Binary Image Analysis
Week 04

Announcement

- Introduction of our course representatives
  - Duan Dao: u7236134@anu.edu.au
  - Maiteryi Singh: u7075106@anu.edu.au

- Lab Engagement: Ask questions related to your Lab assignment.
- Watch the video about how to write a good report.

Announcement: Mid-term Exam

EXAMS MANAGER

Semester 1 - Mid Semester, 2021 : Final Timetable

To view a different exam period, click here.

Unit Codes ENGN6528

You can search for multiple units by putting a comma between your unit codes.

SEARCH : FINAL TIMETABLE

Displaying records 1 to 1 of 1

Exam Code	Exam Title	Date	Time	Writing Time (minutes)	Reading Time (minutes)	Venue	Exam Conditions
ENGN6528_Semester 1	Computer Vision	Tuesday 20/04/2021	12:00pm	90	15	Online Exam - Wattle	Any material ,

Displaying records 1 to 1 of 1

Updated: 9 March 2021 / Responsible Officer: Registrar, Student Administration / Page Contact: Examinations Officer

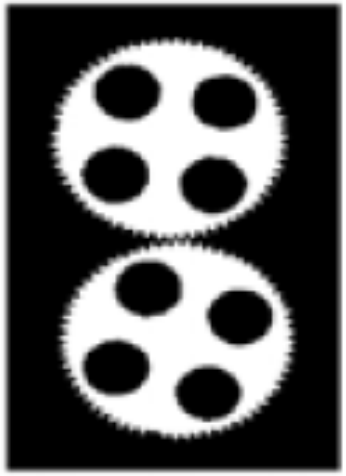
Announcement: Mid-term Exam

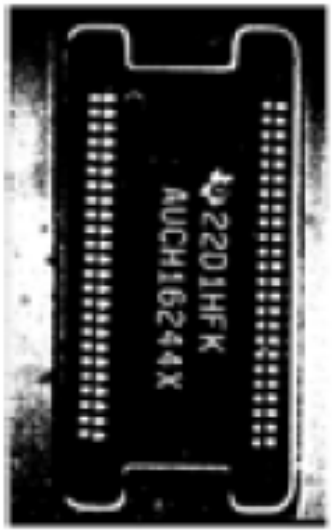
- Sample mid-term papers will be posted online.
- We will also provide answers in the video format. It requires students to check the answers by watching the video by yourself.

Contents

- Image threshold (histogram binarization)
- Mathematical Morphology

Why binary ?





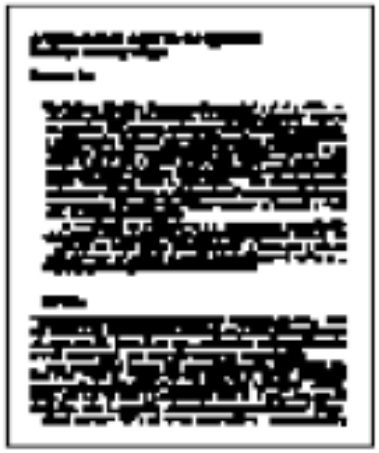
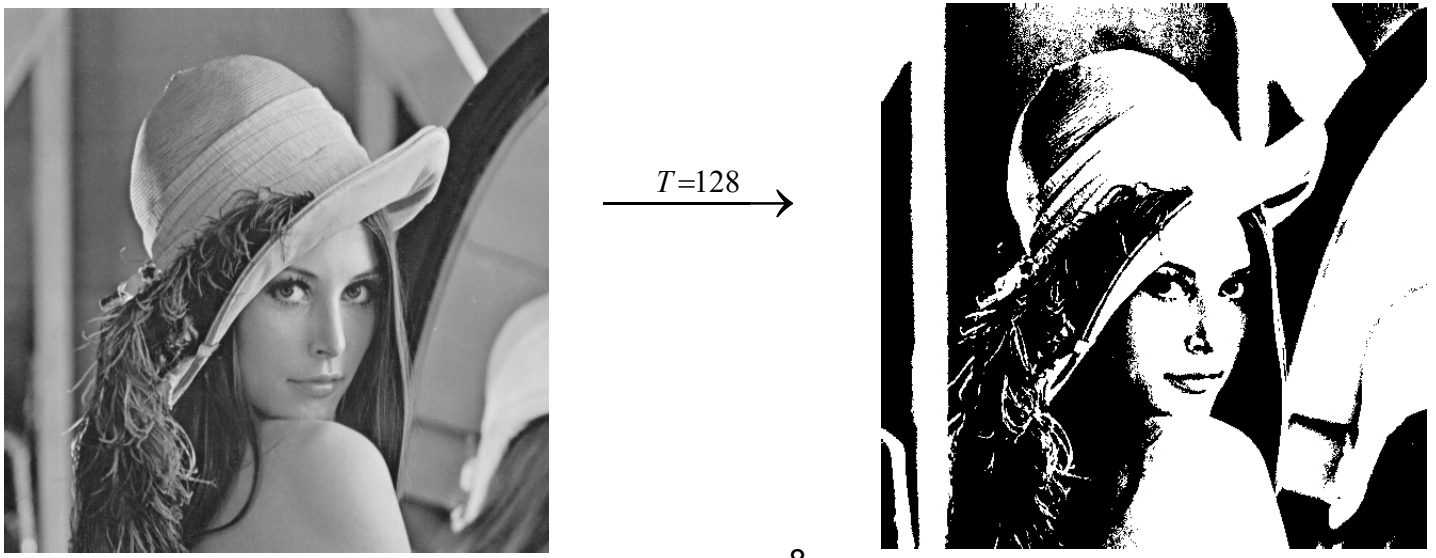


Image threshold (binarization)

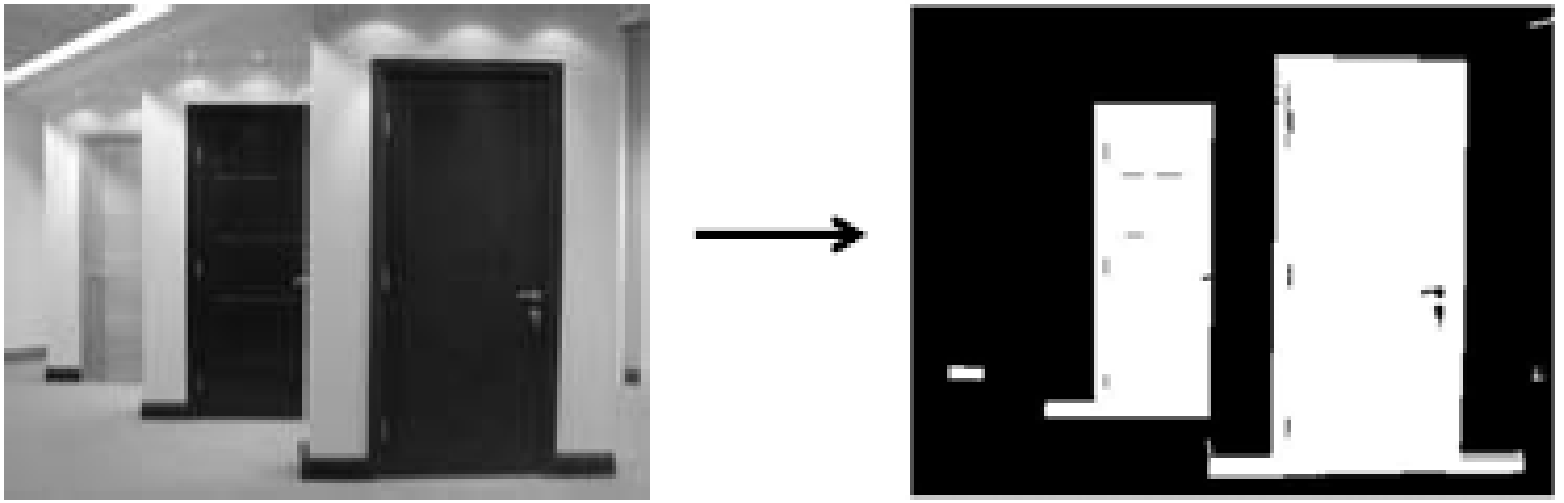
- What is **Thresholding** ?
  - It is a pixel labeling operation.
  - It assigns a binary value to each pixel.
    - **Binary Value 1**: pixels have higher intensity values
    - **Binary Value 0:** pixels have lower intensity values



Thresholding

- Given a grayscale image or an intermediate matrix → threshold to create a binary output.

Example: intensity-based detection



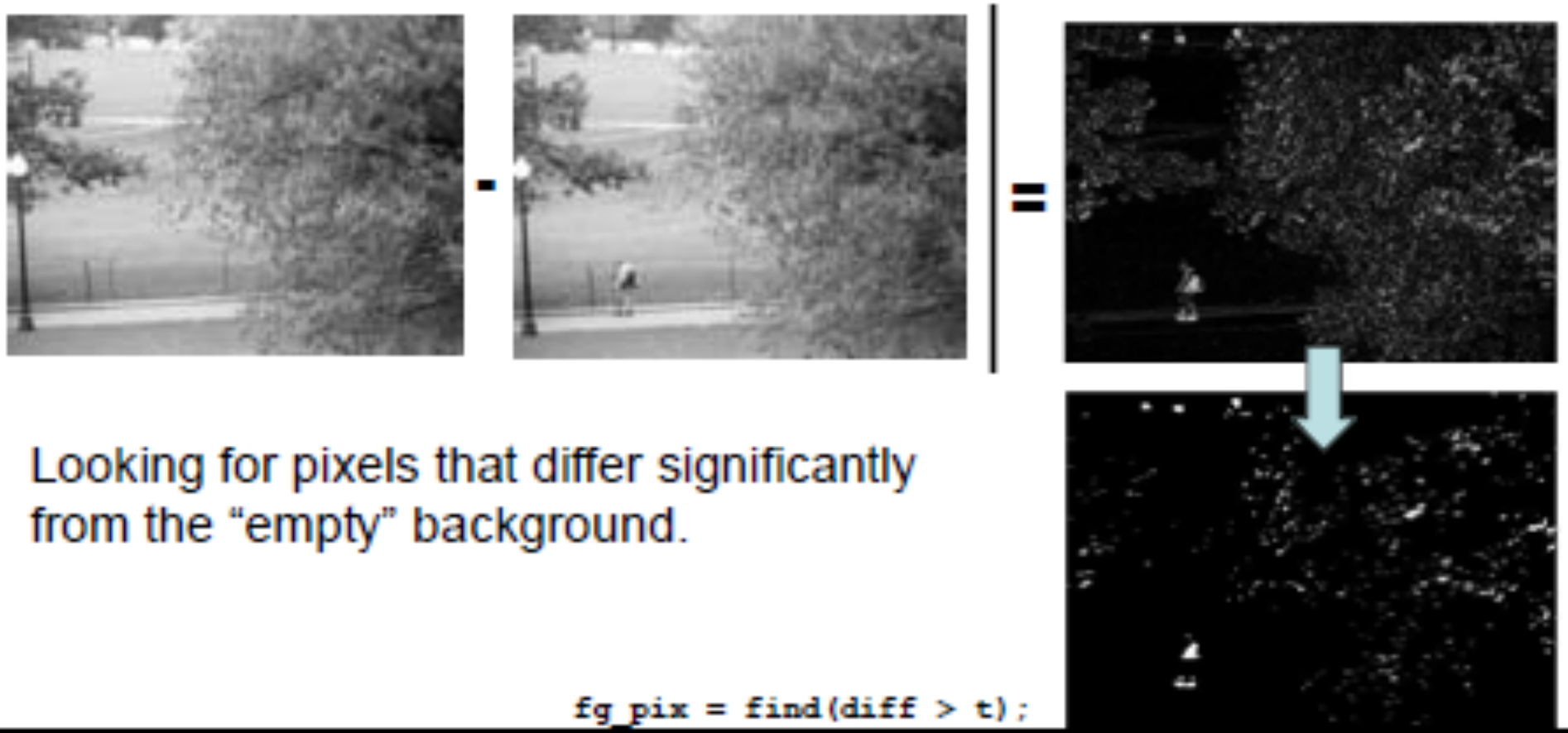
```
fg_pix = find(im < 65);
```

Looking for dark pixels

Thresholding

- Given a grayscale image or an intermediate matrix → threshold to create a binary output.

Example: background subtraction

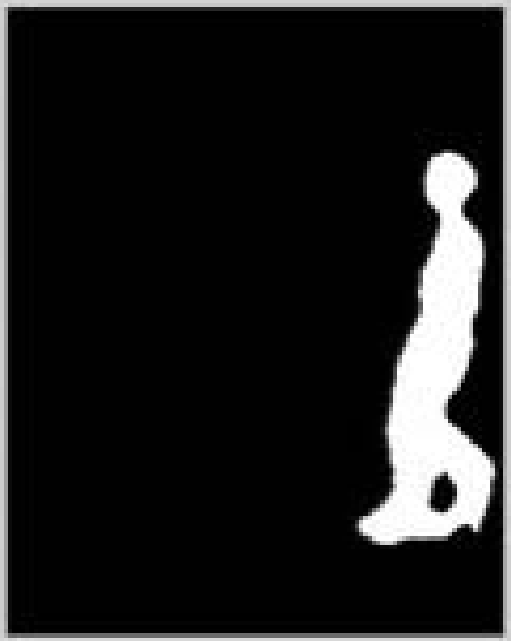


Looking for pixels that differ significantly from the "empty" background.

```
fg_pix = find(diff > t);
```

Example using binary image analysis: Bg subtraction + blob detection

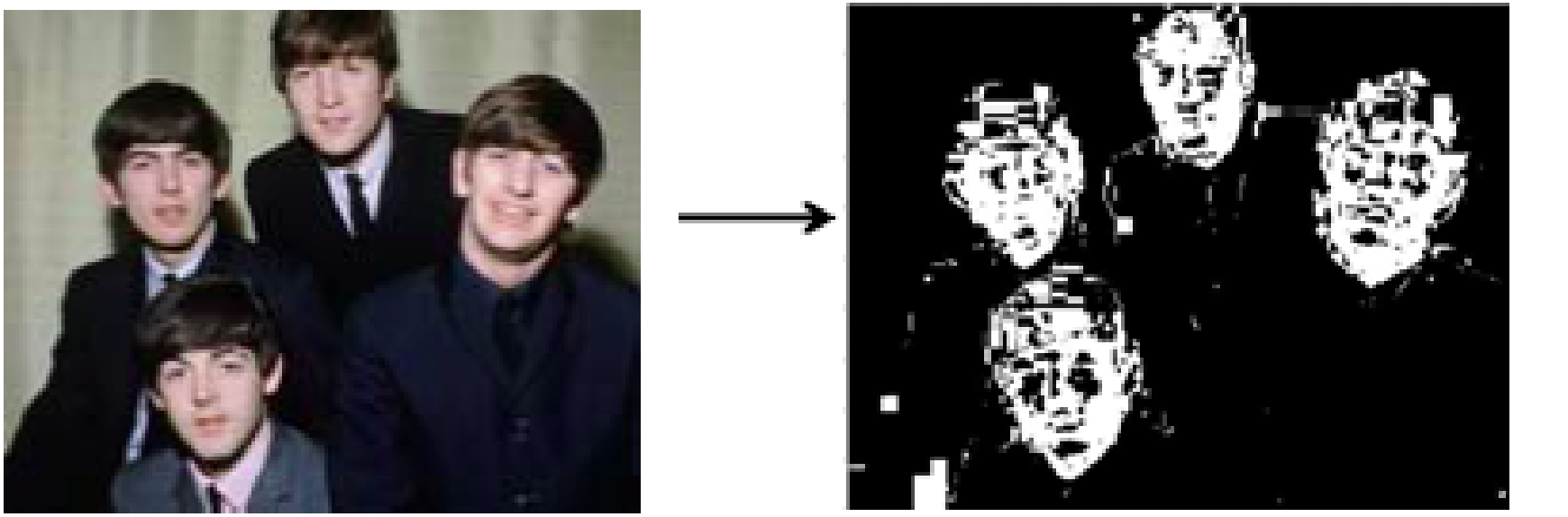
...



Thresholding

- Given a grayscale image or an intermediate matrix → threshold to create a binary output.

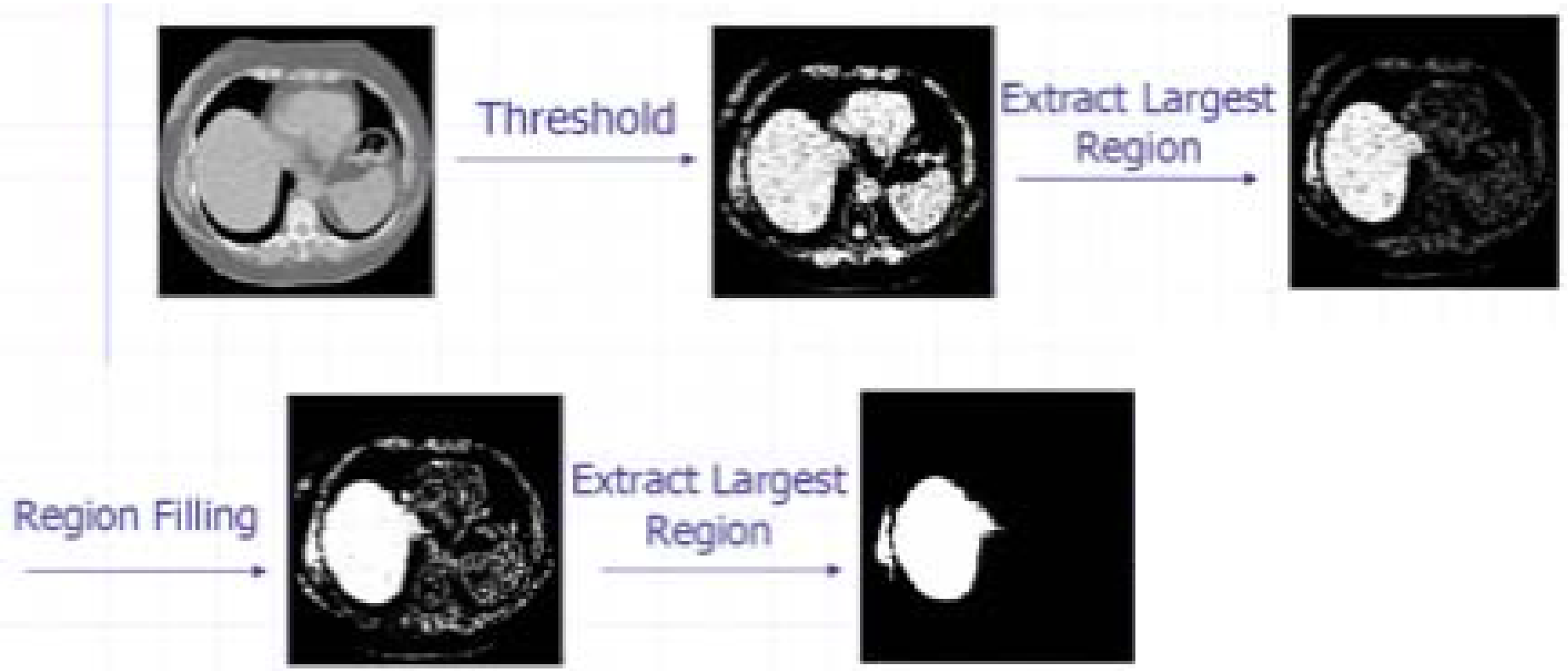
Example: color-based detection



```
fg_pix = find(hue > t1 & hue < t2);
```

Looking for pixels within a certain hue range.

Example using binary image analysis: segmentation of a liver

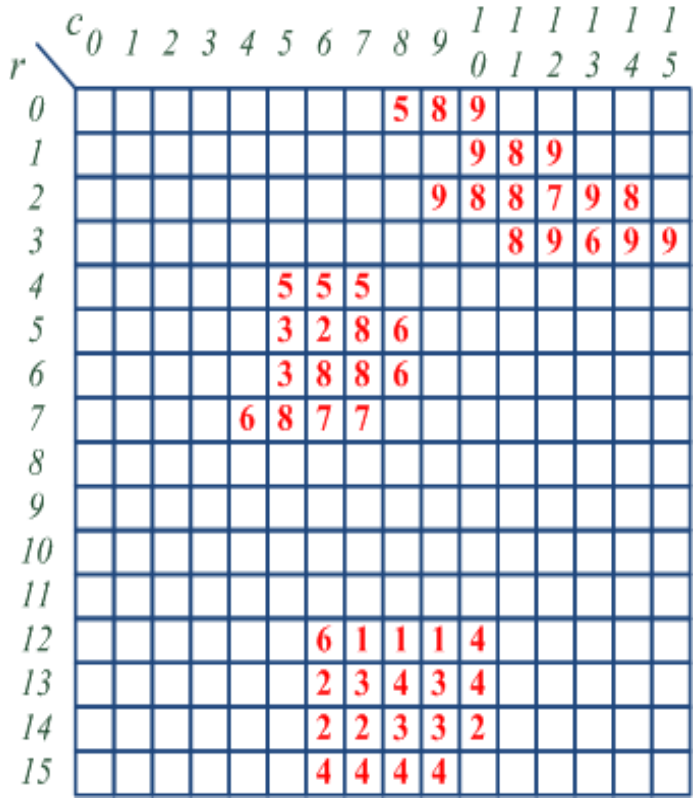


- Mathematical Formulation

$$B(r,c) = \begin{cases} 1 & if \quad I(r,c) \geq T \\ 0 & if \quad I(r,c) < T \end{cases}$$

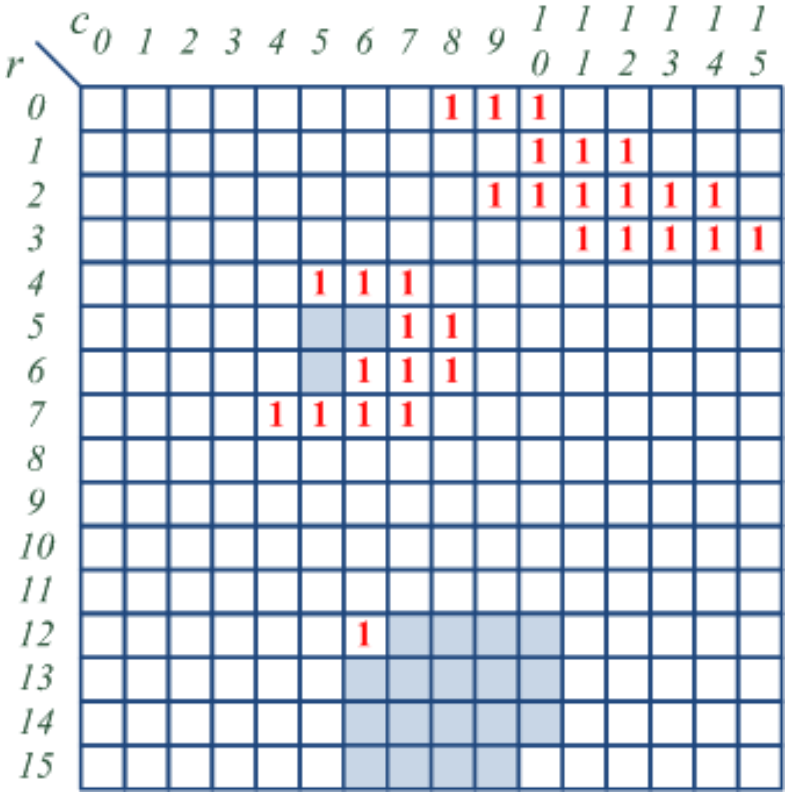
  - (r,c) : row and column
  - $I(.,.)$: gray-level intensity image
  - T : intensity threshold
  - $B(.,.)$: binary intensity image

How to select an appropriate threshold ?



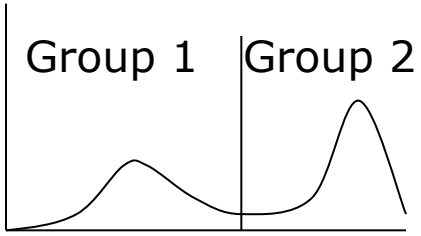
$I(.,.)$

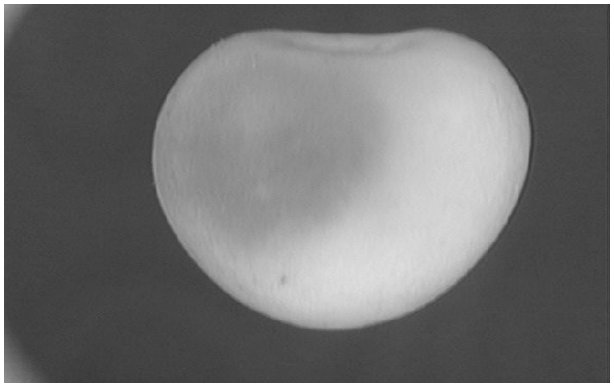
$T = 5$

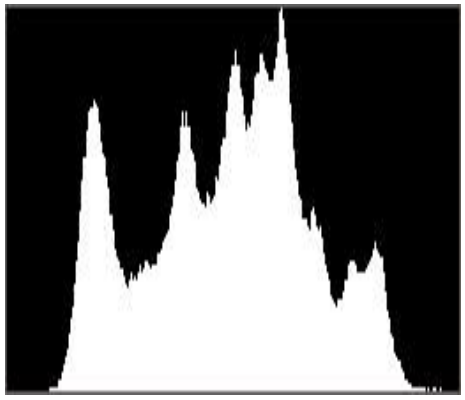


$B(.,.)$

- Approach: Histogram analysis

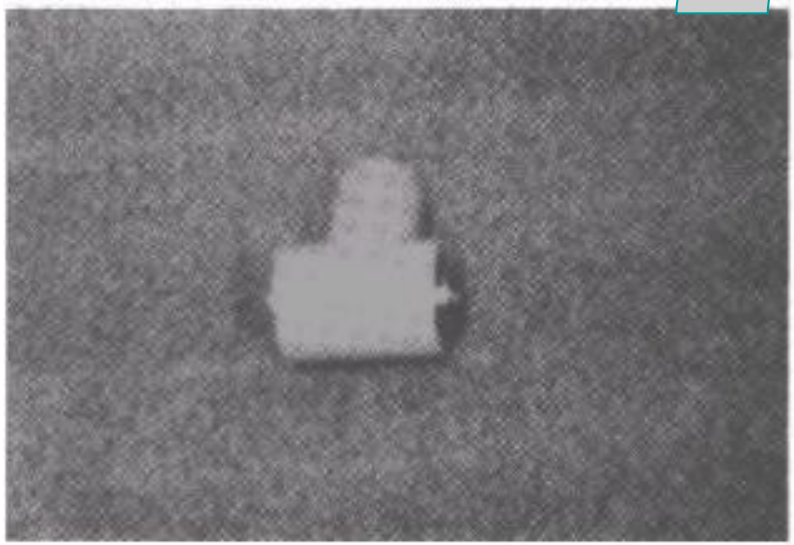






Where to pick an optimal threshold?

$$h(m)$$



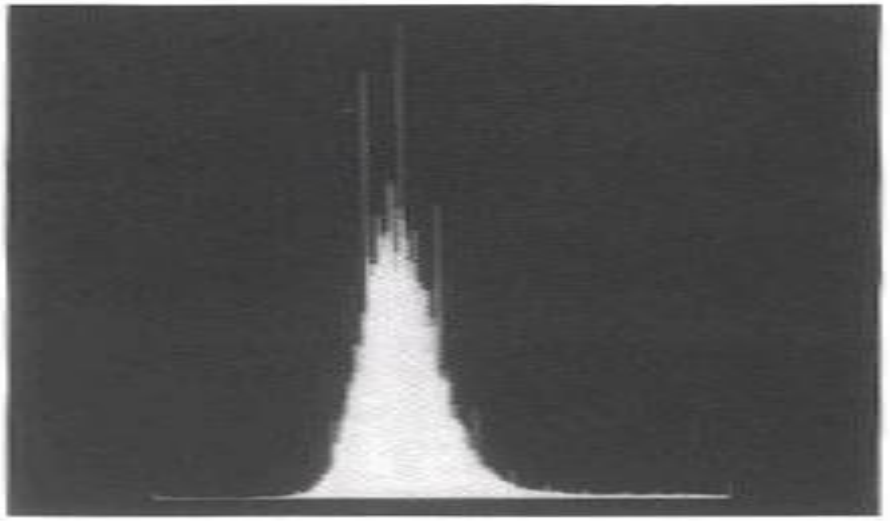
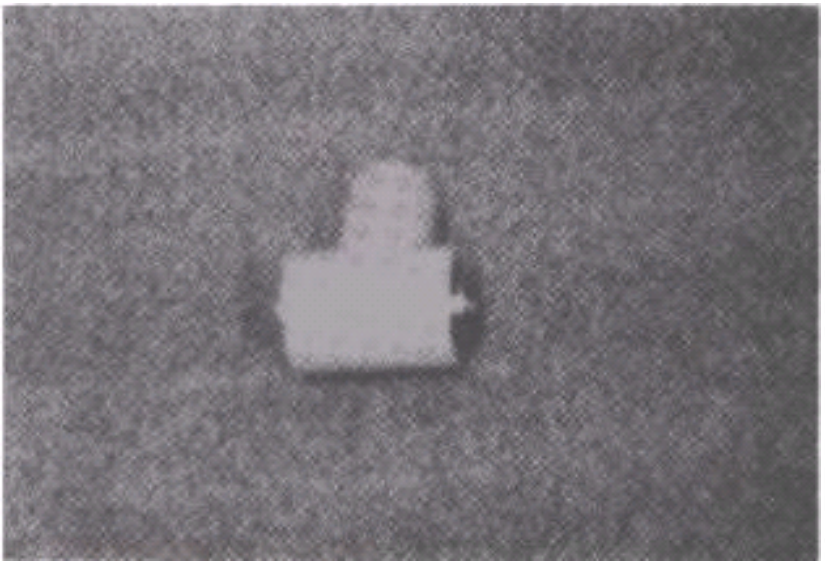


Figure 2.6 BNC T-connector against a dark background.

Figure 2.7 Histogram of the BNC T-connector image of Fig. 2.6.

Histogram



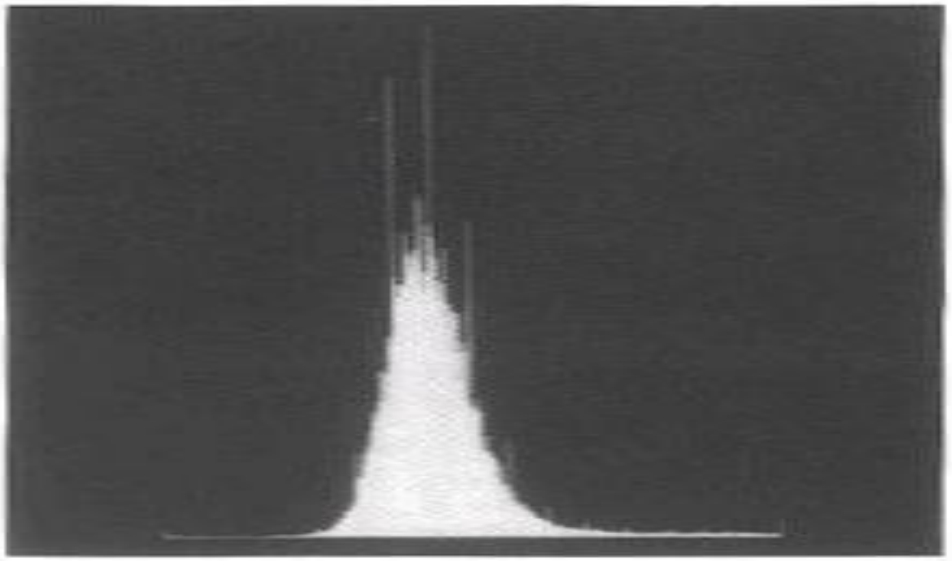
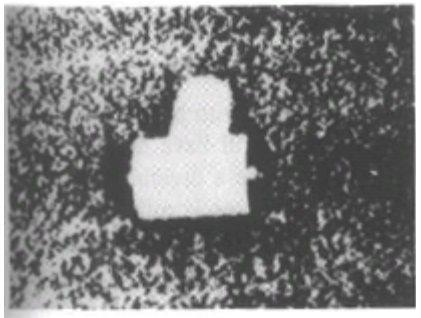
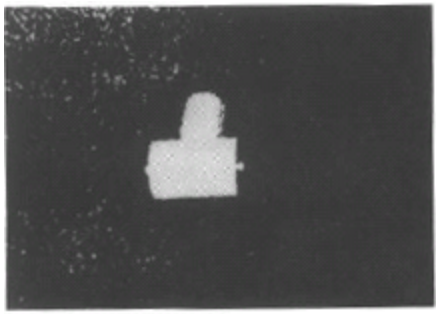


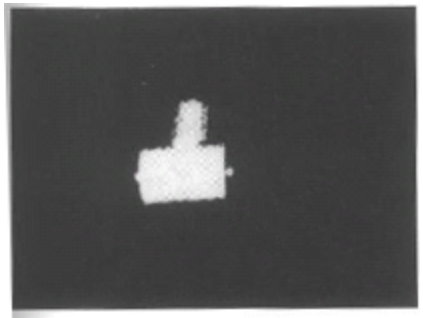
Figure 2.7 Histogram of the BNC T-connector image of Fig. 2.6.



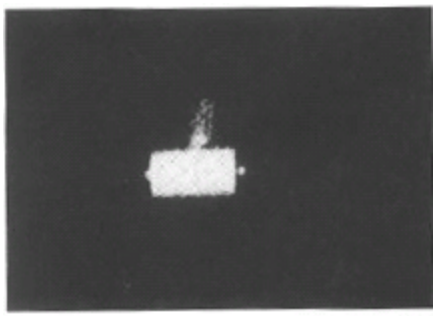
(a)



(b)



(c)



(d)

T=110 **T=130** **T=150** **T=170**

Basic Global Thresholding

Simple Idea:

1. Select an initial estimate for T.
2. Segment image using T into two groups G1 and G2.
3. Compute the average gray level value as μ_1 and μ_2
4. Compute a new threshold value:

$$T = \frac{1}{2}(\mu_1 + \mu_2)$$

5. Repeat steps 2 through 4 until the difference in T in successive iteration is smaller than a predefined parameter T_0

How to find an optimal threshold ?

- There is no generic solution.
- The optimal solution is often problem-dependent;

- A heuristic binarisation algorithm based on K-L divergence:

  – fitting a 2-mode Gaussian mixture model to the histogram: minimizing K-L divergence.

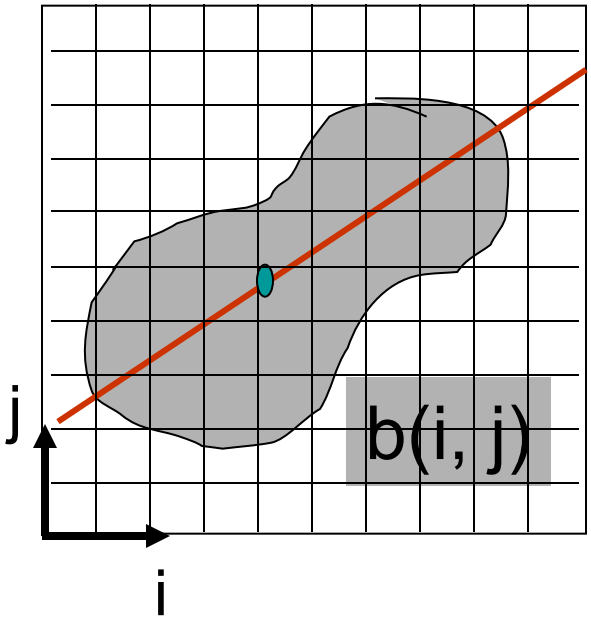
Region Property Computation

Region property computation

- For each connected region, we can compute many of its properties, e.g.
  - Areas
  - Perimeter
  - Centre of gravity
  - circularity,
  - Major axis
  - Minor axis
  - mean and standard deviation of radial distance
  - bounding box
  - extremal axis length from bounding box
  - **second order moments** (row, column, mixed)
  - …

Region properties

- Assume:
  b(i, j) is discrete,
  only one object is in the image.



- Areas (0-th Moment) $A = \sum\sum b_{ij}$

- Center of Mass (First Moment)

$$\bar{x} = \frac{1}{A}\sum\sum i\, b_{ij} \qquad \bar{y} = \frac{1}{A}\sum\sum j\, b_{ij}$$

- Second Moments:

$$a' = \sum\sum i^2\, b_{ij} \qquad b' = 2\sum\sum ij\, b_{ij} \qquad c' = \sum\sum j^2\, b_{ij}$$

Note: a',b',c' are defined w.r.t origin

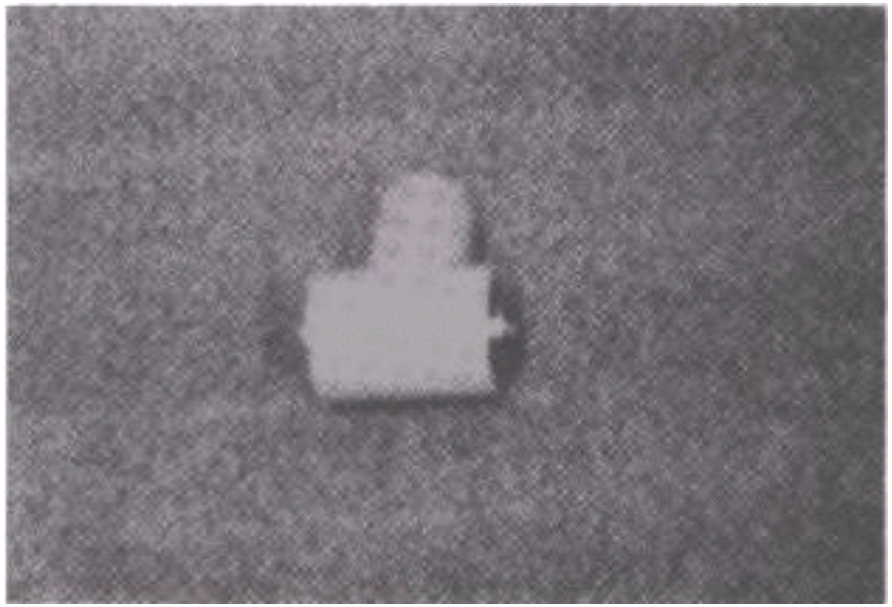
Mathematical Morphology

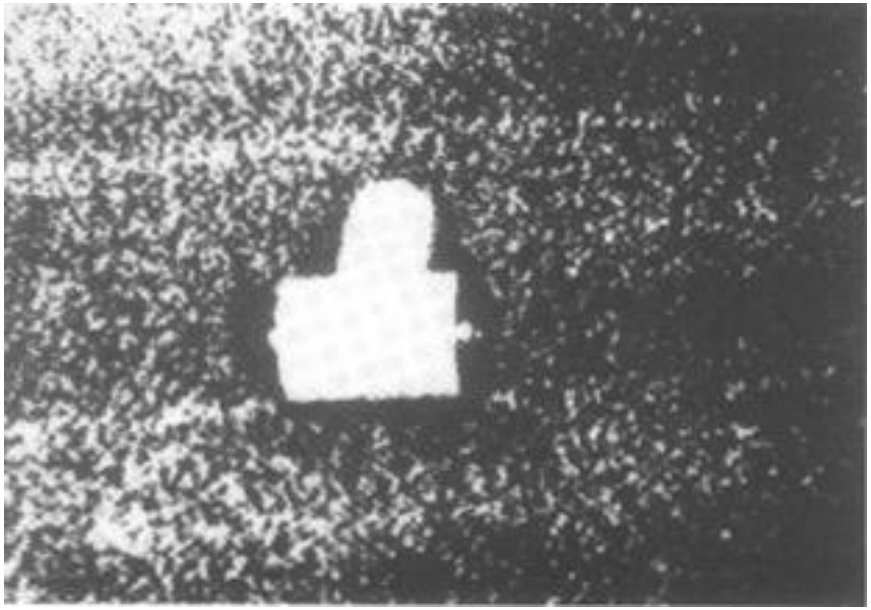
- **Morphology**: originally, a branch of biology science that studies the form and structure of animals and plants

- Mathematical Morphology in image processing is used to extract image components for representation and description of region shape, such as boundaries, skeletons, and the convex hull.

Motivation: remove small noisy regions

- Example result by a binarisation algorithm





Mathematical Morphology

In binary image processing, mathematical morphology consists of two basic operations,

dilation , erosion

and several composite operations

closing , opening, . . .

Logic Operations Involving Binary Image

- The principal logic operations used in image processing: AND, OR, and NOT(complement)
- Logic operations are performed on a pixel-wise basis, except the NOT operation.
- Their properties are summarized below.

LE 9.1
three basic
cal operations.

p	q	p AND q (also $p \cdot q$)	p OR q (also $p + q$)	NOT (p) (also $\bar{p}$)
0	0	0	0	1
0	1	0	1	1
1	0	0	1	0
1	1	1	1	0

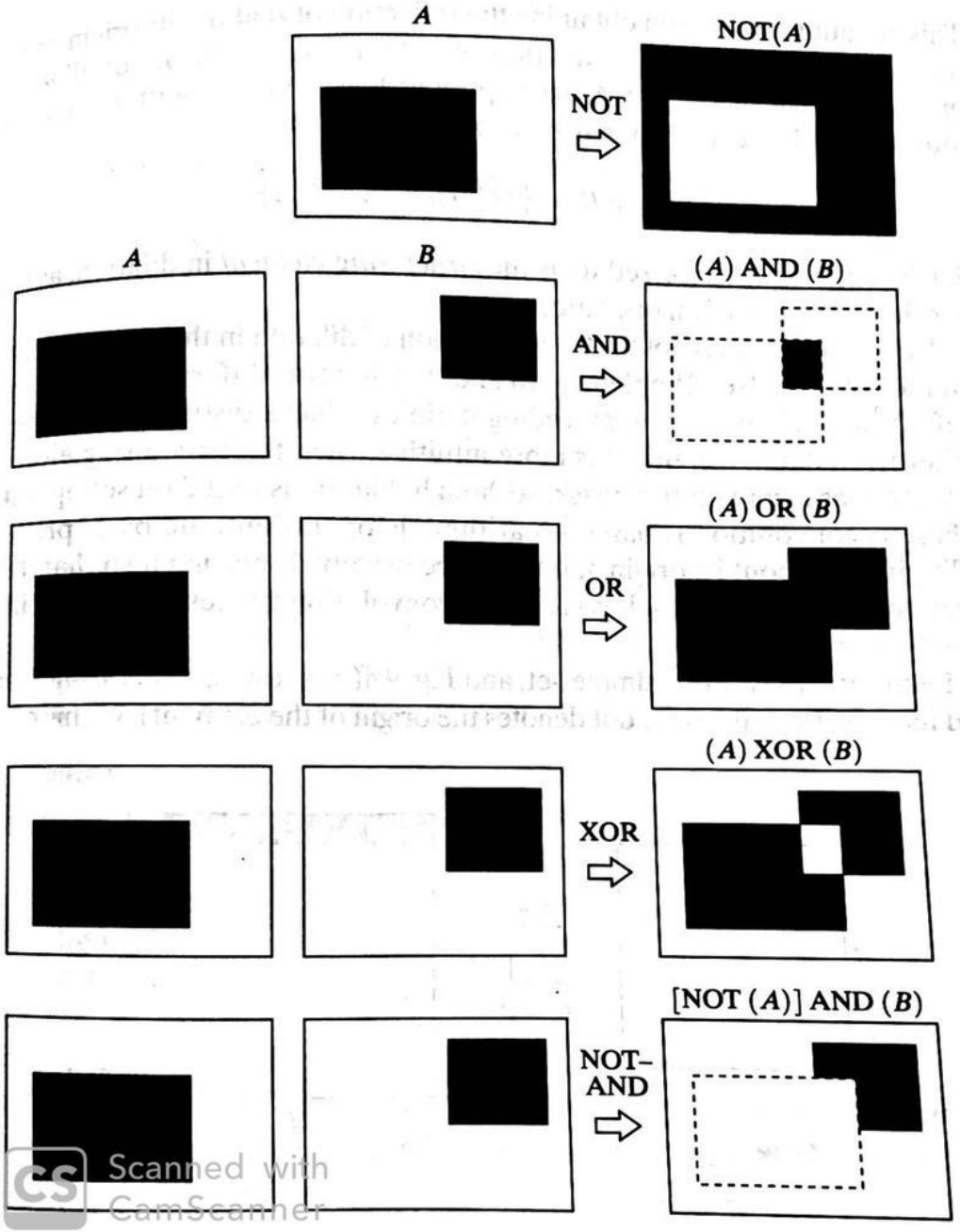


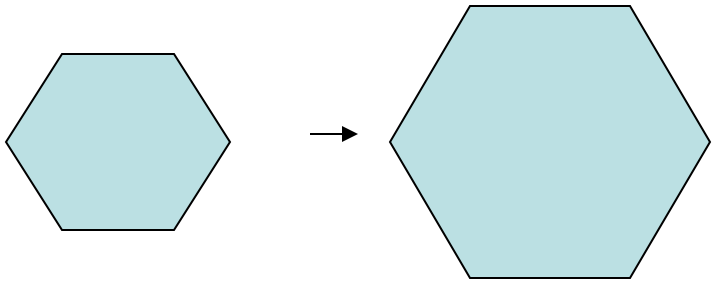
FIGURE 9.3 Some logic operations between binary images. Black represents binary 1s and white binary 0s in this example.

Dilation

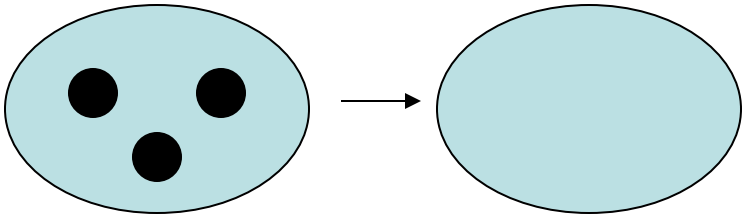
Dilation expands the connected sets of 1s of a binary image.

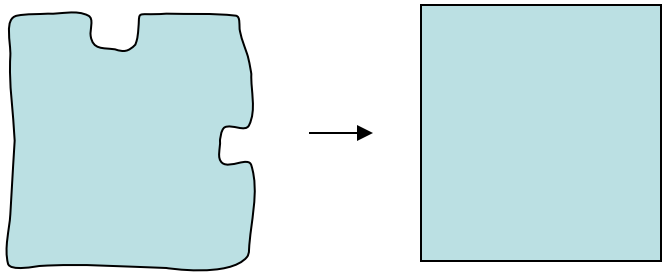
It can be used for

1. growing features



2. filling holes and gaps



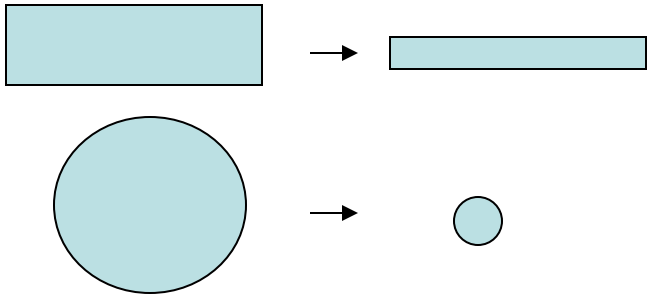


Erosion

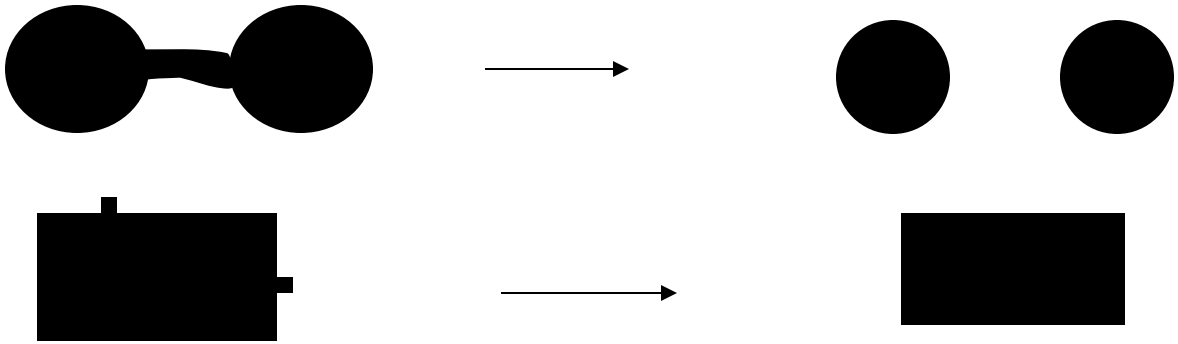
Erosion shrinks the connected sets of 1s of a binary image.

It can be used for

1. shrinking features



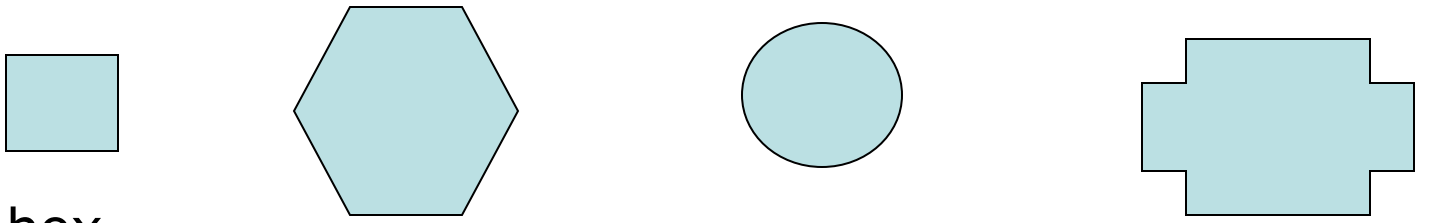
2. Removing bridges, branches and small protrusions



Structuring Elements

A structuring element is a shape mask used in the basic morphological operations.

They can be any shape and size that is digitally representable, and each has an origin.



box(length,width)

disk(diameter)

Opening and Closing

- Closing is the composite operation of dilation followed by erosion (with the same structuring element)

- Opening is the composite operation of erosion followed by dilation (with the same structuring element)

- Detailed explanations
  - Dilation
  - Erosion
  - Opening
  - Closing

Matlab

- `N = hist(Y,M)`
- `L = bwlabel (BW,N);`
- `STATS = regionprops(L,PROPERTIES) ;`
  - `'Area'`
  - `'Centroid'`
  - `'BoundingBox'`
  - `'Orientation', …`
- `IM2 = imerode(IM,SE);`
- `IM2 = imdilate(IM,SE);`
- `IM2 = imclose(IM, SE);`
- `IM2 = imopen(IM, SE);`

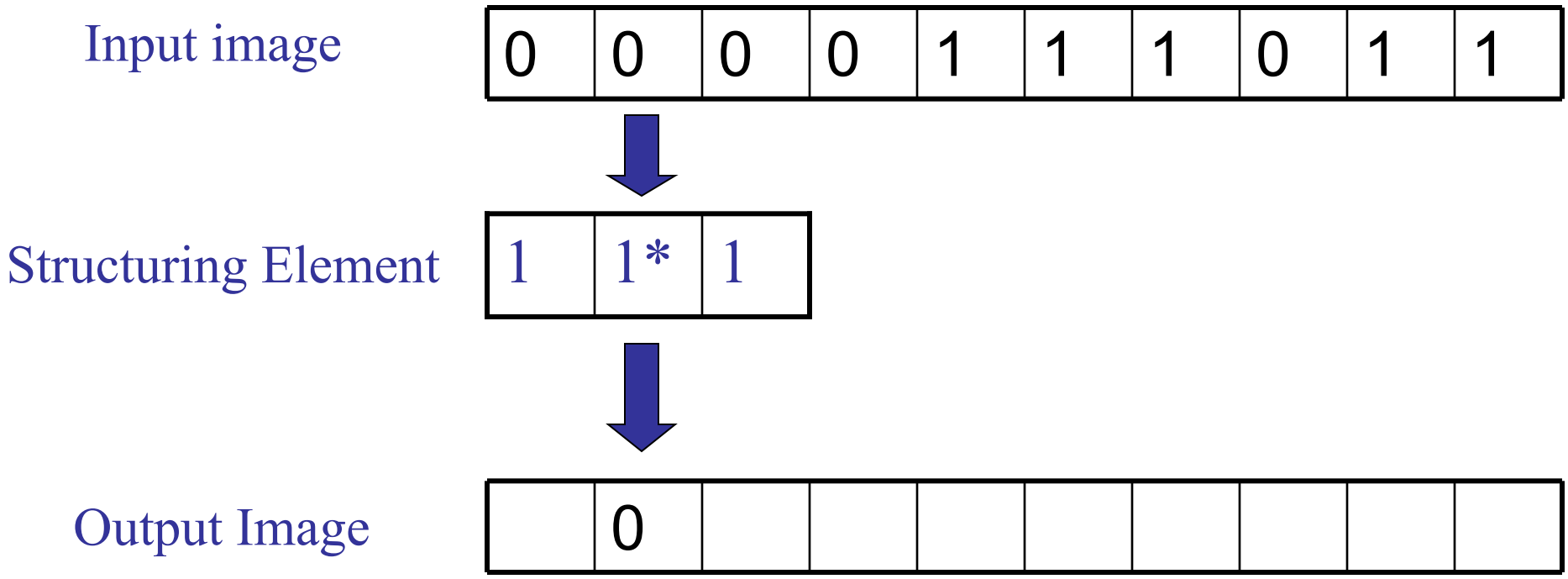
DILATION

dilate(B,S) takes binary image B, places the origin of structuring element S over each 1-pixel, and ORs the structuring element S into the output image at the corresponding position.

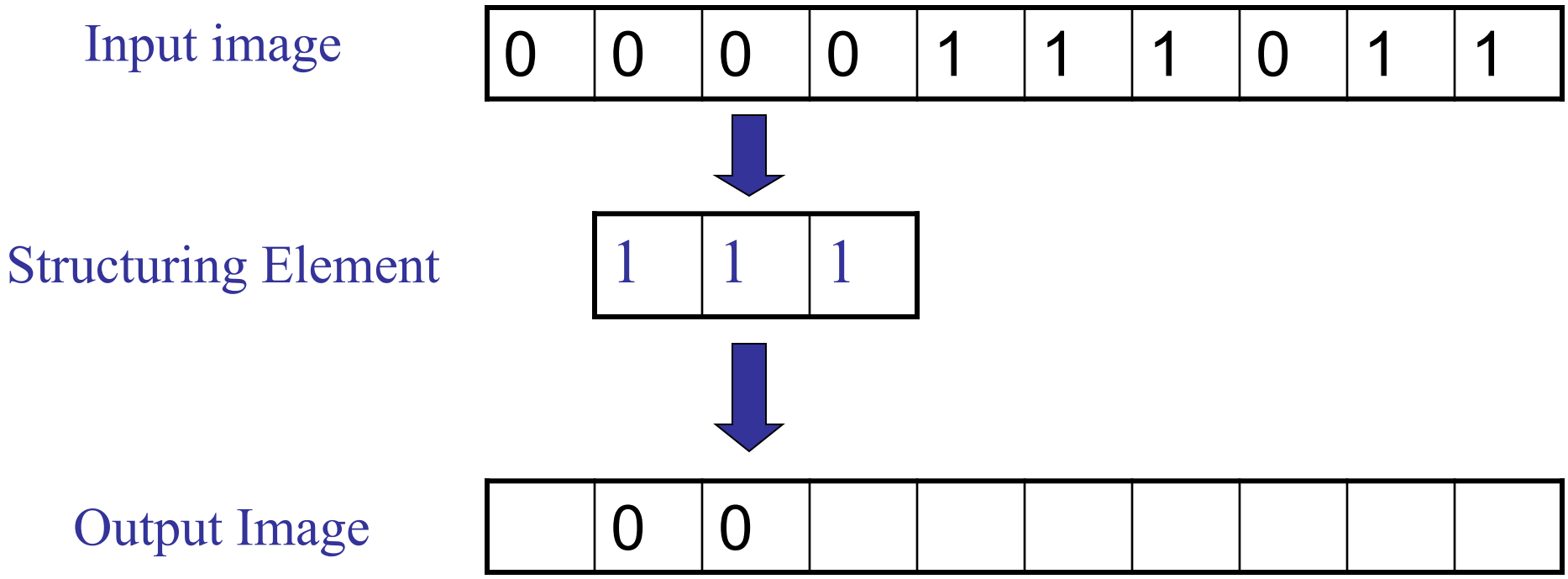
Dilation

- **Dilation** is the set of all points in the image, where the origin of the structuring element “hits” the foreground.
- Consider each foreground pixel in the input image
  - If the structuring element origin hits the foreground image, “OR” the structuring element to the image.
- **Input:**
  - **Binary Image**
  - **Structuring Element, containing only 1s!!**

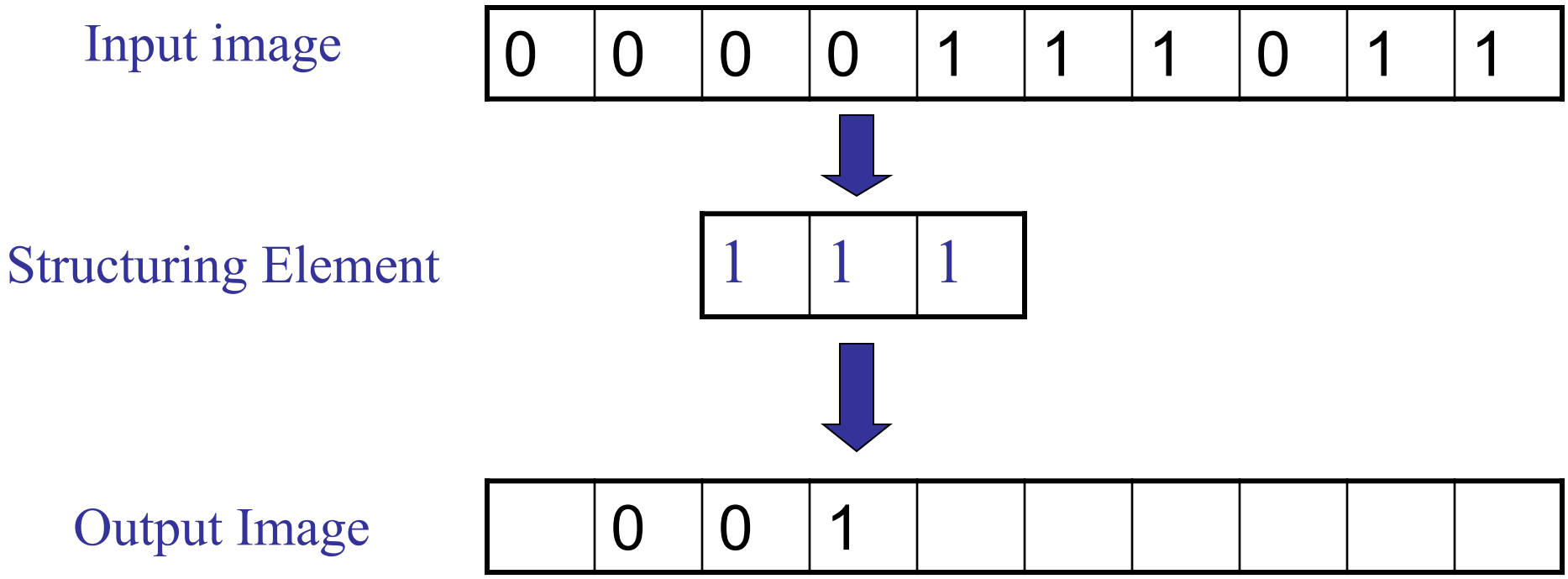
Example for Dilation



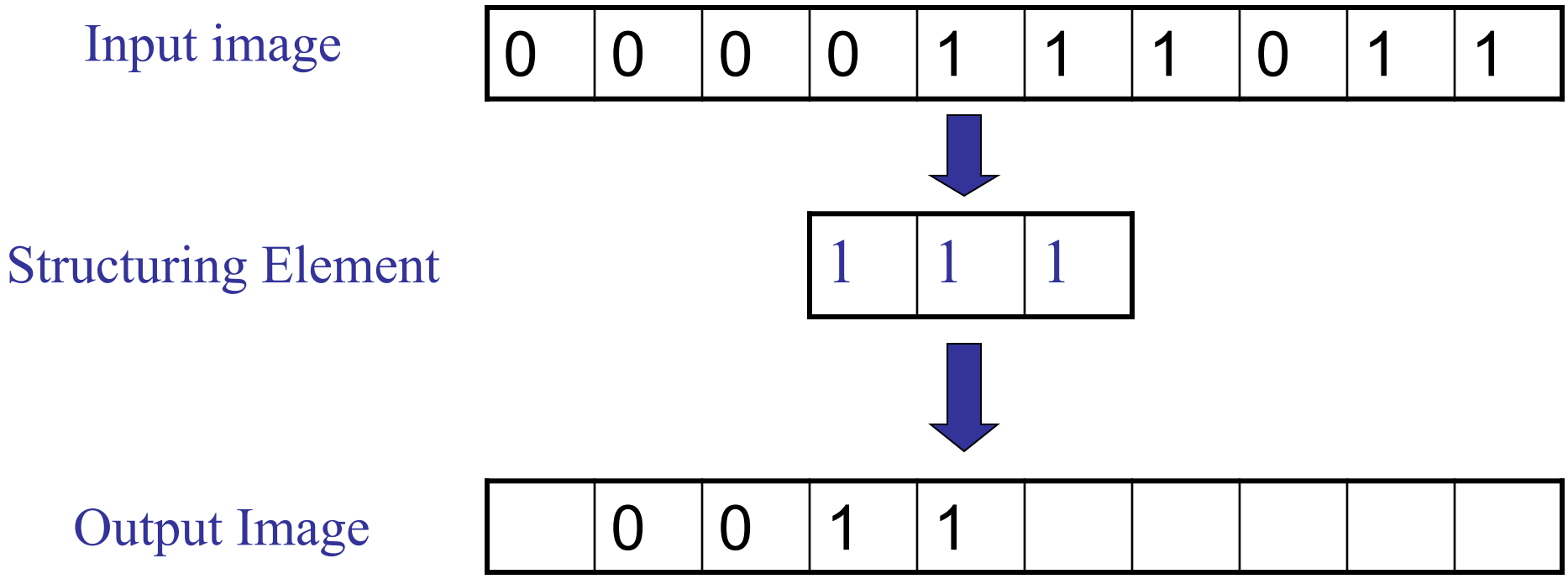
Example for Dilation



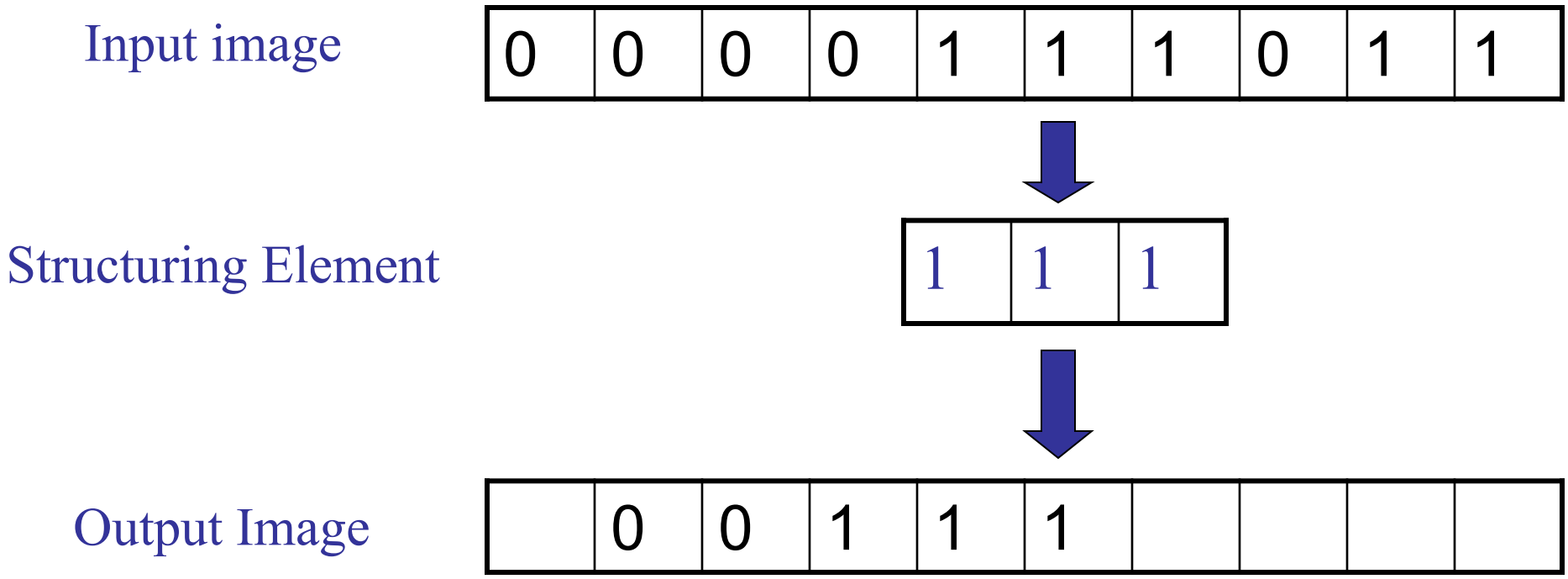
Example for Dilation



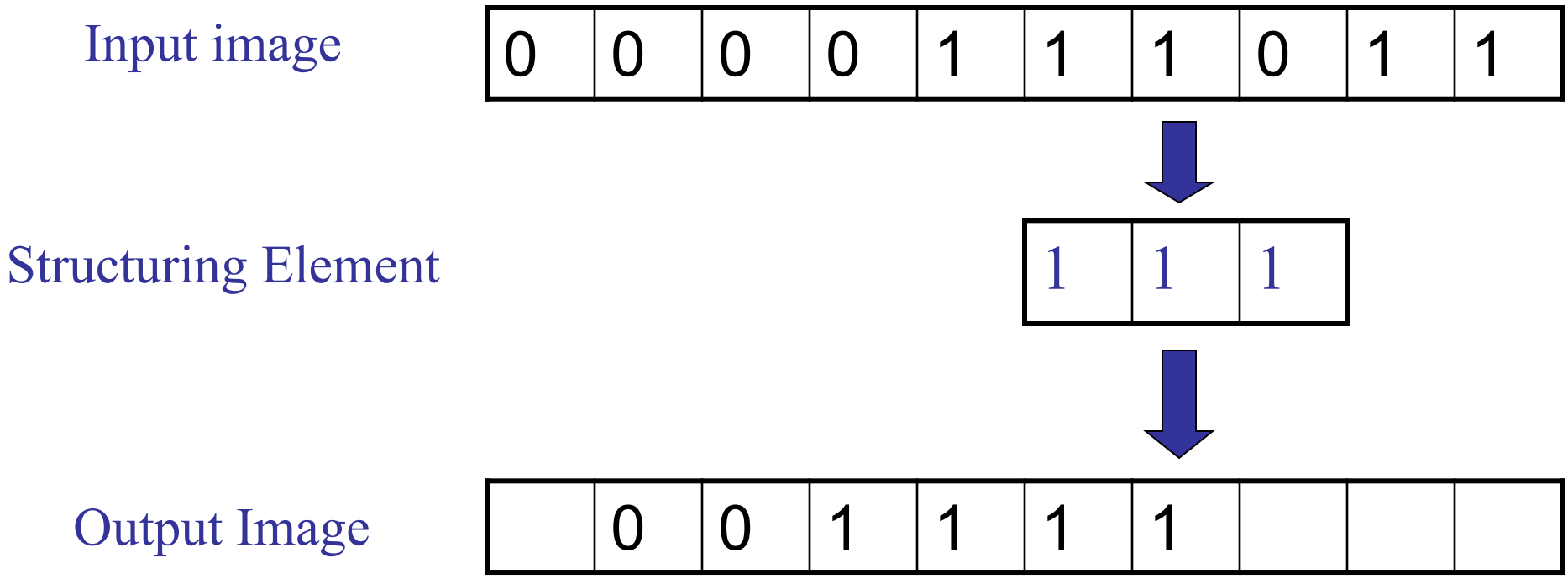
Example for Dilation



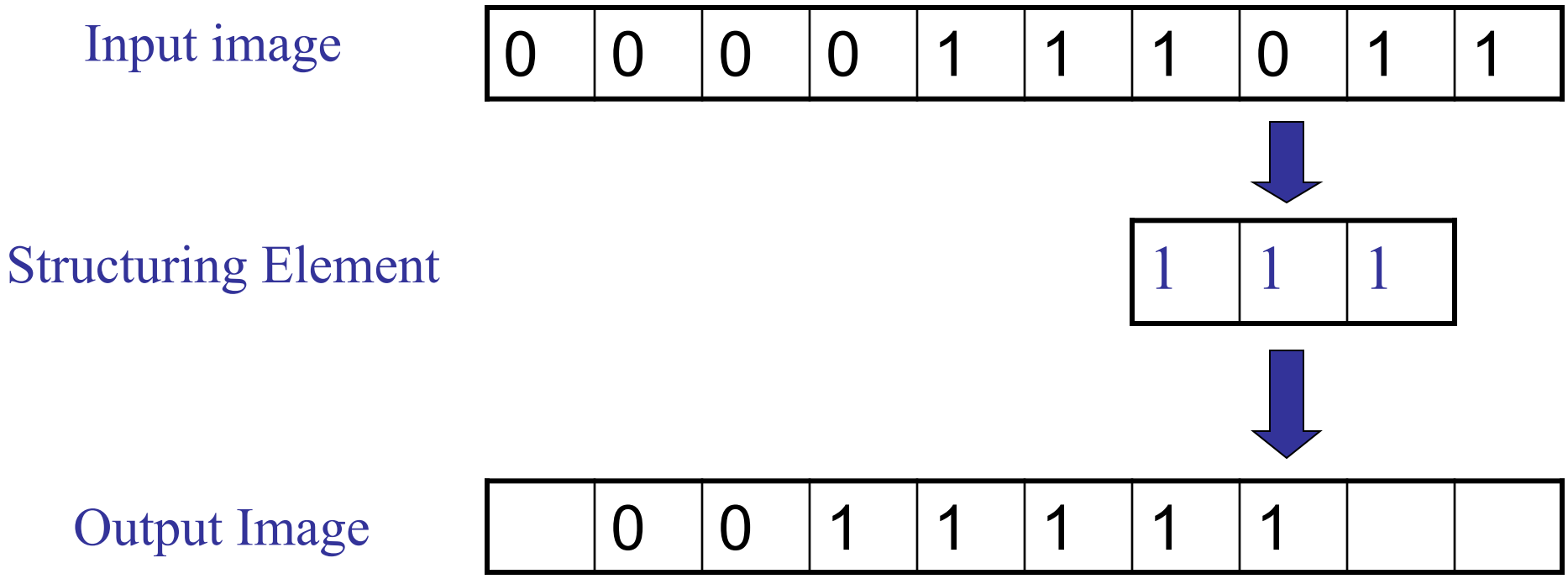
Example for Dilation



Example for Dilation

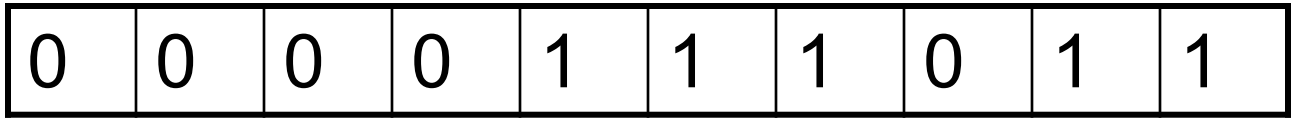


Example for Dilation

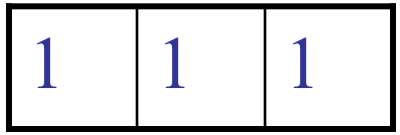


Example for Dilation

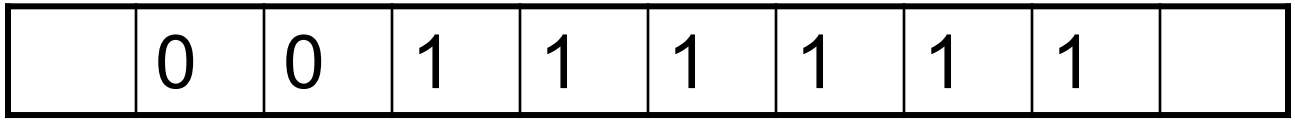
Input image



Structuring Element

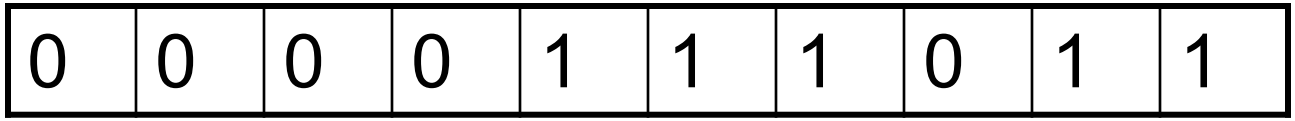


Output Image

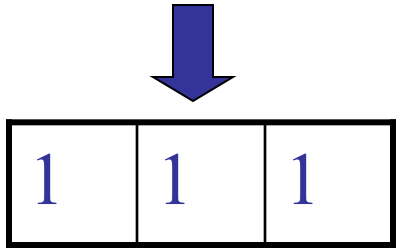


Example for Dilation

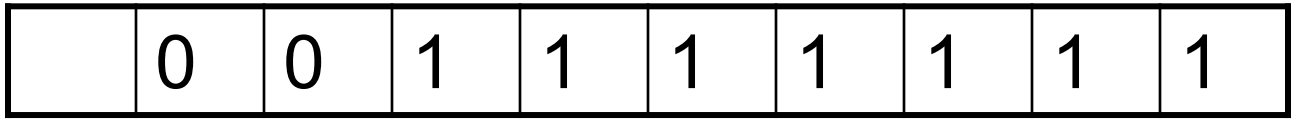
Input image



Structuring Element

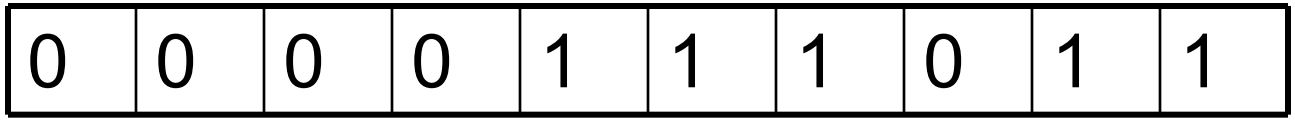


Output Image

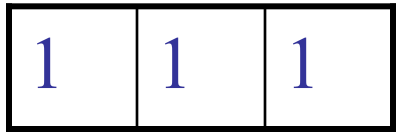


Example for Dilation

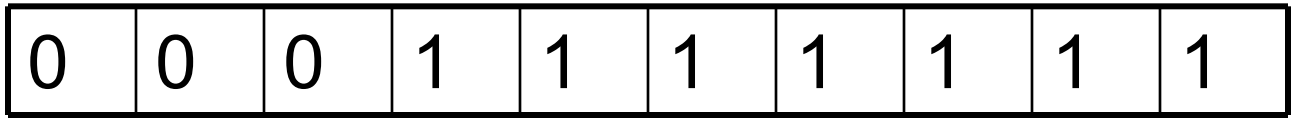
Input image



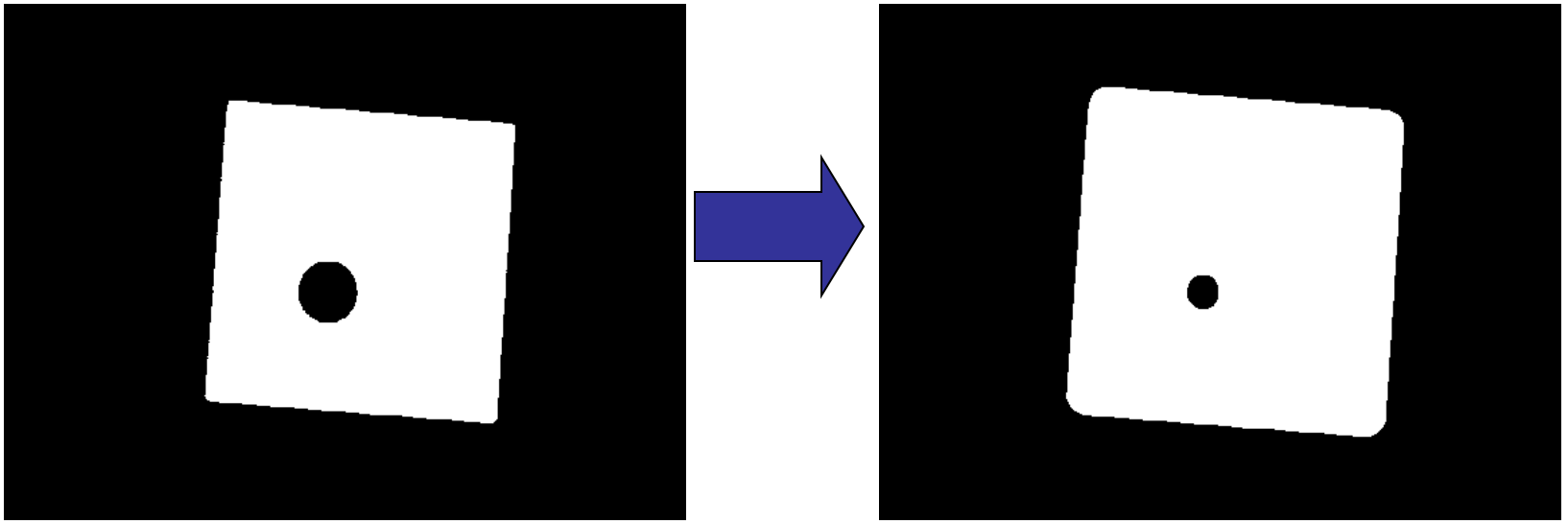
Structuring Element



Output Image



Another Dilation Example

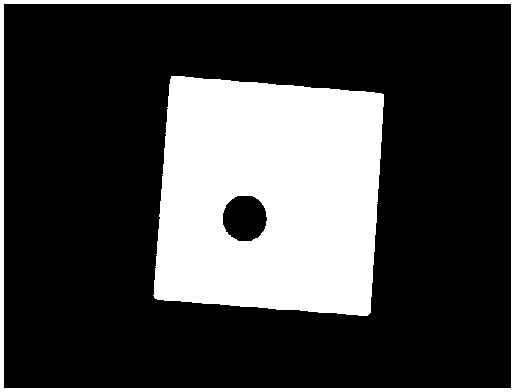
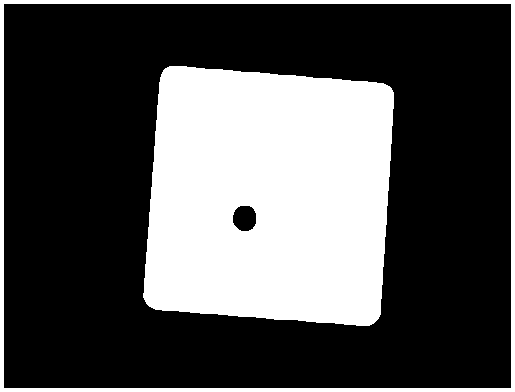
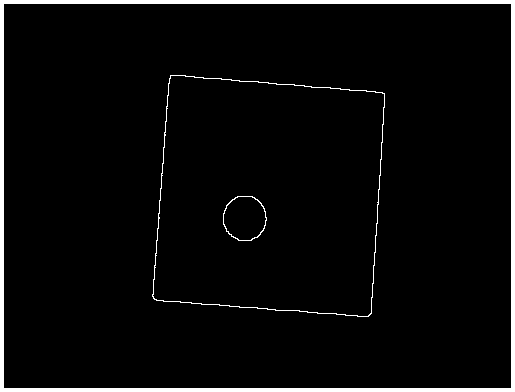


- Image get lighter, more uniform intensity

Edge detection example

- Edge Detection

1. Dilate input image
2. Subtract input image from dilated image
3. Edges remain!

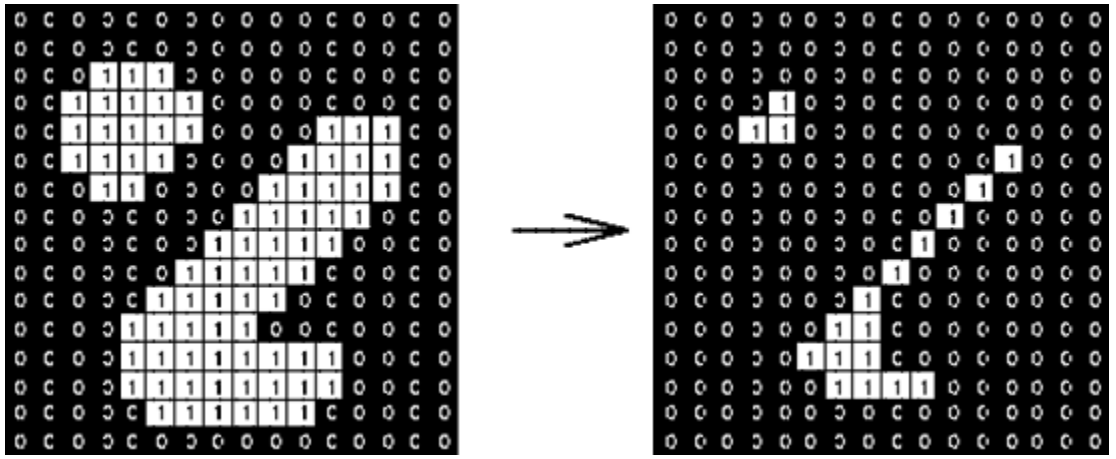




EROSION

erode(B,S) takes a binary image B, places the origin of structuring element S over every pixel position, and ORs a binary 1 into that position of the output image, if and only if every position of S (with a 1) covers a 1 in B.

Erosion

- **Erosion** is a basic morphological operation



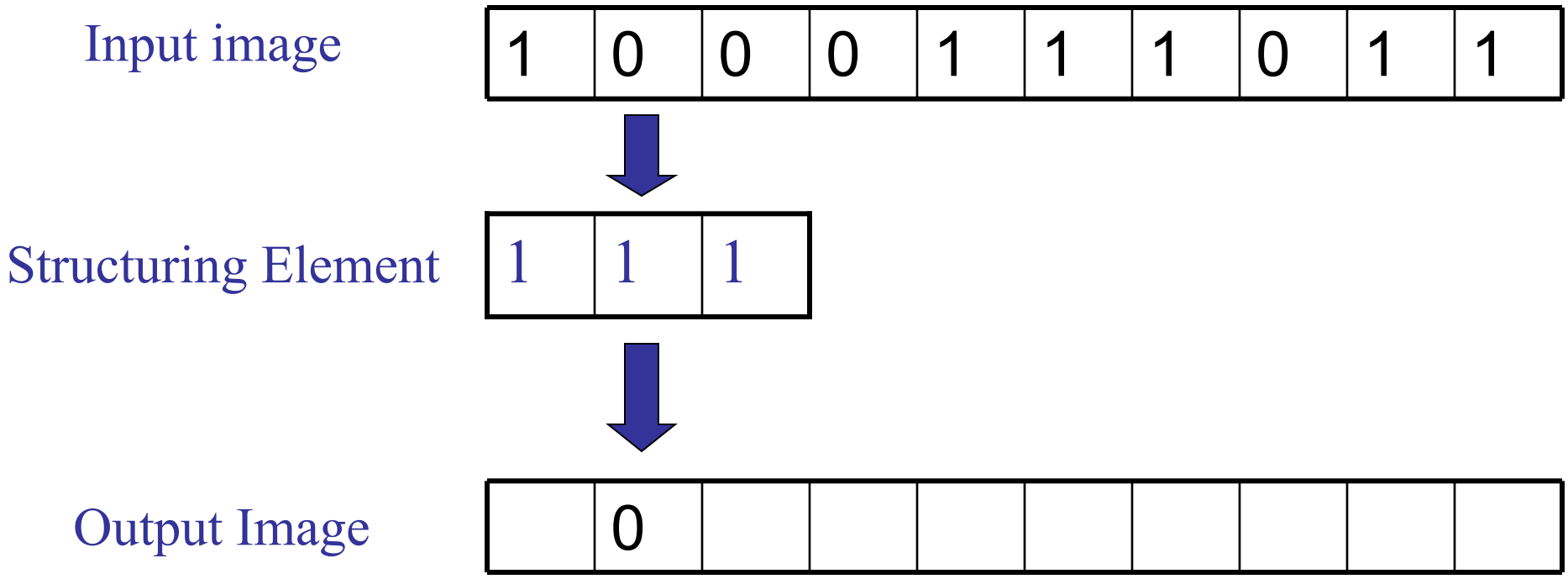
- Applied **Structuring Element**:

1	1	1
1	1	1
1	1	1

Set of coordinate points =

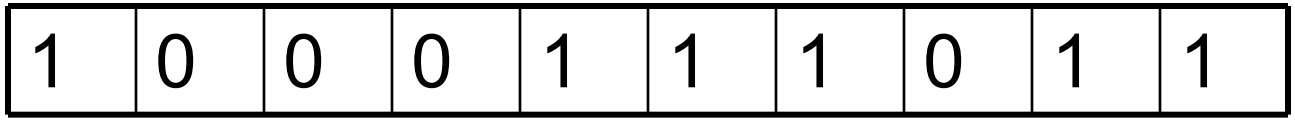
{ (-1, -1), (0, -1), (1, -1),
(-1, 0), (0, 0), (1, 0),
(-1, 1), (0, 1), (1, 1) }

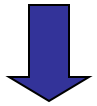
Example for Erosion



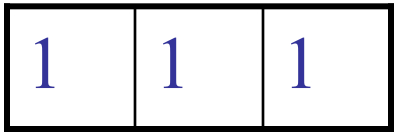
Example for Erosion

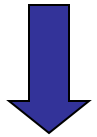
Input image



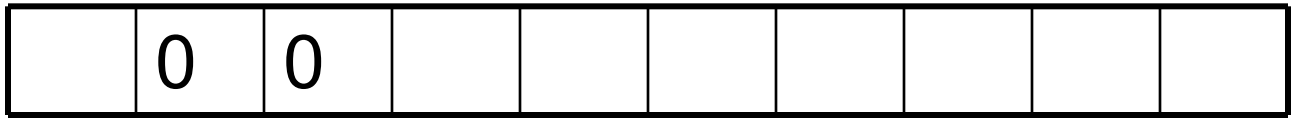


Structuring Element

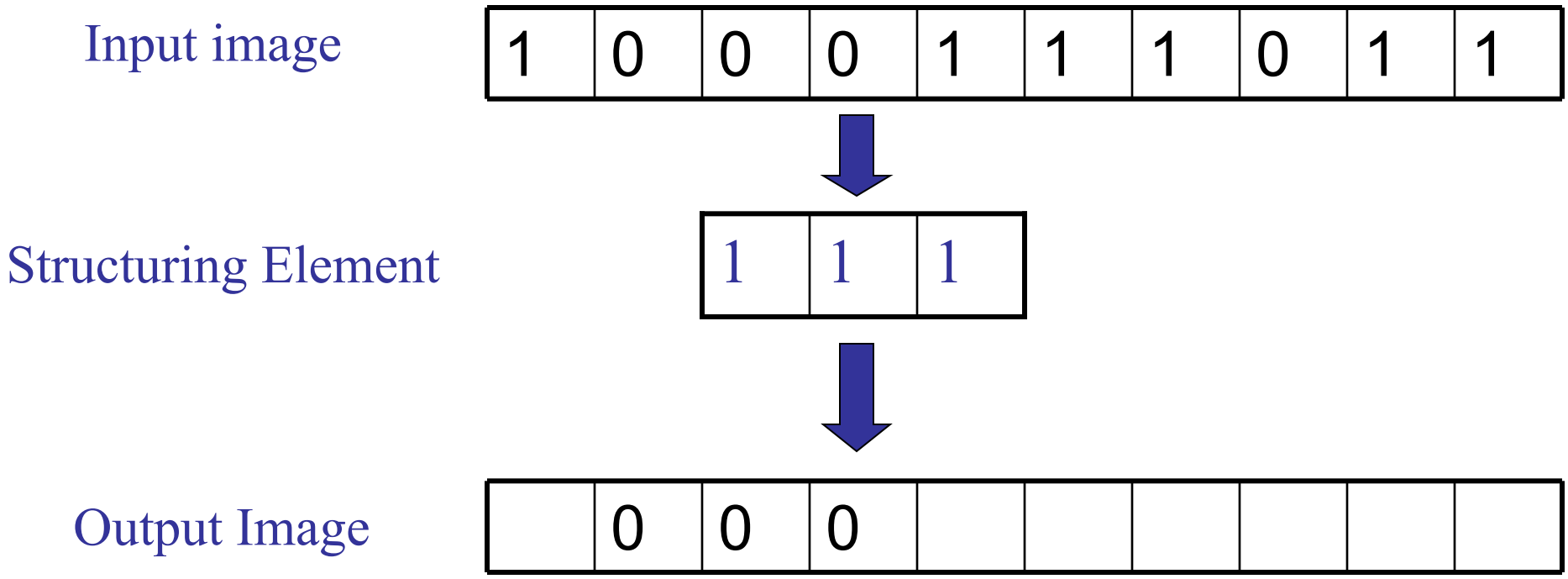




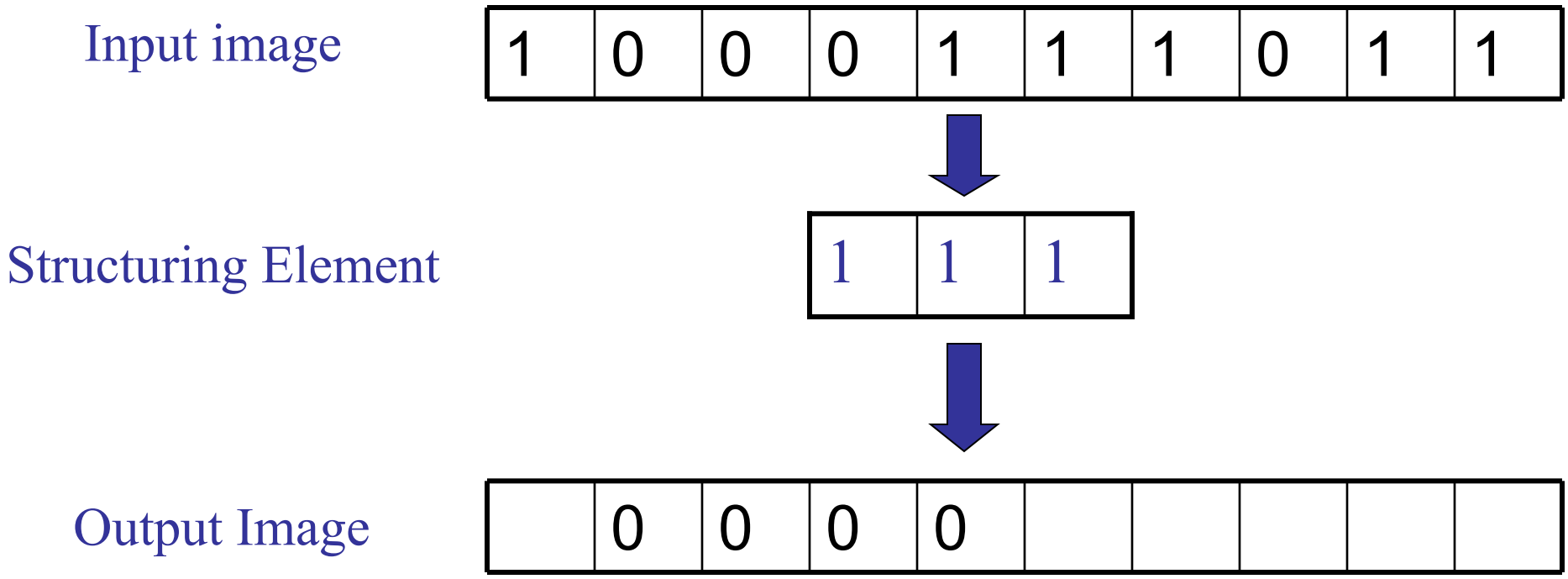
Output Image



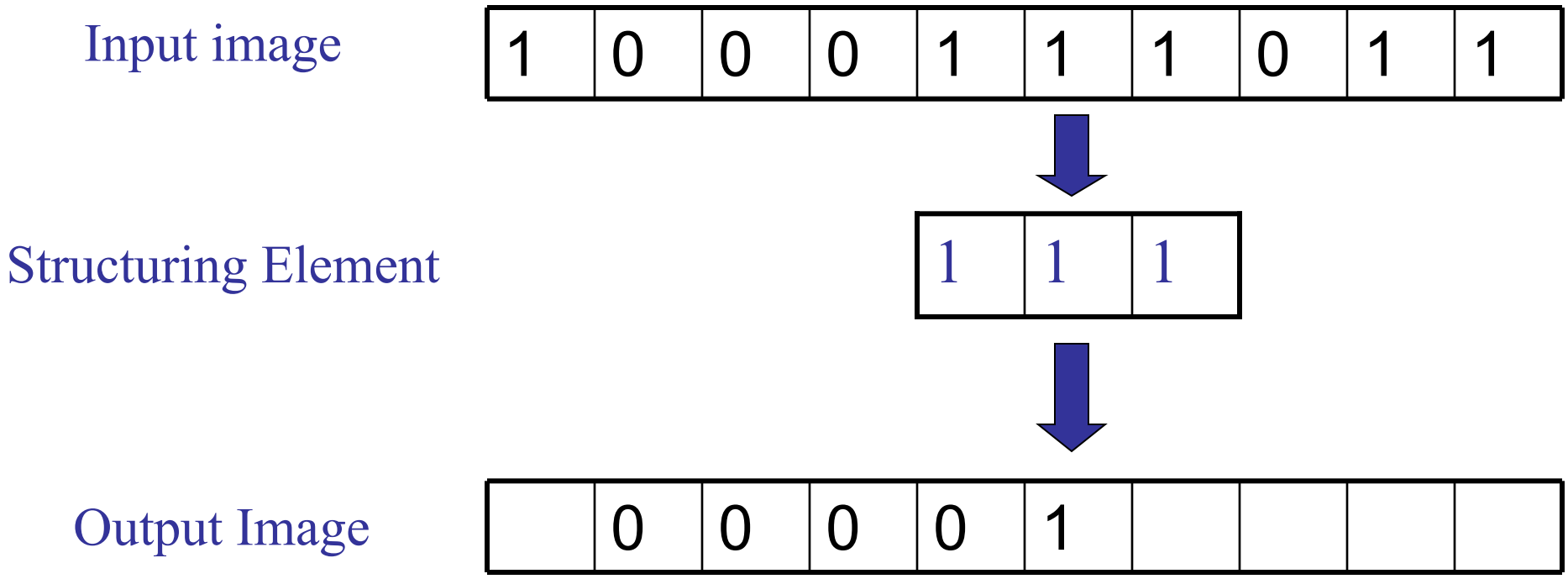
Example for Erosion



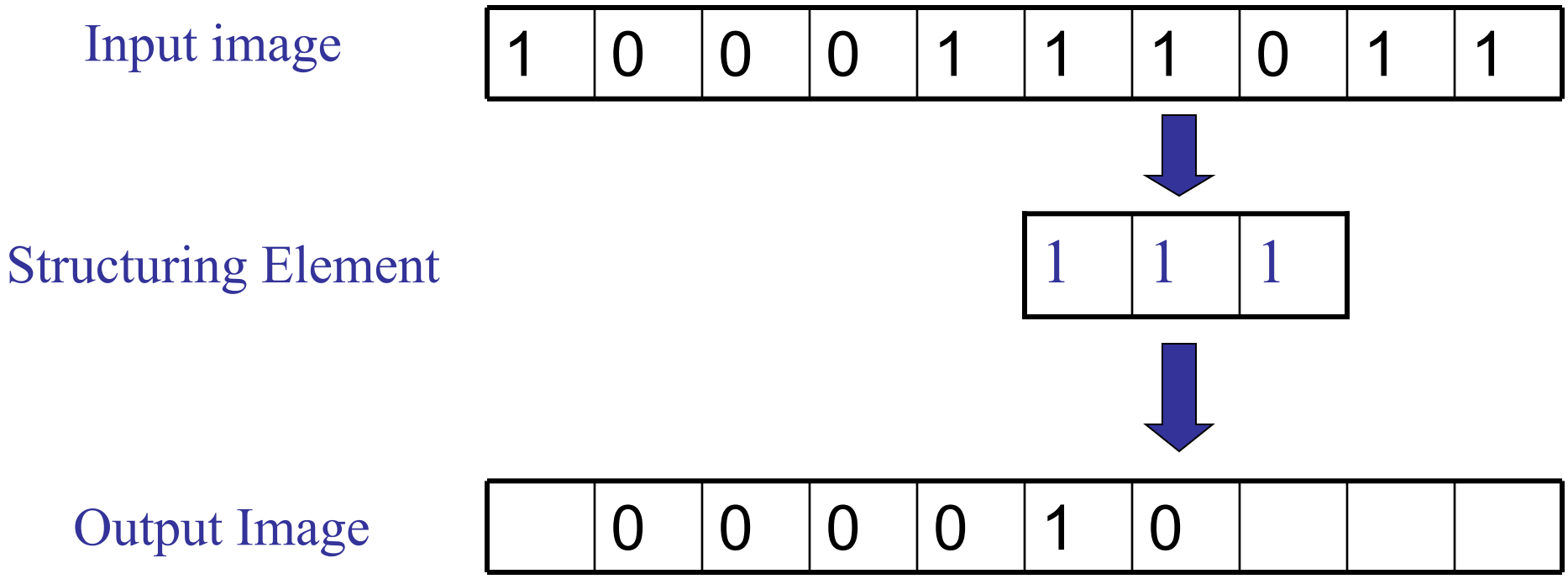
Example for Erosion



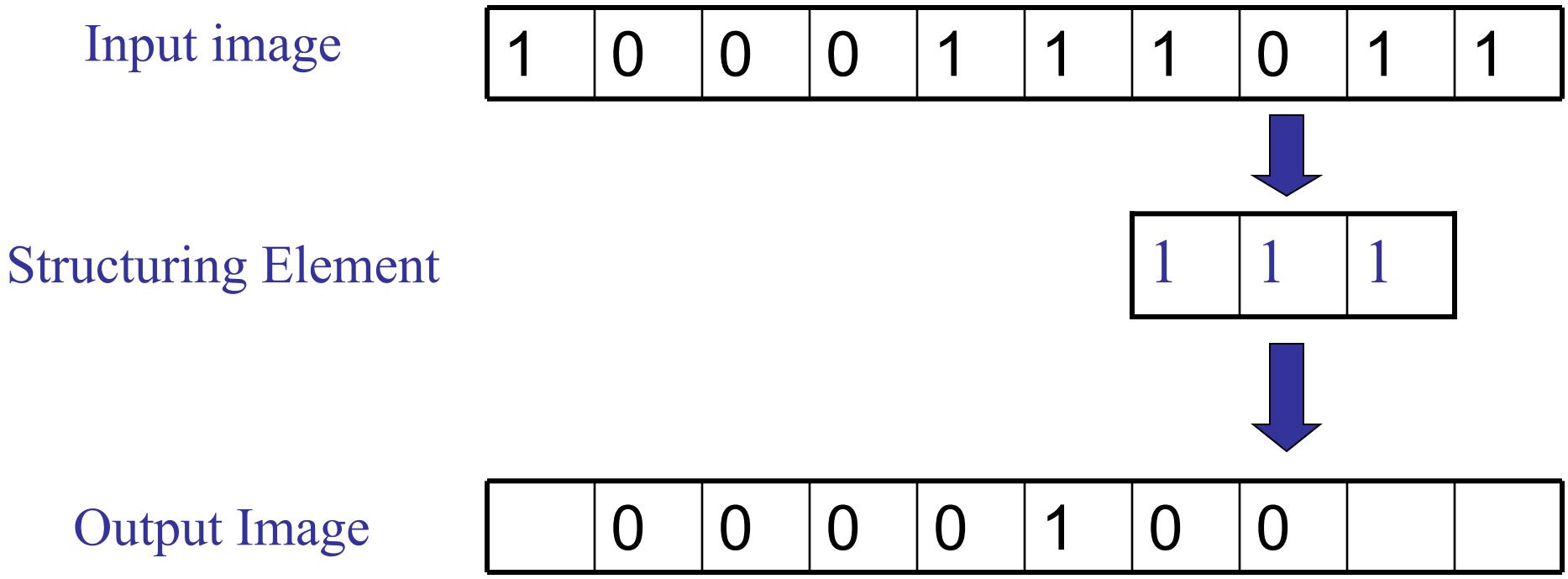
Example for Erosion



Example for Erosion

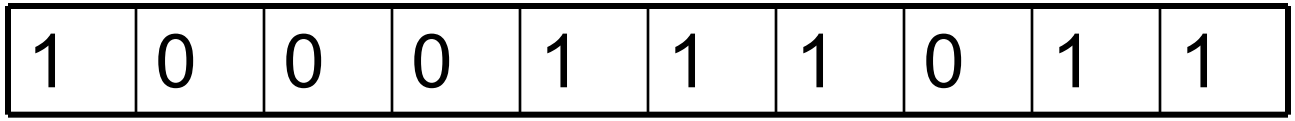


Example for Erosion

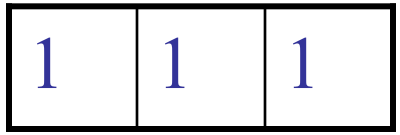


Example for Erosion

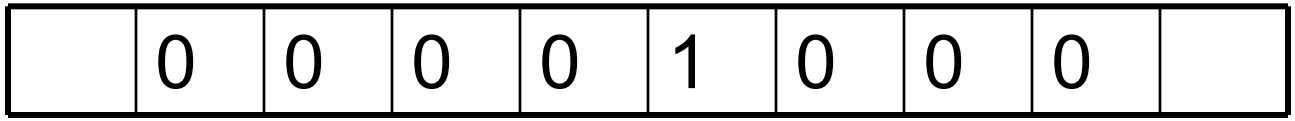
Input image



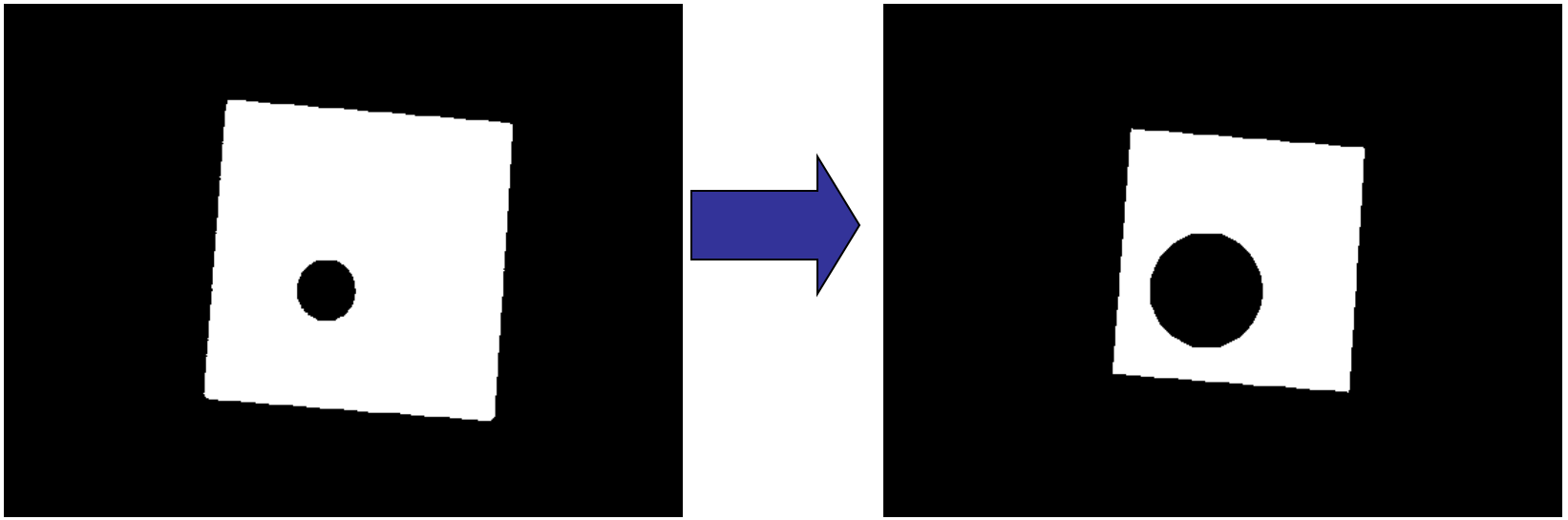
Structuring Element



Output Image



An example of erosion

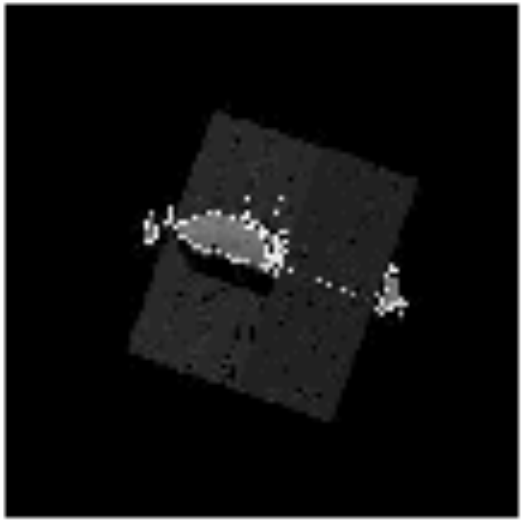


Binary image contour extraction

image

```
Erosed=erode(image);
Contour=
  double(image)-double(Erosed);
Contour=~Contour;
```

Contour



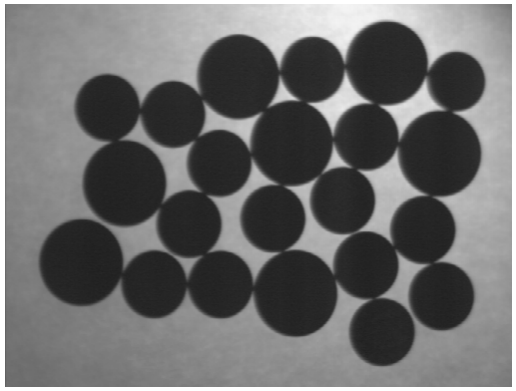
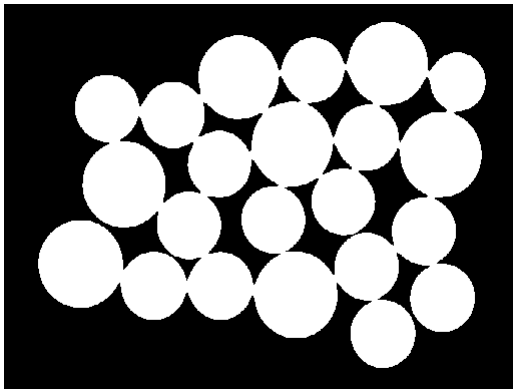
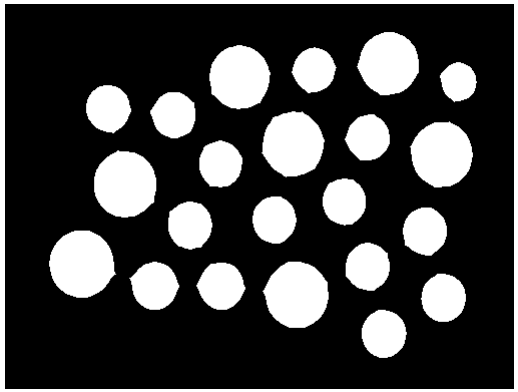
Satellite image with contour.

Erosion can be used to find contour

Dilation can be also used for it.

Counting Coins

- Counting coins is difficult because they touch each other!
- Solution: Binarization and Erosion separates them!

OPENING

Opening & Closing

- Important operations
- Derived from the two basic operations
  - Dilation
  - Erosion
- Usually applied to binary images.
- Opening and closing are dual operations

Opening & Closing

- Opening is the *dual* of closing:
-
  - *i.e.* opening the foreground pixels with a particular structuring element
  - is equivalent to closing the background pixels with the same element.

Opening

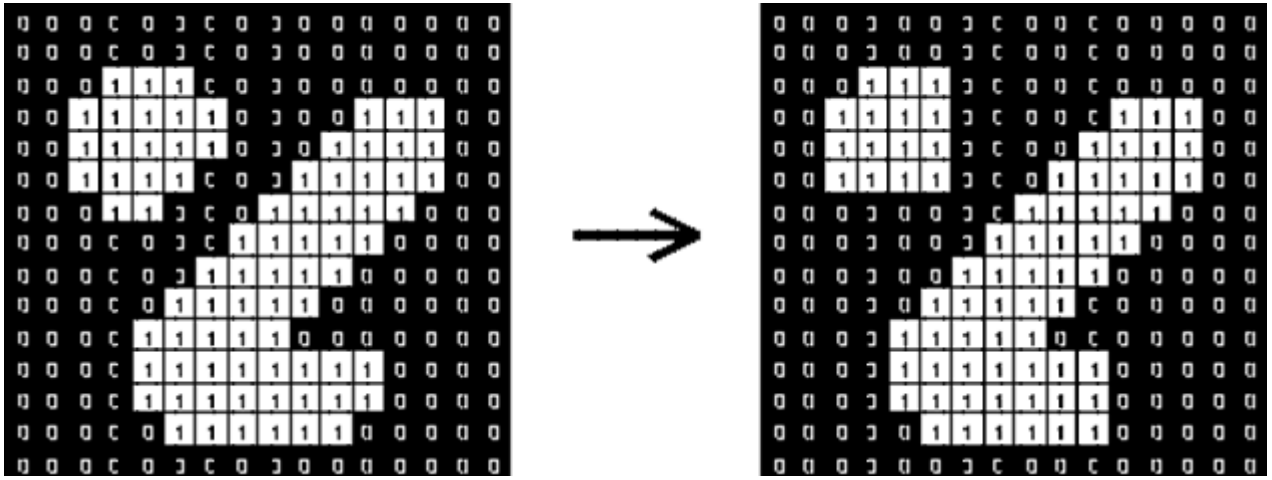
- Similar to Erosion
  - Spot and noise removal
  - Less destructive
- First Erosion, followed by dilation
- the s*ame structuring element for both operations.*
- Input:
  - Binary Image
  - Structuring Element, containing only 1s!

Opening

- Opening is **idempotent:** Repeated application has no further effects!

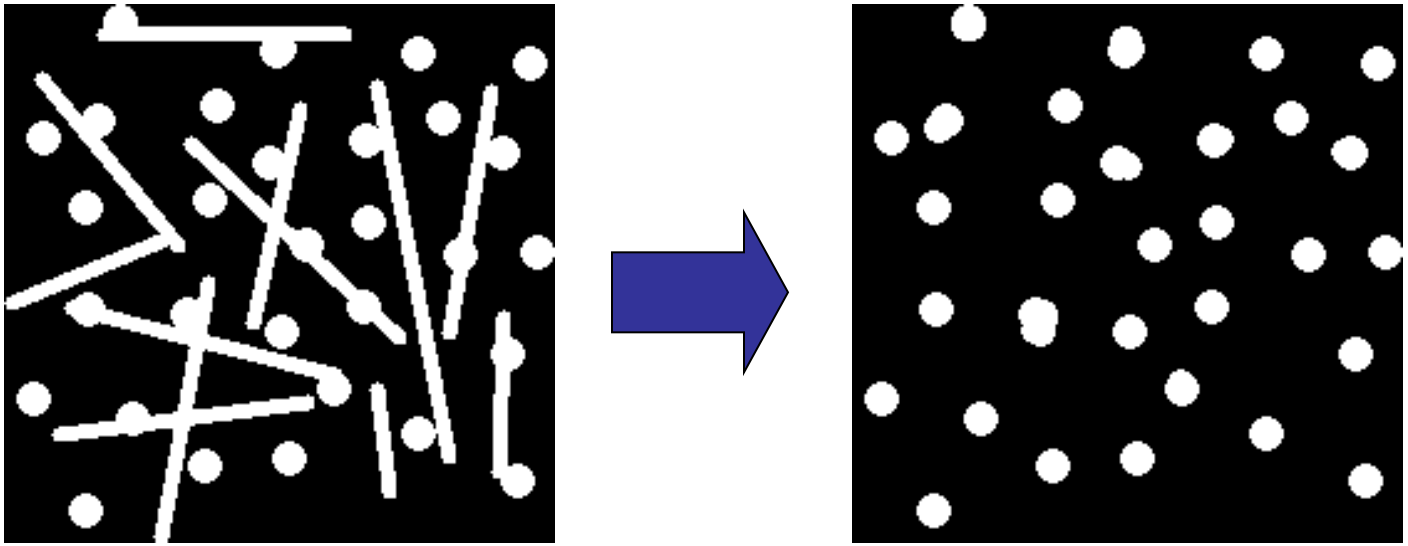
Opening

- Structuring element: 3x3 square



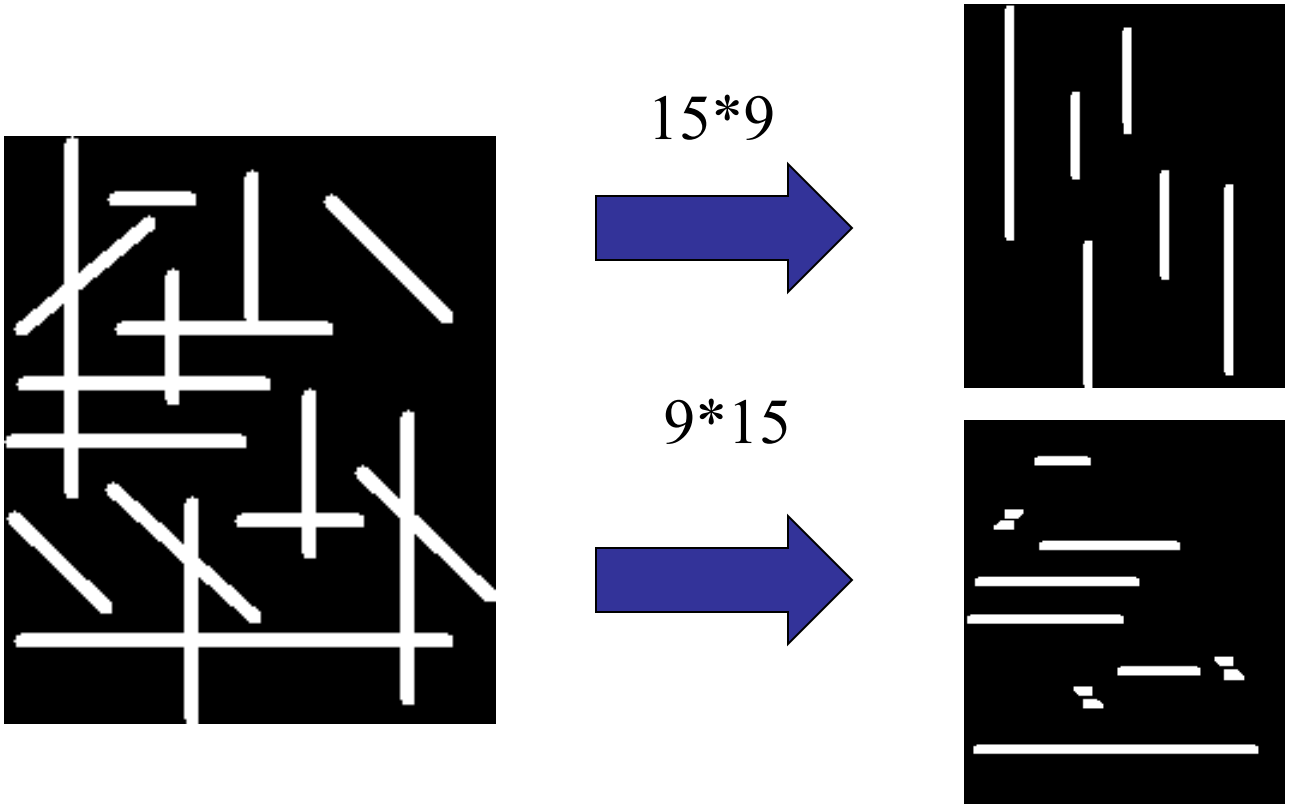
Opening Example

- Opening with a 11 pixel diameter disc



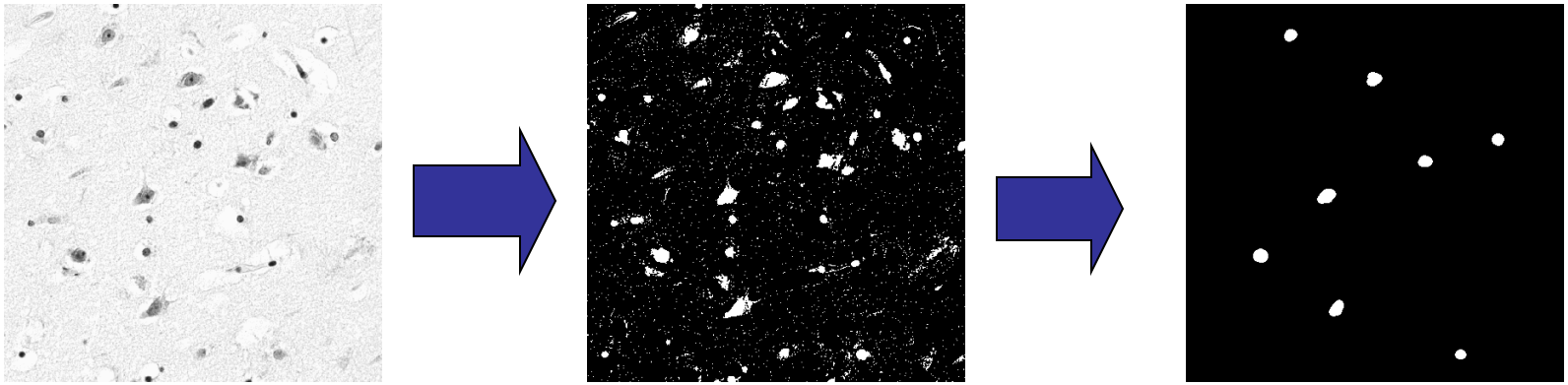
Opening Example

- 3x9 and 9x3 Structuring Element



Use Opening for Separating Blobs

- Use large structuring element that fits into the big blobs
- Structuring Element: 11 pixel disc



Use Opening to extract corners

Original

Opening

Corners

1. What kind of structuring element was used in the opening?
2. How did we get the corners?

CLOSING

Closing

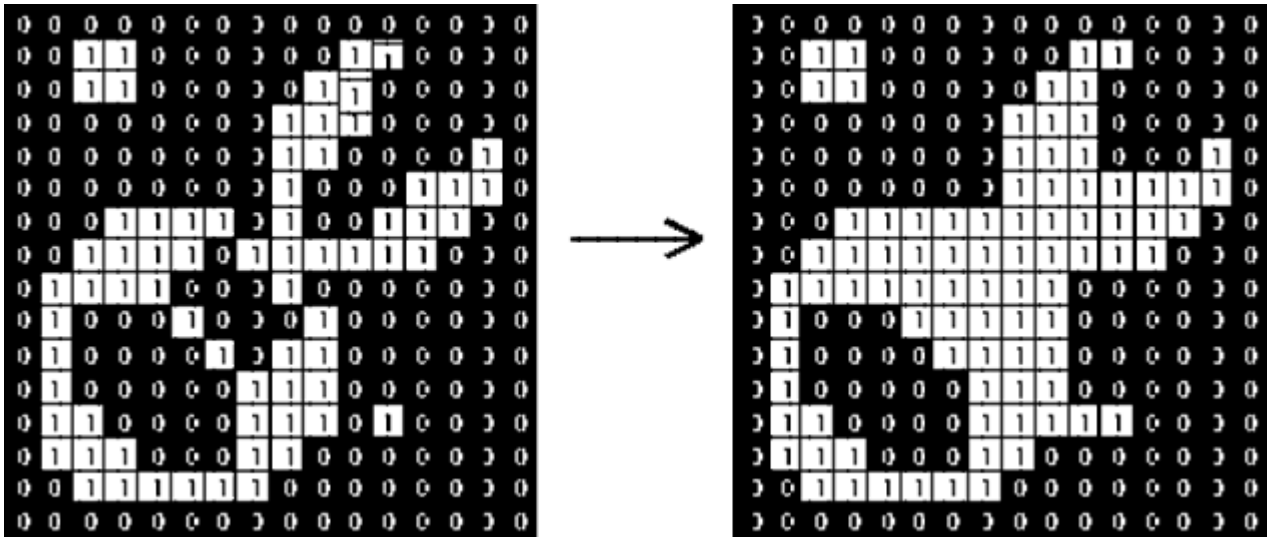
- Similar to Dilation
  - Removal of holes
  - Tends to enlarge regions, shrink background
- Closing is defined as a Dilatation, followed by an Erosion *using the same structuring element for both operations.*
- Dilation next erosion!
- Input:
  - Binary Image
  - Structuring Element, containing only 1s!

Closing

- Closing is **idempotent:** Repeated application of ‘closing’ has no further effects!

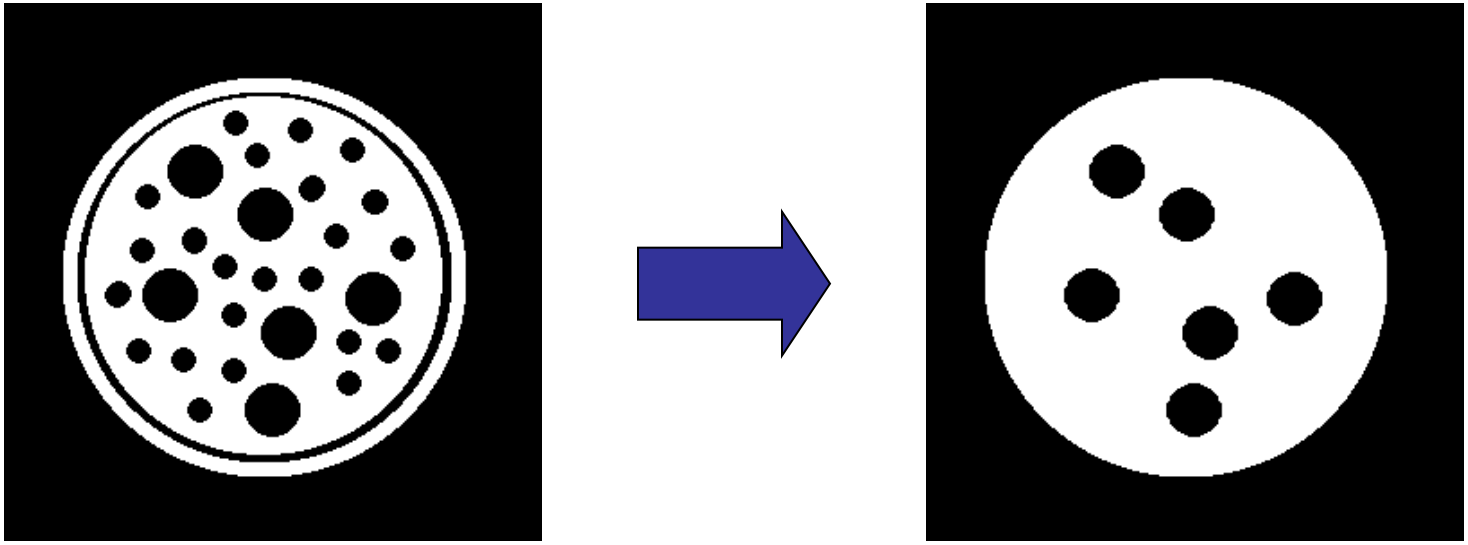
Closing

- Structuring element: 3x3 square



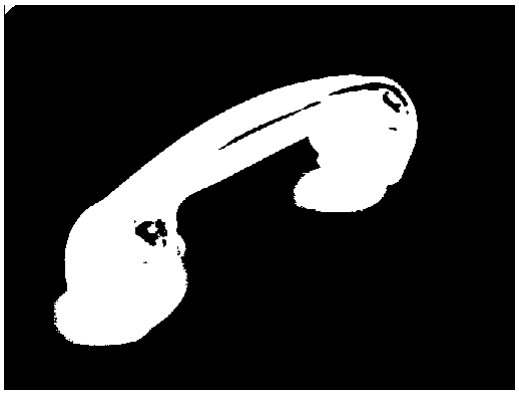
Closing Example

- Closing operation with a 22 pixel disc
- Closes small holes in the foreground

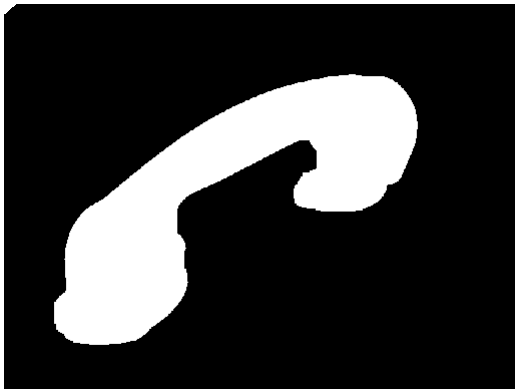


Closing Example

1. Threshold
2. Closing with disc of size 20



Threshold



closing

Reading

- Chapter 9.1- Chapter 9.3, in Digital Image Processing 2nd edition.