CSSE463: Image Recognition Day 33

- This week
  - Today:
    - *Graph-theoretic* approach to segmentation
  - Tuesday: Project workday
  - Thursday: Test 2
  - Friday: status reports due
- Next week:
  - Monday – Friday: presentations
- Questions?

Segmentation

- Goal: we want the pixels in each region to be similar to each other, but different than ones in other regions.
- Measures of *affinity*:
  - Color, distance, edges, texture, etc.

Graph-theoretic segmentation

- J Shi and J Malik, Normalized Cuts and Image Segmentation, IEEE TPAMI, 22(8), Aug 2000.
  - Posted in Angel > Papers in Angel
- Much of the next set of slides is from Forsyth & Ponce, section 14.5.

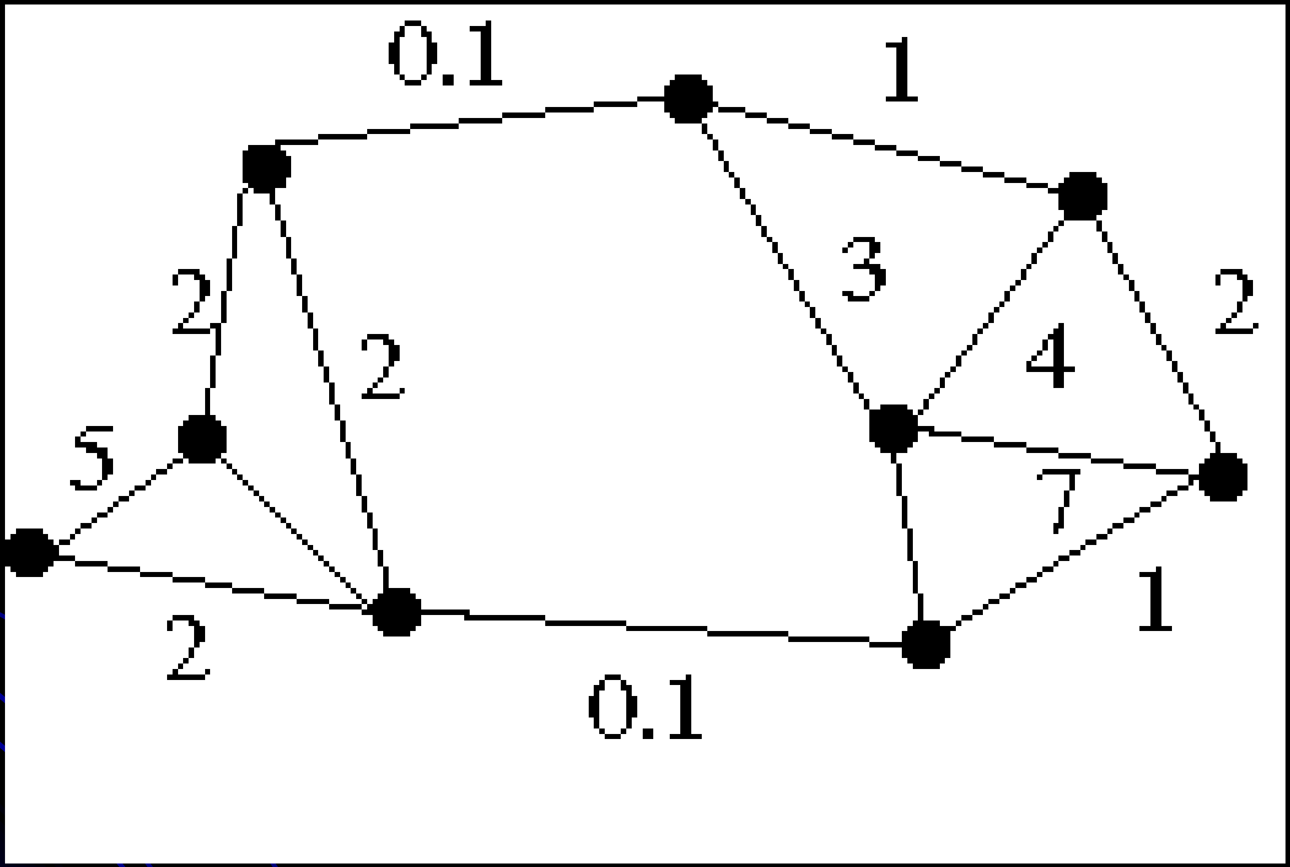
- But first, what's a graph? (on board)
  - Undirected vs. directed
  - Weight matrices

What's graph-based segmentation?

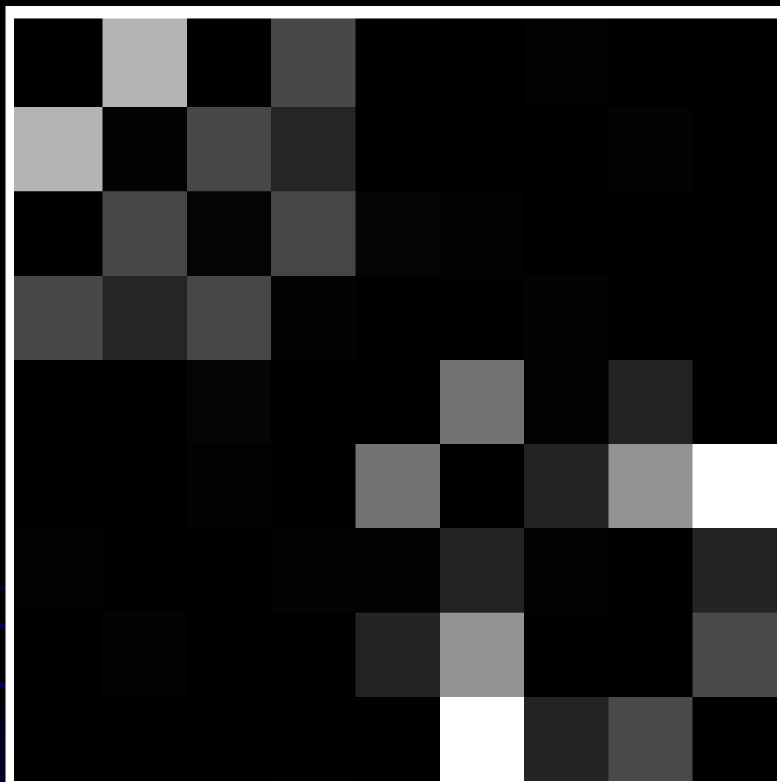
- From images to graphs:
  - Each pixel is a vertex
  - Each edge represents an affinity between pixels
    - Color, texture, distance, edges, etc.
- Cut the graph into two subgraphs (regions) with high affinity within each subgraph and low affinity across subgraphs
- Recurse
- How?
  - An exact solution is NP-complete.

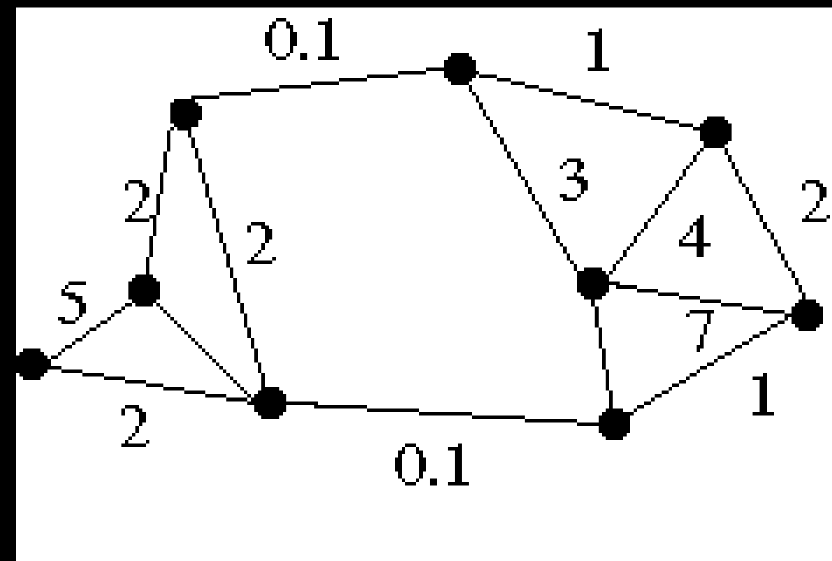
Q1

Split such that large weights (affinity) within cluster, low weights between clusters



Weight matrix





Brighter = Larger weights

Ordered such that each cluster (blocks) is on the diagonal.
Cutting the graph gives two separate blocks

Measuring Affinity

- Recall: goal is for similar pixels to have higher affinity

Intensity

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

Distance

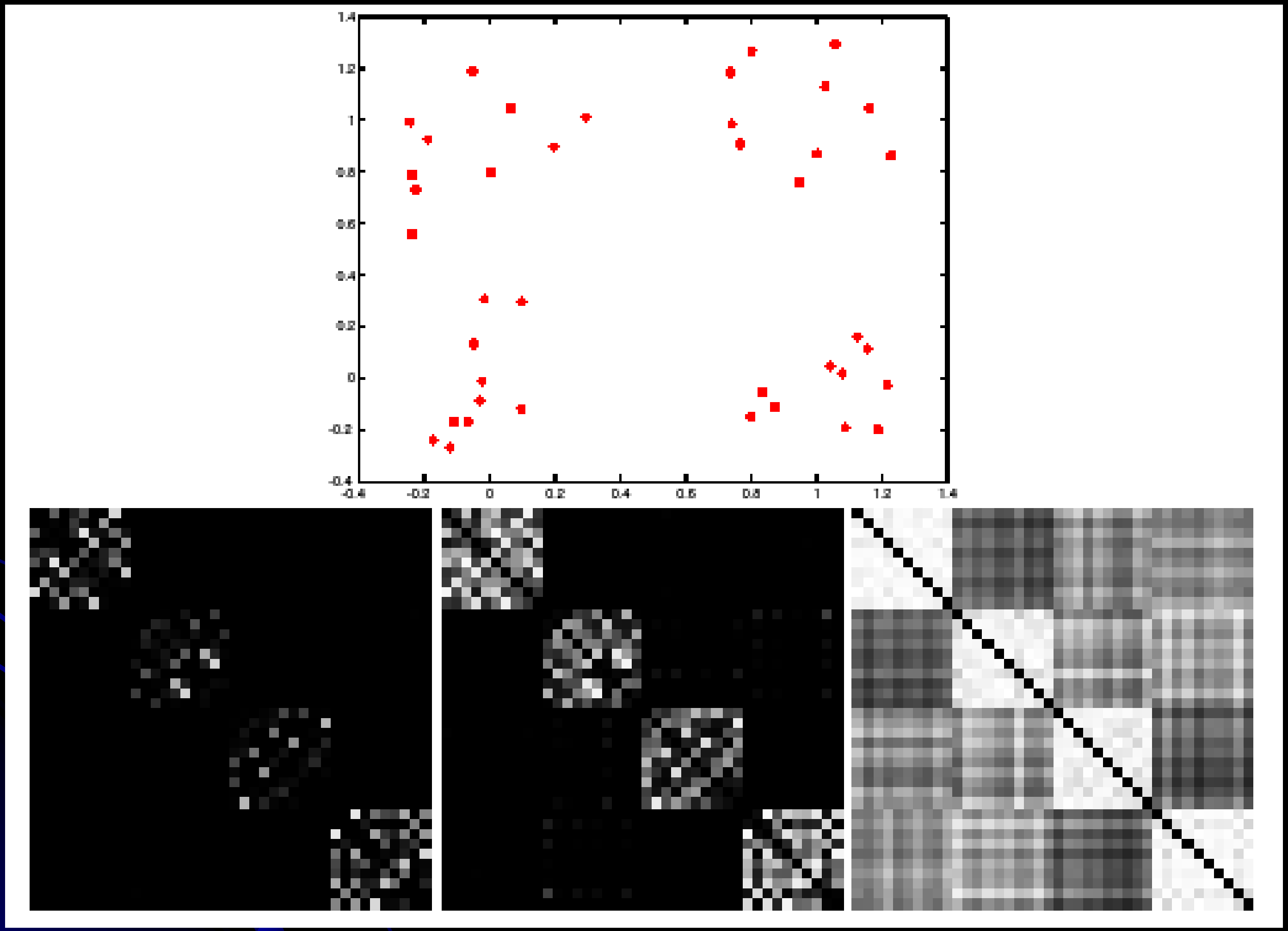
$$aff(x,y) = \exp\left\{-\left(\frac{1}{2\sigma_d^2}\right)\left(\left\| x - y \right\|^2\right)\right\}$$

Color

What if c(x) == c(y)?
What if they are very different?

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

Scale affects affinity



Affinity by distance with σ_d = 0.1, σ_d=0.2, σ_d=1

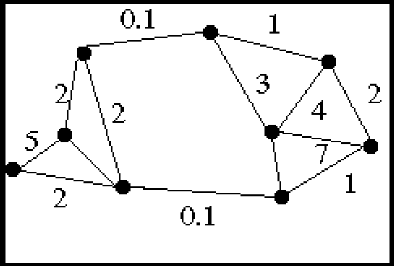
How to segment

- There are some techniques that ignore across-cluster distance (and only use within-cluster affinity).
  - We'll ignore these

- Want to minimize the cut between clusters A and B:

$$cut(A,B) = \sum_{\substack{u \in A \\ v \in B}} w(u,v)$$

- In this graph, cut(left, right) = ___



- This creates two segments like discussed above. However, it favors small regions

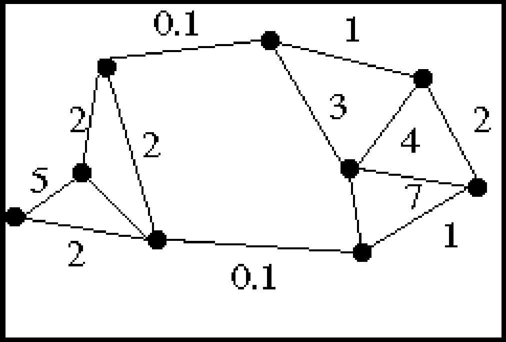
The normalized cuts (*ncuts*) approach minimizes affinities between clusters and maximizes affinities within clusters

- Weight the cut by the total association with the whole graph

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

$$assoc(A,V) = \sum_{\substack{u \in A \\ v \in V}} w(u,v)$$

- Example.



This summation double-counts the affinities within a cluster, which is OK.

Ncut(left, right) = 0.2 / [2(0.1 + 0.1 + 2(2+2+5+2))] +
0.2 / [2(0.1 + 0.1+ 2(1+2+3+4+7+1))]

How do we find min Ncut()?

- Let W be the matrix of weights
- Let d be the total affinity of pixel d with the rest of the image (row sums of W), and D be a matrix with the d(i) on the diagonal
- Let x be a vector whose elements are 1 if item is in A, -1 if it's in B,
- Let y = f(x,d) defined in the paper.
- 1 is the vector with all ones.

$$D(i,i) = \sum_j w(i,j)$$

- Criterion becomes

$$\min_{\mathrm{y}} \left(\frac{y^T (D - W) y}{y^T D y} \right)$$

subject to the constraint

$$y^T D 1 = 0$$

See p. 890, Shi and Malik for details

Normalized cuts

- To do this, we can solve the generalized eigenvalue problem

$$\max_y \left(y^T (D - W) y \right) \text{ subject to } \left(y^T D y = 1 \right)$$

- which gives

$$(D - W) y = \lambda D y$$

- The authors showed that the solution, y, corresponding to the second smallest eigenvalue solves this problem (the smallest is a trivial solution).

- When y is discrete, this is np-hard.
- Solution: assume y is continuous, solve, then threshold to give the clusters (approximation)

- Recursively split clusters until we pass a threshold on the cut value.

Results with color + texture

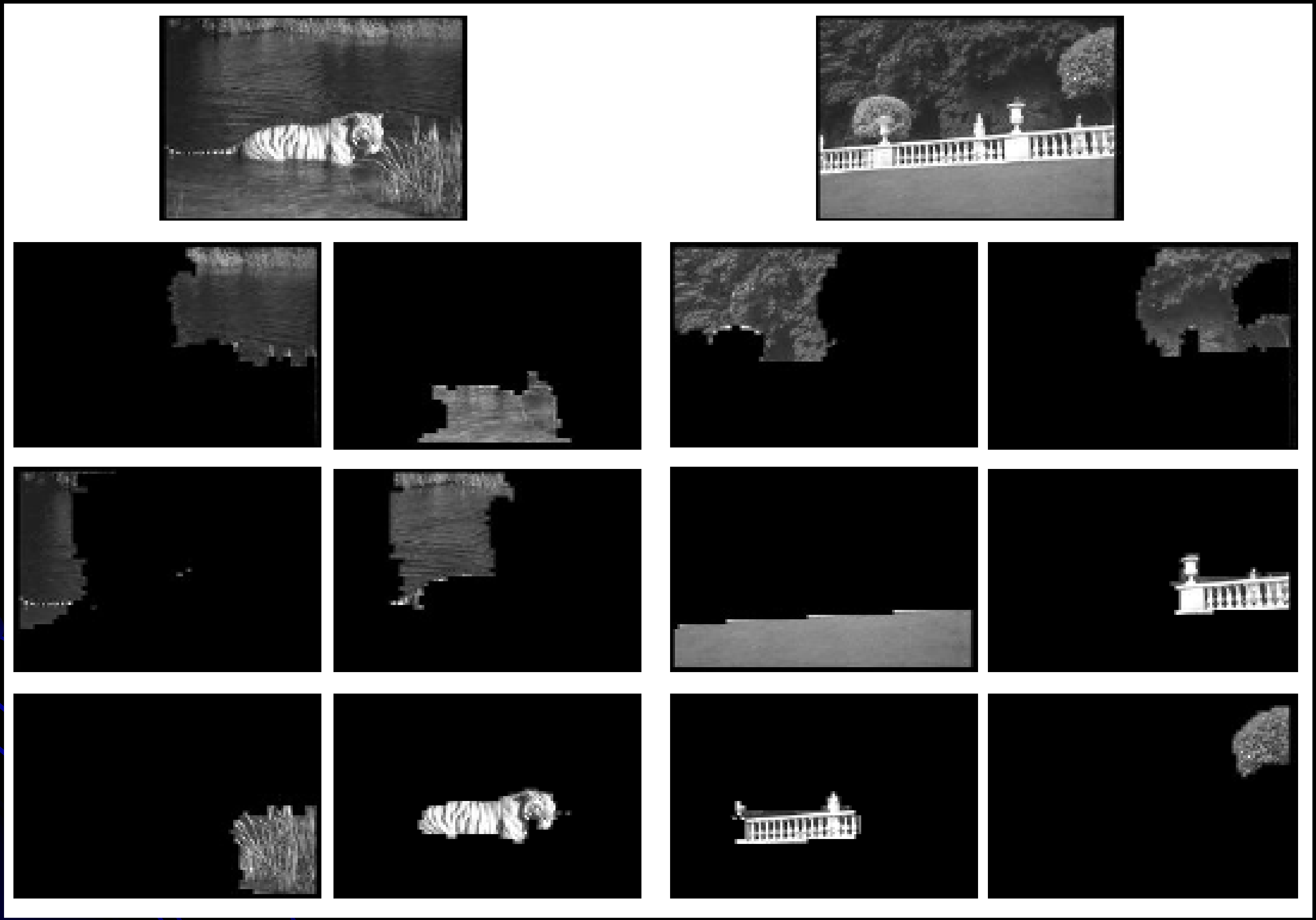


Figure from "Image and video segmentation: the normalised cut framework", by Shi and Malik, copyright IEEE, 1998

Results with motion

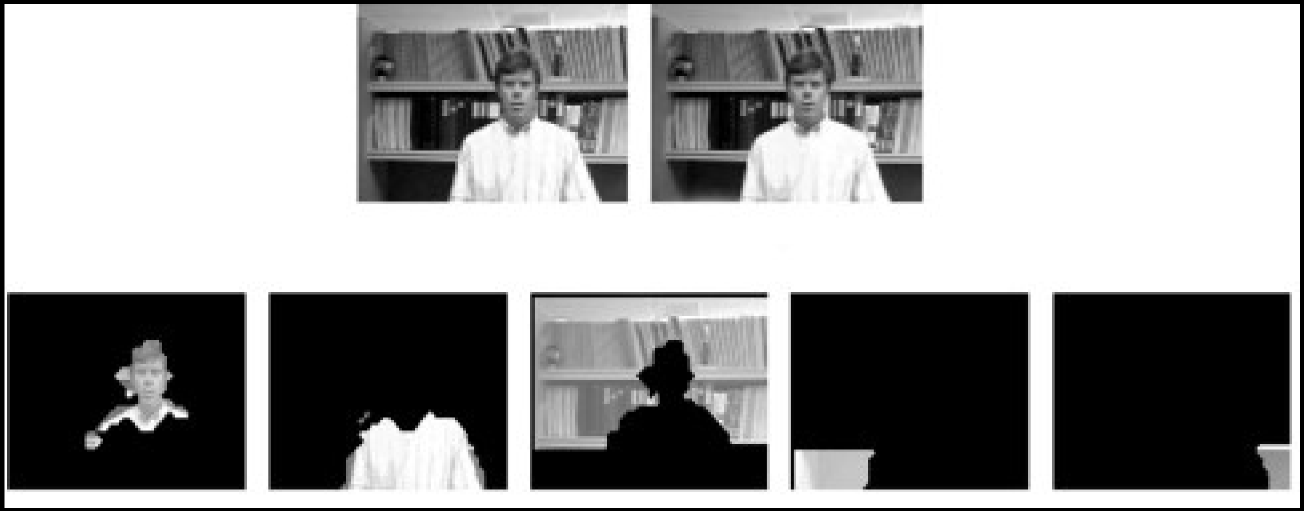


Figure from "Normalized cuts and image segmentation," Shi and Malik, copyright IEEE, 2000

This uses distances too