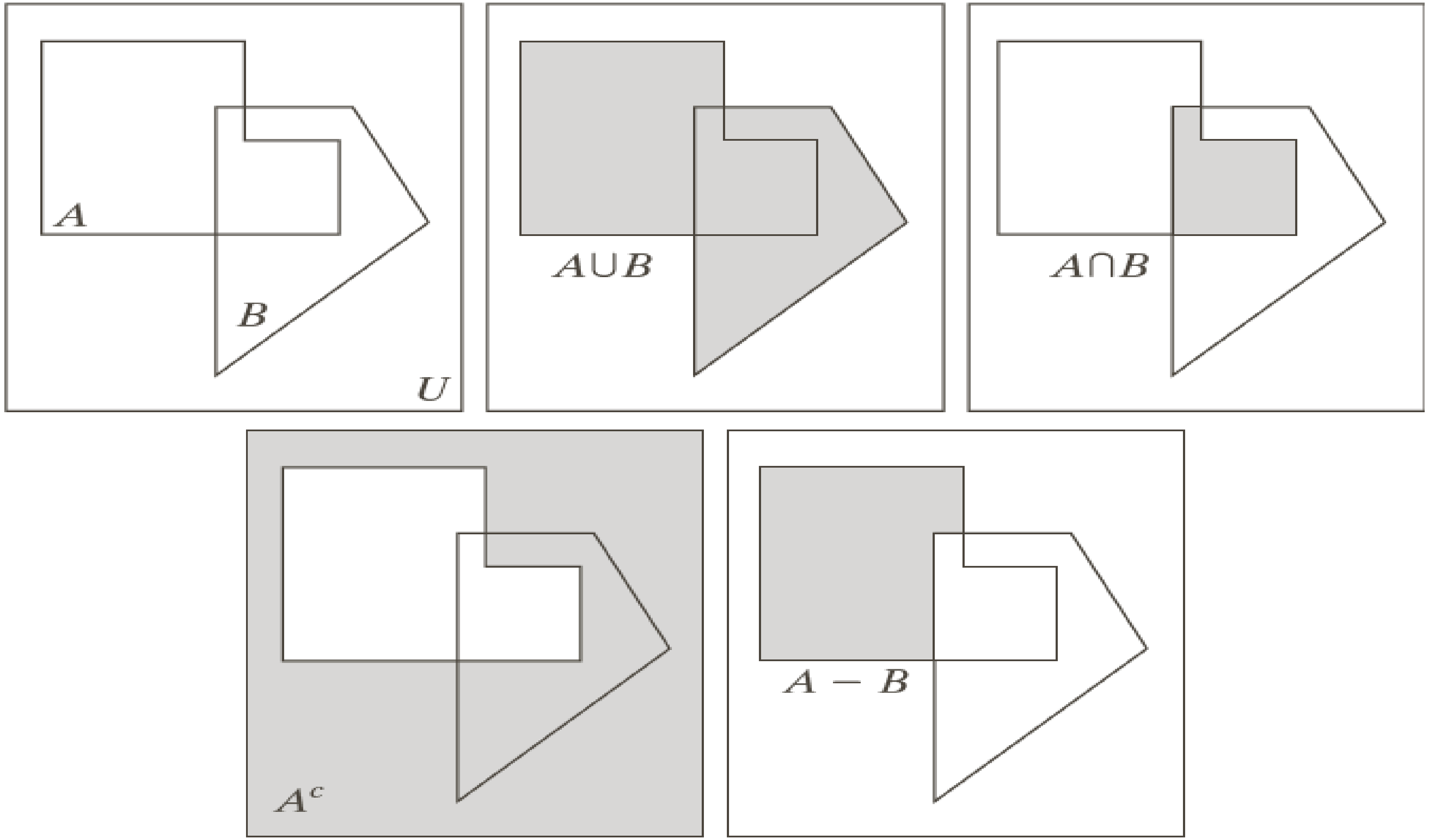


FIGURE 2.28 Digital subtraction angiography. (a) Mask image. (b) A live image. (c) Difference between (a) and (b). (d) Enhanced difference image. (Figures (a) and (b) courtesy of The Image Sciences Institute, University Medical Center, Utrecht, The Netherlands.)

An Example of Image Multiplication

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.)

Set and Logical 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 and Logical Operations

- Let A be the elements of a gray-scale image

  The elements of A are triplets of the form (x, y, z), where x and y are spatial coordinates and z denotes the intensity at the point (x, y).

$$A = \{(x, y, z) \mid z = f(x, y)\}$$

- The complement of A is denoted A^c

$$A^c = \{(x, y, K - z) \mid (x, y, z) \in A\}$$

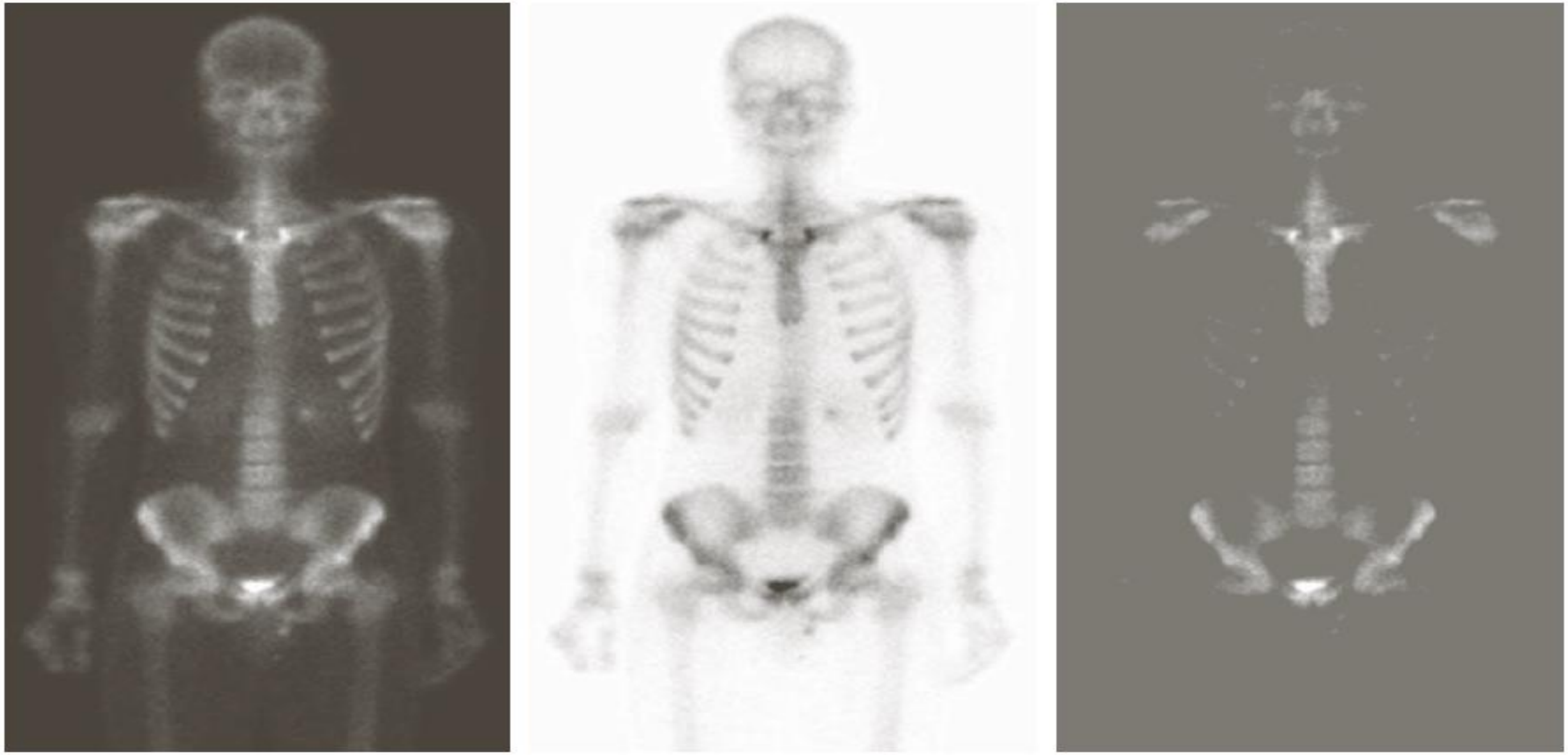
$$K = 2^k - 1; k \text{ is the number of intensity bits used to represent } z$$

Set and Logical Operations

- The union of two gray-scale images (sets) A and B is defined as the set

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

Set and Logical Operations



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.)

Set and Logical Operations

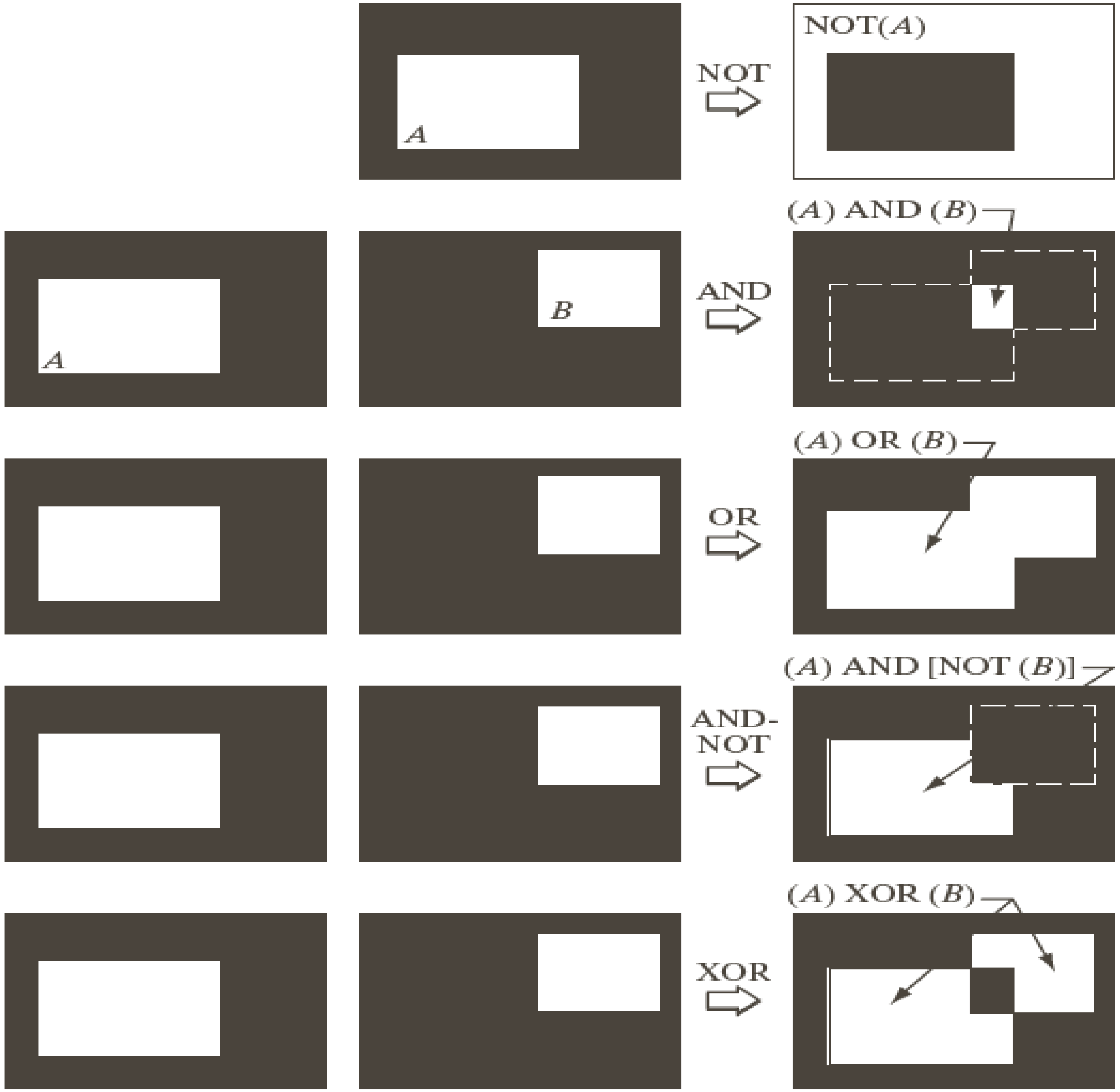


FIGURE 2.33 Illustration of logical operations involving foreground (white) pixels. Black represents binary 0s and white binary 1s. The dashed lines are shown for reference only. They are not part of the result.

Spatial Operations

- Single-pixel operations

  Alter the values of an image's pixels based on the intensity.

$$s = T(z)$$

e.g.,

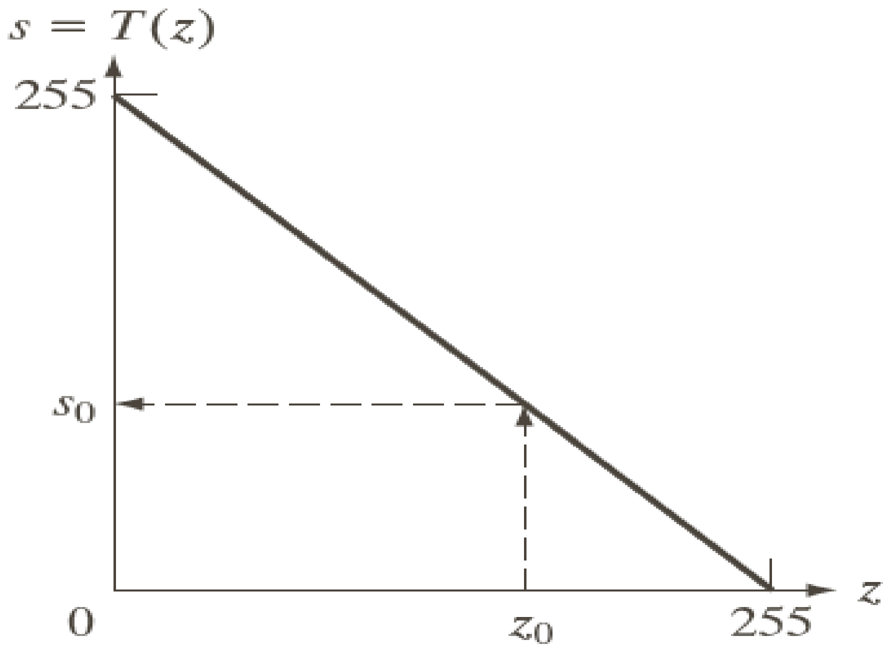
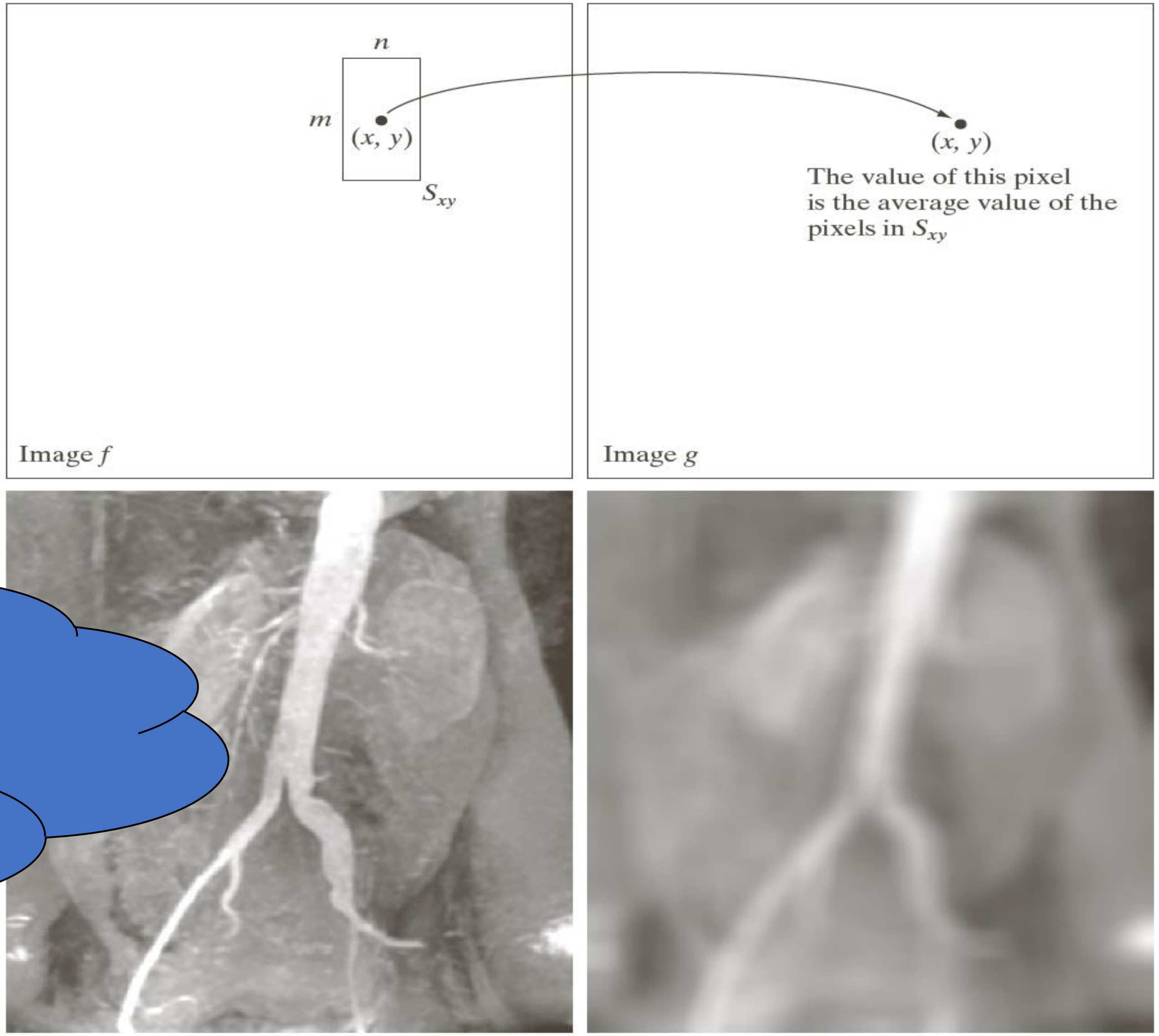


FIGURE 2.34 Intensity transformation function used to obtain the negative of an 8-bit image. The dashed arrows show transformation of an arbitrary input intensity value z_0 into its corresponding output value s_0.

Spatial Operations

- Neighborhood operations



The value of this pixel is determined by a specified operation involving the pixels in the input image with coordinates in S_{xy}

Geometric Spatial Transformations

- Geometric transformation (rubber-sheet transformation)
  — A spatial transformation of coordinates

$$(x, y) = T\{(v, w)\}$$

  — intensity interpolation that assigns intensity values to the spatially transformed pixels.

- Affine transform

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

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

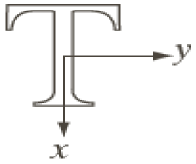
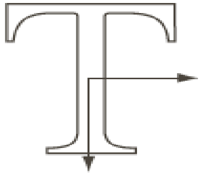
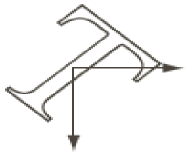
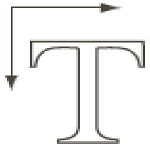
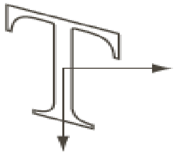
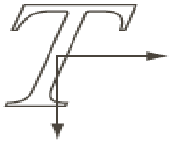
Transformation Name	Affine Matrix, T	Coordinate Equations	Example
Identity	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = v$ $y = w$	
Scaling	$\begin{bmatrix} c_x & 0 & 0 \\ 0 & c_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = c_x v$ $y = c_y w$	
Rotation	$\begin{bmatrix} \cos\theta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

Image Registration

- Input and output images are available but the transformation function is unknown.

  Goal: estimate the transformation function and use it to register the two images.

- One of the principal approaches for image registration is to use ***tie points*** (also called ***control points***)

➢ The corresponding points are known precisely in the input and output (**reference**) images.

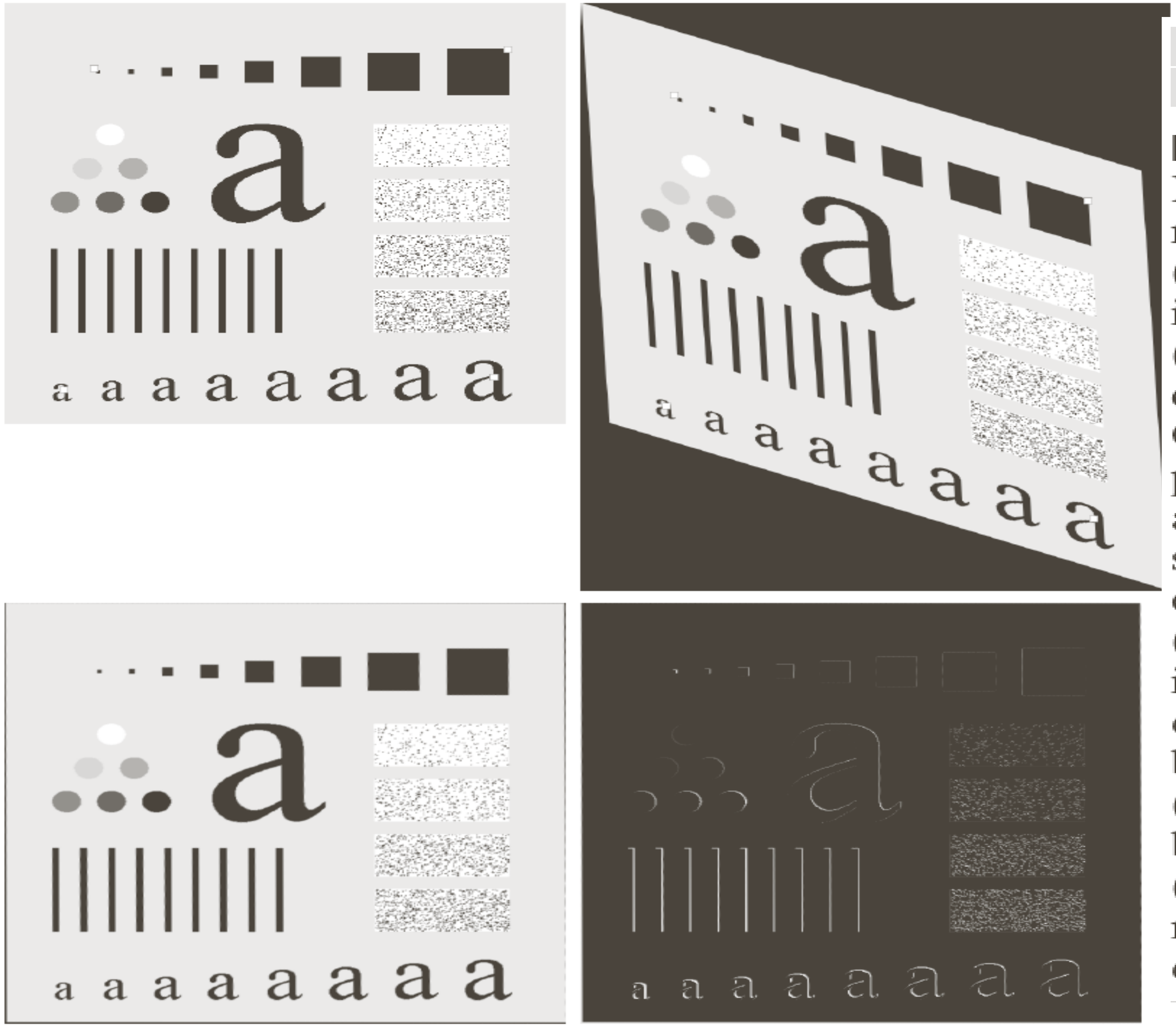
Image Registration

- A simple model based on bilinear approximation:

$$\begin{cases} x = c_1 v + c_2 w + c_3 vw + c_4 \\ y = c_5 v + c_6 w + c_7 vw + c_8 \end{cases}$$

Where (v, w) and (x, y) are the coordinates of tie points in the input and reference images.

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 Transform

- A particularly important class of 2-D linear transforms, denoted T(u, v)

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

where $f(x,y)$ is the input image,

$r(x,y,u,v)$ is the *forward transformation kernel*,

variables u and v are the transform variables,

u = 0, 1, 2, ..., M-1 and v = 0, 1, ..., N-1.

Image Transform

- Given T(u, v), the original image f(x, y) can be recoverd using the inverse tranformation of T(u, v).

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

where $s(x,y,u,v)$ is the *inverse transformation kernel*,

x = 0, 1, 2, ..., M-1 and y = 0, 1, ..., N-1.

Image Transform

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

Example: Image Denoising by Using DCT Transform

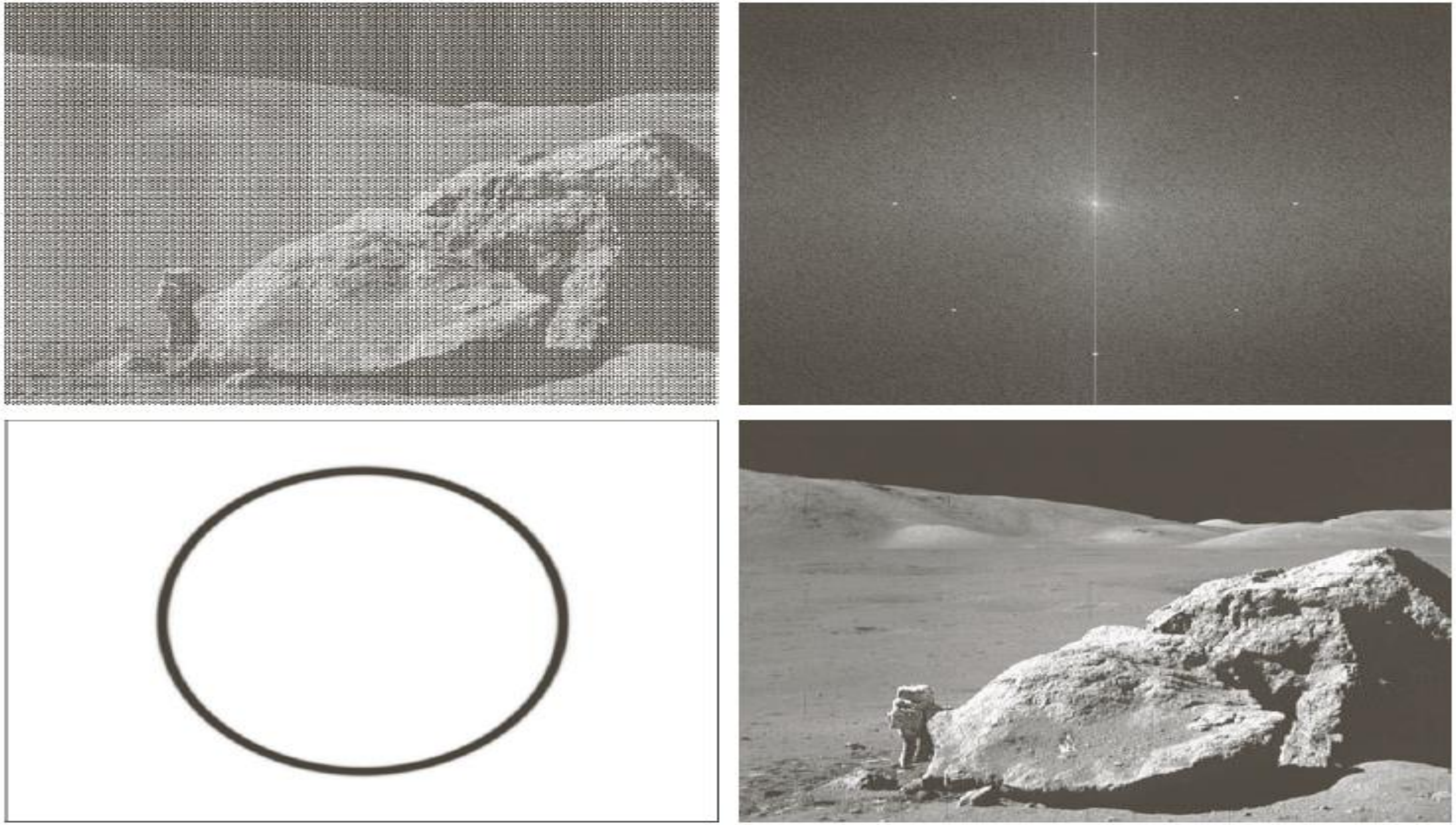


FIGURE 2.40
(a) Image corrupted by sinusoidal interference. (b) Magnitude of the Fourier transform showing the bursts of energy responsible for the interference. (c) Mask used to eliminate the energy bursts. (d) Result of computing the inverse of the modified Fourier transform. (Original image courtesy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

Forward Transform Kernel

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

The kernel $r(x,y,u,v)$ is said to be SEPERABLE if

$$r(x,y,u,v)=r_1(x,u)r_2(y,v)$$

In addition, the kernel is said to be SYMMETRIC if $r_1(x,u)$ is functionally equal to $r_2(y,v)$, so that

$$r(x,y,u,v)=r_1(x,u)r_1(y,u)$$

Kernels for 2-D Fourier Transform

The *forward* kernel

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

Where $j = \sqrt{-1}$

The *inverse* kernel

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

2-D Fourier Transform

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

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

Probabilistic Methods

Let z_i, $i = 0, 1, 2, \ldots, L-1$, denote the values of all possible intensities in an $M \times N$ digital image. The probability, $p(z_k)$, of intensity level z_k occurring in a given image is estimated as

$$p(z_k) = \frac{n_k}{MN},$$

where n_k is the number of times that intensity z_k occurs in the image.

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

The mean (average) intensity is given by

$$m = \sum_{k=0}^{L-1} z_k \, p(z_k)$$

Probabilistic Methods

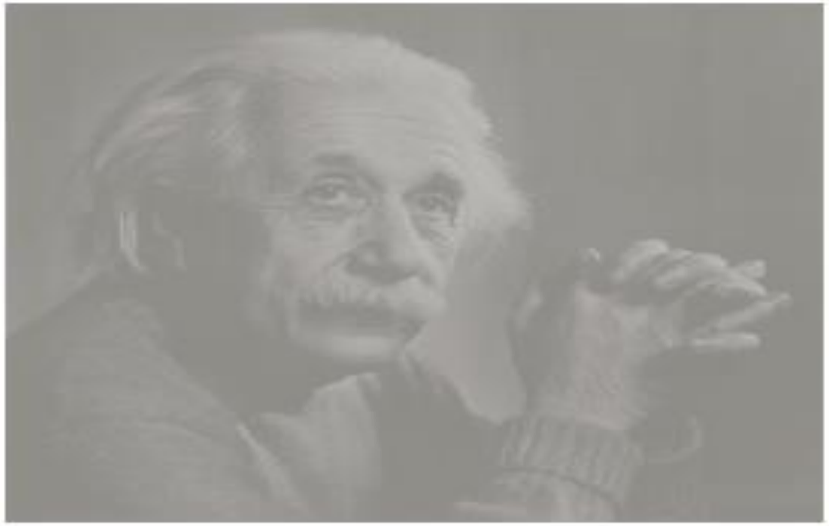
The variance of the intensities is given by

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

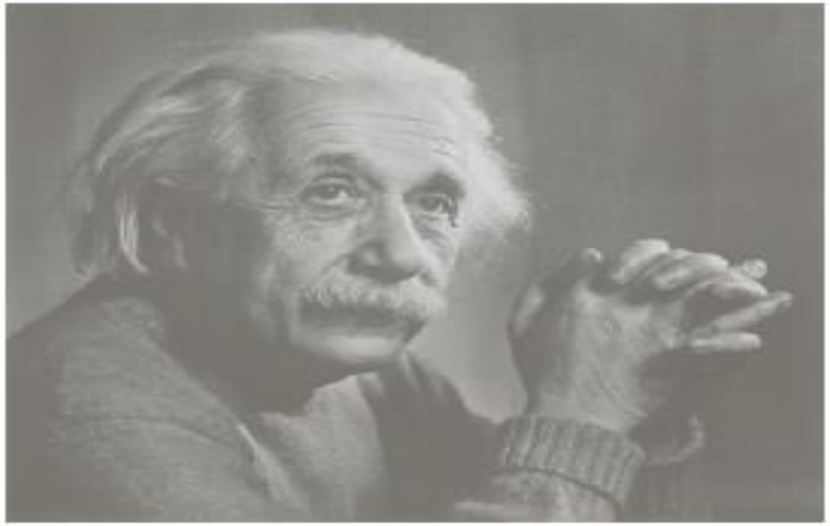
The n^{th} moment of the intensity variable z is

$$u_n(z) = \sum_{k=0}^{L-1} (z_k - m)^n p(z_k)$$

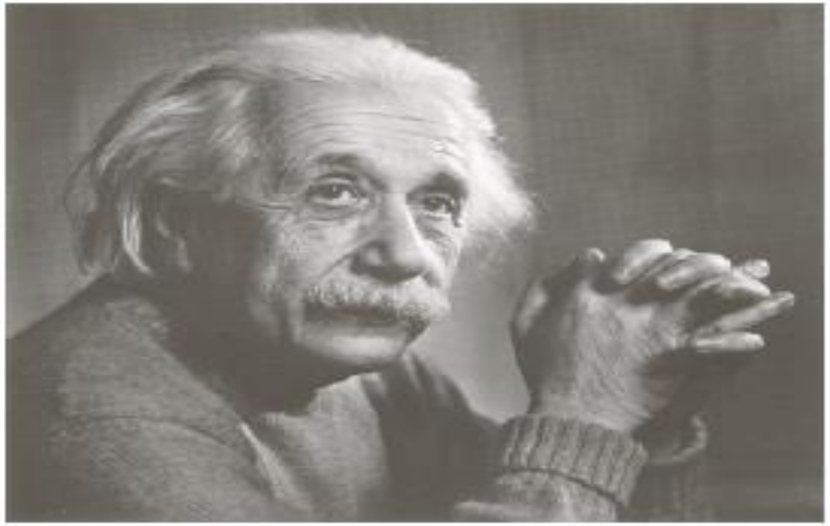
Example: Comparison of Standard Deviation Values



$\sigma = 14.3$



$\sigma = 31.6$



$\sigma = 49.2$

shutterstock.com · 1092234560

Thank You

End of Chapter 2