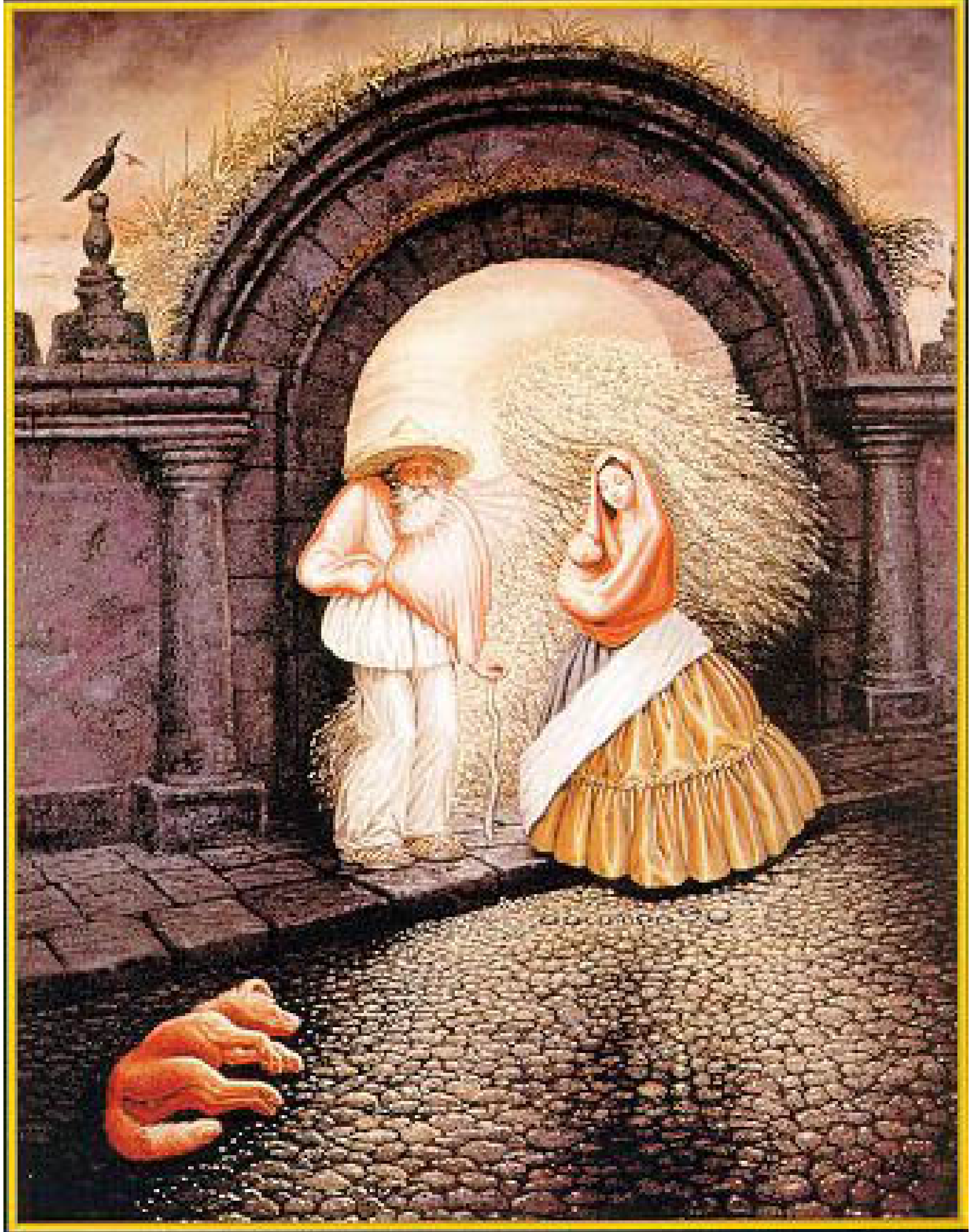
EECS 442 – Computer vision

Segmentation & Clustering

- Segmentation in human vision
- K-mean clustering
- Mean-shift
- Graph-cut

Reading: Chapters 14 [FP]

Some slides of this lectures are courtesy of prof F. Li, prof S. Lazebnik, and various other lecturers



Segmentation

- Compact representation for image data in terms of a set of components
- Components share “common” visual properties
- Properties can be defined at different level of abstractions

General ideas

- Tokens
  - whatever we need to group (pixels, points, surface elements, etc., etc.)
- Bottom up segmentation
  - tokens belong together because they are locally coherent
- Top down segmentation
  - tokens belong together because they lie on the same object

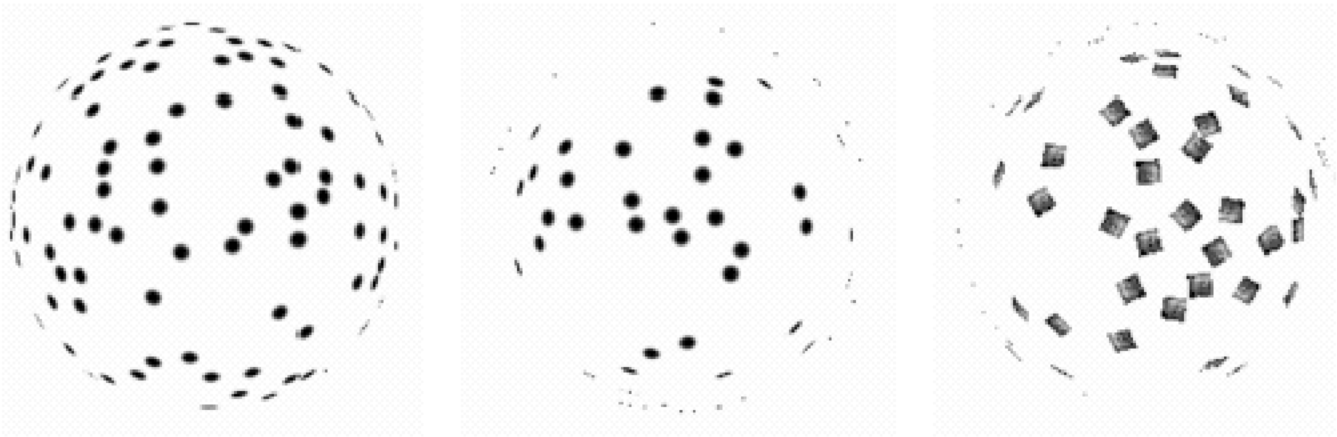
> These two are not mutually exclusive

What is Segmentation?

- Clustering image elements that "belong together"
  - Partitioning
    - Divide into regions/sequences with coherent internal properties
  - Grouping
    - Identify sets of coherent tokens in image

Slide credit: Christopher Rasmussen

What is Segmentation?

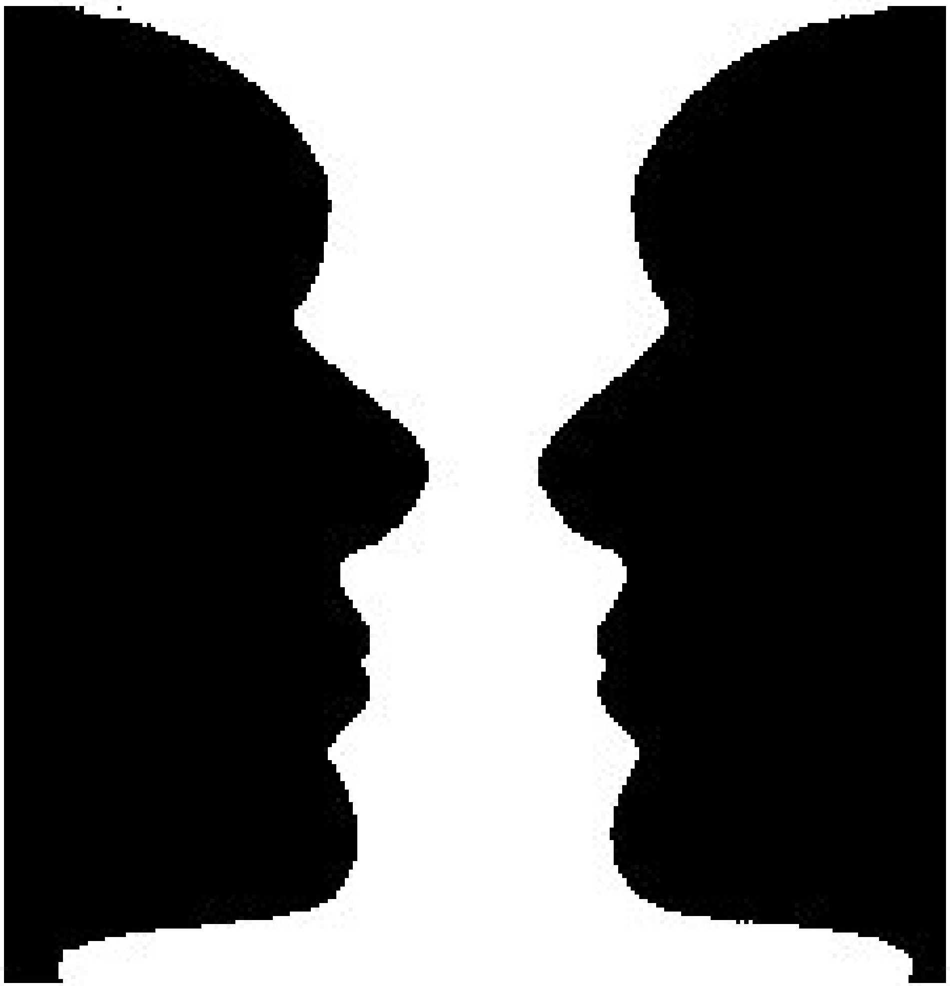


Why do these tokens belong together?

Basic ideas of grouping in human vision

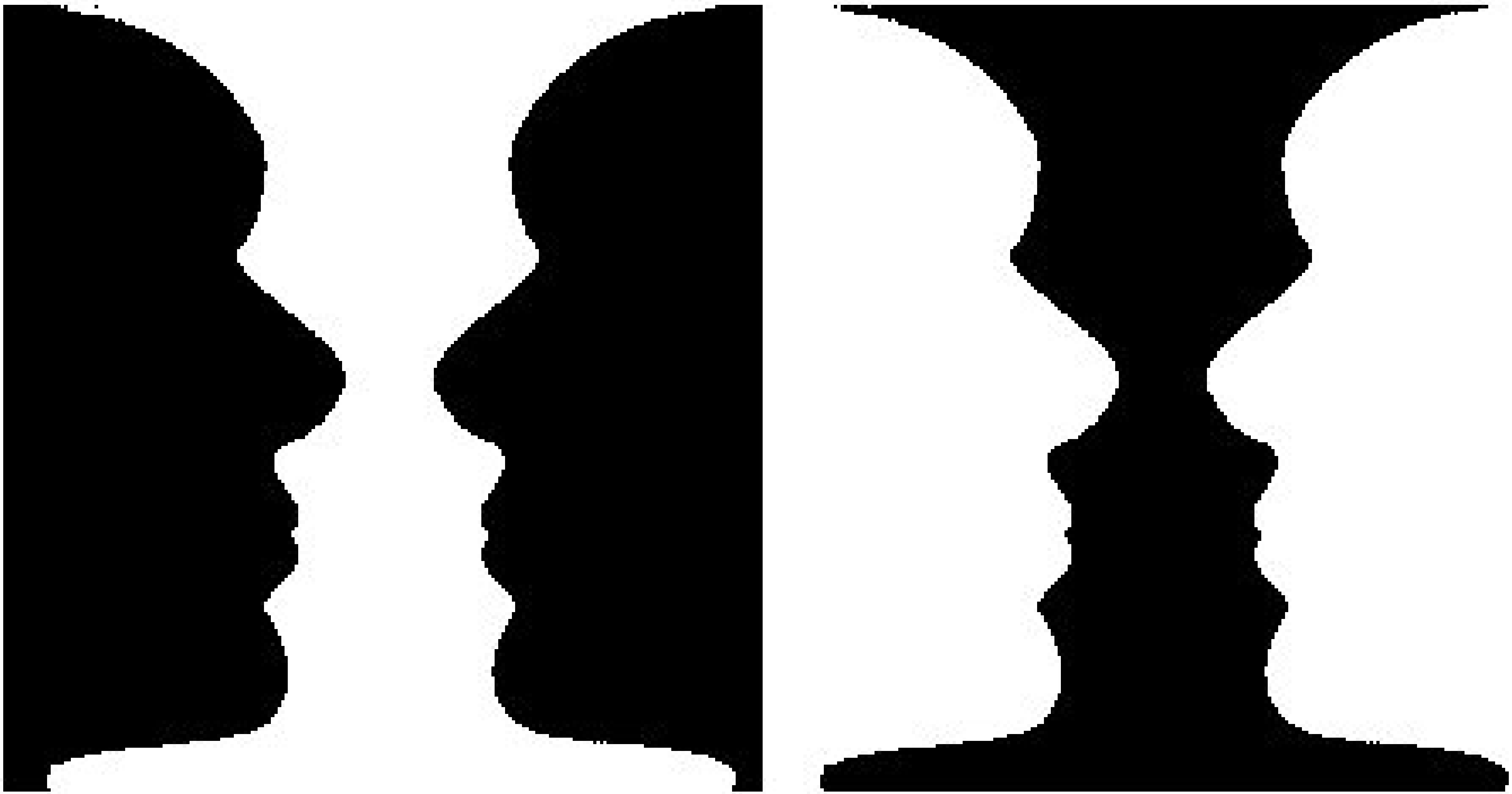
- Figure-ground discrimination
- Gestalt properties

Figure-ground discrimination



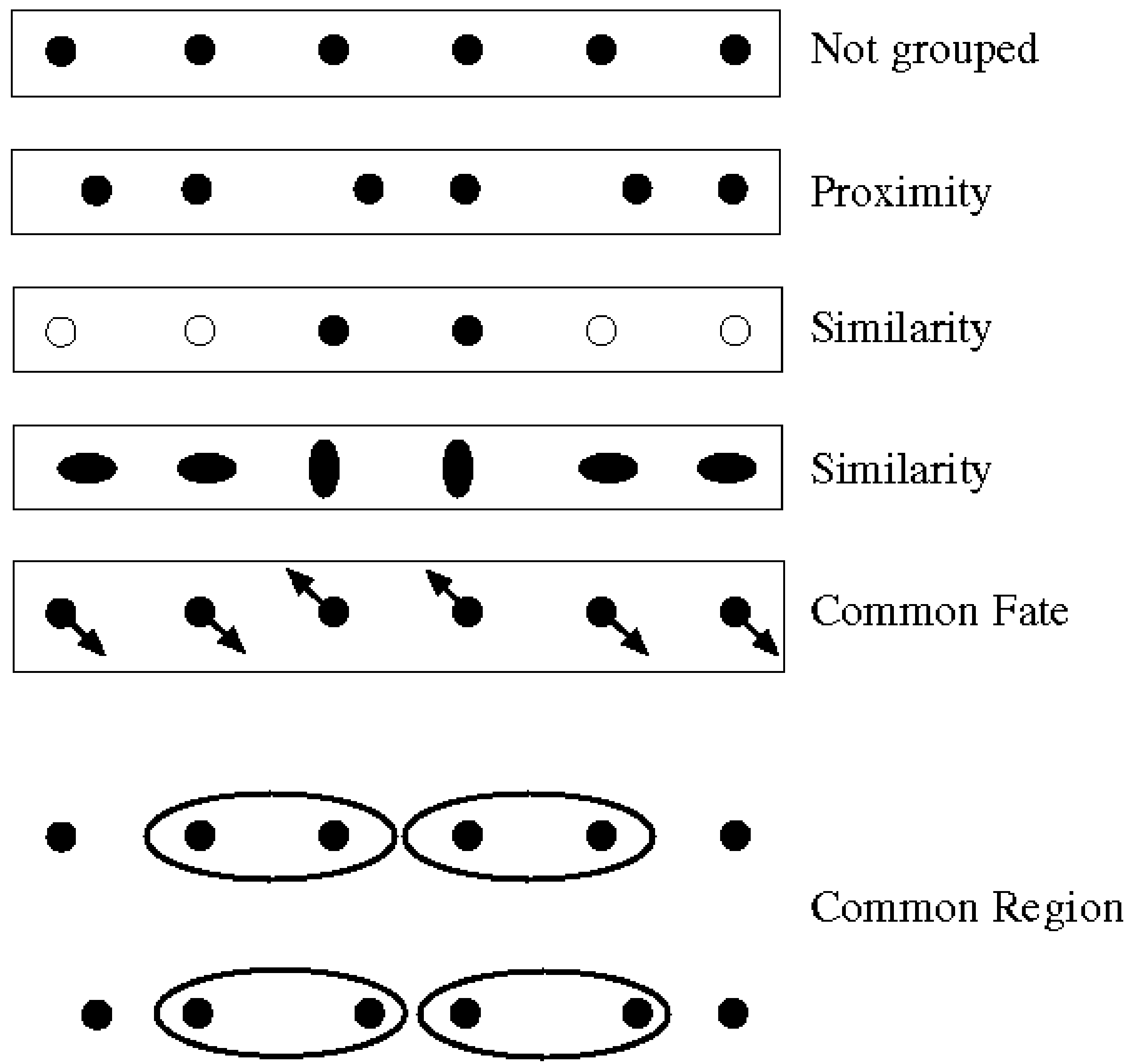
- Grouping can be seen in terms of allocating some elements to a figure, some to ground
- Can be based on local bottom-up cues or high level recognition

Figure-ground discrimination

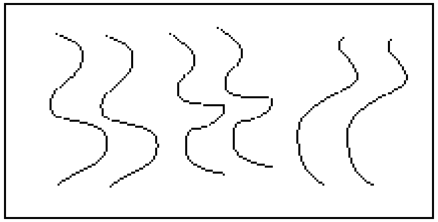


Gestalt properties

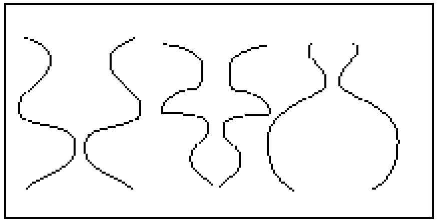
– A series of factors affect whether elements should be grouped together



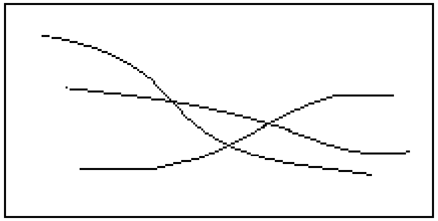
Gestalt properties



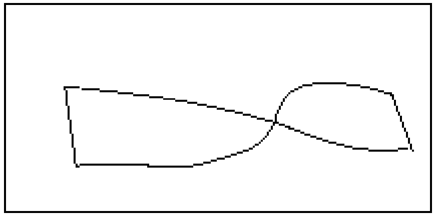
Parallelism



Symmetry

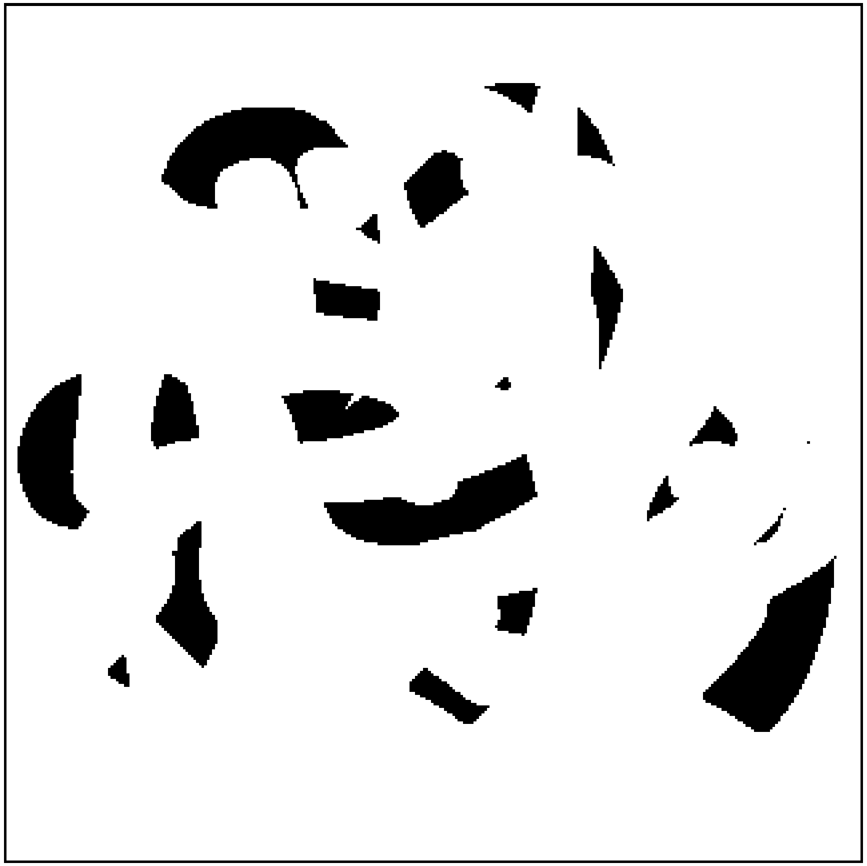


Continuity



Closure

Gestalt properties

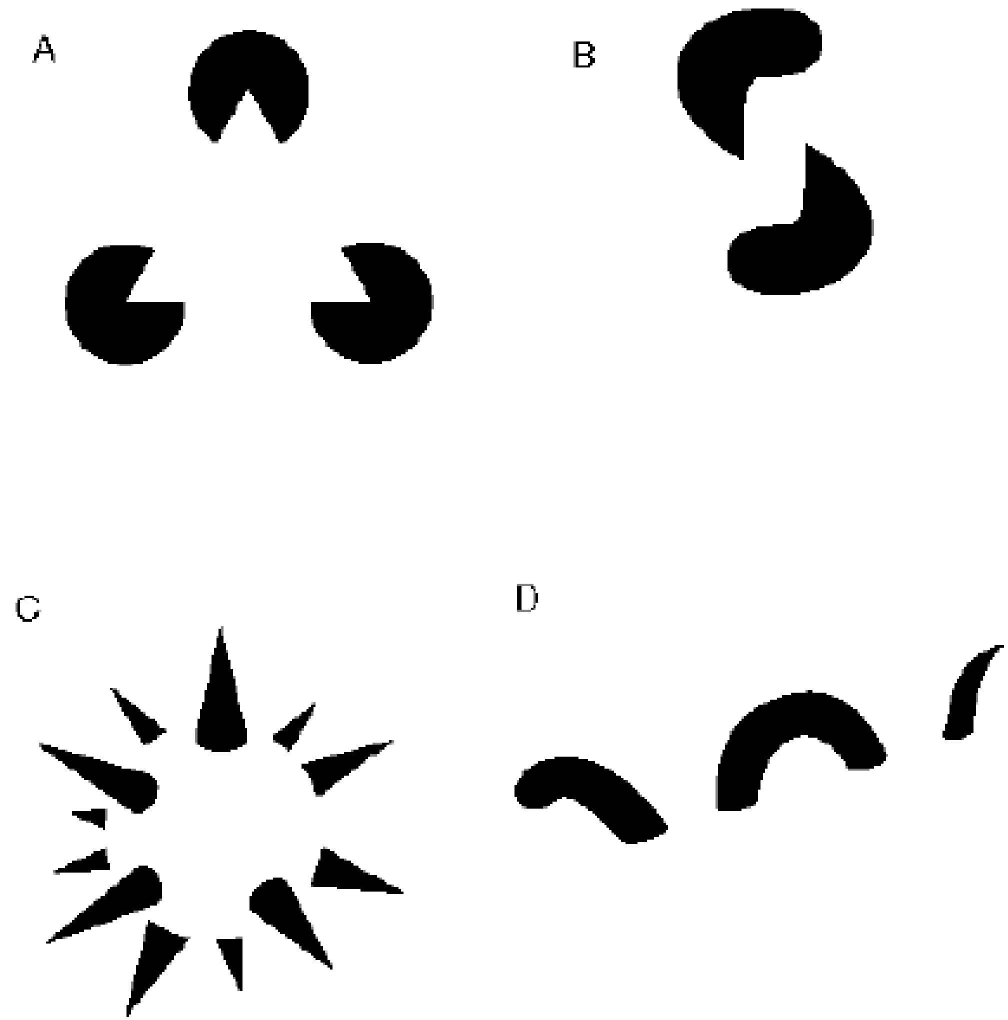


Gestalt properties

Grouping
by occlusions

Gestalt properties

Grouping by invisible completions



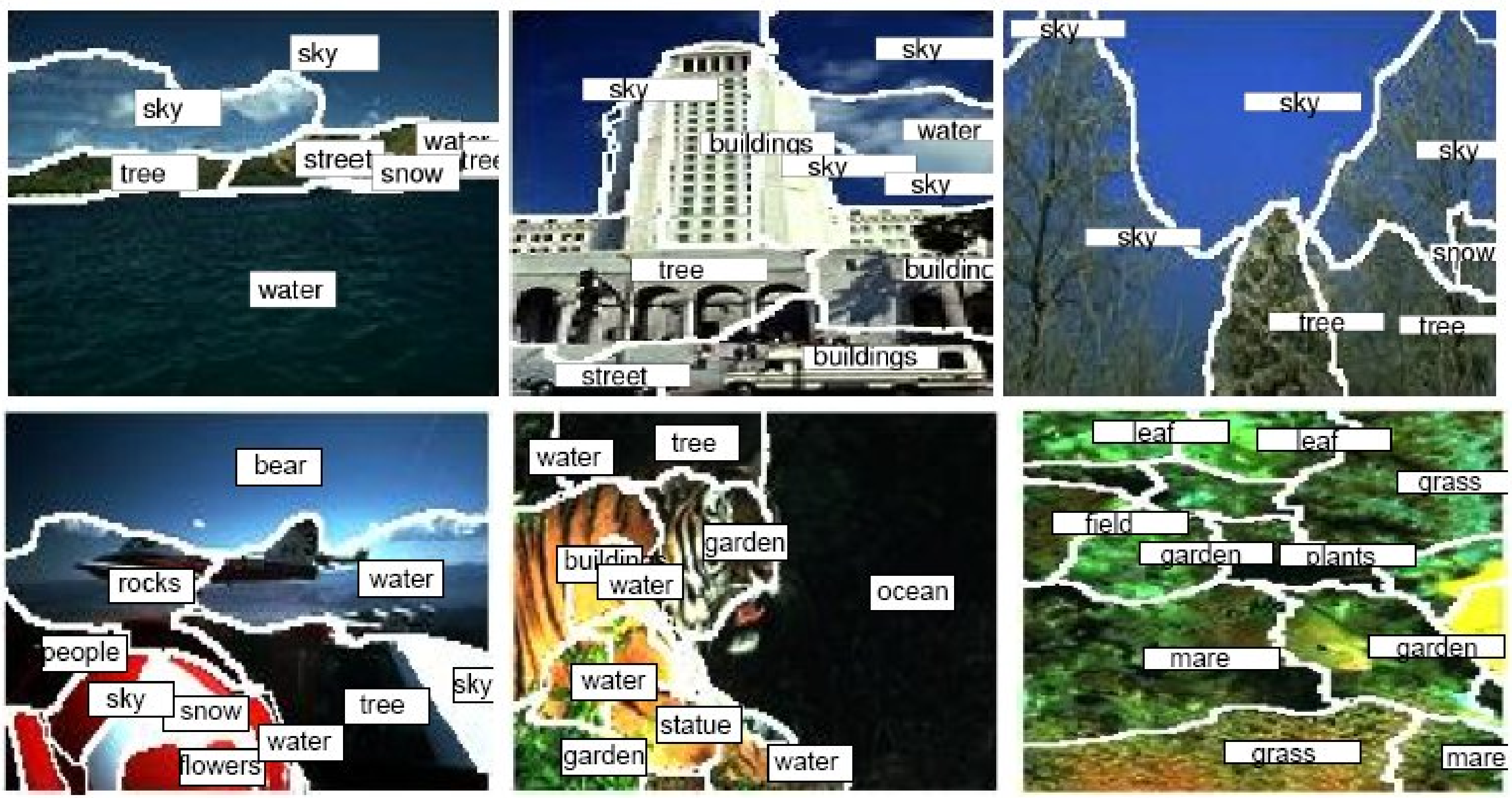
Emergence

Segmentation in computer vision

J. Malik, S. Belongie, T. Leung and J. Shi. "*Contour and Texture Analysis for Image Segmentation*". IJCV 43(1),7-27,2001.

Segmentation in computer vision

Object Recognition as Machine Translation, Duygulu, Barnard, de Freitas, Forsyth, ECCV02



Segmentation as clustering

Cluster together tokens that share similar visual characteristics

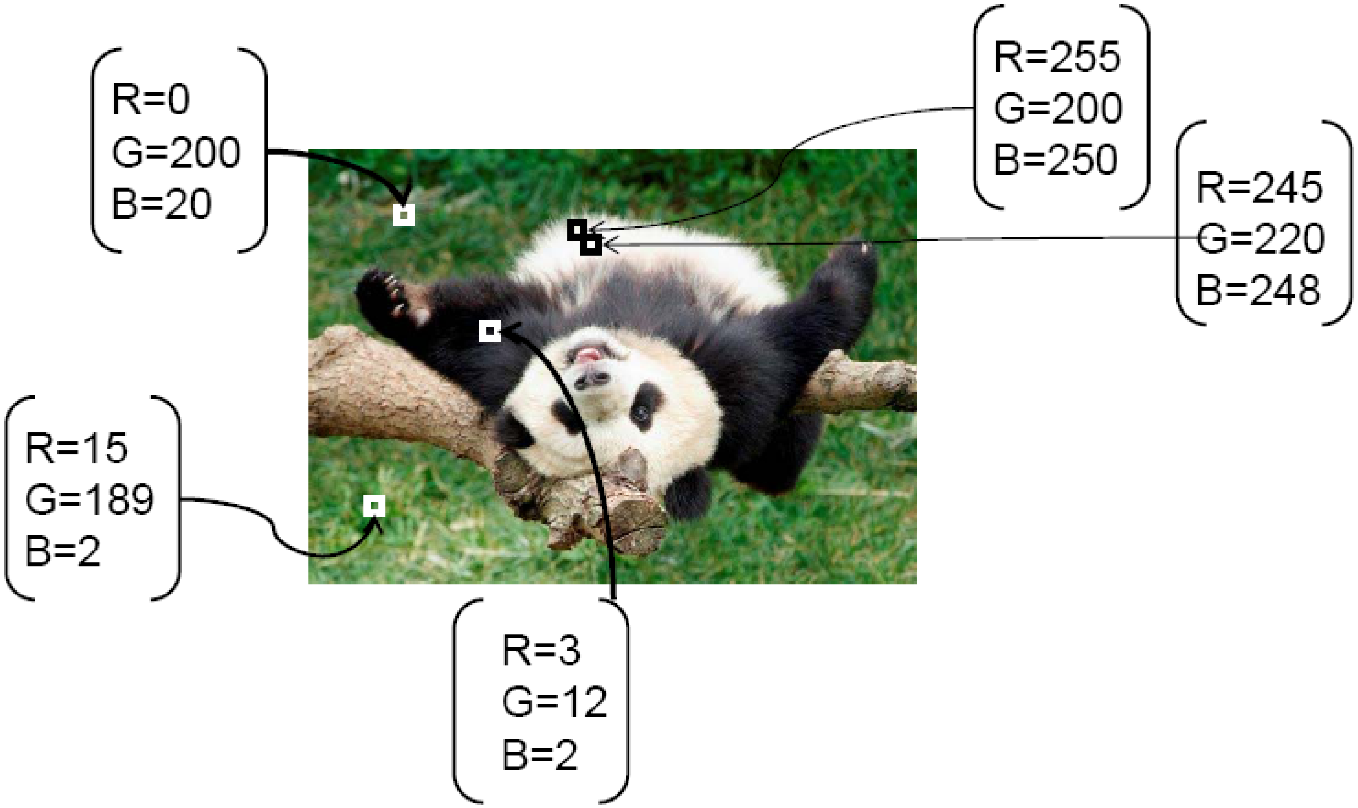
- K-mean
- Mean-shift
- Graph-cut

Feature Space

- Every token is identified by a set of salient visual characteristics. For example:
  - Position
  - Color
  - Texture
  - Motion vector
  - Size, orientation (if token is larger than a pixel)

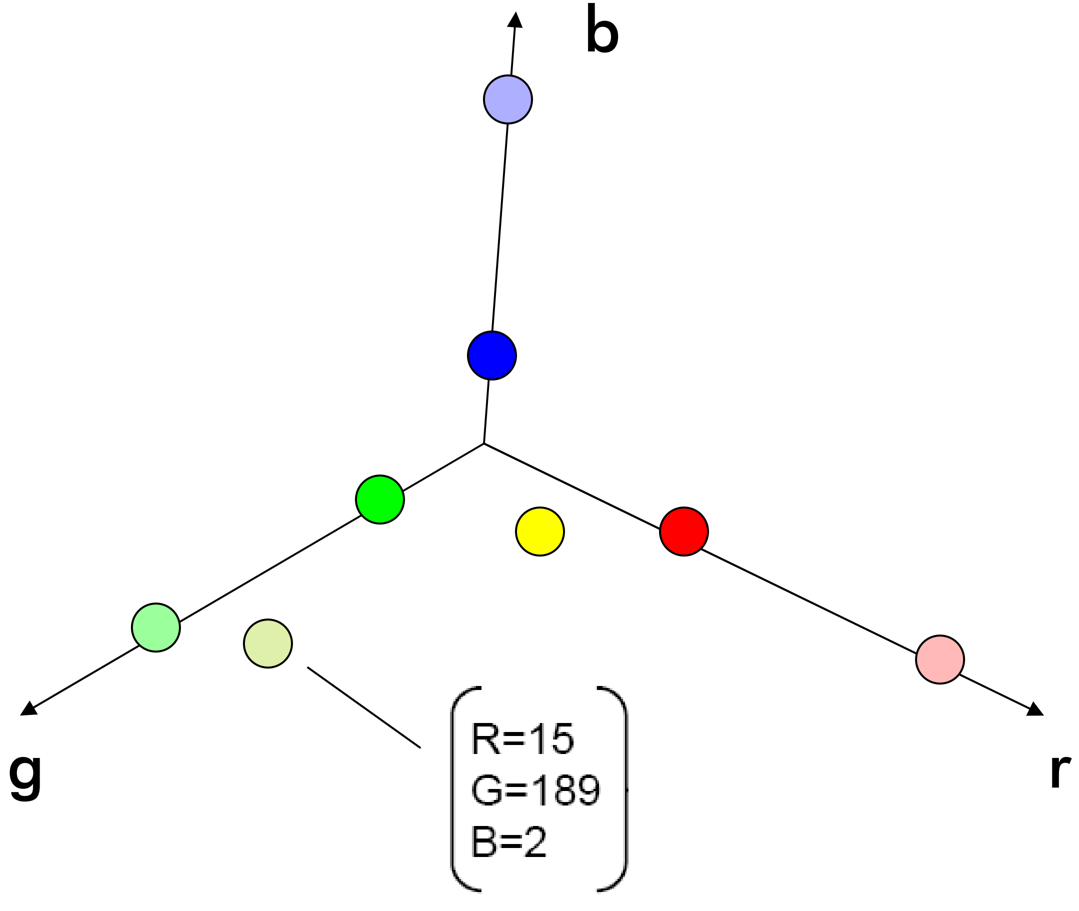
Slide credit: Christopher Rasmussen

Feature Space

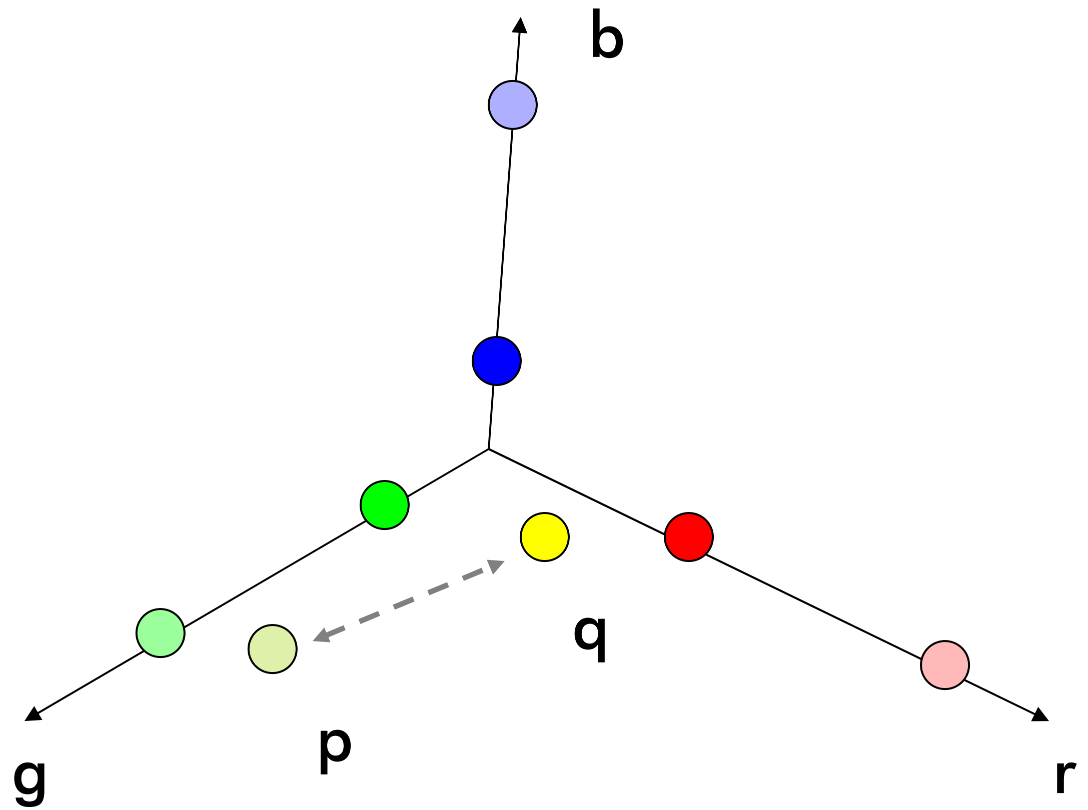


Source: K. Grauman

Feature space: each token is represented by a point

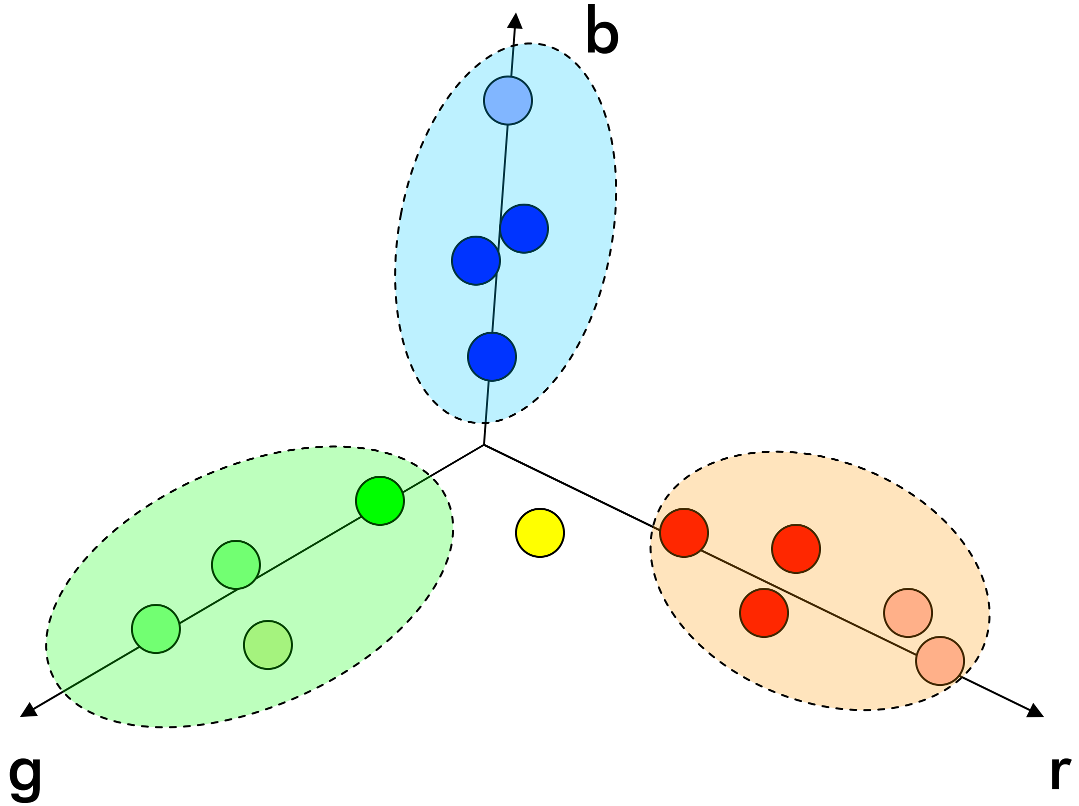


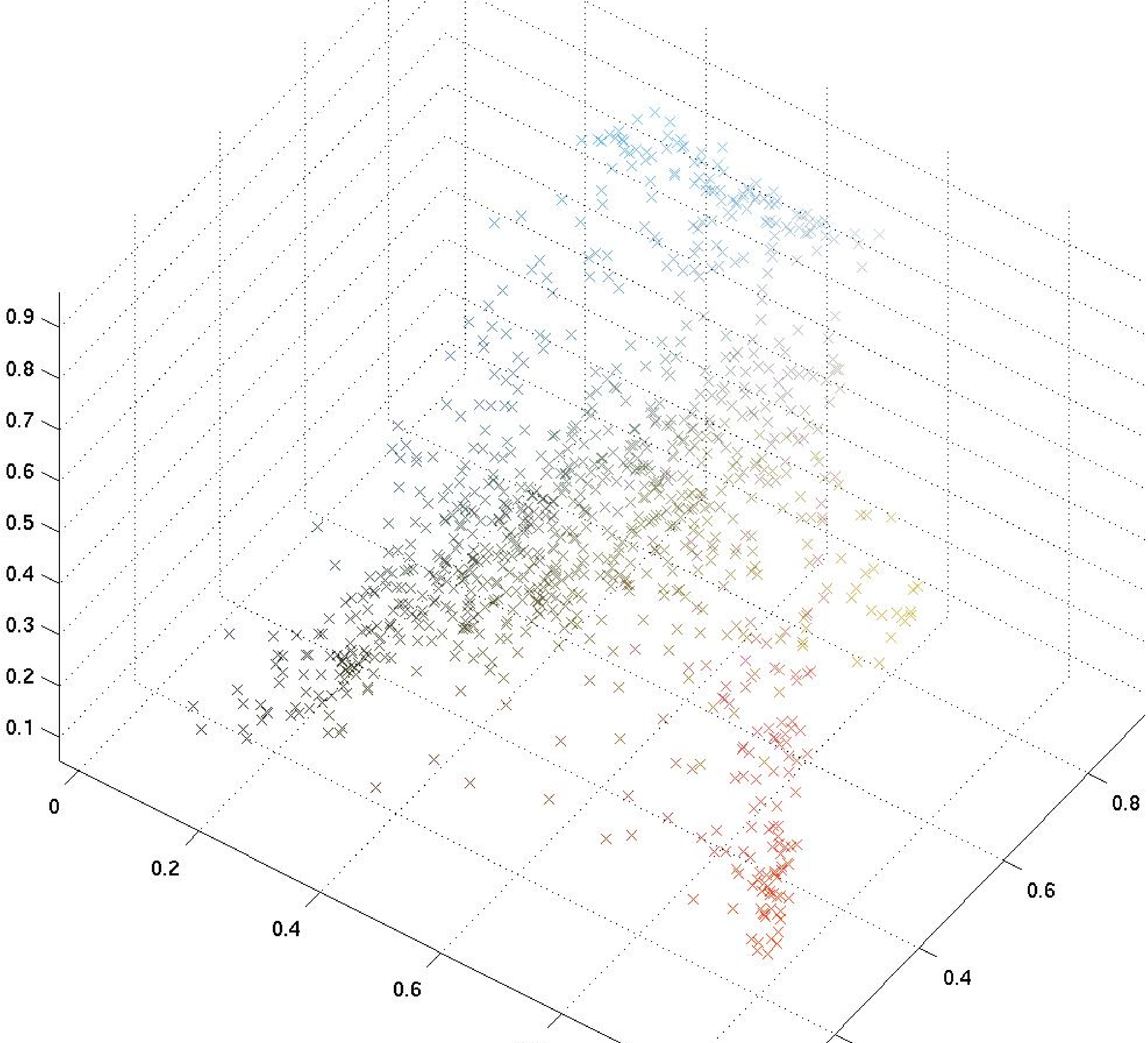
Token similarity is thus measured by distance between points ("feature vectors") in feature space

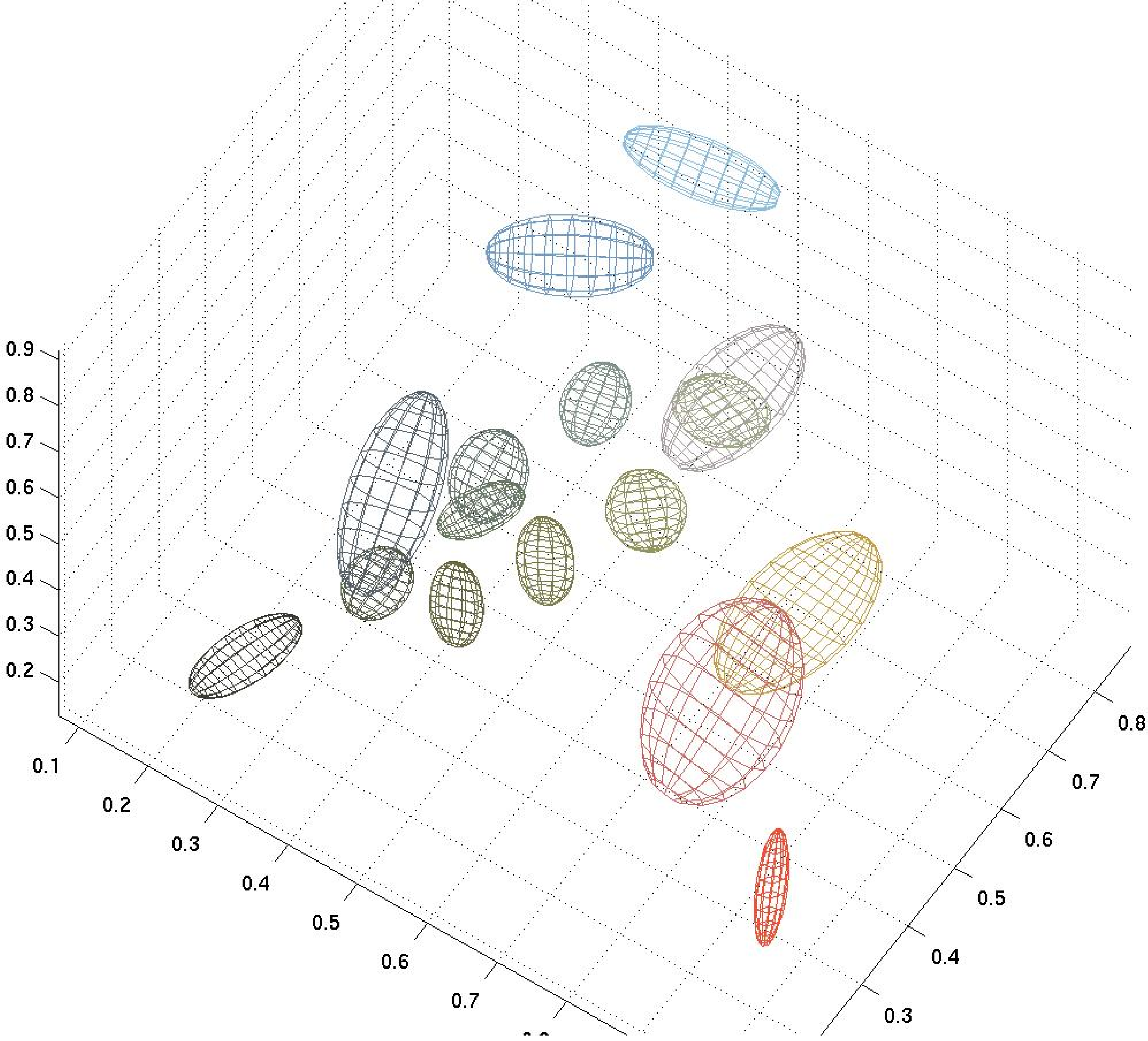


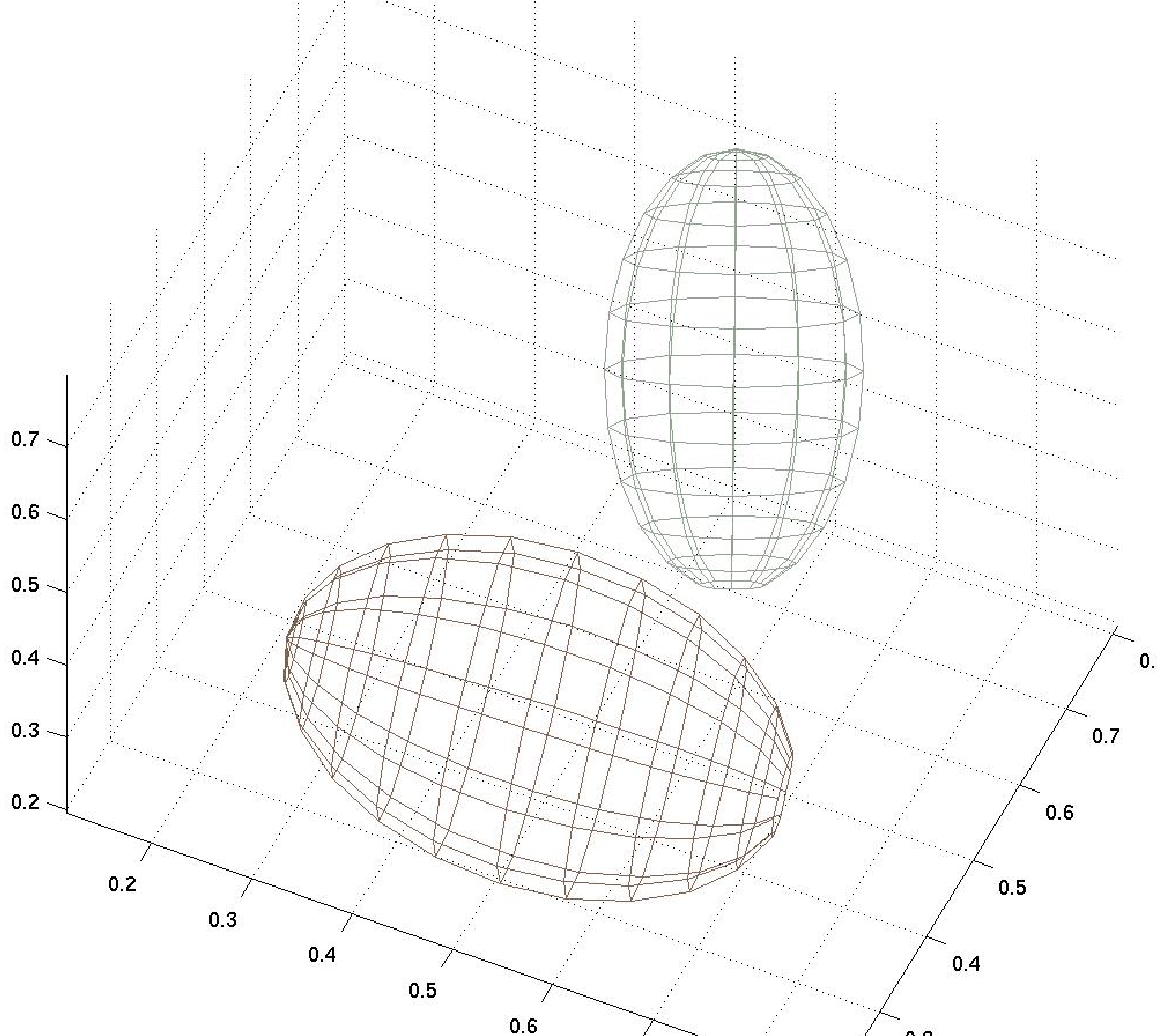
$$\sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \cdots + (p_n - q_n)^2} = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2}.$$

Cluster together tokens with high similarity




0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0.2
0.4
0.6
0.8
0.6
0.4


0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.7
0.6
0.5
0.4
0.3

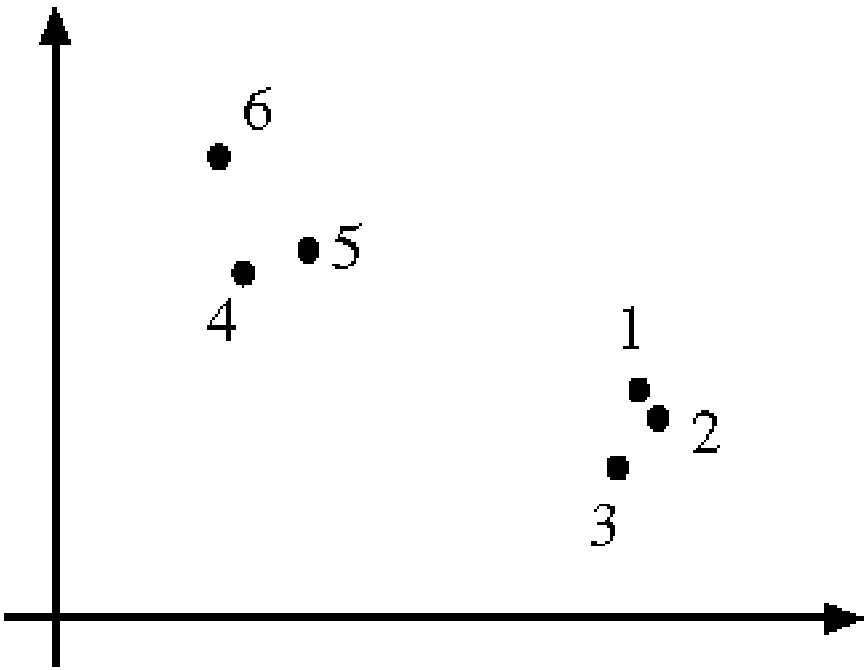


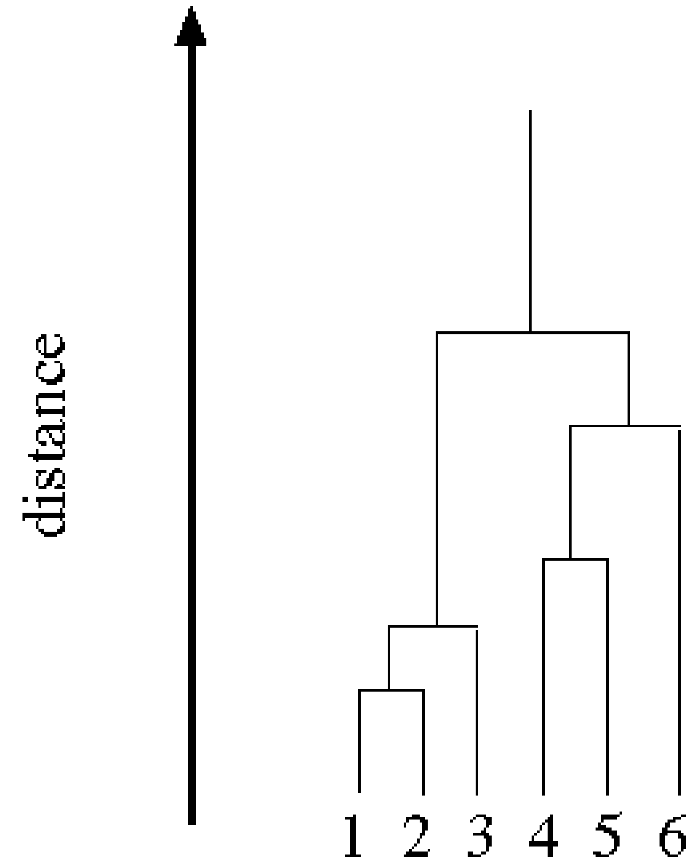
0.7
0.6
0.5
0.4
0.3
0.2
0.2
0.3
0.4
0.5
0.6
0.8
0.7
0.6
0.5
0.4

- Agglomerative clustering
  - Add token to cluster if token is similar enough to element of clusters
  - Repeat

- Divisive clustering
  - Split cluster into subclusters if along best boundary
  - Boundary separates subclusters based on similarity
  - Repeat

Hierachical structure of clusters (Dendrograms)





- rand('seed',12);
- X = rand(100,2);
- Y = pdist(X,'euclidean');
- Z = linkage(Y,'single');
- [H, T] = dendrogram(Z);

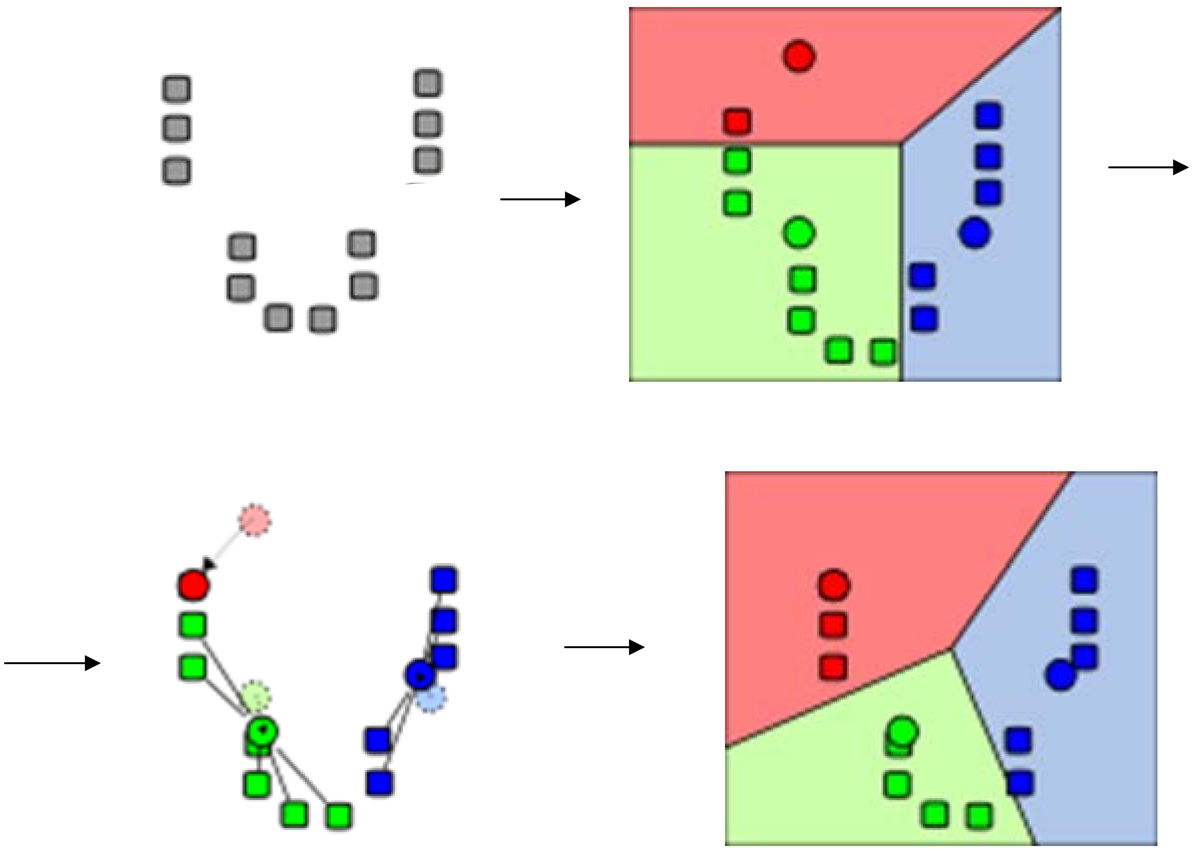
K-Means Clustering

- Initialization: Given K categories, N points in feature space. Pick K points randomly; these are initial cluster centers (means) $m_1, \ldots, m_K$. Repeat the following:

1. Assign each of the N points, x_i, to clusters by nearest m_i
2. Re-compute mean m_i of each cluster from its member points
3. If no mean has changed more than some ε, stop

Slide credit: Christopher Rasmussen

Example: 3-means Clustering



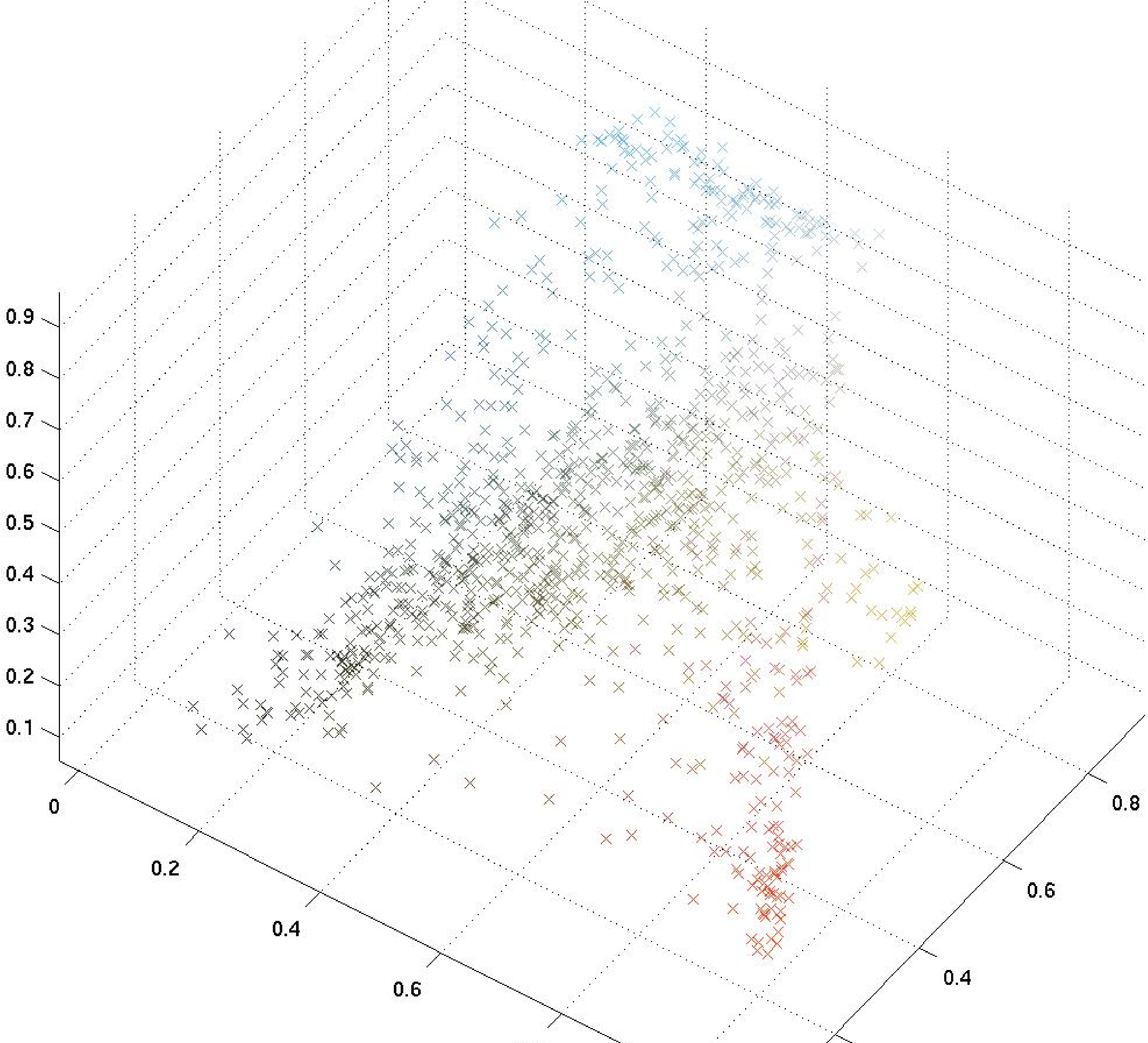
Source: wikipedia

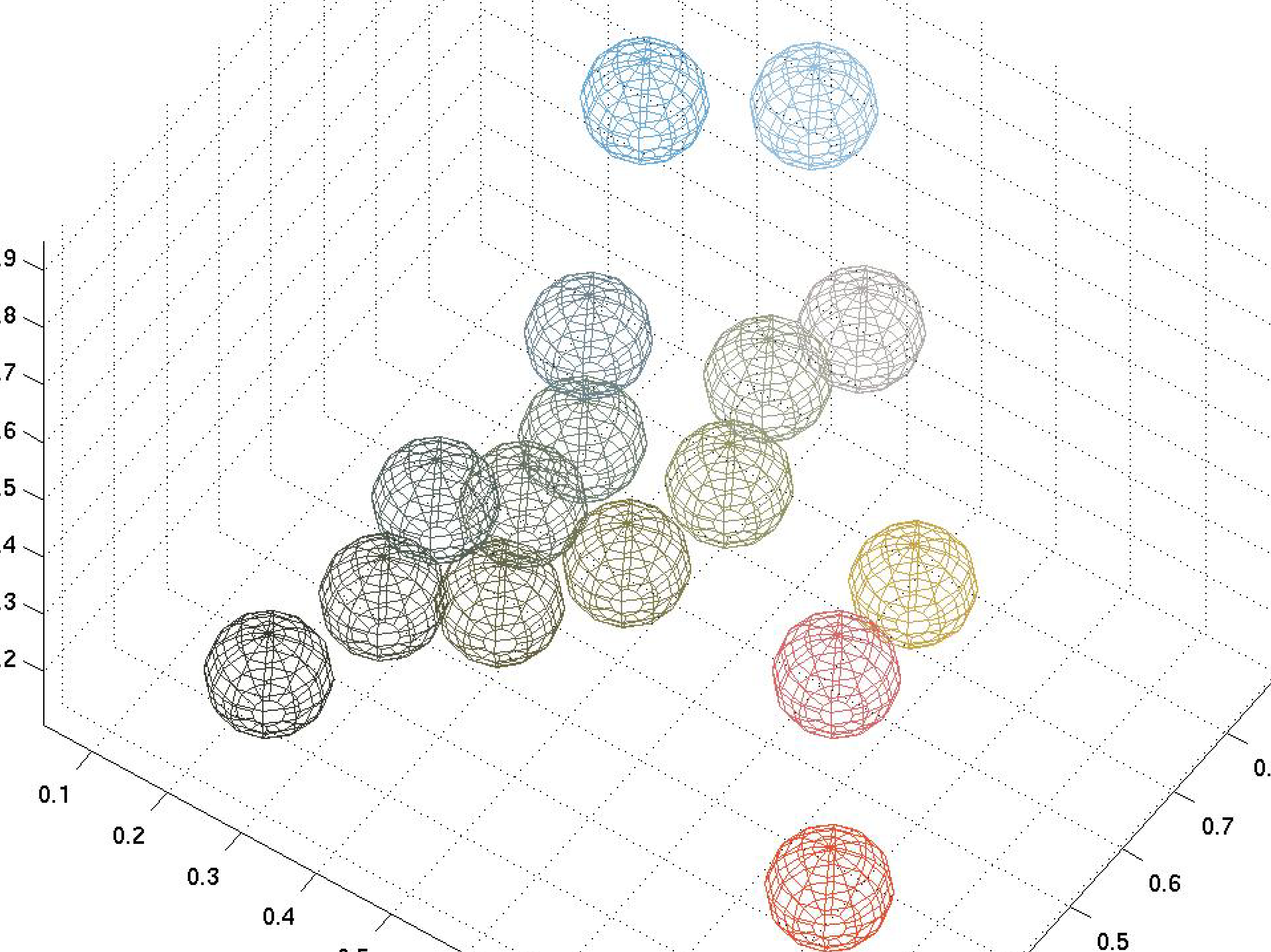
- Effectively carries out gradient descent to minimize:

$$\boxed{e(\mathbf{m}_i) = \sum_{i=1}^{n_C} \sum_{j;c_j=i} |\mathbf{x}_j - \mathbf{m}_i|^2}$$

$$\frac{\partial e}{\partial \mathbf{m}_k} = \sum_{j;c_j=k} -2(\mathbf{x}_j - \mathbf{m}_k) = 0$$

$$\mathbf{m}_k = \frac{\sum_{j;c_j=k} \mathbf{x}_j}{\sum_{j;c_j=k} 1} = \frac{1}{n_k} \sum_{j;c_j=k} \mathbf{x}_j$$

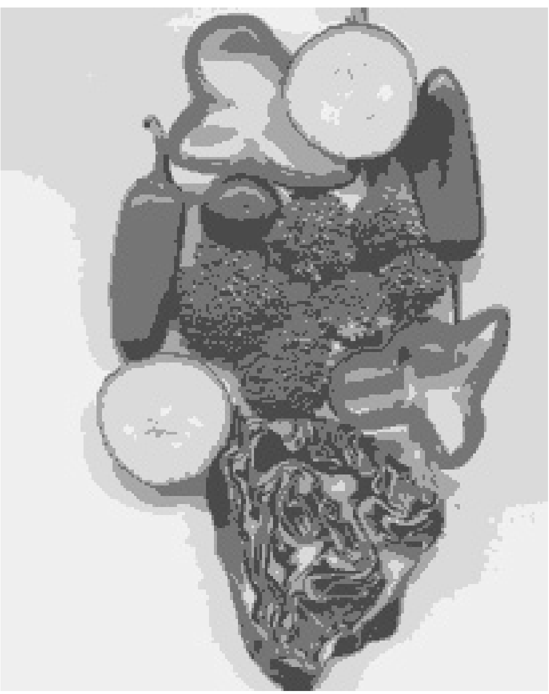

0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0.2
0.4
0.6
0.8
0.6
0.4


0.1
0.2
0.3
0.4
0.7
0.6
0.5

K-means clustering using intensity alone and color alone

Image

Clusters on intensity

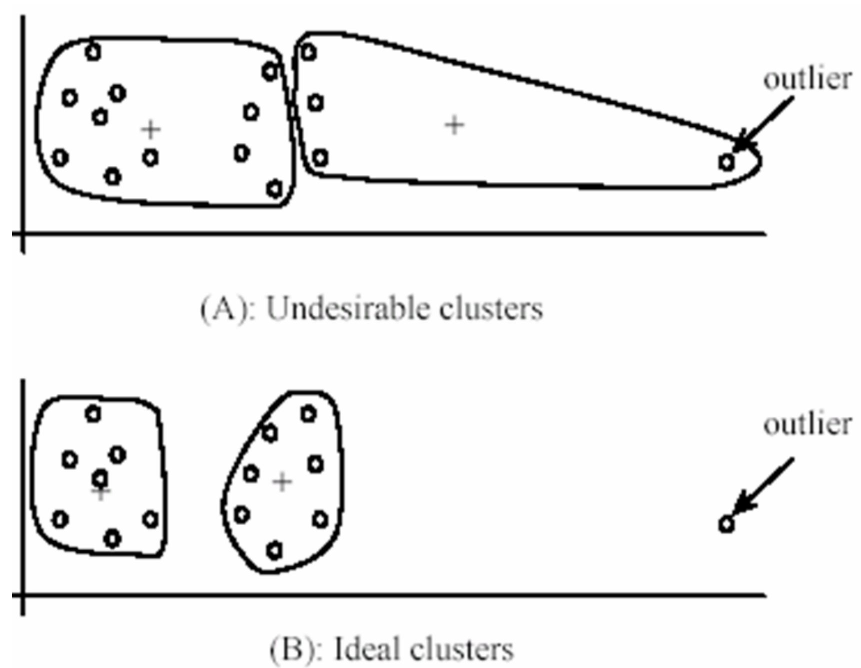


Clusters on color

K-Means pros and cons

- Pros
  - Simple and fast
  - Converges to a local minimum of the error function

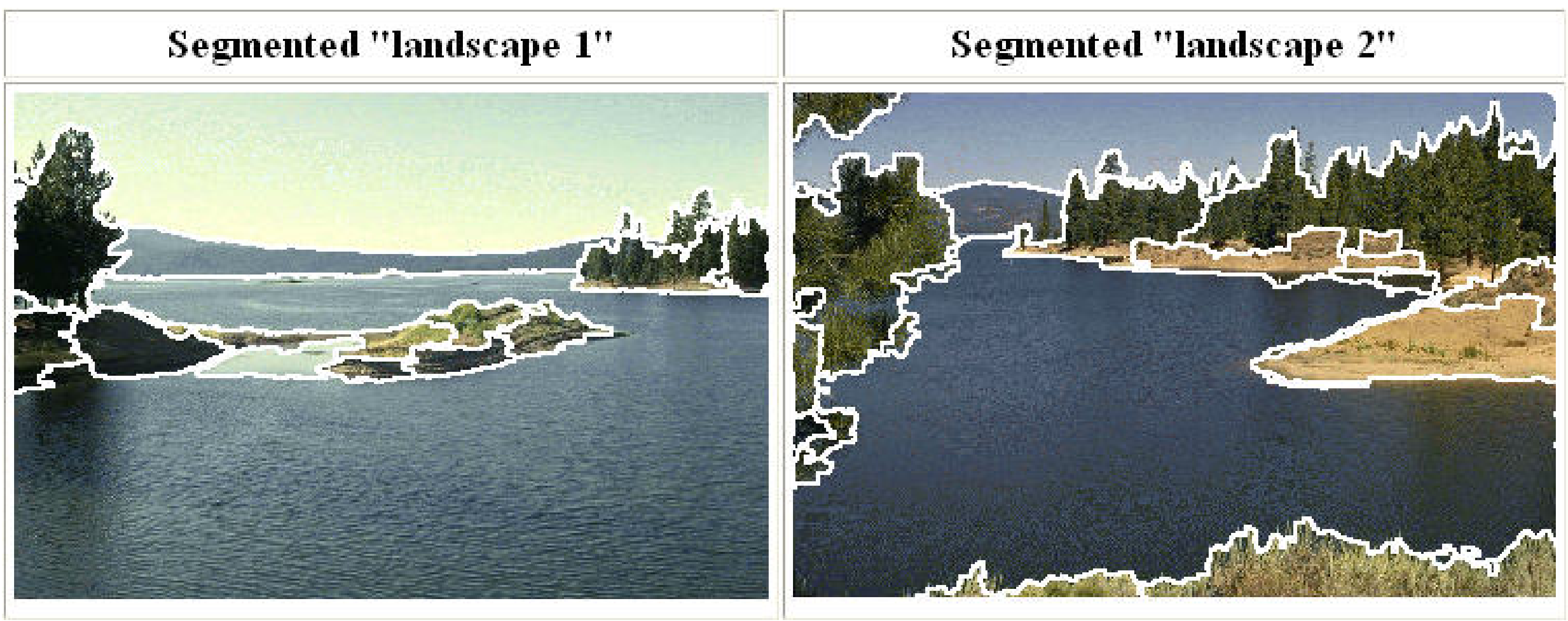
- Cons
  - Need to pick K
  - Sensitive to initialization
  - Only finds "spherical" clusters
  - Sensitive to outliers



Mean shift segmentation

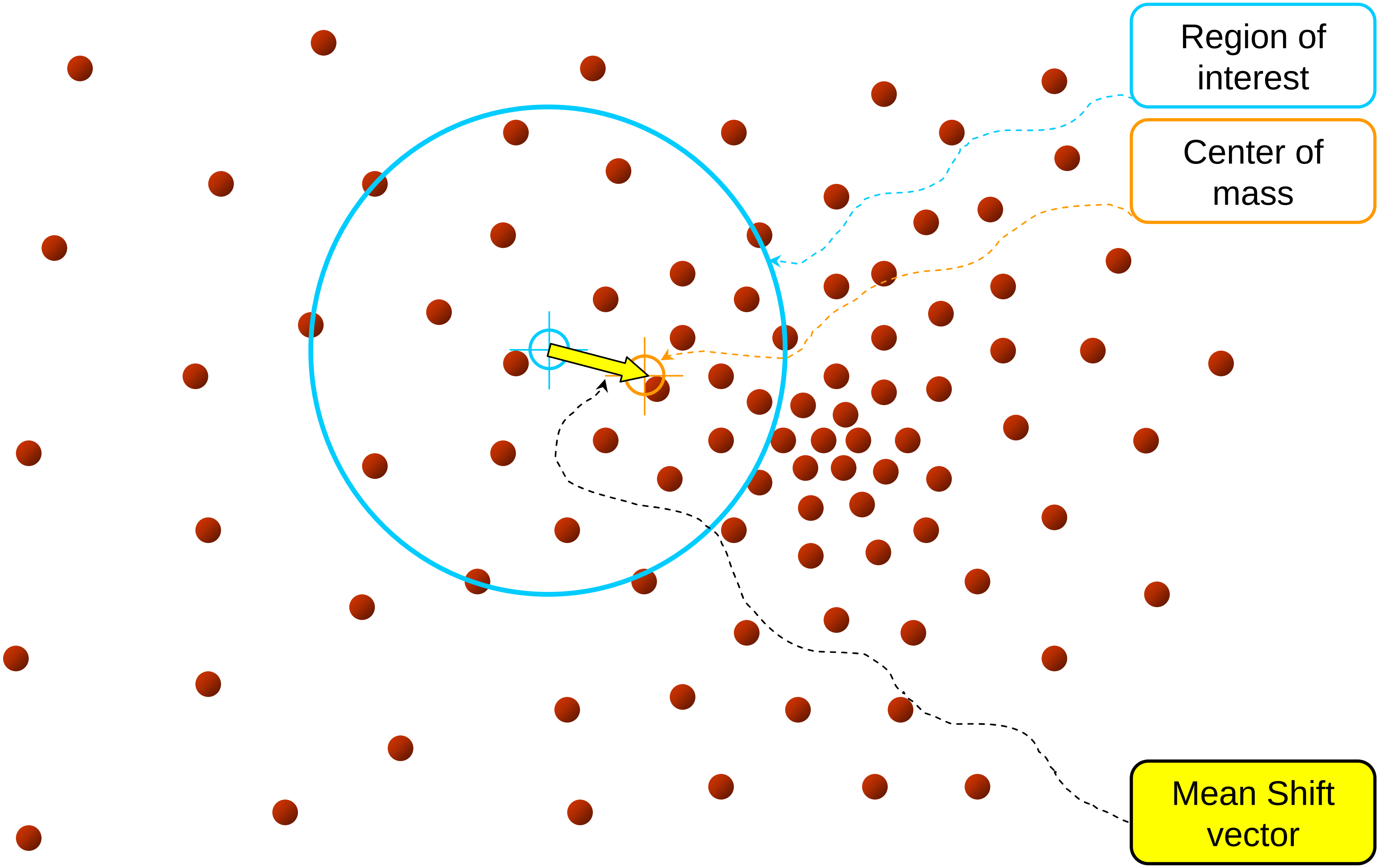
D. Comaniciu and P. Meer, Mean Shift: A Robust Approach toward Feature Space Analysis, PAMI 2002.

- An advanced and versatile technique for clustering-based segmentation

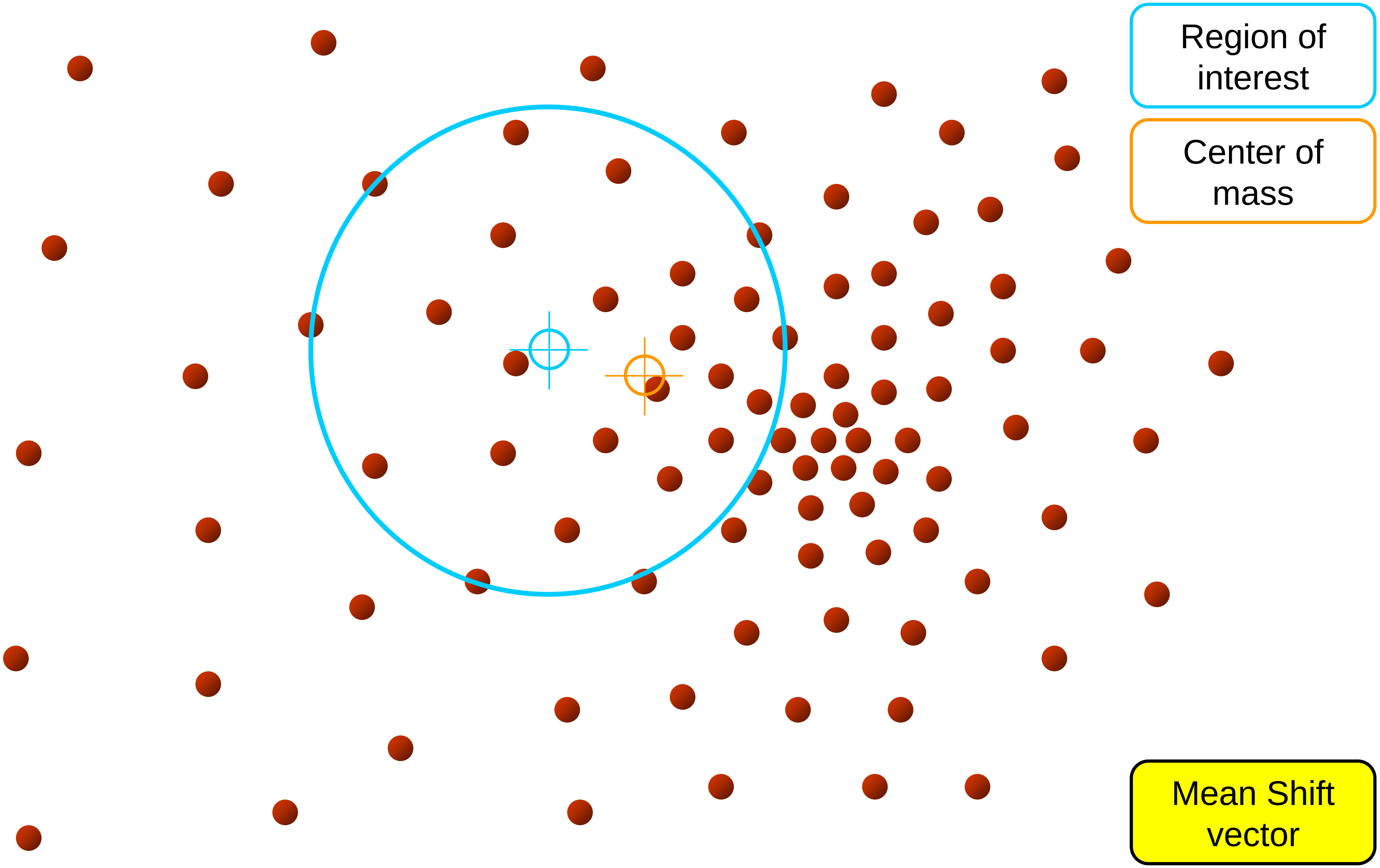


Mean shift segmentation

- The mean shift algorithm seeks a *mode* or local maximum of density of a given distribution
  - Choose a search window (width and location)
  - Compute the mean of the data in the search window
  - Center the search window at the new mean location
  - Repeat until convergence

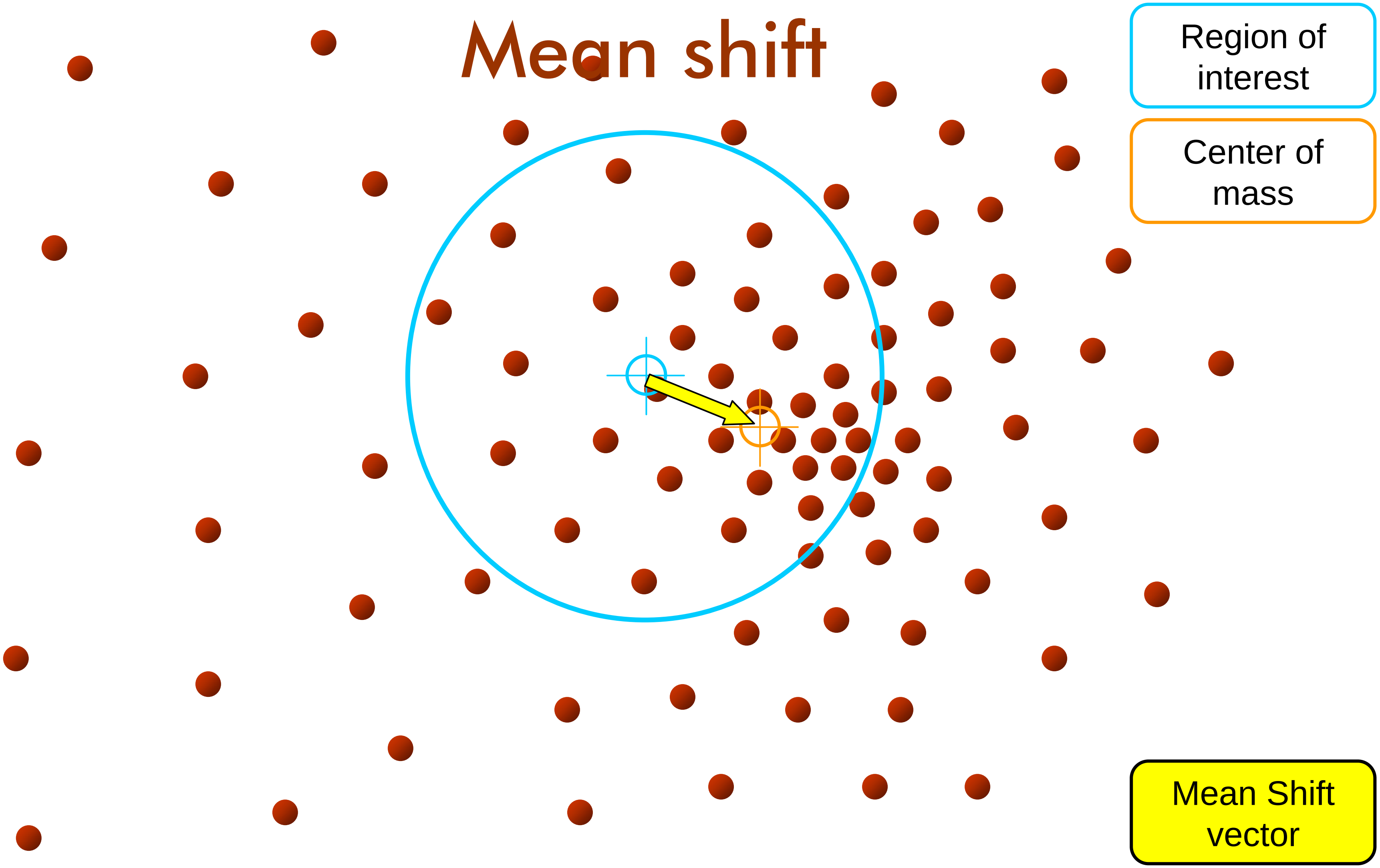


Slide by Y. Ukrainitz & B. Sarel



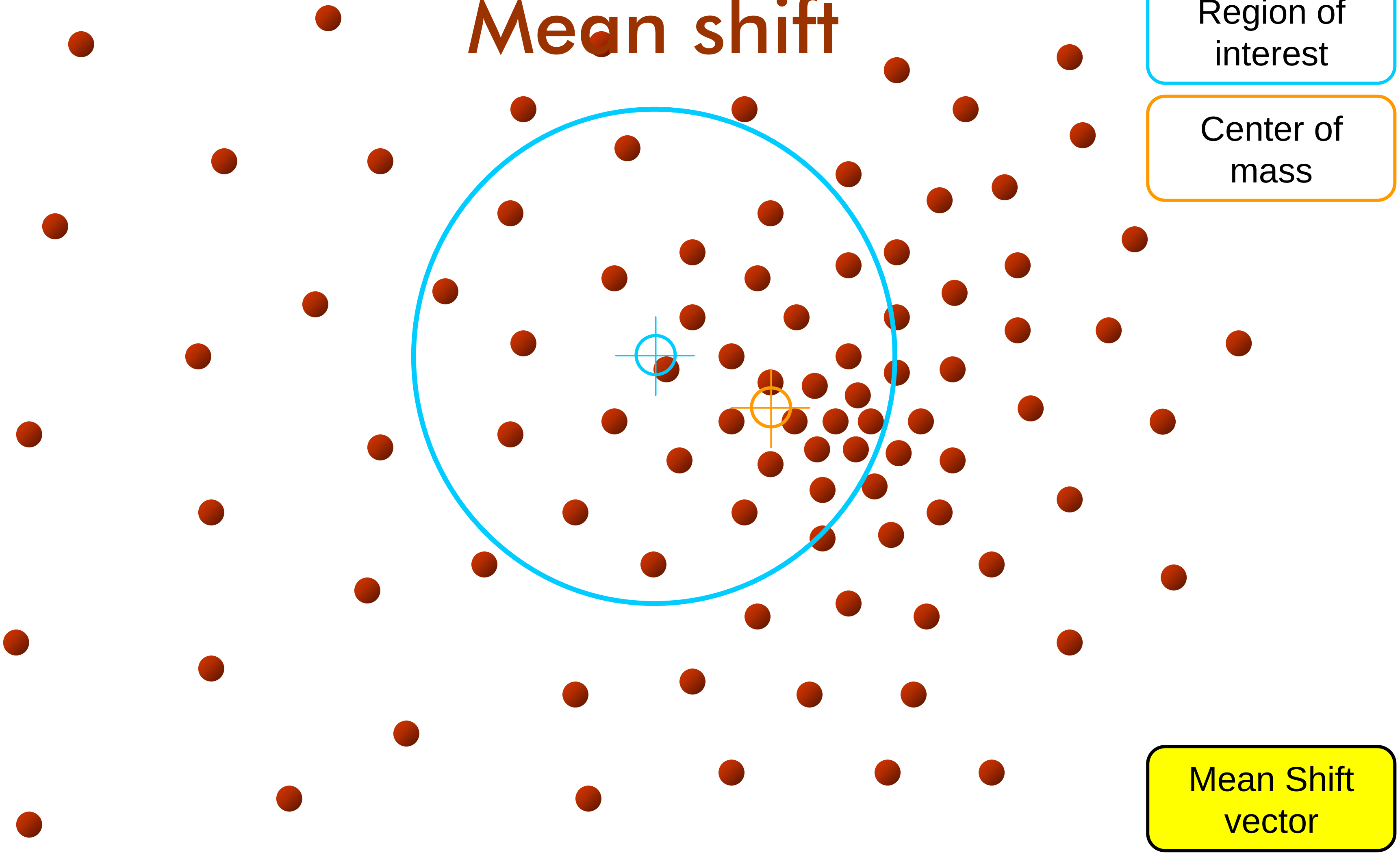
Slide by Y. Ukrainitz & B. Sarel

Mean shift

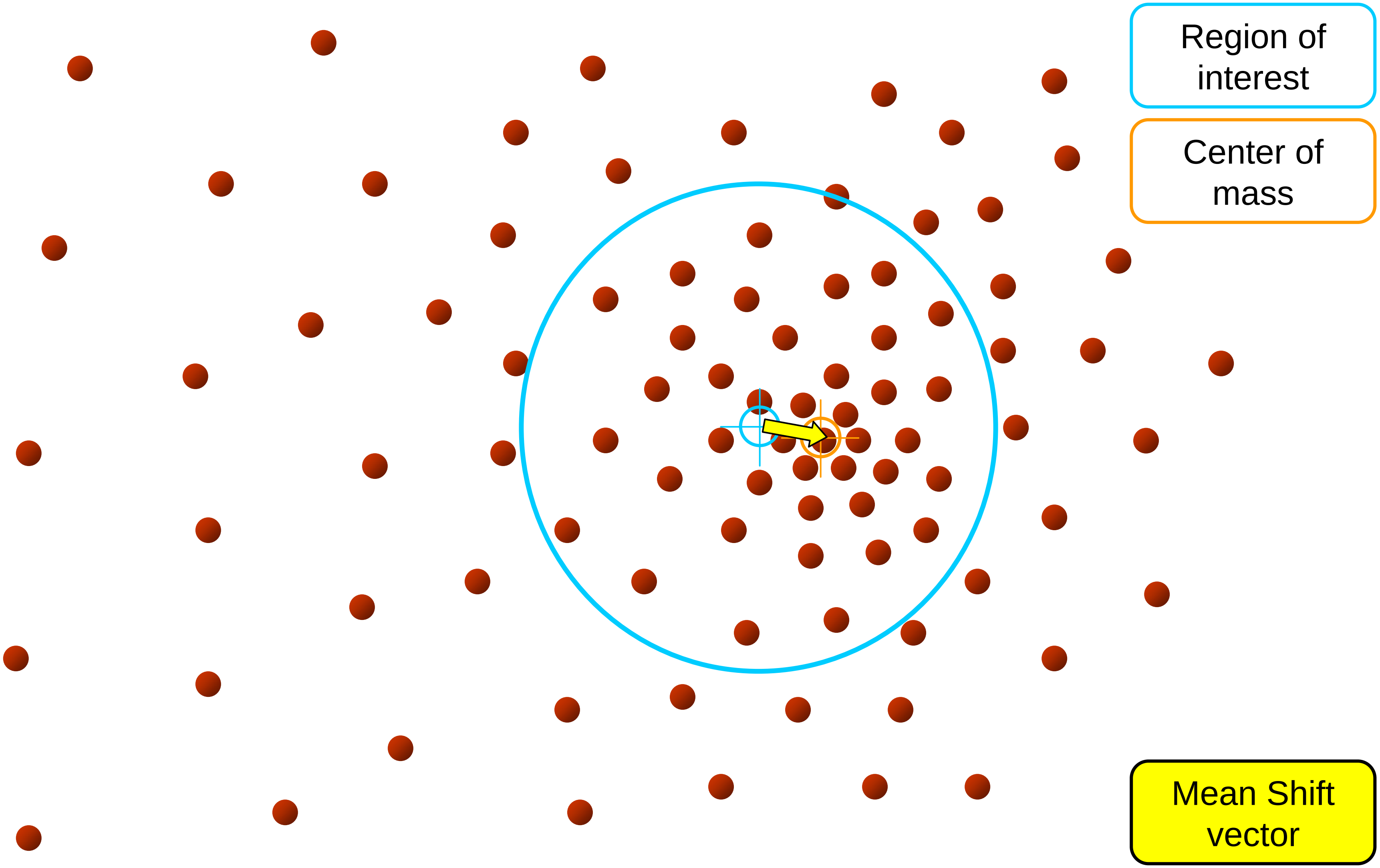


Slide by Y. Ukrainitz & B. Sarel

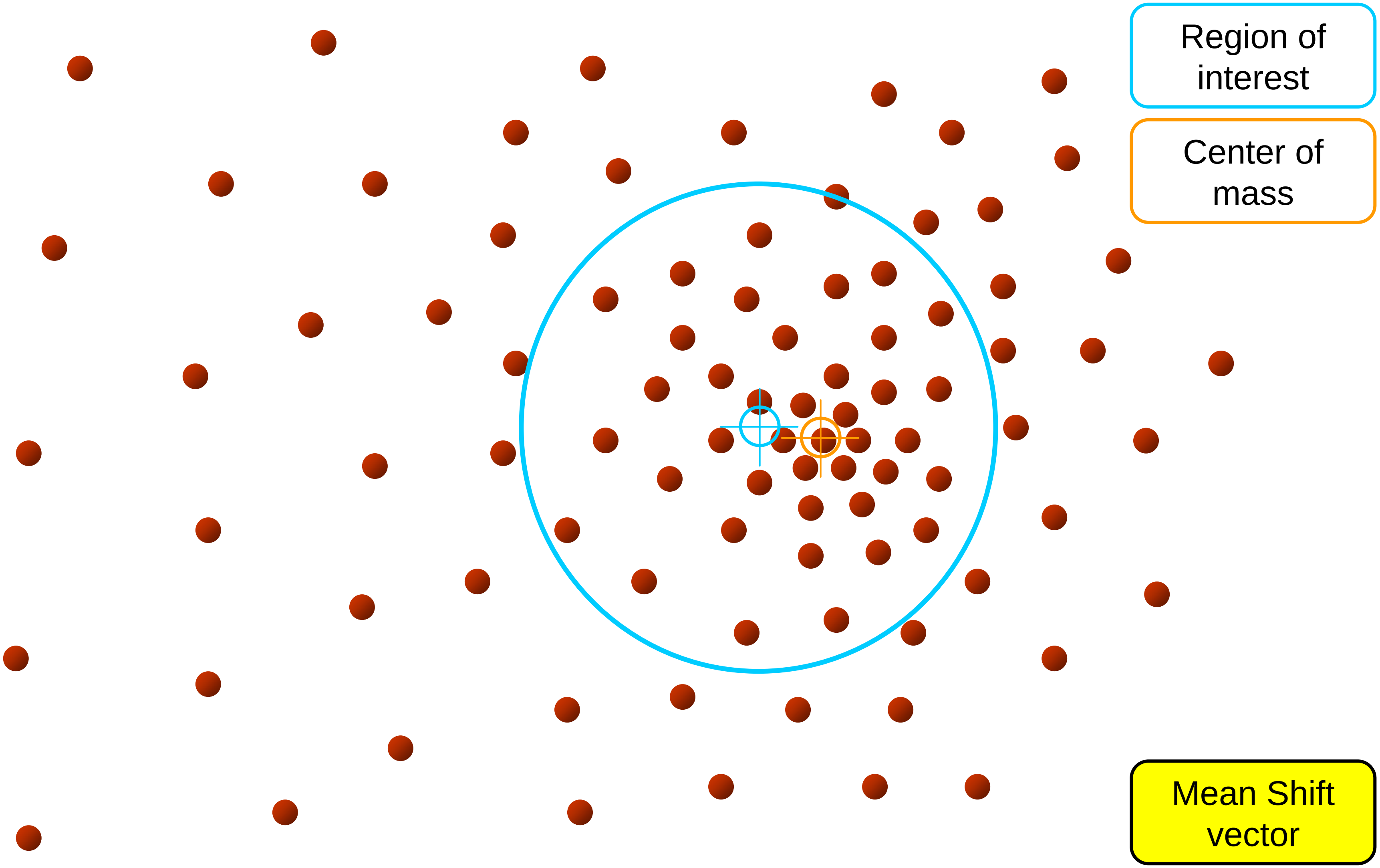
Mean shift



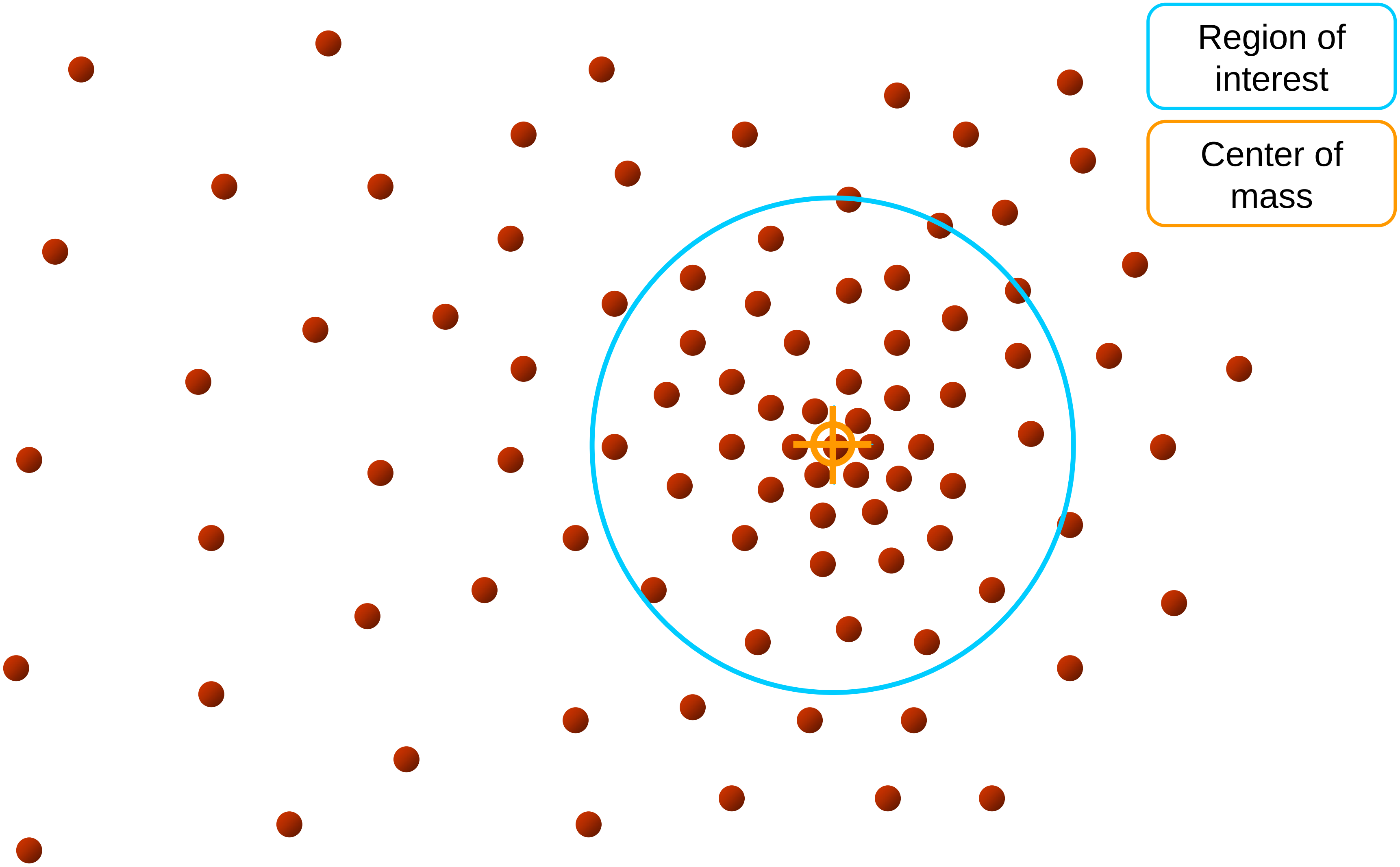
Slide by Y. Ukrainitz & B. Sarel



Slide by Y. Ukrainitz & B. Sarel



Slide by Y. Ukrainitz & B. Sarel



Slide by Y. Ukrainitz & B. Sarel

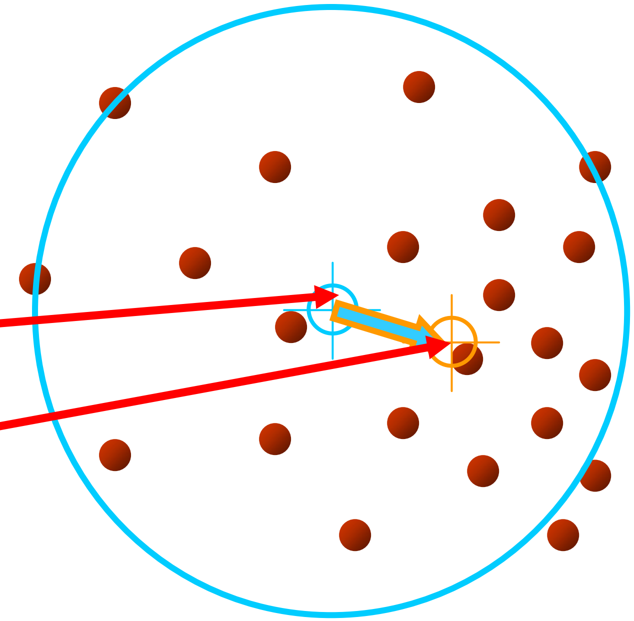
Computing The Mean Shift

<u>Simple Mean Shift procedure</u>:

- Compute mean shift vector

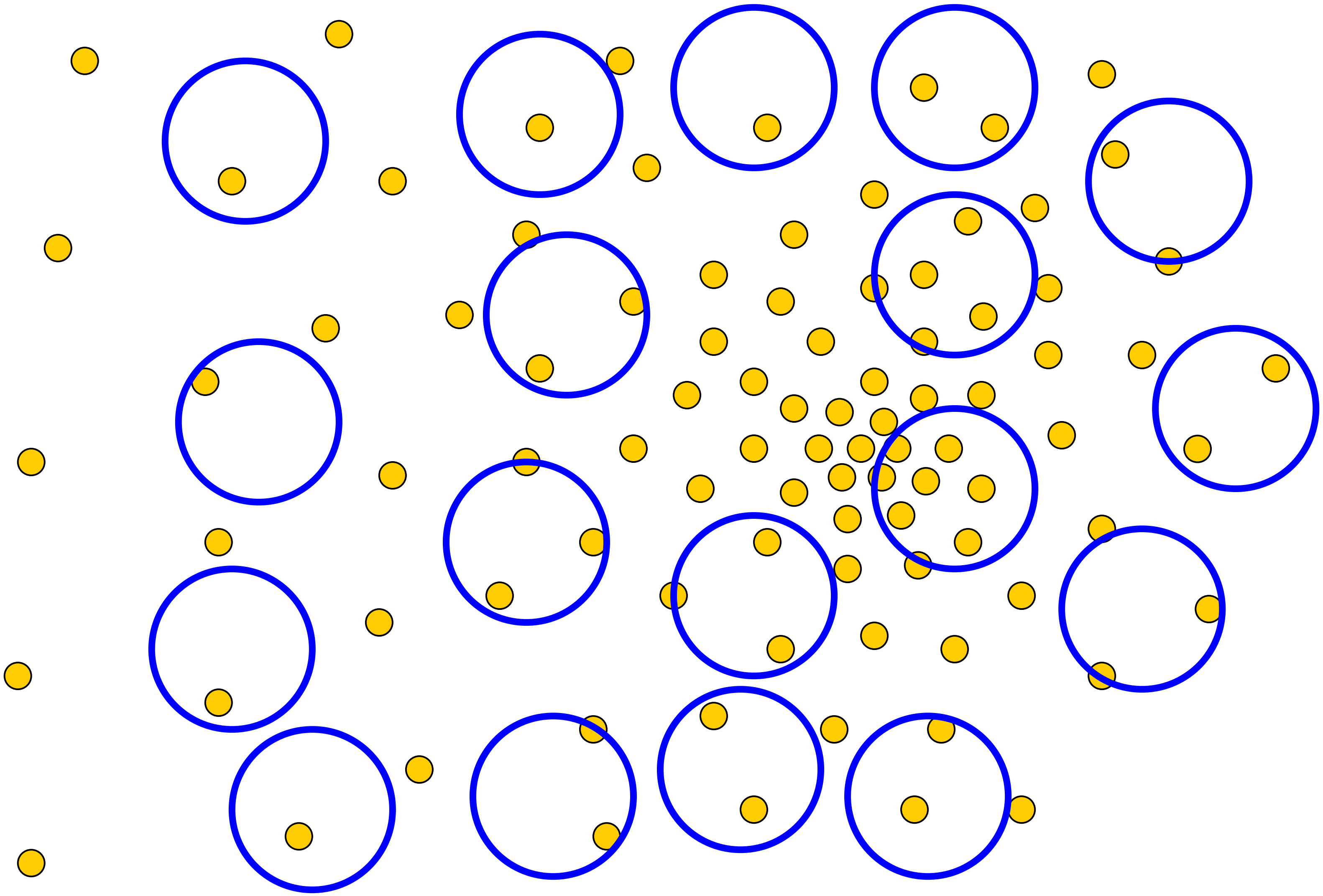
- Translate the Kernel window by m(x)

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



$$\mathrm{g}(\mathbf{x}) = -k'(\mathbf{x})$$

Real Modality Analysis

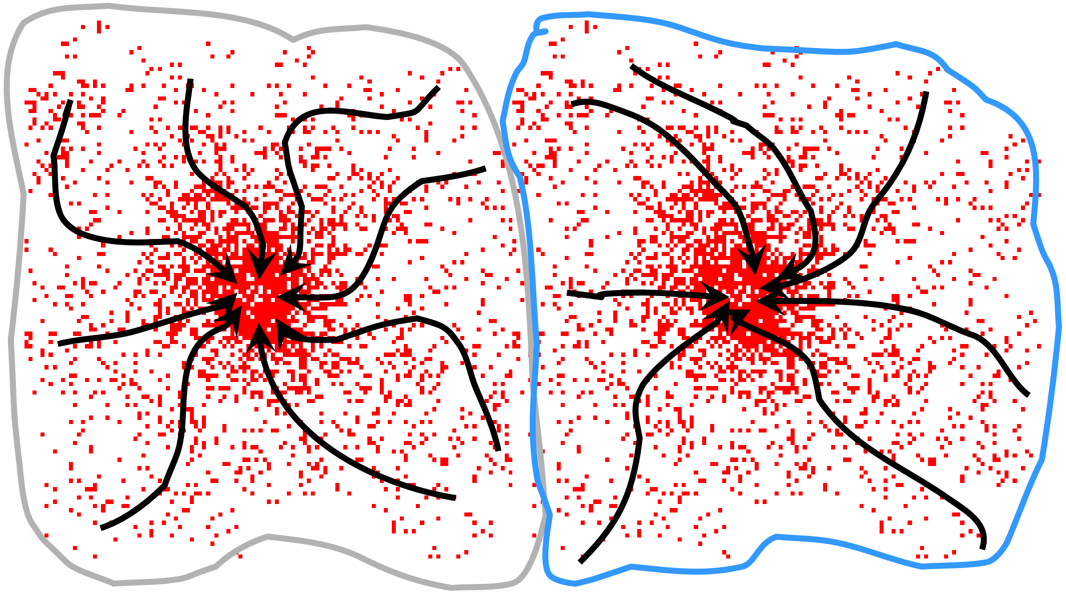


- Tessellate the space with windows

- Merge windows that end up near the same "peak" or model

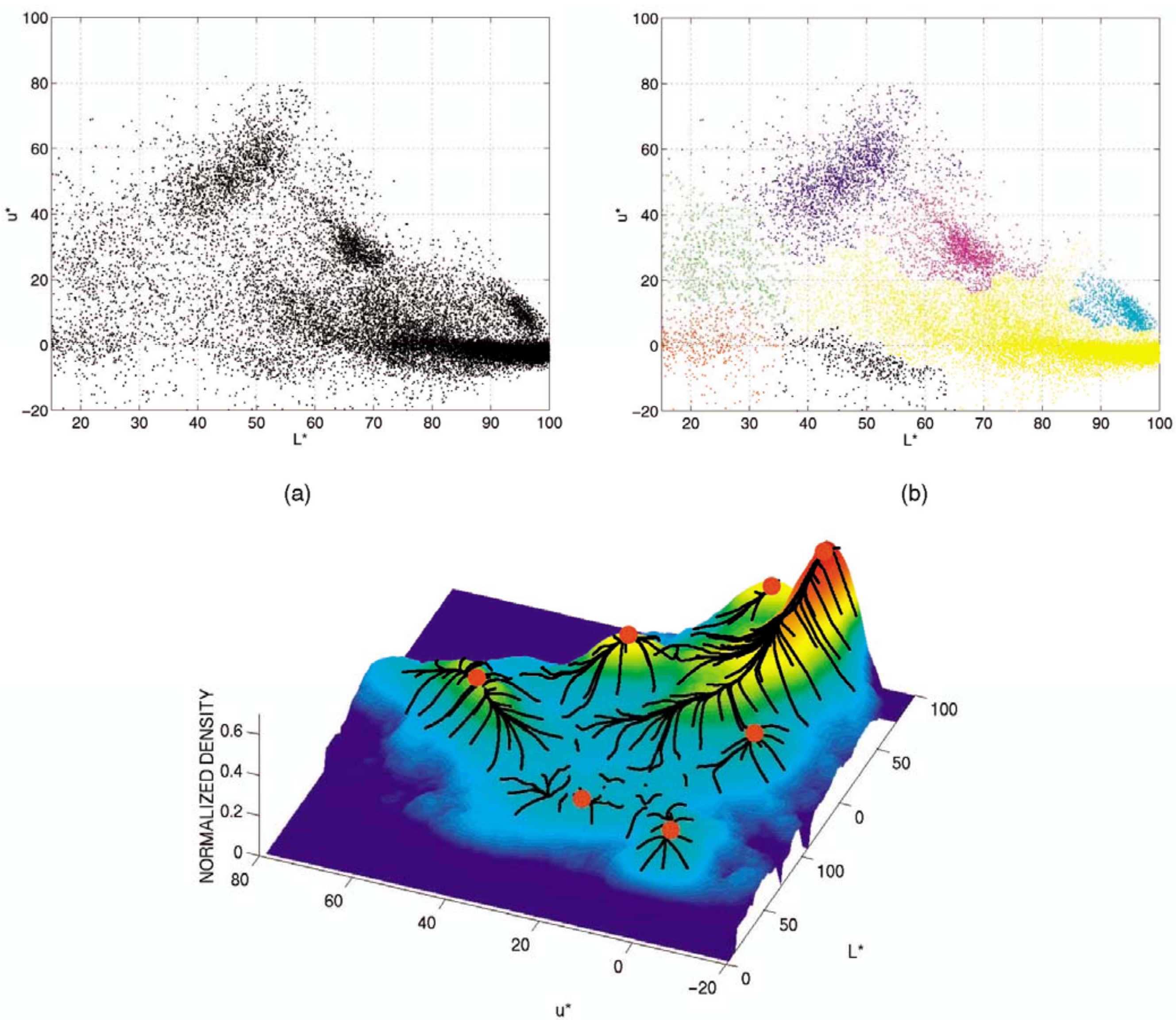
Attraction basin

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



Slide by Y. Ukrainitz & B. Sarel

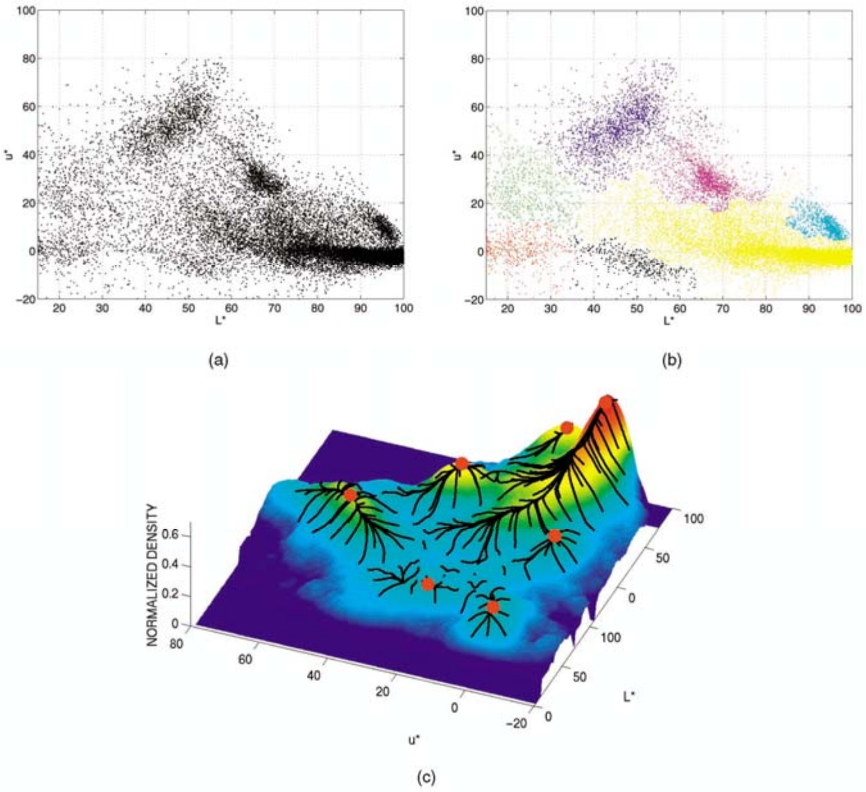
Attraction basin



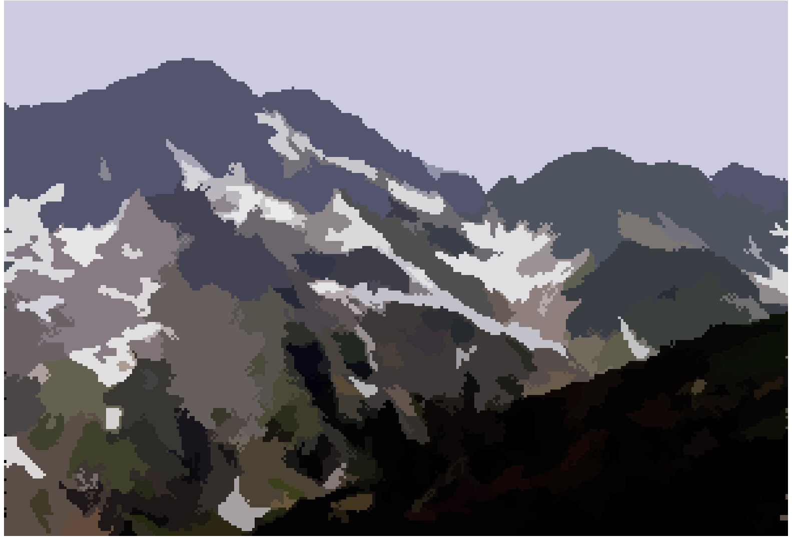
Slide by Y. Ukrainitz & B. Sarel

Segmentation by Mean Shift

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



Mean shift segmentation results



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

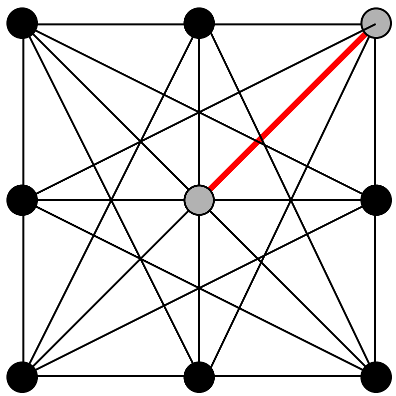
Mean shift pros and cons

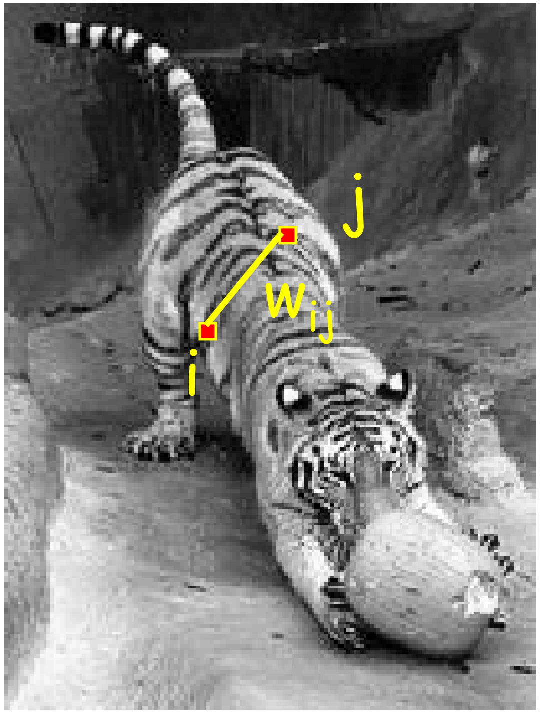
- Pros
  - Does not assume spherical clusters
  - Just a single parameter (window size)
  - Finds variable number of modes
  - Robust to outliers
- Cons
  - Output depends on window size
  - Computationally expensive
  - Does not scale well with dimension of feature space

Graph-based segmentation

- Represent features and their relationships using a graph
- Cut the graph to get subgraphs with strong interior links and weaker exterior links

Images as graphs





- Node for every pixel
- Edge between every pair of pixels
- Each edge is weighted by the *affinity* or similarity of the two nodes

Source: S. Seitz

Measuring Affinity

Distance

$$aff(x,y) = \exp\left\{-\left(\frac{1}{2\sigma_d^2}\right)\left(\|x-y\|^2\right)\right\}$$

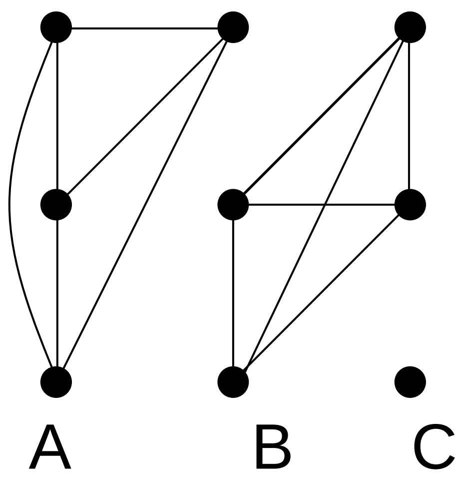
Intensity

$$aff(x,y) = \exp\left\{-\left(\frac{1}{2\sigma_i^2}\right)\left(\|I(x)-I(y)\|^2\right)\right\}$$

Color

$$aff(x,y) = \exp\left\{-\left(\frac{1}{2\sigma_t^2}\right)\left(\|c(x)-c(y)\|^2\right)\right\}$$

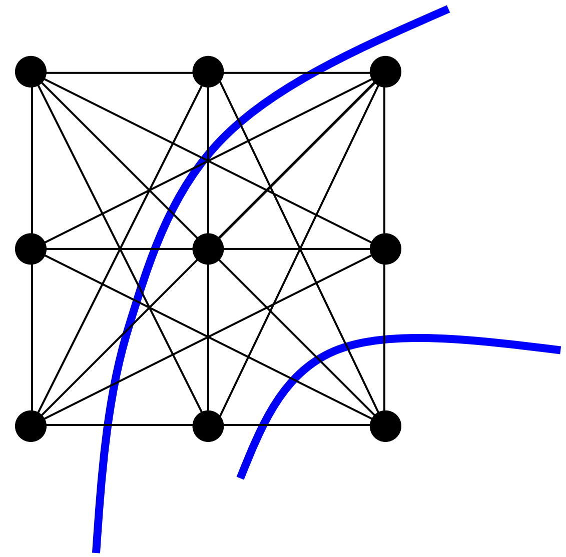
Segmentation by graph partitioning



- Break Graph into sub-graphs
  - Break links (cutting) that have low affinity
    - similar pixels should be in the same sub-graphs
    - dissimilar pixels should be in different sub-graphs
- Sub-graphs represents different image segments

Source: S. Seitz

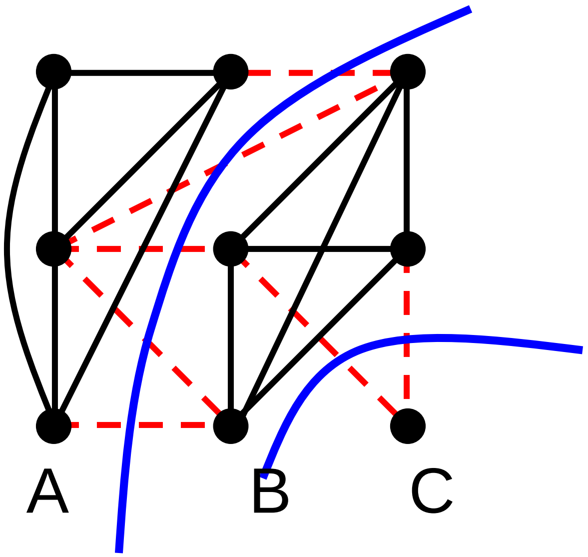
Graph cut



- CUT: Set of edges whose removal makes a graph disconnected
- Cost of a cut: sum of weights of cut edges
- A graph cut gives us a segmentation
  - What is a "good" graph cut and how do we find one?

Source: S. Seitz

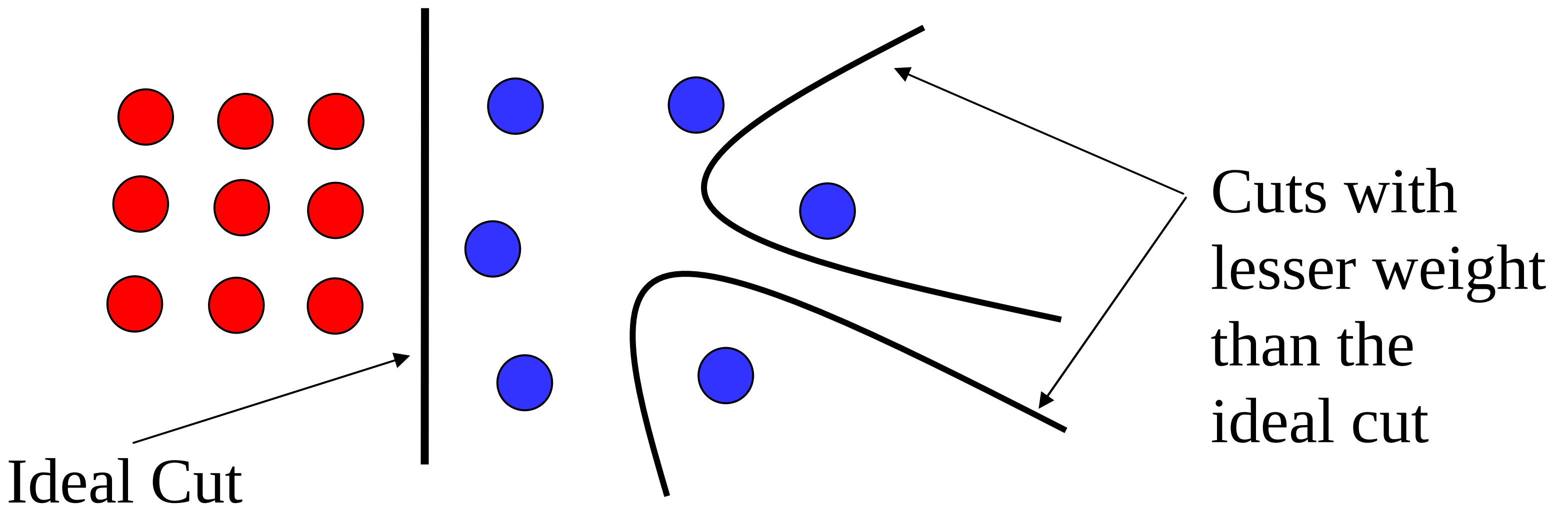
Graph cut



- CUT: Set of edges whose removal makes a graph disconnected
- Cost of a cut: sum of weights of cut edges
- A graph cut gives us a segmentation
  - What is a "good" graph cut and how do we find one?

Minimum cut

- We can do segmentation by finding the *minimum cut* in a graph
  - Efficient algorithms exist for doing this
- Drawback: minimum cut tends to cut off very small, isolated components



Normalized cut

J. Shi and J. Malik. Normalized cuts and image segmentation. PAMI 2000

- IDEA: normalizing the cut by component size
- The *normalized cut* cost is:

$$\frac{cut(A,B)}{assoc(A,V)} + \frac{cut(A,B)}{assoc(B,V)}$$

assoc(*A*, *V*) = sum of weights of all edges in V that touch A

- The exact solution is NP-hard but an approximation can be computed by solving a *generalized eigenvalue* problem

Normalized cuts: Pro and con

- Pros
  - Generic framework, can be used with many different features and affinity formulations
- Cons
  - High storage requirement and time complexity
  - Bias towards partitioning into equal segments

* Slide from Khurram Hassan-Shafique CAP5415 Computer Vision 2003

Normalized cuts: Results

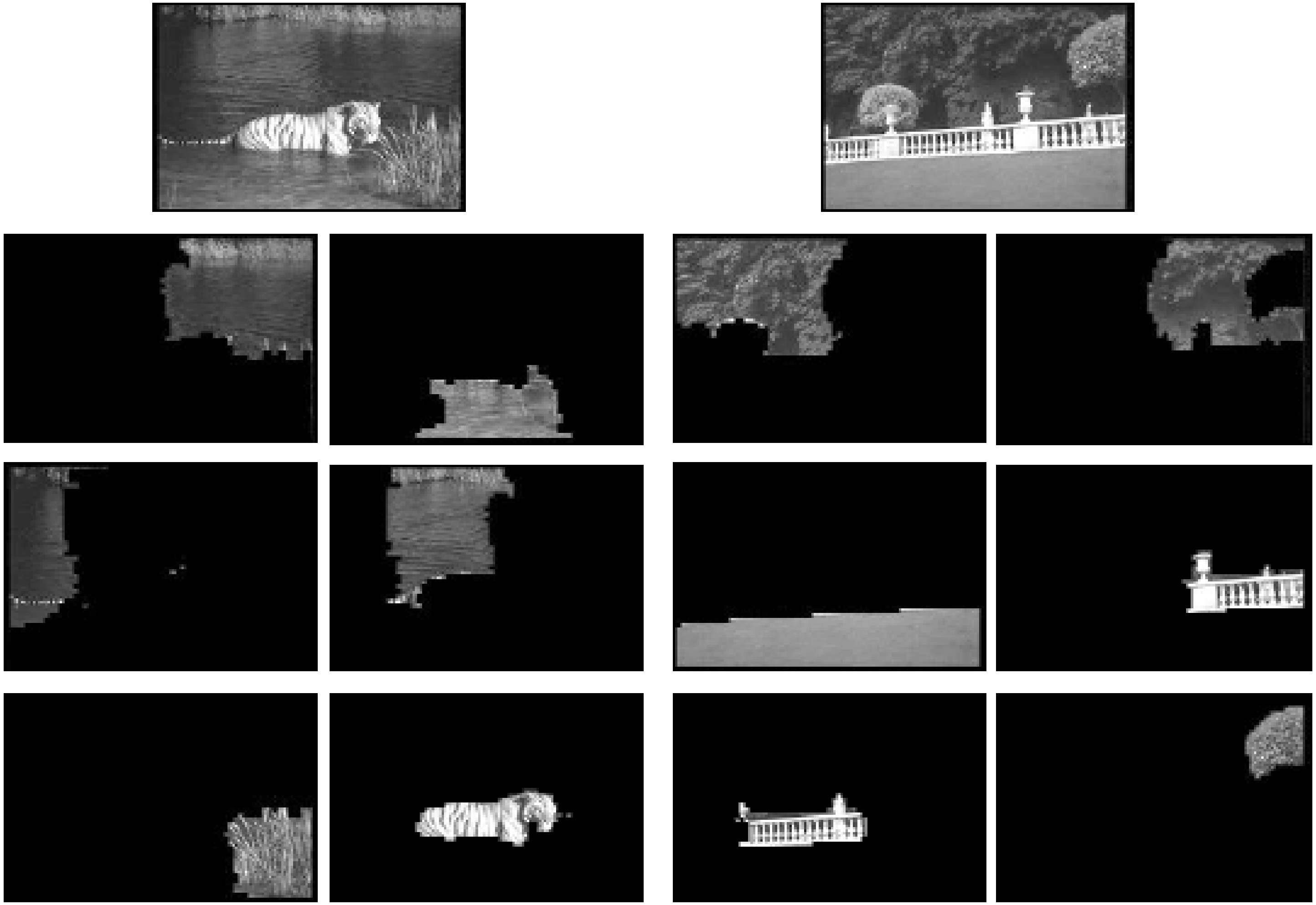


Figure from “Image and video segmentation: the normalised cut framework”, by Shi and Malik, copyright IEEE, 1998

Normalized cuts: Results

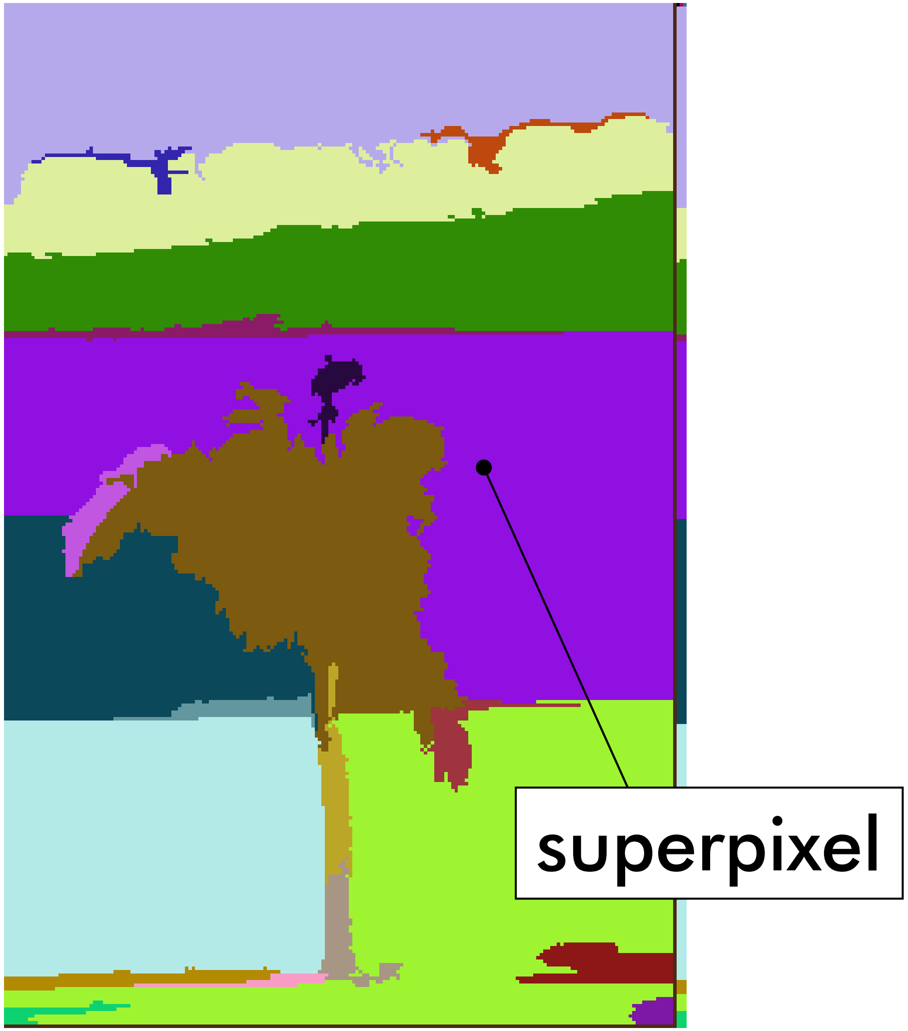
F igure from “Normalized cuts and image segmentation,” Shi and Malik, copyright IEEE, 2000

Contour and Texture Analysis for Image Segmentation

J. Malik, S. Belongie, T. Leung and J. Shi. "*Contour and Texture Analysis for Image Segmentation*". IJCV 43(1),7-27,2001.

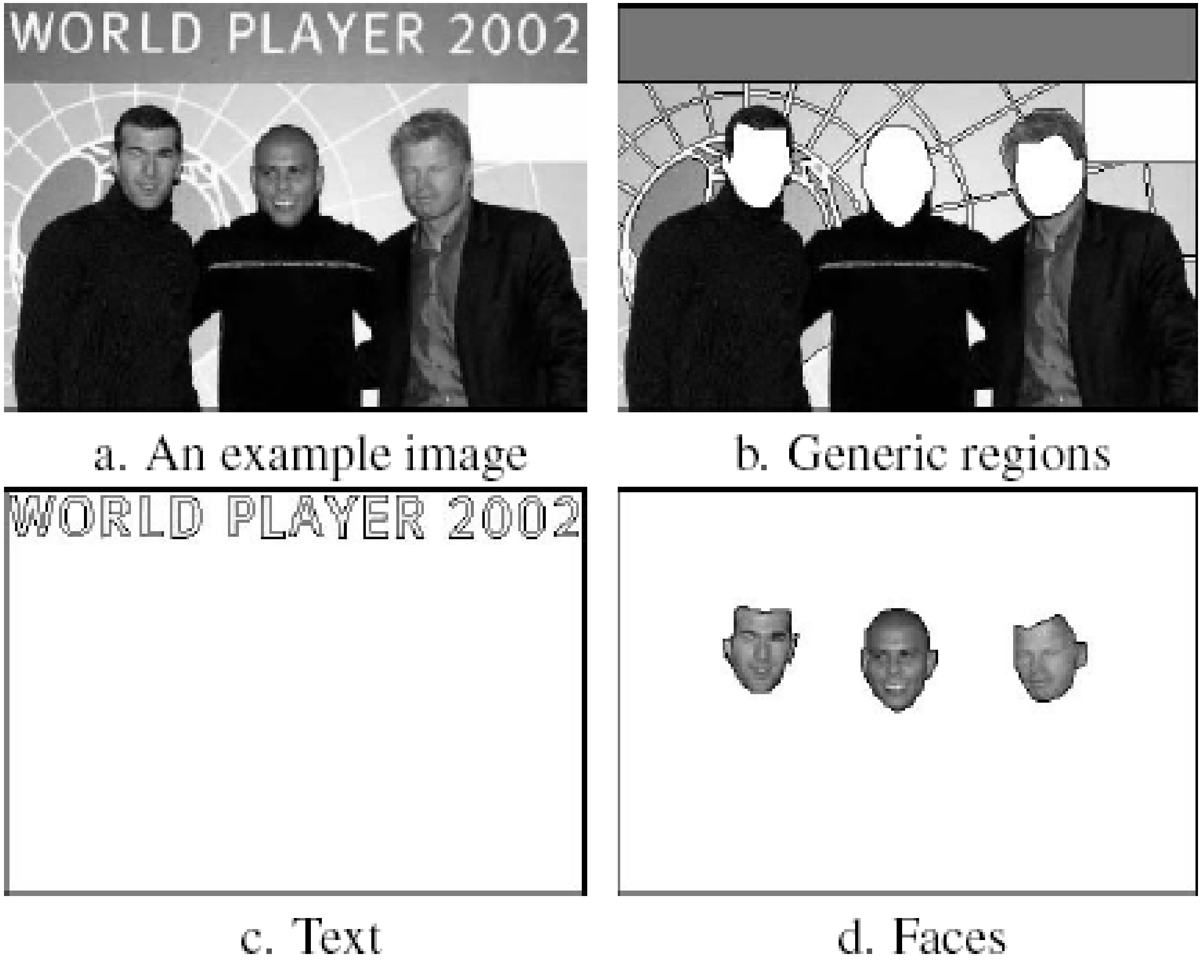
Efficient Graph-Based Image Segmentation

Efficient Graph-Based Image Segmentation Pedro F. Felzenszwalb and Daniel P. Huttenlocher
International Journal of Computer Vision, Volume 59, Number 2, September 2004



Integrating top-down and bottom-up segmentation

Z.W. Tu, X.R. Chen, A.L. Yuille, and S.C. Zhu. Image parsing: unifying segmentation, detection and recognition. IJCV 63(2), 113-140, 2005.



a. An example image
b. Generic regions
c. Text
d. Faces

EECS 442 – Computer vision

Object Recognition