Introduction to Computer Vision

Segmentation

Today

- Image Segmentation

Image source

Image Segmentation

- Aim: to partition an image into a collection of set of pixels
  - Meaningful regions (coherent objects)
  - Linear structures (line, curve, …)
  - Shapes (circles, ellipses, …)

Image Segmentation

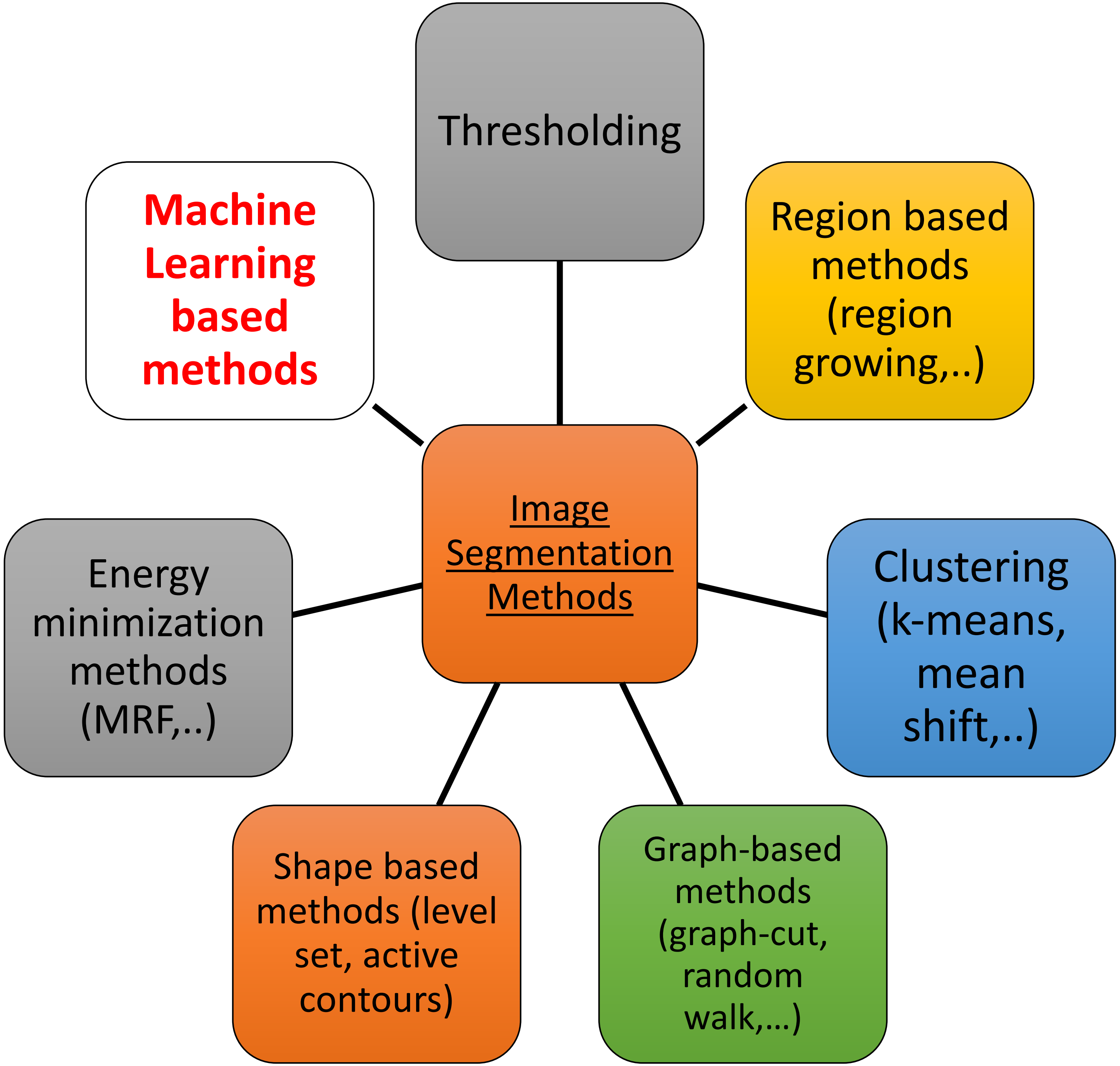
- Aim: to partition an image into a collection of set of pixels
  - Meaningful regions (coherent objects)
  - Linear structures (line, curve, …)
  - Shapes (circles, ellipses, …)

- Content based image retrieval
- Medical Imaging applications (tumor delineation,..)
- Object detection (face detection,…)
- 3D Reconstruction
- Object/Motion Tracking
- Object-based measurements such as size and shape
- Object recognition (face recognition,…)
- Fingerprint recognition,
- Video surveillance

…

Image Segmentation

- One of the oldest and most widely studied problems
  - *Early techniques -> region splitting or merging*
  - *More recent techniques -> Energy minimization, hybrid methods, and deep learning*



Thresholding

Machine Learning based methods

Region based methods (region growing,..)

Energy minimization methods (MRF,..)

Image Segmentation Methods

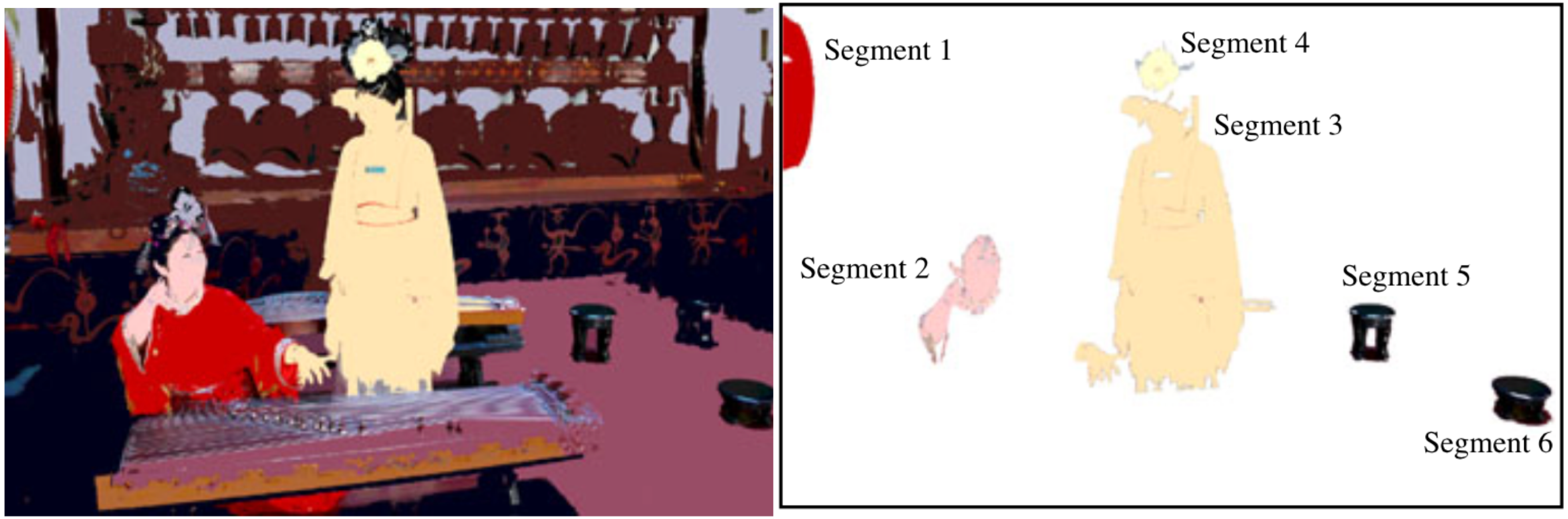
Clustering (k-means, mean shift,..)

Shape based methods (level set, active contours)

Graph-based methods (graph-cut, random walk,…)

Basics of Image Segmentation

- <u>Definition</u>: *Image segmentation* partitions an image into regions called segments.



Basics of Image Segmentation

- Definition: *Image segmentation* partitions an image into regions called segments.

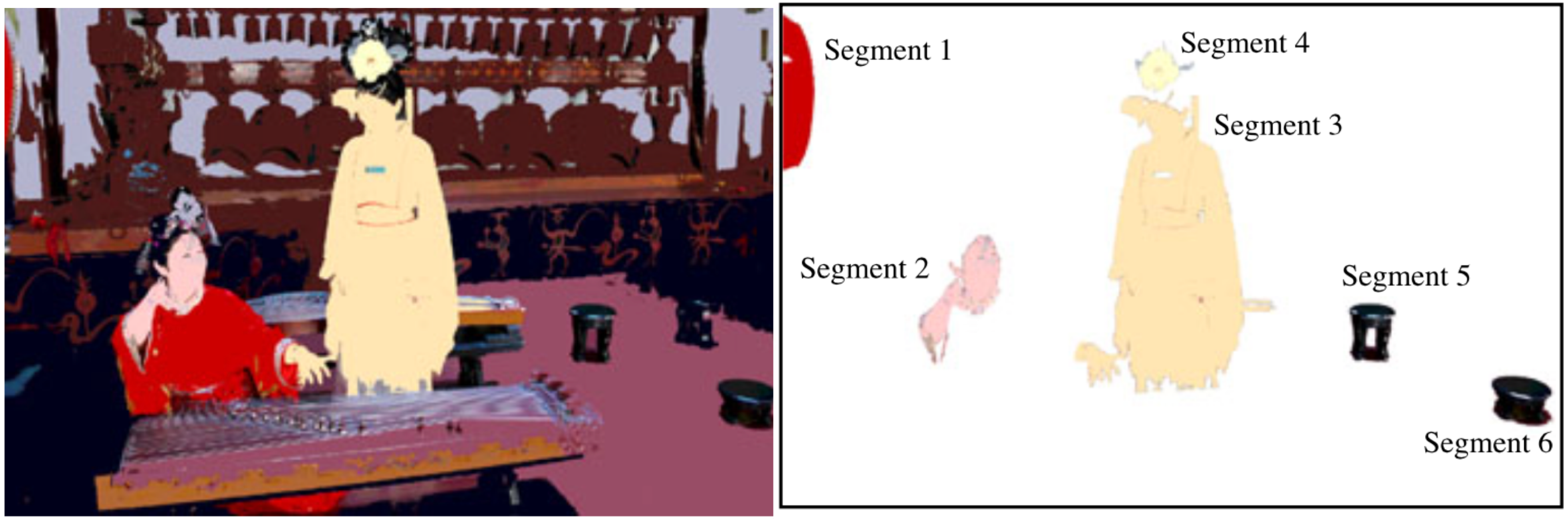


Image segmentation creates segments of connected pixels by analyzing the image w.r.t. some similarity criteria:

intensity, color, texture, histogram, features, …

Binary Images

- A global threshold T can be used to map a scalar image I into a binary image

Image Binarization

- A global threshold T can be used to map a scalar image I into a binary image

$$J(x,y) = \begin{cases} 0 & \text{if } I(x,y) < T \\ 1 & \text{otherwise.} \end{cases}$$

Image Binarization

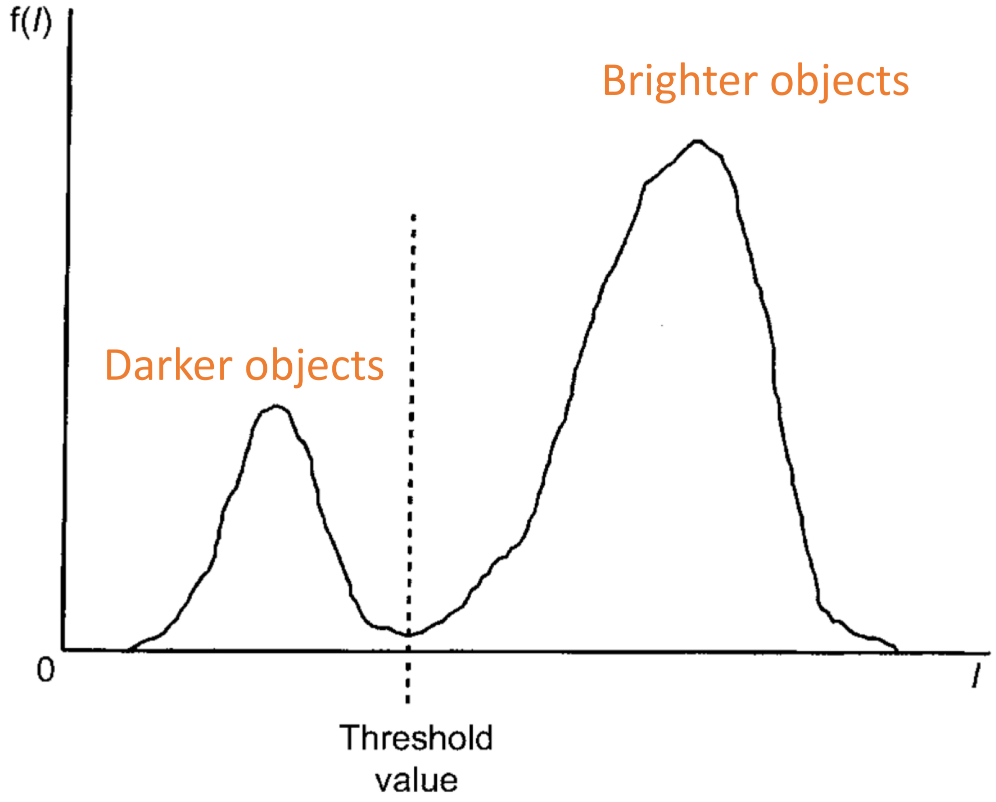
- A global threshold T can be used to map a scalar image I into a binary image

$$J(x,y) = \begin{cases} 0 & \text{if } I(x,y) < T \\ 1 & \text{otherwise.} \end{cases}$$

- The global threshold can be identified by an optimization strategy aiming at creating “large” connected regions and at reducing the number of small-sized regions, called *artifacts*.

Image Binarization

- Thresholding: Most frequently employed method for determining threshold is based on histogram analysis of intensity levels.



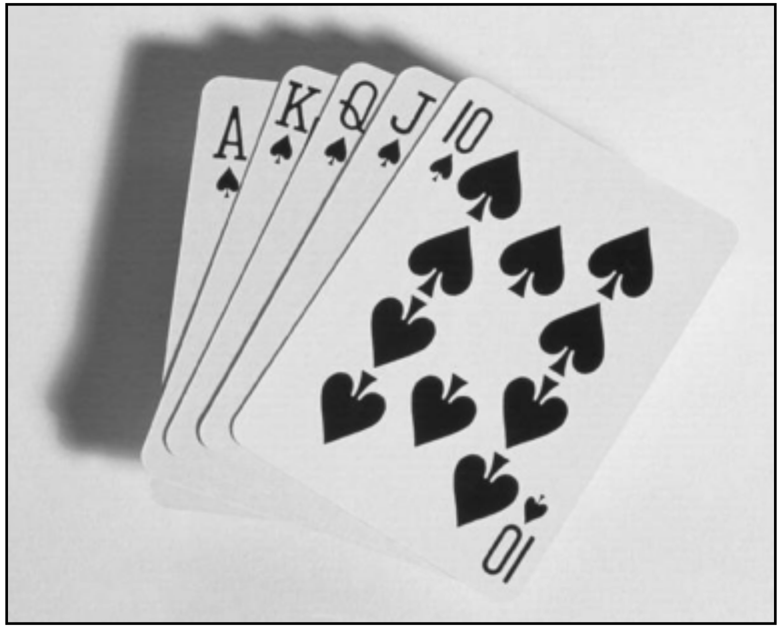
Peak on the left of the histogram corresponds to dark objects

Peak on the right of the histogram corresponds to brighter objects

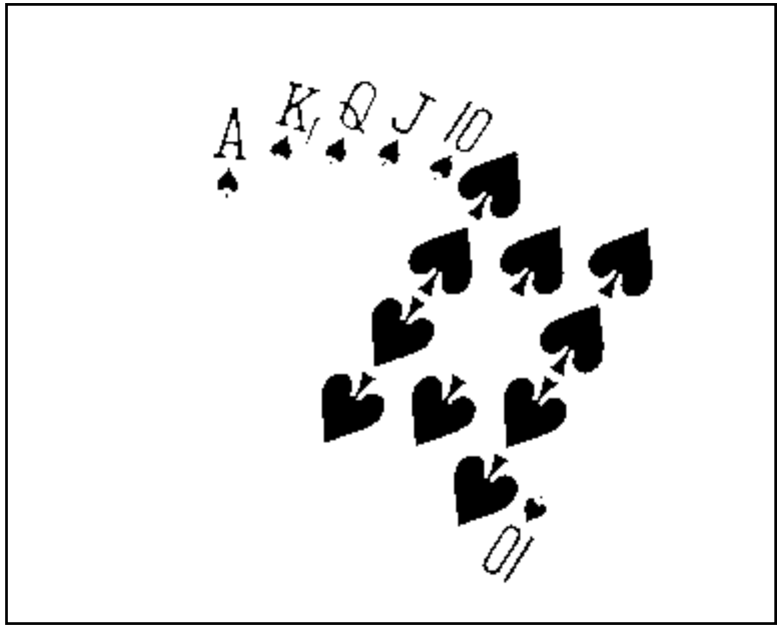
DIFFICULTIES

1. The valley may be so broad that it is difficult to locate a significant minimum
2. Number of minima due to type of details in the image
3. Noise
4. No visible valley
5. Histogram may be multi-modal

Thresholding Example

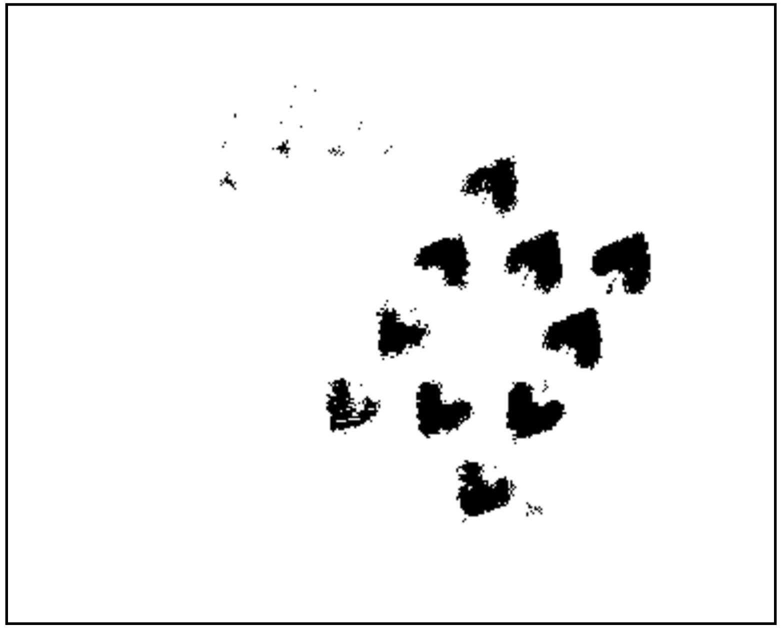


Original Image

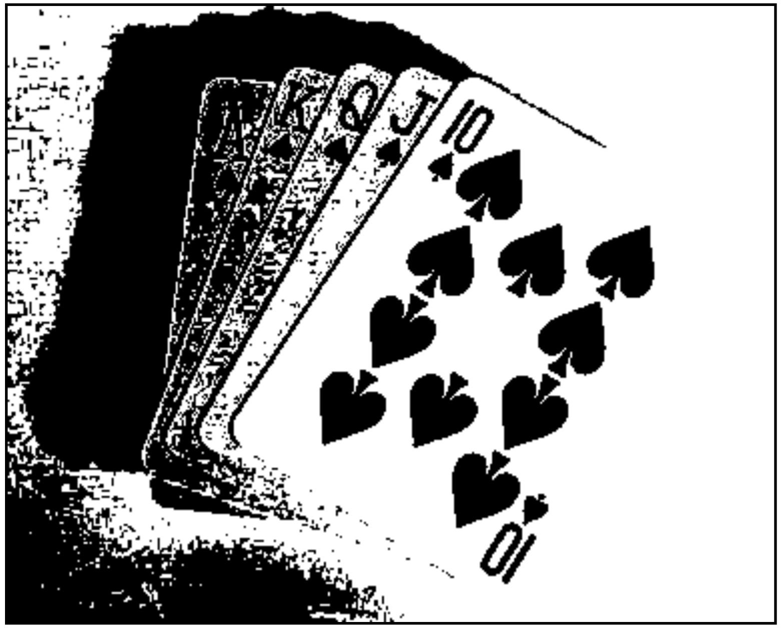


Thresholded Image

Thresholding Example 2

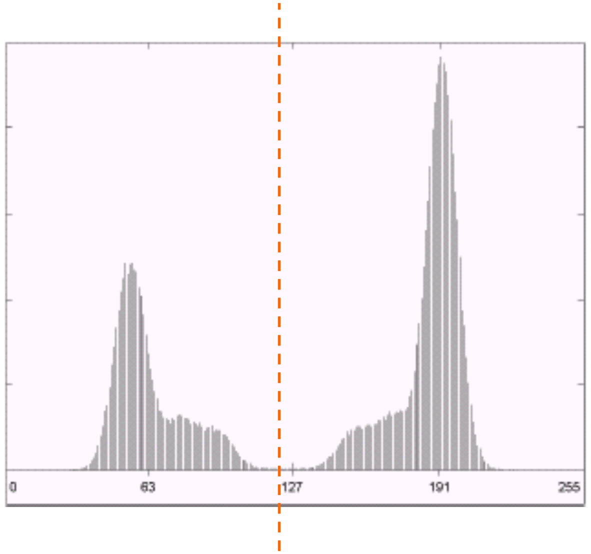


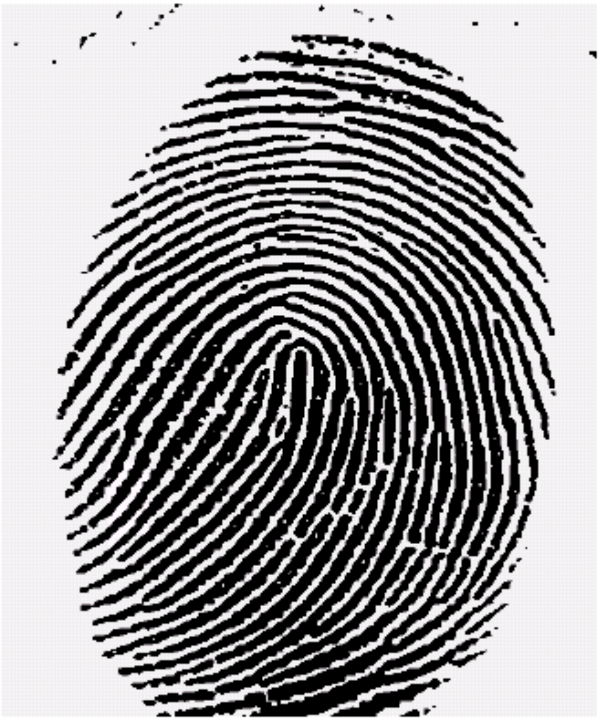
Threshold Too Low



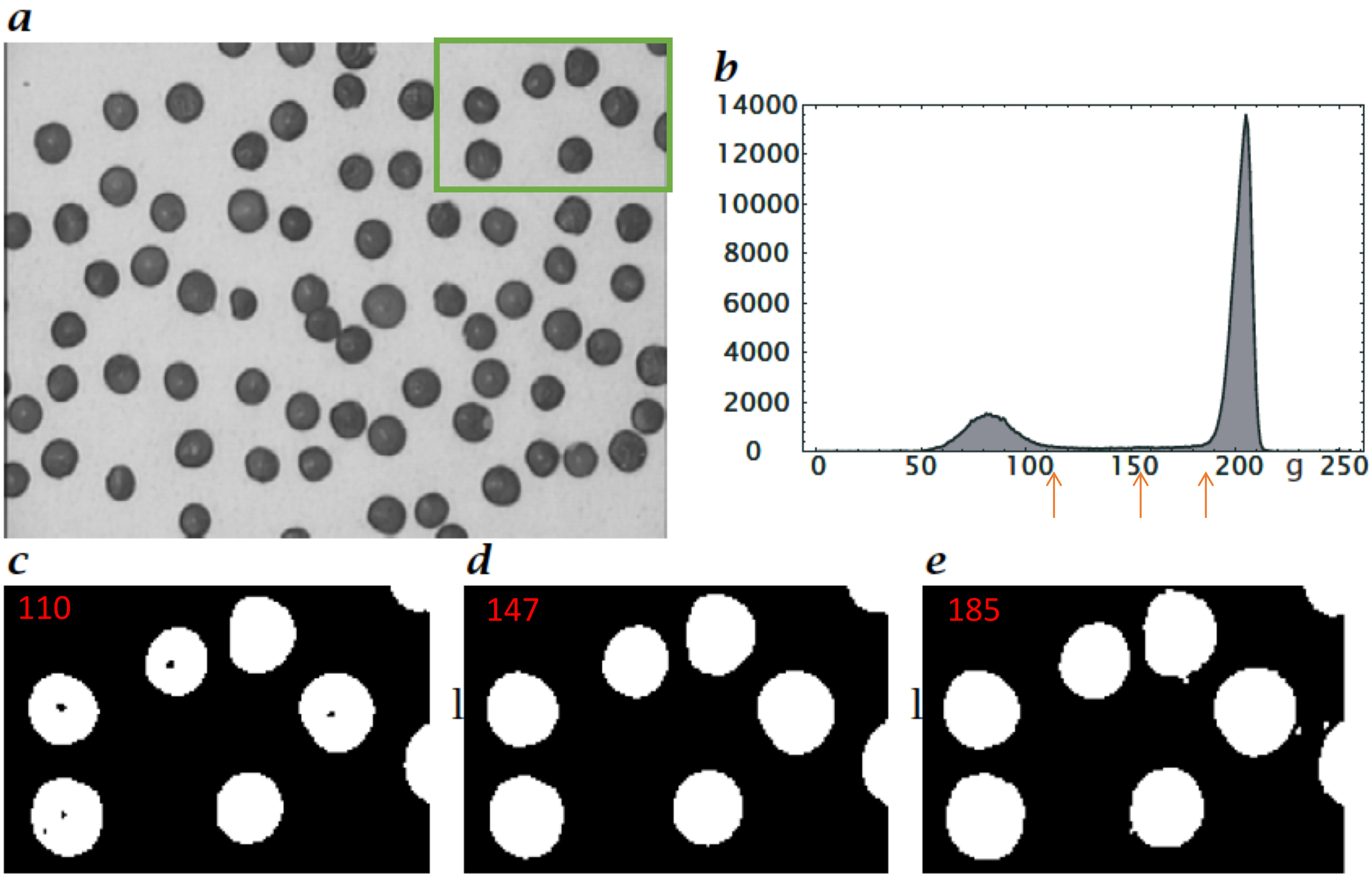
Threshold Too High

Thresholding Example 3





Thresholding Example-4



Otsu Thresholding

- **<u>Definition:</u>** The method uses the grey-value histogram of the given image I as input and aims at providing the best threshold in the sense that the “overlap” between two classes, set of object and background pixels, is minimized (i.e., by finding the best balance).

Otsu Thresholding

- **<u>Definition:</u>** The method uses the grey-value histogram of the given image I as input and aims at providing the best threshold in the sense that the "overlap" between two classes, set of object and background pixels, is minimized (i.e., by finding the best balance).

- Otsu's algorithm selects a threshold that maximizes the <u>between-class variance</u> σ_b^2. In the case of two classes,

$$\sigma_b^2 = P_1(\mu_1 - \mu)^2 + P_2(\mu_2 - \mu)^2 = P_1 P_2(\mu_1 - \mu_2)^2$$

where P_1 and P_2 denote class probabilities, and μ_i the means of object and background classes.

- Let c_I be the relative cumulative histogram of an image I, then P_1 and P_2 are approximated by $c_I(u)$ and $1 - c_I(u)$, respectively.
- u is assumed to be the chosen threshold.

Otsu Thresholding Algorithm

```
Compute histogram $H_I$ for $u = 0, \ldots, G_{\max}$ ;
Let $T_0$ be the increment for potential thresholds; $u = T_0$; $T = u$; and $S_{\max} = 0$;
while $u < G_{\max}$ do
    Compute $c_I(u)$ and $\mu_i(u)$ for $i = 1, 2$ ;
    Compute $\sigma_b^2(u) = c_I(u)[1 - c_I(u)][\mu_i(u) - \mu_2(u)]^2$;
    if $\sigma_b^2(u) > S_{\max}$ then
        $S_{\max} = \sigma_b^2(u)$ and $T = u$;
    end if
    Set $u = u + T_0$
end while
```

<u>probabilities</u>

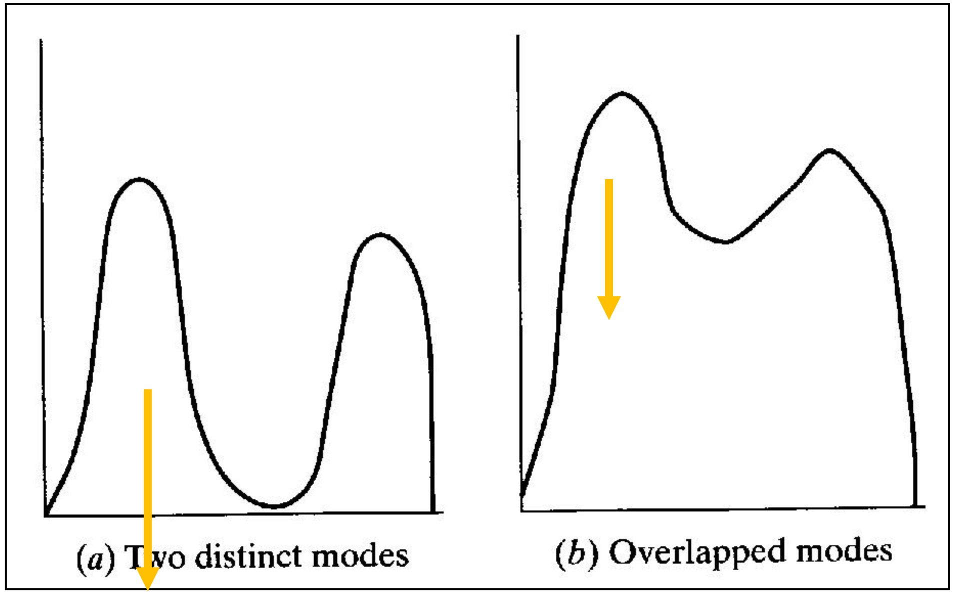
$$P_1 = \sum_{i=0}^{u} p(i)$$

$$P_2 = \sum_{i=u+1}^{G_{max}} p(i)$$

<u>Class means</u>

$$\mu_1 = \sum_{i=0}^{u} ip(i)/P_1$$

$$\mu_2 = \sum_{i=u+1}^{G_{max}} ip(i)/P_2$$



(*a*) Two distinct modes

(*b*) Overlapped modes

Example: Otsu Thresholding

$T = 63$

$T = 92$

$T = 141$

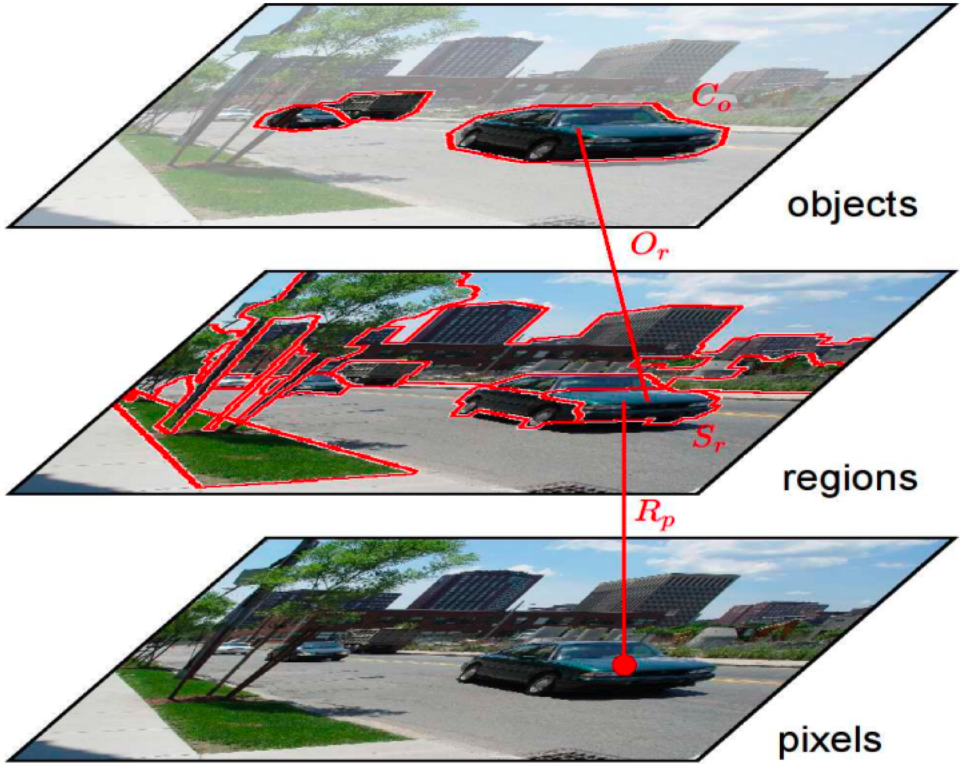
$T = 162$

$T = 187$

$T = 230$

Region Based Segmentation

Region Based Segmentation-Basics



Region: Spatial proximity + similarity

A group of connected pixels with similar properties

Closed boundaries

Computation of regions depends on similarity

Regions may correspond to objects in a scene or parts of the objects

Region Growing

- For segment generation in grey-level or color images, we may start at one *seed pixel* $(x,y,I(x,y))$ and add recursively adjacent pixels that satisfy a "similarity criterion" with pixels contained in the so-far grown region around the *seed pixel.*

Region Growing

- For segment generation in grey-level or color images, we may start at one *seed pixel* $(x,y,I(x,y))$ and add recursively adjacent pixels that satisfy a "similarity criterion" with pixels contained in the so-far grown region around the *seed pixel.*

- Defining similarity criteria alone is not an effective basis for segmentation
- It is necessary to consider the adjacency spatial relationship between pixels

Region Growing

- For segment generation in grey-level or color images, we may start at one ***seed pixel*** (x,y,I(x,y)) and add recursively adjacent pixels that satisfy a "similarity criterion" with pixels contained in the so-far grown region around the ***seed pixel.***

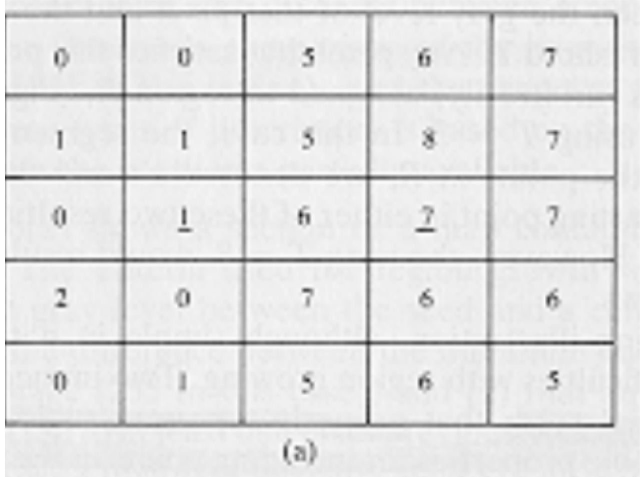
- Defining similarity criteria alone is not an effective basis for segmentation
- It is necessary to consider the adjacency spatial relationship between pixels

Algorithm

1. The absolute intensity difference between candidate pixel and the seed pixel must lie within a specified range
2. The absolute intensity difference between a candidate pixel and the running average intensity of the growing region must lie within a specified range;
3. The difference between the standard deviation in intensity over a specified local neighborhood of the candidate pixel and that over a local neighborhood of the candidate pixel must (or must not) exceed a certain threshold

Seeded Segmentation (Region Growing)

1. Choose the seed pixel

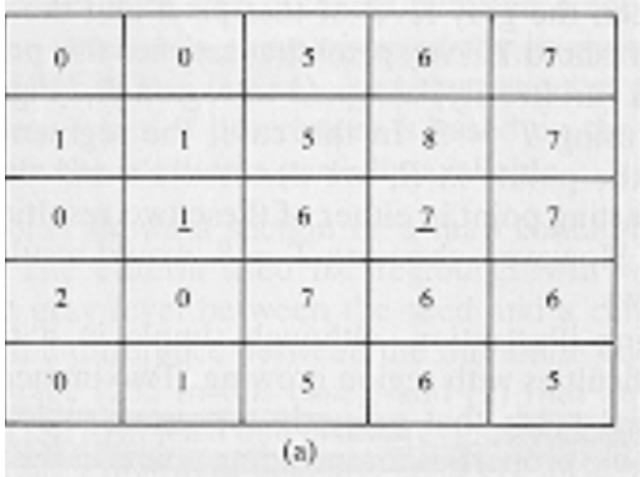


0	0	5	6	7
1	1	5	8	7
0	1	6	7	7
2	0	7	6	6
0	1	5	6	5

(a)

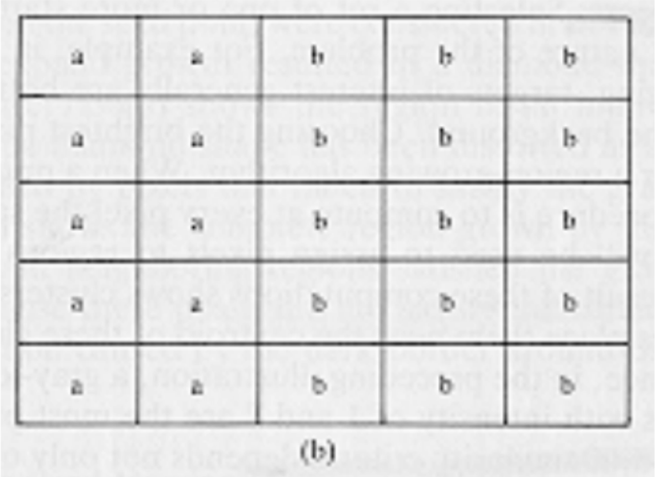
Seeded Segmentation (Region Growing)

1. Choose the seed pixel
2. Check the neighboring pixels and add them to the region if they are similar to the seed



0	0	5	6	7
1	1	5	8	7
0	1	6	7	7
2	0	7	6	6
0	1	5	6	5

(a)

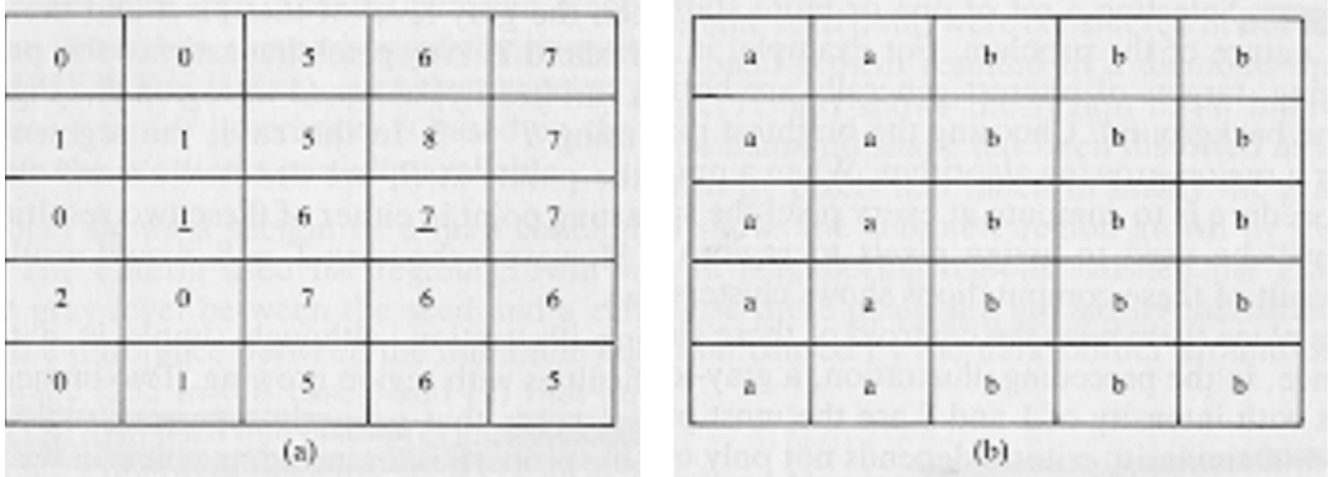


a	a	b	b	b
a	a	b	b	b
a	a	b	b	b
a	a	b	b	b
a	a	b	b	b

(b)

Seeded Segmentation (Region Growing)

1. Choose the seed pixel
2. Check the neighboring pixels and add them to the region if they are similar to the seed
3. Repeat step 2 for each of the newly added pixels; stop if no more pixels can be added



0	0	5	6	7
1	1	5	8	7
0	$\underline{1}$	6	$\underline{7}$	7
2	0	7	6	6
0	1	5	6	5

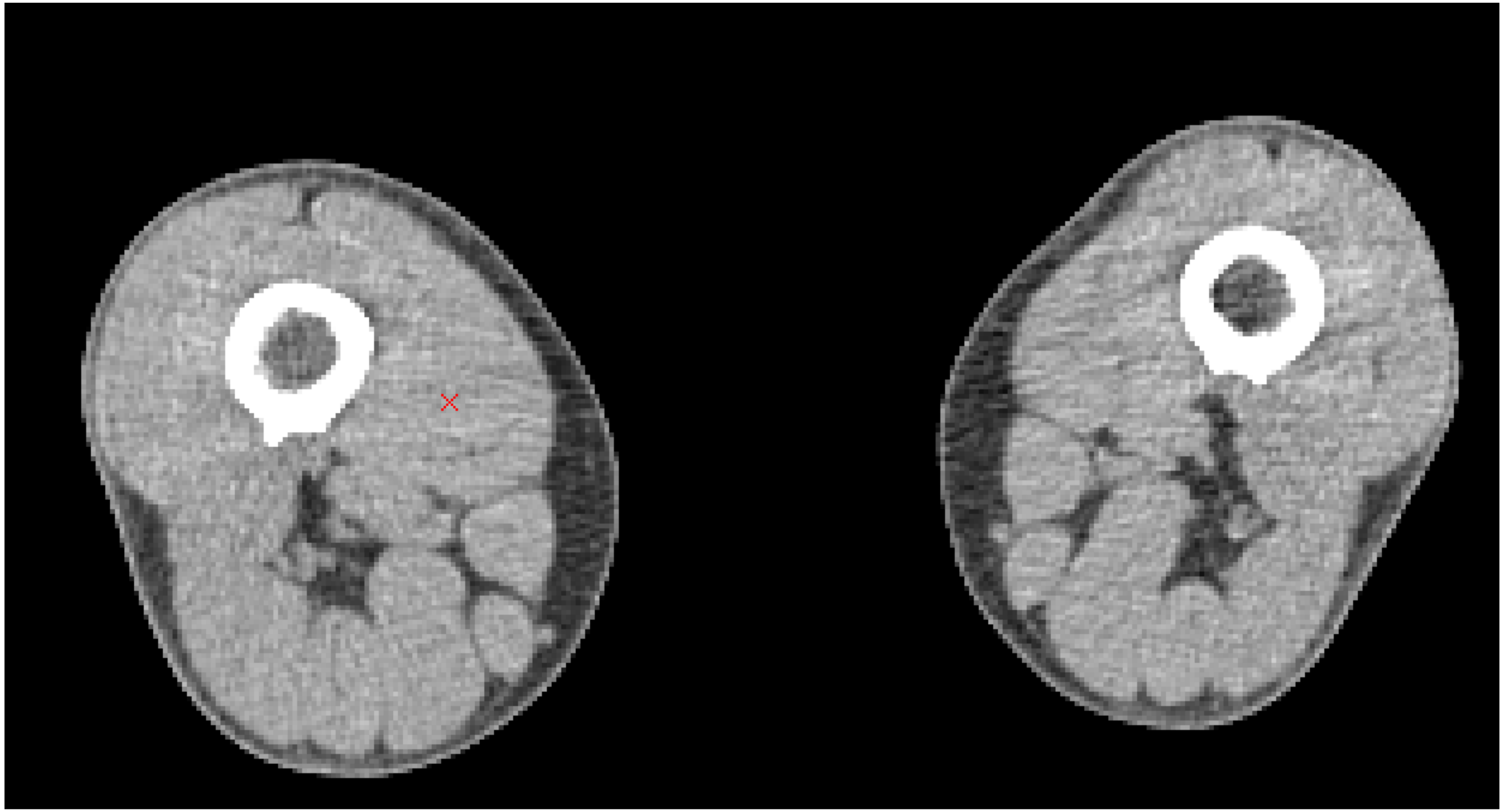
(a)

a	a	b	b	b
a	a	b	b	b
a	a	b	b	b
a	a	b	b	b
a	a	b	b	b

(b)

$$|\text{neighboring pixels} - \text{seed}| < Threshold$$

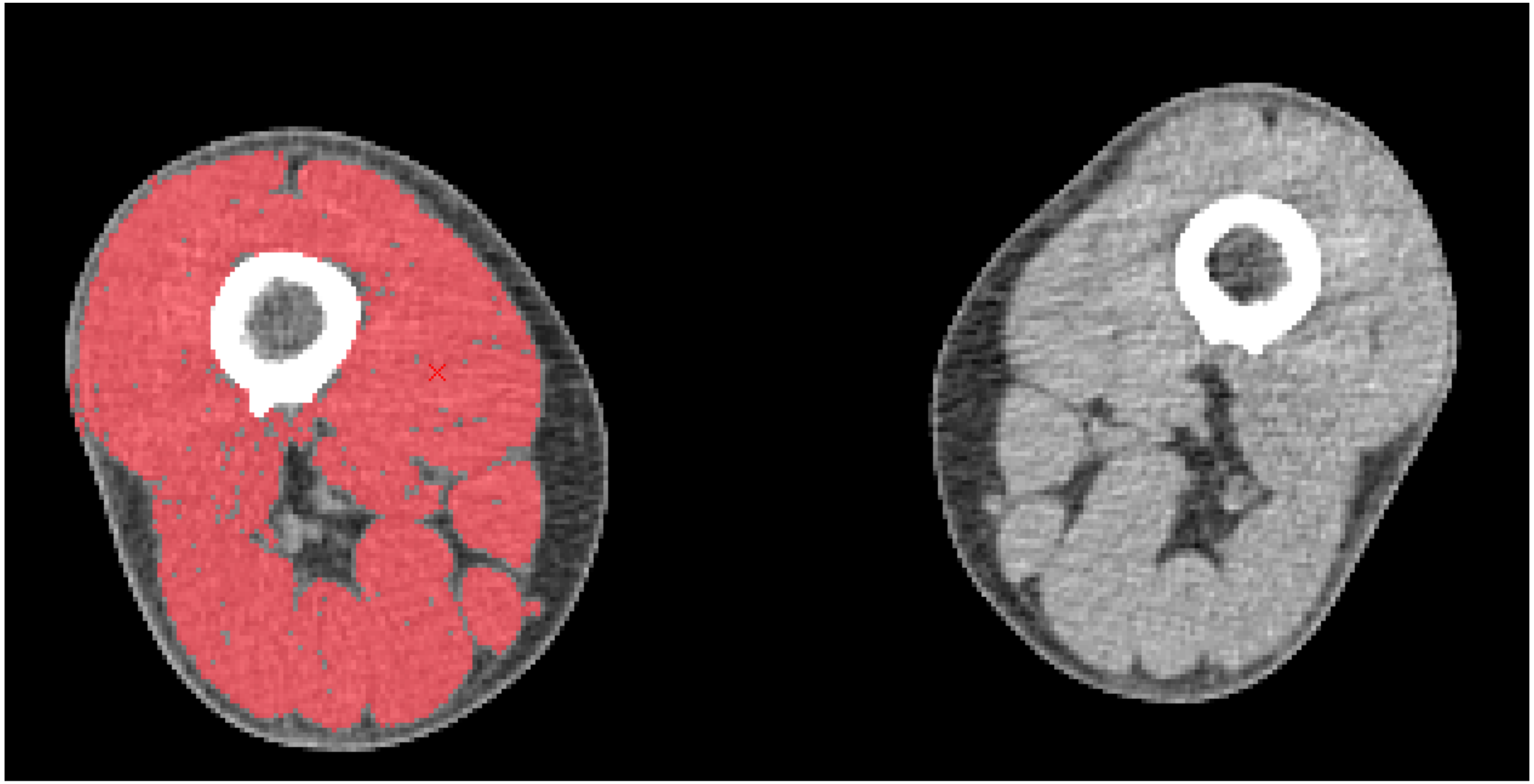
Ex: Muscle/Bone Segmentation in CT Scans



17

175

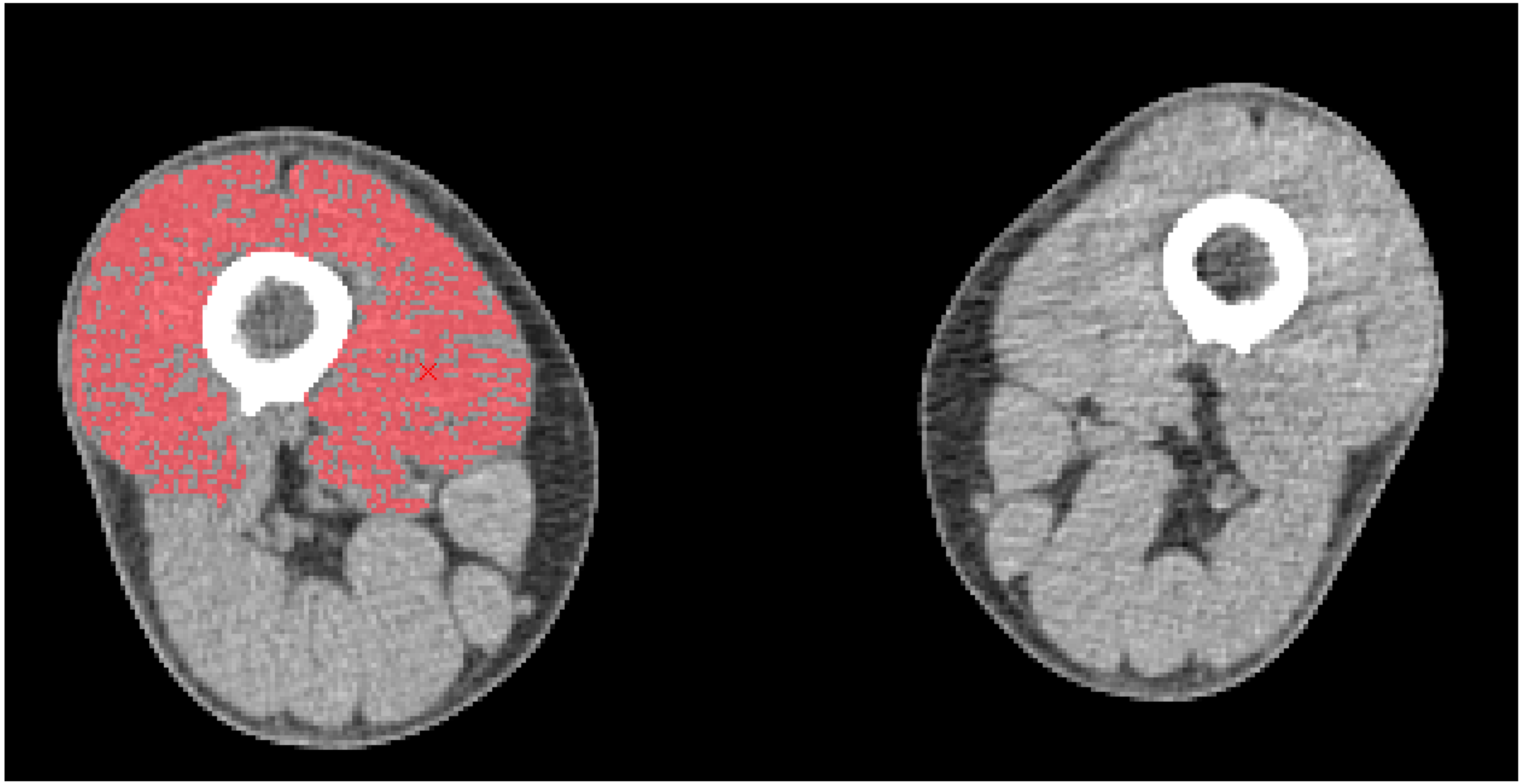
Ex: Muscle/Bone Segmentation in CT Scans



17

175

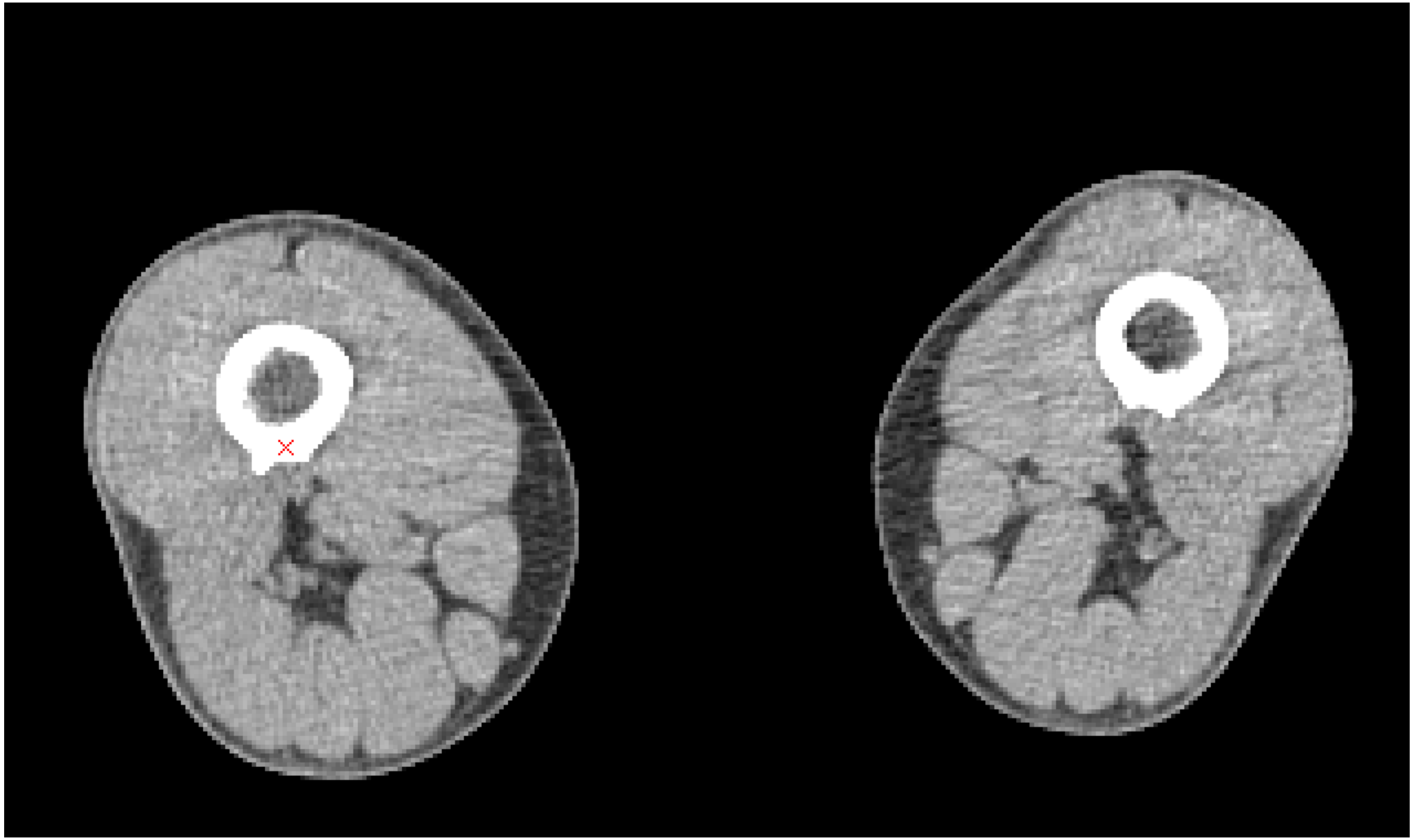
Ex: Muscle/Bone Segmentation in CT Scans



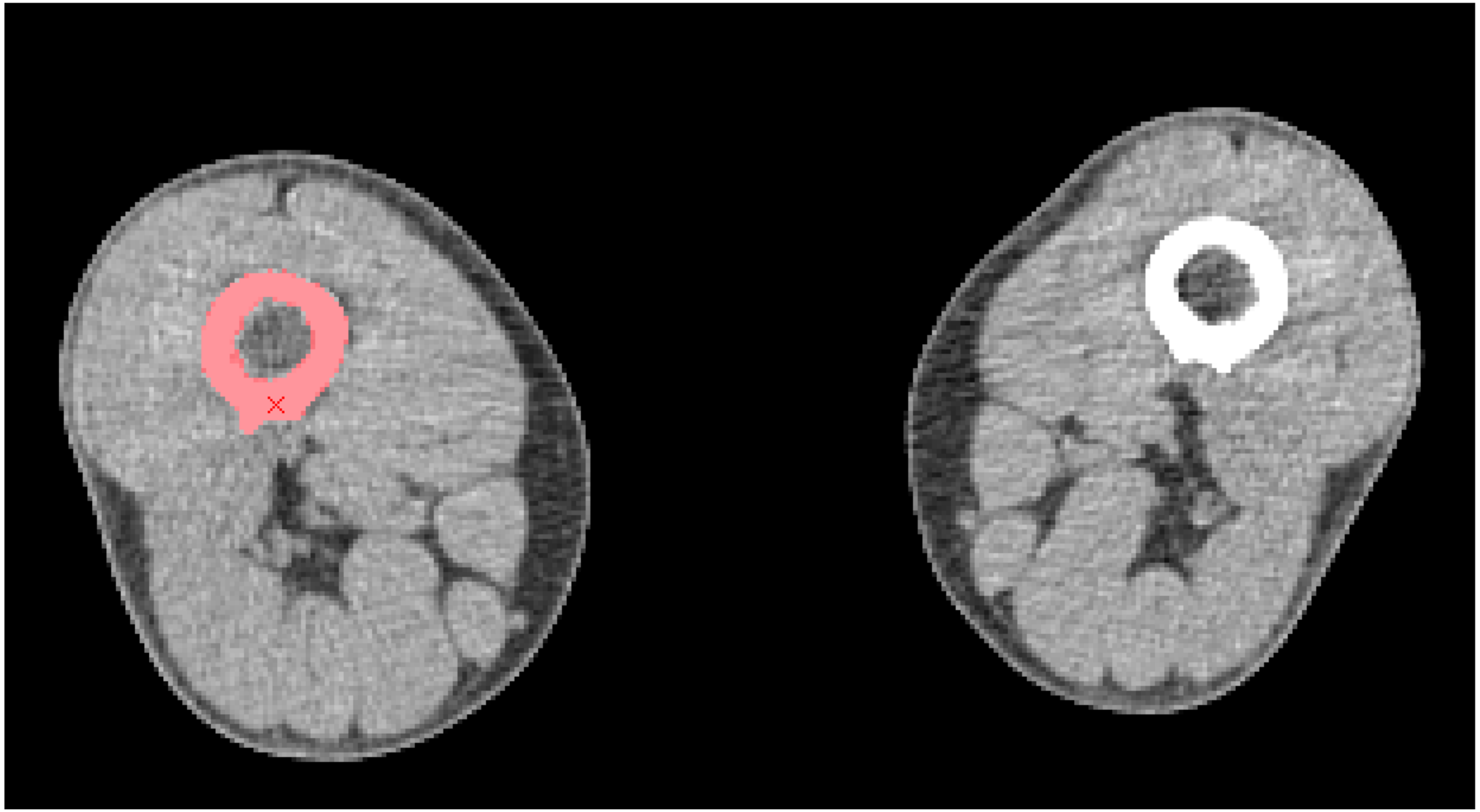
47

175

Ex: Muscle/Bone Segmentation in CT Scans



Ex: Muscle/Bone Segmentation in CT Scans



106

1688

Region splitting and Merging Segmentation

- Region splitting:
  - Unlike region growing, which starts from a set of seed points, region splitting starts with the whole image as a single region and subdivides it into subsidiary regions recursively while a condition of homogeneity is not satisfied.

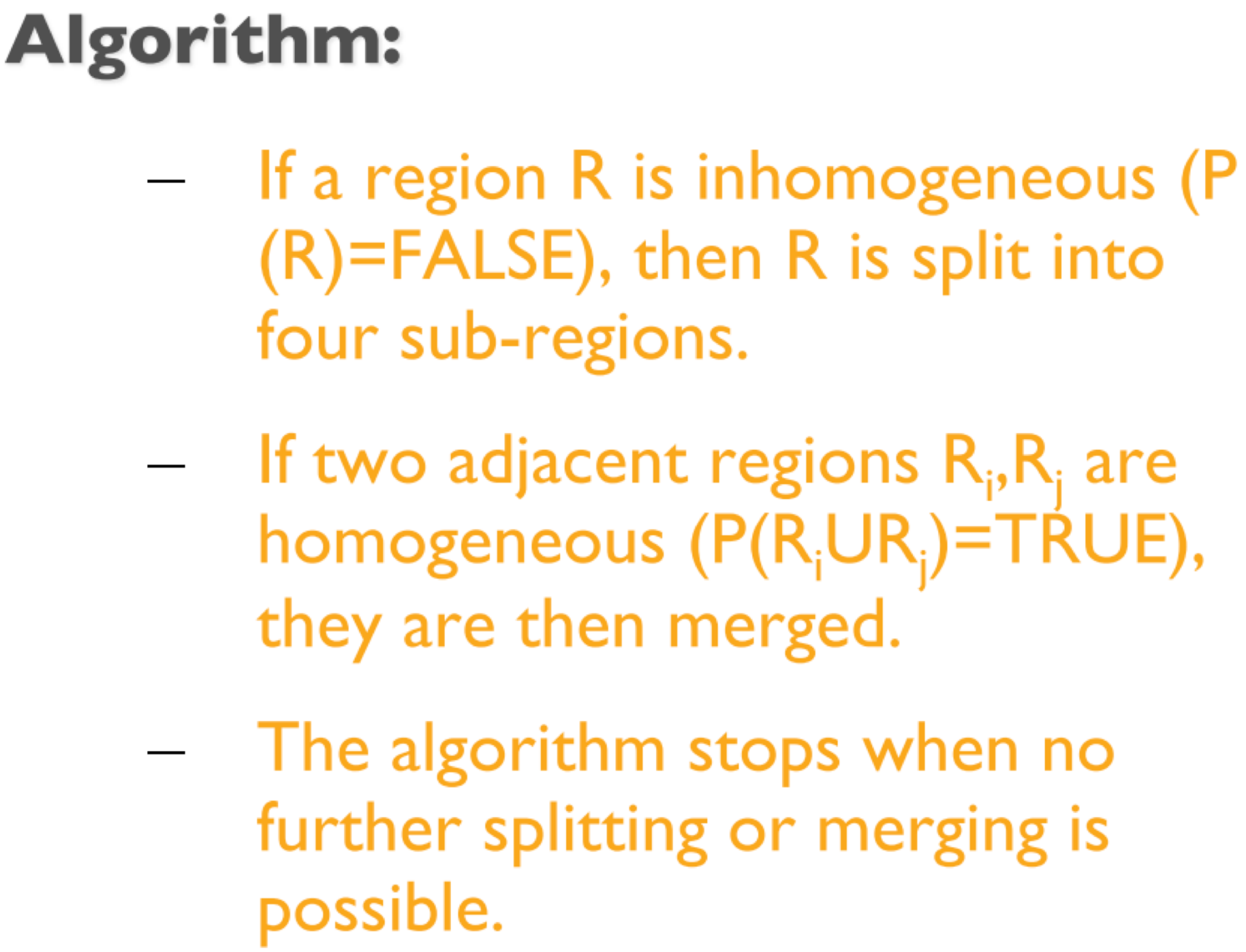
Region splitting and Merging Segmentation

- Region splitting:
  - Unlike region growing, which starts from a set of seed points, region splitting starts with the whole image as a single region and subdivides it into subsidiary regions recursively while a condition of homogeneity is not satisfied.
- Region merging:
  - Region merging is the opposite of splitting, and works as a way of avoiding over-segmentation

Region splitting and Merging Segmentation

- Region splitting:
  - Unlike region growing, which starts from a set of seed points, region splitting starts with the whole image as a single region and subdivides it into subsidiary regions recursively while a condition of homogeneity is not satisfied.
- Region merging:
  - Region merging is the opposite of splitting, and works as a way of avoiding over-segmentation
  - Start with small regions (2x2 or 4x4 regions) and merge the regions that have similar characteristics (such as gray level, variance).

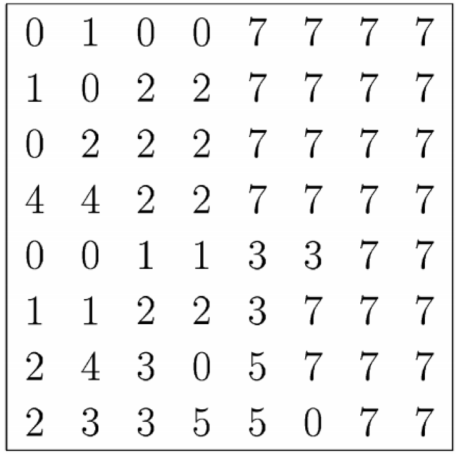
Region splitting and Merging Segmentation



Algorithm:

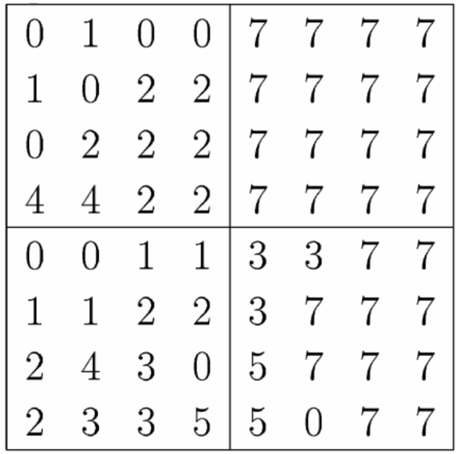
- If a region R is inhomogeneous (P (R)=FALSE), then R is split into four sub-regions.
- If two adjacent regions R_i,R_j are homogeneous ($P(R_i U R_j)$=TRUE), they are then merged.
- The algorithm stops when no further splitting or merging is possible.

Region splitting and Merging Segmentation



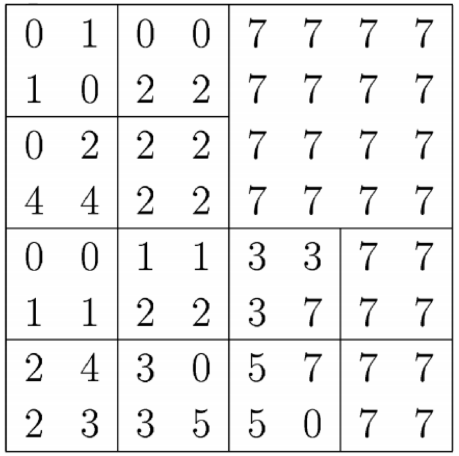
0	1	0	0	7	7	7	7
1	0	2	2	7	7	7	7
0	2	2	2	7	7	7	7
4	4	2	2	7	7	7	7
0	0	1	1	3	3	7	7
1	1	2	2	3	7	7	7
2	4	3	0	5	7	7	7
2	3	3	5	5	0	7	7

original image



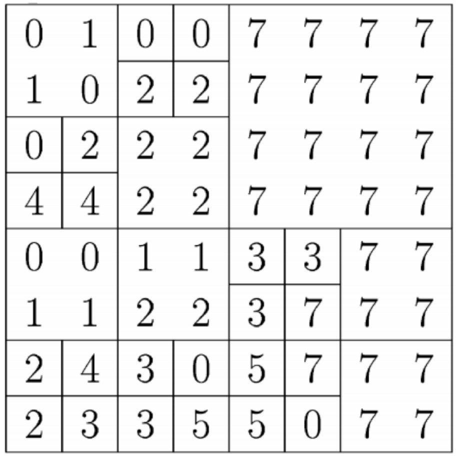
0	1	0	0	7	7	7	7
1	0	2	2	7	7	7	7
0	2	2	2	7	7	7	7
4	4	2	2	7	7	7	7
0	0	1	1	3	3	7	7
1	1	2	2	3	7	7	7
2	4	3	0	5	7	7	7
2	3	3	5	5	0	7	7

split 1



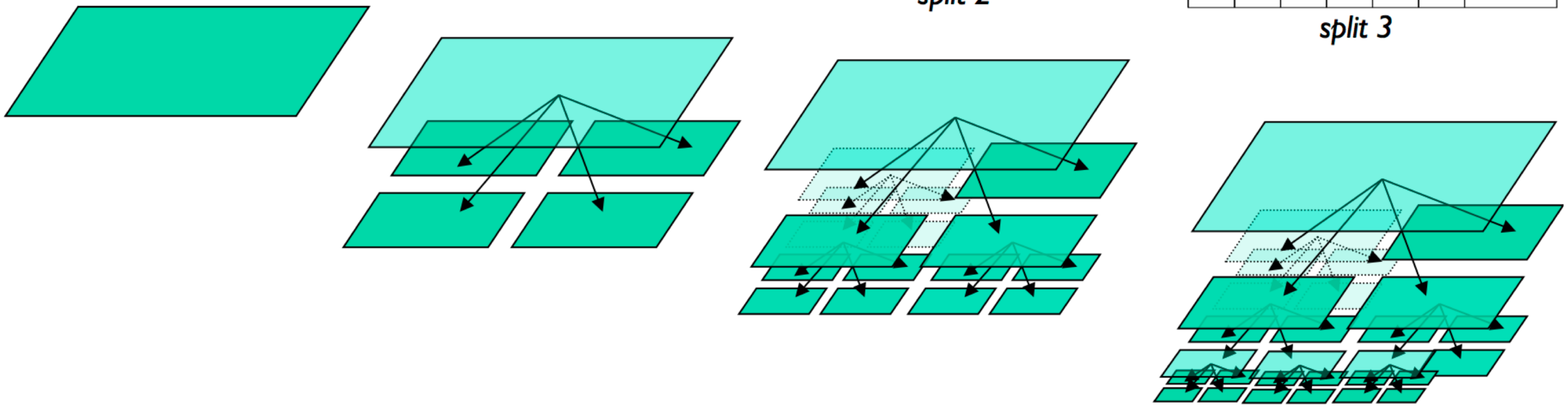
0	1	0	0	7	7	7	7
1	0	2	2	7	7	7	7
0	2	2	2	7	7	7	7
4	4	2	2	7	7	7	7
0	0	1	1	3	3	7	7
1	1	2	2	3	7	7	7
2	4	3	0	5	7	7	7
2	3	3	5	5	0	7	7

split 2



0	1	0	0	7	7	7	7
1	0	2	2	7	7	7	7
0	2	2	2	7	7	7	7
4	4	2	2	7	7	7	7
0	0	1	1	3	3	7	7
1	1	2	2	3	7	7	7
2	4	3	0	5	7	7	7
2	3	3	5	5	0	7	7

split 3

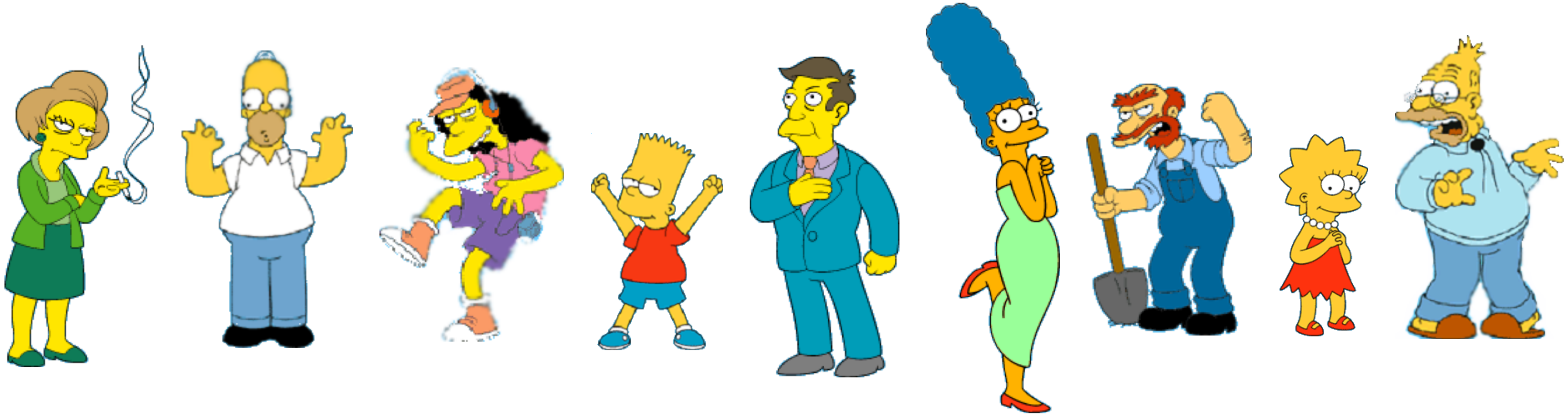


Clustering Based Segmentation Methods

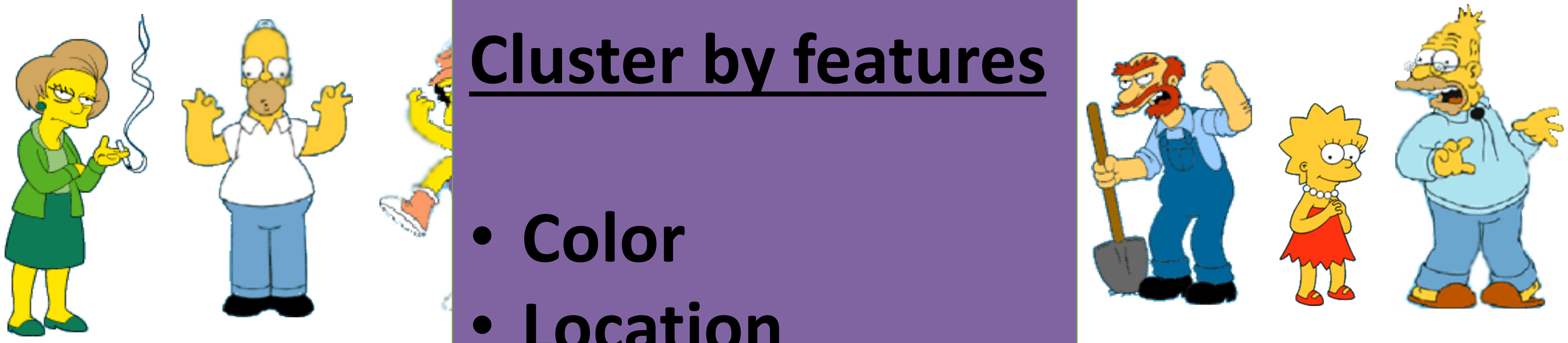
What is Clustering?

- Organizing data into classes such that:
    - High intra-class similarity
    - Low inter-class similarity
- Finding the class labels and the number of classes directly from the data (as opposed to *classification tasks*)

What is a natural grouping?

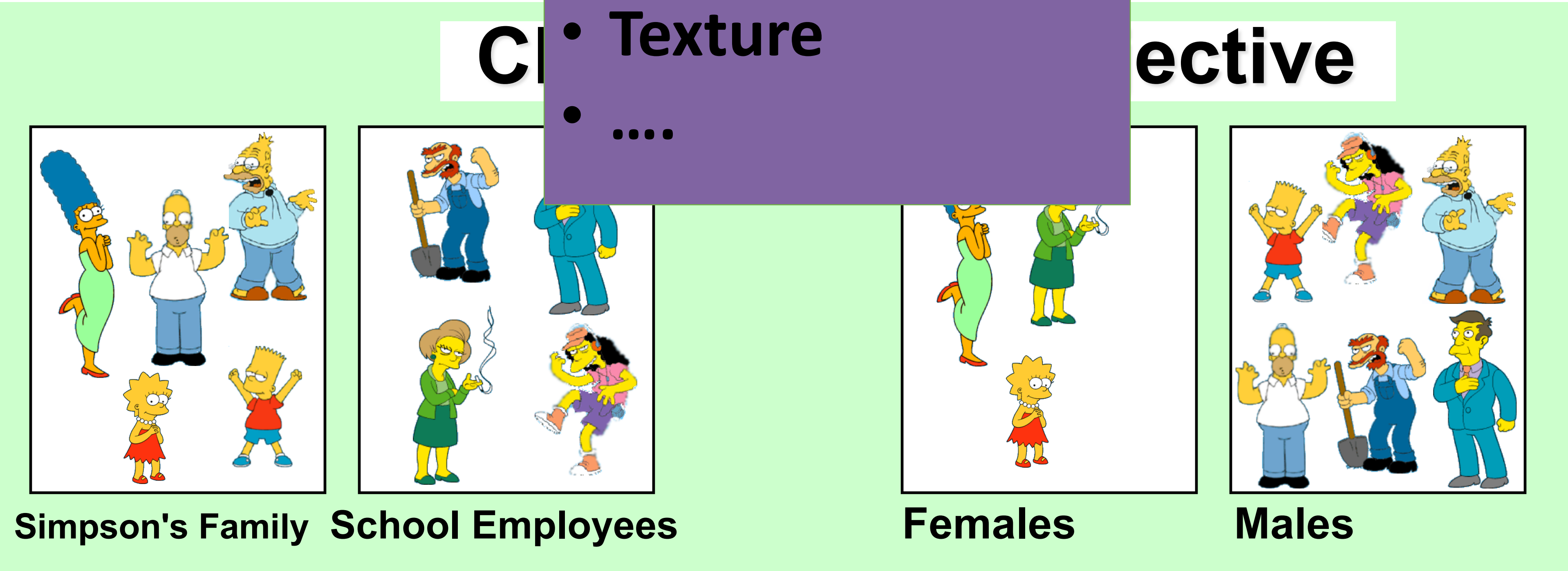


What is a natural grouping?



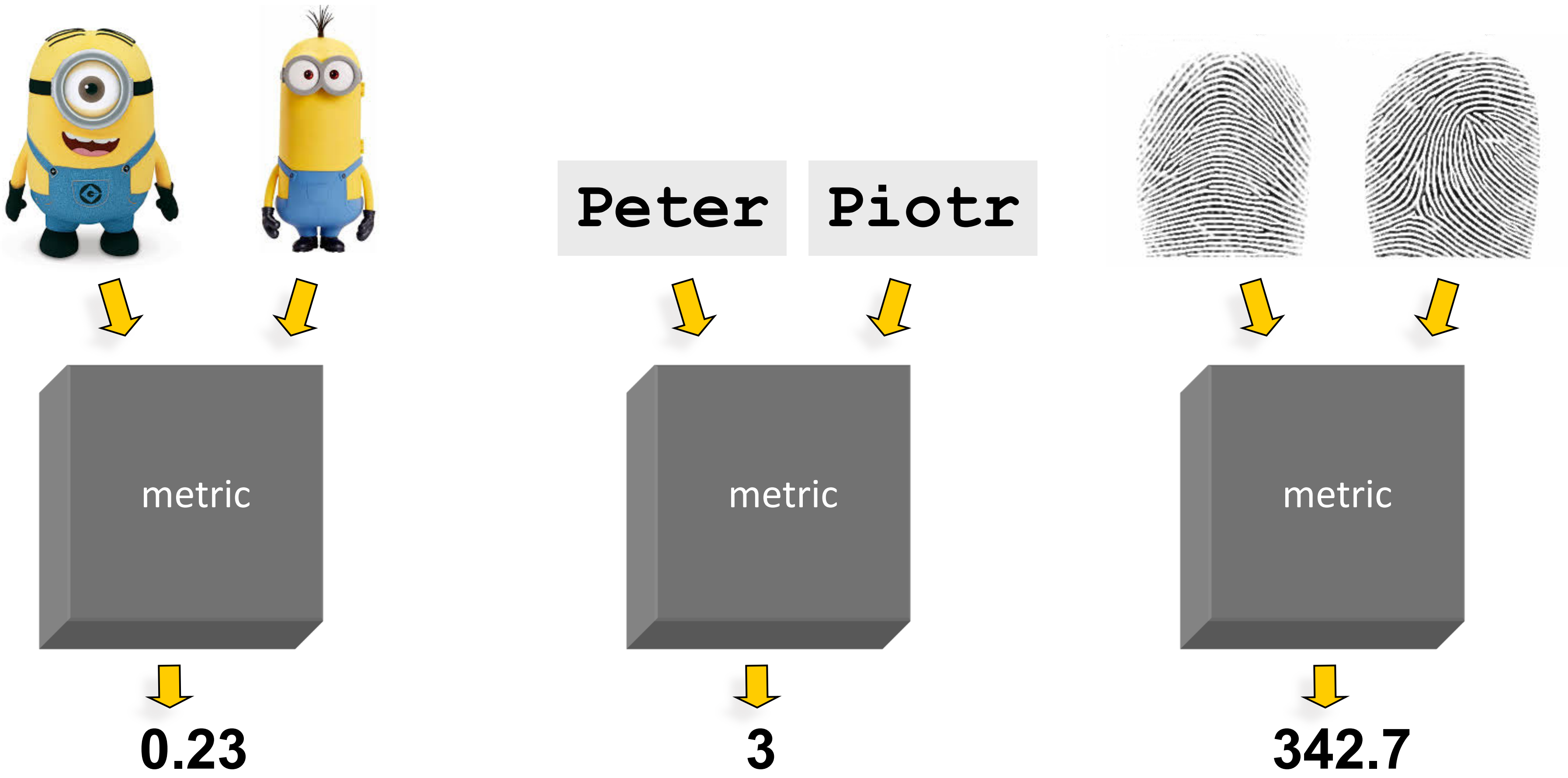
Cluster by features

- Color
- Location
- Texture
- ….



Simpson's Family **School Employees** **Females** **Males**

Distance metrics



Peter Piotr

metric metric metric

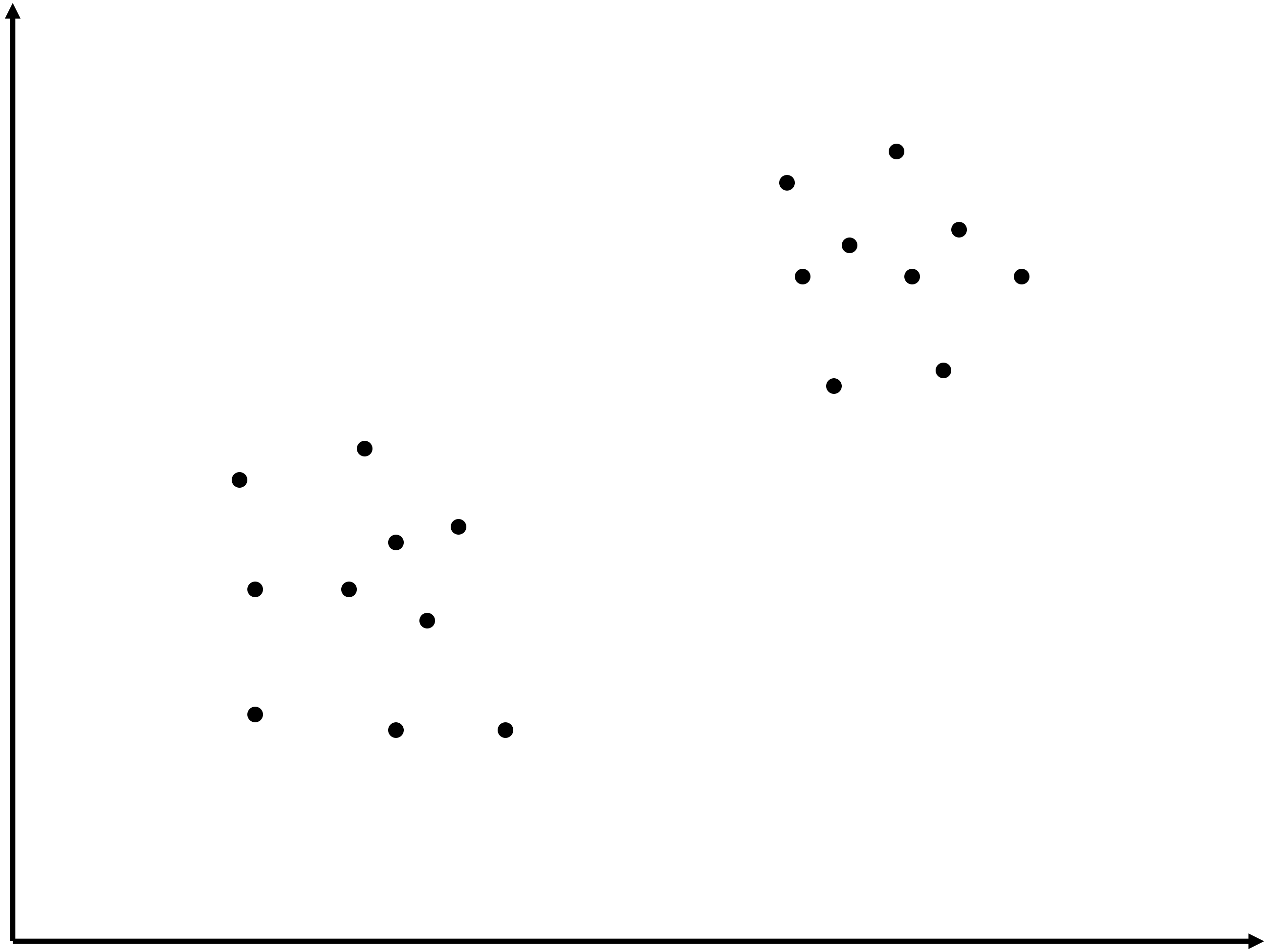
0.23

3

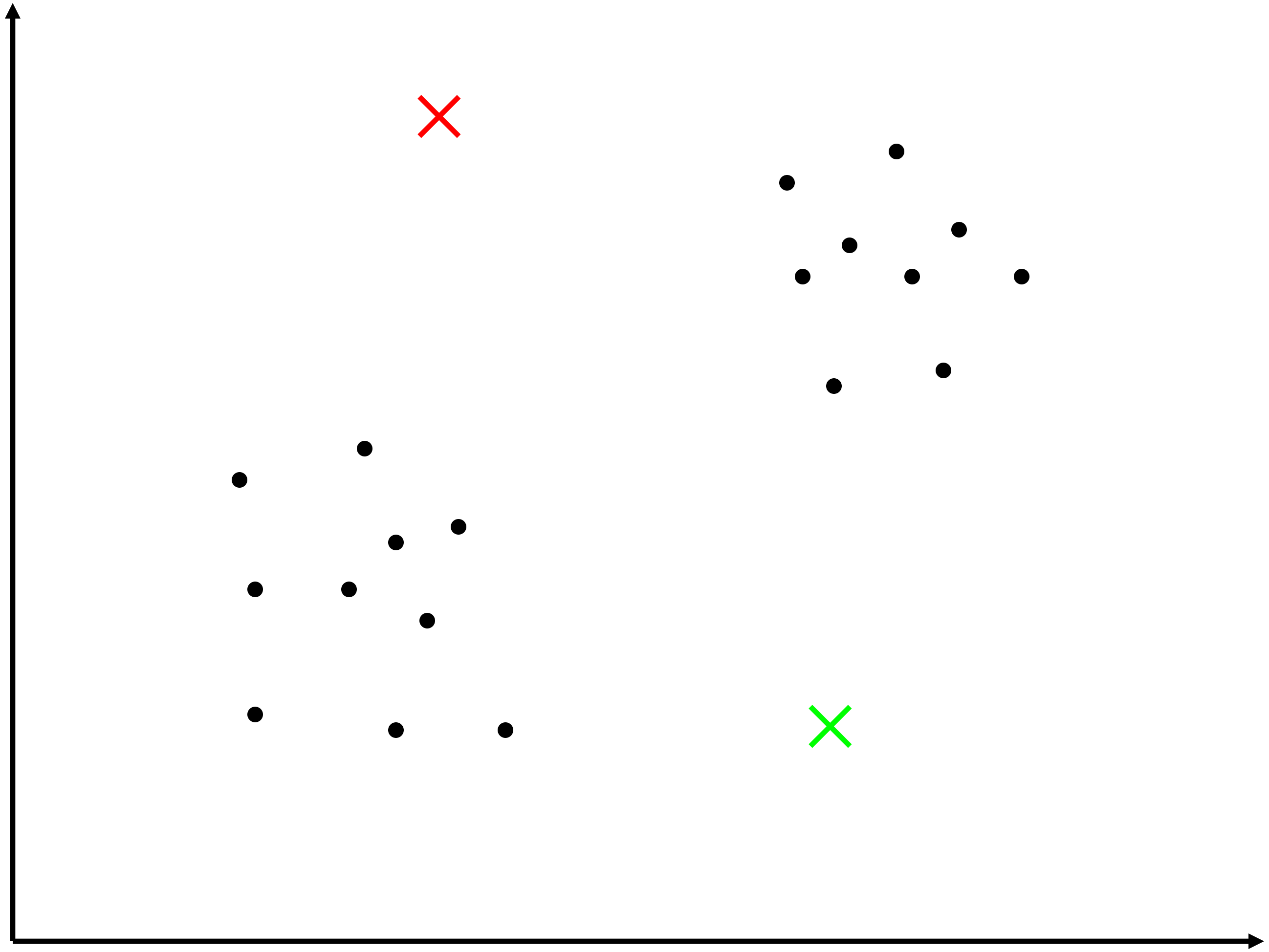
342.7

Question: Can we use any function as a distance metric?

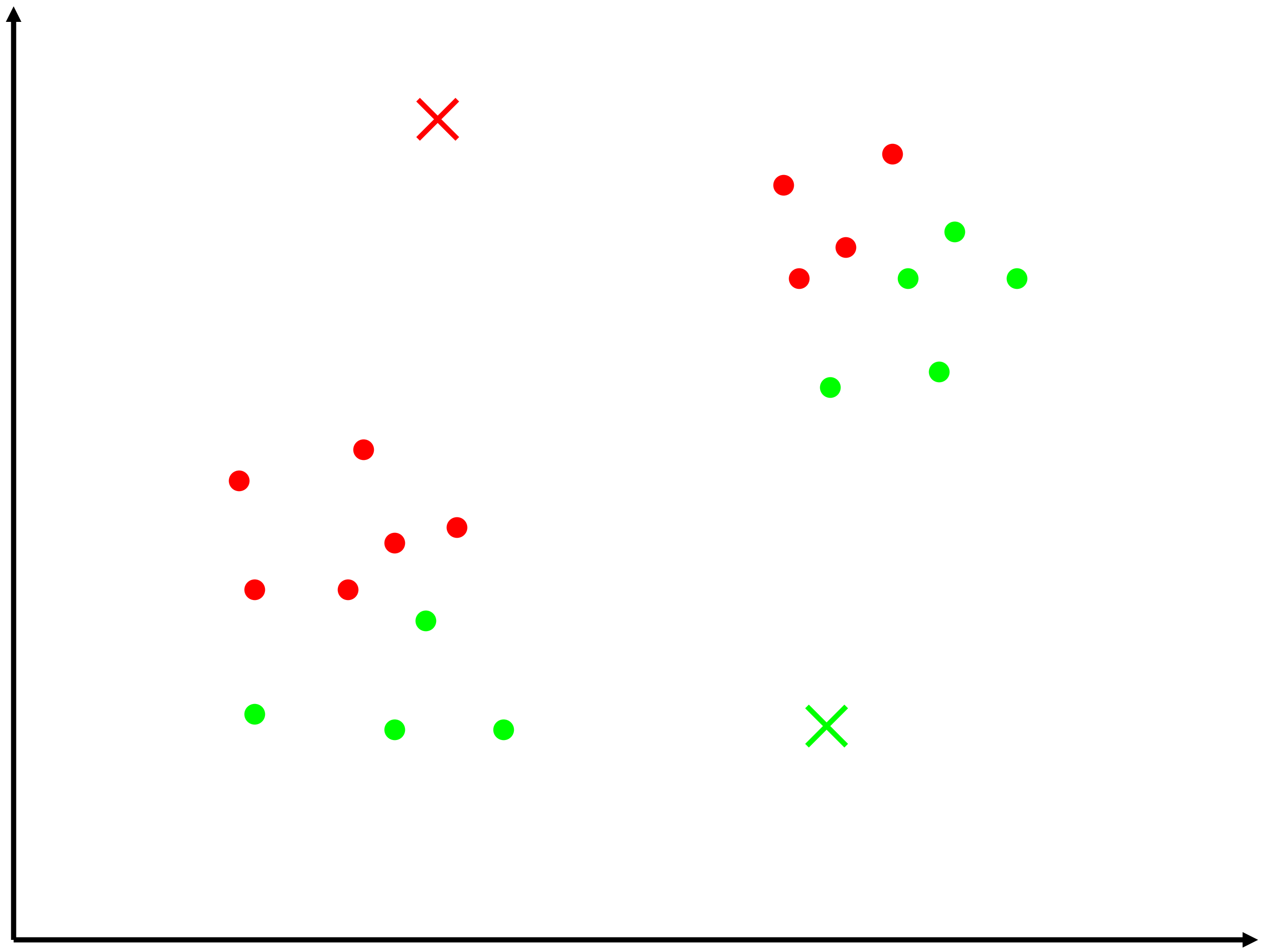
K-means Clustering



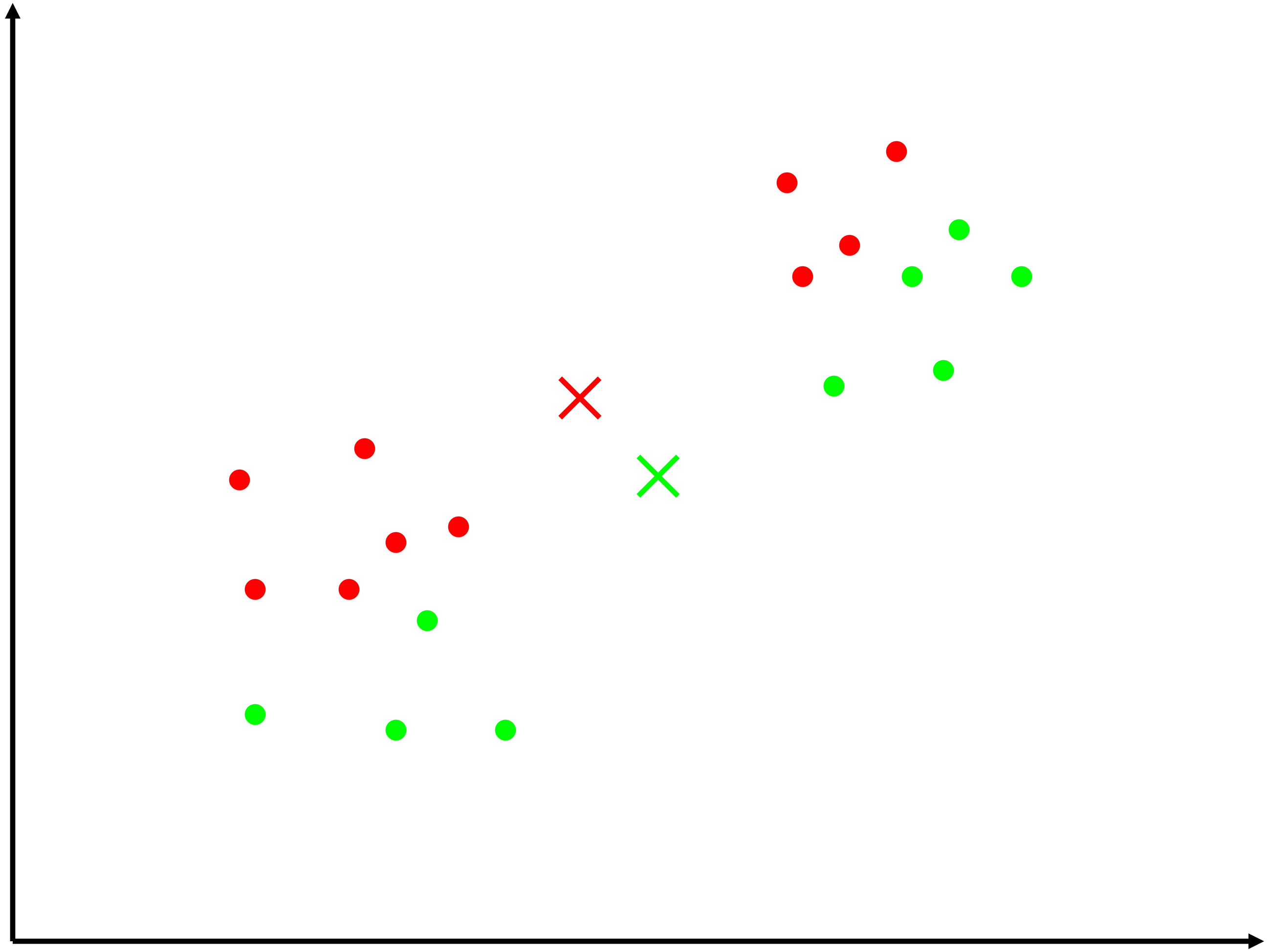
K-means Clustering



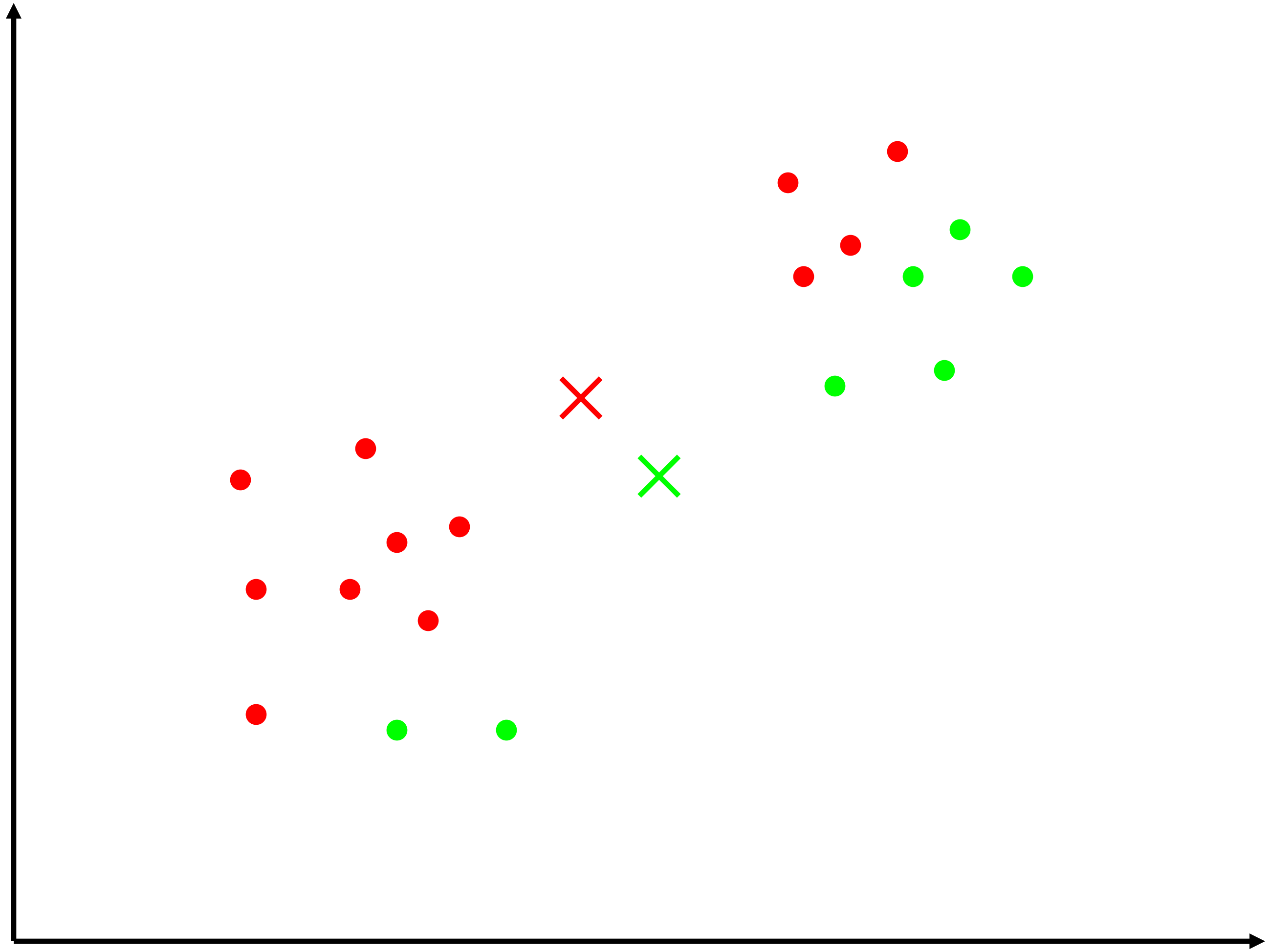
K-means Clustering



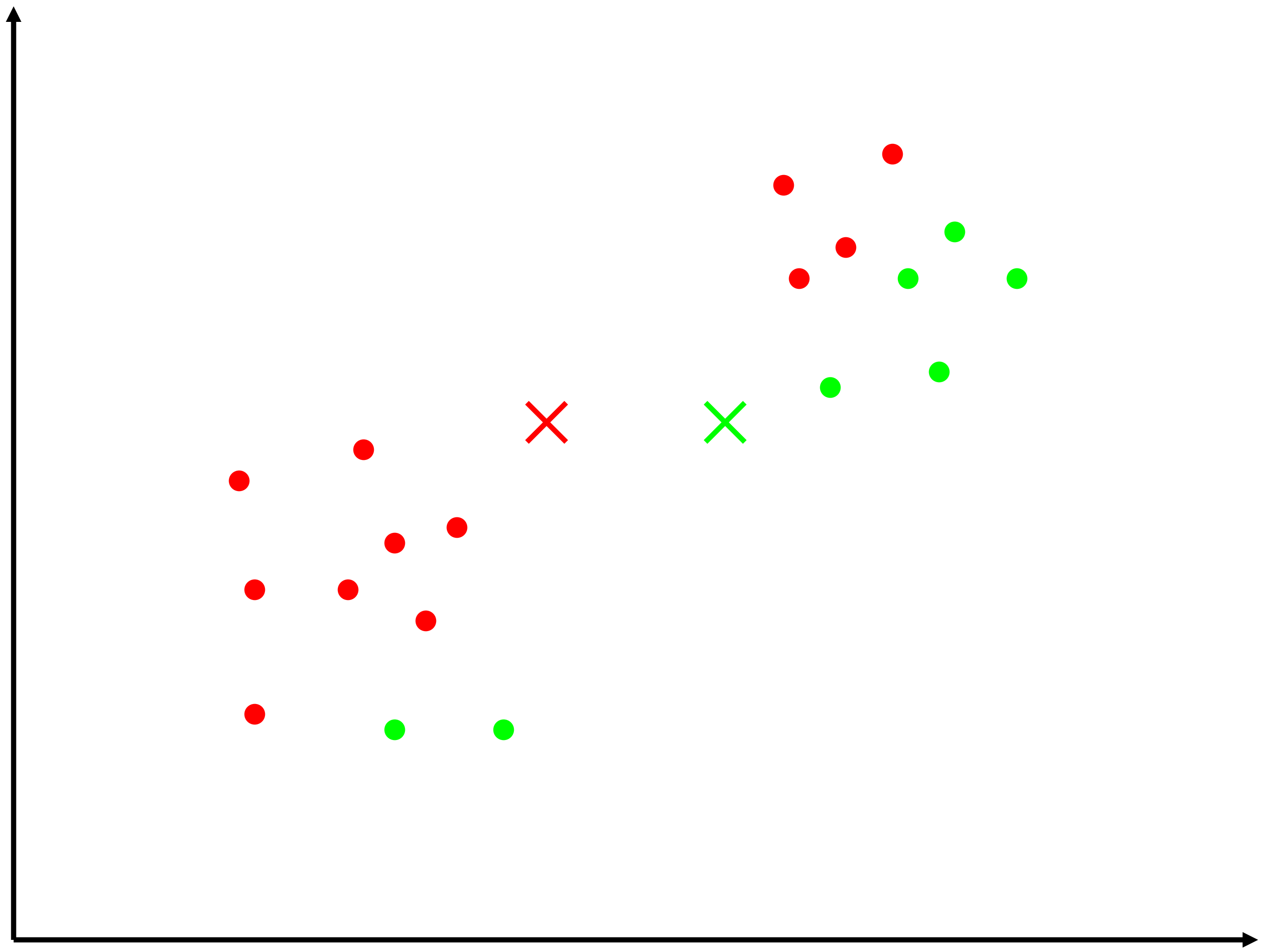
K-means Clustering



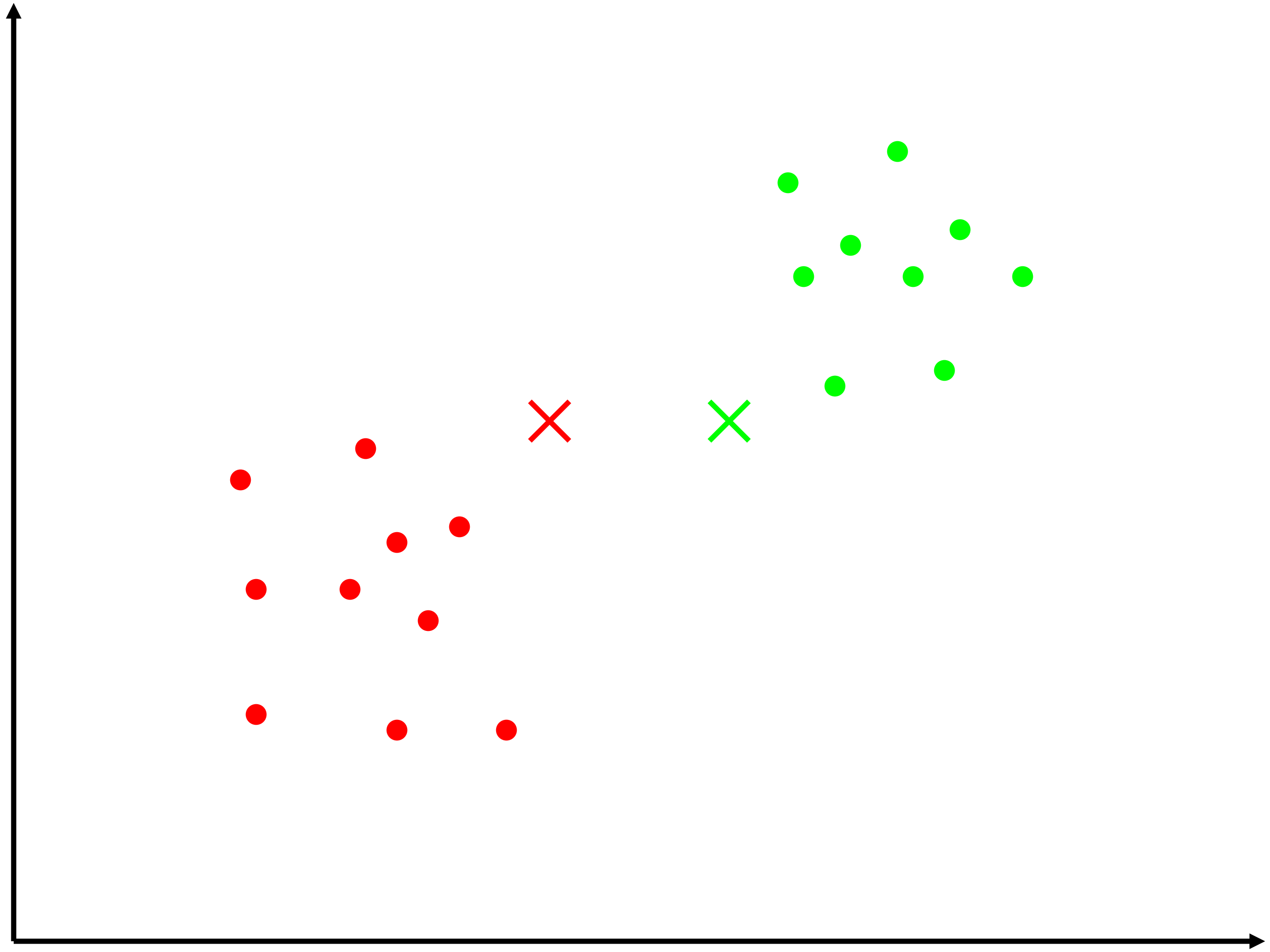
K-means Clustering



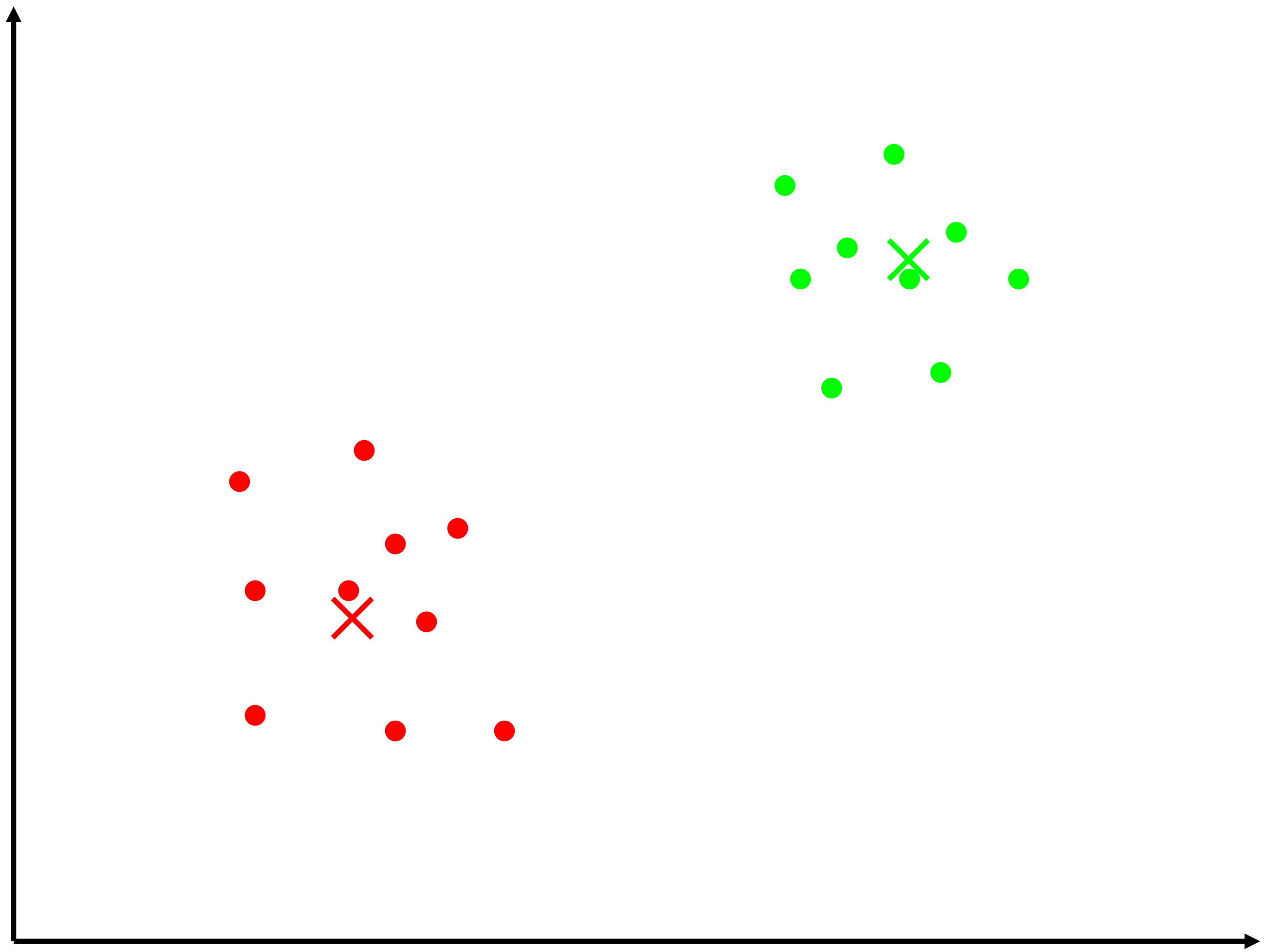
K-means Clustering



K-means Clustering

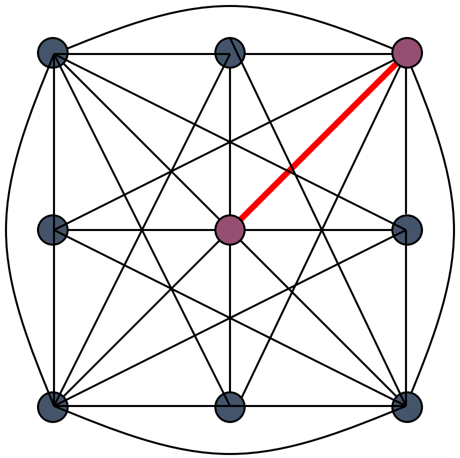


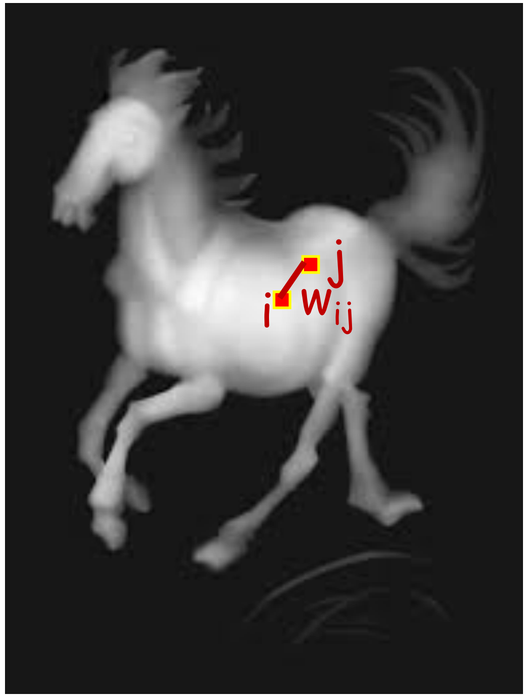
K-means Clustering



- “Optimization” Even in segmentation? ☺

Graph-based segmentation





- **Node (●)** = a pixel
- **Edge** = connectivity between a pair of neighboring pixels
- **Edge weight** (w_{ij}) = the similarity (or dissimilarity) of the pair nodes

Minimize a particular cost over the edges and nodes

Source: http://slazebni.cs.illinois.edu/spring18/

Felzenszwalb & Huttenlocher algorithm

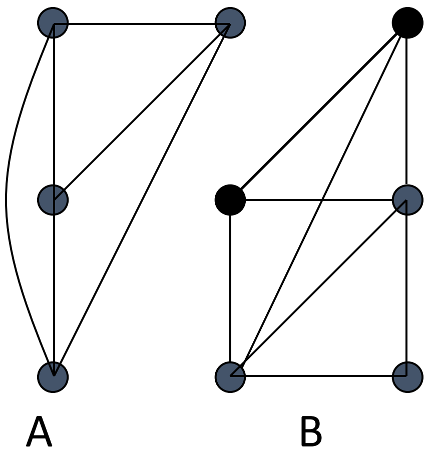
- Graph definition:
  - Vertices are pixels, edges connect neighboring pixels, weights correspond to *dissimilarity* in (x,y,r,g,b) space
- The algorithm:
  - Start with each vertex in its own component
  - For each edge in increasing order of weight:
    - If the edge is between vertices in two different components A and B, merge if the edge weight is lower than the internal dissimilarity within either of the components
    - Threshold is the minimum of the following values, computed on A and B:
      - (Highest-weight edge in minimum spanning tree of the component) + (k / size of component)

Source: http://slazebni.cs.illinois.edu/spring18/

Example results

http://www.cs.brown.edu/~pff/segment/

Segmentation by graph cuts



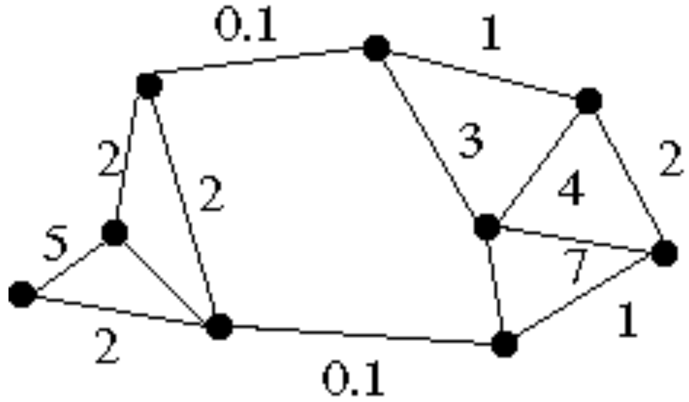
- Break graph into segments
  - Delete links that cross between segments
  - Easiest to break links that have low *affinity*
    - similar pixels should be in the same segments
    - dissimilar pixels should be in different segments

Source: S. Seitz

Slide Source: http://slazebni.cs.illinois.edu/spring18/

Segmentation by graph cuts

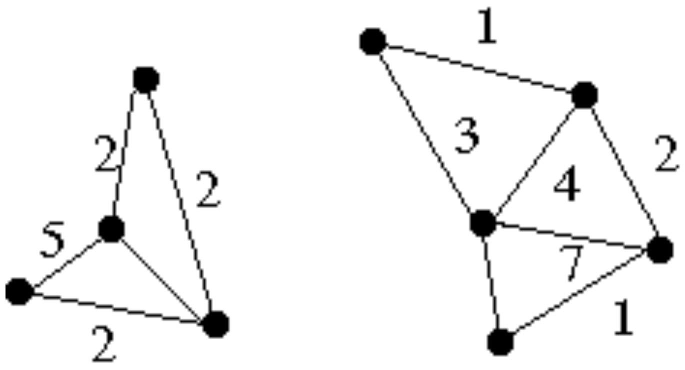
- A graph cut is a set of edges whose removal disconnects the graph
- Cost of a cut: sum of weights of cut edges
- Two-way minimum cuts can be found efficiently



Source: http://slazebni.cs.illinois.edu/spring18/

Segmentation by graph cuts

- A graph cut is a set of edges whose removal disconnects the graph
- Cost of a cut: sum of weights of cut edges
- Two-way minimum cuts can be found efficiently



Source: http://slazebni.cs.illinois.edu/spring18/

Segmentation as labeling

- Suppose we want to segment an image into foreground and background
  - Binary pixel labeling problem
  - Naturally arises in interactive settings

Source: http://slazebni.cs.illinois.edu/spring18/

Segmentation as labeling

- Suppose we want to segment an image into foreground and background
  - Binary pixel labeling problem
  - Naturally arises in interactive settings

User scribbles

Source: http://slazebni.cs.illinois.edu/spring18/

Labeling by energy minimization

- Define a labeling **c** as an assignment of each pixel to a class (foreground or background)

- Find the labeling that minimizes a global energy function:

$$E(\mathbf{c}\,|\,\mathbf{x}) = \sum_{i} \underbrace{f_i(c_i,\mathbf{x})}_{} + \sum_{i,j\in\varepsilon} \underbrace{g_{ij}(c_i,c_j,\mathbf{x})}_{}$$

Pixels

Unary potential (local data term): score for pixel i and label c_i

Neighboring pixels

Pairwise potential (context or smoothing term)

- These are known as Markov Random Field (MRF) or Conditional Random Field (CRF) functions

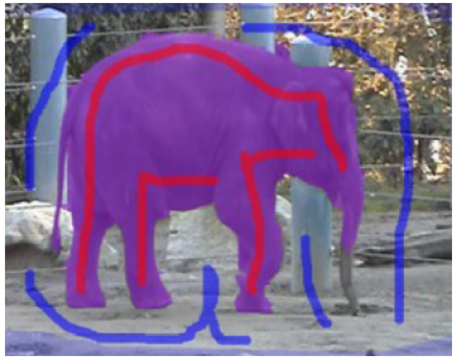
Source: http://slazebni.cs.illinois.edu/spring18/

Segmentation by energy minimization

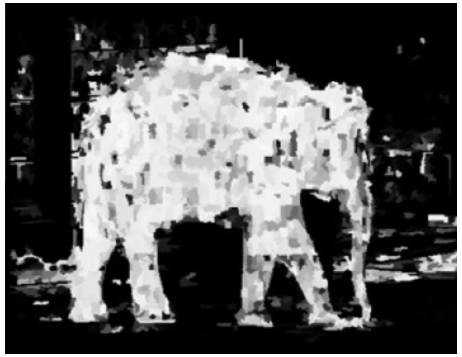
$$E(\mathbf{c} \mid \mathbf{x}) = \sum_i f_i(c_i, \mathbf{x}) + \sum_{i,j \in \varepsilon} g_{ij}(c_i, c_j, \mathbf{x})$$

- Unary potentials: $f_i(c, \mathbf{x}) = -\log P(c \mid \mathbf{x}_i)$
  - Cost is infinity if label does not match the user scribble
  - Otherwise, it is computed based on a color model of user-labeled pixels

User scribbles



P(foreground | x_i)



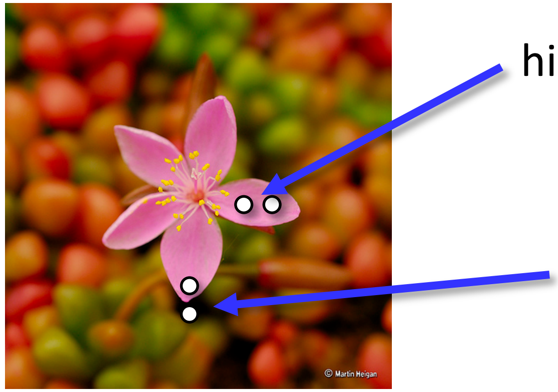
Source: http://slazebni.cs.illinois.edu/spring18/

Segmentation by energy minimization

$$E(\mathbf{c} \mid \mathbf{x}) = \sum_i f_i(c_i, \mathbf{x}) + \sum_{i,j \in \varepsilon} g_{ij}(c_i, c_j, \mathbf{x})$$

- Unary potentials: $f_i(c, \mathbf{x}) = -\log P(c \mid \mathbf{x}_i)$
- Pairwise potentials: $g_{ij}(c, c', \mathbf{x}) = w_{ij} \left| c - c' \right|$
  - Neighboring pixels should have the same label unless they look very different

Affinity between pixels i and j



high affinity (similarity)

low affinity

Source: http://slazebni.cs.illinois.edu/spring18/

Slide Credits and References

- R.Klette, Concise Computer Vision, 2014.
- Y. Ukrainitz & B. Sarel, Weizmann
- Thu Huong Nhuyen, TU Dresden
- Jens N. Kaftan and Andr`e A. Bell and Til Aach . Mean Shift Segmentation Evaluation of Optimization Techniques
- Cheng, Y. (1995). Mean Shift, Mode Seeking, and Clustering. IEEE Transactions on Pattern Analysis and Machine Intelligence, 17(8):790–799.
- KEINOSUKE FUKUNAGA, AND LARRY D. HOSTETLER. The Estimation of the Gradient of a Density Function, with Applications in Pattern – Recognition. IEEE TRANSACTIONS ON INFORMATION THEORY, VOL. IT-21, NO. 1, JANUARY 1975
- Comaniciu, D. and Meer, P. (1997). Robust Analysis of Feature Spaces: Color Image Segmentation.
- In IEEE Conference on Computer Vision and Pattern Recognition. CVPR 1997, pages 750–755.
- Comaniciu, D. and Meer, P. (1999). Mean Shift Analysis an Applications. In International Conference on Computer Vision. ICCV 1999, volume 2, pages 1197–1203.
- Comaniciu, D. and Meer, P. (2002). Mean Shift: A Robust Approach Toward Feature Space Analysis. IEEE Transactions on Pattern Analysis and Machine Intelligence, 24(5):603–619.
- K. Fukunaga, L.D. Hostetler, The estimation of the gradient of a density function, with applications in pattern recognition, IEEE Trans. Inf. Theory 21 (1975) 3240.
- Yiping Hong , Jianqiang Yi and Dongbin Zhao . Improved mean shift segmentation approach for natural images

- These lecture also includes content from Ulas Bagci and Svetlana Lazebnik .