

A Neurosurgery Patient Anatomy Application

Michael Gordon

CIS4914, Senior Project
Department of CISE
University of Florida

Advisor: Dr. Frank J. Bova, *email:* bova@neurosurgery.ufl.edu
Department of Neurosurgery
University of Florida, Gainesville, FL 32608

Abstract

Neuroimaging has allowed for doctors to diagnose patients and make decisions on treatment. Often patients must make life-altering decisions on treatment options. Due the nature of cancer in the brain, the side effects of a treatment option can vary. Doctors do their best to provide patients with the information they need to make these decisions. CT and MRI scans effectively communicate to medical experts what is going on in the brain, but it is not as clear to the average person.

With all this in mind, we decided to produce a pipeline to segment and visualize a patient's brain. This visualization can be presented to the patient to better help explain their condition and give them more information to make decisions on treatment options. In this report, we will discuss how set up server to process the brain, used FreeSurfer to segment the brain and used VTK to render it on an iPhone.

1.Introduction.

Neuroimaging technology is frequently used by medical experts to diagnosis and treat patients. Magnetic Resonance Imaging (MRI) and computed tomography (CT) scans allow for visualization of patient's brains. The brain scans can be used to spot abnormalities in the brain and guide neurosurgeons when deciding treatment options. However, these scans relay on medical expertise to understand. This report will summarize the methods used to transform 2D slices of a brain scan into a 3D volume. The software developed provides an interface for doctors to submit MRI T1 scans to go through the transformation. The results can then be visualized on an iPhone and displayed to patients.

1.1 Problem Domain.

Visualization of medical data is an area of high interest in computer science and neuroscience. 2D slices of MRI scans can be converted to 3D volumes. These 3D volumes can be displayed on a computer. Our pipeline aims to produce a segmented 3D volume from a series of 2D DICOM files. With that being accomplished we can visualize a patient's brain.

1.2 Previous Work (Literature Search).

In 2002, Insight Segmentation and Registration Toolkit (ITK) was released as an open source application development framework.[1] This software aids the medical community in developing tools that preform image segmentation and registration. Image segmentation is the process of breaking up an image into multiple segments.[2] These segments are often critical underlying structures in the image. Image registration is the process of taking information from several images and combining them.[3] In the context of our brain scans, a CT scan produces a series of DICOM files. Each of the DICOMs are a single slice of the brain scan. These images need to be combined via registration and then segmented anatomically in order to be visualized in 3D.

ITK assist with this process, but it does not perform a visualization. For that there exists the Visualization Toolkit (VTK). VTK is an open source framework that assisting can be used for developing 3D graphics, performing image processing and performing scientific visualization.[4] VTK provides a rendering abstraction layer for OpenGL. This allows for one to easy read and render different formats of data. VTK can easily render DICOMs and NIFTIs.

In our research we encountered two interesting pieces of software that would assist us in our challenge. First, we encountered quickNAT, this uses a convolution neural network (CNN) to perform automatic segmentation.[5] This software runs fast, the paper suggests the segmentation can be done in 20 seconds. However, the software requires on training data and it was pretrained on healthy brains. We would need to train the CNN with a large sample of brains with tumors.

The second software we found was FreeSurfer, an open source software suite for processing and analyzing brain MRI images.[6] FreeSurfer provides many tools to process brain scans. Key features that we need include skull stripping, image registration, sub-cortical segmentation, cortical segmentation and cortical surface reconstruction. The segments produced are too detailed for our needs.

From what we learned about these toolkits described, they are all very complicated and are the results of years of development. We decided it was best to make use of the existing frameworks from VTK and FreeSurfer and mold them to our needs.

2. Our Solution.

Considering the fact, we did not have access to a large number of scans with brain tumors, we decided against using a CNN. Instead we used FreeSurfer to do the intensive processing needed to produce our 3D Brain volume. Since FreeSurfer gives a wide variety of segments, we decided it was a good idea to group common structures into a single segment. With the file processed in a way we desired, we then focused on the task of rendering on the iPhone. We used VTK to achieve this goal. Lastly, we set up two servers, one to perform processing of data and the other to serve the data to the phone.

2.1 Using FreeSurfer.

FreeSurfer is a vast program that allows for all sorts of processing to be done on the brain. Our main interest was brain segmentation. In order for to use FreeSurfer, a file in the NIFTI format is needed. This file must be a weighted T1 scan. Since the pipeline was set up using Python, a small package dicom2nifti allowed for converting a series of DICOM files directly into one NIFTI. Using the recon-all command provided, this initializes a pipeline (pipeline stages shown in Appendix a).

The key steps in this pipeline are as follows: skull stripping, non-linear registration, volumetric labeling, intensity normalization, white matter segmentation, surface atlas registration, surface extraction and grial labeling. [7]

Skull stripping removes the skull from the scan as it is not a point of interest. With this out of the way the volumetric labeling can begin. The non-linear registration stage preforms registration. This registration takes care to accurately model deformations in the brain. Sub-cortical brain structures are labeled based on an atlas provided in FreeSurfer. Intensity normalization homogenizes the signal intensity of the white and grey matter to distinguish the two tissue types. White matter is then segmented. Surface atlas registration maps the subject's mesh to a sphere. This data is warped so it be aligned with the template data in FreeSurfer. Once this sphere is generated, it can be used to preform surface extraction which provides us with a 3d surface of the subject's brain. Lastly, grial labeling gives us the surface parcellation.

At the end of this pipeline the file `aparc.a2009s+aseg.nii` that contains a 3D segmented brain is obtained..

2.2 Our processing.

Additionally, we decided to implement a method to reduce the number of structures to fit our needs. This processing is done by sampling using a lookup table provided by FreeSurfer and a look up table we created. We do a many to one mapping of the voxel data from the FreeSurfer table value to a value on our table. This effectively merges the segments for us.

2.3 VTK on iOS.

After obtaining a volume, rendering of the brain was possible. In order to do this VTK 6.3 on iOS was used. VTK allowed us to set up a renderer and add volumes to that renderer. First, we tried adding each brain segment as its own volume, allowing us direct control over the opacity and color. Since we had many segments, this was difficult for the phone to handle. We decided we must use one NIFTI and choose the colors for segment via a color transfer function. Each voxel has an associated value, and we can map the data of each voxel to a specific color. Using our label map, voxels were set to particular values to correspond to particular labels. We mapped each of those values to a color. Now that all of our segments were colored, we needed to add for a way to interact with the volume. We decided to add buttons that would allow for the 'adding' and 'removing' of segments. The button would make the segment disappear by changing the opacity of the segment to 0. Wireframes of the app can be seen in appendix B.

2.4. The Servers.

In order to facilitate the transfer of data to the phone, two servers were set up. One for serving the results of processing to the phone and another for receiving requests to process data.

The processing server listens for POST requests that upload an archive file containing DICOMs. These DICOMs are converted to niftis and sent to the FreeSurfer pipeline. After that our processing is done (as outlined in 2.2). After this the segmented brain is ready to be sent to the medical server. The medical server can also make requests to get the status of a processing study. By checking the status of the processing, the physician can get data about whether the data sent was adequate, if processing has started, and if the data is available it can be sent to the medical server.

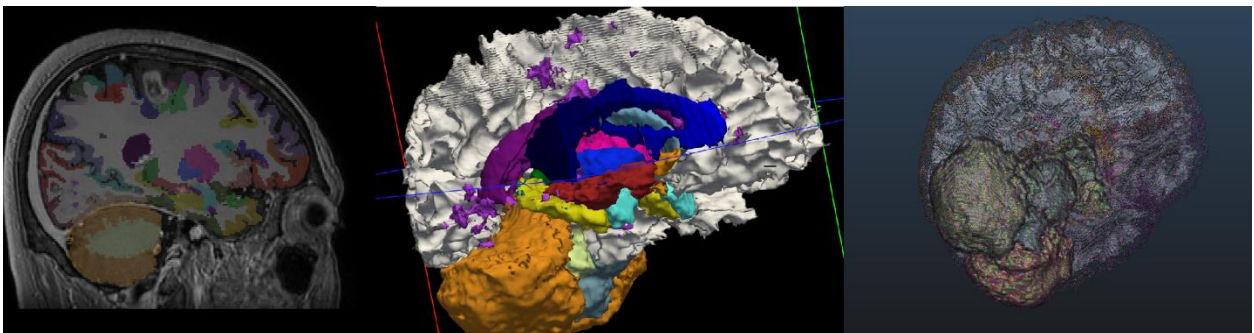
Since recon-all can take several hours and program crashing is possible, great care is taken to keep track of the processing stage of recon-all. By accessing the log file, the server can restart recon-all at the appropriate stage in the pipeline. Studies can be processed in parallel. Following the guidelines on FreeSurfer's website, the server is set up to run several recon-all's.[8] The limit is set to 3GB per process and 1 core. If a lot of cores are available, multiple cores can be mapped per recon-all process. Both servers communicate using API tokens for authentication.

The file server is set up to accept logins from the client. After logged in the client can request a list of studies it has access to. Using GET requests, the client can request a study. If the data has been processed, the 3d volume can be sent to the brain and rendered on the phone. POST requests to this server can be made to add a new study and upload dicoms to be processed. These DICOMs will be sent to the processing server. New data can be submitted for a study in the case that invalid data was submitted before.

3. Results.

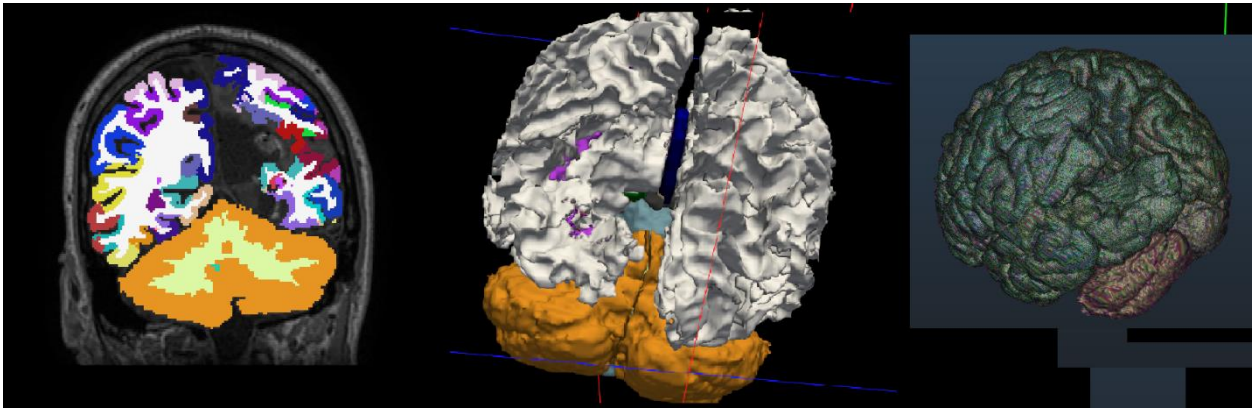
Our pipeline successfully produced several segmented brains. Here are two of our sample data visualized. This data comes from The Cancer Imaging Archive.[9] On the left is the data visualized using FreeSurfer's freeview. On the right is our result visualized using VTK.

Subject 1:



Here we show a slice of the original series with the FreeSurfer segmentation on top. Then is a 3d view of the brain from Freeview. The last image is our final result displayed using VTK. This segmentation took approximately 9 hours.

Subject 2:



This example seems particularly interesting as there is a hole in the segmentation where an abnormality occurs. As can be seen that hole translates to the 3d models as well.

These scans took a significant amount of time to process. It turns out that large tumors are hard for FreeSurfer to process. In one scan the program ran for 30 plus hours and did not complete. It is advised that for these sorts of scans that the segmentation of the white matter be manually edited to correct these defects in order to produce a segmentation.

4. Conclusions.

Using the FreeSurfer pipeline, our team produced segment volumes for display on the iPhone. Overall, this was a very research heavy project. A lot of time was devoted to understanding the process of producing an accurately segmented brain. Knowledge of registration and segmentation techniques was required to know what pipelines best suited our needs. This project also gave us experience working with rendering 3d models. VTK provided an easy to use framework that allowed us to rapidly deploy our application.

In the future efforts can be re-directed towards using a faster segmentation pipeline. With approval from the IRB, access to many patient data could provide a CNN with the appropriate training needed to accurately segment cancer patient's brains.

Standards and Constraints

Standards: All medical images start in the DICOM format and the NIFTI standard is required for FreeSurfer. VTK 6.3 ,FreeSurfer 6.0, Python 3.8, and Django were used to deploy our application.

Constraints: The servers were developed to run on Ubuntu 18.04. The app was developed for iOS.

5. Acknowledgments

Special thanks to our advisor Dr. Frank Bova for supervising this project. His guidance and medical expertise were pivotal in allowing us to approach the problem. Thanks also goes to Kyle Scott for leading the project and providing additional medical knowledge. Thanks to Dr. Tina Kapur and Bruce Fischl from Harvard for putting us in the right direction with FreeSurfer.

6. References.

- [1] M. McCormick, X. Liu, L. Ibanez, J. Jomier, and C. Marion, "ITK: enabling reproducible research and open science," ITK: enabling reproducible research and open science, 31-Jan-2014. [Online]. Available: <https://www.frontiersin.org/articles/10.3389/fninf.2014.00013/full>. [Accessed: 14-Apr-2020].
- [2] Y. Tan, "Image Segmentation," Image Segmentation - an overview | ScienceDirect Topics. [Online]. Available: <https://www.sciencedirect.com/topics/computer-science/image-segmentation>. [Accessed: 18-Apr-2020].
- [3] T. Fox, "Image Registration," Image Registration - an overview | ScienceDirect Topics. [Online]. Available: <https://www.sciencedirect.com/topics/biochemistry-genetics-and-molecular-biology/image-registration>. [Accessed: 18-Apr-2020].
- [4] W. Schroeder, B. Lorensen, and K. W. Martin, The visualization toolkit: an object-oriented approach to 3D graphics. S.l.: Kitware, 2006.
- [5] A. G. Roy, S. Conjeti, N. Navab, and C. Wachinger, "QuickNAT: A fully convolutional network for quick and accurate segmentation of neuroanatomy," NeuroImage, vol. 186, pp. 713–727, 2019.
- [6] FreeSurferMethodsCitation - Free Surfer Wiki. [Online]. Available: <https://surfer.nmr.mgh.harvard.edu/fswiki/FreeSurferMethodsCitation>. [Accessed: 18-Apr-2020].
- [7] E. Grossner, R. Bernier, E. Brenner, and F. G. Hillary, "Processing stream overview of the FreeSurfer recon-all function used to extract gray matter volume," https://www.researchgate.net/figure/Processing-stream-overview-of-the-FreeSurfer-recon-all-function-used-to-extract-gray_fig1_325432970
- [8] UserContributions/FAQ - Free Surfer Wiki. [Online]. Available: <https://surfer.nmr.mgh.harvard.edu/fswiki/UserContributions/FAQ#Q.HowcanIreducethetimeofrecon-allinagroupofpatients.3F>. [Accessed: 18-Apr-2020].
- [9] National Cancer Institute Clinical Proteomic Tumor Analysis Consortium (CPTAC). (2018). Radiology Data from the Clinical Proteomic Tumor Analysis Consortium Glioblastoma Multiforme [CPTAC-GBM].The Cancer Imaging Archive.

7. Appendices

Appendix A – FreeSurfer Processing Pipeline

Autorecon Processing Stages

1. Motion Correction and Conform
2. NU (Non-Uniform intensity normalization)
3. Talairach transform computation
4. Intensity Normalization 1
5. Skull Strip
6. EM Register (linear volumetric registration)
7. CA Intensity Normalization
8. CA Non-linear Volumetric Registration
9. Remove Neck
10. LTA with Skull
11. CA Label (Volumetric Labeling, ie Aseg) and Statistics
12. Intensity Normalization 2 (start here for control points)
13. White matter segmentation
14. Edit WM With ASeg
15. Fill (start here for wm edits)
16. Tessellation (begins per-hemisphere operations)
17. Smooth1
18. Inflate1
19. QSphere
20. Automatic Topology Fixer
21. Final Surfs (start here for brain edits for pial surf)
22. Smooth2
23. Inflate2
24. Spherical Mapping
25. Spherical Registration
26. Spherical Registration, Contralateral hemisphere
27. Map average curvature to subject
28. Cortical Parcellation - Desikan_Killiany and Christophe (Labeling)
29. Cortical Parcellation Statistics
30. Cortical Ribbon Mask
31. Cortical Parcellation mapping to Aseg

Appendix B – Technology Transfer

We intend to transfer all code over to the control of Dr Bova. It is advised that both servers be setup on separate computers as the processing server will require a lot of resources.

The processing server can be set up by extracting the processingServer directory to a desired location. On a Linux machine, FreeSurfer version 6 and python 3.6 with pip is required to be set

up on this machine. Running `pip install -r requirements.txt` will install all other necessary dependencies. The server can be configured by running `python setup.py`. After configuration it is advised to set up a superuser and run the server. An access token must be set up through the `/admin/` portal. This token should be stored in the `medicalServer/token.py` on the other server.

The medical file server is set up through the same process. Note this server only requires python 3.6 with pip. The token retrieved from this server should go in `processingServer/token.py`.

8.Biography.

Michael Gordon was born on March 20, 1996. He attends UF as a Computer Science major. He has experience working in C++, Java, Python, OCaml and JavaScript. His field of interest include operating systems, programming language development, and computer networking. Michael hopes to gain experience working in industry and to take that experience and start his own freelance programming company.