CAP6411
Computer Vision Systems
Lecture 8

Alper Yilmaz
Office: CSB 250
Email: yilmaz@cs.ucf.edu
Web: http://www.cs.ucf.edu/courses/**cap6411/cap6411**/spring2006

Removing Camera Motion

- Affine Transformation (Anandan)
- Projective Transformation (Mann-Pickard)

Projective Flow

$$u\, f_x + v\, f_y + f_t = 0$$

Optical Flow const. equation

$$\mathbf{u}^T \mathbf{f}_{\mathbf{x}} + f_t = 0$$

$$\mathbf{x}' = \frac{\mathbf{Ax} + \mathbf{b}}{\mathbf{C}^T \mathbf{x} + 1}$$

Projective transform Corrected!!

$$\mathbf{u} = \mathbf{x}' - \mathbf{x} = \frac{\mathbf{Ax} + \mathbf{b}}{\mathbf{C}^T \mathbf{x} + 1} - \mathbf{x}$$

Projective Flow

$$\varepsilon_{flow} = \sum (\mathbf{u}^{\mathrm{T}} \mathbf{f}_{\mathbf{X}} + f_{\mathbf{t}})^2$$

$$(\sum \phi \phi^T)\mathbf{a} = \sum (\mathbf{x}^{\mathrm{T}} \mathbf{f}_x - f_t)\phi$$

$$\mathbf{p} = [a_1, a_2, b_1, a_3, a_4, b_2, c_1, c_2]^T$$

$$\phi^t = [f_x x, f_x y, f_x, f_y x, f_y y, f_y, x f_t - x^2 f_x - xy f_y, y f_t - xy f_x - y^2 f_y]$$

Clustering / Segmentation

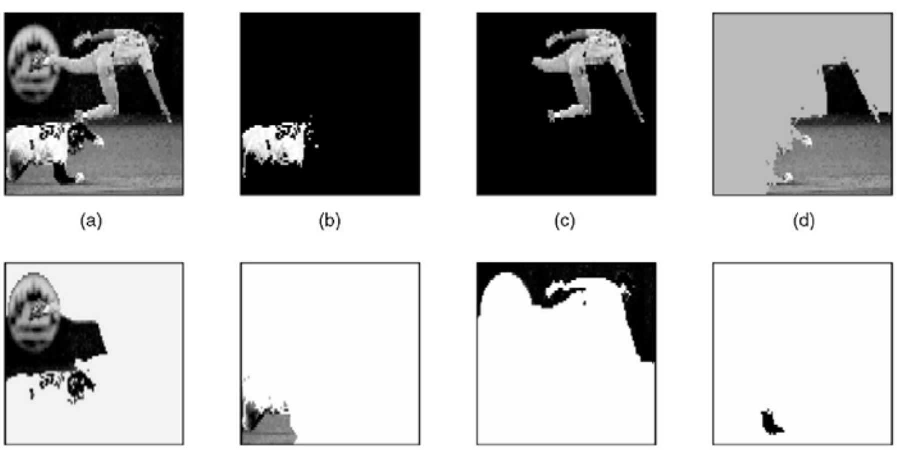
Slide credits go to David Lowe

Segmentation by Clustering

- Data reduction - obtain a compact representation for *interesting* image data in terms of a set of components
- Find components that belong together (form clusters)

Segmentation by Clustering

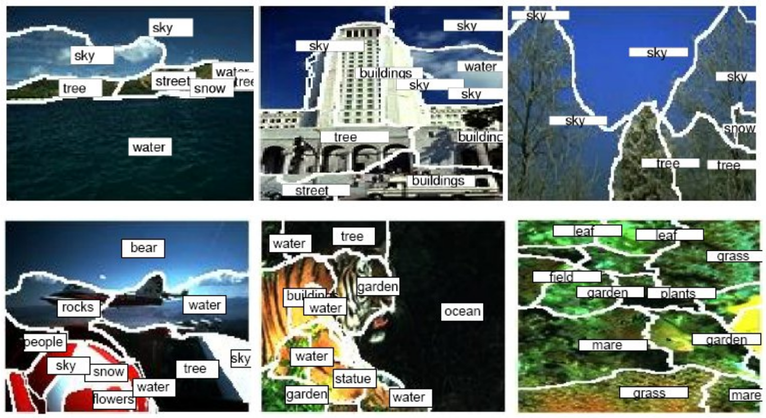
Segmentation by Clustering



(a) (b) (c) (d)

(e) (f) (g) (h)

Segmentation by Clustering



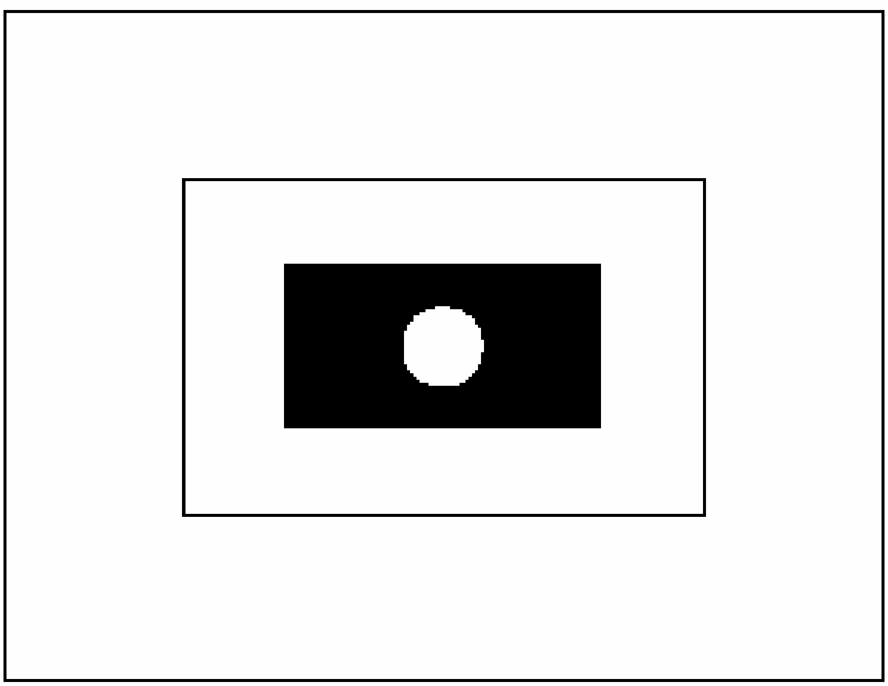
What is Segmentation?

- **Clustering image elements that "belong together"**
  - **Partitioning**
    - Divide into regions/sequences with coherent internal properties
  - **Grouping**
    - Identify sets of coherent tokens in image
- **Tokens:** Whatever we need to group
  - Pixels
  - Features (corners, lines, etc.)
  - Larger regions with uniform colour or texture
  - Discrete objects (e.g., people in a crowd)
  - Etc.

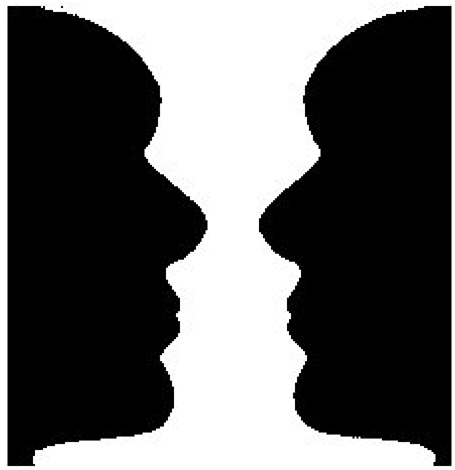
Slide credit: Christopher Rasmussen

Perception

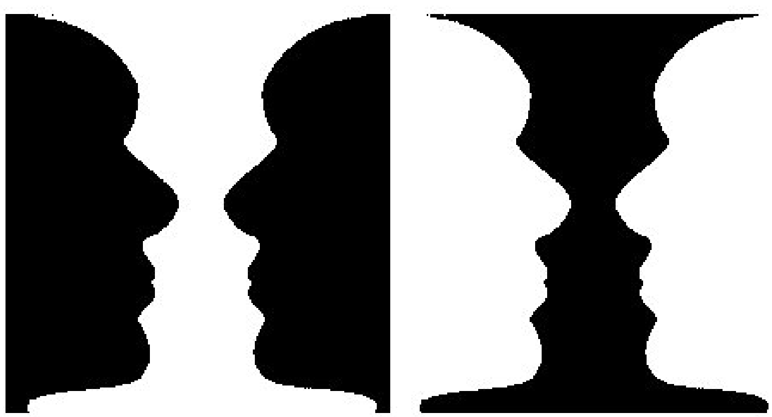
Why do these tokens belong together?



Perception



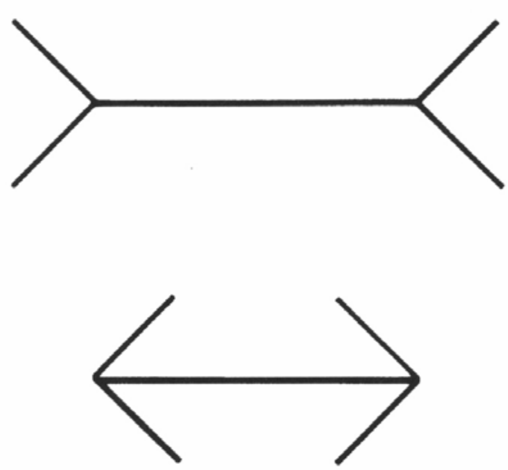
Perception



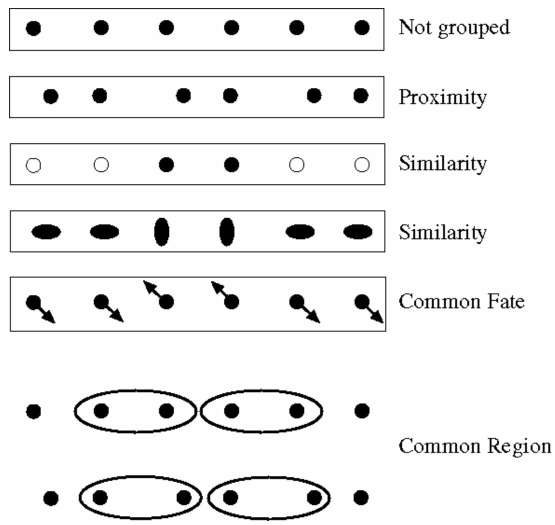
Basic ideas of grouping in human vision

- Figure-ground discrimination
  - allocating some elements to a figure, some to ground
  - local bottom-up cues or high level recognition
- Gestalt properties
  - relationships results in collection of elements (Muller-Lyer effect)
  - A series of factors affect whether elements should be grouped together
    - Gestalt factors

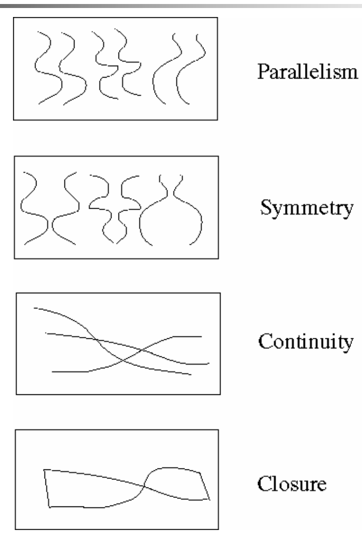
Muller-Lyer Illusion



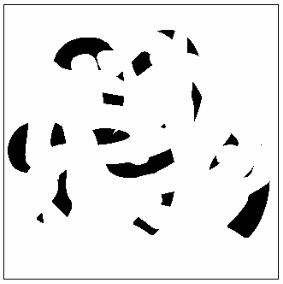
Grouping



Grouping



Can you read 5 numerals?



- Occlusion cues seem to be very important in grouping.

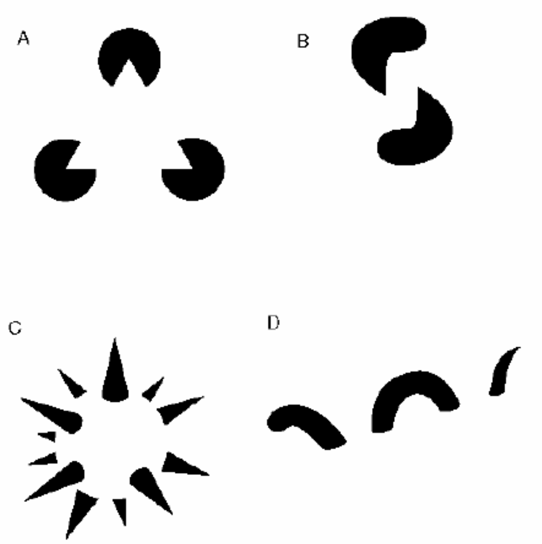
Occlusion

- Now it is easy

Illusion

- A curious phenomenon where you see an object that appears to be occluding

Groupings by Invisible Completions



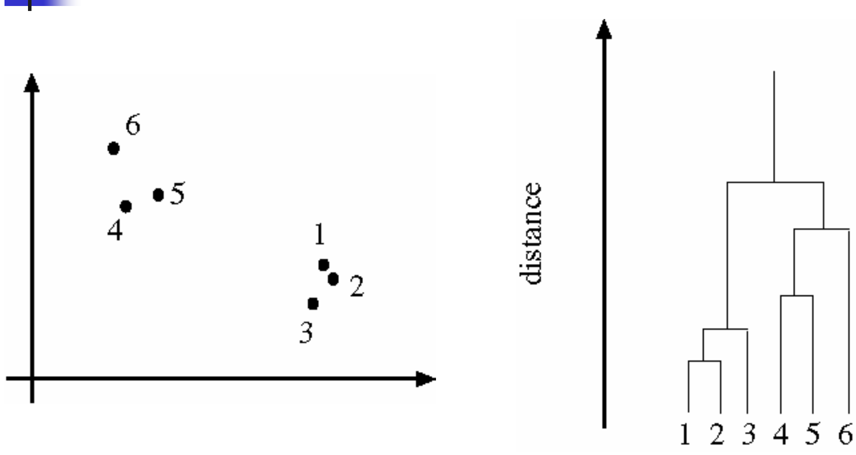
24

* Images from Steve Lehar's Gestalt papers: http://cns-alumni.bu.edu/pub/slehar/Lehar.html

Segmentation as clustering

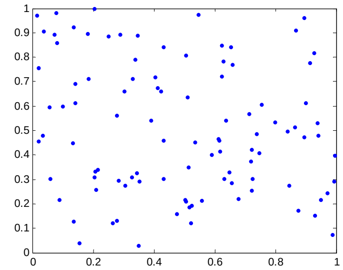
- Cluster pixels, tokens, etc. together
- Agglomerative clustering
  - attach to closest cluster
  - repeat
- Divisive clustering
  - split cluster along best boundary
  - repeat
- Point-Cluster distance
  - single-link clustering
  - complete-link clustering
  - group-average clustering
- Dendrograms
  - generate a picture of output as clustering process continues

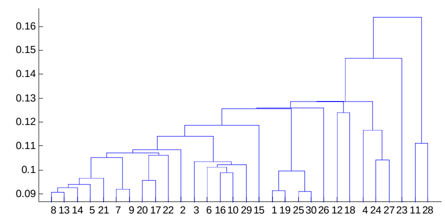
Dendrogram



Matlab Code

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

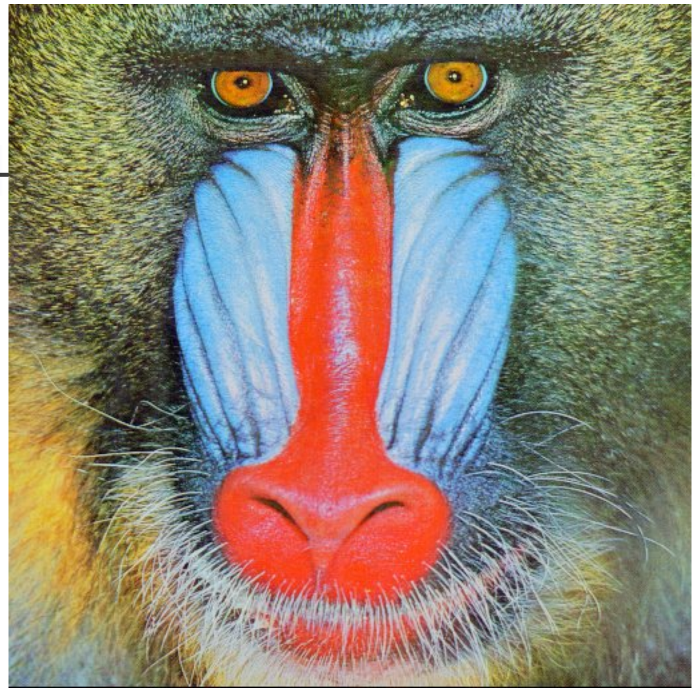


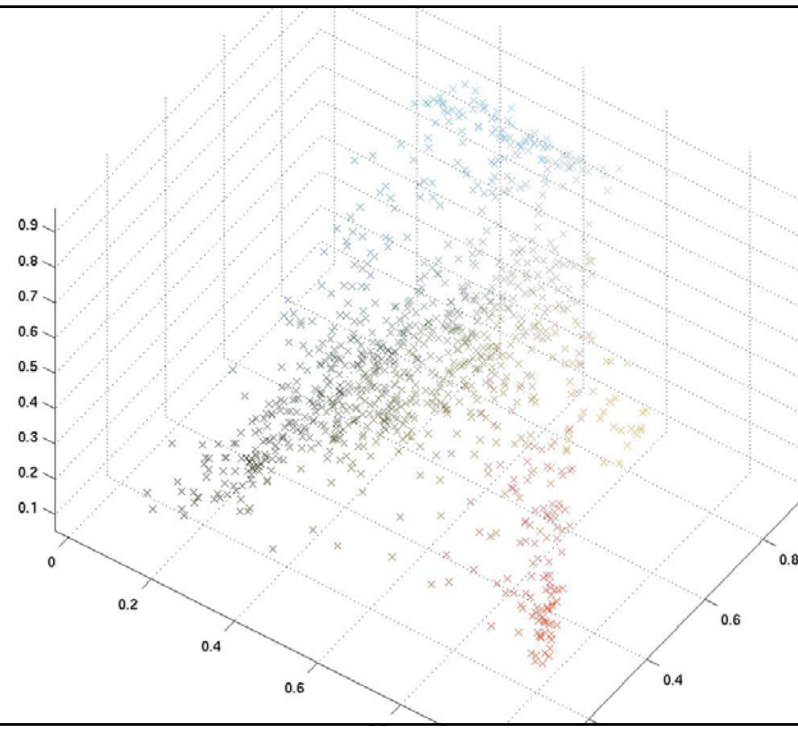


Feature Space

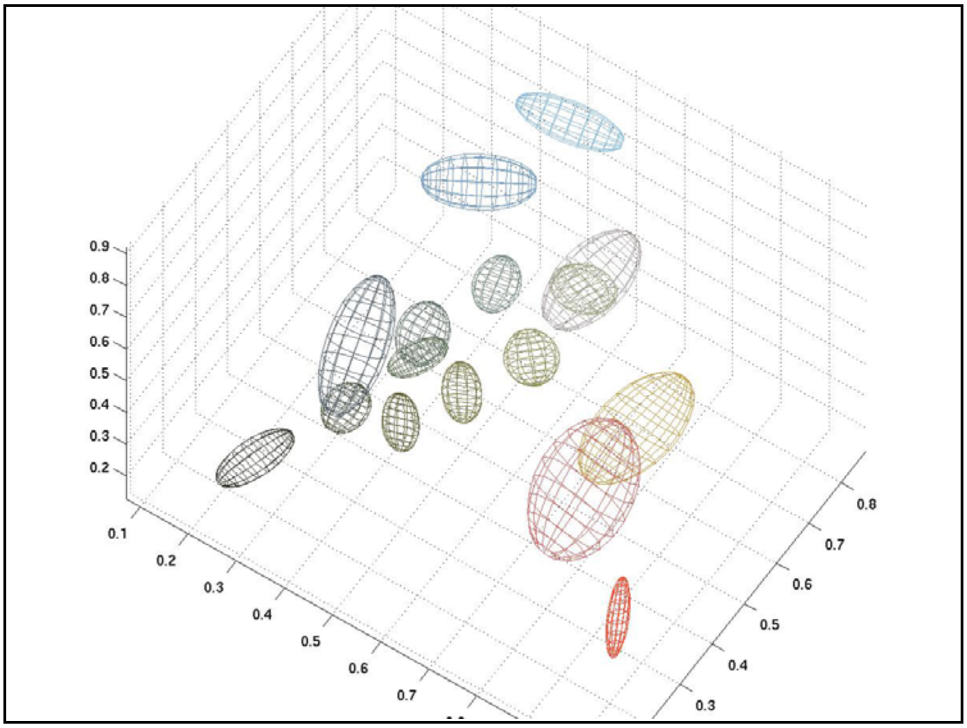
- Every token is identified by a set of salient visual characteristics called *features*.
  - One pixel wide
    - Position, color, texture, motion vector
  - Multiple pixels
    - Size, orientation
- The choice of features and how they are quantified implies a *feature space*
  - Each token is represented by a point
- Token similarity is measured by distance between points (“feature vectors”) in feature space

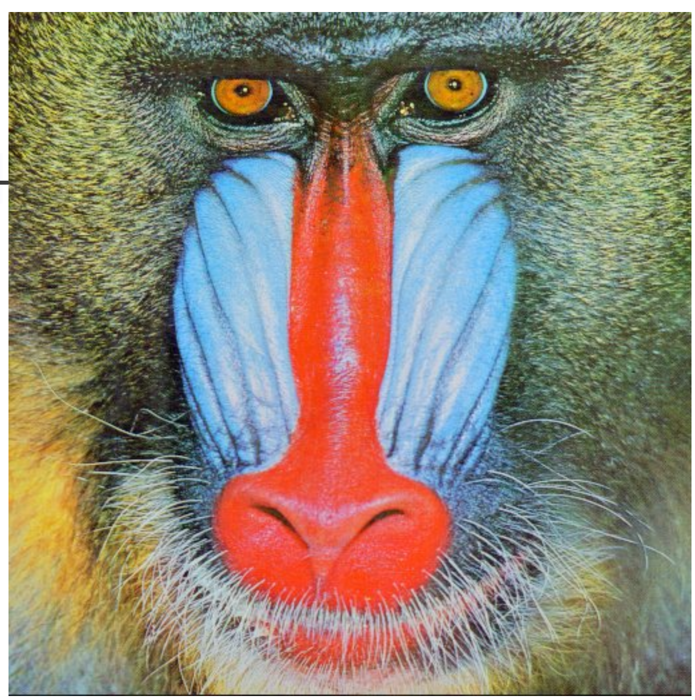
Slide credit: Christopher Rasmussen

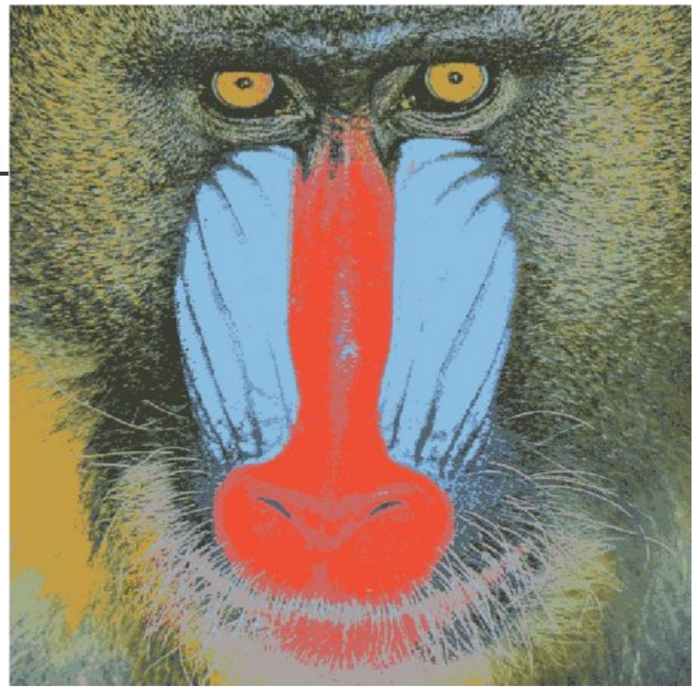


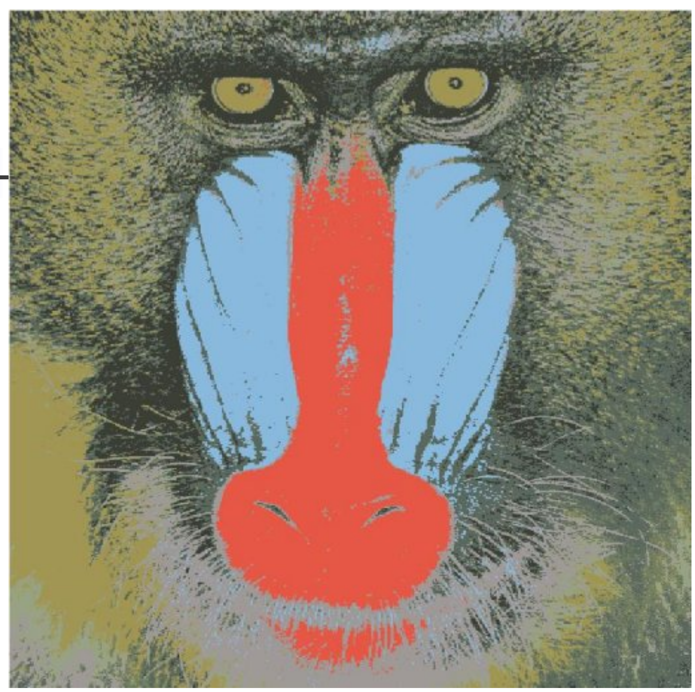


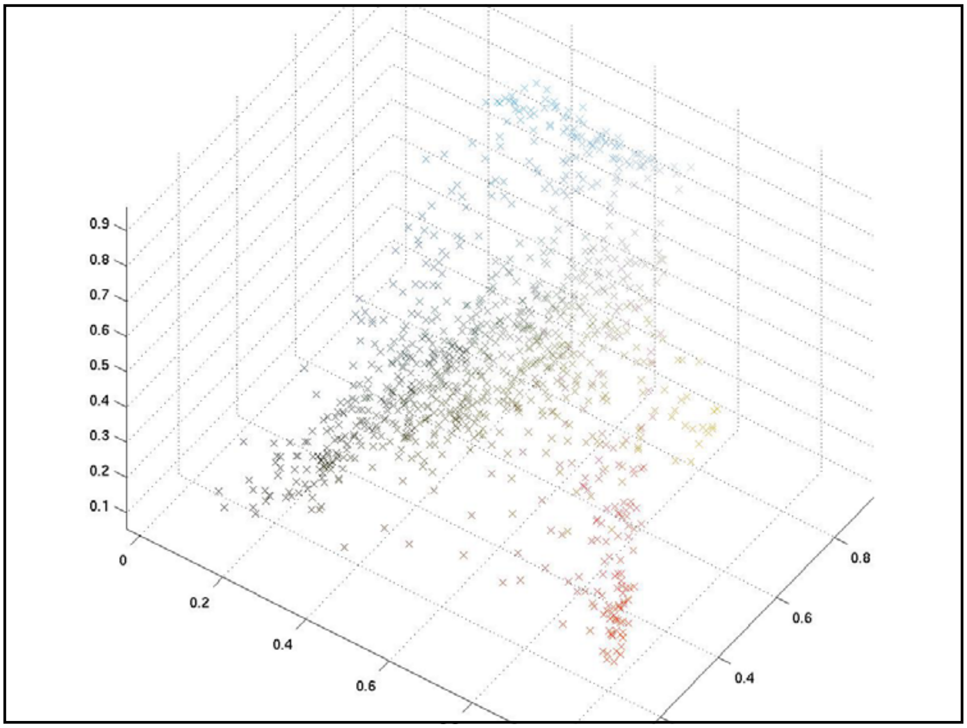
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0.2
0.4
0.6
0.8
0.6
0.4

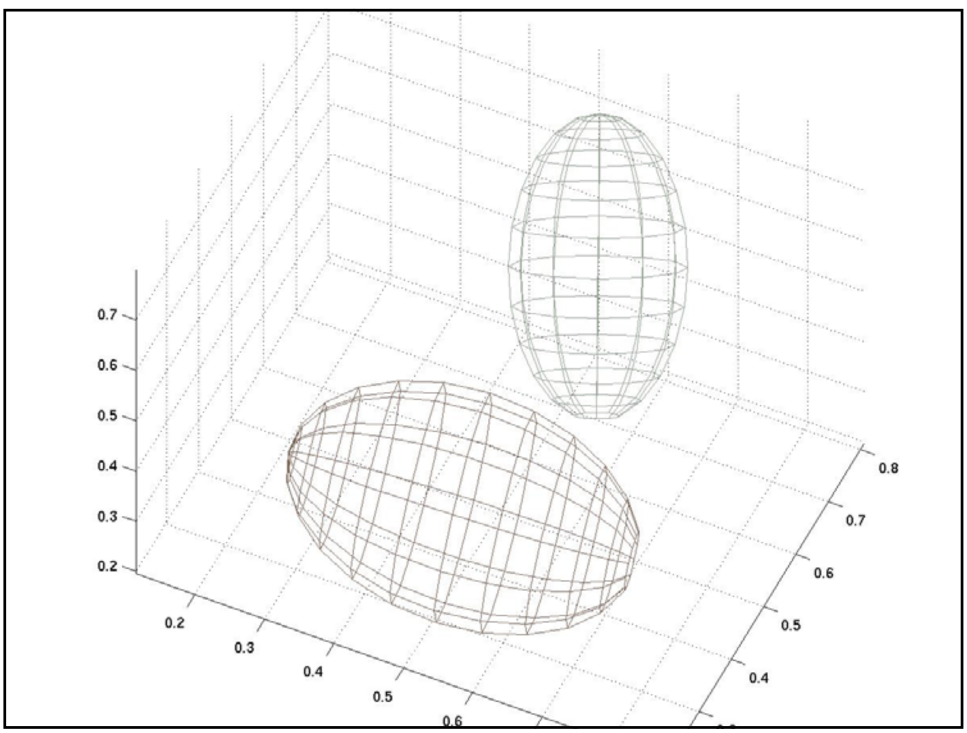

0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.7
0.6
0.5
0.4
0.3








0.9
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


0.7
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

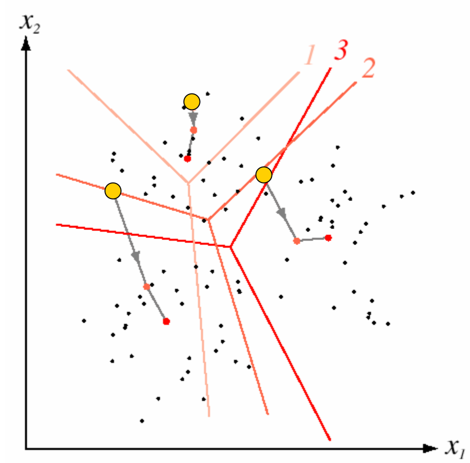
K-Means Clustering

- 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_j, to clusters by nearest m_i (make sure no cluster is empty)
  2. Recompute mean m_i of each cluster from its member points
  3. If no mean has changed more than some ¢, stop
- Effectively carries out gradient descent to minimize:

$$\sum_{i \in \text{clusters}} \left\{ \sum_{j \in \text{elements of i'th cluster}} \left\| x_j - \mu_i \right\|^2 \right\}$$

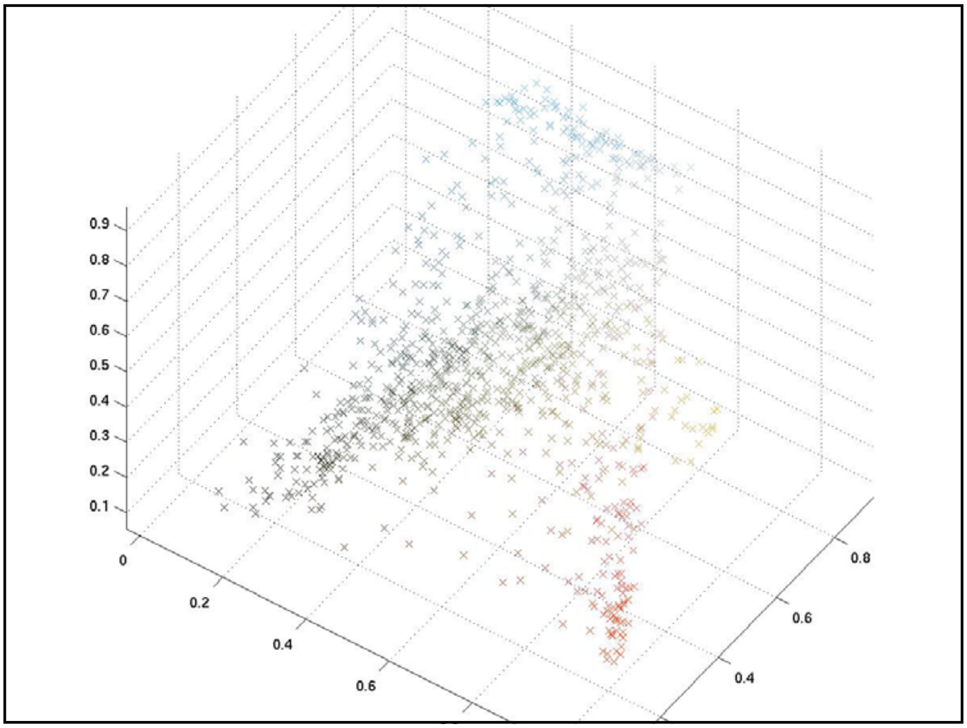
Slide credit: Christopher Rasmussen

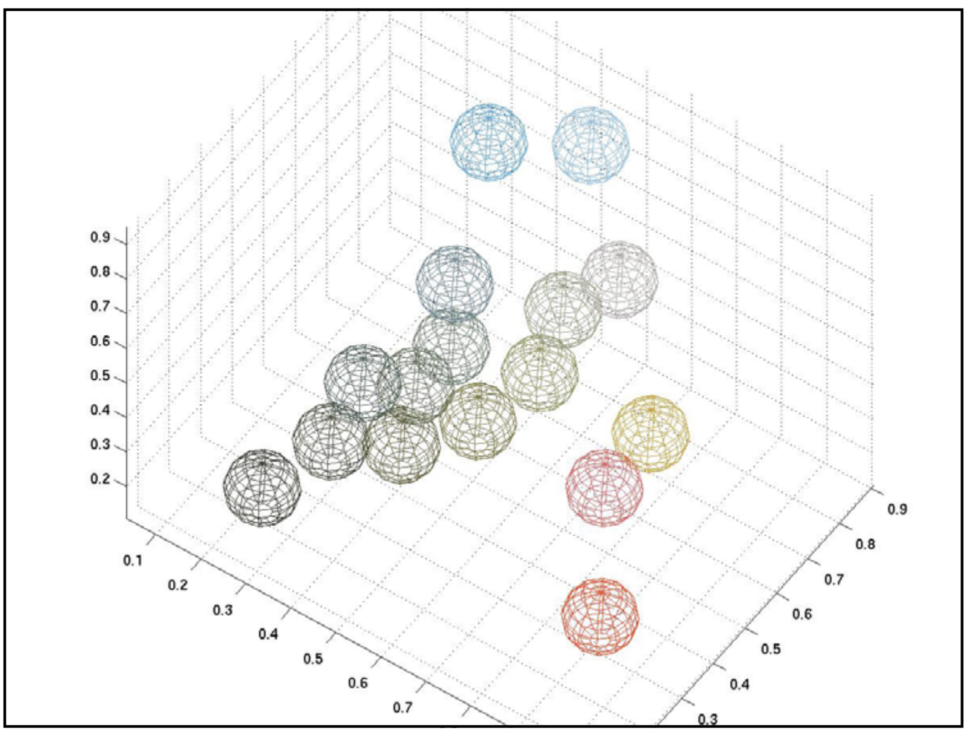
Example: 3-means Clustering

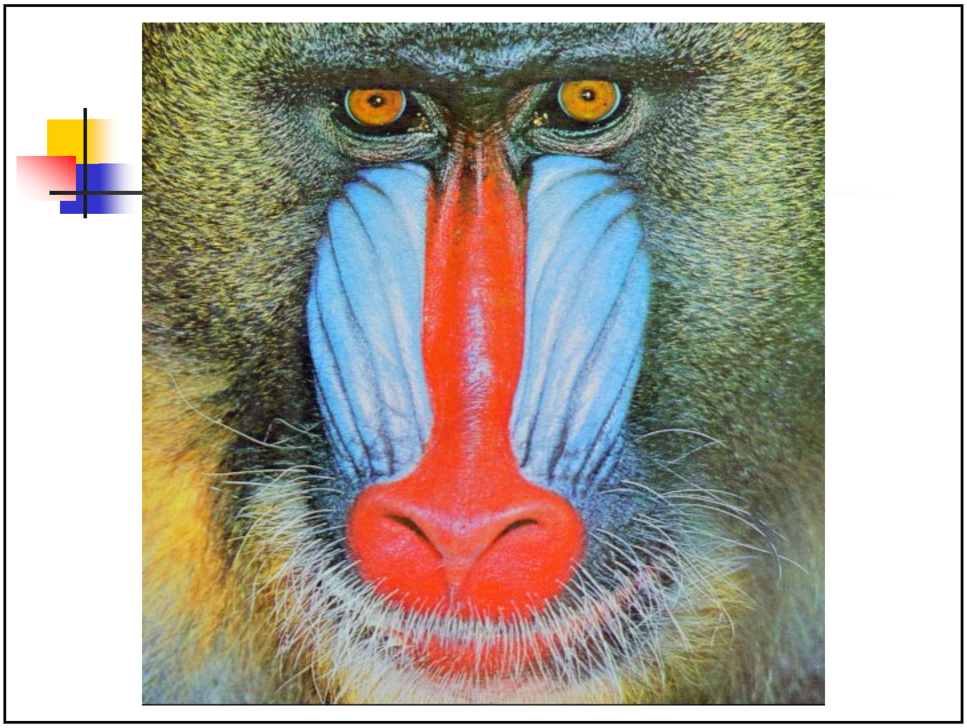


from
Duda et al.

Convergence in 3 steps


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
0.9
0.8
0.7
0.6
0.5
0.4
0.3



K-means clustering using intensity alone and color alone

Image

Clusters on intensity

Clusters on color

K-Means

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

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

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