Scientific Data Visualization

CS880

Error Visualization in 3D data sets Using the Data-Error model :

Project Report

Submitted by:

Varsha Thirumakil (vt2000)

Under Supervision of:

Prof. R. Daniel Bergeron

UNIVERSITY OF NEW HAMPSHIRE, DURHAM

December 2014

CONTENTS

CHAPTER 1

INTRODUCTION

This report discusses the outcome of work done in development of "Error Visualization of 3D data sets" on Java for OpenGL (JOGL) Platform. The proposed project is about the visualization of error in different 3D data slices. The user can view different slices of the data sources in a very flexible manner that is designed using the MREBuilder. To give user interactiveness sliders are designed so that user can control color of the data slices to see them in various modes. The error data related to the data source is visualized efficiently. We have also designed a toggle button as part of user interface so that user can differentiate between the error data slice and the original data slice. The slices are positioned inside the box along x , y and z axes.

Its always been challenging for scientific researchers to draw conclusions from large scientific data that is displaying the terabyte or gigabyte of data sets interactively would be tedious task. Fortunately, these behavior can be recognized through low resolution views generated effectively using 3D data slices. They are eventually generated using MREBuilder.

As summary the goals of this project are:

- Design an error model for representing localized error informations from data reduction done using the MREBuilder.
- Extract the error data from a tool based on which generating multiple low resolutions with error information from a single data set is done.
- Visualize these representations by applying textures.
- Design sliders for user interactions which would permit user to do transformations and control color change.

CHAPTER 2

BACKGROUND

Overview of the terms used :

Data Source :

[1] The *PhysicalDatasource* class provides the key interface to physical files; it is the first point of access to the data. Granite does not predefine any particular data format for data; the intent is to access the original data as generated by the scientist's application. Consequently, it is necessary to provide a metadata file that describes how that data is organized. The metadata file must describe both *file* and *data* attributes and is written in XML notation; we call these *fdl* files (fdl is an abbreviation for *file definition language*). These data which is used to analyze or perform different computations is called data sources.

Data Slice :

The 3D data source is large and visualizing it is very difficult. So we can generate and view desired slices of the 3D data set. They make the job of the user easy. [1] The *ISBoundsIterator* supports block iteration over a datasource or a data block and helps in generating the slices as the user specifies the lower and upper bounds.

Multi resolution dataset :

The hierarchy of resolution of data set along with the error information is the multi resolution data. It is useful in analyzing the data given at different levels to get more detailed information about the data. Initially, the error and data have same size at the resolution as we go down the hierarchy lower resolutions are generated.

Error Visualization:

In a given data the visualization of the different types of error present at different resolution level are seen in error visualization.

Types of Errors :

Here we are dealing with different kinds of errors such as SNR, AVGABS, MAXABS, STDDEV etc., .

SNR : This is the signal to noise ratio error associated with the data slice.

Avgabs : This is the average absolute error of the data slice.

[2] Average absolute error can be used to reduce the impact of these high values in the error set.

Maxabs : This is the maximum absolute error of the data slice.

[2] Maximum absolute error captures the largest differences between the original and reconstructed data sets. Because of this, the values of large error regions can be dominated by a few high error values.

Stddev : This is the standard deviation error associated with the data slice.

MREBuilder : This is the tool which is used to generate the error files at different resolutions. Using this we can render the data with the error levels.

CHAPTER 3

PROJECT DESCRIPTION

The project is about the visualizing the errors at different resolution level and making it user interactive with adding GUI for the project , and positioning it in the 3D axes which makes it easy to view the slice in all the axes respectively.

3.1 Functionality :

The functionality is rendering the slice using the ViewSlice class and mapping it to the quad object in the 3D axes with the sliders for moving them in the X, Y and Z axes. The slices can be translated, rotated and scaled accordingly . There is also the toggle panel which helps the user to switch between the original data source and also the error data of specific data set. In addition, there is an alpha panel which shows the color change in the slices rendered.

3.2 Approach :

Map the data slice to a Quad object

To test the working of texture mapping to the quad, as a first step we mapped dataset generated to the cube.

-> -> (texture representing the slice of the actual data set)

We used the Buffered Image subclass(image with the accessible buffer of image data).

From this subclass we get a BufferedImage object .

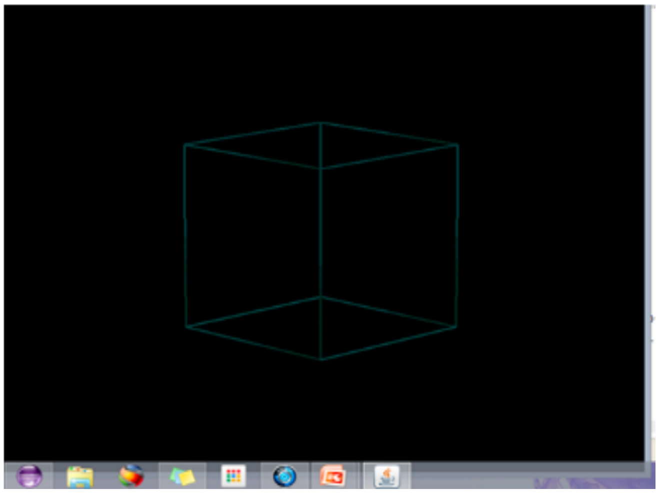
This object will be used along with AWTTextureIO class , which will in turn map the desired object with the dataslice.

We designed the user interface with sliders , radiobutton and the checkboxes. Using the MREBuilder tool the error files are generated for the data sources at various resolution . The error resolutions are specified as the e1,e2 and so on . They also generate files of different types of errors such as avgabs, snr and stdev. These are later mapped on to the quad object on the canvas in the cube axes.

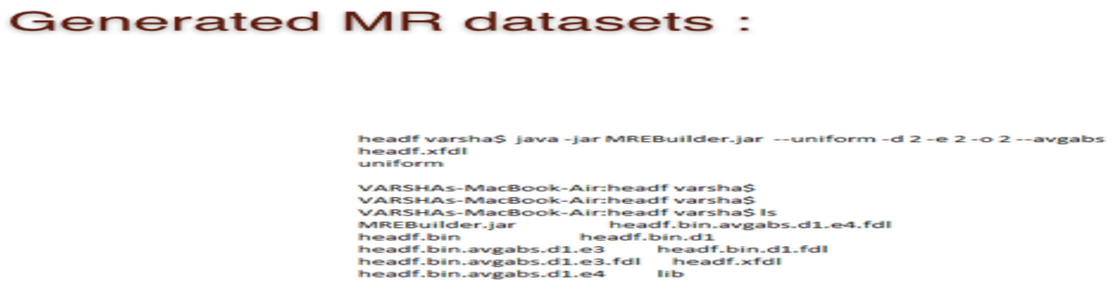
3.3 Detailed Specifications

Box Axes :

To represent the slices rendered.



MR Data Set :

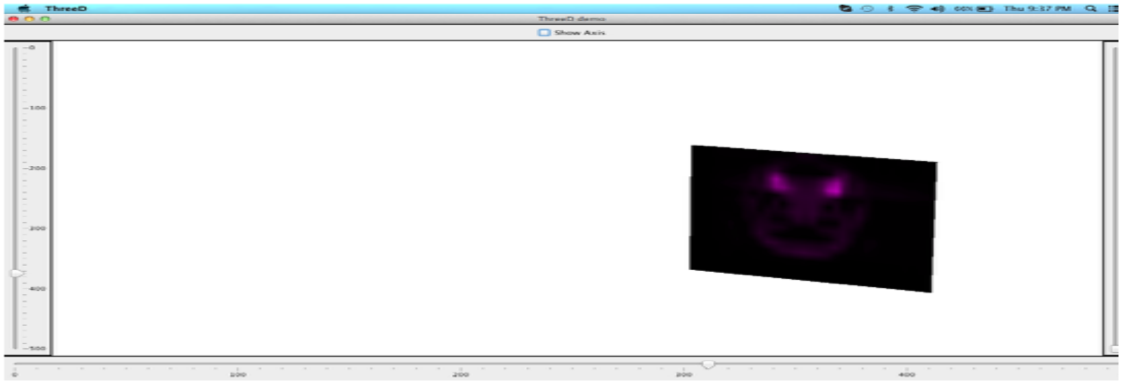


Generated MR datasets :

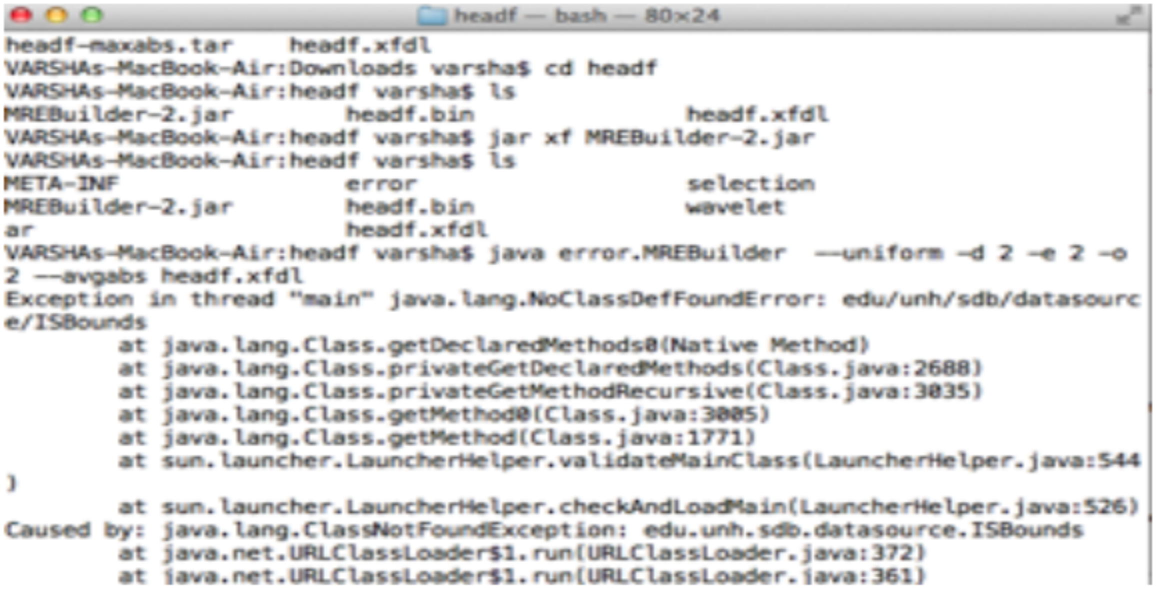
```
headf varsha$  java -jar MREBuilder.jar  --uniform -d 2 -e 2 -o 2 --avgabs
headf.xfdl
uniform

VARSHAs-MacBook-Air:headf varsha$
VARSHAs-MacBook-Air:headf varsha$
VARSHAs-MacBook-Air:headf varsha$ ls
MREBuilder.jar                  headf.bin.avgabs.d1.e4.fdl
headf.bin                   headf.bin.d1
headf.bin.avgabs.d1.e3          headf.bin.d1.fdl
headf.bin.avgabs.d1.e3.fdl       headf.xfdl
headf.bin.avgabs.d1.e4          lib
```

data/headf.bin.avgabs.d1.e3.fdl

(error data – average absolute)



Issues faced while working with MREBuilder

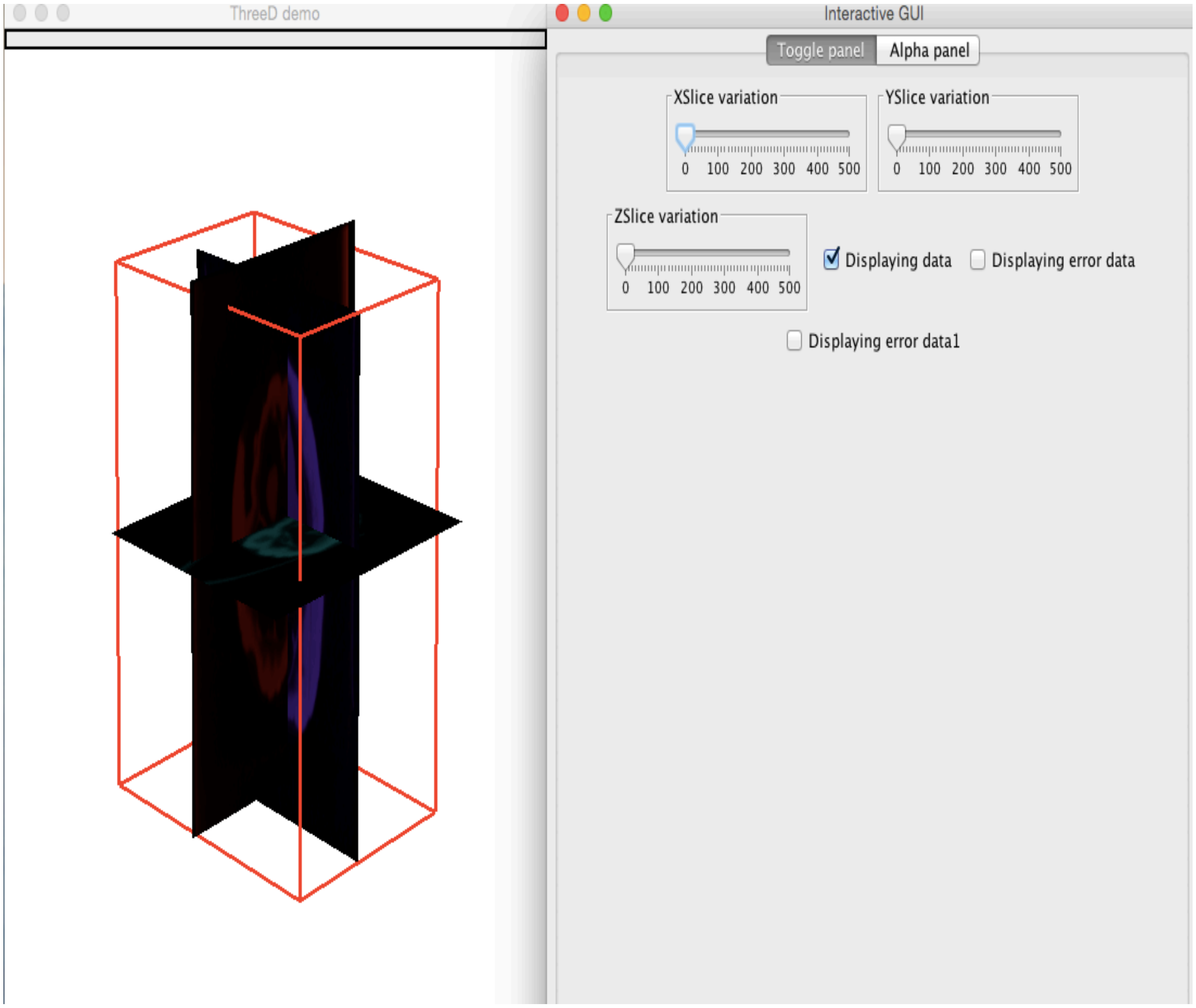


```
headf-maxabs.tar    headf.xfdl
VARSHAs-MacBook-Air:Downloads varsha$ cd headf
VARSHAs-MacBook-Air:headf varsha$ ls
MREBuilder-2.jar        headf.bin                       headf.xfdl
VARSHAs-MacBook-Air:headf varsha$ jar xf MREBuilder-2.jar
VARSHAs-MacBook-Air:headf varsha$ ls
META-INF                error                           selection
MREBuilder-2.jar        headf.bin                       wavelet
ar                      headf.xfdl
VARSHAs-MacBook-Air:headf varsha$ java error.MREBuilder  --uniform -d 2 -e 2 -o
2 --avgabs headf.xfdl
Exception in thread "main" java.lang.NoClassDefFoundError: edu/unh/sdb/datasourc
e/ISBounds
        at java.lang.Class.getDeclaredMethods0(Native Method)
        at java.lang.Class.privateGetDeclaredMethods(Class.java:2688)
        at java.lang.Class.privateGetMethodRecursive(Class.java:3035)
        at java.lang.Class.getMethod0(Class.java:3005)
        at java.lang.Class.getMethod(Class.java:1771)
        at sun.launcher.LauncherHelper.validateMainClass(LauncherHelper.java:544
)
        at sun.launcher.LauncherHelper.checkAndLoadMain(LauncherHelper.java:526)
Caused by: java.lang.ClassNotFoundException: edu.unh.sdb.datasource.ISBounds
        at java.net.URLClassLoader$1.run(URLClassLoader.java:372)
        at java.net.URLClassLoader$1.run(URLClassLoader.java:361)
```

CHAPTER 4

RESULTS

<u>Displaying the walnut data slice along the GUI required to view desired slice</u>

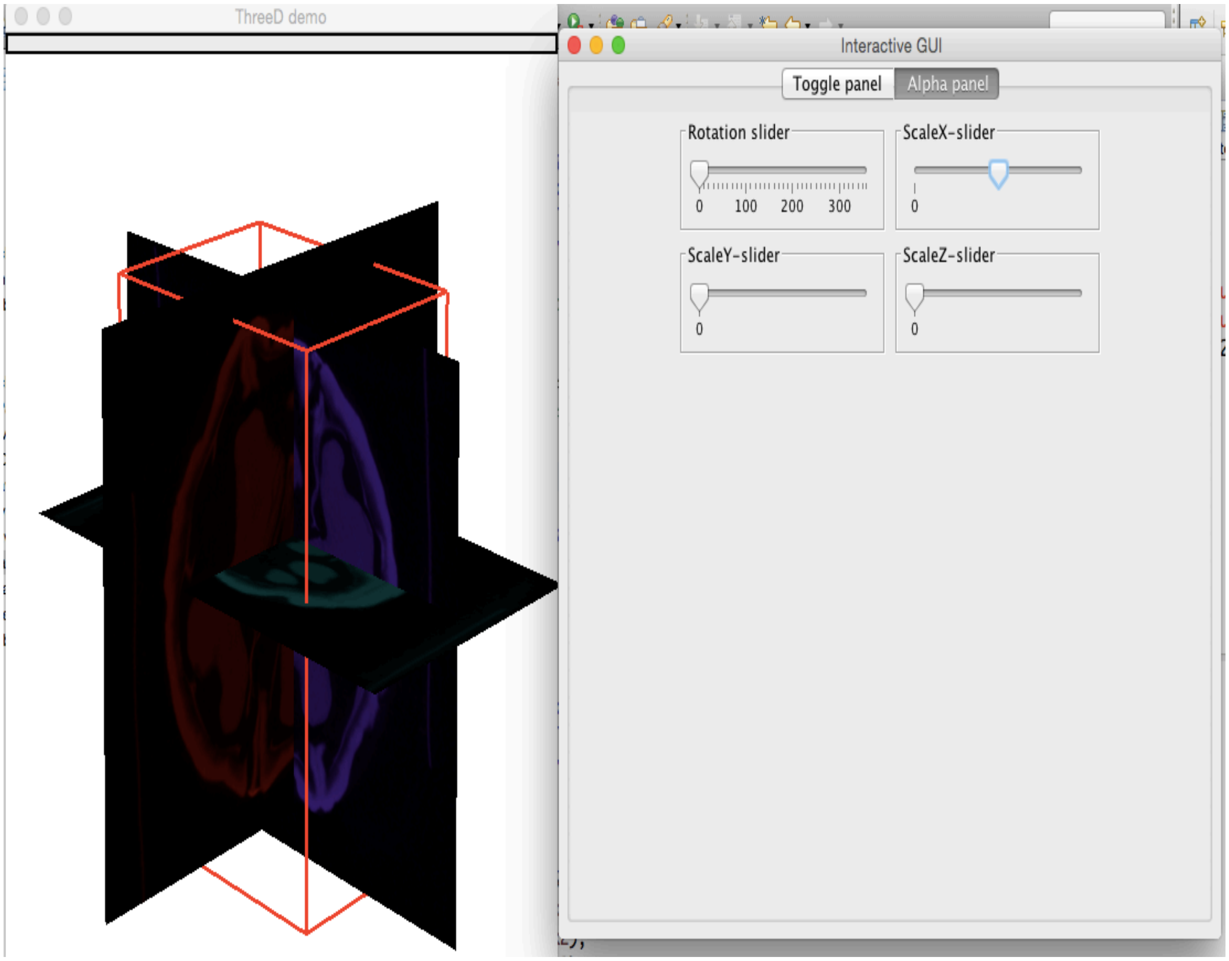


☑ Displaying data

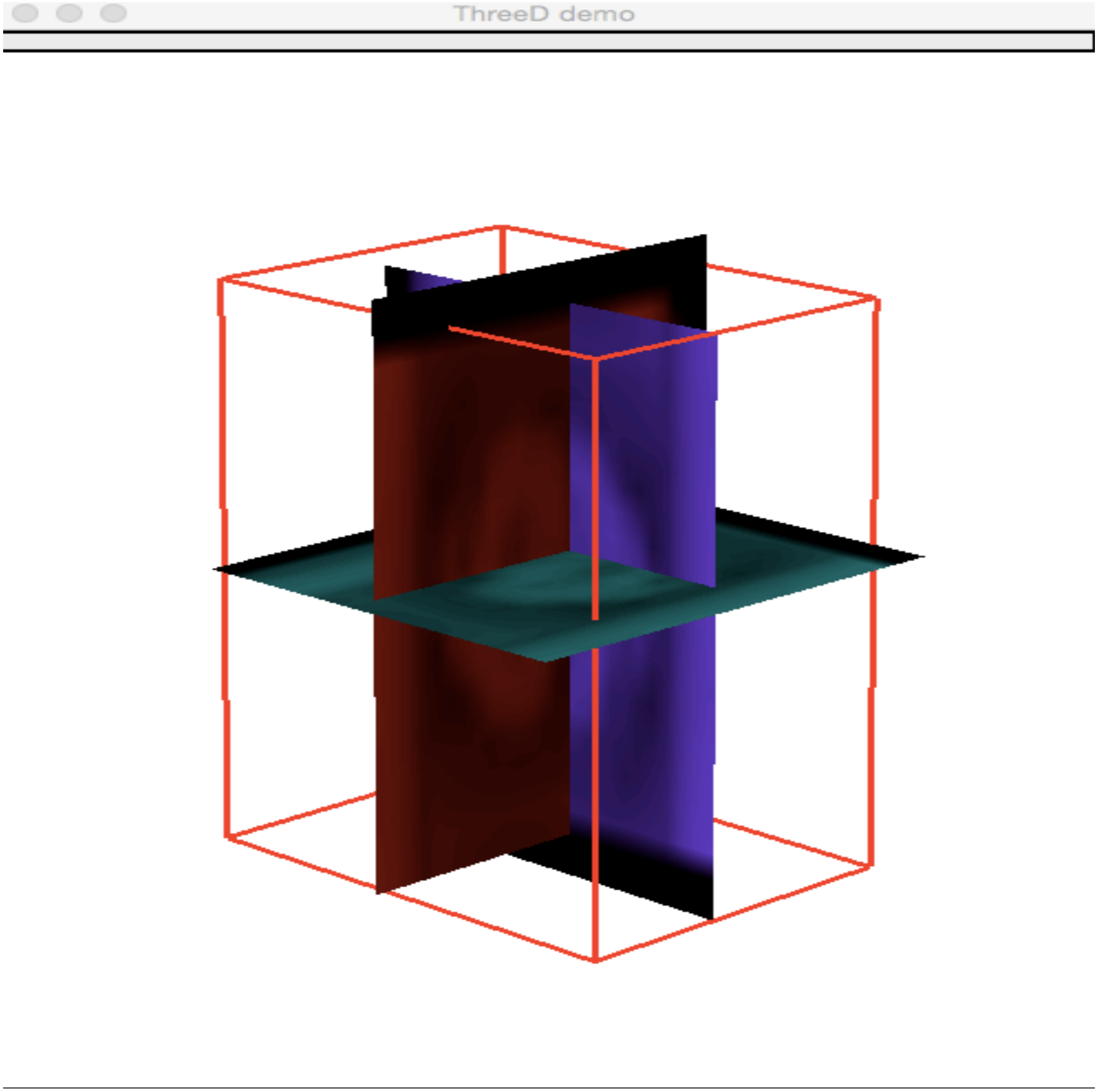
□ Displaying error data

□ Displaying error data1

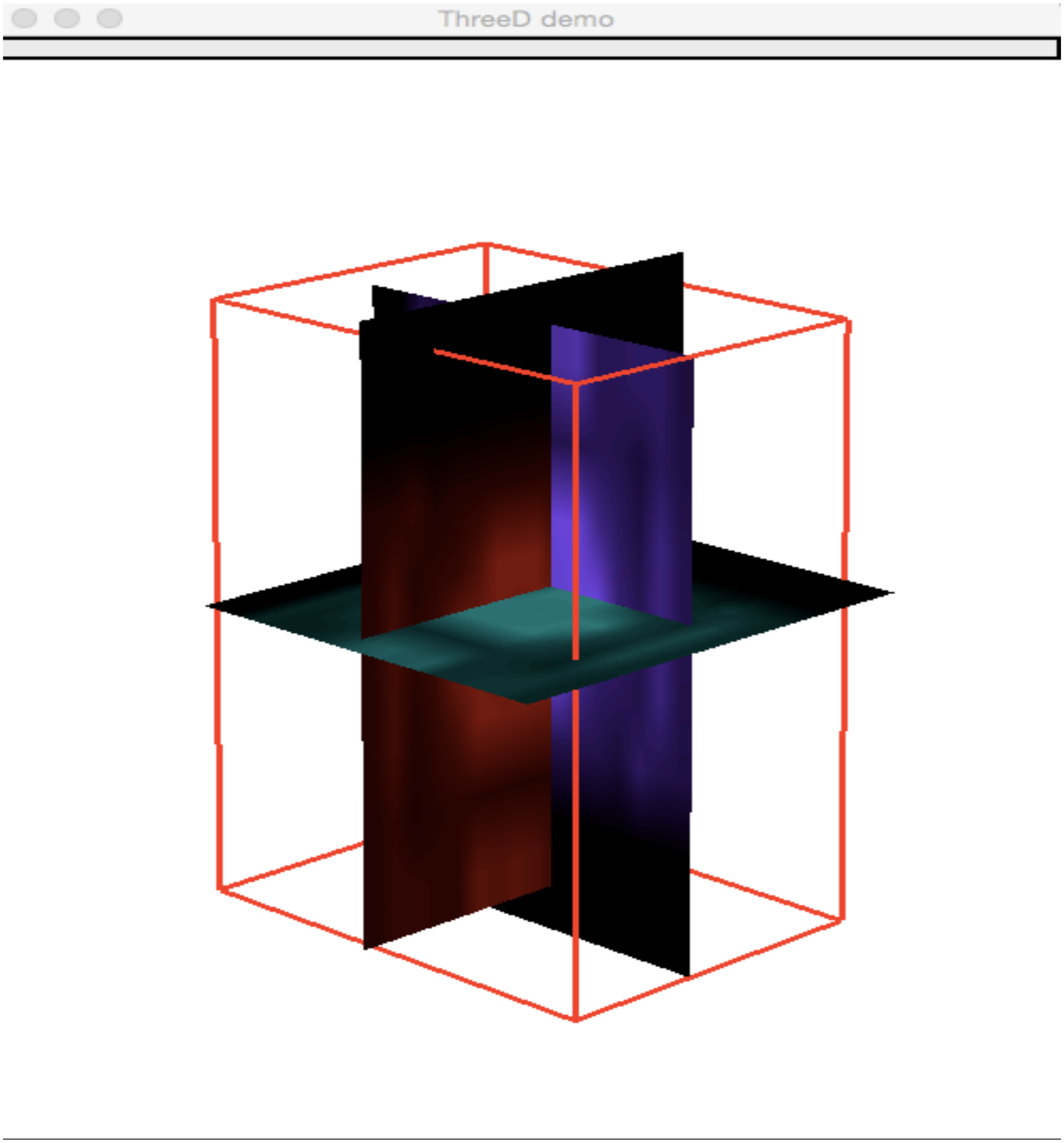
Displaying the walnut data slice with UNIFORM SCALING :



Displaying the error data slice1



Displaying the error data slice2



CHAPTER 5

CONCLUSIONS

Conclusion:

The main goal of this project were to develop an error model that gives user a clear picture of the variations among the error data slice and original data slice. We have presented flexible mechanisms so that user can view low resolution data sets with error informations. As an addition error information generated simultaneously and low resolution can also be visualized. User can interact with data source by viewing different slices of the data source, error related to the specified data source.

Future Enhancement:

The future enhancement of the project will be providing more user interactive features by adding slice number to the slider. This would make easy for the user to know which slice they working on at pre. We can also implement different ways to represent the error in the data like mapping fog in the places of error.

CHAPTER 6

REFERENCES

[1] Derived from the course website :

http://www.cs.unh.edu/~cs880/granite/DatasourceTutorial.pdf

[2]ERROR-DRIVEN ADAPTIVE RESOLUTIONS FOR LARGE SCIENTIFIC DATA SETS

- Samuel H. Vohr
- http://www.cs.unh.edu/star/papers/svohr_thesis.pdf

APPENDICES

Specific build/install instructions:

- ***Working with MREBuilder***

The multiresolution with error builder (MREBuilder) generates a hierarchy of multiresolution data with error information from a single, high resolution data source. The original data set is provided through a Granite File Descriptor Language (FDL) file or as a binary file if the --binary option is used.

The low resolution data and error are written to the same directory as the original data set. The resolution levels and error function names are appended to the original file name to make the file names for the new data sets.

For instance the following is the average absolute error for headf.bin and headf.xfdl was generated as

headf.bin.d1, headf.bin.d1.fdl

headf.bin.avgabs.d1.e3, headf.bin.avgabs.d1.e3.fdl

headf.bin.avgabs.d1.e4, headf.bin.avgabs.d1.e4.fdl

- ***Installation procedures***

To generate the low resolution data sets and errors we used the MRE Builder.This made the task of generating Multi Resolution data easier. Steps we followed to produce data sets :

- Download MREBuilder.jar from course website.
- Files that are to be stored in the directory :

dataset.xfdl, dataset.bin, MREBuilder.jar

- To avoid issues create lib folder to store granite.jar .

(granite library's class are searched in the path lib/..)

[2] User's guide

Usage

MREBuilder [OPTIONS] FILE

Options

--uniform

Use uniform decimation to build lower resolutions (default).

--wavelet

Use wavelet decomposition to build lower resolutions.

Resolutions

-n NUMBER-OF-RESOLUTIONS

The number of lower resolutions to generate. The default value is 1.

-e NUMBER-OF-ERROR-RESOLUTIONS

The number of error resolutions to generate. The default value is 1.

-o ERROR-RESOLUTION-OFFSET

The resolution offset between a low resolution data set and its first error set. The default value is 0.

Error Functions

--maxabs

Calculate the maximum absolute error for each region.

--avgabs

Calculate the average absolute error for each region.

--stddev

Calculate the standard deviation for each region.

--snr

Calculate the signal-to-noise ratio for each region.