CS 664
Segmentation (2)

Daniel Huttenlocher

Cornell University
Faculty of Computing and Information Science

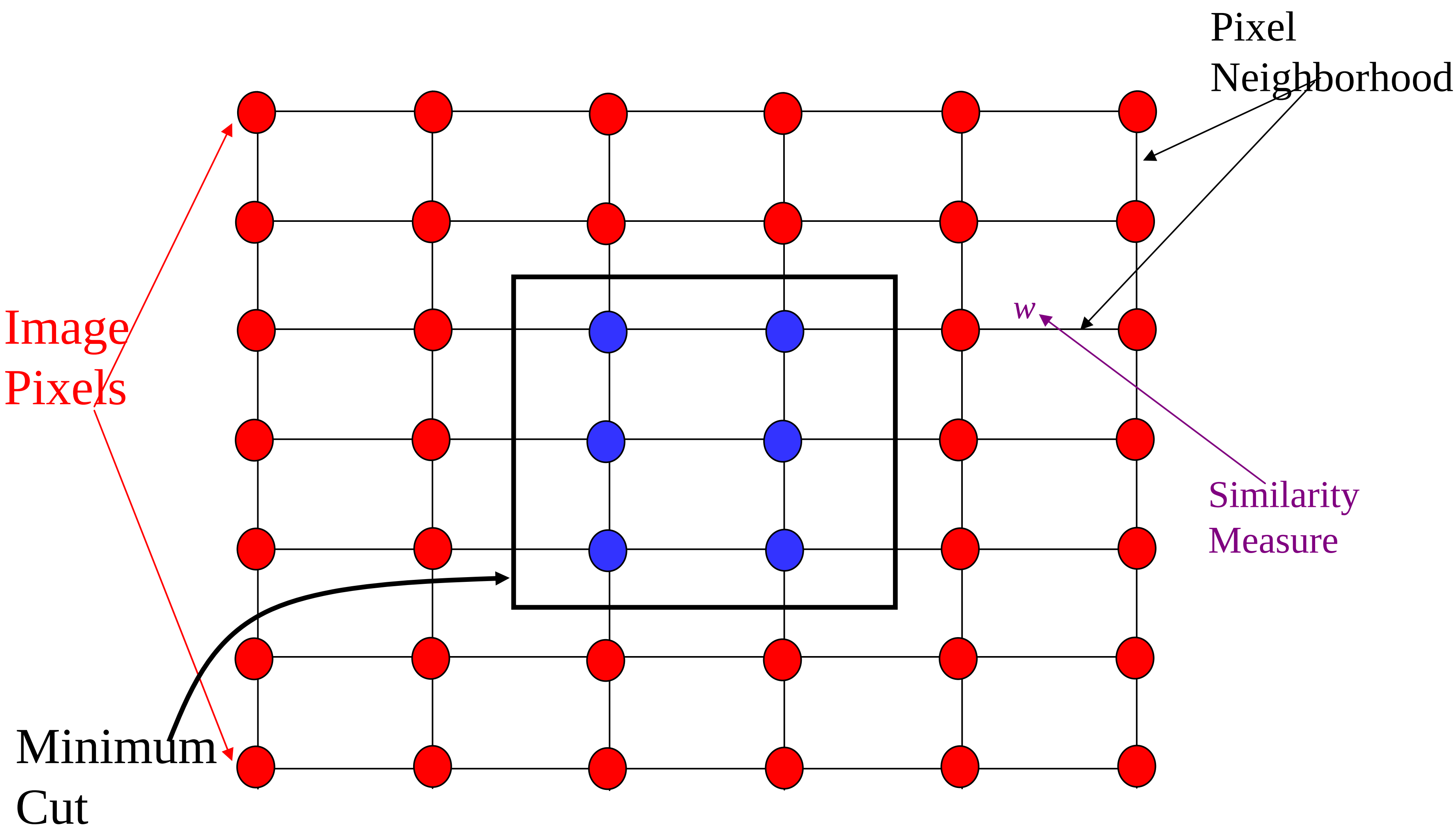
Recap

- Last time covered perceptual organization more broadly, focused in on pixel-wise segmentation
- Covered local graph-based methods such as MST and Felzenszwalb-Huttenlocher method
- Today
  - Cut-based methods such as grab cut, normalized cuts
  - Iterative local update methods such as mean shift

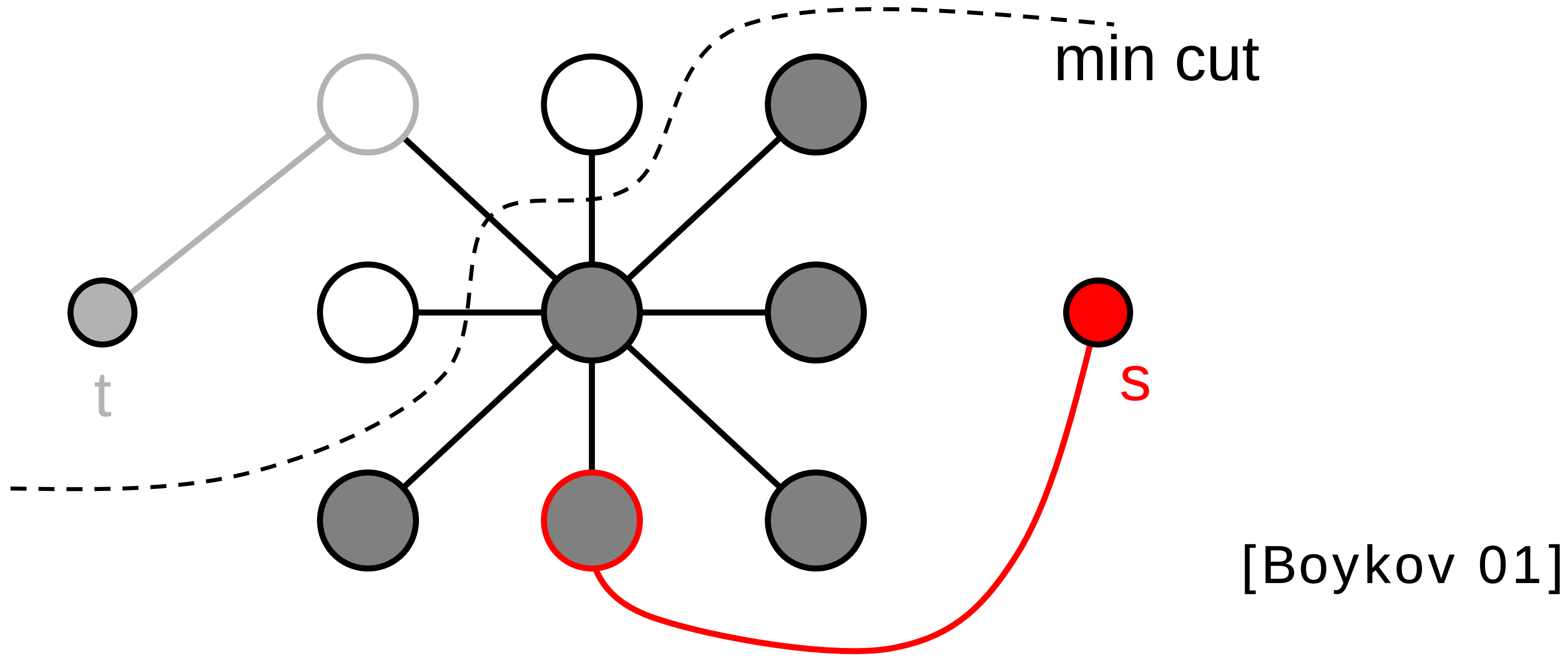
Cut Based Techniques

- For costs, natural to consider minimum cost cuts
  - Removing edges with smallest total cost, that cut graph in two parts
  - Graph only has finite-weight edges
- Manually assisted techniques, foreground vs. background
- General segmentation, recursively cut resulting components
  - Question of when to stop

Image Segmentation & Minimum Cut



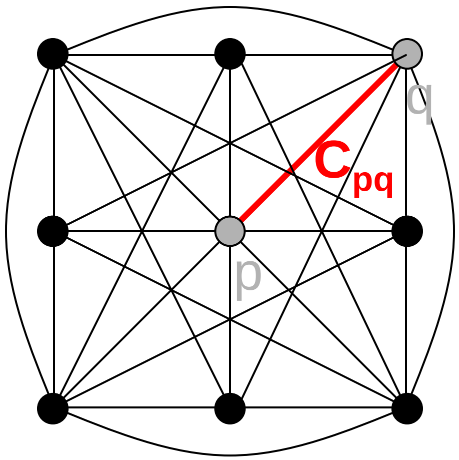
Segmentation by Min (s-t) Cut

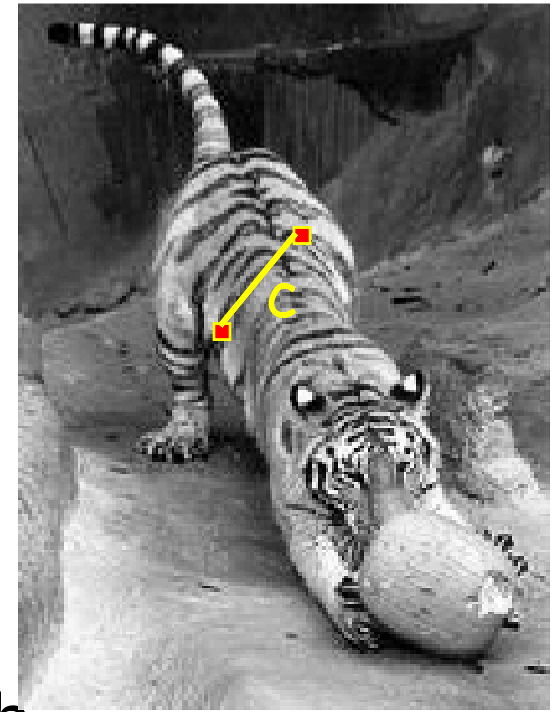


- Manually select a few fg and bg pixels
  - Infinite cost link from each bg pixel to the "t" node, and each fg pixel to "s" node
  - Compute min cut that separates s from t

Grabcut [Rother et al., SIGGRAPH 2004]

Automatic Cut-Based Segmentation

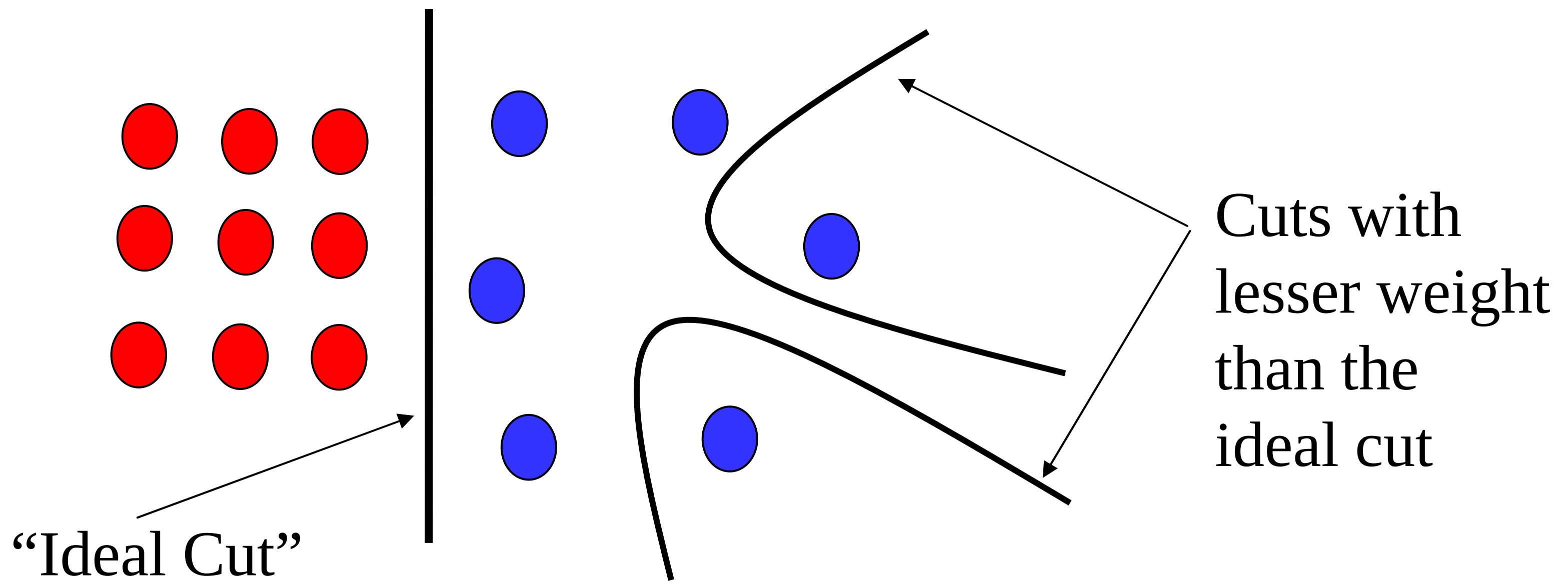




- *Fully-connected* graph
  - Node for every pixel
  - Link between *every* pair of pixels, p,q
  - Cost for each link measures *similarity*

Drawbacks of Minimum Cut

- Weight of cut proportional to number of edges – preference for small regions
  - Motivation for Shi-Malik normalized cuts



* Slide from Khurram Hassan-Shafique CAP5415 Computer Vision 2003

Normalized Cuts

- A number of normalization criteria have been proposed
- One that is commonly used [Shi&Malik]

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

- Where cut(A,B) is standard definition

$$\sum_{i \in A, j \in B} w_{ij}$$

- And $\text{assoc}(A,V) = \sum_j \sum_{i \in A} w_{ij}$

Cornell University

Computing Normalized Cuts

- Has been shown this is equivalent to an integer programming problem, minimize

$$\frac{y^T (D-W) y}{y^T D y}$$

- Subject to the constraint that $y_i \in \{1,b\}$ and $y^T D 1 = 0$
  - Where 1 vector of all 1's
- W is the affinity matrix
- D is the degree matrix (diagonal)

$$D(i,i) = \sum_j w_{ij}$$

Cornell University

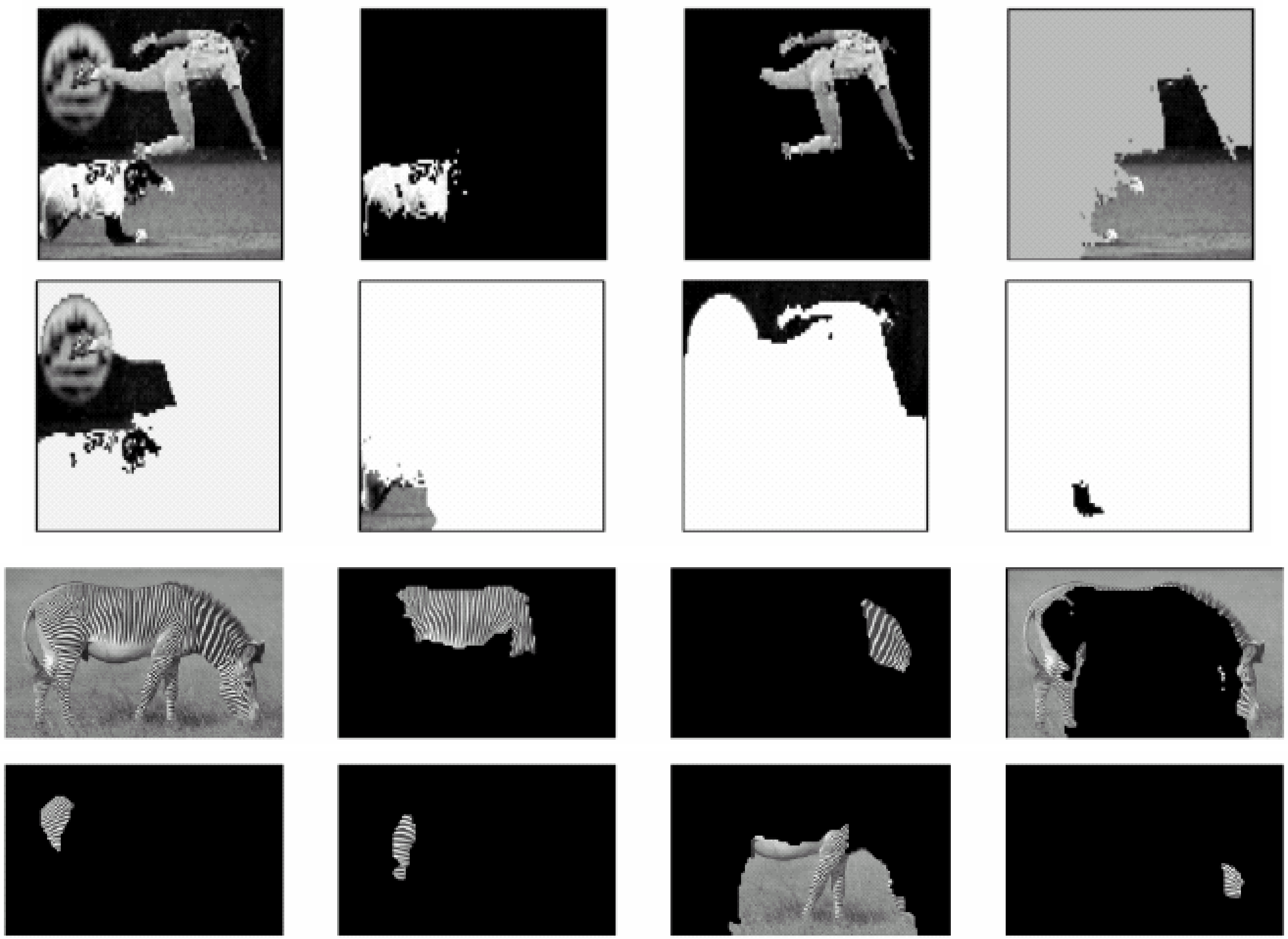
Approximating Normalized Cuts

- Integer programming problem NP hard
  - Instead simply solve continuous (real-valued) version
  - This corresponds to finding second smallest eigenvector of

$$(D-W)y_i = \lambda_i Dy_i$$

- Widely used method
  - Works well in practice
    - Large eigenvector problem, but sparse matrices
    - Often resolution reduce images, e.g, 100x100
  - But no longer clearly related to cut problem

Normalized Cut Examples



Another Look [Weiss 99]

- Consider eigen analysis of affinity matrix

$$W = [\ w_{ij}\ ]$$

  - Note W is symmetric; for images $w_{ij} = w_{ji}$
  - W also essentially block diagonal
    - With suitable rearrangement of rows/cols so that vertices with higher affinity have nearer indices
    - Entries far from diagonal are small (though not quite zero)
- Eigenvectors of W
  - Recall for real, symmetric matrix forms an orthogonal basis
    - Axes of decreasing “importance”

Structure of W

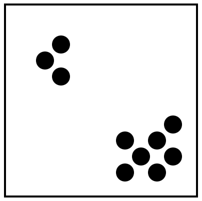
- Eigenvectors of block diagonal matrix consist of eigenvectors of the blocks
  - Padded with zeroes
- Note rearrangement so that clusters lie near diagonal only conceptual
  - Eigenvectors of permuted matrix are permutation of original eigenvectors
- Can think of eigenvectors as being associated with high affinity “clusters”
  - Eigenvectors with large eigenvalues
  - Approximately the case

Structure of W

- Consider case of point set where affinities

$$w_{ij}=\exp(-(y_i-y_j)^2/\sigma^2)$$

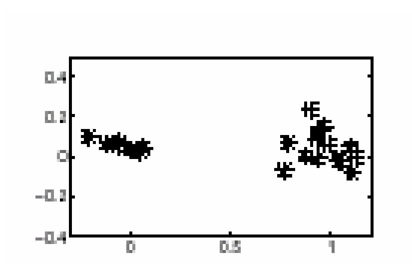
- With two clusters
  - Points indexed to respect clusters for clarity
- Block diagonal form of W
  - Within cluster affinities A, B for clusters
  - Between cluster affinity C



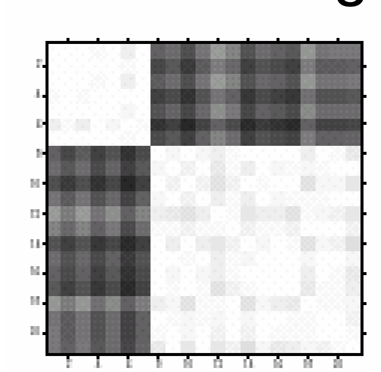
$$W=\begin{pmatrix} A & C \\ C^T & B \end{pmatrix}$$

First Eigenvector of W

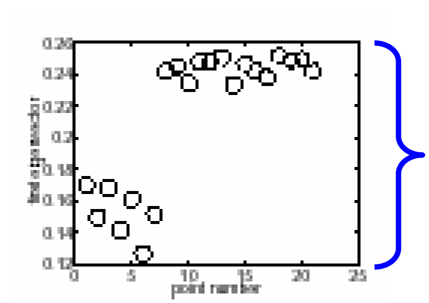
- Recall, vectors x_i satisfying $Wx_i=\lambda_i x_i$
- Consider ordered by eigenvalues λ_i
  - First eigenvector x_1 has largest eigenvalue λ_1
- Elements of first eigenvector serve as "index vector" [Perona,Freeman]
  - Selecting elements of highest affinity cluster



Points in plane



W

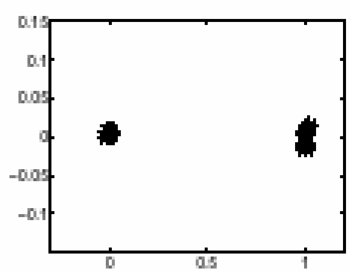


Magnitude of elements

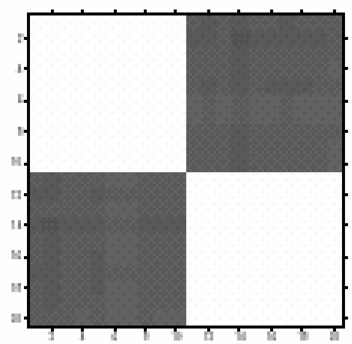
Elements of x_1

Clustering

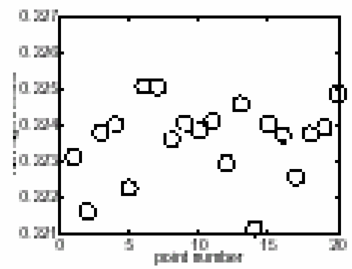
- First eigenvector of W has been suggested as clustering or segmentation criterion
  - For selecting most significant segment
  - Then recursively segment remainder
- Problematic when nonzero non-diagonal blocks (similar affinity clusters)



Points in plane



W



Elements of x_1

Understanding Normalized Cuts

- Intractable discrete graph problem used to motivate continuous (real valued) problem
  - Find second *smallest* “generalized eigenvector”

$$(D-W)x_i = \lambda_i D x_i$$

  - Where D is (diagonal) degree matrix $d_{ii} = \sum_j w_{ij}$
- Can be viewed in terms of first two eigenvectors of normalized affinity matrix
  - Let $N = D^{-1/2} W D^{-1/2}$
  - Note $n_{ij} = w_{ij} / (\sqrt{d_{ii}} \sqrt{d_{jj}})$
    - Affinity normalized by degree of the two nodes

Normalized Affinities

- Can be shown that
  - If x is an eigenvector of N with eigenvalue λ then $D^{-1/2}x$ is a generalized eigenvector of W with eigenvalue $1-\lambda$
  - The vector $D^{-1/2}1$ is an eigenvector of N with eigenvalue 1
- It follows that
  - Second smallest generalized eigenvector of W is ratio of first two eigenvectors of N
  - So ncut uses normalized affinity matrix N and first two eigenvectors rather than affinity matrix W and first eigenvector

Contrasting W and N

- Three simple point clustering examples
  - W, first eigenvector of W, ratio of first two eigenvectors of N (generalized eigenvector of W)

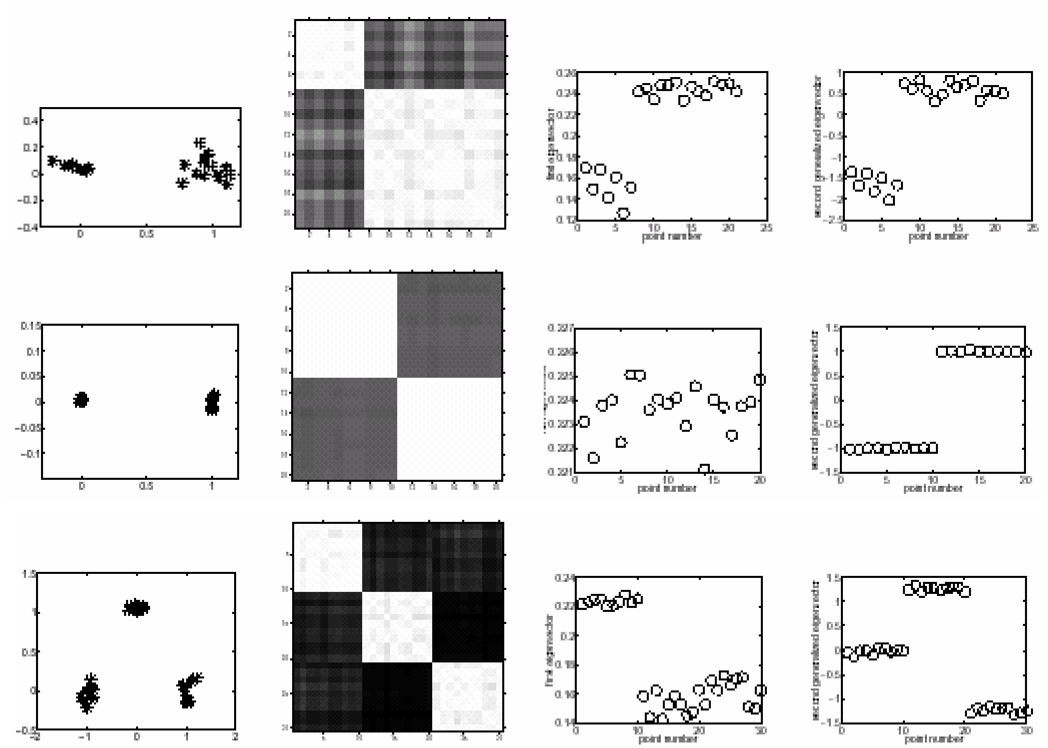
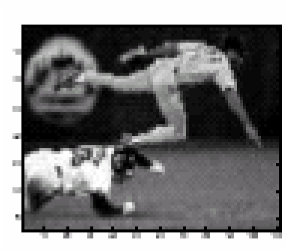


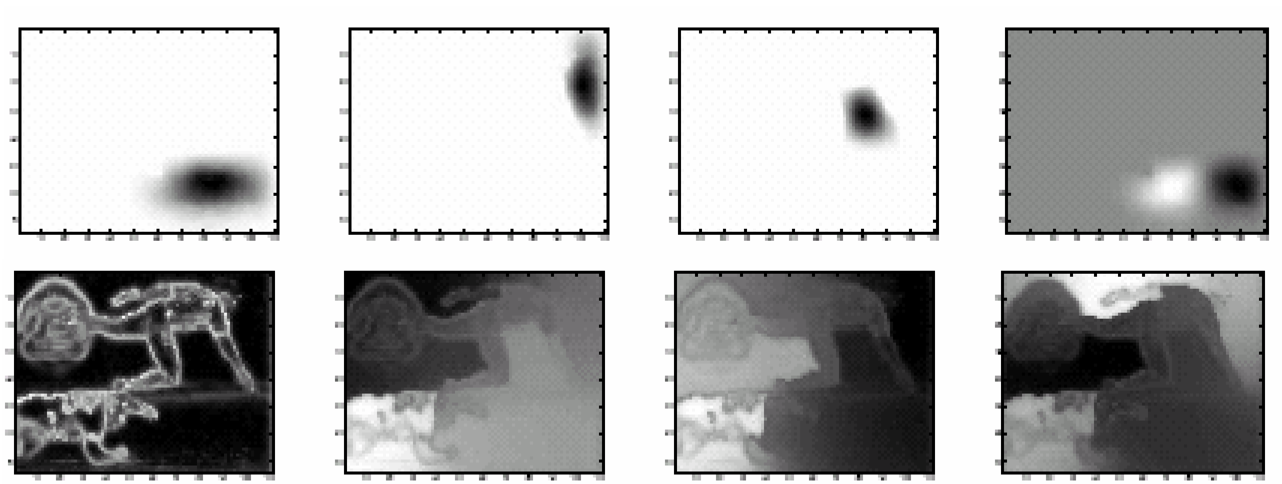
Image Segmentation

- Considering W and N for segmentation
  - Affinity a negative exponential based on distance in x,y,b space
- Eigenvectors of N more correlated with regions



First 4 eigenvectors of W

First 4 eigenvectors of N

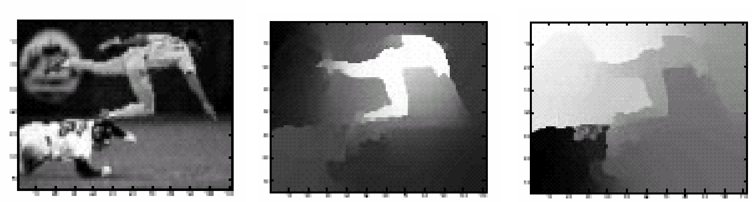


Using More Eigenvectors

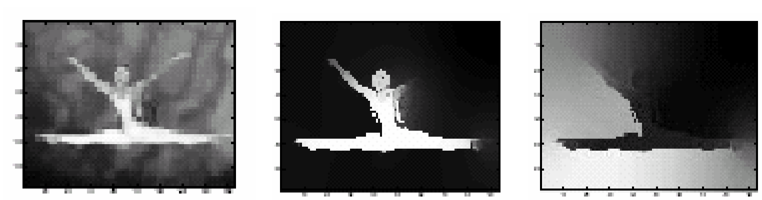
- Based on k largest eigenvectors
  – Construct matrix Q such that (ideally) $q_{ij}=1$ if i and j in same cluster, 0 otherwise
- Let V be matrix whose columns are first k eigenvectors of W
- Normalize rows of V to have unit Euclidean norm
  – Ideally each node (row) in one cluster (col)
- Let $Q=VV^T$
  – Each entry product of two unit vectors

Normalization and k Eigenvectors

- Normalized affinities help correct for variations in overall degree of affinity
  - So compute Q for N instead of W
- Contrasting Q with ratio of first two eigenvectors of N (ncut criterion)
  - More clearly selects most significant region
    - Using k=6 eigenvectors
  - Row of Q matrix vs. ratio of eigenvectors of N



Q N



Q N

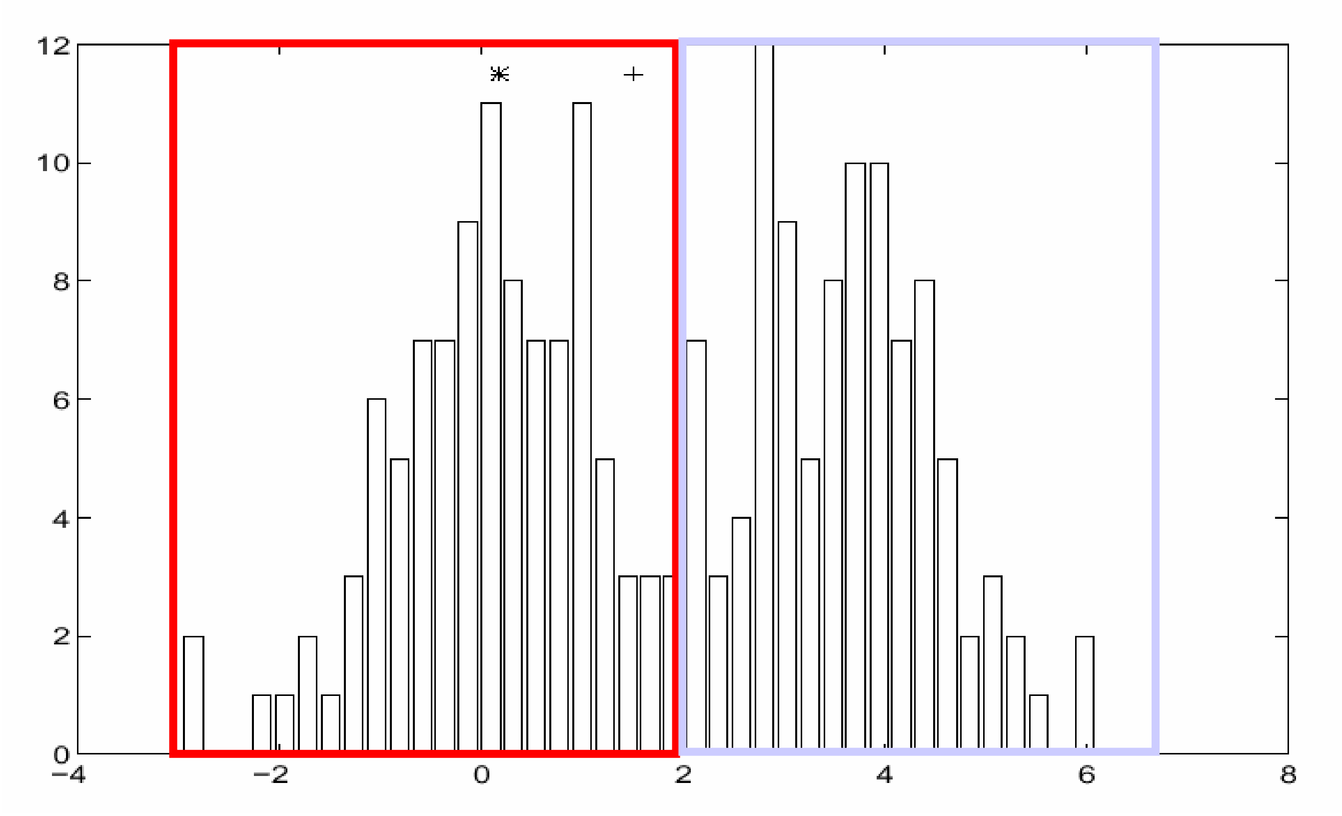
Spectral Methods

- Eigenvectors of affinity and normalized affinity matrices
- Widely used outside computer vision for graph-based clustering
  - Link structure of web pages, citation structure of scientific papers
  - Often directed rather than undirected graphs

Iterative Clustering Methods

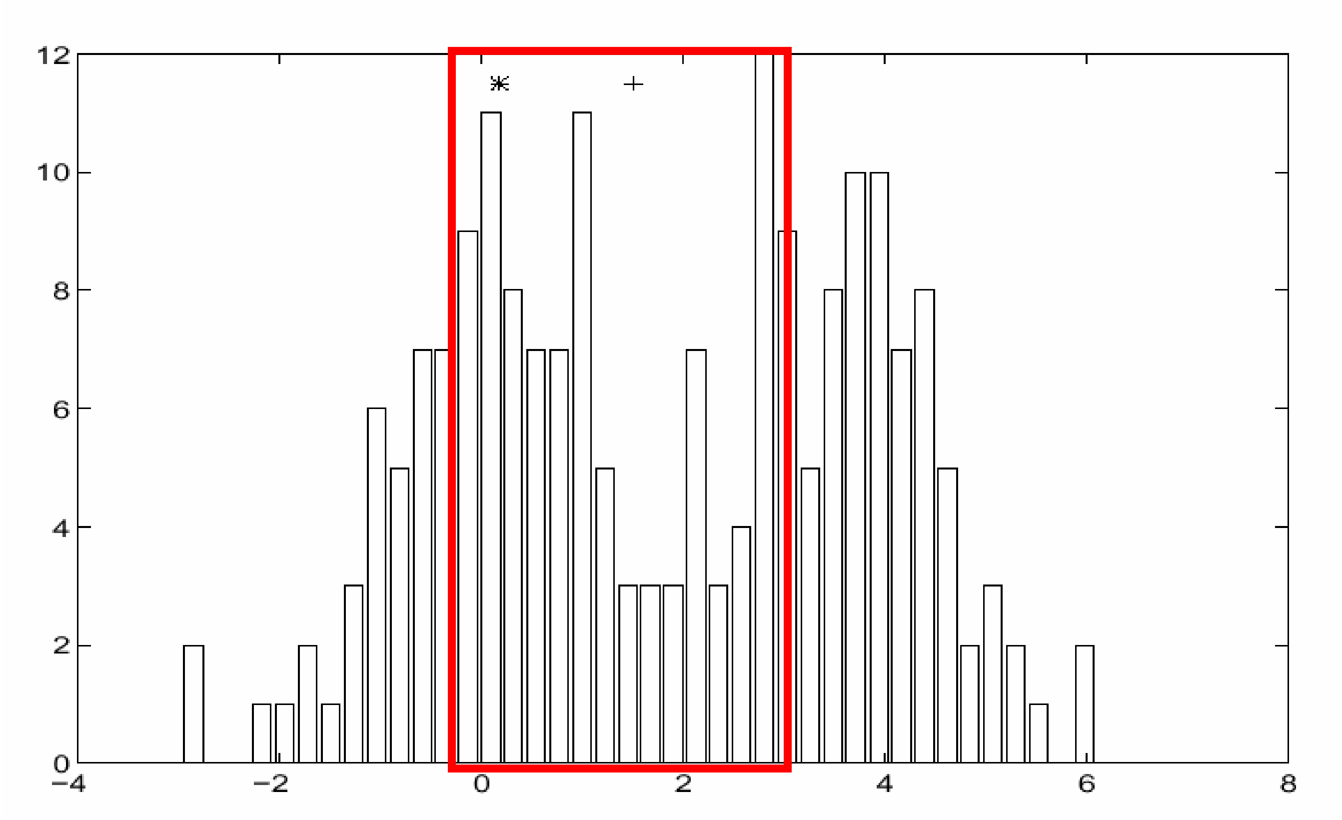
- Techniques such as k-means, but for image segmentation generally have no idea about number of regions
- Mean-shift a nonparametric method

Finding Modes in a Histogram



- How Many Modes Are There?
  - Easy to see, less easy to compute

Mean Shift [Comaniciu & Meer]



- Iterative Mode Search
  1. Initialize random seed, and window W
  2. Calculate center of gravity (“mean”) of W and shift

Mean Shift

- Used both for segmentation and for edge preserving filtering
- Operates on collection of points $X=\{x_1, \ldots, x_n\}$ in R^d
- Replace each point with value derived from mean shift procedure
  - Searches for a local density maximum by repeatedly shifting a d-dimensional hyper-sphere of fixed radius h
  - Differs from most clustering, such as k-means in that no fixed number of clusters

Mean Shift Procedure

- For given point $x \in X$ let $y_1, \ldots, y_T$ denote successive locations of that point

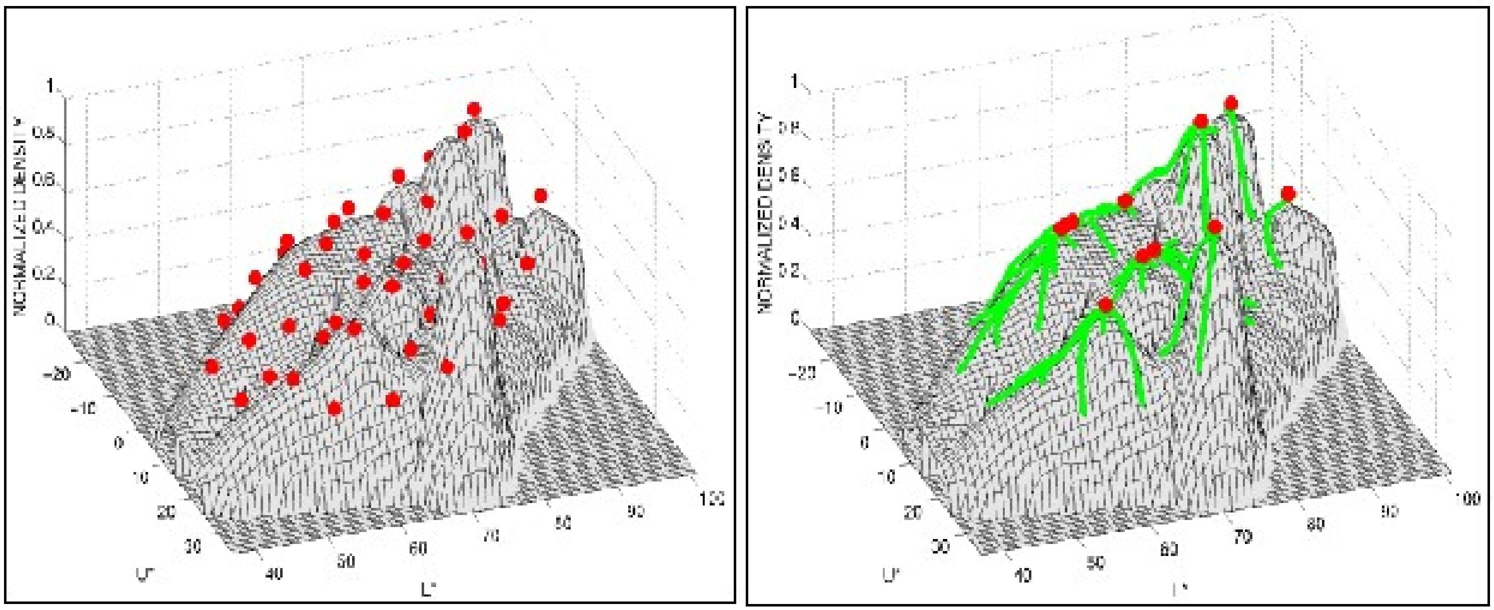
$$y_1 = x$$

$$y_{k+1} = 1/|S(y_k)| \sum_{x \in S(y_k)} x$$

  – Where $S(y_k)$ is the subset of X contained in a hyper-sphere of radius h centered at y_k
    - The radius h is a fixed parameter of the method
- For a point set X, the mean shift procedure is applied separately to all the points

Mean-Shift

- Initialize window around each point
  - Where it shifts determines which region it's in
  - Multiple points will shift to the same region



Mean shift trajectories

Mean Shift Image Filtering

- Map each image pixel to point in u,v,b space

  $x_i = (u_i, v_i, b_i/\sigma)$

  – Analogous for color images, with three intensity values instead of one

  – Scale factor σ normalizes intensity vs. spatial dimensions

- Perform mean shift for each point

  – Let $Y_i = (U_i, Vi, B_i)$ denote mean shifted value

- Assign result $z_i = (u_i, v_i, B_i)$

  – Original spatial coords, mean shifted intensity

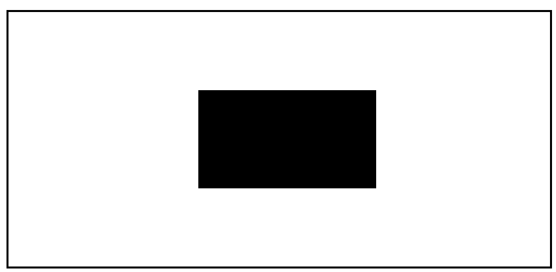
Mean Shift Example

Mean Shift Example

Figure 2: The *house* image, 255 × 192 pixels, 9603 colors.

Edge Preserving Filtering

- Mean shift tends to preserve edges
- Edges are where intensity is changing rapidly
- Rapid changes in intensity will result in lower density regions in joint spatial-intensity space
- Mean shift finds local density maxima

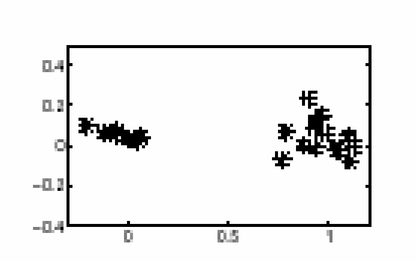


Mean Shift Clustering

- Run mean shift procedure for each point
- Cluster resulting convergence points that closer than some small constant
- Assign each point label of its cluster
- Analogous to filtering, but with added step of merging cluster that are nearby in the joint spatial-intensity domain

About Mean Shift

- Convergence to local density maximum
  - Where “local” determined by sphere radius
- Consider simple point set



- Over wide range of sphere radii end up with two clusters
  - Relationship to MST