Image Processing & Computer Vision

Color Segmentation Using Clustering

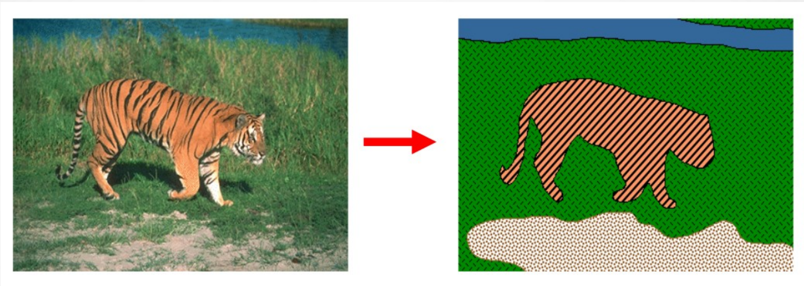
Mayada Hadhoud
Computer Engineering Department
Cairo University

Agenda

- **Segmentation Techniques**
  - K means
  - Mean Shift
  - Hierarchical Clustering

Image Segmentation

Goal : *identify groups of pixels that go together*



Clustering

Group together similar data points and represent them with a single token

__Key Challenges:__

1) What makes two points/images/patches similar?

2) How do we compute an overall grouping from pairwise similarities?

Why do we Cluster?

- ***Summarizing data***
  - *Look at large amounts of data*
  - *Patch-based compression*
  - *Represent a large continuous vector with the cluster number*
- ***Segmentation***
  - *Separate the image into different regions*
- ***Prediction***
  - *Images in the same cluster may have the same labels*

Clustering Techniques

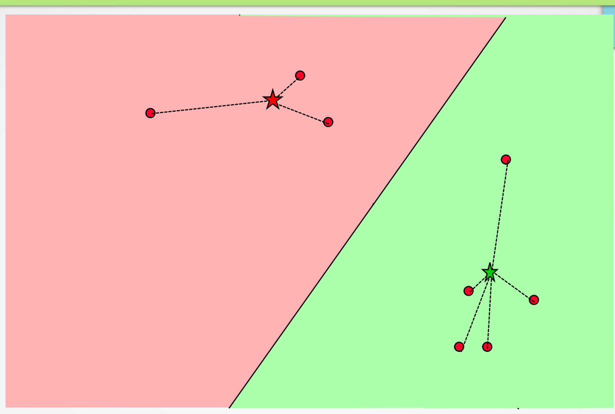
- ***Kmeans***
- ***Mean Shift***
- ***Hierarchical Clustering***

• K-means Clustering

- Choose a fixed number of clusters
- Choose cluster centers and point-cluster allocations to minimize error

- **Algorithm**
  - fix cluster centers; allocate points to closest cluster
  - fix allocation; compute best cluster centers
- x could be any set of features for which we can compute a distance (careful about scaling)

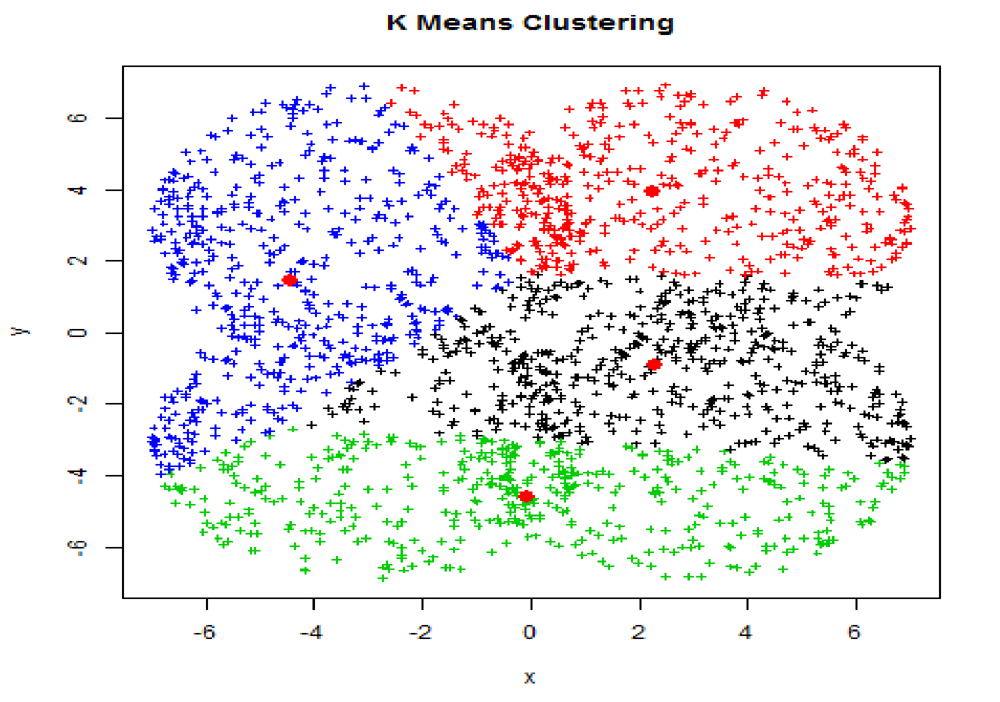
• K-Means

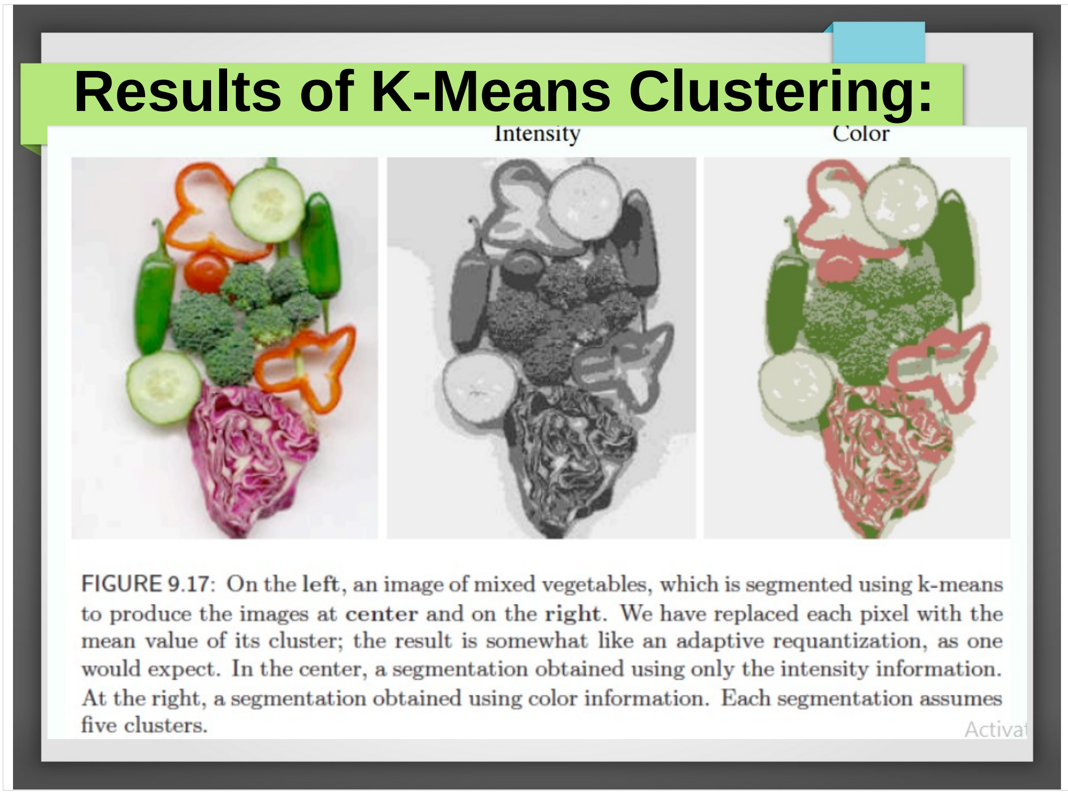


• K-Means

```
Choose k data points to act as cluster centers

Until the cluster centers are unchanged

    Allocate each data point to cluster whose center is nearest

    Now ensure that every cluster has at least
    one data point; possible techniques for doing this include .
    supplying empty clusters with a point chosen at random from
    points far from their cluster center.

    Replace the cluster centers with the mean of the elements
    in their clusters.

end
```

Algorithm 16.5: *Clustering by K-Means*


K Means Clustering
y
6
4
2
0
-2
-4
-6
-6
-4
-2
0
2
4
6
x



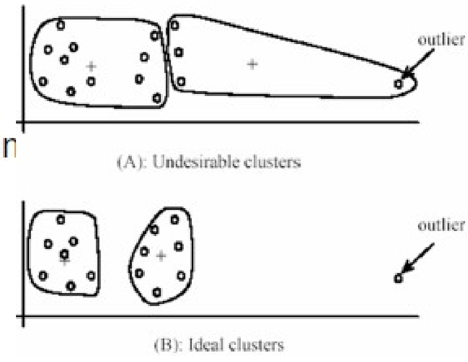
- I gave each pixel the mean intensity or mean color of its cluster --- this is basically just vector quantizing the image intensities/colors. Notice that there is no requirement that clusters be spatially localized and they're not.

Finding the Best Set of Clusters

- It can be proved that the k-means algorithm will always terminate, but it **does not necessarily find the best set of clusters**
- The initial selection of centroids can significantly affect the result.
  - **Solution**: Try different initial selection and take the best
  - But what should be k
  - **Try different k**

K-Means pros and cons

- Pros
  - Simple and fast
  - Converges to a local minimum of the error function
- Cons
  - Need to pick K
  - Sensitive to initialization
  - Sensitive to outliers
  - Only finds “spherical” clusters



(A): Undesirable clusters

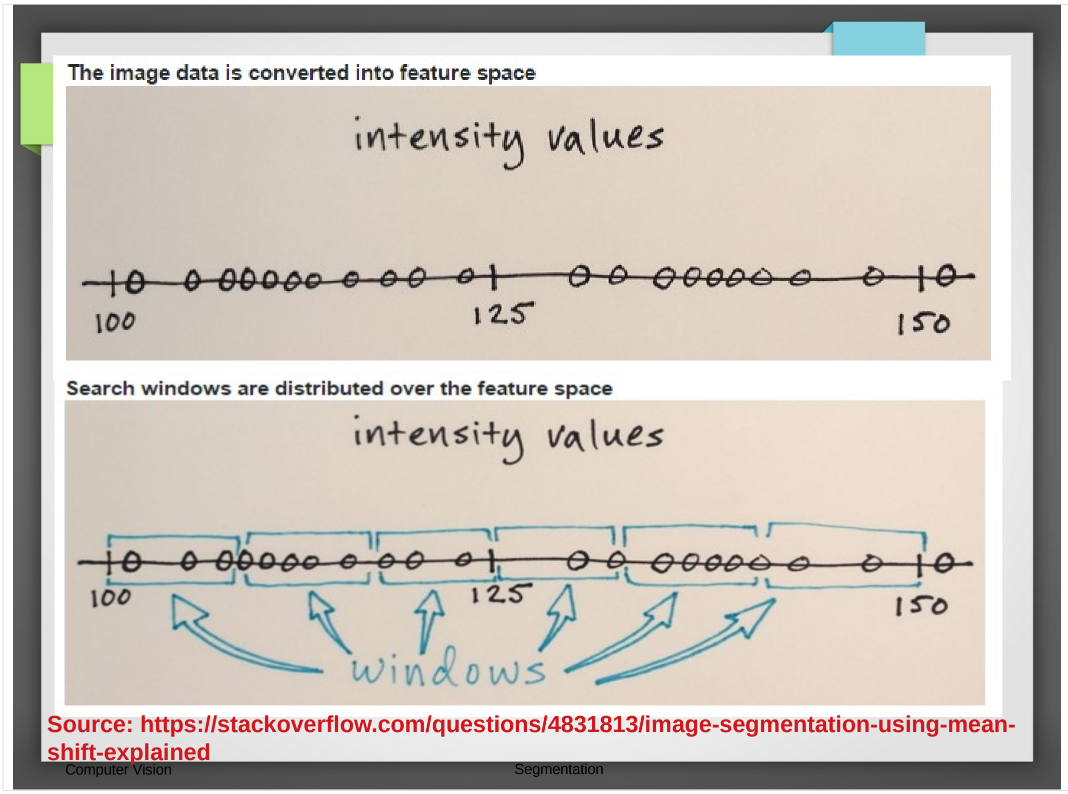
(B): Ideal clusters

Mean Shift Algorithm

The mean shift algorithm seeks a ***mode*** *or* ***local maximum*** of density of a given distribution

Steps:

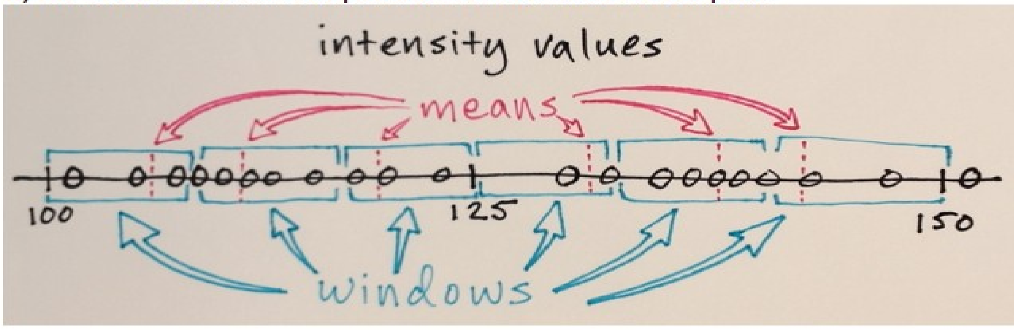
1. Choose a search window (width and location)
2. Compute the mean of the data in the search window
3. Center the search window at the new mean location
4. Repeat until convergence



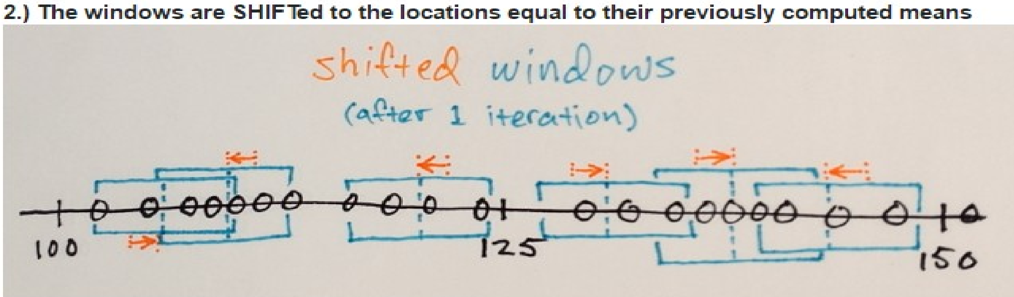
- Source: https://stackoverflow.com/questions/4831813/image-segmentation-using-mean-shift-explained

Mean-Shift iterations:

1.) The MEANs of the data samples within each window are computed

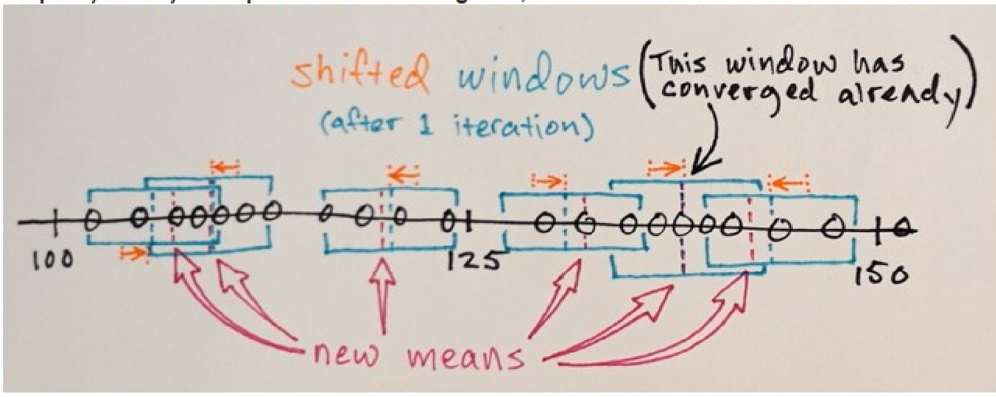


2.) The windows are SHIFTed to the locations equal to their previously computed means

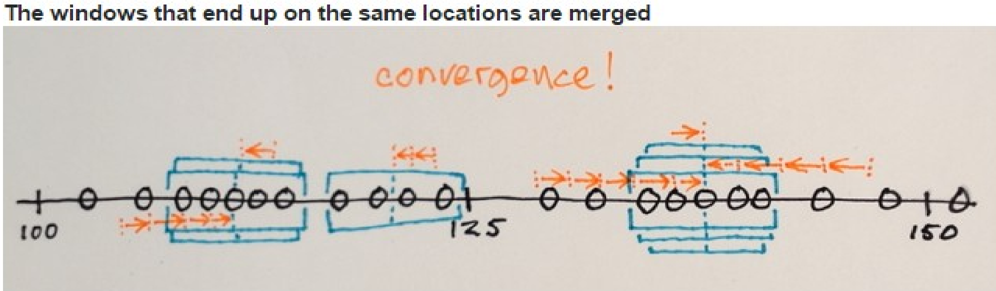


Source: https://stackoverflow.com/questions/4831813/image-segmentation-using-mean-shift-explained

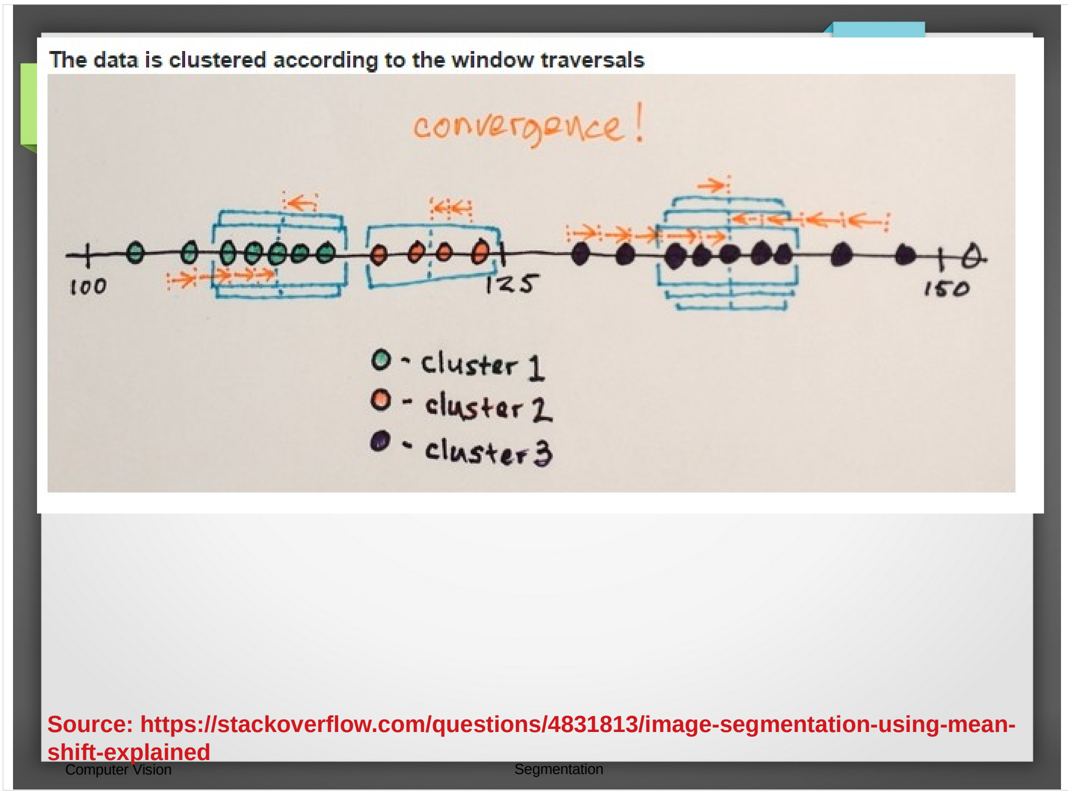
Steps 1.) and 2.) are repeated until convergence, i.e. all windows have settled on final locations



The windows that end up on the same locations are merged



Source: https://stackoverflow.com/questions/4831813/image-segmentation-using-mean-shift-explained

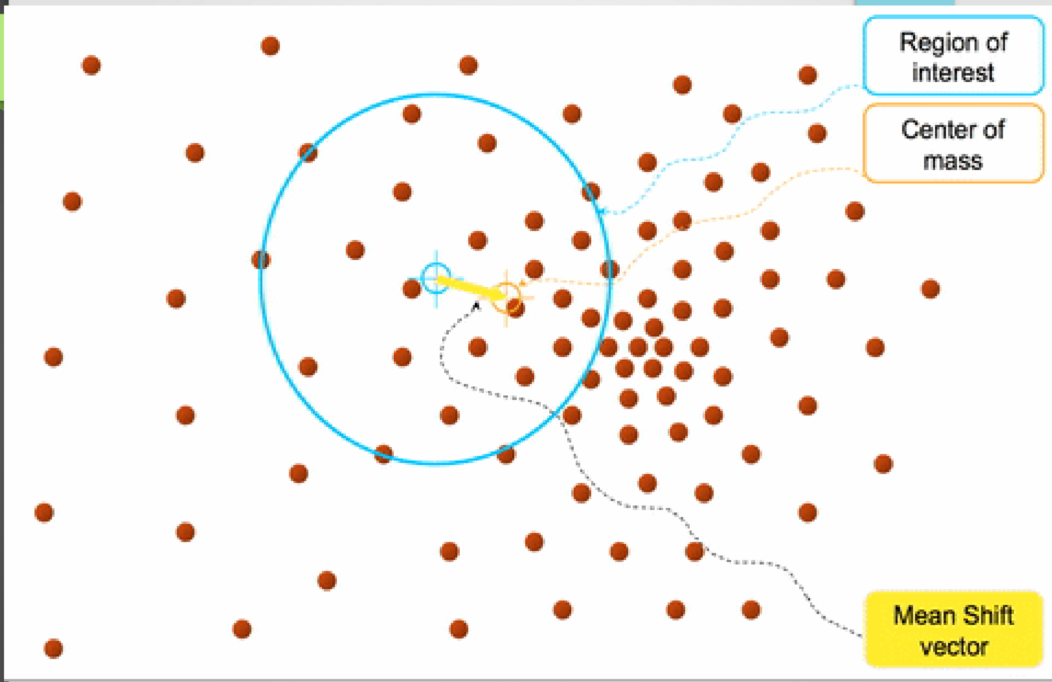


- all data that was traversed by windows that ended up at, say, location “2”, will form a cluster associated with that location.

- Choosing different window sizes might produce different results.

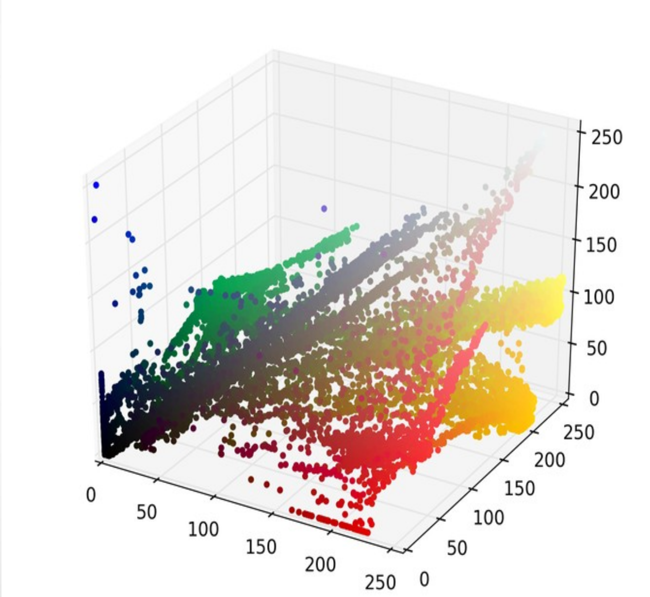
Mean Shift Clustering/Segmentation

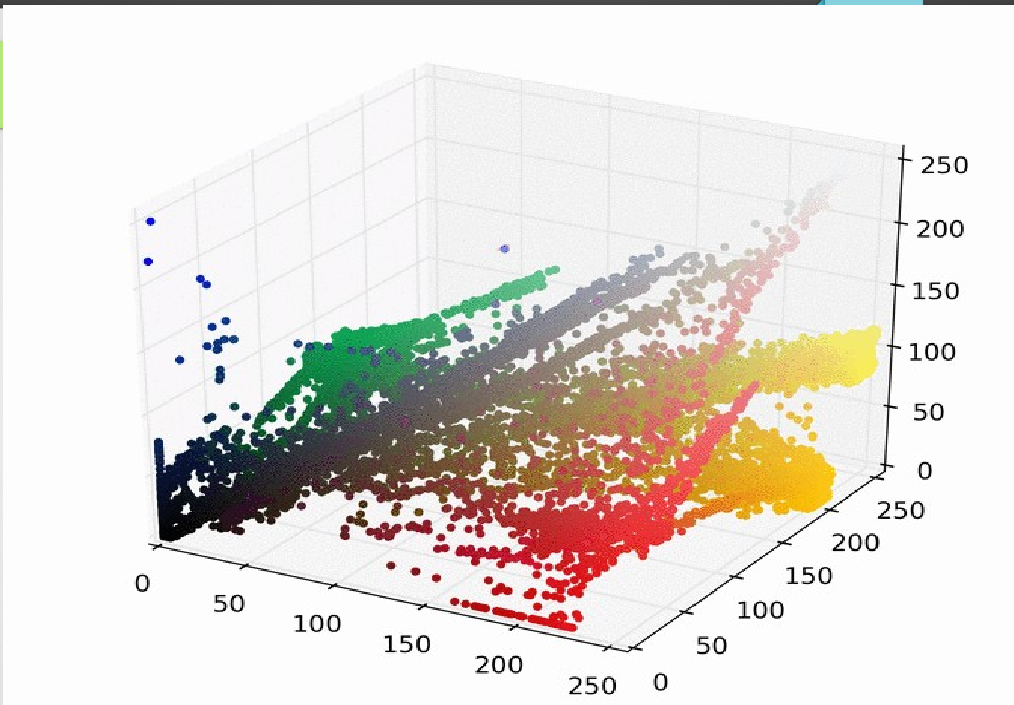
- Find features (color, gradients, texture, etc)
- Initialize windows at individual pixel locations
- Perform mean shift for each window until convergence
- Merge windows that end up near the same “peak” or mode



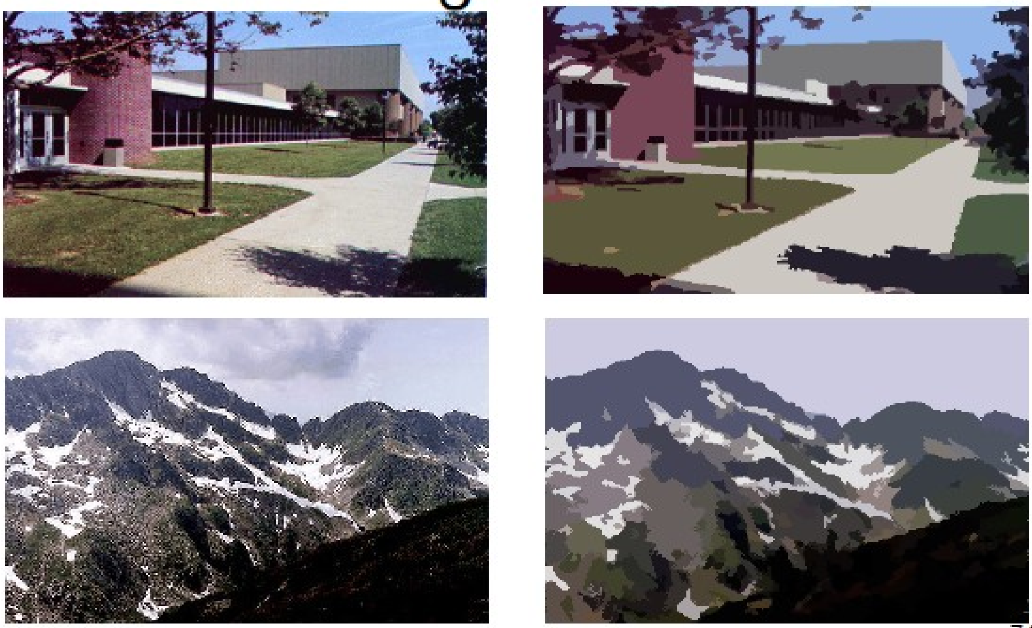
Region of interest
Center of mass
Mean Shift vector


Diesel #2
Regular
Plus
V-Power
Premium
Diesel
87
89
93

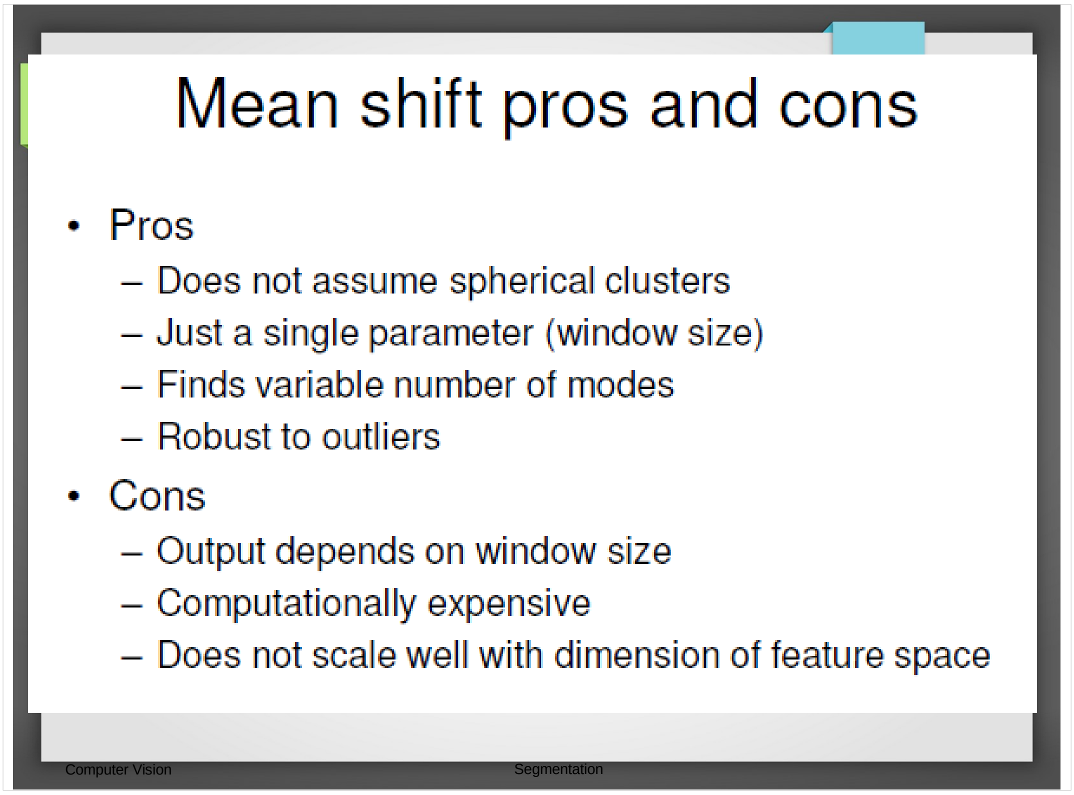

0
50
100
150
200
250
0
50
100
150
200
250
0
50
100
150
200
250


0
50
100
150
200
250
0
50
100
150
200
250
0
50
100
150
200
250

Mean shift segmentation results



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



- Mean Shift and Clustering : Consider a set of points in two-dimensional space. Assume a circular window centered at C and having radius r as the kernel. Mean shift is a hill climbing algorithm which involves shifting this kernel iteratively to a higher density region until convergence. Every shift is defined by a mean shift vector. The mean shift vector always points toward the direction of the maximum increase in the density. At every iteration the kernel is shifted to the centroid or the mean of the points within it. The method of calculating this mean depends on the choice of the kernel.

Hierarchical Clustering

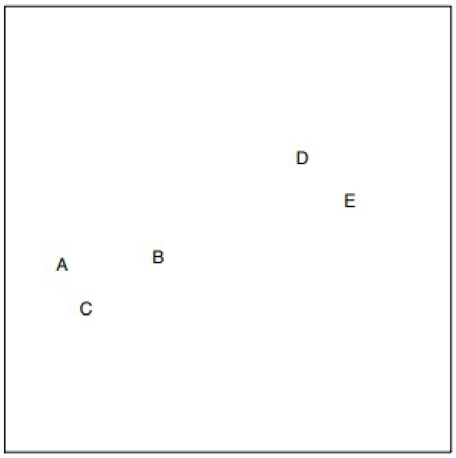
Hierarchical clustering *is an alternative approach that does not require a pre-specified choice of K, and which provides a deterministic answer (no randomness)*

***Hierarchical clustering* approaches:**

- Bottom-up

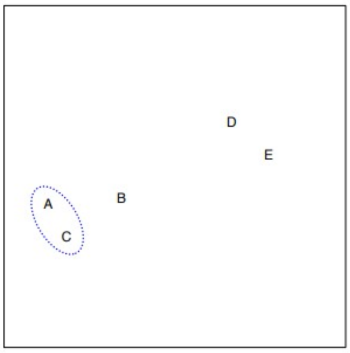
- Top-down

Hierarchical Clustering

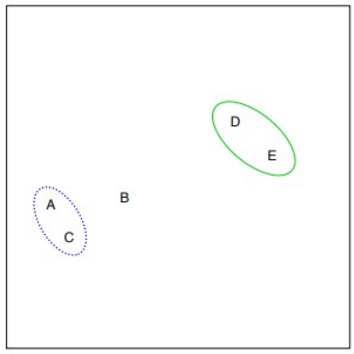


Each point starts as its own cluster

Hierarchical Clustering

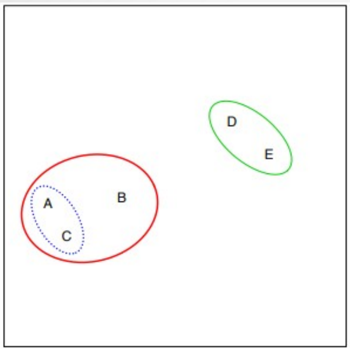


We merge the two clusters (points) that are closet to each other

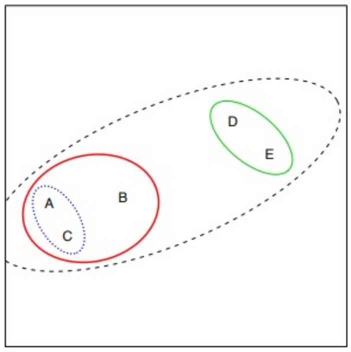


Then we merge the next two closest clusters

Hierarchical Clustering



Then the next two closest clusters…

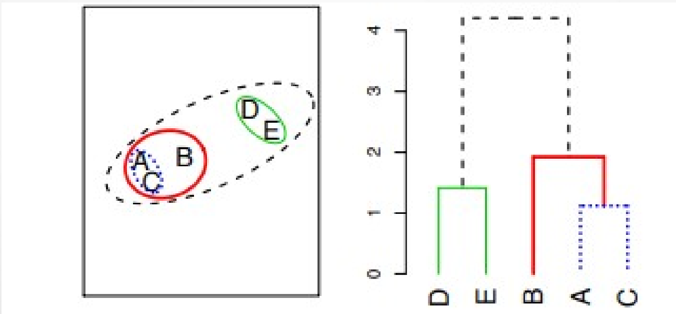


Until at last all of the points are all in a single cluster

Hierarchical Clustering

Agglomerative Hierarchical Clustering

- Start with each point in its own cluster.
- Identify the two closest clusters. Merge them.
- Repeat until all points are in a single cluster



Hierarchical Clustering

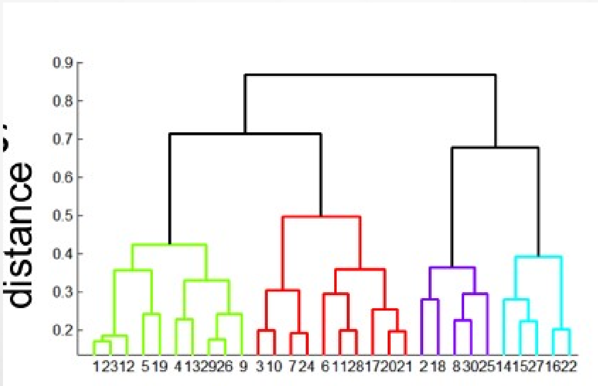
- At our first step, each cluster is a single point, so **<u>we start by merging the two observations that have the lowest dissimilarity</u>**
- But after that…we need to think about distances not between points, but between sets (clusters)
- The dissimilarity between two clusters is called the **<u>linkage</u>**
- i.e., Given two sets of points, G and H, a linkage is a dissimilarity measure d(G, H) telling us how different the points in these sets are

Linkage	*Description*
Complete	Maximal inter-cluster dissimilarity. Compute all pairwise dissimilarities between the observations in cluster A and the observations in cluster B, and record the *largest* of these dissimilarities.
Single	Minimal inter-cluster dissimilarity. Compute all pairwise dissimilarities between the observations in cluster A and the observations in cluster B, and record the *smallest* of these dissimilarities.
Average	Mean inter-cluster dissimilarity. Compute all pairwise dissimilarities between the observations in cluster A and the observations in cluster B, and record the *average* of these dissimilarities.
Centroid	Dissimilarity between the centroid for cluster A (a mean vector of length p) and the centroid for cluster B. Centroid linkage can result in undesirable *inversions*.

Hierarchical Clustering

How many clusters??

- Clustering creates a dendrogram (a tree)
- Threshold based on **max number of clusters** or based on **distance between merges**



Hierarchical Clustering

<u>Good</u>
- Simple to implement, widespread application.
- Clusters have adaptive shapes.
- Provides a hierarchy of clusters.
- No need to specify number of clusters in advance

<u>Bad</u>
- May have imbalanced clusters.
- Still have to choose number of clusters or threshold.
- Does not scale well.