

Computer Vision

Paulo Dias





- Morphological operations
 - Dilation, erosion
 - Opening, closing
- Segmentation
 - Thresholding
 - Region growing

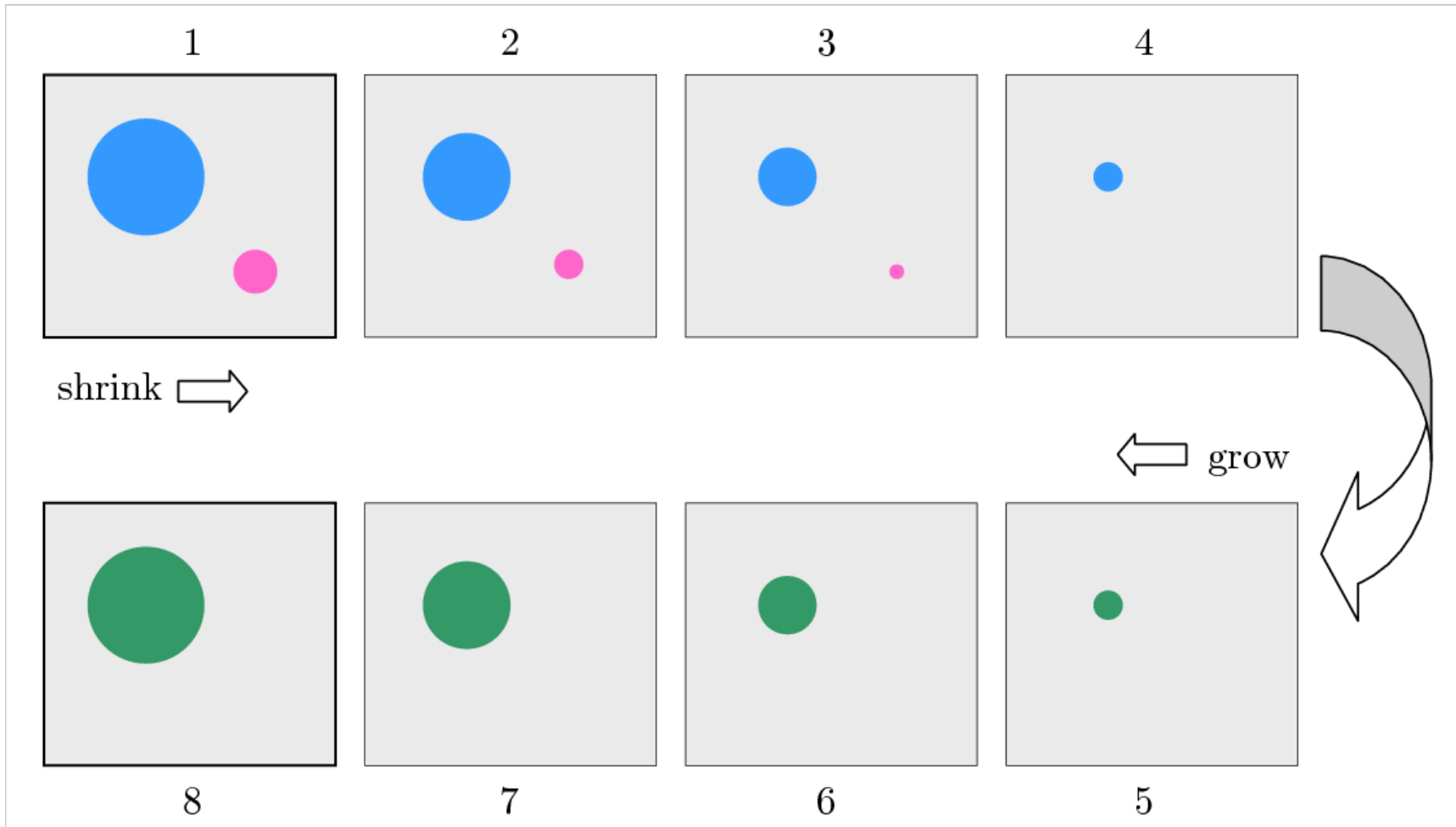


- Morphological operations
 - Dilation, erosion
 - Opening, closing
- Segmentation
 - Thresholding
 - Region growing



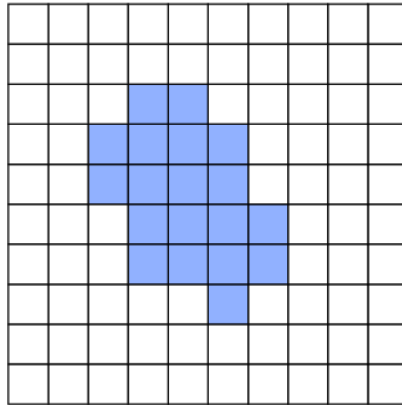
- Morphological mathematics operates on images as a **set of points**
- **Modify in a control way** the structure/morphology of an image
- Typically used in **binary images**
- Can be used in graylevel or colour image as well
- Used in Image Processing for
 - Filtering
 - Segmentation
 - Object description

- Main idea

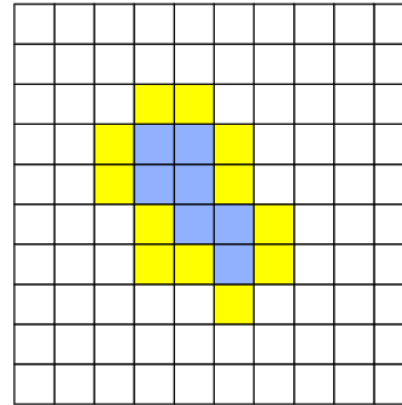


- Main idea

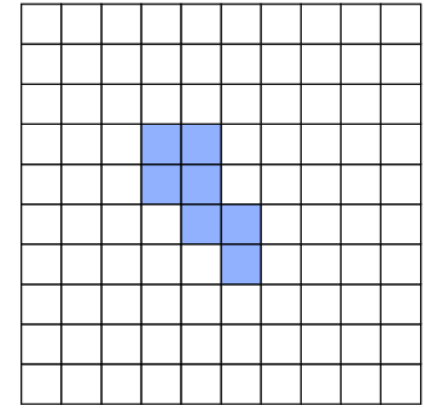
Erosion



(a)

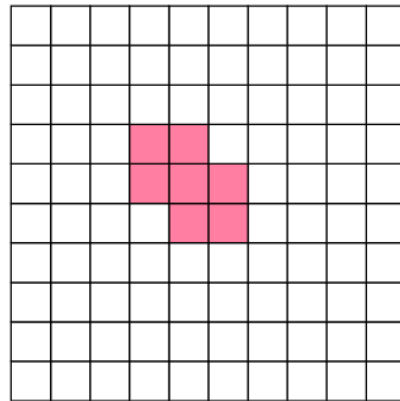


(b)

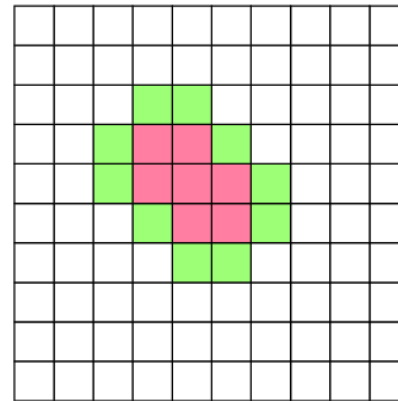


(c)

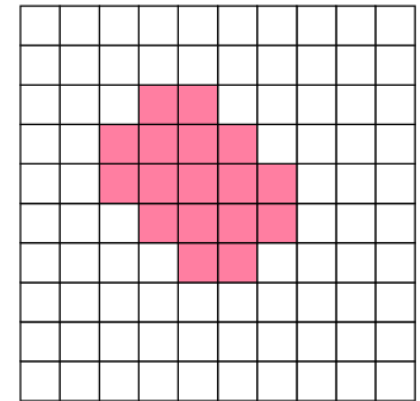
Dilation



(a)



(b)




(c)



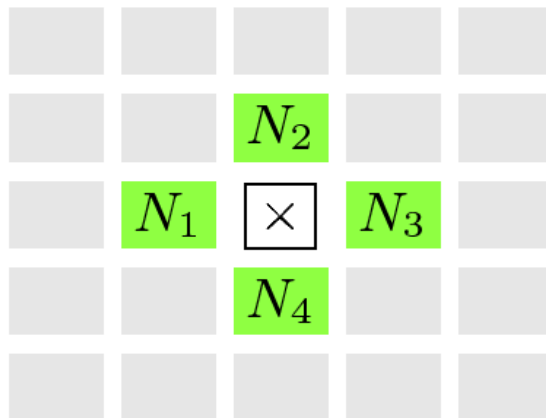
- Structuring element

$$H = \begin{array}{|c|c|c|} \hline \square & \bullet & \square \\ \hline \bullet & \blacksquare & \bullet \\ \hline \square & \bullet & \square \\ \hline \end{array}$$

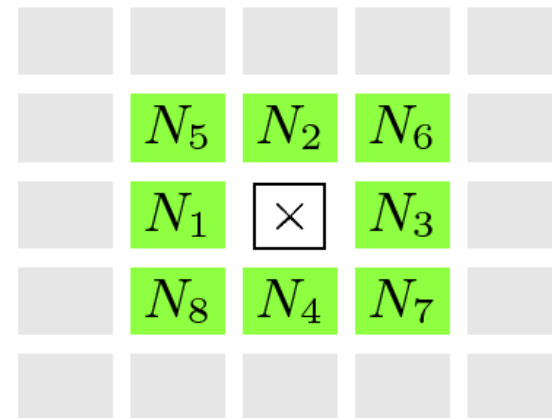
 origin (hot spot)

- Neighbourhood

\mathcal{N}_4



\mathcal{N}_8



- Main Morphological operations
 - Dilation
 - Erosion

} Basic Operations

 - Opening
 - Closing

} Composed Operations

Dilation



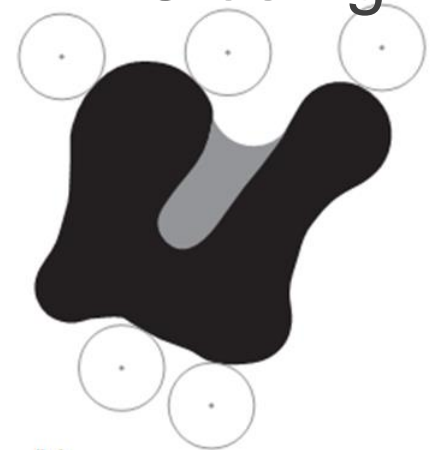
Erosion




Opening



Closing


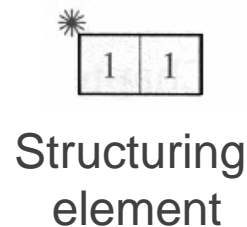


- Dilation
 - gradually enlarge the boundaries of regions
 - small holes and gaps are filled




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

Original image



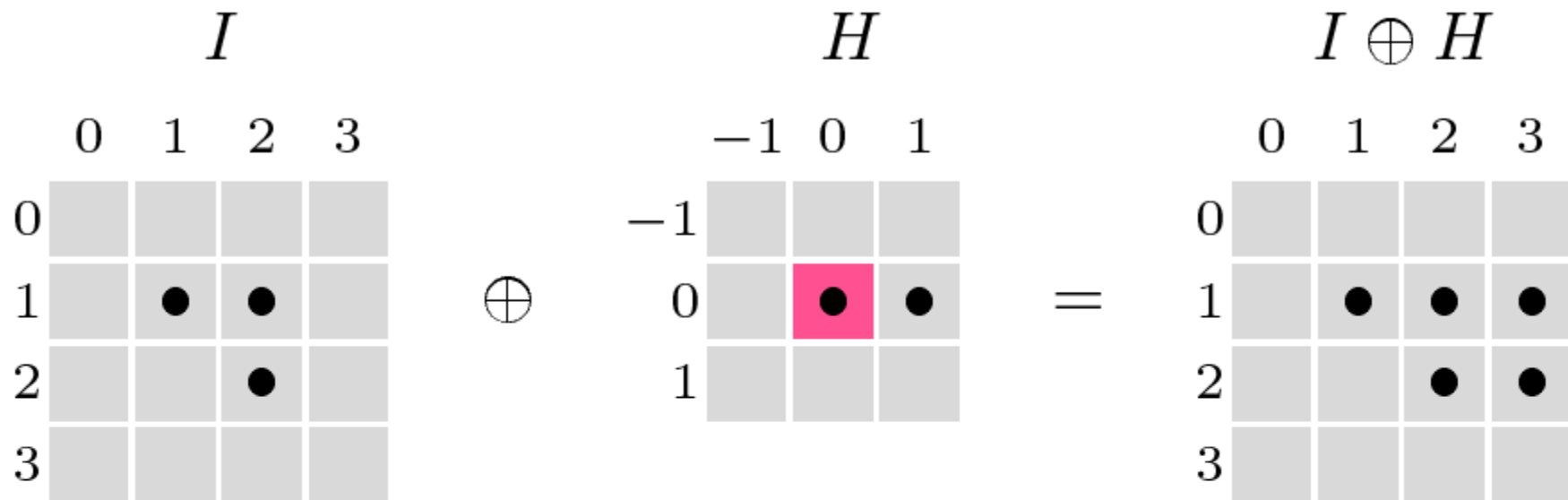
0	1	1	0	0
0	1	1	0	0
0	1	1	0	0
0	1	1	1	1
0	0	0	0	0

Dilation



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

$I \oplus X$




$$I \equiv \{(1, 1), (2, 1), (2, 2)\}, \quad H \equiv \{(\mathbf{0}, \mathbf{0}), (\mathbf{1}, \mathbf{0})\}$$

$$I \oplus H \equiv \{ (1, 1) + (\mathbf{0}, \mathbf{0}), (1, 1) + (\mathbf{1}, \mathbf{0}), \\ (2, 1) + (\mathbf{0}, \mathbf{0}), (2, 1) + (\mathbf{1}, \mathbf{0}), \\ (2, 2) + (\mathbf{0}, \mathbf{0}), (2, 2) + (\mathbf{1}, \mathbf{0}) \}$$


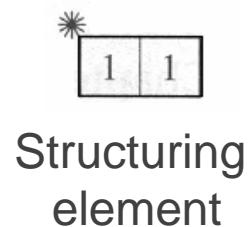


- Erosion
 - Dual of the dilation operation
 - Erode away the boundaries of regions of foreground
 - holes and gaps are increased




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

Original image



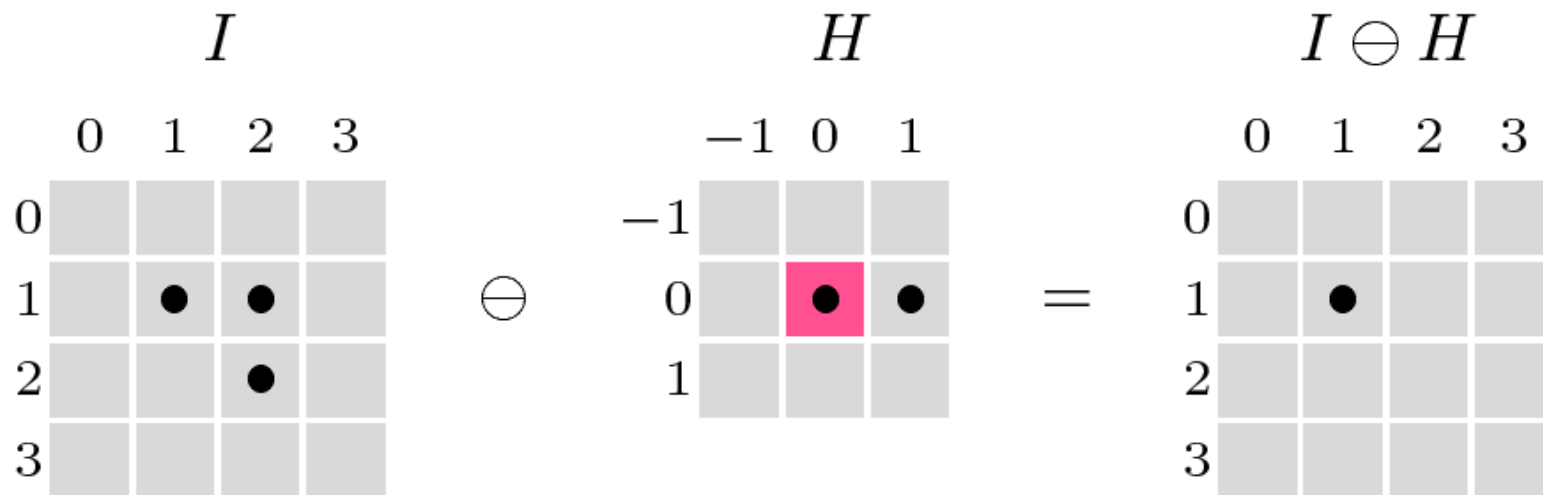
0	1	0	0	0
0	1	0	0	0
0	1	0	0	0
0	1	1	1	0
0	0	0	0	0

Erosion



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

$I \ominus X$

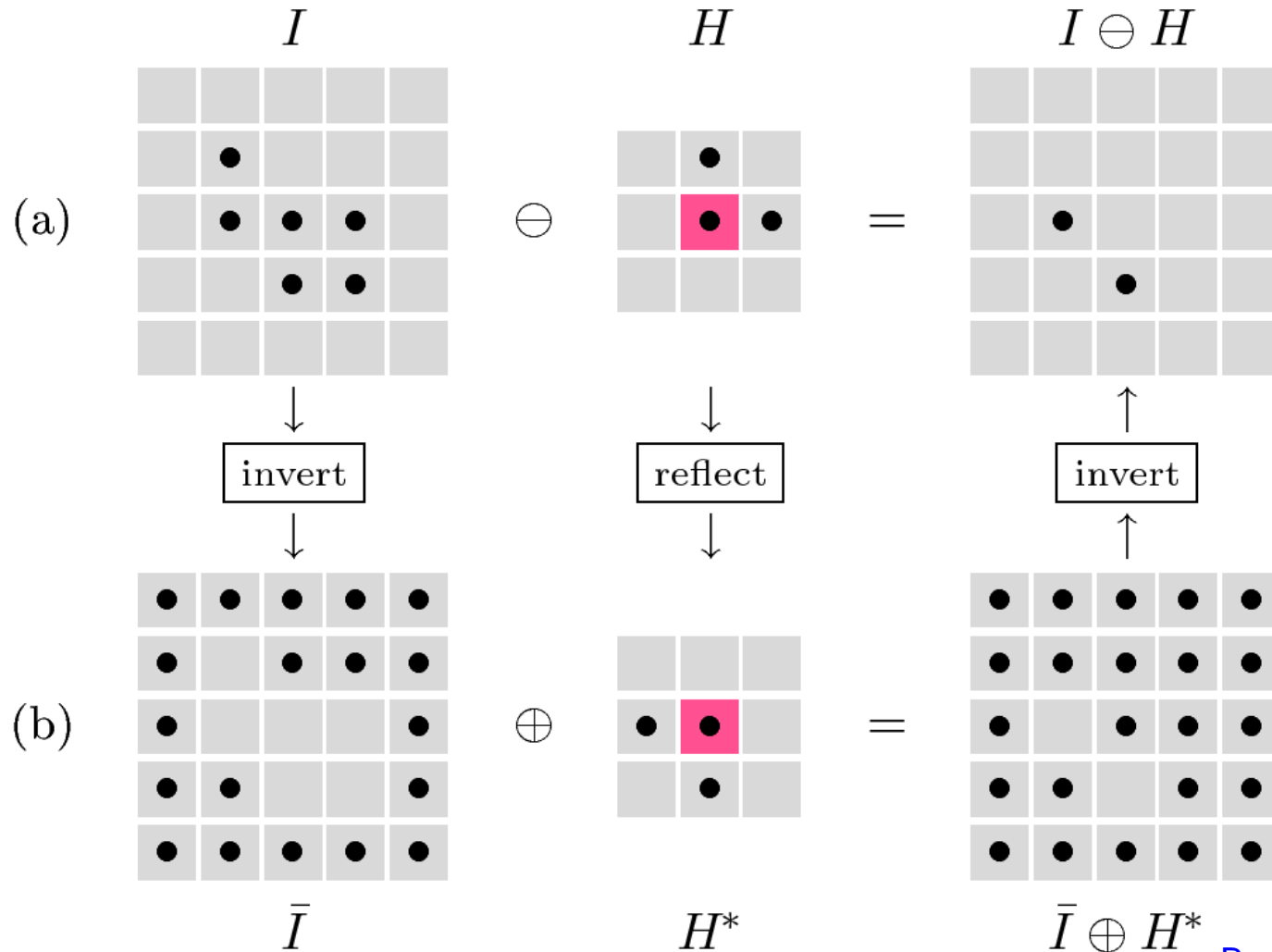


$$I \equiv \{(1, 1), (2, 1), (2, 2)\}, \quad H \equiv \{(\mathbf{0}, \mathbf{0}), (\mathbf{1}, \mathbf{0})\}$$

$$I \ominus H \equiv \{(1, 1)\} \text{ because}$$

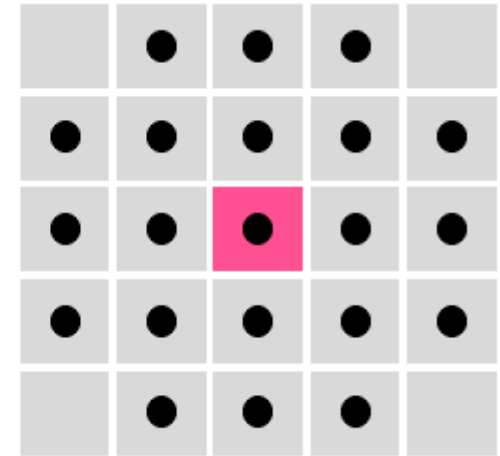
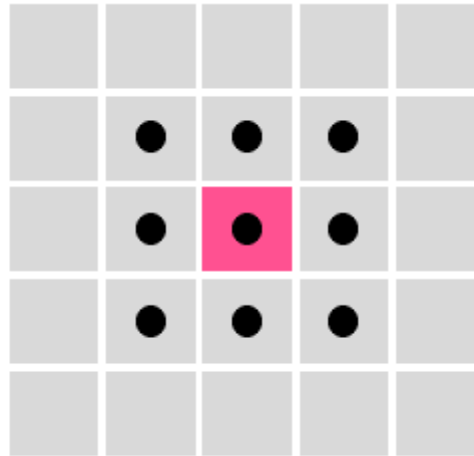
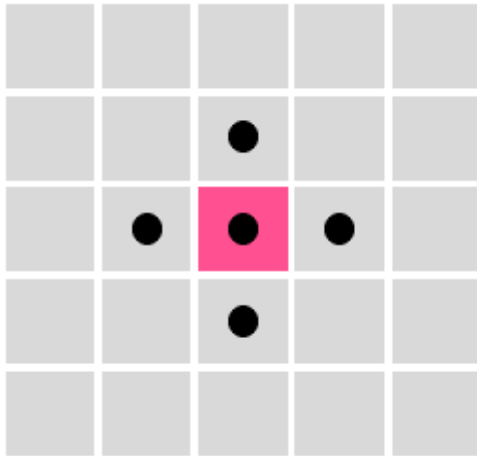
$$(1, 1) + (\mathbf{0}, \mathbf{0}) = (1, 1) \in I \quad \text{and} \quad (1, 1) + (\mathbf{1}, \mathbf{0}) = (2, 1) \in I$$

- Dilation and Erosion are dual operations





- Typical structuring elements






- Morphological operations
 - Dilation, erosion
 - Opening, closing
- Segmentation
 - Thresholding
 - Region growing



- Opening
 - Erosion followed by dilation
 - Idempotent operation
 - Results will not change applied multiple time
 - Union of all objects that fit in Structuring Element
 - Circular kernel:
 - Smooth edges of object
 - Broke thin connections


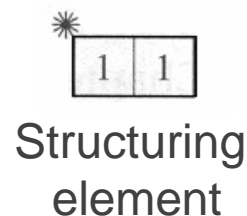


$$I \circ X = (I \ominus X) \oplus X$$




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

Original image



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

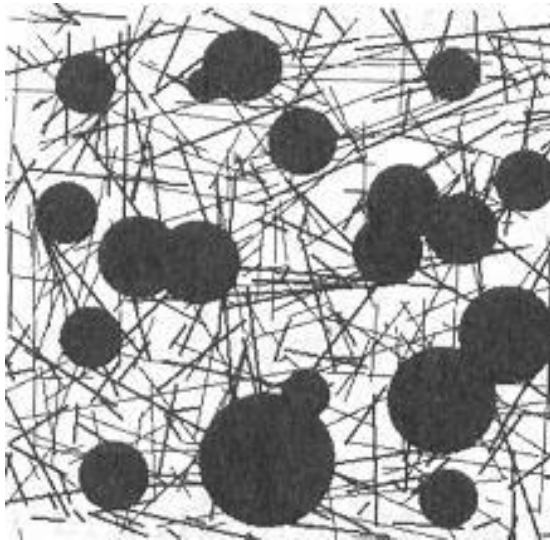
Opening



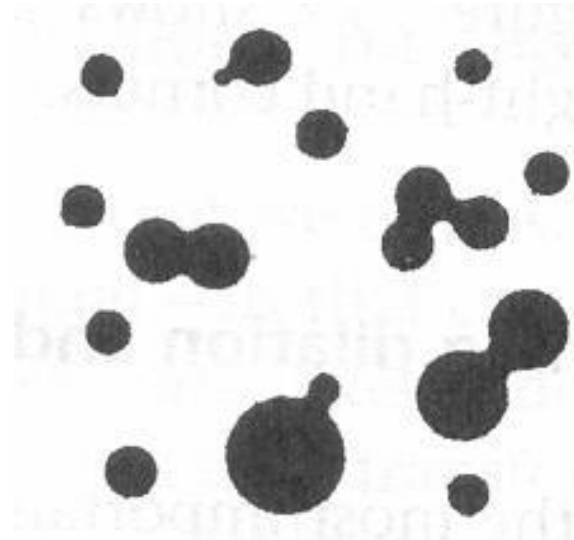
0	1	1	0	0
0	1	1	0	0
0	0	0	0	0
0	1	1	1	0
0	0	1	1	0

$I \circ X$

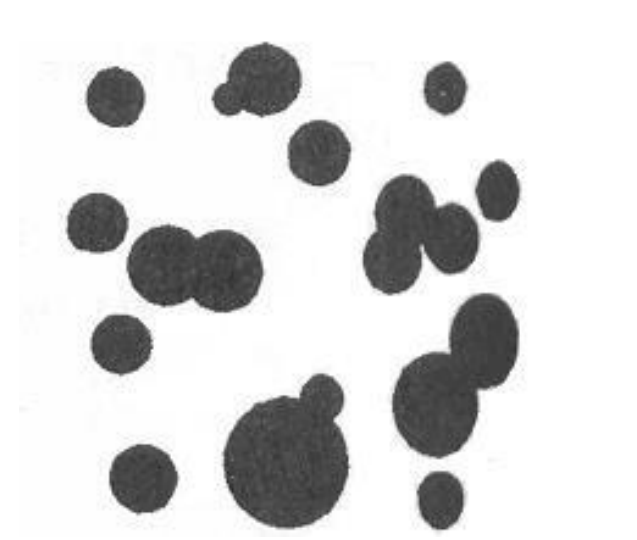
- Opening example
 - Circular structuring element
 - Radius of structuring element must be larger than subsets to remove



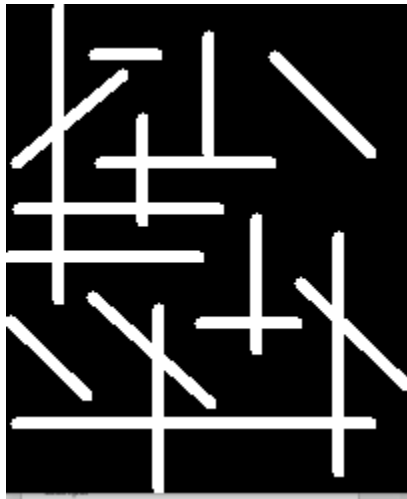
Original image



After erosion

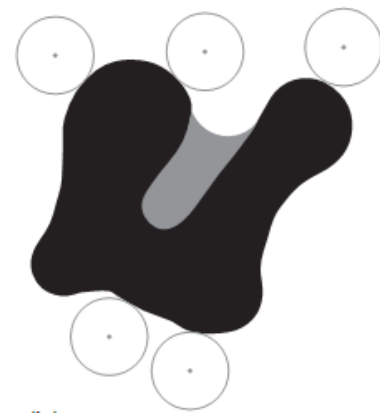


After dilation






- Closing
 - Dilation followed by erosion
 - Dual to opening
 - Idempotent operation
 - Results will not change applied multiple time


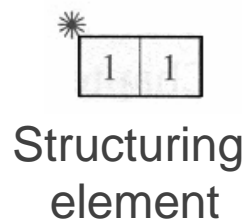


$$I \bullet X = (I \oplus X) \ominus X$$



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

Original image



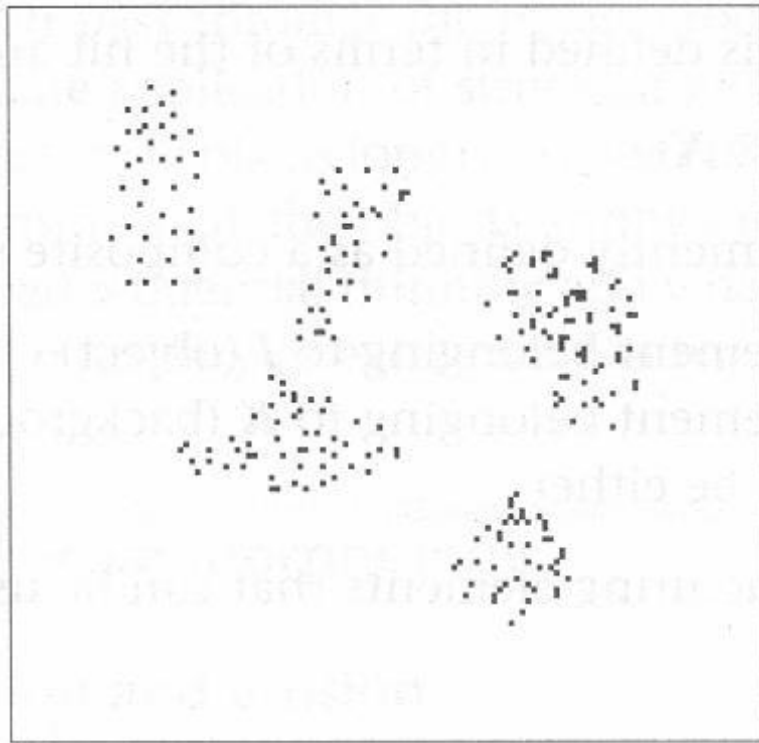
0	1	1	1	0
0	1	1	1	0
0	1	1	0	0
0	1	1	1	1
0	0	1	1	1

Closing

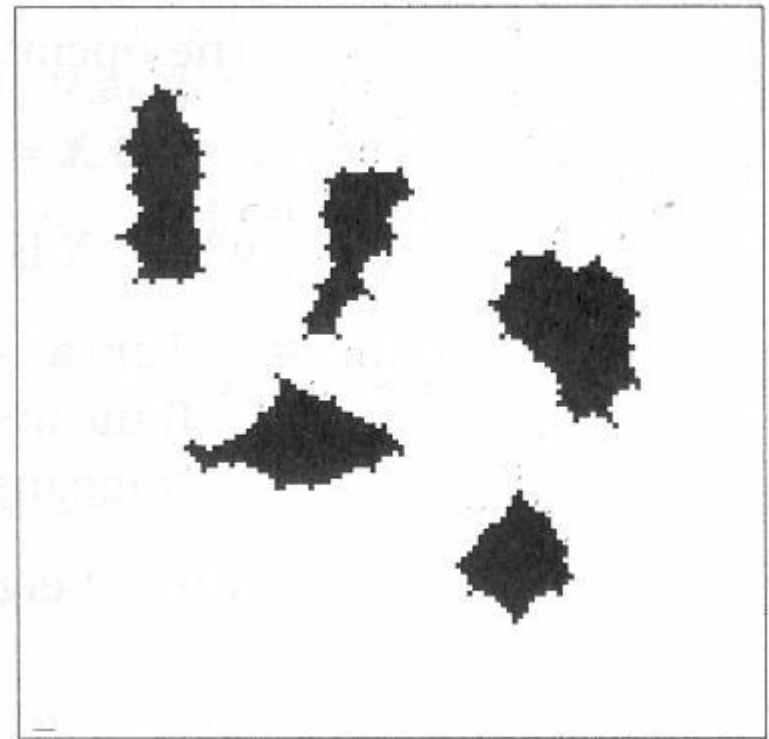
?

$I \bullet X$

- Closing example



Original image



After closing

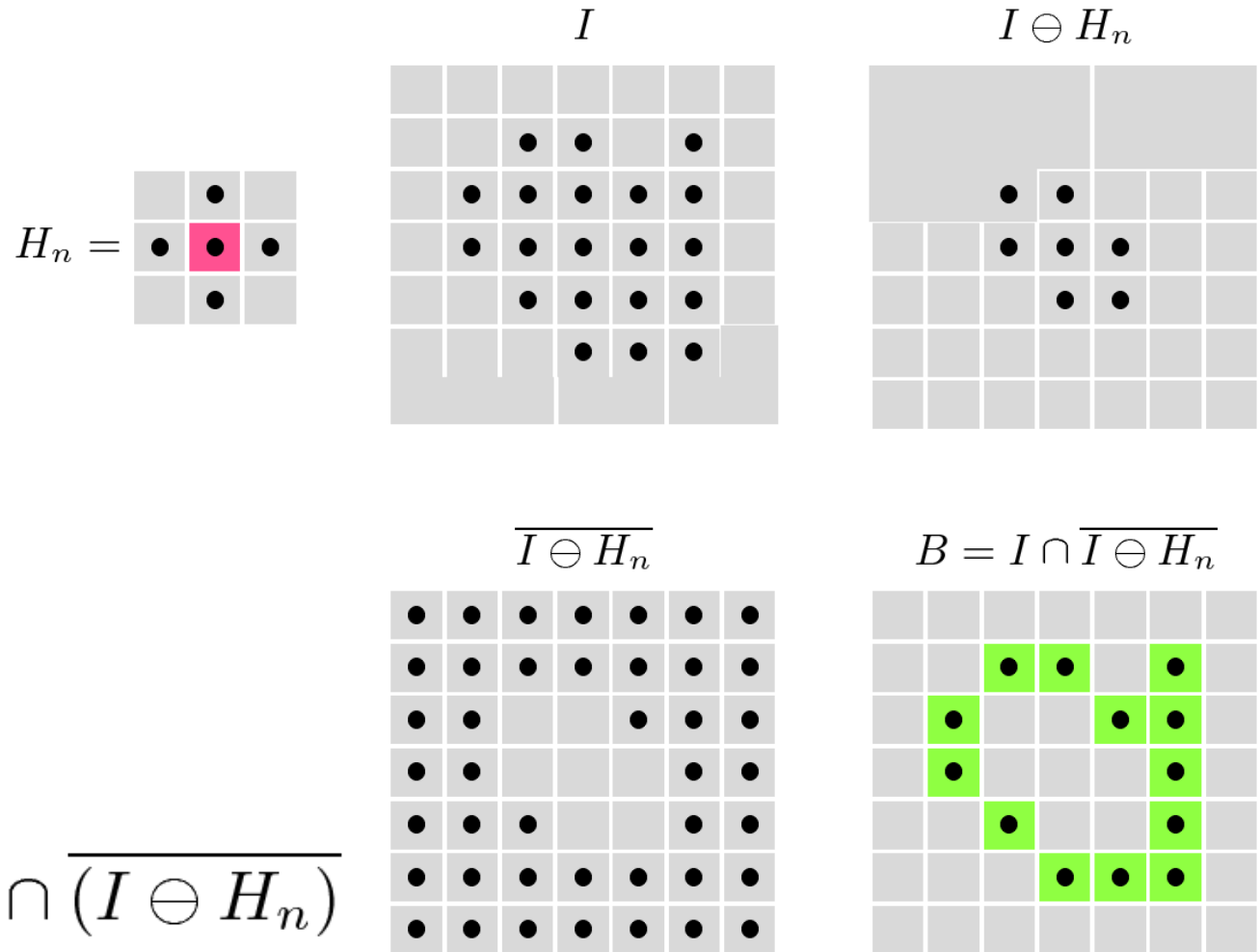
- Edge detection with morphology (outlining)
 - Since erosion results in an isotropic contraction of images, can be used for edge detection:

$$Edge = I - (I \ominus X)$$

- Erosion of objects and then subtraction from original (using 3x3 or 5x5 structuring element)
- Size of structuring element will have impact of contour thickness



- Edge detection with morphology (outlining)



$$I' = I \ominus H_n$$

$$B = I \cap \overline{I'} = I \cap \overline{I \ominus H_n}$$

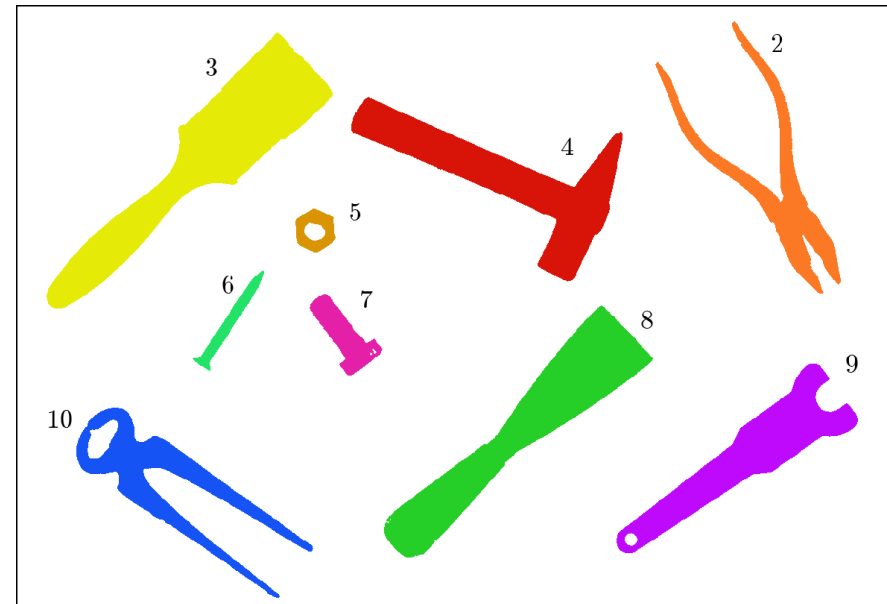
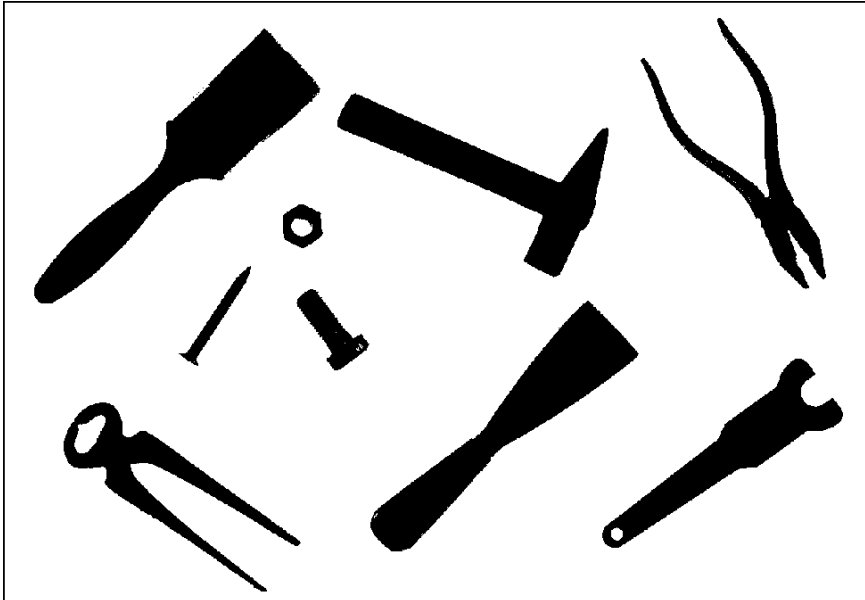


- Morphological operations
 - Dilation, erosion
 - Opening, closing
- **Segmentation**
 - Thresholding
 - Region growing



- Segmentation means **dividing image in regions**
- Often applied before image analysis
- Typical approach is to group pixels with similar properties

- Examples



Burger and Burge

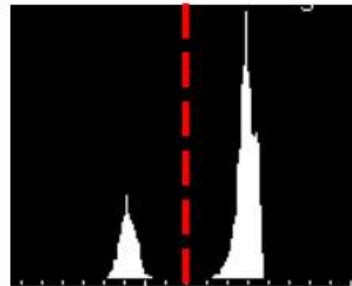
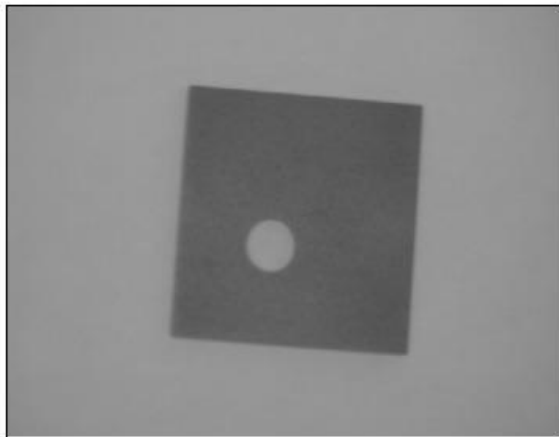


- No segmentation methods that can be used in every case
- No “perfect” segmentation method
- Typical segmentation are based in:
 - pixel intensity
 - regions
 - edges

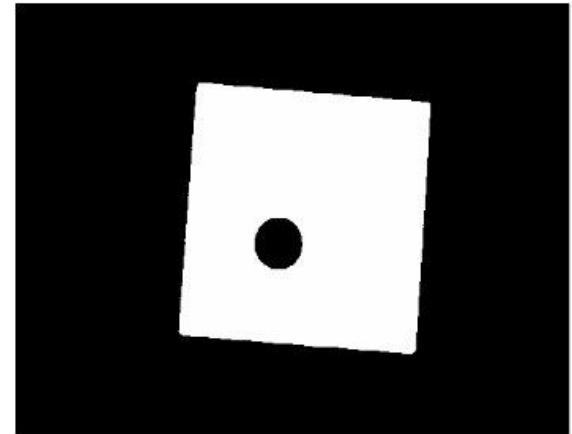


- Morphological operations
 - Dilation, erosion
 - Opening, closing
- Segmentation
 - **Thresholding**
 - Region growing

- Thresholding
 - Oldest segmentation method
 - Appropriate when object of interest have homogeneous intensity different from background
 - Not easy to find the adequate value



Adequate
threshold

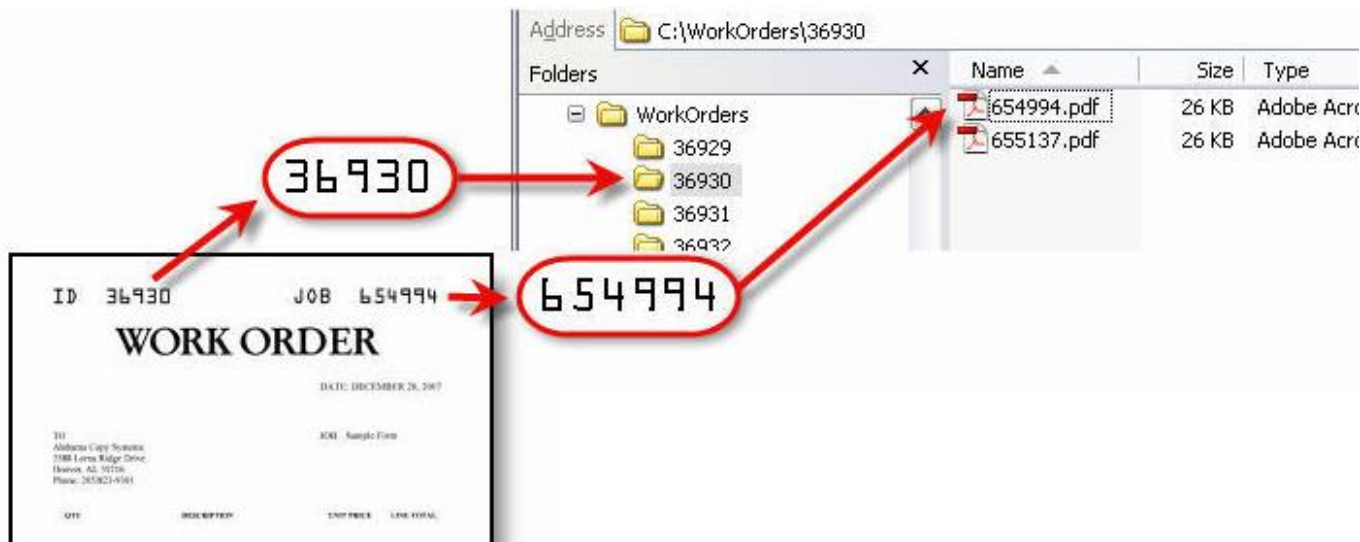
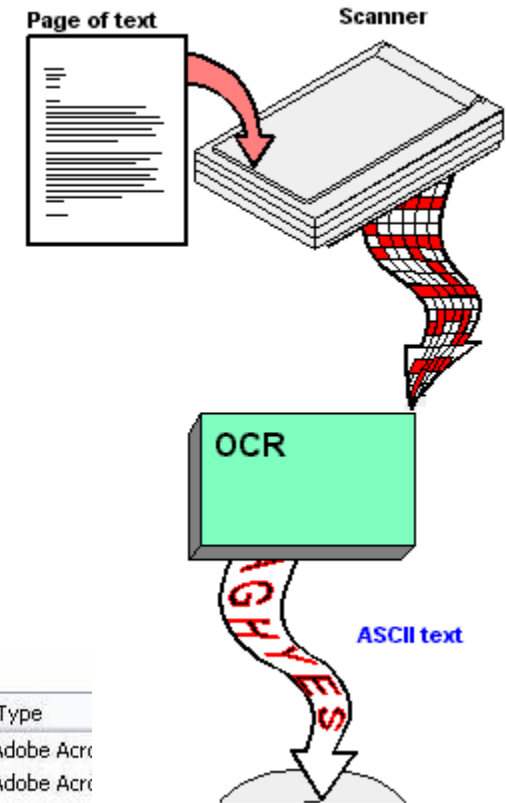


Segmentation - Thresholding

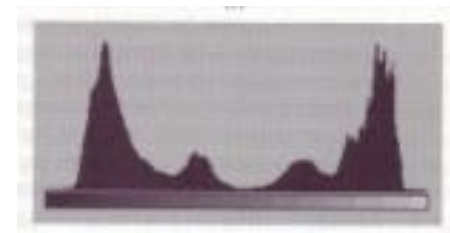
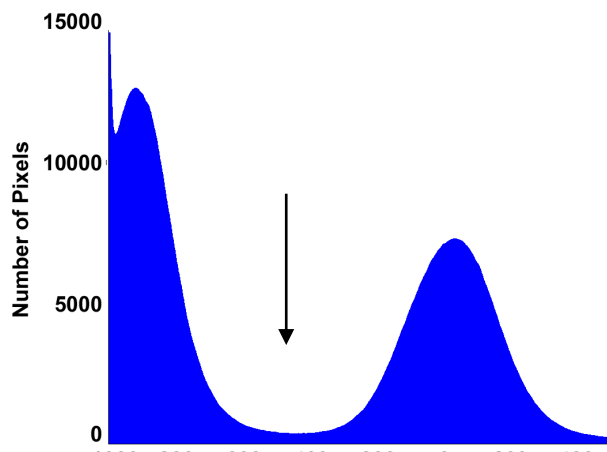


- Oldest segmentation approach
- Appropriate whenever the intensity of the objects of interest is homogeneous and they are different from the background

Example: *OCR (Optical Character Recognition)*



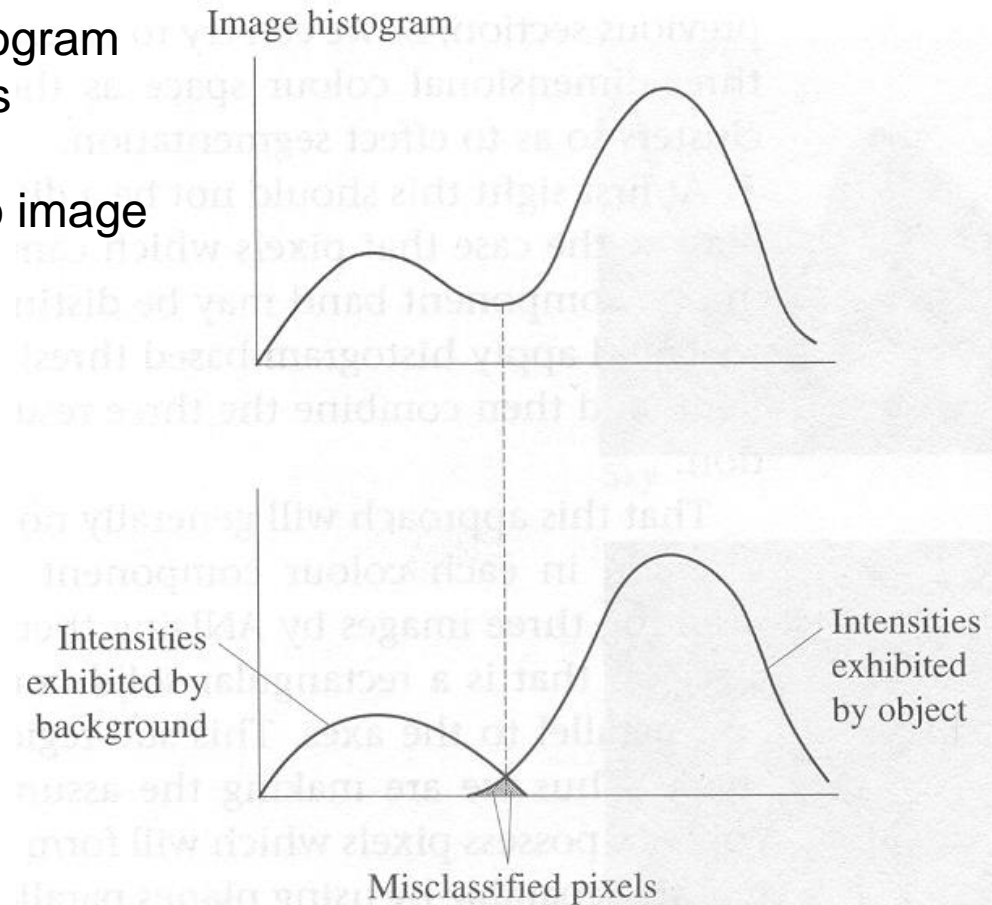
- If the **threshold** value is unknown, analyze the histogram to choose an adequate threshold value
- For a bimodal histogram, the threshold value corresponds to the **valley** between the **peaks**



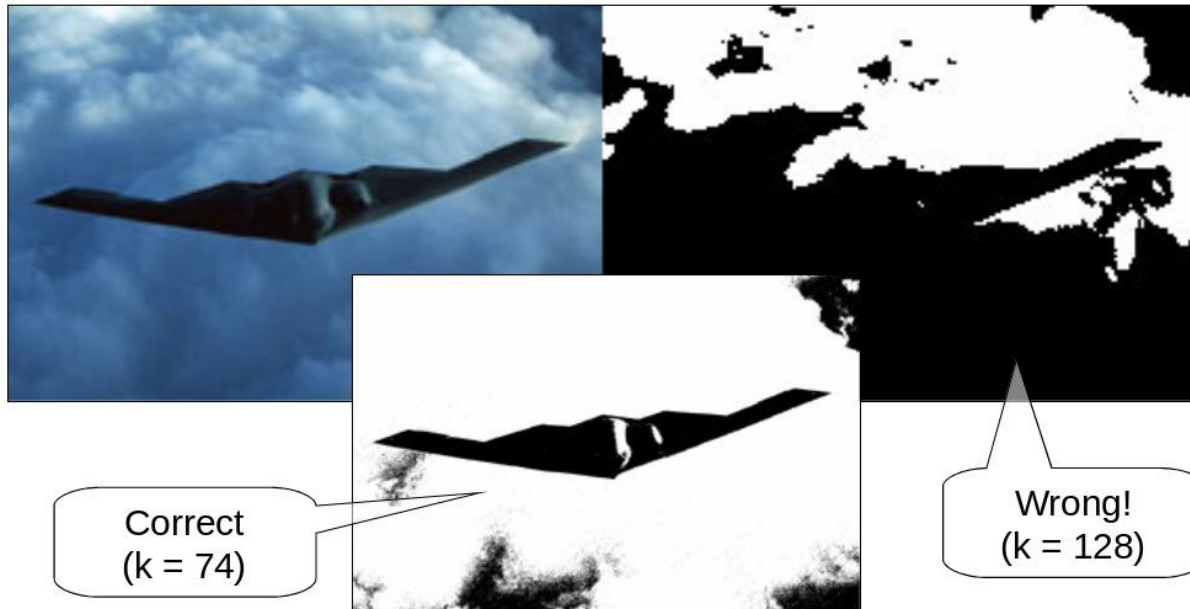
Segmentation - Thresholding



- This approach can produce “**classification errors**”, depending on the image histogram and the intensity values of the objects
- The **thresholding** can be applied to image sub-regions



- Several approaches for threshold selection
 - Global
 - Variable
 - Local - depends on properties of neighbouring pixels
 - Adaptive – depends on spatial coordinates
 - Otsu's method – based on probabilistic analysis obtained from histogram





- Morphological operations
 - Dilation, erosion
 - Opening, closing
- Segmentation
 - Thresholding
 - Region growing



- Grow region by **aggregation of pixels** starting at a **seed point**
- All neighbouring pixels that **comply the rule** are labelled as belonging to the region
- A problem is to obtain “good” seed pixels
- Seed can be obtained using
 - Histograms
 - Interactively
 - ...



- Flood-Filling

1: REGIONLABELING(I)

I : binary image (0 = *background*, 1 = *foreground*)

The image I is labeled (destructively modified) and returned.

2: Initialize $m \leftarrow 2$ (the value of the next label to be assigned).

3: Iterate over all image coordinates (u, v) .

4: **if** $I(u, v) = 1$ **then**

5: FLOODFILL(I, u, v, m) ▷ use any of the 3 versions below

6: $m \leftarrow m + 1$.

7: **return** the labeled image I .

8: FLOODFILL($I, u, v, label$) ▷ **Recursive Version**

9: **if** coordinate (u, v) is within image boundaries **and** $I(u, v) = 1$ **then**

10: Set $I(u, v) \leftarrow label$

11: FLOODFILL($I, u+1, v, label$)

12: FLOODFILL($I, u, v+1, label$)

13: FLOODFILL($I, u, v-1, label$)

14: FLOODFILL($I, u-1, v, label$)

15: **return**.



- Flood-Filling

```
16: FLOODFILL( $I, u, v, label$ )                                ▷ Depth-First Version
17:   Create an empty stack  $S$ 
18:   Put the seed coordinate  $\langle u, v \rangle$  onto the stack: PUSH( $S, \langle u, v \rangle$ )
19:   while  $S$  is not empty do
20:     Get the next coordinate from the top of the stack:
         $\langle x, y \rangle \leftarrow \text{POP}(S)$ 
21:     if coordinate  $(x, y)$  is within image boundaries and  $I(x, y) = 1$ 
        then
22:       Set  $I(x, y) \leftarrow label$ 
23:       PUSH( $S, \langle x+1, y \rangle$ )
24:       PUSH( $S, \langle x, y+1 \rangle$ )
25:       PUSH( $S, \langle x, y-1 \rangle$ )
26:       PUSH( $S, \langle x-1, y \rangle$ )
27:   return.
```

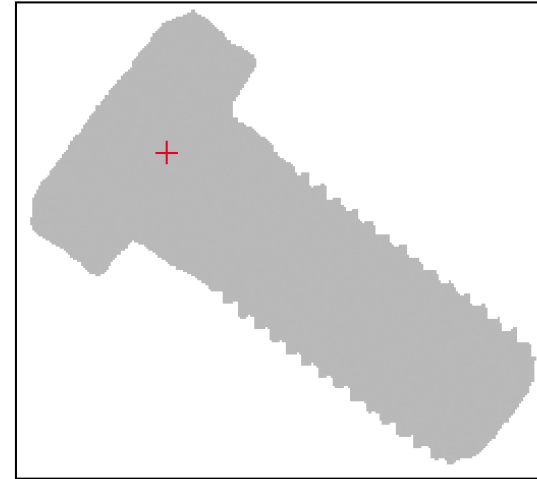


- Flood-Filling

```
28: FLOODFILL( $I, u, v, label$ )                                ▷ Breadth-First Version
29:   Create an empty queue  $Q$ 
30:   Insert the seed coordinate  $\langle u, v \rangle$  into the queue: ENQUEUE( $Q, \langle u, v \rangle$ )
31:   while  $Q$  is not empty do
32:     Get the next coordinate from the front of the queue:
        $\langle x, y \rangle \leftarrow$  DEQUEUE( $Q$ )
33:     if coordinate  $\langle x, y \rangle$  is within image boundaries and  $I(x, y) = 1$ 
       then
34:       Set  $I(x, y) \leftarrow label$ 
35:       ENQUEUE( $Q, \langle x+1, y \rangle$ )
36:       ENQUEUE( $Q, \langle x, y+1 \rangle$ )
37:       ENQUEUE( $Q, \langle x, y-1 \rangle$ )
38:       ENQUEUE( $Q, \langle x-1, y \rangle$ )
39:   return.
```

- Flood-Filling

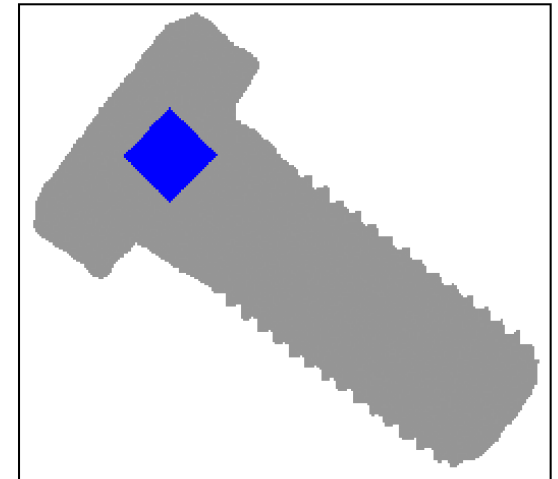
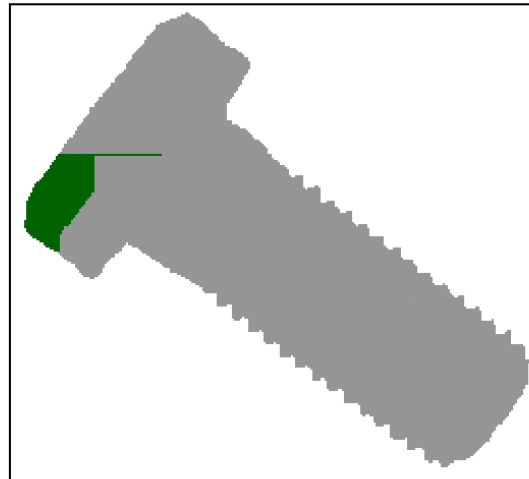
(a)
Original



depth-first

breadth-first

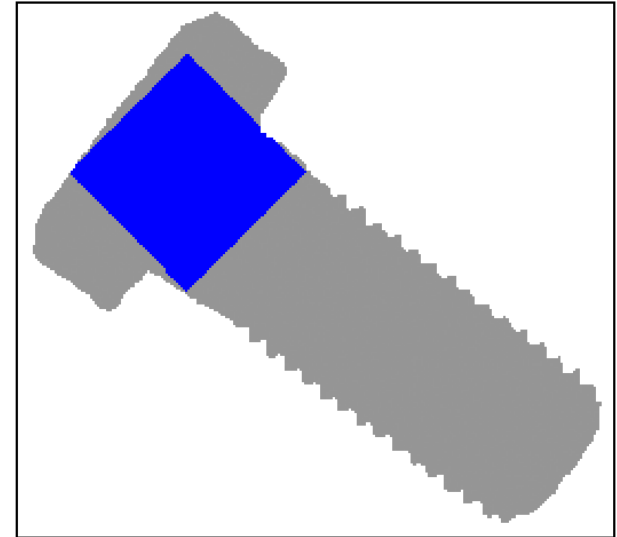
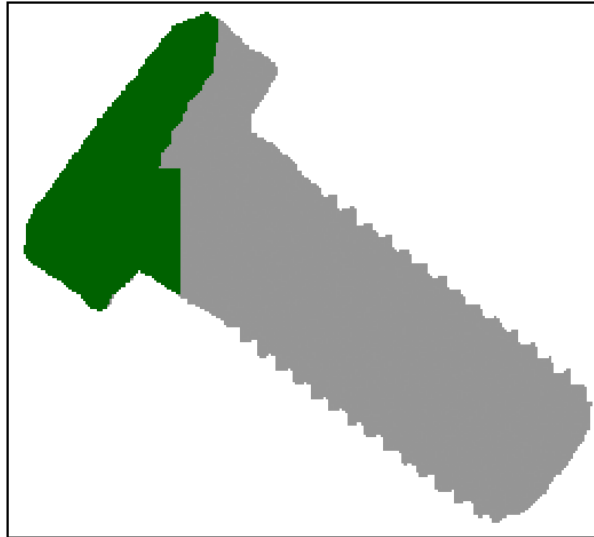
(b)
 $K = 1.000$



- Flood-Filling

(c)

$K = 5.000$



(d)

$K = 10.000$

