Stanford Exploration Project, Report 70, December 18, 1997, pages 1–249

Interval velocity estimation with event picking

Carlos A. Cunha Filho[1]

keywords: *velocity, algorithm, modeling, estimation*

ABSTRACT

Most interval-velocity estimation methods are formulated as an iterative optimization scheme which requires an initial model to start the inversion. In addition, ray-tracing or approximate equations are used to relate the model to the objective function. For the horizontally-layered earth, it is possible to directly relate the interval velocity between two interfaces to the difference in shapes of the two reflected wavefronts corresponding to those interfaces. I have devised a velocity estimation method based on this relation, using beam-stacks to define a probability density function for the horizontal slowness, and an automatic picking algorithm to define the reflection events. The results show that good resolution is achieved when the method is applied to synthetic and real data.

INTRODUCTION

For an isotropic, plane-layered earth, a multiples-free, common-shot reflection seismogram can be interpreted as the result of a dynamically equivalent experiment. This experiment would consist of a different earth model and an unusual recording system. The alternative earth model would have doubled spatial dimensions and null reflectivities, with the angle-dependent one-way transmissivities of the interfaces replaced by the two-way angle-dependent transmissivities. The recording system would consist of receivers located above each interface, with an interconnection between receivers that share a common horizontal coordinate (Figure **??**). In addition, each receiver would have an "angular sensitivity" (radiation pattern) equal to the angle-dependent reflectivity of the interface below it. By analogy to the "exploding reflector model," we could name it the "listening reflector model".

With this model in mind, it is possible to devise an interval velocity scheme in which we upward continue the receivers with different velocities, and compare the results with the original data. The correct velocity of a given interval will be the one that minimizes the phase difference between the reflectors at the top of the interval before and after the continuation. Once the average interval velocities were

[1]**email:** not available

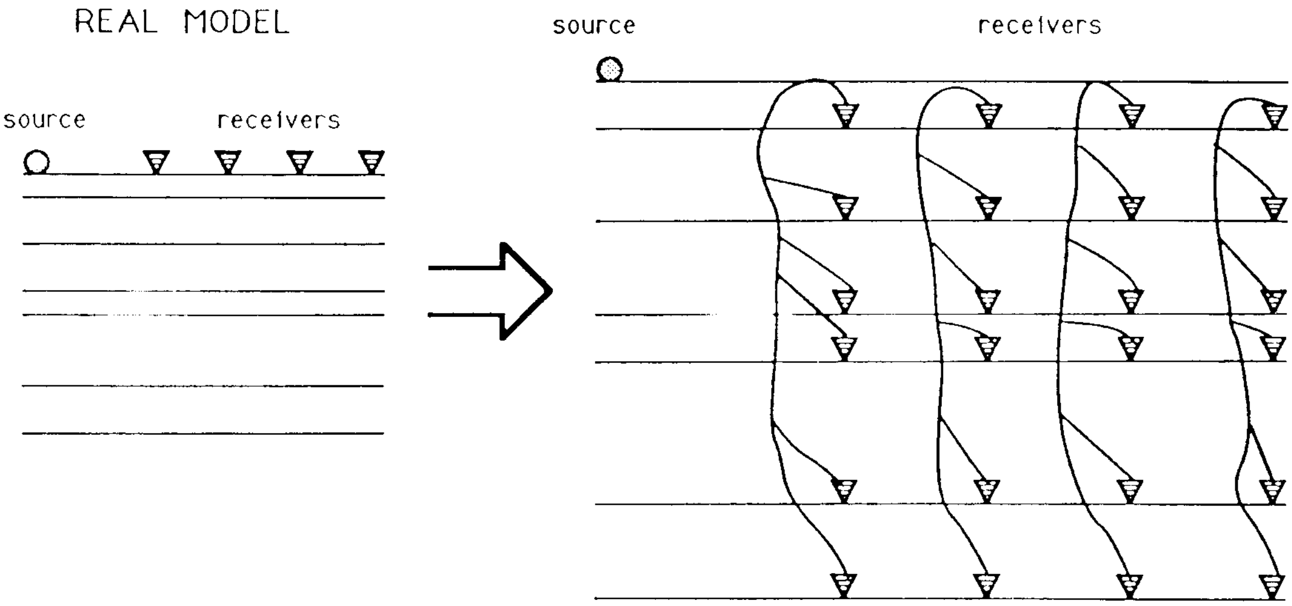

Figure 1: A reflection seismic experiment in a plane-layered earth and its equivalent multiple-free "listening reflector model." carlos3-lstmod [NR]

determined, one could refine the estimation (solve the dynamics) using the amplitude ratios before and after the continuation.

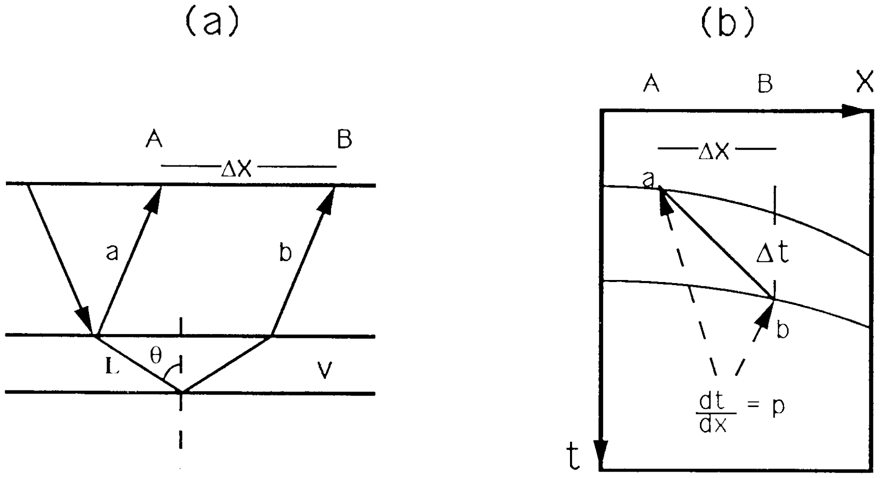

Figure 2: *(a)* Two rays with the same horizontal slowness, reflected from adjacent interfaces. *(b)* The two points in the **x-t** domain, corresponding to the rays **a** and **b**. The derivative of the reflection curve at both points is the same. carlos3-vofp [NR]

Claerbout (1978) derived an interval velocity estimation scheme that, although formulated in a different way, is closely related to this principle. His formulation is based on the analytical relation involving the interval velocity between two adjacent reflectors, the temporal and spatial coordinates of points in the two wavefronts that share the same horizontal slowness, and the horizontal slowness. The derivation of the basic equation from Figure **??** is straightforward:

$$\frac{\Delta x}{2} = L \sin\theta = v \frac{\Delta t}{2} \sin\theta$$

$$v^2 = \frac{\Delta x}{\Delta t} \frac{v}{\sin\theta} = \frac{\Delta x}{\Delta t} \frac{1}{p} \tag{1}$$

Using equation (1) as the underlying principle, Gonzalez and Claerbout (1979) derived a wave-equation-based velocity estimation method. Deregowski (1979) used the same equation to develop a tangency-picking scheme to perform interval velocity analysis. Both methods were implemented in the framework of the retarded Snell midpoint coordinates.

Equation (1) represents a direct link between the interval velocity and some measurable attributes of the data. In contrast to indirect interval velocity estimation methods, which are based either on ray-tracing or on the initial fitting of the stacking spectrum, direct methods *do not have* a cumulative *increase of error* with depth.

The next two sections describe two different velocity estimation schemes that are also based in equation (1). The first uses an automatic-picking process developed by Zhang and Claerbout (1990) and improved by Zhang (1991) to select the reflection events and compute the horizontal slownesses. The other is based on the same picking process but uses beam-stacks (Biondi, 1990) to provide a probability distribution for the horizontal slownesses.

AN EVENT-PICKING BASED ALGORITHM

The interval velocity estimation problem can be formulated as a least-square optimization scheme. With this formulation, we search for the velocity parameter that best fits equation (1) considering several points of two adjacent reflections. To avoid division by zero, and to increase the contribution from large horizontal slownesses (where the velocity information is better resolved), I define the objective function as follows:

$$\varepsilon = \sum_{i=1}^{N} [(\frac{\Delta x_i}{\Delta t_i}) - p_i w]^2 \qquad (2)$$

where $w = v^2$ and index i refers to the sampled points in one of the reflections. The solution that minimizes ε is

$$v = \sqrt{\frac{\sum_{i=1} p_i (\frac{\Delta x_i}{\Delta t_i})}{\sum_{i=1} p_i^2}} \qquad (3)$$

Here, equation (3) is implemented, with the help of an event-picking algorithm (Zhang, 1991) to define the several reflection curves. The picking algorithm defines the relative time-shifts between samples of adjacent traces. This information is used to trace events in the x–t domain starting at all samples of the first trace. Because the events should coincide with the true reflections wherever they are strong enough, the algorithm computes the stacking-power (or semblance) for each event to select the best candidates for reflections. Figure **??** shows synthetic data generated from a velocity model (with moderate velocity contrasts to avoid strong multiples contamination) overlaid with the six selected picked events. The selection was based entirely on the stacking power, but some interactive procedure will be required when strong multiples or other coherent noises are present.

This algorithm also uses the picked reflection-curves to compute also the horizontal slownesses at the sampled offsets. For any two pairs of adjacent reflections, the terms inside the sum of equation (3) are defined by using the sampled points of the bottom reflection as reference. For each of these points the algorithm finds the matching point in the upper curve, that is, the point with the same horizontal slowness as that of the

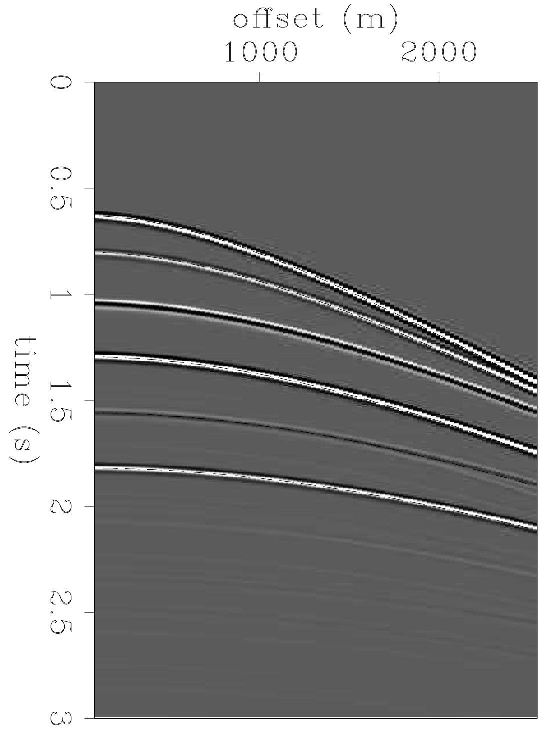

Figure 3: Synthetic data overlaid with automatically-picked reflection events. carlos3-datapk [NR]

reference point in the bottom curve. To find the matching point, the algorithm uses a spline interpolation of the inverse function $x_i = x_i(p_i)$ (defined for each sampled point i in the upper curve). Figure **??**a shows the functions $p(x)$ corresponding to all six selected events shown in Figure **??**. Clearly the horizontal slowness evaluated with this method is very sensitive to even small errors in the $t_i(x_i)$ picked curve associated with each reflection. This sensitivity requires the p-curves to be filtered before the velocity estimation step. First, the values of p that substantially differ from the local mean are removed. Then the gaps are filled by interpolation, and convolutional smoothing is applied. Figure **??**b shows the resulting filtered version of Figure **??**a.

It is worth mentioning that the horizontal slowness curves $p(x)$ need to have a monotonic behavior, so that the inverse functions $x(p)$ can be uniquely defined. In principle, this condition should always be satisfied when the plane-layered assumption is valid, but the intervention of noise may result in a nonmonotonic $p(x)$.

After the points sharing the same horizontal slownesses are matched, application of equation (3) is straightforward. Both the resulting estimated model and the true model are represented in Figure **??**. Although the relative errors in the estimated velocities are not large, we should expect to achieve much worse results when dealing with real data because of the high sensitivity of the picked-based horizontal slowness evaluation used in this algorithm.

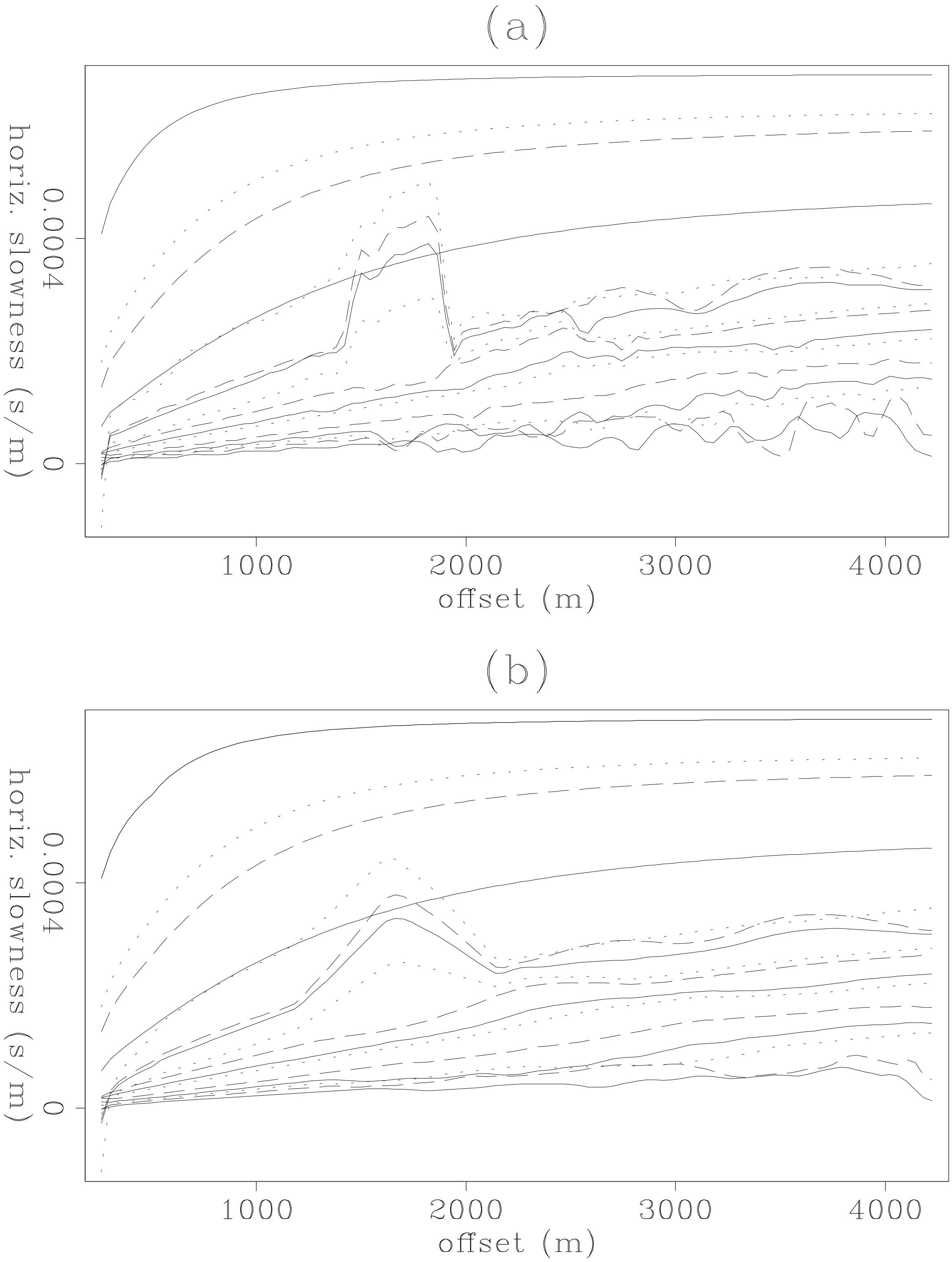

Figure 4: *(a)* Curves of p as a function of offset for all six selected reflectors and *(b)* filtered curves. carlos3-pickedp [NR]

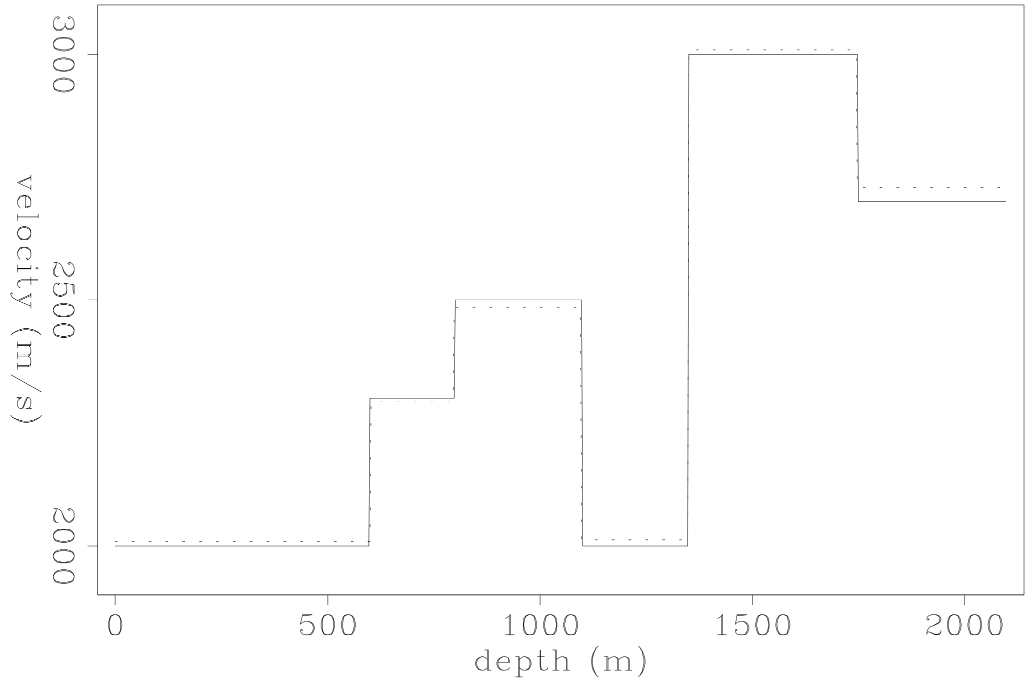

Figure 5: The **continuous** line represents the true velocity model used to generate the synthetic data of Figure **??**, and the **dotted** line represents the velocity model obtained with the picked-based method. carlos3-vel1 [NR]

A HYBRID ALGORITHM

To avoid the problems arising from the poor estimation of p in the picked-based method, I devised a hybrid algorithm, in which the events are still defined through the picking process, but instead of estimating horizontal slownesses, this algorithm uses a probability distribution $\phi(x,p)$ which is computed for each event by using beam-stacks.

Beam-stacks represent a mapping of the x-t plane onto the p-x-t cube (Biondi, 1990). For each point (x,t) in the object plane a sum is performed over a finite path, centered at that point and parametrized by p, with the result corresponding to the point (p,x,t) in the image cube. Different transforms can be defined depending on the aperture of path (number of points in the sum) and the specific choice of parametric curve: *slant-stack* ("infinite" aperture) and *local slant-stack* (finite aperture) for a straight-line path, *parabolic-transform* for an "infinite" aperture parabolic path, and so on.

The concept of beam-stacks can be generalized to include semblance as well as stacking as the basic transform operation. Biondi (1990) compared the semblance spectrum obtained with straight, parabolic, and hyperbolic finite aperture curves. His results indicated that the hyperbolic-stack has a superior resolution power.

In the hybrid scheme, we interpret the hyperbolic semblance spectrum $\phi_j(x,p)$ as the (non normalized) probability that a given point x of event j have a horizontal slowness p. Figure ??a shows one plane of the three-dimensional beam-stack of the synthetic data, and Figure ??b shows the two-dimensional beam-stack $\phi_2(x,p)$ corresponding to the second event ($t_0 \approx 0.8$ sec.) of the same data.

For a given p, v, and a point x^j of reflection j equation 1 can be used to define the related point x^{j-1} in the reflection above $(j-1)$ which share the same horizontal slowness

$$x^{j-1} = f(x^j, p, v).$$

I define the function $\phi(x^j,p,v)$, as the likelihood that point x^j and x^{j-1} share the same horizontal slowness p, in a medium with velocity v, as the sum of their beam-stacks

$$\Phi(x^j,p,v) = \phi(x^j,p) + \phi(x^{j-1},p), \tag{4}$$

where $\phi(x^j,p)$ is the beam-stack spectrum (whose maximum value is 1.0).

If the velocity is the true velocity, there will be one (and only one) point x^j for each value of p, for which Φ will assume its maximum value (2.0). If the velocity is not the true velocity, Φ will not assume its maximum value for all ps. Based on this assumption I define

$$\Psi(v) = \sum_{p=0}^{p_{max}} \max_{x_{min}}^{x_{max}} \{\Phi(x^j,p,v)\}. \tag{5}$$

The estimated velocity for the interval between $j-1$ and j is the velocity for which Ψ is maximum

$$v_{\text{est}} \mid \quad \Psi(v_{\text{est}}) = \max \Psi(v). \tag{6}$$

Differently from the picked-based method, the hybrid scheme requires the introduction of a scanning through a set of discretized velocities. For each pair of adjacent reflection events, the interval velocity is estimated by the following procedure:

For each velocity $\quad v = v_0 + i_v.dv$

  For each horizontal slowness $\quad p = p_0 + i_p.dp$

    For each offset $\quad x = x_0 + i_x.dx$

    compute $\quad \phi_{j-1}(x,p)$ # beam-stack of upper event

    compute $\quad \phi_j(x,p)$ # beam-stack of bottom event

    For each offset $\quad x^j = x_0 + i_x.dx \quad$ of the bottom event

    solve $\quad \frac{\Delta x}{\Delta t}(x^{j-1}) = p.v^2 \quad$ for $\quad x^{j-1}$ # using spline interpolation

  Evaluate $\quad \psi(i_v,i_p) = \max\{\phi_j(x^j,p) + \phi_{j-1}(x^{j-1},p\}$

$v_j = v(i'_v)$; where i'_v is such that $\chi(i'_v) = \max\{\sum_{i_p} \psi(i_v,i_p)\}$

The estimated velocity v_j corresponds to the interval velocity between reflectors $j-1$ and j. Instead of repeating this procedure for all pairs of adjacent reflectors, the

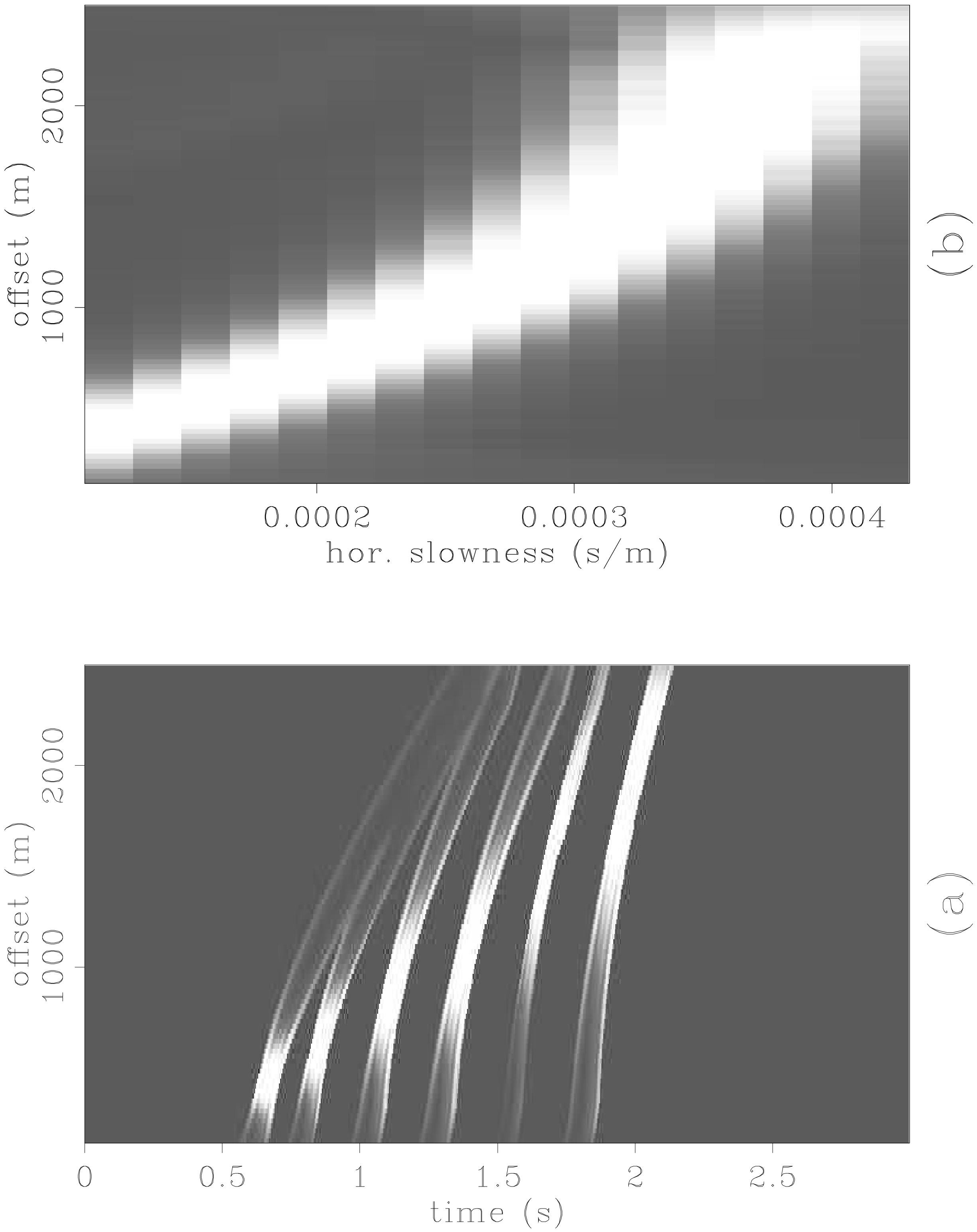

Figure 6: *(a)* A plane from the beam-stack of the data shown in Figure **??** for p =0.17 msec/meter and *(b)* a cut in the same beam-stack along the second event of that data. carlos3-beamst [NR]

actual algorithm has an internal loop in j that make it more computationally efficient. Also, the beam-stacks are computed only once, before the loop in v. Figure **??** compares the true model with the results of two inversion algorithms: one (dashed line in the figure) finds the interval velocity model whose Dix's related hyperbolas best fit the picked events; the other (dotted line in the figure) is the result from the hybrid scheme inversion. The first thing we observe in this figure is that the velocities estimated by the hybrid scheme fits better the true model than both the Dix's fitting and the purely picking-based scheme. Most important, the error in the hybrid-method estimation does not change with depth and, actually, is of the same order as the velocity discretization interval (dv) used for the inversion.

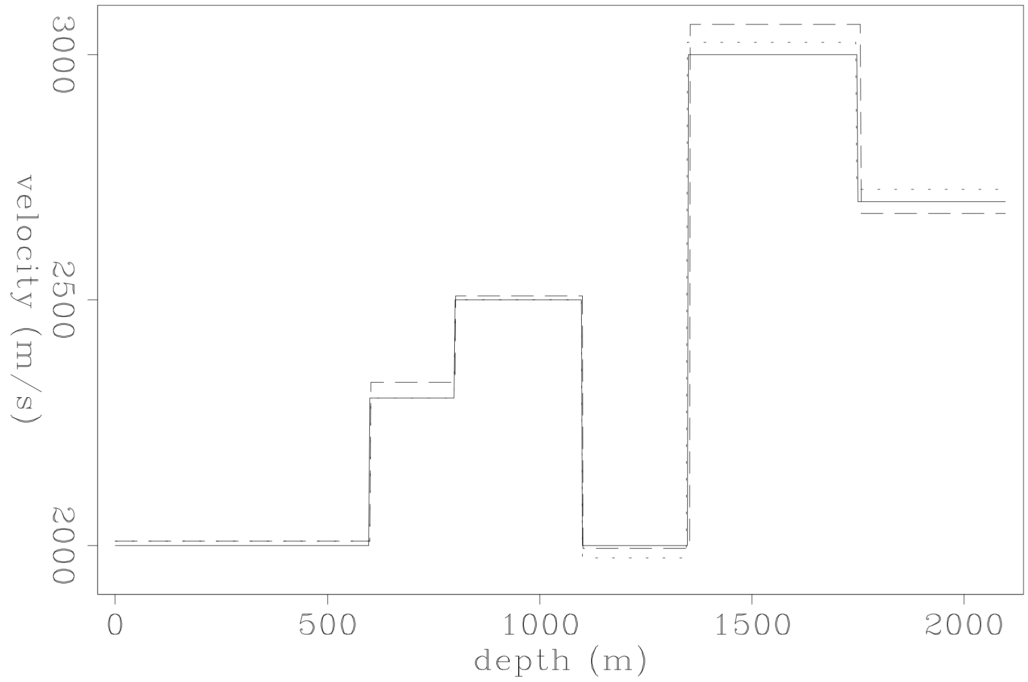

Figure 7: The **continuous** line represents the true velocity model used to generate the synthetic data shown in Figure **??**; the **dashed** line represents the velocity model estimated Dix's least squares fitting of the picked events; and the **dotted** line represents the velocity model obtained with the hybrid method. $\boxed{\text{carlos3-vel2}}$ [NR]

A REAL DATA EXAMPLE

I tested both algorithms in a common midpoint gather from Brazil offshore (Figure **??**a). The subsurface is flat-structured up to 3.1 sec ($\approx$ 4.6 Km) and have slightly dipping layers below. The data is strongly contaminated by water-layer reverberation that was not effectively removed by conventional deconvolution and dereverberation processes. The relatively long cable allows for the observation of crossing primary-reflections, and some coherent noise with linear stepout is also present. To cope with all these factors I applied a special processing sequence to the data prior to the inversion step.

Processing sequence

Figure **??** shows the processing sequence applied to the data to alleviate the influence of noise and event-crossing in the velocity inversion. Although muting could be used to eliminate event-crossing, this would result in the discharge of useful information from a particularly important part of the reflections. Multiples and other coherent noise were attenuated by filtering techniques while the problem of interference between reflections was solved by a differential treatment of the shallow reflections.

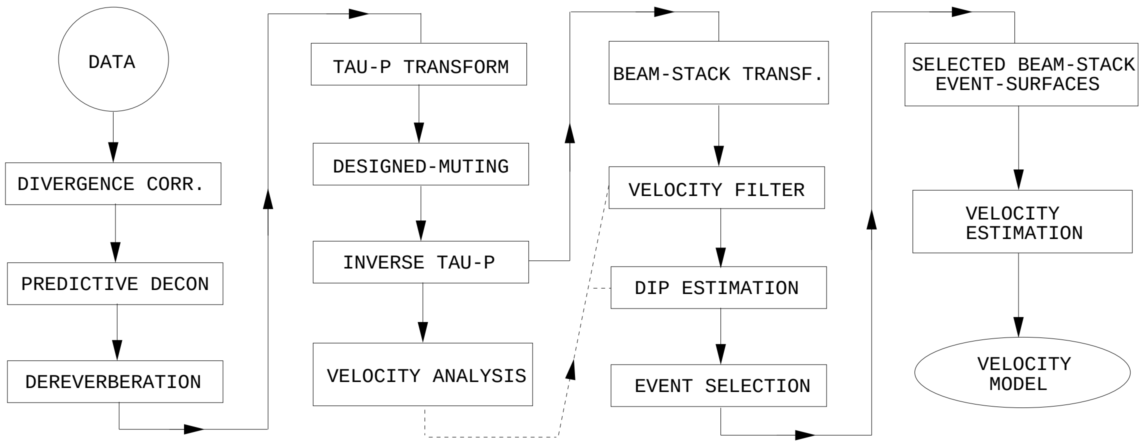

Figure 8: The processing sequence applied to the marine data shown in Figure **??***a*. carlos3-flux [NR]

It is clear from Figure **??***a* that despite the application of predictive deconvolution and a dereverberation algorithm (described in Cunha and Muir, 1989), complex multiples and peg-legs are still dominant features in the data. More effective in the elimination of the multiples and linear noises was the application of a filter in the $\tau - p$ domain. The direct and inverse $\tau - p$ transform were performed by an algorithm developed by Kostov (1989); Figure **??***a* shows the shape of the cutoff curve. The high-slowness part of the shallow reflections is located inside the rejecting region to avoid the reflections interfering in the subsequent processing of the reflections below them. Figure **??***b* shows the filtered data transformed back to the $x - t$ domain.

A comparison between the velocity spectra of the data before and after the $\tau - p$ filtering (Figure **??**) shows a good improvement in the resolution of primary events after the filtering. The times of the picks associated with primaries were used as the starting points for the event-detection algorithm. The dashed line in Figure **??***b* defines the cutoff velocity function for the velocity filter that was applied to the beam-stacked data. The application of this filter is essential to avoid the interference of multiples, not only in the event-picking algorithm, but also in the final inversion step.

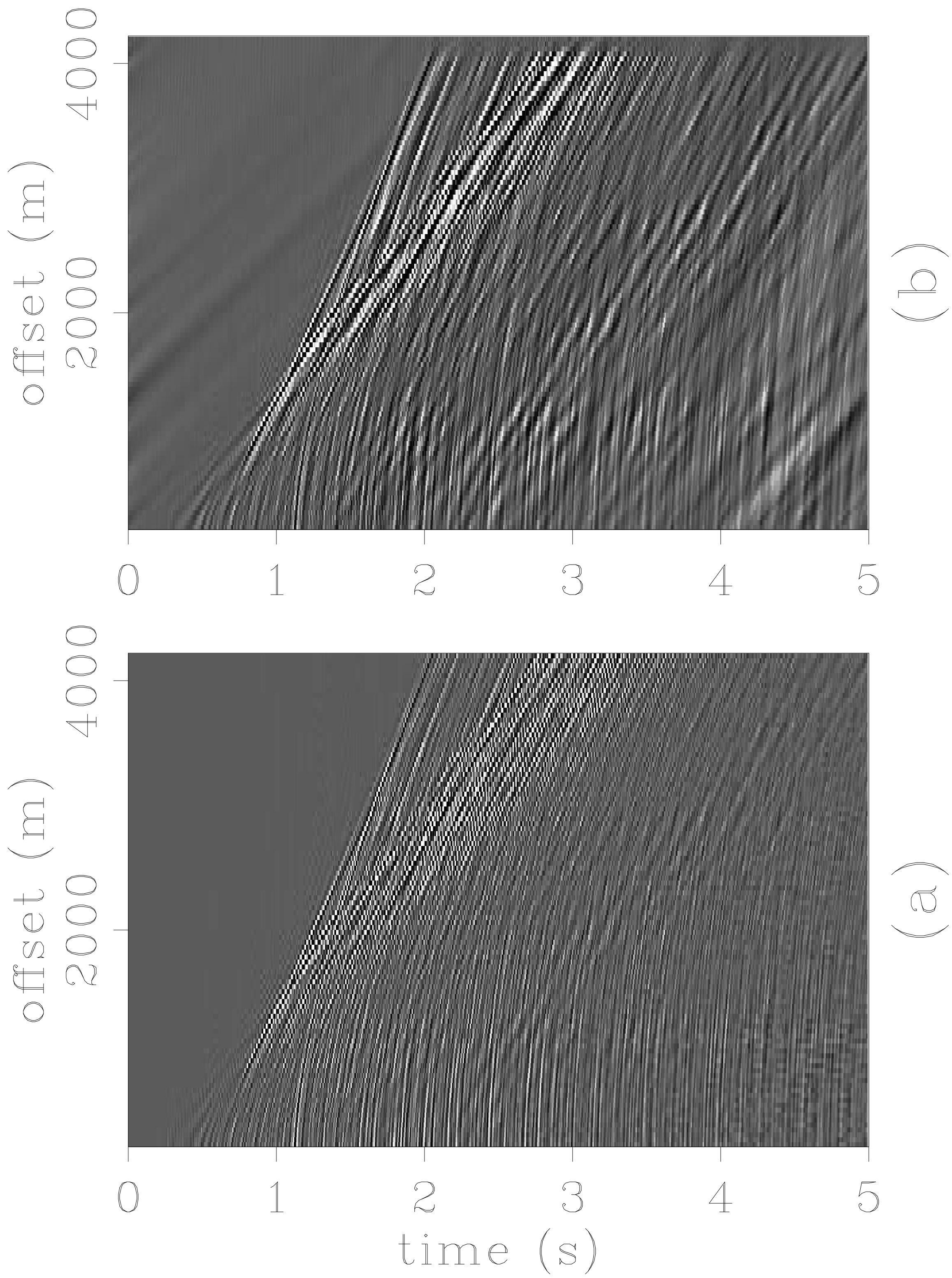

Figure 9: *(a)* CMP gather from Brazil offshore after deconvolution and dereverberation. *(b)* The same data after filtering in the $\tau - p$ domain. carlos3-cmps [NR]

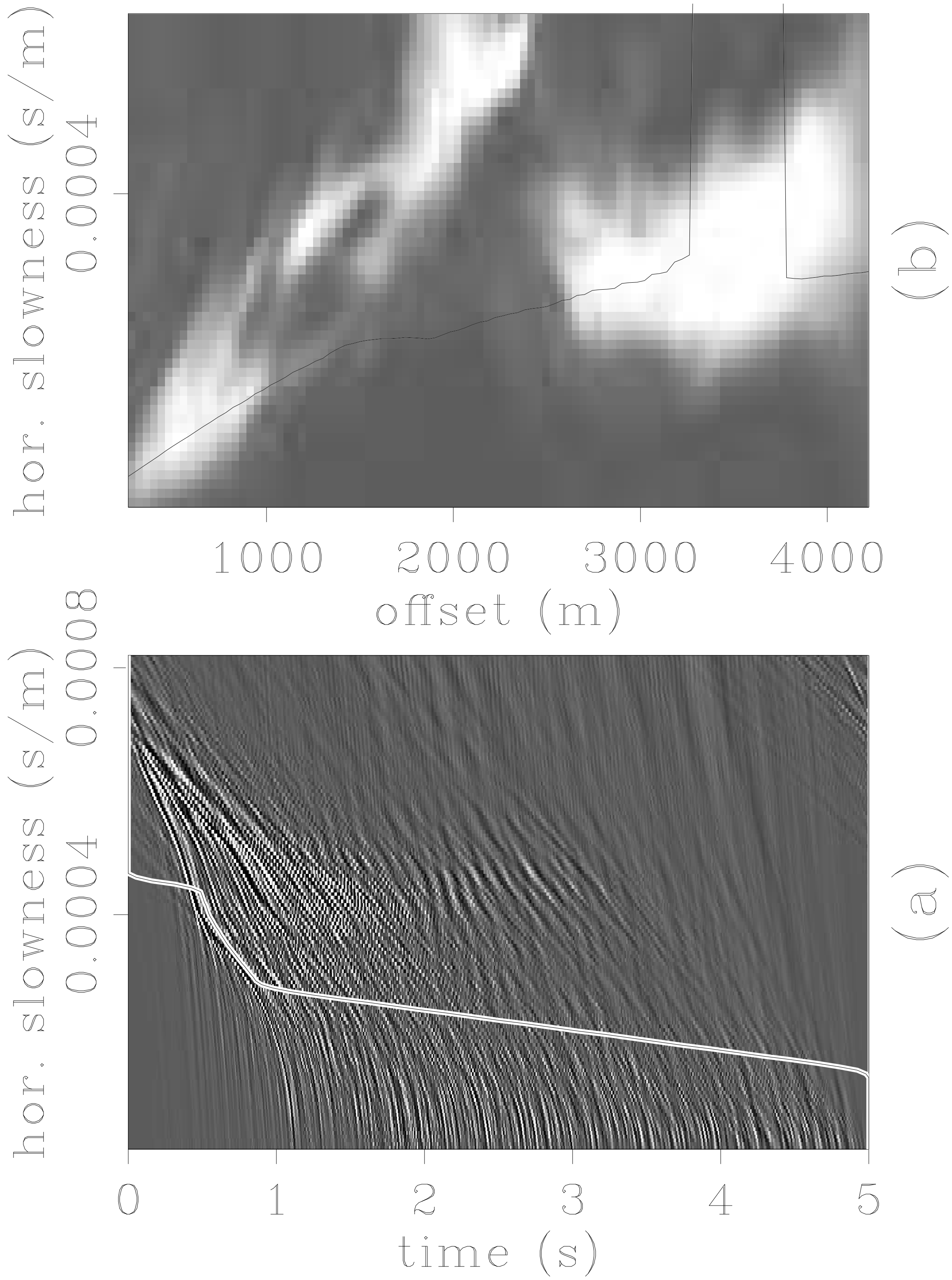

Figure 10: *(a)* The shape of cutoff curve used in the $\tau - p$ domain filter. The bandpass region lies at the left side of the curve. *(b)* A surface slice of the filtered beam-stack cube along the 8th event. The overlaid curve corresponds to the mapping of the velocity-cutoff curve of Figure**??***b* into this domain. The transition between the pass (to the left of the curve) and cut (to the right of the curve) regions is actually smooth. carlos3-taupbeam [NR]

A total of sixteen events were selected for the inversion. Their a priori nearest-offset times were selected from the velocity spectrum, but the event-picking algorithm had a pre-specified degree of freedom to reestimate them. The four initial events (which cross the reflections below) were selected with the help of a normal-moveout overlay program (Claerbout, 1987).

The last step before the velocity estimation algorithm selects the slices along the event-mapped surfaces in the beam-stack cube (Figure **??**b). The slices corresponding to the shallow events are taken from a non-[$\tau - p$]filtered version of the beam-stack cube. To optimize the estimation, the beam-stack of the shallow events (1 to 9) was computed apart from the deep events (9 to 16), so that a lower slowness interval could be used for the latter.

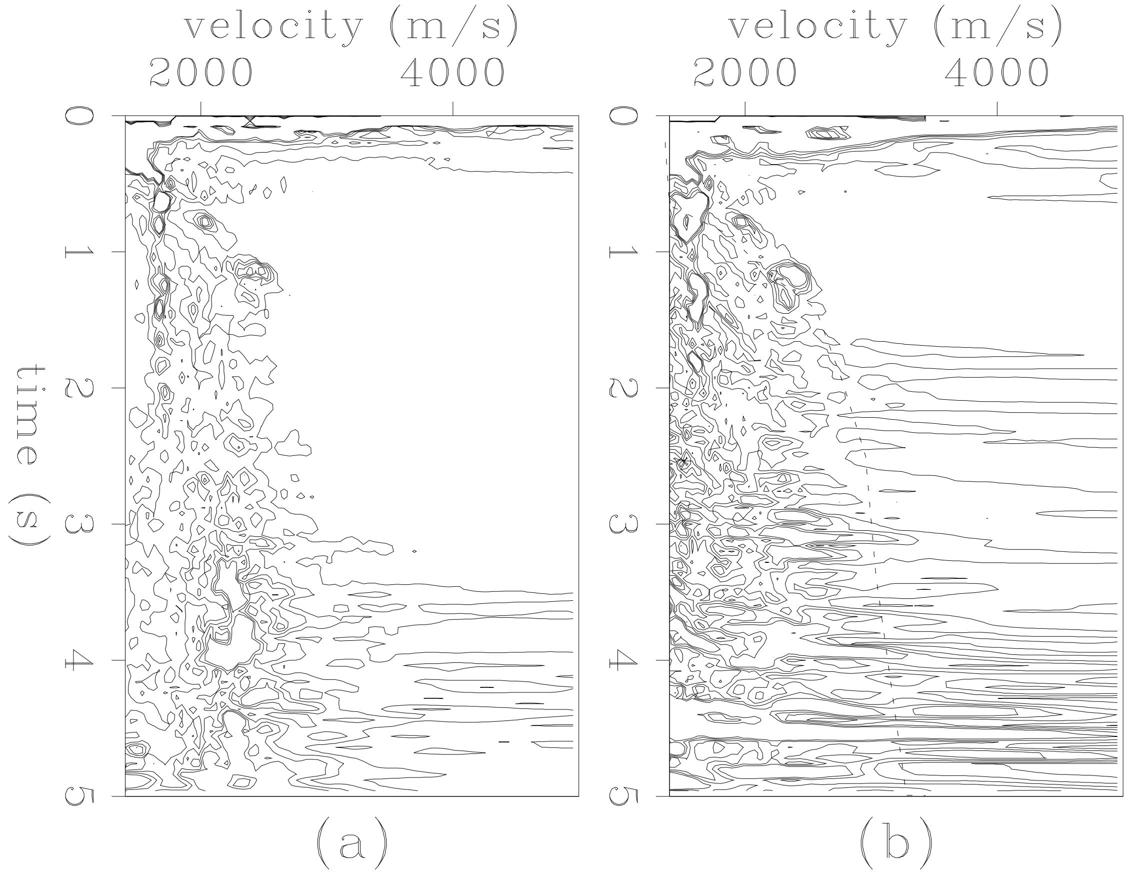

Figure 11: Velocity spectrum of *(a)* the deconvolved data, and *(b)* the same data after the $\tau - p$ filtering. The dashed line represents the cutoff curve used in the velocity-filter applied to the beam-stack. carlos3-velans [NR]

Results

Figure **??**a shows the picked reflection-events overlaid with the data shown in Figure **??**a. For clarity, the same estimated events are plotted without the data in

Figure **??**b. The detection of these events (which cross the trend of multiples and linear noise) was feasible because of the filters that were applied in the $\tau - p$ and beam-stack domains.

A sonic log from a well drilled 1 Km away from the location of the CMP gather was used to generate an interval velocity profile. This well-velocity profile was smoothed with a high-cut Walsh-spectrum filtering, to emphasize its "blockiness" and allow a better comparison with the result of the inversion. The basic lithology includes calcilutite in the top 1.5 kilometers, followed by shale in the next half kilometer, then shale-sand from 2 to 3.5 Km, shale again to a depth of 4.65 Km, and finally a calcarenite-calcilutite sequence. The low velocity interval around 1.5 Km corresponds to a highly porous sand-calcilutite transitional formation with occurrences of shale and salt lenses.

Figure **??** compares the well-velocity profile with the velocity profile estimated from a Dix's least-squares fitting of the picked events (**??**a), the result from the picking-based algorithm (**??**b), and the result from hybrid method (**??**c). The results from the three inversions are reasonably similar for the shallow layers, but the Dix-based as well as the picked-based inversions completely fail at the deeper layers, while the hybrid scheme maintains the same resolution at all depths. Surprisingly, the deep, high-velocity, carbonate sequence was almost perfectly resolved by the hybrid method. Although [for the real data case] the errors are large in comparison to the synthetic case, they are smaller than we should expect given the low signal-to-noise ratio of the real data.

CONCLUSIONS

Interval velocity estimation algorithms are usually based on either a model driven ray tracing or on an analytic approximation that relates RMS and interval velocities. For the specific case of a horizontally-layered subsurface, it is possible to retrieve the interval velocity from attributes directly measured from the data, without the use of approximations or a priori models. The hybrid scheme described herein successfully estimated the velocities when applied to synthetic and real data. The resolution obtained in the real data example was higher than expected, given the low signal-to-noise ratio of the data. As expected, the errors in the estimations obtained with this method did not increase with depth.

ACKNOWLEDGMENTS

I would like to thank Lin Zhang for providing me with his dip-estimation code and for several suggestions that contributed to this work.

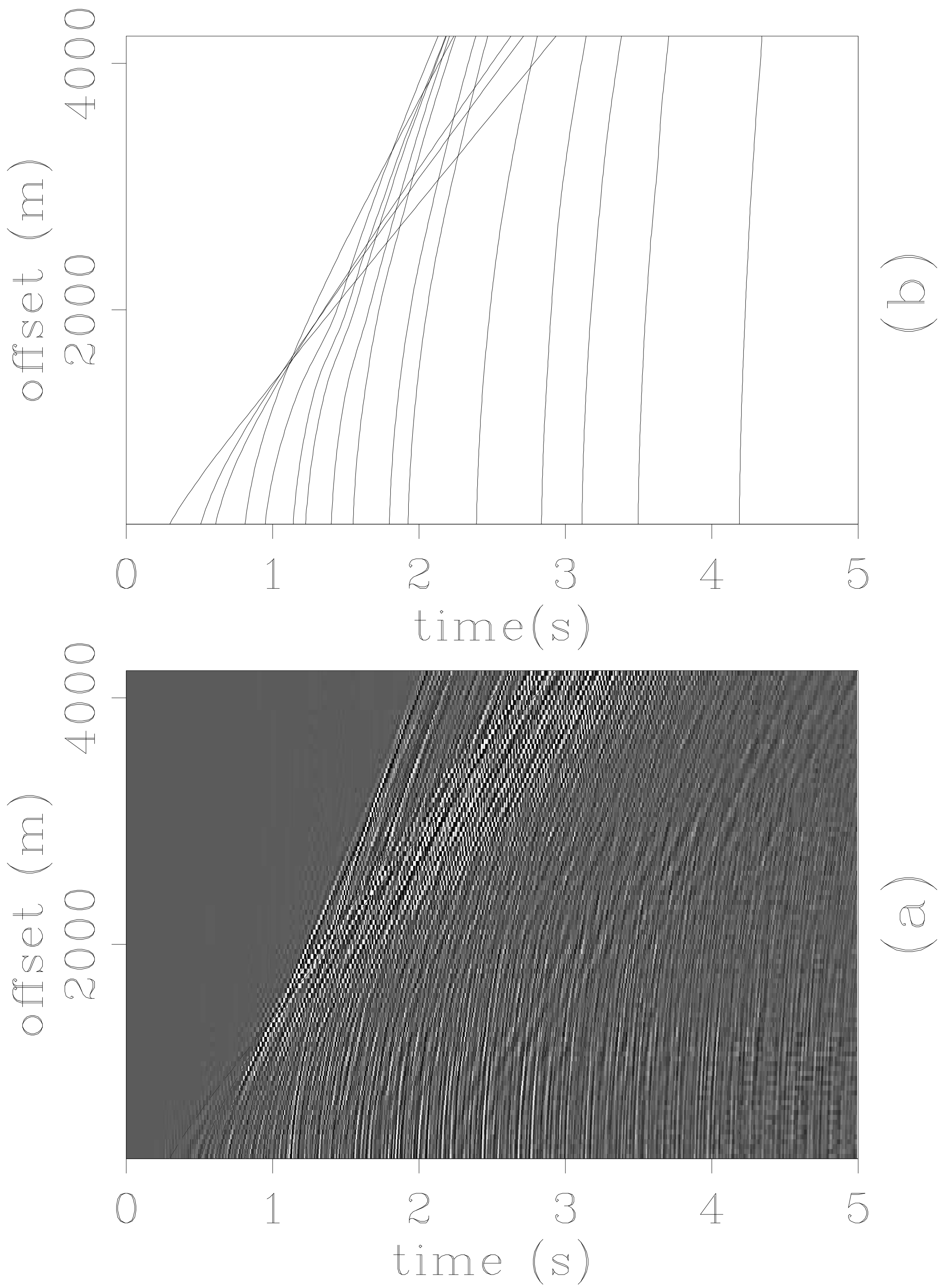

Figure 12: *(a)* Final estimation of the reflected wavefronts overlaid with the data shown in Figure **??**a. *(b)* The same wavefronts of *a*. carlos3-events [NR]

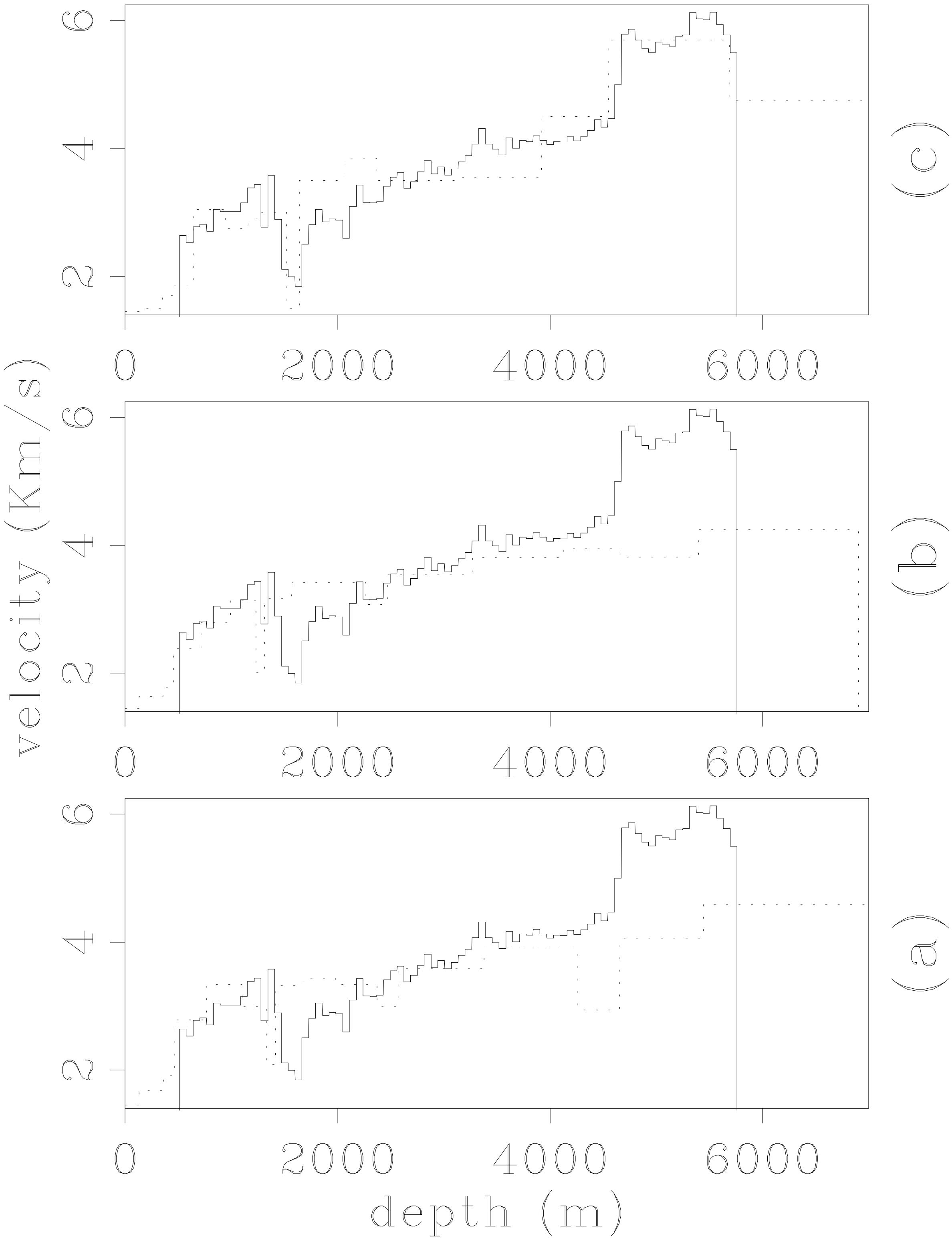

Figure 13: The **continuous** lines are the sonic-derived velocity from a well located 1 Km away from the CMP position. The **dotted** lines are the results of *(a)* the least-squares Dix-hyperbola fitting of the picked events, *(b)* the picking-based algorithm, and *(c)* the hybrid algorithm. carlos3-vels [NR]

REFERENCES

Biondi, B., 1990, Seismic velocity estimation by beam stack: Ph.D. thesis, Stanford University, also SEP-**64**.

Claerbout, J.F., 1978, Snell Waves: SEP-**15**, 57-72.

Claerbout, J.F., 1987, Interpretation with the overlay program: SEP-**51**, 269-300.

Cunha Filho, C.A., and Muir, F., 1989, Separation of coverted waves in marine data: SEP-**61**, 213-233.

Deregowski, S.M., 1979, A technique for computing interval velocities from common midpoint gathers: SEP-**16**, 205-225.

Gonzalez-Serrano, A., and Claerbout, J.F., 1979, Wave equation velocity analysis: SEP-**16**, 181-204.

Kostov, C., 1989, Finite-aperture slant-stack transforms: SEP-**61**, 261-280.

Zhang, L., and Claerbout, J.F., 1990, Automatic dip-picking by non-linear optimization: SEP-**67**, 123-138.

Zhang, L., 1991, Automatic picking and its applications: SEP-**70**, 275-292.