SAND97-1689C
SAND--97-1689C

Visualization Tools for Comprehensive Test Ban Treaty Research

CONF-970967--8

Tony L. Edwards, J. Mark Harris, Randall W. Simons, Christopher J. Young
Sandia National Laboratories

Sponsored by U.S. Department of Energy
Comprehensive Test Ban Treaty Research and Development Program

ABSTRACT

This paper focuses on tools used in Data Visualization efforts at Sandia National Laboratories under the Department of Energy CTBT R&D program. These tools provide interactive techniques for the examination and interpretation of scientific data, and can be used for many types of CTBT research and development projects. We will discuss the benefits and drawbacks of using the tools to display and analyze CTBT scientific data. While the tools may be used for everyday applications, our discussion will focus on the use of these tools for visualization of data used in research and verification of new theories. Our examples focus on uses with seismic data, but the tools may also be used for other types of data sets.

Keywords: data visualization, scientific visualization, Comprehensive Test Ban Treaty, CTBT.

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under Contract DE-AC04-94AL85000

1. Introduction

Scientific visualization is one of the many reasons that computer graphics is used heavily today. Scientists are not artists, but they do need to view their data graphically, and their data sets exist on computer systems, so it is natural to display this data on the computer.

Visualization often helps people extract information from data faster and more efficiently. The large volumes of event data and the wealth of contextual information to support decision making make data visualization a portion of CTBT research.

Three-dimensional visualization is currently a popular method for displaying scientific data. This method gives a user an unlimited range of views for their data. They may look at the data from any angle, zoom in or out, and slice volumes of data into single planes. Often, new information about the data is gleaned from this highly flexible method of examining the data.

The following sections will provide an overview of applications available commercially and via the Internet that were used by the Data Visualization Team at Sandia National Laboratories. These tools provide the capabilities to display data in 2-D and 3-D. Some of the tools are best suited to rapid prototyping, while others are better for high speed visualization. The tools to be discussed are: AVS Express, The Visualization Toolkit, OpenGL, and MATLAB.

2. Research Environment

Our research environment consists of Sun Ultra1 and Ultra2 workstations with Creator3D graphics. The Ultra1 workstations contain a 167mhz processor and the Ultra2 workstations contain two 200mhz processors. The Creator3D graphics cards contain hardware enhancements for accelerated 24-bit, double- and Z-buffered 3-D rendering. Many of our visualization efforts use GSETT II data (Group of Scientific Experts 1992) and GSETT III reviewed event bulletins from the International Data Center.

3. Advanced Visual Systems (AVS) Express

AVS Express is a data visualization application that is available from Advanced Visual Systems (http://www.avs.com). This product allows for rapid visualization of data via a large set of predefined modules. These modules provide image processing, picking, graphing, hardcopy output, GIS, Database, GUI creation, math routines, etc. Most of what the user needs is here and if it is not available you can create your own through C, C++ or FORTRAN. AVS Express will run on Unix and Windows NT and 95 platforms. AVS Express takes advantage of hardware acceleration if available or uses a software renderer otherwise.

AVS Express uses a visual network editor (see Figure 1) to develop an application. All of the user's modules are networked together via input and output data ports. The data flows from files and databases through the different modules and finally into a two or three dimensional viewer. Different filters (slice, orthoplot, etc.) may be applied to the data before it is displayed.

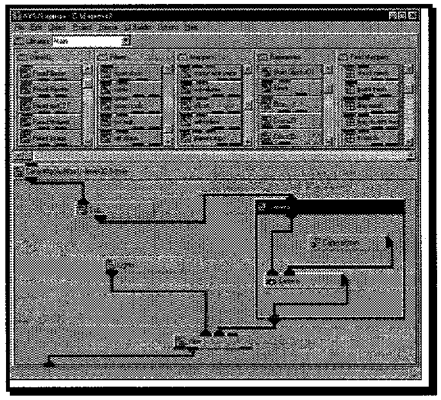

Figure 1. AVS Express Network Editor

We used the GIS and database modules to visualize data from a CSS 3.0 database from GSETT II data(Anderson, et. al. 1990) (See Figure 2). We were able to create spherical and 2-D mapping projections of GIS data sets supplied and use the database module to access an Oracle database (Simons, et. al. 1996). The GIS modules provide access to CIA World Data Bank II and Digital Elevation Map (DEM) data. You may also read data from other data files or via remote processes.

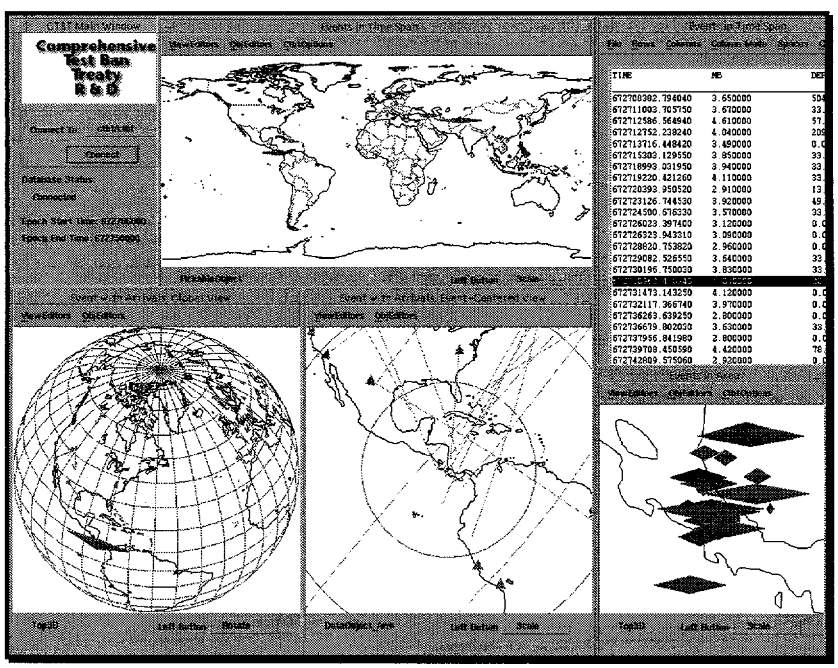

Figure 2: Data Visualization with AVS Express

AVS Express is a relatively expensive product, but it does offer a rich suite of tools to visualize any data. This product did run slow on our machines due to the amount of data and the fact that AVS Express does not support OpenGL on the Sun platforms at the time of this writing. Much of the poor performance could be due to the overwhelming module support the product contains. If the data set is small and specialized, this product is overkill. If you view many types of data and need fast prototyping or visualization, this product could be a great tool for your research.

4. The Visualization Toolkit

The Visualization Toolkit (VTK) is a 3D graphics package that grew from research at the General Electric Corporate R&D Center. It is still supported by the staff at General Electric. The product is freely available via the Internet (http://www.cs.rpi.edu/~martink/) or may be purchased with the official book (*The Visualization Toolkit: An Object Oriented Approach to 3D Graphics,* Will Schroeder, Ken Martin, Bill Lorensen, Prentice Hall). The software in the book is several versions old now, but the book is still great hardcopy reference. The software has support for many graphics libraries and platform support for Unix and Windows NT and 95 based PCs.

VTK was written in C++ (you will need a C++ compiler to compile the package) in order to take advantage of the object-oriented capabilities of the language. You may write code in C++ or use the Tcl Interpreter (information on Tcl can be found via the Internet at http://sunscript.sun.com). Using the Tcl Interpreter can provide for rapid prototyping, but we found that creating an application was just as easy using C++.

VTK, like AVS Express, provides a large suite of tools for 3-D data visualization. VTK provides all of the necessary routines to create and manage cameras, geometry support, analysis routines, hard copy support, interaction, etc. There is no support for GIS mapping or database connectivity. This capability must be created by the developer of the application.

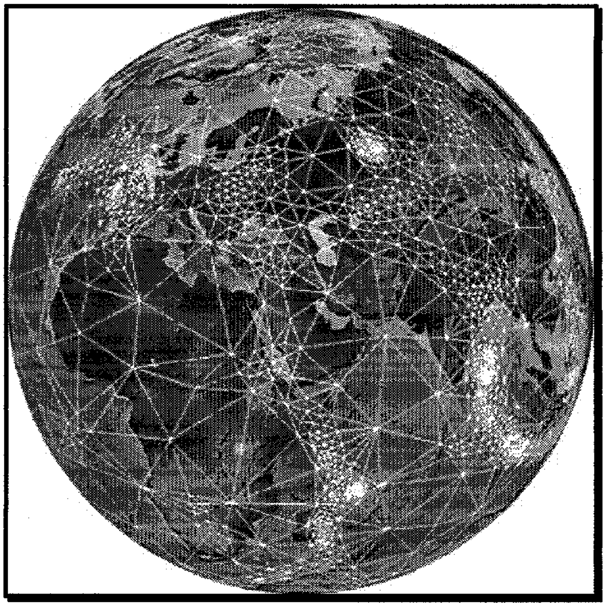

Figure 3: Visualization using The Visualization Toolkit

To investigate VTK, we created an application to view Knowledge Base data on a 3-D globe (See Figure 3). We had to create the necessary code to read coastline data and the Knowledge Base data, but we used VTK's interaction routines, OpenGL support, and earth object to create the heart of our application. The amount of code that we had to write to support our application was minimal.

VTK also supports viewing of data with stereoscopic glasses. We used this to visualize many different data sets and found stereoscopic viewing to be a valuable asset to 3-D data visualization. We used the CrystalEyes product from StereoGraphics (http://www.stereographics.com) to view our data in stereo. Stereoscopic viewing is accomplished by showing the left image on the display while the right lens of CrystalEyes is closed, then showing the right image on the display while the left lens is closed. Because this happens a minimum of 120 times per second, you perceive left and right views simultaneously. The result is amazingly real depth perception.

The performance proved to be quite slow, but this is due to VTK's limited capability of taking advantage of graphics hardware. VTK has support for numerous windowing and graphics packages. This has great functionality, but it does limit performance.

5. OpenGL

OpenGL is an application programming interface (API) that is available on almost all platforms used today. OpenGL is supported by strong industry standards and conformance. Online information regarding the OpenGL standard can be found at http://www.sgi.com/Technology/OpenGL/.

We implemented several data visualization applications using OpenGL and C++. Figure 4 shows an OpenGL application that displays output from the Waveform Correlation Event Detection System (Young, et. al. 1996). This application was created to animate the data over a set period of time. Our initial data set looked at 1 hour of data stepped at 10 second increments. The performance is excellent because we wrote software that was optimized for the graphics hardware.

Programming data visualization this way does require the developer to create all of the graphics code from scratch, but once this is done, a developer should be able to support other data sets fairly easily. We created a graphics user interface using Motif and created the necessary basic OpenGL elements (lighting, camera positions, rotation, zoom, pan, etc.). We also had to handle the GIS mapping (coastlines) and file input and output. Data input into OpenGL can be done fairly easy by inputting your data set as points, but some thought must be put into what you want to visualize. Once we created the basic elements we are able to display data in a variety of ways by changing the data set we wished to view.

By programming a visualization system by hand, a developer is able to create exactly the elements required to display their data and consequently performance is optimized. In contrast, much overhead goes into commercial and public domain packages to support the mythical average user, and performance can be affected. Clearly there is a big trade off with the time it takes to develop an OpenGL graphics package vs. performance. There are many toolkits available to help with design, but it is still time consuming.

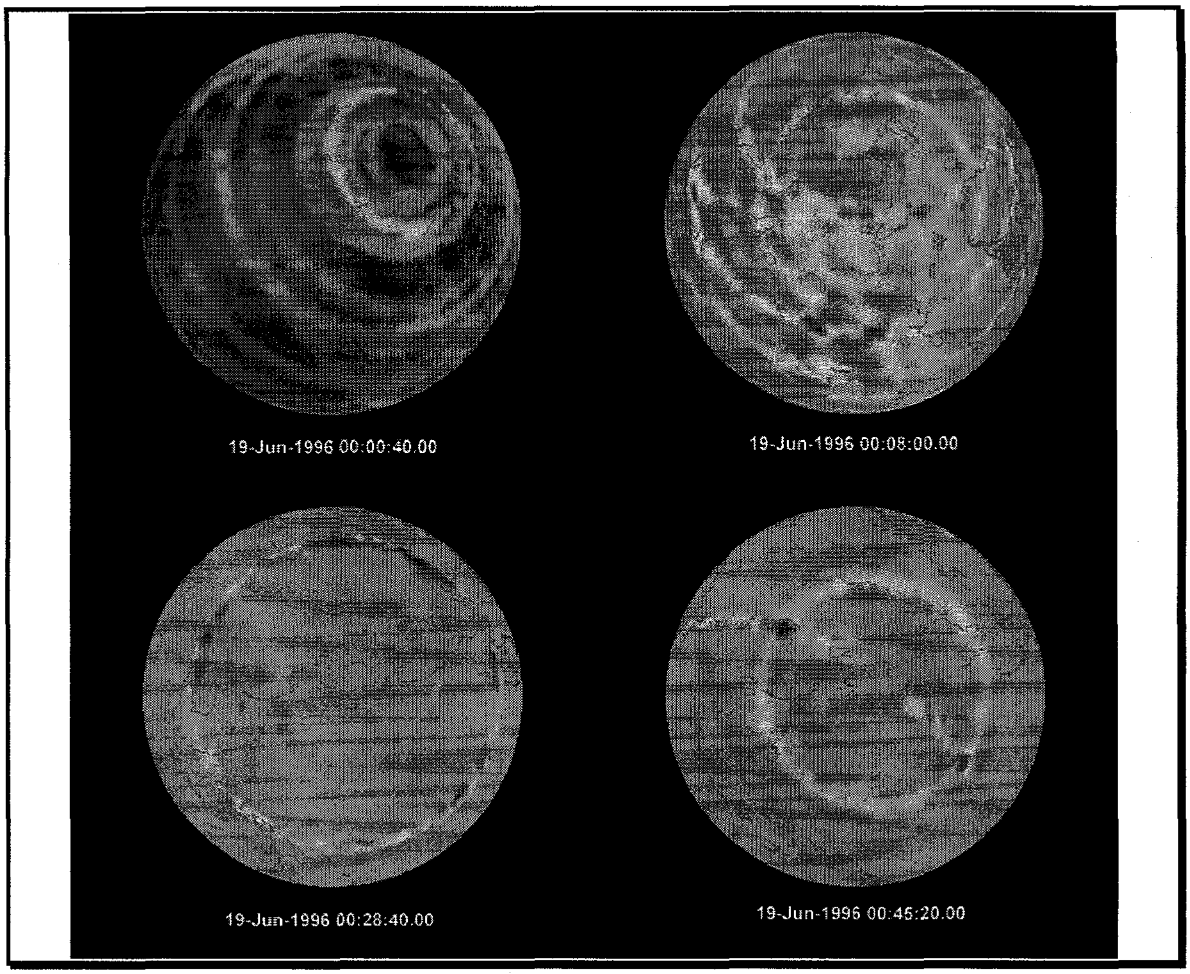

Figure 4: Data Visualization with OpenGL

6. MATLAB

MATLAB is a commercial product from The MathWorks, Inc. (http://www.mathworks.com). MATLAB is an application for technical computing that combines numeric computation, advanced graphics and visualization, and a high-level programming language. We have found MATLAB to be an excellent environment for analysis, algorithm prototyping, and application development. Many data analysis procedures, such as spectral estimation, filtering, and interpolation, are available at the command line to operate on data in matrix variables. Other processing capabilities are available from The MathWorks in the form of toolboxes. These include things such as wavelets, neural networks, and system identification.

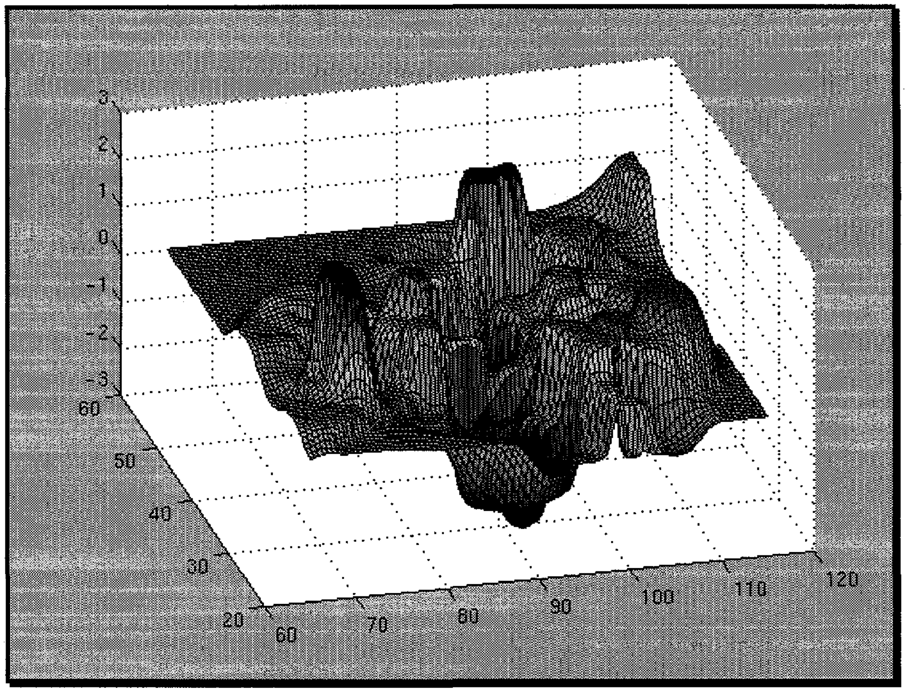

Figure 5: Spatial Data Visualization with MATLAB

We routinely use MATLAB to process and visualize spatial data, such as the interpolation of travel-time error corrections as shown in Figure 5. Using the high-level plotting commands in MATLAB, the interpolated surface may quickly be viewed and manipulated to aid in the development of the algorithms.

To support seismic waveform processing algorithm development, we developed MatSeis, a seismic analysis GUI programmed in MATLAB (http://www.ctbt.rnd.doe.gov/ctbt/data/matseis/matseis.html). MatSeis (Harris, et. al. 1996) makes it easier to access seismic data in MATLAB from our CSS 3.0 database. The standard MATLAB environment is available from the command window. Data displayed in MatSeis may be accessed on the command line for manual processing and the results plotted using MATLAB or entered back into MatSeis. The basic MatSeis display (see Figure 6) is an interactive data profile (time vs. epicentral distance) integrating event origin, waveform, travel-time, and arrival data, all of which are selectable objects. Graphical plot controls, data manipulation, and signal processing functions are provided. Current capabilities include filtering, beaming, spectral analysis (PSD and spectrogram), FK analysis, vespagram creation, database editing, and mapping. We continue to develop modules as needed by our group.

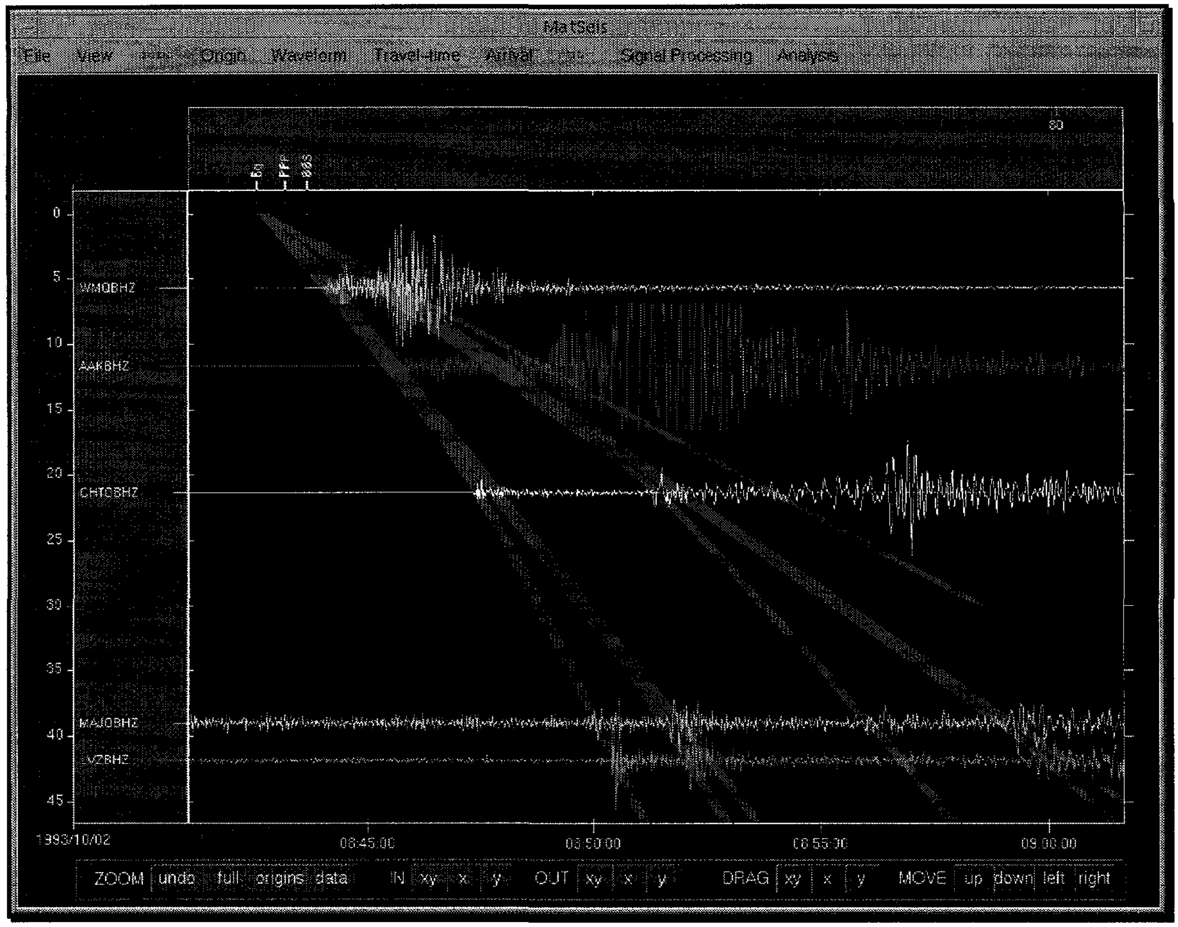

Figure 6: MatSeis Application using MATLAB

7. Conclusions

With the increased amount of data that must be analyzed to monitor the CTBT, the need for visualizing the data in more than two dimensions is becoming apparent. Interactive displays for this type of data are essential. There are many programming tools and applications available in the market today for scientific visualization. This paper discussed several tools that Sandia National Laboratories researchers used to visualize data for CTBT research. We feel that these tools could fulfill the needs of many CTBT researchers regardless of scientific field.

To aid researches in deciding which of the tools are most appropriate for them, we have categorized them according to the following criteria: cost, creation method, support, performance, and platforms available (see Table 1). The creation methods for application development were by hand coding or visual interface builder. A visual interface builder allows you to build an application by connecting individual modules together to make an application. The performance characteristics in Table 1 were based on our observations. When building any type of data visualization application, performance is driven by the amount of data to be viewed and how the application was programmed. While using OpenGL, we were able to program the software in several ways. When we programmed the software within the specific parameters of the hardware vendor, we were able to get the highest performance.

	Cost	Creation Method	Support	Performance	Platforms
AVS/Express	$15K+	Visual interface and/or code	Vendor	Slow to Medium	Unix, PC
Visualization Toolkit	$0	code	Developers and Newsgroups	Slow to Medium	Unix, PC
OpenGL	$0 to $1000+	code	Vendor and Newsgroups	Medium to High	Unix, PC, Mac
MATLAB	$1000 to $4000	code	Vendor	Medium	Unix, PC, Mac

Table 1: Visualization Tool Characteristics

When starting to develop an application for data viewing, the developer must be aware of the potential end user. Who is the end user, how often will this application be used, and what systems will be used are important questions to ask. If a scientist needs to view an application only once, than a commercial off the shelf (COTS) package may be the best application for development. Conversely, if the application needs to handle real-time data and be used routinely, a COTS application may not be a good choice for development.

8. Additional Topics

There are other important areas of interest that a developer or scientist should consider while implementing a visualization application. We will discuss a few of these topics below.

For some problems, the user may wish to explore more exotic technologies. For example, large data sets could be explored with Virtual Reality technologies. Visualizing in this method gives users a better sense of depth compared to traditional display technologies. With the use of the Virtual Reality Modeling Language (VRML), it is even possible to share data viewing across the World Wide Web with other scientists.

GIS systems can also provide valuable information. For example, displaying event data on actual topological maps could prove to be a great asset. Of course a GIS is more than just maps and land formations; it also has the capability to store critical attributes about the area, such as mines, industrial areas, geological formations, etc.

We are currently developing software to view correction data used for the CTBT knowledge base. The viewing of the algorithms and the data is very important in creating an accurate application. One particular area of visualization need is in knowledge base grid editing. The grid must be checked for adequate coverage, proper interpolation, and referenced to contextual information such as geologic and political boundaries.

Another very important issue in visualization is Human Computer Interaction (HCI). The most important item you will encounter when developing your application is color (Tufte 1990). The use of color can display the correct information and can also convey the wrong information. The user must be aware of how color is used in your application.

9. Conclusions and Recommendations

We have discussed several tools used in scientific visualization at Sandia National Laboratories. While these tools ranged in cost from cheap to expensive, no tool stands out as a clear answer to all of our visualization needs. A scientist must understand his audience, data needs, visualization experience, and application's main goal to decide which tools is the best for them. There are many ideas and approaches to scientific visualization. It is important to try many different methods of visualization to make sure a user understands the data correctly. One must take into account textures, lighting, modeling approaches, proper data set creation, colors, hardware, software, etc.

Future visualization efforts at Sandia National Laboratories will focus on helping researchers and users extract information from increasingly large, complex spatially oriented data sets, such as those found in the CTBT Knowledge Base.

10. References

Anderson, J., Farrell, W. E., Garcia, K., Given, J., and Swanger, H. (1990), CSS Version 3 Database: Schema Reference Manual, Science Applications International Corporation, Arlington, VA.

Simons, R. W., Young, C. J., Edwards, T. L. (1996), Data Visualization for Comprehensive Test Ban Treaty Monitoring, Proceedings of the 18th Seismic Research Symposium on Monitoring a CTBT.

Harris, M., and Young, C. (1996), MatSeis: A Seismic GUI and Tool-box for MATLAB, Seismol. Res. Lett., 67(2), 267-269.

Young, C., Marris, M., Beiriger, J., Moore, S., Trujillo, J., Withers, M., and Aster, R. (1996), The Waveform Correlation Event Detection System Project, Phase 1: Issues in Prototype Development and Testing, Sandia National Laboratories, SAND Report SAND96-1916.

Tufte, E. R. (1990), *Envisioning Information*, Graphics Press, Cheshire, Connecticut.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.