Reproducible Ventricle Segmentation Using Insight Toolkit

Release 0.00

Patrick Marion

October 18, 2007

marioj@rpi.edu

Abstract

This technical report describes how to use Insight Toolkit (www.itk.org) to perform elementary ventricle segmentation on data from the Designed Database of MR Brain Images of Healthy Volunteers hosted at MIDAS (http://insight-journal.org/dspace/).

Contents

1 Preliminaries **1**

2 Reading the code **2**

3 Running the code **3**

4 Results **3**

This paper has been written as part of a homework assignment for the class Open Source Software Practice taught the fall semester of 2007 at Rensselaer Polytechnic Institute. The motivation of the homework assignment is to practice open source with open data.

1 Preliminaries

The procedure described by this report is reproducible given the following resources:

- Insight Toolkit 3.4
- CMake 2.4
- MR image data from MIDAS

Source code implementing the segmentation procedure is distributed with this paper but the data is not. The data set used by the author may be downloaded freely from MIDAS at http://hdl.handle.net/1926/659 [2] [1]. The data should have the filename Normal012-T2.mha. Download it and place it in the source directory along with SegmentVentricles.cxx.

2 Reading the code

Ventricle segmentation is implemented by code in the file SegmentVentricles.cxx. The segmentation is carried out by the ConnectedThresholdImageFilter class. Next, a slice of the segmented data is saved to a PNG file so it may be compared with a baseline image. The slice is selected using the ExtractImageFilter class. Documentation of these classes may be found online:

- itk::ConnectedThresholdImageFilter
- itk::ExtractImageFilter

The ConnectedThresholdImageFilter requires 4 parameters: a seed point (in three dimensions), an upper and lower bound, and a replace value. Starting from the seed point, the filter chooses connected voxels with values inside the bounds and replaces them with the replace value. In my implementation, the replace value is hard coded to be 255. The remaining parameters are provided as program input on the command line. The parameters are set in code like this:

```
connectedThreshold->SetSeed( seedIndex ); //seedIndex previously defined
connectedThreshold->SetLower( atoi( argv[LowerBound] ) );
connectedThreshold->SetUpper( atoi( argv[UpperBound] ) );
connectedThreshold->SetReplaceValue( 255 );
```

Next, the data flow pipeline is constructed. The pipeline is very simple, a data reader is connected to the filter which is connected to a data writer. More information about data flow behavior may be found in the Insight Toolkit Software Guide [3]. The connections are implemented in code like this:

```
   connectedThreshold->SetInput( reader->GetOutput() );
   writer->SetInput( connectedThreshold->GetOutput() );
```

Finally, a slice of the segmented data is extracted and saved as a PNG image file. The purpose of generating a PNG file is so that an image comparison test may be carried out between segmented data and a baseline image provided by the author.

The ExtractImageFilter class collapses the output of the ConnectedThresholdImageFilter to two dimensions. The Z dimension is collapsed to the Z index of the seed point provided to the ConnectedThresholdImageFilter. In code, this look like:

```
   extractFilter->SetInput( connectedThreshold->GetOutput() );

   region.SetIndex(sliceIndex);
   region.SetSize(sliceSize);
   extractFilter->SetExtractionRegion(region);
```

Segmented data and the PNG image are written to disk using the `Update()` method of the `ImageFileWriter` class:

```
   dataWriter->Update();
   sliceWriter->Update();
```

3 Running the code

If you have downloaded the data used by the author, then simply configure CMake and generate, then build and test the project. If you have a different data set, or a different filename, then the CMake variable `DataFileName` must be redefined at the CMake configuration stage.

The file `CMakeLists.txt` defines two tests:

- SegmentVentricles
- ImageCompare

SegmentVentricles is initially disabled by CMake. The test is disabled because the data has not been distributed with the code. After you download the data, you may enable the SegmentVentricles test during the CMake configuration stage. If enabled, SegmentVentricles will be run as if from the command line:

```
./SegmentVentricle Normal012-T2.mha Segmented_Normal012-T2.mha 86.5 136.5 71 1200 2100
```

In the above command, [86.6 135.5 71] are the coordinates of a seed point. 1200 and 2100 are lower and upper bounds to the threshold filter. Note, SegmentVentricles outputs a PNG file in addition to the segmented data output.

The ImageCompare test will compare the PNG output by SegmentVentricles to a baseline image distributed with this report.

```
./ImageCompare Segmented_Normal012-T2.mha.png baseline.png
```

4 Results

`ConnectedThresholdImageFilter` is capable of segmenting brain ventricles, but requires a seed point and bounds selected by hand. The segmented region captures the general shape of the ventricles but is far from perfect. Therefore the segmentation procedure detailed in this report should be considered educational, but not a viable method for real segmentation.

References

[1] E Bullitt, G Gerig, S Aylward, S Joshi, JK Smith, W Lin, and MG Ewend. Vessel tortuosity and brain tumor malignancy: A blinded study. *Academic Radiology*, 12:1232–1240, 2005. 1

[2] Elizabeth Bullitt. Normal-012-t2. http://hdl.handle.net/1926/659, 2007. 1

[3] L. Ibanez, W. Schroeder, L. Ng, and J. Cates. *The ITK Software Guide*. Kitware, Inc. ISBN 1-930934-10-6, http://www.itk.org/ItkSoftwareGuide.pdf, first edition, 2003. 2