

Thị giác máy tính

L2 (Continue): Binary images



SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

1

Binary images



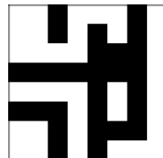
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

2

Binary images

- Two pixel values: foreground (object, 1) and background (0)
- Be used
 - To mark region(s) of interest
 - As results of thresholding method

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



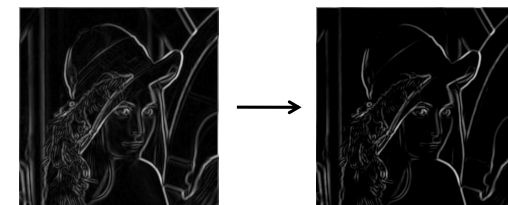
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

3

Thresholding

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

Example: edge detection



Gradient magnitude

`fg_pix = find(gradient_mag > t);`

Looking for pixels where gradient is strong.



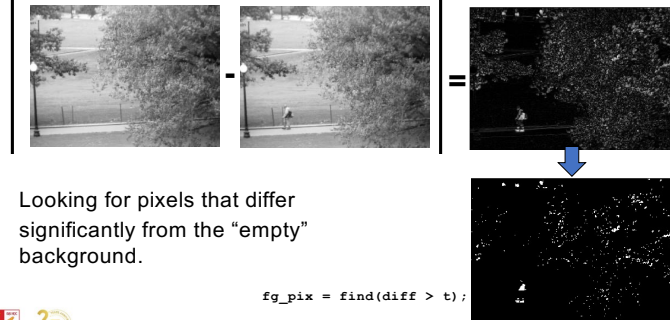
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

4

Thresholding

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

Example: background subtraction



SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

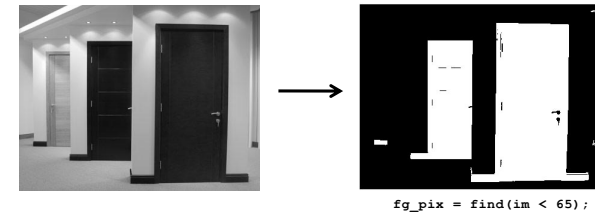
5

5

Thresholding

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

Example: intensity-based detection



Looking for dark pixels



SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

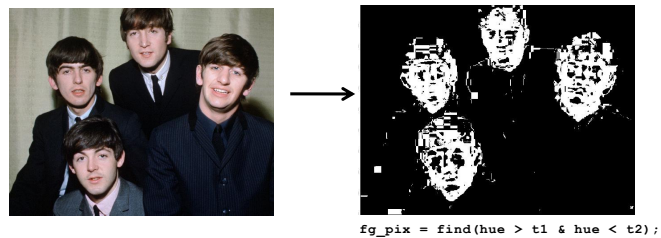
6

6

Thresholding

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

Example: color-based detection



Looking for pixels within a certain hue range.



SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

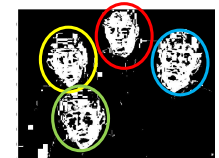
Slide credit: Kristen Grauman

7

7

Issues

- What to do with “noisy” binary outputs?
 - Holes
 - Extra small fragments
- How to demarcate multiple regions of interest?
 - Count objects
 - Compute further features per object



SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

Slide credit: Kristen Grauman

8

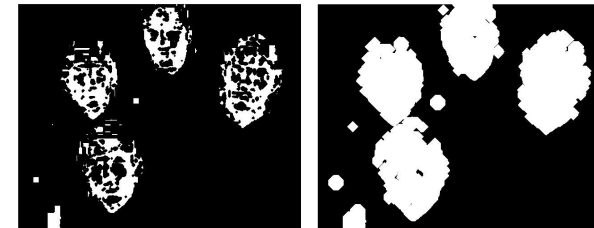
8

Morphological operators

- Change the shape of the foreground regions via intersection/union operations between a scanning structuring element and binary image.
- Useful to clean up result from thresholding
- Main components
 - Structuring element
 - Operators:
 - Basic operators: Dilation, Erosion
 - Others: Opening, Closing, ...

Dilation

- Expands connected components
- Grow features
- Fill holes

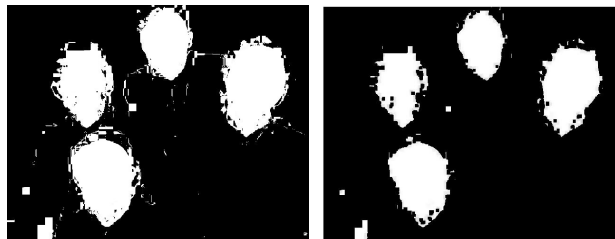


Before dilation

After dilation

Erosion

- Erode connected components
- Shrink features
- Remove bridges, branches, noise

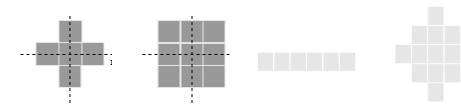


Before erosion

After erosion

Structuring elements

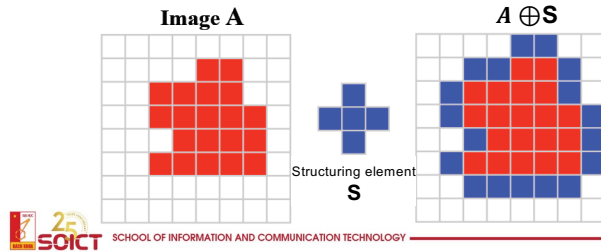
- **Masks** of varying shapes and sizes used to perform morphology, for example:



- Scan mask (structuring element) over the **object (foreground) borders (inside and outside)** and transform the binary image

Dilation

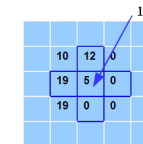
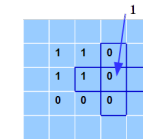
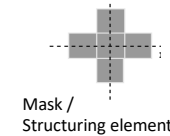
- Moving S on each pixel of A
 - check if the intersection (pixels belonging to object) is not empty
 - If yes, the center of B belongs to the result image
- If a pixel of S is onto object pixels (A), then the central pixel belongs to object
 - Otherwise (i.e. all pixels of are background), set to background (no change)



13

Dilation

- As max filter
- Can be applied both on
 - binary images
 - or grayscale images

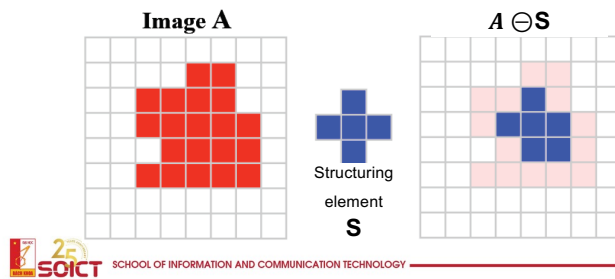


14

14

Erosion

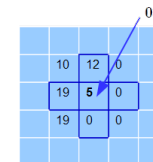
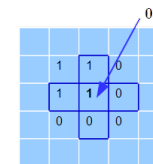
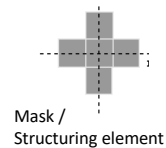
- We put the element S on each pixel x of A
 - like convolution
- If all pixels of S are onto object pixels (A), then the central pixel belongs to object
 - Otherwise (i.e. a mask pixel is background), set to background



15

Erosion

- As min filter
- Can be applied both on
 - binary images
 - or grayscale images

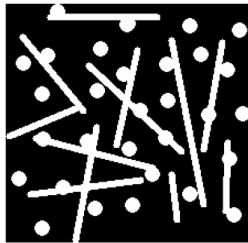


16

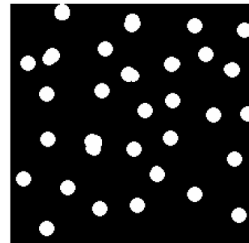
16

Opening

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



Before opening



After opening



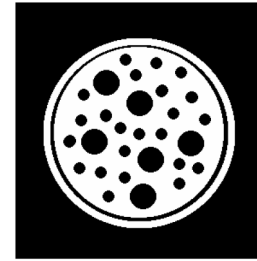
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

17

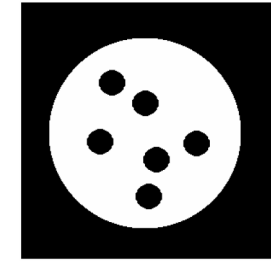
17

Closing

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



Before closing



After closing

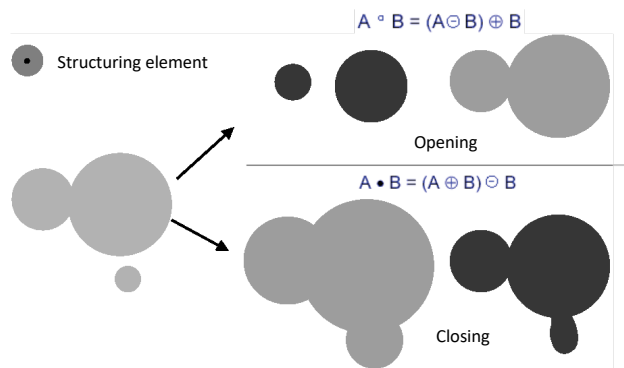
demo: <http://bigwww.epfl.ch/demo/jmorpho/start.php>

SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

18

18

Opening vs Closing



SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

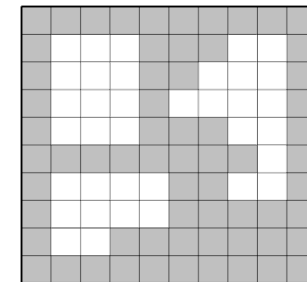
19

19

Connected component labeling

- We loop over all the image to give a **unique number (label)** for each region
- All pixels from the **same region** must have the **same number (label)**
- Objectifs:
 - Counting objects
 - Separating objects
 - Creating a mask for each object
 - ...

← Background
← Segmented objects



SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

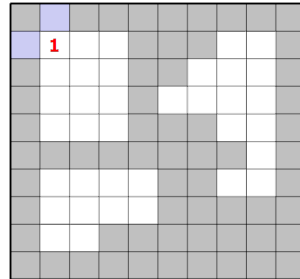
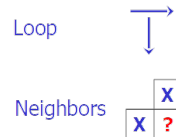
20

20

Connected component labeling

First loop over the image

- For each pixel in a region, we set
 - or the smallest label from its **top** or **left** neighbors
 - or a new label



SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

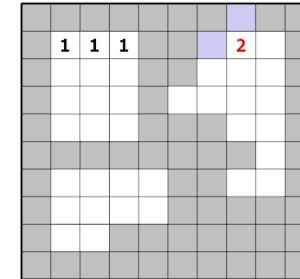
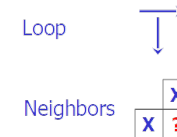
21

21

Connected component labeling

First loop over the image

- For each pixel in a region, we set
 - or the smallest label from its **top** or **left** neighbors
 - or a new label



SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

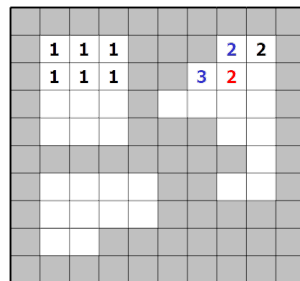
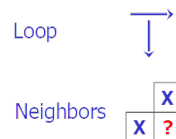
22

22

Connected component labeling

First loop over the image

- For each pixel in a region, we set
 - or the smallest label from its **top** or **left** neighbors
 - or a new label



SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

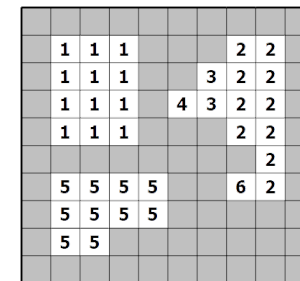
23

23

Connected component labeling

First loop over the image

- For each pixel in a region, we set
 - or the smallest label from its **top** or **left** neighbors
 - or a new label



SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

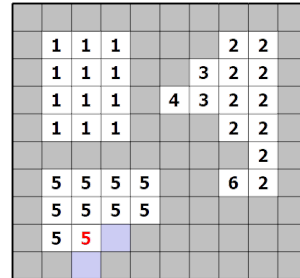
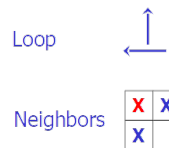
24

24

Connected component labeling

Second loop over the image

- For each pixel in a region, we set
 - the smallest from its **own label** and the labels from its **down** and **right** neighbors



SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

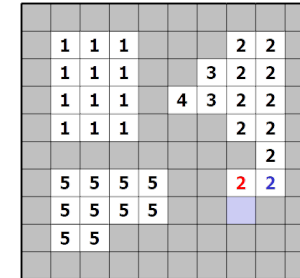
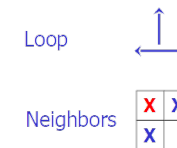
25

25

Connected component labeling

Second loop over the image

- For each pixel in a region, we set
 - the smallest from its **own label** and the labels from its **down** and **right** neighbors



SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

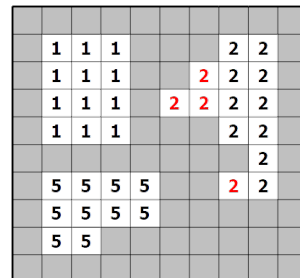
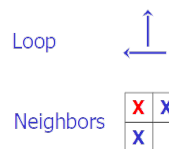
26

26

Connected component labeling

Second loop over the image

- For each pixel in a region, we set
 - the smallest from its **own label** and the labels from its **down** and **right** neighbors



SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

27

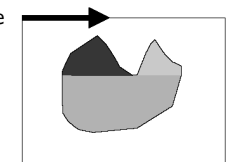
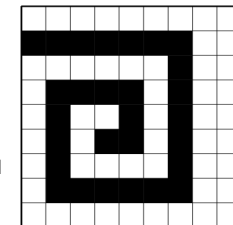
27

Connected component labeling

- Two loops are enough?
 - example: *spiral region* !

Solutions

- We continue, **go and back two ways**, until **no new change** in labels
- It is possible to do only one loop: manage a table of equivalences when 2 different labels are neighbors



SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

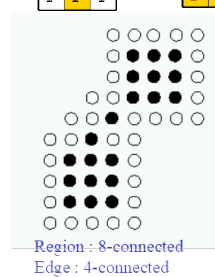
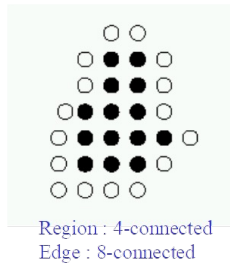
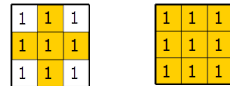
28

28

CC labeling: how many neighbors?

- **Advice:** Use different connexities for edges and regions

- 4-Connexity for regions
- 8-Connexity for edges



SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

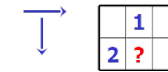
29

29

CC labeling: how many neighbors?

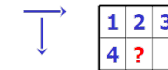
- Regions labeling

- We use 4-connexity
- Each loop, we compare 2 neighbors



- Edge labeling

- 8-connexity
- Each loop, we compare 4 neighbors



SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

30

30



VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

Thank you for
your attention!

soict.hust.edu.vn/
fb.com/groups/soict



31