Devorah Gurwitz

A Graphical Interface to Find and Display Lesions

August 21, 2008

1. Introduction -- About MS

Multiple sclerosis (MS) is a chronic disease of the nervous system. In patients with MS, the body directs antibodies at proteins in the myelin sheath, which protects the neurons in the central nervous system.[1] The resulting patches of inflammation are called lesions. The progression of the disease is often tracked by observing changes in the lesions. [2]

2. Previous Work – Finding Lesions Automatically

To aid in tracking this disease, Dr. Sturm and her colleagues devised an automatic segmentation algorithm. MRIs of brains stricken by MS are mapped into areas of white matter (WM), gray matter (GM), cerebral spinal fluid (CSF), and lesions. The algorithm takes into account partial volume effects, noise, and inhomogeneiety in dividing the brain.[3]

Using the segmented models, Dr. Sturm created a lesion finder to find significant lesions in the segmented MRIs. (Lesions are deemed significant if they are found in WM and are of above a given size threshold.) Upon detecting lesions, the finder automatically produces lesion seeds. Region growing procedures are then used to generate lesions throughout the slices. The produced image files contain lesions that can be browsed, measured, rendered, and rotated using an interactive user interface. [4]

The lesion browser runs as a plugin to ImageJ, an open-source, Java based image processing program developed by the National Institute of Health. [5]

3. My Work

3.1. Determining the Region of Interest

A major flaw of the lesion finder was that it searched each slice of the MRI, pixel by pixel. The actual brain image, at its largest point, only constitutes about a third of the MRI; the rest is merely back background. Searching through the background for lesions that could not possibly exist there added unnecessary processing time.

I wrote an algorithm to locate the approximate boundaries of the brain. Because a brain resembles an oblate spheroid, it is at its thickest point approximately on the middle slice, at the midpoints of the width and height. I searched the image down the midpoint lines of the middle

slice [Figure 1] to find the first non-black pixel on each side. The resulting four pixels were used to calculate the region of interest (ROI) of the MRI [Figure 2]. [6] The pixels outside the ROI (on all slides) are ignored while processing the image and searching for lesions.

Figure 1

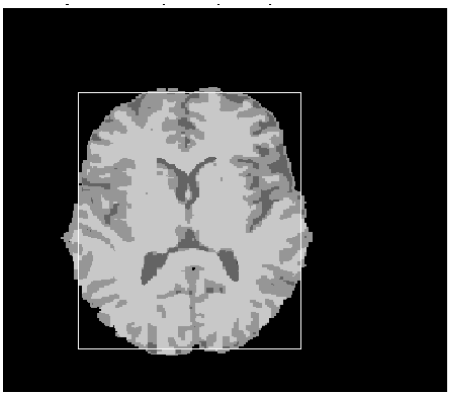

Figure 2

3.2. Creating the brain mask

Dr. Sturm's lesion finder generated separate image files for each lesion. If several lesions were found in one MRI, several new stacks would open, and the user would be required to sift through each one separately to view the lesions. Moreover, the lesions were depicted as "floating in space" – the lesion appeared as a single spot on a black background. The lack of reference points made it difficult to draw conclusions based on the lesions. On the other hand, the MRIs themselves were *too* detailed. The differing colors of WM, GM, and CSF prevented the user from focusing exclusively on the lesion.

ImageJ provides a "Find Edges" feature[7] that uses a Sobel edge filter[8] to find the edges of an image. When applied to the MRIs, however, the image that resulted was still too detailed, due to the differences in color between WM, GM, and CSF [Figure 3]. To produce just an outline of the original brain, I converted all the varying shades of gray to a uniform white, adapted a "Fill Holes" algorithm from the BinaryFiller class[9] to remove stray black spots, and inverted the image to produce a black outline on a white background.

The lesion stacks were all merged into one stack, which was then merged with the brain mask. This produced a single stack of all the lesions found in the MRI. The shape of the brain

provides reference points for comparison and diagnosis, but without the interfering details of the other parts of the brain [Figure 4].

Figure 3

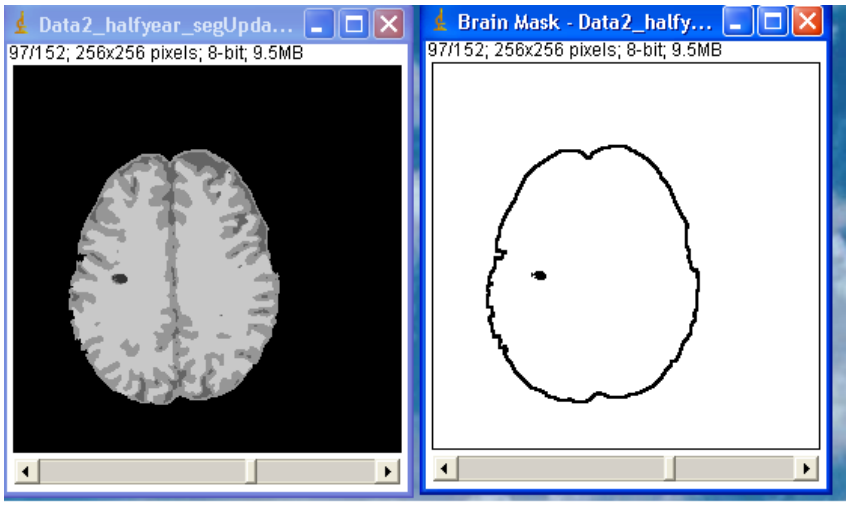

Figure 4

3.3. The Graphical User Interface

Because clinicians often compare a recent MRI to previous baseline ones, the browser was designed to work with several open images at once. Upon starting the program, if no images are open, the user is prompted to open one primary MRI (either by importing a raw image or by opening a standard image file). This MRI is the one that will be segmented and searched for

lesions. The user is then offered the option of opening multiple additional MRIs for comparison purposes. Once the images are opened, the browsing tools on the GUI allow the user to navigate all open stacks simultaneously [Figure 5].

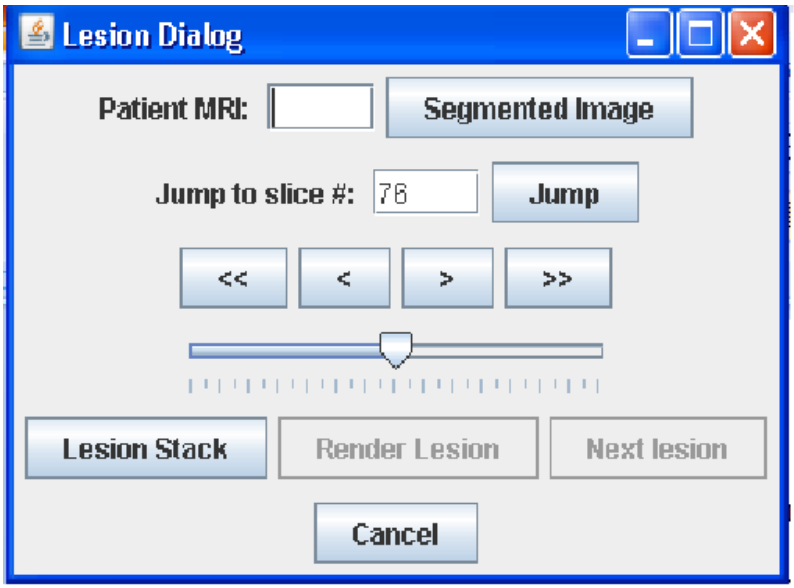

Figure 5

The "lesion stack" button generates the lesion finder on the primary MRI, finds the brain mask, and combines the image stacks and described above.

To enable users to navigate the lesions smoothly, a button is provided to navigate to the next lesion (on all open stacks and on the brain mask). Another allows the user to select a lesion by number, and prepares the lesion for rendering through an ImageJ plugin for volume rendering[10] [Figure 6].

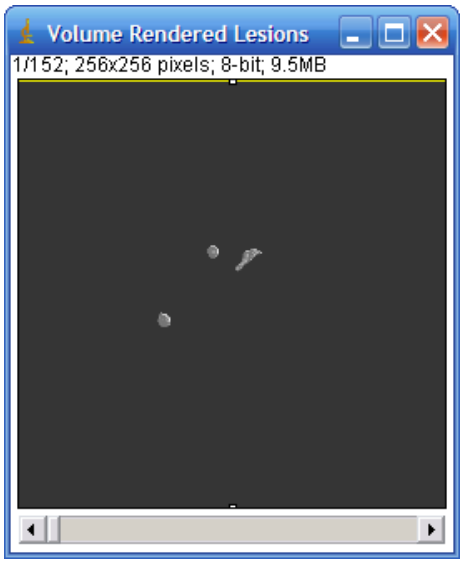

Figure 6

4. Conclusion

The resulting lesion browser is fully automatic. It segments the image, locates the lesions, and shows the stack of the lesions with the brain outline for reference without requiring user participation at all. All open images (including those opened after the program is in use) can be browsed together, providing medical clinicians with a one-glance comparison of old and new MRIs and lesion stacks. The browser is recognizably efficient in contrast to other commercial lesion finders, which require the user to select the regions of interest and visualize the lesions. [11] [12]

---

[1] "Multiple sclerosis: Definition - MayoClinic.com." *Mayo Clinic medical information and tools for healthy living - MayoClinic.com*. 20 Aug. 2008 <http://www.mayoclinic.com/health/multiple-sclerosis/DS00188>.

[2] "lesion - multiple sclerosis encyclopaedia." *All About Multiple Sclerosis : MS news and information on diagnosis, symptoms & treatments*. 20 Aug. 2008 <http://www.mult-sclerosis.org/lesion.html>.

[3] Li, L., Li, X., Wie, X., Sturm, D., Lu, H., Liang Z., *Quantitative analysis of multiple sclerosis: a feasibility study*, Medical Imaging 2006 (SPIE), Volume 6143.

[4] Sturm, Li, Liang, *Visualization of MS Lesions: a Browser Tool*, Computer-Aided Surgery, Medical Robotics, and Medical Imaging, ISRACAS'07, May 2007.

[5] Rasband, W.S., ImageJ, U.S. National Institute of Health, Bethseda, Maryland, USA, <http://rsb.info.nih.gov/ij, 1997-2006>.

[6] " Roi (ImageJ API) ." *RSB Home Page.* 20 Aug. 2008 <http://rsb.info.nih.gov/ij/developer/api/ij/gui/Roi.html>.

[7] " What is the algorithm used in Find Edges? — ImageJ Documentation."*Welcome to the ImageJ Information and Documentation Portal — ImageJ Documentation.* 21 Aug. 2008 <http://imagejdocu.tudor.lu/imagej-documentation-wiki/faq/what-is-the-algorithm-used-in-find-edges>.

[8] "Feature Detectors - Sobel Edge Detector." *Informatics Homepages Server*. 21 Aug. 2008 <http://homepages.inf.ed.ac.uk/rbf/HIPR2/sobel.htm>.

[9] "ImageJ (Java2HTML)." *RSB Home Page.* 21 Aug. 2008 <http://rsb.info.nih.gov/ij/docs/source/ij/plugin/filter/BinaryFiller.java.html>.

[10] Abramoff, Michael. "Bio-medical Imaging in Java." *Bio-medical Imaging in Java*. 20 Aug. 2008 <http://bij.isi.uu.nl/>

[11] The Multiple Sclerosis (MS) Lesion Finder Tool, <http://www.xinapse.com/Manual/ms_lesion_finder.html>

---

[12] Gurwitz, Li, Liang, Sturm, *Computer-Aided Tracking of MS Lesions – a Browser Tool*, submitted for consideration for Medical Imaging 2009 (SPIE).