GEOPHYSICS, VOL. 74, NO. 6 (NOVEMBER-DECEMBER 2009); P. WCA65–WCA73, 15 FIGS.
10.1190/1.3242751

Simultaneous time imaging, velocity estimation, and multiple suppression using local event slopes

Dennis Cooke[1], Andrej Bóna[2], and Benn Hansen[3]

ABSTRACT

Starting with the double-square-root equation we derive expressions for a velocity-independent prestack time migration and for the associated migration velocity. We then use that velocity to identify multiples and suppress them as part of the imaging step. To describe our algorithm, workflow, and products, we use the terms velocity-independent and oriented. While velocity-independent imaging does not require an input migration velocity, it does require input p-values (also called local event slopes) measured in both the shot and receiver domains. There are many possible methods of calculating these required input p-values, perhaps the simplest is to compute the ratio of instantaneous spatial frequency to instantaneous temporal frequency. Using a synthetic data set rich in multiples, we test the oriented algorithm and generate migrated prestack gathers, the oriented migration velocity field, and stacked migrations. We use oriented migration velocities for prestack multiple suppression. Without this multiple suppression step, the velocity-independent migration is inferior to a conventional Kirchhoff migration because the oriented migration will flatten primaries and multiples alike in the common image domain. With this multiple suppression step, the velocity-independent are very similar to a Kirchhoff migration generated using the known migration velocity of this test data set.

INTRODUCTION

In conventional prestack time migration, the input data are sorted in just one domain (shot gathers, offset gathers, or CMP gathers) and the user must supply a migration velocity field. With the migration and multiple suppression technique we present here, no input-velocity information is required, but in lieu of input velocities, the user needs to supply information about the prestack local event slopes (or p-values) of the input data in two domains; the shot domain and receiver domain. With the local event slopes from shot and receiver gathers, we perform a velocity-independent migration and solve for the associated migration velocity. Then those velocities are used to discriminate between primaries and multiples and to suppress the multiples.

Another way to state this requirement for input p-values is to say that our input data are $A = f(x,t,p)$. It is interesting to compare this requirement for $A = f(x,t,p)$ to both Kirchhoff and Stolt-FK migration algorithms (Stolt, 1978; Yilmaz, 2001). In Kirchhoff migration, the input data are $A = f(x,t)$ but there is no need for p-value information. In contrast, with F-K migration, the imaging is a coordinate translation of each point based on the ratio of K and F coordinates (or p-value) but there is no access to (x,t) information. The requirement for both (x,t) and p-value information is significant. There are various methods to obtain the p-values in the time-space domain, but there are questions of cost, resolution, and sensitivity to noise that go beyond the scope of this paper. However, we will discuss a simple and inexpensive way to obtain p-values (at least for our random noise-free synthetic data set) and mention techniques other authors use to obtain p-values.

Ottolini (1983) presents the concept of using local slopes for what he terms "velocity-independent imaging." Fomel (2007) extends the ideas of Ottolini and presents a method similar to what we develop here. He derives his results from reflection geometric principles and uses local slope information from the midpoint and offset domains. He demonstrates results for both simple and field data sets, and derives an expression for migration velocity. He does not discuss using that velocity for multiple suppression. He uses the term velocity-independent imaging in reference to Ottolini (1983) and "oriented" to refer to attributes generated using local slope information.

In contrast to Fomel (2007), we derive our imaging and velocity expressions from the double-square-root equation and work with shot and receiver gathers. Like Fomel, we present a 2D solution, but a 3D solution is presented in Bóna and Cooke (2009). We arrive at

Manuscript received by the Editor 30 December 2008; revised manuscript received 30 May 2009; published online 15 December 2009.
[1]Santos Ltd., Adelaide, Australia. E-mail: dennis.cooke@santos.com.
[2]Curtin University of Technology, Perth, Australia. E-mail: a.bona@curtin.edu.au.
[3]Formerly Santos Ltd., Adelaide, Australia; presently Hess, London, U. K. E-mail: benn.hansen@gmail.com.

the same result as Fomel (2007), taking into account the transformation from shot-receiver coordinates to midpoint-offset coordinates. The main distinction between Fomel's work and what is presented here is the use of oriented velocities to identify and suppress multiples. The synthetic data set used here is designed to be richer in multiples than Fomel's real data set.

Billette and Lambaré (1998) and Lambaré (2006) develop a somewhat similar approach they call stereotomography. Although both stereotomography and our method require local-event-slope information about a reflection in both shot and receiver domains, there is an important difference between stereotomography and what we and Fomel develop: Stereotomography is an iterative nonlinear solution for an interval velocity model, although we and Fomel (2007) derive analytical expressions for migration and migration velocity. Oriented migration velocity is an average over all layers; thus oriented migration is a time migration. Stereotomography solves for an interval velocity model and thus is appropriate for a depth migration.

Hua and McMechan (2003) also use p-values from the shot and receiver domains. They estimate p-values using a local slant stack and use the results for parsimonious prestack Kirchhoff depth migration. Their migration is considerably faster than a conventional Kirchhoff migration, but it results in some degradation of image quality.

ORIENTED ATTRIBUTES: THEORY

The double-square-root equation

$$t = \sqrt{t_0^2 + \frac{(x_r - x_m)^2}{V^2}} + \sqrt{t_0^2 + \frac{(x_s - x_m)^2}{V^2}} \qquad (1)$$

describes the traveltime of seismic reflections and diffractions in a homogeneous medium (Claerbout, 1985). In this equation, t is traveltime, V is velocity, t_0 is vertical traveltime to the reflector/ diffractor and x_m, x_s, and x_r are the horizontal coordinates of the migrated reflection point, source location, and receiver location. For all (x,t) points in a shot or receiver gather, t, x_s, and x_r are known and V, t_0 (or equivalently z) and x_m are unknowns. We will solve for those unknowns below.

We denote the partial derivative of equation 1 with respect to x_r by p_s (this is the p-value measured in a common-shot gather):

$$p_s(x,t) = \frac{\partial t}{\partial x_r} = \frac{x_r - x_m}{V^2\sqrt{t_0^2 + (x_r - x_m)^2/V^2}}, \qquad (2)$$

and the partial derivative of equation 1 with respect to x_s is defined by p_r (this is p measured in a common-receiver gather):

$$p_r(x,t) = \frac{\partial t}{\partial x_s} = \frac{x_s - x_m}{V^2\sqrt{t_0^2 + (x_s - x_m)^2/V^2}}. \qquad (3)$$

As we discuss later, p_s and p_r can be measured for every sample in shot and receiver gathers, respectively. Once p_s and p_r are known, equations 1–3 represent a system of three equations with three unknowns — x_m V, and t_0. The solution of these equations for the unknowns is derived in Appendix A, and is given by

$$x_m = x_s - dp_s\frac{t - dp_r}{t(p_r - p_s) + 2dp_sp_r}, \qquad (4)$$

$$V^2 = \frac{x_s - x_m}{tp_s} + \frac{x_r - x_m}{tp_r}, \qquad (5)$$

and

$$t_0 = \frac{|x_s - x_m|}{V}\frac{\sqrt{1/V^2 - p_s^2}}{|p_s|}. \qquad (6)$$

Here, $d = x_r - x_s$ is the signed source-receiver offset. We note that equations 4–6 above are equivalent to equations 18–20 in Fomel (2007) transformed from midpoint-offset coordinates to shot-receiver coordinates.

IMPLEMENTING THE ORIENTED ALGORITHM

The oriented algorithm requires input p_s and p_r values for each sample of every trace; ideally, it needs a shot at every receiver location and a receiver at every shot location (called "reciprocal shot-receiver geometry" here). If the acquisition geometry does not supply this, the first step in implementing the oriented algorithm is to interpolate the required traces. Although this is a trivial task for 2D seismic data, it can be a challenge for 3D seismic data. Surface-related multiple elimination (SRME) requires this same reciprocal shot-receiver geometry, and there is a body of literature discussing interpolation procedures to obtain it (Baumstein and Hadidi, 2004; Matson and Abma, 2005; Matson and Zhang, 2007).

Once the reciprocal shot-receiver geometry is obtained, the workflow is:

1) Sort prestack data by common shot. Calculate and save p_s for all points in this shot gather.
2) Sort prestack data by common receiver. Calculate and save p_r for all points in the receiver gather.
3) For each sample on each input trace, time t is known, p_s and p_r are available from the previous two steps, and x_s and x_r are known from the acquisition geometry. Use equations 4–6 to solve for x_m, V, and t_0.

In Kirchhoff imaging, reflection amplitudes recorded for a given source-receiver pair are distributed over an ellipse defined by source-receiver distance and migration velocity. The kinematics of Kirchhoff imaging will copy the input amplitude to many different output locations and relies on constructive and destructive interference to construct an image of the reflector.[4] Oriented imaging differs from Kirchhoff imaging in that (1) it does not require an input migration velocity; (2) it requires the measurement of prestack horizontal slowness at both the source and receiver; and (3) one input amplitude will be copied to just one output location with no amplitude correction factors (in the version presented here).

MEASURING THE LOCAL EVENT SLOPES

In conventional NMO velocity analysis of a CMP gather, the seismic processor is required to interpret or pick a velocity for between five and fifteen reflectors on each CMP analyzed. Similarly, in the stereotomography workflow of Lambaré (2006), the processor picks or interprets p-values at multiple source-receiver offsets for each reflector in both shot and receiver domains. Fomel (2007) takes a different approach to measuring p-values than Billette and Lambaré

[4]Kirchhoff time imaging also includes amplitude and phase factors applied to those amplitudes as per Yilmaz (2001).

(1998). He defines the local event slope as a data attribute that best represents the p-value at every point in a prestack gather. Fomel (2002) measures local event slopes using a process called plane-wave destruction. We take a similar approach to that of Fomel: we do not require the user to pick or interpret p-values. Instead, we calculate p_r and p_s attributes, not using plane-wave destruction, but with a simplistic instantaneous p-value approach via the equation (Barnes, 1996)

$$p_{\text{inst}} = k/f.$$

In this equation, k is instantaneous spatial frequency and f is instantaneous temporal frequency, where instantaneous frequency is defined by Taner et al. (1979). When generating p, phase unwrapping and/or some temporal and spatial smoothing are needed. Note that this instantaneous p-value approach cannot deal with two or more local event slopes that exist at the same (x,t) point. Examples of our p_{inst} values are shown and discussed below. We note that any of these three methods of obtaining p-values (picking, plane-wave destructors, and p_{inst}) could be used with Billette and Lambaré's stereotomography or velocity-independent migration.

APPLICATION TO SYNTHETIC DATA

Next we apply the oriented algorithm to a synthetic 2D data set and display the output migration, velocity-analysis, and multiple-suppression results. Later, we offer a combined discussion of all results. Figure 1 shows a velocity model used to generate acoustic finite-difference shot gathers in Seismic Un*x. For brevity, we do not show the corresponding density model. Figure 2 shows one of the 200 shot gathers generated from this velocity model. Because of the high impedance contrasts in this model, the resultant shot gathers show strong water-bottom and interbed multiples, diffractions, and diffracted multiples.

No random noise has been added to this model. Figure 3 shows the p_s values from this shot in color, displayed with the input shot gather underneath in grayscale. As discussed above, the p-values were calculated by the simple ratio of spatial instantaneous frequency divided by temporal instantaneous frequency. For each (x,t) point in this shot, there is a corresponding p_r value, which we do not show here. With these p_r and p_s values, along with the shot and receiver coordinates and the arrival time of each sample, equations 4–6 are applied to calculate the basic oriented attributes: the location of the migrated data and the migration velocity. The resulting attributes are shown in Figures 4–6. In these figures, the attributes are not calculated for all traces because some traces near the start of the line did not have reciprocal shot-receiver paths.

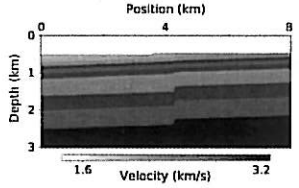

Figure 1. Velocity model used to generate finite-difference shot gathers as in Figure 2.

Given the information in the attributes of Figures 4–6, the oriented migration consists of a simple resorting or mapping of the data from shot gathers, where seismic amplitudes are a function of arrival time and receiver location, to common-image gathers, where amplitudes are sorted by the x_m and t_0 attributes. The output of this remapping is an oriented image gather. An example is shown in Figure 7. The horizontal coordinate in Figure 7 corresponds to source-receiver offset, which facilitates comparison with a Kirchhoff migration, below. An alternative gather would sort the migrated gather by t_0 and reflection angle. An estimate of the reflection angle is readily available with oriented analysis as p_s, p_r, and oriented velocity V (an average velocity) are known already. That reflection angle is

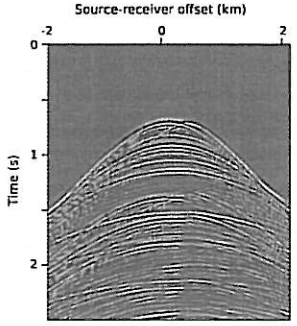

Figure 2. Example finite-difference shot gather generated from the velocity model of Figure 1. There are only nine primary reflections in this model. Much of the energy shown here comes from water-bottom and interbed multiples, diffractions, and multiple diffractions.

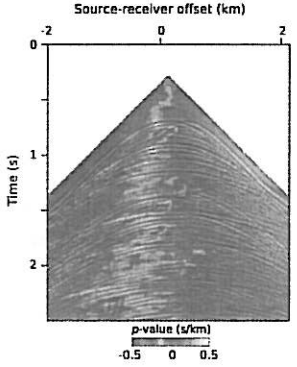

Figure 3. Shown are p-values (in color) calculated from Figure 2. Input shot is posted in grayscale.

$$\theta = \frac{1}{2}(\arccos(p_r V) - \arccos(p_s V)).$$

Fomel (2007) derives a velocity-independent solution for this angle:

$$\sin^2(\theta) = (x_r - x_s)(p_r - p_s)/(2t).$$

Prestack comparison of Kirchhoff migration and oriented migration

Figure 8 shows a gather from a prestack Kirchhoff migration of the same input data set and location shown in Figure 7. This Kirchhoff migration is actually a prestack depth migration (PSDM) with the depth axis converted back to time. A depth migration was used as a standard of comparison here to ensure that primaries (and not multiples) were imaged properly. This was guaranteed by using a PSDM with the input migration velocities taken straight from Figure 1. There is little lateral velocity variation in the velocity model of Figure 1, thus this depth migration and the optimal time migration should be quite similar. The most striking difference between the Kirchhoff gather and the oriented gather is treatment of multiples: On the Kirchhoff gather, the primary reflections (annotated with arrows) are flat although the many multiple reflections still have residual moveout. In contrast, in the oriented gather in Figure 7, all primary and multiple reflections are flattened. This appears to present a problem with the oriented workflow because those flattened multiples will not be removed with a Radon filter or stacking across offset — the two techniques used most commonly to suppress multiples. This potentially significant problem with multiples can be overcome using oriented velocities.

Figure 9 illustrates how oriented velocities can be used to suppress multiples. The left-hand panel shows the oriented velocity at-

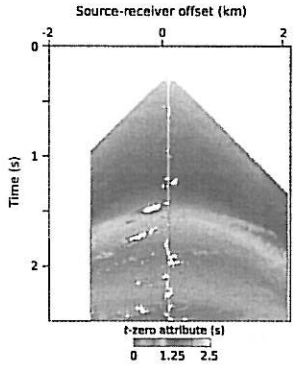

Figure 4. Oriented t-zero attribute for the shot gather in Figure 2.

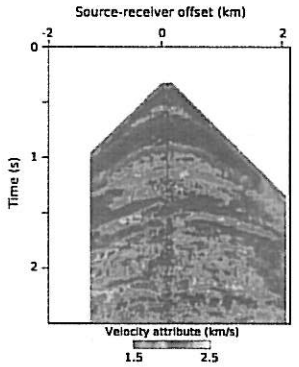

Figure 5. Oriented velocity attribute for the shot gather in Figure 2.

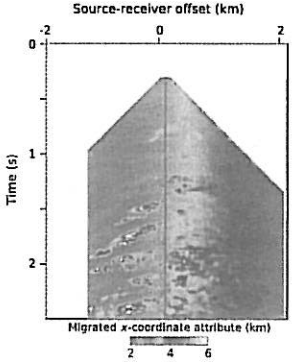

Figure 6. Oriented migrated x-coordinate attribute for the shot gather in Figure 2.

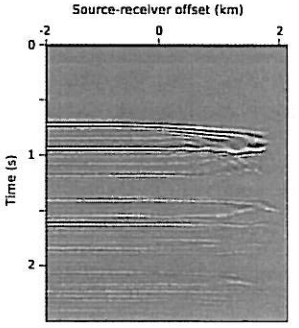

Figure 7. Oriented migration after sorting to a common-image gather (CIG). There is no velocity filtering applied here.

tribute calculated with equation 5 after those velocities have been mapped to a common-image gather. The expected velocity difference between primary and multiple reflections can be observed in this panel. Generally, the yellower and redder colors correspond to fast primaries and the bluer colors correspond to slower multiples. The middle panel shows the corresponding prevelocity-filtered image gather. The rightmost panel is a version of the middle panel wherein all reflections slower than the predefined velocity cutoff are not mapped from the shot gather into the image gather. Thus, the oriented multiple suppression is just a migration with a velocity cutoff filter applied. To implement this multiple-suppression scheme, the user must review the oriented velocities and interpret a cutoff between primary and multiple velocities. In the section below, we show how to form this velocity cutoff.

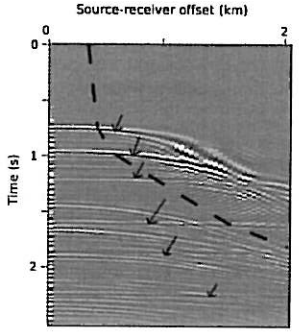

Figure 8. Kirchhoff CIG at same location as for Figure 7. Primary reflections are marked with arrows. A far-offset mute is marked with the dashed line.

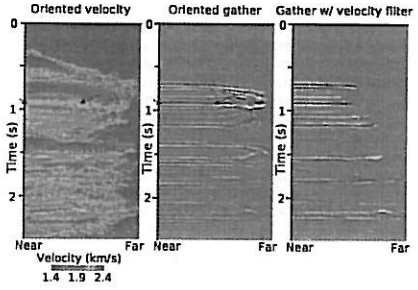

Figure 9. Oriented velocities in CIG domain (left); corresponding oriented CIG gather without velocity filtering (middle) and with velocity filtering (right). Arrows indicate the location of the reflector for which the velocity graph in Figure 14 is extracted.

Poststack comparison between conventional Kirchhoff migration and oriented migration

The migrated gathers discussed in the previous sections are stacked to generate 2D sections, and the results are shown in Figures 10–12. Figure 10 shows the stack of the oriented gathers before any multiple suppression via oriented velocity filtering. Figure 11 shows the stack of the velocity-filtered gathers (such as in Figure 9). For comparison, Figure 12 shows the stack of the Kirchhoff gathers. Each of the three sets of gathers had the same far-offset mute applied before stacking. Figure 13 shows a stacked version of the oriented-velocity attribute. On the left is the oriented-velocity attribute (as in Figure 9, left panel) after applying a far-offset mute and then averaging across offset. On the right side are the same velocities displayed with the stack of Figure 10 and a line plot (in blue) of the oriented velocities at a single location. This line plot shows that some of the reflections are imaged by high velocities and others by slow velocities. As can be anticipated from the velocity model of Figure 1, the fast reflections are primaries and the slow reflections are multiples. The velocity cutoff used to generate the right panel of Figure 9 and the stack of Figure 11 is defined by the manually interpreted red line, which connects the fast or primary reflections. In the case presented here, data points with an oriented velocity more than 200 m/s slower than this red line were excluded when mapping from shot gathers to common image gathers (CIGs).

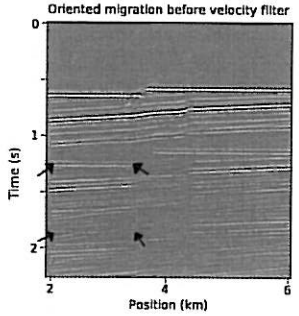

Figure 10. Oriented migration and stack with no velocity filter. The middle panel of Figure 9 shows a typical input gather.

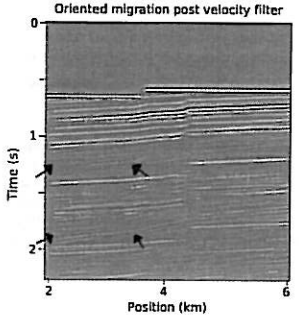

Figure 11. Oriented migration and stack after application of multiple suppression via a velocity filter. The right panel of Figure 9 shows a typical input gather.

Figure 12 shows a stack of the Kirchhoff gathers. The difference between the Kirchhoff stack and oriented stack of Figure 10 is considerable because of the lack of multiple suppression in the initial oriented stack of Figure 10. As discussed previously, the multiples in the oriented image gather are flattened and will not stack out as they do for the Kirchhoff gathers. To deal with this, our oriented multiple-suppression technique discriminates against those multiples based on their oriented velocity attribute; the results are shown in Figure 11. As annotated by the arrows in Figures 11 and 12, in some places the oriented multiple suppression is superior, although the Kirchhoff multiple suppression is better in others. This variation in the effectiveness of oriented multiple suppression probably results from the appropriateness of the 200 m/s mentioned above.

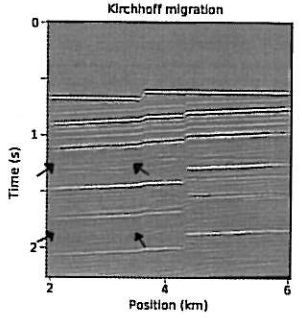

Figure 12. Stack of Kirchhoff gathers. Figure 8 shows a typical input gather.

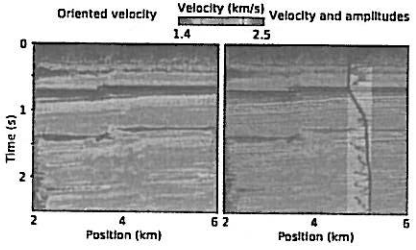

Figure 13. Poststack view of oriented velocities. The left side shows oriented velocities. The right side is a combined display of oriented migration with no multiple suppression (in grayscale), velocities (in color), line plot of the velocities at a single location (blue line), and manually interpreted primary velocity function (red line). All data with an oriented velocity 200 m/s slower than this red line were filtered in forming the stack of Figure 11.

DISCUSSION

We have applied the oriented algorithm to the problems of imaging, velocity analysis, and multiple suppression. Next we discuss the quality, cost, and value of the above results and attempt a comparison with more traditional seismic processing workflows designed to solve the same problems.

Calculation of the p-values

The three oriented processing products (velocity-independent imaging, oriented velocity analysis, and oriented multiple suppression) carry the same one-time cost of computing p-values for every input (x,t) point in every shot and receiver gather. Here we have calculated those p-values using a single-valued or instantaneous method (discussed above) that is very inexpensive; computation time is 1.6 s per shot gather (1250 time points and 200 traces, single CPU).

Details in Figures 2 and 3 illustrate an important limitation of our instantaneous p-value approach. Where two local event slopes are present on the input shot, our current approach solves for the dominant p-value and ignores the weaker one(s). If that weaker arrival is diffraction, our current implementation of oriented imaging will not image that diffraction. Another possible limitation of our instantaneous p-value approach is its stability in the presence of noise. Here we have stabilized the p-value calculations with a (15×15 point) 2D smoother, but we have not explored the issue of stability in the presence of noise. Our smoother is rather small compared to the smoother used on traditional NMO and migration-velocity-analysis results (one to two cable lengths or up to hundreds of traces). We keep our p-value smoother short because we wish to minimize the averaging of adjacent primaries and multiples in oriented multiple suppression. There are other methods to calculate p-values, i.e., curvelets (Douma and de Hoop, 2007) or local slant stacks (Billette and Lambaré, 1998). Schleicher et al. (2009) give a recent and thorough discussion of the topic. Although a full exploration and comparison of the different methods of calculating p-values and their performance in the presence of noise is beyond the scope of this paper, note that Fomel (2002) uses plane-wave destruction filters to compute p-values in the presence of noise and shows how to compute multivalued p-values. Cost-wise, we estimate the plane-wave destruction method to be similar to or faster than a Radon transform.

Velocities

Perhaps the most expensive step in traditional seismic processing workflows is NMO and migration velocity analysis. This expense arises not from computational needs, but from the manpower-intensive process of interpreting and picking velocities. Current prestack time-migration velocity analyses use some variation of the following iterative workflow (Yilmaz, 2001):

- Interpret, pick, and smooth the NMO velocity field (this picking can be automated for some data sets).
- Scale this NMO velocity field by a factor that might vary between 0.85 and 0.98. The purpose of the scalar is to remove the 1/cos(dip) effect. Choosing this scalar is another manual interpretation step.

- Perform prestack time migration with this velocity field.
- Inspect resultant the image gathers for residual normal moveout (RNMO). If RNMO is present, update the current velocity field and go to step 3. If RNMO is not present, then the current migration velocities are correct.

The above iterative approach is required because the input NMO velocities are flawed: they are in the unmigrated location and are faster than true velocity (for a homogeneous medium) by a factor of 1/cos(reflector dip). Dip moveout (Yilmaz (2001)) can remove the 1/cos(dip) problem, but it will not put the velocities in the correct spatial location. Thus it is used infrequently in modern 3D migration-velocity-analysis workflows. In comparison, the oriented velocity field will be in the proper migrated location and the velocities will not be scaled by the reflector dip. The need for expensive velocity interpretation iterations is greatly reduced with the oriented method, but as mentioned below, oriented multiple suppression will need a velocity interpretation step.

What sort of average are the oriented velocities? For a flat-dip, layer-cake earth, the oriented velocity would be equal to the stacking or NMO velocity because for this simple case, a stack and a migration are the same operation. Taner and Koehler (1969) showed that the optimal stacking velocity for this layer-cake earth is an rms average. For both NMO and oriented velocities, this is only true for short shot-receiver offsets.

Obviously, the input p-values are susceptible to measurement errors in the presence of noise. To what degree does this impact the output oriented velocities? Figure 14 details velocity as a function of offset taken from a single reflector in Figure 9 and shows velocity oscillations at the far offsets. A close inspection of Figures 2 and 3 shows the cause of these velocity oscillations. In these figures, the reflector detailed in Figure 14 has a zero-offset apex just under 1 s. At a source-receiver offset of approximately -1.5 km and greater, this reflection is overlain by direct arrivals and the p-values oscillate. These oscillations in input p-value translate into oscillations in output velocity. The magnitude of the velocity oscillations for this reflector is 1.0% to 2.5%. The left panel of Figure 9 shows an even stronger example of these velocity oscillations at slightly more than 1.5 s, which can be related to the interference between this primary and the first water-bottom reflection. There are three different strategies to minimize these velocity oscillations: (1) smooth the input p-values, (2) stack the velocities in Figure 9 to produce Figure 13, or (3) extract multivalued p-values. Multivalued p-values are not used here, but examples can be found in Fomel (2002).

Another potential error with the oriented method concerns application to nonhyperbolic data. The double-square-root equation 1 assumes a homogeneous velocity field and thus the reflections described by it are hyperbolic. Consequently, equations 4–6 will have an error when applied to nonhyperbolic data. To examine this error, we apply the oriented method to nonhyperbolic traveltimes generated by the following equation from Alkhalifah and Tsvankin (1995):

$$t^2 \approx t_0^2\left[1+\left(\frac{x}{t_0V}\right)^2-2\eta\frac{(x/t_0V)^4}{1+(1+2\eta)(x/t_0V)^2}\right], \quad (7)$$

where $\eta \equiv (\varepsilon-\delta)/(1+2\delta)$.

In the above equation, ε and δ are parameters that characterize velocity anisotropy. We generate synthetic arrival times (raw traveltimes are not shown here) using the parameters $t_0 = 2$ s, $V = 2.2$ km/s, $\varepsilon = 0.15$, and $\delta = 0.10$ where the source-receiver offset x varies from 0 to 4 km. For these parameters, the far-offset reflection has an angle of incidence of about 42°. Figure 15 shows the results of applying the oriented method to these arrival times and compares the results to two-term NMO. This figure shows that the oriented method, although not exact, is superior to a two-term NMO equation in flattening nonhyperbolic traveltimes. Figures 14 and 7 also hint at this. The velocities in Figure 14 are increasing with offset and not constant as expected for hyperbolic reflections. Compared to a homogeneous velocity field, the strong vertical velocity gradient in our input velocity model (see Figure 1) leads to earlier far-offset traveltimes, flatter p-values (that is closer to 0), and imaging velocities generally increasing with offset. Those p-values and velocities are used by equation 6 to flatten the data as part of the imaging process. The result (Figure 7) shows very little residual NMO.

Multiple suppression

The oriented method has the drawback that it flattens all primary and multiple reflections and thus the slower multiples will not stack out as with a conventional migration and stack. When using the oriented method, it is incumbent upon the user to review the output velocity attribute (prestack and/or poststack) and decide if the presence of multiples is indicated by that velocity field. If multiples are

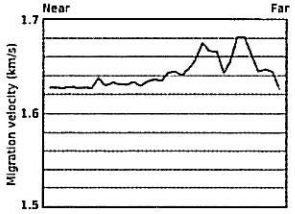

Figure 14. Line plot of the oriented velocities extracted from Figure 9 at the reflector annotated by the arrows. The far-offset displayed here is located at the right-hand arrow, middle panel of Figure 9, which is approximately at the critical angle for this shallow reflector.

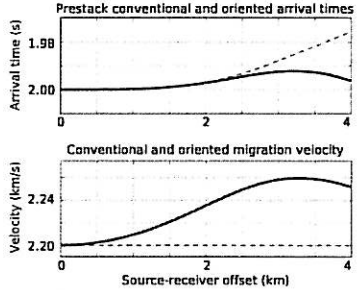

Figure 15. Top plot: anisotropic traveltimes flattened with the oriented approach (bold line) and flattened with two-term NMO (lighter line). Bottom plot: oriented velocities from equation 5 (bold line) and NMO velocity (lighter line).

present, the user needs to interpret and pick a velocity cutoff separating primaries and multiples, then remap the data from shot domain to the image gathers excluding arrivals slower than the cutoff. This velocity interpretation and picking step is a cost burden added by using the oriented workflow. At first glance, this added interpretation cost is very similar to the cost discussed above that the oriented method avoids — that is, interpretation costs associated with conventional velocity analysis. However, the oriented workflow has two significant advantages over a conventional migration-velocity-analysis workflow. First, the oriented work flow avoids the iterations required by the conventional velocity-interpretation workflow. Second, the oriented workflow performs multiple suppression prestack although the conventional workflow can only suppress multiples upon stacking. Conventional workflows need to use additional steps such as SRME or Radon transforms to suppress multiples prestack.

When judging the potential value of the oriented method, it is interesting to compare it to the Radon multiple method of Foster and Mosher (1992). The Radon transform is one of the best tools for removing multiples and organized noise before stack. A restriction of the Radon transform is that the primary and multiple arrivals modeled need to have their apex at zero source-receiver offset. Diffracted multiples, ship noise, and out-of-plane reflections are examples of organized noise that frequently do not have their apex at CMP source-receiver offset = 0. Thus, the Radon transform has problems modeling and removing these types of noise. The oriented method does not have this zero-offset apex restriction so it is able to model and remove these noise types. The synthetic data set used here has multiples and diffracted multiples with apexes at nonzero offsets that were suppressed successfully with the oriented method.

Imaging

The cost of velocity-independent imaging is far less than a traditional Kirchhoff time migration because it is just a one-to-one mapping of amplitudes from the shot gathers to common image gathers based on the t_0 and x_m oriented attributes. For the data set used here, the velocity-independent imaging computation time was 30 s and the Kirchhoff migration time was 8 min. Redoing the velocity-independent migration with a different migration velocity cutoff for multiple suppression does not require that p-values be recalculated, but redoing the Kirchhoff migration with a different velocity field does require a recalculation of the migration operators. A comparison of Figures 11 and 12 demonstrates the quality of velocity-independent time migration. The velocity-independent image quality is comparable to the Kirchhoff migration of Figure 12 in most respects, but it is inferior at and near the faults. We strongly suspect this imaging problem is caused by our above-mentioned instantaneous p-value calculation approach that cannot deal with crossing reflections and diffractions in the input shot and receiver gathers. Fomel (2002) discusses the theory and shows examples of solving for multivalued p-values.

CONCLUSIONS

We have proposed and evaluated a prestack time migration that is velocity independent. We call this oriented migration in reference to similar work by previous authors. Our oriented algorithm requires input p-values from both the shot and receiver domains and will output an imaging velocity along with the migrated image. The velocity-independent nature of this algorithm has the drawback of imaging undesired multiple reflections along with the desired primary reflections. We have shown how to identify and remove these multiples by using the oriented imaging velocity. Using synthetic shot gathers rich in multiples, we have shown that the quality of the oriented imaging (when multiple suppression is included) is comparable to a traditional Kirchhoff migration. In this comparison, the Kirchhoff migration had access to the model velocities although the oriented migration did not.

We have discussed the cost and value of oriented imaging, oriented velocity analysis, and oriented multiple suppression when compared to more traditional seismic approaches. Our conclusion is that the true value of the oriented approach comes from the velocity analysis and multiple suppression. Oriented velocity analysis avoids the interpretive and expensive iterations used in traditional time-migration velocity analysis. Oriented multiple suppression can be done at the same time as the velocity analysis because it is based on those oriented velocities. Oriented multiple suppression can accommodate shifted-apex noise and thus we anticipate it will perform better than Radon multiple suppression for some types of noise.

Here we have limited our testing and evaluation of the oriented workflow to 2D synthetic data and single-valued p-values. Important future work is to test and evaluate the oriented algorithm with multivalued p-values and 3D data.

ACKNOWLEDGMENTS

We would like to thank the reviewers and editor for their insightful comments and suggestions that improved this paper.

APPENDIX A

DERIVATION OF THE MIGRATION EQUATIONS

If we substitute equations 2 and 3 into equation 1, we obtain

$$t = \frac{1}{V^2}\left(\frac{x_s - x_m}{p_s} + \frac{x_r - x_m}{p_r}\right). \quad (A\text{-}1)$$

Squaring of equations 2 and 3 results in

$$V^2t_0^2 = \frac{(x_s - x_m)^2}{p_s^2V^2} - (x_s - x_m)^2 \quad (A\text{-}2)$$

and

$$V^2t_0^2 = \frac{(x_r - x_m)^2}{p_r^2V^2} - (x_r - x_m)^2,$$

which after equating, gives

$$\frac{(x_s - x_m)^2}{p_s^2V^2} - (x_s - x_m)^2 = \frac{(x_r - x_m)^2}{p_r^2V^2} - (x_r - x_m)^2. \quad (A\text{-}3)$$

Solving for V^2 in equation A-1 gives

$$V^2 = \frac{x_s - x_m}{p_st} + \frac{x_r - x_m}{p_rt}, \quad (A\text{-}4)$$

which is equation 5. Solving for V^2 in A-3 gives

$$V^2 = \frac{(x_s - x_m)^2/p_s^2 - (x_r - x_m)^2/p_r^2}{(x_s - x_m)^2 - (x_r - x_m)^2}. \quad \text{(A-5)}$$

Equating equations A-4 and A-5 as

$$\frac{(x_s - x_m)^2 - (x_r - x_m)^2}{t} = \frac{x_s - x_m}{p_s} - \frac{x_r - x_m}{p_r}$$

and expressing x_r with the use of the offset $d = x_r - x_s$ gives

$$\frac{-d^2 - 2d(x_s - x_m)}{t} = \frac{x_s - x_m}{p_s} - \frac{d + x_s - x_m}{p_r}.$$

We can solve for x_m, yielding

$$x_s - x_m = \frac{(d/p_r) - (d^2/t)}{(1/p_s) - (1/p_r) + (2d/t)}$$

$$= dp_s \frac{t - dp_r}{t(p_r - p_s) + 2dp_sp_r},$$

which leads to equation 4. Equation 6 can be obtained by taking the square root of equation A-2.

REFERENCES

Alkhalifah, T. A., and I. Tsvankin, 1995, Velocity analysis for transversely isotropic media: Geophysics, **60**, 1550–1566.

Barnes, A. E., 1966, Theory of 2-D complex seismic trace analysis: Geophysics, **61**, 264–272.

Baumstein, A., and M. T. Hadidi, 2004, 3D SRME: Data reconstruction and application to field data: 74th Annual International Meeting, SEG, Expanded Abstracts, E25–E33.

Billette, F., and G. Lambaré, 1998, Velocity macro-model estimation from seismic reflection data by stereotomography: Geophysics Journal International, **135**, 671–690.

Bóna, A., and D. Cooke, 2009, Velocity-less imaging of linearly inhomogeneous media in 3D: Presented at the 20th Annual Conference and Exhibition, ASEG.

Claerbout, J. F., 1985, Imaging the earth's interior: Blackwell.

Douma, H., and M. de Hoop, 2007, Leading-order seismic imaging using curvelets: Geophysics, **72**, no. 6, S231–S248.

Fomel, S., 2002, Application of plane-wave destruction filters: Geophysics, **67**, 1946–1960.

——, 2007, Velocity-independent time-domain seismic imaging using local event slopes: Geophysics, **72**, no. 3, S139–S147.

Foster, J. D., and C. C. Mosher, 1992, Suppression of multiple reflections using the radon transform: Geophysics, **57**, 386–395.

Hua, B., and G. A. McMechan, 2003, Parsimonious 2D prestack Kirchhoff depth migration: Geophysics, **68**, 1043–1051.

Lambaré, G., 2006, Stereotomography: Past, present and future: CSEG Recorder, 31.

Matson K., and R. Abma, 2005, Fast 3D surface-related multiple elimination using azimuth moveout for multiples: 75th Annual International Meeting, SEG, Expanded Abstracts, 2064–2068.

Matson, K., and J. Zhang, 2007, Improving the accuracy of fast 3D SRME: 77th Annual International Meeting, SEG, Expanded Abstracts, 2481–2484.

Ottolini, R., 1983, Velocity independent seismic imaging: Stanford Exploration Project, **37**, 59–68.

Schleicher, J., J. C. Costa, L. T. Santos, A. Novais, and M. Tygel, 2009, On the estimation of local slope: Geophysics, **74**, no. 4, P25–P33.

Stolt, R., 1978, Migration by Fourier transform: Geophysics, **43**, 23–48.

Taner, M. T., and F. Koehler, 1969, Velocity spectra — Digital computer derivation and applications of velocity function: Geophysics, **34**, 859–881.

Taner, M., F. Koehler, and R. Sherrif, 1979, Complex seismic trace analysis: Geophysics, **44**, 1041–1063.

Yilmaz, Ö., 2001, Seismic data processing, 2nd ed.: SEG.