UNIVERSITÀ DI PARMA

Binary Images
Morphological Processing

Summary

UNIVERSITÀ DI PARMA

- Binary images
- Thresholding
- Mathematic Morphology
  - (also for non binary images)
- Connected components extraction

UNIVERSITÀ DI PARMA

Binary Images

Binary Images

UNIVERSITÀ DI PARMA

- In some cases it is necessary/useful to acquire/produce and/or process images whose pixels can only have 0/1 logical values
  - PBM or XBM image formats

```
0001001000 1000
0001111000 1000
0001001000 1000
```

Binary Images

UNIVERSITÀ DI PARMA

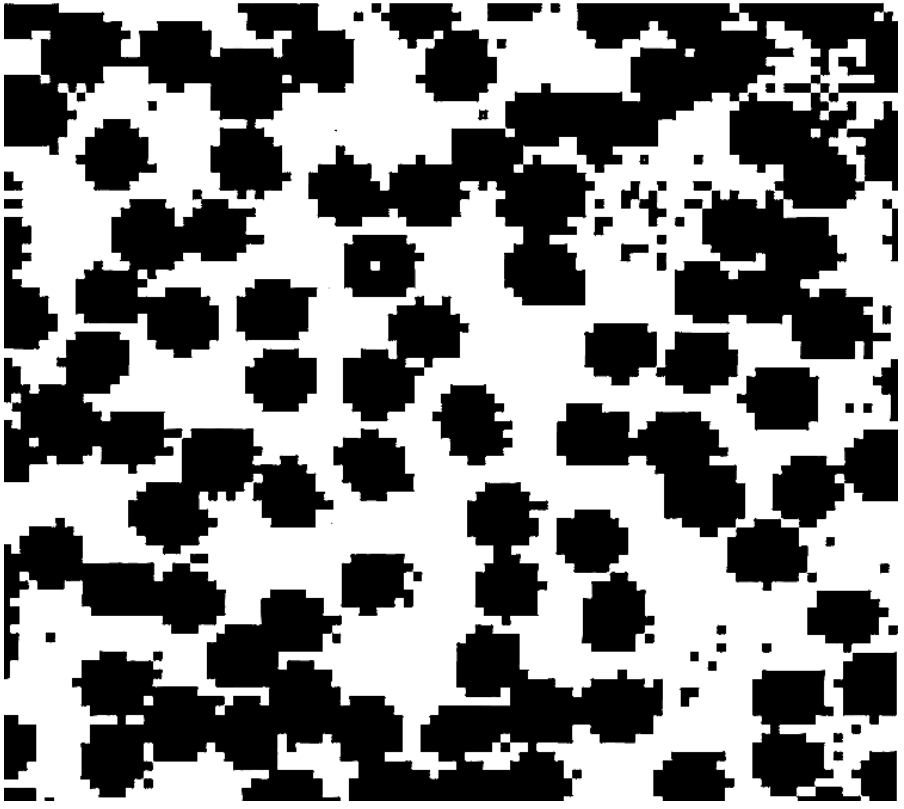
- 1 actually can be considered as !=0
- 255 is often used for simplicity

0	0	0	255	0	0	255	0	0	0	255	0	0	0
0	0	0	255	255	255	255	0	0	0	255	0	0	0
0	0	0	255	0	0	255	0	0	0	255	0	0	0

Example

UNIVERSITÀ DI PARMA

- Why binary images?
  - They are simpler to process
    - Example: Image of blood cells
      - We would like to estimate red blood cells
      - Anyway they are not isolated (<50%)
      - How we can separate them?
  - Often output of other processing techniques
    - i.e. edge detection

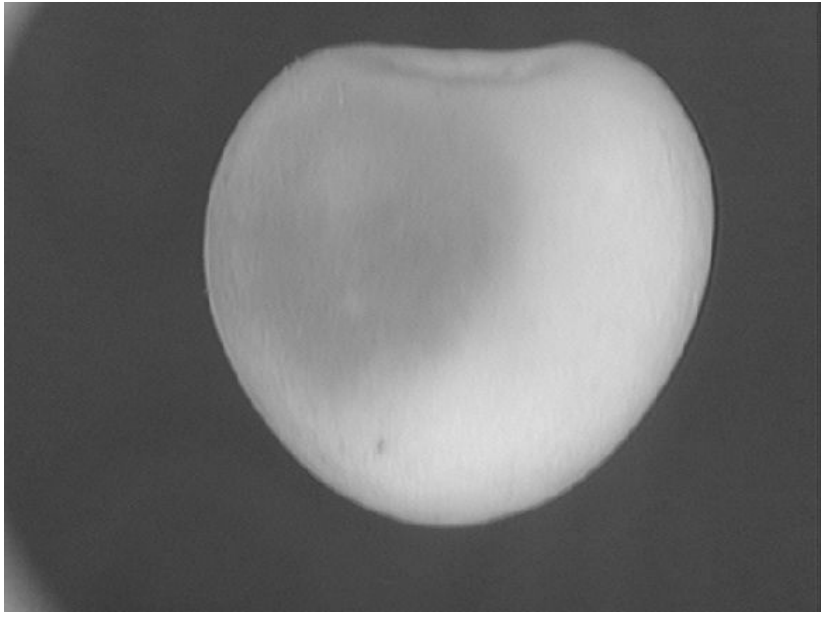


How to obtain binary images?

UNIVERSITÀ DI PARMA

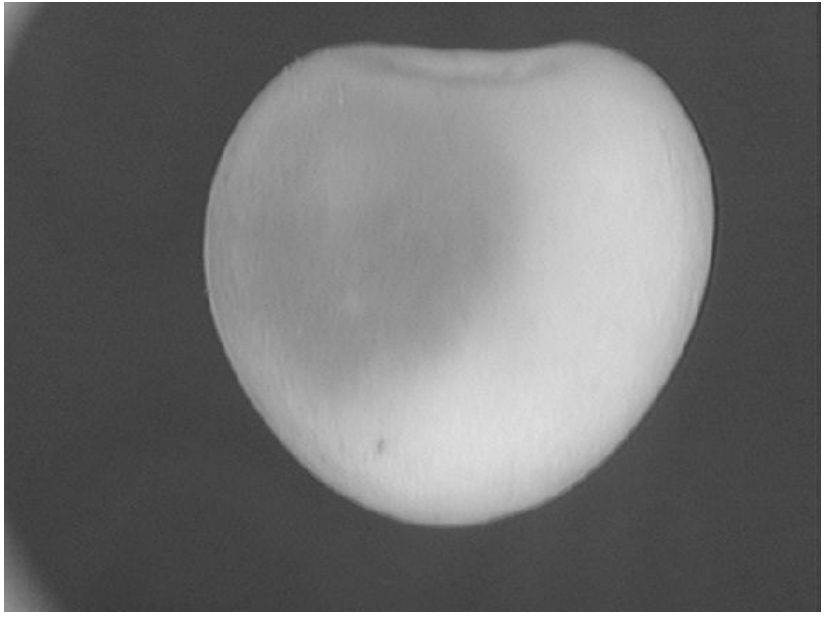
- Basically no B/W cameras
- Besides specific processing $\longrightarrow$ thresholding
  - Pixel above a specific value $\longrightarrow$ 1
  - Pixel below a specific value $\longrightarrow$ 0

- We would like to discard a bruised apple
- How we can select the bruised part?
  - Extract “dark” pixels
- We can only use pixel brightness
- Thresholding:



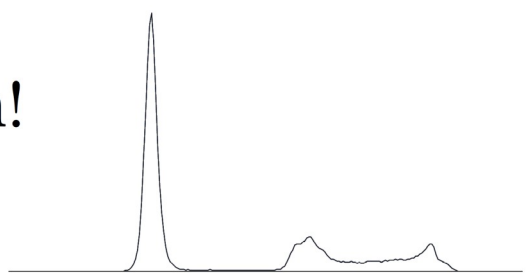
$$out(r,c)=\begin{cases}0 & where \quad in(r,c) < th\\ 1 & where \quad in(r,c) \geq th\end{cases}$$

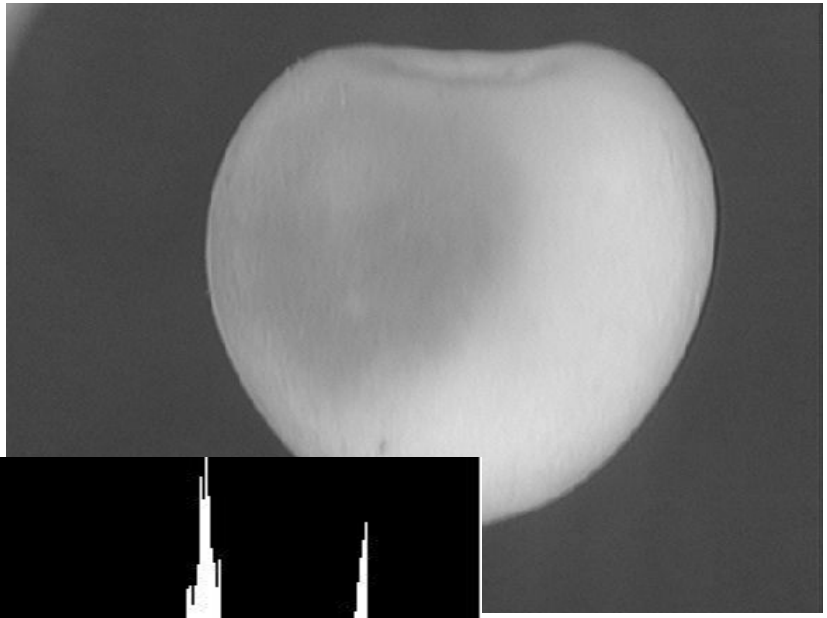
- Anyway we have:
  - “Good” apple part
  - Bruised apple part
  - Background
- How we decide about the threshold?



$$out(r,c)=\begin{cases}0 & where \quad in(r,c) < th \\ 1 & where \quad in(r,c) \geq th\end{cases}$$

- Anyway we have:
  - “Good” apple part
  - Bruised apple part
  - Background
- How we decide about the threshold?
- Histogram!



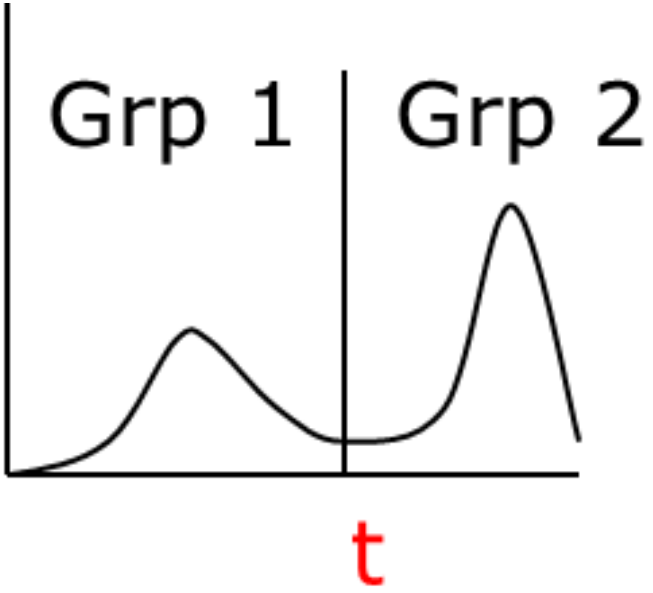


Otsu's Method

UNIVERSITÀ DI PARMA

- Best threshold computation
- Problem: find a *th* that minimize weighted sum of each group variance

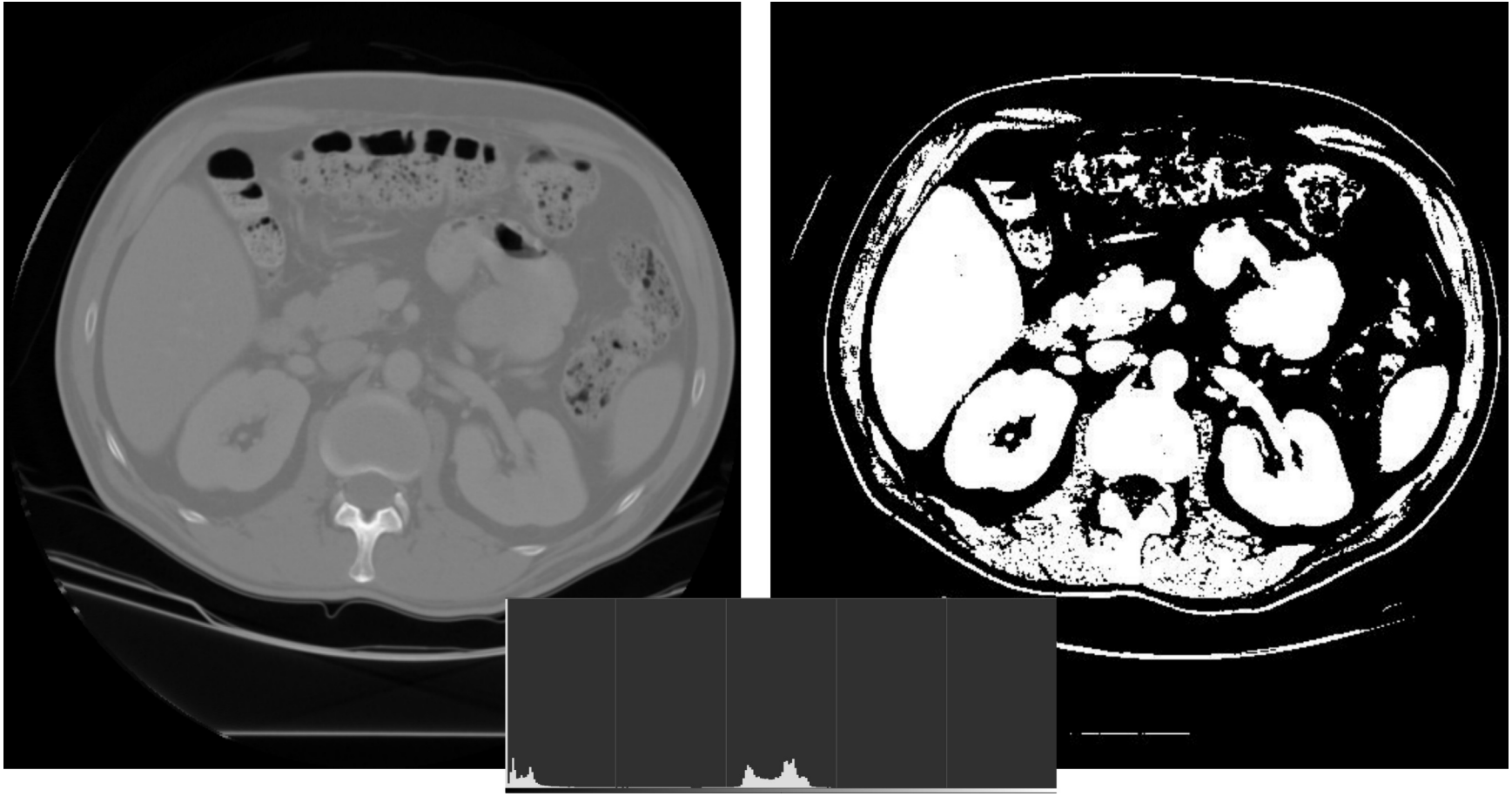
$$\sigma^2(th)=\sigma_1^2(th)\cdot w_1(th)+\sigma_2^2(th)\cdot w_2(th)$$



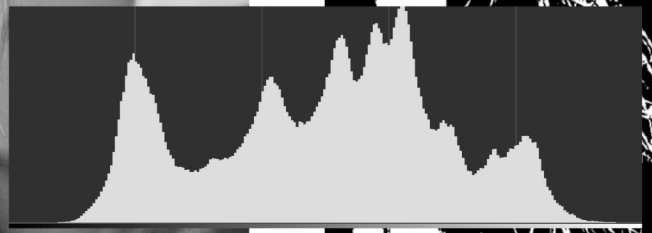
- $w_1(th)$ & $w_2(th)$ are the probability of belonging to group 1 or 2

$$w_1(th)=\sum_{i-0}^{th-1} h(i) \qquad w_2(th)=\sum_{i=th}^{L} h(i)$$

Otsu's Method



Otsu's Method



UNIVERSITÀ DI PARMA

Morphological Processing

Mathematical Morphology

UNIVERSITÀ DI PARMA

- Used to extract image components that are useful in the representation and description of region shape, such as
  - boundaries extraction
  - skeletons
  - convex hull
  - morphological filtering
  - thinning
  - pruning

Mathematical Morphology

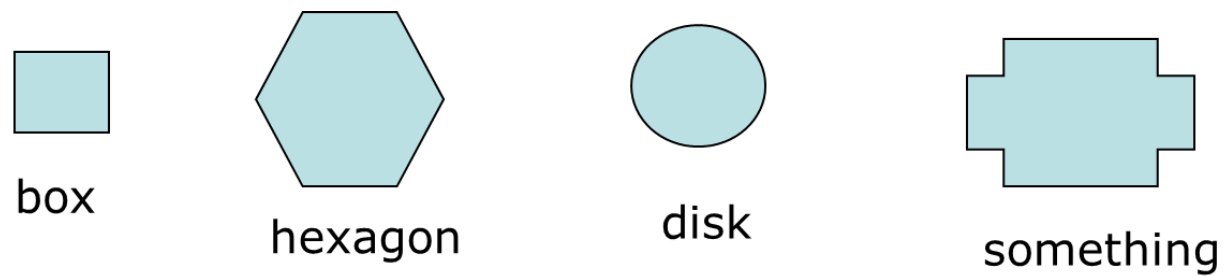
UNIVERSITÀ DI PARMA

- Main operations
  - Erosion $\longrightarrow \ominus$
  - Dilation $\longrightarrow \oplus$
  - Opening $\longrightarrow \circ$
  - Closing $\longrightarrow \bullet$
  - Hit or Miss $\longrightarrow \otimes$

Structuring Element

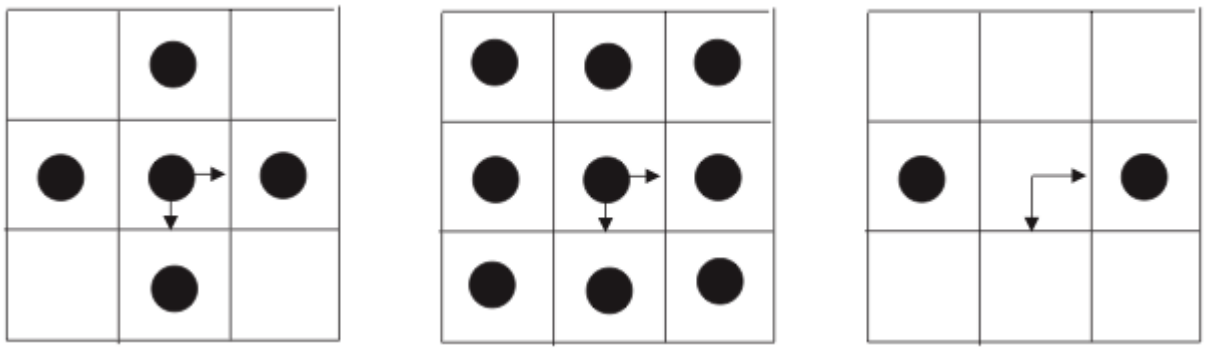
UNIVERSITÀ DI PARMA

- Small “mask” to probe the image under study
  - Convolution like approach
- Structuring Element features an origin
- Shape and size must be adapted to geometric properties we would like to mask/obtain



Structuring Element

UNIVERSITÀ DI PARMA



Erosion

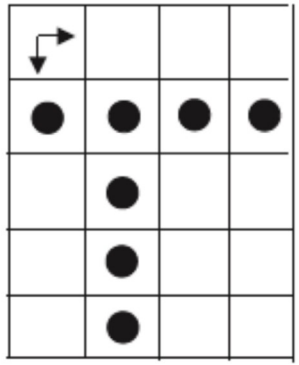
UNIVERSITÀ DI PARMA

- The structuring element B is “moved” (B_z) on the image A for a set of values z belonging to an Euclidean Space E
- The result is the set of z values for which B_z is in A

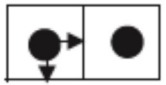
$$A \ominus B = \left\{ z \in E^2 : B_z \subseteq A \right\}$$

- Practically the structuring element is superimposed on all the pixel of the image.
  - When it completely fits the binary image the result is 1, otherwise 0
  - Logical AND
  - It shrinks image elements

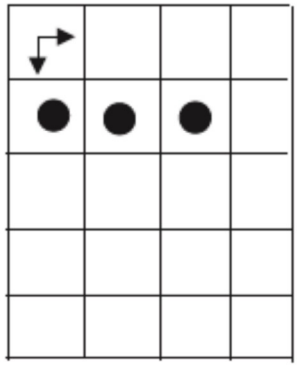
Erosion



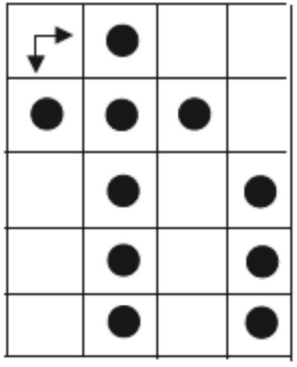
A



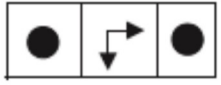
B



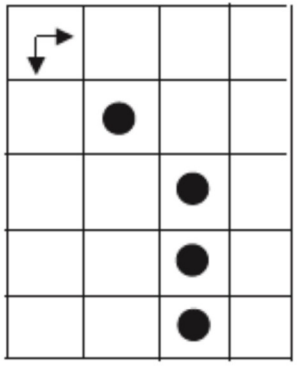
$A \ominus B$



A



B



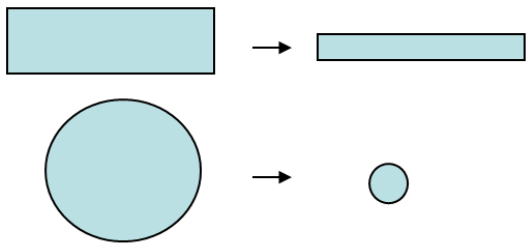
$A \ominus B$

Erosion

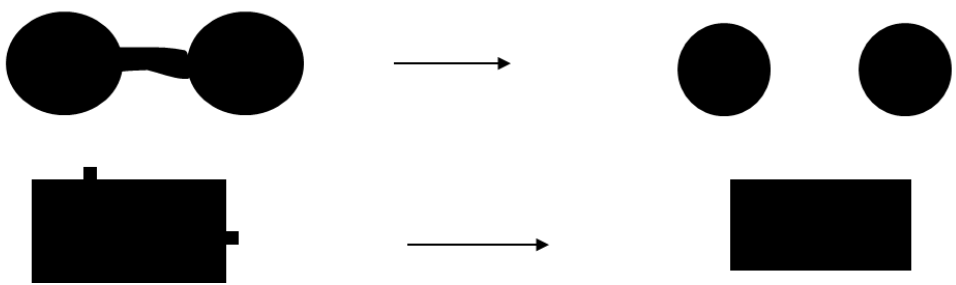
Erosion shrinks the connected sets of 1s of a binary image.

It can be used for

1. shrinking features



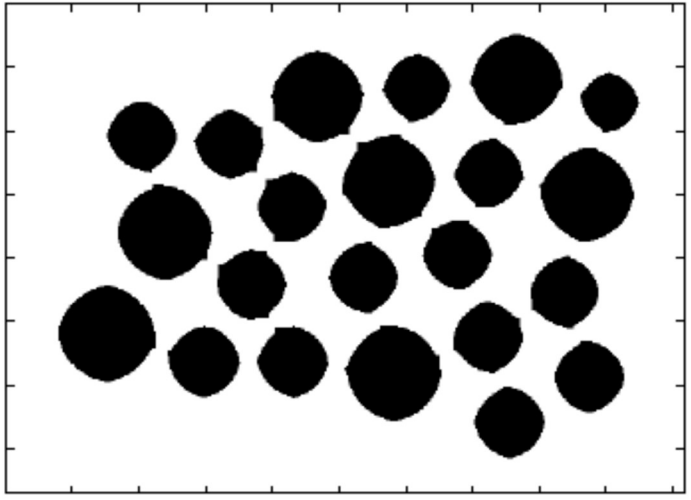
2. Removing bridges, branches and small protrusions



Erosion

UNIVERSITÀ DI PARMA

- Consider the following image

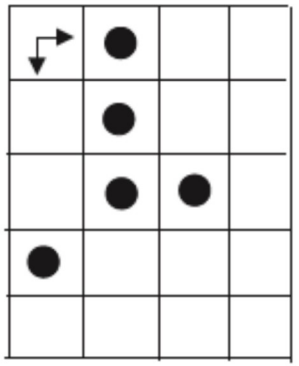


- We can use erosion to separate elements

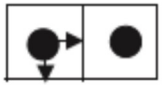
Dilation $\oplus$

- Again the structuring element is superimposed on all the pixel of the image.
  - When it hits the binary image the result is 1, otherwise 0
  - Logical OR
- It “dilates” image elements

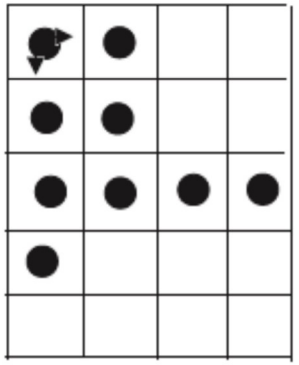
Dilation $\oplus$



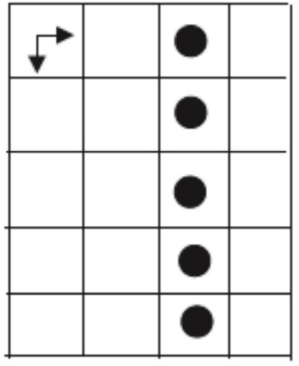
A



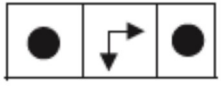
B



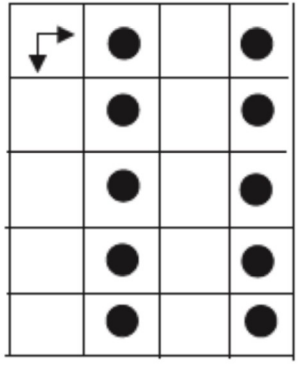
$A \oplus B$



A



B



$A \oplus B$

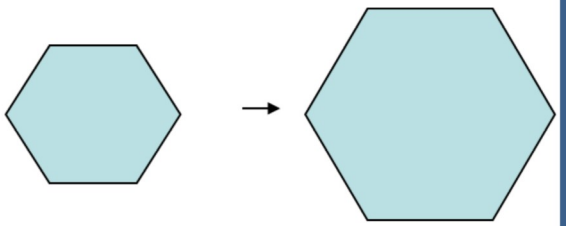
Dilation $\oplus$

UNIVERSITÀ DI PARMA

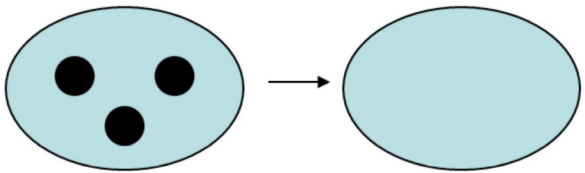
Dilation expands the connected sets of 1s of a binary image.

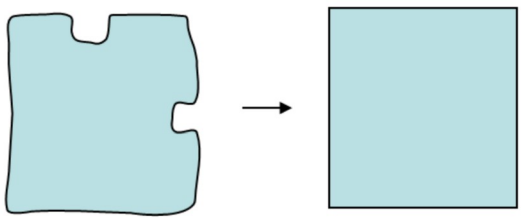
It can be used for

1. growing features



2. filling holes and gaps





Erosion + Dilation

UNIVERSITÀ DI PARMA

- We can use erosion and dilation to remove small details or to fill some gaps
- But they also affect other part of the image…
- Solution
  - Combine them!

Opening/Closing

UNIVERSITÀ DI PARMA

- Opening
  - Erosion followed by dilation

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

- Closing
  - Dilation followed by erosion

$$A \cdot B = (A \oplus B) \ominus B$$

- The same structuring element is used

Examples

(a) (b) (c) (d) (e) (f)

Figure 3.21 Binary image morphology: (a) original image; (b) dilation; (c) erosion; (d) majority; (e) opening; (f) closing. The structuring element for all examples is a 5×5 square. The effects of majority are a subtle rounding of sharp corners. Opening fails to eliminate the dot, since it is not wide enough.

Source: Szeliski

Hit or Miss ⊗

UNIVERSITÀ DI PARMA

- Two “disjoint” structuring elements are used: A e B
  - $A \cap B = \emptyset$
- Two erosion are used and joined
  - $I \otimes X = (I \ominus A) \cap (I^c \ominus B)$
  - Where I^c is the complement for I and X = (A,B)
- The idea is that A should match shapes and B background
  - It can be seen as an “extended” SE

	1	
0	1	1
0	0	

Hit or Miss ⊗

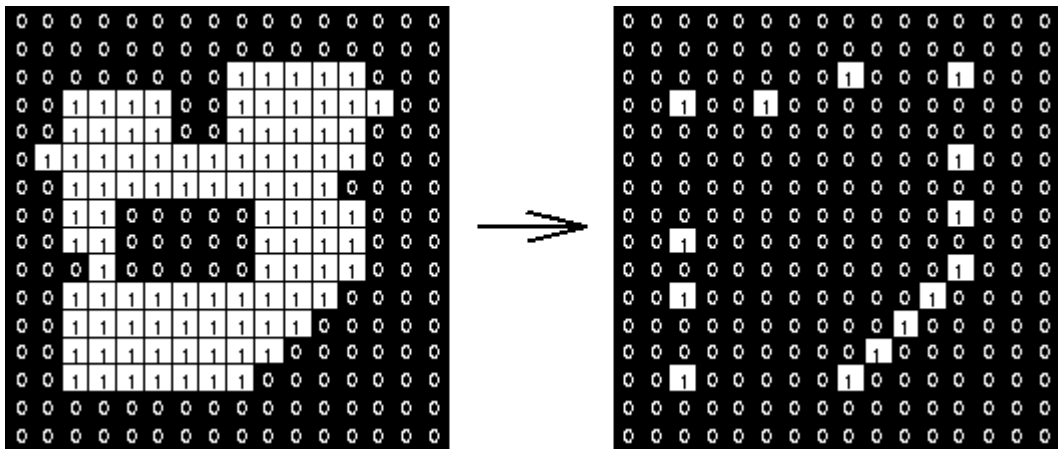
UNIVERSITÀ DI PARMA

	1	
0	1	1
0	0	

	1	
1	1	0
	0	0

	0	0
1	1	0
	1	

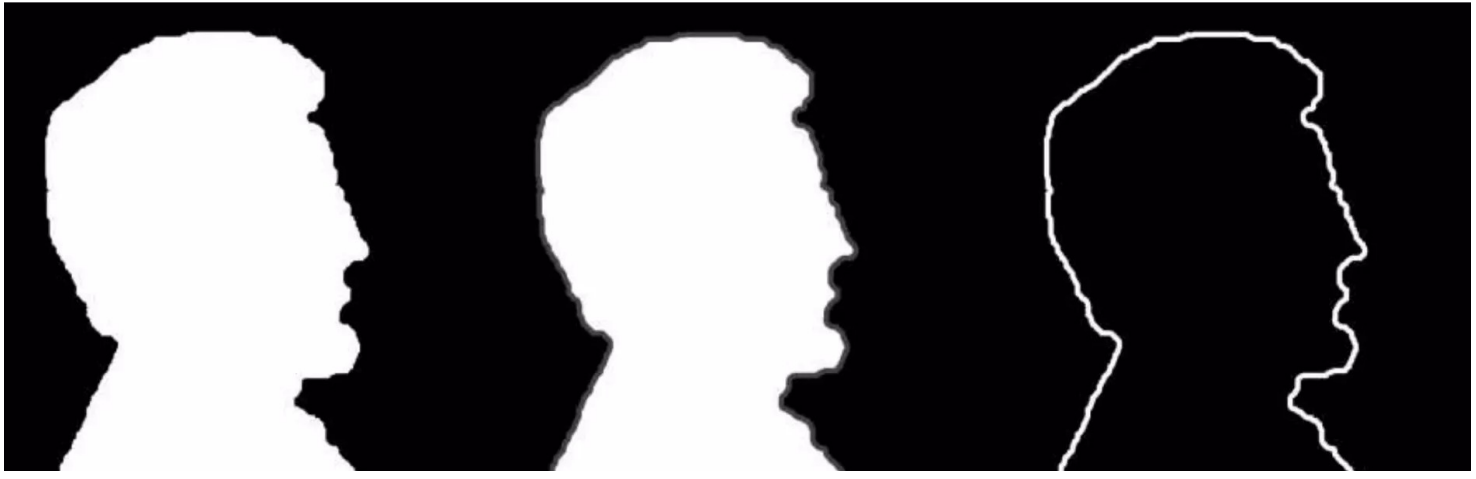
0	0	
0	1	1
	1	



Border Extraction

UNIVERSITÀ DI PARMA

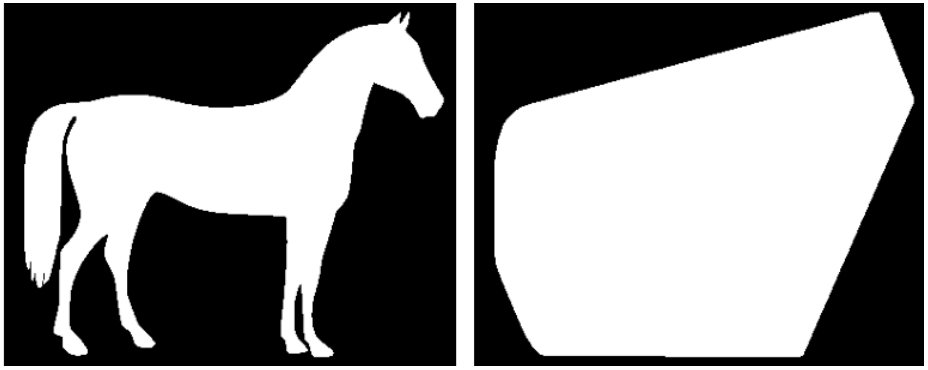
- Erode image using a given SE
- The difference between original image and eroded one is the "border"
- The thickness of the contour depends on the SE size



Convex Hull

UNIVERSITÀ DI PARMA

- A set of points X is defined as **convex set** if any straight line segment between two X points completely lies inside X
- The **convex hull** H of a S set is the smallest convex set that contains S
- **Convex deficiency** is H-S



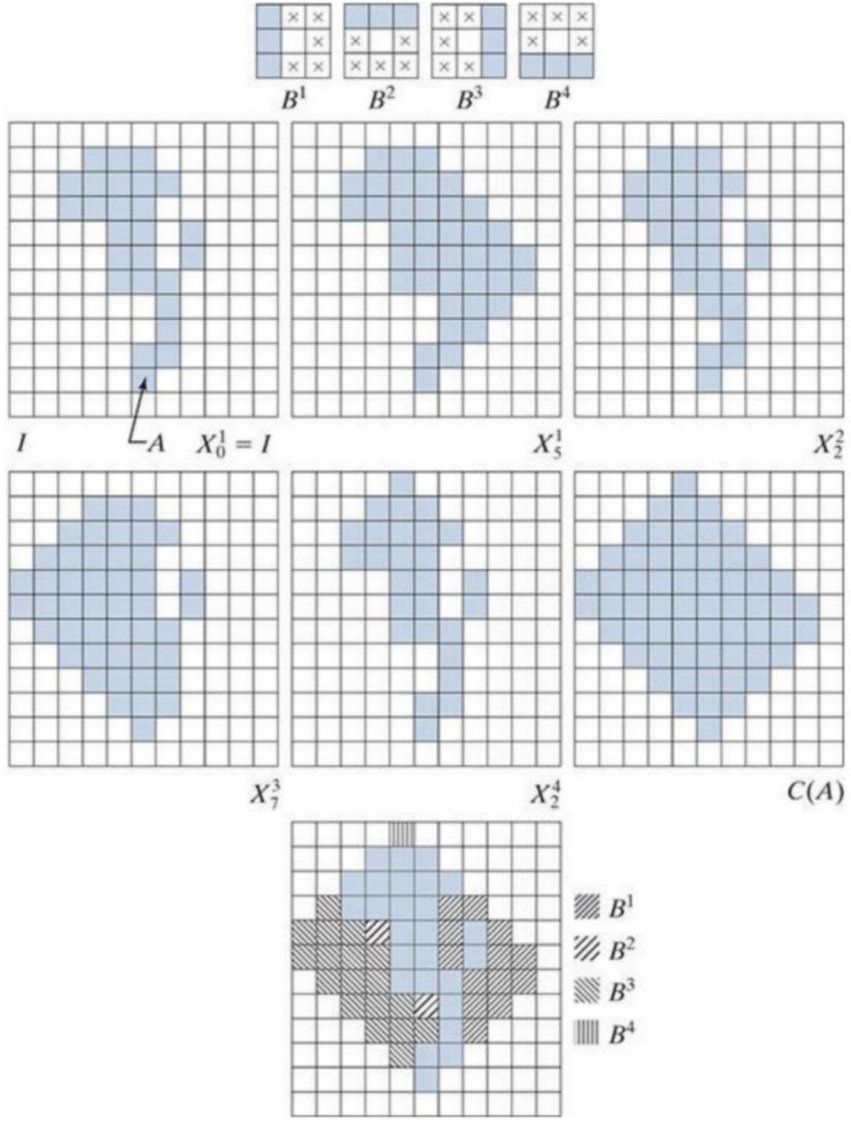
Convex Hull

UNIVERSITÀ DI PARMA

- Convex Hull can be computed using a set of hit and miss transform
  - Apply each SE to the original shape until the result is different
  - The union of the 4 resulting shapes is a convex hull

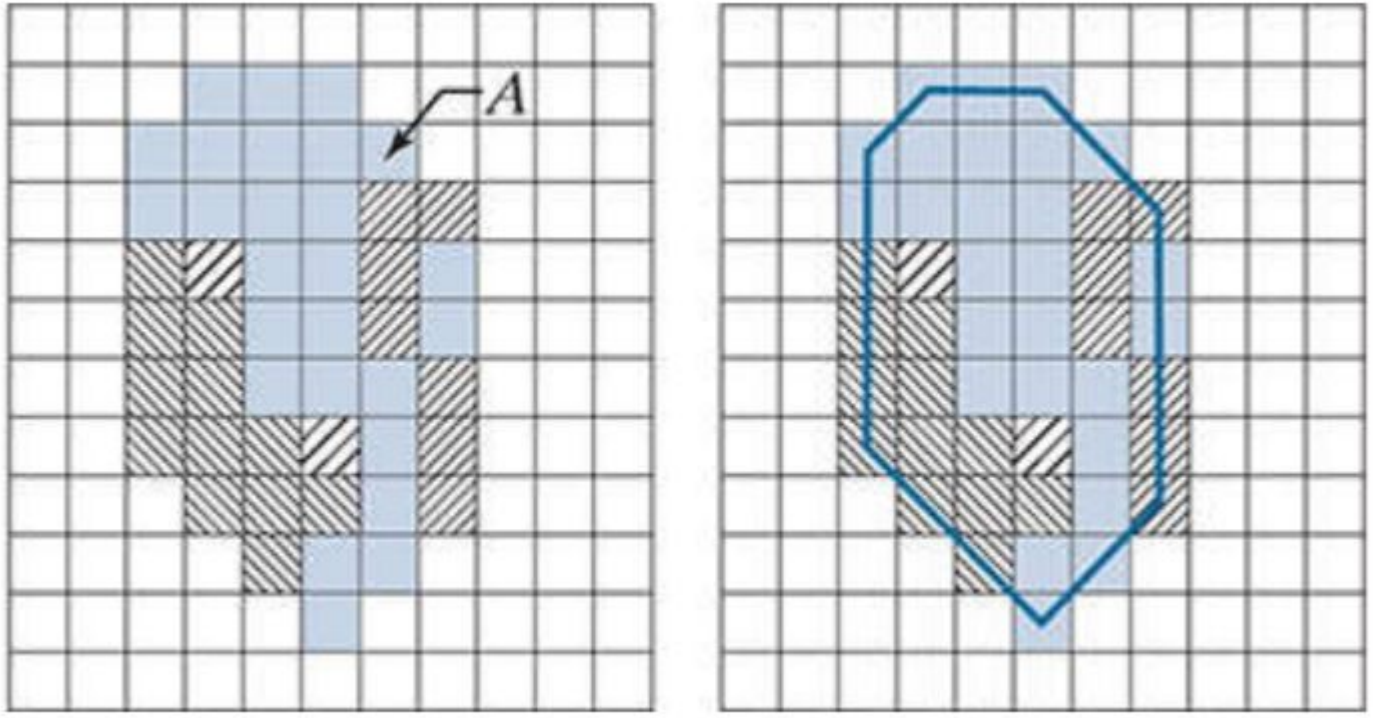
Convex Hull

Not optimized!

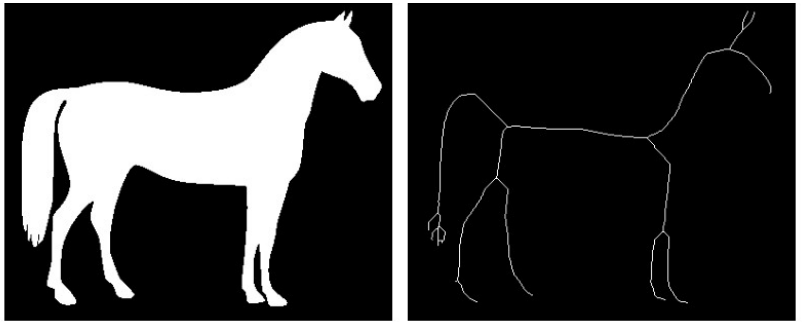


Convex Hull

Edge pixels can be used to “trim” the result
Namely, limiting the convergence in the horizontal or vertical directions



- Skeleton is a thin version of a shape
  - Single pixel thickness
  - Equidistant to shape borders
  - Topological equivalence
- Again a hit and miss approach
  - Up to 8 SEs



UNIVERSITÀ DI PARMA

Morphological Processing for Grey Level Images

Grayscale Morphology

UNIVERSITÀ DI PARMA

- Mathematical morphology can be extended to grayscale images
- Applications
  - Contrast enhancement
  - Texture description
  - Edge detection
  - Thresholding

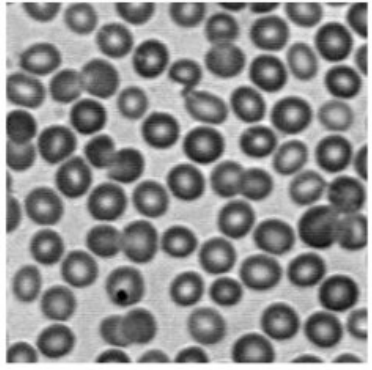
Grayscale Morphology

UNIVERSITÀ DI PARMA

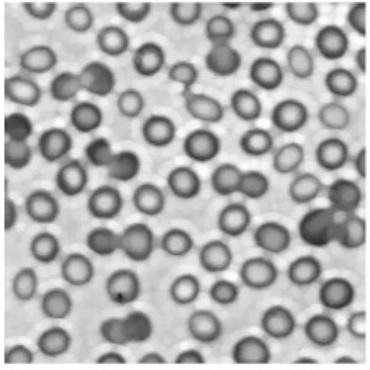
- In a grayscale image there is no “object” and “background”
- Structuring element is still a binary mask
- Min and Max operators are used
  - Dilation ⟶ max element under the mask wins
  - Erosion ⟶ min element under the mask wins
- Opening & Closing as before

Grayscale Morphology

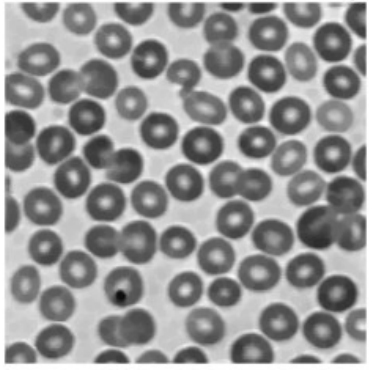
original



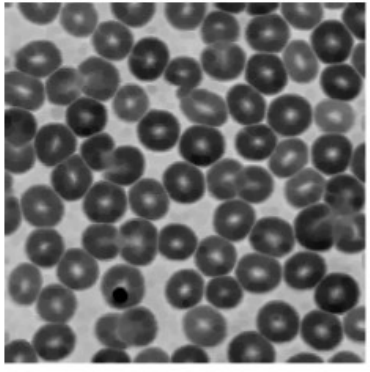
dilation



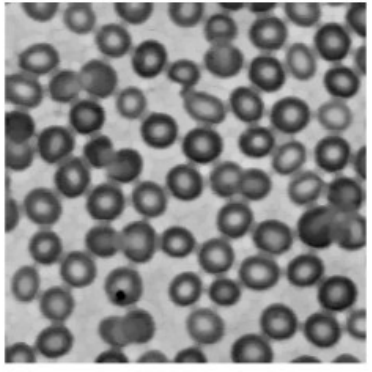
closing



erosion



opening



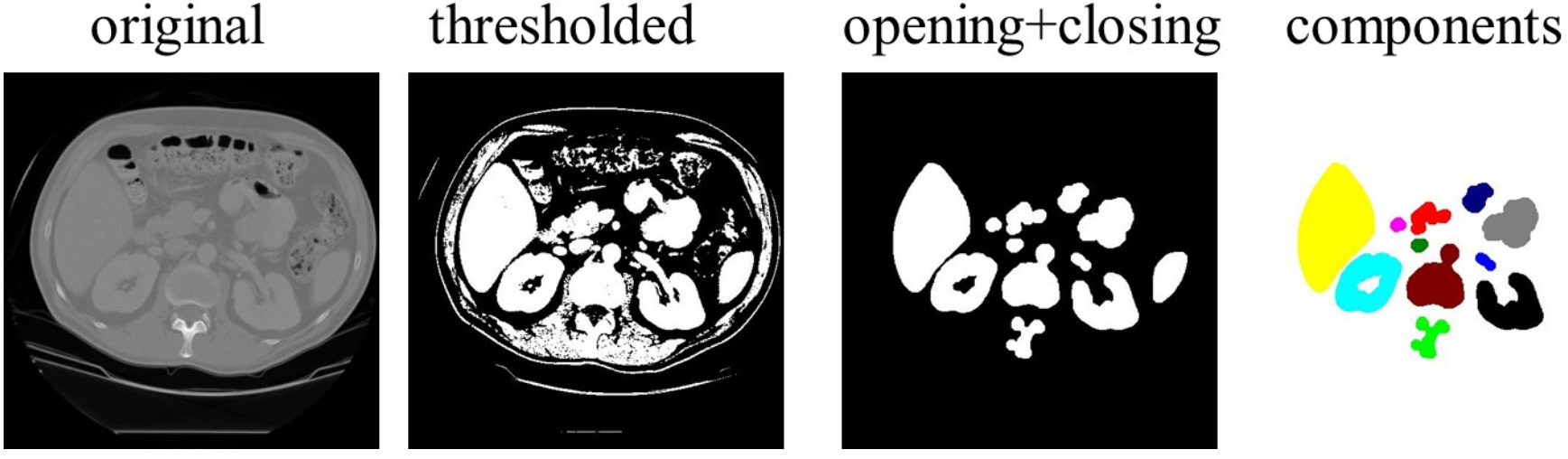
UNIVERSITÀ DI PARMA

Connected Components

Connected Components

UNIVERSITÀ DI PARMA

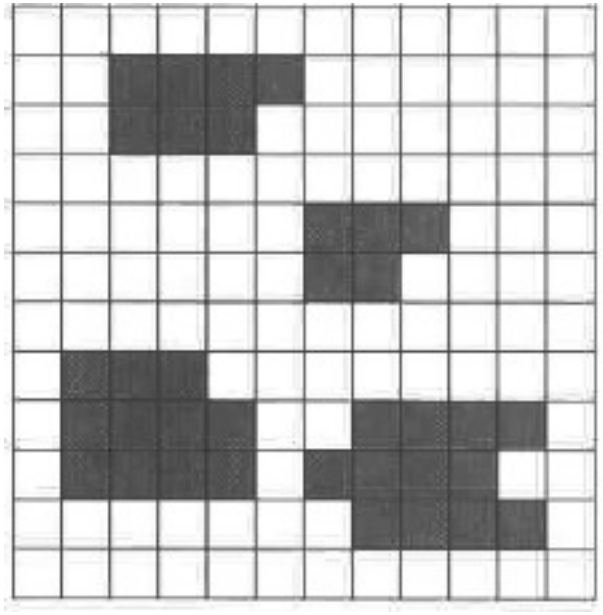
- A connected component is a **contiguous** group of pixels that have the **same value**
  - Actually not only for binary images

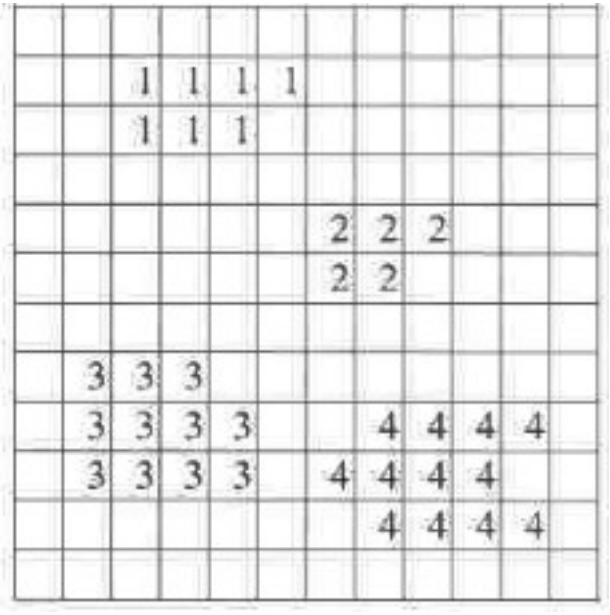


Labelling algorithms for CC

UNIVERSITÀ DI PARMA

- We assign a unique label to each component



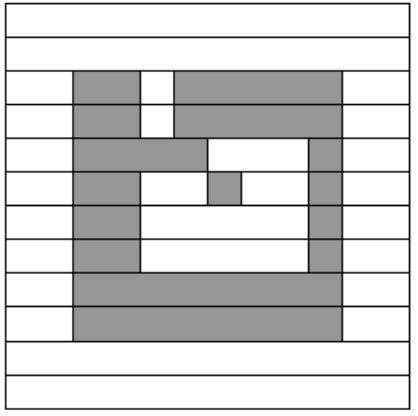


Source: R. Jain

Labbelling algorithms for CC

UNIVERSITÀ DI PARMA

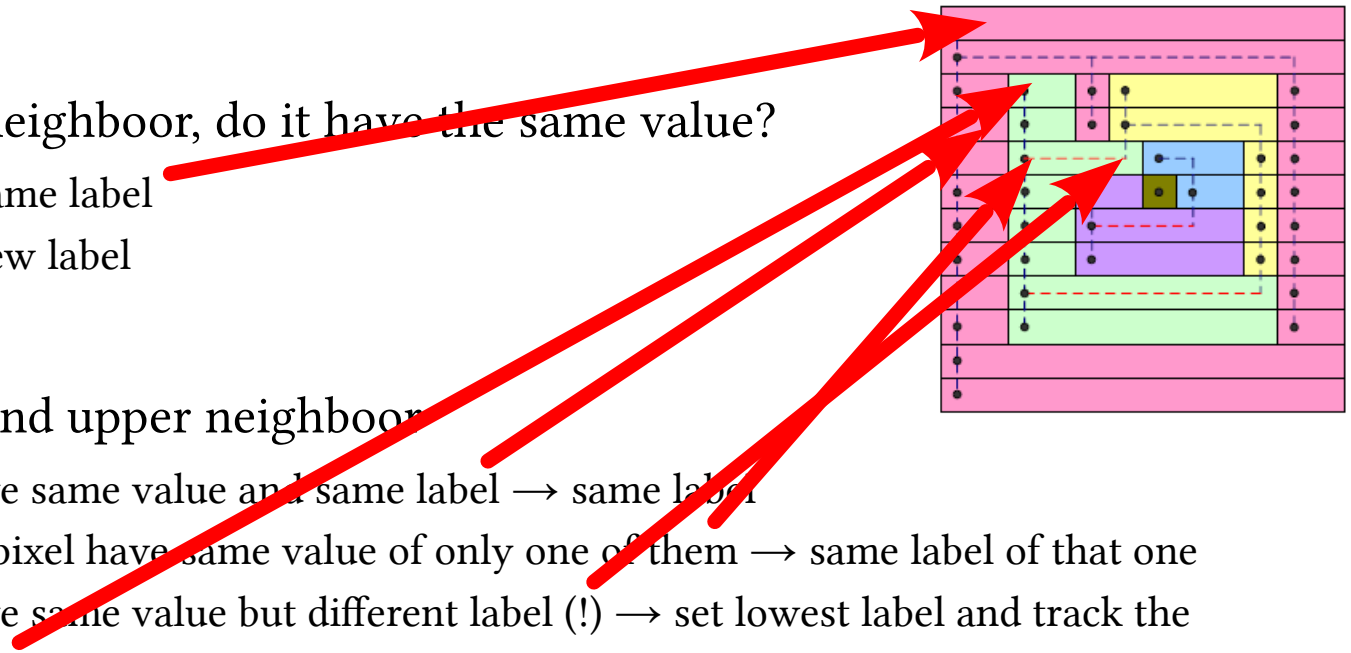
- Different approaches can be used:
  - Recursive Tracking
  - Parallel Growing
  - Row-by-Row (widely used)



Row by Row Labelling algorithm

UNIVERSITÀ DI PARMA

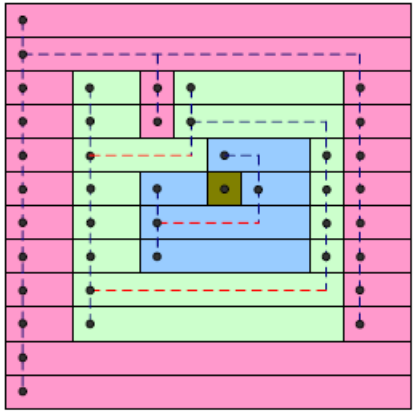
- 1st pass:
  - First row:
    - Check left neighboor, do it have the same value?
      - Yes → same label
      - No → new label
  - Other rows
    - Check left and upper neighboor
      - They have same value and same label → same label
      - Current pixel have same value of only one of them → same label of that one
      - They have same value but different label (!) → set lowest label and track the equivalence
      - Otherwise → new label



Row by Row Labelling algorithm

UNIVERSITÀ DI PARMA

- 2nd pass:
  - During 1st pass we have tracked label equivalences
  - Row by row change labels with lowest equivalent

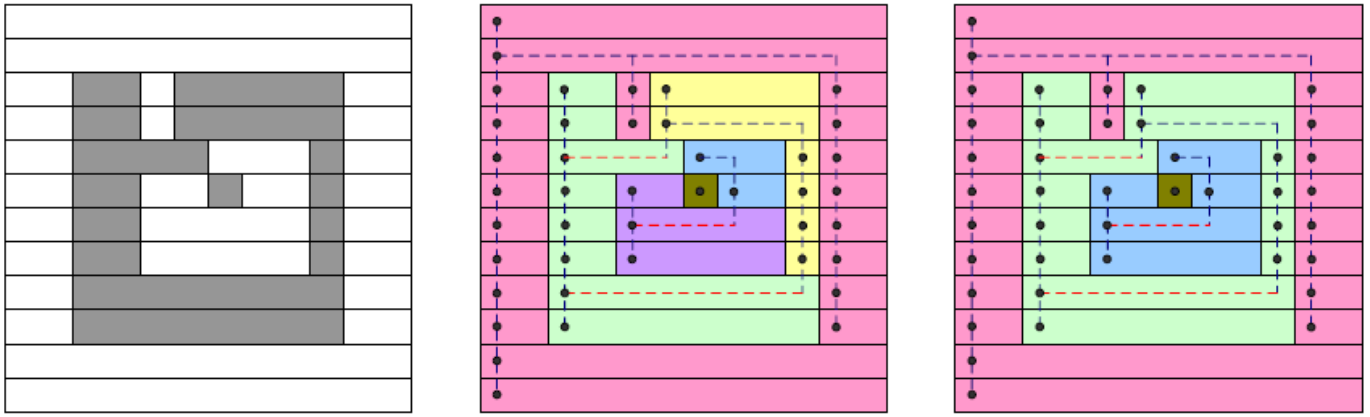


Labelling algorithms for CC

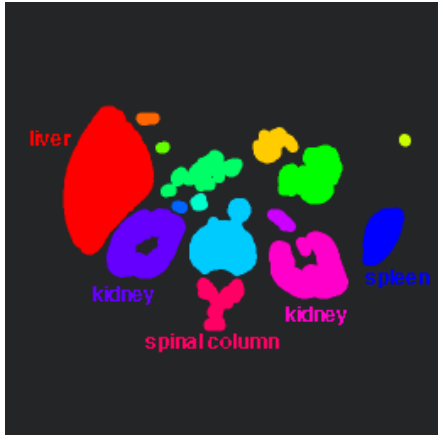
UNIVERSITÀ DI PARMA

- Row by Row

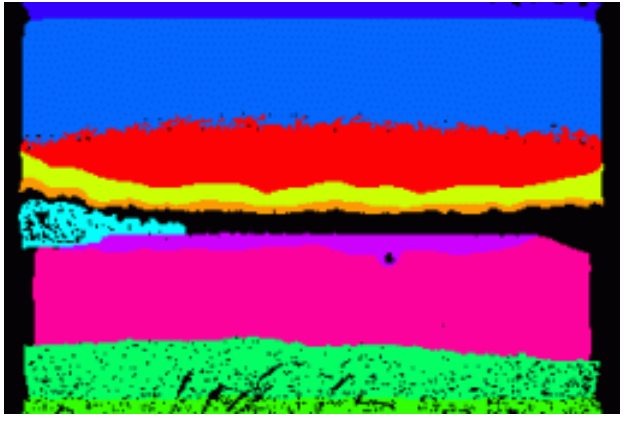
Source: Szeliski



Connected Components



connected components of 1's from cleaned, thresholded image



connected components of cluster labels