COMP 448/548: Medical Image Analysis

Medical image segmentation: Simple techniques

Çiğdem Gündüz Demir
cgunduz@ku.edu.tr

1

Last lecture: Image filtering

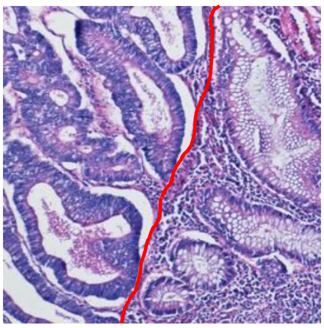
- Filtering forms a new image, each pixel of which is a function of the pixels in that pixel's local neighborhood
- Smoothing/denoising
  - Linear filters: Average, Gaussian
    - ➢ They reduce noise but at the same time blur edges
  - Nonlinear filters: Median
- Edge detection
  - Prewitt, Sobel, Laplacian
    - ➢ They give similar responses for edges and noise
  - Laplacian of Gaussian (LoG)
  - Canny edge detector
- Texture feature extraction
- Template matching

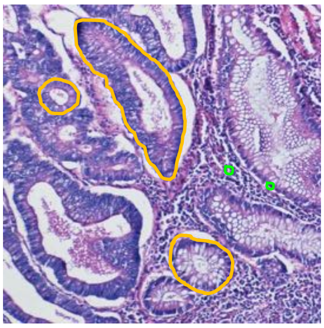
For thorough reading on computer vision, download the following book from its author's website
R. Szeliski, *Computer Vision: Algorithms and Applications*, 2nd ed.
http://szeliski.org/Book/

For detailed reading on cluster analysis, download the following book chapter from its author's website
P-N. Tan, M. Steinbach, A. Karpatne, V. Kumar, *Introduction to Data Mining*, 2nd ed.
https://www-users.cs.umn.edu/~kumar001/dmbook/ch8.pdf

2

Next: Image segmentation

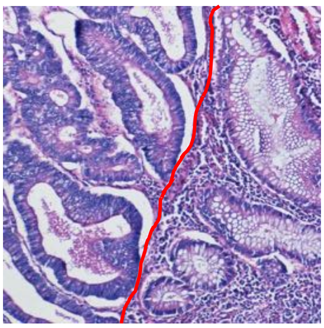




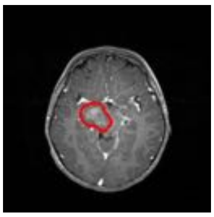
- Aims to partition an image into *meaningful* parts with respect to a particular application
  - These parts are groups of pixels that go together
  - This partitioning can be unsupervised
  - Semantic segmentation assigns labels to the partitions
- Usually the initial step to understand an image
  - Thus, its success greatly affects the performance of the remaining steps and the overall system
- However, image segmentation is not always (indeed mostly) that trivial

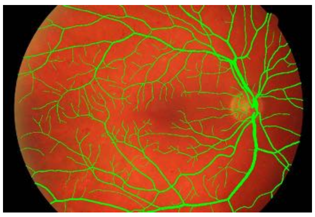
3

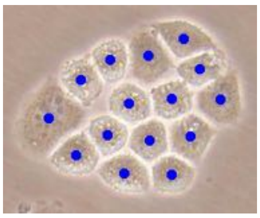
Medical image segmentation

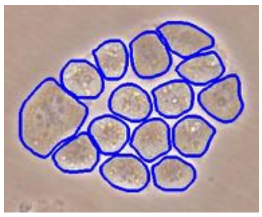


- Typically deals with
  - Locating biological structures (cells, glands, vessels, tumor, lesions, etc.) on an image
    - ➢ Similar to object detection in other domains
    - ➢ Could be in the form of detecting approximate object locations (e.g., centroids, bounding boxes) or finding their exact boundaries
  - Dividing a heterogeneous image into its homogenous regions
    - ➢ Similar to scene segmentation in other domains



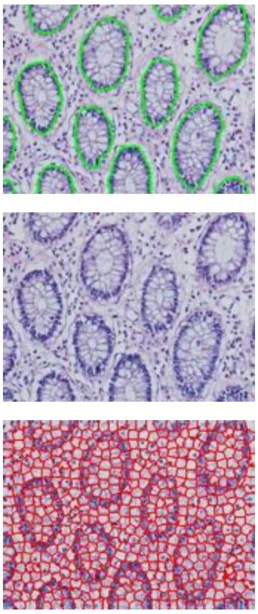






4

Segmentation algorithms



- Partition an image based on similarity or discontinuity
  - *Similarity:* Find regions of pixels that are similar to each other according to a metric
  - *Discontinuity:* Find abrupt changes in the characteristics of pixels (e.g., edges defined as the rapid changes in pixel intensities)
- Similarity (or discontinuity) is usually defined on the gray-level intensity, color intensity or texture of pixels/voxels
- Segmentation algorithms usually run on pixels/voxels
  - However, it is also possible to use other types of primitives
  - For example, super-pixels found by the SLIC algorithm
  - In this case, structural characteristics can also be used

5

Medical image segmentation

- In this course, you will learn different techniques to construct a segmentation algorithm for medical images
- First, you will learn the traditional way
  - Basic concepts and simple techniques
    - Connected components, thresholding, clustering, mathematical morphology, …

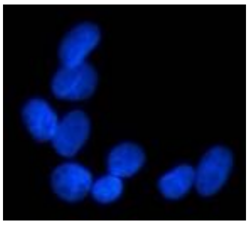
    TODAY
  - More advanced techniques
    - Distance transforms, texture analysis, region growing, graph-based segmentation, active contour models, …

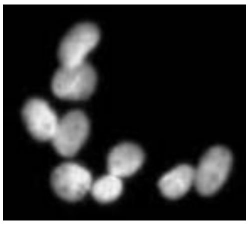
    NEXT LECTURE
- Then, you will learn how to use deep neural networks for segmentation

  A COUPLE OF WEEKS LATER

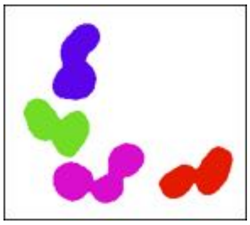
6

Simple example: Cell segmentation


RGB image


Blue channel

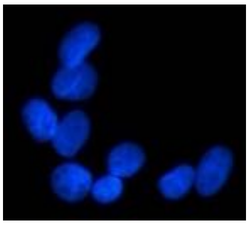

Binarized with the Otsu's threshold

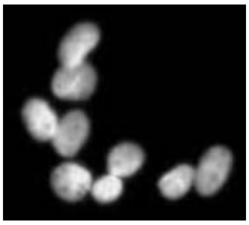

Connected components of the binarized image

UNDERSEGMENTATION

7

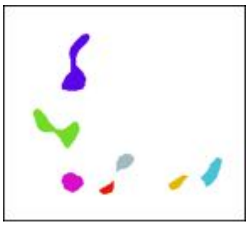
Simple example: Cell segmentation

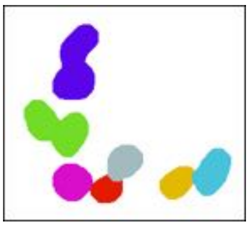

RGB image


Blue channel


Binarized with the Otsu's threshold


Morphological erosion on the binarized image

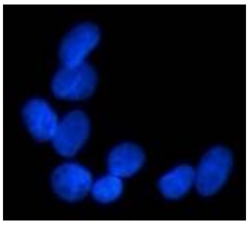

Connected components of the eroded image

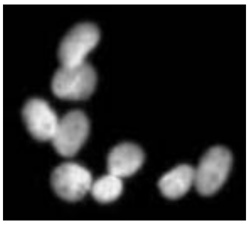

Morphological dilation on the connected components

BETTER ☺
BUT STILL UNDERSEGMENTATION

8

Simple example: Cell segmentation


RGB image

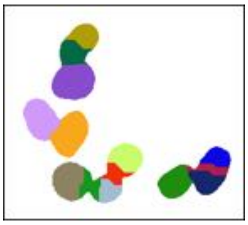

Blue channel


Binarized with the Otsu's threshold


Outer distance transform on the binarized image


Regional maxima of the outer distance transform

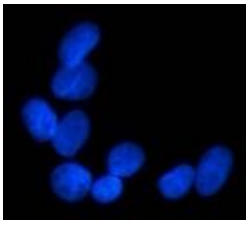

Connected components of the regional maxima

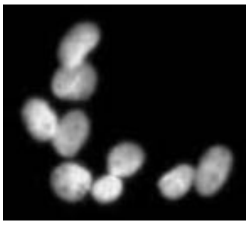

Connected components after marker-controlled region growing

NO UNDERSEGMENTATION BUT THIS TIME OVERSEGMENTATION

9

Simple example: Cell segmentation

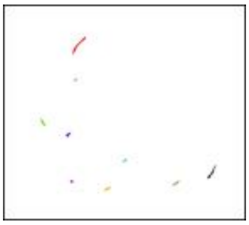

RGB image


Blue channel


Binarized with the Otsu's threshold


Outer distance transform on the binarized image


Regional maxima after suppressing maxima whose height is less than 1


Connected components of the regional maxima

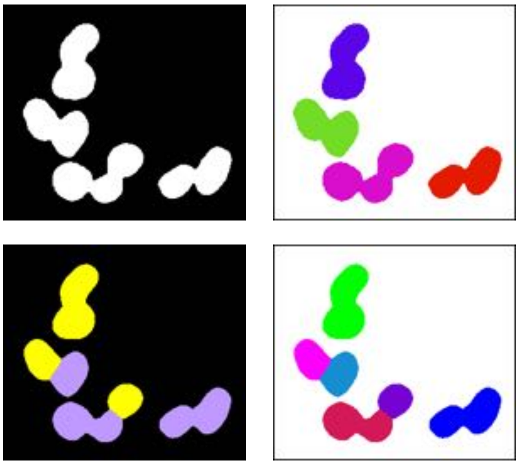

Connected components after marker-controlled region growing


Connected components after majority filtering

10

Connected components

- A connected component is a set of pixels of the same label that are not separated by boundary pixels



11

Connected components

- Components can be
  - 4-connected or 8-connected in 2D images
  - 6-connected or 18-connected or 26-connected in 3D volumes

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

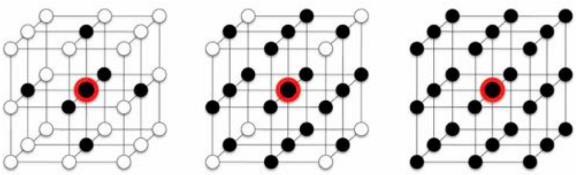
→

1	1	0	0
1	1	0	0
0	0	2	2
0	0	2	2

4-connectivity

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

8-connectivity



6-connectivity *18-connectivity* *26-connectivity*

12

Common operations on connected components

- Small component elimination
  - Eliminate components that have fewer than *areaThr* pixels
- Hole filling
  - Fill holes (isolated background pixels) inside the components
  - A hole is defined with respect to the selected connectivity
- Mathematical morphology
  - Erosion, dilation, opening, closing
  - Also others (tophat, hit-or-miss, thinning, thickening, skeletonizing, etc.)
- Majority filtering

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

This is a hole with respect to 4-connectivity but not a hole with respect to 8-connectivity

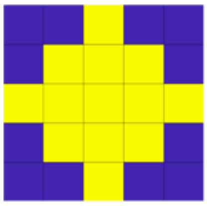
13

Mathematical morphology

- Defined on binary images (or their connected components) but can also be generalized for grayscale images
- Binary morphology is commonly used to post-process segmented images
- Modifies the shapes of objects/regions in an image
  - Using the pixels in a local neighborhood determined by a *structuring element*
    - ➢ Similar to the filtering operation
  - With the hope of correcting their shapes
    - ➢ For example, by filling holes, linking close but unconnected pixels, breaking narrow bridges, and thinning objects

14

Binary erosion



- For each foreground pixel p
  - Locate a structuring element S, considering p as the center point
  - If there are any background pixels in S, make p background
  - Otherwise keep it as foreground

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

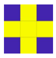
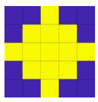
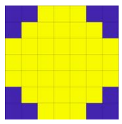
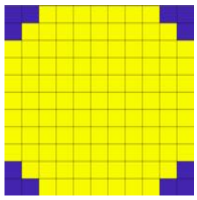
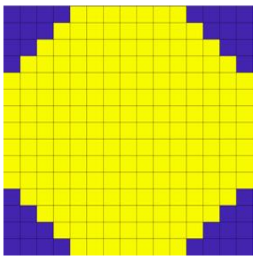
Keep it as foreground

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

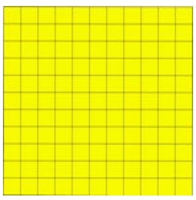
Make it background

15

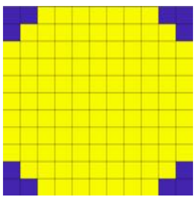
Structuring element

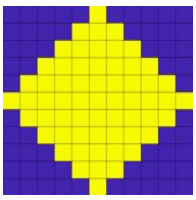
- Kernel containing ones and zeros
- Can have a structured shape (such as square, disk, line, and diamond) or an arbitrary shape
- Its size is typically odd



Square structuring element with edge length = 11



Disk structuring element with diameter = 11



Diamond structuring element with size = 11

16

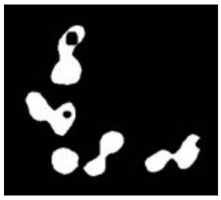
Binary erosion

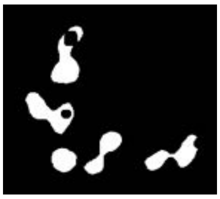
- Shrinks foreground objects
- Enlarges holes inside foreground objects
- Breaks narrow foreground bridges
  - Can be used to split undersegmented objects
- The shape and size of its structuring element affects the results (which pixels are eroded)

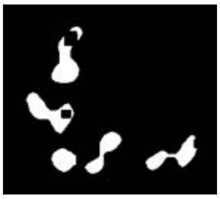

Binary image before erosion

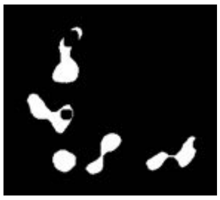

Erosion with disk, d = 3

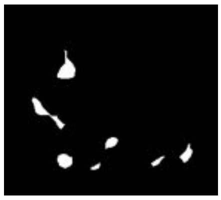

Erosion with disk, d = 5


Erosion with disk, d = 7


Erosion with disk, d = 9


Erosion with square, e = 9

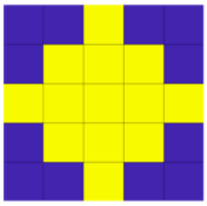

Erosion with disk, d = 11


Erosion with disk, d = 21

17

Binary dilation

- For each background pixel p
  - Locate a structuring element S, considering p as the center point
  - If there are any foreground pixels in S, make p foreground
  - Otherwise keep it as background



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

Make it foreground

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

Keep it as background

18

Binary dilation

- Enlarges foreground objects
- Shrinks holes inside foreground objects (sometimes fills these holes completely)
- Links close but unconnected pixels
- The shape and size of its structuring element affects the results (which pixels are dilated)


Binary image before dilation


Dilation with disk, d = 3


Dilation with disk, d = 5


Dilation with disk, d = 7


Dilation with disk, d = 11


Dilation with disk, d = 21

19

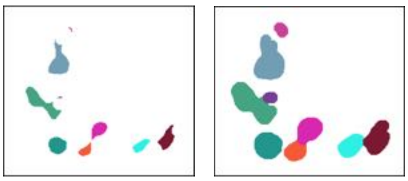
Binary opening

- Erosion removes small regions and breaks (opens) narrow bridges, but shapes of larger regions also get affected
- Opening first erodes an image to remove small regions and narrow bridges and then dilates the eroded regions with the same structuring element to restore the original shape
  - Too small regions will not survive during erosion, and thus, they are not restored back


Binary image before opening


Erosion with disk, d = 15


Dilation with disk, d = 15


Sometimes, finding connected components on the eroded image and then dilating each of them separately is more effective to split undersegmented objects

20

Binary closing

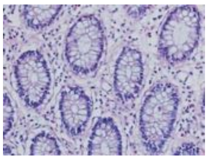
- Dilation fills holes and links close but unconnected pixels (close small gaps), but shapes of all foreground regions also get affected
- Closing first dilates an image to fill the holes and close the small gaps and then erodes the dilated regions with the same structuring element to restore the original shape
  - Completely filled holes and closed gaps will not be opened again

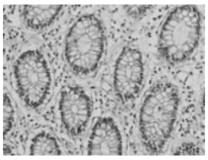

Binary image before closing

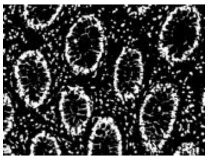

Dilation with disk, d = 15


Erosion with disk, d = 15

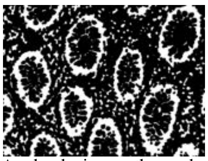
Simple algorithm for gland segmentation

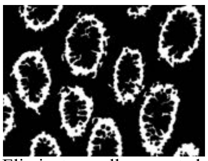


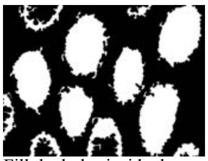

Convert to its grayscale

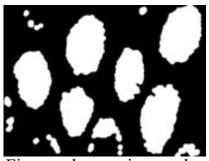

Find nucleus pixels by thresholding the grayscale

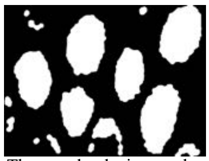
Finding connected components of the nucleus pixels and then filling the holes inside them will not help for successful segmentation


Apply closing on the nucleus pixels to close gaps


Eliminate small connected components


Fill the holes inside the remaining large components


First apply opening on the filled components to smooth their boundaries

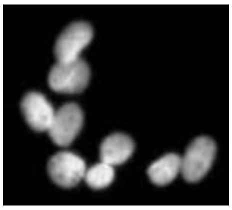

Then apply closing on the opened components for further smoothing

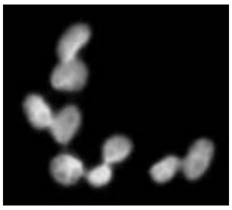

Eliminate small regions


Find connected components

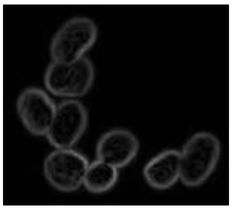
Grayscale erosion and dilation

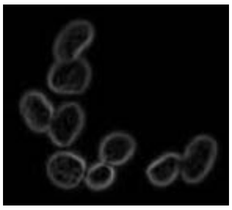
- Erosion assigns to each pixel the minimum value in the local neighborhood defined by a structuring element
- Dilation assigns to each pixel the maximum value in the local neighborhood defined by a structuring element
- They are non-linear filters


Blue channel


Eroded with disk, d = 7


Dilated with disk, d = 7


Dilated – original


Original – eroded

23

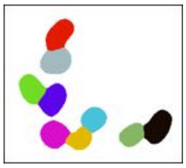
Majority filtering

- For each pixel p
  - Locate a structuring element S, considering p as the center point
  - Use the majority of pixel labels inside S as the label of p

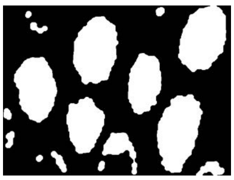
0	0	0	1	1	0	2	0	0	0
0	0	1	1	1	0	2	0	0	0
0	0	1	1	1	0	2	2	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	3	3	3	3	3	3	0
0	0	0	3	3	3	4	4	0	0
0	0	0	0	3	3	3	4	0	0
0	0	0	0	0	0	4	4	0	0
0	0	0	0	4	4	4	4	0	0
0	0	0	4	4	4	4	0	0	0

Its label will be 3 (four 0s, five 3s, four 4s)


Before


After


Binary image

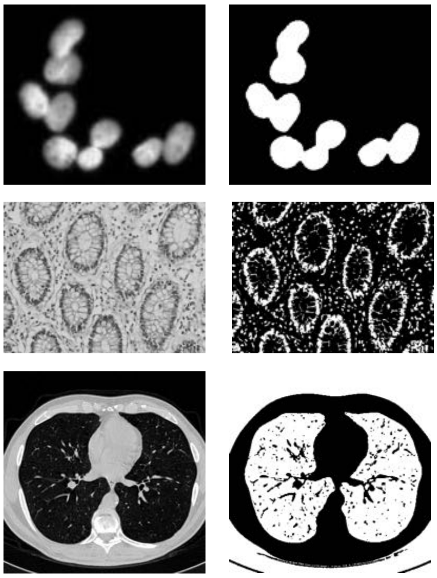

Opening followed by closing


Majority filtering

24

Thresholding

- Separates pixels (or superpixels/regions) into two groups based on their selected feature F and a given threshold T
  - Assign a pixel p to Group 1 if F(p) >= T
  - Assign it to Group 2 if f(p) < T
- Intensity, color, texture, filter response of a pixel/voxel are commonly used features



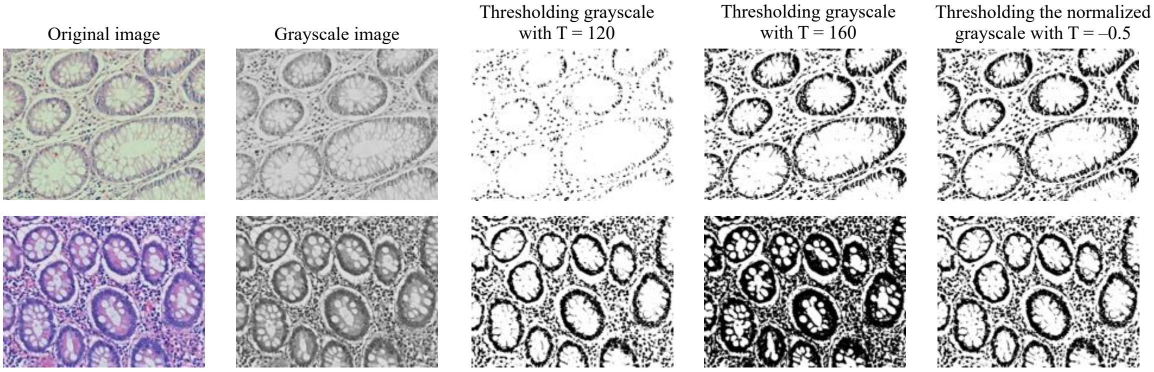
25

How to select a *"good"* threshold T?

1. Empirically
   - Difficult to find a common threshold that works for all images
   - Image normalization helps

$$p' = \frac{p - \mu}{\sigma}$$

Z-normalization also helps other algorithms, including the training of deep neural networks

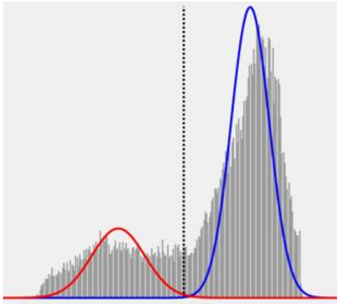


26

How to select a *"good"* threshold T?

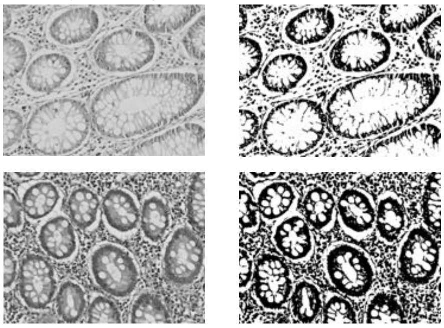
2. Finding a global threshold based on the histogram of pixel values
   - Otsu's method searches for the threshold T that minimizes the intra-class variance (the variance of background and foreground pixel values weighted by the class probabilities)

$$\sigma_w^2(t) = w_0(t)\,\sigma_0^2(t) + w_1(t)\,\sigma_1^2(t)$$



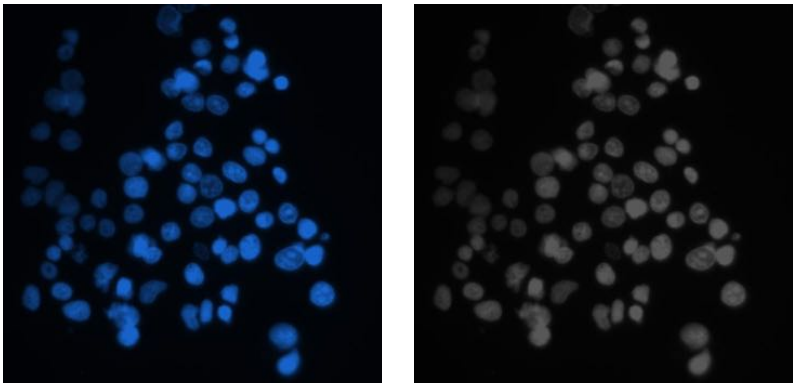
Grayscale image

Thresholding grayscale with the Otsu's thresholds (T = 171 And T = 152)

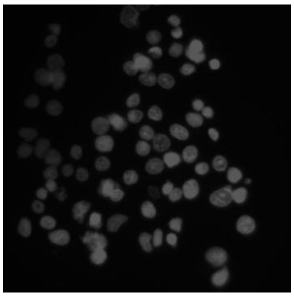


How to select a *"good"* threshold T?

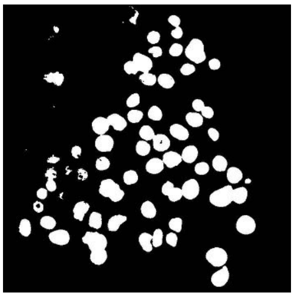
3. Using adaptive (local) thresholds
   - Global threshold does not work well when there are variations in the pixel values, for example, due to illumination



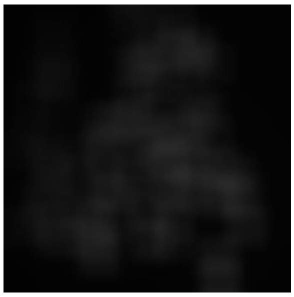
Using an average filter for adaptive thresholding



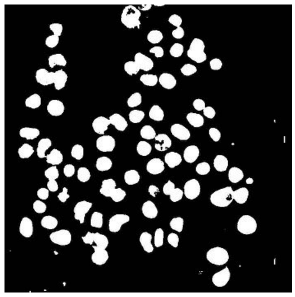
Grayscale image



BW image obtained by thresholding the grayscale with the Otsu's threshold

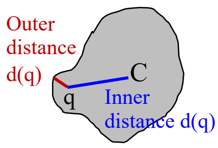


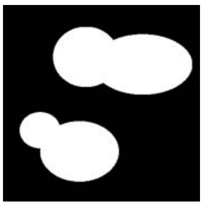
The grayscale image is smoothed with a 49x49 average filter



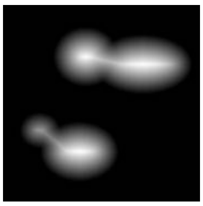
BW image where grayscale > filter response

Regional maxima/minima

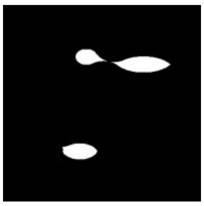




Binary image



Outer distance transform



Threshold with 40

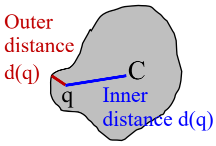
Threshold with 30

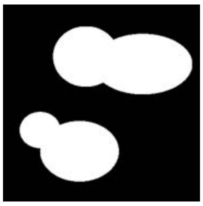
Outer distances are expected to be larger at the cell centers. However, when there are cells with different sizes, a global threshold may not work.

- Regional maxima (minima) are connected components of pixels with the same intensity value, surrounded by pixels with a lower (greater) value

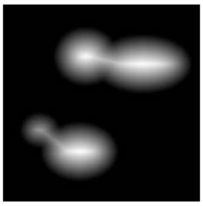
5	5	5	5	5	5	5	5	5	5
5	5	5	5	5	5	5	5	5	5
5	5	6	7	5	5	1	5	5	5
5	5	7	7	5	5	4	6	5	5
5	5	5	5	5	5	5	1	5	5
5	3	2	1	5	5	5	5	9	5
5	4	5	3	5	5	5	5	5	5
5	3	5	3	5	1	1	1	5	5
5	3	3	3	5	1	5	1	5	5
5	5	5	5	5	1	1	1	1	5

Regional maxima/minima

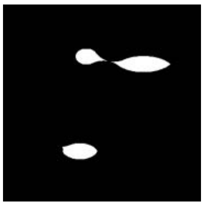




Binary image



Outer distance transform



Threshold with 40

Threshold with 30

Outer distances are expected to be larger at the cell centers. However, when there are cells with different sizes, a global threshold may not work.

Due to noise, finding regional maxima directly on an outer distance map may lead to spurious cell locations.

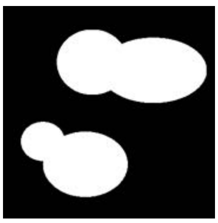
31

H-maxima/minima transform

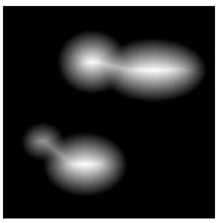
- H-maxima (minima) transform suppresses all maxima (minima) whose height is less than H
- Height of the remaining maxima is decreased (increased) by H

1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	7	7	1	1	1	2	2	1
1	1	7	7	1	1	1	2	2	1
1	1	1	1	1	1	1	1	1	1
1	2	2	2	2	1	1	2	2	2
1	2	3	3	2	1	1	2	4	2
1	2	5	5	3	1	1	2	4	2
1	2	2	3	2	1	1	2	2	2
1	1	1	1	1	1	1	1	1	1

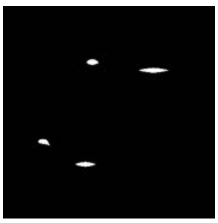
Before applying h-maxima transform



Binary image



Outer distance transform



Regional maxima after applying h-maxima transform with h = 4

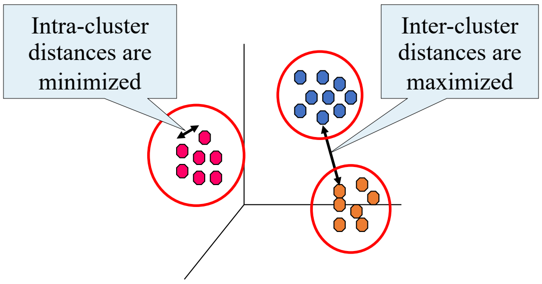
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	1	5	5	1	1	1	1	1	1
1	1	5	5	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
1	2	2	2	2	1	1	2	2	2
1	2	3	3	2	1	1	2	2	2
1	2	3	3	3	1	1	2	2	2
1	2	2	3	2	1	1	2	2	2
1	1	1	1	1	1	1	1	1	1

After applying h-maxima transform with h = 2

32

Clustering

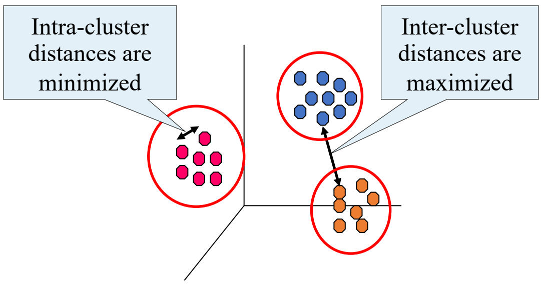
- Finds groups of pixels/voxels such that the pixels/voxels in a group will be similar (or related) to one another and different from (or unrelated to) the pixels/voxels in other groups
- Pixels/voxels are points in a high-dimensional space
  - Grayscale: 1D
  - Color: 3D
  - Color and location: 5D
  - Responses from 13 Gabor filters: 13D
  - …



33

K-means

- Extremely simple but also efficient clustering algorithm
- Represents each cluster as a mean vector, which is the average of points (pixels/voxels) assigned to this cluster
- But here we have a problem
  - To assign the point to the clusters, we should have calculated the mean vectors
  - To calculate the mean vectors, we should have assigned the points to the clusters



34

K-means

```
start with randomly initialized mean (cluster)
vectors m_i

do

   for each cluster, estimate samples that belong
   to the mean vector m_i  (estimate a dataset D_i)      } E-step

   for each cluster, compute the mean vector m_i
   that minimizes/maximizes the criterion function       } M-step

until there is no (or small) change in m_i
```

This is an example of the expectation-maximization (EM) algorithm

35

K-means

D_i estimation

Each cluster contains samples that are most similar to m_i

$$D_i = \left\{ x \;\middle|\; \underbrace{\| x - m_i \|^2}_{\text{Euclidean distance}} = \min_j \| x - m_j \|^2 \right\}$$

m_i computation

Set a value as to minimize/maximize a criterion function

$$\frac{\partial E}{\partial m_i} = 0 \qquad E = \underbrace{\frac{1}{2} \sum_{i=1}^{k} \sum_{x \in D_i} \| x - m_i \|^2}_{\text{Sum of squared error}}$$

$$\frac{\partial E}{\partial m_i} = \frac{1}{2} \sum_{x \in D_i} -2\,(x - m_i) = -\sum_{x \in D_i} x + \sum_{x \in D_i} m_i$$

$$m_i = \frac{\sum_{x \in D_i} x}{n_i}$$

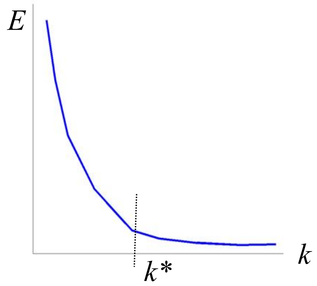
Number of samples in D_i

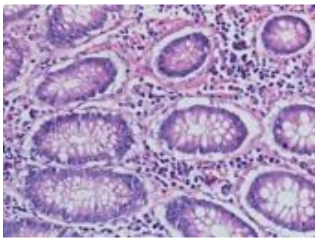
Possible to use other distance (similarity) metrics and criterion functions

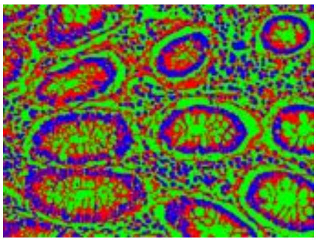
36

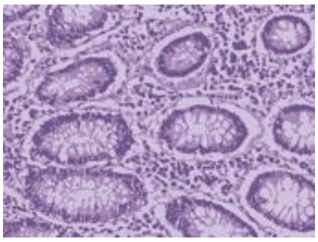
K-means

- How to select the number of clusters?
  - May have task-specific knowledge
  - May have some limitations (e.g., the maximum number of bits to represent a cluster id)
  - Can check how error decreases as a function of the cluster number









37

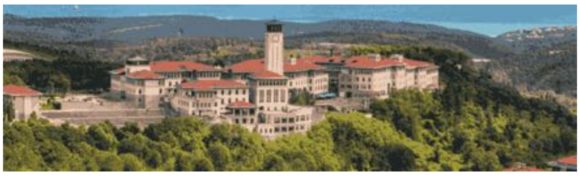

Original image


K = 8


K = 3


K = 12

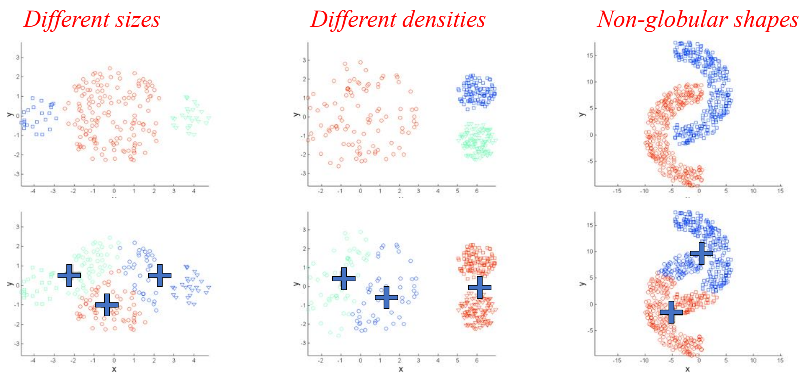

K = 4


K = 16

38

K-means

- Limitations: It detects spherical clusters but cannot handle clusters of different sizes, densities, and/or non-globular shapes and it is sensitive to outliers

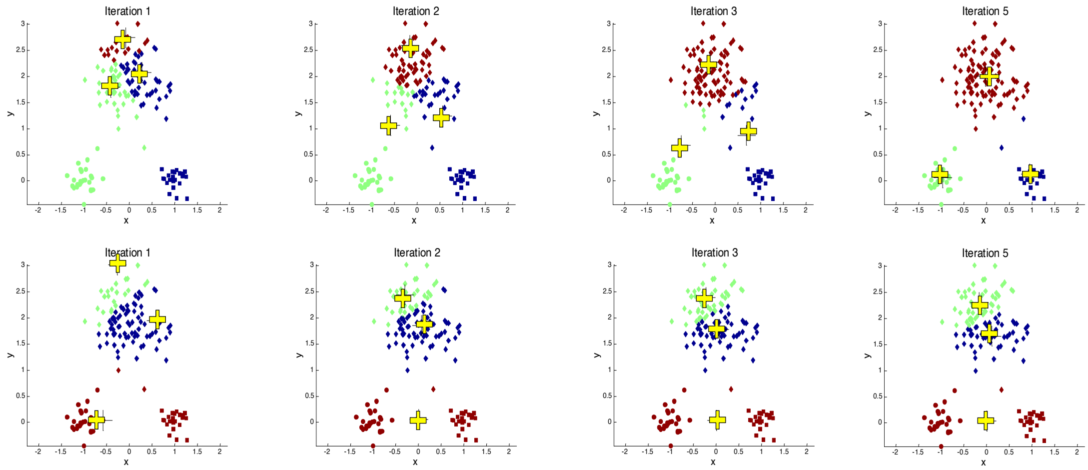


Slide: Tan, Steinbach, Kumar, Introduction to Data Mining

39

K-means

- Potential problem: Final clusters highly depends on the initial mean vectors



Slide: Tan, Steinbach, Kumar, Introduction to Data Mining

40

Thank you!

Next time:

More on medical image segmentation

41