Effects of smoothing CRS stack attributes on inversion

I. Koglin

email: *Ingo.Koglin@gpi.uni-karlsruhe.de*

keywords: *inversion, macro-velocity model, migration, smoothing*

ABSTRACT

The aim of 2D inversion by means of traveltimes and common-reflection-surface stack attributes is to finally provide a 2D macro-velocity model of the subsurface. Here, the inversion is performed by a trace-by-trace and a layer-stripping method using common-reflection-surface stack attributes along identified (primary) reflection events. High frequency and low frequency fluctuations of the attributes along such events require a sophisticated smoothing method. Thus, the inversion performed after the smoothing becomes more robust.

INTRODUCTION

The aim of this work is to provide a more stable 2D inversion which additionally uses common-reflection-surface (CRS) stack attributes. Thus, the input for the inversion is smoothed with several different statistical methods. The 2D layered velocity model obtained by such a "robust" inversion serves, e. g., as macro-velocity model for a time or depth migration. I use a special inversion algorithm implementation of Majer (2000) based on the Hubral and Krey (1980) algorithm. This algorithm, namely the layer-stripping horizon inversion, takes as input traveltimes and data-derived CRS stack attributes (Jäger et al., 2001). It is compared with the conventional trace-by-trace Dix inversion algorithm (Dix, 1955). Each triplet of CRS stack attributes (α, R_{NIP}, R_{N}) determines a stacking surface to simulate a zero-offset (ZO) sample at a point $P(x_0, t_0)$ in the ZO section. Here, x_0 denotes the surface location in terms of the midpoint coordinate x where the normal ray emerges, and t_0 is the two-way ZO traveltime. The angle α is the emergence angle of the normal ray measured versus the surface normal. Two theoretical eigenwave experiments (Hubral, 1983) are associated with the radii of curvature R_{NIP} and R_{N}. R_{NIP} is the radius of the wavefront curvature at x_0 originating from a point source at the normal incidence point (NIP). This NIP is the endpoint of the normal ray in the depth domain. An exploding reflector experiment in the vicinity of the NIP yields the so-called normal wave emerging with radius R_{N} at x_0.

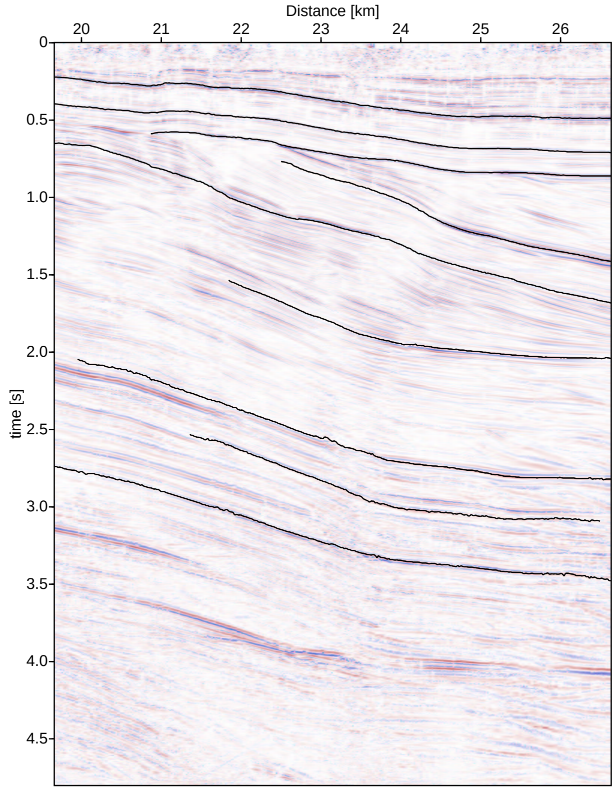


Figure 1: A real data subset with the identified primary reflections events (shown as black lines) which are used for the proposed inversion.

PICKING SEISMIC EVENTS

For the proposed inversion process, the CRS attributes have to be provided along identified (primary) reflection events. Thus, picking is necessary which was performed within the simulated ZO section of the *Initial Fresnel CRS stack* (Vieth, 2001). The Initial Fresnel CRS stack makes use of the first Fresnel zone. The first Fresnel zone is calculated by the CRS attributes and its projection to the surface serves as ZO aperture size for the stack. The resolution and continuity of reflection events within such a simulated ZO section is enhanced compared to the Initial CRS stack with a user-defined ZO aperture size. Thus, the semi-automatic picker can follow the maximum amplitudes from trace to trace easier. Interactive control is still necessary because reflection events do not have to be recorded at every receiver, intersect each other, or sometimes cannot be distinguished from noise by the picker. The semi-automatic picker follows the maximum amplitudes by centering a user-given window at the currently found maximum amplitude to search the maximum amplitude within this window in the next trace. The picker in its current implementation does not interpolate between samples to find the maximum which yields a smaller error because this error is negligible compared to the error obtained by picking the maximum amplitude. The maximum amplitude does, in general, not represent the true ZO two-way traveltime. However, picking the zero crossing is not as easy as picking the maximum amplitude because most source wavelets have one global maximum but many zero crossings. The CRS stack attributes are obtained by coherency analysis which is also more reliable at the maximum amplitudes than at the zero crossings. The error within the picked traveltimes will results in depth shifts of the inverted interfaces. This can be corrected before the inversion is performed if the source wavelet is known. An example of picked reflection events is shown in Figure 1 as black lines. The directly extracted CRS attribute "radius of NIP wavefront curvature", R_{NIP}, along one picked reflection event is shown in Figure 2(a).

SMOOTHING BY MEANS OF STATISTICAL METHODS

The obtained CRS attributes (α, R_{NIP}, R_{N}) extracted along the picked reflection events are not necessarily smooth, see Figure 2(a). As the subsequent inversion procedure is very sensitive on variations of the CRS attributes, smoothing the attributes prior to an inversion is required.

There are several filters available for smoothing purposes. For the following methods, the smoothing can be performed firstly with a pre-defined window of length $2n+1$ in time-direction (index τ) and secondly with a pre-defined window of length $2m+1$ in trace-direction (index ξ), i. e., along the identified reflection event:

1. The arithmetic mean, $\overline{x}$, is the normalized sum of all values ($x_{\tau,\xi}$) within the pre-defined window, see Figure 2(b).

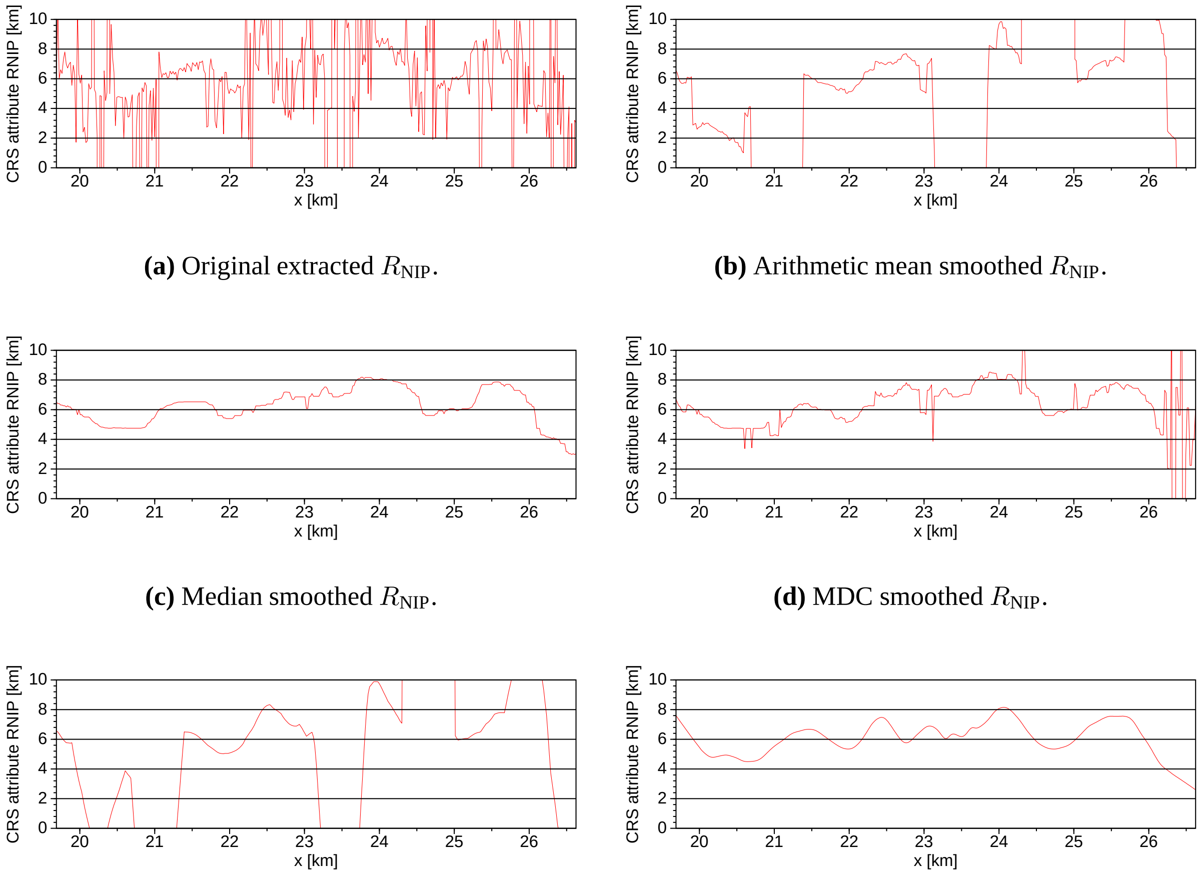
2. The median sorts the data of the window in an increasing order and takes the value in the middle of the sequence, see Figure 2(c).

3. A combination of the arithmetic mean and the median, called mean difference cut (MDC), calculates at first the arithmetic mean for the entire window. Then, the deviation of the values from the arithmetic mean is computed. If the deviation is greater than a given percentage, the value is excluded from further calculations. If data remain in this interval, the arithmetic mean is calculated again. In the case that no data are left in this interval. the median is taken as output, see Figure 2(d).

4. The weighted arithmetic mean, $\overline{x}_W$, is also a sum over all values ($x_{\tau,\xi}$) of the pre-defined window. Before the values are summed up, they are multiplied with a triangular weight function, see Figure 2(e).

Another smoothing algorithm is the robust locally weighted (RLW) regression of Cleveland (1979). The first step is to choose a weight function with the following properties:

- $W(x) > 0$ for $|x| < 1$,
- $W(-x) = W(x)$,
- $W(x)$ is a monotonously decreasing function for $x \geq 0$, and
- $W(x) = 0$ for $|x| \geq 1$.



(a) Original extracted R_{NIP}.

(b) Arithmetic mean smoothed R_{NIP}.

(c) Median smoothed R_{NIP}.

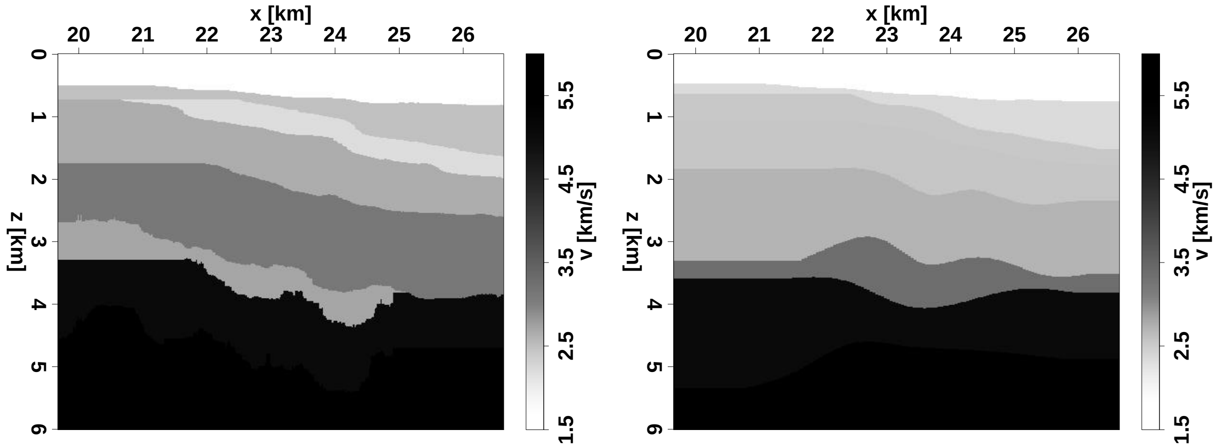
(d) MDC smoothed R_{NIP}.

(e) Weighted arithmetic mean smoothed R_{NIP}.

(f) RLW regression smoothed R_{NIP}.

Figure 2: Smoothing results obtained with a time window length of 7 samples and a trace window length of 37 samples: (a) R_{NIP} directly extracted at the picked ZO two-way traveltimes out of the entire attribute section. (b) after smoothing in time- and trace-direction with the arithmetic mean. (c) after smoothing in time- and trace-direction with the median. (d) after smoothing in time- and trace-direction with the MDC and a threshold of 5 percent. (e) after smoothing in time- and trace-direction with the weighted arithmetic mean and triangular weight function. (f) after smoothing only in trace-direction with the RLW regression and three iterations.

Examples of such a weight function are a boxcar, a triangle, or the cosine function (within the boundaries from $-\pi$ to $+\pi$). The second step is to fit a polynomial of dth order to the points (x_i, y_i) within the window using weighted least squares with weights $w_k(x_i)$. $i = 1, \ldots, n$ denotes all points of one picked reflection event, where n is the maximum number of points. $k = 1, \ldots, n$ represents the kth weight function corresponding to the ith point. This initial fit, $\hat{y}_i$, is the locally weighted regression. Now, the residual $(y_i - \hat{y}_i)$ is calculated to get a new weight function, δ_i, that has large weights for small residuals and small weights for large residuals. The fitted values are calculated again with a new set of weights, $\delta_i w_k(x_i)$, which are multiplied



(a) Dix inversion result.

(b) Horizon inversion result.

Figure 3: Inversion results using the RLW regression smoothed CRS attributes as input.

with the original data. The last step is repeated several times and the result is the robust locally weighted regression. The number of iterations is given by the parameter $nsteps$. The length of the smoothing window is obtained by $r = fn$, where r is rounded to the nearest integer neighbor and f is a factor between zero and one. If f is close to zero, the window length for smoothing is short. Thus, the curve of fitted points remains rough. If f gets closer to one, more points are considered for the fit. That leads to a smoother curve. This filter was designed to gain the best fit for data for which $y_i = g(x_i) + \epsilon_i$, where g is a smooth function and ϵ_i is a random variable with mean zero and constant scale. As it is obvious by comparing Figures from 2(b) to 2(f) against each other, the RLW regression is in favor.

INVERSION BY MEANS OF CRS ATTRIBUTES

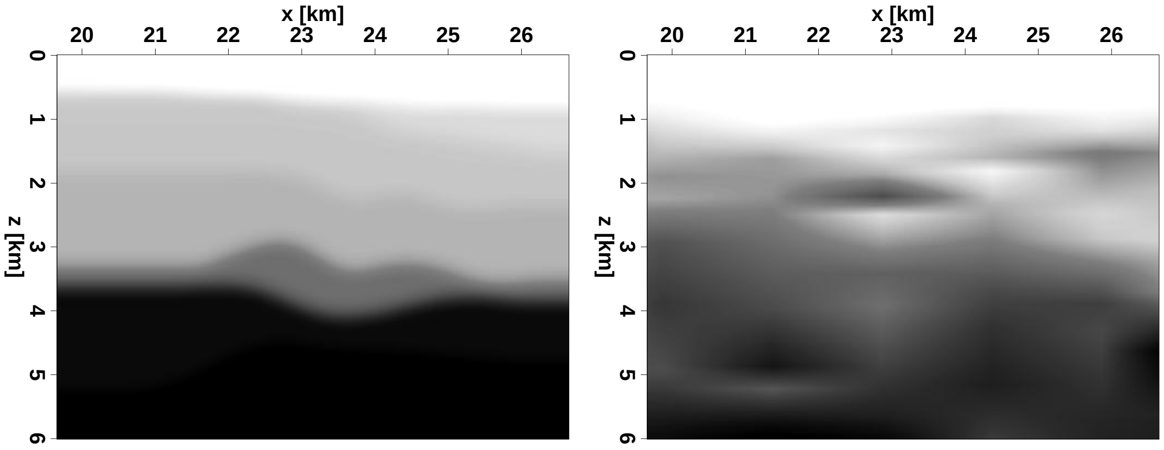
The frequently used Dix inversion algorithm uses the CRS attribute R_{NIP} to find the depth points along the corresponding vertical depth line. R_{NIP} is related to the stacking velocity as follows:

$$v_{stack}^2 = \frac{2v_0 R_{\text{NIP}}}{t_0 \cos^2\alpha} \tag{1}$$

The Dix inversion assumes that the subsurface consists of horizontal interfaces. Thus, $\alpha = 0$ and $\cos\alpha = 1$. The interval velocities can be calculated with Equation 1 recursively from top to bottom by

$$v_{i,j} = \sqrt{2v_0 \frac{R_{\text{NIP},i,j} - R_{\text{NIP},i,j-1}}{t_{i,j} - t_{i,j-1}}}\,. \tag{2}$$

This is done for all identified reflection events (index j) in one and the same trace (index i) and then for each trace. Thus, the Dix inversion is a trace-by-trace inversion method. After all traces



(a) Macro-velocity model from the horizon inversion result.

(b) Provided macro-velocity model.

Figure 4: Two velocity models: (a) Macro-velocity model after a Backus averaging of Figure 3(b). (b) Macro-velocity model obtained by interpolating between certain 1D interval velocities computed from several CMP stacks.

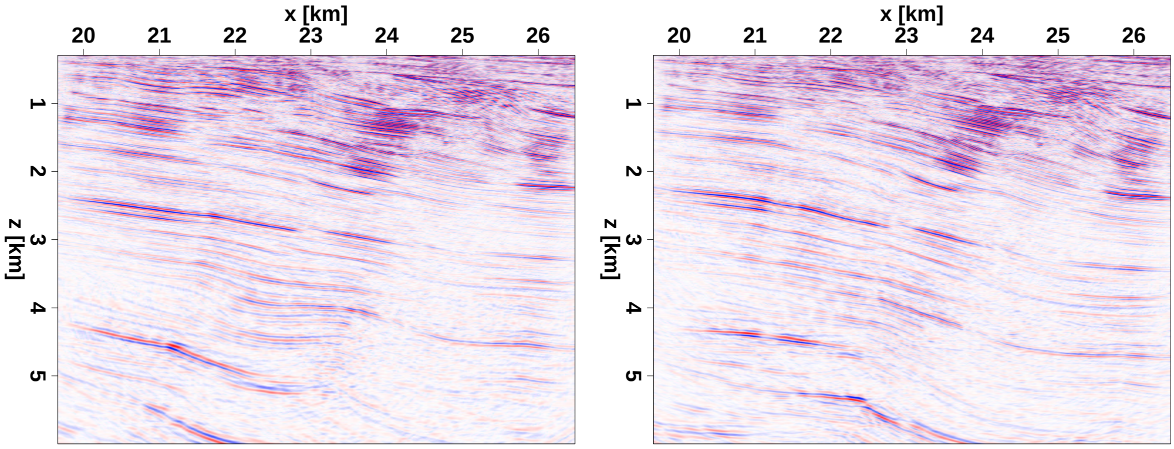
are processed, all interval velocities belonging to one layer are averaged to obtain the constant interval velocity related to this layer.

The horizon inversion (Majer, 2000) is a layer-stripping inversion method which means that all depth points of one interface are firstly inverted. Then, the next interface is inverted considering the results of the previously inverted interface. This inversion method uses the CRS attributes α and R_{NIP} to obtain the interface depth points of identified reflection events. Therefore, the CRS attribute α represents the direction of the ZO ray at its starting point measured against the surface normal. With the CRS attribute R_{NIP}, the end point of the ZO ray is found by applying the propagation law and the refraction law (Hubral and Krey, 1980) at shallower interfaces till the NIP wavefront shrinks to a point, i. e., $R_{\text{NIP}} = 0$. This point is the end point of the “back-propagated” ray and belongs to the searched for interface. The entire interface is constructed by spline approximation between all points of one interface. The constant layer velocity is, as for the Dix inversion, obtained by averaging all interval velocities of one layer. This constant layer velocity and the approximated interface are used for the calculation of the next interface.

Comparing the results of both inversion methods (Figure 3), it is obvious that the horizon inversion yields the smoother interfaces. Also the deviation of interval velocities from the calculated mean velocity of one layer is smaller for the horizon inversion than for the Dix inversion.

A DEPTH MIGRATION EXAMPLE

The horizon inversion provided an interval velocity model (Figure 3(b)) that must be transformed into a macro-velocity model for a subsequent Kirchhoff depth migration. Therefore, a Backus averaging (Backus, 1962) is performed to obtain the desired macro-velocity model (Figure 4(a)).



(a) Migrated image using the macro-velocity model obtained by the horizon inversion.

(b) Migrated image using the provided macro-velocity model.

Figure 5: Results of the Kirchhoff depth migration using the macro-velocity models obtained by (a) the horizon inversion using input smoothed by the RLW regression or (b) the macro-velocity model obtained by interpolating between certain 1D interval velocities computed from several CMP stacks.

The corresponding macro-velocity model of the real data subset is shown in comparison (Figure 4(b)). It is clear from the construction of interfaces by the horizon inversion that the current implementation of the horizon inversion algorithm cannot account for lateral velocity variations. A proposed solution can be to separate the input in smaller target ranges and merge the resulting velocity models. Or the layers should be build up by blocks of constant velocities or smoothly varying gradients.

A Kirchhoff depth migration of the simulated ZO section (Figure 1) using the velocity models of Figures 4(a) and 4(b) result in the migrated images of the subsurface shown in Figures 5(a) and 5(b), respectively.

As I do not know the true image of the subsurface, it is in the responsibility of the interpreter to decide which resulting image (Figure 5(a) or 5(b)) is the better representation of the subsurface. Please notice that I used more information of the subsurface to obtain directly a data-derived velocity model. The conventional CMP stack only uses ZO traveltimes and stacking velocities. The horizon inversion uses also ZO traveltimes but CRS attributes α and R_{NIP} in addition.

CONCLUSIONS

The presented processing steps (CRS stack – horizon inversion – Kirchhoff depth migration) provided an interpretable image of the subsurface. This image (Figure 5(a)) is purely data-derived. Especially, the velocity model is not obtained by iteratively improving an initial velocity model. Compared to velocity model obtained by CMP stacks, more information of the subsurface have entered the inversion process. I presented a sophisticated smoothing algorithm, the robust locally weighted regression. This algorithm is suited best to prepare the CRS stack attributes in order to ensure a stable inversion.

PUBLICATIONS

Detailed results are presented and discussed in Koglin (2001).

ACKNOWLEDGMENTS

The real data set was kindly provided by the oil and gas company BEB. This work was supported by the sponsors of the *Wave Inversion Technology (WIT) Consortium*, Karlsruhe, Germany.

REFERENCES

Backus, G. E. (1962). Long wave elastic anisotropy produced by horizontal layering. *Journal of Geophysical Research*, 67:4427–4441.

Cleveland, W. S. (1979). Robust locally weighted regression and smoothing scatterplots. *Journal of the American Statistical Association*, 74:829–836.

Dix, C. H. (1955). Seimic velocity from surface measurements. *Geophysics*, 20(1):68–86.

Hubral, P. (1983). Computing true amplitude reflections in a laterally inhomogeneous earth. *Geophysics*, 48(8):1051–1062.

Hubral, P. and Krey, T. (1980). *Interval velocities from seismic reflection time measurements*. Society of Exploration Geophysicists.

Jäger, R., Mann, J., Höcht, G., and Hubral, P. (2001). Common-Reflection-Surface Stack: Image and attributes. *Geophysics*, 66(1):97–109.

Koglin, I. (2001). Picking and Smoothing of Seismic Events and CRS Attributes, Application for Inversion. Master's thesis, Universität Karlsruhe, Germany.

Majer, P. (2000). Inversion of seismic parameters: Determination of the 2-D iso-velocity layer model. Master's thesis, Universität Karlsruhe, Germany.

Vieth, K.-U. (2001). *Kinematic wavefield attributes in seismic imaging*. PhD thesis, Universität Karlsruhe, Germany.