

Generation of Connectivity-Preserving Surface Models of Multiple Sclerosis Lesions

Oscar Meruvia-Pastor^a, Mei Xiao^{a,1}, Jung Soh^a, Christoph W. Sensen^a

^a*SunCenter of Excellence for Visualgenomics
Department of Biochemistry and Molecular Biology
Faculty of Medicine, University of Calgary
3330 Hospital Drive NW
Calgary, Alberta, T2N 4N1, Canada*

Abstract. Progression of multiple sclerosis (MS) results in brain lesions caused by white matter inflammation. MS lesions have various shapes, sizes and locations, affecting cognitive abilities of patients to different extents. To facilitate the visualization of the brain lesion distribution, we have developed a software tool to build 3D surface models of MS lesions. This tool allows users to create 3D models of lesions quickly and to visualize the lesions and brain tissues using various visual attributes and configurations. The software package is based on breadth-first search based 3D connected component analysis and a 3D flood-fill based region growing algorithm to generate 3D models from binary or non-binary segmented medical image stacks.

Keywords. Multiple Sclerosis, Lesions, Connected Components, Segmentation

1. Introduction

Medical image slices are frequently reconstructed into a 3D surface model to facilitate realistic visualization. Such reconstruction usually results in a single 3D model built from all regions of interest in a stack. When the target object is an anatomically well-defined structure, the reconstructed 3D model can be explored as is. However, when the reconstructed model represents a randomly distributed collection of objects with similar characteristics, such as brain lesions formed by the progression of multiple sclerosis (MS) ([1, 2]), it is useful to select certain clinically meaningful portions of the model to investigate in more detail.

Topological information contained in 3D image stacks is very useful. For example, instead of visualizing all the lesions together, users might want to distinguish each individually connected lesion from the whole model. It will be useful for the users to visualize the selected brain lesions and other brain tissues such as the cortex and subcortex with different visual attributes such as color and transparency. Connectivity can play a key role in these cases and we need well-connected models in order to

¹ Corresponding Author: mxiao@ucalgary.ca

separate them easily and quickly. Therefore, a well formatted surface mesh structure is needed.

The goal of our tool is to quickly and accurately create 3D models from a segmented medical image stack, such that connected 3D regions are easily separable from the whole model. In this case, the users can easily select the lesions and related brain tissues and view them in any visual configurations they want. There are two main algorithms that can be used to create 3D models of MS lesions from MRI stacks: the marching cubes algorithm and its various extensions ([3-9]), and region-growing based surface creation algorithms ([10-13]). Most marching cubes algorithms work on binary-labeled medical image stacks. However, to study the dynamics of MS lesion changes, it is important to use non-binary segmented image stacks that can store both lesion and brain tissue information. Region growing algorithms can work on non-binary segmented stacks.

We have developed a method for quickly generating surface models from a medical image stack, which can preserve connectivity of voxels such that connected components of a model can be easily selected and visualized. Our method uses breadth-first search based 3D connected component analysis and 3D flood-fill based region growing.

2. Materials and Methods

MS patients were recruited for MRI scanning in Halifax, Nova Scotia, Canada. Each patient has been scanned six times at an interval of once a month. T1, T2, and T2 FLAIR MRI images were obtained for use in our model creation tool. A sequence of image processing steps was performed on the head MRI data generated from each scan session, to convert them to input image data for our software.

First, the brain area was separated from each scan by using the FMRIB (Functional MRI of the Brain) Software Library (FSL, <http://www.fmrib.ox.ac.uk/fsl>). This step was performed on the T1 image set. Second, after brain extraction from the images, for each patient, all the scans were registered together. This step was required in order to make meaningful comparisons when visualizing changes in MS lesions of a patient from one scan session to the next. Third, we performed segmentation of the MRI scans to retrieve these types of tissues: cerebrospinal fluid (CSF), cortex, and sub-cortex. Finally, MS lesions were manually segmented from the FLAIR images based on white matter hyperintense areas. The T2 images were consulted frequently, since lesions that were not as clearly bright on FLAIR could be quite apparent on T2. T2 images were also used for confirming lesions identified on FLAIR.

The pipeline of the model creation program consists of two major processing steps (see Figure 1):

- A stack of 2D images that contains the MS lesions goes through 3D connected component analysis. The breadth-first search based 3D connected component analysis will separate the MS lesions into several connected components and a seed for each component is saved into a text file.
- The stack of 2D images in the previous step and the text file that contains the seed information are used in the flood-fill based region growing algorithm. A 3D model that contains all the MS lesions is created and saved into an OBJ file.

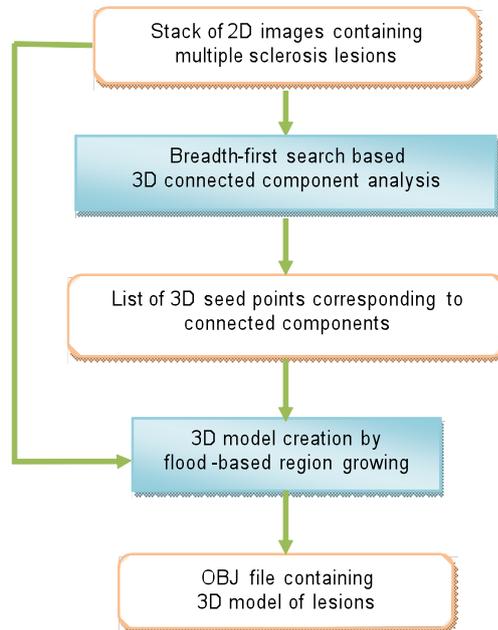


Figure 1. Procedure for building the lesion models.

The flood-fill based region growing algorithm starts by finding the boundary of the region beginning from the seed. Voxels of intensity within a certain range from the seed are searched for until the boundaries are met. For each boundary voxel with a coordinate (x, y, z) , eight surrounding points $(x \pm 0.5, y \pm 0.5, z \pm 0.5)$ are used to create faces. Those points are used as the vertices for the mesh. Each surrounding point is uniquely indexed according to the voxel coordinate (x, y, z) . In order to avoid using duplicated vertices, the index and positions of those surrounding points are saved in a hash table. For each boundary voxel, two triangles are created by traversing the surrounding points clockwise in order to guarantee a closed and consistent mesh. The final mesh vertices are saved in an indexed triangle array. At the end we have a mesh writer to save the contents of the indexed triangle array into an OBJ file, which can be displayed by most 3D mesh visualization and processing tools. The resulting surface might have obvious staircase effects, where a smoothing filter can be applied to each voxel to lessen the effects. The number of smoothing steps can be specified by users.

3. Results

We have created a Java3D™-based 3D medical image processing and visualization software package for facilitating brain lesion studies such as those required in MS. Although there are many automatic image segmentation programs developed for handling medical images, MS lesions usually have fuzzy boundaries (see Figure 2), and hence they still need to be manually segmented by neuroscience experts. The segmented images usually contain information on multiple brain tissue types. Each tissue type may include pixels in certain intensity value ranges, resulting in non-binary image stacks. Our program can be used to detect the 3D connected components in those

non-binary segmented images. Once the users set up the intensity range for the tissue type they are interested in detecting, a breadth-first search algorithm is applied to the image stack, recursively searching the 26 surrounding pixels in an image stack. After the search, the total number of connected components found and the starting pixel of each component will be saved in a text file. If the users are interested, it is also possible to record the volume of each component and even the positions of pixels comprising each component.

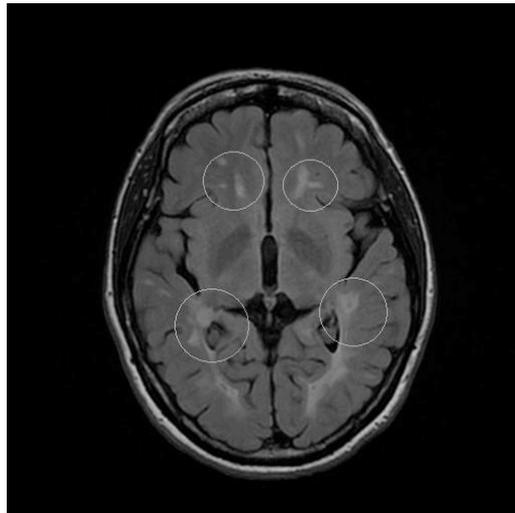


Figure 2. A slice from the MRI scan of an MS patient. Highlighted by the circles are inflammation areas.

The locations and volumes of MS lesions are of great significance for clinical studies to understand the damage to the brain tissues and cognitive abilities of the MS patients. By flood-fill based region growing from a pixel in an image stack, the boundaries of a separated MS lesion can be detected and triangular faces can be used to approximate the shape of that specific lesion. The mesh models that are created (see Figure 3) can be loaded into a 3D visualization program. By displaying and manipulating the visual attributes of 3D models for different types of brain tissues and lesions, MS lesion development patterns can be investigated in the context of surrounding anatomical structures.

Polygonal mesh models also have an advantage for lesion pattern studies. Unlike volume rendered models, the inner structures of polygonal models allow us to start from one vertex and traverse all the connected vertices to extract the 3D connected component and visually distinguish it from the rest of the model, in real time. For example, in our software package, we have a simple interface that allows the user to double click on a polygonal model, such that the single lesion that is connected to the clicked point can be extracted from the whole lesion model and displayed in another visualization window (see Figure 4). The connectivity calculation required for this component separation feature is performed by using the *vtkPolydataConnectivityFilter* class from VTK ([14, 15]).

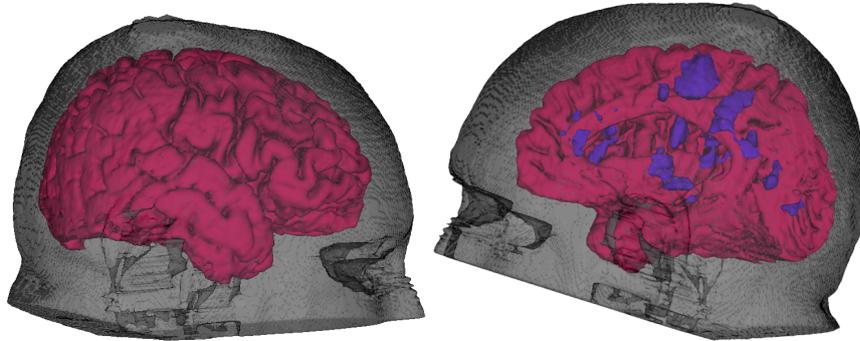


Figure 3. Rendering of a cortex model generated by our surface model building method (left) and rendering of cortex and MS lesion models generated by our surface model building method (right).

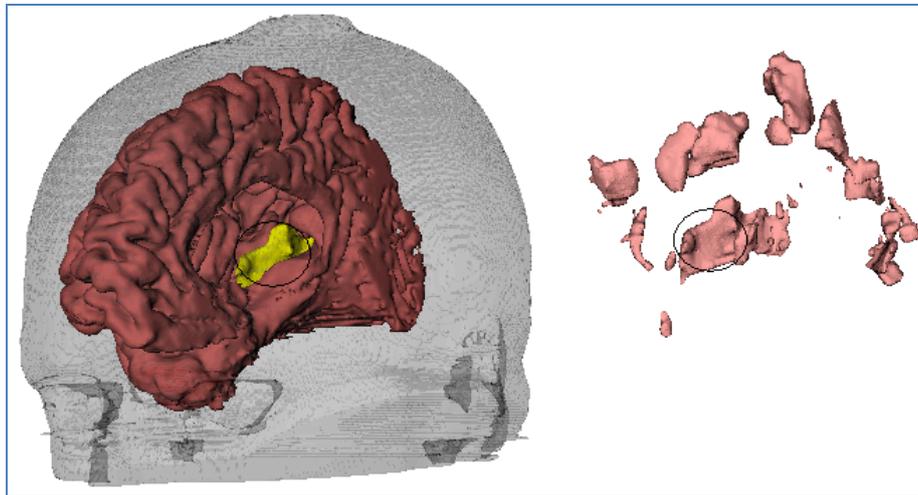


Figure 4. A connected lesion can be selected and highlighted by double clicking on it.

4. Discussion

3D models reconstructed from medical image stacks such as MRI and CT have been widely used in diagnosis and medical research. The marching cubes algorithm can be used to create high-resolution 3D mesh models. However, the marching cubes algorithm creates at least one triangle per voxel while it passes through the surfaces, which creates a huge amount of triangles ([13]). Depending on the implementation of the marching cubes algorithm, the mesh might not be consistent with the actual layout of the voxels. For example, Figure 5 shows two screen shots after applying *vtkPolydataConnectivityFilter* to two different mesh models. They both were generated from the same image stack (one MRI stack of MS lesion images). The left model was rendered using our approach. The right model was rendered using the *vtkMarchingContourFilter*. We can see that on the left image, once we double click on a lesion, the whole connected lesion is selected and its visual attributes can be changed as one single model. However, on the right image, once we double click on a lesion,

only part of the lesion is selected as shown by the changed color, and the rest of the same lesion has not been selected as shown by the same color as the rest of the whole model. If the connectivity of mesh models cannot be fully detected, the developmental patterns of the MS lesions can not be accurately expressed.

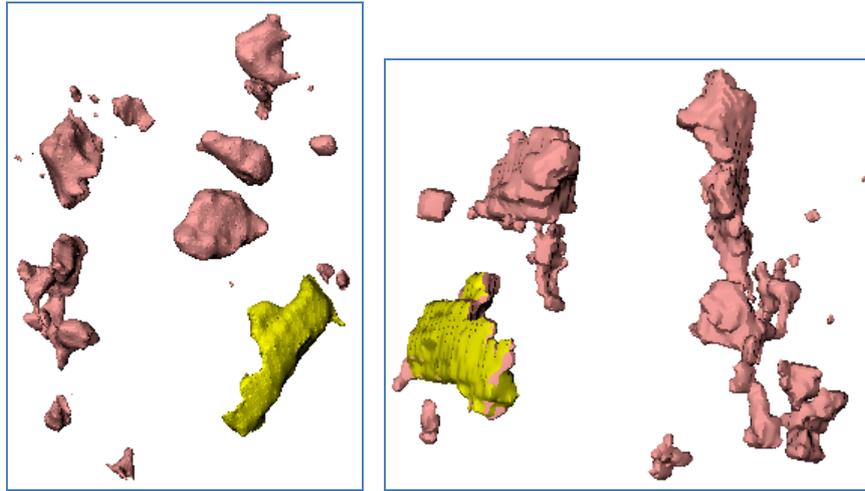


Figure 5. Comparison of connectivity in surface models. The model on the left is created using our surface model building software. The selection can clearly separate a connected component from the whole model. On the right, the model is created using the VTK marching cubes algorithm. The selection retrieves only part of a connected component from the whole model.

Marching cubes algorithms are usually applied to a whole image stack and every voxel will be checked. Therefore, for MRI and CT image stacks, it usually takes a long time to create a high-resolution mesh. Some versions of the marching cube applications also try to find the connected components, but processing a high-resolution mesh to find the connected components is very time-consuming and needs a large amount of computing resources. We tested a marching cube based program Afront ([16]) on the same MS lesion image stack. It took longer time and did not create a similar mesh as shown in Figure 5. Lewiner et al (2003) ([5]) developed a software package to create topologically correct manifold mesh. We tested the C++ implementation of the program. For any implicit functions applied to a grid, the resulting surface is consistent and the quality of the mesh seems to be very good. However, they did not provide any mesh creation method from medical image stacks. Therefore, the rendering time for our MS lesion image stacks in their program is not known.

Our software provides a valuable tool to study disease development by creating 3D models of the observed patterns in medical images. By building consistent mesh models quickly and efficiently directly from a medical image stack, researchers can create a large number of models that represent different individuals and different developmental stages of a disease. The ability to retrieve connected components from such a model is a key feature of our software which enables the user to effortlessly focus on a clinically important part from the whole image stack.

In addition to double clicking a certain portion of the model to highlight the region of interest, other cutting-based model creation tools will add more flexibility to our method. For example, Xiao et al (2010) ([17]) provided a generic model building

algorithm. By using a virtual dissection method, various models can be built quickly and efficiently. Together with our current model building tool, a highly flexible virtual dissection and selection based biological scene creating tool can be developed for studies and discoveries of disease patterns.

Acknowledgement

This work has been supported by Genome Canada through Genome Alberta; Alberta Science and Research Authority; Western Economic Diversification; the Governments of Canada and of Alberta through the Western Economic Partnership Agreement; the iCORE/Sun Microsystems Industrial Research Chair program; the Alberta Network for Proteomics Innovation; and the Canada Foundation for Innovation. We thank Heather Angka, Carl Helmick, Jordan Fisk and John Fisk for MRI data acquisition and processing. We also thank Megan Smith for comments on the manuscript.

References

- [1] M. Inglese, R. I. Grossman, M. Filippi, Magnetic resonance imaging monitoring of multiple sclerosis lesion evolution, *Journal of Neuroimaging* **15**(4 Suppl) (2005), 22S-29S.
- [2] F. Zipp, A new window in multiple sclerosis pathology: non-conventional quantitative magnetic resonance imaging outcomes, *Journal of the Neurological Sciences* **287**(1 Suppl) (2009), S24-S29.
- [3] H. C. Hege, M. Seebass, D. Stalling, M. Zöckler, A generalized marching cubes algorithm based on non-binary classifications, *Konrad-Zuse-Zentrum für Informationstechnik Berlin Technical Report SC-97-05* (1997).
- [4] W. E. Lorensen, H. E. Cline, Marching cubes: a high resolution 3D surface construction algorithm, *ACM SIGGRAPH Computer Graphics* **21**(4) (1987), 163-169.
- [5] T. Lweiner, H. Lopes, A. W. Vieira, G. Tavares, Efficient implementation of Marching Cubes' cases with topological guarantees, *Journal of Graphics Tools* **8**(2) (2003), 1-15.
- [6] G. M. Nielson, Dual marching cubes, In *Proc. IEEE Conf. Visualization* (2004), 489-496.
- [7] G. M. Nielson, On marching cubes. *IEEE Transactions on Visualization and Computer Graphics* **9**(3) (2003), 283-297.
- [8] S. Raman, R. Wenger, Quality isosurface mesh generation using an extended marching cubes lookup table, *Computer Graphics Forum* **27**(3) (2008), 791-798.
- [9] S. Schaefer, J. Warren, Dual marching cubes: primal contouring of dual grids, In *Proc. 12th Pacific Conf. Computer Graphics and Applications* (2004), 70-76.
- [10] M. del Fresno, M. Venere, A. Clausse, A combined region growing and deformable model method for extraction of closed surfaces in 3D CT and MRI scans. *Computerized Medical Imaging and Graphics* **33**(5) (2009), 369-376.
- [11] I. Cohen, D. Gordon, VS: a surface-based system for topological analysis, quantization and visualization of voxel data, *Medical Image Analysis* **13** (2009), 245-256.
- [12] B. Reitinger, A. Bornik, R. Beichel, Consistent mesh generation for non-binary medical datasets, In: *Bildverarbeitung für die Medizin* (2005), 183-187.
- [13] Y. Xi, Y. Duan, A novel region-growing based iso-surface extraction algorithm, *Computers & Graphics* **32**(6) (2008), 647-654.
- [14] S. Pieper, B. Lorensen, W. Schroeder, R. Kikinis, The NA-MIC Kit: ITK, VTK, Pipelines, Grids and 3D Slicer as an open platform for the medical image computing community, In: *Proc. 3rd IEEE Int. Symp. on Biomedical Imaging* (2006), 698-701.
- [15] W. Schroeder, K. Martin, B. Lorensen, *The Visualization Toolkit*, Prentice-Hall, 2006.
- [16] J. Schreiner, C. E. Scheidegger, C. T. Silva, High-quality extraction of isosurfaces from regular and irregular grid, *IEEE Transactions on Visualization and Computer Graphics* **12**(5) (2006), 1205-1212.
- [17] M. Xiao, J. Soh, O. Meruvia-Pastor, E. J. Schmidt, B. Hallgrímsson, C. W. Sensen, Building generic anatomical models using virtual model cutting and iterative registration, *BMC Medical Imaging* **10**(5) (2010).