# Medical Image Analysis

CS 778 / 578

Computer Science and Electrical Engineering Dept.
West Virginia University

March 2, 2011

# Outline

# Outline

# Graph Cuts

- Reduce the problem of image segmentation to the problem of graph cuts.
- We can then use existing algorithms for graph cuts to perform segmentation.

We will present an **interactive** technique for segmenting an object from the background (only 2 regions).
The user must specify some hard constraints :

- some pixels which are known to belong to the object,
- and some pixels which are known to belong to the background.

# Graph

Let $G = \{V, E\}$ be a graph which consists of nodes $V$ and edges $E$.
In general, the edges may be

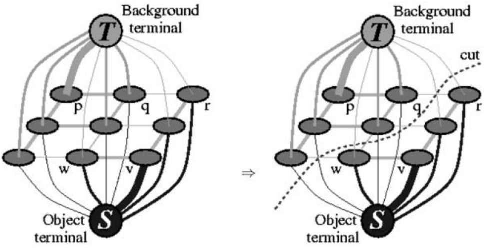
- directed or undirected
- weighted or unweighted

The algorithm presented here uses a weighted, undirected graph.
Additionally, all weights are assumed to be positive.

# Outline

# The min cut problem

- Let the set of nodes include 2 special nodes (terminal nodes) labeled S and T.
- An S/T cut (or just ”cut”) is a subset C of E such that there is no path between S and T when C is removed from E.
- The cost of a cut is the sum of all of the weights of the edges in C.
- The min-cut problem is to find the cut with minimum weight.




(b) Graph. (c) Cut.

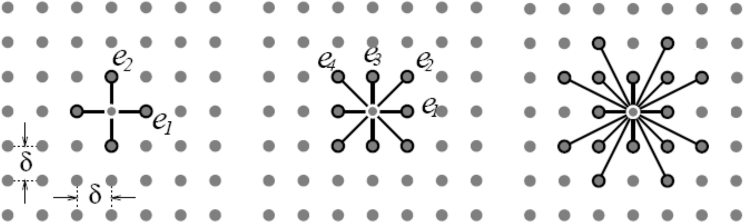
# Outline

# Overview

- Convert an input to the segmentation problem into an input to the graph cuts problem:
  - Convert image and initialization into a graph, G.
  - Give high weights to edges between similar nodes
  - Give low weights to edges between dissimilar nodes
- Find minimum cost cut, C, of the graph G.
- Convert the output C, into a segmentation.

# Create the graph, G

Convert the image into an undirected, weighted graph.




(a) 4 n-system (b) 8 n-system (c) 16 n-system

# Create the graph, G

Convert the image into an undirected, weighted graph.

- Create a node $v \in V$ for each pixel in the image.
- Pick a neighborhood system to use
  - for example : 4 or 8 neighbor system in 2D
  - for example : 6 or 26 neighbor system in 3D
- For each pair of neighbors create an edge (n-link).
- Set the weight for each n-link to $B_{p,q}$ (pixel similarity).

$B_{p,q}$ is a measure of the similarity of image intensities at pixels $p$ and $q$.

$$B_{p,q} = c \exp\left(-\frac{(I_p - I_q)^2}{2\sigma^2}\right)\frac{1}{dist(p,q)}$$

# Create the graph, G

Incorporate the user supplied initialization - a set of pixels O and B in object and background.

- Create 2 terminal nodes, S and T.
  - S node represents the "object"
  - T node represents the "background"
- For each image vertex, p, create an edge to S ("obj terminal")
  - If $p \in O$ then $w_{p,S} = K$ (a large constant)
  - If $p \in B$ then $w_{p,S} = 0$
  - Else, $w_{p,S} = \lambda R_p(\text{"}bkg\text{"})$
- For each image vertex, p, create an edge to T ("bkg terminal")
  - If $p \in O$ then $w_{p,T} = 0$
  - If $p \in B$ then $w_{p,T} = K$
  - Else, $w_{p,T} = \lambda R_p(\text{"}obj\text{"})$

$R_p(A)$ is a region-based function reflecting how well the intensity at p fits into "object" or "background" initialization.

# Create the graph, G

Use the initialization regions to define histograms for region-based segmentation

Let

- $Pr(I|O)$ be the intensity histogram of the object initialization
- $Pr(I|B)$ be the intensity histogram of the background initialization

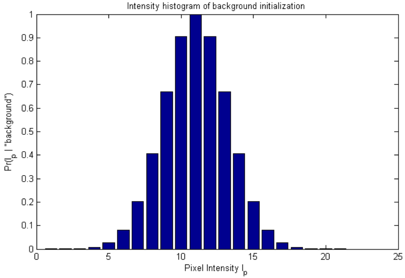
Use negative log-likelihoods to determine weights

$$R_p(\text{"}obj\text{"}) = -\ln Pr(I_p|O)$$
$$R_p(\text{"}bkg\text{"}) = -\ln Pr(I_p|B)$$

Similar penalty terms have previously been used in MAP-MRF approaches to segmentation.

# Region-based weights

For the object and background regions create a histogram of image intensities.



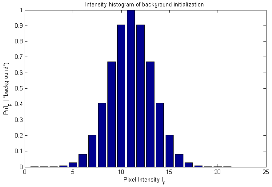

Create edges between each pixel and both terminal nodes.

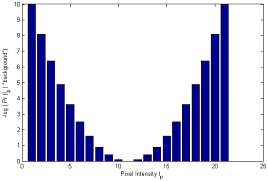
The weight on the edge between a pixel and the object node (S) is the negative log likelihood of that pixel belonging to the background initialization.

# Region-based weights

Negative log converts:

- low probabilities into high values.
- high probabilities into low values.

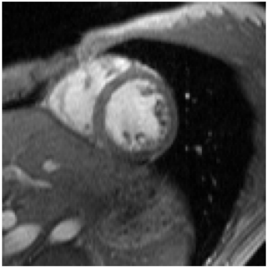


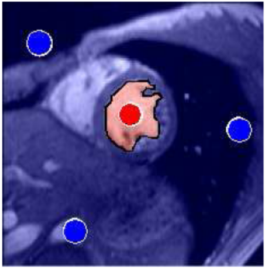




So, if the pixel is unlikely to belong to the background, the edge to the object node has high weight. This edge is unlikely to be cut.

Likewise, edge weight to the background node depends on the object initialization.

# Find min cut





# Find min cut

Known algorithms for finding the min cut:

- Ford-Fulkerson method (Edmunds-Karp algorithm) - $O(VE^2)$

The 'maxflow' algorithm optimized for graph cuts on grid graphs was presented in:
"An Experimental Comparison of Min-Cut/Max-Flow Algorithms for Energy Minimization in Computer Vision.", Yuri Boykov and Vladimir Kolmogorov. In IEEE Transactions on Pattern Analysis and Machine Intelligence, September 2004.

- Outperforms other algorithms
- Seems to scale linearly in the number of pixels.

# The output

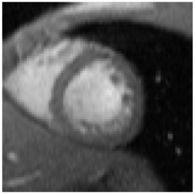
The function $A_p(C)$ returns the region of pixel, p, given the cut, C.

Recall:

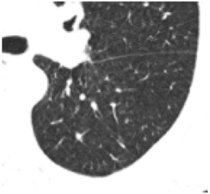
- S node represents the "object"
- T node represents the "background"

$$A_p(C) = \{ \begin{array}{ll} \text{"obj"} & \text{if } \{p, T\} \in C \\ \text{"bkg"} & \text{if } \{p, S\} \in C \end{array}$$

# Results

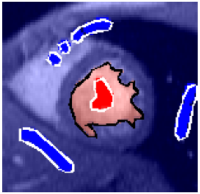


(c) Cardiac MR

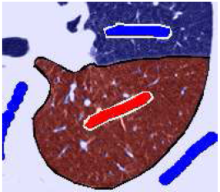


(e) Lung CT

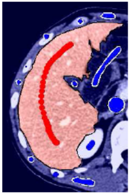
(g) Liver MR



(d) LV Segment

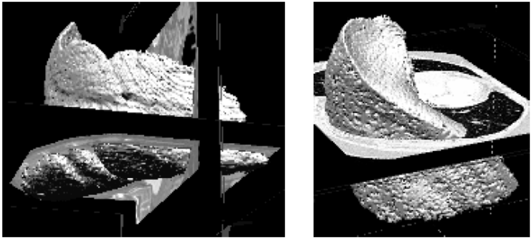


(f) Lobe Segment



(h) Liver Segment

# Results



(a) Liver, 144x170x170 (b) Lung, 253x165x205

# Timing

| method | 2D examples | | | | | |
|---|---|---|---|---|---|---|
| | *Bell* photo (255x313) | | Lung CT (409x314) | | Liver MR (511x511) | |
| | N4 | N8 | N4 | N8 | N4 | N8 |
| DINIC | 2.73 | 3.99 | 2.91 | 3.45 | 6.33 | 22.86 |
| H_PRF | 1.27 | 1.86 | 1.00 | 1.22 | 1.94 | 2.59 |
| Q_PRF | 1.34 | 0.83 | 1.17 | 0.77 | 1.72 | 3.45 |
| Our | 0.09 | 0.17 | 0.22 | 0.33 | 0.20 | 0.45 |

| method | 3D examples | | | | | |
|---|---|---|---|---|---|---|
| | Heart MR (127x127x12) | | Heart US (76x339x38) | | Kidney MR (99x66x31) | |
| | N6 | N26 | N6 | N26 | N6 | N26 |
| DINIC | 20.16 | 39.13 | 172.41 | 443.88 | 3.39 | 8.20 |
| H_PRF | 1.38 | 2.44 | 18.19 | 47.99 | 0.19 | 0.50 |
| Q_PRF | 1.30 | 3.52 | 23.03 | 45.08 | 0.19 | 0.53 |
| Our | 0.70 | 2.44 | 13.22 | 90.64 | 0.20 | 0.58 |

# Summary

Graph cuts for image segmentation

- Is fast.
- Results are dependent on user initialization.
- Incorporates edge and region information.