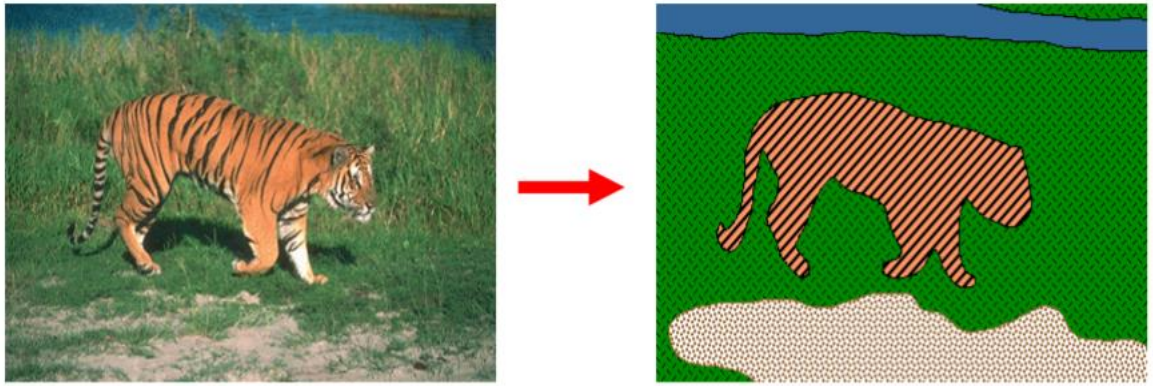
Computer Vision: Segmentation

Image Segmentation



Aim: To partition an image into a collection of pixels that go together

- – Meaningful regions (coherent objects)
- – Linear structures (line, curve, …)
- – Shapes (circles, ellipses, …)

Basically, image segmentation partitions an image into regions called segments.

Image Segmentation

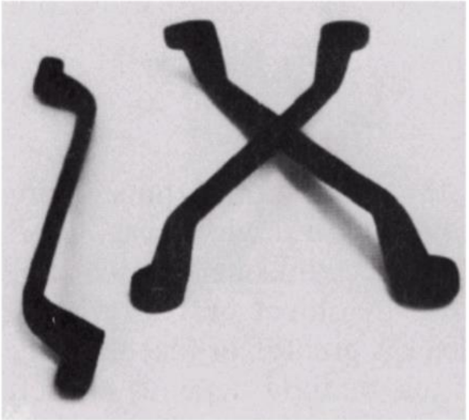
Goal:

find coherent "blobs" or specific "objects"

lower level tasks
(e.g. "superpixels")

large grey area
in-between

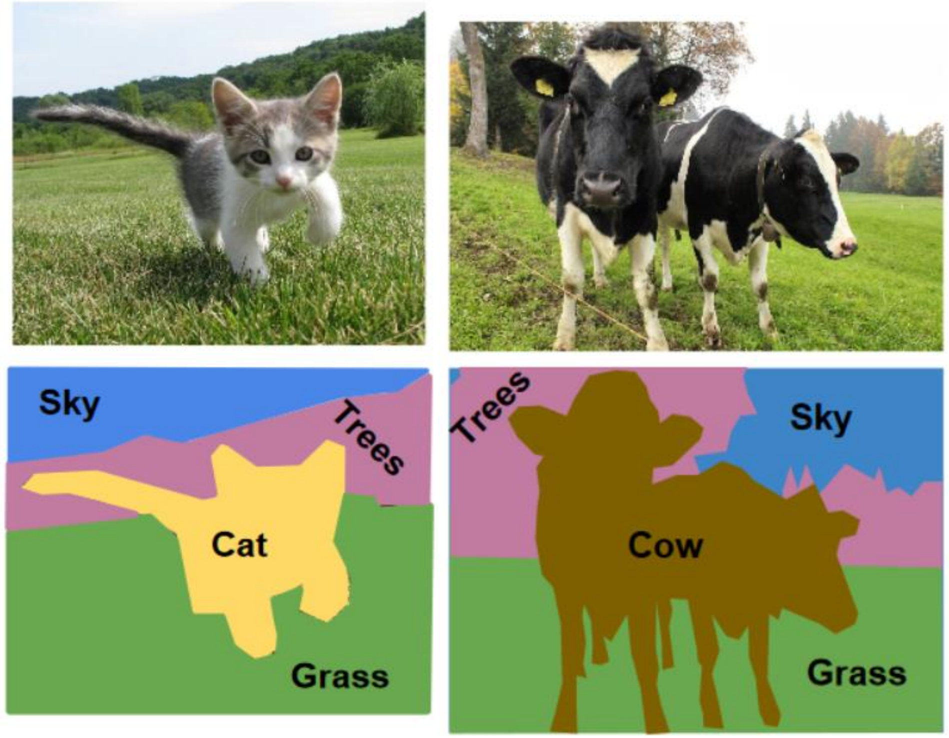
higher level tasks
(e.g. cars, humans, or organs)

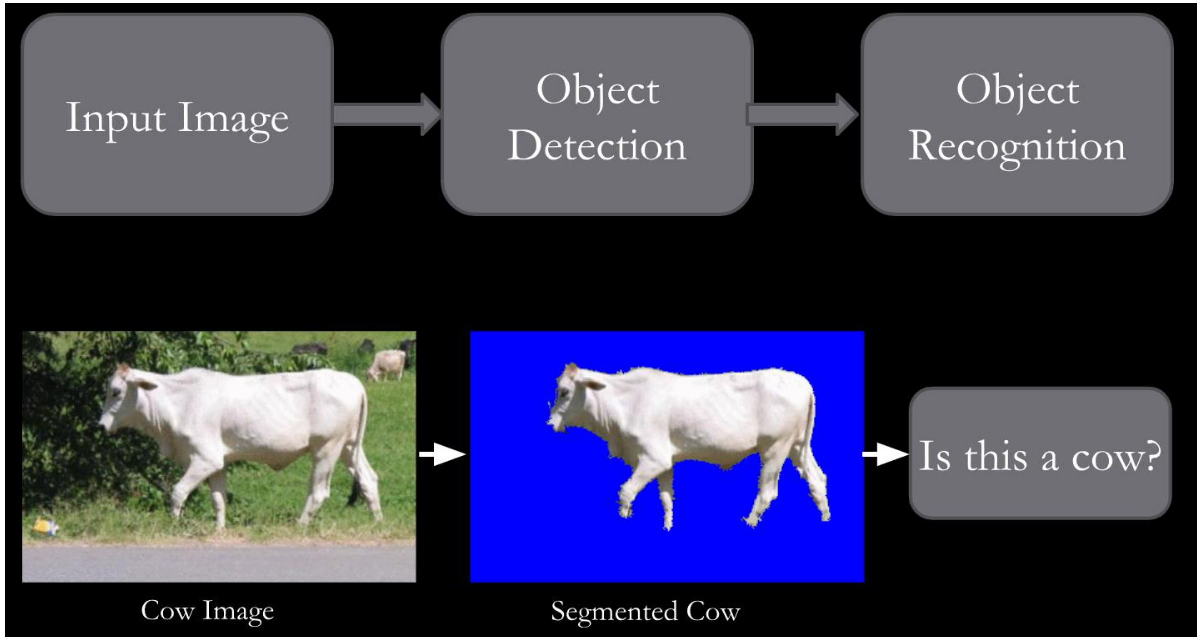


The tools become blobs

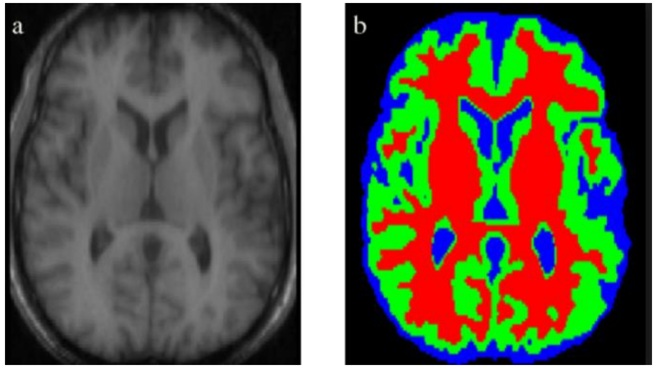
The house, grass, and sky make different blobs

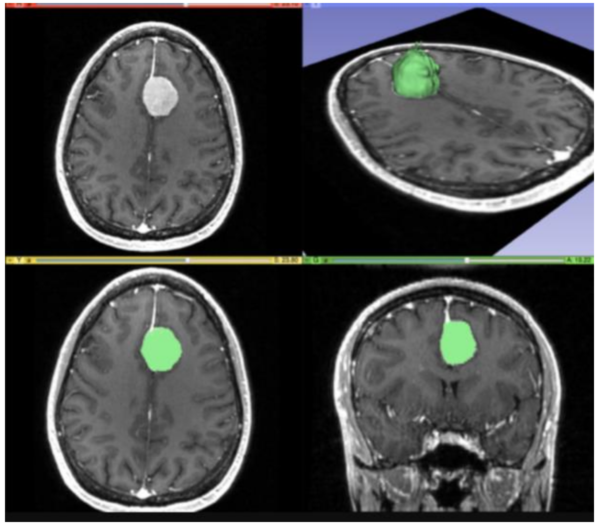
Applications: Object Classification

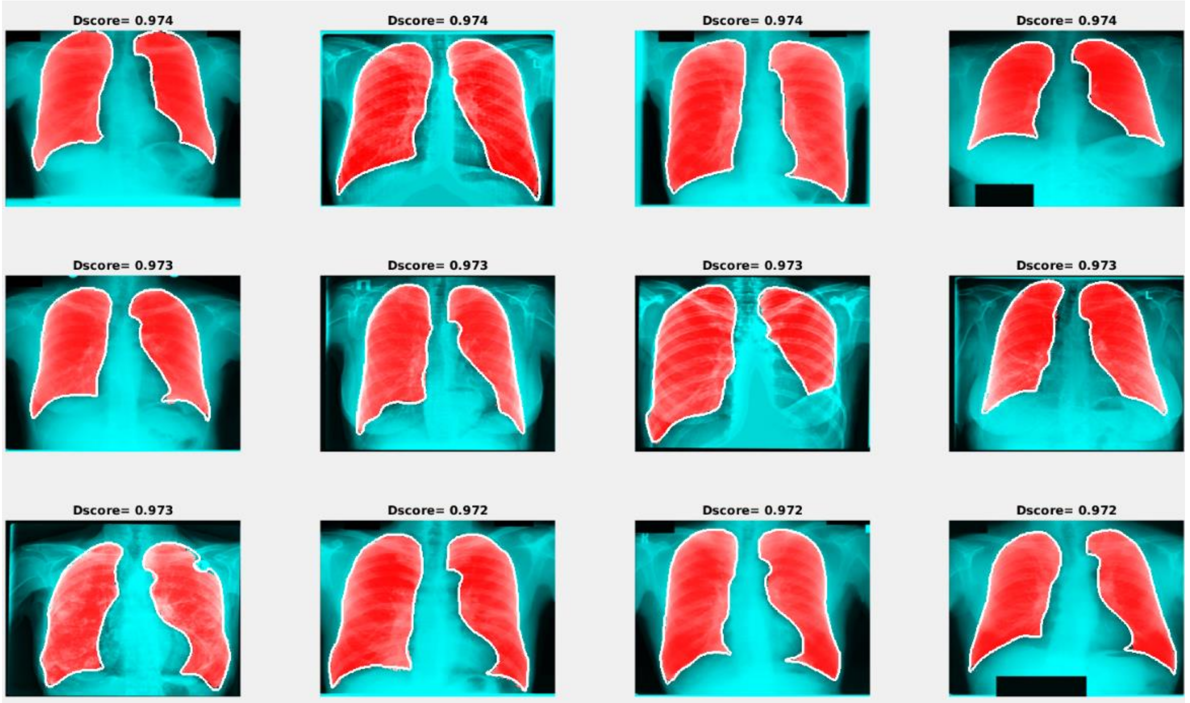




Applications: Medical Imaging

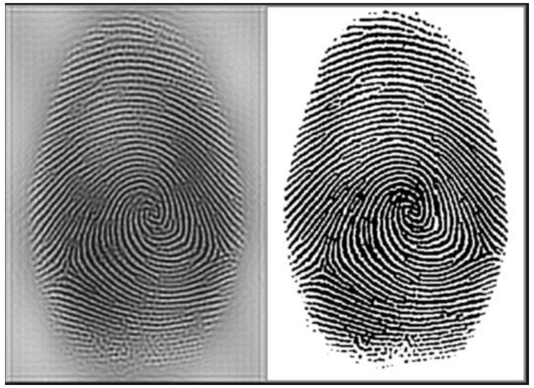






Applications: Biometrics

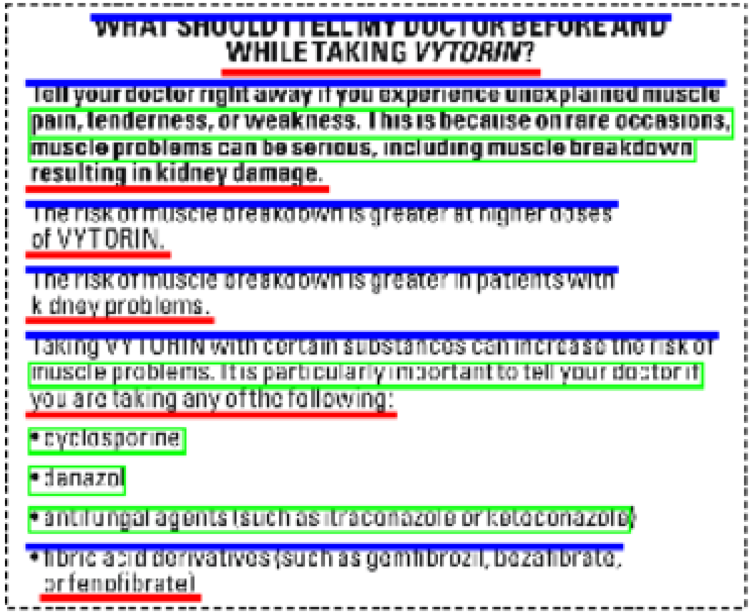
Skin color based segmentation

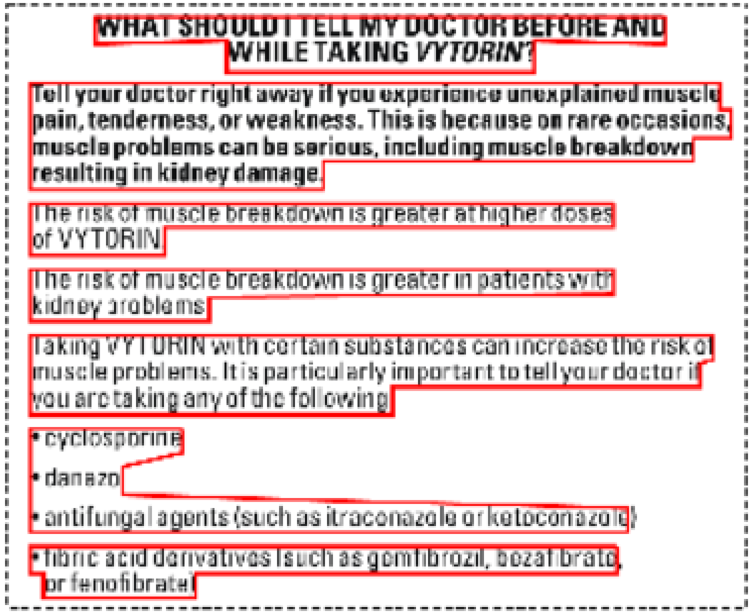


Fingerprint segmentation

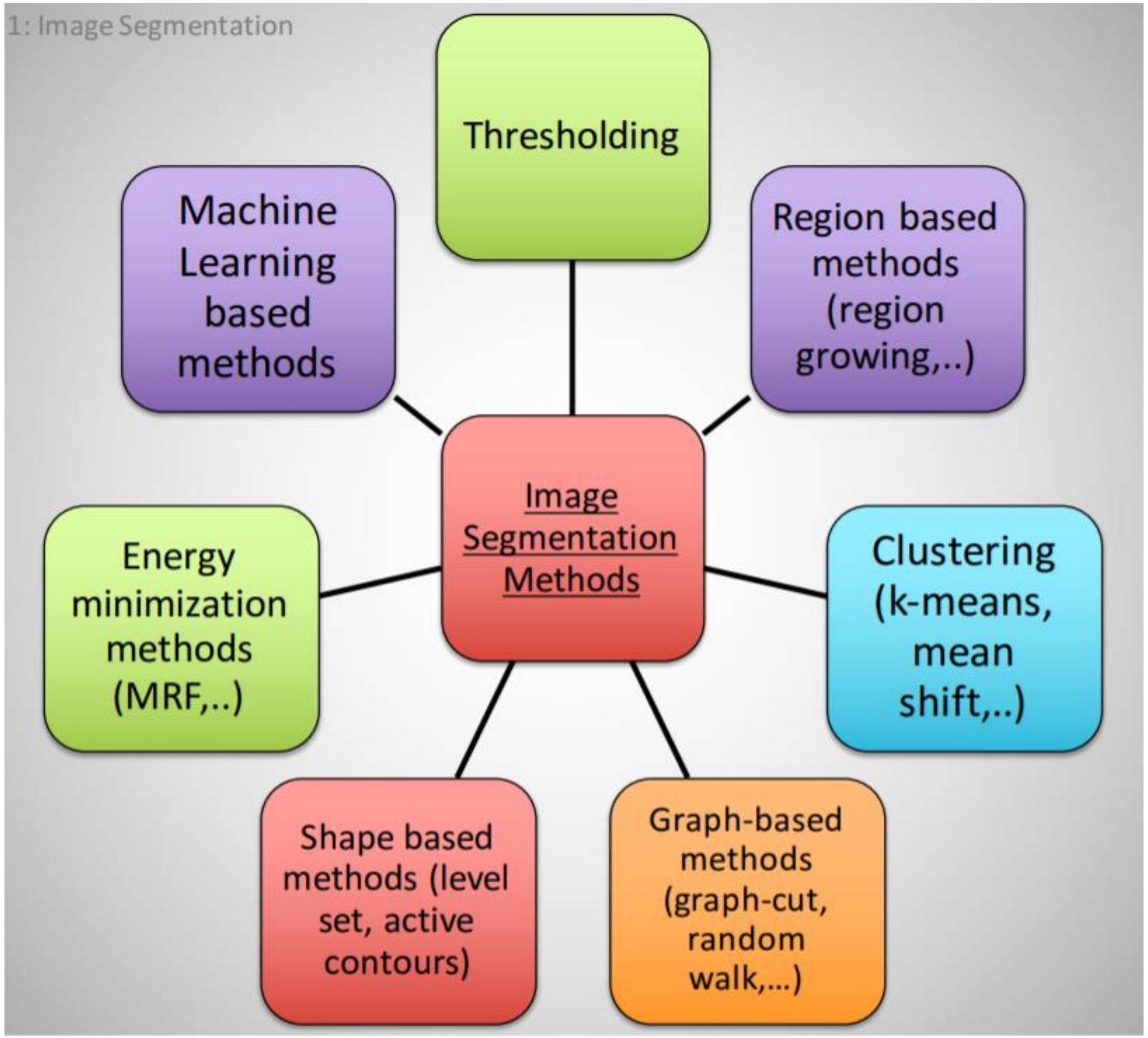
Face Detection

Document Segmentation





Any ideas on how can we perform segmentation?


1: Image Segmentation
Thresholding
Machine Learning based methods
Region based methods (region growing,..)
Image Segmentation Methods
Energy minimization methods (MRF,..)
Clustering (k-means, mean shift,..)
Shape based methods (level set, active contours)
Graph-based methods (graph-cut, random walk,...)

Thresholding

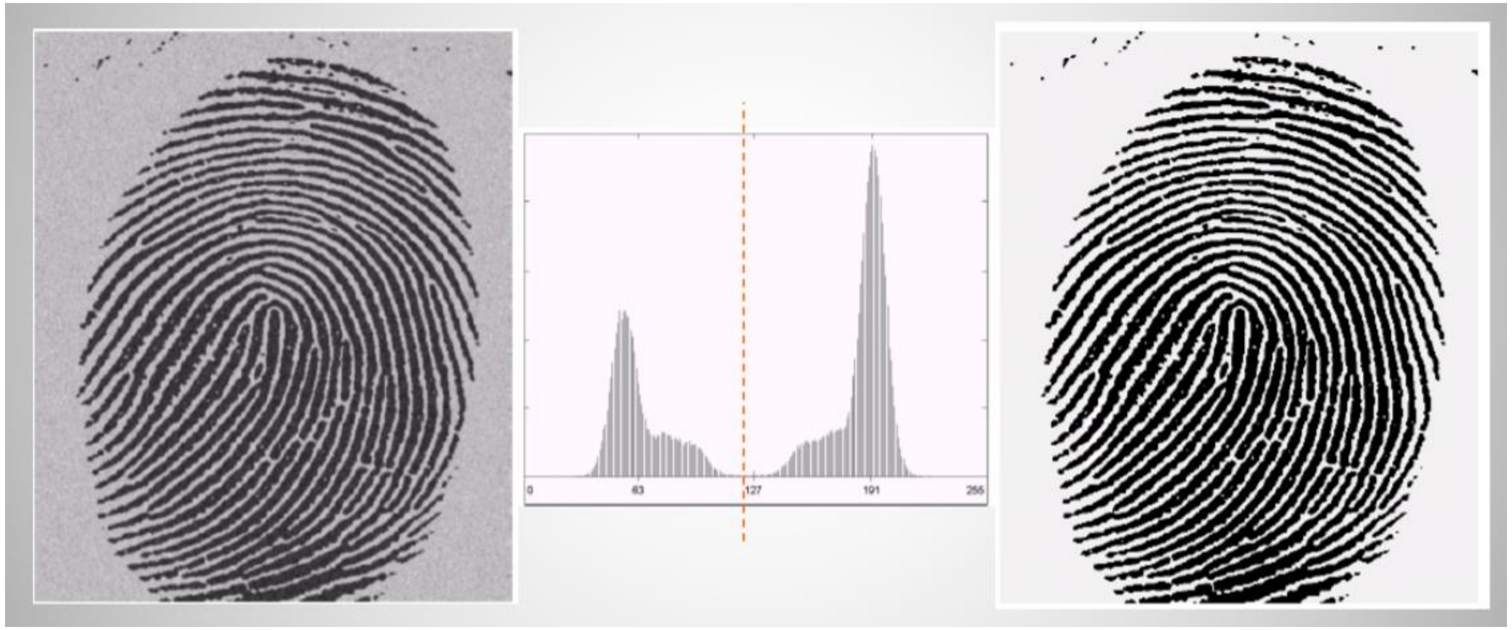
- Image binarization applies often just one global threshold T for mapping a scalar image I into a binary image

Thresholding

- Image binarization applies often just one global threshold T for mapping a scalar image I into a binary image

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

Thresholding



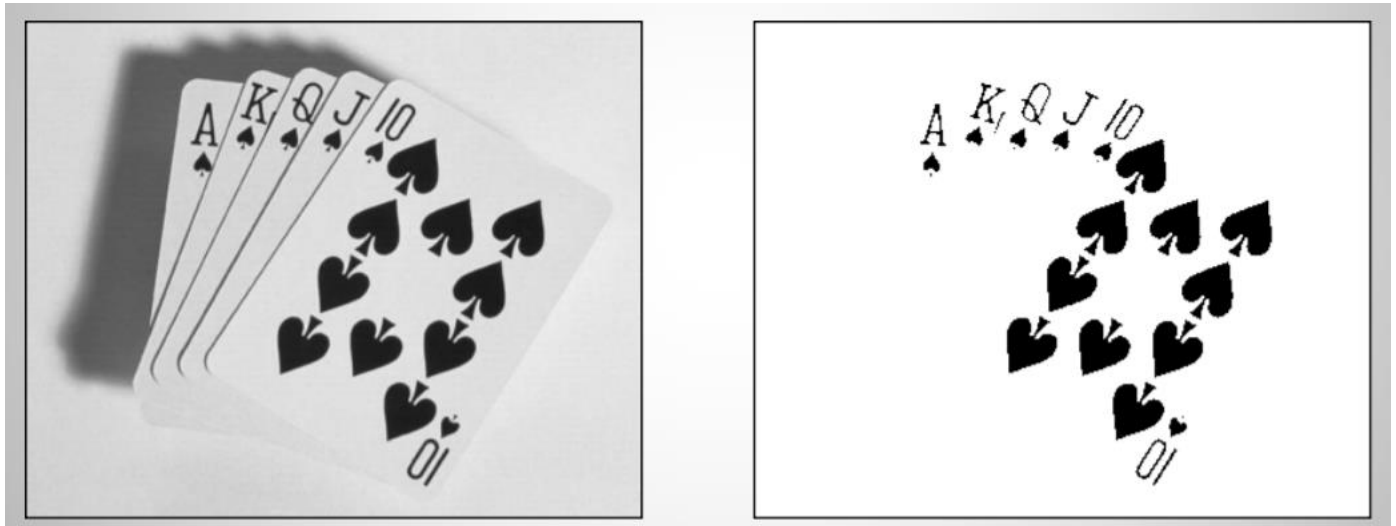
Thresholding

- Image binarization applies often just one global threshold T for mapping a scalar image I into a binary image

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

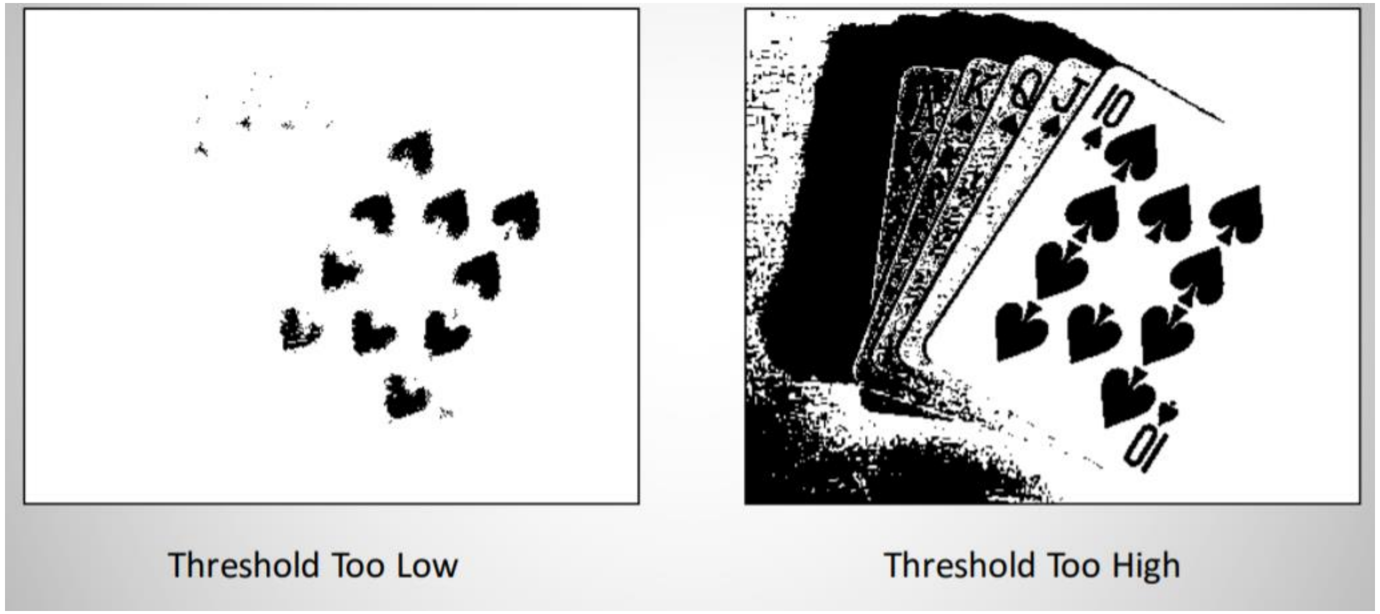
- The global threshold can be identified by an optimization strategy aiming at creating “large” connected regions and at reducing the number of small-sized regions, called *artifacts*.

Thresholding



Original Image Thresholded Image

Thresholding



Otsu Thresholding

- 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)
- **Minimize within class variance** of foreground and background pixels!
- Iteratively check for each threshold and select the one with the least within class variance!

Otsu Thresholding

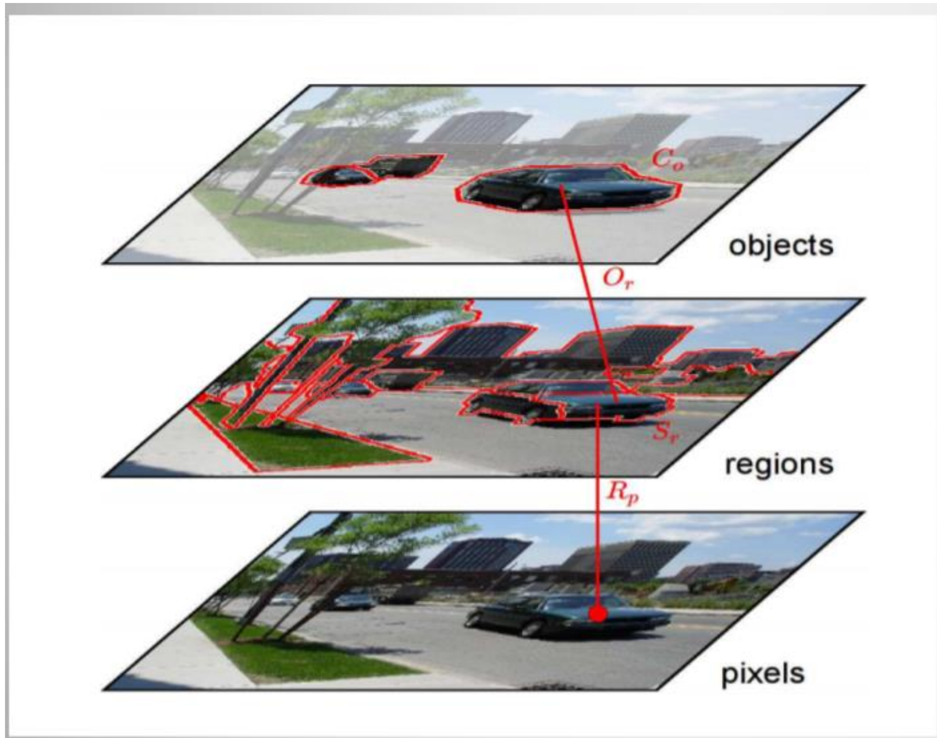
- The method uses the grey-value histogram of the given image I as input and aims at providing the best threshold in the sense that the "overlap" between two classes, set of object and background pixels, is minimized (i.e., by finding the best balance)
- **Minimize within class variance** of foreground and background pixels!
- Iteratively check for each threshold and select the one with the least within class variance!

$$\text{Within Class Variance} \quad \sigma_W^2 = W_b\,\sigma_b^2 + W_f\,\sigma_f^2$$

$W_{b/f}$ = Probability of Class b/f

σ_{i} = Variance of class b/f

Region based Segmentation



Region:
A group of connected pixels with similar properties

Closed boundaries

Computation of regions is based on similarity

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

Spatial proximity + similarity

Region based Segmentation: Seeded Segmentation

- 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: Seeded Segmentation

- 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;

Region growing: Seeded Segmentation

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)

Region growing: Seeded Segmentation

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)

Region growing: Seeded Segmentation

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

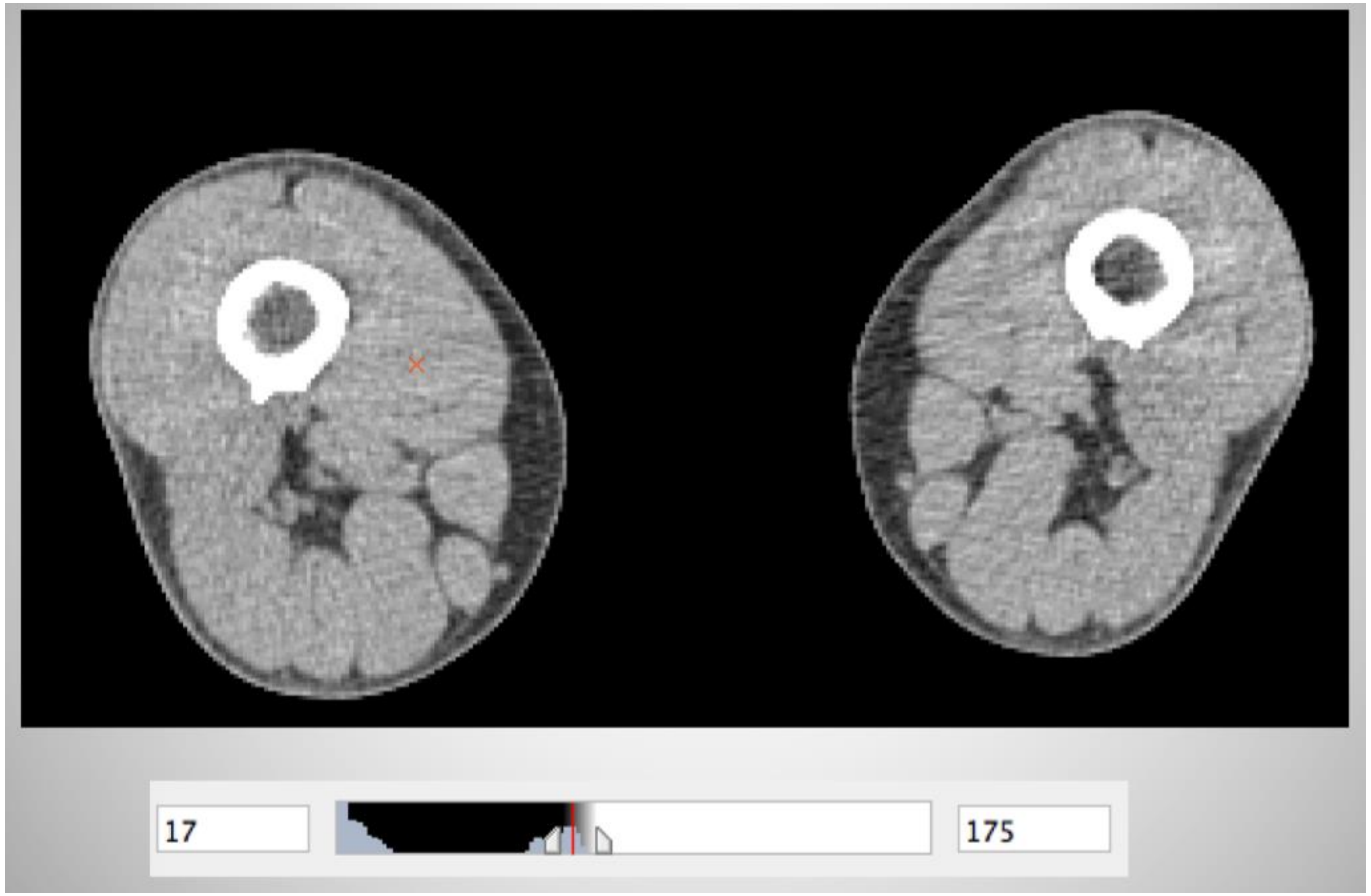
0	0	5	6	7
1	1	5	8	7
0	1	6	7	7
2	0	7	6	6
0	1	5	6	5

(a)

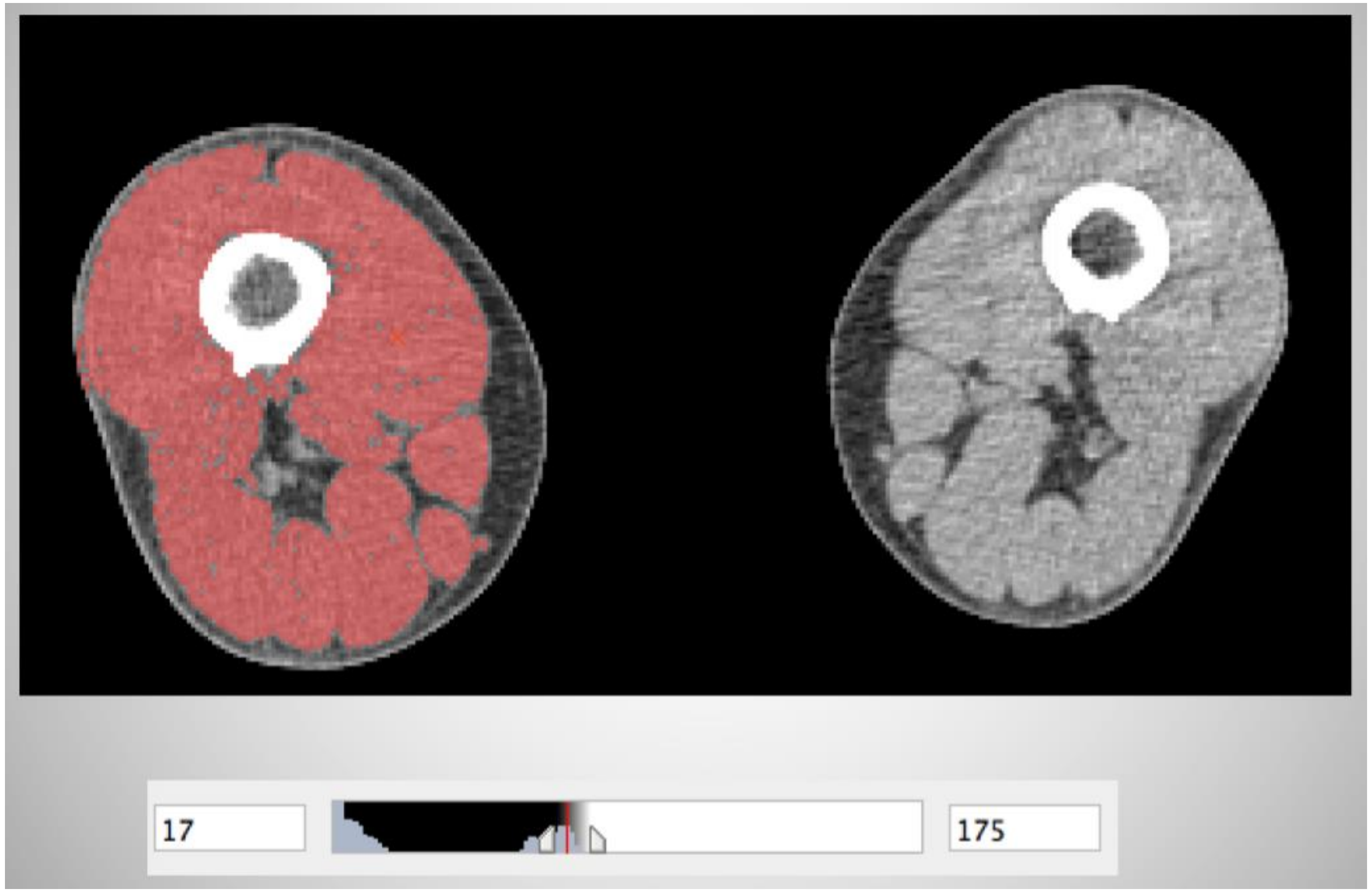
a	a	b	b	b
a	a	b	b	b
a	a	b	b	b
a	a	b	b	b
a	a	b	b	b

(b)

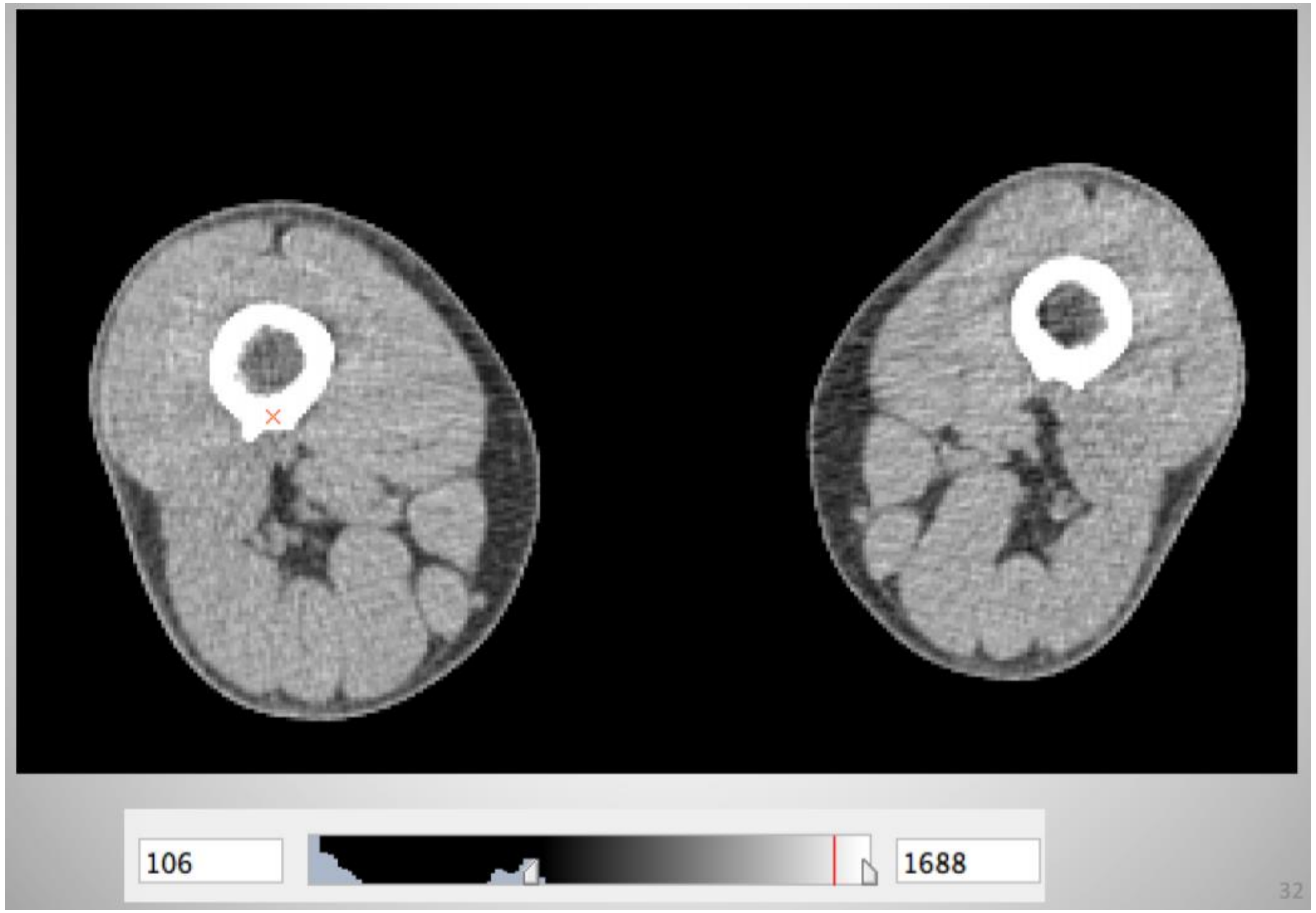
Region growing: Example



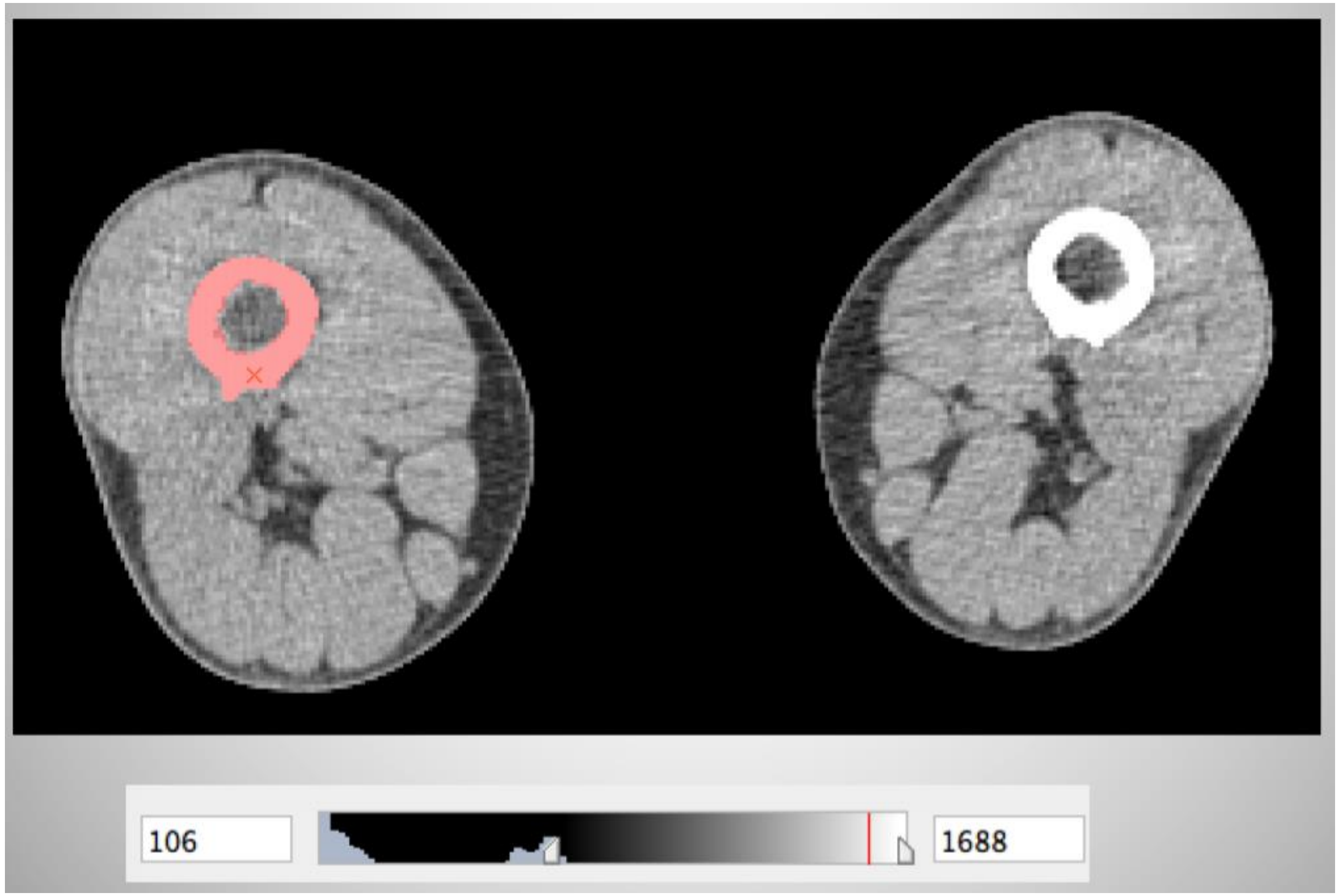
Region growing: Example



Region growing: Example



Region growing: Example

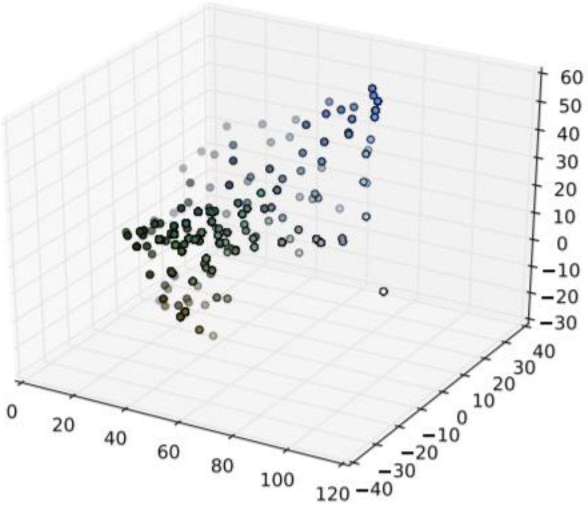


Clustering based Segmentation

- Organizing data into classes such that:
  - – High intra-class similarity
  - – Low inter-class similarity

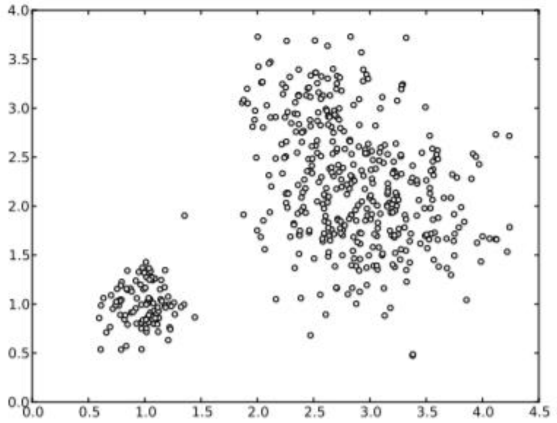
• Finding the class labels and the number of classes directly from the data
(as opposed to classification tasks)

Clustering based Segmentation



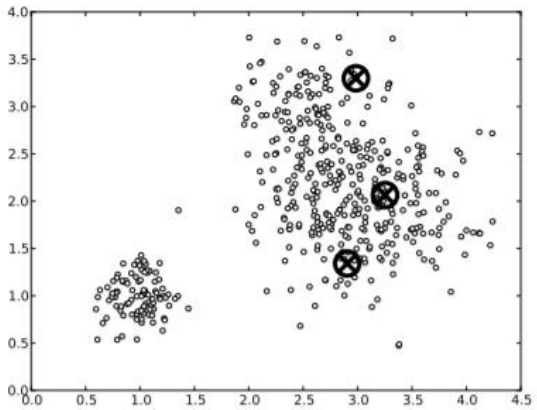
- Pixels are points in a high dimensional space
  - color: 3d
  - color+location:5d
- Cluster pixels into segment

Clustering based Segmentation: k-means



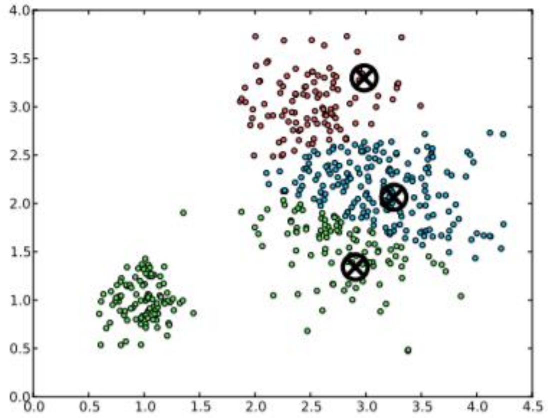
1. Randomly initialize K cluster centers, $c_1, \ldots, c_k$
2. Given cluster centers, determine points in each cluster
   - For each point p, find the closest c_i. Put p into cluster i.
3. Given points in each cluster, solve for c_i
   - Set c_i to be the mean of points in cluster i
4. If c_i have changed, repeat Step 2

Clustering based Segmentation: k-means



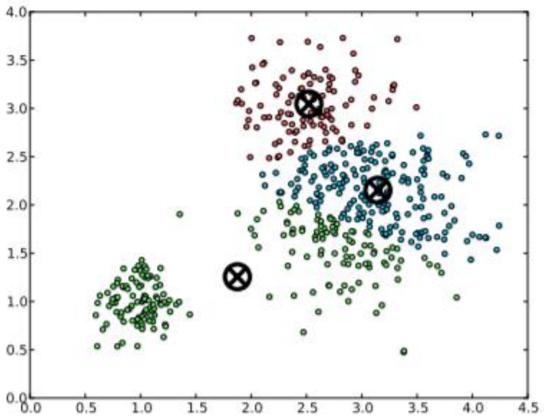
1. Randomly initialize K cluster centers, $c_1, \ldots, c_k$
2. Given cluster centers, determine points in each cluster
   - For each point p, find the closest c_i. Put p into cluster i.
3. Given points in each cluster, solve for c_i
   - Set c_i to be the mean of points in cluster i
4. If c_i have changed, repeat Step 2

Clustering based Segmentation: k-means



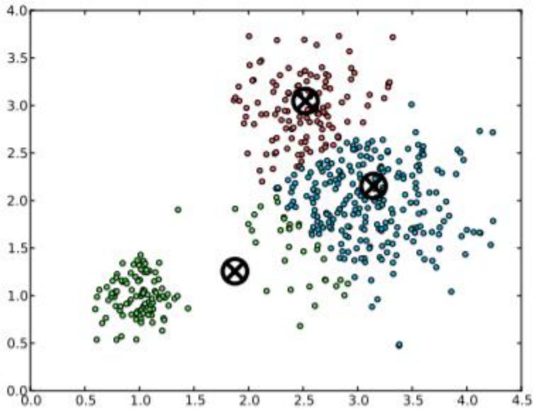
1. Randomly initialize K cluster centers, $c_1, \ldots, c_k$
2. Given cluster centers, determine points in each cluster
   - For each point p, find the closest c_i. Put p into cluster i.
3. Given points in each cluster, solve for c_i
   - Set c_i to be the mean of points in cluster i
4. If c_i have changed, repeat Step 2

Clustering based Segmentation: k-means



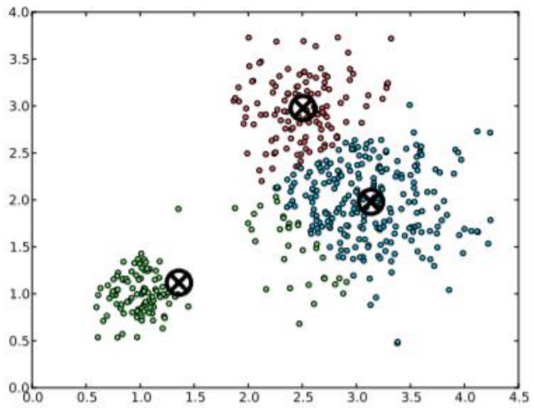
1. Randomly initialize K cluster centers, $c_1, \ldots, c_k$
2. Given cluster centers, determine points in each cluster
   - For each point p, find the closest c_i. Put p into cluster i.
3. Given points in each cluster, solve for c_i
   - Set c_i to be the mean of points in cluster i
4. If c_i have changed, repeat Step 2

Clustering based Segmentation: k-means



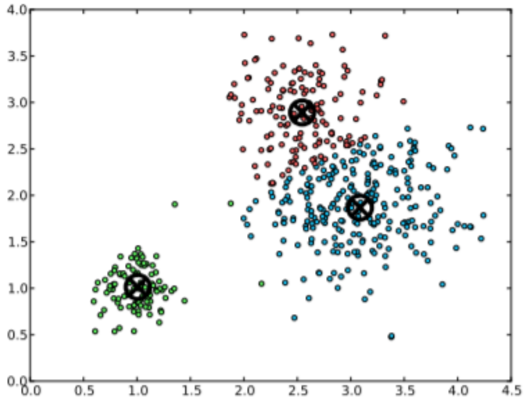
1. Randomly initialize K cluster centers, $c_1, \ldots, c_k$
2. Given cluster centers, determine points in each cluster
   - For each point p, find the closest c_i. Put p into cluster i.
3. Given points in each cluster, solve for c_i
   - Set c_i to be the mean of points in cluster i
4. If c_i have changed, repeat Step 2

Clustering based Segmentation: k-means



1. Randomly initialize K cluster centers, $c_1, \ldots, c_k$
2. Given cluster centers, determine points in each cluster
   - For each point p, find the closest c_i. Put p into cluster i.
3. Given points in each cluster, solve for c_i
   - Set c_i to be the mean of points in cluster i
4. If c_i have changed, repeat Step 2

Clustering based Segmentation: k-means



1. Randomly initialize K cluster centers, $c_1, \ldots, c_k$
2. Given cluster centers, determine points in each cluster
   - For each point p, find the closest c_i. Put p into cluster i.
3. Given points in each cluster, solve for c_i
   - Set c_i to be the mean of points in cluster i
4. If c_i have changed, repeat Step 2

Clustering based Segmentation: k-means

Clustering based Segmentation: Mean Shift

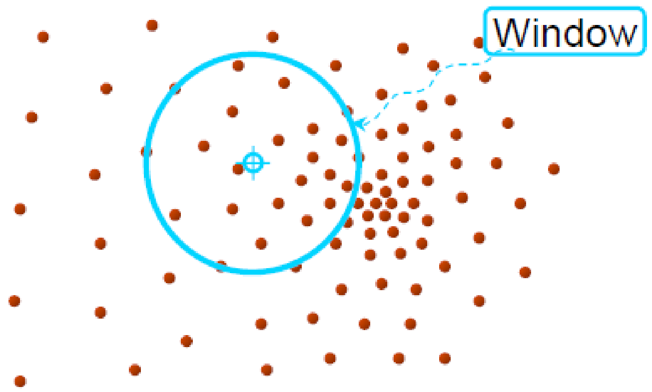
- Challenge of K-means- finding the right K!

- The mean shift clustering algorithm seeks the modes or **local maximums** of density of a given distribution

Mean Shift Clustering

The mean shift algorithm seeks the *modes* or local maximums of density of a given distribution

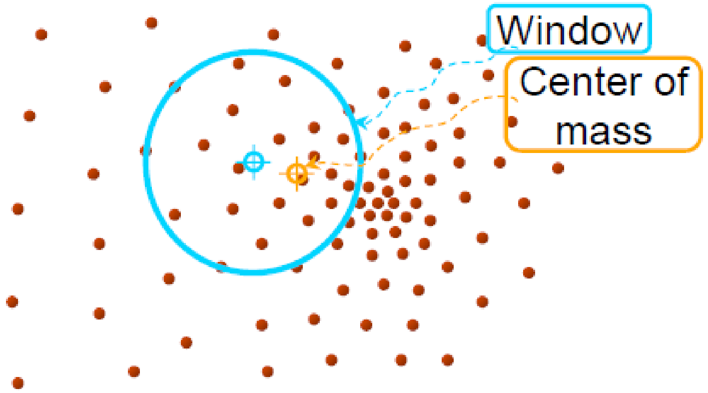
– Choose a search window (size and location)



Mean Shift Clustering

The mean shift algorithm seeks the *modes* or local maximums of density of a given distribution

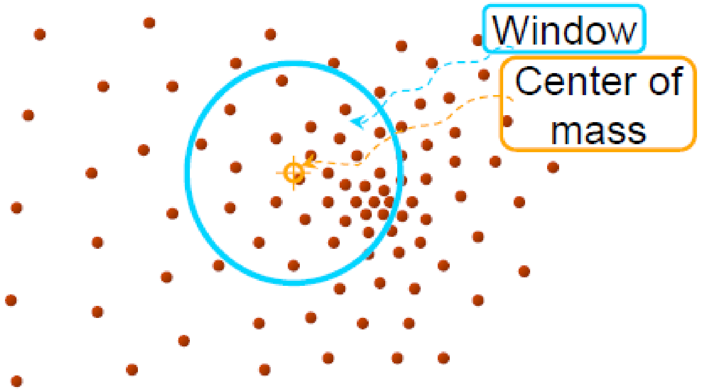
- Choose a search window (size and location)
- Compute the mean of the data in the search window



Mean Shift Clustering

The mean shift algorithm seeks the *modes* or local maximums of density of a given distribution

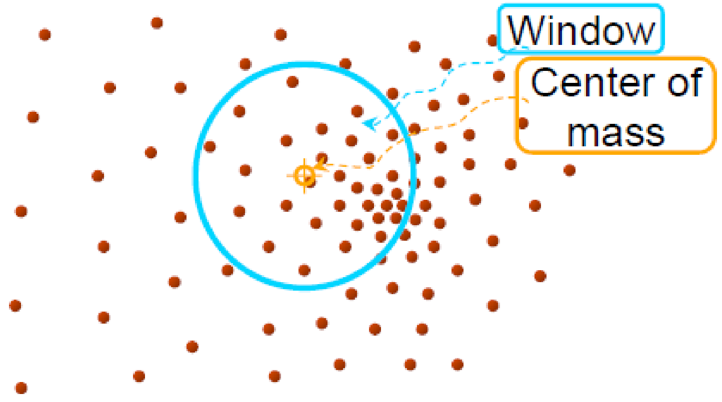
- Choose a search window (size and location)
- Compute the mean of the data in the search window
- Center the search window at the new mean location



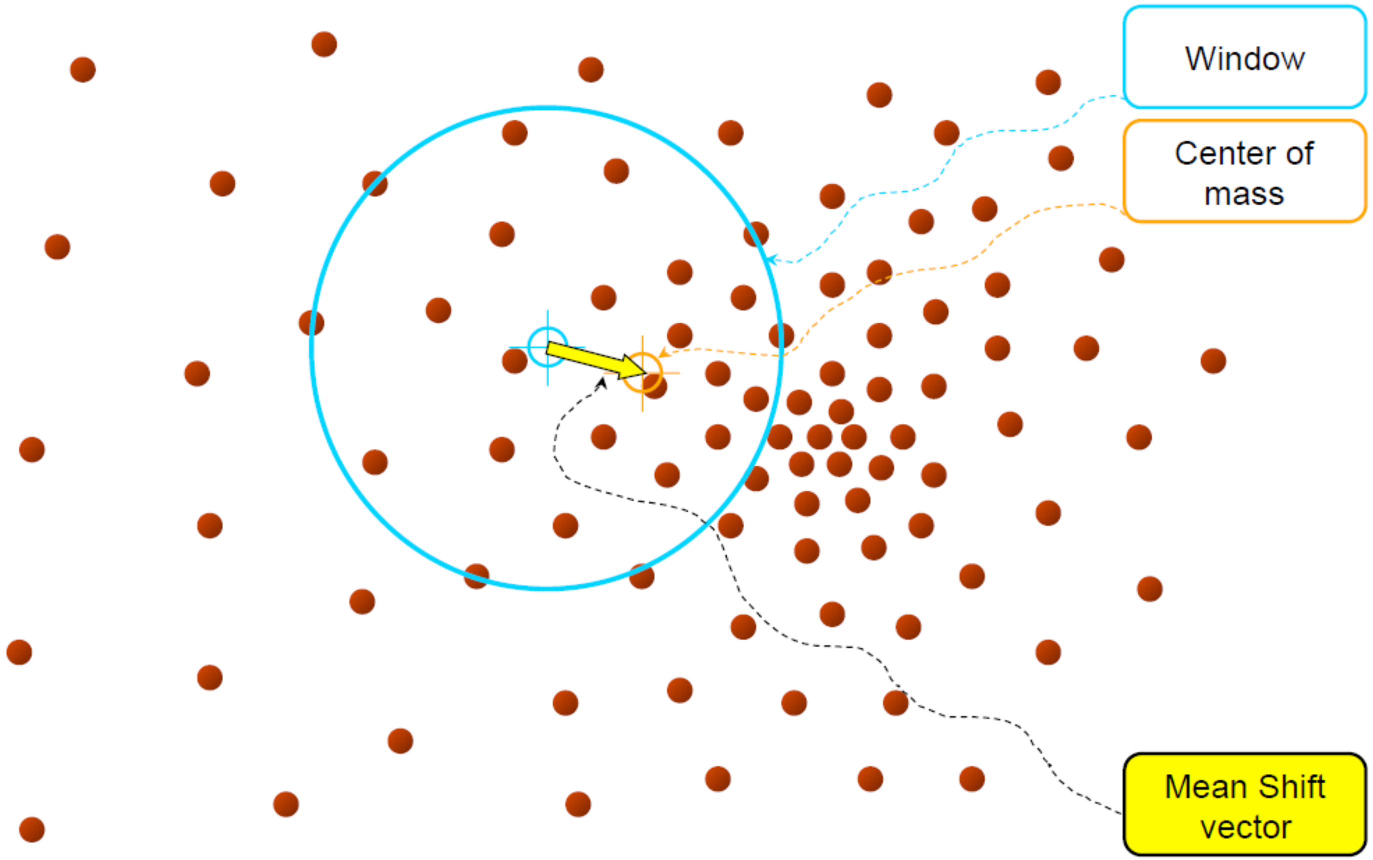
Mean Shift Clustering

The mean shift algorithm seeks the *modes* or local maximums of density of a given distribution

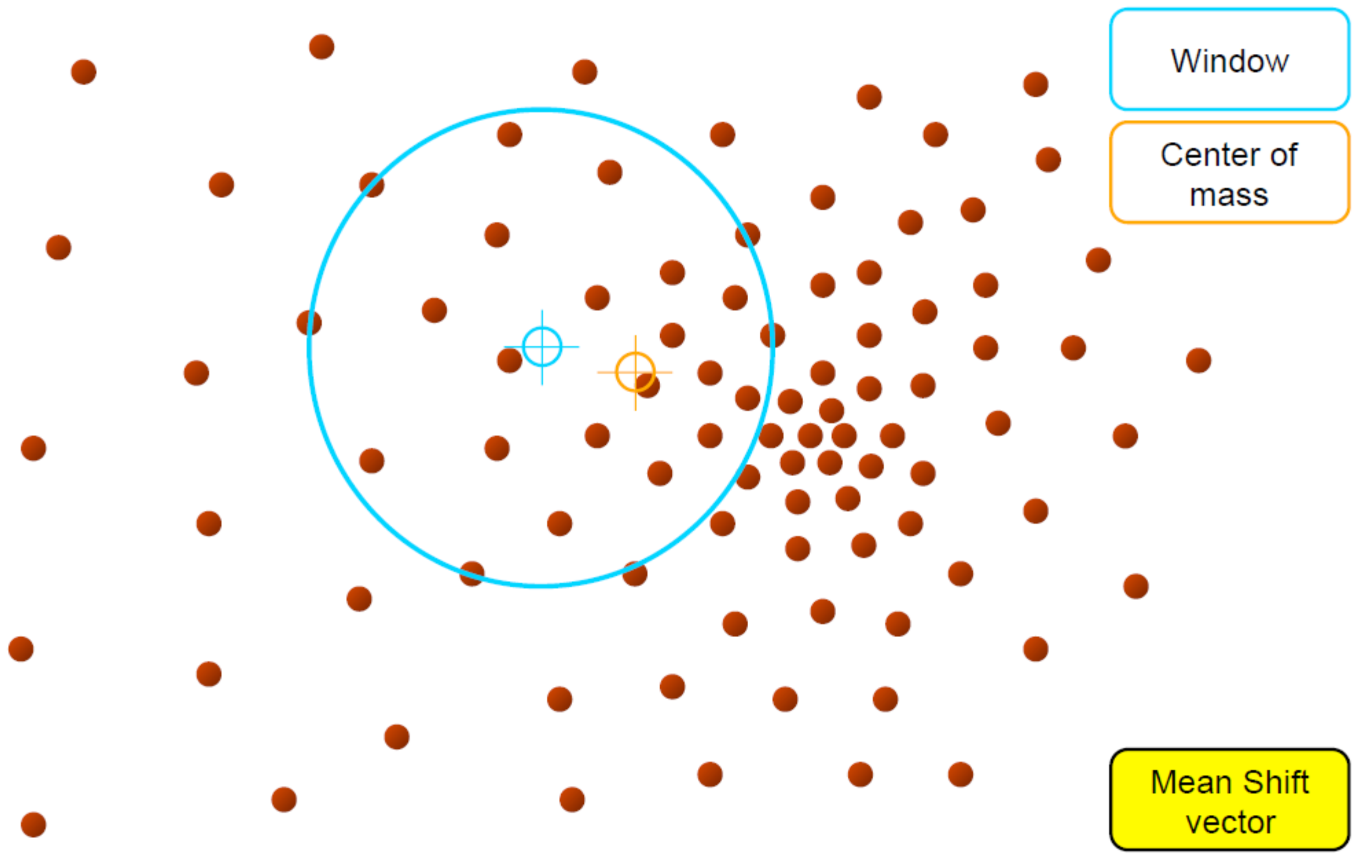
- Choose a search window (size and location)
- Compute the mean of the data in the search window
- Center the search window at the new mean location
- Repeat until convergence



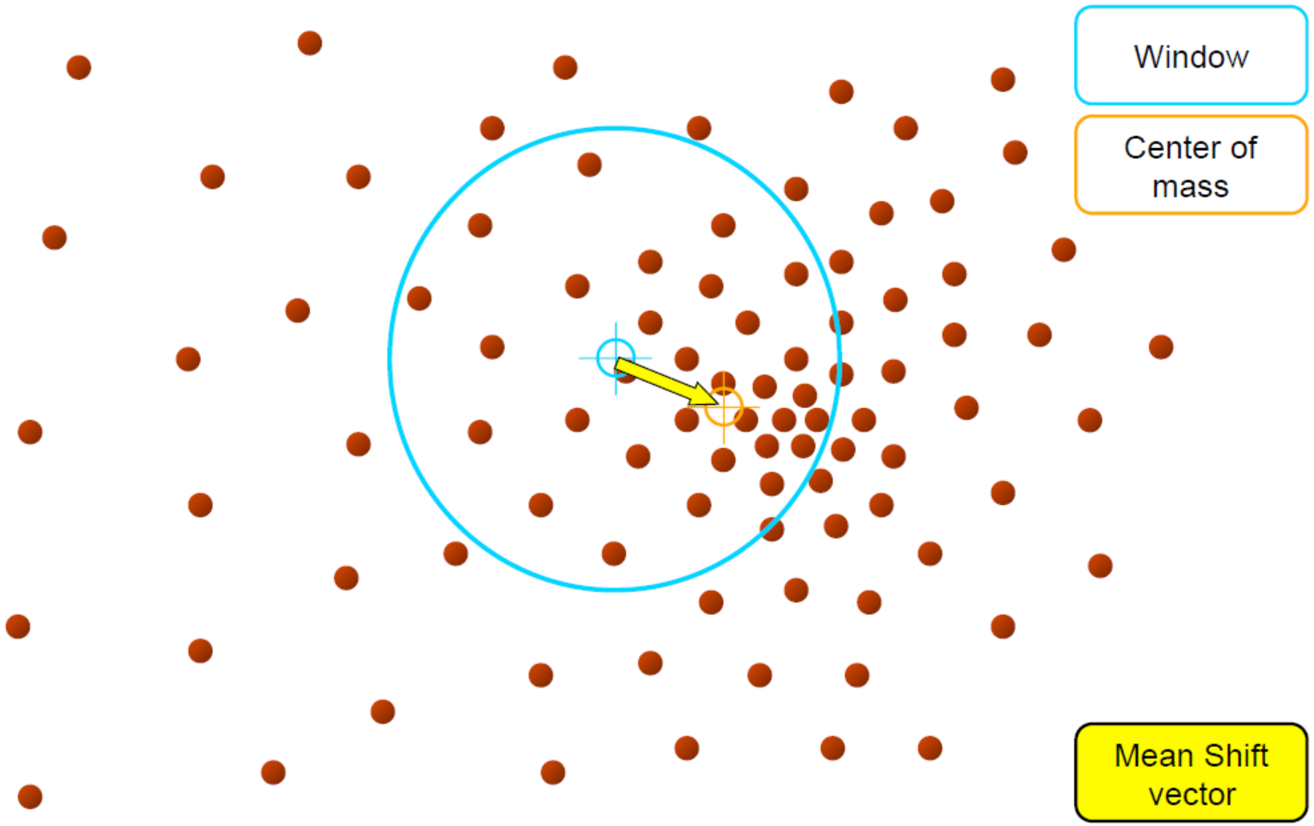
Mean Shift Clustering



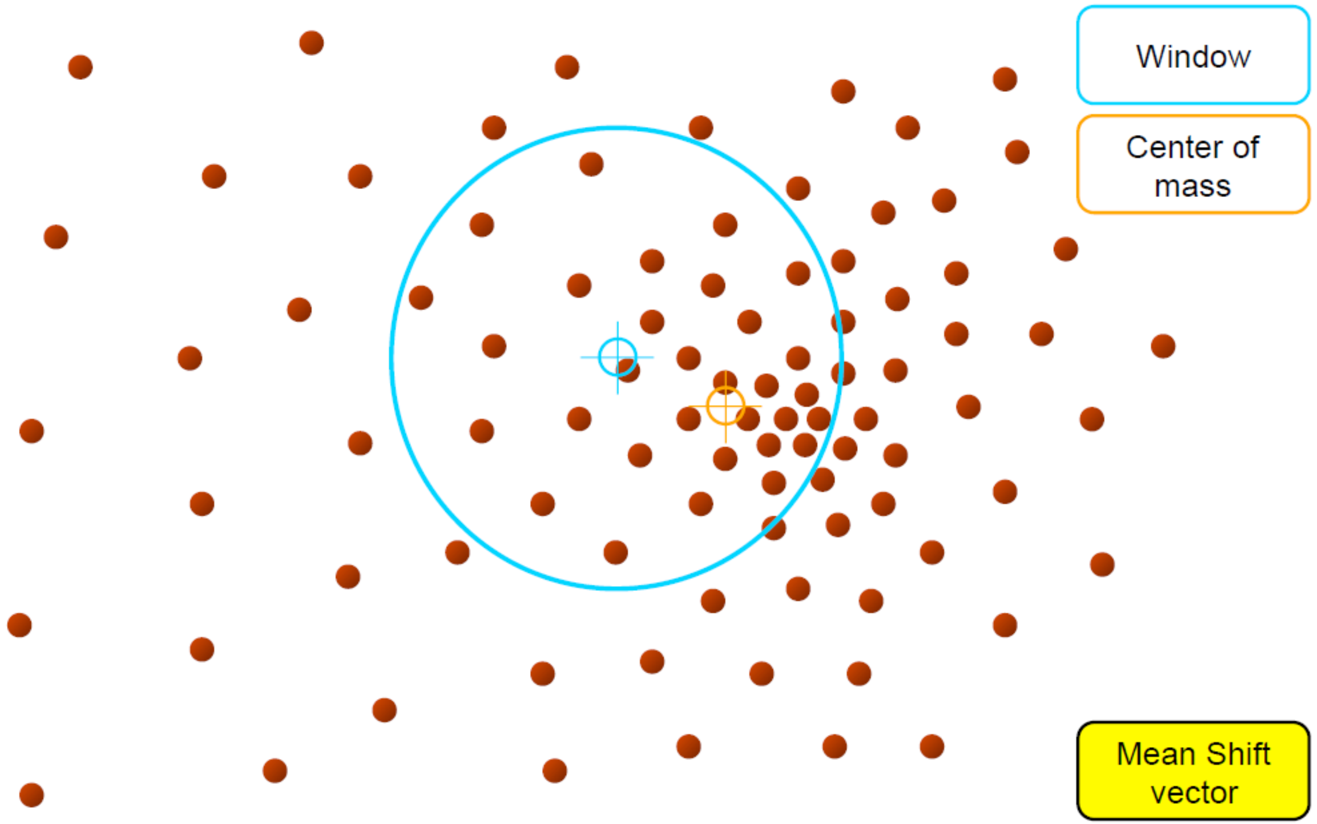
Mean Shift Clustering



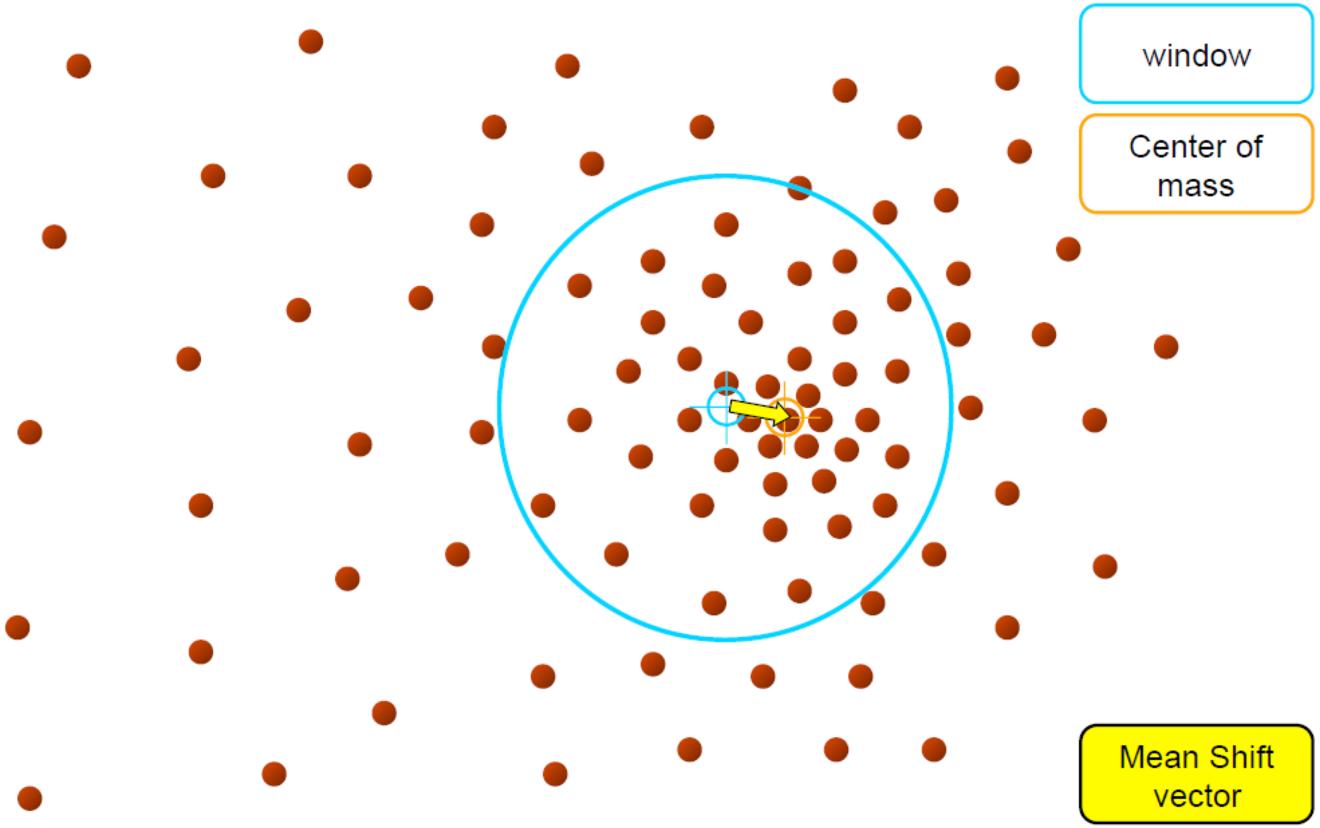
Mean Shift Clustering



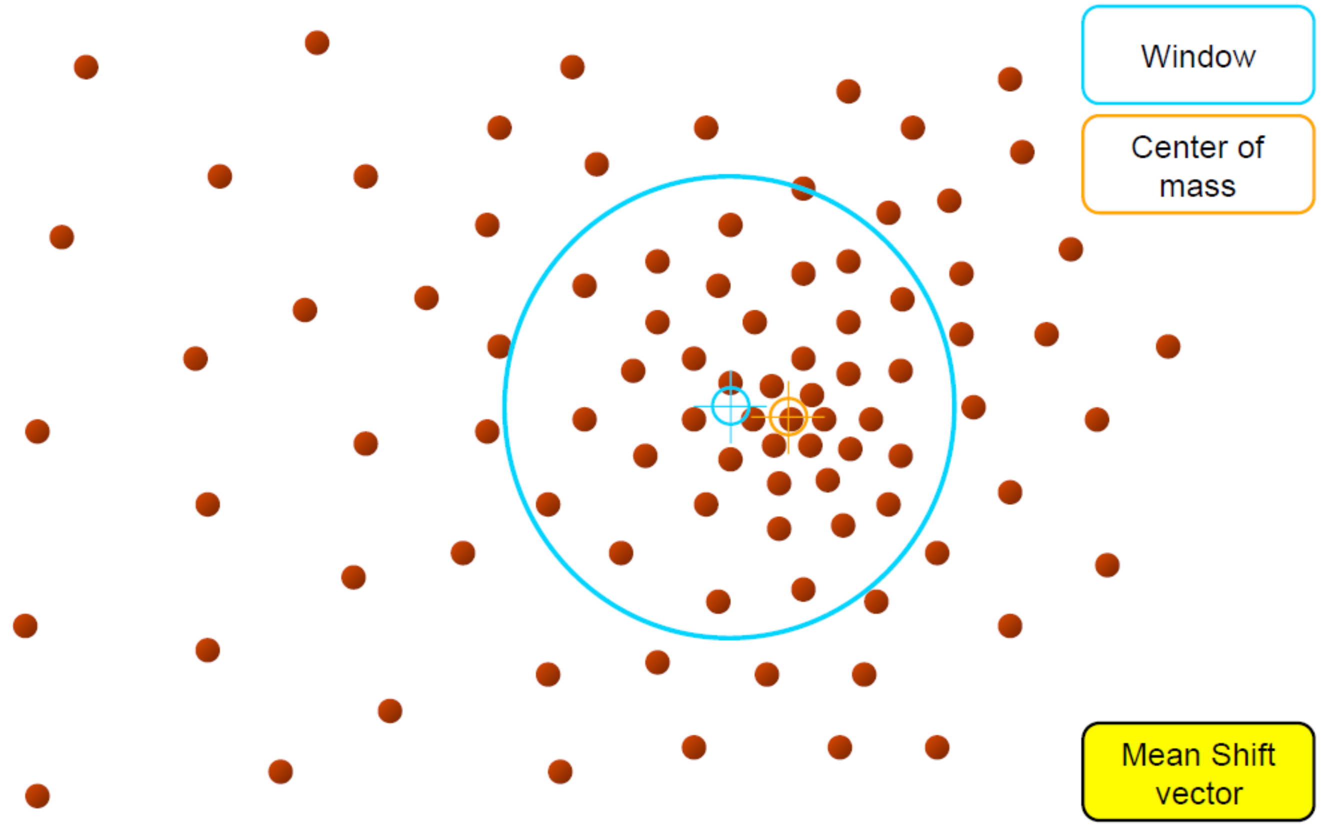
Mean Shift Clustering



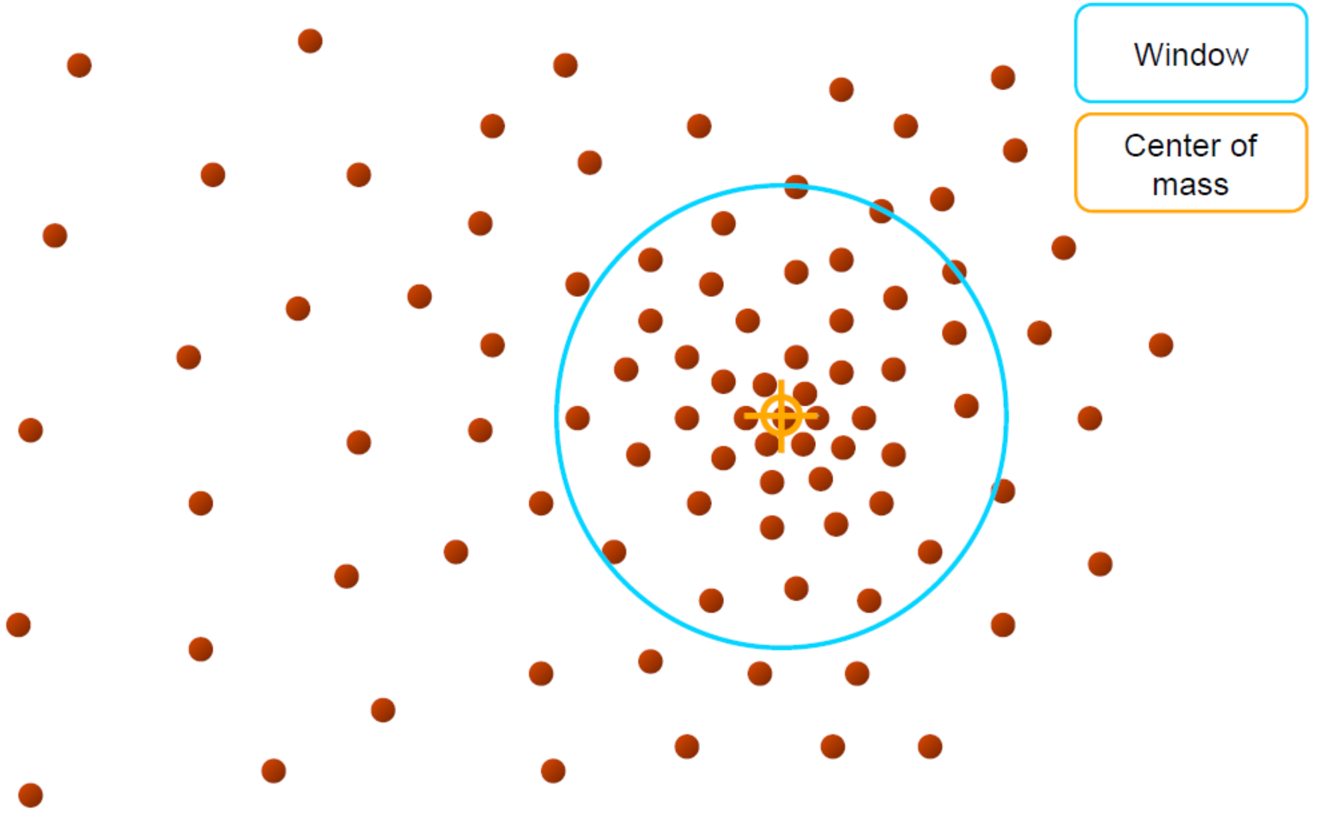
Mean Shift Clustering



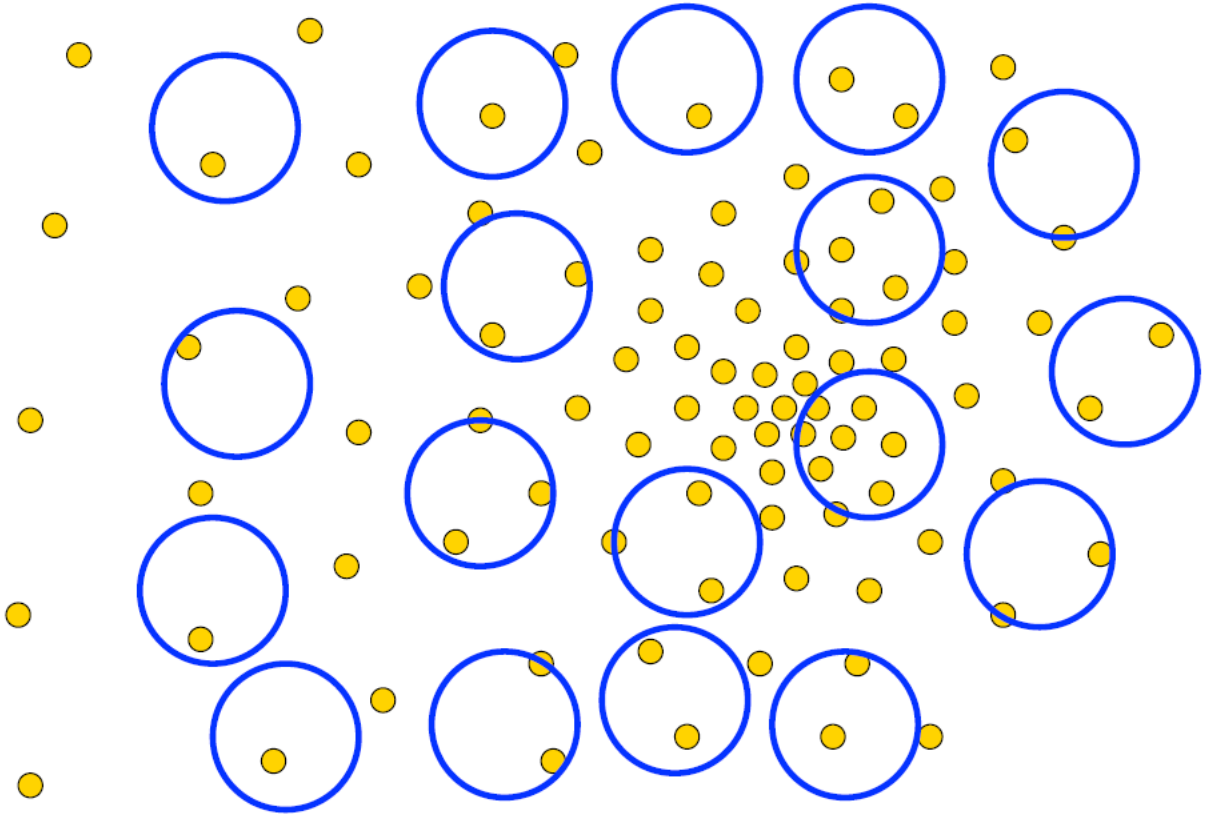
Mean Shift Clustering



Mean Shift Clustering



Mean Shift Clustering

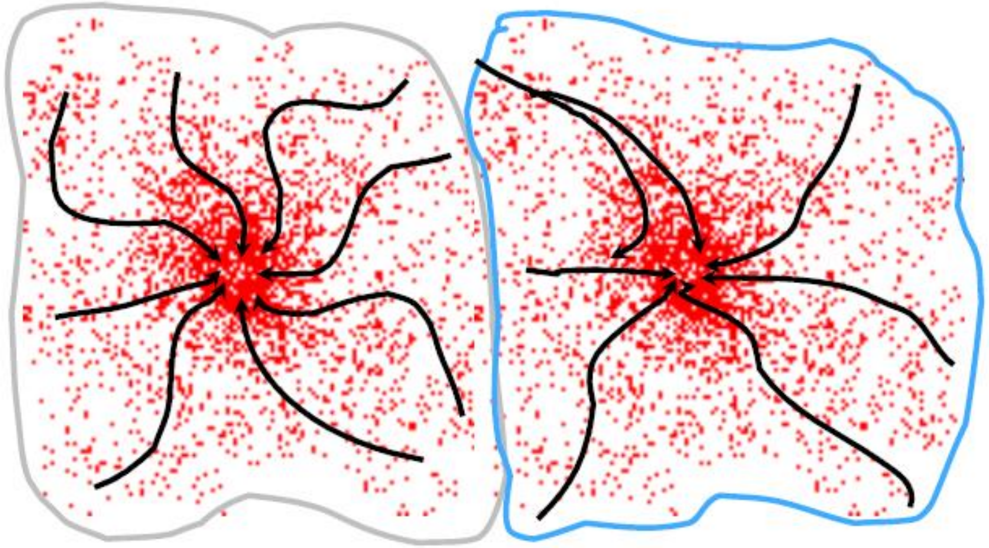


• **Tesselate the space with windows**

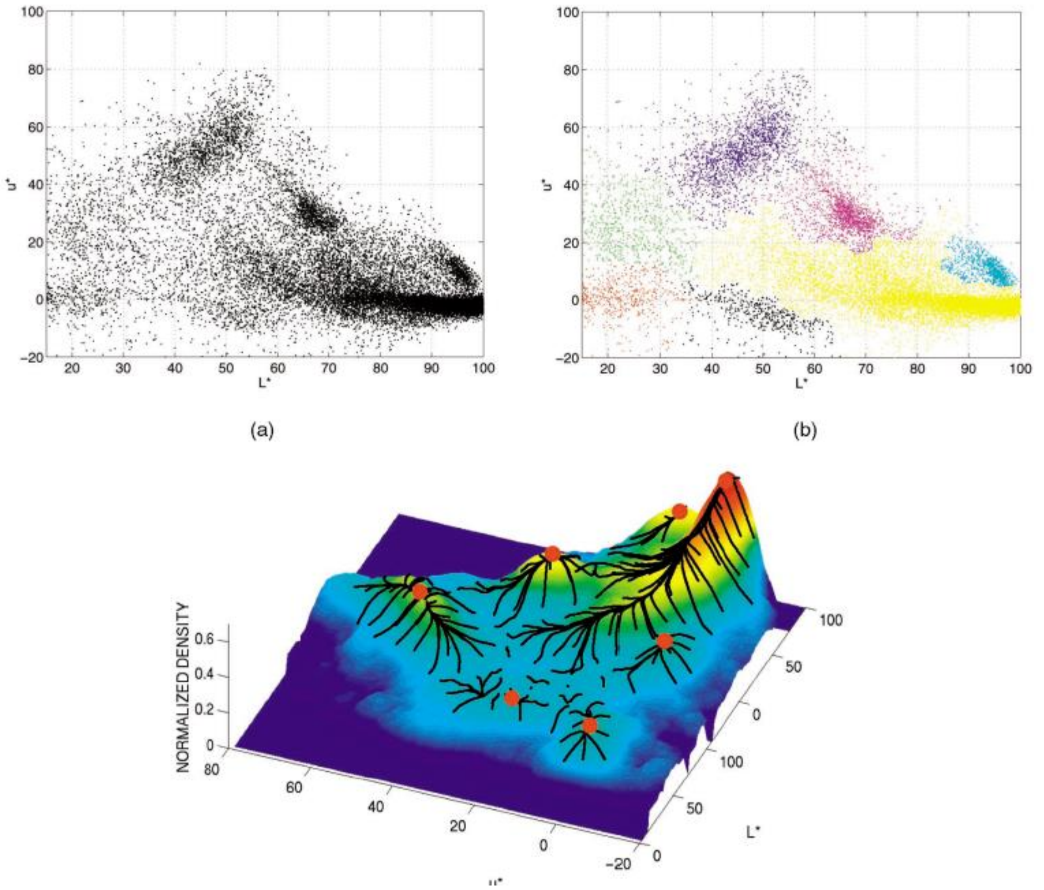
•Merge windows that end up near the same "peak" or model

Mean Shift Clustering

- Attraction basin: the region for which all trajectories lead to the same mode
- Cluster: all data points in the attraction basin of a mode

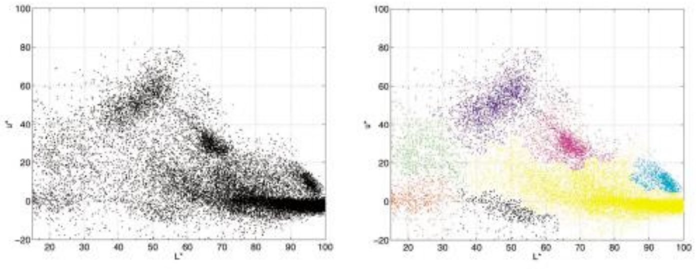


Mean Shift Clustering

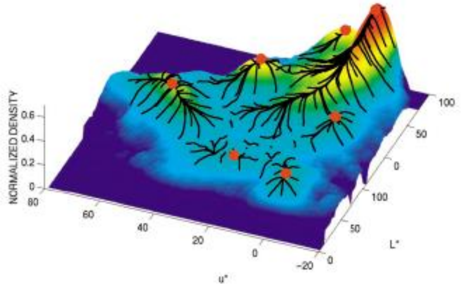


Clustering based Segmentation: Mean Shift

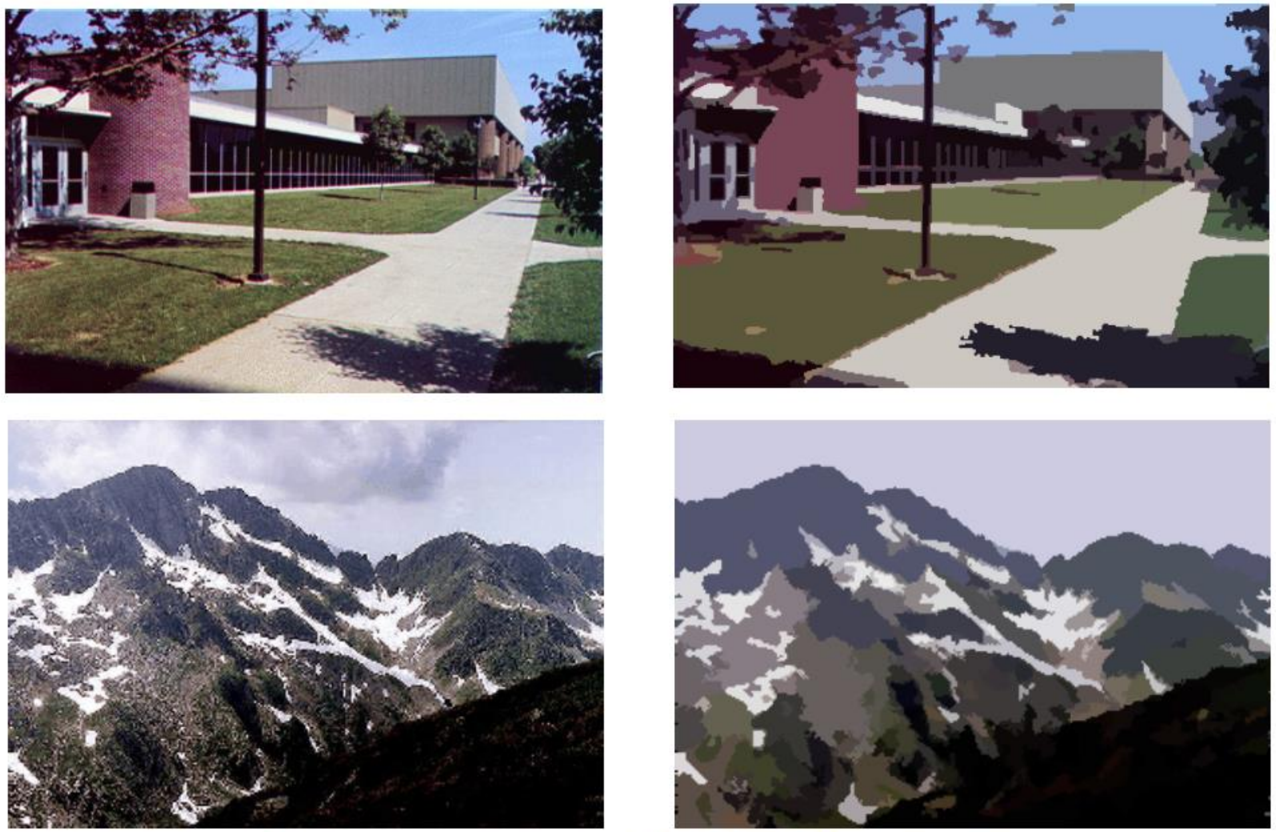
- Find features (color, gradients, texture, etc)
- Plot points in a joint feature-spatial space, e.g. (u, v, R, G, B)
- Initialize windows at individual pixel locations
- Perform mean shift for each window until convergence
- Merge windows that end up near the same "peak" or mode



(a) (b)



Clustering based Segmentation: Mean Shift Results



http://www.caip.rutgers.edu/~comanici/MSPAMI/msPamiResults.html

Thank You ☺