CEE598 - Visual Sensing for Civil Infrastructure Eng. & Mgmt.

Session 11 – Segmentation And Clustering

Mani Golparvar-Fard
Department of Civil and Environmental Engineering
3129D, Newmark Civil Engineering Lab
e-mail: mgolpar@illinois.edu

Available at a web site near you…

- For most local feature detectors, executables are available online:
  - http://www.mathworks.com/matlabcentral/fileexchange/9272-harris-corner-detector
  - http://mi.eng.cam.ac.uk/~er258/work/fast.html
  - http://robots.ox.ac.uk/~vgg/research/affine
  - http://www.cs.ubc.ca/~lowe/keypoints/
  - http://www.cs.unc.edu/~ccwu/siftgpu/
  - http://www.mathworks.com/matlabcentral/fileexchange/18441
  - http://www.vision.ee.ethz.ch/~surf

Choosing a detector

- What do you want it for?
  - Precise localization in x-y: Harris
  - Good localization in scale: Difference of Gaussian
  - Flexible region shape: MSER (maximally stable extremal regions)

- Best choice often application dependent
  - Harris-/Hessian-Laplace/DoG work well for many natural categories
  - MSER can work well for buildings and printed things

- Why choose?
  - Get more points with more detectors

- There have been extensive evaluations/comparisons
  - [Mikolajczyk et al., IJCV'05, PAMI'05]
  - http://www.it.lut.fi/publications/files/publications/642/thesis.pdf
  - All detectors/descriptors shown here work well

Comparison of Keypoint Detectors

Table 7.1 Overview of feature detectors.

Feature Detector	Corner	Blob	Region	Rotation invariant	Scale invariant	Affine invariant	Localization Repeatability	Localization accuracy	Robustness	Efficiency
Harris	√			√			+++	+++	+++	++
Hessian		√		√			++	++	++	+
SUSAN	√			√			++	++	++	+++
Harris-Laplace	√	(√)		√	√		+++	+++	++	+
Hessian-Laplace	(√)	√		√	√		+++	+++	+++	+
DoG	(√)	√		√	√		++	++	++	++
SURF	(√)	√		√	√		++	++	++	+++
Harris-Affine	√	(√)		√	√	√	+++	+++	++	++
Hessian-Affine	(√)	√		√	√	√	+++	+++	+++	++
Salient Regions	(√)	√		√	√	(√)	+	+	++	+
Edge-based	√			√	√	√	+++	+++	+	+
MSER			√	√	√	√	+++	+++	++	+++
Intensity-based			√	√	√	√	++	++	++	++
Superpixels			√	√	(√)	(√)	+	+	+	+

http://homes.esat.kuleuven.be/~tuytelaa/FT_survey_interestpoints08.pdf

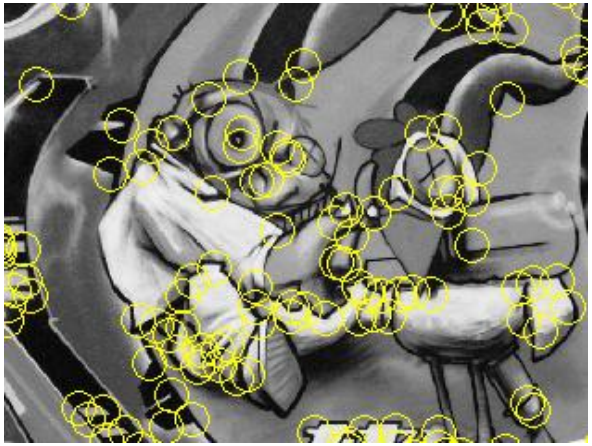
Tuytelaars Mikolajczyk 2008

Choosing a descriptor

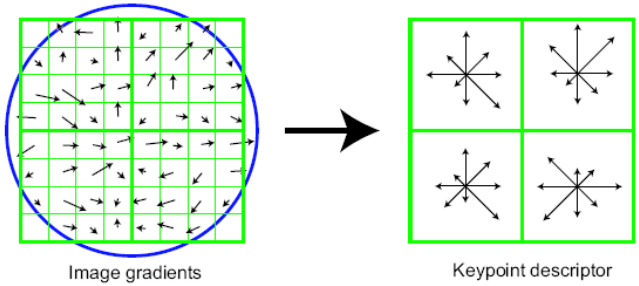
- Again, need not stick to one
- For object instance recognition or stitching, SIFT or variant is a good choice

Things to remember

- Keypoint detection: repeatable and distinctive
  - Corners, blobs, stable regions
  - Harris, DoG



- Descriptors: robust and selective
  - Spatial histograms of orientation
  - SIFT

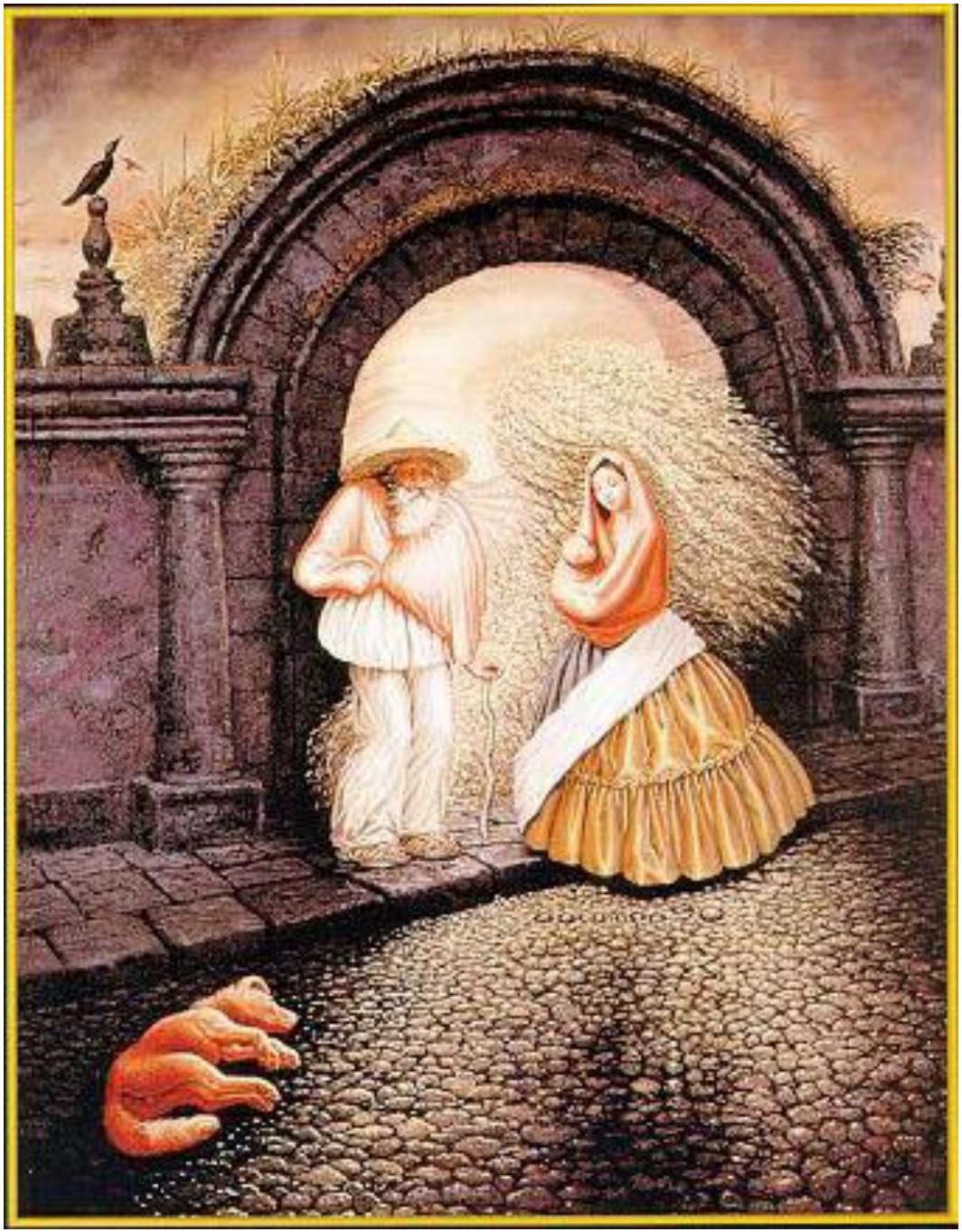


Outline

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

Reading: Chapter 14 [FP]

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



Segmentation and Clustering

- 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

- Clustering
  - Group together similar points and represent them with a single token
    - Token: whatever we need to group (pixels, points, surface elements, etc.)

Segmentation and Clustering

Key Challenges:

1. What makes two points/images/patches similar?

2. How do we compute an overall grouping from pairwise similarities?

Why do we cluster?

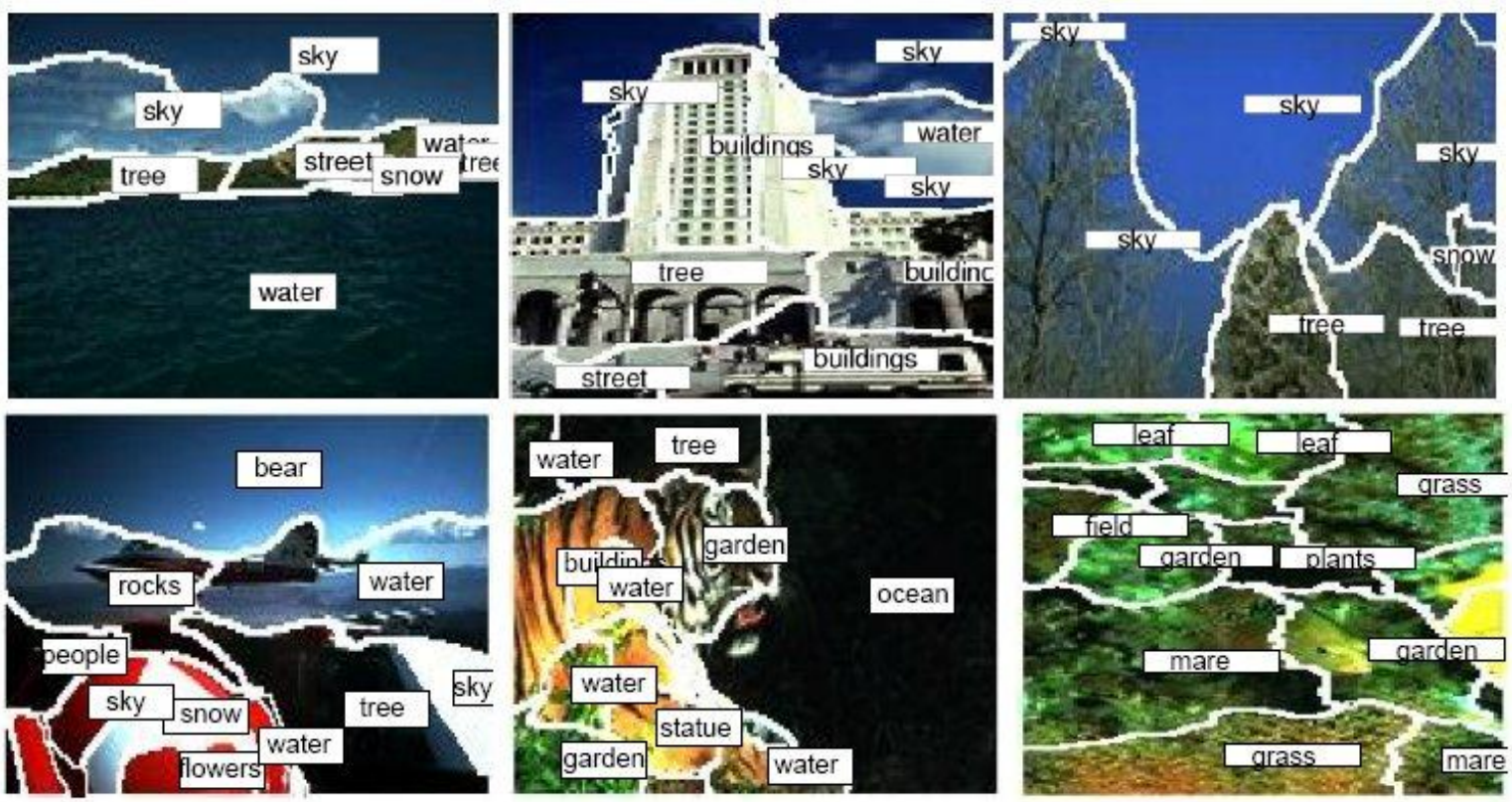
- **Summarizing data**
  - Look at large amounts of data
  - Patch-based compression or denoising
  - Represent a large continuous vector with the cluster number
- **Counting**
  - Histograms of texture, color, SIFT vectors
- **Segmentation**
  - Separate the image into different regions
- **Prediction**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

Segmentation in Computer Vision

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

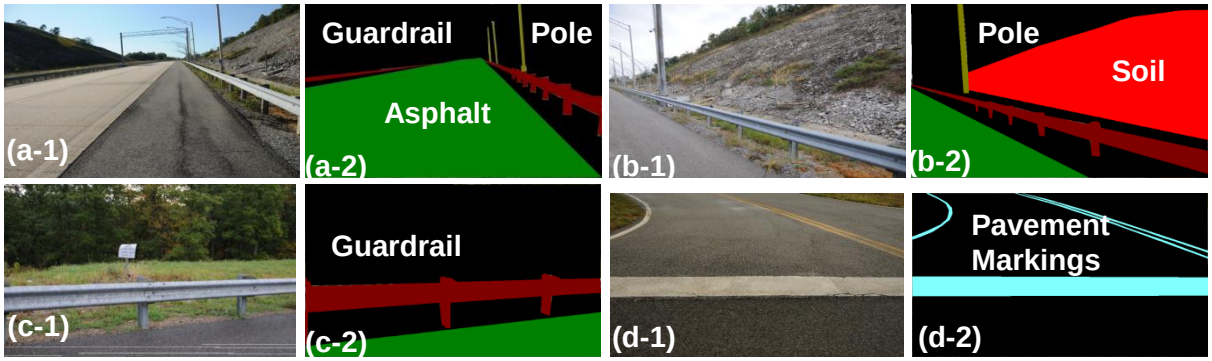
Segmentation in Computer Vision

- B. Duygulu, de Freitas, D. Forsyth (2002). "Object Recognition as Machine Translation". *ECCV*

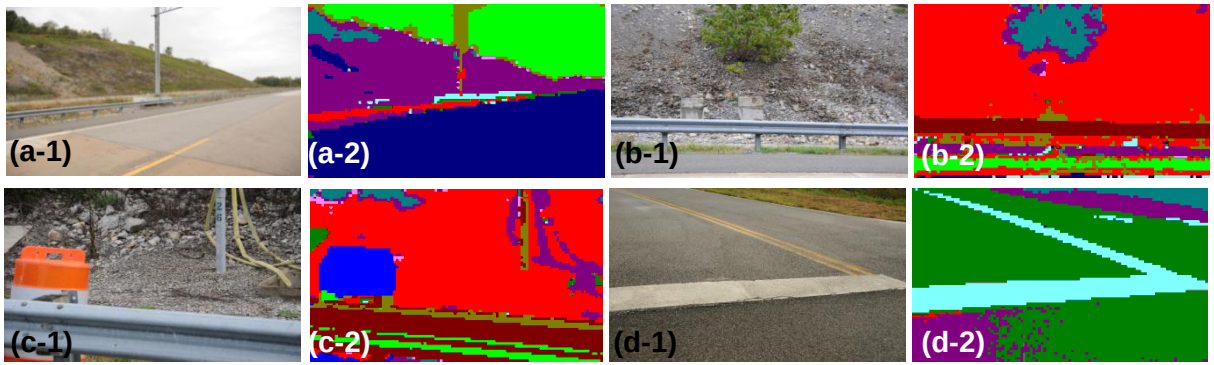


Segmentation in Civil Engineering

- B. Uslu, M. Golparvar-Fard, and J.M. de la Garza (2011). "Image-based 3D reconstruction and Recognition for Enhanced Highway Condition Assessment." *Proc., ASCE International Workshop on Computing in Civil Engineering*



Supervised segmentation of the ground truth images



The segmentation and asset recognition results

How do we cluster?

- K-means
  - Iteratively re-assign points to the nearest cluster center
- Agglomerative clustering
  - Start with each point as its own cluster and iteratively merge the closest clusters
- Spectral clustering
  - Split the nodes in a graph based on assigned links with similarity weights

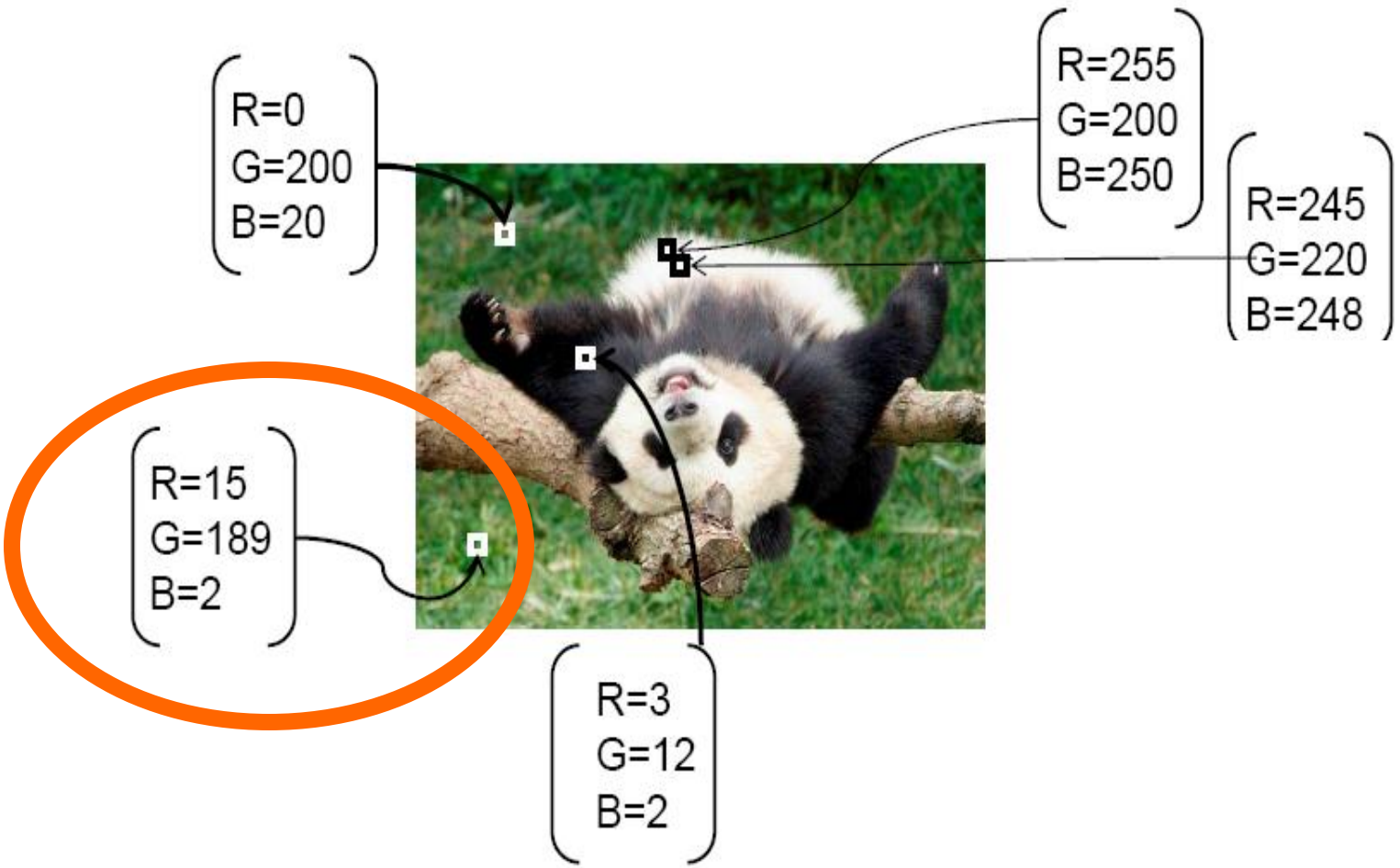
Feature Space

- 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)

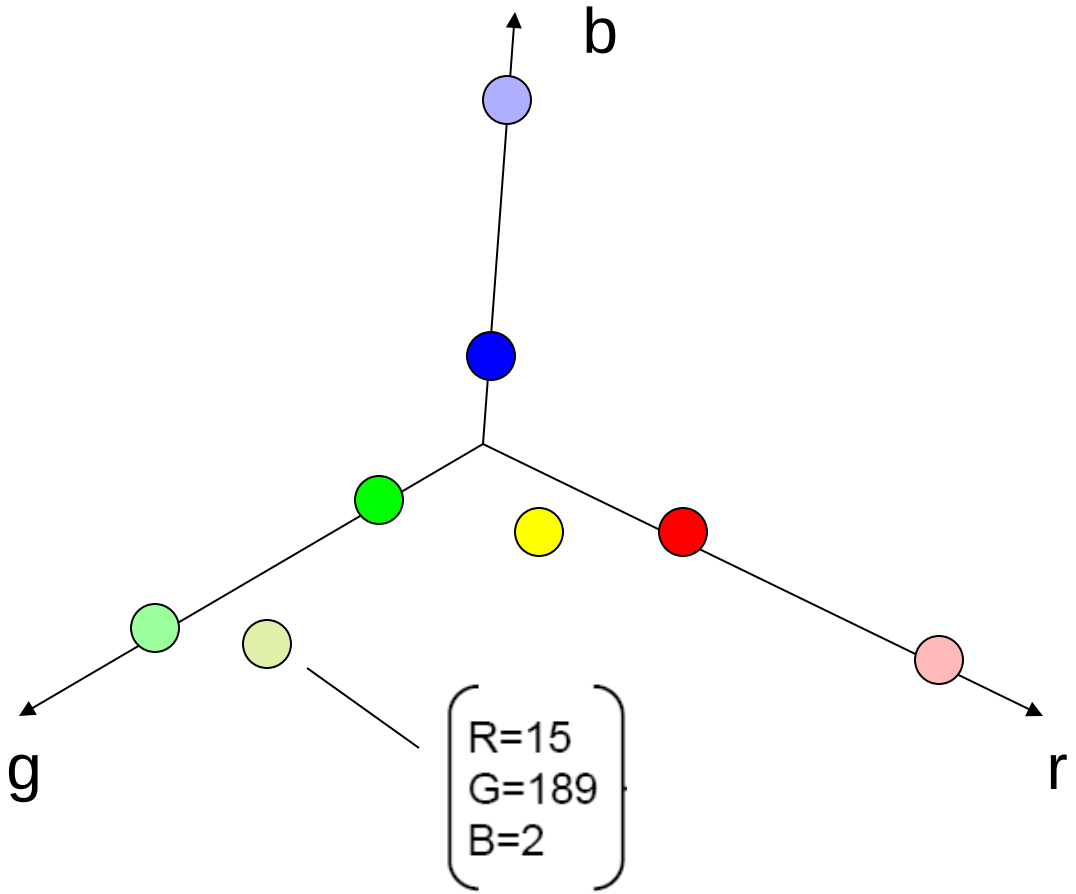
Slide credit: Christopher Rasmussen

Feature Space

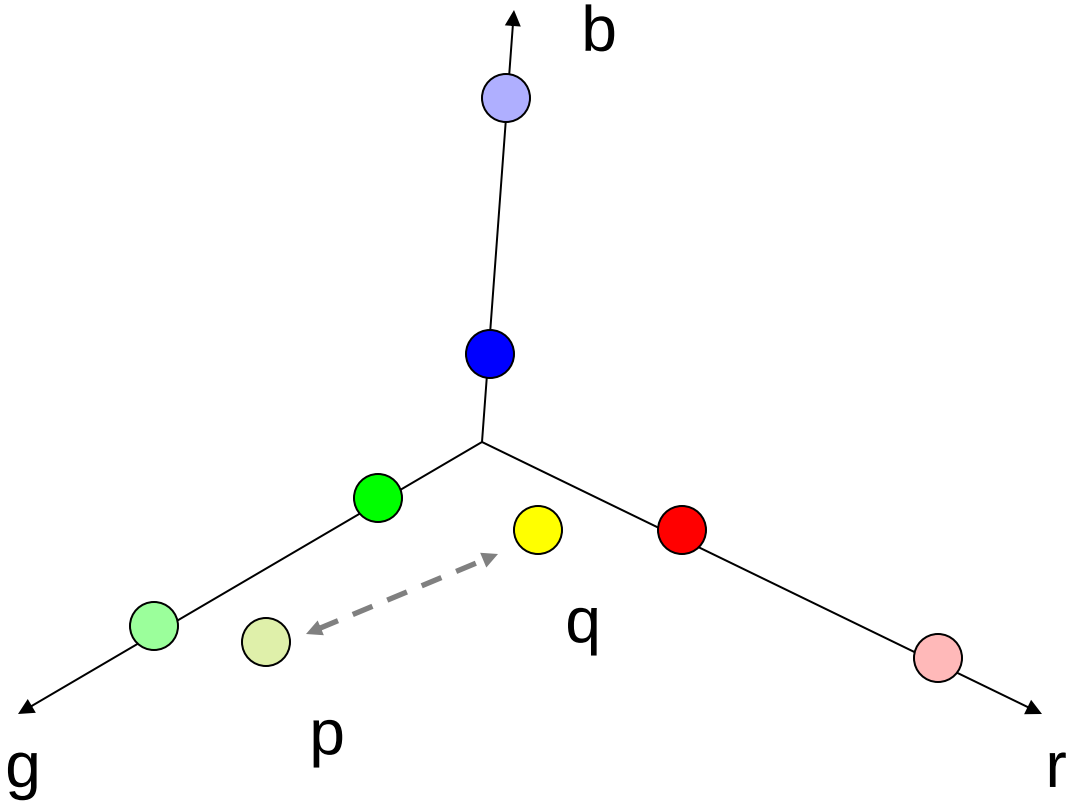


Source: K. Grauman

Feature space:
each token is represented by a point

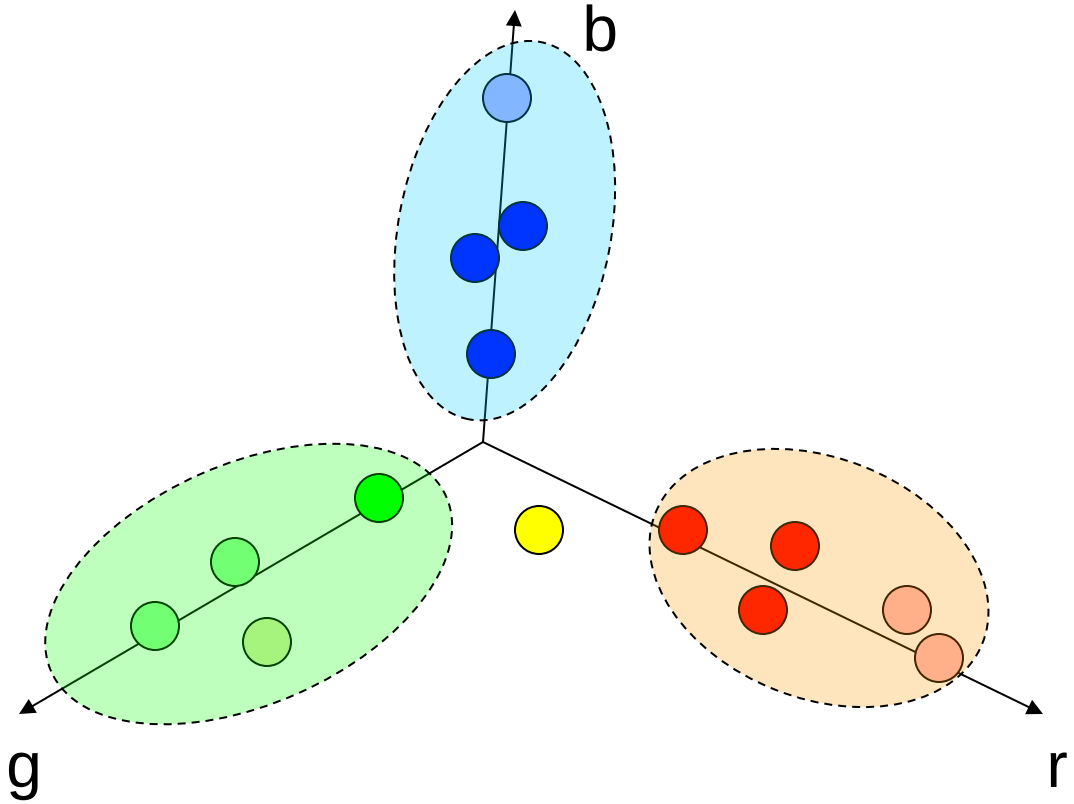


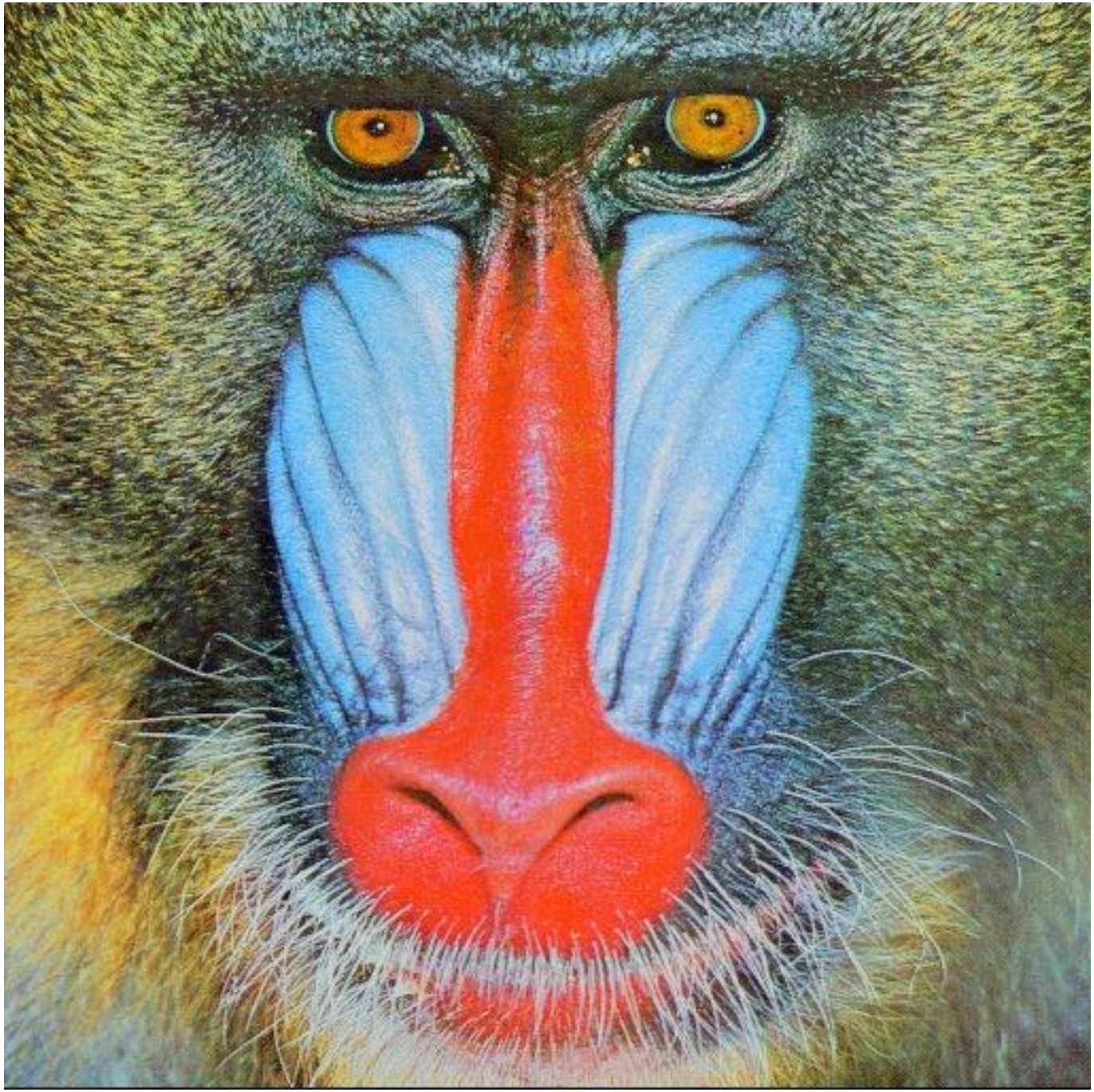
Token similarity is thus measured by distance between points ("feature vectors") in feature space

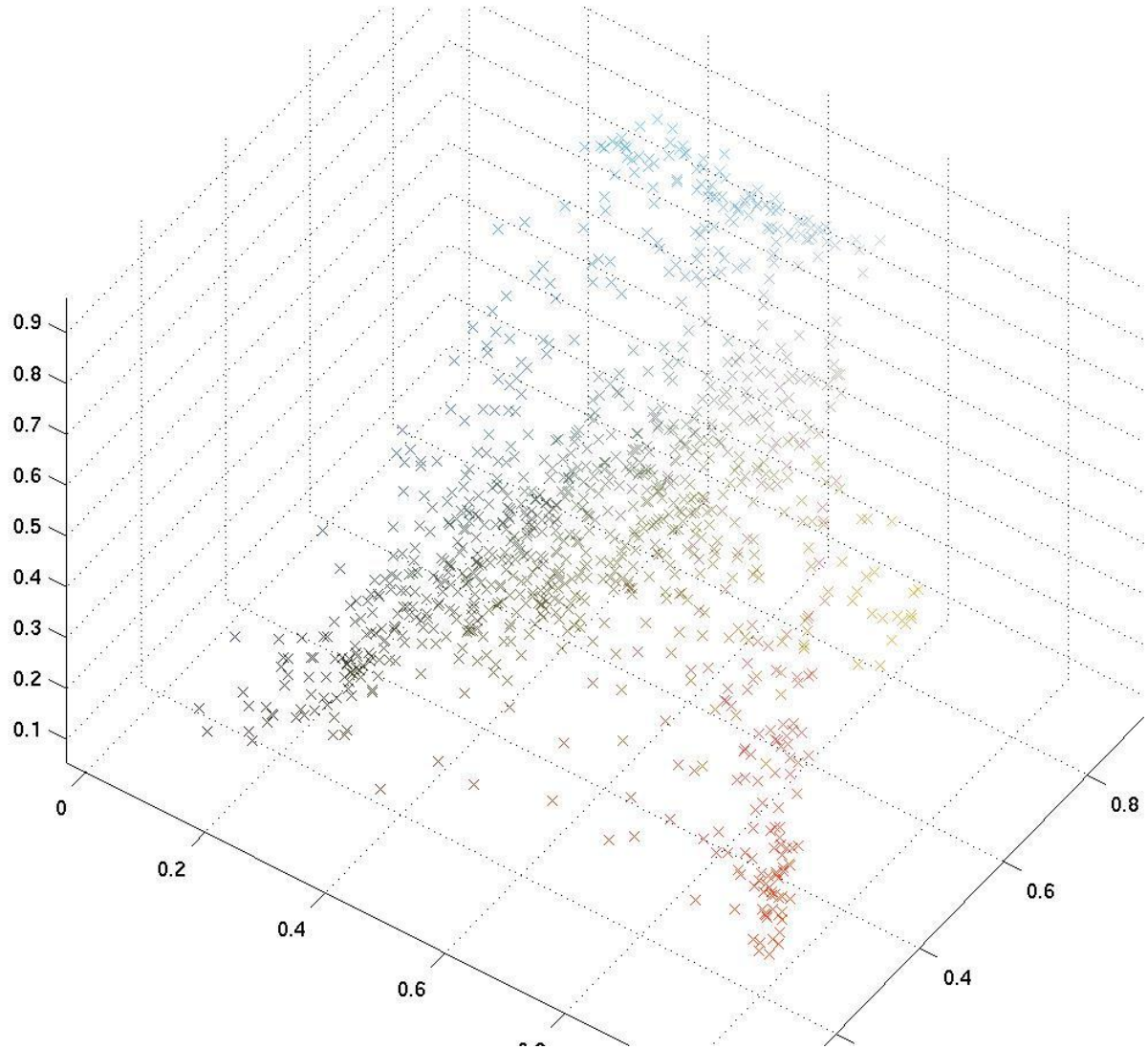


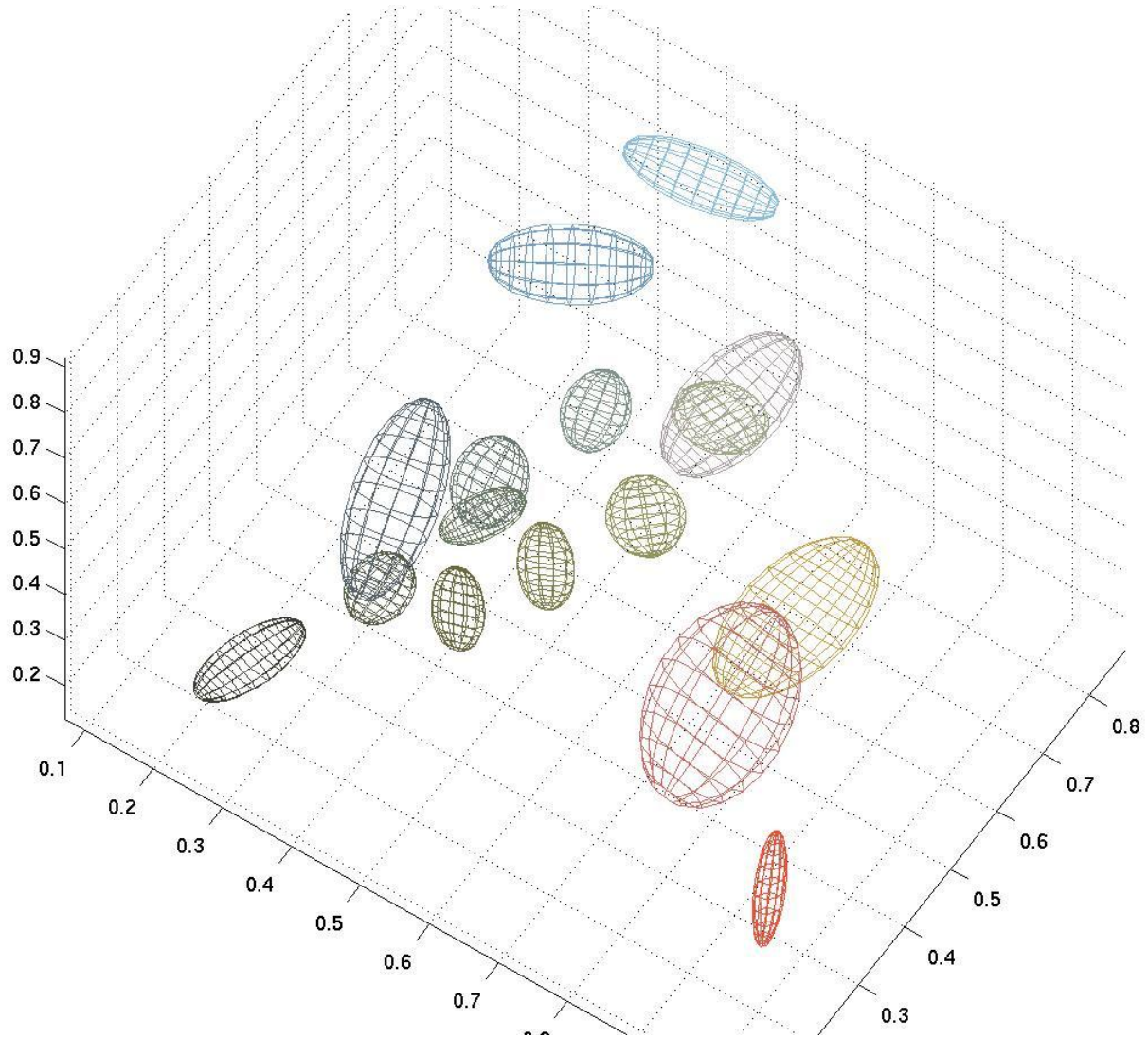
$$\sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \cdots + (p_n - q_n)^2} = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2}$$

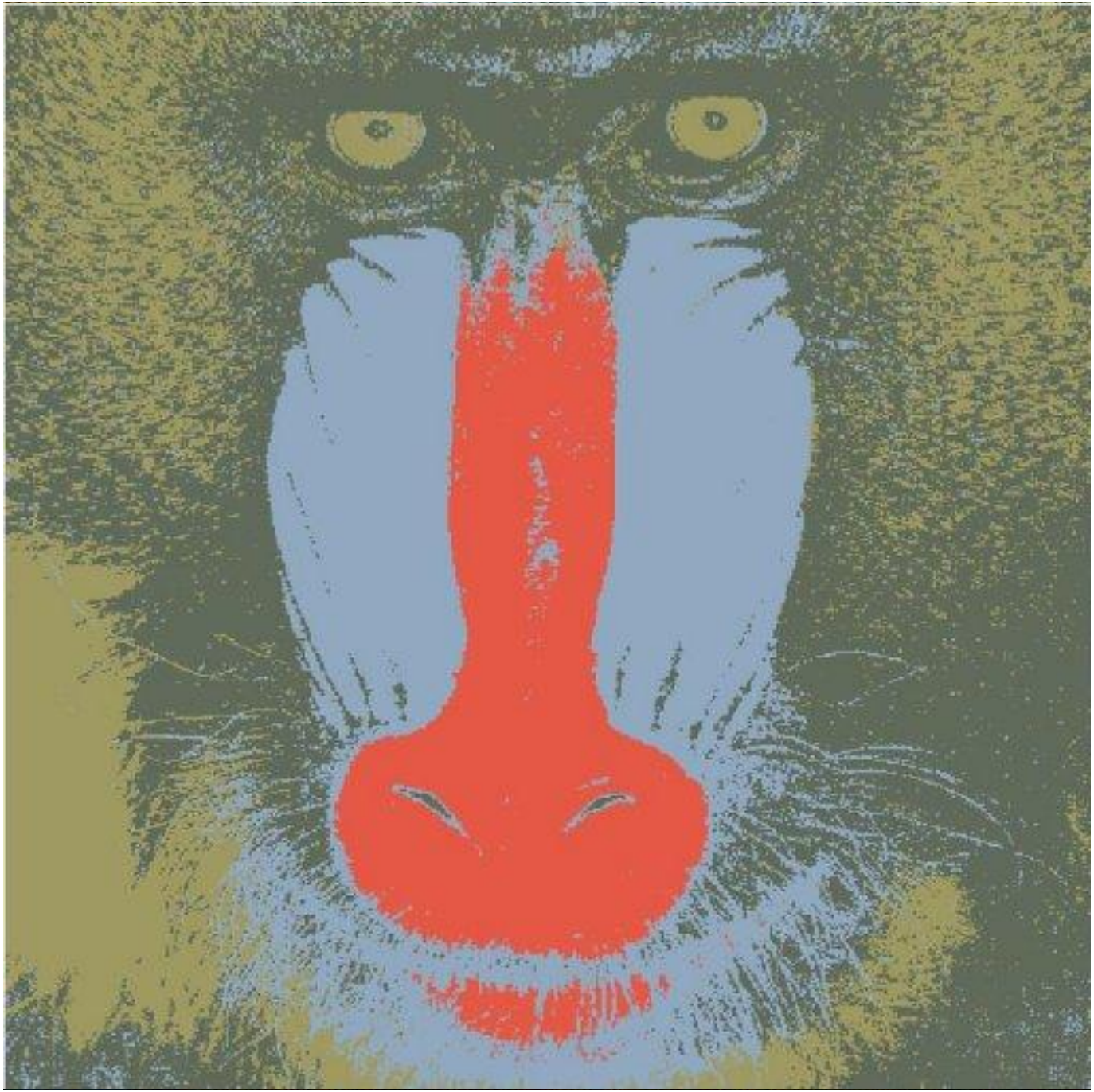
Cluster together tokens with high similarity

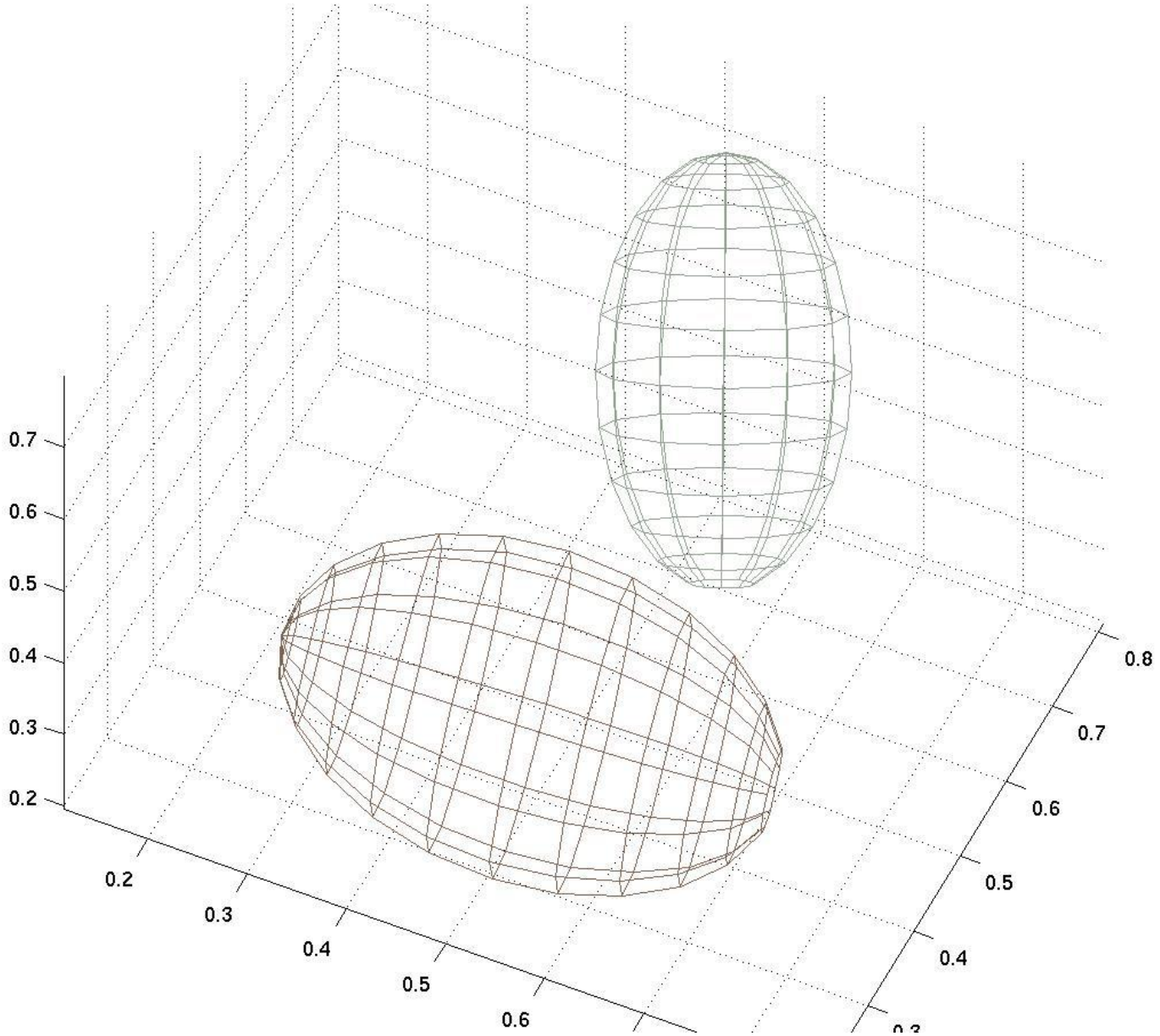





0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0.2
0.4
0.6
0.8
0.6
0.4



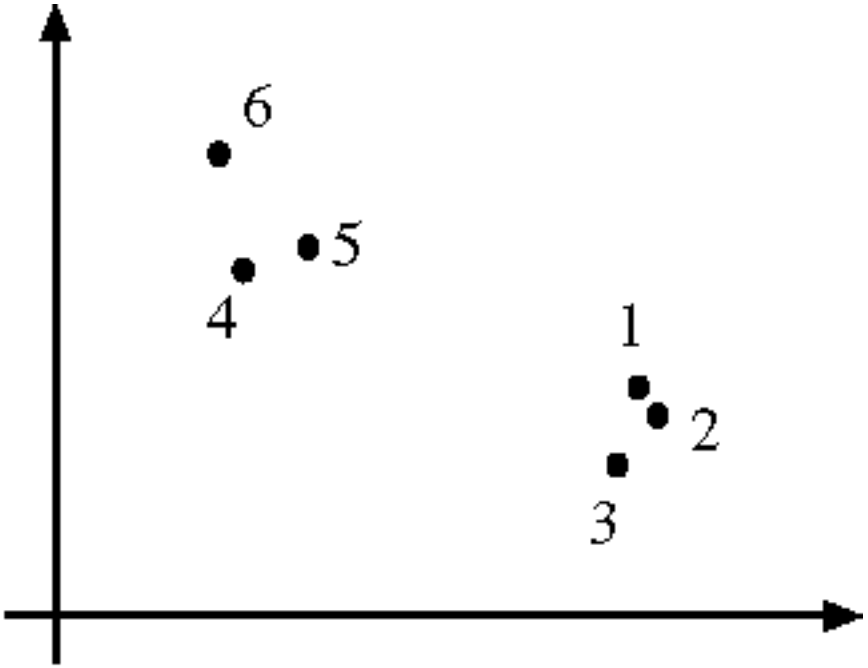


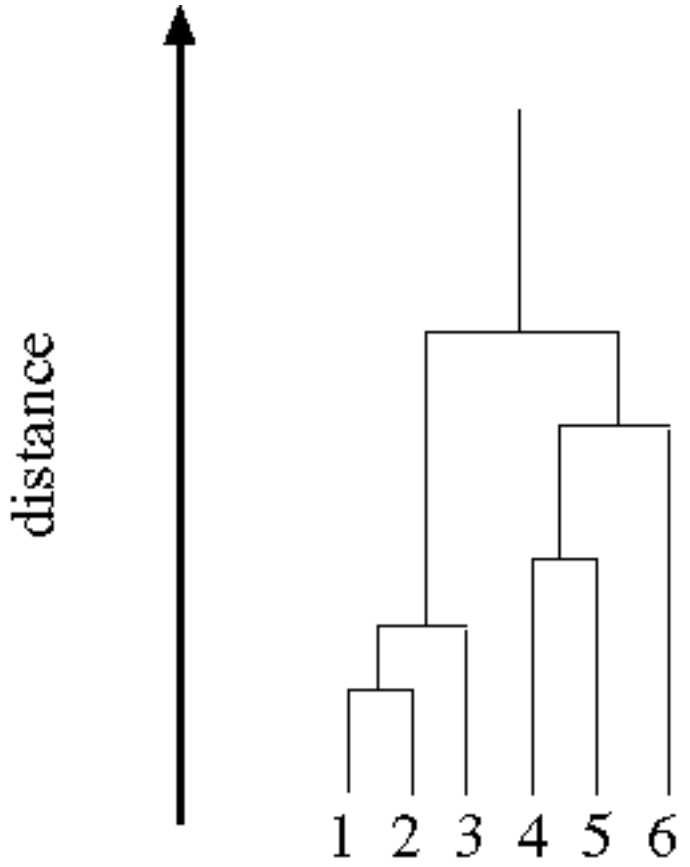


- Agglomerative clustering
  - Add token to cluster if token is similar enough to element of clusters
  - Repeat
- Divisive clustering
  - Split cluster into subclusters if tokens are dissimilar enough within cluster
  - Boundary separates subclusters based on similarity
  - Repeat

Hierarchical structure of clusters

- Dendrograms



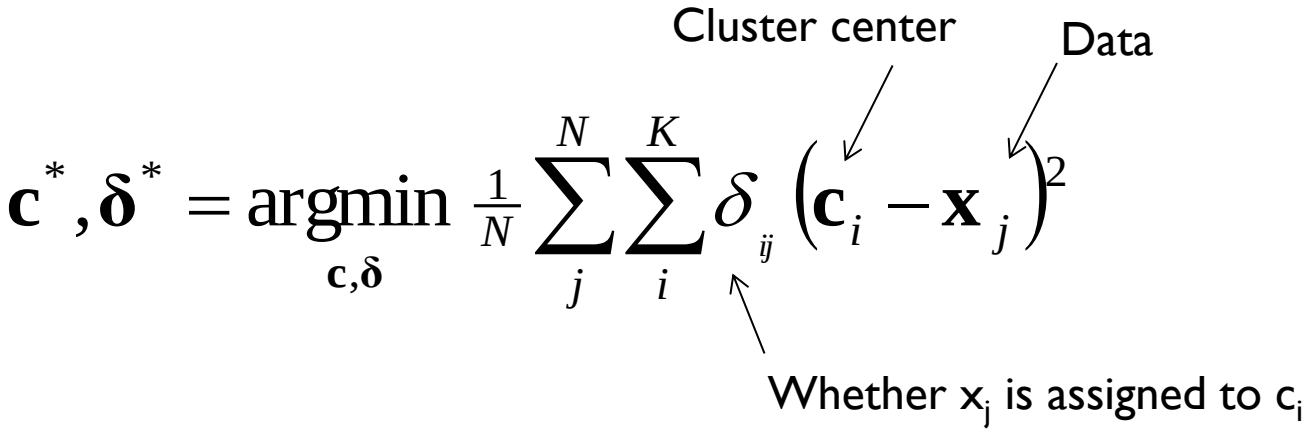


- rand('seed',12);
- X = rand(100,2);
- Y = pdist(X,'euclidean');
- Z = linkage(Y,'single');
- [H,T] = dendrogram(Z);

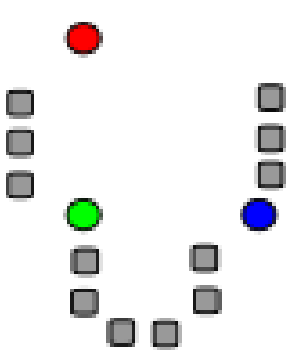
Clustering for Summarization

Goal: cluster to minimize variance in data given clusters

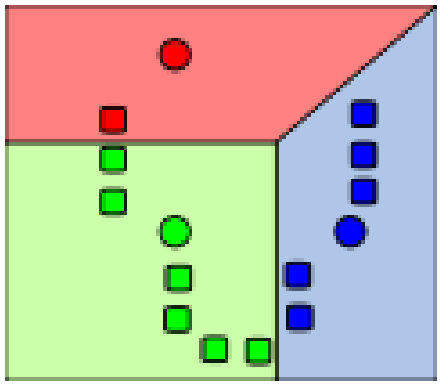
- Preserve information



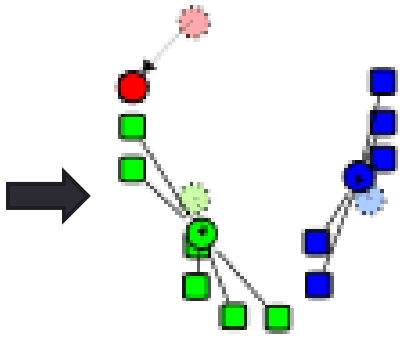
K-means



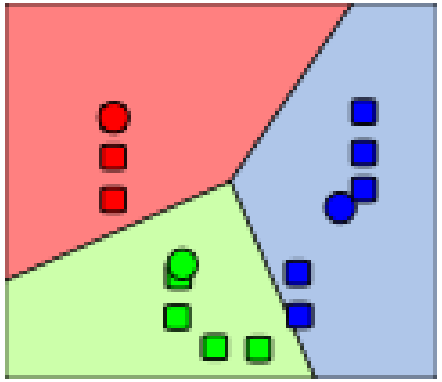
0. Initialize Cluster Centers



1. Assign Points to Clusters



2. Re-compute Means



Repeat (1) and (2)

Illustration Source: wikipedia

K-means

1. Initialize cluster centers: $\mathbf{c}^0$; t=0

2. Assign each point to the closest center

$$\boldsymbol{\delta}^t = \underset{\boldsymbol{\delta}}{\operatorname{argmin}} \frac{1}{N} \sum_j \sum_i \delta_{ij} \left(\mathbf{c}_i^{t-1} - \mathbf{x}_j \right)^2$$

3. Update cluster centers as the mean of the points

$$\mathbf{c}^t = \underset{\mathbf{c}}{\operatorname{argmin}} \frac{1}{N} \sum_j \sum_i \delta_{ij}^t \left(\mathbf{c}_i - \mathbf{x}_j \right)^2$$

4. Repeat 2-3 until no points are re-assigned (t=t+1)

K-means: design choices

- Initialization
  - Randomly select K points as initial cluster center
  - Or greedily choose K points to minimize residual
- Distance measures
  - Traditionally Euclidean, could be others
- Optimization
  - Will converge to a *local minimum*
  - May want to perform multiple restarts

Illustration Source: wikipedia

Clustering

- K-means clustering using intensity alone and color alone

Image

Clusters on intensity

Clusters on color

How to choose the number of clusters?

- Minimum Description Length (MDL) principal for model comparison
- Minimize Schwarz Criterion
  - also called Bayes Information Criteria (BIC)
- Validation set
  - Try different numbers of clusters and look at performance
    - When building dictionaries (discussed later), more clusters typically work better

How to evaluate clusters?

- Generative
  - How well are points reconstructed from the clusters?
- Discriminative
  - How well do the clusters correspond to labels?
    - Purity
  - Note: unsupervised clustering does not aim to be discriminative

K-means demos

General:

http://home.dei.polimi.it/matteucc/Clustering/tutorial_html/AppletKM.html

Color clustering:

http://www.cs.washington.edu/research/imagedatabase/demo/kmcluster/

MATLAB implementations:

- http://www.mathworks.com/matlabcentral/fileexchange/8379-kmeans-image-segmentation
- http://www.mathworks.com/matlabcentral/fileexchange/24616-kmeans-clustering

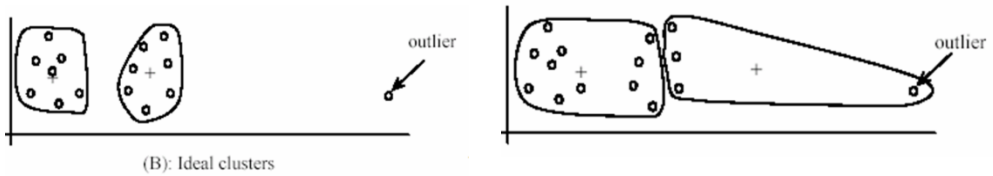
Conclusions: K-means

Pros

- Finds cluster centers that minimize conditional variance (good representation of data)
- Simple to implement, widespread application

Cons

- Prone to local minima (Sensitive to initialization)
- Need to choose K
- All clusters have the same parameters (e.g., distance measure is non-adaptive)
- Can be slow: each iteration is $O(KNd)$ for N d-dimensional points
- Sensitive to outliers



Common similarity/distance measures

- P-norms
  - City Block (L1)
  - Euclidean (L2)
  - L-infinity

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

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

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

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

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

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

Segmentation as clustering

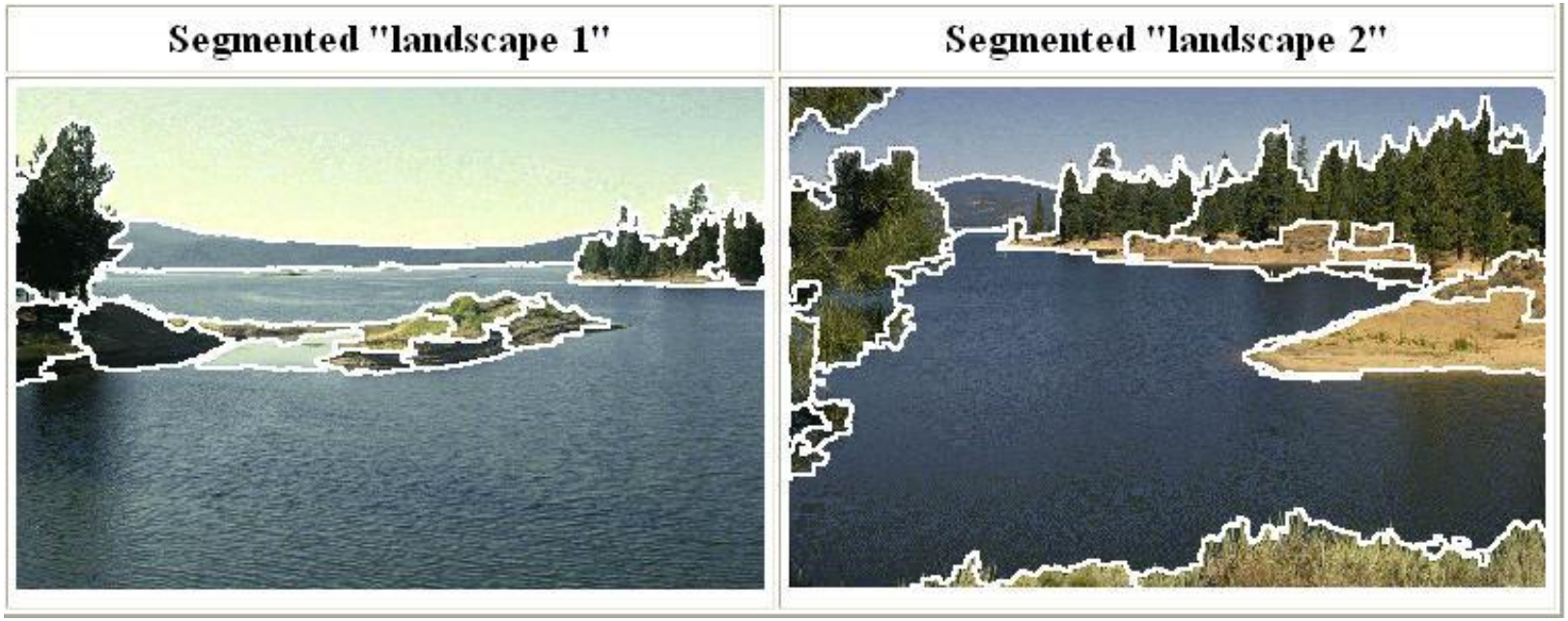
Cluster together tokens that share similar visual characteristics

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

Mean shift segmentation

- An advanced and versatile technique for clustering-based segmentation

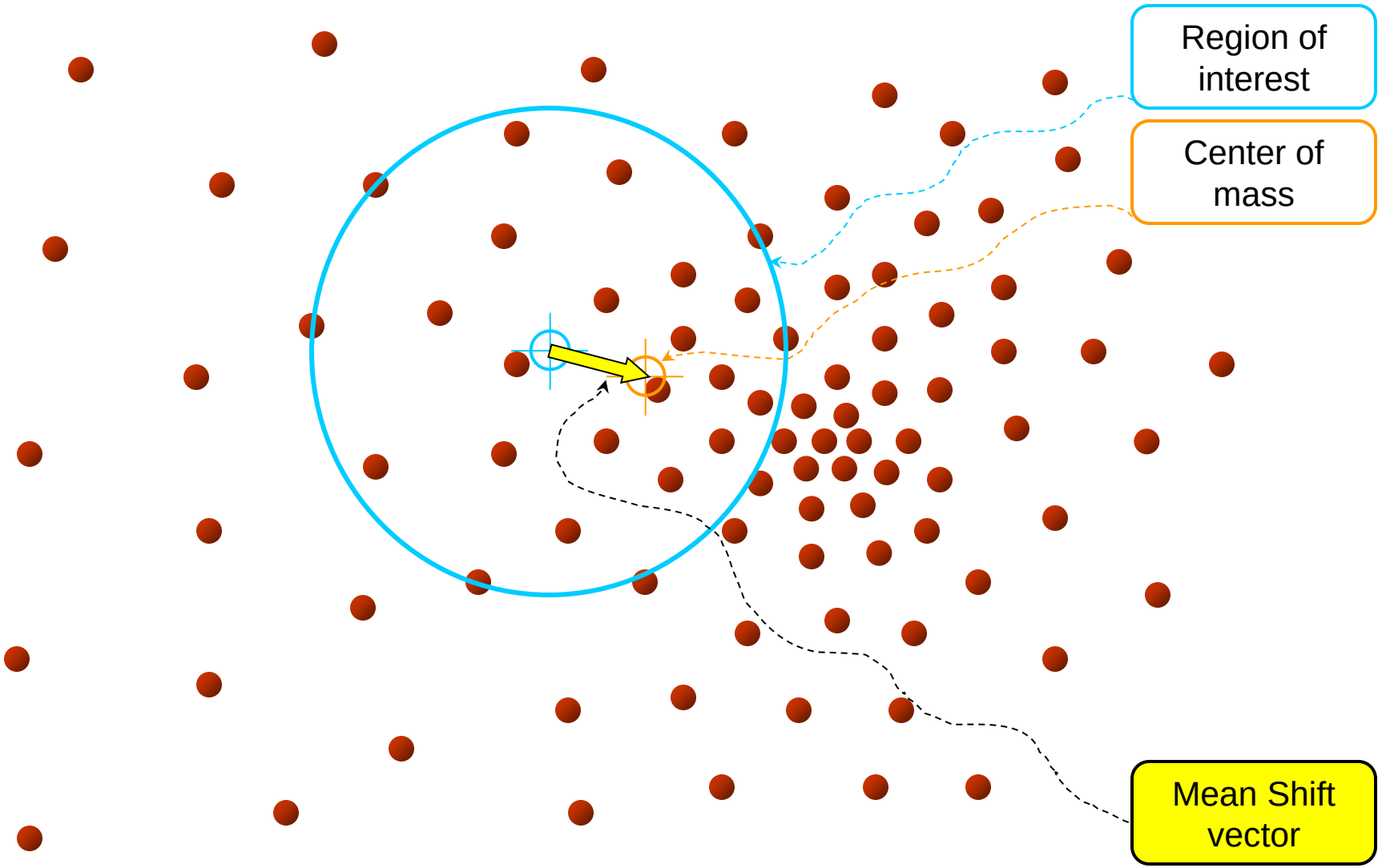
- D. Comaniciu and P. Meer, Mean Shift: A Robust Approach toward Feature Space Analysis, PAMI 2002.



Mean shift segmentation

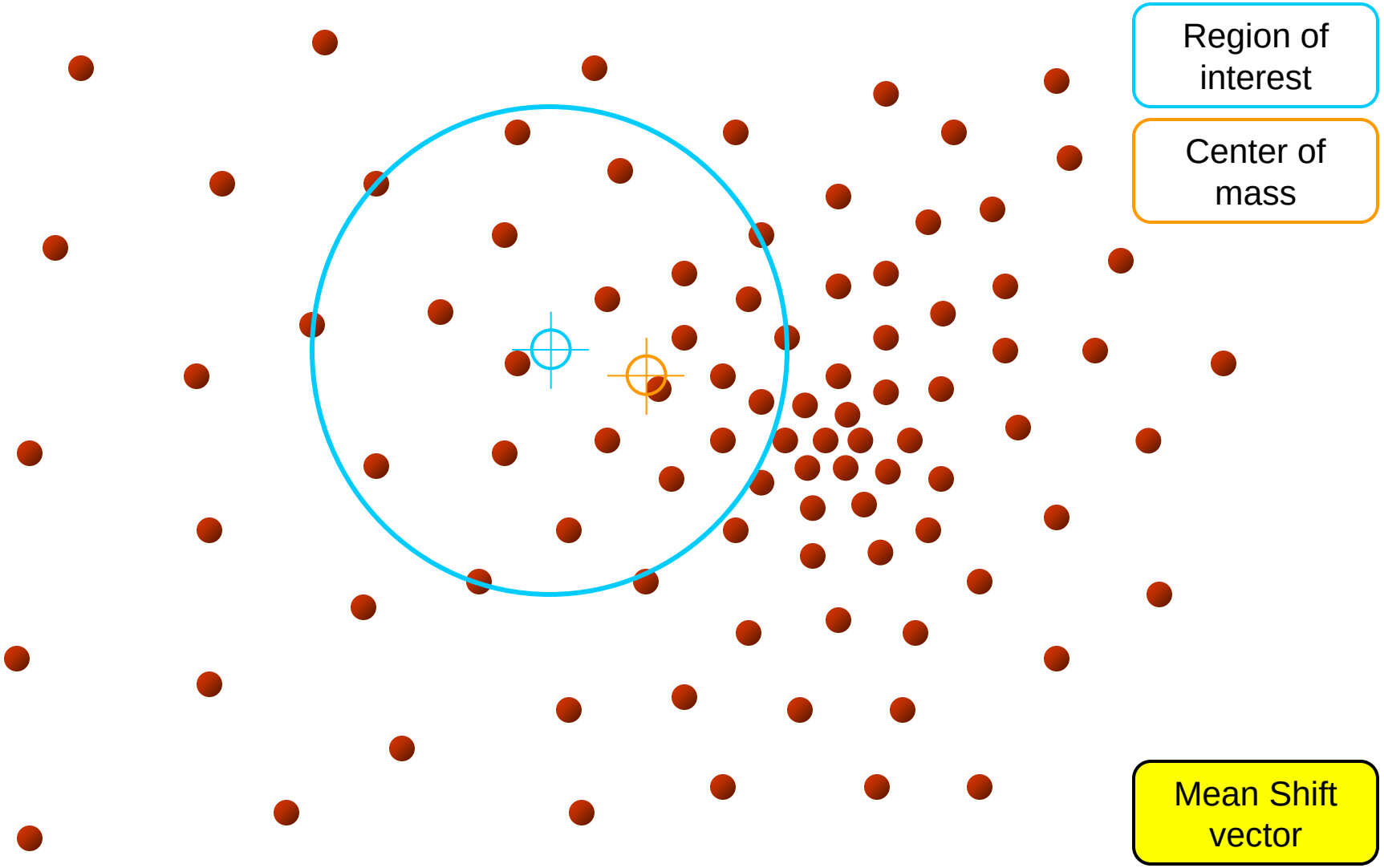
- The mean shift algorithm seeks a *mode* or local maximum of density of a given distribution
  - Choose a search window (width and location)
  - Compute the mean of the data in the search window
  - Center the search window at the new mean location
  - Repeat until convergence

Mean shift



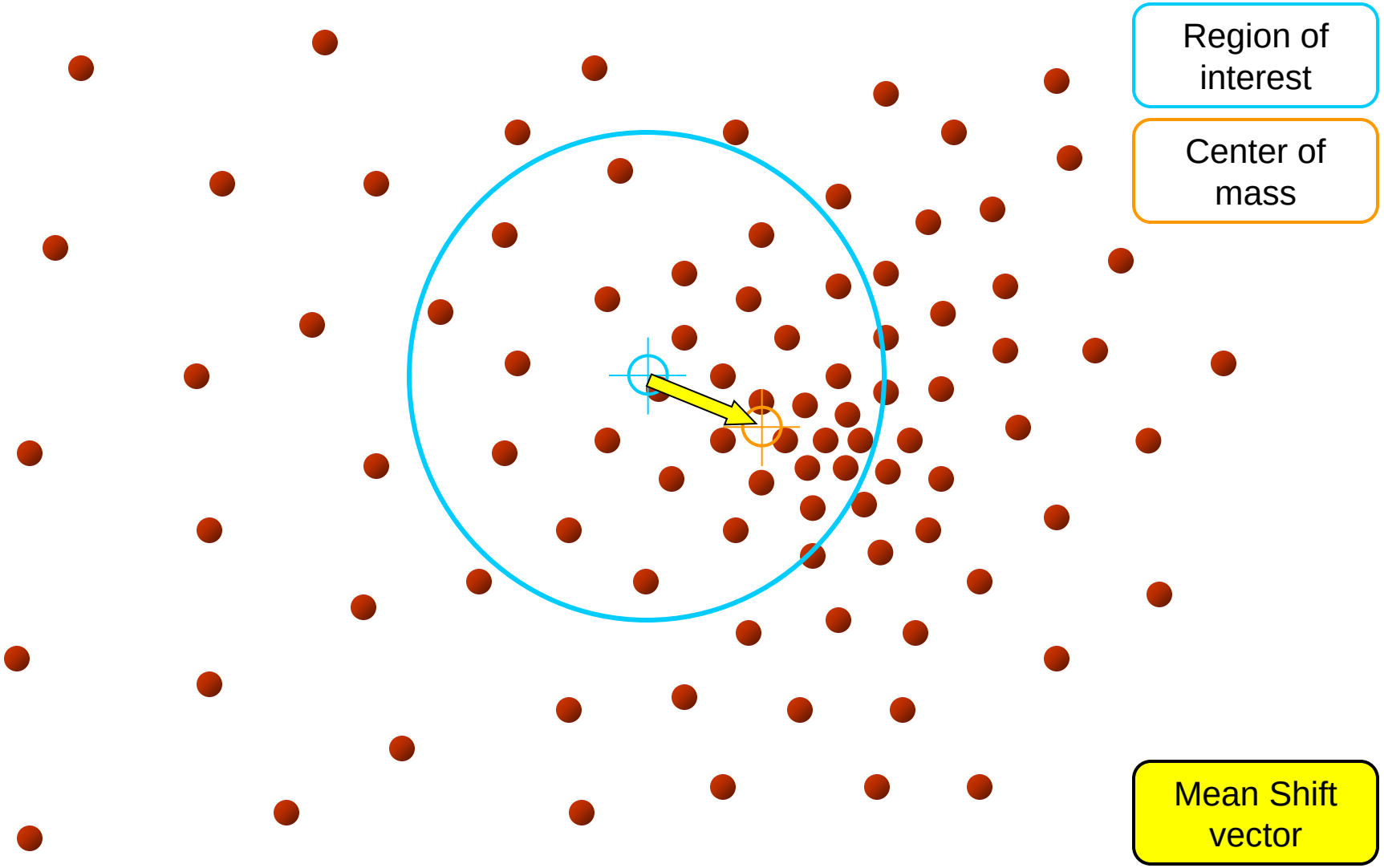
Slide by Y. Ukrainitz & B. Sarel

Mean shift



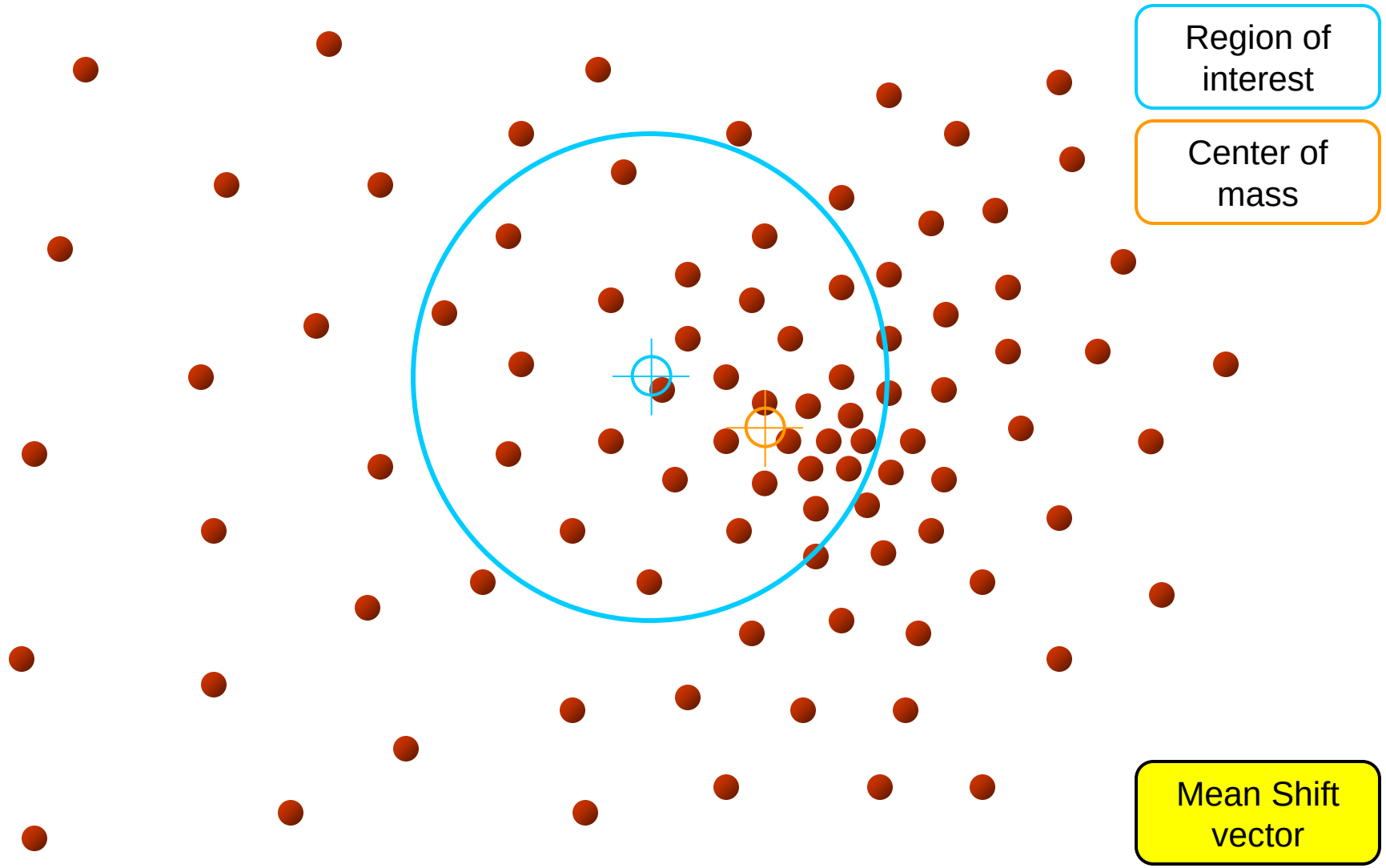
Slide by Y. Ukrainitz & B. Sarel

Mean shift



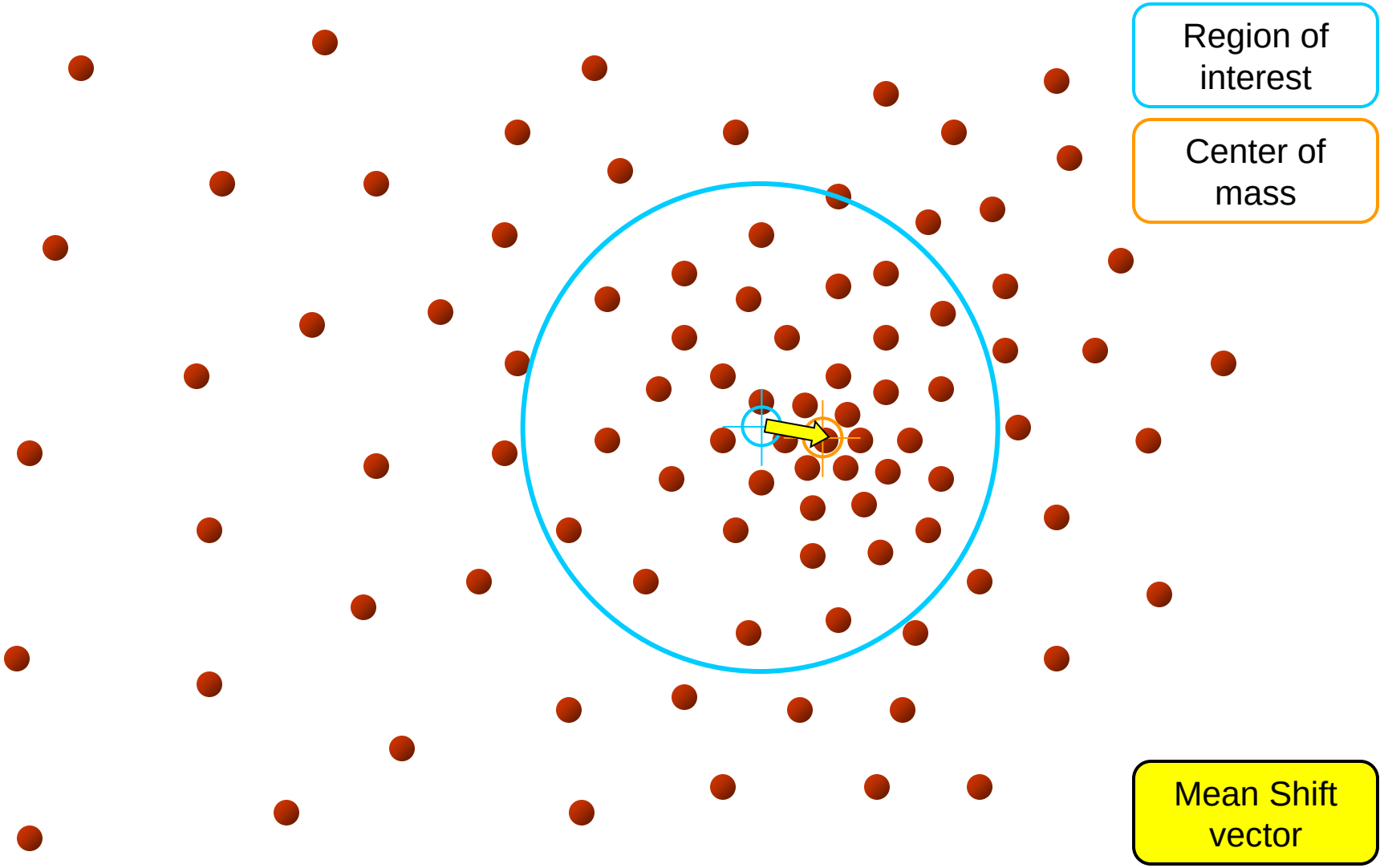
Slide by Y. Ukrainitz & B. Sarel

Mean shift



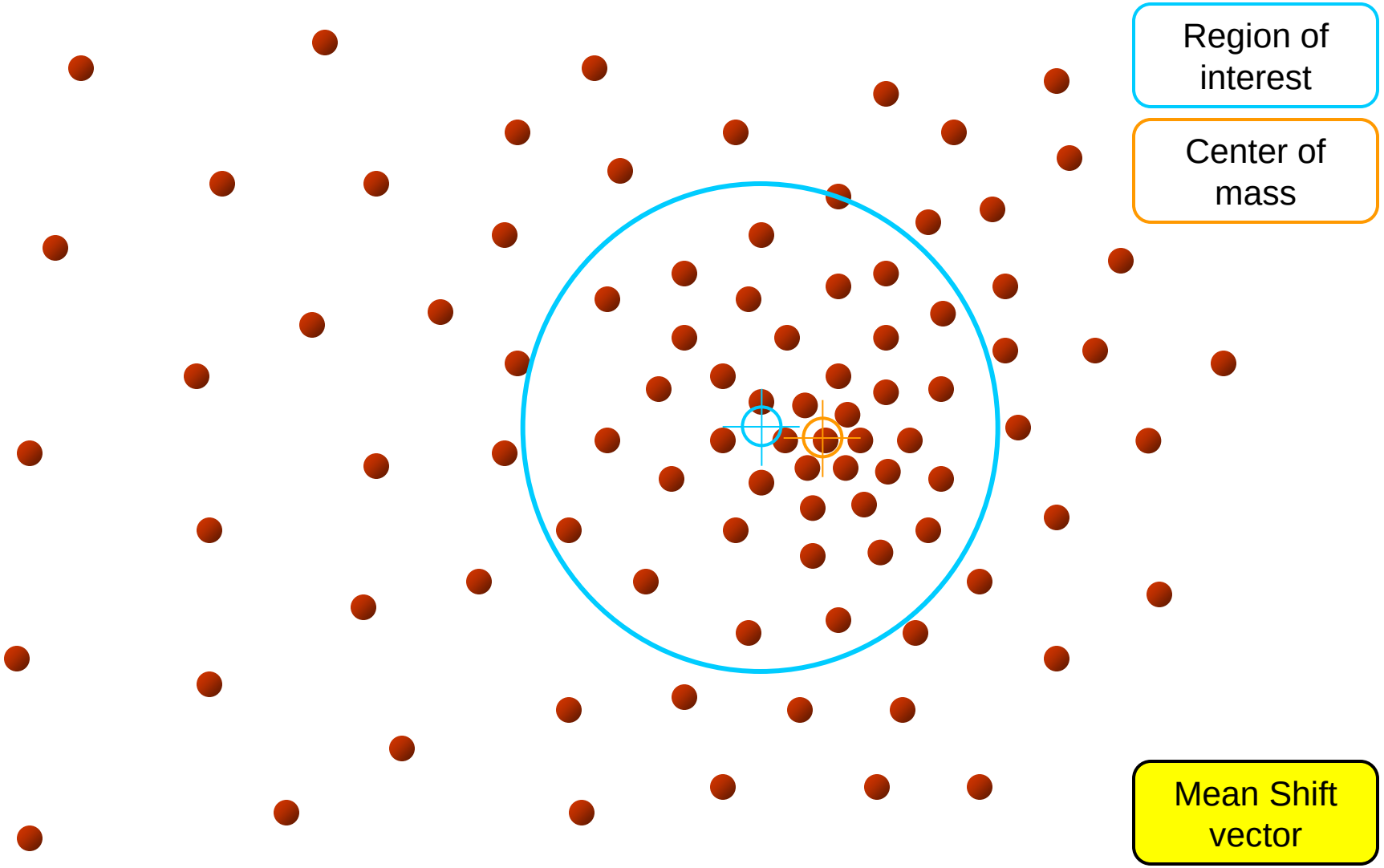
Slide by Y. Ukrainitz & B. Sarel

Mean shift



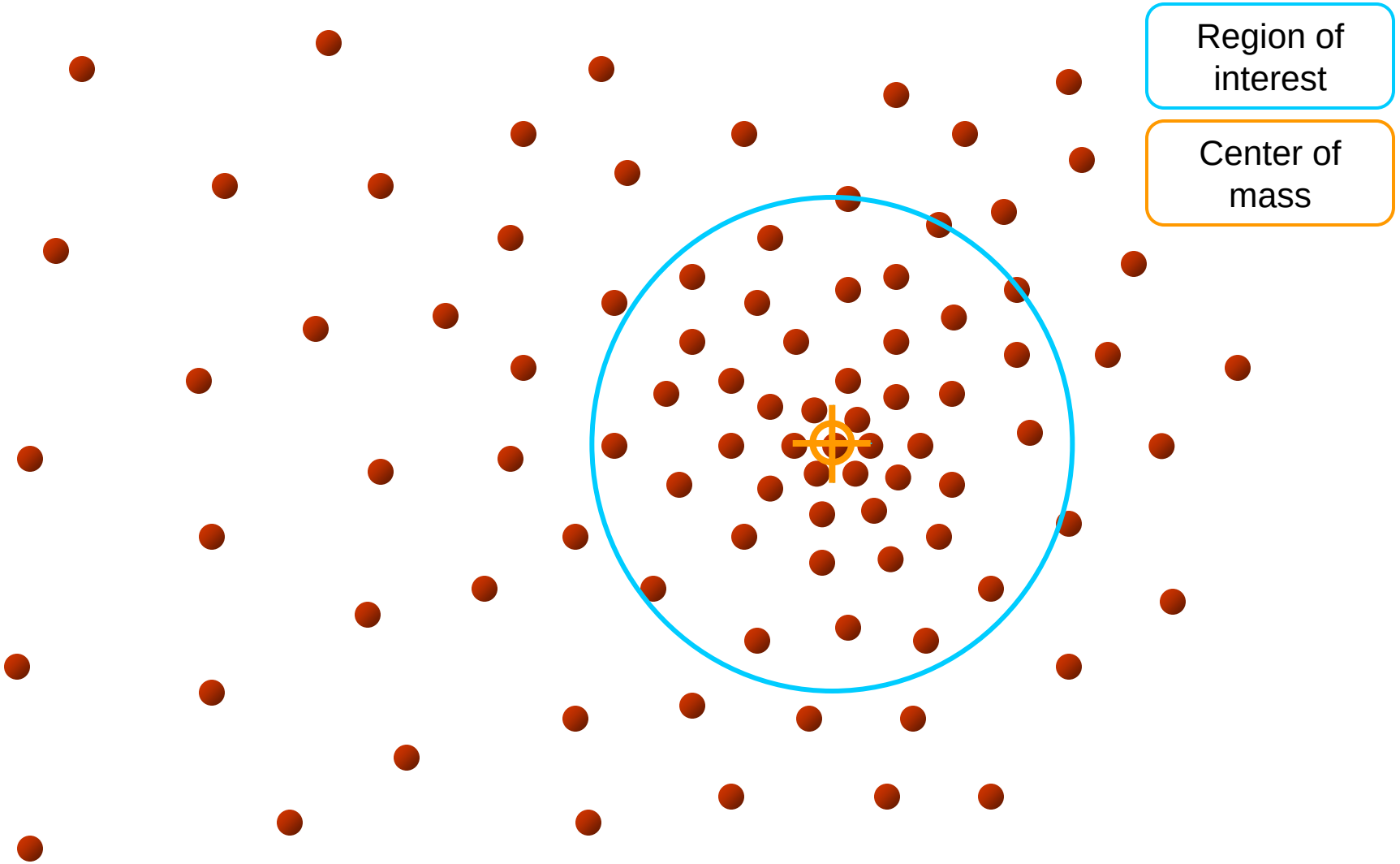
Slide by Y. Ukrainitz & B. Sarel

Mean shift



Slide by Y. Ukrainitz & B. Sarel

Mean shift



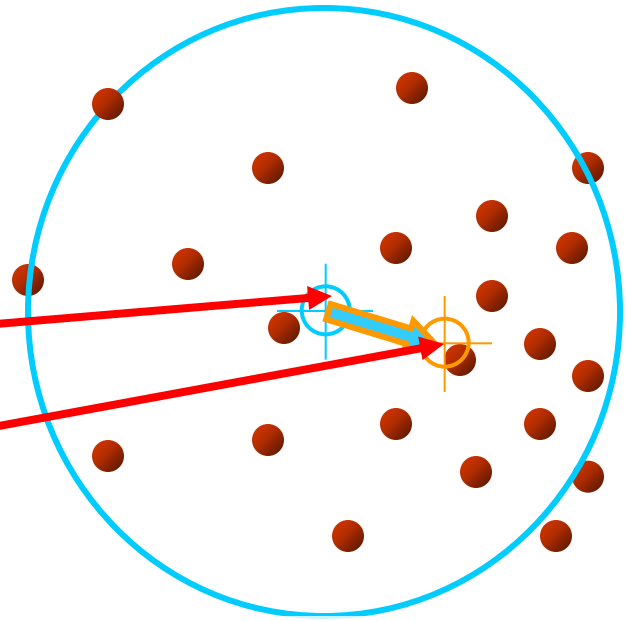
Slide by Y. Ukrainitz & B. Sarel

Computing The Mean Shift

- Simple Mean Shift procedure:
- Compute mean shift vector

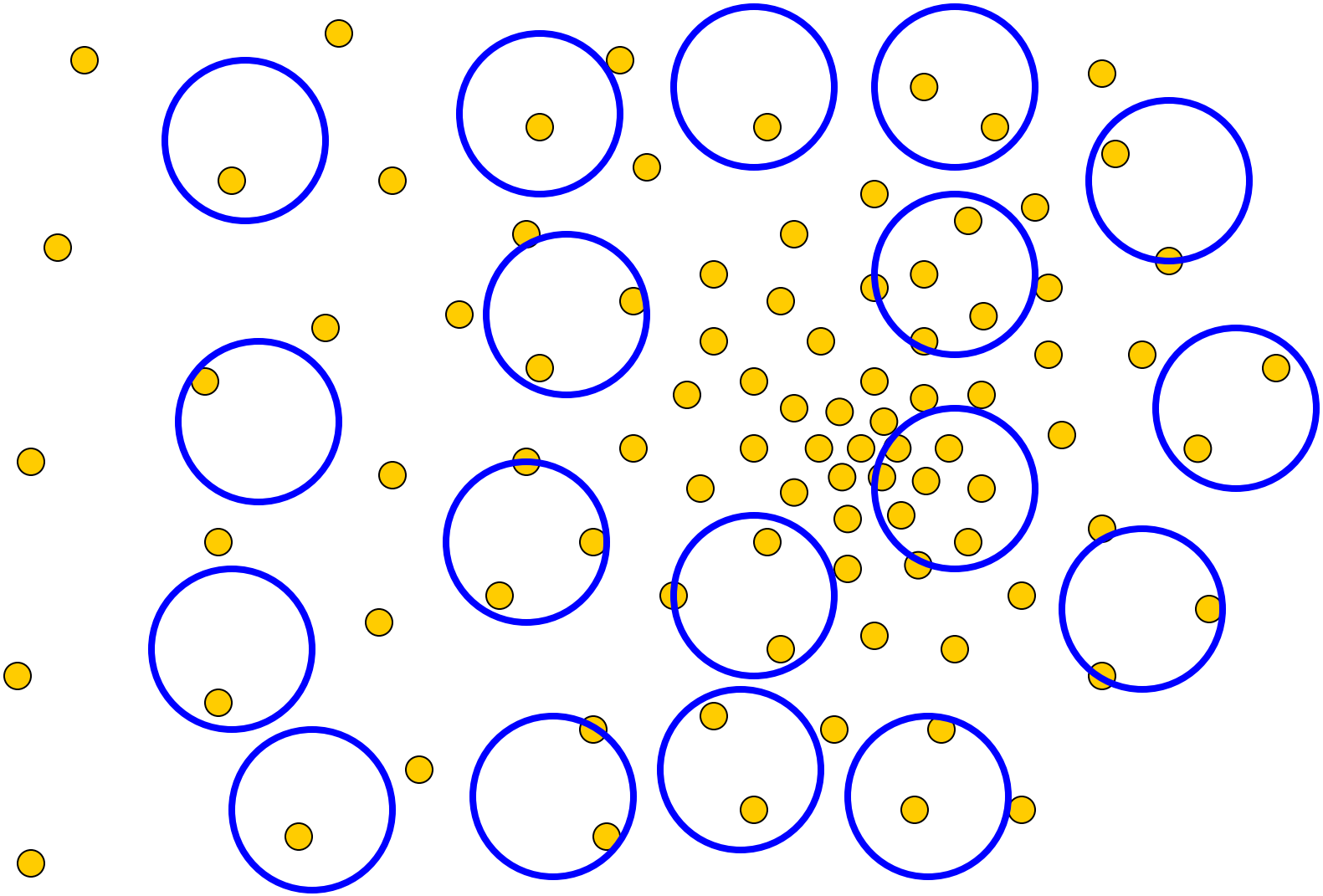
- Translate the Kernel window by **m(x)**

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



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

Real Modality Analysis

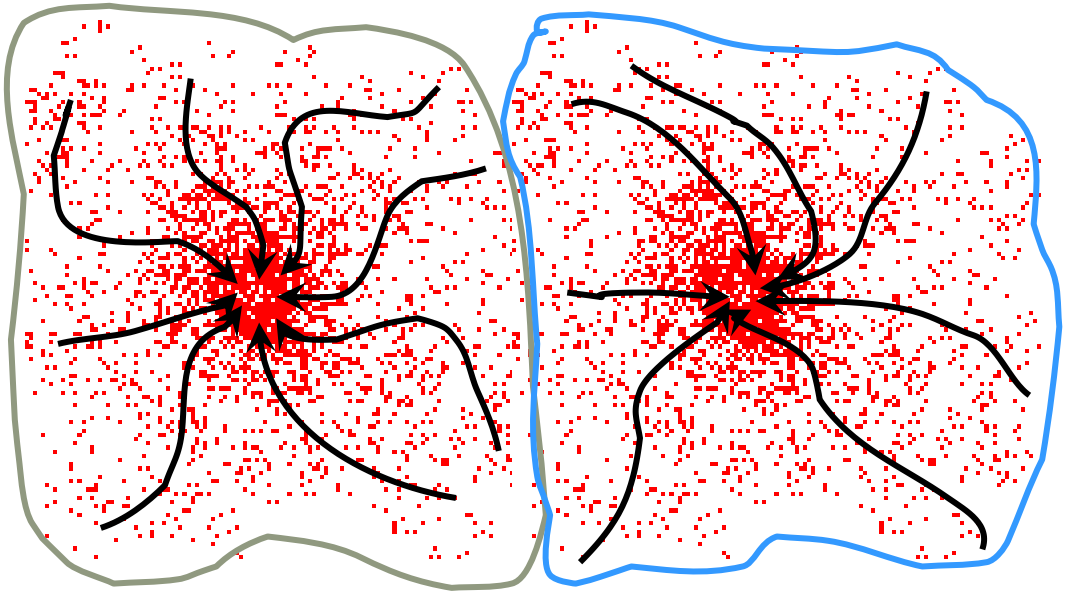


- **Tessellate the space with windows**

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

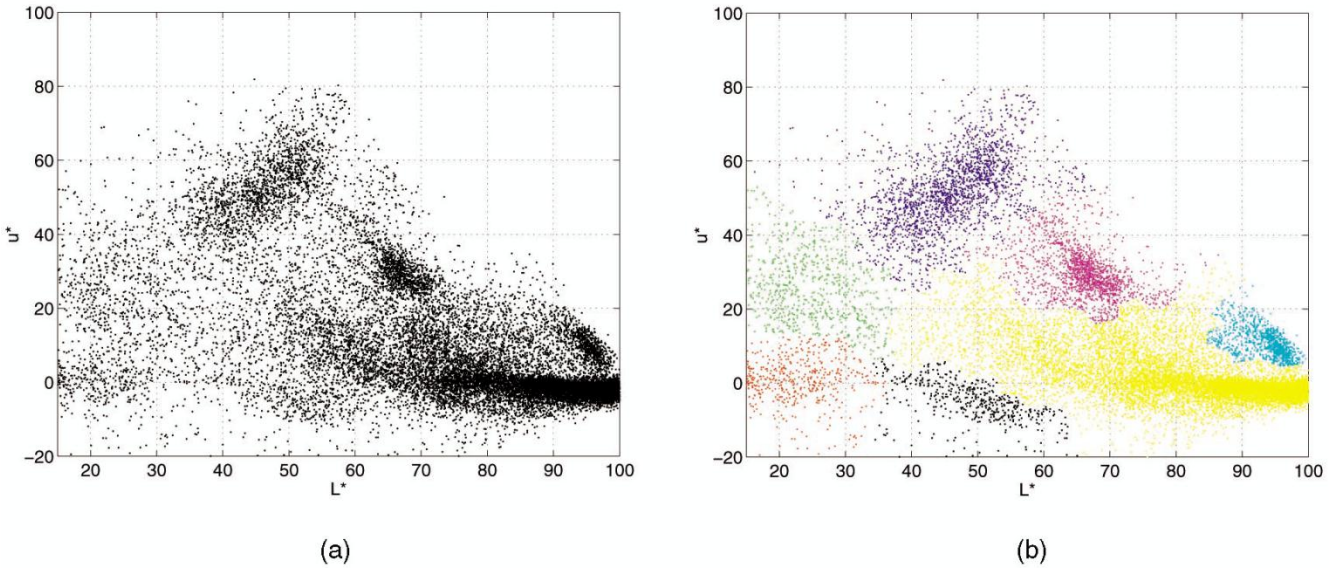
Attraction basin

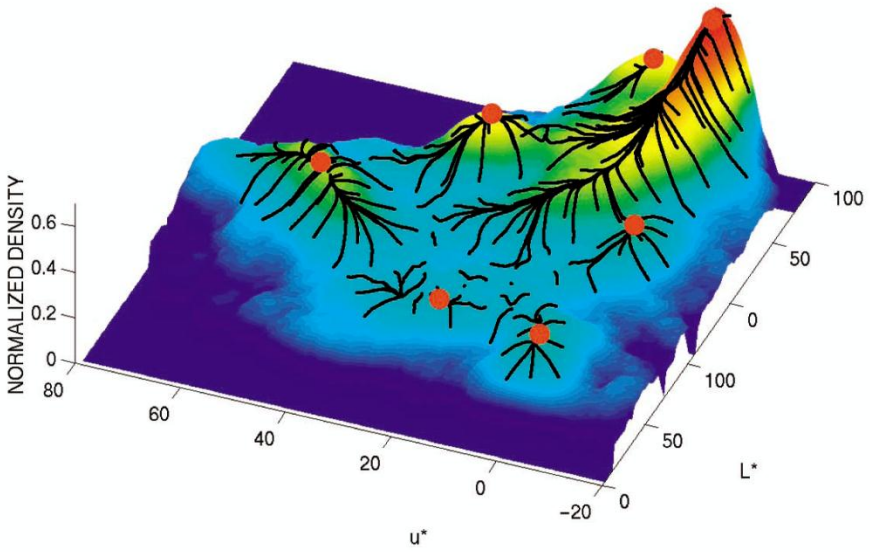
- Attraction basin: the region for which all trajectories lead to the same mode
- Cluster: all data points in the attraction basin of a mode



Slide by Y. Ukrainitz & B. Sarel

Attraction basin

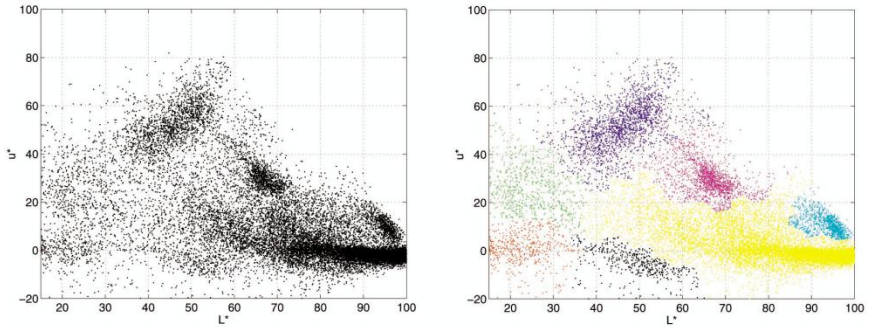


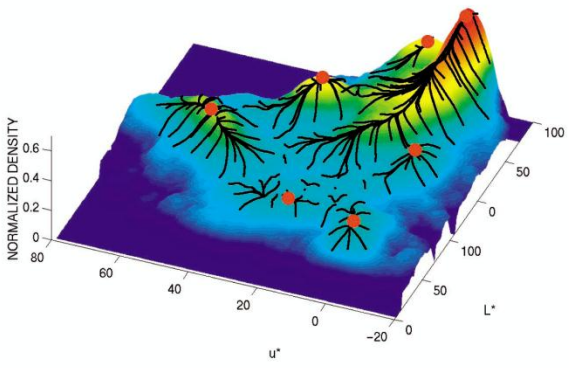


Slide by Y. Ukrainitz & B. Sarel

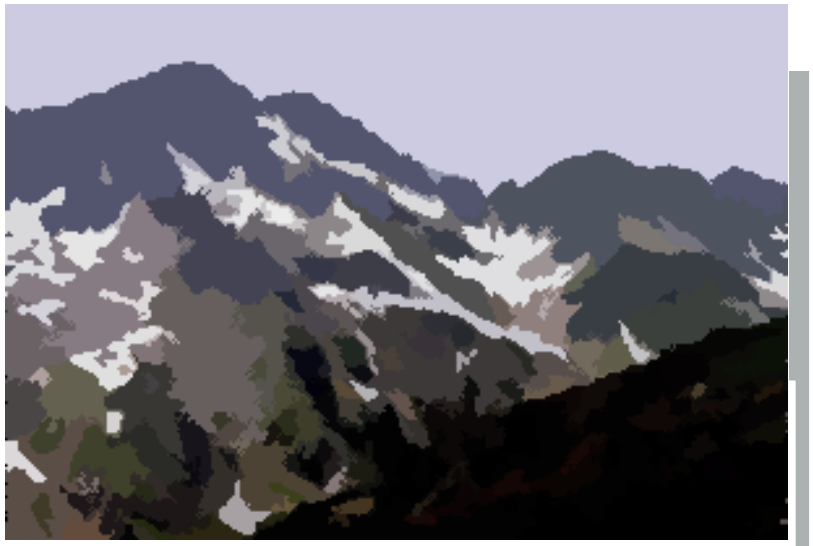
Segmentation by Mean Shift

- 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

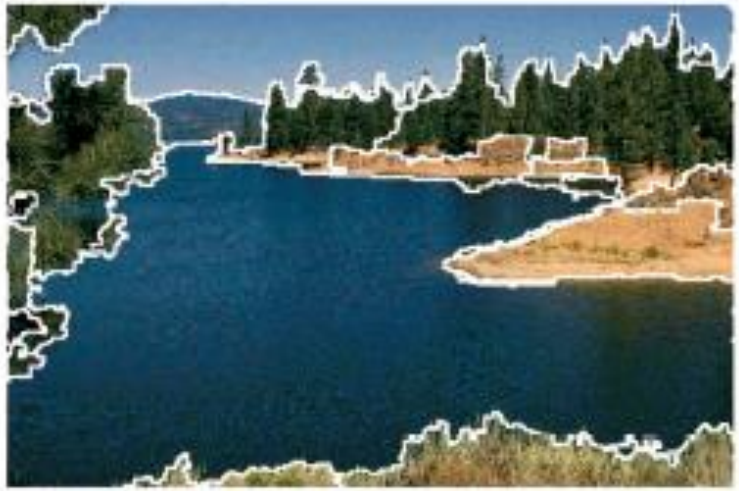




Mean shift segmentation results



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

- Matlab Implementation
  - http://www.mathworks.com/matlabcentral/fileexchange/10161

Segmentation as clustering

Cluster together tokens that share similar visual characteristics

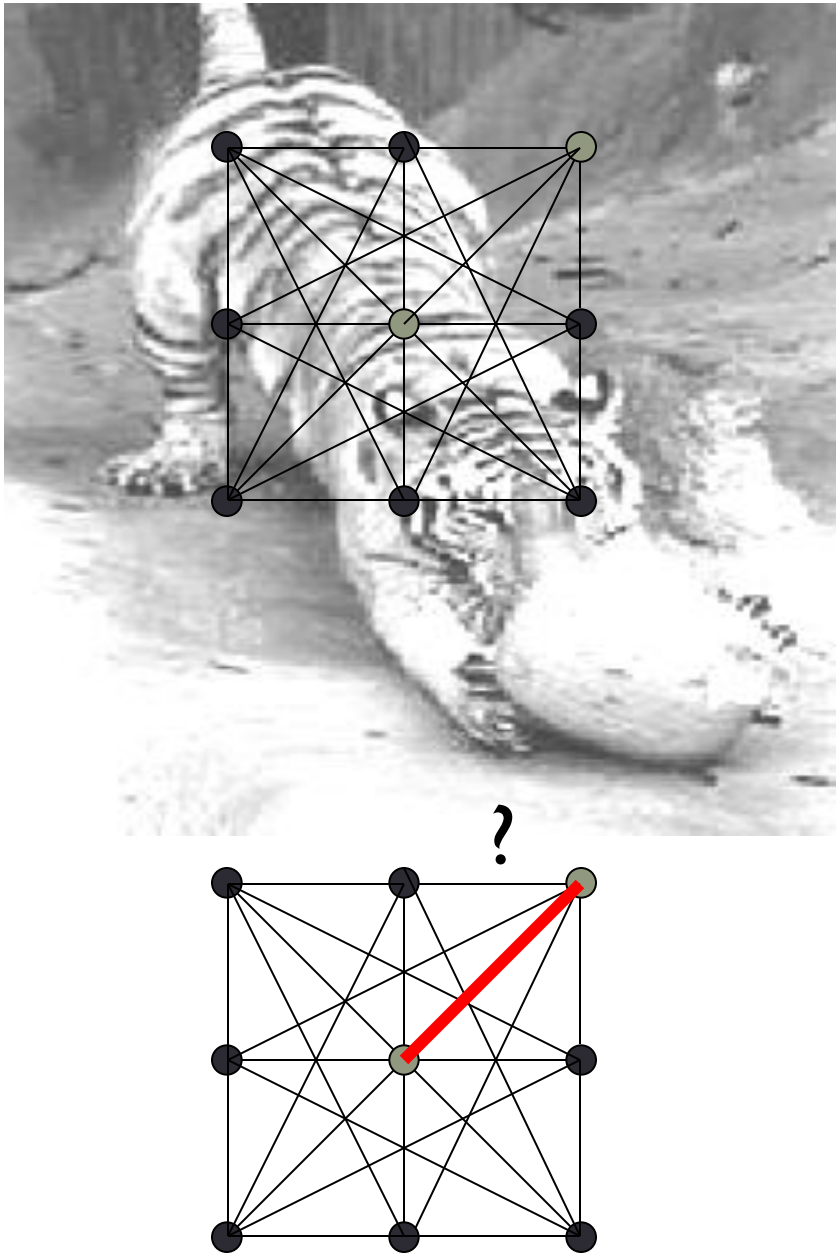
- K-mean
- Mean-shift
- Graph-cut

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

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

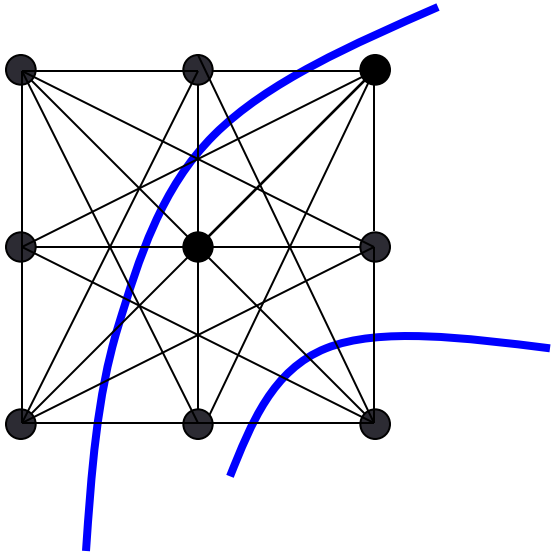
Intensity

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

Color

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

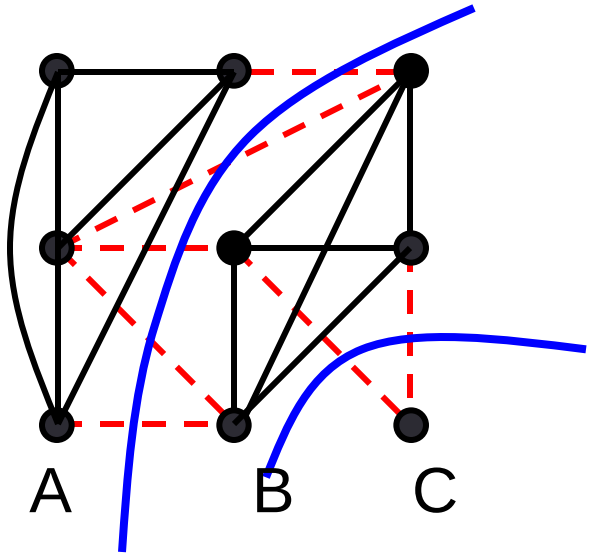
Segmentation by graph partitioning



- 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

Source: S. Seitz

Segmentation by graph partitioning



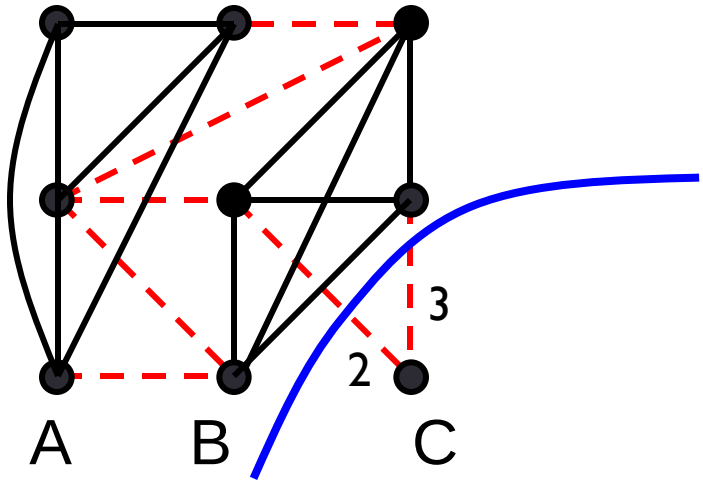
- Break Graph into sub-graphs
  - Break links (cutting) that have low affinity
    - similar pixels should be in the same sub-graphs
    - dissimilar pixels should be in different sub-graphs

•Sub-graphs represents different image segments

• Graph-cut: technique to cut a graph optimally

Source: S. Seitz

Segmentation by graph partitioning

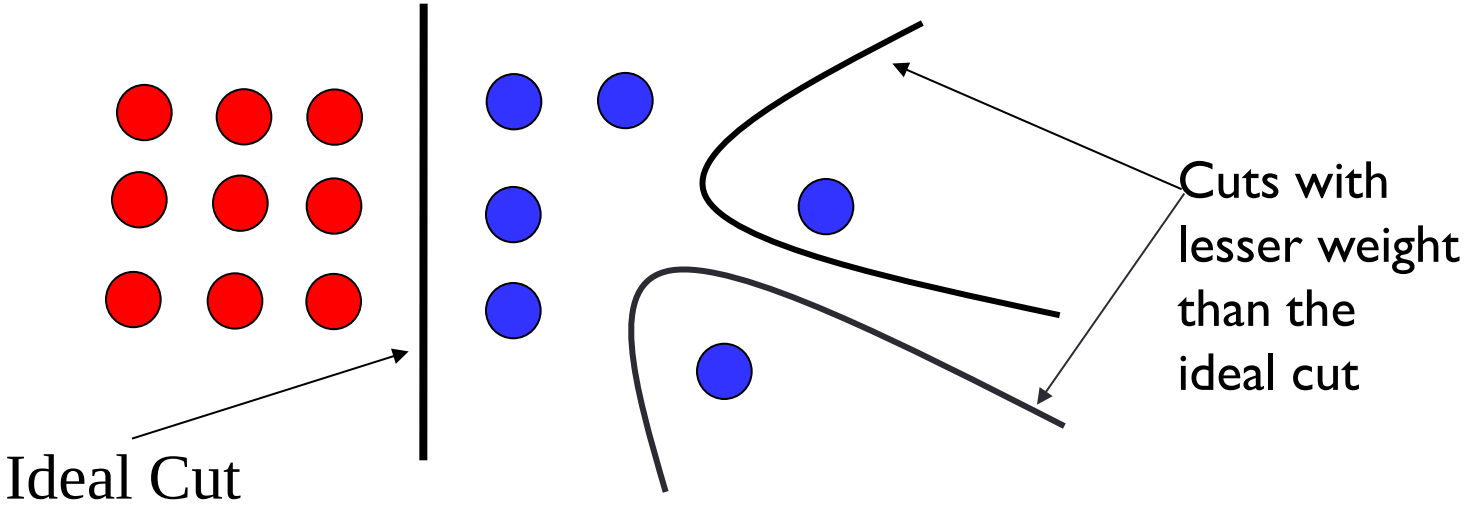


- CUT: Set of edges whose removal makes a graph disconnected
- Cost of a cut: sum of weights of cut edges
- Example: Cost of the blue cut?

Source: S. Seitz

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



* Slide from Khurram Hassan-Shafique CAP5415 Computer Vision 2003

Normalized cut

- 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

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

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

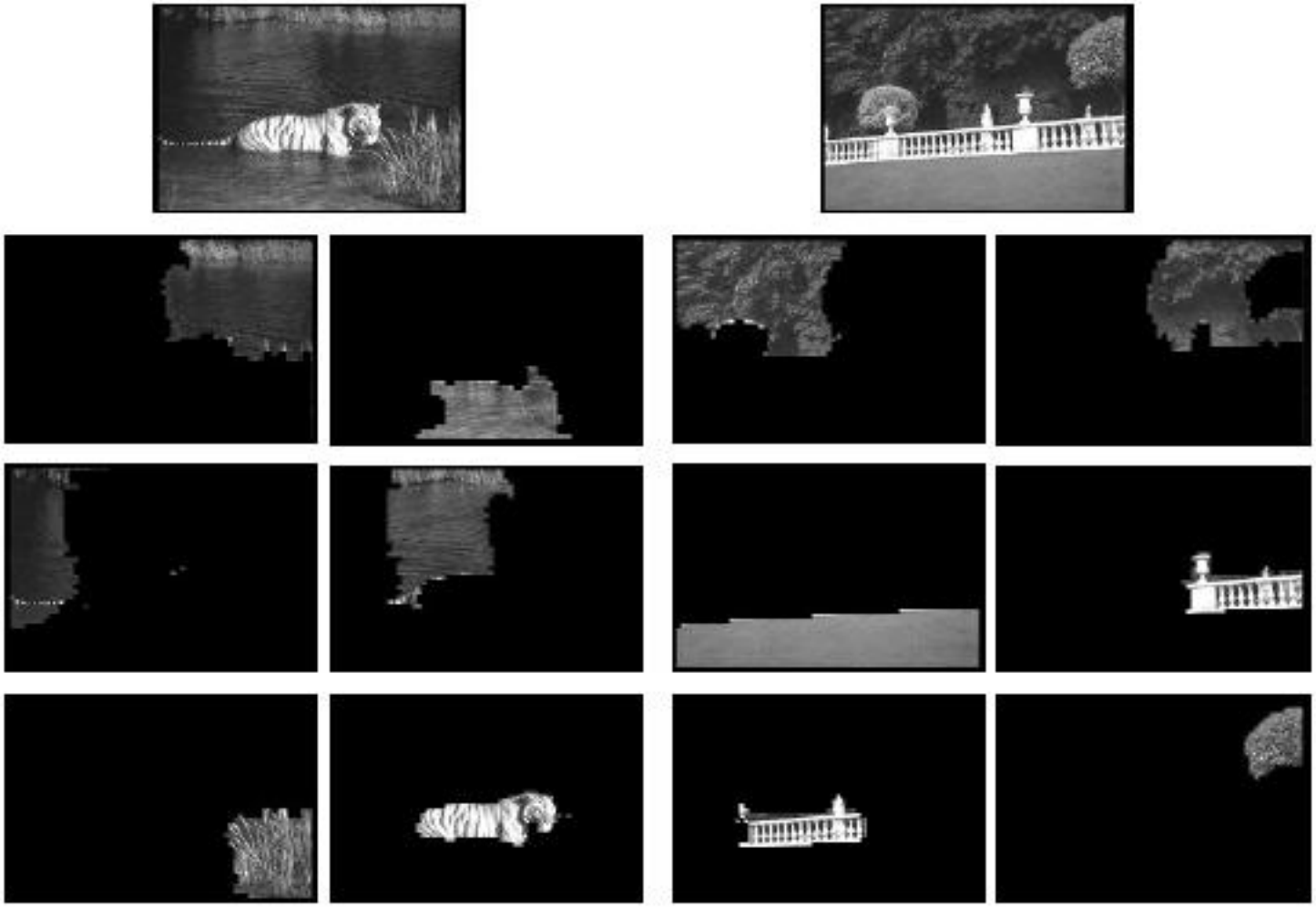


Figure from "Image and video segmentation: the normalised cut framework", by Shi and Malik, copyright IEEE, 1998

Normalized cuts: Results

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

Contour and Texture Analysis for Image Segmentation

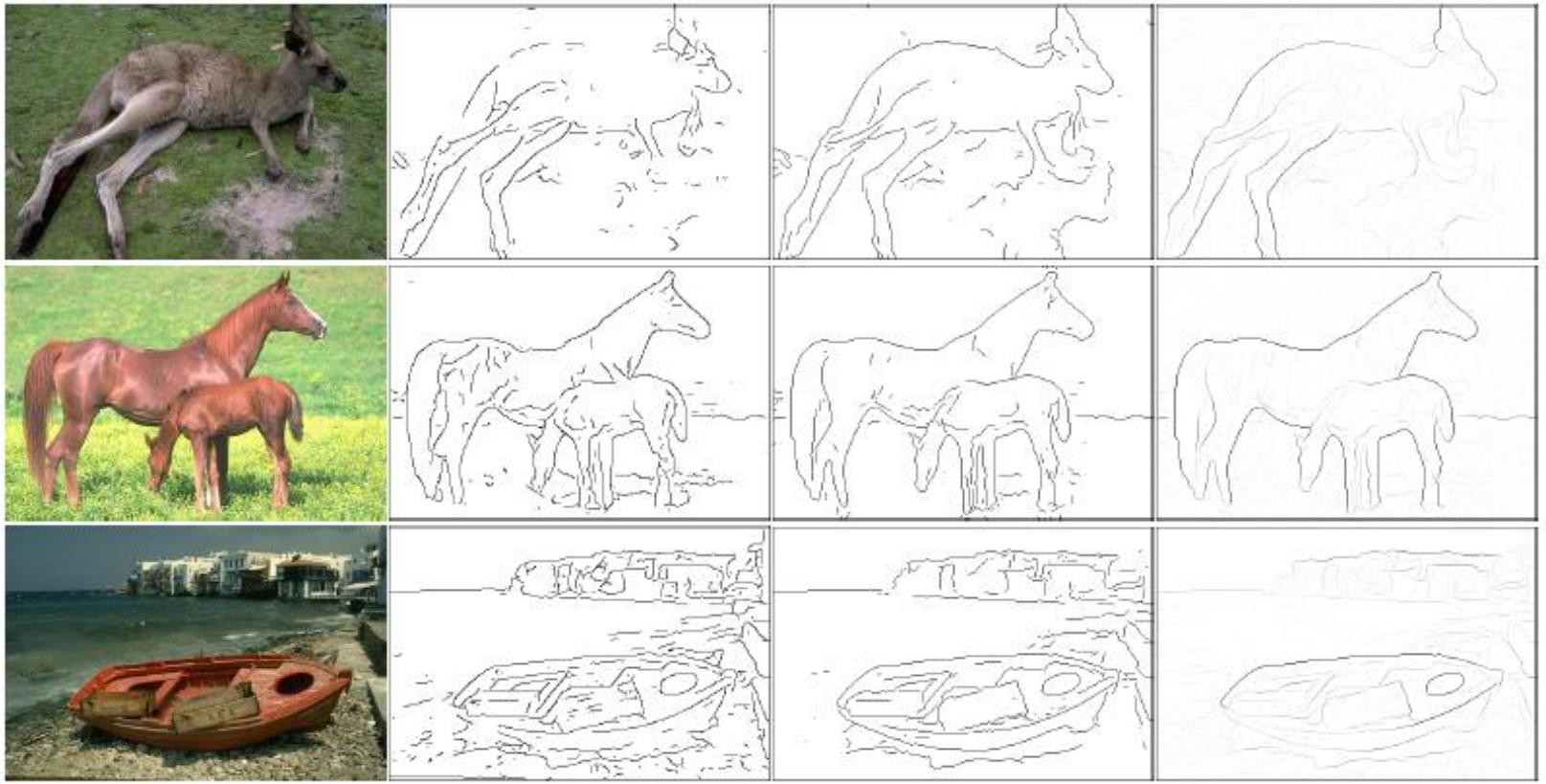
- J. Malik, S. Belongie, T. Leung and J. Shi. "*Contour and Texture Analysis for Image Segmentation*". IJCV 43(1),7-27,2001.

Contour and Texture Analysis for Image Segmentation

- Using Contours to Detect and Localize Junctions in Natural Images"
  M. Maire, P. Arbelaez, C. Fowlkes, and J. Malik. CVPR 2008

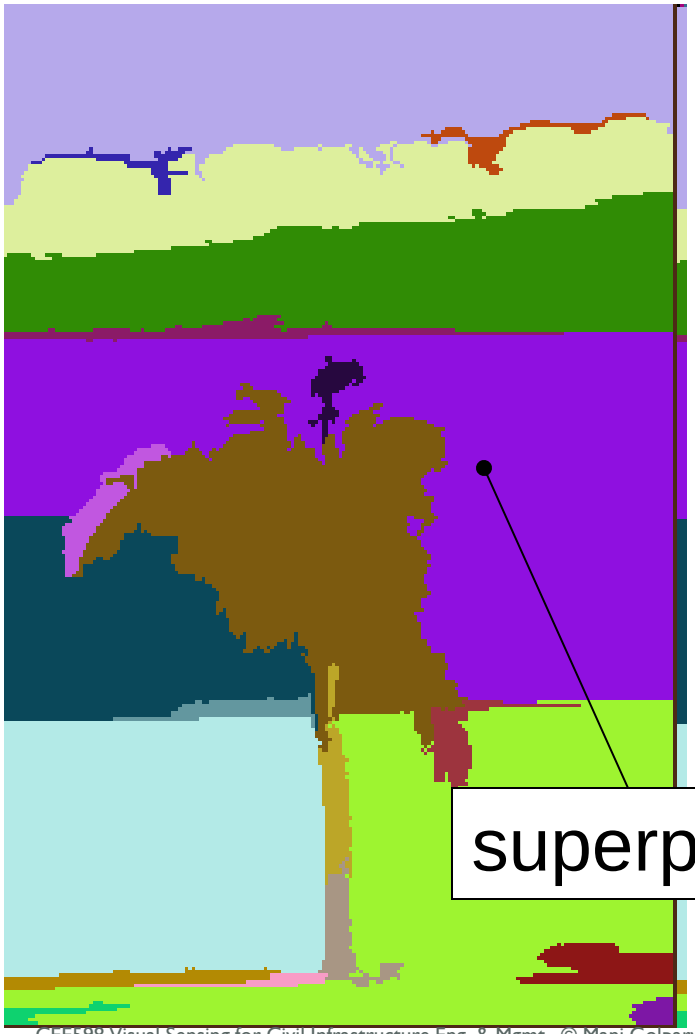
Now on CUDA

http://www.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/resources.html



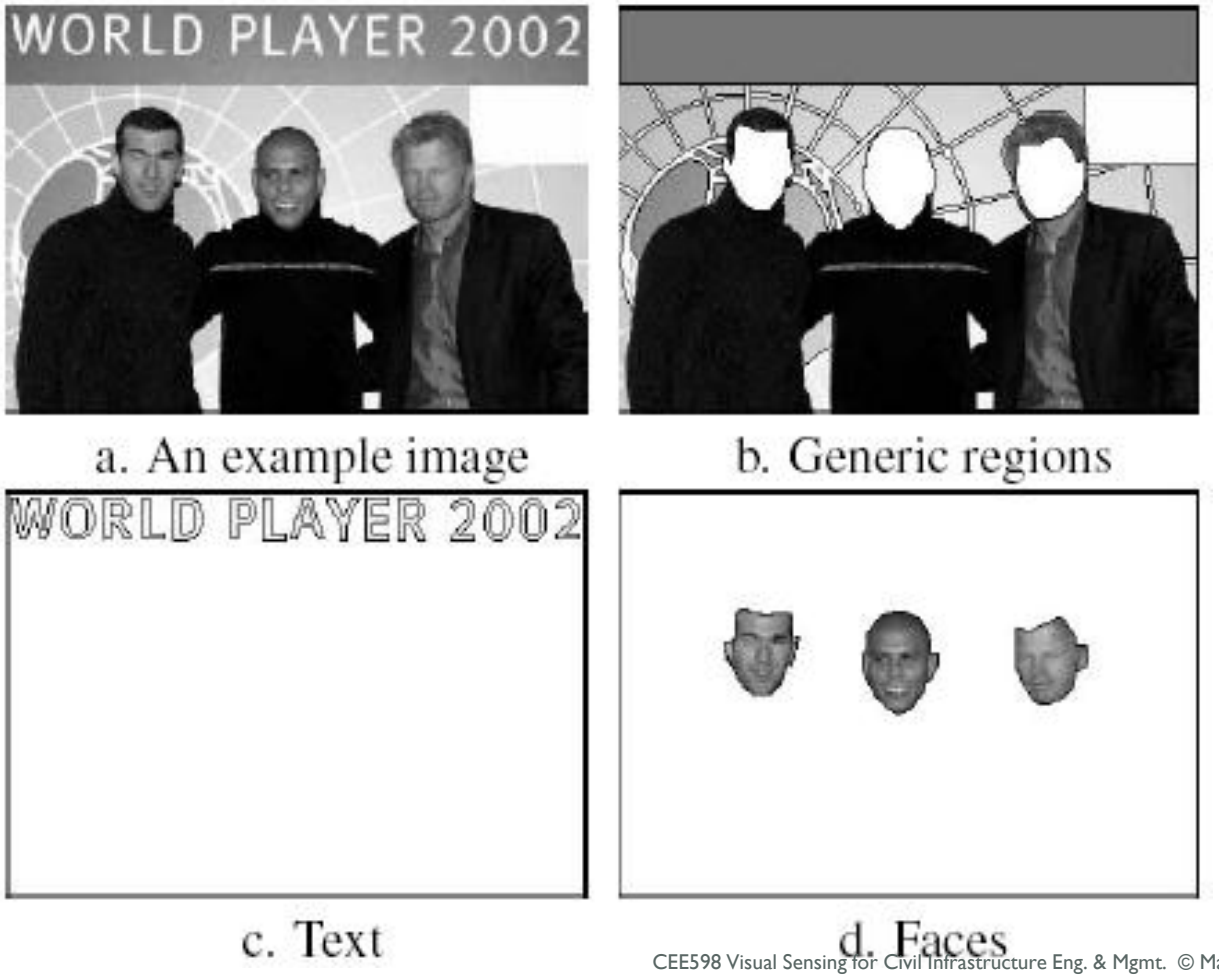
Efficient Graph-Based Image Segmentation

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.



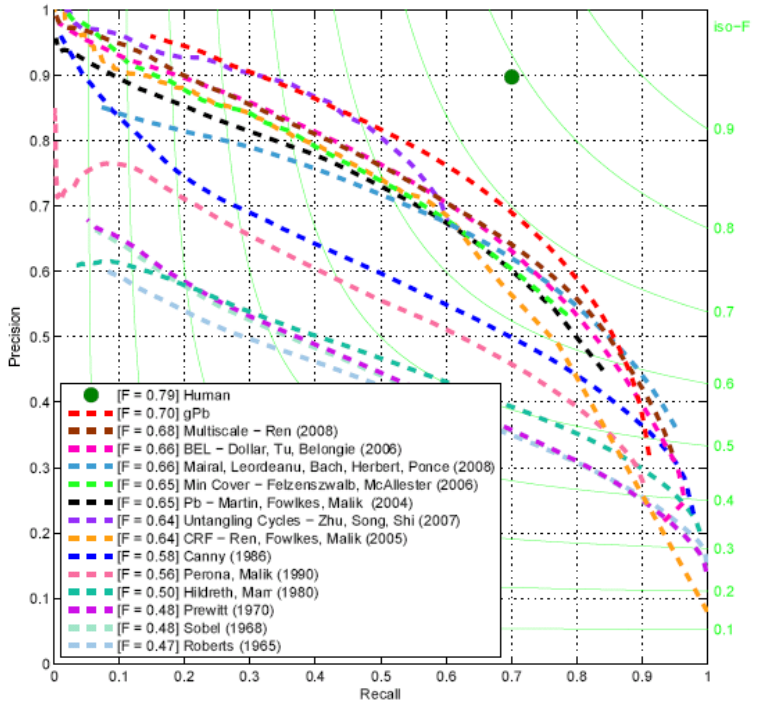


Fig. 1. **Evaluation of contour detectors on the Berkeley Segmentation Dataset (BSDS300) Benchmark** [2]. Leading contour detection approaches are ranked according to their maximum F-measure ($\frac{2 \cdot Precision \cdot Recall}{Precision + Recall}$) with respect to human ground-truth boundaries. Iso-F curves are shown in green. Our gPb detector [3] performs significantly better than other algorithms [2], [17], [18], [19], [20], [21], [22], [23], [24], [25], [26], [27], [28] across almost the entire operating regime. Average agreement between human subjects is indicated by the green dot.

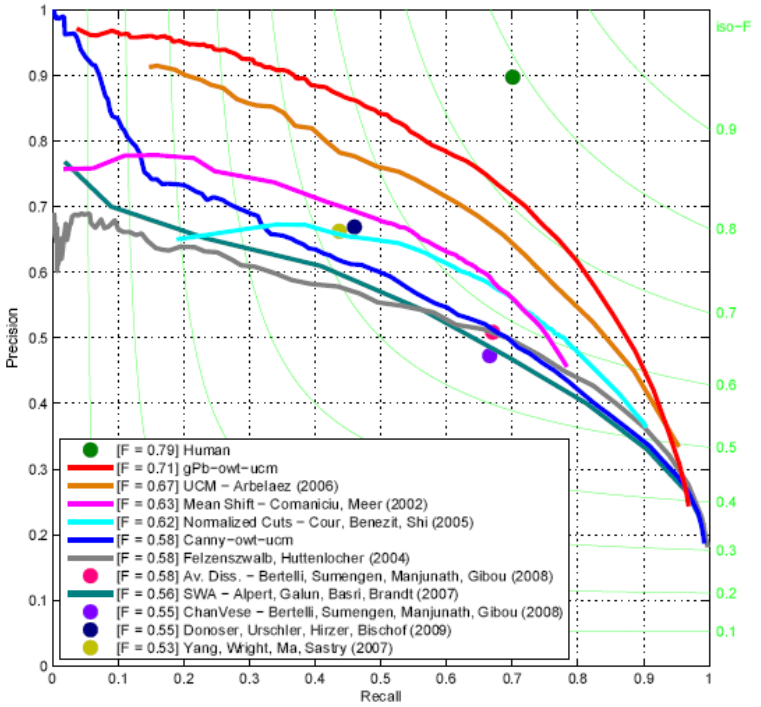


Fig. 2. **Evaluation of segmentation algorithms on the BSDS300 Benchmark.** Paired with our gPb contour detector as input, our hierarchical segmentation algorithm gPb-owt-ucm [4] produces regions whose boundaries match ground-truth better than those produced by other methods [7], [29], [30], [31], [32], [33], [34], [35].