Rescale Scalars Mesh Filter

Release 1.00

Wen Li[1,2], Vincent A. Magnotta[1,2,3]

August 25, 2010

[1]Department of Radiology, The University of Iowa, Iowa City, IA 52242
[2]Department of Biomedical Engineering, The University of Iowa, Iowa City, IA 52242
[3]Department of Psychiatry, The University of Iowa, Iowa City, IA 52242

Abstract

This documents is about the filter itk::RescaleScalarsQuadEdgeMeshFilter. It takes an input mesh and generates an output mesh with user specifying scalar values in the range of $[min, max]$.

This paper is accompanied with the source code, input data, parameters and output data that we 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.

Acknowledgements

This work was funded in part by NIH/NINDS award NS050568.

Contents

1 Introduction

This filter is to perform linear transformation of input mesh scalars, by using the minimum and maximum values as parameters passed through command line. It searches the original input mesh's minimum and maximum scalar values and changes the output mesh's scalar values into the range of $[min, max]$, where *min* and *max* are passed by the user via the command line.

2 How to Build

This contribution includes

- Rescale Scalars Filter
- Tests for the filter
- All the LaTeX source files of this paper

The source code is in Source directory and named as

- itkRescaleScalarsQuadEdgeMeshFilter.h
- itkRescaleScalarsQuadEdgeMeshFilter.txx

The testing code is in Testing directory as

- itkRescaleScalarsQuadEdgeMeshFilterTest.cxx

2.1 Building Executables and Tests

In order to build the whole, it is enough to configure the directory with CMake. As usual, an out-of-source build is the recommended method.

In a Linux environment it should be enough to do the following:

- `ccmake SOURCE_DIRECTORY`
- `make`
- `ctest`

Where SOURCE_DIRECTORY is the directory where you have expanded the source code that accompanies this paper.

This will configure the project, build the executables, and run the tests and examples.

2.2 Building this Report

In order to build this report you can do

- `ccmake SOURCE_DIRECTORY`
- Turn ON the CMake variable
  - `BUILD_REPORTS`
- `make`

This should produce a PDF file in the binary directory, under the subdirectory `Documents/Report007`.

3 How to Use the Filter

This section illustrates the minimum operations required for running the filter. The code shown here is available in the Testing directory of the code that accompanies this paper. You can download the entire set of files from the Insight Journal web site.

The source code presented in this section can be found in the `Testing` directory under the filename.

- itkRescaleScalarsQuadEdgeMeshFilterTest

In order to use this filter you should include the header for the Rescale Scalars filter, the reader and writer types and the `itk::QuadEdgeMesh` itself.

```
#include "itkQuadEdgeMesh.h"
#include "itkRescaleScalarsQuadEdgeMeshFilter.h"

#include "itkQuadEdgeMeshVTKPolyDataReader.h"
#include "itkQuadEdgeMeshScalarDataVTKPolyDataWriter.h"
```

The Scalar type associated with the nodes in the mesh, and the mesh dimension are defined in order to declare the Mesh type

```
  typedef double Coord;
  const unsigned int Dimension = 3;

  // Declaration of the type of Mesh
  typedef itk::QuadEdgeMesh< Coord, Dimension > MeshType;
```

Next, declare the type of the rescale scalars filter and instantiate it.

```
  typedef itk::RescaleScalarsQuadEdgeMeshFilter< MeshType > RescaleScalarsType;
  RescaleScalarsType::Pointer filter = RescaleScalarsType::New( );
```

In order to read the input mesh, you need to declare a reader type and create an instance of it.

```
  // Here read a mesh
  typedef itk::QuadEdgeMeshVTKPolyDataReader< MeshType >  ReaderType;
  ReaderType::Pointer reader = ReaderType::New();
```

The output of the reader is passed as input to the rescale scalars filter.

```
  filter->SetInput( reader->GetOutput() );
```

The minimum and maximum scalar values of the output mesh are set by the user.

```
filter->SetOutputMinimum(atof(argv[3]));
filter->SetOutputMaximum(atof(argv[4]));
```

The execution of the filter can be triggered by calling the `Update()` method.

```
filter->Update();
```

The output of the filter can be passed to a writer.

```
typedef itk::QuadEdgeMeshScalarDataVTKPolyDataWriter< MeshType >  WriterType;
WriterType::Pointer writer = WriterType::New();
writer->SetInput( filter->GetOutput() );
writer->SetFileName(argv[2]);
writer->Update();
```

The minimum and maximum scalar values before and after the rescale scalars filter are printed out to screen.

```
std::cout<<"Input min: "<<filter->GetInputMinimum()<<std::endl;
std::cout<<"Input max: "<<filter->GetInputMaximum()<<std::endl;

std::cout<<"Output min: "<<filter->GetOutputMinimum()<<std::endl;
std::cout<<"Output max: "<<filter->GetOutputMaximum()<<std::endl;
```

The results in the following section were generated with calls similar to

```
RescaleScalarsQuadEdgeMesh inputMesh.vtk outputMesh.vtk \
0.0 1.0
```

4 Results

Figure 1 shows the input mesh. The scalar values of it are in the range of $[0.0, 1.0]$

Figure 2 shows the output mesh which is a copy of the input mesh but has the scalar values in the range of $[0.5, 1.0]$

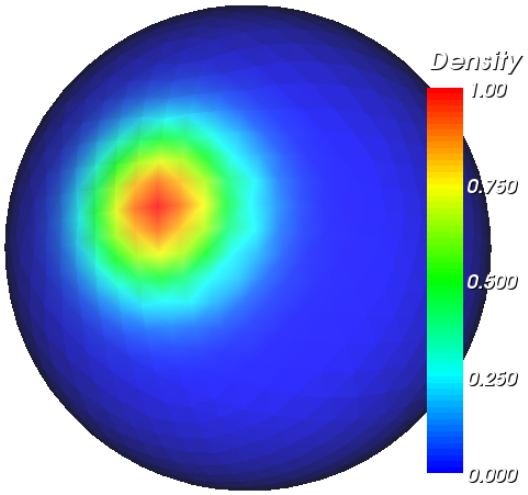

Figure 1: Input Mesh to the Rescale Values filter.

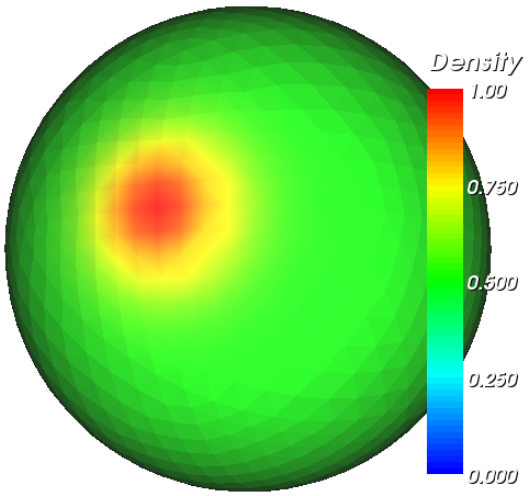

Figure 2: Output Mesh of the Rescale Values filter.

References