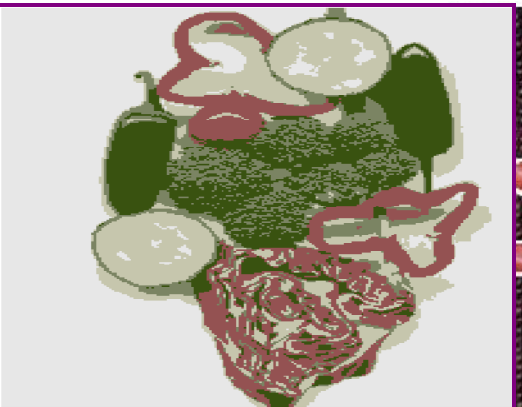
计算机视觉

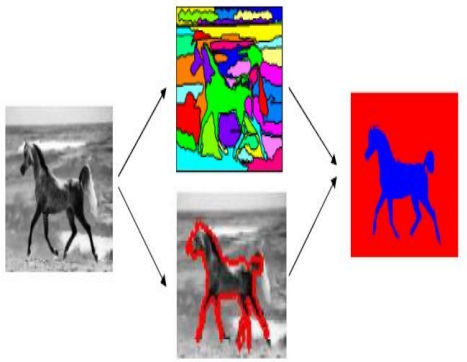
Computer Vision

Lecture 12: Segmentation

张 超

信息科学技术学院 智能科学系



Outline for Segmentation

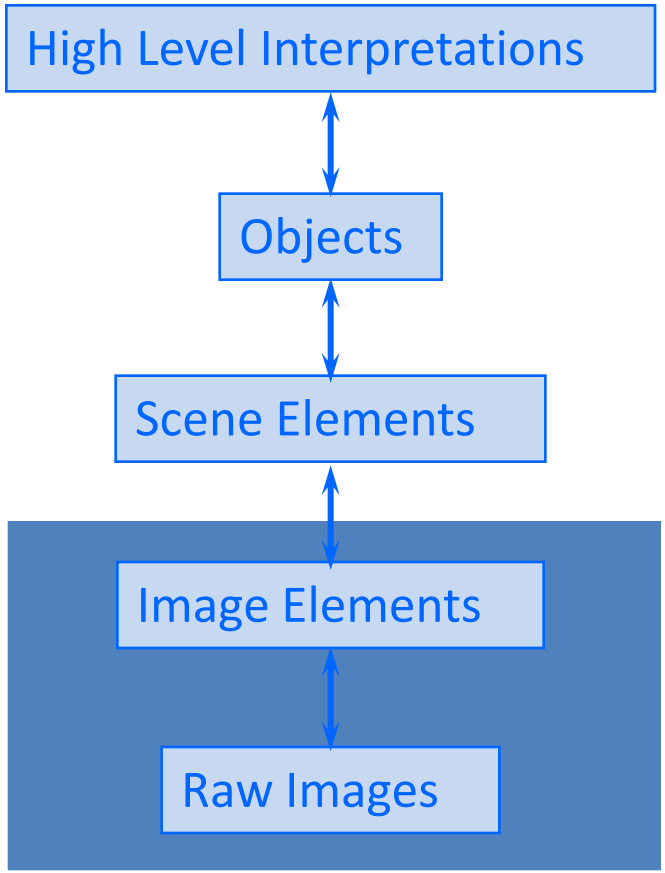
- Introduction
  - Goal of Segmentation
- Segmentation and grouping
- Image Segmentation by Clustering Pixels
  - Segmentation as clustering
  - K-means clustering
  - Mean shift
- Spectral Clustering(Wei Fangyun)
- Integrating top-down and bottom-up segmentation for recognition

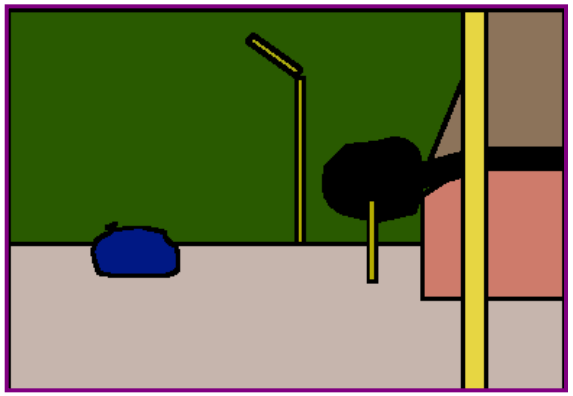
Outline for Segmentation

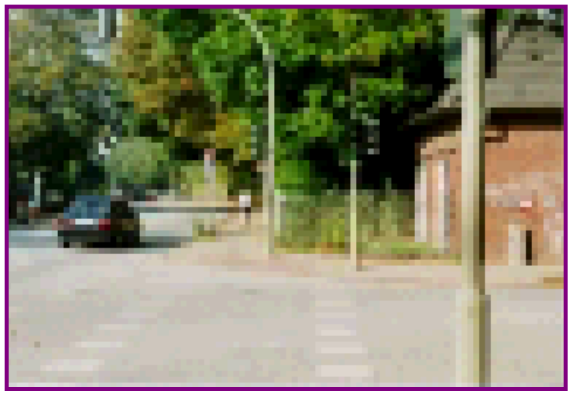
- Introduction
  - Goal of Segmentation
- Segmentation and grouping
- Image Segmentation by Clustering Pixels
  - Segmentation as clustering
  - K-means clustering
  - Mean shift
- Spectral Clustering(Wei Fangyun)
- Integrating top-down and bottom-up segmentation for recognition

Goal of Segmentation

- **Segment a scene into image elements which may correspond to meaningful scene elements**







Goal of Segmentation

- ***"Segmenting an image into image elements which may correspond to meaningful scene elements"***
  - ***Summarize information->compact representation***

- **What sort of image elements may correspond to meaningful scene elements?**
  - **Depends on type and complexity of images**

- **Segmentation has two objectives:**
  - **To decompose the image into parts for further analysis, gather features that belong together**
  - **To perform a change of representation: more meaningful, more efficient for further analysis**

- **Segmentation is not a well defined problem.**

Why do segmentation?

- To obtain primitives for other tasks
- For perceptual organization, recognition
- For graphics, image manipulation

Outline for Segmentation

- Introduction
  - Goal of Segmentation
- Segmentation and grouping
- Image Segmentation by Clustering Pixels
  - Segmentation as clustering
  - K-means clustering
  - Mean shift
- Spectral Clustering(Wei Fangyun)
- Integrating top-down and bottom-up segmentation for recognition

Segmentation and Grouping

- **Motivation: An awful lot of data, inference**
- **Obtain a compact representation from an image/motion sequence/set of tokens**
- **Should support application**
- **Broad theory is absent at present**
- **Grouping (or clustering)**
  - **collect together tokens that "belong together"**
- **Fitting**
  - **associate a model with tokens**
  - **issues**
    - **which model?**
    - **which token goes to which element?**
    - **how many elements in the model?**

General ideas

- tokens
  - whatever we need to group (pixels, points, surface elements, etc., etc.)
- top down segmentation
  - tokens belong together because they lie on the same object
- bottom up segmentation
  - tokens belong together because they are locally coherent
- These two are not mutually exclusive

Examples of Grouping in Vision

Determining image regions

Grouping video frames into shots

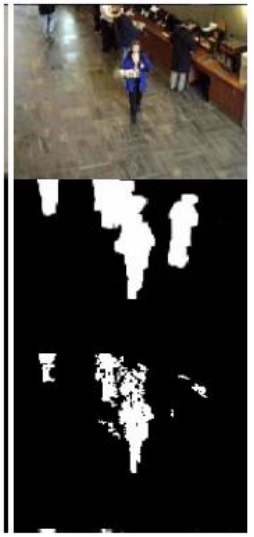
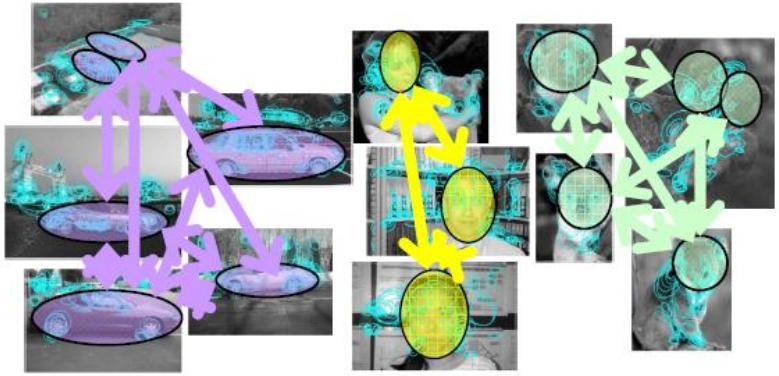


Figure-ground

What things should be grouped?

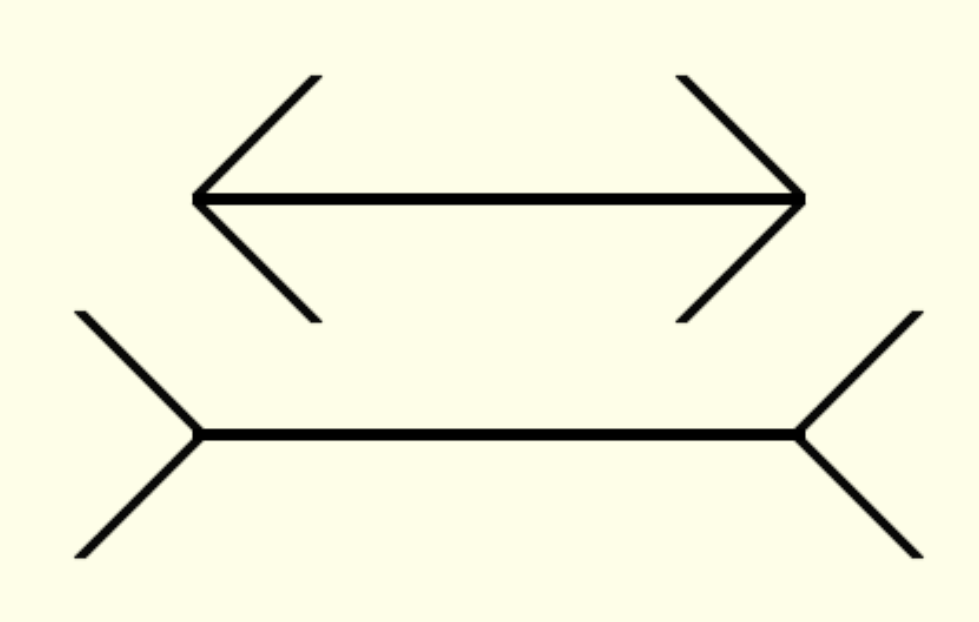
What cues indicate groups?



Inspiration from psychology

- **The Gestalt school: Grouping is key to visual perception**

The Muller-Lyer illusion



http://en.wikipedia.org/wiki/Gestalt_psychology

Gestalt

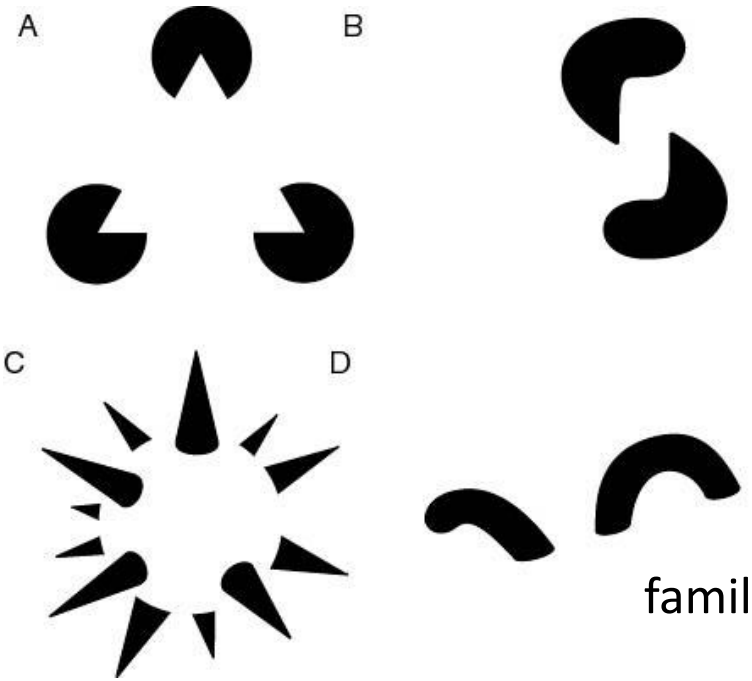
- **Gestalt: whole or group**
  - **Whole is greater than sum of its parts**
  - **Relationships among parts can yield new properties/features**

- **Psychologists identified series of factors that predispose set of elements to be grouped (by human visual system)**

The Gestalt school

- **Elements in a collection can have properties that result from relationships**
  - **"The whole is greater than the sum of its parts"**

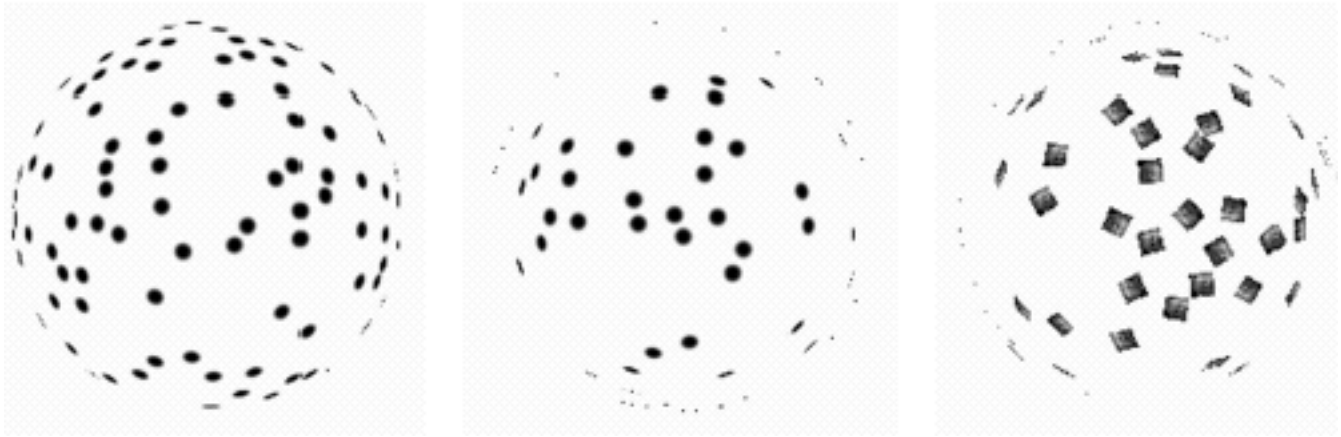
subjective contours



occlusion

familiar configuration

http://en.wikipedia.org/wiki/Gestalt_psychology

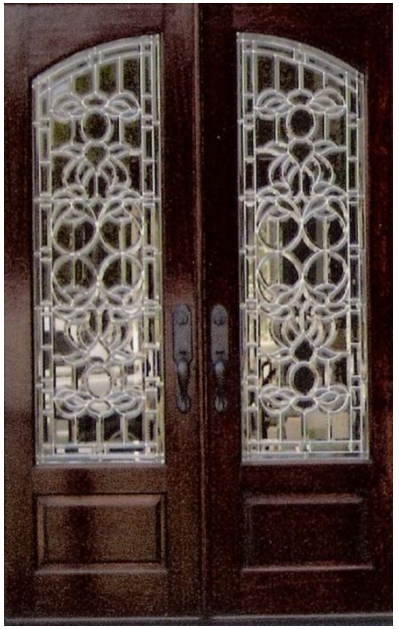


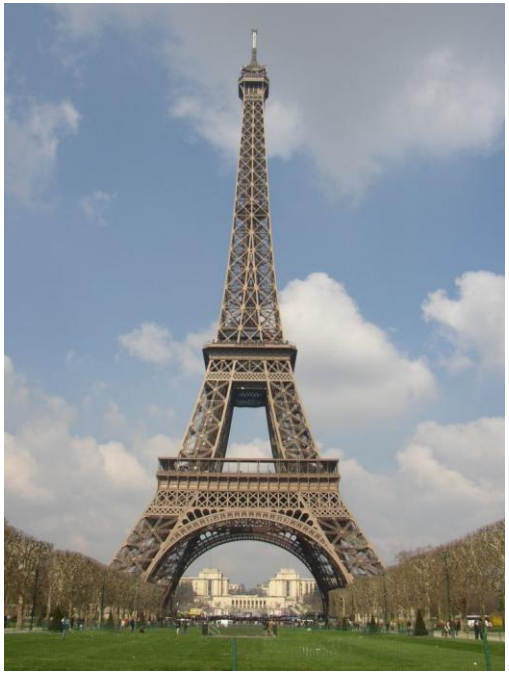
Why do these tokens belong together?

Similarity

http://chicagoist.com/attachments/chicagoist_alicia/GEESE.jpg, http://wwwdelivery.superstock.com/WI/223/1532/PreviewComp/SuperStock_1532R-0831.jpg

Symmetry





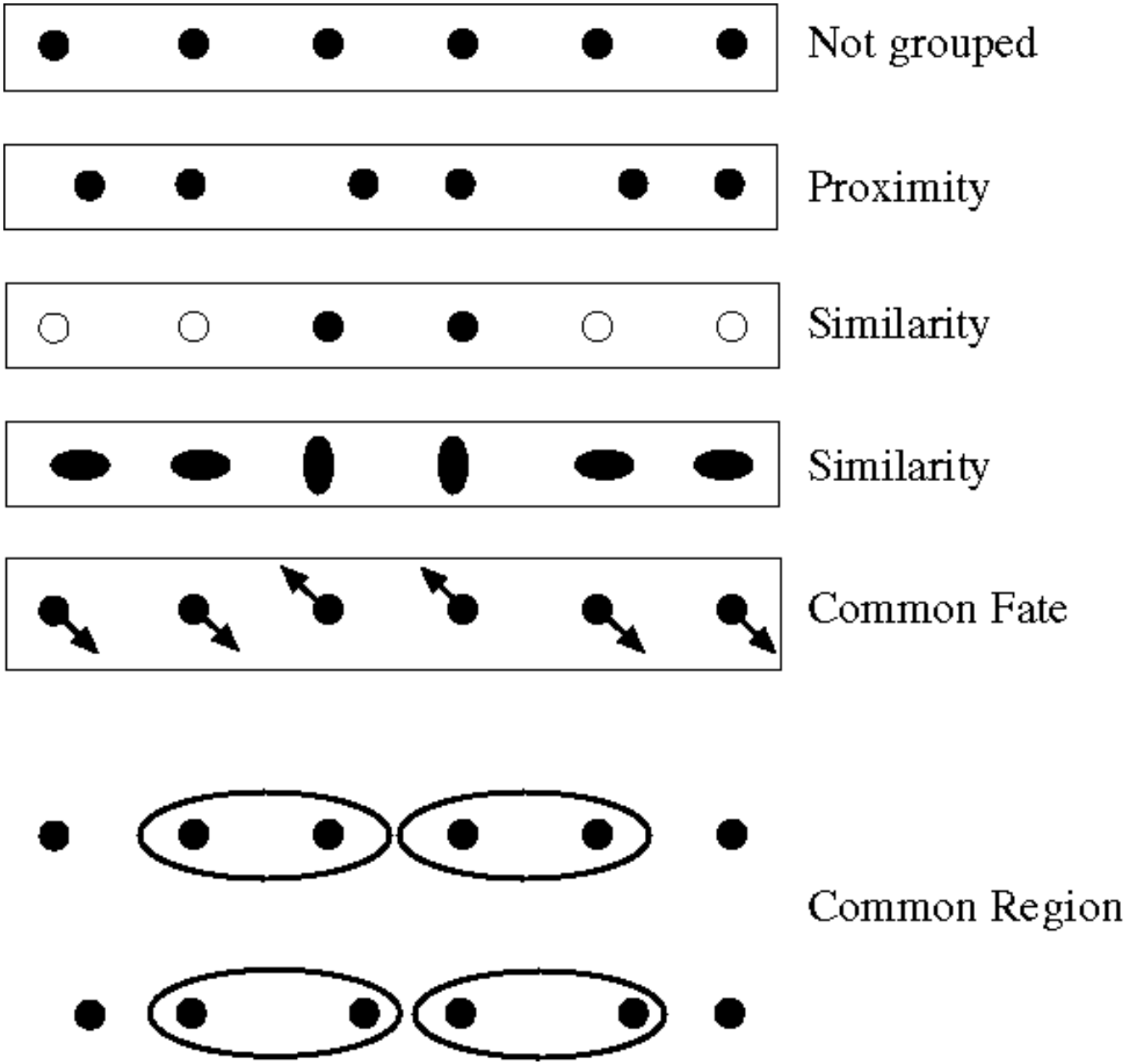
http://seedmagazine.com/news/2006/10/beauty_is_in_the_processingtim.php

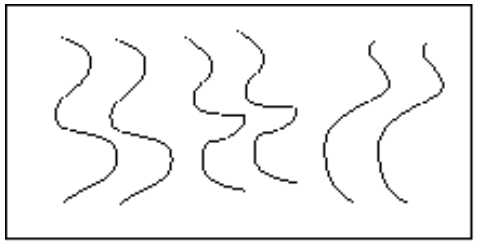
Common fate

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

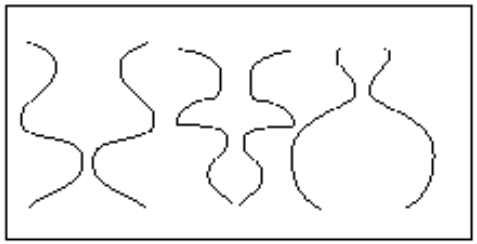
Proximity

http://www.capital.edu/Resources/Images/outside6_035.jpg

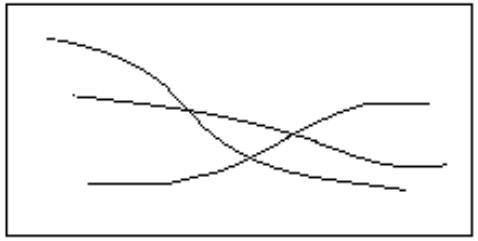

Not grouped
Proximity
Similarity
Similarity
Common Fate
Common Region



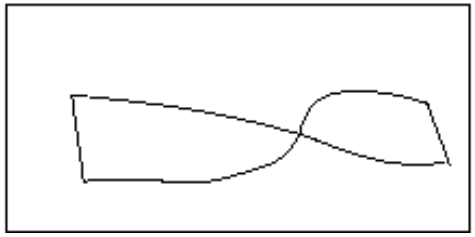
Parallelism



Symmetry

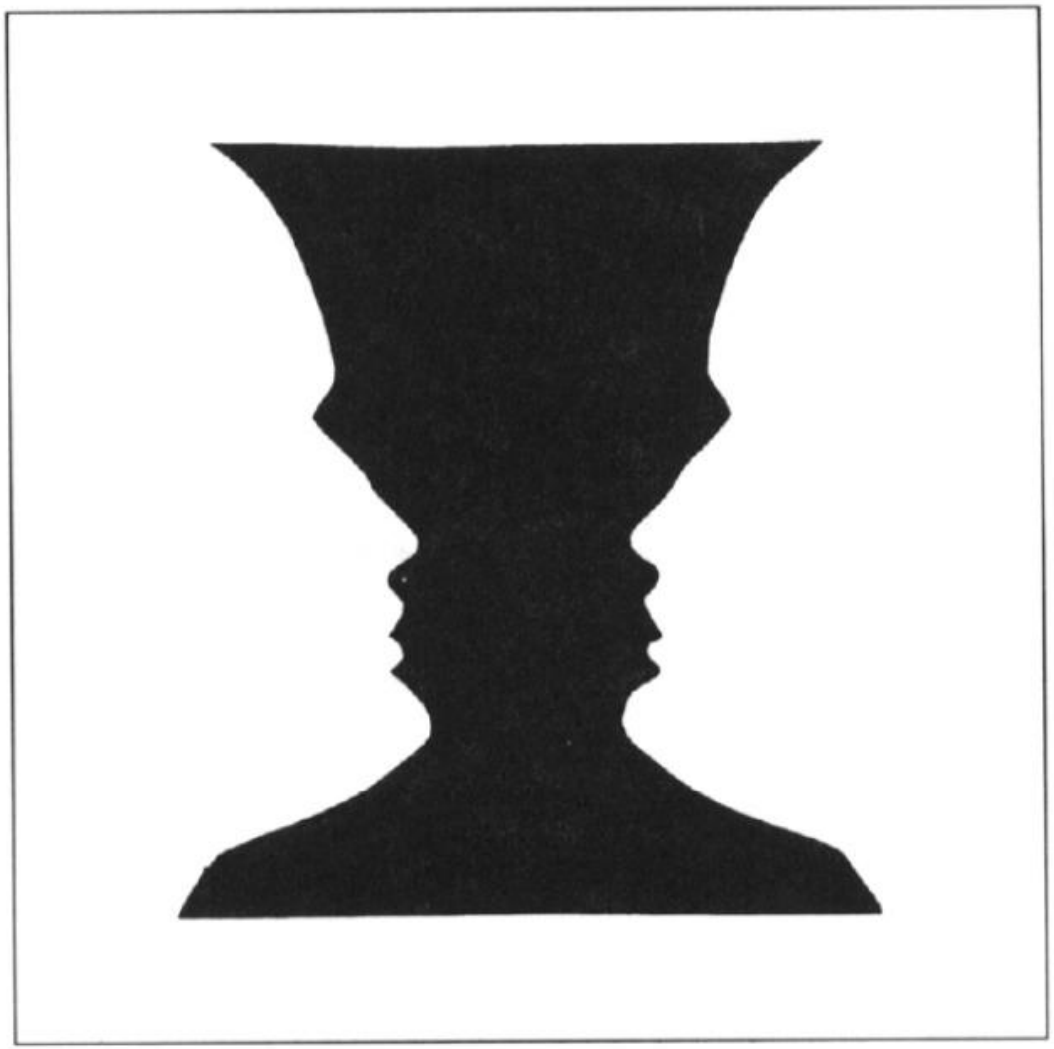


Continuity



Closure

Figure-Ground Discrimination



The Ultimate Gestalt?

Outline for Segmentation

- Introduction
  - Goal of Segmentation
- Segmentation and grouping
- Image Segmentation by Clustering Pixels
  - Segmentation as clustering
  - K-means clustering
  - Mean shift
- Spectral Clustering(Wei Fangyun)
- Integrating top-down and bottom-up segmentation for recognition

Segmentation via Clustering

- **Segmentation:**
  Divide image into regions of similar contents
- **Clustering:**
  Aggregate pixels into regions of similar contents

Segmentation via Clustering

- We speak of segmenting foreground from background
- Segmenting out skin colors
- Segmenting out the moving person
- How do these relate to “similar regions”?

Feature Space

- **Every token is identified by a set of salient visual characteristics called *features*. For example:**
  - **Position**
  - **Intensity or Color**
  - **Texture**
  - **Motion vector**
  - **Size, orientation (if token is larger than a pixel)**
- **The choice of features and how they are quantified implies a *feature space* in which each token is represented by a point**
- **Token similarity is thus measured by distance between points (“feature vectors”) in feature space**

Segmentation and Clustering

- **Defining regions**
  - **Should they be compact? Smooth boundary?**
- **Defining similarity**
  - **Color, texture, motion, …**
- **Defining similarity of regions**
  - **Minimum distance, mean, maximum**

Common similarity/distance measures

- **P-norms**

$$\|\mathbf{x}\|_p := \left(\sum_{i=1}^{n} |x_i|^p \right)^{1/p}$$

  - **City Block (L1)**

$$\|\boldsymbol{x}\|_1 := \sum_{i=1}^{n} |x_i|$$

  - **Euclidean (L2)**

$$\|\boldsymbol{x}\| := \sqrt{x_1^2 + \cdots + x_n^2}$$

  - **L-infinity**

$$\|\mathbf{x}\|_\infty := \max\left(|x_1|, \ldots, |x_n|\right)$$

Here x_i is the distance between two points

- **Mahalanobis**
  - **Scaled Euclidean**

$$d(\vec{x}, \vec{y}) = \sqrt{\sum_{i=1}^{N} \frac{(x_i - y_i)^2}{\sigma_i^2}}$$

- **Cosine distance**

$$\text{similarity} = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|}$$

Simple Agglomerative Clustering

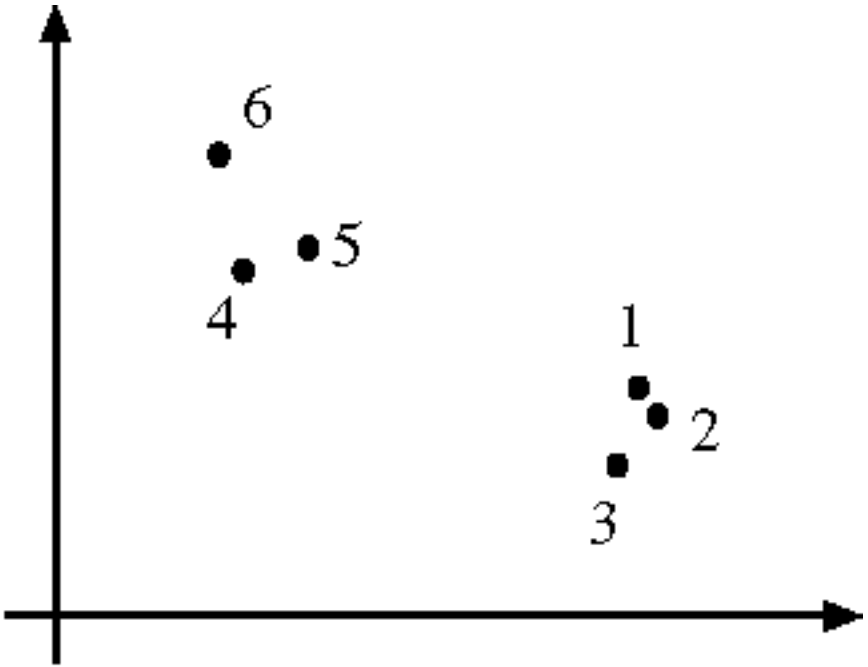
- Start with each pixel in its own cluster
- Iterate:
  - Find pair of clusters with smallest inter-cluster distance
  - Merge
- Stopping threshold

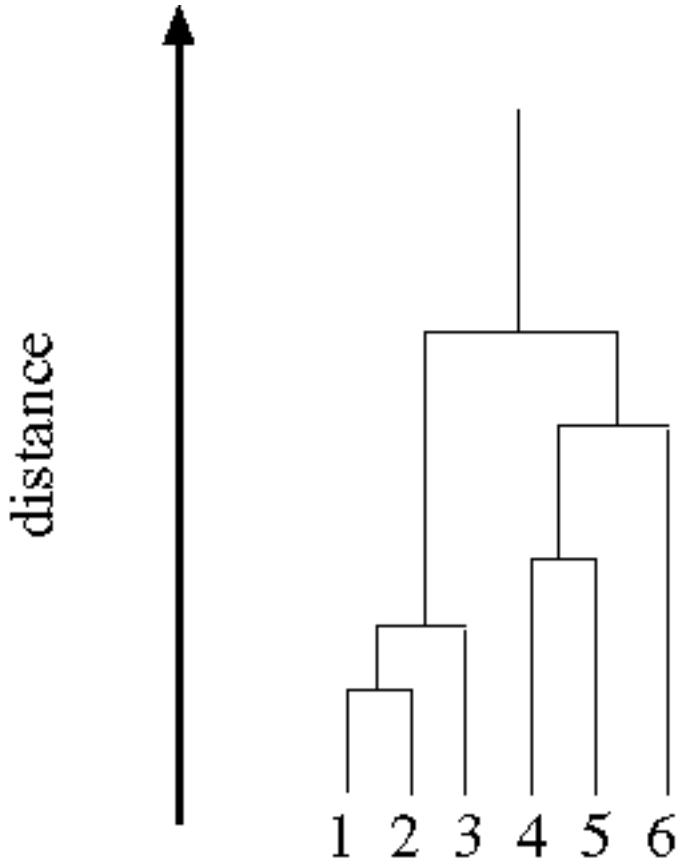
Simple Divisive Clustering

- Start with whole image in one cluster
- Iterate:
  - Find cluster with largest intra-cluster variation
  - Split into two pieces that yield largest inter-cluster distance
- Stopping threshold

Segmentation as Clustering

- **Point-Cluster distance**
  - **single-link clustering**
  - **complete-link clustering**
  - **group-average clustering**
- **Dendrograms (树状图)**
  - **yield a picture of output as clustering process continues**

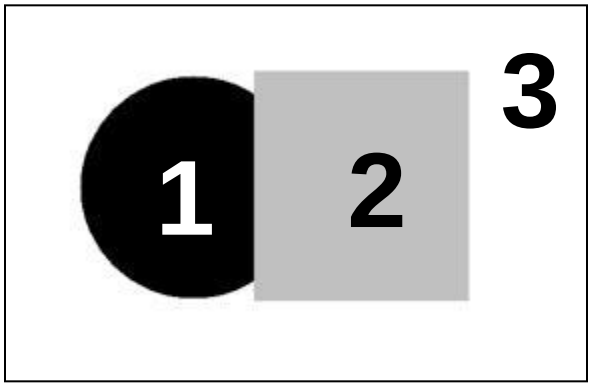




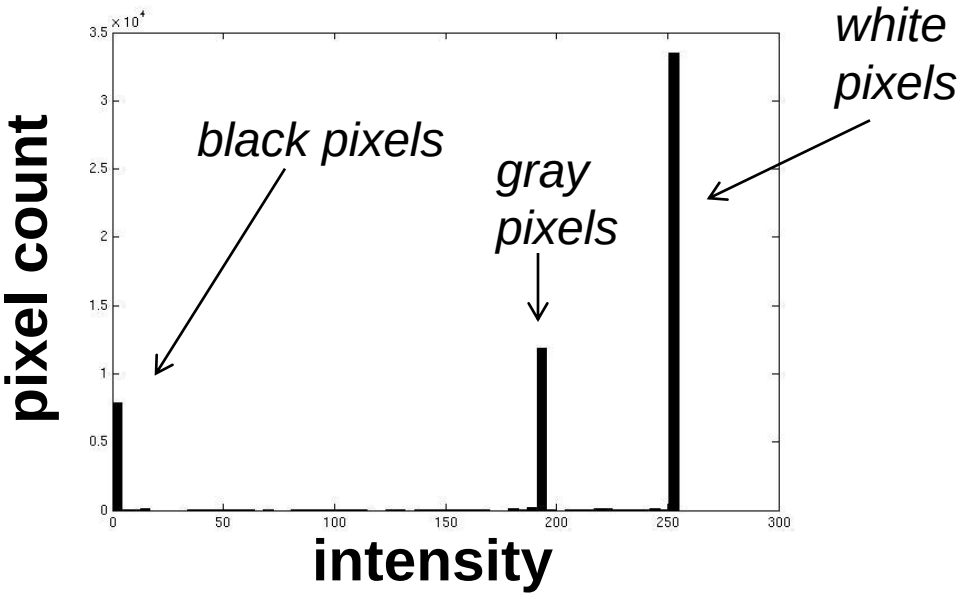
Difficulties with Simple Clustering

- Many possibilities at each iteration
- Computing distance between clusters or optimal split expensive
- Heuristics to speed this up:
  - For agglomerative clustering, approximate each cluster by average for distance computations
  - For divisive clustering, use summary (histogram) of a region to compute split

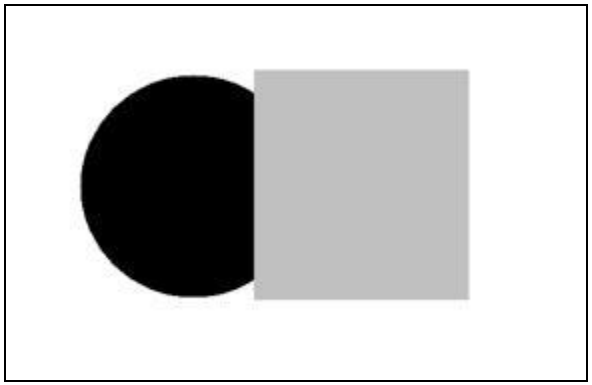
Image segmentation: toy example



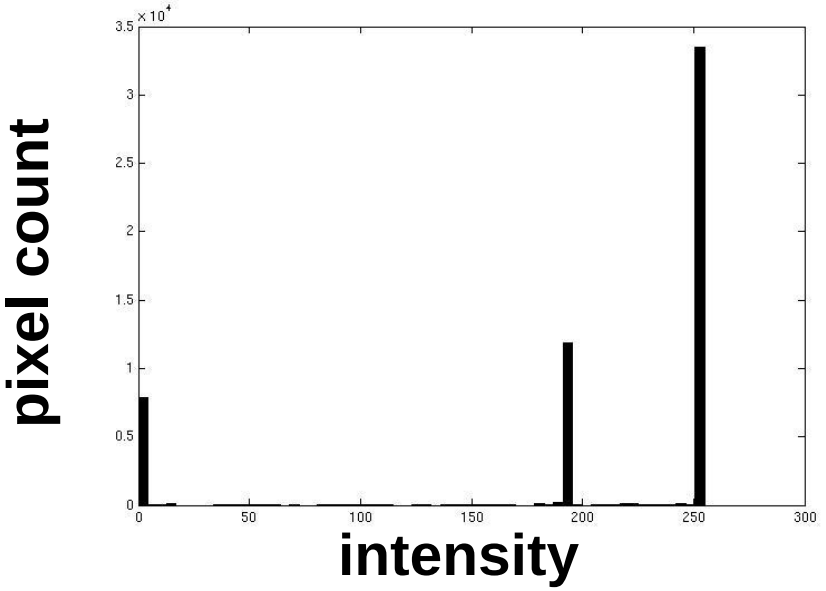
input image

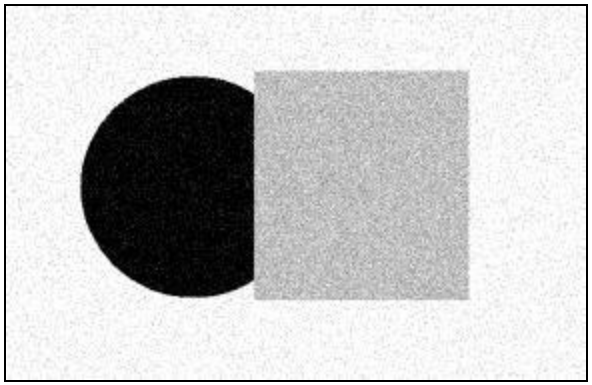


- **These intensities define the three groups.**
- **We could label every pixel in the image according to which of these primary intensities it is.**
  - **i.e., *segment* the image based on the intensity feature.**
- **What if the image isn't quite so simple?**

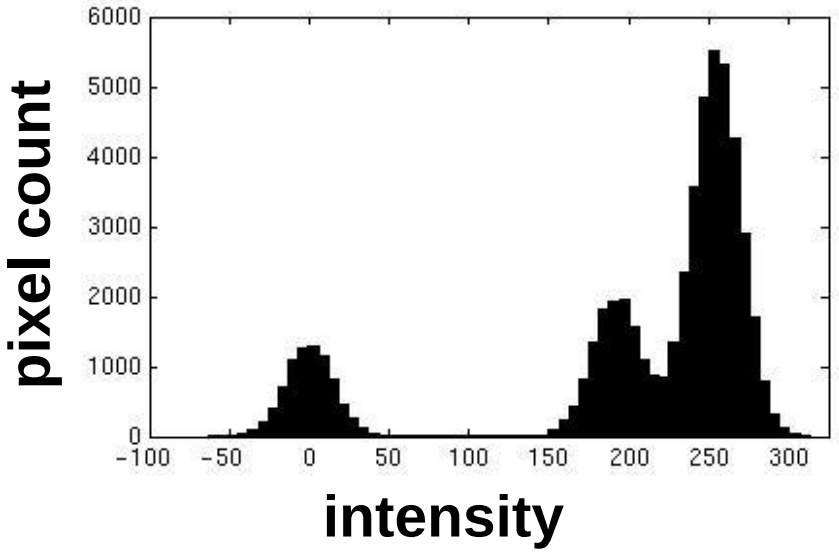


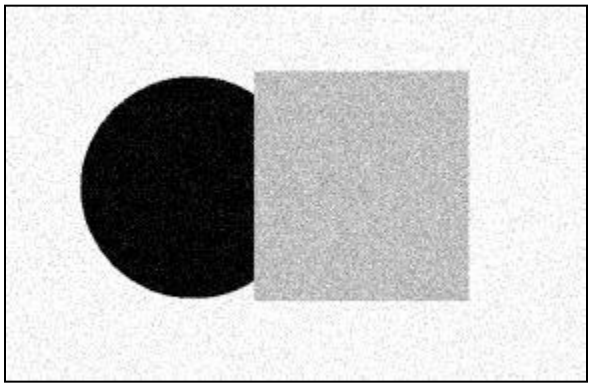
input image



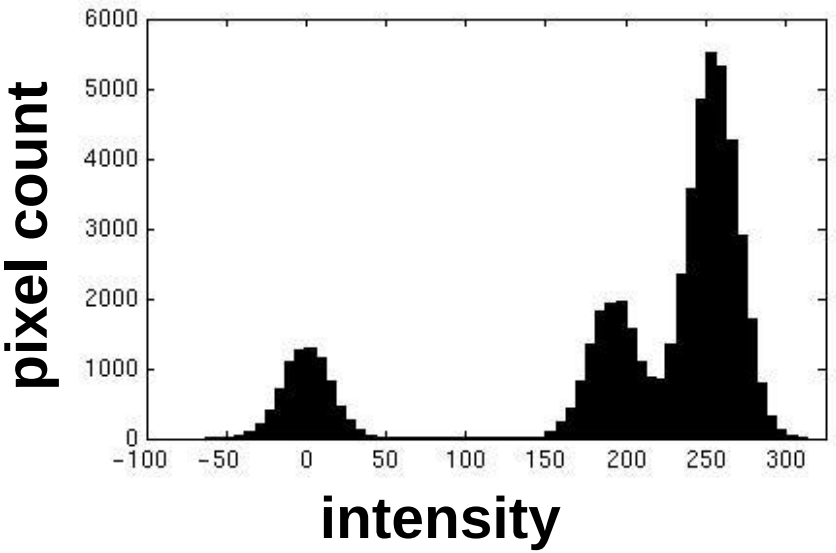


input image

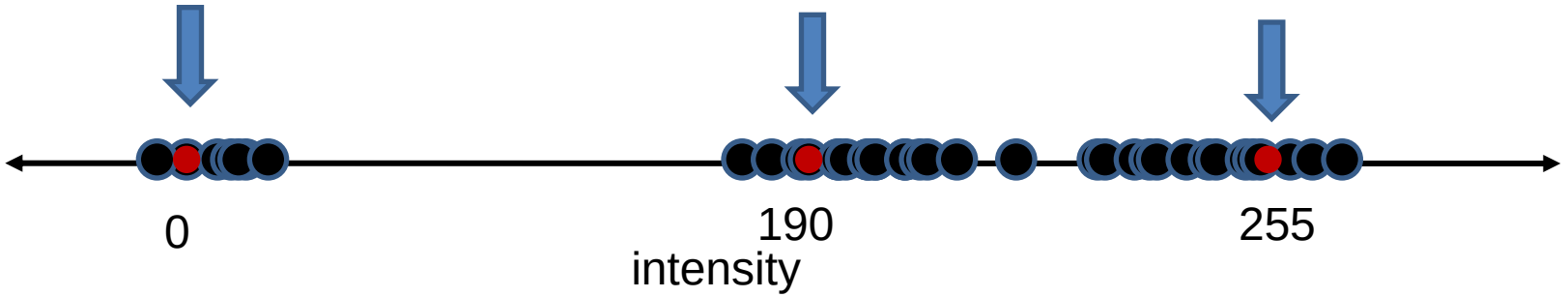


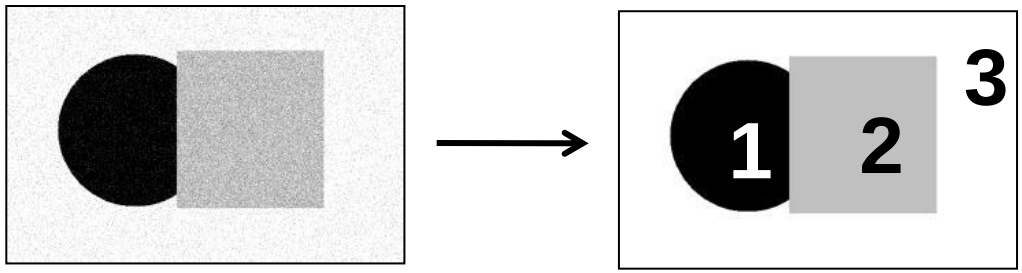


input image



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



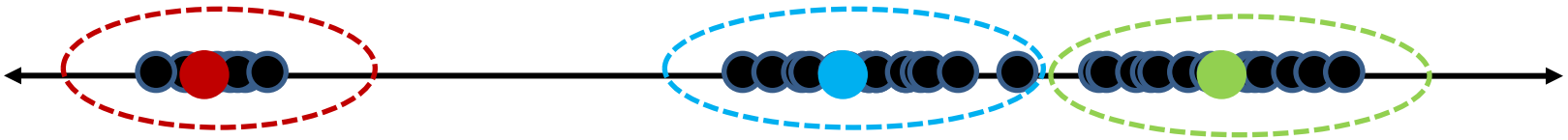


- **Goal: choose three "centers" as the representative intensities, and label every pixel according to which of these centers it is nearest to.**
- **Best cluster centers are those that minimize SSD between all points and their nearest cluster center c_i:**

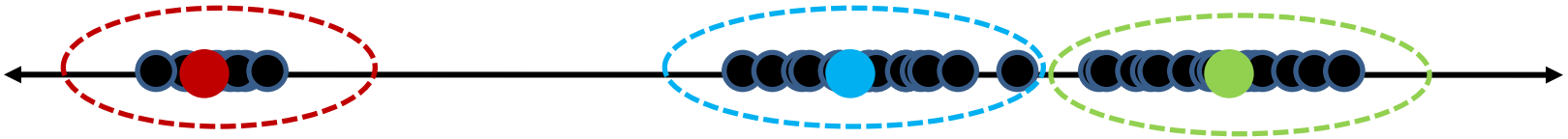
$$\sum_{\text{clusters } i} \sum_{\text{points p in cluster } i} \|p - c_i\|^2$$

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.



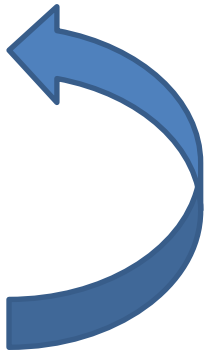
Outline for Segmentation

- Introduction
  - Goal of Segmentation
- Segmentation and grouping
- Image Segmentation by Clustering Pixels
  - Segmentation as clustering
  - K-means clustering
  - Mean shift
- Spectral Clustering(Wei Fangyun)
- Integrating top-down and bottom-up segmentation for recognition

K-means clustering

- **Basic idea: randomly initialize the *k* cluster centers, and iterate between the two steps we just saw.**

  1. **Randomly initialize the cluster centers, c_1, ..., 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**



Properties

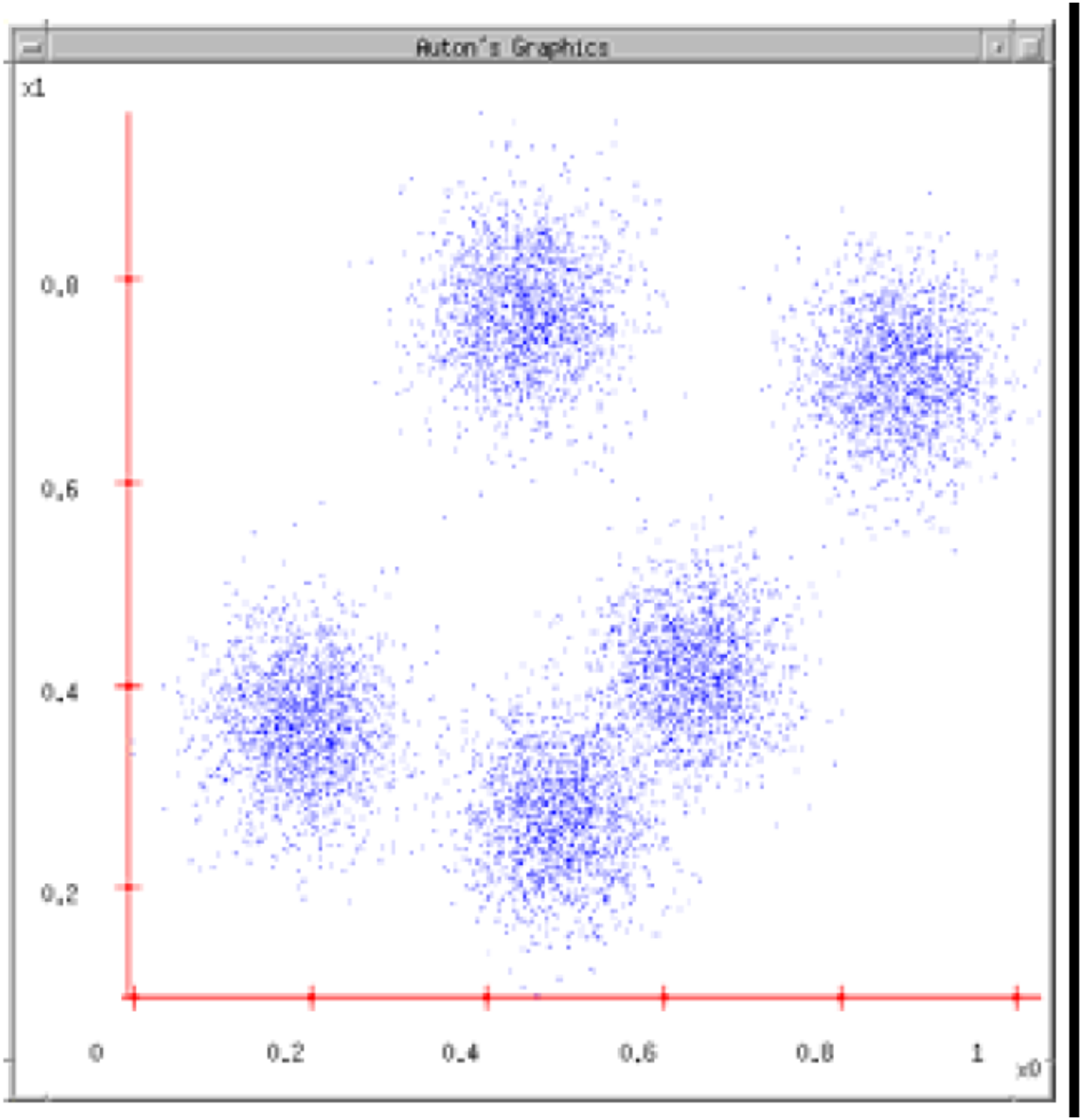
- **Will always converge to *some* solution**
- **Can be a "local minimum"**
  - **does not always find the global minimum of objective function:**

$$\sum_{\text{clusters } i} \sum_{\text{points p in cluster } i} \|p - c_i\|^2$$

Source: Steve Seitz

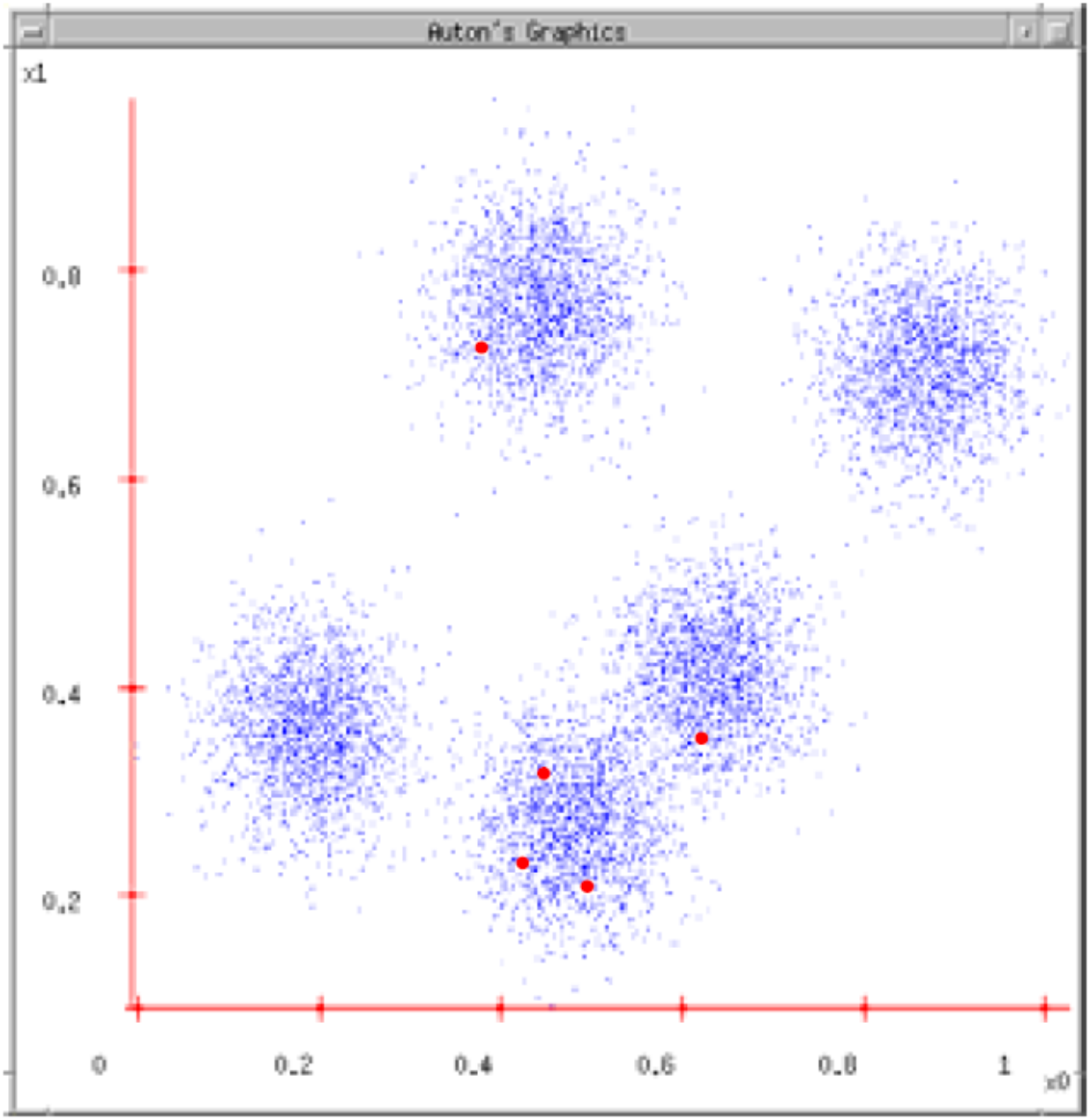
K-means

1. Ask user how many clusters they'd like. *(e.g. k=5)*



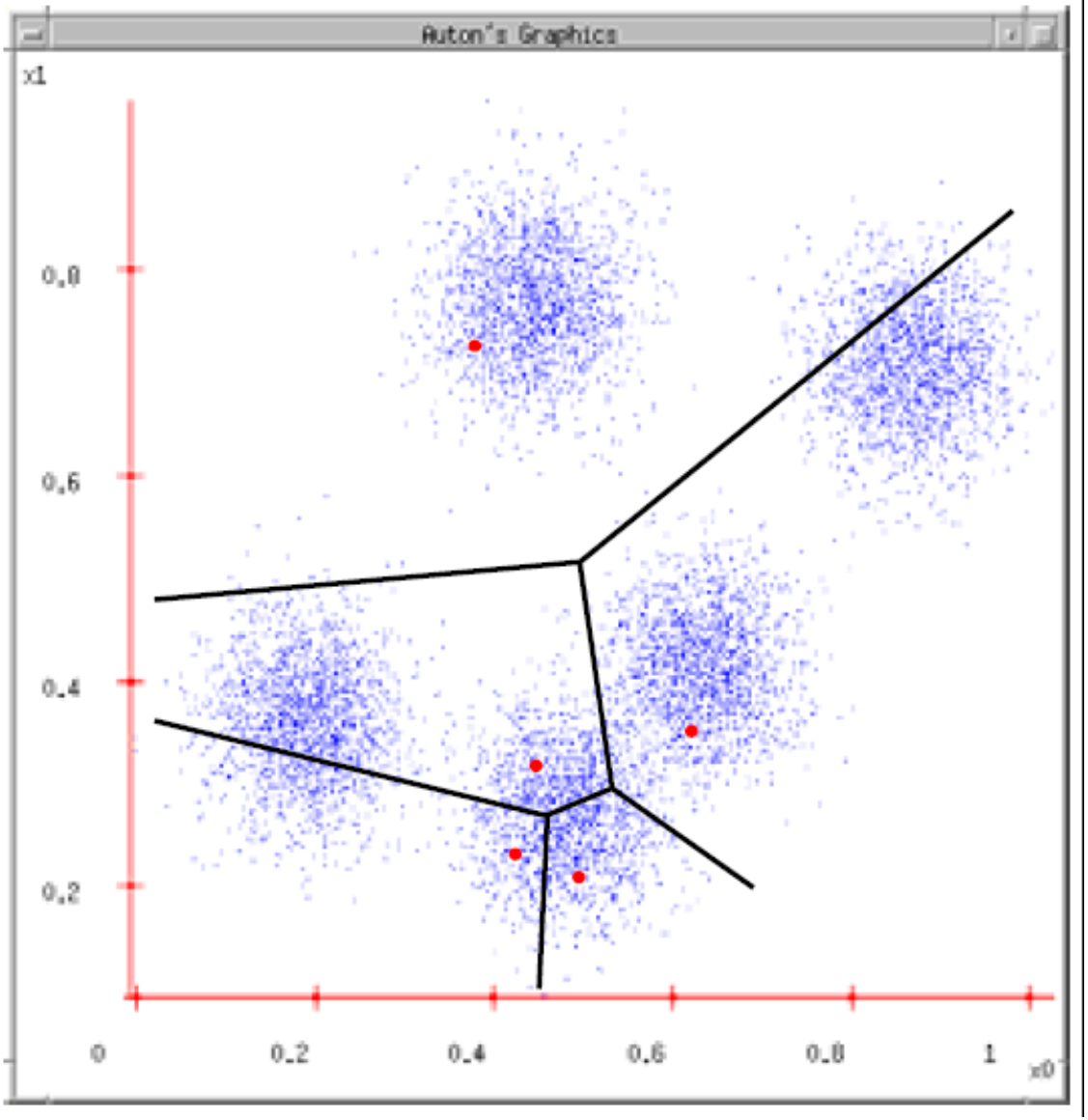
K-means

1. Ask user how many clusters they'd like. *(e.g. k=5)*
2. Randomly guess k cluster Center locations



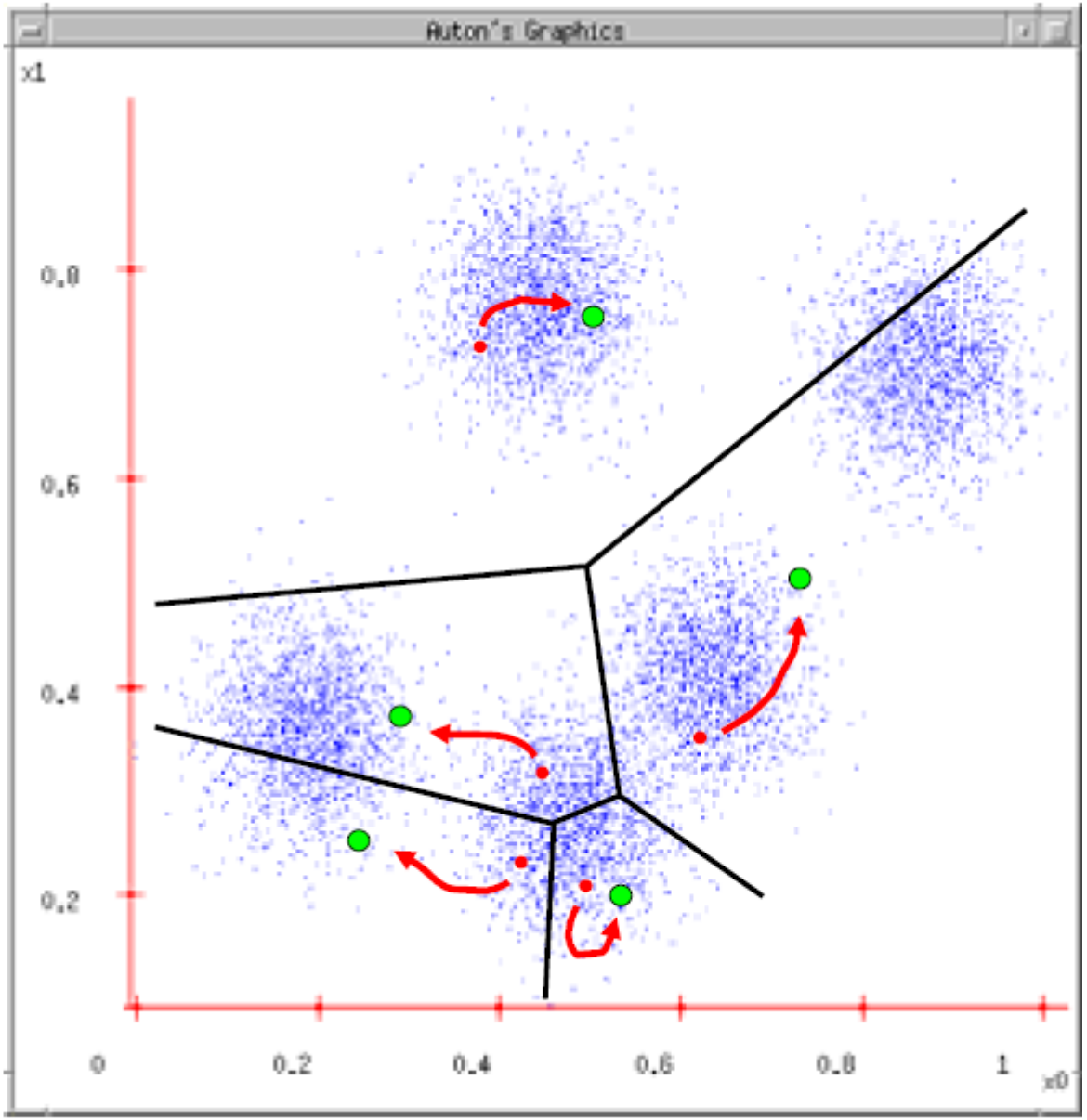
K-means

1. Ask user how many clusters they'd like. *(e.g. k=5)*
2. Randomly guess k cluster Center locations
3. Each datapoint finds out which Center it's closest to. (Thus each Center "owns" a set of datapoints)



K-means

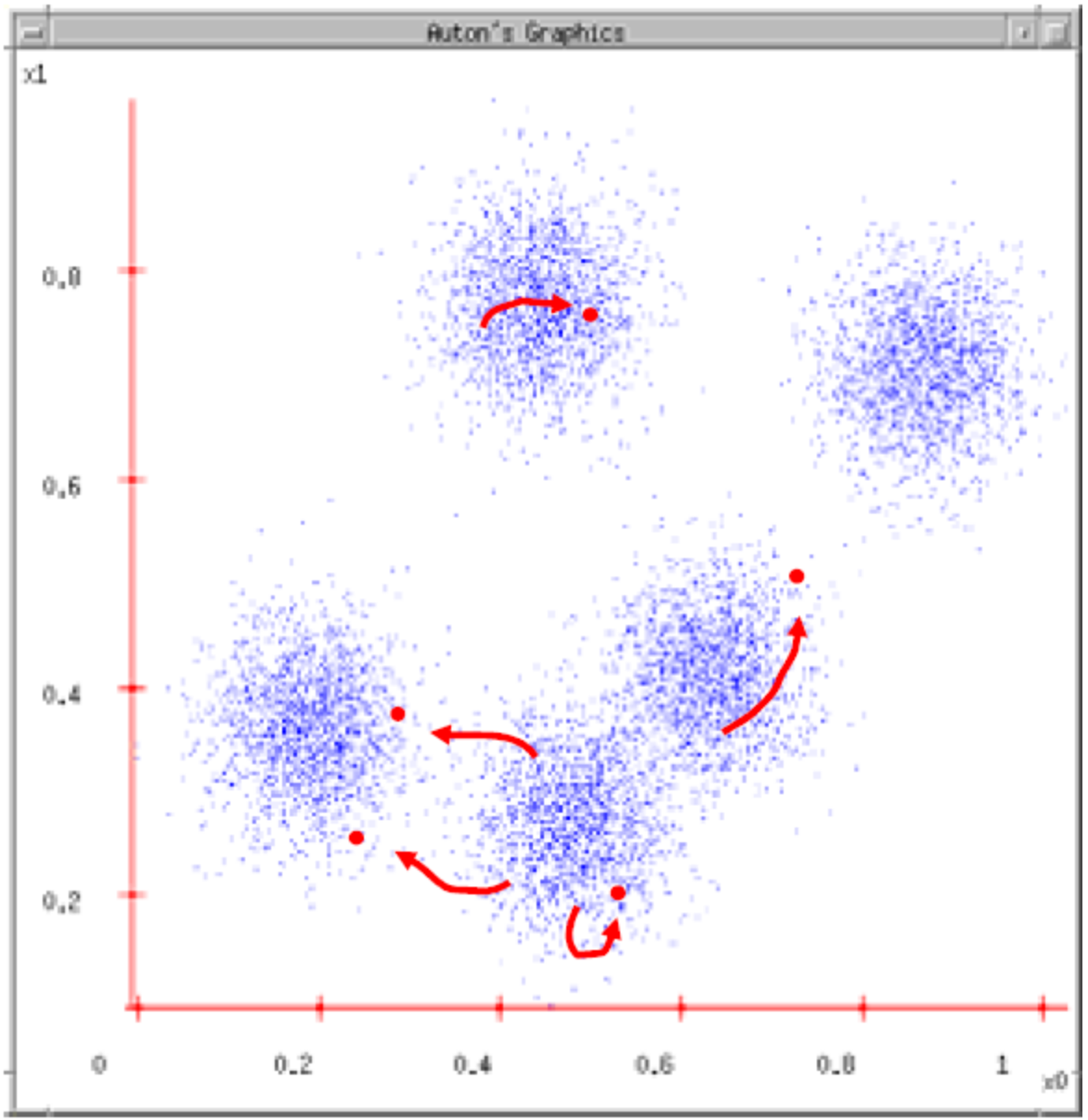
1. Ask user how many clusters they'd like. *(e.g. k=5)*
2. Randomly guess k cluster Center locations
3. Each datapoint finds out which Center it's closest to.
4. Each Center finds the centroid of the points it owns



Andrew Moore

K-means

1. Ask user how many clusters they'd like. *(e.g. k=5)*
2. Randomly guess k cluster Center locations
3. Each datapoint finds out which Center it's closest to.
4. Each Center finds the centroid of the points it owns…
5. …and jumps there
6. …Repeat until terminated!



Andrew Moore

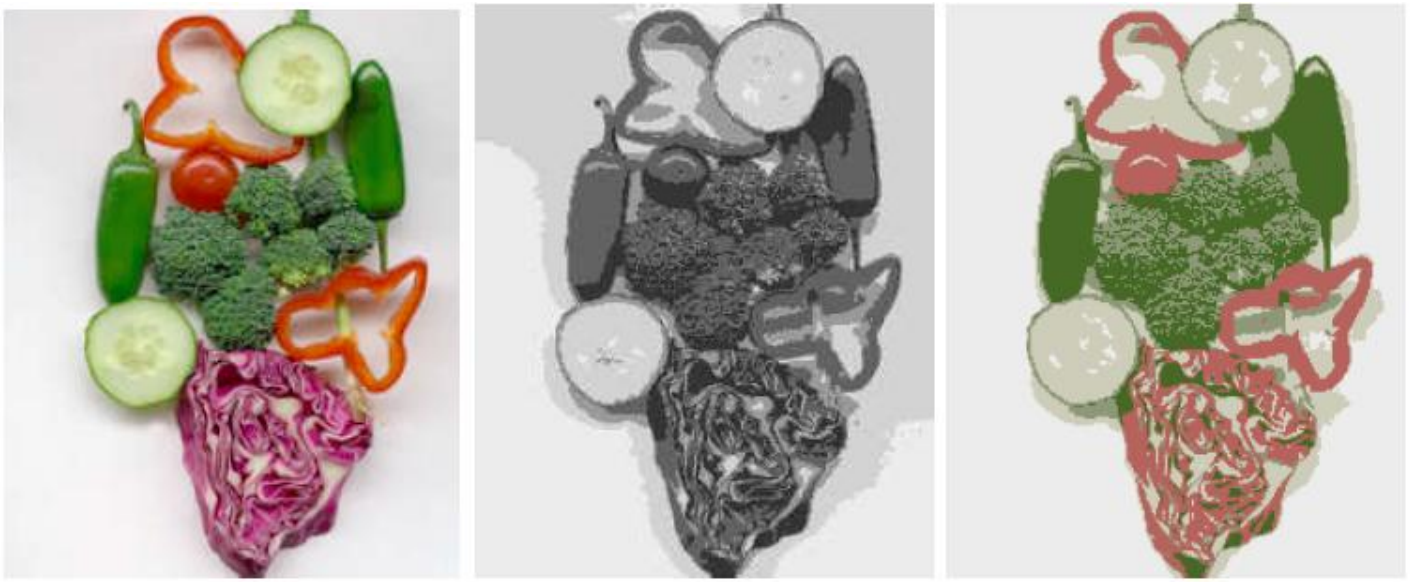


FIGURE 9.17: On the **left**, an image of mixed vegetables, which is segmented using k-means to produce the images at **center** and on the **right**. We have replaced each pixel with the mean value of its cluster; the result is somewhat like an adaptive requantization, as one would expect. In the center, a segmentation obtained using only the intensity information. At the right, a segmentation obtained using color information. Each segmentation assumes five clusters.

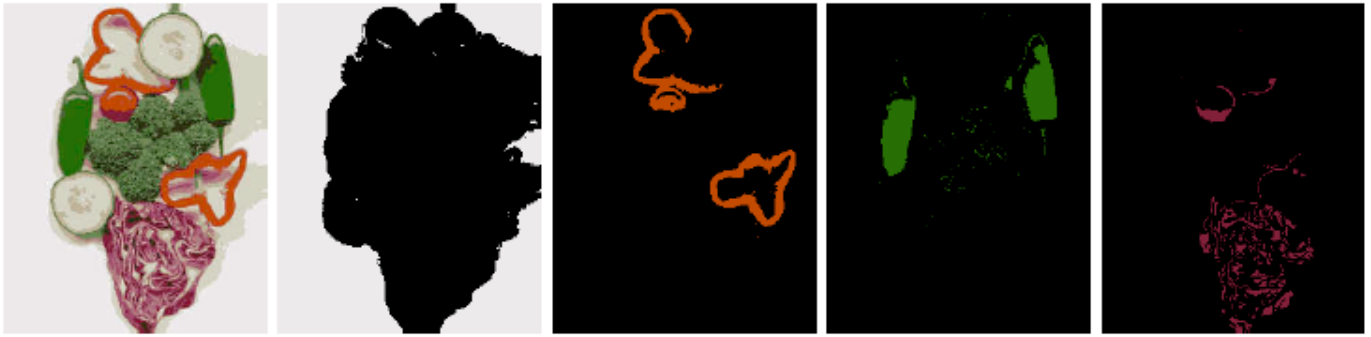


FIGURE 9.18: Here we show the image of vegetables segmented with k-means, assuming a set of 11 components. The left figure shows all segments shown together, with the mean value in place of the original image values. The other figures show four of the segments. Note that this approach leads to a set of segments that are not necessarily connected. For this image, some segments are actually quite closely associated with objects, but one segment may represent many objects (the peppers); others are largely meaningless. The absence of a texture measure creates serious difficulties, as the many different segments resulting from the slice of red cabbage indicate.

FIGURE 9.19: Five of the segments obtained by segmenting the image of vegetables with a k-means segmenter that uses position as part of the feature vector describing a pixel, now using 20 segments rather than 11. Note that the large background regions that should be coherent have been broken up because points got too far from the center. The individual peppers are now better separated, but the red cabbage is still broken up because there is no texture measure.

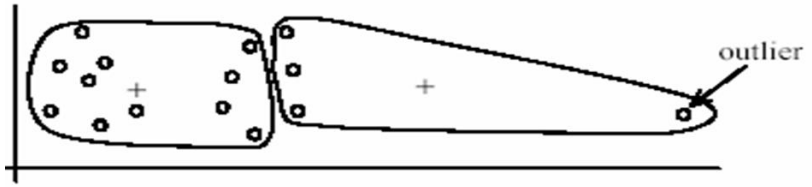
K-means: pros and cons

Pros

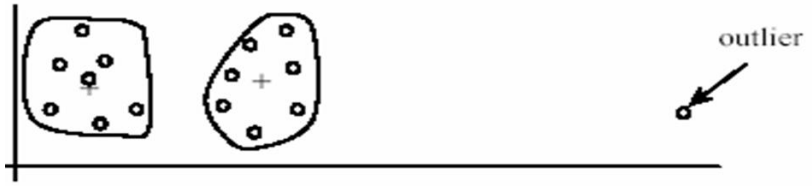
- Simple, fast to compute
- Converges to local minimum of within-cluster squared error

Cons/issues

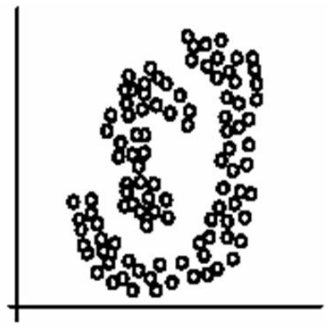
- Setting k?
- Sensitive to initial centers
- Sensitive to outliers
- Detects spherical clusters
- Assuming means can be computed



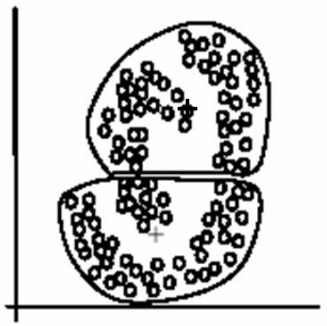
(A): Undesirable clusters



(B): Ideal clusters



(A): Two natural clusters

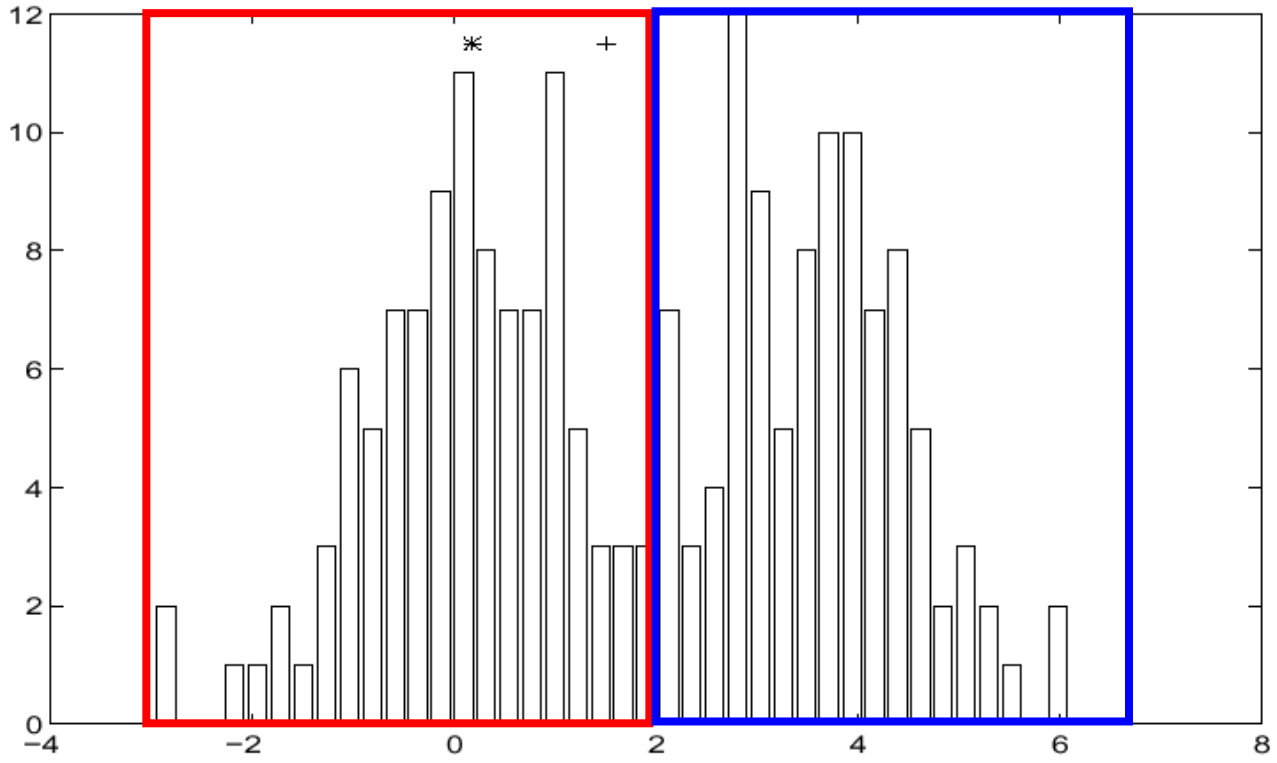


(B): *k*-means clusters

Outline for Segmentation

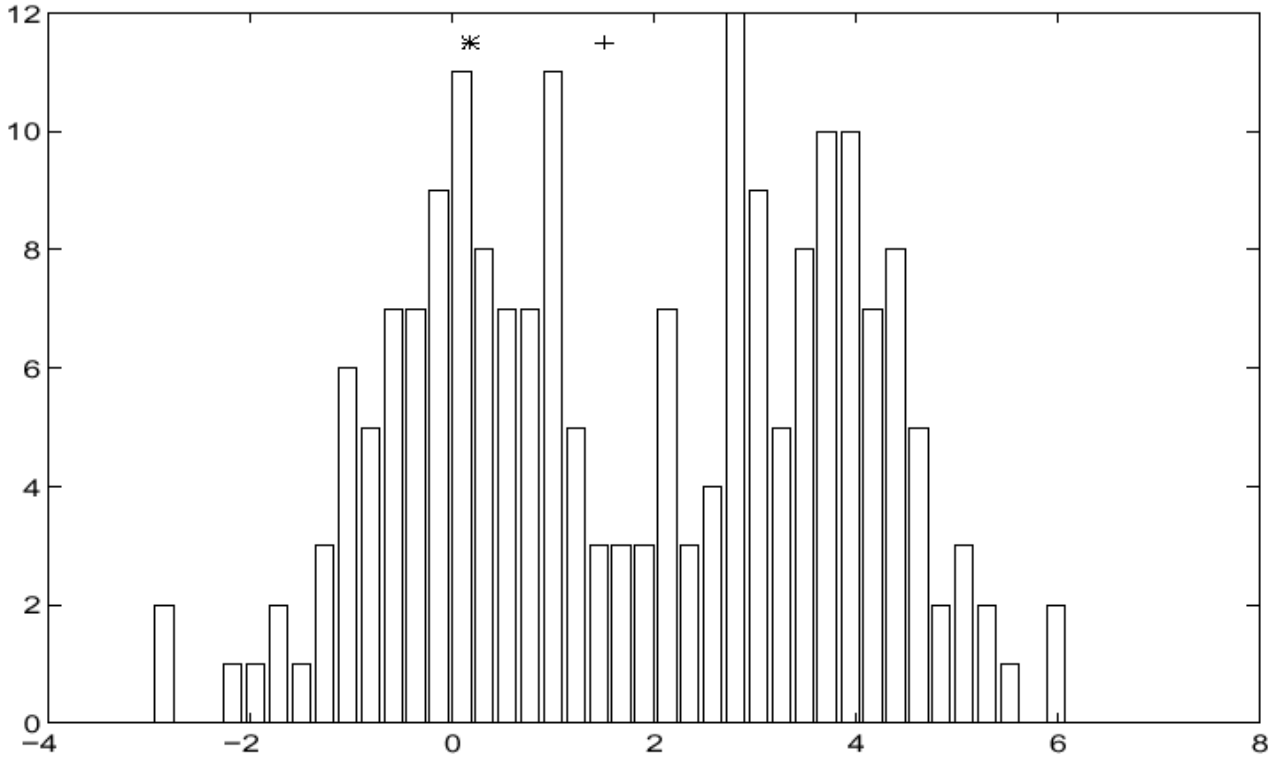
- Introduction
  - Goal of Segmentation
- Segmentation and grouping
- Image Segmentation by Clustering Pixels
  - Segmentation as clustering
  - K-means clustering
  - Mean shift
- Segmentation， clustering and Graph
  - Normalized Cut
  - Graph cut
- Integrating top-down and bottom-up segmentation for recognition

Finding Modes in a Histogram



- **How Many Modes Are There?**
  - **Easy to see, hard to compute**

Mean Shift [Comaniciu & Meer*]



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

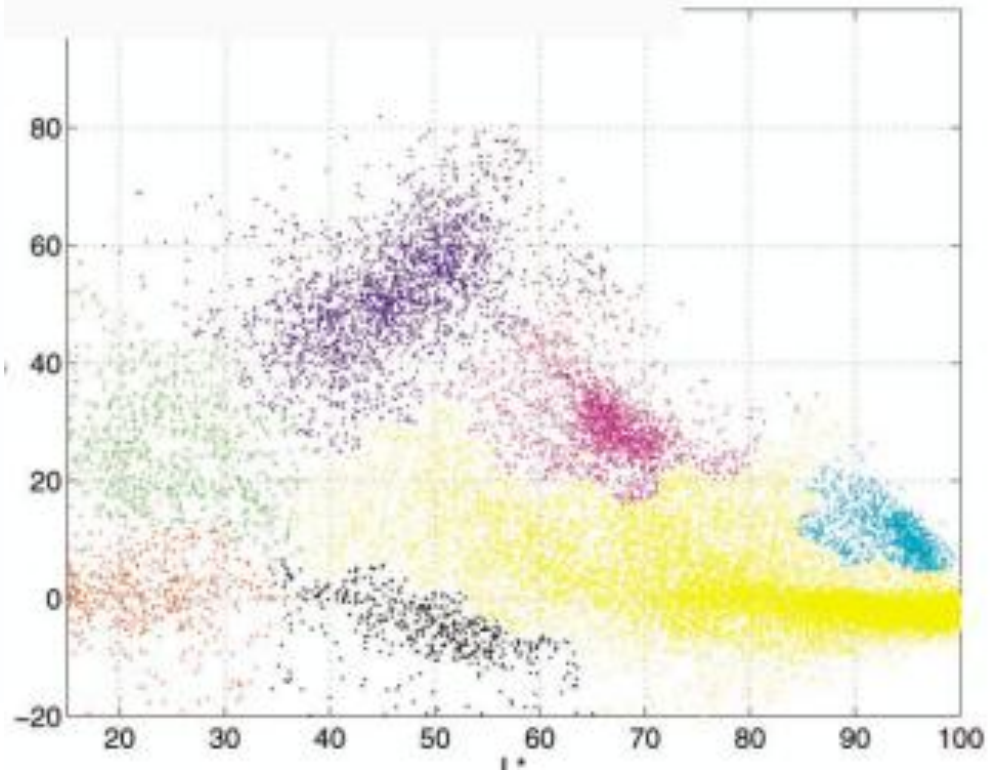
*D. Comaniciu and P. Meer, Mean Shift: A Robust Approach toward Feature Space Analysis, PAMI 2002.

Mean shift algorithm

- **The mean shift algorithm seeks *modes* or local maxima of density in the feature space**

image

Feature space (L*u*v* color values)



Kernel density estimation (KDE)

- A non-parametric way to estimate the probability density function of a random variable.
- Inferences about a population are made based only on a finite data sample.
- Also termed the Parzen–Rosenblatt window method,
    - Named fter Emanuel Parzen and Murray Rosenblatt, who are usually credited with independently creating it in its current form

Kernel density estimation

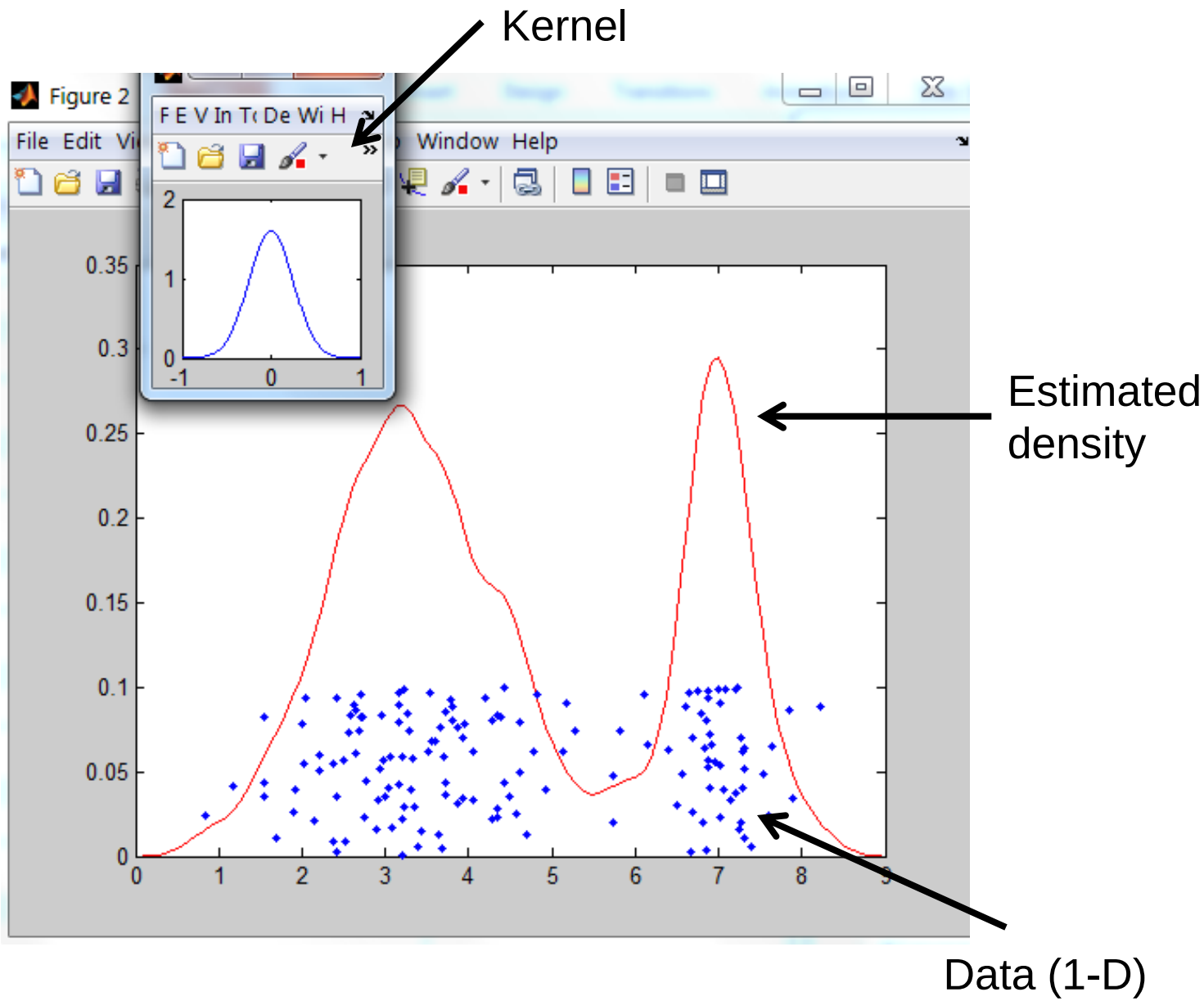
Kernel density estimation function

$$\widehat{f}_h(x) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right)$$

Gaussian kernel

$$K\left(\frac{x - x_i}{h}\right) = \frac{1}{\sqrt{2\pi}} e^{-\frac{(x-x_i)^2}{2h^2}}.$$

Kernel density estimation



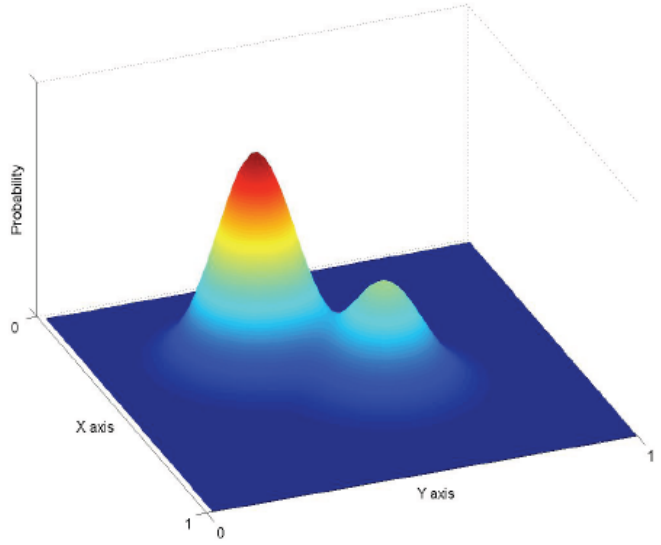
Meanshift Clustering

- Assumption:

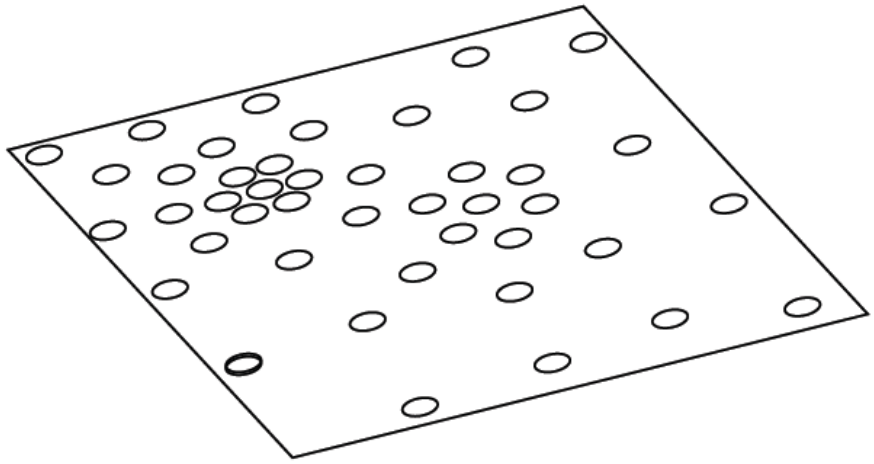
  There is an underlying pdf governing data properties in R^N

- Clustering based on estimating the underling pdf

- Each cluster corresponds to one mode of the pdf



assumed pdf



real data samples

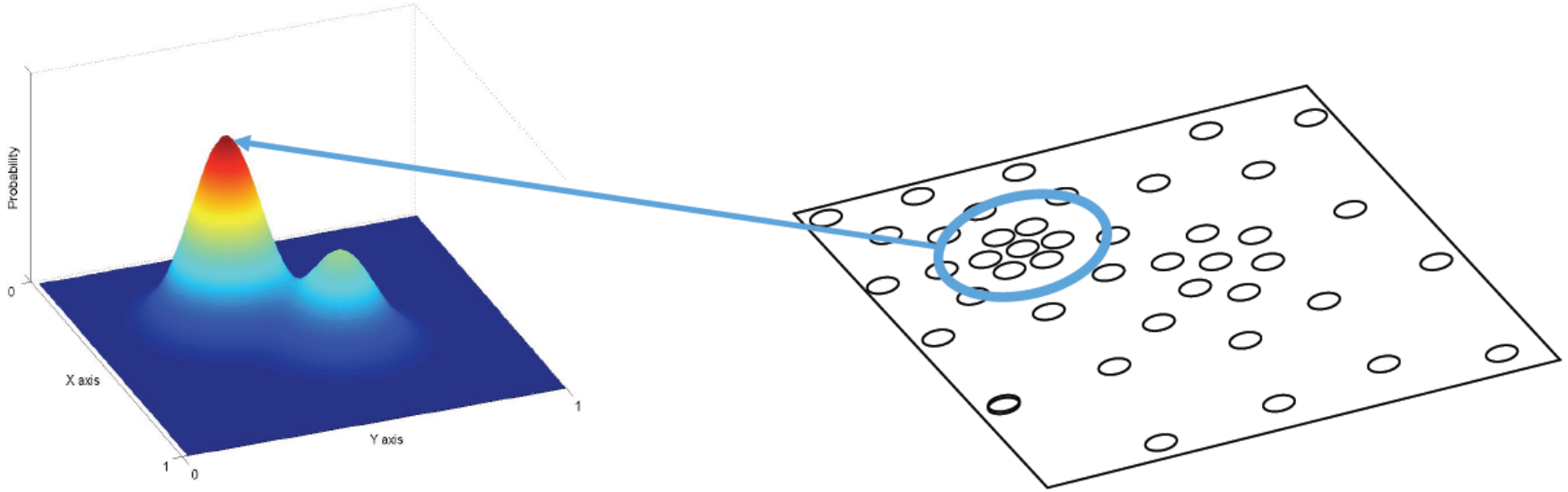
Meanshift Clustering

- Assumption:

  There is an underlying pdf governing data properties in R^N

- Clustering based on estimating the underling pdf

- Each cluster corresponds to one mode of the pdf



assumed pdf

real data samples

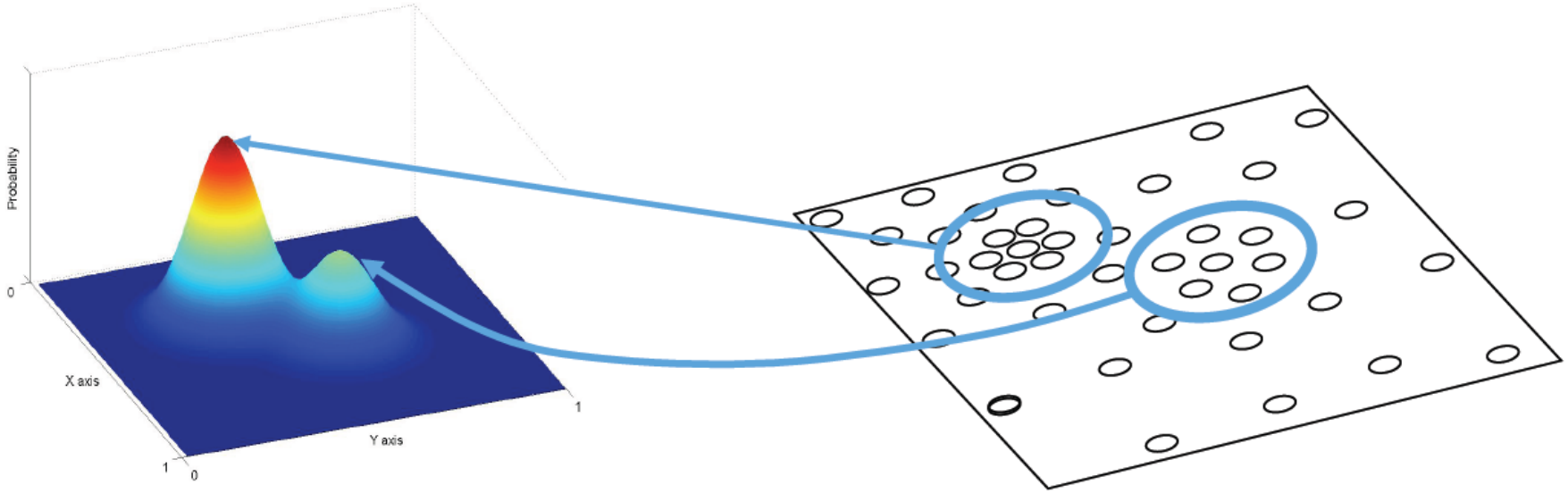
Meanshift Clustering

- Assumption:

  There is an underlying pdf governing data properties in R^N

- Clustering based on estimating the underling pdf

- Each cluster corresponds to one mode of the pdf



assumed pdf

real data samples

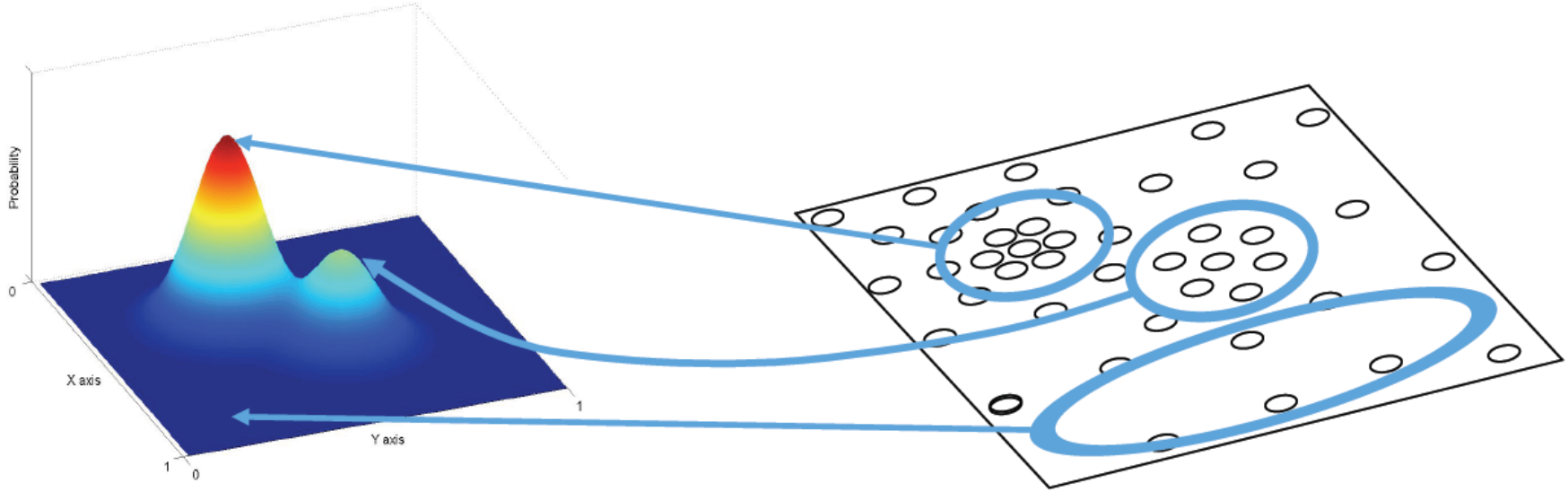
Meanshift Clustering

- Assumption:

  There is an underlying pdf governing data properties in R^N

- Clustering based on estimating the underling pdf

- Each cluster corresponds to one mode of the pdf



assumed pdf

real data samples

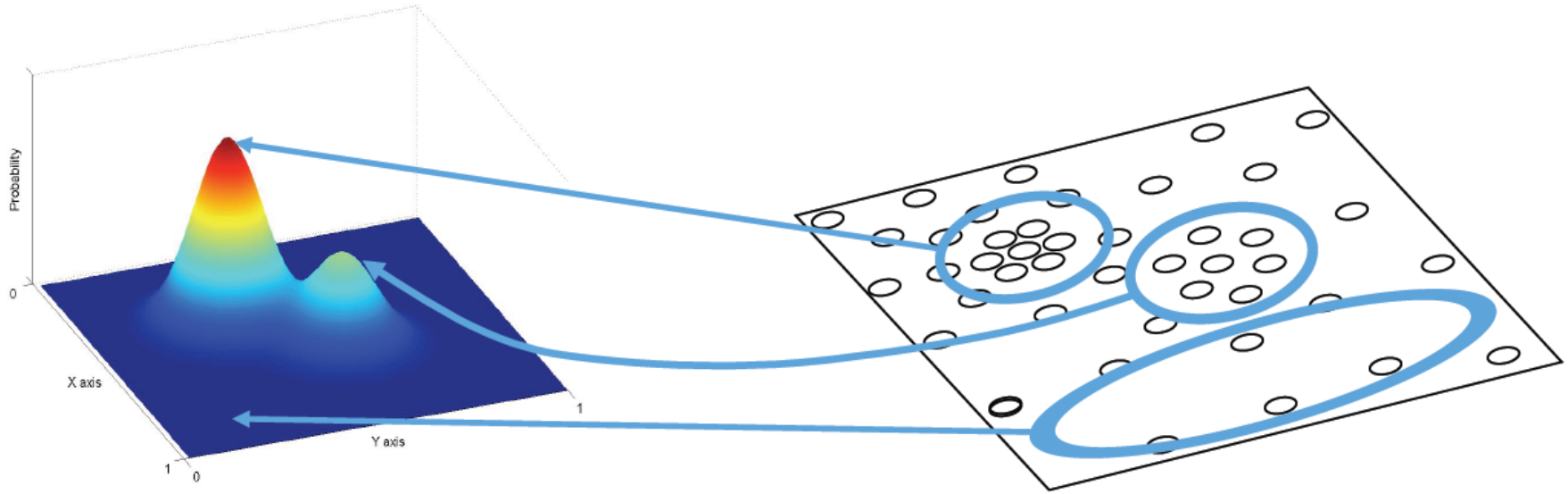
Parametric Density Estimation

- Assumption:

  The functional form of the pdf is known

- Example:

  Mixture of Gaussians $\sum_{k=1}^{K} C_k \exp\left(-\frac{1}{2}(x-\mu_k)^T \Sigma_k^{-1}(x-\mu_k)\right)$



assumed pdf

real data samples

Parametric Density Estimation

- Assumption:

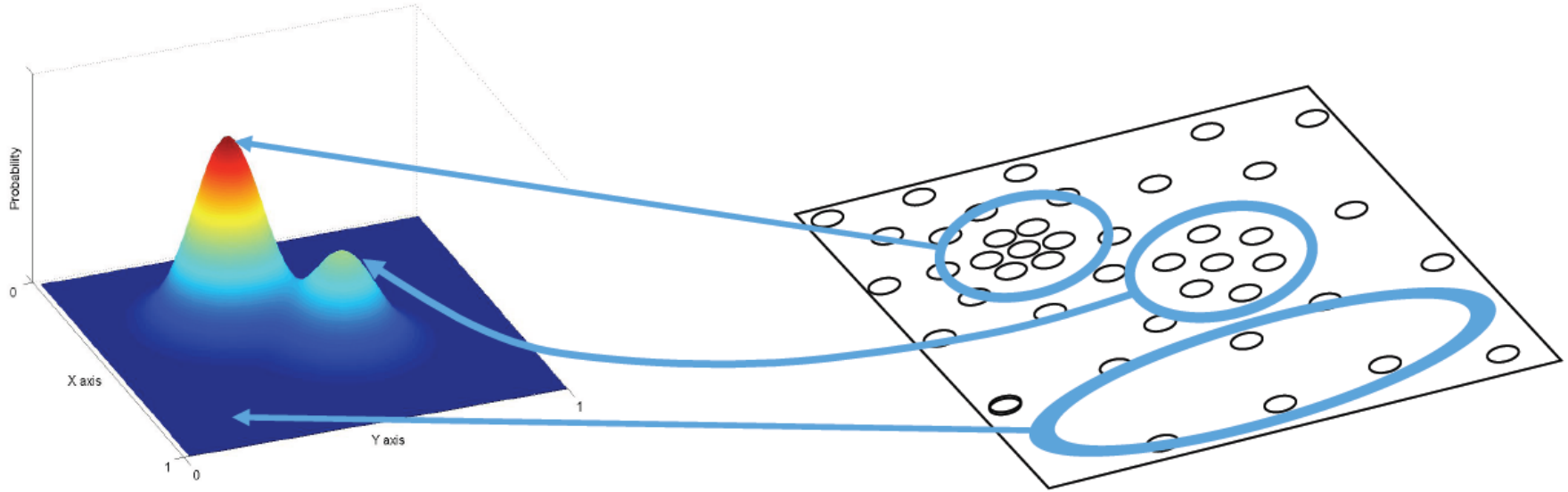
  The functional form of the pdf is known

- Example:

  Mixture of Gaussians

$$\sum_{k=1}^{K} C_k \exp\left(-\frac{1}{2}(x-\mu_k)^T \Sigma_k^{-1}(x-\mu_k)\right)$$

estimate



assumed pdf

real data samples

Nonparameteric Density Estimation

- Given N data samples
- Estimate the pdf, P(x), at some value x by
- "Counting" data samples, x_n, "around" x

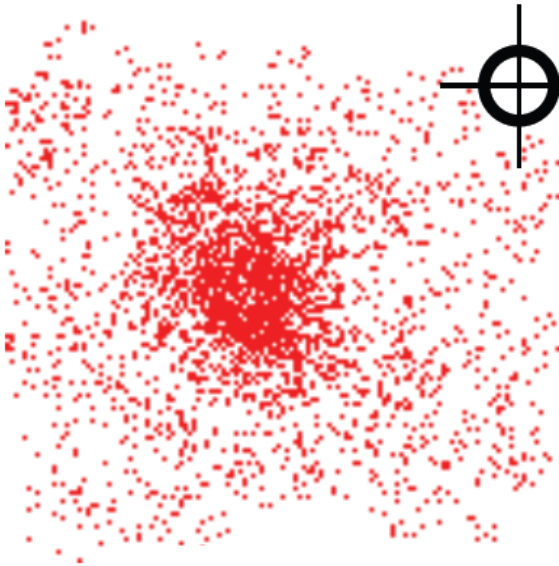
data

$$P(x) = \frac{1}{N} \sum_{n=1}^{N} K(x - x_n)$$

kernel

Nonparameteric Density Estimation

- Given N data samples
- Estimate the pdf, P(x), at some value x by
- "Counting" data samples, x_n, "around" x



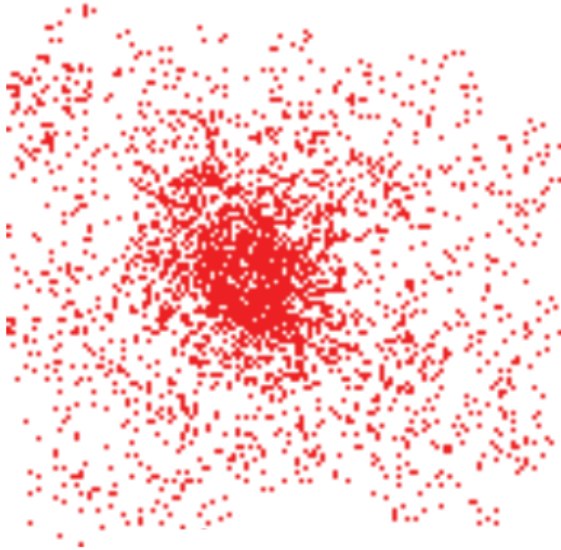
data

$$P(x) = \frac{1}{N} \sum_{n=1}^{N} K(x - x_n)$$

kernel

Nonparameteric Density Estimation

- Given N data samples
- Estimate the pdf, P(x), at some value x by
- "Counting" data samples, x_n, "around" x



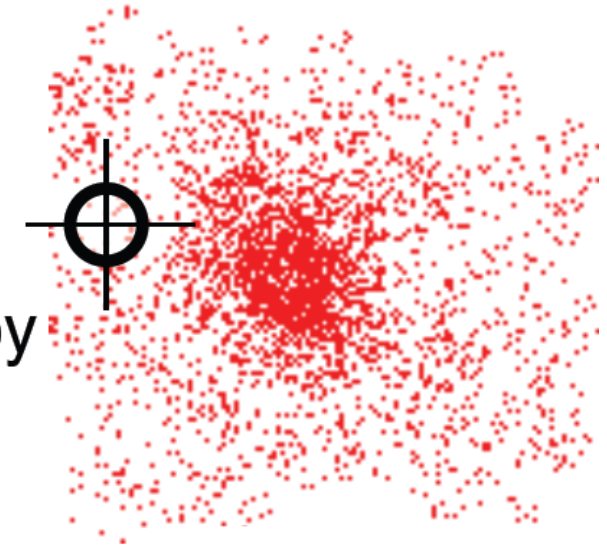
data

$$P(x) = \frac{1}{N} \sum_{n=1}^{N} K(x - x_n)$$

kernel

Nonparameteric Density Estimation

- Given N data samples
- Estimate the pdf, P(x), at some value x by
- "Counting" data samples, x_n, "around" x



data

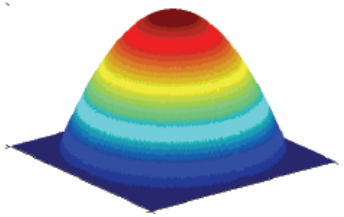
$$P(x) = \frac{1}{N} \sum_{n=1}^{N} K(x - x_n)$$

kernel

Common Kernels

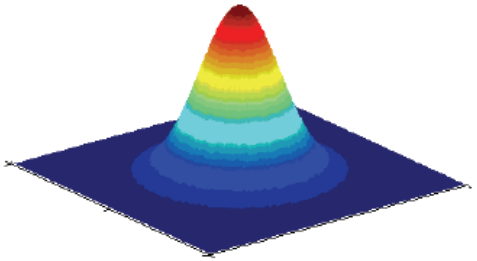
- Epanechnikov

$$K_E(x) = \begin{cases} c(1 - \|x\|^2) & , \quad \|x\| \leq 1 \\ 0 & , \quad \text{otherwise} \end{cases}$$



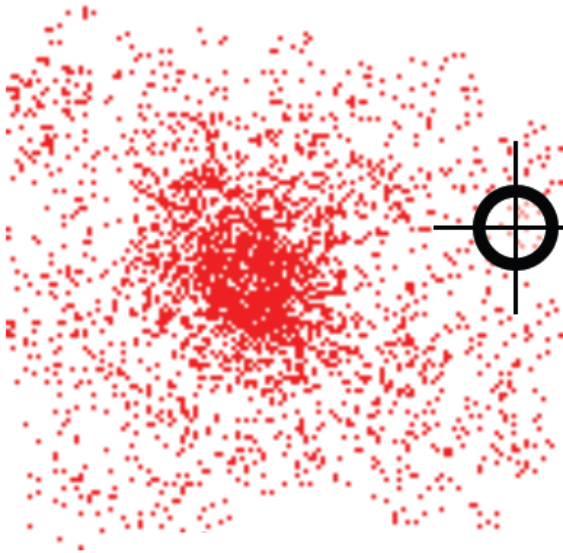
- Gaussian

$$K_G(x) = \exp\left(-\frac{1}{2}\|x\|^2\right)$$



Nonparameteric Density Estimation

- Given N data samples
- Estimate the pdf, P(x), at some value x by
- "Counting" data samples, x_n, "around" x



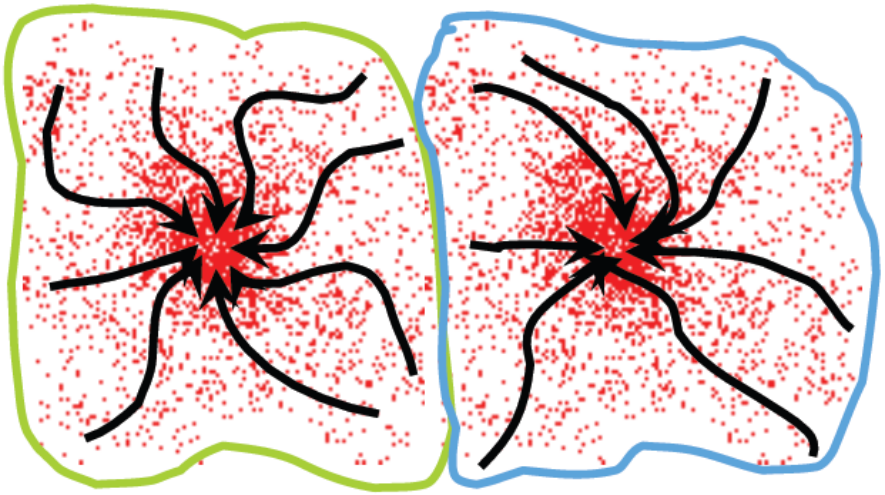
data

$$P(x) = \frac{1}{N} \sum_{n=1}^{N} K(x - x_n)$$

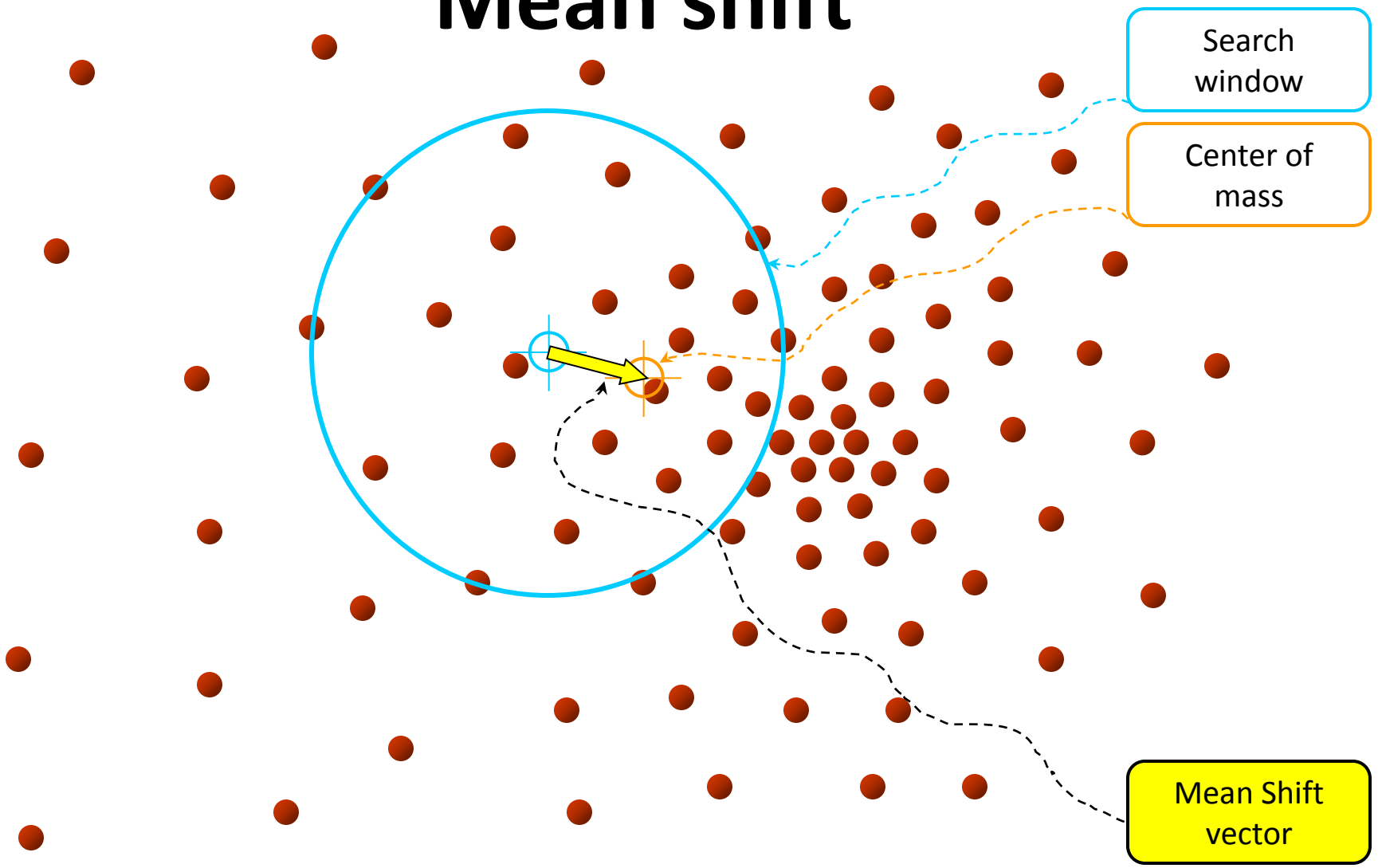
kernel

Meanshift Clustering

- Goal = Identify modes ⇒ No need to estimate the pdf
- Meanshift:
  - Estimate the gradient of the pdf
  - Cluster data that are in the attraction basin of a mode
  - All trajectories within the basin lead to the mode

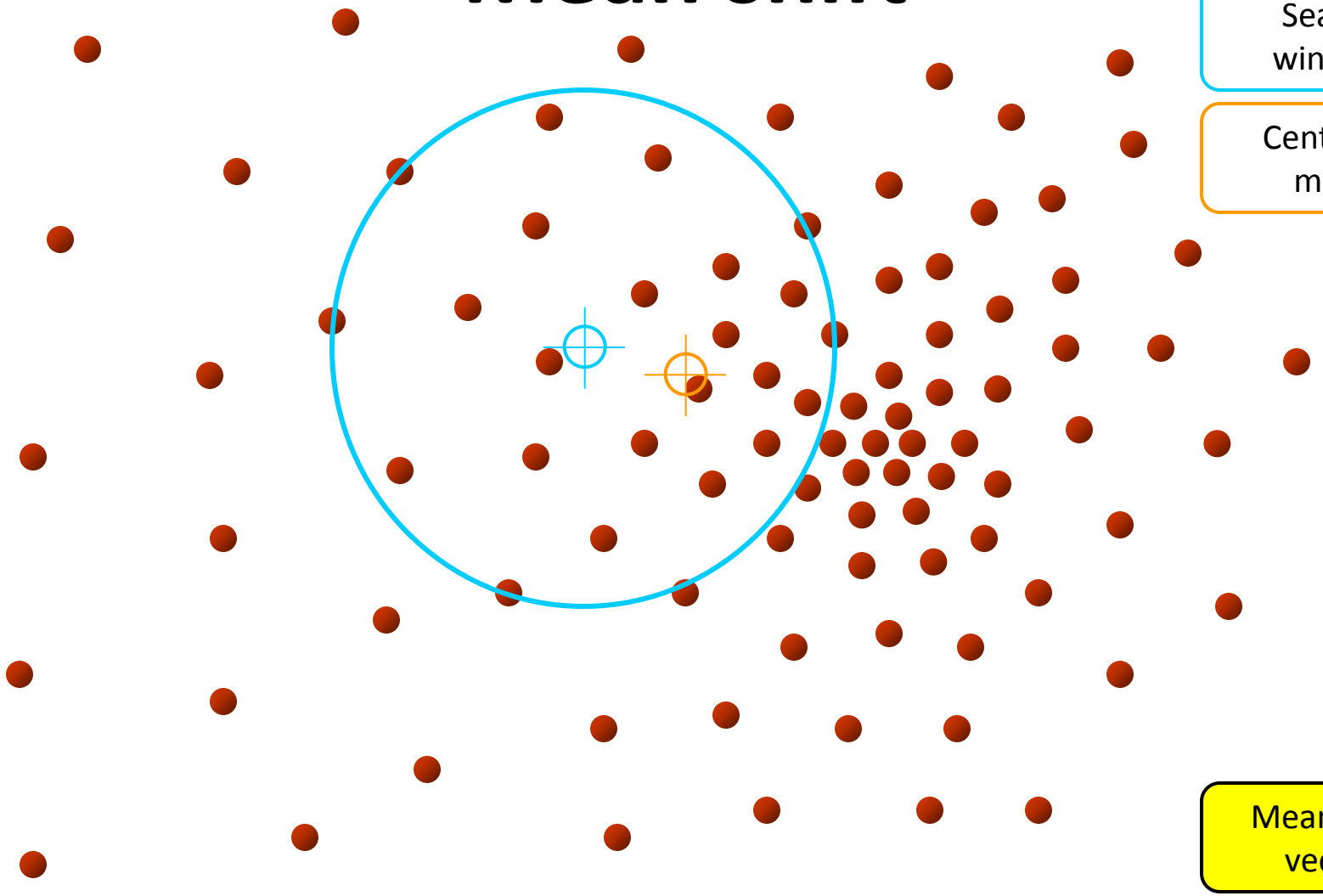


Mean shift



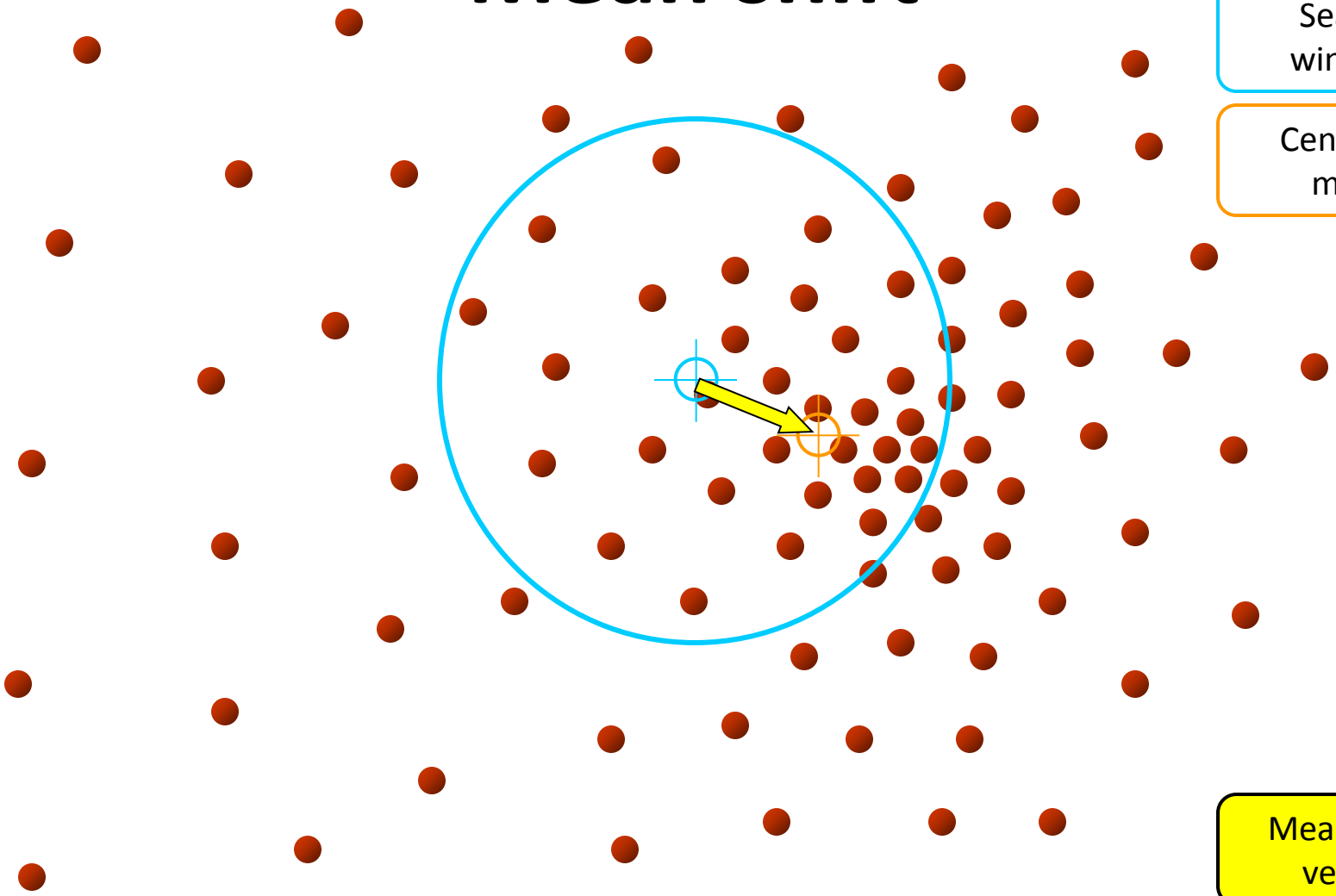
Slide by Y. Ukrainitz & B. Sarel

Mean shift



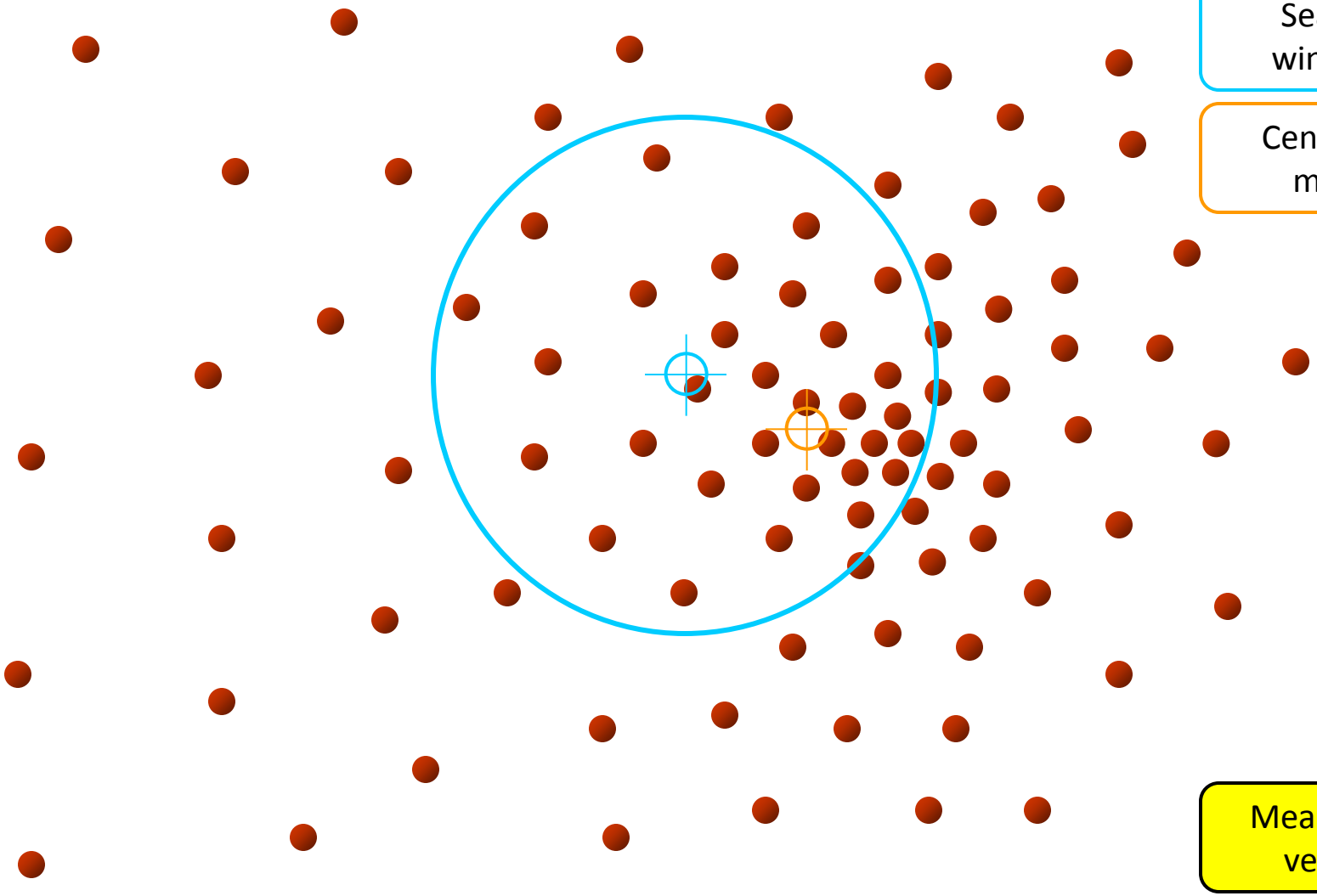
Slide by Y. Ukrainitz & B. Sarel

Mean shift



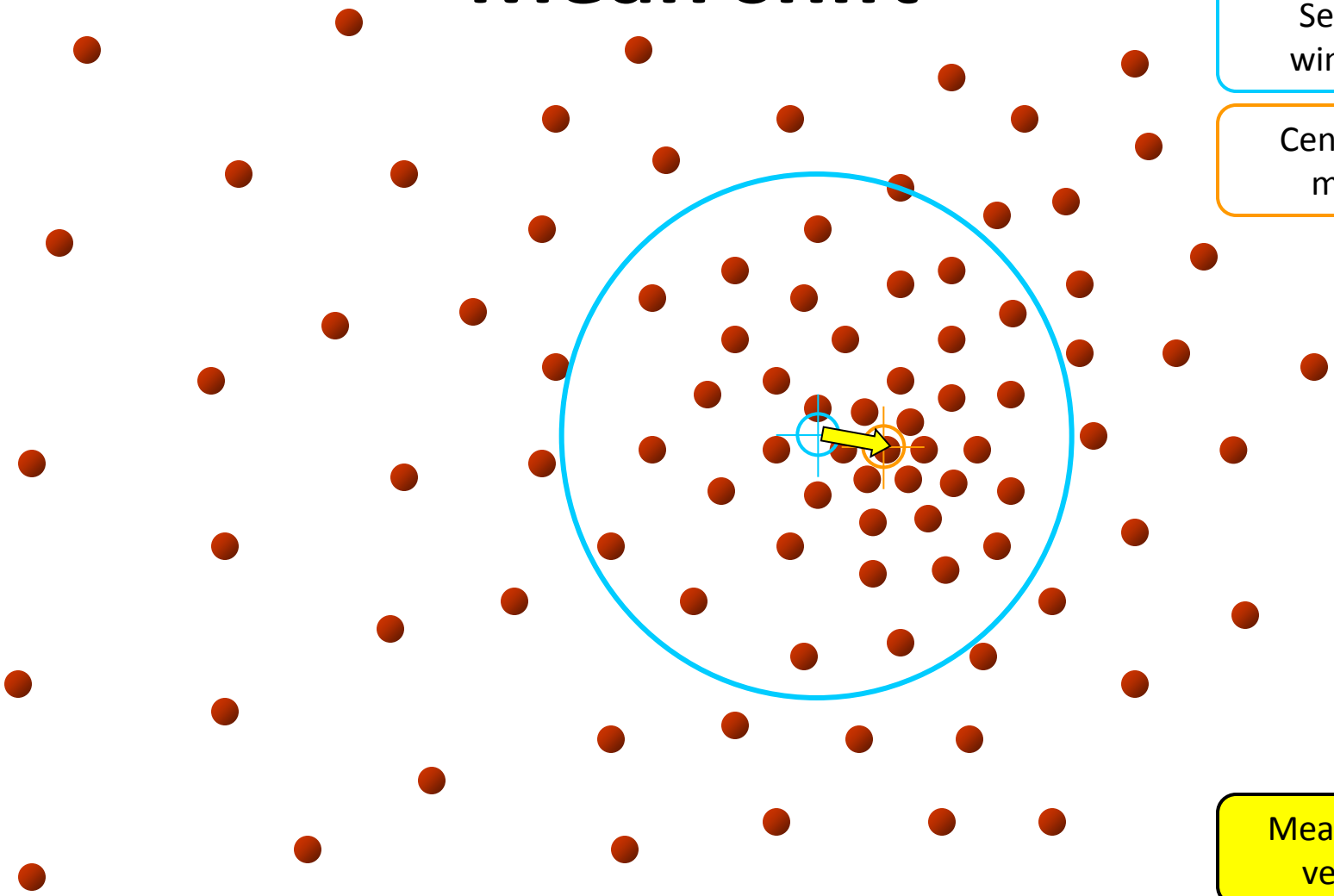
Slide by Y. Ukrainitz & B. Sarel

Mean shift



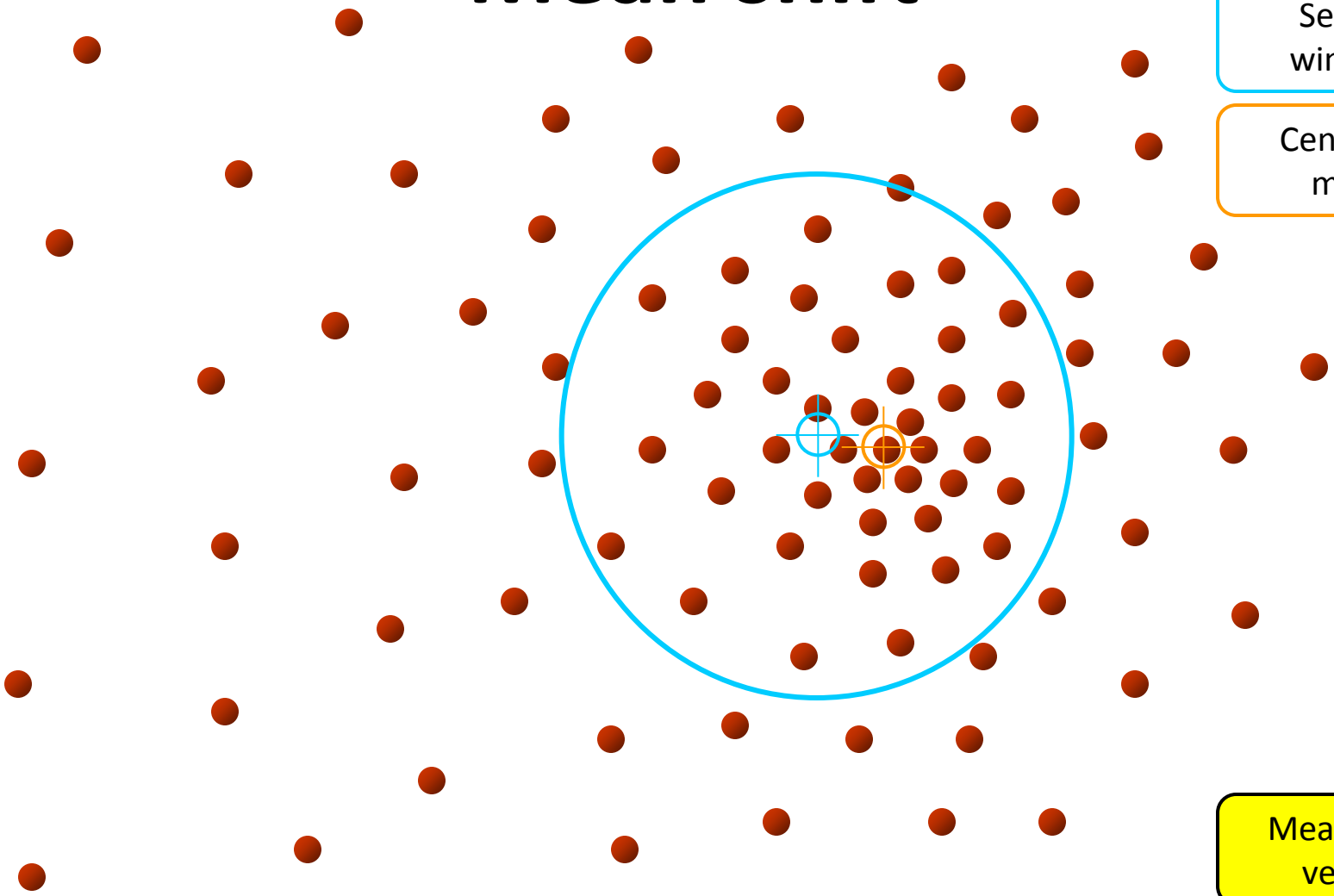
Slide by Y. Ukrainitz & B. Sarel

Mean shift



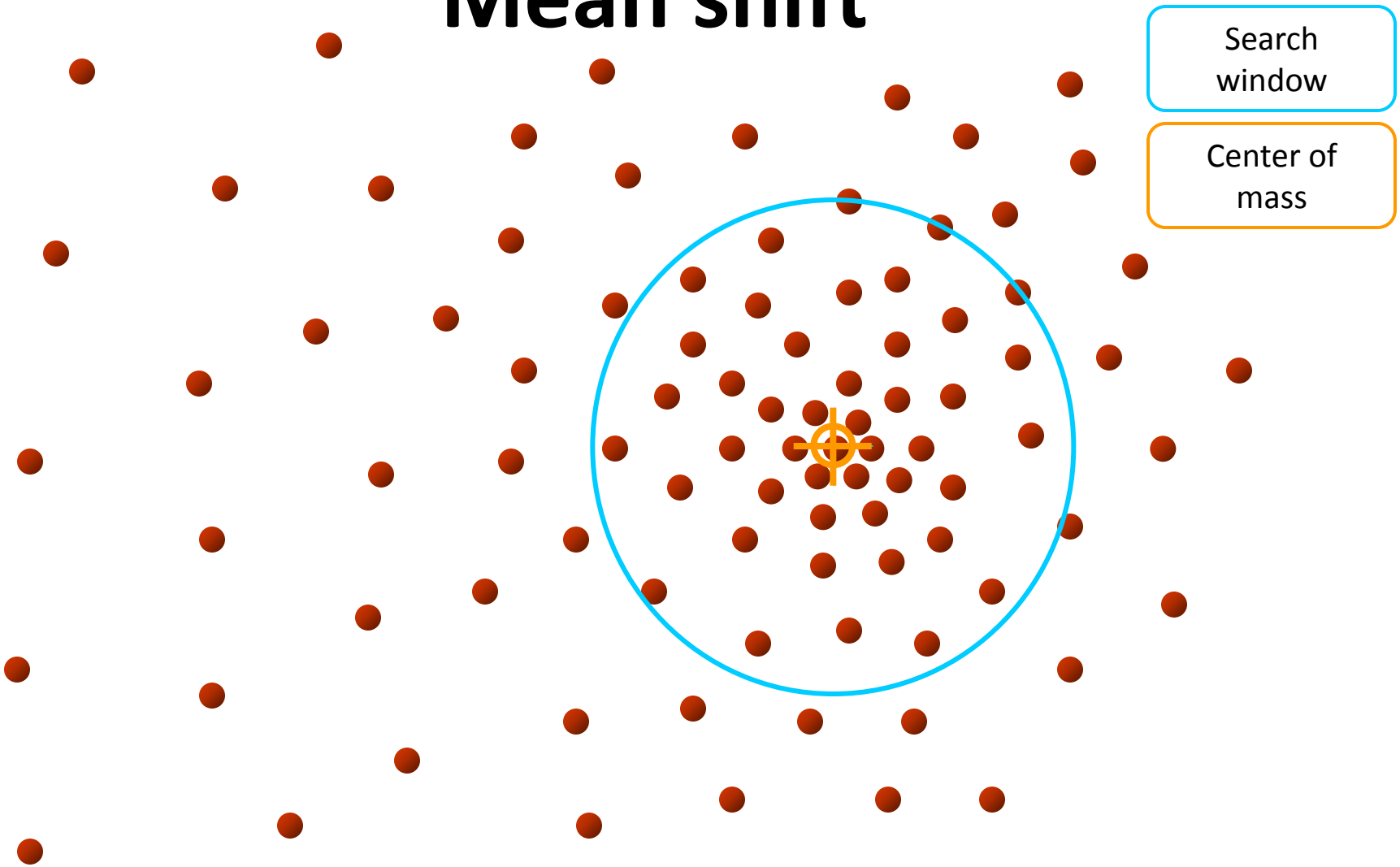
Slide by Y. Ukrainitz & B. Sarel

Mean shift



Slide by Y. Ukrainitz & B. Sarel

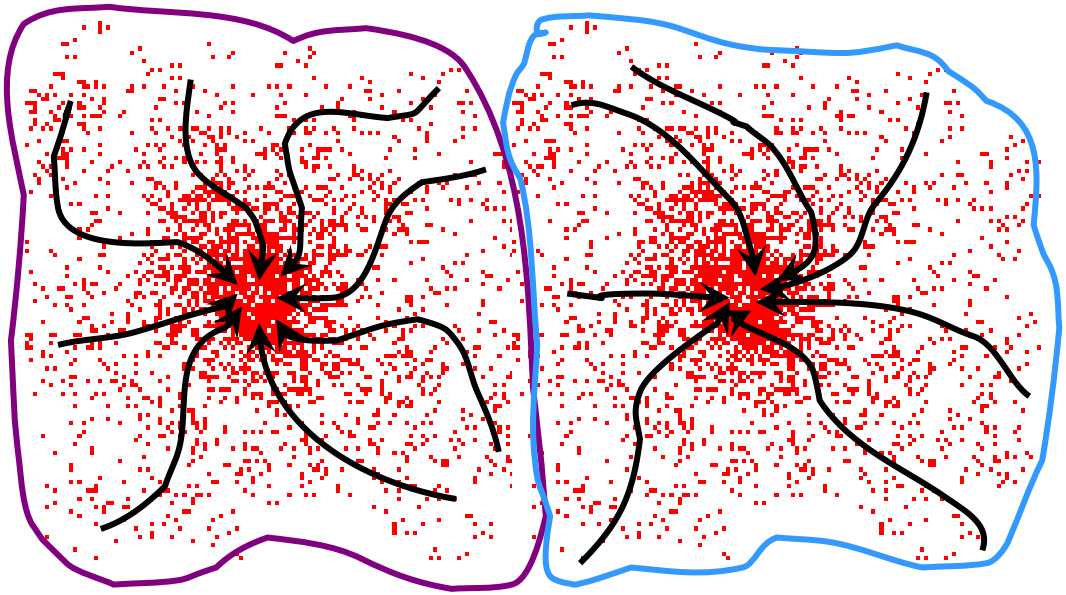
Mean shift



Slide by Y. Ukrainitz & B. Sarel

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

The Mean Shift Algorithm

- Originally intended to find modes in scattered data
- Strategy
  - start at a promising estimate of mode
  - iterate until the estimate doesn't change
    - fit a model of probability density to some points near estimate
    - find the peak of this model
- Model
  - smoothing kernel
  - the update takes a special form
    - shift the mode to a weighted mean of the nearby points
    - hence the name.

Start with an estimate of the mode $\boldsymbol{y}^{(0)}$ and a set of n data vectors $\boldsymbol{x}_i$
of dimension d, a scaling constant h, and g the derivative of the kernel profile

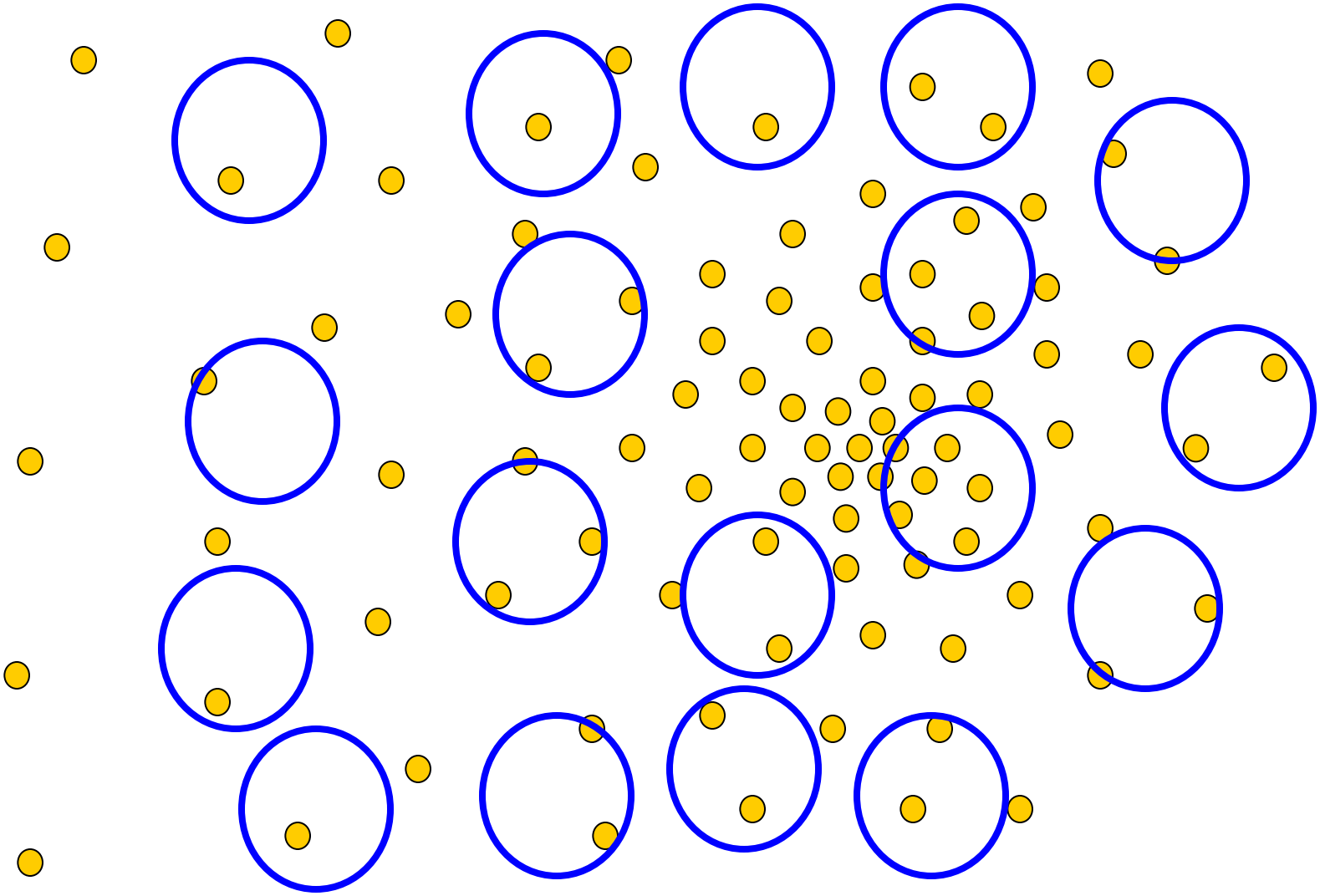
Until the update is tiny

Form the new estimate

$$\boldsymbol{y}^{(j+1)} = \frac{\sum_i \boldsymbol{x}_i g(\| \frac{\boldsymbol{x}_i - \boldsymbol{y}^{(j)}}{h} \|^2)}{\sum_i g(\| \frac{\boldsymbol{x}_i - \boldsymbol{y}^{(j)}}{h} \|^2)}$$

Algorithm 9.5: Finding a Mode with Mean Shift.

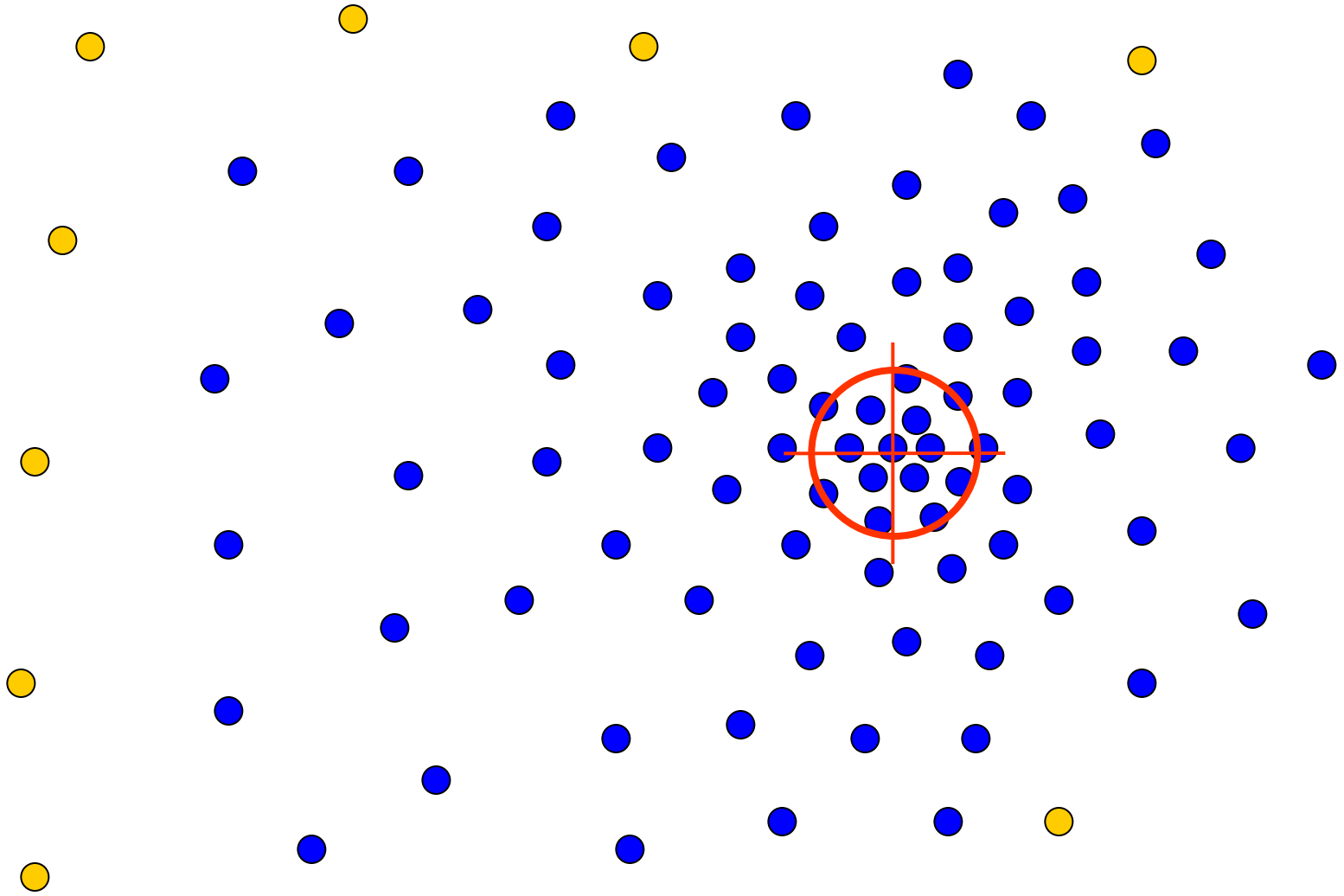
Real Modality Analysis



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

For each data point $\boldsymbol{x}_i$

Apply the mean shift procedure (Algorithm 9.5), starting with $\boldsymbol{y}^{(0)} = \boldsymbol{x}_i$

Record the resulting mode as $\boldsymbol{y}_i$

Cluster the $\boldsymbol{y}_i$, which should form small tight clusters.
A good choice is an agglomerative clusterer with group average distance,
stopping clustering when the group average distance exceeds a small threshold

The data point $\boldsymbol{x}_i$ belongs to the cluster that its mode $\boldsymbol{y}_i$ belongs to.

Algorithm 9.6: Mean Shift Clustering.

For each pixel, p_i, compute a feature vector $\boldsymbol{x}_i = (\boldsymbol{x}_i^s, \boldsymbol{x}_i^r)$ representing spatial and appearance components, respectively.

Choose h_s, h_r the spatial (resp. appearance) scale of the smoothing kernel.

Cluster the $\boldsymbol{x}_i$ using this data and mean shift clustering (Algorithm 9.6).

(Optional) Merge clusters with fewer than t_{min} pixels with a neighbor; the choice of neighbor is not significant, because the cluster is tiny.

The i'th pixel belongs to the segment corresponding to its cluster center (for example, one could label the cluster centers $1 \ldots r$, and then identify segments by computing a map of the labels corresponding to pixels).

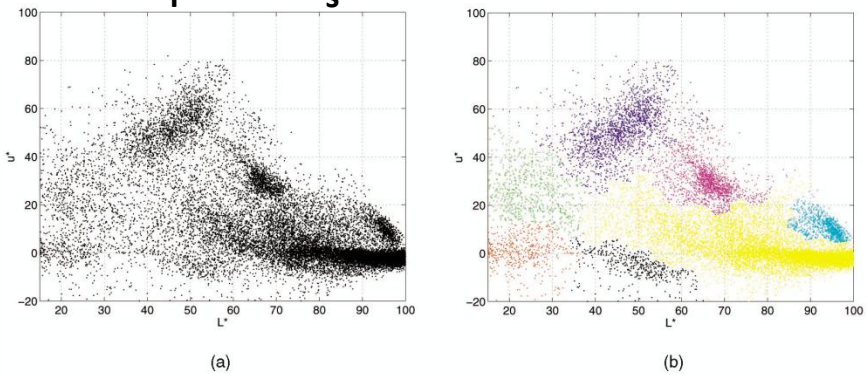
Algorithm 9.7: Mean Shift Segmentation.

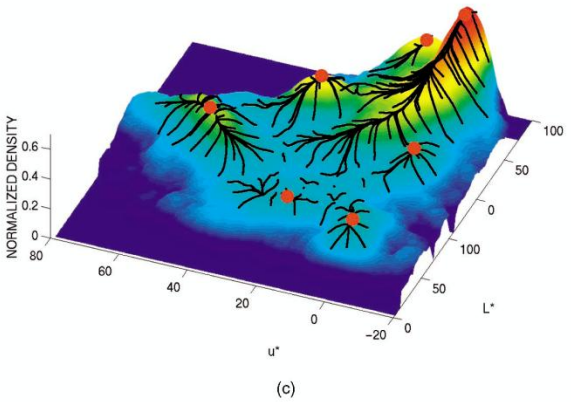
Mean shift clustering

- **The mean shift algorithm seeks *modes* of the given set of points**
  1. **Choose kernel and bandwidth**
  2. **For each point:**
     a) **Center a window on that point**
     b) **Compute the mean of the data in the search window**
     c) **Center the search window at the new mean location**
     d) **Repeat (b,c) until convergence**
  3. **Assign points that lead to nearby modes to the same cluster**

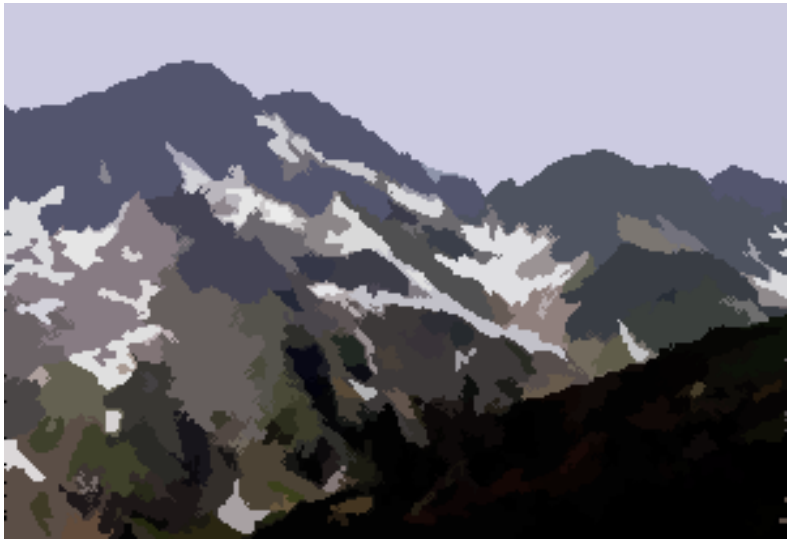
Segmentation by Mean Shift

- Compute features for each pixel (color, gradients, texture, etc); also store each pixel's position
- Set kernel size for features K_f and position K_s
- Initialize windows at individual pixel locations
- Perform mean shift for each window until convergence
- Merge modes that are within width of K_f and K_s





Mean shift segmentation results



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

Mean shift segmentation results

Mean shift pros and cons

- **Pros**
  - **Good general-purpose segmentation**
  - **Flexible in number and shape of regions**
  - **Robust to outliers**
- **Cons**
  - **Have to choose kernel size in advance**
  - **Not suitable for high-dimensional features**
- **When to use it**
  - **Oversegmentation**
  - **Multiple segmentations**
  - **Tracking, clustering, filtering applications**
    - **D. Comaniciu, V. Ramesh, P. Meer: *Real-Time Tracking of Non-Rigid Objects using Mean Shift*, *Best Paper Award*, IEEE Conf. Computer Vision and Pattern Recognition (CVPR'00), Hilton Head Island, South Carolina, Vol. 2, 142-149, 2000**

Outline for Segmentation

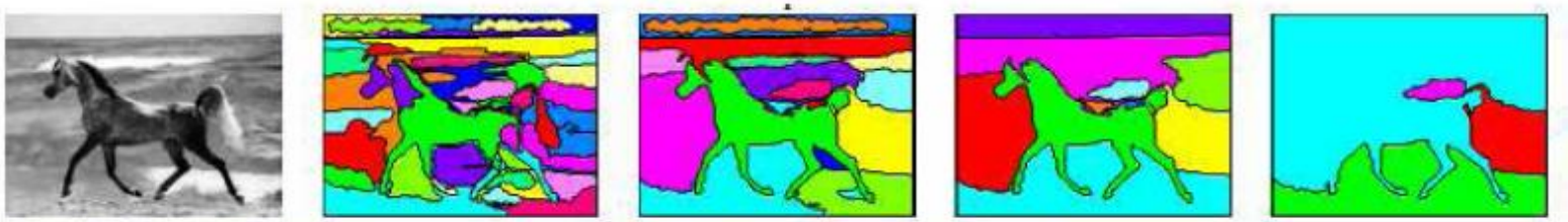
- Introduction
  - Goal of Segmentation
- Segmentation and grouping
- Image Segmentation by Clustering Pixels
  - Segmentation as clustering
  - K-means clustering
  - Mean shift
- Spectral Clustering(Wei Fangyun)
- Integrating top-down and bottom-up segmentation for recognition

Outline for Today

- Introduction
  - Goal of Segmentation
- Segmentation and grouping
- Bottom-up segmentation
  - Segmentation as clustering
    - K-means clustering
  - Model-free clustering
    - Mean shift
- Spectral Clustering(Wei Fangyun)
- Integrating top-down and bottom-up segmentation for recognition

Bottom-up segmentation

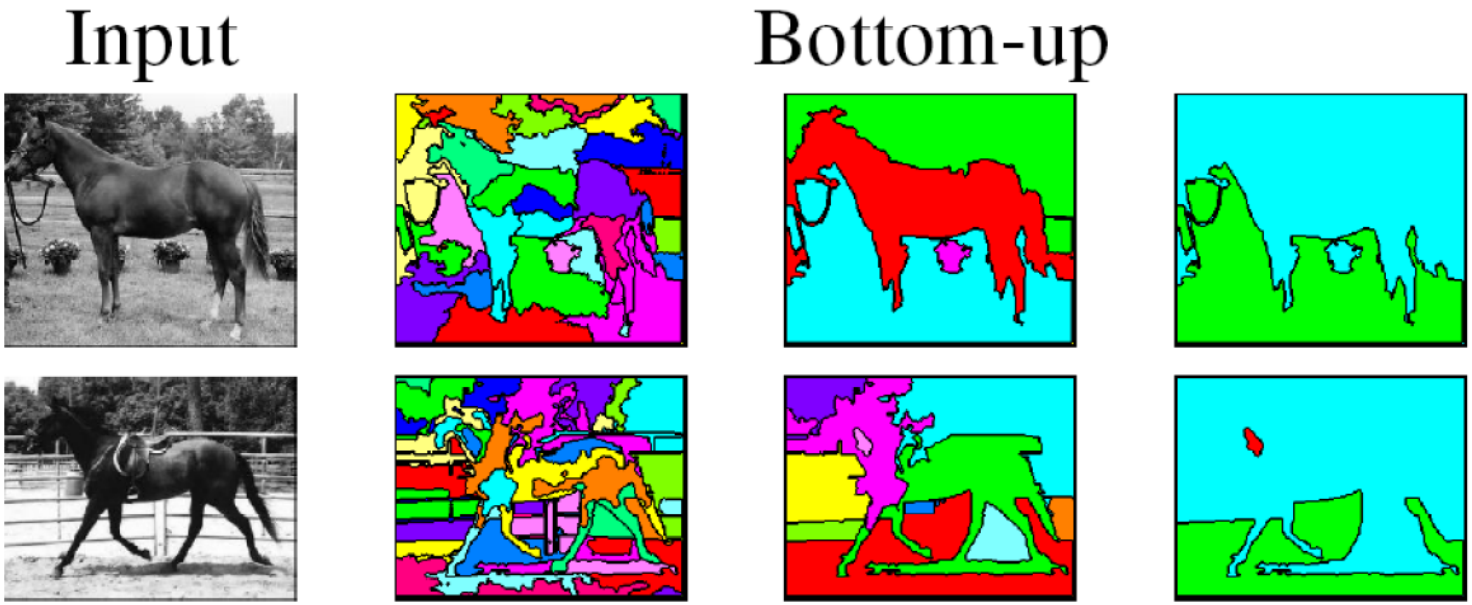
- **Find homogeneous regions in image**
- **Fine to coarse scale**
- **Can be done iteratively by merging similar neighboring regions**



Source: Borenstein et al.: Combining Top-down and Bottom-up Segmentation

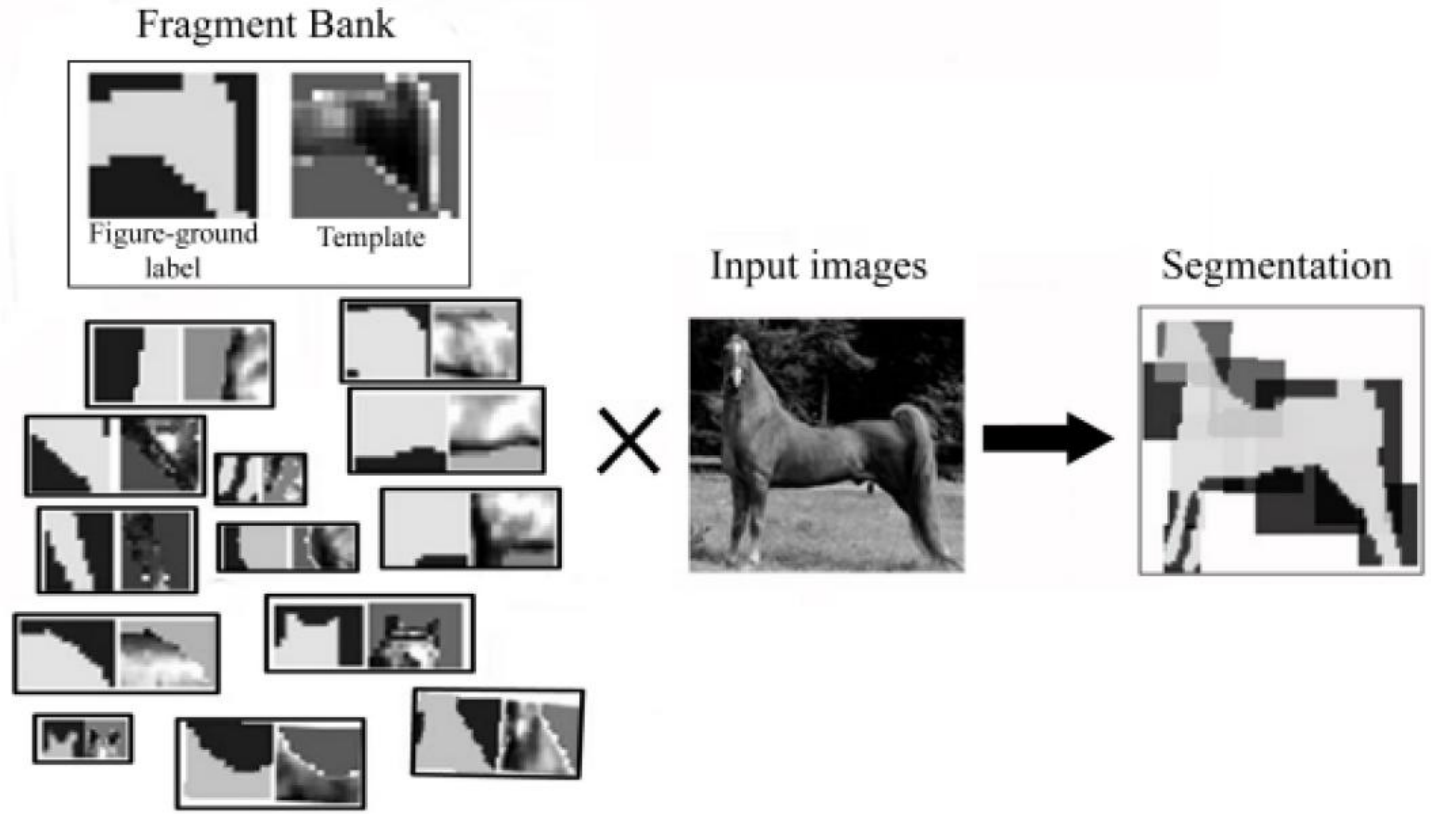
Examples of bottom-up segmentation

- Using Normalized Cuts, Shi & Malik, 1997

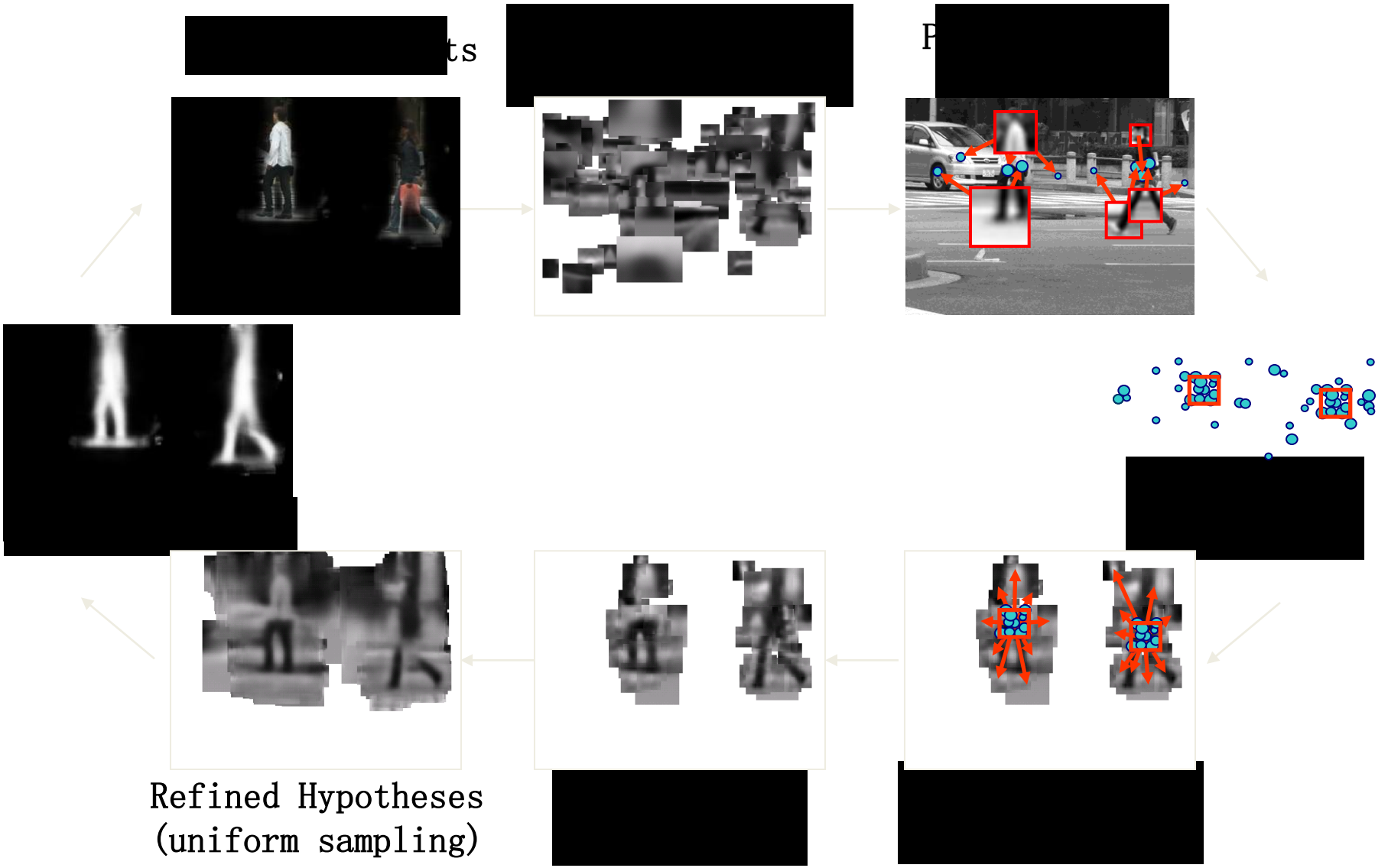


Borenstein and Ullman, ECCV 2002

Jigsaw approach: Borenstein and Ullman, 2002



Implicit Shape Model - Liebe and Schiele, 2003



Liebe and Schiele, 2003, 2005

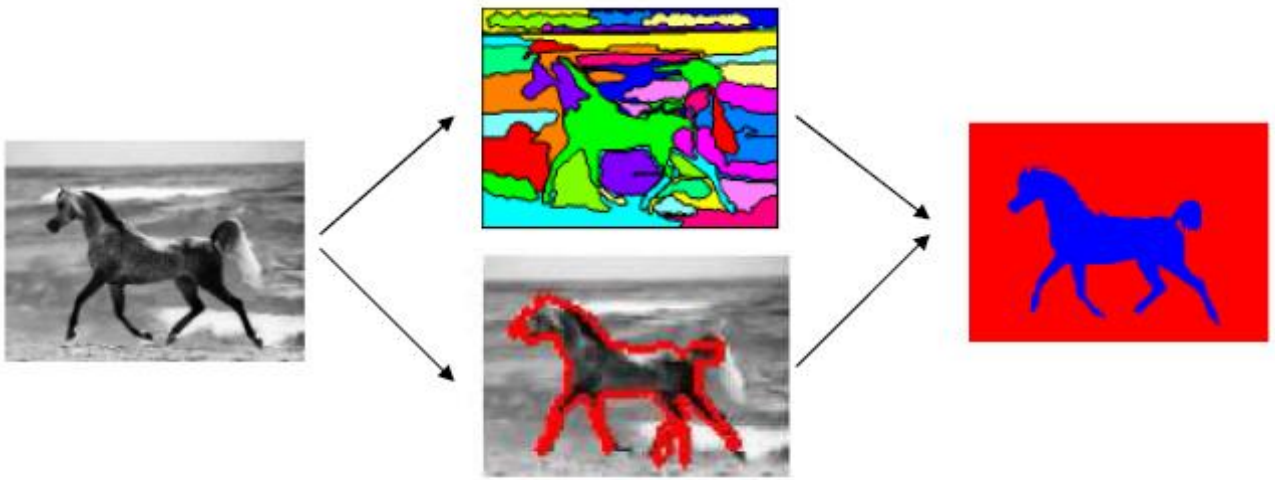
Top-down segmentation

- **Based on knowledge about the object we want to segment**
- **Build a model for the object category**
- **A training and segmentation phase**
- **Models: ASM, AAM, Snakes etc.**

Source: Borenstein et al.: Combining Top-down and Bottom-up Segmentation

Combining top-down and bottom-up

- **Use the information in top-down model to combine segments from the bottom-up segmentation**



Results

Using LPS Model for Cow

Layered Pictorial Structures (LPS)

Image

Segmentation

Slide credit: P. Kumar

Results

Using LPS Model for Cow

In the absence of a clear boundary between object and background

Image

Segmentation

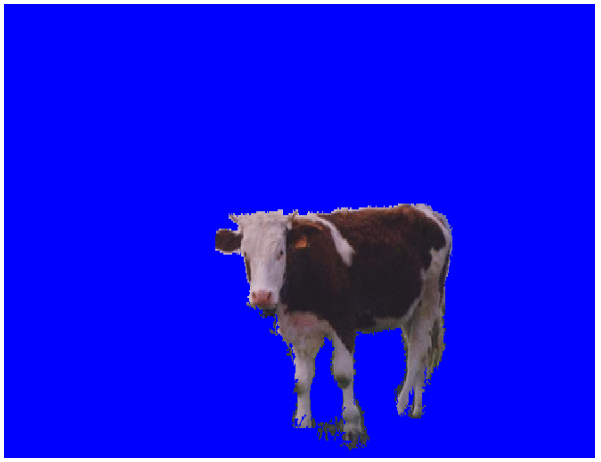
Slide credit: P. Kumar

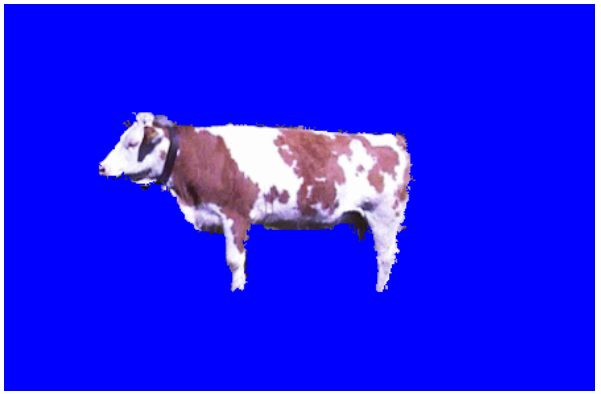
Results

Using LPS Model for Cow

Image

Segmentation





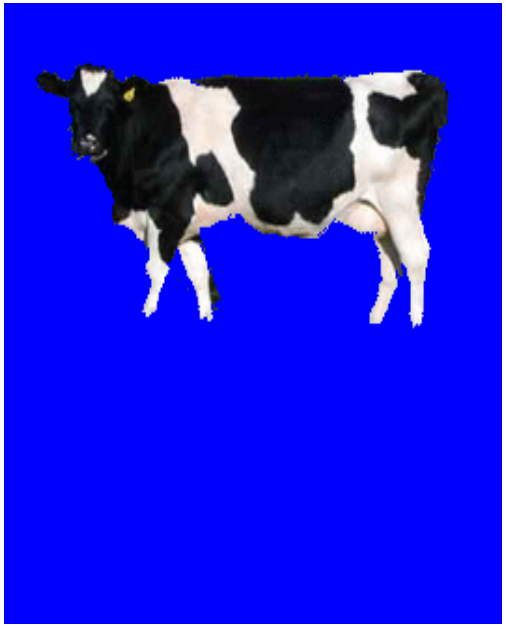
Slide credit: P. Kumar

Results

Using LPS Model for Cow

Image

Segmentation



Results

Using LPS Model for Horse

Image

Segmentation

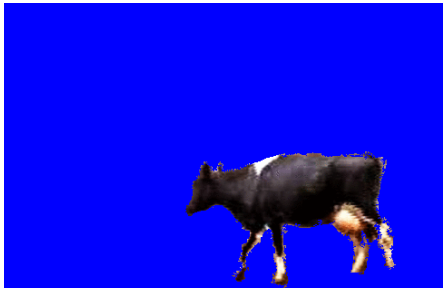
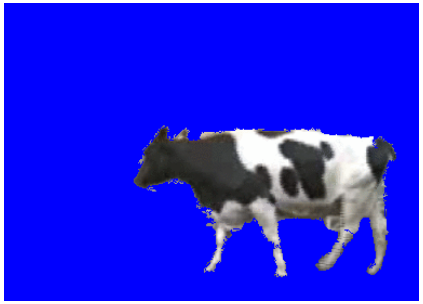
Results

Using LPS Model for Horse

Image

Segmentation

Results

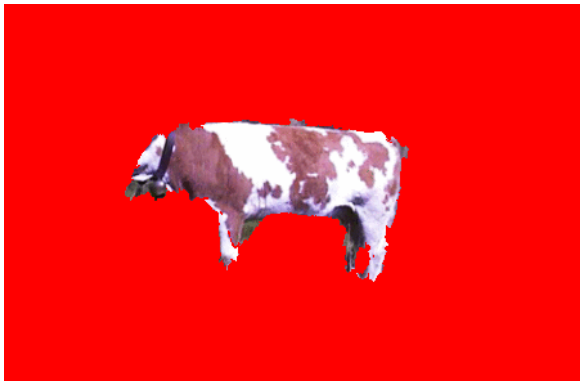
Image	The Method	Leibe and Schiele
		
		
		

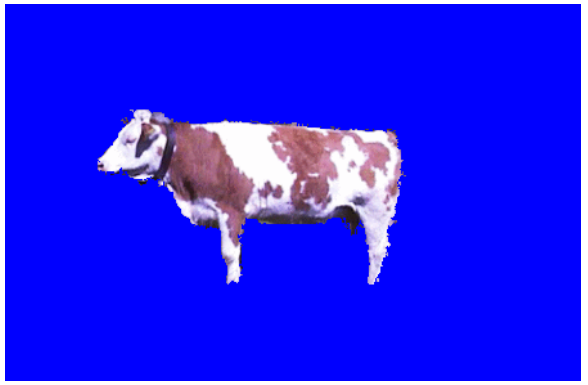
Results

Shape

Appearance

Shape+Appearance





Without $\phi_x(\mathbf{D}|m_x)$

Without $\phi_x(m_x|\theta)$

Quality of segmentation

- How similar are points in a segment
  - Maximize the difference between segments
  - Minimize the difference between the points in a segment
- Ability to find interest regions
- Fundament for further interpretation
  - Base segmentation on the right features (pixel intensities, texture etc.)
  - Choose the best suited method for the purpose at hand

Quality of segmentation

- **Segmentation is a mid-level step –**
  later steps include:
  - **Noise reduction – e.g. minimum region size, erosion/dilation, median filter etc.**
  - **Merging of regions – based on region features**
  - **Higher level interpretation – classification to foreground/background or object categories etc.**

Evaluating Segmenters

- **Collect "correct" segmentations**
  - **from human labellers**
  - **these may not be perfect, but ...**

- **Now apply your segmenter**
  - **Count**
    - **% human boundary pixels close to your boundary pixels -- Recall**
    - **% of your boundary pixels close to human boundary pixels -- Precision**

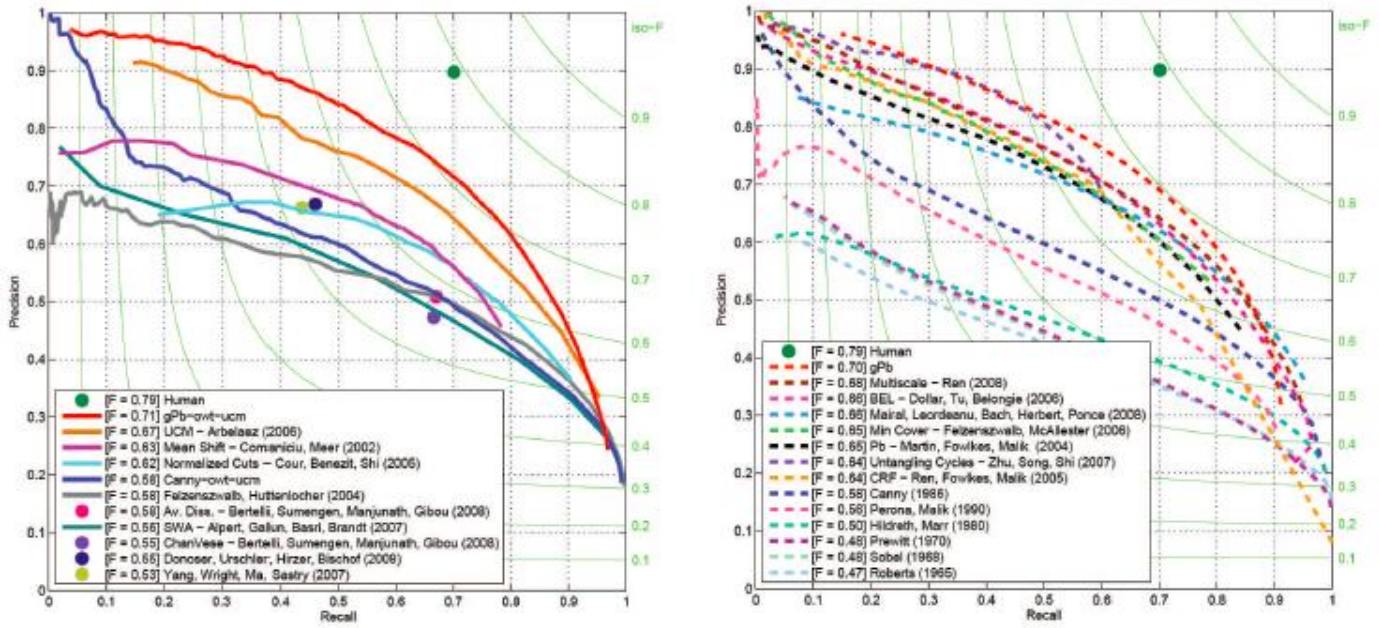


FIGURE 9.25: Segmenters and edge detectors can be evaluated by comparing the predicted boundaries to object boundaries that people mark on images. A natural comparison involves precision (the percentage of the marked boundary points that are real ones) and recall (the percentage of real boundary points that were marked); the F measure summarizes precision and recall into a single number $F = 2PR/(P + R)$. On the **left**, these measures for various segmenters; on the **right**, for various edge detectors. *This figure was originally published as Figures 1 and 2 of "Contour Detection and Hierarchical Image Segmentation" by P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, IEEE Transactions on Pattern Analysis and Machine Intelligence, 2011,* © *IEEE, 2011.*

Segmentation codes

Code is now available for many important image segmenters. The EDISON codes (from Rutgers' computer vision group, available at `http://coewww.rutgers.edu/riul/research/robust.html`) implement mean shift image segmentation (Section 9.3.5). The same web page distributes a variety of other mean shift codes. Pedro Felzenszwalb distributes code for his segmenter (Section 9.4.2) at `http://people.cs.uchicago.edu/~pff/segment/`. Jianbo Shi distributes code for normalized cuts at `http://www.cis.upenn.edu/~jshi/software/`. Greg Mori distributes code for computing superpixels using normalized-cut algorithms at `http://www.cs.sfu.ca/~mori/research/superpixels/`. Yuri Boykov distributes code for min-cut problems at `http://vision.csd.uwo.ca/code/`; this includes codes for extremely large grids. Vladimir Kolmogorov distributes a min-cut code at `http://www.cs.ucl.ac.uk/staff/V.Kolmogorov/software.html`.

Summary

- **Segmentation to find object boundaries or mid-level regions, tokens.**
- **Bottom-up segmentation via clustering**
  - **General choices -- features, affinity functions, and clustering algorithms**
- **Grouping also useful for quantization, can create new feature summaries**
  - **Texton histograms for texture within local region**
- **Example clustering methods**
  - **K-means**
  - **Mean shift**
  - **Graph cut, normalized cuts**

Literature suggestions

- **Chapter 9.1, 9.3 and 9.4**