Auto-Segmentation of Spine CT for Data-Intensive Analysis of Surgical Outcome.

Group Number: 21

Team members: Ben Ramsay, Nicolas Eng
Mentors: Dr. Jeff Siewerdsen, Dr. Tharindu de Silva

Project Objective:

The goal of our project is to develop an automatic segmentation method for spine CT images using max-flow/min-cut optimization.

Introduction and Motivation:

We are motivated for this specific task, since our work will be a major component of “Spine Cloud”, a multi-year project proposed by Dr. Siewerdsen of the I-STAR lab. “Spine Cloud” hopes to curate a database consisting of patient demographics, images, 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.

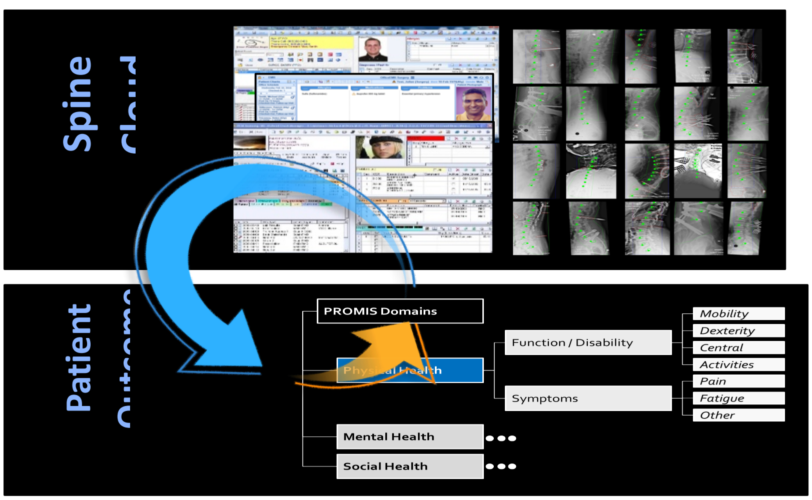


Figure I1: Spine Cloud workflow

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 CT is handed manually which, while accurate, is often time-consuming and very inefficient. 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 produce an accurate segmentation. With these techniques, each voxel is treated independently in the CT image, which prohibits local neighborhood relations like smoothness and curvature.

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 to minimize the energy function.

Graph Cut Background:

For a weighted graph with two vertices and α,β terminals, 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 cut C is the sum of the edge weights. Therefore, the minimum cut is the cheapest cut possible that separates the two terminals. This cut represents the Min-Cut / Max-Flow algorithm.

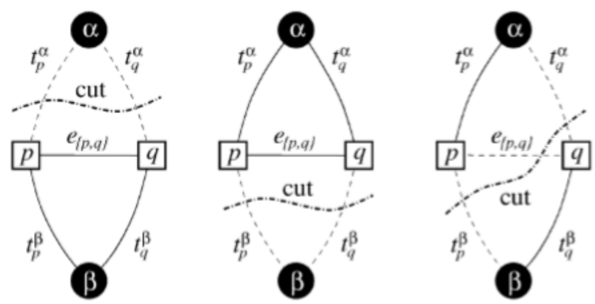


Figure B1: Examples of Different Graph Cuts Separating the two Terminals **α, β**

As an example of how our method applies graph-cuts, we first start with a CT image of the spine (Figure B2). To define the nodes of the graph, we take each voxel in the CT image and create a node (gray) and then define two terminal nodes (orange and blue). Each voxel node is connected by an edge to every neighboring voxel node as well as to both terminal nodes.

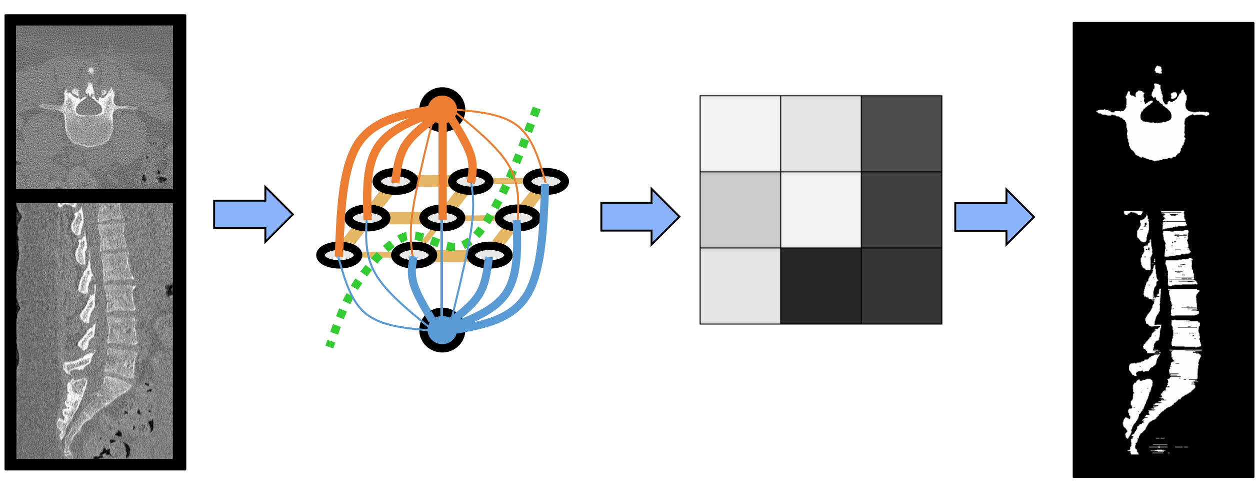


Figure B2: Example of Graph-Cut Segmentation for Spine CTs

The values of the terminal link weights are based on prior information detailing the likelihood that certain voxel are a part of the spine or the background. These priors ideally will have different likelihoods for spine vs background allowing for the best separation between bone and background. Voxel node to voxel node edges are determined by measurable quantities of the image and contribute to how smooth the segmentation will appear. Ultimately, the edge weights determine the final cut that will be made since the cut made must minimize the sum of the weights which are cut. In this minimization, the terminal nodes are separated from each other and the resulting segmentation is assigned based on which terminal node a given voxel is connect to.

Overall Technical Approach:

Our segmentation relies on the implementation of a 2D max-flow min-cut optimization method. The MATLAB code to execute this optimization was given to us by our mentors. Upon receiving the code we explored the parameter space to see what affects certain parameters would have on the segmentation. To access the accuracy of our segmentation we implemented root mean squared error(RMSE) and dice coefficient metrics. Our implementation was tested using these metrics on the manually segmented N20 spine CT dataset. Once we had a good understanding of the algorithm, we shifted our method from a 2D implementation to a 3D implementation. We also incorporated information of centroid positions into our algorithm which informed us of the center of each vertebrae in the CT image. Once our methodology was working in the N20, our mentors encouraged us to continue to adjust the algorithm to segment the N200 dataset.

Since the N200 is not manually segmented, the only way to evaluate the performance is qualitatively. For this reason, our mentors encouraged us to segment one of the N200 members so that our segmentation method could be validated in the N200. Our approach to the N200 dataset was to first segment the spine without the spinous process, and then attempt to incorporate the spinous process and address certain anomalies like instrumentation and tumors that might throw off the segmentation. The segmented N200 dataset will be directly used as a data repository for the Spine Cloud project.

Specific Technical Approach:

2D Max-Flow / Min-Cut Implementation for N20

Pre-Processing / Defining Weights By Intensities.

In order to find quantitative ways to better separate bone and background within the CT image, we first began by taking advantage of an already manually segmented dataset. Using the manually segmented spine region as a mask to identify the spine in the CT image (red outline in Figure T1).

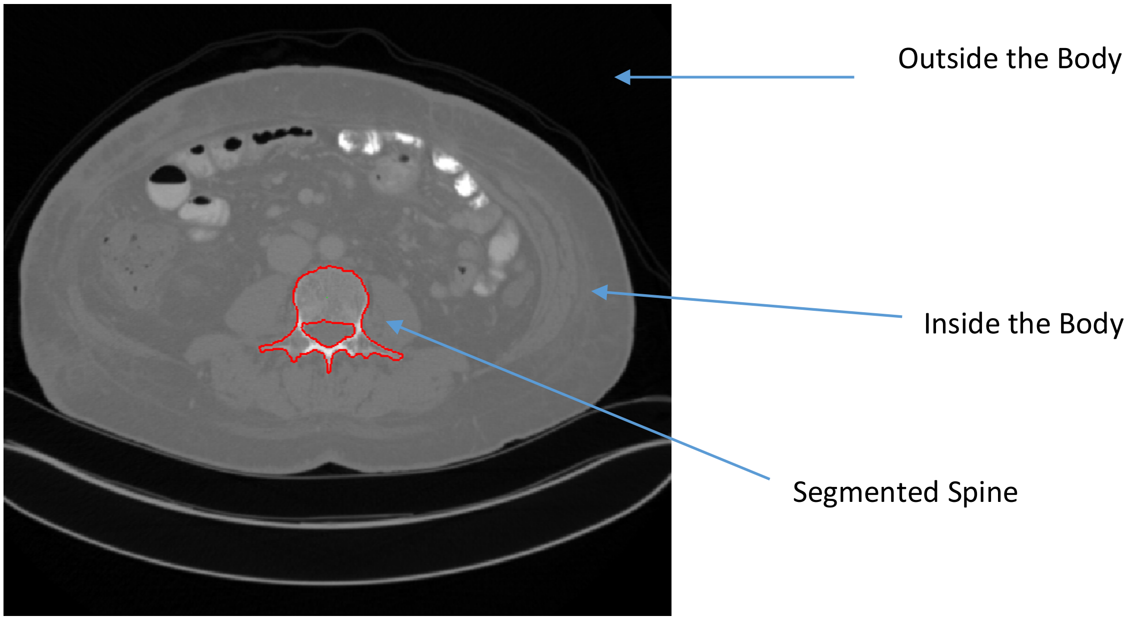


Figure T1: Axial Slice 261 of Patient 3 where the manually segmented part is outlined in red.

The Hounsfield intensities were extracted for both the inside and outside of the segmentation. Spine intensity profile histograms were made of the of the foreground versus the background. These values were normalized to the max value within the entire CT image (Figure T2). All depicted images are of a single slice in Patient 3 from the N20 dataset, though we applied this process to the entire spine.

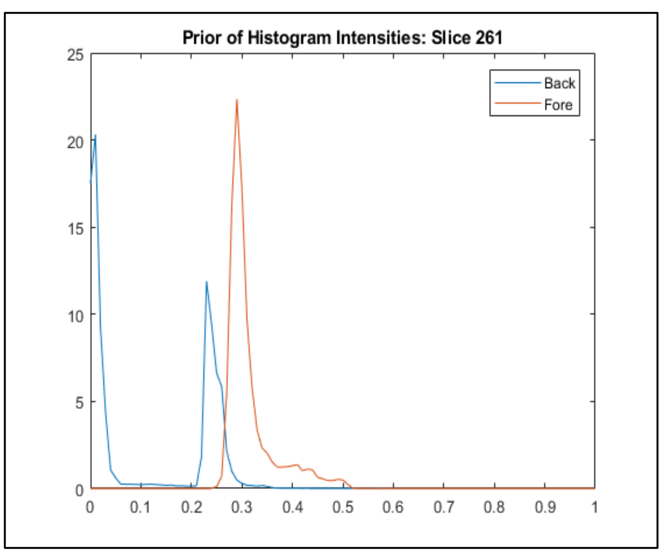


Figure T2: Intensity profile of Fore (Spine) vs. Back (All Else) for Spine 3 slice 261

We noticed that there were two major peaks for the histogram defining the back weights. The first due to the total black background outside of the body and the second from the gray intensities within the body (Shown with the arrows in Figure T1). Since there is no overlap of outside of the body intensity values and the actual spine intensity values, we can better separate the spine from the body by rescaling the histogram intensities. We decided on the best normalization from 0.2 to 0.6 purely qualitatively since we were simultaneously developing our quantitative metrics RMSE and Dice Coefficient at the time. The renormalized histogram is depicted in Figure T3

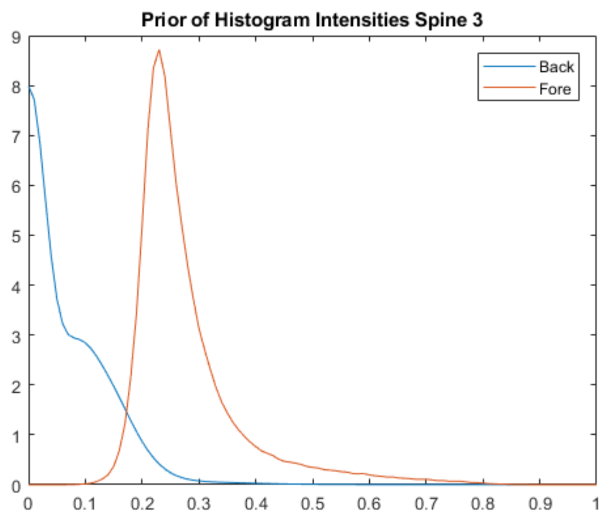


Figure T3: Rescaled intensity profile of Fore (Spine) vs. Back (All Else) for Spine 3

Our terminal node weights were determined solely based on these intensity histograms. After normalization, the intensity of the voxel would either be within the bounds of 0 to 1 or outside of it. If the value was either less than or greater than 1, it was assigned to the background. If the voxel's intensity value was between 0 and 1, then that voxel's connection to the background terminal node would be the value of the back weight histogram in the rescaled intensity profile for that voxel's intensity. Similarly, the weight connecting the node to the spine terminal node would be set by the value of the foreground histogram for the node's intensity value. The concept is that voxels who have intensities commonly found in the spine will be anchored to the spine terminal node with strong weights. Since strong weight likely won't be cut, these voxels will likely be segmented as the spine. Conversely, voxels with intensities that are normally associated with soft tissue or air will have a strong connection to the background terminal node and will be segmented as such. Visualizations of the fore and back weights for a single CT slice may be seen in Figures T4 and T5, respectively.

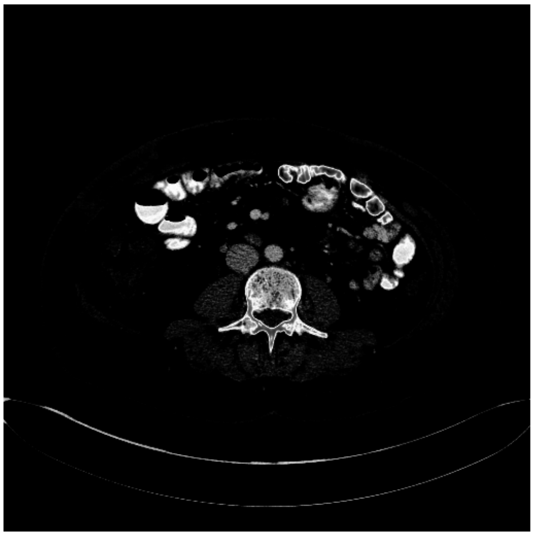


Figure T4: Fore weights of the spine

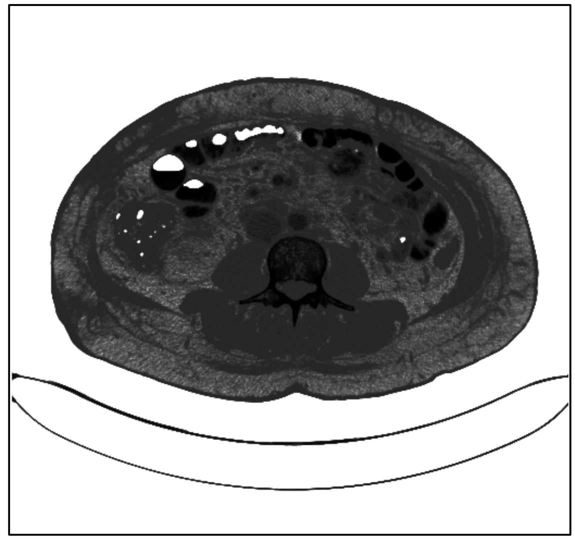


Figure T5: Back weights of the spine

With just this pre-processing step and using histogram intensity profiles to define weights, we obtain a qualitatively good segmentation of the spine. However, other parts of the anatomy were included within the segmentation of the spine (see Figure T6 in slice 367, and tissues like slice 518). This is likely due to the N20 dataset coming from The Cancer Imaging Archives(TCIA) and the patient likely being exposed to contrast before the scan. This would cause parts of the soft tissues that absorb contrast and have a higher intensity in the CT scan. In turn, this would have a strong weight to the fore terminal node and cause it to be segmented as part of the spine. After discussing our results with our mentors, they suggested taking advantage of centroid positions of the dataset. This allowed us to add an additional parameter for the weights to not only segment based on intensity but also by location.

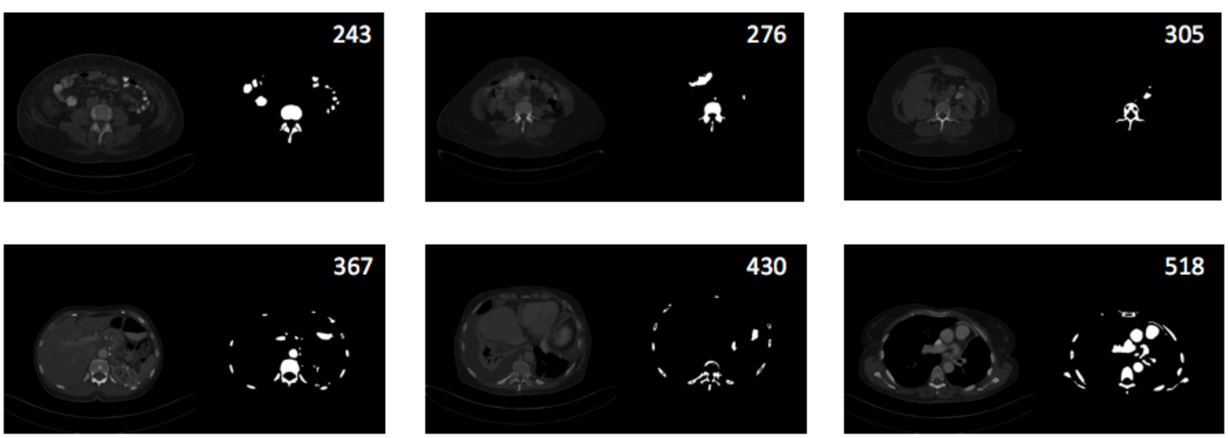


Figure T6: Automatically Segmented Slices of Patient 3. Note that while the spine is clearly segmented well, other anatomy has been included in the segmentation

Distance Weighting:
To improve our algorithm and get rid of incorrectly segmented anatomy, we first tried a radial distance weighting transform in 2D for axial slices. We tried it for a single slice in patient 3 (Slice 229) hoping that it would be possible to extrapolate the centroid positions in 3D based on tangent connections between centroids in a slice-based manner. Below details our distance transform component that we added to our fore and back weights in addition to the histogram intensities.

$$\boldsymbol{Distance\ Transform}$$
$$\boldsymbol{\alpha_1 e^{-\alpha_2\|x-c\|}}$$
$$c\ is\ centroid\ position$$
$$\alpha_1 - scaling\ parameter$$
$$x\ is\ pixel\ location$$
$$\alpha_2 - exponential\ parameter$$

After adding the distance transform component to the weights, we tested various parameter values to determine how it would affect the segmentation. The results of the parameter sweep of α_2 are summarized in Figure T7. Overall, with a higher α_2 component, the amount of inccorecty segmented anatomy was reduced. However, if the $\alpha_2$value became too large, parts of the spine were no longer segmented. An α_2 value of .03 appeared to be a suitable choice for segmentation of this slice.

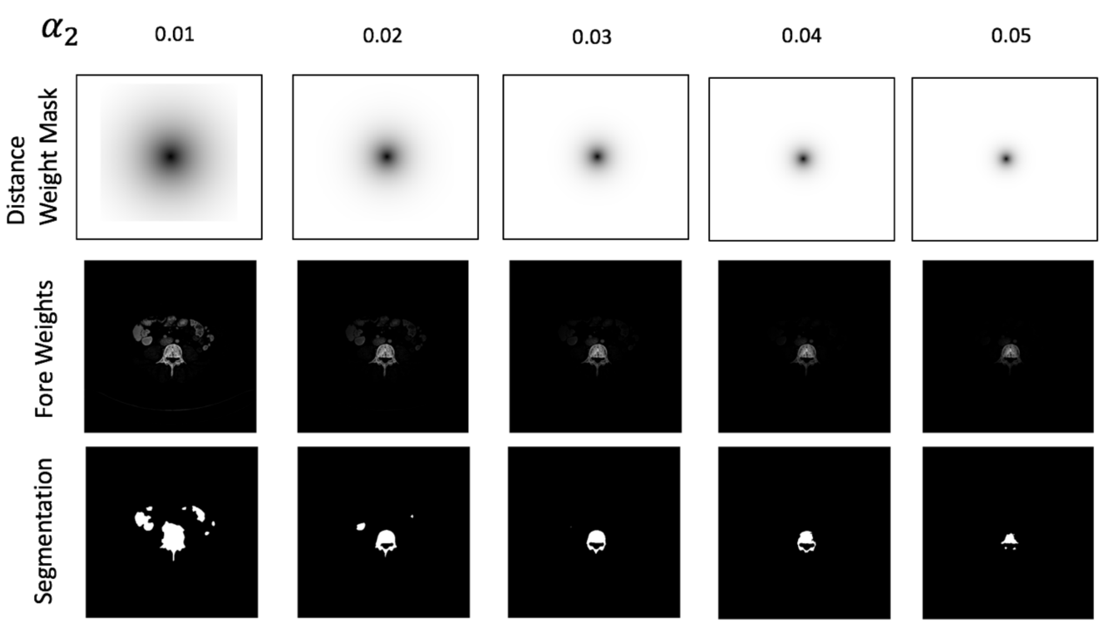
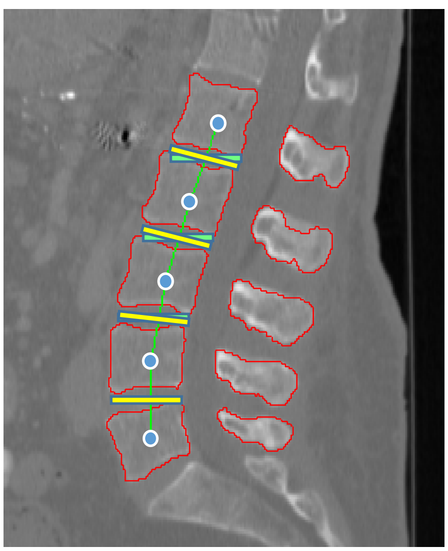


Figure T7: Parameter sweep for α_2 with distance weighting

3D Max Flow / Min Cut Optimization:
After adding distance weighting to our algorithm, we then were tasked with moving from a 2D slice-based approach to a 3D implementation of Max Flow / Min Cut. Adapting our 2D distance

weighting function to a 3D approach had two options. The first option was to model the spine as a series of spheres where the center is the centroid and the distance transform would decay the further a voxel was from the centroid. The second option was modeling the spine as a cylinder where the axis would be formed by the vectors connecting adjacent centroids. Here the distance transform would decay radially from the axis of the cylinder and the top and bottom of the cylinder would be capped with semi-spheres. We chose to purse the 3D method segmentation of the N20 with a cylindrical model as we decided the cylinder, rather than the sphere, better approximated the morphology of the spine.

The limitation of using a cylindrical method was that it would segment between centroids in the inter-disc space. This was the fault of the model assuming the spine was a continuous cylinder instead of a series of cylinders. To address this problem, a convolutional method was adapted that could move along the axis of the cylinder and detect locations where the spine transitioned to inter-disc space. This was accomplished with the knowledge that the inter-disc space was often populated with lower intensity values than the vertebral bodies, allowing the convolutional method to discriminate between the two locations. However, the convolutional cylinders that measured the intensity values along the axis, had to be angled orthogonally to the vector connecting adjacent centroids. This allowed for detection of inter-disc spaces that were not necessarily in the axial plane. An example of this axial to orthogonal orientation manipulation can be seen in Figure T8.



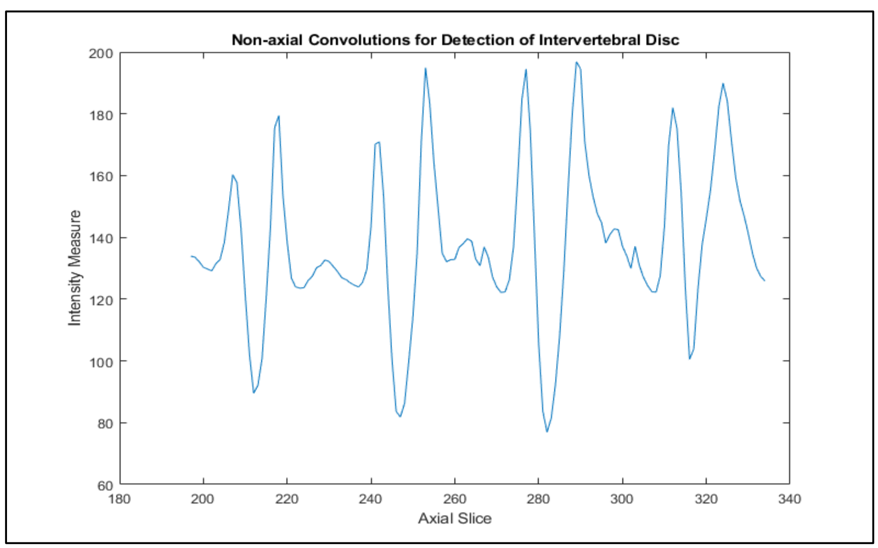


Figure T8: Axis connecting centroids locations. Axial orientation seen in green. Orthogonal Orientation shown in yellow.

Figure T9: Output of the non-axial convolutional inter-dic detection method. Large dips indicate the axial slice location of the interdisc space

Upon using the inter-disc detection, each axial slice is given a weight. Figure T9 shows an example of the inter-disc detection for 150 slices. The inter-disc space may be identified by two sharp peak, representing the cortical bone caps, and a large dip in between, indicating the location of the inter-disc space. However, this knowledge must be applied to the weights or the segmentation will not change. A crude way of applying this method, which we employed, was to multiply the detection output by the fore weights. Thus low peaks, indicative of inter-disc space, would lower the fore weights at inter-disc space locations. Causing them not to be segmented. Future work would include how to more accurately apply the knowledge of inter-disc space to ensure they aren't segmented while using a cylindrical model.

N200 Implementation:

We adapted our 3D-graph cut implementation to the N200 dataset, but not without facing certain challenges. In terms of positives when comparing to the N20 dataset, many of the Spine CTs in the N200 did not have the aorta and had clean gradients that separate bone from background. However, the N200 dataset had extremely noisy images in comparison to the N20 dataset. More importantly, it had many different parts of the spine (Lumbar, Thoracic, Full Spine) whereas the N20 just contained lumbar. Therefore, significant changes to parameter values and weight definitions for the algorithm are necessary for accurate segmentation.

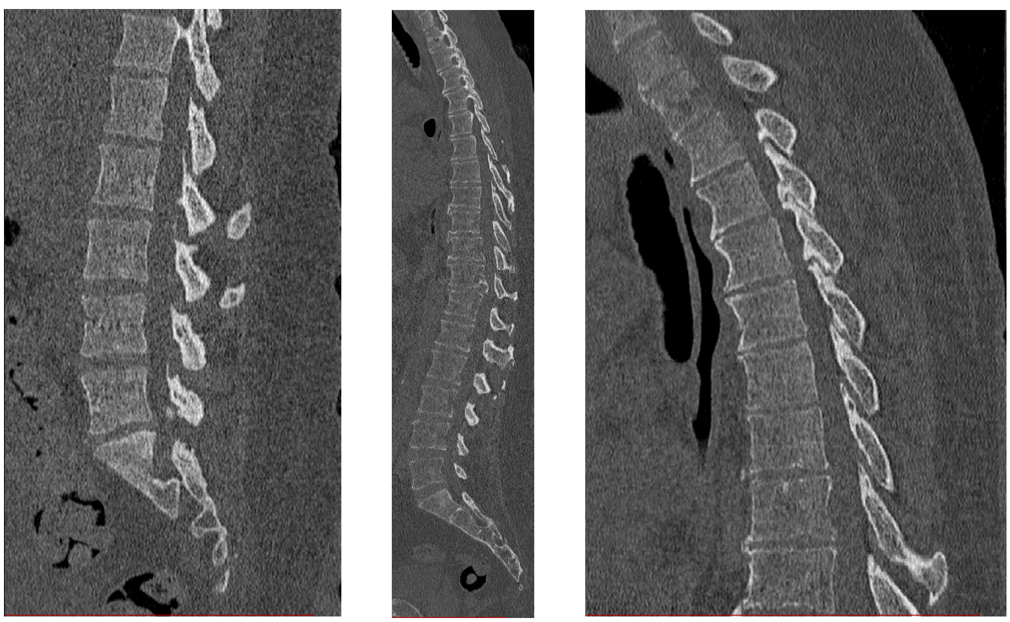


Figure T10: Various N200 Patients

Unfortunately, only a single N200 CT image (Spine 0018) had been manually segmented. Our mentors encouraged us to segment more images in order to validate our model which we are still in progress with in current time. In order to define weights in the network, we could no longer rely on manual segmentation as a way to have intensity profiling of the spine versus the outside. Therefore, we developed an automatic intensity profiling function that created a box mask around the centroid of the vertebrate within the image. In combination with spherical distance weighting, we ran our algorithm on Spine 0018 (Figure T11) and Spine 0001 which had abnormal and regular morphology respectively. For Spine 0018, we could quantify validation and achieve a dice coefficient of 0.6545 and RMSE of 4.85 voxels. For Spine 0001, we could not validate; however, visually Spine 0001 looked segmented well (Figure T12).

Figure T11: Abnormal Morphology of Spine 0018. A combination of Blur in the CT image with patient scoliosis

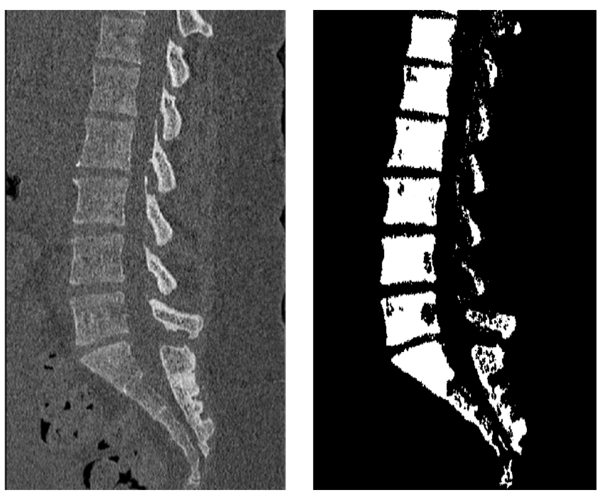


Figure T12: Segmentation of Spine 0001 Although we cannot verify quantitatively without an existing segmentation. The Automatic Segmentation looks well-defined capturing interdisc space well and vertebrate.

Development of Quantitative Metrics:

Two metrics were used to assess the accuracy of the segmentation output, dice coefficient and root mean square error (RMSE). Both metrics require an underlying knowledge of the “true” spine and thus the metrics mostly saw use when segmenting the N20 dataset. Dice coefficient ranges from a value of 0 to 1. An output of 0 indicates absolutely no overlap where a value of 1 indicates a perfect segmentation. RMSE measure the distance from the outside of the segmentation to the outside of the "true" spine.

Dependencies:

Dependency	Plan to Resolve	Date Expected	Contingency Plan
Access to I-STAR Lab	Gain Access	Completed	-
Workstation / MATLAB	Gain Access / Download	Completed	Remote access using TeamViewer
TCIA Collection (N20 and N200 Datasets)	Obtain from Mentors	Completed	-
Existing Generalized Implementation	Obtain from Mentors	Completed	-
Max Flow / Min Cut Segmentation Method	-	-	Consult Mentors & Explore alternative segmentation methods
Mentor Scheduling	Consult Mentors	Completed (Bi-weekly meetings)	Skype / Accommodate with remote meetings
Centroid Positions of N20	Obtain from mentors	Completed	-
Centroid Positions of N200	Obtain from mentors	Completed	-

Our project did not have any challenging dependencies since it was mostly a research and algorithm based project instead of a design based project. We had easy access to mentors, and as our project progressed we added only two dependencies: the centroid positions of N20 and N200 which we easily obtained after our first meeting.

Management Summary:
In the beginning of the project, Ben developed the quantitative validation metrics while Niko implemented the slice-based algorithm with intensity profiles and distance weighting. After the initial few weeks, Ben and Niko worked almost entirely together on all parts. This occurred because of how much all the work ended up being related, but also because we had an unexpected limiting factor of a single workstation which made a lot of work difficult to parallelize.

Management Plan:
We went into lab together 2-4 times a week for 4-6 hours having daily interactions with our mentor Dr. de Silva to update him on our progress. Additionally, we had biweekly lab meetings with the "Spine Cloud" group of the I-STAR lab where we presented our progress.

Accomplished:
Over the course of our project, our project plan evolved due to certain factors and difficulties we faced.

The milestones we successfully completed are below.

1. Milestone 1: Implementation of the 2D Max-Flow/Min-Cut segmentation algorithm for the spine
   - Planned Date: March 9th
   - Expected Date: March 9th
   - Status: Completed
2. Milestone 2: Analyze Basic Parameter Sensitivity on Single N20 Case
   - Planned Date: March 23rd
   - Expected Date: March 23rd
   - Status: Completed
3. Milestone 3: Implementation of quantitative accuracy metrics (RMSE & Dice)
   - Planned Date: March 30th
   - Expected Date: March 30th
   - Status: Completed
4. Milestone 4: Use centroid to create axial cylindrical distance weighting
   - Planned Date: March 30th
   - Expected Date: March 30th
   - Status: Completed
5. Milestone 5: Implementation of the 3D Max-Flow/Min-Cut segmentation algorithm for the spine
   - Planned Date: April 6th
   - Expected Date: April 6th
   - Status: Completed
6. Milestone 6: Create non-axial cylindrical distance weighting
   - Planned Date: April 6th
   - Expected Date: April 6th
   - Status: Completed
7. Milestone 7: Identify intervertebral disc space using non-axial convolutional cylinder
   - Planned Date: April 13th
   - Expected Date: April 13th
   - Status: Completed
8. Milestone 8: Analyze Basic Parameter Sensitivity on entire N20
   - Planned Date: April 18th

- Expected Date: April 18th
- Status: Completed

9. Milestone 9: Apply Distance Weighting for N200
    - Planned Date: April 27th
    - Expected Date: April 27th
    - Status: Completed
10. Milestone 10: Automate Intensity Profiling for N200
    - Planned Date: May 8th
    - Expected Date: May 8th
    - Status: Completed

To Be Done:

11. Milestone 12: Manually Segment Patients in the N200 dataset for quantitative Validation
    - Planned Date: May 15th
    - Expected Date: May 15th
    - Status: Working
12. Milestone 13: Accurately segment the N200 lumbar without spinous process
    - Planned Date: May 15th
    - Expected Date: May 15th
    - Status: Working
13. Milestone 14: Accommodate abnormalities in the N200 dataset
    - Planned Date: May 15th
    - Expected Date: May 15th
    - Status: Working

Additionally over the course of our project, our deliverables changed.

Original Deliverables

- **Minimum:** (Expected by March 9th) **Completed**
    1. Access accuracy of the N20 manually segmented validation dataset
    2. Implementation of 3D Max-Flow/Min-Cut segmentation algorithm for the spine
- **Expected:** (Expected by April 20th)
    1. Analysis of parameter sensitivity
    2. Evaluation of segmentation accuracy

3. Generation of a large N200 dataset for SpineCloud
4. Manual Segmentation of Patients in N200 for Validation **(In Progress)**

- **Maximum:** (Expected by May 15th)
  1. Method for patient specific parameter selection
  2. Method to accommodate spine anomalies

Updated Deliverables

- **Minimum:** (Expected by March 9th) **Completed**
  1. Access accuracy of the N20 manually segmented validation dataset
  2. Implementation of 3D Max-Flow/Min-Cut segmentation algorithm for the spine
- **Expected:** (Expected by May 6th) **In Progress**
  1. Analysis of parameter sensitivity on N20 **(Completed)**
  2. Generate and Evaluate Segmentation Accuracy on N20 **(Completed)**
  3. Automate Intensity Profiling for N200 **(Completed)**
  4. Manual Segmentation of Patients in N200 for Validation **(In Progress)**
- **Maximum:** (Expected by May 15th) **To Be Done**
  1. Accurate Segmentation of N200 Lumbar CT (without Spinous Process) **(In Progress)**
  2. Address anomalies in N200 (spine instrumentation, cancer)

Difficulties / Lessons Learned:

We found that validation in a scientific field must be quantitative. Therefore, metrics like RMSE and dice coefficient must be used objectively in order to determine how well the algorithm is working. Additionally, we both became more familiar with working as a single part of a larger multi-scale and multi-year project within a lab setting. We had to learn how to navigate the expectations of different supervisors (Postdoc vs. PI vs. Class Professor). This experience was especially fruitful and played a role in how we designed our project timeline in order to reach goals.

Additionally, we learned about the difficulties with tuning algorithms for different datasets. In particular, algorithm accuracy is heavily dataset specific. While we learned how to parameter tune on the N20 dataset in an effective way, the N200 dataset had its own properties that made it difficult to translate our findings in an easy manner. Another difficulty we had was that the size of the data set in combination with only having one workstation made it difficult to parallelize workflow. In retrospect, we would have requested two workstations in order to work independently and achieve further progress.

Future Work

- Accommodate Irregularities in N200
- Patient Specific Parameter Selection
- Employ cylindrical weighting in N200

- Manually Segment more of N200 for quantitative validation