Austen Hsiao
Paper Summary 1

Summary of *Graph Cuts and Efficient N-D Image Segmentation* by Yuri Boykov and Gareth Funka-Lea

Graph cutting is a method to split up a graph. To decide the "best" cut, partitions come with a price, by minimizing this price we get the "best" overall grouping. For a physical picture, the global minimum for the cost function would provide data in a way where every object is partitioned. This begs the question—how do we get a graph out of an image?

Boykov and Funka-Lea's solution created vertices for each pixel. Two specific kinds of edges exist to connect vertices—n-links, which join neighbors, and t-links, which join pixel nodes to terminal nodes. These terminal nodes represent the foreground, which they call "source", and a background, which the authors call "sink". The best "seeds" (terminal nodes) are chosen manually. But the author notes that too many seeds defeat the purpose of the algorithm, since at a certain point, we're manually defining boundaries. To make the "best" cuts, if two neighbors are very similar, the edge between them will have a very large cost to cut. Conversely, if two neighbors are very different, the edge between them will have a relatively low cost to cut.

This is about as far as I want to get before I succumb to headaches from reading the authors' cost function. I snipped a picture of the function below. I skimmed most of the paper, so I don't know if they covered this-- but this kind of work would have profound effects on early cancer detection and any image to be processed would be enhanced by the application of a high-pass filter for stronger edge detection.

$$
\begin{aligned}
|C| &= \sum_{p \notin \mathcal{O} \cup \mathcal{B}} \lambda \cdot R_p(A_p(C)) + \sum_{\{p,q\} \in \mathcal{N}} B_{p,q} \cdot \delta_{A_p(C) \neq A_q(C)} \\
&= E(A(C)) - \sum_{p \in \mathcal{O}} \lambda \cdot R_p(\text{“obj”}) - \sum_{p \in \mathcal{B}} \lambda \cdot R_p(\text{“bkg”}).
\end{aligned}
$$