Image Segmentation

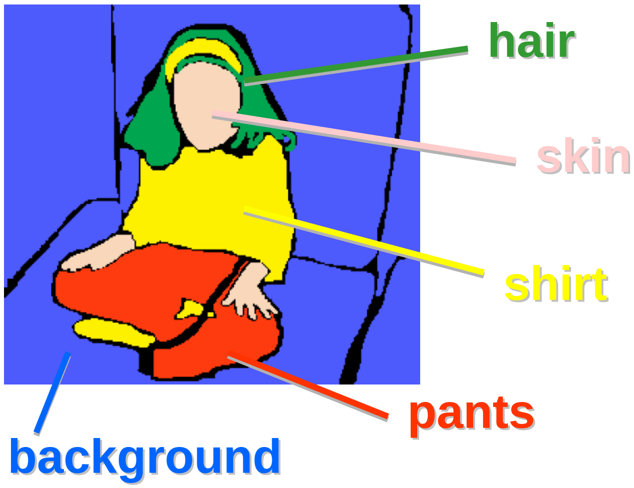
Problem Statement

Unsupervised Segmentation

Spectral Segmentation

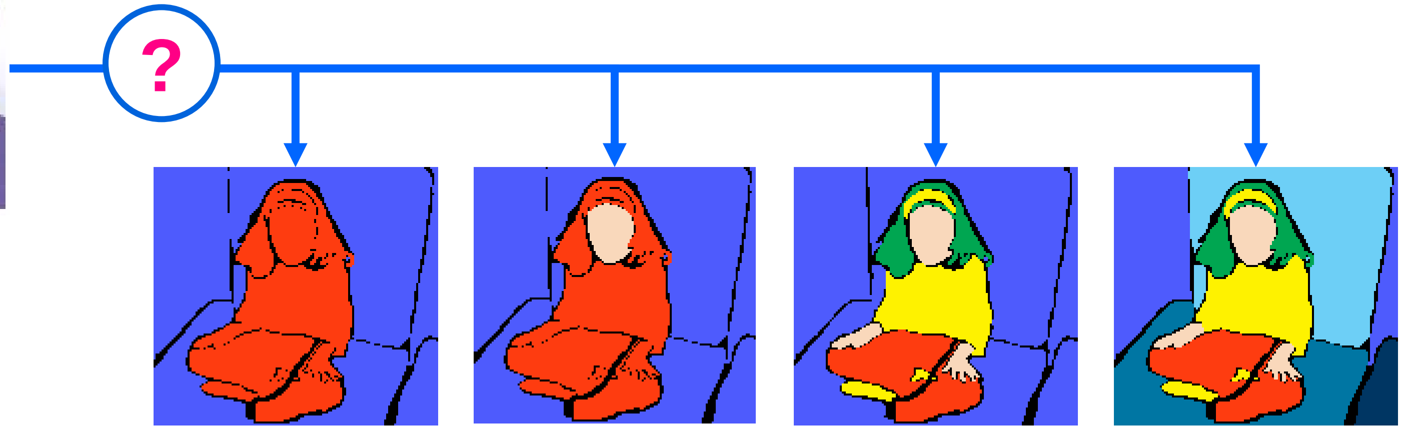
Image Segmentation

- Partitioning of an image into *coherent regions*
- Grouping image pixels into *coherent regions*



“Correct” Segmentation

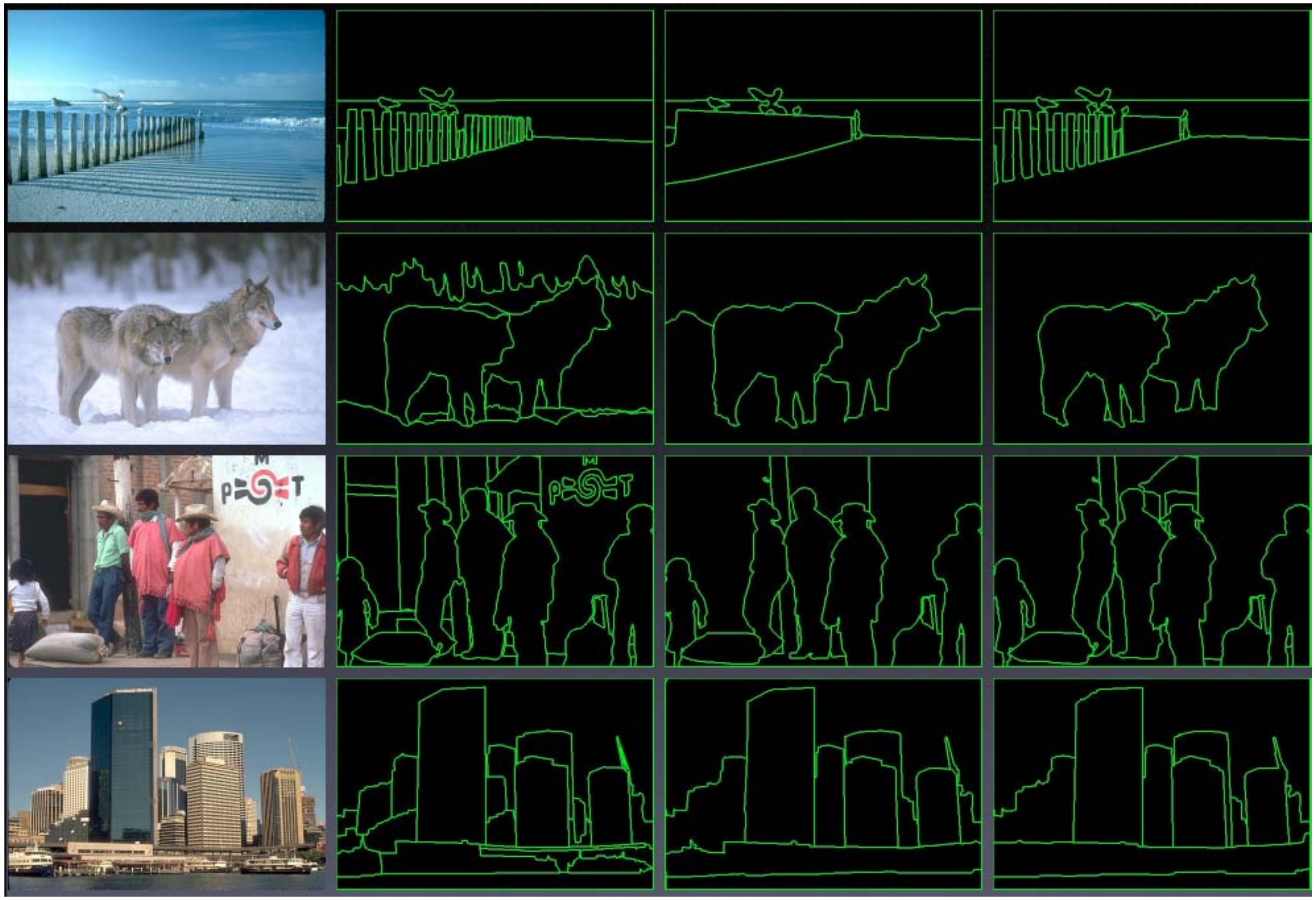
Input Image



Alternative Segmentations

- What is the “correct” partitioning/grouping?
  - No single correct answer
  - Interpretation depends on prior world knowledge
  - World knowledge is difficult to represent

“Correct” Segmentation



“Correct” Segmentation

- What is the “correct” partitioning/grouping?
  - No single correct answer
  - Interpretation depends on prior world knowledge
  - World knowledge is difficult to represent
- Typical assumptions (low level vision):
  - Brightness/color coherence
  - Texture coherence
  - Motion coherence

Hierarchical Segmentation

- Image partitioning is inherently hierarchical
- Segmentation should construct a tree structure, rather than a “flat” structure
- Low-level coherence is appropriate for lower hierarchy levels
- Mid- and high-level knowledge may be used higher in the hierarchy:
  - symmetries
  - object models, etc.

Bottom-up vs. Top-down

- Bottom-up: group similar pixels together
- Top-down: split regions

Normalized Cuts

- Use graph partitioning to perform image segmentation.

- Clearly defined criterion to optimize
- Efficient algorithms for approximating the optimum
- Formulation lends itself naturally to hierarchical partitioning

Normalized Cuts

- Represent an image as a weighted undirected graph, G = (V, E).
- Each image pixel is a vertex, $v \in V$.
- Nearby pixels are connected by edges.
- The weight of each edge w(i,j) is a function of the similarity between its endpoint pixels, i and j, for example:

$$w(i,j) = e^{-\frac{\|F(i)-F(j)\|^2}{\sigma^2}}$$

Normalized Cuts

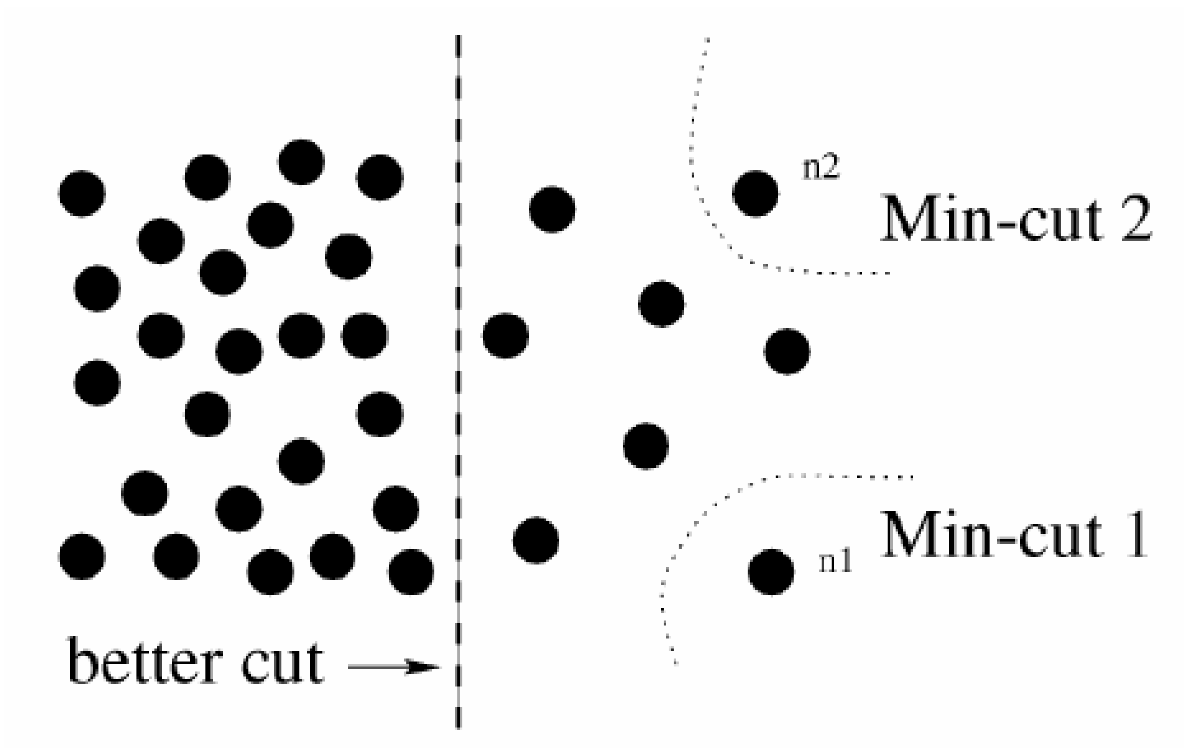
- Goal: find an optimal partitioning of V into A and B, such that $A \cup B = V$, $A \cap B = \varnothing$

- What is an optimal partitioning?
- Idea 1: find the partitioning that minimizes the (cost of the) cut:

$$cut(A,B) = \sum_{i \in A, j \in B} w(i,j)$$

Problem

- Minimal cut partitioning favors cutting off small isolated groups of vertices:



Idea

- Instead of looking at the total sum of edges across the cut, consider the fraction of all edge connections:

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

- where

$$assoc(A,V) = \sum_{i \in A, j \in V} w(i,j)$$

Idea

- Another heuristic: maximize the fraction of the connections inside each group:

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

- Surprise!

$$Ncut(A,B) = 2 - Nassoc(A,B)$$

- Kill two birds with one stone!

And Now, The Bad News...

- Minimizing the normalized cut is NP-complete (even when the graph is a regular grid).

- However, we can solve the problem approximately!

Matrix Notation

- Affinity matrix W: $$W_{ij} = w(i,j)$$
- A diagonal matrix D: $$D_{ii} = \sum_{j \in V} w(i,j)$$
- Indicator vector for A $\in$ V:

$$x_A(i) = \begin{cases} 1 & i \in A \\ -1 & i \in V - A \end{cases}$$

Back to Normalized Cuts

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

$$Ncut(x) = \frac{\sum_{(x_i>0,x_j<0)} -W_{ij}x_i x_j}{\sum_{x_i>0} D_{ii}} + \frac{\sum_{(x_i<0,x_j>0)} -W_{ij}x_i x_j}{\sum_{x_i<0} D_{ii}}$$

$$Ncut(x) = \frac{(1+x)^T(D-W)(1+x)}{k\,1^T D 1} + \frac{(1-x)^T(D-W)(1-x)}{(1-k)1^T D 1}$$

Back to Normalized Cuts

- After some rather ugly algebra, we get:

$$\min_x Ncut(x) = \min_y \frac{y^{\mathrm{T}}(D-W)y}{y^{\mathrm{T}}Dy}$$

- with the condition

$$y(i) \in \{1, -b\} \text{ and } y^{\mathrm{T}} D \mathbf{1} = 0$$

Real-Valued Normalized Cuts

- Allow y to take on real values (rather than requiring $y \in \{1,-b\}$, and solve the generalized eigenvalue problem:

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

- The second condition on y ($y^T D1 = 0$) will be satisfied automatically!

Proof

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

$$y = D^{-\frac{1}{2}} z$$

$$(D - W) D^{-\frac{1}{2}} z = \lambda D^{\frac{1}{2}} z$$

$$D^{-\frac{1}{2}} (D - W) D^{-\frac{1}{2}} z = \lambda z$$

$$z_1^{\mathrm{T}} z_0 = 0$$

$$y_1^{\mathrm{T}} D y_0 = y_1^{\mathrm{T}} D \mathbf{1} = 0$$

Rayleigh Quotient

- Let A be a real symmetric matrix. Under the constraint that x is orthogonal to the j-1 smallest eigenvectors $x_1,\ldots,x_{j-1}$, the Rayleigh quotient

$$\frac{x^{\mathrm{T}} A x}{x^{\mathrm{T}} x}$$

is minimized by the next smallest eigenvector x_j, and its minimum value is the j-th eigenvalue λ_j.

Finally...

- We have that:

$$z_1 = \arg \min_{z^T z_0 = 0} \frac{z^T D^{-\frac{1}{2}} (D - W) D^{-\frac{1}{2}} z}{z^T z}$$

- And, consequently:

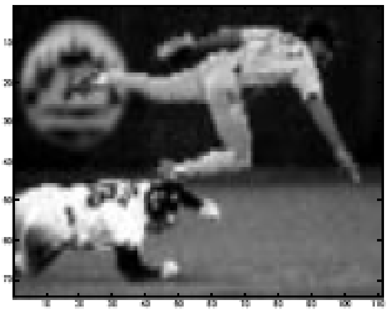
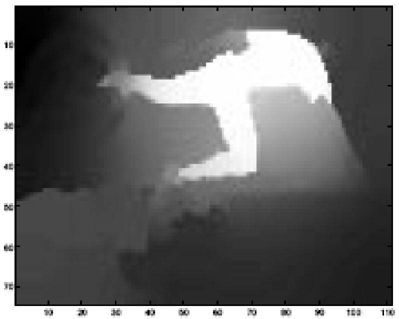
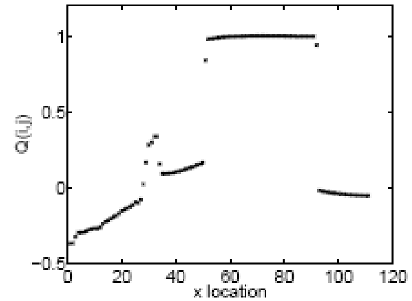
$$y_1 = \arg \min_{y^T D \mathbf{1} = 0} \frac{y^T (D - W) y}{y^T D y}$$

- The second smallest generalized eigenvector is the solution for our real-valued normalized cut problem!

N-cuts Segmentation

- Given an image, set up the weighted graph $G = (V,E)$.
- Solve $(D - W)x = \lambda Dx$ for the smallest eigenvectors.
- Use the second smallest eigenvector to bi-partition the graph.
- Recurse on each of the resulting parts, if necessary.

Partitioning

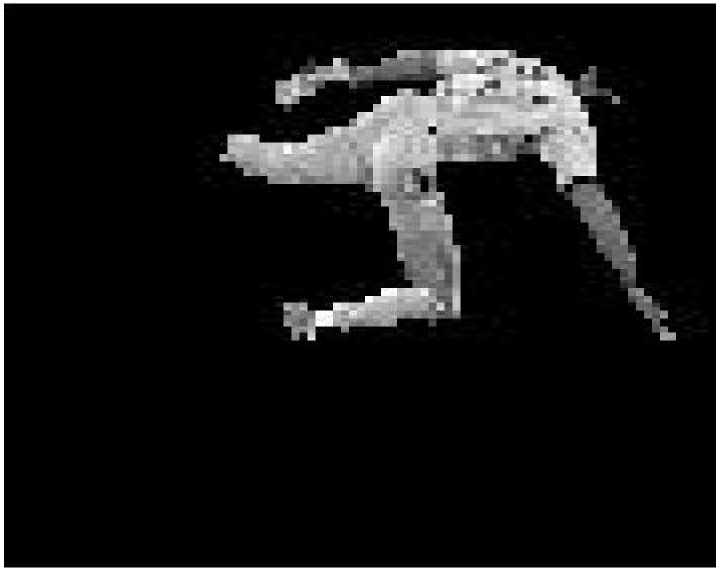




- The second smallest eigenvector has continuous values.
- Choose a splitting threshold to find a discrete vector
  - use the median
  - find a splitting point that minimizes Ncut(A,B)

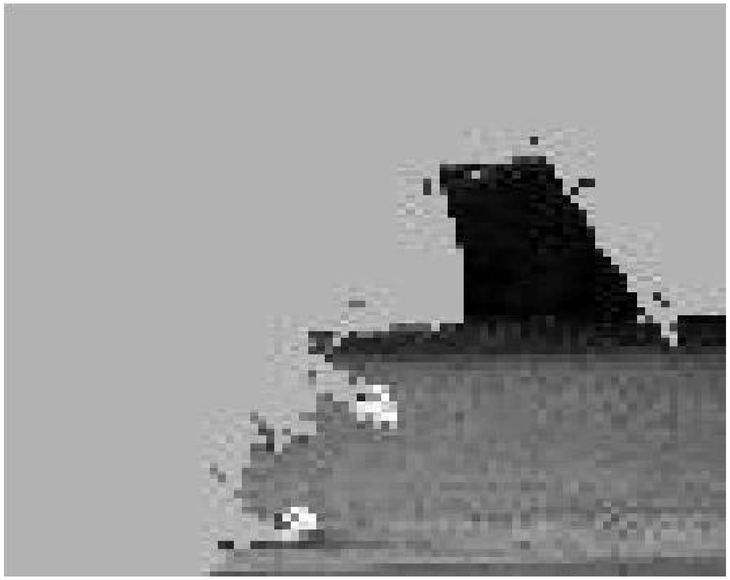
Example

The first partition

The second partition

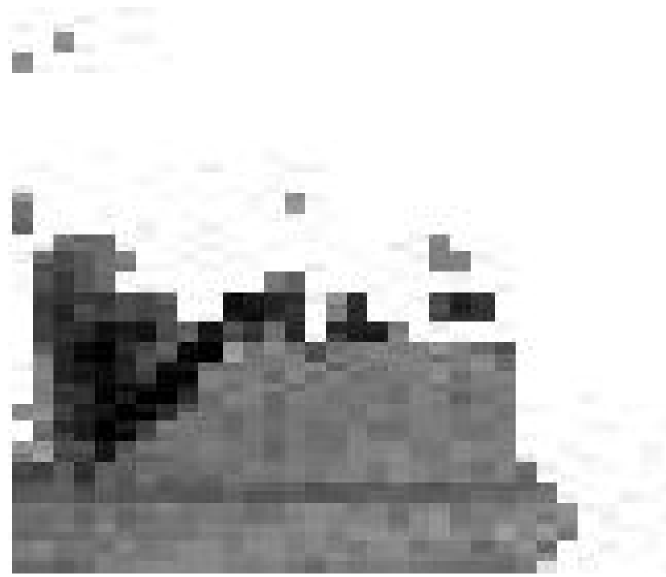


The third partition



The fourth partition

The fifth partition



The sixth partition