A Study of 3D Level Set Image Segmentation Based On Edge Detection

A Dissertation as a course requirement for

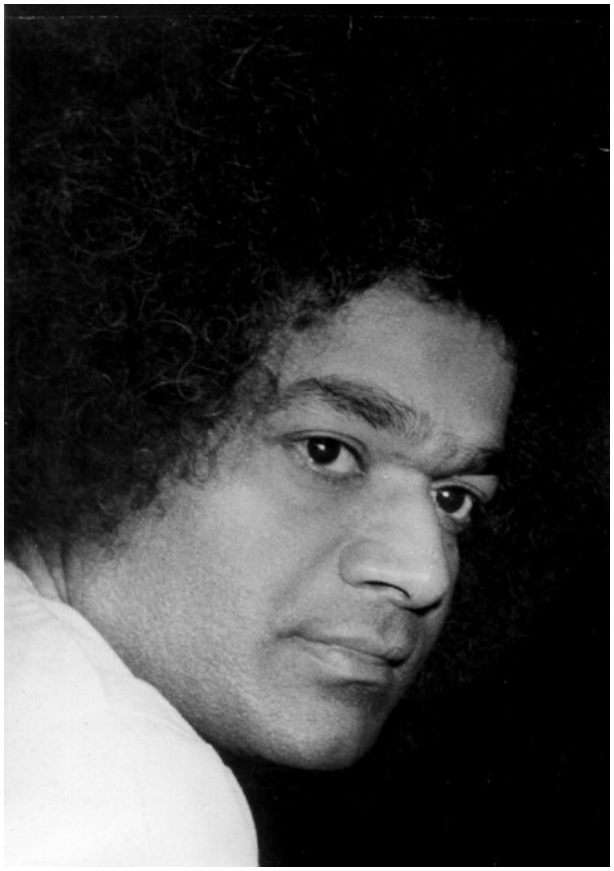
Master of Sciences (Mathematics)
with Specialization in Computer Science

Ankit Anand
(Reg.No.: 16002)

SRI SATHYA SAI INSTITUTE OF HIGHER LEARNING
(Deemed to be University)

Department of Mathematics and Computer Science
Prasanthi Nilayam Campus

March 2018



Dedicated to The Ultimate Master
and
to my loving parents...

DEPARTMENT OF MATHEMATICS AND COMPUTER SCIENCE

Sri Sathya Sai Institute of Higher Learning

(Deemed to be University)

Prashanthi Nilayam Campus, A.P.-515134

Certificate

This is to certify that this Dissertation titled **A Study of 3D Level Set Image Segmentation Based On Edge Detection** is a bonafide record of the original work done under my supervision as a Course requirement for the Degree of Master of Science (Mathematics) with Specialization in Computer Science.

Dr. N. Uday Kiran
Dissertation Supervisor

Place: Prashanthi Nilayam
Date: 26^{th} March, 2018

Dr. Pallav Kumar Baruah
Head of the Department

Declaration

The Dissertation titled **A Study of 3D Level Set Image Segmentation Based on Edge Detection** was carried out by me under the supervision of **Dr. N. Uday Kiran**, Department of Mathematics and Computer Science, Prasanthi Nilayam Campus, as a Course requirement for the Degree of Master of Science (Mathematics) with Specialization in Computer Science and has not formed the basis for the award of any degree, diploma or any other such title by this or any other University.

Place: Prashanthi Nilayam
Date: 26^{th} March, 2018

Sri Ankit Anand
Regd. No. 16002
II M. Sc. Mathematics
Prashanti Nilayam Campus

ACKNOWLEDGMENTS

First of all, I offer my gratitude and love to my beloved Bhagwan Sri Sathya Sai Baba, the divine founder chancellor of this hallowed University, for giving me an opportunity to study in this university. He is the source of all my inspirations to accomplish this thesis work successfully.

I express my deepest sense of love and gratitude to my parents Sri U. P. Yadav and Smt. Leena Devi for supporting me in all conditions and giving me a life that is fulfilling in every sense of the word.

I would like to express my most sincere gratitude to my supervisor Dr. N. Uday Kiran, for the useful comments, remarks and engagement throughout this thesis work. Without his help and care, this thesis work could not have been accomplished. I am deeply grateful for making this project work such an enjoyable experience.

My sincere thanks to Dr. Niranjan, warden of the Sri Sathya Hostel for Senior Students who allowed us to work late nights and took care of us in completing our thesis works.

My special thanks to Sri V.Srivarun for their indefatigable efforts, creative support and the excellent resources provided.

I would also like to express my gratitude to all my teachers of DMACS family, who have inspired me every moment. I would like to thank Sreedharan Sir for livening up many dull and mentally tiring days with discussions on the greater aspects of life and the world in general. I am thankful for his help and guidance throughout the project.

I take this opportunity to thank Shaarad and PK and all other peers for their constant encouragement and co-operation throughout my course and thesis work.

ABSTRACT

With the advancements in computing, we see a huge growth in the use of images. Volumetric or 3D images have become prevalent especially in the context of medical image processing. In the medical domain, one of the major steps is to perform image segmentation. Several methods have been proposed for segmentation based on partial differential equations, variational formulation and probability based approaches. Level set based methods is one of the tools used for image segmentation. We have implemented Canny Edge Detector based threshold level set framework for 3D image segmentation. We started with Canny edge detector but shifted to Laplacian as Laplacian edges gave better results. We have also evaluated the results using an available package for evaluation.

Keywords: *Canny Edge, Threshold Level Set Segmentation.*

Contents

List of Figures

Chapter 1

Introduction

Image segmentation has been of great interest in a number of fields such as CAD, medicine etc. A number of applications demand segmentation as an initial step. But how can we get segments out of an image? Also, there has been an increase in the use of 3D images for various applications and getting segments out of these images adds up to the problem of segmentation.
3D images are generally large in size. There are a number of formats available for such images. Some of them are $NIfTI$, $DICOM$ etc. Since these are not m-by-n *jpeg* or *png* images, we need special modules for their IO. Further, we need specialized softwares to work with them.
An edge in an image is defined as the location of sharp change in the image. They can be viewpoint dependent or independent. A viewpoint independent edge reflects the inherent properties of the objects present in the image. A viewpoint dependent edge reflects the geometry of the scene. Detection of edges can be a preliminary step in segmentation. Again, working in 3D adds up to the problem of detection of edges. This is because of the fact that the curve which forms the edge is a curve in 3D space. Also, if we consider the Canny edge detector, we see that gradient can be in any of the infinite number of directions possible in the 3D coordinate system. Categorizing the continuous gradient directions into a set of discrete directions is complicated.

1.1 Aim

Aim is to construct level set based 3D image segmentation pipeline based on edge detection and evaluate it using existing metrics. We also wish to incorporate this pipeline to the open-source software $3DSlicer$ for easy access.

1.2 Motivation

Image segmentation is one of the main steps in medical image processing. Before we apply any high-level processing, the image has to broken down into major structural components. For example, in a radiation treatment planning, radiologists have to trace the outlines of the critical structures in the image. First step towards segmentation is to capture and enhance edges. Generally, first derivative based methods are used for this. Some of the other approaches to finding edges are second-order based approaches and phase-congruency based methods. One of the famous detectors based on first derivatives is the Canny edge detector. This is a gradient based method with edge thinning and hysteresis to capture true edges.
For segmenting an image using level set region growing based methods, the edges can be used to restrict any further growth of region beyond what is the required.

1.3 Canny Edge Detection

A filter is a small array in which the values determine the nature of the process [6]. There are a large variety of filters. Some of are averaging, derivative-based and order-statistics based filters.
An edge in an image is a boundary describing a sharp change from one region to another. So, the general idea for the selection of edges is to use a derivative filter. Performance criteria for an edge detector should consider [2]:
(i) reliability of the detection

(ii) accuracy of localization
(iii) requirement of one response per edge
These criteria were proposed by Canny [1].
Reliable detection of edges means the probabilities of failing to mark real edges and falsely marking non-edge points should be low. As these probabilities monotonically decreasing functions of output signal to noise ratio, an optimal strategy as stated in [1] would be to minimize the noise. Effectively, this criterion points to the convolution of a filter to the image such that the noise is removed. The second criterion proposed by Canny was that the edges marked by the detector should track as closely as possible the true edges. This criterion effectively translates to use of a filter that can capture sharp changes in the image. The third criterion proposed by Canny was to make sure that there would be very low probabilities of falsely marking edge points and failing to mark true edges.
In view of the criteria for marking genuine edges, we perform Gaussian blurring for reducing noise, then apply derivative filter to indicate the changes and finally apply non-maximum suppression to obtain single response response for a given edge.
The algorithm consists of following steps:
(i) Apply the Gaussian filter to smoothen the image
(ii) Find the intensity gradient for each pixel of the image
(iii) Apply non-maximum suppression
(iv) Apply double threshold
(v) Track the edges by hysteresis
Gaussian filter is used to blur the image thereby removing noise from the image. This step is necessary in the sense that derivative filters are sensitive to noise. This filter will slightly smooth the image. The size of filter is important in different applications. Generally, larger filters lead to higher error in detection of edges. Also, larger filters decrease the detector's sensitivity to noise. So, there is a trade-off between sensitivity to noise and sensitivity to edges.

The next step is to find intensity gradient for each pixel of the image. Some of the common filters used are Sobel and Prewitt filters. If the first derivatives returned by any of the these filters are G_x, G_y and G_z, then the edge gradient i.e. magnitude of the gradient is

$$G = \sqrt{(G_x^2 + G_y^2 + G_z^2)} \tag{1.1}$$

$$\theta_x = cos^{-1}\frac{G_x}{G} \tag{1.2}$$

$$\theta_y = cos^{-1}\frac{G_y}{G} \tag{1.3}$$

$$\theta_z = cos^{-1}\frac{G_z}{G} \tag{1.4}$$

where θ_x, θ_y and θ_z are the angles that the gradient makes with the x-, y- and z-coordinates. We now need some method for edge-thinning. The algorithm specifies non-maximum suppression for edge-thinning. The basic idea is to suppress all pixels other than the local maxima. In this method if a pixel is part of an edge, the magnitude of gradient at that pixel must be greater than or equal to that of pixels in the direction of gradient and the direction opposite to the gradient.

The next step is double thresholding. This step is required since we will still have some responses present due to noise and color variations. If a pixel's gradient value is higher than the upper threshold, it is called a strong edge pixel. If a pixel's gradient value is smaller than the lower threshold, it is called a weak edge pixel. The weak edge pixels need to be dropped. So, the pixel values lower than the lower threshold are set to zero. The pixel values higher than the upper threshold are preserved.

By tracking edge by hysteresis, we mean that the pixel values in between the two thresholds are preserved only if they are connected to some strong edge pixel. This ensures that the resultant image will not have connectivity issues.

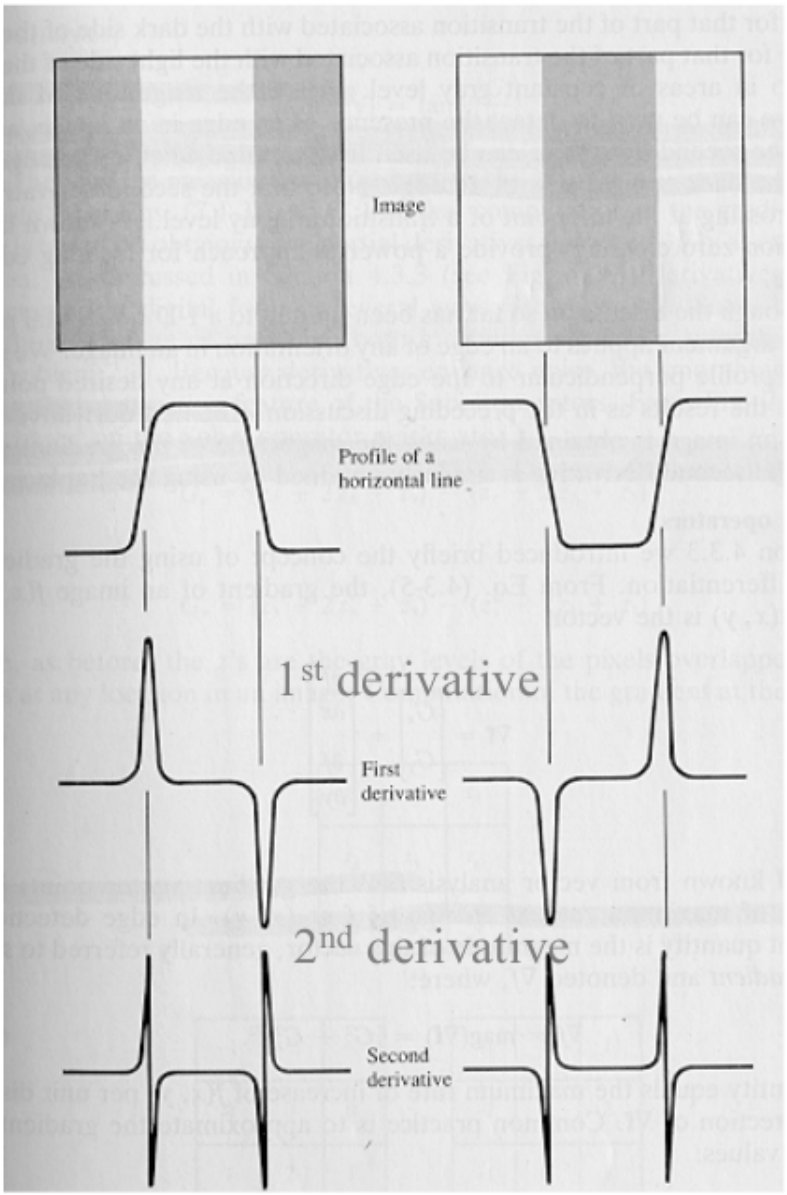


Figure 1.1: 1st and 2nd Derivatives []

1.4 Level Set Segmentation

Segmenting an image means dividing the image into its constituent parts [2]. There are a number of methods to segment an image. Some of them are clustering based, histogram based, edge detection based, level set based and region growing based methods. The result of image segmentation is a set of segments that collectively cover the entire image.

An interface is defined as the set of points that lie at the boundary of a region and thus introduces the notion of *inside* and *outside* of the region. In *explicit* representation of the interface, we either explicitly mention the points that belong to the interface or parameterize the interface and then discretize the parameter to approximate the interface. In *implicit* representation, we specify the interface as the isocontour of a function [5].

Similarly, there are two approaches to track the evolution of a interface:

(i) Lagrangian approach

(ii) Eulerian approach

In the Lagrangian approach, the evolution of a closed interface is tracked by parametric functions while in the Eulerian approach, the evolution is tracked using the concept of level sets.

The two methods have their own advantages and disadvantages. In the parametric or Lagrangian approach, one has to explicitly take care of the evolution if there happens to be any deformation and "*surgery*". Also, to detect whether a point belongs to a region, the method is that if any curve from that point to any point outside the region cuts the interface even number of times, it is outside the region while if it cuts the interface odd number of times the point is inside the region. Level set or Eulerian approach removes the problems of dealing with complicated deformations and topological repairs that happen during interface evolution. Further, we can determine whether a point is inside the region just by looking at the local sign of the implicit function. But, to store the interface we need to discretize the whole coordinate space.

Level set based methods play an important role in image segmentation. The general idea is to embed the evolving contour as the zero level set of a function. This function is then evolved under the control of a partial differential equation. A generic level set equation used in the literature[3] is

$$\frac{\partial \psi}{\partial t} = -\alpha \mathbf{A(x)}.\nabla\psi - \beta P(x)|\nabla\psi| + \gamma \mathbf{Z(x)}\kappa|\nabla\psi| \tag{1.5}$$

where A is an advection term, P is an expansion or propagation term and Z is a modifier term for the mean curvature κ. The scalars α, β, and γ are the weights for the respective influence of each term in the evolution.
In view of both frameworks discussed above, there are two ways to track the evolution of an interface. One way is to track an arbitrary point while the other is to track the whole contour. The first approach is called the Lagrangian approach. The second approach is known as the Eulerian approach. The following example is taken from [5]. Given an externally generated vector field $\vec{V}$ and we want to move all the points on a surface with this velocity, one way to do this is to solve the differential equation

$$\frac{dx}{dt} = \vec{V}(\vec{x}) \tag{1.6}$$

On the other hand, if the surface that we are dealing with is given by $\psi(\vec{x}) = 0$, the evolution can also be tracked by

$$\psi_t + \vec{V} \cdot \nabla\psi = 0 \tag{1.7}$$

Literature[5] says that if the evolving surface changes topology such as detachment and reattachment of boundary elements, evolution with Lagrangian formulation becomes complicated. Such is not the case with Eulerian formulation since we are not tracking elements here.

If the velocity is self-generated, we get the PDE

$$\psi_t + V_n|\nabla\psi| = 0 \tag{1.8}$$

where V_n is the normal component of $\vec{V}$. If the velocity is proportional to the curvature of the moving interface, we get

$$\psi_t = a\kappa|\nabla\psi| \tag{1.9}$$

where a is a constant. When $a > 0$, the interface moves in the direction of concavity. When $a < 0$, the interface moves in the direction of convexity. Then from equations 1.7 and 1.9, the convection-diffusion equation is

$$\psi_t + \vec{V}.\nabla\psi = a\kappa|\nabla\psi| \tag{1.10}$$

The only extra term that we have in equation 1.5 is $P(x)|\nabla\psi|$. In this term P is called the propagation image or speed image. It controls the propagation of level set through the coordinate space. The vector field A in the equation 1.5 attracts the evolving surface to the required features.
Finding the advection and propagation terms depend on the target the method one intends to use and the target application. Advection term is a vector field that attracts the evolving contour to the boundaries of the desired objects. Expansion term controls how level set propagates through the coordinate space. So, this term has higher values in regions where the surface can evolve quickly and close to zero values near the important features.
For implementation, user can be asked to provide initial level set and then the involved PDE can be solved iteratively with appropriate stopping criteria till the user gets satisfactory results.

1.4.1 Evaluation of Segmentation

Literature gives a number of ways to evaluate segmentation. The paper [8] gives a list of metrics that can be used for evaluation of segmentation of medical images. We discuss a few of them here.
Taha[8] discusses the use of True Positive Rate (TPR) also called sensitivity and False Negative Rate(FNR) also called specificity. Sensitivity measures the portion of positive pixels in the ground truth that are identified as positive in the segmented image. Specificity measures the portion of negative pixels in the ground truth that are marked as negative in the segmented image. Two other metrics similar to these are False Positive Rate (FPR) also called fallout and False Negative Rate.

$$Recall = Sensitivity = TPR = \frac{TP}{TP + FN} \tag{1.11}$$

$$Specificity = TNR = \frac{TN}{TN + FP} \tag{1.12}$$

$$Fallout = FPR = \frac{FP}{FP + TN} = 1 - TNR \tag{1.13}$$

$$FNR = \frac{FN}{FN + TP} = 1 - TPR \tag{1.14}$$

A volume based metric discussed in [8] is volumetric similarity. Volumetric similarity is defined as

$$VolumetricSimilarity = 1 - \frac{|FN - FP|}{2TP + FP + FN} \tag{1.15}$$

Intuitively, this metric, being a measure of similarity, is nothing but $1 - VD$ where VD is the volumetric difference. Here, VD is the ratio of difference in the volume of segmented image and the ground truth to the total volume of both segmented region and the ground truth.

Chapter 2

Implementations and Results

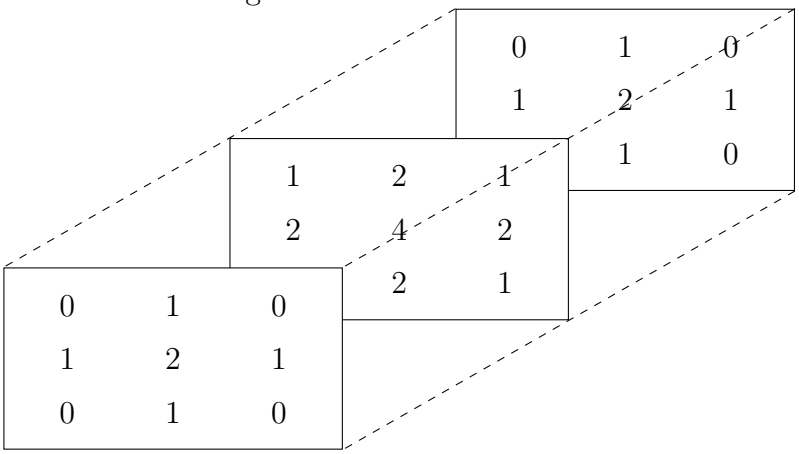
Our objective was to understand level set based methods for 3D image segmentation. To that end, we started with the implementation of Canny edge detector(CED) for 3D images. Then, we moved on to build a level set based pipeline for segmentation. We have also evaluated the segmented images. We have also added an interface to 3DSlicer so that the pipeline can be used easily. We have considered medical images for evaluation.

We have used C++ on Ubuntu 16.04 LTS with 7.7 GB memory, processor: $Intel^{®}$ $Core^{TM}$ i7-6700T CPU @ 2.80GHz 8, OS type: 64-bit.
We have also used the $ITK4.9$ [itk] for the header files required for the level set based implementations. We have also used $3DSlicer$ for visualization purposes.

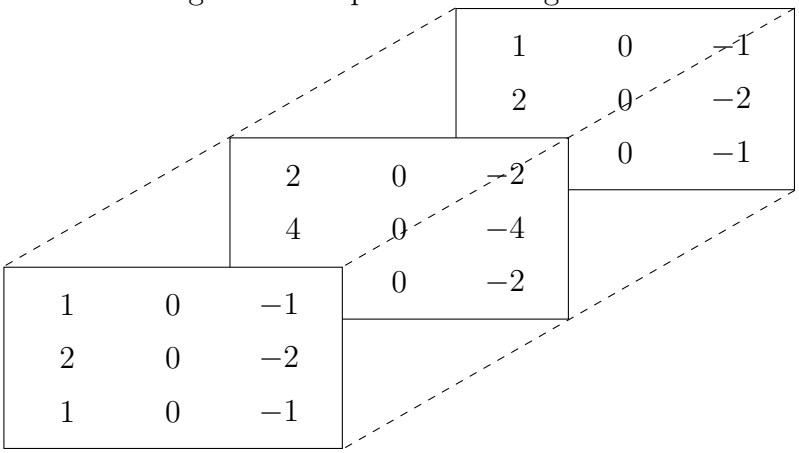
2.1 CED on DICOM Images

We have implemented the CED for 3D images using the algorithm for 2D given in [1]. The implementations are done using the ITK software. The testing is done on nrrd images. Each step mentioned in the pseudocode is implemented as an independent module(accepting arguments in the command-line).

Method The CED as mentioned in [1] involves modules for Gaussian blur, gradient, non-maximum suppression and double thresholding. We are using a variant of Gaussian filter to denoise the image. We consider the neighbourhood as a cube for each pixel with side 3 voxels. We used the following matrix for blurring:



The application of this filter causes unintended change in the intensity of each pixel. We therefore, divide the resultant value of each pixel by 48.
Next comes the intensity gradient module. Here, we used the following matrix for finding the x-component of the gradient:



Similar matrices are used for finding y- and z-components of the gradient. We have used the same neighbourhood as before and the pixels' values at the centre of each nieghbourhood is replaced by the magnitude of the gradient. The angles that the gradient makes with the three coordinate axes are cal-

culated using the cosines of the ratios of the corresponding magnitudes. The maximum error is that of $\pi/24$ with respect to x-, y- and z-axes.
In the non-maximum suppression module, a pixel value less than the value in the direction of gradient or in the direction opposite to the gradient is suppressed. The angles of the gradient with respect to x-, y- and z-coordinates are discretized to 12 directions. This discretization is a main source of error in this algorithm.
We then apply double thresholding to remove the remaining spurious pixels present because of noise or some colour variation. So, if a pixel's value is lower than the lower threshold, it is suppressed. If a pixel's value is larger than the upper threshold, it is preserved. A pixel's value lying in between the two thresholds is suppressed if it is not connected to a pixel with a value higher than the upper threshold.

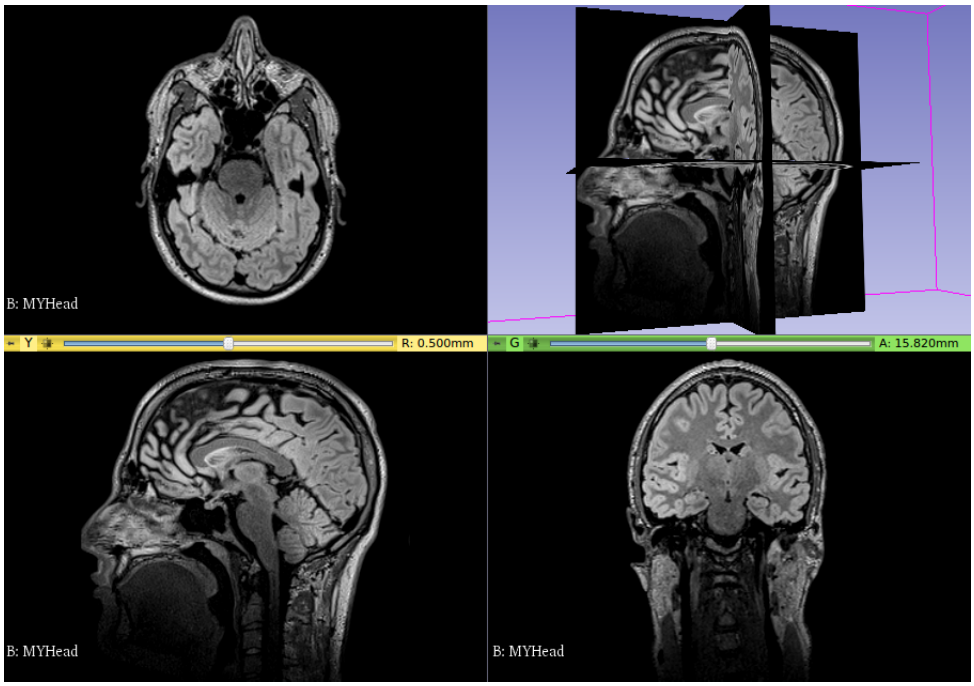


Figure 2.1: Input Image

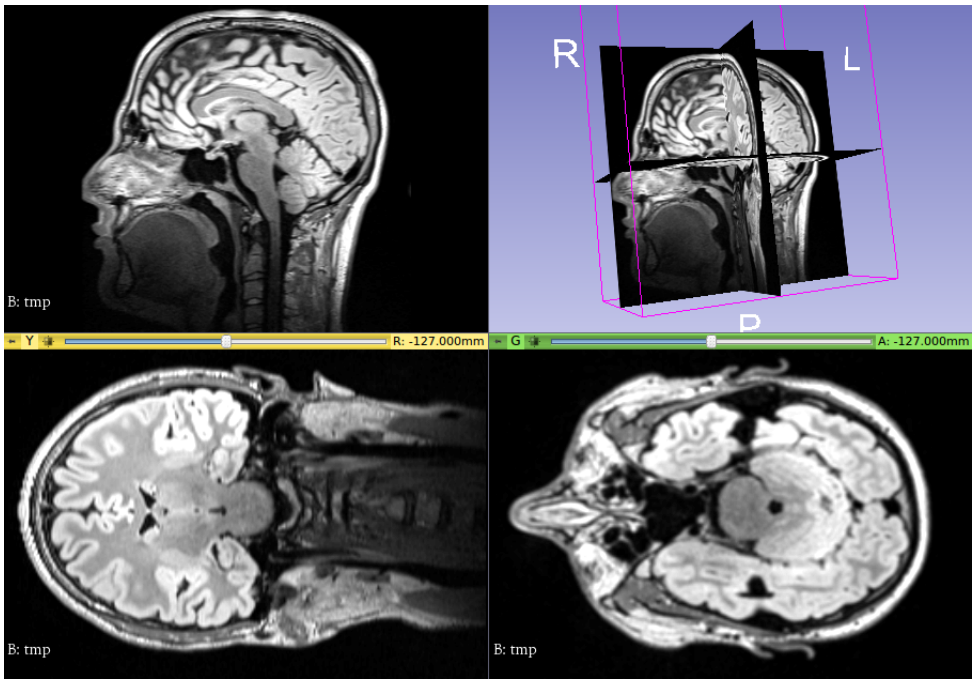


Figure 2.2: Result of Gaussian blurring

2.2 Thershold Level Set Segmentation

Method The level set pipeline is an implementation of threshold-connected component segmentation. The pipeline consists of:
(i) A FastMarching module to get the input level set for the main module.
(ii) Threshold level-set segmentation module to carry out the iterations.
(iii) A binary thresholding module to get binary output.
The PDE involved cannot be solved using any direct formulation. So, we have an iterative procedure as highilghted by (ii) above. In this pipeline, the curvature and the propagation terms are given a weight of 1.0 each. The

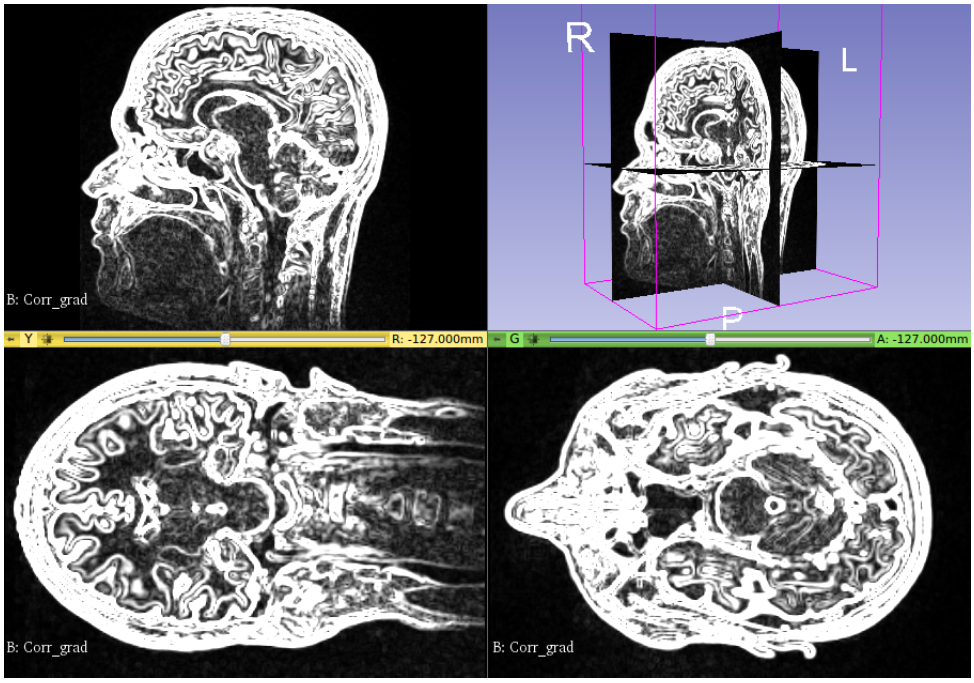


Figure 2.3: Gradient of Fig. 2.1

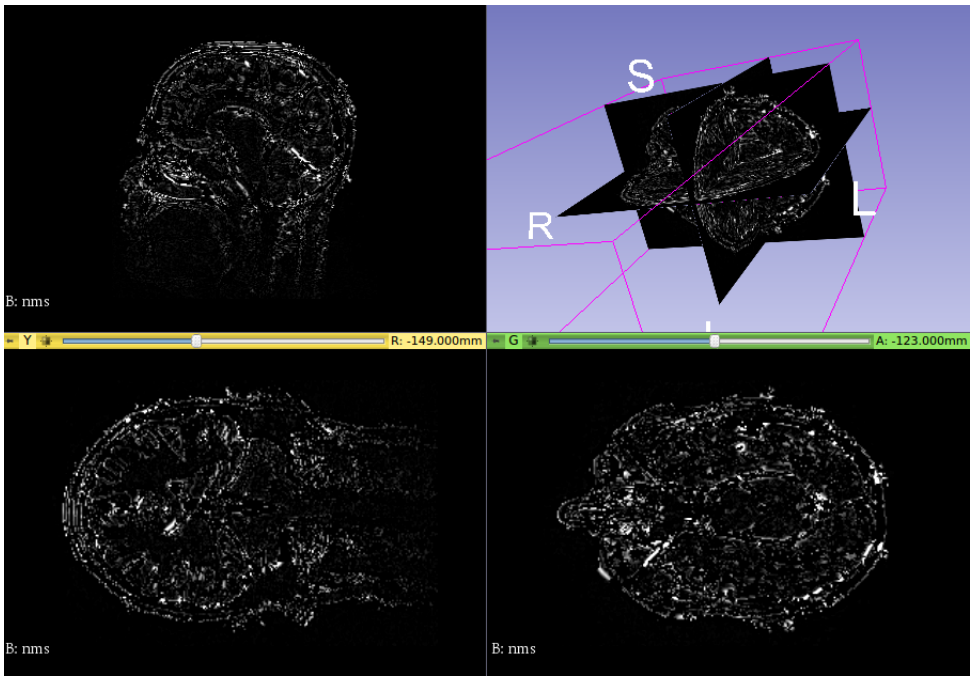


Figure 2.4: NMS of Fig. 2.3

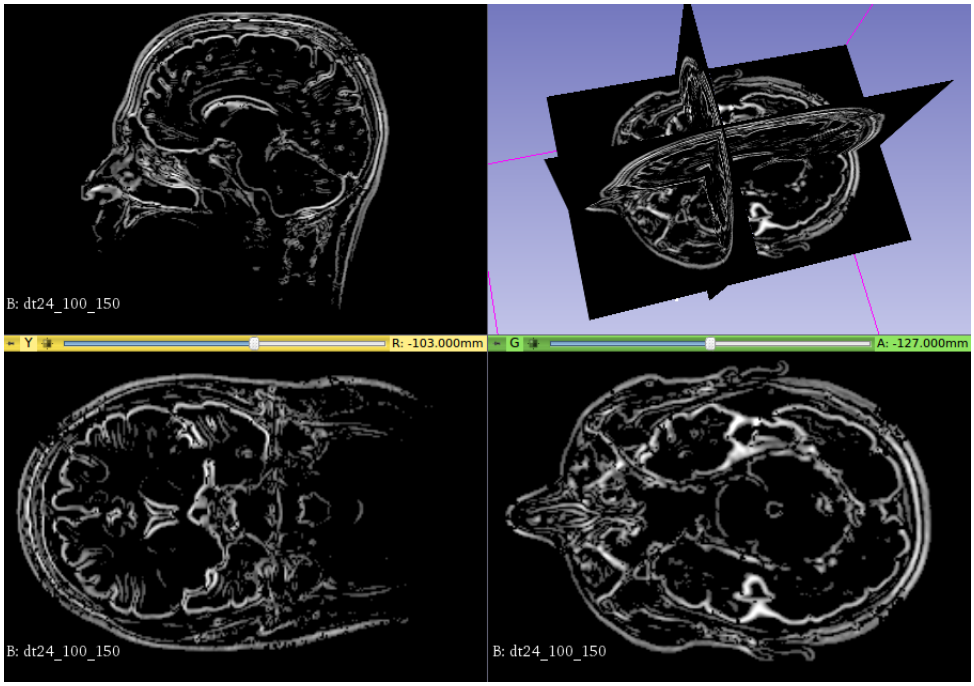


Figure 2.5: Edges of Fig. 2.1

FastMarching module is used to create initial level set from the seeds entered by the user [7]. The speed term is calculated in such a way that pixel values inside the region will be positive while pixel values outside the region will be negative

$$P(\mathbf{x}) = \begin{cases} \text{f}(\mathbf{x}) \text{ - L} & x \leq 0 \\ \text{U - f}(\mathbf{x}) & 0 \leq x \leq 100 \end{cases}$$

The calculation is depicted in the figure below. During the calculation of speed image, the edges were calculated using both Canny's single derivative approach as well as the second derivative approach [4]. The output of the threshold level-set module is a scalar image such that the pixels in the inside of the region of interest is negative while the pixel outside the region have positive values. So, lower and upper thresholds for binary thresholding module are -1000 and 0.

Since the implementation is iterative, we need some stopping criteria. The

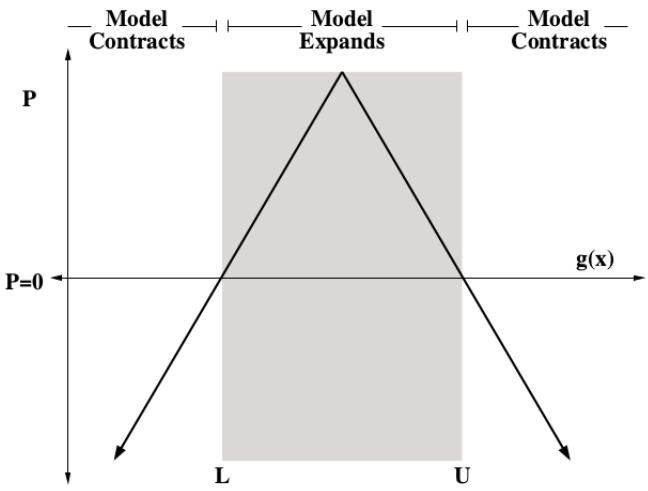


Figure 2.6: Calculation of Propagation term [3]

criteria used are:
(i) Maximum number of iterations.
(ii) Maximum RMS error.
The iterations stop if any one of these criteria gets satisfied. We have set maximum number of iterations to be 1500 and acceptable RMS error to be 0.005. The pipeline built is shown in figure below. We have found that the second-order derivative filter works better as it produces closed and rounded contours. Further, second-derivative filters give better localization when the edges are not very sharp.
The implementation requires that the user inputs four pixels (as seeds) from the region of interest. Multiple seeds prove to be very useful in the segmentation of a single connected region of the image. This can be made dynamic where the user first enters the number of seeds he wants to provide and then proceed to provide their details. This is because multiple seeds start evolution at different parts of the connected region thereby making the segmentation procedure fast. Also, multiple seeds can be used to segment disconnected regions in the image. The module added to the $3DSlicer$ also asks the user to provide exactly four seeds for segmentation.

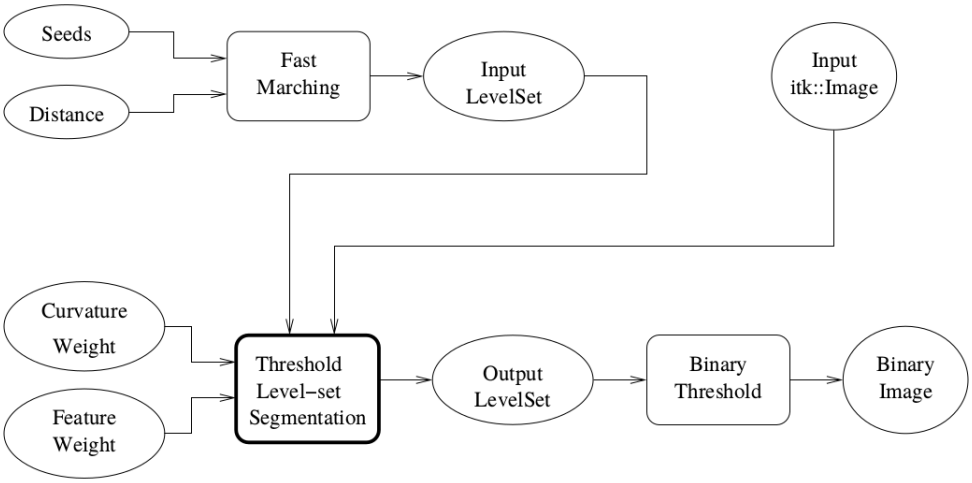


Figure 2.7: ThresholdLevelSet Segmentation Pipeline[3]

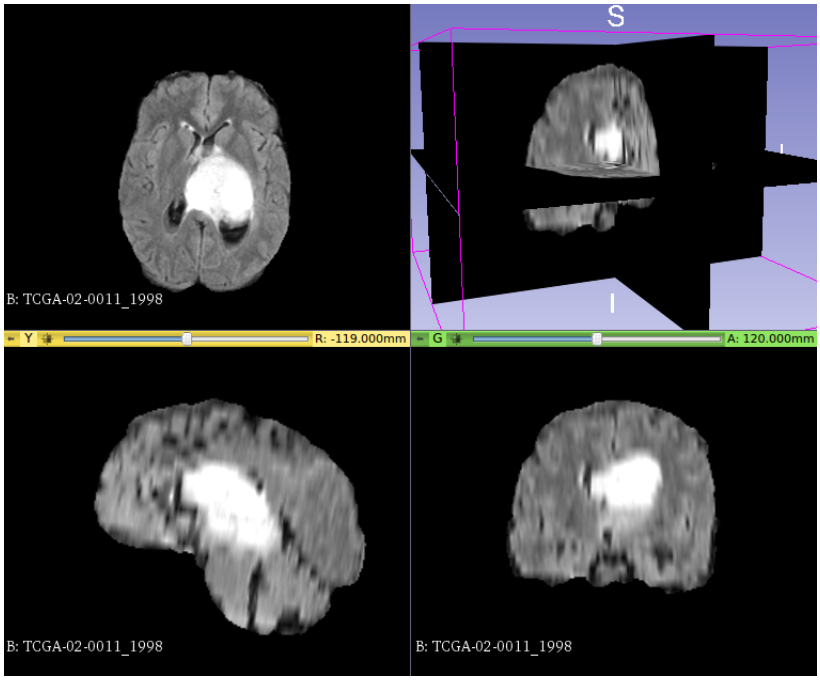


Figure 2.8: Sample input image for segmentation with highlighted ROI

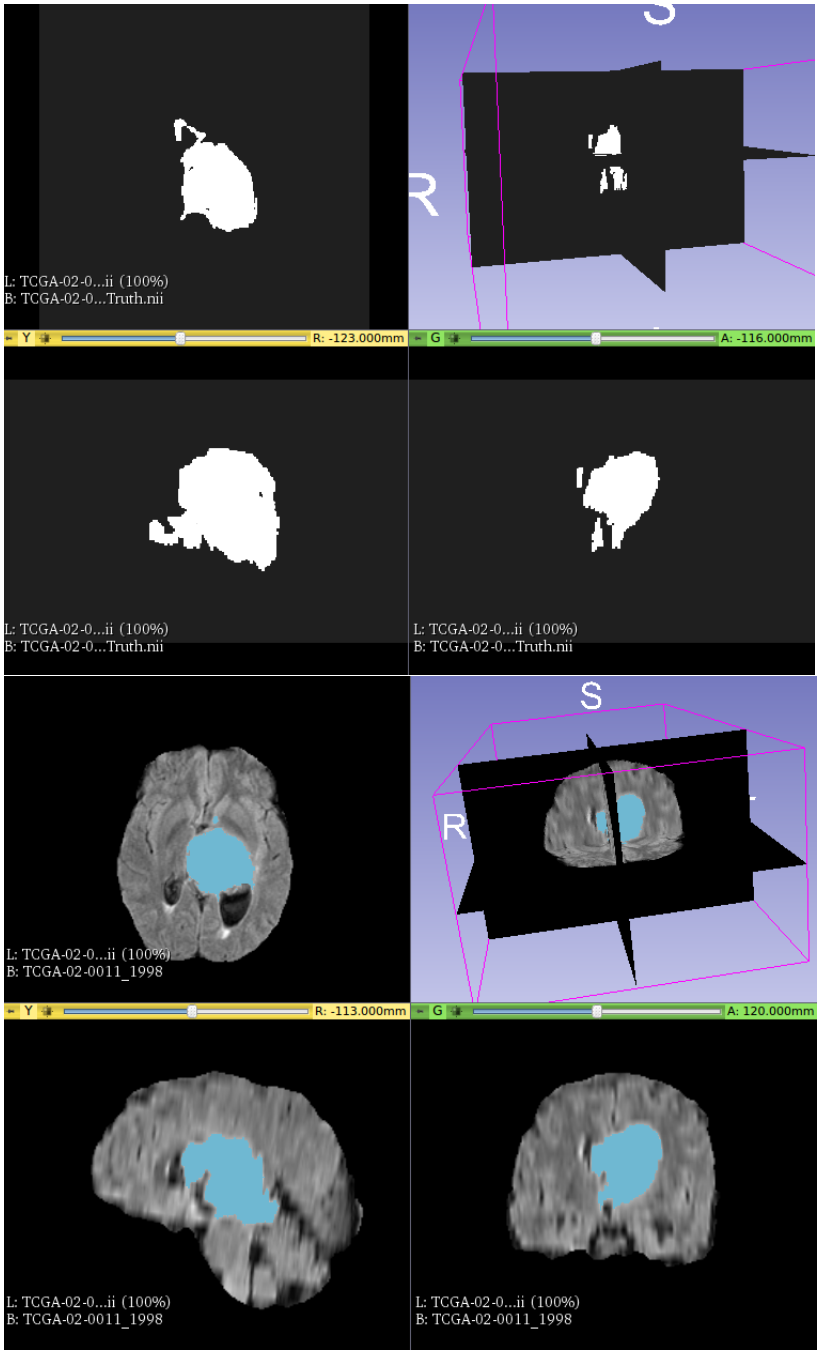


Figure 2.9: Segmentation of Fig. 2.8

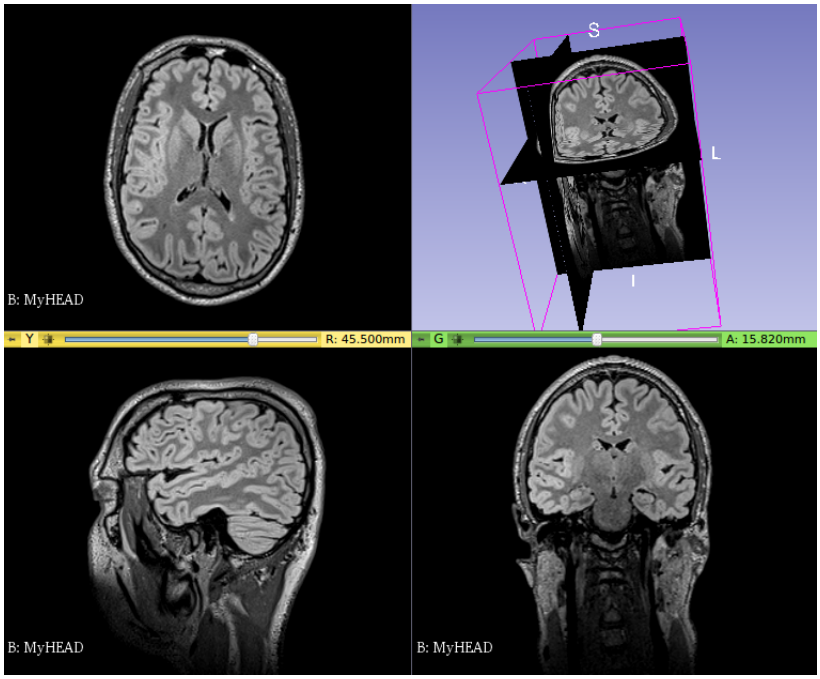


Figure 2.10: Sample input image for segmentation

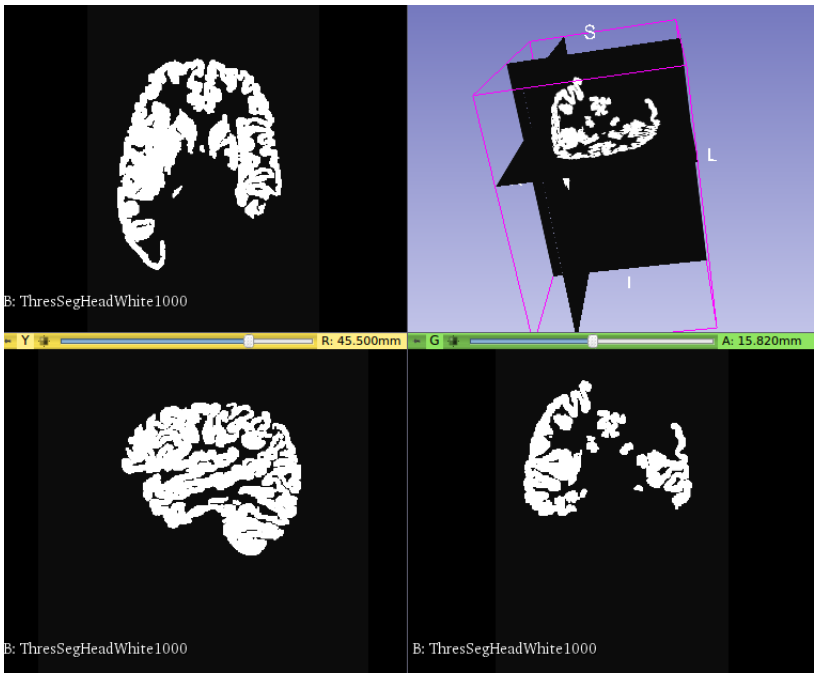


Figure 2.11: Segmentation of white matter from Fig. 2.10

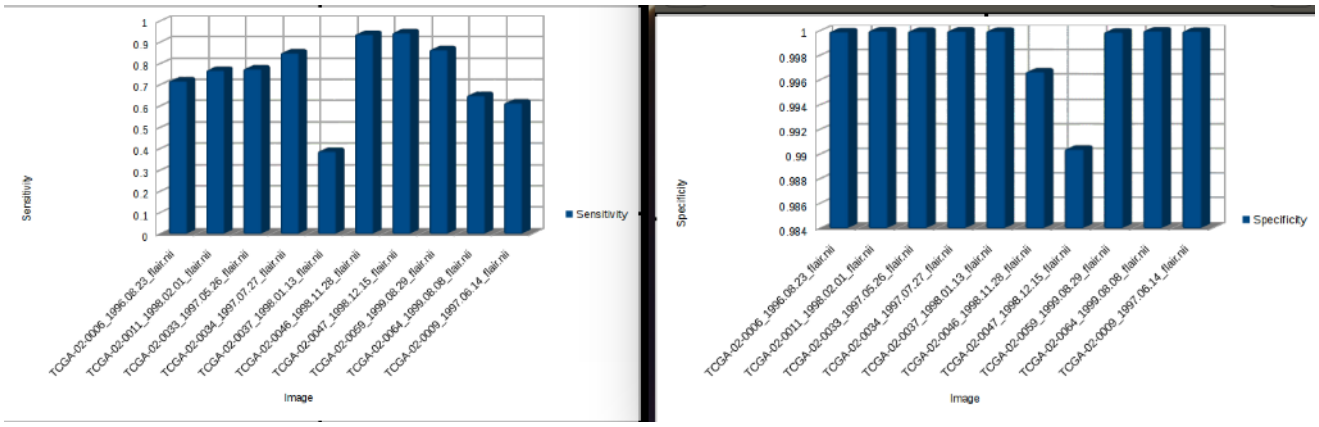


Figure 2.12: Sensitivity and Specificity Plots for 10 Segmentations

Dataset We used a 3D image data called $MyHEAD(.nrrd)$ for Canny edge detection. This is a scalar 256x256x176 image with pixel type short. Pixels' values range from 0 to 562.

For evaluation of segmentation pipeline, we used pre-operative $TCGA\_GBM$ dataset. $TCGA\_GBM$ consists mainly of 102 scalar NIfTI images with ground truths. We randomly picked 10 images for the evaluation. These consist of NIfTI images of size 240x240x155 with pixel type short.

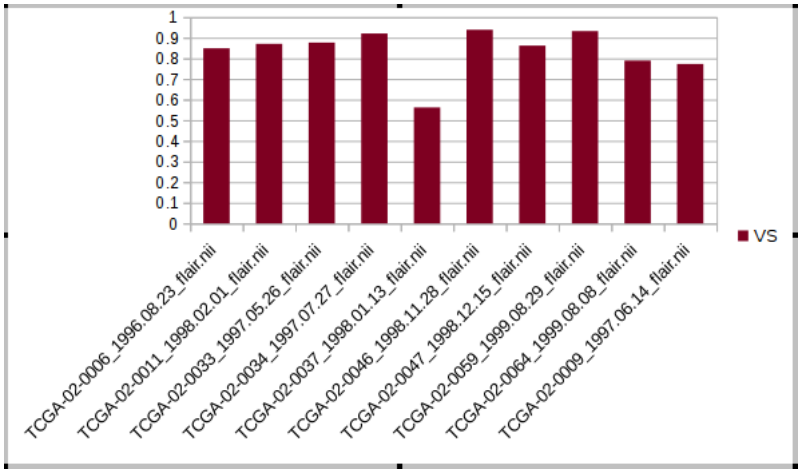


Figure 2.13: Volumetric Similarities of 10 Segmentations

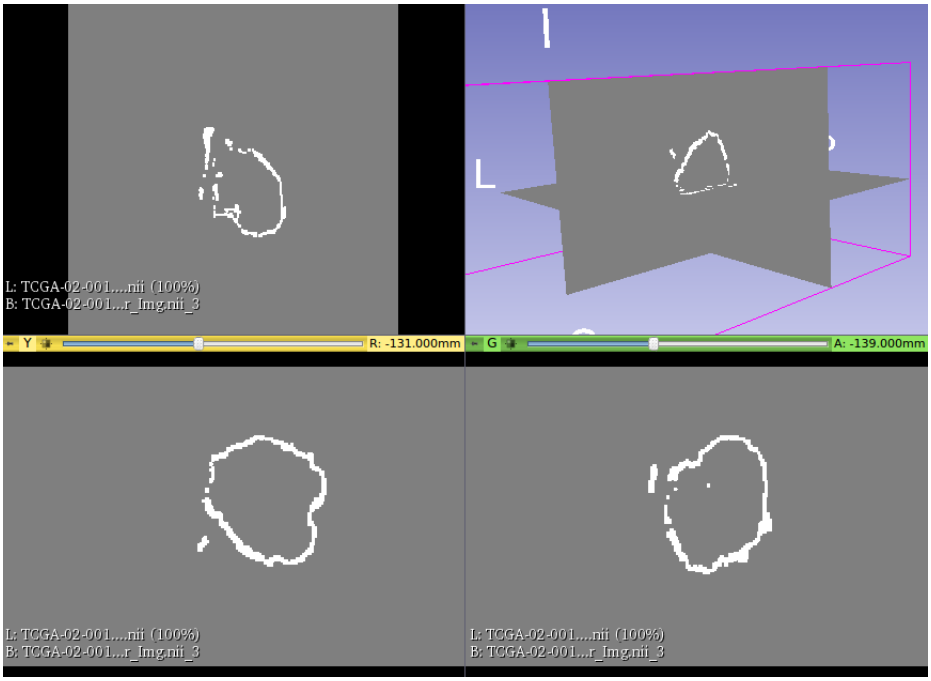


Figure 2.14: Error Image for Segmentation Fig. 2.9

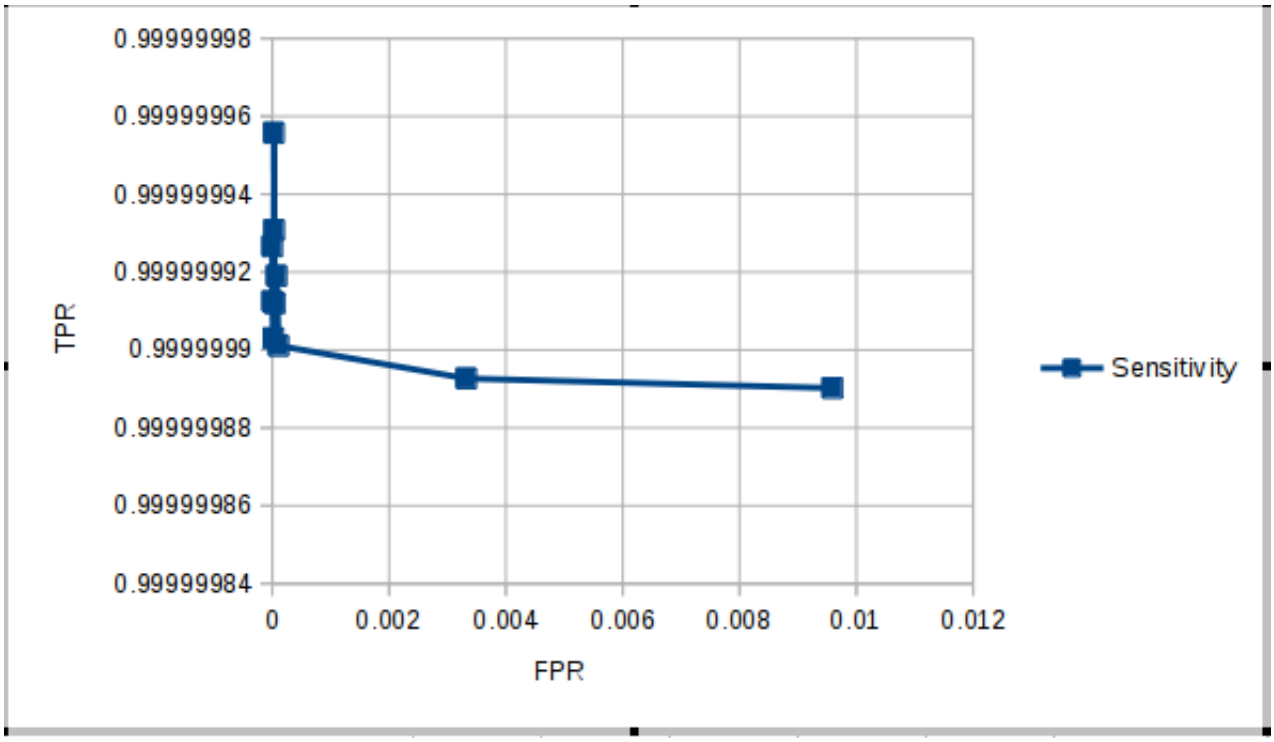


Figure 2.15: ROC curve for 10 segmentations

Chapter 3

Conclusion and Future Work

In many real world applications of image segmentation, an interactive environment is required. The user should be able to input data for segmentation, see the results and feed the inputs again for segmentation according to the quality of the results in a loop. Also, the implementation should be such that in the absence of some of the inputs (or when some of the inputs are improper), the algorithm should be able to use statistical tools to get some idea about the unspecified inputs and run the program to show some results. The use of segmentation in the medical field requires that the results should be highly accurate. Further, the implementation should also take into consideration easy display, storage and transfer of images in the network both at local as well as at global level.

As part of the work, we implemented Canny edge detector for 3D images. We built a threshold level set segmentation pipeline based on Canny edge detector for 3D images. We found out that Laplacian worked better than Canny's first derivative for finding edges as a intermediate step for level set based segmentation which is in line with theory. We also evaluated our segmentation using some existing metrics. We also integrated our implementation to $3DSlicer$ for easy access.

Future Work

- The implementation can be made more accurate by studying the effect of each of the terms in the level set equation considered. We also wish to look for other terms that can be included in the level set equation.
- The incorporation to the 3DSlicer can be made more robust by handling all types of user inputs possible. We can also add all the modules used for segmentation instead for incorporating the whole segmentation pipeline as a single module.
- The implementation can be optimized to a large extent by studying the cache complexity of each module. We can also parallelize the algorithms to improve the speedup. Moreover, it is possible to parallelize as all the computations involved are local in nature.

Bibliography

[1] J. Canny. A computational approach to edge detection. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 8(6), 1986.

[2] Pierre Kornprobst Gilles Aubert. *Mathematical Problems in Image Processing.*

[3] Luis Ibez Hans J. Johnson, Matthew M. McCormick. *The ITK Software Guide.* 01 2017.

[4] David Marr and E. Hildreth. Theory of edge detection. *Proceedings of the Royal Society of London Series B*, 207:187–217, 1980.

[5] Stanley Osher and R P. Fedkiw. *The Level Set Methods and Dynamic Implicit Surfaces*, volume 57. 01 2004.

[6] Richard E. Woods Rafael C. Gonzalez. *Digital Image Processing 2/E.*

[7] J. Sethian. Fast marching methods. *SIAM Review 41*, pages 199–235, 1999.

[8] Abdel Aziz Taha and Allan Hanbury. Metrics for evaluating 3d medical image segmentation: analysis, selection, and tool. *BMC Medical Imaging*, 15(1):29, Aug 2015.