Introduction to Image Processing

Prof. Alexandre Zaghetto

http://alexandre.zaghetto.com

zaghetto@unb.br

University of Brasília

Department of Computer Science

LISA: Laboratory of Imagens, Signals and Acoustics

Topic 02
Digital Image Fundamentals

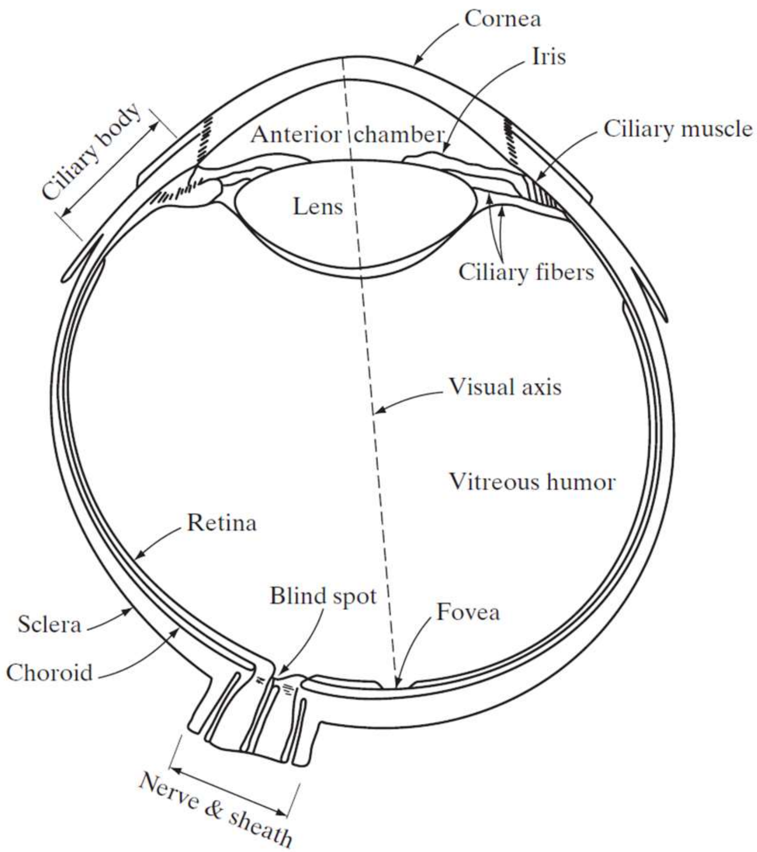
1. Elements of Visual Perception

• Digital image processing field is built on a foundation of mathematical and probabilistic formulations.

• Human intuition and analysis play a central role in the choice of one technique versus another ➔ Subjective, visual judgments.

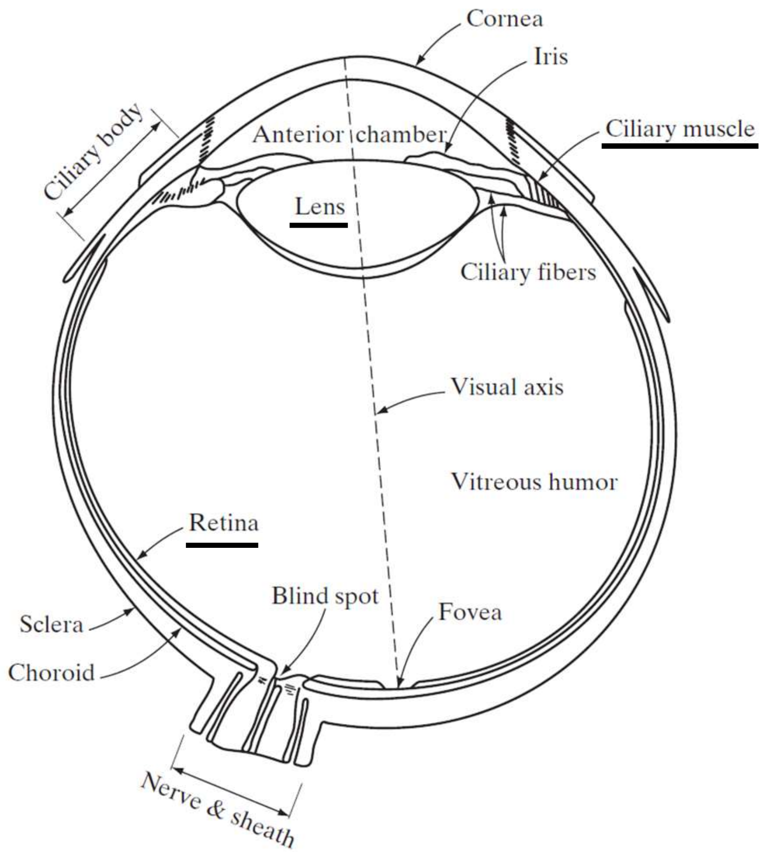
1.1 Structure of the Human Eye

• The **eye** is nearly a sphere, with an average diameter of approximately 20 mm.



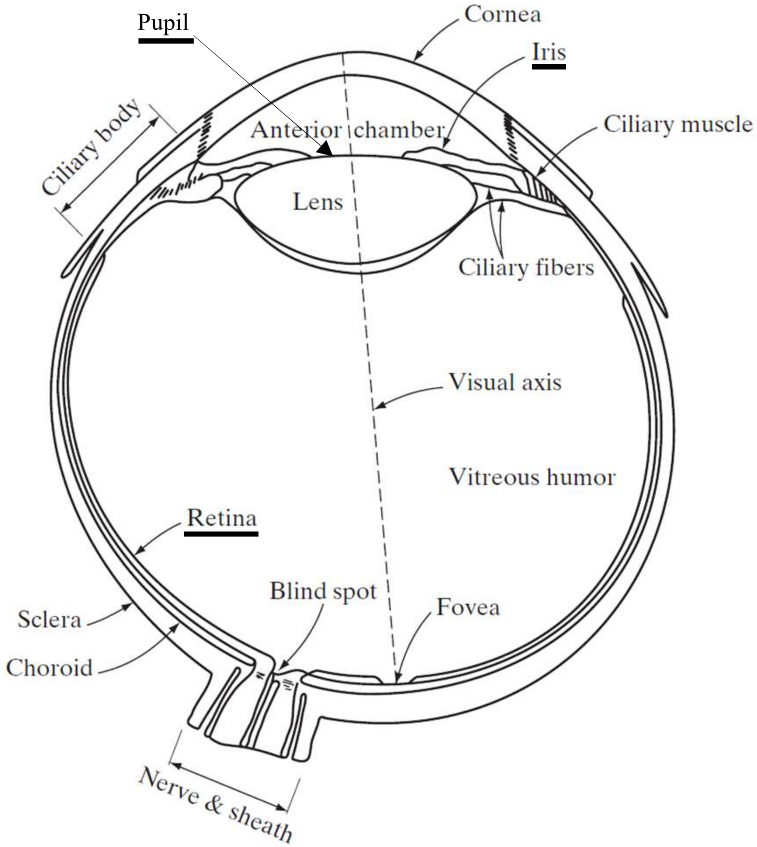
1.1 Structure of the Human Eye

• The **ciliary muscle** changes the shape of the **lens** in order to focus light on the **retina**.



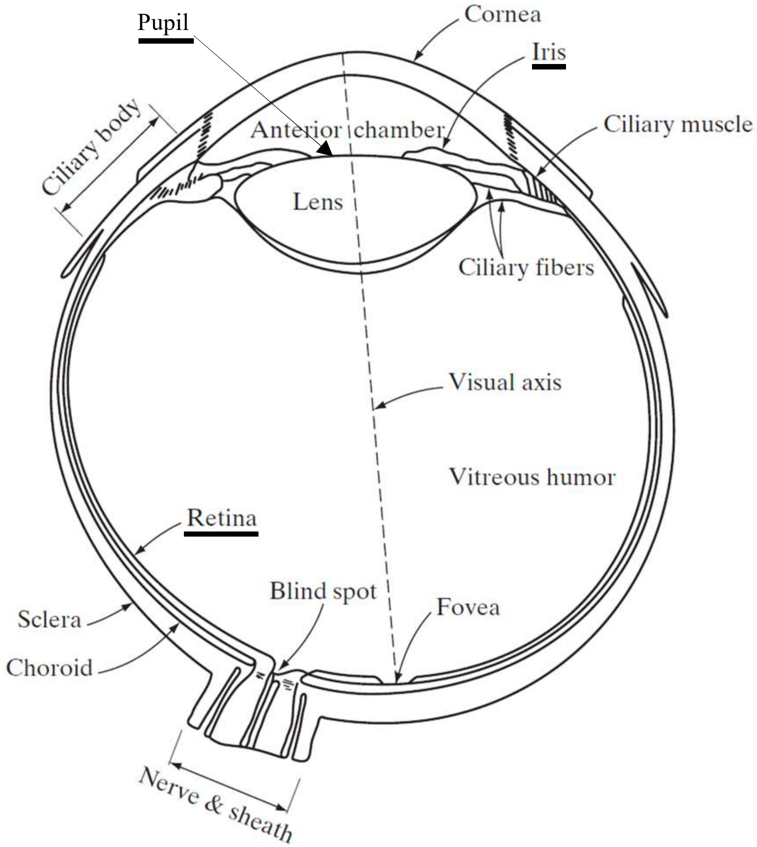
1.1 Structure of the Human Eye

- The **iris** is responsible for controlling the diameter and size of the **pupil** and thus the amount of light reaching the **retina**.



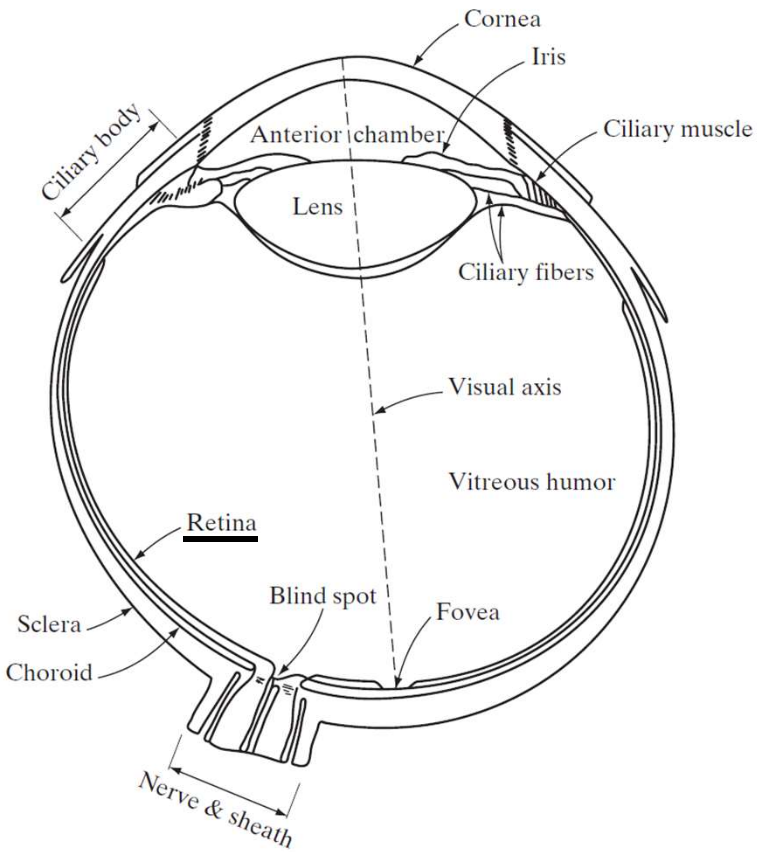
1.1 Structure of the Human Eye

• The **pupil** is a hole located in the centre of the **iris** that allows light to strike the **retina**.



1.1 Structure of the Human Eye

• The **retina** is the innermost membrane of the eye. Light from an object outside the eye is imaged on the retina.



1.1 Structure of the Human Eye

• Vision is afforded by the distribution of discrete light receptors over the surface of the retina.

- There are two classes of receptors: **cones** and **rods**.

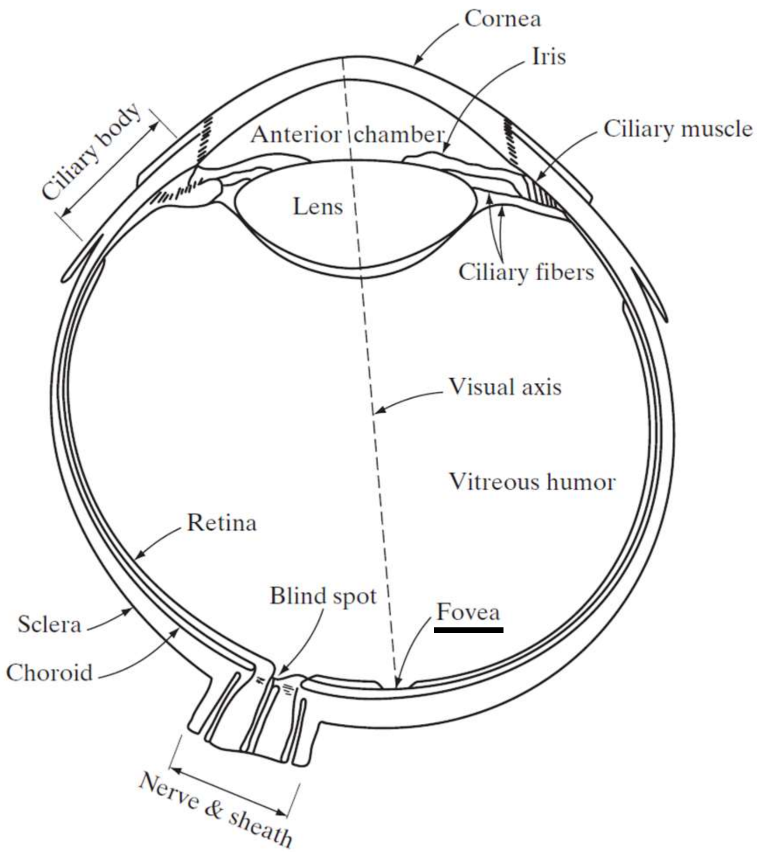
• The cones in each eye number between 6 and 7 million. They are located primarily in the central portion of the retina, called the **fovea.**

- Humans can resolve **fine details** with these cones.
- Highly sensitive to color.
- Cone vision is called **photopic** or **bright-light vision.**

• We can view the fovea as a square sensor array of size 1.5 mm x 1.5 mm.

1.1 Structure of the Human Eye

• The cones in each eye number between 6 and 7 million. They are located primarily in the **fovea.**



1.1 Structure of the Human Eye

- The number of rods is much larger: 75 to 150 million distributed over the retinal surface.

  - Rods serve to give a general, **overall picture** of the field of view.

  - They are not involved in color vision.

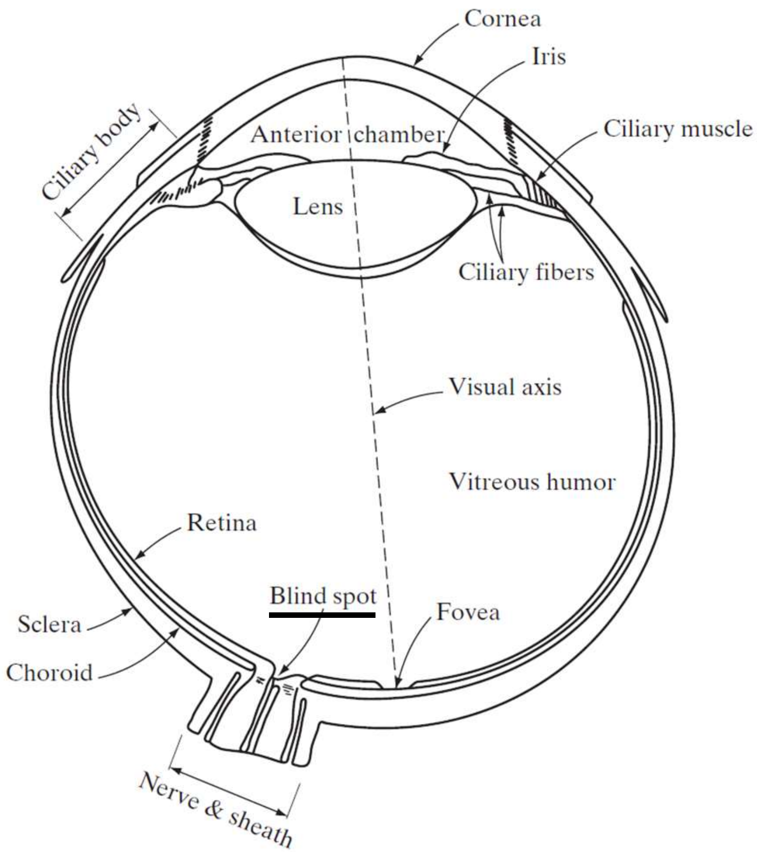
  - Sensitive to low levels of illumination (**scotopic** or **dim-light vision**).

  - For example, objects that appear brightly colored in daylight when seen by moonlight appear as colorless forms because only the rods are stimulated

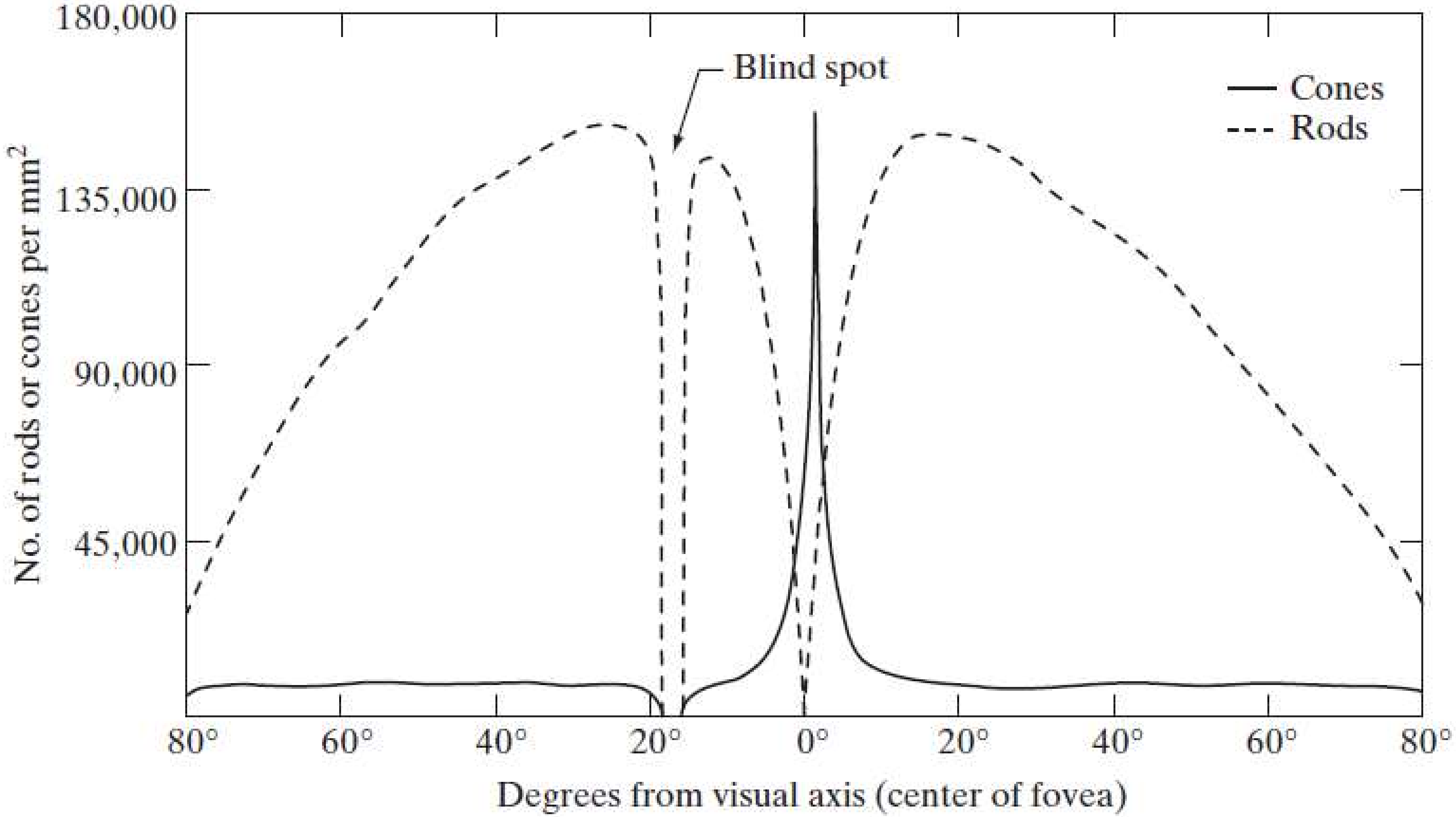
- **Blind spot**: region of emergence of the optic nerve from the eye (the absence of receptors).

1.1 Structure of the Human Eye

- **Blind spot**: region of emergence of the optic nerve from the eye (the absence of receptors).

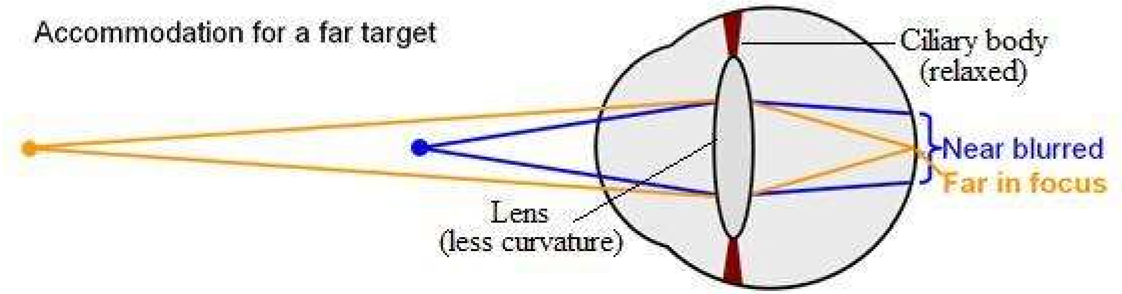


1.1 Structure of the Human Eye

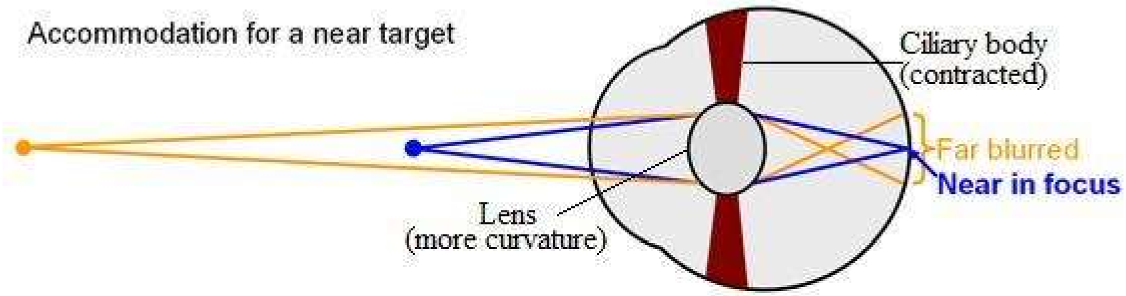


1.2 Image Formation in the Eye

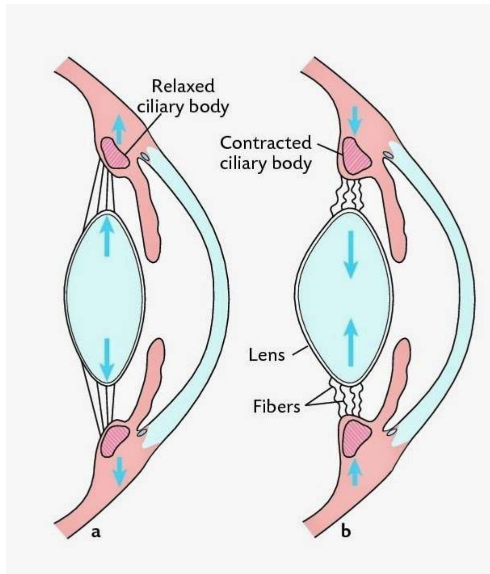
- **Focal length**: distance between the center of the lens and the retina. Varies from 14 mm (maximum refractive power) to about 17 mm (minimum refractive power).



minimum refractive power

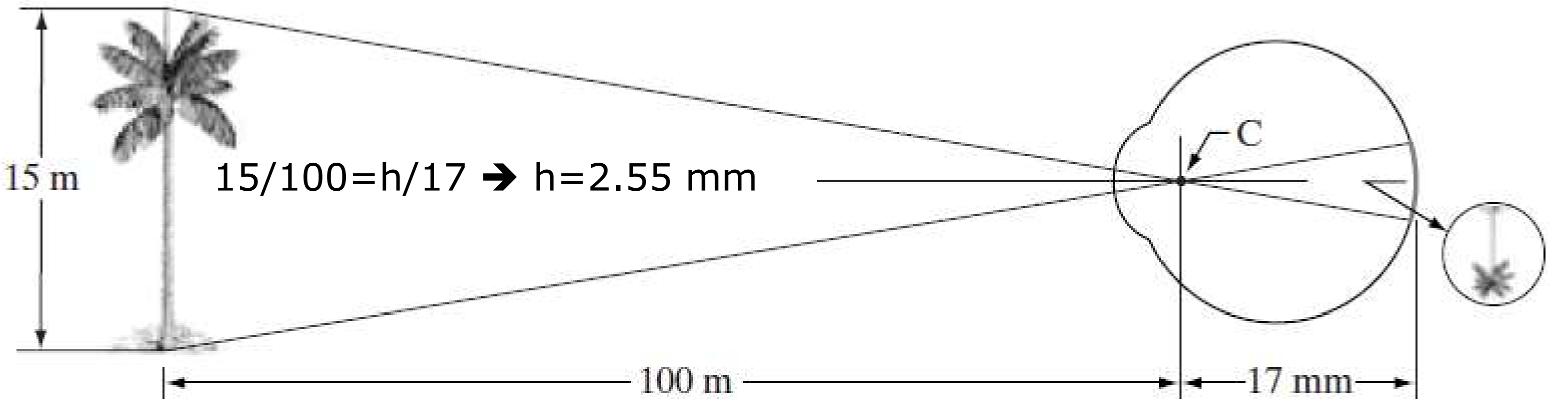


maximum refractive power



1.2 Image Formation in the Eye

- **Focal length**: distance between the center of the lens and the retina. Varies from 14 mm to about 17 mm.



- The retinal image is reflected primarily in the area of the fovea.

- Perception then takes place by the relative excitation of light receptors, which transform radiant energy into electrical impulses that are ultimately decoded by the brain.

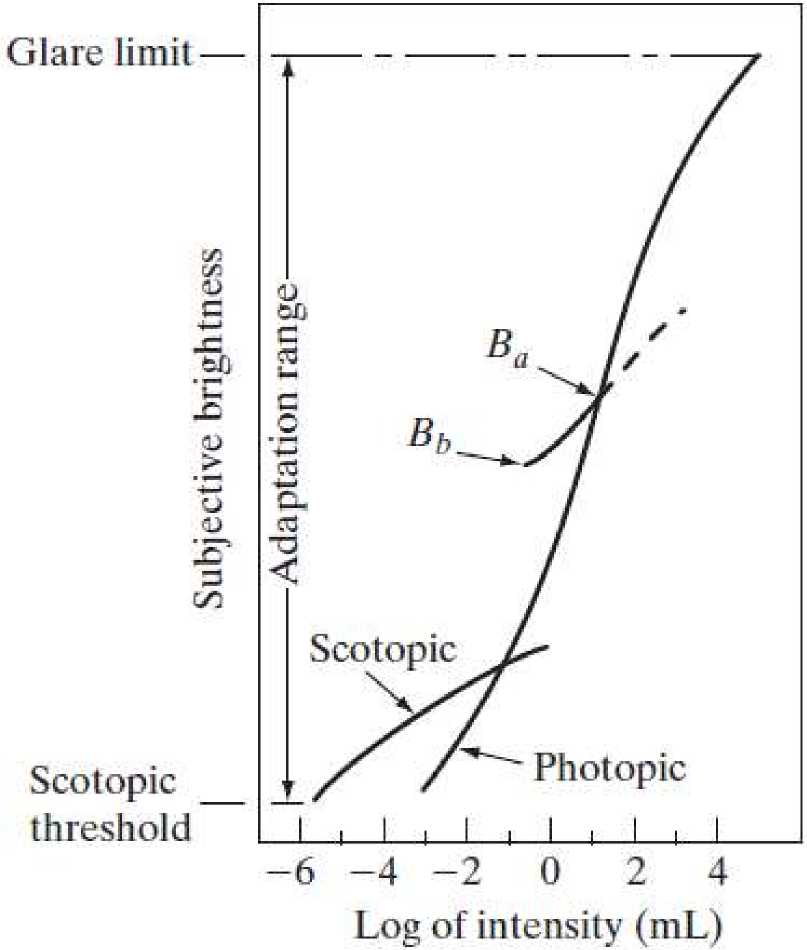
1.3 Brightness Adaptation and Discrimination

- The range of light intensity levels to which the human visual system can adapt is enormous:

  - on the order of 10^{10} from the **scotopic** threshold to the **glare** limit.

- Experimental evidence indicates that subjective brightness is a logarithmic function of the light intensity incident on the eye.

1.3 Brightness Adaptation and Discrimination



(mL – millilambert)

- The visual system cannot operate over such a range simultaneously.

- Rather, it accomplishes this large variation by changes in its overall sensitivity, a phenomenon known as **brightness adaptation**.

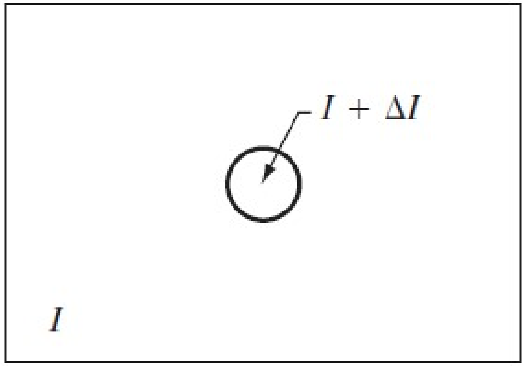
- For any given set of conditions, the current sensitivity level of the visual system is called the **brightness adaptation level** (e.g. B_a).

- In this example, stimuli below B_b are perceived as indistinguishable blacks.

1.3 Brightness Adaptation and Discrimination

- The ability of the eye to discriminate between changes in light intensity at any specific adaptation level is also of considerable interest.

- A classic experiment: *background* + *flashes.*



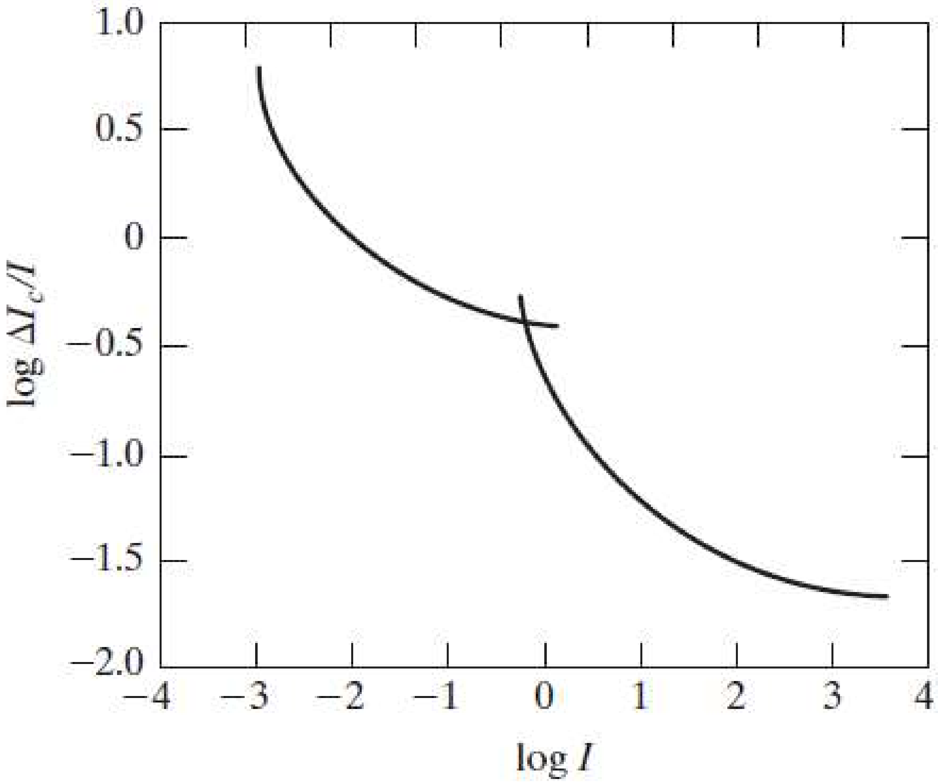
- *Weber ratio*: $\Delta I_c / I$ ➔ ΔI_c is the increment of illumination discriminable 50% of trials with background illumination I. A small value means "good" brightness discrimination, a large value means "poor" brightness discrimination.

1.3 Brightness Adaptation and Discrimination

- Typical plot of Weber ratio as a function of intensity.

Brightness discrimination is poor (the Weber ratio is large) at low levels of illumination. Vision is carried out by activity of the rods

Brightness discrimination improves (the Weber gets smaller) at high levels of illumination. Vision is carried out by activity of cones.



1.3 Brightness Adaptation and Discrimination

- Another experiment: background illumination is held constant and the intensity of the other source is now allowed to vary incrementally from never being perceived to always being perceived.

  - The typical observer can discern a total of one to two dozen different intensity changes.

- Roughly, this result is related to the number of different intensities a person can see at any one point in a monochrome image.

- This result does not mean that an image can be represented by such a small number of intensity

  - As the eye roams about the image, the average background changes, thus allowing a *different* set of incremental changes to be detected

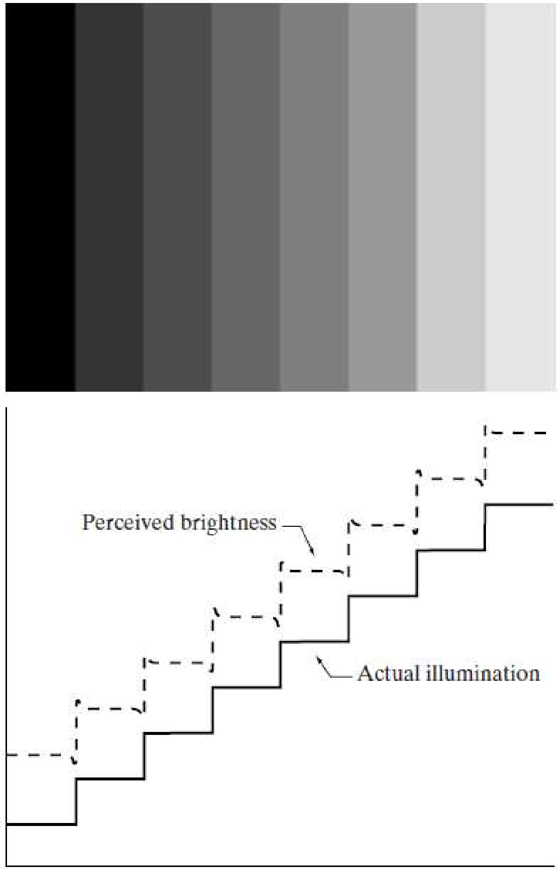
1.3 Brightness Adaptation and Discrimination

• However, perceived brightness is not a simple function of intensity:

- Mach Bands.
- Simultaneous contrast.
- Other examples.

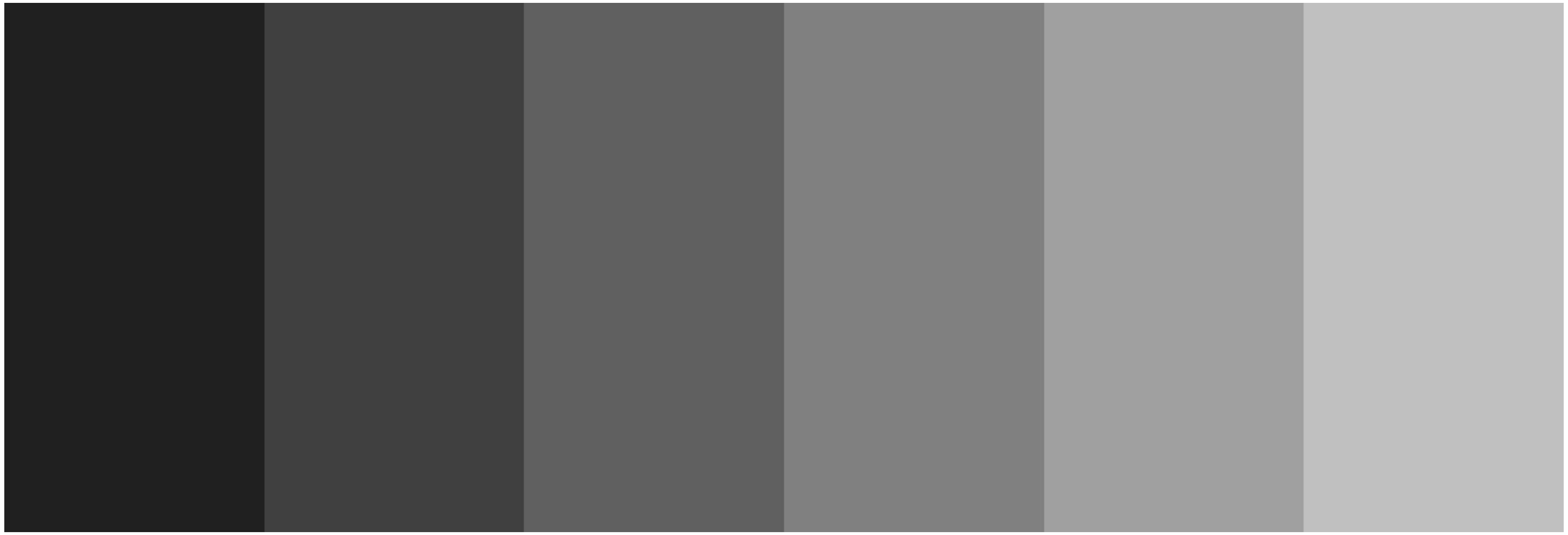
1.3 Brightness Adaptation and Discrimination

- Mach bands: undershoot or overshoot around the boundary of regions of different intensities



1.3 Brightness Adaptation and Discrimination

- Mach bands: undershoot or overshoot around the boundary of regions of different intensities

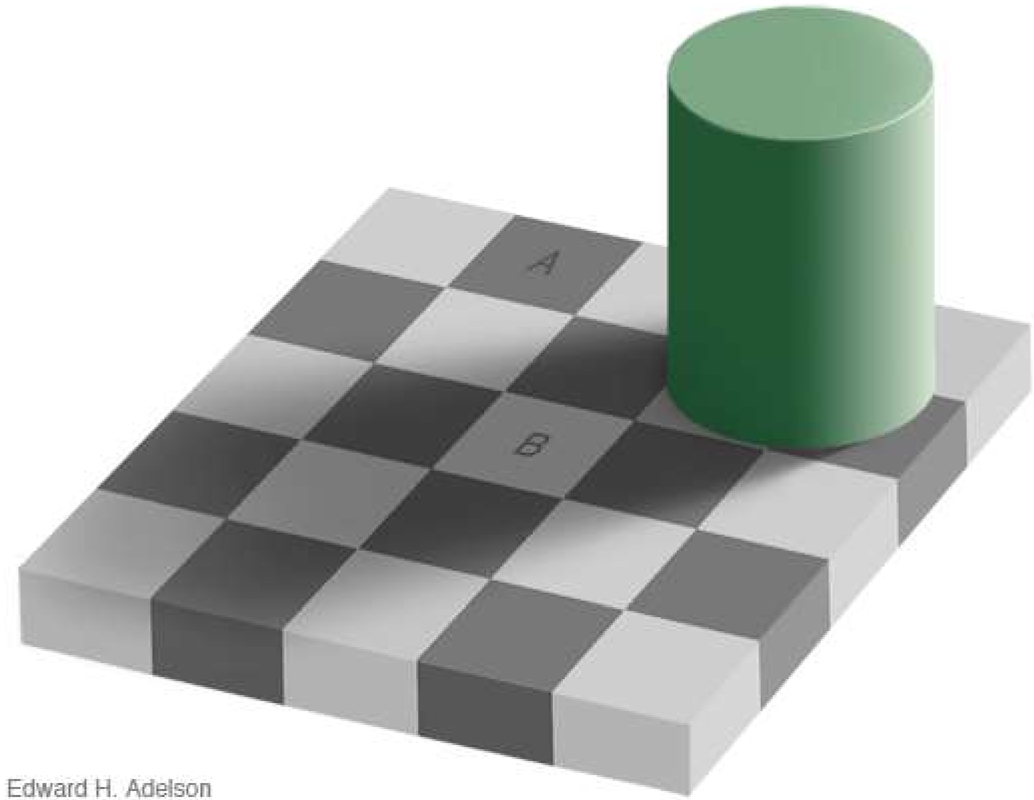


1.3 Brightness Adaptation and Discrimination

• Simultaneous contrast: perceived brightness does not depend simply on its intensity.

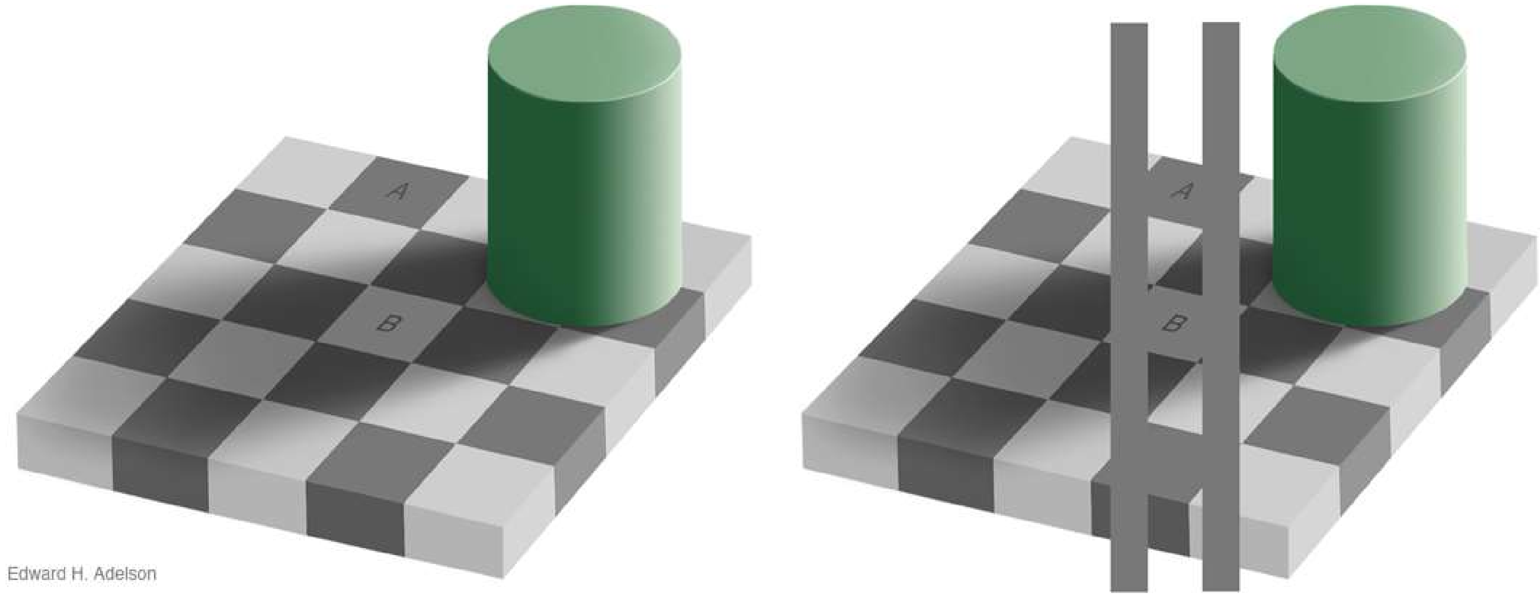
1.3 Brightness Adaptation and Discrimination

- Intensities of A and B are the same?



1.3 Brightness Adaptation and Discrimination

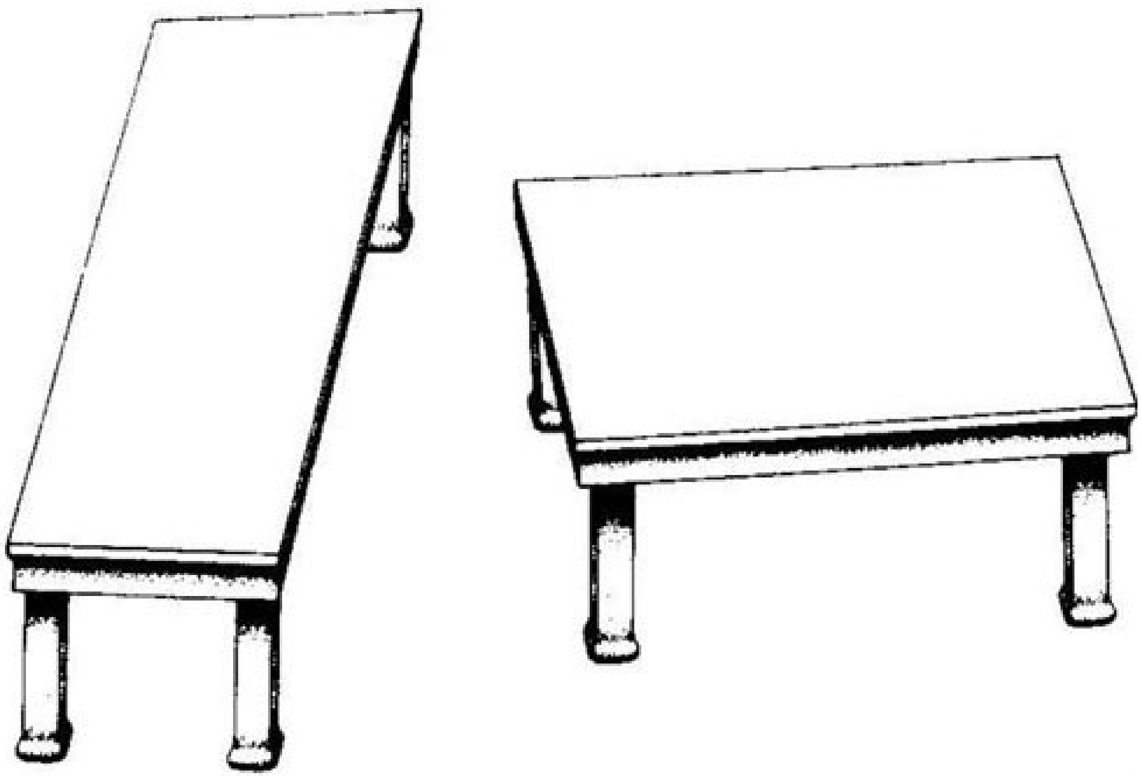
- Intensities of A and B are the same?



http://web.mit.edu/persci/people/adelson/checkershadow_illusion.html

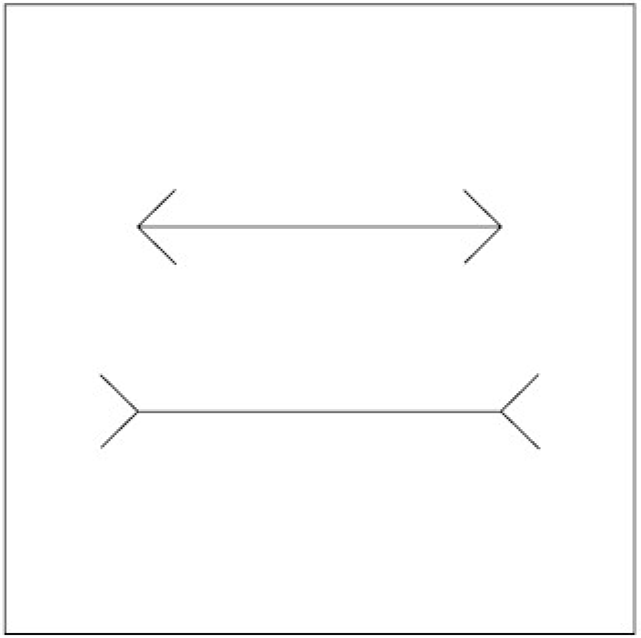
1.4 Optical Illusions

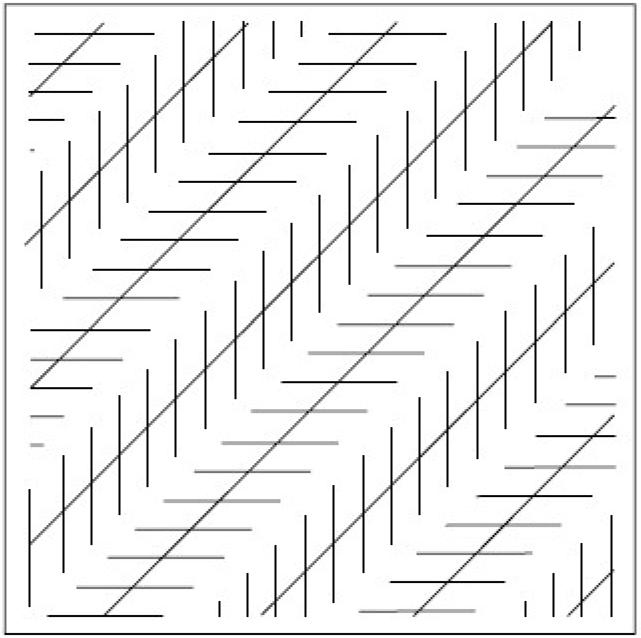
- Turning the Tables: the tabletop on the left is identical in shape and size to the one on the right.



1.4 Optical Illusions

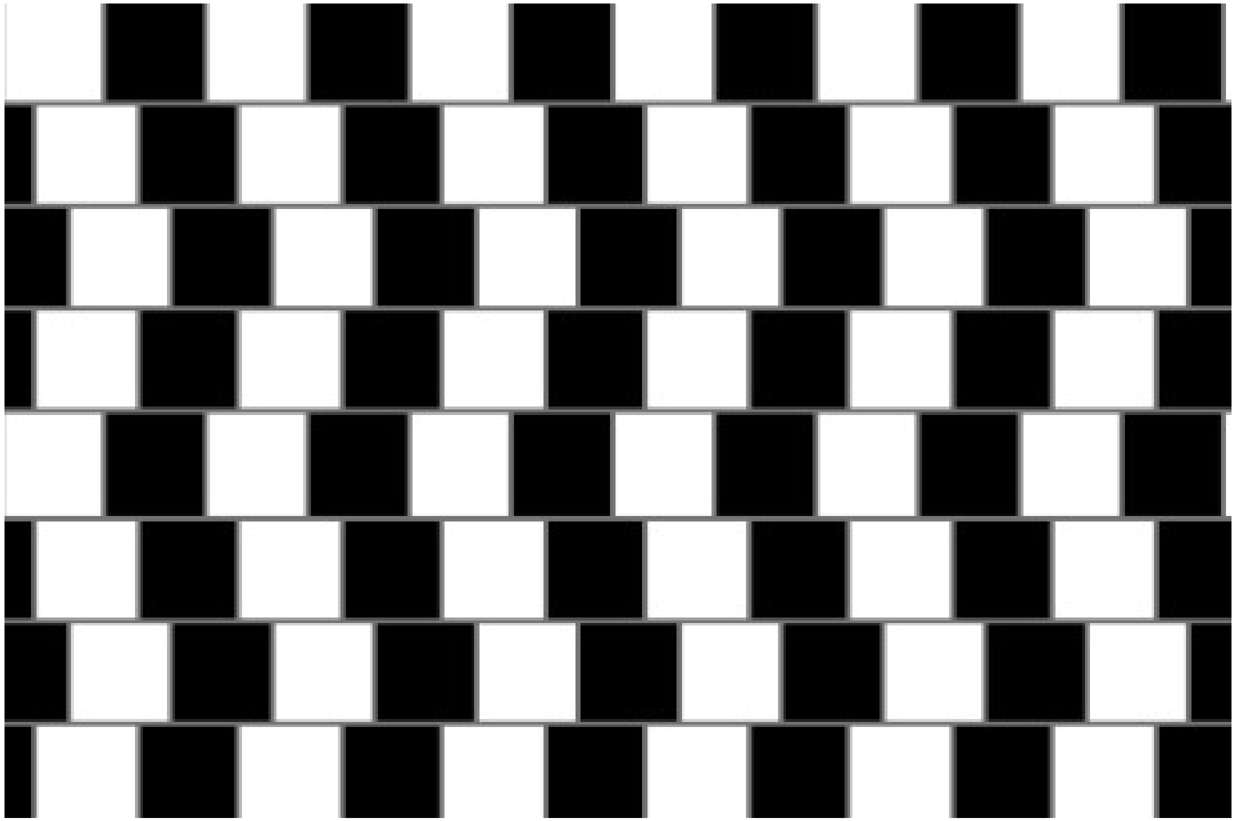
• Left: The two horizontal line segments are of the same length, but one appears shorter than the other; Right: all lines that are oriented at 45° are equidistant and parallel.





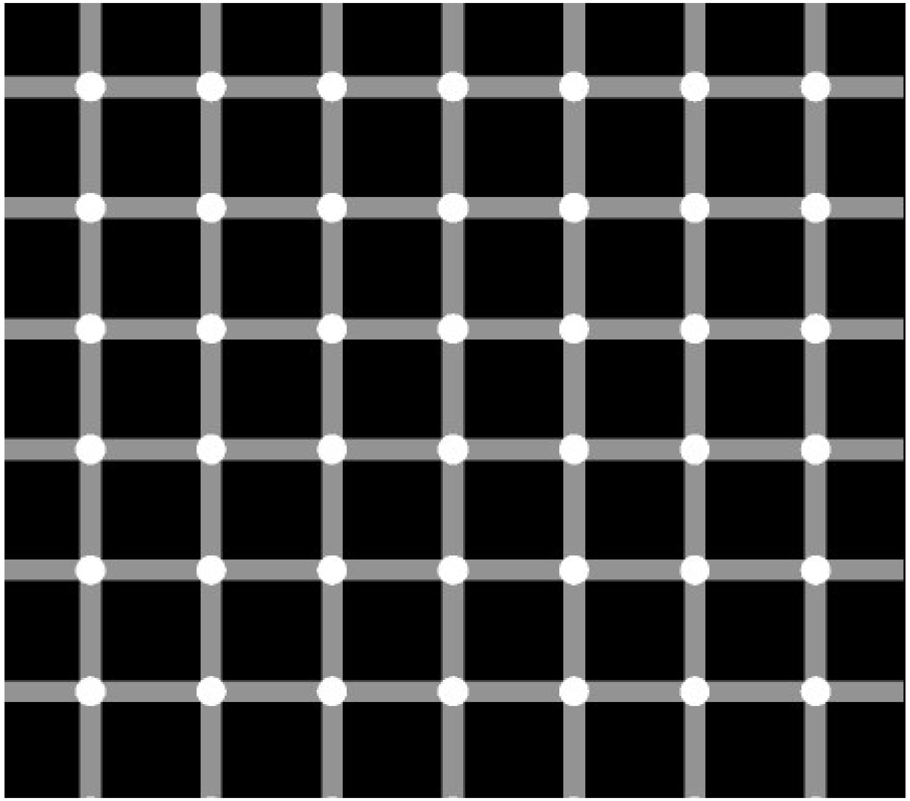
1.4 Optical Illusions

- Lines are parallel.



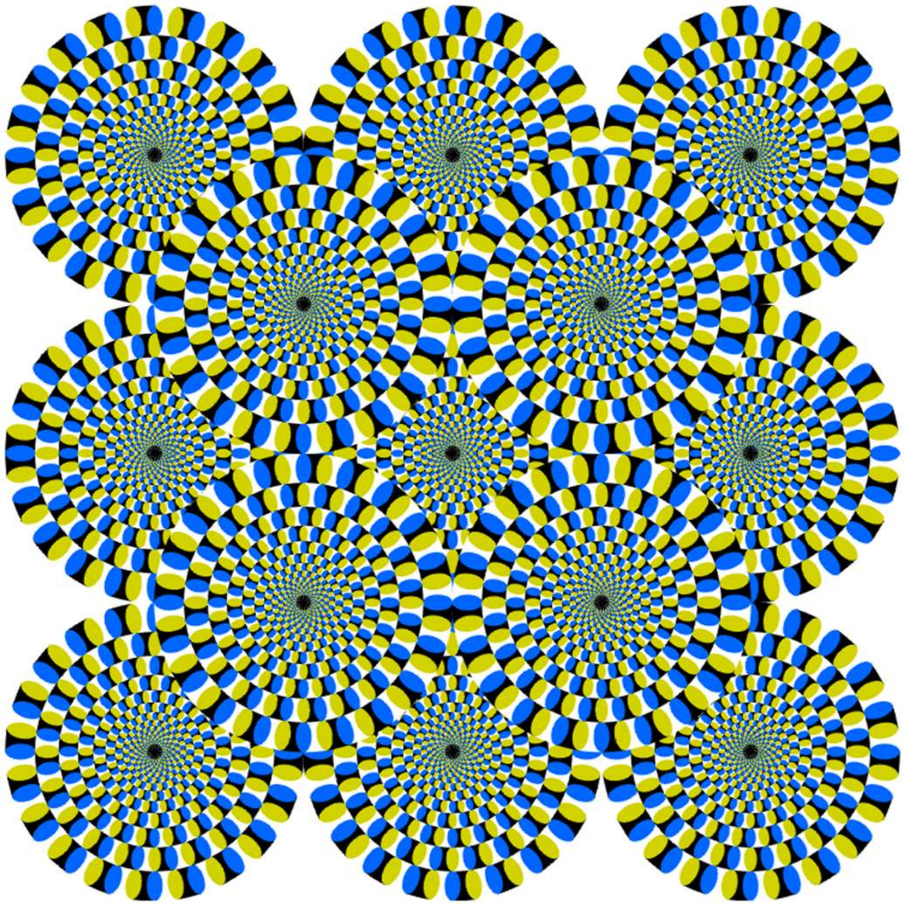
1.4 Optical Illusions

- Are the dots in between the squares white, black or grey?

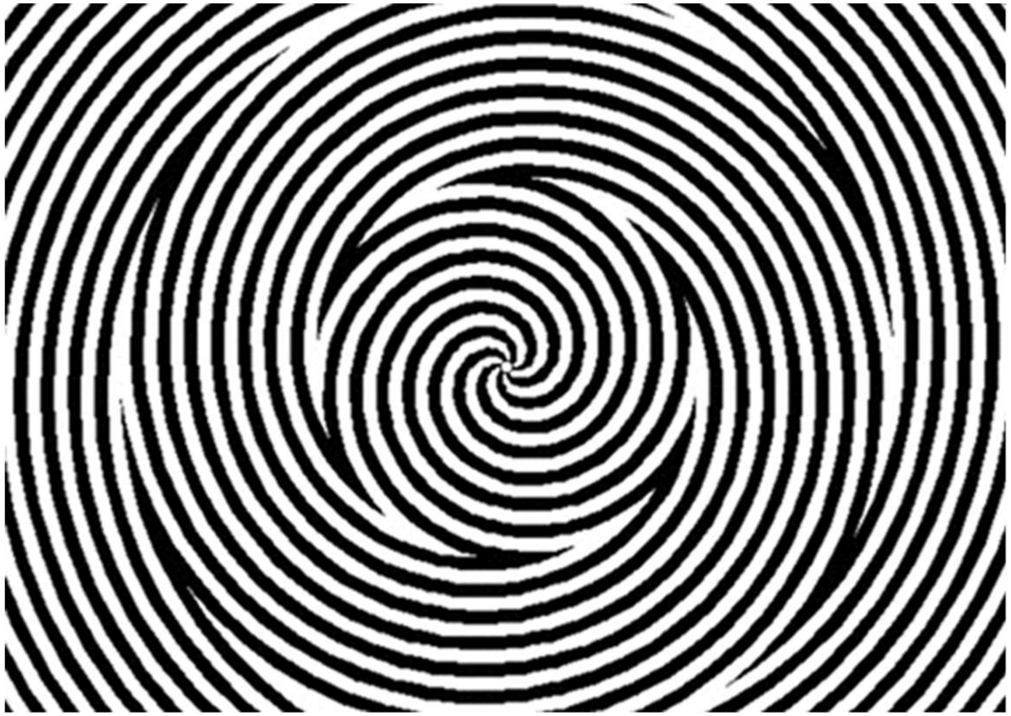


1.4 Optical Illusions

- Rotation snakes.

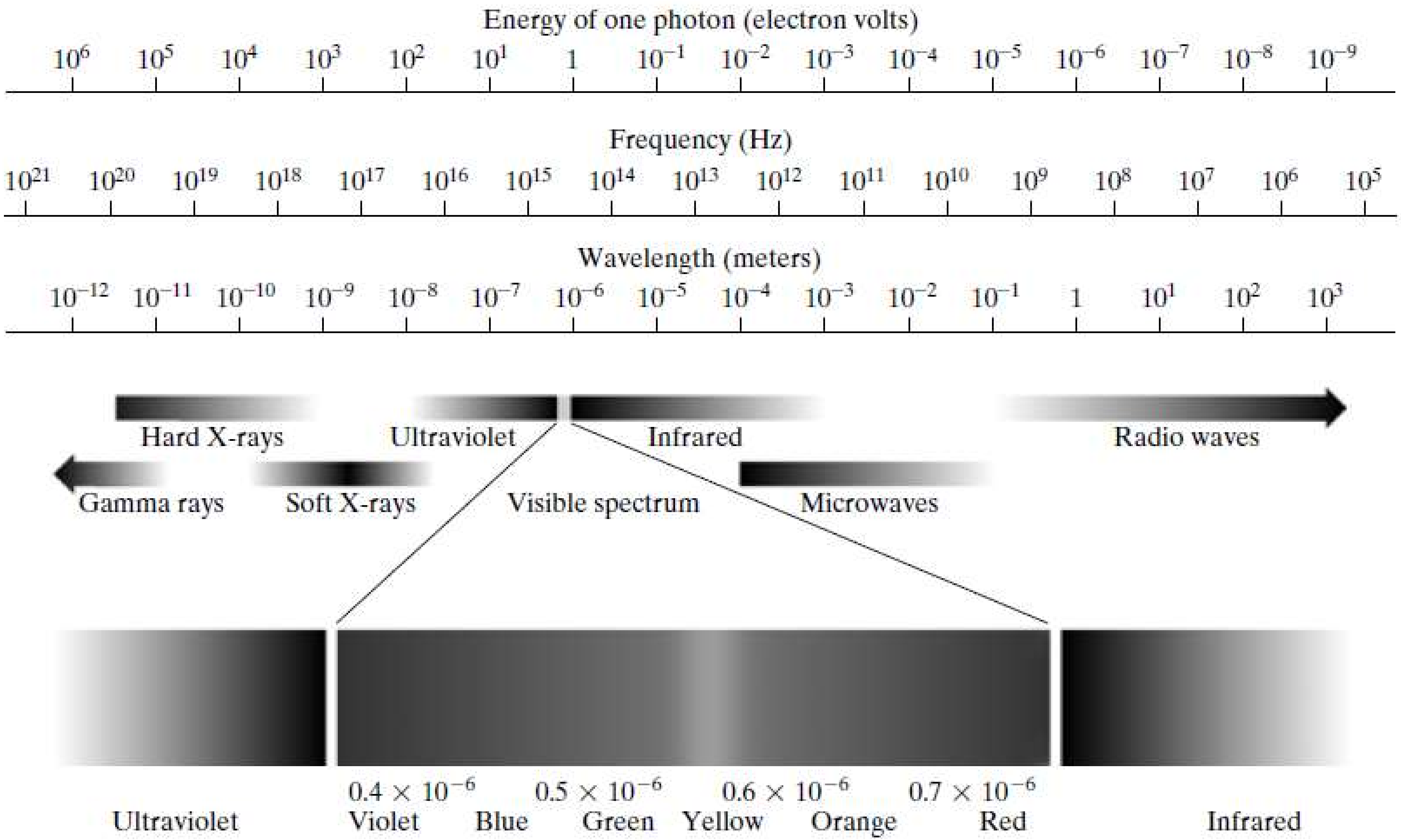


1.4 Optical Illusions



1.4 Optical Illusions

2. Light and the Electromagnetic Spectrum

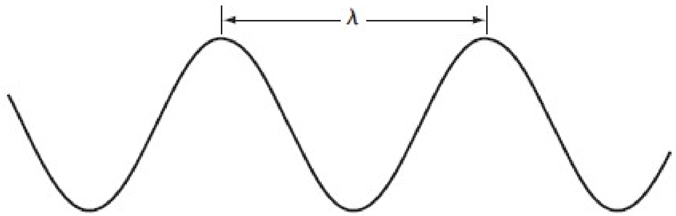


2. Light and the Electromagnetic Spectrum

- The electromagnetic spectrum can be expressed in terms of wavelength, frequency, or energy.

- Wavelength λ (m) and frequency ν (Hz) are related by the expression

$$\lambda = \frac{c}{\nu}$$



where c is the speed of light (2.988 x 10^8 m/s).

- The energy (eV) is given by

$$E = h\nu$$

where h (4,136 x 10^{-15} eV) is Plank's constant.

2. Light and the Electromagnetic Spectrum

- Electromagnetic waves can be visualized as propagating sinusoidal waves with wavelength λ.

- Or they can be thought of as a stream of massless particles, each traveling in a wavelike pattern and moving at the speed of light.

  - Each massless particle contains a certain amount (or bundle) of energy.

    - ✓ Each bundle of energy is called a photon.

- Energy is proportional to frequency.

  - The higher frequency the more energy per photon.

2. Light and the Electromagnetic Spectrum

• Light is a particular type of electromagnetic radiation that can be sensed by the human eye.

0,43 µm 0,79 µm

• **For convenience**, the color spectrum is divided into six broad regions: violet, blue, green, yellow, orange, and red.

• The colors that humans perceive in an object are determined by the nature of the light reflected from the object.

2. Light and the Electromagnetic Spectrum

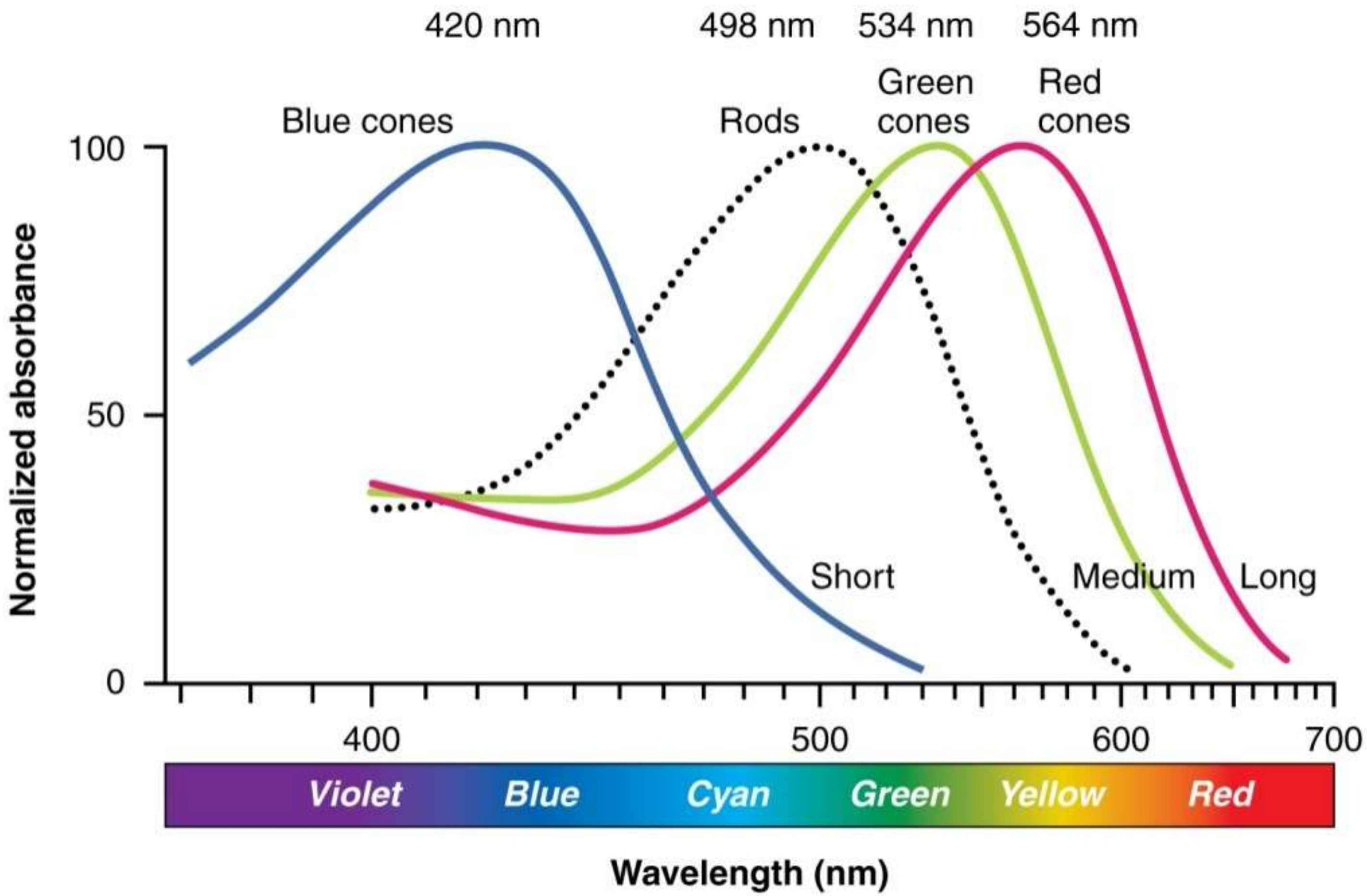
- Body that reflects light and is relatively balanced in all visible wavelengths appears white to the observer.

- However, a body that favors reflectance in a limited range of the visible spectrum exhibits some shades of color.

- Light that is void of color is called **achromatic** or **monochromatic light**.

- The term **gray level** generally is used to describe monochromatic intensity because it ranges from black, to grays, and finally to white.

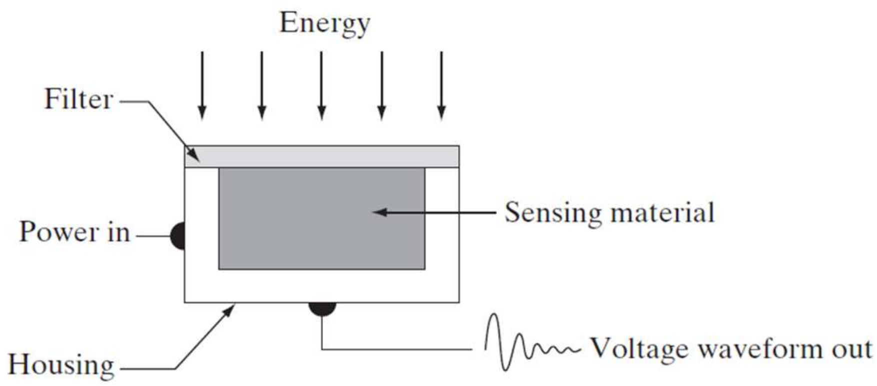
2. Light and the Electromagnetic Spectrum

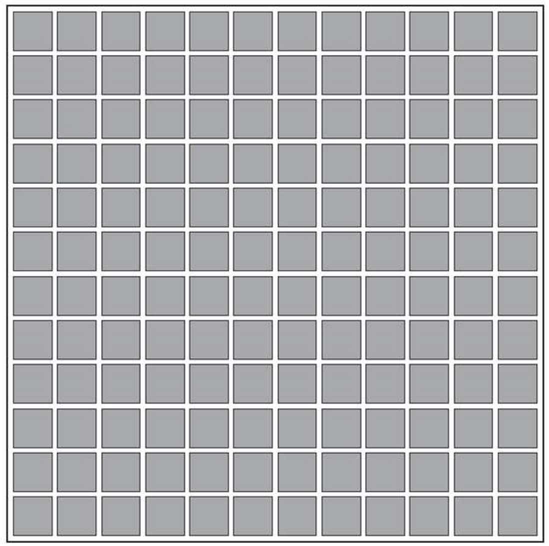


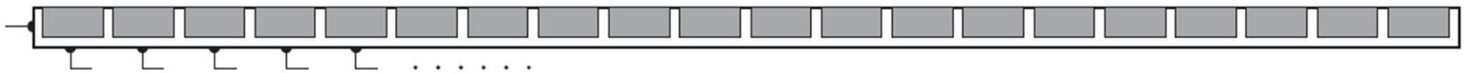
3. Image Sensing and Acquisition

- Incoming energy is transformed into a voltage. A digital quantity is obtained from each sensor by digitizing its response.

- Sensor arrangements used to transform illumination energy into digital images:







3.1 Image Acquisition Using a Single Sensor

- Single sensor:

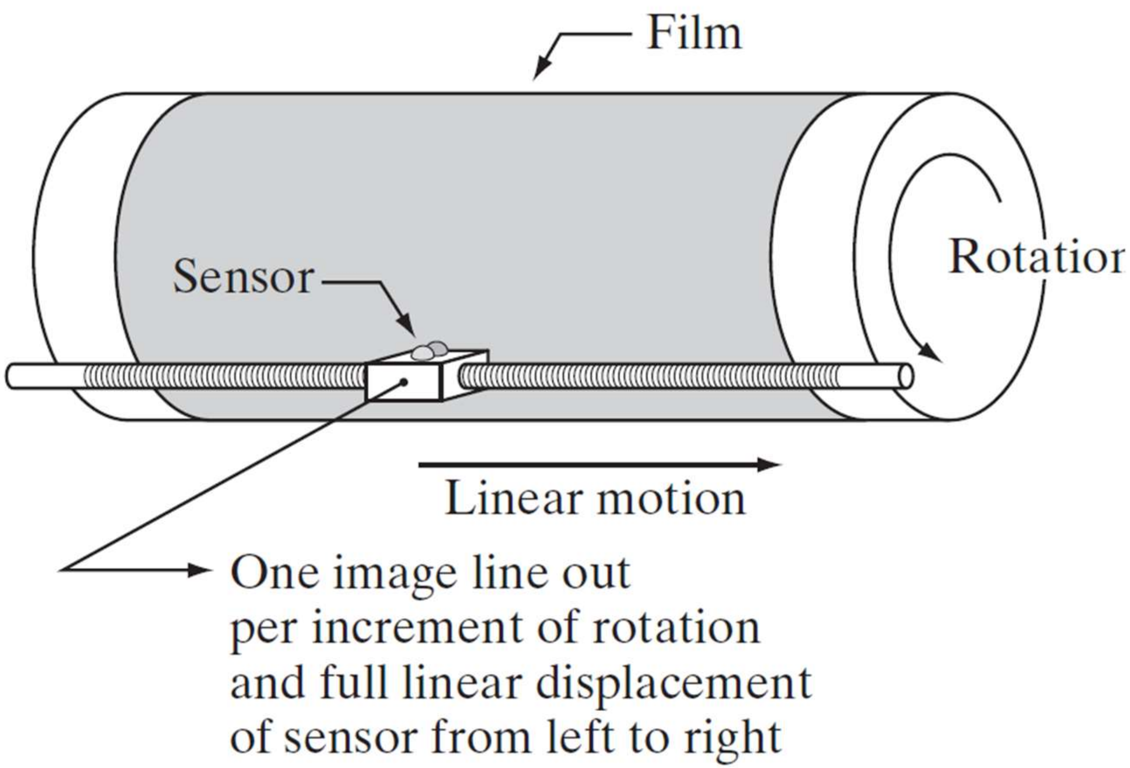


FIGURE 2.13 Combining a single sensor with motion to generate a 2-D image.

3.2 Image Acquisition Using Sensor Strips

- In-line arrangement of sensors:

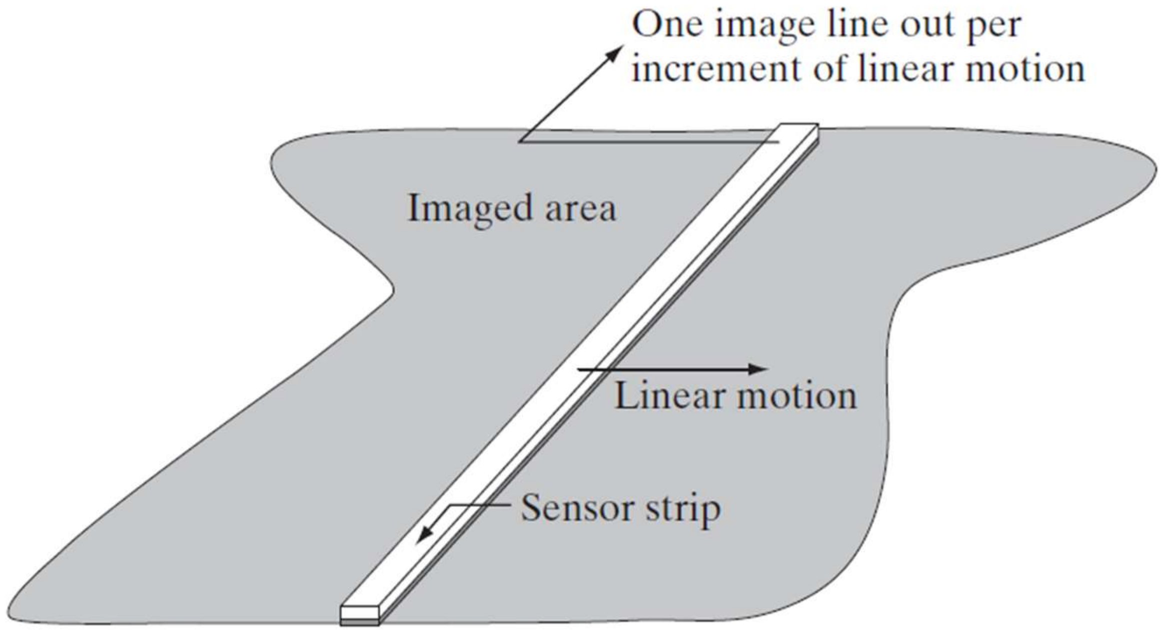


FIGURE 2.14 (a) Image acquisition using a linear sensor strip.

3.3 Image Acquisition Using Sensor Arrays

- Sensors arranged in the form of a 2-D array.

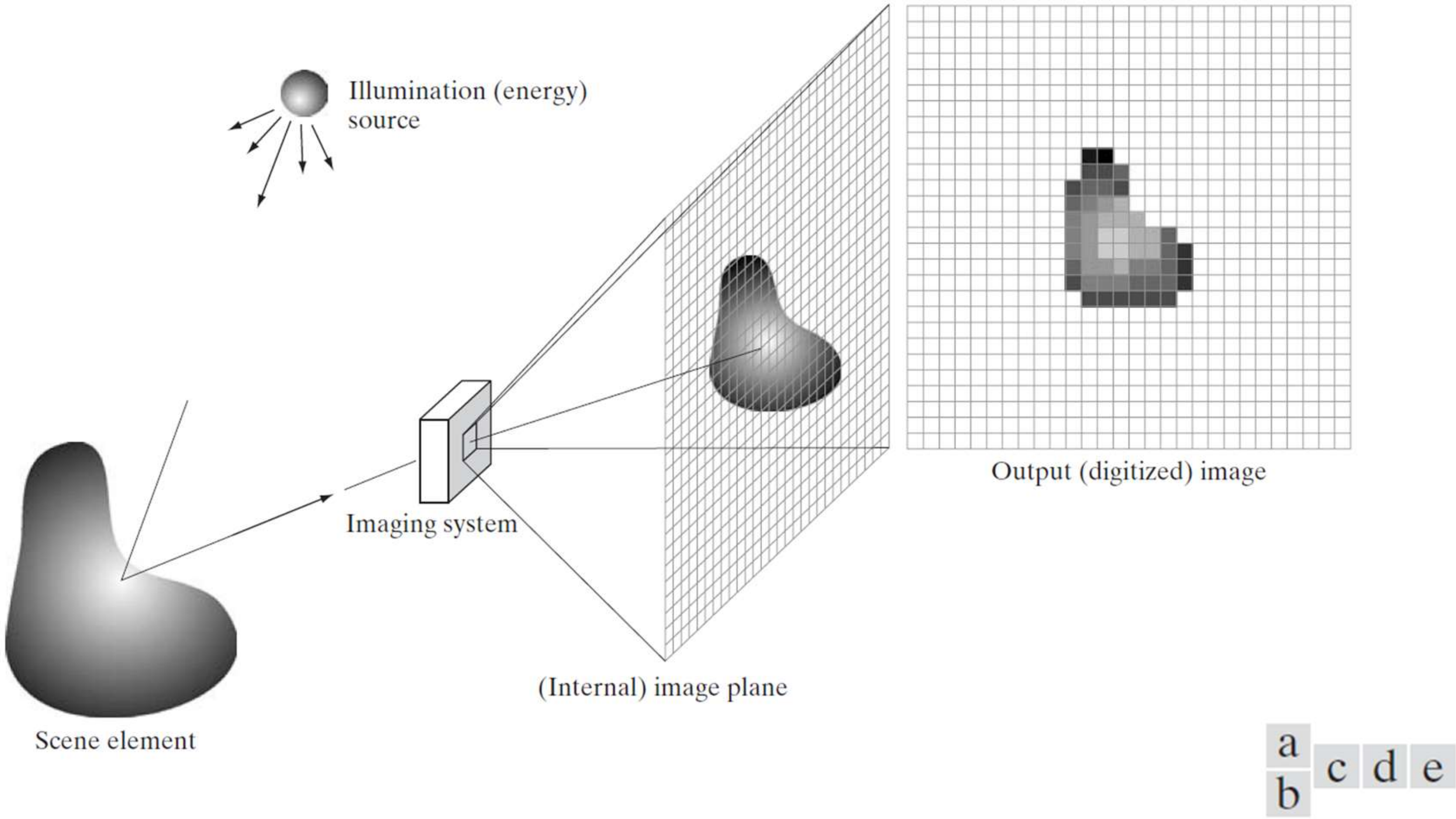


FIGURE 2.15 An example of the digital image acquisition process. (a) Energy ("illumination") source. (b) An element of a scene. (c) Imaging system. (d) Projection of the scene onto the image plane. (e) Digitized image.

3.4 A Simple Image Formation Model

- As introduced before, we denote images by two dimensional functions of the form $f(x,y)$.

- The function $f(x,y)$ may be characterized by two components: (1) the amount of source **illumination incident** on the scene, and (2) the amount of **illumination reflected** by the objects in the scene.

- These are called the illumination and reflectance components and are denoted by $i(x,y)$ and $r(x,y)$.

$$f(x, y) = i(x, y)r(x, y)$$

$$0 < i(x, y) < \infty \ (\mathrm{lm/m^2})$$

$$0 < r(x, y) < 1.$$

3.4 A Simple Image Formation Model

- Let the intensity (gray level) of a monochrome image at any coordinates *(x_0, y_0)* be denoted by

$$l = f(x_o, y_o)$$

- The interval [L_{min}, L_{max}] is called the gray scale.

$$L_{\min} \leq l \leq L_{\max}$$

- Common practice is to shift this interval numerically to the [0, L-1] interval where 0 is considered black and L-1 is considered white on the gray scale.

4. Image Sampling and Quantization

- To create a digital image, we need to convert the continuous sensed data into digital form. This involves two processes: sampling and quantization.

- An image may be continuous with respect to the x- and y-coordinates, and also in amplitude.

- To convert it to digital form, we have to sample the function in both coordinates and in amplitude.

  - Digitizing the **coordinate values** is called **sampling**.

  - Digitizing the **amplitude value**s is called **quantization**.

4. Image Sampling and Quantization

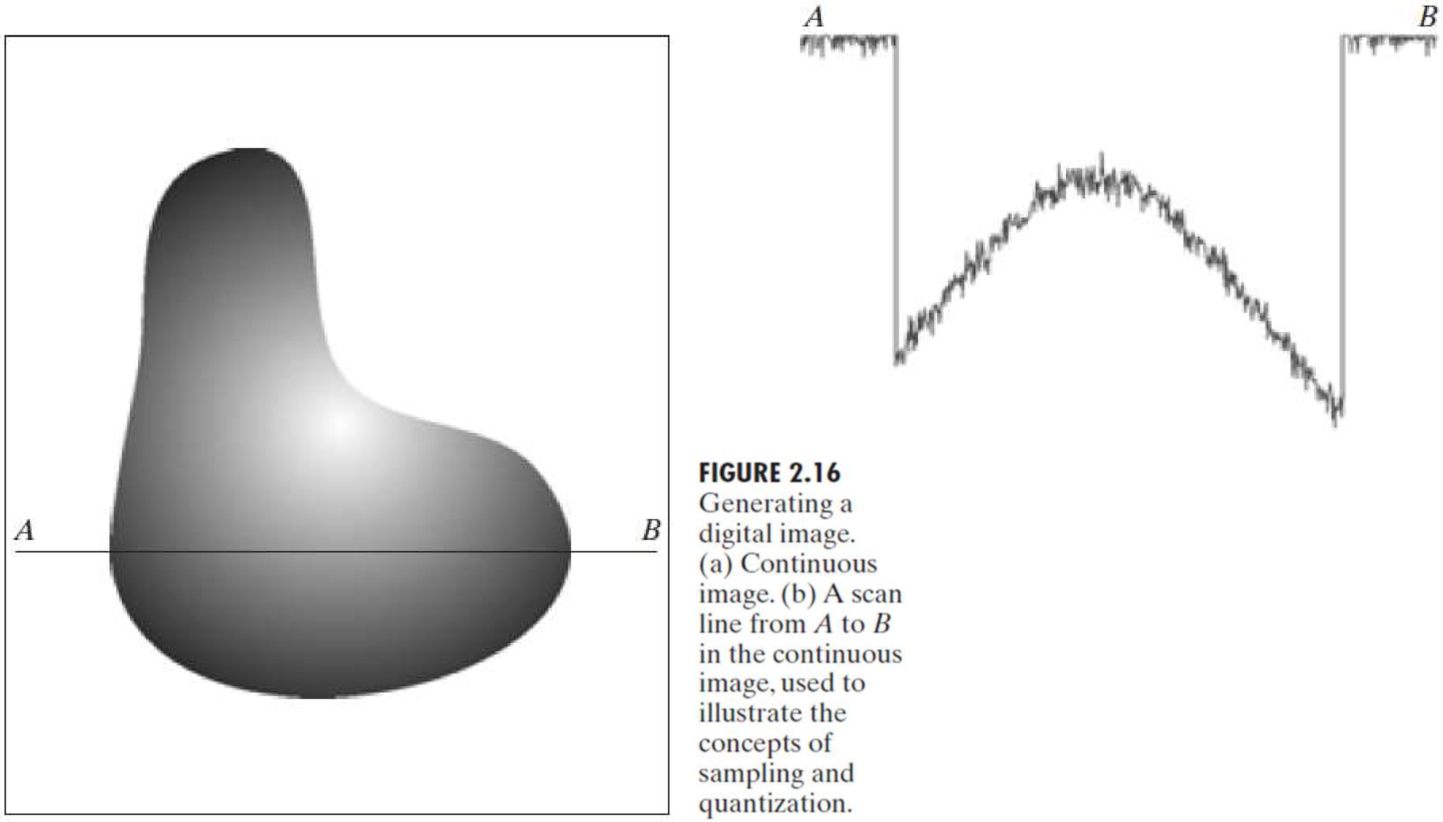


FIGURE 2.16 Generating a digital image. (a) Continuous image. (b) A scan line from *A* to *B* in the continuous image, used to illustrate the concepts of sampling and quantization.

4. Image Sampling and Quantization

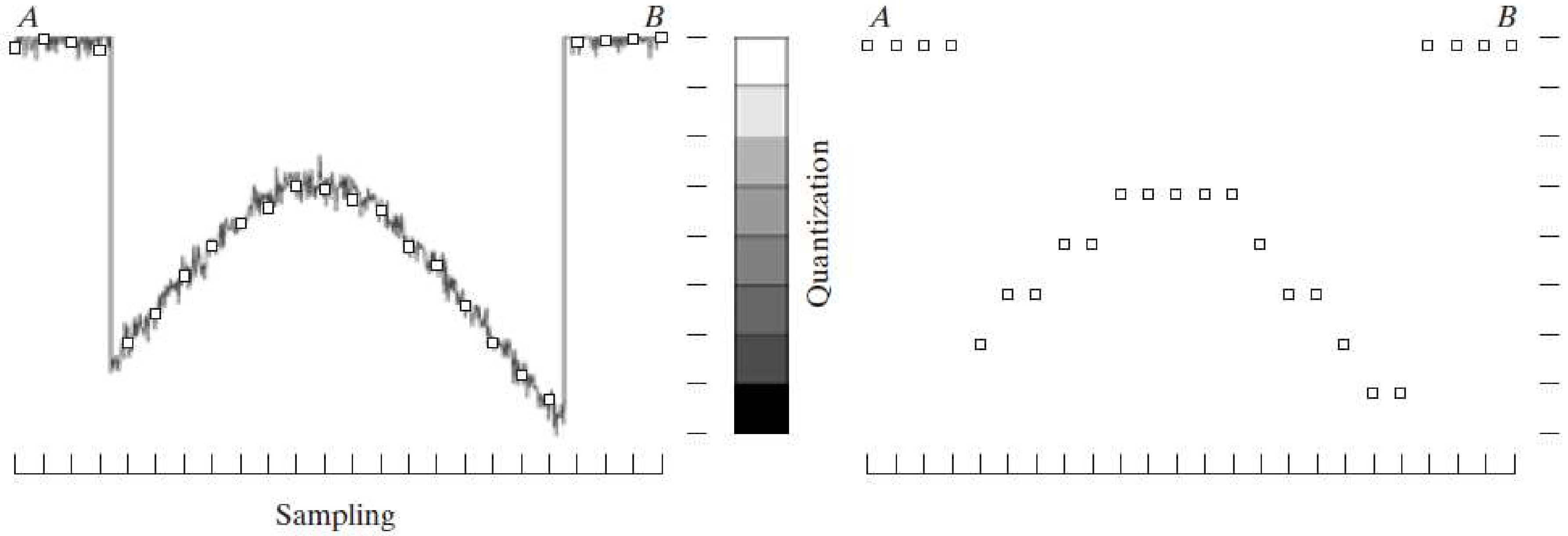
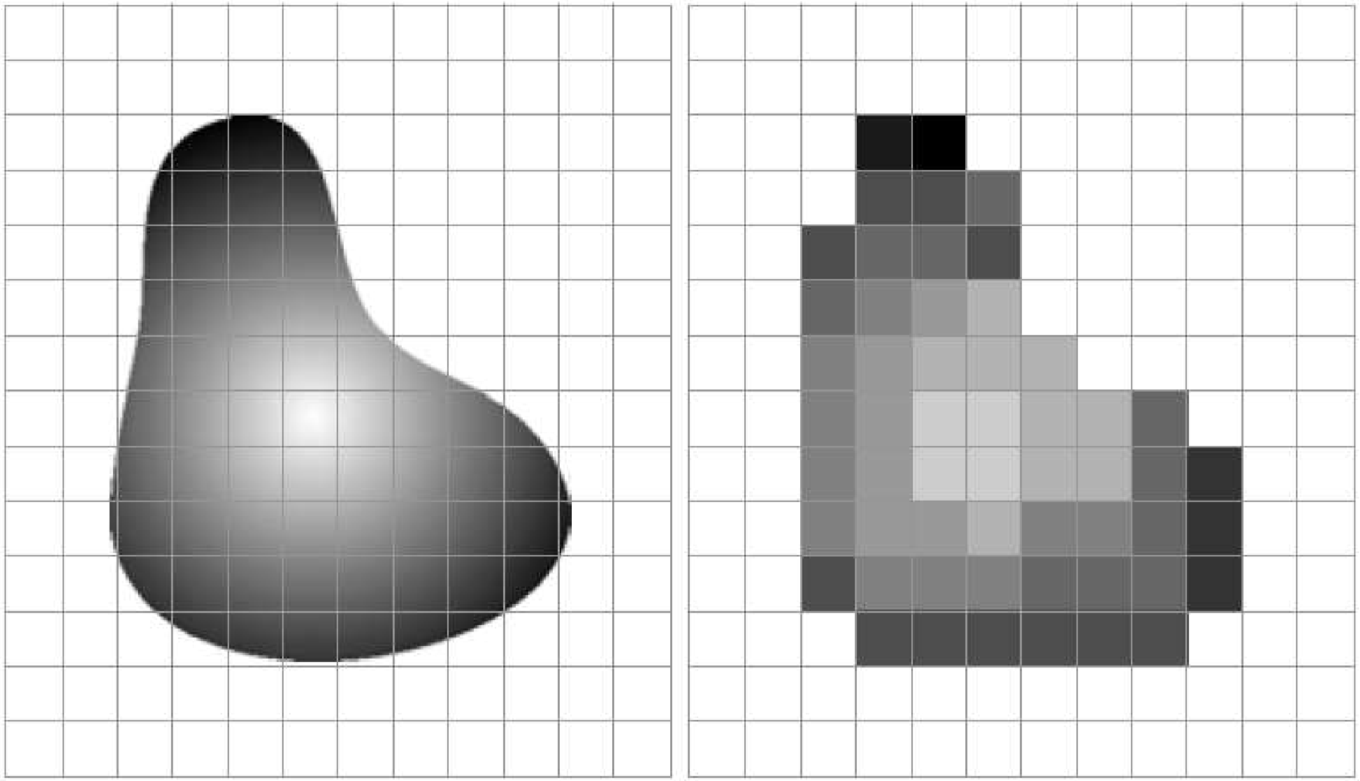


FIGURE 2.16
(c) Sampling and quantization.
(d) Digital scan line.

4. Image Sampling and Quantization



a b

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

4. Image Sampling and Quantization

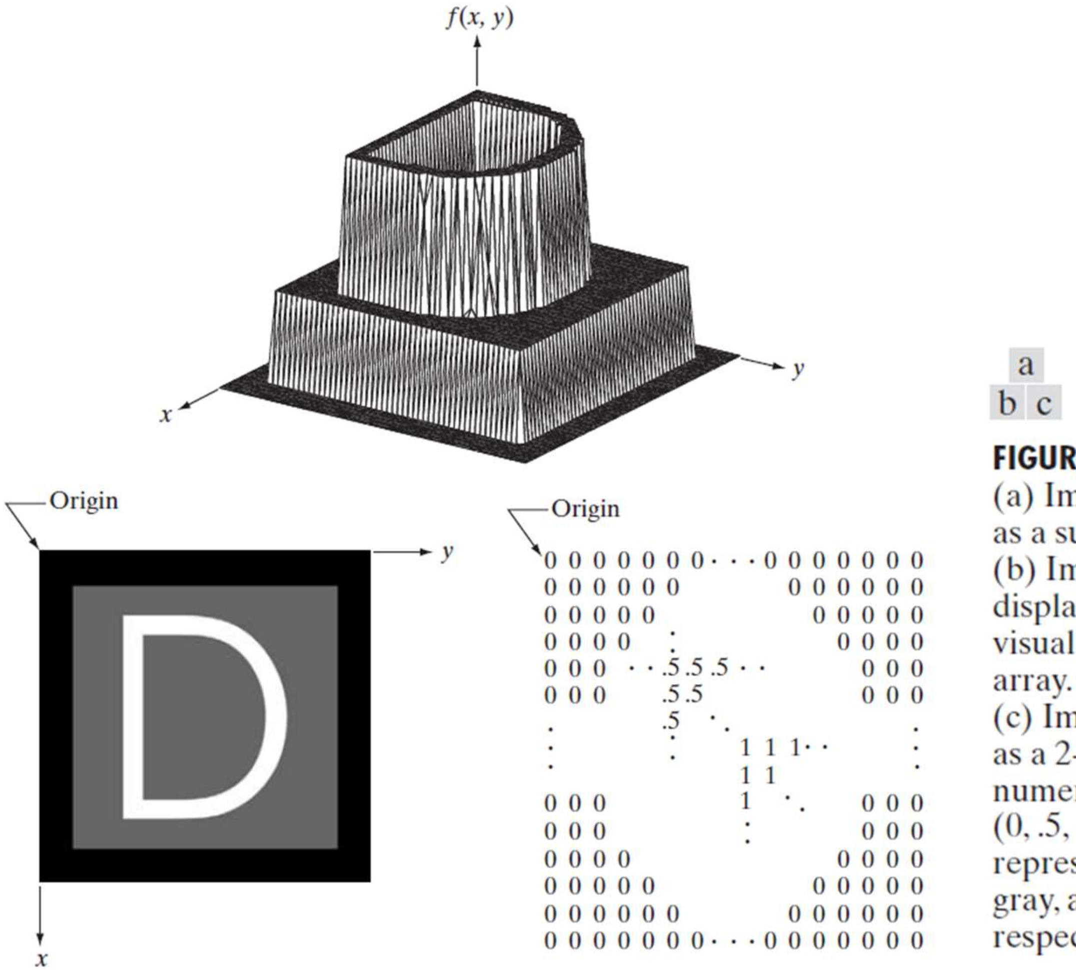


FIGURE 2.18
(a) Image plotted as a surface.
(b) Image displayed as a visual intensity array.
(c) Image shown as a 2-D numerical array (0, .5, and 1 represent black, gray, and white, respectively).

4.1 Representing Digital Images

• In equation form, we write the representation of an M x N numerical array as:

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

• Or,

$$\mathbf{A} = \begin{bmatrix} a_{0,0} & a_{0,1} & \cdots & a_{0,N-1} \\ a_{1,0} & a_{1,1} & \cdots & a_{1,N-1} \\ \vdots & \vdots & & \vdots \\ a_{M-1,0} & a_{M-1,1} & \cdots & a_{M-1,N-1} \end{bmatrix}$$

• Each element of this matrix is called an **image element**, **picture element** or **pixel**.

4.1 Representing Digital Images

• This digitization process requires that decisions be made regarding the values for M, N, and for the number, L, of discrete intensity levels.

• There are no restrictions placed on M and N, other than they have to be positive integers. However, due to storage and quantizing hardware considerations, the number of intensity levels typically is an integer power of 2:

$$L = 2^k$$

• We assume that the discrete levels are equally spaced and that they are integers in the interval [0, L-1].

• The number, *b*, of bits required to store a digitized image is

$$b = M \times N \times k.$$

4.1 Representing Digital Images

- Example (M = 8 row, N = 8 column , k = 8 *bits*):

	0	1	2	3	4	5	6	7
0	8	9	43	98	15	13	10	14
1	9	11	94	50	15	13	10	14
2	9	11	161	29	14	40	96	17
3	9	14	170	21	15	148	173	19
4	9	23	153	17	23	168	166	19
5	9	53	110	15	43	175	168	20
6	10	101	67	15	74	174	167	20
7	11	141	41	14	117	173	164	20

N (columns), M (rows)

$$b = 8 \times 8 \times 8 = 512 \text{ bits}$$

4.1 Representing Digital Images

- MATLAB functions:
  - clear all
  - close all
  - x = zeros(M,N)
  - x = ones(M,N)
  - y = uint8(x)
  - imshow(y)
  - figure

4.2 Spatial and Intensity Resolution

• Intuitively, **spatial resolution** is a measure of the smallest discernible detail in an image.

• Can be stated in a number of ways, with **line pairs per unit distance**, and **dots (pixels) per unit distance.**

• **Image size** by itself does not tell the complete story. To say that an image has, say, a resolution 1024 x 1024 pixels is not a meaningful statement without stating the spatial dimensions encompassed by the image.

4.2 Spatial and Intensity Resolution

- Intuitively, **spatial resolution** is a measure of the smallest discernible detail in an image.

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.

4.2 Spatial and Intensity Resolution

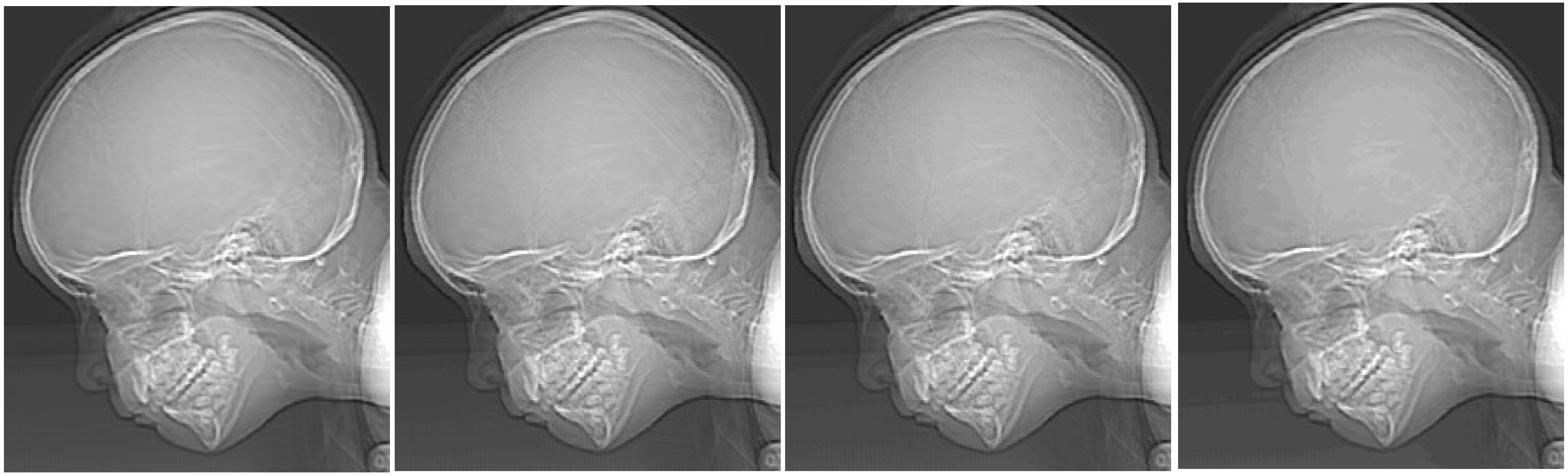
a b c
d e f

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.

4.2 Spatial and Intensity Resolution

- **Intensity resolution** refers to the smallest discernible change in intensity level.

- For example, an image whose intensity is quantized into 256 levels has 8 bits of intensity resolution.

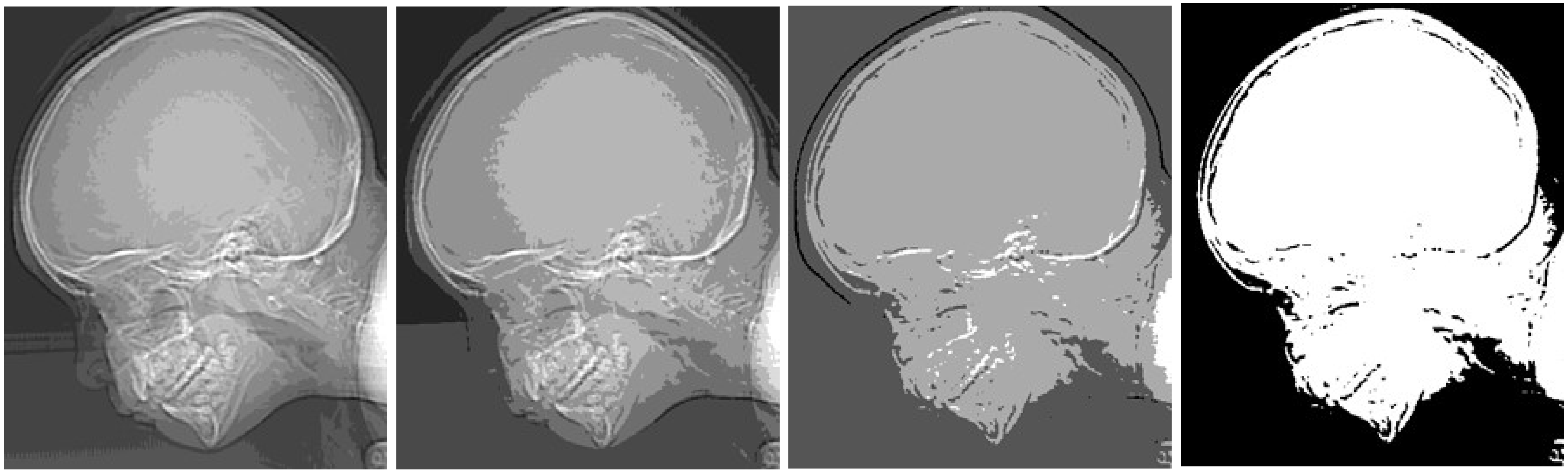


L = 256 128 64 32

4.2 Spatial and Intensity Resolution

- **Intensity resolution** refers to the smallest discernible change in intensity level.

- For example, an image whose intensity is quantized into 256 levels has 8 bits of intensity resolution.



L = 16 8 4 2

4.2 Spatial and Intensity Resolution

- Effects on image quality produced by varying N and k simultaneously (Huang [1965]).

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.)

4.2 Spatial and Intensity Resolution

- Effects on image quality produced by varying N (N=M) and k simultaneously (Huang [1965]).

- Points lying on an **isopreference curve** correspond to images of equal subjective quality

For a fixed value of **N**, the perceived quality for this type of image is nearly independent of the number of intensity levels used.

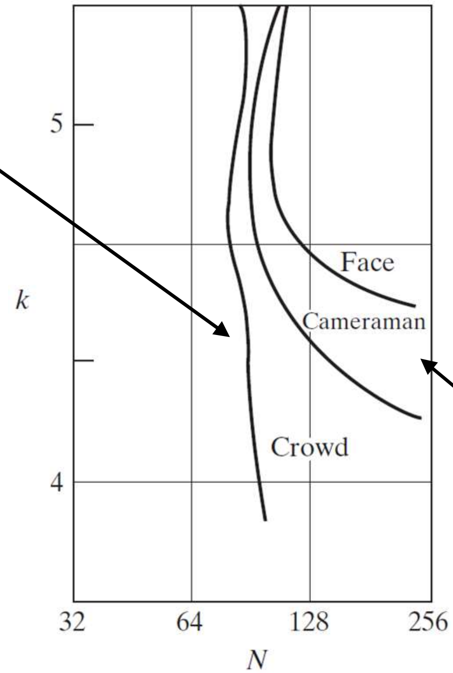


FIGURE 2.23 Typical isopreference curves for the three types of images in Fig. 2.22.

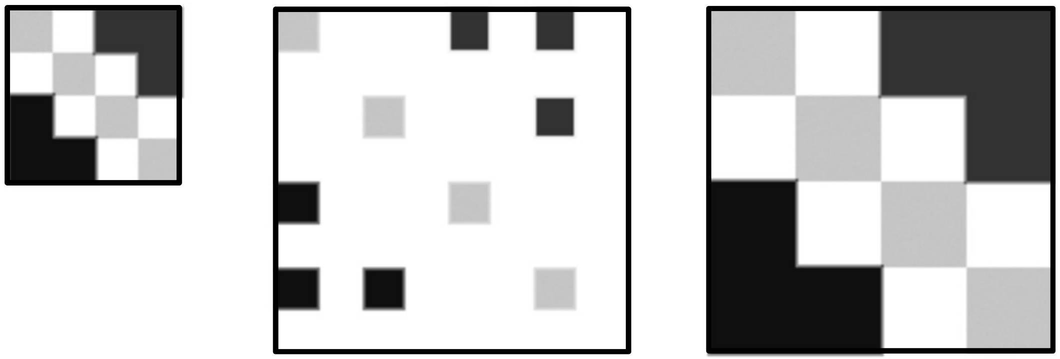
Quality remained the same in some intervals in which the number of samples was increased, but the number of intensity levels actually decreased.

4.3 Image Interpolation

• **Interpolation** (resampling/resizing: shrinking and zooming) is the process of using known data to estimate values at unknown locations.

• Suppose that an image of size 4 x 4 pixels has to be enlarged 2 times to 8 x 8 pixels.

- **Nearest neighbor interpolation** assigns to each new location the intensity of its nearest neighbor in the original image.



4.3 Image Interpolation

- **Bilinear interpolation**:

$$f(x,y) = \sum_{i=0}^{1}\sum_{j=0}^{1} a_{ij}x^i y^j = a_{00} + a_{10}x + a_{01}y + a_{11}xy$$

a_{00}	a_{10}
a_{01}	a_{11}

$$a_{00} = f(0,0)$$
$$a_{10} = f(1,0) - f(0,0)$$
$$a_{01} = f(0,1) - f(0,0)$$
$$a_{11} = f(1,1) + f(0,0) - (f(1,0) + f(0,1))$$

- **Bicubic interpolation**:

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

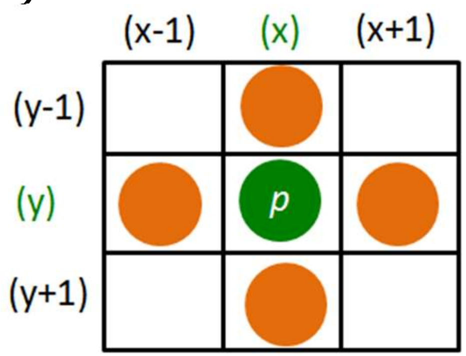
5. Some Basic Relationships between Pixels

- **Neighbors of a Pixel:**

➢ A pixel p at coordinates (x,y) has four horizontal and vertical neighbors whose coordinates are given by

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

➢ This set of pixels, called the **4-neighbors** of p, is denoted by $N_4(p)$.



4-neighbors N4(p)

Image: http://cse19-iiith.vlabs.ac.in/theory.php?exp=diff

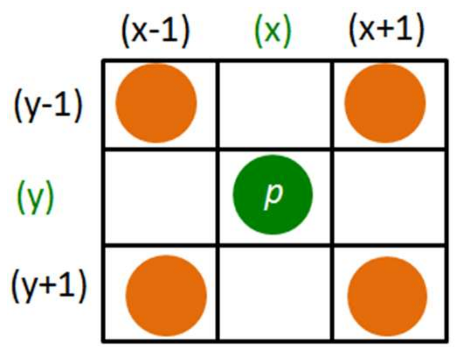
5. Some Basic Relationships between Pixels

- **Neighbors of a Pixel:**

  - The four **diagonal neighbors** of p have coordinates

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

and are denoted by $N_D(p)$.



Diagonal neighbors Nd(p)

Image: http://cse19-iiith.vlabs.ac.in/theory.php?exp=diff

5. Some Basic Relationships between Pixels

- **Neighbors of a Pixel:**

  - Diagonal neighbors together with the 4-neighbors are called the **8-neighbors** of p, denoted by

$$N_8(p) = N_4(p) \cup N_D(p)$$

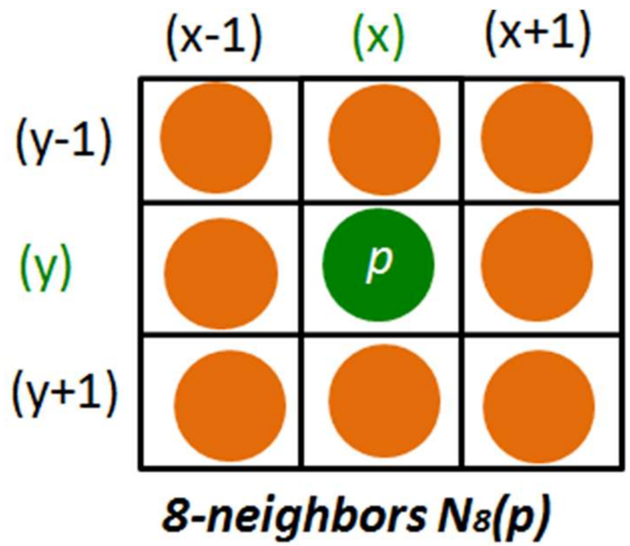


Image: http://cse19-iiith.vlabs.ac.in/theory.php?exp=diff

5. Some Basic Relationships between Pixels

- **Adjacency, Connectivity, Regions, and Boundaries**

➢ Two pixels p and q with values from V (set of intensity values used to define adjacency) are **4-adjacent** if q is in the set $N_4(p)$. If V = {1}:

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	**1**	**1**	0	0	0	0
0	0	0	**1**	0	0	0	0	0
0	0	0	0	**1**	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

5. Some Basic Relationships between Pixels

- **Adjacency, Connectivity, Regions, and Boundaries**

➢ Two pixels p and q with values from V (set of intensity values used to define adjacency) are **4-adjacent** if q is in the set $N_4(p)$. If V = {1}:

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	**1**	**1**	0	0	0	0
0	0	0	**1**	0	0	0	0	0
0	0	0	0	**1**	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

5. Some Basic Relationships between Pixels

- **Adjacency, Connectivity, Regions, and Boundaries**

  - Two pixels p and q with values from V (set of intensity values used to define adjacency) are **4-adjacent** if q is in the set $N_4(p)$. If V = {1}:

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	**1**	**1**	0	0	0	0
0	0	0	**1**	0	0	0	0	0
0	0	0	0	**1**	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

5. Some Basic Relationships between Pixels

- **Adjacency, Connectivity, Regions, and Boundaries**

  ➢ Two pixels p and q with values from V (set of intensity values used to define adjacency) are **4-adjacent** if q is in the set $N_4(p)$. If V = {1}:

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	**1**	**1**	0	0	0	0
0	0	0	**1**	0	0	0	0	0
0	0	0	0	**1**	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

5. Some Basic Relationships between Pixels

- **Adjacency, Connectivity, Regions, and Boundaries**

➢ Two pixels p and q with values from V (set of intensity values used to define adjacency) are **4-adjacent** if q is in the set $N_4(p)$. If V = {1}:

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	**1**	**1**	0	0	0	0
0	0	0	**1**	0	0	0	0	0
0	0	0	0	**1**	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

5. Some Basic Relationships between Pixels

- **Adjacency, Connectivity, Regions, and Boundaries**

➢ Two pixels p and q with values from V (set of intensity values used to define adjacency) are **4-adjacent** if q is in the set $N_4(p)$. If V = {1}:

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	**1**	**1**	0	0	0	0
0	0	0	**1**	0	0	0	0	0
0	0	0	0	**1**	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

5. Some Basic Relationships between Pixels

- **Adjacency, Connectivity, Regions, and Boundaries**

➢ Two pixels p and q with values from V (set of intensity values used to define adjacency) are **4-adjacent** if q is in the set $N_4(p)$. If V = {1}:

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	**1**	**1**	0	0	0	0
0	0	0	**1**	0	0	0	0	0
0	0	0	0	**1**	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

5. Some Basic Relationships between Pixels

- **Adjacency, Connectivity, Regions, and Boundaries**

➢ Two pixels p and q with values from V (set of intensity values used to define adjacency) are **4-adjacent** if q is in the set $N_4(p)$. If V = {1}:

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	**1**	**1**	0	0	0	0
0	0	0	**1**	0	0	0	0	0
0	0	0	0	**1**	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

5. Some Basic Relationships between Pixels

- **Adjacency, Connectivity, Regions, and Boundaries**

  - Two pixels p and q with values from V (set of intensity values used to define adjacency) are **4-adjacent** if q is in the set $N_4(p)$. If V = {1}:

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	**1**	**1**	0	0	0	0
0	0	0	**1**	0	0	0	0	0
0	0	0	0	**1**	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

5. Some Basic Relationships between Pixels

- **Adjacency, Connectivity, Regions, and Boundaries**

  - Two pixels p and q with values from V (set of intensity values used to define adjacency) are **4-adjacent** if q is in the set $N_4(p)$. If V = {1}:

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	**1**	**1**	0	0	0	0
0	0	0	**1**	0	0	0	0	0
0	0	0	0	**1**	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

5. Some Basic Relationships between Pixels

- **Adjacency, Connectivity, Regions, and Boundaries**

➢ Two pixels p and q with values from V (set of intensity values used to define adjacency) are **4-adjacent** if q is in the set $N_4(p)$. If V = {1}:

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	**1**	**1**	0	0	0	0
0	0	0	**1**	0	0	0	0	0
0	0	0	0	**1**	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

5. Some Basic Relationships between Pixels

- **Adjacency, Connectivity, Regions, and Boundaries**

  ➢ Two pixels p and q with values from V (set of intensity values used to define adjacency) are **4-adjacent** if q is in the set $N_4(p)$. If V = {1}:

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	**1**	**1**	0	0	0	0
0	0	0	**1**	0	0	0	0	0
0	0	0	0	**1**	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

5. Some Basic Relationships between Pixels

- **Adjacency, Connectivity, Regions, and Boundaries**

➢ Two pixels p and q with values from V (set of intensity values used to define adjacency) are **4-adjacent** if q is in the set $N_4(p)$. If V = {1}:

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	**1**	**1**	0	0	0	0
0	0	0	**1**	0	0	0	0	0
0	0	0	0	**1**	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

5. Some Basic Relationships between Pixels

- **Adjacency, Connectivity, Regions, and Boundaries**

  - Two pixels p and q with values from V (set of intensity values used to define adjacency) are **8-adjacent** if q is in the set $N_8(p)$. If V = {1}:

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	**1**	**1**	0	0	0	0
0	0	0	**1**	0	0	0	0	0
0	0	0	0	**1**	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

5. Some Basic Relationships between Pixels

• **Adjacency, Connectivity, Regions, and Boundaries**

➢ Two pixels p and q with values from V (set of intensity values used to define adjacency) are **8-adjacent** if q is in the set $N_8(p)$. If V = {1}:

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	**1**	**1**	0	0	0	0
0	0	0	**1**	0	0	0	0	0
0	0	0	0	**1**	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

5. Some Basic Relationships between Pixels

- **Adjacency, Connectivity, Regions, and Boundaries**

  ➢ Two pixels p and q with values from V (set of intensity values used to define adjacency) are **8-adjacent** if q is in the set $N_8(p)$. If V = {1}:

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	**1**	**1**	0	0	0	0
0	0	0	**1**	0	0	0	0	0
0	0	0	0	**1**	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

5. Some Basic Relationships between Pixels

- **Adjacency, Connectivity, Regions, and Boundaries**

➢ Two pixels p and q with values from V (set of intensity values used to define adjacency) are **8-adjacent** if q is in the set $N_8(p)$. If V = {1}:

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	**1**	**1**	0	0	0	0
0	0	0	**1**	0	0	0	0	0
0	0	0	0	**1**	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

5. Some Basic Relationships between Pixels

- **Adjacency, Connectivity, Regions, and Boundaries**

  - Two pixels p and q with values from V (set of intensity values used to define adjacency) are **8-adjacent** if q is in the set $N_8(p)$. If V = {1}:

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	**1**	**1**	0	0	0	0
0	0	0	**1**	0	0	0	0	0
0	0	0	0	**1**	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

5. Some Basic Relationships between Pixels

- **Adjacency, Connectivity, Regions, and Boundaries**

   ➢ Two pixels p and q with values from V (set of intensity values used to define adjacency) are **8-adjacent** if q is in the set $N_8(p)$. If V = {1}:

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	**1**	**1**	0	0	0	0
0	0	0	**1**	0	0	0	0	0
0	0	0	0	**1**	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

5. Some Basic Relationships between Pixels

- **Adjacency, Connectivity, Regions, and Boundaries**

  - Two pixels p and q with values from V (set of intensity values used to define adjacency) are **8-adjacent** if q is in the set $N_8(p)$. If V = {1}:

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	**1**	**1**	0	0	0	0
0	0	0	**1**	0	0	0	0	0
0	0	0	0	**1**	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

5. Some Basic Relationships between Pixels

- **Adjacency, Connectivity, Regions, and Boundaries**

  - Two pixels p and q with values from V (set of intensity values used to define adjacency) are **8-adjacent** if q is in the set $N_8(p)$. If V = {1}:

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	**1**	**1**	0	0	0	0
0	0	0	**1**	0	0	0	0	0
0	0	0	0	**1**	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

5. Some Basic Relationships between Pixels

- **Adjacency, Connectivity, Regions, and Boundaries**

  ➢ Two pixels p and q with values from V (set of intensity values used to define adjacency) are **8-adjacent** if q is in the set $N_8(p)$. If V = {1}:

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	**1**	**1**	0	0	0	0
0	0	0	**1**	0	0	0	0	0
0	0	0	0	**1**	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

5. Some Basic Relationships between Pixels

- **Adjacency, Connectivity, Regions, and Boundaries**

➢ Two pixels p and q with values from V (set of intensity values used to define adjacency) are **8-adjacent** if q is in the set $N_8(p)$. If V = {1}:

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	**1**	**1**	0	0	0	0
0	0	0	**1**	0	0	0	0	0
0	0	0	0	**1**	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

5. Some Basic Relationships between Pixels

- **Adjacency, Connectivity, Regions, and Boundaries**

  ➢ Two pixels p and q with values from V (set of intensity values used to define adjacency) are **8-adjacent** if q is in the set $N_8(p)$. If V = {1}:

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	**1**	**1**	0	0	0	0
0	0	0	**1**	0	0	0	0	0
0	0	0	0	**1**	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

5. Some Basic Relationships between Pixels

- **Adjacency, Connectivity, Regions, and Boundaries**

  ➢ Two pixels p and q with values from V (set of intensity values used to define adjacency) are **8-adjacent** if q is in the set $N_8(p)$. If V = {1}:

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	**1**	**1**	0	0	0	0
0	0	0	**1**	0	0	0	0	0
0	0	0	0	**1**	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

5. Some Basic Relationships between Pixels

- **Adjacency, Connectivity, Regions, and Boundaries**

  ➢ Two pixels p and q with values from V (set of intensity values used to define adjacency) are **8-adjacent** if q is in the set $N_8(p)$. If V = {1}:

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	**1**	**1**	0	0	0	0
0	0	0	**1**	0	0	0	0	0
0	0	0	0	**1**	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

5. Some Basic Relationships between Pixels

- **Adjacency, Connectivity, Regions, and Boundaries**

  - Two pixels p and q with values from V (set of intensity values used to define adjacency) are **8-adjacent** if q is in the set $N_8(p)$. If V = {1}:

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	**1**	**1**	0	0	0	0
0	0	0	**1**	0	0	0	0	0
0	0	0	0	**1**	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

5. Some Basic Relationships between Pixels

- **Adjacency, Connectivity, Regions, and Boundaries**

  - Two pixels p and q with values from V (set of intensity values used to define adjacency) are **8-adjacent** if q is in the set $N_8(p)$. If V = {1}:

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	**1**	**1**	0	0	0	0
0	0	0	**1**	0	0	0	0	0
0	0	0	0	**1**	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

5. Some Basic Relationships between Pixels

- **Adjacency, Connectivity, Regions, and Boundaries**

  ➢ Two pixels p and q with values from V (set of intensity values used to define adjacency) are **8-adjacent** if q is in the set $N_8(p)$. If V = {1}:

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	**1**	**1**	0	0	0	0
0	0	0	**1**	0	0	0	0	0
0	0	0	0	**1**	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

5. Some Basic Relationships between Pixels

- **Adjacency, Connectivity, Regions, and Boundaries**

  - Two pixels *p* and *q* with values from V (set of intensity values used to define adjacency) are **m-adjacent** if:

    a. q is in $N_4(p)$; or

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

- Is introduced to eliminate the ambiguities that often arise when 8-adjacency is used.

5. Some Basic Relationships between Pixels

- **Adjacency, Connectivity, Regions, and Boundaries**
    - ➢ **m-adjacent:**

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	**1**	**1**	0	0	0	0
0	0	0	**1**	0	0	0	0	0
0	0	0	0	**1**	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

5. Some Basic Relationships between Pixels

- **Adjacency, Connectivity, Regions, and Boundaries**

  - **m-adjacent:**

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	**1**	**1**	0	0	0	0
0	0	0	**1**	0	0	0	0	0
0	0	0	0	**1**	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

5. Some Basic Relationships between Pixels

- **Adjacency, Connectivity, Regions, and Boundaries**

  - **m-adjacent:**

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	**1**	**1**	0	0	0	0
0	0	0	**1**	0	0	0	0	0
0	0	0	0	**1**	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

5. Some Basic Relationships between Pixels

- **Adjacency, Connectivity, Regions, and Boundaries**

  - **m-adjacent:**

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	**1**	**1**	0	0	0	0
0	0	0	**1**	0	0	0	0	0
0	0	0	0	**1**	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

5. Some Basic Relationships between Pixels

- **Adjacency, Connectivity, Regions, and Boundaries**

  - **m-adjacent:**

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	**1**	**1**	0	0	0	0
0	0	0	**1**	0	0	0	0	0
0	0	0	0	**1**	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

5. Some Basic Relationships between Pixels

- **Adjacency, Connectivity, Regions, and Boundaries**
  - **m-adjacent:**

$N_4(p) \cap N_4(q) \notin V$

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

$N_4(p) \cap N_4(q) \notin V$

5. Some Basic Relationships between Pixels

- **Adjacency, Connectivity, Regions, and Boundaries**

  - **m-adjacent:**

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	**1**	**1**	0	0	0	0
0	0	0	**1**	0	0	0	0	0
0	0	0	0	**1**	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

5. Some Basic Relationships between Pixels

- **Adjacency, Connectivity, Regions, and Boundaries**

  - **m-adjacent:**

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	**1**	**1**	0	0	0	0
0	0	0	**1**	0	0	0	0	0
0	0	0	0	**1**	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

5. Some Basic Relationships between Pixels

- **Adjacency, Connectivity, Regions, and Boundaries**

  - **m-adjacent:**

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	**1**	**1**	0	0	0	0
0	0	0	**1**	0	0	0	0	0
0	0	0	0	**1**	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

5. Some Basic Relationships between Pixels

- **Adjacency, Connectivity, Regions, and Boundaries**

  - **m-adjacent:**

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	**1**	**1**	0	0	0	0
0	0	0	**1**	0	0	0	0	0
0	0	0	0	**1**	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

5. Some Basic Relationships between Pixels

- **Adjacency, Connectivity, Regions, and Boundaries**

  - **m-adjacent:**

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	**1**	**1**	0	0	0	0
0	0	0	**1**	0	0	0	0	0
0	0	0	0	**1**	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

5. Some Basic Relationships between Pixels

- **Adjacency, Connectivity, Regions, and Boundaries**

  - **m-adjacent:**

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

5. Some Basic Relationships between Pixels

- **Adjacency, Connectivity, Regions, and Boundaries**

  - **m-adjacent:**

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	**1**	**1**	0	0	0	0
0	0	0	**1**	0	0	0	0	0
0	0	0	0	**1**	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

5. Some Basic Relationships between Pixels

- **Adjacency, Connectivity, Regions, and Boundaries**

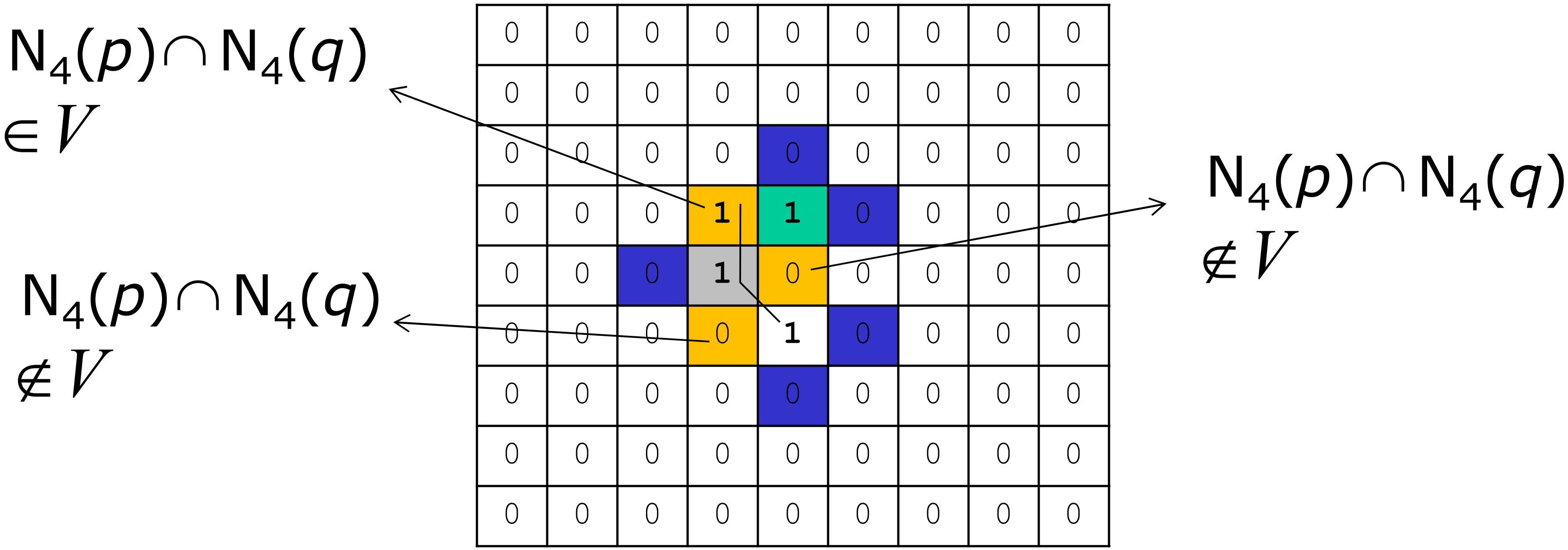
  - **m-adjacent:**

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

5. Some Basic Relationships between Pixels

- **Adjacency, Connectivity, Regions, and Boundaries**

  - **m-adjacent:**



5. Some Basic Relationships between Pixels

- **Adjacency, Connectivity, Regions, and Boundaries**

    - **m-adjacent:**

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	**1**	**1**	0	0	0	0
0	0	0	**1**	0	0	0	0	0
0	0	0	0	**1**	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

5. Some Basic Relationships between Pixels

- **Adjacency, Connectivity, Regions, and Boundaries**

  - **m-adjacent:**

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	**1**	**1**	0	0	0	0
0	0	0	**1**	0	0	0	0	0
0	0	0	0	**1**	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

5. Some Basic Relationships between Pixels

- **Adjacency, Connectivity, Regions, and Boundaries**

  - **m-adjacent:**

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	**1**	**1**	0	0	0	0
0	0	0	**1**	0	0	0	0	0
0	0	0	0	**1**	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

5. Some Basic Relationships between Pixels

- **Adjacency, Connectivity, Regions, and Boundaries**
  - **m-adjacent:**

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	**1**	**1**	0	0	0	0
0	0	0	**1**	0	0	0	0	0
0	0	0	0	**1**	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

5. Some Basic Relationships between Pixels

- **Adjacency, Connectivity, Regions, and Boundaries**

  - **m-adjacent:**

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	**1**	**1**	0	0	0	0
0	0	0	**1**	0	0	0	0	0
0	0	0	0	**1**	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

5. Some Basic Relationships between Pixels

- **Adjacency, Connectivity, Regions, and Boundaries**

  - **m-adjacent:**

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

5. Some Basic Relationships between Pixels

- **Adjacency, Connectivity, Regions, and Boundaries**

  ➢ **m-adjacent:**

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	**1**	**1**	0	0	0	0
0	0	0	**1**	0	0	0	0	0
0	0	0	0	**1**	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

5. Some Basic Relationships between Pixels

- **Adjacency, Connectivity, Regions, and Boundaries**

  - **m-adjacent:**

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	**1**	**1**	0	0	0	0
0	0	0	**1**	0	0	0	0	0
0	0	0	0	**1**	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

5. Some Basic Relationships between Pixels

- **Adjacency, Connectivity, Regions, and Boundaries**

  - **m-adjacent:**

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

5. Some Basic Relationships between Pixels

- **Adjacency, Connectivity, Regions, and Boundaries**

  - **m-adjacent:**

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

5. Some Basic Relationships between Pixels

- **Adjacency, Connectivity, Regions, and Boundaries**

  - **m-adjacent:**

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	**1**	**1**	0	0	0	0
0	0	0	**1**	0	0	0	0	0
0	0	0	0	**1**	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0

5. Some Basic Relationships between Pixels

- **Adjacency, Connectivity, Regions, and Boundaries**

  ➢ **m-adjacent:**

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

5. Some Basic Relationships between Pixels

- **Adjacency, Connectivity, Regions, and Boundaries**

  - **m-adjacent:**

$\mathrm{N}_4(p) \cap \mathrm{N}_4(q) \in V$

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

$\mathrm{N}_4(p) \cap \mathrm{N}_4(q) \notin V$

5. Some Basic Relationships between Pixels

- **Adjacency, Connectivity, Regions, and Boundaries**

  - **m-adjacent:**

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

5. Some Basic Relationships between Pixels

- **Adjacency, Connectivity, Regions, and Boundaries**

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

$$(x,y) = (x_0,y_0), (x_1,y_1), (x_2,y_2), \ldots, (x_n,y_n) = (s,t)$$

  - Pixels (x_i,y_i) and (x_{i-1},y_{i-1}) are adjacent for $1 \leq i \leq n$.

- In this case, n is the length of the path.

- We can define 4-, 8-, or m-paths depending on the type of adjacency specified.

5. Some Basic Relationships between Pixels

- **Adjacency, Connectivity, Regions, and Boundaries**

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

(x_{12},y_{12})

(x_0,y_0)

5. Some Basic Relationships between Pixels

- **Adjacency, Connectivity, Regions, and Boundaries**

  ➢ If $(x_0,y_0) = (x_n,y_n)$ the path is a **closed** path.

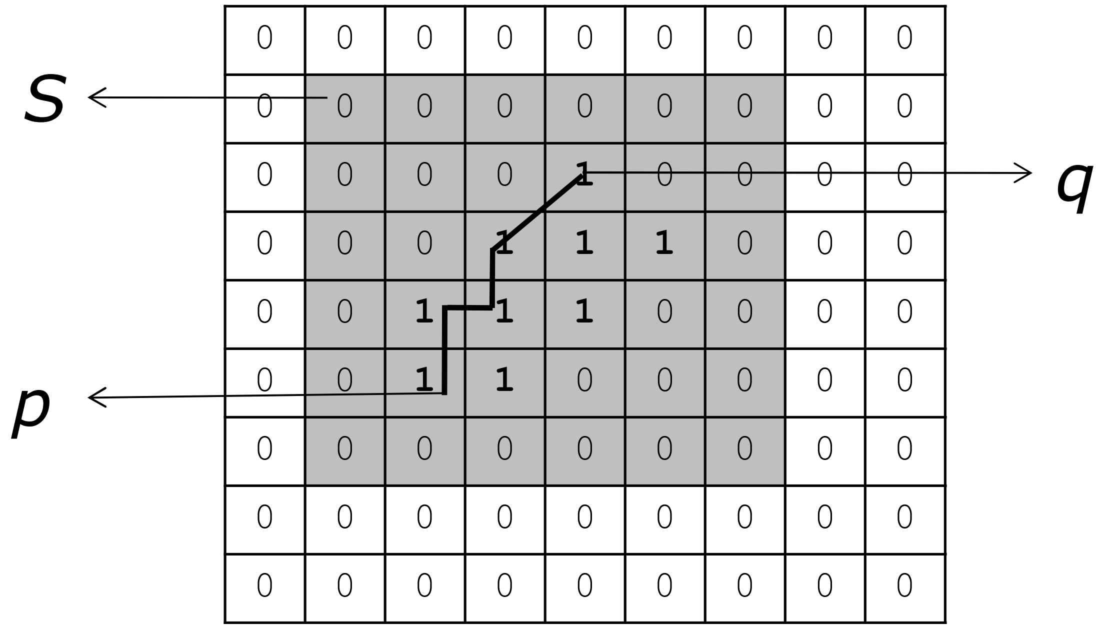
$$(x_0,y_0) = (x_{10},y_{10})$$

0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0
0	1	1	1	0	0	0	0	0
0	1	0	1	0	0	0	0	0
0	1	0	0	**1**	0	0	0	0
0	**1**	0	0	**1**	0	0	0	0
0	0	**1**	**1**	0	0	0	0	0
0	0	0	0	0	0	0	0	0

5. Some Basic Relationships between Pixels

- **Adjacency, Connectivity, Regions, and Boundaries**

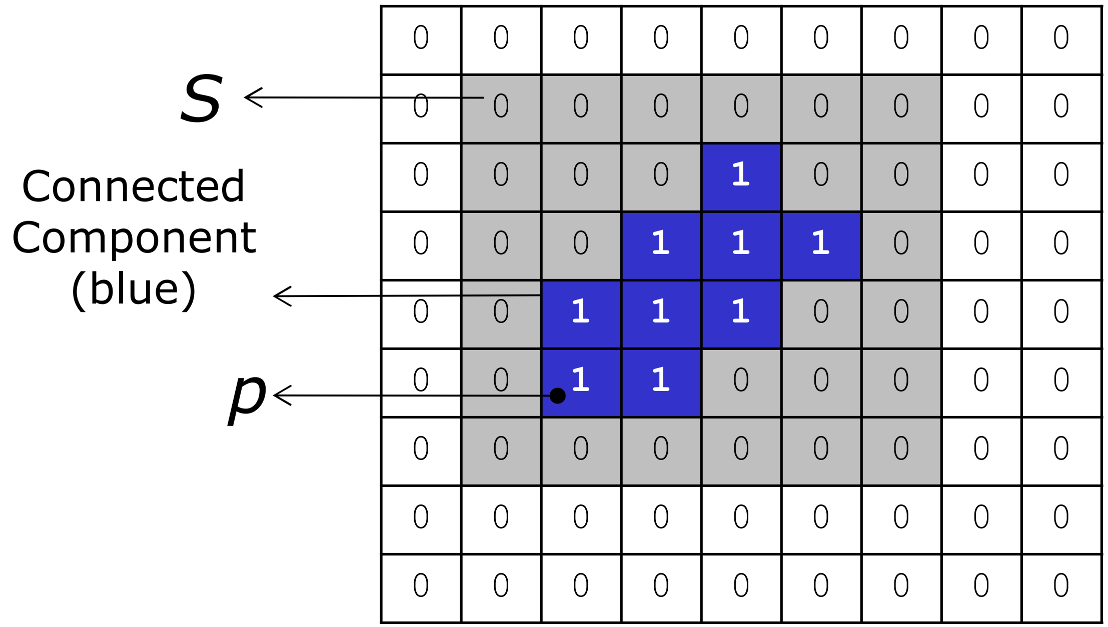
➢ Let S represent a subset of pixels in an image. Two pixels *p* and *q* are said to be **connected** in S if there exists a path between them consisting entirely of pixels in S.



5. Some Basic Relationships between Pixels

- **Adjacency, Connectivity, Regions, and Boundaries**

  - For any pixel *p* in S, the set of pixels that are connected to it in S is called a **connected component** of S.



5. Some Basic Relationships between Pixels

• **Adjacency, Connectivity, Regions, and Boundaries**

➢ If it only has one connected component, then S is called a **connected set** or **a region.**

S

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

Connected set or region

5. Some Basic Relationships between Pixels

- **Adjacency, Connectivity, Regions, and Boundaries**

➢ Two regions, R_i and R_j are said to be adjacent if their union forms a connected set. For our definition to make sense, the type of adjacency used must be specified.

R_1 R_2

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

5. Some Basic Relationships between Pixels

- **Adjacency, Connectivity, Regions, and Boundaries**

  - In our example, R_1 and R_2 are 8-adjacent.

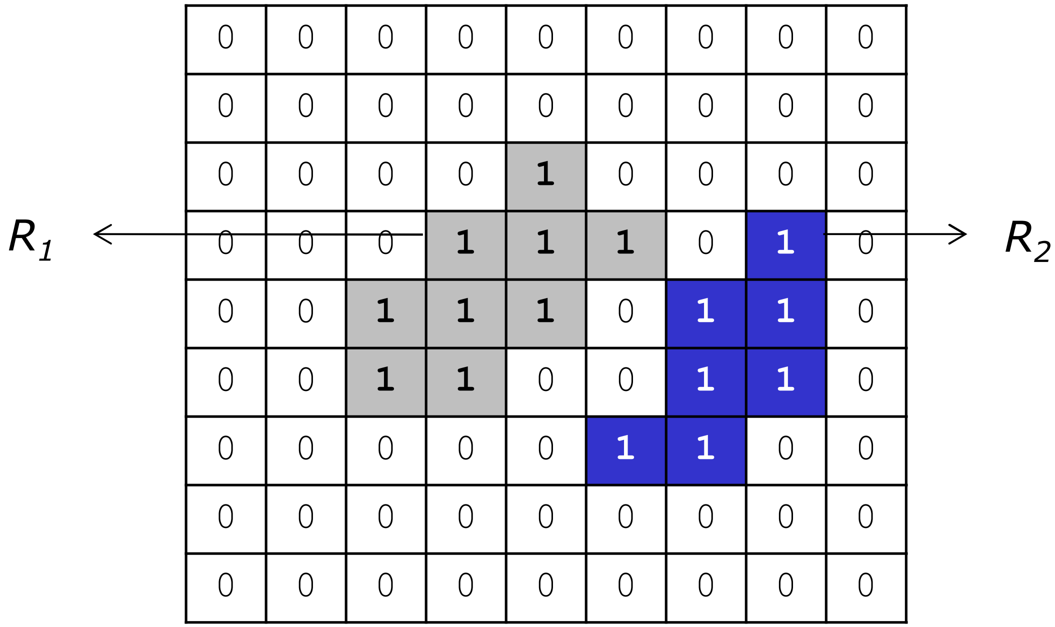
R_1 R_2

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

5. Some Basic Relationships between Pixels

- **Adjacency, Connectivity, Regions, and Boundaries**

  ➢ Regions that are not adjacent are said to be **disjoint**. In our example, R_1 and R_2 are disjoint if 4-adjacency is considered.

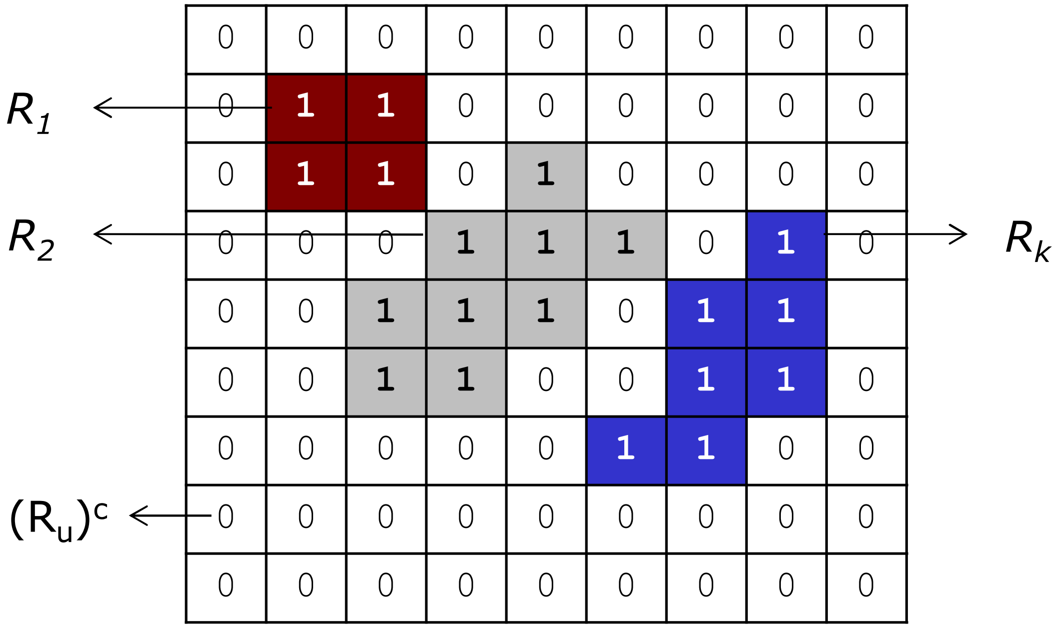


5. Some Basic Relationships between Pixels

- **Adjacency, Connectivity, Regions, and Boundaries**

  ➢ Suppose that an image contains k disjoint regions.

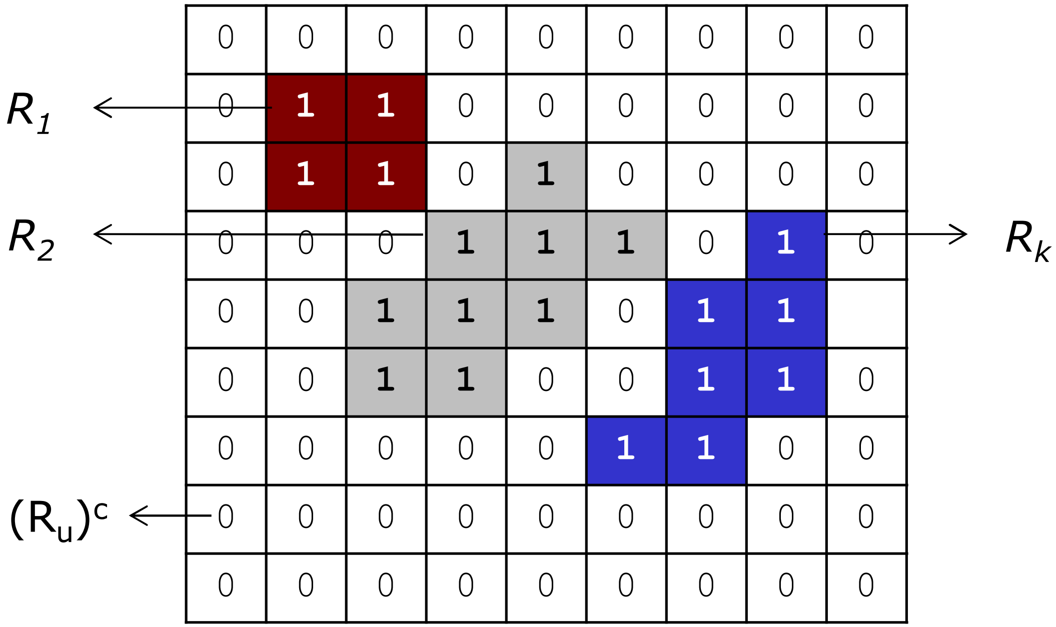
  ➢ Let R_u denote the union of all the k regions, and let denote $(R_u)^c$ its complement.



5. Some Basic Relationships between Pixels

- **Adjacency, Connectivity, Regions, and Boundaries**

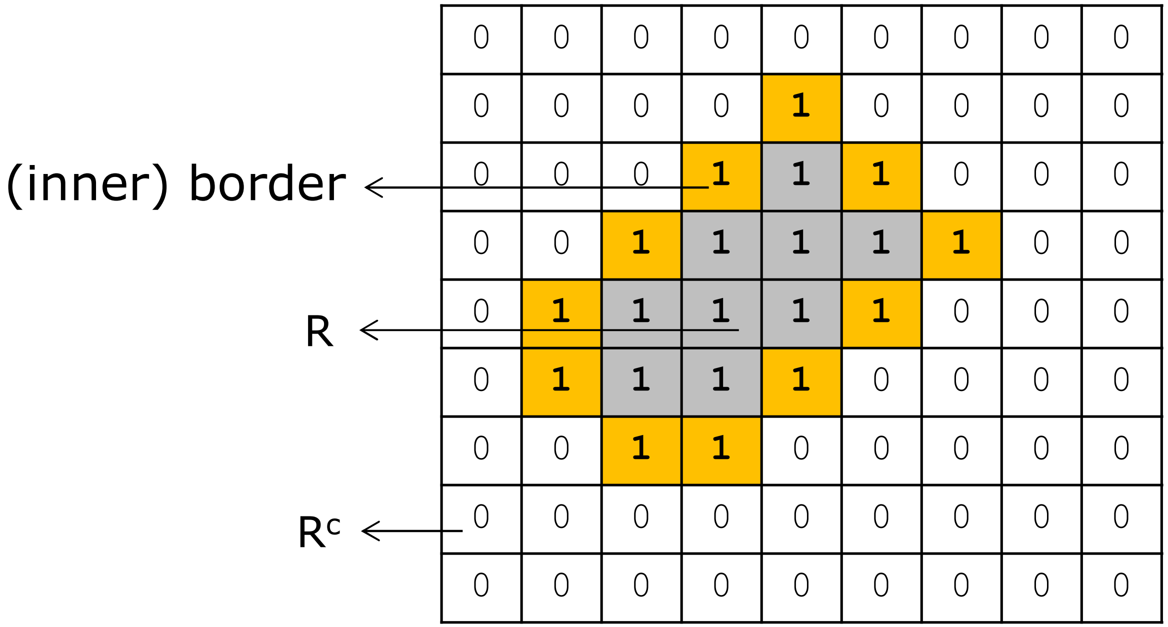
  - We call R_u the **foreground** and $(R_u)^c$ the **background**.



5. Some Basic Relationships between Pixels

- **Adjacency, Connectivity, Regions, and Boundaries**

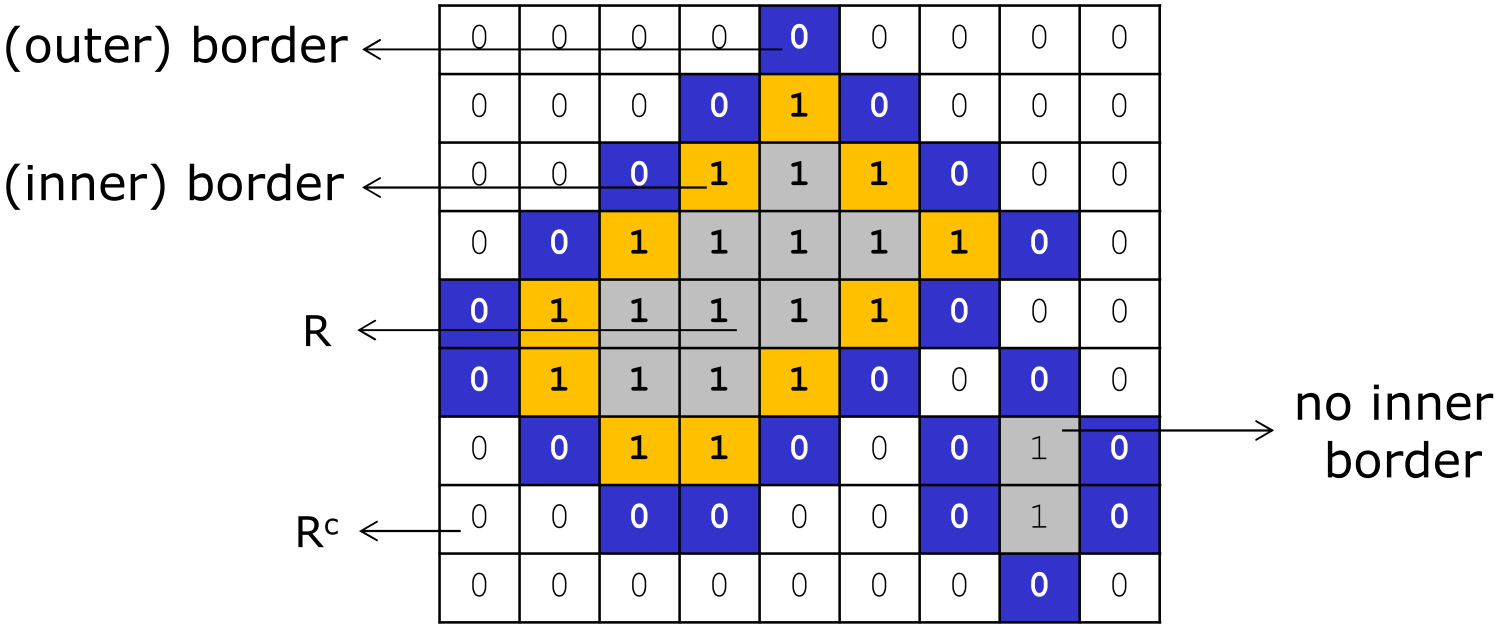
  ➢ The **boundary** (also called the border or contour) of a region R is the set of points that are adjacent to points in the complement of R (inner border).



5. Some Basic Relationships between Pixels

- **Adjacency, Connectivity, Regions, and Boundaries**

  ➢ Border-following algorithms usually are formulated to follow the **outer border.**



5. Some Basic Relationships between Pixels

- **Adjacency, Connectivity, Regions, and Boundaries**

➢ The concept of an **edge** is found frequently in discussions dealing with **regions** and **boundaries**.

➢ There is a key difference between these concepts, however.

✓ The **boundary** of a finite region forms a **closed path**.

✓ **Edges** are formed from pixels with **derivative values** that exceed a preset threshold (intensity discontinuities).

5. Some Basic Relationships between Pixels

• Distance Measures

➢ 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) $D(p, q) \geq 0 \quad (D(p, q) = 0 \quad \text{iff} \quad p = q),$

(b) $D(p, q) = D(q, p)$, and

(c) $D(p, z) \leq D(p, q) + D(q, z).$

5. Some Basic Relationships between Pixels

- **Distance Measures**

  - Euclidean distance

$$D_e(p,q) = [(x-s)^2 + (y-t)^2]^{\frac{1}{2}}$$

	0	1	2	3	4	5	6	7	8
0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	1	0	0	0	0
2	0	0	0	1	1	1	0	0	0
3	0	0	1	1	1	1	1	0	0
4	0	1	1	1	1	1	0	0	0
5	0	1	1	1	1	0	0	0	0
6	0	0	1	1	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0

$(2,3)$

$$D_e(p,q) = [(5-2)^2 + (1-3)^2]^{\frac{1}{2}} \cong 3{,}6$$

$(5,1)$

5. Some Basic Relationships between Pixels

- **Distance Measures**

  - D_4 distance (called the city-block distance):

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

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

	0	1	2	3	4	5	6	7	8
0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	1	0	0	0	0
2	0	0	0	1	1	1	0	0	0
3	0	0	1	1	1	1	1	0	0
4	0	1	1	1	1	1	0	0	0
5	0	1	1	1	1	0	0	0	0
6	0	0	1	1	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0

(2,3)

$$D_4(p,q) = |5-2| + |1-3| = 5$$

(5,1)

5. Some Basic Relationships between Pixels

- **Distance Measures**

  - D_8 distance (called the chessboard distance):

$$D_8(p,q) = \max(|x-s|,|y-t|)$$

	0	1	2	3	4	5	6	7	8
0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	1	0	0	0	0
2	0	0	0	1	1	1	0	0	0
3	0	0	1	1	1	1	1	0	0
4	0	1	1	1	1	1	0	0	0
5	0	1	1	1	1	0	0	0	0
6	0	0	1	1	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0

$(2,3)$

$$D_8(p,q) = \max(|5-2|,|1-3|) = 3$$

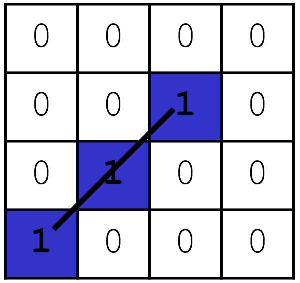
$(5,1)$

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

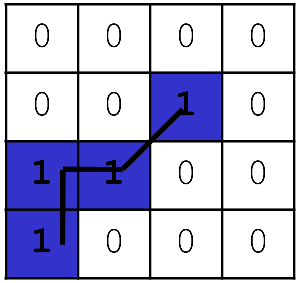
5. Some Basic Relationships between Pixels

- **Distance Measures**

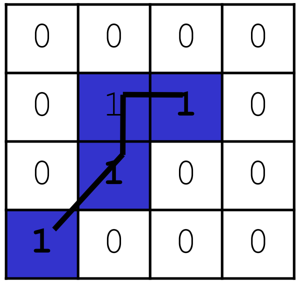
  ➢ D_m distance (shortest m-path between the points two pixels *p* e *q* (V={1}):



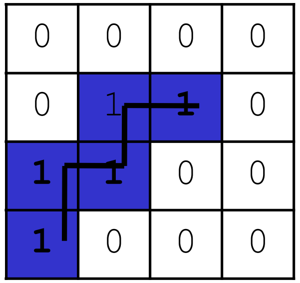
$D_m\ (p,q) = 2$



$D_m\ (p,q) = 3$



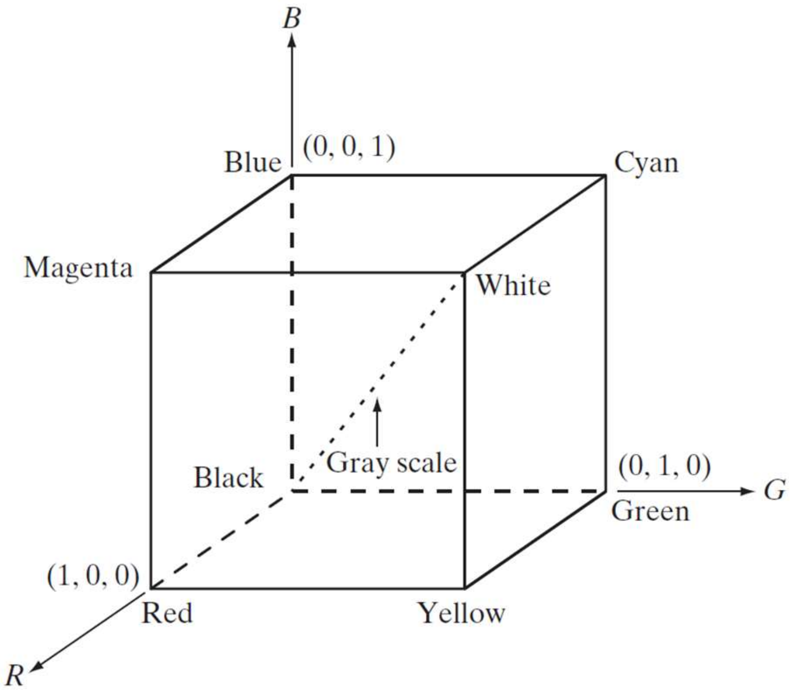
$D_m\ (p,q) = 3$

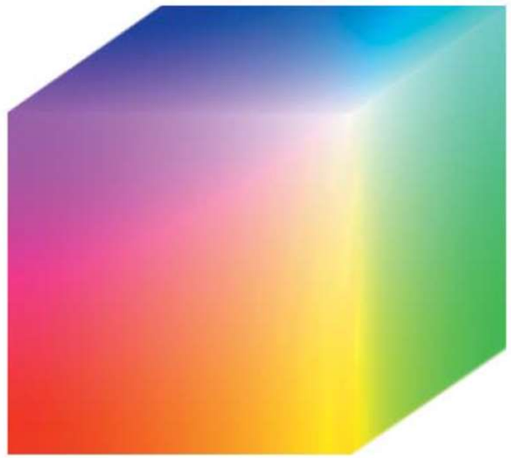


$D_m\ (p,q) = 4$

6. Color Models

- **RGB** (24 *bits* ➔ 256^3 colors ➔ 16.777.216 of colors):





6. Color Models

- **RGB** (24 *bits* ➔ 256^3 colors ➔ 16.777.216 of colors):

6. Color Models

- **YC_bC_r:**

✓ The human visual system is more sensitive to luminance (brightness) than to color.

✓ However, RGB does not take advantage of this fact. The luminance is spread over the components R, G and B, making them equally relevant.

✓ Another useful representation, called YC_bC_r, separates the luminance component Y from the color components, C_b and C_r, also called chrominance.

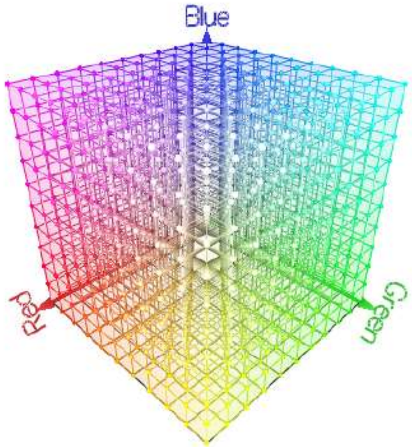
✓ The Y component can be viewed as a grayscale version of the color image.

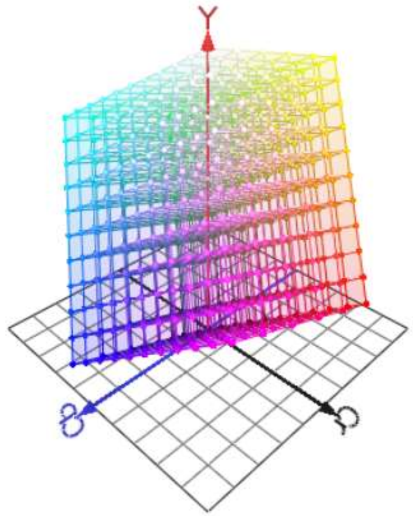
6. Color Models

- **RGB ↔ YCbCr:**

$$Y = 0,299R + 0,587G + 0,114B$$
$$C_b = 0,564(B - Y)$$
$$C_r = 0,713(R - Y)$$

$$R = Y + 1,402C_r$$
$$G = Y - 0,344C_b - 0,714C_r$$
$$B = Y + 1,772C_b$$





6. Color Models

- **RGB → YCbCr:**

Y

Cb

Cr

6. Color Models

- **Grayscale images** (8 *bits* ➔ 256 gray levels):

6. Color Models

- **Binary images** (1 *bit* ➔ 2 levels):

7. Mathematical Tools

- **Array versus Matrix Operations**

  - *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*:

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

7. Mathematical Tools

- **Linear versus Nonlinear operations**

  - The output of a linear operation due to the sum of two inputs is the same as performing the operation on the inputs individually and then summing the results (*additivity*).

  - **T**he output of a linear operation to a constant times an input is the same as the output of the operation due to the original input multiplied by that constant *(homogeneity).*

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

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

7. Mathematical Tools

• Arithmetic Operations

- Arithmetic operations between images are array operations.

- Operations are performed between corresponding pixel pairs in *f* and *g*.

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

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

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

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

7. Mathematical Tools

- **Arithmetic Operations**

  - Example (*image averaging/denoising*):

$$g(x, y) = f(x, y) + \eta(x, y)$$

    - The assumption is that at every pair of coordinates (x, y) the noise is uncorrelated† and has zero average value.

    - If the noise satisfies the constraints just stated, and an image $\overline{g}$ is formed by averaging K different noisy images,

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

7. Mathematical Tools

- **Arithmetic Operations**

  - Example (*image averaging/denoising*):

    - ✓ Then if follows that,

$$E\{\overline{g}(x, y)\} = f(x, y)$$

    - ✓ Variance and standard deviation of $\overline{g}$:

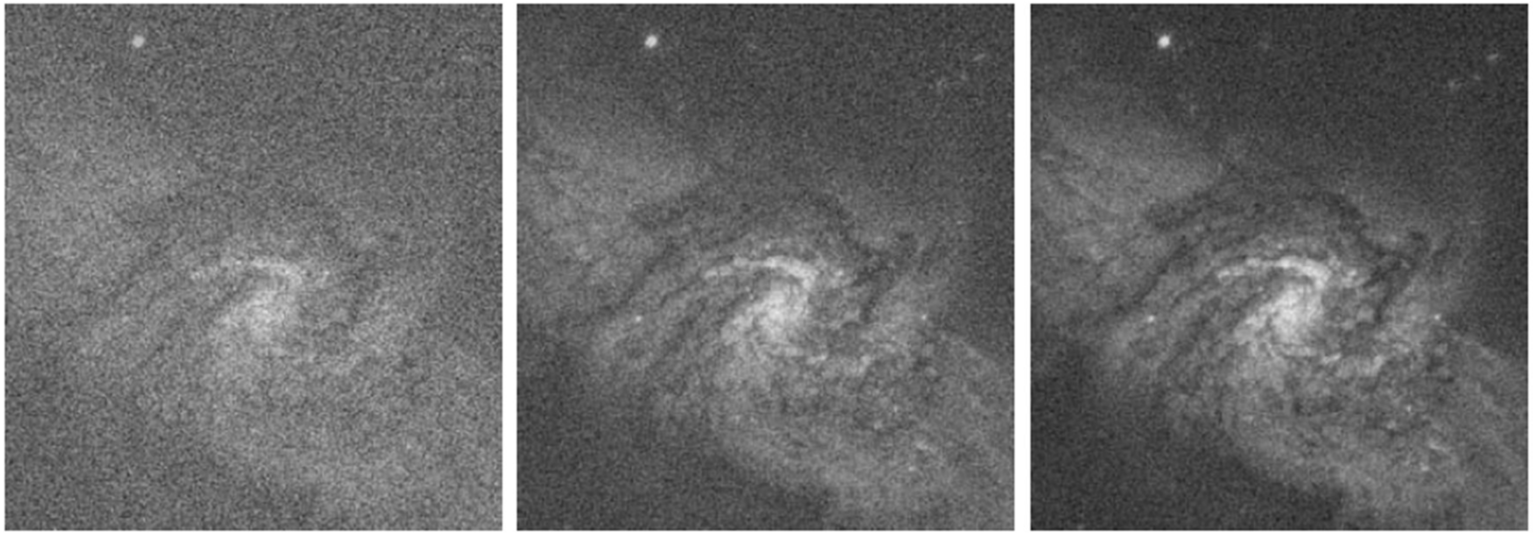
$$\sigma^2_{\overline{g}(x,y)} = \frac{1}{K}\sigma^2_{\eta(x,y)}$$

$$\sigma_{\overline{g}(x,y)} = \frac{1}{\sqrt{K}}\sigma_{\eta(x,y)}$$

7. Mathematical Tools

- **Arithmetic Operations**

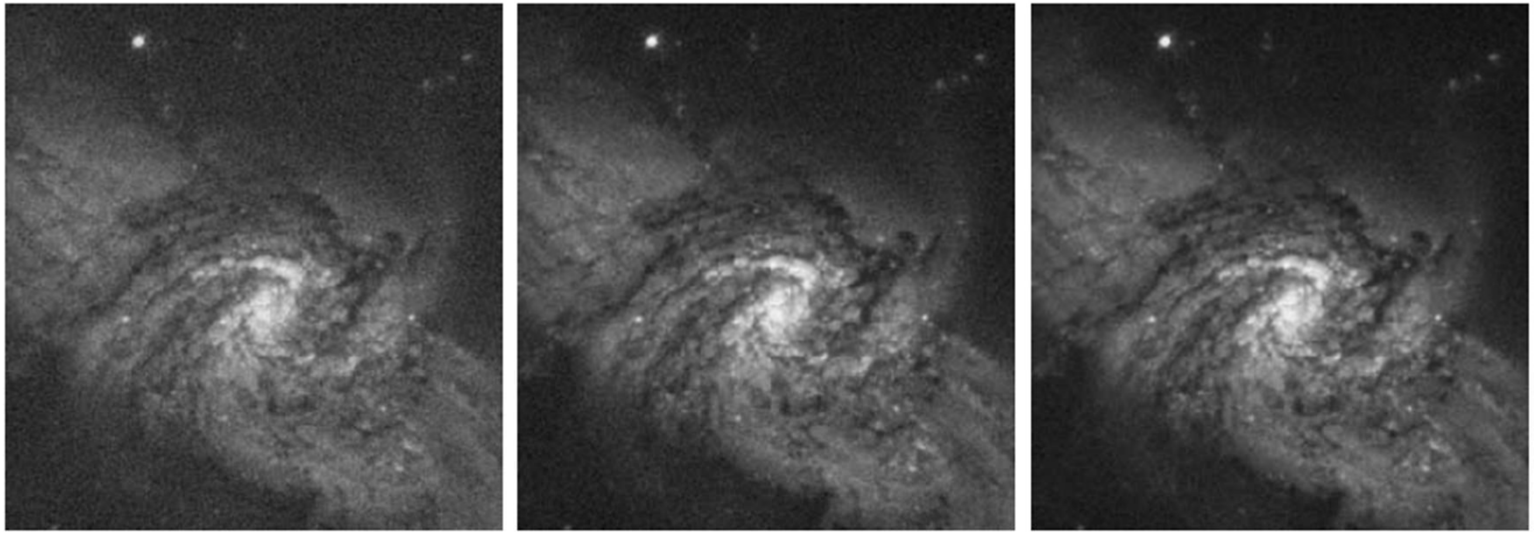
  - Example (*image averaging/denoising*):



7. Mathematical Tools

• Arithmetic Operations

➢ Example (*image averaging/denoising*):



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.)

7. Mathematical Tools

- **Arithmetic Operations**

  - MATLAB: s160Lena.m

7. Mathematical Tools

- **Arithmetic Operations**

  - Example (*image subtraction/mask mode radiography*):

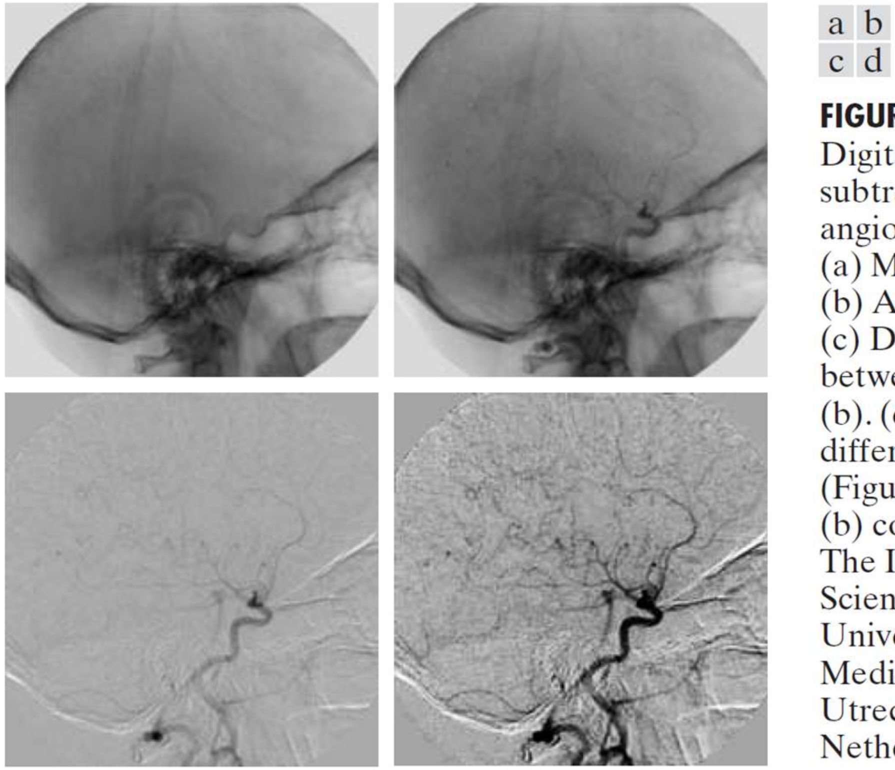
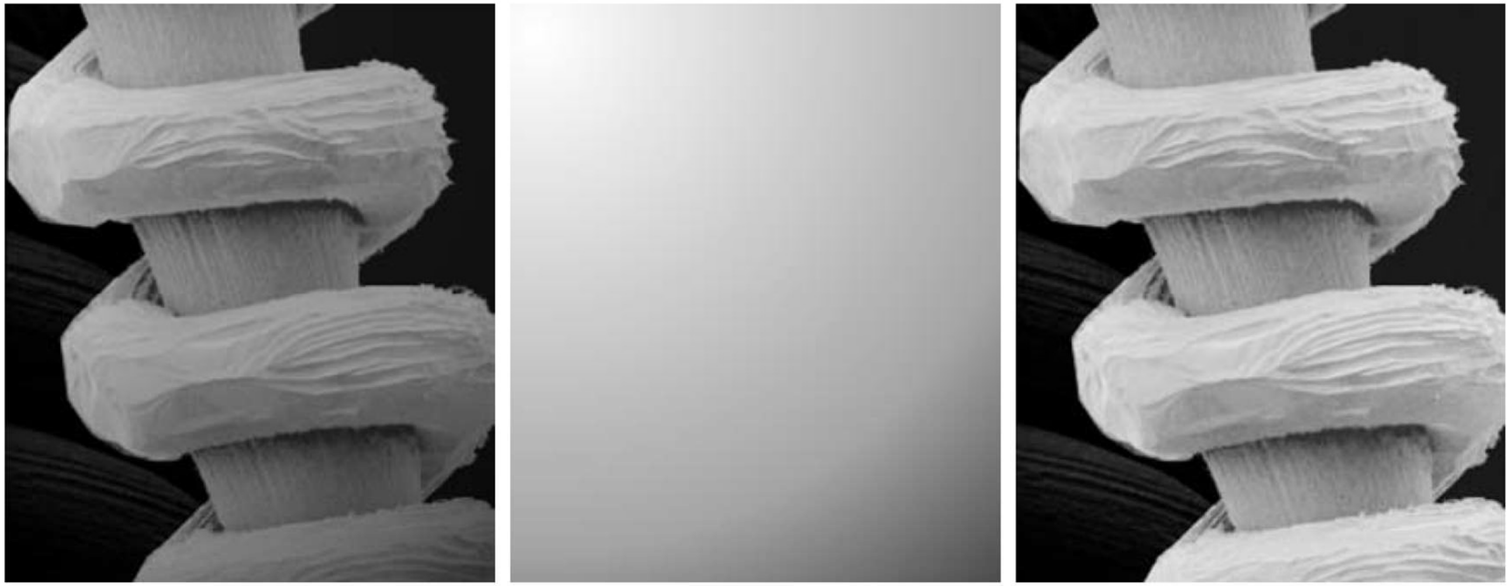


FIGURE 2.28 Digital subtraction angiography. (a) Mask image. (b) A live image. (c) Difference between (a) and (b). (d) Enhanced difference image. (Figures (a) and (b) courtesy of The Image Sciences Institute, University Medical Center, Utrecht, The Netherlands.)

7. Mathematical Tools

• Arithmetic Operations

➢ Example (*image multiplication/shading correction*):



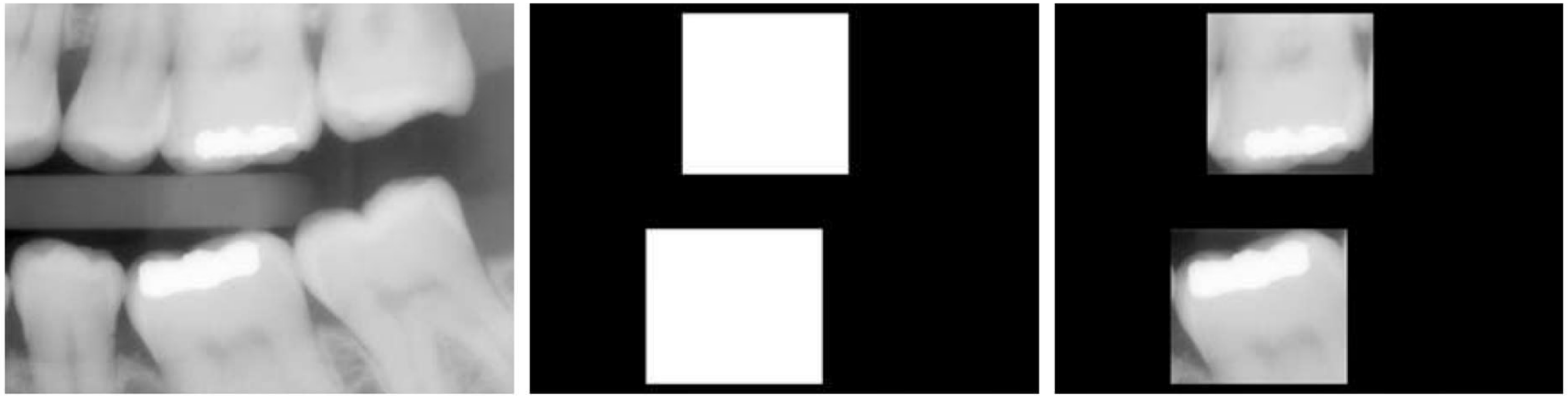
a b c

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 Michael Shaffer, Department of Geological Sciences, University of Oregon, Eugene.)

7. Mathematical Tools

- **Arithmetic Operations**

  - Example (*image multiplication/region of interest*):



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).

7. Mathematical Tools

- **Arithmetic Operations**

  - A few comments about implementing image arithmetic operations.

  - In practice, most images are displayed using 8 bits (or three separate 8-bit channels).

  - Thus, we expect image values to be in the range from 0 to 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 K+1 intensity levels is as follows:

$$f_m = f - \min(f) \qquad f_s = K\left[\, f_m/\max(f_m)\right]$$

7. Mathematical Tools

• Set and Logical Operations

- Let A be a set composed of ordered pairs of real numbers. If $a = (a_1, a_2)$ is an element of A, then we write $a \in A$
- Similarly, if a is not an element of A, we write $a \notin A$
- The set with no elements is called the null or empty set and is denoted by the symbol ∅.
- A set is specified by the contents of two braces: {·}. For example: $C = \{w | w = -d, d \in D\}$
- One way in which sets are used in image processing is to let the elements of sets be the coordinates of pixels.

7. Mathematical Tools

- **Set and Logical 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, denoted as

$$A \subseteq B$$

  - The *union* of two sets A and B is denoted by

$$C = A \cup B$$

  - Similarly, the *intersection* of two sets A and B, denoted by

$$D = A \cap B$$

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

$$A \cap B = \varnothing$$

7. Mathematical Tools

- **Set and Logical 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:

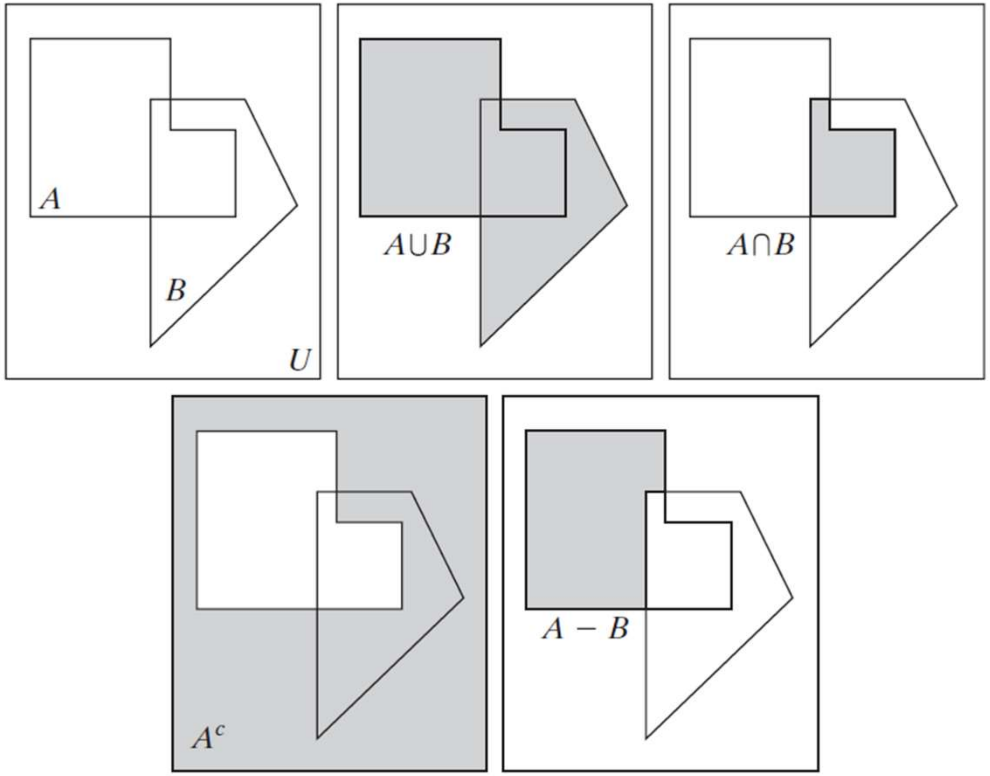
$$A^c = \{w | w \notin A\}$$

  - The *difference* of two sets A and B is defined as

$$A - B = \{w | w \in A, w \notin B\} = A \cap B^c$$

7. Mathematical Tools

- **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 members of the set operation indicated.

7. Mathematical Tools

- **Set and Logical Operations**

  - Example:

    - Let the elements of a gray-scale image be represented by a set *A* whose elements are triplets of the form (x,y,z), where x and y are spatial coordinates and z denotes intensity.

    - We can define the complement of *A* as the set

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

    - Let *A* denote an 8-bit gray-scale image (K=255).

7. Mathematical Tools

- **Set and Logical Operations**

  - Example:

    - Let the elements of a gray-scale image be represented by a set *A* whose elements are triplets of the form (x,y,z), where x and y are spatial coordinates and z denotes intensity.

    - We can define the complement of *A* as the set

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

    - Let *A* denote an 8-bit gray-scale image (K=255).

7. Mathematical Tools

- **Set and Logical Operations**

  - Example:

    - ✓ Suppose that we want to form the negative of *A* using set operations. We simply form the set

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

7. Mathematical Tools

• Set and Logical Operations

➢ Example:

✓ The union of two gray-scale sets A and B may be defined as the set

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

✓ Array formed from the maximum intensity between pairs of spatially corresponding elements.

7. Mathematical Tools

- **Set and Logical Operations**

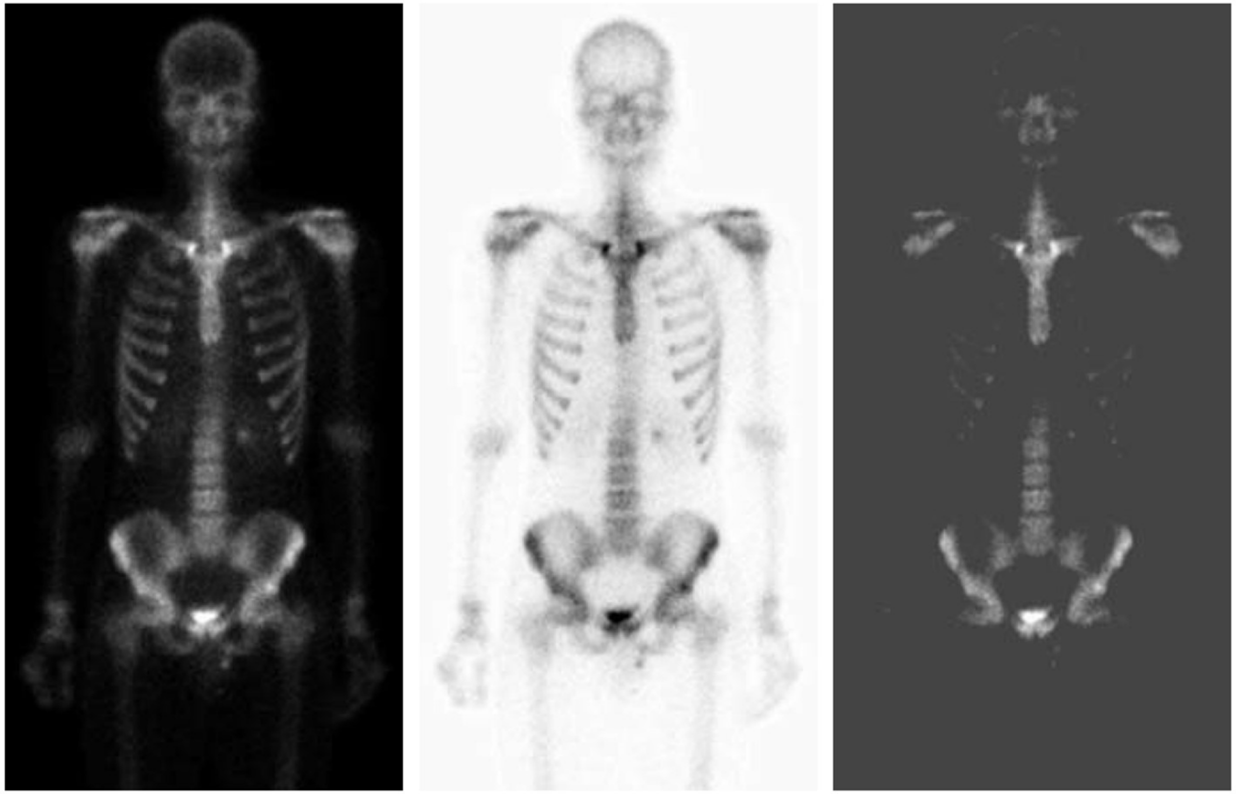
  - Example:

    - ✓ Suppose that A again represents the image in, and let B denotes a rectangular array of the same size as A, but in which all values of *z* are equal to 3 times the mean intensity, *m*, of the elements of *A*.

    - ✓ The result of performing the set union, is show in the next slide.

7. Mathematical Tools

- **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.)

7. Mathematical Tools

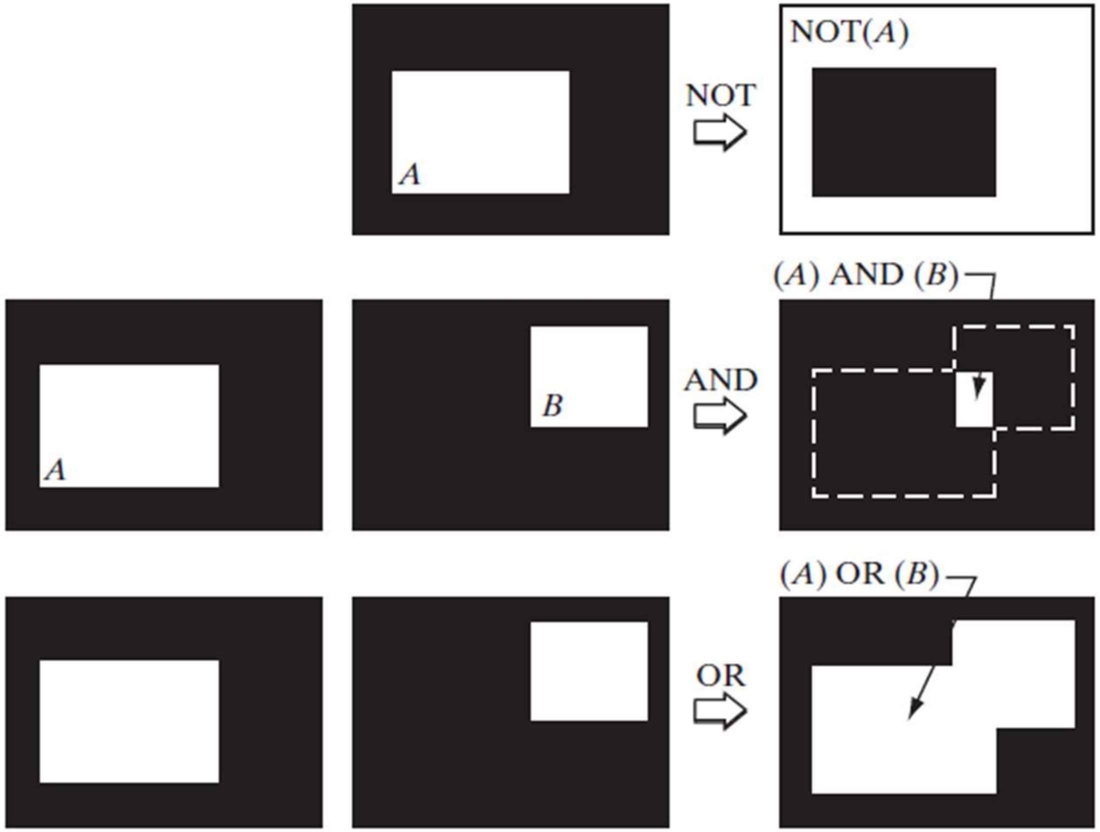
- **Set and Logical Operations**

  - When dealing with binary images, we can think of *foreground* (1-valued) and *background* (0-valued) sets of pixels.

  - In this case, it is common practice to refer to *union*, *intersection*, and *complement* as the OR, AND, and NOT *logical operations*.

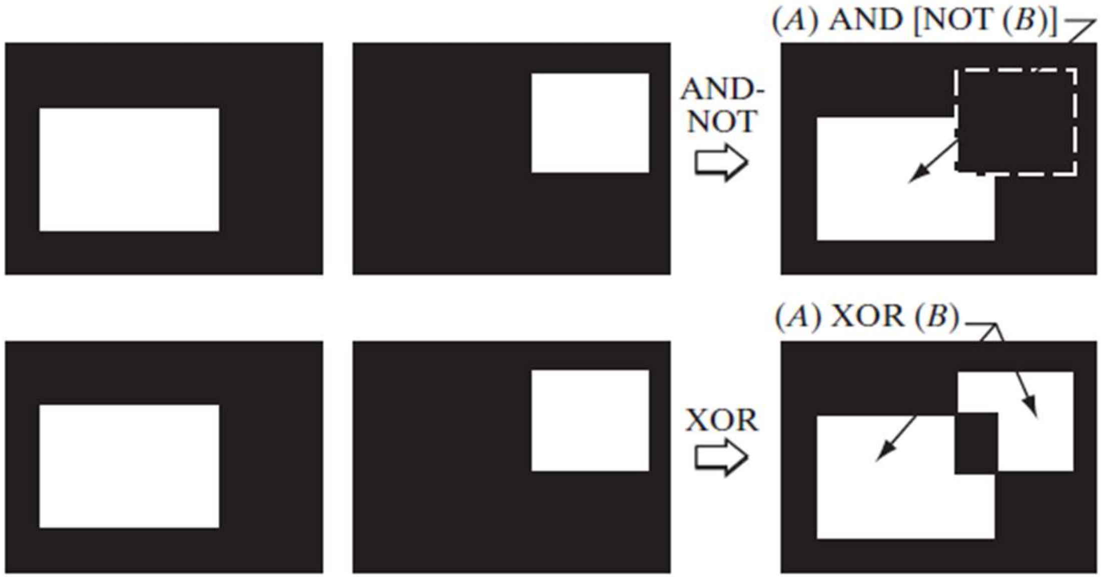
7. Mathematical Tools

- **Set and Logical Operations**



7. Mathematical Tools

- **Set and Logical Operations**



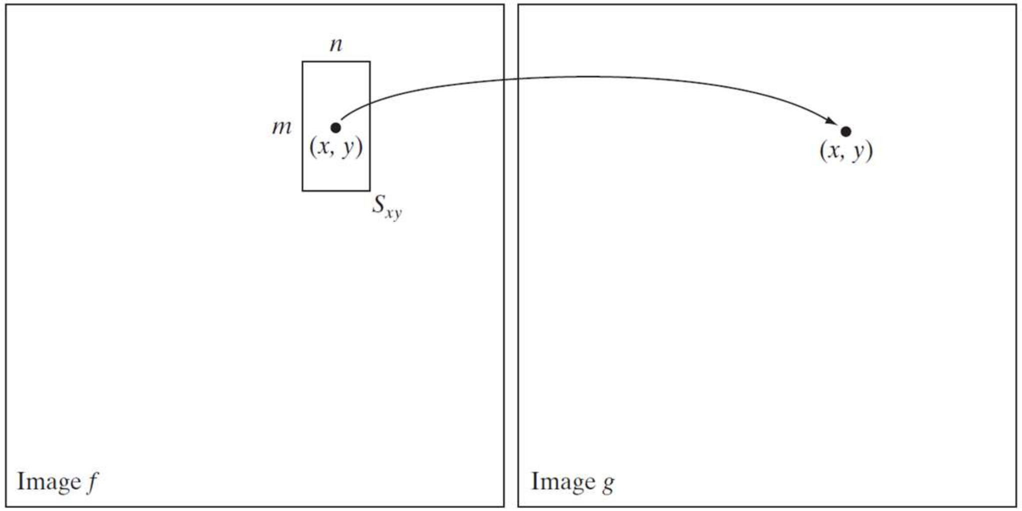
7. Mathematical Tools

- **Spatial Operations**

  - *Single-pixel operations*: alter the values pixels based on their intensity.

$$s = T(z)$$

  - *Neighborhood operations*: alter the values of pixels based on their neighborhood.



7. Mathematical Tools

- **Spatial Operations**

  - *Geometric spatial*: modify the spatial relationship between pixels in an image.

  - The transformation of coordinates may be expressed as

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

  - For example,

$$(x, y) = T\{(v, w)\} = (v/2, w/2)$$

shrinks the original image to half its size in both spatial directions.

7. Mathematical Tools

- **Spatial Operations**

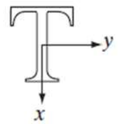
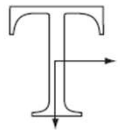
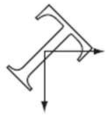
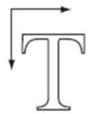
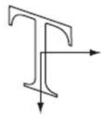
  - One of the most commonly used spatial coordinate transformations is the *affine transform*:

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

  - This transformation can scale, rotate, translate, or shear a set of coordinate points, depending on the value chosen for the elements of matrix T.

7. Mathematical Tools

• Spatial Operations

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)	$\begin{bmatrix} 1 & s_h & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x = v$ $y = s_h v + w$	

7. Mathematical Tools

- **Spatial Operations**

  - *Image registration*: we have available the input and output images, but the specific transformation that produced the output image from the input generally is unknown.

  - The problem, then, is to estimate the transformation function and then use it to register the two images.

  - 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

$$x = c_1 v + c_2 w + c_3 v w + c_4$$

$$y = c_5 v + c_6 w + c_7 v w + c_8$$

7. Mathematical Tools

• Spatial Operations

- If we have four pairs of corresponding tie points in both images, we can write eight equations and use them to solve for the eight unknown coefficients.

- These coefficients constitute the model that transforms the pixels of one image into the locations of the pixels of the other to achieve registration.

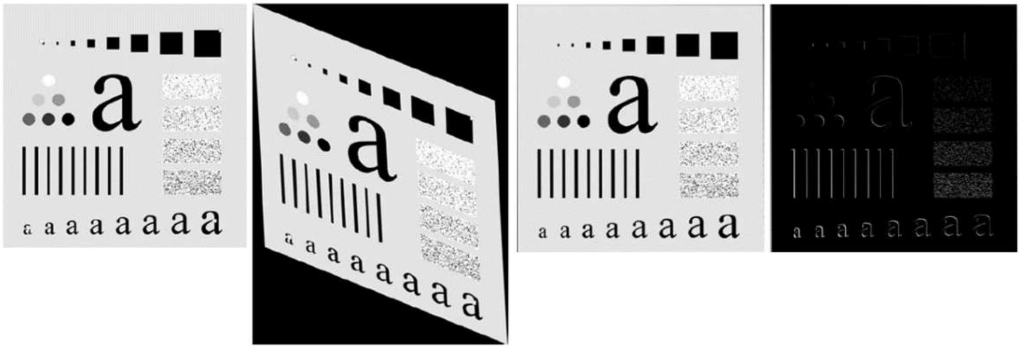


FIGURE 2.37 Image registration. (a) Reference image. (b) Input (geometrically distorted image). Corresponding tie points are shown as small white squares near the corners. (c) Registered image (note the errors in the border). (d) Difference between (a) and (c), showing more registration errors.

7. Mathematical Tools

- **Vector and Matrix Operations**

  - Color images are formed in RGB color space by using red, green, and blue component images.

  - Each pixel has three components, which can be organized in the form of a column vector.

$$\mathbf{z} = \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix}$$

  - Thus, an RGB color image of size MxN can be represented by three component images of this size, or by a total of MN 3-D vectors.

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

7. Mathematical Tools

• Image Transforms

- In some cases, image processing tasks 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.

- A particularly important class of 2-D linear transforms, can be expressed in the general form.

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

where *f* is the input image, *r* is called the forward transformation kernel, and the equation is evaluated for and *u* = 0, 1,..., M–1 and *v* = = 0, 1,..., N-1.

7. Mathematical Tools

• Image Transforms

- T is called the *forward transform* of f.
- Given T we can recover f using the *inverse transform*.

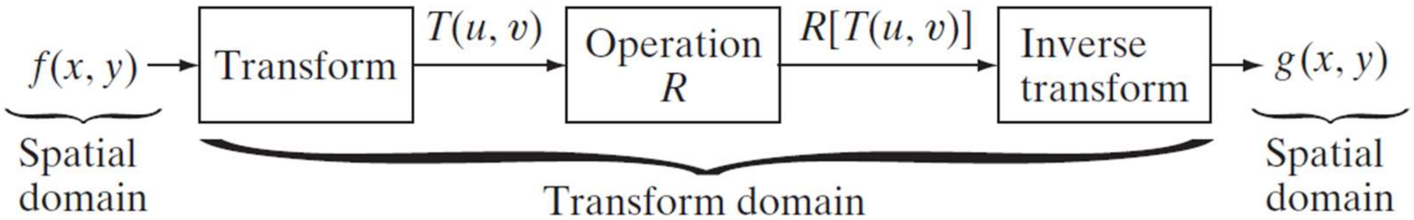
$$f(x, y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} T(u, v)s(x, y, u, v)$$

- The equation is evaluated for and x = 0, 1,…, M−1 and y = = 0, 1,..., N-1.
- And s is called the inverse transformation kernel.

7. Mathematical Tools

• Image Transforms

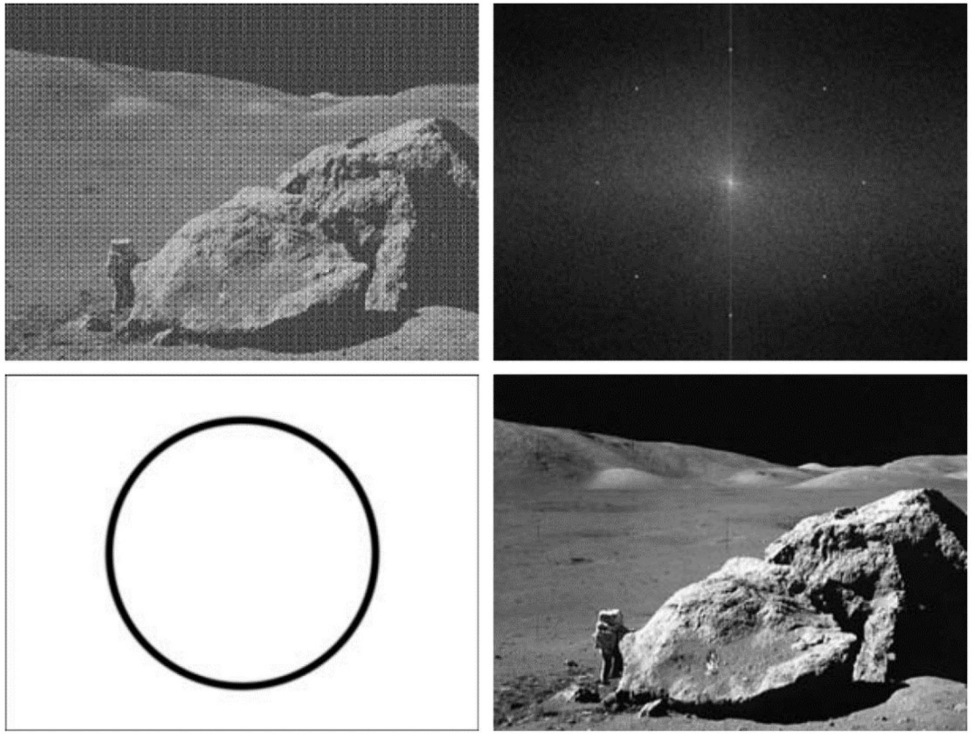
➢ General approach for operating in the linear transform domain.



7. Mathematical Tools

• Image Transforms

- General approach for operating in the linear transform domain.



a b
c d

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.)

7. Mathematical Tools

• Probabilistic Methods

➢ We may treat intensity values as random quantities.

✓ Probability: $p(z_k) = \dfrac{n_k}{MN}$

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

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

➢ We use these statistical quantities to derive intensity enhancement algorithms.

MATLAB Tips and Examples

- **Useful functions**

```
clear all
close all
clc
I = imread('cameraman.tif');
[X, map] = gray2ind(I, 16);
II = ind2gray(X,map);
imshow(II);
imwrite()
imresize()
rgb2ycbcr()
```

Suggested reading: Gonzales & Woods DIP - Chapter 2