Computer Vision

Paulo Dias

deti departamento de electrónica telecomunicações e informática

universidade de aveiro

Summary

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

Summary

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

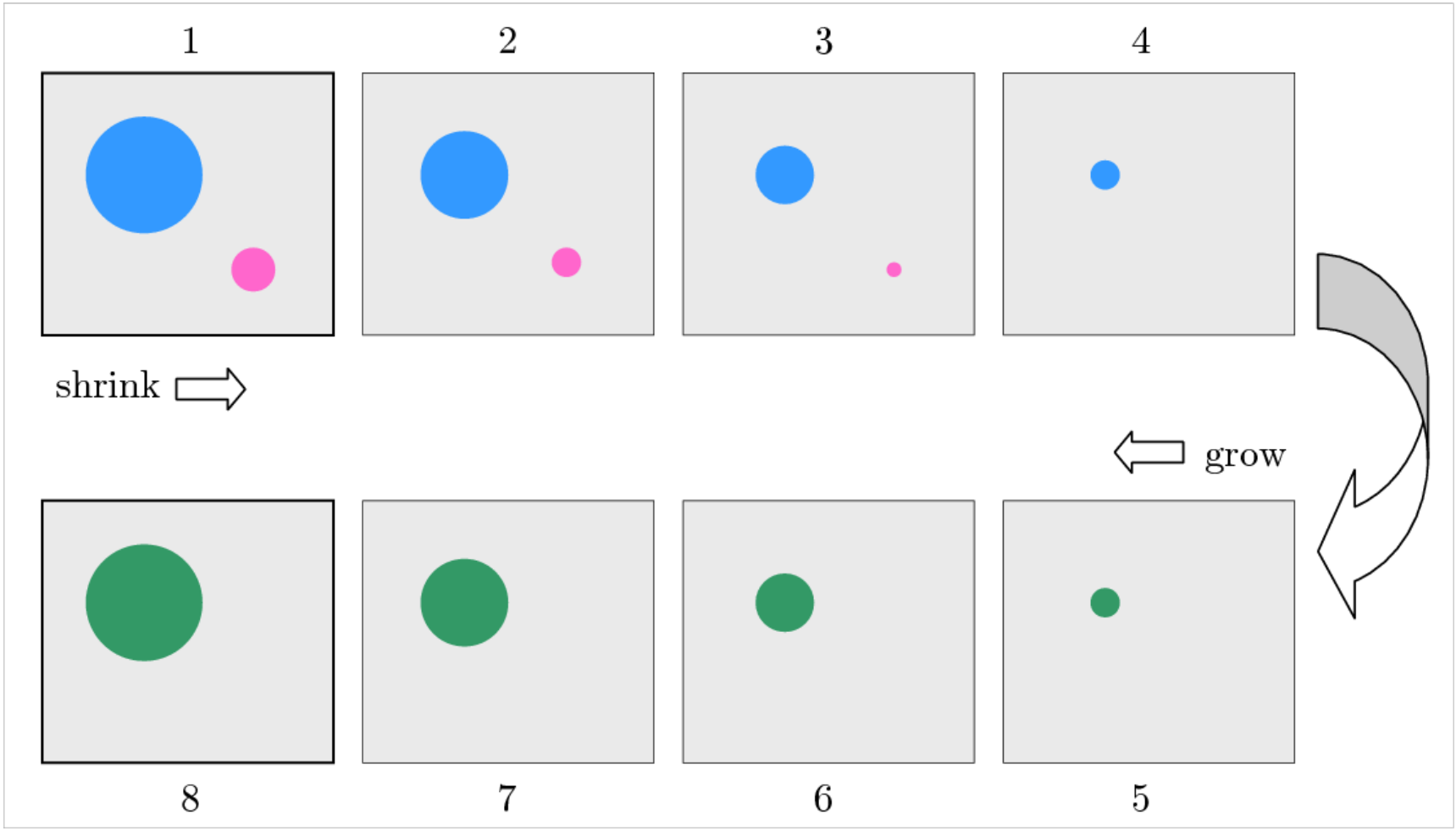
Morphological operations

- 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 ou colour image as well

- Used in Image Processing for
  - Filtering
  - Segmentation
  - Object description

Morphological operations

- Main idea

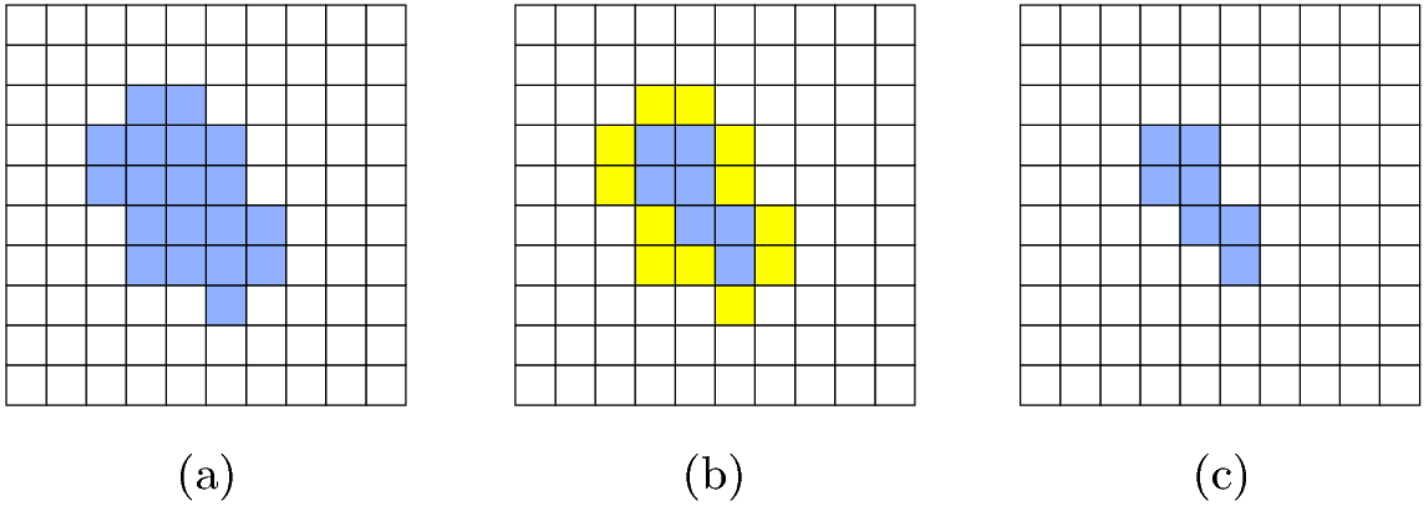


Burger and Burge

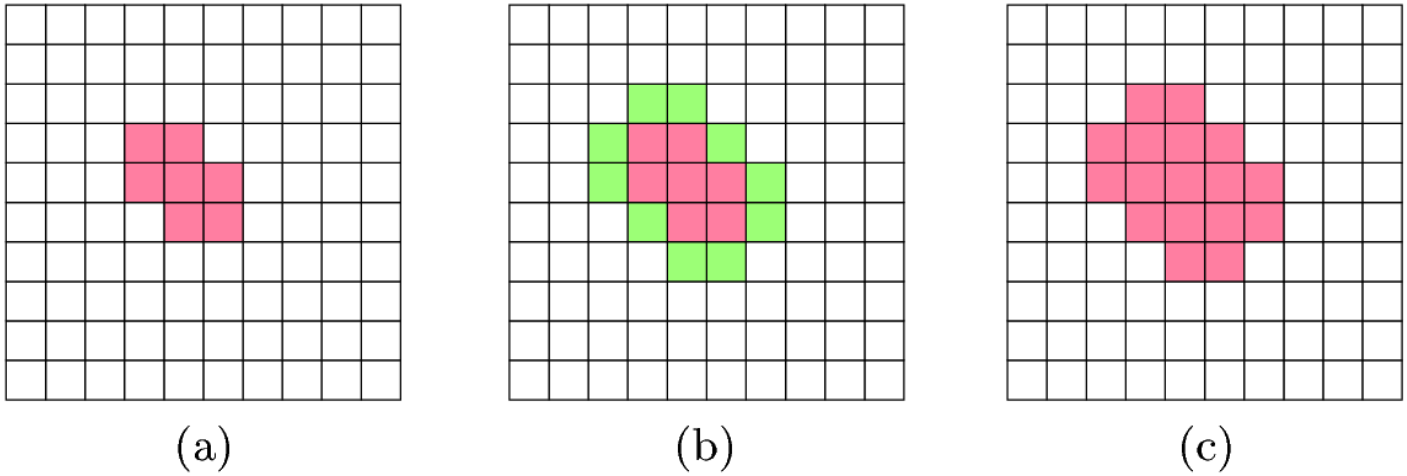
Morphological operations

- Main idea

Erosion



Dilation

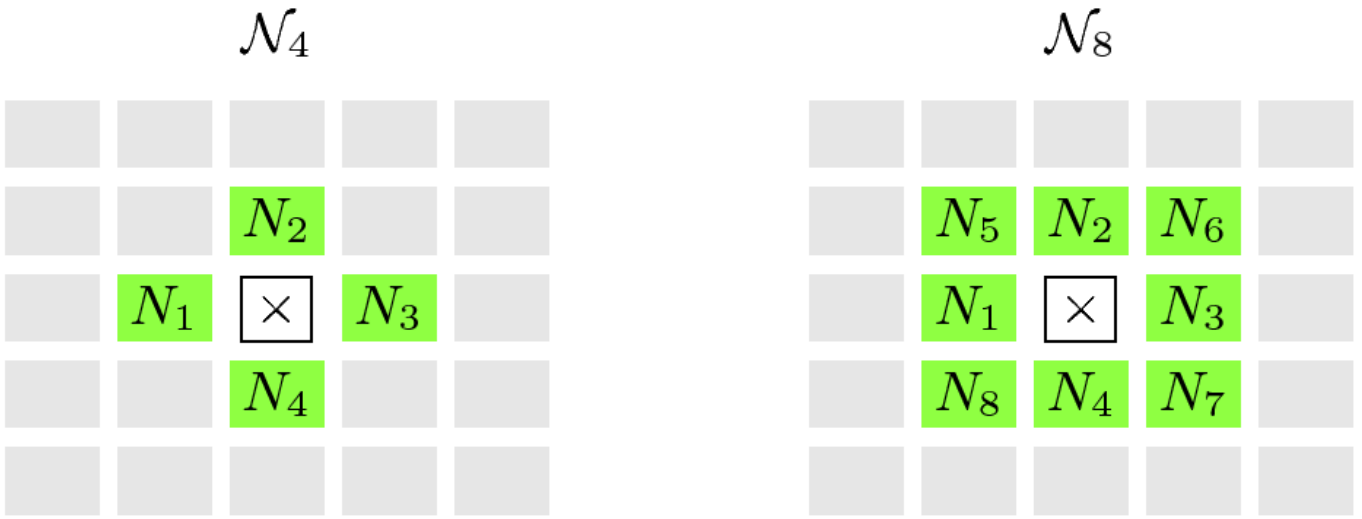


Burger and Burge

Morphological operations

- Structuring element

- Neighbourhood



Burger and Burge

Morphological operations

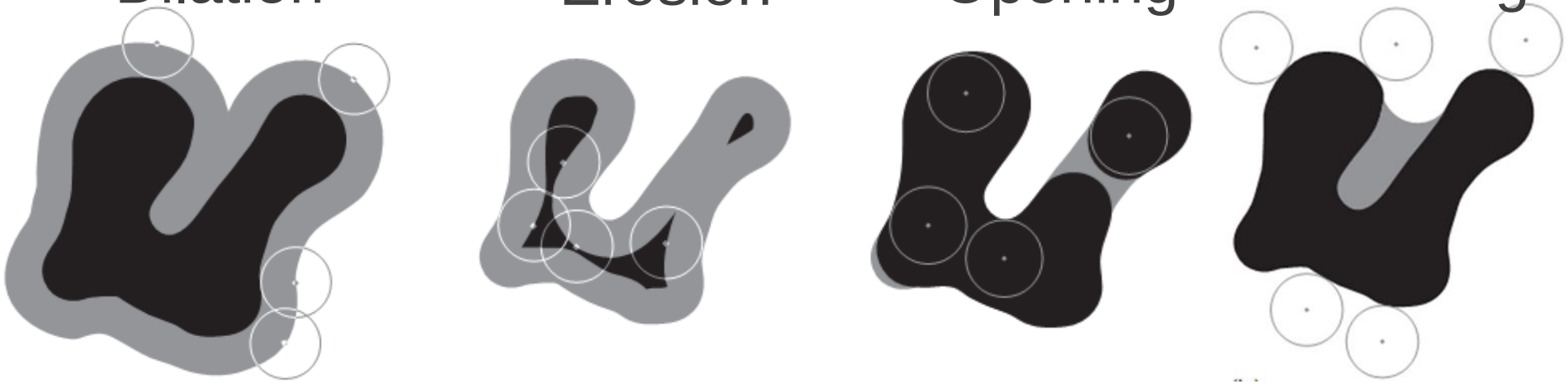
- Main Morphological operations
  - Dilation
  - Erosion

  } Basic Operations

  - Opening
  - Closing

  } Composed Operations

Dilation Erosion Opening Closing



Morphological operations - Dilation

- 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

1	1

Structuring element

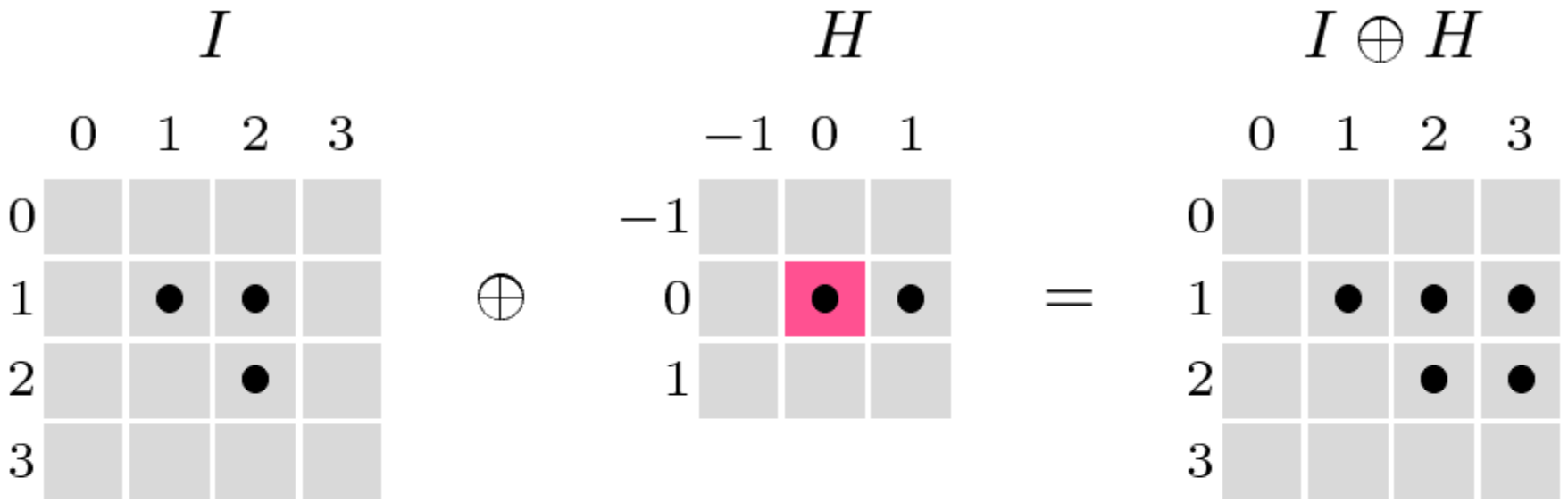
0	1	1	0	0
0	1	1	0	0
0	1	1	0	0
0	1	1	1	1
0	0	0	0	0

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

$I \oplus X$

Morphological operations - Dilation



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

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

Morphological operations - Erosion

- 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

1	1

Structuring element

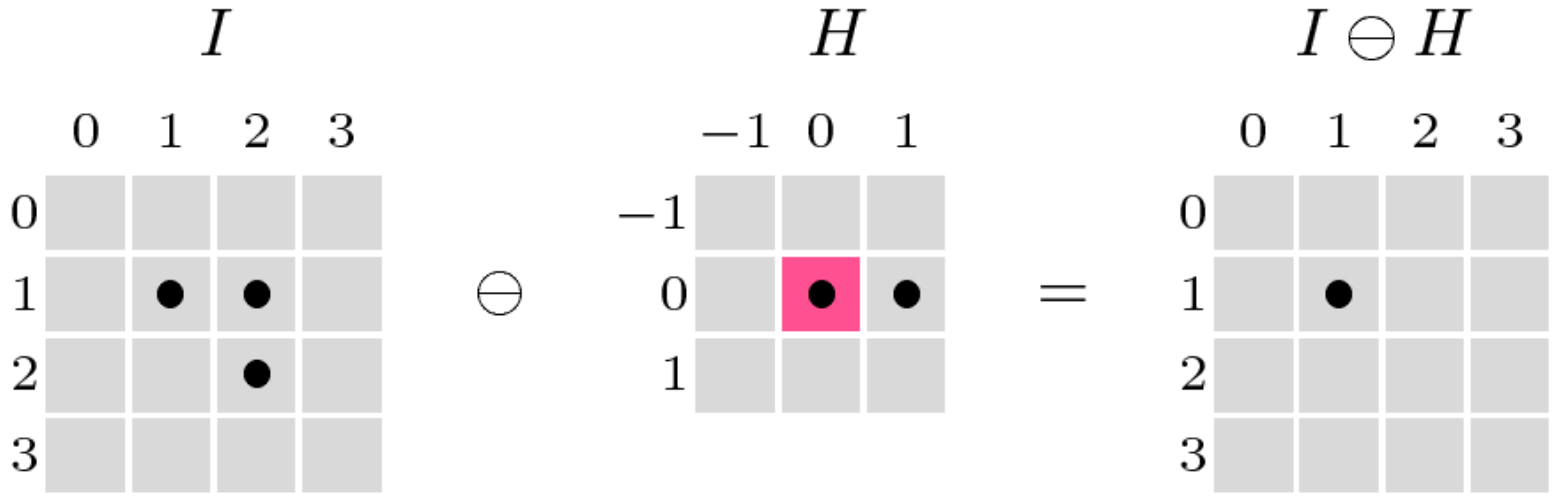
0	1	0	0	0
0	1	0	0	0
0	1	0	0	0
0	1	1	1	0
0	0	0	0	0

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

$I \ominus X$

Morphological operations - Erosion



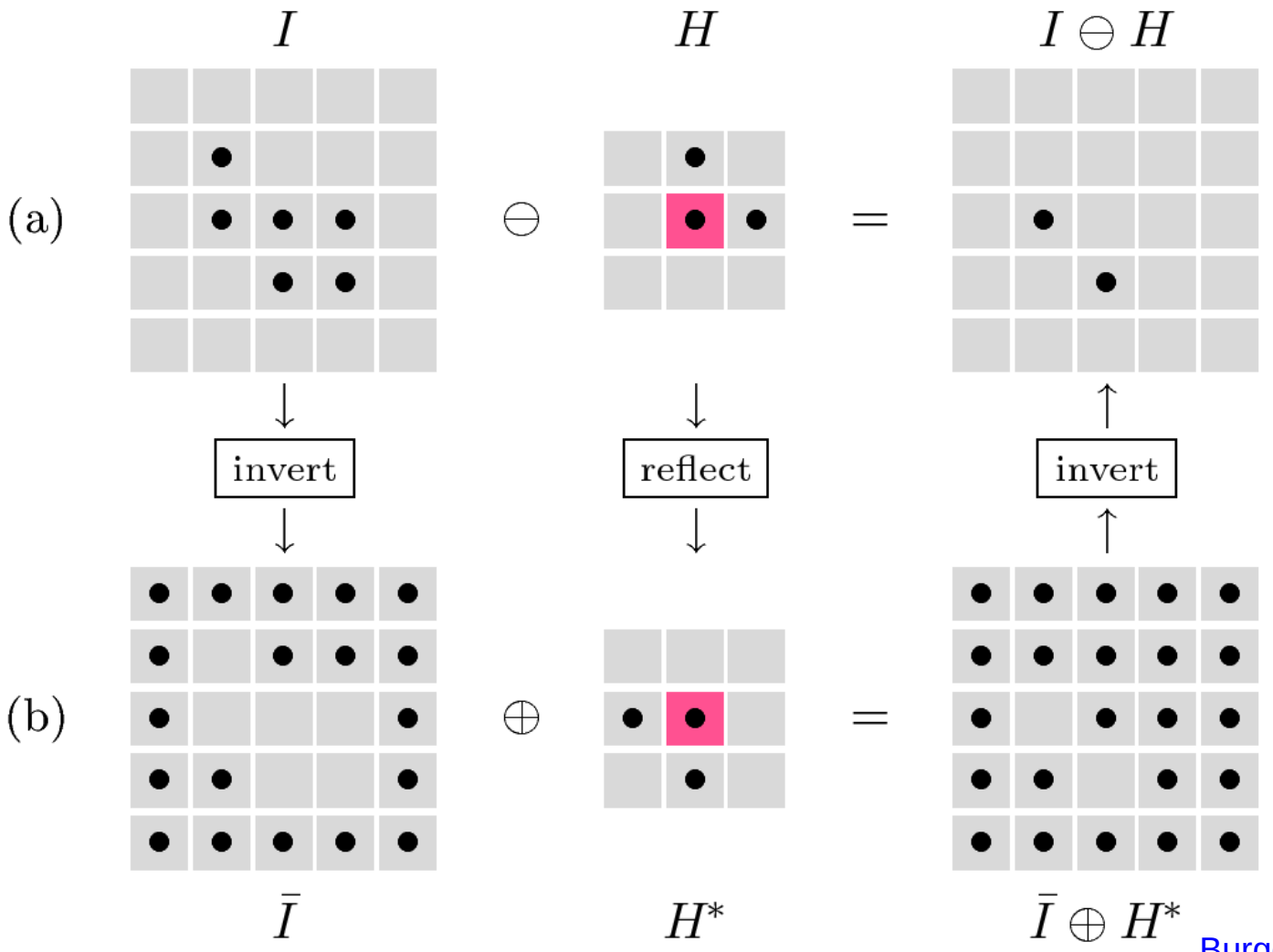
$$I \equiv \{(1,1),(2,1),(2,2)\},\ H \equiv \{(\mathbf{0},\mathbf{0}),(\mathbf{1},\mathbf{0})\}$$

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

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

Morphological operations

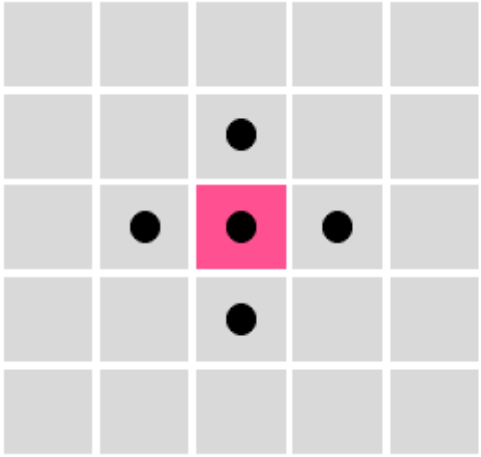
- Dilation and Erosion are dual operations

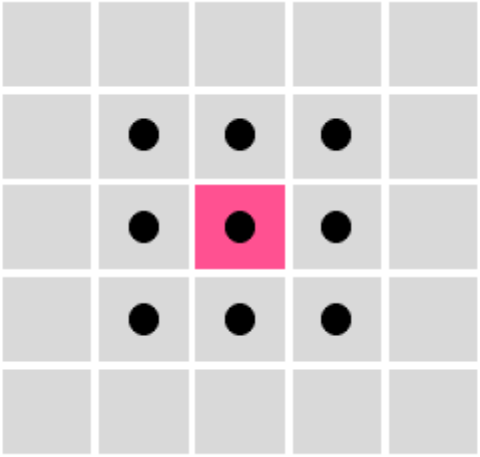


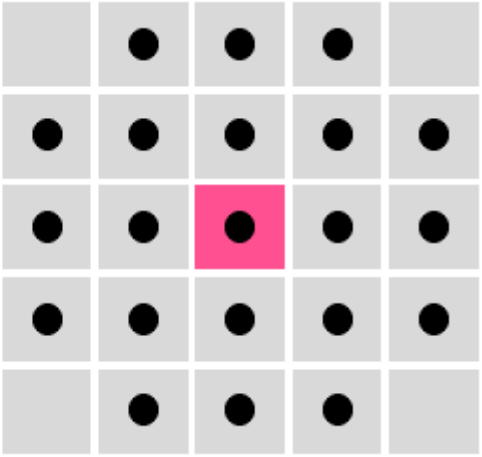
Burger and Burge

Morphological operations

- Typical structuring elements







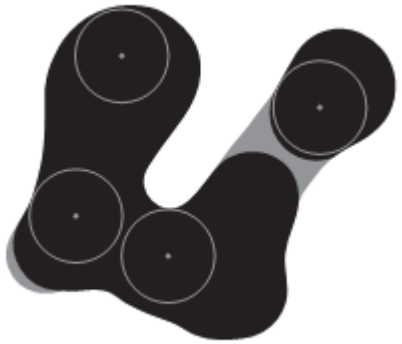
Burger and Burge

Summary

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

Morphological operations - Opening

- 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



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

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

1	1

Structuring element

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

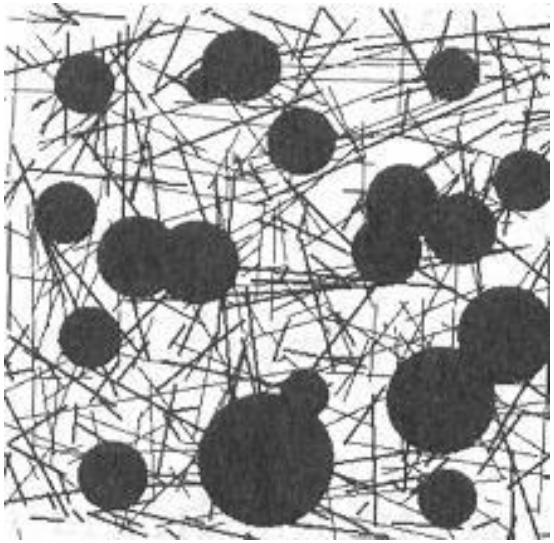
Opening

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

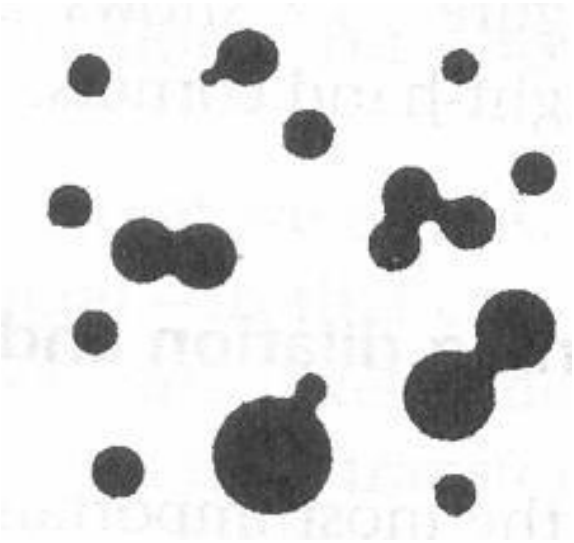
$I \bigcirc X$

Morphological operations - Opening

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



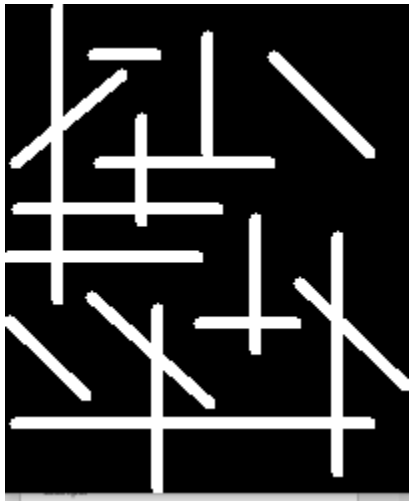
Original image

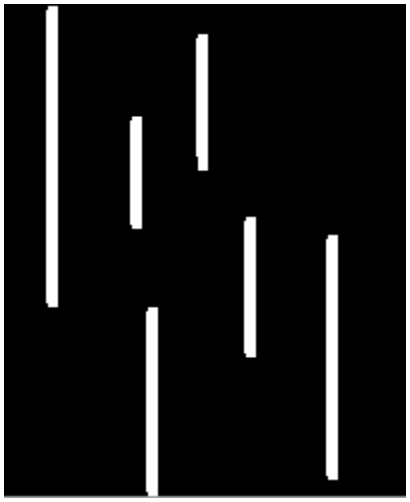


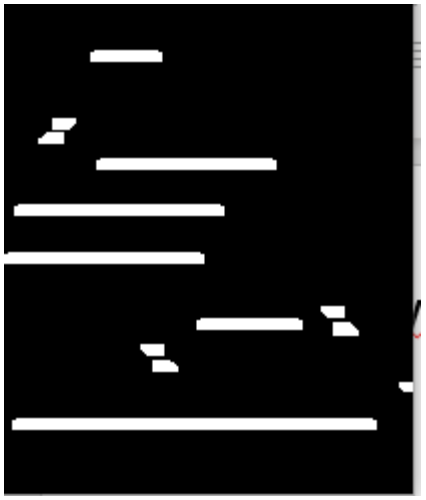
After erosion

After dilation

Opening with different structuring elements

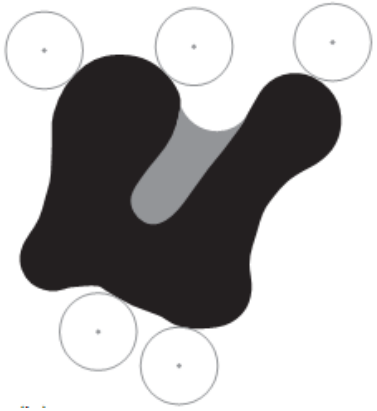






Morphological operations - Closing

- 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

1	1

Structuring element

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

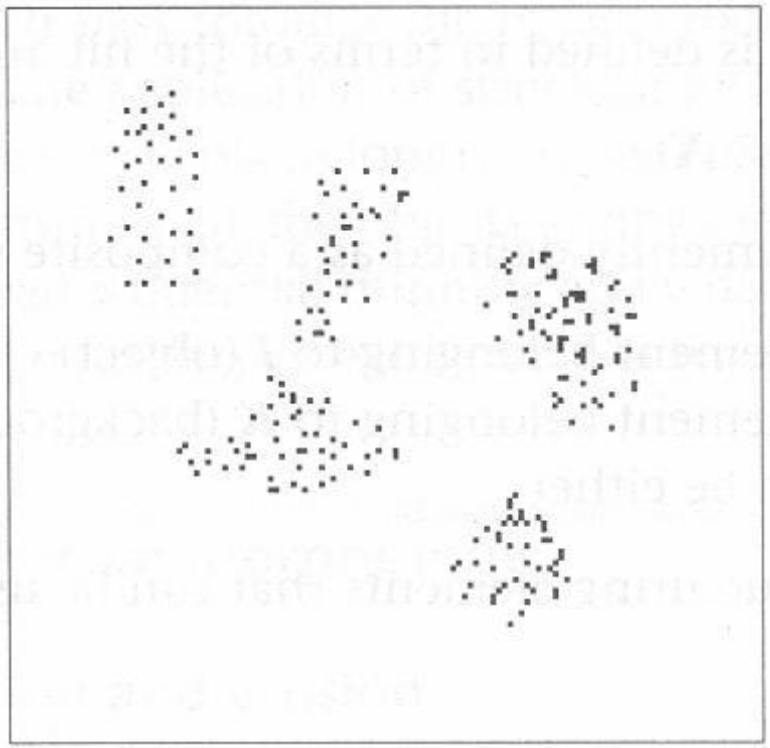
Closing

?

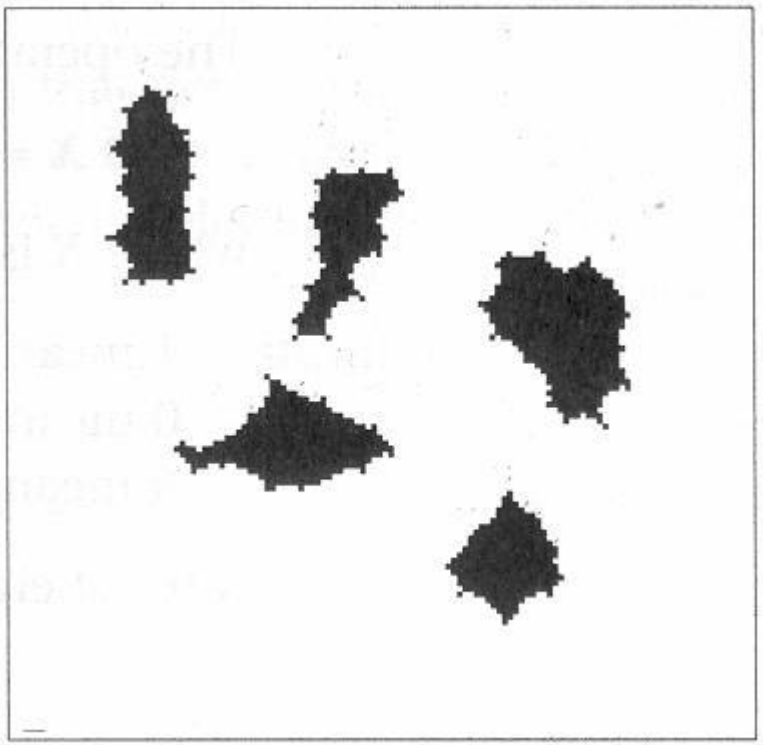
$I \bullet X$

Morphological operations - Closing

- Closing example



Original image



After closing

Morphological operations

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

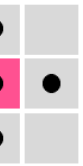
$$Edge = I - (I \ominus X)$$

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

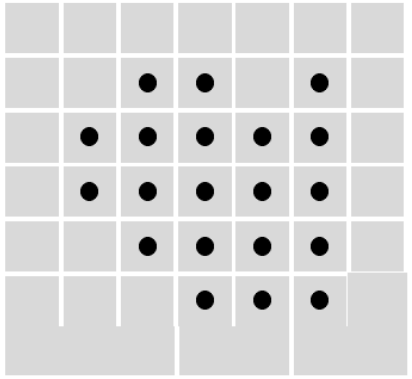
Morphological operations

- Edge detection with morphology (outlining)

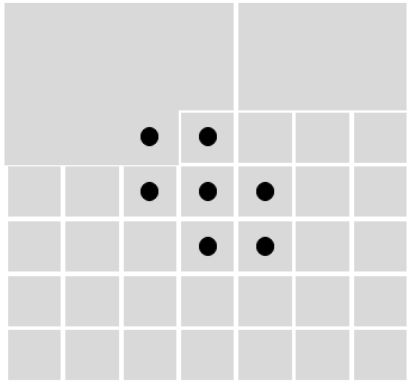
$H_n =$



I



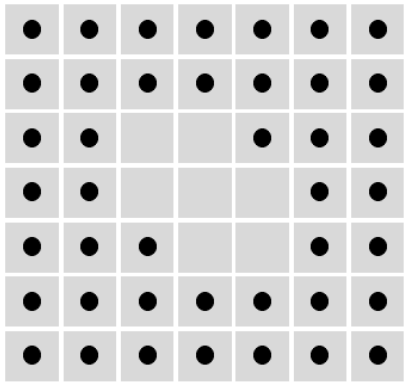
$I \ominus H_n$



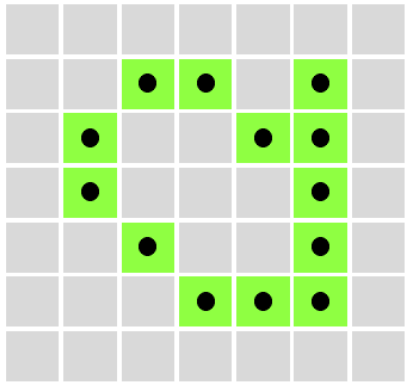
$$I' = I \ominus H_n$$

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

$\overline{I \ominus H_n}$



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



Summary

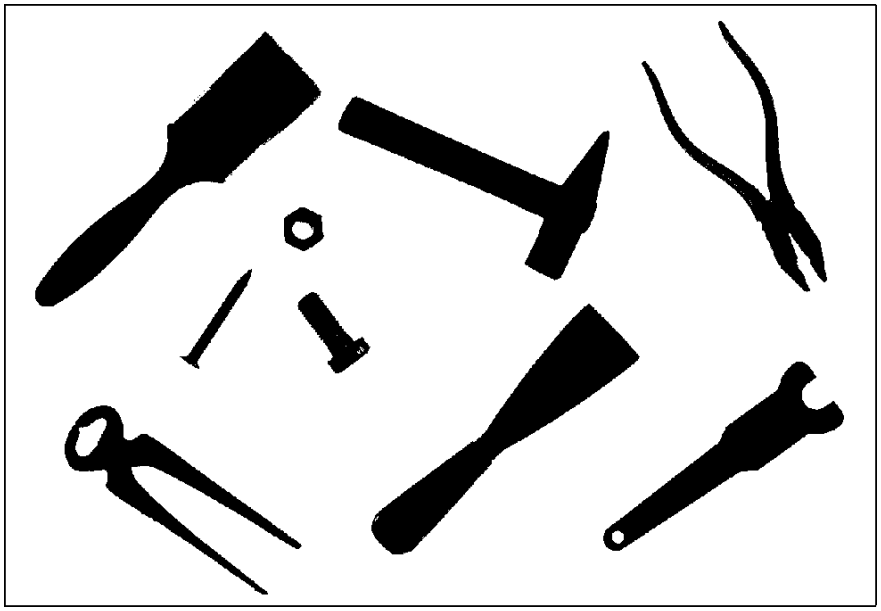
- Morphological operations
  - Dilation, erosion
  - Opening, closing
- Segmentation
  - Thresholding
  - Region growing

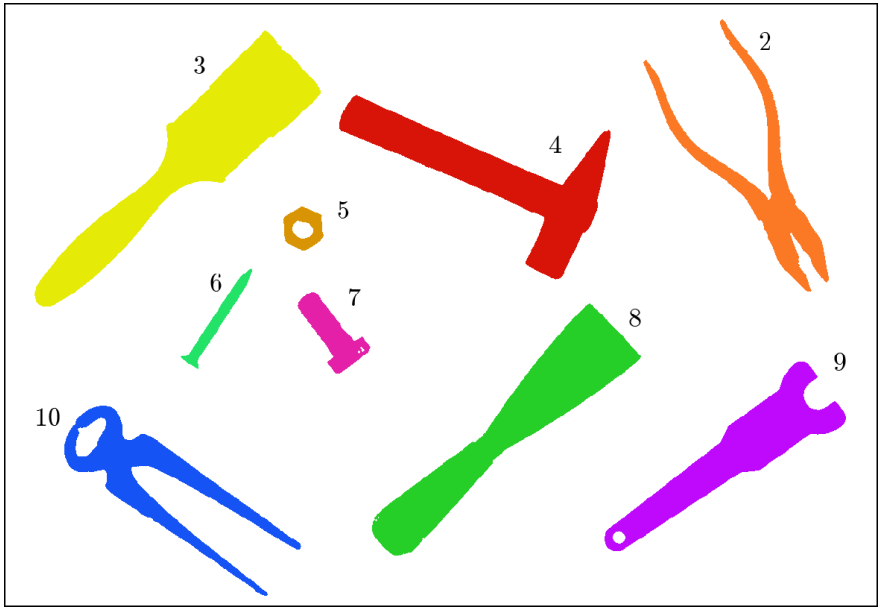
Segmentation

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

Segmentation

- Examples





Burger and Burge

Segmentation

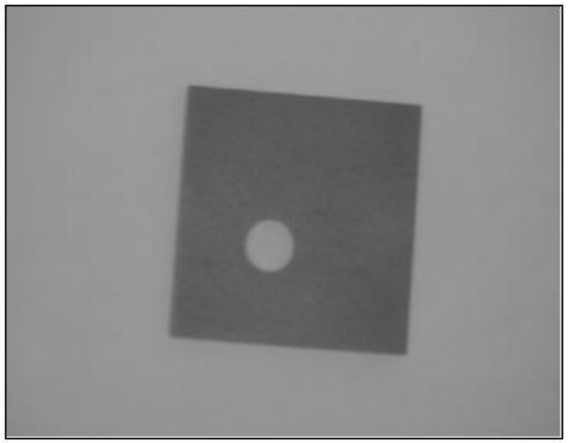
- No segmentation methods that can be used in every case
- No “perfect” segmentation method
- Typical segmentation are based in:
  - pixel intensity
  - regions
  - edges

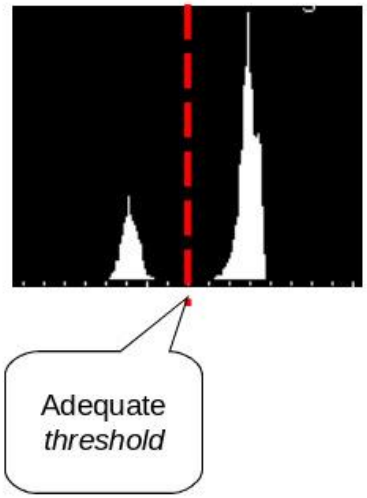
Summary

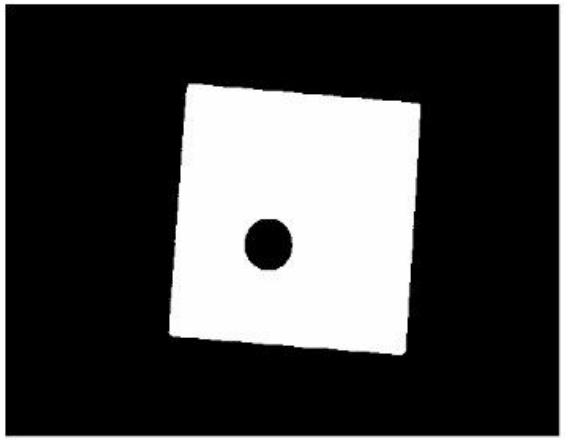
- Morphological operations
  - Dilation, erosion
  - Opening, closing
- Segmentation
  - Thresholding
  - Region growing

Segmentation - Thresholding

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





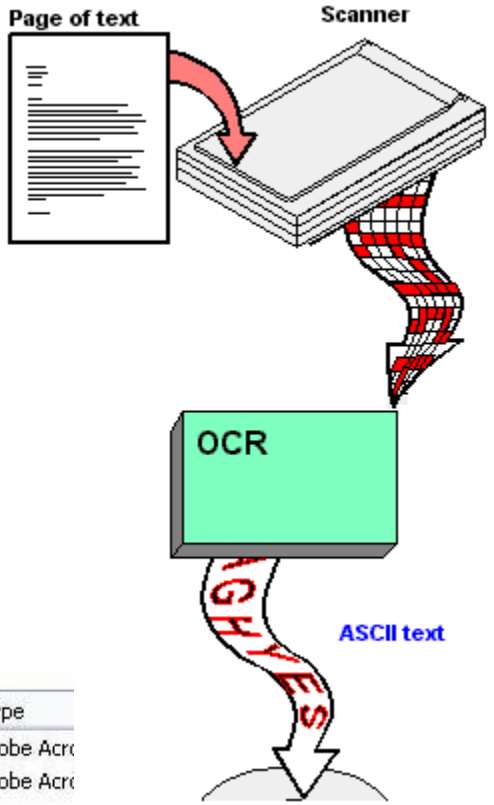


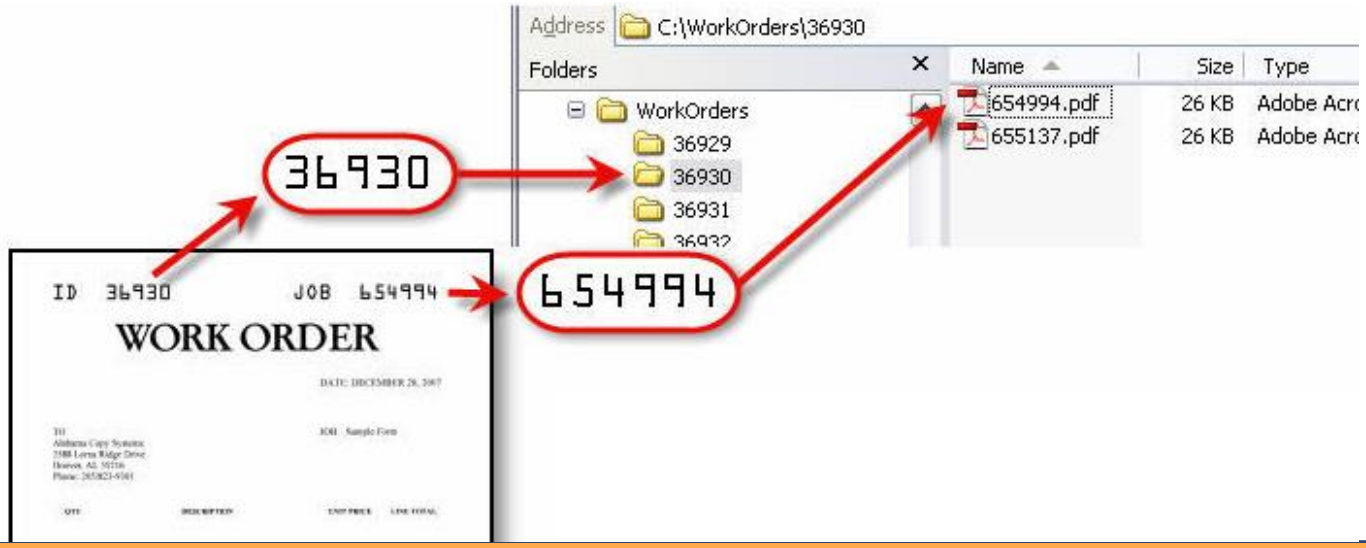
Segmentation - Thresholding

- 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*)

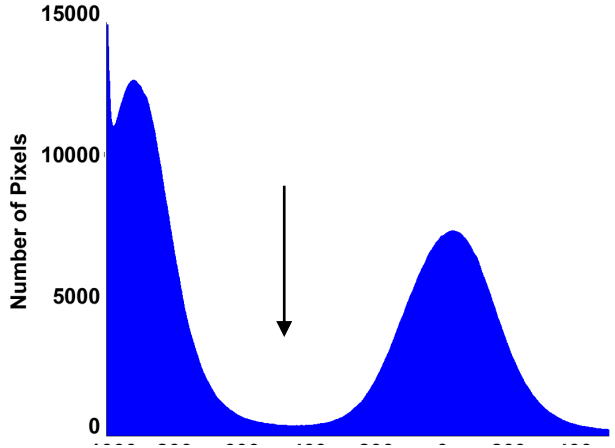


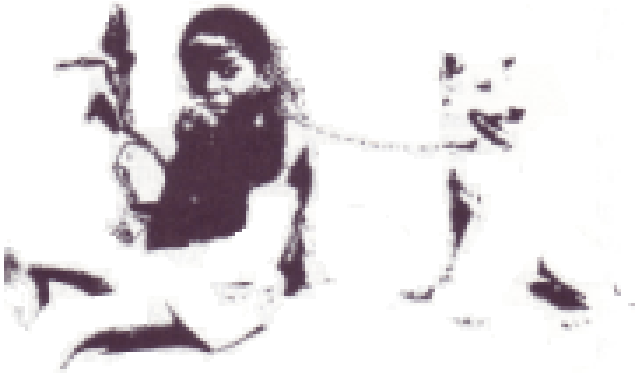


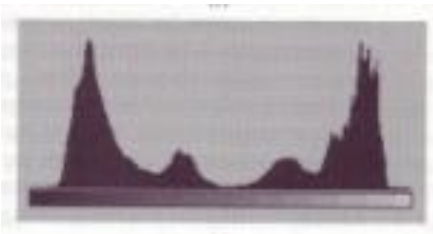
Segmentation - Thresholding

- 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



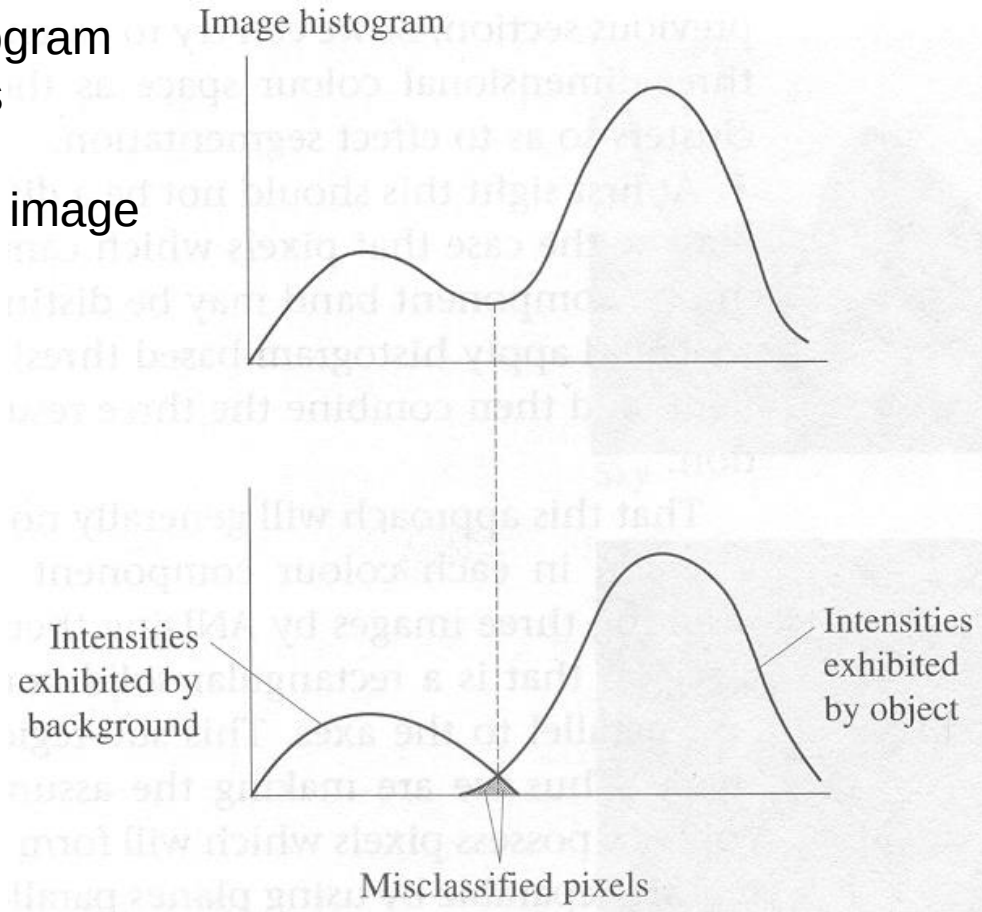




Segmentation - Thresholding

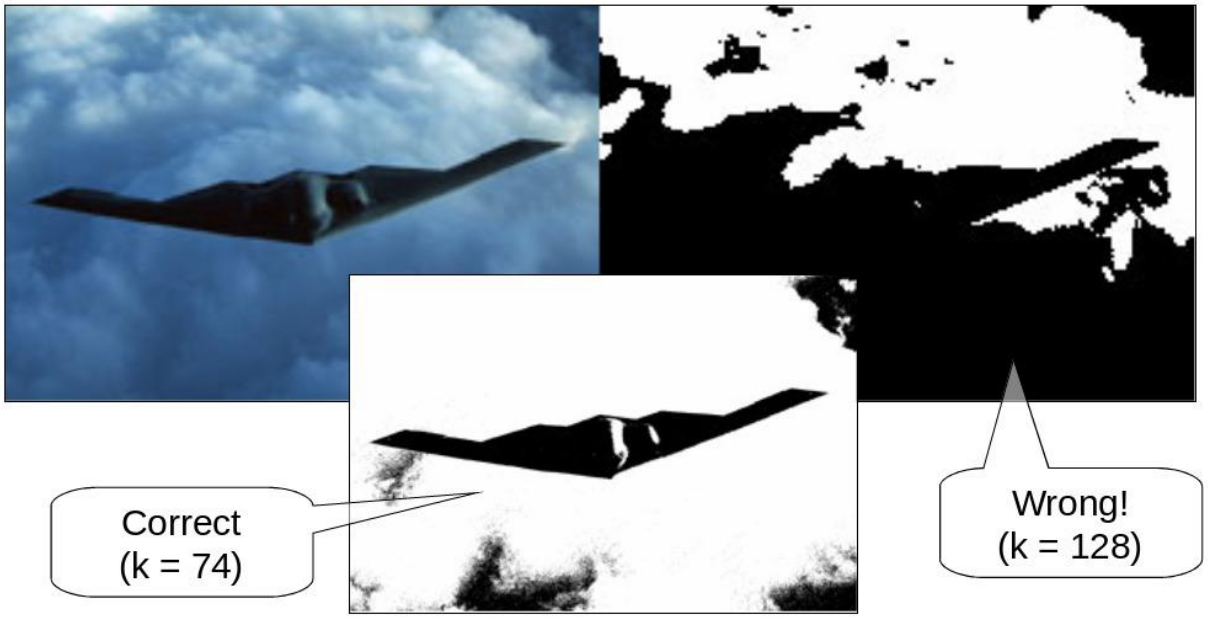
- This approach can produce "classification errors", depending on the image histogram and the intensity values of the objects

- The *thresholding* can be applied to image sub-regions



Segmentation - Thresholding

- 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



Summary

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

Segmentation – 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
  - …

Segmentation – Region growing

- Flood-Filling

```
1:  RegionLabeling(I)
        I: binary image (0 = background, 1 = foreground)
        The image I is labeled (destructively modified) and returned.

2:      Initialize m ← 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 ← 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) ← 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.
```

Segmentation – Region growing

- 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:
$\langle x, y \rangle \leftarrow$ 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**.

Segmentation – Region growing

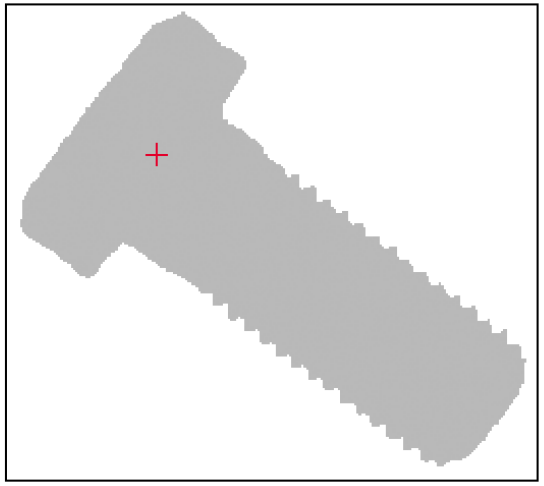
- Flood-Filling

```
28: FloodFill(I, u, v, label)                    ▷ Breadth-First Version
29:     Create an empty queue Q
30:     Insert the seed coordinate ⟨u, v⟩ into the queue: Enqueue(Q, ⟨u, v⟩)
31:     while Q is not empty do
32:         Get the next coordinate from the front of the queue:
                ⟨x, y⟩ ← Dequeue(Q)
33:         if coordinate ⟨x, y⟩ is within image boundaries and I(x, y) = 1
                then
34:             Set I(x, y) ← label
35:             Enqueue(Q, ⟨x+1, y⟩)
36:             Enqueue(Q, ⟨x, y+1⟩)
37:             Enqueue(Q, ⟨x, y−1⟩)
38:             Enqueue(Q, ⟨x−1, y⟩)
39:     return.
```

Segmentation – Region growing

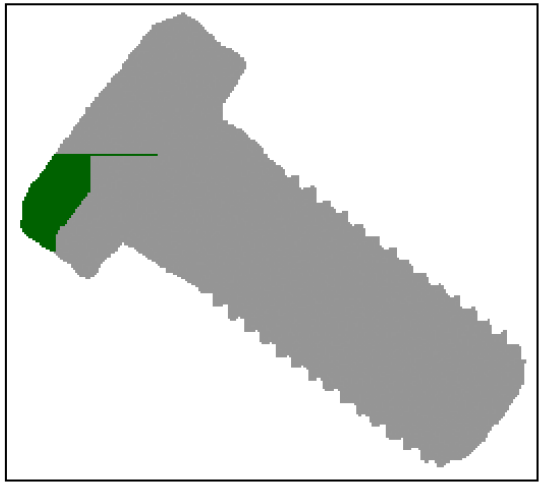
- Flood-Filling

(a)
Original

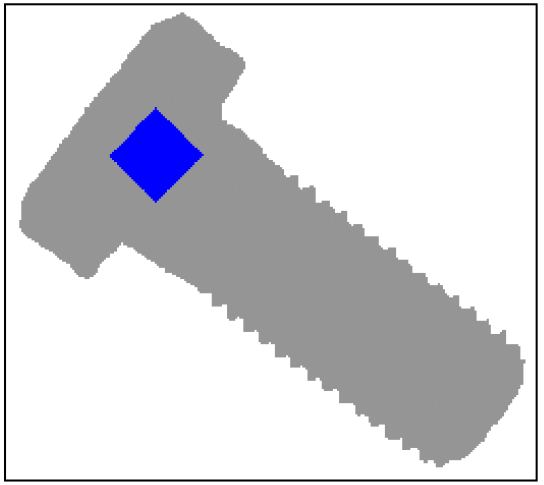


(b)
$K = 1.000$

depth-first



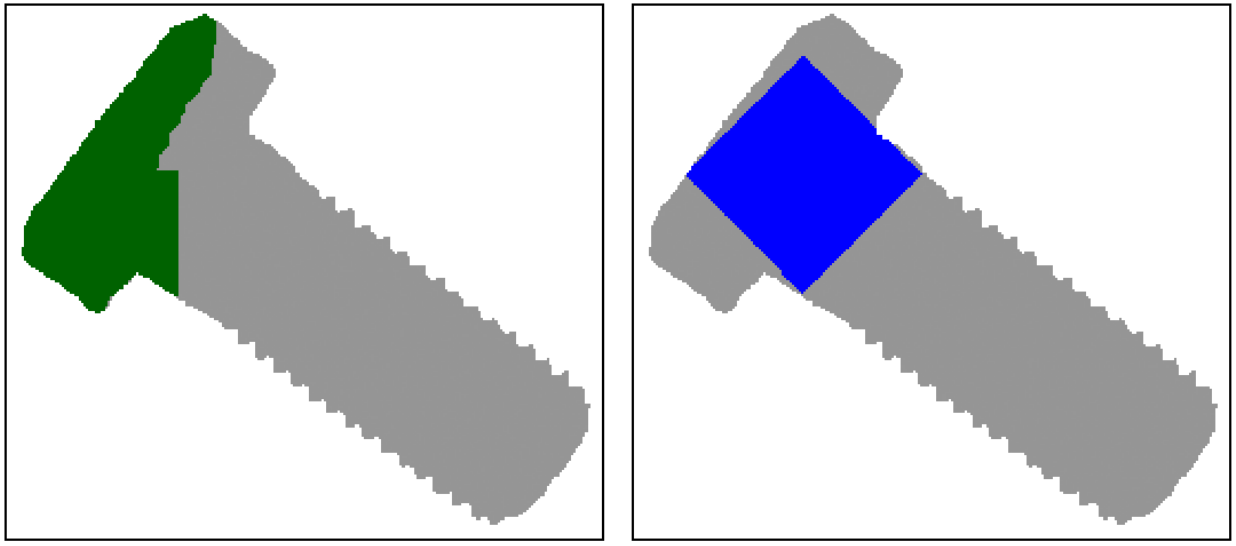
breadth-first



Segmentation – Region growing

- Flood-Filling

(c)
$K = 5.000$



(d)
$K = 10.000$