Image Analysis

Umberto Castellani
Robotics, vision and control

Image analysis

- **Image analysis** aims at extracting usefull information from images.
- The output depends by the task, e.g.:
  - Object localization,
  - Object recognition,
  - Object reconstruction
- A proper **image analysis pipeline** can be design combining several image processing tecniques

Image analysis pipeline

- Image formation,
- Image filtering and enhancing,
- Feature extraction and segmentation,
- Shape-characterization and measuring

Image formation

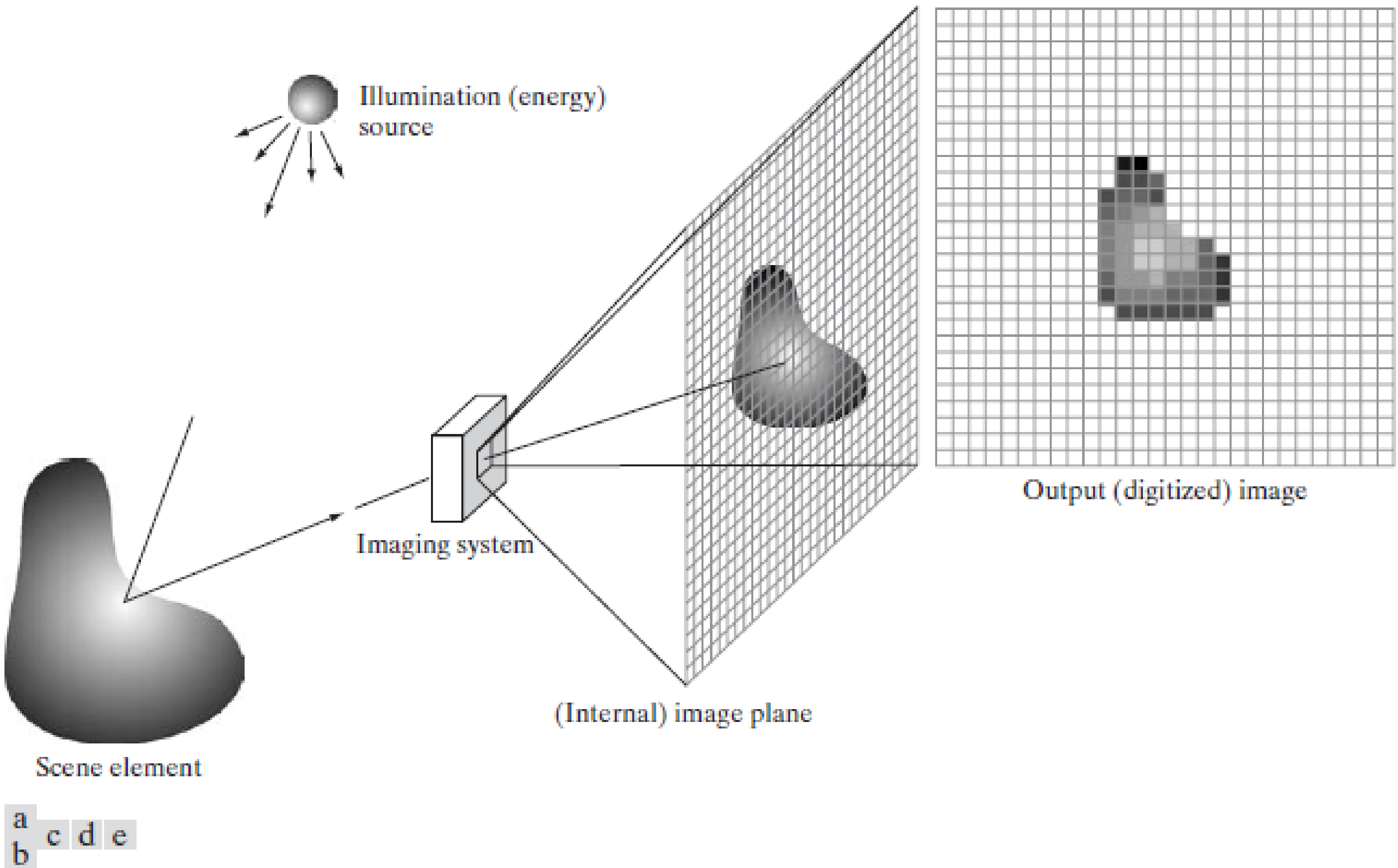


FIGURE 2.15 An example of the digital image acquisition process. (a) Energy ("illumination") source. (b) An element of a scene. (c) Imaging system. (d) Projection of the scene onto the image plane. (e) Digitized image.

Image dimension

Image dimension= number of pixel (horizontalxvertical)

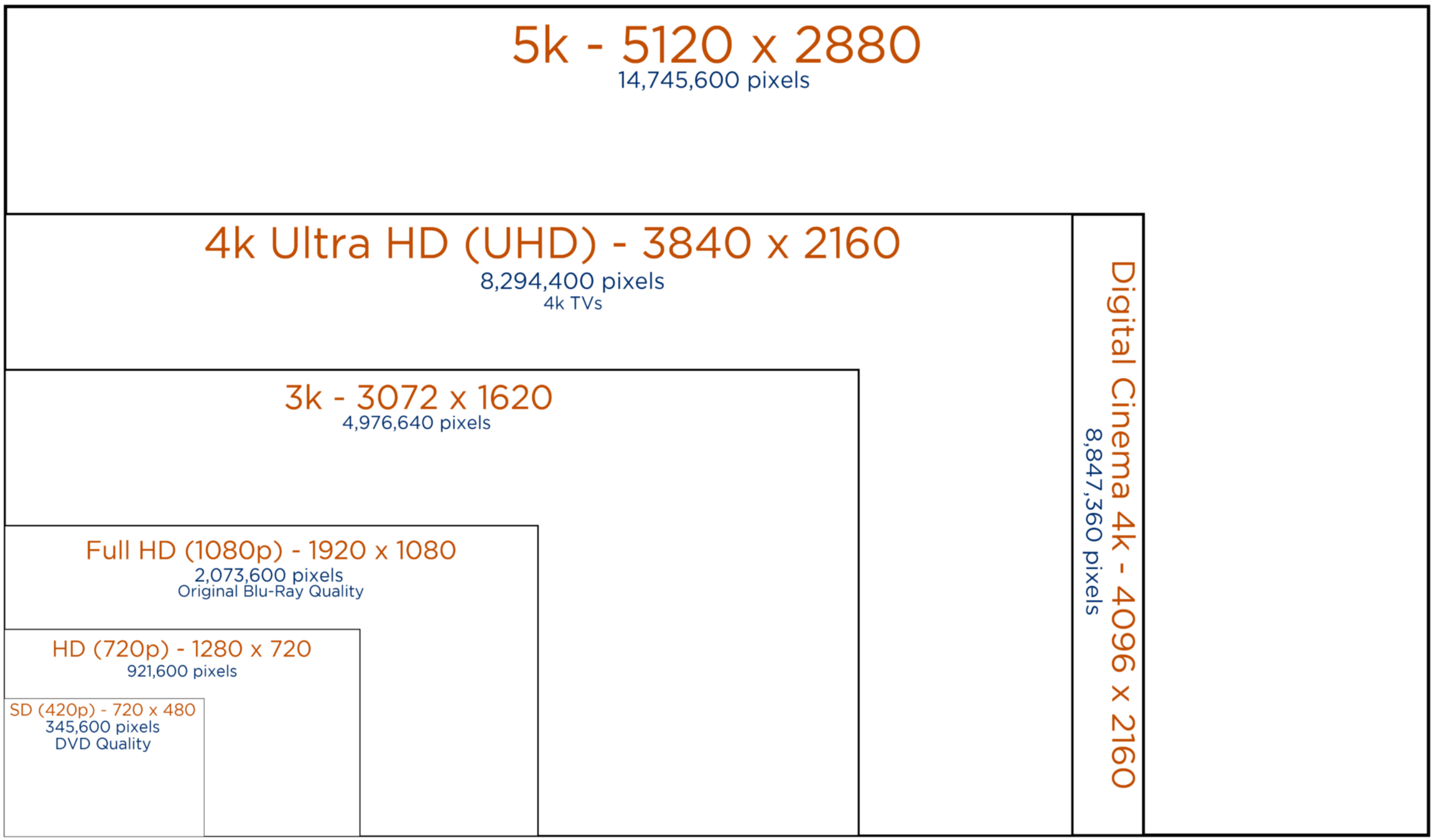
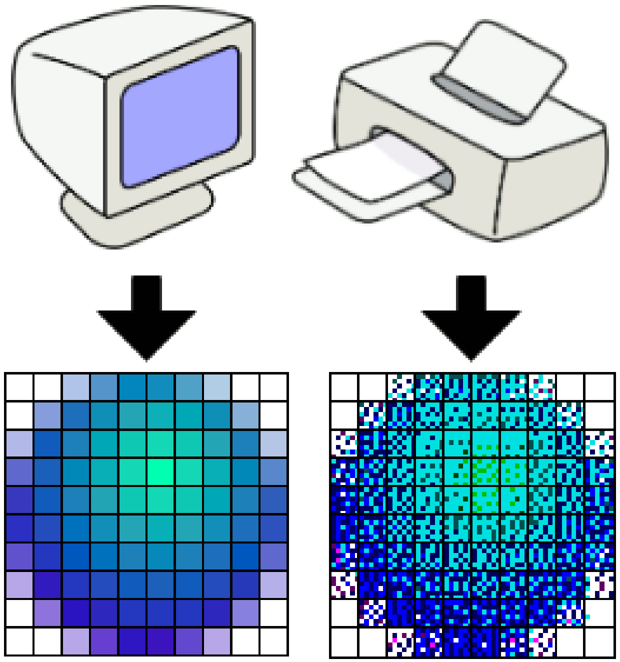


Image resolution

Image resolution= number of point per area (dpi, or ppi)

Conversion table (approximate)

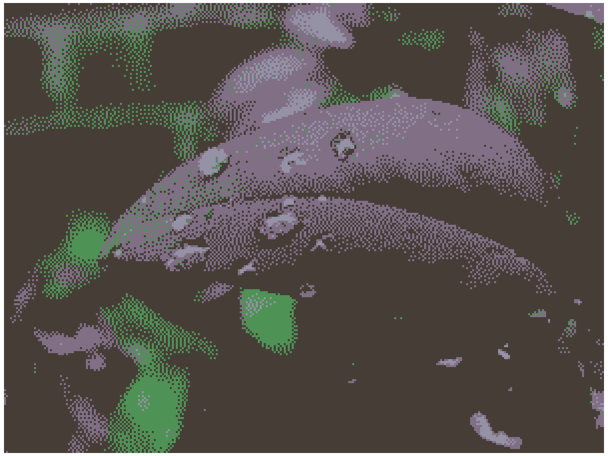
DPI (dot/in)	dpcm (dot/cm)	Pitch (µm)
72	28	353
96	38	265
150	59	169
300	118	85
2540	1000	10
4000	1575	6



Color depth

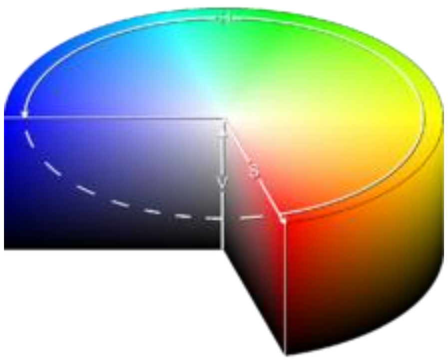
Color depth= is the number of bits used to indicate the color of a single pixel (for each of color components).

8 bit per color channel



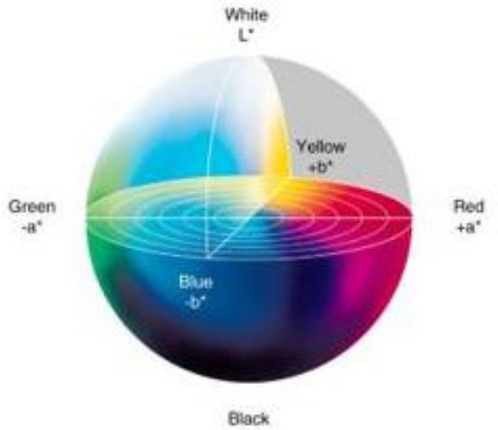
2 bit per color channel

Color space



Hue Saturation Lightnes (HSL)

RGB



L*a*b*

CMY

Image histogram

- An **image histogram** is a type of histogram that acts as a graphical representation of the tonal distribution in a digital image. It plots the number of pixels for each tonal value. By looking at the histogram for a specific image a viewer will be able to judge the entire tonal distribution at a glance.

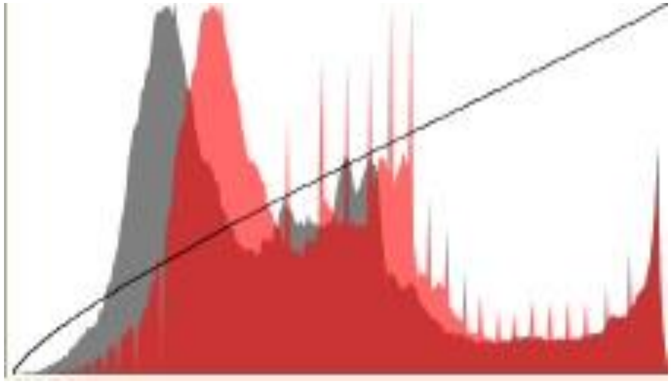
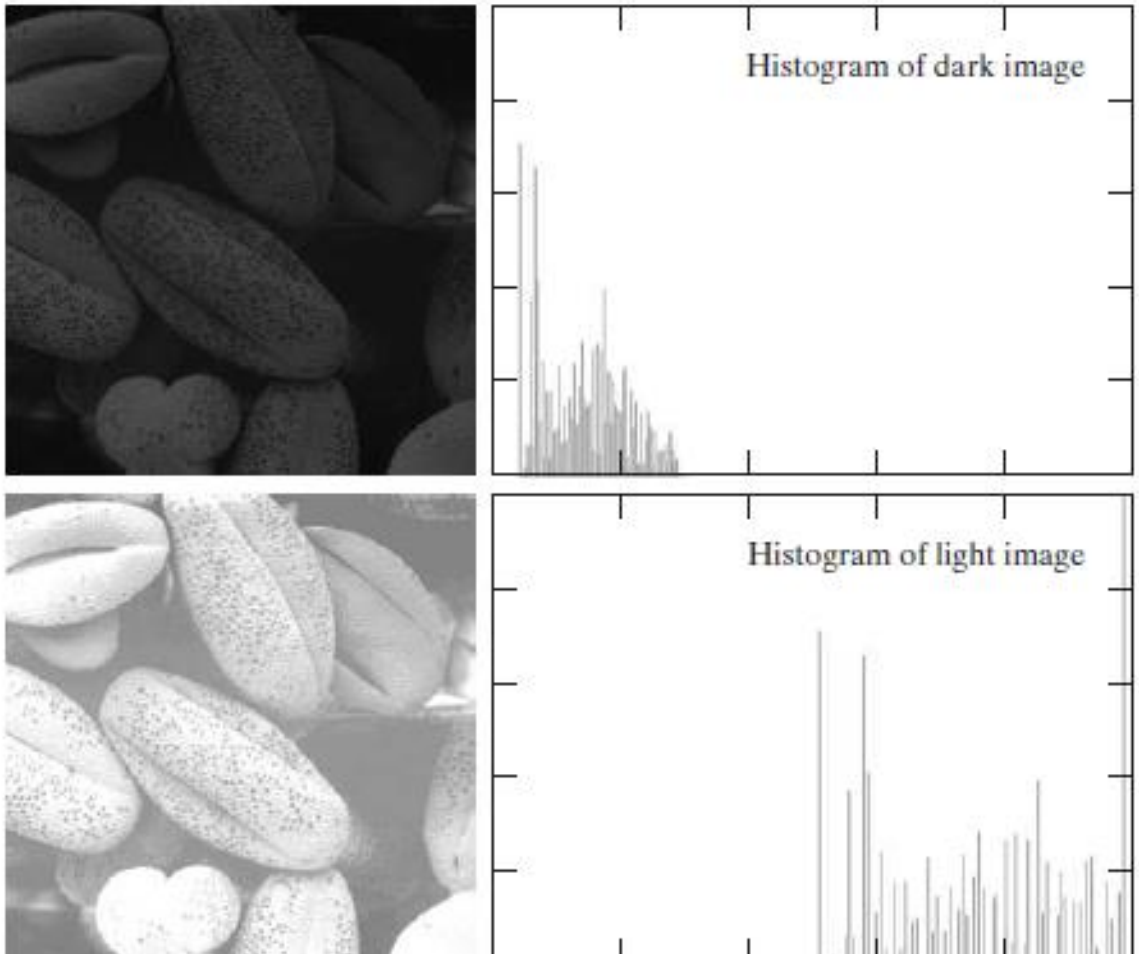
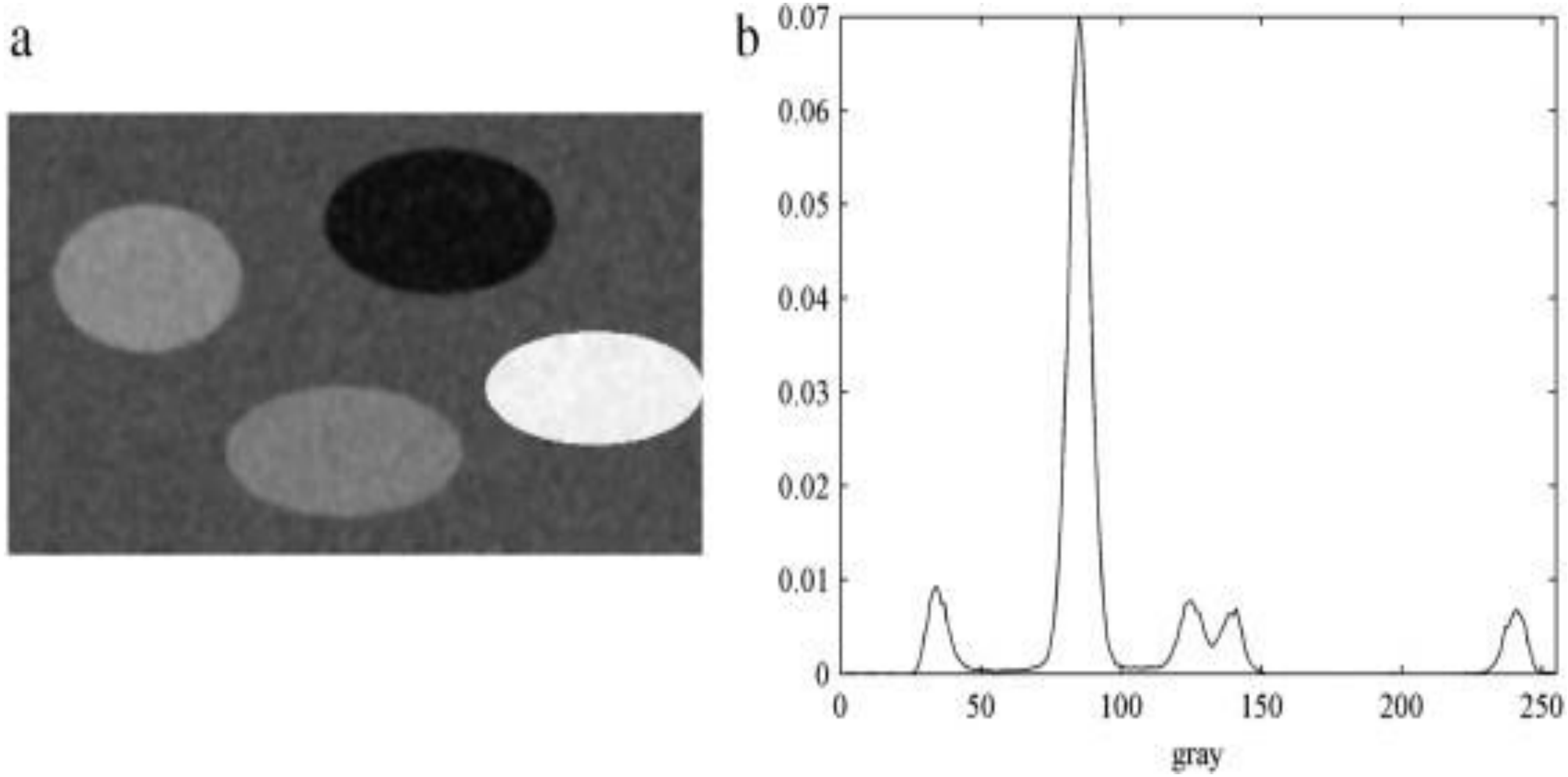


Image histogram

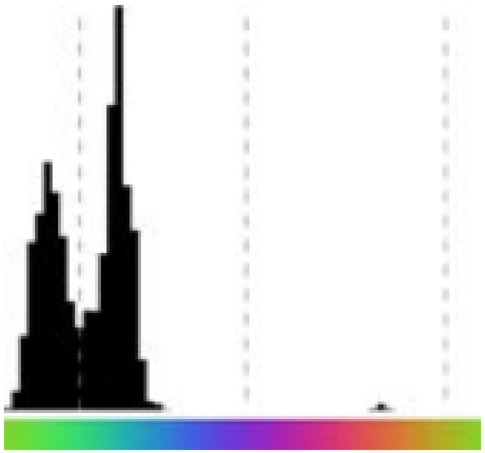


Histogram-based segmentation



Histogram-based segmentation

Input image



Histogram of the HUE component on HSV color-space

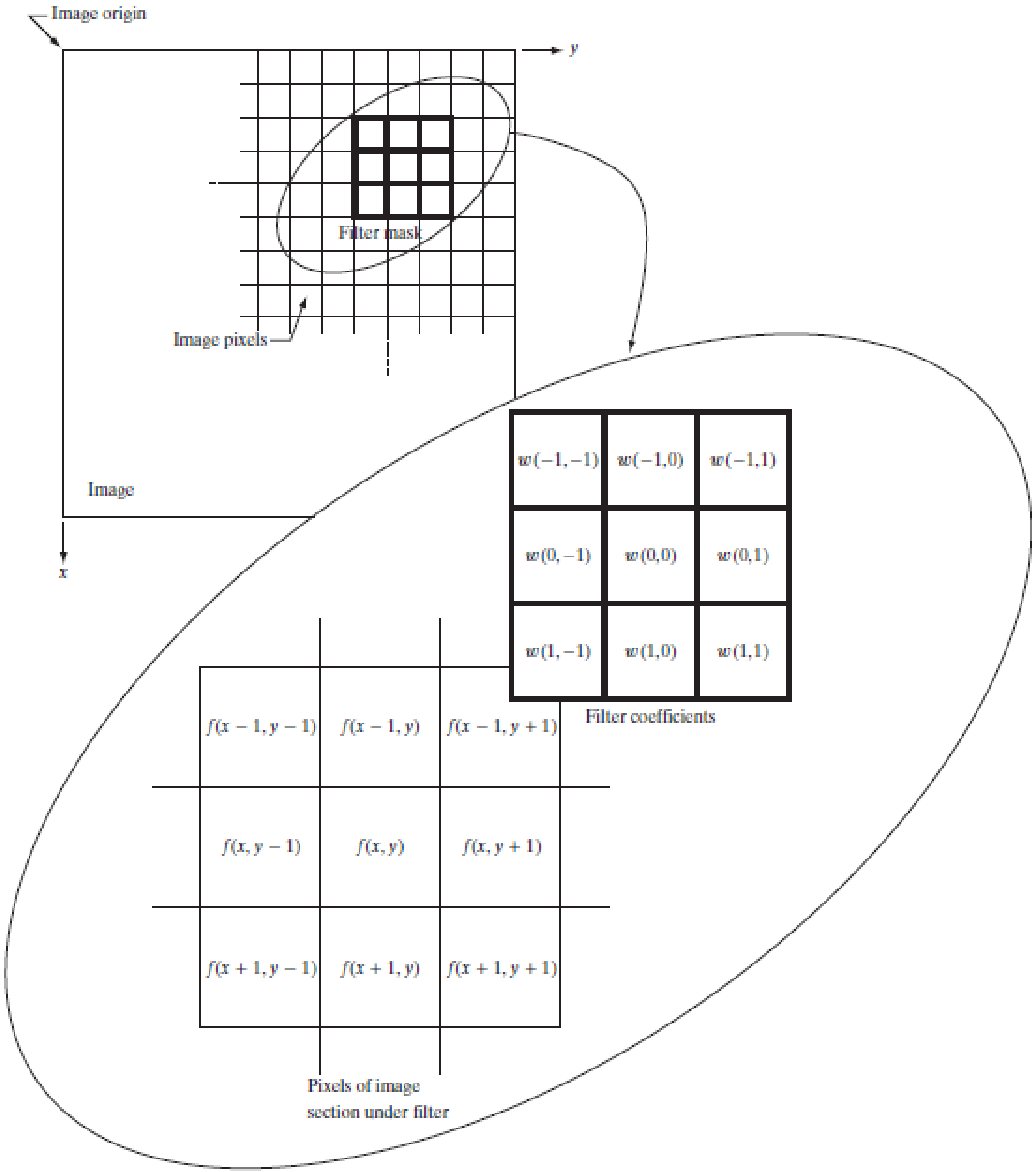
Image segmentation

Image Filtering

INPUT → **FILTER** → **OUTPUT**

Filtering is a technique for modifying or enhancing an image.

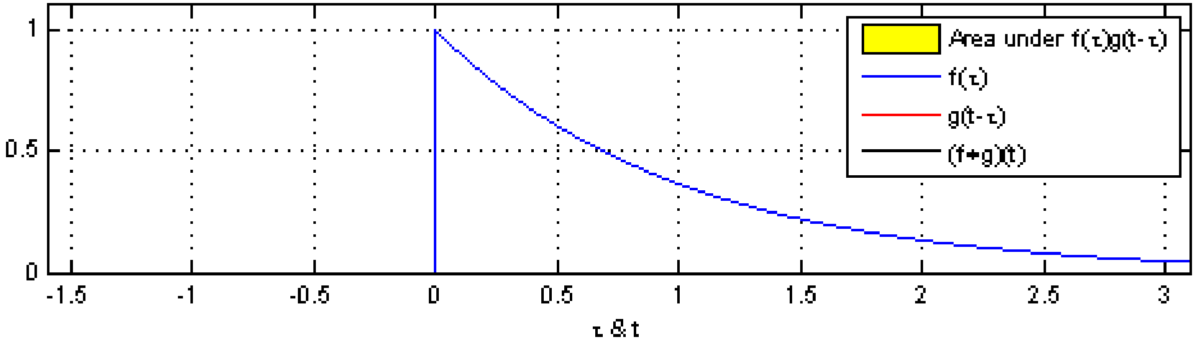
Image Filtering



The mechanics of linear spatial filtering using a 3x3 filter mask

Convolution

$$(f * g)(t) = \int_0^t f(\tau)g(t-\tau)\,d\tau \text{ for } f,g:[0,\infty] \to \mathbb{R}$$



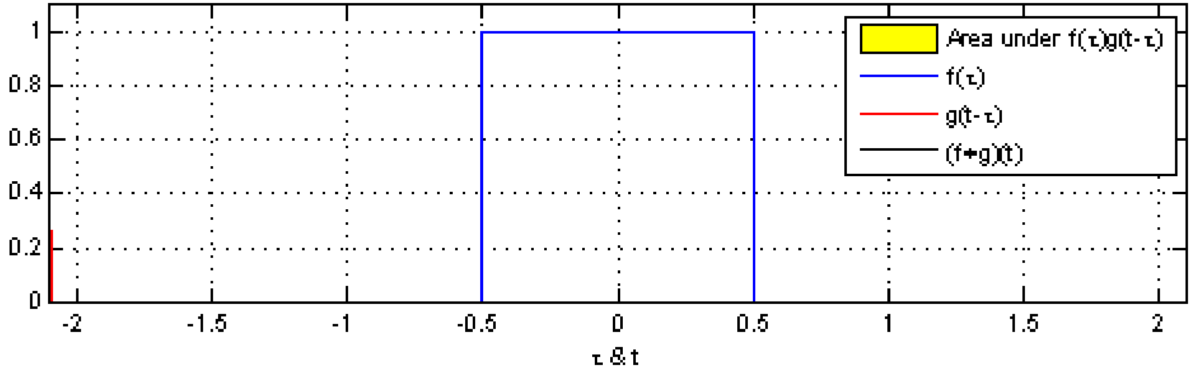
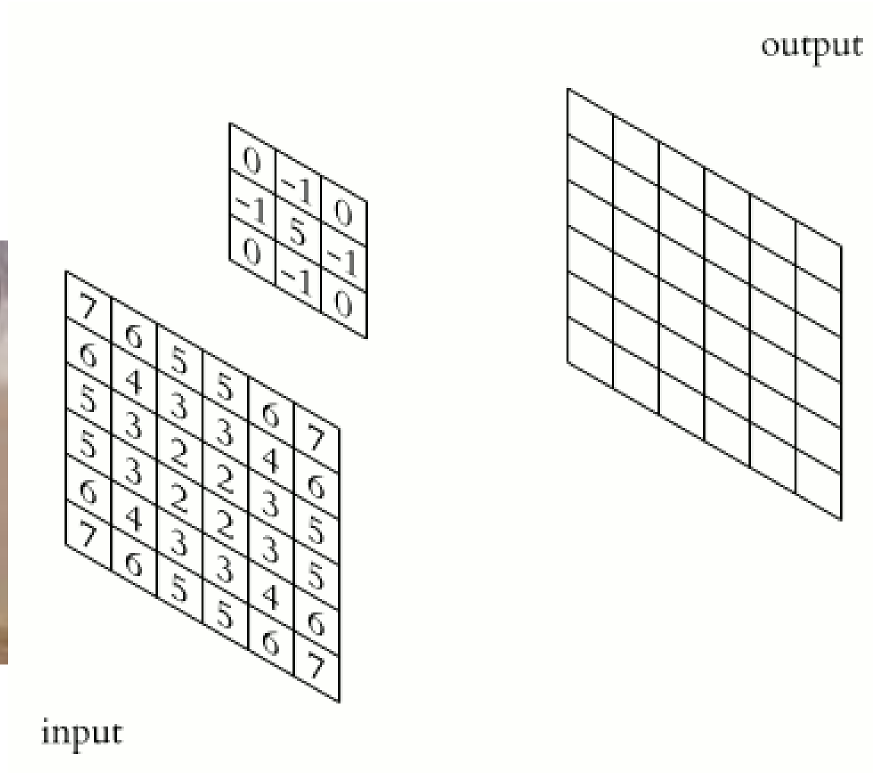


Image-convolution



Filtro di sharpening

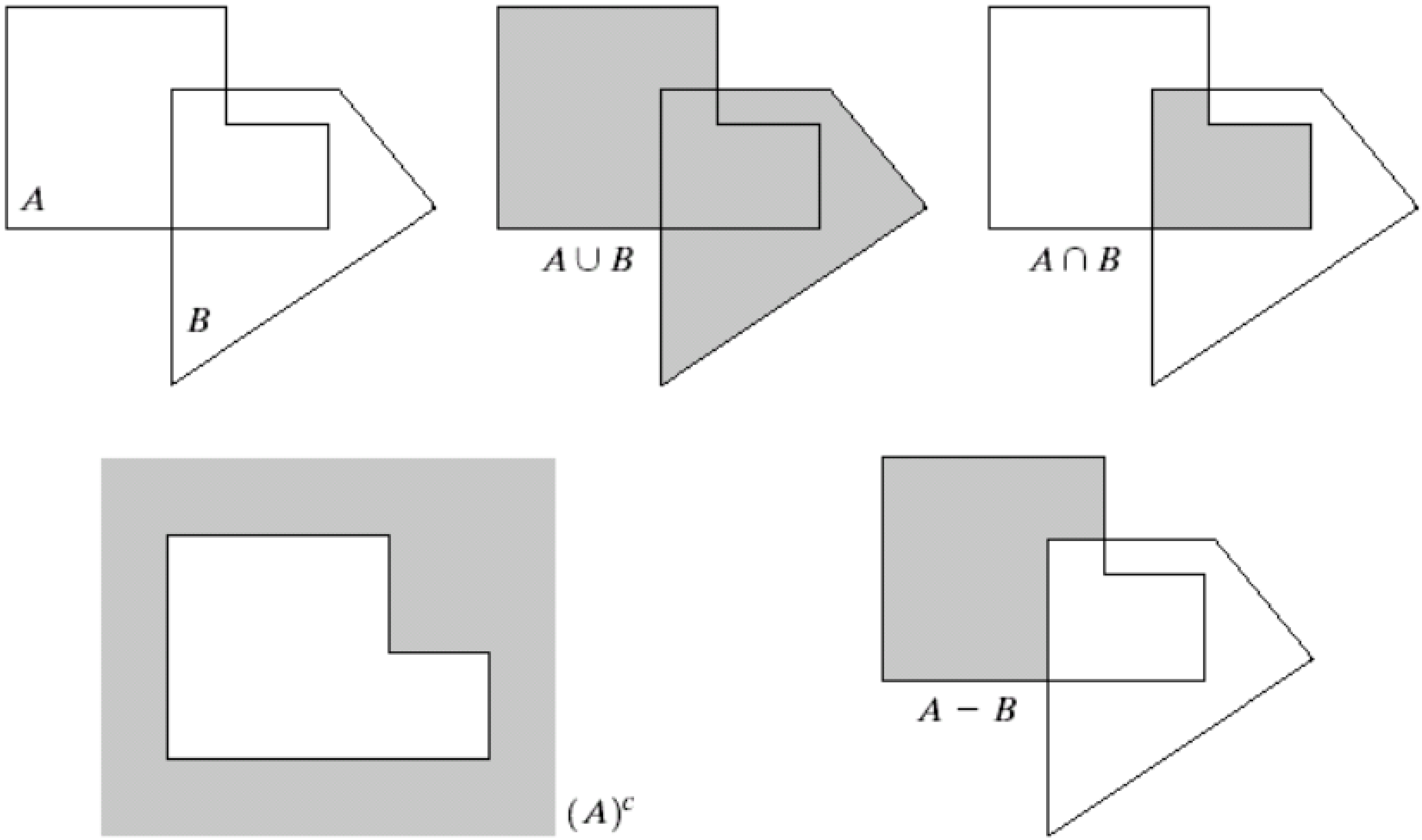
Morphological image processing

- **Morphological image processing** involves a set of methods and tools for extracting image components that are useful in the representation and description of region shape, such as boundaries, skeletons, and the convex hull.
- Usually it works on binary images,
  - from image-binarization,
  - from image segmentation

Mathematical morphology and Set theory

Set operators	Denotations
A Subset B	$A \subseteq B$
Union of A and B	$C = A \cup B$
Intersection of A and B	$C = A \cap B$
Disjoint	$A \cap B = \varnothing$
Complement of A	$A^c = \{ w \mid w \notin A \}$
Difference of A and B	$A - B = \{ w \mid w \in A, w \notin B \}$
Reflection of A	$\hat{A} = \{ w \mid w = -a \text{ for } a \in A \}$
Translation of set A by point $z(z_1, z_2)$	$(A)_z = \{ c \mid c = a + z, \text{ for } a \in A \}$

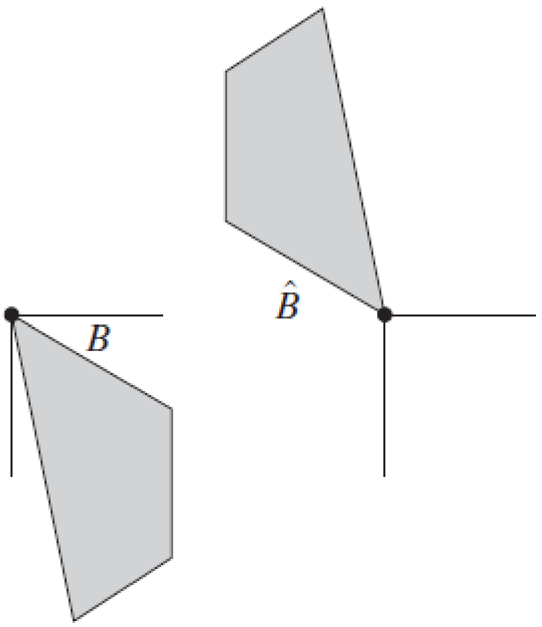
Mathematical morphology and Set theory



Mathematical morphology and Set theory

The **reflection** of a set B is defined as:

$$\hat{B} = \{w \mid w = -b, \quad \text{for} \quad b \in B\}$$

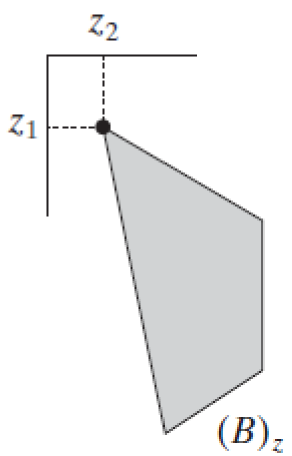


If B is the set of pixels (2-D points) representing an object in an image, then $\hat{B}$ is simply the set of points in B whose (x, y) coordinates have been replaced by $(-x, -y)$.

Mathematical morphology and Set theory

The **translation** of a set B by a point $\boldsymbol{z} = (z_1, z_2)$ is defined as:

$$(B)_z = \{c \mid c = b + z, \quad \text{for} \quad b \in B\}$$



If B is the set of pixels representing an object in an image, then $(B)_z$ is the set of points in B whose (x, y) coordinates have been replaced by $(x + z_1, y + z_2)$.

Structuring Element

- The processing in image morphology is based on **structuring elements** (SEs), i.e., the analog of convolution kernel in image filtering.
- A SE is a small set or subimage used to probe an image under study for properties of interest.

the origin of a SE must be specified

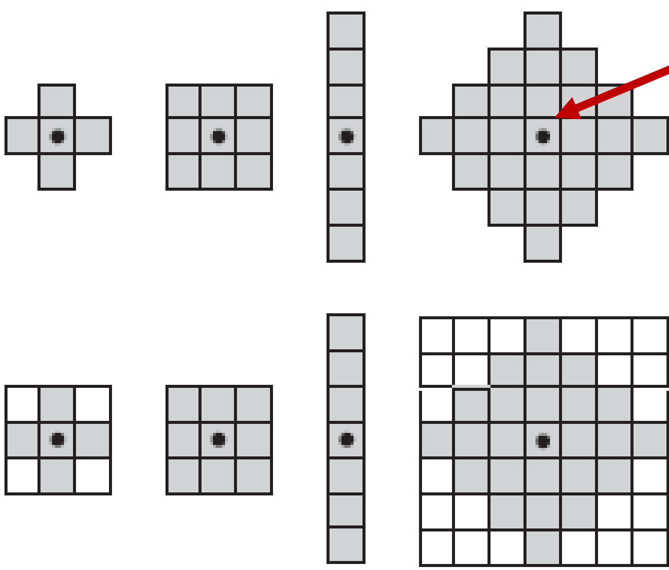
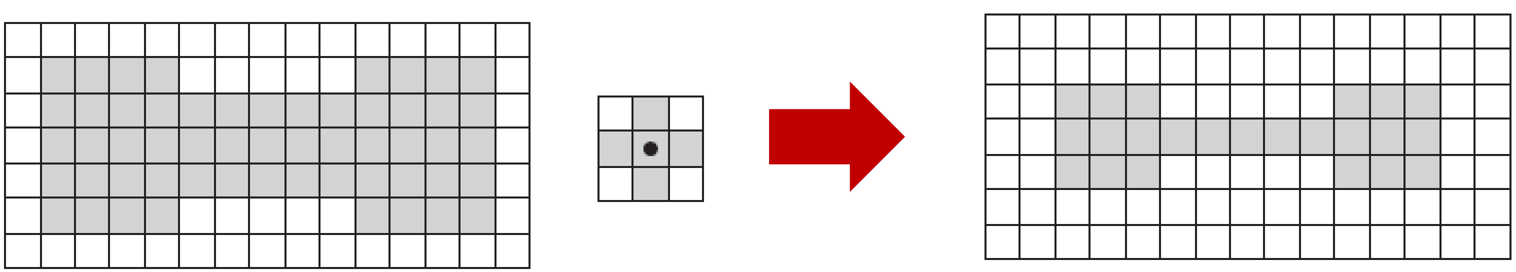


FIGURE 9.2 First row: Examples of structuring elements. Second row: Structuring elements converted to rectangular arrays. The dots denote the centers of the SEs.

Morphological operations

- The morphological operation between a binary image A and a SE B is defined as:
  - For each element (i,j) in A:
    - shift the SE B in (i,j)
    - If SE is verified then A(i,j)=1, otherwise A(i,j)=0



Morphological operations

- $\ominus$ Erosion
- $\oplus$ Dilation
- $\circ$ Opening
- $\bullet$ Closing
- $\otimes$ [Hit-or-Miss transform]

Erosion

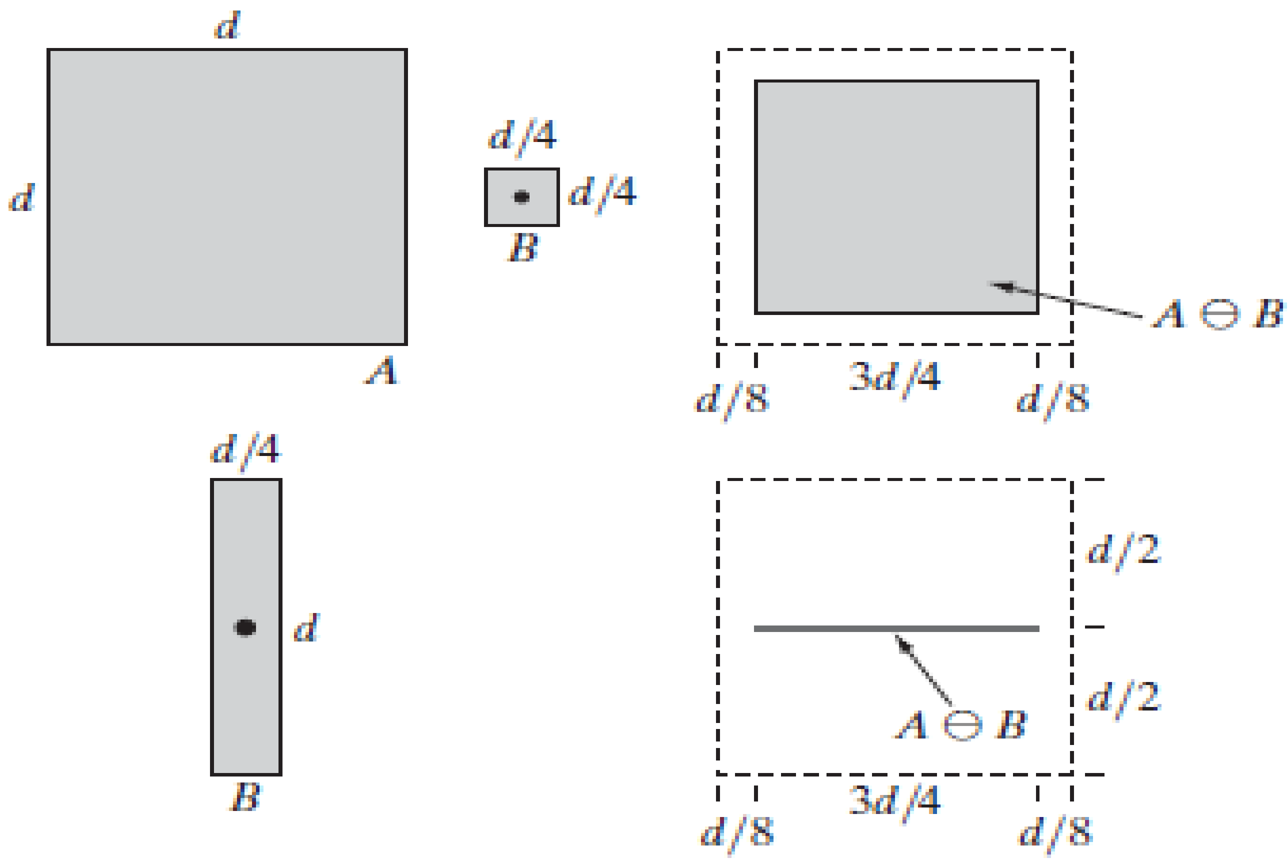
With A and B as sets in Z^2, the erosion of A by B, denoted $A \ominus B$, is defined as

$$A \ominus B = \{z | (B)_z \subseteq A\}$$

In words, this equation indicates that the erosion of A by B is the set of all points z such that B, translated by z, is contained in A. In the following discussion, set B is assumed to be a structuring element.

we can view erosion as a **morphological filtering** operation in which image details smaller than the structuring element are filtered (removed) from the image.

Erosion



a b c
d e

FIGURE 9.4 (a) Set A. (b) Square structuring element, B. (c) Erosion of A by B, shown shaded. (d) Elongated structuring element. (e) Erosion of A by B using this element. The dotted border in (c) and (e) is the boundary of set A, shown only for reference.

Dilation

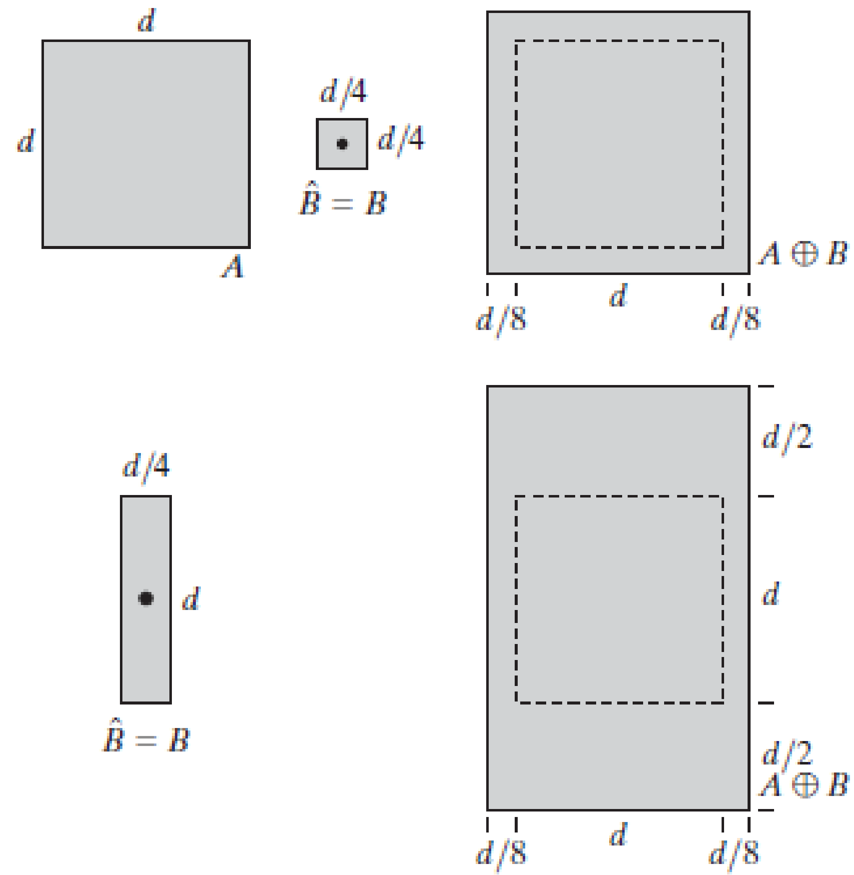
With A and B as sets in Z^2, the *dilation* of A by B, denoted $A \oplus B$, is defined as

$$A \oplus B = \left\{ z \,|\, (\hat{B})_z \cap A \neq \varnothing \right\}$$

This equation is based on reflecting B about its origin, and shifting this reflection by z (see Fig. 9.1). The dilation of A by B then is the set of all displacements, z, such that $\hat{B}$ and A overlap by at least one element.

Unlike erosion, which is a shrinking or thinning operation, dilation **"grows"** or **"thickens"** objects in a binary image. The specific manner and extent of this thickening is controlled by the shape of the structuring element used.

Dilation



a b c
d e

FIGURE 9.6
(a) Set A.
(b) Square structuring element (the dot denotes the origin).
(c) Dilation of A by B, shown shaded.
(d) Elongated structuring element. (e) Dilation of A using this element. The dotted border in (c) and (e) is the boundary of set A, shown only for reference

Opening

The *opening* of set A by structuring element B, denoted $A \circ B$, is defined as

$$A \circ B = (A \ominus B) \oplus B$$

Thus, the opening A by B is the erosion of A by B, followed by a dilation of the result by B.

Opening generally smoothes the contour of an object, breaks narrow isthmuses, and eliminates thin protrusions.

Opening

$$A \circ B = (A \ominus B) \oplus B$$

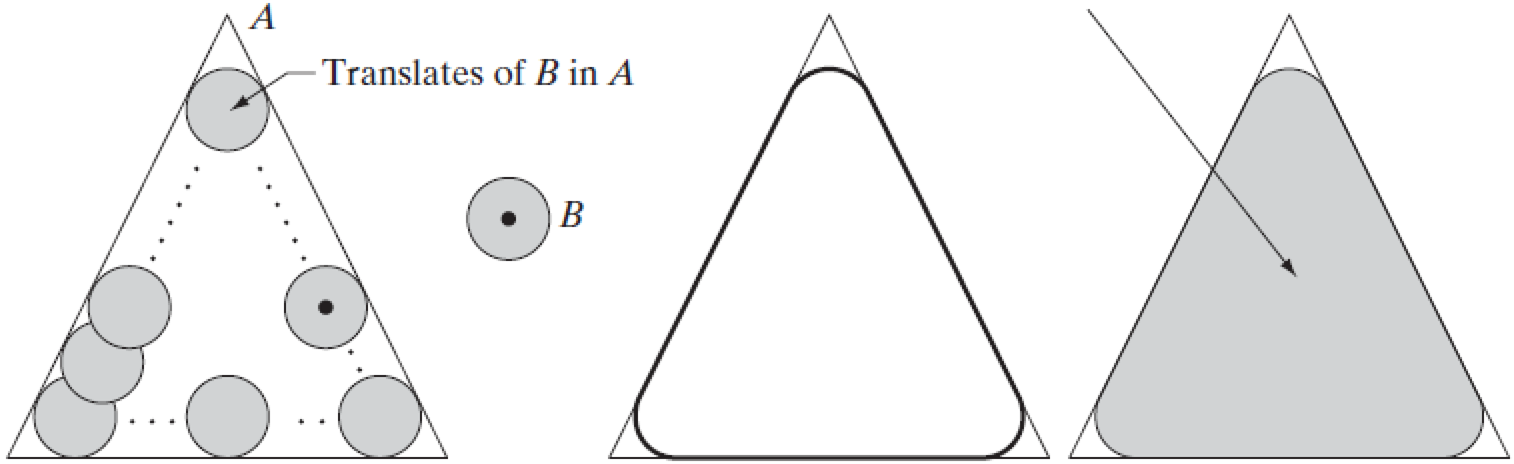
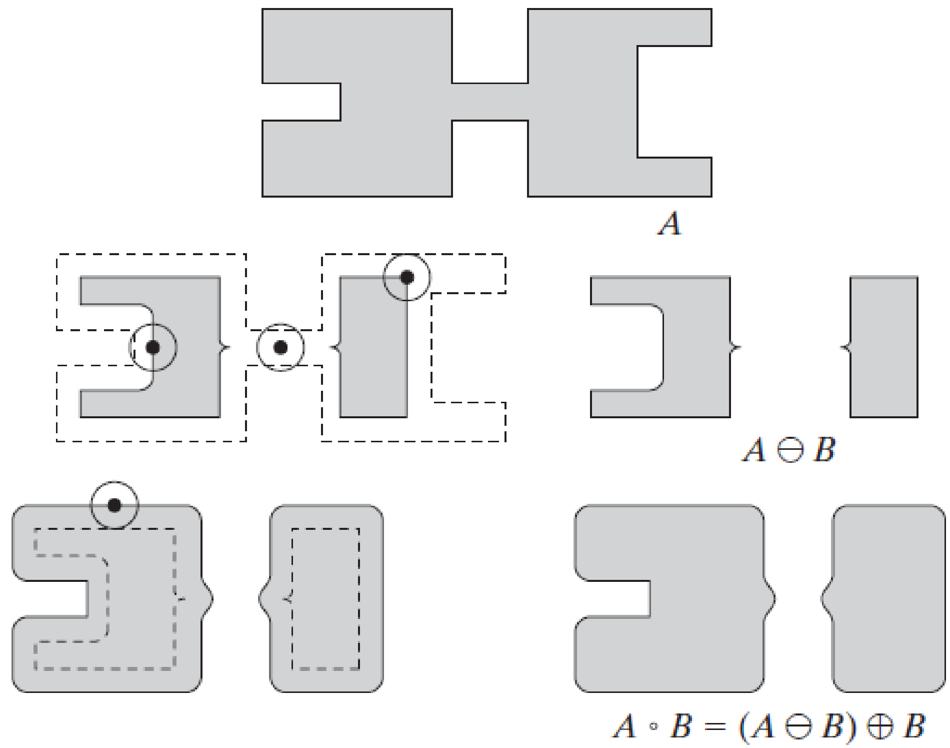


FIGURE 9.8 (a) Structuring element B "rolling" along the inner boundary of A (the dot indicates the origin of B). (b) Structuring element. (c) The heavy line is the outer boundary of the opening. (d) Complete opening (shaded). We did not shade A in (a) for clarity.

Opening



When completed the **erosion** a disjoint figure is obtained (c). Note the elimination of the bridge between the two main sections. Its width was thin in relation to the diameter of the structuring element;
After the **dilation** outward pointing corners were rounded, whereas inward pointing corners were not affected.

Closing

Similarly, the *closing* of set A by structuring element B, denoted $A \bullet B$, is defined as

$$A \bullet B = (A \oplus B) \ominus B$$

which says that the closing of A by B is simply the dilation of A by B, followed by the erosion of the result by B.

Closing also tends to smooth sections of contours but, as opposed to opening, it generally fuses narrow breaks and long thin gulfs, eliminates small holes, and fills gaps in the contour.

Closing

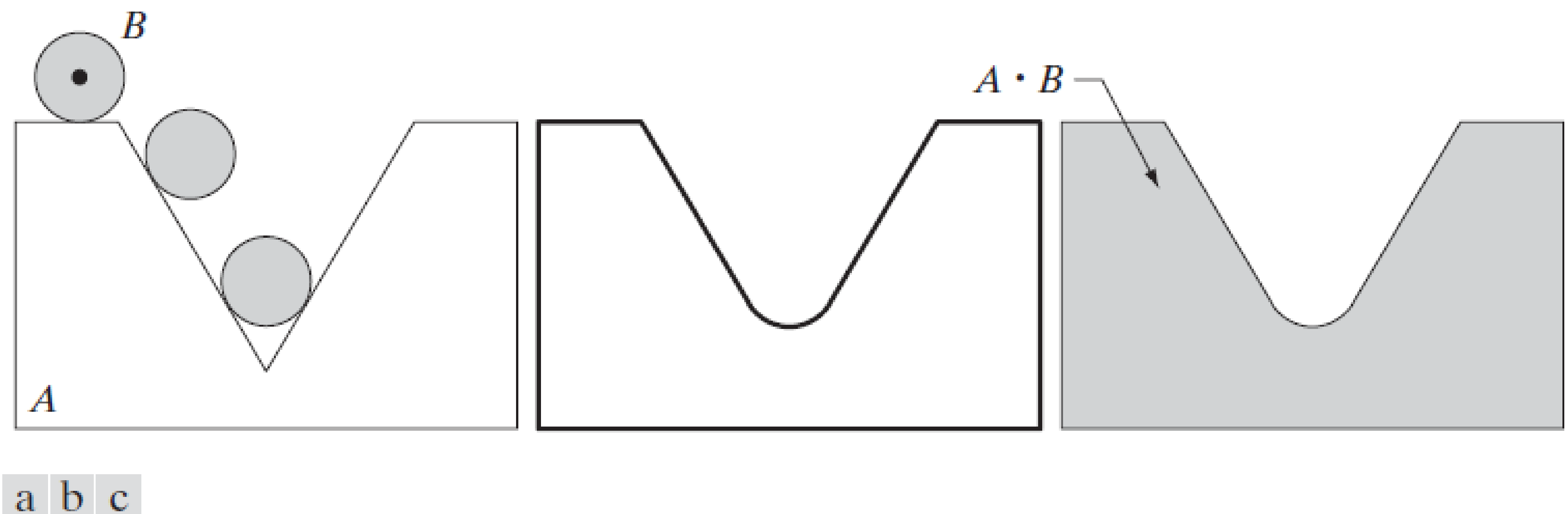
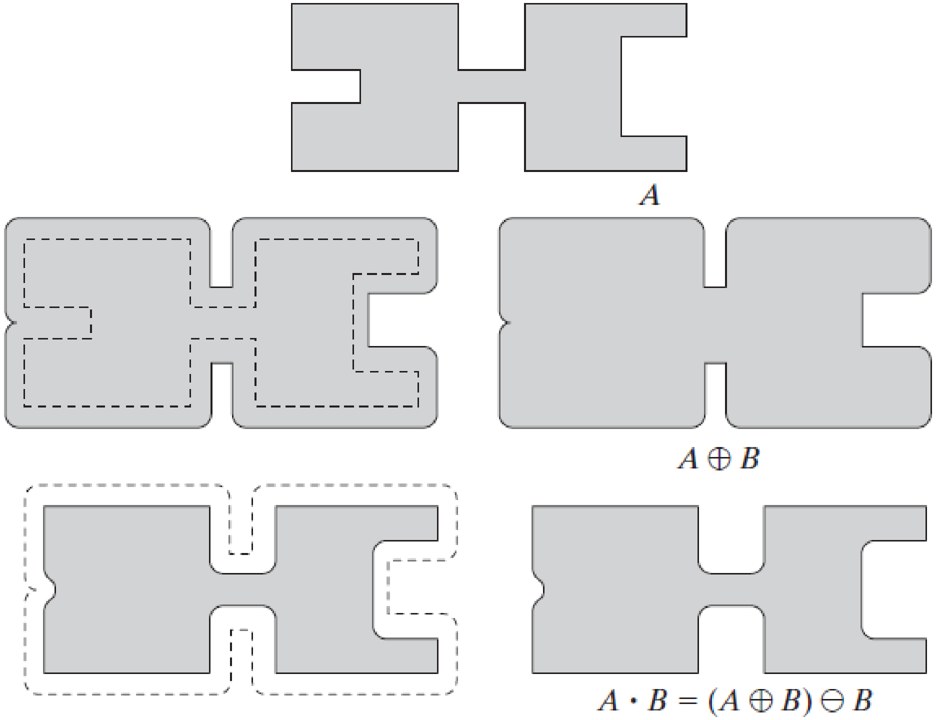


FIGURE 9.9 (a) Structuring element B "rolling" on the outer boundary of set A. (b) The heavy line is the outer boundary of the closing. (c) Complete closing (shaded). We did not shade A in (a) for clarity.

Closing



We note that the inward pointing corners were rounded, whereas the outward pointing corners remained unchanged. The leftmost intrusion on the boundary of was reduced in size significantly, because the disk did not fit there.

Algorithms

- Boundary extraction,
- Hole filling,
- Extraction of connected components,

Boundary extraction

The boundary of a set A, denoted by $\beta(A)$, can be obtained by first eroding A by B and then performing the set difference between A and its erosion. That is,

$$\beta(A) = A - (A \ominus B)$$

where B is a suitable structuring element.

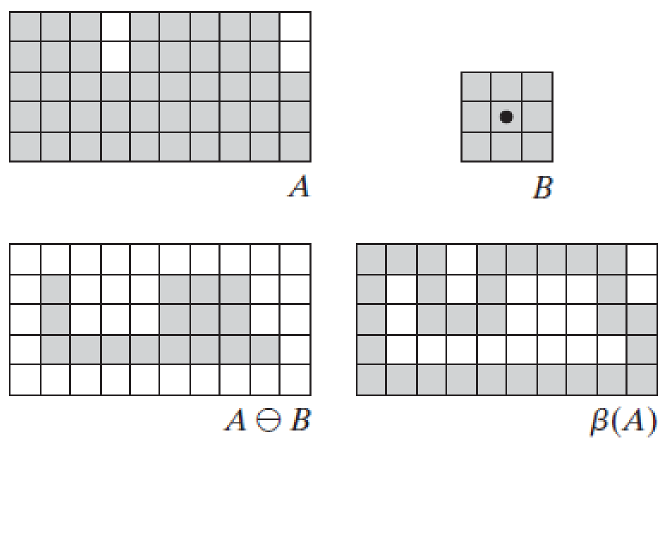
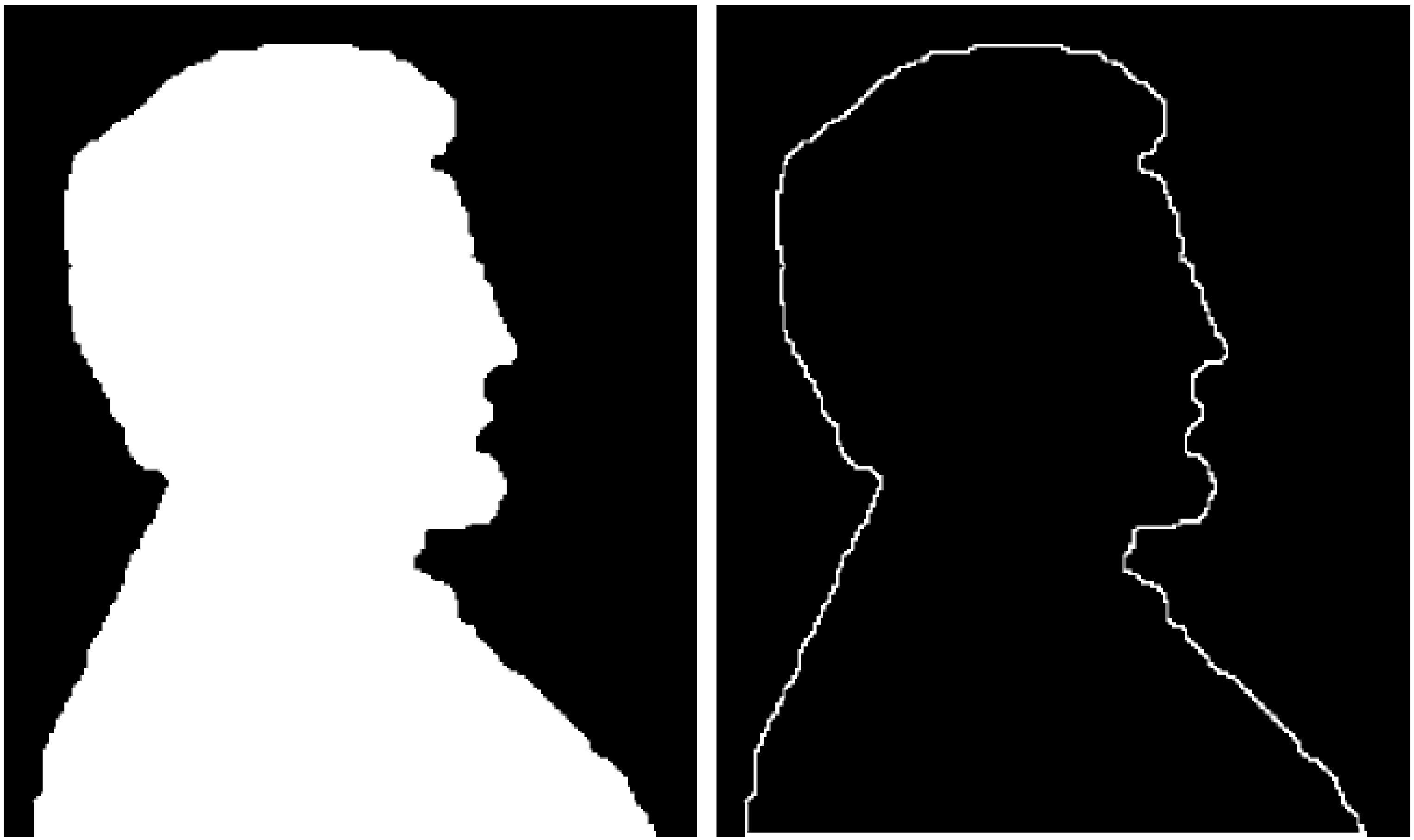


FIGURE 9.13 (a) Set A. (b) Structuring element B. (c) A eroded by B. (d) Boundary, given by the set difference between A and its erosion.

Boundary extraction



Hole filling

- A **hole** may be defined as a background region surrounded by a connected border of foreground pixels.

An **hole filling algorithm** can be define using:
-set dilation,
-complementation, and
-intersection.

Hole filling

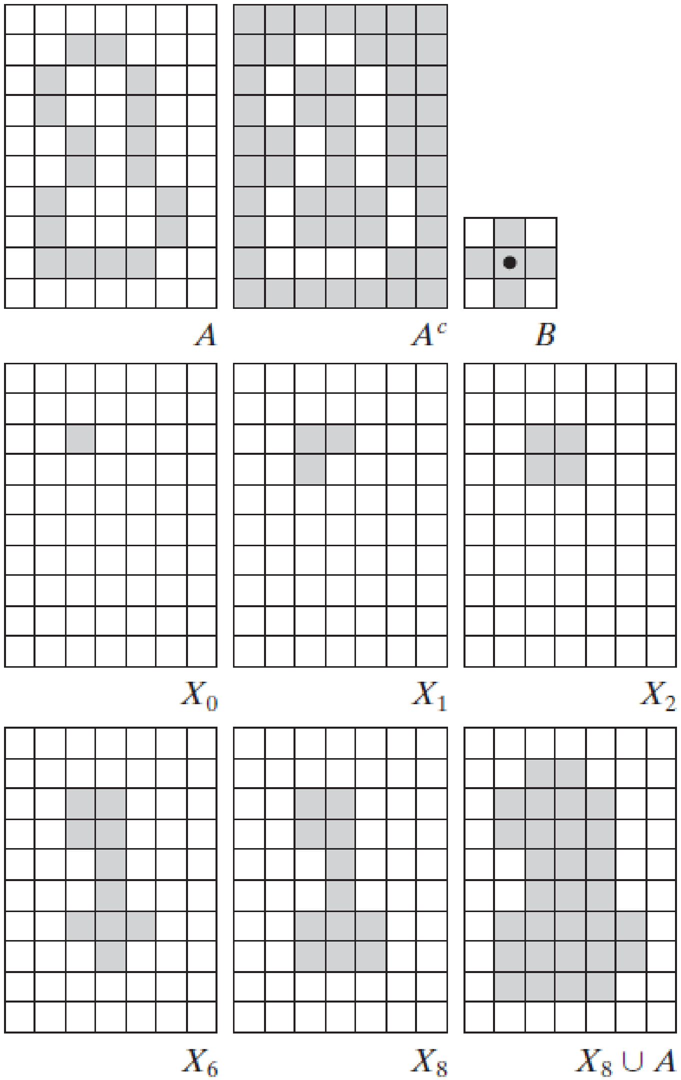
Let A denote a set whose elements are 8-connected boundaries, each boundary enclosing a background region (i.e., a hole). Given a point in each hole, the objective is to fill all the holes with 1s.

We begin by forming an array, X_0, of 0s (the same size as the array containing A), except at the locations in X_0 corresponding to the given point in each hole, which we set to 1. Then, the following procedure fills all the holes with 1s:

$$X_k = (X_{k-1} \oplus B) \cap A^c \qquad k = 1, 2, 3, \ldots$$

where B is the symmetric structuring element in Fig. 9.15(c). The algorithm terminates at iteration step k if $X_k = X_{k-1}$. The set X_k then contains all the filled holes. The set union of X_k and A contains all the filled holes and their boundaries.

Hole filling



a	b	c
d	e	f
g	h	i

FIGURE 9.15 Hole filling. (a) Set A (shown shaded). (b) Complement of A. (c) Structuring element B. (d) Initial point inside the boundary. (e)–(h) Various steps of Eq. (9.5-2). (i) Final result [union of (a) and (h)].

Hole filling

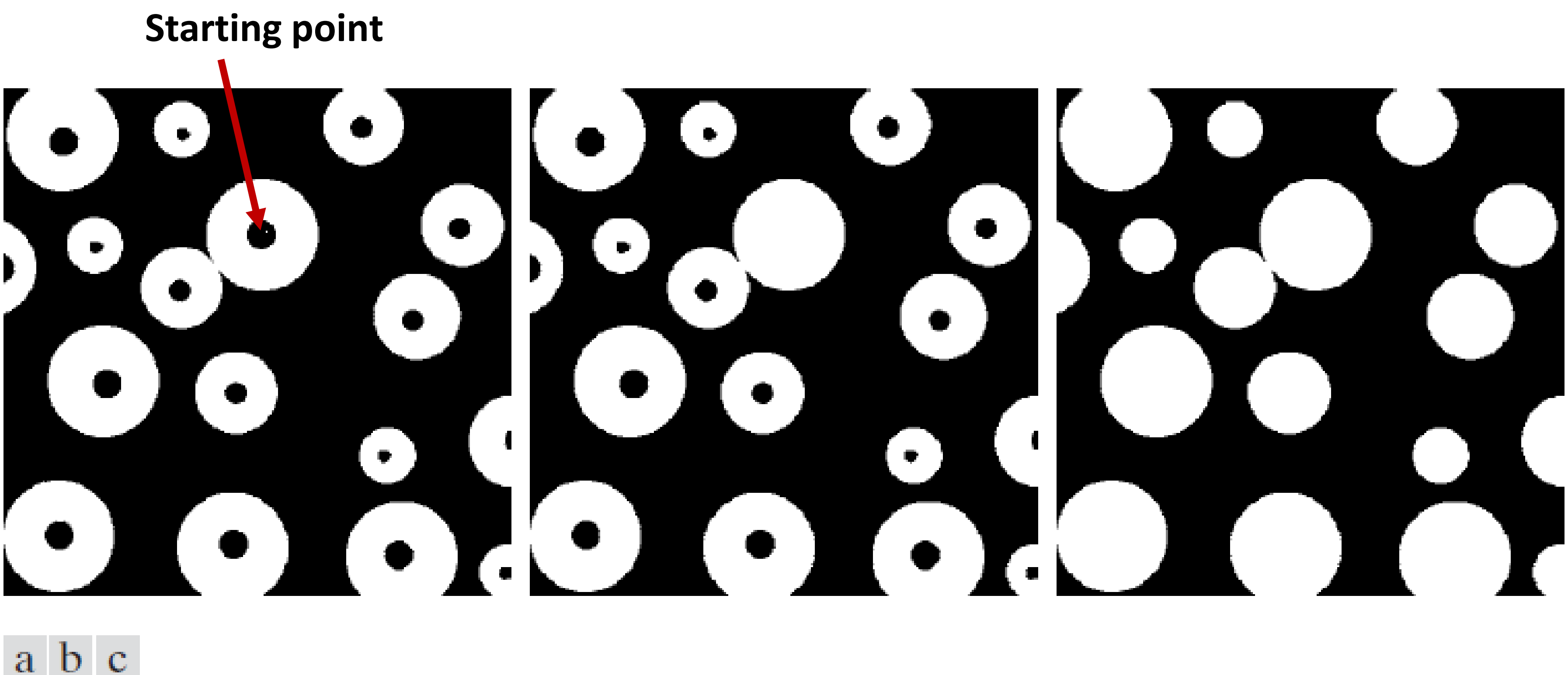
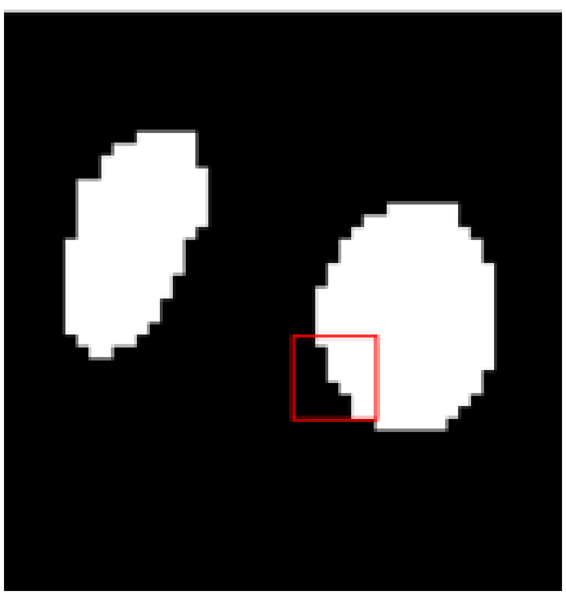
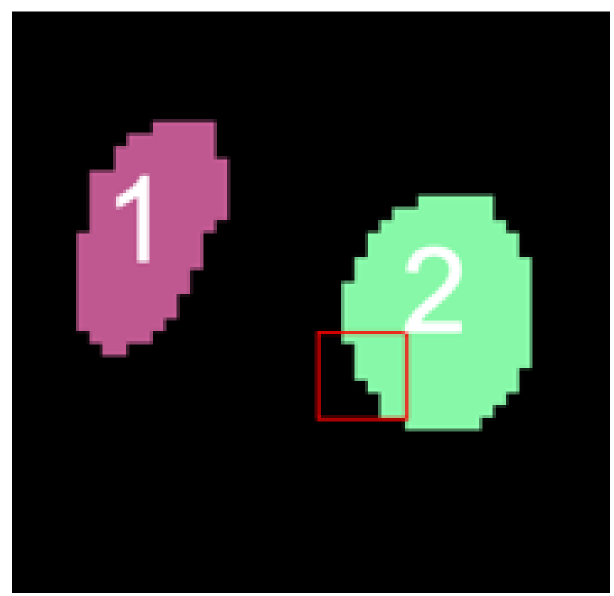


FIGURE 9.16 (a) Binary image (the white dot inside one of the regions is the starting point for the hole-filling algorithm). (b) Result of filling that region. (c) Result of filling all holes.

Connected components

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





Connected components

- Let be $\mathbf{A}$ set containing one or more connected components, and form an array $\mathbf{X_0}$ (of the same size as the array containing $\mathbf{A}$) whose elements are 0s (background values), except at each location known to correspond to a point in each connected component in $\mathbf{A}$, which we set to 1 (foreground value). The objective is to start with $\mathbf{X_0}$ and find all the connected components.
- The following iterative procedure is defined:

$$X_k = (X_{k-1} \oplus B) \cap A \qquad k = 1, 2, 3, \ldots$$

The procedure terminates when $X_k = X_{k-1}$, with X_k containing all the connected components of the input image.

Connected component

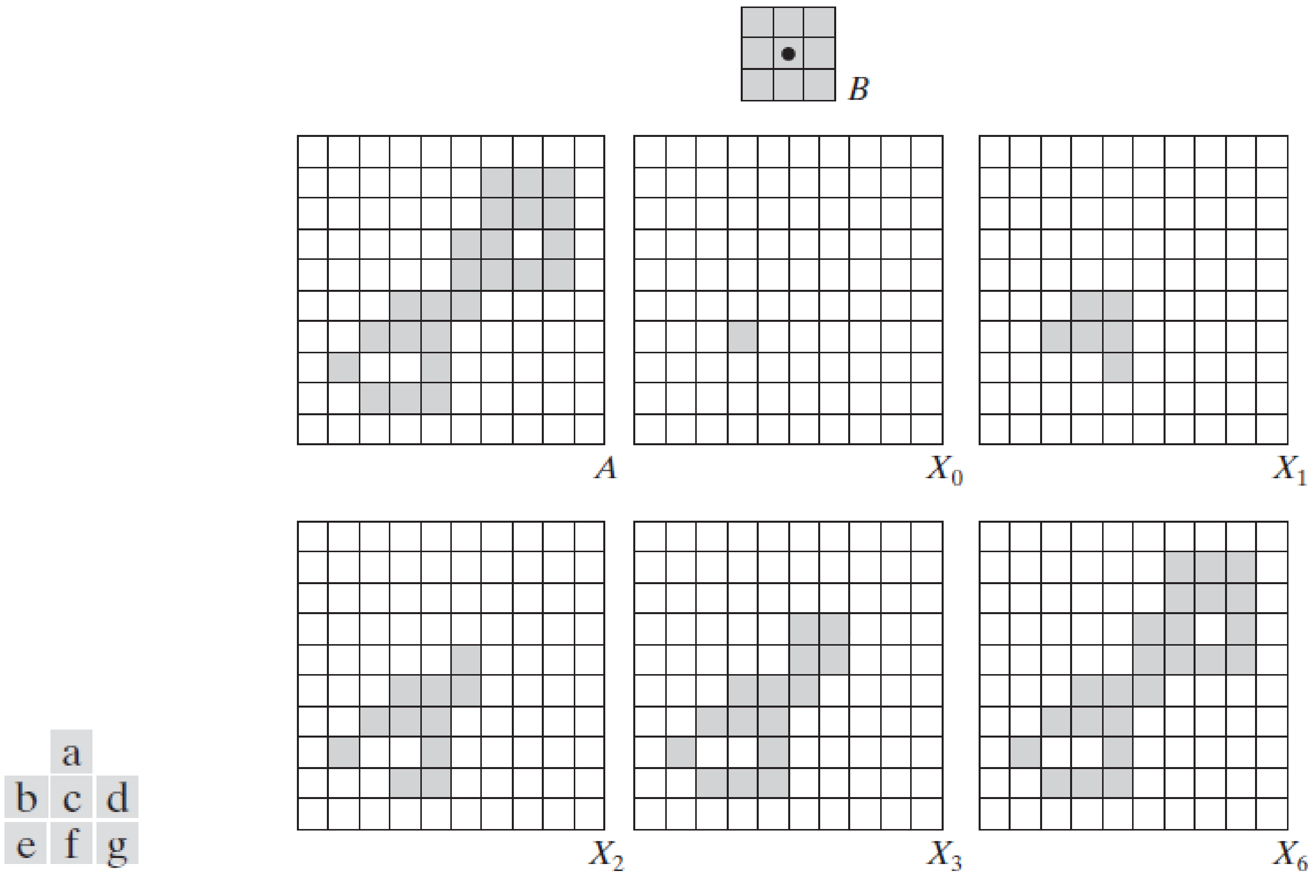


FIGURE 9.17 Extracting connected components. (a) Structuring element. (b) Array containing a set with one connected component. (c) Initial array containing a 1 in the region of the connected component. (d)–(g) Various steps in the iteration of Eq. (9.5-3).

Shape characterization and measuring

- Area,
- Perimeter,
- Bounding box,
- Centroid,
- Circularity,
- Convex Hull,
- Main orientation
- Eccentricity,
- Elongation

Area

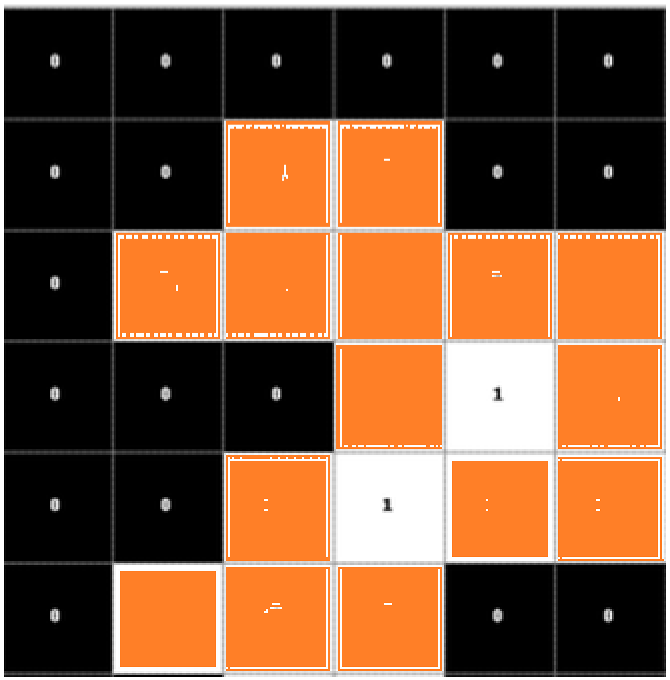
- Actual number of pixels in the region, returned as a scalar.

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

AREA=17

Perimeter

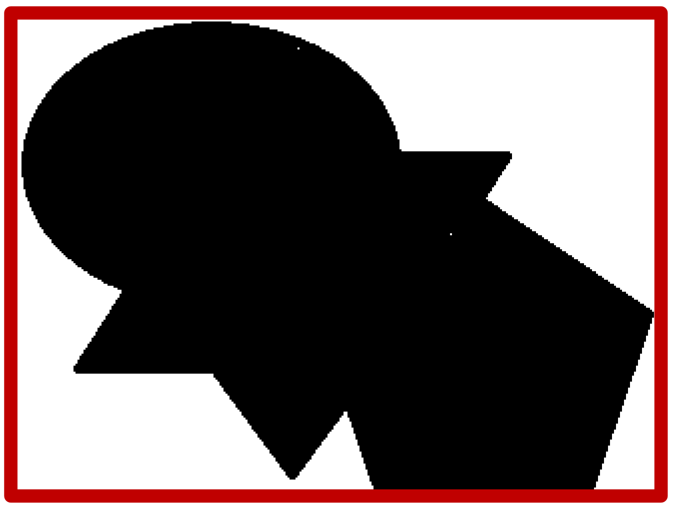
- Actual number of pixels in the contour of the region, returned as a scalar.



PERIMETER=15

Bounding box

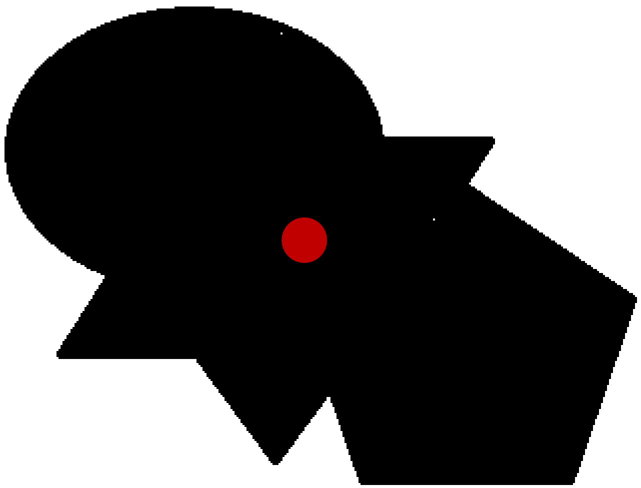
- the smallest box containing the region



Or…

Centroid

- Center of mass of the region

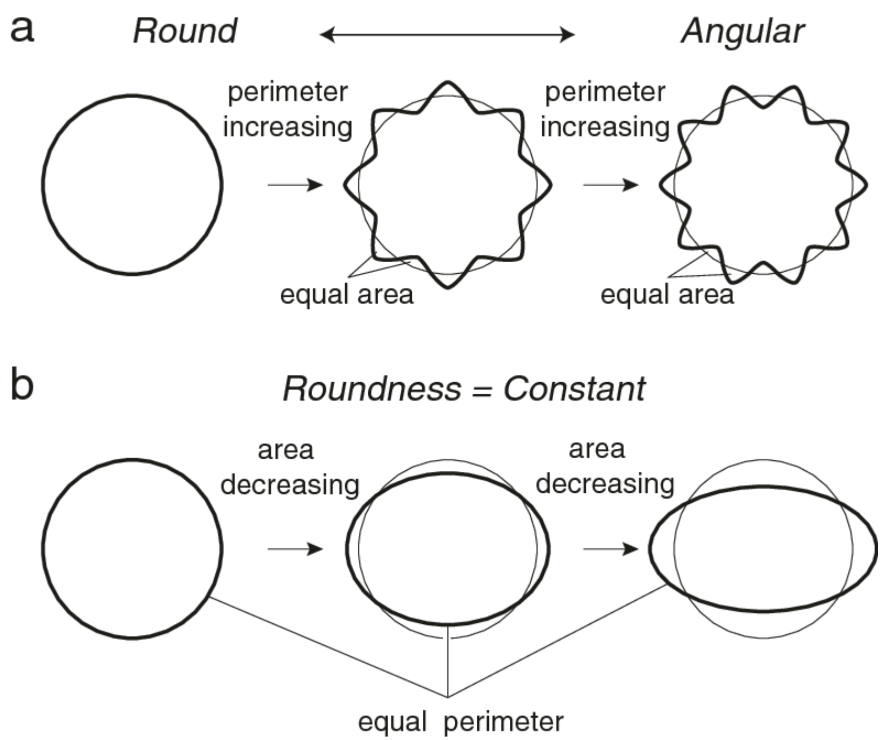


$$\mathbf{C} = \frac{\mathbf{x}_1 + \mathbf{x}_2 + \cdots + \mathbf{x}_k}{k}$$

Circularity

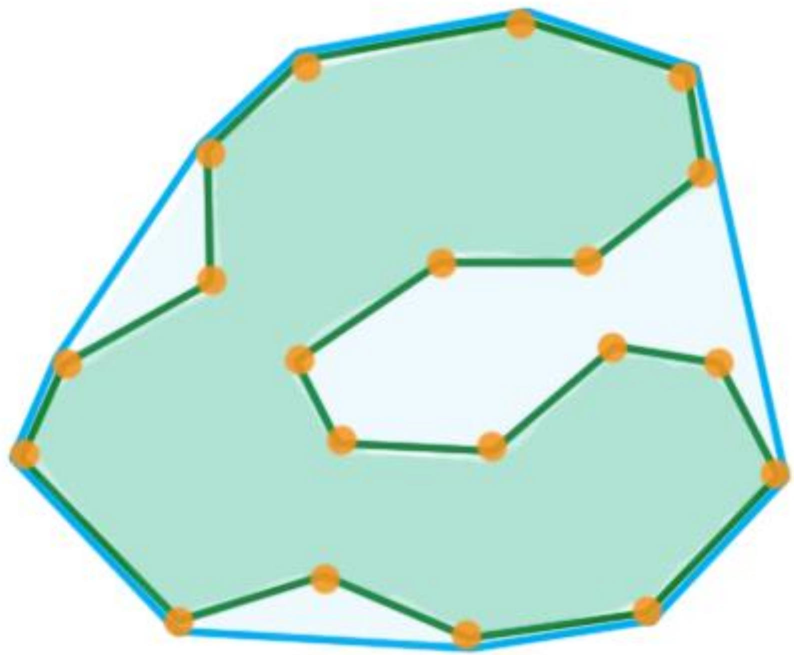
Cir=(4*Area*pi)/(Perimeter²)

- Roundness or compactness of objects. For a perfect circle, the circularity value is 1



Convex hull

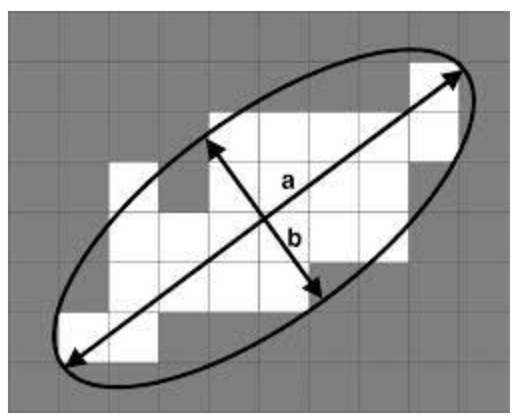
- the smallest convex shape enclosing a given shape.



NOTE: from the convex hull it is possible to extract other properties like the area of the convex hul

Major axis

- The major axis is the (x,y) endpoints of the longest line that can be drawn through the object
  - The major axis endpoints (x_1,y_1) and (x_2,y_2) are found by computing the pixel distance between every combination of border pixels in the object boundary and finding the pair with the maximum length, or
  - The major axis can be computed as the major axis of the ellipse that is estimate from the image

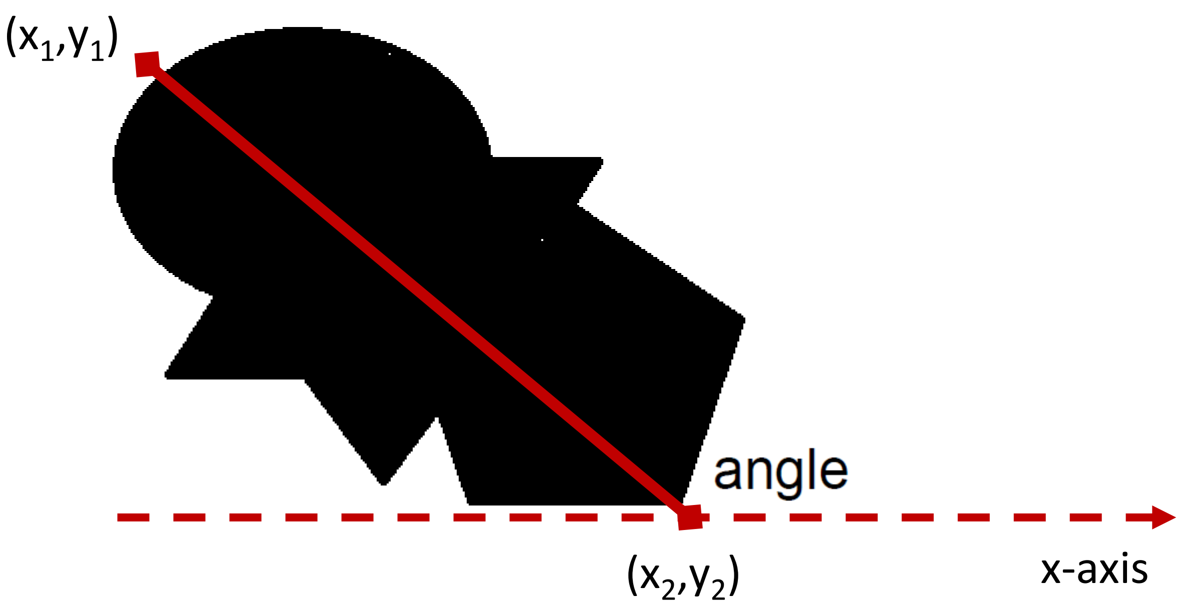


Main orientation

- The **main orientation** is the angle between the major-axis and the x-axis of the image:

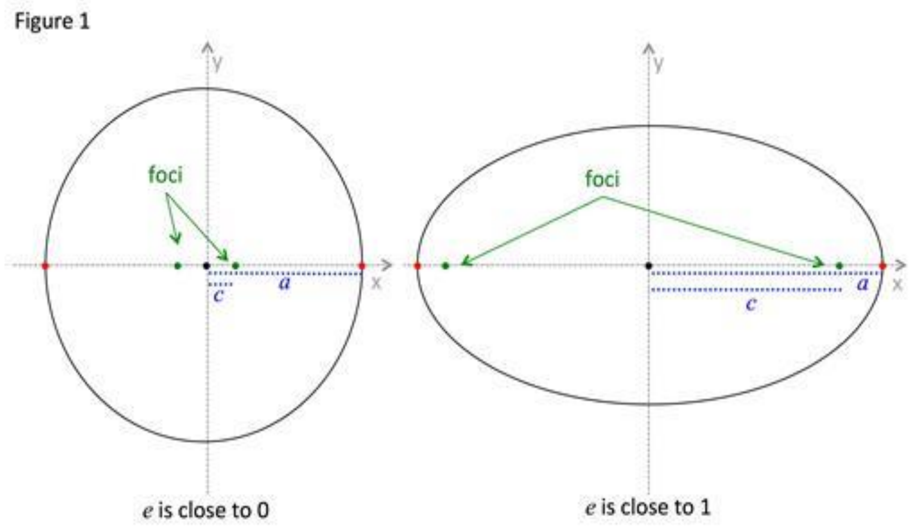
$$\text{angle} = \tan^{-1}\left(\frac{y_2 - y_1}{x_2 - x_1}\right)$$

(x_1,y_1) and (x_2,y_2) are the end points of the major-axis



Eccentricity

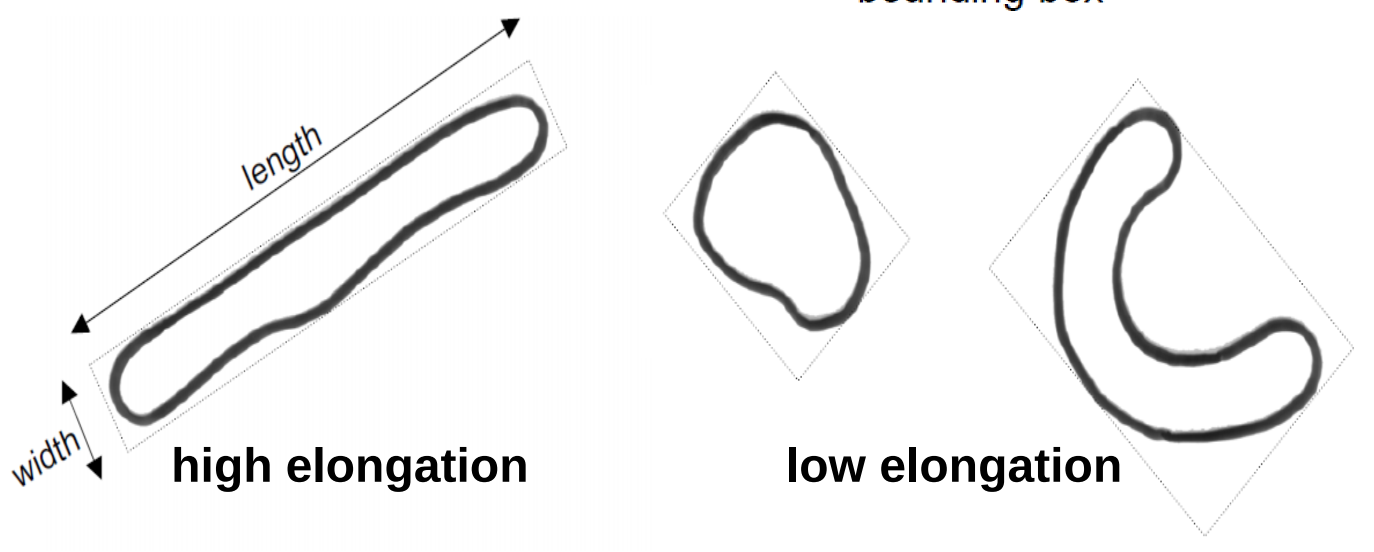
- Eccentricity of the ellipse that is fitted to the image. The eccentricity is the ratio of the distance between the foci of the ellipse and its major axis length.
  - The value is between 0 and 1. (0 and 1 are degenerate cases. An ellipse whose eccentricity is 0 is actually a circle, while an ellipse whose eccentricity is 1 is a line segment.)



Elongation

- Elongation is the ratio between the length and width of the object bounding box:

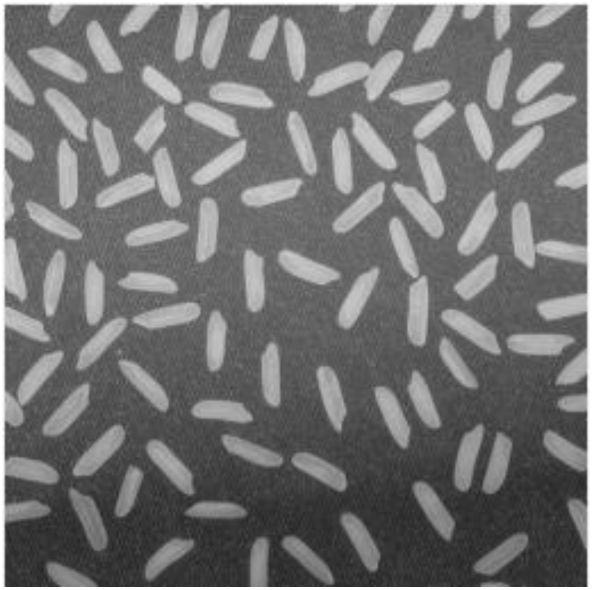
$$\text{elongation} = \frac{\text{width}_{\text{bounding-box}}}{\text{length}_{\text{bounding-box}}}$$



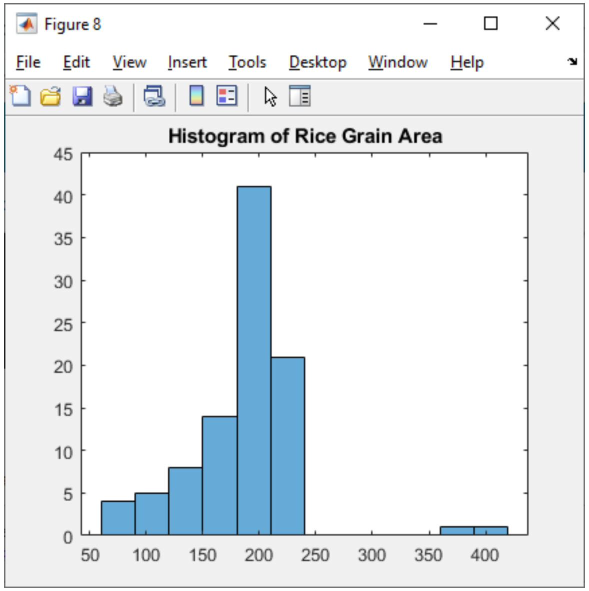
Ecersise 1

- Run the Matlab demo available on image analysis:

`https://it.mathworks.com/help/images/correcting-nonuniform-illumination.html`



Input



Output

Homework1

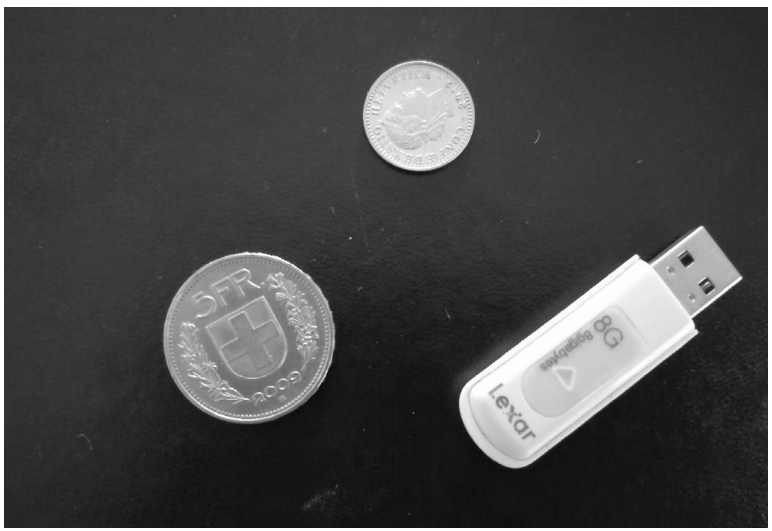
- Try to obtain the same results with coins image:

```
I = imread('eight.tif');
figure(1);
imshow(I);
```

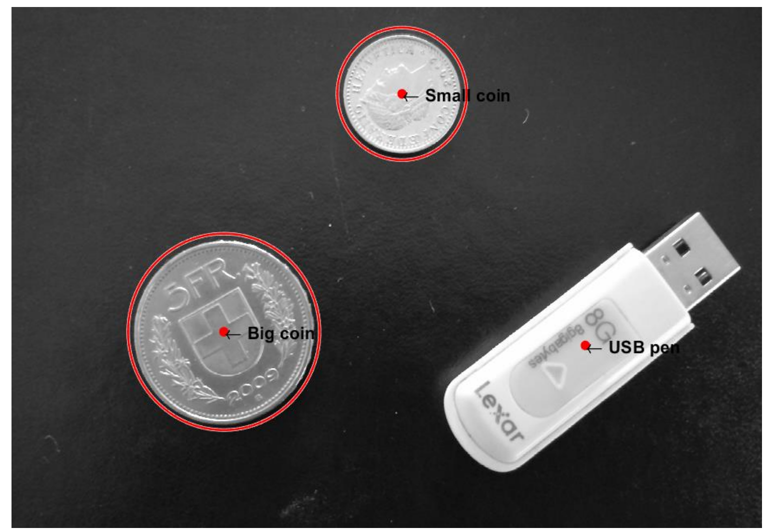
Suggestion: change the parameters and the combination of morphological operator to obtain a reliable binary image.

Homework2

- Use moprhological operators and region properties to infer the following information from this image:



Input image

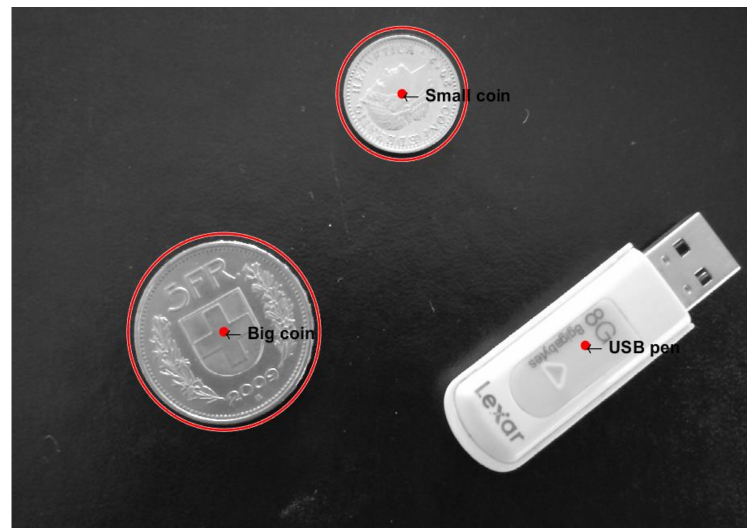


Output analysis

Homework3

- Use moprhological operators and region properties to infer the following information from a given image:

Suggestion: exploit the funcion '`regionprops`' to detect circular objects and compute the size of regions.

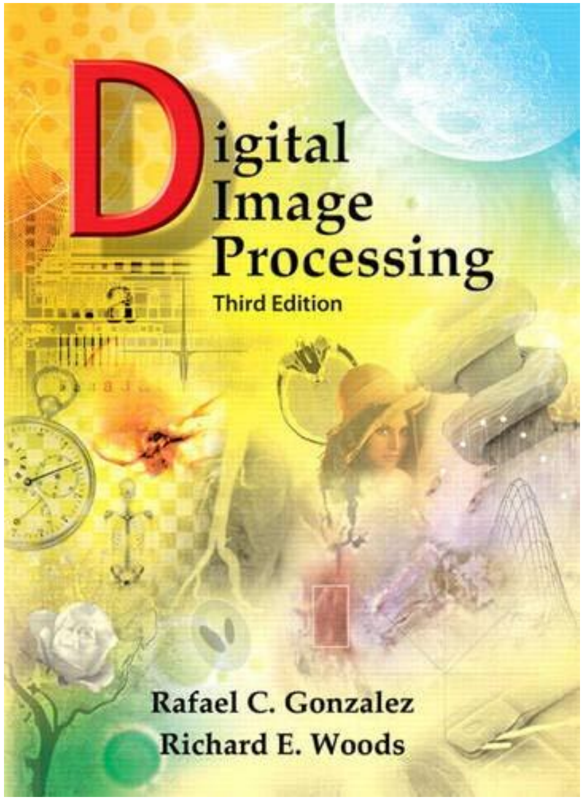


Output analysis

Homework4

- You decide the scene and take a picture of that. You should take object with different characteristics and implement the right combination of morhological operations and region propertis to recognize and localize them.

Refs.



- Gonzalez and Woods, «Digital Image Processing». Chapter 9.