Stanford Exploration Project, Report 75, November 17, 1997, pages 1–25

Monte Carlo automatic velocity picks

David E. Lumley[1]

keywords: *velocity, random, modeling*

ABSTRACT

I develop a method for making automatic velocity picks from velocity semblance scans, using a Monte Carlo nonlinear fitting technique. Given a single velocity semblance scan, random interval velocity models are generated by performing many Monte Carlo random walks. For each trial random interval velocity model, the corresponding rms velocity is calculated, and the semblance is integrated along the trial rms velocity path through the velocity scan. After several random walks, one interval velocity model, and its associated rms velocity function, is retained as the global best fit to the velocity scan which maximizes semblance. The result is a nonlinearly optimal rms velocity function, and an associated geologically reasonable interval velocity model, at each surface CMP location. I tested the method on 300 marine CMP gathers, and the Monte Carlo picks gave reasonable rms and interval velocity models which enhanced the stacked and prestack migrated images.

INTRODUCTION

The process of making manual velocity picks from velocity semblance scans can be an extremely time consuming bottleneck in routine preprocessing of seismic reflection data, and may even represent a powerful deterrent to performing unaliased spatial velocity analyses for 3-D surveys. A robust method of making automatic velocity picks is therefore of practical interest.

Velocity scans (or spectra), as discussed by Taner and Koehler (1985) for example, are computed by stacking or migrating seismic reflection data along hyperbolic trajectories h parameterized by moveout velocity $\bar{v}_s$ and zero-offset traveltime τ. The summation procedure over a range of $\bar{v}_s$ and τ values at a given CMP location produces a single velocity scan. Semblance S, one of many possible coherency criteria, is defined as the ratio of stacked energy to input energy:

[1]**email:** not available

$$S(\bar{v}_s, \tau) = \frac{\left(\sum_h \mathrm{Data}\left[h(\bar{v}_s, \tau)\right]\right)^2}{\sum_h \left(\mathrm{Data}\left[h(\bar{v}_s, \tau)\right]\right)^2} \tag{1}$$

Figure **??**a (left) is an example of a CMP gather, and Figure **??**b is the corresponding velocity semblance scan. Velocity picking is the process of defining an optimal stacking velocity trajectory, $\bar{v}_s(\tau)$, along semblance peaks in each velocity scan $S(\bar{v}_s, \tau)$. Dix (1955) and Claerbout (1976) showed that stacking velocity is exactly equivalent to rms velocity $\bar{v}_r$ in a 1-D earth at a fixed (not necessarily vertical) propagation angle. For somewhat vertical incident-angle arrivals in a somewhat 1-D earth, the relationship becomes approximate:

$$\bar{v}_s^2(\tau) \sim \bar{v}_r^2(\tau) = \frac{1}{\tau} \int_0^\tau v_i^2(t)\, dt \tag{2}$$

where v_i is the interval velocity model, explicitly a function of the vertical traveltime τ, but implicitly a function of vertical depth z.

Conventionally, the stacking velocities $\bar{v}_s$ are picked manually from velocity scans by human interpretation at a workstation. This can be extremely time consuming and tedious, and may require several man-months for a typical 3-D exploration survey. Hence, it is of practical interest to develop a robust and fast automatic velocity picking computer algorithm. The automatic velocity picks can then be subsequently evaluated by an interpreter for quality control and refined as necessary, saving much time in the overall procedure.

I present a method to simultaneously obtain an optimal fit to the velocity semblance scan and a geologically reasonable interval velocity model, using a nonlinear Monte Carlo optimization. There are two key criteria to my approach: (1) the $\bar{v}_s$ pick must maximize the semblance integral along the rms path in the scan, and (2) the $\bar{v}_s$ pick must correspond to a physically realizable and geologically reasonable v_i interval velocity model. In a related work, Zhang (1991) uses constrained linear and nonlinear optimization to perform automatic velocity picking. Zhang's algorithm gives an accurate fit to the semblance peaks, however, the resulting $\bar{v}_s$ picks can yield unreasonable and even unphysical (imaginary) interval velocities after Dix inversion. Toldi (1989) uses a linearized conjugate gradient method with v_i constraints to find an interval velocity model which maximizes semblance. Toldi's results give physical interval velocities, but the fitting algorithm may easily become trapped in a local minimum since it is a linearization about a initial starting model of the full nonlinear optimization problem. Finally, Rothman (1985) proposed a method for generating random $v_i(x, z)$ earth models, calculating the resulting non-hyperbolic moveout at each CMP gather, and performing a nonlinear optimization of the velocity stack power by simulated annealing. My approach is not as ambitious as that of Rothman, but based on criteria (1) and (2), may be viewed as a desirable extension to the methods of Toldi and Zhang.

This paper proceeds as follows. First I discuss a method to find a global parametric $\bar{v}_r$ and v_i fit by a controlled parameter search. Then I discuss the Monte Carlo random walk perturbations to the parametric $\bar{v}_r$ solution, along with issues of convergence and constraints. Finally, I present the results of the application of the algorithm to 300 marine CMP gathers.

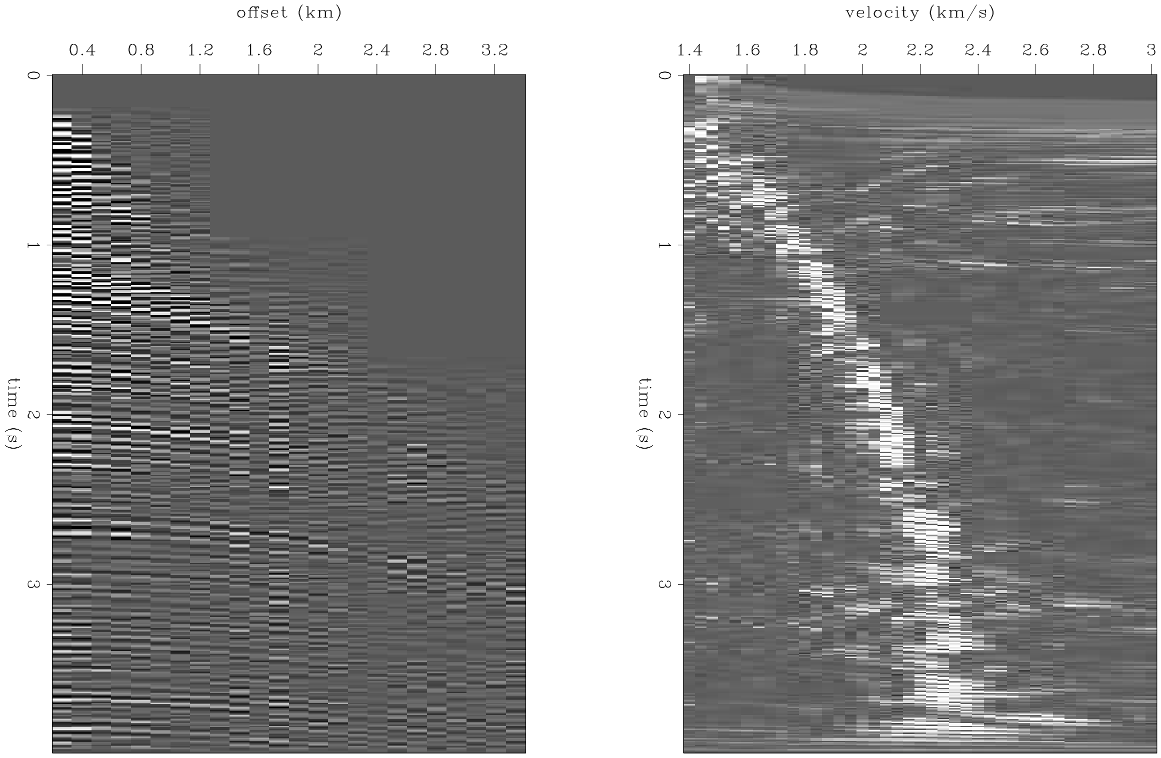


Figure 1: A representative CMP gather (a) and its corresponding velocity semblance scan (b). david1-cmpscan [ER]

PARAMETRIC OPTIMIZATION

The algorithm proceeds in two basic steps. The first is a controlled search for the best fit of a parametric velocity function to the semblance scan. The second part is the Monte Carlo random walk maximization of semblance based on the initial parametric fit and interval velocity model constraints.

Parametric velocity fitting

It is desirable to start the Monte Carlo process at an initial estimate of interval velocity that is in the right "ball park" relative to the semblance information. Such an initial estimate can save time in minimizing the number of Monte Carlo random walks needed to obtain satisfactory convergence, and provides a basis for v_i and

$\bar{v}_r$ velocity constraints. To derive an initial velocity estimate, I use the following parametric form of rms velocity:

$$\bar{v}_s \sim v_o + \alpha t^\beta \ . \tag{3}$$

This equation is a general power-law fit to many typical classes of rms velocity functions, ranging from constant velocity, to constant gradient, to polynomial fits.

The parameter v_o is the $\bar{v}_r$ value at time zero (although the t value could be shifted if required), and for marine data may have a range of 1.4 – 1.6 km/s. The parameter α is related to the rms velocity gradient with traveltime. In fact, for the special case that $\beta = 1$, α is precisely the gradient. It has a typical range of 0.1 – 1.0 km/s/s. The parameter β is related to the rate of increase or decrease in velocity with time, and is "loosely" related to the "curvature" of the velocity function with time. For a constant velocity medium, $\alpha = 0$, and for a constant gradient medium, $\beta = 1$. Typical marine sediment velocity profiles have an exponent of $\beta \sim 0.5$.

Given the parametric form (3), an analytic expression for the interval velocity can be easily obtained after a little algebra, by assuming (3) is an rms average:

$$v_i \sim \sqrt{v_o^2 + 2v_o\alpha(1+\beta)t^\beta + (1+2\beta)\alpha^2 t^{2\beta}} \ . \tag{4}$$

In practice, I make a coarse discretization of the parameter space $\{v_o, \alpha, \beta\}$, perform an exhaustive search over that domain for the rms curve (3) that, when integrated against the velocity semblance scan, maximizes total semblance. This process gives an optimal three parameter set $\{v_o^*, \alpha^*, \beta^*\}$, which, when substituted into (4) yields the initial estimate of interval velocity. Figure **??**a demonstrates the initial parametric fit on a typical semblance scan. The curve that lies closest to the semblance peaks is the $\bar{v}_s$ curve given by (3) evaluated at the $\{v_o^*, \alpha^*, \beta^*\}$ which maximize semblance. The smooth curve of higher average velocity is the parametric interval velocity curve v_i given by (4) evaluated at $\{v_o^*, \alpha^*, \beta^*\}$.

The implementation of the parametric velocity search is diagrammed below:

```
INITIAL PARAMETRIC VELOCITY ESTIMATE
====================================


        * Choose parametric form:  Vrms = Vo + a * t**b

                - Vo = Vrms(t=0)
                - a ~ vertical velocity gradient
                - b ~ velocity "curvature"

                        1.4 <  Vo  < 1.6
                        0.0 <   a  < 1.0
```

```
                0.0 <   b  < 2.0

* Exhaustive search over finite parameter domain {Vo,a,b}

* Retain Vrms*(Vo*,a*,b*) that maximizes semblance

* Vrms* --> Vint*  is analytic


* RESULT:

        - a Vrms* parametric fit that is globally "optimal"
        - a Vint* model that is physically reasonable
        - a starting point for MC random walks and constraints
```

The parametric fit in Figure **??**a seems remarkably reasonable. The amount of velocity misfit is subtle compared to the Monte Carlo fit shown in Figure **??**b. This can be seen by examining a section stacked with parametric $\bar{v}_s$ velocities. Figure **??** shows a section stacked with a 1-D velocity function, which is an optimal Monte Carlo fit at midpoint 7 km. Figure **??** has been stacked with the smoothed 2-D parametric velocity field. Figure **??** is the optimal stack using the smoothed 2-D Monte Carlo velocity picks, to be discussed in the next section.

The 1-D velocity stack in Figure **??** is coherent at the west where the single Monte Carlo velocity analysis was done, but fades to the east due to significant lateral velocity variation that is not accounted for by the 1-D stacking velocity. The 2-D parametric velocity stack is very coherent across the entire line in Figure **??**. Finally, the 2-D Monte Carlo velocity stack is the best, as shown in Figure **??**. The Monte Carlo stack is superior in its resolution of shallow fault structure and fault diffractions, as well as within the fault block at 12.5 km and 2 seconds depth. However, for a "quick and dirty velocity field," the parametric search can be very fast and effective.

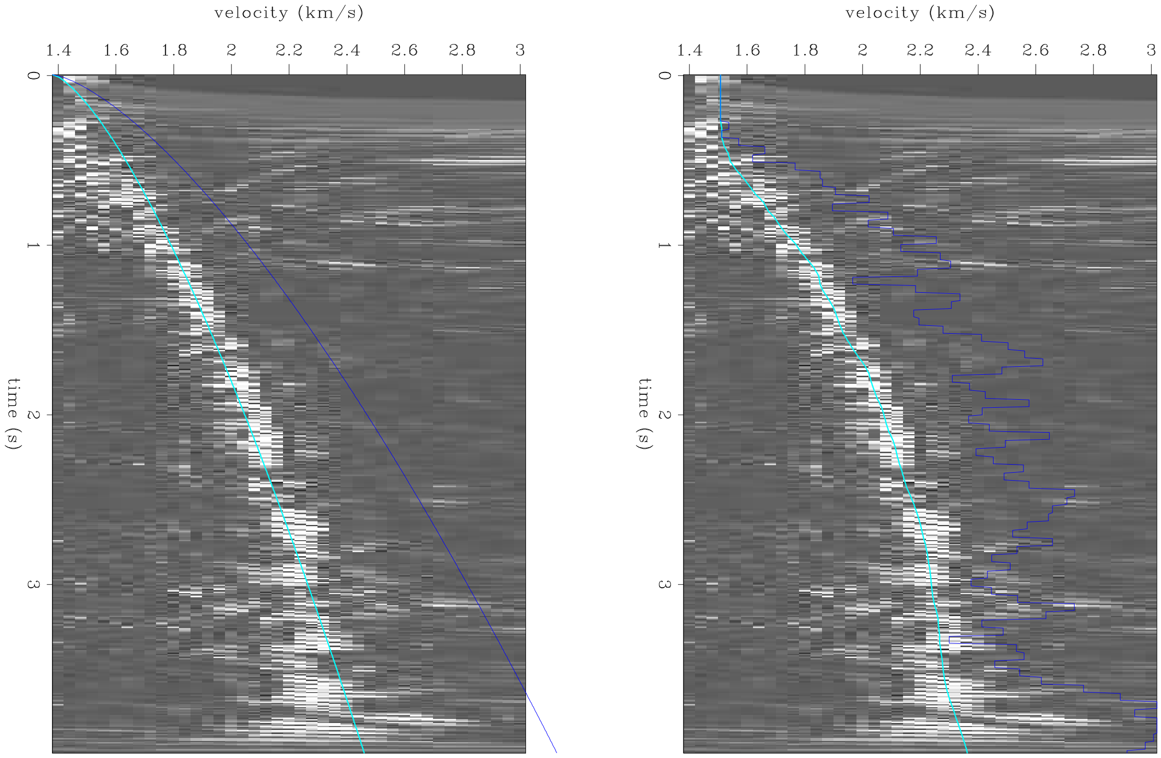


Figure 2: (a) Initial parametric velocity fit, (b) Monte Carlo nonlinear velocity fit. $\boxed{\text{david1-cmpfit}}$ [ER]

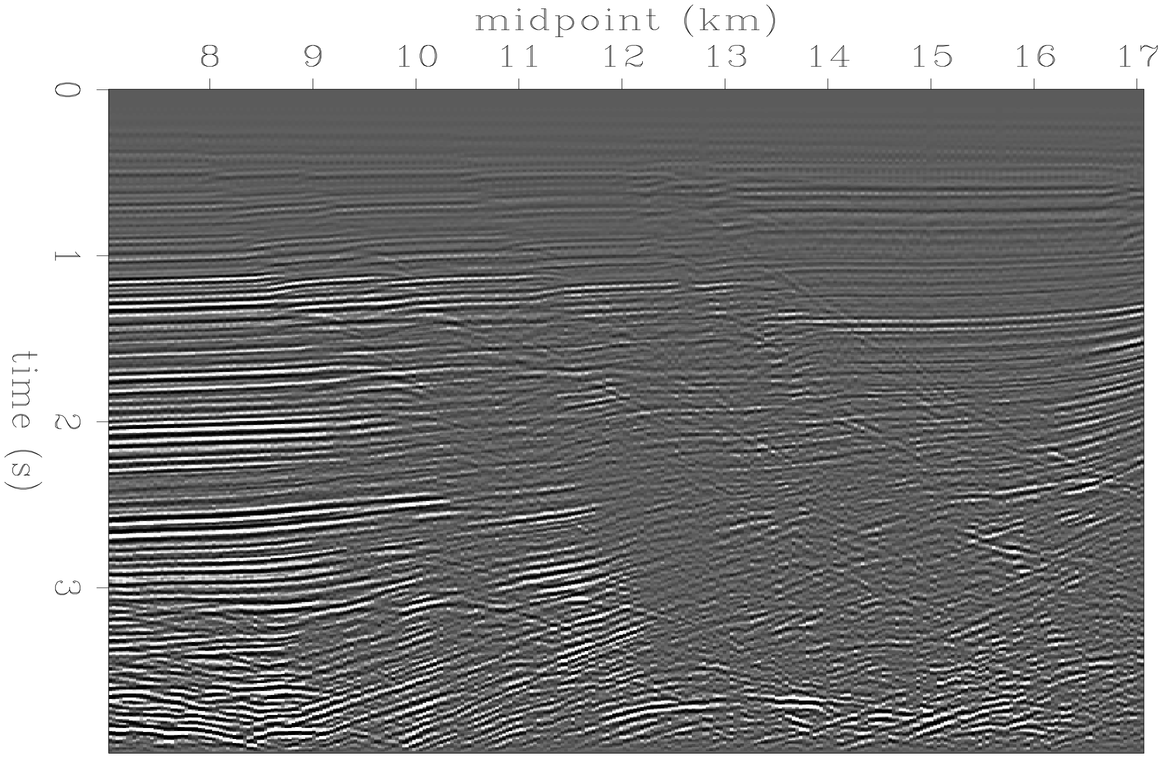


Figure 3: Stacked section using a 1-D velocity function, obtained by an optimal Monte Carlo fit at midpoint 7 km. david1-stack1 [ER]

MONTE CARLO OPTIMIZATION

The Monte Carlo fitting algorithm is a two-step process. The outer loop ranges over a sequence of N_w Monte Carlo "random walks." The inner loop consists of a sequence of N_s random "footsteps" for a given random walk.

Random walks

On a single random walk, the set of N_s footsteps explore the model space in the vicinity of the initial interval velocity model by making random perturbations to that initial model. For a single footstep, all the interval velocities v_i are perturbed simultaneously. I use a random uniform distribution with a standard deviation of about 10% of the interval velocity value at that depth (time). The next step is performed by returning to the initial model and repeating the process. This means that all steps *for a single walk* are within about 20% of the initial interval velocity model. I tried letting the steps concatenate such that each step takes you farther away from the initial model, but this seemed to be a waste of time without some directional guidance in the step direction. I use multiple walks to achieve this goal.

At the end of N_s steps, the best fitting interval velocity model is retained as the

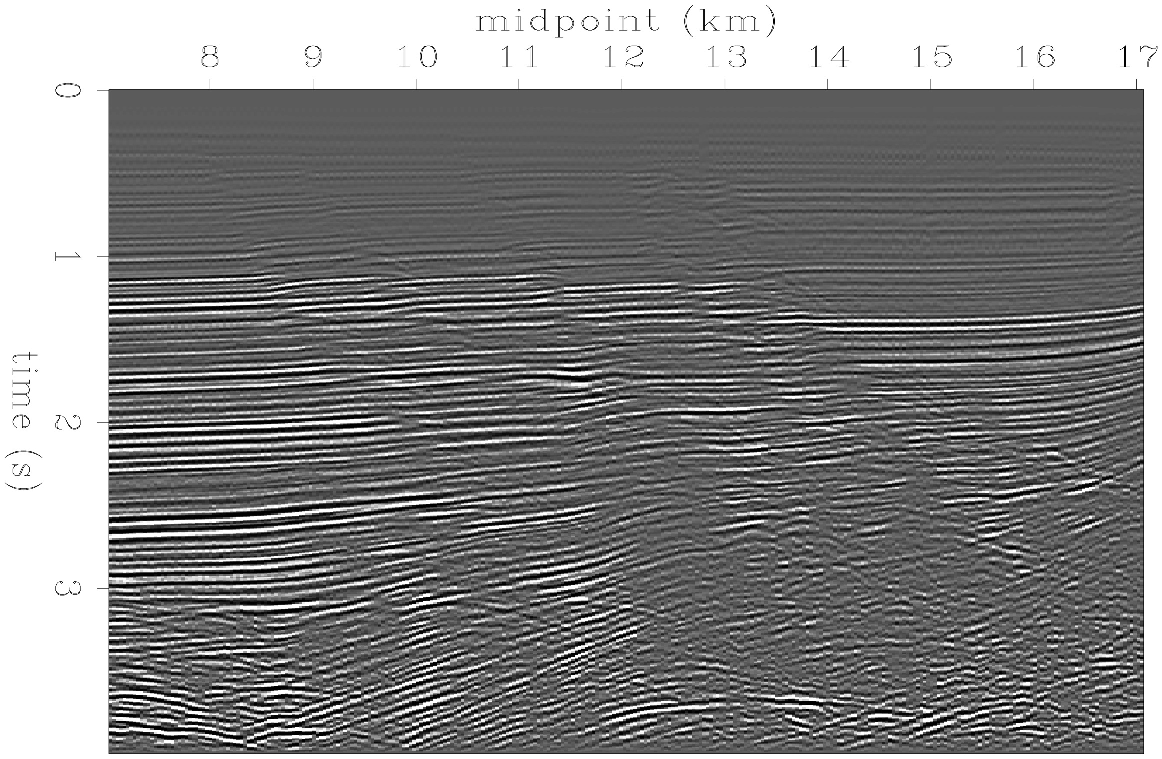


Figure 4: Stacked section using 2-D smoothed parametric velocities, obtained by the controlled parameter search method. $\boxed{\text{david1-gstack}}$ [ER]

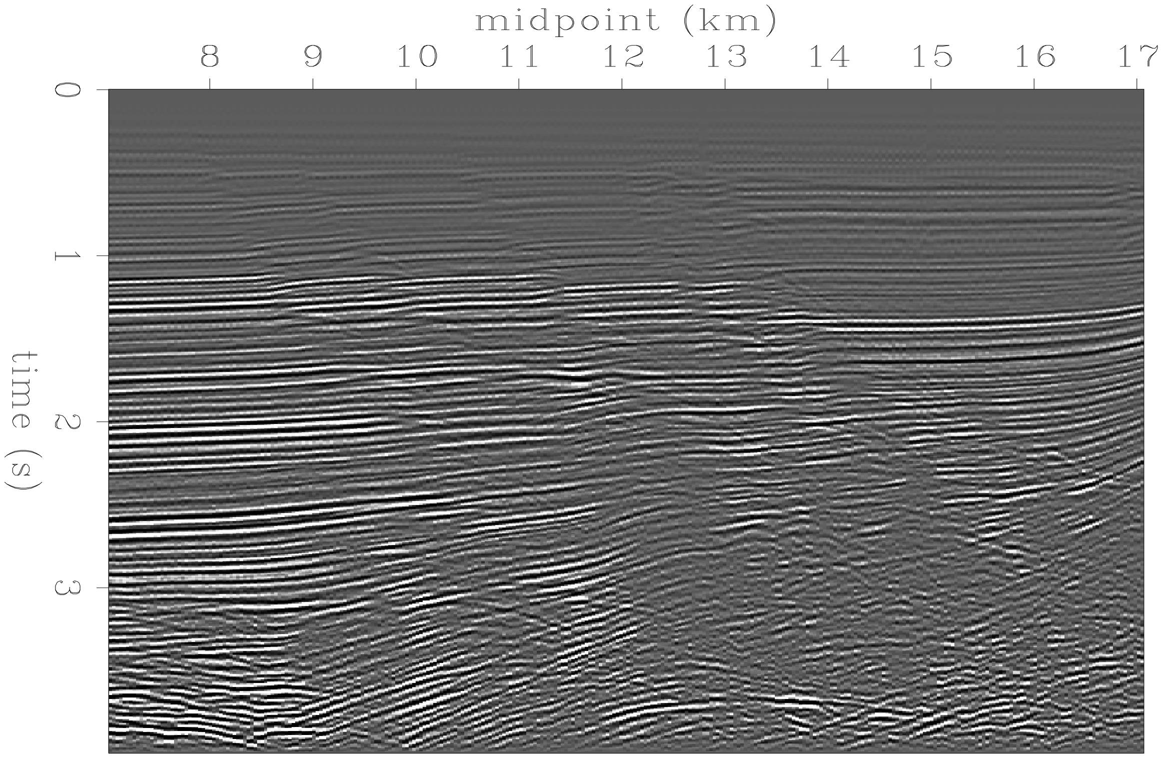


Figure 5: Optimal stacked section using 2-D smoothed Monte Carlo velocities, obtained by the nonlinear random walk search method. david1-mcstack [ER]

initial model for the next random walk. Then the footstep procedure is repeated until N_w walks have been completed, or some measure of "convergence" has been achieved. The final best fit model at the end all random walks is the Monte Carlo optimal interval velocity v_i^*, and its associated rms curve is the optimal stacking velocity trajectory $\bar{v}_s^*$.

Velocity Constraints

At each footstep I impose some constraints on the trial random interval velocity model before integrating along the semblance scan. I impose a global minimum and maximum interval velocity value. For the marine data example, I used $v_{min} = 1.55$ km/s (water velocity), and $v_{max} = 3.0$ km/s. The standard deviation σ_v of the uniform distribution from which I draw random velocity perturbations, implicitly imposes a velocity corridor type of constraint on the local minimum and maximum allowed velocities, *for a single walk*. Since the initial model to be perturbed changes from walk to walk, these velocity corridors change also. Only the v_{min}, v_{max} values are global bounds for all trial models. I also allow the surface velocity v_o (with possible error tolerance) to be specified to anchor the interval velocities to some reality. This is because, at shallow times, there is usually little or no semblance information available to constrain a velocity fit, due to the refraction mute pattern.

I allow the user to specify the number of interval velocity transmission layers per second of traveltime, which may be perturbed to fit the semblance scan. For this data example, choosing the number of layers to be on the order of 10-20 per second gave reasonable results. Too few layers will not allow the velocity curve enough flexibility to fit the semblance peaks, and too many layers will tend to give unnecessary high-frequency velocity variations which average out to a smoother 10-20 layer response. Note that the number of layers does not affect the speed of my algorithm, since I assume that each time sample may contain a velocity variation when I do the rms and semblance integrations.

I also impose an important continuity constraint for each step of a walk. The parameter v_z is a bound on the maximum velocity gradient at each point in depth (time). For the data example, I set this value to be 200 m/s, which allows enough oscillation to fit scattered semblance peaks, but not enough to be geologically unreasonable in appearance. Furthermore, I weight the maximum allowable velocity gradient by the local value of (unit normalized) semblance at that vertical traveltime. This has the desirable effect that maximum gradient velocity perturbations are allowed at strong semblance peaks (i.e., where strong reflectors obviously must exist), and nearly zero gradient velocity perturbations (smooth) at semblance gaps (where strong reflectors apparently do not exist). These continuity constraints impose some geological sanity on the acceptable class of random models, and serve as a prefilter to the test phase in which the acceptable trial random models are integrated against the semblance scan and measured for optimality.

It is important to realize that the interval velocities inverted from *traveltime* data cannot be too discontinuous, because there is no resolution in such velocity analysis data to support thin layers of large velocity contrast (e.g., thin carbonate streaks). Only an *average* interval velocity can be resolved, and this average must be reasonably continuous by the very definition of an "average." This is the underlying physical basis for the continuity constraints I impose. In stark contrast, the interval velocities inverted from reflection *amplitude* data must be highly discontinuous, since it is these very impedance discontinuities which generate the reflection amplitudes. But the latter is not applicable here.

Convergence

Without a notion of convergence, the Monte Carlo technique could flounder around all over model space forever, wasting valuable CPU time. In order to overcome this possibility, I have defined a measure of "convergence." I know the maximum semblance value at each time sample (which will occur at some variable velocity) by doing a `max()` sift through the velocity semblance scan, looping over velocity at each fixed time. Therefore I know the peak possible integrated semblance value by summing all the temporal maxima. My definition of convergence occurs when three consecutive optimal random models differ in integrated semblance by less than 0.1% of each other, normalized to the peak semblance value. I use this convergence in both the footstep and the random walk loops. Finally, I have global convergence criterion on the maximum number of random walks to be performed; if the best fitting semblance integral value has not changed in the last 10 random *walks*, then I assume I am at the global maximum attainable within the physical constraints listed above. These convergence parameters are specified as input to the algorithm, but the values listed above seemed to give favorable results in less than 10 CPU seconds per velocity scan on an IBM RS/6000 Model 530.

The following is a flowchart description of the Monte Carlo fitting procedure:

```
MONTE CARLO VELOCITY
====================

        * Start with optimal parametric fit  Vest* (interval)

        * Loop over Nw random "walks"

           * Loop over Ns random "steps"

                *  Perturb Vest(t) for ALL z simultaneously = Vmc (int)

                *  Check Vint constraints

                *  Calculate corresponding Vmc (rms)

                *  If Fit(Vmc) > Fit(Vest*);  then Vest* = Vmc

                *  Check convergence logic

           *  Check convergence logic

        * End Loops

        * RESULT:

                - a Vrms* NONLINEAR fit that is globally "optimal"
                - a Vint* model that is physically reasonable
                - Semblance MISFIT error estimates (use for QC?)
```

A MARINE DATA EXAMPLE

Marine data and preprocessing

I tested the Monte Carlo automatic velocity picking algorithm on 300 CMPs of Gulf Coast data, donated to Stanford Exploration Project by Western Geophysical. Out of historical interest, this is the data set that Hale (1984) used in his classic paper on dip moveout analysis (DMO).

Each CMP gather is 24-fold and of 4 second record length. The midpoint spacing is 0.033 km, and the offset spacing is 0.134 km. I bandpassed the 4 ms data with a sixth-order Butterworth 10–40 Hz filter and decimated to 8 ms. I applied a mute at water velocity with a 250 ms ramp to complete the preprocessing prior to velocity analysis.

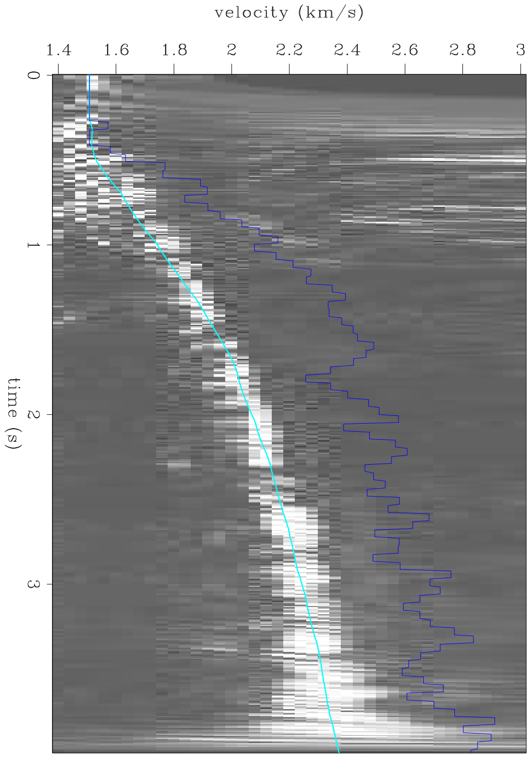
I ran my version of an NMO stacking velocity analysis on each of the 300 preprocessed CMP gathers (7200 traces total). I chose a stacking velocity range of 1.4–3.0 km/s in 0.040 km/s increments. I perform offset weighting, $\cos^2\theta$ weighting, and a time dependent divergence correction, during the computation of the velocity semblance spectra. Next, I do a little post-processing of the scans by doing a time-gated "AGC" using peak semblance, as a function of velocity, within a moving time window of 0.5 seconds, followed by a lateral moving average of 10 CMPs ($\pm$ 165 m). This has the effect of normalizing and enhancing small semblance peaks as a function of time, and boosting semblance S/N by about 3:1. This careful preprocessing and velocity analysis has resulted in an excellent quality of the velocity semblance scans prior to picking, as is visible in Figure (**??**) as compared to the single fold CMP velocity scan in Figure (**??**).

Monte Carlo velocity picks

Next I ran the Monte Carlo automatic velocity fitting algorithm on all 300 velocity semblance scans. The total CPU time requires an average of 10 CPU seconds per velocity scan on an IBM RS/6000 Model 530 with the following search parameters. For the initial parametric interval velocity estimate, I searched over the surface velocity range v_o = 1.4–1.6 km/s in 0.5 km/s increments, the velocity gradient parameter range α = 0.1–0.5 km/s/s in 0.1 km/s/s increments, and the time exponent range β = 0.1–1.0 in 0.1 unit increments.

For the subsequent Monte Carlo search, I used the following parameters: 20 interval velocity layers per second, a maximum adjacent vertical velocity increment of 0.2 km/s, a random walk velocity search corridor bound of 0.8 and 1.2 times the best current interval velocity model, a surface velocity of 1.525 km/s constant from 0.0–0.2 seconds, a global minimum and maximum interval velocity of 1.525 and 3.0 km/s, a standard deviation of 10% of the best current v_i value for the random velocity perturbations drawn from a uniform distribution, a convergence error tolerance of 0.1% of the peak maximum semblance integral value, a "stale" convergence criterion of 10 unchanged consecutive random walks, and a maximum of 100 random walks each consisting of 100 random steps, resulting in a maximum of 10,000 random trial interval velocity models. These parameters were discussed in the previous section explaining the Monte Carlo method.

Figures **??**–**??** show velocity scans and their Monte Carlo picks, as selected at every 30 CMPs (1.0 km) along the line from the entire survey velocity cube. One can see that the quality and shape of the semblance peak trajectories varies along the line, and that the Monte Carlo picks are very good in most circumstances. Note that to the east, the dipping events are caused by the uprise of the flanks of a salt dome (which peaks just east of 17 km at about 2 seconds depth). Some interval velocity picks show a deep high velocity zone which may be due to the presence of salt and/or steep dip (remember these are NMO stacking, not migration, velocities).



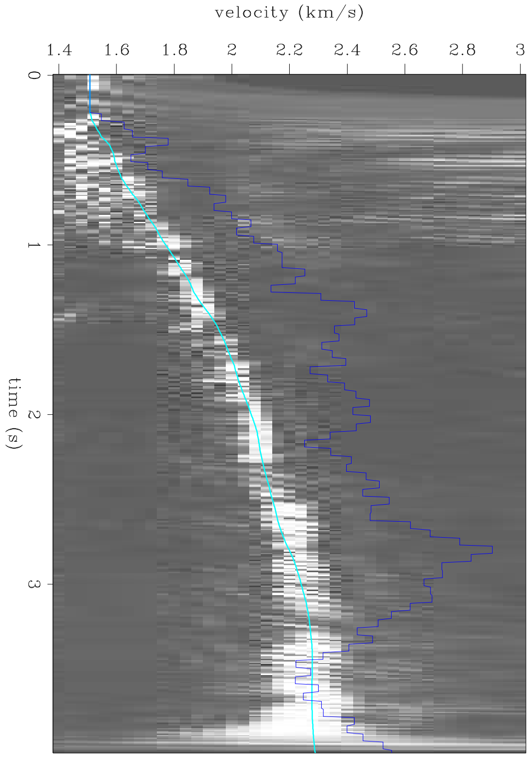


Figure 6: Velocity semblance scans at (a) 7.0 km, and (b) 8.0 km, overlain with Monte Carlo automatic rms and interval velocity picks. david1-fit12 [ER]

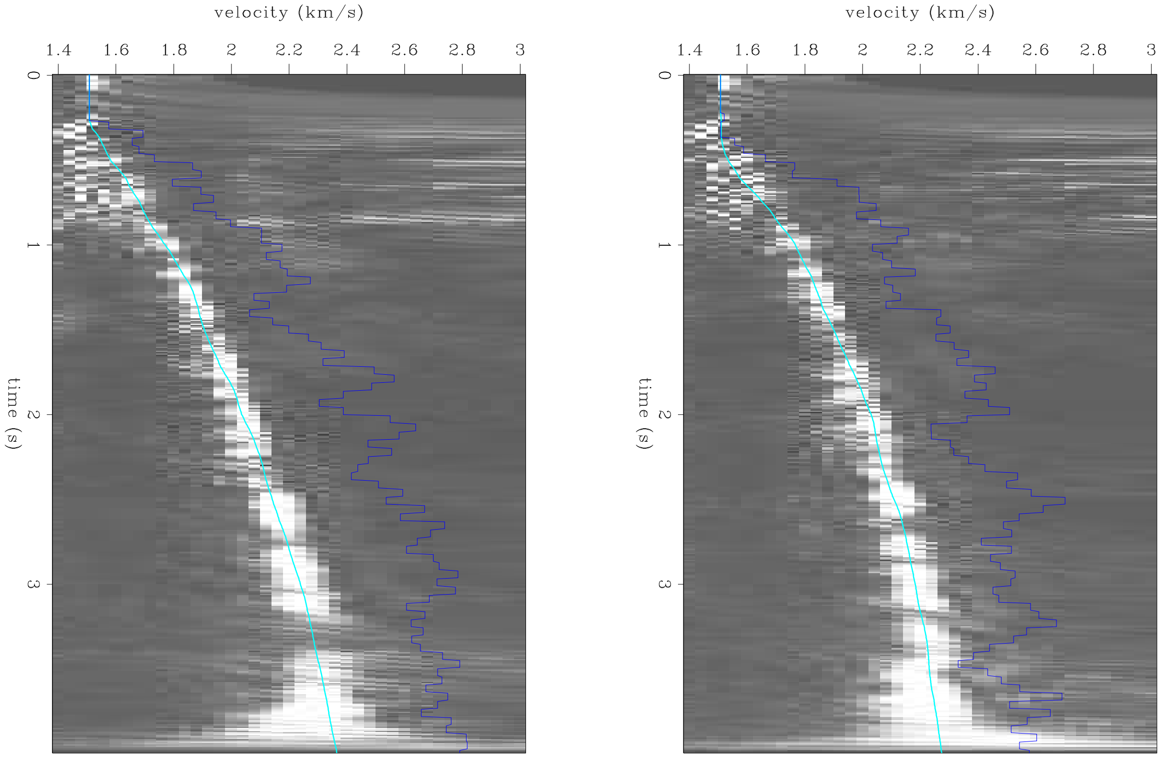


Figure 7: Velocity semblance scans at (a) 9.0 km, and (b) 10.0 km, overlain with Monte Carlo automatic rms and interval velocity picks. david1-fit34 [ER]

Figure **??** shows the unsmoothed contour plot of all 300 Monte Carlo rms velocity picks in the upper panel, and the smoothed rms velocities (a triangle of 330 m half-width) in the lower panel. It is evident that the automatic picks exhibit a large degree of lateral consistency, even though they are not constrained to do so. This gives some reassurance that the Monte Carlo algorithm is somewhat robust, at least for these 300 CMP gathers. The smoothed velocities in Figure **??**b are used for the subsequent Kirchhoff prestack time migration. Note that the velocity varies laterally, and sags in velocity toward the east where the shallow low velocity sediments have been downthrown by the slump faults activated by the salt dome.

Figure **??** shows the unsmoothed raster plot of all 300 Monte Carlo interval velocity picks, and the smoothed v_i contours below. In general, the interval velocities are much more noisy than the rms velocities, as is to be expected. However, there is a definite low frequency trend that parallels the rms trends and the downthrown low velocities. This seems rather promising that there really is some low frequency information in the interval velocities. For example, Figure **??**b could be used directly as a first estimate of depth migration velocity, after a vertical time-to-depth conversion. In fact, it would be interesting to use the smoothed interval velocity model to convert the time migration to depth.

Figure **??** is a useful byproduct of the Monte Carlo velocity fitting procedure.

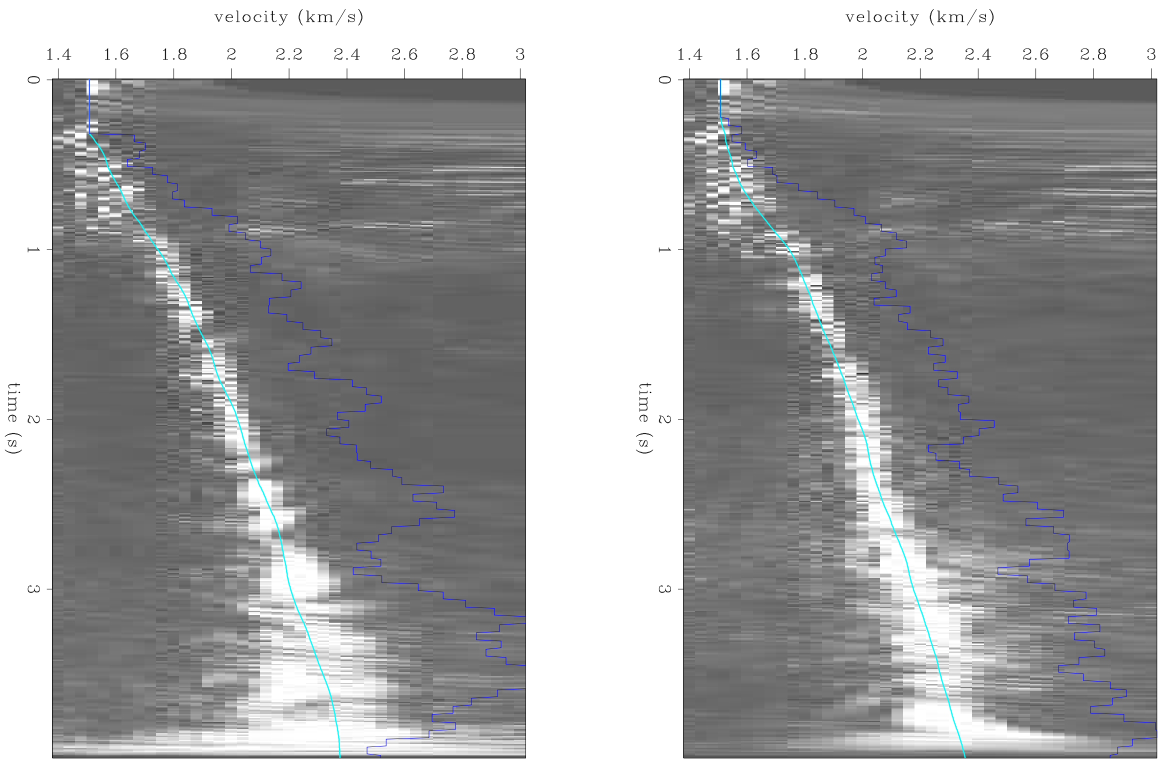


Figure 8: Velocity semblance scans at (a) 11.0 km, and (b) 12.0 km, overlain with Monte Carlo automatic rms and interval velocity picks. david1-fit56 [ER]

It is a plot of the relative misfit error of the velocity picks. Since the maximum possible integrated semblance value S_{max} can be calculated from each velocity scan, the relative misfit error E measure is also available:

$$E = 1 - S_{mc}/S_{max} \ , \tag{5}$$

where S_{mc} is the optimal integrated semblance value that the Monte Carlo algorithm converged to for a given velocity scan. Figure **??** shows that the fits are fairly consistent at about 20% relative misfit error, which is rather good. Figure **??** could also be extremely useful as a Quality Control tool, since bad velocity picks will cause the error to spike in the plot. For example, one can quickly decide that the velocity picks around midpoints at 11 km are not as favorable as the contiguous CMP velocity picks. For an interpreter, a quick look at this plot can tell you (a) how good the overall picks are quantitatively, and (b) which areas might need to be edited and repicked manually to produce the final velocity field estimate.

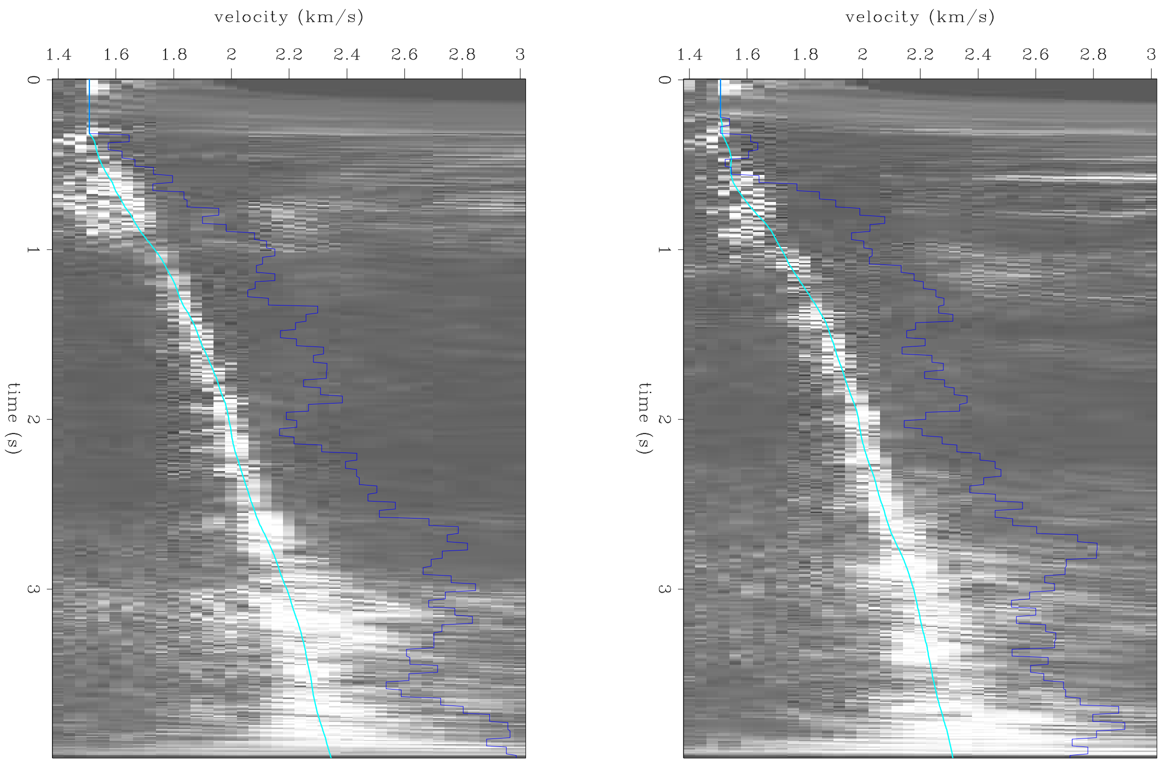


Figure 9: Velocity semblance scans at (a) 13.0 km, and (b) 14.0 km, overlain with Monte Carlo automatic rms and interval velocity picks. david1-fit78 [ER]

Migrations

Figure **??** shows a Kirchhoff prestack time migration with the 1-D velocity field described before. Although the previous 1-D stack was not very coherent to the east, the prestack migration has done an amazing job. This seems to indicate that time migration is *less* sensitive to velocity error than NMO stacking. Figure **??** is the Kirchhoff prestack time migration using the optimal 2-D Monte Carlo velocity field of Figure **??**b. The faulted blocks are very well defined from the base right up to the surface. Figure **??** is an overlay of the smoothed Monte Carlo interval velocity field on the prestack time migration of Figure **??**. Above 3 seconds there is a fairly good low frequency correlation between migrated structure and interval velocity. However, below 3 seconds the velocities are rather dubious in reliability. There is definite bias due to structural dip, as seen at 12 km, 3.5 seconds, and at the easternmost edge of the section, kilometer 17. It would be interesting to compute migration velocity spectra for this data set, and repeat the process of making the automatic Monte Carlo velocity picks and remigrating the data.

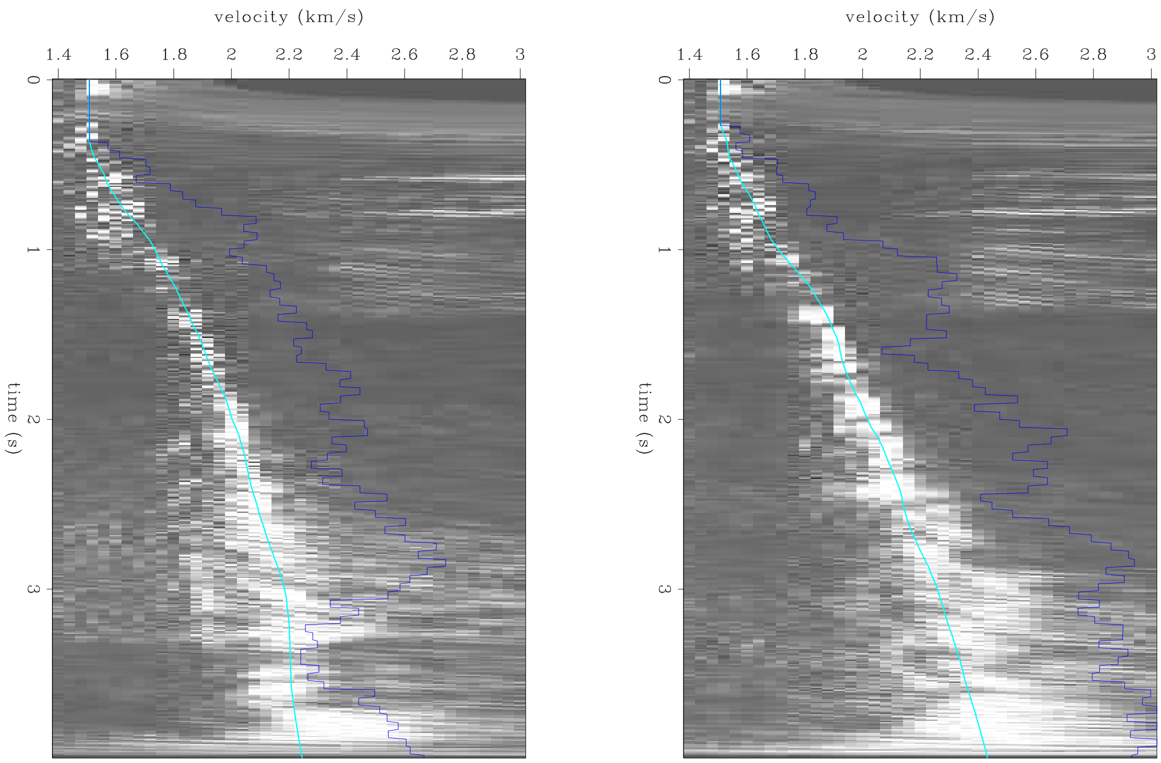
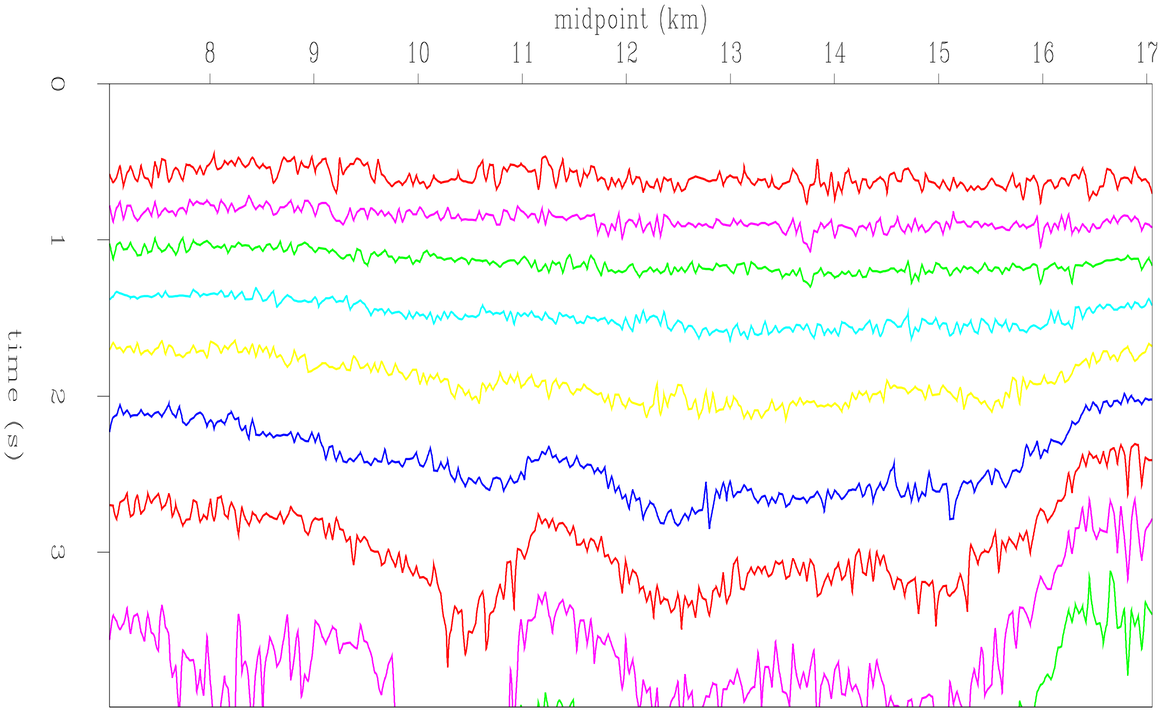


Figure 10: Velocity semblance scans at (a) 15.0 km, and (b) 16.0 km, overlain with Monte Carlo automatic rms and interval velocity picks. $\boxed{\text{david1-fit90}}$ [ER]

CONCLUSION

I have developed a method to perform automatic velocity picks using a nonlinear Monte Carlo random search method. I solve the problem by generating random trial interval velocity models, computing their associated rms velocity paths, and integrating along the rms paths through velocity semblance scans. The interval velocity that maximizes semblance is retained as the best fit. I use various constraints to ensure that the interval velocity model is geologically reasonable.

I tested the method on 300 marine CMP gathers from the Gulf Of Mexico. The test was successful in that a reasonable 2-D rms and interval velocity model was picked that enhanced the stack and prestack migrated sections. The relative misfit error along the line averaged 20%, and can be used as a QC tool to quickly determine regions that may require manual editing and repicking of velocities, although none was required here. The Monte Carlo algorithm requires about 10 CPU seconds per semblance scan to make the optimal pick, based on run times on an IBM RS/6000 Model 530.



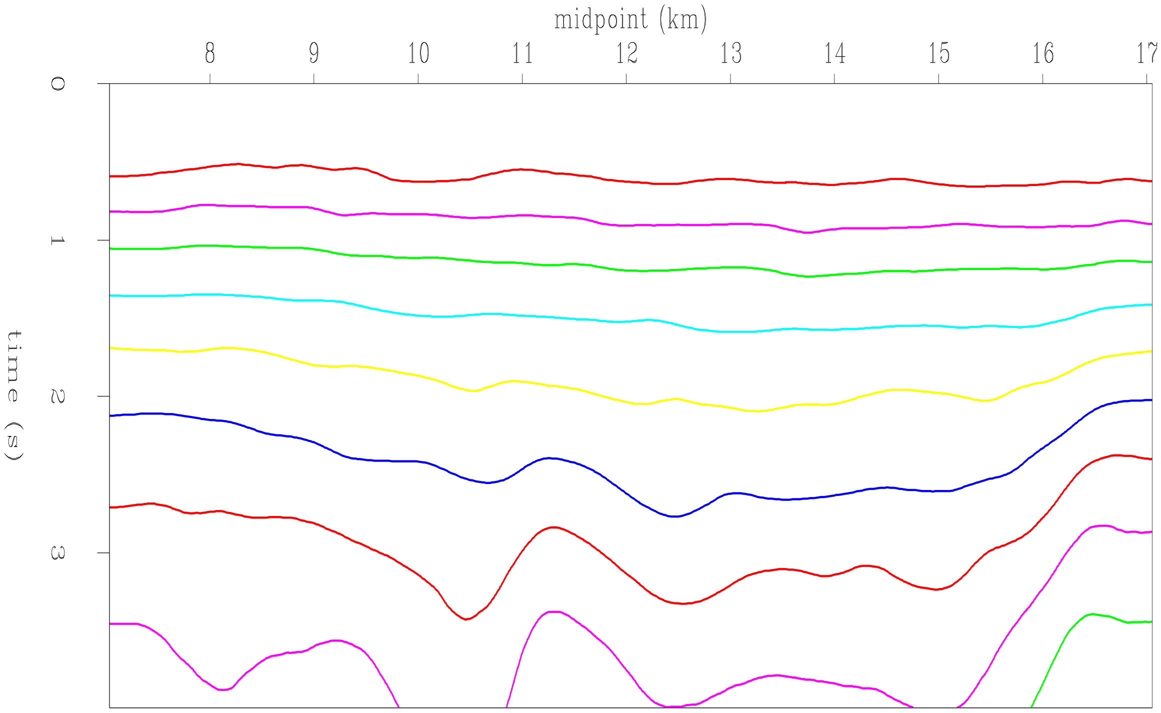
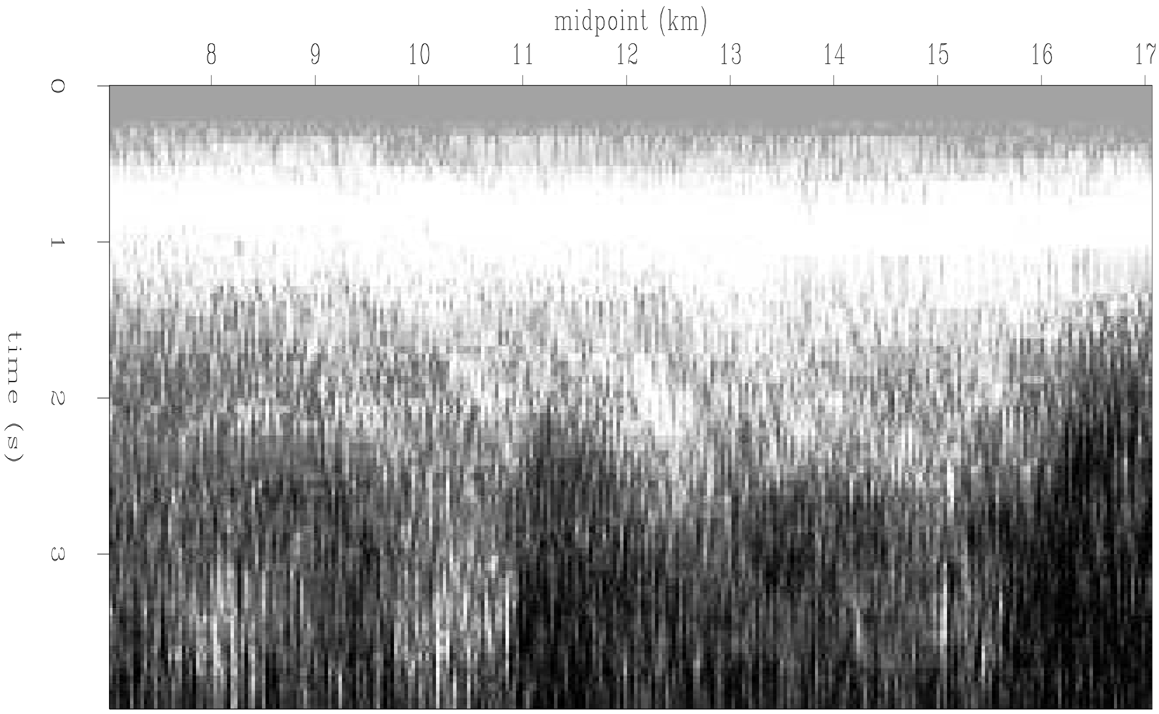


Figure 11: (a) Upper panel: unsmoothed Monte Carlo rms velocity picks, (b) Lower panel: laterally smoothed Monte Carlo rms velocities using a triangle of 330 m half-width. $\boxed{\text{david1-mcvels}}$ [ER]



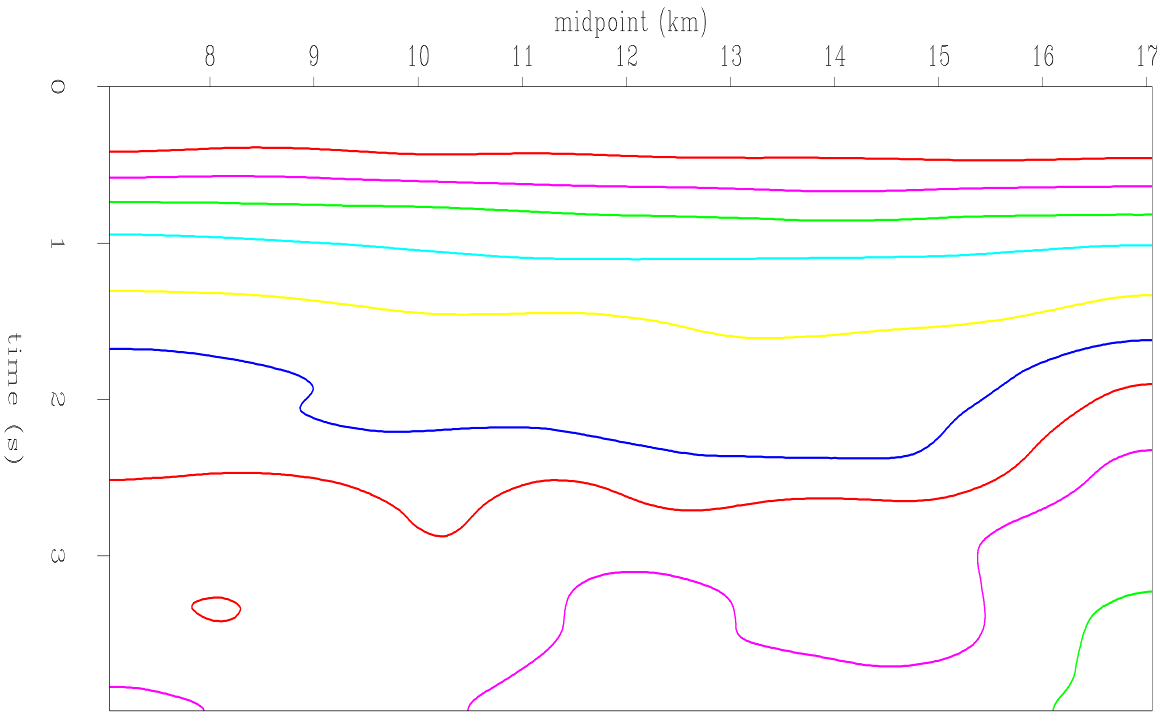


Figure 12: (a) Upper panel: unsmoothed Monte Carlo interval velocity picks, (b) Lower panel: laterally smoothed Monte Carlo interval velocities using a triangle of 330 m half-width. david1-mcvints [ER]

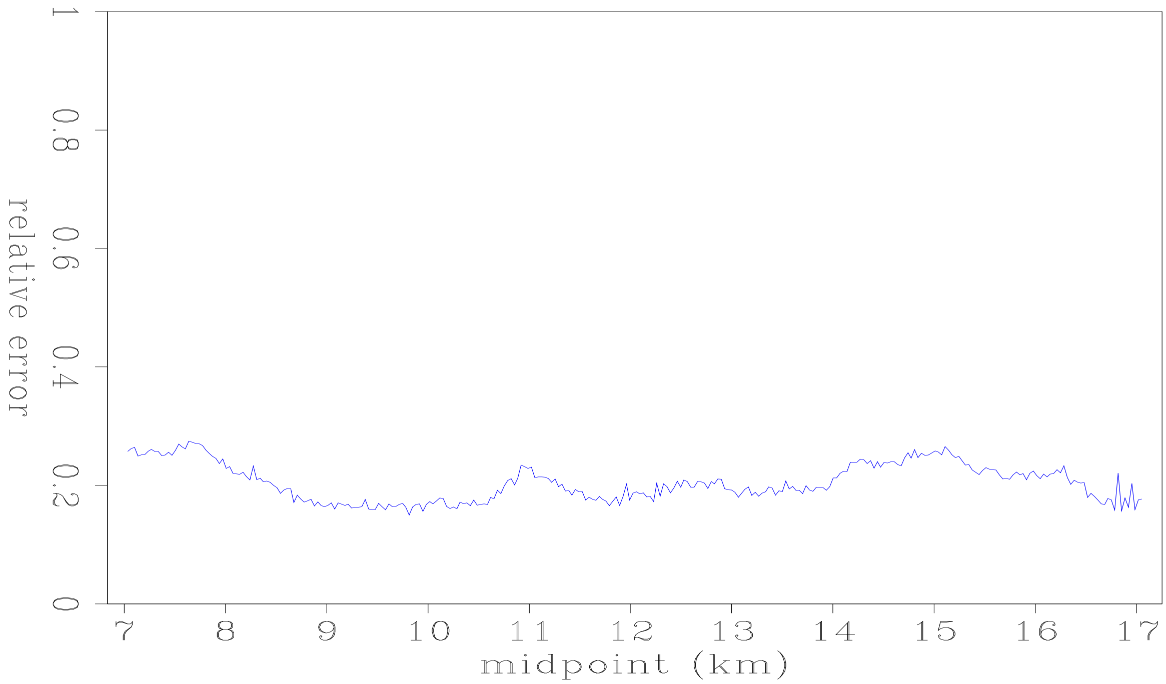


Figure 13: A plot of the Monte Carlo velocity fit error as a function of surface midpoint coordinate. A perfect fit would maximize semblance and represent 0.00 fit error on the plot. A very poor fit would achieve a low semblance integral value and represent a value near 1.00 in the fit error. david1-mcverr [ER]

ACKNOWLEDGMENTS

I would like to thank Jon Claerbout for several productive discussions we had regarding velocity analysis and velocity picking. I also appreciate the efforts of Jim Black (IBM), Jon Claerbout and Lin Zhang for loading this data set from tape to disk this summer; my Monte Carlo project would not have been as fun without an interesting data set to play with.

REFERENCES

Claerbout, J. F., 1976, A brief derivation of the Dix Theorem: Stanford Exploration Project SEP-8, 166–168.

Dix, C. H., 1955, Seismic velocities from surface measurements: Geophysics, **20**, 68–86.

Hale, D., 1984, Dip-moveout by Fourier Transform: Geophysics, **49**, 741–757.

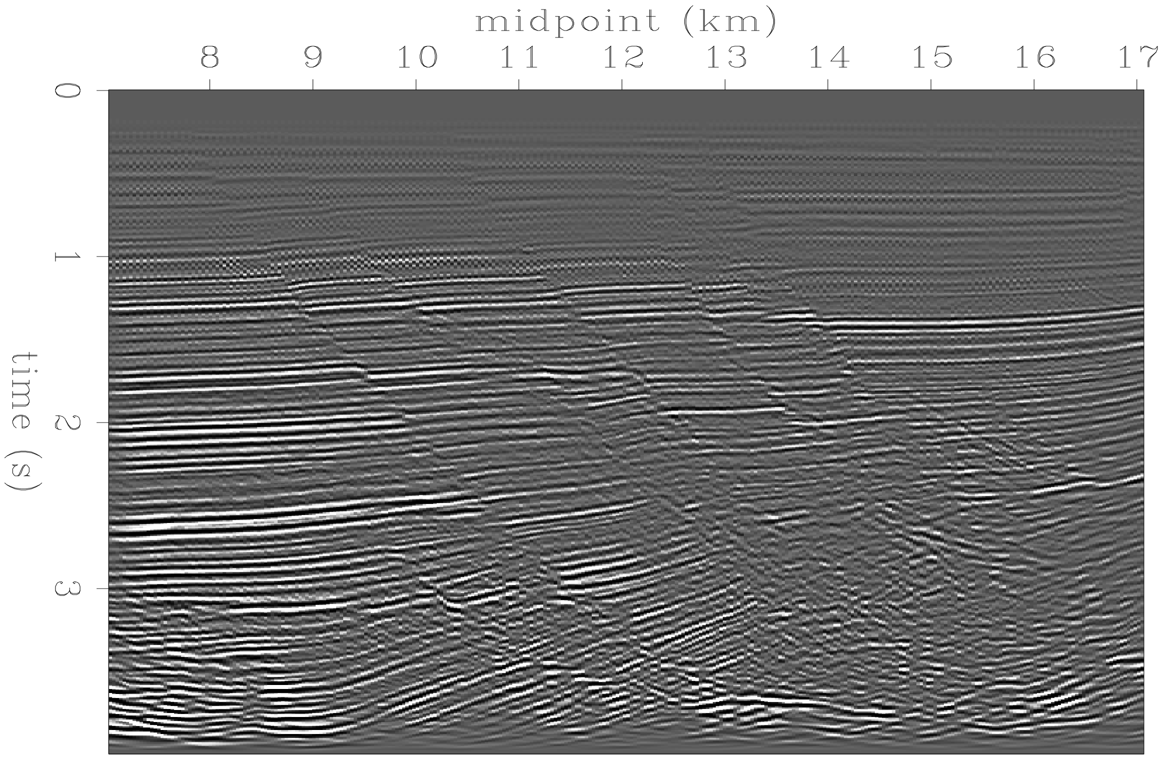


Figure 14: Kirchhoff prestack time migration using a 1-D velocity corresponding to the single Monte Carlo fit at midpoint 7 km. david1-mig1 [ER]

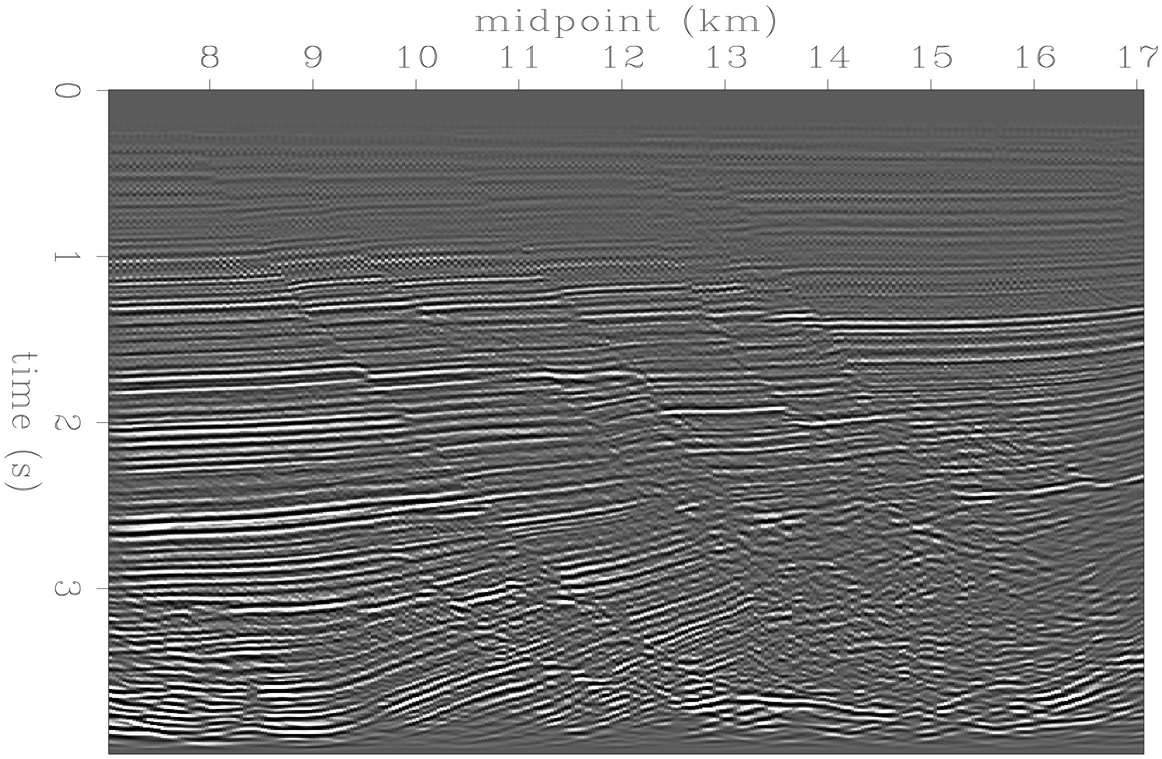


Figure 15: Kirchhoff prestack time migration using the optimal 2-D Monte Carlo smoothed velocity field. david1-mcmig [ER]

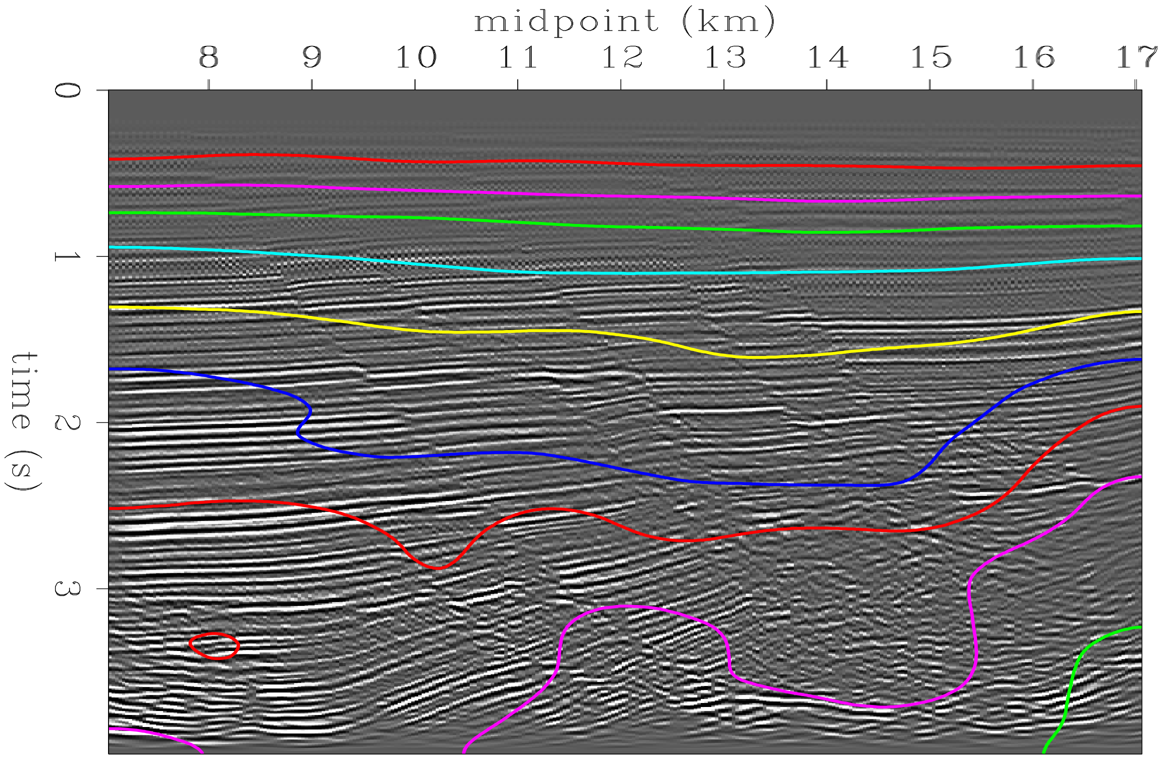


Figure 16: Overlay of the smoothed Monte Carlo interval velocities upon best Kirchhoff prestack time migration. david1-mcvintmig [ER]

Rothman, D. H., 1985, Velocity estimation by simulated annealing: problems and prospects: Stanford Exploration Project SEP-44, 47–54.

Taner, M. T., and Koehler, F., 1985, Velocity spectra – digital computer derivation and applications of velocity functions: Geophysics, **50**, 1928–1950.

Toldi, J. L., 1989, Velocity analysis without picking: Geophysics, **54**, 191–199.

Zhang, L., 1991, Automatic picking and its applications: Stanford Exploration Project SEP-70, 275–292.