Image segmentation

Lihi Zelnik-Manor, Computer Vision

Today's class

- Segmentation and grouping
  - Gestalt cues
  - By clustering (mean-shift)
  - By boundaries (watershed)

- Superpixels and multiple segmentations

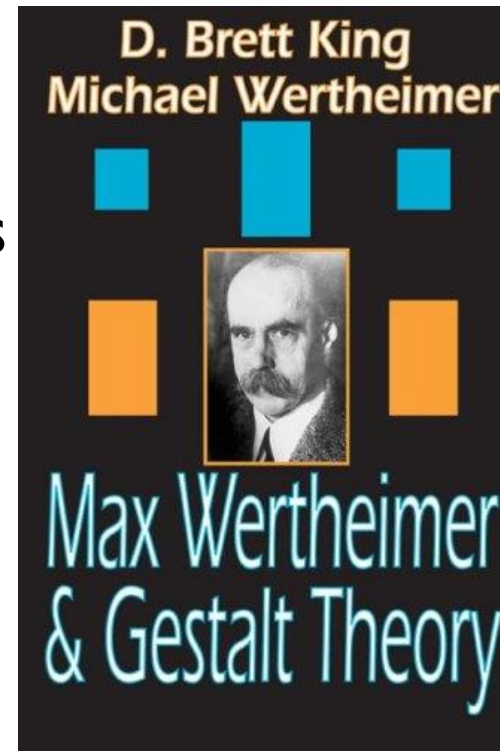
over segmentation and super pixel are the same

Gestalt psychology or **gestaltism**

- German: *Gestalt* - "form" or "whole”
  - Berlin School, early 20th century
  - Kurt Koffka, Max Wertheimer, and Wolfgang Köhler

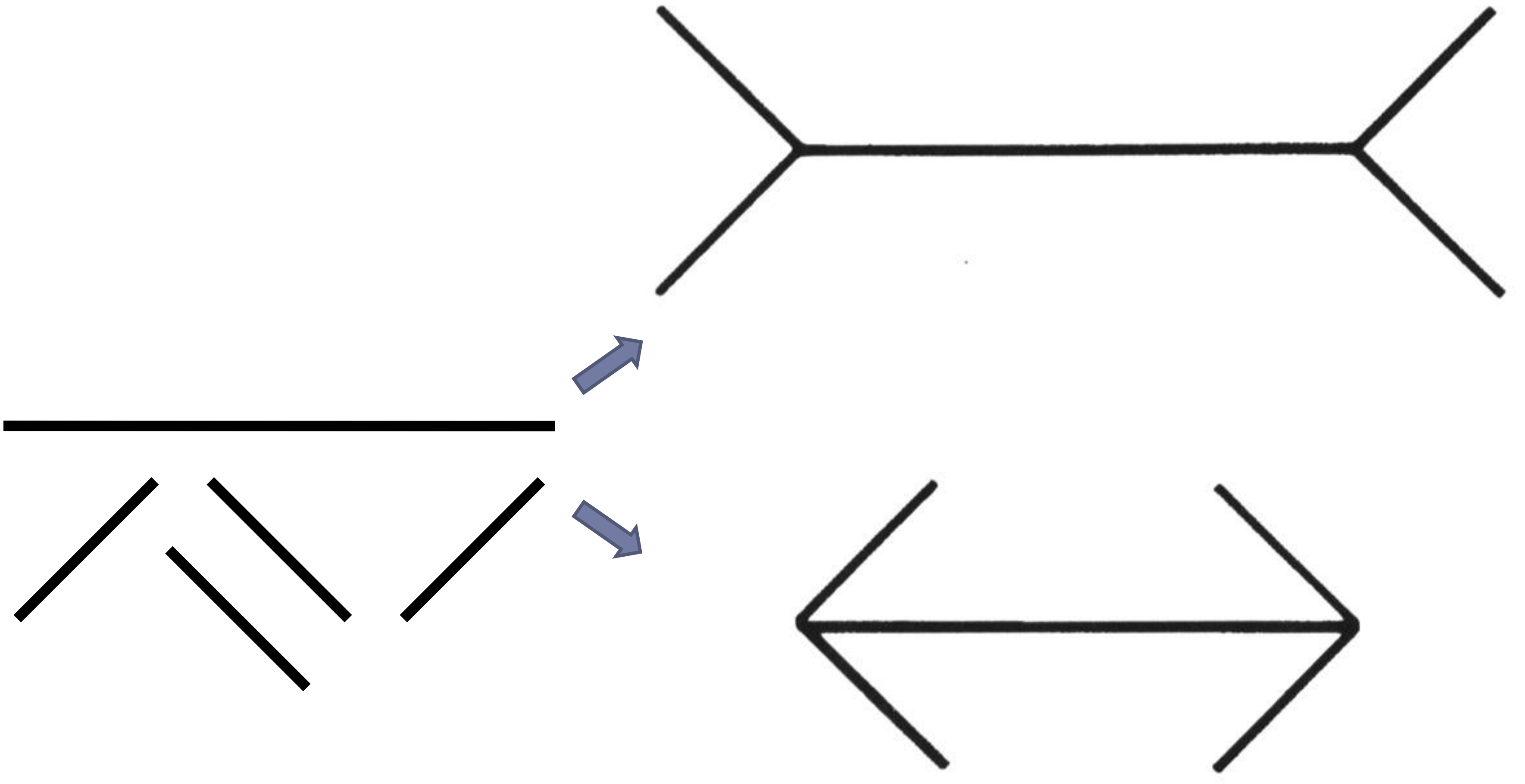
- View of brain:

- whole is more than the sum of its parts
- holistic
- parallel
- analog
- self-organizing tendencies



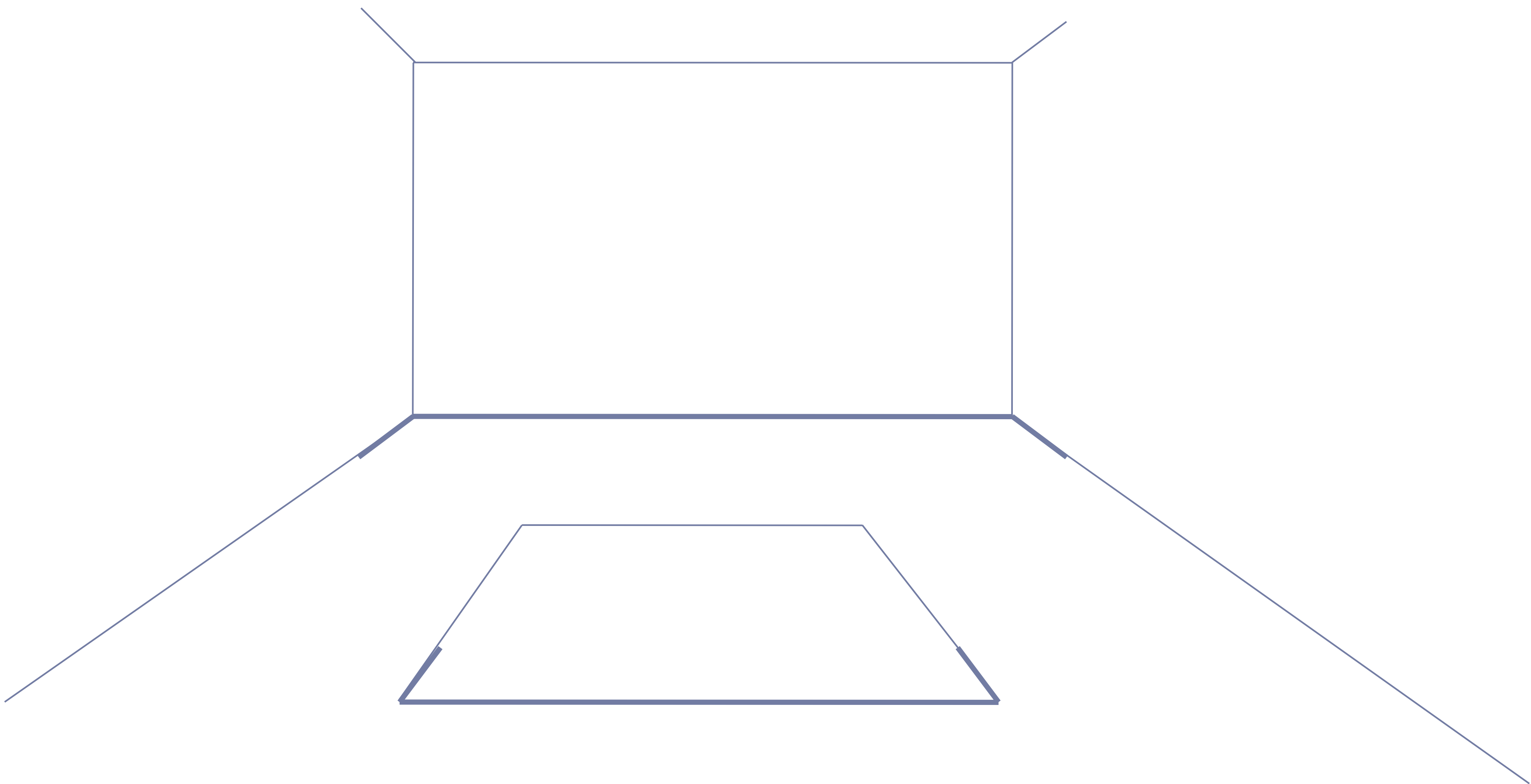
Slide from S. Saverese

Gestaltism

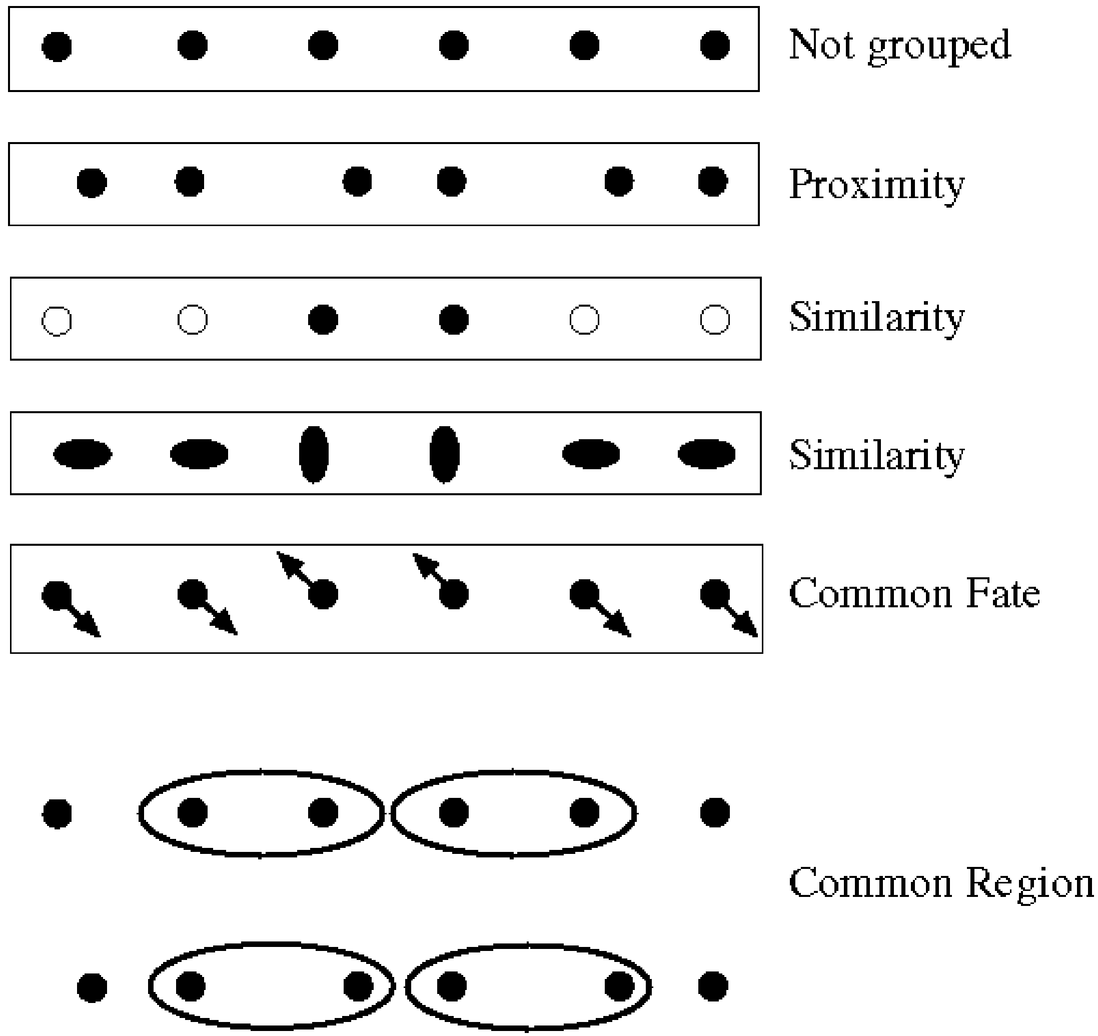


The Muller-Lyer illusion

We perceive the interpretation, not the senses

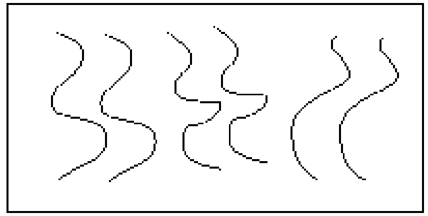


Principles of perceptual organization

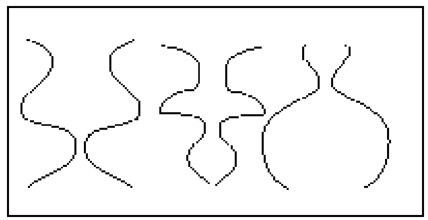


From Steve Lehar: The Constructive Aspect of Visual Perception

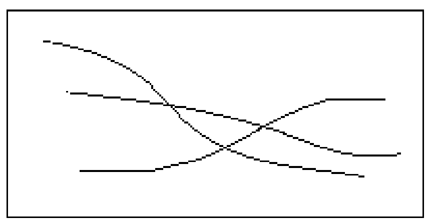
Principles of perceptual organization



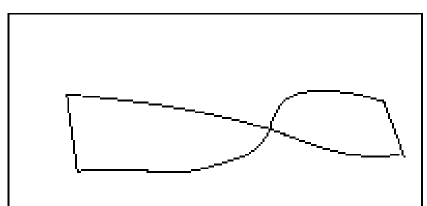
Parallelism



Symmetry

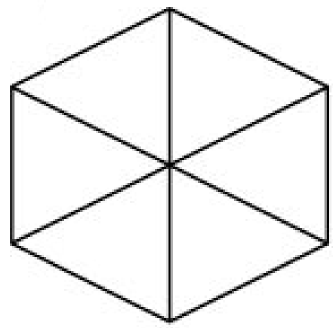


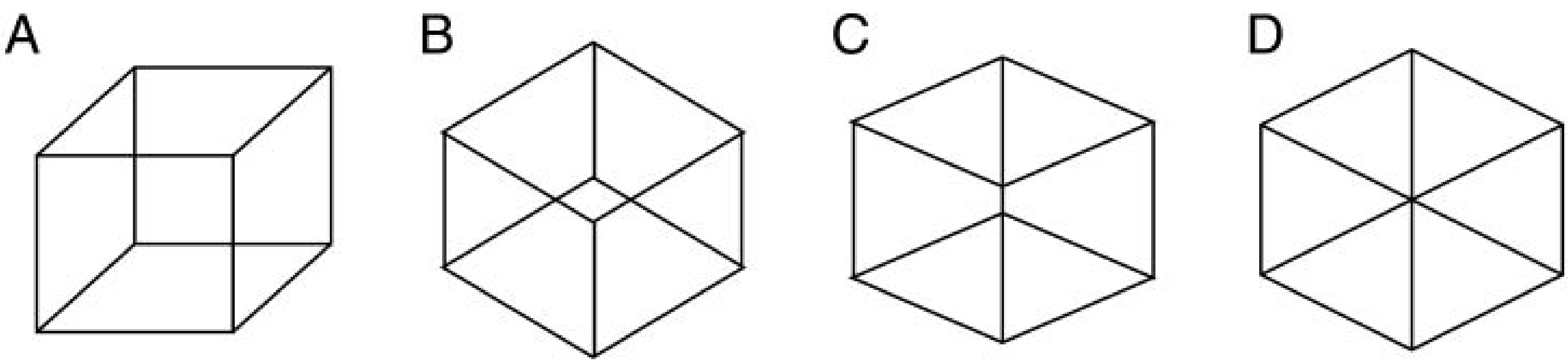
Continuity



Closure

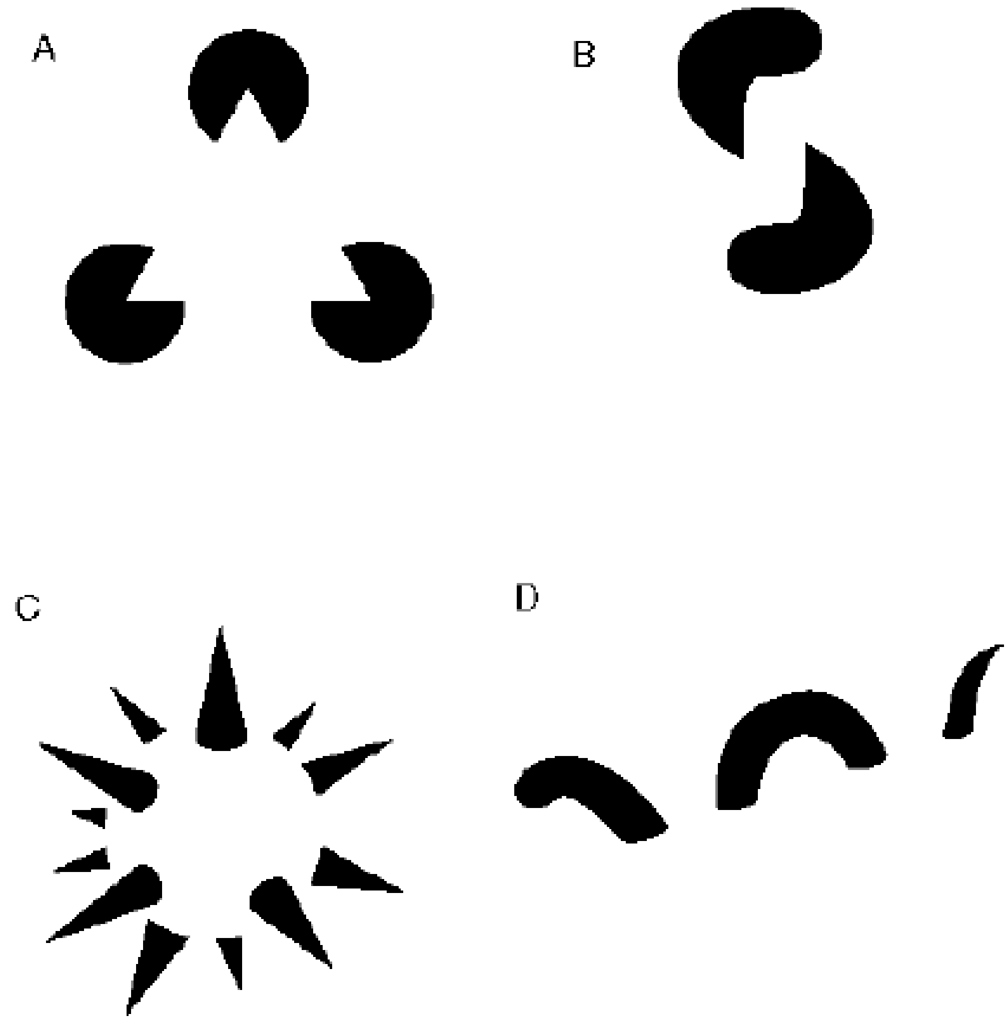
Gestaltists do not believe in coincidence





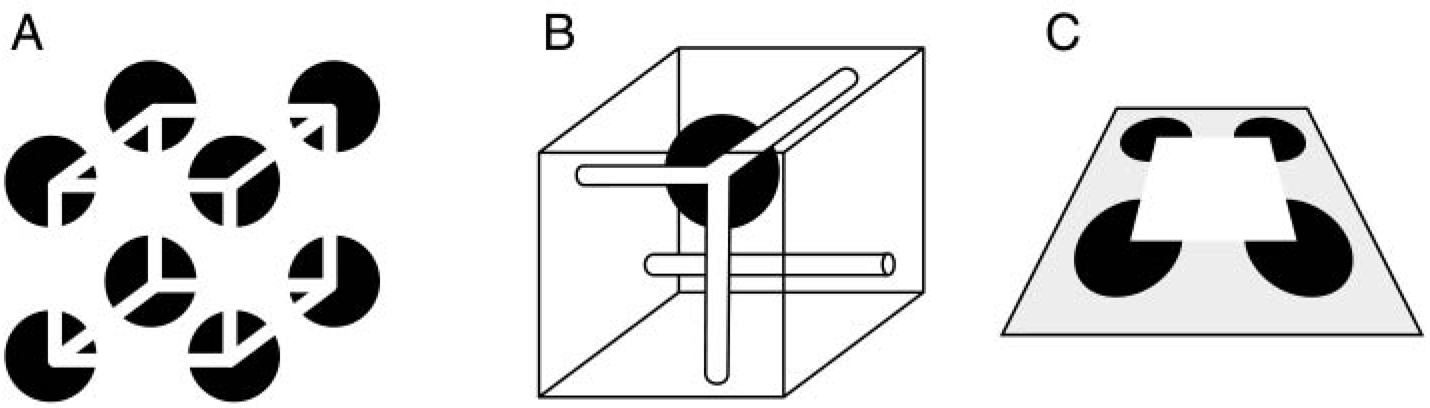
Emergence

Grouping by invisible completion



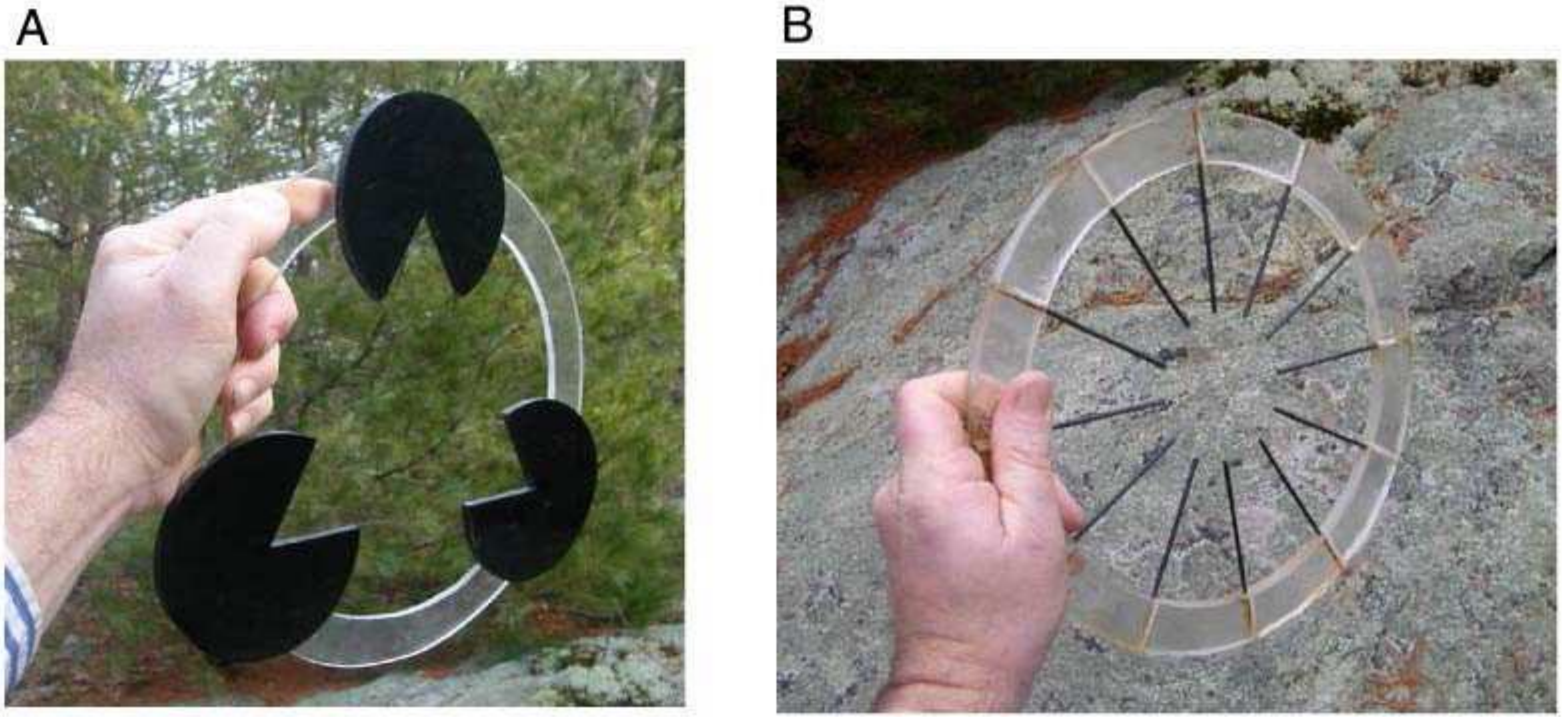
From Steve Lehar: The Constructive Aspect of Visual Perception

Grouping involves global interpretation



From Steve Lehar: The Constructive Aspect of Visual Perception

Grouping involves global interpretation



From Steve Lehar: The Constructive Aspect of Visual Perception

Gestalt cues

- Good intuition and basic principles for grouping
- Basis for many ideas in segmentation and occlusion reasoning
- Some (e.g., symmetry) are difficult to implement in practice

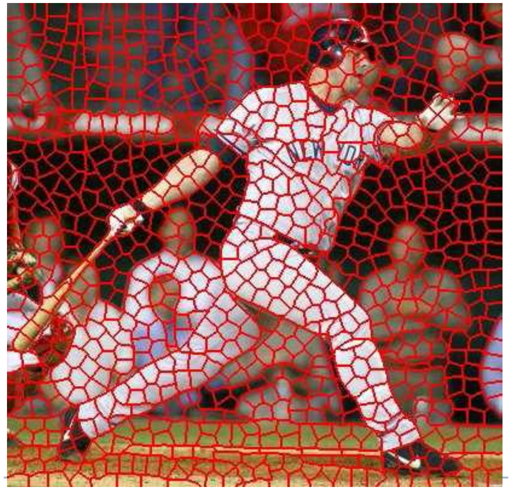
Image segmentation

Goal: Group pixels into meaningful or perceptually similar regions

Segmentation for efficiency: “superpixels”

[Felzenszwalb and Huttenlocher 2004]

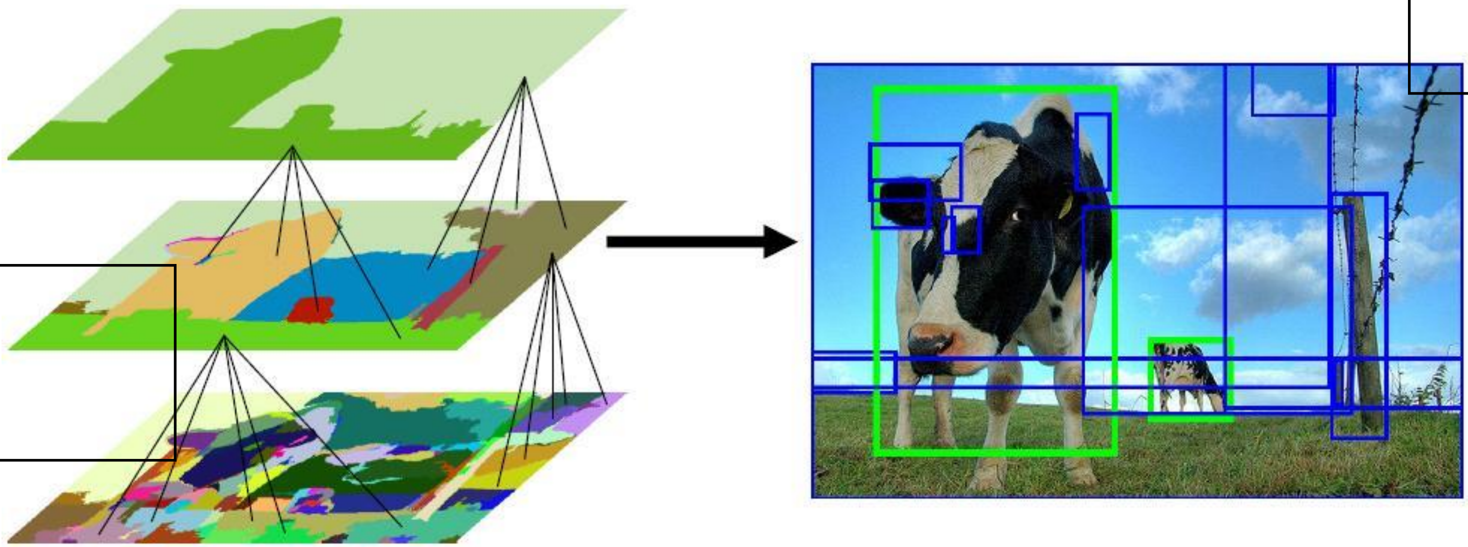
Bottom up



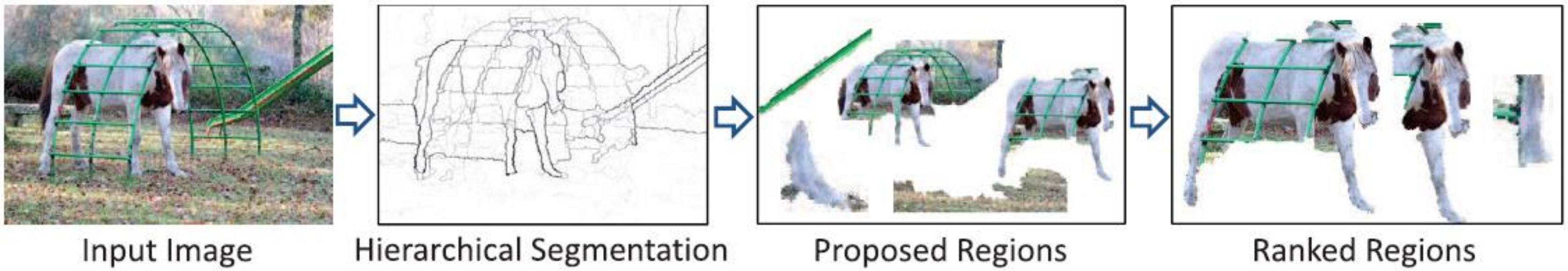
[Hoiem et al. 2005, Mori 2005]

Segmentation for feature support

Segmentation for object proposals



"Selective Search" [Sande, Uijlings et al. ICCV 2011, IJCV 2013]



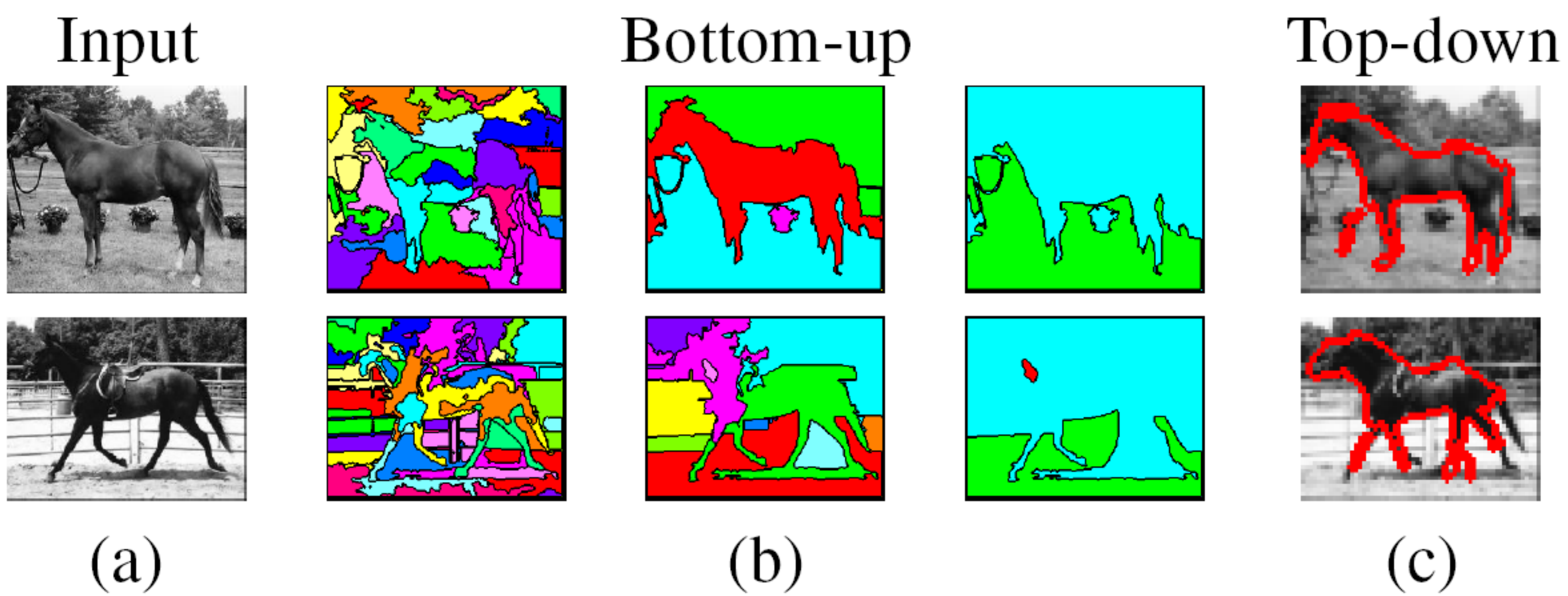
[Endres & Hoiem ECCV 2010, IJCV 2014]

Segmentation for image editing

Rother et al. 2004

Major processes for segmentation

- Bottom-up: group tokens with similar features
- Top-down: group tokens that likely belong to the same object

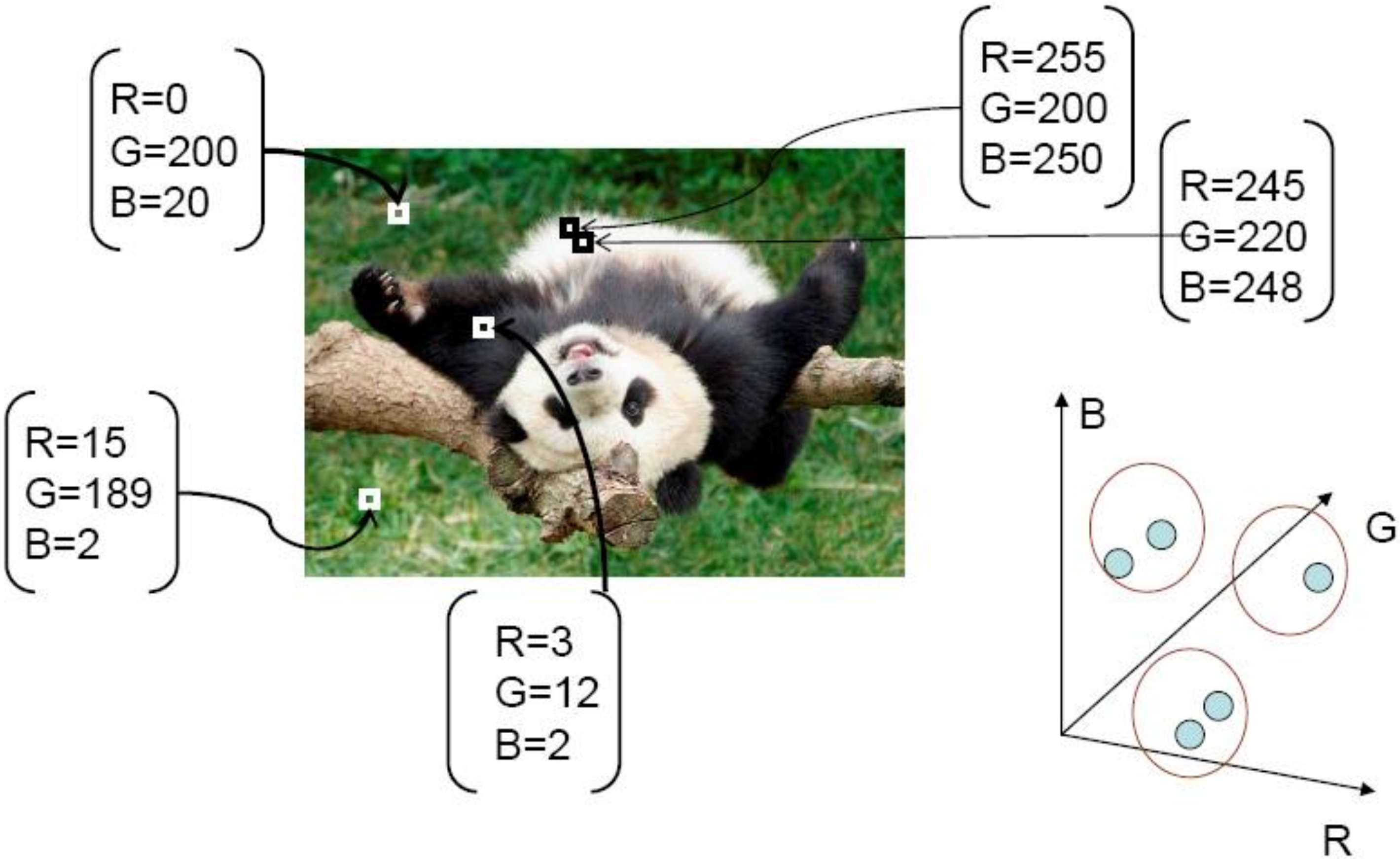


[Levin and Weiss 2006]

Segmentation using clustering

- Kmeans
- Mean-shift

Feature Space



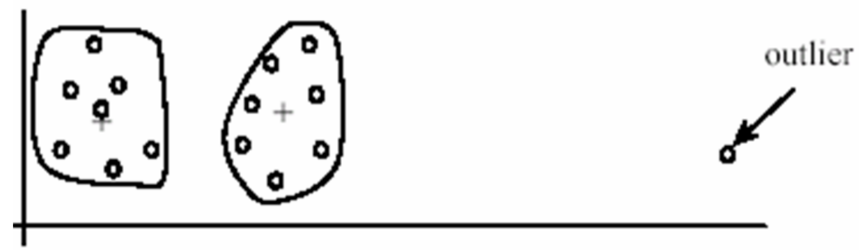
Source: K. Grauman

K-means clustering using intensity alone and color alone

Input image	Clusters on intensity	Clusters on color
		

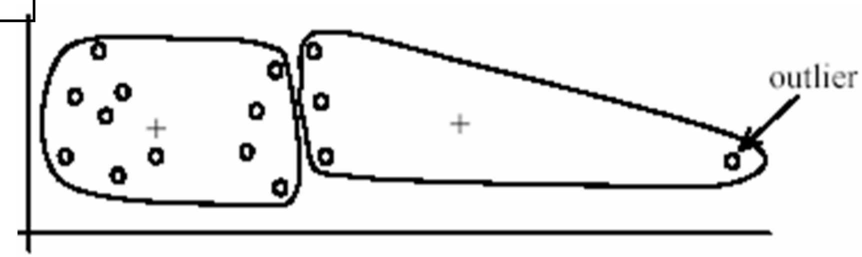
K-Means pros and cons

- Pros
  - Simple and fast
  - Easy to implement



(B): Ideal clusters

- Cons
  - Need to choose K
  - Sensitive to outliers



- Usage
  - Rarely used for pixel segmentation

It;s not good for segmentation
it's good for clustering
its good for quantization, in different way as we have seen above
Dimensional reduction

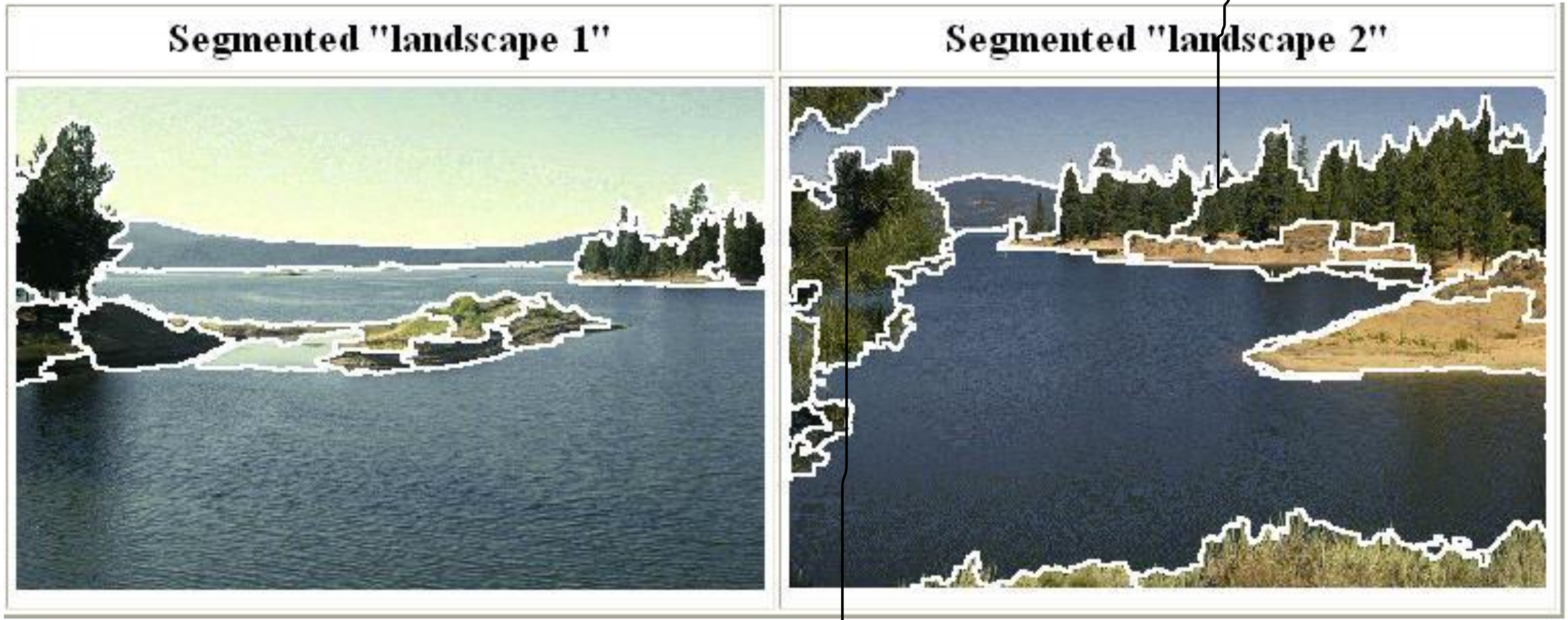
one of the method for segmentation, based on clustering, not as k means, it based on probability

Mean shift segmentation

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

- Versatile technique for clustering-based segmentation

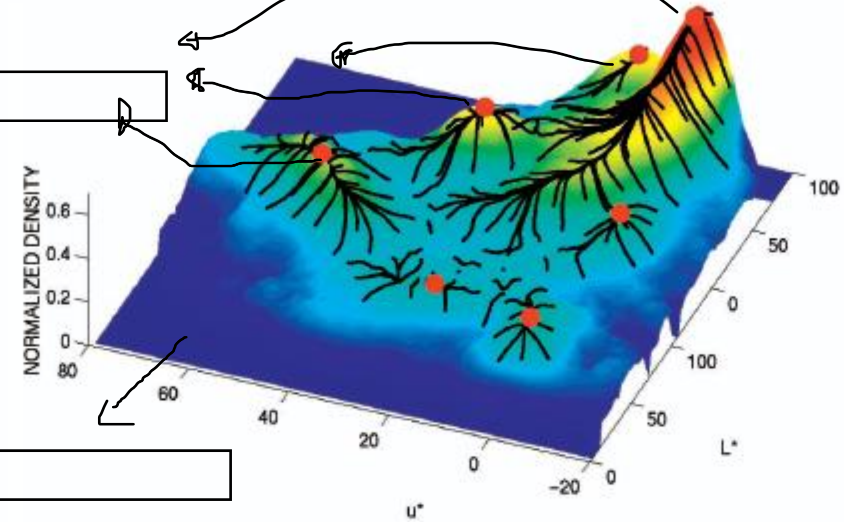
one of the problems, is segmentation that no need to be segmentated

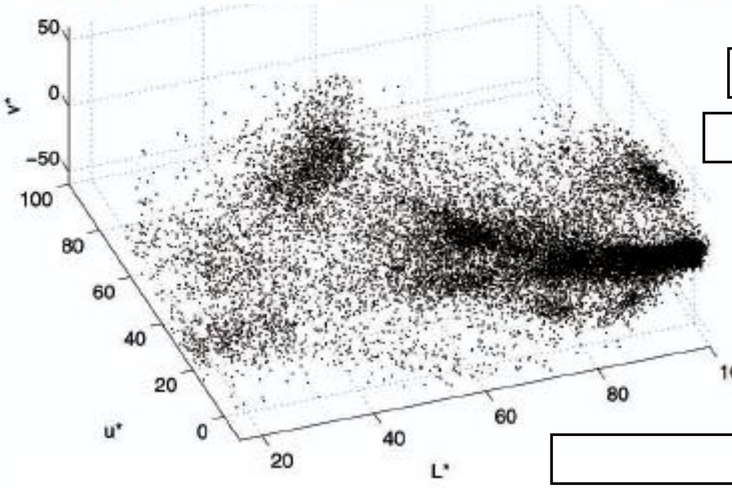


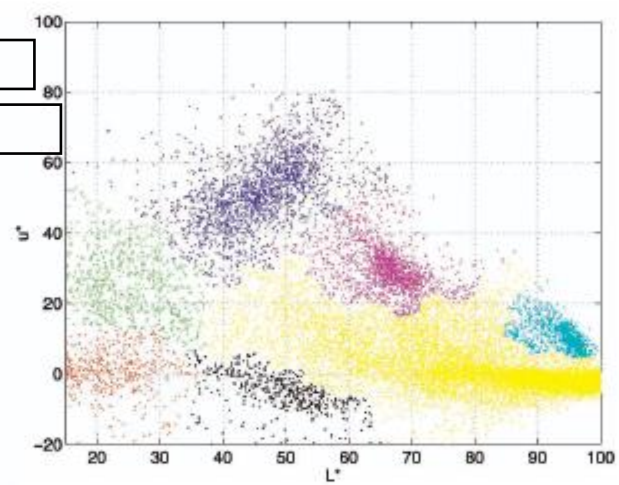
this suppose to be more segmented

Mean shift algorithm

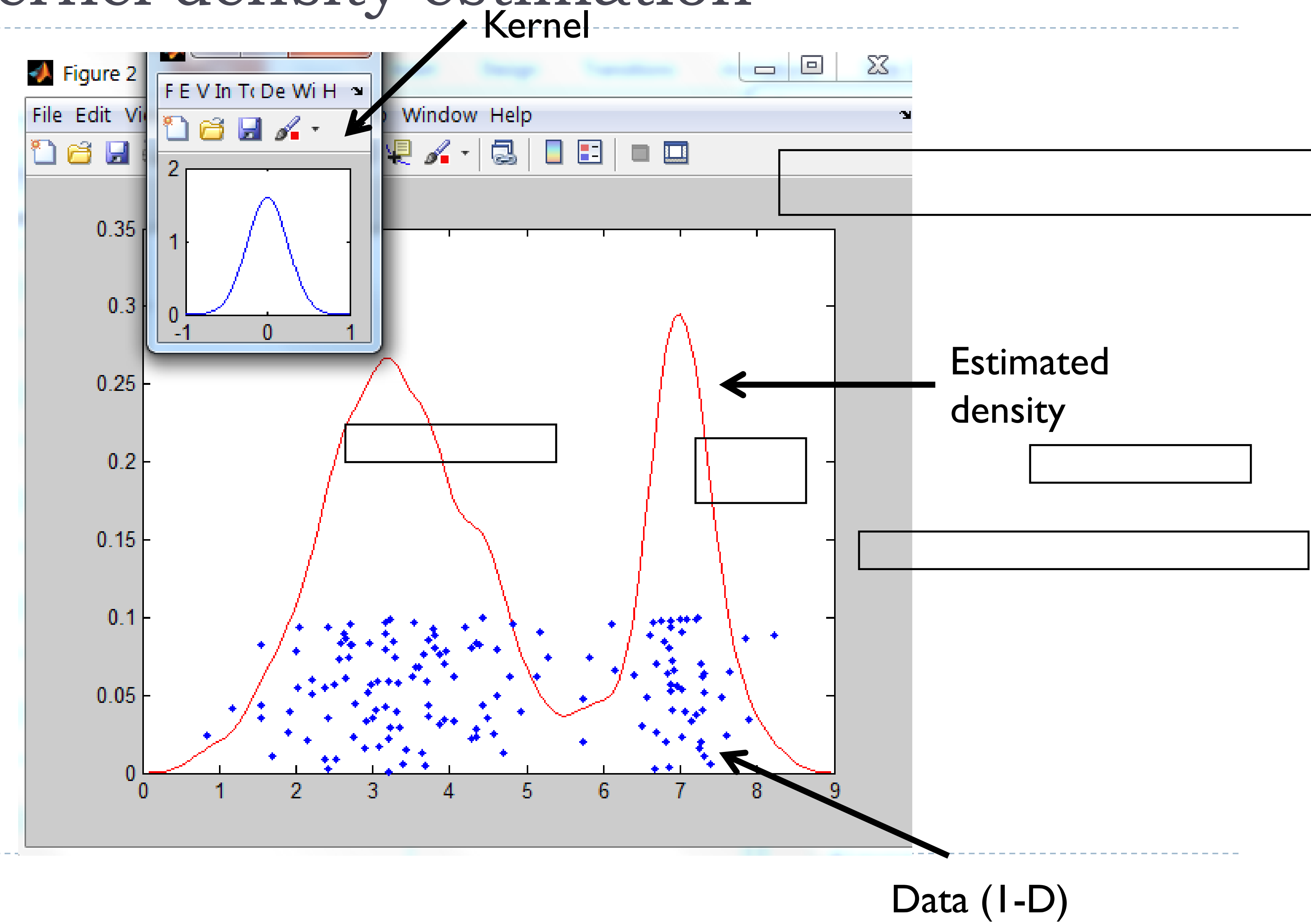
- Try to find *modes* of this non-parametric density







Kernel density estimation



Kernel density estimation

Kernel density estimation function

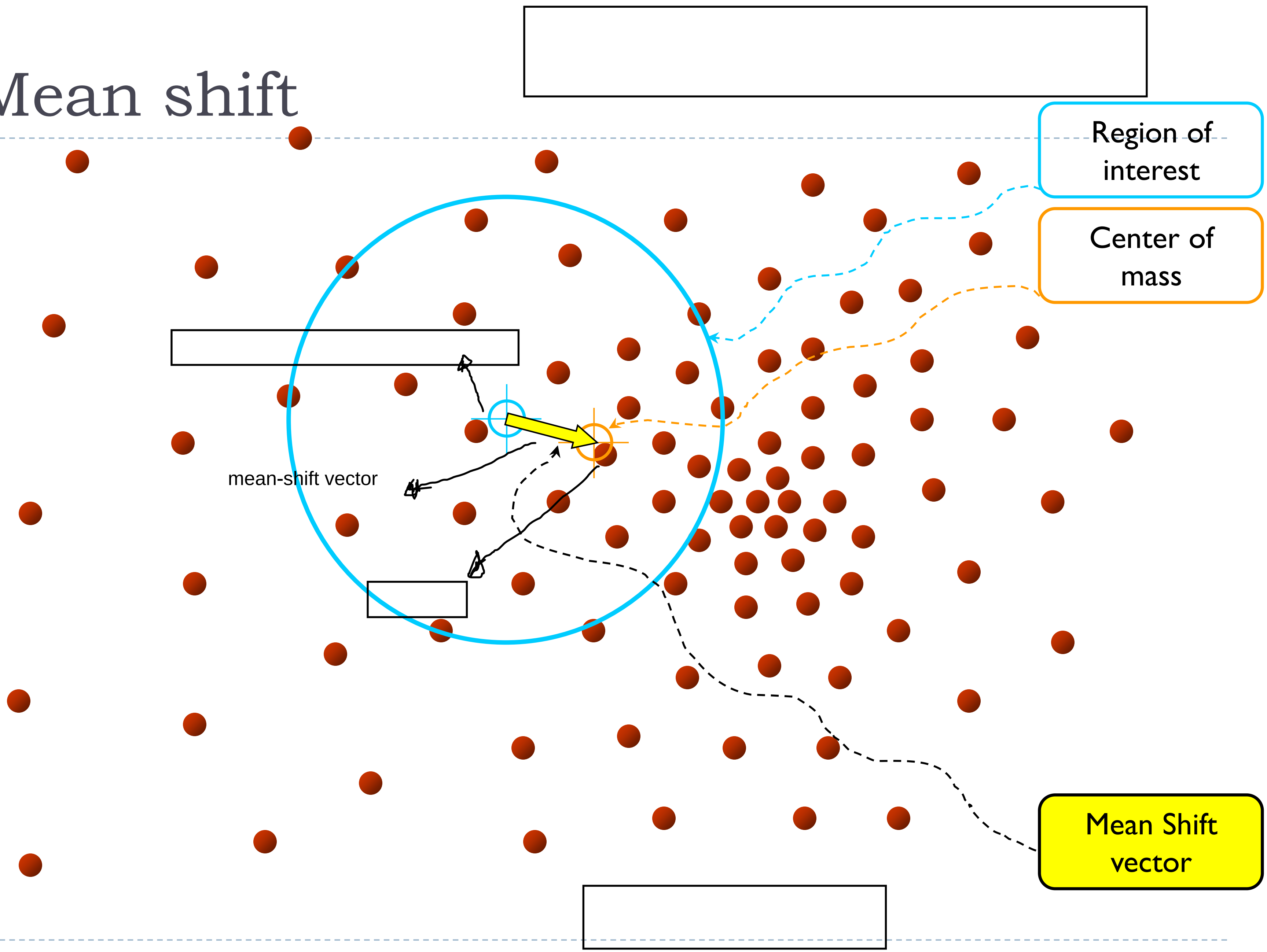
$$\widehat{f}_h(x) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right)$$

x x_i

Gaussian kernel

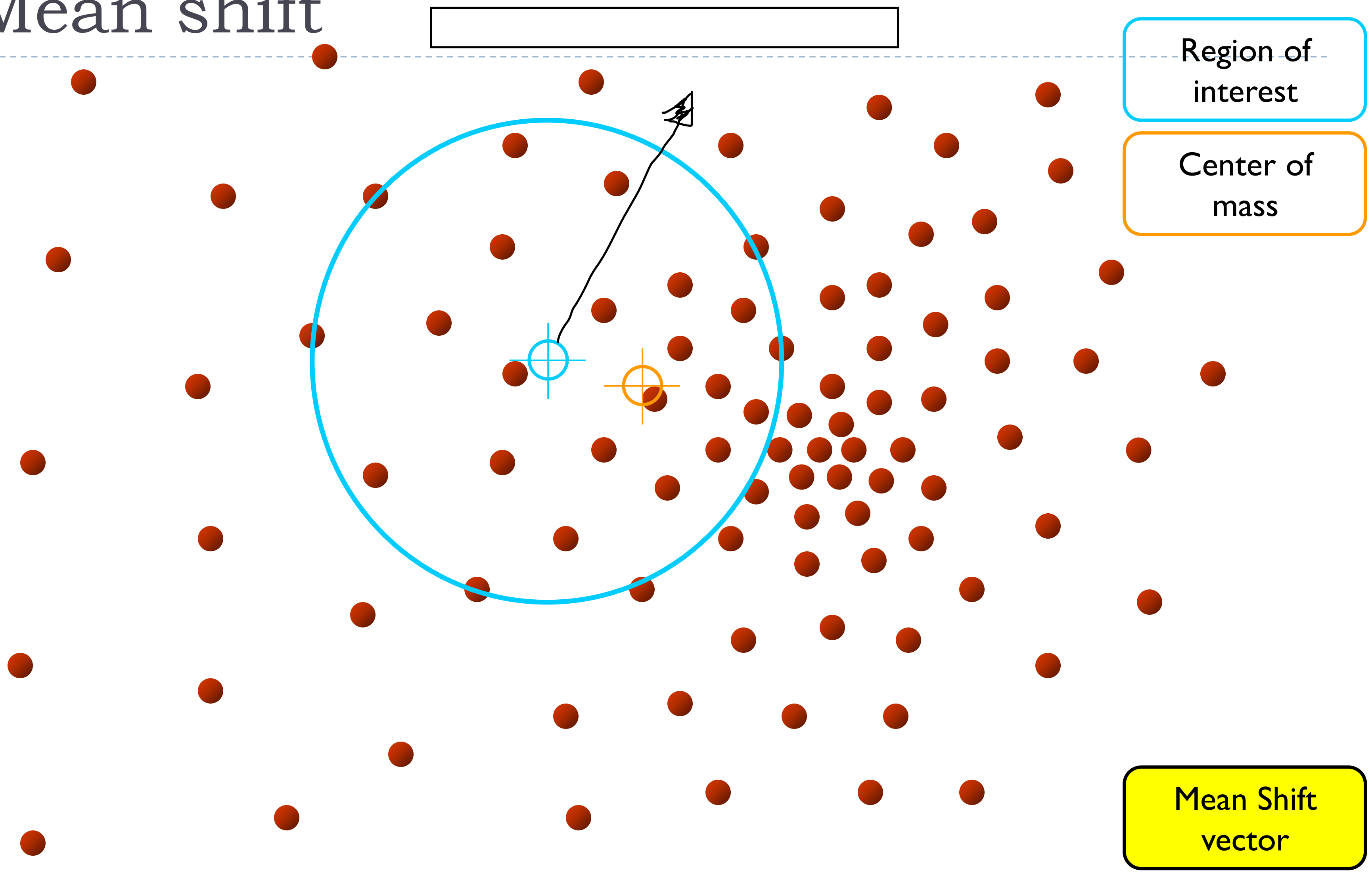
$$K\left(\frac{x - x_i}{h}\right) = \frac{1}{\sqrt{2\pi}} e^{-\frac{(x-x_i)^2}{2h^2}}.$$

Mean shift



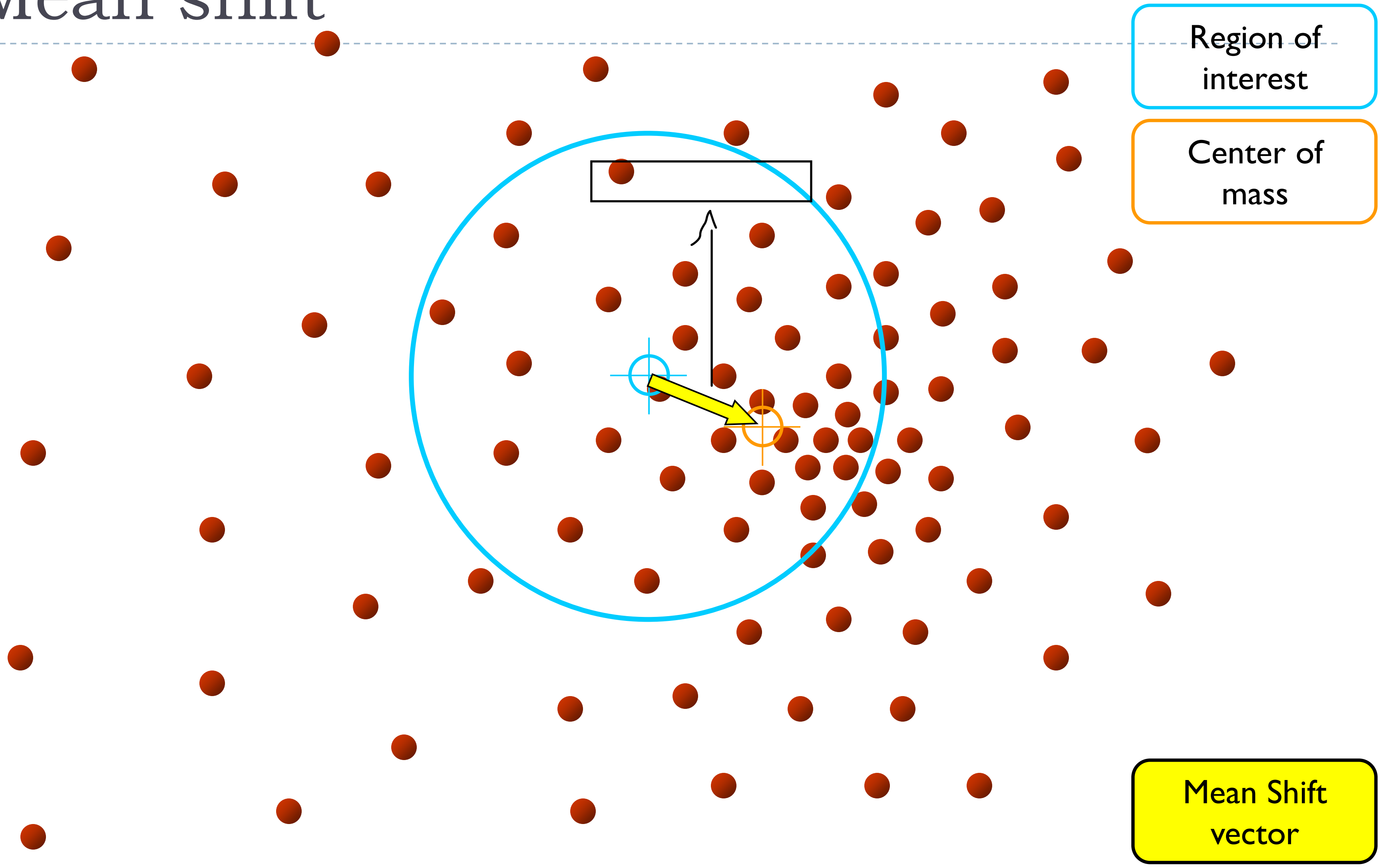
Region of interest
Center of mass
Mean Shift vector
mean-shift vector

Mean shift

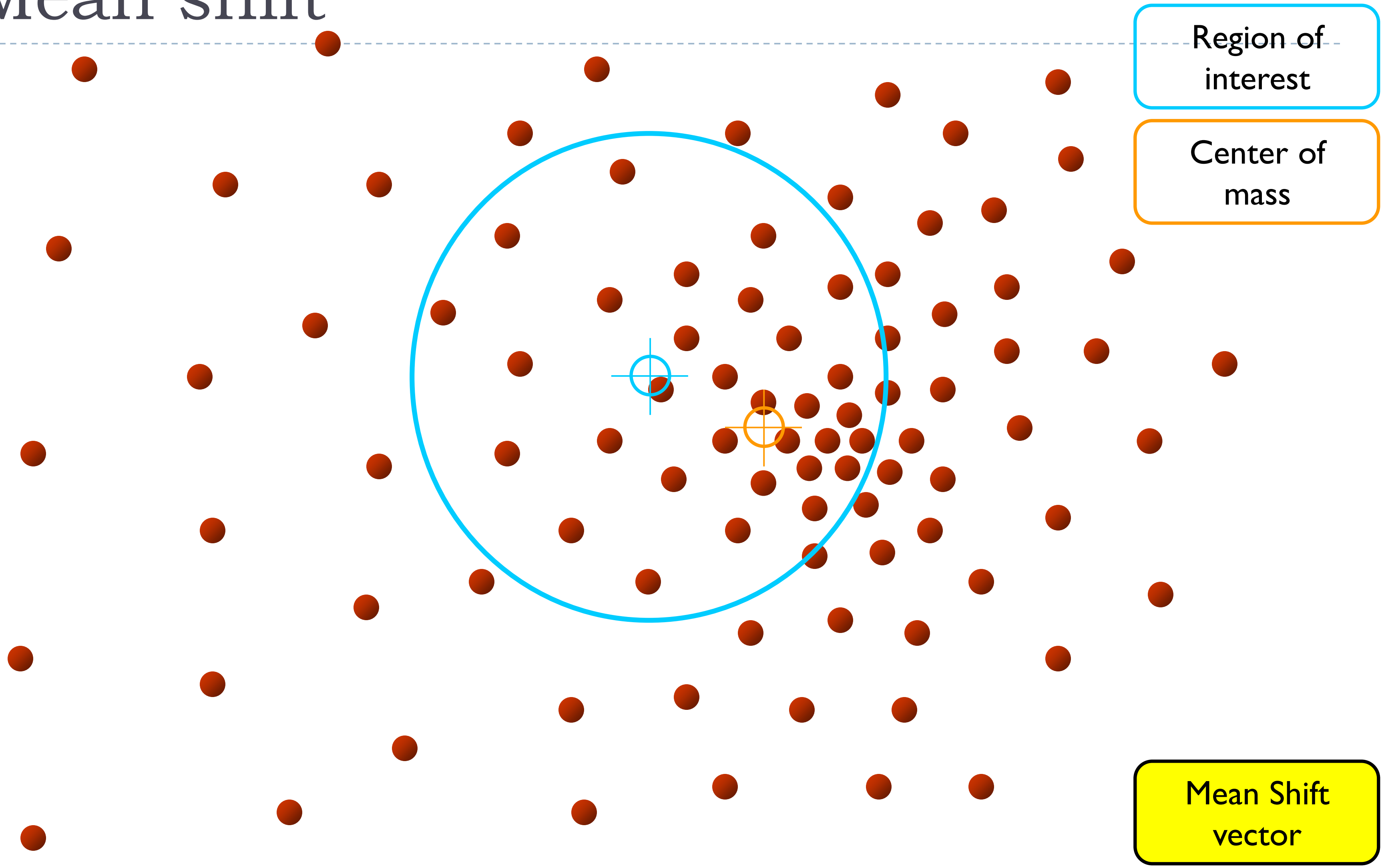


Slide by Y. Ukrainitz & B. Sarel

Mean shift

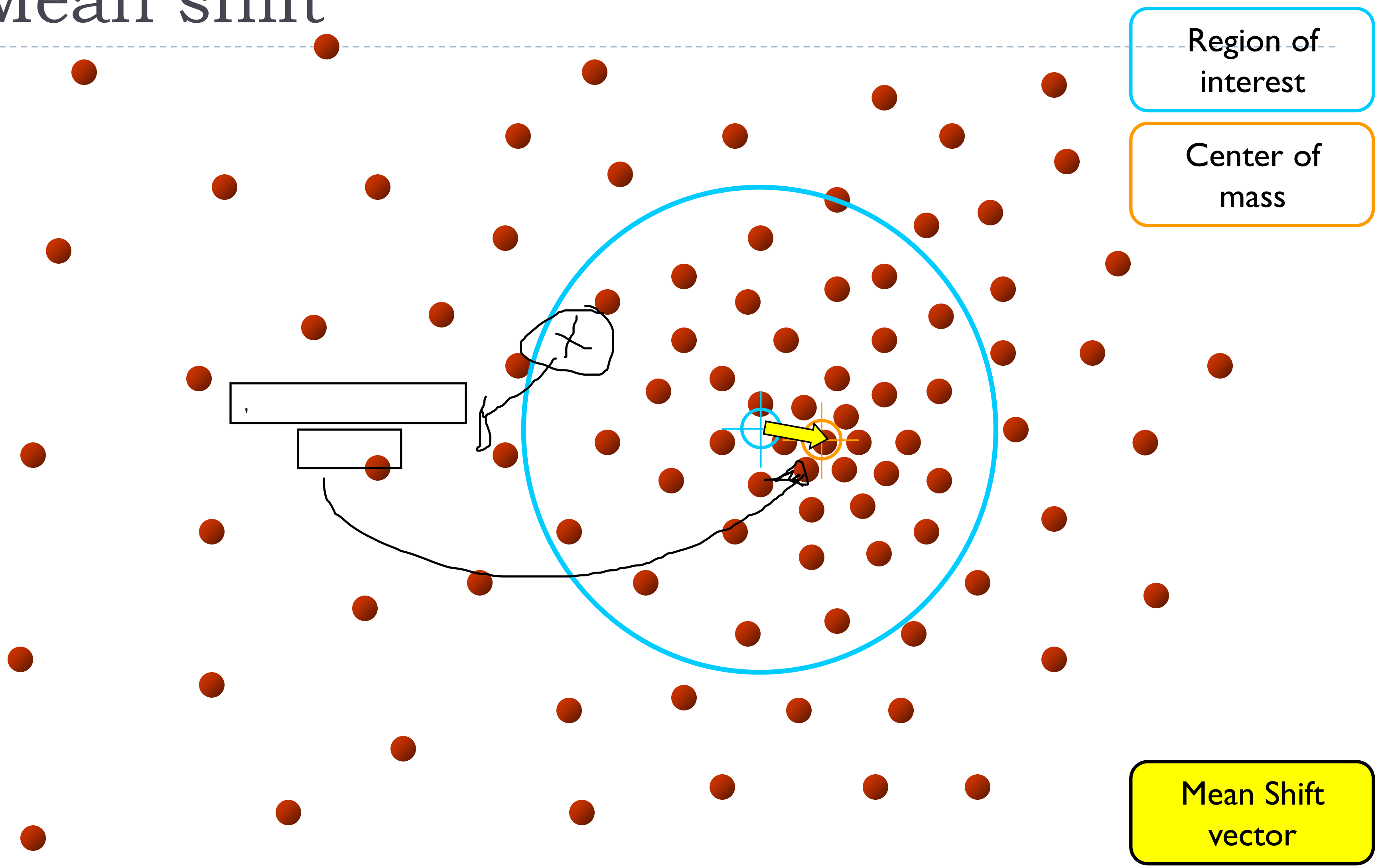


Mean shift



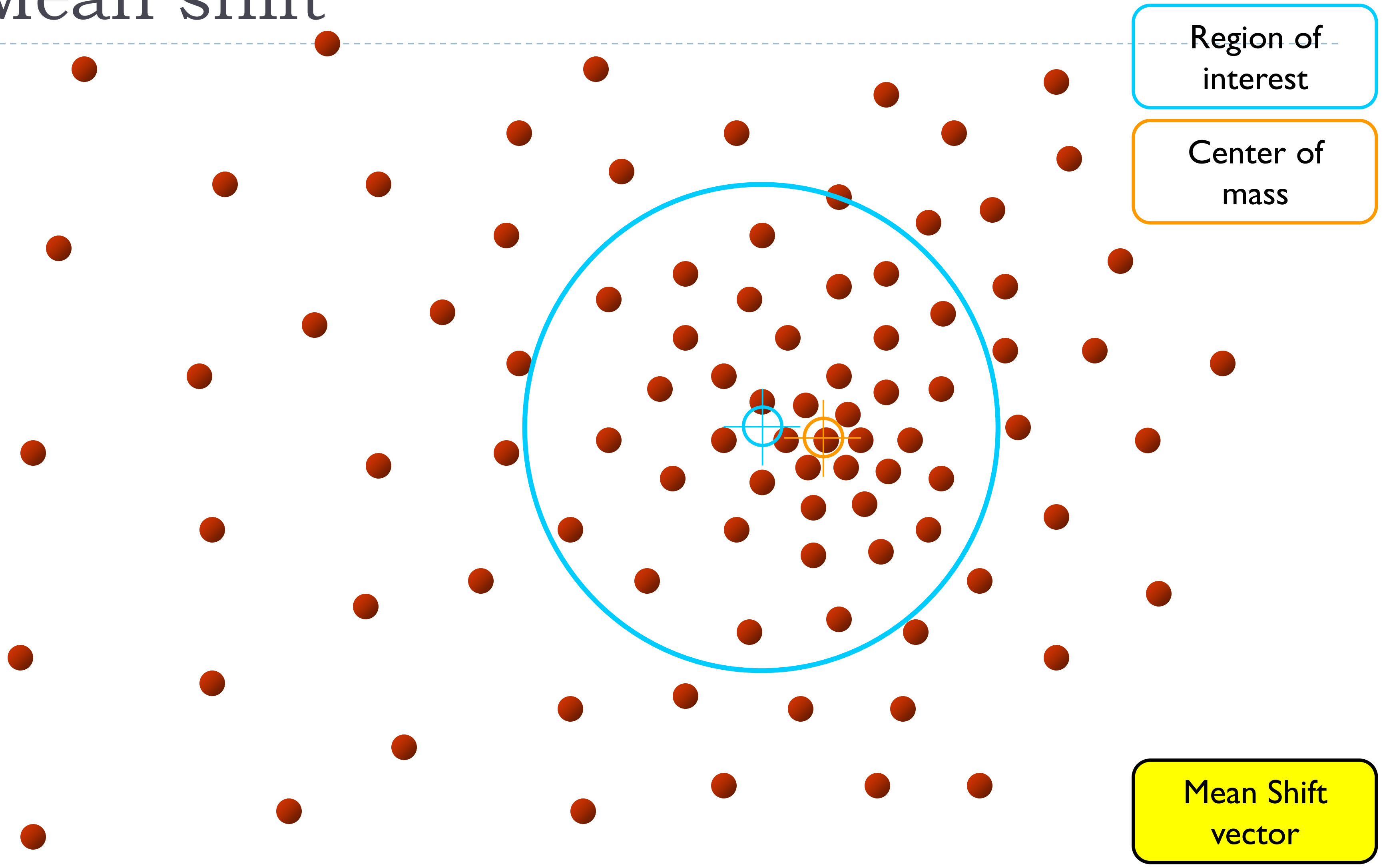
Slide by Y. Ukrainitz & B. Sarel

Mean shift



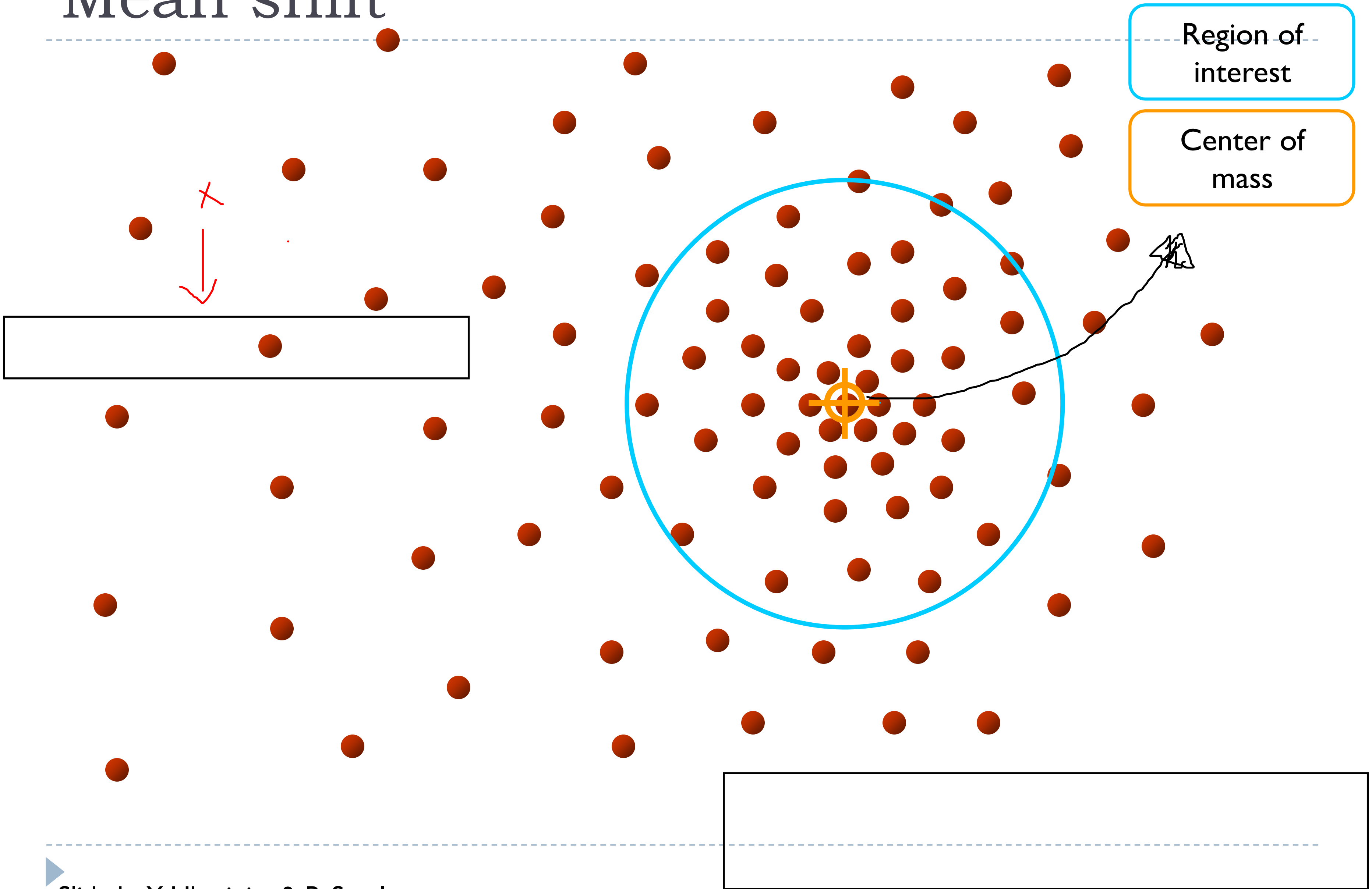
Slide by Y. Ukrainitz & B. Sarel

Mean shift



Slide by Y. Ukrainitz & B. Sarel

Mean shift



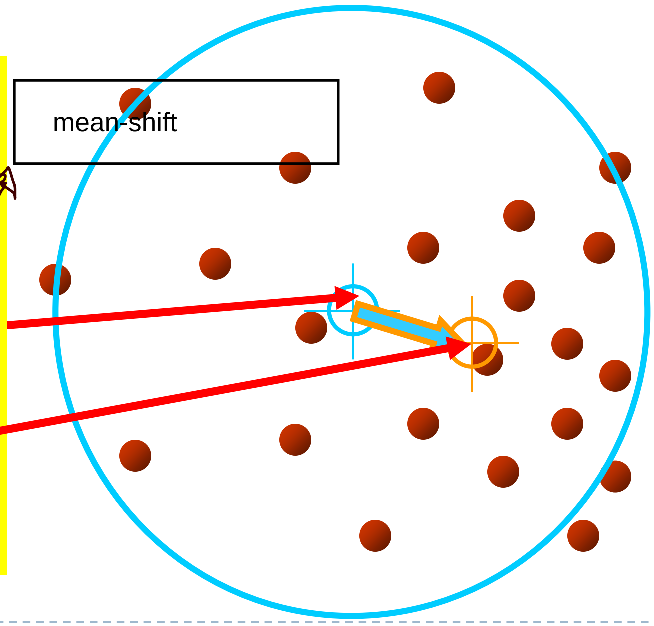
Computing the Mean Shift

Simple Mean Shift procedure:

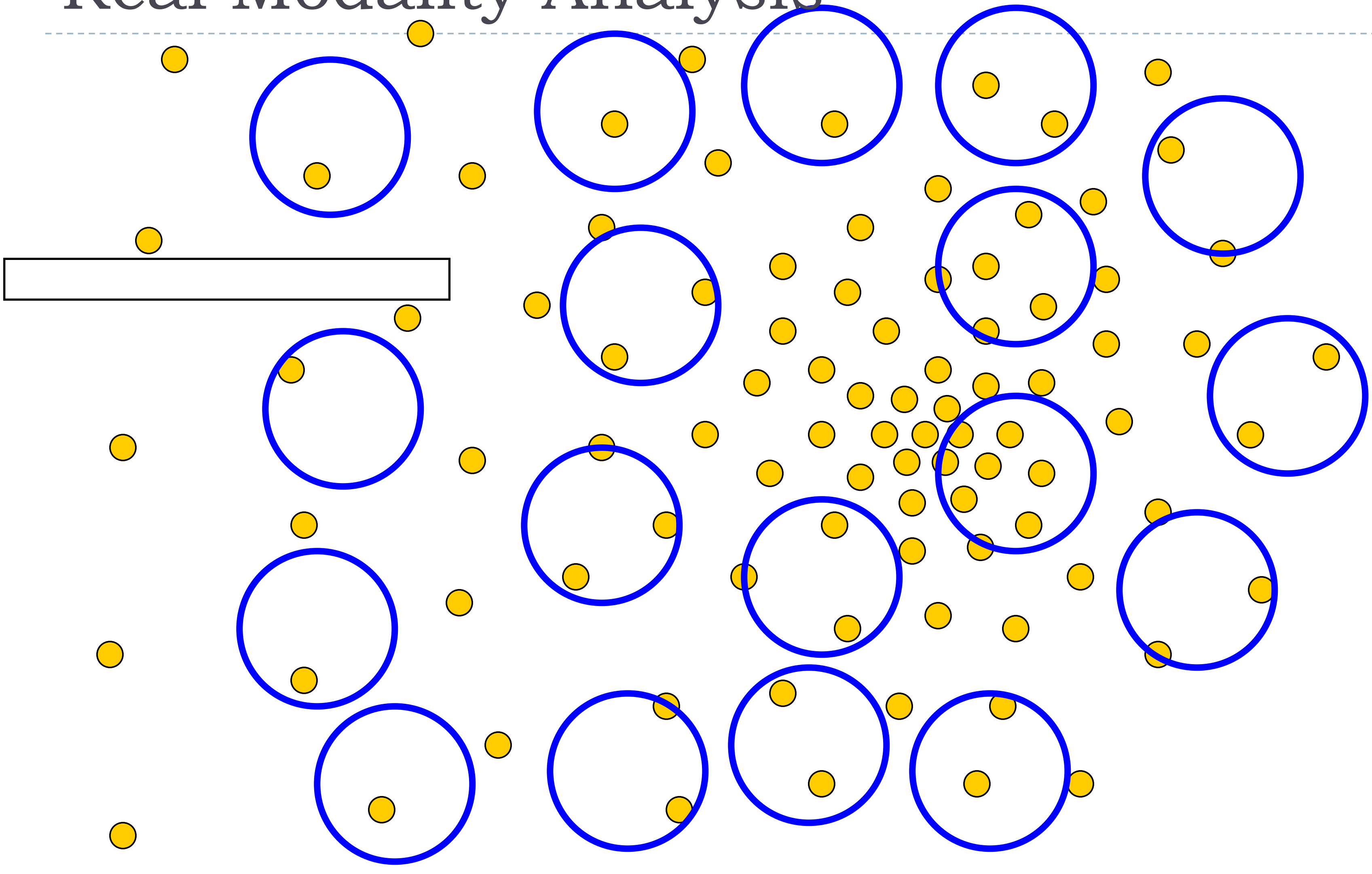
- Compute mean shift vector

- Translate the Kernel window by **m(x)**

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

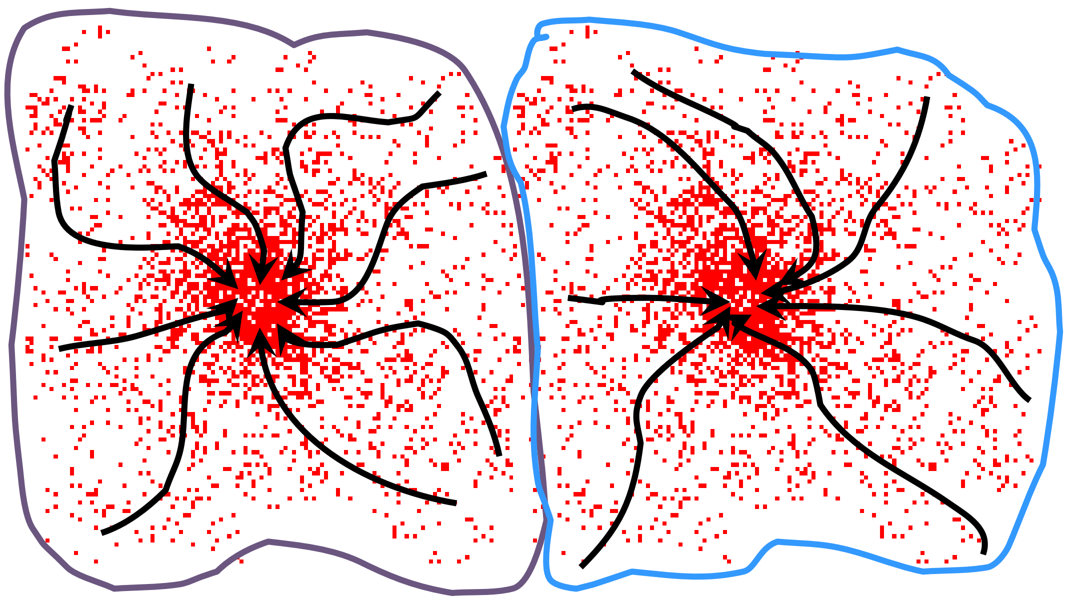


Real Modality Analysis

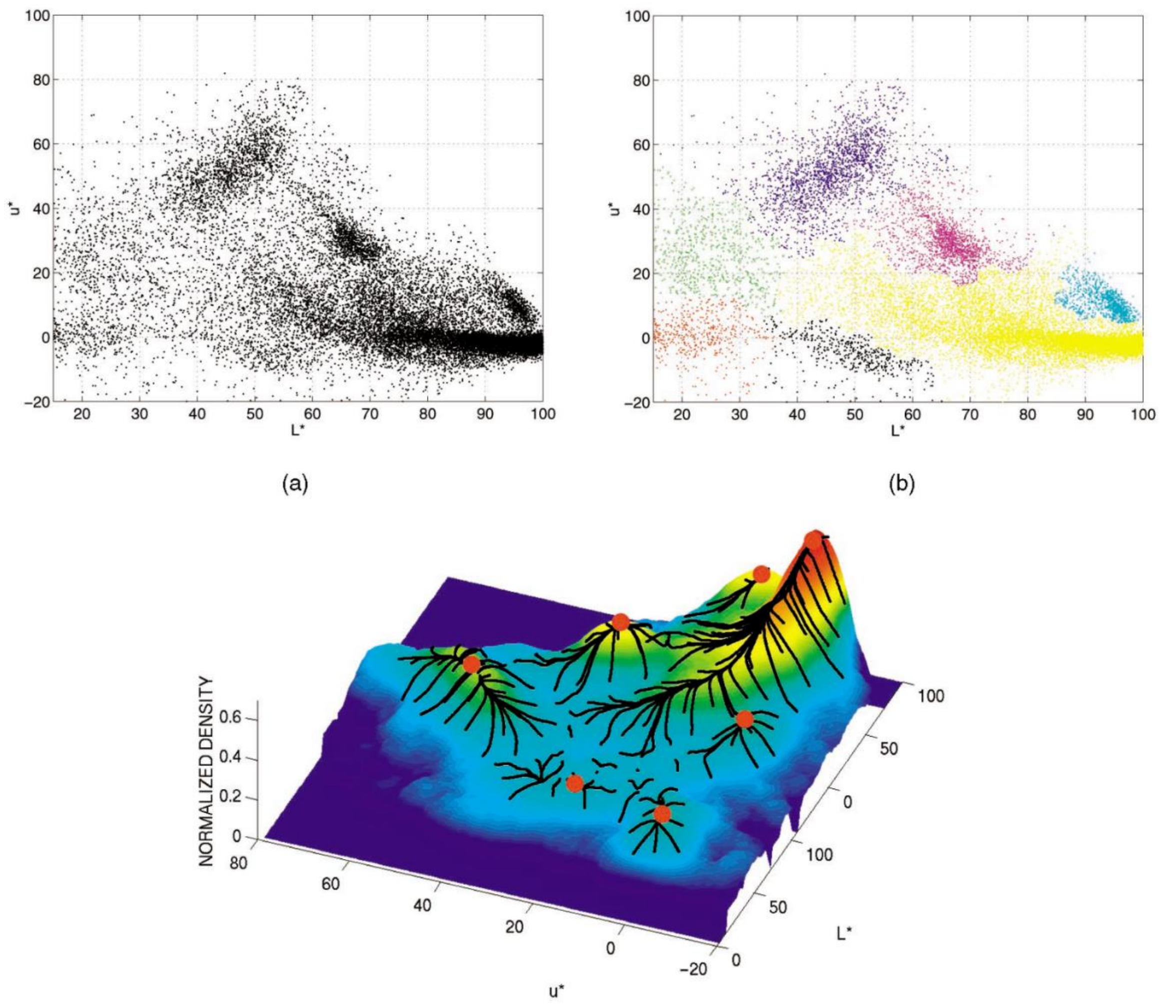


Attraction basin

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



Attraction basin

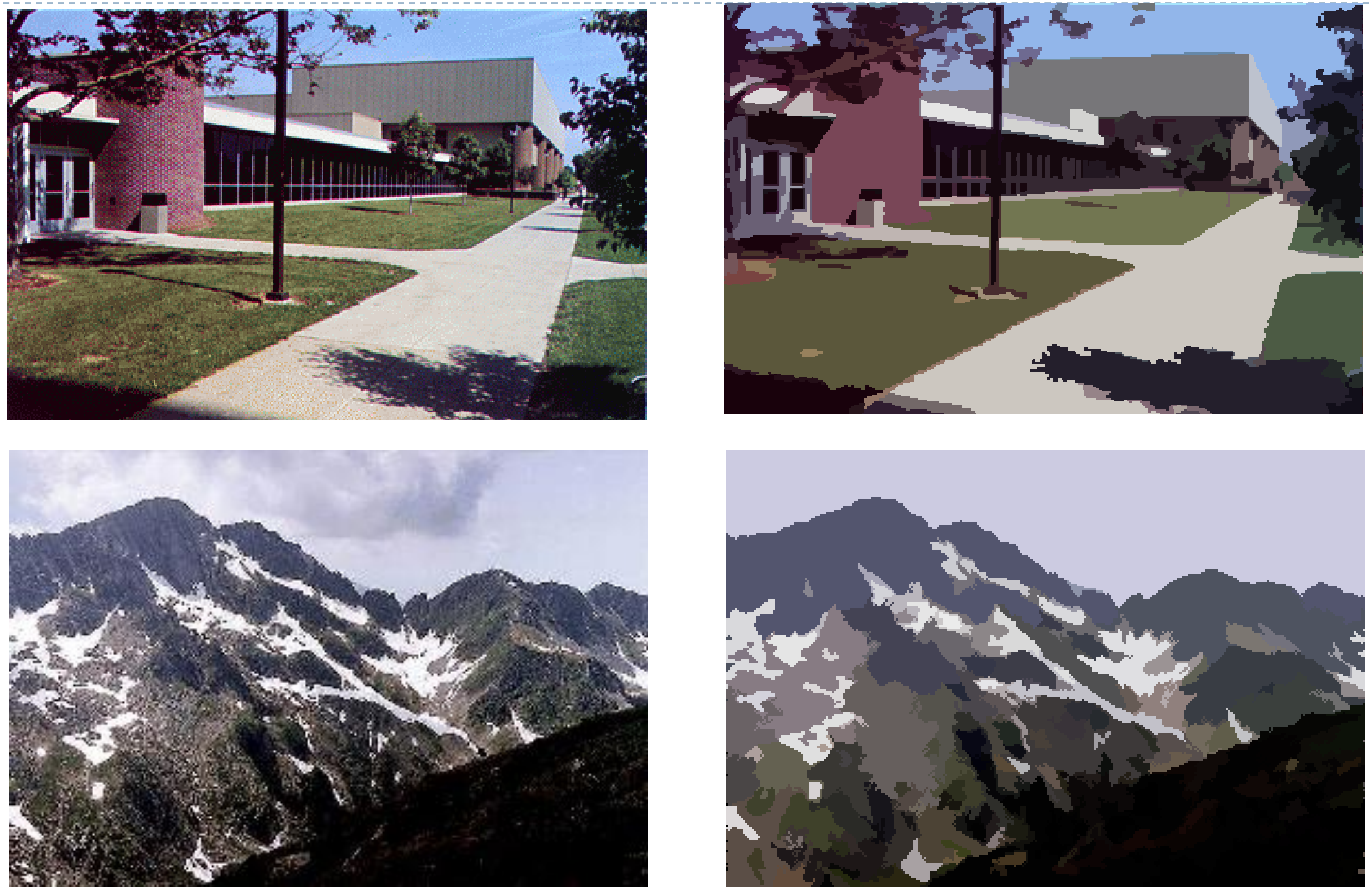


Mean shift clustering

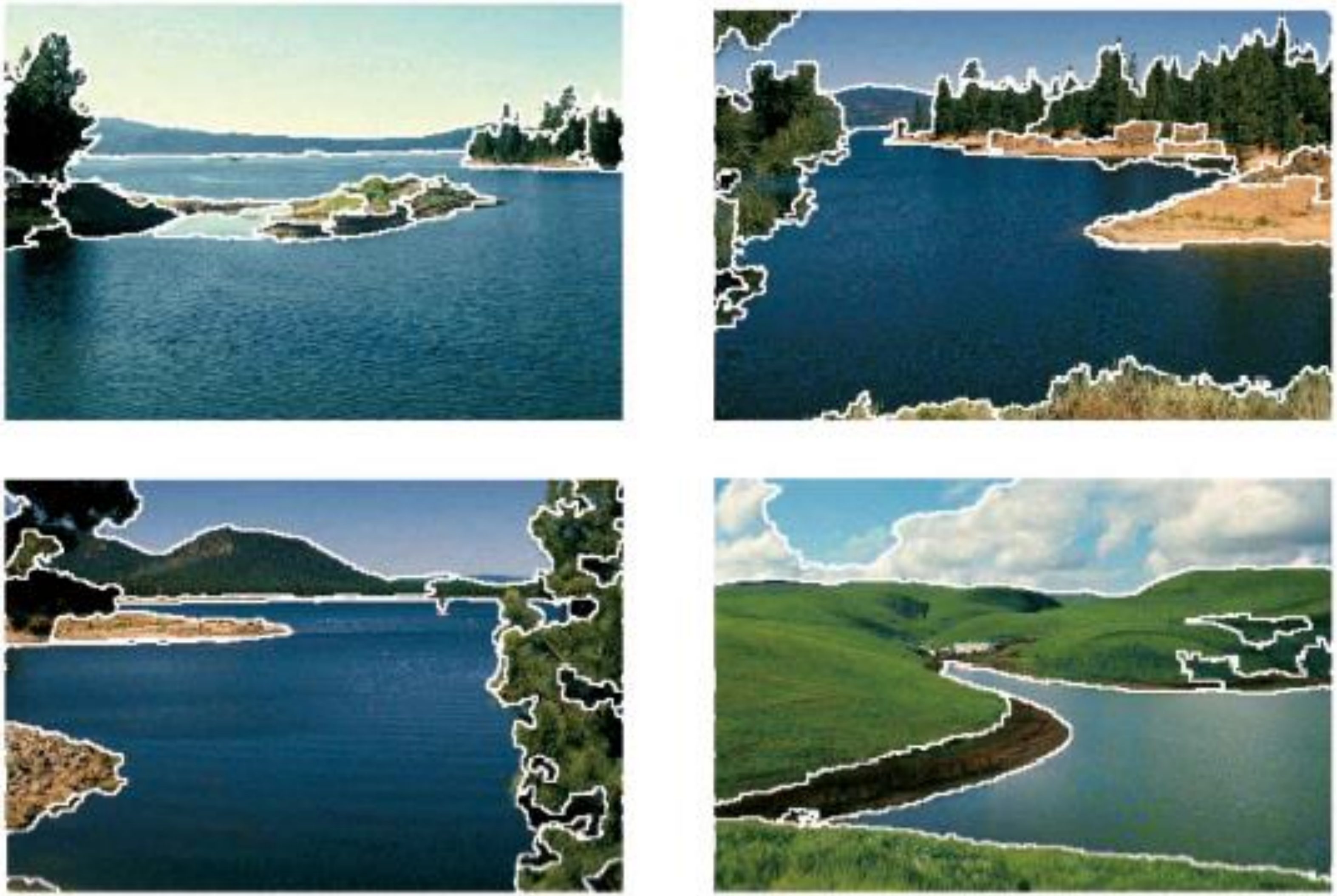
- The mean shift algorithm seeks *modes* of the given set of points
  1. Choose kernel and bandwidth
  2. For each point:
     a) Center a window on that point
     b) Compute the mean of the data in the search window
     c) Center the search window at the new mean location
     d) Repeat (b,c) until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

mean shift

Mean shift segmentation results



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



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

Mean-shift: other issues

- Speedups
  - Binned estimation – replace points within some "bin" by point at center with mass
  - Fast search of neighbors – e.g., k-d tree or approximate NN
  - Update all windows in each iteration (faster convergence)

- Other tricks
  - Use kNN to determine window sizes adaptively

- Lots of theoretical support

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

Mean shift pros and cons

- Pros
  - Good general-purpose segmentation
  - Flexible in number and shape of regions
  - Robust to outliers
  - General mode-finding algorithm (useful for other problems such as finding most common surface normals)
- 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

Mean-shift reading

- Nicely written mean-shift explanation (with math)

http://saravananthirumuruganathan.wordpress.com/2010/04/01/introduction-to-mean-shift-algorithm/

  - Includes .m code for mean-shift clustering

- Mean-shift paper by Comaniciu and Meer

http://www.caip.rutgers.edu/~comanici/Papers/MsRobustApproach.pdf

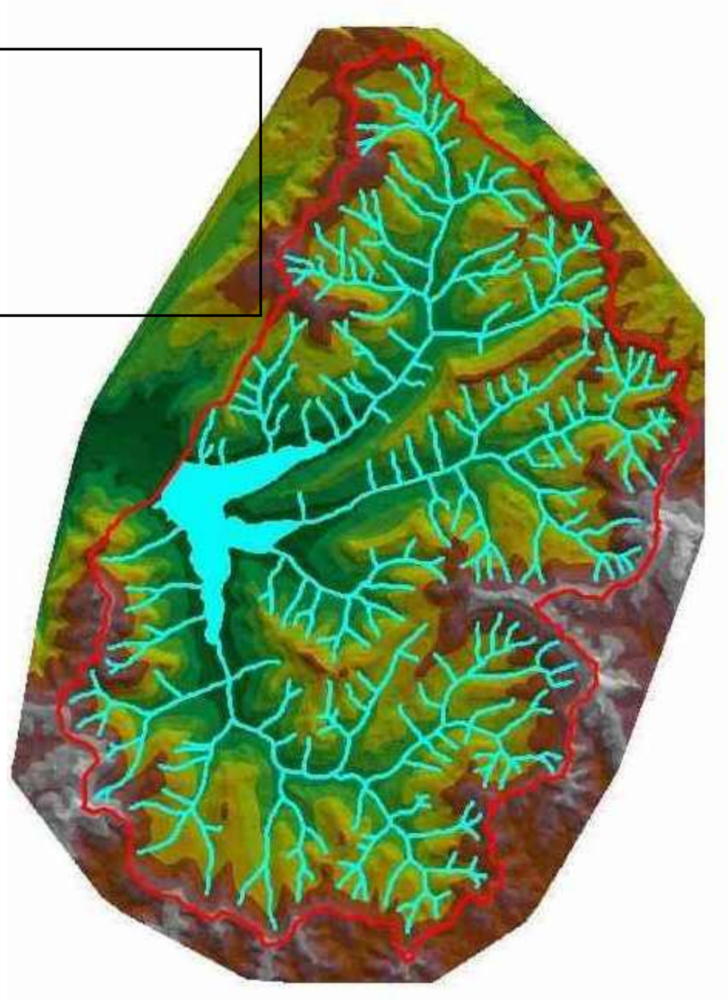
- Adaptive mean shift in higher dimensions

http://mis.hevra.haifa.ac.il/~ishimshoni/papers/chap9.pdf

Superpixel algorithms

- Goal is to divide the image into a large number of regions, such that each regions lie within object boundaries
- Examples
  - Watershed
  - Felzenszwalb and Huttenlocher graph-based
  - Turbopixels
  - SLIC

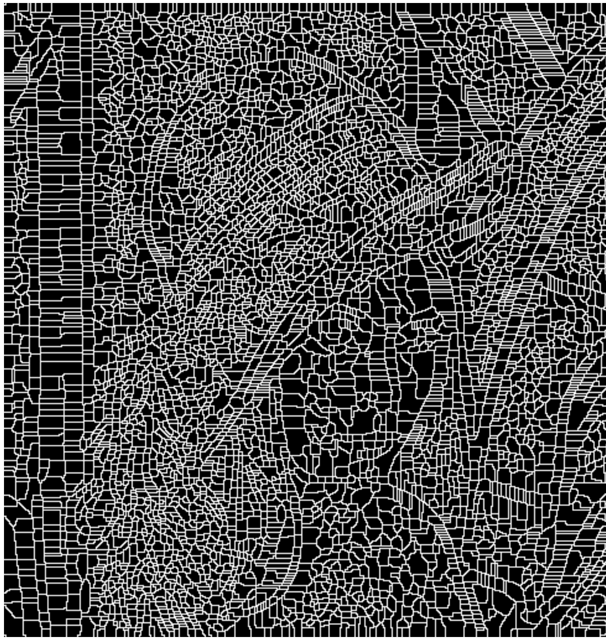
Watershed algorithm



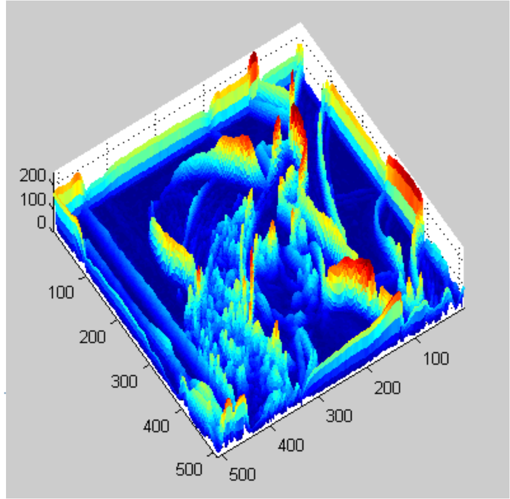
Watershed segmentation

Image

Gradient



Watershed boundaries



Meyer's watershed segmentation

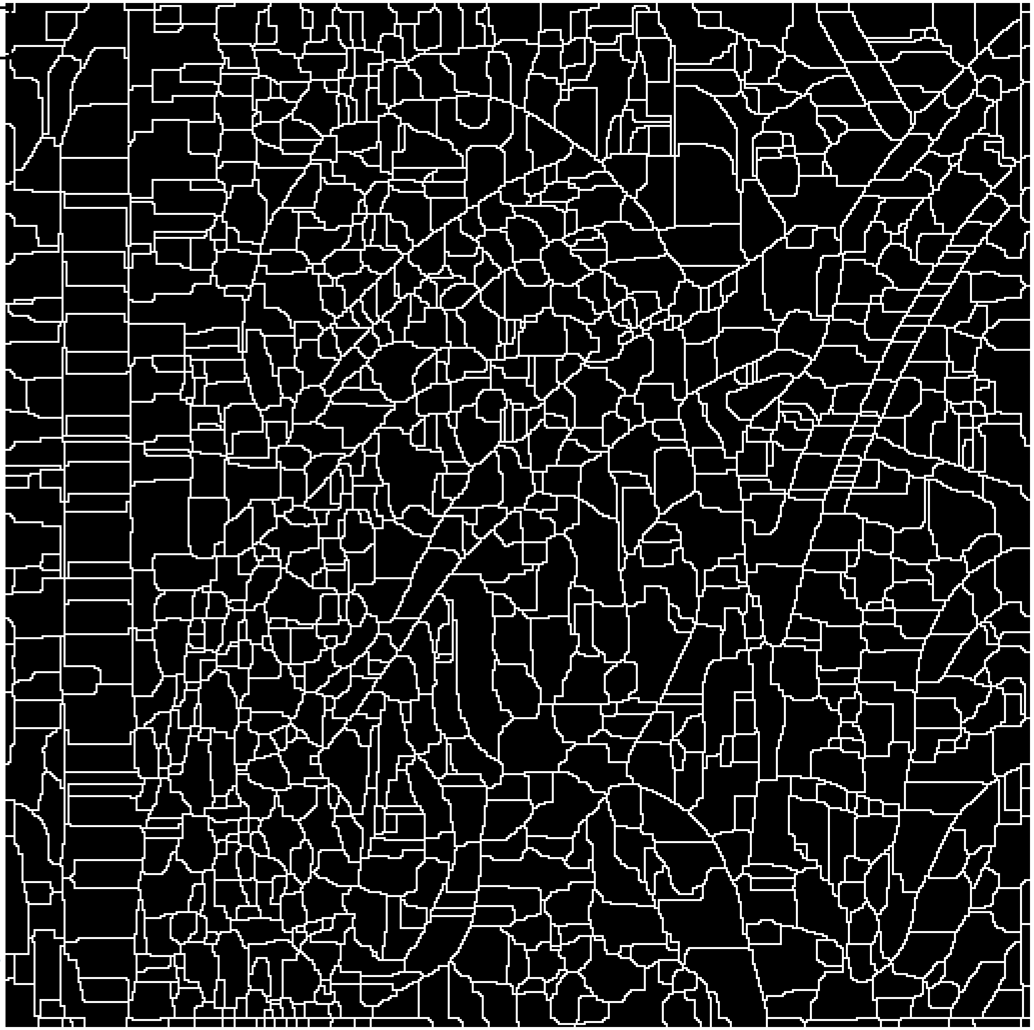
1. Choose local minima as region seeds
2. Add neighbors to priority queue, sorted by value
3. Take top priority pixel from queue
   1. If all labeled neighbors have same label, assign that label to pixel
   2. Add all non-marked neighbors to queue
4. Repeat step 3 until finished (all remaining pixels in queue are on the boundary)

Matlab: seg = watershed(bnd_im)

Meyer 1991

Simple trick

- Use Gaussian or median filter to reduce number of regions



Watershed usage

- Use as a starting point for hierarchical segmentation
  - Ultrametric contour map (Arbelaez 2006)

- Works with any soft boundaries
  - Pb (w/o non-max suppression)
  - Canny (w/o non-max suppression)
  - Etc.

Watershed pros and cons

- Pros
  - Fast (< 1 sec for 512x512 image)
  - Preserves boundaries
- Cons
  - Only as good as the soft boundaries (which may be slow to compute)
  - Not easy to get variety of regions for multiple segmentations
- Usage
  - Good algorithm for superpixels, hierarchical segmentation

Felzenszwalb and Huttenlocher: Graph-Based Segmentation

http://www.cs.brown.edu/~pff/segment/

Algorithm 1 *Segmentation algorithm.*

The input is a graph $G = (V, E)$, with n vertices and m edges. The output is a segmentation of V into components $S = (C_1, \ldots, C_r)$.

0. Sort E into $\pi = (o_1, \ldots, o_m)$, by non-decreasing edge weight.
1. Start with a segmentation S^0, where each vertex v_i is in its own component.
2. Repeat step 3 for $q = 1, \ldots, m$.
3. Construct S^q given S^{q-1} as follows. Let v_i and v_j denote the vertices connected by the q-th edge in the ordering, i.e., $o_q = (v_i, v_j)$. If v_i and v_j are in disjoint components of S^{q-1} and $w(o_q)$ is small compared to the internal difference of both those components, then merge the two components otherwise do nothing. More formally, let C_i^{q-1} be the component of S^{q-1} containing v_i and C_j^{q-1} the component containing v_j. If $C_i^{q-1} \neq C_j^{q-1}$ and $w(o_q) \leq MInt(C_i^{q-1}, C_j^{q-1})$ then S^q is obtained from S^{q-1} by merging C_i^{q-1} and C_j^{q-1}. Otherwise $S^q = S^{q-1}$.
4. Return $S = S^m$.

Felzenszwalb and Huttenlocher: Graph-Based Segmentation

http://www.cs.brown.edu/~pff/segment/

+ Good for thin regions
+ Fast
+ Easy to control coarseness of segmentations
+ Can include both large and small regions
- Often creates regions with strange shapes
- Sometimes makes very large errors

SLIC (Achanta et al. PAMI 2012)

http://infoscience.epfl.ch/record/177415/files/Superpixel_PAMI2011-2.pdf

1. Initialize cluster centers on pixel grid in steps S
   - Features: Lab color, x-y position
2. Move centers to position in 3x3 window with smallest gradient
3. Compare each pixel to cluster center within 2S pixel distance and assign to nearest
4. Recompute cluster centers as mean color/position of pixels belonging to each cluster
5. Stop when residual error is small

+ Fast 0.36s for 320x240
+ Regular superpixels
+ Superpixels fit boundaries
- May miss thin objects
- Large number of superpixels

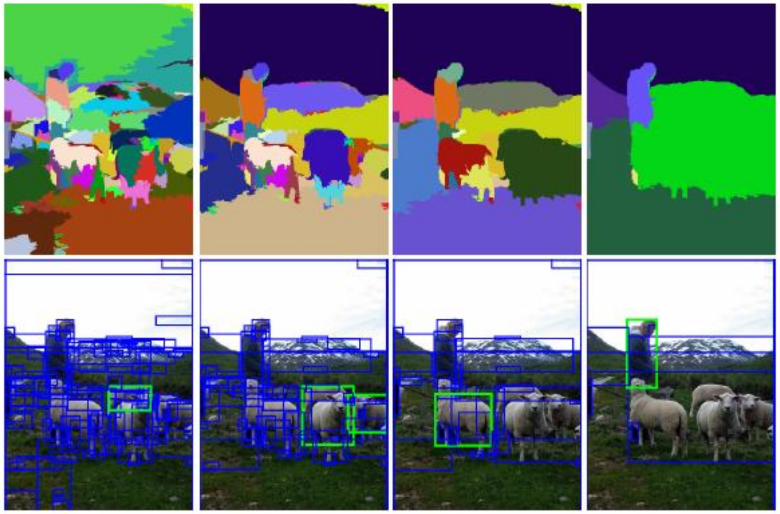
Choices in segmentation algorithms

- Oversegmentation
  - Watershed + Pb ← my favorite
  - Felzenszwalb and Huttenlocher 2004 ← pretty good

    http://www.cs.brown.edu/~pff/segment/
  - SLIC ← also a good option
  - Turbopixels
  - Mean-shift

- Larger regions
  - Hierarchical segmentation (e.g., from Pb)
  - Normalized cuts
  - Mean-shift
  - Seed + graph cuts (discussed later)

Multiple segmentations

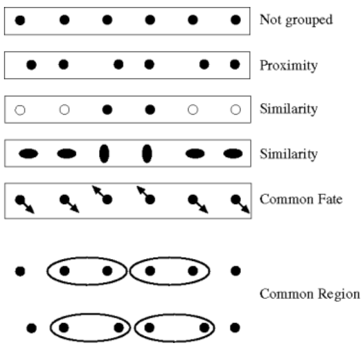


- When creating regions for pixel classification or object detection, don't commit to one partitioning

- Strategies:
  - Hierarchical segmentation
    - Occlusion boundaries hierarchy: Hoiem et al. IJCV 2011 (uses trained classifier to merge)
    - Pb+watershed hierarchy: Arbeleaz et al. CVPR 2009
    - Selective search: FH + agglomerative clustering
  - Vary segmentation parameters
    - E.g., multiple graph-based segmentations or mean-shift segmentations
  - Region proposals
    - Propose seed superpixel, try to segment out object that contains it (Endres Hoiem ECCV 2010, Carreira Sminchisescu CVPR 2010)

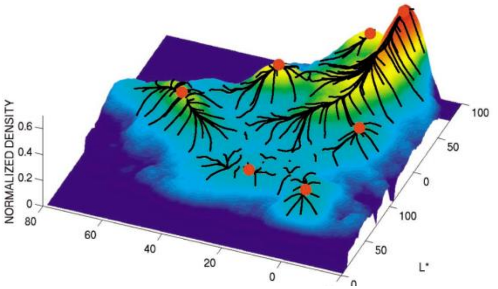
Things to remember

- Gestalt cues and principles of organization

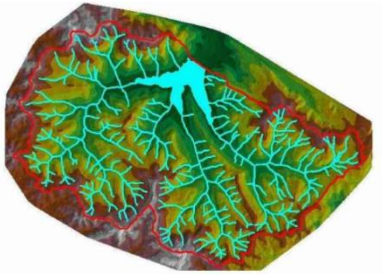


- Uses of segmentation
  - Efficiency
  - Better features
  - Propose object regions
  - Want the segmented object

- Mean-shift segmentation
  - Good general-purpose segmentation method
  - Generally useful clustering, tracking technique



- Watershed segmentation
  - Good for hierarchical segmentation
  - Use in combination with boundary prediction



Slide credits

- Slides borrowed from Derek Hoiem

End – segmentation part 1

Now you know how it works