Level Set Image Segmentation Based On Edge Detection for 3D Medical Images

Lokeswara Rao Bonta[1] Ankit Anand,[1], A. Ranga Shaarad[1], and N Uday Kiran[1]

Sri Sathya Sai Institute of Higher Learning, Andhra Pradesh 515134, INDIA
lrbonta.sssihl@gmail.com,ankitanandjmp95@gmail.com rangashaarad@gmail.com
nudaykiran@sssihl.edu.in

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. In this work, we have constructed a level set based 3D image segmentation pipeline based on edge detection and evaluated it using existing metrics. We also incorporated this pipeline to the open-source software 3DSlicer for easy access.

Keywords: Threshold Level Set Segmentation, 3D Slicer, Canny Edge Detector

1 Introduction and Background

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. 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. 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.

Segmenting an image means dividing the image into its constituent parts [4]. 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.

Our contribution in this work is extending canny edge detector and threshold level set segmentation algorithms to 3D medical images. The contribution also includes the integration of the above tools in the open source software called 3D slicer[1].

The reminder of this paper is organized as below: Section 2 describes the methods used in this paper, which includes canny edge detector for 3D images and threshold level set segmentation for 3D medical images. Section 3 presents experimental evaluation, which contains the various evaluation metrics used, brief description of data, hardware and software framework, results and analysis. The paper end with concluding remarks and future work in section 4.

2 Methods Used

2.1 Canny Edge Detection(3D) on 3D Images

Method: We have extended CED method discussed in [3](which is for 2D images) to 3D images. The CED as mentioned in [3] involves modules for Gaussian blur, gradient, non-maximum suppression and double thresholding. We are using a variant of Gaussian filter to de-noise the image. We consider the neighbourhood as a cube for each pixel with side 3 voxels. We used the following matrix for blurring:

$$\begin{array}{|ccc|}\hline 0 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & 0 \\ \hline\end{array} \quad \begin{array}{|ccc|}\hline 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \\ \hline\end{array} \quad \begin{array}{|ccc|}\hline 0 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & 0 \\ \hline\end{array}$$

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:

$$\begin{array}{|ccc|}\hline 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \\ \hline\end{array} \quad \begin{array}{|ccc|}\hline 2 & 0 & -2 \\ 4 & 0 & -4 \\ 2 & 0 & -2 \\ \hline\end{array} \quad \begin{array}{|ccc|}\hline 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \\ \hline\end{array}$$

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 neighbourhood is replaced by the magnitude of the gradient. The angles that the gradient makes with the three coordinate axes are calculated 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.

2.2 Threshold Level Set Segmentation

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[5] is

$$\frac{\partial\psi}{\partial t} = -\alpha\mathbf{A}(\mathbf{x}).\nabla\psi - \beta P(x)|\nabla\psi| + \gamma\mathbf{Z}(\mathbf{x})\kappa|\nabla\psi| \tag{1}$$

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.

Finding the advection and propagation terms depend on the target 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.

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 Fast-Marching module is used to create initial level set from the seeds entered by the user [8]. 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 [6]. 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.

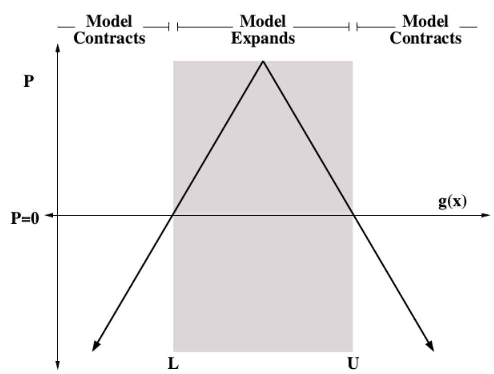


Fig. 1. Calculation of Propagation term [5]

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

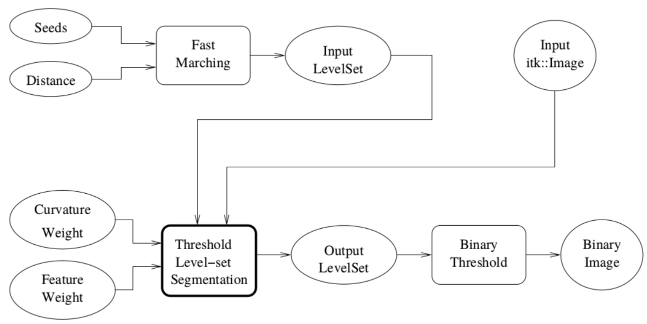


Fig. 2. ThresholdLevelSet Segmentation Pipeline[5]

3 Experiments, Results and Discussion

3.1 Evaluation Metrics

Literature gives a number of ways to evaluate segmentation. The paper [9] gives a list of metrics that can be used for evaluation of segmentation of medical images. Taha[9] discusses the use of True Positive Rate (TPR) also called sensitivity and True Negative Rate(TNR) also called specificity as given in the equations below.

$$Recall = Sensitivity = TPR = \frac{TP}{TP + FN} \tag{2}$$

$$Specificity = TNR = \frac{TN}{TN + FP} \tag{3}$$

A volume based metric discussed in [9] is volumetric similarity. Volumetric similarity is defined as

$$VolumetricSimilarity = 1 - \frac{|FN - FP|}{2TP + FP + FN} \tag{4}$$

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.

3.2 Dataset Used

For evaluation of segmentation pipeline, we used pre-operative TCGA GBM dataset from BraTS(Brain Tumour Segmentation) - 2017[2] grand challenge . 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.

3.3 Experimental Framework and Implementation Details

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 used the $ITK4.9$ for the header files required for the level set based implementations. We have also used $3DSlicer$ for visualization purposes.

We have implemented the CED for 3D images using the algorithm for 2D given in [3]. 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).

3.4 Results and Discussion

The sensitivity and specificity of tumour segmentation of 10 images picked from the dataset using threshold level set segmentation method is given in the figure 3. We can see from the plots that the sensitivity values are 0.75 on an average, which tells that segmentation quality is very good and specificity values is 0.95 on an average, which is expected because of class imbalance problem(the number of negative values are far greater the positive values which makes the true negative rate/specificity very high). The goodness of segmentation also can be observed from figure 5 which shows that the volumetric segmentation we have obtained is 90% on average. The results on a sample image are given in figure 3.4(a), 3.4(b). 3.4(c) shows error image(segmeted image - ground truth). We can clearly see that the segmented section with our method is very close to the ground truth. The images are rendered using 3D slicer.

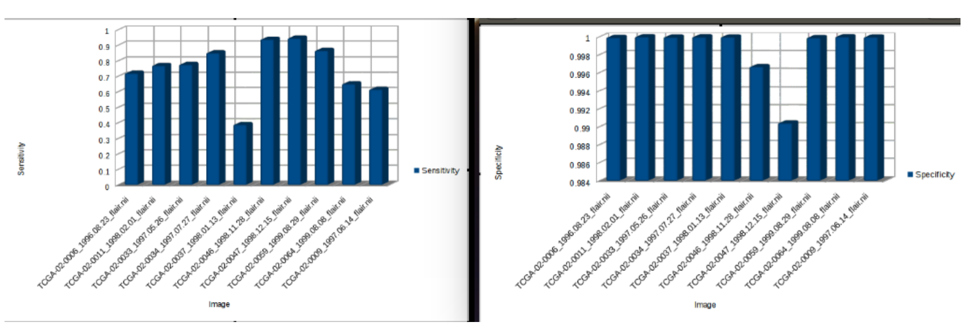
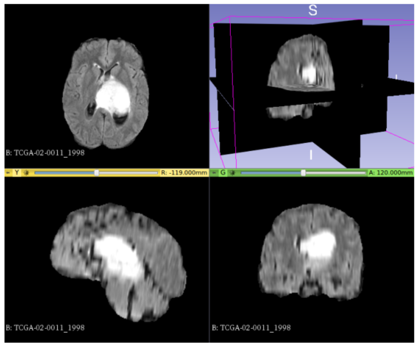
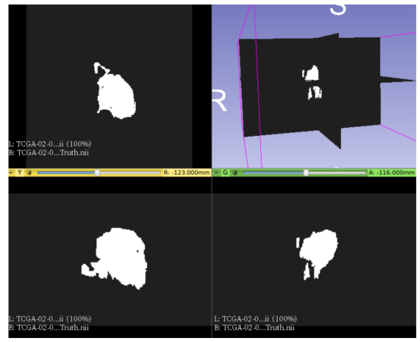


Fig. 3. Sensitivity and Specificity Plots for 10 Segmentations



(a)Input Image



(b) Level-Set based Segmentation

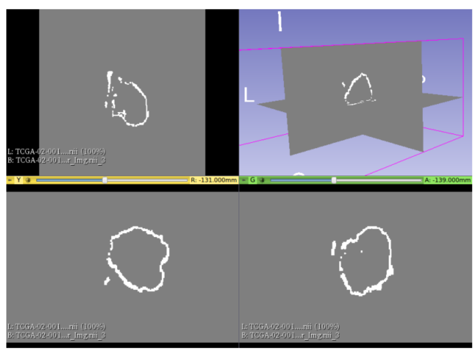


Fig. 4. (c)Error Image for Segmentation

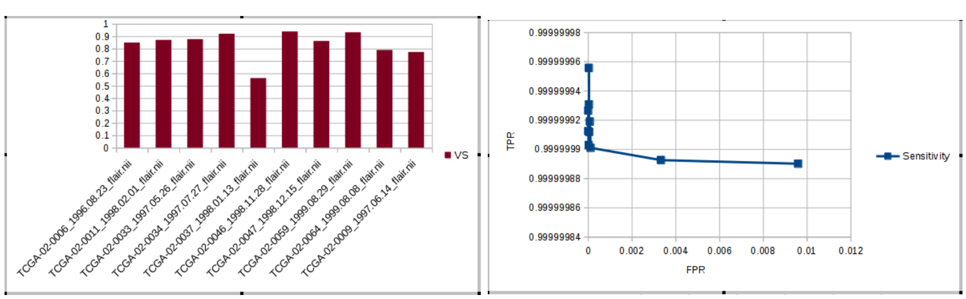


Fig. 5. Volumetric Similarities and ROC curve for 10 segmentation

4 Conclusions and Future Work

In this 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 the Cannys first derivative for finding edges as a intermediate step for level set based segmentation worked well and in-line with theory. We also evaluated our segmentation using some existing metrics and integrated our implementation to 3D Slicer for easy access. The implementation can be made more accurate by studying the effect of each of the terms in the level set equation considered. We can experiment with the other terms that can be included in the level set equation.The implementation can be parallelized to improve the speedup. Moreover, it is possible to parallelize as all the computations involved are serial in nature.

Acknowledgments

Our work is dedicated to Bhagawan Sri Sathya Sai Baba, Founder Chancellor of Sri Sathya Sai Institue of Higher Learning.

References

1. 3dslicer. https://www.slicer.org/.
2. S Bakas, H Akbari, A Sotiras, M Bilello, M Rozycki, J Kirby, J Freymann, K Farahani, and C Davatzikos. Segmentation labels and radiomic features for the preoperative scans of the tcga-lgg collection. *The Cancer Imaging Archive*, 2017.
3. J. Canny. A computational approach to edge detection. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 8(6), 1986.
4. Pierre Kornprobst Gilles Aubert. *Mathematical Problems in Image Processing.*
5. Luis Ibez Hans J. Johnson, Matthew M. McCormick. *The ITK Software Guide.* 01 2017.
6. David Marr and E. Hildreth. Theory of edge detection. *Proceedings of the Royal Society of London Series B*, 207:187–217, 1980.
7. Stanley Osher and R P. Fedkiw. *The Level Set Methods and Dynamic Implicit Surfaces*, volume 57. 01 2004.
8. J. Sethian. Fast marching methods. *SIAM Review 41*, pages 199–235, 1999.
9. 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.