

BBM 413

Fundamentals of

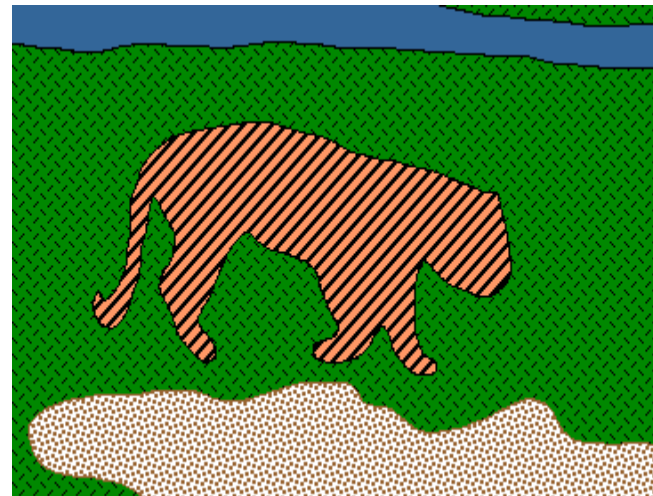
Image Processing

Erkut Erdem
Dept. of Computer Engineering
Hacettepe University

Segmentation – Part I

Image segmentation

- Goal: identify groups of pixels that go together

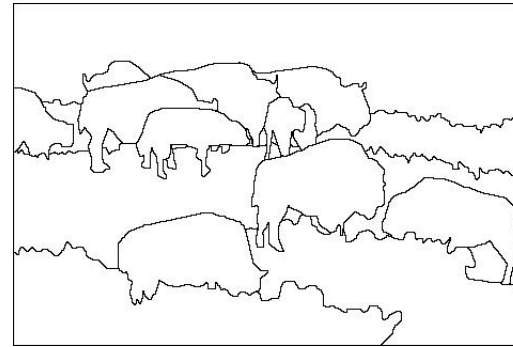


The goals of segmentation

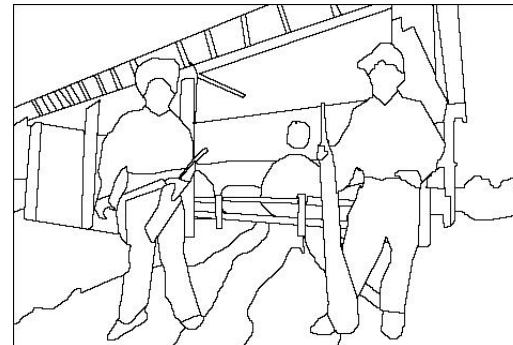
- Separate image into coherent “objects”



image



human segmentation



The goals of segmentation

- Separate image into coherent “objects”
- Group together similar-looking pixels for efficiency of further processing

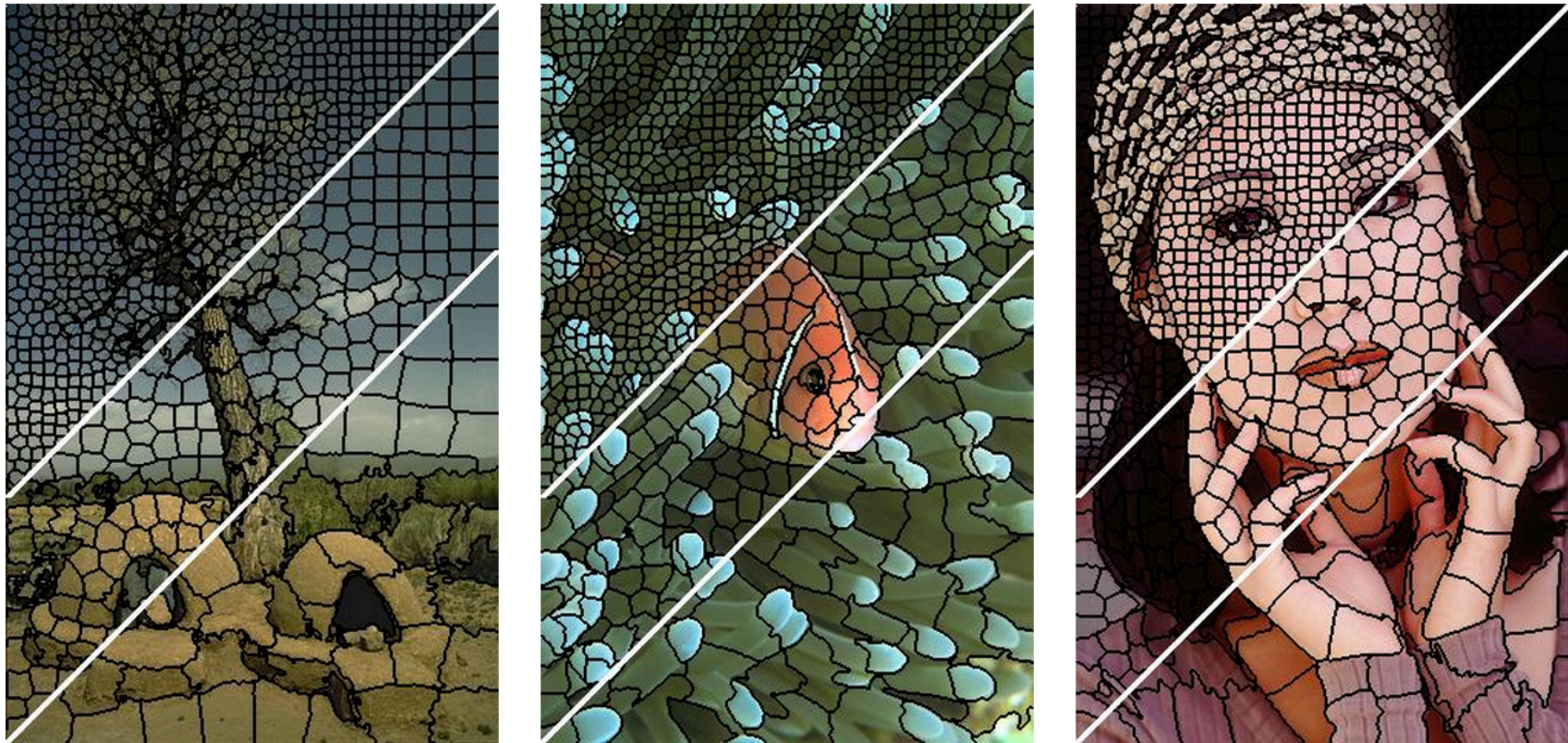
“superpixels”



The goals of segmentation

- Separate image into coherent “objects”
- Group together similar-looking pixels for efficiency of further processing

“superpixels”



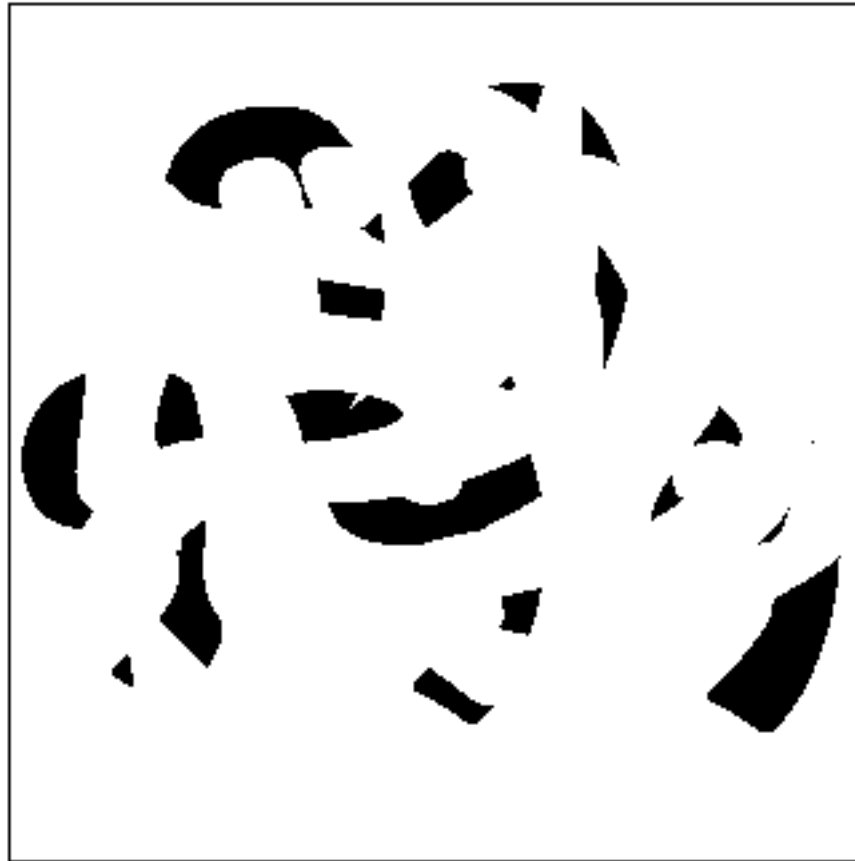
Segmentation

- 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

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

Segmentation is a global process



What are the occluded numbers?

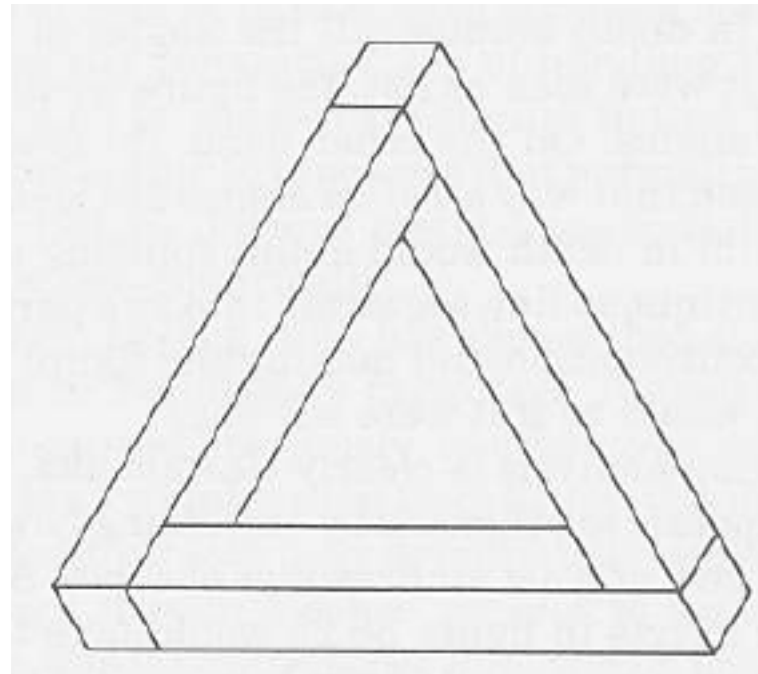
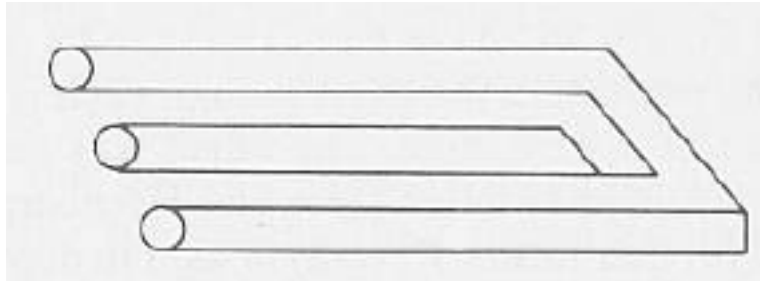
Segmentation is a global process



What are the occluded numbers?

Occlusion is an important cue in grouping.

... but not too global

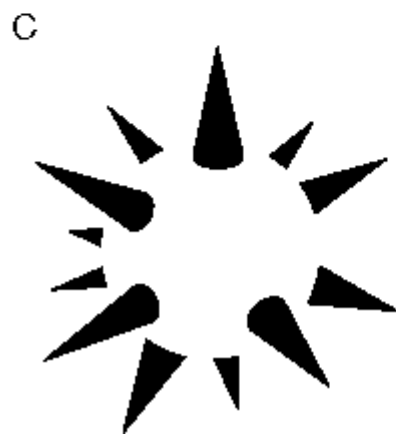
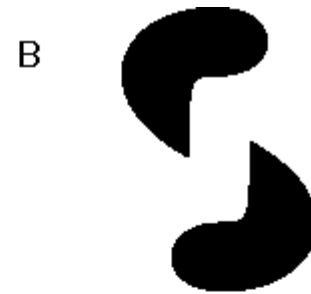
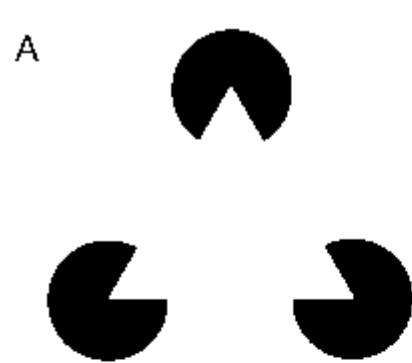




Magritte, 1957

Slide credit: B. Freeman and A. Torralba

Groupings by Invisible Completions



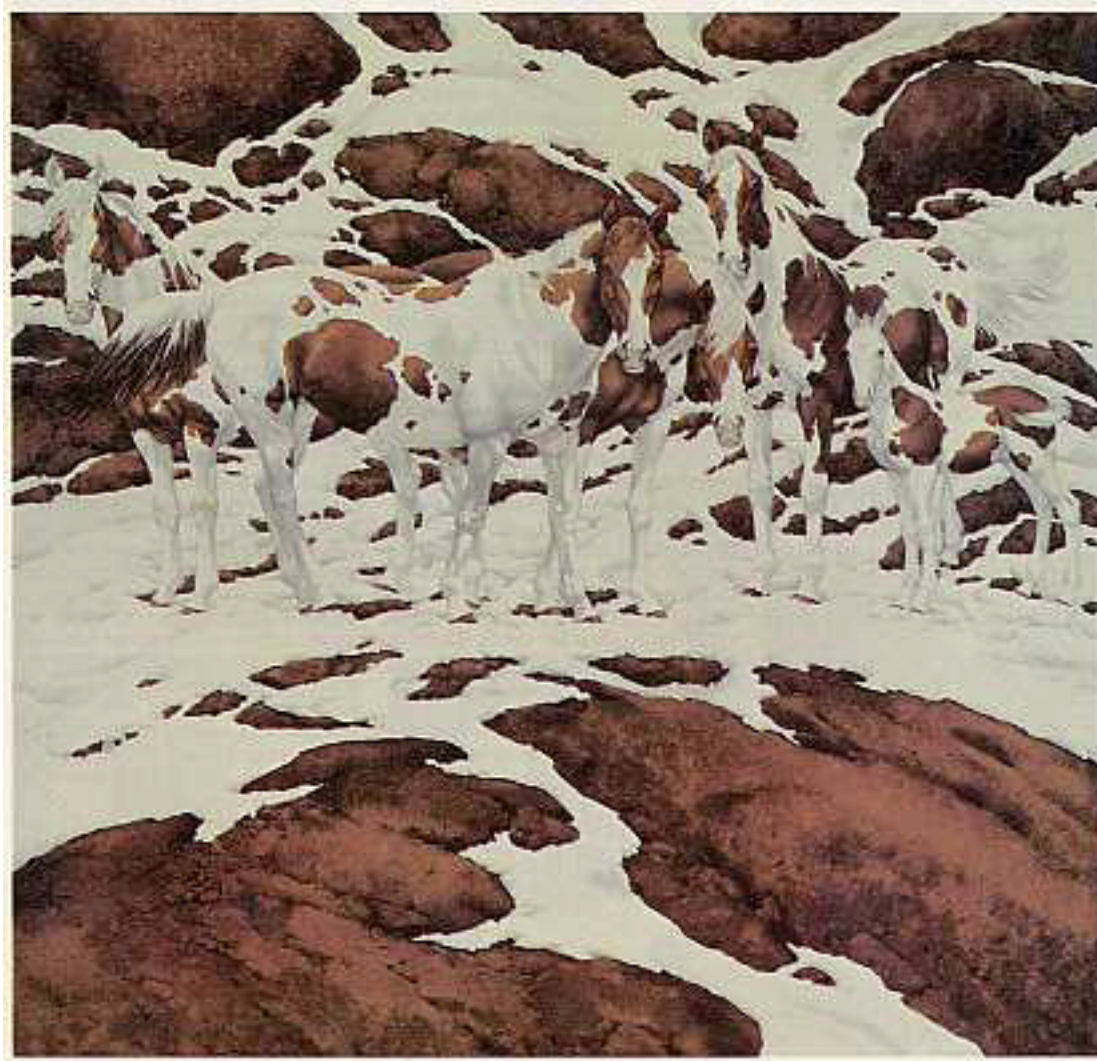
Groupings by Invisible Completions



1970s: R. C. James

Slide credit: B. Freeman and A. Torralba

Groupings by Invisible Completions



2000s: Bev Doolittle

Slide credit: B. Freeman and A. Torralba

Perceptual organization

“...the processes by which the bits and pieces of visual information that are available in the retinal image are structured into the larger units of perceived objects and their interrelations”

Stephen E. Palmer, *Vision Science*, 1999

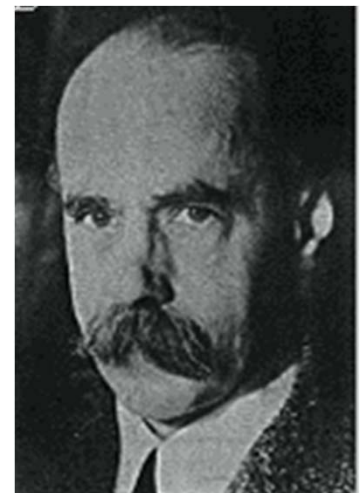


Gestalt Psychology

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

“I stand at the window and see a house, trees, sky. Theoretically I might say there were 327 brightnesses and nuances of colour. Do I have “327”? No. I have sky, house, and trees.”

Max Wertheimer (1880-1943)



Gestalt Psychology

WOLFGANG METZGER

LAWS OF SEEING

Laws of Seeing, Wolfgang Metzger, 1936
(English translation by Lothar Spillmann,
MIT Press, 2006)



Not grouped



Proximity



Similarity



Similarity

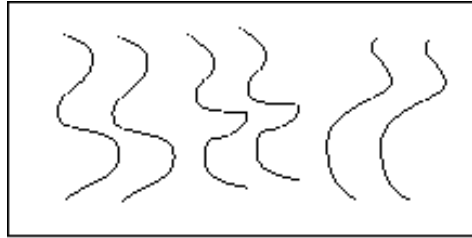


Common Fate

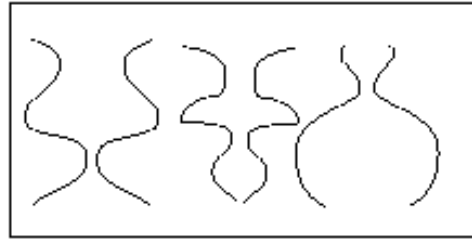


Common Region

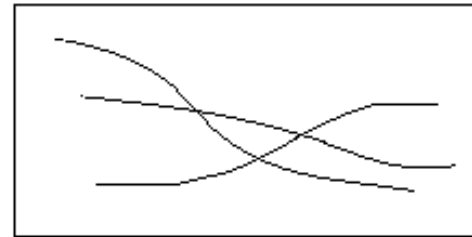




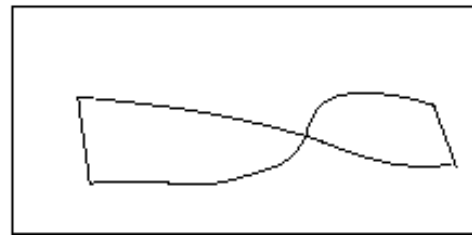
Parallelism



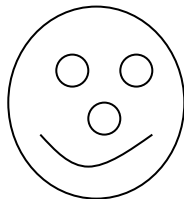
Symmetry



Continuity



Closure

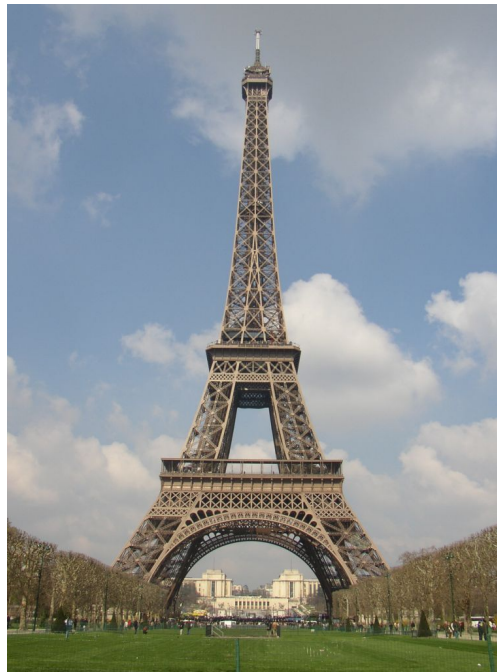


Familiarity

Similarity



Symmetry



Common fate



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



Slide credit: K. Grauman

Proximity

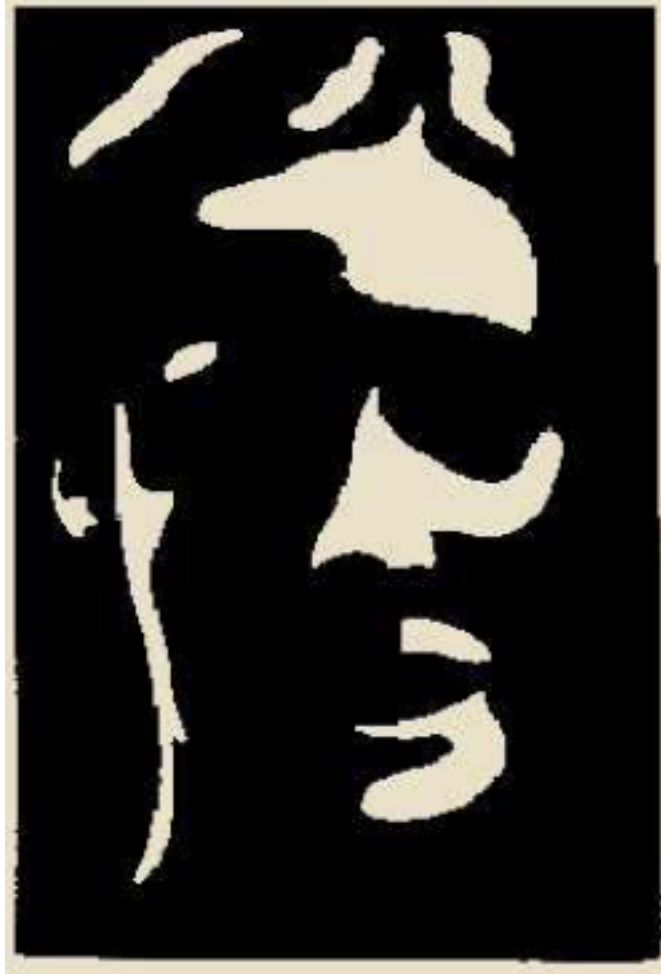


Familiarity



Slide credit: B. Freeman and A. Torralba

Familiarity

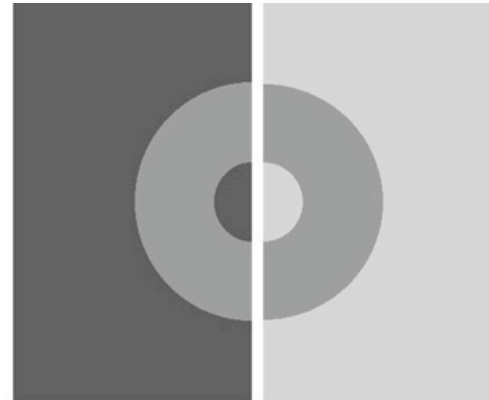


Slide credit: B. Freeman and A. Torralba

Influences of grouping



a



b



c

Grouping influences other perceptual mechanisms such as lightness perception

Emergence



http://en.wikipedia.org/wiki/Gestalt_psychology

Slide credit: S. Lazebnik

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

Segmentation methods

- Segment foreground from background
- Histogram-based segmentation
- Segmentation as clustering
 - K-means clustering
 - Mean-shift segmentation
- Graph-theoretic segmentation
 - Min cut
 - Normalized cuts
- Interactive segmentation

A simple segmentation technique: Background Subtraction

- If we know what the background looks like, it is easy to identify “interesting bits
- Applications
 - Person in an office
 - Tracking cars on a road
 - surveillance
- Approach:
 - use a moving average to estimate background image
 - subtract from current frame
 - large absolute values are interesting pixels
 - trick: use morphological operations to clean up pixels

Movie frames from which we want to extract the foreground subject



Images: Forsyth and Ponce, *Computer Vision: A Modern Approach*

Slide credit: B. Freeman

Two different background removal models

Background estimate
Average over frames

Foreground estimate

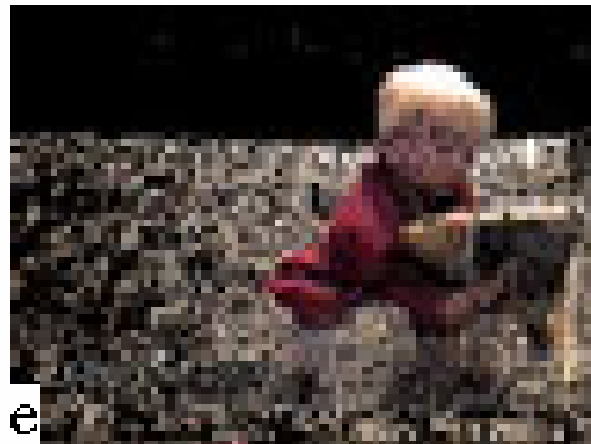
Foreground estimate



EM background estimate

low thresh

high thresh

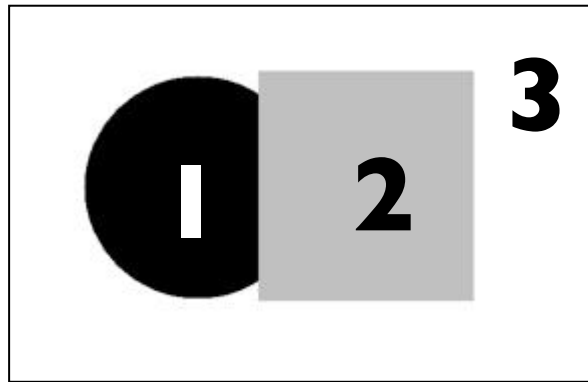


EM

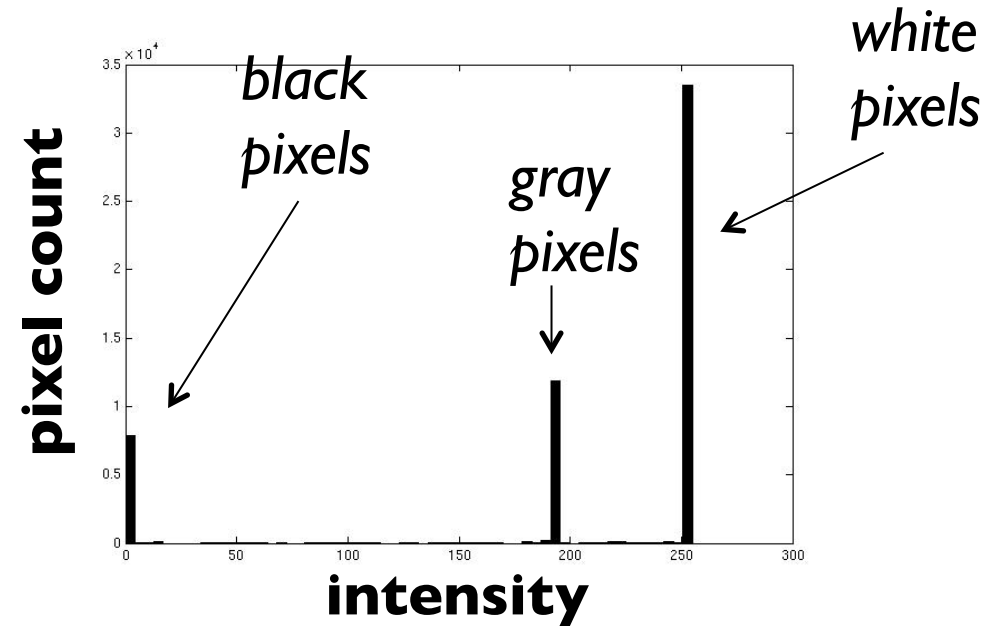
Segmentation methods

- Segment foreground from background
- Histogram-based segmentation
- Segmentation as clustering
 - K-means clustering
 - Mean-shift segmentation
- Graph-theoretic segmentation
 - Min cut
 - Normalized cuts
- Interactive segmentation

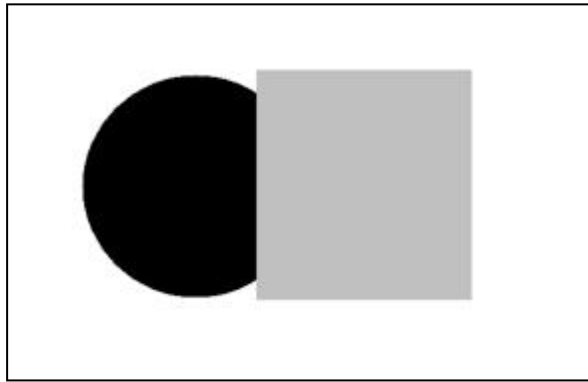
Image segmentation: toy example



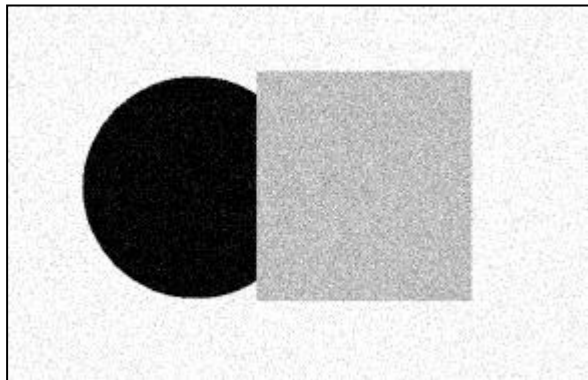
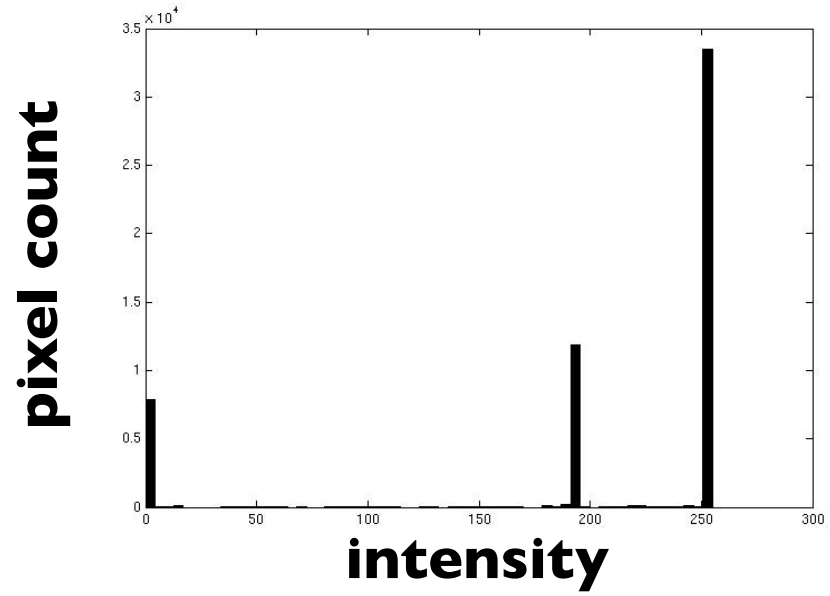
input image



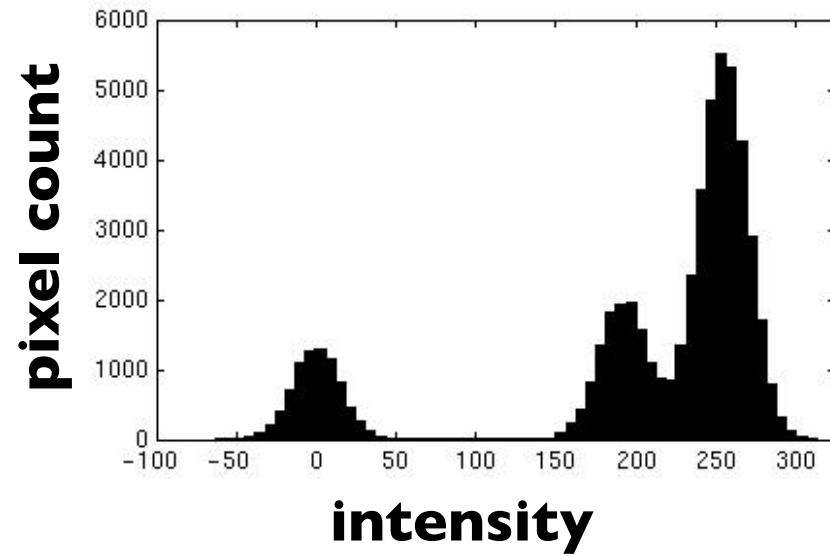
- These intensities define the three groups.
- We could label every pixel in the image according to which of these primary intensities it is.
 - i.e., *segment* the image based on the intensity feature.
- What if the image isn't quite so simple?

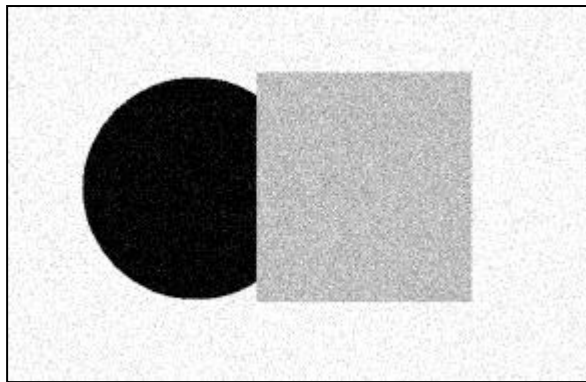


input image

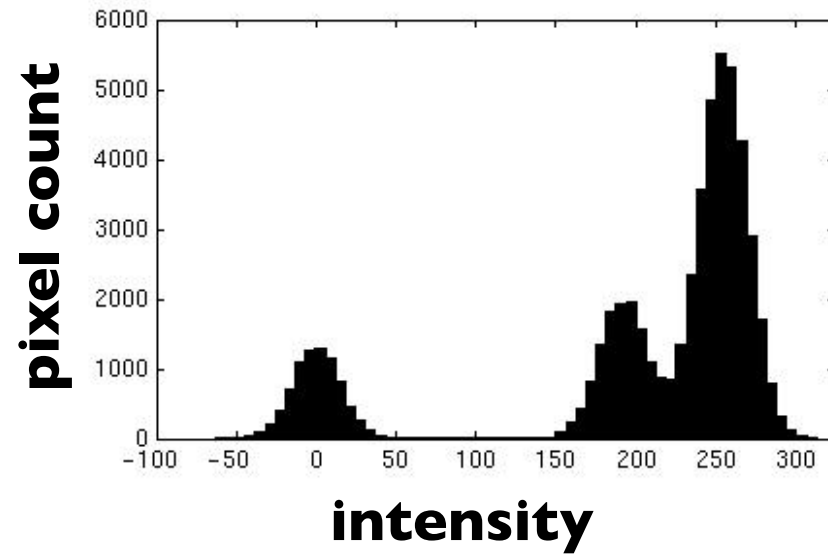


input image

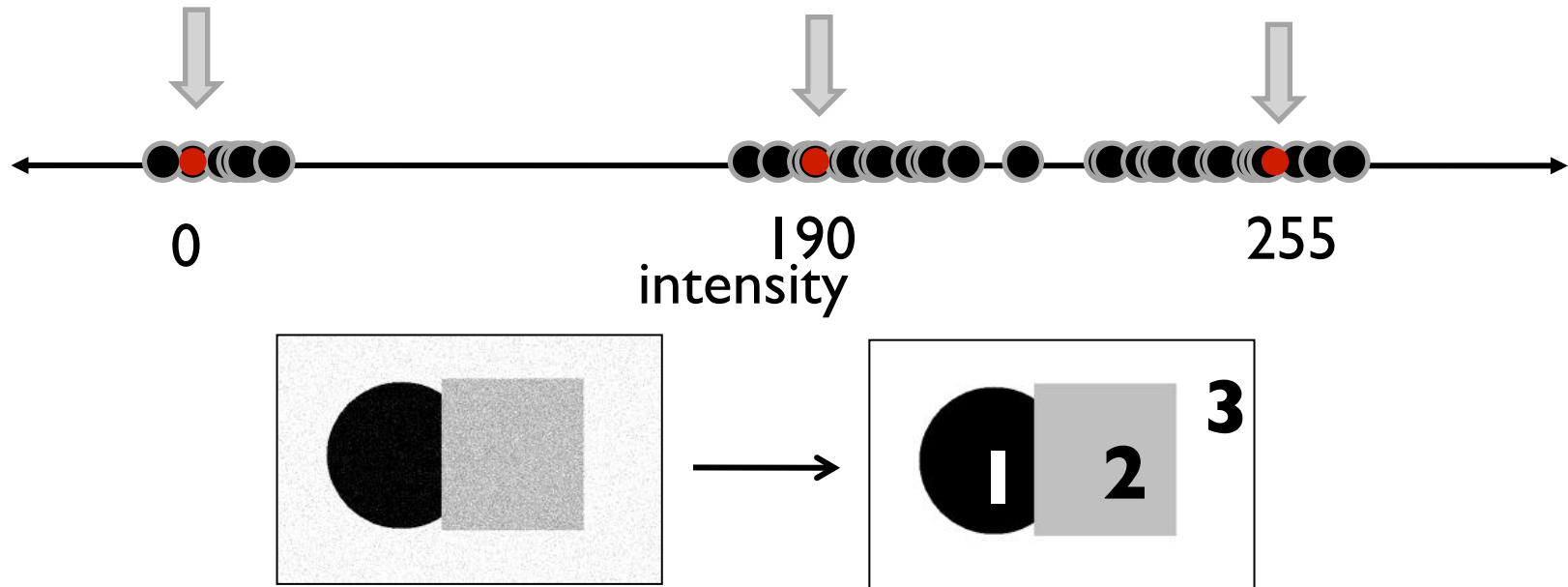




input image



- 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 :

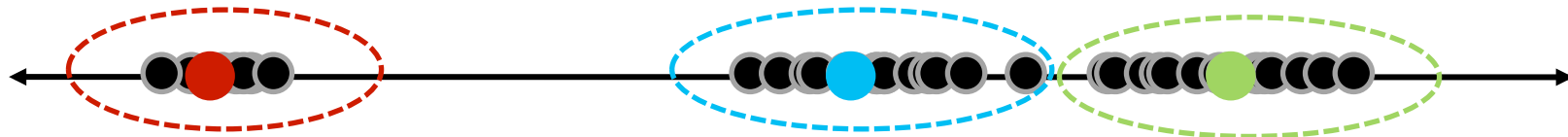
$$\sum_{\text{clusters } i} \sum_{\text{points } p \text{ in cluster } i} \|p - c_i\|^2$$

Segmentation methods

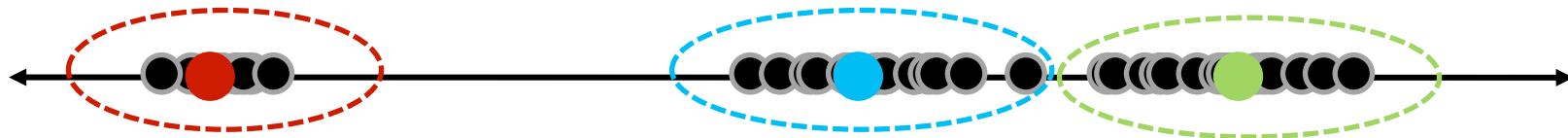
- Segment foreground from background
- Histogram-based segmentation
- **Segmentation as clustering**
 - K-means clustering
 - Mean-shift segmentation
- Graph-theoretic segmentation
 - Min cut
 - Normalized cuts
- Interactive segmentation

Clustering

- With this objective, it is a “chicken and egg” problem:
 - If we knew the **cluster centers**, we could allocate points to groups by assigning each to its closest center.



- If we knew the **group memberships**, we could get the centers by computing the mean per group.



Segmentation as clustering

- Cluster together (pixels, tokens, etc.) that belong together...
- Agglomerative clustering
 - attach closest to cluster it is closest to – repeat
- Divisive clustering
 - split cluster along best boundary – repeat
- Dendrograms
 - yield a picture of output as clustering process continues

Greedy Clustering Algorithms

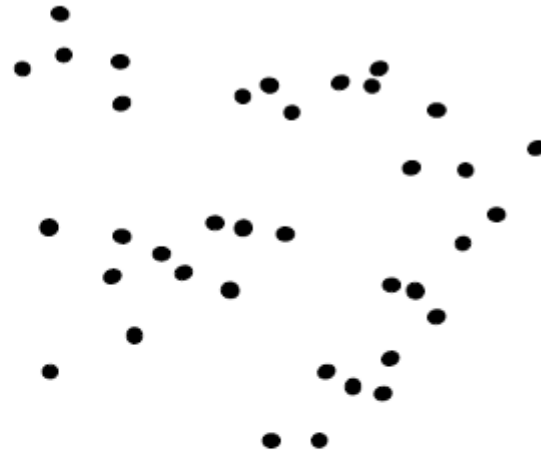
Algorithm 15.3: Agglomerative clustering, or clustering by merging

```
Make each point a separate cluster
Until the clustering is satisfactory
    Merge the two clusters with the
        smallest inter-cluster distance
end
```

Algorithm 15.4: Divisive clustering, or clustering by splitting

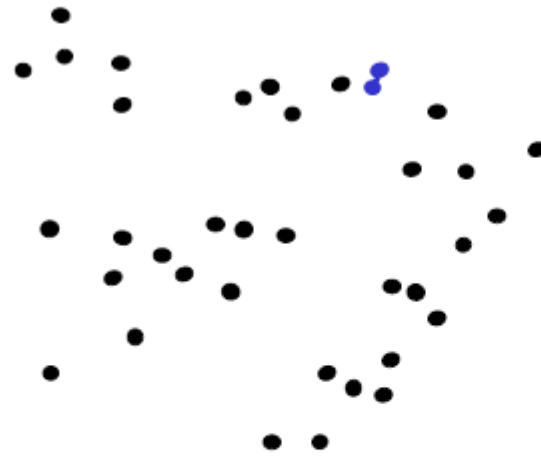
```
Construct a single cluster containing all points
Until the clustering is satisfactory
    Split the cluster that yields the two
        components with the largest inter-cluster distance
end
```

Agglomerative clustering



1. Say "Every point is its own cluster"

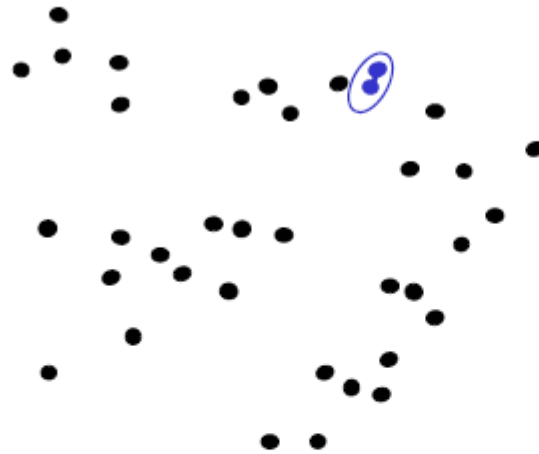
Agglomerative clustering



1. Say "Every point is its own cluster"
2. Find "most similar" pair of clusters



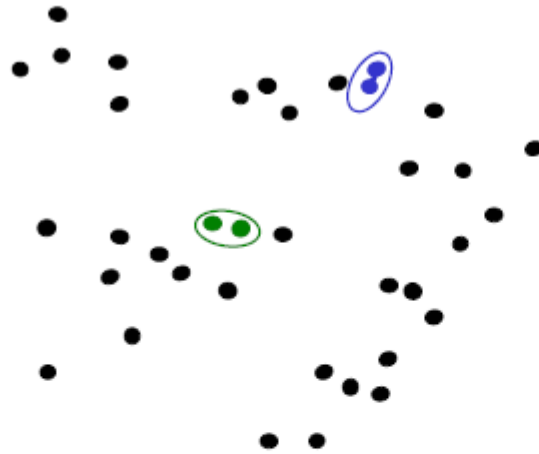
Agglomerative clustering



1. Say "Every point is its own cluster"
2. Find "most similar" pair of clusters
3. Merge it into a parent cluster



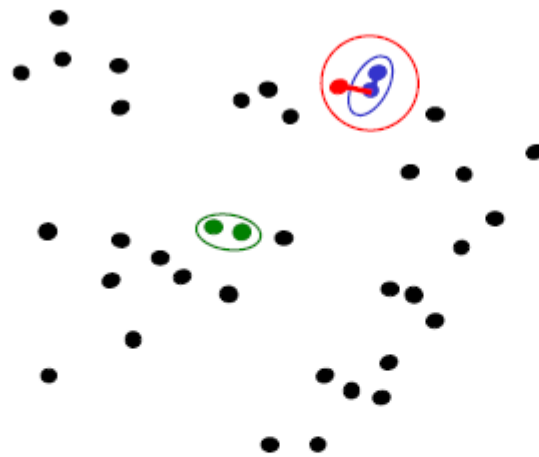
Agglomerative clustering



1. Say "Every point is its own cluster"
2. Find "most similar" pair of clusters
3. Merge it into a parent cluster
4. Repeat



Agglomerative clustering



1. Say "Every point is its own cluster"
2. Find "most similar" pair of clusters
3. Merge it into a parent cluster
4. Repeat



Copyright © 2001, 2004, Andrew W. Moore

K-means and Hierarchical Clustering: Slide 44

Slide credit: D. Hoiem

Common similarity/distance measures

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

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

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

$$\|\mathbf{x}\| := \sqrt{x_1^2 + \dots + x_n^2}$$

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

Here x_i is the distance btw. two points

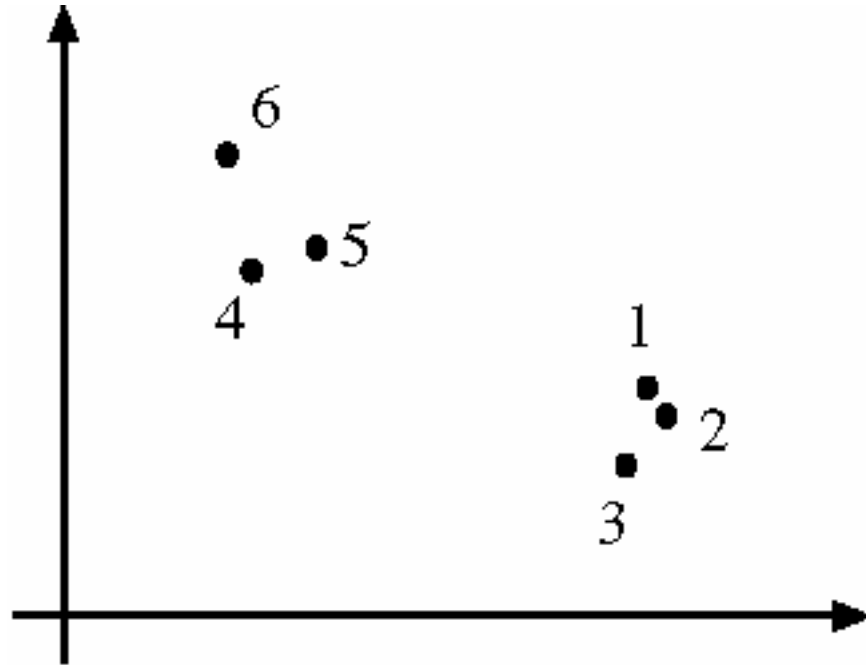
- Mahalanobis
 - Scaled Euclidean

$$d(\vec{x}, \vec{y}) = \sqrt{\sum_{i=1}^N \frac{(x_i - y_i)^2}{\sigma_i^2}}$$

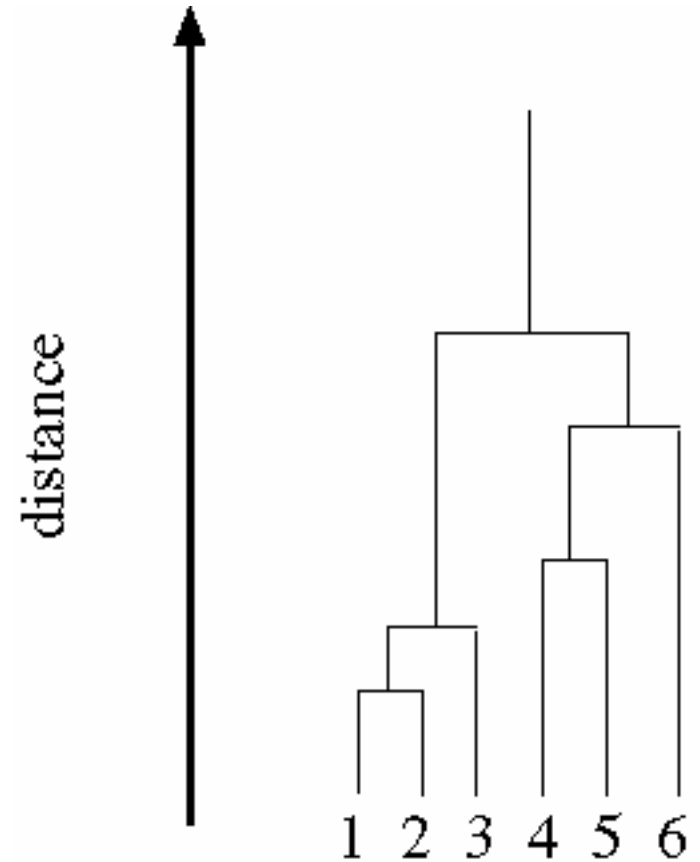
- Cosine distance

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

Dendograms



Data set

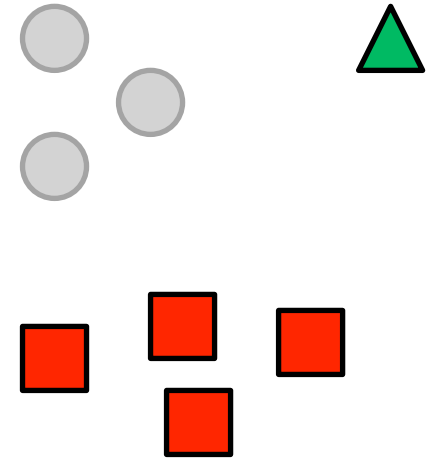


Dendrogram formed by agglomerative clustering using single-link clustering.

Agglomerative clustering

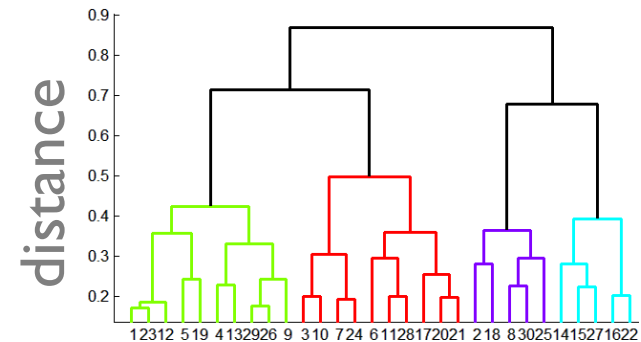
How to define cluster similarity?

- Average distance between points, maximum distance, minimum distance
- Distance between means or medoids



How many clusters?

- Clustering creates a dendrogram (a tree)
- Threshold based on max number of clusters or based on distance between merges



Agglomerative clustering

Good

- Simple to implement, widespread application
- Clusters have adaptive shapes
- Provides a hierarchy of clusters

Bad

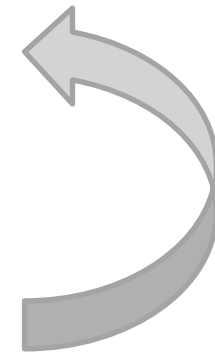
- May have imbalanced clusters
- Still have to choose number of clusters or threshold
- Need to use an “ultrametric” to get a meaningful hierarchy

Segmentation methods

- Segment foreground from background
- Histogram-based segmentation
- **Segmentation as clustering**
 - K-means clustering
 - Mean-shift segmentation
- Graph-Theoretic Segmentation
 - Min cut
 - Normalized cuts

K-means clustering

- Basic idea: randomly initialize the k cluster centers, and iterate between the two steps we just saw.
 1. Randomly initialize the cluster centers, c_1, \dots, c_k
 2. Given cluster centers, determine points in each cluster
 - For each point p , find the closest c_i . Put p into cluster i
 3. Given points in each cluster, solve for c_i
 - Set c_i to be the mean of points in cluster i
 4. If c_i have changed, repeat Step 2



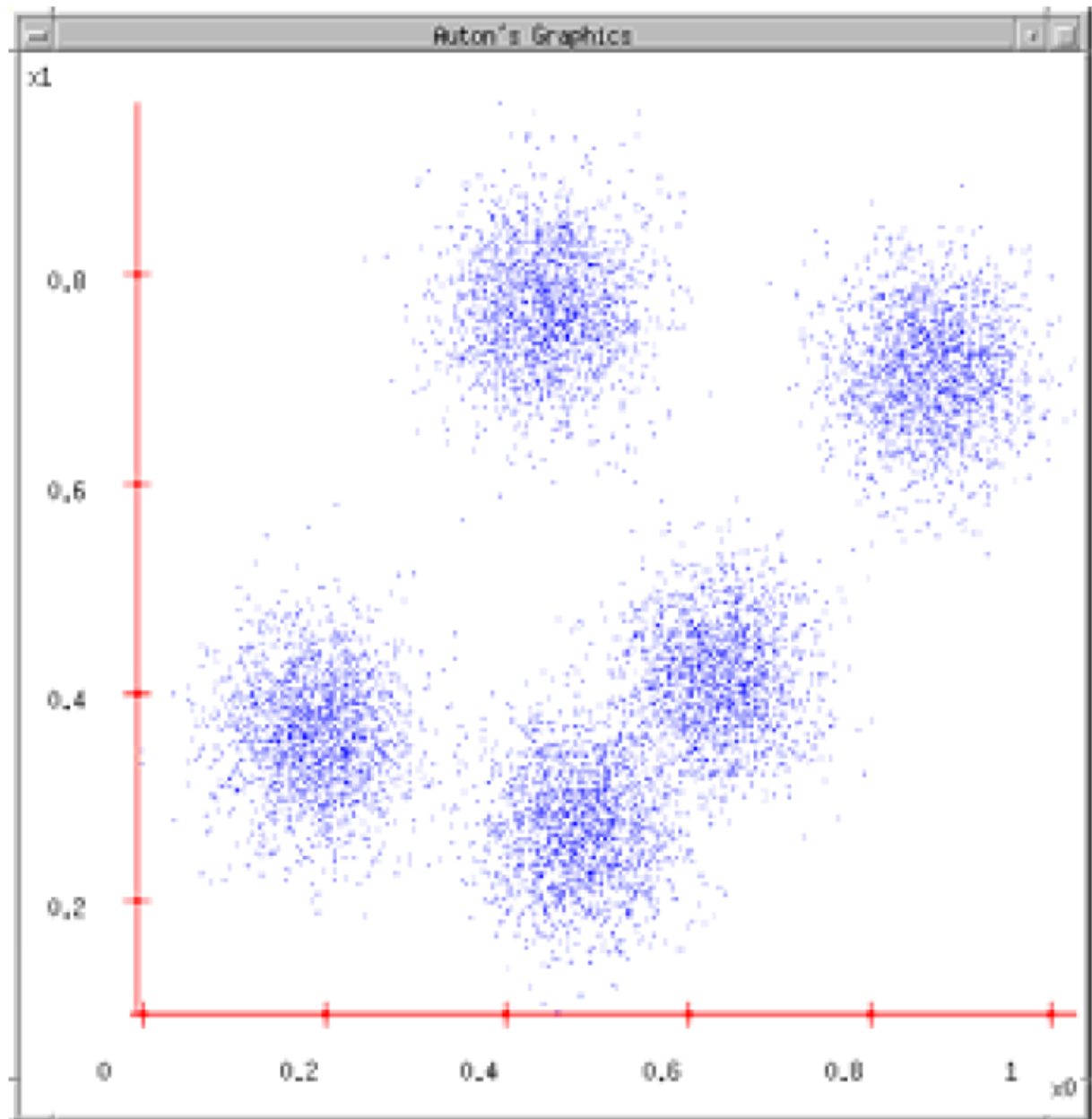
Properties

- Will always converge to *some* solution
- Can be a “local minimum”
 - does not always find the global minimum of objective function:

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

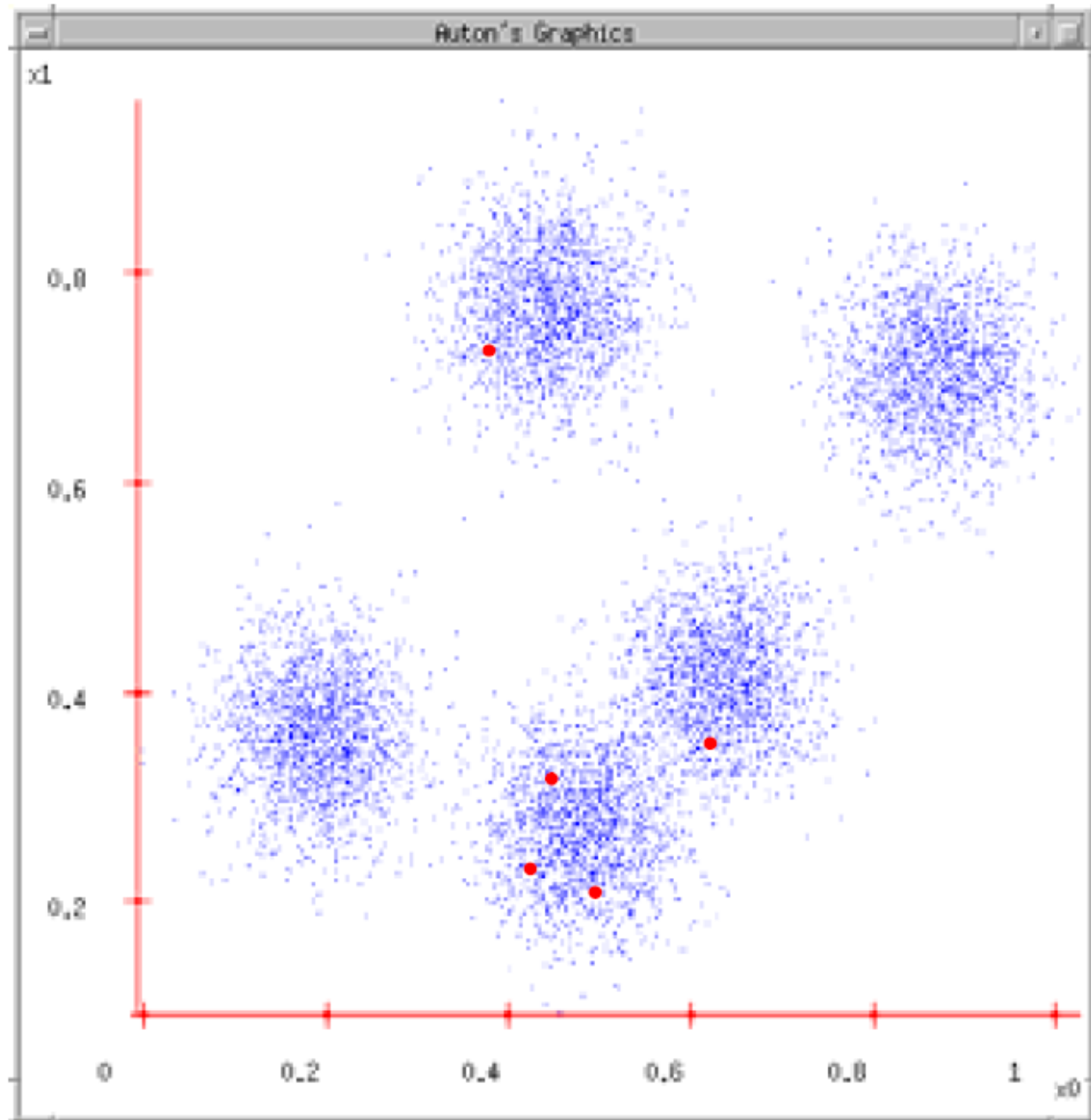
K-means

1. Ask user how many clusters they'd like.
(e.g. $k=5$)



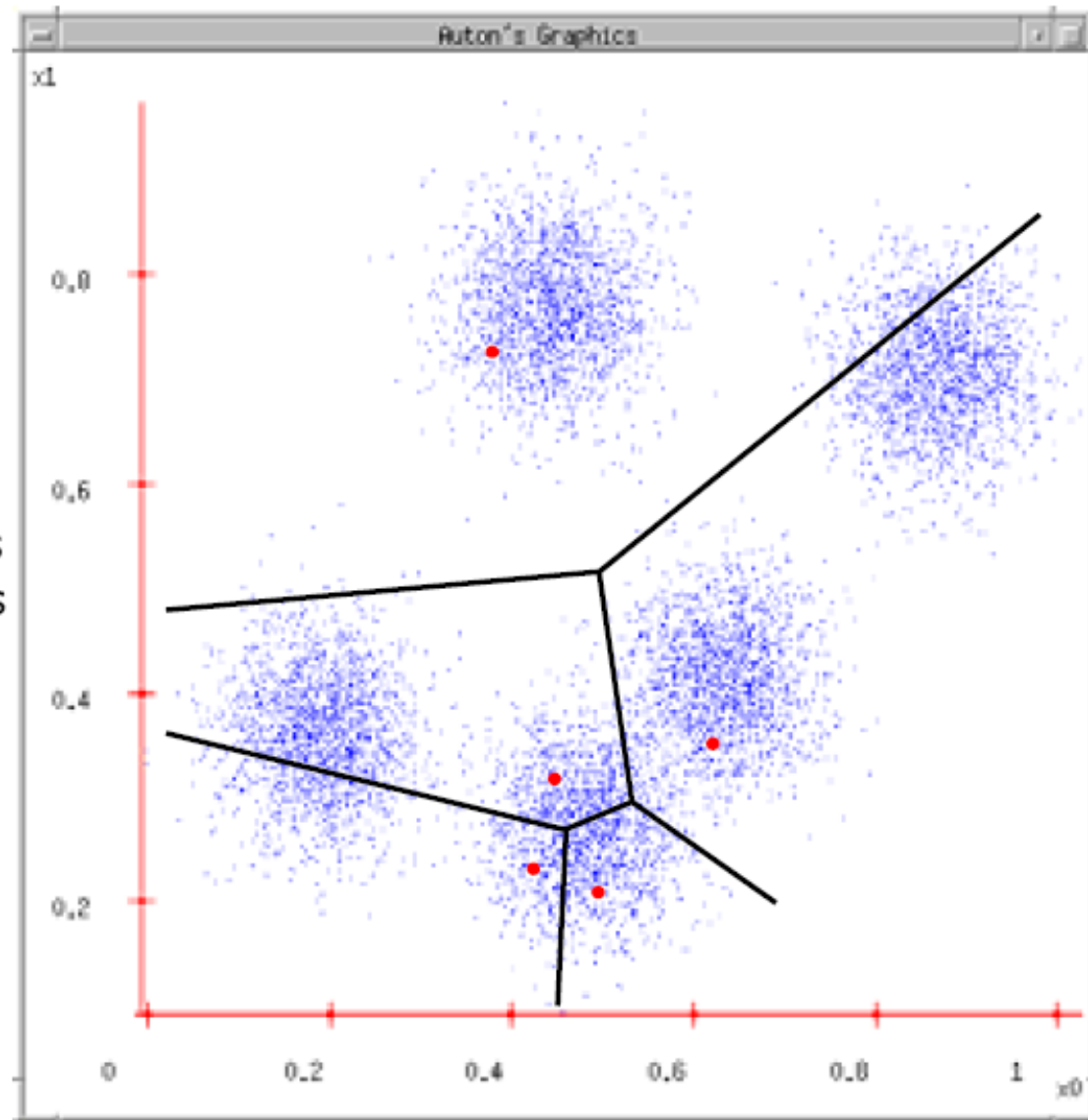
K-means

1. Ask user how many clusters they'd like.
(e.g. $k=5$)
2. Randomly guess k cluster Center locations



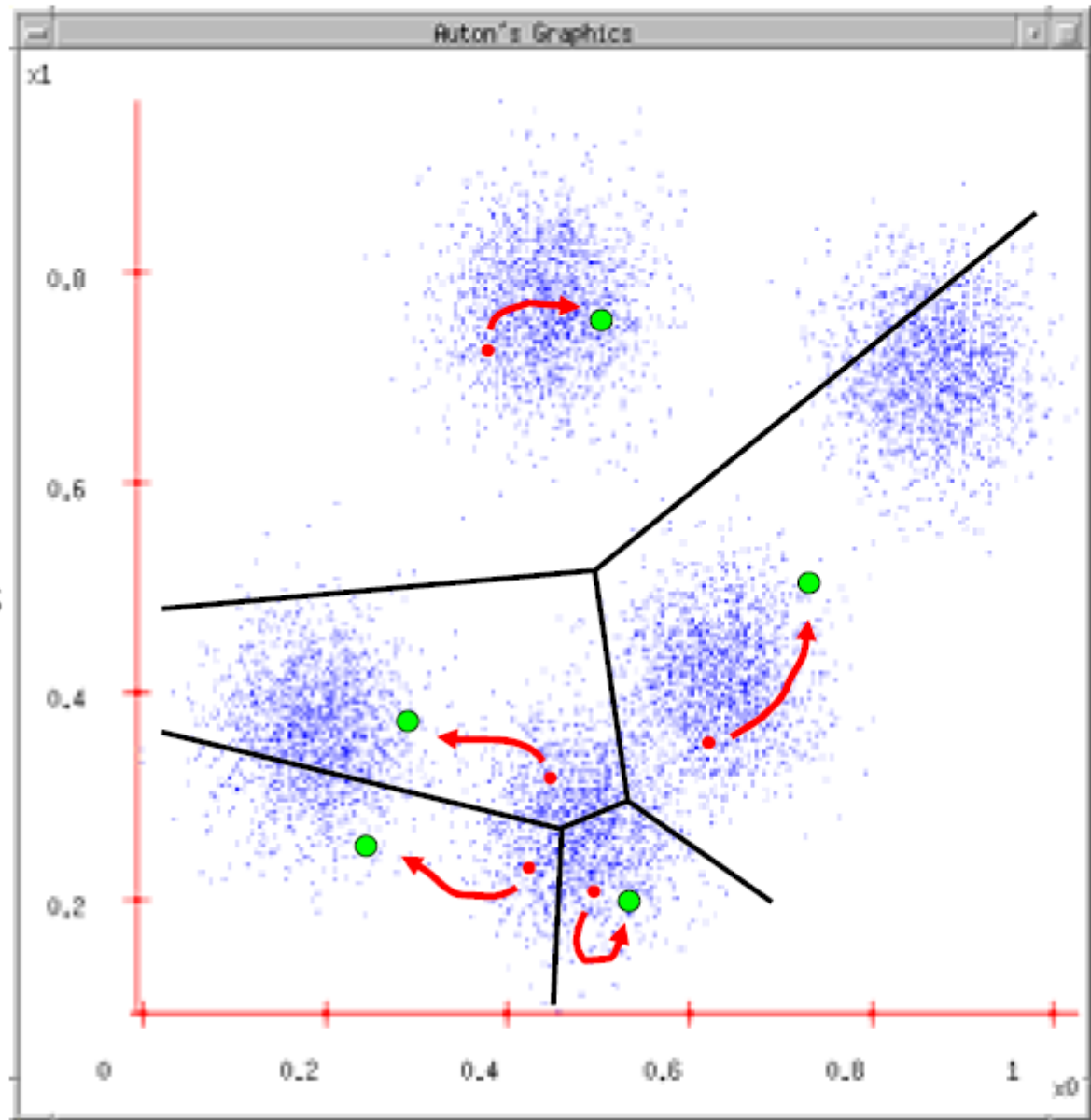
K-means

1. Ask user how many clusters they'd like. (e.g. $k=5$)
2. Randomly guess k cluster Center locations
3. Each datapoint finds out which Center it's closest to. (Thus each Center "owns" a set of datapoints)



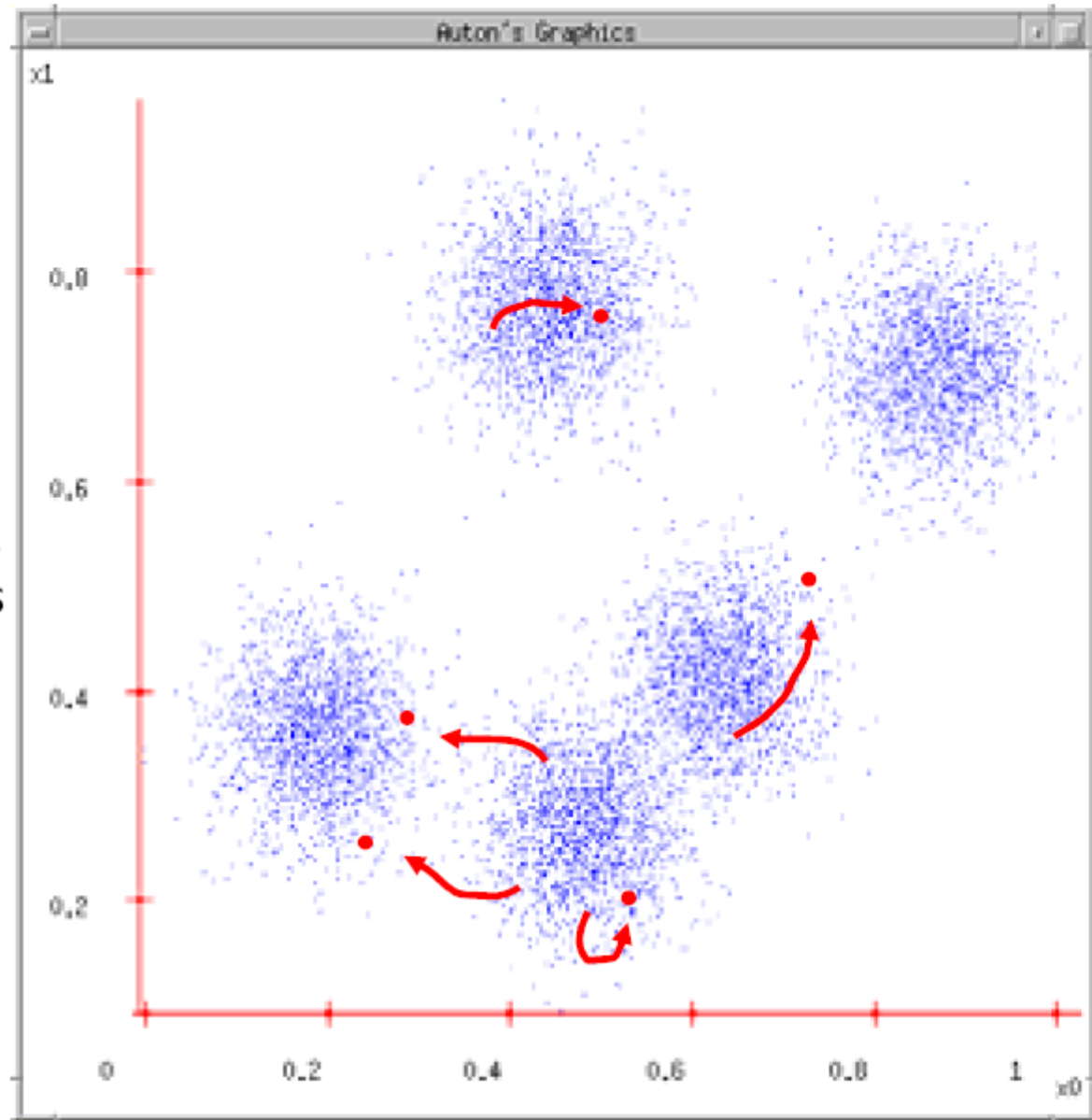
K-means

1. Ask user how many clusters they'd like.
(e.g. $k=5$)
2. Randomly guess k cluster Center locations
3. Each datapoint finds out which Center it's closest to.
4. Each Center finds the centroid of the points it owns



K-means

1. Ask user how many clusters they'd like.
(e.g. $k=5$)
2. Randomly guess k cluster Center locations
3. Each datapoint finds out which Center it's closest to.
4. Each Center finds the centroid of the points it owns...
5. ...and jumps there
6. ...Repeat until terminated!



K-means clustering

- Java demo:

<http://kovan.ceng.metu.edu.tr/~maya/kmeans/index.html>

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

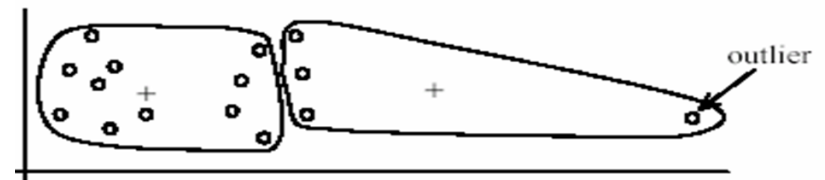
K-means: pros and cons

Pros

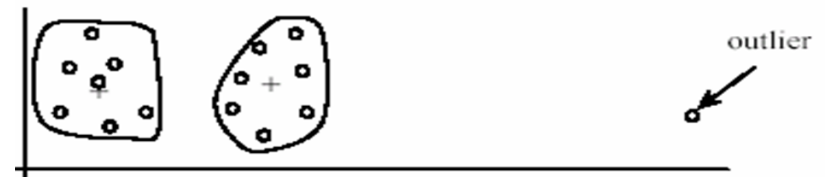
- Simple, fast to compute
- Converges to local minimum of within-cluster squared error

Cons/issues

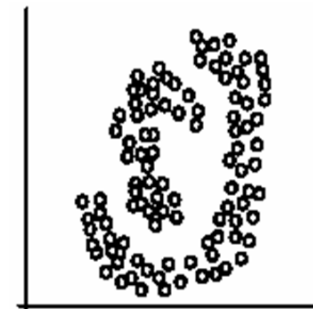
- Setting k ?
- Sensitive to initial centers
- Sensitive to outliers
- Detects spherical clusters
- Assuming means can be computed



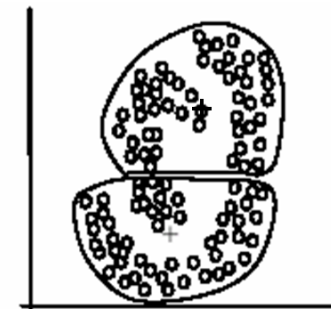
(A): Undesirable clusters



(B): Ideal clusters



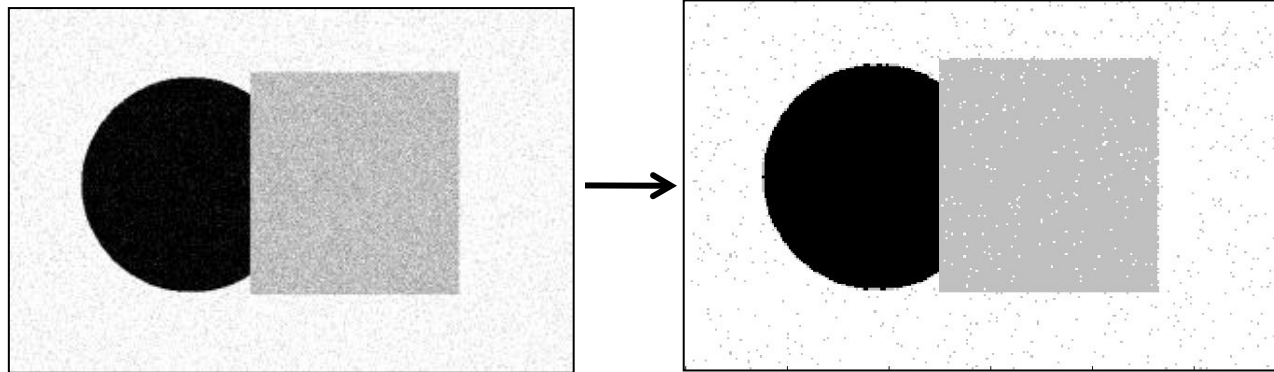
(A): Two natural clusters



(B): k -means clusters

An aside: Smoothing out cluster assignments

- Assigning a cluster label per pixel may yield outliers:

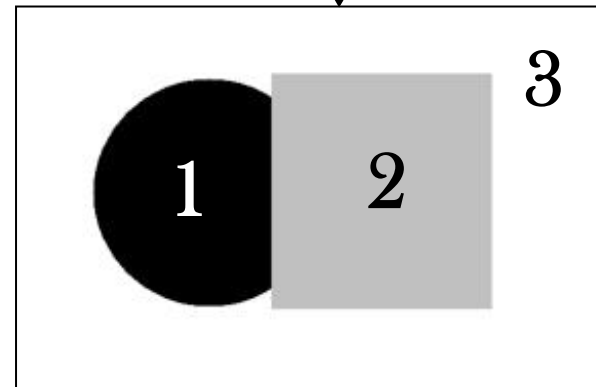


original

labeled by cluster
center's intensity



- How to ensure they are spatially smooth?



Segmentation as clustering

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

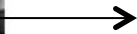
Grouping pixels based on intensity similarity



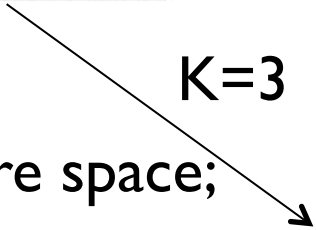
Feature space: intensity value (1-d)



K=2



K=3



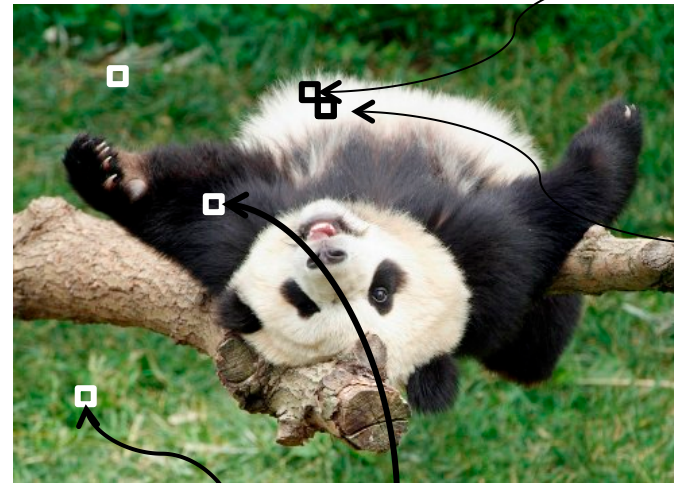
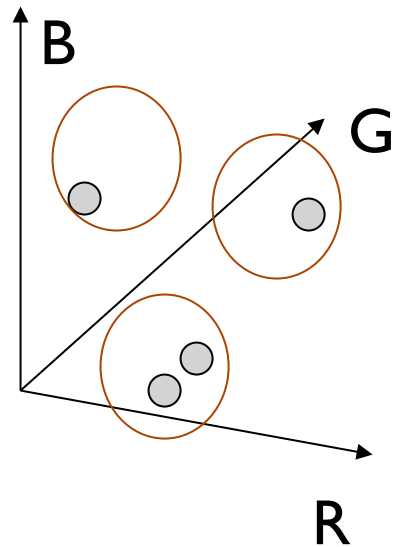
quantization of the feature space;
segmentation label map



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



R=255
G=200
B=250

R=245
G=220
B=248

R=15
G=189
B=2

R=3
G=12
B=2

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.



Segmentation as clustering

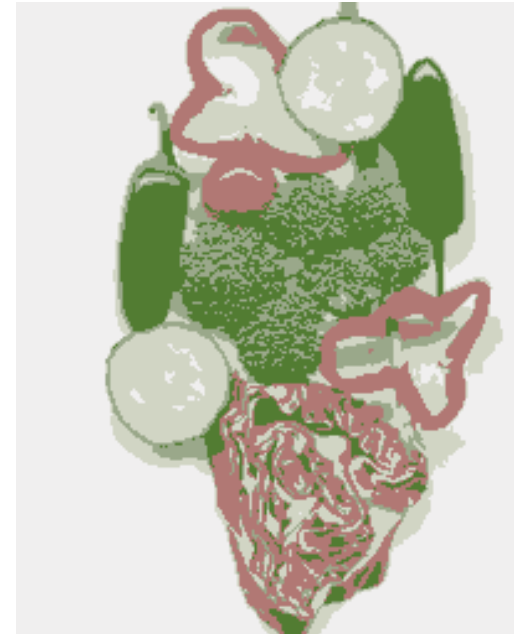
Image



Clusters on intensity (K=5)



Clusters on color (K=5)



K-means clustering using intensity alone and color alone

Segmentation as clustering

Image



Clusters on color



K-means using color alone, 11 segments

Segmentation as clustering



K-means using color alone,
11 segments.

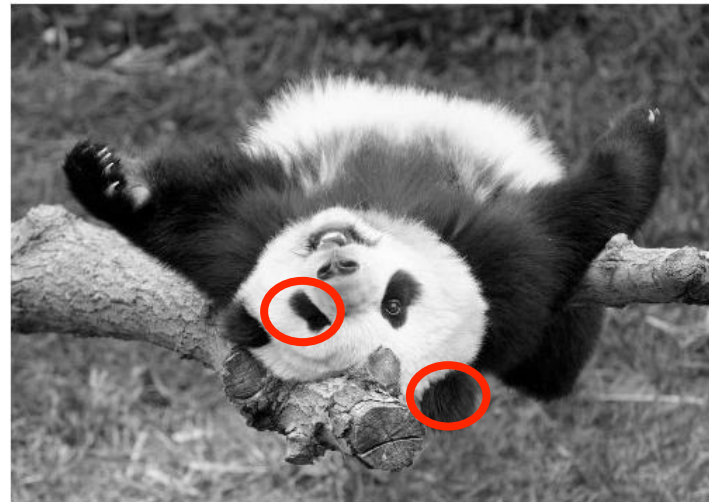
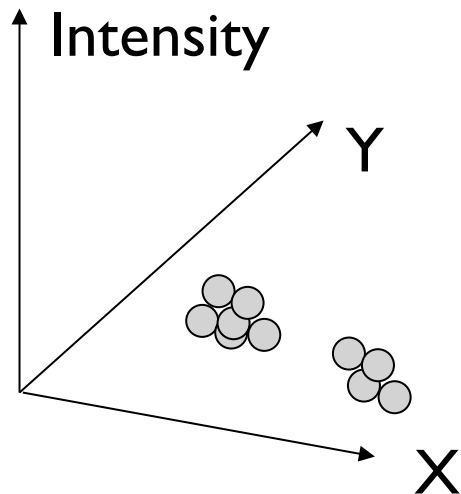
***Color alone
often will not
yield salient segments!***



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

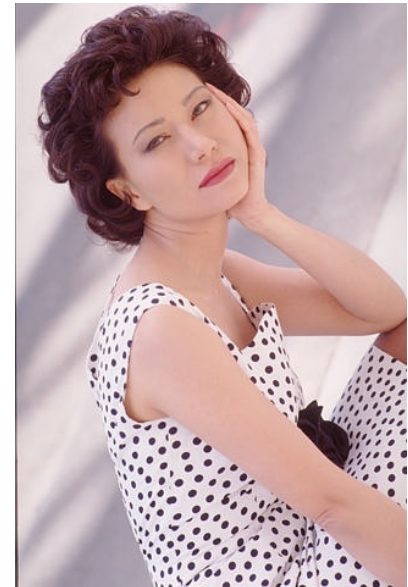


Both regions are black, but if we also include position (x,y) , then we could group the two into distinct segments; way to encode both similarity & proximity.

Slide credit: K Grauman

Segmentation as clustering

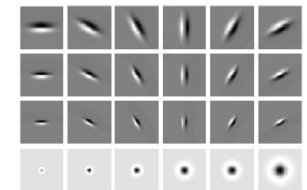
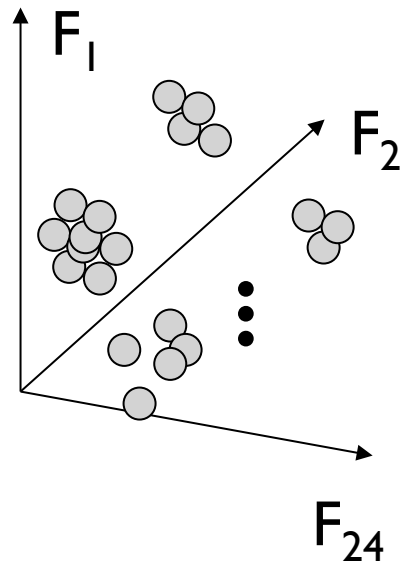
- Color, brightness, position alone are not enough to distinguish all regions...



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

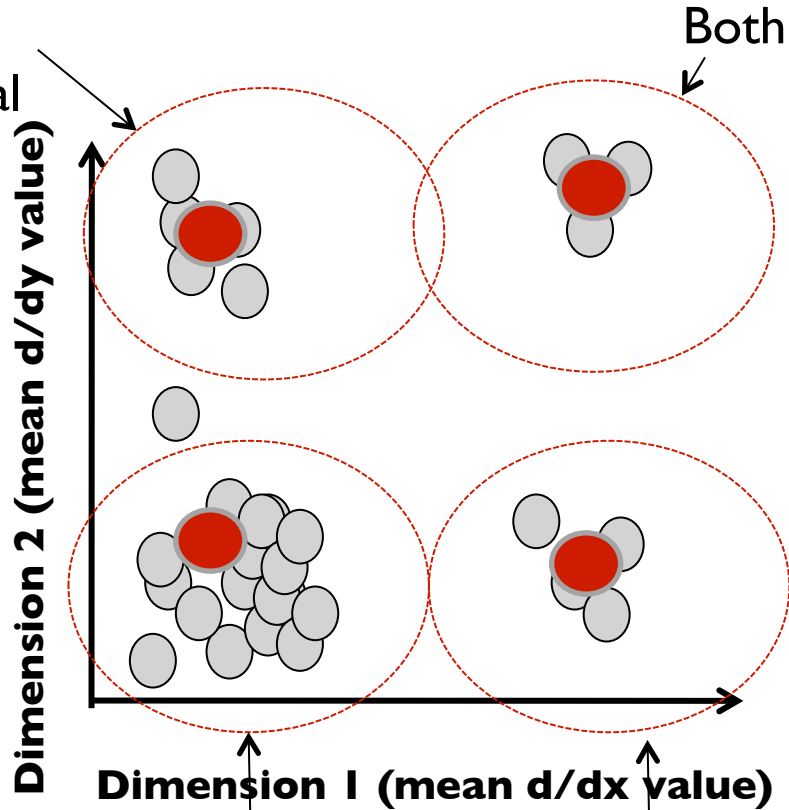


**Filter bank
of 24 filters**

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

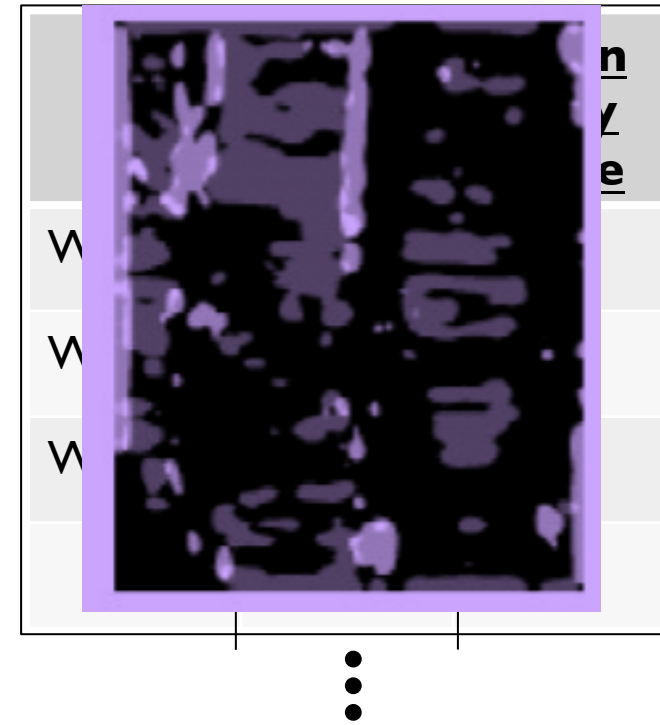
Texture representation example

Windows with primarily horizontal edges



Windows with small gradient in both directions

Windows with primarily vertical edges



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*

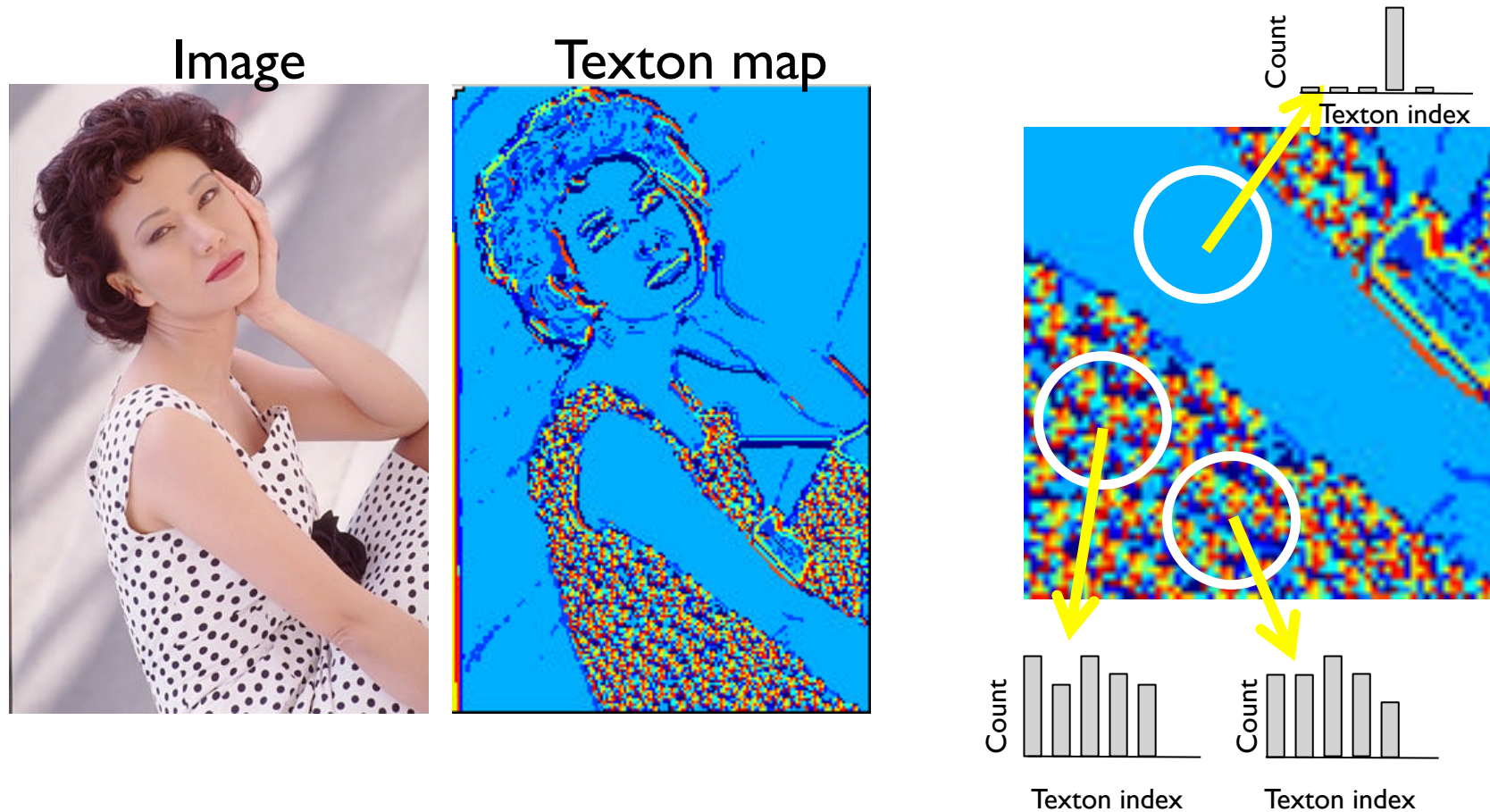
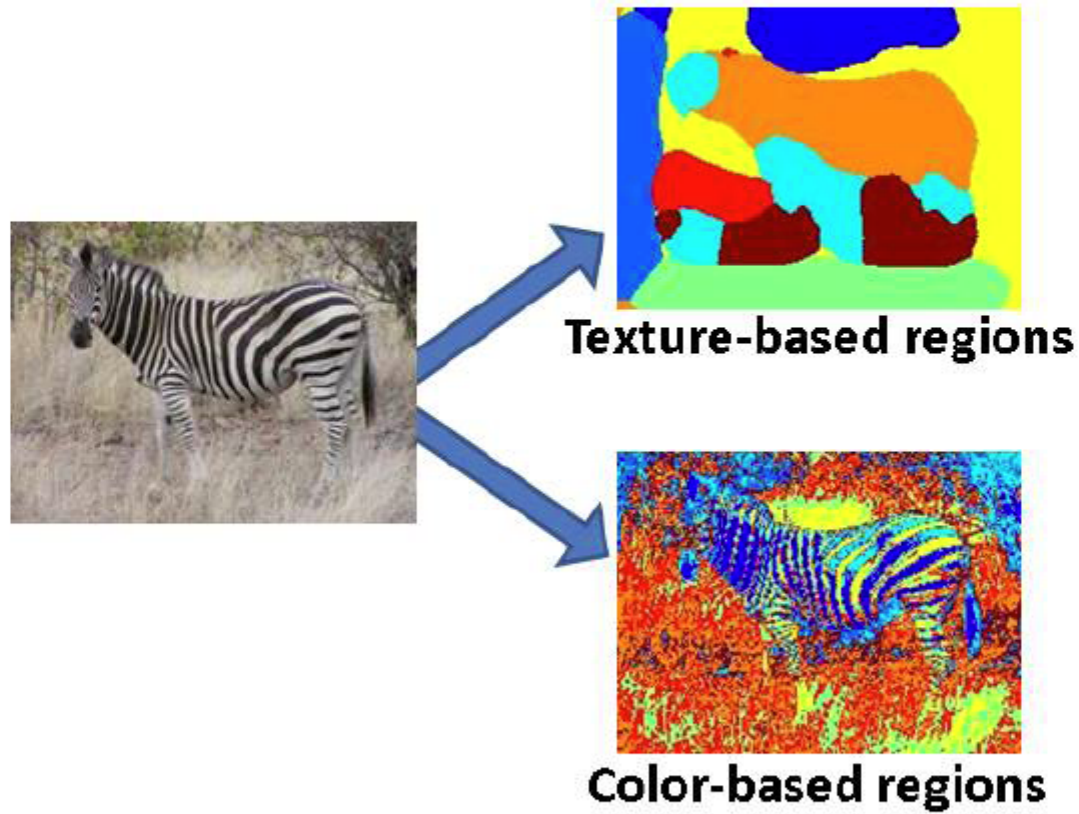


Image segmentation example



Pixel properties vs. neighborhood properties

query



query

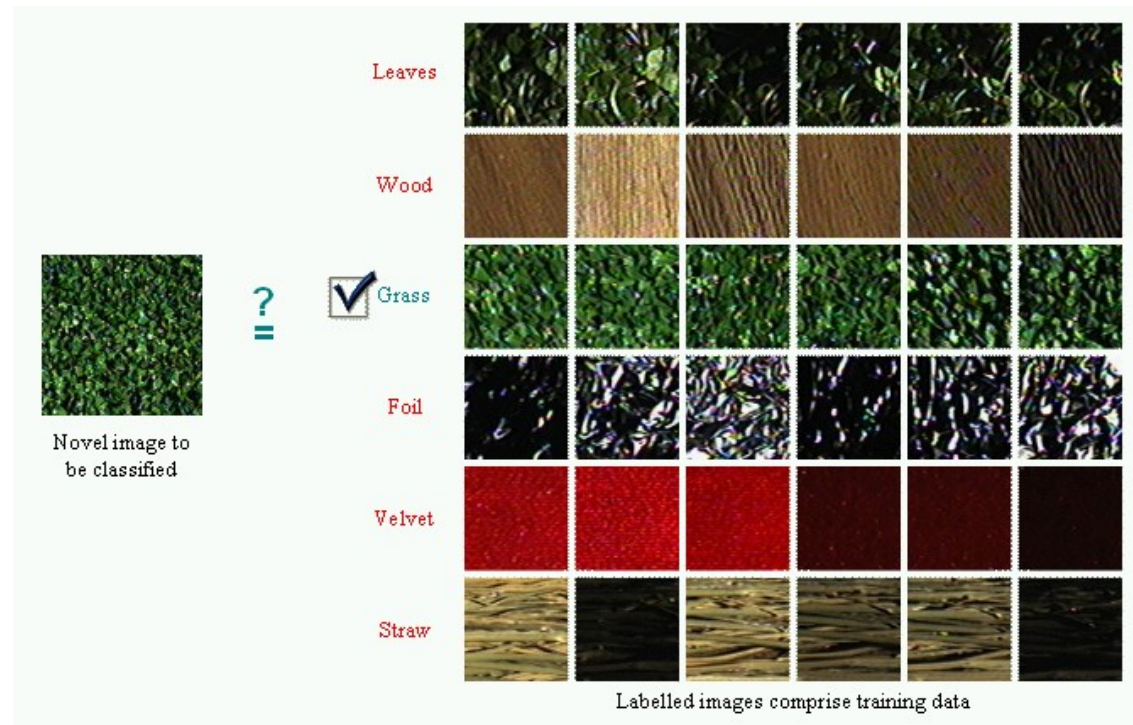


These look very similar in terms of their color distributions (histograms).

How would their *texture* distributions compare?

Material classification example

For an image of a single texture, we can classify it according to its global (image-wide) texton histogram.



Material classification example

Nearest neighbor classification: label the input according to the nearest known example's label.



Novel Image



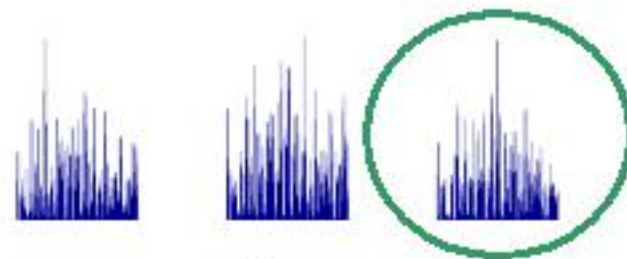
Model

$$\chi^2 =$$

$$\chi^2(h_i, h_j) = \frac{1}{2} \sum_{k=1}^K \frac{[h_i(k) - h_j(k)]^2}{h_i(k) + h_j(k)}$$

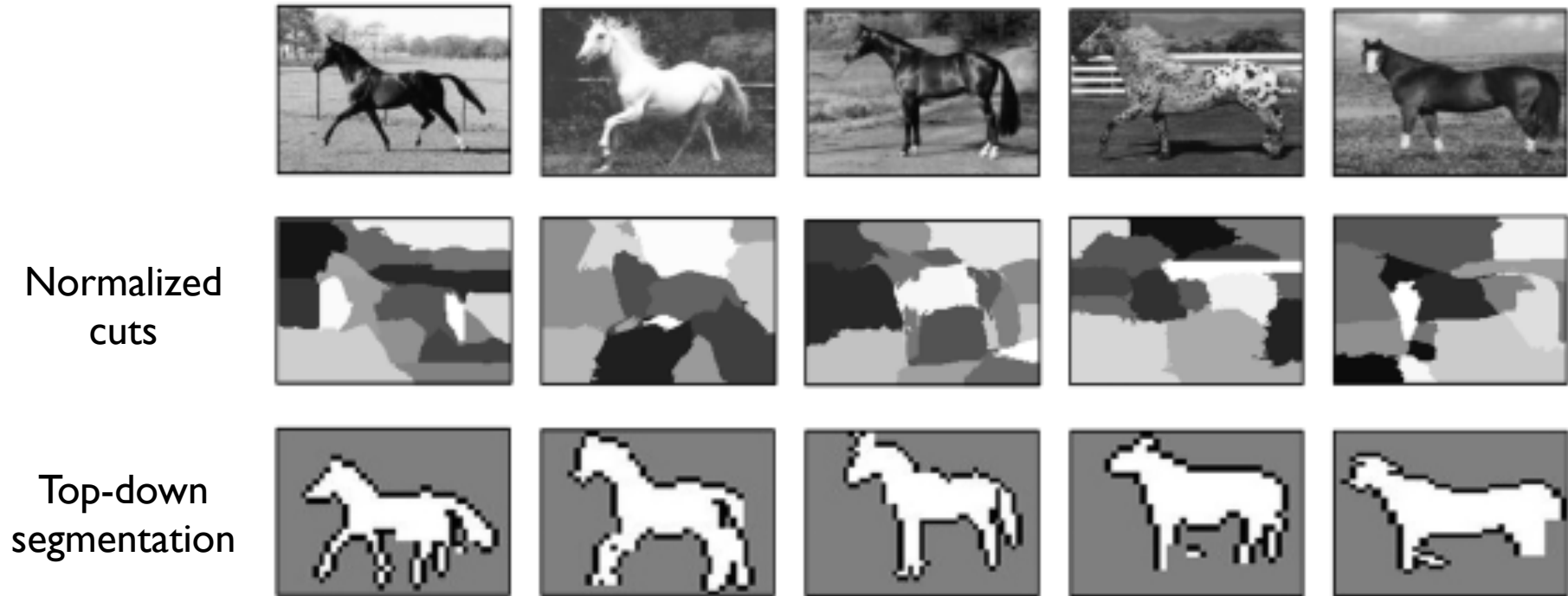


Plastic



Grass

Reading Assignment #5



- E. Borenstein and S. Ullman, [Class-specific, top-down segmentation](#), ECCV 2002
- Due on 3rd of January

Segmentation methods

- Segment foreground from background
- Histogram-based segmentation
- Segmentation as clustering
 - K-means clustering
 - Mean-shift segmentation
- Graph-theoretic segmentation
 - Min cut
 - Normalized cuts
- Interactive segmentation

Next week