Stanford Exploration Project, Report 93, November 11, 1997, pages 75–86

Visualization of irregularly sampled seismic data with AVS

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

keywords: *two-dimensional, three-dimensional*

ABSTRACT

We present a visualizer for multi-dimensional irregularly sampled data sets using SEPlib90 (SEP's library to handle irregularly sampled data) data access routines and AVS's (Application Visualization System) visualization environment. We developed two AVS modules that read, stack and regularize irregularly sampled data sets, and two modules that extract 2-D slices from n-dimensional data and generate the corresponding surface in 3-D space. These modules can be combined with standard AVS modules to display the data. Information on the reliability of interpolated data is also generated and displayed simultaneously with the seismic data for improved interpretation.

INTRODUCTION

Visualization of multi-dimensional, irregularly sampled data sets is useful in 3-D seismic as well as in many other geophysical problems, for example, to analyze AVO as a function of azimuth in CDP gathers in a 3-D land data set. The problem with most existing seismic visualizers, including SEP's primary multi-dimensional seismic visualizer (Biondi and van Trier, 1993), is that they operate on regularly sampled data. In the last year, SEP has been developing a generalization of the SEPlib software package, called SEPlib90(Biondi et al., 1996), capable of handling irregularly sampled data. Using SEPlib90 utilities, an irregularly sampled data set can be gridded onto a regularized space. We present here four AVS modules: `Read Sep90`, `Infill90`, `nD to 2D Slicer`, and `field to plane`, built using SEPlib90 data access routines, which can be combined with standard AVS modules to visualize multi-dimensional irregularly sampled data sets. The first module, `Read Sep90`, reads a SEPlib90 data set in its entirety or allowing a sub-set by specifying windowing parameters. The second module, `Infill90`, stacks a data set according to a user selected stacking axis and regularizes it by interpolating the data across a user selected dimension. This module also generates information on the reliability of regularized data. The third

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

module, `nD to 2D Slicer`, takes 2-D orthogonal slices from both the n-dimensional data array and the corresponding reliability information. The fourth module, `field to plane`, generates from a 2-D slice the corresponding surface in 3-D space, assigning vertex transparency values or modifying saturation values, according to the reliability information. This paper is in two parts. The first part describes the four new modules interface and operation. The second part presents an AVS network built using the new modules and some standard AVS modules, which can be use to visualize multi-dimensional irregularly sampled data.

AVS MODULES DEVELOPED

Read Sep90

The `Read Sep90` module reads a SEPlib90 data set into AVS using SEPlib90 data access routines. It also allows the selection of a sub-set by specifying windowing parameters, so the whole data set doesn't have to be loaded when memory is an issue. Figure 1 shows the AVS user interface for the `Read Sep90` module. It takes as input a SEPlib90 dataset and produces three outputs to downstream modules: a *Header Values* structure, a *Data Values* 2-D Field and a *Grid Values* n-D Field. The Header values structure below is a memory representation of both SEP90 *header keys* and *Header Values File*:

```
/*****************************************************
       Keys Object
******************************************************/
typedef struct KEY {
        char    name[MAXLABEL]; /* Name of Header Key */
        int     type;           /* Type of Header Key   */
        int     pos;            /* Header Key Position in Header Record */
        } KEY_o;

/*****************************************************
       Header Object
******************************************************/
typedef struct HEADER {
        int     n_keys;         /* Number of Header Keys */
        KEY_o   *keys;          /* Vector of lenght n_keys containig key
          objects       */
        int     n_records;      /* Number of Header Records */
        int     record_size;    /* Number of bytes in Header Records */
        char    *values;        /* Vector containing Records Values     */
        }  HEADER_o;
```

The *Data Values* 2-D Field is a memory representation of the SEPlib90 *Data Values File*. The SEPlib90 data types supported are: `scalar_byte, scalar_int, scalar_int, scalar_float, vec3_byte` and `vec3_float`. The corresponding AVS output types are: scalars `char, short, int, float`, and 3-vector `char` and `float`. The third output, `Grid Values`, provides information about the superimposed, regular grid. The The `Grid Values` file is a n-D, scalar *AVSfield_int*, where each value represents is equal to the index of the corresponding trace and header in the *Data Values* Field and *Header Values* structure[2] or `-1` if there is no trace at that gridding location.

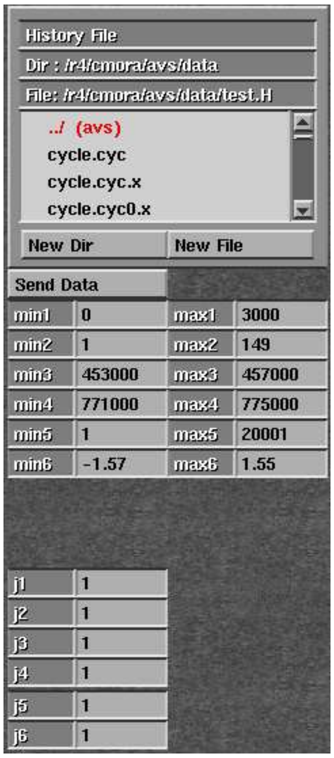

Figure 1: Control panel for the `Read Sep90` module. bob2-ReadSep90 [NR]

A `file browser` widget is used to select the name of the History file associated with the input data set. The `min` and `max` type-in widgets allow the specification of minimum and maximum values for each axis. Axis 1 refers to time axis, and Axis 2 to Axis n refer to the Gridding coordinate system axes. The `j` widgets allow the user to specify an under-sampling rate along each Gridding coordinate system axis. Once the user has finished with the parameter specification, the `Send Data` one-shot widget should be selected to read the SEPlib90 data set and pass the output information to downstream modules.

Infill90

The `Infill90` module stacks a data set according to a given stacking axis and regularizes it by linearly interpolating the data across a given dimension (later this will be expanded to multi-dimensional interpolation). This module generates information on the reliability of interpolated data which can be displayed simultaneously with the

[2]Because of efficiency issues, we require the data to be in the same order as the headers (same_record_number = 1).

seismic data to help accurate interpretation (for example whether a given anomaly is caused by an interpolation artifact, unusually low fold in a region, or is due to some actual data irregularity). It takes as input a 2-D AVS Field representing the Data Values and a n-D AVS Field representing the Grid values (typically both are output for `Read Sep90` module), and produces as output a (n+1)-D AVS Field representing the regularized data and a n-D AVSfloat Field representing the reliability information associate to each trace.

Figure 2 shows the user interface for this module. The `Stack` boolean widget,

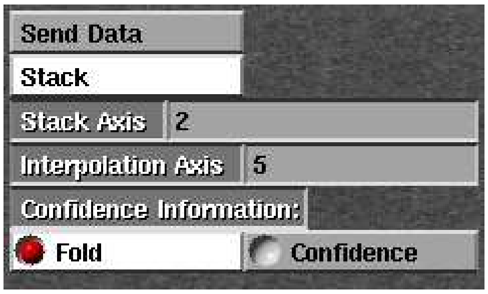

Figure 2: Control panel for the `Infill90` module. bob2-Infill90 [NR]

allows the user to specify if the data should be stacked according to the stacking axis indicated by the `Stack Axis` type-in widget (reducing data volume) or sent in its entirety down stream. The `Interpolate Axis` widget allows the user to specify the dimension to perform linear interpolation on. The choice widgets `Fold` and `Confidence`, allows the users to select the type of reliability information that should be generated. If `Fold` is selected, the reliability information will based on information obtained by stacking along the `Stack Axis`. The number of traces corresponding to a given bin (where bin is defined as the output trace obtained by stacking) is normalized by maximum number of traces (resulting float values between 0. and 1.). If `Confidence` is selected, the output will represent how close each trace is to known values. Values equal to 1 are associated to known traces and values less than 1 are associated to interpolated traces. The reliability information value associate to interpolated traces close to known traces will be higher than those associated to interpolate traces far from known traces. In both cases, values close to 1 indicate reliable data. Finally, once the user has finished with the parameter specification, the `Send Data` one-shot widget should be selected to execute the module and pass the output information to downstream modules.

nD to 2D Slicer

The `nD to 2D Slicer` generates 2-D slices from both the regularized data field and the associated reliability information. This module receives as input a n-D field representing a regularized data set and a (n-1)-D field with the associated reliability information. The user interface for this module is shown in Figure 3.

The user must select a 2-D field from the n-D volume. This is done by

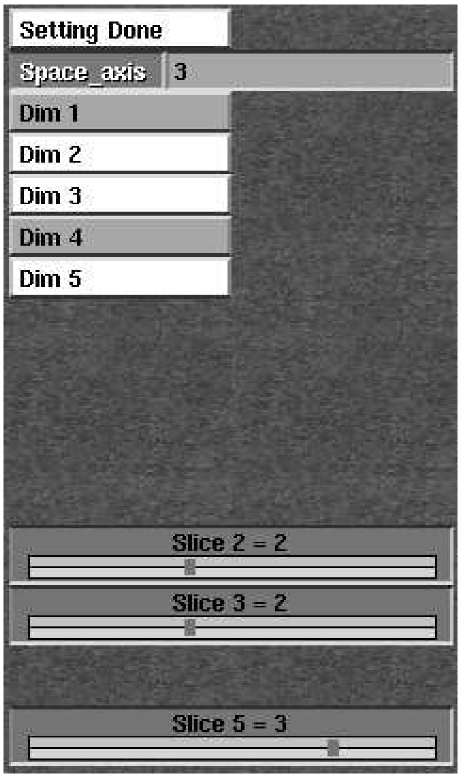

Figure 3: Control panel for the `nD to 2D` Slicer module. bob2-nDto2D [NR]

- using the `dim i` widgets, for which dimensions will be constant in the slice and
- selecting the constant value with the `slice i` widget.

Because the output is a 2-D field but we wish to view the data in a 3-D space a third space axis must be specified using the `Space Axis` widget. The `Setting Done` boolean, passes the output information to the downstream modules.

field to plane

The `field to plane` module receives as input a 2-D data slice, the associated reliability information, and two color maps, and generates as an output an AVS geometry. Figure 4 shows the user interface to this module.

Figure 4: Control panel for the `field to plane module.` bob2-fieldtoplane [NR]

The choice widgets `Transparency` and `Saturation`, allow the users to select how the reliability information associated to the data will be displayed. If `Transparency` is selected, the output surface will have vertex transparency values according to the reliability information, if `Saturation` is selected, the reliability information will be used to modify color saturation values of the vertices. The `Tunning` widget allows the user to adjust the contrast between real and interpolated traces. The `Invert Confidence` widget inverts the reliability values. In this case, the value associated to known traces will be 0 and values associated to interpolated traces will be greater than 0. This can be very useful if the user wants to display the interpolated traces rather than real traces.

AVS NETWORK TO VISUALIZE IRREGULARLY SAMPLED DATA

Using the new modules described above, and combining them with standard AVS modules, we built an AVS network to visualize irregularly sampled data. Figure 5 shows an example network. In this network, `Read Sep90`, `Infill90` and `nD to 2D Slicer` modules are connected directly passing each output to downstream modules, reliability information from the `nD to 2D Slicer` module is also directly connected to the `field to plane` module while data output is passed through the standard AVS module `clamp` in order to clip the data values. Finally, the `field to plane` output is passed to AVS's `geometry viewer`.

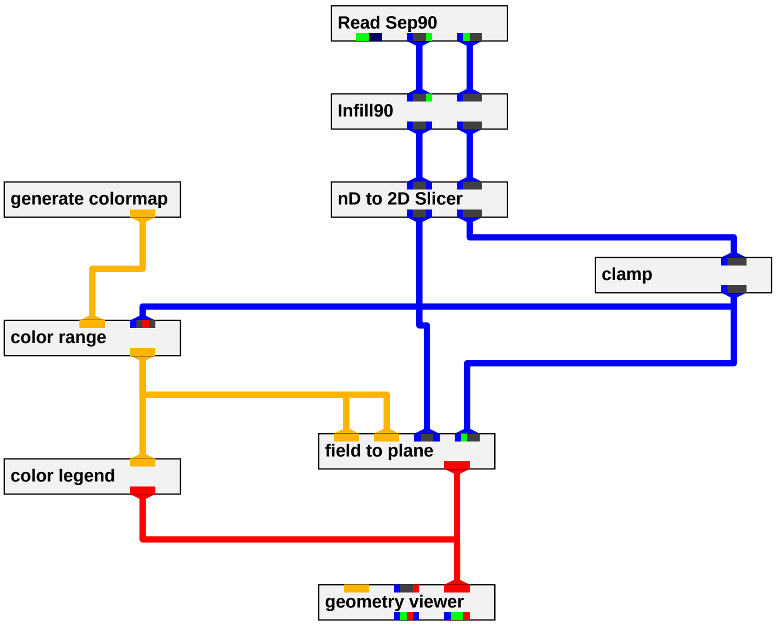

Figure 5: AVS network to visualize irregularly sampled data. bob2-net [NR]

We tested the new modules with a 3-D land survey provided by Arco. This data set was acquired in a Button-patch geometry. We gridded the data onto a 6-D space (`time, trace_number_in_bin, cmp_x, cmp_y, AOFFSET` and `azimuth`). Using the `Infill90` module we stacked the data across `trace_number_in_bin` axis and interpolated it across the `cmp_x` axis. We took slices to display the `time` and `AOFFSET` dimensions. Figure 6 shows the resulting geometry displayed without any reliability information and Figure 7 shows the same slice with the trace transparency indicating low fold coverage. Figure 8 show the same slice with transparency indicating high fold coverage (confidence inverted). Finally, Figure 9 shows two slices corresponding to different azimuth values. In this the case transparency indicate interpolated traces.

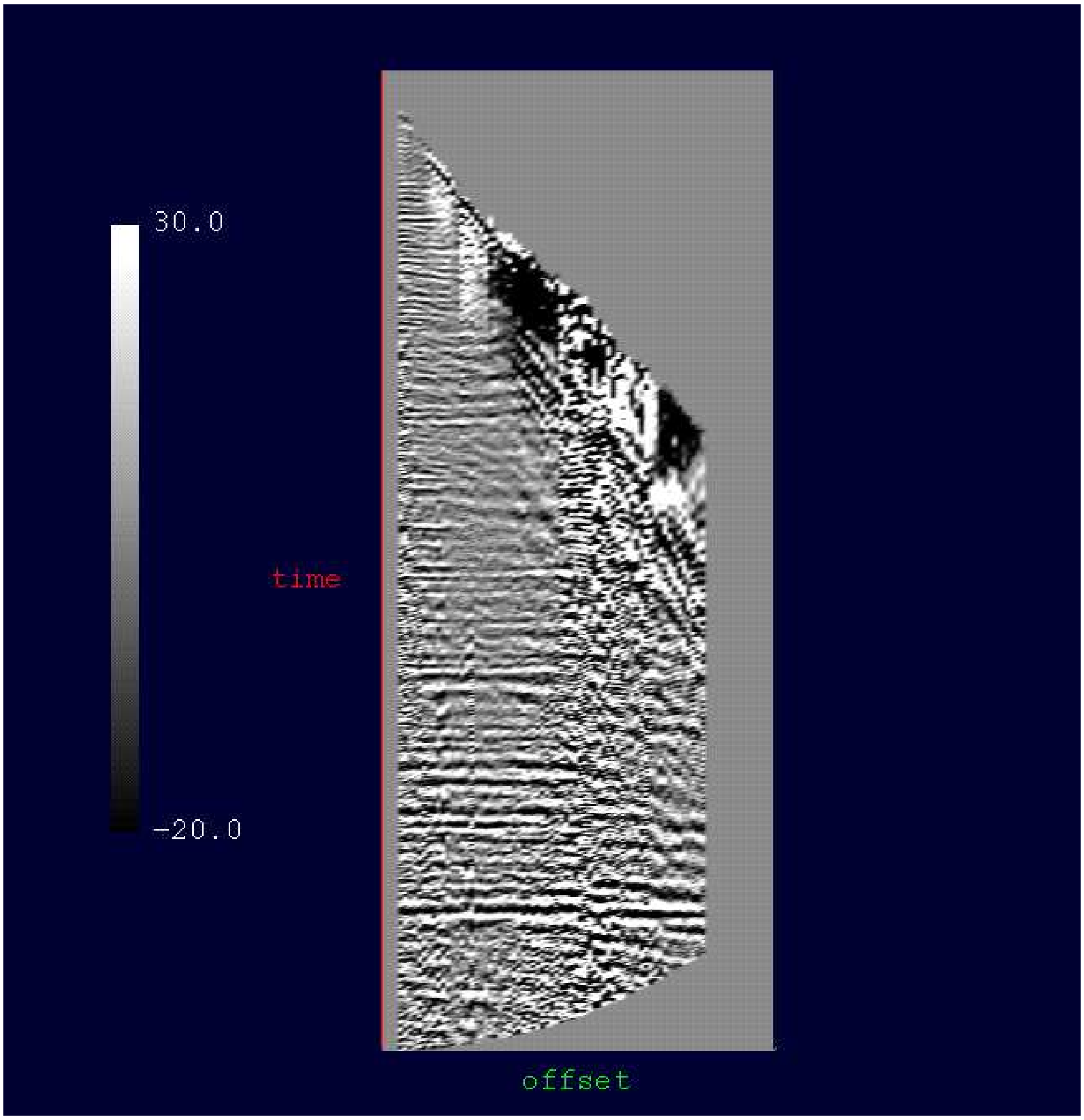

Figure 6: CMP gather display without confidence information. bob2-cmpwc [NR]

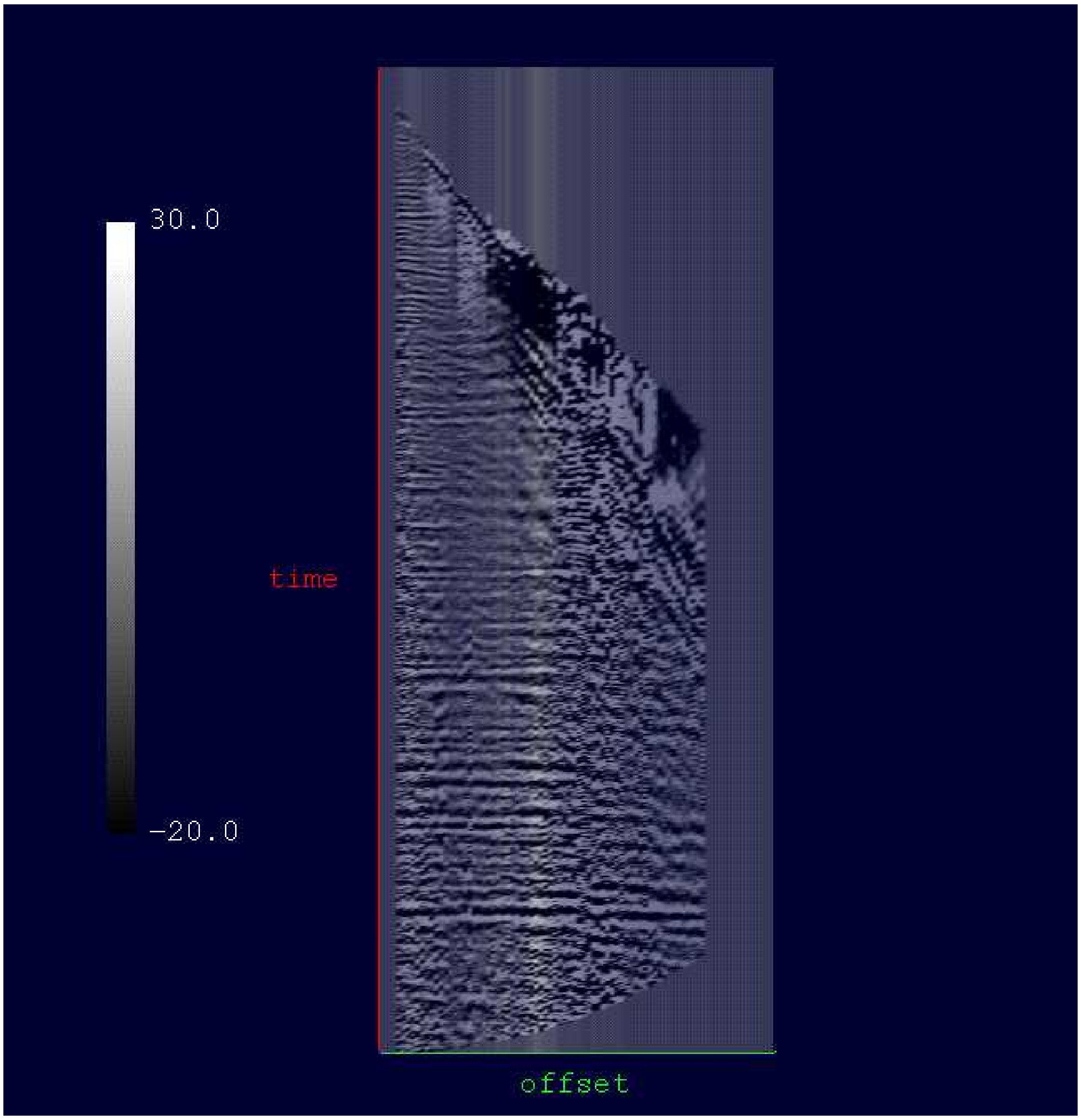

Figure 7: CMP gather. Low fold regions are more transparent. In this black and white image, transparency is not as evident, areas that are fairly transparent appear lighter than higher fold, less transparent regions. bob2-cmpc [NR]

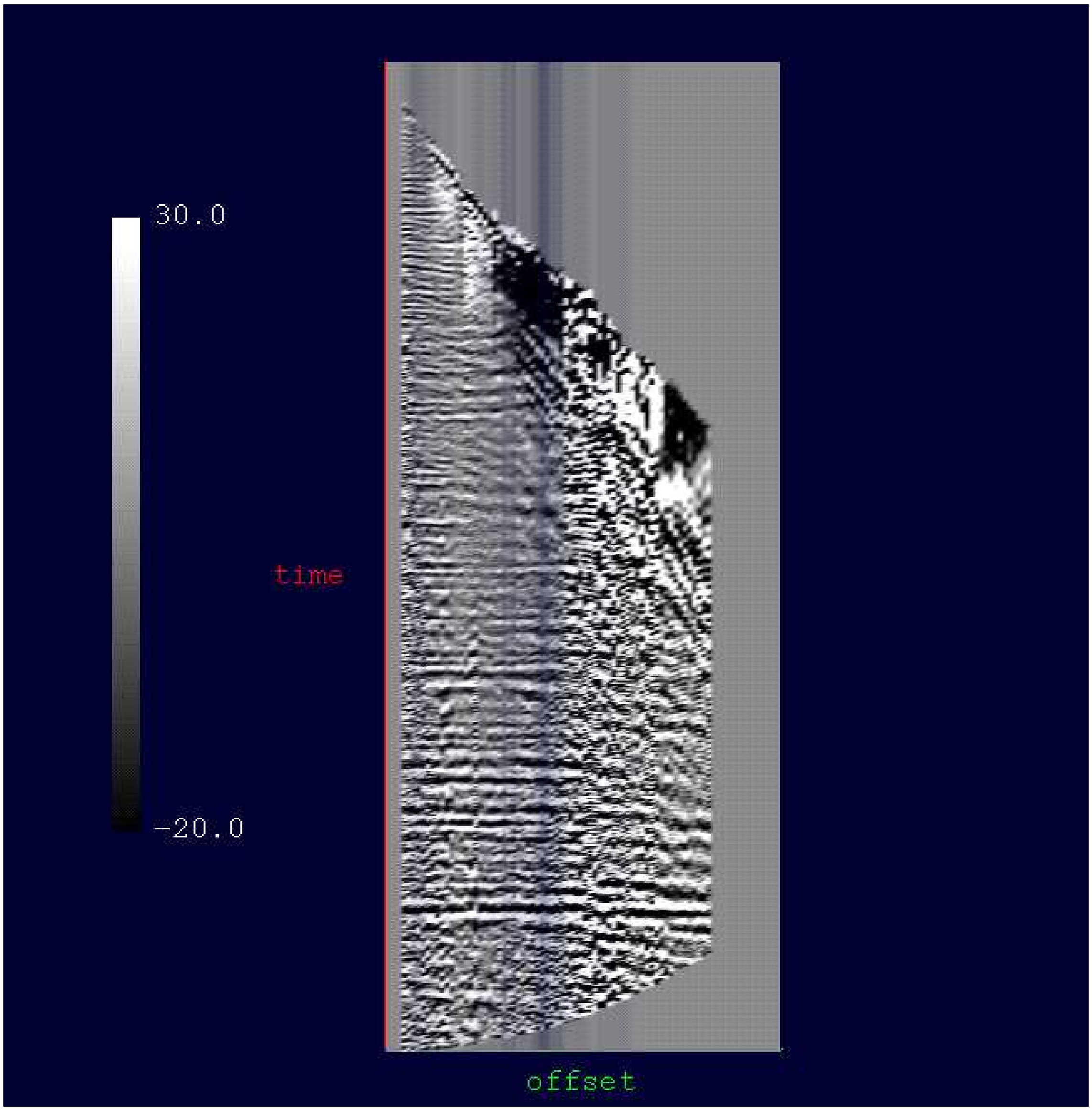

Figure 8: CMP gather. High fold regions are more transparent. bob2-cmpci [NR]

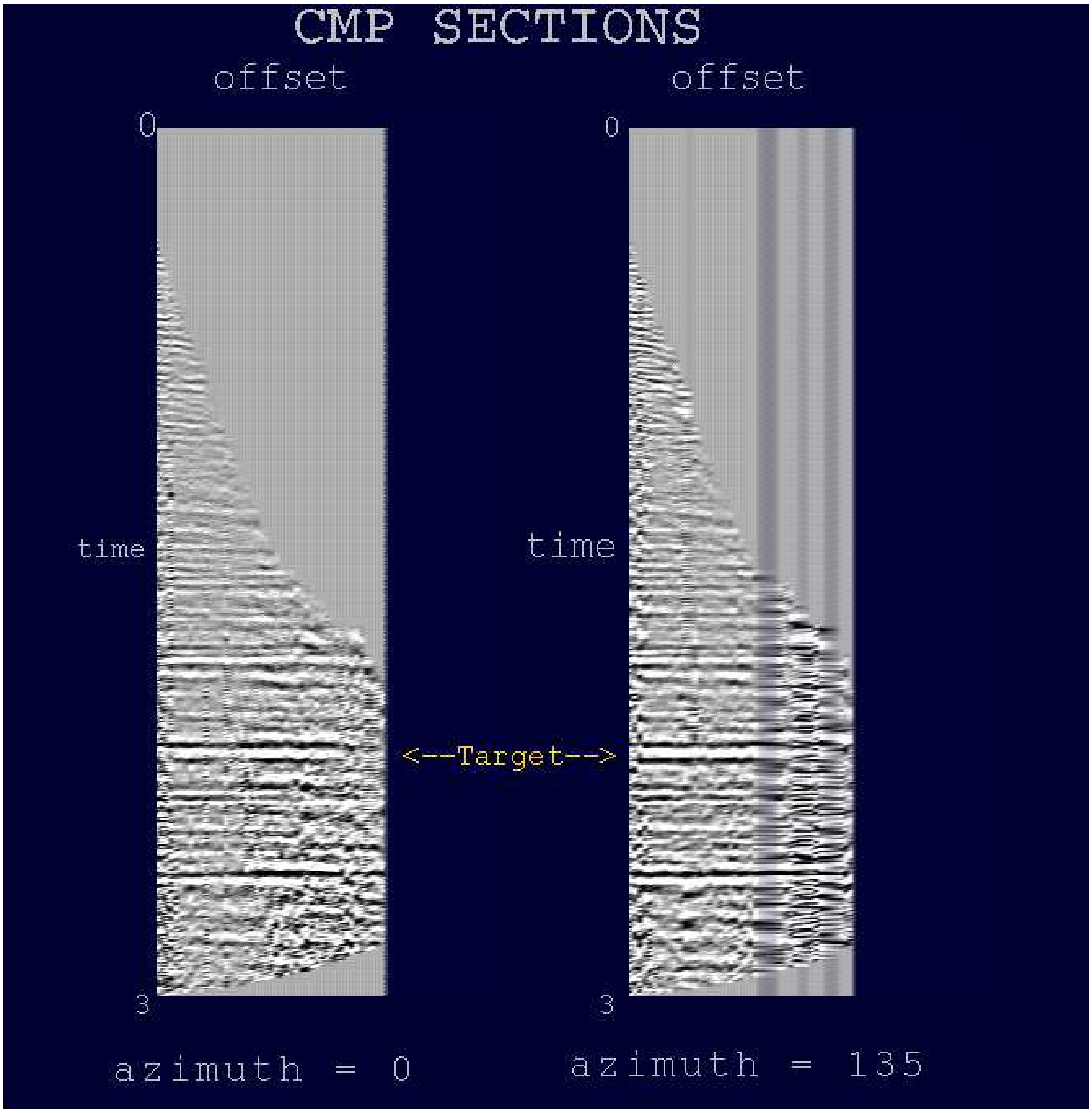

Figure 9: CMP gathers corresponding to different azimuth values, transparency indicates interpolated traces bob2-cmpgather [NR]

CONCLUSIONS

We have presented a visualizer for multi-dimensional irregularly sampled data built using SEPlib90 and AVS. This visualizer adds the ability to visualize irregularly sampled 3-D prestack data to our 3-D seismic data visualizing platform. Information on the reliability of interpolated data can be displayed together with the data to improve the accuracy of the interpretation.

REFERENCES

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

Biondi, B., Clapp, R. G., and Crawley, S., 1996, Seplib90: Seplib for 3-d prestack data: SEP–**92**, 343–364.