Binary Image Analysis

Sedat OZER
Department of Computer Engineering
Bilkent University
sedat@cs.bilkent.edu.tr

Outline

- Introduction to binary image analysis
- Thresholding
- Mathematical morphology
- Pixels and neighborhoods
- Region Growing

Announcement

- Make sure to use Slack for group projects (discussions, ideas, finding group members, etc.)

- If you are sending an email about forming a group, make sure to cc all your group members in your email!

- TA: Furkan: furkan.huseyin@bilkent.edu.tr
- TA: Aziza: aziza.saber@bilkent.edu.tr

- Have an idea by Monday Oct. 12.
  - Send an email to me to check with me first.

Announcement

- TA: Furkan: furkan.huseyin@bilkent.edu.tr
- TA: Aziza: aziza.saber@bilkent.edu.tr

- Have **an idea** by Monday Oct. 12.
  - Send an email to me (**sedat@cs.bilkent.edu.tr**) to check your group members and project idea with me first. The subject of your email should be**: CS 484-555 Final Project Idea**
    - Include all your group members and their info (department, and whether they are undergrad or grad level students),
    - Talk about whether you found what data and code to start from,
    - Talk about your project idea in details that you plan to work on.

- Submit your final group members & **final idea** by Wednesday Oct. 14.

- Your first HW will be sent out next week.

- Familiarize yourself with Jupyter Notebook, Python and Colab.
- Conda is also useful to learn (especially for Deep learning related projects).
-

Sample Project Ideas

- Autonomous Path Planning with Deep Reinforcement Learning
- Face Detection for Android Applications with Deep learning
- Image inpainting with GANs
- Object detection with YOLO and Faster-RCNN
- Object/Actor segmentation with Deep Learning
- Object Tracking in Videos with Deep Learning
- Video generations with GANs
- Scene understanding /parsing with GANs
- Person re-identification with GANs
- Activity Recognition/Detection in videos with Deep Learning

- **I also posted more info on Slack.**

Binary image analysis

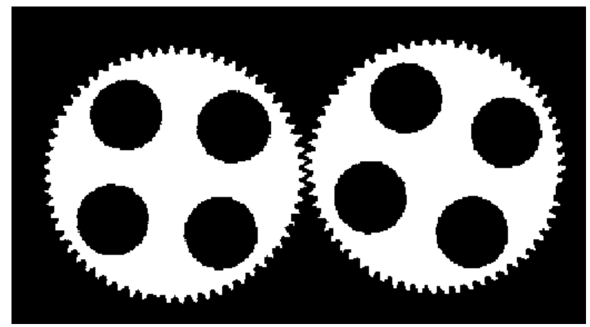
- Binary image analysis consists of a set of operations that are used to produce or process binary images, usually images of 0’s and 1’s where
  - 0 represents the background,
  - 1 represents the foreground.

```
000100100001000
000111100001000
000100100001000
```

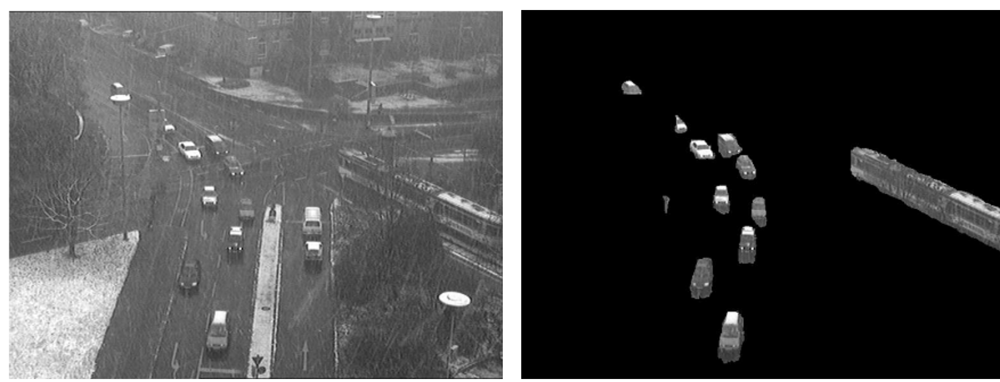
Some application areas

- Document analysis

- Industrial inspection

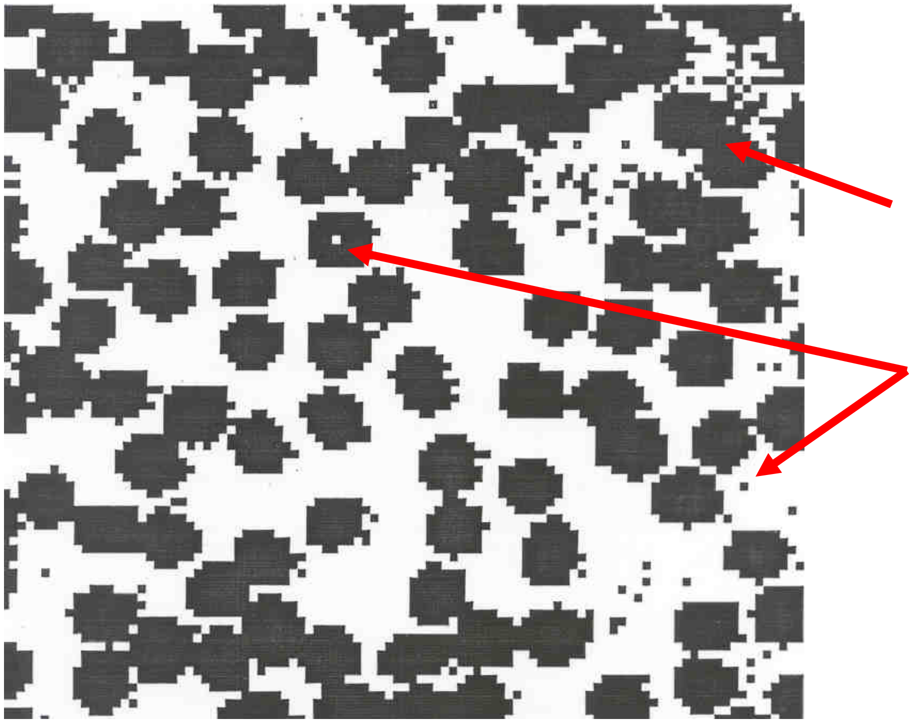


- Surveillance



Adapted from Shapiro and Stockman;
Cheung and Kamath

Example: red blood cell image



- Many blood cells are separate objects.
- Many touch each other → bad!
- Salt and pepper noise is present.
- How useful is this data?

- 63 separate objects are detected.
- Single cells have area of about 50 pixels.

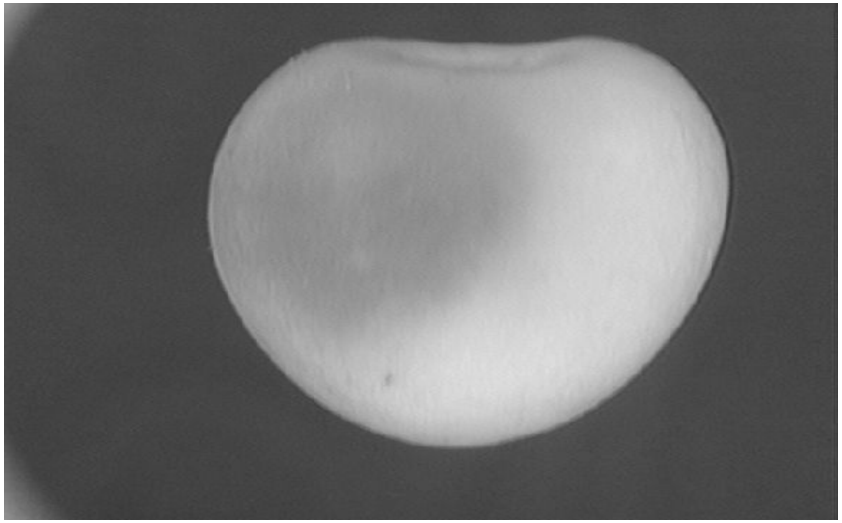
Adapted from Linda Shapiro, U of Washington

Thresholding

- Binary images can be obtained by thresholding.
- Assumptions for thresholding:
  - Object region of interest has intensity distribution different from background.
  - Object pixels likely to be identified by intensity alone:
    - intensity > a
    - intensity < b
    - a < intensity < b
- Works OK with flat-shaded scenes or engineered scenes.
- Does not work well with natural scenes.

Use of histograms for thresholding

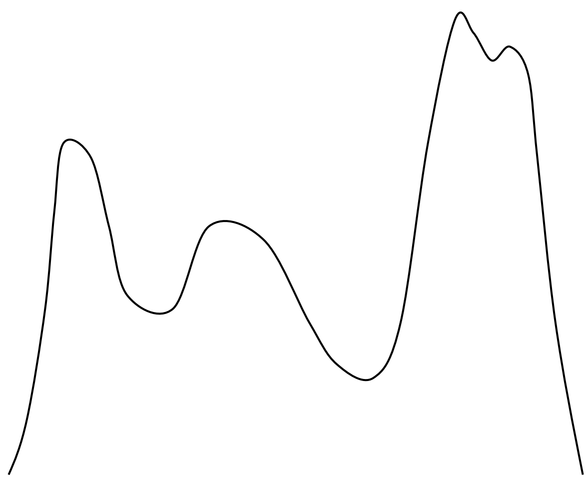
- Background is black.
- Healthy cherry is bright.
- Bruise is medium dark.
- Histogram shows two cherry regions (black background has been removed).



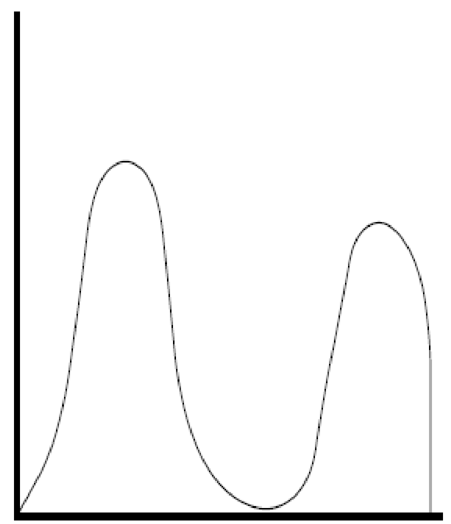
Adapted from Shapiro and Stockman

Automatic thresholding

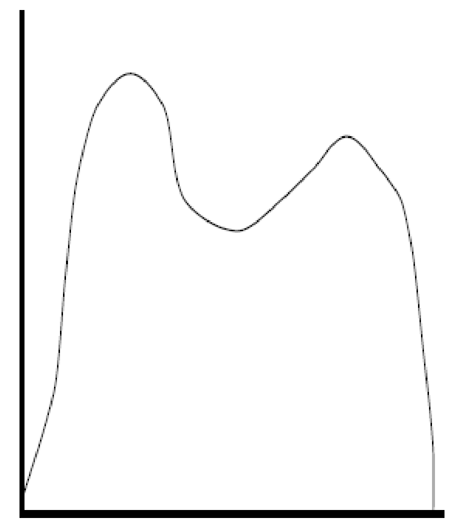
- How can we use a histogram to separate an image into 2 (or several) different regions?



Is there a single clear threshold? 2? 3?



Two distinct modes

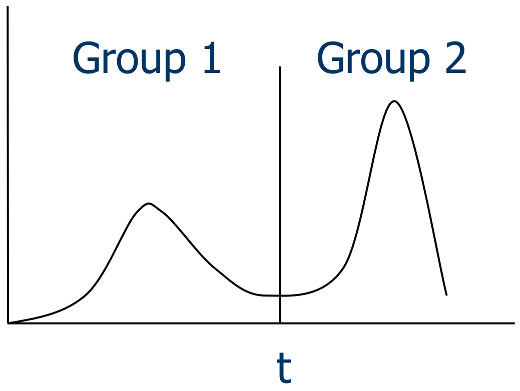


Overlapped modes

Adapted from Shapiro and Stockman

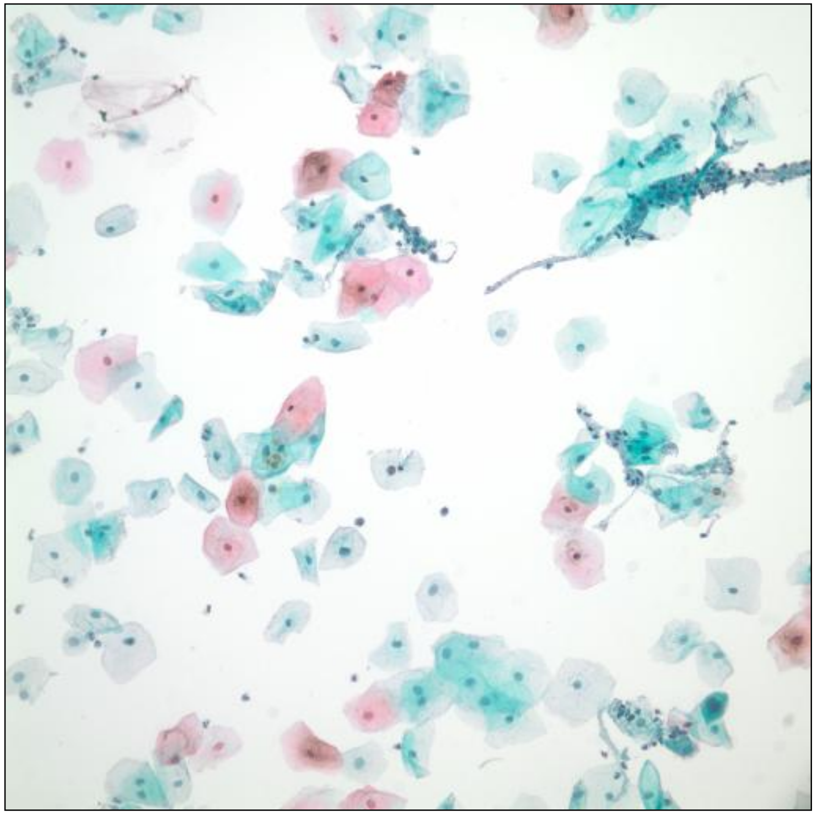
Automatic thresholding: Otsu's method

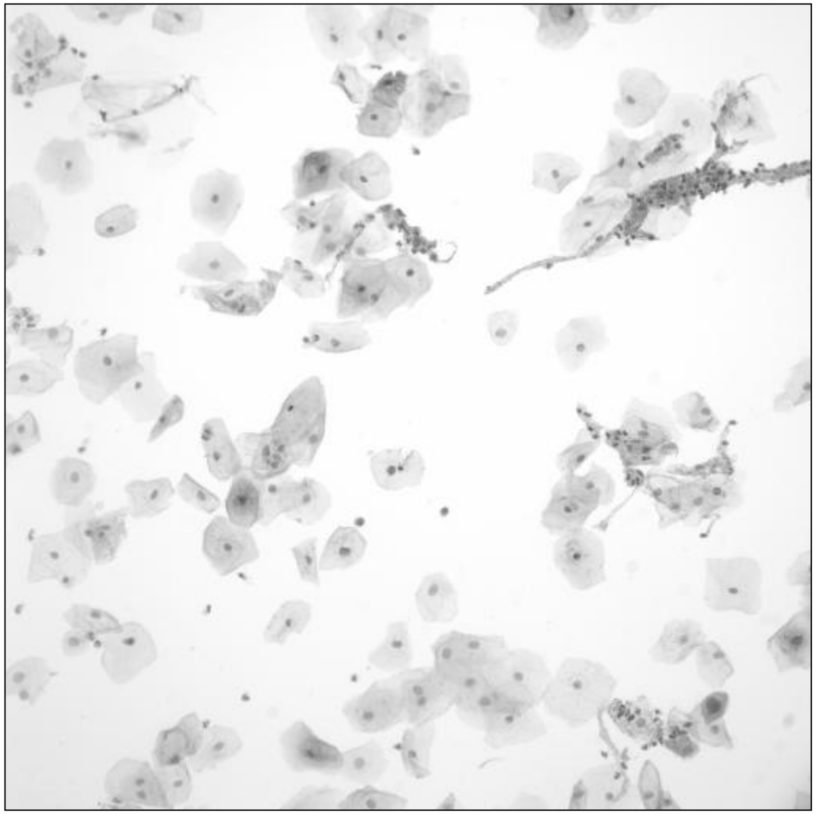
- Assumption: the histogram is bimodal.
- Method: find the threshold t that minimizes the <u>weighted sum of within-group variances</u> for the two groups that result from separating the gray levels at value t.
- The best threshold t can be determined by a simple sequential search through all possible values of t.



- If the gray levels are strongly dependent on the location within the image, local or dynamic thresholds can also be used.

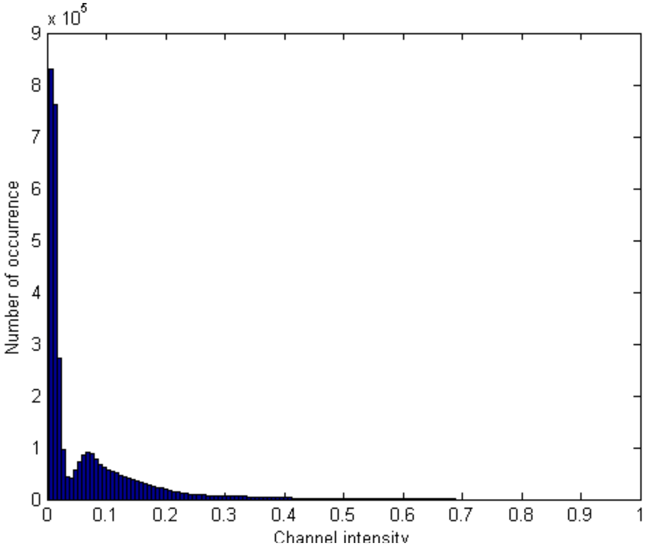
Automatic thresholding

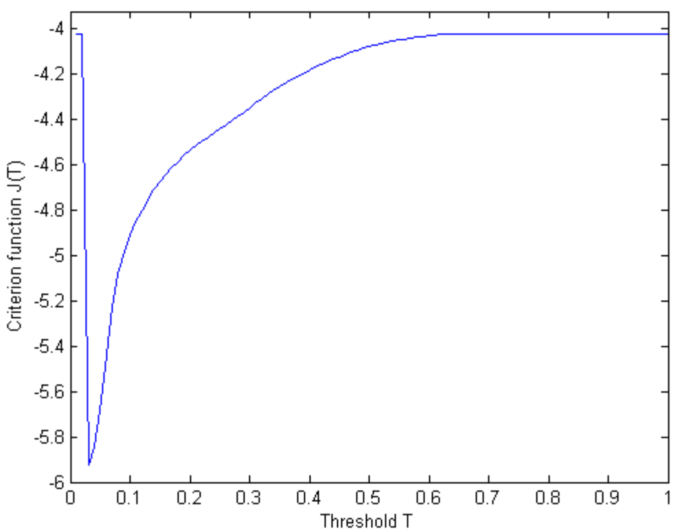


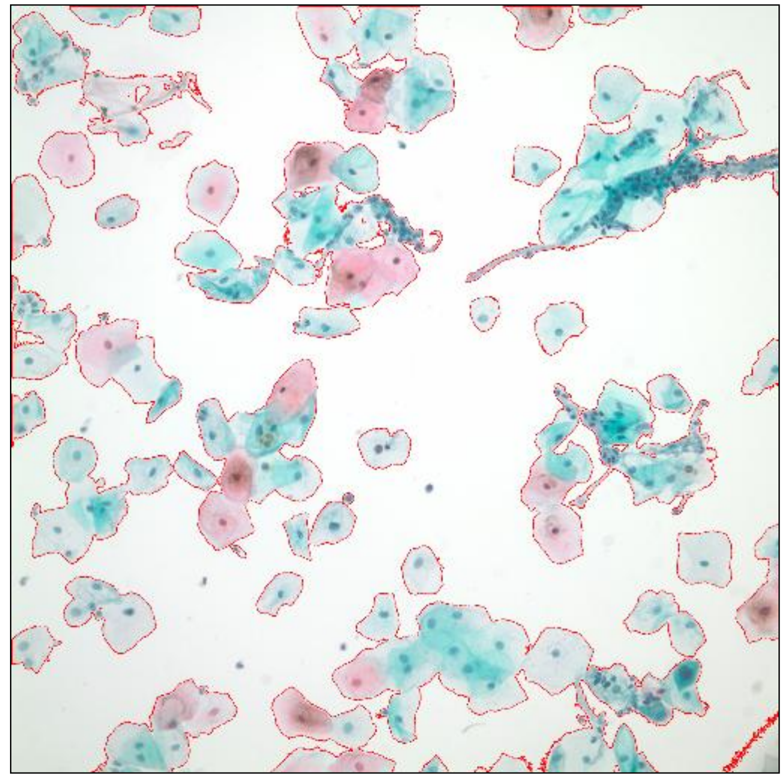


A Pap smear image example: RGB image (left) and grayscale image (right).

Automatic thresholding







Histogram of the negative image (top-left), sum of within-group variances versus the threshold (bottom-left), resulting mask overlayed as red on the original image (top).

Otsu's method

- http://www.labbookpages.co.uk/software/imgProc/otsuThreshold.html

- Wiki: https://en.wikipedia.org/wiki/Otsu%27s_method

Mathematical morphology

- The word morphology refers to form and structure.
- In computer vision/image processing, it is used to refer to the shape of a region.
- The language of mathematical morphology is set theory where sets represent objects in an image.
- We will discuss morphological operations on binary images whose components are sets in the 2D integer space Z^2.

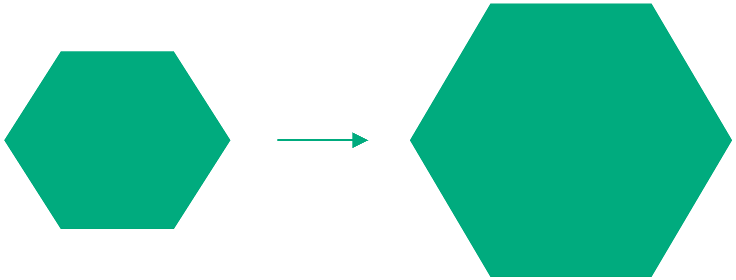
Mathematical morphology

- Mathematical morphology consists of two basic operations
  - dilation
  - erosion

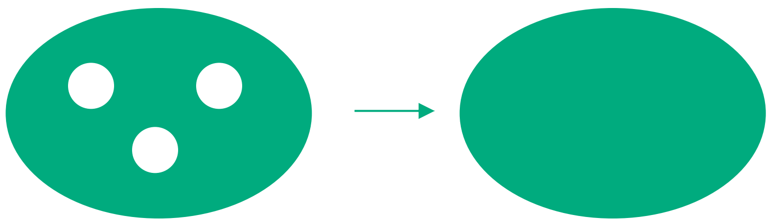
  and several composite relations
  - opening
  - closing
  - conditional dilation
  - …

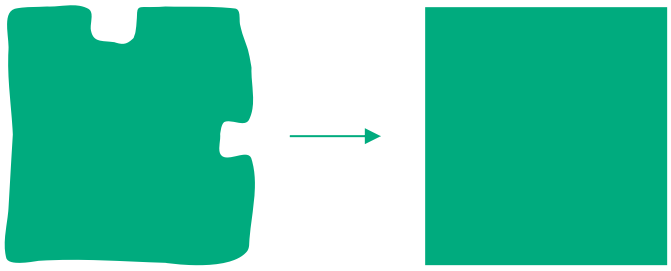
Dilation

- Dilation expands the connected sets of 1s of a binary image.
- It can be used for
  - growing features



  - filling holes and gaps

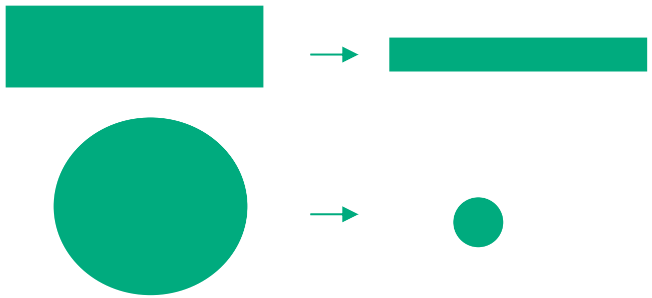




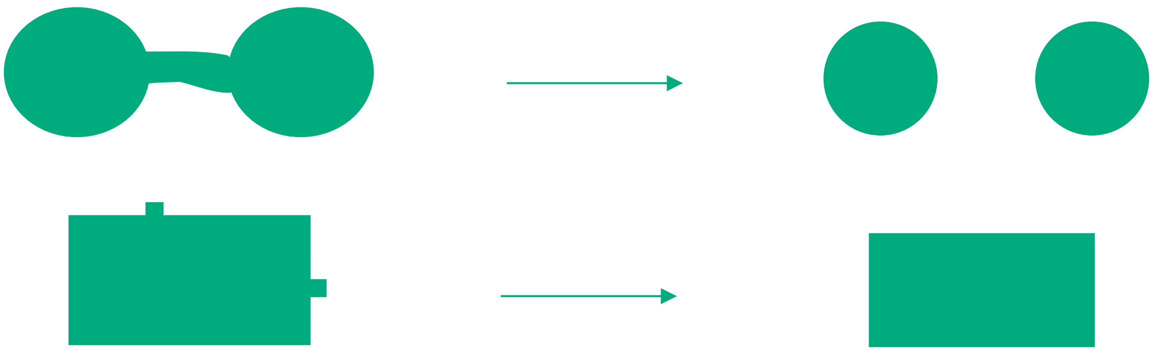
Adapted from Linda Shapiro, U of Washington

Erosion

- Erosion shrinks the connected sets of 1s of a binary image.
- It can be used for
  - shrinking features



  - removing bridges, branches and small protrusions



Adapted from Linda Shapiro, U of Washington

Basic concepts from set theory

- Let A be a set in Z^2. If $a = (a_1, a_2)$ is an element of A, we write $a \in A$; otherwise, we write $a \notin A$.
- Set A being a *subset* of set B is denoted by $A \subseteq B$.
- The *union* of two sets A and B is denoted by $A \cup B$.
- The *intersection* of two sets A and B is denoted by $A \cap B$.
- The *complement* of a set A is the set of elements not contained in A:

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

- The *difference* of two sets A and B, denoted by $A - B$, is defined as

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

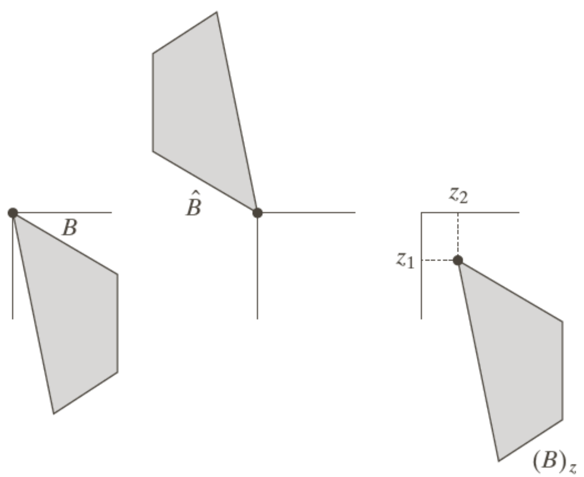
Basic concepts from set theory

- The *reflection* of set B, denoted by $\check{B}$, is defined as

$$\check{B} = \{w | w = -b, \forall b \in B\}.$$

- The *translation* of set A by point $z = (z_1, z_2)$, denoted by A_z, is defined as

$$A_z = \{w | w = a + z, \forall a \in A\}.$$



a b c

FIGURE 9.1 (a) A set, (b) its reflection, and (c) its translation by z.

Adapted from Gonzales and Woods

Structuring elements

- Structuring elements are small binary images used as shape masks in basic morphological operations.
- They can be any shape and size that is digitally representable.
- One pixel of the structuring element is denoted as its origin.
- Origin is often the central pixel of a symmetric structuring element but may in principle be any chosen pixel.

Structuring elements

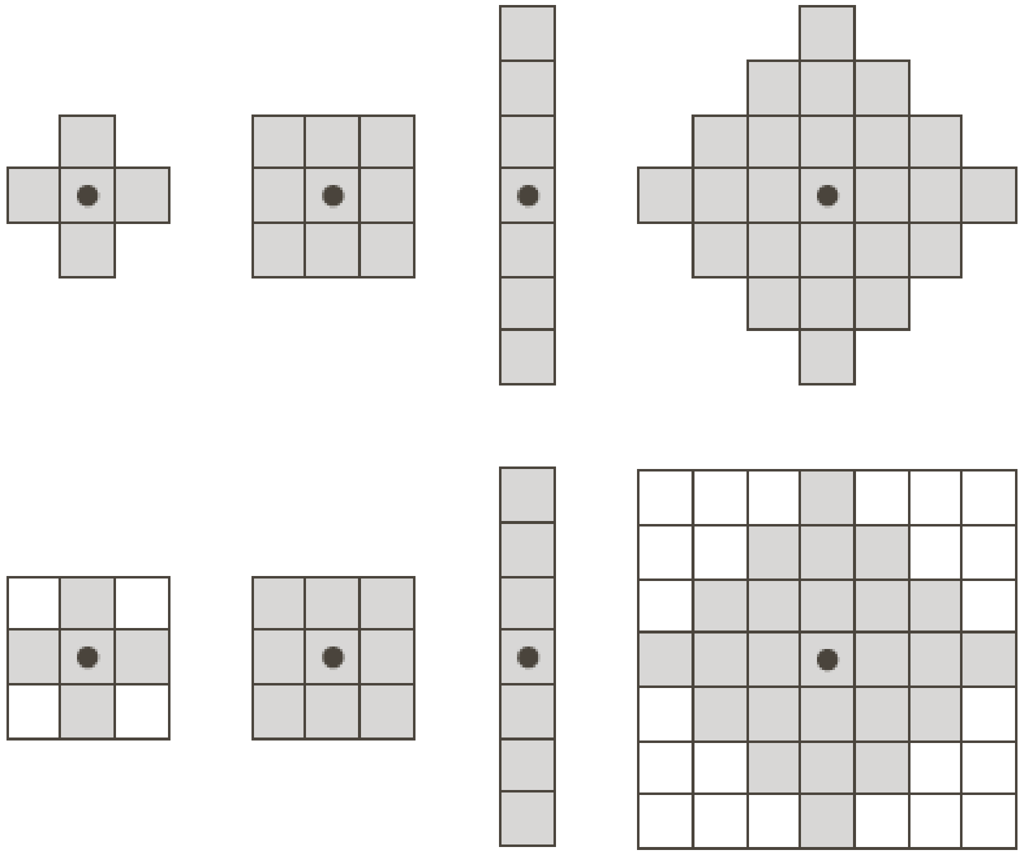


FIGURE 9.2 First row: Examples of structuring elements. Second row: Structuring elements converted to rectangular arrays. The dots denote the centers of the SEs.

Adapted from Gonzales and Woods

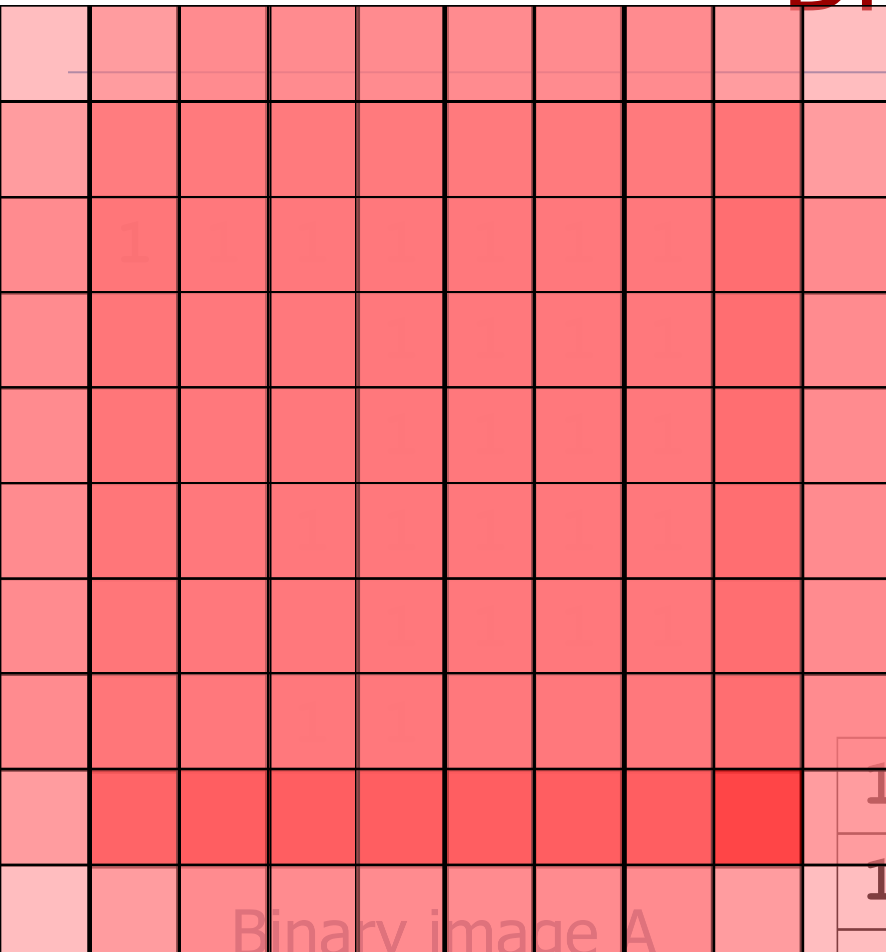
Dilation

- The *dilation* of binary image A by structuring element B is denoted by $A \oplus B$ and is defined by

$$A \oplus B = \{z | \check{B}_z \cap A \neq \emptyset\},$$
$$= \bigcup_{a \in A} B_a.$$

  - First definition: The dilation is the set of all displacements z such that $\check{B}_z$ and A overlap by at least one element.
  - Second definition: The structuring element is swept over the image. Each time the origin of the structuring element touches a binary 1-pixel, the entire translated structuring element is ORed to the output image, which was initialized to all zeros.

Dilation



Binary image A

1	1	1
1	1	1
1	1	1

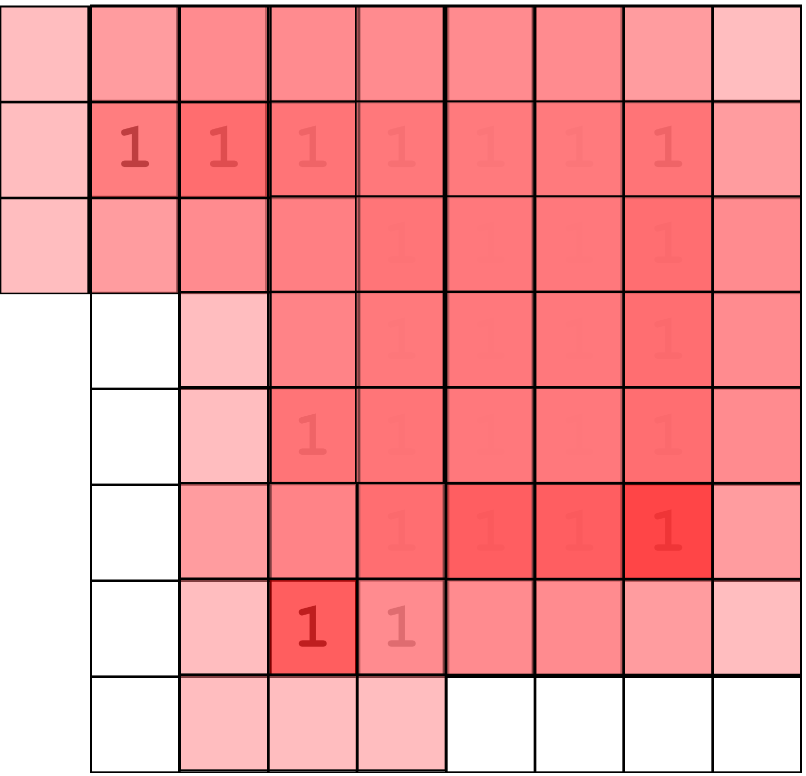
Structuring element B

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

Dilation result

(1st definition)

Dilation



Binary image A

1	1	1
1	1	1
1	1	1

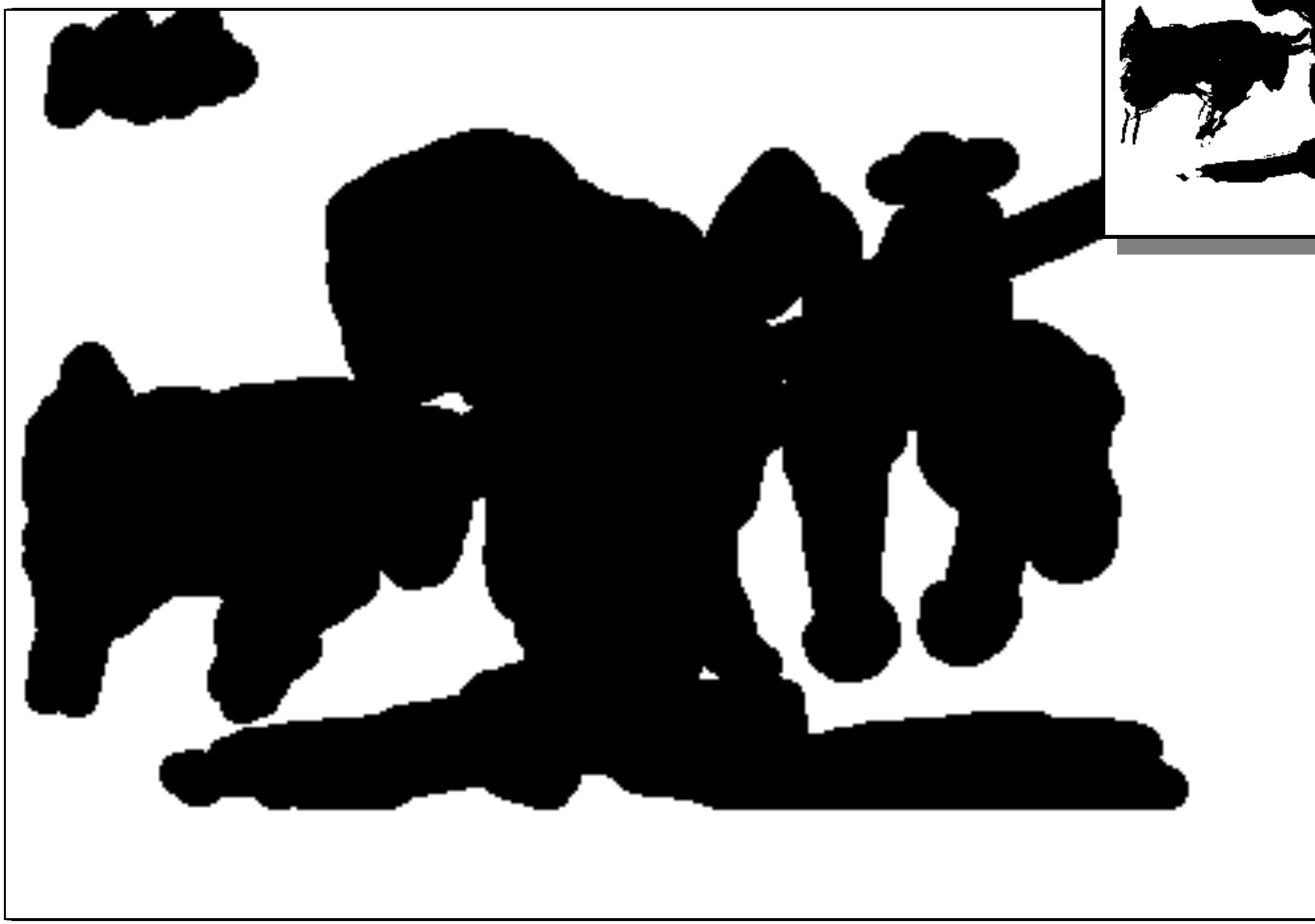
Structuring element B

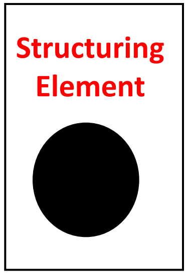
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
	1	1	1	1	1	1	1
	1	1	1	1	1	1	1
	1	1	1	1	1	1	1
	1	1	1	1	1	1	1
	1	1	1	1			

Dilation result

(2nd definition)

Dilation

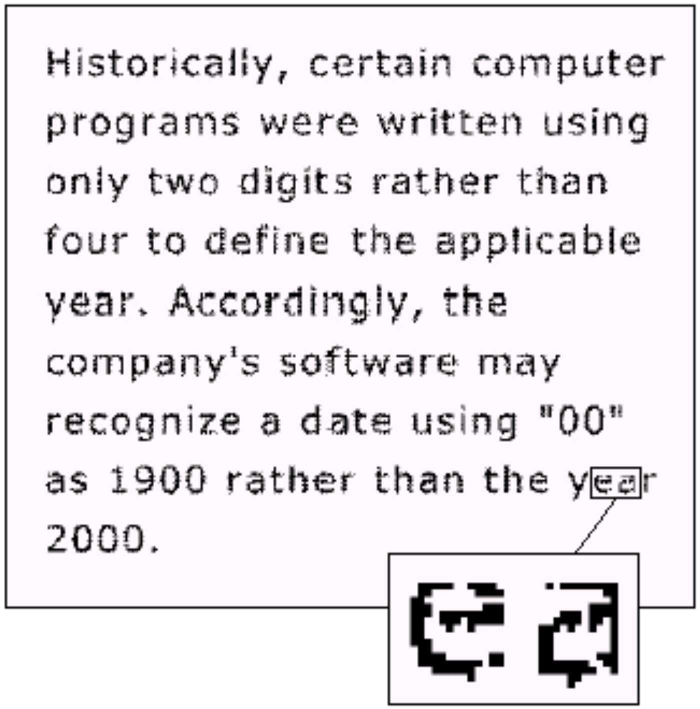


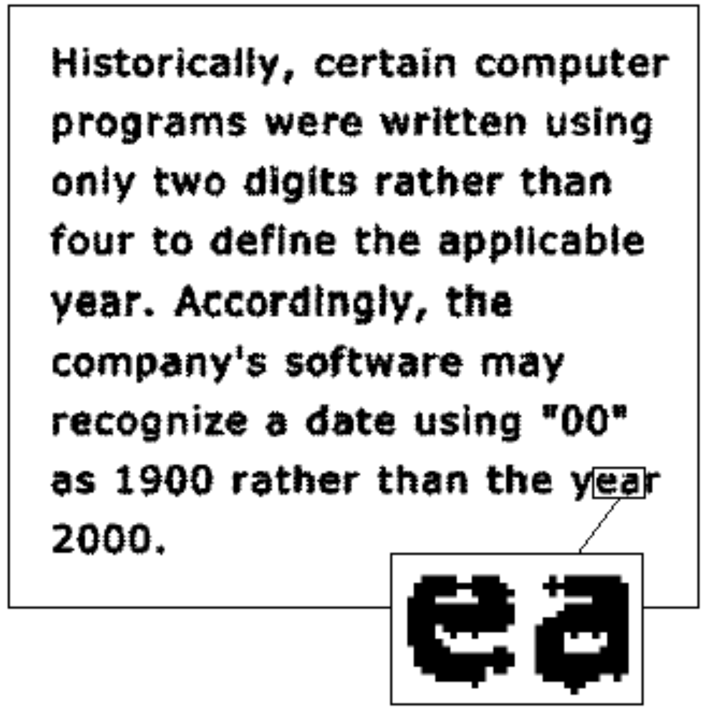


Adapted from John Goutsias, Johns Hopkins Univ.

Pablo Picasso, *Pass with the Cape*, 1960

Dilation





0	1	0
1	1	1
0	1	0

a c
b

FIGURE 9.5
(a) Sample text of poor resolution with broken characters (magnified view).
(b) Structuring element.
(c) Dilation of (a) by (b). Broken segments were joined.

Adapted from Gonzales and Woods

Erosion

- The *erosion* of binary image A by structuring element B is denoted by $A \ominus B$ and is defined by

$$A \ominus B = \{z | B_z \subseteq A\},$$
$$= \{a | a + b \in A, \forall b \in B\}.$$

  - First definition: The erosion is the set of all points z such that B, translated by z, is contained in A.
  - Second definition: The structuring element is swept over the image. At each position where every 1-pixel of the structuring element covers a 1-pixel of the binary image, the binary image pixel corresponding to the origin of the structuring element is ORed to the output image.

Erosion

1	1	1
1	1	1
1	1	1

Binary image A

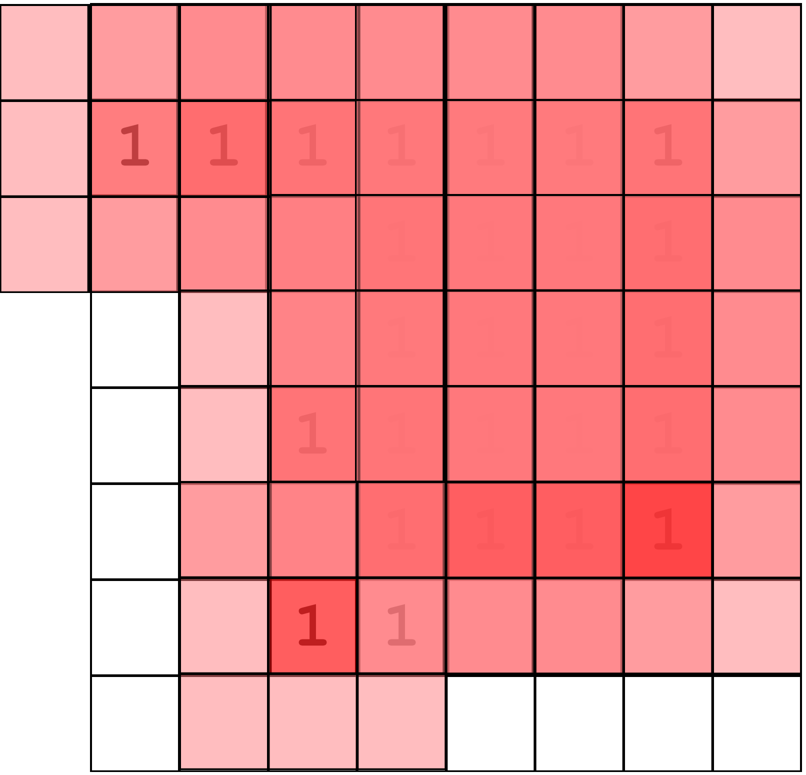
Structuring element B

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	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

Erosion result

(1st definition)

Erosion



Binary image A

1	1	1
1	1	1
1	1	1

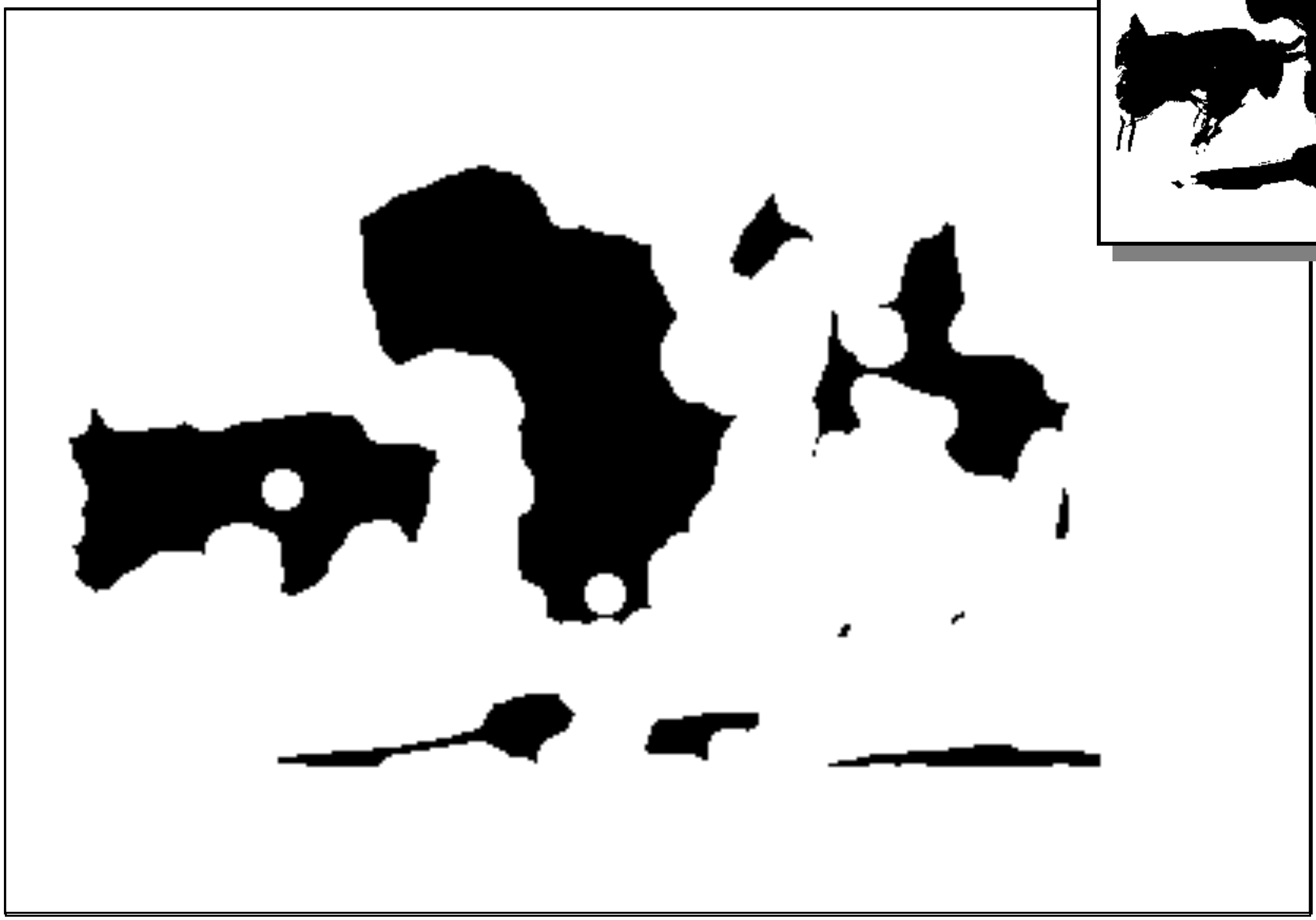
Structuring element B

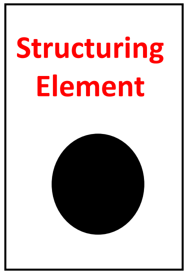
				1	1		
				1	1		
				1	1		

Erosion result

(2nd definition)

Erosion





Adapted from John Goutsias, Johns Hopkins Univ.

Pablo Picasso, *Pass with the Cape*, 1960

Erosion

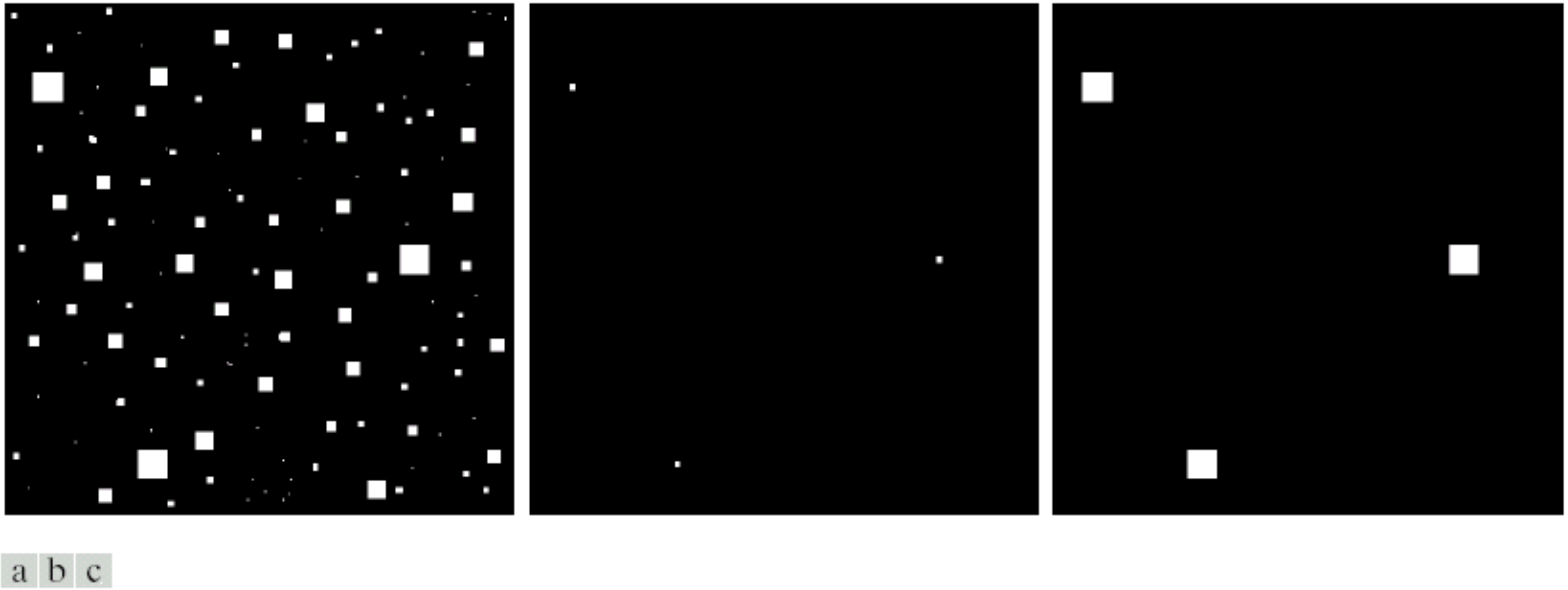


FIGURE 9.7 (a) Image of squares of size 1, 3, 5, 7, 9, and 15 pixels on the side. (b) Erosion of (a) with a square structuring element of 1's, 13 pixels on the side. (c) Dilation of (b) with the same structuring element.

Adapted from Gonzales and Woods

Opening

- The *opening* of binary image A by structuring element B is denoted by $A \circ B$ and is defined by

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

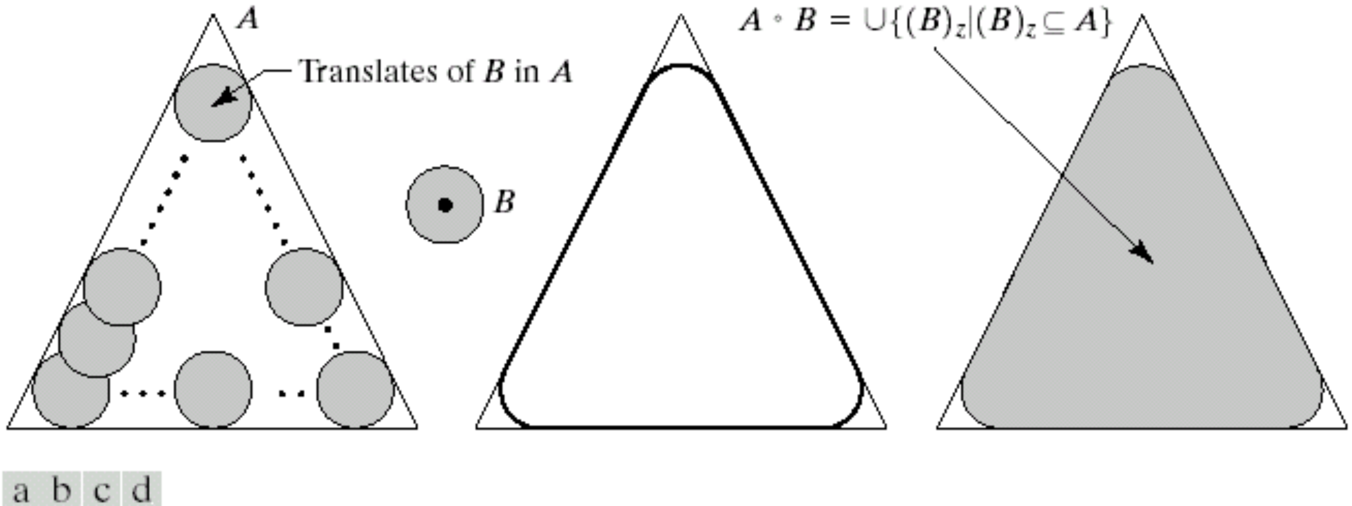


FIGURE 9.8 (a) Structuring element B "rolling" along the inner boundary of A (the dot indicates the origin of B). (c) The heavy line is the outer boundary of the opening. (d) Complete opening (shaded).

Opening

1	1	1	1	1	1	1	
			1	1	1	1	
			1	1	1	1	
		1	1	1	1	1	
			1	1	1	1	
		1	1				

Binary image A

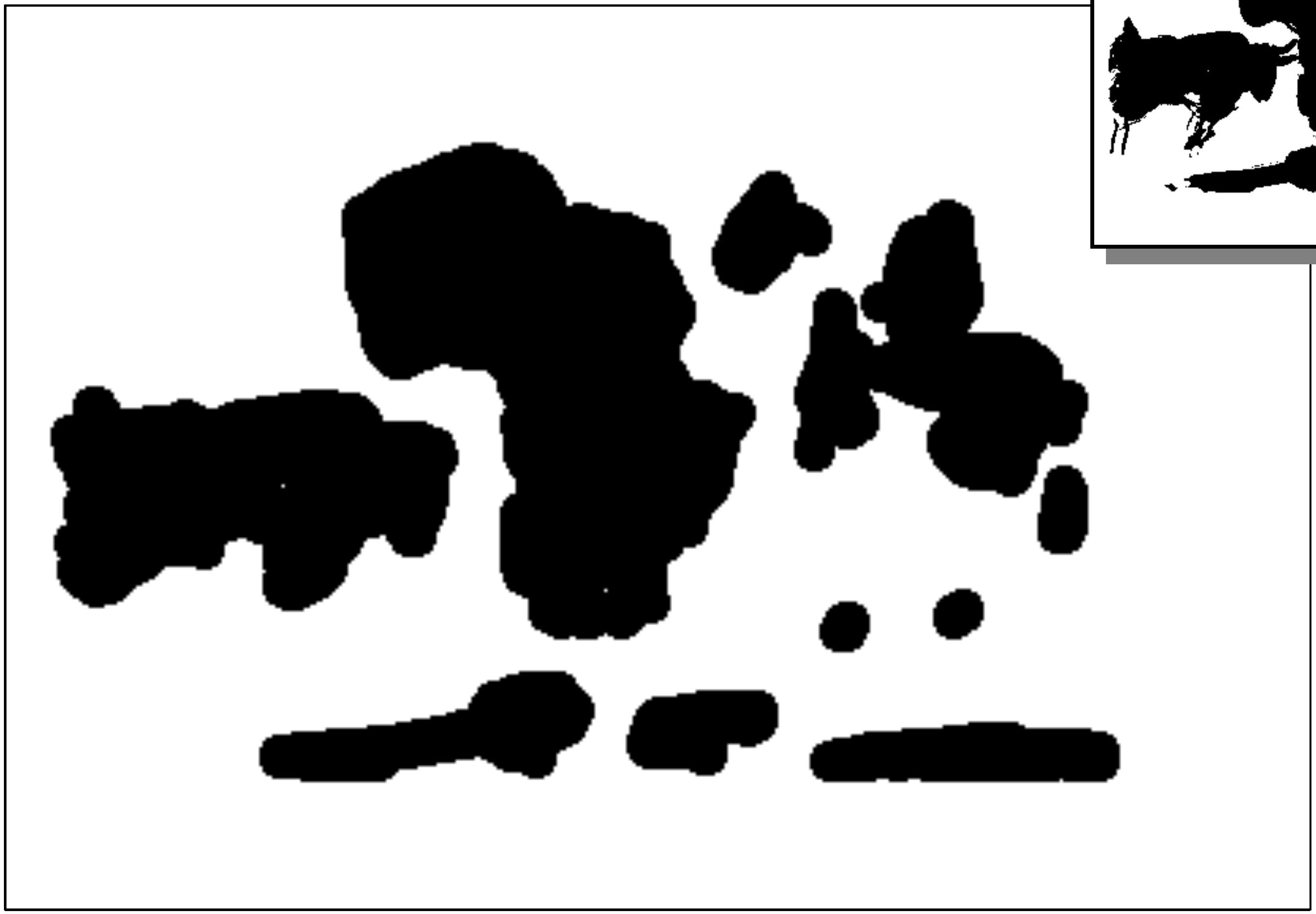
1	1	1
1	1	1
1	1	1

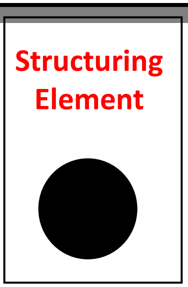
Structuring element B

			1	1	1	1	
			1	1	1	1	
			1	1	1	1	
			1	1	1	1	
			1	1	1	1	

Opening result

Opening





Adapted from John Goutsias, Johns Hopkins Univ.

Pablo Picasso, *Pass with the Cape*, 1960

Closing

- The *closing* of binary image A by structuring element B is denoted by $A \bullet B$ and is defined by

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

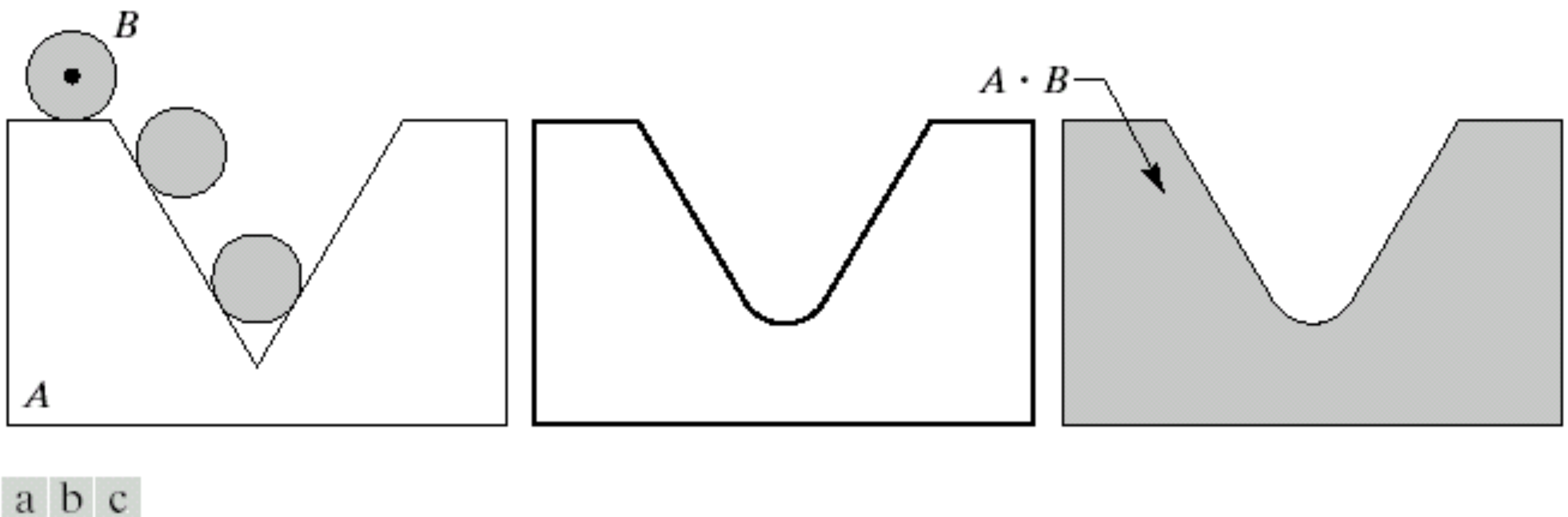


FIGURE 9.9 (a) Structuring element B "rolling" on the outer boundary of set A. (b) Heavy line is the outer boundary of the closing. (c) Complete closing (shaded).

Closing

1	1	1	1	1	1	1	
			1	1	1	1	
			1	1	1	1	
		1	1	1	1	1	
			1	1	1	1	
		1	1				

Binary image A

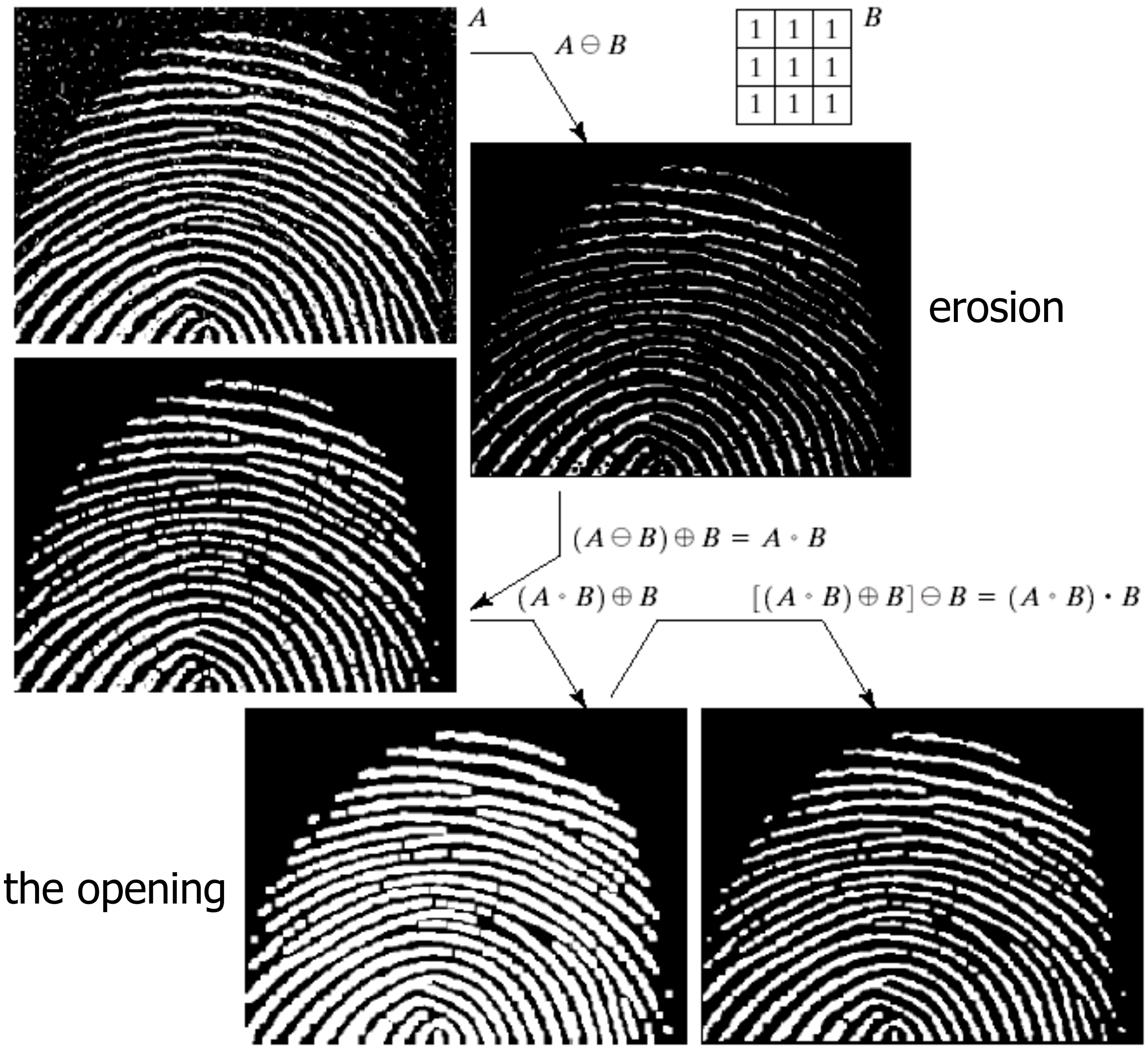
1	1	1
1	1	1
1	1	1

Structuring element B

1	1	1	1	1	1	1	
		1	1	1	1	1	
		1	1	1	1	1	
		1	1	1	1	1	
		1	1	1	1	1	
		1	1				

Closing result

Examples



erosion

Opening of A

Dilation of the opening

Closing of the opening

a b c d e f

FIGURE 9.11 (a) Noisy image. (c) Eroded image. (d) Opening of A. (d) Dilation of the opening. (e) Closing of the opening. (Original image for this example courtesy of the National Institute of Standards and Technology.)

Adapted from Gonzales and Woods

Properties

- Dilation and erosion are duals of each other with respect to set complementation and reflection, i.e.,

$$(A \ominus B)^c = A^c \oplus \check{B}.$$

- Opening and closing are duals of each other with respect to set complementation and reflection, i.e.,

$$(A \bullet B)^c = A^c \circ \check{B}.$$

Properties

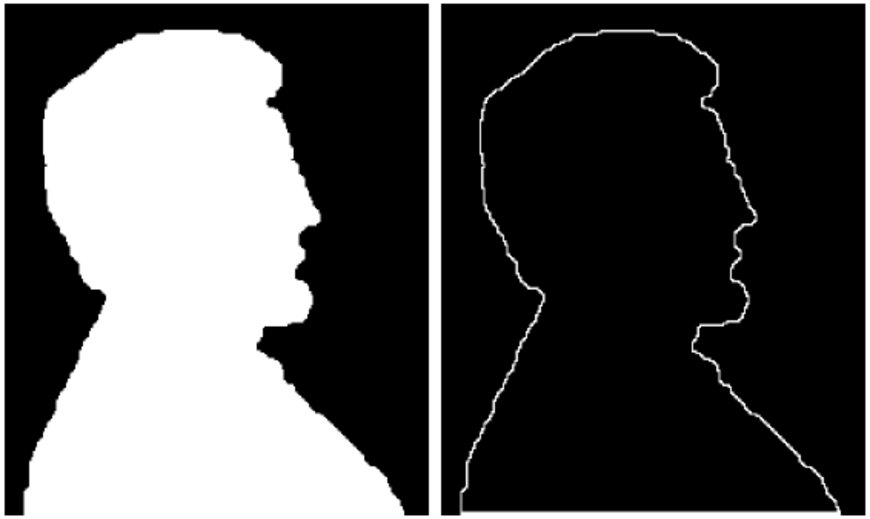
- Opening satisfies the following properties:
  - $A \circ B$ is a subset of A.
  - If C is a subset of D, then $C \circ B$ is a subset of $D \circ B$.
  - $(A \circ B) \circ B = A \circ B$.
- Closing satisfies the following properties:
  - A is a subset of $A \bullet B$.
  - If C is a subset of D, then $C \bullet B$ is a subset of $D \bullet B$.
  - $(A \bullet B) \bullet B = A \bullet B$.

Boundary extraction

- The *boundary* of a set A can be obtained by first eroding A by B and then performing the set difference between A and its erosion, i.e.,

$$\text{boundary}(A) = A - (A \ominus B)$$

where B is a suitable structuring element.



a b

FIGURE 9.14 (a) A simple binary image, with 1's represented in white. (b) Result of using Eq. (9.5-1) with the structuring element in Fig. 9.13(b).

Conditional dilation

- Given an original binary image B, a processed binary image C, and a structuring element A, let $C_0 = C$ and $C_n = (C_{n-1} \oplus A) \cap B$. The *conditional dilation* of C by A with respect to B is defined by

$$C \oplus |_B A = C_m$$

where the index m is the smallest index satisfying $C_m = C_{m-1}$.

- Given a structuring element that when applied to a binary image
  - removes the components that do not satisfy certain shape and size constraints, and
  - leaves a few 1-pixels of those components that do satisfy the constraints,

conditional dilation recovers the latter components using what remains of them after the erosion.

Region filling

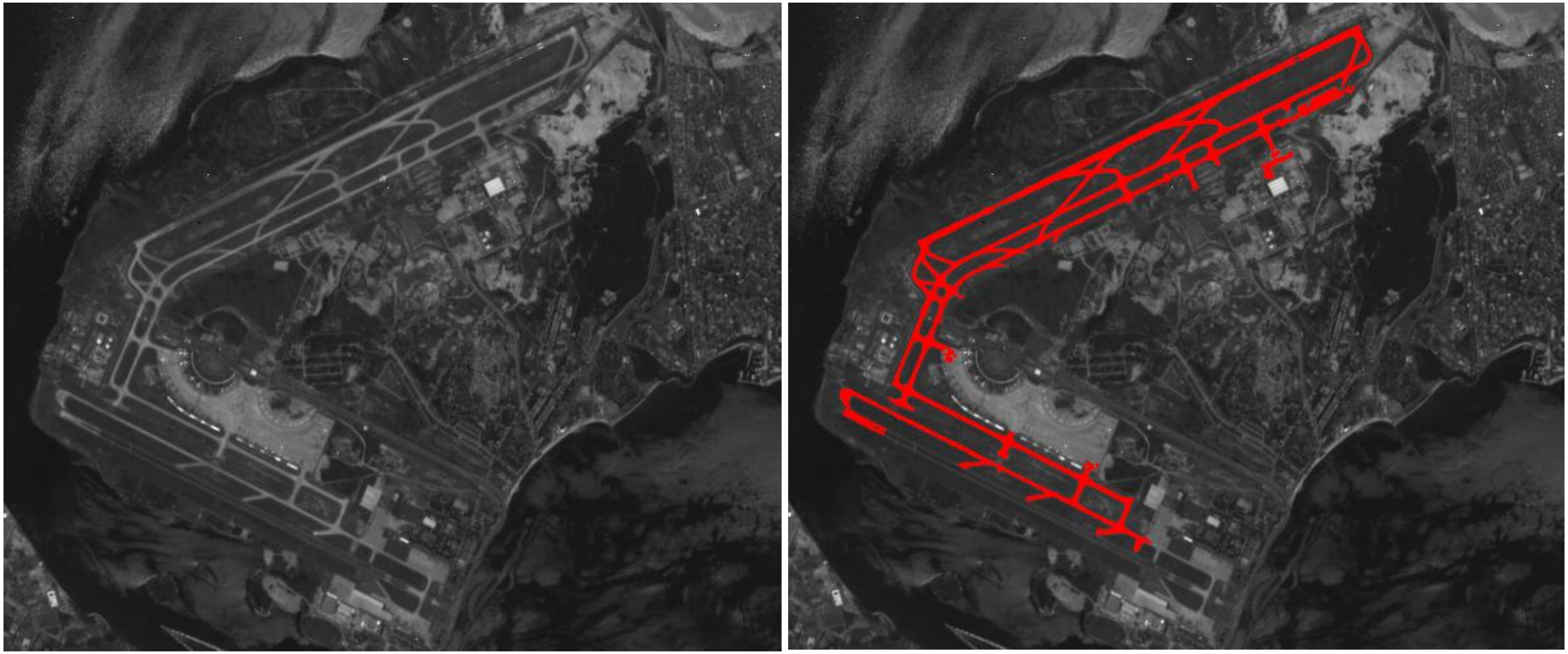
- Given set A containing the boundary points of a region, and a point p inside the boundary, the following procedure *fills the region* with 1's:

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

where $X_0 = p$ and B is the cross structuring element. The procedure terminates at iteration step k if $X_k = X_{k-1}$.

- The set union of X_k and A contains the filled set and its boundary.

Examples

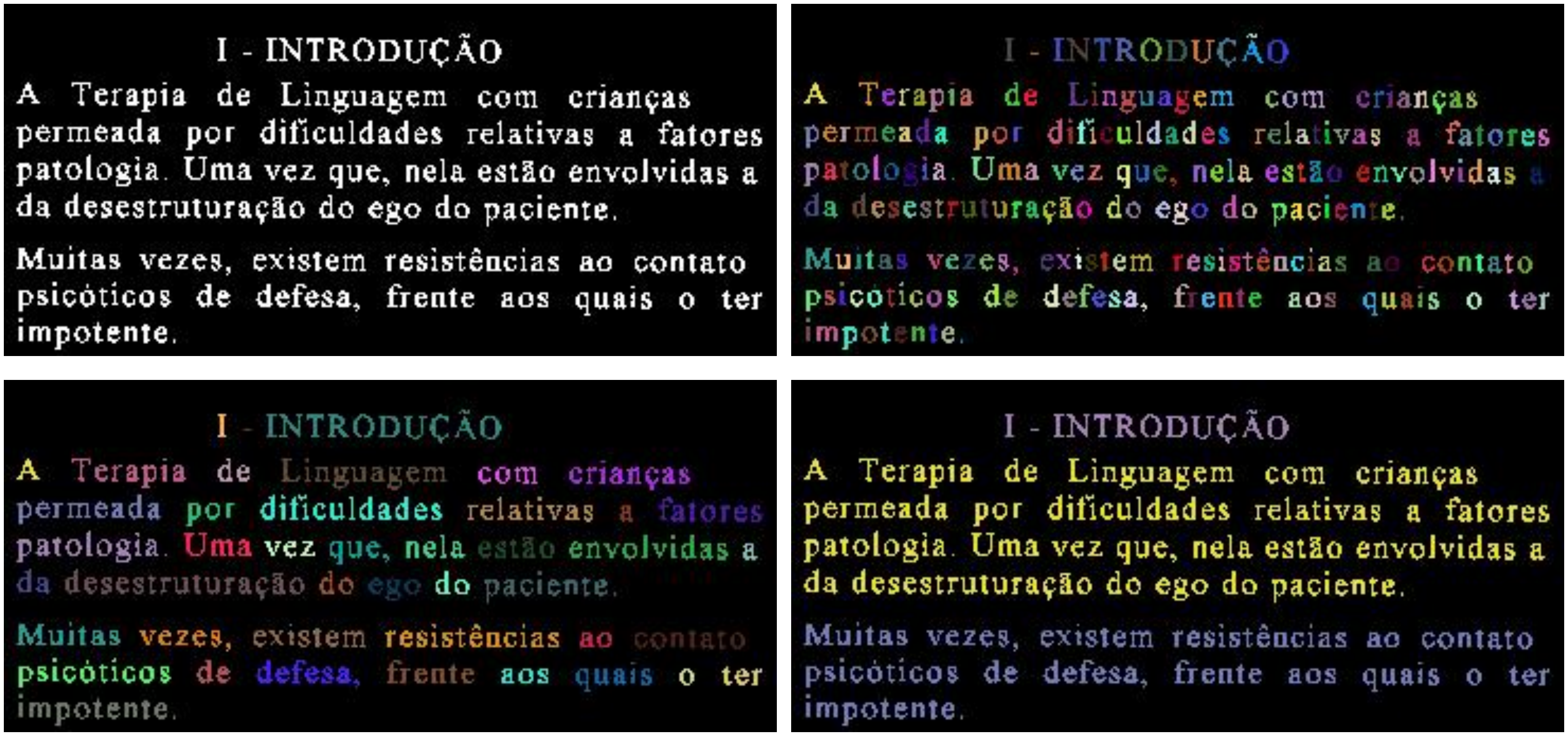


Detecting runways in satellite airport imagery

http://www.mmorph.com/mxmorph/html/mmdemos/mmdairport.html

Note: These links do not work anymore, but you can use the Wayback Machine for older versions.

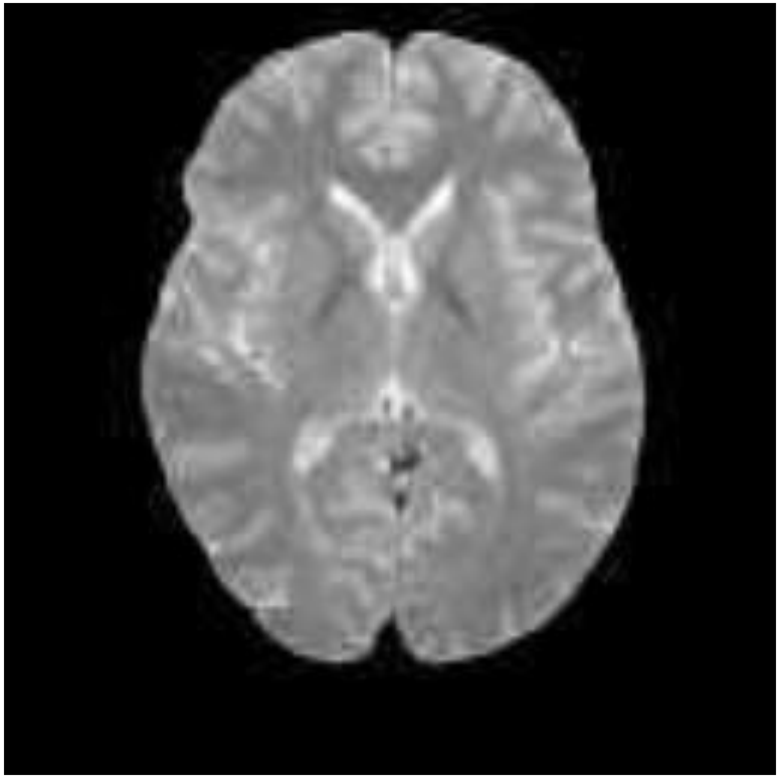
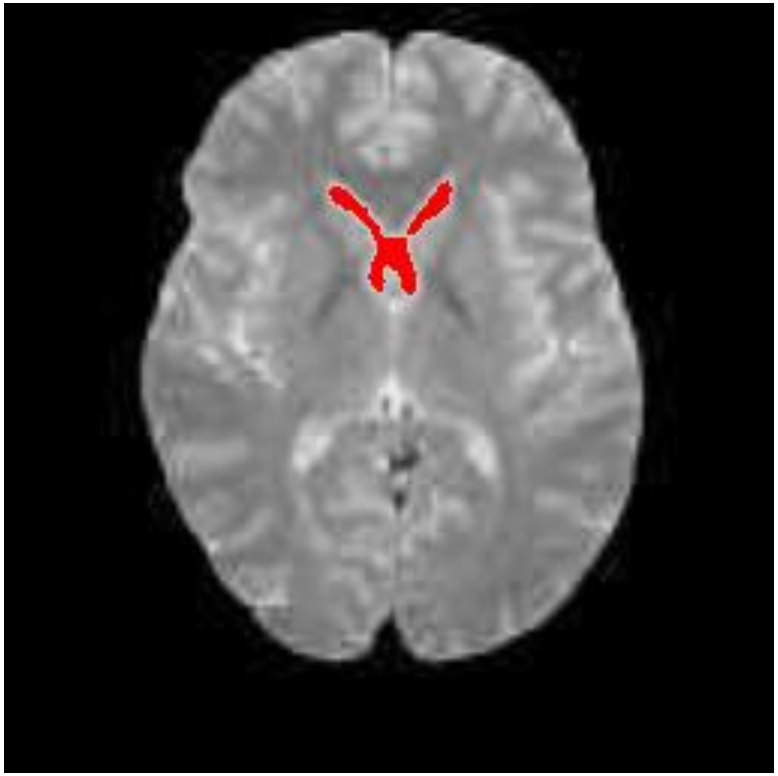
Examples



Segmenting letters, words and paragraphs

http://www.mmorph.com/mxmorph/html/mmdemos/mmdlabeltext.html

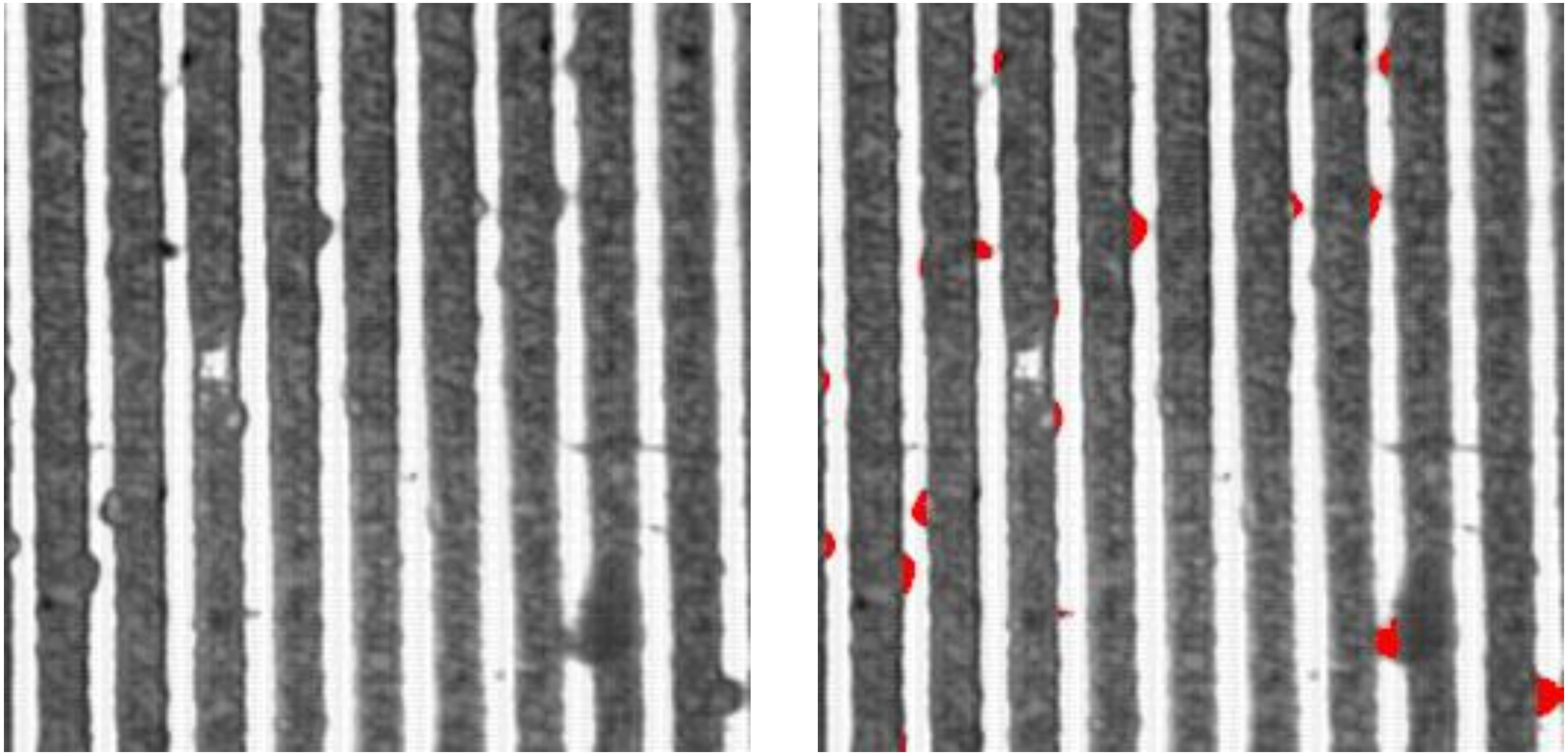
Examples

Extracting the lateral ventricle from an MRI image of the brain

http://www.mmorph.com/mxmorph/html/mmdemos/mmdbrain.html

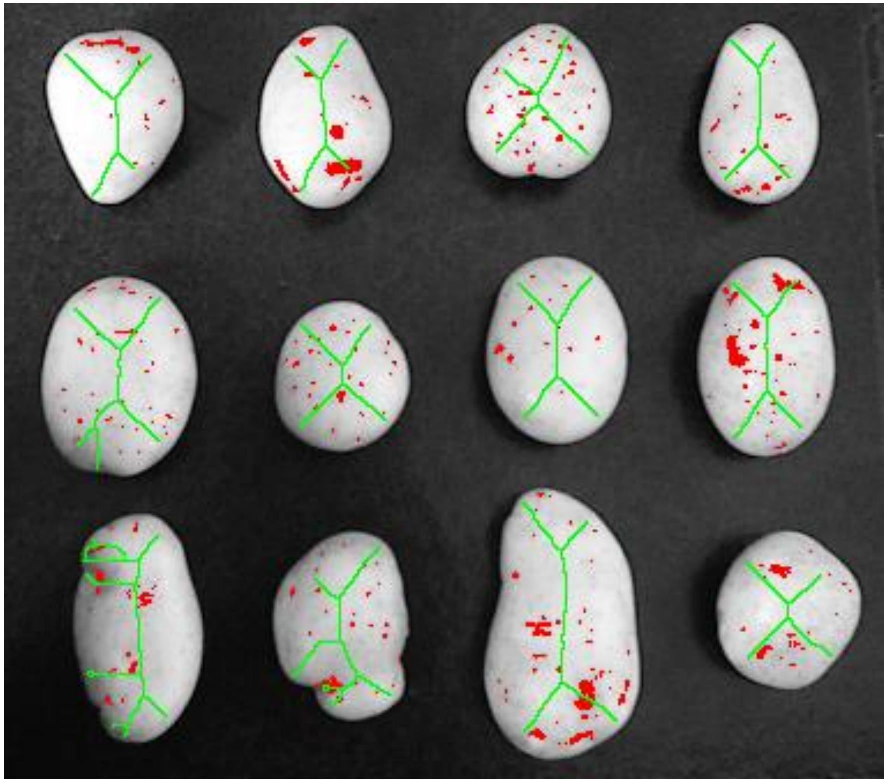
Examples



Detecting defects in a microelectronic circuit

http://www.mmorph.com/mxmorph/html/mmdemos/mmdlith.html

Examples



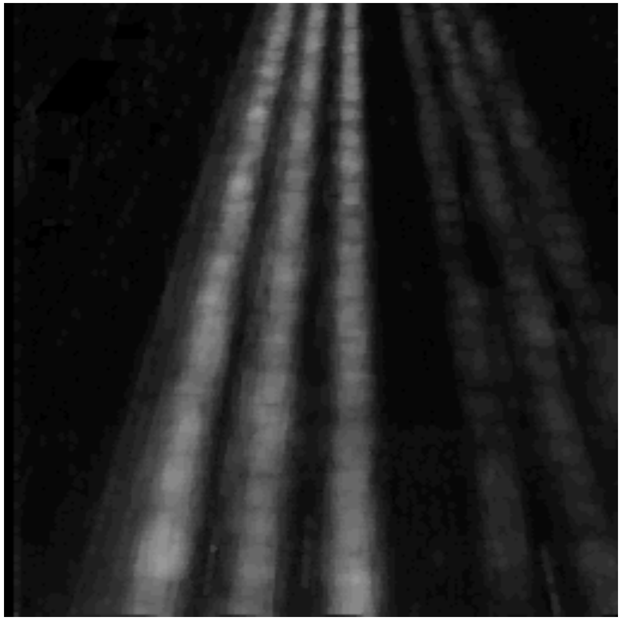
Grading potato quality by shape and skin spots

http://www.mmorph.com/mxmorph/html/mmdemos/mmdpotatoes.html

Examples

Traffic scene

Temporal average

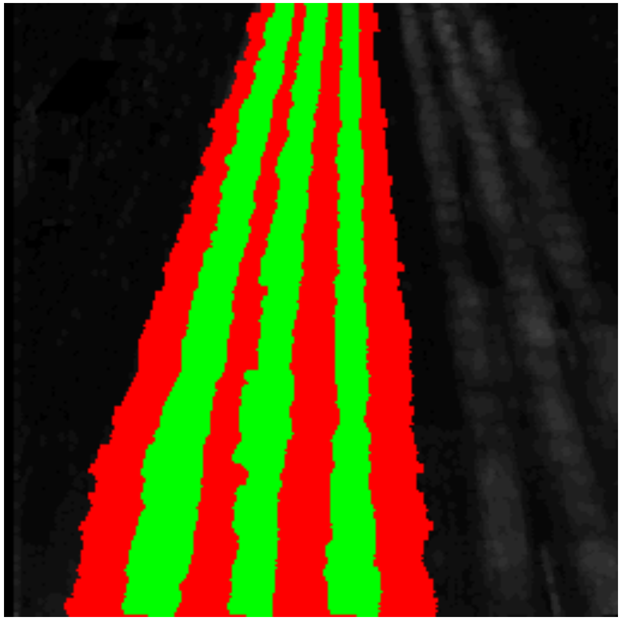


Average of differences

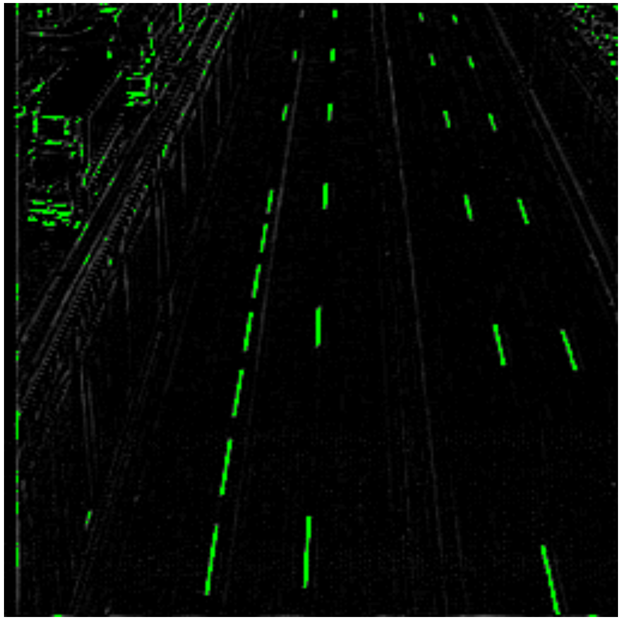
Lane detection example

Adapted from CMM/ENSMP/ARMINES

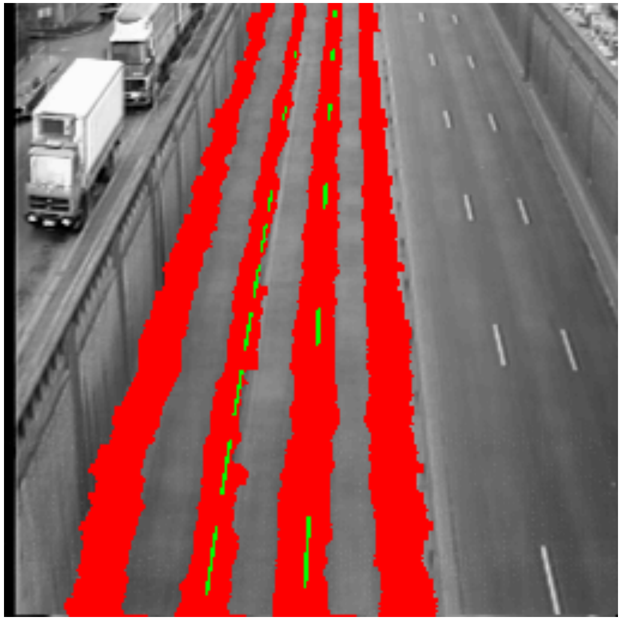
Examples



Threshold and dilation to detect lane markers



White line detection (top hat)



Detected lanes

Lane detection example

Adapted from CMM/ENSMP/ARMINES

Pixels and neighborhoods

- In many algorithms, not only the value of a particular pixel, but also the values of its neighbors are used when processing that pixel.
- The two most common definitions for neighbors are the 4-neighbors and the 8-neighbors of a pixel.

	N	
W	*	E
	S	

a) four-neighborhood N_4

NW	N	NE
W	*	E
SW	S	SE

b) eight-neighborhood N_8

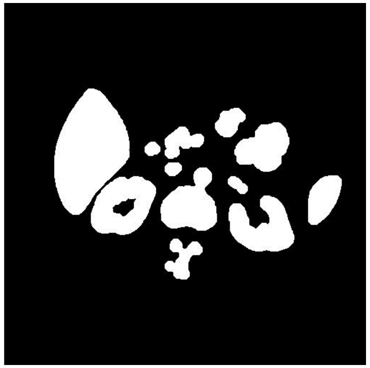
Figure 3.2: The two most common neighborhoods of a pixel.

Connected components analysis

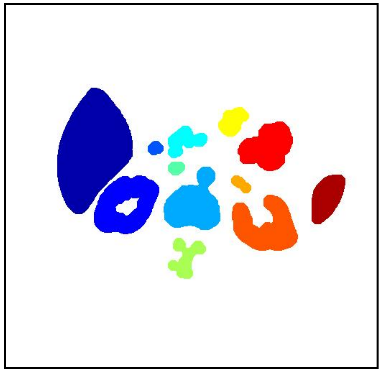
- Once you have a binary image, you can identify and then analyze each connected set of pixels.
- The connected components operation takes in a binary image and produces a labeled image in which each pixel has the integer label of either the background (0) or a component.

Original image

Thresholded image



After morphology



Connected components

Connected components analysis

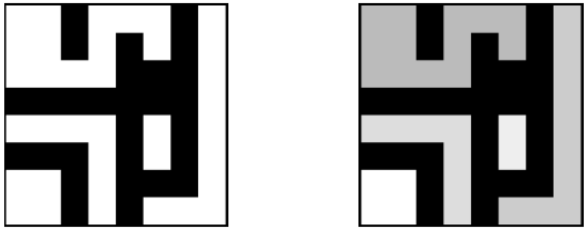
- Methods for connected components analysis:
  - Recursive tracking (almost never used)
  - Parallel growing (needs parallel hardware)
  - Row-by-row (most common)
    - Classical algorithm
    - Run-length algorithm (see Shapiro-Stockman)

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

a) binary image

1	1	0	1	1	1	0	2
1	1	0	1	0	1	0	2
1	1	1	1	0	0	0	2
0	0	0	0	0	0	0	2
3	3	3	3	0	4	0	2
0	0	0	3	0	4	0	2
5	5	0	3	0	0	0	2
5	5	0	3	0	2	2	2

b) connected components labeling



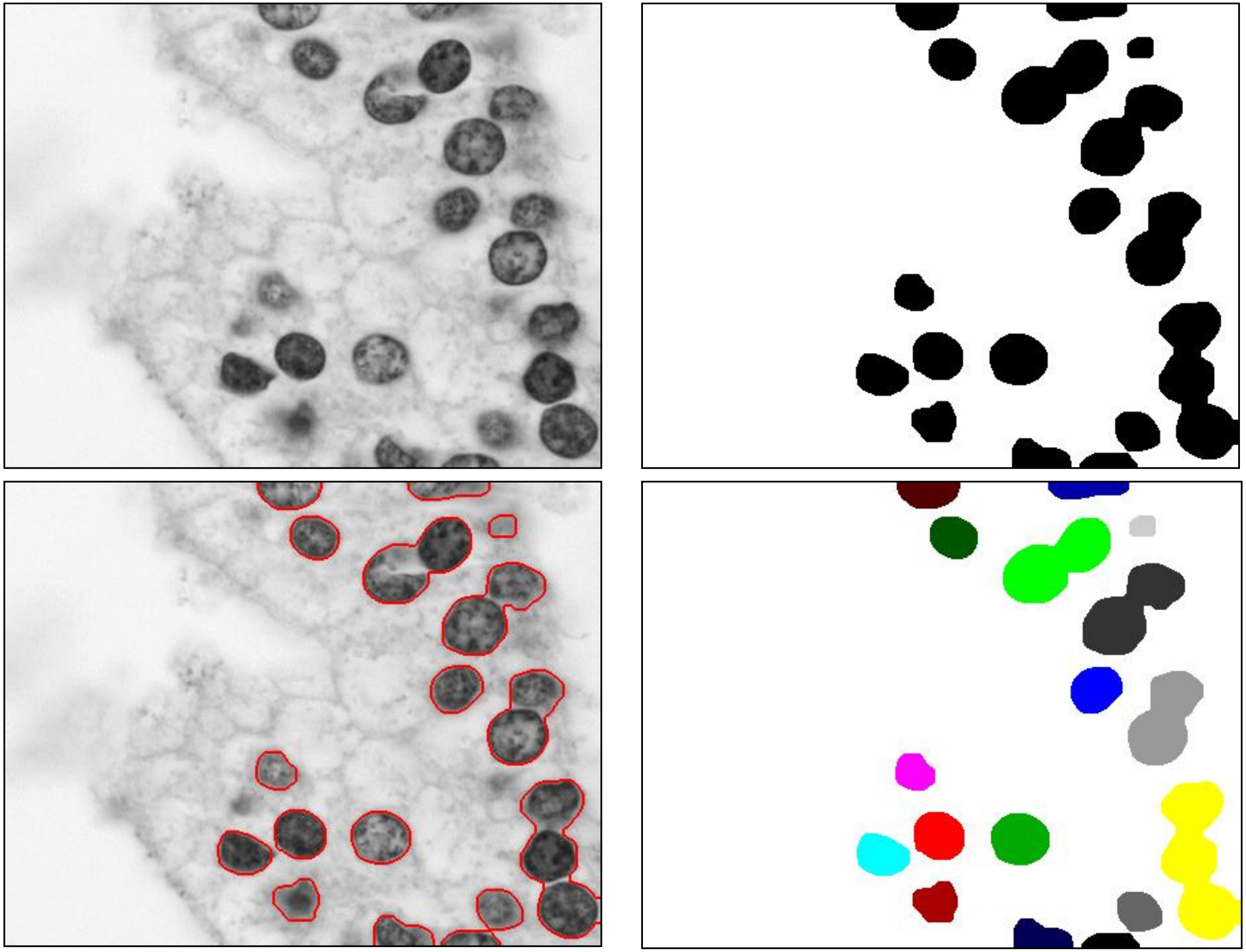
c) binary image and labeling, expanded for viewing

Adapted from Shapiro and Stockman

Connected components analysis

- Row-by-row labeling algorithm:
  1. The first pass propagates a pixel's label to its neighbors to the right and below it. (Whenever two different labels can propagate to the same pixel, these labels are recorded as an equivalence class.)
  2. The second pass performs a translation, assigning to each pixel the label of its equivalence class.
- A union-find data structure is used for efficient construction and manipulation of equivalence classes represented by tree structures.

Connected components analysis



Connected components analysis