Title: 3D Visualization Tool Creation Using Java and OpenGL

Author: Casey O'Donnell

Mentor: Dr. Lee Elson, Jet Propulsion Laboratories

Abstract:

WebWinds[1] is a freely distributed software package offering an array of different scientific visualization tools and methods of data distribution and manipulation. The addition of several 3D visualization tools adds to the existing power of this application suite. The process of porting[2] C[3] source code to Java[4] is a major component of adding these tools to the WebWinds tool suite. Observations on programming techniques, code maintenance, and program design are also addressed.

Introduction:

The primary goal of this project was the porting of several visualization tools from a previous program called LinkWinds[5] to a new scientific visualization package called WebWinds. It became necessary to port several 3D visualization tools to WebWinds, OrthoView, IsoView and VolumeView. Each involved some form of volumetric rendering. Volumetric rendering is the process directly mapping 3D data into a 3D space on the computer. Pseudo volume rendering is a similar process, but instead of mapping every data point onto the screen some sort of generalization is used. The points can instead be mapped to some sort of graphics primitive (line, triangle, quadrilateral, etc.). Isosurface creation is a process of creating 3D contour plots (Hearn and Baker 398 - 399).

Each one of these applications addressed a different method of volumetric data representation. Some of the older source code was very useful in bringing these applications quickly to WebWinds. However, other parts of the source code were written in such a fashion that working with them was very difficult, so this source code was either recreated or almost completely rewritten using more advanced methods. All three applications were completed successfully, although time will need to be invested in debugging the applications and several features need to be added.

Several of the algorithms used for rendering the volumetric data have been improved since the writing of the original LinkWinds source code. Making use of some of these newer methods would be a very useful improvement for WebWinds. Since all of these applications are almost completely tied to the speed of the processor of the computer it is being run on, more efficient algorithms would allow for greater performance all around.

---

[1] http://webwinds.jpl.nasa.gov/

[2] The process of translating software to run on a different system or the results of doing so.

[3] A very popular imperative programming language. In ANSI form it is relatively easy to move from one computer platform to another, but it is often used in combination with platform specific (Wang 1).

[4] A very popular imperative programming language. Designed to be highly object oriented and platform independent (Flanagan 3-4).

[5] http://linkwinds.jpl.nasa.gov/

Materials and Methods:

Java and OpenGL[6] were the programming languages used in the creation of these visualization tools. However, OpenGL is not available directly from within Java, so native (platform dependant) libraries were necessary. This was accomplished using the Java Native Interfaces (JNI) that allow Java programs to access routines that are machine dependent. Once the foundation for 3D visualization tools in WebWinds was laid, the process of bringing some of the older LinkWinds 3D visualization tools was possible. Because of the complexity of the WebWinds code base, I felt that bringing one of the simpler 3D tools over would assist in my initial understanding of the nature of WebWinds, but also prevent the frustration of attempting to start with one of the more complex tools.

OrthoView (Figure #1) appeared to be the simplest application to begin with. OrthoView creates a pseudo volume rendering of a 3D data set using no color information. It is primarily used to allow a user to better understand the make-up of a 3D data set. It creates the pseudo volume rendering based upon a high and low value, plotting any values that lie between those two points. It is a pseudo volume rendering, it uses a line as it's graphics primitive. Another WebWinds tool Histogram[7] allows the user to specify the two high and low values used by OrthoView. This linking of different applications in WebWinds provides for a huge degree of user control, but tends to create a great deal of ambiguity during the testing phase of application development.

IsoView (Figure #2) was the second application to undergo porting. IsoView creates an isosurface based upon a user selected contour value. An isosurface is a form of pseudo volume rendering where a 3D surface is built up based on an arbitrary "isovalue," or contour value. The color of the isosurface is then based on the "isovalue's" color in the data set's color database. IsoView uses triangles as it's graphics primitive. However, it actually makes use of triangle-strips, which are triangles that have two common vertexes. These are then positioned in strips, which create the 3D surface. IsoView uses the commonly used "marching cube" method for creating its graphics surfaces. This method "marches" through the entire data set, so it is very CPU intensive. IsoView incorporates a process called decimation to reduce the overall complexity of the data set, so that it may be rendered more easily. When the user is interacting with the 3D model, decimation is used to make it possible for the user to interact with the model in real time.

VolumeView (Figure #3) was the third and final application incorporated into WebWinds application suite. VolumeView presents the user with a true volume rendering. It plots every point in a data set onto the screen. It also allows the user to control opacity levels of certain colors. In this fashion, the user can make certain values more or less opaque allowing them to better understand the underlying structure of the 3D data set. VolumeView also makes use of user controlled decimation and decimation levels calculated in real time, allowing for real time control of the 3D model.

---

[6] A highly portable 2D and 3D graphics application programming interface (API).
[7] http://webwinds.jpl.nasa.gov/rel2/files/HistoApp.html

Both VolumeView and IsoView make use of several projection types. This allows the user to look at the data in a projection that is more useful to a user than forcing a particular viewing style.

Results:

All three of the WebWinds tools were finished, although the tools still need to undergo testing, and several features need to be added, but the basic functionality of all of the applications was completed. As WebWinds continues to evolve, some maintenance of the applications will be necessary.

Discussion:

A great deal of my time during this project was not spent trying to understand the existing framework of WebWinds as I had anticipated. Instead I found myself spending the majority of my time deciphering code written by previous programmers. Not only did I have to understand their methods, often times it was necessary to interpret what their original intentions had been.

Some sections of the newly created WebWinds source code share many of the peculiarities of the original LinkWinds source code. In many cases I've tried to comment on what I found to be the purpose of a given section of source code. In addition to commenting the source code, attempts to simplify and clarify the code were made.

Maintainable source code is written in a fashion that is not only readable to a programmer, but also easily modified to add functionality or fix bugs. The source code I dealt with on a regular basis was very difficult to deal with. This prevented my ability to examine many of the applications for possible improvements, because I often found most of my time being spent dealing with situations where changing one seemingly minor aspect of an application would inadvertently and unexpectedly affect other parts of the application.

All computer programmers and even more importantly computer scientists need to be acutely aware of the ability for other readers to understand the nature of their programs. This not only allows others to learn from what others are saying, but when improvements are being done on older source code, the time is spent on the improvements, and not on trying to understand what the original programmers intentions were.

This process of building greater and greater complexity on top of existing systems requires that those systems are not only complete and reliable, but that programmers can easily use them.

Contrary to my initial suspicions, learning to understand the way WebWinds was structured was a very simple task. This was due almost completely to the very good design and structure of the underlying components of WebWinds. I found that adding functionality for components to communicate with one another was very easy. Many of

the 2D tools within WebWinds shared many characteristics, so these were abstracted[8] in a fashion

Since many of the 3D applications being incorporated into WebWinds are actually ports of older C source code, the structure of these applications was not re-engineered, which would have provided a much greater level of abstraction. However, because the different 3D tools were not re-engineered, I found that they had a great deal of duplicate source code. Another possible source of improvement for later versions of WebWinds would be the re-examination of these applications for possible improvements.

Conclusions:

From my work this summer, my primary conclusion has been one of the importance of good code writing, and in source code documentation. I found well documented existing source code to be far more useful than any sort of general overall documentation. Furthermore, the importance of good coding skills and well designed software, and it's direct link to the ease of code maintenance and the debugging of software.

Acknowledgments:

Dr. Lee Elson, Jet Propulsion labs – Mentor and technical writing assistance.
Dave Hanks, Jet Propulsion labs – Technical assistance and moral support.
Martin Orton, Jet Propulsion labs – Technical assistance, moral support and WebWinds designer.
Dr. Hari Nair, Jet Propulsion labs – Technical assistance and moral support.
Dr. Ken Kopecky, Drake University – Monetary assistance for housing and transportation.

[8] The act of separating the essential qualities of an idea or object from the details of how it works or is composed (Headington and Riley 1).

Bibliography:

Flanagan, David. Java In a Nutshell, 2nd Edition. California: O'Reilly & Associates, 1997.

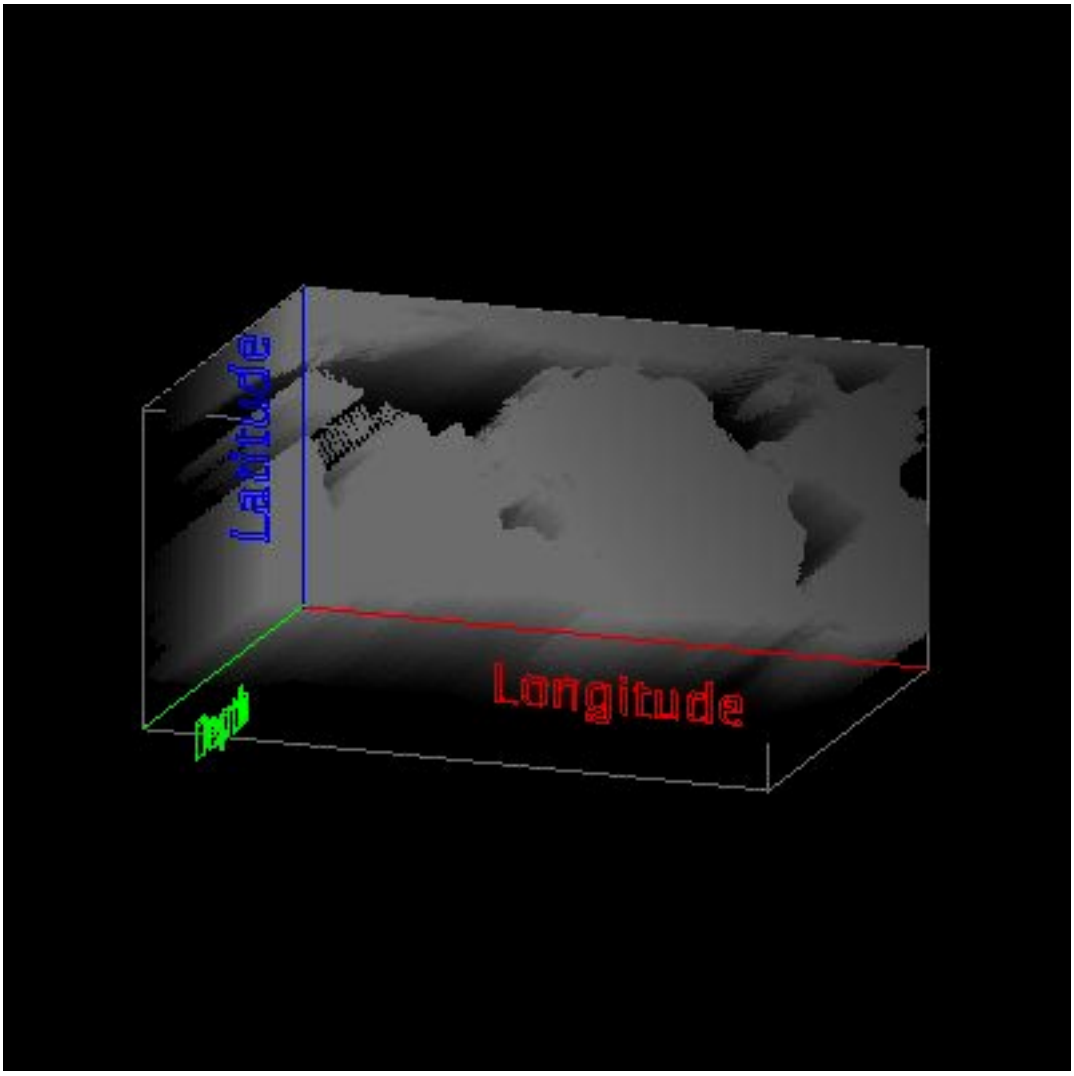
Headington, Mark and Riley, David. Data Abstraction and Structures Using C++. Massachusetts: D.C. Heath and Company, 1994.

Hearn, Donald and Baker, Pauline. Computer Graphics C Version. New Jersey: Prentice Hall, Inc, 1997.

OpenGL Architecture Review Board. OpenGL Reference Manual, 2nd Edition. Massachusetts: Addison-Wesley, 1997.

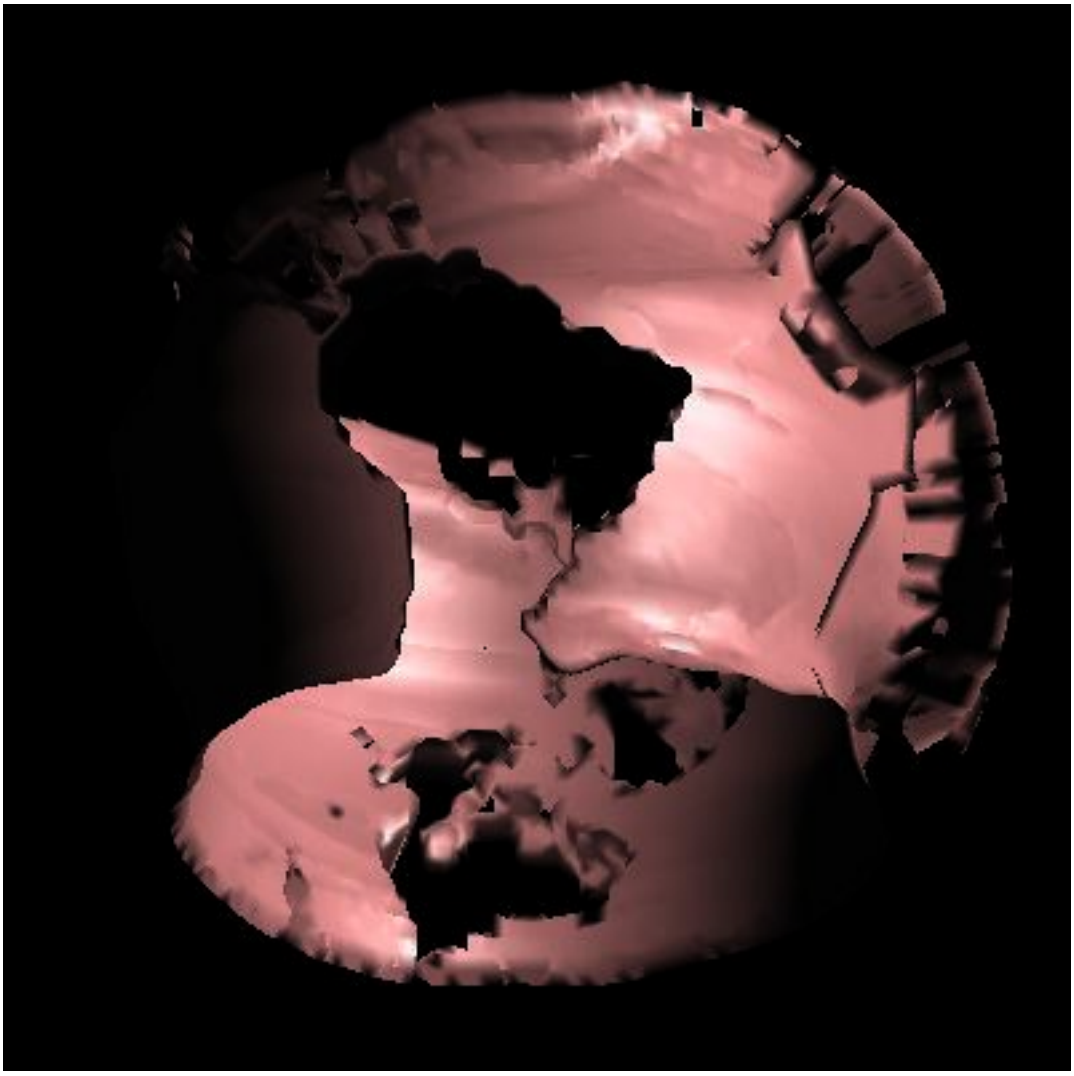
Wang, Paul. ANSI C on UNIX. Boston: PWS Publishing Company, 1993

Figure #1:



OrthoView rendering ocean temperature data.

Figure #2:



IsoView rendering the same data set, with a “isoval” in the middle of the temperature range.

Figure #3:

VolumeView rendering the same ocean temperature data set.