Lecture 11

Clustering: K-means and Mean Shift

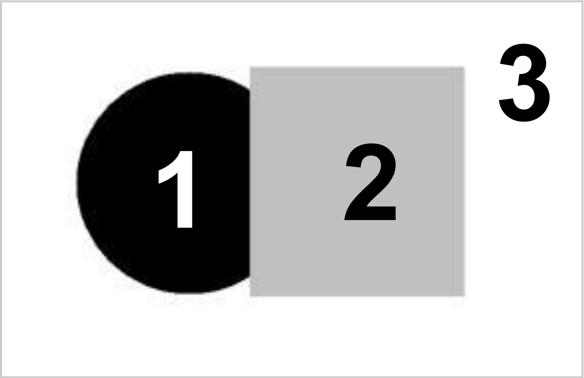
What will we learn today?

- **K-means clustering**
- Mean-shift clustering

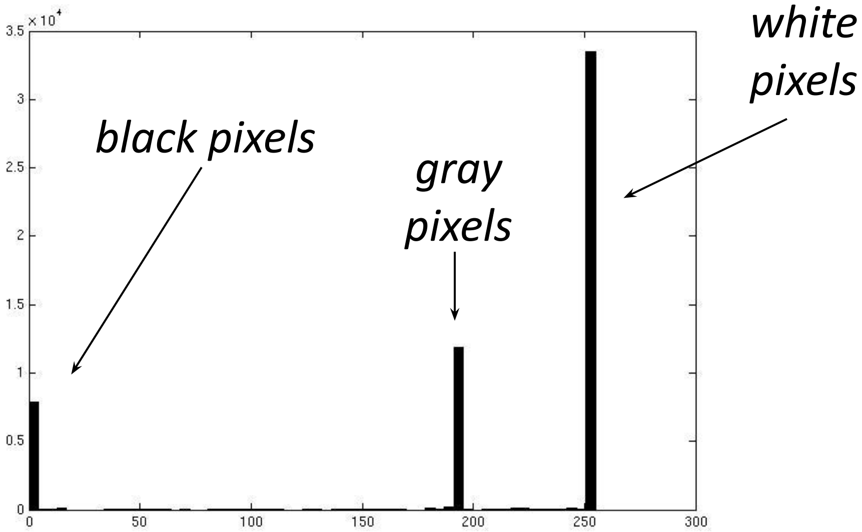
Reading: Szeliski Chapters: 5.2.2, 7.5.2

D. Comaniciu and P. Meer, Mean Shift: A Robust Approach toward Feature Space Analysis, PAMI 2002.

Image Segmentation: Toy Example



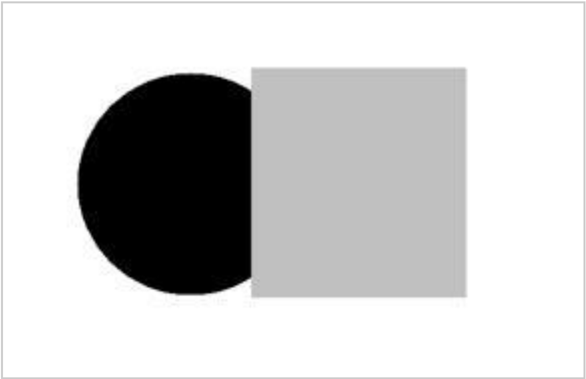
input image



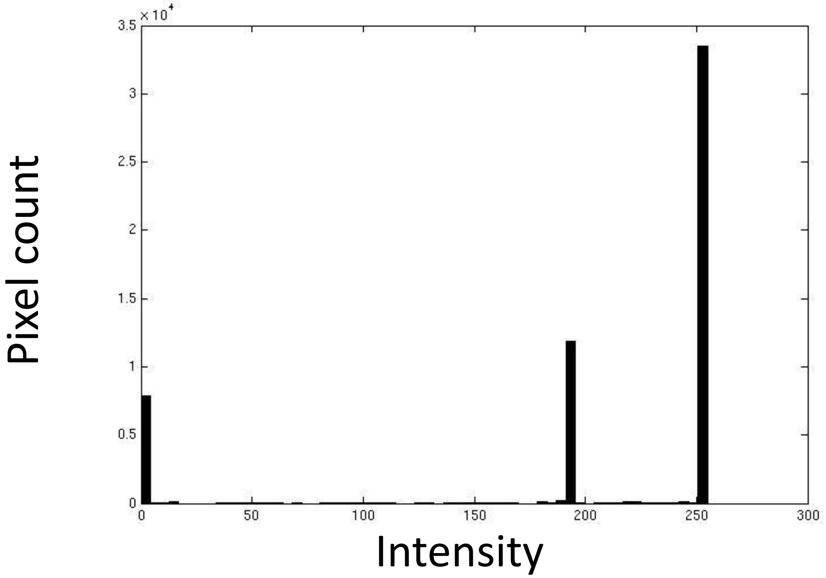
intensity

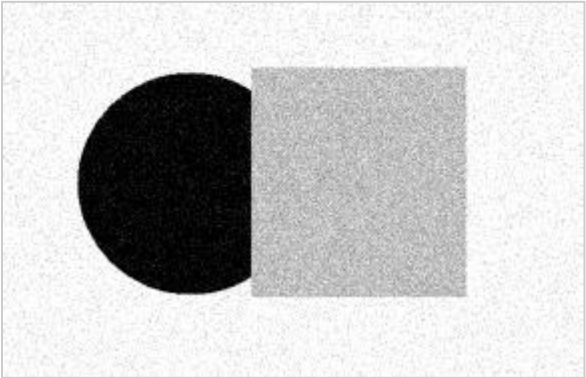
- These intensities define the three groups.
- We could label every pixel in the image according to which of these primary intensities it is.
  - i.e., segment the image based on the intensity feature.
- What if the image isn't quite so simple?

Slide credit: Kristen Grauman

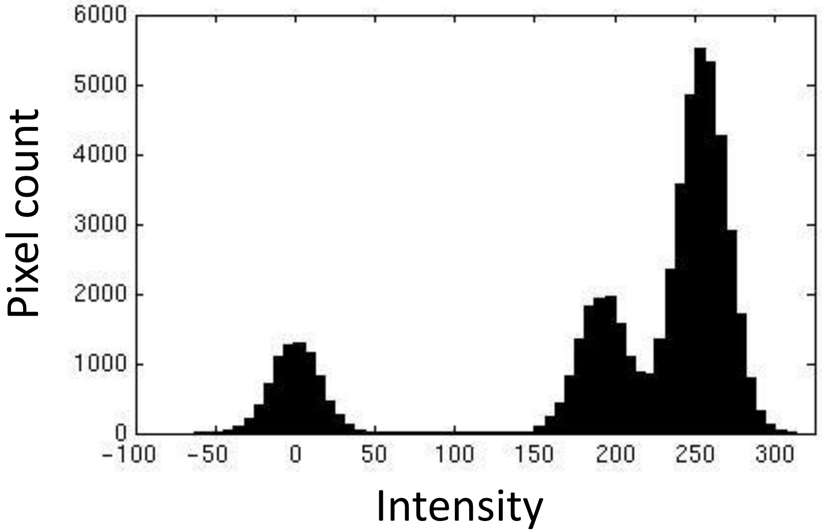


Input image

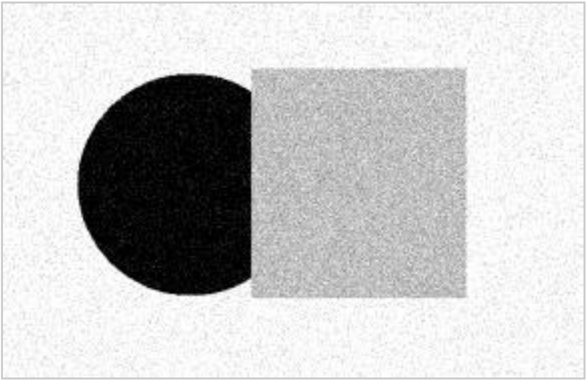




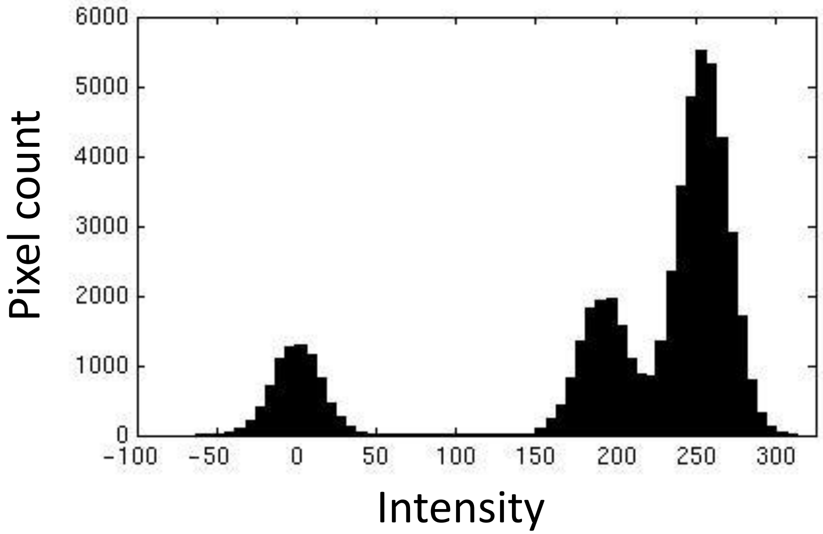
Input image



Slide credit: Kristen Grauman

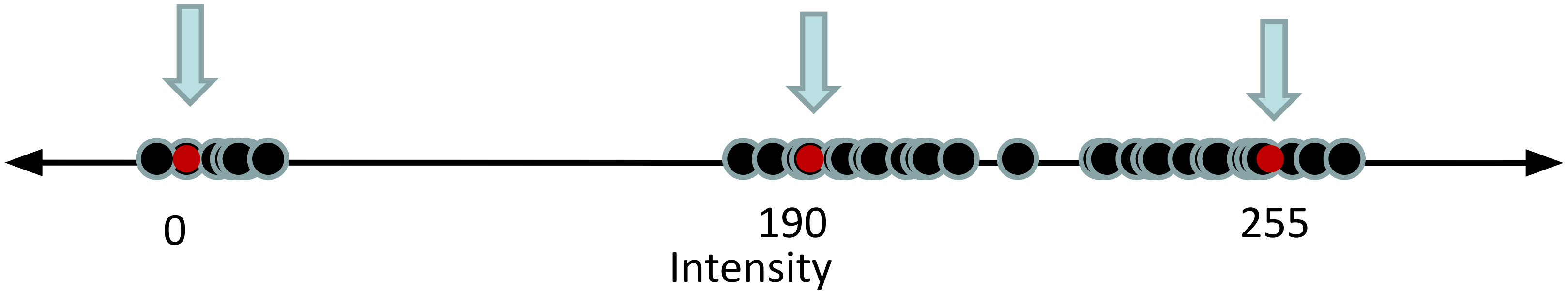


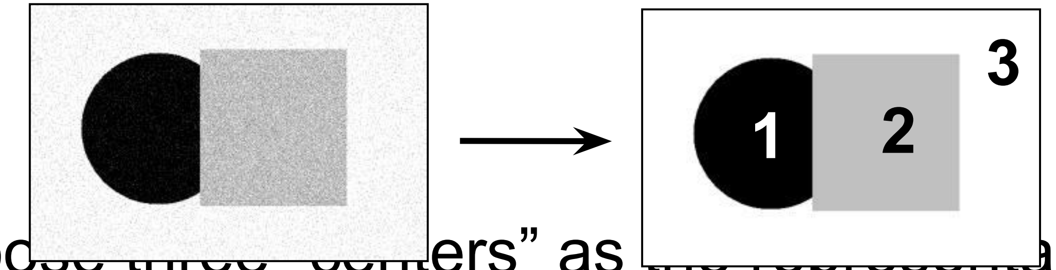
Input image



- Now how to determine the three main intensities that define our groups?
- We need to cluster.

Slide credit: Kristen Grauman





- Goal: choose three “centers” as the representative intensities and label every pixel according to which of these centers it is nearest to.

- Best cluster centers are those that minimize Sum of Square Distance (SSD) between all points and their nearest cluster center c_i:

$$SSD = \sum_{cluster\,i} \sum_{x \in cluster\,i} (x - c_i)^2$$

Slide credit: Kristen Grauman

Clustering

Goal: cluster to minimize variance in data given clusters

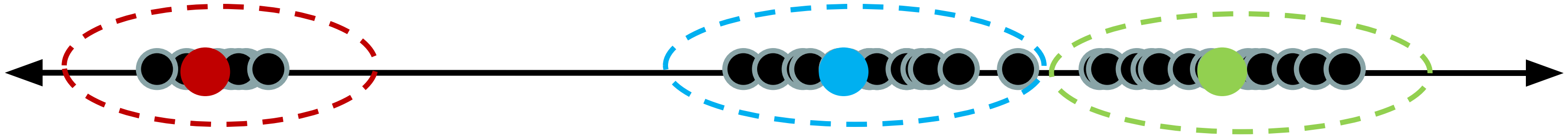
Cluster center Data

$$c^*,\ \delta^* = \underset{c,\,\delta}{\arg\min} \frac{1}{N} \sum_{j}^{N} \sum_{i}^{K} \delta_{ij} \left(c_i - x_j\right)^2$$

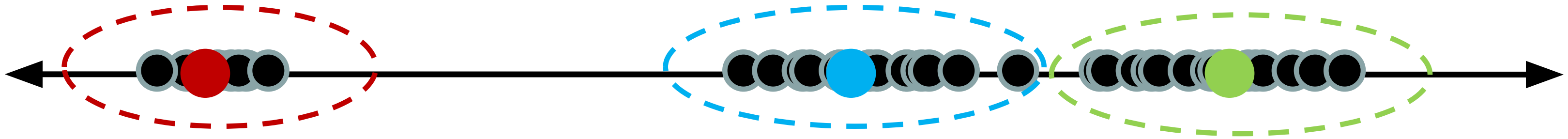
Whether x_j is assigned to c_i

Clustering

- With this objective, it is a "chicken and egg" problem:
  - If we knew the *cluster centers*, we could allocate points to groups by assigning each to its closest center.



  - If we knew the *group memberships*, we could get the centers by computing the mean per group.



Slide credit: Kristen Grauman

K-means clustering

1. Initialize ($t = 0$): cluster centers $c_1, ..., c_K$

2. Compute δ^t : assign each point to the closest center
    - δ^t denotes the set of assignment for each x_j to cluster c_i at iteration t

$$\delta^t = \underset{\delta}{\operatorname{argmin}} \frac{1}{N} \sum_j^N \sum_i^K \delta_{ij}^{t-1} \left(c_i^{t-1} - x_j\right)^2$$

3. Computer c^t : update cluster centers as the mean of the points

$$c^t = \underset{c}{\operatorname{argmin}} \frac{1}{N} \sum_j^N \sum_i^K \delta_{ij}^{t} \left(c_i^{t-1} - x_j\right)^2$$

4. Update $t = t + 1$, Repeat Step 2-3 till stopped

Slide: Derek Hoiem

K-means clustering

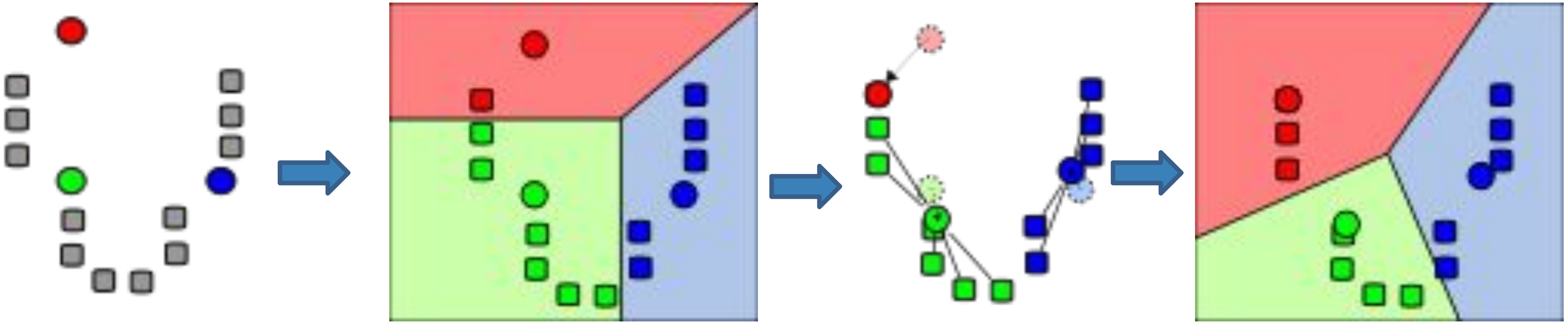
1. Initialize ($t = 0$): cluster centers $c_1, \ldots, c_K$
   - Commonly used: random initialization
   - Or greedily choose K to minimize residual
2. Compute δ^t : assign each point to the closest center
   - Typical distance measure:
     - Euclidean $sim(x, x') = x^T x'$
     - Cosine $sim(x, x') = x^T x' / (\|x\| \cdot \|x'\|)$
     - Others
3. Computer c^t : update cluster centers as the mean of the points

$$c^t = \underset{c}{\operatorname{argmin}} \frac{1}{N} \sum_j^N \sum_i^K \delta_{ij}^t \left(c_i^{t-1} - x_j\right)^2$$

4. Update $t = t + 1$, Repeat Step 2-3 till stopped
   - c^t doesn't change anymore.

Slide: Derek Hoiem

K-means clustering



1. Initialize Cluster Centers

2. Assign Points to Clusters

3. Re-compute Means

Repeat (2) and (3)

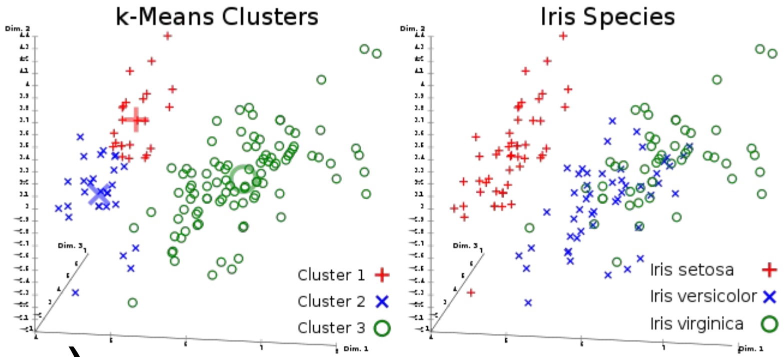
- Online demo:

https://kmeans.js.org/en/

K-means clustering

- Converges to a *local minimum* solution
  - Initialize multiple runs

- Better fit for spherical data

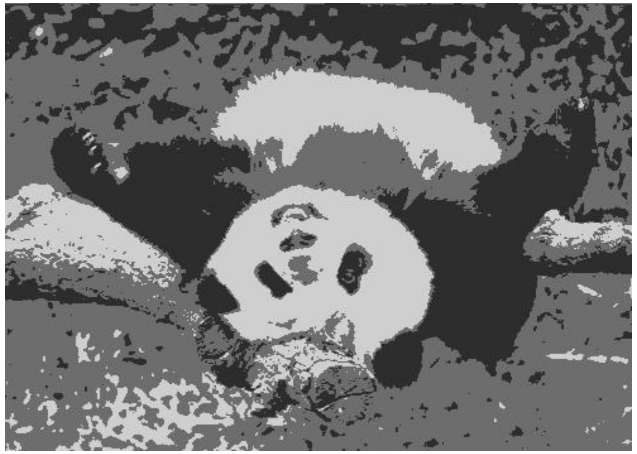


- Need to pick K (# of clusters)

Segmentation as Clustering

Original image

2 clusters



3 clusters

K-Means++

- Can we prevent arbitrarily bad local minima?

1. Randomly choose first center.
2. Pick new center with prob. proportional to $(x - c_i)^2$
   - (Contribution of *x* to total error)
3. Repeat until *K* centers.

- Expected error $= O(\log K)$ * optimal

Arthur & Vassilvitskii 2007

Slide credit: Steve Seitz

Feature Space

- Feature space: what measurements do we include in x_i?
- Depending on what we choose as the *feature space*, we can group pixels in different ways.

- Grouping pixels based on intensity similarity

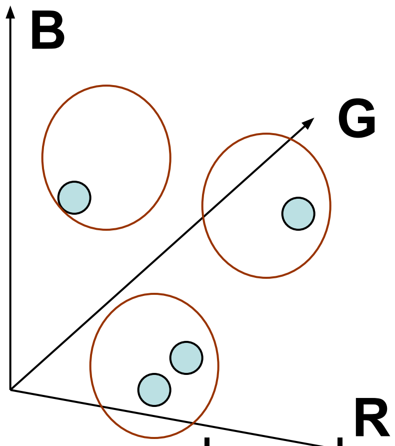
- Feature space: intensity value (1I

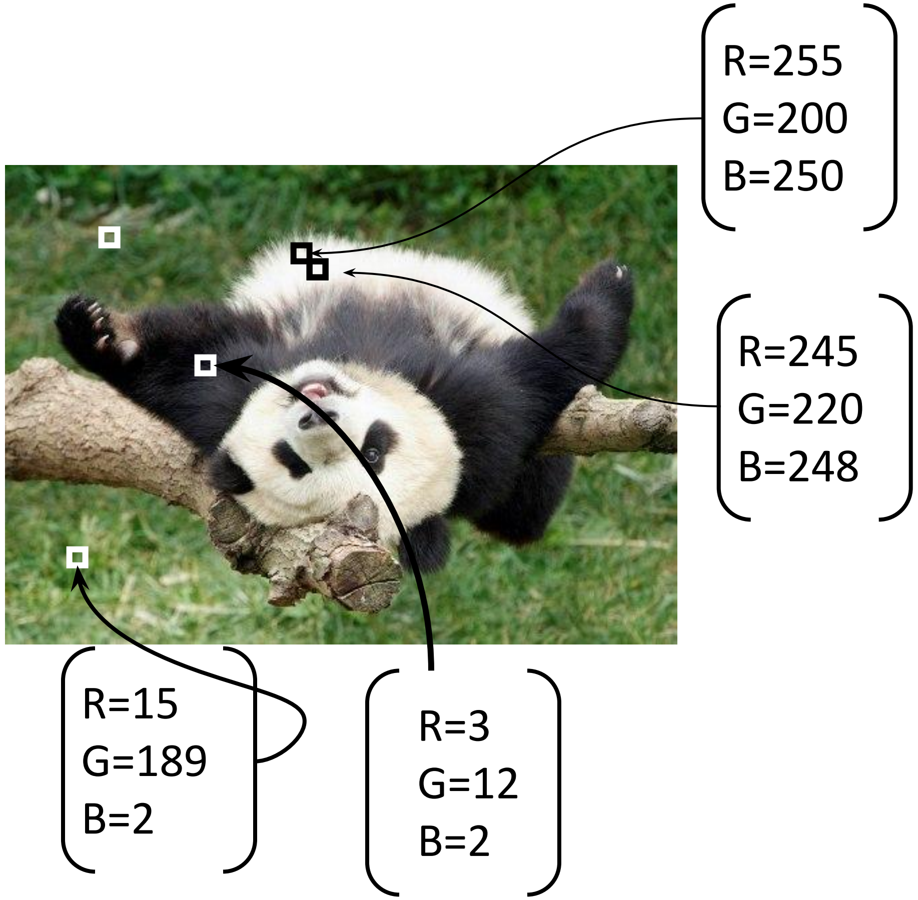
Slide credit: Kristen Grauman

Feature Space

- Depending on what we choose as the *feature space*, we can group pixels in different ways.

- Grouping pixels based on color similarity





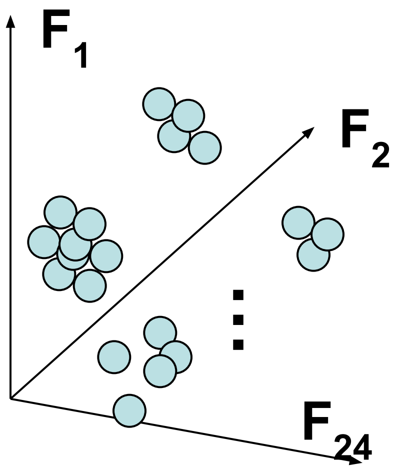
- Feature space: color value (3D)

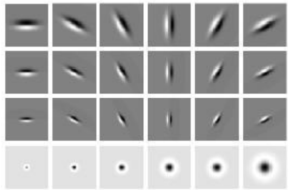
Slide credit: Kristen Grauman

Feature Space

- Depending on what we choose as the *feature space*, we can group pixels in different ways.

- Grouping pixels based on texture similarity





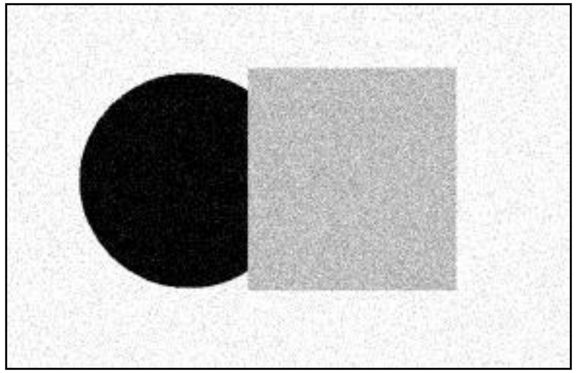
Filter bank of
24 filters

- Feature space: filter bank responses (e.g., 24D)

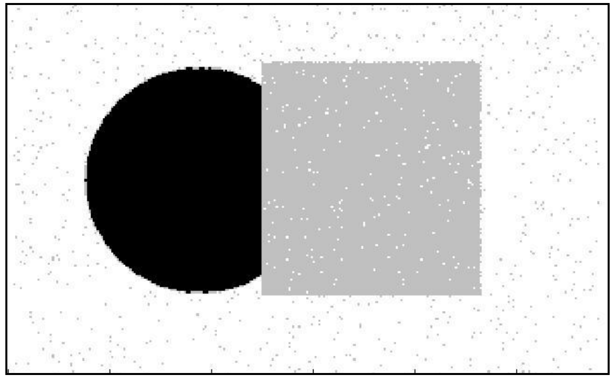
Slide credit: Kristen Grauman

Smoothing Out Cluster Assignments

- Assigning a cluster label per pixel may yield outliers:



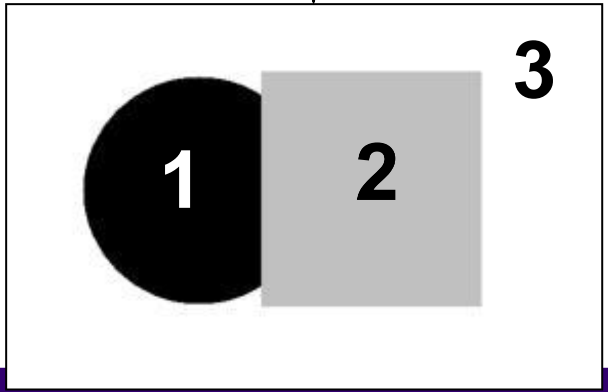
Original



Labeled by cluster center's intensity

?

- How can we ensure they are spatially smooth?

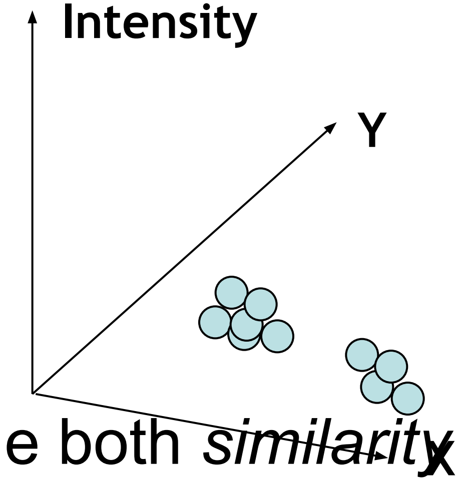


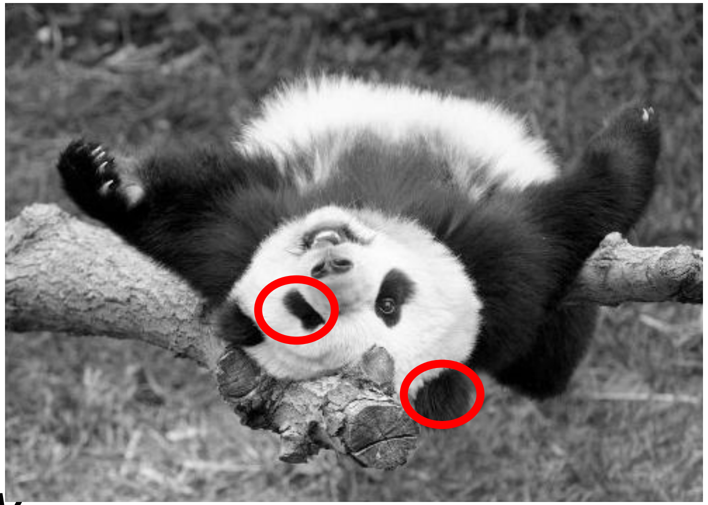
Slide credit: Kristen Grauman

Segmentation as Clustering

- Depending on what we choose as the *feature space*, we can group pixels in different ways.

- Grouping pixels based on *intensity+position* similarity





⇒ Way to encode both *similarity* and *proximity.*

Slide credit: Kristen Grauman

K-Means Clustering Results

- K-means clustering based on intensity or color is essentially vector quantization of the image attributes
  - Clusters don't have to be spatially coherent

Image

Intensity-based clusters

Color-based clusters

Image source: Forsyth & Ponce

K-Means Clustering Results

- K-means clustering based on intensity or color is essentially vector quantization of the image attributes
  - Clusters don't have to be spatially coherent
- Clustering based on (r,g,b,x,y) values enforces more spatial coherence

Image source: Forsyth & Ponce

How to evaluate clusters?

- Generative
  - How well are points reconstructed from the clusters?

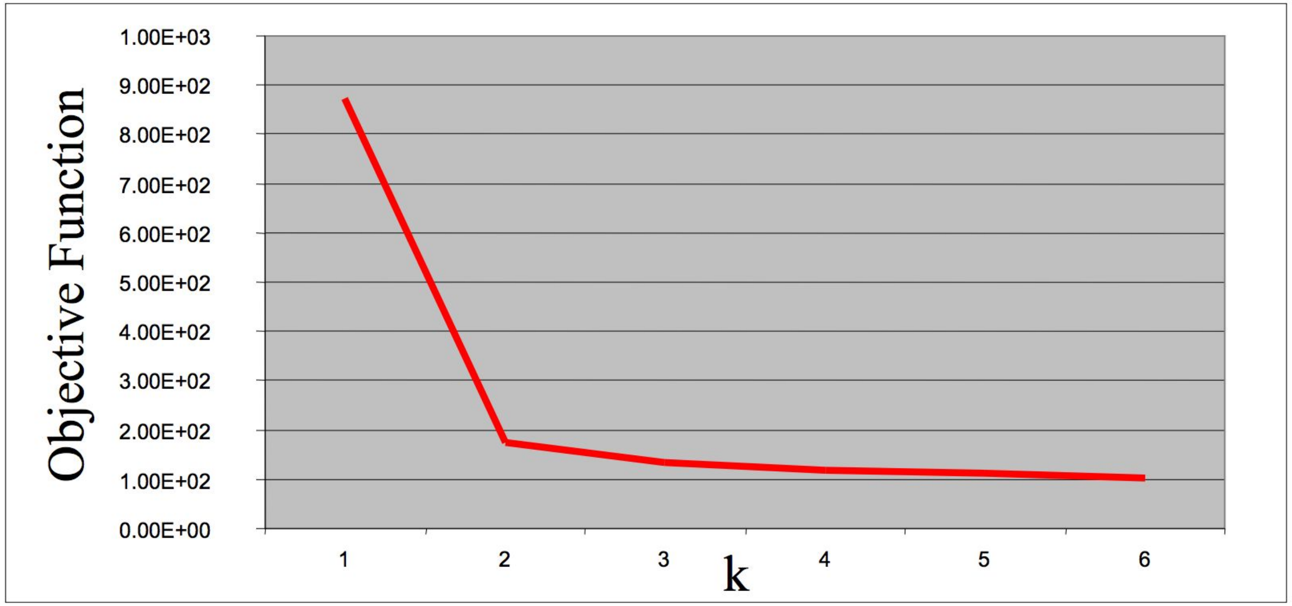
- Discriminative
  - How well do the clusters correspond to labels?
    - Can we correctly classify which pixels belong to the panda?
  - Note: unsupervised clustering does not aim to be discriminative as we don’t have the labels.

How to choose the number of clusters?

Try different numbers of clusters in a validation set and look at performance.

We can plot the objective function values for k equals 1 to 6…

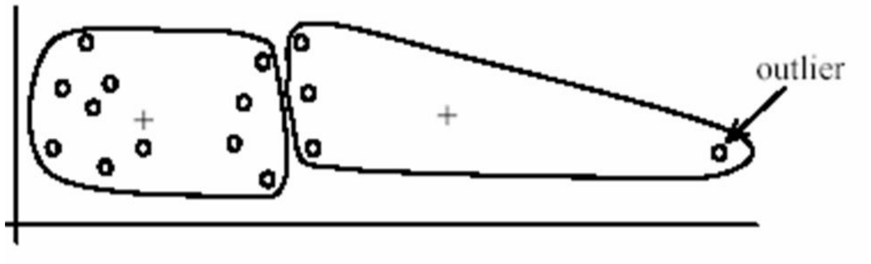
The abrupt change at k = 2, is highly suggestive of two clusters in the data. This technique for determining the number of clusters is known as “knee finding” or “elbow finding”.

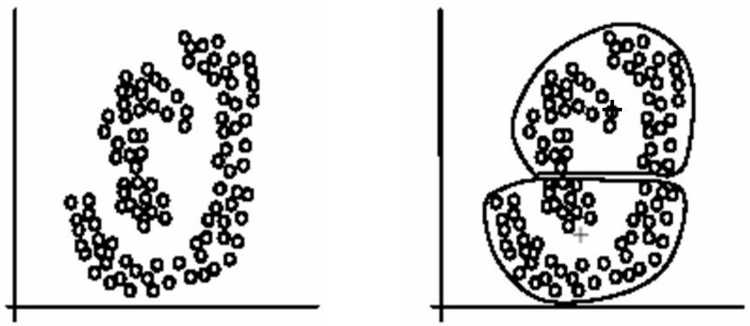


K-Means pros and cons

- Pros
  - Finds cluster centers that minimize conditional variance (good representation of data)
  - Simple and fast, Easy to implement
- Cons
  - Need to choose K
  - Sensitive to outliers
  - Prone to local minima
  - All clusters have the same parameters (e.g., distance measure is non-adaptive)
  - *Can be slow: each iteration is O(KNd) for N d-dimensional points
- Usage
  - Unsupervised clustering
  - Rarely used for pixel segmentation

(B): Ideal clusters





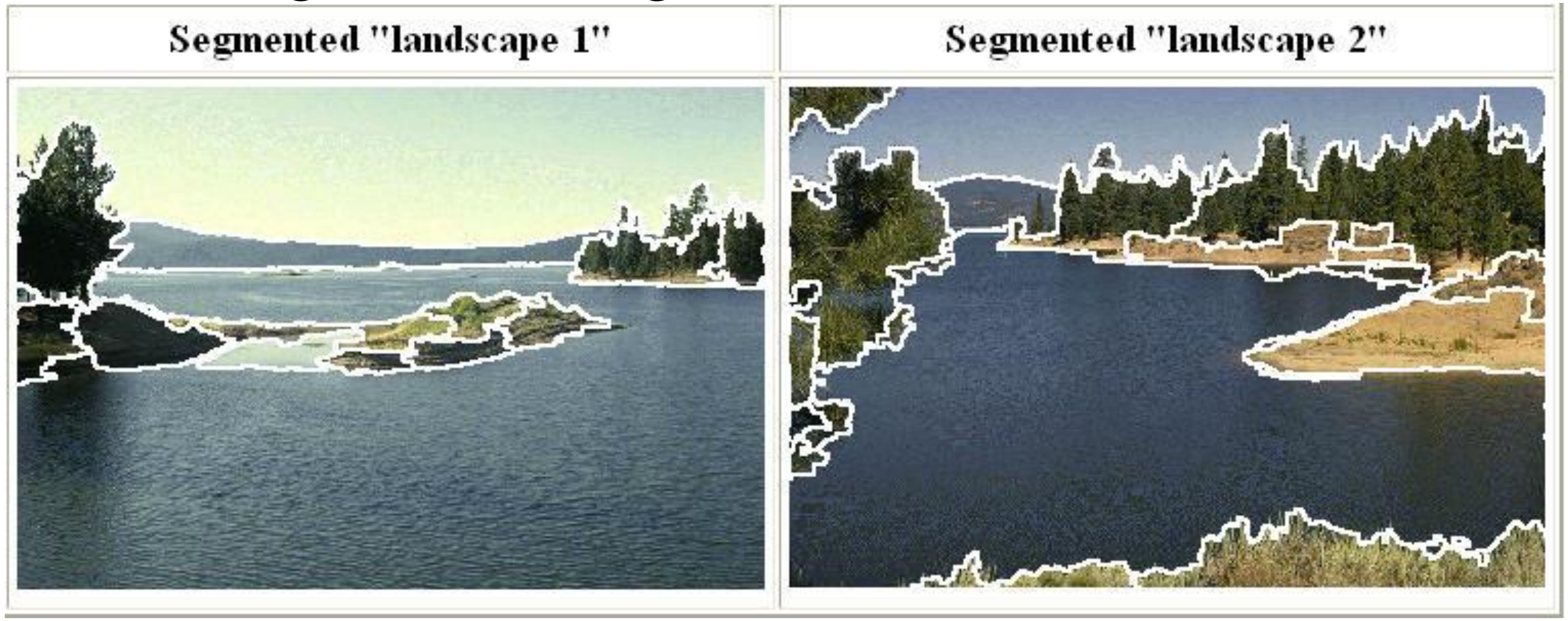
(A): Two natural clusters (B): *k*-means clusters

What will we learn today?

- K-means clustering
- **Mean-shift clustering**

Mean-Shift Segmentation

- An advanced and versatile technique for clustering-based segmentation

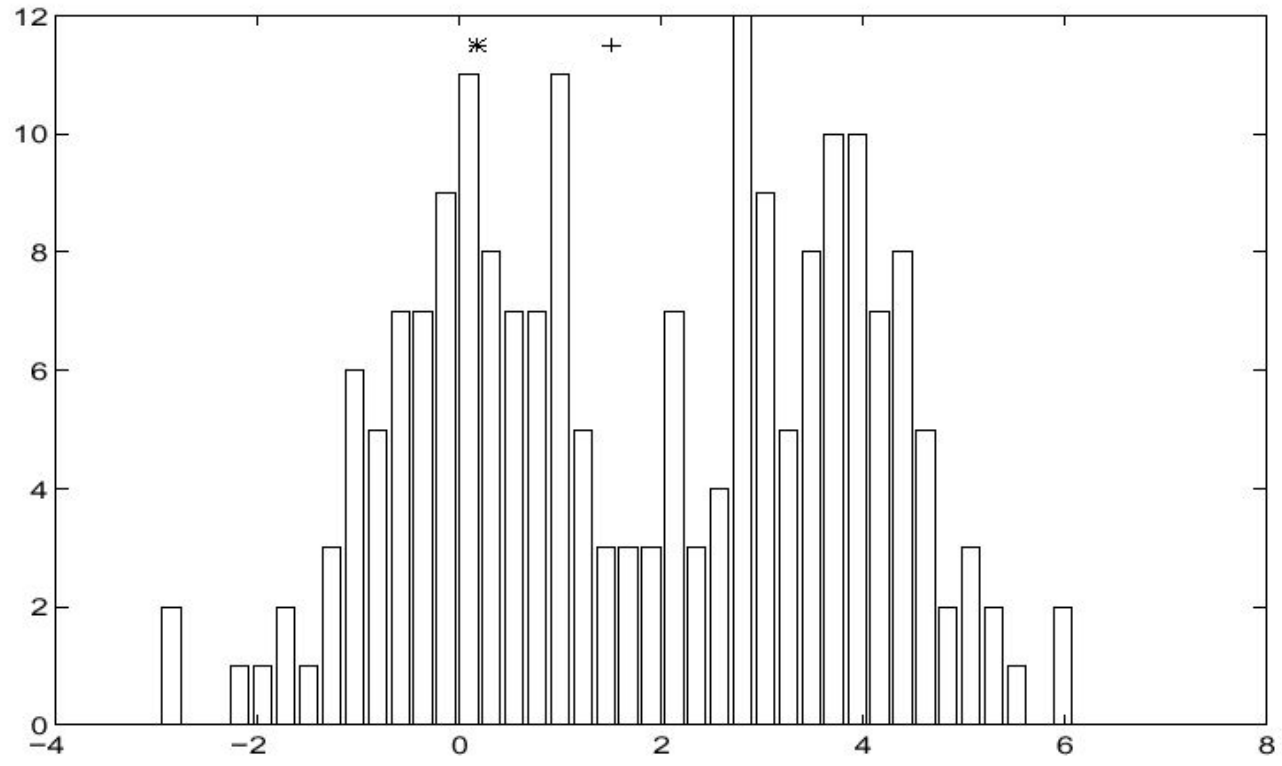


http://www.caip.rutgers.edu/~comanici/MSPAMI/msPamiResults.html

D. Comaniciu and P. Meer, Mean Shift: A Robust Approach toward Feature Space Analysis, PAMI 2002.

Slide credit: Svetlana Lazebnik

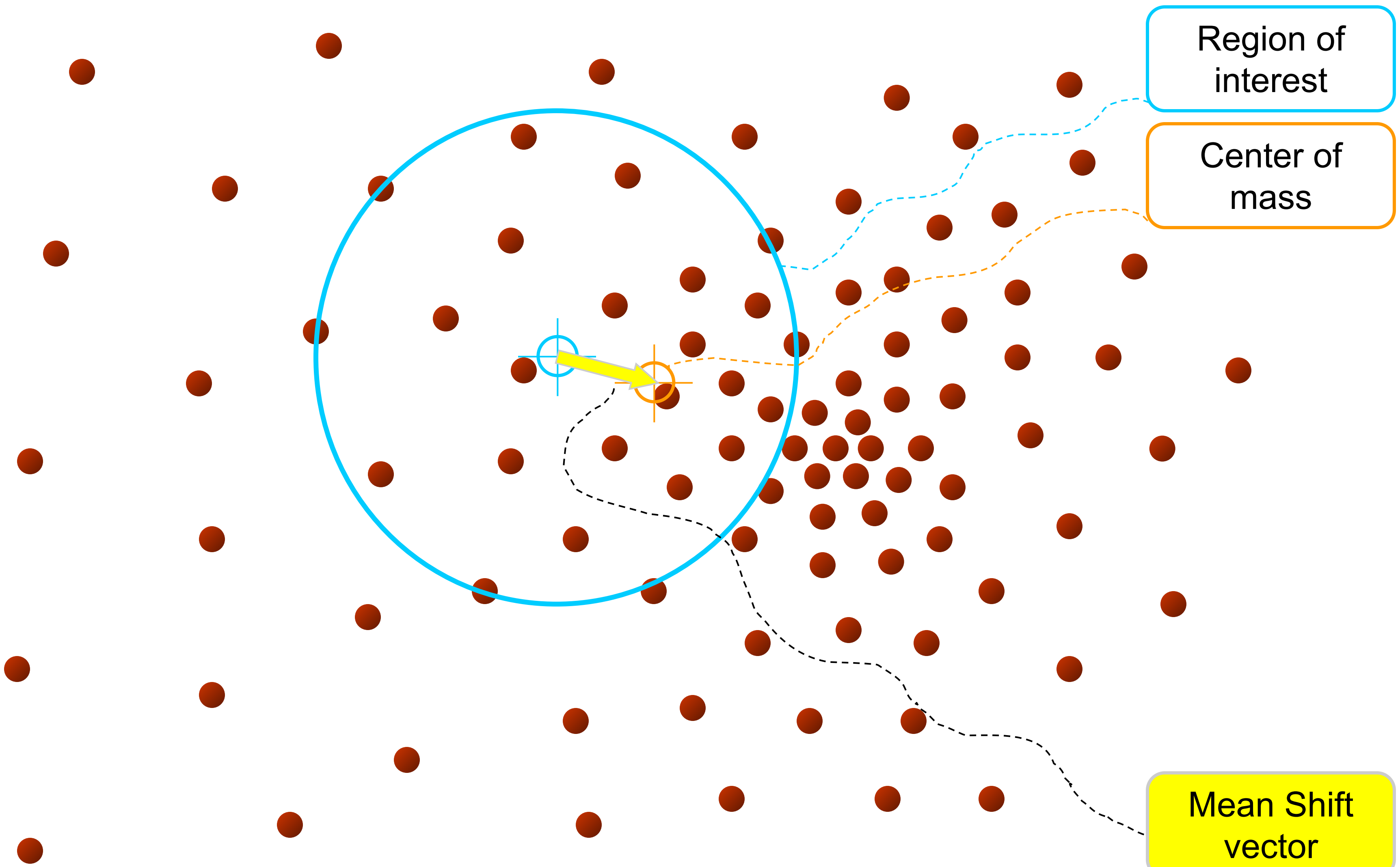
Mean-Shift Algorithm



- Iterative Mode S
    1. Initialize random seed, and window W
    2. Calculate center of gravity (the "mean") of W: $\sum_{x \in W} x H(x)$
    3. Shift the search window to the mean
    4. Repeat Step 2 until convergence

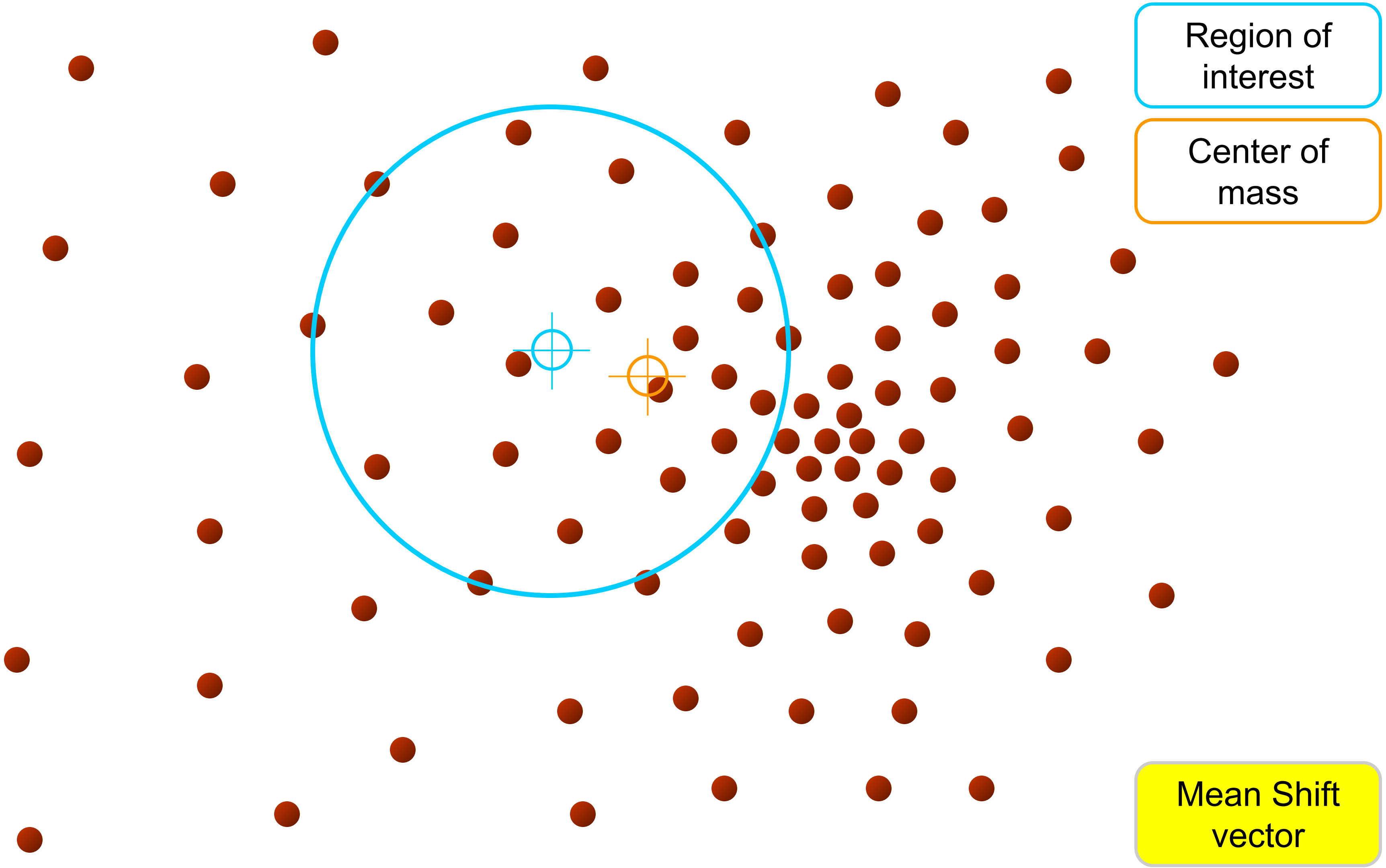
Slide credit: Steve Seitz

Mean-Shift



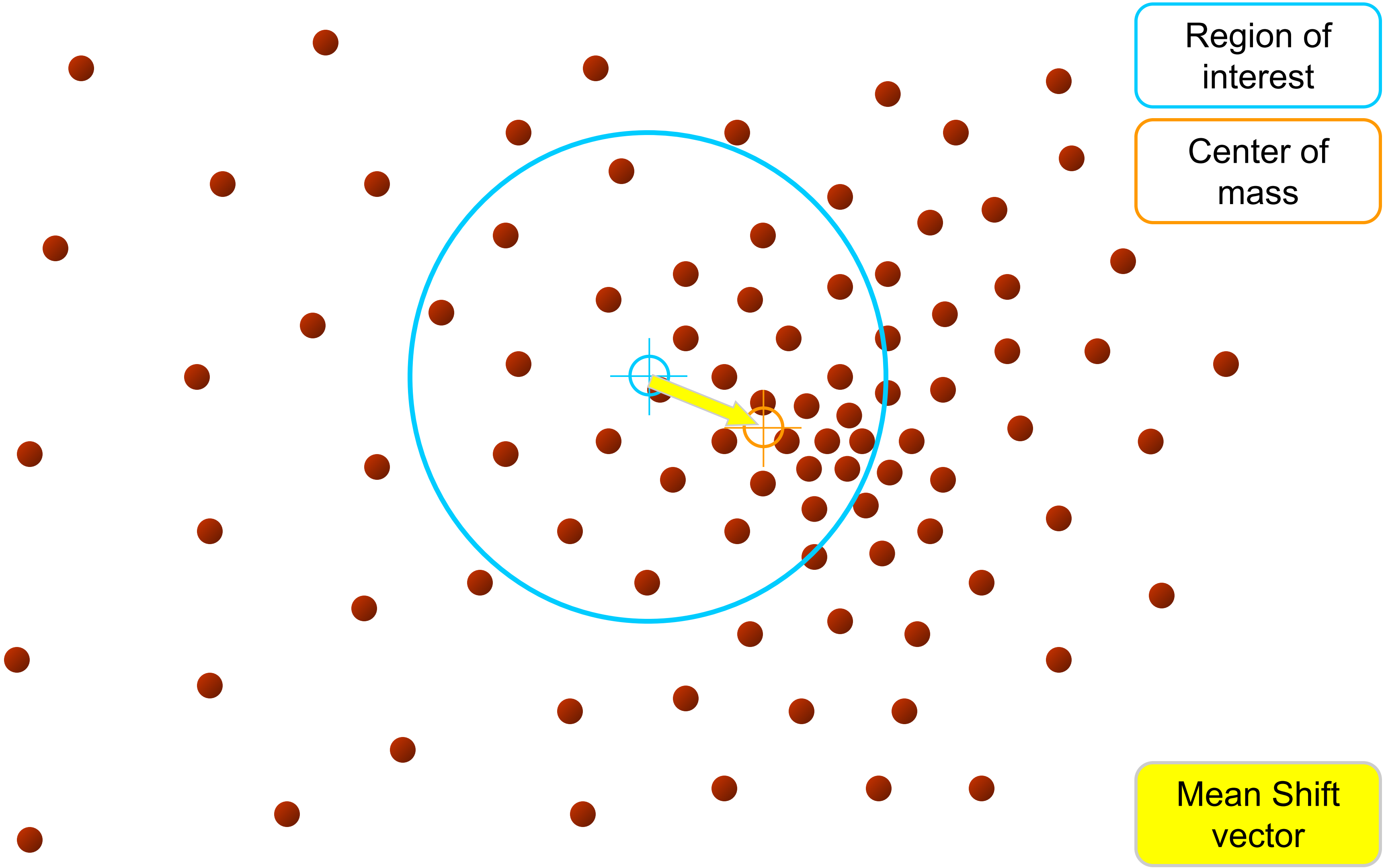
Slide by Y. Ukrainitz & B. Sarel

Mean-Shift



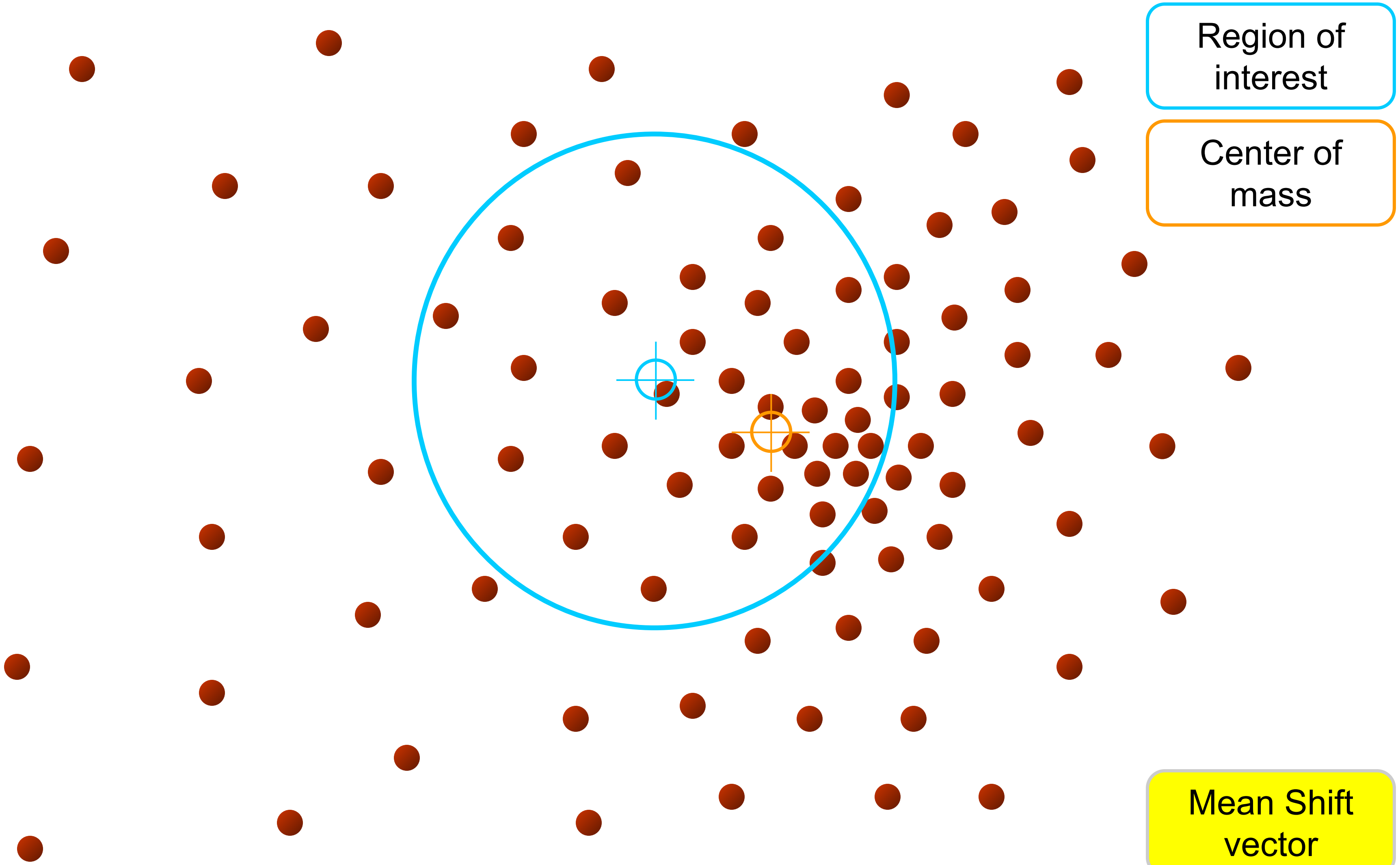
Slide by Y. Ukrainitz & B. Sarel

Mean-Shift



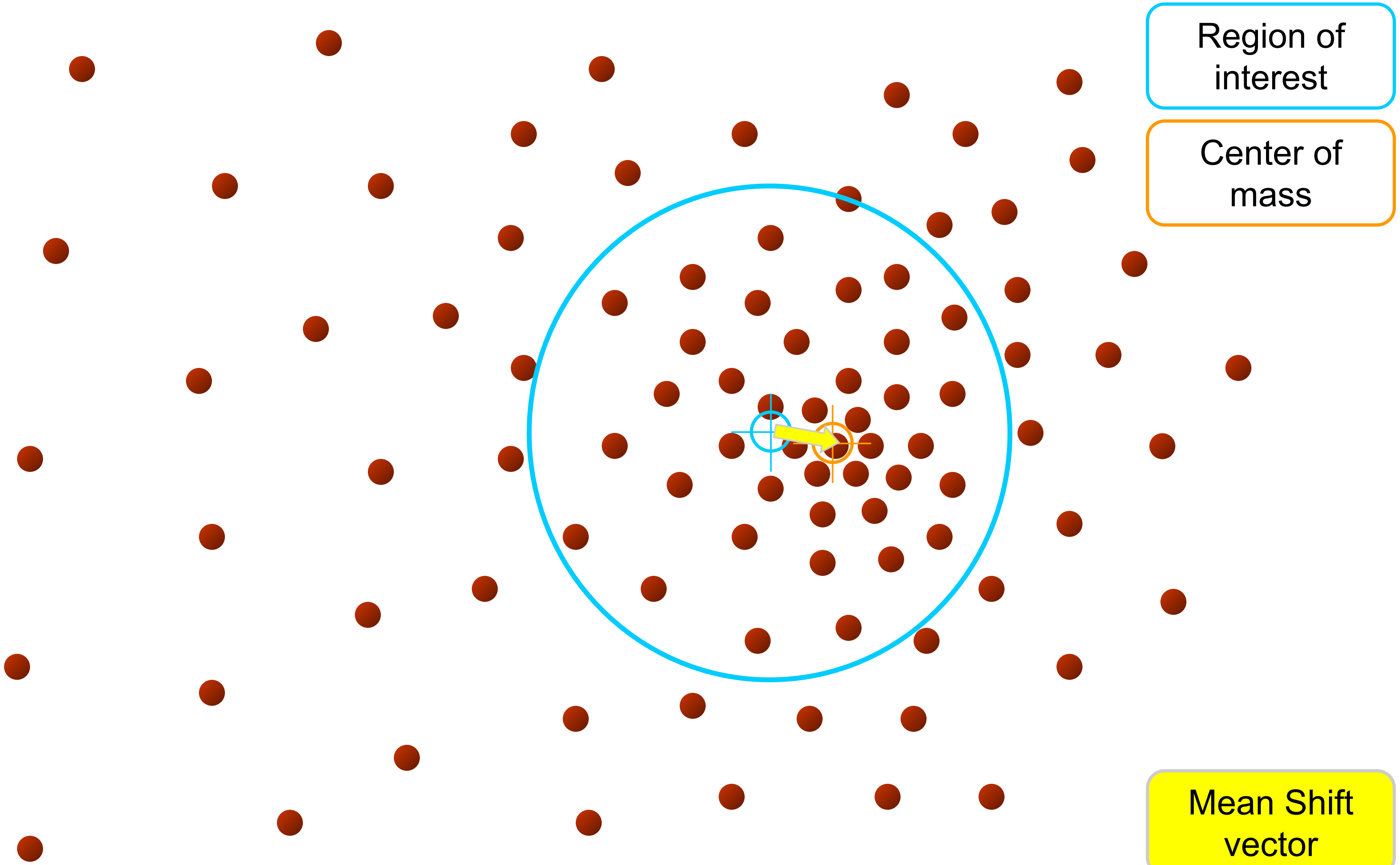
Slide by Y. Ukrainitz & B. Sarel

Mean-Shift



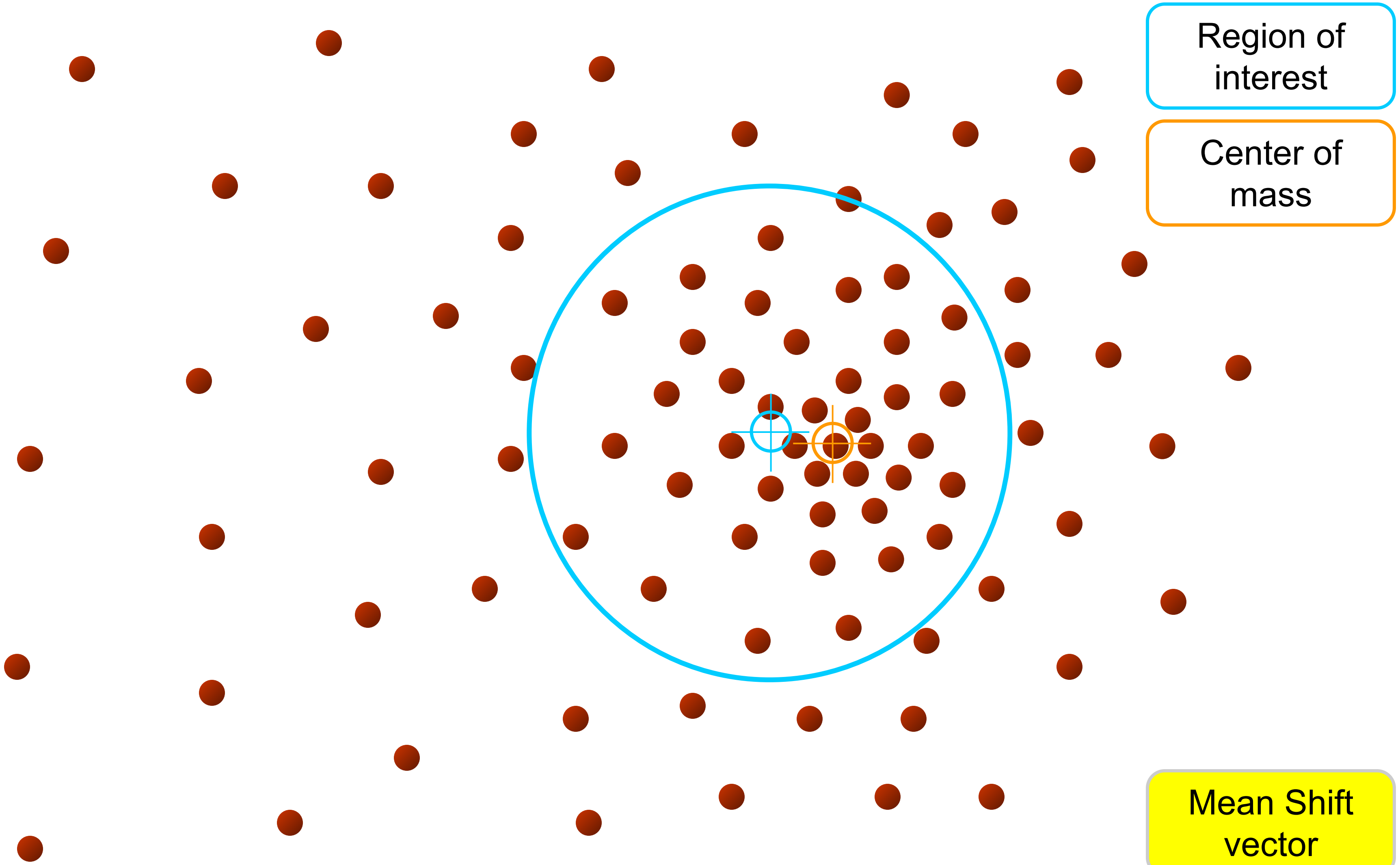
Slide by Y. Ukrainitz & B. Sarel

Mean-Shift



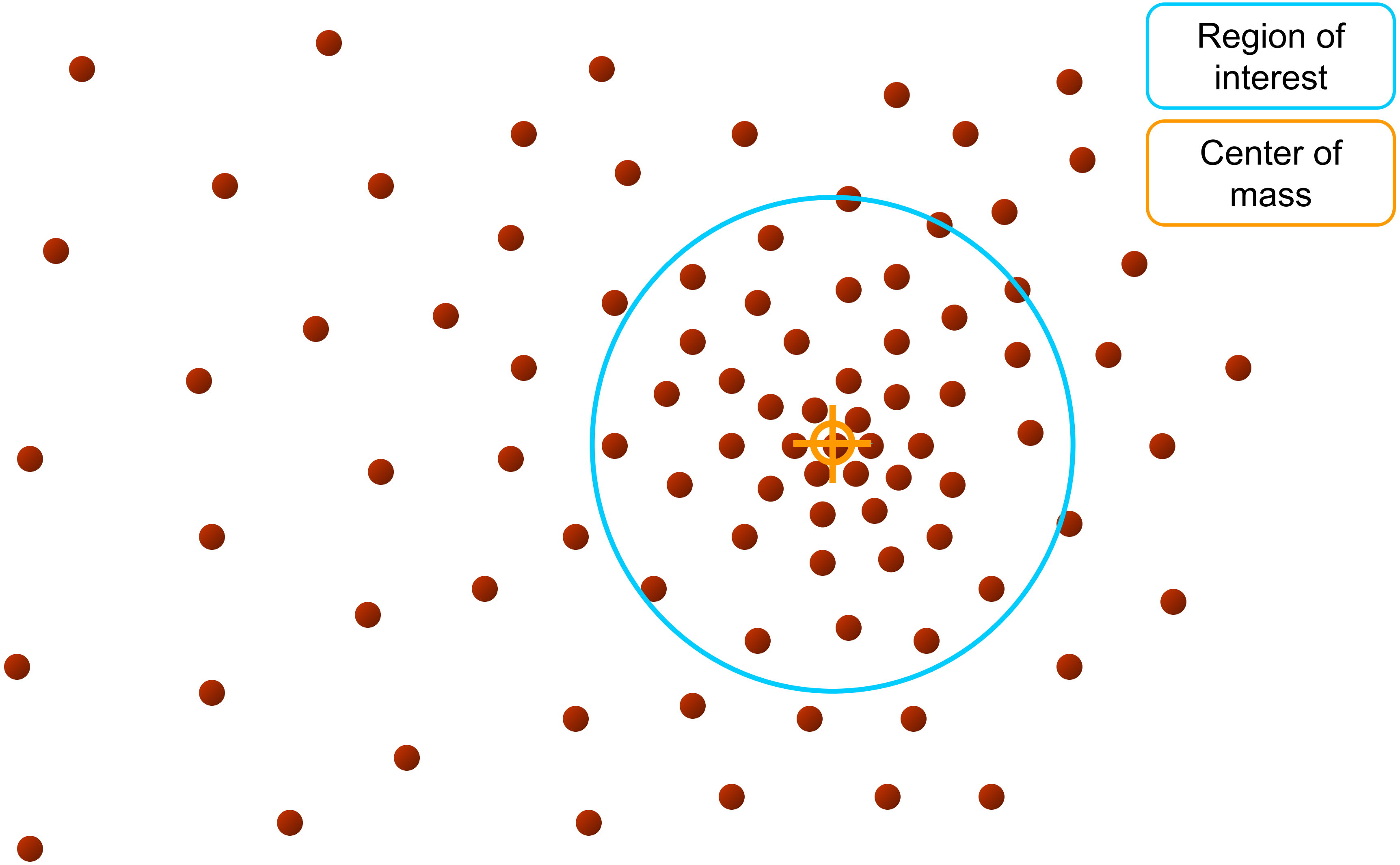
Slide by Y. Ukrainitz & B. Sarel

Mean-Shift



Slide by Y. Ukrainitz & B. Sarel

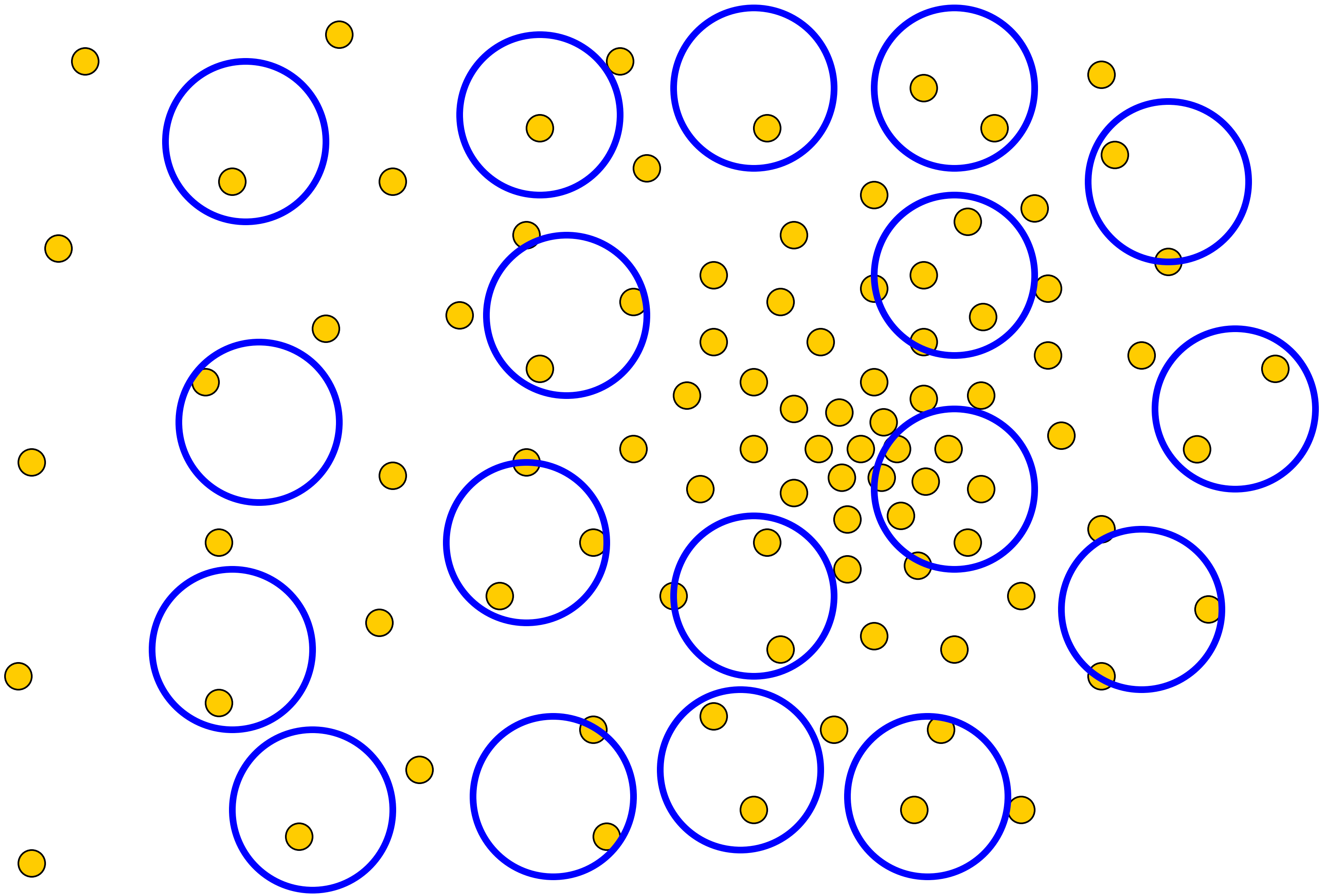
Mean-Shift



Slide by Y. Ukrainitz & B. Sarel

26-Oct-17

Real Modality Analysis

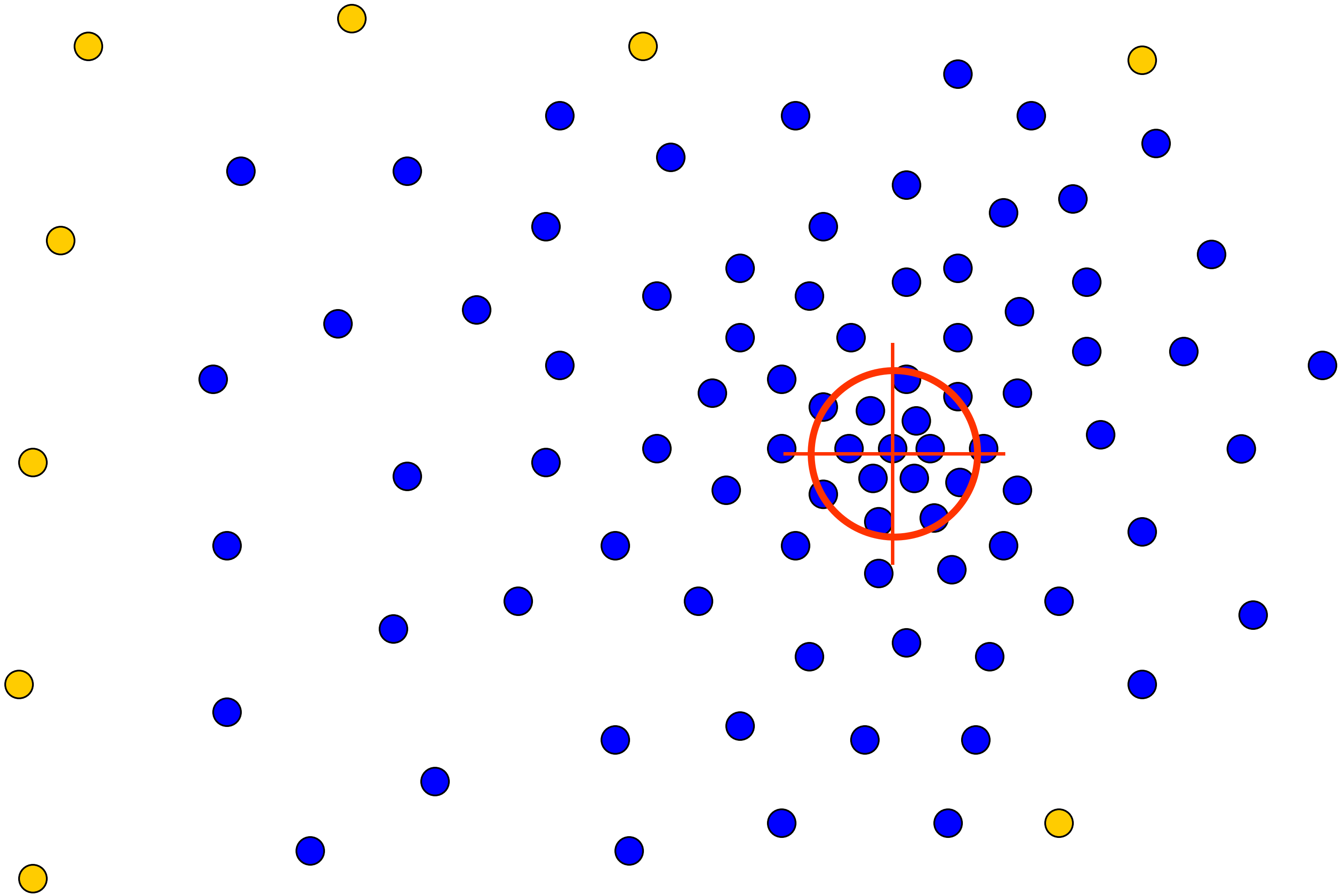


Tessellate the space with windows

Run the procedure in parallel

Slide by Y. Ukrainitz & B. Sarel

Real Modality Analysis

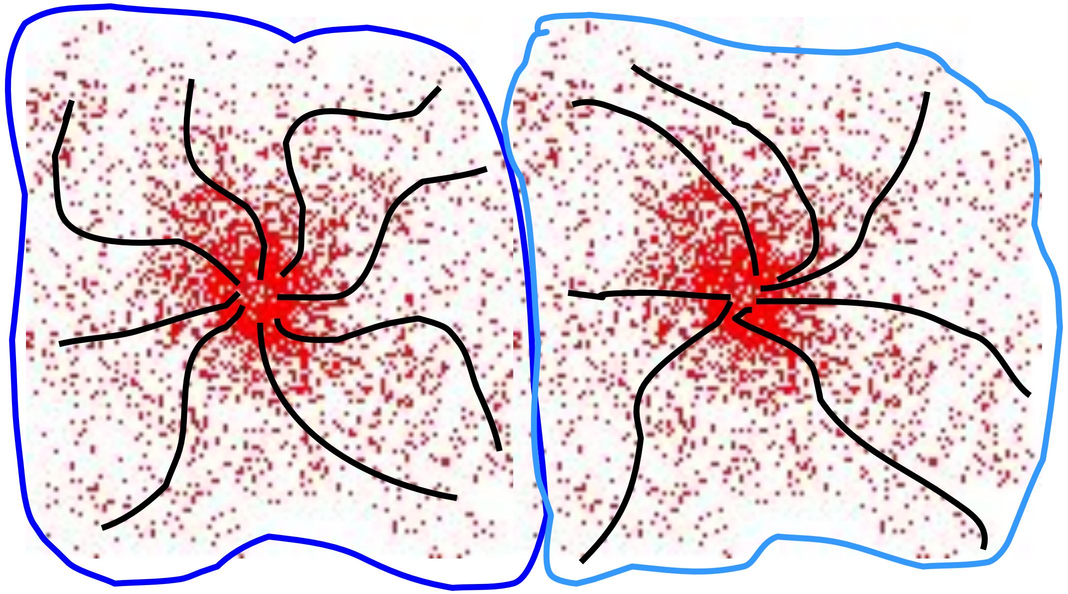


The blue data points were traversed by the windows towards the mode.

Slide by Y. Ukrainitz & B. Sarel

Mean-Shift Clustering

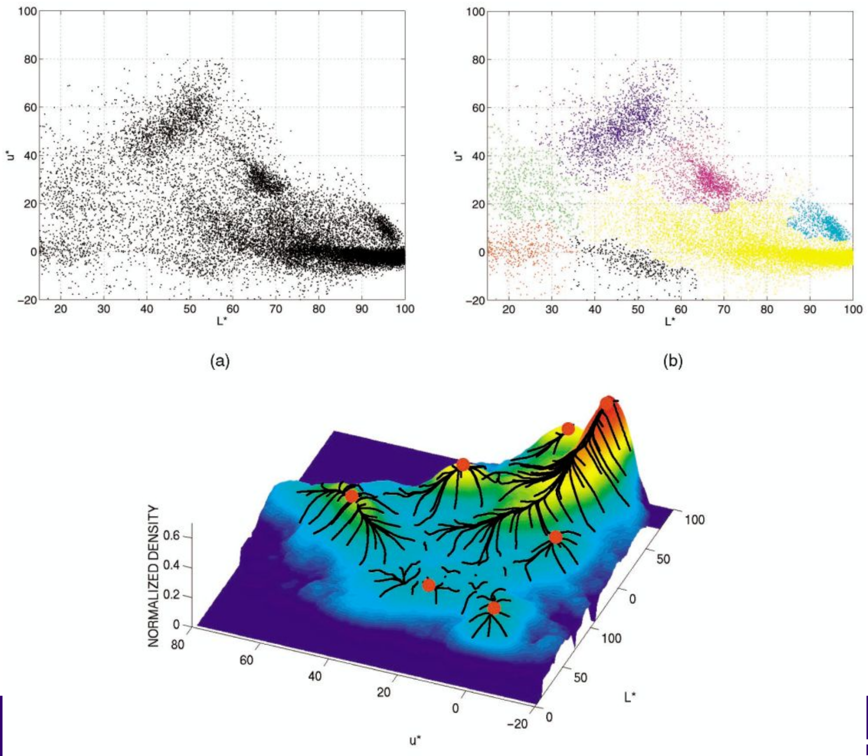
- Cluster: all data points in the attraction basin of a mode
- Attraction basin: the region for which all trajectories lead to the same mode



Slide by Y. Ukrainitz & B. Sarel

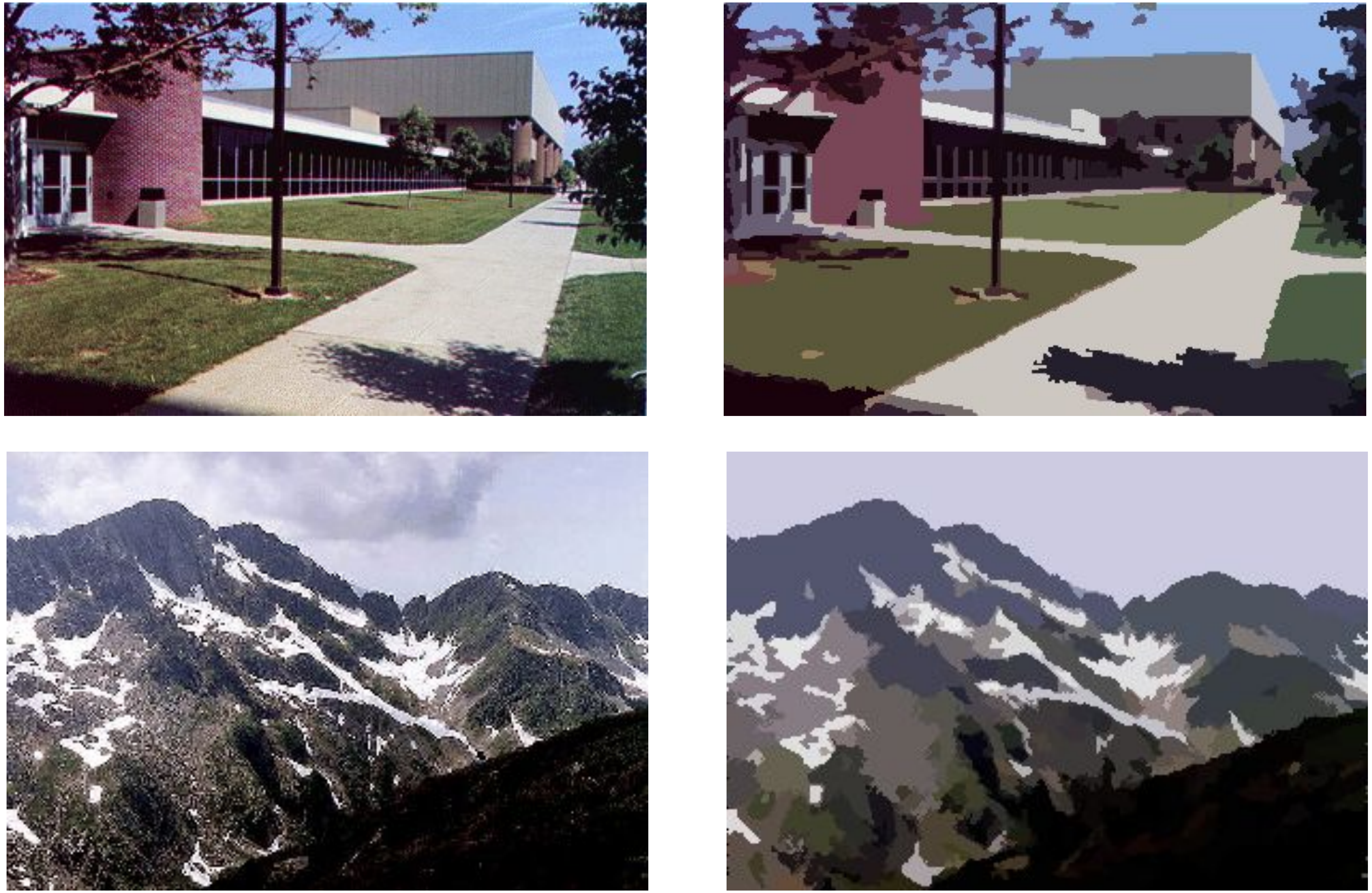
Mean-Shift Clustering/Segmentation

- 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



Slide credit: Svetlana Lazebnik

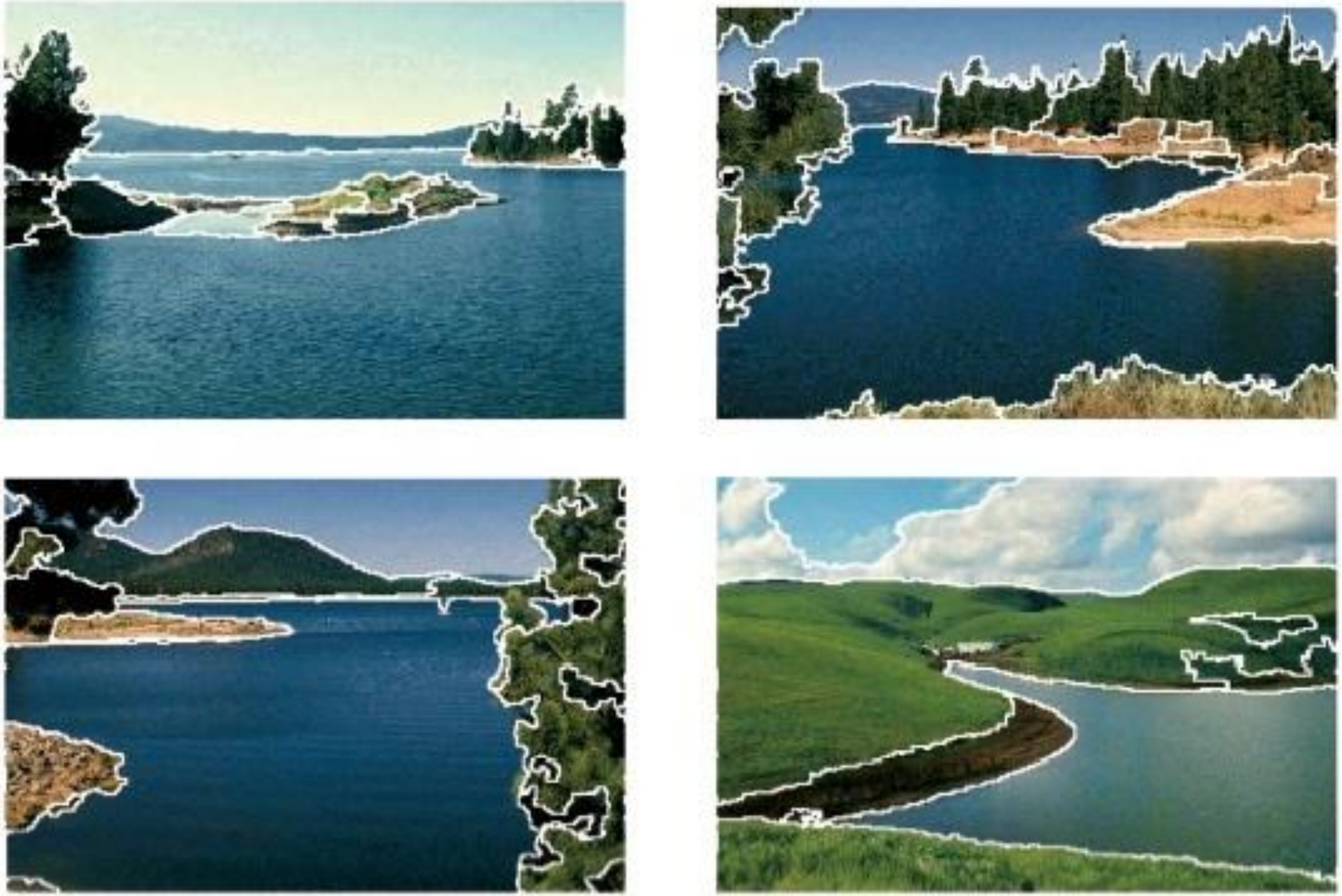
Mean-Shift Segmentation Results



Slide credit: Svetlana Lazebnik

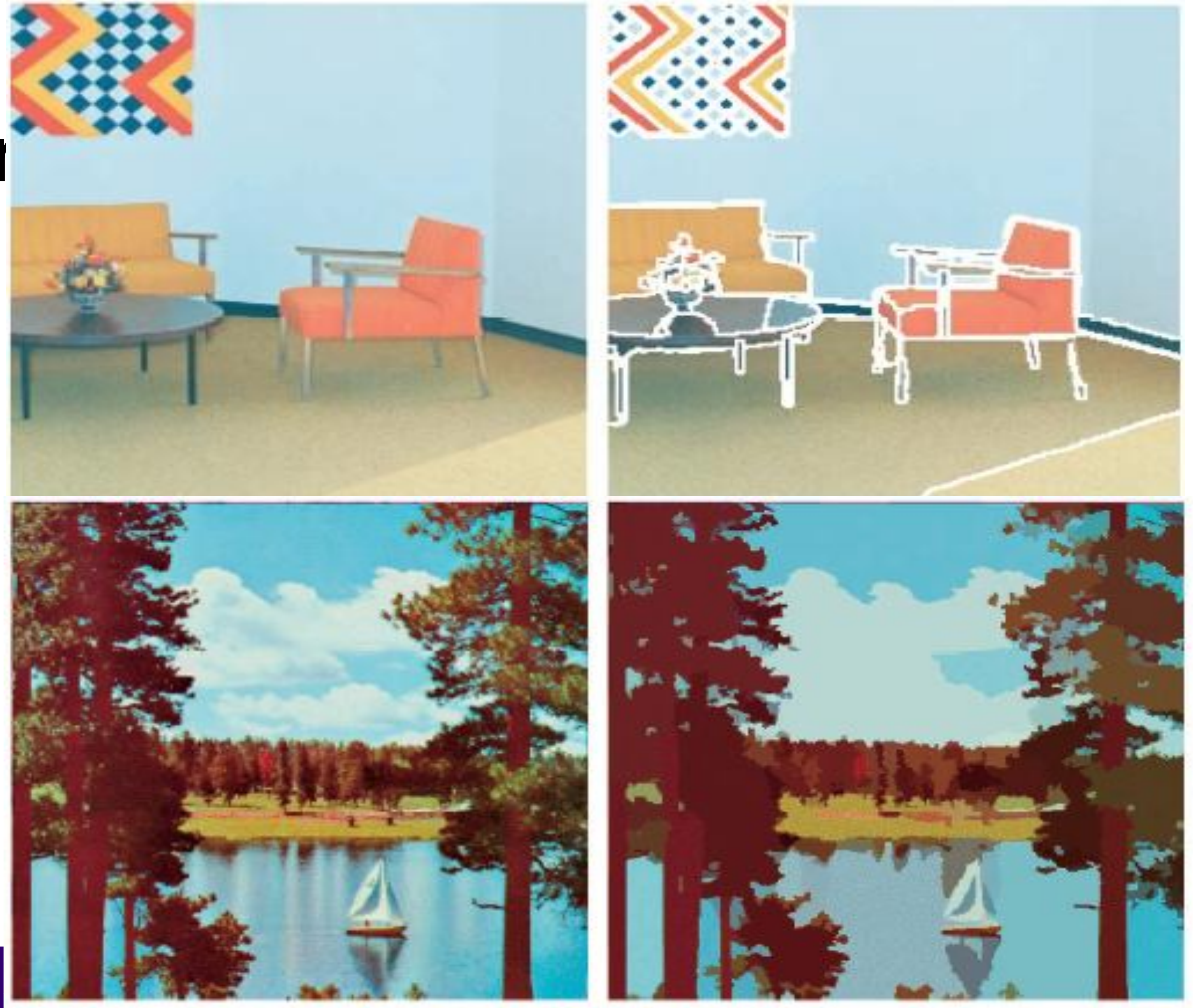
http://www.caip.rutgers.edu/~comanici/MSPAMI/msPamiResults.html

More Results

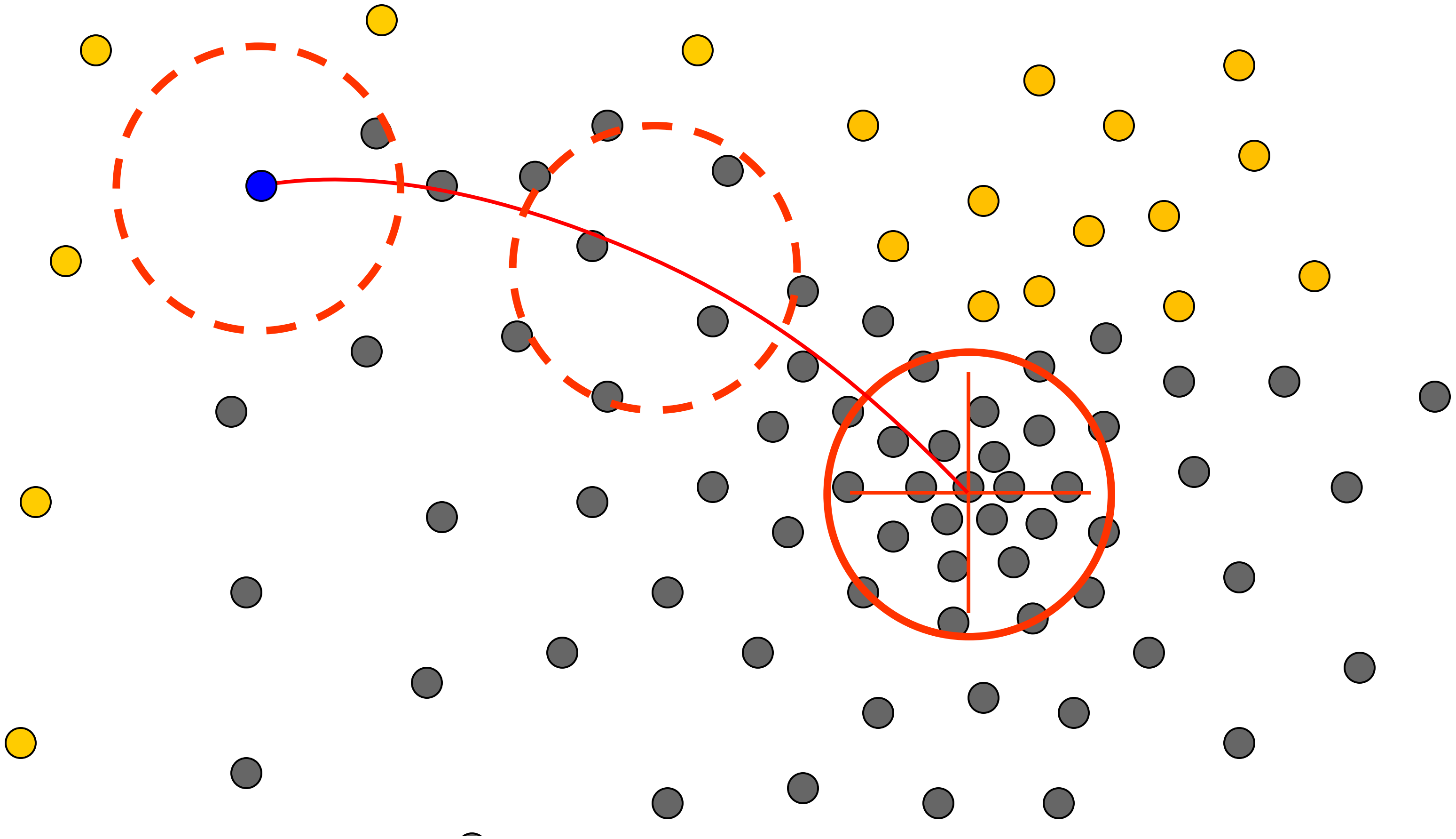


Slide credit: Svetlana Lazebnik

Mor



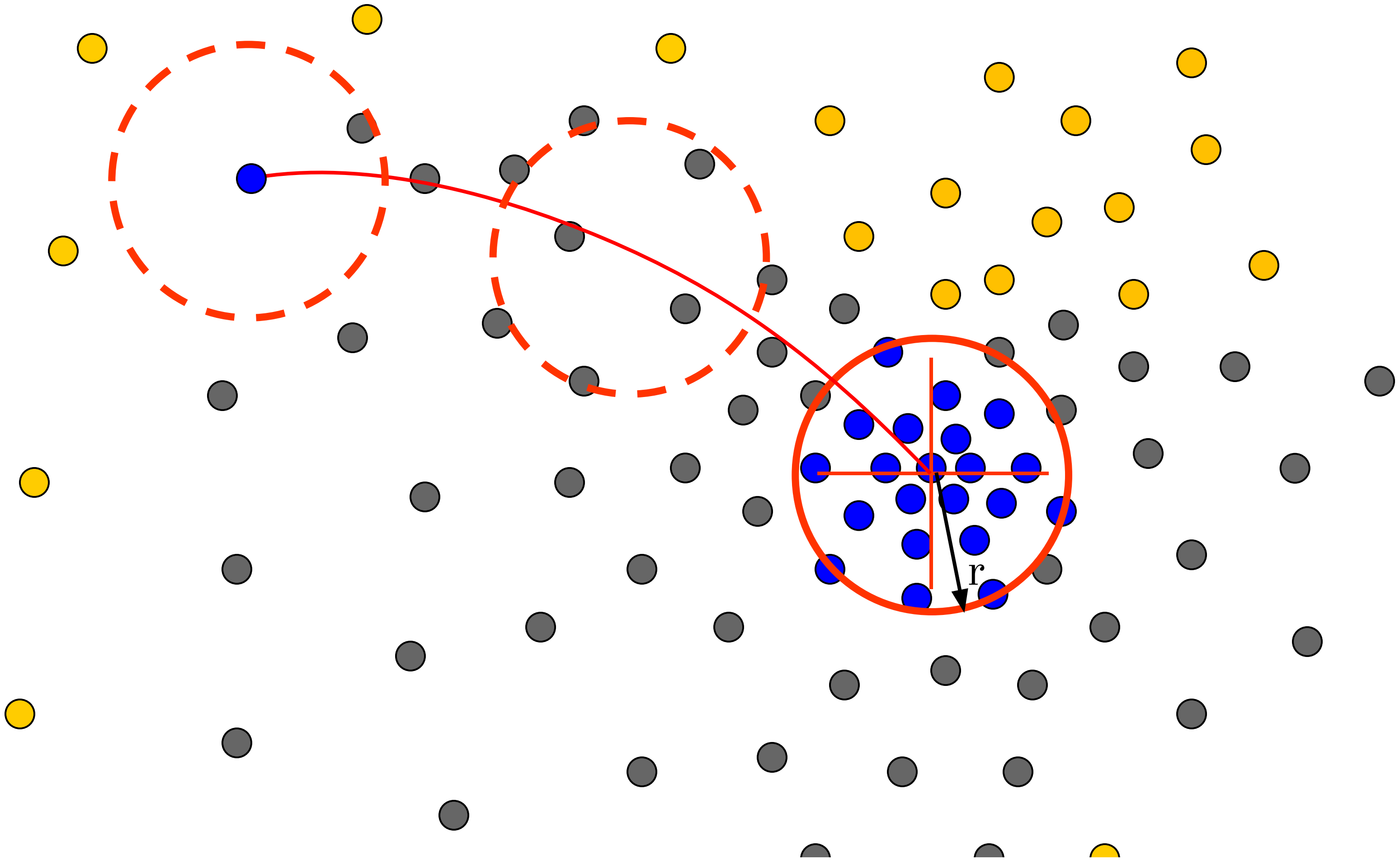
Problem: Computational Complexity



- Need to shift many windows…
- Many computations will be redundant.

Slide credit: Bastian Leibe

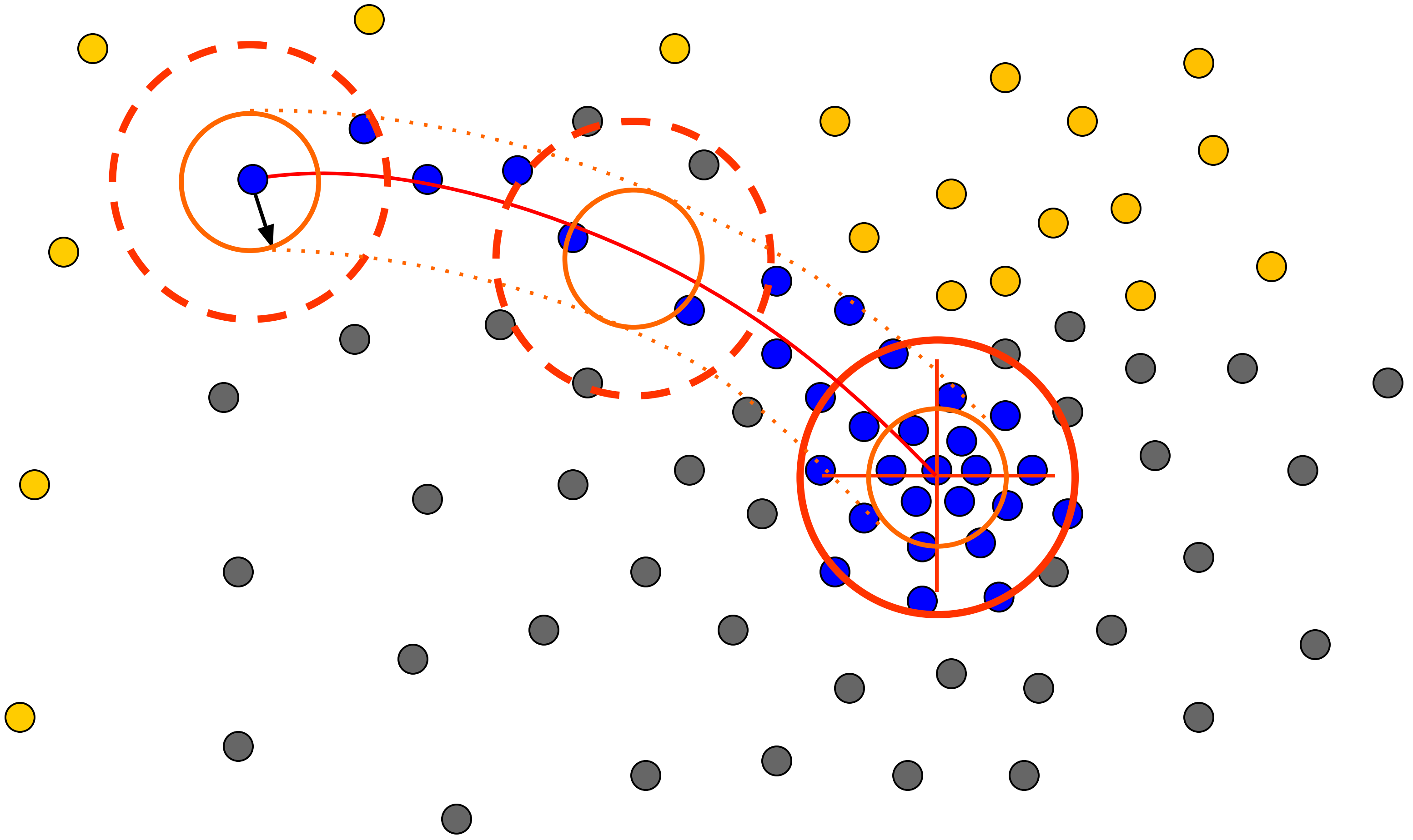
Speedups: Basin of Attraction



1. Assign all points within radius r of end point to the mode.

Slide credit: Bastian Leibe

Speedups



2. Assign all points within radius r/c of the search path to the mode -> reduce the number of data points to search.

Slide credit: Bastian Leibe

26-Oct-17

Technical Details

Given n data points $\mathbf{x}_i \in \mathbb{R}^d$, the multivariate kernel density estimate using a radially symmetric kernel[1] (e.g., Epanechnikov and Gaussian kernels), $K(\mathbf{x})$, is given by,

$$\hat{f}_K = \frac{1}{nh^d} \sum_{i=1}^{n} K\left(\frac{\mathbf{x} - \mathbf{x}_i}{h}\right), \quad (1)$$

where h (termed the *bandwidth* parameter) defines the radius of kernel. The radially symmetric kernel is defined as,

$$K(\mathbf{x}) = c_k k(\|\mathbf{x}\|^2), \quad (2)$$

where c_k represents a normalization constant.

Comaniciu & Meer, 2002

Other Kernels

A kernel is a function that satisfies the following requirements :

1. $\int_{R^d} \phi(x) = 1$
2. $\phi(x) \geq 0$

Some examples of kernels include :

1. Rectangular $\phi(x) = \begin{cases} 1 & a \leq x \leq b \\ 0 & else \end{cases}$
2. Gaussian $\phi(x) = e^{-\frac{x^2}{2\sigma^2}}$
3. Epanechnikov $\phi(x) = \begin{cases} \frac{3}{4}(1 - x^2) & if\ |x| \leq 1 \\ 0 & else \end{cases}$

source

Technical Details

Taking the derivative of: $\hat{f}_K = \frac{1}{nh^d} \sum_{i=1}^{n} K\left(\frac{\mathbf{x} - \mathbf{x}_i}{h}\right)$

$$\nabla \hat{f}(\mathbf{x}) = \underbrace{\frac{2c_{k,d}}{nh^{d+2}} \left[\sum_{i=1}^{n} g\left(\left\|\frac{\mathbf{x} - \mathbf{x}_i}{h}\right\|^2\right)\right]}_{\text{term 1}} \underbrace{\left[\frac{\sum_{i=1}^{n} \mathbf{x}_i g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right)}{\sum_{i=1}^{n} g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right)} - \mathbf{x}\right]}_{\text{term 2}}, \qquad (3)$$

where $g(x) = -k'(x)$ denotes the derivative of the selected kernel profile.

- Term1: this is proportional to the density estimate at x (similar to equation 1 from two slides ago).
- Term2: this is the mean-shift vector that points towards the direction of maximum density.

Comaniciu & Meer, 2002

Technical Details

Finally, the mean shift procedure from a given point x_t is:

1. Compute the mean shift vector **m**:

$$\left[\frac{\sum_{i=1}^{n} \mathbf{x}_i g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right)}{\sum_{i=1}^{n} g\left(\left\|\frac{\mathbf{x}-\mathbf{x}_i}{h}\right\|^2\right)} - \mathbf{x}\right]$$

2. Translate the density window:

$$\mathbf{x}_i^{t+1} = \mathbf{x}_i^t + \mathbf{m}(\mathbf{x}_i^t).$$

3. Iterate steps 1 and 2 until convergence.

$$\nabla f(\mathbf{x}_i) = 0.$$

Comaniciu & Meer, 2002

Mean-Shift pros and cons

- Pros
  - General, application-independent tool
  - Model-free, does not assume any prior shape (spherical, elliptical, etc.) on data clusters
  - Just a single parameter (window size h)
    - h has a physical meaning (unlike k-means)
  - Finds variable number of modes
  - Robust to outliers
- Cons
  - Output depends on window size
  - Window size (bandwidth) selection is not trivial
  - Computationally (relatively) expensive (~2s/image)
  - Does not scale well with dimension of feature space

Slide credit: Svetlana Lazebnik

Summary

- K-means clustering
- Mean-shift clustering