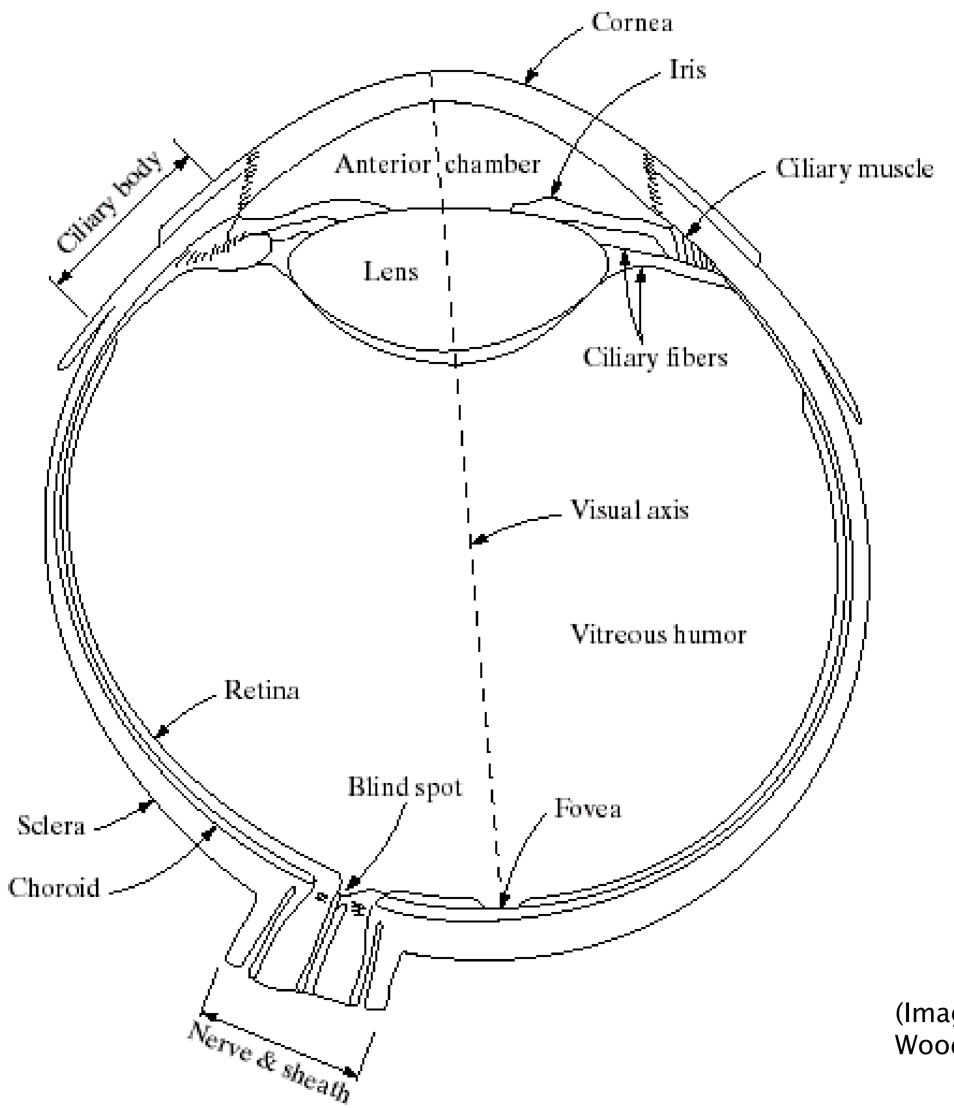
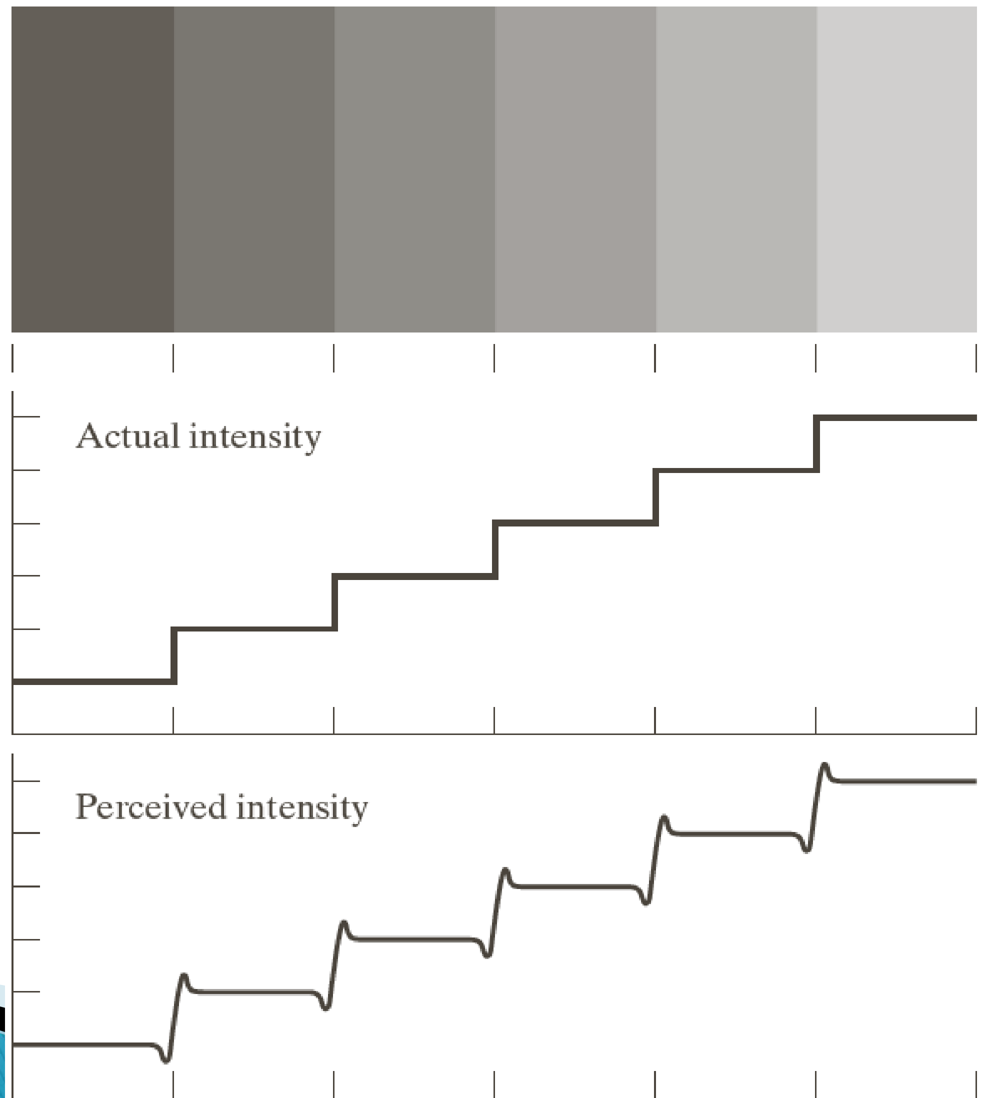
Digital Image Fundamentals

Simplified Diagram of a cross section of the human eye.

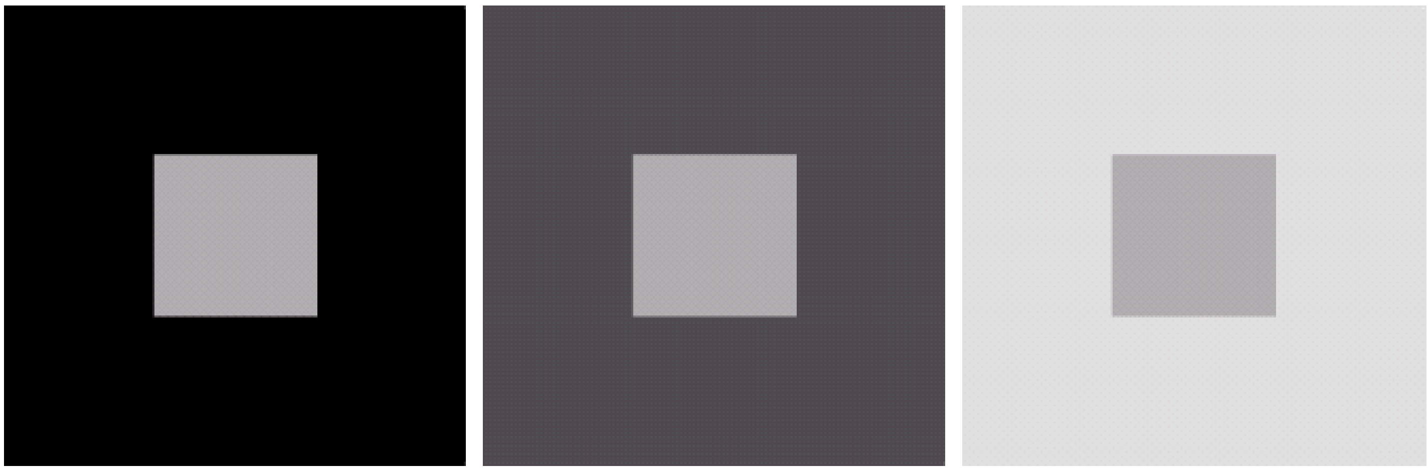


(Images from Rafael C. Gonzalez and Richard
Wood, Digital Image Processing, 2nd Edition.

Illustration of the Mach band effect



Simultaneous Contrast

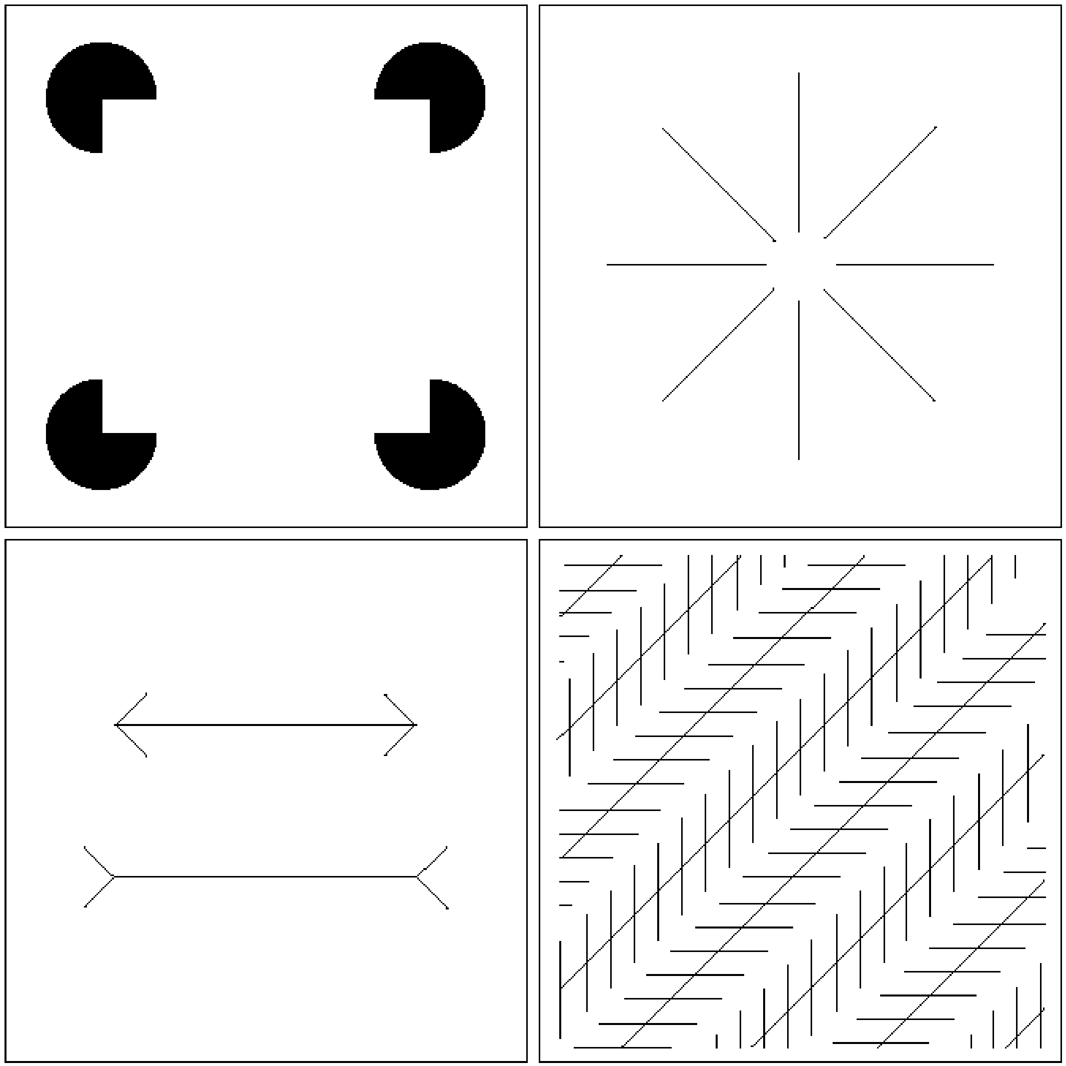


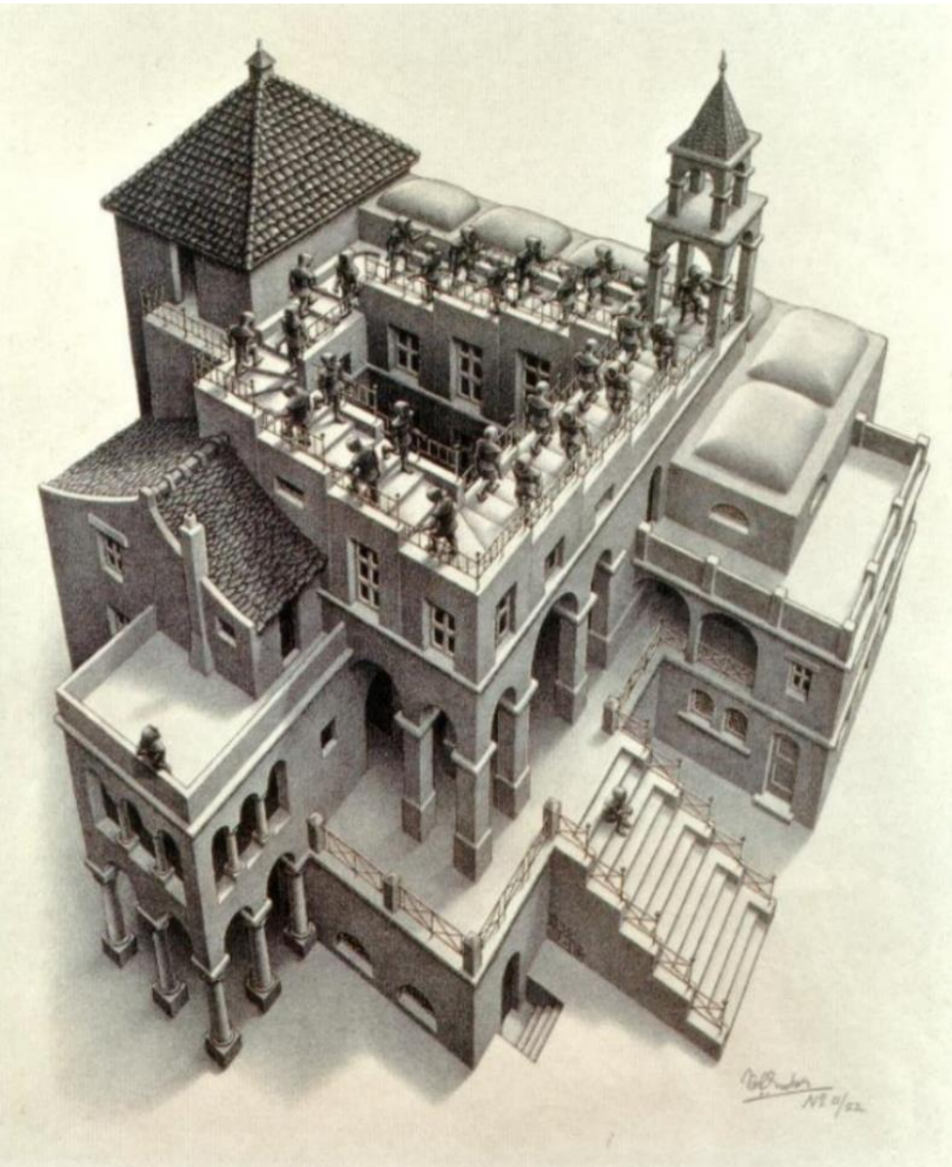
a b c

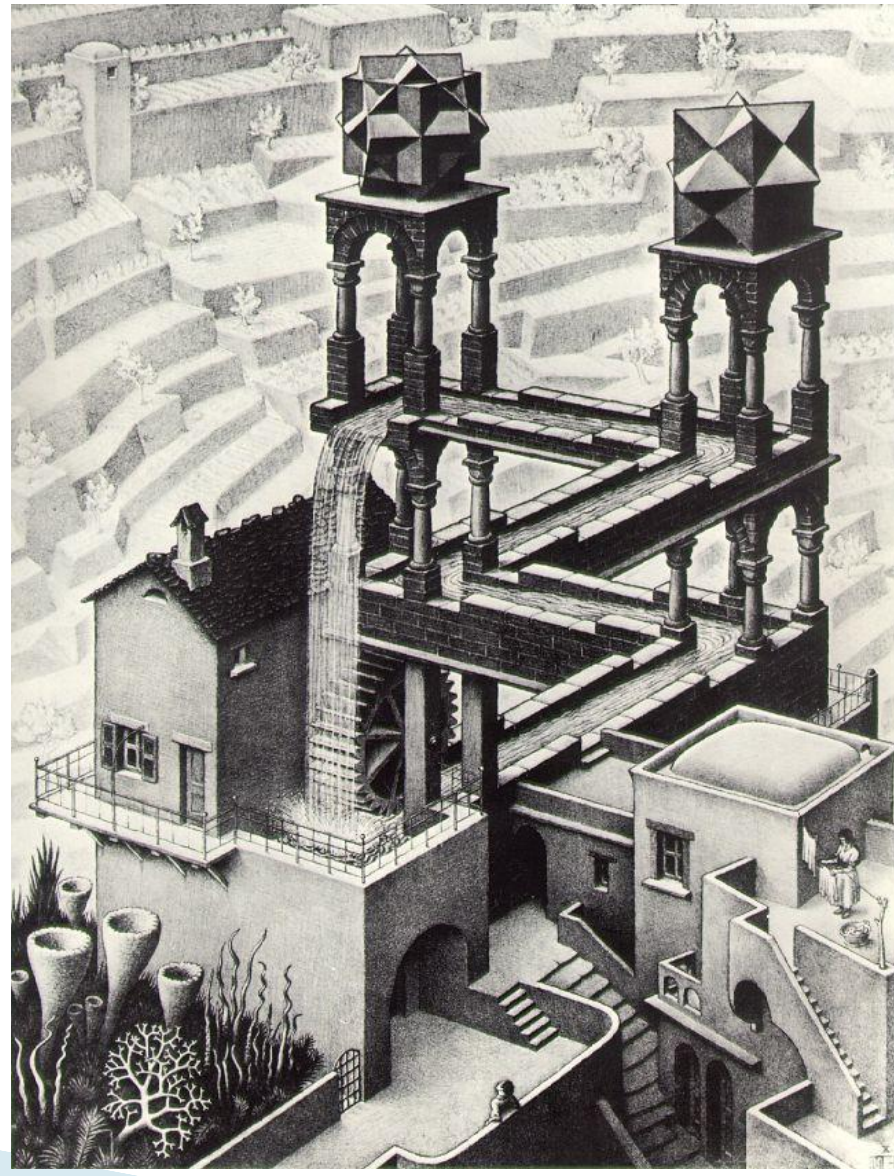
FIGURE 2.8 Examples of simultaneous contrast. All the inner squares have the same intensity, but they appear progressively darker as the background becomes lighter.

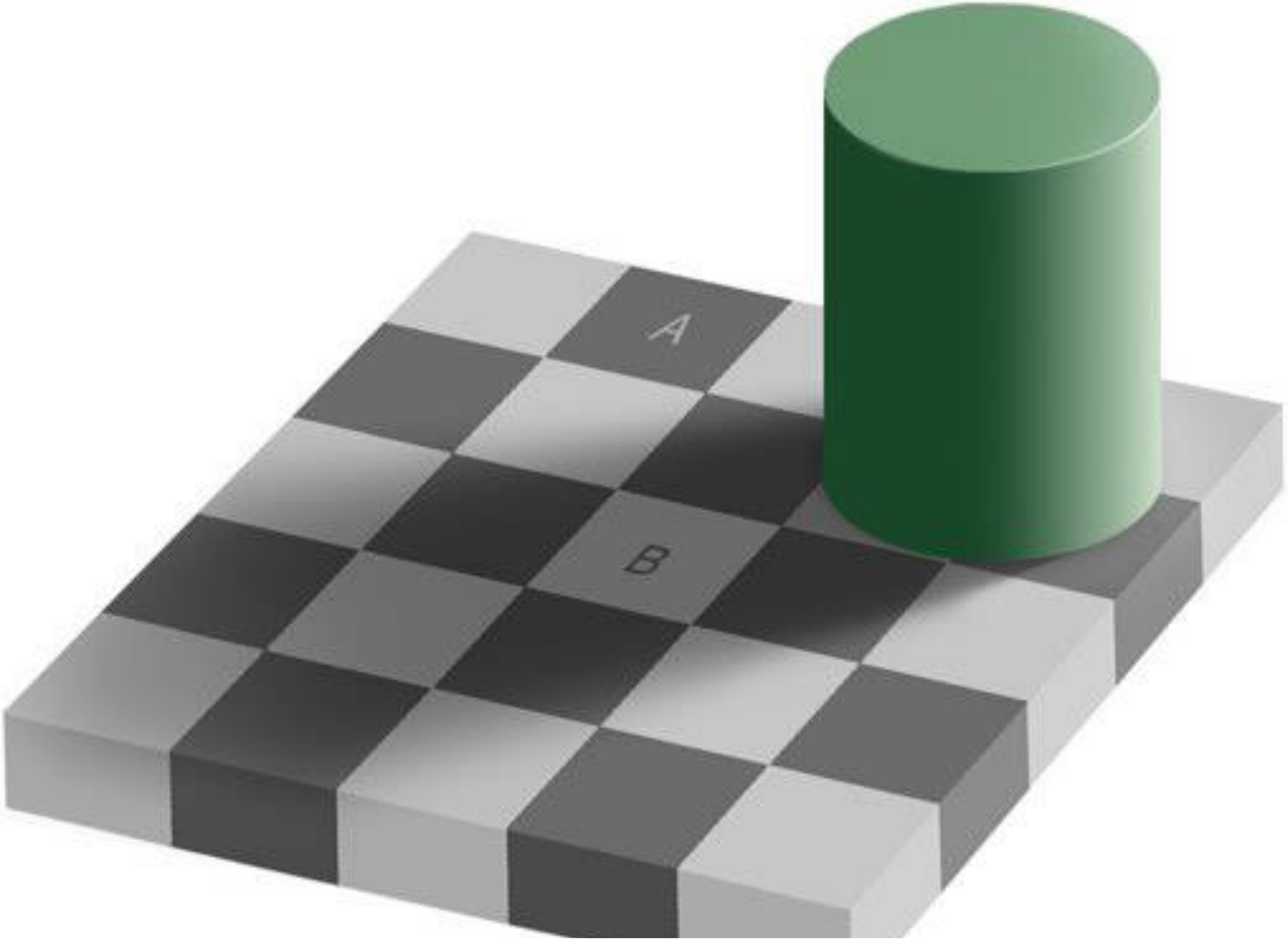
(Images from Rafael C. Gonzalez and Richard E. Wood, Digital Image Processing, 2nd Edition.

Optical illusion

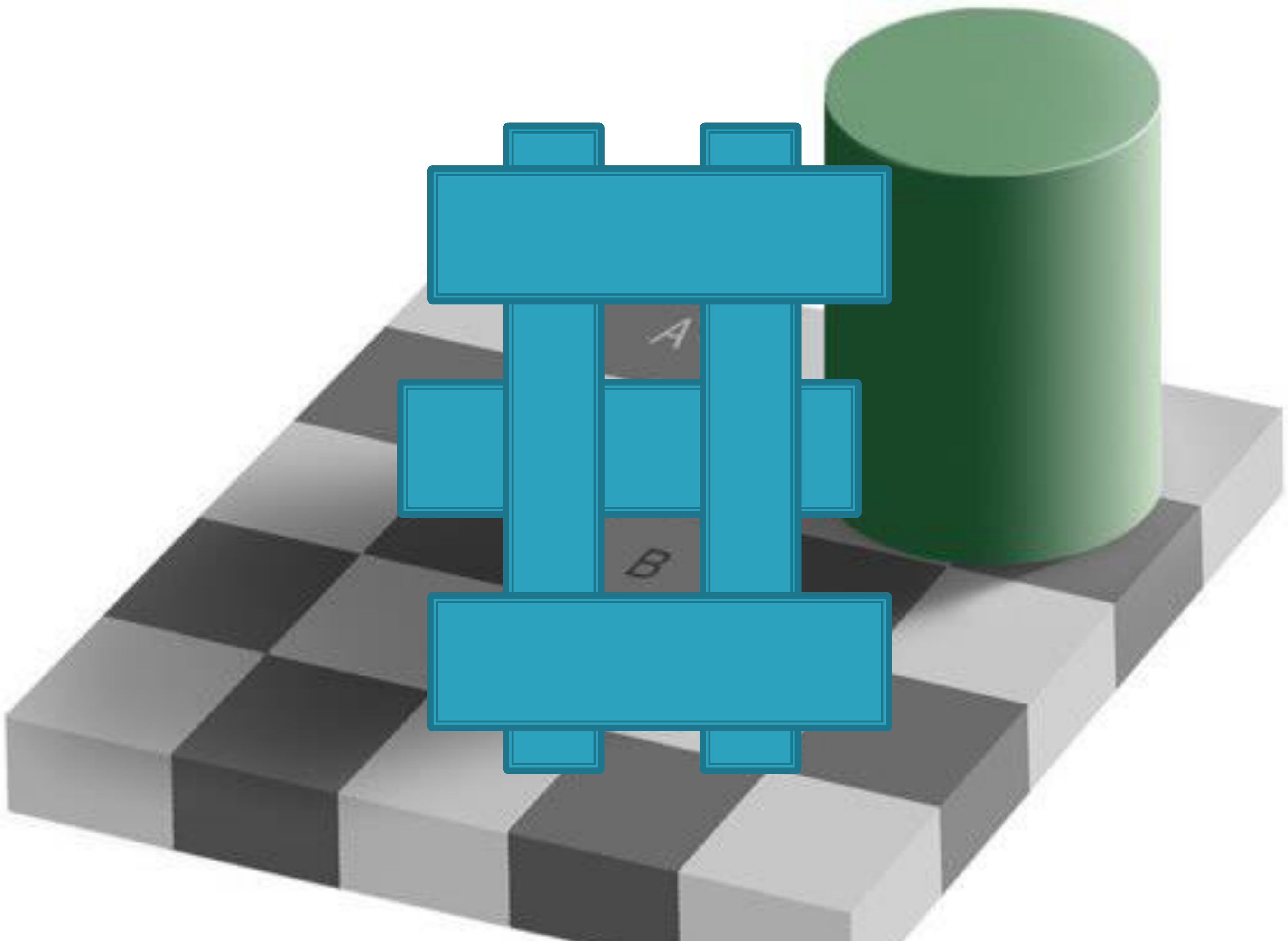




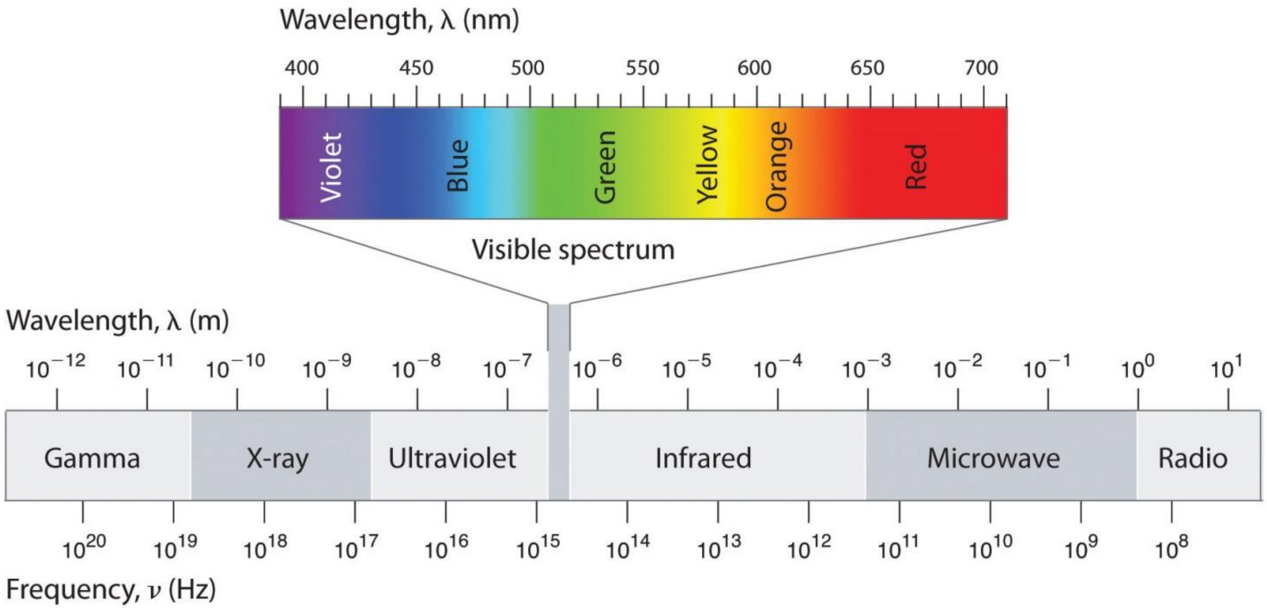




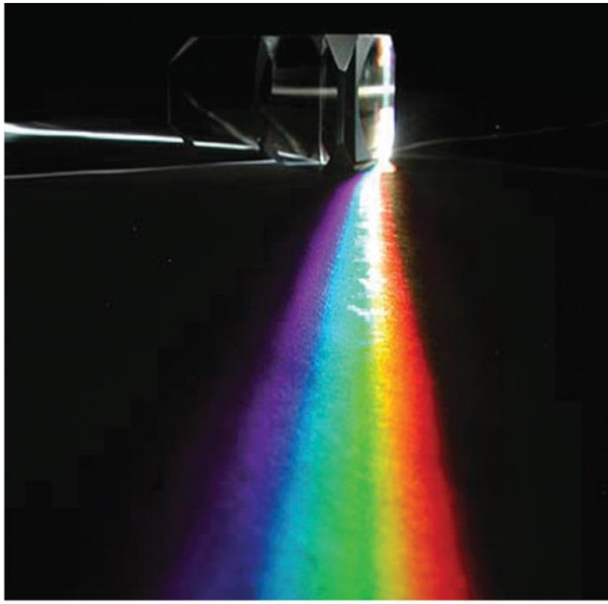
Are the squares above marked A & B the same color or different?



Proof: They are the same. The shadow causes an illusion of white.

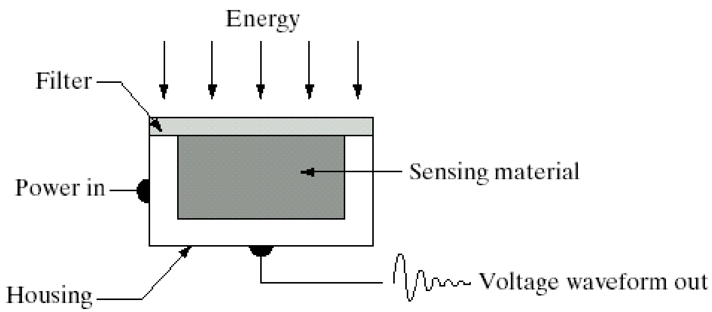

Wavelength, λ (nm)
400
450
500
550
600
650
700
Violet
Blue
Green
Yellow
Orange
Red
Visible spectrum
Wavelength, λ (m)
10^{-12} 10^{-11} 10^{-10} 10^{-9} 10^{-8} 10^{-7} 10^{-6} 10^{-5} 10^{-4} 10^{-3} 10^{-2} 10^{-1} 10^{0} 10^{1}
Gamma
X-ray
Ultraviolet
Infrared
Microwave
Radio
10^{20} 10^{19} 10^{18} 10^{17} 10^{16} 10^{15} 10^{14} 10^{13} 10^{12} 10^{11} 10^{10} 10^{9} 10^{8}
Frequency, ν (Hz)

(a)

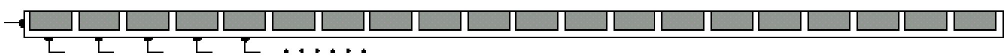


(b)

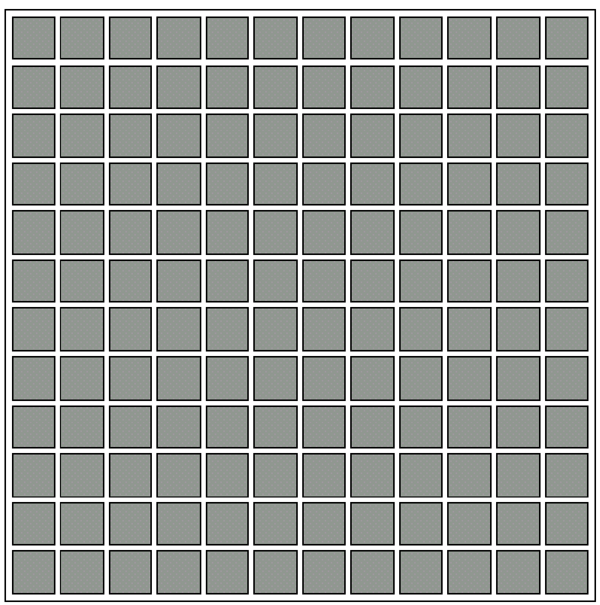
Image Sensors



Single sensor

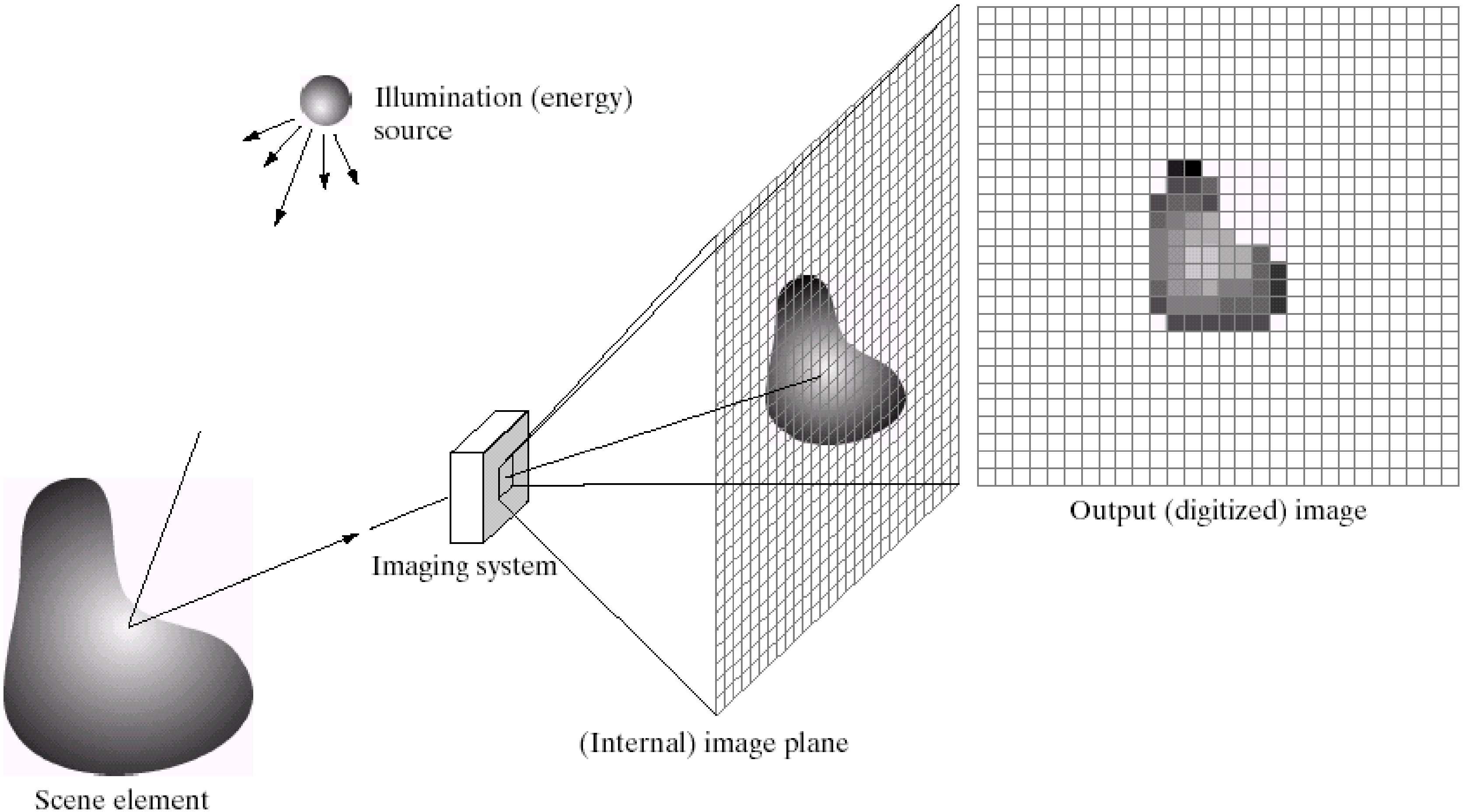


Line sensor

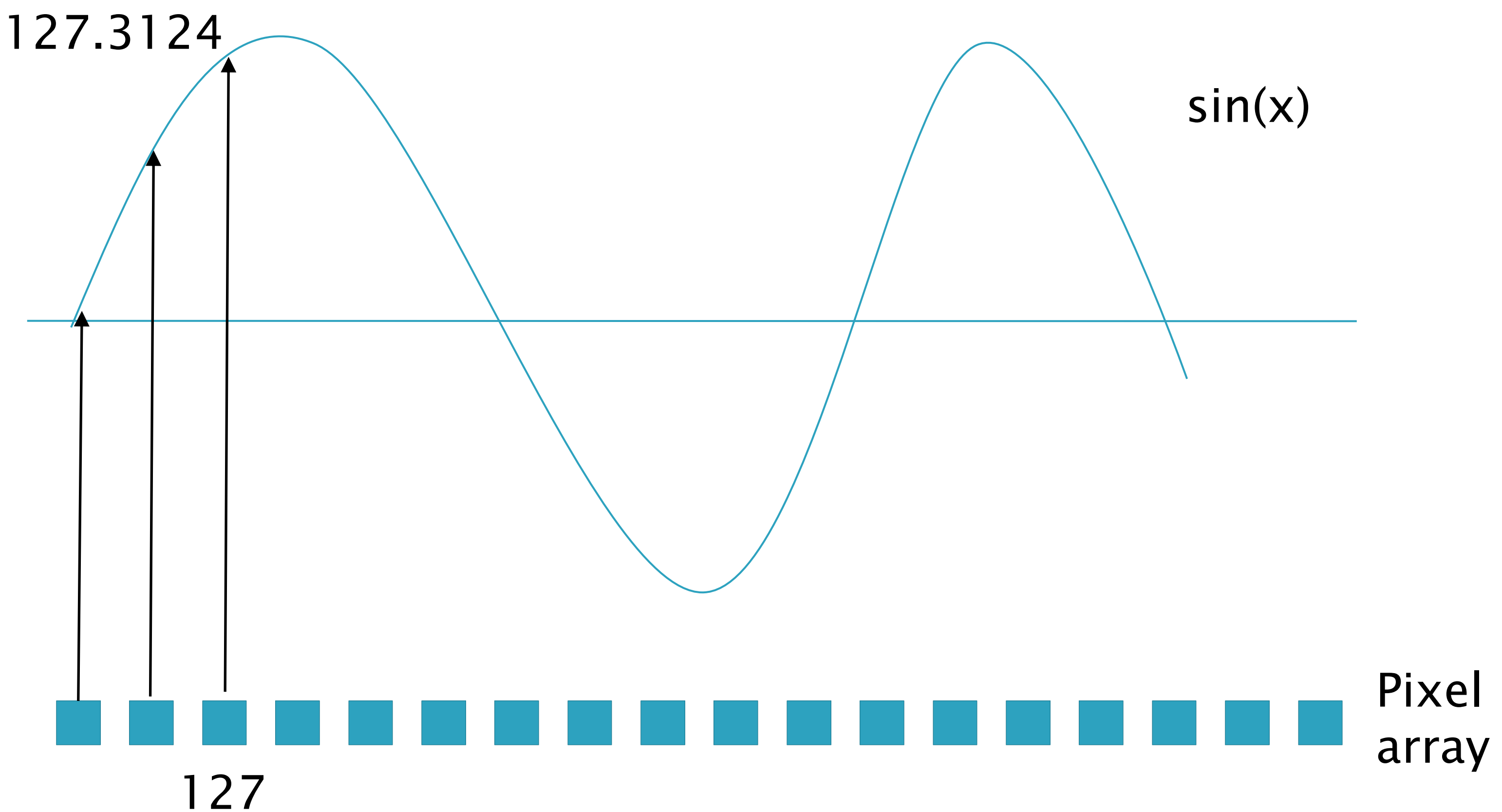


Array sensor

Digital Image Acquisition Process

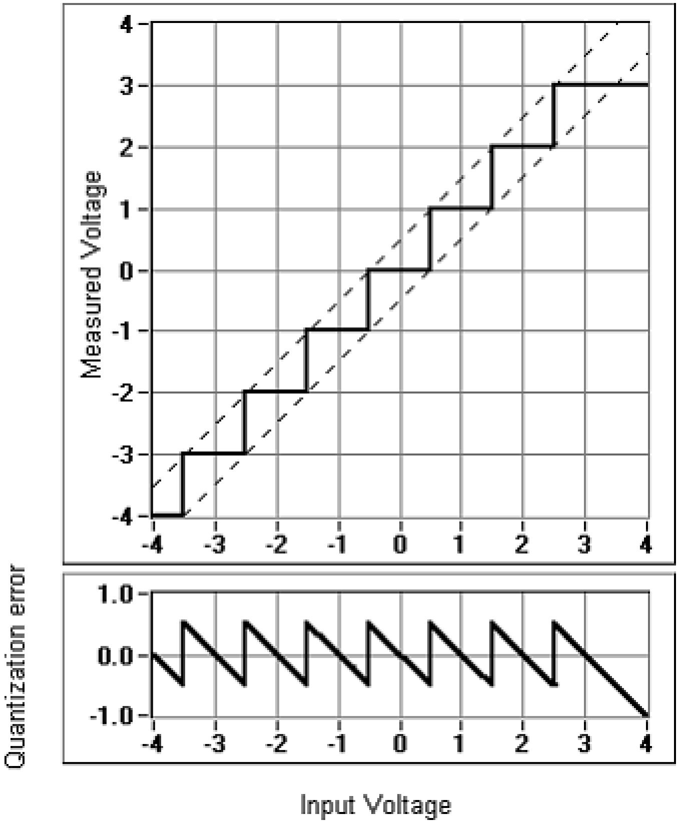


Example of Sampling and Quantization



Quantization

Range	Value
0~0.5	0
0.5~1.5	1
1.5~2.5	2
2.5~3.5	3
3.5~4.5	4
4.5~5.5	5
5.5~6.5	6



Basic Concepts in Sampling and Quantization

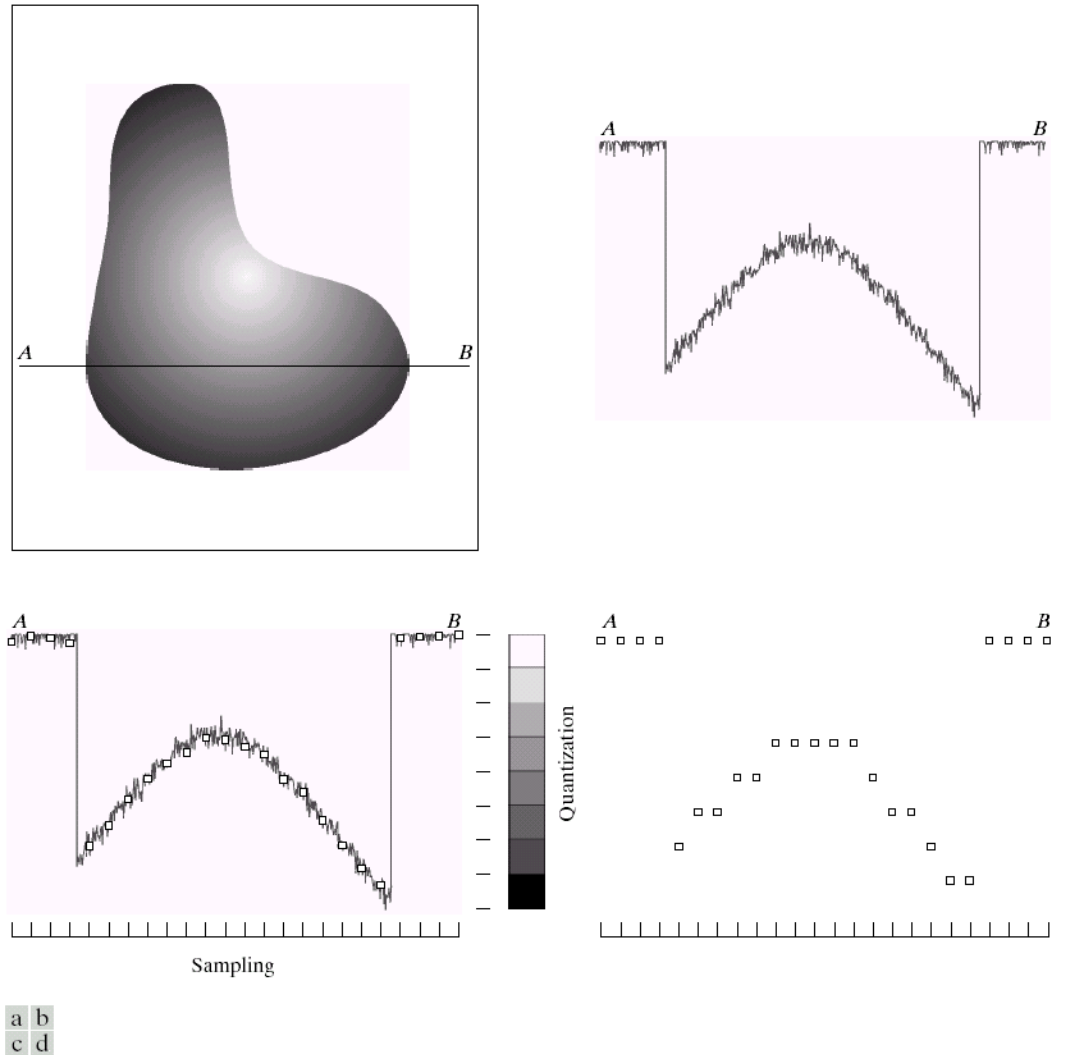
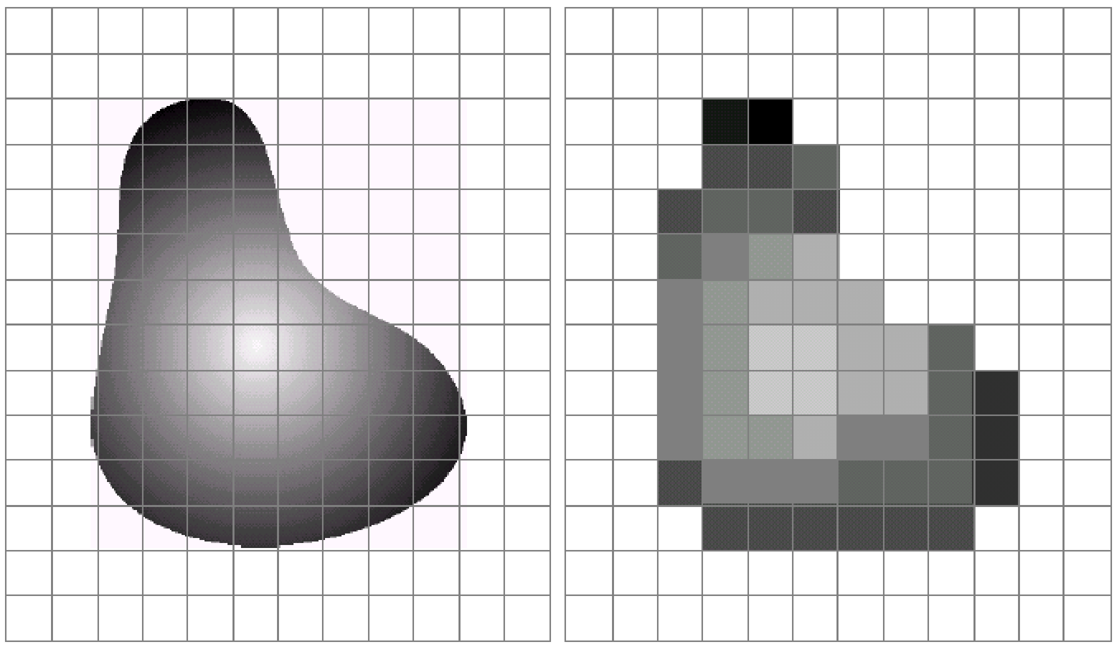


FIGURE 2.16 Generating a digital image. (a) Continuous image. (b) A scan line from A to B in the continuous image, used to illustrate the concepts of sampling and quantization. (c) Sampling and quantization. (d) Digital scan line.

Sampling and Quantization

- Sensor can measure a limited number of **samp les** at a **discrete** set of energy levels.

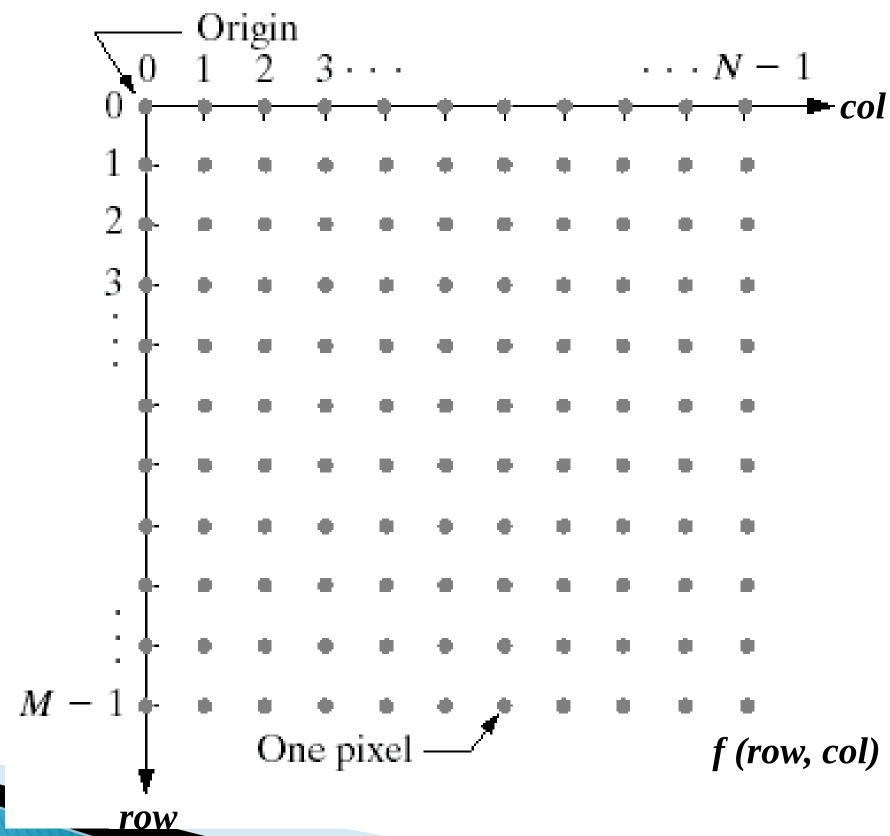


a b

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

Coordinate for Digital Images

- A digital image is composed of M rows and N columns of pixels

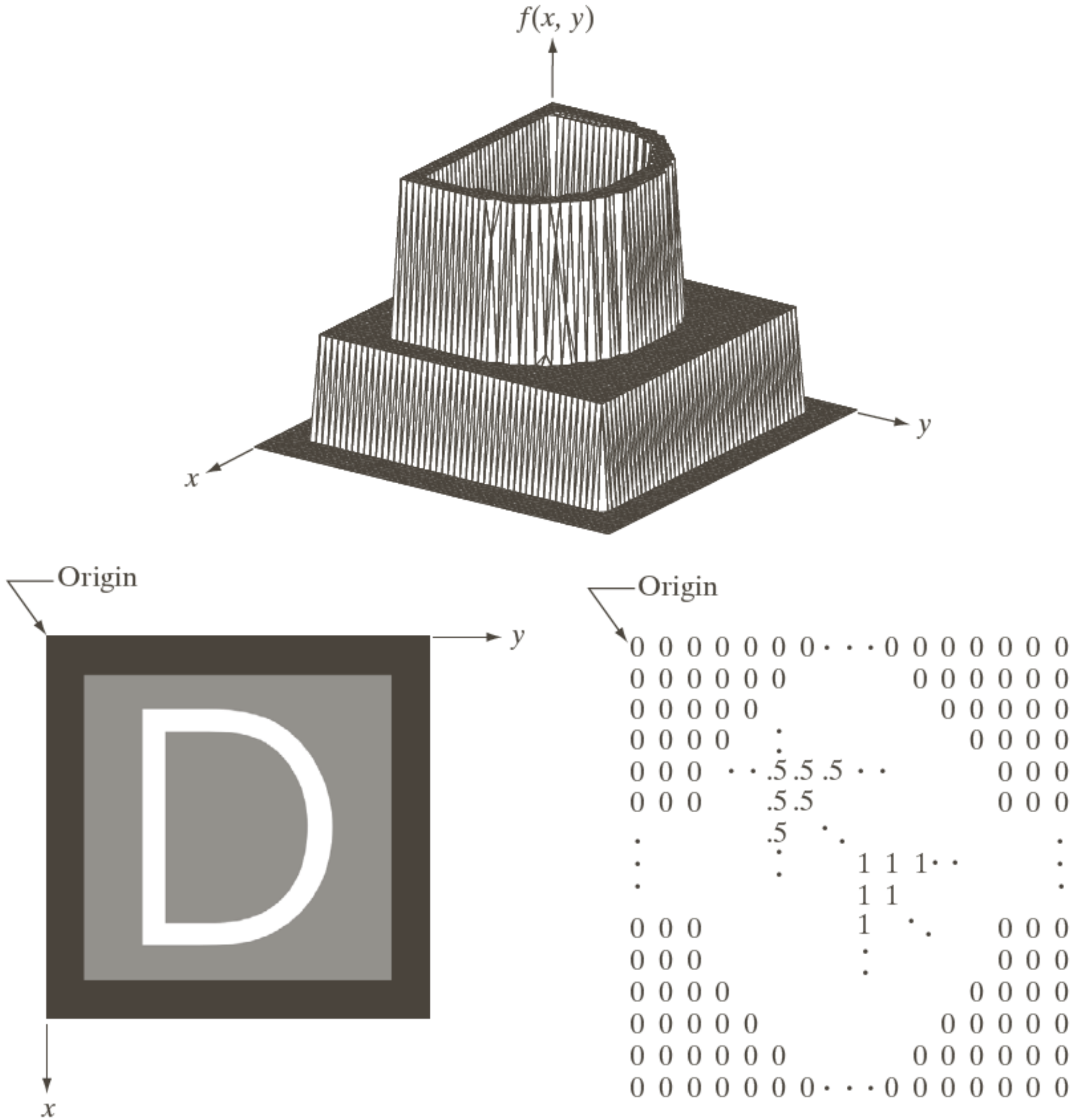


Representing Digital Images

- The representation of an M×N numerical array as

$$f(x,y)=\begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0,N-1) \\ f(1,0) & f(1,1) & \dots & f(1,N-1) \\ \dots & \dots & \dots & \dots \\ f(M-1,0) & f(M-1,1) & \dots & f(M-1,N-1) \end{bmatrix}$$

Image Representation



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).

Number of Storage Bits

- Assumption
  - Image size (resolution): M x N
  - Intensity level: [0, 255] / 8 bits

- Gray Image
  - 1 pixel: 1 byte
  - # of pixels: M x N
  - M x N pixels → M x N x 1 bytes
- Color Image (R, G, B)
  - 1 pixel: 3 bytes
  - # of pixels: M x N
  - M x N pixels → M x N x 3 bytes

Saturation and Noise

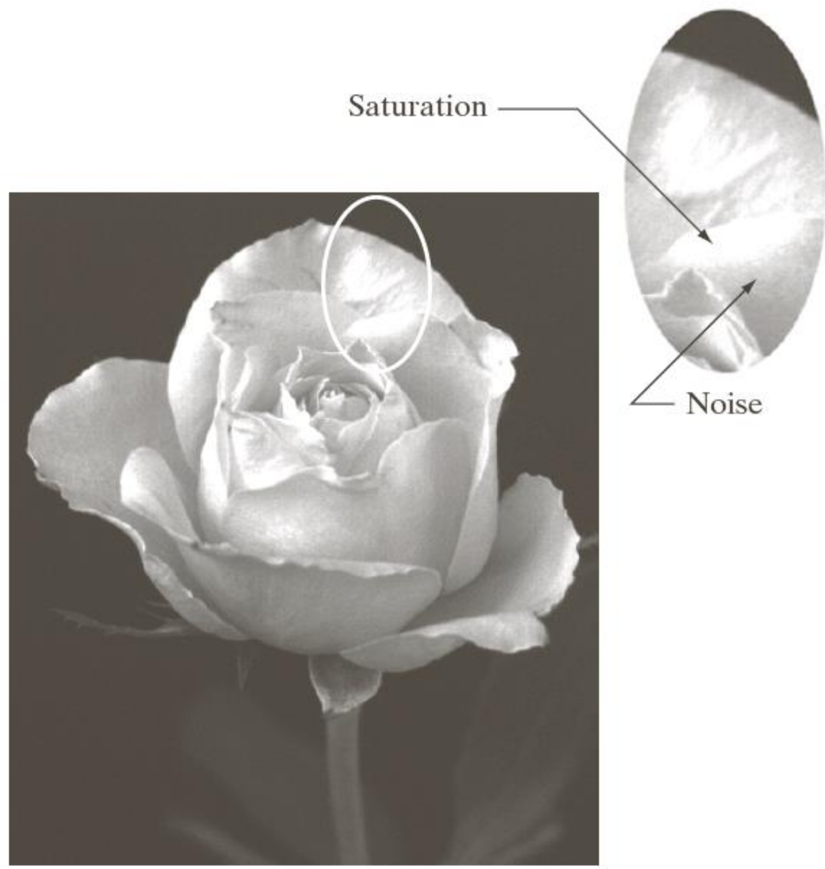


FIGURE 2.19 An image exhibiting saturation and noise. Saturation is the highest value beyond which all intensity levels are clipped (note how the entire saturated area has a high, *constant* intensity level). Noise in this case appears as a grainy texture pattern. Noise, especially in the darker regions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

Image Resolution

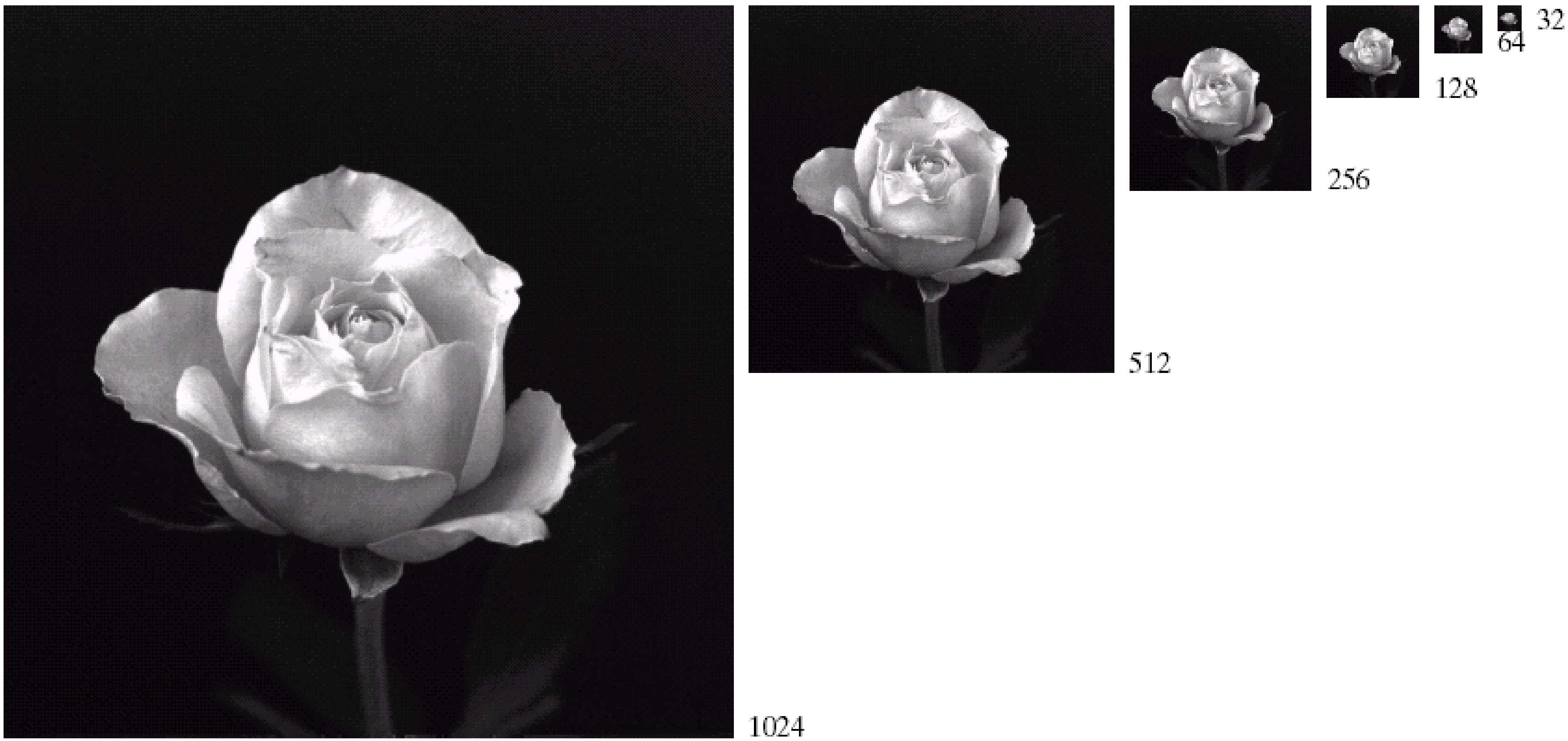


FIGURE 2.19 A 1024 × 1024, 8-bit image subsampled down to size 32 × 32 pixels. The number of allowable gray levels was kept at 256.

Image Resolution

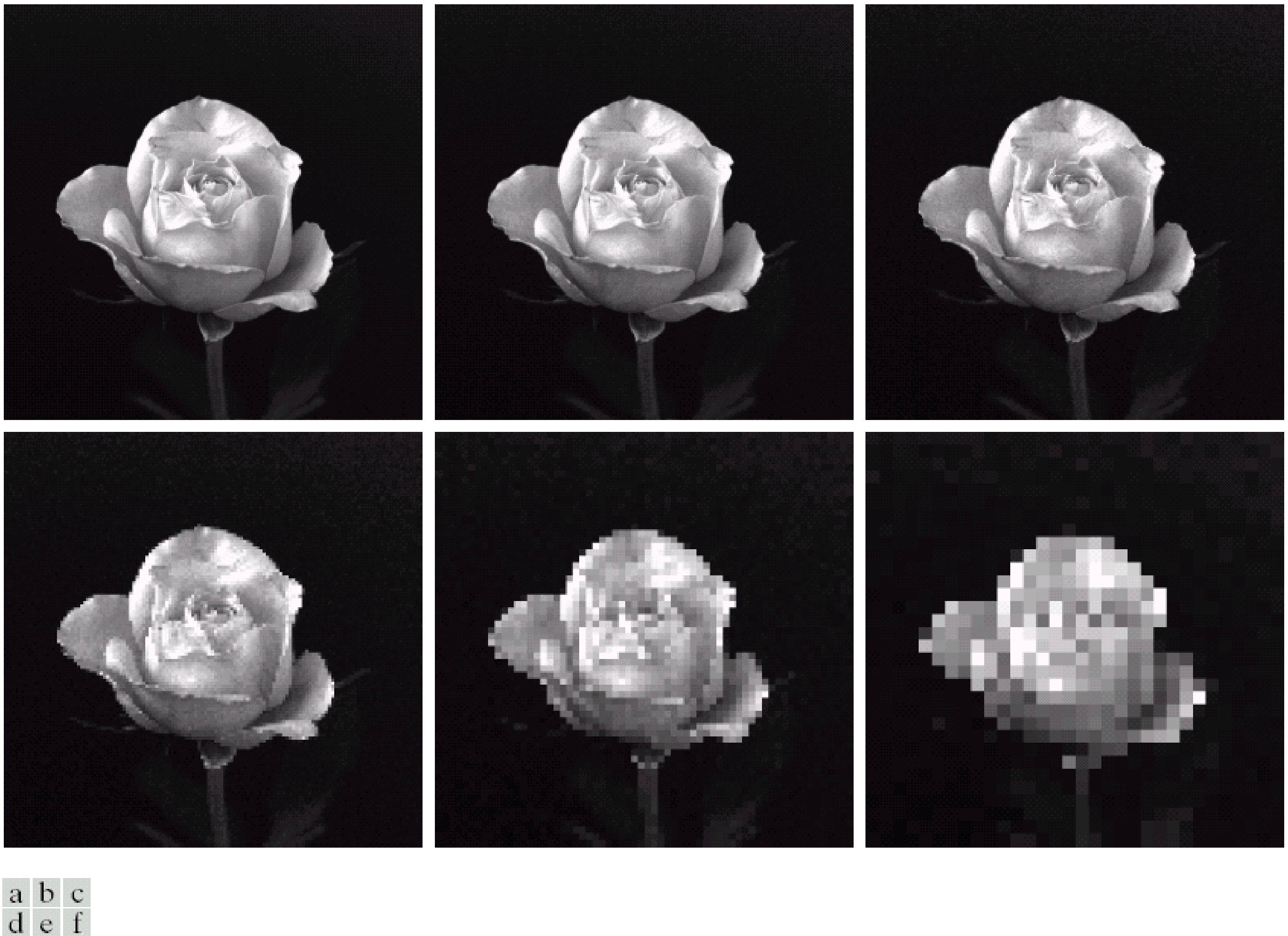


FIGURE 2.20 (a) 1024 × 1024, 8-bit image. (b) 512 × 512 image resampled into 1024 × 1024 pixels by row and column duplication. (c) through (f) 256 × 256, 128 × 128, 64 × 64, and 32 × 32 images resampled into 1024 × 1024 pixels.

Gray Levels

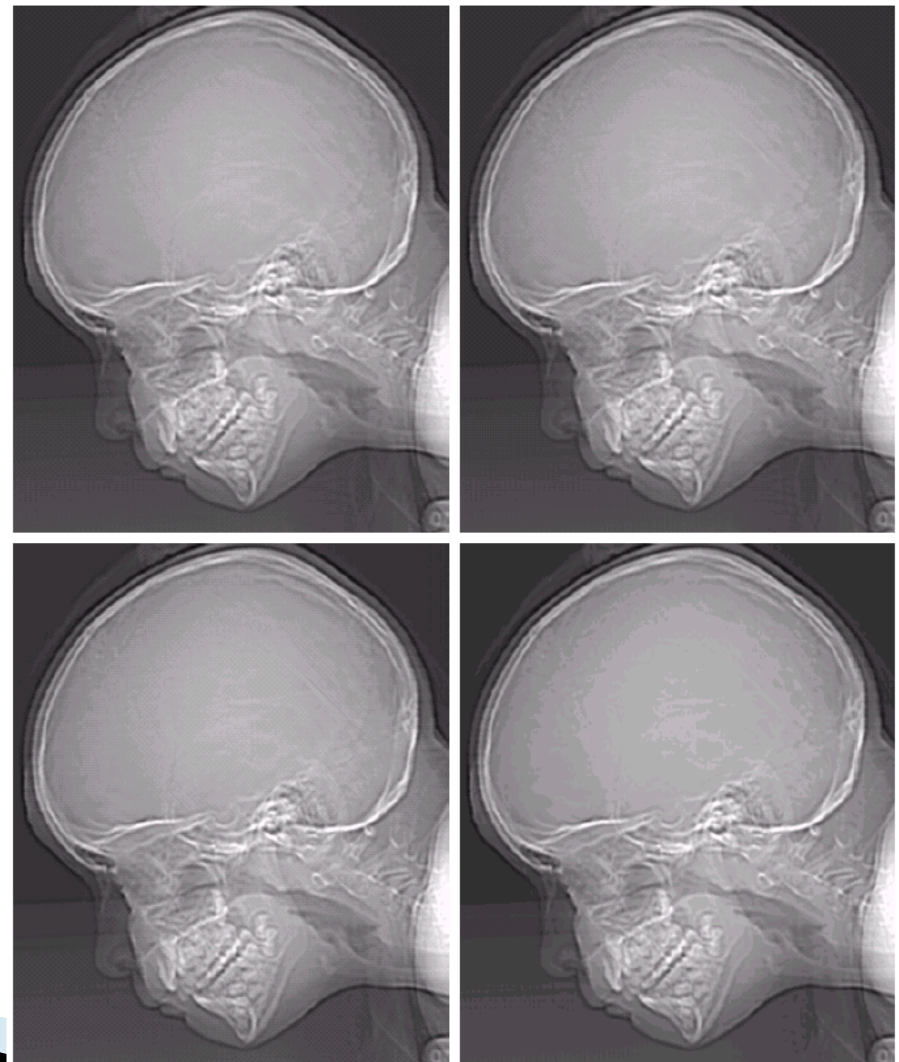


FIGURE 2.21 (a) 452×374, 256-level image. (b)–(d) Image displayed in 128, 64, and 32 gray levels, while keeping the spatial resolution constant.

Gray Levels

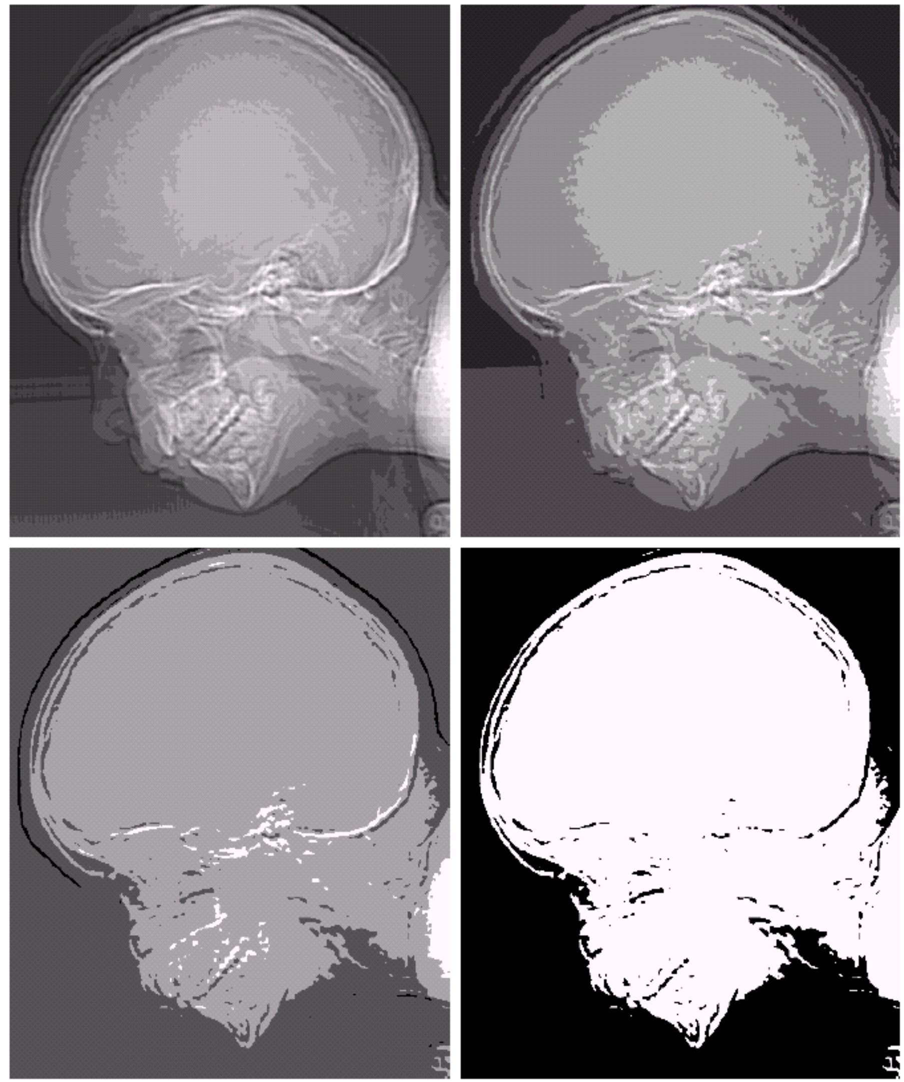
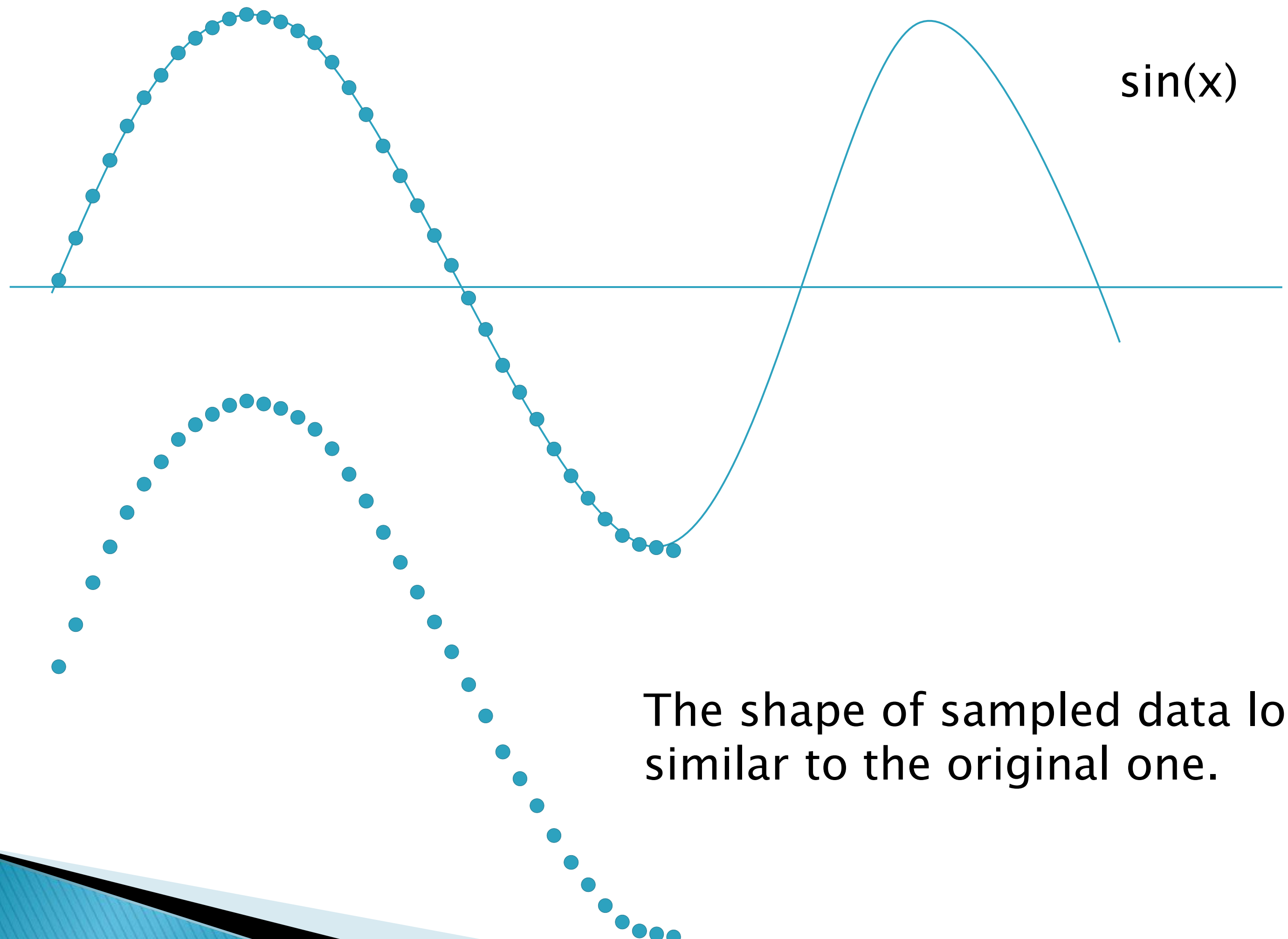


FIGURE 2.21 *(Continued)* (e)–(h) Image displayed in 16, 8, 4, and 2 gray levels. (Original courtesy of Dr. David R. Pickens, Department of Radiology & Radiological Sciences, Vanderbilt University Medical Center.)

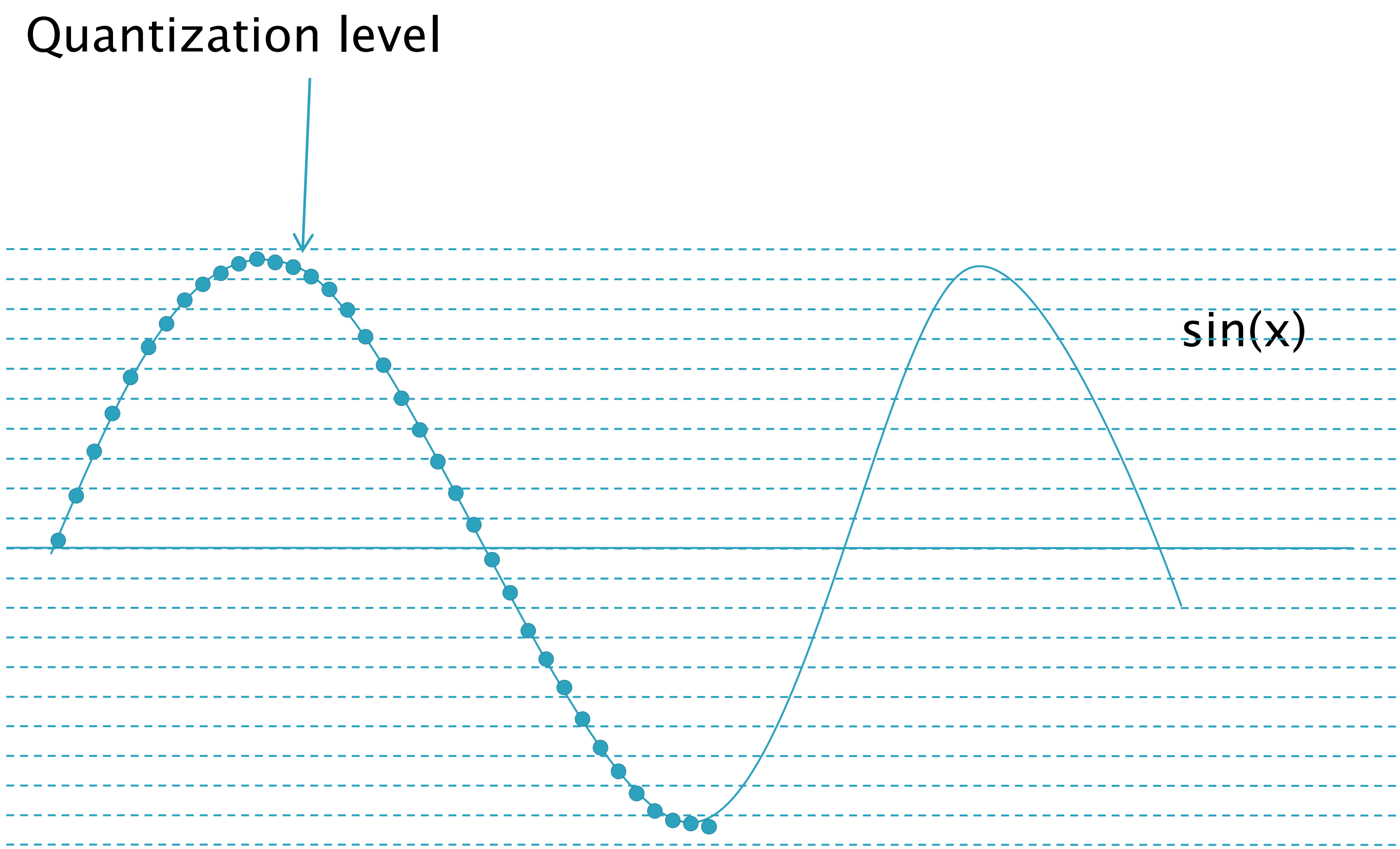
Quantization Level (1/4)

Sampling Process

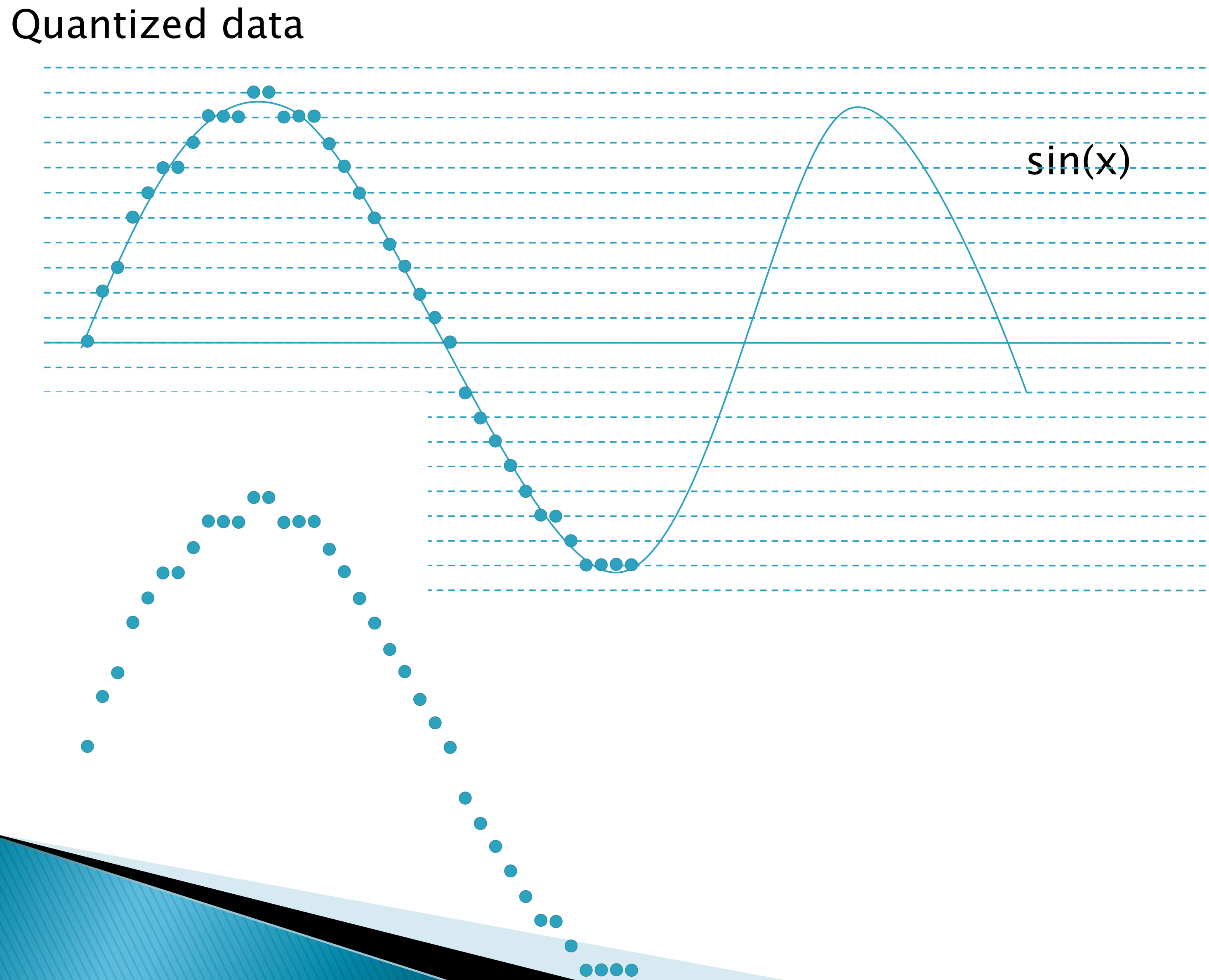


The shape of sampled data looks similar to the original one.

Quantization Level (2/4)

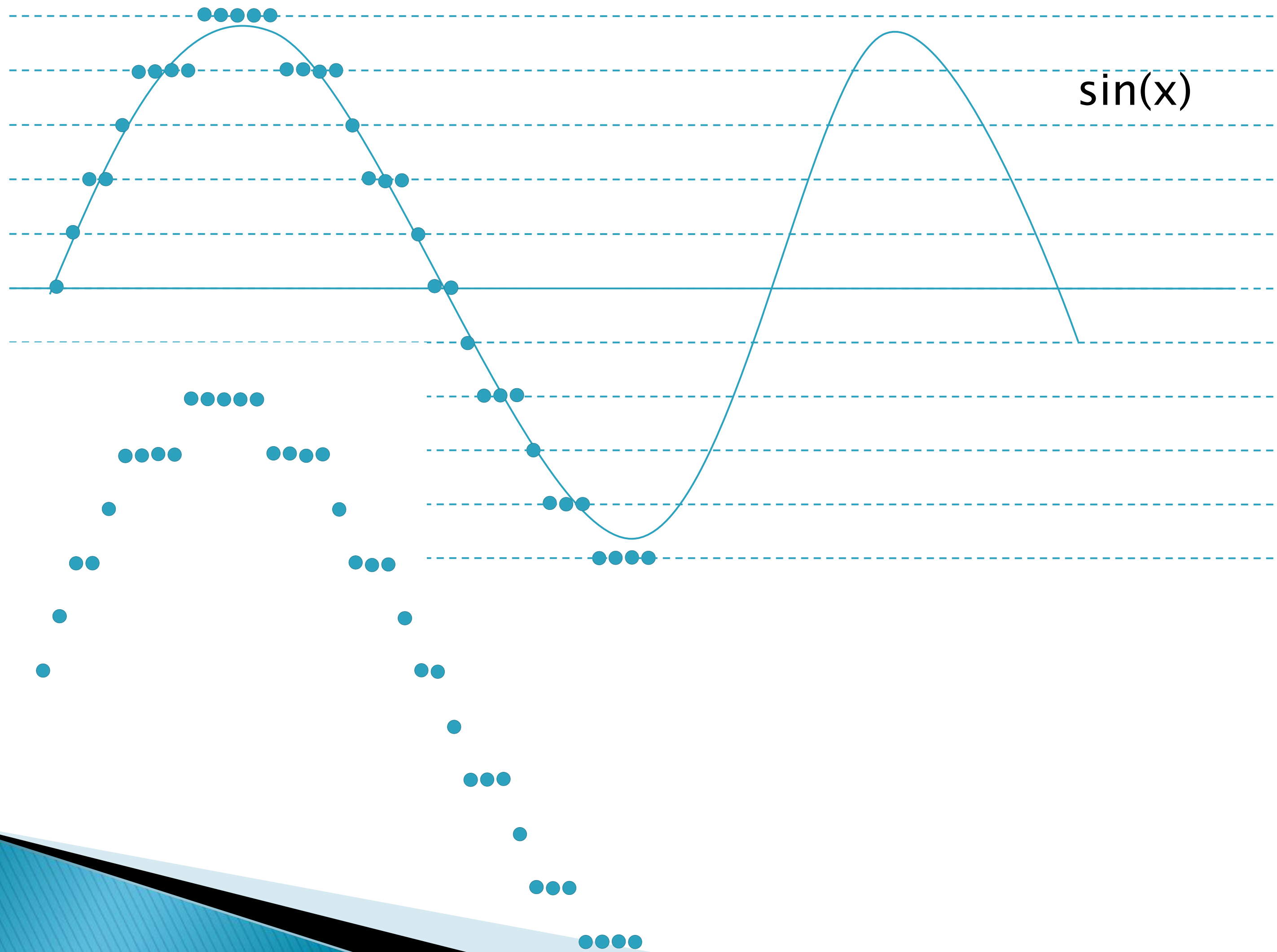


Quantization Level (3/4)

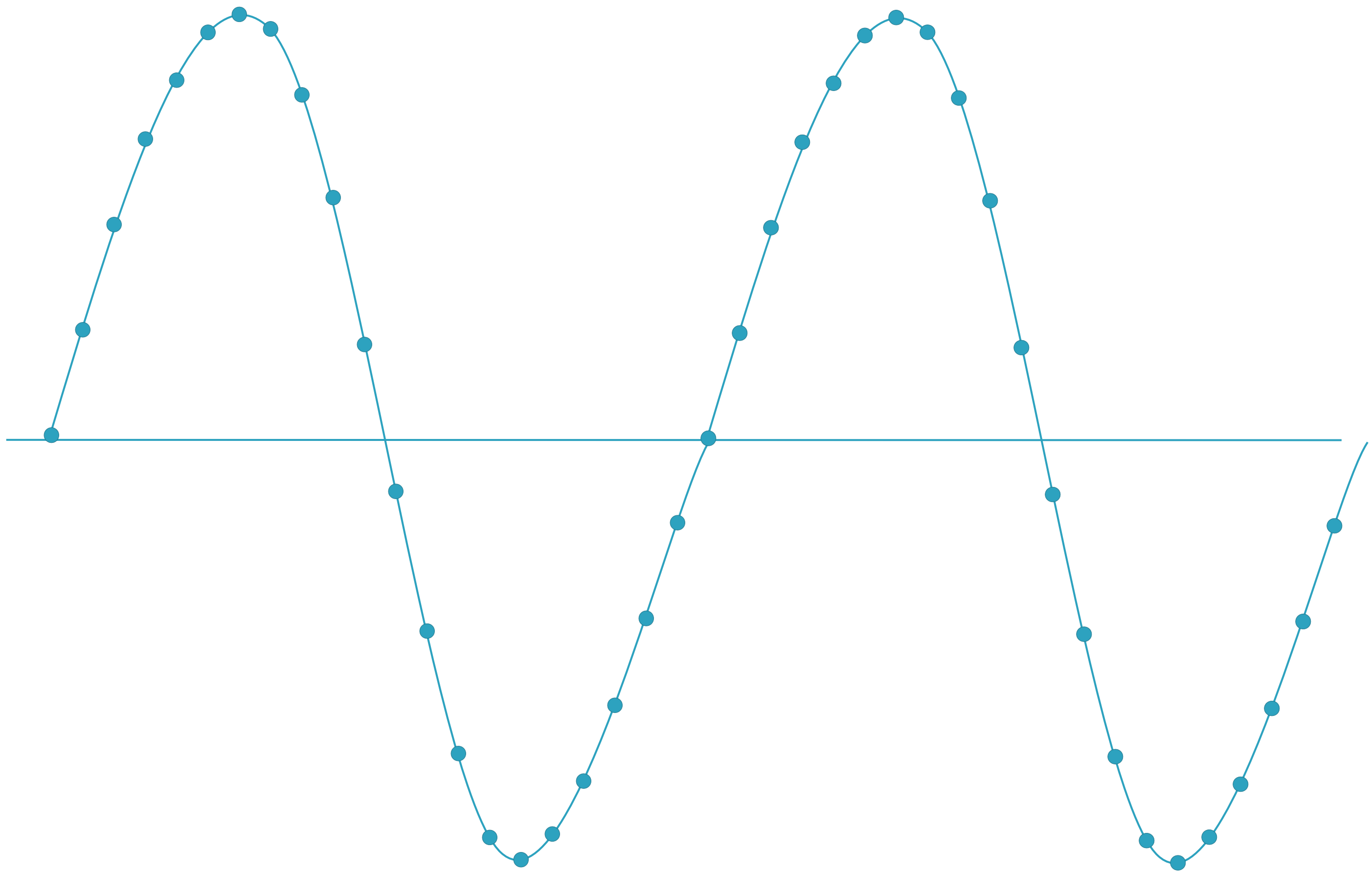


Quantization Level (4/4)

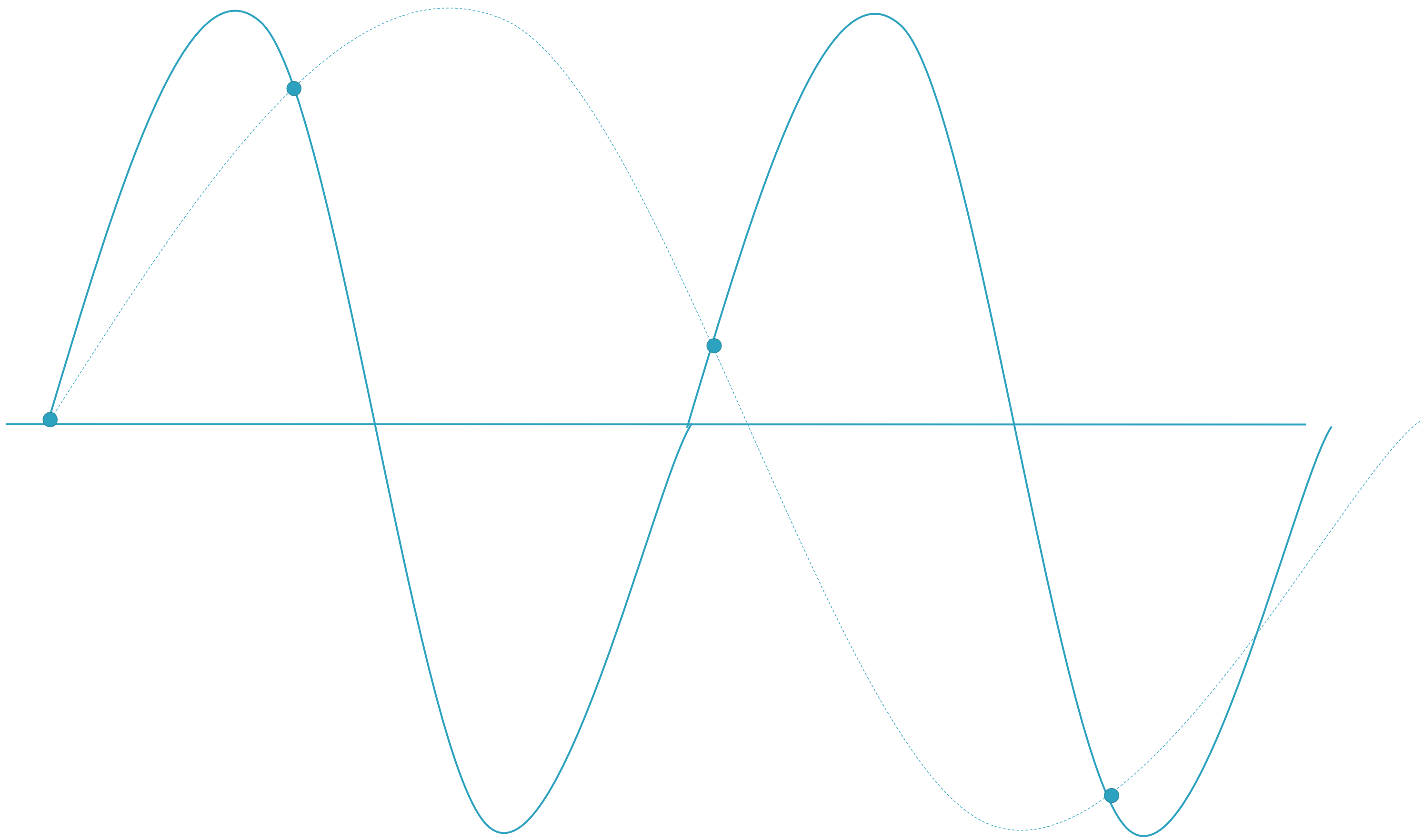
Quantized data



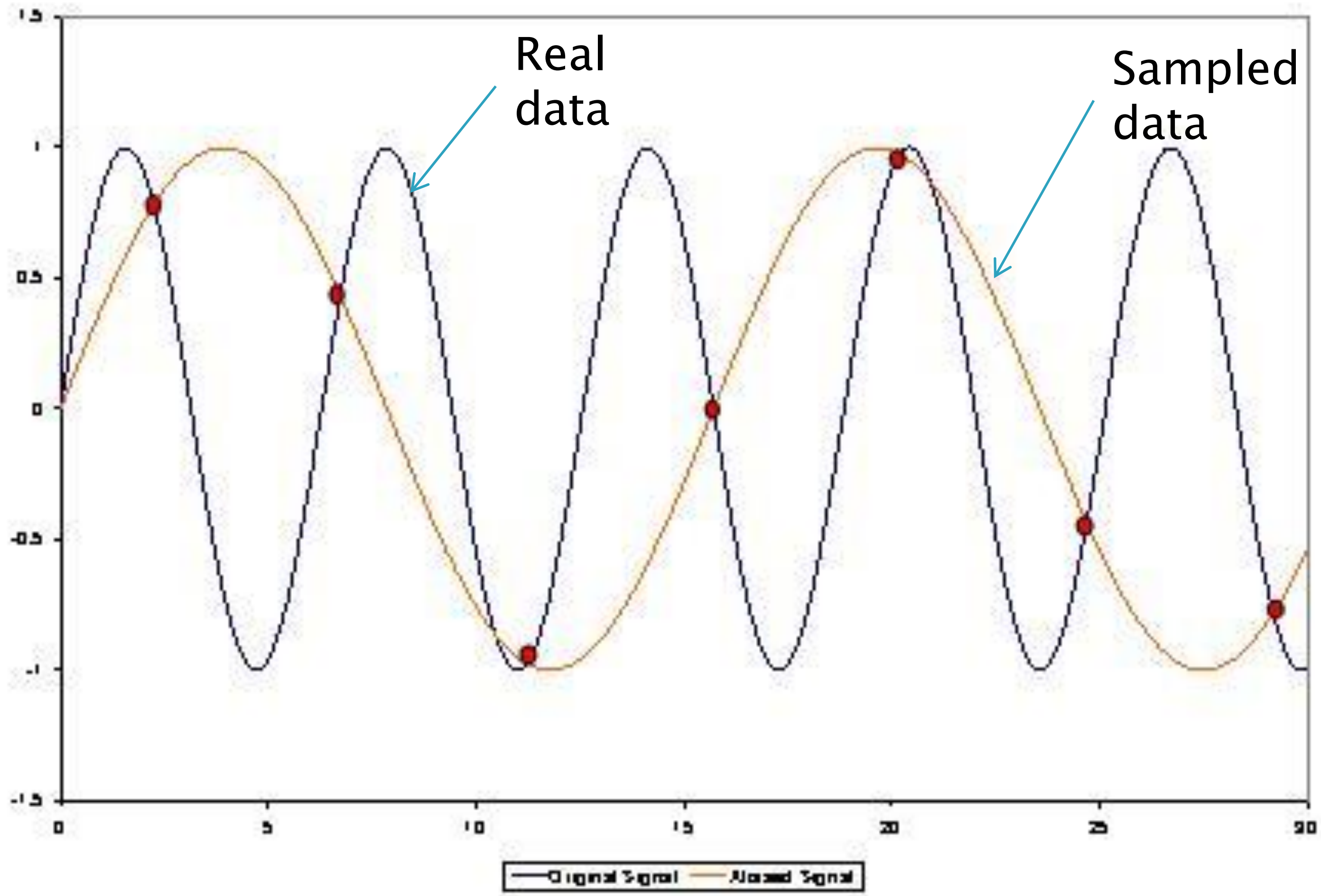
Sampling and Aliasing (1/3)



Sampling and Aliasing (2/3)



Sampling and Aliasing (3/3)



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).

Image Zoom

a b c
d e f

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.

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.)

Pixel ADD/SUB (1/2)

- Suppose a pixel with n-bit accuracy.
  - Range: $0 \sim 2^n-1$
  - Pixel value cannot be smaller than 0 or more than 2^n-1.

  - Example ($n = 8$)
    - Range: $0 \sim 255$

    - Pixel value add/sub
      - $128 + 255 \rightarrow 255$
      - $128 - 255 \rightarrow 0$

Pixel ADD/SUB (2/2)

- Program Example
  - unsigned char pix1, pix2;

  - pix1 + pix2 = ?
    - pix1: 255, pix2 = 1

  - pix1 – pix2 = ?
    - pix1= 0, pix2= 1

Maxtrix

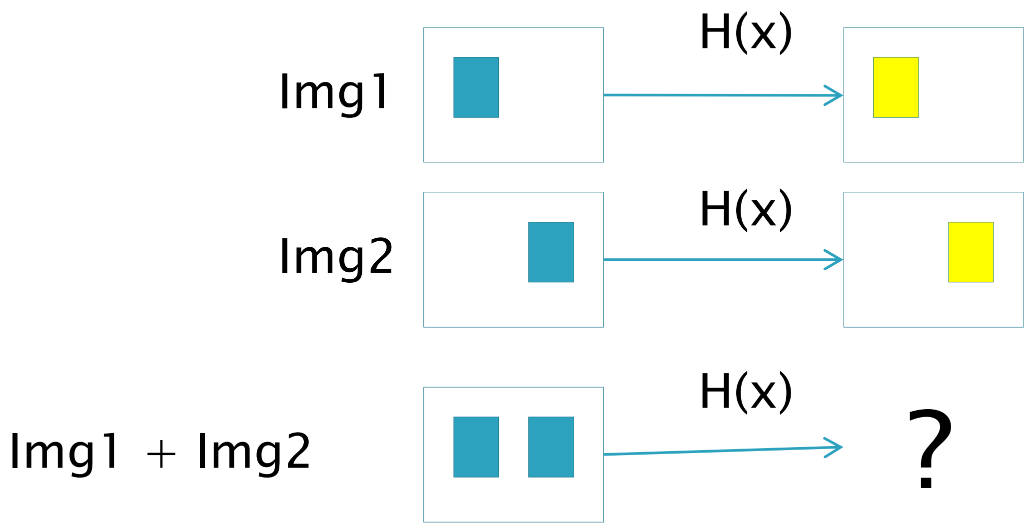
- Array Product

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

- Matrix Product

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

Linera vs Non-Linear (1/4)



- $H(Img1+Img2)=H(Img1)+H(Img2)$
- $H(Img1+Img2)\neq H(Img1)+H(Img2)$

Linera vs Non-Linear (2/4)

- Linear vs non-linear

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

  - Linear

$$H(X) = \begin{bmatrix} 3 & 1 \\ -2 & 2 \end{bmatrix} X$$

$$\begin{bmatrix} 3 & 1 \\ -2 & 2 \end{bmatrix}(3f_1 - f_2) = 3\begin{bmatrix} 3 & 1 \\ -2 & 2 \end{bmatrix}\begin{bmatrix} 0 & 2 \\ 2 & 3 \end{bmatrix} - \begin{bmatrix} 3 & 1 \\ -2 & 2 \end{bmatrix}\begin{bmatrix} 6 & 5 \\ 4 & 7 \end{bmatrix}$$

$$= 3\begin{bmatrix} 2 & 9 \\ 4 & 2 \end{bmatrix} - \begin{bmatrix} 22 & 22 \\ -4 & 4 \end{bmatrix} = \begin{bmatrix} -16 & 5 \\ 16 & 2 \end{bmatrix}$$

$$\begin{bmatrix} 3 & 1 \\ -2 & 2 \end{bmatrix}(3f_1 - f_2) = \begin{bmatrix} 3 & 1 \\ -2 & 2 \end{bmatrix}\begin{bmatrix} -6 & 1 \\ 2 & 2 \end{bmatrix}$$

$$= \begin{bmatrix} -16 & 5 \\ 16 & 2 \end{bmatrix}$$

Linera vs Non-Linear (3/4)

- Non-linear

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

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

Working next with the right side, we obtain

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

Linera vs Non-Linear (4/4)

- Linear

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

H is said to be a *linear operator* if

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

- Example

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

Image Rotation (1/5)

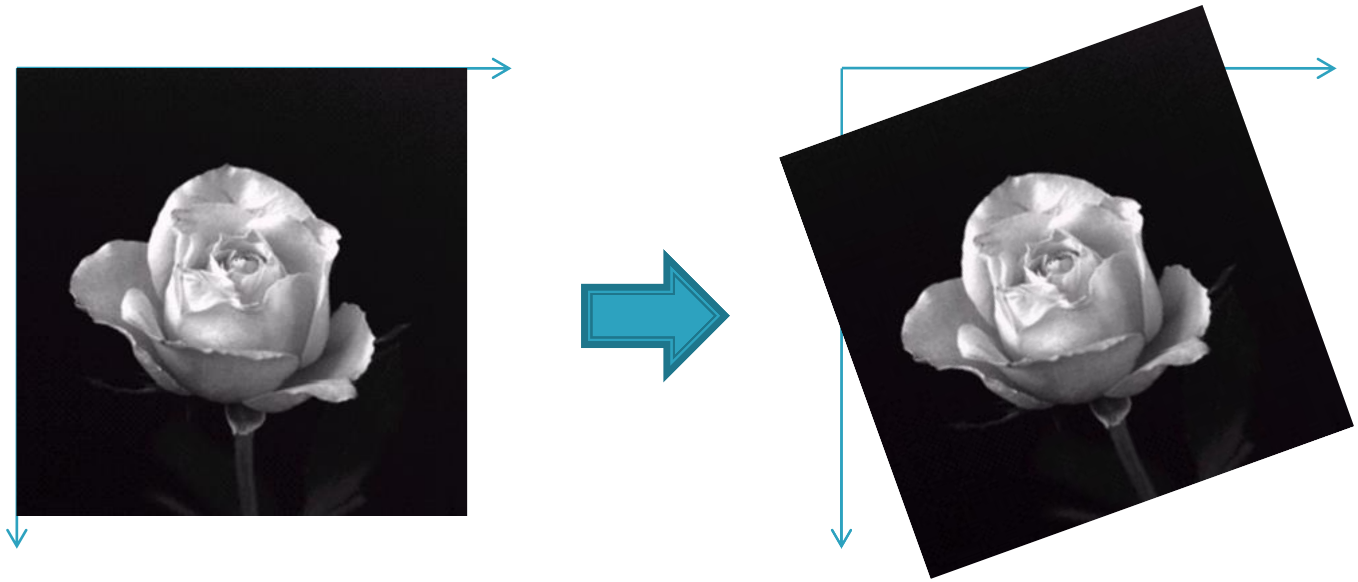
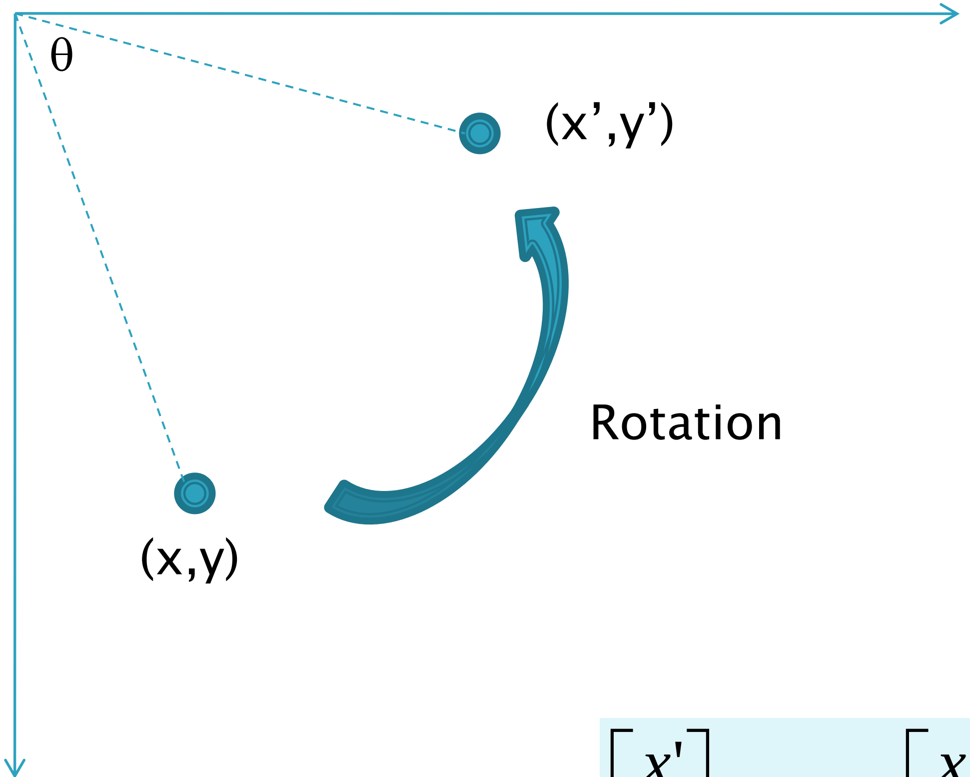


Image Rotation (2/5)



$$\begin{bmatrix} x' \\ y' \end{bmatrix} = R(\theta)\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix}$$

Image Rotation (3/5)

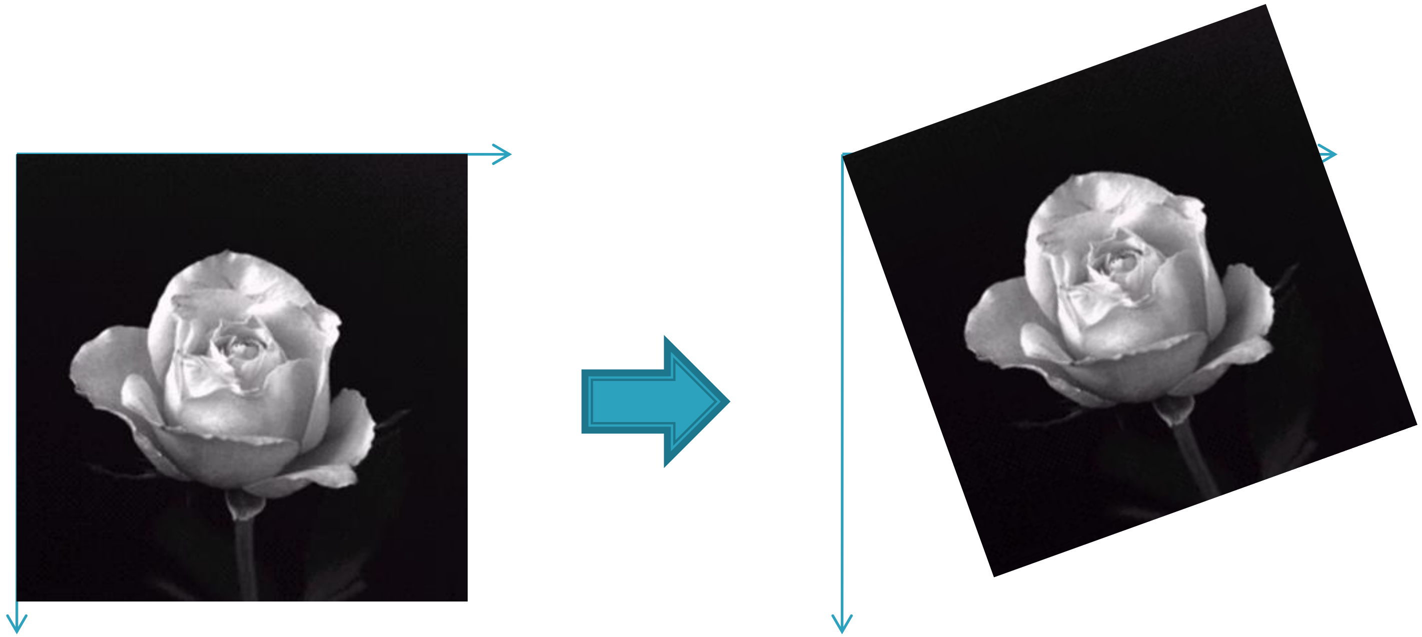


Image Rotation (4/5)

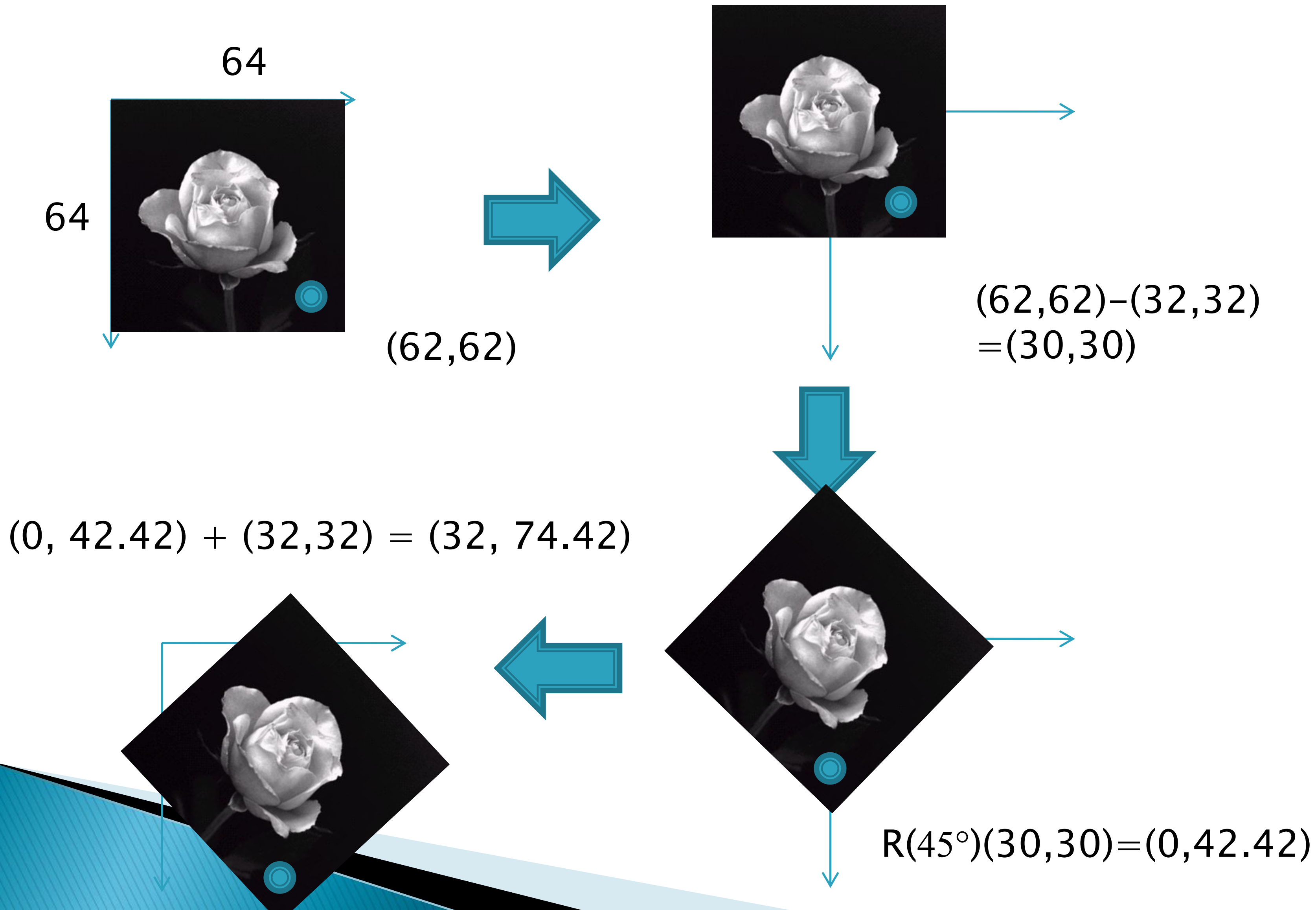


Image Rotation (5/5)

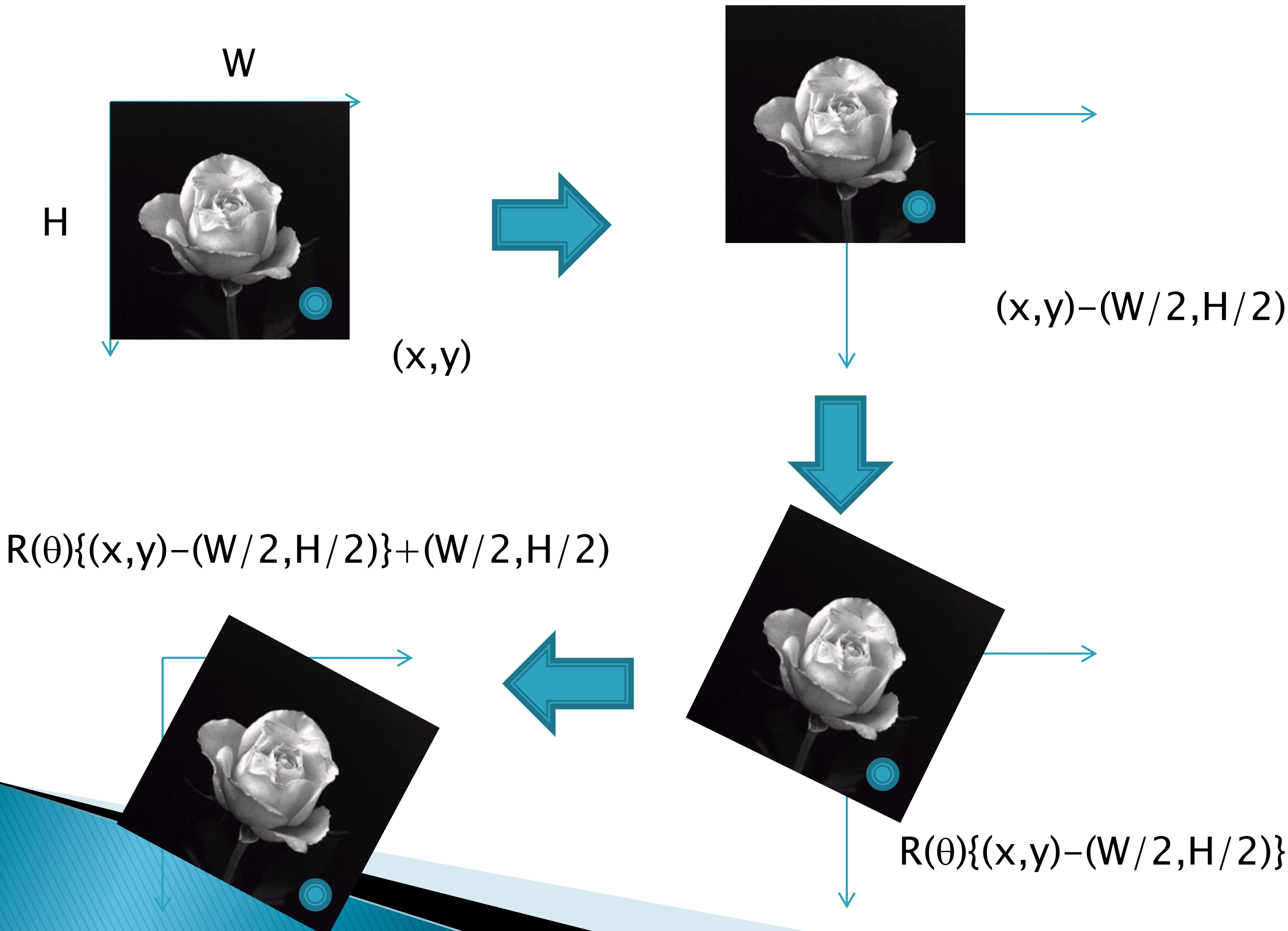
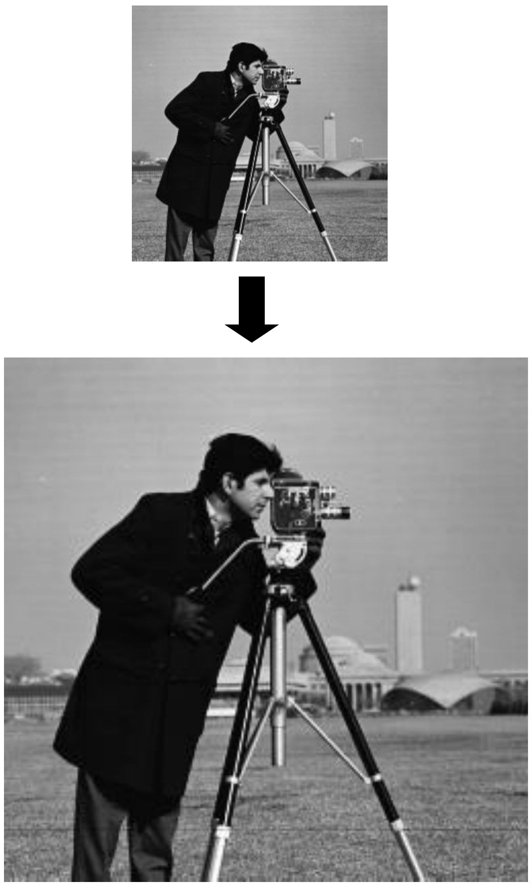
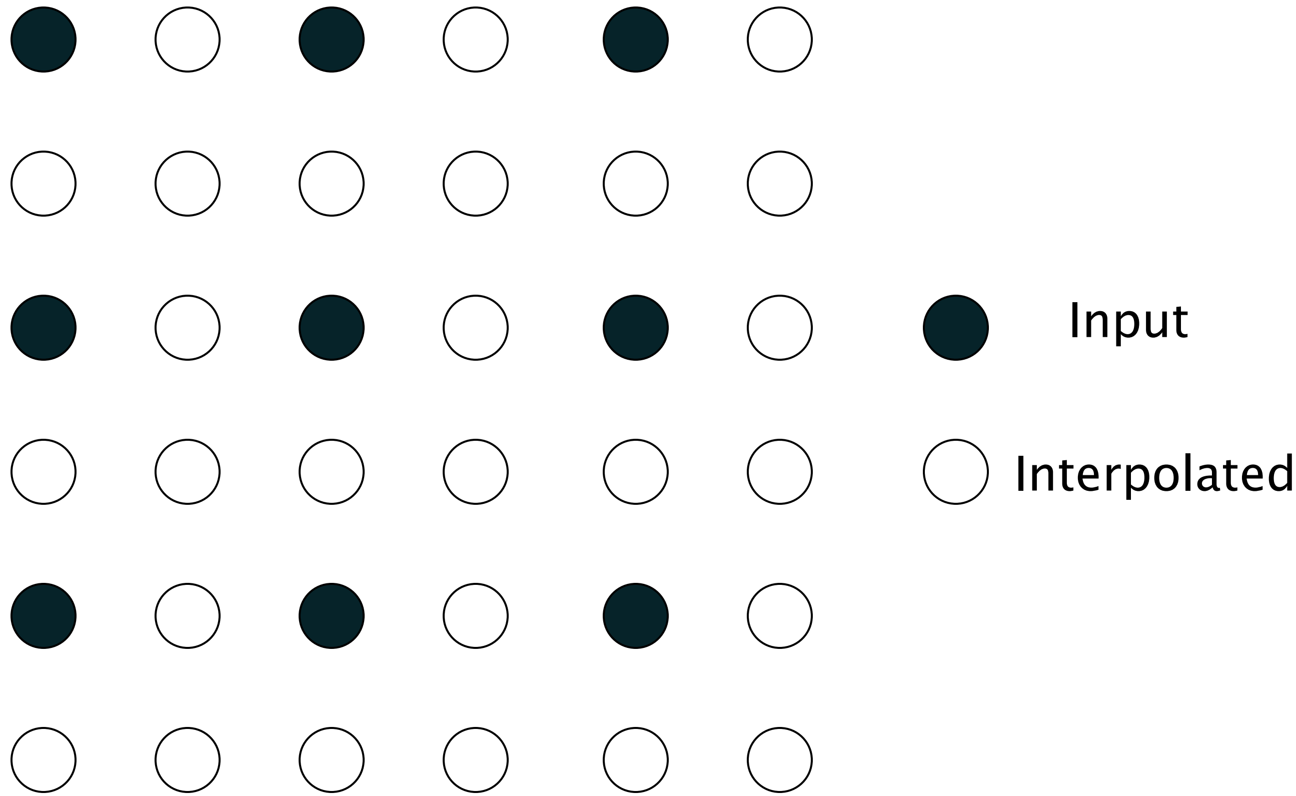
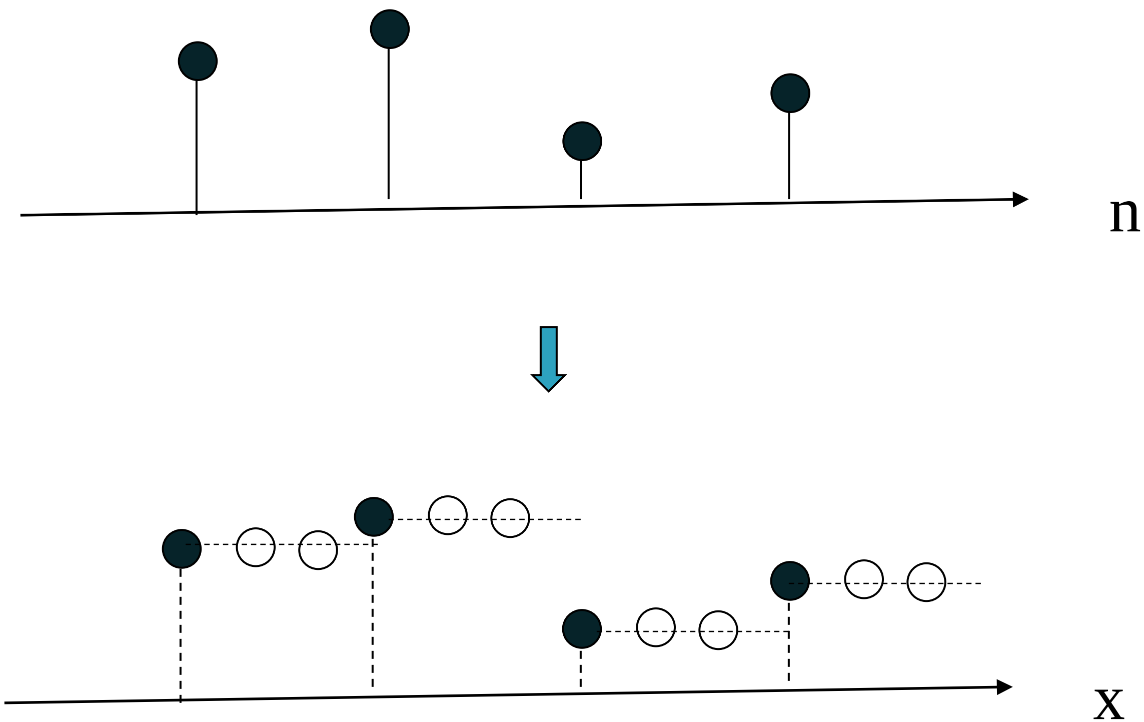


Image Interpolation

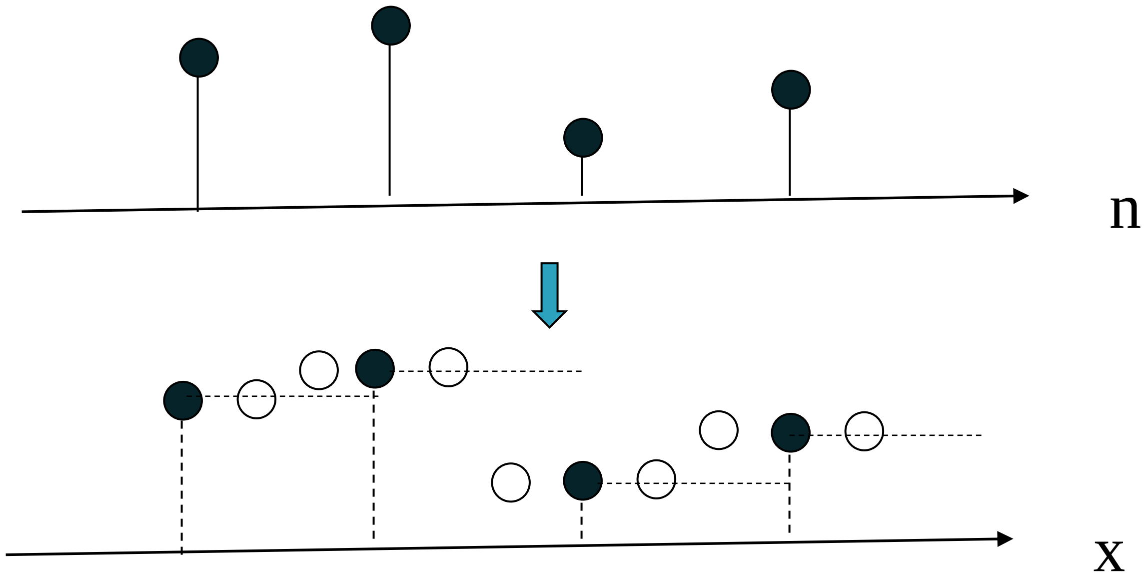




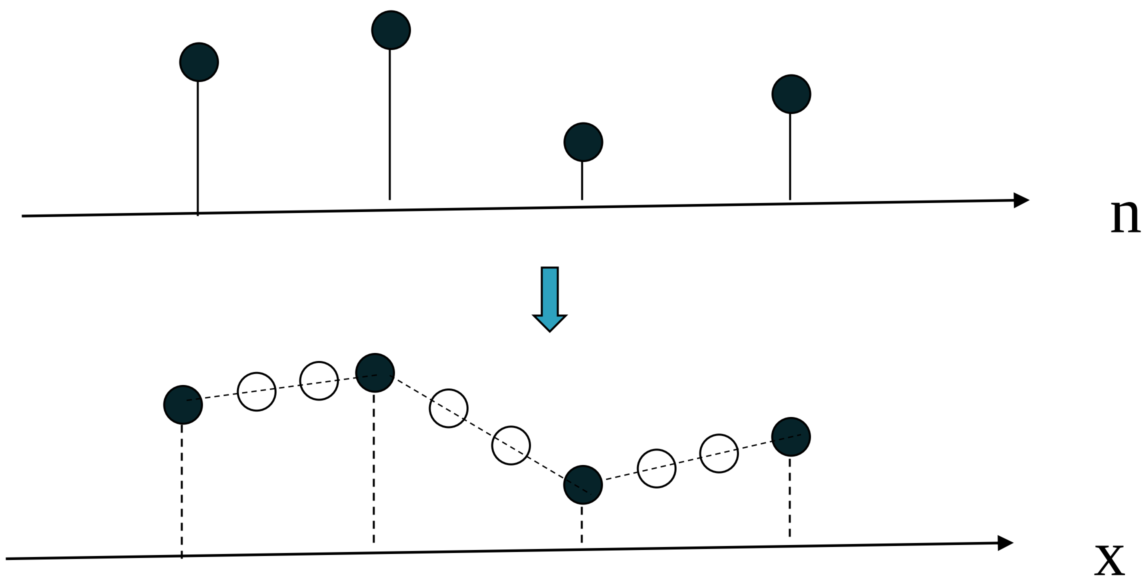
Replication (N times)



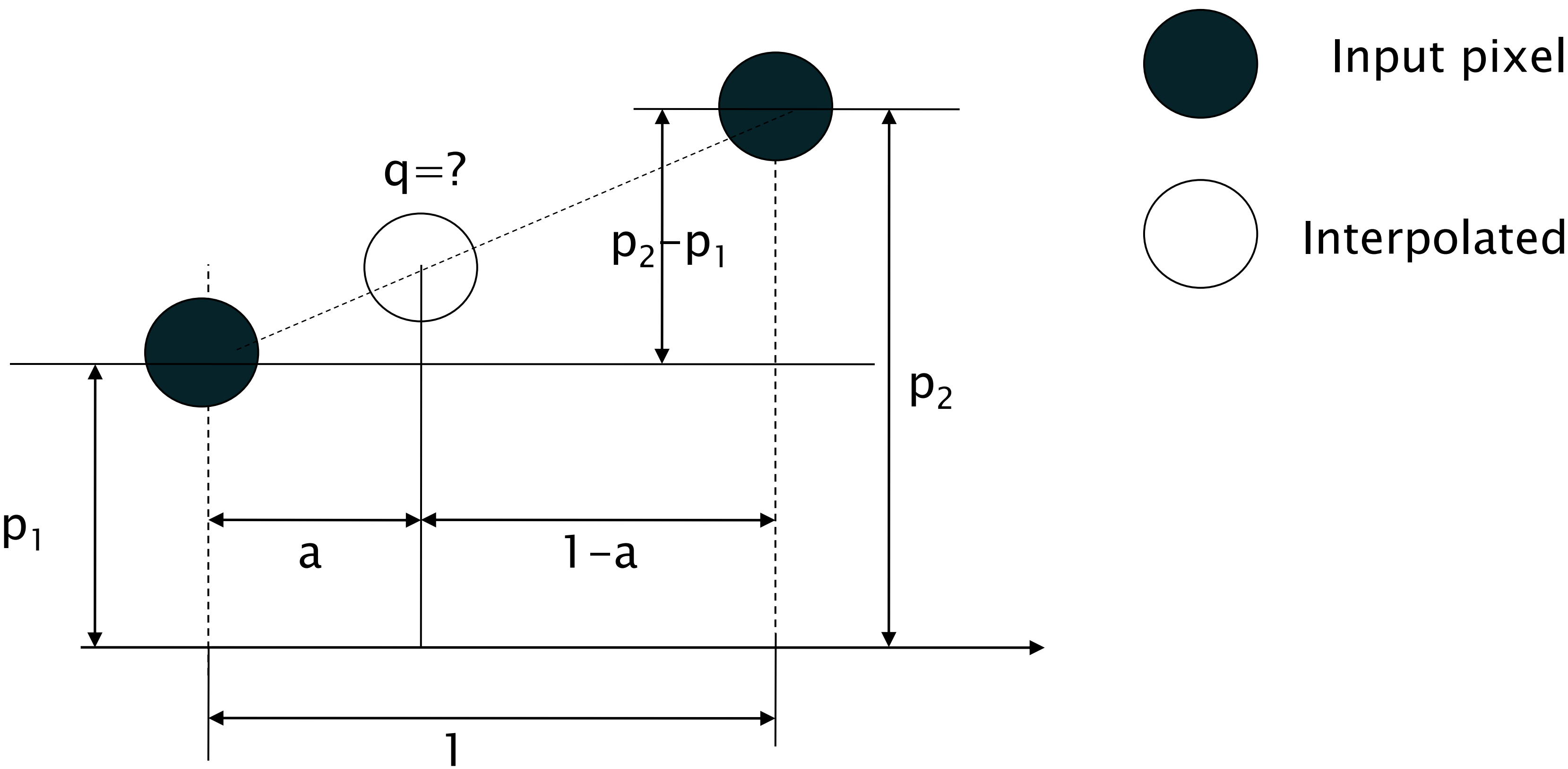
Nearest Neighbor



Linear Interpolation (1/3)

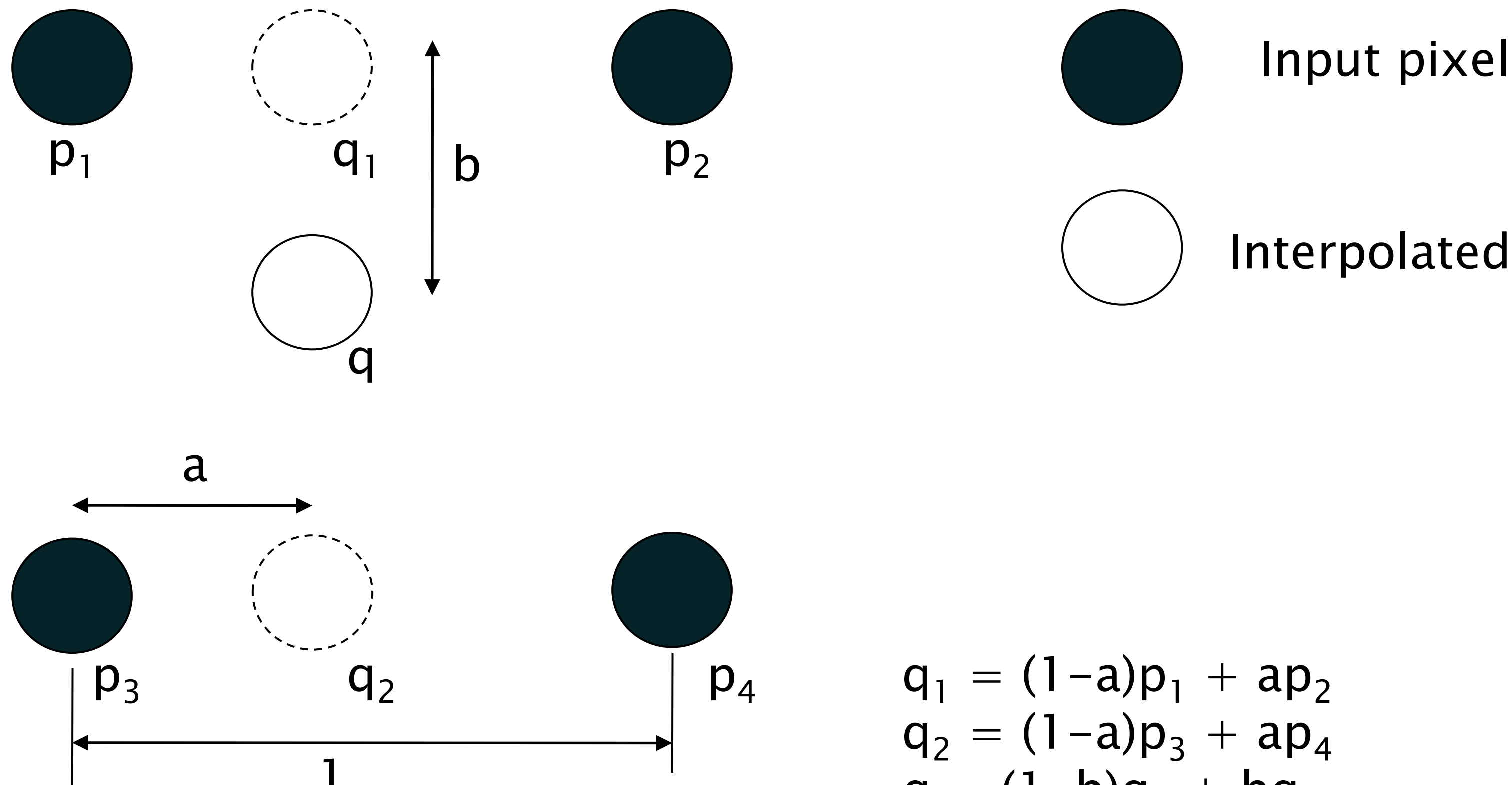


Linear Interpolation (2/3)



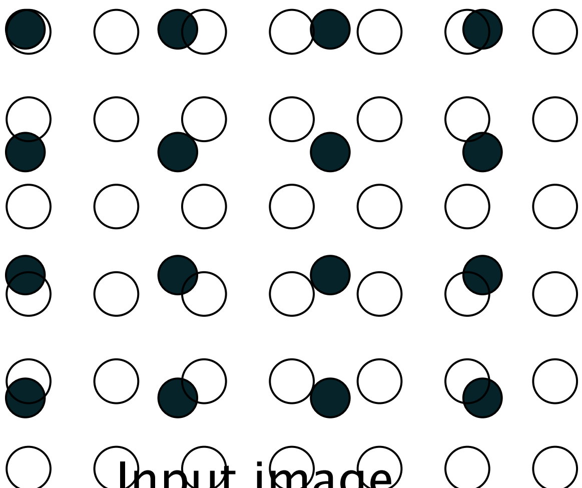
$$q = p_1 + (p_2-p_1)a$$
$$= (1-a)p_1 + ap_2$$

Linear Interpolation (3/3)

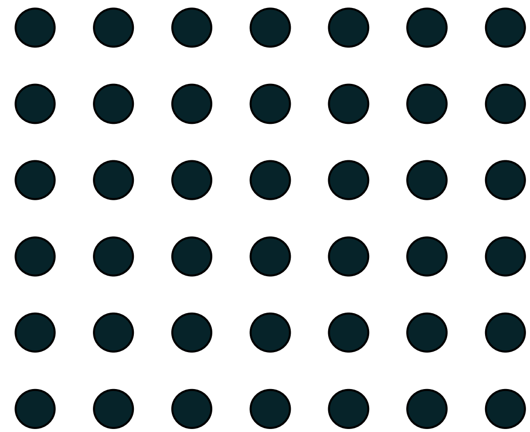


$$q_1 = (1-a)p_1 + ap_2$$
$$q_2 = (1-a)p_3 + ap_4$$
$$q = (1-b)q_1 + bq_2$$

Image Interpolation (Bilinear)



Input image



Output image

```
for(h=0; h<6; h++) for(w=0; w<7; w++)
{
        .........
        output[h*6+w] = Interpolation(input, w’,h’);
}
```

Image Interpolation

Bilinear Interpolation

low-resolution image (100×100)

high-resolution image (400×400)

Image Interpolation

Bicubic Interpolation

low-resolution image (100×100)

high-resolution image (400×400)

Image Interpolation

Edge-Directed Interpolation (Li&Orchard'2000)

→

low-resolution image (100×100)

high-resolution image (400×400)

Image Artifact

Original image

Interpolated image

Assignment #0

- Create the following images using C/C++.