Confidence Connected Segmentation of the Lateral Ventricles

Release 0.01

Philip Radin[1]

October 17, 2007

[1]Rensselaer Polytechnic Institute, Troy, N.Y.

Abstract

This document describes an implementation of the Confidence Connected Segmentation algorithm used to segment the Lateral Ventricles of a healthy volunteer. The algorithm is used as described in the the Insight Toolkit ITK www.itk.org. The code of the algorithm is written following the ITK CodingStyle as described in the directory `Insight/Documentation/Style.pdf`

This paper is accompanied with the source code, input data, parameters and output data that the authors used for validating the algorithm described in this paper. This adheres to the fundamental principle that scientific publications must facilitate reproducibility of the reported results.

Contents

1 Introduction

Lateral Ventricles are two curved shaped structures located within the fore section of the brain. [4] They contain a substance known as Cerebrospinal fluid. This fluid provides a protective cushioning feature to the brain. Each of the two lateral ventricles is composed of anterior, posterior, and inferior horns. [2]

Using filters from the Insight Toolkit, we developed an application to segment the lateral ventricles from the MRI dataset of a healthy volunteer.

2 Experiment

We have attempted to segment the lateral ventricles using the ITK provided Confidence Connected algorithm. An ITK application was used to explore the various input parameters to produce an image of acceptable presentation. Input MRI data of healthy individual was provided through the MIDAS repository resulting in a segmented image.

3 Methods Used

The following ITK code was used to implement the Confidence Connected Region Growing algorithm. `itk::ConfidenceConnectedImageFilter` was used as input filter with the following parameters set. Input data[1] was obtained via MIDAS at Kitware. The provided ITK filter processed the input image resulting in an output image.

$$confidenceConnected->SetMultiplier(2.5);$$
$$confidenceConnected->SetNumberOfIterations(5);$$
$$confidenceConnected->SetReplaceValue(255);$$
$$confidenceConnected->SetInitialNeighborhoodRadius(2);$$

```
#if defined(_MSC_VER)
#pragma warning ( disable : 4786 )
#endif

#ifdef __BORLANDC__
#define ITK_LEAN_AND_MEAN
#endif
#include "itkConfidenceConnectedImageFilter.h"
#include "itkImage.h"
#include "itkCastImageFilter.h"
#include "itkCurvatureFlowImageFilter.h"

#include "itkImageFileReader.h"
#include "itkImageFileWriter.h"


int main( int argc, char *argv[] )
{
  if( argc < 5 )
```

```
    {
    std::cerr << "Missing Parameters " << std::endl;
    std::cerr << "Usage: " << argv[0];
    std::cerr << " inputImage  outputImage seedX seedY seedZ" << std::endl;
    return 1;
    }

  typedef   unsigned short           InternalPixelType;
  const     unsigned int    Dimension = 3;
  typedef itk::Image< InternalPixelType, Dimension >  InternalImageType;
  typedef unsigned short OutputPixelType;
  typedef itk::Image< OutputPixelType, Dimension > OutputImageType;

  typedef  itk::ImageFileReader< InternalImageType > ReaderType;
  typedef  itk::ImageFileWriter<  OutputImageType  > WriterType;

  ReaderType::Pointer reader = ReaderType::New();
  WriterType::Pointer writer = WriterType::New();

  reader->SetFileName( argv[1] );
  writer->SetFileName( argv[2] );

  typedef itk::ConfidenceConnectedImageFilter<InternalImageType, InternalImageType>
    ConnectedFilterType;
  ConnectedFilterType::Pointer confidenceConnected = ConnectedFilterType::New();

  confidenceConnected->SetInput( reader->GetOutput() );
  writer->SetInput( confidenceConnected->GetOutput() );

  confidenceConnected->SetMultiplier( 2.5 );
  confidenceConnected->SetNumberOfIterations( 5 );
  confidenceConnected->SetReplaceValue( 255 );

  InternalImageType::IndexType  index;

  index[0] = atoi( argv[3] );
  index[1] = atoi( argv[4] );
  index[2] = atoi( argv[5] );

  confidenceConnected->SetSeed( index );
  confidenceConnected->SetInitialNeighborhoodRadius( 2 );

  try
    {
    writer->Update();
    }
  catch( itk::ExceptionObject & excep )
    {
    std::cerr << "Exception caught !" << std::endl;
    std::cerr << excep << std::endl;
    }

  return 0;
}
```

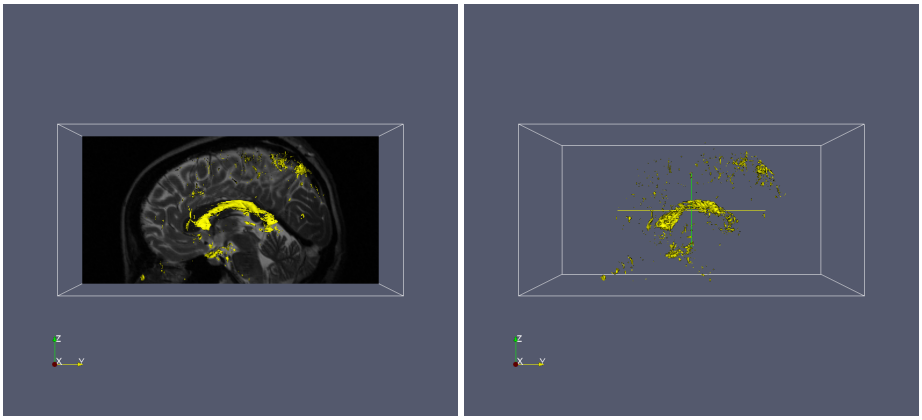
4 Instructions to Reproduce

CMake is executed in the local directory of the source to produce a build file for the local system. At this time, the source is then compiled resulting in an executable ConfidenceConnectedSegmentation for a unix environment or ConfidenceConnectedSegmentation.exe for the corresponding Win32 environment. ConfidenceConnectedSegmentation takes 5 input parameters `<inputimage> <outputimage> <xseed> <yseed> <zseed>`. The seed coordinates werew obtained by searching for white matter in a slice presented by ImageViewer.

The following results were obtained by inputing the coordinates, $X = 1035$, $Y = 129.5$, $Z = 67.0$ to ConfidenceConnectedSegmentation. The datasets *Normal018-T2.mhd* and *output.mhd* map to the inputimage and outputimage parameters respectively. The resulting *output.mhd* is viewable in either ImageViewer.exe or ParaView 3.0.

5 Results

These images represent respective input images and output images used in this application of ConfidenceConnectedSegmentation using the provided parameters. The provided ITK algorithms [3] sufficiently segmented the intended region of our healthy volunteer brain, the lateral ventricles.



(a) Input Image (b) Output Image

Figure 1: Normal018-T2

6 Software Requirements

You need to have the following software installed:

- Insight Toolkit 3.4

- CMake 2.4

Note that other versions of the Insight Toolkit are also available in the testing framework of the Insight Journal. Please refer to the following page for details

http://www.insightsoftwareconsortium.org/wiki/index.php/IJ-Testing-Environment

References

[1] Elizabeth Bullitt. T2. *http://insight-journal.org/dspace/handle/123456789/695*, 2007. 2

[2] David H. Hubbel. *Eye, Brain, and Vision*. W. H. Freeman, 1995. (document)

[3] L. Ibanez, W. Schroeder, L. Ng, and J. Cates. *The ITK Software Guide*. Kitware, Inc. ISBN 1-930934-15-7, http://www.itk.org/ItkSoftwareGuide.pdf, second edition, 2005. 4

[4] Ph.D Keith A. Johnson, M.D. J. Alex Becker. Whole brain atlas. Technical report, Harvard, http://www.med.harvard.edu/AANLIB/home.html, 2007. (document)

Confidence Connected Segmentation of the Lateral Ventricles

Release 0.01

Philip Radin[1]

October 18, 2007

[1]Rensselaer Polytechnic Institute, Troy, N.Y.

Abstract

This document describes an implementation of the Confidence Connected Segmentation algorithm used to segment the Lateral Ventricles of a healthy volunteer. The algorithm is used as described in the the Insight Toolkit ITK www.itk.org. The code of the algorithm is written following the ITK CodingStyle as described in the directory `Insight/Documentation/Style.pdf`

This paper is accompanied with the source code, input data, parameters and output data that the authors used for validating the algorithm described in this paper. This adheres to the fundamental principle that scientific publications must facilitate reproducibility of the reported results.

Contents

1 Introduction

Lateral Ventricles are two curved shaped structures located within the fore section of the brain. [4] They contain a substance known as Cerebrospinal fluid. This fluid provides a protective cushioning feature to the brain. Each of the two lateral ventricles is composed of anterior, posterior, and inferior horns. [2]

Using filters from the Insight Toolkit, we developed an application to segment the lateral ventricles from the MRI dataset of a healthy volunteer.

2 Experiment

In this paper we attempt to segment the lateral ventricles using the ITK provided Confidence Connected algorithm. An ITK application was used to explore the various input parameters to produce an image of acceptable presentation. Input MRI data of healthy individual was provided through the MIDAS repository resulting in a segmented image.

3 Methods Used

The following ITK code was used to implement the Confidence Connected Region Growing algorithm. `itk::ConfidenceConnectedImageFilter` was used as input filter with the following parameters set. Input data[1] was obtained via MIDAS at Kitware. The provided ITK filter processed the input image resulting in an output image.

$$confidenceConnected->SetMultiplier(2.5);$$
$$confidenceConnected->SetNumberOfIterations(5);$$
$$confidenceConnected->SetReplaceValue(255);$$
$$confidenceConnected->SetInitialNeighborhoodRadius(2);$$

```
#if defined(_MSC_VER)
#pragma warning ( disable : 4786 )
#endif

#ifdef __BORLANDC__
#define ITK_LEAN_AND_MEAN
#endif
#include "itkConfidenceConnectedImageFilter.h"
#include "itkImage.h"

#include "itkExtractImageFilter.h"
#include "itkImageFileReader.h"
#include "itkImageFileWriter.h"


int main( int argc, char *argv[] )
{
  if( argc < 5 )
    {
```

```
    std::cerr << "Missing Parameters " << std::endl;
    std::cerr << "Usage: " << argv[0];
    std::cerr << " inputImage  outputImage seedX seedY seedZ" << std::endl;
    return 1;
    }

  typedef   unsigned short           InternalPixelType;
  const     unsigned int  Dimension = 3;
  typedef itk::Image< InternalPixelType, Dimension >  InternalImageType;
  typedef unsigned short OutputPixelType;
  typedef itk::Image< OutputPixelType, Dimension > OutputImageType;

  typedef  itk::ImageFileReader< InternalImageType > ReaderType;
  typedef  itk::ImageFileWriter<  OutputImageType  > WriterType;

  ReaderType::Pointer reader = ReaderType::New();
  WriterType::Pointer writer = WriterType::New();

  reader->SetFileName( argv[1] );
  writer->SetFileName( argv[2] );

  typedef itk::ConfidenceConnectedImageFilter<InternalImageType, InternalImageType>
    ConnectedFilterType;
  ConnectedFilterType::Pointer confidenceConnected = ConnectedFilterType::New();

  confidenceConnected->SetInput( reader->GetOutput() );
  writer->SetInput( confidenceConnected->GetOutput() );

  confidenceConnected->SetMultiplier( 2.5 );
  confidenceConnected->SetNumberOfIterations( 5 );
  confidenceConnected->SetReplaceValue( 255 );

  InternalImageType::IndexType  index;

  index[0] = atoi( argv[4] );
  index[1] = atoi( argv[5] );
  index[2] = atoi( argv[6] );

  confidenceConnected->SetSeed( index );
  confidenceConnected->SetInitialNeighborhoodRadius( 2 );

  try
    {
    writer->Update();
    }
  catch( itk::ExceptionObject & excep )
    {
    std::cerr << "Exception caught !" << std::endl;
    std::cerr << excep << std::endl;
    }

// PNG

typedef unsigned short Input2DPixelType;
```

```
typedef unsigned char Output2DPixelType;

typedef itk::Image< Input2DPixelType, 3 > Input2DImageType;
typedef itk::Image< Output2DPixelType, 2 > Output2DImageType;
typedef itk::ImageFileWriter< Output2DImageType > Writer2DType;

Writer2DType::Pointer writer2D = Writer2DType::New();

typedef itk::ExtractImageFilter< Input2DImageType, Output2DImageType > Filter2DType;
Filter2DType::Pointer filter2D = Filter2DType::New();

confidenceConnected->Update();
Input2DImageType::RegionType inputRegion = confidenceConnected->GetOutput()->GetLargestPossibleRegion()

Input2DImageType::SizeType size = inputRegion.GetSize();
size[2] = 0;

Input2DImageType::IndexType start = inputRegion.GetIndex();
const unsigned int sliceNumber = atoi( argv[6] );
start[0] = 0;
start[1] = 0;
start[2] = sliceNumber;

Input2DImageType::RegionType desiredRegion;
desiredRegion.SetSize( size );
desiredRegion.SetIndex( start );

filter2D->SetInput( confidenceConnected->GetOutput());
filter2D->SetExtractionRegion( desiredRegion );

writer2D->SetInput( filter2D->GetOutput() );
writer2D->SetFileName( argv[3] );

try
{
writer2D->Update();
}
catch( itk::ExceptionObject & err )
{
std::cerr << "ExceptionObject caught !" << std::endl;
std::cerr << err << std::endl;
return EXIT_FAILURE;
}

  return 0;
}
```

4 Instructions to Reproduce

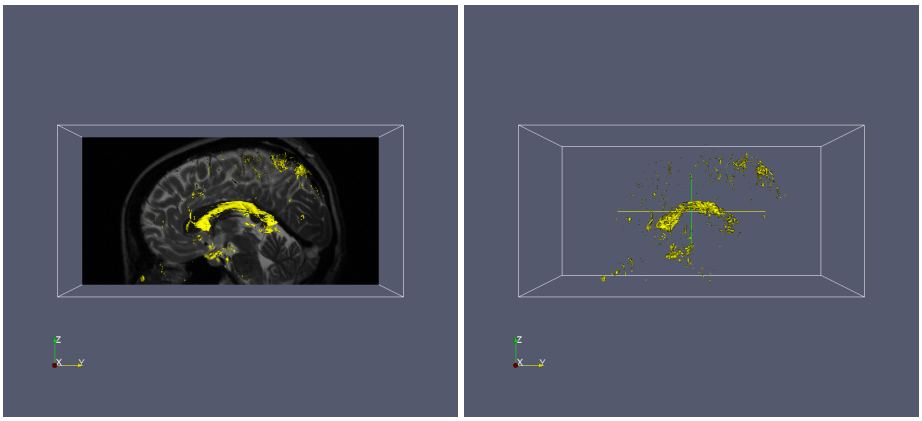
CMake is executed in the local directory of the source to produce a build file for the local system. At this time, the source is then compiled resulting in an executable ConfidenceConnectedSegmentation for a unix environment or ConfidenceConnectedSegmentation.exe for the corresponding Win32 environment. Con-

fidenceConnectedSegmentation takes 5 input parameters `<inputimage> <outputimage> <outputpng> <xseed> <yseed> <zseed>`. The seed coordinates are obtained by searching for white matter in a slice presented by ImageViewer.

The following results were obtained by inputing the coordinates, $X = 103.5$, $Y = 129.5$, $Z = 67.0$ to ConfidenceConnectedSegmentation. The datasets *Normal018-T2.mha* and *output.mha* map to the inputimage and outputimage parameters respectively. The resulting *output.mha* is viewable in either ImageViewer.exe or ParaView 3.0. A PNG file is produced representing Zth slice given as an argument at runtime.

5 Results

These images represent respective input images and output images used in this application of ConfidenceConnectedSegmentation using the provided parameters. The provided ITK algorithms [3] sufficiently segmented the intended region of our healthy volunteer brain, the lateral ventricles.



(a) Input Image (b) Output Image

Figure 1: Normal018-T2

6 Software Requirements

You need to have the following software installed:

- Insight Toolkit 3.4
- CMake 2.4

Note that other versions of the Insight Toolkit are also available in the testing framework of the Insight Journal. Please refer to the following page for details

http://www.insightsoftwareconsortium.org/wiki/index.php/IJ-Testing-Environment

References

[1] Elizabeth Bullitt. T2. *http://insight-journal.org/dspace/handle/123456789/695*, 2007. 3

[2] David H. Hubbel. *Eye, Brain, and Vision*. W. H. Freeman, 1995. 1

[3] L. Ibanez, W. Schroeder, L. Ng, and J. Cates. *The ITK Software Guide*. Kitware, Inc. ISBN 1-930934-15-7, http://www.itk.org/ItkSoftwareGuide.pdf, second edition, 2005. 5

[4] Ph.D Keith A. Johnson, M.D. J. Alex Becker. Whole brain atlas. Technical report, Harvard, http://www.med.harvard.edu/AANLIB/home.html, 2007. 1