Report Seminar Presentation
Nicolas Eng

Project – Auto-Segmentation for Spine CTs for Data-Intensive Analysis of Surgical Outcome.

Team members: Ben Ramsay, Nicolas Eng

Project Objective:
The overall goal of our project is to develop a state-of-the-art automatic segmentation method for spine CT images using max-flow/min-cut optimization.

Here are four project milestones that span our initial work to maximum deliverables that we hope to reach for a successful project.

- **Minimum** deliverable (3/9): Extend initial algorithm for automatic segmentation of spine CT and larger datasets
- **Expected** deliverable (3/23): Evaluate accuracy vs. parameter selection on N=20 spine CT dataset (already manually segmented)
- **Expected** deliverable (4/20): Accurate segmentation of N=200 spine CT dataset from The Cancer Imaging Archive (TCIA)
- **Maximum** deliverable (5/15): Develop methods to extend algorithm for patient-specific parameter selection in order to have accurate segmentation for a wide-variety of cases.

We are motivated for this specific task, since our work will be a major component of "Spine Cloud" a multi-year project proposed by the Dr. Siewerdsen's I-STAR lab. "Spine Cloud" hopes to curate a database consisting of patient demographic data, image and specific anatomy, surgical procedures, and pathologies. Once organized, we hope to correlate these defined clinical variables and automatic image analysis to patient surgical outcomes. By developing this highly quantitative approach on how to approach future spine surgeries, "Spine Cloud" will provide more favorable and consistent outcomes.

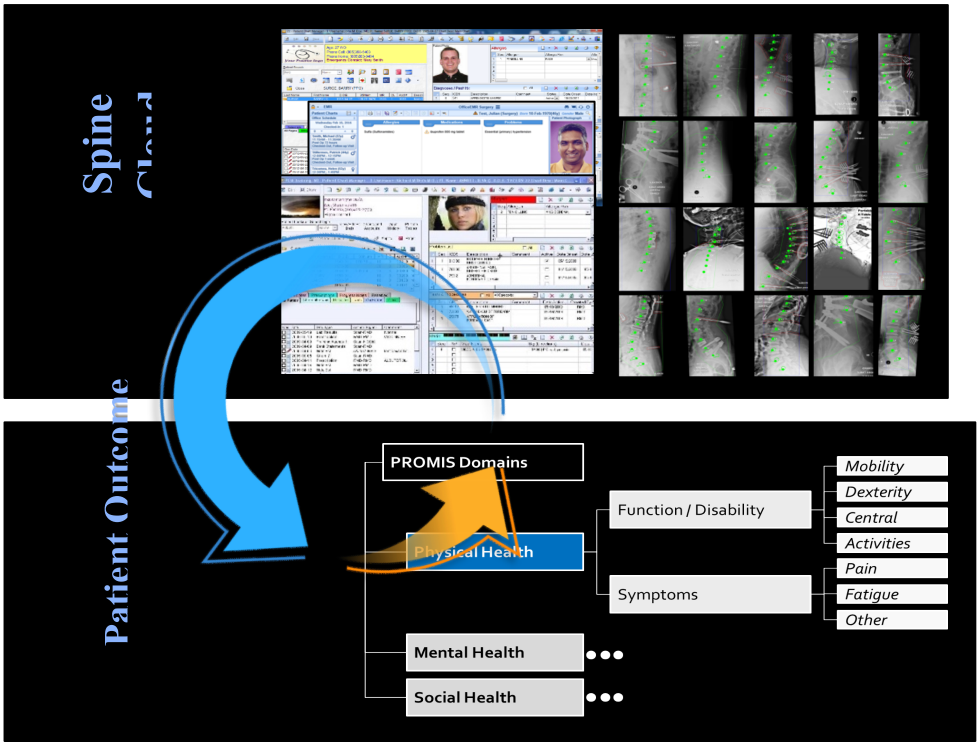

Figure 1: Spine Cloud workflow

A necessary component of "Spine Cloud" is a large database of annotated spine CT images based on accurate, automatic segmentation. Currently within the I-STAR lab, segmentation of spine CTs is handed manually which while accurate is often time-consuming. While there are simple techniques for auto-segmentation like Thresholding and Region Growing that are computationally efficient and easy to implement, they often fail to give accurate segmentation. With these techniques, each voxel is treated independent in the CT image, which prohibits local neighborhood relations like smoothness and curvature to be accounted for during segmentation. Instead, we propose to treat segmentation as an energy minimization problem which can account for local relations by transforming the CT image into a graph and using Max-Flow / Min-Cut Optimization in order to minimize the energy function.

Paper Selection:

I have chosen to review Boykov's classic paper "Fast Approximate Energy Minimization via Graph Cuts". This paper was first presented as conference paper in the "International Conference for Computer Vision" in 1999, and thereafter, submitted to the journal IEEE Transaction on PAMI in 2001. I have chosen this particular paper since it details how to use graph cuts to minimize many different kinds of energy functions. The methods detailed are highly versatile since many computer vision can be formulated in terms of energy minimization like stereo, image restoration, segmentation, and motion. I am particularly interested in this paper since our own graph-cut implementation that we are using to do binary label segmentation of spine CTs is based off of the theory in this paper.

Paper Background:

$$E(f) = E_{smooth}(f) + E_{data}(f).$$

Figure 2: Energy Equations

Many computer vision problems can be naturally formulated into solving an energy minimization problem. The overall energy function is comprised of two factors: $E_{smooth}(f)$ and $E_{data}(f)$, a smooth component and data component respectively. For a given labeling f, The smooth component expresses the similarity of neighboring pixels based on measurable quantities (i.e. intensity), and the data component expresses how labeling compares to observed data or priors. While formulation of an energy function is possible, determining the minima is often difficult due to computational costs, many local minima, and a large possible label space.

Paper Goals:
The authors hope to answer the question: How does one minimize an energy function in a quick and computationally efficient manner? There have been previous methods proposed like simulated annealing which can theoretically approach the global minimum of any arbitrary energy function. However, a limitation of this approach is that the methodology is restricted to standard moves meaning only a single pixel can change at each iteration. Therefore, while in theory the approach is attractive, often simulated annealing can be computationally expensive and take exponential time.

In order to solve the problem in a computationally efficient manner, the authors present two innovative graph cut methods α-β swaps and α-expansion. These two graph cut methods allow for large moves meaning many pixel labels can change at each iteration. Additionally, there are aa number of nice properties detailed in their results. For binary labelling an exact global minimum can be reached in polynomial time. For multi-labeling with α-expansion, minima have guaranteed bounds of a factor within global minimum

Graph Cuts Review:

For a weighted graph with two vertices named terminals (α,β in figure 3), the graph cut C is a set of edges that separates the two terminals in the induced graph such that no flow can go from one terminal node to the other. The cost of the cut C is the sum of the edge weights. Therefore, the minimum cut is the cheapest cut possible that separates the two terminal.

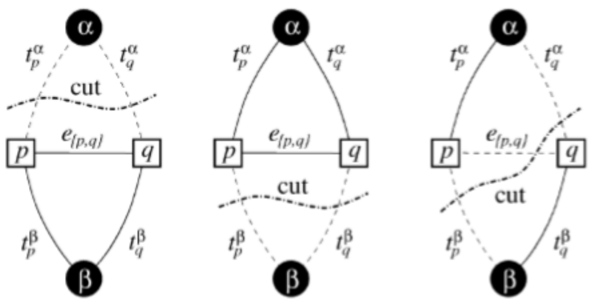

Figure 3: Examples of different graph cuts separating the two terminals **α, β**

Example of Graph-Cuts for Segmentation:

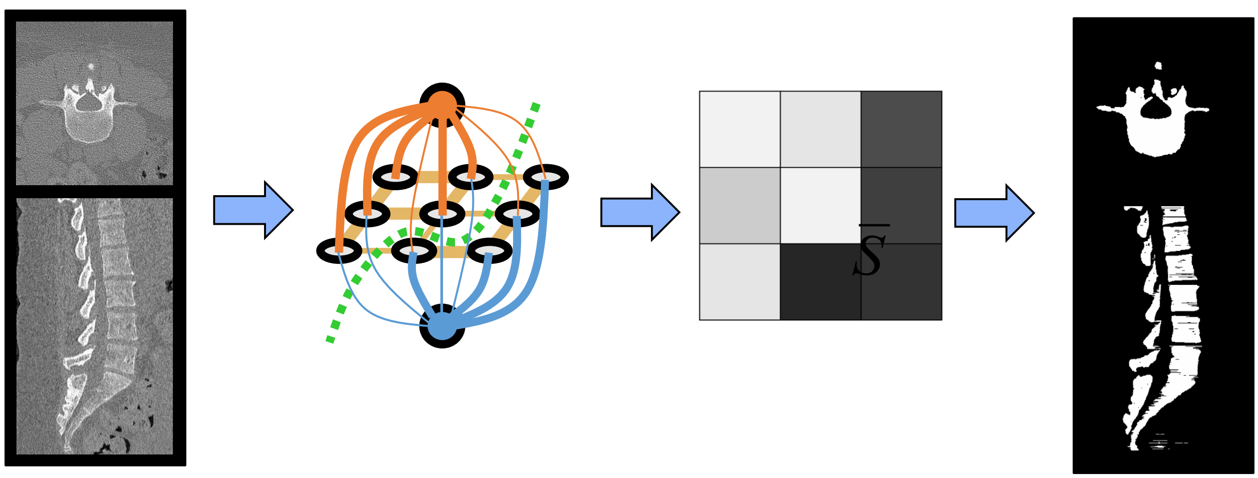

Figure 4: Example of Graph-Cut Segmentation for Spine CTs

As an example of graph-cuts applied to images, we first start with a CT image of the spine like in Figure 4. Then, to define the graph we take each voxel in the CT image and set each as a voxel node (gray ellipses) and then additionally define two terminal nodes (orange and blue circles). Each voxel node is connected through edges to neighboring voxel nodes as well as to both terminal nodes. The weights are defined in order to have the best separation between bone and background. Terminal node to voxel node edges are determined by priors contribute toe the E_{data} component of the energy function. Voxel node to voxel node edges are determined by measurable quantities of the data and contribute to the E_{smooth} component of the energy function. Ultimately, the edge weights determine the overall energy function, E, and the minimum cut which is the cheapest cut to separate the two terminal nodes minimizes the energy function.

Proposed Graph-Cut Methods

α-β swaps

1. Start with an arbitrary labeling f
2. Set success := 0
3. For each pair of labels $\{\alpha, \beta\} \subset \mathcal{L}$
   3.1. Find $\hat{f} = \arg\min E(f')$ among f' within one α-β swap of f (Section 3)
   3.2. If $E(\hat{f}) < E(f)$, set $f := \hat{f}$ and success := 1
4. If success = 1 goto 2
5. Return f

Figure 5: Pseudo-Code for α-β swaps

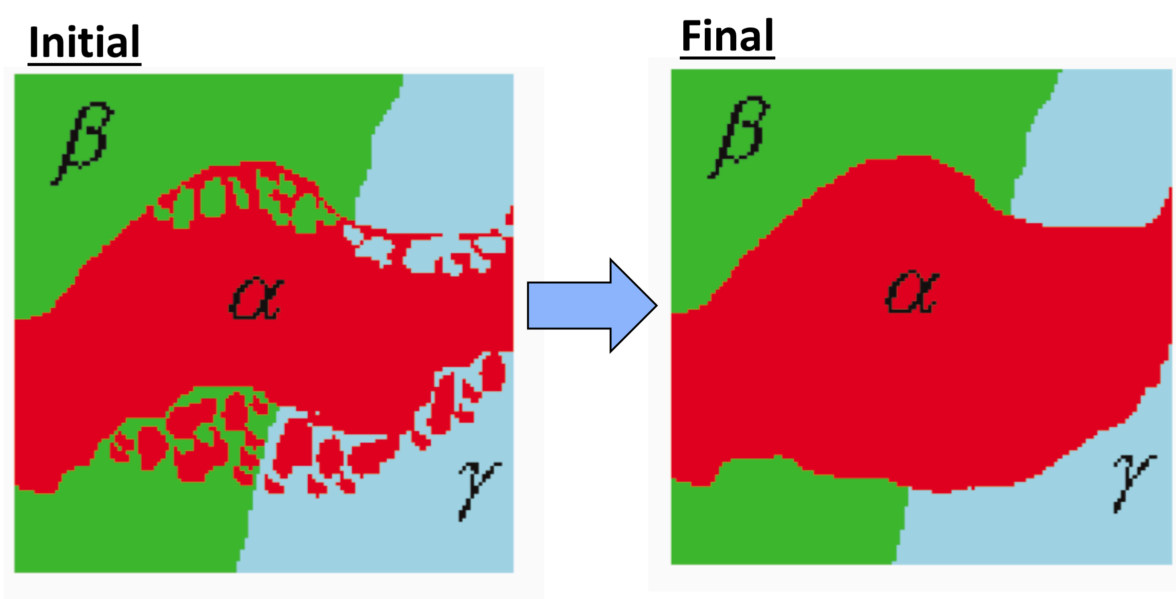

Figure 6: Visualization of α-β swap

Explanation of α-β swaps

First starting with an arbitrary labeling of the graph f, we explore many alternative labelings f' that are chosen with various α-β swaps where an α-β swap is defined to be arbitrary swapping of labels for α, β pairs. Then the f' that best minimizes the energy function is f_{hat}, and if the new labelling f_{hat} has a lower energy than the original labeling f, the algorithm repeats with f_{hat} as the new labelling. If not, then we have reached the minimum of the energy function.

α-Expansions

1. Start with an arbitrary labeling f
2. Set success := 0
3. For each label $\alpha \in \mathcal{L}$
   3.1. Find $\hat{f} = \arg\min E(f')$ among f' within one α-expansion of f (Section 4)
   3.2. If $E(\hat{f}) < E(f)$, set f := $\hat{f}$ and success := 1
4. If success = 1 goto 2
5. Return f

Figure 7: Pseudo-Code for α-Expansion

Explanation of α-Expansions
First starting with an arbitrary labeling of the graph *f*, we explore many alternative labelings *f'* that are chosen with various α-expansions where an α-expansion is defined to be a move which assigns arbitrary labels to α. Then the *f'* that best minimizes the energy function is f_{hat}, and if the new labelling f_{hat} has a lower energy than the original labeling *f*, the algorithm repeats with f_{hat} as the new labelling. If not, then we have reached the minimum of the energy function.

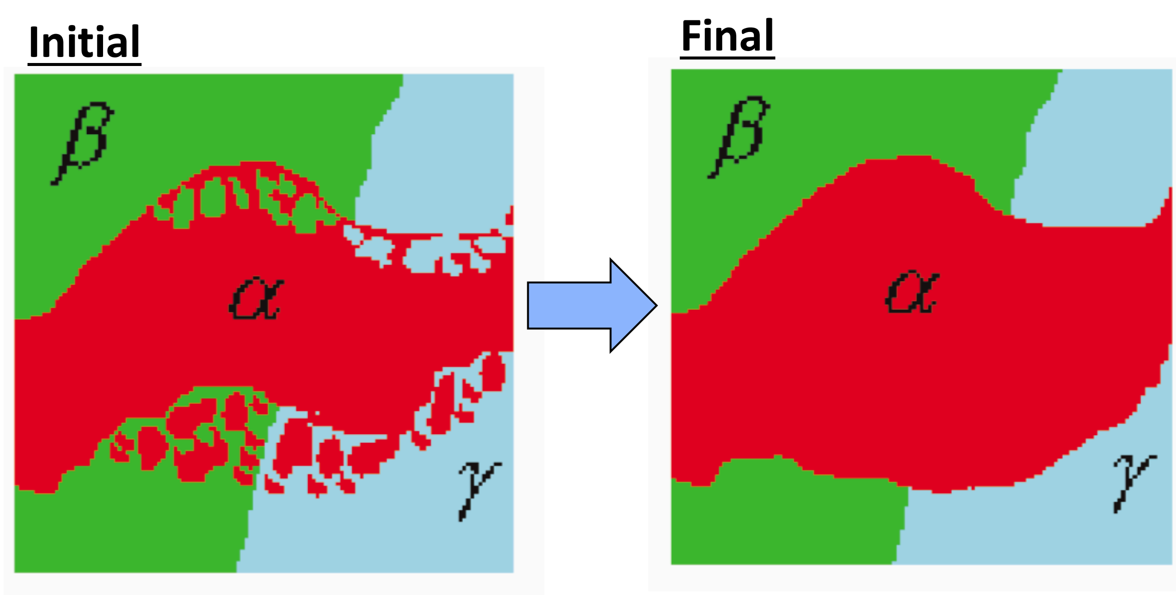

Figure 8: Visualization of α-Expansions

Results

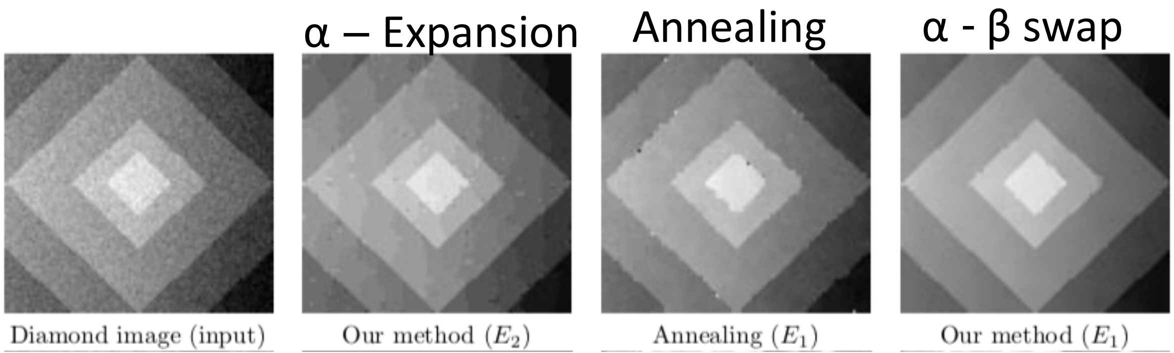

Figure 9: Image Restoration task

	E			E_{smooth}		
	Our results	Annealing	Ratio	Our results	Annealing	Ratio
	Diamond (image restoration, E_1)					
First cycle ($t = 36$)	1,577	55,892	**35.5**	637	9,658	**15.2**
Last cycle ($t = 389$)	1,472	15,215	**10.3**	576	8,475	**14.7**
Best annealing ($t = 417,317$)	—	1,458	—	—	571	—

Figure 10: Energy vs. Time Table

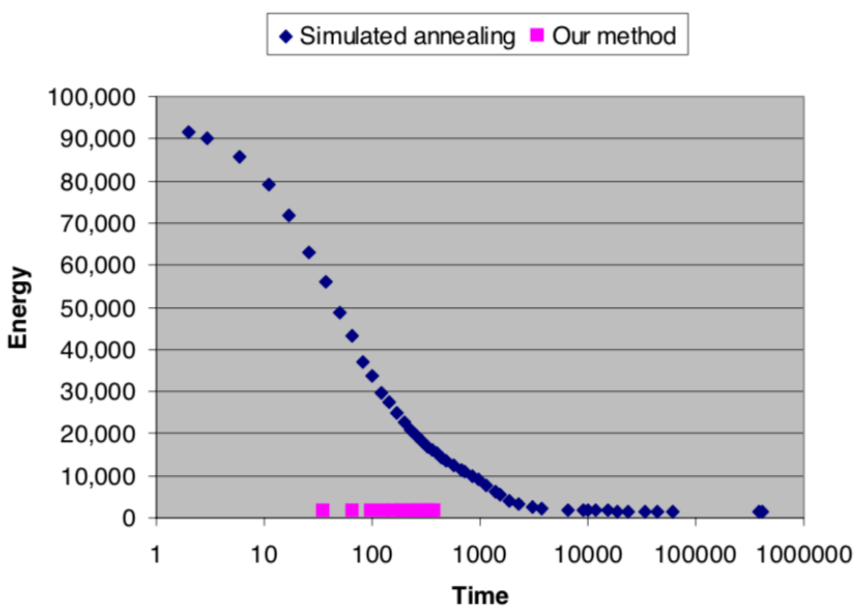

Figure 11: Energy vs. Time Graph

After developing the two graph cut algorithms the authors tested their results on 3 different tasks image restoration, motion, and stereo. I have chosen to only details the results of the image restoration task as the findings are identical for the motion and stereo examples. Image restoration is blurring an image like the diamond image from (Figure 9) on the left and trying to restore it to its original clarity. The authors compared α-β swaps, α-expansion, and annealing for this task, keeping track of the energy values and time for convergence. The α-β swaps reached the global minimum fast in comparison to the annealing. This shows the power of the graph-cut algorithms as α-β swaps reached the energy global minimum in 389 seconds while it took >400000 seconds for the annealing to reach the energy global minimum.

Conclusion / Paper Assessment:

Pros:

- Generalizable to many computer vision problems (i.e. segmentation, stereo, image restoration, motion)
- Computationally efficient and speedy when compared to the then-current algorithms
- Binary labelling guarantees reaching global minimum
- For multi-labeling with α-expansion, minima have guaranteed bounds of a factor within global minimum

Cons

- The Smoothness Functions are limited to pairs of adjacent pixels
- Graph cut methods take a discrete approach

References

Boykov, Y., et al. "Fast Approximate Energy Minimization via Graph Cuts." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 11, 2001, pp. 1222–1239., doi:10.1109/34.969114.