Stanford Exploration Project, Report 67, January 13, 1998, pages 1–136

Automatic dip-picking by non-linear optimization

Lin Zhang and Jon Claerbout[1]

keywords: *algorithm, optimization, accuracy, linear, field data, resolution*

ABSTRACT

We have developed a new automatic dip-picking method that is superior to the dip-scan method in several aspects. In our method, we estimate the dip of an event through the relative time-shift between neighboring traces. The optimal time-shift is defined to be the minimizer of a non-quadratic objective function that measures the discrepancies between the neighboring traces after these traces are shifted relative to one another. To eliminate aliasing effects, data-dependent weighting functions are included in the objective function. This non-linear optimization problem is solved by searching. Once a preliminary solution is obtained, the objective function is approximately reduced to a quadratic form and the residual time-shift is then estimated by solving a linear equation. In the end, the time-shift is converted into the dip.
Examples with synthetic and field data show that the combination of the linear and non-linear optimizations enables our algorithm to have the properties of antialiasing, high resolution and high accuracy. The applications of the algorithm include event-picking, moveout corrections, local dip-filtering and missing data interpolation.

INTRODUCTION

Picking is a process of identifying the events or estimating the parameters of the events on seismic sections. This process can be done manually, or automatically by computers. Because human eyes have strong capability of recognizing seismic events on a noisy background, manual picking are often more robust than any computer algorithms. But manual picking has the disadvantage of being inaccurate in estimating the parameters of events. Also, manual picking is inefficient and expensive. When a seismic section contains a great many of events, automatic picking becomes more than necessary.

A commonly used algorithm for automatic dip-picking is dip-scan. The basic principle of the algorithm is that the optimal estimate of the dip of an event corresponds

[1]**email:** not available

to the maximum value of the local slant-stack. This method does not work when data is insufficiently sampled. Moreover, the accuracy of the estimation is limited by the dip-sampling interval of the dip-scan and its spatial resolution is limited by the spatial extent of the dip-scan.

Claerbout (1990) proposed an automatic dip-picking method that is based on a plane-wave destructor operator. Because the plane-wave destructor is a 2-D filter that can be represented by a 2×2 differencing star, this algorithm is computationally efficient and inexpensive. It has its spatial resolution equal to the spatial-sampling interval of data. The disadvantage of the method is that the finite difference operator is susceptible to spatial aliasing as well as distortions at spatial frequencies that are high but not yet aliased. Using a high-order finite differencing operator would reduce the distortions, but it will also reduce the spatial resolution.

While studying the missing data interpolation, Nichols (1990) used the plane-wave destructor operator to estimate the dips of events. He solved the aliasing problem by using the smoothed version of data to obtain the initial estimates, and then refining them after the missing traces are interpolated. The application of this method is limited to data that has low frequency and low wavenumber components.

In this paper, we describe a new algorithm that does automatic dip-picking. We assume that the events on a seismic section do not intersect. In other words, at each point on the seismic section, there is a unique value that defines the dip of the event at that point. We also assume that data is adequately sampled along the time axis, but we do not make any assumptions on the spatial-sampling rate. In our method, the relative time-shift between neighboring traces is estimated by minimizing a non-quadratic objective function. Data-dependent weighting functions are included in the objective function to remove the effects of aliasing on picking. After a preliminary estimate of the relative time-shift is resolved, the objective function is approximately reduced to a quadratic form and the residual time-shift is obtained by solving a linear equations. Finally the time-shift is converted into the dip.

The paper is organized in four parts: first, we will formulate the non-linear optimization problem and explain why the effects of aliasing on dip-picking disappear; then we show that the residual time-shift can be estimated efficiently and accurately by solving a linear optimization problem; third, we will use examples with synthetic and field data to confirm the results predicted by theory and illustrate the applications of the algorithm. Finally, in conclusion, we will discuss the possible improvements and extensions of the algorithm.

NON-LINEAR OPTIMIZATION

The dip p of an event at a point is related to the relative time-shift τ between neighboring traces around that point:

$$p = \frac{\tau}{\Delta x}. \tag{1}$$

where Δx is the spatial-sampling interval. Therefore estimating the relative time-shift is equivalent to estimating the dip. The optimal estimate of τ is defined to be the relative time-shift that will make the neighboring traces resemble each other. The discrepancies between these traces after shifting are measured by an objective function.

Objective function

Suppose $u_1(t)$ and $u_2(t)$ are two traces of seismic data $u(t, x)$:

$$u_1(t) = u(t, x_1), \qquad u_2(t) = u(t, x_2)$$

with $\Delta x = x_2 - x_1$ being the distance between the two traces. We want to construct an objective function that measures the misfitting of two traces when they are shifted relative to one another. The standard least squares method suggests the following two expressions

$$[u_1(t) - u_2(t+\tau)]^2 \quad \text{or} \quad [u_2(t) - u_1(t-\tau)]^2. \tag{2}$$

These two expressions are useful when seismic data is adequately sampled along the spatial axis. However, when data is aliased, these expressions as the functions of τ, have many local minima, as shown in Figure 1c. This may lead to a totally wrong estimate of the time-shift τ. The solution to this problem is to use some special information of data as constraints to reduce, or even to remove, the ambiguities.

There are many ways to constrain the solutions. The two commonly used methods are: adding extra terms to the objective function and multiplying weighting functions to the objective function. The method of adding extra terms can possibly preserve some of the important properties of the objective function, such as quadratic forms. But the relative weights between the two terms of the objective function are not easy to determine. The weighting method does not have this problem. But it will make the objective function non-quadratic when the weighting function are data-dependent. Since the expressions in equation (2) are not quadratic functions of time-shift τ anyway (thought they are quadratic functions of data), the weighting function method is apparently easy to apply.

Human eyes can eliminate the aliasing effects on dip-picking by looking at the amplitude information of data, especially the breaks of first arrivals. This suggests using the following objective function:

$$E(\tau) = \frac{1}{N}\sum_{t}^{N}\left\{\frac{1}{\|u_1^2(t)\|}[u_1(t) - u_2(t+\tau)]^2 \frac{1}{\|u_2^2(t)\|}[u_2(t) - u_1(t-\tau)]^2\right\}, \tag{3}$$

where $\|u\|$ represents the smoothed version of u, and the summation is undertaken in the neighborhood of point (t, x). The reason to choose this kind of weighting function is clearly shown in Figures 1a and 1b. The penality is extremely large when the first breaks of traces do not match with each other. Figures 1a and 1b also show why both expressions in equation (2) must be included in the objective function. Figures 1c and 1d show that after the weighting functions are applied, false local minima become less ambiguous.

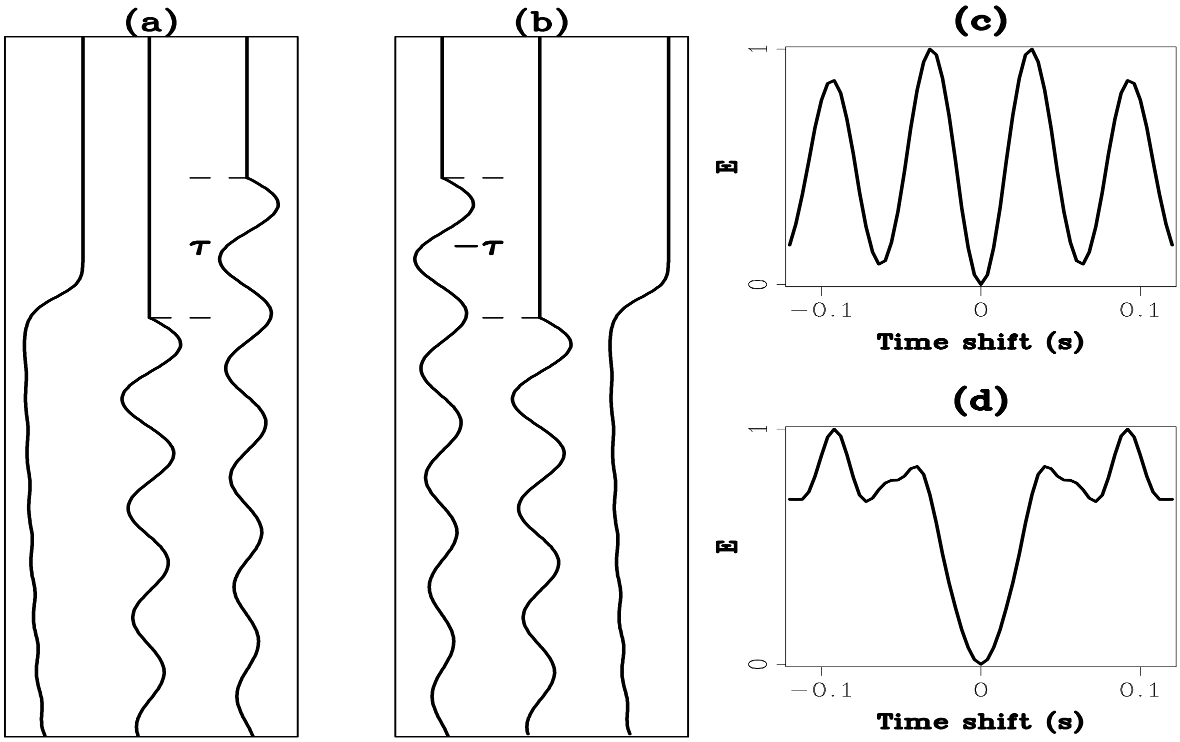

Figure 1: Illustrations of the objective function: (a) three traces are $\|u_1^2(t)\|^{-1}$, $u_1(t)$ and $u_2(t+\tau)$ from left to right, the weighting function $\|u_1^2(t)\|^{-1}$ causes the penalty to be extremely high when τ is not zero; (b) similar to (a) where three traces are $u_1(t-\tau)$, $u_2(t)$ and $\|u_2^2(t)\|^{-1}$; (c) the objective function computed from traces in (a) and (b) without the weighting functions, it has several suspicious minima; (d) the objective function with the weighting functions, suspicious minima are lifted up. lin1-objjill [NR]

An alternative form of the objective function is

$$Q(p) = \frac{1}{N}\sum_t^N \left\{ \frac{1}{\sqrt{\|u_1^2(t-\frac{\tau}{2})\| \|u_2^2(t+\frac{\tau}{2})\|}} [u_1(t-\frac{\tau}{2}) - u_2(t+\frac{\tau}{2})]^2 \right\} \tag{4}$$

which has one term and displays better symmetry around time t. But the disadvantage of this form is that the weighting function is τ-dependent.

Algorithm

Up till now, we have treated τ as a constant. In the real world, however, the seismic reflections are almost always curved events. Therefore, τ changes its value from place to place. To deal with this nonstationarity, we consider the curved events as piecewise linear and chop a seismic section into many subsections. The optimal τ in a subsection is the minimizer of the objective function that is computed from the data samples in the subsection. This is a one-dimensional non-linear optimization problem and can be conveniently solved by searching. A straight forward implementation of the algorithm is to read the data samples in a subsection, then to compute the objective function for all possible τ, and finally to find the τ that minimizes the objection function. These steps are repeated for every subsection. This implementation is actually cumbersome and computationally expensive because many boundaries must be properly handled and the objective function may be computed more than once at the points shared by several subsections.

We have found an efficient way to implement the algorithm. The scheme of the implementation is:

```
Initialization :
  for each point
    compute :E(τ) = [u1(t) − u2(t)]^2/||u1^2(t)|| + [u2(t) − u1(t)]^2/||u2^2(t)||
  convolve :E(τ) = E(τ) ∗ Win(w1, w2)
  for each point
  [ set :Emin(τ) = E(τ)
  [ set :τopt = 0
Seraching :
  for each τ
  [ for each point
  [   compute :E(τ) = [u1(t) − u2(t + τ)]^2/||u1^2(t)|| + [u2(t) − u1(t − τ)]^2/||u2^2(t)||
  [ convolve :E(τ) = E(τ) ∗ Win(w1, w2)
  [ for each point
  [   if(E(τ) < Emin(τ))
  [     [ set :Emin(τ) = E(τ)
  [     [ set :τopt = τ
```

The only kind of boundaries that the program has to deal with is the truncation of data, but it can be easily handled by padding zeros. The code of this implementation vectorizes fully.

LINEAR OPTIMIZATION

In last section, we have described how to estimate the relative time-shift through the non-linear optimization. Although our method has several nice properties, yet it also has a defect: the accuracy of the estimation is limited to the time-sampling interval. This is because data is available only at the integer multiples of the sampling interval. One way to solve the problem is to interpolate data along time axis before the algorithm is applied to it. However, in addition to the cost of interpolation, the computational cost of the non-linear optimization algorithm increases much faster than does the accuracy of the estimation. For example, let k be the ratio of the new sampling interval to the original sampling interval, the computational cost will increases by a factor of k^2 while the accuracy increases only by a factor of k. Another way is to interpolate the computed objective function. But the solutions obtained by using this method will depend on the choice of interpolation algorithms.

In this section, we will describe an elegant algorithm that does not require heavy computation, and yet has the potential to find the exact solution.

Objective function

Let τ_0 be an estimate of the relative time-shift obtained through the non-linear optimization. We assume the true solution is

$$\tau = \tau_0 + \Delta\tau, \tag{5}$$

where $\Delta\tau$ is a residual time-shift. We will estimate $\Delta\tau$ by solving a linear optimization problem. First, we linearize the model.

$$\begin{aligned} & u_1(t) - u_2(t+\tau) \\ = \; & u_1(t) - u_2(t+\tau_0+\Delta\tau) \\ \approx \; & u_1(t) - u_2(t+\tau_0) - u_2'(t+\tau_0)\Delta\tau, \end{aligned}$$

and

$$\begin{aligned} & u_2(t) - u_1(t-\tau) \\ = \; & u_2(t) - u_1(t-\tau_0-\Delta\tau) \\ \approx \; & u_2(t) - u_1(t-\tau_0) + u_1'(t-\tau_0)\Delta\tau. \end{aligned}$$

Then we substitute these expressions into equation (3). The new objective function becomes a quadratic function of the unknown $\Delta\tau$:

$$\begin{aligned} \hat{E}(\Delta\tau) \;=\; & \frac{1}{N}\sum_t^N \left\{ \frac{1}{\|u_1^2(t)\|}[u_1(t) - u_2(t+\tau_0) - u_2'(t+\tau_0)\Delta\tau]^2 + \right. \\ & \left. \frac{1}{\|u_2^2(t)\|}[u_2(t) - u_1(t-\tau_0) + u_1'(t-\tau_0)\Delta\tau]^2 \right\}. \end{aligned} \tag{6}$$

The optimal estimation of $\Delta\tau$ is obtained by finding the minimizer of $\hat{E}(\Delta\tau)$. The standard least squares technique gives

$$\Delta\tau = -\frac{\sum_t^N \left\{ \frac{1}{\|u_1^2(t)\|}[u_2(t+\tau_0) - u_1(t)]u_2'(t+\tau_0) + \frac{1}{\|u_2^2(t)\|}[u_2(t) - u_1(t-\tau_0)]u_1'(t-\tau_0) \right\}}{\sum_t^N \left\{ \frac{1}{\|u_1^2(t)\|}[u_2'(t+\tau_0)]^2 + \frac{1}{\|u_2^2(t)\|}[u_1'(t-\tau_0)]^2 \right\}}. \tag{7}$$

This solution looks similar in expression to the solution of the plane-wave destructor. It actually has two distinguished features. First it includes weighting functions. Second, it does not require to compute the partial derivatives of data with respect to the spatial axis, which is difficult to do in high accuracy when spatial-sample interval is large. The second feature comes from the fact that we estimate the relative time-shifts instead of the dips. The computation of the time-derivatives of data is not a problem as we have assumed that data is adequately sampled in the time axis.

Thus, the overall optimal relative time-shift is

$$\tau_{opt} = \tau_0 + \Delta\tau \tag{8}$$

and the optimal dip is

$$p_{opt} = \frac{\tau_{opt}}{\Delta x}. \tag{9}$$

Clearly the linear optimization can be run iteratively. Examples with synthetic and field data show that first iteration provides a solution of sufficient accuracy.

The quality of picking

As two neighboring traces will not always be matched perfectly just by shifting, it is useful to define some quantity that measures the quality of the picking process. This quantity will have its value between 0 and 1. It will equal to 0 when input is all-zero or white noise, equal to 1 when the matching can be made perfectly. In the plane-wave destructor method, Claerbout (1990) chose the normalized correlation of the partial derivatives of data as a measure. Here, we will use the normalized correlation of data itself:

$$Q^2 = 1 - \frac{E(\tau_{opt})}{W(\tau_{opt})}, \tag{10}$$

where

$$W(\tau) = \frac{1}{N}\sum_t^N \left\{ \frac{1}{||u_1^2(t)||}[u_1^2(t) + u_2^2(t+\tau)] + \frac{1}{||u_2^2(t)||}[u_2^2(t) + u_1^2(t-\tau)] \right\}.$$

By inserting $E(\tau)$ in equation (3) into this definition, we have

$$Q = \sqrt{\frac{2\sum_t^N \left[\frac{1}{||u_1^2(t)||}u_1^2(t)u_2^2(t+\tau_{opt}) + \frac{1}{||u_2^2(t)||}u_2^2(t)u_1^2(t-\tau_{opt})\right]}{\sum_t^N \left\{\frac{1}{||u_1^2(t)||}[u_1^2(t) + u_2^2(t+\tau_{opt})] + \frac{1}{||u_2^2(t)||}[u_2^2(t) + u_1^2(t-\tau_{opt})]\right\}}}. \tag{11}$$

The computations of the linear optimization and the quality measure can be done in a similar way to that of the nonlinear optimization. The code vectorizes fully.

EXAMPLES

Examples with synthetic data will show the accuracy of the algorithm and examples with field data will illustrate how noises will affect the output.

Synthetic data

We begin with a simple synthetic section shown in Figure 2a. The events on this section have a constant time-dip. The time-shift corresponding to this dip is not an integer multiples of the time-sampling interval Δt. We want to show that the non-linear optimization gives an optimal time-shift that is the integer multiple of Δt closest to the true solution, and that the linear optimization improves the solution to high accuracy. From 2b, we see that these are indeed true. The time-dip 0.275 gives a time-shift of $4.4\Delta t$ per Δx. The solution of the non-linear optimization is $4\Delta t$. The overall solution is around $4.4 \sim 4.5\Delta t$.

Data in the second example is shown in Figure 3a. It was first generated by Claerbout (1990) to test the plane wave destructor operator. At far offsets, data suffers from spatial aliasing. A trace at the offset of 3.5 km is shifted on purpose. In this example, we want to confirm two results: (1)our algorithm can handle data aliased spatially; (2) the spatial resolution of our algorithm is approximately equal to the spatial-sampling interval of data. Figure 3b shows the result of dip-picking. In this figure, we use short line-segments to represent local dips and overlay them on data. We see the line segments follow the dips of events everywhere. This result is superior to that of the plane wave destructor method. The algorithm also picks correctly the steep dips caused by the shifted trace, which proves the high resolution of the algorithm.

Now we use the same data to illustrate three applications of the algorithm. The first one is event-picking. We can connect the short line-segments displayed in Figure 3b to make curves that follow the events. Figure 4a shows the result of the event-picking. This procedure is useful in velocity analysis and seismic data interpretation. The second application is to do moveout corrections. Once events are picked, we can shift the traces to flatten the curves in Figure 4a. This gives us a result shown in Figure 4b. Because the strategy of shifting is to make the traces at far offsets resemble the traces at inner offsets, this algorithm is good for stacking data after moveout corrections.

The last application is local dip-filtering. The picked time-shift τ_{opt} at (t, x) tells us that the data sample $u(t + \tau, x + \Delta x)$ matches the data sample $u(t, x)$ best. It also tells us that the dip at this location is $\tau_{opt}/\Delta x$. A local dip-filtering can be done by subtracting $u(t + \tau_o pt, x + \Delta x)$ from $u(t, x)$ whenever the dip at (t, x) is not in the passband. Figures 5a and 5b show the results of a low-pass dip-filter and a high-pass dip-filter. This application would become very interesting if our algorithm is generalized to multiple dip-picking.

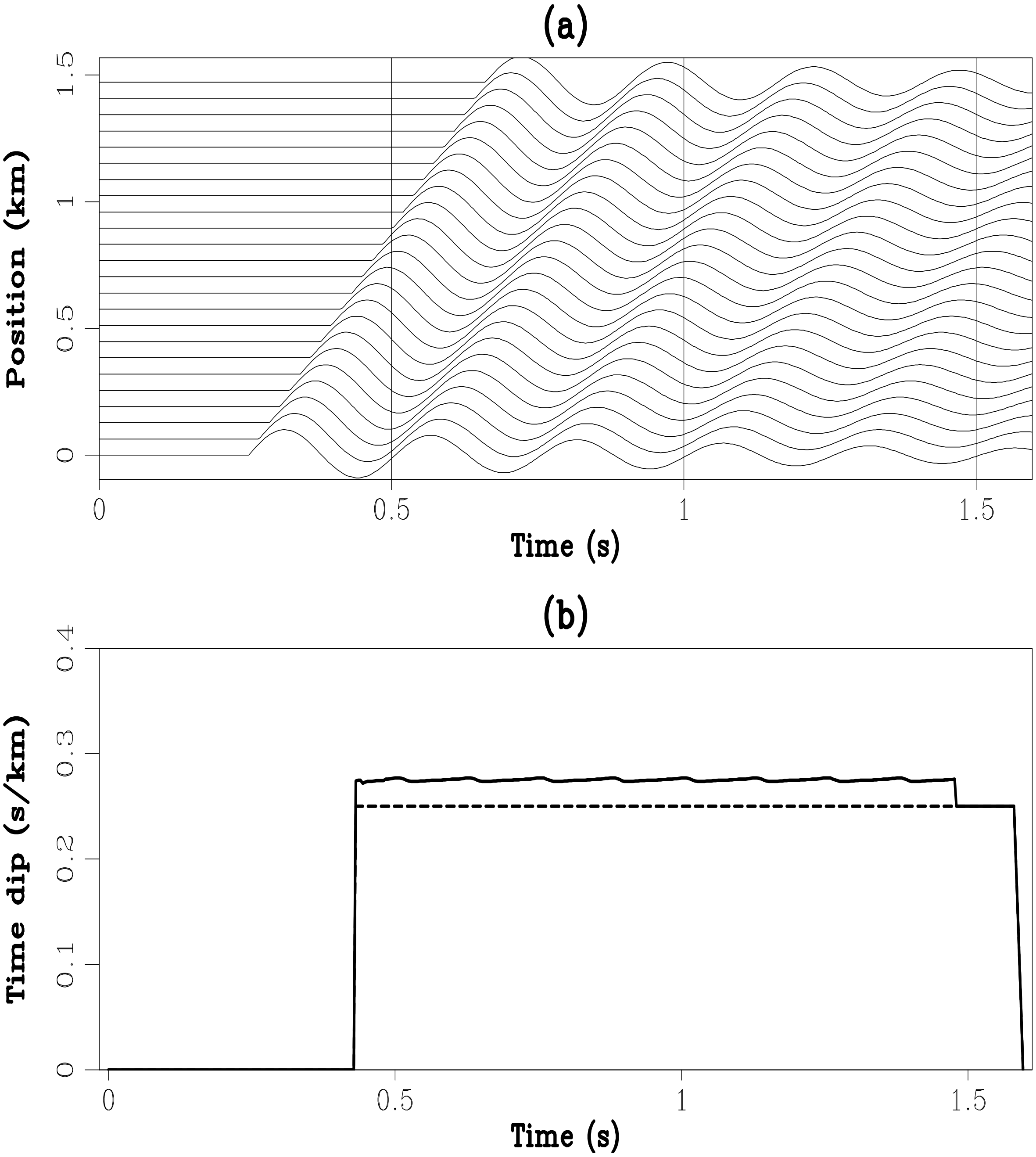

Figure 2: Accuracy of the picking algorithm: (a) plane waves with a constant time-dip equal to 0.275; (b) the dash-line shows the picked time-dip at a fixed horizontal location by nonlinear optimization, and the solid line shows the final result of the algorithm. lin1-linepick [NR]

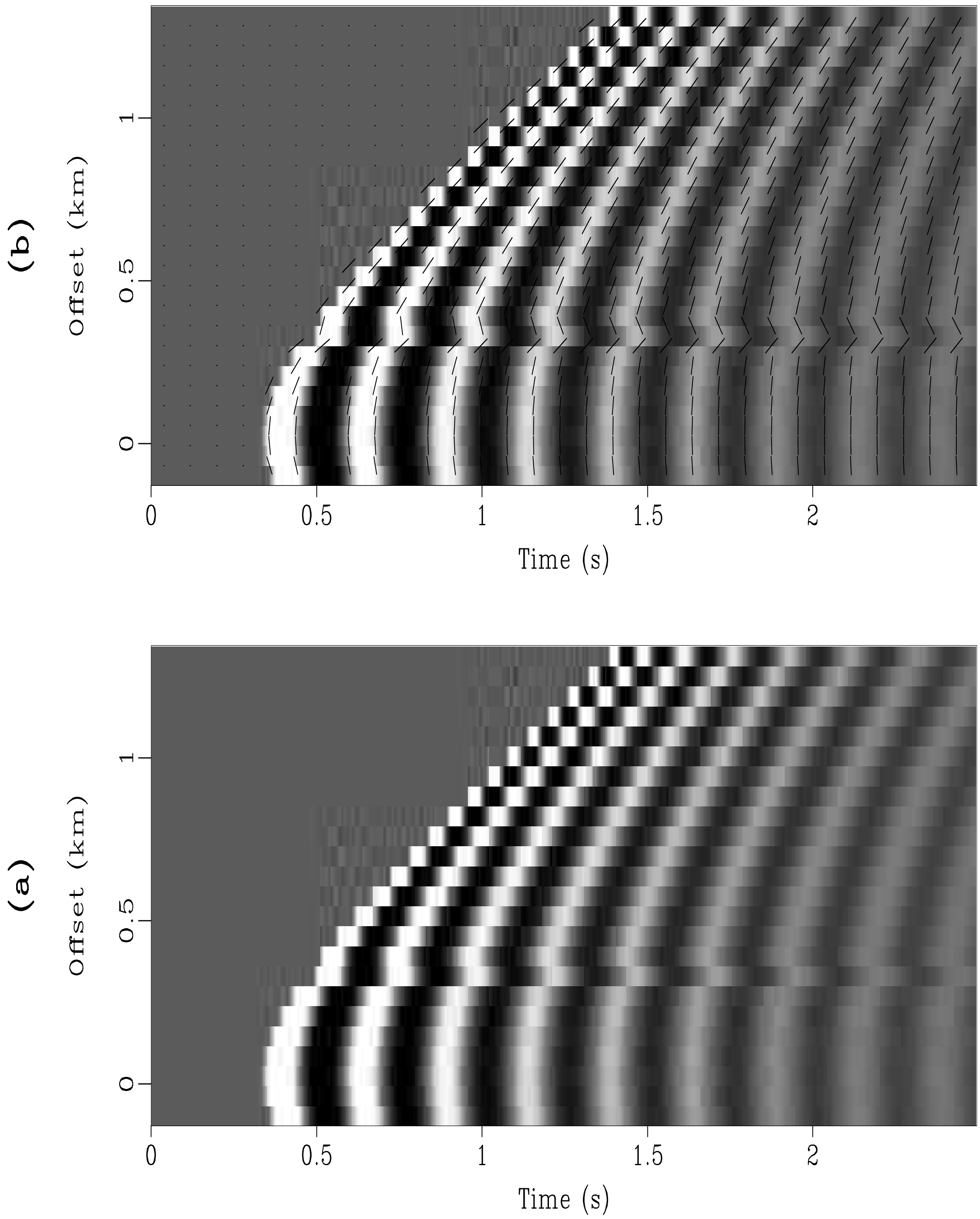

Figure 3: An example of dip-picking with synthetic data: (a) a synthetic CMP gather in which one trace is over-shifted on purpose, and events are aliased spatially at far offsets; (b) picked dips are represented by short dipping line-segments and are overlain on data. $\boxed{\text{lin1-sthyppick}}$ [NR]

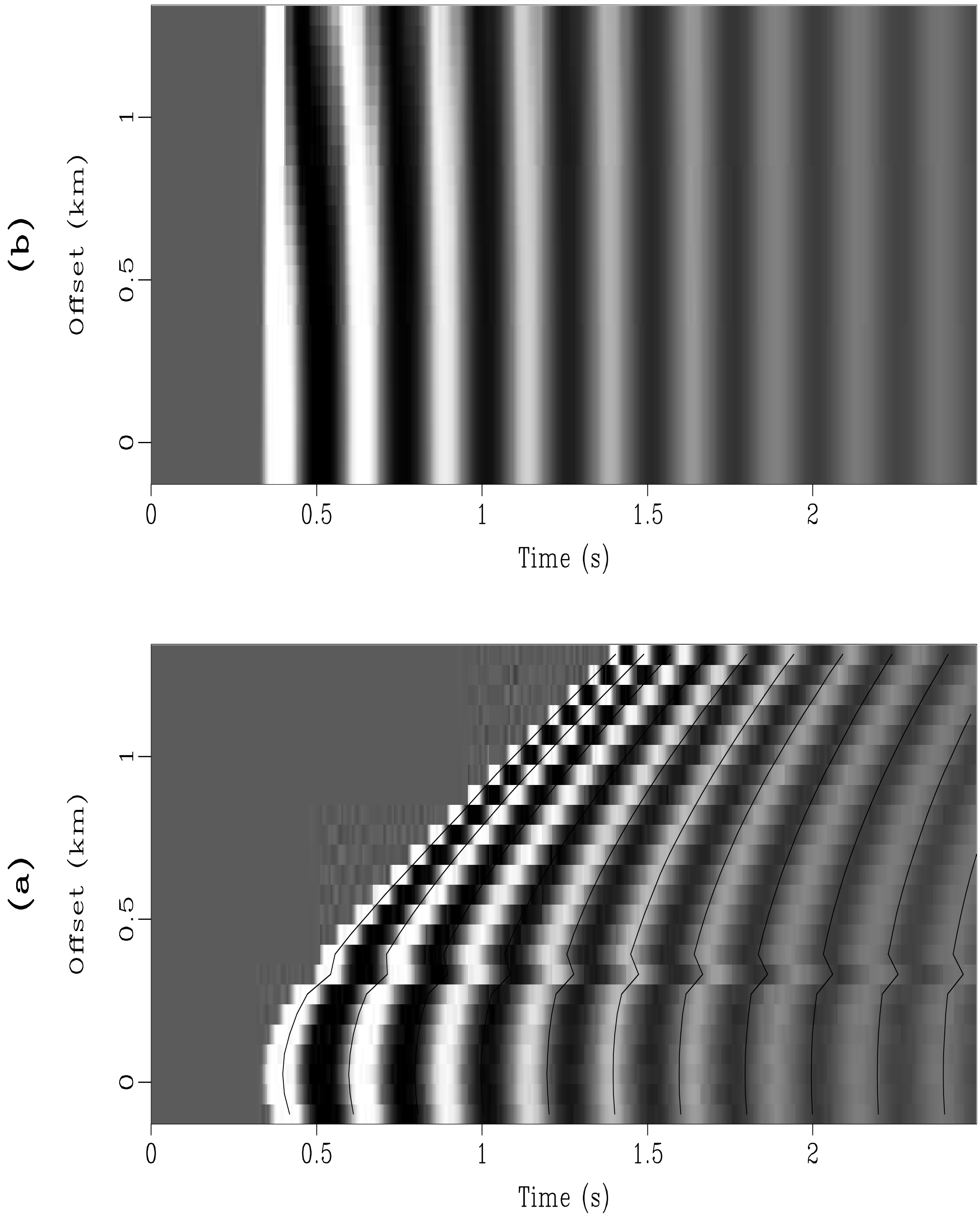

Figure 4: Applications of the algorithm: (a) event-picking; (b) moveout corrections. lin1-sthypappl [NR]

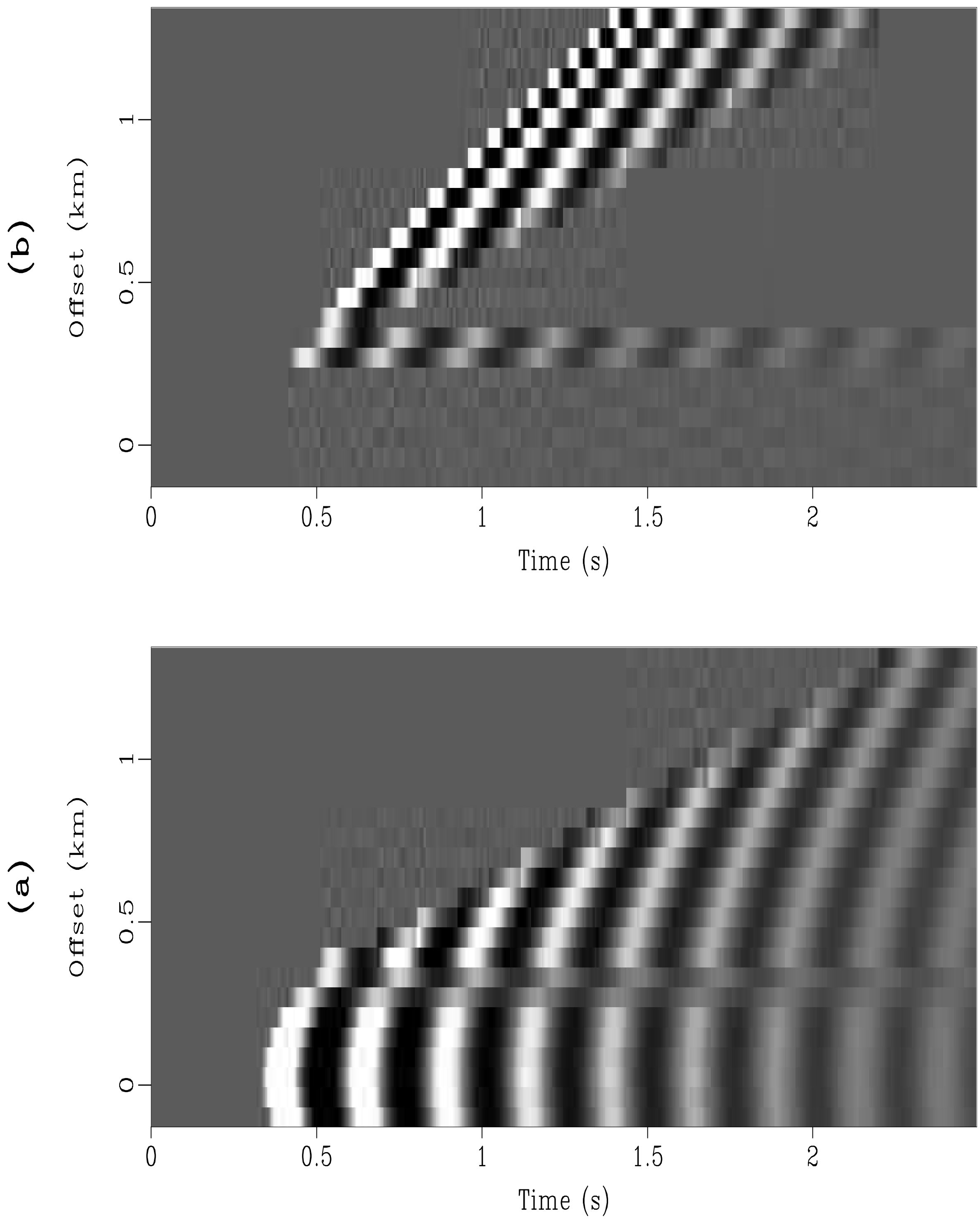

Figure 5: Examples of local dip-filtering: (a) low-pass; (b) high-pass. lin1-sthypdfil [NR]

Field data

We will show two field-data examples. First, let us look at a common receiver gather recorded in a Walk-Away marine survey. The gather is displayed in Figure 6a. This gather is collected with many shots on the surface and a common receiver down into a well. The first arrival and several events that follow it are down-going waves. These events are strong and are expected to be picked correctly. After 0.75 seconds, the up-going waves reflected from a deep subsurface start to interfere with the down-going waves. We are curious to know the results of our algorithm when events intersect. Figure 6b shows the result of picking. The picked curves are overlain directly on the data. We see that the algorithm picks the first arrival and several early arrivals quite well. Carefully looking at the picked curve for the first arrival, we see that a "pull-down" of the first arrival at offset -0.6 km is correctly picked. This "pull-down" probably is caused by a low velocity anomaly of the medium. This observation proves the high resolution and high accuracy of our algorithm. At the lower-left corner of Figure 6, events interfere with one another. We see that the picked curves more or less follow the relatively strong events. Since this situation violates our assumption, we do not expect to have good picking. The same thing happens at offset 0.7 km and arrival time 0.76 sec, where two events merge to one.

Our second example uses a common midpoint (CMP) gather. We want to demonstrate the power of our algorithm for moveout corrections. In Figure 7a, we show this CMP gather together with the picked curves. The events between the second and third picked curves are strongly aliased at far offset. The result of moveout correction is show in Figure 7b. We see that events are flattened nicely. To compare our method with normal moveout (NMO) corrections, we apply two methods to the same CMP gather. The results, displayed in Figure 8, are similar except that NMO correction makes the wavelet stretch out at far offset whereas our method does not. In this example, the method described in this paper gives good result because the signal-to-noise ratio of data is high. When noises are strong, NMO will give better result than does our method.

CONCLUSIONS

We have described an automatic dip-picking algorithm. The algorithm combines the linear and non-linear optimization procedures so that it is antialiasing, high resolution and high accuracy. Examples with synthetic and field data show several applications of the algorithm in seismic data processing. The disadvantage of the algorithm is its sensitiveness to noises. One possible way to attack this problem is to use absolute values to replace least squares in the objective function. An important extension of the algorithm is to allow multiple dips. The success of this extension will make the algorithm much more valuable than it is now.

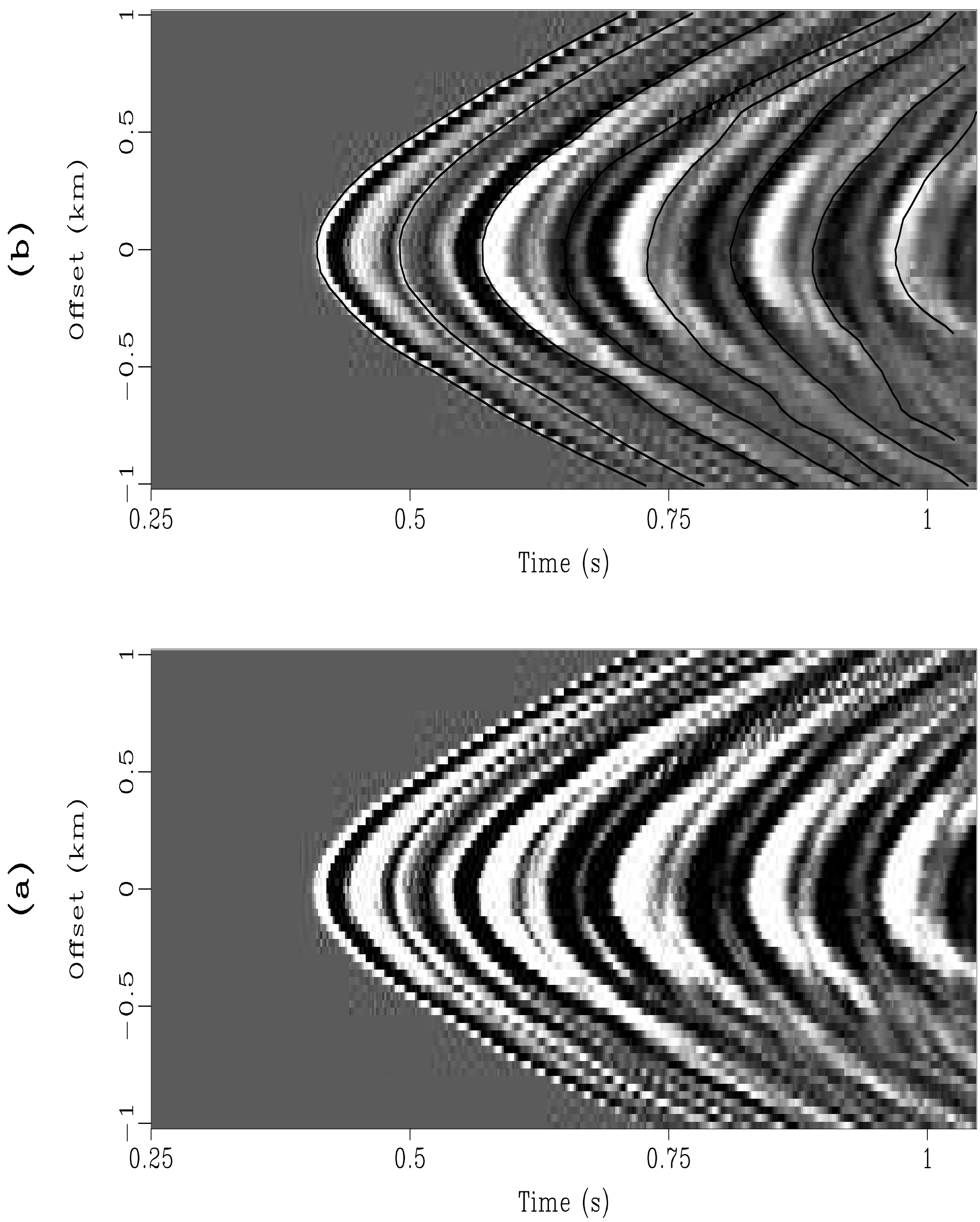

Figure 6: An example of event-picking with a field data: (a) a common receiver gather from a Walk-Away marine seismic survey; (b) the result of event-picking. $\boxed{\text{lin1-vsppick}}$ [NR]

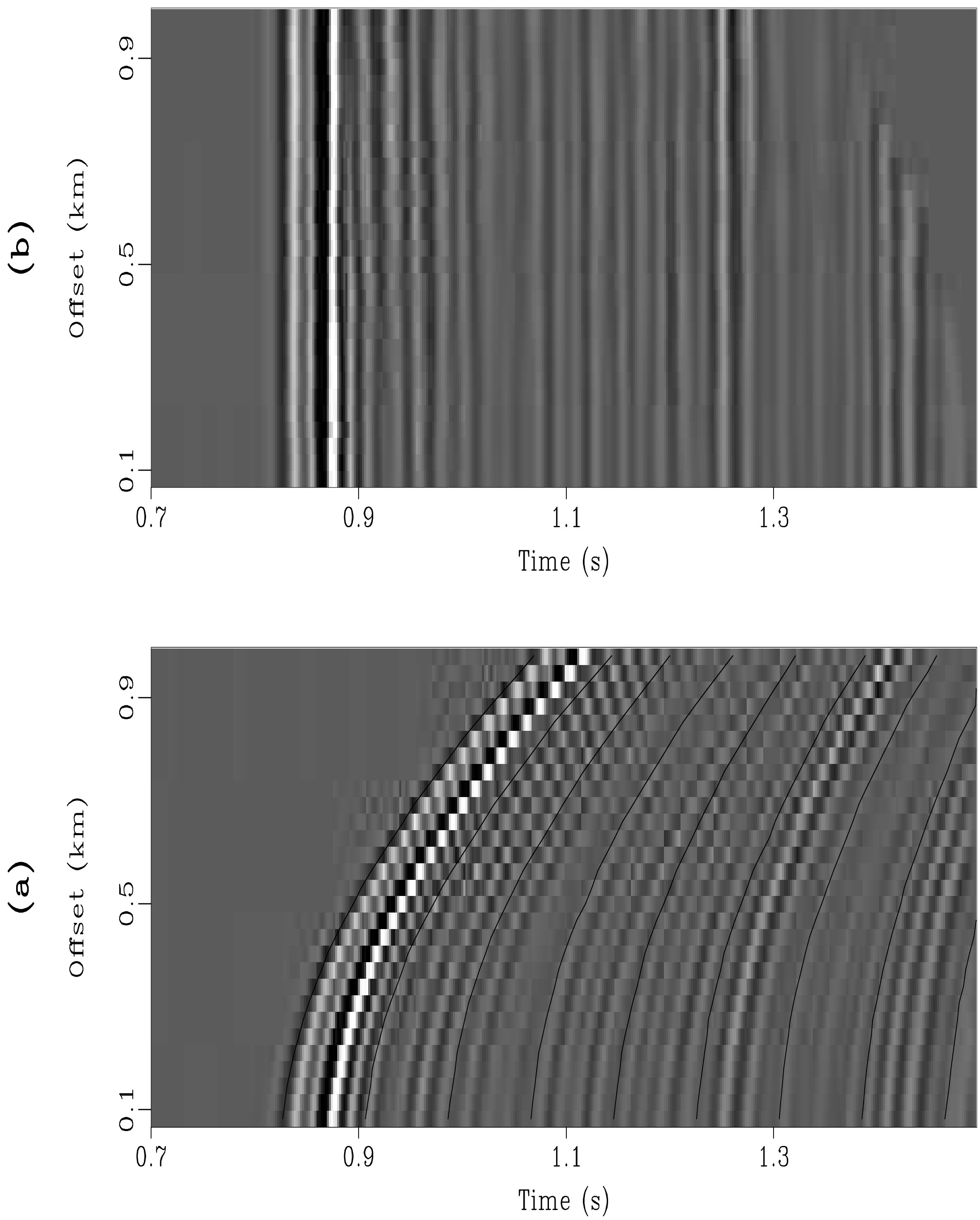

Figure 7: An example of moveout correction with a field data: (a) a CMP gather from a marine seismic survey; (b) the result of moveout correction. lin1-cmppick [NR]

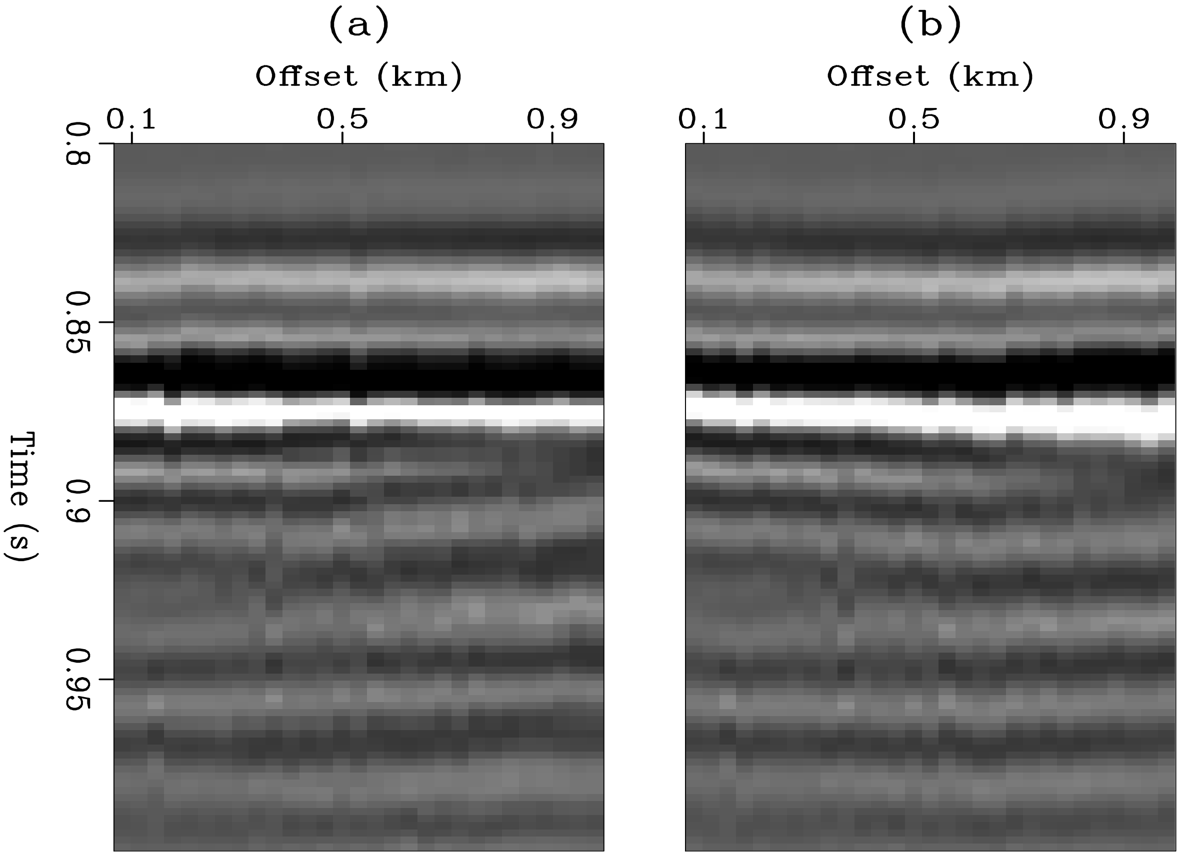

Figure 8: Comparison of two methods on moveout corrections: (a) our method; (b) NMO. lin1-cmpnmo [NR]

REFERENCES

Claerbout, J., 1990, Univariate problems: SEP–**65**, 216-219.

Nichols, D., 1990, Estimation of missing data by least squares: SEP–**65**, 271-293.