TDS3651
Visual Information Processing

Binary Image Processing
Lecture 5

Faculty of Computing and Informatics
Multimedia University

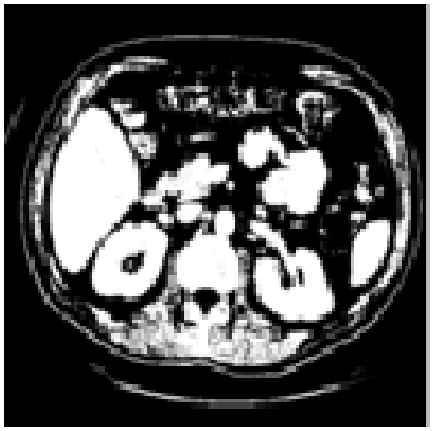
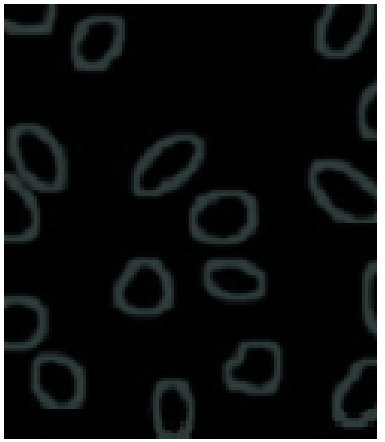
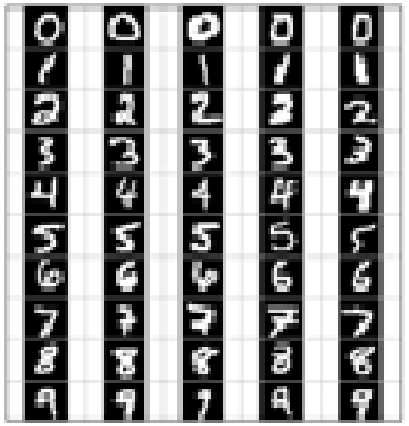
Previously created by John See, edited by Lai-Kuan, Wong, Yuen Peng, Loh
Some graphics and materials are taken from UTexas at Austin CS376 & CS381V notes, Stanford Uni's CS131 notes and Gonzalez & Wood's DIP textbook

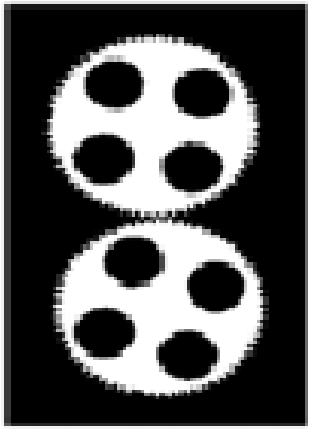
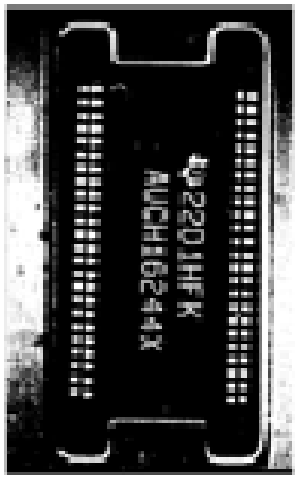
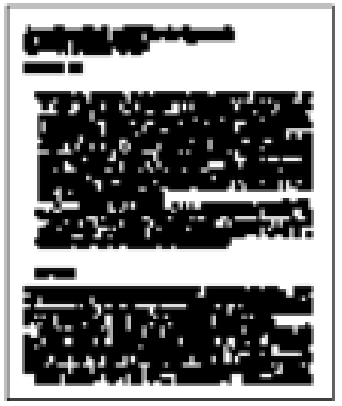
Lecture Outline

- Thresholding
- Morphological Operations
- Blob Extraction (and labelling)
- Blob Description

What are binary images?
And, are they of any good use?

Binary Images

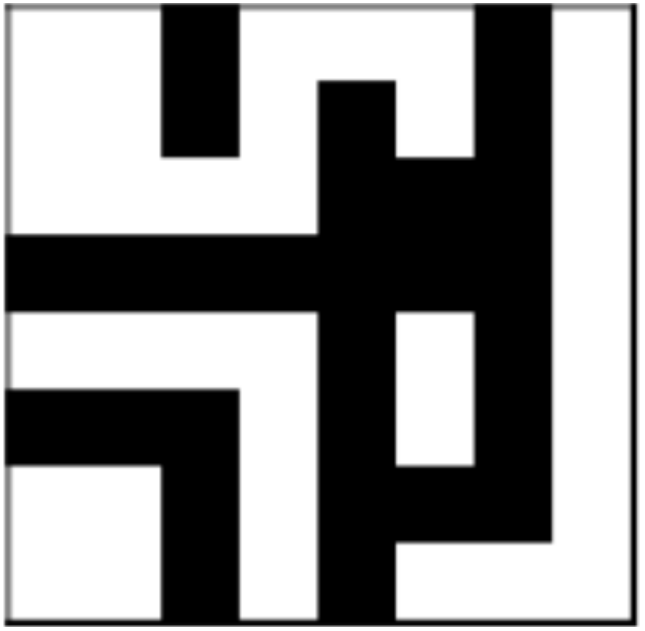
Binary image processing: Tasks

- Convert the image into binary form
  - Thresholding
- Clean up the thresholded image
  - Morphological operators
- Extract separate blobs
  - Connected components
- Describe the blobs with region properties

Binary images

- **Two** pixel values
  - Foreground and background
  - Mark region(s) of interests (ROI)

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



Thresholding

- **Grayscale image ⇒ Binary image**
- Useful if object of interest's intensity distribution is distinct from background
- Here are some scenarios…

$$F_T(i,j) = \begin{cases} 1 & if\ F(i,j) \geq T \\ 0 & otherwise \end{cases}$$

$$F_T(i,j) = \begin{cases} 1 & if\ T_1 \leq F(i,j) \leq T_2 \\ 0 & otherwise \end{cases}$$

$$F_T(i,j) = \begin{cases} 1 & if\ F(i,j) \in Z \\ 0 & otherwise \end{cases}$$

Thresholding: Uses

- Given a grayscale image or an intermediate matrix of values ⇒ **threshold** to create a **binary output**

Edge detection

Gradient magnitude

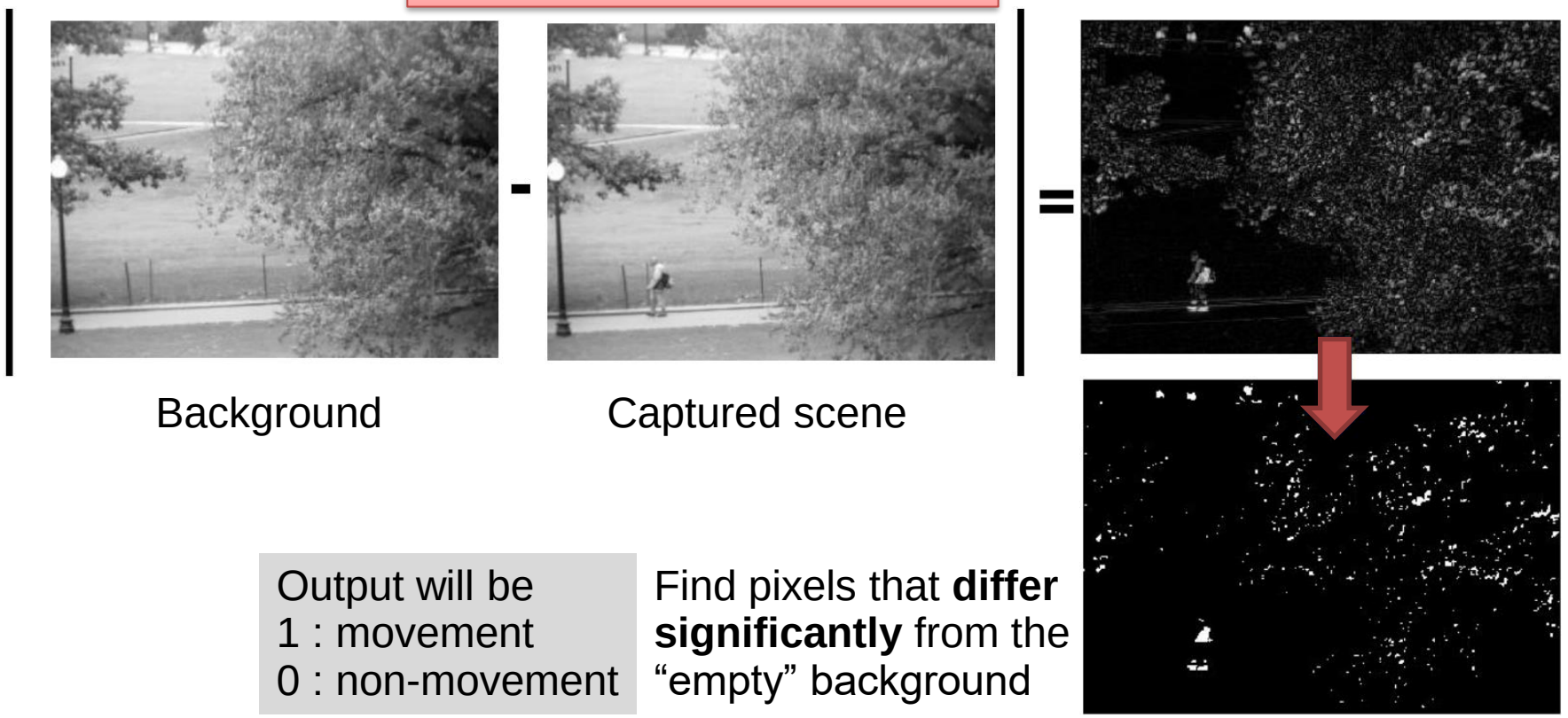
Find strong gradient that are above a certain **threshold**

Output will be
1 : edge
0 : non-edge

Thresholding: Uses

- Given a grayscale image or an intermediate matrix of values ⇒ **threshold** to create a **binary output**

Background Subtraction



Background

Captured scene

Output will be
1 : movement
0 : non-movement

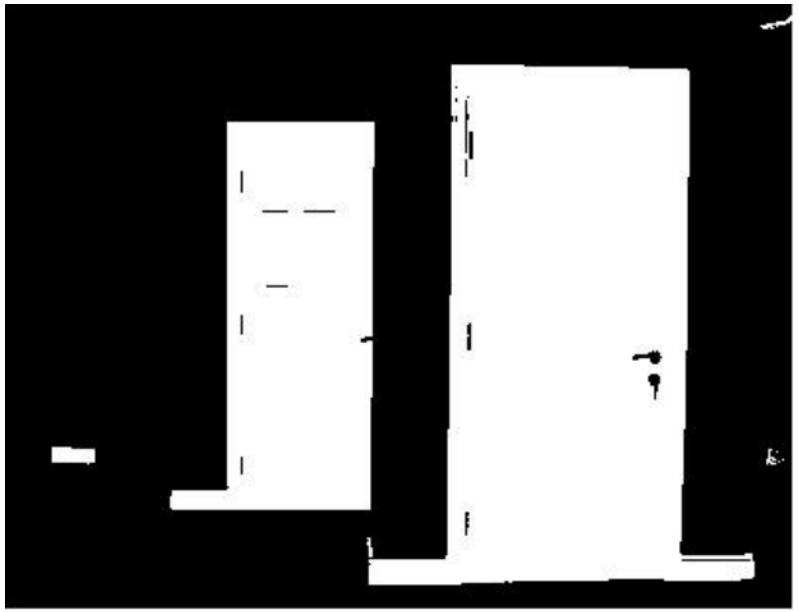
Find pixels that **differ significantly** from the "empty" background

Thresholding: Uses

- Given a grayscale image or an intermediate matrix of values ⇒ **threshold** to create a **binary output**

Intensity-based Detection

Image



Find pixels that are dark, **corresponding to the detection** of dark-intensity doors

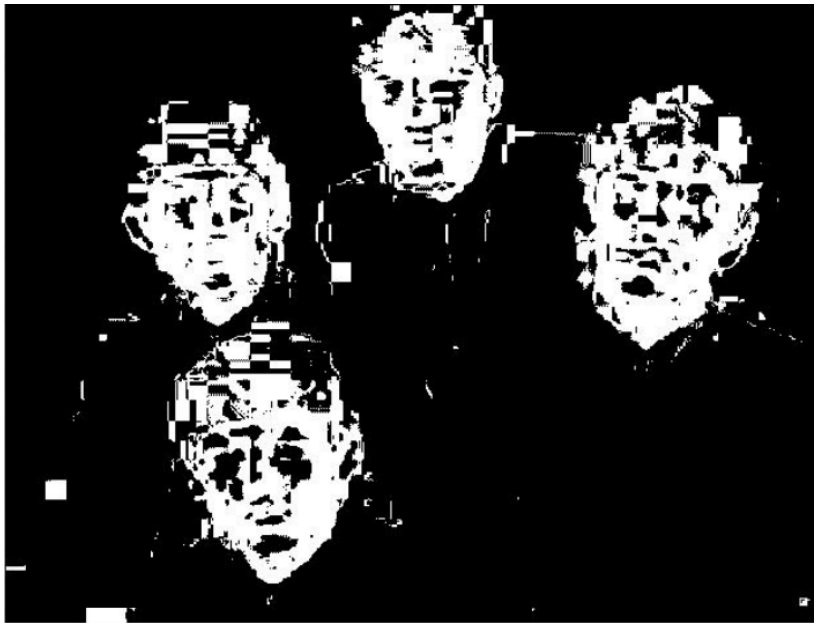
Output will be
1 : object of interest
0 : background

Thresholding: Uses

- Given a grayscale image or an intermediate matrix of values ⇒ **threshold** to create a **binary output**

Colour-based Detection

Image



Find pixels that are **within a certain hue (colour)** range, e.g. skin colour

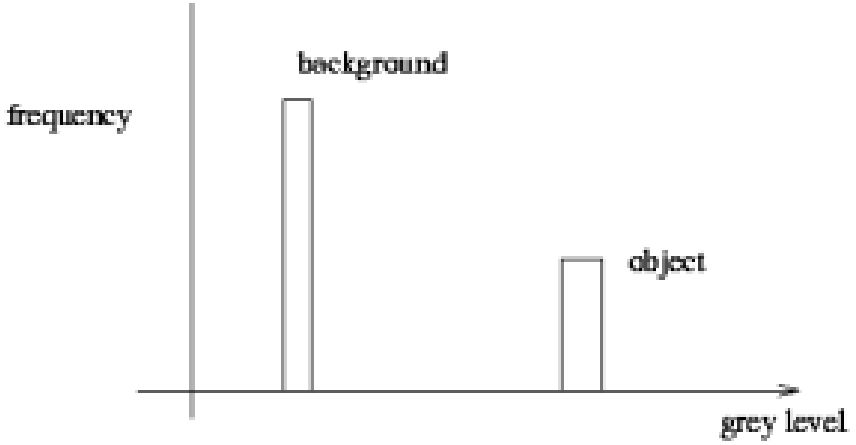
Output will be
1 : object of interest
0 : background

What to use?

- What is a suitable information to use to convert this gif into a binary image/clip of only the dog?

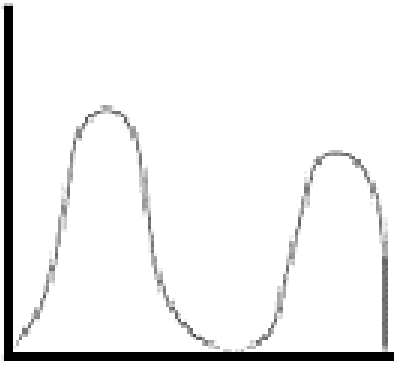
A. Edges
B. Background
C. Color
D. All of the above

Histograms: Nice case

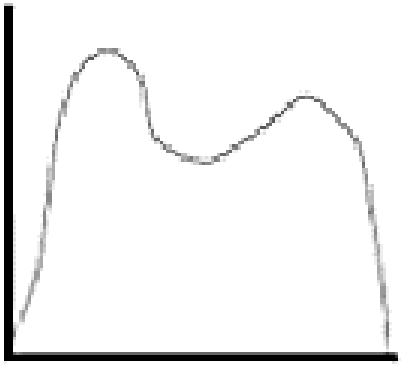


Ideal histogram: light object on a dark background

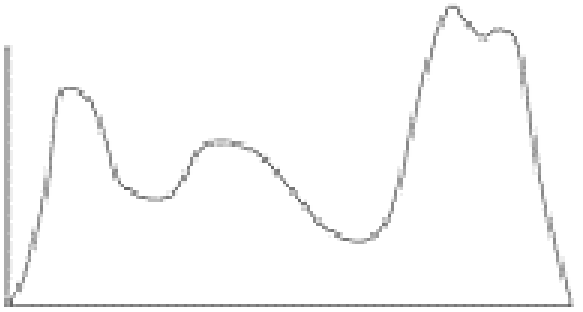
Histograms: Not-so-nice case



Two distinct modes
Easy



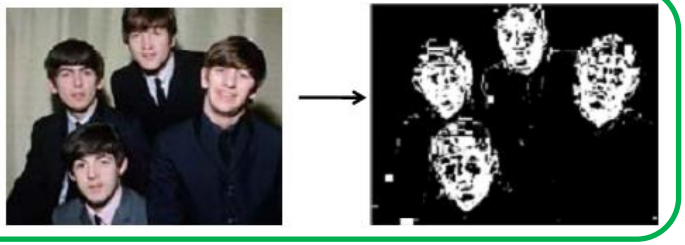
Overlapping modes
Challenging!



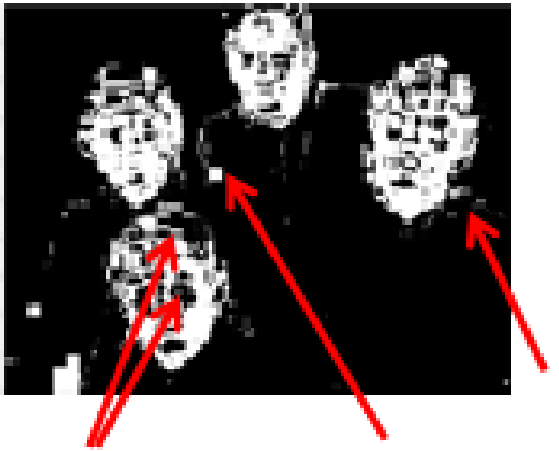
…

Two tasks lie ahead…

Revisiting this task ⇒



1. What to do with “noisy” binary outputs?



2. How to demarcate (mark) multiple regions of interest?

Dealing with noisy binary output?

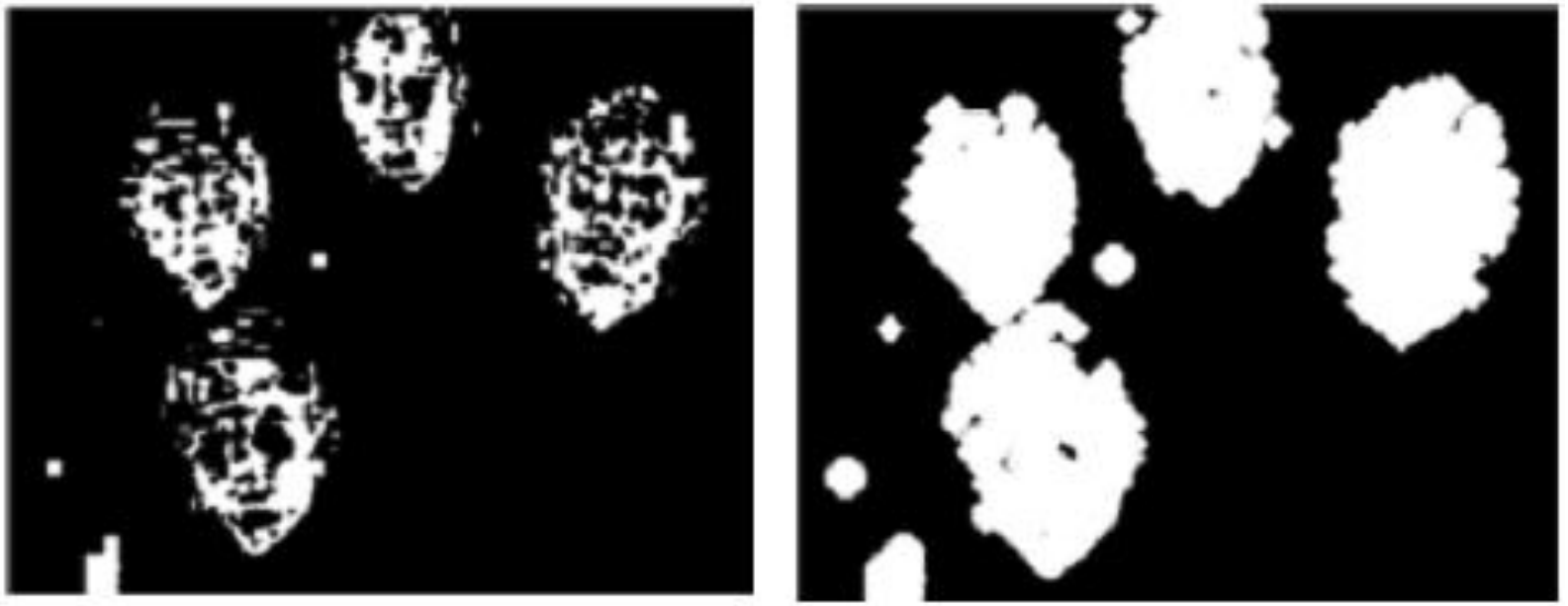
Morphological operations

Morphological operators

- Change the shape of the foreground regions via intersection/union operations between a scanning structuring element and binary image
- Useful for cleaning up result from thresholding
- Basic operators:
  - **Dilation**
  - **Erosion**

Dilation

- Expands connected regions
- “Grow” pixels
- Fill holes



Before dilation

After dilation

Erosion

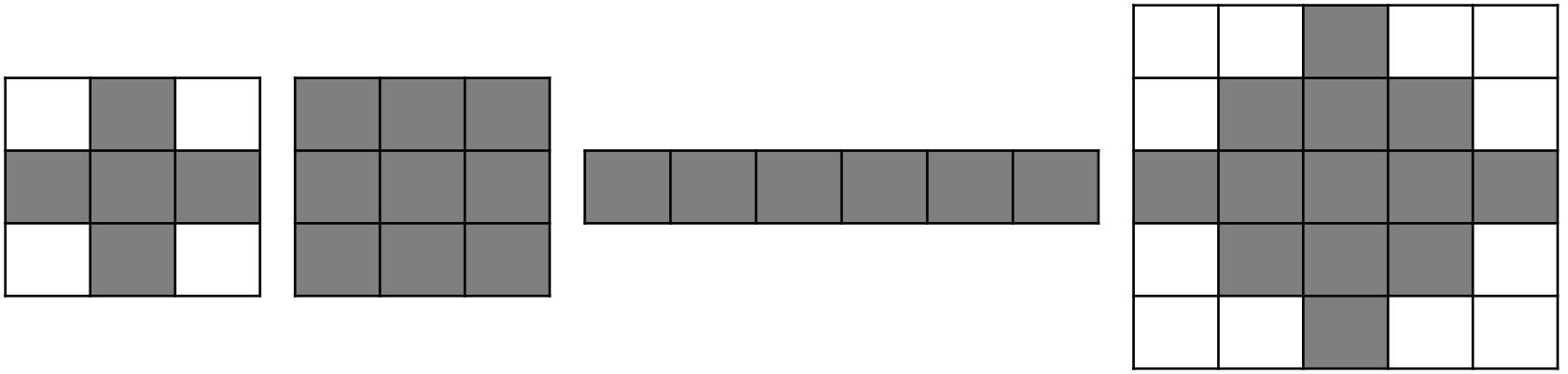
- Erodes connected regions
- “Trim” pixels
- Removes bridges, branches, noise

Before erosion

After erosion

Structuring elements

- Masks of varying shapes and sizes are used for morphological operations:

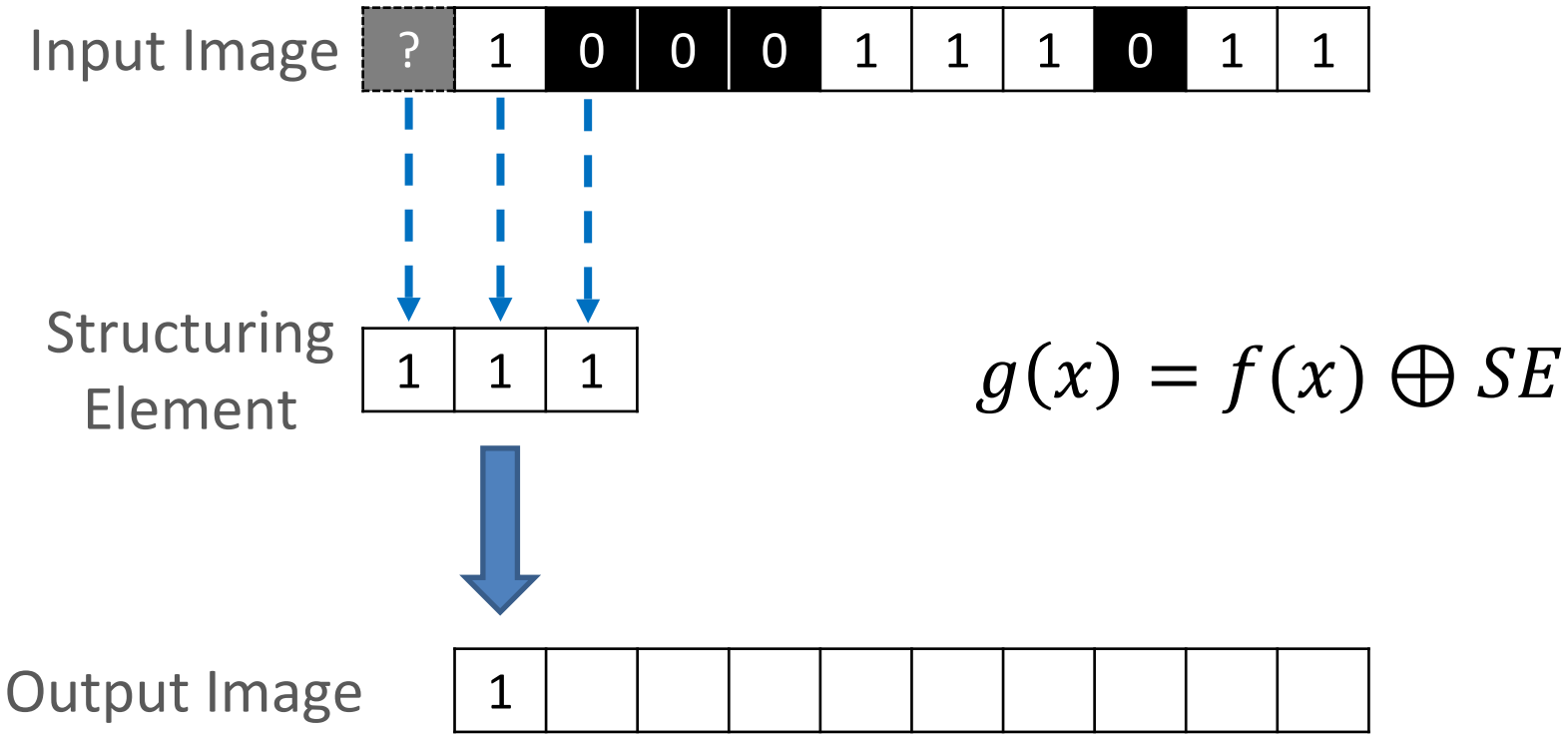


- Scan mask across foreground pixels to transform the binary image

Dilation vs. Erosion

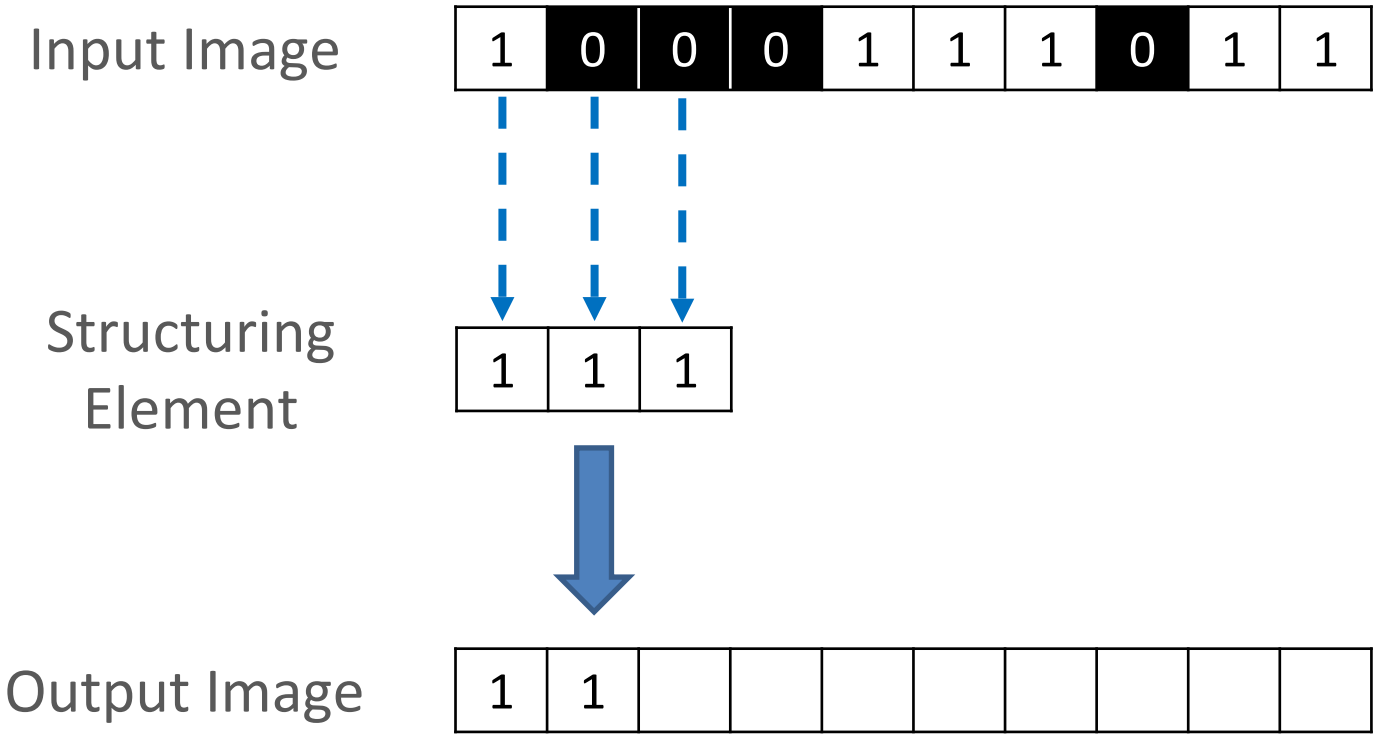
- At each position:
  - **Dilation:** If current pixel is foreground, **OR** the structuring element with the input image

1-D Example of Dilation

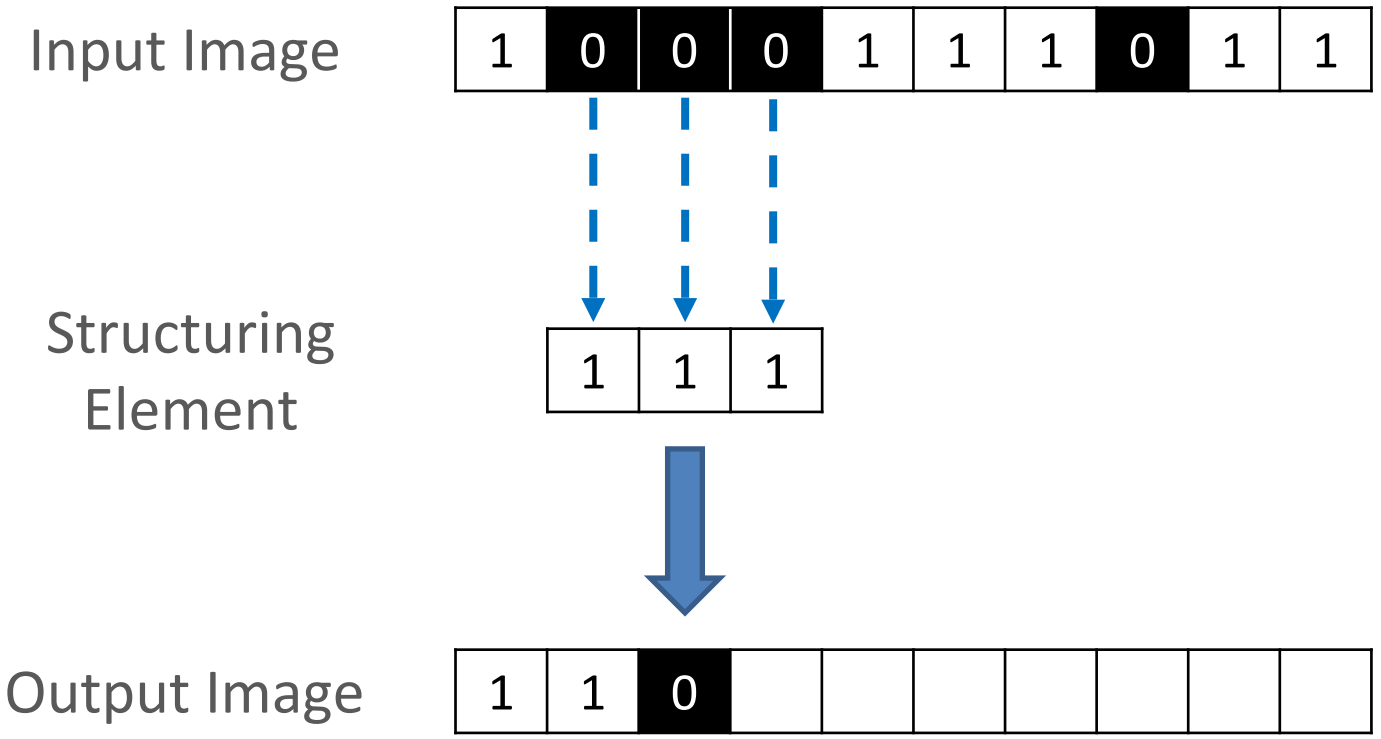


$$g(x) = f(x) \oplus SE$$

1-D Example of Dilation

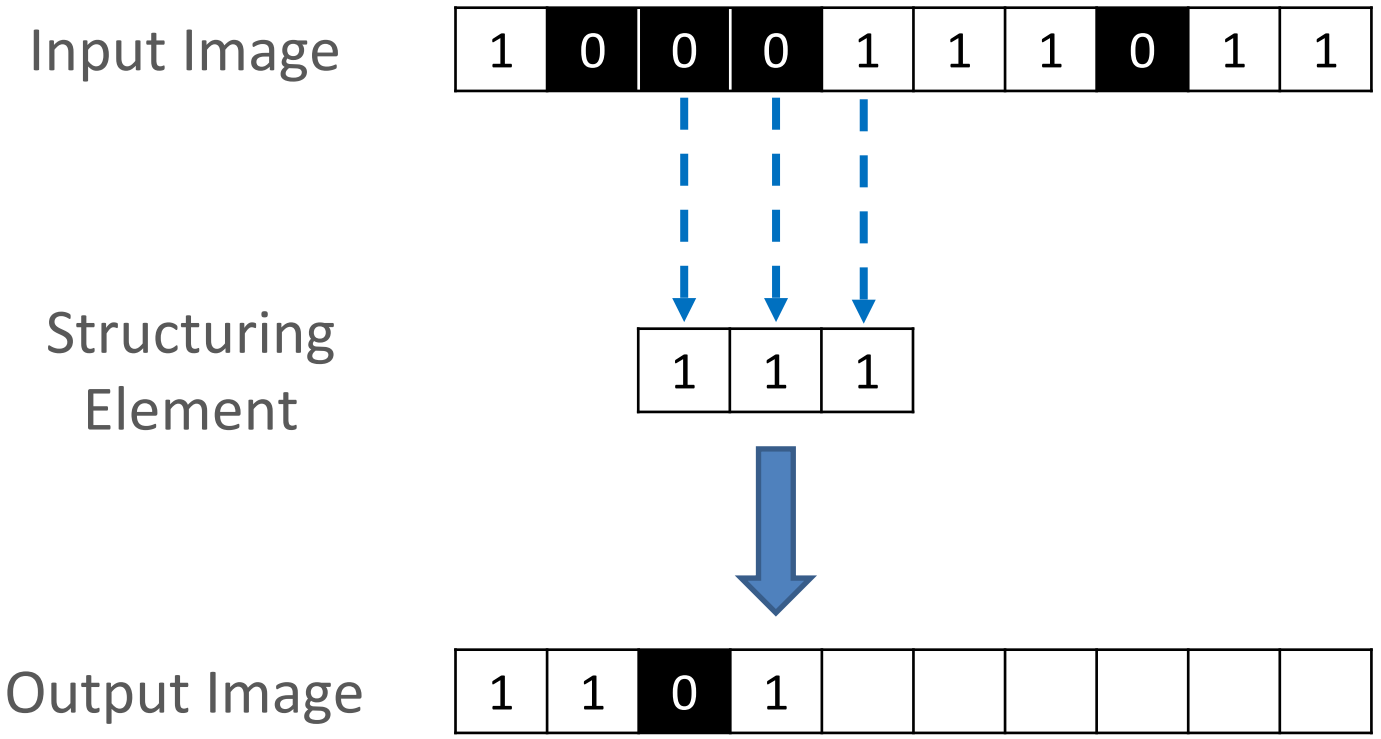


1-D Example of Dilation

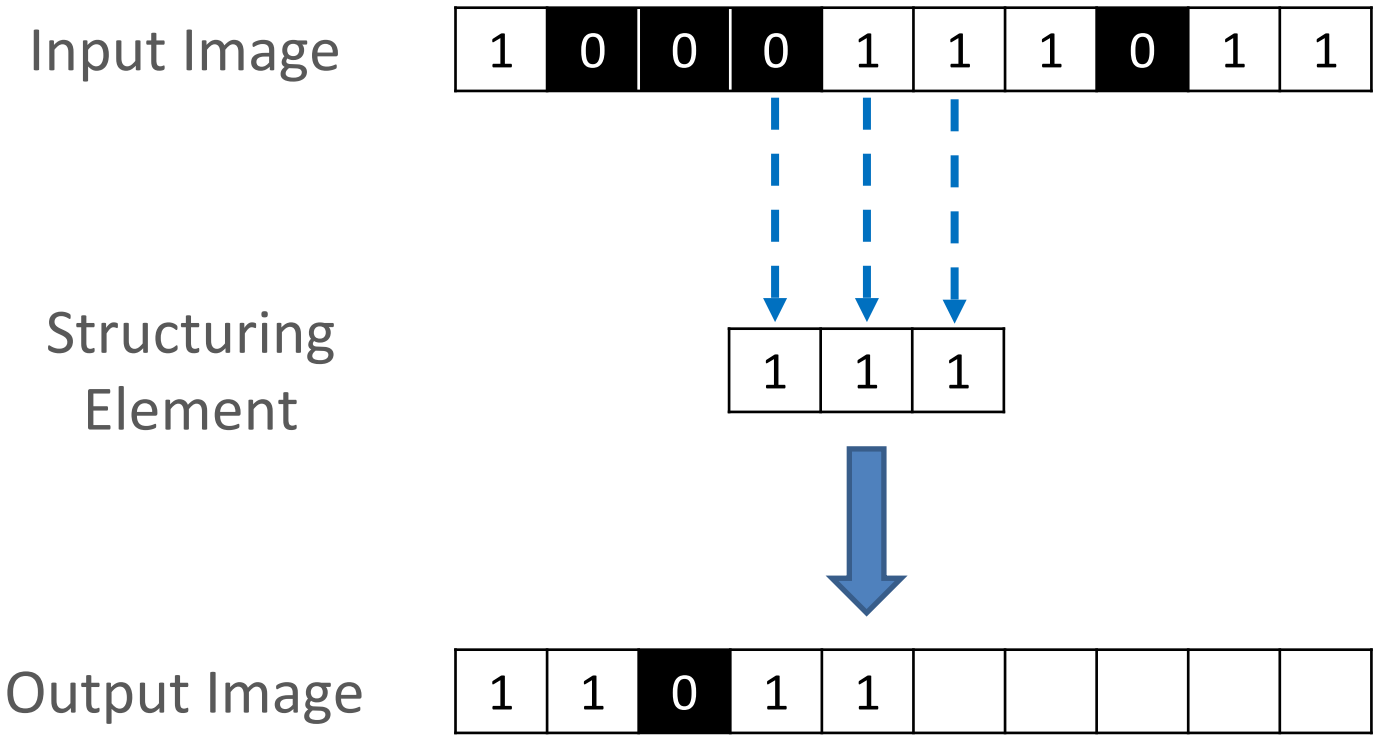


Remember: OR-ing only happens when there is a FOREGROUND PIXEL!

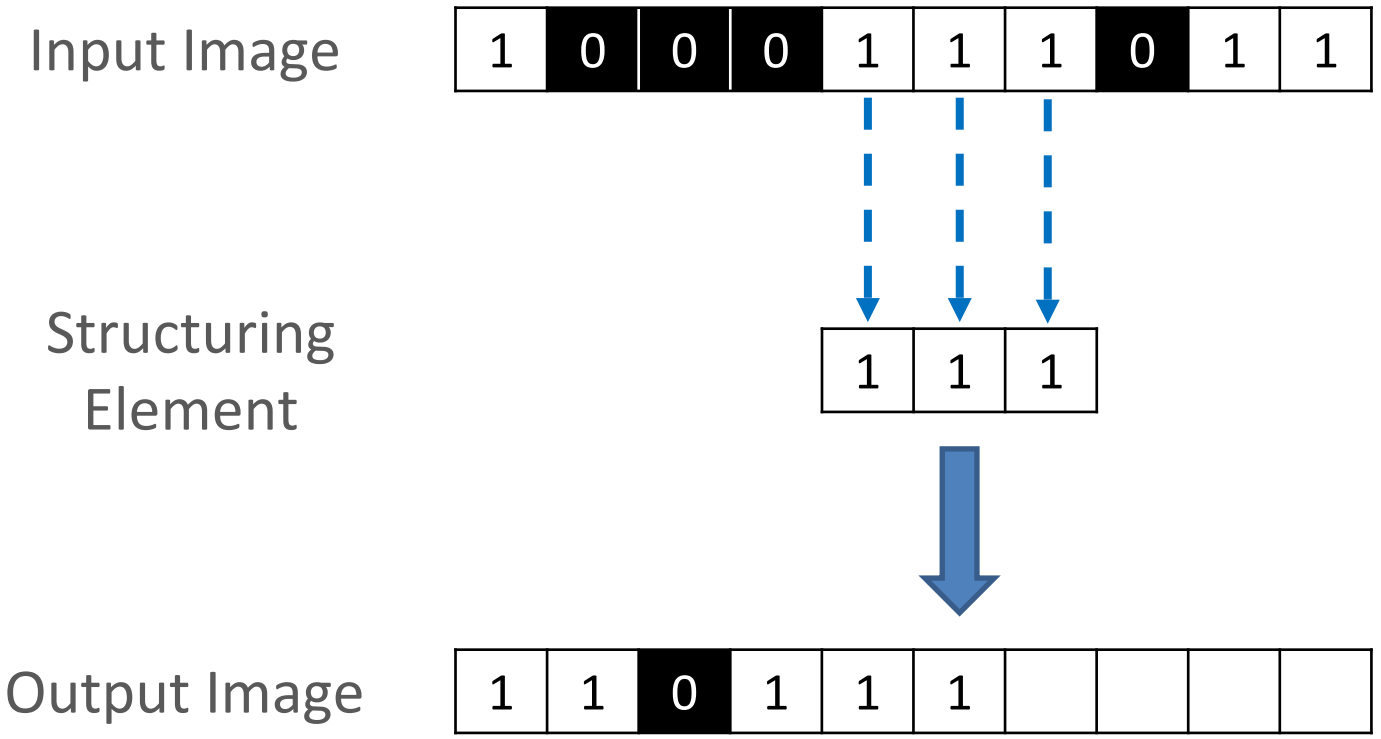
1-D Example of Dilation



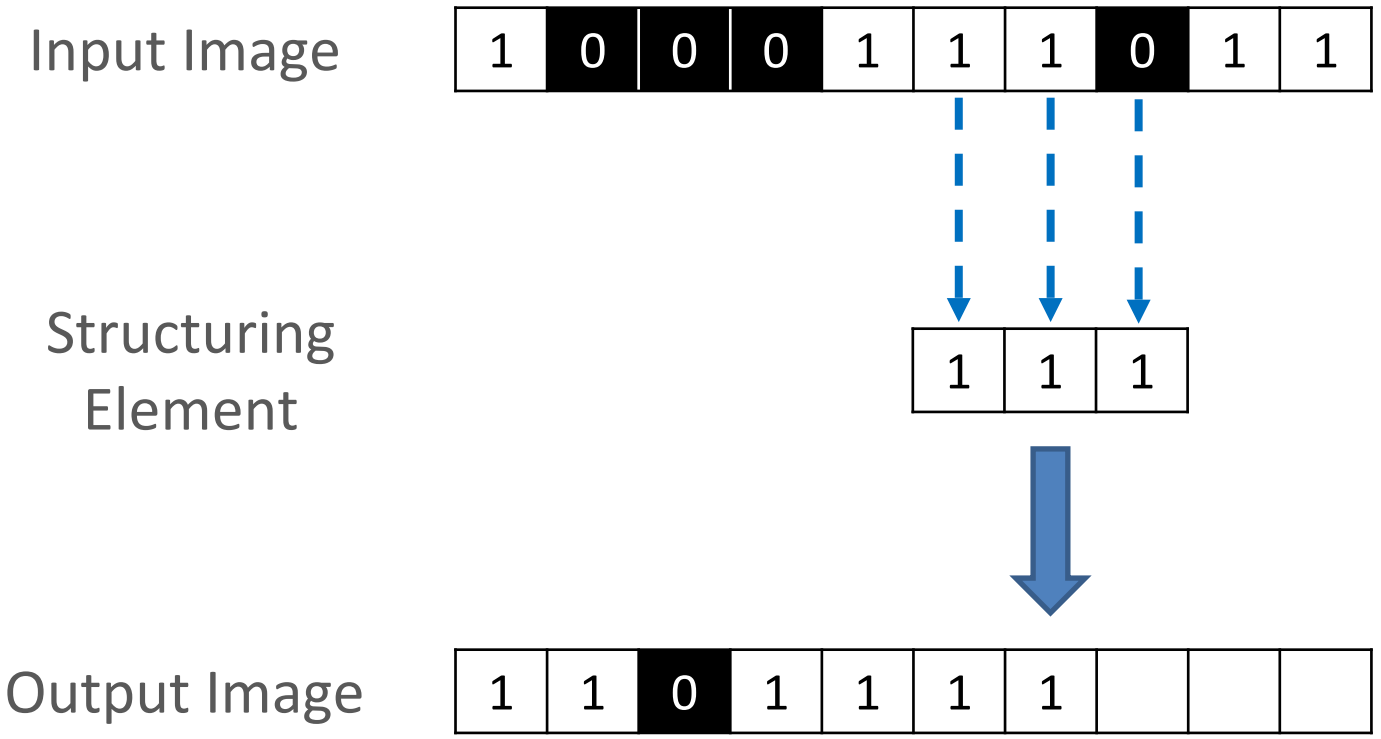
1-D Example of Dilation



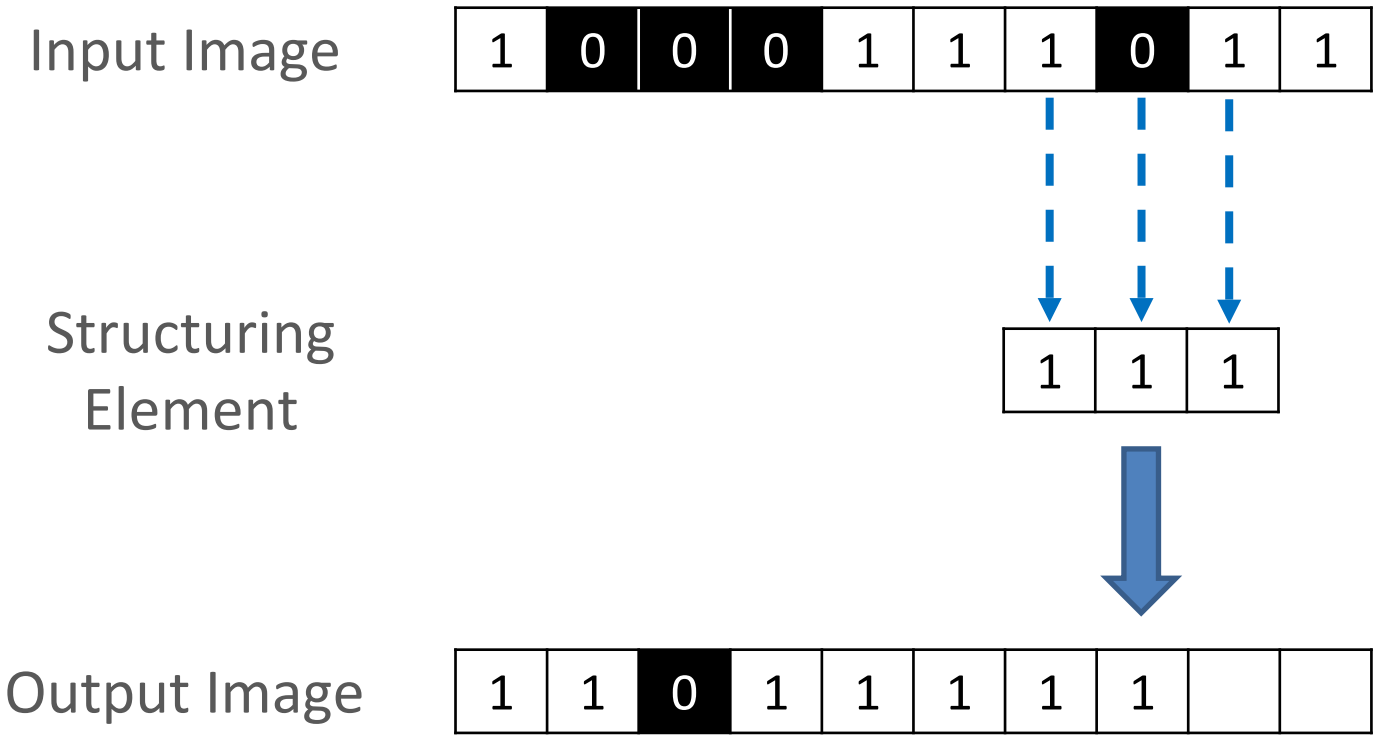
1-D Example of Dilation



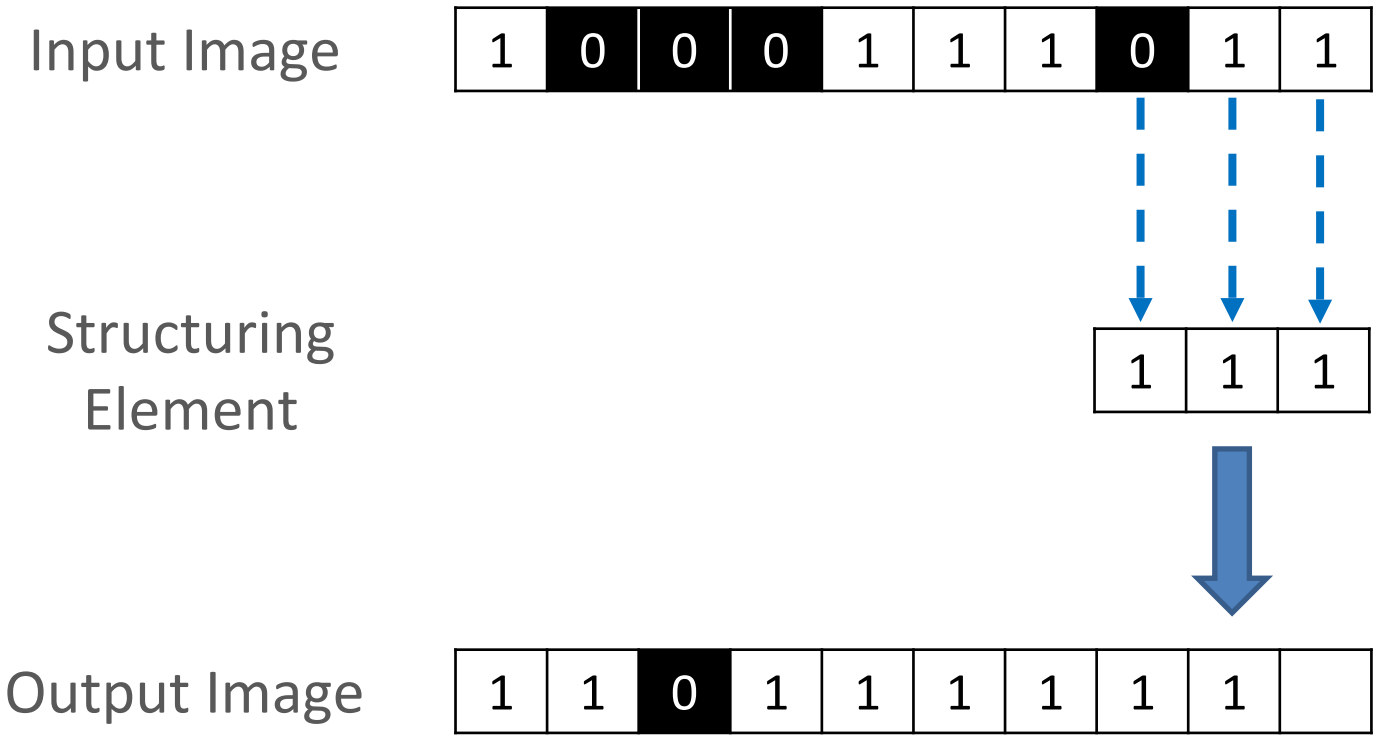
1-D Example of Dilation



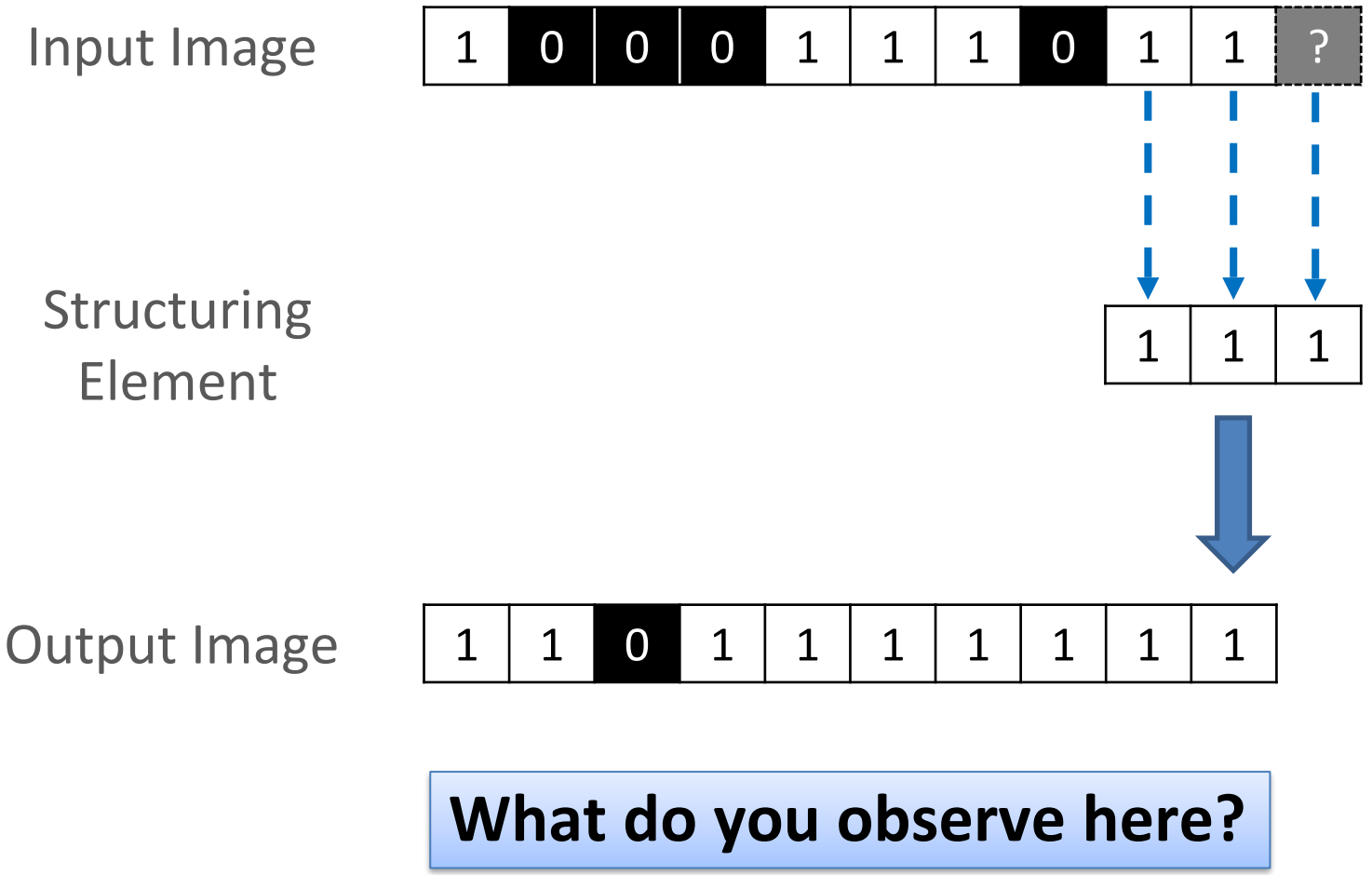
1-D Example of Dilation



1-D Example of Dilation



1-D Example of Dilation

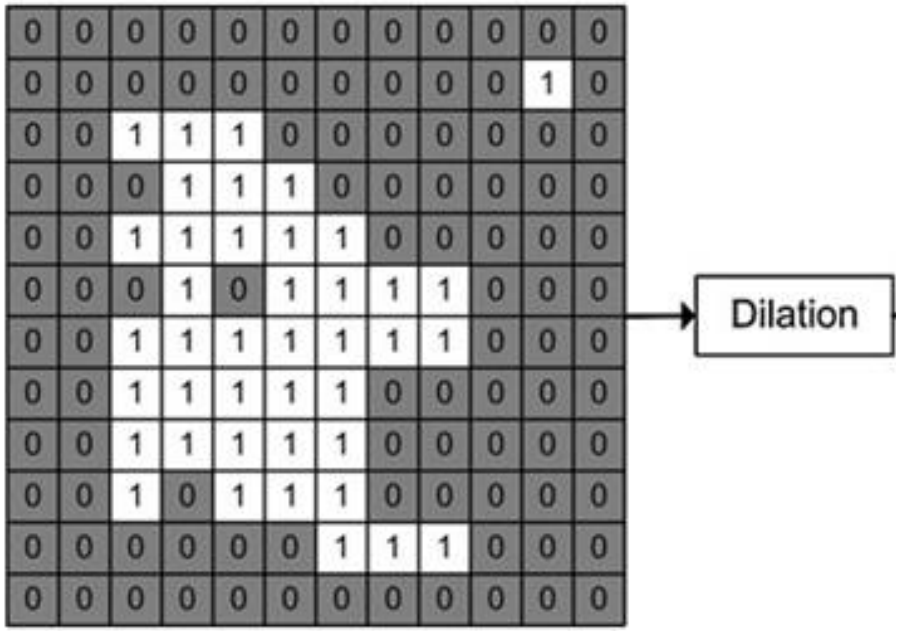


What do you observe here?

2-D Example of Dilation

- Using a 3x3 structuring element, with the middle centre pixel taken as the reference pixel:

1	1	1
1	1	1
1	1	1



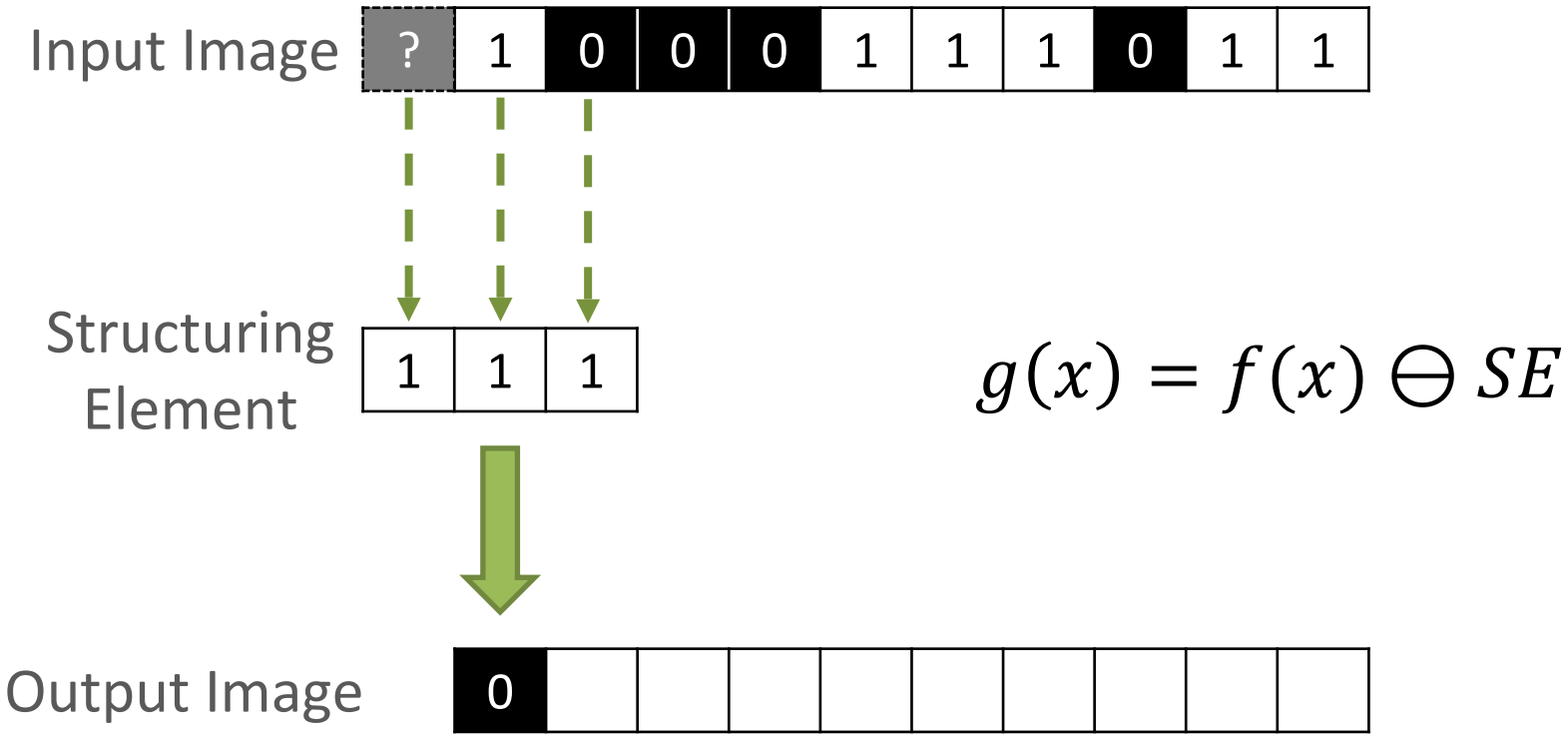
A. More 1s, and thicker white regions
B. More 0s and thinner white regions
C. No change

What would be the output?

Dilation vs. Erosion

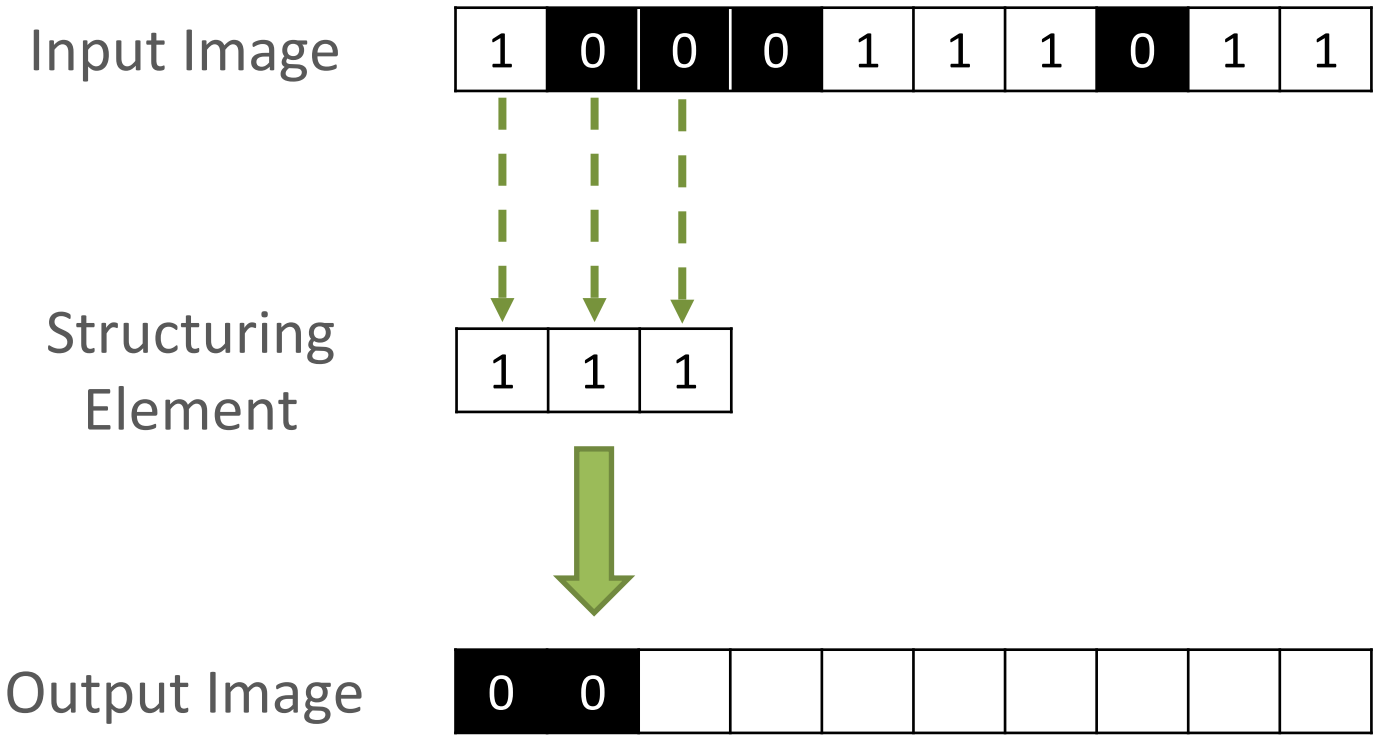
- At each position:
  - **Dilation:** If **current pixel** is foreground, OR the structuring element with the input image
  - **Erosion:** If **every pixel** under the structuring element's nonzero entries is foreground, **OR** the current pixel with structuring element (at reference point).

1-D Example of Erosion

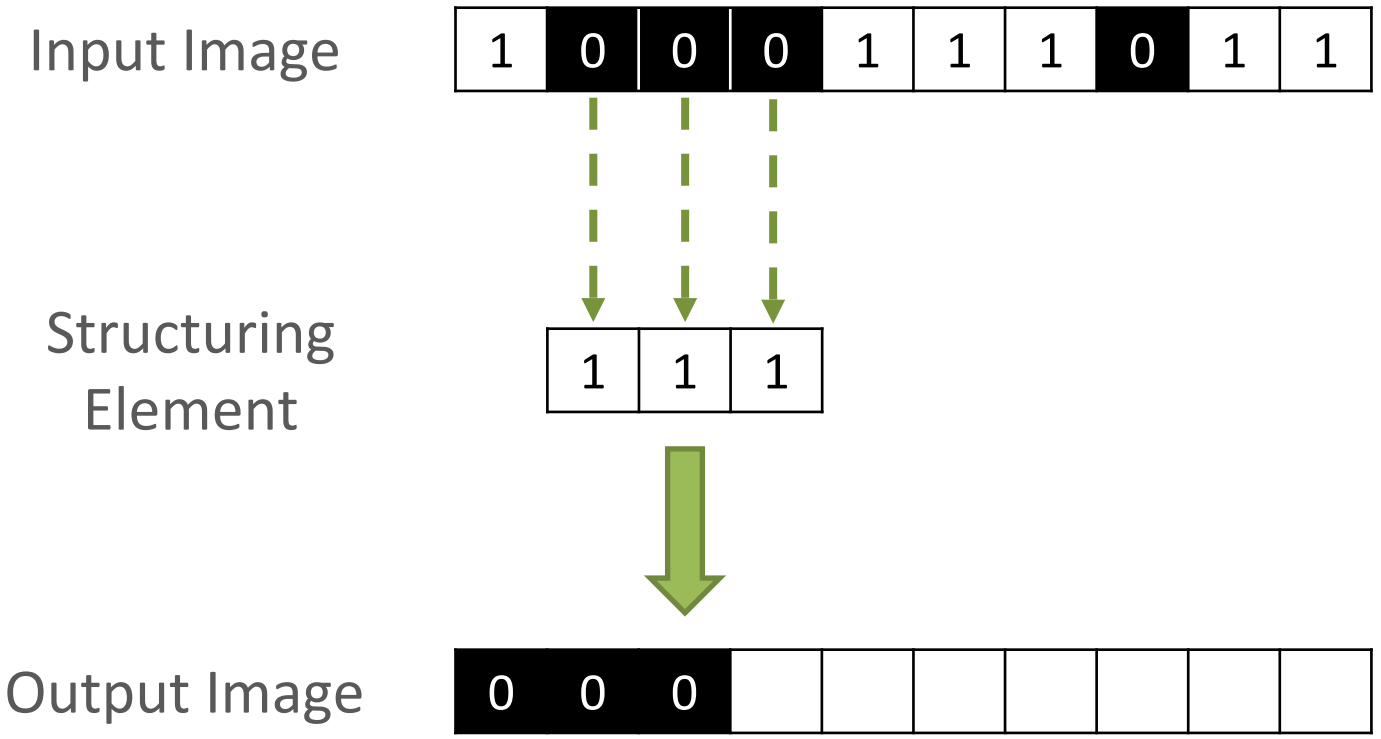


$$g(x) = f(x) \ominus SE$$

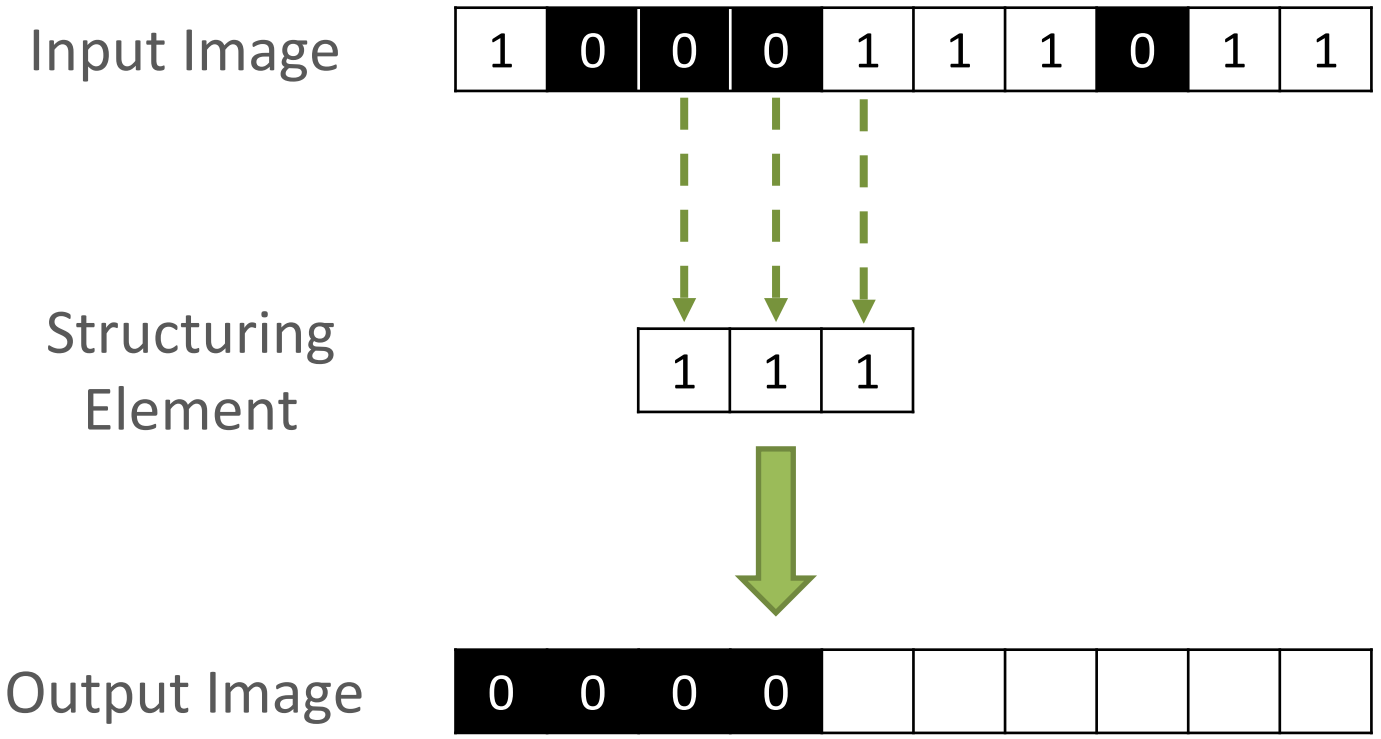
1-D Example of Erosion



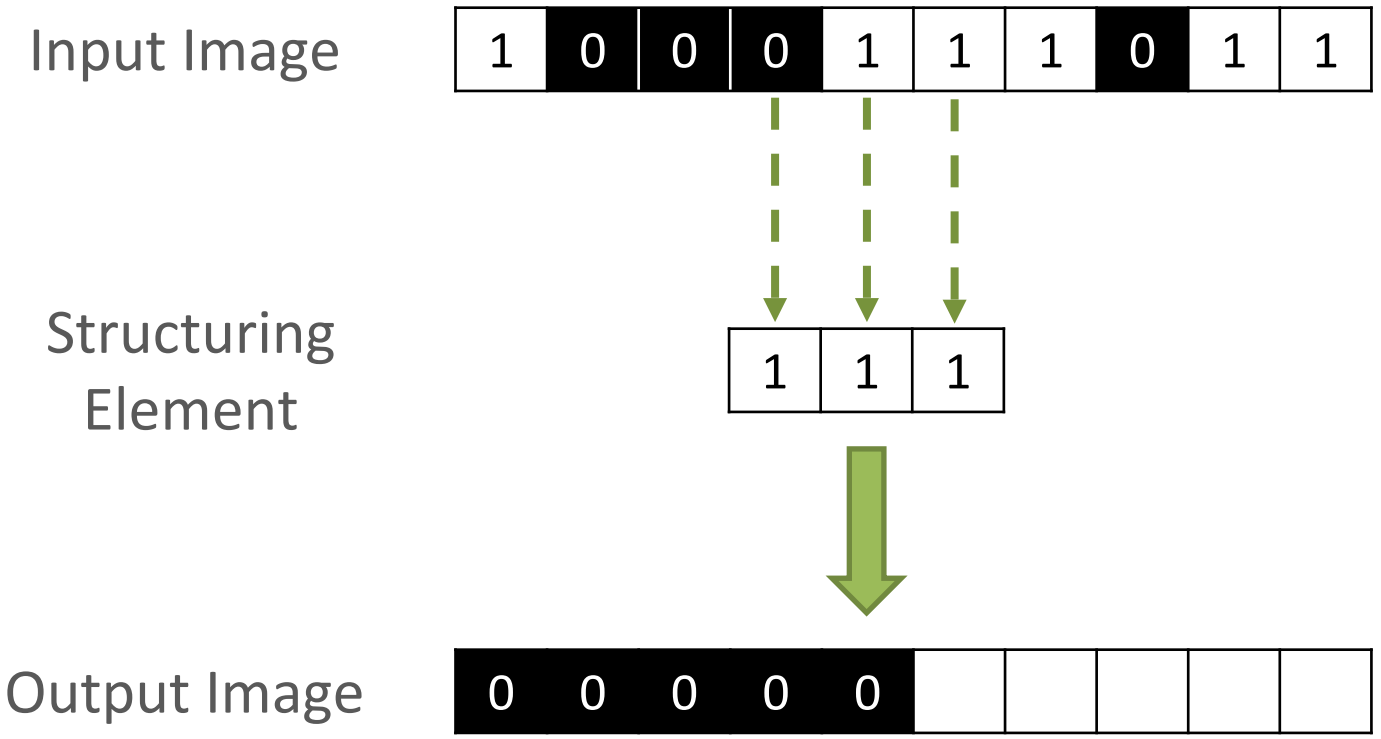
1-D Example of Erosion



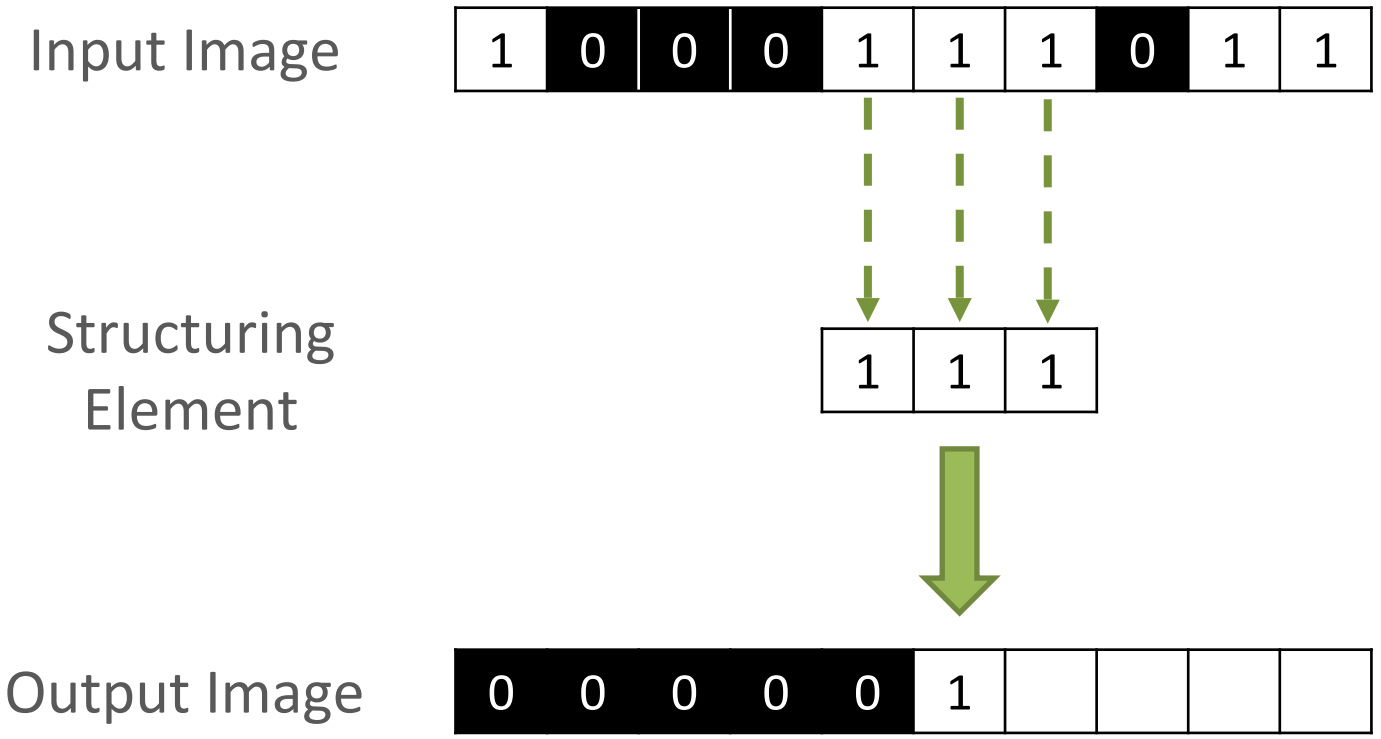
1-D Example of Erosion



1-D Example of Erosion

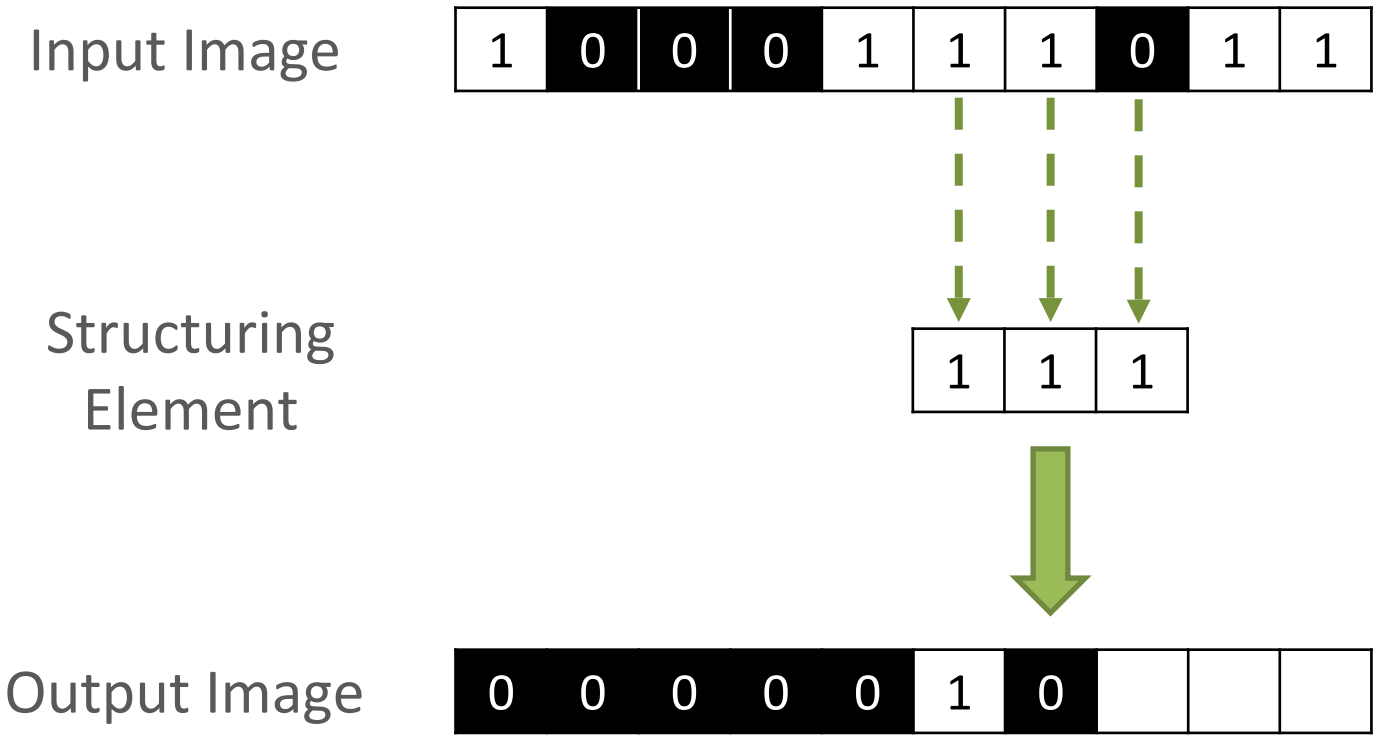


1-D Example of Erosion

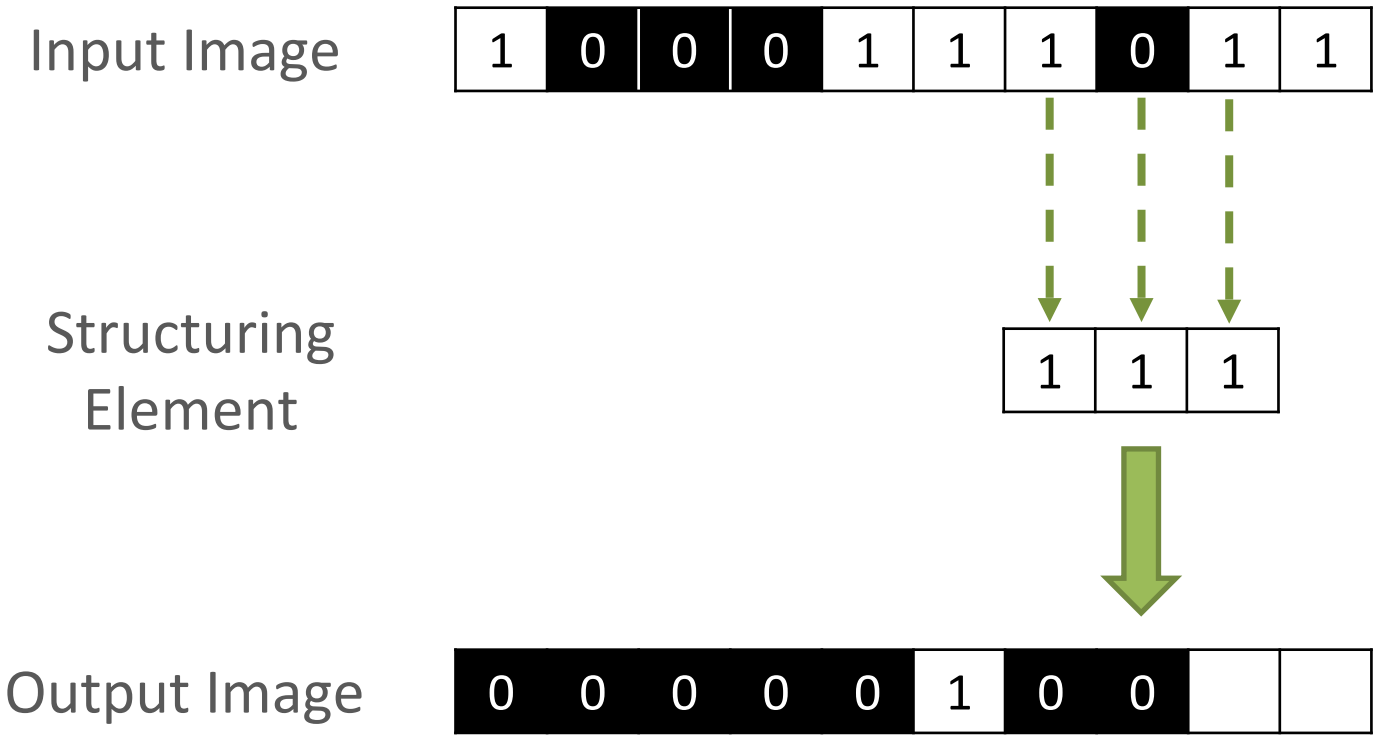


Remember: OR-ing only happens when ALL are FOREGROUND PIXELS!

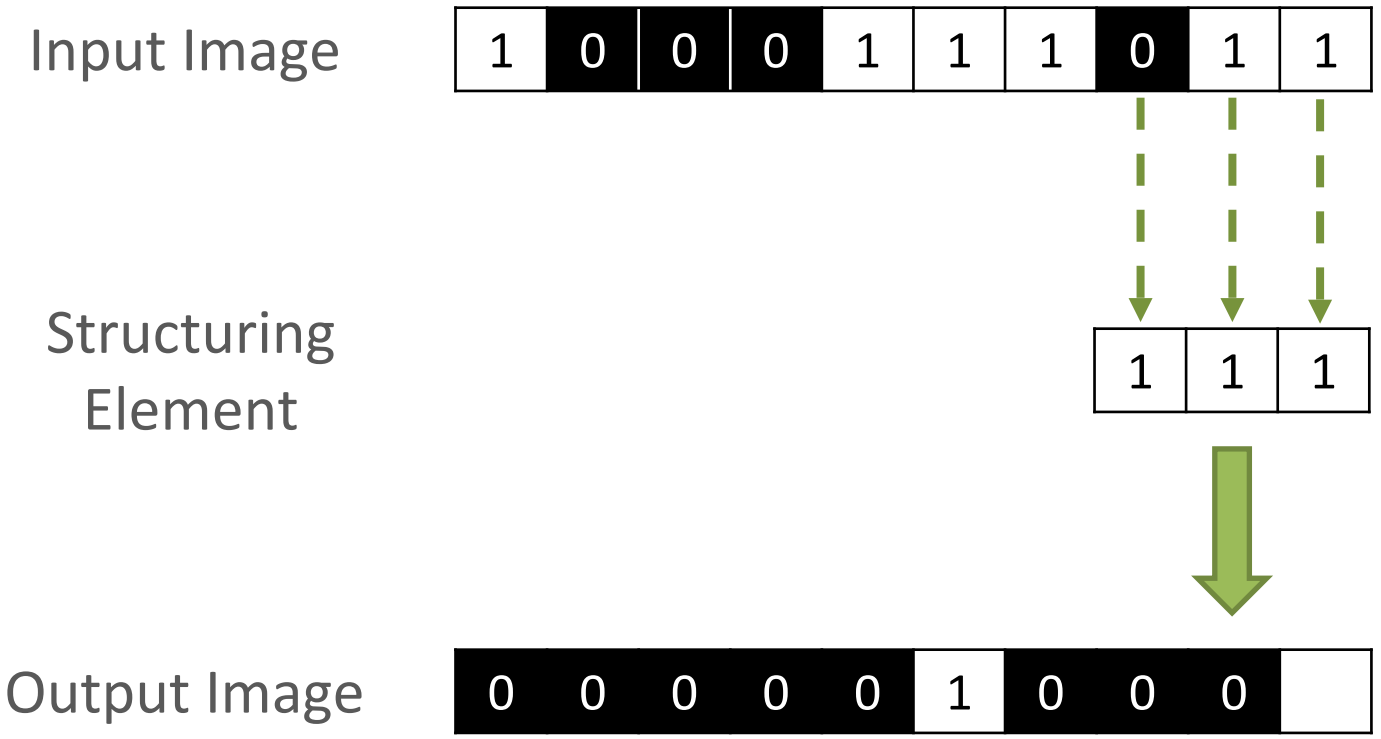
1-D Example of Erosion



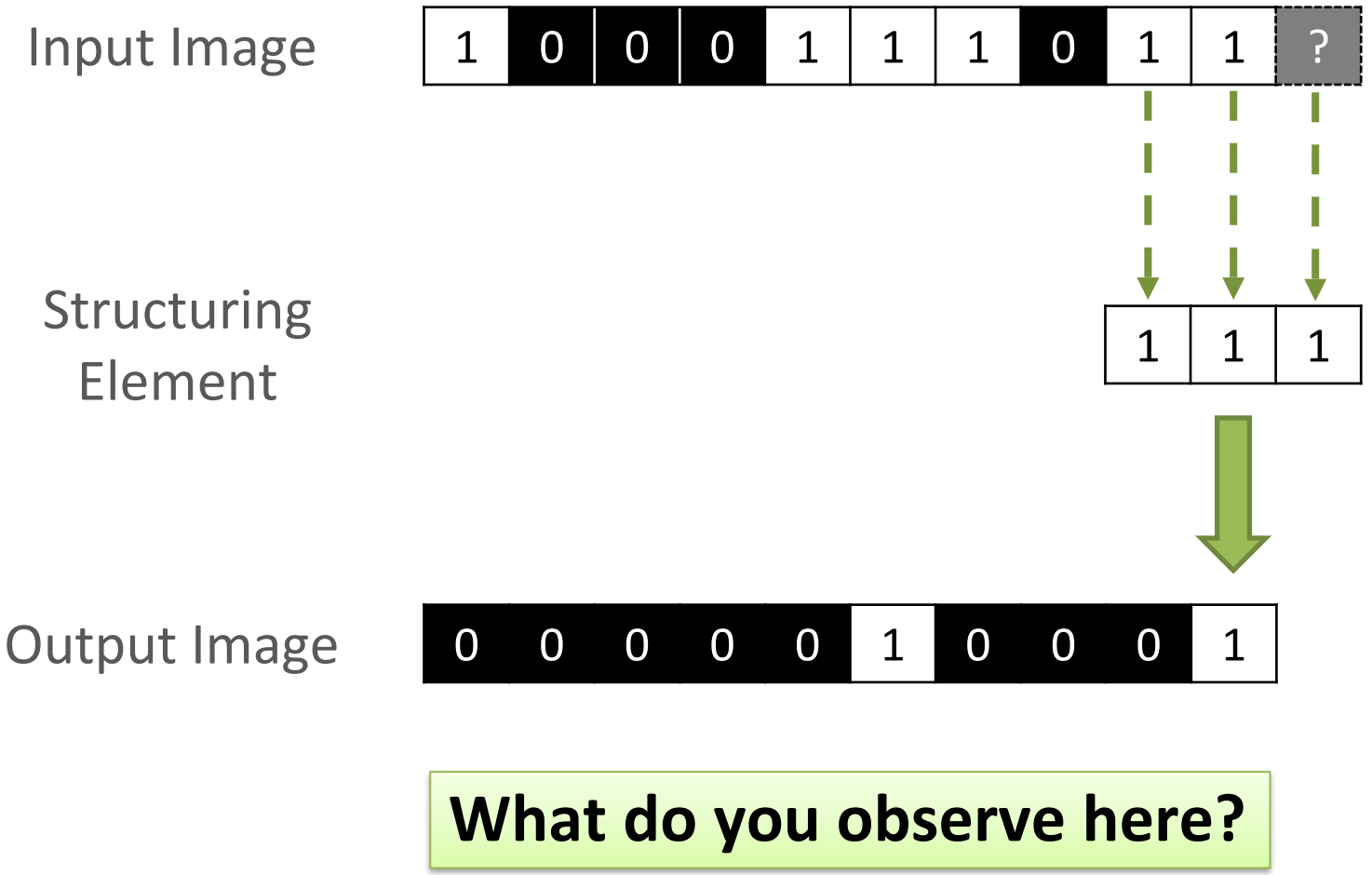
1-D Example of Erosion



1-D Example of Erosion



1-D Example of Erosion

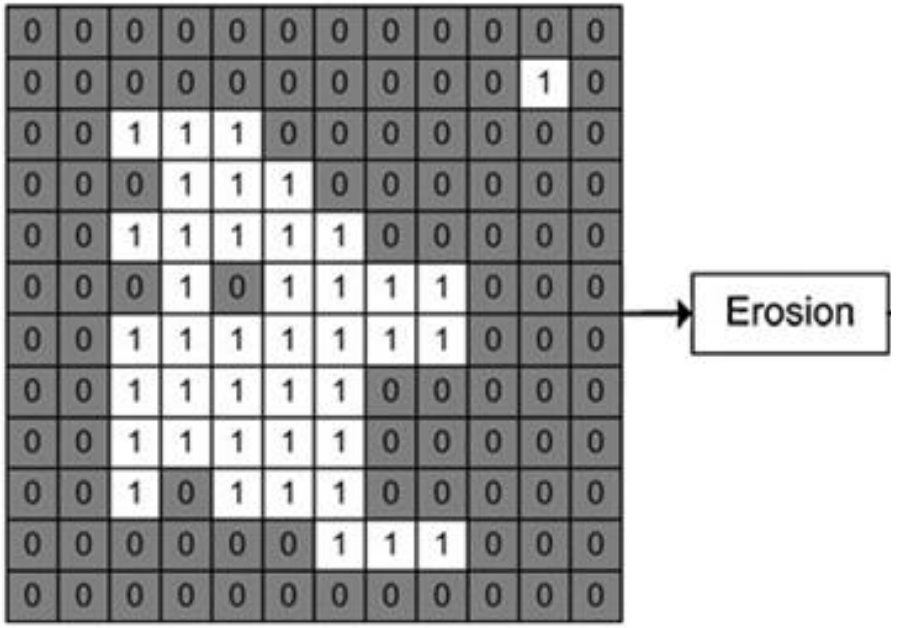


What do you observe here?

2-D Example of Erosion

- Using a 3x3 structuring element, with the middle centre pixel taken as the reference pixel:

1	1	1
1	1	1
1	1	1



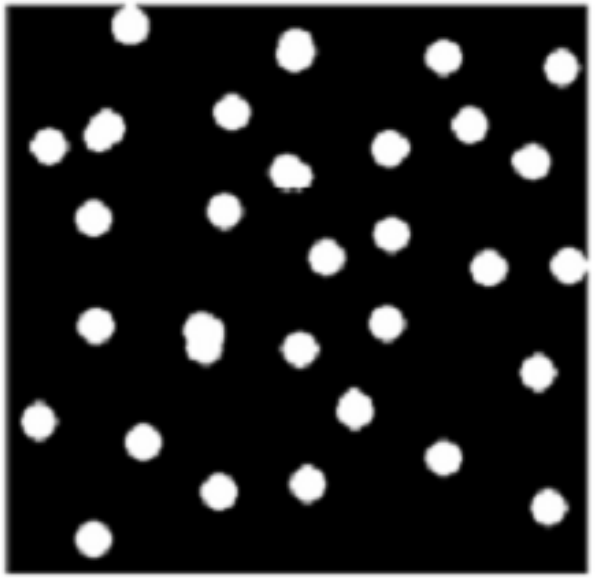
A. More 1s, and thicker white regions
B. More 0s and thinner white regions
C. All the white disappear

What is the output again?

Opening

- **Erode**, then **dilate**
- Remove small objects, keep original shape

Before opening



After opening

Closing

- **Dilate**, then **erode**
- Fill holes, but keep original shape

Before closing

After closing

Morphology operators on grayscale images

- Dilation and erosion typically performed on binary images
- If image is grayscale: for dilation take the neighbourhood **max**, for erosion take the **min**

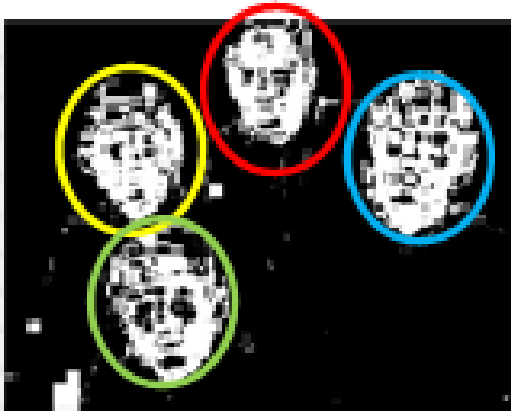
Marking multiple regions of interest?

Connected components / Region labelling

Two tasks lie ahead…

1. What to do with “noisy” binary outputs?

2. How to demarcate (mark) multiple regions of interest?
   - Count objects
   - Compute further features from objects



Connected Components

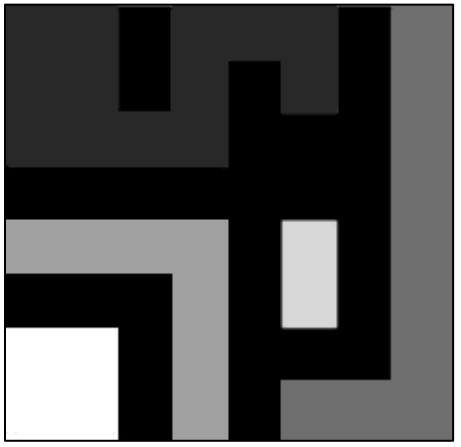
- Identify distinct regions of “connected pixels” by labelling

Binary Image

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

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

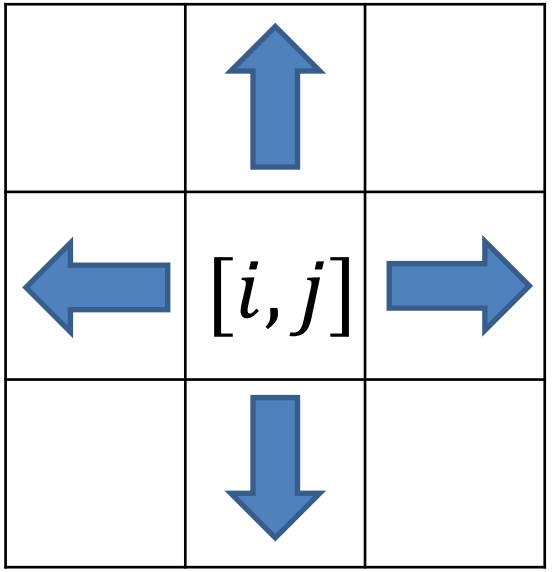
Connected Components Labeling



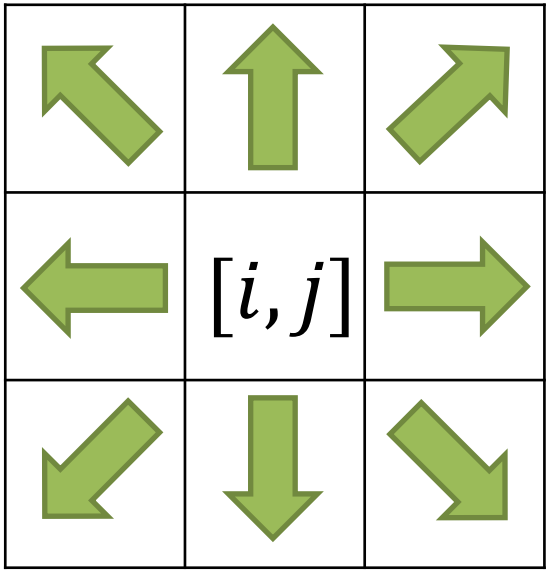
From Computer Vision (Shapiro & Stockman)

Connectedness

- Defining which pixels are considered neighbours



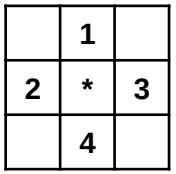
4-connected



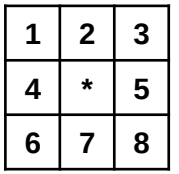
8-connected

Recursive method

1. Negate the binary image (so that 1-pixels become negative-1's)
   - to distinguish unprocessed pixels (-1) from those of component label 1
2. Find an unlabelled pixel (-1), assign it a new label
3. Search to find its neighbours (that have value -1)
   - following scan-line order



Four-neighborhood



Eight-neighborhood

4. Recursively repeat to find their neighbours till there are no more left
5. Go back to (2) and repeat.

Connected components

Connected components of 1’s from thresholded image

Connected components of cluster labels

Region properties

- Given connected components, can compute simple features per blob, such as:
  - **Area** (number of pixels in the region)
  - **Centroid** (average x and y position of pixels in region)
  - **Bounding box** (min and max coordinates)
  - **Circularity** (ratio of mean distance to centroid over std.)

Circularity

- Measure of “round-ness” of a region
- Circularity values – **large** $\Longrightarrow$ more circular in shape
- There are a few methods. Simplest formula:

$$C = \frac{4 \times Area \times \pi}{(Perimeter)^2}$$

The circularity measure expects a value between 0 and 1. Closer to 1, rounder the component.

Circularity

- Haralick (1974) proposed a quite robust measure for circularity:

circularity

$$C_2 = \frac{\mu_R}{\sigma_R} \tag{3.12}$$

where μ_R and σ_R are the mean and standard deviation of the distance from the centroid of the shape to the shape boundary and can be computed according to the following formulas.

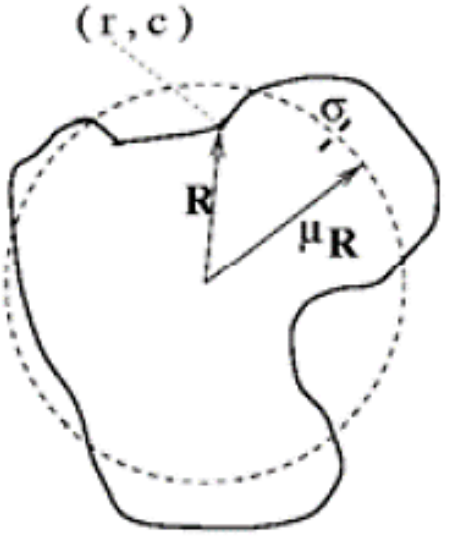
mean radial distance:

$$\mu_R = \frac{1}{K}\sum_{k=0}^{K-1} \|(r_k, c_k) - (\bar{r}, \bar{c})\| \tag{3.13}$$

standard deviation of radial distance:

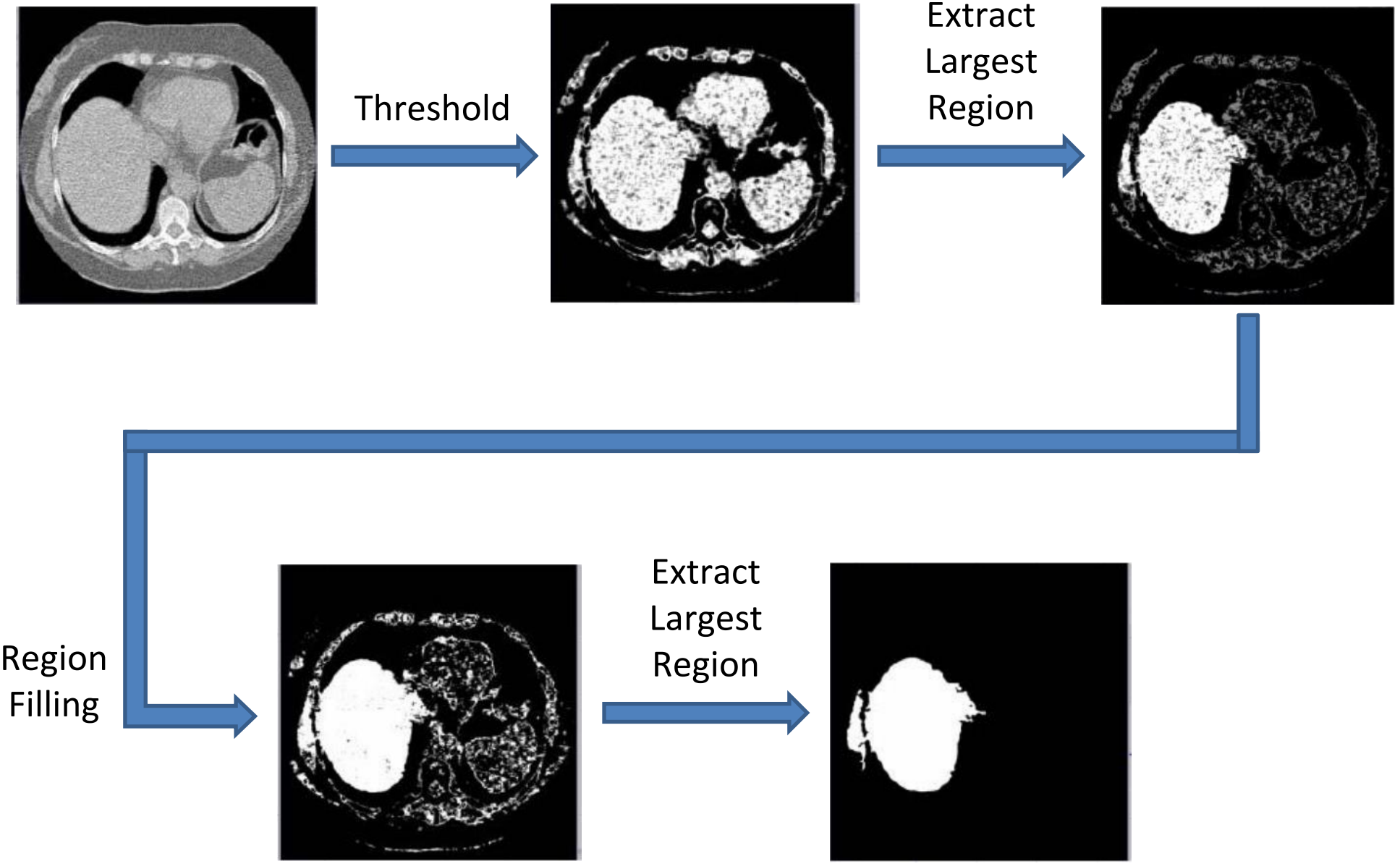
$$\sigma_R = \left(\frac{1}{K}\sum_{k=0}^{K-1} [\|(r_k, c_k) - (\bar{r}, \bar{c})\| - \mu_R]^2\right)^{1/2} \tag{3.14}$$

where the set of pixels (r_k, c_k), $k = 0, \ldots, K-1$ lie on the perimeter P of the region. The circularity measure C_2 increases monotonically as the digital shape becomes more circular and is similar for digital and continuous shapes.

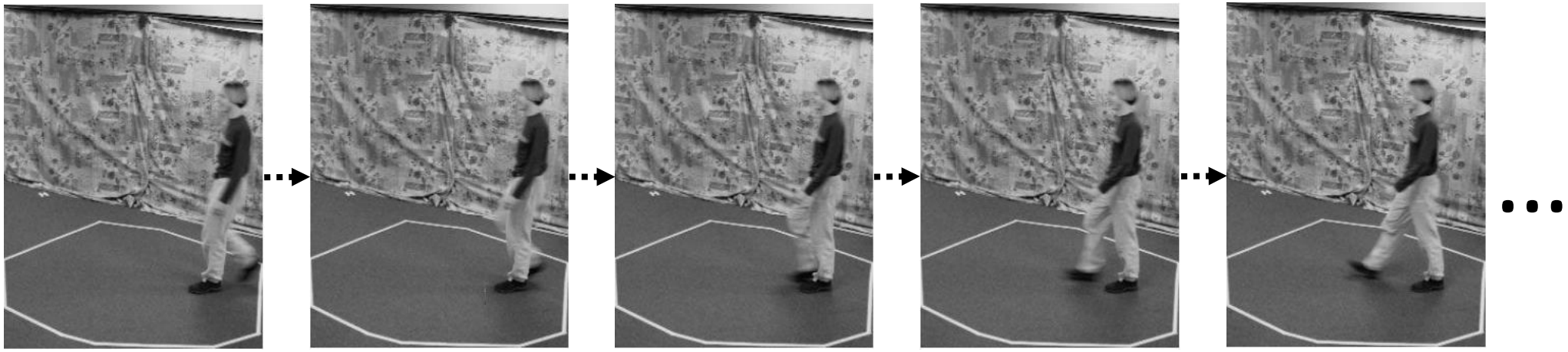


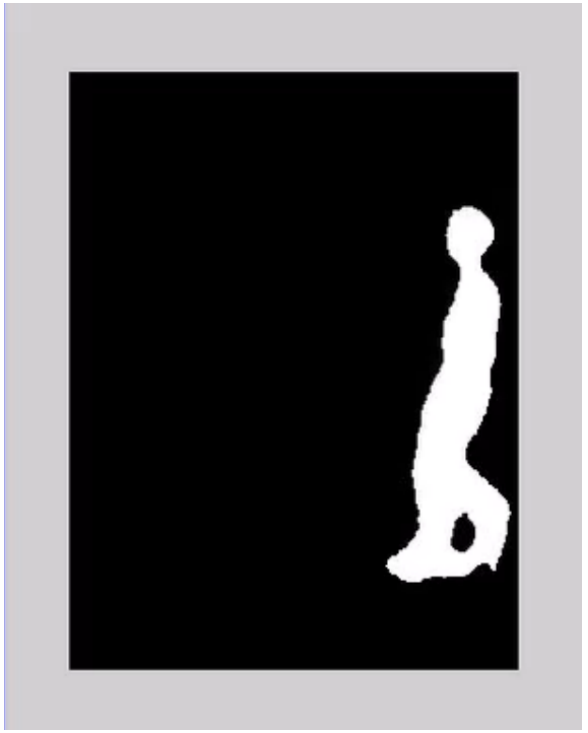
Application Examples

Example: Segmentation of a liver



Example: Background subtraction + blob detection





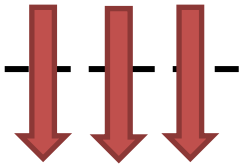
Binary images

- **Pros**
  - Can be fast to compute, easy to store
  - Simple processing techniques available
  - Lead to some useful compact shape descriptors
- **Cons**
  - Hard to get “clean” silhouettes
  - Noise is common in realistic scenarios
  - Can be too coarse of a representation
  - Not 3D in nature

Techniques that get us what we want

- Operations, techniques

Derivative filters
Smoothing, morphology
Thresholding
Connected components
Matched filters
Histograms



- Features, representations

Edges, gradients
Blobs/regions
Local patterns
Textures
Color distributions

Summary

- **Binary Images**
  - Thresholding
- **Morphological operations**
  - Dilation, erosion, opening, closing
- **Connected components & Region labelling**
- **Region properties**
  - Area, centroid, bounding box, circularity

Recommended Reading

- [Gonzalez & Woods] Chapter 9 & 10 (10.3 in particular)
- [Forsyth & Ponce] Chapter 9 (9.5.4 in particular)