SPAMALIZE's whole-brain segmentation algorithm (BrainStripper).

The following discussion assumes a MRI T1-weighted image volume, which has the following relationship between values for each type of tissue:
Background < CSF < GM < WM < blood.
Images other than a T1-weighted MRI image may have a different ranking for the relative tissue values, but as long as the desired tissue components are ranked contiguously this will not matter. Non-contiguously ranked tissue groups could be accomodated via an adjustment to the initial step in the algorithm, which sets undesired pixels to the background value.

The primary job of this algorithm is to create a volume bit-mask of the brain; the final step is to apply this mask to the original data and extract the image values for the brain only. Thus, a copy of the original data is maintained throughout the algorithm, and it is referred to at several stages. The mask is built up and refined throughout the algorithm. The following example assumes the MRI image volume has been scaled from 0-255, but this is not necessary (although it is faster).

This algorithm is intended to include all brain pixels, and may err slightly on the side of including some non-brain pixels. This balance could easily be shifted to exclude all non-brain pixels (at the expense of also perhaps excluding some true brain pixels) by having a less aggressive final recovery stage.

I. Define image values for brain.
  A. Select largest desired brain value (white matter, defined by WM/Top threshold).
  B. Select smallest desired brain value (grey matter, defined by CSF/GM threshold).

II. Set non-brain pixels to 0.
  A. Set background, CSF values to 0
  B. Set values higher than white matter (e.g. coronal arteries) to 0.

III. Apply axial mask
  A. Sum central 25% of axial slices, convert to bit-mask
  B. Fill holes, concave edges in mask
  C. Remove any pixels not connected to mask, dilate mask slightly
  D. Set all pixels outside of vertical column defined by axial mask to 0.
  E. Extract rectangular subset of data defined by axial mask

IV. Find the single largest connected blob
  A. Coronal
  B. Sagital
  C. Sum coronal, sagital masks

V. Remove isolated non-brain pixels
  A. Iteratively find three largest connect blobs in mask, exclude all others
    1. Axial
    2. Coronal
    3. Sagital

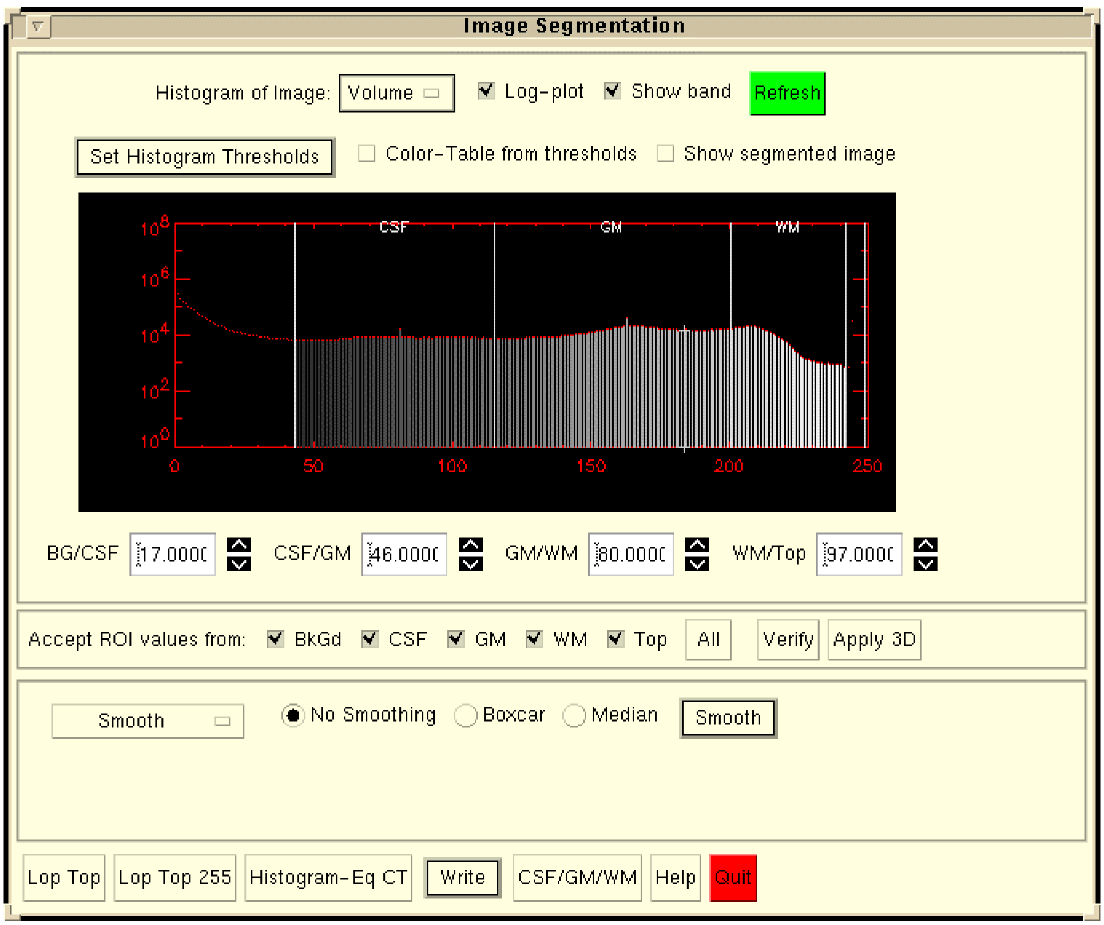
VI. Recover missing brain pixels.
  A. Operate in following order on mask from previous step:
    1. Coronal
    2. Sagital
    3. Axial
  B. Dilate mask, remove remaining isolated pixels
  C. Fill holes

VII. Apply mask to original data, extract values for pixels in brain.

Comments on individual steps:

Step I. Set tissue thresholds using a histogram.

Performed prior to beginning the actual whole-brain segmentation algorithm. It can be completed in a few seconds by having the user select the appropriate locations from a histogram of the entire image volume. Normally the histogram and thresholds should apear as follows:



For many of our MRI data, an associated text-file exists which contains previously deteremined thresholds. If this file exists, BrainStripper will automatically read and assign these threshold values. Otherwise, you have to find the thresholds yourself.

This step could be completely automated for data with similar histograms, such as T1-weighted MRI data volumes with the same size and acquisition pulse sequence. By selecting the button “Set Histogram Thresholds -> Automated”, BrainStripper will do its best to select the thresholds based on the shape of the histogram and a few guidelines. If this does not work, or for any reason you want to change the threshold locations, there are several ways to do this. You can click on the histogram at the desired x-axis location (left mouse button:CSF/GM; middle mouse button:GM/WM; right mouse button:WM/Top). You can also enter a number into any of the text-boxes below the histogram, or just click on the text-box arrows to change any of the thresholds by a value of 1.0.

To get to the BrainStripper menu, click on the button which reads “Smooth” when the menu appears, and select “Find Brain” from the pull-down menu. Clicking on “Go” from the “Find Brain” menu will launch the BrainStripper program, which performs the following steps:

Step II.
This step breaks many links between the brain and the skull, and also removes the brightest structures (e.g. coronal arteries), which tend to use 30-50% of the available pixel values. This step also provides a noise-free background, and aids in separating "ghost" artifacts from the main object.

Step III.
Applying an axial mask aids in excluding the sides, front, and back of the skull/scalp, which facilitates subsequent coronal and sagital searches for the largest connected object. If the sides of the brain are intact, a sagital search tends to find the scalp instead of the brain in sagital slices near the edge of the brain. Extracting a subset of the data is not necessary for the algorithm, but reduces the execution time by 20-50% (depending on the fraction of the volume occupied by the brain), since the excluded pixels are no longer searched etc.

Step IV.
The desired brain object is occasionally missed, or only a subset is found, in each orientation. By combining the results of searches in two orthogonal orientations, the size of the missing holes is greatly reduced. At this stage, 90% or more of the brain has been found, but there are usually undesired groups of pixels outside the brain, pieces of skull/scalp, and missing sections of brain that must be addressed.

Step V.
Each search only acts on the mask developed during previous searches, so pixels are only removed, not added, during this stage. The three largest blobs are found and included in the mask. By searching for 3 blobs, the frontal lobes and cerebellum can be included in axial views, both hemispheres can be included in coronal views, and fewer areas are excluded near the midline in sagital views. After this stage, few non-brain pixels remain, but there may be isolated holes in each of the orthogonal orientations, and the outer edges of the brain my be truncated in several locations.

Step VI.
The same mask from Step V is used for each search, to prevent the final mask from growing too much in one dimension. Holes in the mask are filled, outliers are removed, and the mask is applied to the original data. The masked data are compared to the corresponding slice in the output array

(which is empty for the first pass) and the logical OR of all non-zero pixels is placed back into the output array. By iteratively gathering true brain pixels, holes in each orthogonal dimenson are filled, pixels near and just outside of the brain edges are recovered, and the retained brain volume does not grow to include previously excluded skull/scalp in any dimension. At this point, the non-zero pixels include the entire brain, completetly contained non-brain volumes (ventricles), and some CSF surrounding the brain. For 3-dimensional rendered viewing, the surrounding CSF can easily be thresholded away.

<u>Description of morphological operator subroutines used by BrainStripper:</u>
DILATE
ERODE
SPAM_SMOOTH_IMG
SEG_FILL_CONVEX
SEG_BREAK_ISTHMUS
SEG_LARGEST_BLOB
SEG_FIND_THREE_BLOBS
SEG_CLEAN_IMG
SEG_FILL_HOLES

DILATE
Fill in all interior holes smaller than a kernal (typically 3x3 pixels) expand the edges of an object, and increase the thickness of a line or point by the amount of the kernal size. Dilation can bridge narrow gaps between isolated objects, smooth edges, and increase the size of an object.

ERODE
Remove isolated groups of pixels smaller than the kernal size (typicallyu 3x3 pixels) in any dimension, expand the size of interior holes, and reduce the size of larger objects. Erosion can break narrow isthmus connecting otherwise isolated objects, remove small objects, smooth edges, and reduce the size of an object.

An erosion/dilation or a dilation/erosion will leave objects larger than the kernal size unaffected, but will remove isolated objects or reshape features smaller than the kernal size. The order is important; different effects are achieved by using erosion or dilation first.

SPAM_SMOOTH_IMG
A 2D median filter with kernal size 3x3 pixels. Removes isolated "salt-and-pepper" spots and local minima, maxima while preserving edges.

SEG_FILL_CONVEX
Fills interior holes, making edges convex. Makes a bit-mask by searching for first, last non-zero element in each row, then sets all intervening elements to "1". Repeats for all columns. Does not search for isolated holes or edge concavities, but can fill very large holes. This is useful when the size of the hole may be very large compared to the size of the image, where a dilation operator may not work as expected.

SEG_BREAK_ISTHMUS
Erode, then dilate with a 2D 5x5 kernal.

SEG_LARGEST_BLOB
Finds the largest single connected group of pixels within an image. Searches between lowest GM value and highest WM value for all unique globs. A glob is defined as a group of pixels that are spatially connected to one another and have values between the specified thresholds. Two globs are separated by pixels outside of the thresholds, with no connectivity path linking them. The globs are sorted according to the number of pixels in each glob. The largest glob is tested to see if it corresponds to the background; if it does, the next largest glob is tested, and so on. The pixel locations from the first non-background glob are used to create a mask, which is then applied to the original data. All non-glob pixels are set to the background value, so the returned image contains only the pixel values corresponding to the largest non-background glob. This program does not perfrom any morphological (dilation/erosion) operations.

SEG_FIND_THREE_BLOBS
Finds the three largest non-background blobs in an image. Rejects any blobs with less than 16 pixels, so that small isolated blobs are excluded.

SEG_CLEAN_IMG
Erode, then dilate with a 3x3 kernal, and then dilate again with a 3x3 kernal. This removes outlying small groups of pixels, but preserves objects larger than the kernal and also preserves pixels near but outside of these larger objects. In some situations a larger kernal is used (e.g. 5x5), and a smaller kernal may be used for the second dilation in order to avoid including too many background pixels. A preliminary erosion/dilation may be employed to more thoroughly remove isolated groups. Using two erosion/dilation pairs with smaller kernals works better in some situations than using a single erosion/dilation operator with a larger kernal.

SEG_FILL_HOLES
Dilate, then erode the image using a kernal size of 7x7 for a maximum brain dimension of 96 pixels. A larger kernal is used for images with a maximum brain dimension larger than 96 pixels.