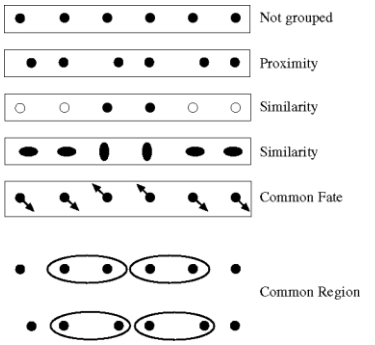
Graph-based Segmentation

Computer Vision
CS 543 / ECE 549
University of Illinois

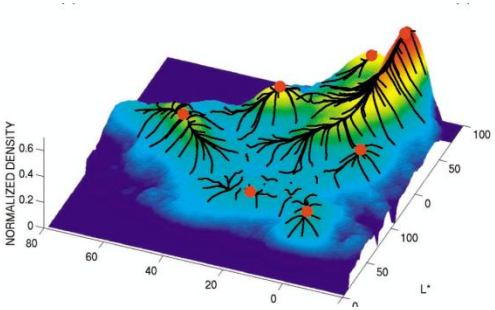
Derek Hoiem

Last class

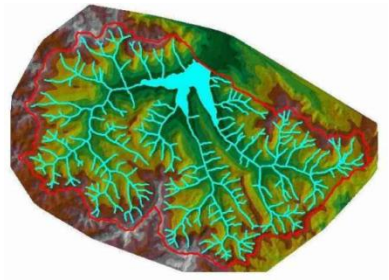
- Gestalt cues and principles of organization



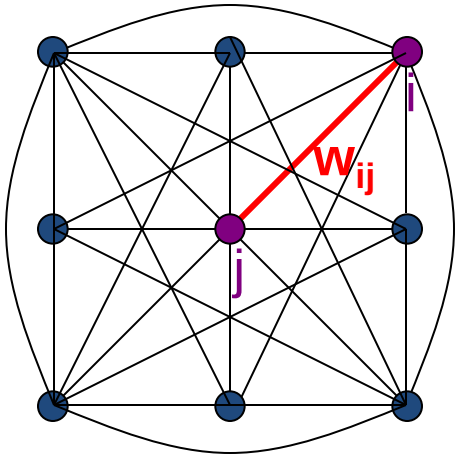
- Mean-shift segmentation
  - Good general-purpose segmentation method
  - Generally useful clustering, tracking technique

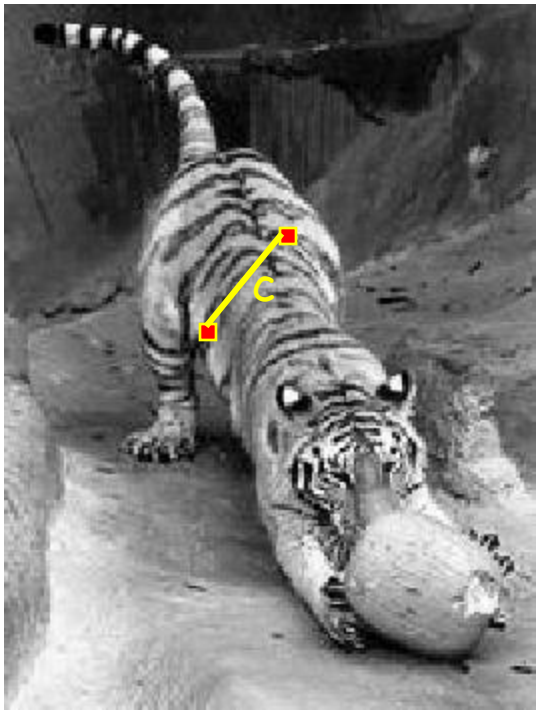


- Watershed segmentation
  - Good for hierarchical segmentation
  - Use in combination with boundary prediction



Images as graphs

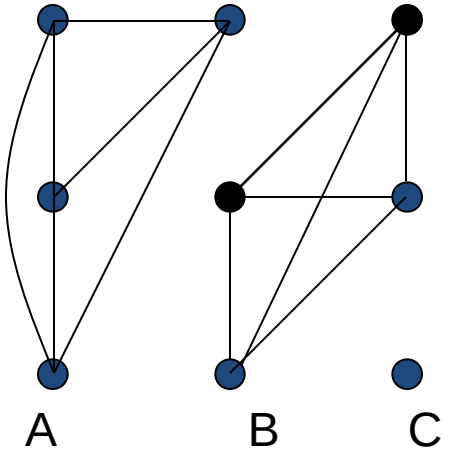




- *Fully-connected* graph
  - node for every pixel
  - link between *every* pair of pixels, **p**,**q**
  - similarity $\mathbf{w_{ij}}$ for each link

Source: Seitz

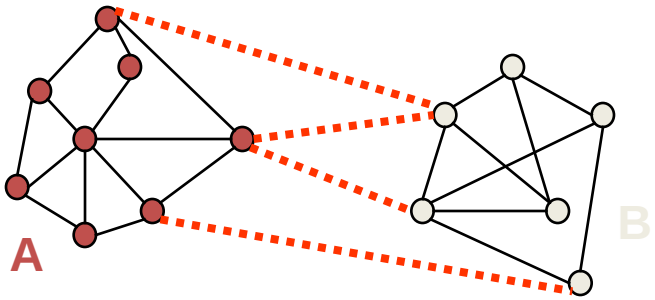
Segmentation by Graph Cuts



- Break Graph into Segments
  - Delete links that cross between segments
  - Easiest to break links that have low cost (low similarity)
    - similar pixels should be in the same segments
    - dissimilar pixels should be in different segments

Source: Seitz

Cuts in a graph



- Link Cut
  - set of links whose removal makes a graph disconnected
  - cost of a cut:

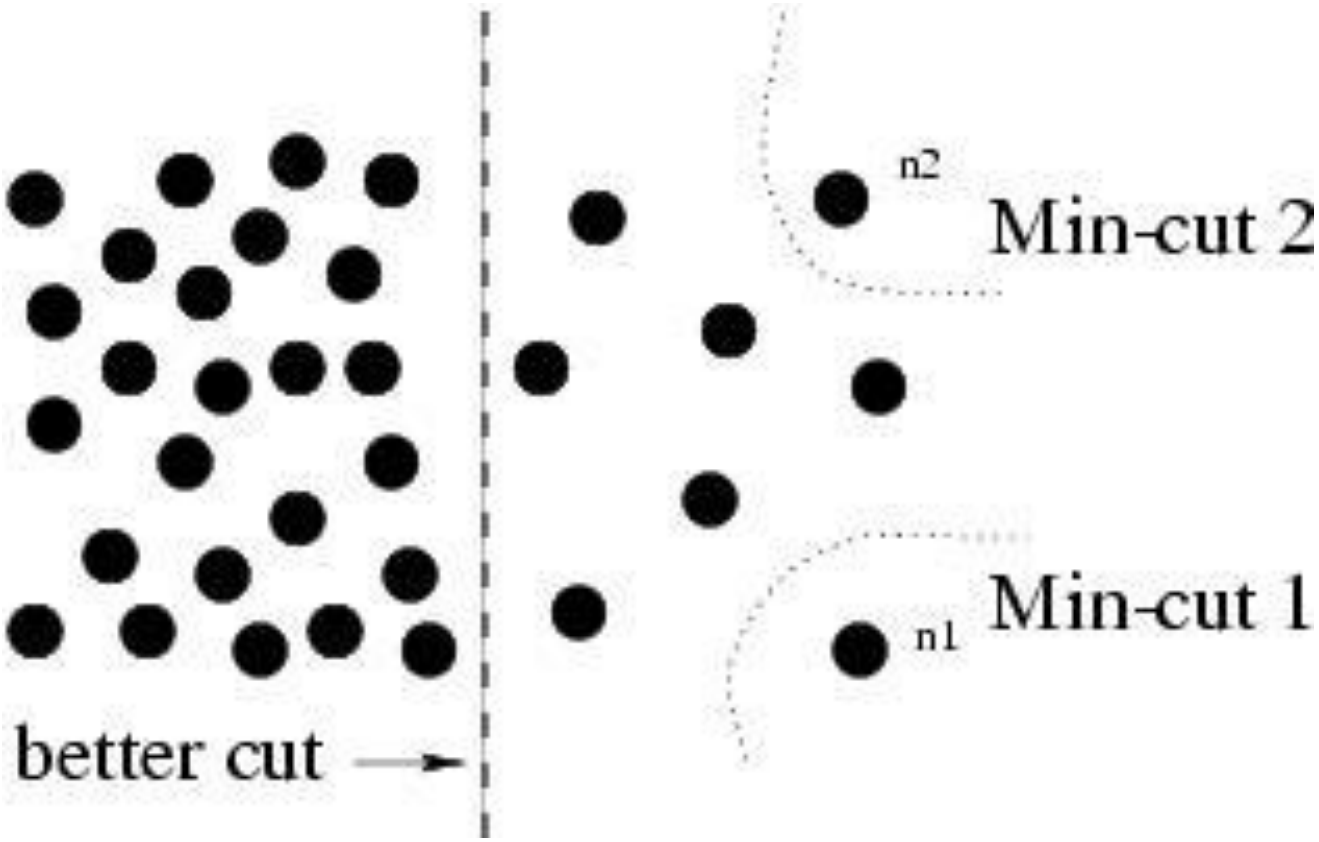
$$cut(A, B) = \sum_{p \in A, q \in B} c_{p,q}$$

One idea: Find minimum cut

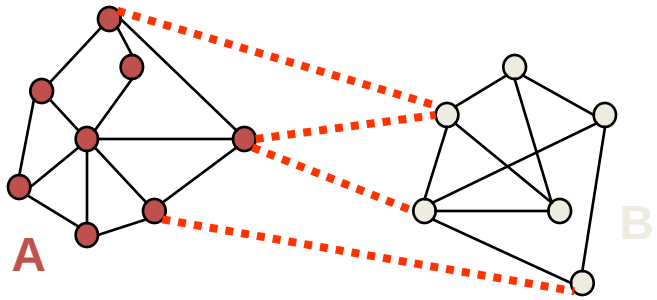
- gives you a segmentation
- fast algorithms exist for doing this

Source: Seitz

But min cut is not always the best cut...



Cuts in a graph



Normalized Cut

- a cut penalizes large segments
- fix by normalizing for size of segments

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

- volume(A) = sum of costs of all edges that touch A

Source: Seitz

Recursive normalized cuts

1. Given an image or image sequence, set up a weighted graph: G=(V, E)
   - Vertex for each pixel
   - Edge weight for nearby pairs of pixels

2. Solve for eigenvectors with the smallest eigenvalues: $(D - W)y = \lambda Dy$
   - Use the eigenvector with the second smallest eigenvalue to bipartition the graph
   - Note: this is an approximation

4. Recursively repartition the segmented parts if necessary

Details: http://www.cs.berkeley.edu/~malik/papers/SM-n**cut**.pdf

Normalized cuts results

Normalized cuts: Pro and con

- Pros
  - Generic framework, can be used with many different features and affinity formulations
  - Provides regular segments
- Cons
  - Need to chose number of segments
  - High storage requirement and time complexity
  - Bias towards partitioning into equal segments
- Usage
  - Use for oversegmentation when you want regular segments