Stanford Exploration Project, Report 89, November 12, 1997, pages 135–146

Stanford Exploration Project, Report 89, November 12, 1997, pages 135–146

Velocity model building in AVS

Carmen B. Mora, Robert G. Clapp, and Biondo Biondi[1]

keywords: *modeling, three-dimensional, velocity*

ABSTRACT

For the last three years SEP has been developing tools within Advanced Visual System (AVS), to better handle 3-D seismic data. Our newest module attempts to include the entire velocity model building process within our AVS environment. It combines some of the best elements of GOCAD with the higher level of flexibility offered by AVS. An interactive model building process was tested by combining the new module, along with other AVS modules developed at SEP, to build a 3-D velocity model from a land survey dataset provided by INTEVEP S.A.

INTRODUCTION

For the last several years SEP has been developing the capability to handle 3-D datasets. On the software end this has meant the development of a new processing environment (SEP90) and new visualization tools (Biondi and van Trier, 1993). Building velocity models for 3-D datasets have also required a whole new set of tools. The industry standard for model building, GOCAD, has attempted to fill this need. GOCAD has many desirable features [such as Discrete Smooth Interpolator (DSI) smoothing and a layered approach] but also several drawbacks [problems handling (3+)-D datasets and including user supplied code]. As a result, we have continued (Clapp et al., 1994) in our efforts to build an interface between AVS, which specializes in handling multi-dimensional datasets and easy user-inclusion of code, with GOCAD. We have extended the initial efforts described in (Clapp et al., 1994) to create one super module `Project Manager` that offers the ability to display GOCAD surfaces, to create surfaces [in the manner similar to (Clapp and Biondi, 1994)], and to grid simple models (Berlioux, 1993). All of this is provided within an interface that allows many of the useful features of GOCAD.

[1]**email:** cmora@intevep.pdv.com, bob@sep.stanford.edu, biondo@sep.stanford.edu

OVERVIEW

GOCAD is an object oriented software package to create and handle 3-D geometrical shapes. It allows the representation of geological models by the partitioning of the 3-D space into domains. Each domain is characterized by bounding surfaces and a set of properties (such as velocity) describing the domain. GOCAD is widely used in building geologic models and is an excellent tool when it comes to altering surfaces and making them more geologically feasible. However, its ability to handle multi-dimensional fields or to be customized through inclusion of user supplied code is limited.

These weaknesses led SEP to find a platform that would allow this flexibility, which is essential in a research environment. AVS, with its modular approach, meets these needs. The problem then became providing an interface so SEP could take advantage of GOCAD's capabilities within our AVS environment. With `Project Manager` we feel we have met our preliminary needs. `Project Manager` operates similar to GOCAD's user interface. It creates a temporary project file (named Work.PJ in the user's home directory) in which it stores all of the active surfaces. From this knowledge base a variety of project and surface related functions can be performed.

Different functions are selected by navigating a menu tree that appears in the control panel portion of the AVS display (see Figure 1). such as gridding a model, an additional control panel is popped-up in the user's window environment (Figure 2). The `Project Manager's` functions can be broken into two categories: surface and project. Surface functions operate on a single surface, while the project functions concern an entire geologic model.

When a menu option that requires additional parameters is selected,

SURFACE OPTIONS

GOCAD's triangulated surface type (tsurf_t) class is described by vertices and series of triangles describing the connectivity between the vertices (Berlioux, 1994). `Project Manager` provides several options to create, edit, and save surfaces.

Display Surface

Display Surface has the same functionality as the `READ GOCAD` module in (Clapp et al., 1994). It allows the user to display any surface in the working project. The user can select the color of the surface by using the *color* widget.

In addition `Project Manger` uses AVS's ability to create object hierarchies. The object hierarchy can be helpful when dealing with complicated models where surfaces might represent faults, salt structures, and sand-shale interfaces. By using the

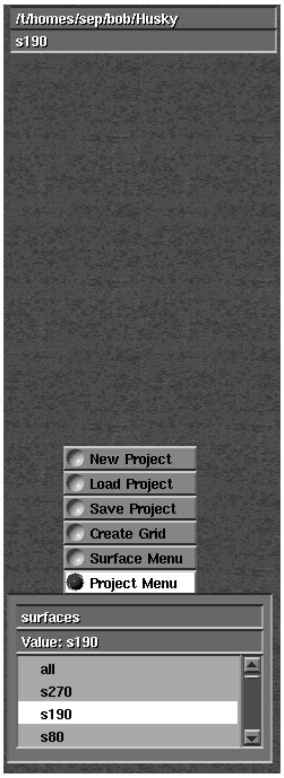

Figure 1: Control panel for Project Manager. Shows the name of the current project, current surface, provides a menu of possible options, and lists the surfaces in the current project. bob1-project [NR]

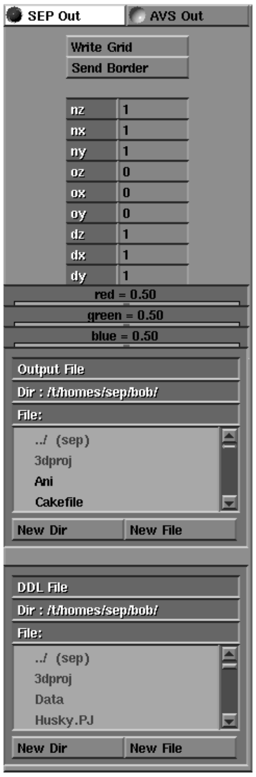

Figure 2: Control panel for Create Grid option. The interface allows the user to set type of output, execute modules, view boundaries of gridded region, set SEPLIB gridding parameters, name an output file, and select the DDL file containing the partition. bob1-creategrid [NR]

object hierarchy one can create a group, "faults", composed of all of the faults. Specific display parameters (color, transparency, etc.) can then be set for the object "faults". These parameters will automatically be transfered to all of the child objects of "faults", avoiding the time consuming task of setting the exact same parameter for each fault. The object hierarchy can be created by using the *new group* widget to create an AVS parent object to which the surface will belong. An object can be added to an existing parent object by using the *current group* widget.

Once the user has selected all the desired parameters, the surface is displayed by selecting it from the *surfaces* widget.

Edit Surface

This portion of `Project Manager` is still under construction. Eventually it will allow the altering of vertices (position and control node status) and DSI smoothing of surfaces.

Load Surface

Load Surface allows a user to load either binary or ASCII surfaces into the current project. When loading binary surfaces, it creates a temporary directory where it stores ASCII representations of the binary surfaces, then loads the surfaces into the current project (this feature is not directly available in GOCAD and has proven useful.)

When loading an ASCII surface a file browser widget is used to choose the file. For loading binary surfaces, a file browser widget is used to select the project, then a choice browser widget is used to select which (or all) the surface(s) to be loaded. In either case if a surface with the same name already exists in the current project the user is prompted whether or not to overwrite the current surface.

Delete Surface

Delete Surface simply deletes a surface (or all) surface(s) from the current project. Once the *delete surface* option is selected, the user choose which surface to delete from the *surfaces* widget. If "all" is selected, the user is prompted for reassurance.

Save ASCII

Using *Save ASCII* an individual surface, or the entire project can be saved in GOCAD's ASCII surface form. The user uses a file browser to select the destination. If they wish to save an entire project, the destination name will be the directory where all the surface files are stored, otherwise the selected name will be used for the chosen surface.

Create Surface

Create Surface follows a similar approach to that taken in (Clapp and Biondi, 1994). The user picks a single surface from a 3-D depth migrated cube. The *Create Surface* algorithm constructs GOCAD format surfaces using two kinds of GOCAD object: plines (polygonal lines) and tsurf_t (surfaces). A closed pline is constructed from the outer two lines picked from a 3-D dataset. The remaining picked lines are loaded in as a set of vertices (vset), GOCAD's DSI algorithm is then applied to attract the surfaces to the picked points.

`Project Manager` takes as input a 3-D AVS field which contains the picked lines, obtained from the `Create Line` module. The 3-D field has dimensions of 3 by maximum number of points selected in a line by number lines. The points in the field represent the location of a single surface picked from a 3-D cube. The *Surface Name* parameter is used to indicate the name of the new surface to be created. If the surface already exists in the project, the user will be asked to confirm the replacement of the existing surface.

The *N_Split* widget allows the user to indicate the number of times to split the surface's triangles. Each split adds a point in the middle of each segment splitting one triangle into four triangles. With the additional points the DSI has more flexibility to smooth the surface. The additional flexibility is needed because the original picked points are set as immovable. The *Number of Iteration* widget allows the user specify the number of iterations of the DSI algorithm. Once the user has finished with the parameter specification, the *write Project* boolean should be selected.

PROJECT FUNCTIONS

GOCAD offers two forms of representation for its 3-D objects: one ASCII and one binary. The ASCII form is more portable and easier to edit by hand. The binary form is more compact and allows associated geometry objects, like an entire 3-D model, to be combined intelligently. `Project Manager` uses the binary form as its main representation tool, but can handle either format. To effectively deal with complete 3-D models, several tools that relate directly to a project as a whole are necessary.

New Project

New Project allows the user to create a new GOCAD project. The user is presented with a file browser widget to select the path of the new project and the *new file* option to name the project.

Open Project

Open Project loads an existing project into memory. This function simply copies the selected project file to the temporary directory, then use the *Open Project* subroutine of GOCAD. The project is selected by using the file browser widget to navigate the directory tree and selecting is done by selecting any file within the desired project.

Save Project

Save Project allows the user to save a project. The function closes the existing working project and then recursively copies the project directory to the *Current Project* location.

Grid Project

The gridding function provides a link between the surface representation of GOCAD and the regular grids of SEPLIB. The *Create Grid* option is an AVS version of the GOCAD to SEP gridding program (Berlioux, 1993). Berlioux's generates a 3D velocity grid from a GOCAD model by shooting a ray from all the nodes on the top of the grid and locating where they intersect with surfaces in the model. A GOCAD function is then called to determine the domains that were encountered by the ray.

To define the velocity that goes in a given domain a DDL file is necessary. The DDL file defines the partition of the 3-D space. Berlioux (1993) presents an example of a GOCAD model with the corresponding partition file. Currently the creation the DDL file is done by hand, eventually the ability to create simple DDL files will be added to the module. One change was made to Berlioux's gridding algorithm, the automatic creation of a "free surface", normally the water surface, or ground level. If the "freesurf" surface does not exist in the GOCAD project, it is created temporarily during the execution of the gridding program.

Figure 2 shows the AVS user interface to the *Create Grid* module. The browser widget *Project Name* allows the user to select the GOCAD Project File which contains the model surfaces. The file browser widget *DDL File* allows the user to select the partition file which defines the domains of the model. Two possible outputs are allowed, *SEPLIB* or *AVS*. The two possible output format option is provided for large grids which might exceed AVS's memory limitations When SEPLIB is chosen for the output format, the file browser widget *Output File* allows the specification of the output file where the grid will be save to in standard SEP format. The typein widgets *n1, n2, n3* correspond to the number of grid cells on axis x, y, and z. The widgets *o1, o2, o3* indicate the origin of the grid and *d1, d2, d3* indicate the sampling intervals on axis x, y and z. The grid limits should be smaller than the limits of the surfaces. The widget *Display Border* allows the creation of a geometric object delimiting the limit of the grid that will be created. The colors of the border can be changed through the *Red* , *Blue* and *Green* slider widgets. Finally, once the user has set all the parameters, the *Write Grid* boolean should be selected to create and save the Grid.

3-D DATA EXAMPLE: BUILDING A 3D VELOCITY MODEL

To test the applicability of our model building method and to test the effectiveness of the `Project Manger` as an integrating tool we built a velocity model from a 3-D post-stack dataset provided by INTEVEP S.A. The data was first depth migrated with a velocity model determined from stacking velocity estimation. The resulting 3-D cube was then displayed in AVS, and surfaces were identified and picked. A velocity model was built by defining the velocity within the various domains delineated by the picked surfaces.

Building the surfaces

The creation of each surface was a two-step interactive process. In the first step events are picked on orthogonal vertical slices of the data using the modules `Pick Geom` and `Create Line` as is explained in (Clapp, 1994) The resulting 3-D field was then sent to the `Project Manager` module and a GOCAD surface was created using the *Create Surface* option. These steps were repeated for each surface constructed. Figure 3 shows the AVS network used to construct the GOCAD model.

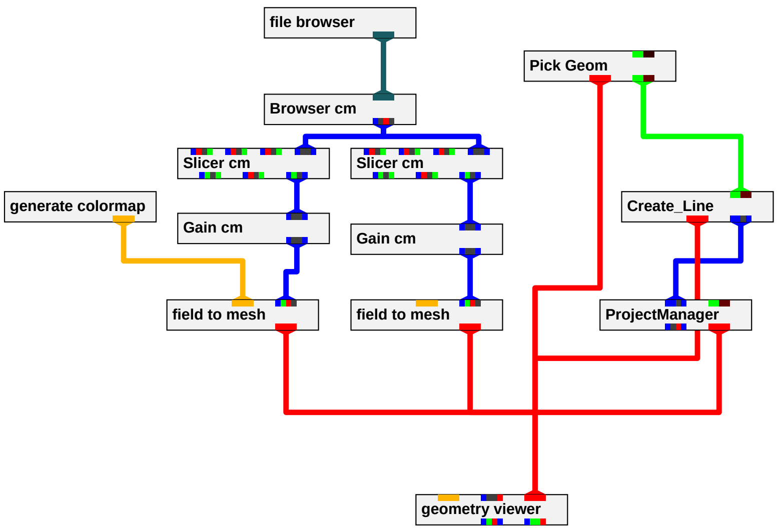

Figure 3: AVS network for picking events and building surfaces from a seismic dataset. bob1-Surfnet [NR]

Modules `browser cm`, `slicer cm` and `gain cm` are presented in (Biondi and van Trier, 1993); these remote modules take advantage of the data handling and computational capabilities of SEP's CM-5 allowing the management of large 3-D seismic surveys. The slice output by those modules, is passed to the local workstation to be visualized in a local `Geometry viewer` scene where events are picked. The `Pick Geom` and `Create line` are local modules, that is, they run on the local workstation. `Pick Geom` determines the location in modeling coordinates of a picked point in the `Geometry Viewer`. The location of the picked point, in this case from a slice of a 3-D cube, is sent to the `Create line` module. The `Create Line` module connects successively picked points to a form a line. Once the seismic event corresponding to a single reflector has been picked on a given slice, the *New Line* widget is selected in the `Create line` module. Another slice of the data is displayed using the `Slicer` widget in `Slicer cm` , and process is repeated until all the lines corresponding to the surface have been defined. Figure 4 shows the creation process after several lines have been selected. Finally, the *Write it* widget in the `Create line` module is selected to send the picked lines to the `Project Manager` module.

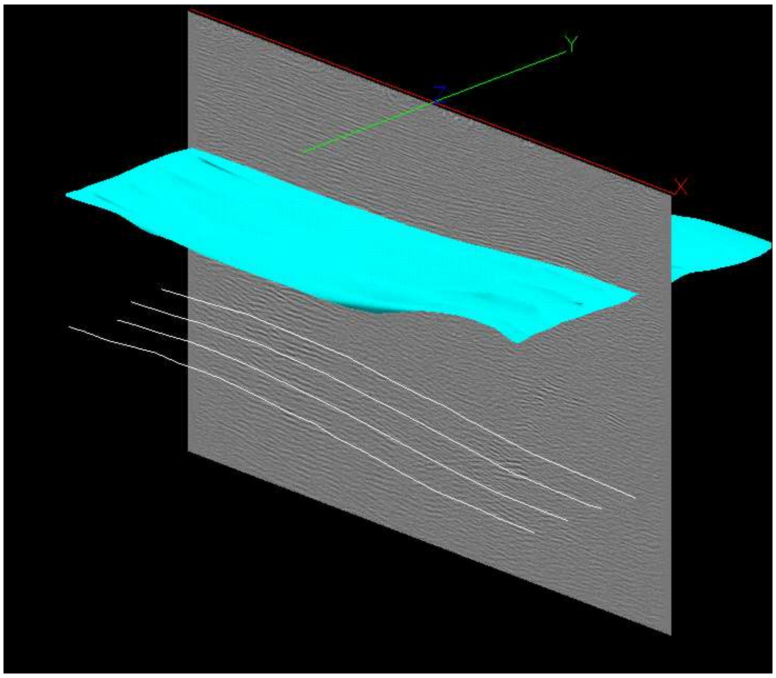

Figure 4: Geometry viewer scene with picked lines, a surface and a seismic data slice. bob1-3Dpick [NR]

Building the 3-D Grid

Once the model has been created a partition file (DDL file) must be created. The DDL file contains the definitions of the various domains (defined by their bounded surfaces) and the properties values (i.e. velocity) associated to each domain [see GOCAD model example presented in (Berlioux, 1993)]. The *Create Grid* option then allows the generation of the 3-D grid from a GOCAD model. Figure 5 shows an example of an AVS network to create a velocity grid from a GOCAD model.

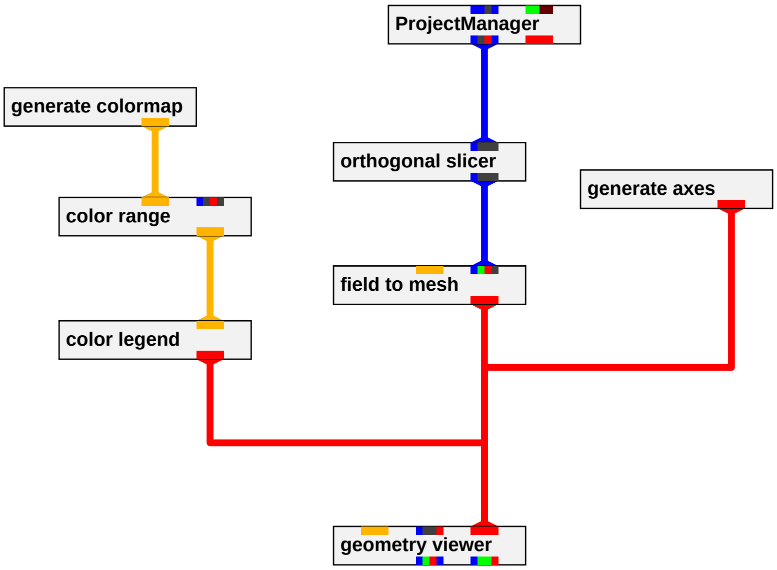

Figure 5: AVS network for creating a velocity Grid from a GOCAD model. bob1-Gridnet [NR]

Figure 6 shows the created surface model superimposed on the 2-D velocity and seismic data slices.

FUTURE WORK

Several additions are still necessary for a complete link between GOCAD and AVS:

- Completion of the *Edit* function.
- Updating to GOCAD++, which will soon be available at SEP.
- Ability to display well log data (e.g. velocity, dip meter, SP, etc).
- Improved gridding capabilities.

The final item poses the greatest problem. The current gridding algorithm is extremely limited in the range of models it can handle. In addition the velocity characterization is limited by the surface approach taken by GOCAD. The documentation of GOCAD++ states that it can grid constant velocity, compaction functions, and arbitrary x, y, and z functions within a layer. This is a significant improvement, but still not ideal. Ideally, we would like to be able at one stage to describe the velocity function within the domains at a very coarse level, and as our knowledge improves, to

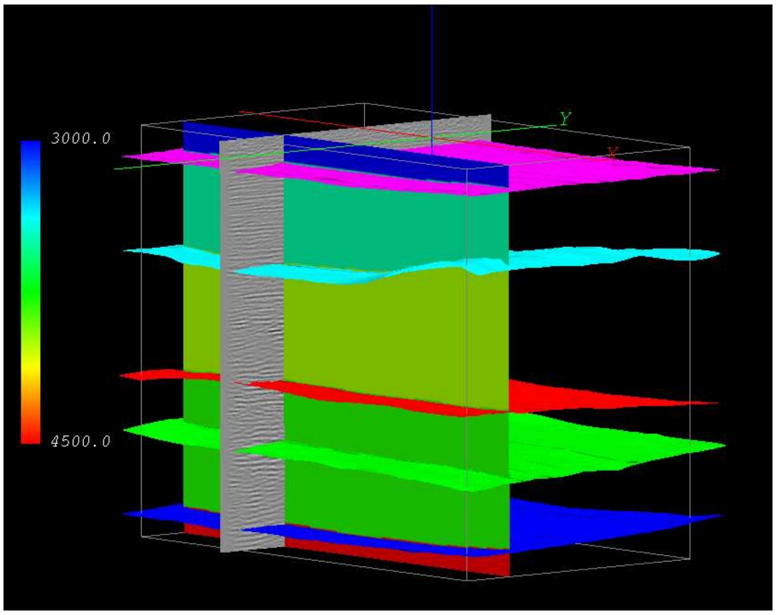

Figure 6: Geometry Viewer scene with the superimposed surfaces on the 2-D velocity and seismic data slices. bob1-3Dgrid [NR]

refine our definition. In addition we would like to have the connectivity between surfaces be recognized, eliminating the high frequency component of the velocity model that is common when characterizing domains. As a result SEP is considering two alternative approaches.

The first approach is to use tools already available in GOCAD. Specifically to characterize the velocity function within the domains by tetrahedra. As time goes along these tetrahedra become more numerous and get smaller. This approach would lean heavily on GOCAD's DSI algorithm to give reasonable values within the tetrahedra and for going from one level of model complexity to another. The boundary between domains would be smoothed by recognizing tetrahedra along the boundary and forcing a degree of continuity across the boundary.

The second alternative is to characterize the velocity function within the domains as a series of 3-D splines. This method is commonly used in 2-D problems and is effective in progressively refining the model (by adding additional coefficients). How to create continuity along the border of regions is unclear. One possible alternative is to define new domains along boundary regions that also have to be characterized by a limited number of spline coefficients.

CONCLUSIONS

The preceding represents the current status of SEP's 3-D model building tools. At this point SEP has incorporated most of GOCAD's important functionality in AVS.

Within AVS a geologic model can now be built, stored, and gridded. The AVS-GOCAD link was tested by the building of a 3-D velocity model using tools developed within SEP.

ACKNOWLEDGMENTS

We would like to thank to Arnaud Berlioux for his valuable support on GOCAD and his explanation of his gridding program.

REFERENCES

Berlioux, A., 1993, 3-D grid with GOCAD: SEP–**79**, 301–318.

Berlioux, A., 1994, Building models with GOCAD: SEP–**80**, 617–634.

Biondi, B., and van Trier, J., 1993, Visualization of multi-dimensional seismic dataset with cm-avs: SEP–**79**, 1–12.

Clapp, R. G., and Biondi, B., 1994, Iterative velocity model building for 3-D depth migration by integrating GOCAD and AVS: SEP–**80**, 635–644.

Clapp, R. G., Biondi, B., and Karrenbach, M., 1994, AVS as a 3-D seismic data visualizing platform: SEP–**82**, 97–106.

Clapp, R. G., 1994, Sep AVS user guide: SEP–**84**, 395–407.