# 3D Workflow for Segmentation and Interactive Visualization in Brain MR images using Multiphase Active Contours

Rengarajan Pelapur[1], V. B. Surya Prasath[2], Juan C. Moreno[3], Michael M. Heck[1]

***Abstract*— In this paper, we are proposing a 3D segmentation and interactive visualization workflow. The segmentation implementation uses a globally convex multiphase active contours without edges. This algorithm has been proven to be initialization independent due to their globally convex formulation and better than other approaches due to robustness to image variations and adaptive energy functionals. The workflow includes a flexible 3D visualization application that can handle very large volumes using multi-resolution hierarchical data formats following the segmentation. We also designed a custom fragment shader that is capable of meaningfully fusing the data from three different volumes: a segmented label volume, a mean value per voxel volume and a skull striped volume for effective visualization without modifying the segmented results. Giving researchers the access to a whole end to end pipeline, from 3D segmentation to custom real time interactive 3D visualization is, in our opinion, a powerful tool focused on an analyst/expert centric workflow.**

## I. Introduction

Segmentation as an ill-posed problem has existed through the history of computer vision and image processing [1], [2], [3], [4]. Automatic/semi-automated analysis has always been of interest to researchers as a means of speeding up the accuracy and analyzing an ever growing volume of data. Breakthroughs in imaging methodologies, such as using safer dyes, increasing resolutions of scanners and inexpensive storage has resulted in a plethora of data that requires analysis and a way to efficiently visualize the data [5]. 3D segmentation and visualization of medium to large volumes included painstaking processes of marking regions of interest for every slice and interpolating the labels through the entire volume [6]. In addition to this, there are no known universal fully automatic segmentation algorithms that are capable of handling tumors or lesions of different shapes and types at the level of accuracy that is required for a successful treatment plan. While deep convolutional neural networks have shown promise in large scale accurate analysis, there is still quite a lot of interdisciplinary work that is required in order for these networks to mature and be approved for use in diagnostics. Throughout this work, we assume that the diagnosticians will still follow the same workflow of having to look through a vast amount of data and tools that help in this regard will be of great interest.

One of the landmark approachs in automatic segmentation is that of Chan and Vese [7] who extended the seminal work of Mumford and Shah [8] based on variational energy minimization formulation. Vese and Chan [9] later extended their active contour without edges model to perform multiphase segmentation. This multiphase approach provides a numerically stable model that is independent of initialization and does not get stuck on local minima, making it an easy to use and powerful algorithm. While applications such as 3D Slicer [10] have provided interactive tools along with a set of basic algorithms to help with 3D segmentation in MR and CT images, we believe that there are specific problems that require advanced algorithms, especially with large data and the need to interactively visualize and customize the visualization. While computer vision and image processing techniques have taken a backseat due to the immense popularity of convolutional neural networks, the need to train extensive networks, while superior in accuracy under certain circumstances, is not practical in an interactive workflow. GeoS [11] was presented as a geodesic algorithm that would help with interactive 3D segmentation in medical imagery. Our work distinguishes itself in three key ways : (1) A globally convex multiphase algorithm that is far superior to edge based techniques, (2) Interactive volume visualization tools using a powerful framework, (3) Customization friendly approach that would help researchers generate high quality visual outputs for further determining diagnostic relevance. We used the MR simulated images from BrainWeb [12] and generated results by processing each individual slice and visualizing the end result as a full 3D volume.

We organized the rest of the paper as follows. Section II briefly introduces the active contour segmentation model and Section III details how the segmentation module and the visualization techniques are integrated for fast interactive workflow. Finally, Section IV concludes the paper.

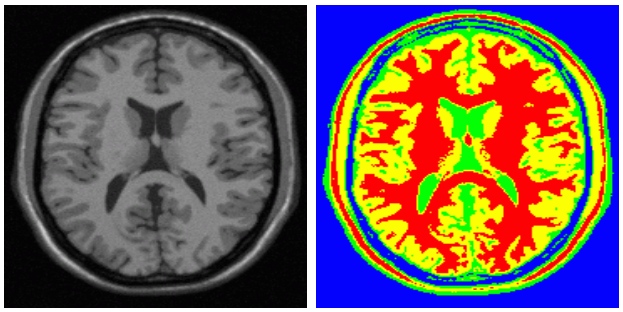
## II. Globally convex multiphase active contours for brain MRI segmentation

We recall the multiphase formulation of Vese and Chan [9] and restrict ourselves to the piecewise constant four phase case here, the general case will be reported elsewhere. Let $\phi_1, \phi_2 : \Omega \subset \mathbb{R}^2 \to \mathbb{R}$ be the two level sets. $H_1 = H(\phi_1)$, $H_2 = H(\phi_2)$ and $\tilde{H}_1 = 1 - H(\phi_1)$, $\tilde{H}_2 = 1 - H(\phi_2)$ where $H$ is the Heaviside function. Our goal is to solve the

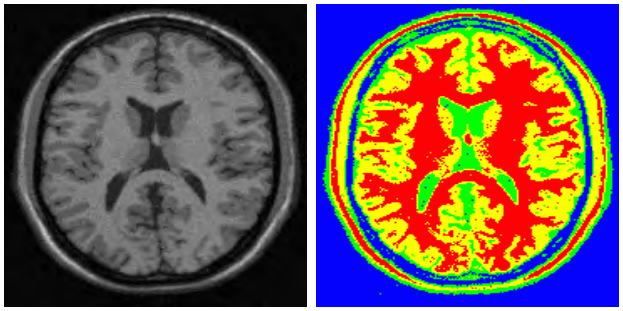
[1]R. Pelapur and M. Heck are with the Materials and Structural Analysis Division, ThermoFisher Scientific, Houston, TX 77084 USA. {Rengarajan.Pelapur, Mike.Heck}@thermofisher.com

[2]V. B. S. Prasath was with the Computational Imaging and Visualization Analysis Lab, Department of Computer Science, University of Missouri-Columbia, Columbia MO 65211 USA. He is currently with the Division of Biomedical Informatics, Cincinnati Children's Hospital Medical Center (CCHMC), Cincinnati, OH 45229 USA. Also affiliated with the Department of Biomedical Informatics, College of Medicine, University of Cincinnati, OH USA. prasaths@missouri.edu

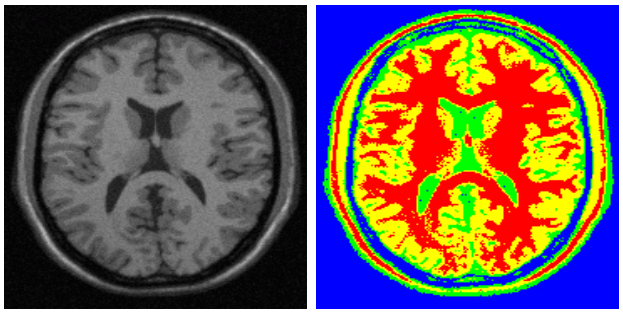
[3]J. C. Moreno is with the Department of Computer Science, University of Beira Interior, 6201-001 Covilhã, Portugal. jcmb@ubi.pt



(a) $n = 3$, $RF = 20$



(b) $n = 5$, $RF = 20$



(c) $n = 5$, $RF = 40$

Fig. 1. Examples of robust four phase active contour segmentation with energy minimization (3) of transaxial slices from the BrainWeb database. We show the input (left) and the segmentation for different noise ($n$) and non-uniformity ($RF$) levels. (a) $n = 3$, $RF = 20$, (b) $n = 5$, $RF = 20$, (c) $n = 5$, $RF = 20$.

following variational energy minimization,

$$\min_{(\mathbf{\Phi},\mathbf{c})} \left\{ F_1(\mathbf{\Phi},\mathbf{c}) = \sum_{k=1}^{2} \int_\Omega \delta(\phi_1)|\nabla\phi_k|\,dx + \sum_{i=0}^{3} \int_\Omega \lambda_{\mathbf{b}(i)}(I - c_{\mathbf{b}(i)})^2 \prod_{k=1}^{2} \omega_{b_k(i)}(H(\phi_k))\,dx, \right\} \quad (1)$$

where $\mathbf{b}(i) = (b_1, b_2)$ are a binary representation of $i \in \{0, 1, 2, 3\}$, $\omega_0(z) = z$ and $\omega_1(z) = 1 - z$. The parameters $\lambda_{\mathbf{b}(i)}$ are weighting fitting terms. The variational problem (1) is solved using a two-step algorithm where at the first step the constant mean values $\mathbf{c} = (c_{11}, c_{10}, c_{01}, c_{00})$ are computed by

$$c_{\mathbf{b}(i)} = \frac{\int_\Omega I \prod_{k=1}^{2} \omega_{b_k(i)}(H(\phi_k))\,dx}{\int_\Omega \prod_{k=1}^{2} \omega_{b_k(i)}(H(\phi_k))\,dx}$$

and at the second step we update the function $\mathbf{\Phi} = (\phi_1, \phi_2)$.

In Vese and Chan [9], the corresponding gradient descent equations are used to implement the active contours in a level set based approach. In the numerical implementation of the above PDEs a non-compactly supported, smooth approximation of the Heaviside function $H_\epsilon(x)$, such that $H_\epsilon(\mathbf{x}) \to H(x)$ as $\epsilon \to 0$ is utilized. Since the above minimization (1) is non-convex the time discretized gradient descent PDEs requires large iterations to convergence. Moreover, the final segmentation result may not correspond to the global minimum of the energy function. Following, Chan et al [13], we can derive a relaxed energy minimization formulation by dropping the Dirac delta ($\delta(\phi)$ in (1)) to obtain,

$$\min_{(\mathbf{u},\mathbf{c})\in\{0,1\}^2\times\mathbb{R}^4} \left\{ \mathcal{F}_2(\mathbf{u},\mathbf{c}) = \sum_{k=1}^{2} \int_\Omega |\nabla u_k|\,dx + \sum_{i=0}^{3} \int_\Omega \lambda_{\mathbf{b}(i)}(I - c_{\mathbf{b}(i)})^2 \prod_{k=1}^{2} \omega_{b_k(i)}(u_k)\,dx \right\}. \quad (2)$$

As proved in [14], the above energy minimization problem can be relaxed by allowing the variable $\mathbf{u}$ to take values in $[0, 1]^2$ and then solve the minimization problem,

$$\min_{(\mathbf{u},\mathbf{c})\in[0,1]^2\times\mathbb{R}^4} \mathcal{F}_2(\mathbf{u},\mathbf{c}), \quad (3)$$

where for fixed $\mathbf{c} \in \mathbb{R}^4$ there exist a function of bounded variation $\mathbf{u} \in [0, 1]^2$ ($\mathbf{u} \in BV_{[0,1]^2}(\Omega)$) solving (2). This formulation allows for a robust automatic segmentation that can be applied to brain MR imagery both with inhomogeneities as well as increasing noise levels. Figure 1 shows example segmentations obtained with the robust energy minimization model on normal brain MRIs from the BrainWeb database at different inhomogeneity and noise levels.

## III. High Performance 3D Segmentation and Visualization

We have implemented the entire workflow, comprised of the multiphase segmentation algorithm for solving the energy minimization (3) and visualization, using a modern version of the Open Inventor [15], [16] toolkit. Open Inventor, at its backend, makes OpenGL calls to perform the rendering while exposing a higher level, object-oriented, application programming interface to the programmer. The arithmetic computation uses the toolkit's concept of *engines*. Open Inventor engines traditionally were used to manipulate the 3D data or geometry elements by *linking* to transformation nodes such as to translate or rotate these elements [17]. A modern evolution of these engines is more generalized to

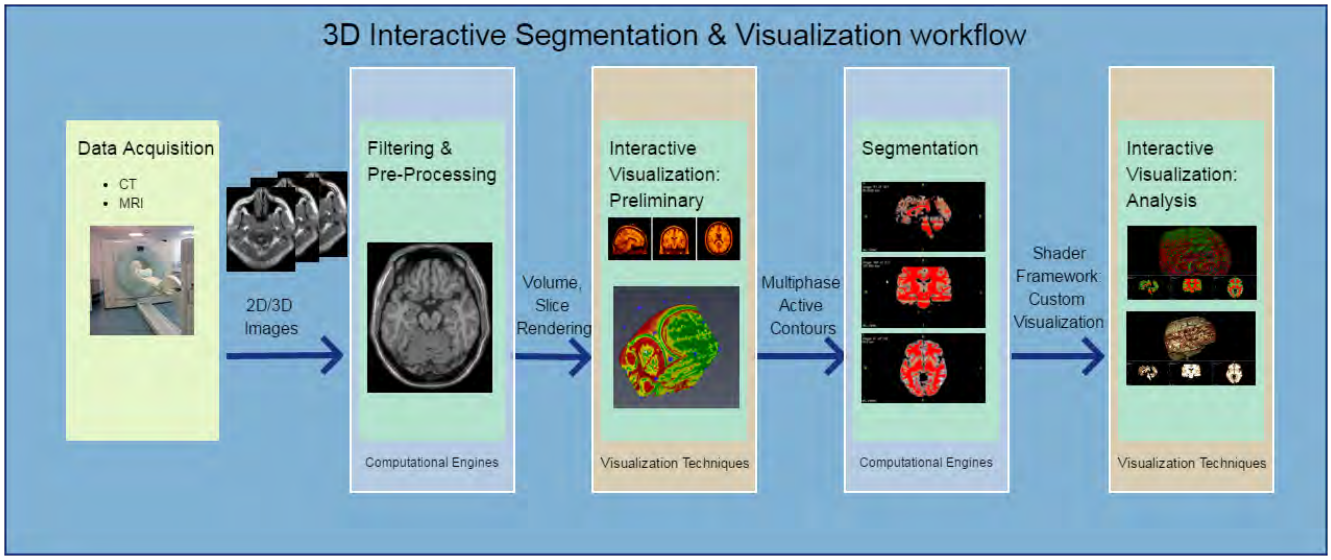
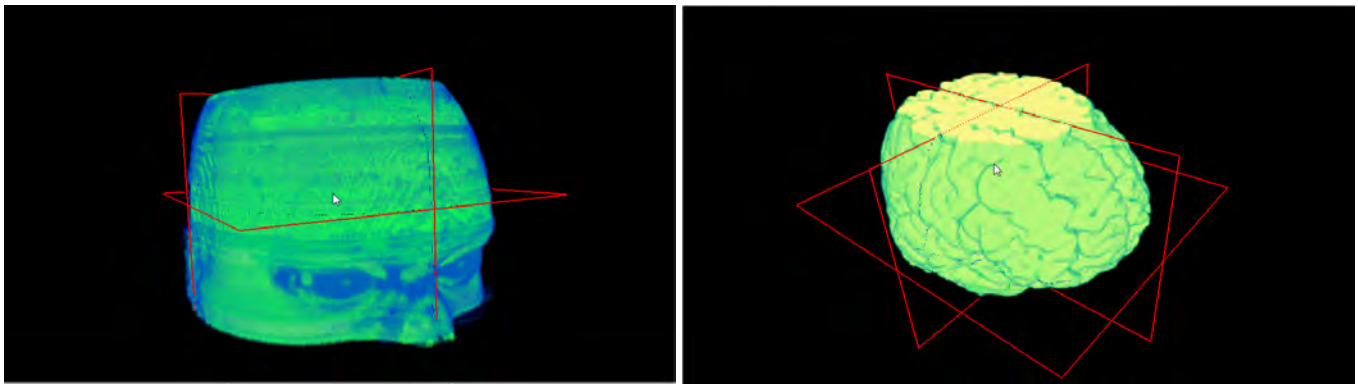



Fig. 2. End to end 3D segmentation and interactive visualization workflow showing the flow of data from acquisition to final visualization analysis. Steps include filtering, pre-processing, several image processing engine based computations, and a custom shader framework to combine resulting volumes.



(a) Volume rendering with all all pixels per slice visualized

(b) Volume rendering after region growing removing the pixels outside of the brain region

Fig. 3. Region growing is used to remove the un-needed pixels for every slice so that the user can visualize just the brain regions in 3D

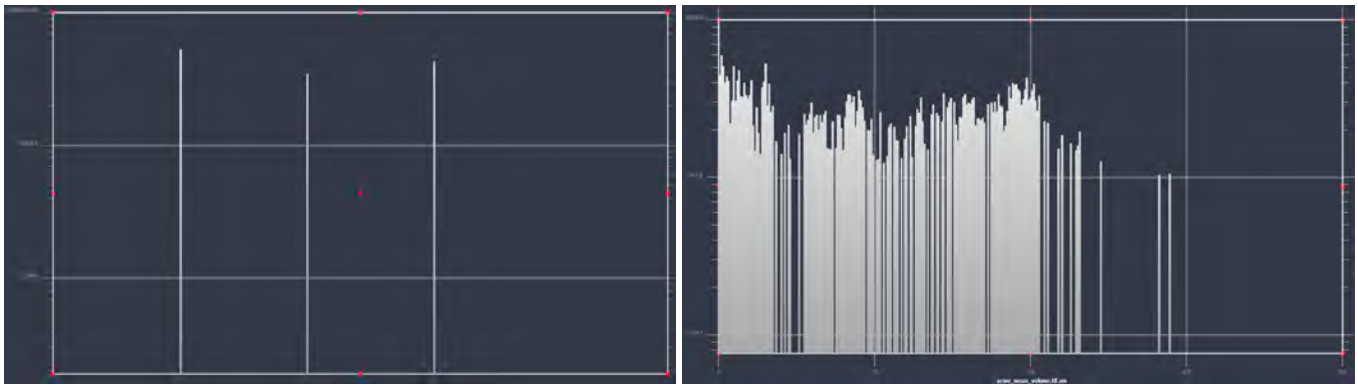
include computation and those specific to image processing are now part of an extension to the toolkit [18]. The object oriented approach to 3D rendering trickles down to arranging basic elements such as geometry (e.g. cones, spheres, cubes), volume data (e.g. large data tiles out of core or arrays), lights, cameras, rendering techniques (e.g. volume rendering, slices) and more as part of a graph structure. These elements are called *nodes* and for more information see [15]. Figure 2 shows the complete workflow from reading in a stack of images or 3D volume, pre-processing the result and a few visualization steps with custom shaders or chosen colormaps applied consistently across the 2D slice visualization as well as in 3D. It is important to note that the same data *nodes* are shared across different rendering methods because of the *scenegraph* architecture of Open Inventor and thus there is no replication of data in memory.

### A. *Multiphase Active Contours for Segmentation*

In the examples shown, our workflow starts with reading in the MR data as a stack of images. The multiphase active contour algorithm (3) is then run on each individual slice. The result is one 2D map per slice where every pixel corresponds to one of the 4 mean values determined by the segmentation which we will refer to as the *mean* volume when all the slices are put together in 3D. Similarly, a *label* volume is when each pixel in the result map is assigned an integer value. The implementation solves the equations mentioned in Section II. Updating the mean values is not performed in every iteration to save computation, rather, is controlled by a parameter. Meaningful defaults are recommended and can be tweaked if necessary by the user.

### B. *Region Growing for Skull Stripping*

Once we have the segmented volume, we would also need to remove the unwanted outside regions to look at the brain.



(a) Histogram of labeled volume. X-Axis scales from 0-300 showing discreet values

(b) Histogram of mean segmented volume

Fig. 4. The mean value to label association is done by looking at the same voxel in both the volumes. While the mean values per label has a variance, the statistics after association can be helpful to visualize.

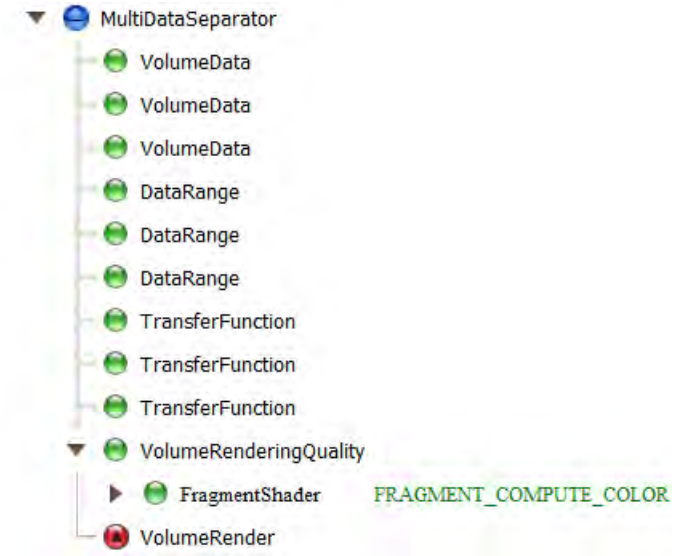



Fig. 5. Open Inventor scenegraph designed specifically with a shading script ('FragmentShader node') that combines the volume in a meaningful way. The voxels outside the binary skull stripped volume are made fully transparent. Note that there are 3 volume data nodes : 1) Mean valued 2) Labeled and 3) Binary skull stripped volume.

The procedure followed for this is where the region growing algorithm is applied through every slice for the labels that are not close to the image dimensions. The result for this is predictable due to the input being the segmented data where every region has a piece-wise constant value. Assembling the slices together we can then eliminate the unneeded parts and the resulting output looks as in Figure 3.

### C. Visualization and Shader Framework

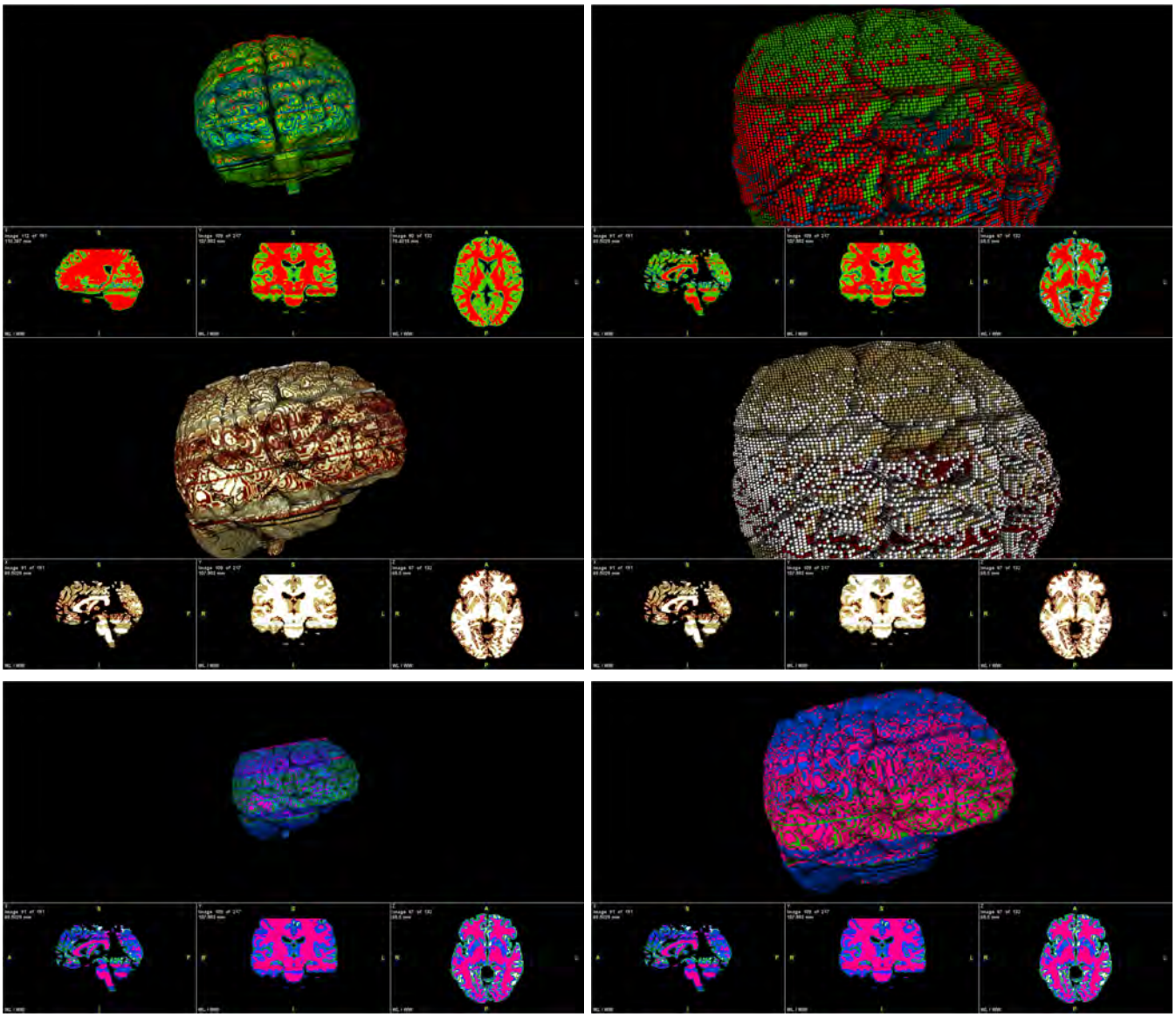
Following the processes of reading in the imagery, pre-processing and segmentation we would have two volumes: 1) Mean value per pixel in every slice through the volume and 2) Integer label per pixel in every slice through the volume. Since the multiphase active contour implementation works on a 2D slice, in order to interpolate and determine a 3D structure we assign an integer label in sorted order of mean values per slice. The histograms for both the volumes are shown in Figure 4. Instead of ad-hoc thresholding the volumes for ease of visualization, we are using a shader framework to combine the three volumes : the two mentioned above and the region growing binary volume after removing the extra regions around the brain region. The scenegraph describing the multiple volumes is as shown in Figure 5. The shader code is shown in Code 1.

Code 1. GLSL (GLslang) or OpenGL Shading Language script to blend the 3 volumes

```
//use Open Inventor useful headers to get
    data value
//and colors assigned to the data using the
    transfer functions
//!oiv_include <VolumeViz/vvizGetData_frag.h>
//!oiv_include
    <VolumeViz/vvizTransferFunction_frag.h>

uniform VVizDataSetId Volume1;
uniform VVizDataSetId Volume2;
uniform VVizDataSetId Volume3;

//blending function
vec4 blend(in vec3 texCoord)
{ //get voxel values
 VVIZ_DATATYPE data1 = VVizGetData(Volume1,
     texCoord);
 VVIZ_DATATYPE data2 = VVizGetData(Volume2,
     texCoord);
 VVIZ_DATATYPE data3 = VVizGetData(Volume3,
     texCoord);
 //get assigned color values for each volume
 vec4 color1 = VVizTransferFunction(data1,
     0);
 vec4 color2 = VVizTransferFunction(data2,
     1);
 vec4 color3 = VVizTransferFunction(data3,
     2);
 //return custom color
 //data3 is the voxel value from the skull
     stripped binary volume
 vec4 result_color;
```



(a) Volume Rendering (b) Voxelized Rendering

Fig. 6. Volume rendering and 2D slice rendering at each axis : Saggital, Coronal and Axial using (top row) the *'Physics'*, (middle row) the *'Glow'* and (bottom row) the *'Standard'* colormaps in (a), *Voxelized* rendering in (b). Note that the mouse click and drag in the 2D slice panels results in scrolling through the slices and in the 3D panel results in full manipulation of 3D data including rotation and translation of the camera.

```
 if (data3 > 0.0f){
  result_color.r = color2.r * data1;
  result_color.g = color2.g * data1;
  result_color.b = color2.b * data1;
  result_color.a = color1.a;
 }
 else{
  //result_color = Color123;
  result_color.r = 0.0f;
  result_color.g = 0.0f;
  result_color.b = 0.0f;
  result_color.a = 0.0f;
 }
 return result_color;
}
// Implement VVizComputeFragmentColor for
    slice nodes
vec4 VVizComputeFragmentColor(VVIZ_DATATYPE
    vox, vec3 texCoord)
{
 return blend(texCoord);
}
// Implement VVizComputeFragmentColor for
    SoVolumeRender
vec4 VVizComputeFragmentColor(VVizDataSetId
    data, vec3 rayDir, inout VVizVoxelInfo
    voxelInfoFront, in VVizVoxelInfo
    voxelInfoBack, int maskId)
{
 return blend(voxelInfoFront.texCoord);
}
```

### D. Dataset and Results

We used synthetically generated MR images from the BrainWeb project [12]. Images showing multiple sclerosis (MS) lesions and a normal brain were used to segment and visualize. For a benchmark of the multiphase active contour model used in this work, we refer the readers to detailed analysis against ground-truth with standard metrics [14]. Figure 6 shows the final visualizations with different colormaps that can be used for analysis, for example, the *'Physics'*, *'Glow'* and *'Standard'*. We determine the voxel value to be displayed in realtime and map it to a colormap which is a user defined preference. This enables the user to visualize the fidelity of the obtained segmentations and further analyze the volumes in full 3D. The corresponding voxelized renderings are also shown. These can be useful for the user to perform voxel-exact selection on the volume for analysis and interpretation.

## IV. Conclusion

Brain MR image analysis and visualization is an important component many neurological studies. In this work, we presented our 3D segmentation and interactive visualization with the state of the art segmentation module based on active contours and real-time shader based rendering. Our proposed workflow is flexible such that various pre-filtering and image processing algorithms can be utilized as plug-n-play modules along with visualizations that are extensible by fusing different 3D volumes. We provide a proof of concept experiment with BrainWeb MR images that shows the workings of the proposed workflow for fast segmentation and real-time interactive visualization on 3D volumes. We believe this combination of advanced computational image processing techniques along with volume rendering offers a promising tool in biomedical image informatics.

## References


[1] O. Wirjadi, "Survey of 3d image segmentation methods," Fraunhofer (ITWM), Tech. Rep. 123, 2007. [Online]. Available: http://nbn-resolving.de/urn/resolver.pl?urn:nbn:de:hbz:386-kluedo-15457

[2] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern analysis and Machine Intelligence (PAMI)*, pp. 888–905, 2000.

[3] P. Felzenszwalb and D. Huttenlocher, "Efficient graph-based image segmentation," *International Journal of Computer Vision (IJCV)*, pp. 167–181, 2004.

[4] A. Schenk, G. Prause, and H. Peitgen, "Efficient semiautomatic segmentation of 3d objects in medical images," in *MICCAI*. Springer, 2000, pp. 186–195.

[5] D. Gering, A. Nabavi, R. Kikinis, E. Grimson, N. Hata, P. Everett, F. Jolesz, and W. Wells, "An integrated visualization system for surgical planning and guidance using image fusion and interventional imaging," in *MICCAI*. Springer, 1999, pp. 809–819.

[6] J. Sijbers, P. Scheunders, M. Verhoye, A. V. der Linden, D. V. Dyck, and E. Raman, "Watershed-based segmentation of 3d mr data for volume quantization," *Magnetic Resonance Imaging*, pp. 679–688, 1997.

[7] T. F. Chan and L. A. Vese, "Active contours without edges," *IEEE Transactions on Image Processing*, vol. 10, no. 2, pp. 266–277, 2001.

[8] D. Mumford and J. Shah, "Optimal approximations by piecewise smooth functions and associated variational problems," *Communications in Pure and Applied Mathematics*, vol. 42, no. 5, pp. 577–685, 1989.

[9] L. Vese and T. F. Chan, "A multiphase level set framework for image segmentation using the Mumford and Shah model," *International Journal of Computer Vision*, pp. 271–293, 2002.

[10] A. Fedorov, R. Beichel, J. Kalpathy-Cramer, J. Finet, J. Fillion-Robin, S. Pujol, C. Bauer, D. Jennings, F. Fennessy, M. Sonka, *et al.*, "3d slicer as an image computing platform for the quantitative imaging network," *Magnetic resonance imaging*, pp. 1323–1341, 2012.

[11] A. Criminisi, T. Sharp, and A. Blake, "Geos: Geodesic image segmentation," *Computer Vision–ECCV 2008*, pp. 99–112, 2008.

[12] C. Cocosco, V. Kollokian, R. Kwan, B. Pike, and A. Evans, "Brainweb: Online interface to a 3d mri simulated brain database," in *NeuroImage*, 1997.

[13] T. F. Chan, S. Esedoglu, and M. Nikolova, "Algorithms for finding global minimizers of image segmentation and denoising models," *SIAM Journal on Applied Mathematics*, vol. 66, no. 5, pp. 1632–1648, 2006.

[14] J. C. Moreno, V. B. S. Prasath, H. Proenca, and K. Palaniappan, "Brain MRI segmentation with fast and globally convex multiphase active contours," *Computer Vision Image Understanding*, vol. 125, pp. 237–250, 2014.

[15] J. Wernecke *et al.*, *The Inventor mentor: Programming Object-oriented 3D Graphics with Open Inventor*. Addison-Wesley Reading, MA, 1994.

[16] J. Wernecke, *The Inventor Toolmaker: Extending Open Inventor*. Addison-Wesley Longman Publishing Co., Inc., 1994.

[17] "Open inventor : Concept of engines," http://developer90.openinventor.com/content/141-introduction-engines.

[18] "Open inventor : Imageviz," http://developer97.openinventor.com/APIS/RefManCpp/group___image_viz.html.