CMPSCI 670: Computer Vision
Grouping

University of Massachusetts, Amherst
October 14, 2014

Instructor: Subhransu Maji

Slides credit: Kristen Grauman and others

Administrivia

- Final project guidelines posted
- Milestones
  - October 27: Abstract due
  - December 1, 3: Project presentations
  - December 13: Final report due
- Final project is graded as **two** homework assignments
  - The scope of the project should be roughly equal to **two** homework assignments.
  - Scales linearly with the team size.
  - Form your own teams.
- Alternatively, you can do a literature survey (solo)

Outline

- What are grouping problems in vision?

- Inspiration from human perception
  - Gestalt properties

- Bottom-up segmentation via clustering
  - Algorithms:
    - Mode finding and mean shift: k-means, mean-shift
    - Graph-based: graph cuts, normalized cuts
  - Features: color, texture, …
    - Quantization for texture summaries

Grouping in vision

- Goals:
  - Gather features that belong together
  - Obtain an intermediate representation that compactly describes key image or video parts for further processing
    - This is a computational complexity argument

Examples of grouping in vision

[Figure by J. Shi]

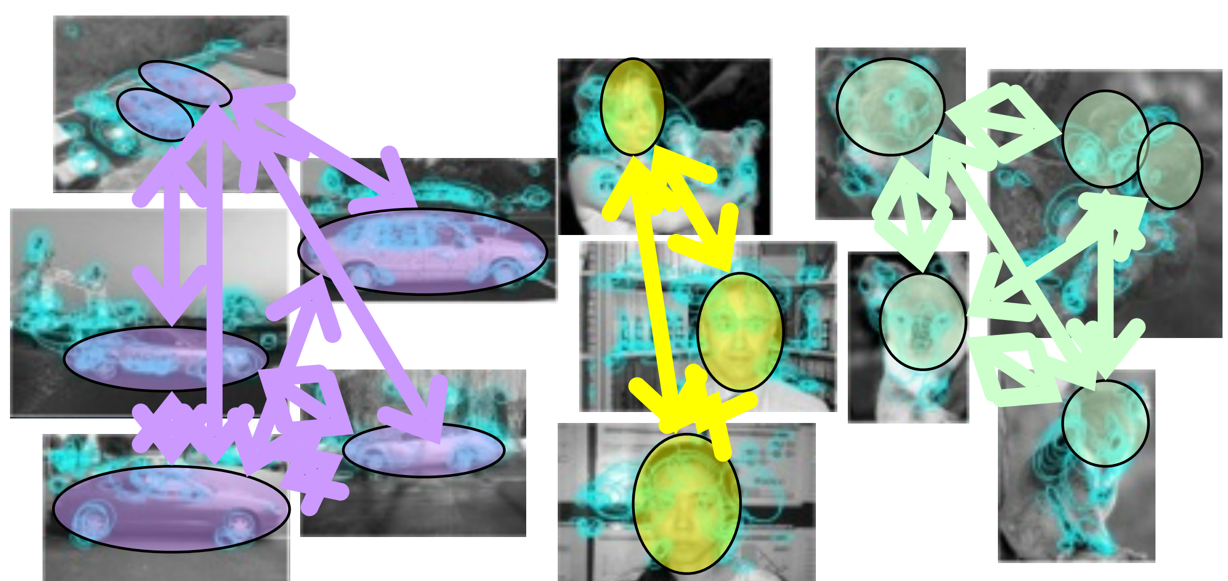
Determine image regions

[http://poseidon.csd.auth.gr/LAB_RESEARCH/Latest/imgs/SpeakDepVidIndex_img2.jpg]

Group video frames into shots

[Figure by Wang & Suter]

Figure-ground



[Figure by Grauman & Darrell]

Object-level grouping

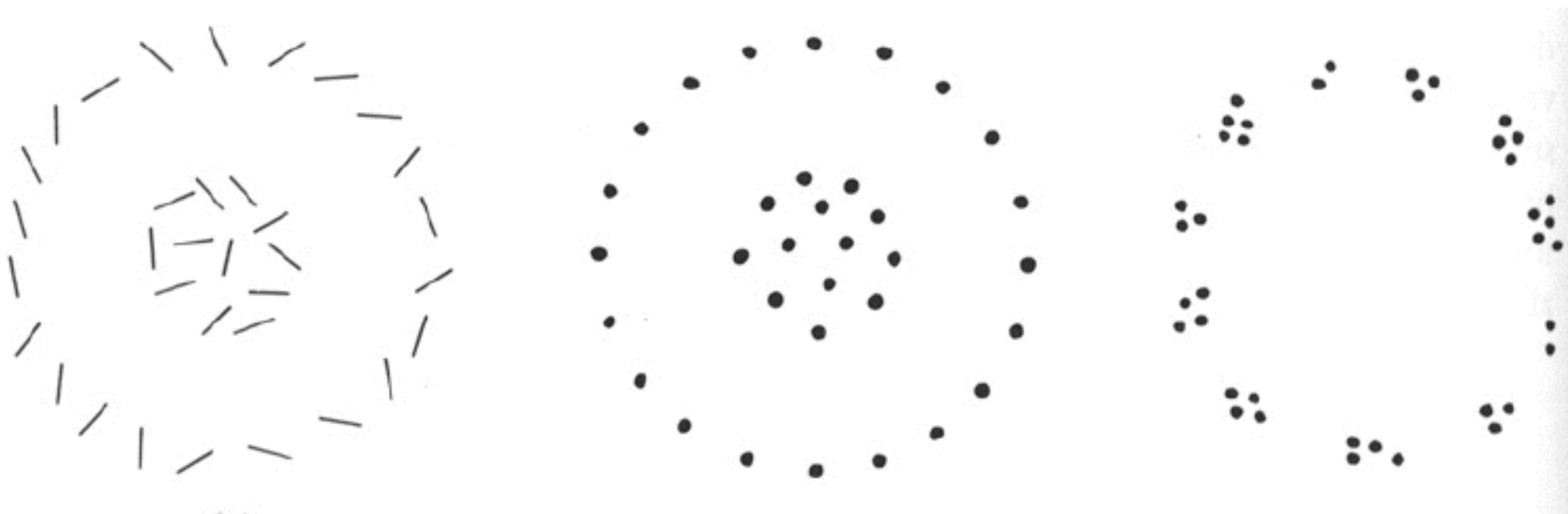
Grouping in vision

- Goals:
  - Gather features that belong together
  - Obtain an intermediate representation that compactly describes key image (video) parts

- Top down vs. bottom up segmentation
  - Top down: pixels belong together because they are from the same object
  - Bottom up: pixels belong together because they look similar

- Hard to measure success
  - What is interesting depends on the application

What are the groups?



Questions …

- What things should be grouped?
- What cues indicate grouping?

Gestalt

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

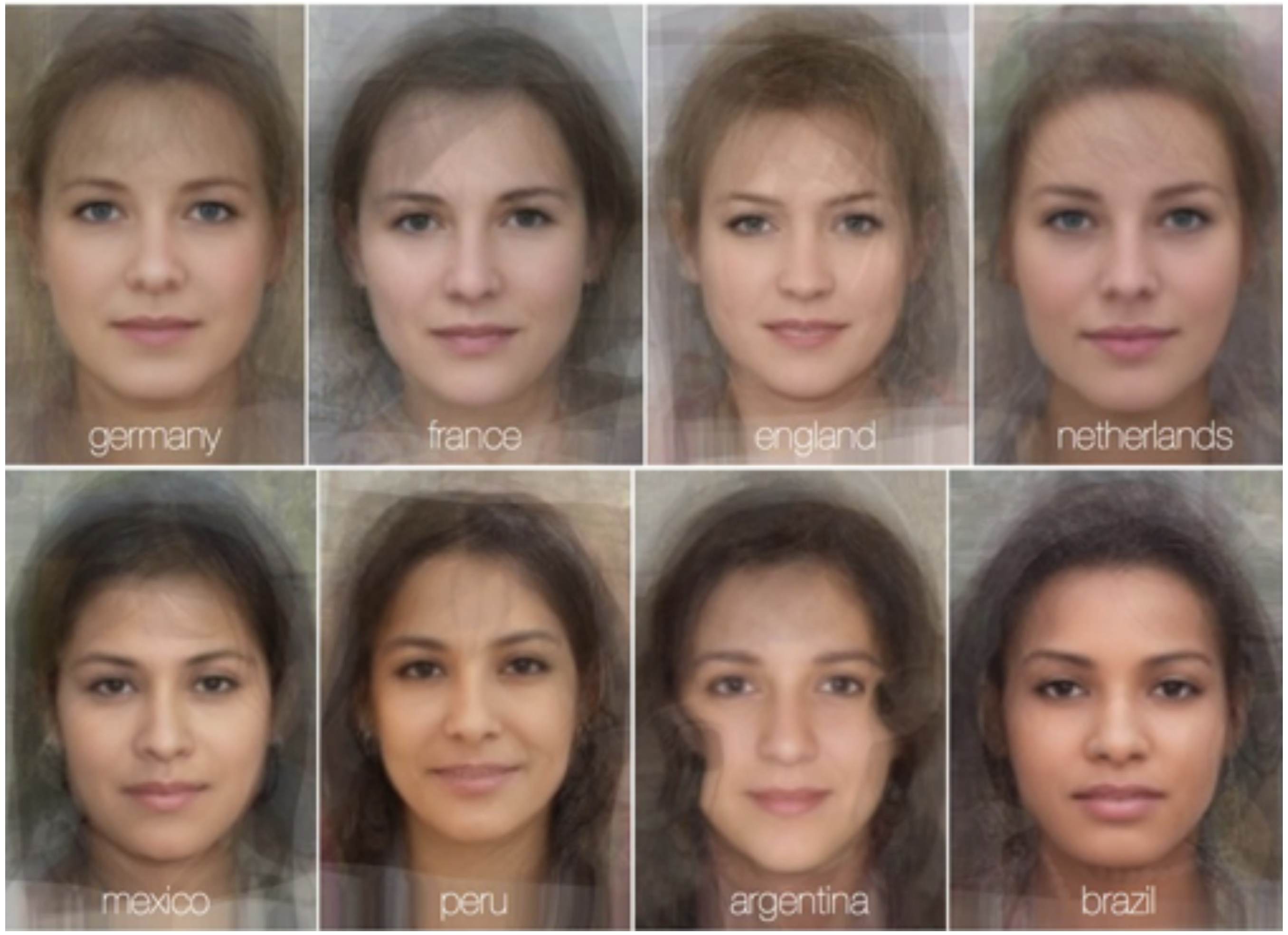
- Psychologists identified series of factors that predispose set of elements to be grouped (by human visual system)

Similarity

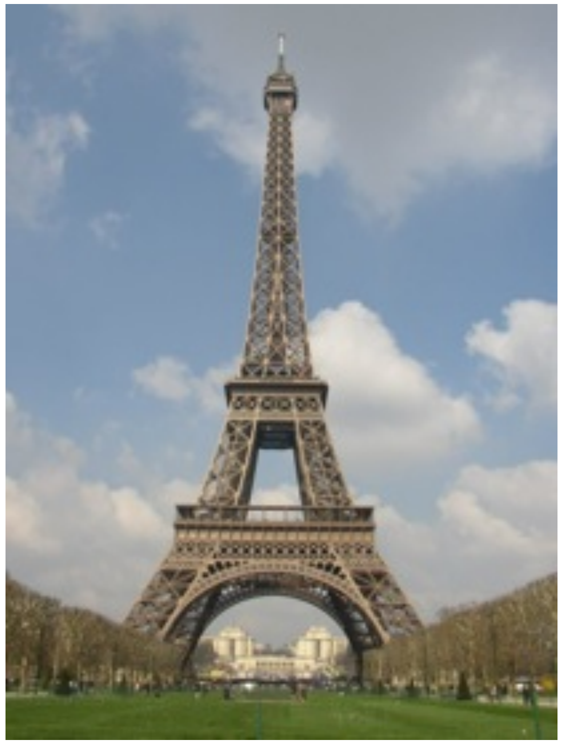
Similarity can occur in the form of shape, color, shading or other qualities.

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

Symmetry



http://themetapicture.com/the-average-woman-from-each-country/



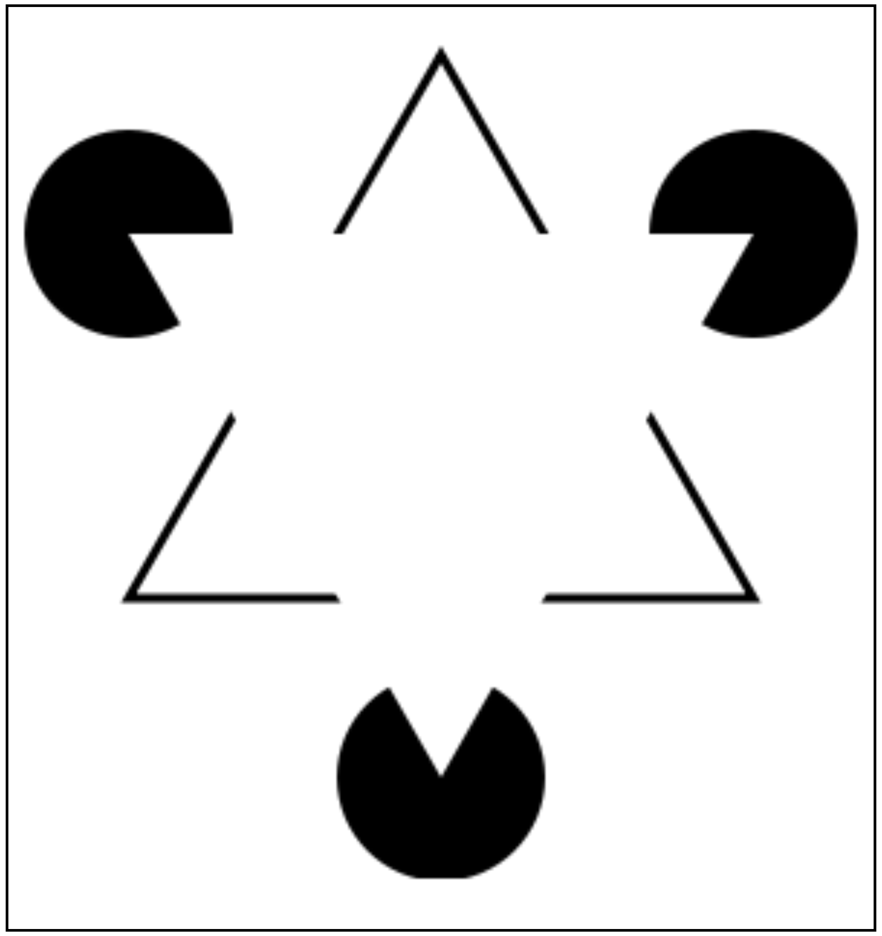
Common fate

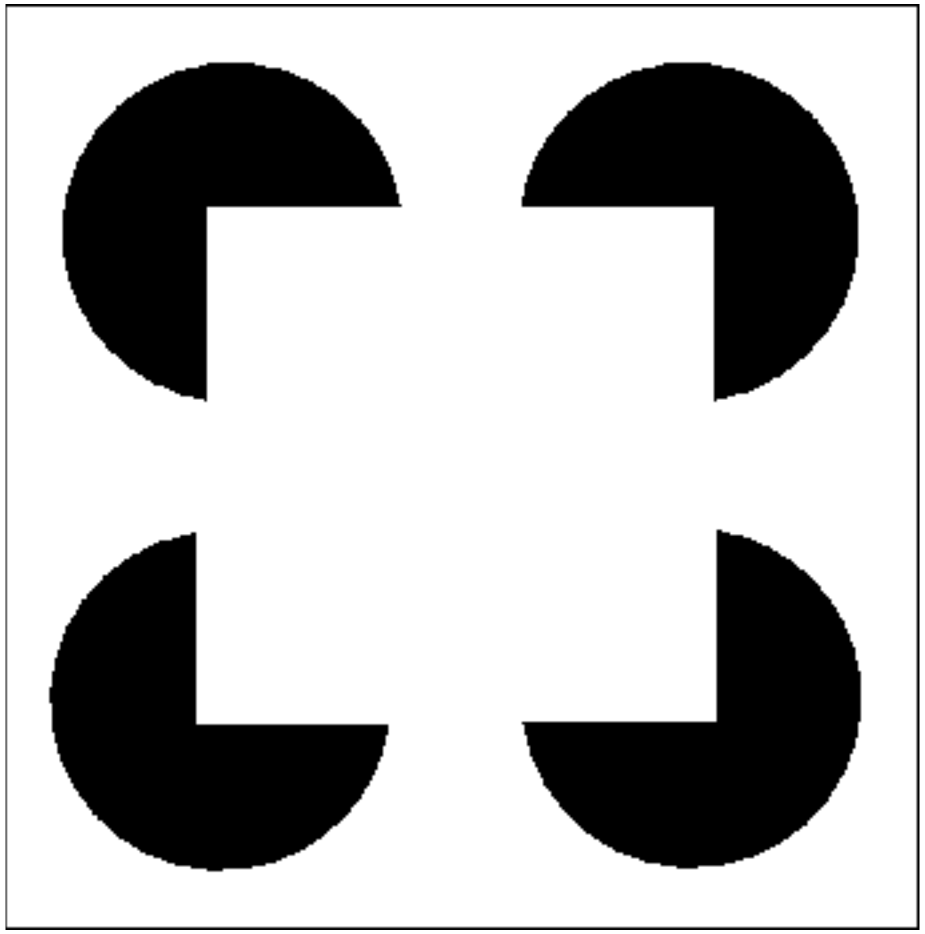
Image credit: Arthus-Bertrand (via F. Durand)

Proximity

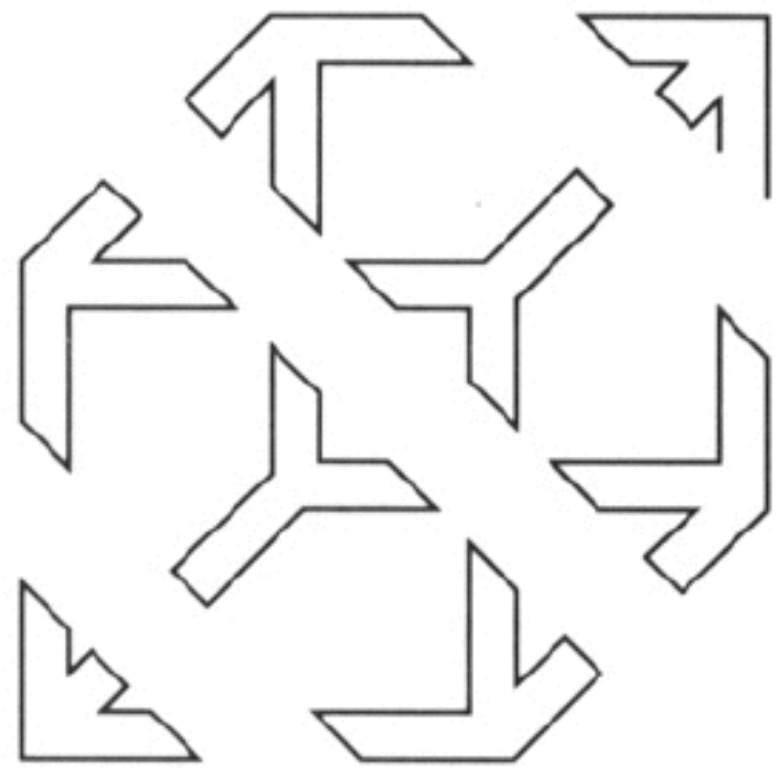
http://www.capital.edu/Resources/Images/outside6_035.jpg

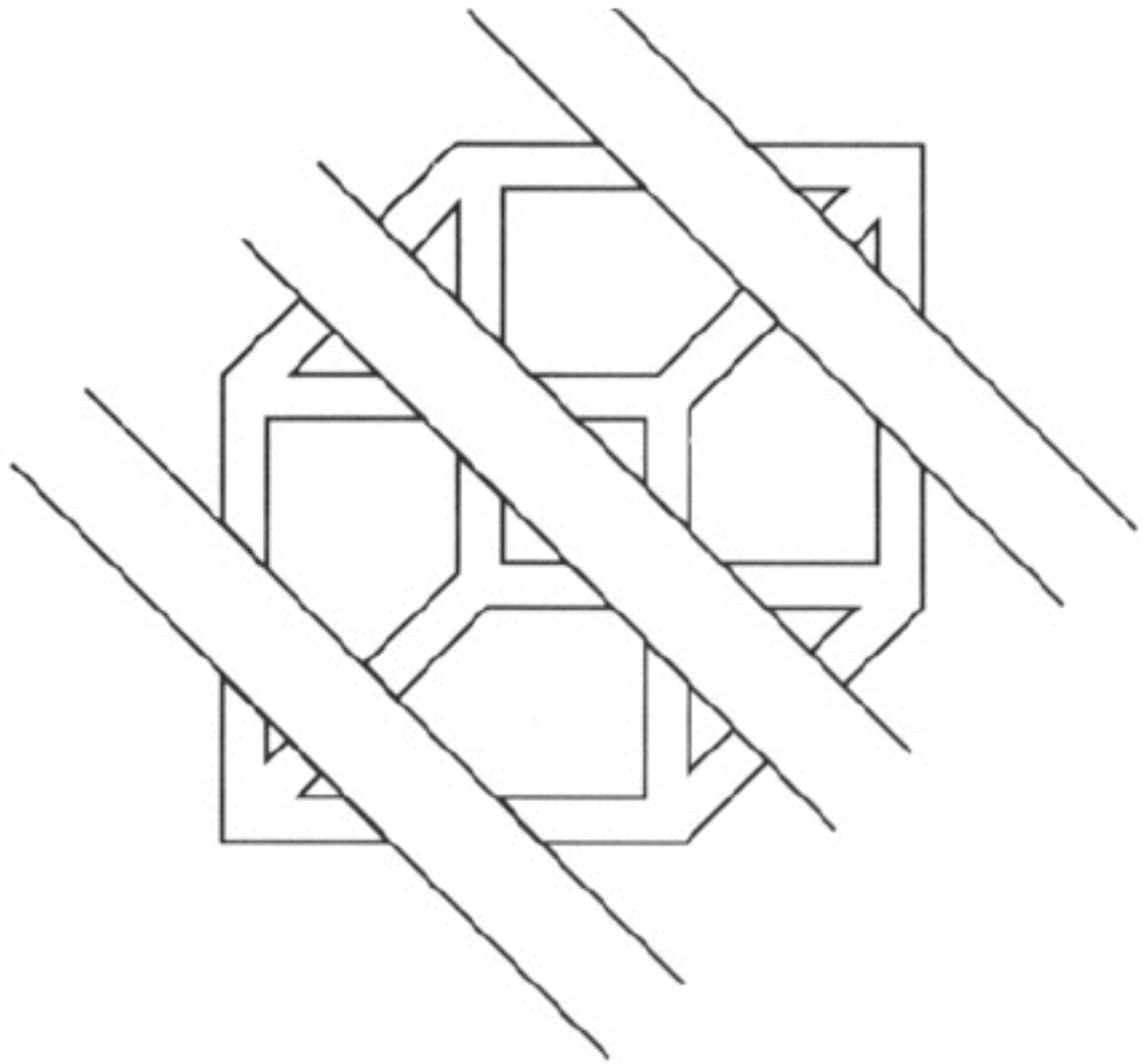
Illusory/subjective contours



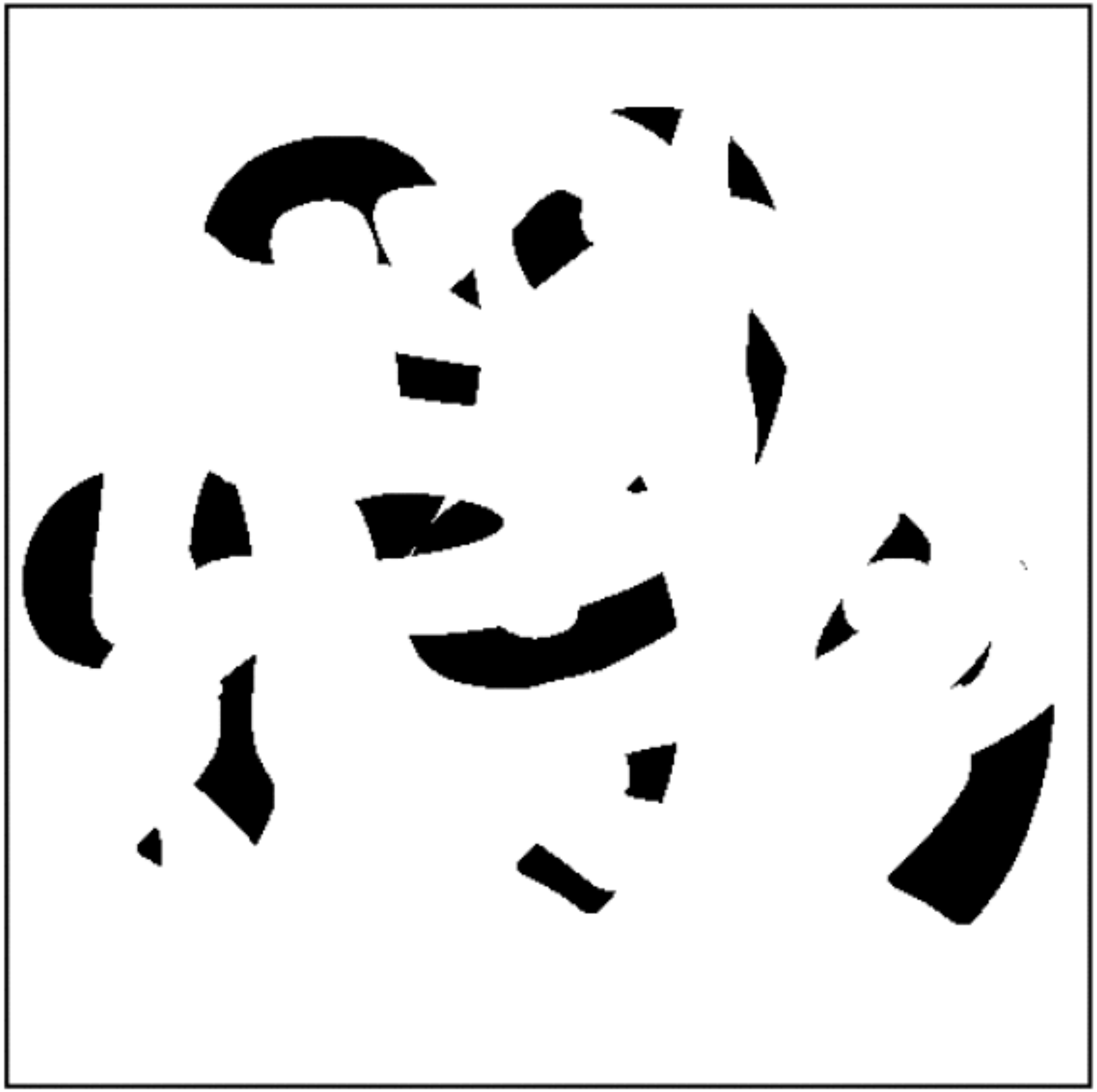


Interesting tendency to explain by occlusion

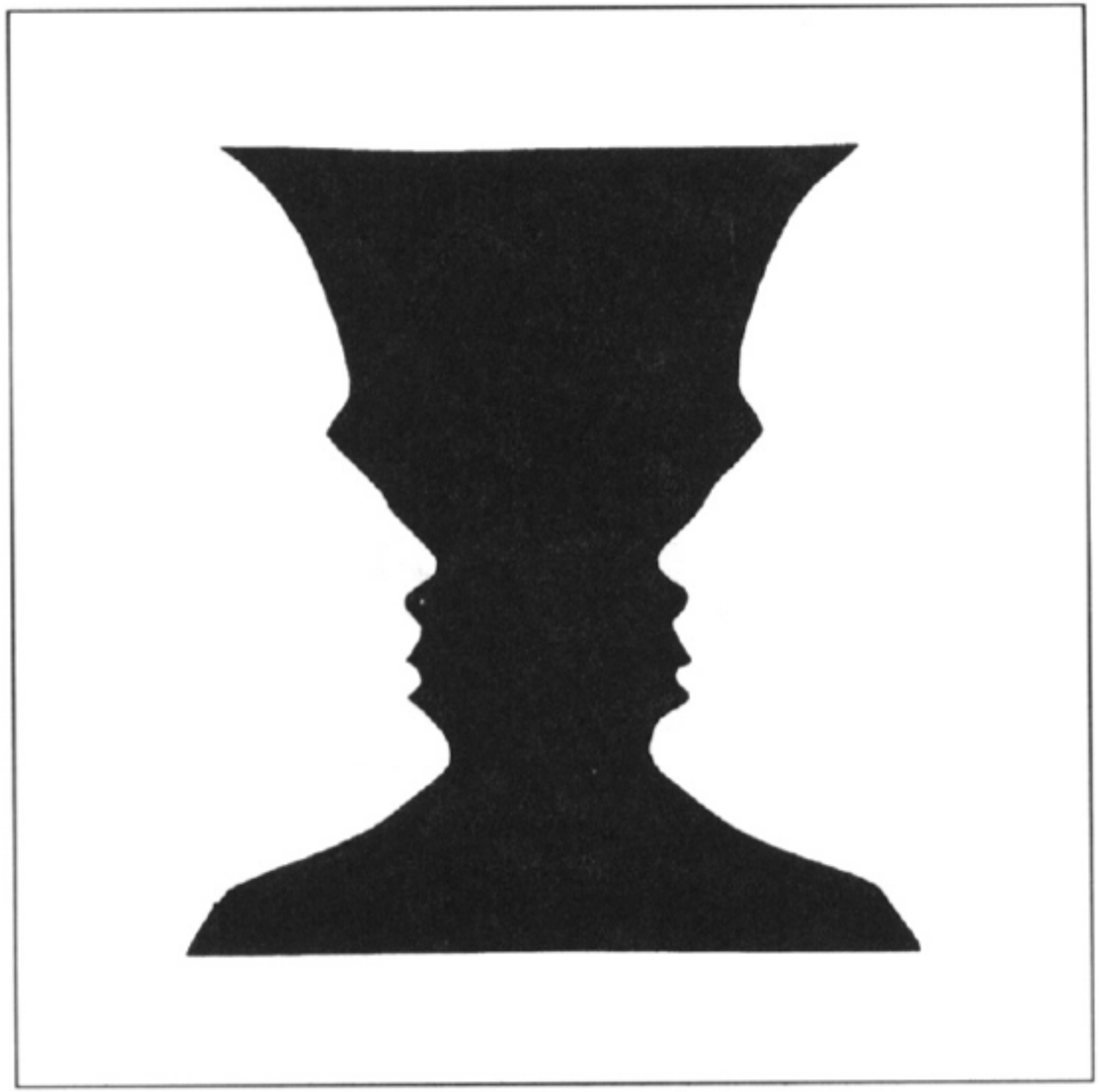




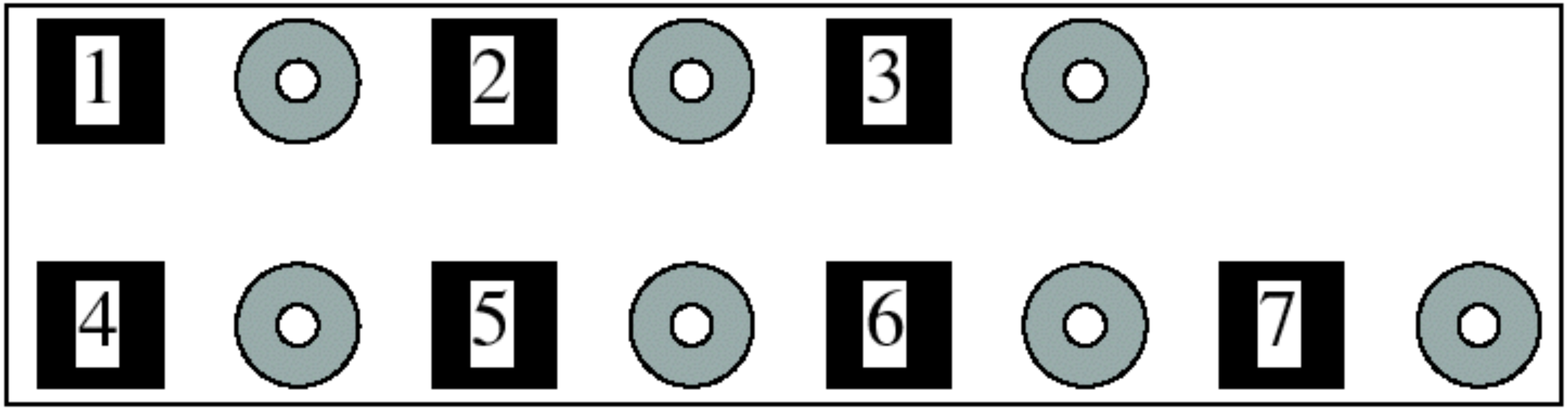
Continuity, explanation by occlusion



D. Forsyth

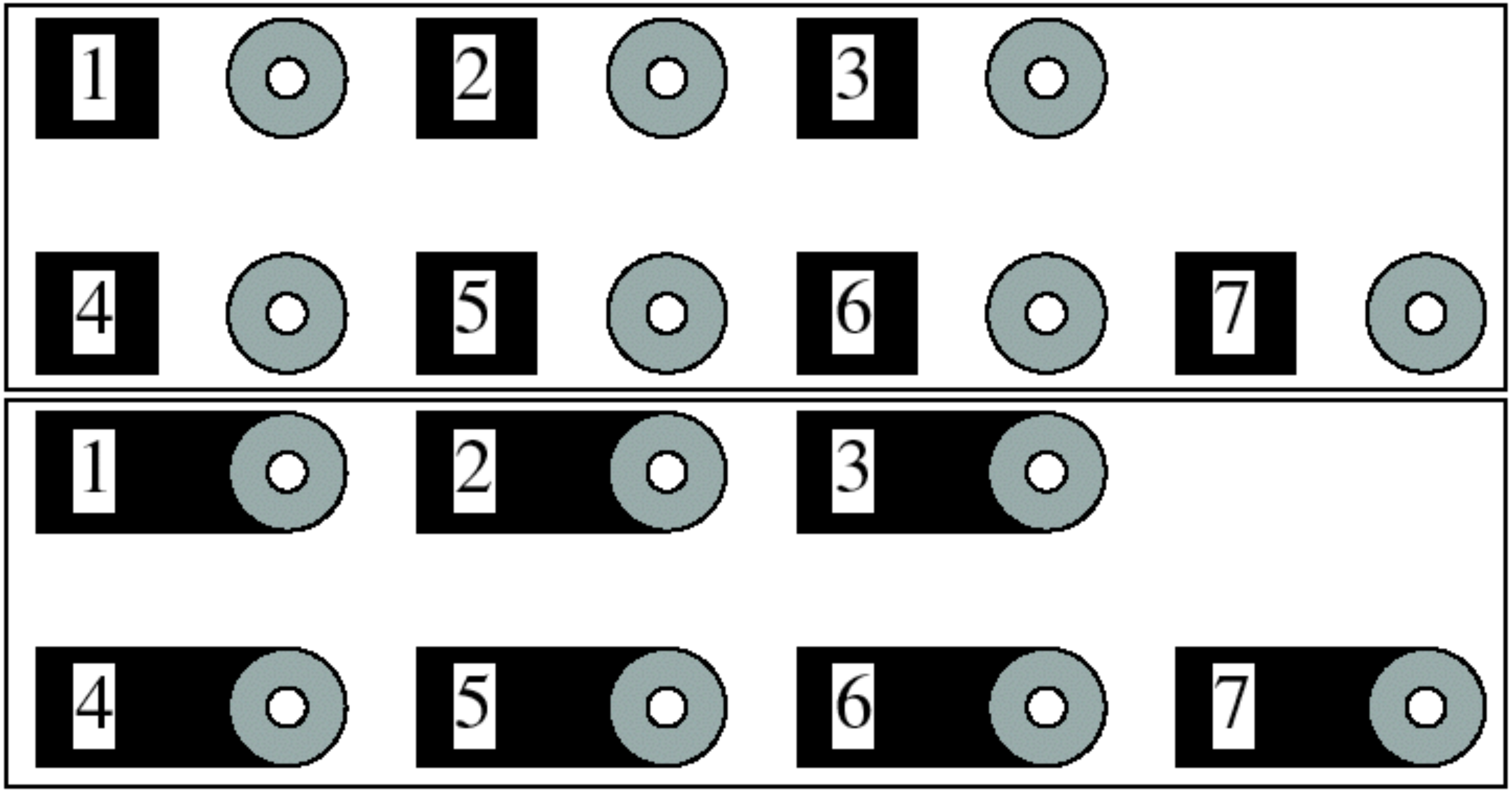


Grouping phenomena in real life



Forsyth & Ponce, Figure 14.7

Grouping phenomena in real life



Forsyth & Ponce, Figure 14.7

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)
- Inspiring observations/explanations; challenge remains how to best map to algorithms.

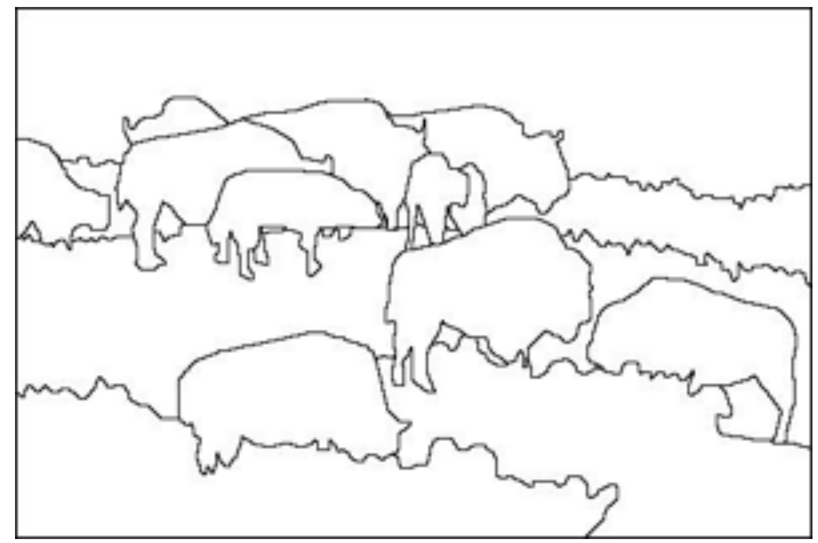
Outline

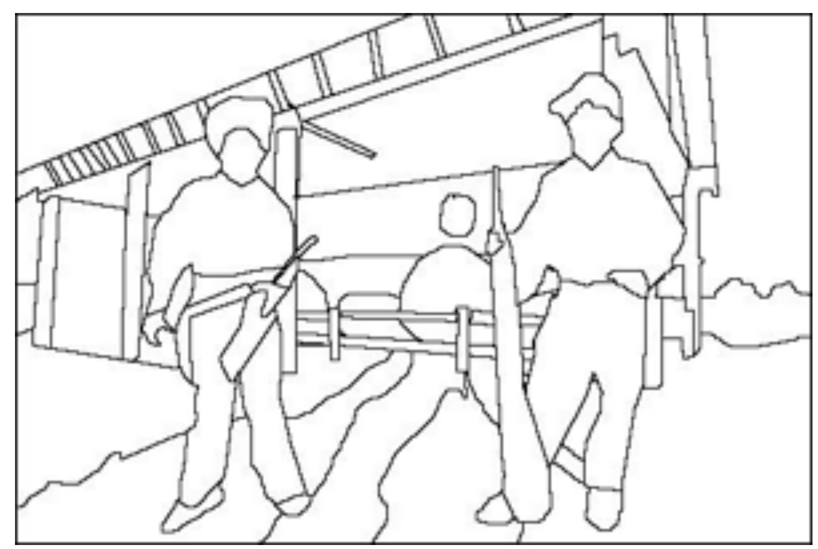
- What are grouping problems in vision?
- Inspiration from human perception
  - Gestalt properties
- Bottom-up segmentation via clustering
  - Algorithms:
    - Mode finding and mean shift: k-means, mean-shift
    - Graph-based: graph cuts, normalized cuts
  - Features: color, texture, …
    - Quantization for texture summaries

The goals of segmentation

Separate image into coherent “objects”

image | human segmentation





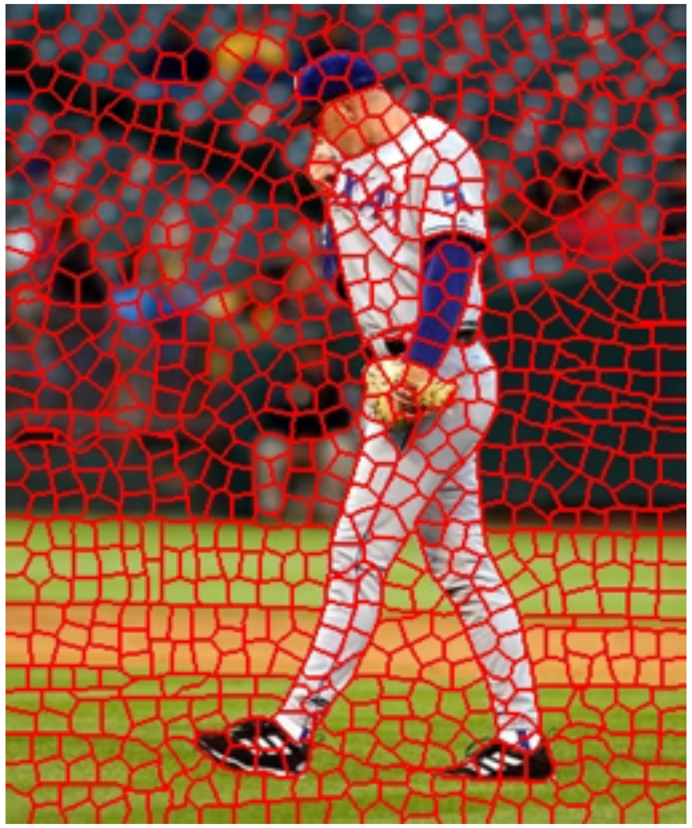
Source: Lana Lazebnik

The goals of segmentation

Separate image into coherent "objects"

Group together similar-looking pixels for efficiency of further processing

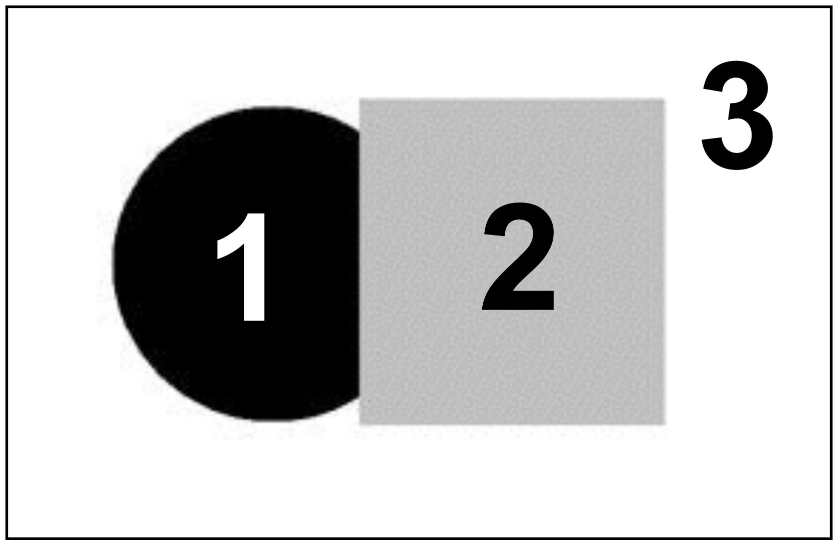
"superpixels"



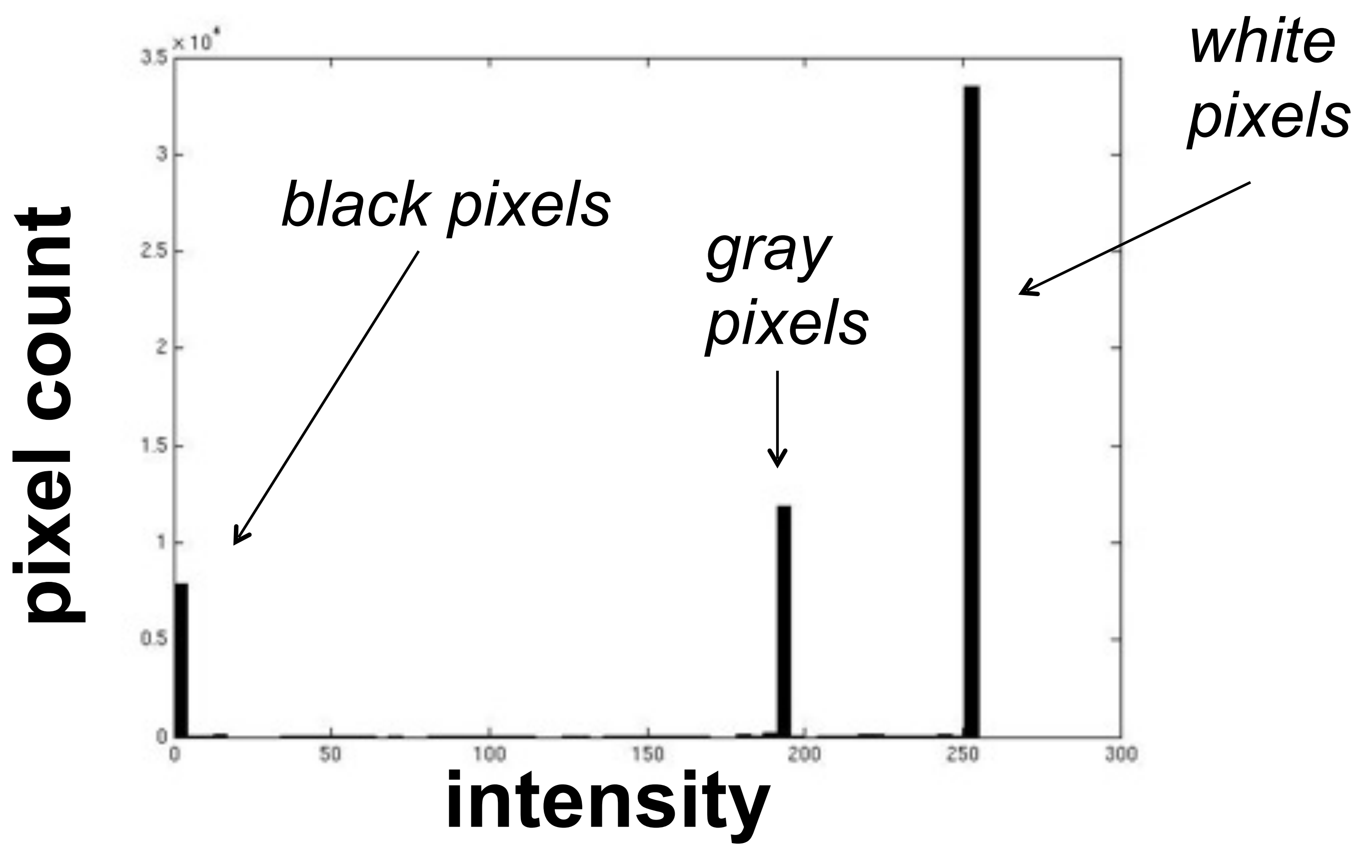
X. Ren and J. Malik. [Learning a classification model for segmentation.]() ICCV 2003.

Source: Lana Lazebnik

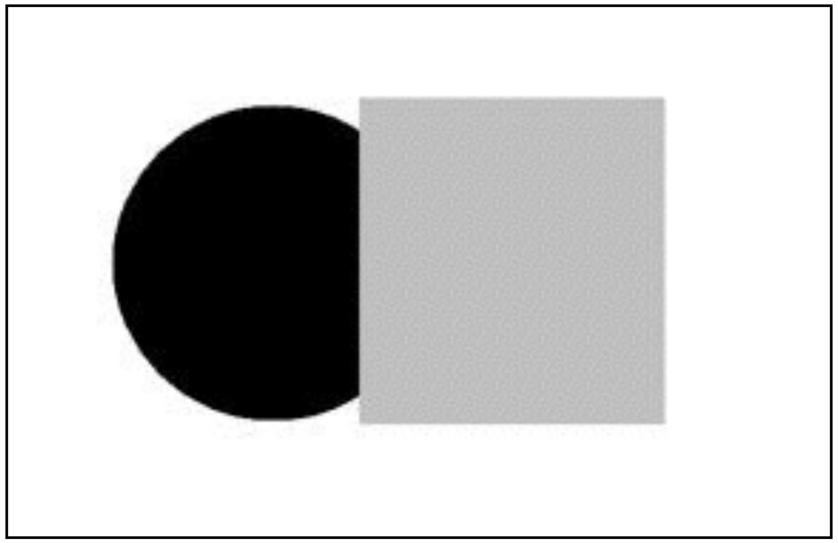
Image segmentation: toy example



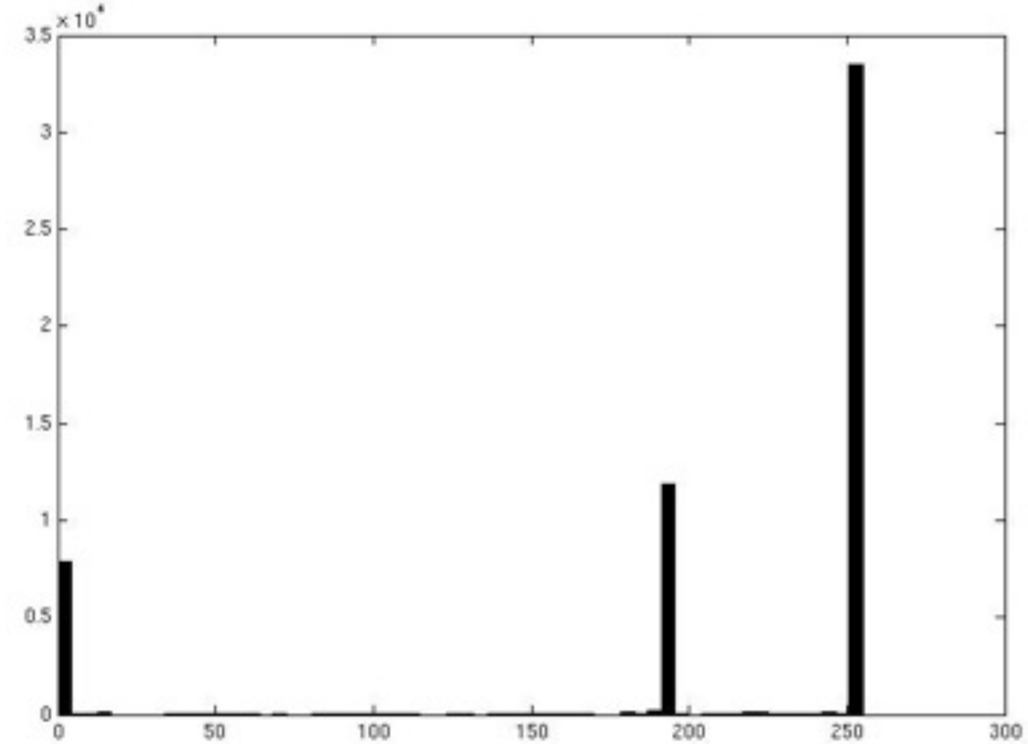
input image



- 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?

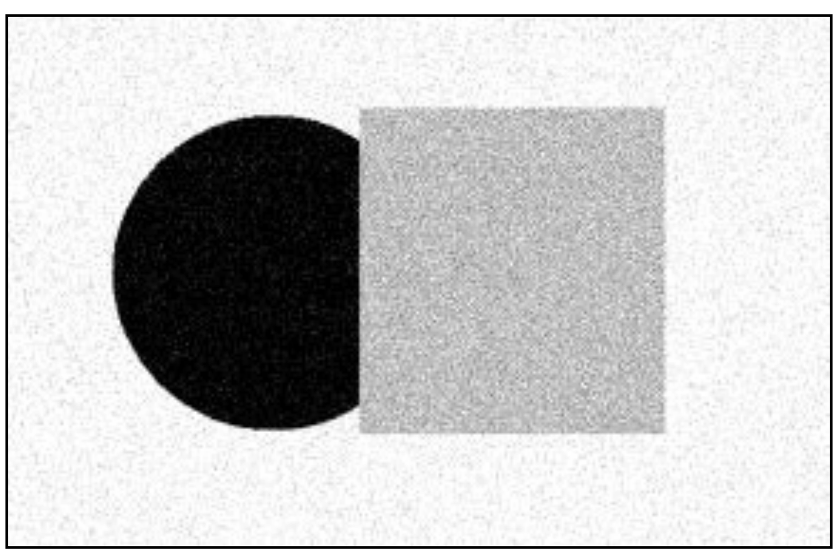


input image

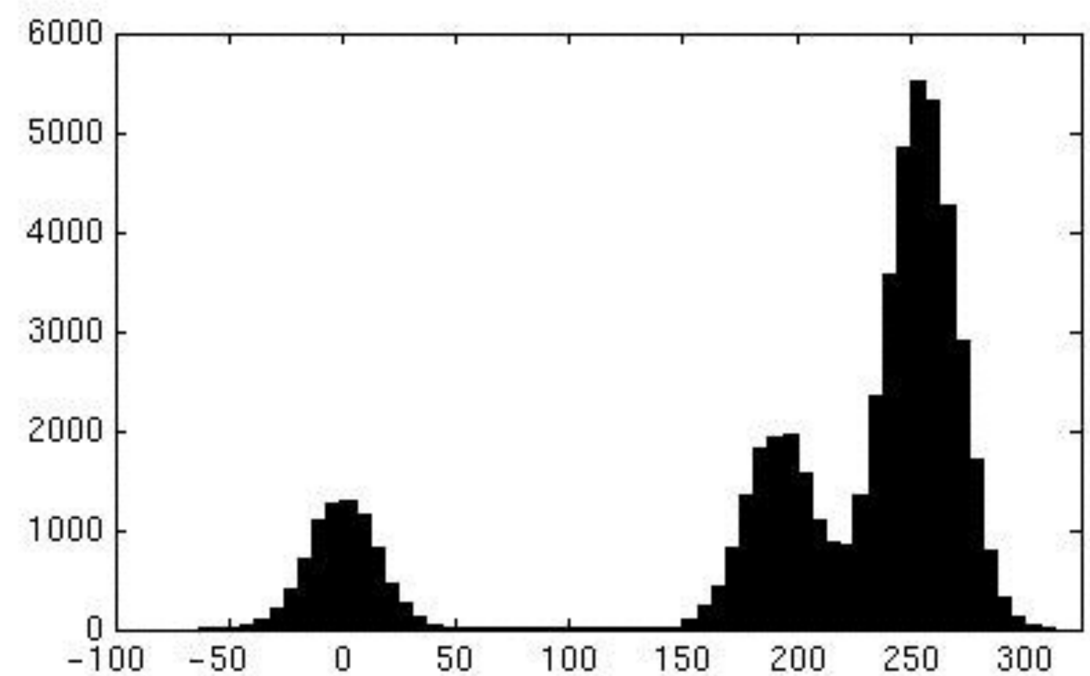


pixel count

intensity

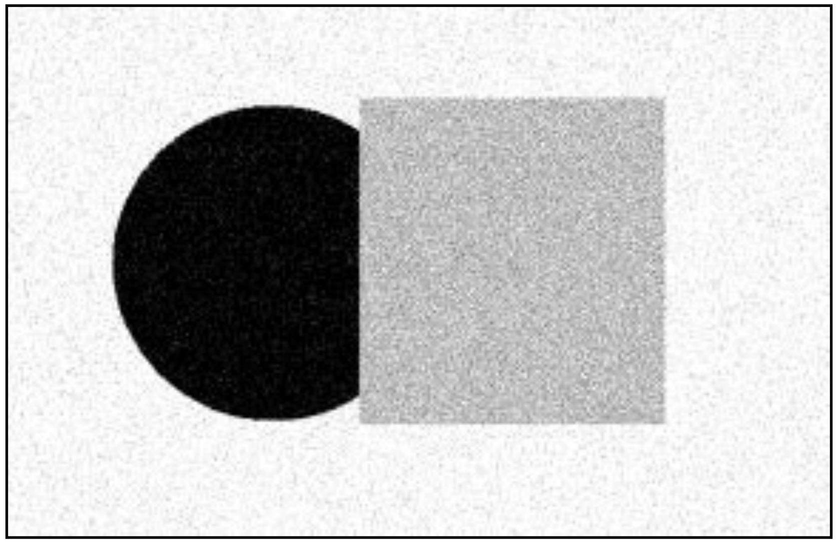


input image

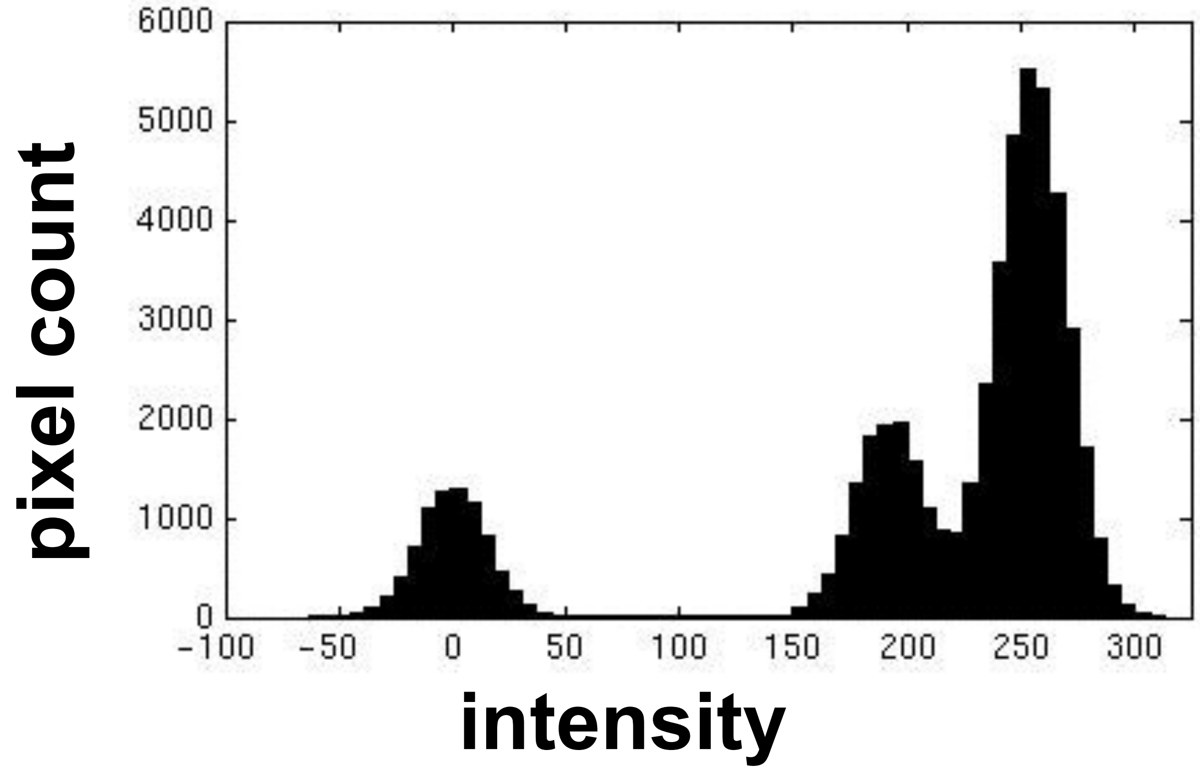


pixel count

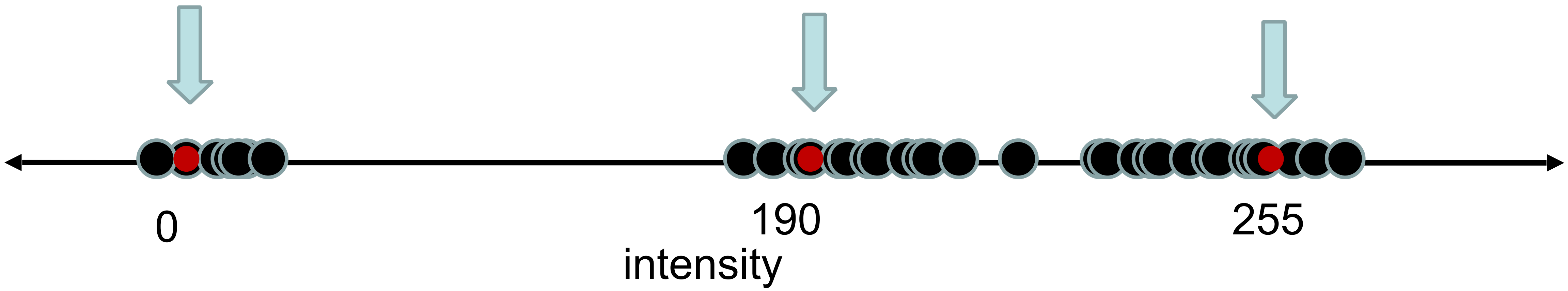
intensity

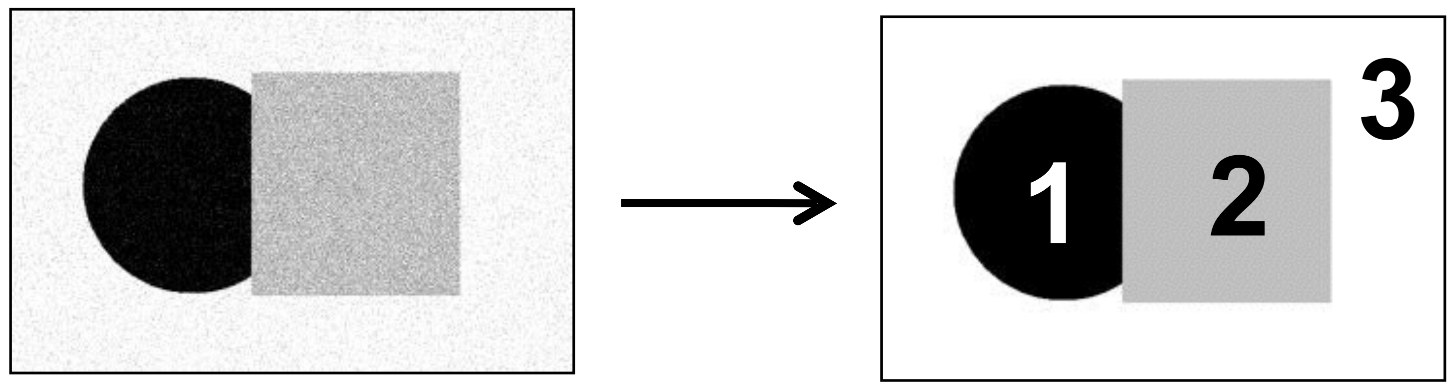


input image



- 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:

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

recall *k*-means

Segmentation as clustering

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

Grouping pixels based on **intensity** similarity

Feature space: intensity value (1-d)

K=2

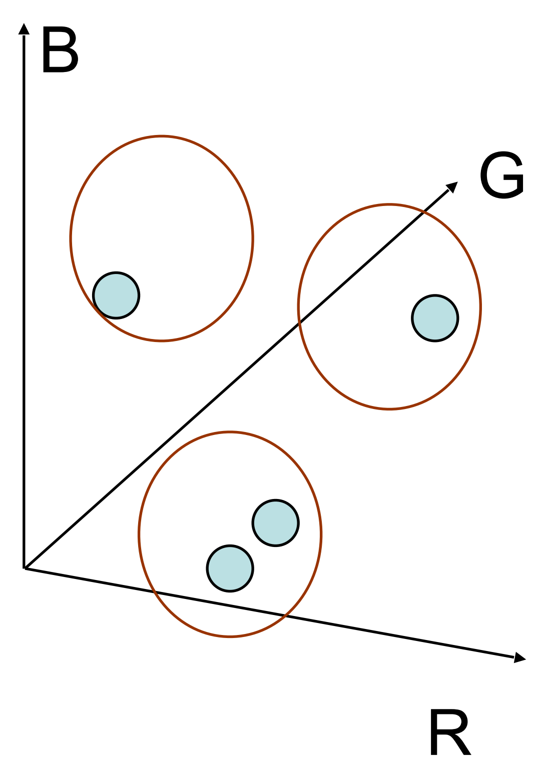
K=3

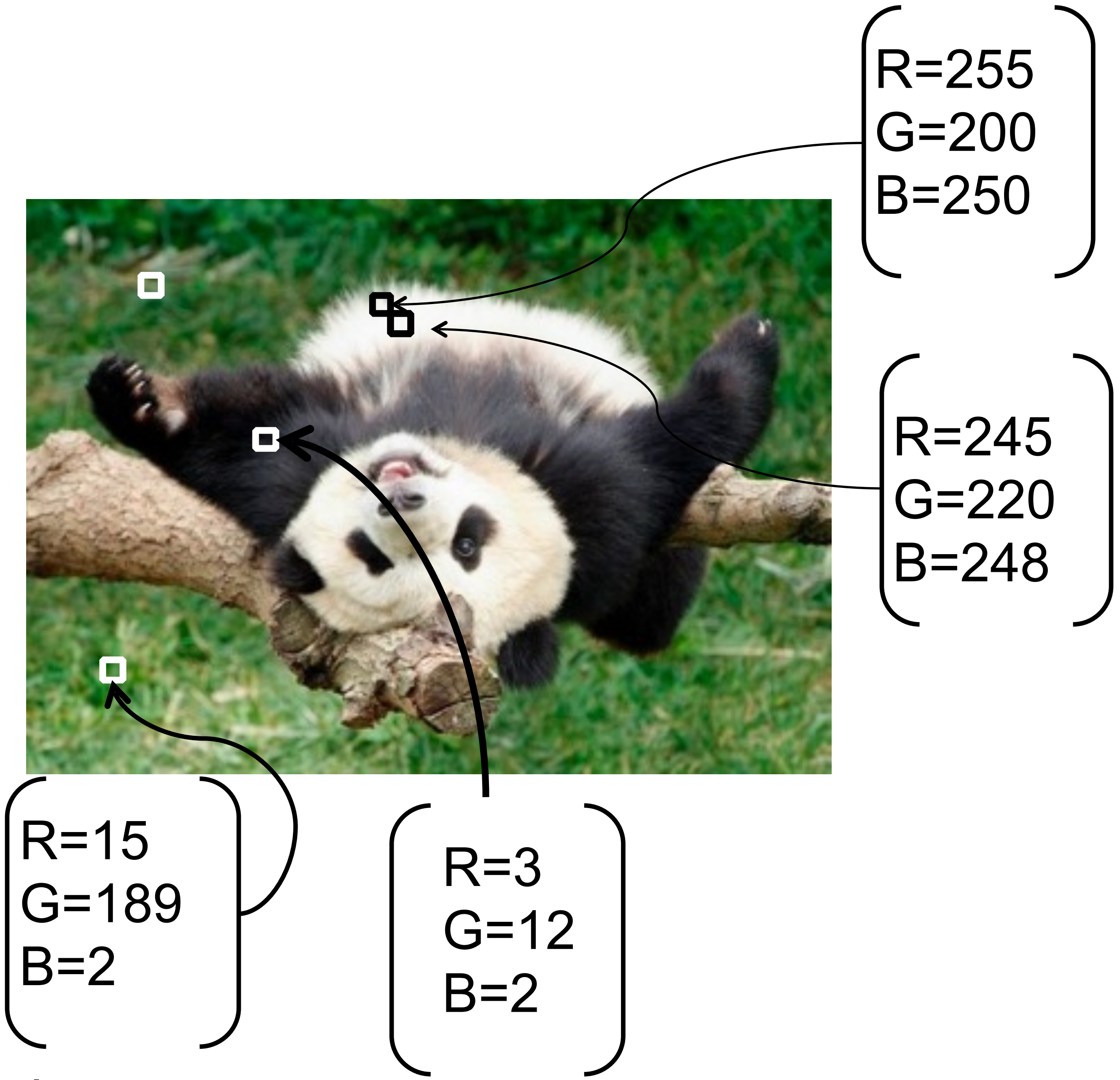
quantization of the feature space;
segmentation label map

Segmentation as clustering

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

Grouping pixels based on **color** similarity





Feature space: color value (3-d)

Segmentation as clustering

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

Grouping pixels based on **intensity** similarity

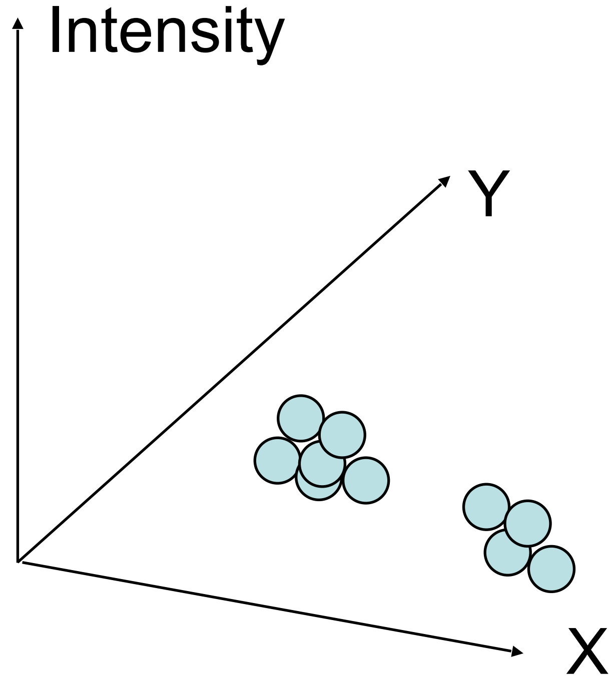
Clusters based on intensity similarity don't have to be spatially coherent.

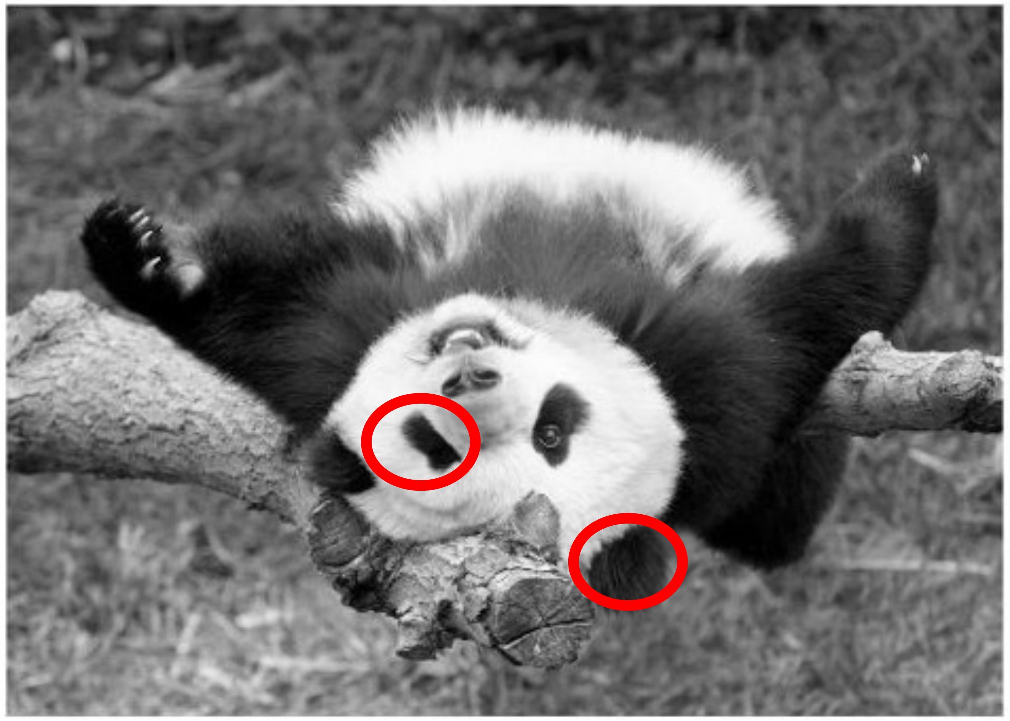
Kristen Grauman

Segmentation as clustering

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

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





Both regions are black, but if we also include **position (x,y),** then we could group the two into distinct segments; way to encode both similarity & proximity.

Kristen Grauman

Segmentation as clustering

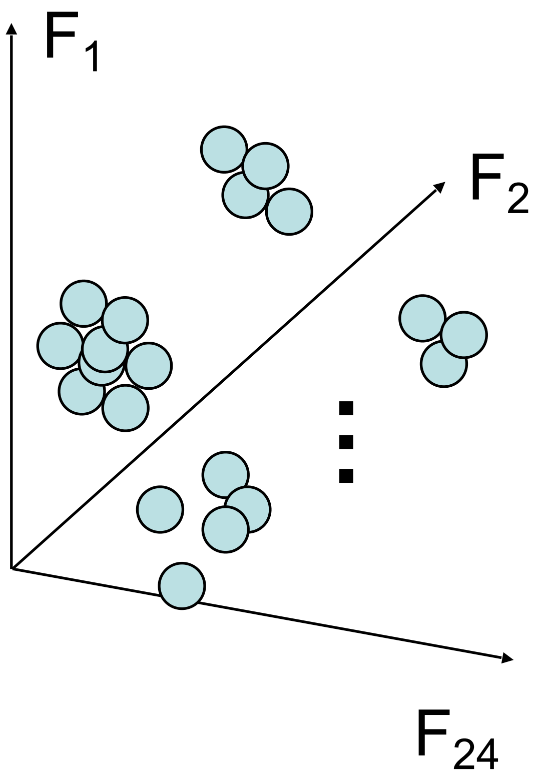
- Color, brightness, position alone are not enough to distinguish all regions…

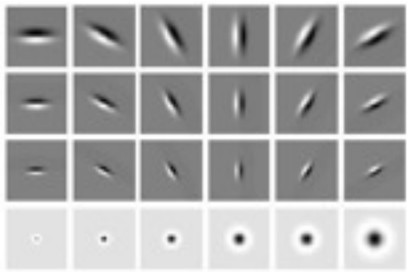


Segmentation as clustering

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

Grouping pixels based on **texture** similarity

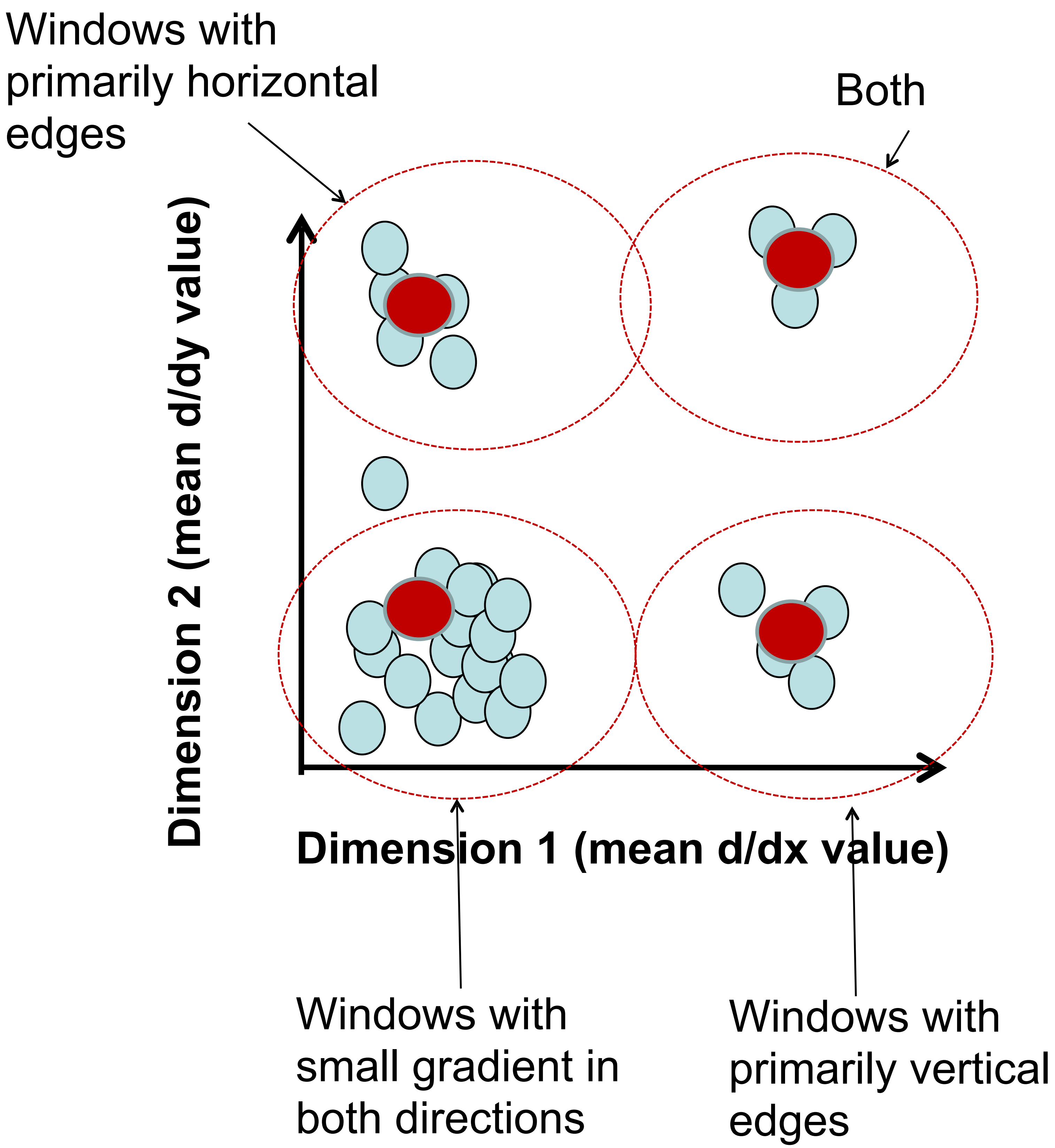


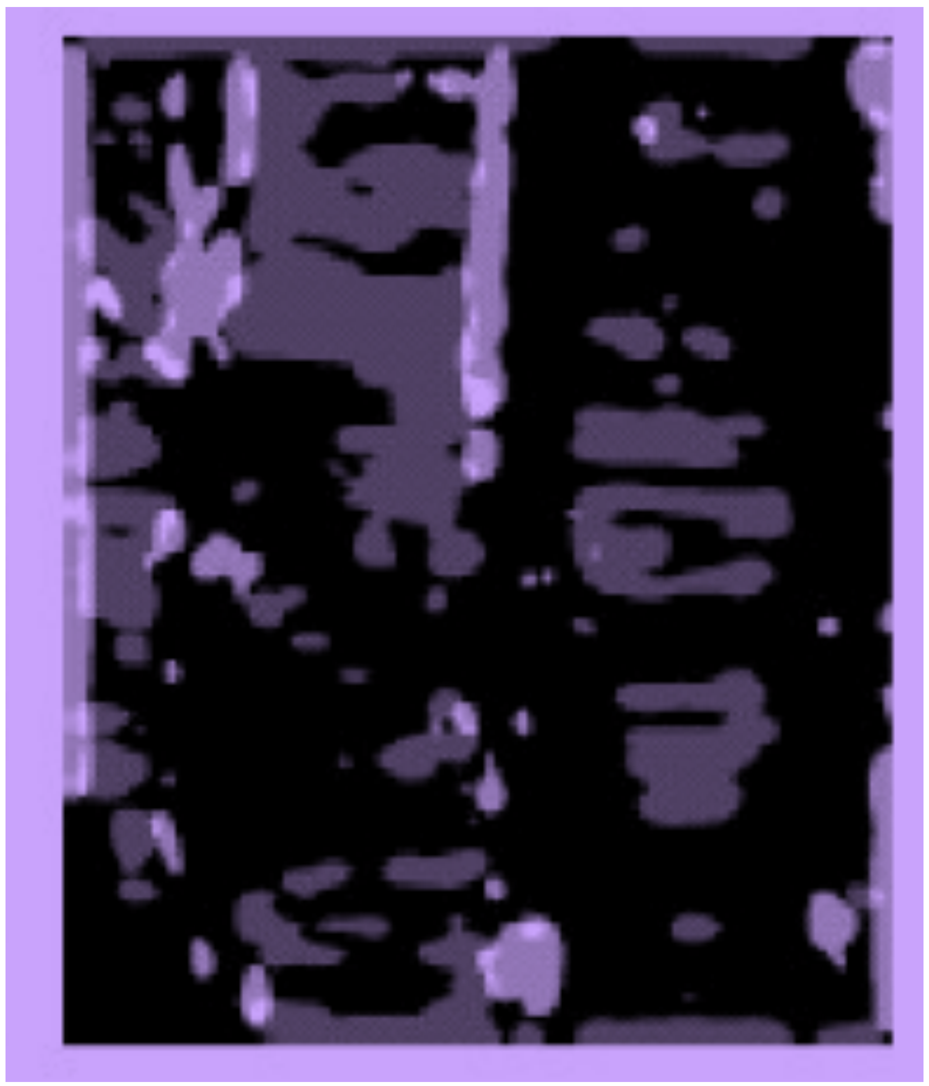


Filter bank of 24 filters

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

Recall: texture representation example





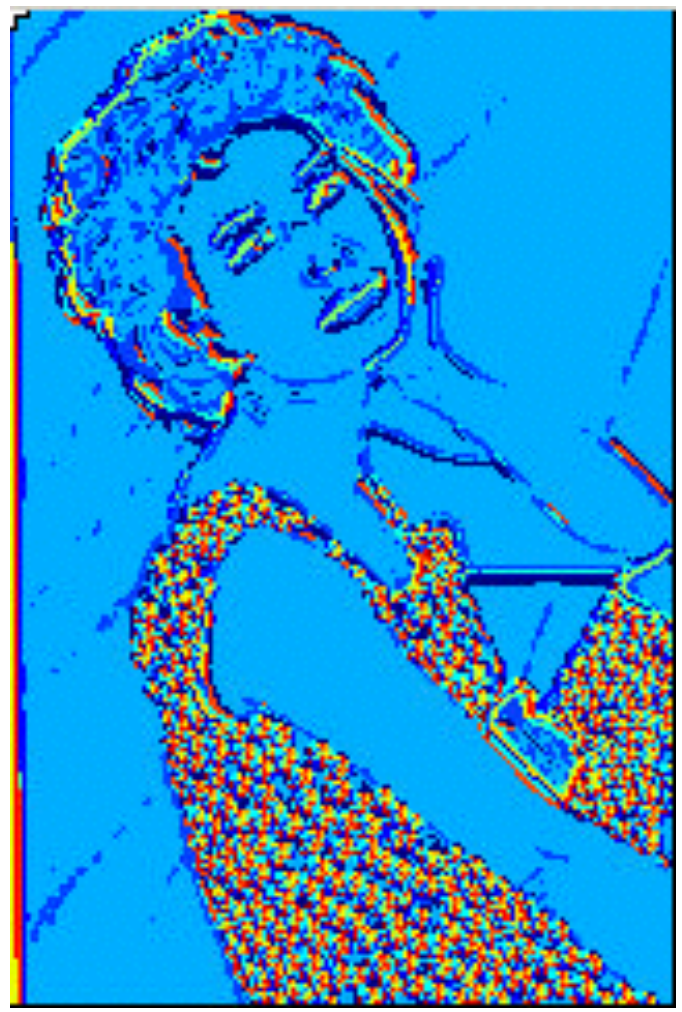
statistics to summarize patterns in small windows

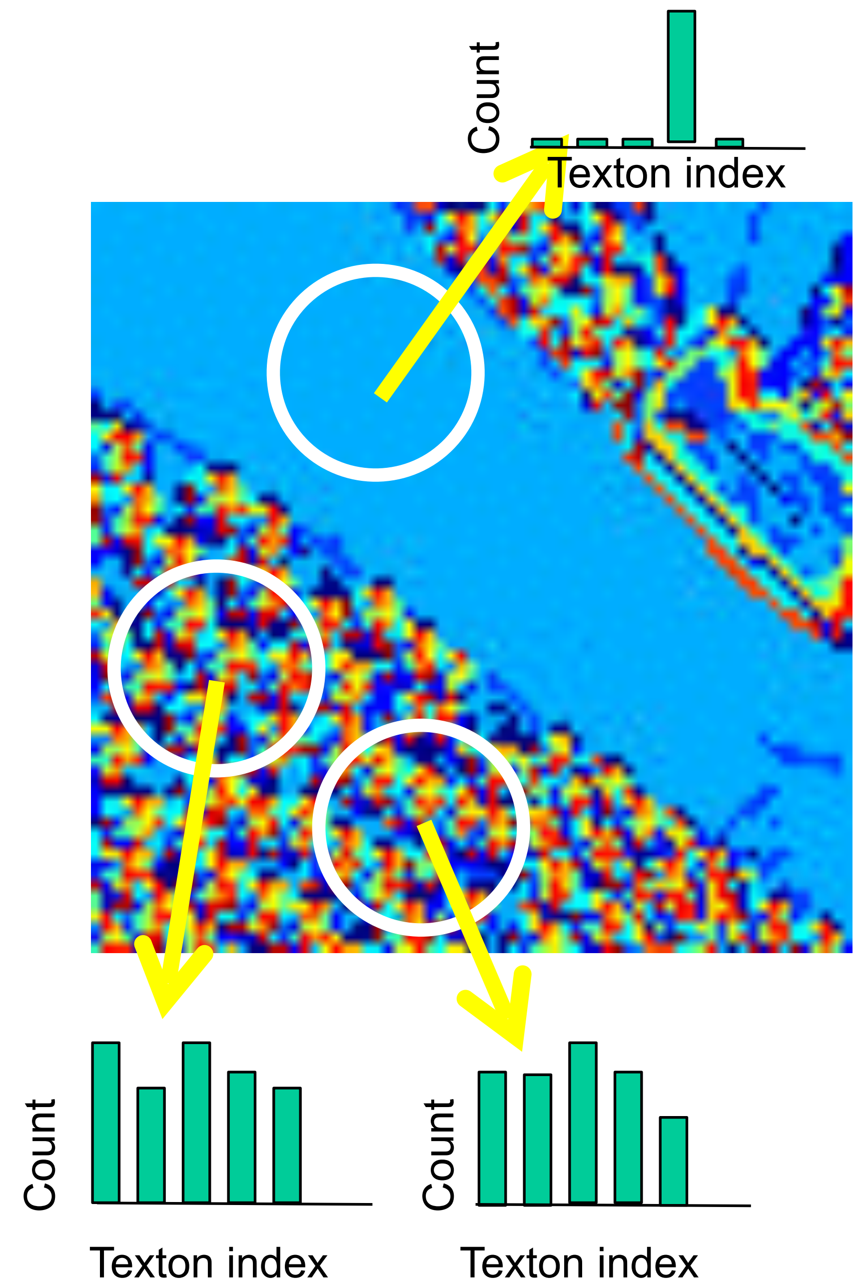
Segmentation with texture features

- Find "textons" by **clustering** vectors of filter bank outputs
- Describe texture in a window based on ***texton histogram***

Image

Texton map

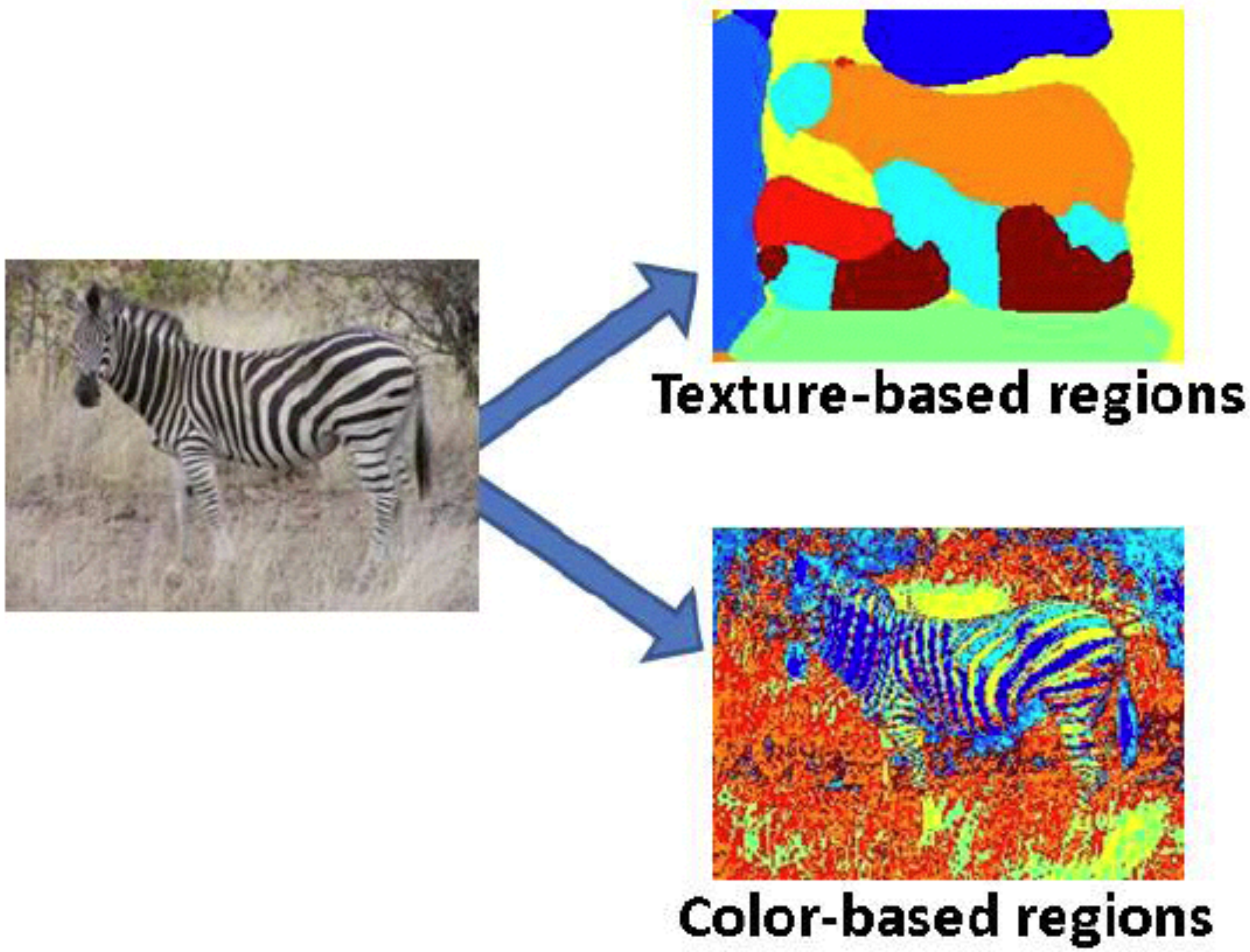




Malik, Belongie, Leung and Shi. IJCV 2001.

Adapted from Lana Lazebnik

Image segmentation example



Outline

- What are grouping problems in vision?

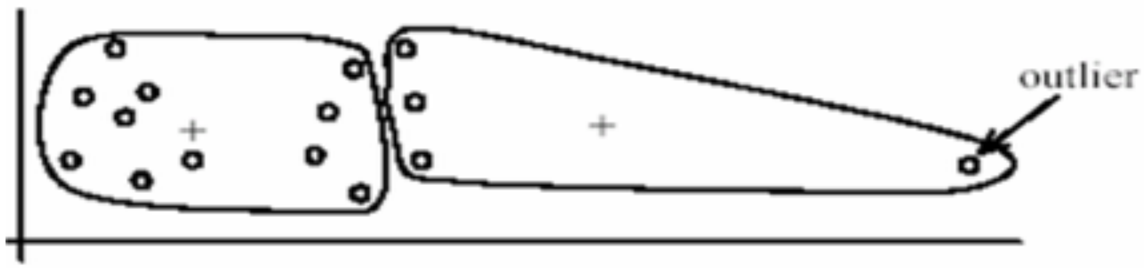
- Inspiration from human perception
  - Gestalt properties

- Bottom-up segmentation via clustering
  - Algorithms:
    - Mode finding and mean shift: k-means, mean-shift
    - Graph-based: graph cuts, normalized cuts
  - Features: color, texture, …
    - Quantization for texture summaries

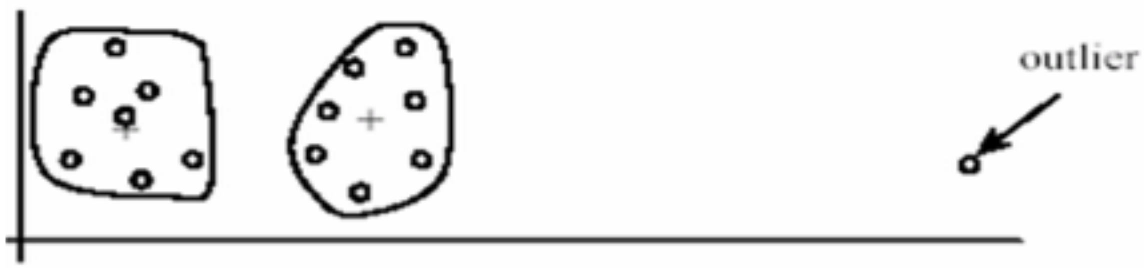
K-means: pros and cons

Pros

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



(A): Undesirable clusters

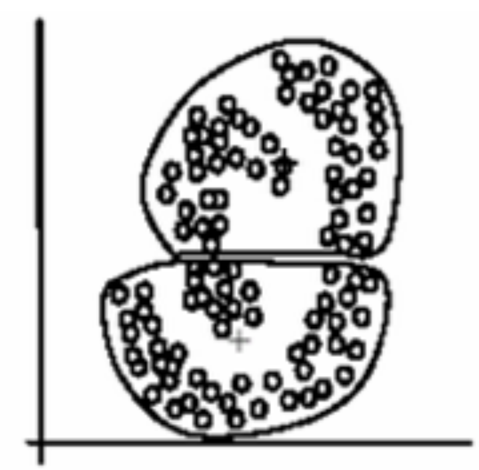


(B): Ideal clusters

Cons/issues

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

(A): Two natural clusters



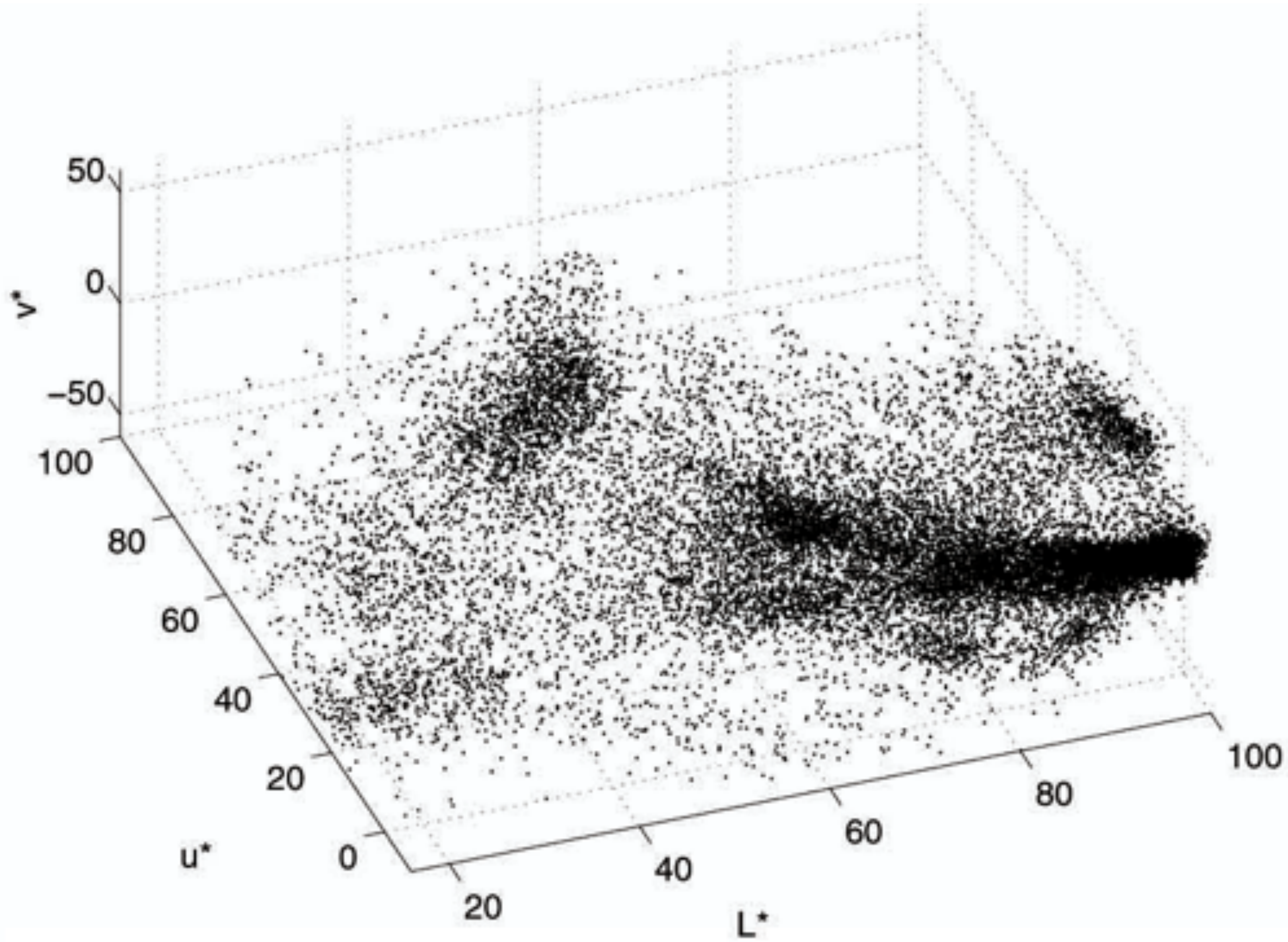
(B): *k*-means clusters

Mean shift algorithm

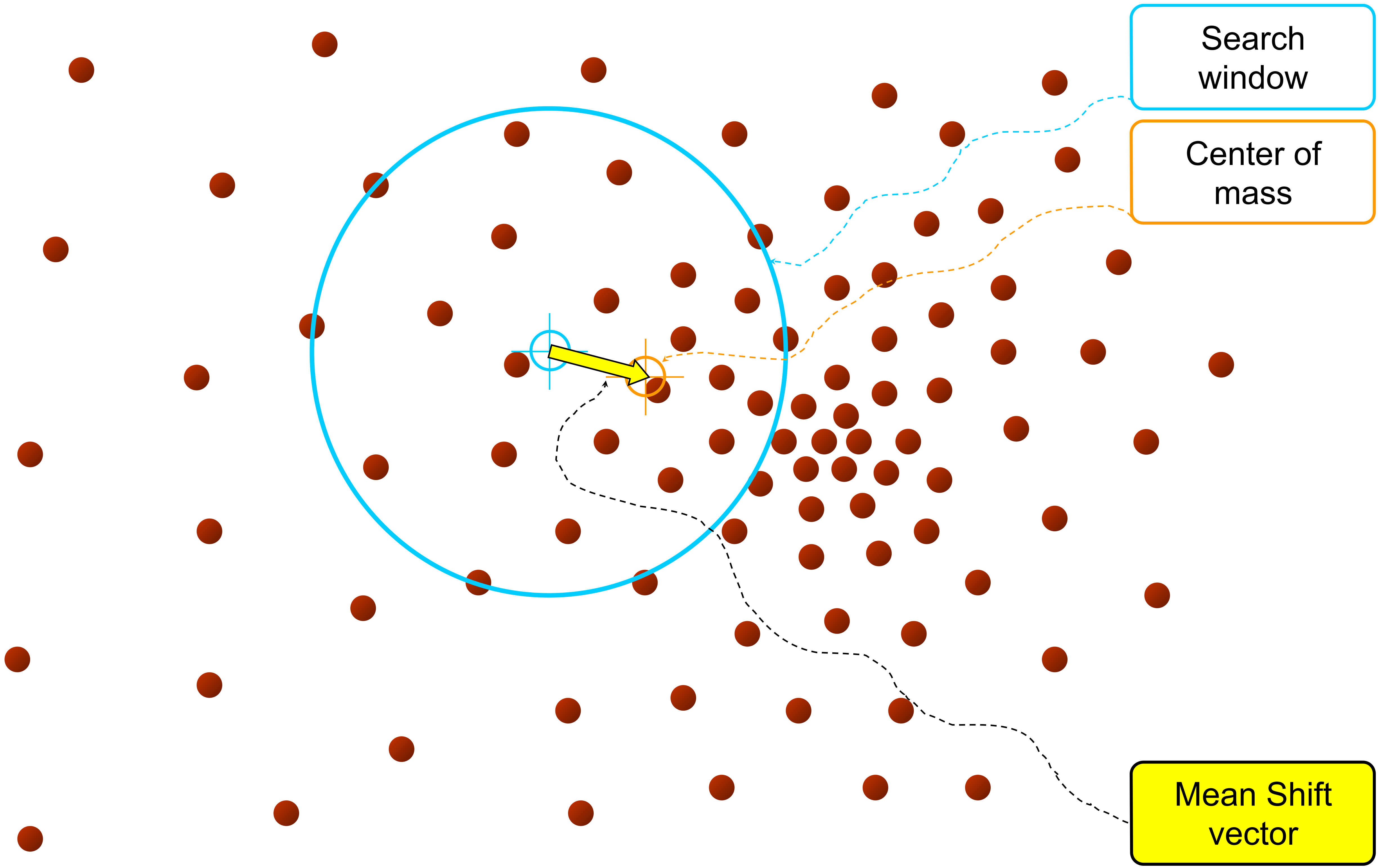
- The mean shift algorithm seeks *modes* or local maxima of density in the feature space

image

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

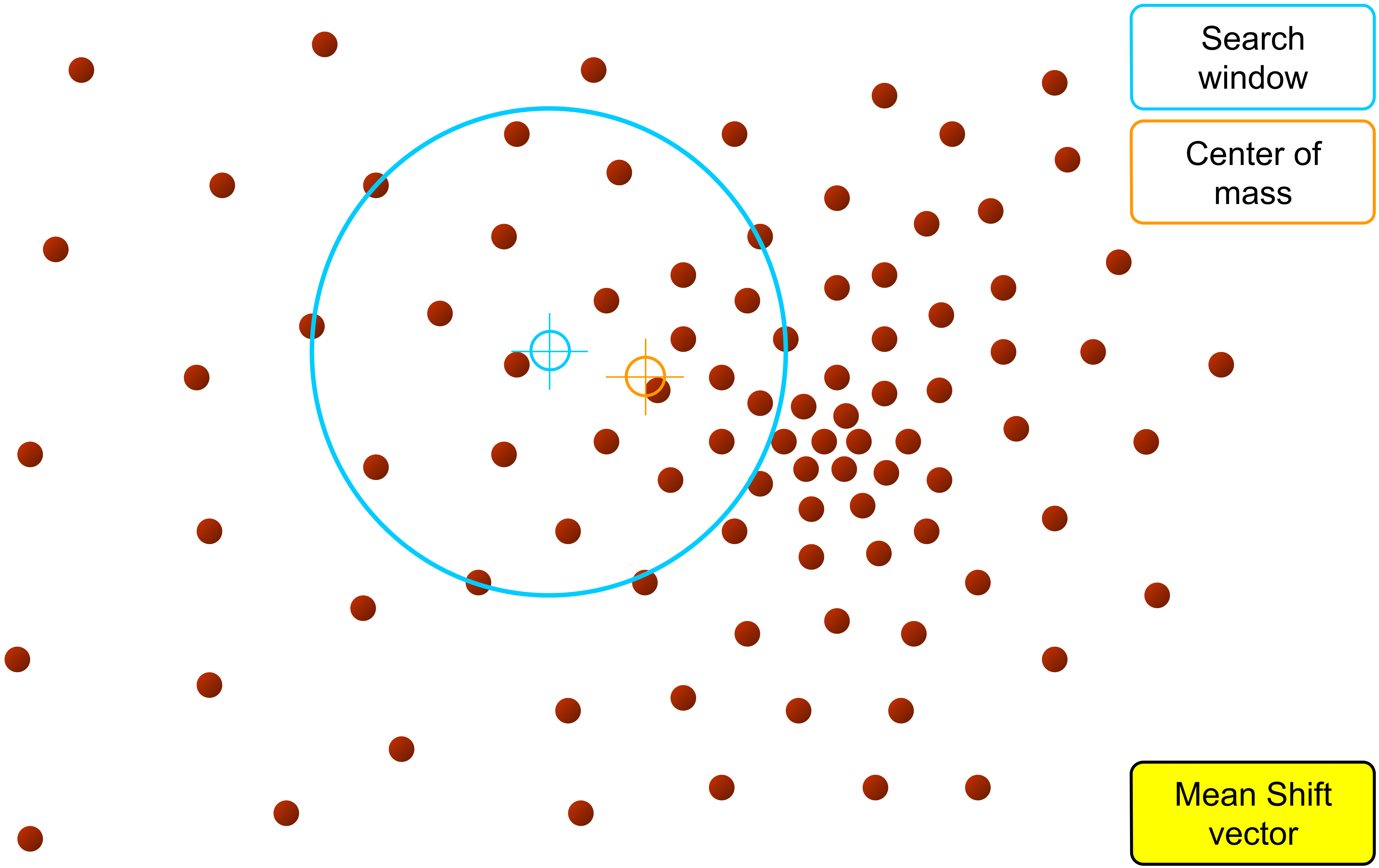


Mean shift



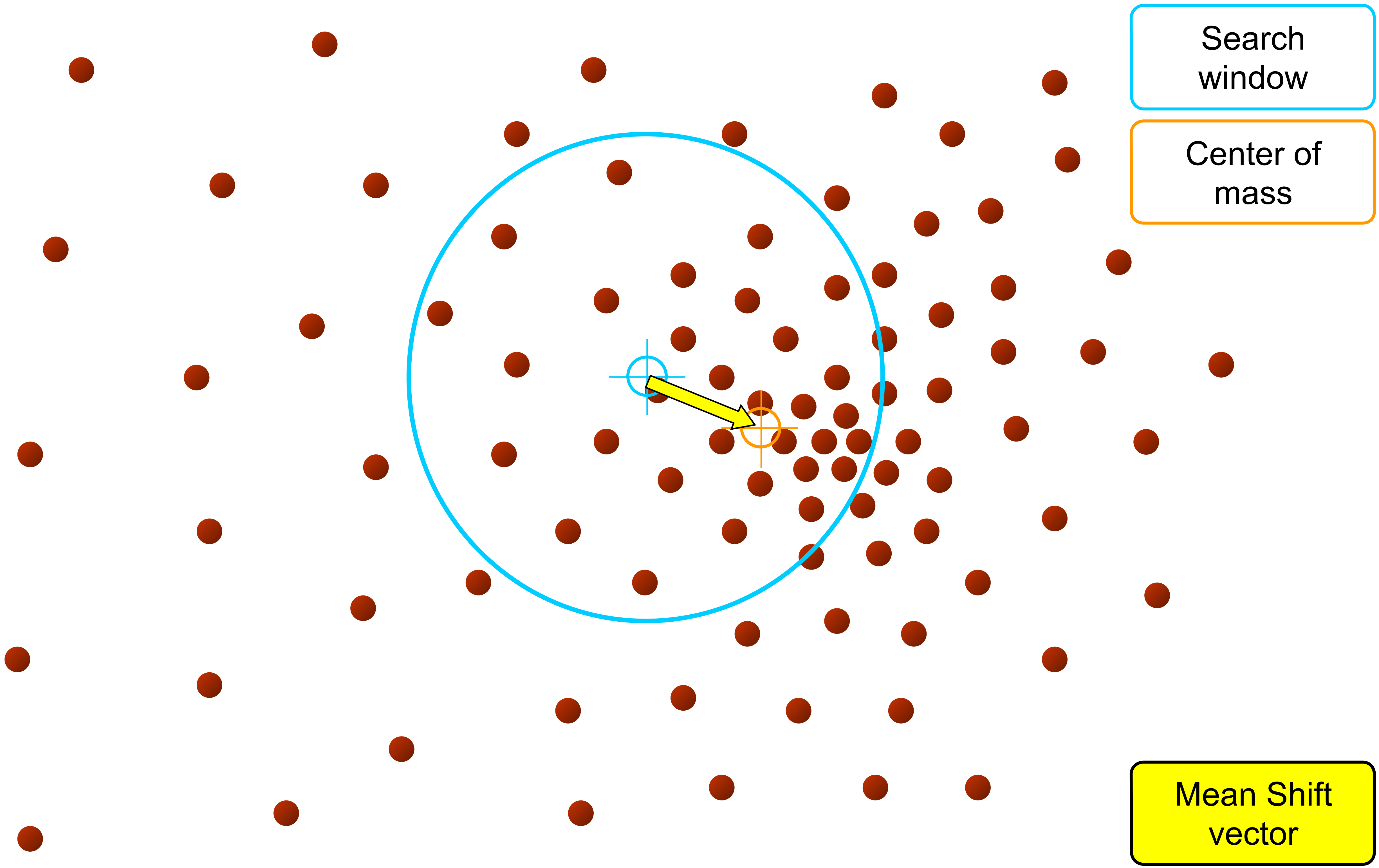
Slide by Y. Ukrainitz & B. Sarel

Mean shift



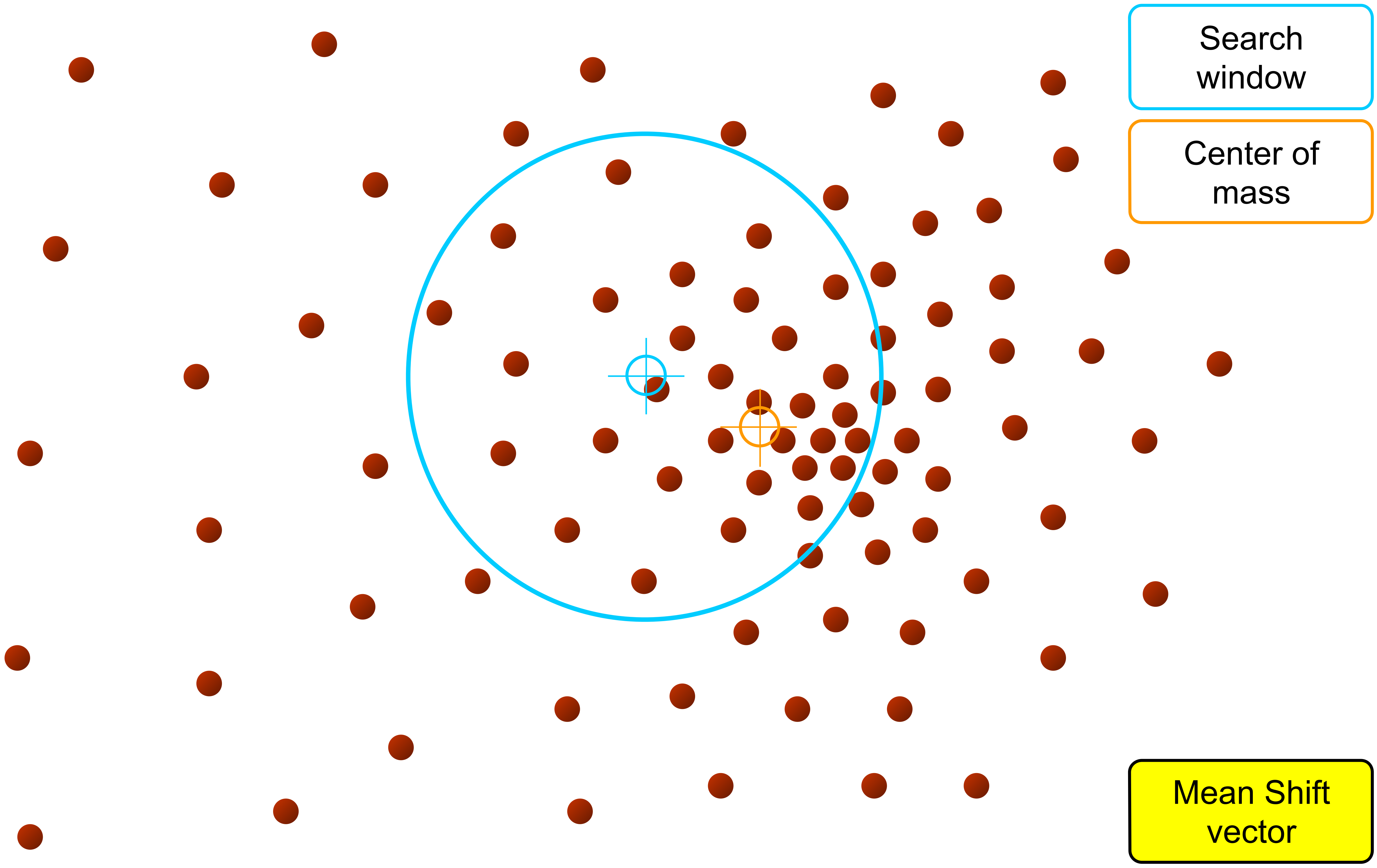
Slide by Y. Ukrainitz & B. Sarel

Mean shift



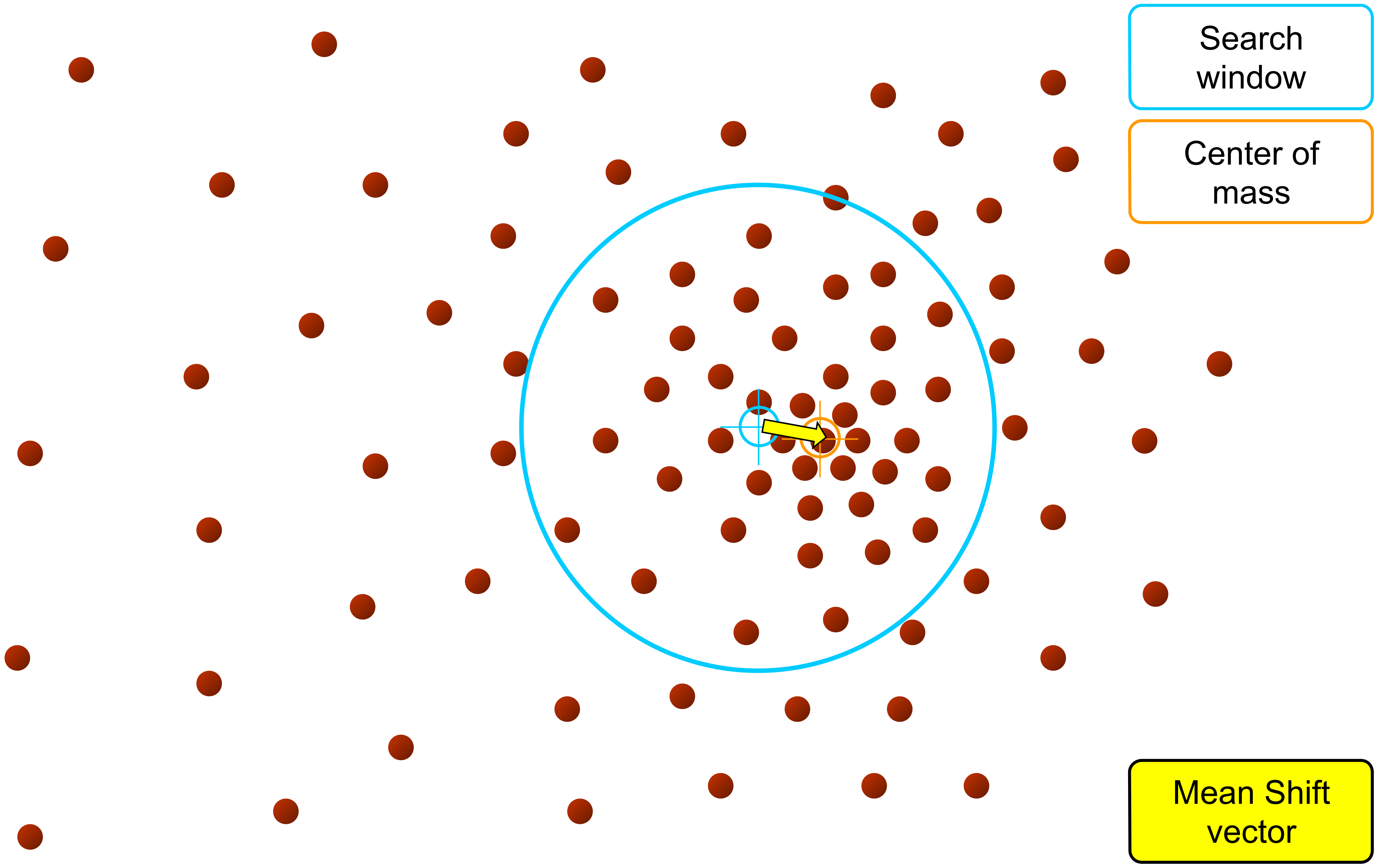
Slide by Y. Ukrainitz & B. Sarel

Mean shift



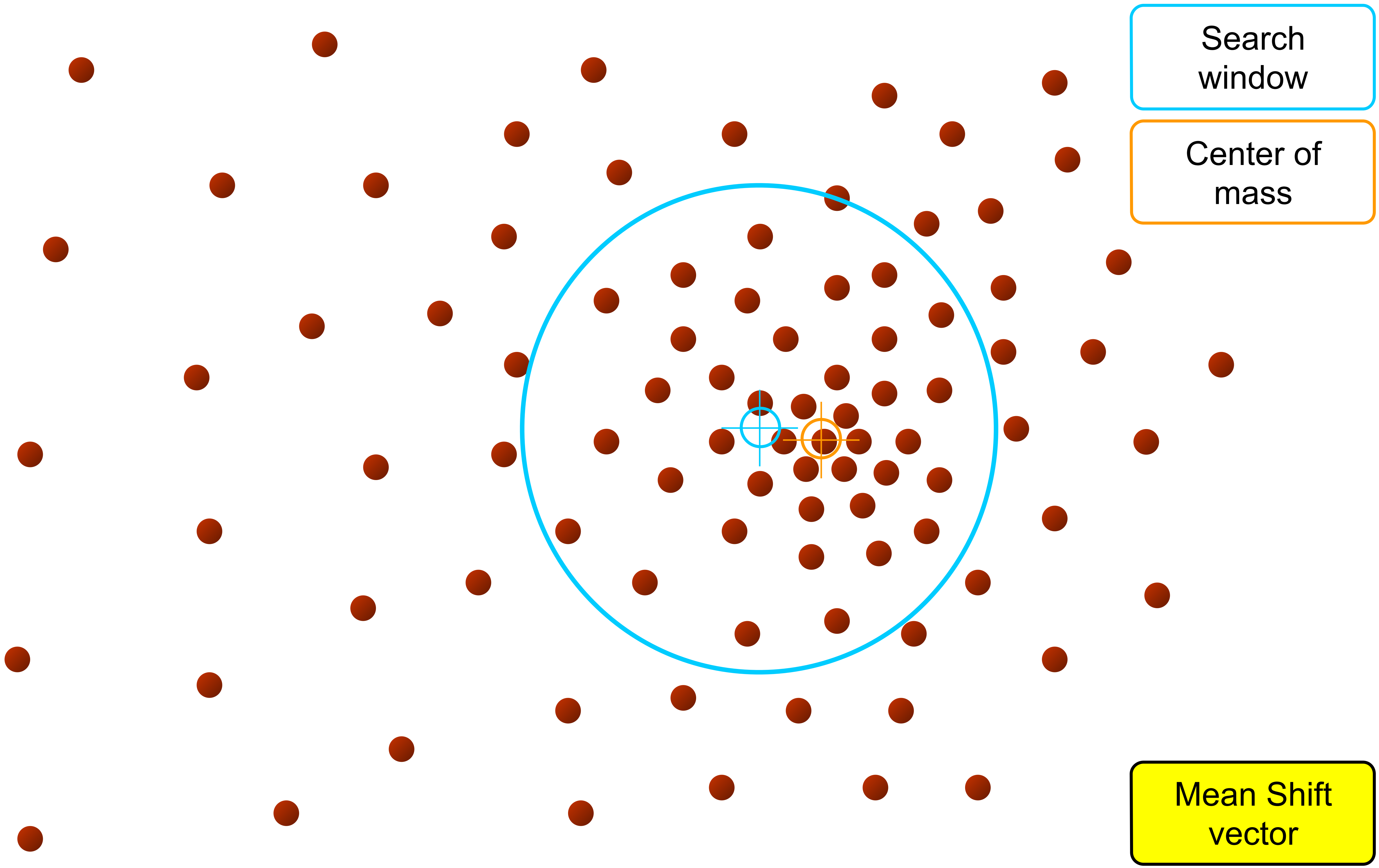
Slide by Y. Ukrainitz & B. Sarel

Mean shift

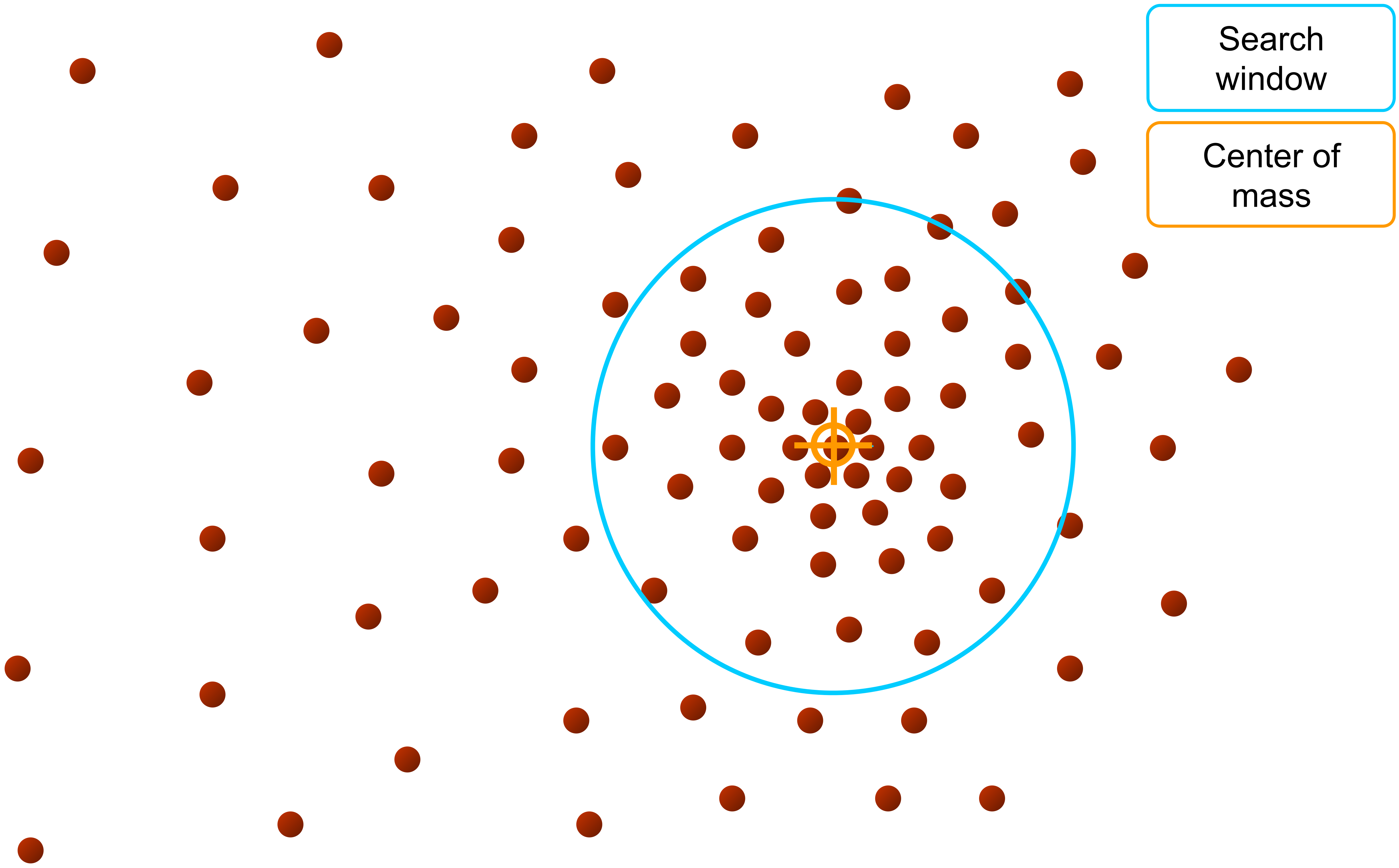


Slide by Y. Ukrainitz & B. Sarel

Mean shift



Mean shift



Slide by Y. Ukrainitz & B. Sarel

Computing the Mean Shift

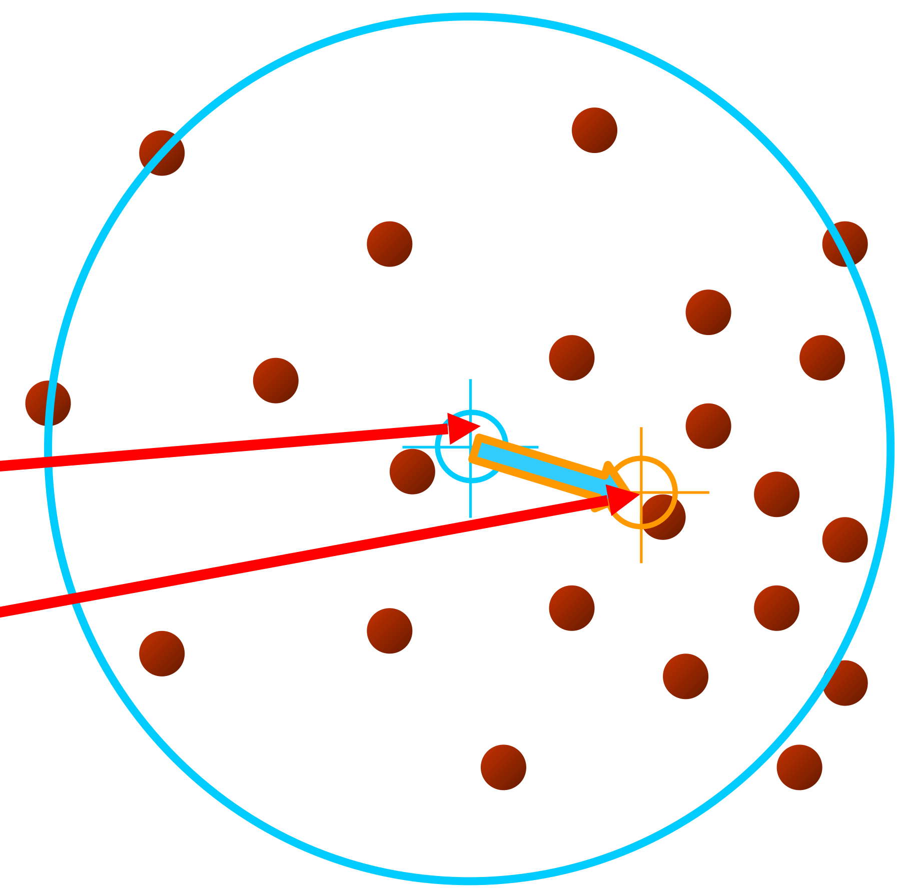
<u>Mean Shift procedure</u>:

For each point, repeat till convergence:

- Compute mean shift vector
- Translate the Kernel window by **m(x)**

$$\exp\left(-\frac{||\mathbf{x}-\mathbf{x}_i||^2}{h}\right)$$

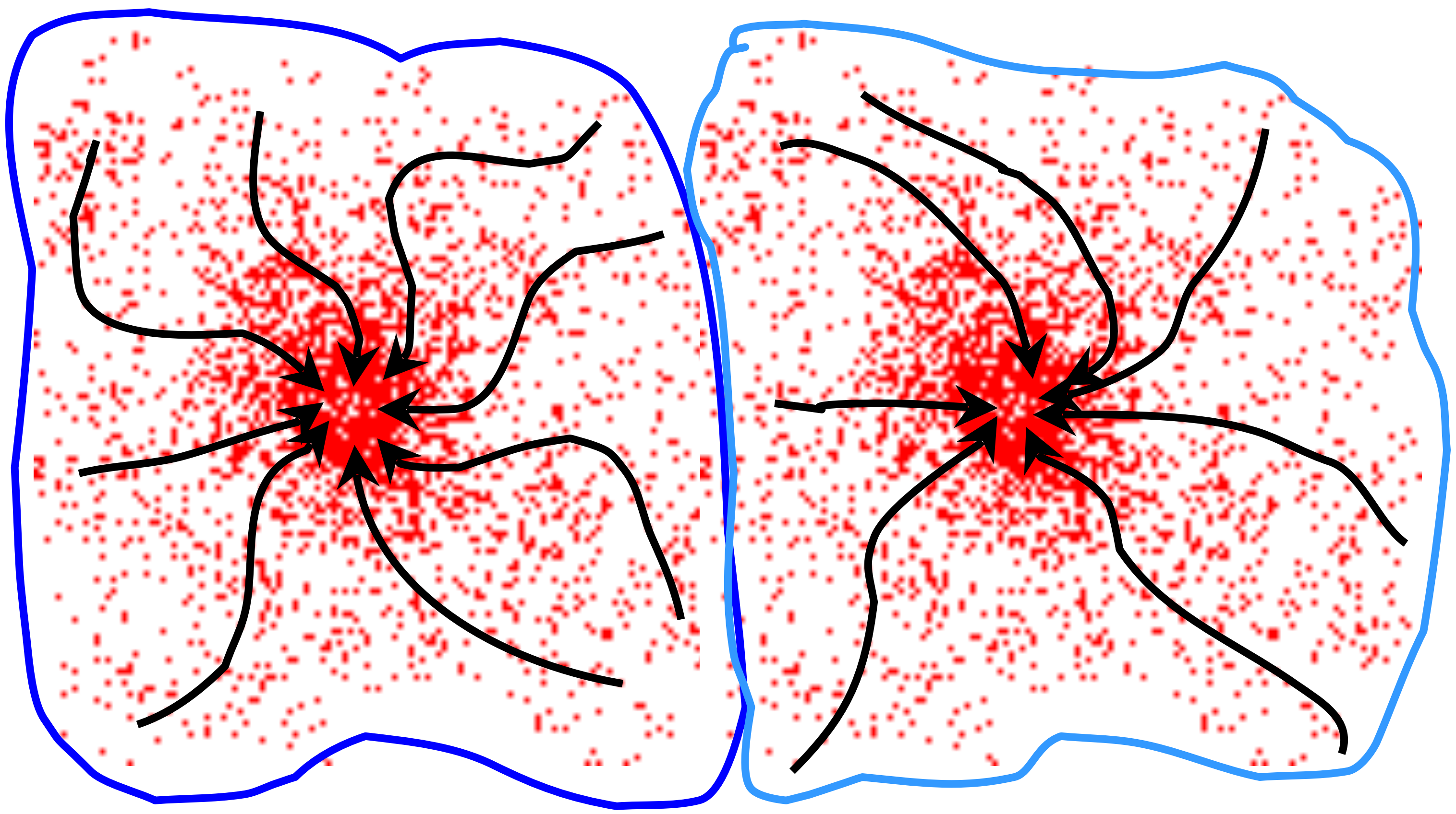
$$\mathbf{m}(\mathbf{x}) = \left[\frac{\sum_{i=1}^{n} \mathbf{x}_i g\left(\frac{\left\|\mathbf{x}-\mathbf{x}_i\right\|^2}{h}\right)}{\sum_{i=1}^{n} g\left(\frac{\left\|\mathbf{x}-\mathbf{x}_i\right\|^2}{h}\right)} - \boxed{\mathbf{x}}\right]$$



Slide by Y. Ukrainitz & B. Sarel

Mean shift clustering

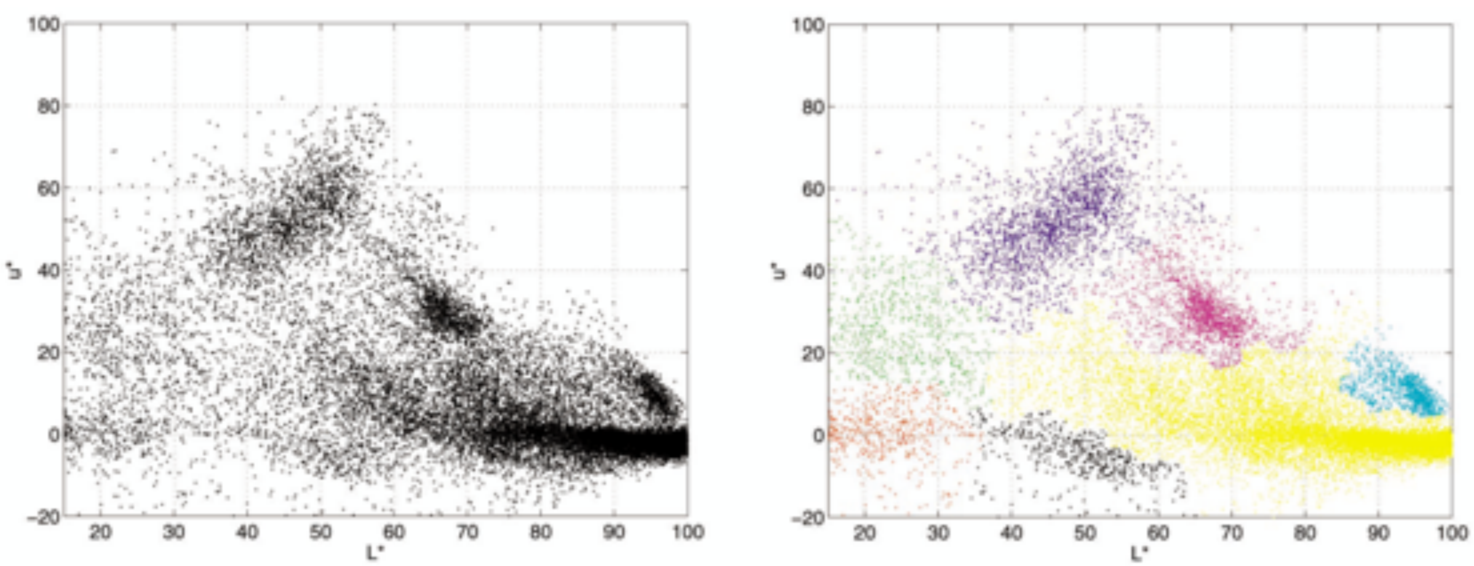
- Cluster: all data points in the attraction basin of a mode
- Attraction basin: the region for which all trajectories lead to the same mode



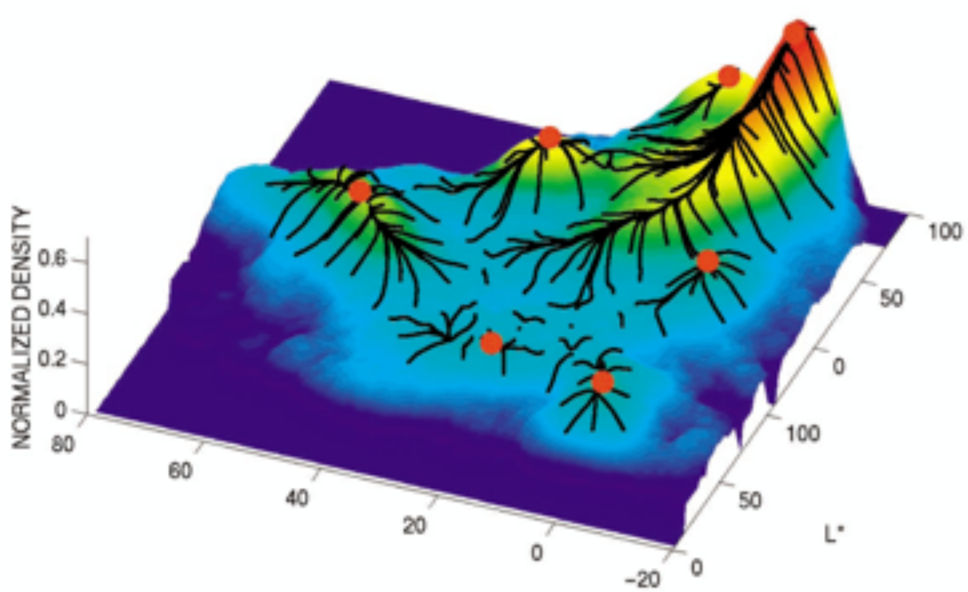
Slide by Y. Ukrainitz & B. Sarel

Mean shift clustering/segmentation

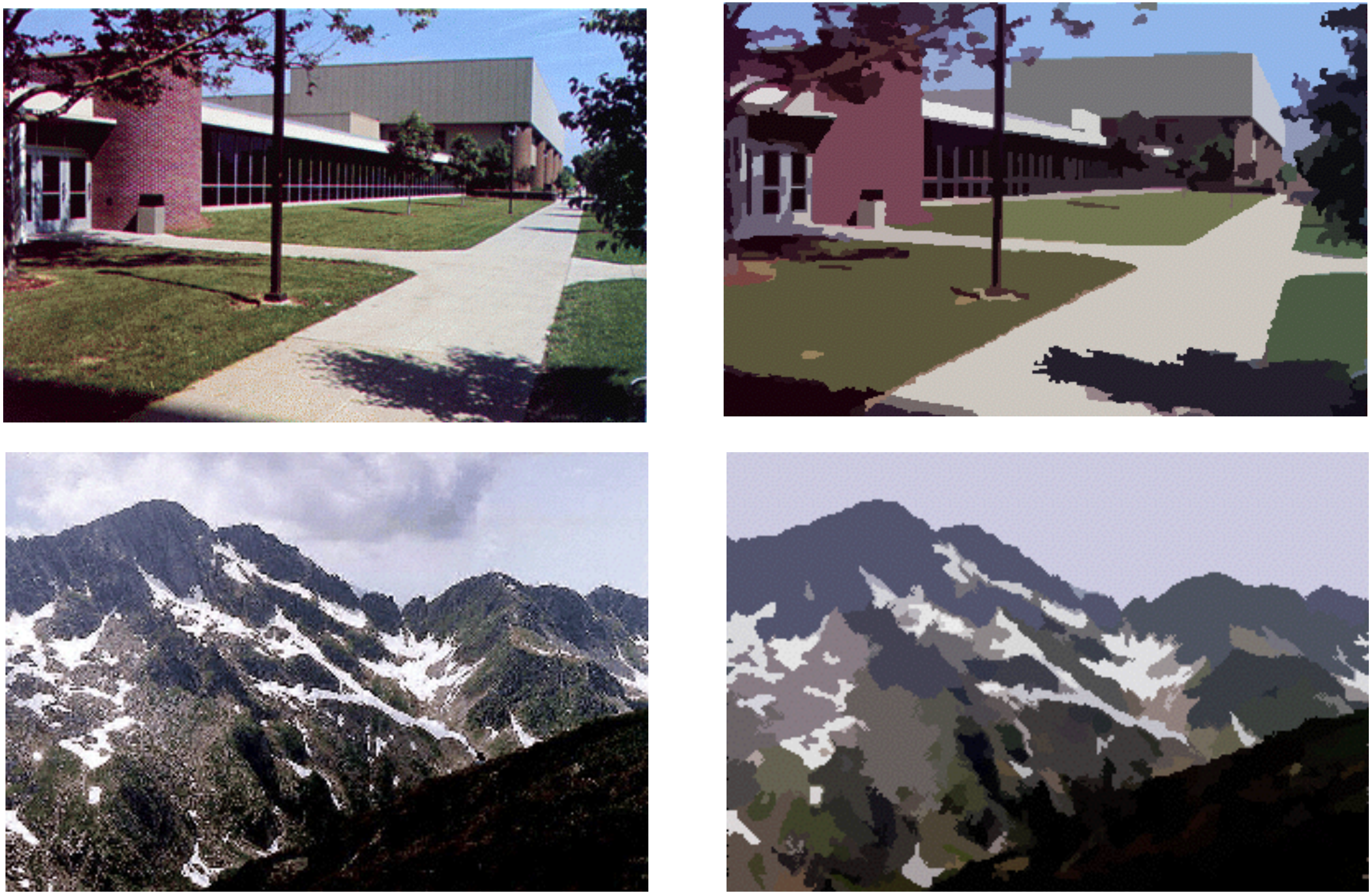
- Find features (color, gradients, texture, etc)
- Initialize windows at individual feature points
- Perform mean shift for each window until convergence
- Merge windows that end up near the same “peak” or mode



(a) (b)



Mean shift segmentation results



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

Mean shift clustering results

Mean shift

- Pros:
  - Does not assume shape on clusters
  - One parameter choice (window size)
  - Generic technique
  - Find multiple modes
- Cons:
  - Selection of window size
  - Is rather expensive: O(dN^2) per iteration
  - Does not work well for high-dimensional features