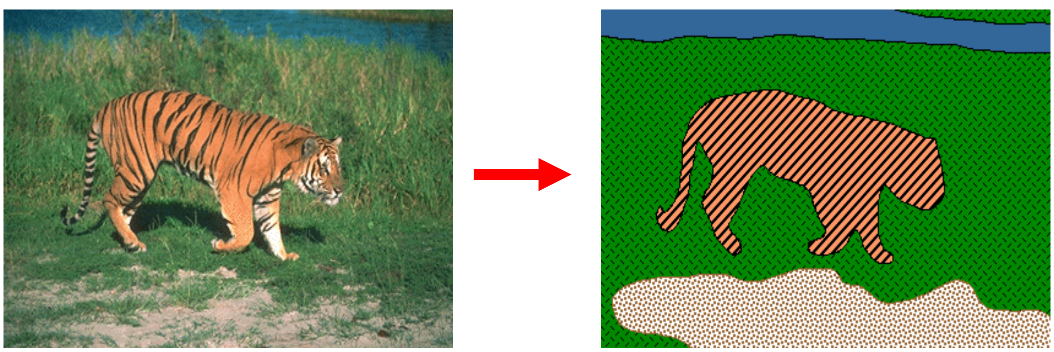
Lecture 7: Segmentation

Thursday, Sept 20

Outline

- Why segmentation?
- Gestalt properties, fun illusions and/or revealing examples
- Clustering
  - Hierarchical
  - K-means
  - Mean Shift
  - Graph-theoretic
    - Normalized cuts



Grouping

- Segmentation / Grouping / Perceptual organization: gather features that belong together
- Need an intermediate representation, compact description of key image (video, motion,…) parts
- Top down vs. bottom up
- Hard to measure success
- (Fitting: associate a model with observed features)

Examples of grouping in vision

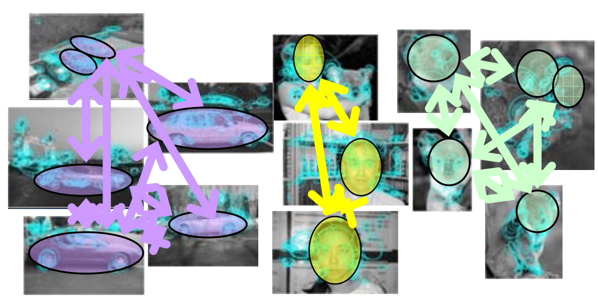
[Figure by J. Shi]

Determine image regions

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

Find shot boundaries

[Figure by Wang & Suter]

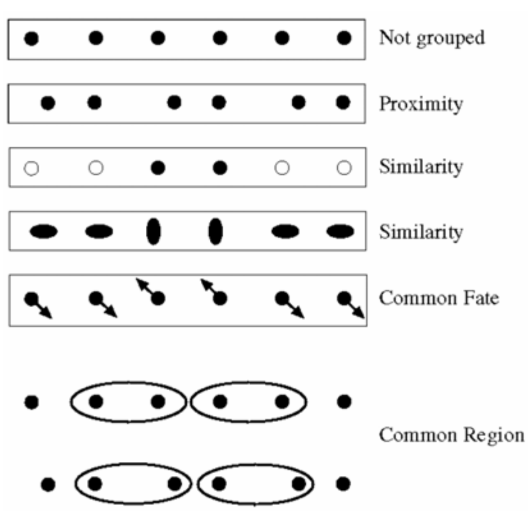


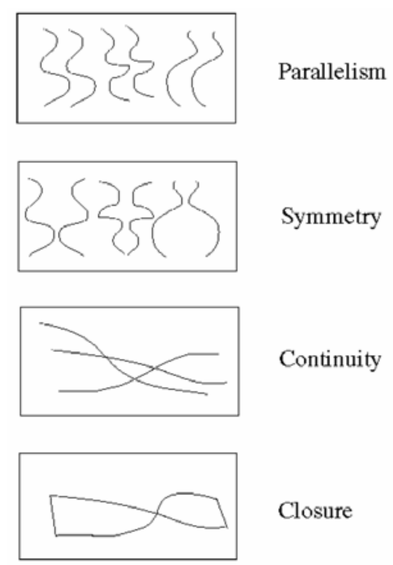
Object-level grouping

Gestalt

- Gestalt: whole or group
- Whole is greater than sum of its parts
- Psychologists identified series of factors that predispose set of elements to be grouped
- Interesting observations/explanations, but not necessarily useful for algorithm building

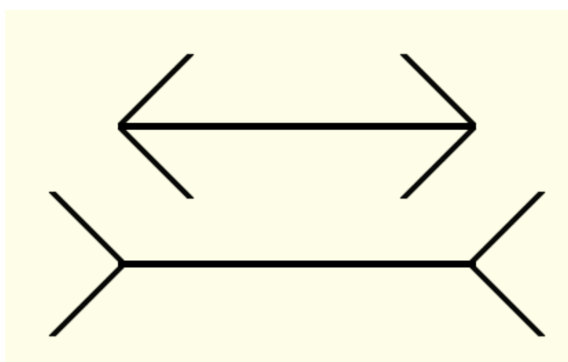
Some Gestalt factors



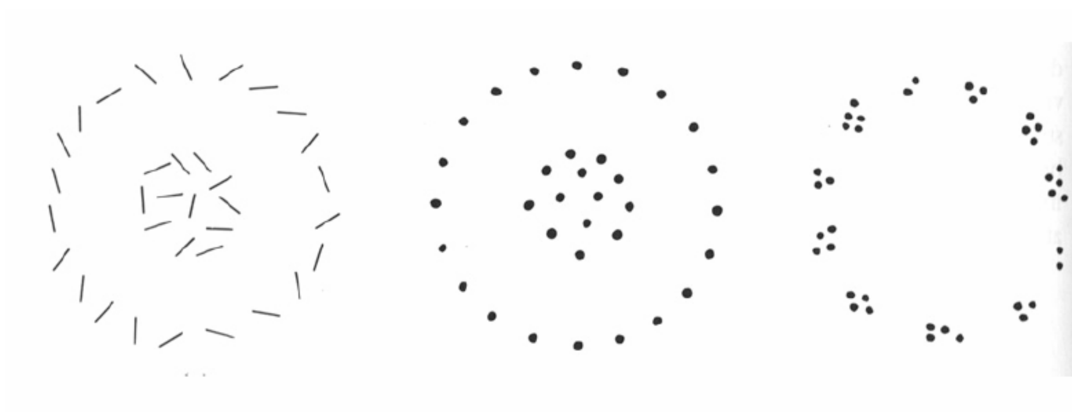


Muller-Lyer illusion

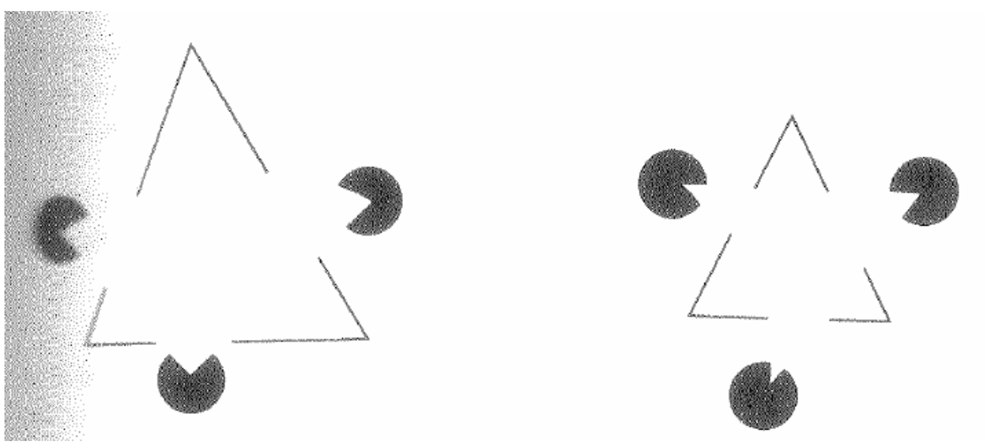
- http://www.michaelbach.de/ot/sze_muelue/index.html



Gestalt principle: grouping key to visual perception.

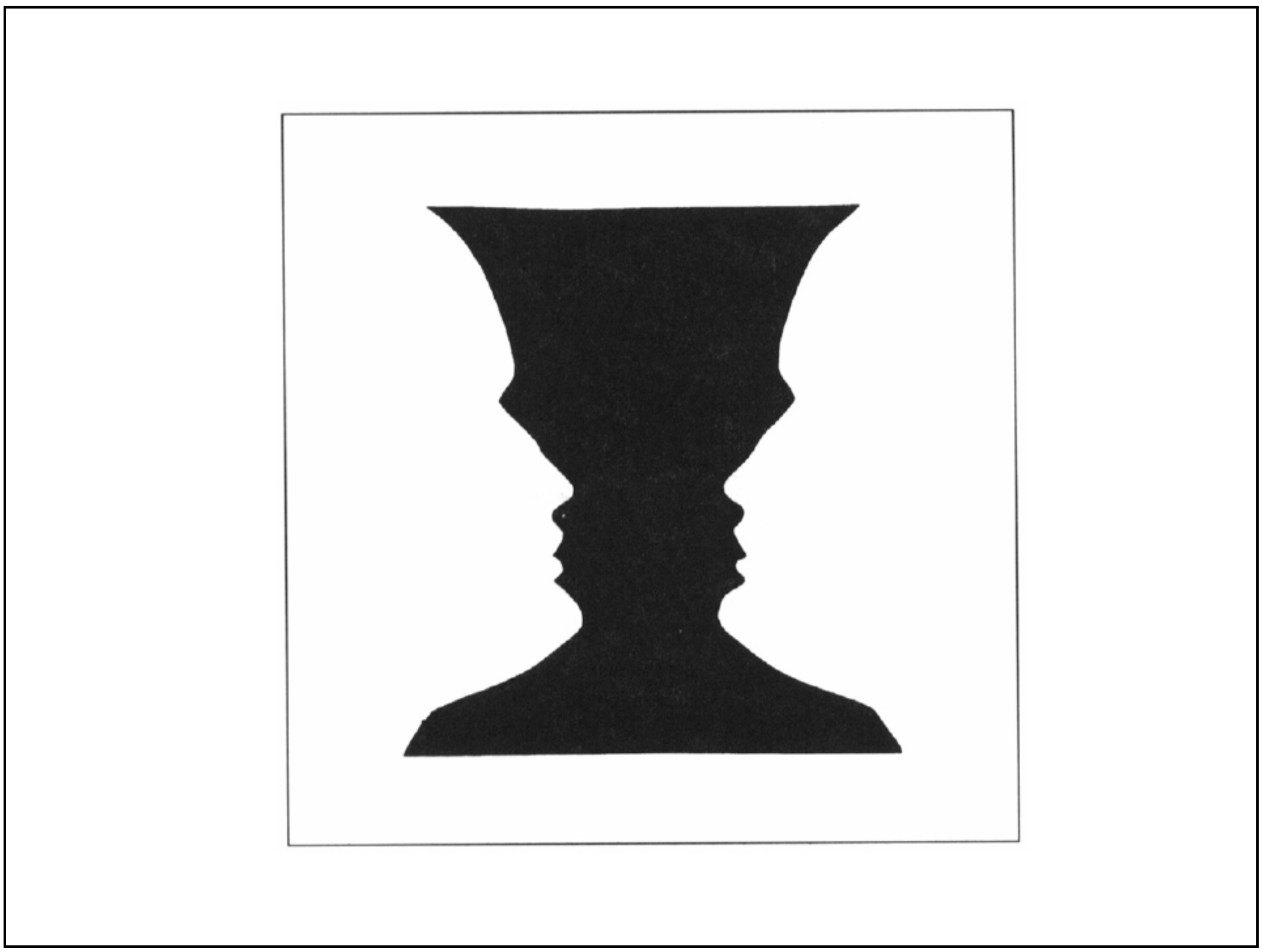


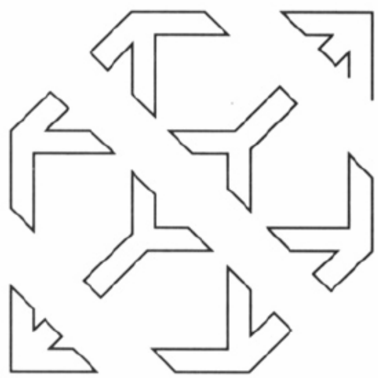
Subjective contours



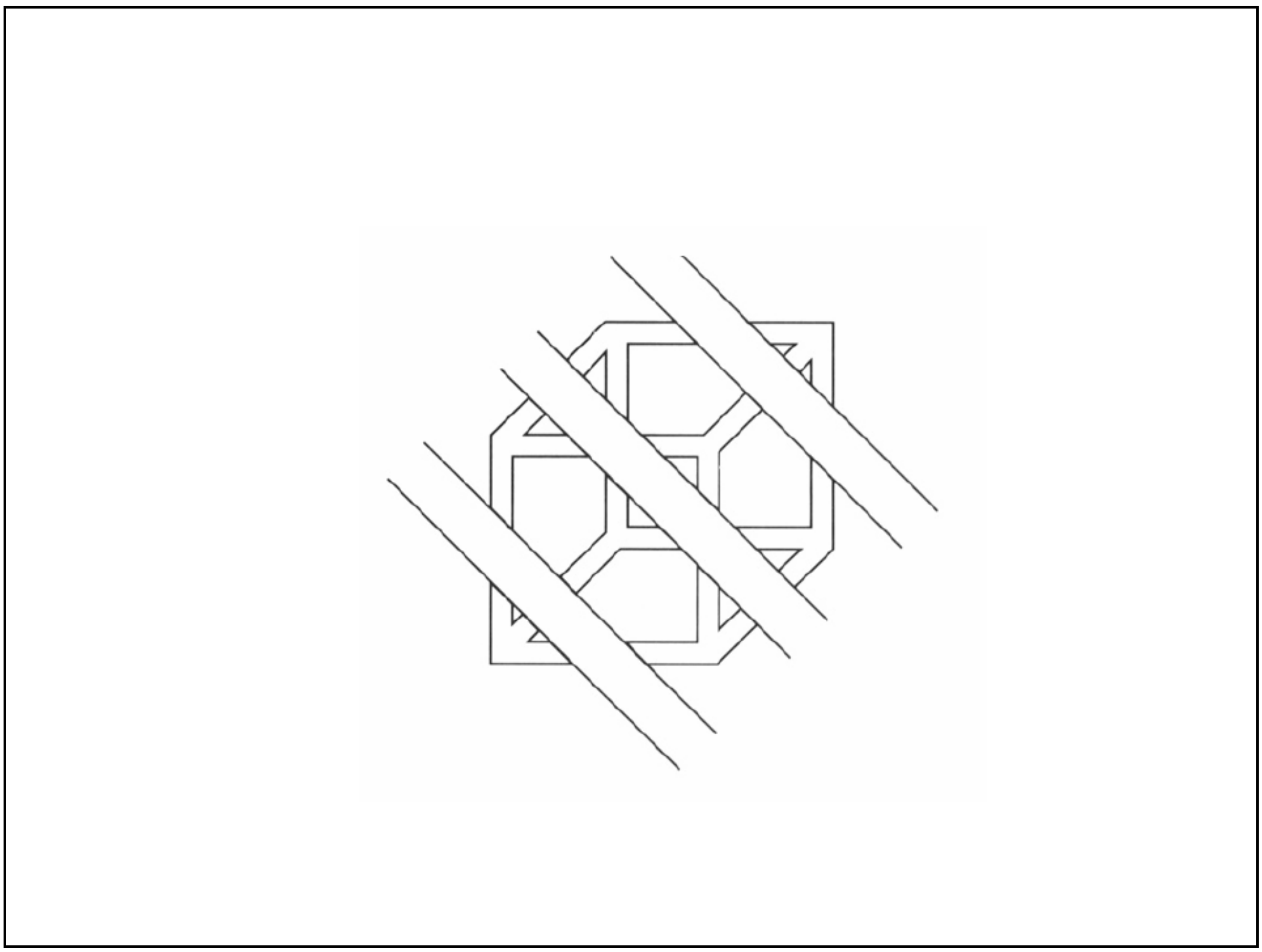
Interesting tendency to explain by occlusion

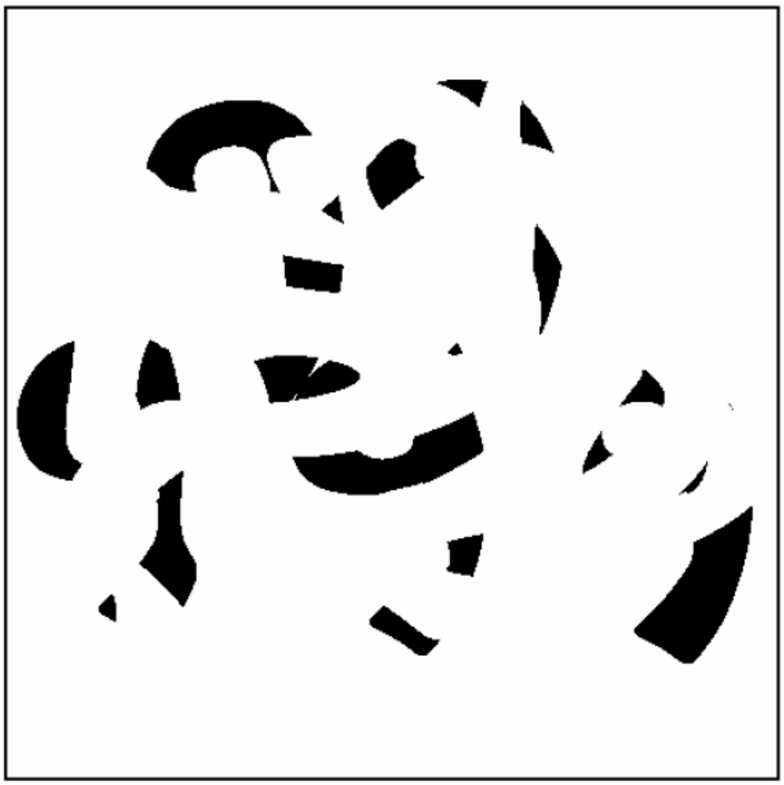
In *Vision*, D. Marr, 1982

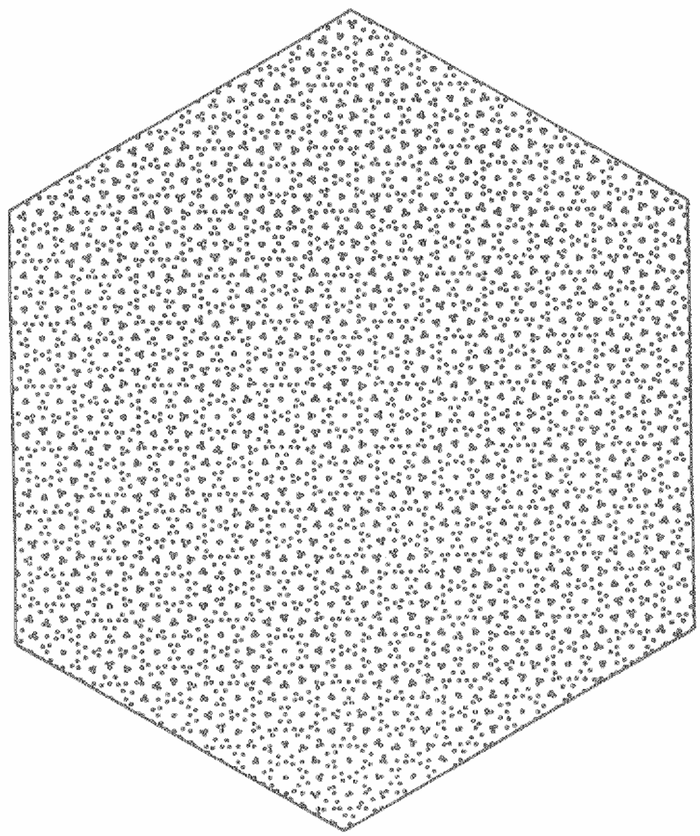




D. Forsyth





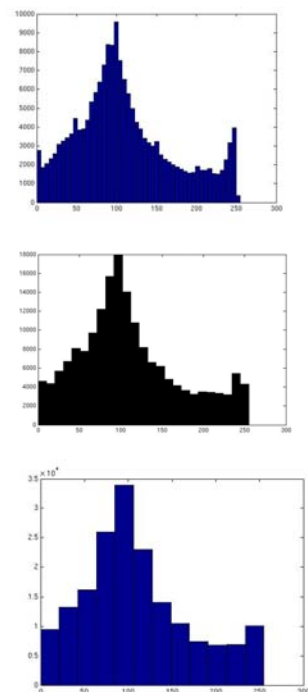


In *Vision*, D. Marr, 1982; from J. L. Marroquin, “Human visual perception of structure”, 1976.

Outline

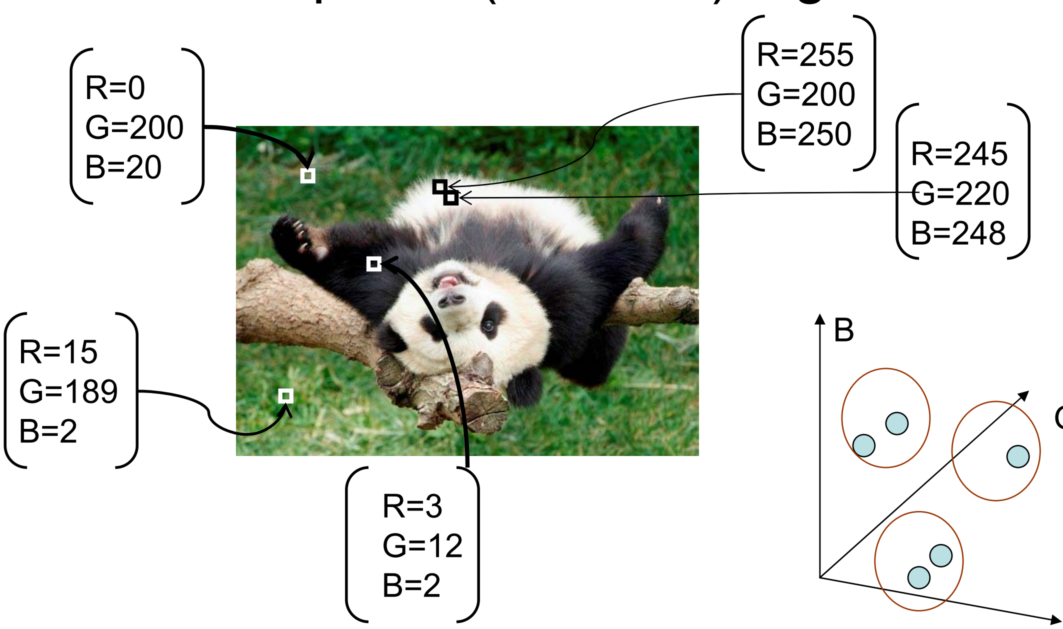
- Why segmentation?
- Gestalt properties, fun illusions and/or revealing examples
- Clustering
  – Hierarchical
  – K-means
  – Mean Shift
  – Graph-theoretic
    - Normalized cuts

Histograms vs. clustering



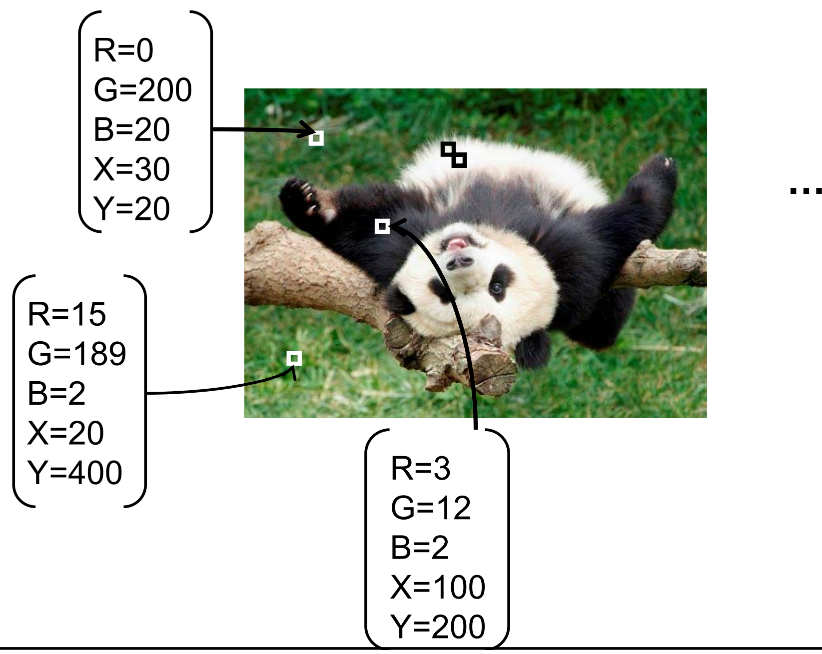
Segmentation as clustering

- Cluster similar pixels (features) together



Segmentation as clustering

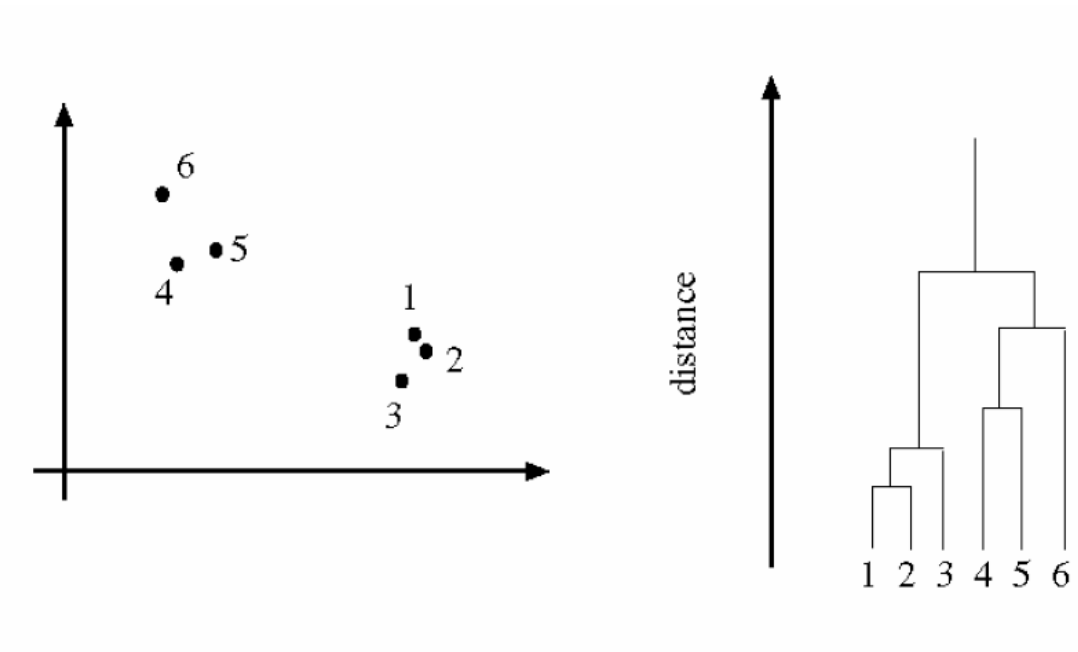
- Cluster similar pixels (features) together



Hierarchical clustering

- **Agglomerative**: Each point is a cluster, Repeatedly merge two nearest clusters
- **Divisive**: Start with single cluster, Repeatedly split into most distant clusters

Dendrogram



Inter-cluster distances

- Single link: min distance between any elements

$$D(C_i, C_j) = \min\{d(x, y) \mid x \in C_i, y \in C_j\}$$

- Complete link: max distance between any elements

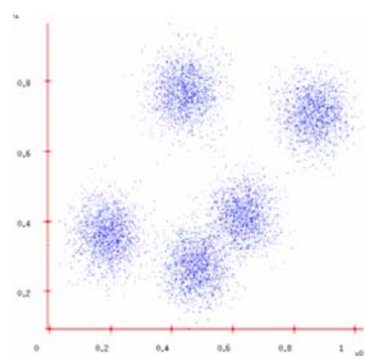
$$D(C_i, C_j) = \max\{d(x, y) \mid x \in C_i, y \in C_j\}$$

- Average link

$$D(C_i, C_j) = \text{avg}\{d(x, y) \mid x \in C_i, y \in C_j\}$$

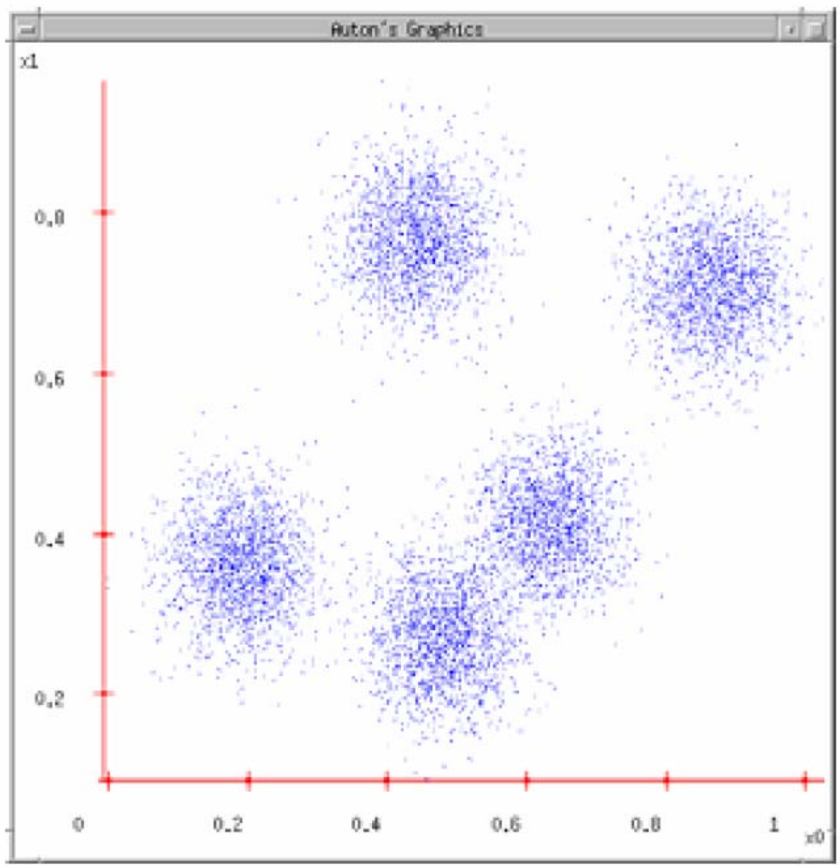
K-means

- Given *k*, want to minimize error among *k* clusters
- Error defined as distance of cluster points to its center
- Search space too large
- *k*-means: iterative algorithm :
  – Fix cluster centers, allocate
  – Fix allocation, compute best centers



K-means

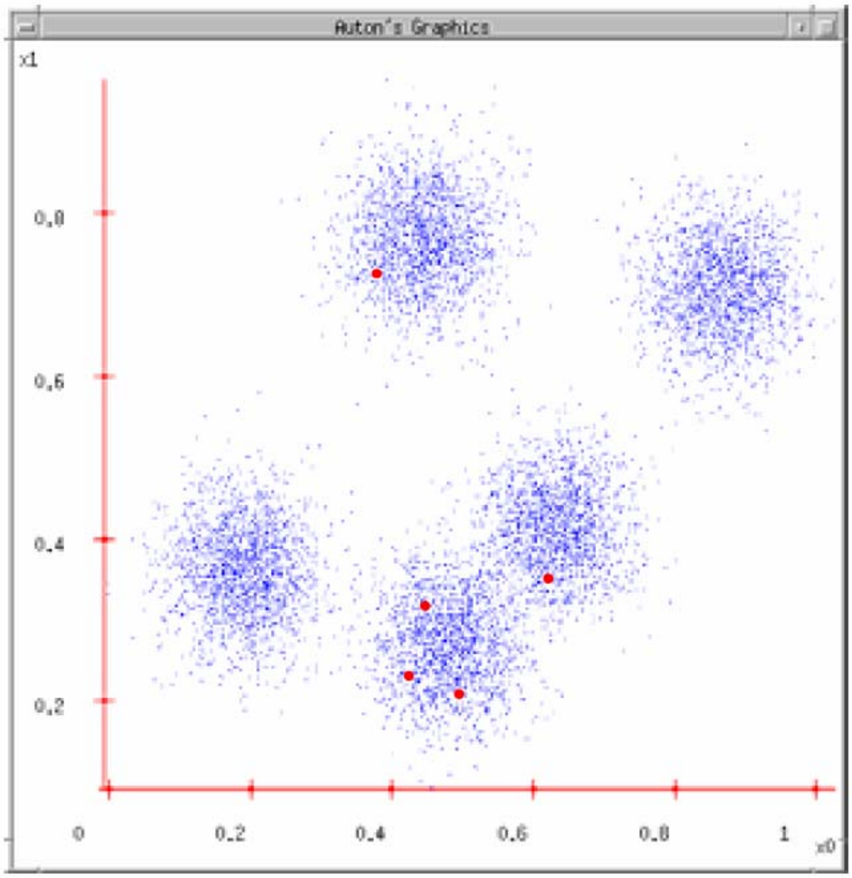
1. Ask user how many clusters they'd like. *(e.g. k=5)*



K-means slides by
Andrew Moore

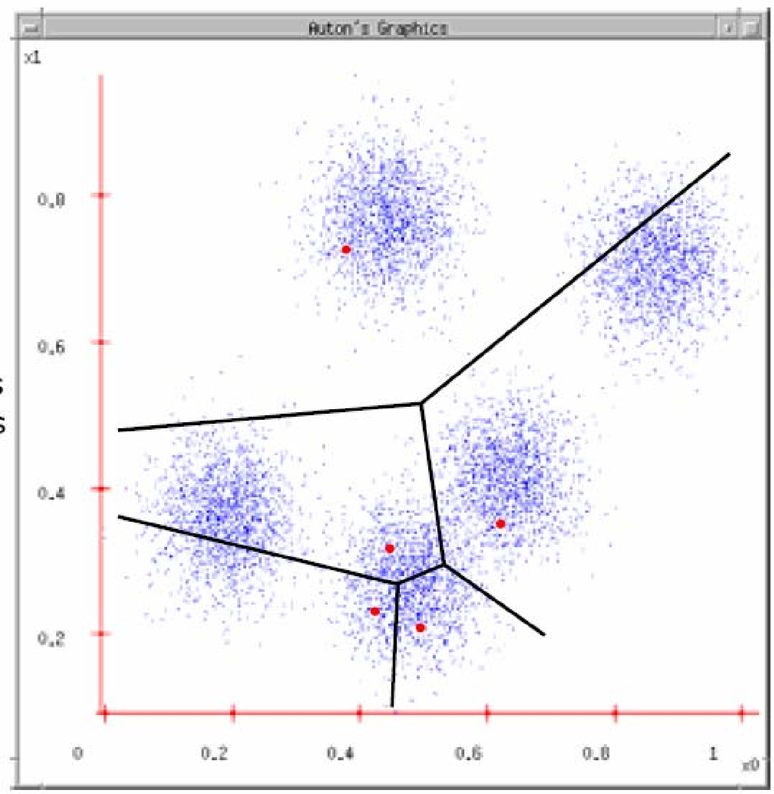
K-means

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



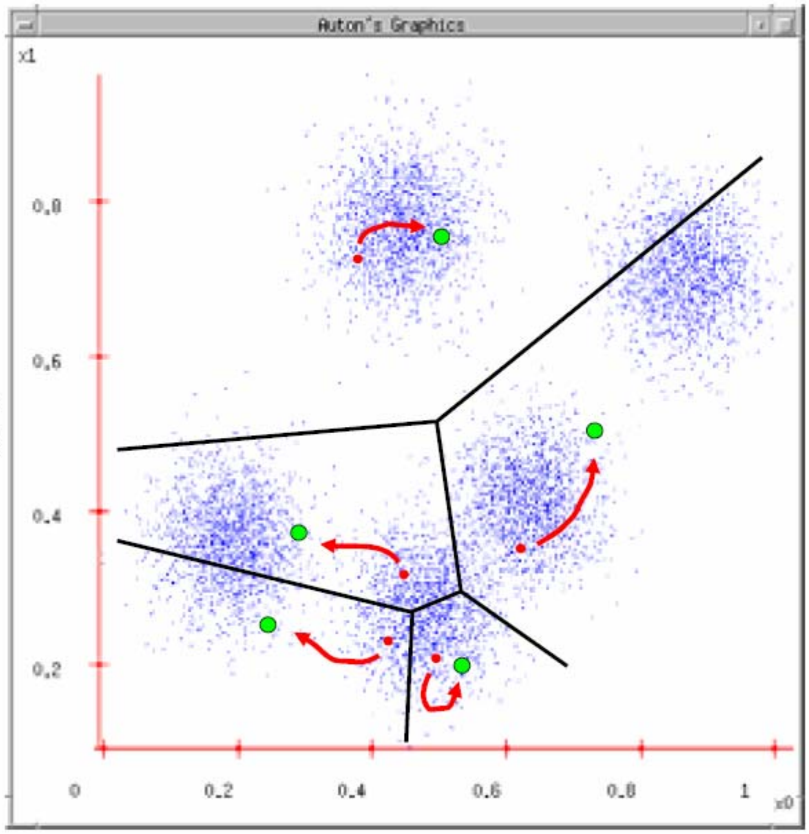
K-means

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)



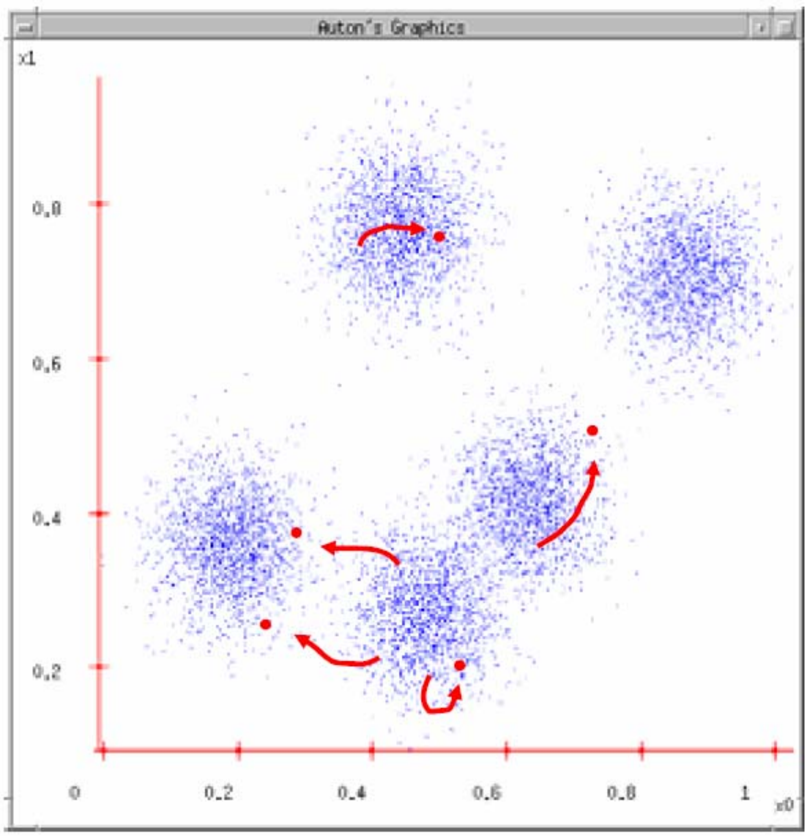
K-means

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

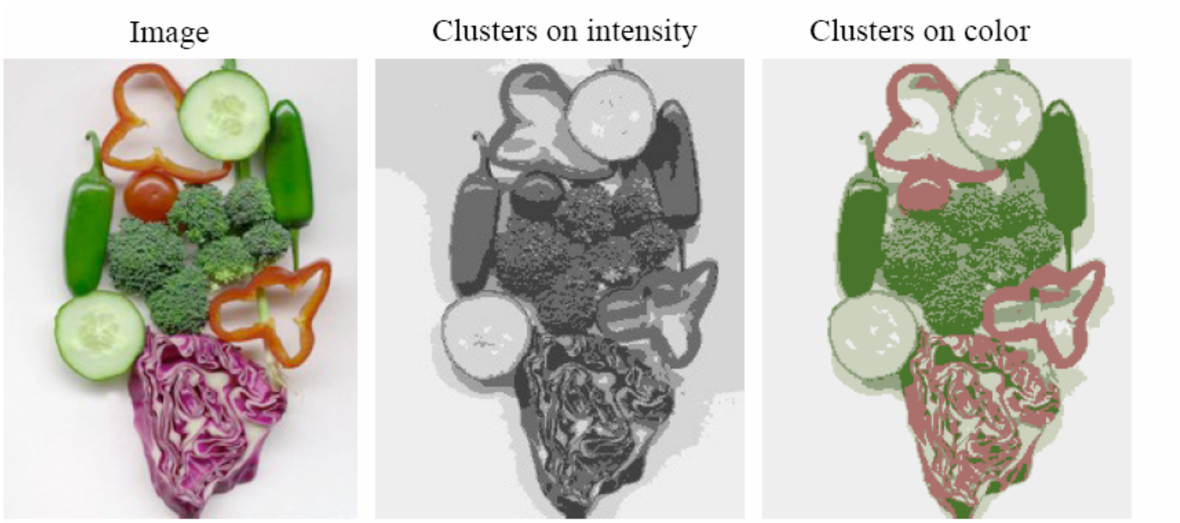


K-means

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 for color-based segementation



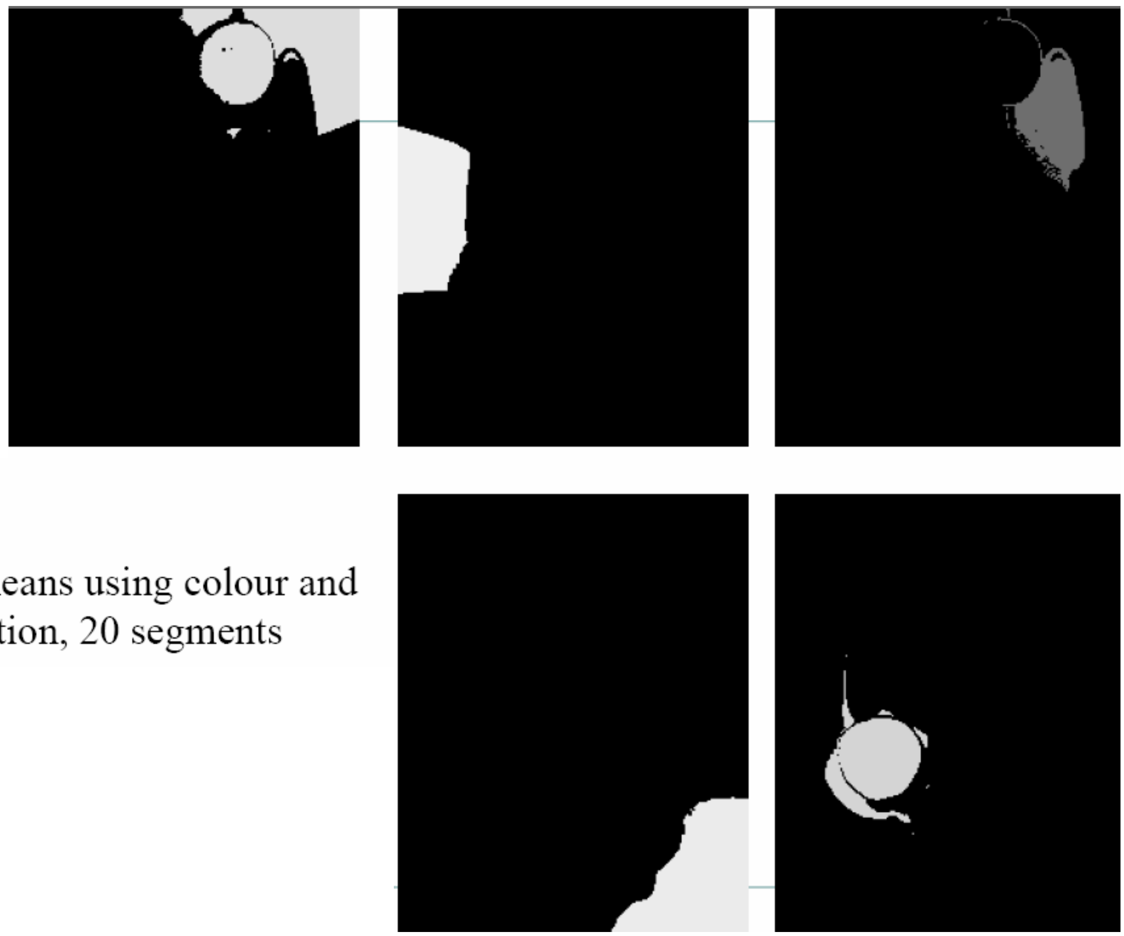
Image

Clusters on color

K-means using color alone, 11 segments

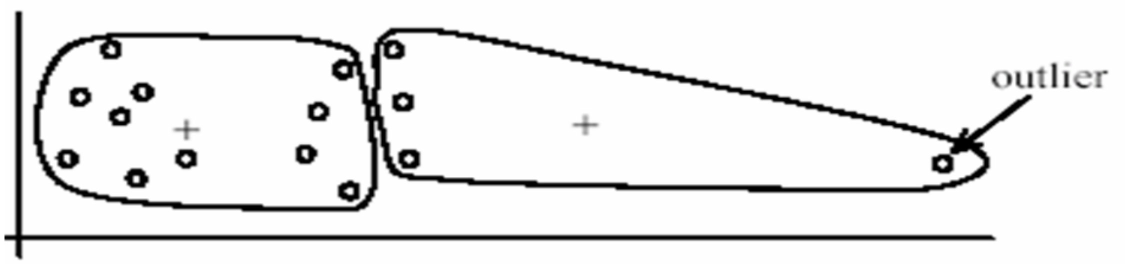
Adapted from David Forsyth, UC Berkeley

K-means using color alone, 11 segments.



K-means using colour and position, 20 segments

K-means and outliers

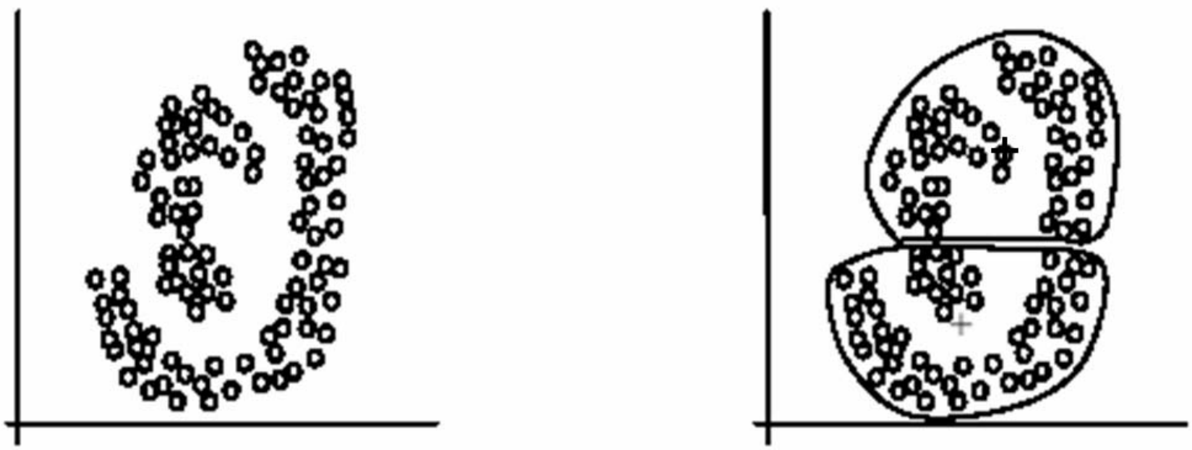


(A): Undesirable clusters

(B): Ideal clusters

K-means

- Use of centroid + spread – doesn't describe irregularly shaped clusters



(A): Two natural clusters (B): *k*-means clusters

K-means

- Pros
  - Simple
  - Converges to local minimum of within-cluster squared error
  - Fast to compute

- Cons/issues
  - Setting k?
  - Sensitive to initial centers (seeds)
  - Usable only if mean is defined
  - Detects spherical clusters
  - Careful combining feature types

Probabilistic clustering

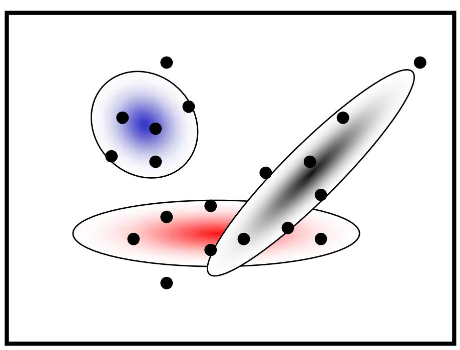
Basic questions

- what's the probability that a point **x** is in cluster m?
- what's the shape of each cluster?

K-means doesn't answer these questions

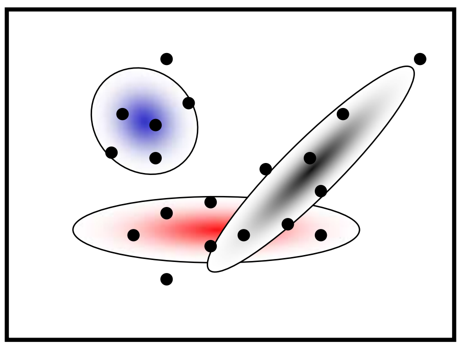
Probabilistic clustering (basic idea)

- Treat each cluster as a Gaussian density function



Slide credit: Steve Seitz

Expectation Maximization (EM)



A probabilistic variant of K-means:

- E step: "soft assignment" of points to clusters
  - estimate probability that a point is in a cluster
- M step: update cluster parameters
  - mean and variance info (covariance matrix)
- maximizes the likelihood of the points given the clusters
- Forsyth Chapter 16 (optional)

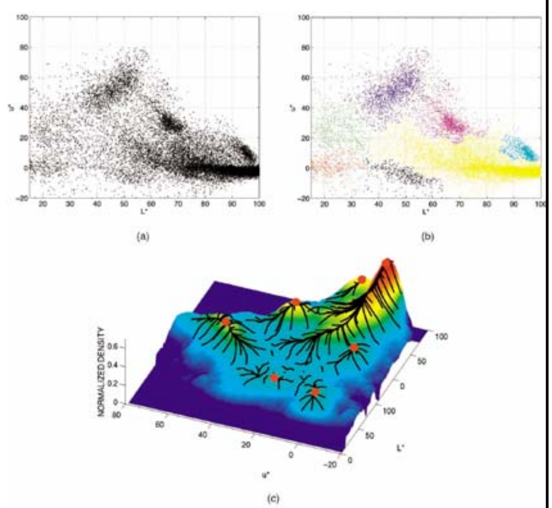
Slide credit: Steve Seitz

Outline

- Why segmentation?
- Gestalt properties, fun illusions and/or revealing examples
- Clustering
  - Hierarchical
  - K-means
  - Mean Shift
  - Graph-theoretic
    - Normalized cuts

Mean shift

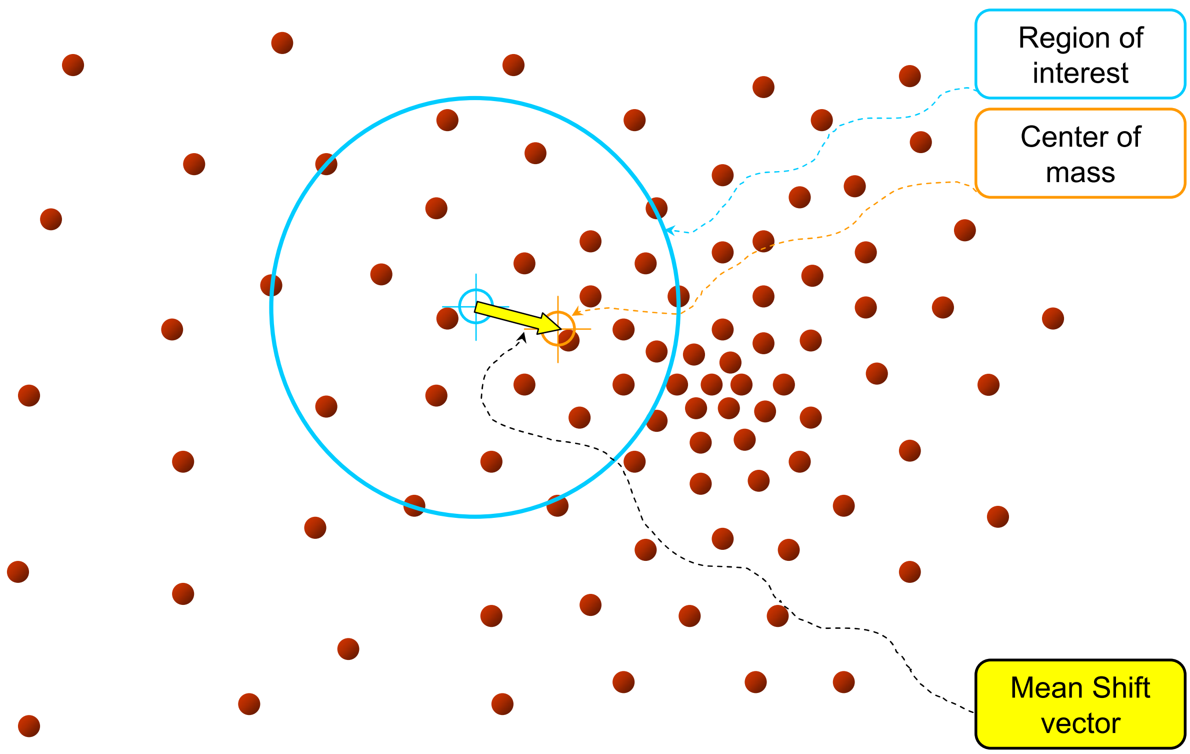
- Seeks the mode among sampled data, or point of highest density
  - Choose search window size
  - Choose initial location of search window
  - Compute mean location (centroid) in window
  - Re-center search window at mean location
  - Repeat until convergence



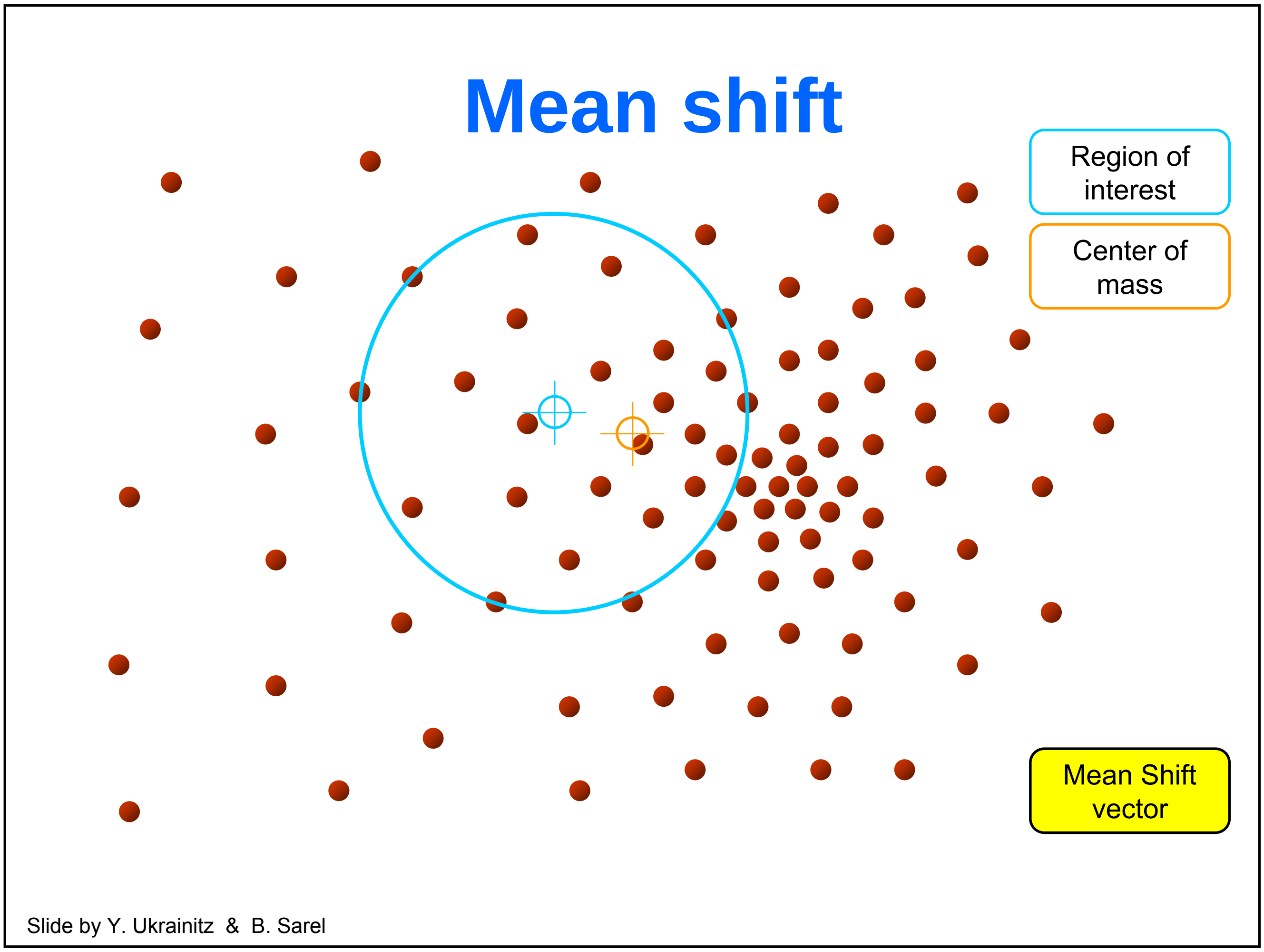
Fukunaga & Hostetler 1975

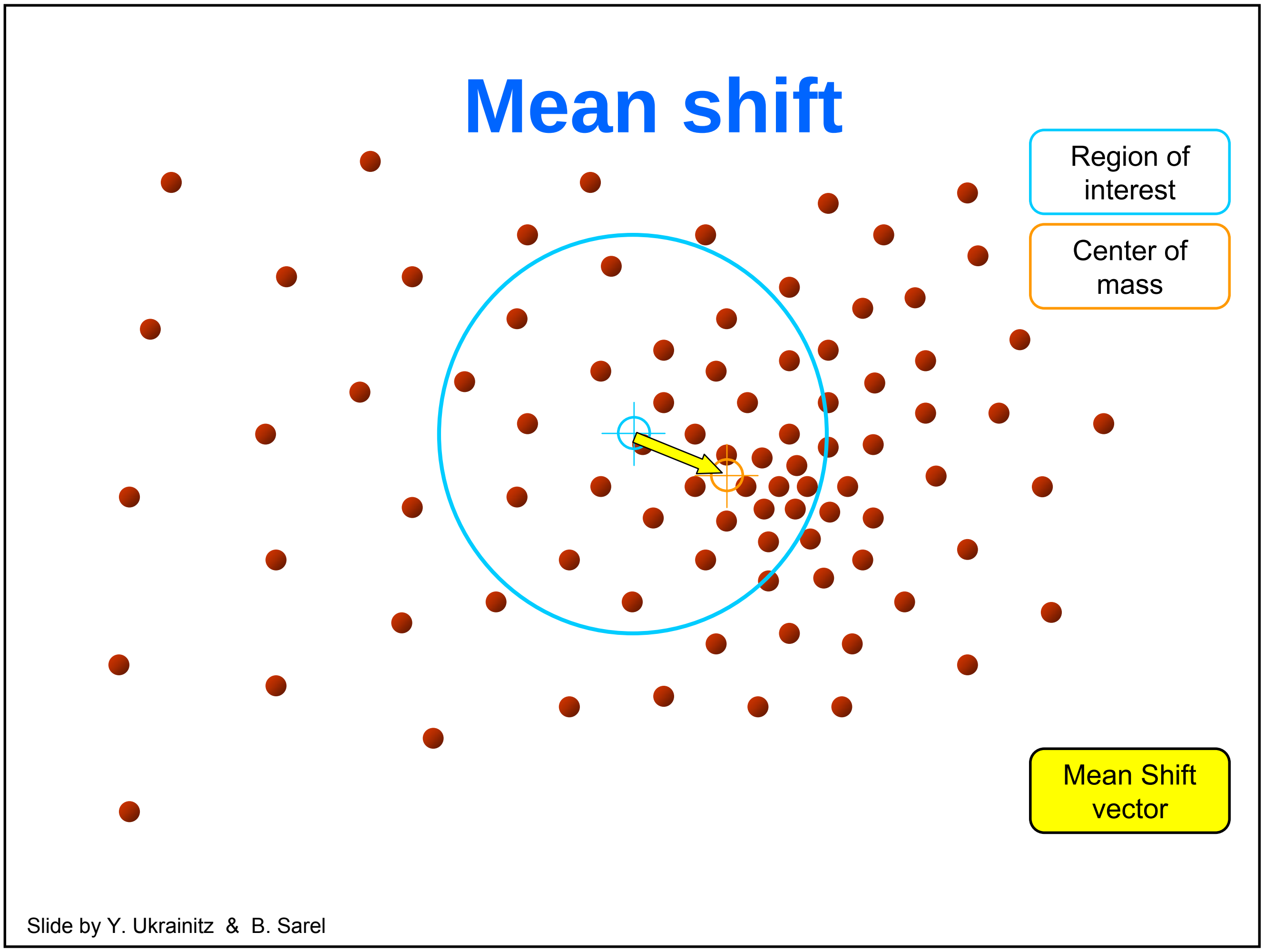
Comaniciu & Meer, PAMI 2002

Mean shift



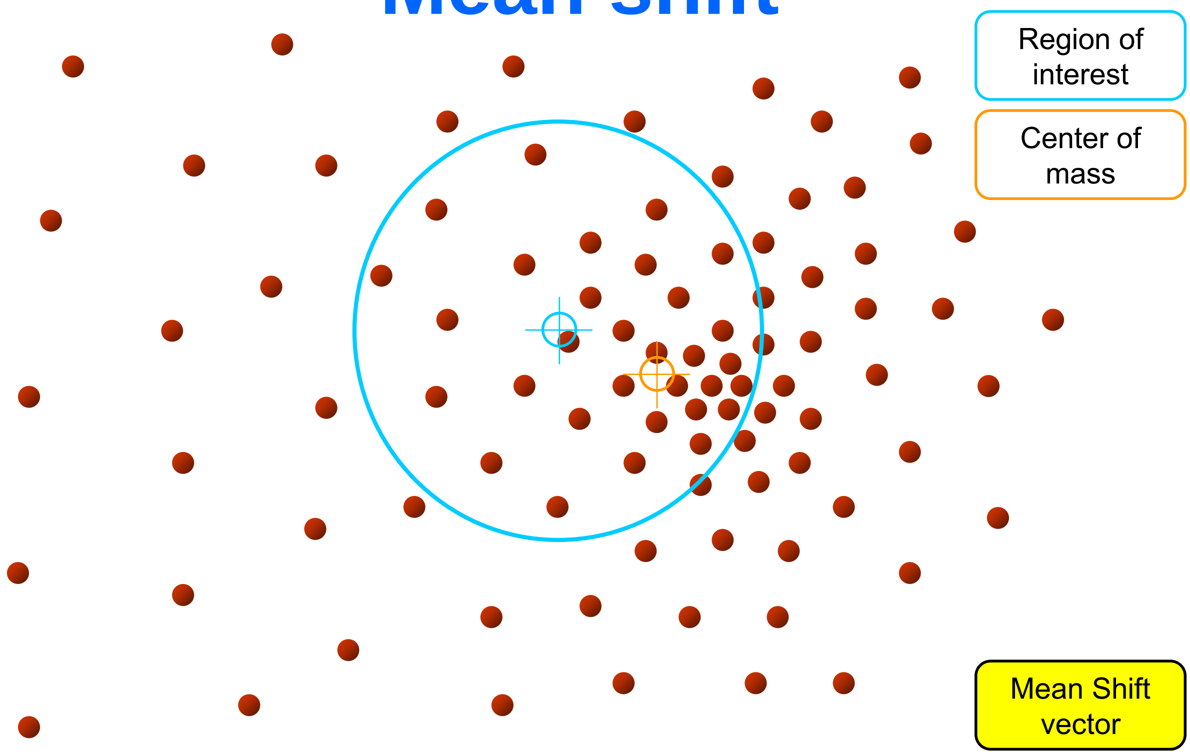
Slide by Y. Ukrainitz & B. Sarel


Mean shift
Region of interest
Center of mass
Mean Shift vector
Slide by Y. Ukrainitz & B. Sarel



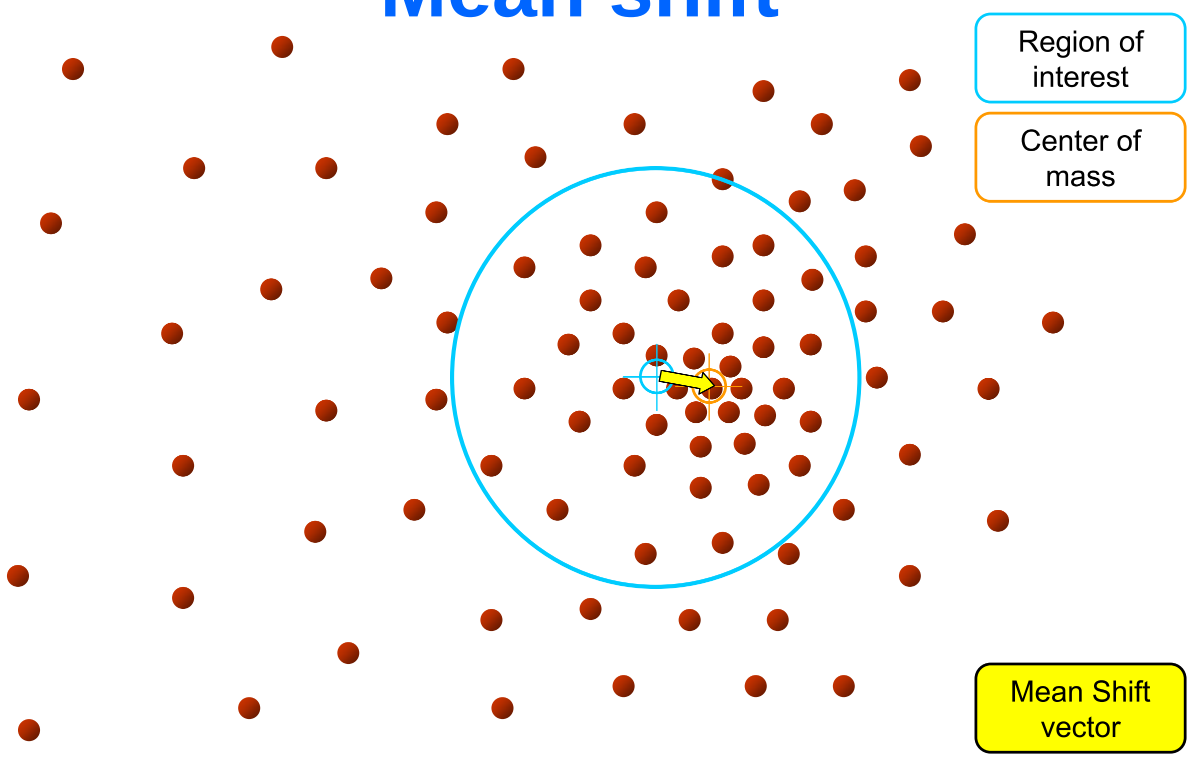
Mean shift
Region of interest
Center of mass
Mean Shift vector
Slide by Y. Ukrainitz & B. Sarel

Mean shift



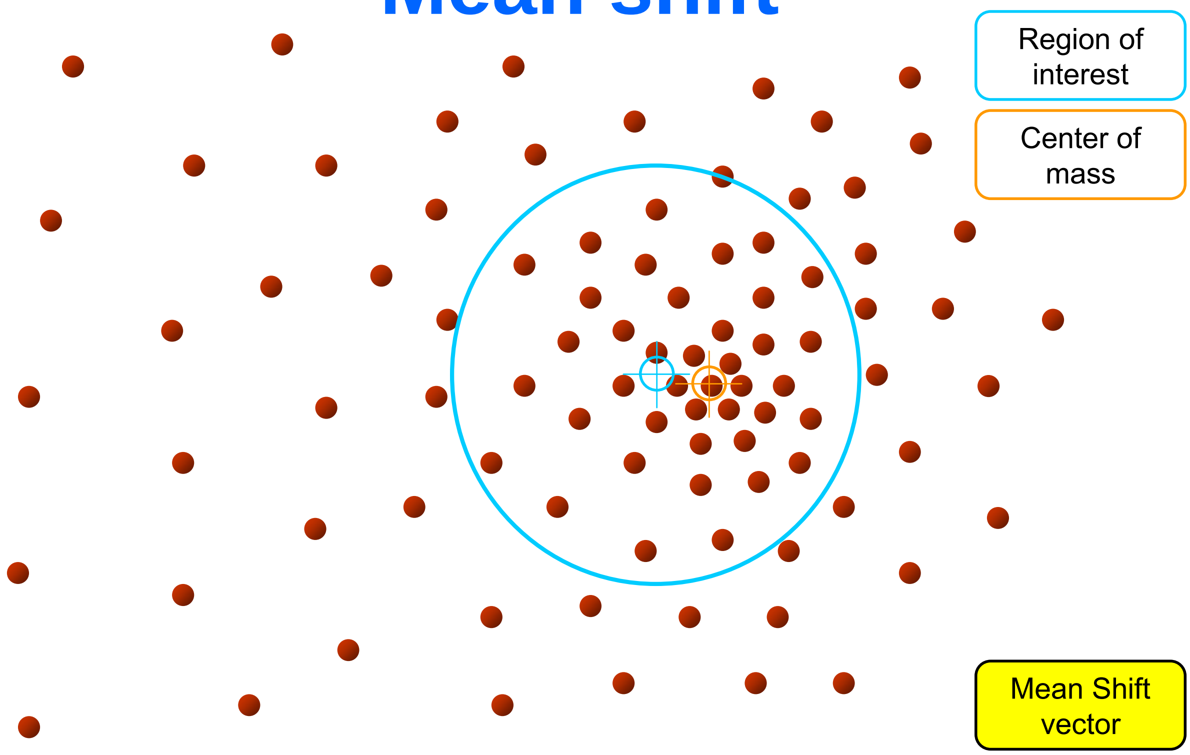
Slide by Y. Ukrainitz & B. Sarel

Mean shift



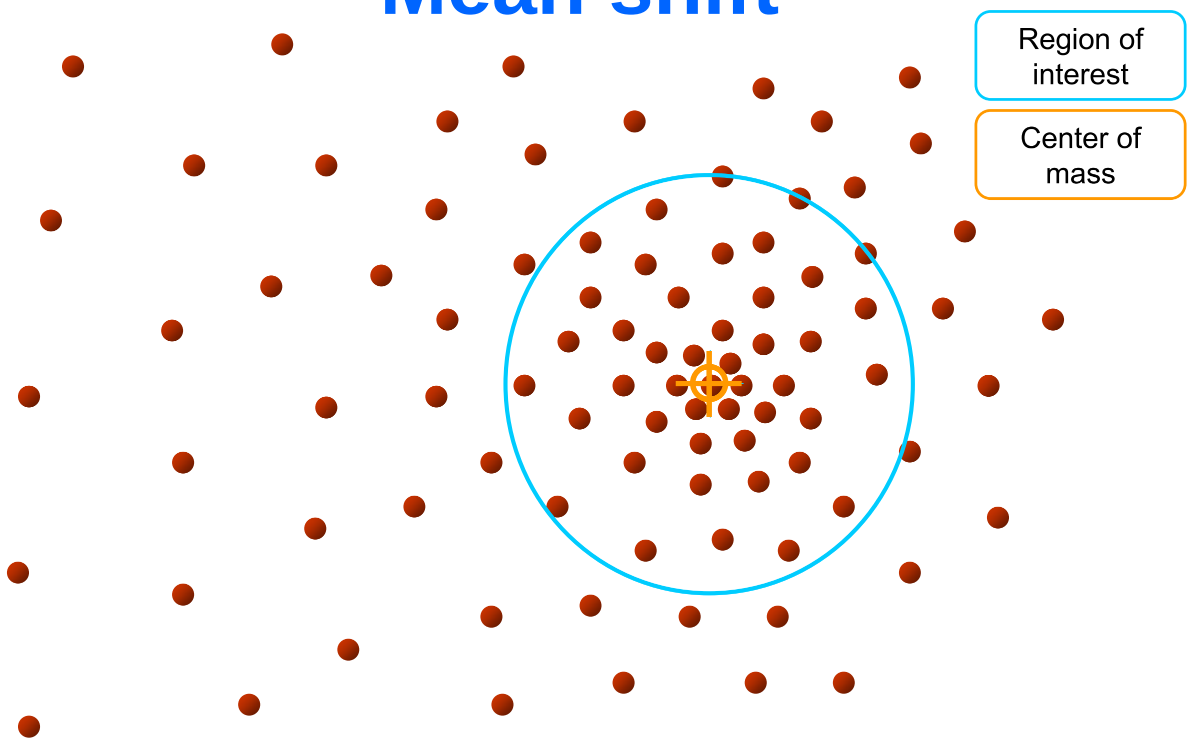
Slide by Y. Ukrainitz & B. Sarel

Mean shift



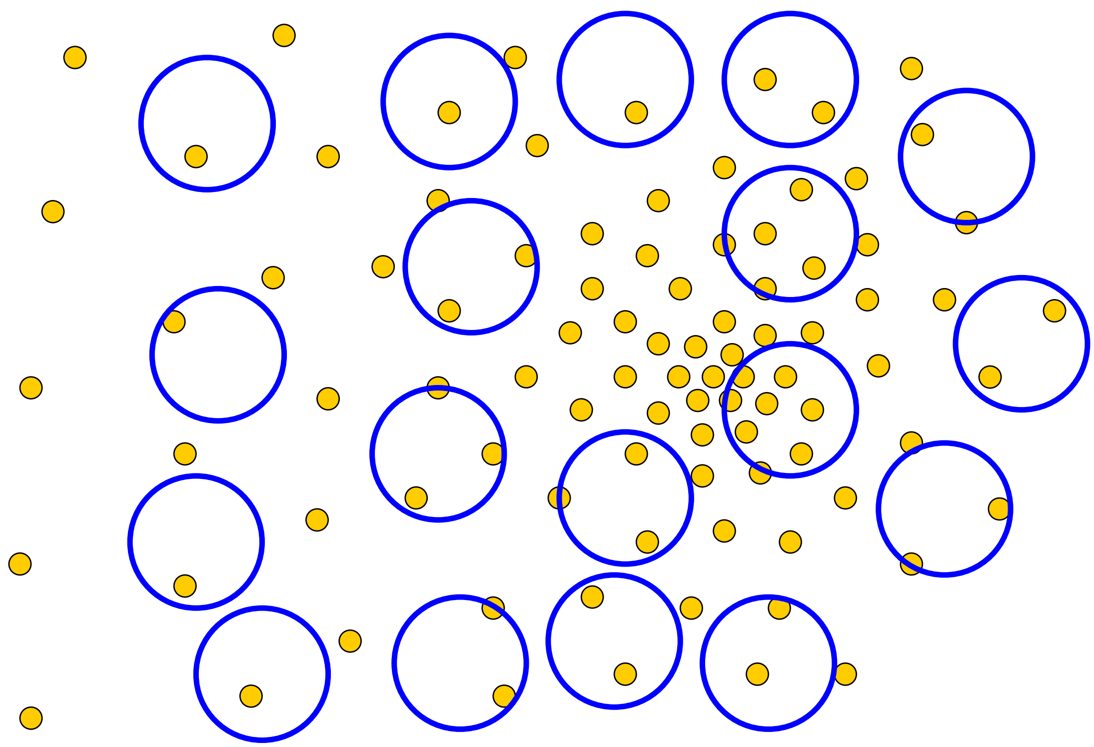
Slide by Y. Ukrainitz & B. Sarel

Mean shift



Slide by Y. Ukrainitz & B. Sarel

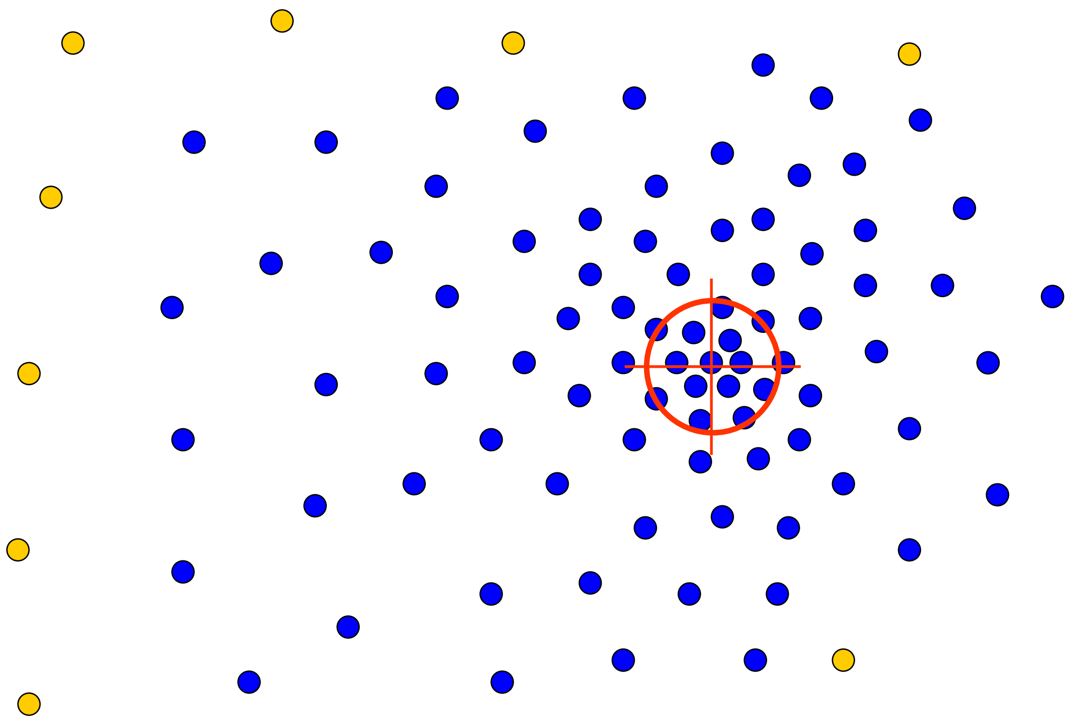
Real Modality Analysis



Tessellate the space with windows

Run the procedure in parallel

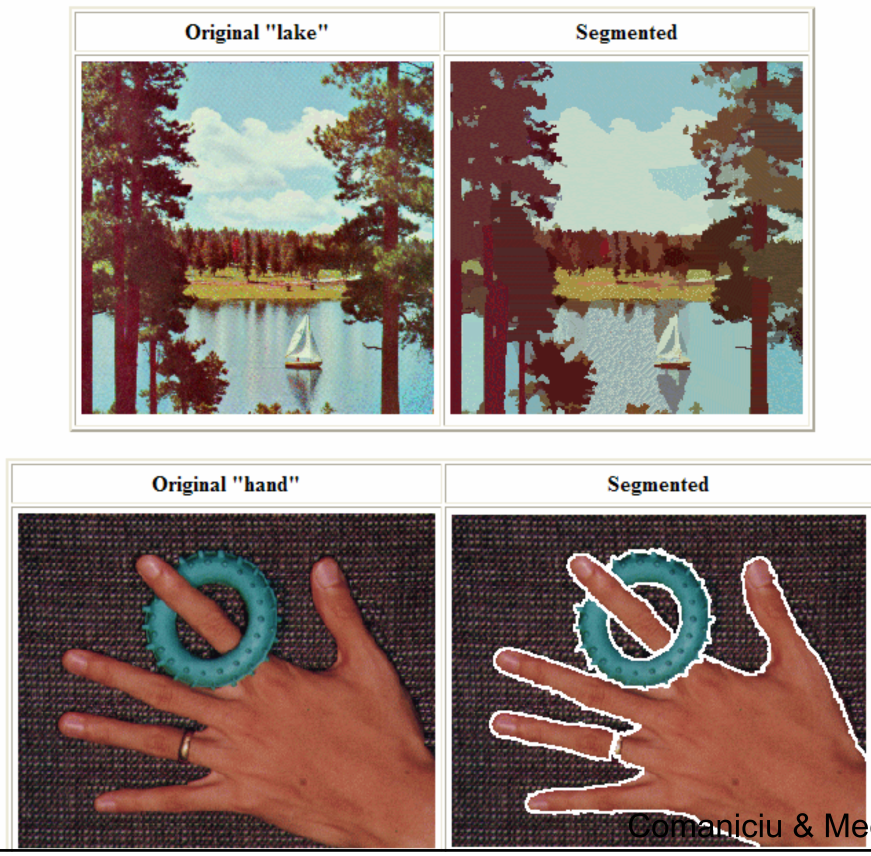
Real Modality Analysis



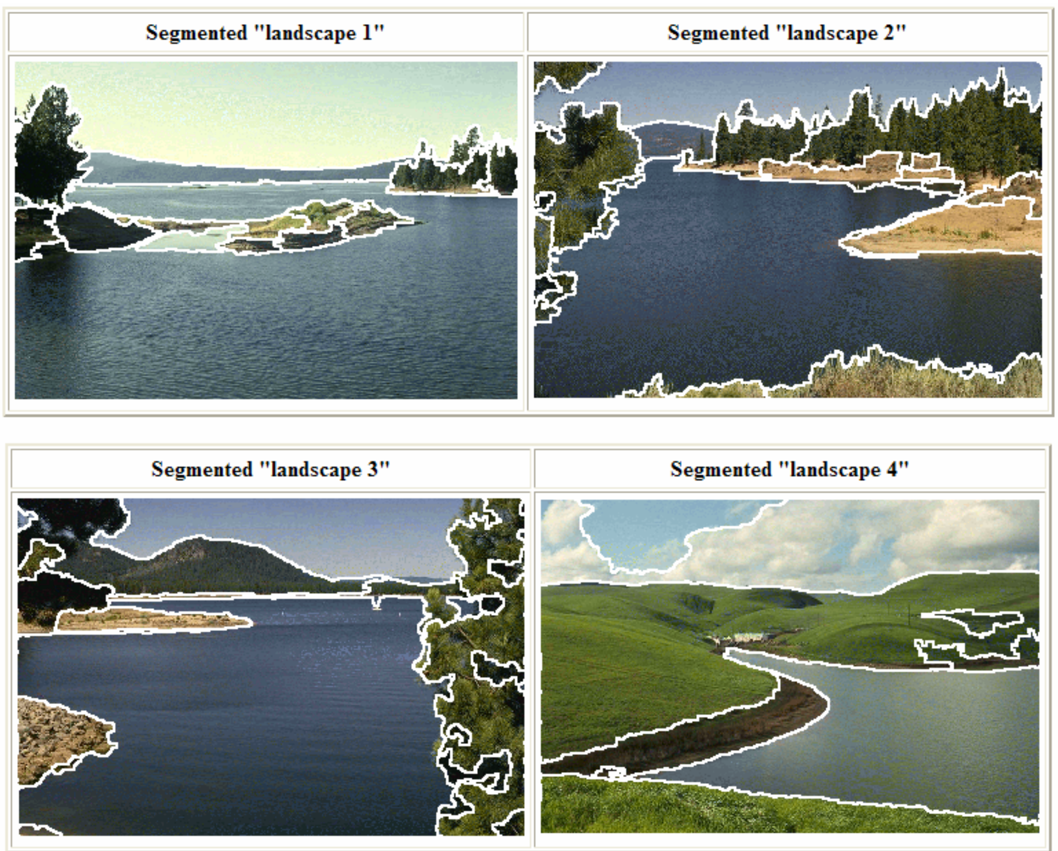
The blue data points were traversed by the windows towards the mode

Slide by Y. Ukrainitz & B. Sarel

Mean shift segmentation



Comaniciu & Meer, PAMI 2002



Comaniciu & Meer, PAMI 2002

CAMSHIFT [G. Bradski]

- Variant on mean shift: “Continuously adaptive mean shift”
- Shown for face tracking for a user interface
- Want mode of color distribution in a video scene
- Dynamic distribution now, since there is motion, scale change
- Adjust search window size dynamically, based on area of face

CAMSHIFT [G. Bradski]

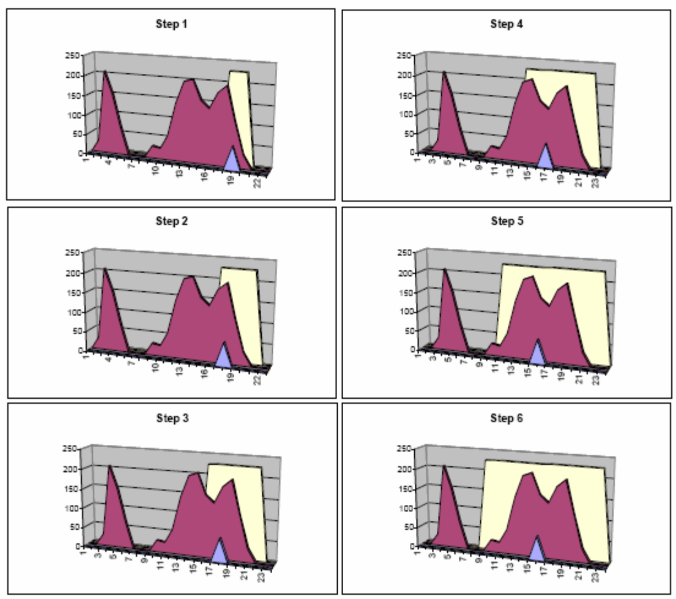


Figure 4: CAMSHIFT in operation down the left then right columns

CAMSHIFT [G. Bradski]

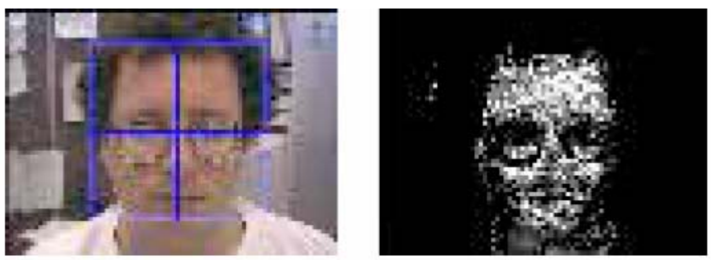


Figure 6: A video image and its flesh probability image

Figure 7: Orientation of the flesh probability distribution marked on the source video image

CAMSHIFT [G. Bradski]

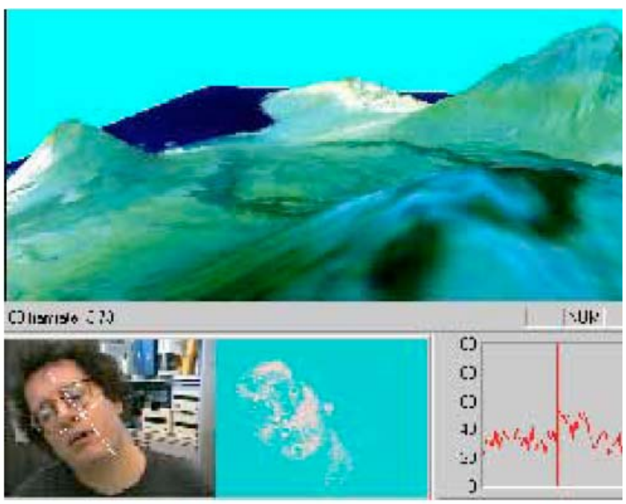


Figure 13: CAMSHIFT-based face tracker used to over a 3D graphic's model of Hawaii

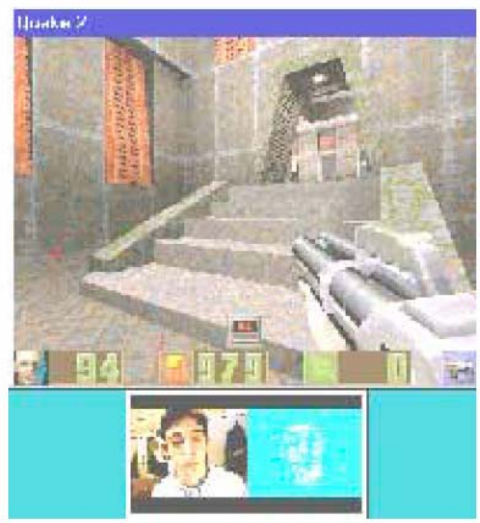


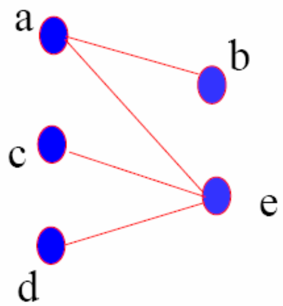
Figure 12: CAMSHIFT-based face tracker used to play Quake 2 hands free by inserting control variables into the mouse queue

Mean shift

- Pros:
  - Does not assume shape on clusters (e.g. elliptical)
  - One parameter choice (window size)
  - Generic technique
  - Find multiple modes
- Cons:
  - Selection of window size
  - Does not scale well with dimension of feature space (but may insert approx. for high-d data…)

Graph-theoretic clustering

Graph representation

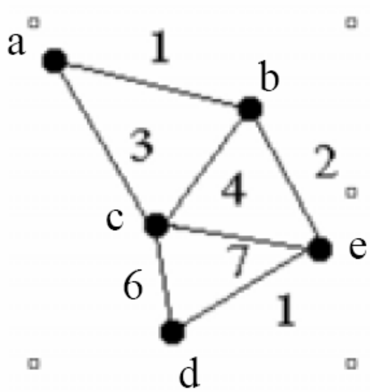


$$
\begin{array}{c} \\ a \\ b \\ c \\ d \\ e \end{array}
\begin{array}{c}
\begin{array}{ccccc} a & b & c & d & e \end{array} \\
\begin{bmatrix}
0 & 1 & 0 & 0 & 1 \\
1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 1 \\
1 & 0 & 1 & 1 & 0
\end{bmatrix}
\end{array}
$$

Adjacency Matrix

* From Khurram Hassan-Shafique CAP5415 Computer Vision 2003

Weighted graph representation



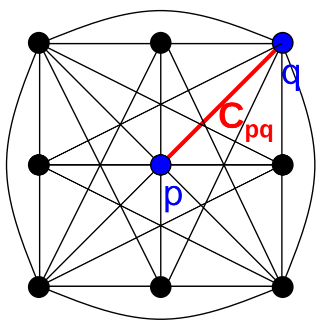
$$\begin{bmatrix} 0 & 1 & 3 & \infty & \infty \\ 1 & 0 & 4 & \infty & 2 \\ 3 & 4 & 0 & 6 & 7 \\ \infty & \infty & 6 & 0 & 1 \\ \infty & 2 & 7 & 1 & 0 \end{bmatrix}$$

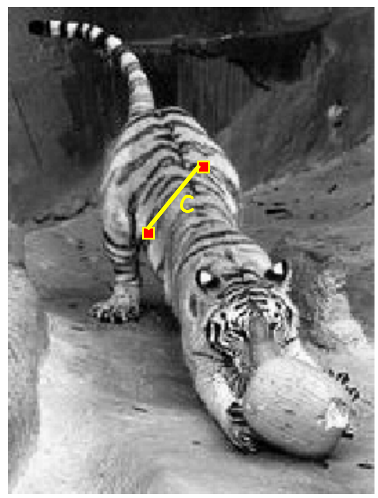
Weight Matrix

(“Affinity matrix”)

* From Khurram Hassan-Shafique CAP5415 Computer Vision 2003

Images as graphs



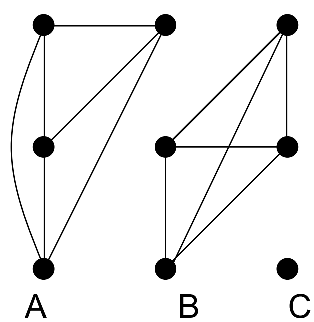


Fully-connected graph

- node for every pixel
- link between *every* pair of pixels, **p**,**q**
- similarity $\mathbf{c_{pq}}$ for each link
  - » similarity is *inversely proportional* to difference in color and position

Slide by Steve Seitz

Segmentation by Graph Cuts



Break Graph into Segments

- Delete links that cross between segments
- Easiest to break links that have low similarity
  - similar pixels should be in the same segments
  - dissimilar pixels should be in different segments

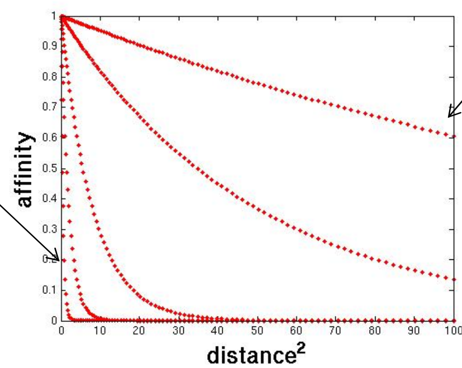
Slide by Steve Seitz

Measuring affinity

- One possibility:

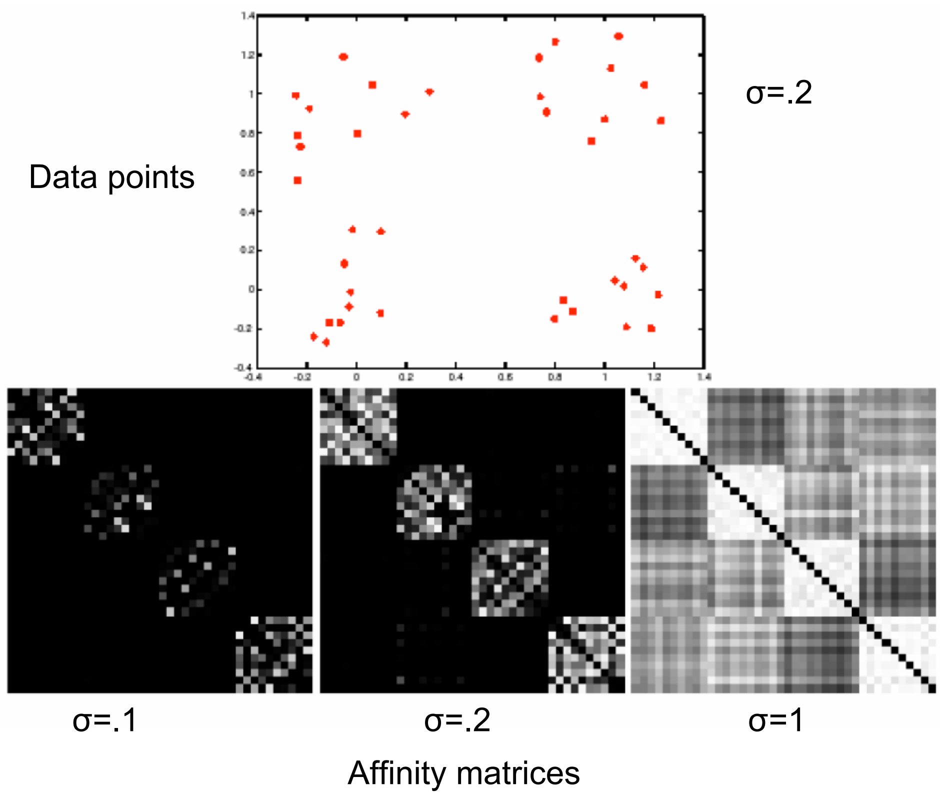
$$aff(x,y) = \exp\left\{-\left(\frac{1}{2\sigma_d^2}\right)\left(\|x-y\|^2\right)\right\}$$

Small sigma: group only nearby points



Large sigma: group distant points

Scale affects affinity



Eigenvectors and cuts

- Want a vector *a* giving the association between each element and a cluster
- Want elements within this cluster to have strong affinity with one another
- Maximize $a^T A a$

  subject to the constraint $a^T a = 1$

- Eigenvalue problem : choose the eigenvector of A with largest eigenvalue

Rayleigh Quotient

- Given a symmetric matrix $\boldsymbol{A}$, find a vector $\boldsymbol{x}$ such that
- $\boldsymbol{x}^T \boldsymbol{A} \boldsymbol{x}$ is maximum AND
- $\|\boldsymbol{x}\|^2 = 1$

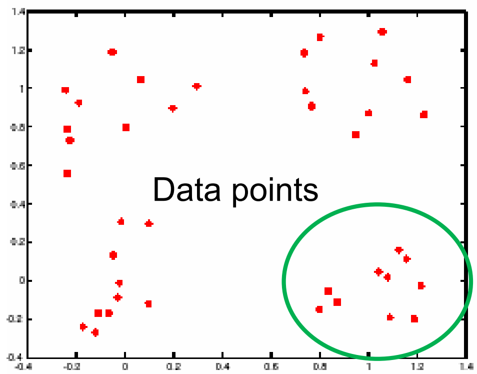
$$\updownarrow$$

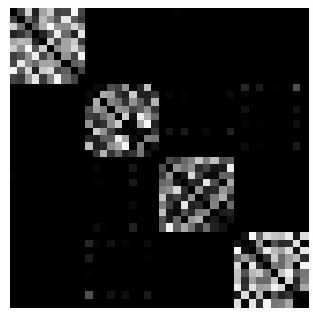
- Find $\boldsymbol{x}$ such that $\frac{\boldsymbol{x}^T \boldsymbol{A} \boldsymbol{x}}{\boldsymbol{x}^T \boldsymbol{x}}$ is maximum.

The solution to this problem is given by the following theorem:

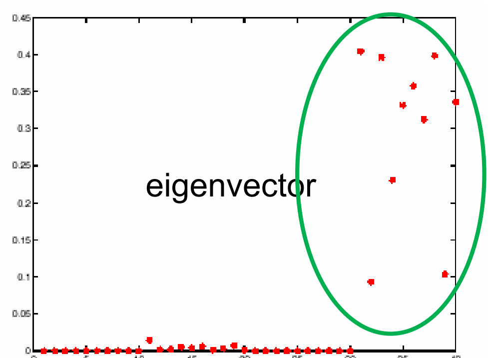
- $\frac{\boldsymbol{x}^T \boldsymbol{A} \boldsymbol{x}}{\boldsymbol{x}^T \boldsymbol{x}}$ reaches its absolute maximum when $\boldsymbol{x}$ is an eigenvector of $\boldsymbol{A}$ corresponding to the *largest* eigenvalue λ_{max}.

Example





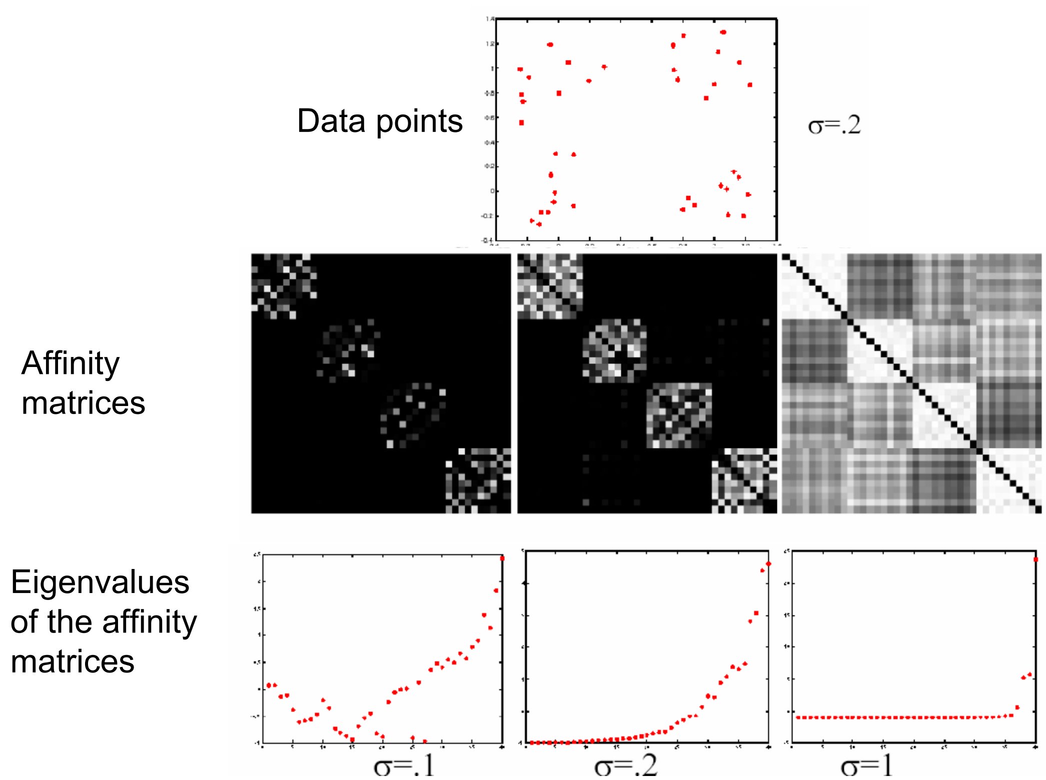
Affinity matrix



Eigenvectors and multiple cuts

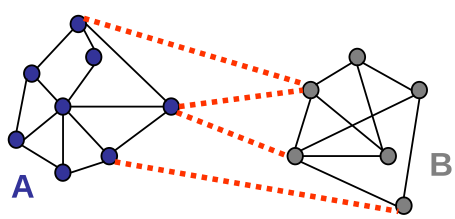
- Use eigenvectors associated with k largest eigenvalues as cluster weights
- Or re-solve recursively

Scale affects affinity, number of clusters



Graph partitioning: Min cut

- Select bipartition that minimizes cut value, i.e., total weight of edges removed



A, B are disjoint sets

$$cut(A,B) = \sum_{p \in A, q \in B} c_{p,q}$$

Fast algorithms exist for this

Min cut

- Problem: weight of cut proportional to number of edges in the cut; min cut penalizes large segments

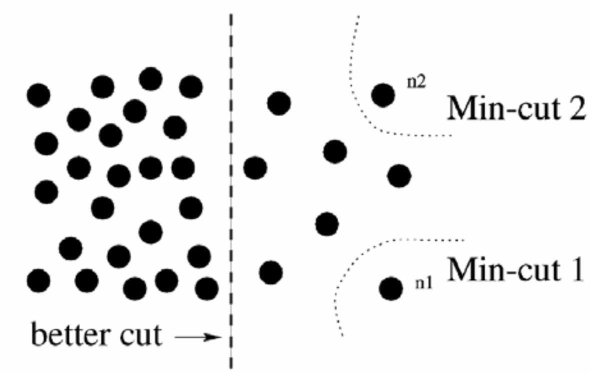


Fig. 1. A case where minimum cut gives a bad partition.

[Shi & Malik, 2000 PAMI]

Normalized cuts

- First eigenvector of affinity matrix captures within cluster similarity, but not across cluster difference
- Would like to maximize the within cluster similarity relative to the across cluster difference

Normalized cuts

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

  = total connection from nodes in A to all nodes in graph (V)

- To get disjoint groups A, B for which within cluster similarity is high compared to their association with rest of graph

Normalized cuts

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

= total connection from nodes in A to all nodes in graph (V)

- Maximize $$\left(\frac{assoc(A,A)}{assoc(A,V)}\right) + \left(\frac{assoc(B,B)}{assoc(B,V)}\right)$$

Normalized cuts

- Exact discrete solution is NP-complete [Papadimitrou 1997] ☹
- But can efficiently approximate via generalized eigenvalue problem [Shi & Malik] ☺

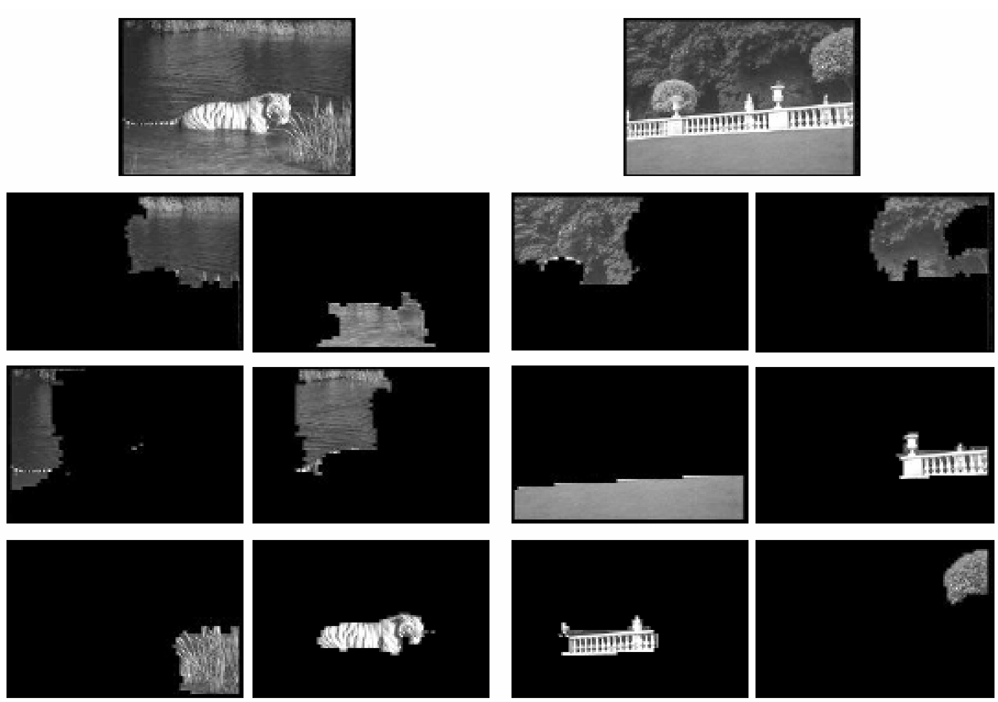
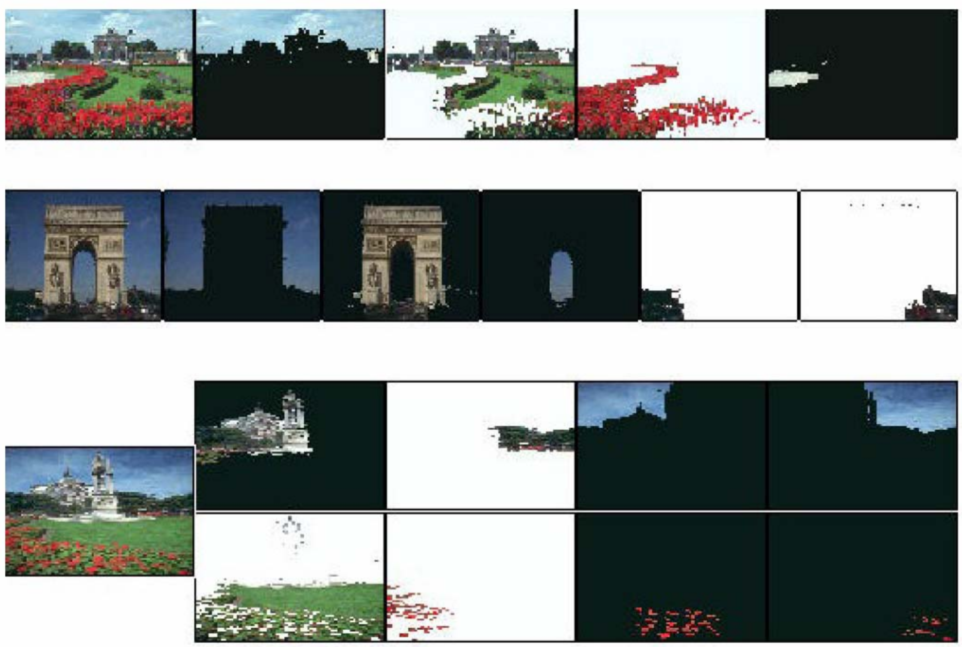


Figure from “Image and video segmentation: the normalised cut framework”,
by Shi and Malik, copyright IEEE, 1998

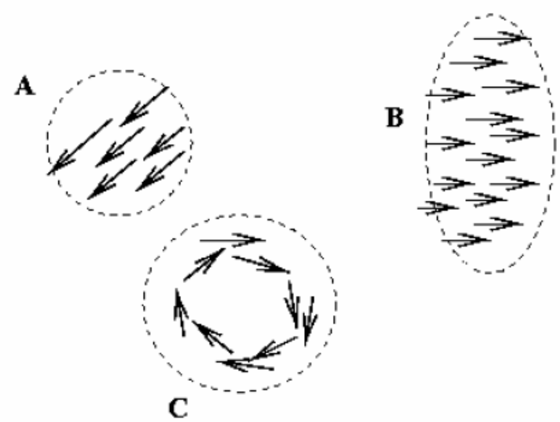
Figure from “Normalized cuts and image segmentation,” Shi and Malik, copyright IEEE, 2000



http://www.cs.berkeley.edu/~malik/papers/SM-ncut.pdf

Examples of grouping in vision

- Assume that neighboring pixels with the same motion are part of the same object
- Objects A, B translate, C rotates



Motion segmentation

Shapiro & Stockman, P. Duygulu

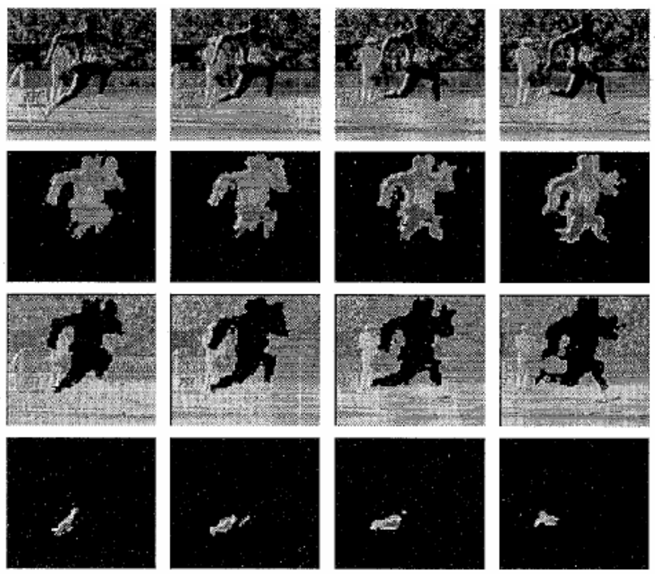


Figure 5: The first row shows an image sequence of Carl Lewis running. Notice that the background is moving to the left as the camera is panning to keep the runner in the center of the image, and therefore background subtraction would not work as an image segmentation technique. The original image size is 200 × 190, and image patches of size 3 × 3 is used to construct the partition graph. Each of the image patches are connected to others that are less than 5 superpixels and 3 image frames away. Row 2 to 4 show the motion segmentation produced by our algorithm. Note these regions found corresponds the runner in row 2, moving background in row 3, and the left lower leg in row 4. The left lower leg is segmented from the runner because it undergoes significant upward rotation in these seven image frames. By recursive cuts and by lowering the maximum allowed *Ncut* value, the other moving limbs can be found.

Features = measure of motion/velocity

Motion Segmentation and Tracking Using Normalized Cuts [Shi & Malik 1998]

K-means vs. graph cuts, mean shift

- Graph cuts / spectral clustering, mean shift: do not require model of data distribution

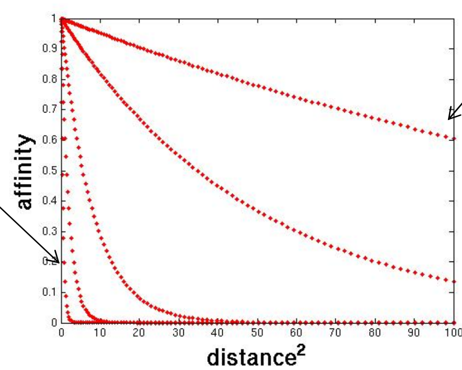
Scale selection for spectral clustering

- How to select scale for analysis?
- What about multi-scale data?

Measuring affinity

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

Small sigma: group only nearby points



Large sigma: group distant points

Segmentation: Caveats

- Can’t hope for magic
- Intertwined with recognition problem
- Have to be careful not to make hard decisions too soon
- Hard to evaluate

Next

- Fitting for grouping
- Read F&P Chapter 15 (ignore fundamental matrix sections for now)
- Problem set 1 due Tues. – estimate time