中山大學
SUN YAT-SEN UNIVERSITY

Lecture 6.2

Segmentation

Pattern Recognition (in Computer Vision)

Jinhua Ma,
School of Computer Science and Engineering, Sun Yat-Sen University

(Tentative) Schedule

- L1. Introduction to PR
- L2. Images and Transformations
- L3. Color and Filters
- L4. Features and Fitting
- L5. Feature Descriptors
- **L6. Clustering and Segmentation**
- L7. Dimensionality Reduction
- L8. Face identification
- L9. Bayesian Decision Theory
- L10. Image Classification
- L11. Regularization and Optimization
- L12. Image Classification with CNNs
- L13. CNN Architectures
- L14. Training Neural Networks
- L15. Object Detection and Image Segmentation
- L16. Recurrent Neural Networks
- L17. Attention and Transformers
- L18. Generative Models
- L19. Self-supervised Learning

What we will learn today?

- Introduction to segmentation and clustering
- K-means clustering
- Mean-shift clustering

Image Segmentation

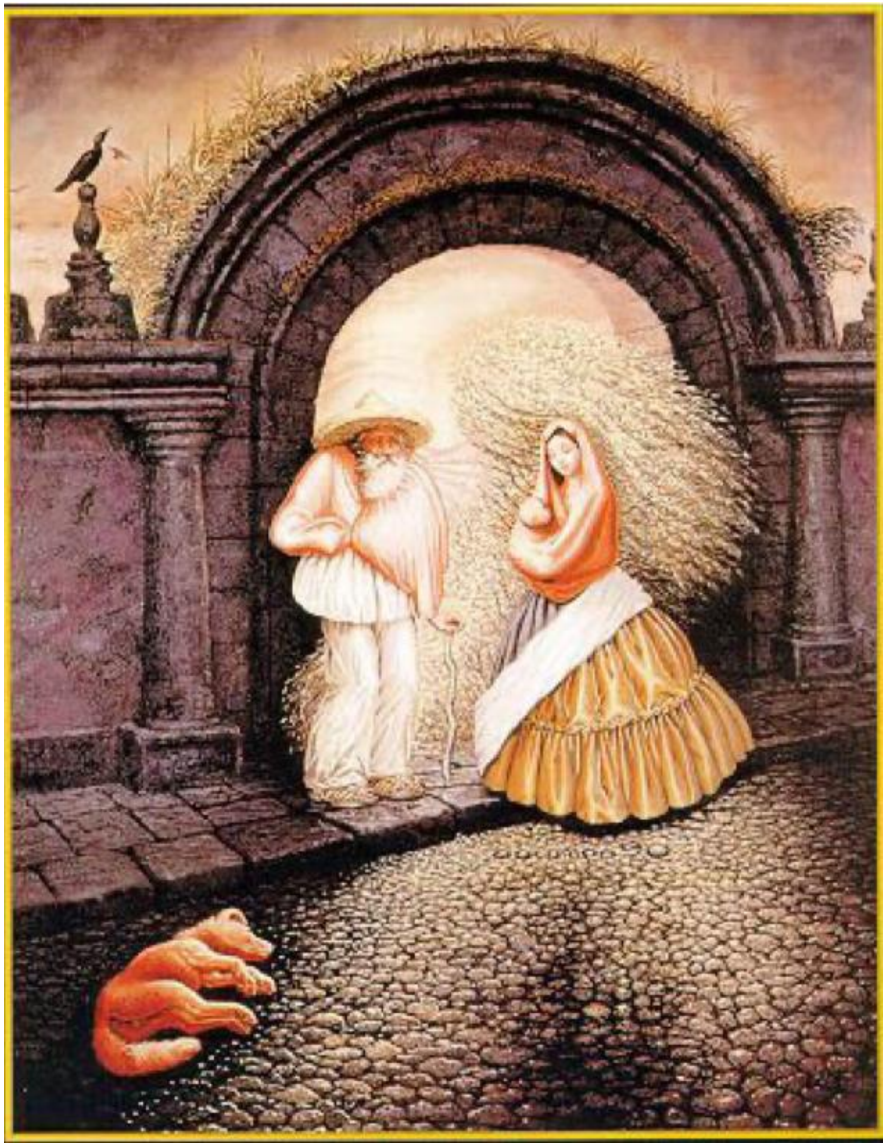
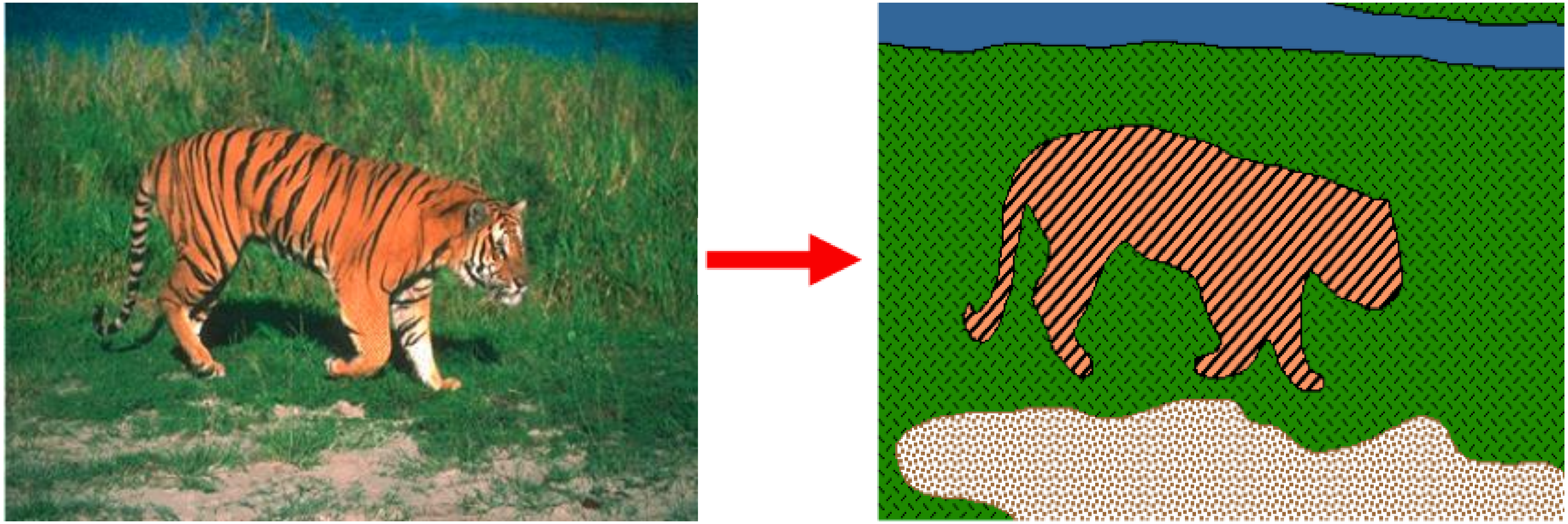


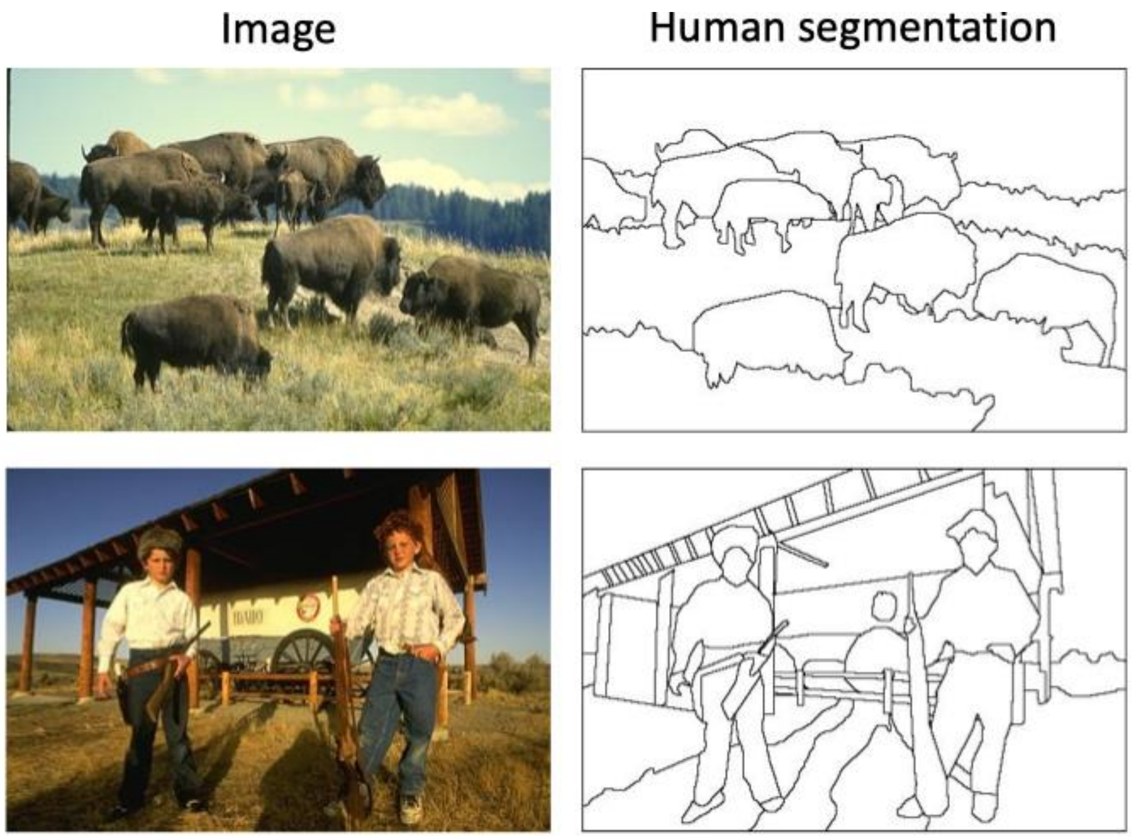
Image Segmentation

- Goal: identify groups of pixels that go together



The Goals of Segmentation

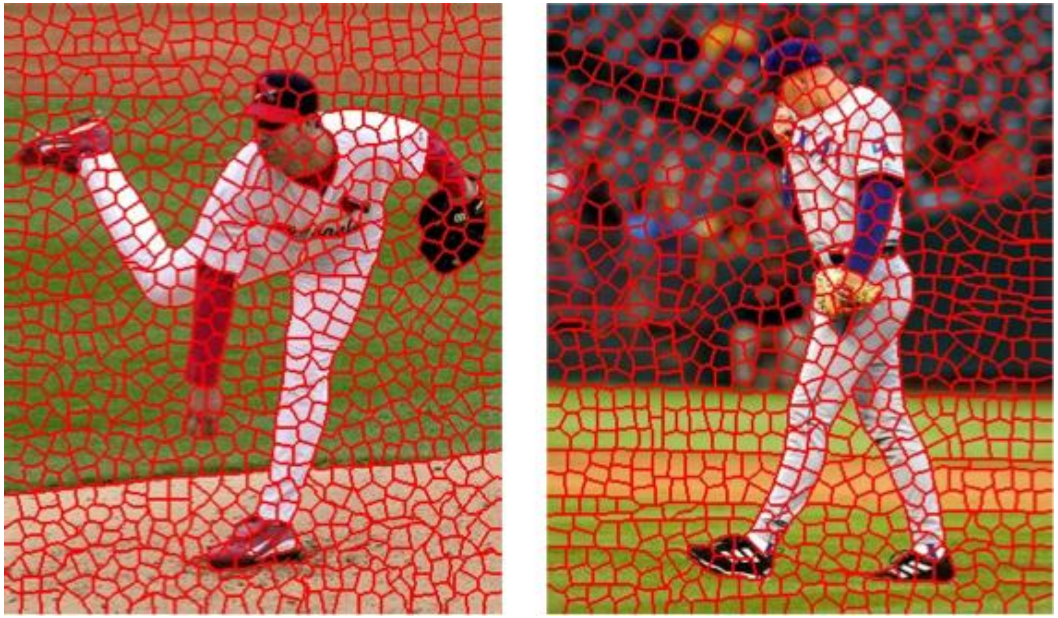
- Separate image into coherent "objects"



The Goals of Segmentation

- Separate image into coherent “objects”
- Group together similar-looking pixels for efficiency of further processing

“superpixels”



X. Ren and J. Malik. Learning a classification model for segmentation. ICCV 2003.

Segmentation for efficiency

[Felzenszwalb and Huttenlocher 2004]

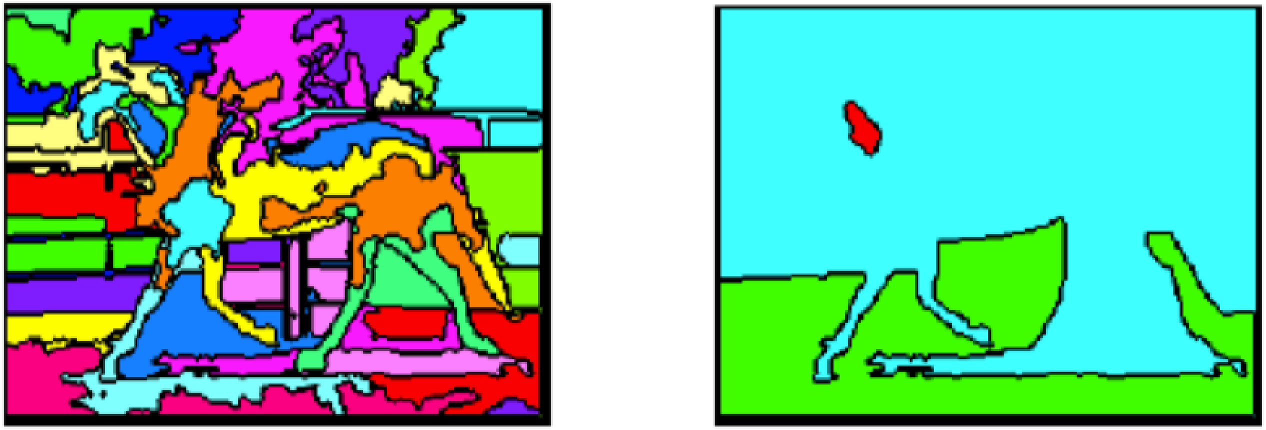
[Hoiem et al. 2005, Mori 2005]

[Shi and Malik 2001]

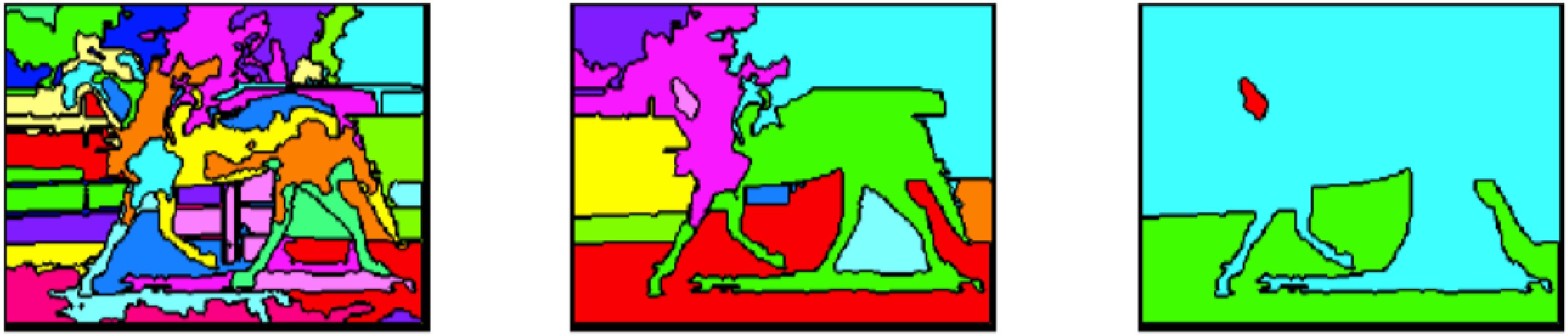
Segmentation as a result

Rother et al. 2004

Types of segmentations



Oversegmentation Undersegmentation



Multiple Segmentations

One way to think about "segmentation" is Clustering

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 or denoising
  - Represent a large continuous vector with the cluster number
- **Counting**
  - Histograms of texture, color, SIFT vectors
- **Segmentation**
  - Separate the image into different regions
- **Prediction**
  - Images in the same cluster may have the same labels

How do we cluster?

- **Agglomerative clustering**
  - Start with each point as its own cluster and iteratively merge the closest clusters
- **K-means**
  - Iteratively re-assign points to the nearest cluster center
- **Mean-shift clustering**
  - Estimate modes of pdf

General ideas

- **Tokens**
  - whatever we need to group (pixels, points, surface elements, etc., etc.)
- **Bottom up clustering (e.g. agglomerative clustering)**
  - tokens belong together because they are locally coherent
- **Top down clustering (e.g. divisive clustering)**
  - tokens belong together because they lie on the same visual entity (object, scene…)

Notes: These two are not mutually exclusive

Examples of Grouping in Vision

Determining image regions

Grouping video frames into shots

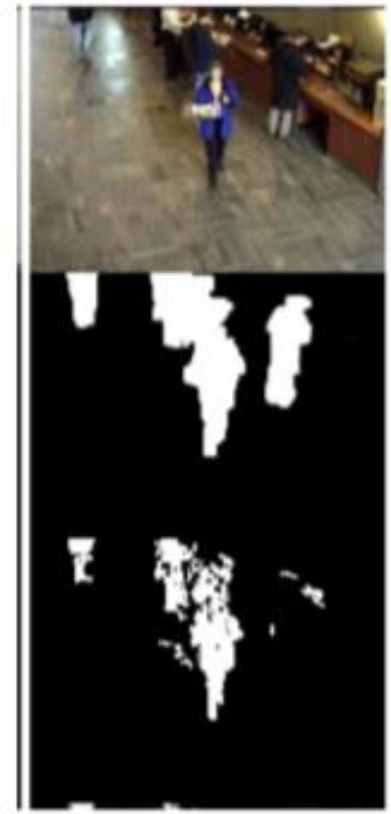
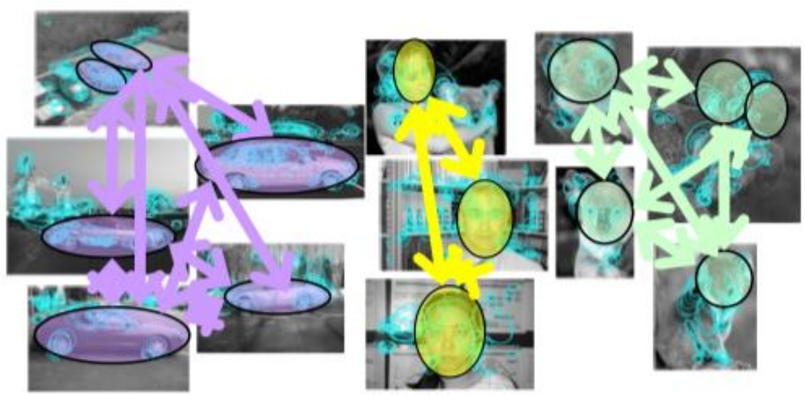


Figure-ground

What things should be grouped?

What cues indicate groups?

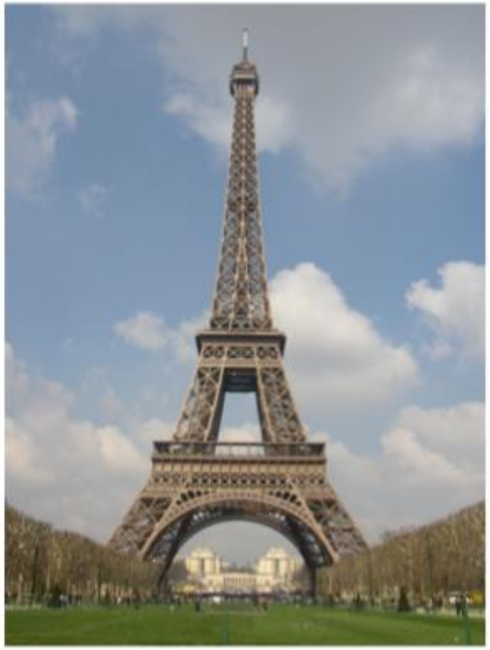


Object-level grouping

Similarity




Symmetry



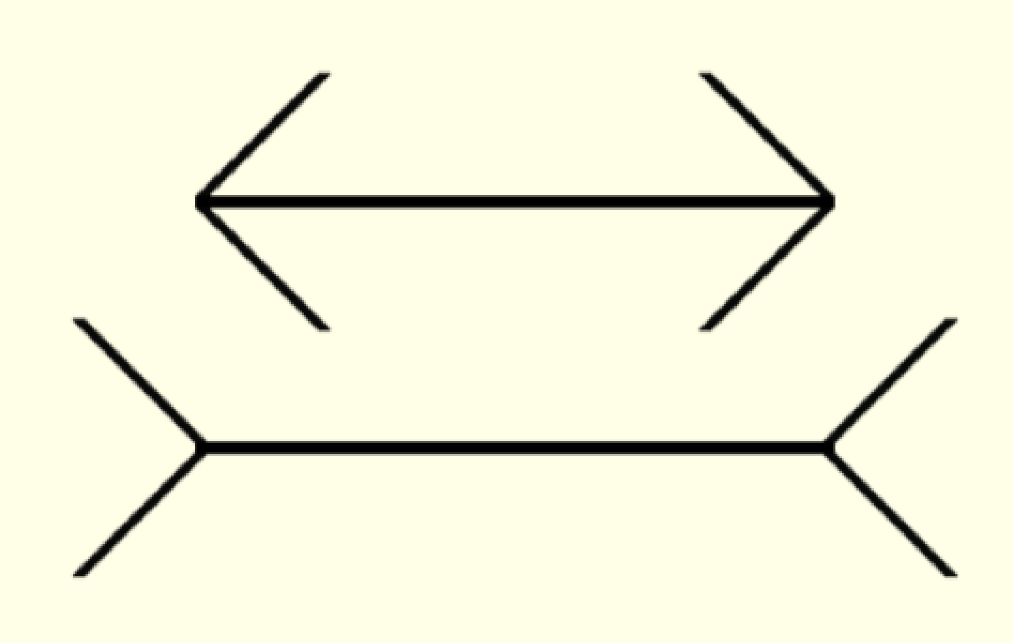
Slide credit: Kristen Grauman

Common Fate

Image credit: Arthus-Bertrand (via F. Durand)

Slide credit: Kristen Grauman

Muller-Lyer Illusion

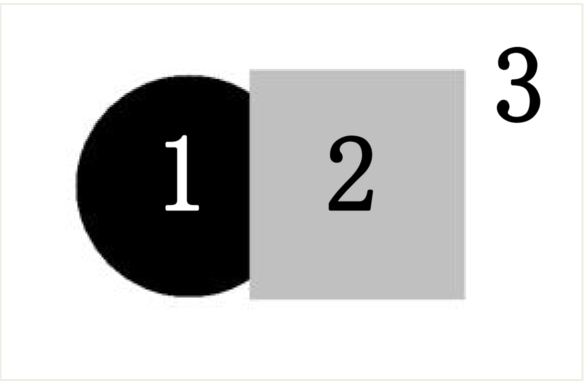


What makes the bottom line look longer than the top line

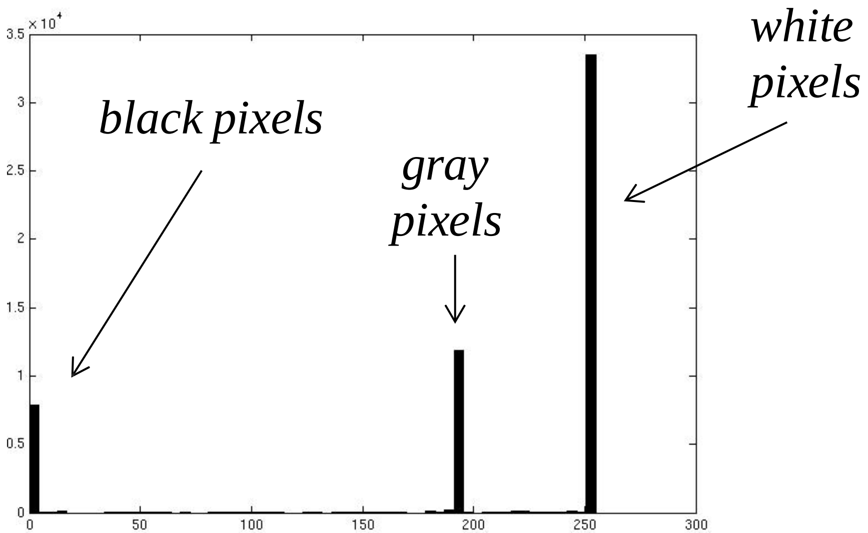
Proximity

Slide credit: Kristen Grauman

Image Segmentation: Toy Example



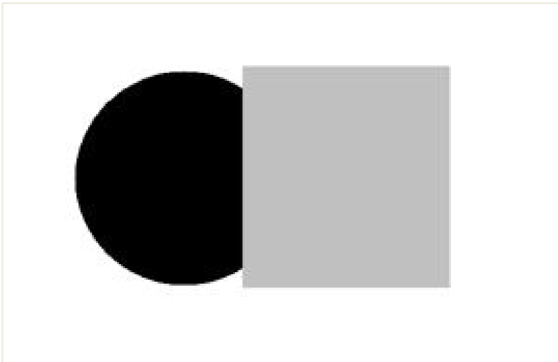
input image



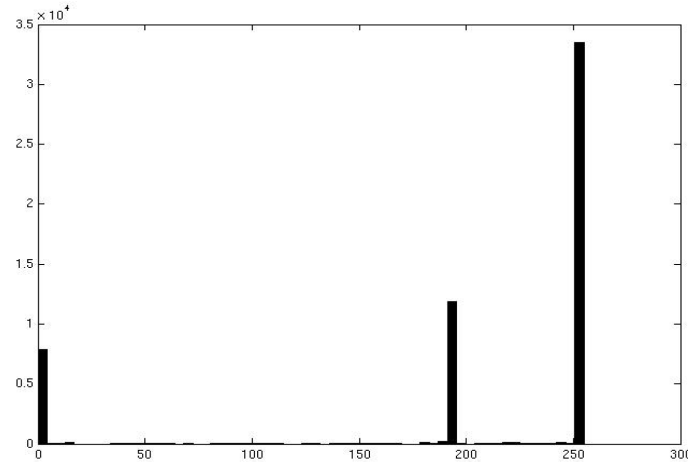
intensity

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

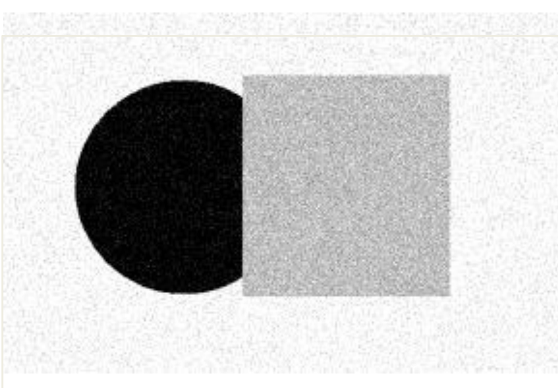
Image Segmentation: Toy Example



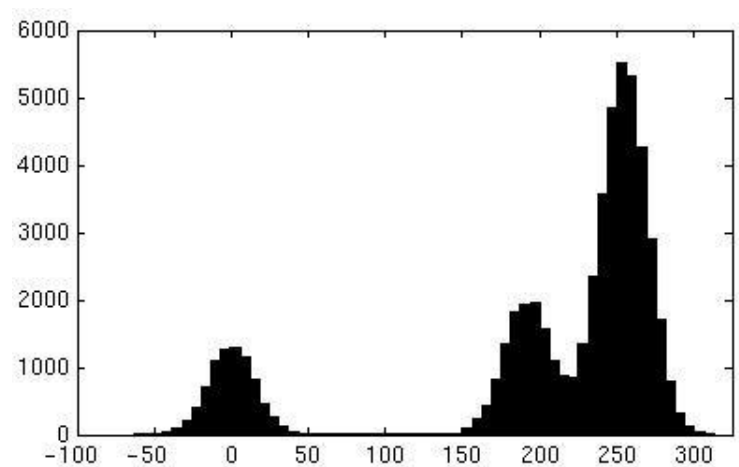
Input image



Intensity

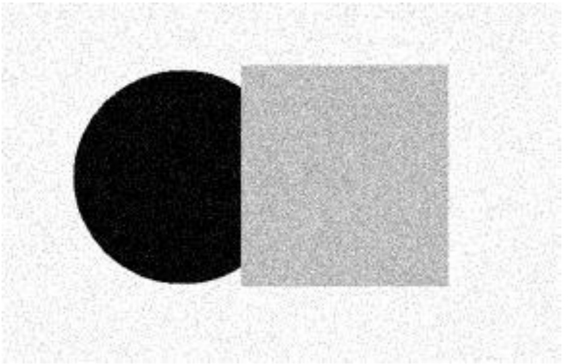


Input image

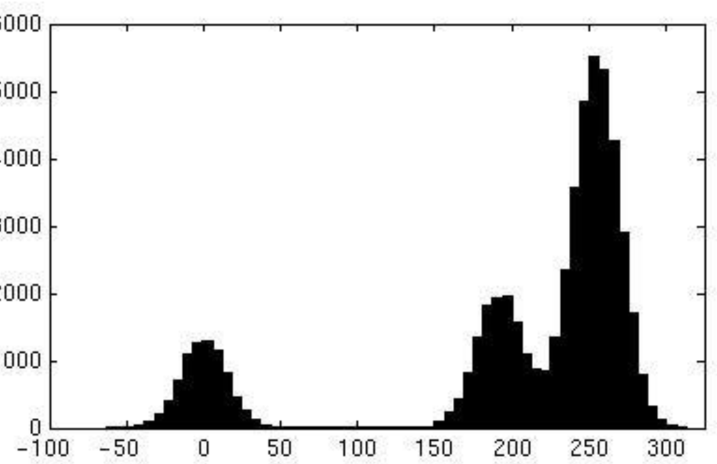


Intensity

Image Segmentation: Toy Example



Input image



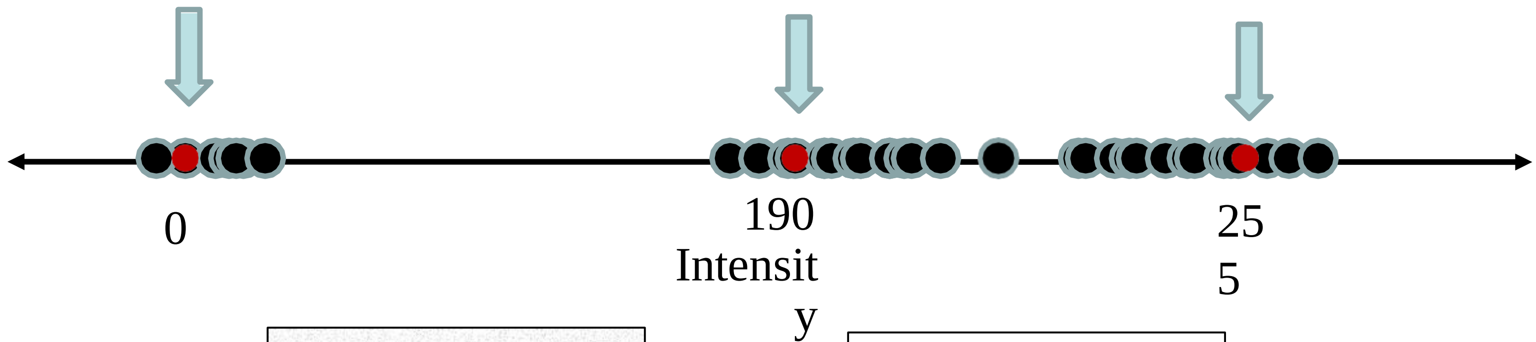
Intensity

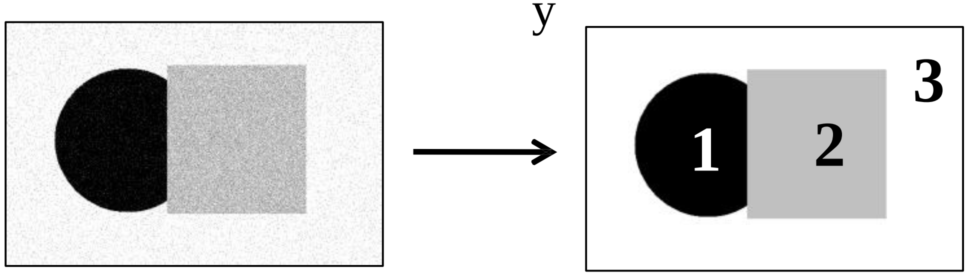
- Now how to determine the three main intensities that define our groups?
- We need to cluster.

What we will learn today?

- Introduction to segmentation and clustering
- K-means clustering
- Mean-shift clustering

K-means clustering: example





- 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

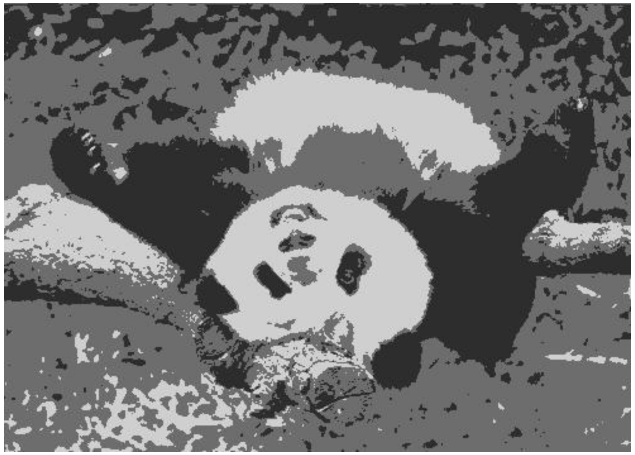
$$SSD = \sum_{Cluster\ i} \sum_{x \in cluster\ i} (x - c_i)^2$$

Feature space

- Grouping pixels based on intensity similarity.
- The feature space is 1D

Original image

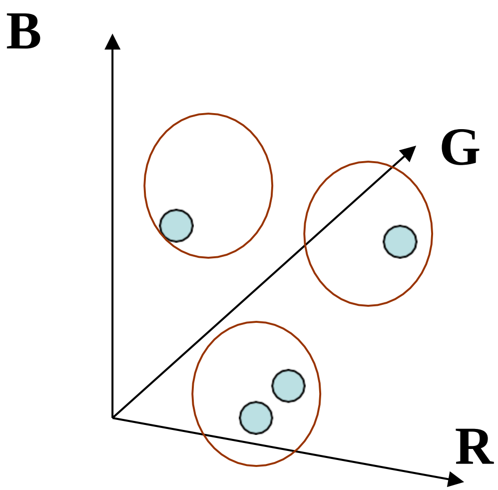
2 clusters

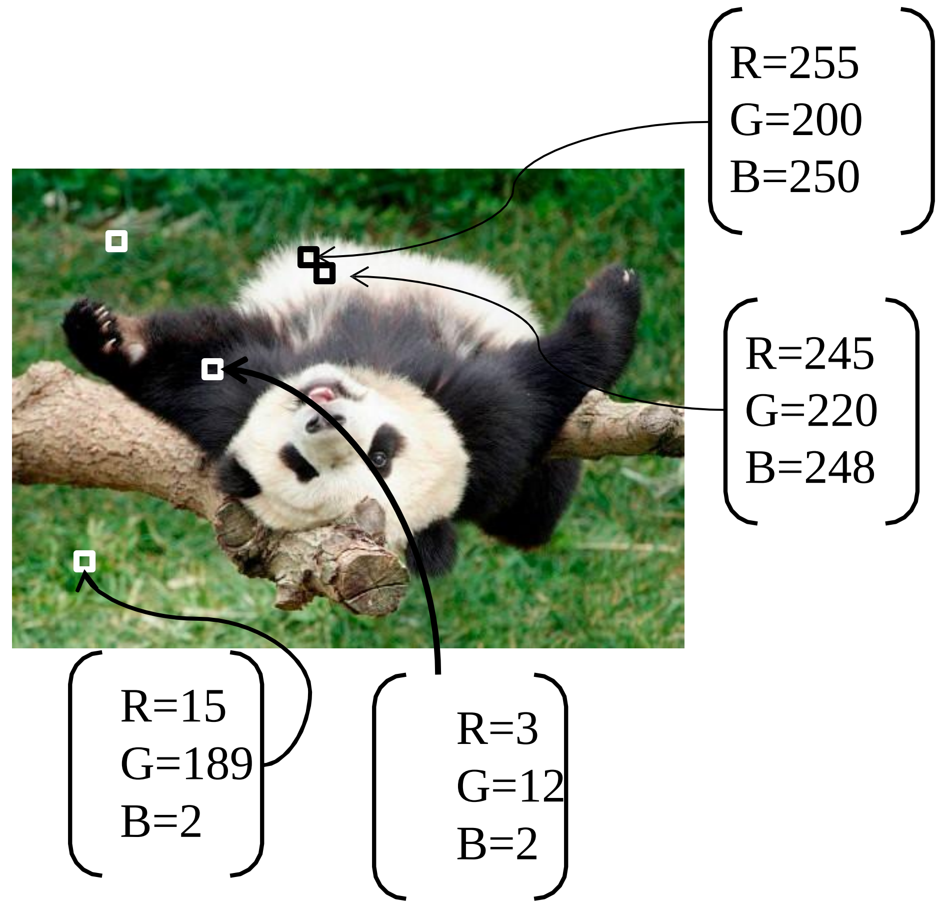


3 clusters

Feature space

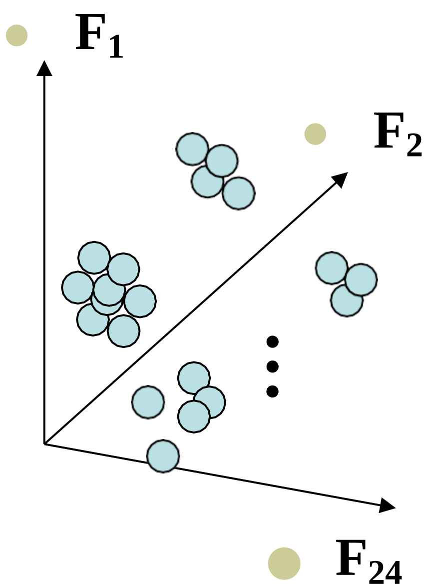
- Grouping pixels based on color similarity
- The feature space is 3D

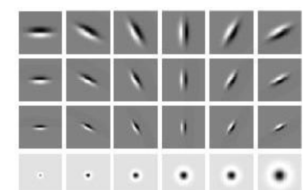




Feature space

- Grouping pixels based on texture similarity
- The feature space is 24D

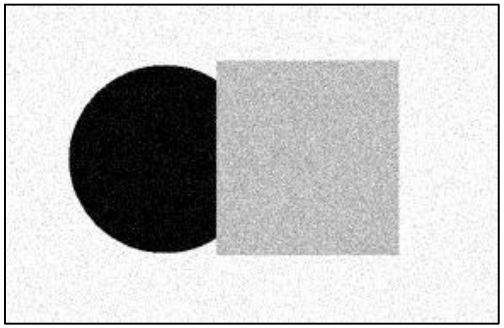




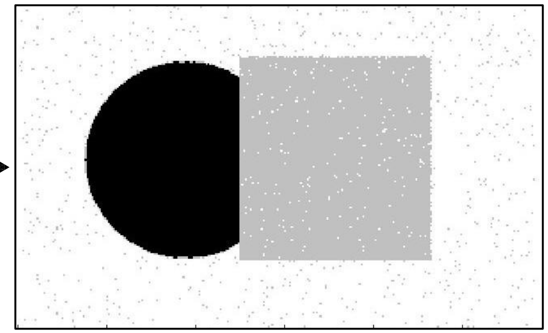
Filter bank of 24 filters

Feature space

- Assigning a cluster label per pixel may yield outliers.



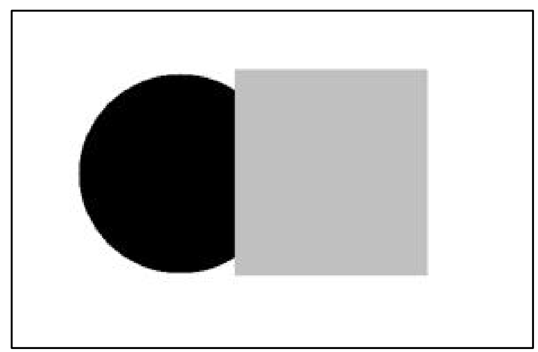
Original



Labeled by cluster center's intensity

?

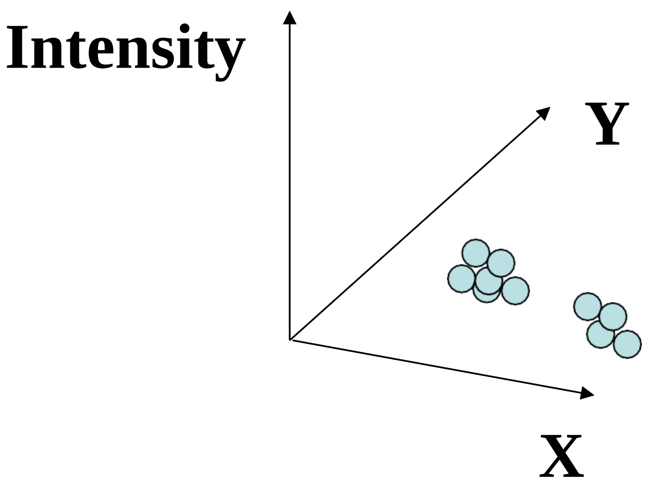
- How to ensure that the clustering results are spatially smooth?

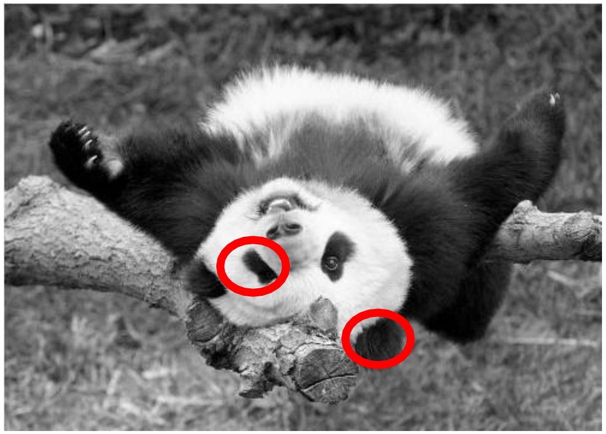


Feature space

- Grouping pixels based on *intensity+position* similarity.

- The feature space is 3D





- Way to encode both similarity and proximity.

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 cluster | Color-based clusters

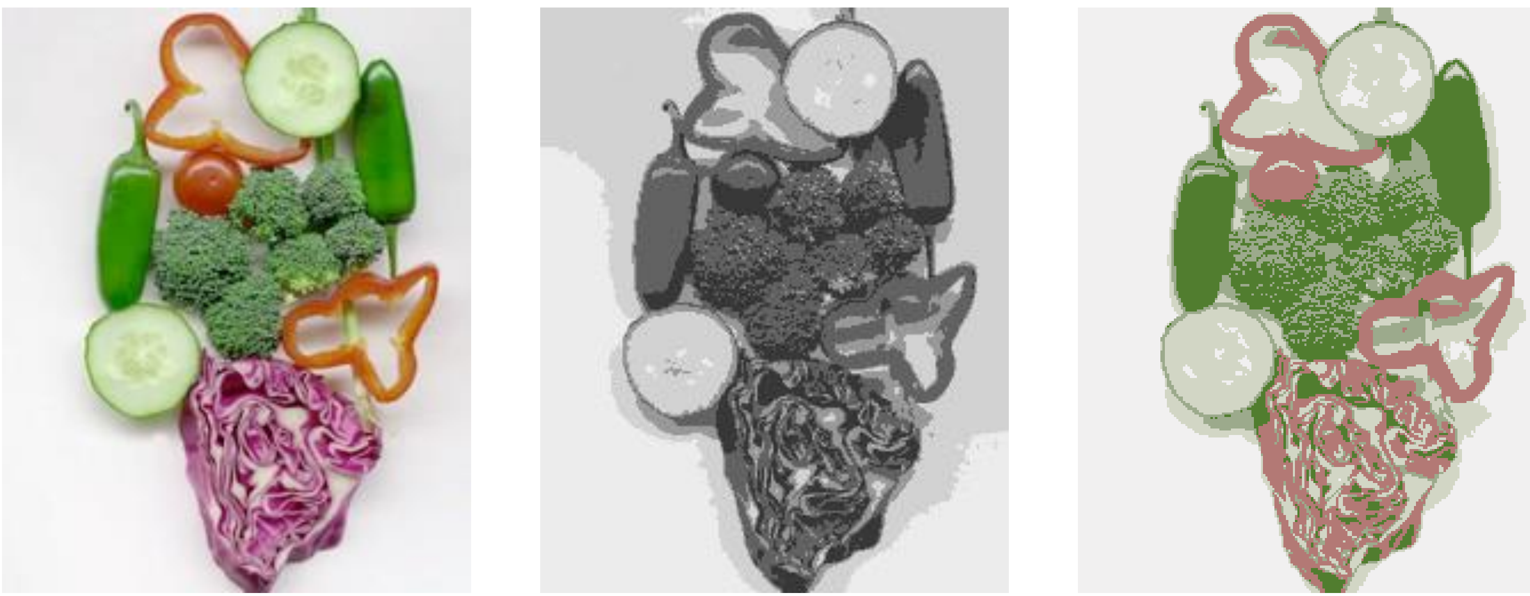


Image source: Forsyth & Ponce

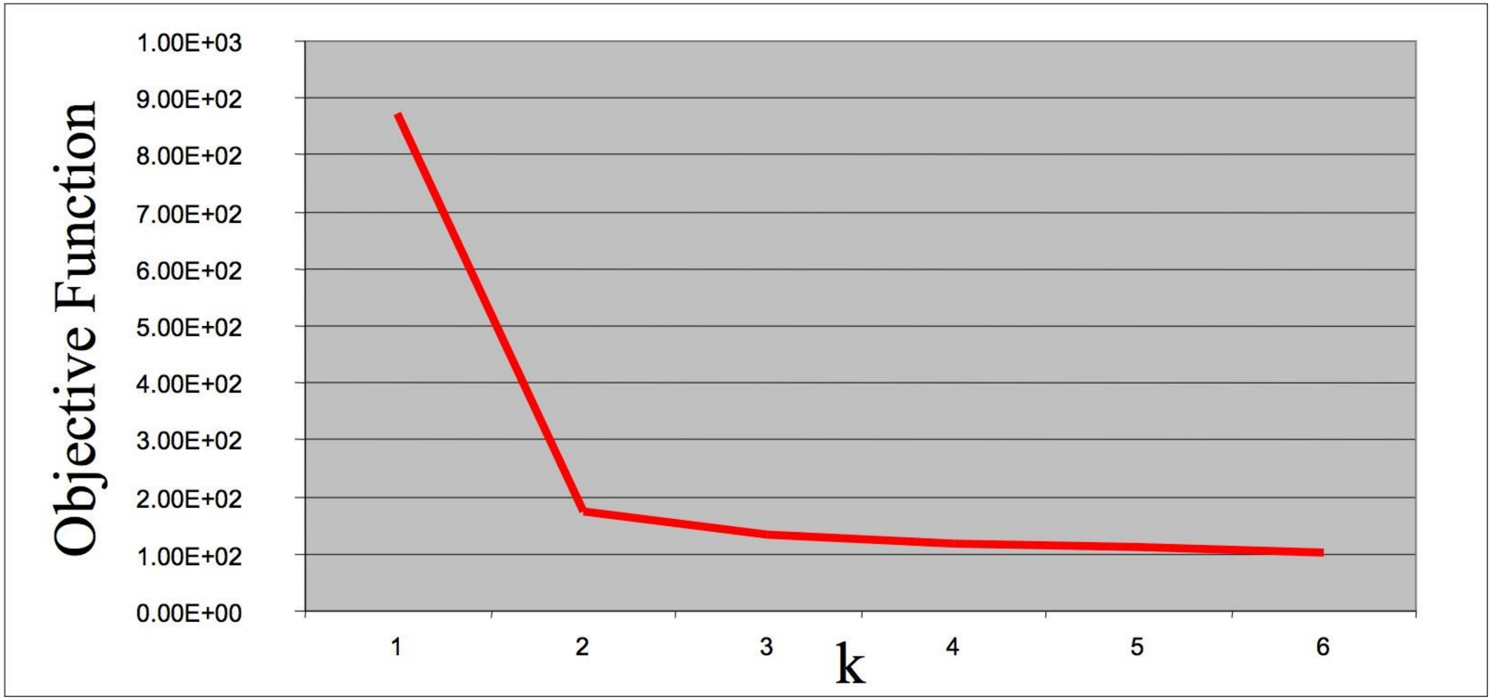
K-Means Clustering Results

- Clustering based on (r,g,b,x,y) values enforces more spatial coherence

Image source: Forsyth & Ponce

How to evaluate clusters?

- **Determine the number of clusters**
  – Use "elbow finding" (or "knee finding"). Try different numbers of clusters in a validation set and look at performance.

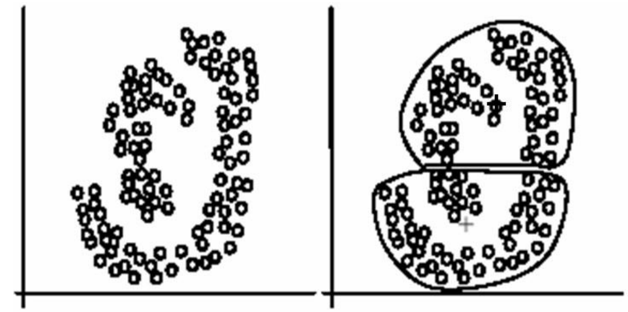


How to evaluate clusters?

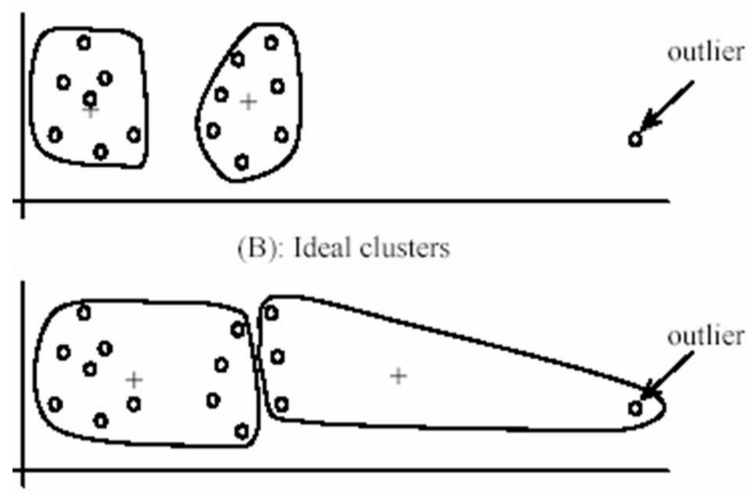
- **Evaluate clustering quality**
  - Internal method:
    - – unsupervised, evaluate the class cohesion and class dispersion
    - – silhouette coefficient; Davies-Bouldin Index(DBI); Dunn Index(DI)
  - External method:
    - – supervised, evaluate the consistency between clustering results and benchmark data(labeled data)
    - – Bcubed precision, Bcubed recall
    - – Jaccard Coefficient(JC), Fowlkes and Mallows Index(FMI), Rand Index(RI)

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
  - Prone to local minima
  - Fit bad for non spherical data
  - Sensitive to outliers
- Usage
  - Unsupervised clustering
  - Rarely used for pixel segmentation



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

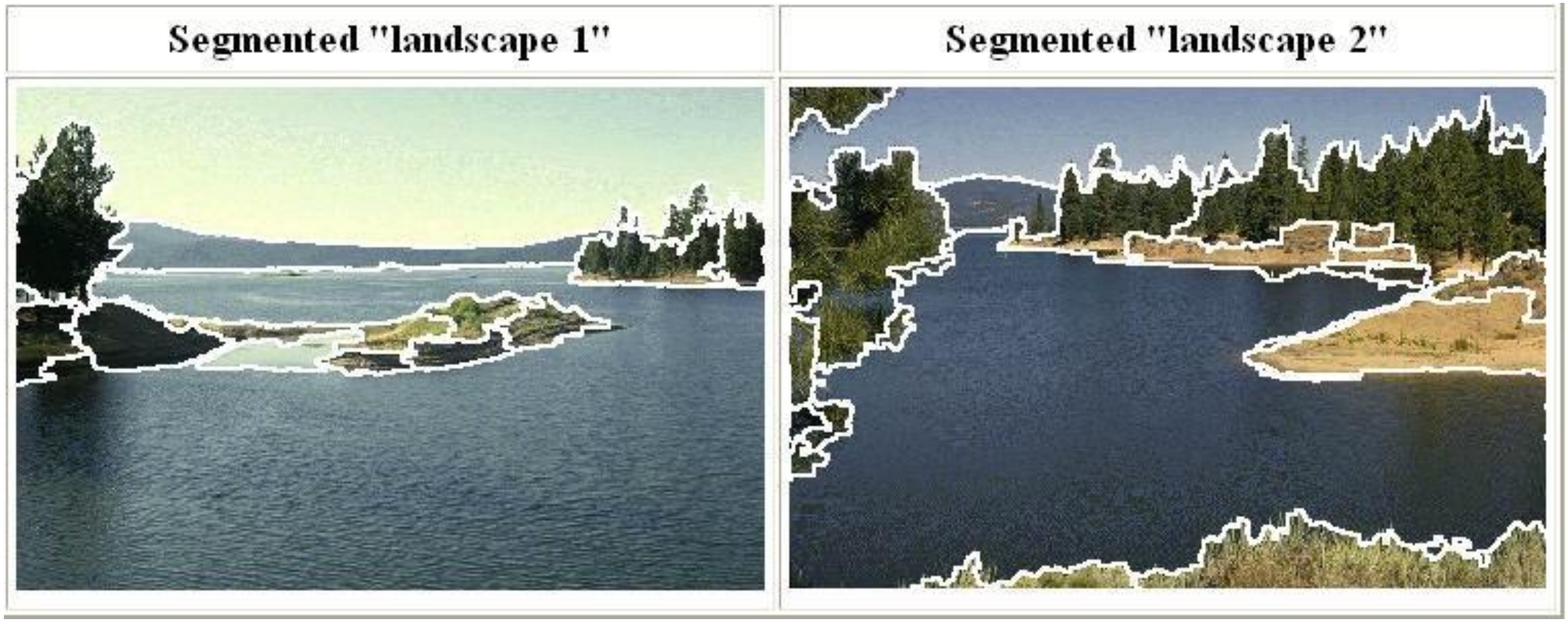


What we will learn today?

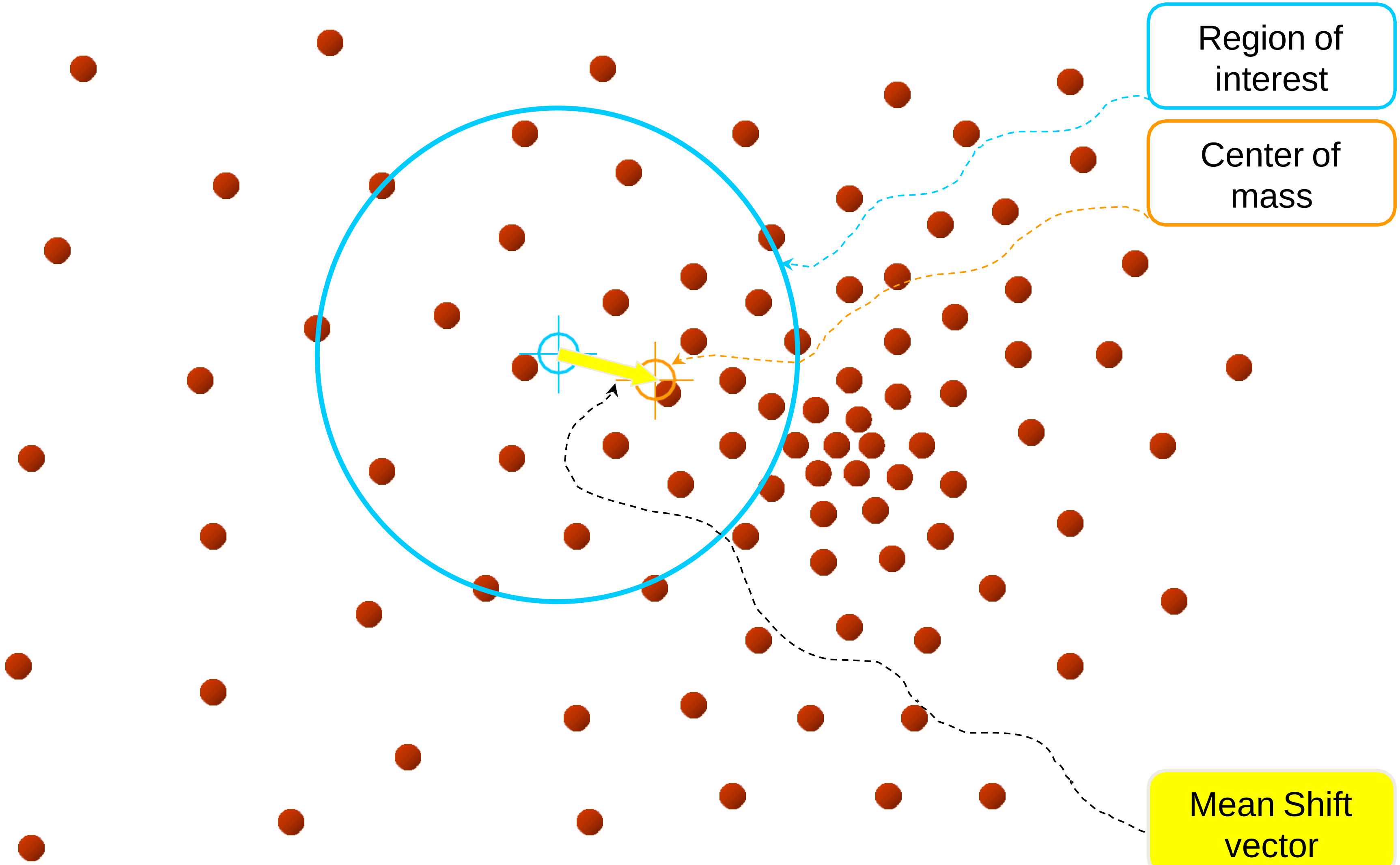
- Introduction to segmentation and clustering
- K-means clustering
- Mean-shift clustering

Mean-Shift Segmentation

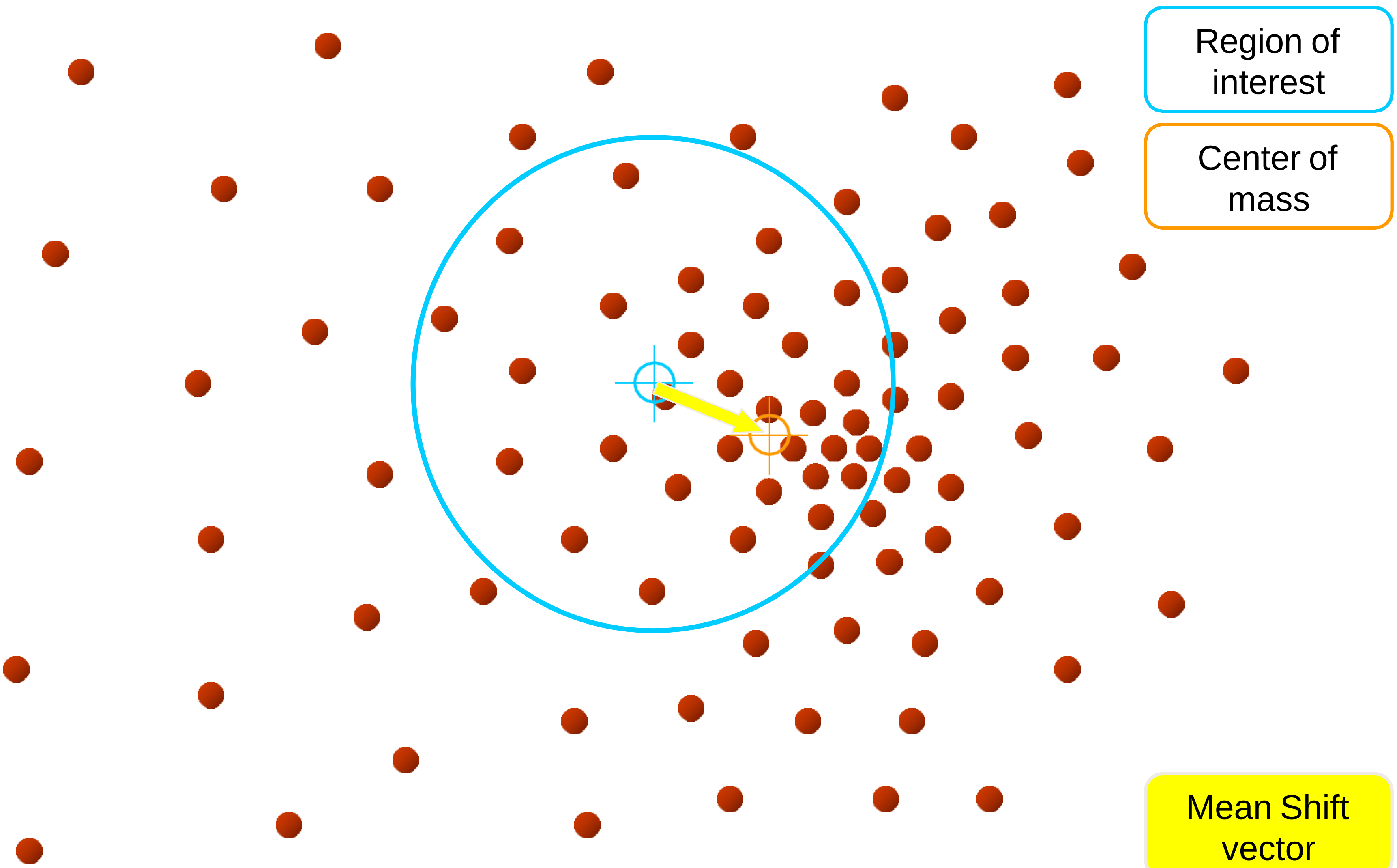
- An advanced and versatile technique for clustering-based segmentation



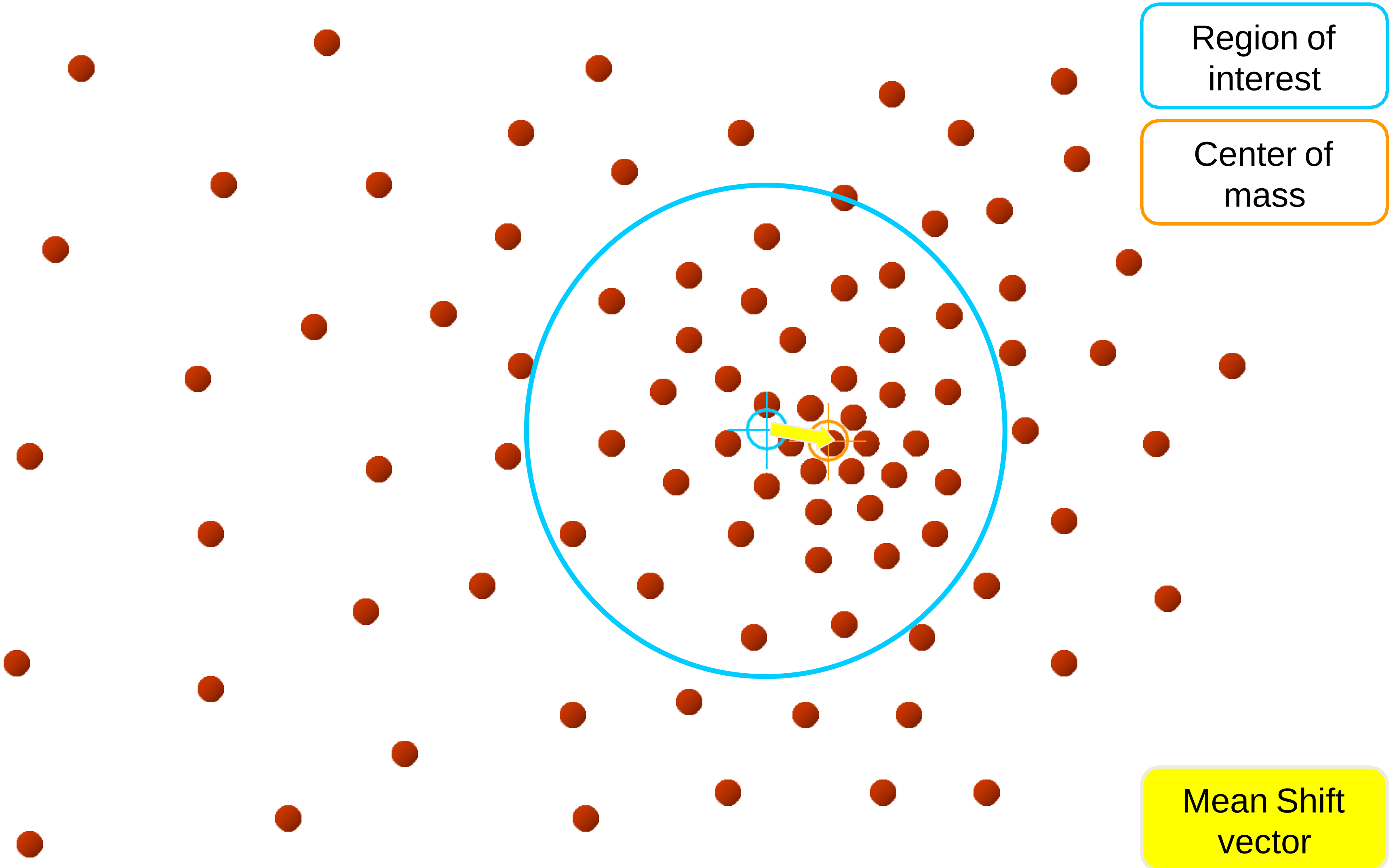
Mean-Shift



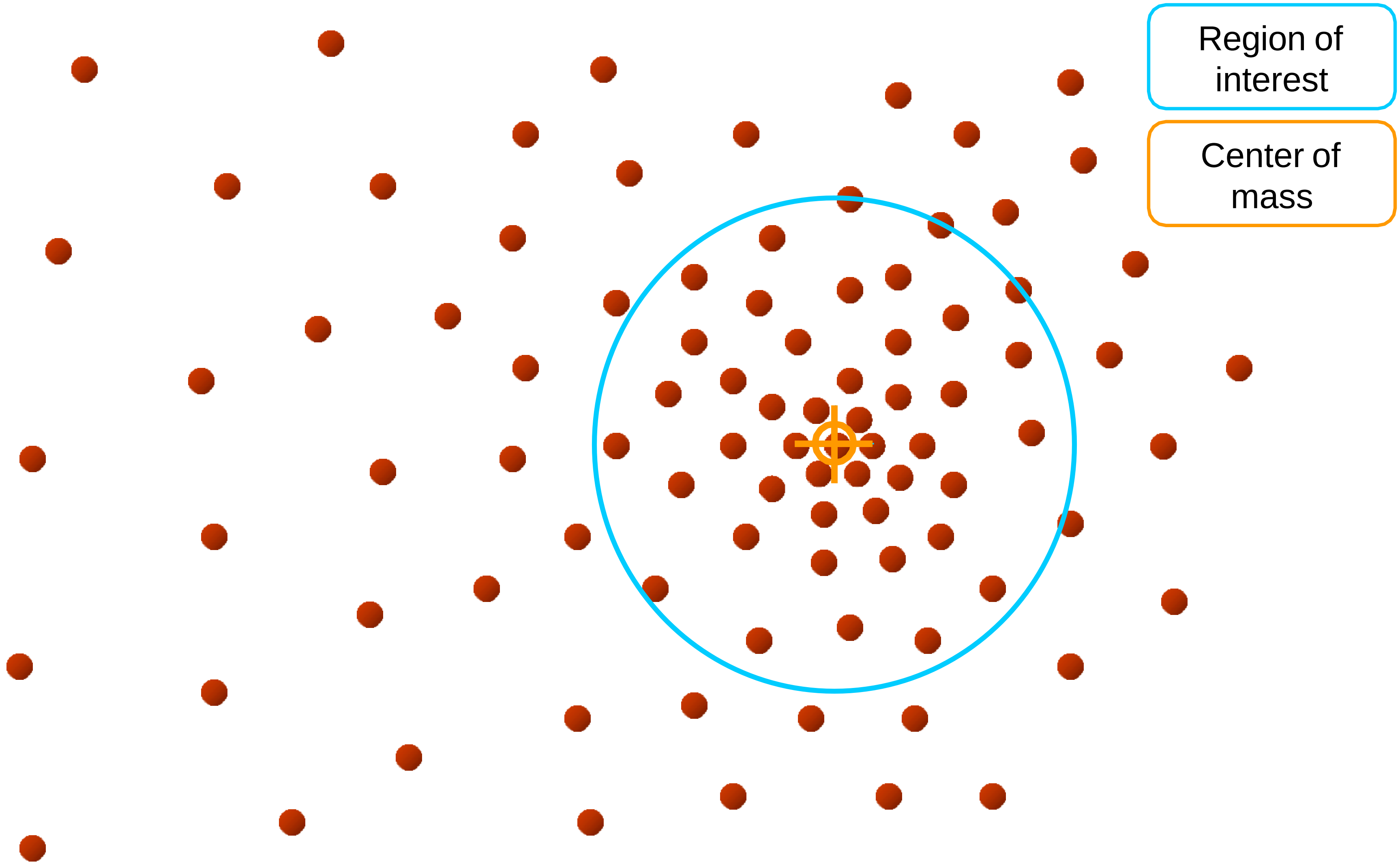
Mean-Shift



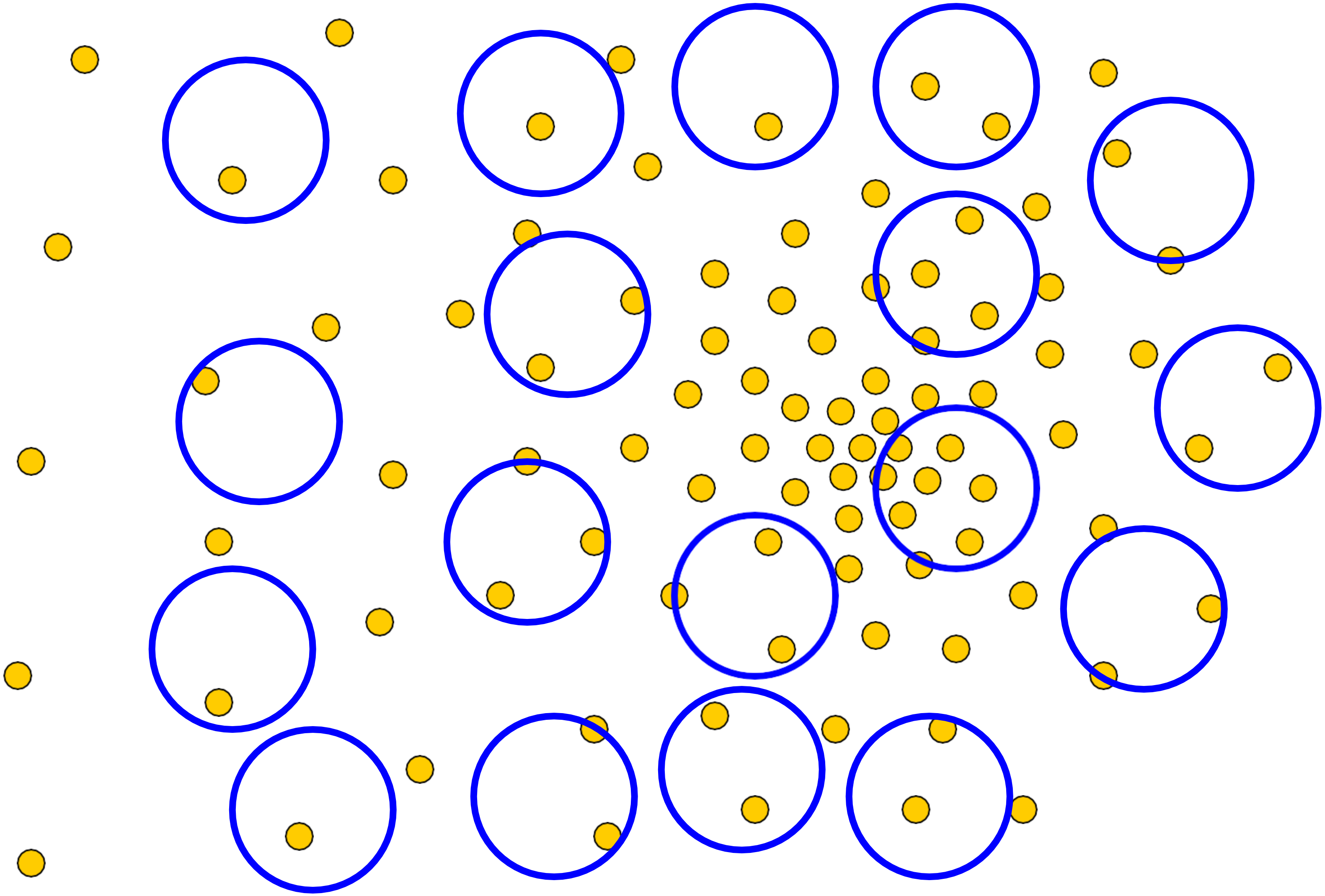
Mean-Shift



Mean-Shift

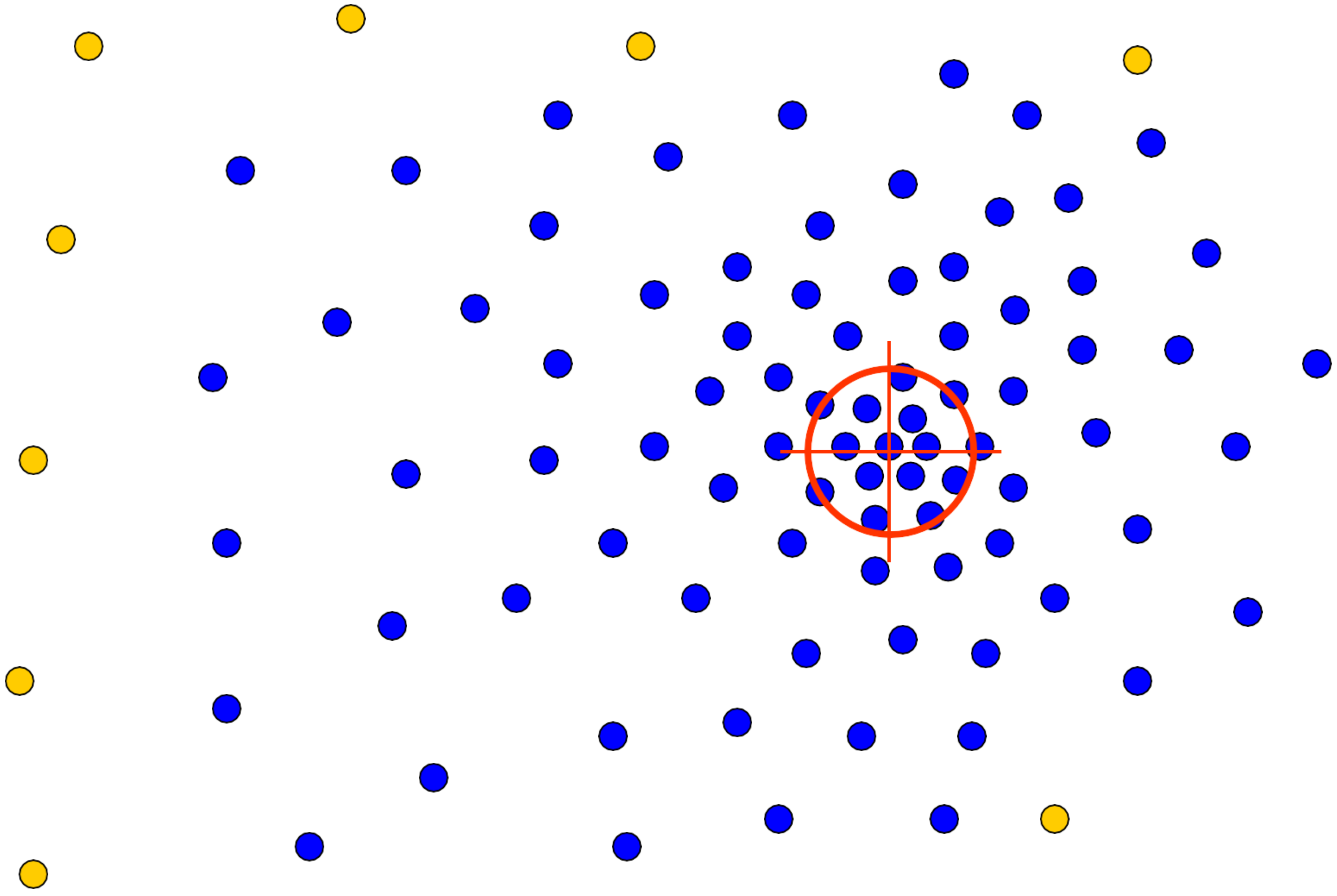


Real Modality Analysis



Tessellate the space with windows Run the procedure in parallel

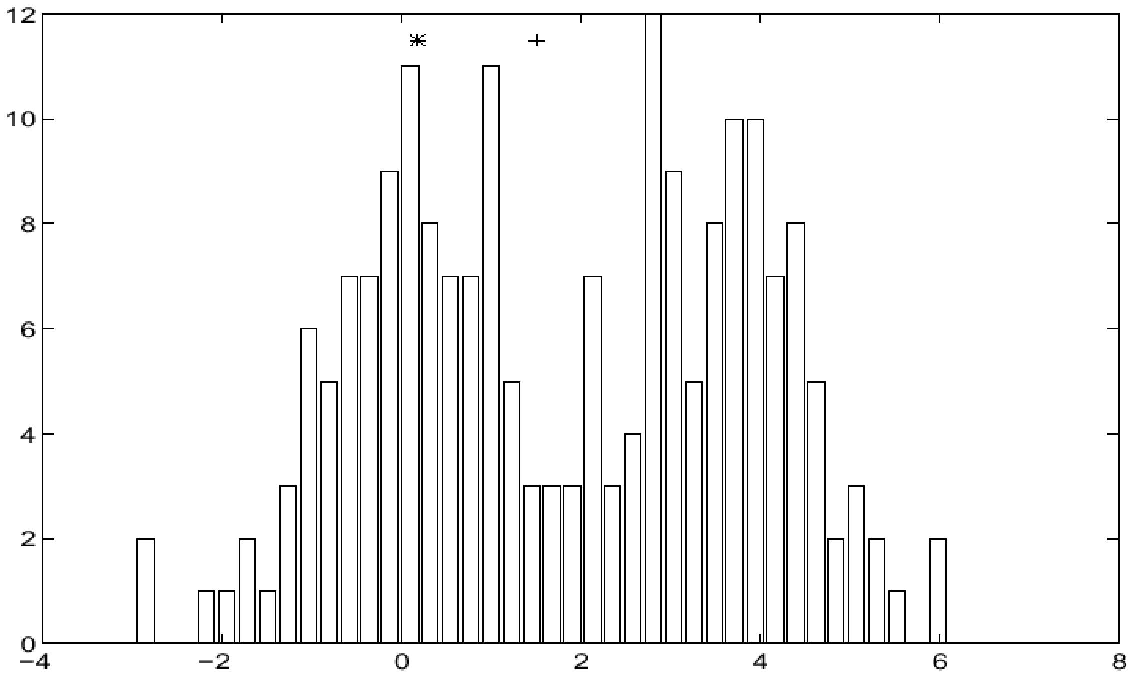
Real Modality Analysis



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

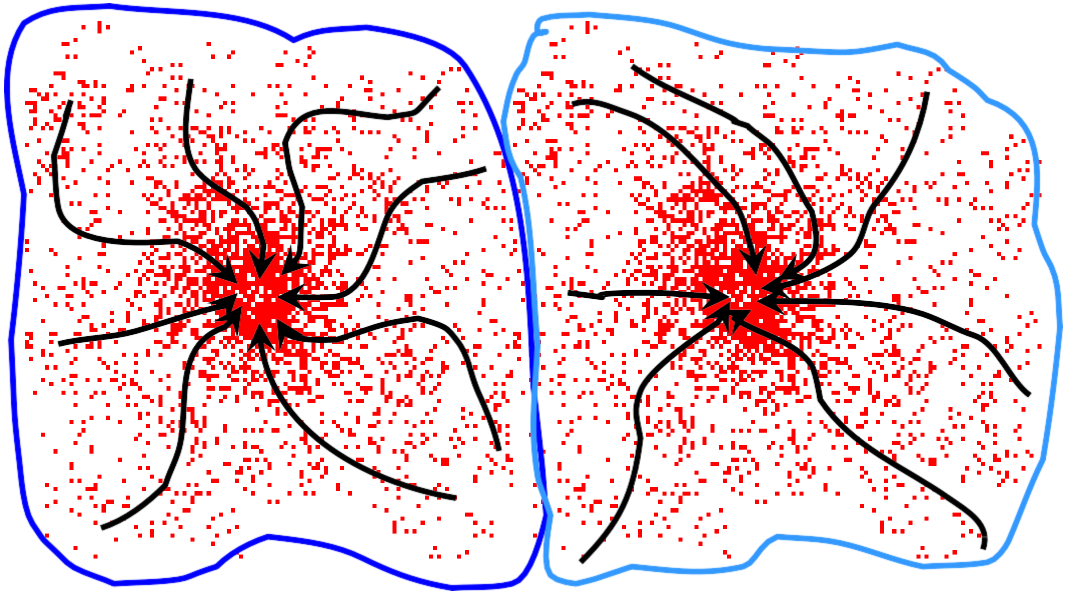
Mean-Shift Algorithm

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



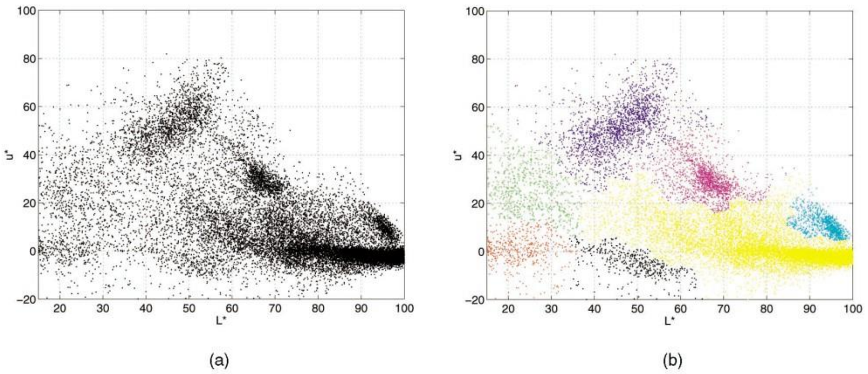
Mean-Shift Clustering

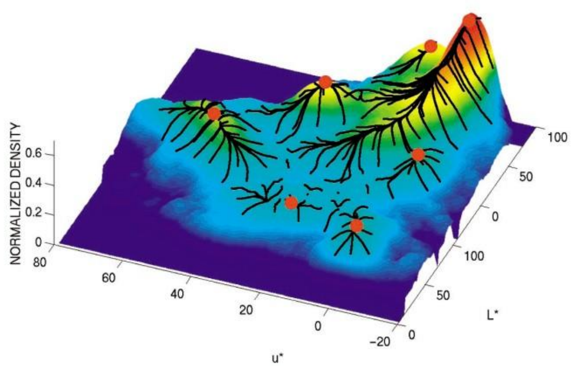
- Cluster: all data points in the attraction basin of a mode
- Attraction basin: the region for which all trajectories lead to the same mode



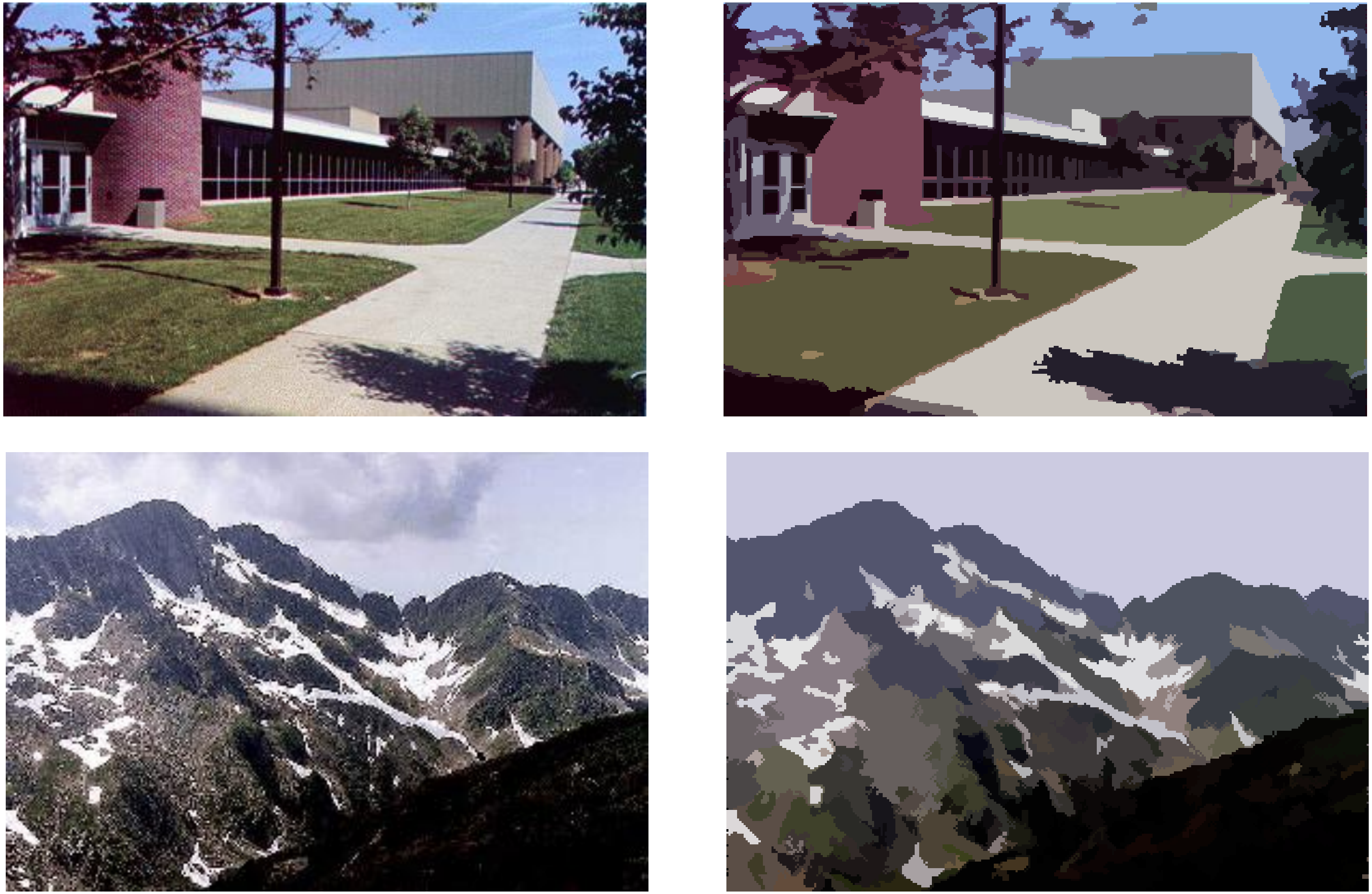
Mean-Shift Clustering/Segmentation

- 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

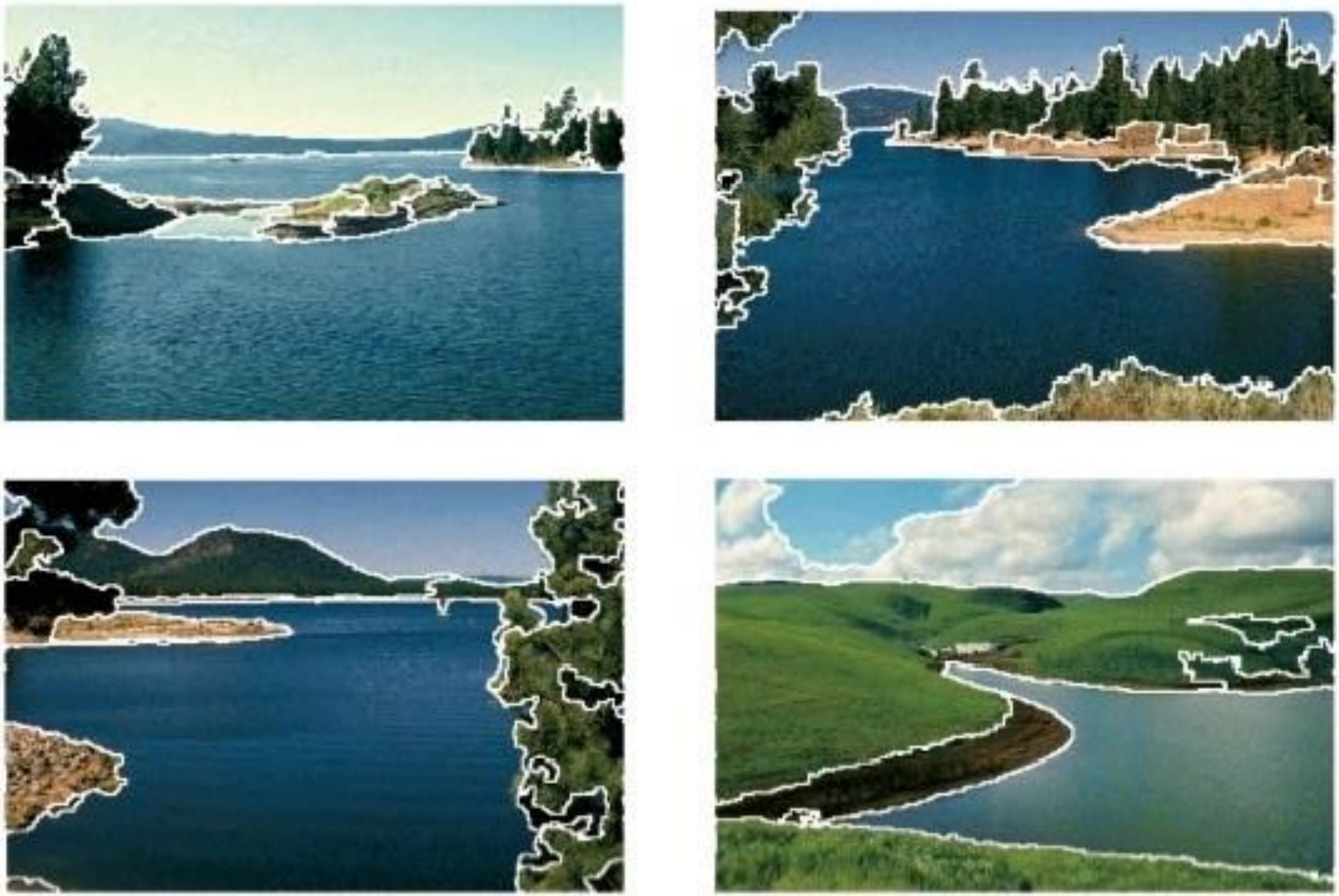




Mean-Shift Segmentation Results

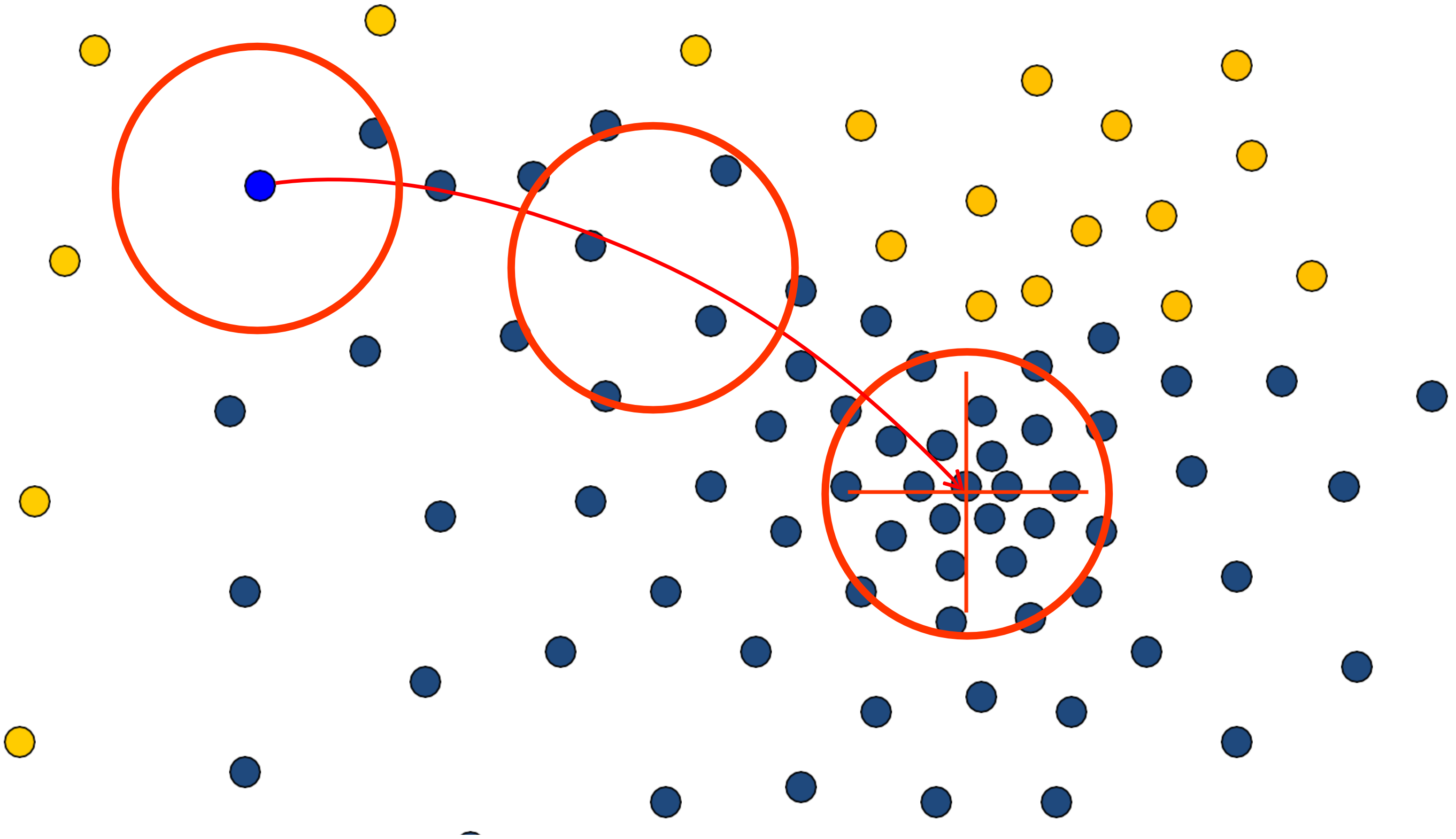


More Results



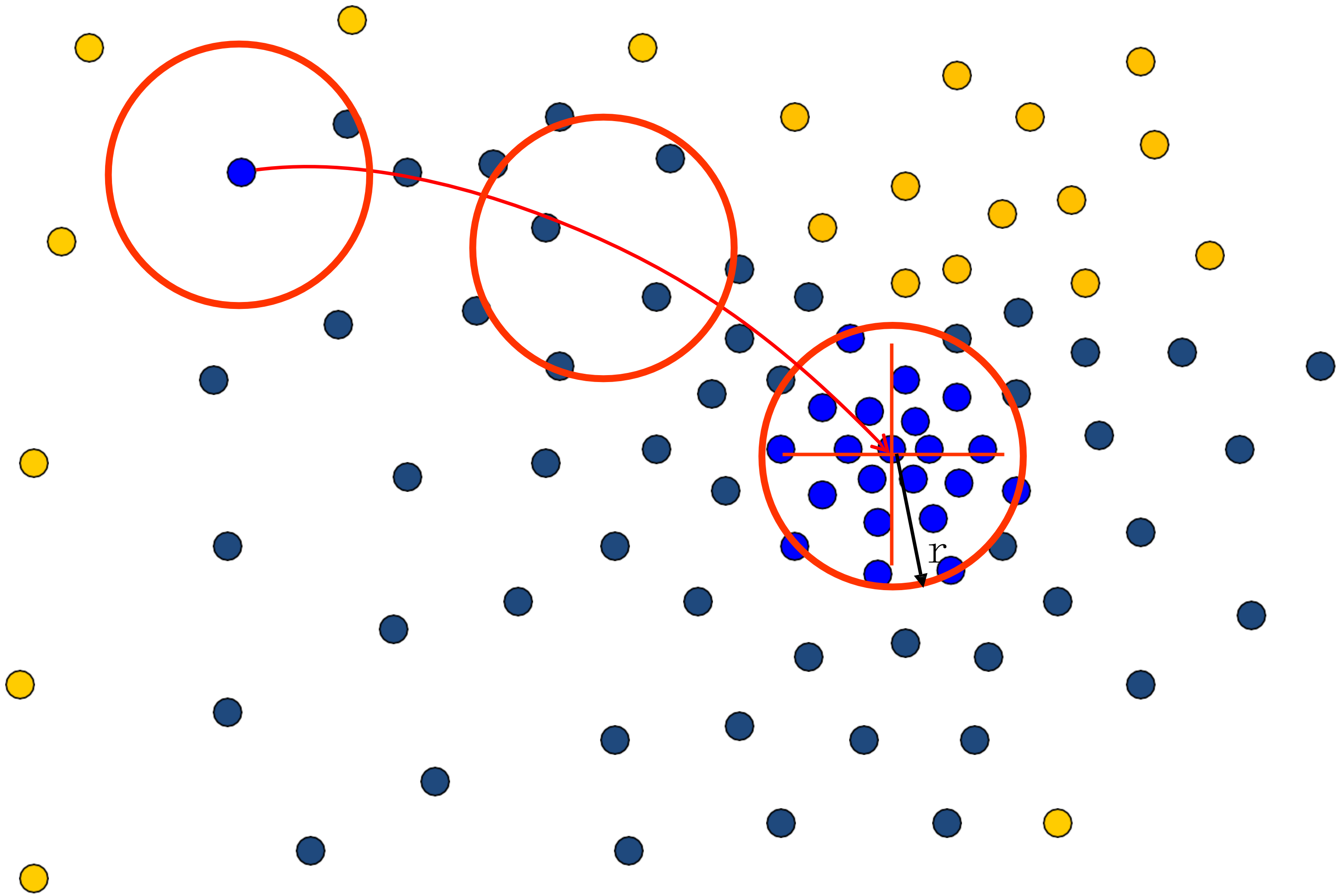
More Results

Problem: Computational Complexity



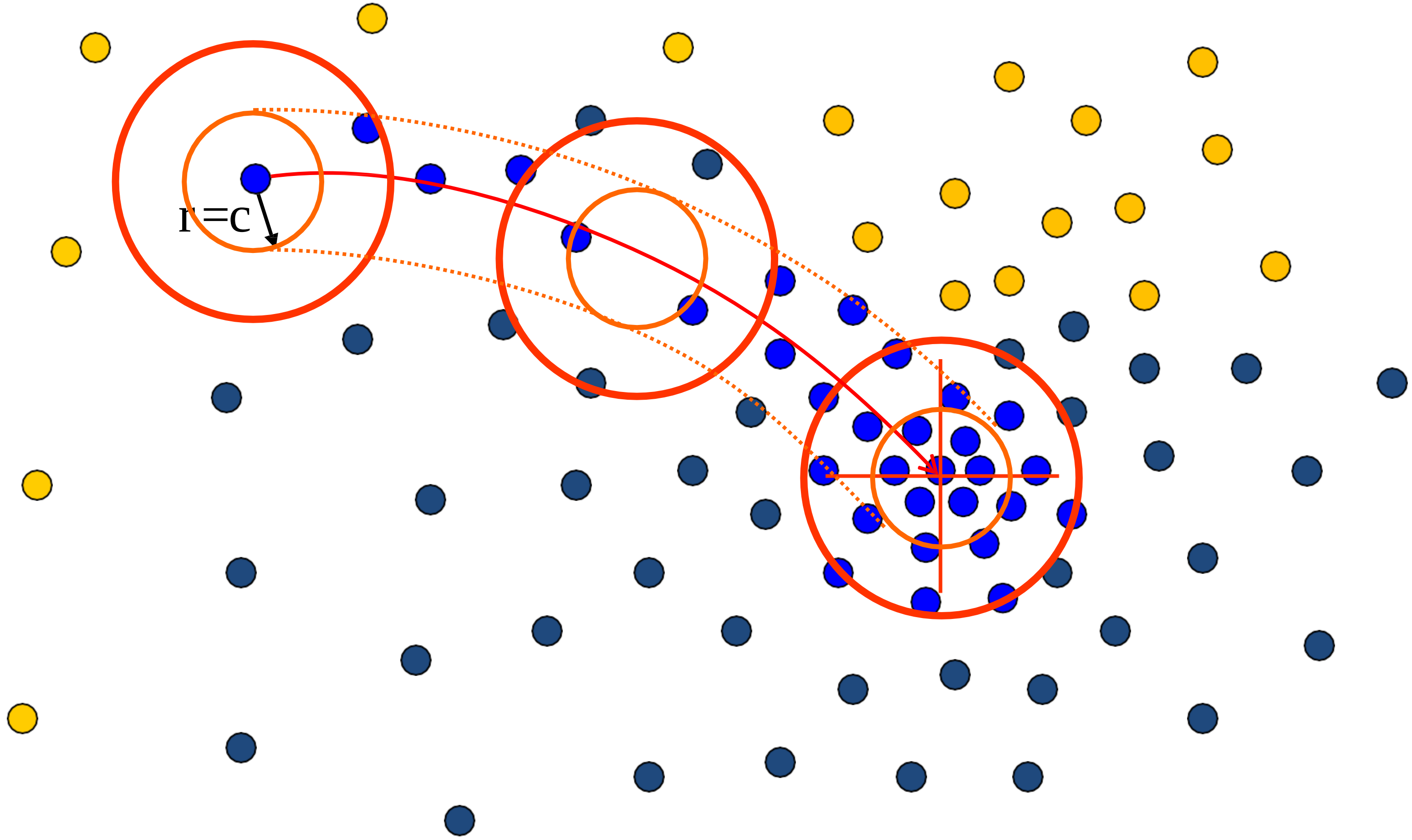
- Need to shift many windows…
- Many computations will be redundant.

Speedups: Basin of Attraction



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

Speedups: Basin of Attraction



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

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,

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

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

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

where c_k represents a normalization constant.

Technical Details

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 $\phi(x) = \begin{cases} \frac{3}{4}(1 - x^2) & if\ |x| \leq 1 \\ 0 & else \end{cases}$

Technical Details

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

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

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.

Technical Details

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

- Computer the mean shift vector m:

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

- Translate the density window:

$$x^{t+1} = x^t + m(x^t)$$

- Iterate steps 1 and 2 until convergence.

$$\nabla f(x) = 0$$

Summary Mean-Shift

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): Smoother density with larger h
- Finds variable number of modes
- Robust to outliers

Cons

- Output depends on window size
- Window size (bandwidth) selection is not trivial
- Computationally (relatively) expensive
- Does not scale well with dimension of feature space

What we have learned today?

- Introduction to segmentation and clustering
- K-means clustering
- Mean-shift clustering

SUN YAT-SEN UNIVERSITY

Next time:

Dimensionality Reduction

Pattern Recognition (in Computer Vision)

Jinhua Ma,
School of Computer Science and Engineering, Sun Yat-Sen University

Most of our slides have been borrowed from the courses of Stanford CS131 and CS231N, Thanks!