

David M. Chen and Robert G. Clapp

Data-fusion of volumes, visualization of paths, and revision of viewing sequences in Ricksep

ABSTRACT

Ricksep is a freely-available interactive viewer for multi-dimensional cubes, capable of simultaneous display of multiple data sets from different viewing angles, animation of movement through the data space, and selection of local regions for data processing and information extraction. Several new features are added to Ricksep to enhance the program's functionality. First, two new data-fusion algorithms synthesize a new data set from two source data sets, one with mostly high-frequency content, such as seismic data, and the other with mostly low-frequency content, such as velocity data. Previously separated high and low-frequency details can now be viewed together. Second, a new projection algorithm, integrated with Ricksep's point-picking capabilities, effectively displays arbitrary paths through the data space. The algorithm responds to path changes in real time, restores depth information lost through ordinary projection techniques, and supports the generation of multiple paths differentiated by point-picking symbols. Third, a viewing history list is maintained to enable Ricksep's users to edit and save a sequence of viewing states. The feature supports undoing and redoing of viewing commands and animation of viewing sequences, a generalization of the viewer's movie feature. A theoretical discussion and several examples using real seismic data show how the new features offer more convenient, accurate ways to manipulate multi-dimensional data sets.

INTRODUCTION

Ricksep is a freely-available interactive viewer created to study multi-dimensional seismic data volumes (Clapp, 2001; Clapp et al., 2001). The viewer follows earlier efforts by SEP to visualize and interact with large multi-dimensional volumes (Claerbout, 1981; Ottolini, 1990; Biondi and van Trier, 1993; Clapp, 1995; Mora et al., 1996). Among Ricksep's most notable features currently are simultaneous display of multiple data volumes from different viewing angles, real-time animation of movement through a volume, and convenient selection of local regions for processing. Since Ricksep was last released, there has been interest in improving the program to solve three important issues. First, in velocity analysis, a seismic volume containing reflection data and a related velocity volume generated from a model are often compared alongside one another. The seismic volume contains many localized geophysical structures, which correspond to high-frequency spatial variations. Meanwhile, the velocity volume changes slowly, resulting in low-frequency spatial variations. To accurately and con-

veniently study the separated high and low-frequency details together, it would be very useful to synthesize a single volume that resembles the original high and low-frequency volumes on small and large scales, respectively. Second, there can be difficulty in displaying a path through a data volume of dimensionality three or higher. Paths are important to seismic data analysis because they represent wells going into the ground. If the data space is two-dimensional, the entire space can be seen and visually tracing out the path is straightforward. If dimensionality is three or higher, however, it becomes very difficult, and in some cases impossible, to see samples along the path as the path travels into the volume's interior. An effective path visualization method is needed for general multi-dimensional volumes. Third, one of Ricksep's strengths is its large set of options for viewing a volume. These options include jumping to a point in the volume, zooming in on a local region, applying various color maps, and switching perspectives. If all such viewing changes can be recorded in a history list, then the program can allow users to backtrack their viewing decisions, much like how a word processor undoes typing changes. The saved history list can be used later for automation of viewing and data processing commands. A new set of features has been created for Ricksep to solve these problems. In Section 2, the Velocity View tool for synthesizing a single volume from high and low-frequency volumes is described. In Section 3, the Path View tool for projecting samples along a path is presented. History List is the new tool for editing and saving a sequence of viewing commands in a Ricksep session, and its functionality is explained in Section 4. The keyboard and menu controls for Velocity View, Path View, and History List are summarized in Section 5. Finally, the conclusion in Section 6 suggests future improvements for the features introduced in this paper.

VELOCITY VIEW: DATA-FUSION OF VOLUMES

Separate spectral information in source volumes

High and low-frequency volumes intrinsically convey different types of geophysical information which are often analyzed together. Fig. 1 shows slices of a high-frequency, seismic volume and a corresponding low-frequency, velocity volume on the left and right, respectively. Although users can try to visually match localized structures on the left to large-scale patterns on the right, it would be easier and more accurate to study a single volume that superimposes the localized structures on the background patterns. Our intended approach is a form of feature-level image data-fusion, the synthesis of a single image using features from multiple source images (Bloch and Maitre, 1997). Differences between the two source volumes can also be understood in the spatial-frequency domain. Let $h(\vec{x})$ and $l(\vec{x})$ denote the high and low-frequency volumes, respectively, where $\vec{x} = (x_1, x_2, \dots, x_M)$ is a vector of dimensionality M that ranges over the entire data space. Similarly, let $H(\vec{w})$ and $L(\vec{w})$ denote the M -dimensional Fourier transforms of $h(\vec{x})$ and $l(\vec{x})$, where $\vec{w} = (w_1, w_2, \dots, w_M)$ is a vector ranging over the entire spatial-frequency space. Spectrum $H(\vec{w})$ can be separated into two regions: a low-frequency region containing mainly the gray background and a high-frequency region containing the fine structures. Similarly, $L(\vec{w})$ can be divided into a low-frequency region containing the patterns we want and a high-frequency region almost void of content.

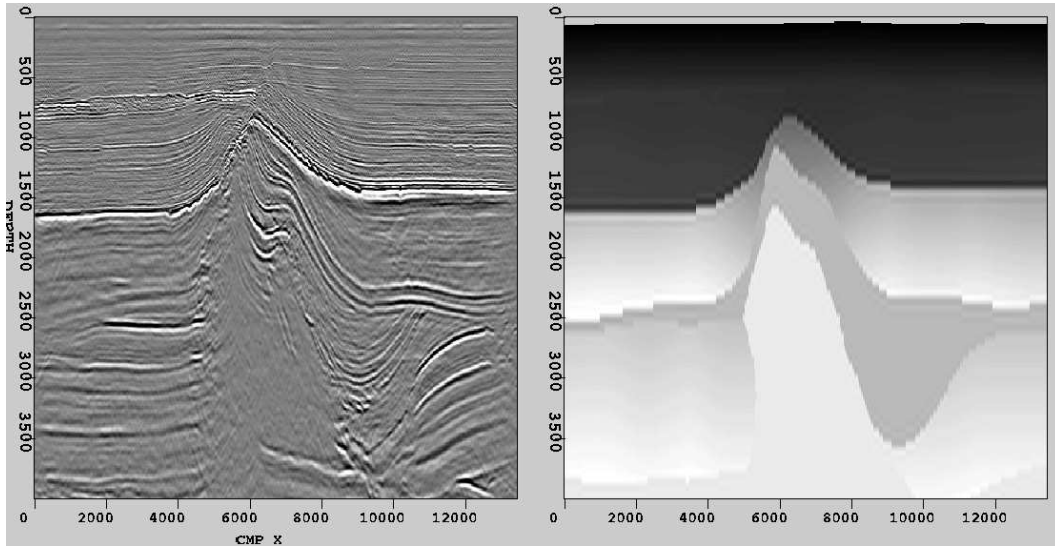


Figure 1: Seismic (left) and velocity (right) data. `bob3-chen-cubes-orig` [NR]

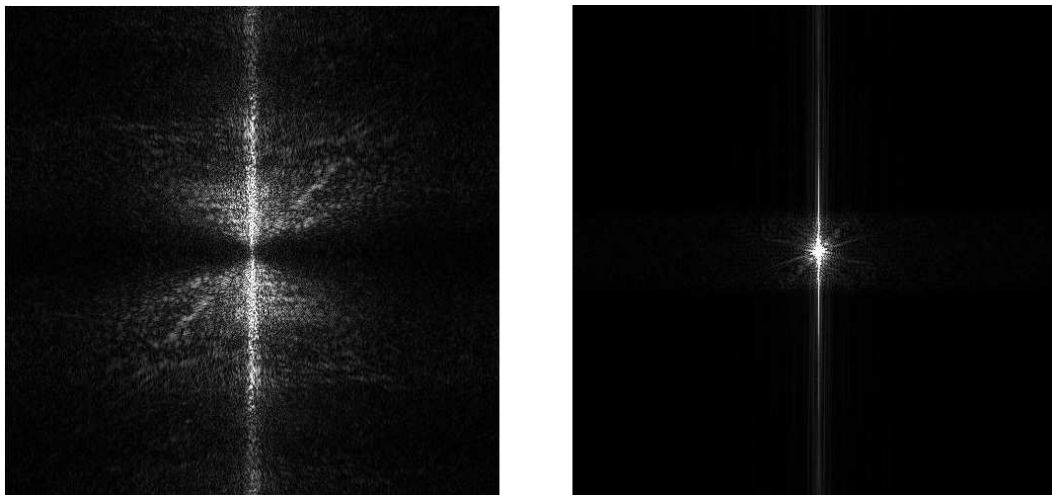


Figure 2: Magnitude spectra of seismic (left) and velocity (right) data. `bob3-chen-fft-orig` [NR]

These facts are confirmed in Fig. 2, which shows the magnitude spectra of the slices from Fig. 1. In this example, $L(\vec{w})$ is nearly zero in all high-frequency regions except around the vertical line $w_1 = 0$, where the regional boundaries in $l(\vec{x})$ introduce high-frequency components. If we add the high-frequency components of $H(\vec{w})$ to $L(\vec{w})$, we would synthesize a new volume that approximately has the background patterns of $l(\vec{x})$ and the foreground details of $h(\vec{x})$. The two data-fusion algorithms presented in Sections 2.2-2.3 follow the framework depicted in Fig. 3. High and low-frequency components are extracted from the two source volumes. These components are properly scaled and added together to form the new volume. The framework can also be used to develop other frequency-selective data-fusion algorithms.

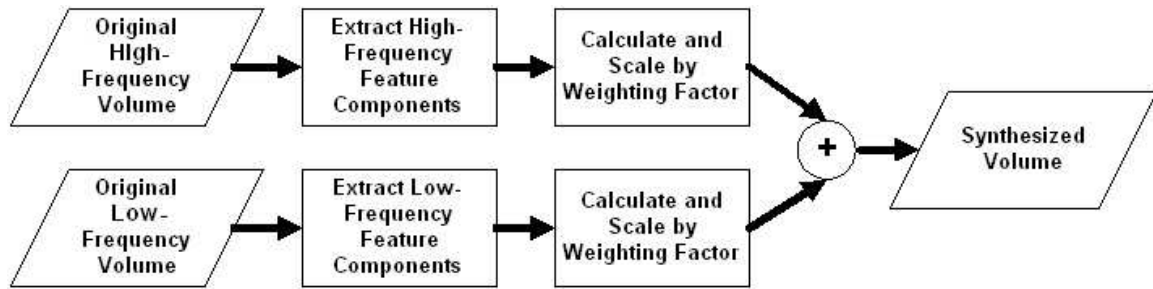


Figure 3: Framework for data-fusion of source volumes. bob3-chen-velview-blocks [NR]

Data-fusion using small intensity deviations

Volume $h(\vec{x})$ will have a global mean, which we take to be the sample mean

$$\bar{h} = \frac{1}{V(S)} \int_S h(\vec{x}) d\vec{x}, \quad (1)$$

$$V(S) = \int_S d\vec{x}, \quad (2)$$

where S represents the data space and the integrals are taken over all S . The value \bar{h} represents the intensity of the gray background of $h(\vec{x})$. If we define a quantity $d(\vec{x}) = h(\vec{x}) - \bar{h}$ for all \vec{x} , we have a measure of the local intensity deviation from the intensity of the gray background. These deviations produce the local structures observed in $h(\vec{x})$. The scarcity of such deviations in $l(\vec{x})$ corresponds to the near-absence of local structures there. It becomes evident that a new volume $m_1(\vec{x}) = l(\vec{x}) + d(\vec{x})$ should be created. This approach adjusts the slowly changing background intensities of $l(\vec{x})$ by the rapidly changing intensity deviations from $h(\vec{x})$. Since volume $d(\vec{x})$ maps to $D(\vec{w}) = H(\vec{w}) - H(0)/V(S)$, with $V(S)$ defined by Eq. 2, the DC component of $H(\vec{w})$ corresponding to the gray background is mostly removed. Then, $M_1(\vec{w}) = L(\vec{w}) + D(\vec{w})$. The algorithm adds the primarily low-frequency $L(\vec{w})$ to the almost DC-less $D(\vec{w})$ to synthesize $M_1(\vec{w})$. In practice, a scaled $\alpha \cdot d(\vec{x})$ is added to $l(\vec{x})$ to avoid significant alteration of local means when going from $l(\vec{x})$ to $m_1(\vec{x})$. The exact value of alpha depends on the relative signal levels in $h(\vec{x})$ and $l(\vec{x})$. Relating back to the framework in Fig. 3, the upper branch's extraction of high-frequency components from $h(\vec{x})$ and scaling

are contained in the term $\alpha \cdot d(\vec{x})$. The lower branch's behavior is simpler: all frequency components are extracted from $l(\vec{x})$ and the scaling factor is unity. The synthesized result of the slices from Fig. 1 is shown in Fig. 4. It can be seen that the first algorithm performs well in presenting the fine details of $h(\vec{x})$. Because we avoided significant alteration of the local means of $l(\vec{x})$ through the scaling factor α , the textural smoothness and regional boundaries of $l(\vec{x})$ are well preserved.

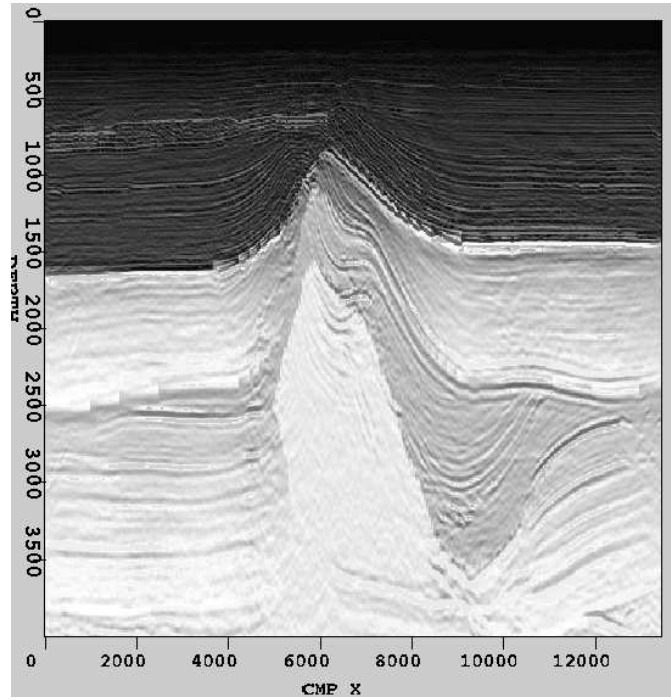


Figure 4: Result using the first data-fusion algorithm. `bob3-chen-velview-merge1` [NR]

Data-fusion using deviation-dependent weighting

Instead of adjusting the intensities of $l(\vec{x})$, the second algorithm synthesizes a new volume from a convex combination of the source volumes $h(\vec{x})$ and $l(\vec{x})$. The relative levels of contribution from $h(\vec{x})$ and $l(\vec{x})$ are determined by how much $h(\vec{x})$ deviates from \bar{h} , where \bar{h} is defined by Eq. 1 and is the intensity of the gray background in $h(\vec{x})$. Specifically, we form

$$m_2(\vec{x}) = \beta(\vec{x}) \cdot h(\vec{x}) + (1 - \beta(\vec{x})) \cdot l(\vec{x}), \quad (3)$$

$$\beta(\vec{x}) = \frac{|h(\vec{x}) - \bar{h}|}{\max\{|h(\vec{x}) - \bar{h}| : \vec{x} \in S\}}, \quad (4)$$

where S is the data space. As can be seen from Eq. 4, $\beta(\vec{x})$ is spatially-varying weighting factor between 0 and 1. As $\beta(\vec{x}) \rightarrow 1$, corresponding to large deviations from \bar{h} , $m_2(\vec{x}) \rightarrow h(\vec{x})$. For $\beta(\vec{x}) \rightarrow 0$, corresponding to small deviations from \bar{h} , $m_2(\vec{x}) \rightarrow l(\vec{x})$. The result $m_2(x)$ leans more towards $h(\vec{x})$ where local structures appear and more towards $l(\vec{x})$ otherwise. Another advantage of using the convex combination in Eq. 2 is that if $\beta(\vec{x})$ varies smoothly

in space, then the synthesized volume $m_2(\vec{x})$ will also vary smoothly in space. Here, in terms of the framework in Fig. 3, the upper branch's extraction of high-frequency components from $h(\vec{x})$ and scaling depicted are contained in the term $\beta(\vec{x}) \cdot h(\vec{x})$. Similarly, the lower branch's extraction of low-frequency components from $l(\vec{x})$ and scaling are contained in the term $(1 - \beta(\vec{x})) \cdot l(\vec{x})$. When combining the slices from Fig. 1, we obtained the synthesized result shown in Fig. 5. The advantage of the second algorithm is that it synthesizes the local structures from $h(\vec{x})$ in the new volume more accurately than does the first algorithm, but this is done at the expense of sacrificing a small amount of textural smoothness inherited from $l(\vec{x})$. Again, the local means from $l(\vec{x})$ are well preserved.

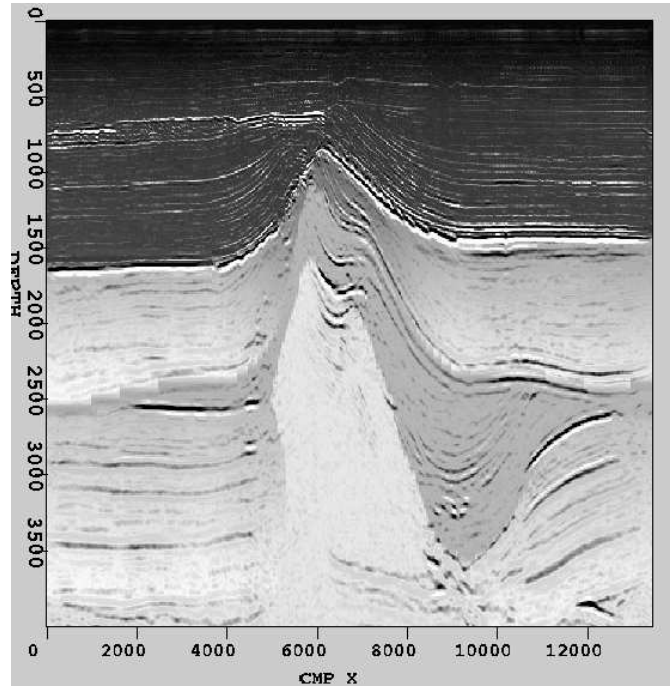


Figure 5: Result using the second data-fusion algorithm. bob3-chen-velview-merge2 [NR]

Flexible control of contributions from sources

In practice, the user will often want to control how closely the synthesized volume resembles the low-frequency volume rather than the high-frequency volume, or vice versa. The interface for Velocity View provides a slider bar to control the level of contribution to the result $m(\vec{x})$ from the source volumes $h(\vec{x})$ and $l(\vec{x})$. As the slider moves from left to right, a linear scale from 0 to 1 is traversed. Denoting the current slider value γ , $m(\vec{x})$ is generated by

$$m(\vec{x}) = \begin{cases} (2 - 2\gamma) \cdot m_i(\vec{x}) + (2\gamma - 1) \cdot l(\vec{x}) & \text{if } \gamma > 0.5 \\ (2\gamma) \cdot m_i(\vec{x}) + (1 - 2\gamma) \cdot h(\vec{x}) & \text{if } \gamma \leq 0.5 \end{cases}, \quad (5)$$

where $i = \{1, 2\}$ indicates which one of the two data-fusion algorithms previously discussed is used. The construction in Eq. 5 has the following desirable behavior that 1) as $\gamma \rightarrow 0$,

$m(\vec{x}) \rightarrow h(\vec{x})$; 2) as $\gamma \rightarrow 1$, $m(\vec{x}) \rightarrow l(\vec{x})$; and 3) as $\gamma \rightarrow 0.5$, $m(\vec{x}) \rightarrow m_i(\vec{x})$. For any value of γ , $m(\vec{x})$ is defined to be a convex combination of $m_i(\vec{x})$ and either $h(\vec{x})$ or $l(\vec{x})$, depending on whether or not $\gamma > 0.5$.

PATH VIEW: VISUALIZATION OF PATHS

Interpolating a path from sample points

A path through a volume can be defined as a parametric curve $\vec{x}(t) = (x_1(t), x_2(t), \dots, x_M(t))$, where M is the number of dimensions and t is a free parameter. In practice, though, what a user specifies is not a continuous $\vec{x}(t)$ but rather a finite, discrete set of vectors corresponding to points along $\vec{x}(t)$, say $\{\vec{x}_i : 1 \leq i \leq N\}$. In each dimension j ($1 \leq j \leq M$), a set $X_j = \{(t_i, (\vec{x}_i)_j) : t_i = i, 1 \leq i \leq N\}$ can be formed, where $(\vec{x}_i)_j$ refers to the j th coordinate of vector x_i . So, from a set of N vectors, M sets each containing N parameter-coordinate pairs are obtained. Each X_j is correctly viewed as a discrete set of samples from the $x_j(t)$, a scalar continuous function. Recovering each $x_j(t)$ from its samples X_j is a standard problem in interpolation. Assuming that the user has sampled densely enough so that there are no high-frequency oscillations for $\vec{x}(t)$ between samples, a good interpolating function for each $x_j(t)$ can be constructed using a cubic spline (Hou and Andrews, 1978). A cubic spline is a curve that passes through the sample points and has continuous first and second derivatives. Thereafter, the interpolated curve can be sampled as densely as needed to give a good visual representation of the path. A very sparse set of starting samples is shown on the left of Fig. 6, marked with x symbols. Using cubic spline interpolation and then sampling at a higher rate, the curve shown on the right of Fig. 6 is obtained. The user only has to specify a minimal set of points, as few as two, to generate a path. In fact, Path View includes an algorithm, described in Section 3.3, to allow the user to input unordered points, such as first specifying two endpoints and then specifying a middle point.

Projecting data samples along the path

After path $\vec{x}(t)$ is calculated from the sample points, proper data processing is still required to clearly display the path. The path has only length but no volume, making it nearly invisible. Thus, the path must attain some volume. A straightforward approach is to define a spherical neighborhood around the path. For each point $\vec{x}(t)$, let the neighborhood be

$$N(x(t), R) = \{\vec{x}(\tau) : \|\vec{x}(t) - \vec{x}(\tau)\| \leq R, \vec{x}(\tau) \in S\}, \quad (6)$$

where S is the data space and R is a variable radius of our selection. Our reason for making R variable is explained near the end of this section. The existence of the neighborhood greatly improves the visibility of the path when the path is buried in the interior of a data volume. Now, instead of a fine string, a bloated tube exists. Ordinary projection techniques still present problems of visibility. Consider the case depicted on the left of Fig. 7. If a plane slices the path in the middle as shown, only an oval will appear on that plane. The rest of the path

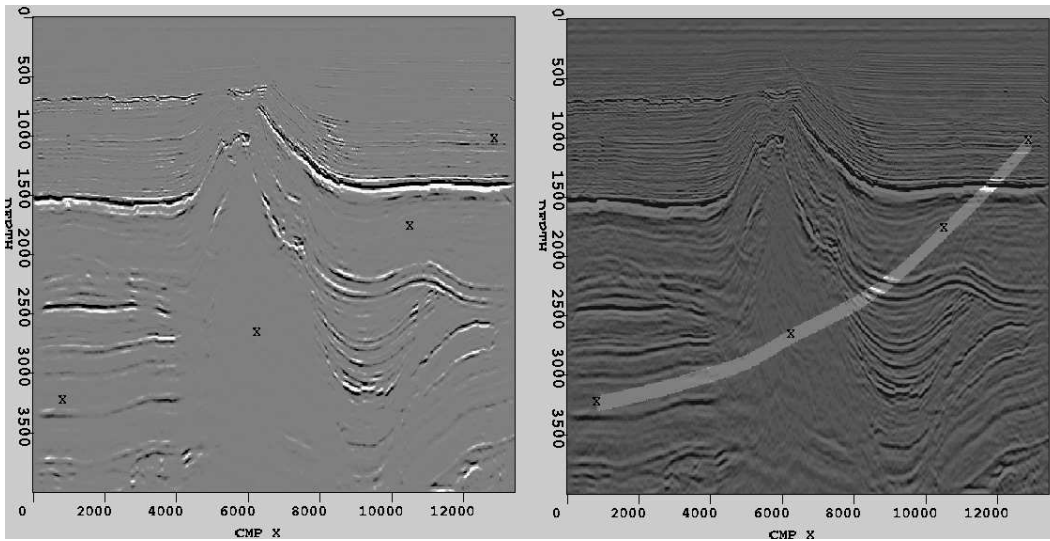


Figure 6: Sample points (left) and interpolated path (right). `bob3-chen-path-interp` [NR]

can only be revealed as we move the plane up and down but each time still is limited to just an oval. Instead, we want to see a longer portion of the path projected onto the plane, depicted on the right side of Fig. 7. The region of intersection should be most clearly visible, but the other regions should cast shadows onto the plane. Moreover, this effect should occur in all three viewing dimensions. A novel projection algorithm is developed to improve path

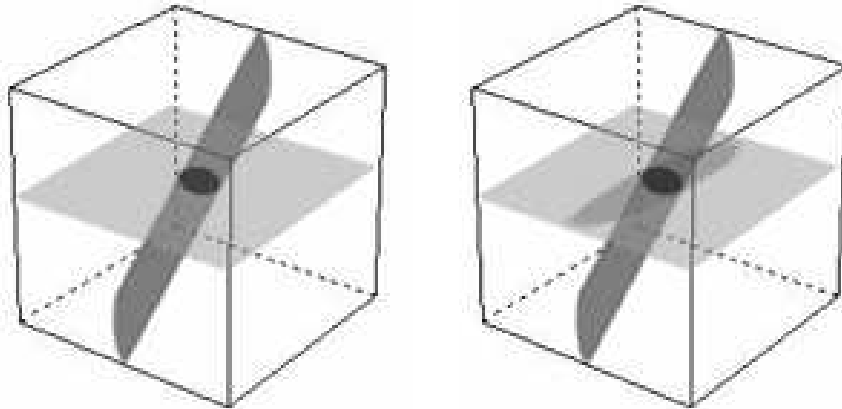


Figure 7: Projection without shadowing (left) and with shadowing (right). `bob3-chen-cube-model` [NR]

visibility. Three new volumes $p_i(\vec{x})$, $p_j(\vec{x})$, and $p_k(\vec{x})$ equal in size to the original volume $p(\vec{x})$ are created, corresponding to the three viewing directions \vec{i} , \vec{j} , and \vec{k} . We now show the algorithm for projection along \vec{i} , which produces a set of \vec{j} - \vec{k} slices, but the method is applied analogously along \vec{j} or \vec{k} . The steps of the algorithm are:

1. Volume $p_{i,low}(\vec{x})$ is filled with a dimmed version of $p(\vec{x})$, specifically $p_{i,low}(\vec{x}) = \alpha \cdot$

$p(\vec{x})$, where we have used $\alpha = 0.5$. This serves as a background for the paths.

- Every path point's influence needs to be extended beyond the point's immediate location. A useful metric is the orthogonal distance between a point P and a plane D, which is defined as the distance between P and the point in D closest to P. For a \vec{j} - \vec{k} slice located along \vec{i} at $x_{s,i}$ and a path point $\vec{x}_p = (x_{p,i}, x_{p,j}, x_{p,k}, \dots)$, the closest point on the slice to \vec{x}_p is $\vec{x}_s = (x_{s,i}, x_{p,j}, x_{p,k}, \dots)$. Thus, the orthogonal distance is $|x_{p,i} - x_{s,i}|$. If this distance is small enough, then a scaled version of the data value at \vec{x}_p in $p(\vec{x})$ should be shown at \vec{x}_s in $p_i(\vec{x})$. Specifically, for all \vec{x}_p on the path,

$$p_i(\vec{x}_s) = \max\{p_{i,fade}(\vec{x}_p), p_{i,low}(\vec{x}_s)\}, \quad (7)$$

$$p_{i,fade}(\vec{x}_p) = p(\vec{x}_p) \cdot e^{-(x_{p,i} - x_{s,i})^2 / (2\sigma_i^2)}, \quad (8)$$

$$\sigma_i = \frac{R_i}{10},$$

where R_i is the range of the data volume along direction \vec{i} . The value $p_{i,fade}(\vec{x}_p)$ defined by Eq. 8 is the data value on the path from the original volume attenuated by a Gaussian fading factor. The fading factor decreases as the path point and the \vec{j} - \vec{k} slice are separated farther. Eq. 7 sets a lower threshold for using the attenuated values.

- The previous step only covers projection at points \vec{x}_p on the path and should be repeated for every point \vec{x}_{pn} in the path neighborhood $N(\vec{x}_p, R)$ defined by Eq. 6. Redefine \vec{x}_s to be the closest point on the \vec{j} - \vec{k} slice to \vec{x}_{pn} : $\vec{x}_s = (x_{s,i}, x_{pn,j}, x_{pn,k}, \dots)$. For all x_p on the path and for all $\vec{x}_{pn} \in N(\vec{x}, R)$,

$$p_i(\vec{x}_s) = \max\{p_{i,fade}(\vec{x}_{pn}), p_{i,low}(\vec{x}_s)\}, \quad (9)$$

$$p_{i,fade}(\vec{x}_{pn}) = p(\vec{x}_{pn}) \cdot e^{-(x_{pn,i} - x_{s,i})^2 / (2\sigma_i^2)}, \quad (10)$$

The variable radius R plays an interesting role here. There is one problem inherent in projection: loss of distance information along the direction of projection. Far and near portions of the path should be displayed differently on a plane of projection. The Gaussian fading in Eq. 10 helps in this regard, but a stronger indicator is needed. Towards this end, R is made to be a monotonically increasing function of the distance between the point \vec{x}_{pn} and \vec{x}_s , the point on the \vec{j} - \vec{k} slice closest to \vec{x}_{pn} .

$$R = \alpha R_i \cdot \sqrt{\frac{|x_{pn,i} - x_{s,i}|}{R_i}}, \quad (11)$$

where $\alpha = 0.2$ and R_i is again the range along direction \vec{i} . Since $0 \leq |x_{pn,i} - x_{s,i}| \leq R_i$, the square root term is between 0 and 1 and increases as \vec{x}_{pn} and \vec{x}_s are farther away. Therefore, on any plane of projection, distant parts of a path will appear dimmer due to Eq. 10 and thicker due to Eq. 11, while near parts of a path will appear undimmed and thinner. Recovery of distance information along the direction of projection is achieved.

Fig. 8 shows projections onto slices in all three viewing dimensions for a path traversing a volume diagonally. As the projection radius grows and the projection intensity fades, the underlying path being projected is farther away from the current slice.

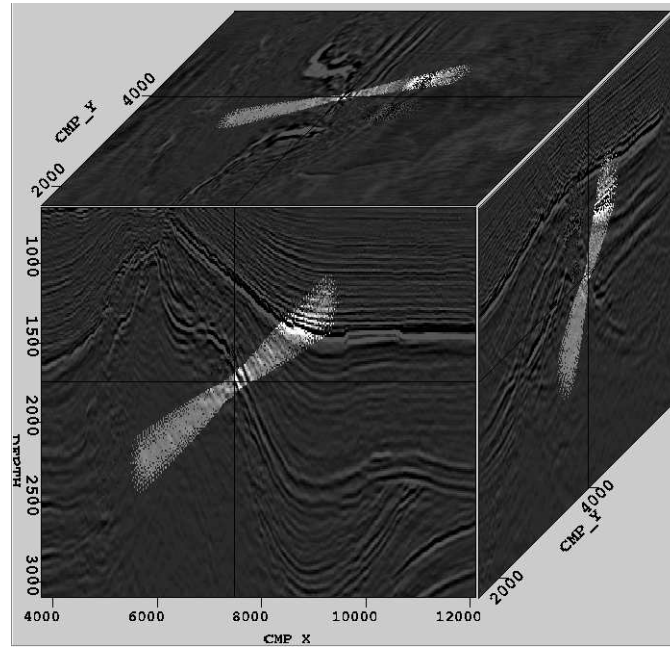


Figure 8: Projection of a diagonal path onto orthogonal planes. bob3-chen-pathview [NR]

Real-time generation of multiple paths

In an active Ricksep session, a user can interactively generate new paths using picks. Picks are symbols used to mark special locations in the data volume. We used picks of type “x” to represent the path samples shown in Fig. 6. Multiple picks of the same symbol define the starting representative samples of a path, which can then be fully generated using the methods described in Sections 3.1-3.2. A different path can be created using a pick of a different symbol. Since updates to the path can now be viewed in real time, when a pick is inserted, moved, or removed, the corresponding path changes immediately. Because picks can be inserted out of order (e.g. in Fig. 6, the middle two picks were inserted after the endpoint picks were inserted), a sorting algorithm is needed. The algorithm has the following steps:

1. For all picks of the same type, which are at N locations $\vec{x}_{p,1}, \vec{x}_{p,2}, \dots, \vec{x}_{p,N}$, calculate the inter-pick distance measure

$$D(n) = \sum_{m=1}^N \|\vec{x}_{p,n} - \vec{x}_{p,m}\|, \quad (12)$$

For $k = \operatorname{argmax}_n D(n)$, the pick at $\vec{x}_{p,k}$ is most distant from all the other picks. Let that pick be the first path sample.

2. Find the closest pick to the previous sample and designate it as the next sample in the sequence.

Table 1: Viewing commands tracked by History List.

Viewing command
Jumping to a point of interest in the volume
Moving gradually along an axis
Zooming in on a region of interest
Changing perspectives (e.g. from cube to flat faces)
Swapping two faces or two axes of the volume
Turning on the Velocity View or the Path View perspectives
Changing volume, background, or overlay colors

- Repeat the previous step, without reusing any pick already designated as a path sample. Stop when N path samples are available.

The sorting algorithm works well for all paths that are sampled at a sufficiently high rate, even those with large amounts of curvature.

HISTORY LIST: REVISION OF VIEWING SEQUENCES

History List maintains a sequential list of all viewing commands used during a Ricksep session. The latest command is appended to the end of the dynamically growing list. Table 1 lists the types of commands recorded. Undoing the latest command corresponds to moving back one step in list. Multiple commands can be undone by moving further back in the list. Similarly, redoing an undone command corresponds to moving forward one step in the list. Fig. 9 shows a series of viewing changes. First, the cursor is moved from the center to the lower left corner of the front face. Second, the viewing perspective is changed from a cube to flat faces. Third, a sub-region of the front face is magnified. At this point, there are four history elements in the list. Fourth, the previous magnification is undone by returning to the third history element in the list. Fifth, instead of reapplying the magnification, a change of perspective from flat faces to front face is requested. This replaces the previous fourth history element (magnification) by a new element (perspective change). Finally, the perspective is changed again, this time from front face to side face, leaving five history elements in the list. Transfer of viewing states between different Ricksep sessions is also possible. A user can generate a sequence of viewing states in one session, save the history information to a file, and load the saved viewing states into a different session. Precise restoration of viewing states is therefore realized. The history file can be externally edited into a new recipe for automated viewing of multiple data sets, giving rise to the possibility of batch processing. Additionally, History List is capable of animating a viewing sequence, thereby extending Ricksep's movie feature to cover all viewing changes in general.

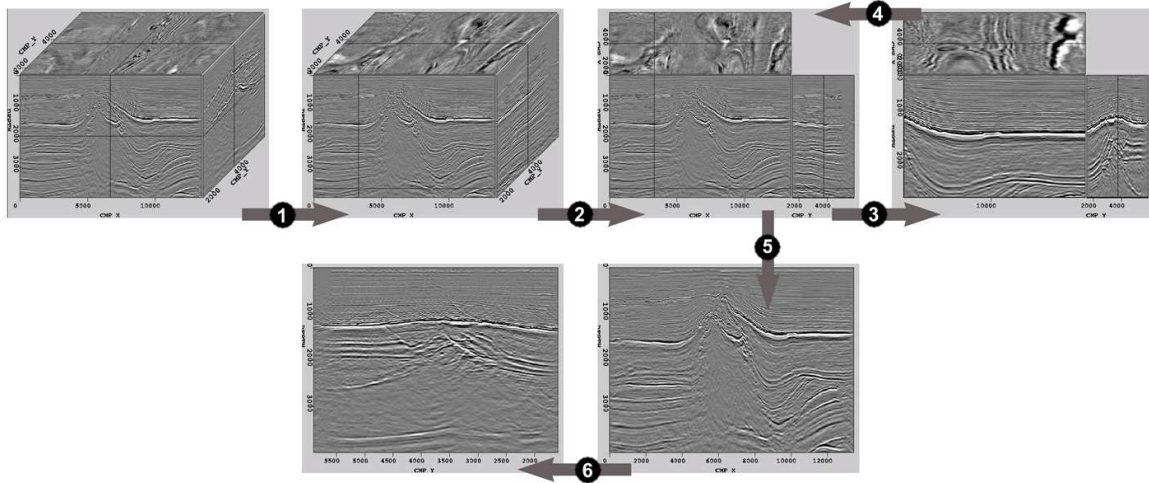


Figure 9: Sequence of viewing changes. bob3-chen-hist-seq [NR]

INTERFACE INSIDE RICKSEP

Controls for Velocity View

Velocity View can be invoked from the command line by adding one parameter (“velview”) to the usual Ricksep invocation for multiple volumes:

```
Rickmulti < seismic.T velocity.T [other arguments] velview=1
```

The order of specifying the input volumes is important, since Velocity View treats “seismic.T” as the high-frequency volume and “velocity.T” as the low-frequency volume. The synthesized volume appears in the place of the high-frequency volume. Moving the VEL VIEW slider on the control panel alters the resemblance of the result to a source volume, as described in Section 2.4. Shifting the slider far left, we obtain the high-frequency volume. Shifting the slider far right, we obtain the low-frequency volume. All of the controls for Velocity View are listed in the above table.

Controls for Path View

The main mechanism for interactively creating paths is the use of picks, as described in Section 3.3. To create new paths, follow these steps:

1. Turn on Path View (see first action listed in the above table).
2. Make a pick anywhere in the cube using the current symbol, say “x”.

Table 2: Controls for Velocity View.

Action	Control
Turn off Velocity View	[menu] VelView > Separate Cubes
Turn on Velocity View and synthesize by adjusting intensities	[menu] VelView > Synthesize by Adjusting Intensities
Turn on Velocity View and synthesize by copying differences	[menu] VelView > Synthesize by Copying Differences
Shift resemblance towards high-frequency volume	[slider] VEL VIEW > left
Shift resemblance towards low-frequency volume	[slider] VEL VIEW > right

3. Make another pick using symbol “x” and observe the formation of a path joining the two picks.
4. Make additional picks using symbol "x" or edit (move, delete, etc.) existing picks of type “x” to alter the “x” path.
5. To start a new path, switch the pick symbol (see second action listed in Table 5.2) to something different, say “y”.

Using different symbols, the user can create multiple paths through the cube. Additionally, the user has the option of using a pre-constructed path file to invoke Path View. This can be done at the command line by adding one parameter (“path”):

```
Ricksep < data.T [ other arguments ] path=pathFile.H
```

The path constructed from the file is associated with the pick symbol “F” (for file) and several representative “F” picks are made on the user’s behalf. This path can thereafter be treated as if it was manually generated from “F” picks. Once in Path View, the user can navigate across the volume as usual. To move along the current path itself, the user has available the controls listed in the table below.

Controls for History List

Viewing history is automatically tracked during each Ricksep session by History List. All controls for History List are listed in Table 4. The current position in the list is indicated in the message bar at the bottom of the Ricksep window. By saving the current history to a file, the user can restore the current viewing state in a different Ricksep session.

Table 3: Controls for Path View.

Action	Control
Turn on/off Path View	[menu] PathView > Path View On/Off [key] Shift + p
Select a different symbol	[menu] Pick > Change Pick Properties [key] Right mouse + c
Jump to start of path	[menu] PathView > Path Start
Jump to middle of path	[menu] PathView > Path Middle
Jump to end of path	[menu] PathView > Path End
Move forward along path	[key] +
Move backward along path	[key] -
Play movie along path in forward direction	[menu] PathView > Animate Path Forward
Play movie along path in backward direction	[menu] PathView > Animate Path Backward
Pause animation	[menu] PathView > Pause Animation [key] Control + p

Table 4: Controls for History List.

Action	Control
Undo last command	[menu] Navigate > Undo Move [key] Control + u
Redo last undone command	[menu] Navigate > Redo Move [key] Control + r
Undo to start of list	[menu] Navigate > Undo to Start
Redo to end of list	[menu] Navigate > Redo to End
Animate undo sequence to start of list	[menu] Navigate > Animate Undo to Start
Animate redo sequence to end of list	[menu] Navigate > Animate Redo to End
Pause animation	[menu] Navigate > Pause Animation [key] Control + p
Save history to file	[menu] Main > Write History File
Load history from file	[menu] Main > Load History File

CONCLUSION

The several new features added to Ricksep are effective aids in data-fusion of volumes, visualization of paths, and revision of viewing sequences. Velocity View provides an accurate, convenient method of visually correlating localized structures and large-scale patterns. Path View enables clear visualization of paths that were previously invisible. History List provides for automation of viewing commands by saving the viewing history in an editable list. Moreover, all three features have simple interfaces and work in conjunction with the viewer's previous capabilities. There is a good foundation on which to build future improvements to Ricksep. For Velocity View, the interface can be changed to allow multiple data-fusions to occur simultaneously. For example, if there are two pairs of high and low-frequency volumes, the two synthesized cubes can both be shown in the same window. For Path View, as a generalization of creating paths from picks, entire surfaces can be generated from picks. The mathematics of multi-dimensional interpolating is not significantly more difficult than the mathematics of one-dimensional interpolation, but implementing this functionality in Ricksep is nontrivial. For History List, other than the message bar informing users of the current position in the list, there is no other information about previous and following viewing states. A better interface would include a preview pane that includes information about viewing states before and after the current state.

REFERENCES

- Biondi, B. and J. van Trier, 1993, Visualization of multi-dimensional seismic data sets with CM-AVS: *SEP*, **79**, 1–12.
- Bloch, I. and H. Maitre, 1997, Data fusion in 2d and 3d image processing: SIBGRAPI (Brazilian Symposium on Computer Graphics and Image Processing), IEEE CS Press, Proceedings of SIBGRAPI, 127–135.
- Claerbout, J. F., 1981, On-line movies: *SEP*, **28**, 395–408.
- Clapp, R. G., M. L. Prucha, et al., 2001, *SEP manual*: *SEP*, **110**, 183–333.
- Clapp, R. G., 1995, *SEP AVS user guide*: *SEP*, **84**, 395–408.
- Clapp, R. G., 2001, Ricksep: interactive display of multi-dimensional data: *SEP*, **110**, 163–172.
- Hou, H. S. and H. C. Andrews, 1978, Cubic splines for image interpolation and digital filtering: *IEEE Transactions on Acoustics, Speech, and Signal Processing*, **26**, 508–517.
- Mora, C. B., R. G. Clapp, and B. Biondi, 1996, Visualization of irregularly sampled seismic data with AVS: *SEP*, **93**, 75–86.
- Ottolini, R., 1990, Seismic movies on the Xview graphics system: *SEP*, **65**, 301–305.

