



## Short Note

# Displaying seismic data with VTK

*Elkin R. Arroyo and Robert G. Clapp*<sup>1</sup>

### INTRODUCTION

In exploration geophysics most simple targets have now been found and exploited. New prospects are structurally complex requiring 3-D acquisition and imaging. The need to effectively build a velocity model for 3-D imaging and to effectively view the resulting image has led to the need to build more sophisticated viewing software.

Over the years SEP has made several attempts to address this need. Ottolini (1983) built an effective 3-D viewing program, later expanded upon in Clapp (2001). Later, several attempts were made to use the proprietary Advance Visual System (AVS)<sup>2</sup> package to build an integrated viewer (Biondi and van Trier, 1993; Clapp et al., 1994; Mora et al., 1995, 1996). These attempts were frustrated by the proprietary nature of the product, the hardware requirements to effectively run it, and the programming expertise needed to use a package originally designed for another field (chemistry).

In the last few years the need for an effective visualization software at SEP has increased while many of the limitations that hampered previous attempts have been overcome. With the increased power and decreased cost of linux workstations with 3-D accelerated graphics, desktop viewing is more feasible. In addition, numerous 3-D visualization packages have become publically available. One such package, Visualization Toolkit(VTK)<sup>3</sup> holds particular promise. It has a diverse user base, C++, Python, Tcl, and Java interfaces, and an intuitive design. In this paper we build a cube and surface viewer using the VTK library. We begin by discussing some of the capabilities of VTK and general implementation design. We then describe how to use it to view seismic data (`VtkCubePlot`) and Gocad surfaces (`VtkGocadPlot`). We conclude with some future directions for the work to make it an even more effective research tool.

---

<sup>1</sup>email: earroyo@numerica.com.co,bob@sep.stanford.edu

<sup>2</sup><http://www.avis.com>

<sup>3</sup><http://www.kitware.com/vtk.html>

## VISUALIZATION AND VTK

Visualization is the transformation of data or information into pictures. Visualization uses graphics to give insight into certain abstract data and symbols. Visualization is a necessary tool to make sense of the flood of information in today's world of computers. Seismic data acquisition system nowadays are capable of acquiring hundreds of gigabytes in a few days and seismic processing itself produces a lot of information that, without visualization, would stay unseen on computers disks and tapes.

There is another advantage of visualization, it utilizes the human vision system. With the introduction of these new tools, it will be possible to use our highly developed visual sense, and therefore to better understand and analyze the seismic information.

### Common visualization tools

There are many commercial and free visualization software packages. It is difficult to categorize visualization tools because most have a variety of functions covering many different applications, and many have overlapping functions. Common visual programming systems include VTK, AVS/Express, Iris Explorer<sup>4</sup>, IBM Data Explorer<sup>5</sup> and Khoros<sup>6</sup>, all of which are based on low-level graphics libraries like OpenGL<sup>7</sup>, Direct3D<sup>8</sup>, Phigs<sup>9</sup> and GKS-3D<sup>10</sup>. The most widely accepted and used is of these OpenGL.

### VTK interaction with SEP Data Structures

In the following section we explain basic concepts of the VTK data model and explain which one we choose to represent the seismic information and which one to represent the GOCAD surfaces. For a more detailed description see ?.

The VTK data model consists of two pieces: an organizing structure (the shape or geometry) and associated information to each element in the structure known as the attribute data. Examples of each are shown in the table below:

Organizing structure	Data attribute
Points, triangles, quadrilaterals, tetrahedral,	Scalars, vectors, normals, texture coordinates

<sup>4</sup>[www.nag.co.uk/Welcome\\_IEC.html](http://www.nag.co.uk/Welcome_IEC.html)

<sup>5</sup>[www.ibm.com/dx](http://www.ibm.com/dx)

<sup>6</sup>[www.khoral.com](http://www.khoral.com)

<sup>7</sup>[www.opengl.org](http://www.opengl.org)

<sup>8</sup>[www.microsoft.com](http://www.microsoft.com)

<sup>9</sup>[http://www.gsi.de/computing/unix/primer/graphical\\_tools/subsection3.1.1.html](http://www.gsi.de/computing/unix/primer/graphical_tools/subsection3.1.1.html)

<sup>10</sup>[web.ansi.org](http://web.ansi.org)

Regular data has an inherent relationship between data points. For example, if we sample an evenly spaced set of points, we do not need to store all point coordinates, just the beginning (o), the spacing between points (d) and the total number of points (n). This is the case of 2D/3D seismic information and also the way SEP structures a data cube.

For irregular data there is not a defined pattern between points or cells and therefore we can represent arbitrary structures. This is the case of the GOCAD surfaces. A big advantage of irregular data is that we can represent information more densely where it changes quickly and less densely where the changes are more subtle.

Attribute data is associated with each point in the structure and consists of scalars (which are single value functions like trace amplitudes), vectors, normals, texture coordinates, tensors, etc.

For implementing our visualization tools we choose to use the class `vtkStructuredPoints` to represent the seismic information and the class `vtkUnstructuredGrid` to represent the GOCAD surfaces. Currently `VtkCubeplot` only supports a single value attribute data for the seismic information which means that we use the VTK class `vtkScalar` to express the trace amplitudes or velocities. The SEP application `VtkGocadplot` currently supports scalar representations and 3D vectors.

### Pipeline structure of VTK

SEPlib and VTK share the ability to send information from one program to others to process the information and to obtain the final best image. This philosophy helps SEPlib and VTK users to easily get acquired with the software. For developing the SEP programs `VtkCubeplot` and `VtkGocadplot`, first we read the information using the set of SEP functions to read SEP data cubes `sreed`. This information is then reordered in the VTK way using the structured storage for the `VtkCubeplot` explained earlier and the unstructured storage format for the `VtkGocadplot`. After that, we implement a set of filters or functions that helps to construct the cutting planes, using the class `vtkCutter`. This takes the seismic information and the defined planes, cuts along the desired plane and interpolates the seismic amplitudes along the cutting surface. Input planes by default are located at center of the SEP data cube. A similar approach is used for the contour values in the `VtkGocadplot` program. We use the class `vtkContourFilter` which takes as input the GOCAD surface, and the user defines the number of contour values and the color map to be used.

## VIEWING SEISMIC DATA

The display of 3-D data is a challenging problem. SEP has developed `Cubeplot` and `Ricksep` in order to facilitate the viewing of 3-D cubes. Both of these options are limited to viewing orthogonal slices of a cube with little concept of the 3-D nature of the data. Often viewing the data in a true 3-D environment can offer additional insight. `VtkCubeplot` offers this ability. As Figure 1 demonstrates the `VtkCubeplot` program can display arbitrary planes of a 3-D cube.

It is then possible to rotate around and through the 3-D environment.

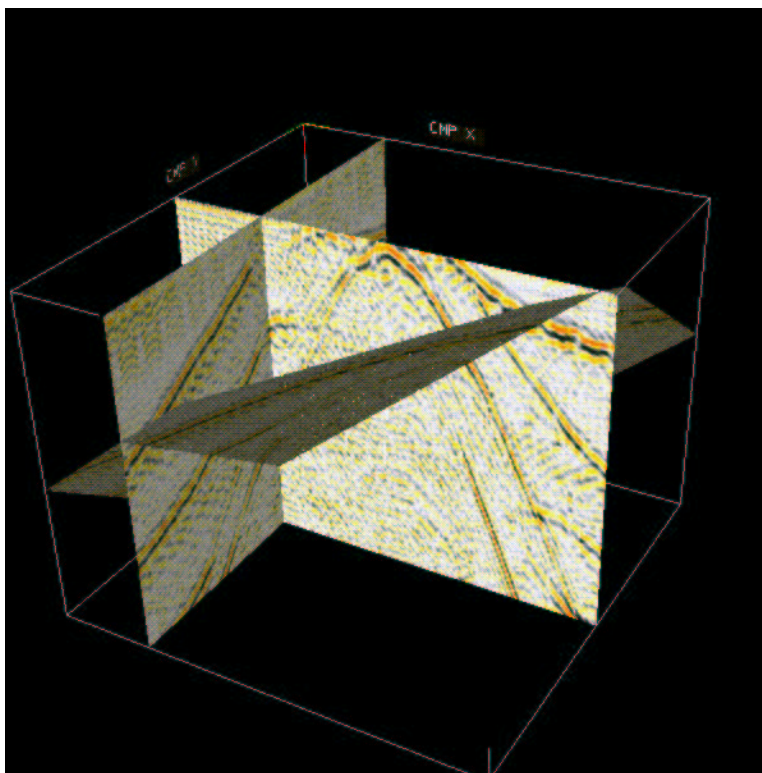


Figure 1: Three slices from the SEG salt model displayed within the VTK environment.  
`bob1-surf` [CR]

### Hard copy and reproducibility

An essential part of a researcher's job is to be able to communicate their work to others. This communication can take the form of a conversation, a presentation, and/or a paper. The last require the ability to output our 3-D environment to another media. The VTK library allows several different output options. The display can be written out in postscript, tiff, cgm, bmp, and vrml. Upon a successful exit, the final view is written to the file set by the `o` command line parameter. The type of output is defined by the `of` parameter.

Another essential component of research is the ability to reproduce results. This extends from the initial raw data to the final outputted image. SEP has always stressed the importance of reproducibility (Schwab et al., 1996). We have incorporated this idea into `vtkCubeplot`. `vtkCubeplot` can output its current display parameters to `ScenarioOut`. It can take as input a set of display parameters (`ScenarioIn`) and operate in batch mode to reproduce a figure.

### Additional command line options

The user can press key ‘3’ and get a stereoscopic view through inexpensive polarized glasses. It can easily switch between normal viewing and stereoscopic view. A 3D image is like an ordinary picture, but a stereo image gives a stronger sense of depth. This is generated by providing two slightly different images of the same object to each eye separately. Other options include:

**co** Colormap

**bg** Background color

**pcn1,pcn2,pcn3** Location of the three cutting planes

**env1,env2,env3** The orientation of the cutting plane.

**op** Opacity

**sc1,sc2,sc3** Value to scale (x,y, and z) of the surface

## VIEWING SURFACES

In addition to being able to slice through regular cubes in a 3-D environment a 3-D viewer offers the ability to display surfaces. The surface can be either a 2-D regular mesh or an irregular mesh with connectivity information. The regular mesh option might be used to display an impulse response, where the z axis is used for the amplitude at various locations. To some extent this would duplicate the functionality of the SEPlib program `Thplot` but with surface rather than line displays and interactivity.

The VTK library also easily lends itself to the display of irregular functions, such as surfaces. Both regular and irregular functions can be displayed using the program `VtkGocadplot`. The program can read SEP regular cubes or irregular surfaces described in terms of the GO-CAD<sup>11</sup> ascii standard. Figure 2 shows the salt body from SEG salt model displayed in VTK.

### Addition command line parameters

In addition to the `VtkCubeplot` commands, the following controls are available within `VtkGocadplot`. Further, the user can provide on the command line:

**vc** Whether (1) or not (0) to view the surface

**vn2** Whether (1) or not (0) to view the contour

**ncv** Number of contours to display

---

<sup>11</sup><http://www.gocad.com/>

**cc** Whether (1) or not (0) to display color line with its own color

**smooth** Whether (1) or not (0) to smooth the surface

**nis** Number of laplacian smoothing to perform

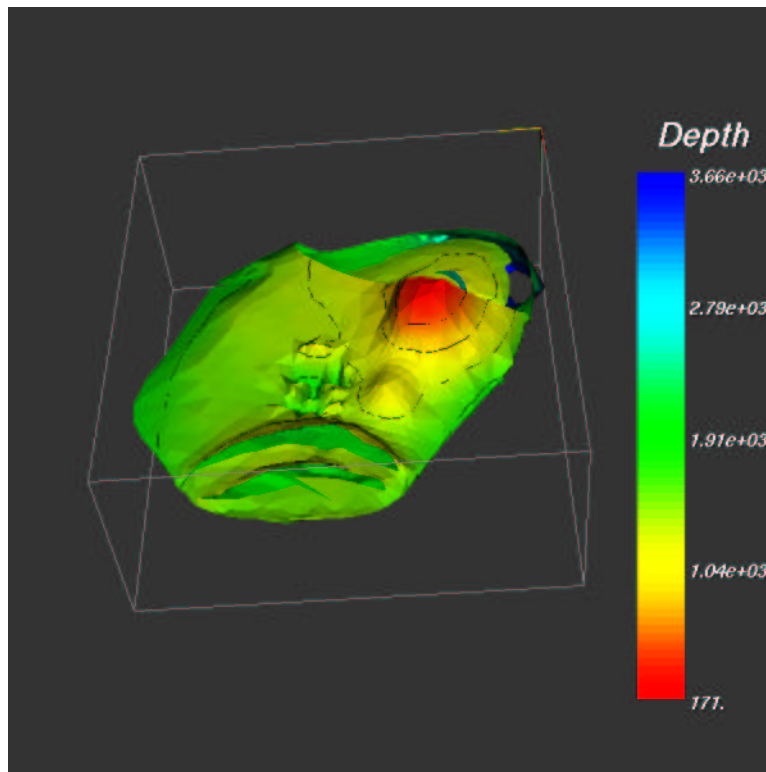


Figure 2: VTK displaying the GOCAD salt surface of the SEG salt model. `bob1-gocad` [CR]

## CONCLUSIONS AND FUTURE WORK

We built both a cube viewer and a surface viewer using the VTK library. To make these viewers more effective research tools, several additions need to be made.

- Geophysical data sets are generally very large. Due to this, SEP has a long background in developing parallel algorithms. One of the next steps in the development of the `VtkCubePlot` and `VtkGocadPlot` therefore will be to take advantage of the parallel capabilities of VTK with MPI.
- The ability to interactively slice through the cube rather than requiring slice selection at initiation.
- The ability to pick and store locations within the space.

- The ability to have an interactive list of multiple objects (planes, triangulated surfaces, lines) which can be displayed and/or not displayed interactively.
- The ability to modify objects such as triangulated surfaces.
- The ability to do texture mapping (e.g. velocity surface model with seismic amplitudes overlaid).
- The ability to annotate in the 3-D environment.
- Arbitrary number of cutting planes, and orientation for each plane, currently only 3 cutting planes are supported and only the time plane has arbitrary orientation capabilities.
- Texture mapping of the seismic data over the GOCAD surfaces.

## REFERENCES

- Biondi, B., and van Trier, J., 1993, Visualization of multi-dimensional seismic datasets with CM-AVS: SEP-79, 1–12.
- Clapp, R. G., Biondi, B., and Karrenbach, M., 1994, AVS as a 3-D seismic data visualizing platform: SEP-82, 97–106.
- Clapp, R., 2001, Ricksep: Interactive display of multi-dimensional data: SEP-110, 163–172.
- Mora, C. B., Clapp, R. G., and Biondi, B., 1995, Velocity model building in avs: SEP-89, 133–144.
- Mora, C. B., Clapp, R. G., and Biondi, B., 1996, Visualization of irregularly sampled seismic data with AVS: SEP-93, 75–86.
- Ottolini, R., 1983, Movie cubes: SEP-35, 235–240.
- Schwab, M., Karrenbach, M., and Claerbout, J., 1996, Making scientific computations reproducible: SEP-92, 327–342.



