MRFs and Segmentation with Graph Cuts

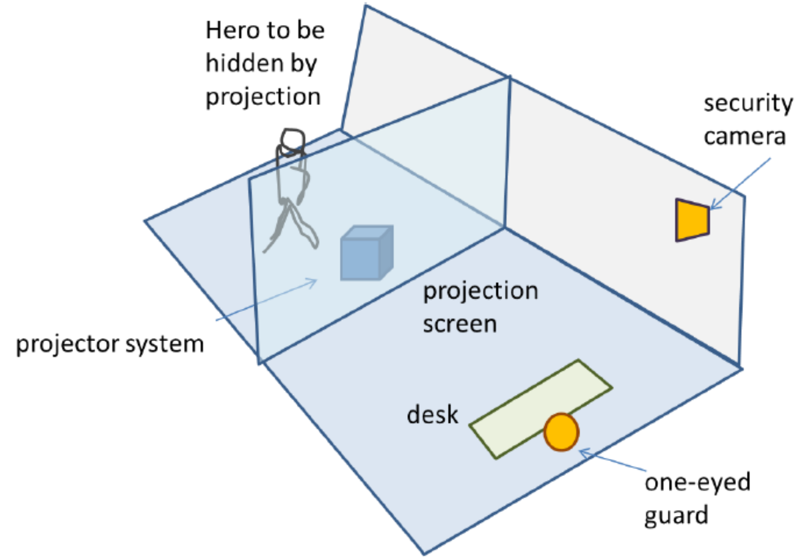
Computer Vision

CS 543 / ECE 549

University of Illinois

Derek Hoiem

HW 3



- Mean 92
- Median 98
- Common mistakes
  - Mission Possible: don't need camera rotation matrix
  - Epipolar geometry: general problems with RANSAC and/or estimating F

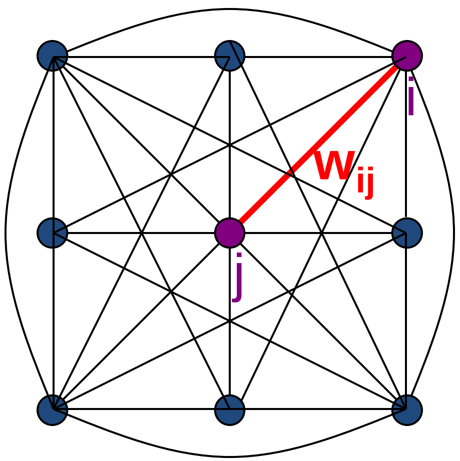
Final projects

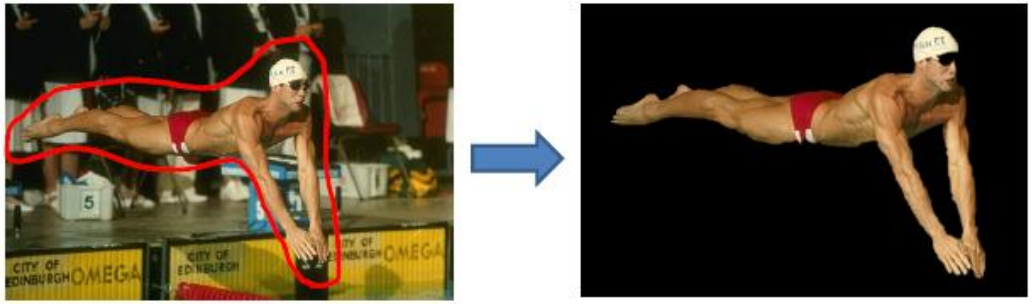
- Make appointments to see Ruiqi or me
  https://docs.google.com/spreadsheet/ccc?key=0AiuVuPXkVyqmdF9LZWNVbVFtYXRmTldWcjFhWWpZLWc&hl=en_US#gid=0

- No late penalty for not meeting us this week, but must get proposal submitted by Thursday
  - If you're not sure, make your best guess; can change later

Today's class

- Review of EM
- MRFs
- Segmentation with Graph Cuts





EM Algorithm: Recap

1. E-step: compute

$$E_{z|x,\theta^{(t)}}\left[\log\left(p(\mathbf{x},\mathbf{z}\mid\theta)\right)\right]=\sum_{\mathbf{z}}\log\left(p(\mathbf{x},\mathbf{z}\mid\theta)\right)p\left(\mathbf{z}\mid\mathbf{x},\theta^{(t)}\right)$$

2. M-step: solve

$$\theta^{(t+1)}=\underset{\theta}{\operatorname{argmax}}\sum_{\mathbf{z}}\log\left(p(\mathbf{x},\mathbf{z}\mid\theta)\right)p\left(\mathbf{z}\mid\mathbf{x},\theta^{(t)}\right)$$

- Determines hidden variable assignments and parameters that maximize likelihood of observed data
- Improves likelihood at each step (reaches local maximum)
- Derivation is tricky but implementation is easy

EM: Mixture of Gaussians

1. Initialize parameters

2. Compute likelihood of hidden variables for current parameters

Given by normal distribution

$$\alpha_{nm} = p(z_n = m \mid x_n, \boldsymbol{\mu}^{(t)}, \boldsymbol{\sigma}^{2(t)}, \boldsymbol{\pi}^{(t)}) = \frac{p(x_n \mid z_n = m, \theta_m)\, p(z_n = m \mid \theta_m)}{\sum_k p(x_n \mid z_n = k, \theta_k)\, p(z_n = k \mid \theta_k)}$$

3. Estimate new parameters for each component, weighted by likelihood

$$\hat{\mu}_m^{(t+1)} = \frac{1}{\sum_n \alpha_{nm}} \sum_n \alpha_{nm} x_n \qquad \hat{\sigma}_m^{2(t+1)} = \frac{1}{\sum_n \alpha_{nm}} \sum_n \alpha_{nm} \left(x_n - \hat{\mu}_m\right)^2 \qquad \hat{\pi}_m^{(t+1)} = \frac{\sum_n \alpha_{nm}}{N}$$

Gaussian Mixture Models: Practical Tips

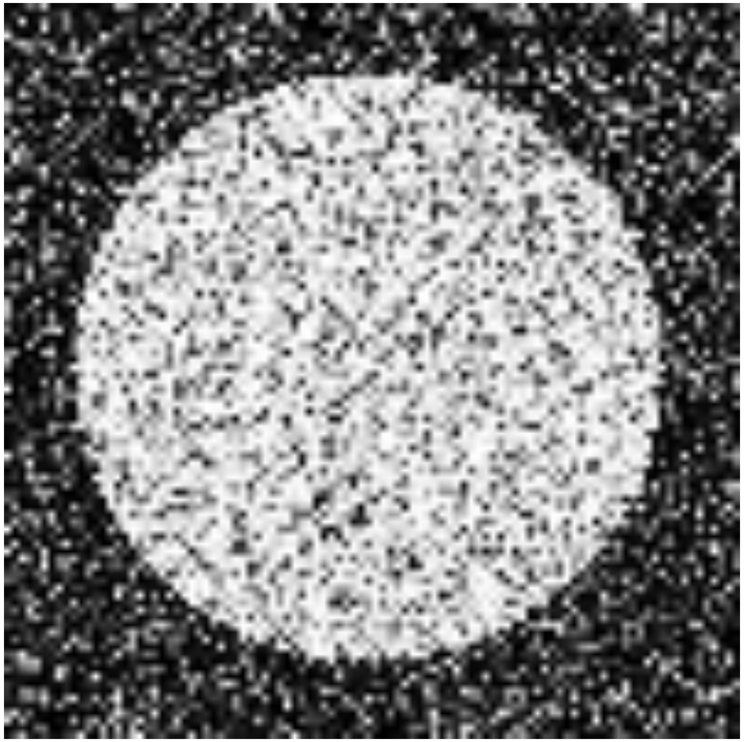
- Design decisions
  - Number of components
    - Select by hand based on knowledge of problem
    - Select using cross-validation or sample data
    - Usually, not too sensitive and safer to use more components
  - Variance
    - "Spherical covariance": dimensions within each component are independent with equal variance (1 parameter but usually too restrictive)
    - "Diagonal covariance": dimensions within each component are not independent with difference variances (N parameters for N-D data)
    - "Full covariance": no assumptions (N*(N+1)/2 parameters); for high N might be expensive to do EM, to evaluate, and may overfit
    - Typically use "Full" if lots of data, few dimensions; Use "Diagonal" otherwise

- Can get stuck in local minima
  - Use multiple restarts
  - Choose solution with greatest data likelihood

EM Demo

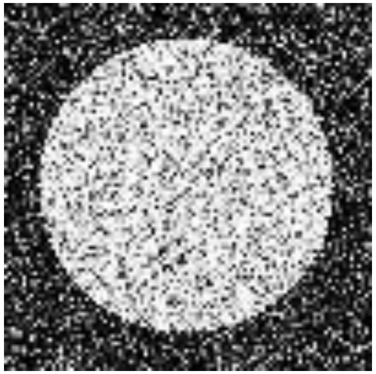
- GMM with images demos

What’s wrong with this prediction?



P(foreground | image)

Solution



P(foreground | image)

Encode dependencies between pixels

Normalizing constant called “partition function”

$$P(\mathbf{y};\theta,data)=\frac{1}{Z}\prod_{i=1..N} f_1(y_i;\theta,data)\prod_{i,j\in edges} f_2(y_i,y_j;\theta,data)$$

Labels to be predicted

Individual predictions

Pairwise predictions

Writing Likelihood as an "Energy"

$$P(\mathbf{y};\theta,data)=\frac{1}{Z}\prod_{i=1..N}p_1(y_i;\theta,data)\prod_{i,j\in edges}p_2(y_i,y_j;\theta,data)$$

-log(.)

$$Energy(\mathbf{y};\theta,data)=\sum_i \psi_1(y_i;\theta,data)+\sum_{i,j\in edges}\psi_2(y_i,y_j;\theta,data)$$

Cost of assignment y_i

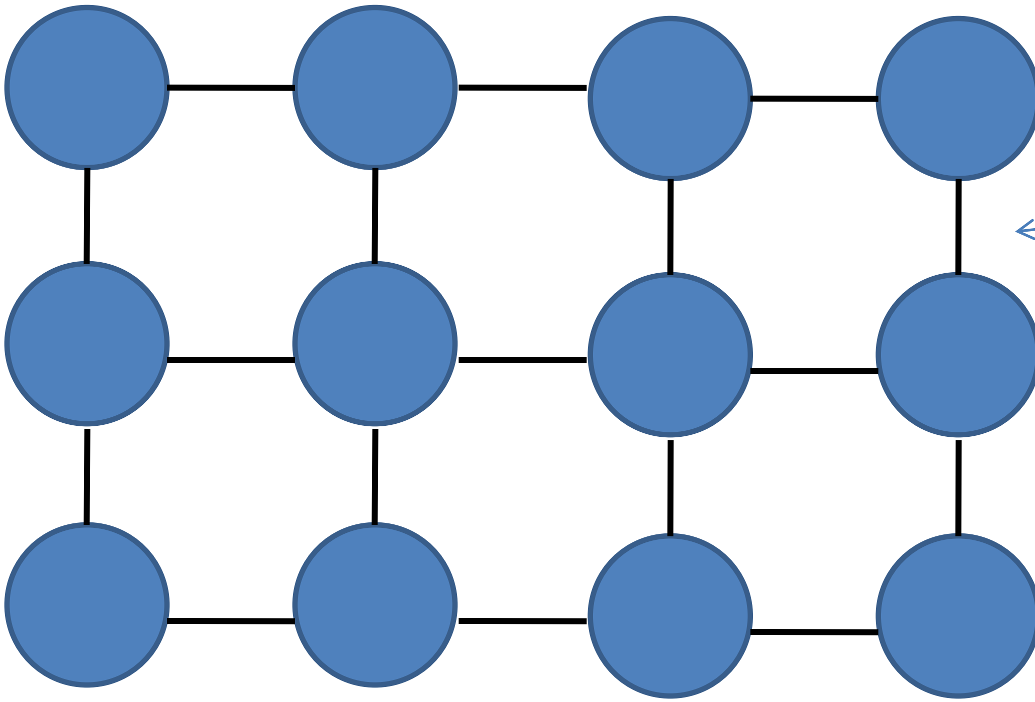
Cost of pairwise assignment y_i ,y_j

Notes on energy-based formulation

$$Energy(\mathbf{y};\theta,data)=\sum_{i}\psi_1(y_i;\theta,data)+\sum_{i,j\in edges}\psi_2(y_i,y_j;\theta,data)$$

- Primarily used when you only care about the most likely solution (not the confidences)
- Can think of it as a general cost function
- Can have larger "cliques" than 2
  - Clique is the set of variables that go into a potential function

Markov Random Fields



Node y_i: pixel label

Edge: constrained pairs

Cost to assign a label to each pixel

Cost to assign a pair of labels to connected pixels

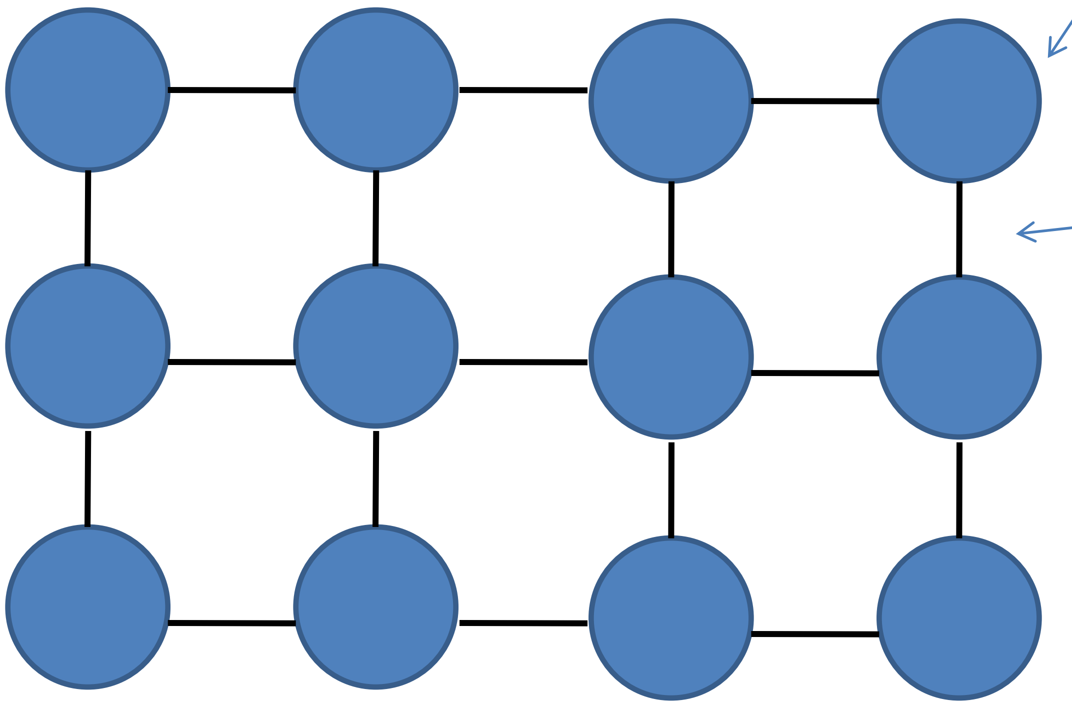
$$Energy(\mathbf{y};\theta,data)=\sum_i \psi_1(y_i;\theta,data)+\sum_{i,j\in edges}\psi_2(y_i,y_j;\theta,data)$$

Markov Random Fields

- Example: “label smoothing” grid

Unary potential

0: $-\log P(y_i = 0 \mid data)$
1: $-\log P(y_i = 1 \mid data)$



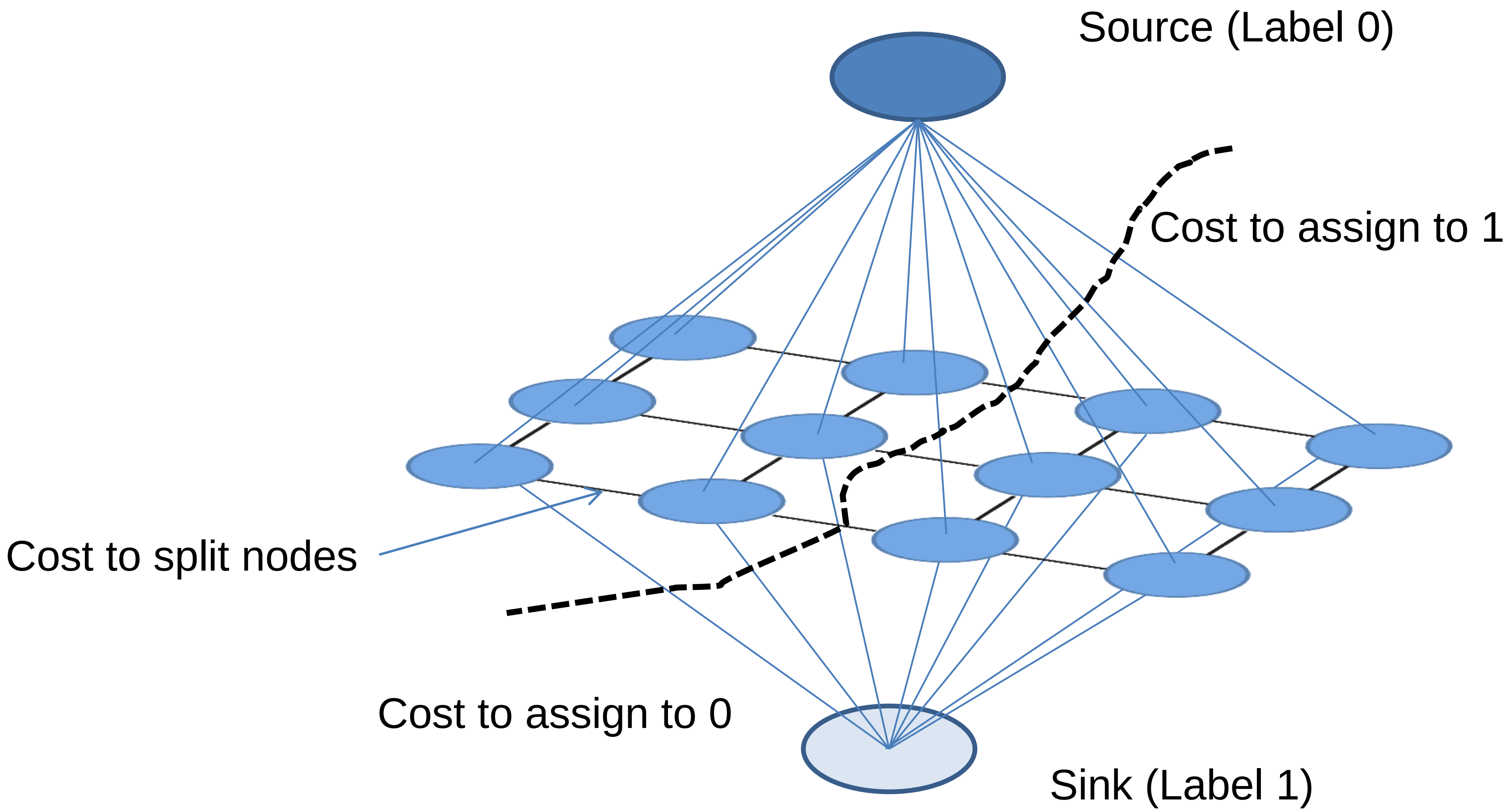
Pairwise Potential

	0	1
0	0	K
1	K	0

K>0

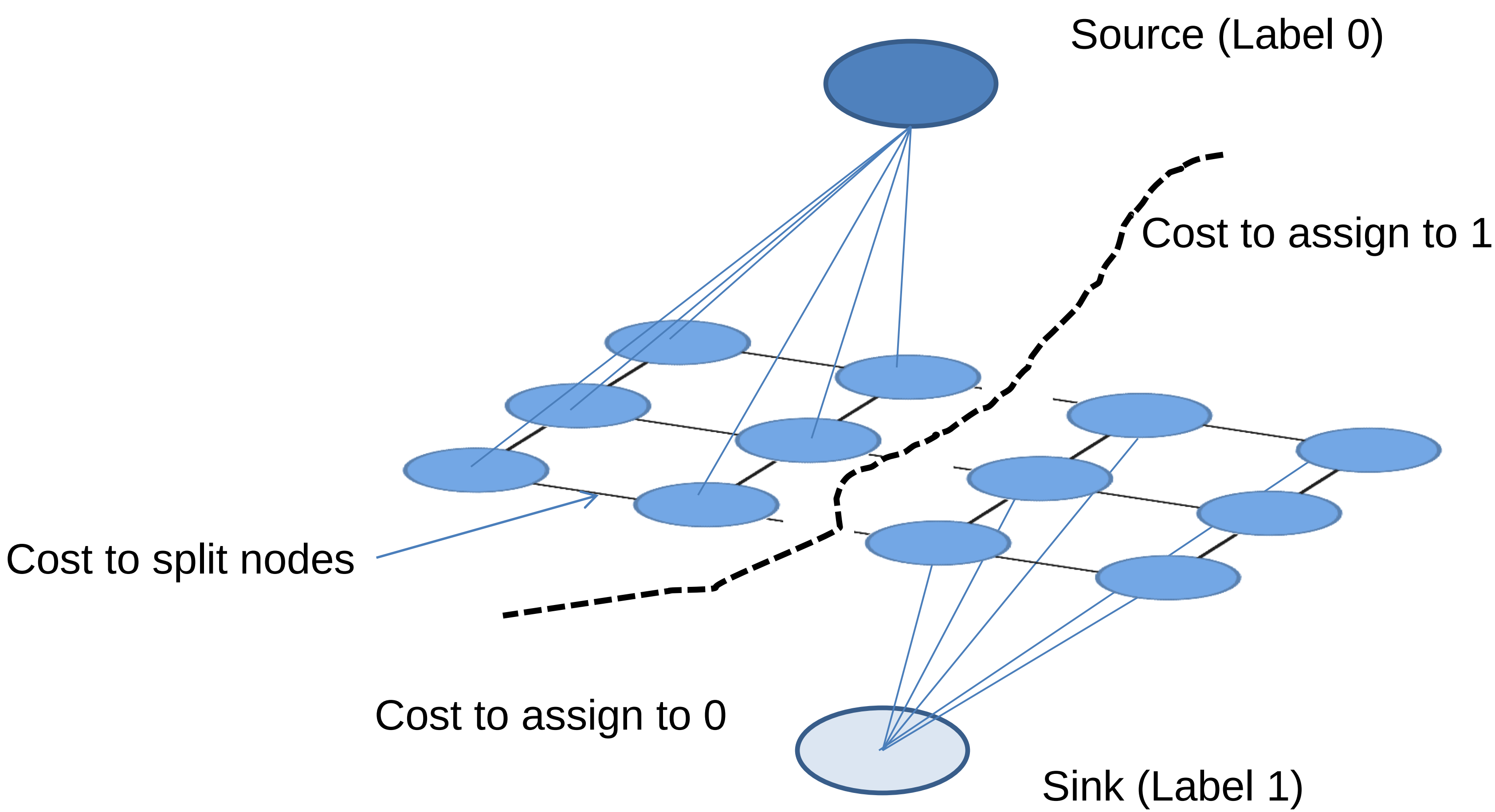
$$Energy(\mathbf{y};\theta,data) = \sum_i \psi_1(y_i;\theta,data) + \sum_{i,j \in edges} \psi_2(y_i, y_j;\theta,data)$$

Solving MRFs with graph cuts



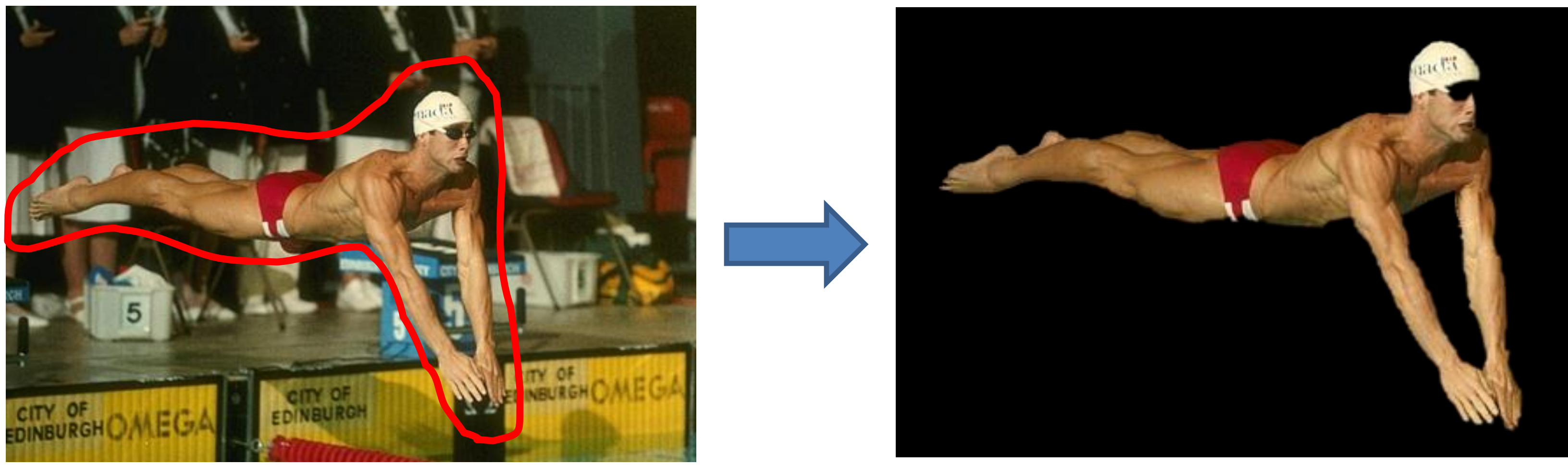
$$Energy(\mathbf{y};\theta,data) = \sum_i \psi_1(y_i;\theta,data) + \sum_{i,j \in edges} \psi_2(y_i,y_j;\theta,data)$$

Solving MRFs with graph cuts



$$Energy(\mathbf{y};\theta,data)=\sum_i \psi_1(y_i;\theta,data)+\sum_{i,j\in edges}\psi_2(y_i,y_j;\theta,data)$$

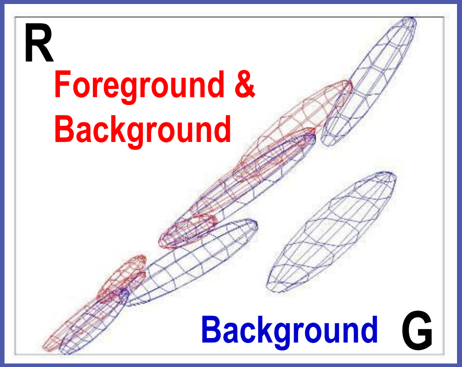
GrabCut segmentation



User provides rough indication of foreground region.

Goal: Automatically provide a pixel-level segmentation.

Colour Model



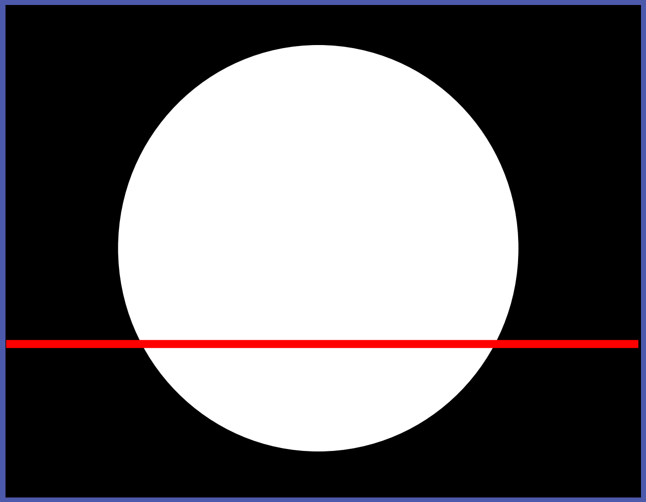
Gaussian Mixture Model (typically 5-8 components)

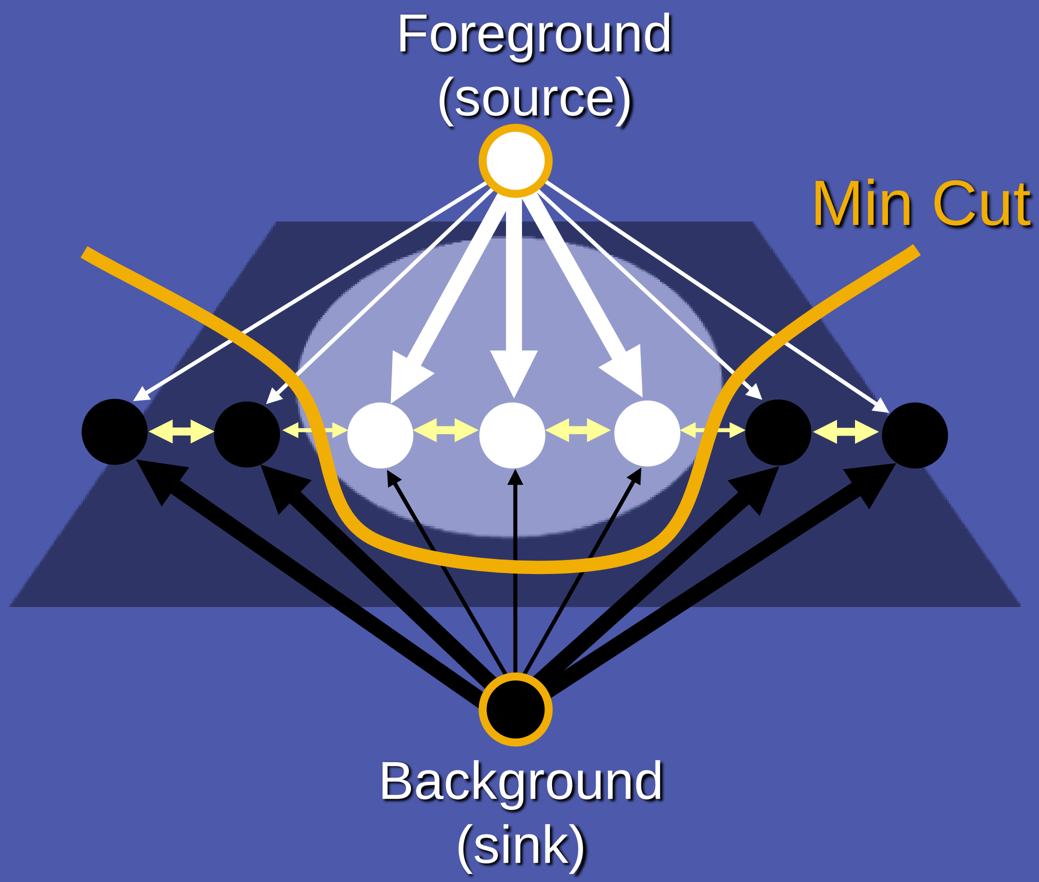
Source: Rother

Graph cuts

Boykov and Jolly (2001)

Image



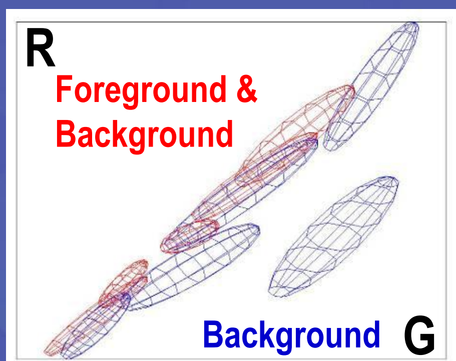


Cut: separating source and sink; Energy: collection of edges

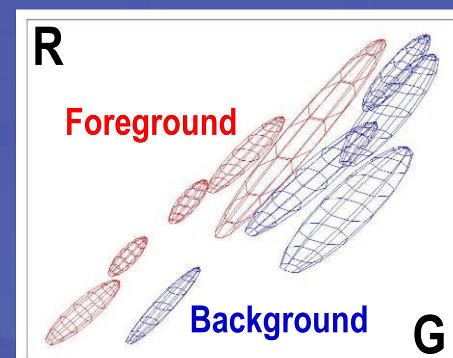
Min Cut: Global minimal enegry in polynomial time

Source: Rother

Colour Model



Iterated
graph cut



Gaussian Mixture Model (typically 5-8 components)

Source: Rother

GrabCut segmentation

1. Define graph
   – usually 4-connected or 8-connected
     • Divide diagonal potentials by sqrt(2)
2. Define unary potentials
   – Color histogram or mixture of Gaussians for background and foreground

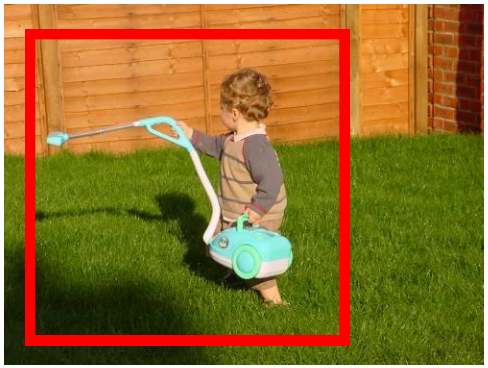
$$unary\_potential(x) = -\log\left(\frac{P(c(x);\theta_{foreground})}{P(c(x);\theta_{background})}\right)$$

3. Define pairwise potentials

$$edge\_potential(x,y) = k_1 + k_2 \exp\left\{\frac{-\left\|c(x)-c(y)\right\|^2}{2\sigma^2}\right\}$$

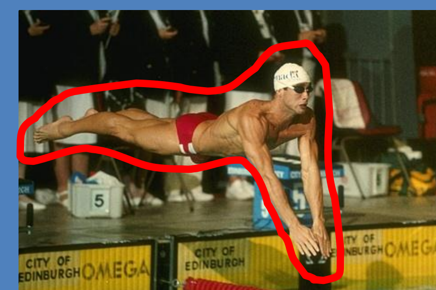
4. Apply graph cuts
5. Return to 2, using current labels to compute foreground, background models

What is easy or hard about these cases for graphcut-based segmentation?



Easier examples

More difficult Examples

	Camouflage & Low Contrast	Fine structure	Harder Case
Initial Rectangle			
Initial Result			

3 Graph-cuts (30 pts)

Let us apply Graph-cuts for foreground/background segmentation. In the "cat" image, you are given a rough polygon of a foreground cat. Apply Graph-cuts to get a better segmentation.

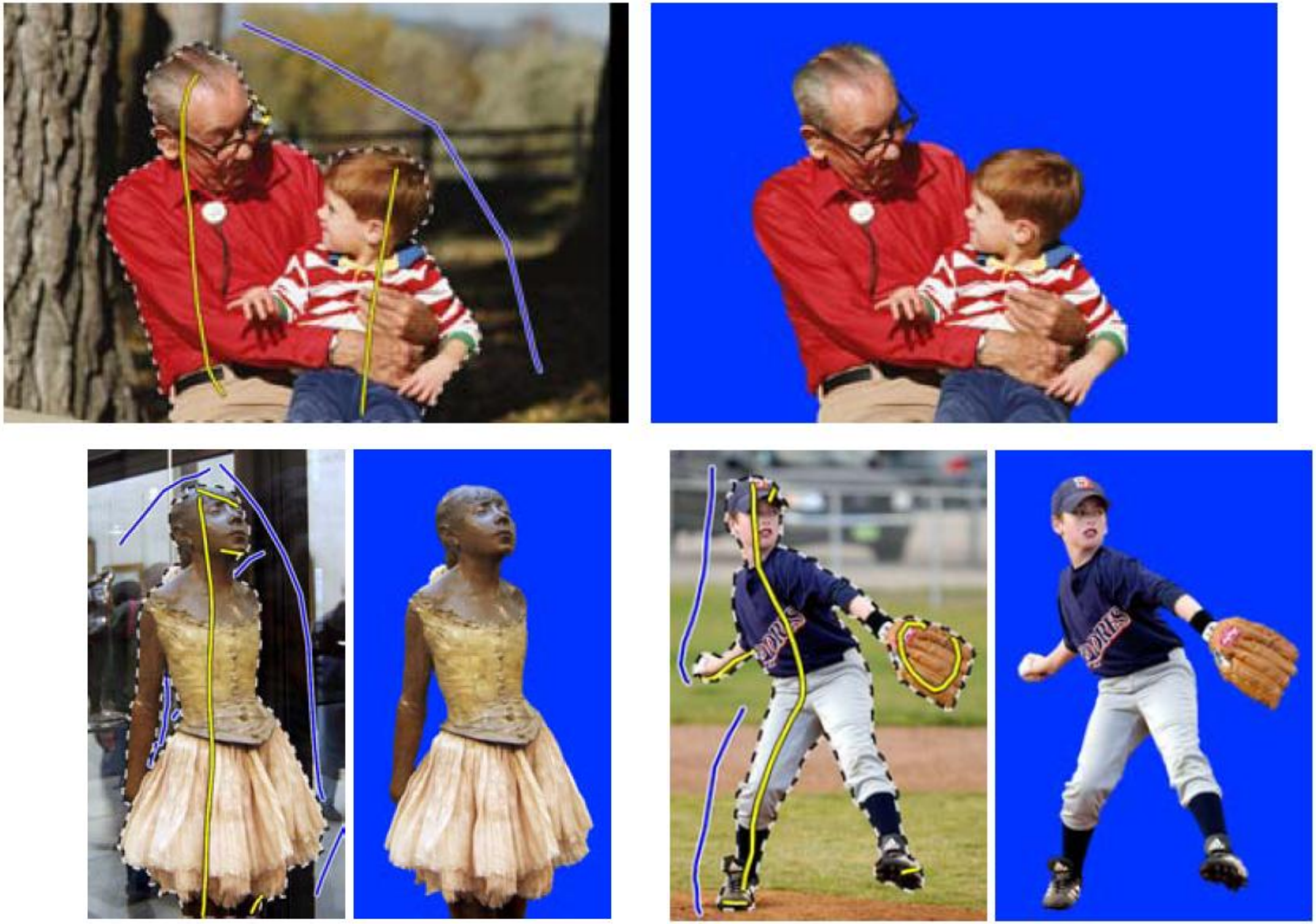
First, you need an energy function. Your energy function should include a unary term, a data-independent smoothing term, and a constrast-sensitive smoothing term. Your unary terms should be $-log[\frac{P(pixel|foreground)}{P(pixel|background)}]$. Your pairwise term should include uniform smoothing and the contrast-sensitive term. To construct the unary term, use the provided polygon to obtain an estimate of foreground and background color likelihood. You may choose the likelihood distribution (e.g., color histograms or color mixture of Gaussians. Yes, you can use MATLAB GMM functions this time). Apply graph cut code for segmentation. You must define the graph structure and unary and pairwise terms and use the provided graph cut code or another package of your choice. Include in your writeup:

- Explain your foreground and background likelihood function. (5 pt)
- Write unary and pairwise term as well as the whole energy function, in math expressions. (5 pt)
- Your foreground and background likelihood map. Display $P(foreground|pixel)$ as an intensity map (bright = confident foreground). (10pt)
- Final segmentation. Create an image for which the background pixels are blue, and the foreground pixels have the color of the input image. (10pt)

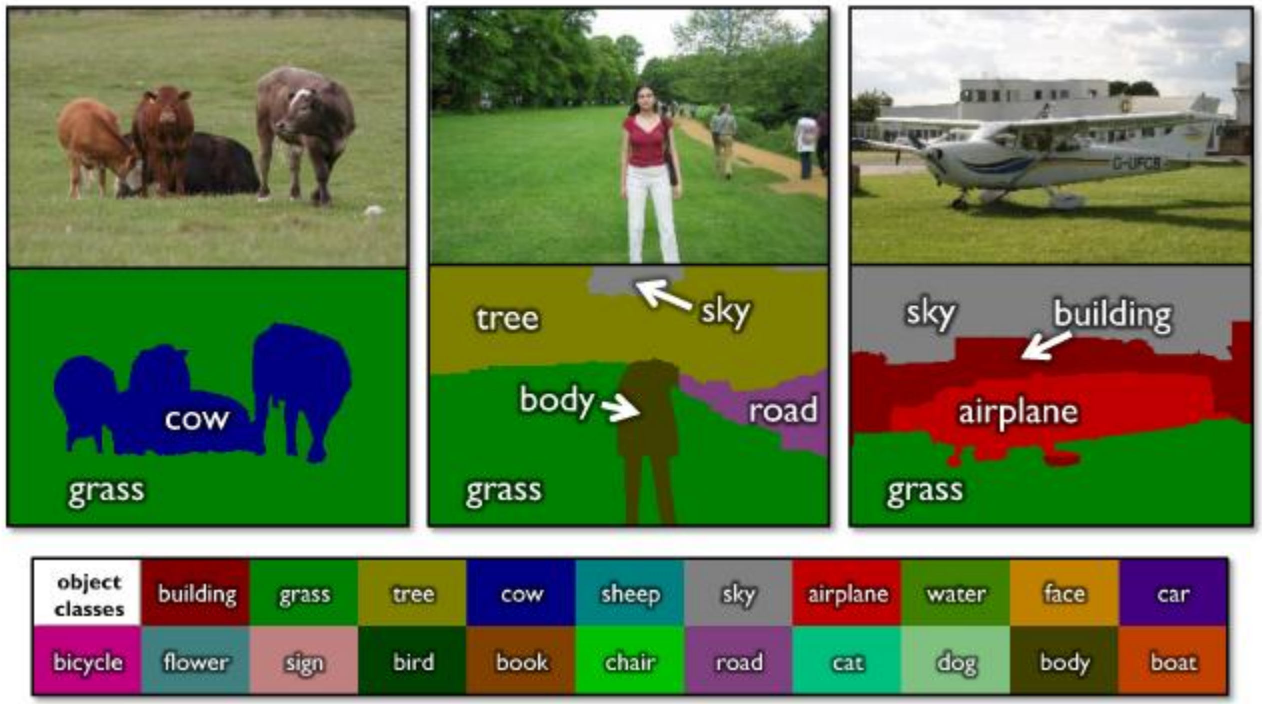
Notes

- look at GraphCut.m in provided code (skip README) – if you have trouble using package get help from Ruiqi
- Use poly2mask to convert the polygon to a foreground mask

Lazy Snapping (Li et al. SG 2004)

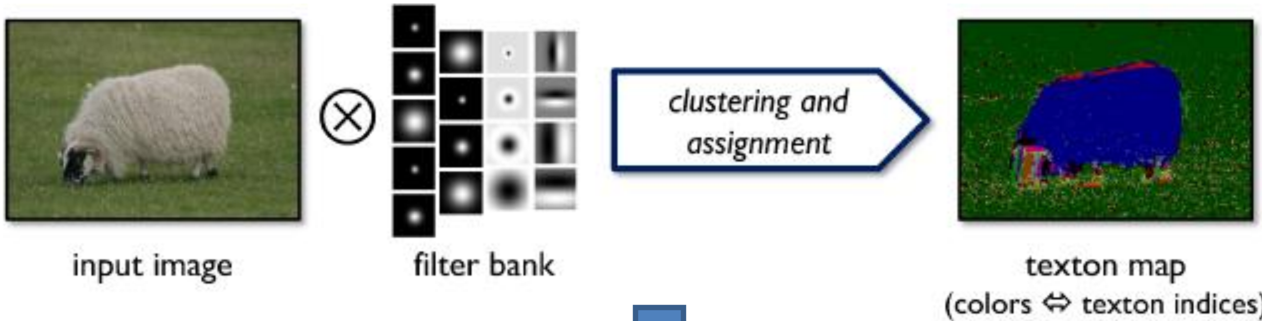


Using graph cuts for recognition



TextonBoost (Shotton et al. 2009 IJCV)

Using graph cuts for recognition



Unary Potentials

+

Alpha Expansion
Graph Cuts

TextonBoost (Shotton et al. 2009 IJCV)

Graph cuts with multiple labels

- Alpha expansion
  - Repeat until no change
    - For $\alpha = 1..M$
    - Assign each pixel to current label or α (2-class graphcut)
  - Achieves "strong" local minimum

- Alpha-beta swap
  - Repeat until no change
    - For $\alpha = 1..M, \beta = 1..M$
    - Re-assign all pixels currently labeled as α or β to one of those two labels while keeping all other pixels fixed

Limitations of graph cuts

- Associative: edge potentials penalize different labels

Must satisfy

$$E^{i,j}(0,0) + E^{i,j}(1,1) \leq E^{i,j}(0,1) + E^{i,j}(1,0)$$

- If not associative, can sometimes clip potentials

- Graph cut algorithm applies to only 2-label problems
  - Multi-label extensions are not globally optimal (but still usually provide very good solutions)

Graph cuts: Pros and Cons

- Pros
  - Very fast inference
  - Can incorporate data likelihoods and priors
  - Applies to a wide range of problems (stereo, image labeling, recognition)
- Cons
  - Not always applicable (associative only)
  - Need unary terms (not used for bottom-up segmentation, for example)
- Use whenever applicable

More about MRFs/CRFs

- Other common uses
  - Graph structure on regions
  - Encoding relations between multiple scene elements
- Inference methods
  - Loopy BP or BP-TRW
    - Exact for tree-shaped structures
    - Approximate solutions for general graphs
    - More widely applicable and can produce marginals but often slower

Further reading and resources

- Graph cuts
  - http://www.cs.cornell.edu/~rdz/graphcuts.html
  - Classic paper: What Energy Functions can be Minimized via Graph Cuts? (Kolmogorov and Zabih, ECCV '02/PAMI '04)

- Belief propagation

  Yedidia, J.S.; Freeman, W.T.; Weiss, Y., "Understanding Belief Propagation and Its Generalizations", Technical Report, 2001: http://www.merl.com/publications/TR2001-022/

Next section: Object Recognition

- Face recognition
- Image categorization and general classification methods
- Object detection
  - Statistical templates
  - Parts-based methods
- Tracking objects