ECE 516 / CS 532
Computer Vision

Class 11
March 5, 2008

Pradeep Sen
Advanced Graphics Lab

Announcements

- HW 3 will be out this weekend

Last time

- Introduction to segmentation

Today

- More on segmentation and fitting models

Segmentation using eigenvalues

- A good cluster is one where elements that are strongly associated with the cluster have large values connecting one another in the affinity matrix

Segmentation using eigenvectors

- Assume you have affinity matrix *A*
- Suppose you have *k* elements and *c* clusters.
- The *n*-th cluster vector can be written by a *k*-component vector (w_n) with float values that tell us how likely is an element in the cluster
- We can set a maximization problem with a constraint
- Solve using Lagrange multipliers
- Turns out that cluster vector w_n is the eigenvector of the affinity matrix *A*

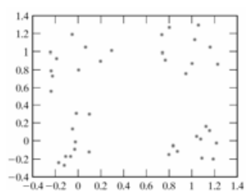
Aside… Eigenvectors

- An eigenvector of a linear transformation is a vector which is only scaled by the linear transformation

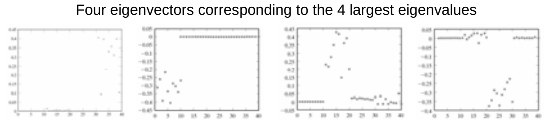
Eigenvalue method

```
Construct an affinity matrix
Compute the eigenvalues and eigenvectors of the
  affinity matrix
Until there are sufficient clusters
   Take the eigenvector corresponding to the
   largest unprocessed eigenvalue
   Zero all components corresponding to elements
   that have already been clustered, and
   threshold the remaining components to
   determine which element belongs to this
   cluster
   If all elements have been accounted for,
   there are sufficient clusters
end
```


Example



Four eigenvectors corresponding to the 4 largest eigenvalues



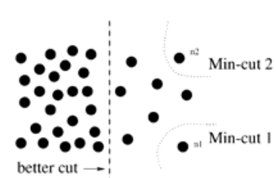
Normalized cuts

- Basic idea is to cut the graph into two connected sub-graphs such that the cut is a small fraction of the total affinity within each group.
- Formalized by a simple formula (see notes)
- Create a vector *y* with components {1, -b} depending which cluster each element belongs to
- Along with a degree matrix *D*, we derive a function to minimize

Minimization for normalized cuts

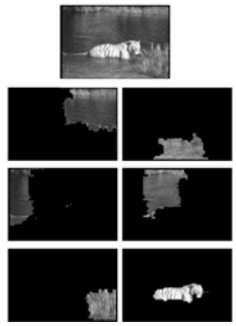
- In general, this is an NP-hard problem
- If we relax the constraints on *y* and allow it to take on real values, then we can solve using eigenvalues

Why not simply use min cut?

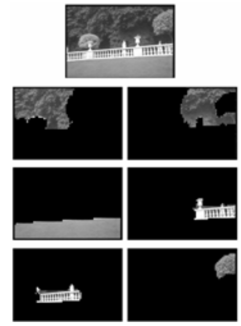


From Shi and Malik (2000)

Examples



Examples



Identifying features in images

- Given an image, suppose we want to identify particular features for segmentation
- For example, identify the straight lines
- Our edge detection algorithm did not segment edges into lines, all it did is tell us which pixels were on a line
- We need to fit models (e.g. line models) to the edge maps to find lines in the image

Identifying lines in images

- Possible approaches:
  - Brute force: traverse the entire space of all possible lines in the image, check if they exist in the image
  - Hough transform: vote for lines to which edges might belong
  - Fitting: guess a possible line in the image, then refine it through iterations

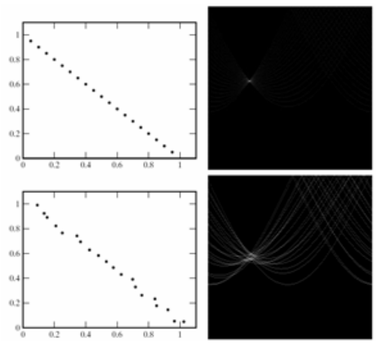
The Hough transform

- You can express a line as a pair (r,θ)
- You can plot these lines as points in *line space*
- A point in cartesian space maps to a curve in line space
- The intersection of two different point curves represents the line segment between the two points

The Hough transform

- We can use the Hough transform to identify line segments in image
- Basic idea
  - for every point on an edge, draw out the line space curve for that point
  - Do this for all points on edges
  - Discretize line segment with a convenient grid and count how many times each grid point (e.g. an individual line) appears
  - The lines that appear more often are the lines in the image

Examples



Reading

- Forsyth, Ch 14 and 15
- "Normalized Cuts and Image Segmentation" by Shi and Malik