Chapter 2

Dip moveout and migration in velocity space

2.1 INTRODUCTION

Accurate estimates of the velocity at which acoustic waves propagate underground are critical for imaging and interpreting seismic reflection data. Traditionally, a distinction is made between the steps of estimating velocity and of imaging. Velocity analysis ordinarily is done first, because a velocity estimate is required for the subsequent imaging processes of stacking, migration, or prestack migration. Unfortunately, the imaging algorithms are often better than the knowledge of velocities; errors in migration techniques can be overshadowed by uncertainties in the velocity field. Moreover, in surveys over complex geologic structure, many conventional methods for estimating velocity will yield inaccurate results. Stacking velocities will not be the same as migration velocities, and neither one will necessarily be readily related to the real acoustic velocities. The processes of velocity analysis and imaging do not then separate so cleanly into two distinct steps, since knowledge of the structure is also needed for accurate velocity analysis.

Most conventional processing of reflection data uses traces sorted into common-midpoint (CMP) gathers in major part because the changes in normal moveout (NMO) of reflection events with offset allow estimation of velocities. In a medium of a single, constant velocity v the traveltimes corresponding to the primary reflection from a planar horizontal interface will satisfy the hyperbolic NMO equation

$$t^2 = t_0^2 + 4h^2/v^2 , \tag{2.1}$$

where t is the two-way traveltime, h is the half-offset, that is, one half the distance between the shot and geophone, and t_0 is the two-way traveltime for zero offset, or coincident shot and geophone. Note that I use velocity here and throughout to mean the compressional wave velocity, estimated from primary reflections. I do not attempt to consider the effects of mode conversions, multiples, attenuation, or anisotropy, because these are usually most easily studied only *after* a good initial estimate of compressional wave velocities is available.

For stratified media equation (2.1) will no longer be exact, but it has long been known that equation (2.1) may still be used for velocity estimation if the velocity parameter v is interpreted as a root-mean-square (rms) average of the overlying interval velocities. This identification is not exact, but it often provides an acceptable estimate (Al-Chalabi, 1974, 1979; Hubral and Krey, 1980; Cordier, 1985).

The straightforward application of hyperbolic NMO to estimate velocity may break down either when the velocity varies laterally or when the reflecting horizons are not horizontal. In the former case the moveout will deviate increasingly from hyperbolic with larger lateral velocity variation. I return to discuss this case later in this chapter, but will be concerned for the most part here with the effects of geologic structure. If the reflectors are dipping, the moveout will remain hyperbolic, but the NMO velocity that must be used in equation (2.1) will be related to the flat bed NMO velocity by

$$v_\theta = \frac{v}{\cos\theta} , \tag{2.2}$$

where θ is the reflector dip (Dix, 1955; Levin, 1971). Processing algorithms that attempt to correct for this effect are known by a variety of names; I will refer to them generically as dip moveout (DMO) methods (e.g., Judson et al., 1978; Yilmaz and Claerbout, 1980; Deregowski and Rocca, 1981; Bolondi et al., 1982; Hale, 1984). Deregowski (1986) provides a good summary of DMO techniques with further references.

In principle DMO does not require a detailed knowledge of velocities. Unfortunately, as Hale (1984) showed, DMO is properly applied *after* NMO. For velocity analysis, one would like to be able to apply DMO *before* NMO, because NMO requires a commitment to a velocity function. In this chapter I introduce a new DMO algorithm that sidesteps this problem. In this method DMO is implemented on multiple constant-velocity stacks, and velocity analysis is postponed until after both NMO and DMO.

The velocities derived from a stacking velocity analysis often are not suitable for migration. One problem, as mentioned above, is the effect of dip on the measured velocity. A second problem is that of spatial mis-positioning of the velocity information. Stacking-velocity analysis gives one information only where there are reflectors, and tells one about the velocities above those reflectors. However, unmigrated reflectors can appear to be located a substantial distance both laterally and vertically from their correct, migrated positions. For migration one needs velocity information in migrated time (or better still, in depth). (See Hatton et al., 1986 for a discussion of this point.) The problem arises of needing to know reflector positions before doing velocity analysis, but needing to know velocities to migrate reflectors to their correct positions. In this chapter I show how migration can be done naturally on constant-velocity stacks. As with DMO, choice of a velocity function can be postponed until after reflector imaging, so the velocities derived are naturally in migrated time.

The algorithms developed here are derived first for a homogeneous medium. The extension to a stratified medium is straightforward and involves the same approximations used when doing conventional NMO stacking, and Kirchhoff migration: NMO and diffraction curves must be approximately hyperbolic. This assumption of hyperbolicity breaks down when lateral velocity variation becomes large, but I show with field data examples that the methods presented here remain useful well beyond the strict assumption of lateral homogeneity.

2.2 IMAGING IN VELOCITY SPACE

What is a velocity space?

Seismic reflection data records a wave field (either pressure or particle velocity) as a function of time and position. By a "velocity space" I mean the result of any transformation of the seismic data that makes them a function of velocity. Specifically, I begin with CMP sorted data $q(y,h,t)$, where q is the wave field, y is midpoint, h is half-offset, and t is time. Most stacking-velocity analysis proceeds by trial stacking at a range of velocities, the principle being that the correct velocity will be the one that produces the best image, or maximizes some statistic such as semblance (Neidell and Taner, 1971). If all the data are stacked this way, the resulting data $q(y,v,t)$ depend on the velocity v instead of on the offset h. NMO stacking constitutes a linear operator that transforms the original data into a particular velocity space, namely that composed of constant-velocity stacks.

The multiple-velocity-stack operator is not the only one that can be used to produce a velocity space. Chun and Jacewitz (1978) suggested migrating a single stacked section at a range of velocities, and Gray (1985) proposed migrating velocity-analysis semblance panels using their stacking velocities. Both of these methods produce what I would term here a velocity space, but neither of them confronts the differences between stacking velocities and migration velocities. Shurtleff (1984) and Tieman (1984) generated another velocity space by performing constant-velocity prestack migration at a range of velocities. In the section on constant-velocity prestack migration I compare the algorithms proposed here with Shurtleff's method, and show that velocity-space DMO and migration can be interpreted as effecting a transformation from one velocity space (constant-velocity stacks) to another (constant-velocity prestack migrations).

The word "velocity" as used here refers to a parameter in a transformation operator such as NMO stacking or prestack migration. For data recorded from a medium with a single constant velocity, this velocity parameter in the operator will coincide in value with the medium velocity, and hence the choice of terminology is natural. This simple identification no longer holds for inhomogeneous media. The velocity parameter then is an rms average of the overlying medium velocities, or some more complicated function of the physical velocities. However, I retain the same terminology in accord with common usage.

DMO on constant-velocity stacks

For clarity, I consider first a simple example of synthetic data. Two reflectors are modeled, one flat and one dipping 45°, in a medium with a constant velocity of 2 km/s. For this example, I simply computed traveltimes and convolved with an unvarying wavelet; no effort was made to include realistic diffractive or waveform effects. Figure 2.1a shows a zero offset section of the resulting data. Similar synthetics were also computed for 24 offsets up to h =1.38 km.

Figure 2.2a shows the result of applying NMO stacking using the medium velocity of 2 km/s. The flat bed is stacked correctly, but the dipping bed is severely attenuated, because the apparent stacking velocity needed to image it is biased upward by the dip cosine correction of equation (2.2). Figure 2.2b shows the stack using a velocity of 2.83 km/s, which is suitable for the 45° dipping horizon. This time, the dipping bed images well, but the flat bed is attenuated.

This dip filtering property of NMO stacking is well known. One approach that is used to cope with it is to stack over a range of velocities, and then to sum the various stacks. However, what is really desired is to include only the flat event from the low velocity (2 km/s) stack and the dipping event from the high velocity (2.83 km/s) stack, and to discard the other events that are incorrectly stacked. This suggests that one should decompose each stack using dip filters and select only the appropriate events. This dip decomposition could be done directly on the data using

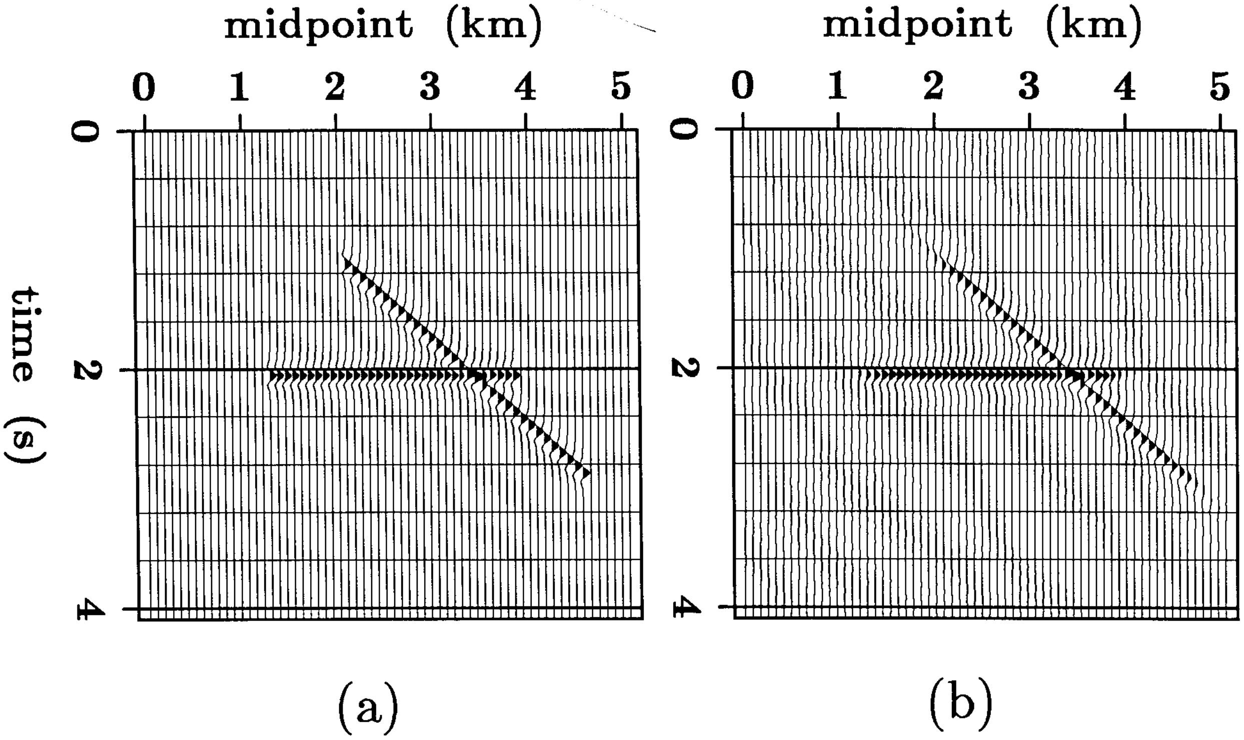


FIG. 2.1. (a) Synthetic zero-offset data showing one flat bed and one bed dipping 45°. Similar data were modeled for a range of offsets up to 1.38 km using a velocity of 2 km/s. Only the kinematics of the bed reflection are modeled, and no diffractive effects are incorporated. (b) The 2 km/s section after applying the velocity-space DMO algorithm to the synthetic data.

slant stacks (Schultz and Claerbout, 1978). Instead, I Fourier transform over the midpoint and time axes to effect a similar decomposition into plane-wave components corresponding to the different dips. If k_y is the midpoint wavenumber, that is, the Fourier variable dual to the midpoint y, and ω is similarly the temporal frequency, the dip information will now be contained in the ratio k_y/ω. If v is the medium velocity, the dip θ of a reflector will be given (e.g., Claerbout, 1984) by

$$\sin\theta = \frac{v k_y}{2\omega} . \tag{2.3}$$

The factor of two enters because two-way traveltimes are used.

Using equation (2.3) one could thus extract from the 2 km/s stack of Figure 2.2a the flat event ($k_y=0$), and from the 2.83 km/s stack of Figure 2.2b the 45° dipping event ($k_y=\omega/\sqrt{2}$). The rest of each Fourier transformed stack can be zeroed out and the stacks inverse transformed and summed together. The resulting stack is shown in Figure 2.1b. The flat and dipping events now stack equally well, as

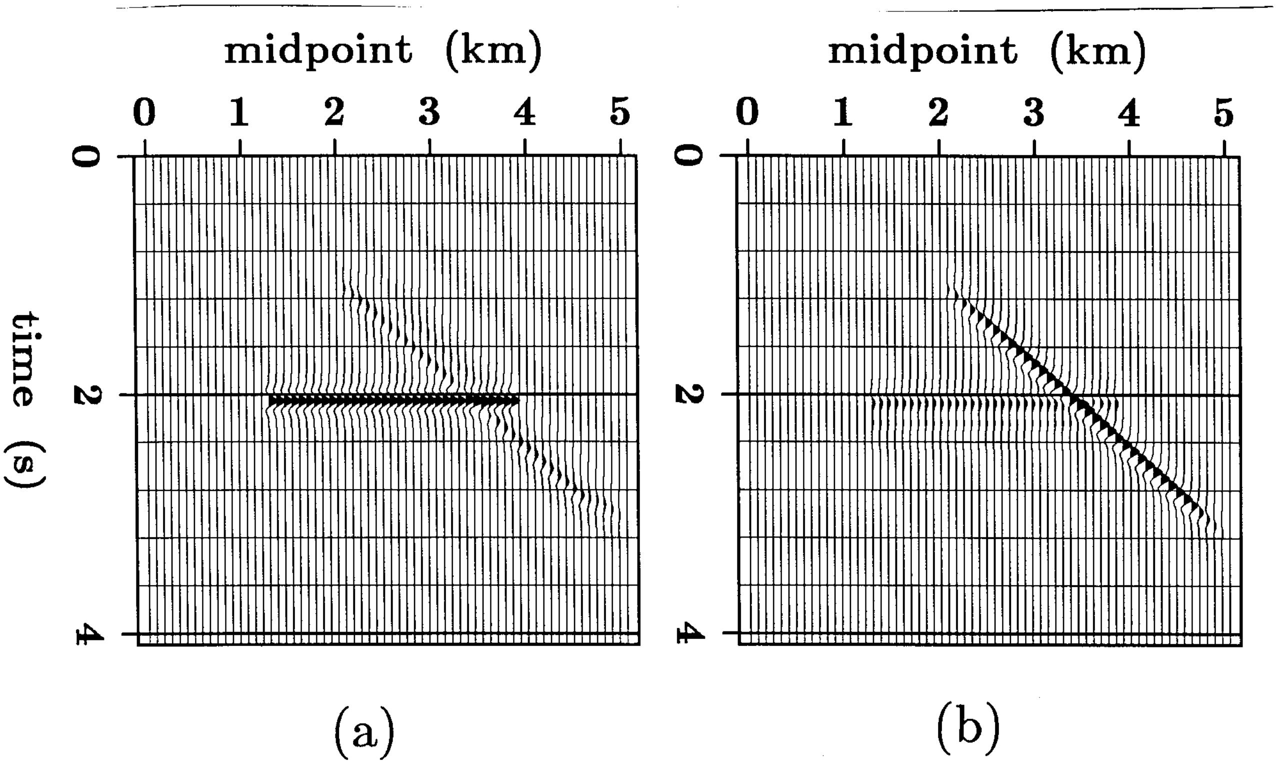


FIG. 2.2. (a) 2 km/s stack of the synthetic data of Figure 2.1a. The flat bed stacks well, but the dipping bed is weak. (b) 2.84 km/s stack of the synthetic data of Figure 2.1a. The dipping bed stacks better at this higher velocity, but the flat bed is now weak.

predicted.

The above description appears to require that the medium velocity and the dips of all important reflectors are somehow known in advance. In fact, neither needs to be known; all velocities and dips can be treated simultaneously. Combining equations (2.2) and (2.3) gives an expression relating the medium velocity v to the velocity v_θ at which a bed with dip θ stacks best:

$$v_\theta = v \left(1 - \frac{v^2 {k_y}^2}{4\omega^2} \right)^{-1/2} . \tag{2.4}$$

Equivalently, equation (2.4) may be solved for v in terms of v_θ to give

$$v = v_\theta \left(1 + \frac{v_\theta^2 {k_y}^2}{4\omega^2} \right)^{-1/2} . \tag{2.5}$$

If one stacks over a range of velocities and then Fourier transforms all the stacks, equation (2.5) may be used to apply the DMO correction simultaneously for all dips and all possible velocities. The combination of multiple velocity stacking and two-dimensional Fourier transformation yields a data space $q(k_y, \omega, v)$ that decomposes the data in terms of both velocity and dip. In this space the DMO correction takes the simple form, suggested by equation (2.5), of a regridding or one-to-one mapping of the velocity axis for each (k_y, ω) pair. The result of this velocity correction after inverse Fourier transformation is a stack at the correct medium velocity in which events of all dips stack equally well. This is how Figure 2.1b was actually generated; 60 constant-velocity stacks were generated spanning a range from 2 km/s to 3.18 km/s, equation (2.5) was applied, and the new 2 km/s stack was selected for display.

Equivalence of velocity-space DMO with Hale's algorithm

The formulation presented here of DMO as a dip-dependent velocity correction applied to constant-velocity stacks is easily implemented, but its relation to other DMO methods may not be readily apparent. In fact, it is formally equivalent to the DMO method of Hale (1984). That is, if one performs NMO at a range of velocities, applies Hale's algorithm to each moveout-corrected section, and then stacks the data, the resulting suite of DMO-corrected stacks will be identical to the ones generated by the velocity space algorithm. The equivalence can be seen if one considers the action of each algorithm on dip-decomposed data. The velocity-space algorithm can be summarized by the following sequence of steps: NMO stacking over a range of velocities, dip decomposition of the stacks by two-dimensional Fourier transformation, correction of the velocities by shifting data between stacks, and inverse Fourier transformation. Jakubowicz (1984) showed that the Hale algorithm can also be implemented by dip-dependent NMO combined with dip filtering. If the two algorithms are written this way, in terms of their actions on each dip component of the data, the formal difference between the methods reduces to an interchange of order of an integral over offset (stacking) and a Fourier transform over time. The two steps commute, so the methods are formally equivalent. I provide a detailed derivation of this equivalence in Appendix A.

The principal difference between the Hale-Jakubowicz DMO algorithm and the velocity-space algorithm lies in whether the dip correction is done before or after stacking. If the velocity function is known, it is computationally cheaper to apply NMO followed by a DMO algorithm such as Hale's, and then stack the data. However, by implementing the dip corrections after multiple-velocity stacking, the velocity-space algorithm postpones velocity analysis until DMO, and requires no prior knowledge of the velocity function.

Migration of DMO-corrected constant-velocity stacks

After the application of the velocity-space DMO correction, the stack at the correct medium velocity will contain all dips, and will closely resemble a zero-offset section, as shown in Figures 2.1a and 2.1b. In particular, diffractions, which contain a wide range of dips, will be stacked correctly. This suggests that it is appropriate to apply a zero-offset migration algorithm to this section. Figure 2.3a shows the result of migrating the synthetic data of Figure 2.2b using a velocity of 2 km/s; the initial model of two reflectors is successfully reconstructed.

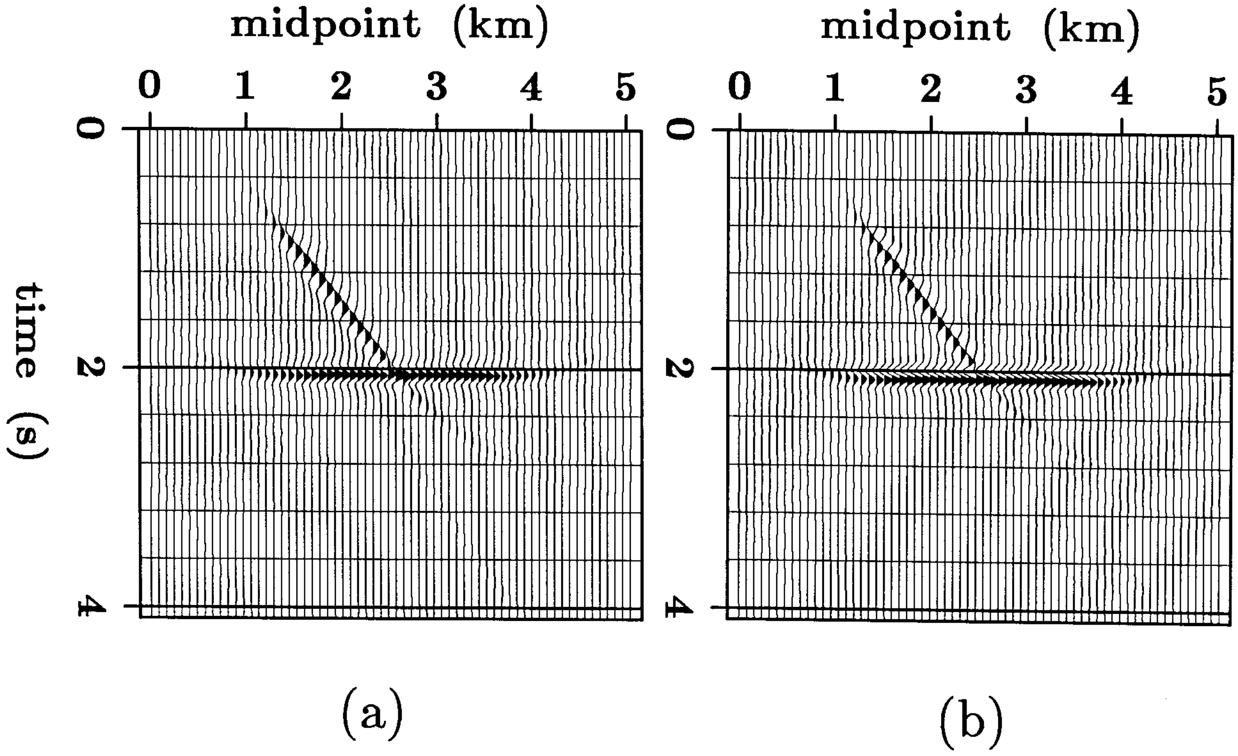


FIG. 2.3. (a) 2 km/s stack of the synthetic data of Figure 2.1a, after DMO and migration. The wavelet is zero-phase, as it is in the zero-offset data. (b) 2 km/s prestack Stolt migration of the synthetic data of Figure 2.1a. The wavelet differs in phase by 45° from the original data.

In practice, one would not know in advance which velocity stack represents the correct medium velocity. However, one can migrate all the DMO-corrected stacks, each at the dip-corrected, or flat-bed, velocity. This can be easily and economically incorporated in the velocity-space DMO algorithm by applying the constant-velocity migration algorithm of Stolt (1978) to the stacks, before inverse Fourier transformation. To do this, one simply applies to each (k_y, v) trace the mapping from unmigrated frequency ω to migrated frequency k_τ given by

$$k_\tau = \omega \left(1 - \frac{v^2 k_y^2}{4\omega^2} \right)^{1/2} , \tag{2.6}$$

and then weights the data by the Jacobian of this transformation, $\partial\omega/\partial k_\tau = k_\tau/\omega$. Note the formal similarity of equation (2.6) to the DMO equation (2.5); in this velocity space, DMO comprises a regridding of the velocity axis and migration a similar regridding of the frequency axis. Since the data is already doubly Fourier transformed, the additional computational cost of incorporating the migration step is minimal.

Constant-velocity prestack migration

Shurtleff (1984) presented a method of imaging and velocity analysis that used the constant-velocity prestack f-k domain migration algorithm of Stolt (1978) to create a velocity space different from the space of constant-velocity stacks discussed so far in this chapter. In this approach, the data are Fourier transformed over offset h as well as over midpoint y and time t, and directly mapped to a migrated image using the change of variable from unmigrated frequency ω to migrated frequency k_τ given by

$$k_\tau = \frac{\omega}{2} \left[\left(1 - \frac{(k_y + k_h)^2 v^2}{4\omega^2} \right)^{1/2} + \left(1 - \frac{(k_y - k_h)^2 v^2}{4\omega^2} \right)^{1/2} \right] . \tag{2.7}$$

This mapping is a generalization of equation (2.6) for non-zero offset data.

Hale (1983) proved that, for constant velocity, the sequence of NMO, DMO, stack, and zero-offset migration is kinematically identical to f-k prestack migration. Since the velocity-space DMO is equivalent to Hale's method, one would expect the result of velocity-space DMO and migration to be extremely similar to that produced using Shurtleff's approach.

Figure 2.3b shows the result of applying Stolt's (1978) prestack f-k migration algorithm to the same data used to produce Figure 2.3a. The two figures, as predicted, are strikingly similar. The kinematics are identical, and the only significant difference is the phase of the wavelets associated with the reflection events. The velocity-space DMO and migration result preserves the zero-phase character of the initial data, whereas the prestack migration result is phase rotated by 45° ($\pi/4$). This phase difference between the DMO result and the prestack migration is predicted in Hale's (1983) asymptotic analysis of DMO. Deregowski and Brown (1983) provide an extensive discussion of the causes of such phase shifts in migration and diffraction.

The close similarity shown here between the result of prestack f-k migration as used by Shurtleff and that of the velocity-space DMO and migration method corroborates the results shown by Morley et al. (1985). Because the results of these two methods can be brought into agreement by a simple phase rotation, I will henceforth treat the results as fundamentally equivalent. The resulting velocity spaces can be used interchangeably for velocity analysis and the extensions to variable velocity are the same. When I wish to distinguish between the approaches, I will refer to them specifically as CVS (constant-velocity-stack) and CVPM (constant-velocity-prestack-migration) methods, respectively.

Morley et al. (1985) reported that the CVS method took 1.5 to 1.7 times as much computer time as the CVPM method. The major computational cost of the CVS algorithm ordinarily lies in the generation of the constant-velocity stacks. For computers with vector-arithmetic capabilities and large core memories, this cost can be reduced greatly if constant-offset sections are transposed in core and NMO is performed for many midpoints simultaneously. Also, high accuracy can be retained at low cost for the many interpolations required by repeated NMO if the data are initially oversampled along the time axis. In these ways, I was able to make the CVS algorithm run as fast as my implementation of the CVPM method. However, I attempted no rigorous comparison of computational costs; I believe both methods are computationally reasonable to implement on current computers.

The CVPM method has the advantage that it can be applied directly to shot profiles, saving the cost of a sort into CMP order. The CVPM method is also more appropriate when a small velocity range is to be sampled finely, because, unlike the CVS method, it does not always require processing all velocities if only a small range is to be examined. The CVS method has the countervailing advantage of offering CMP stacks and DMO-enhanced stacks as intermediate output. The latter method is also more flexible and robust in handling irregularly sampled data, because it does not require the regular sampling in offset that is needed for the CVPM method, and is less sensitive to aliasing and truncation along the offset axis. Approaches to ameliorating some of these potential problems with the CVPM method are discussed further by Li et al. (1987). (The ability to handle data irregularly sampled in offset is also an advantage of CVS over many other DMO algorithms that require generation of constant-offset sections.)

2.3 ALLOWING FOR SPATIALLY VARYING VELOCITY

So far I have treated only the idealized case in which the medium velocity is constant. More generally, the velocity-space methods will work well as long as NMO and diffraction curves remain approximately hyperbolic, so that NMO stacking (hyperbolic NMO summation) and Kirchhoff time migration (hyperbolic diffraction summation) can produce a coherent image. Note, however, that both the CVS and CVPM methods are *time* migration methods; for data with significant lateral velocity variation a coherent image may be produced, but *depth* migration is really needed to position events correctly (Hubral, 1977; Larner et al., 1981; Hatton et al., 1981).

When would velocity variations be so large that these methods fail entirely? Qualitatively, the answer is, when no particular choice of velocity produces an optimal image, or when several velocities yield equally good (or poor!) images. One has to be able to make some judgement about which hyperbola fits the data best; if the data is radically nonhyperbolic, no such decision is possible. Of course, I have not yet specified a criterion for "best" fit. For stacking-velocity analysis, criteria commonly used are maximum energy or maximum semblance in the resulting stack. For evaluating migration, one might wish to use entropy to measure focusing (De Vries and Berkhout, 1984; Harlan et al., 1984). I return to these questions in the section on the effects of lateral velocity variation, and again in the section on velocity analysis.

Vertical velocity variation

For vertically stratified media, the NMO equation (2.1) is no longer exact, but standard processing of seismic data still uses hyperbolic moveout for stacking, with the velocity parameter in equation (2.1) interpreted as some function of the overlying interval velocities, most commonly an rms average. Thus, at each depth, the overlying layers are treated as though their effects could be simulated by a hypothetical replacement medium with a constant velocity.

What will the implications of velocity variation be for the velocity-space DMO and migration algorithms? Consider the collection of constant-velocity stacks $q(y,v,t)$, as constituting a "cube" of data, as in Figure 2.4. At each midpoint y, the velocity as a function of the zero-offset traveltime t describes a curve in the (v,t) plane. In each stack, there will exist a small range of t for which the velocity is correct; earlier or later in the data, the velocity will be too low or too high. Using the idea of an effective constant velocity replacement medium for each stack, however, one can use the velocity-space DMO and migration algorithms just as for constant velocity. The portion of each stack for which the velocity is reasonable will image well; the rest of the stack is discarded when the final image is selected.

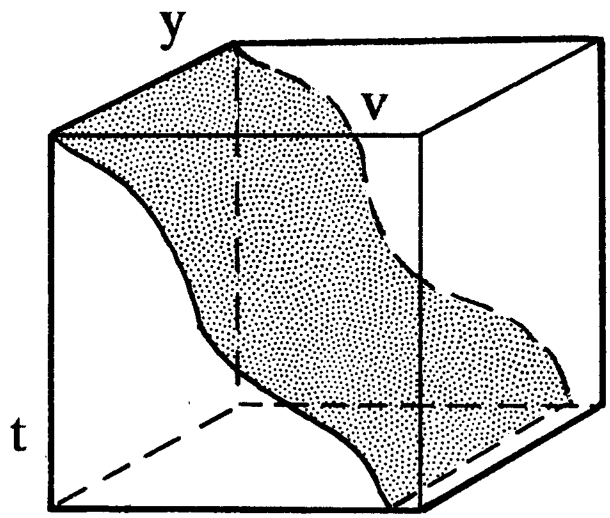
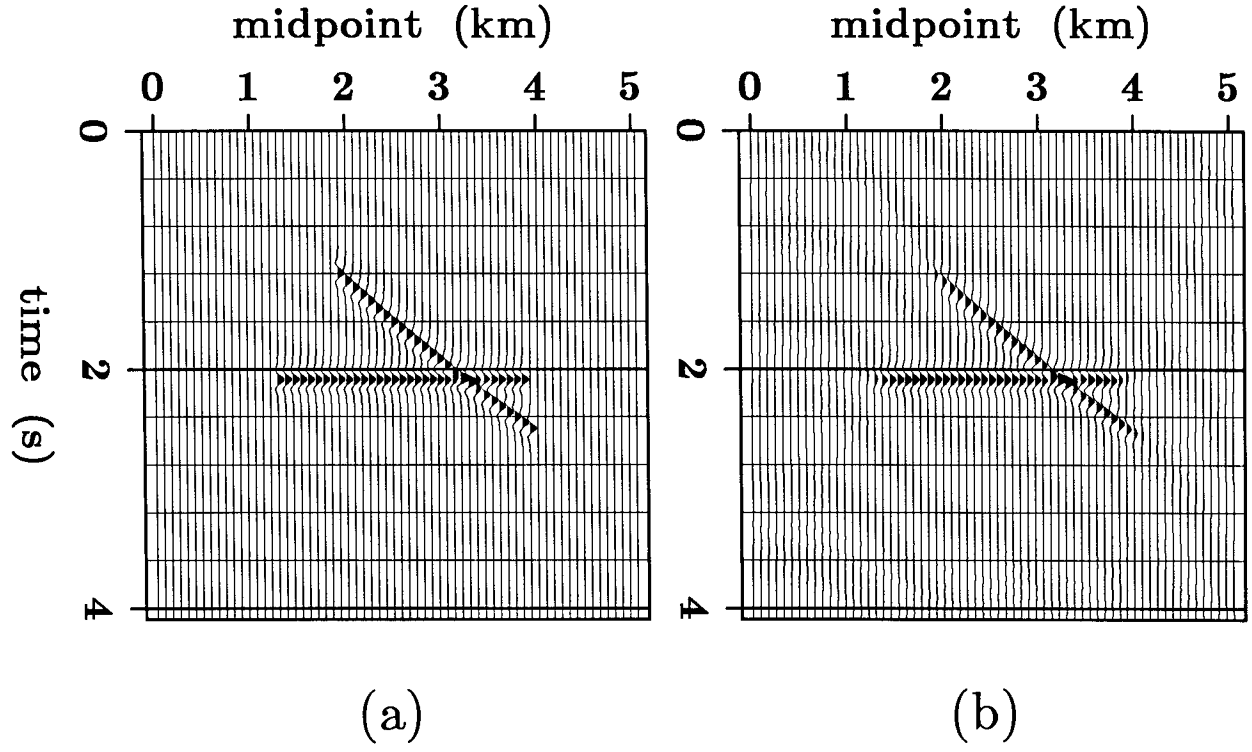


FIG. 2.4. A set of constant-velocity images forms a data cube with axes of midpoint y, time t, and velocity v. An image corresponding to a variable-velocity field can be created by interpolation between the constant velocity images. Schematically, this interpolation "slices" through the data cube along the specified velocity function, as suggested in this figure.

The selection of an image now consists of more than simply choosing the single correct velocity stack. The extraction of the image can be visualized as a process of "slicing" the data cube along the path defined by the velocity function, as shown schematically in Figure 2.4. In practice, this means extracting portions of the final image from each stack, and interpolating between stacks where the velocity falls between one of those used in generating the stacks.

As an example, Figure 2.5a shows a zero-offset section of the same reflector model as in Figures 2.1 through 2.3, but now using a velocity that increases linearly with depth ($v = 1.5 + 0.5z$) instead of a constant 2 km/s velocity. 200 stacks were created from these data, sampled evenly in velocity from 1.34 km/s to 5.32 km/s.

The data, like that in the constant velocity example, were created by tracing rays to compute traveltimes, and convolving with an invariant zero-phase wavelet. Figure 2.5b shows this section after DMO, extracted with a depth-variable velocity function, and Figure 2.6 shows the result extracted after migration. Despite the substantial vertical velocity gradient, the CVS algorithm images the data well.



(a) (b)

FIG. 2.5. (a) Synthetic zero-offset data using a velocity function $v(z)$=1.5+0.5z km/s. The reflector model is the same as used in Figure 2.1a. (b) The $v(z)$ data after applying the velocity-space DMO algorithm. This section was created using the rms average of the interval velocities.

The velocity function used for producing Figures 2.7 and 2.6 was the rms average of the interval velocities. I discuss in section 2.5 the accuracy of this approximation.

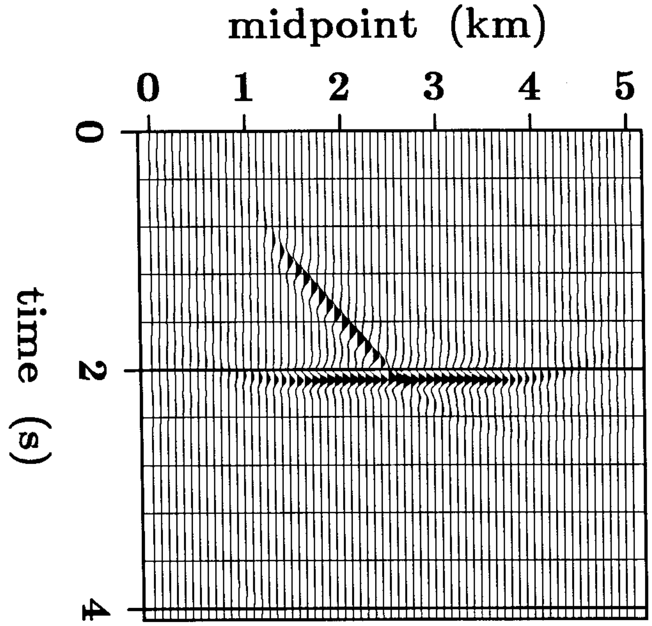


FIG. 2.6. The $v(z)$ data after applying the velocity-space DMO and migration algorithm. This section was created using the rms average of the interval velocities. The results are comparable to the image from constant-velocity data in Figure 2.3a.

Lateral velocity variation

The velocity-space algorithms can also be applied when the velocity varies laterally, provided this variation is not too large. The same concept of an effective homogeneous replacement medium can be used, although now the replacement velocity will also vary laterally. The extraction of an image by interpolation is also the same; the velocity function used will now differ from midpoint to midpoint.

Lateral velocity variation does introduce one new and subtle complication. For constant velocity, both NMO and diffraction curves are hyperbolic, and have the same velocity. For vertically stratified velocity, both curves deviate from hyperbolas, but the best fitting effective velocity will remain the same for both, since the deviations are identical. For a laterally varying velocity, the effective velocities for stacking and for zero-offset migration can differ. Fortunately, these differences are usually small.

2.4 FIELD DATA EXAMPLES

I illustrate the application of velocity-space DMO and migration on two field data sets. Both are from marine surveys, but they differ in structural style, tectonic setting, and degree of lateral velocity variation present. The first is from the Gulf of Mexico, and is dominated by growth faulting associated with a nearby salt dome. The second shows contorted sediments from off the coast of southern California.

Gulf Coast data example

Figure 2.7 shows a portion of a data set collected in the Gulf of Mexico. These data are from the same survey line used for illustration by Hale (1983, 1984) and by Rothman et al. (1985). To process these data 101 constant-velocity stacks were created. These stacks were sampled evenly in squared slowness ($1/v^2$) with an interval of 0.0046 s^2/km^2 covering a range from 0 s^2/km^2 (infinite velocity) to $0.46 s^2/km^2$ (1.47 km/s). This choice of sampling strategy is explained in section 2.6. The stacked section in Figure 2.7 was extracted by interpolation between these stacks. The CVS algorithm was then applied to these data, using DMO to resample from slowness squared to slowness, yielding 101 stacks covering a range from 0.333 s/km (3.0 km/s) to 0.677 s/km (1.48 km/s) in steps of 0.00344 s/km. Figure 2.8 shows the section extracted from the stacks after DMO, and Figure 2.9 shows the section extracted after both DMO and migration. The ability of DMO to counteract the dip selectivity of stacking is exemplified by the well-imaged fault-plane reflections in Figure 2.8. The diffraction tails from the bed truncations at the fault are also more prominent after DMO. In Figure 2.9, migration collapses these diffractions, as expected, and shifts the fault-plane reflections to their correct locations, aligned with the truncations in the flat beds. Some of the fault plane reflections do not align this way; Hale (1983) suggests that this is caused by 3-D effects, and I would concur.

To illuminate the action of the velocity-space algorithm, I examine in detail a small window of data from the left side of Figure 2.7. Figure 2.10a shows a constant-velocity stack of this window at a velocity of 2.20 km/s. This velocity is an appropriate stacking velocity for the flat beds at 2.2 seconds, but the steeply dipping fault-plane reflection is barely visible. Figure 2.10b shows a similar stack, but using a velocity of 2.69 km/s. Now the fault plane reflection is apparent, but the flat beds are attenuated. After application of velocity-space DMO, both the dipping and flat reflectors in the region around 2 seconds stack well using a velocity of 2.20 km/s (Figure 2.11a). Figure 2.11b shows the 2.20 km/s DMO stack of Figure 2.11a after migration using the same velocity.

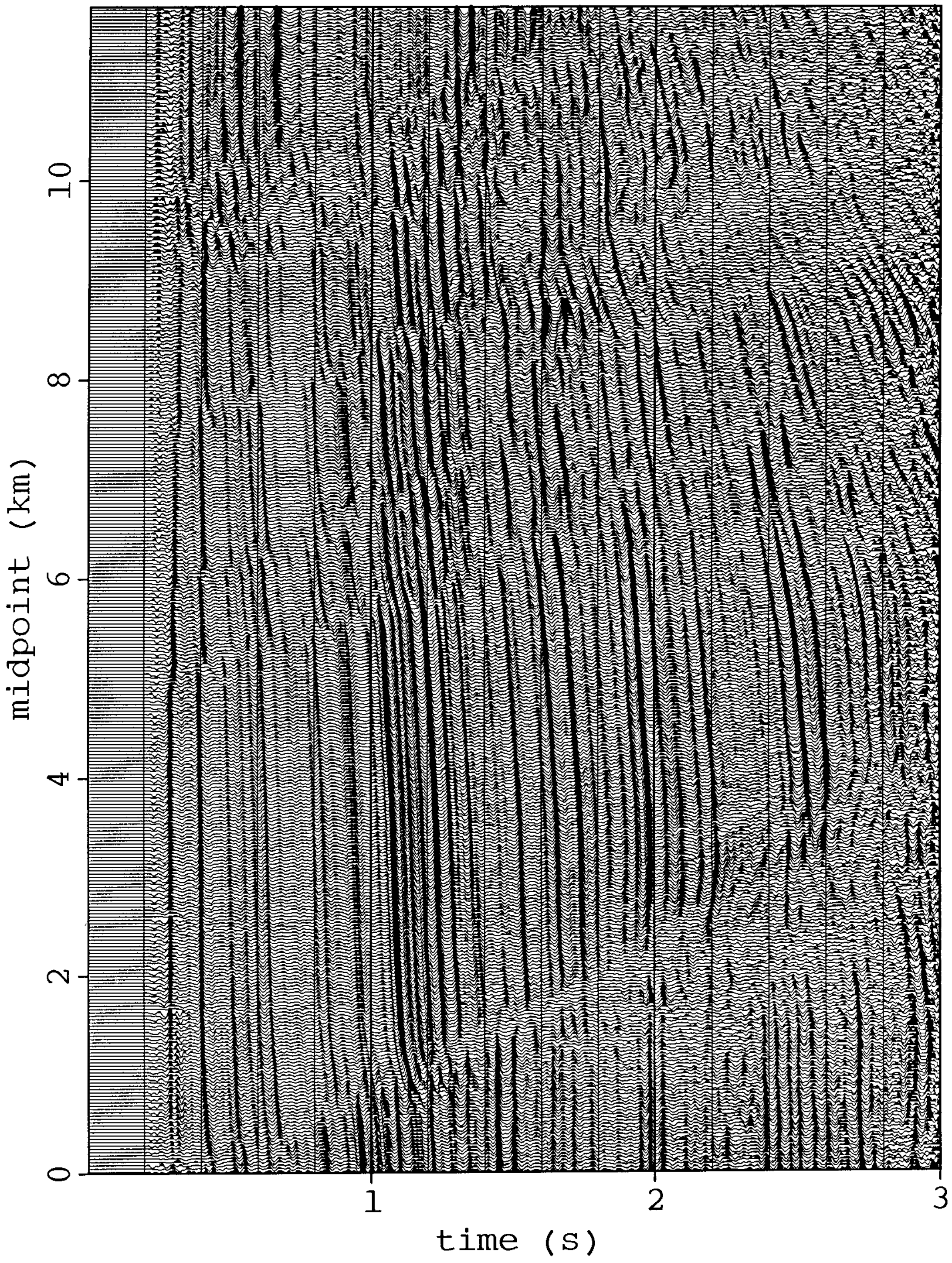


FIG. 2.7. Portion of a data set from the U.S. Gulf coast, extracted from a suite of 101 constant-velocity stacks.

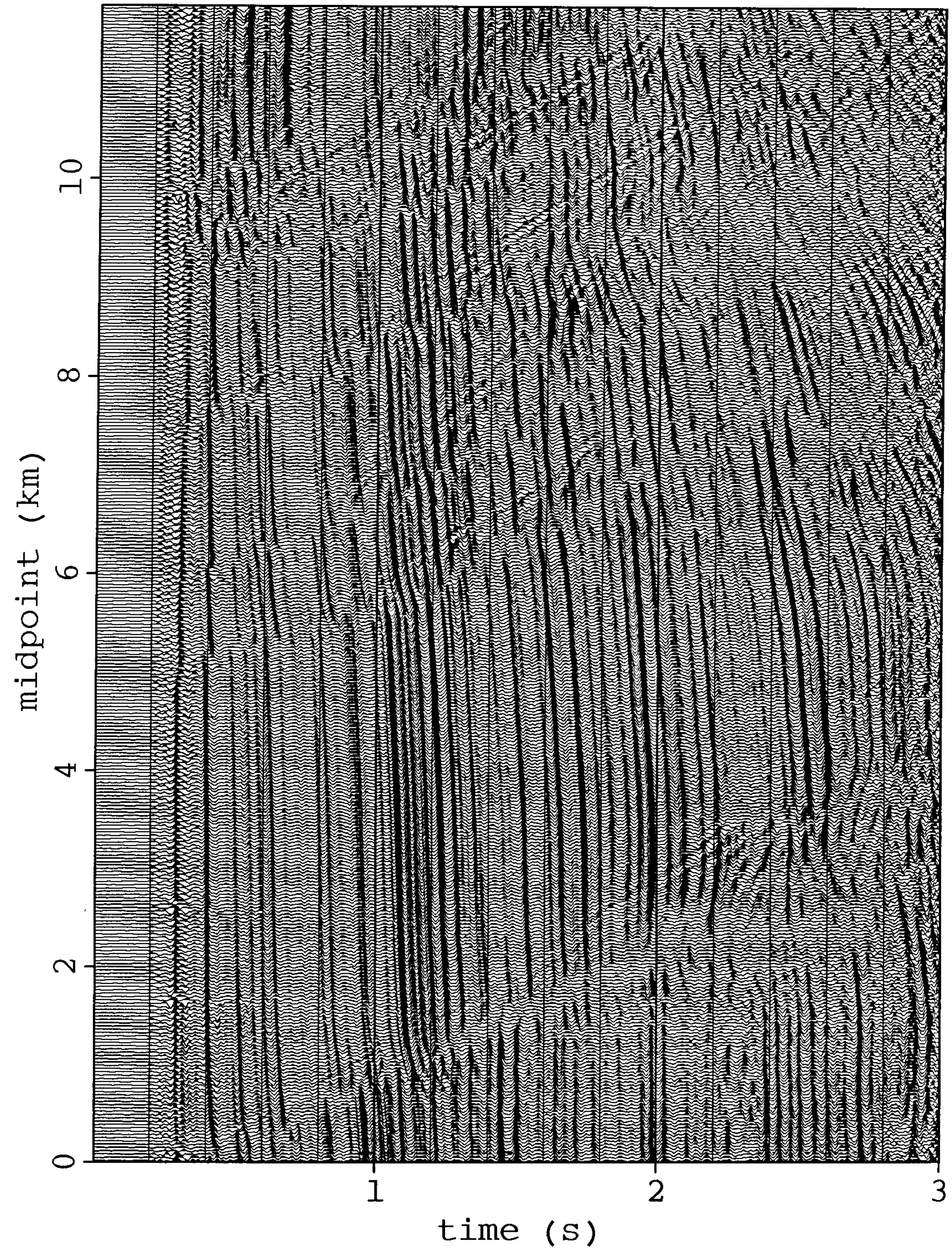


FIG. 2.8. Portion of a data set from the U.S. Gulf coast, extracted from a suite of 101 constant-velocity stacks after DMO correction. The fault-plane reflections and the diffraction tails from bed truncations are more evident than in Figure 2.7.

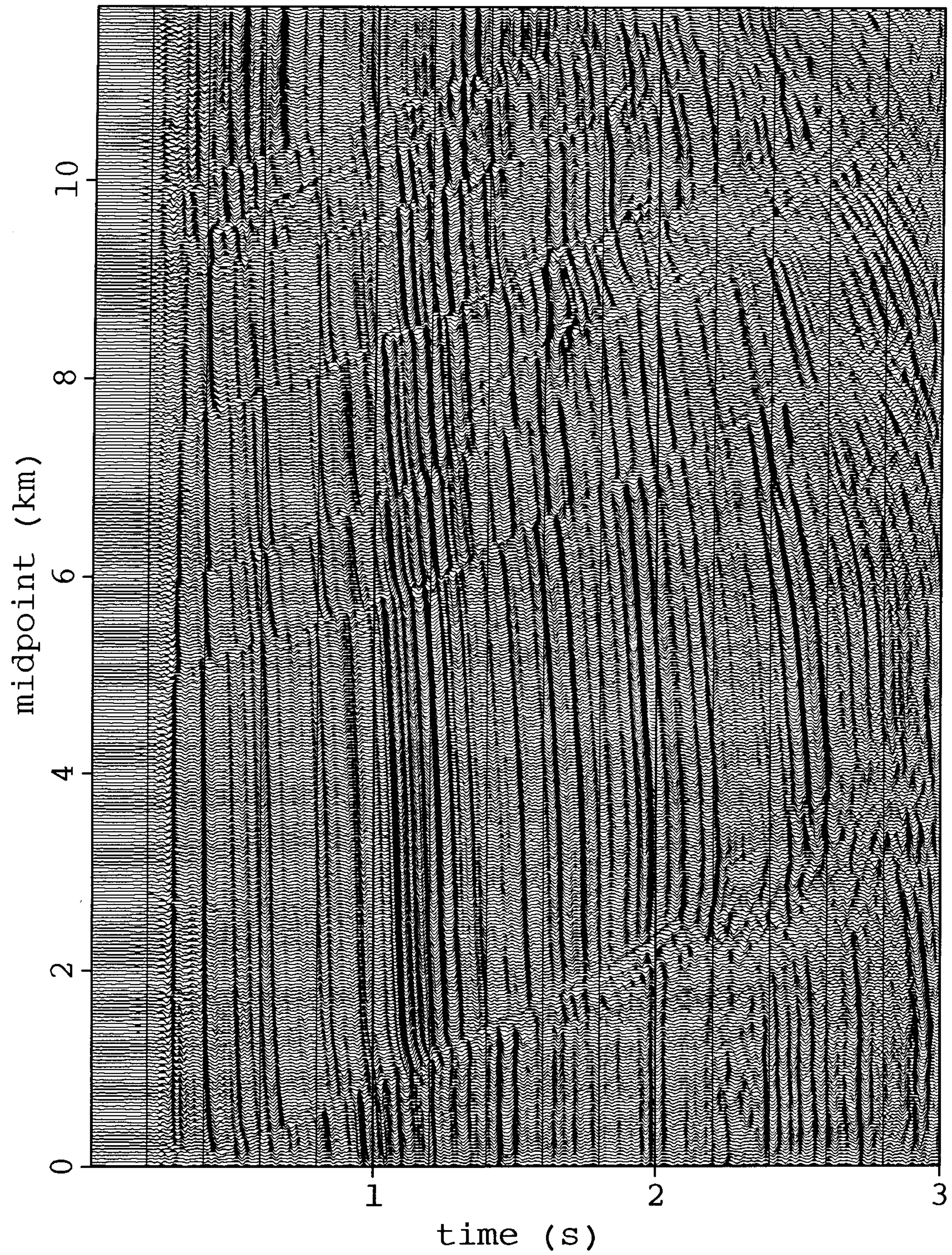


FIG. 2.9. Portion of a data set from the U.S. Gulf coast, extracted from a suite of 101 constant-velocity stacks after DMO correction and migration. The fault-plane reflections align with the bed truncations, and diffraction tails are collapsed.

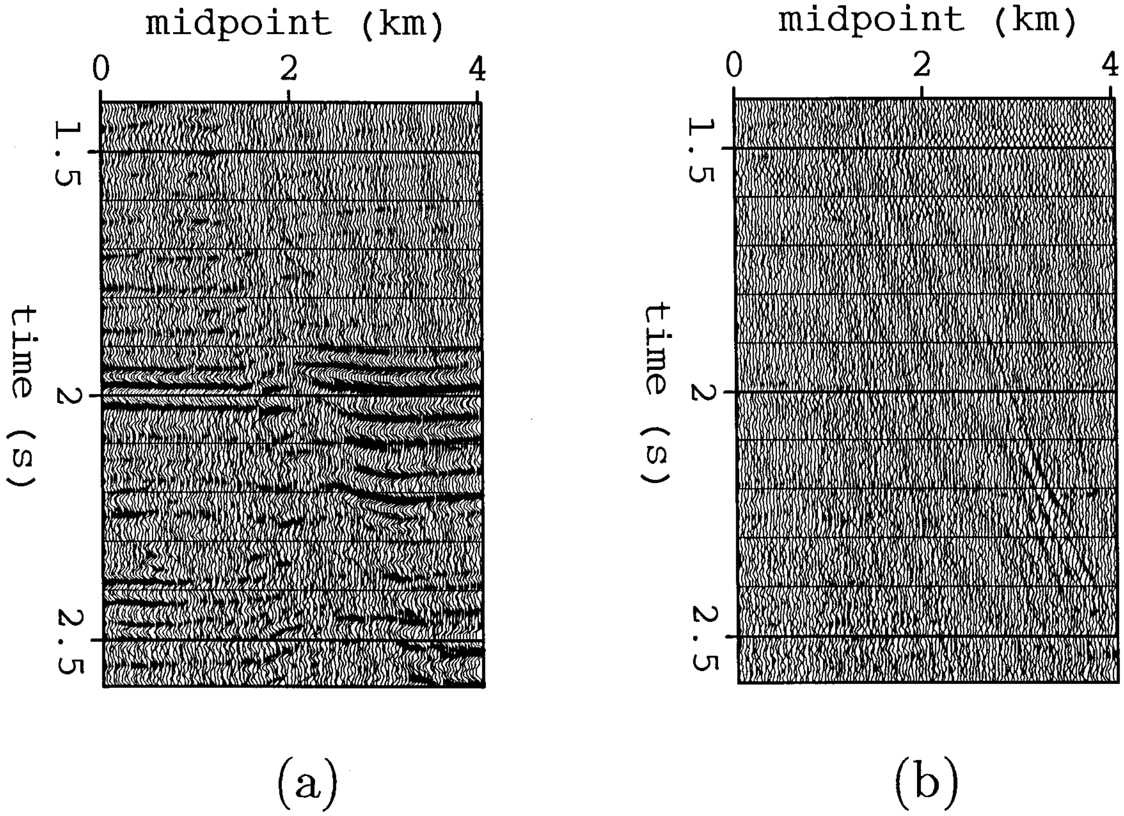


FIG. 2.10. (a) Constant-velocity stack of Gulf coast data at 2.20 km/s. Flat beds stack well, but the fault-plane reflection is missing. (b) Constant-velocity stack of Gulf coast data at 2.69 km/s. The fault-plane reflection is apparent at 2.2 seconds, but the flat beds are attenuated.

The velocity function used for stacking the Gulf coast data is shown in Figure 2.12. Because the velocity is only weakly varying laterally, and the reflectors have only shallow dip, the same velocity function is also used for migration. These data are used here to exemplify the action of the velocity-space algorithm, not because they pose an unusual velocity analysis problem. Note that the velocity contours generally trend down toward the left of Figure 2.12, although the reflecting beds dip upward toward the left. This is an example of how the long-wavelength background velocity model can be partially decoupled from the short wavelength reflectivity image.

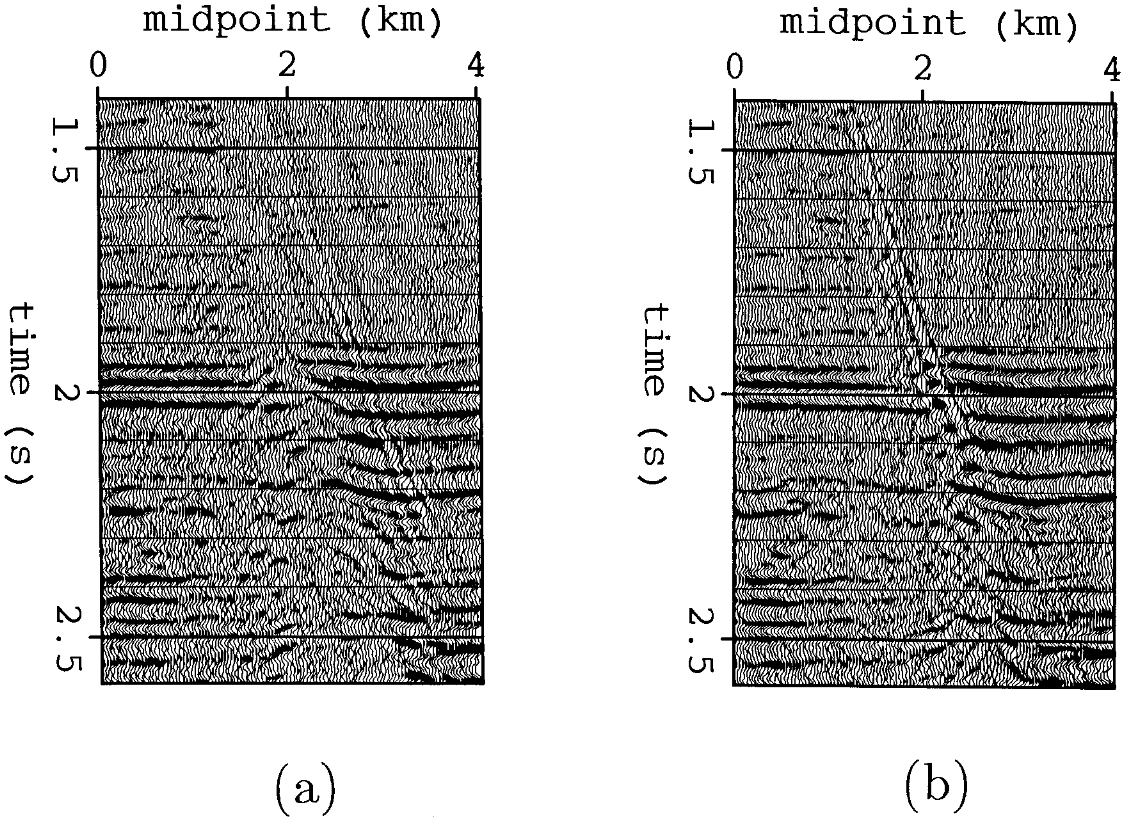


FIG. 2.11. (a) Constant-velocity stack of Gulf coast data at 2.20 km/s after DMO correction. Both the flat beds and the dipping fault-plane reflection stack well now. (b) Constant-velocity stack of Gulf coast data at 2.20 km/s after DMO correction and migration. The diffraction tails are collapsed, and the dipping fault-plane reflection aligns better with the bed truncations.

Offshore California data example

Figure 2.13 shows a portion of a data set from off the coast of southern California. This image is extracted from a suite of 61 stacks. Figure 2.14 shows the application of velocity-space DMO to these data, and Figure 2.15 shows the migrated result. Because of the complicated folding and faulting of the sediments, interpretation of these data is greatly simplified by migration. The structural complication, combined with a dipping sea floor, means that the velocity field is not laterally invariant, but the velocity-space algorithm has nonetheless produced a focused image in most places. No deconvolution or multiple suppression was applied to these data, which has two side effects. The first is that a strong sea floor multiple remains visible cutting across the geologic structure. The second is that the measured velocities are systematically too low; the sea floor reflection has an apparent velocity of about

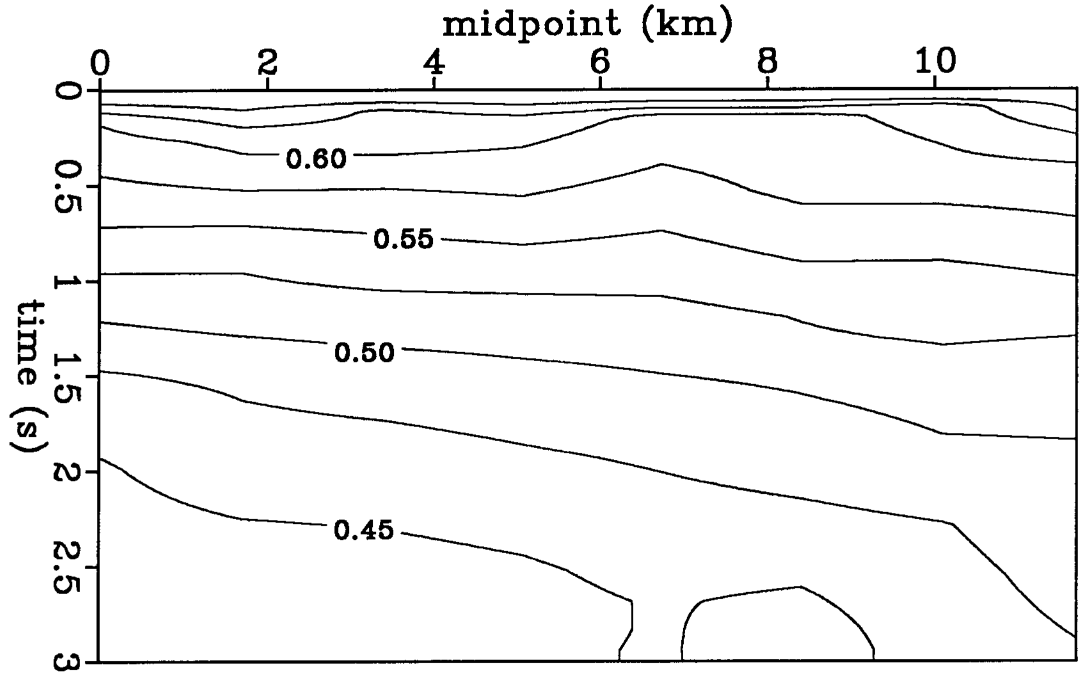


FIG. 2.12. Velocity function used for stacking and migrating the Gulf coast data of Figures 2.7 to 2.11. Contours are in slowness, with an increment of 0.025 s/km.

1.46 km/s, which is lower than that of water. The energy source for this survey was a pair of water guns, with a weak first event followed by a second strong, sharp pulse. This time lag makes all moveout velocities appear low; it can be corrected by proper deconvolution, or by time-shifting the data before velocity analysis.

Figure 2.16 shows the velocity function used for extracting the stacked section on Figure 2.13. For comparison, Figure 2.17 shows the velocity function used for making the migrated section in Figure 2.15. In both cases, what is plotted is not the velocity but its reciprocal, the slowness, for reasons discussed further in section 2.6. Note that for this example, the velocity contours follow the general dip of the beds, and that the velocity contours steepen during migration just as the dipping beds do. In both figures, there is little control on the velocities below about 2.3 seconds because there are few coherent reflections, so the velocity contours shown there mean little. In the next section, I examine some portions of these data in greater detail.

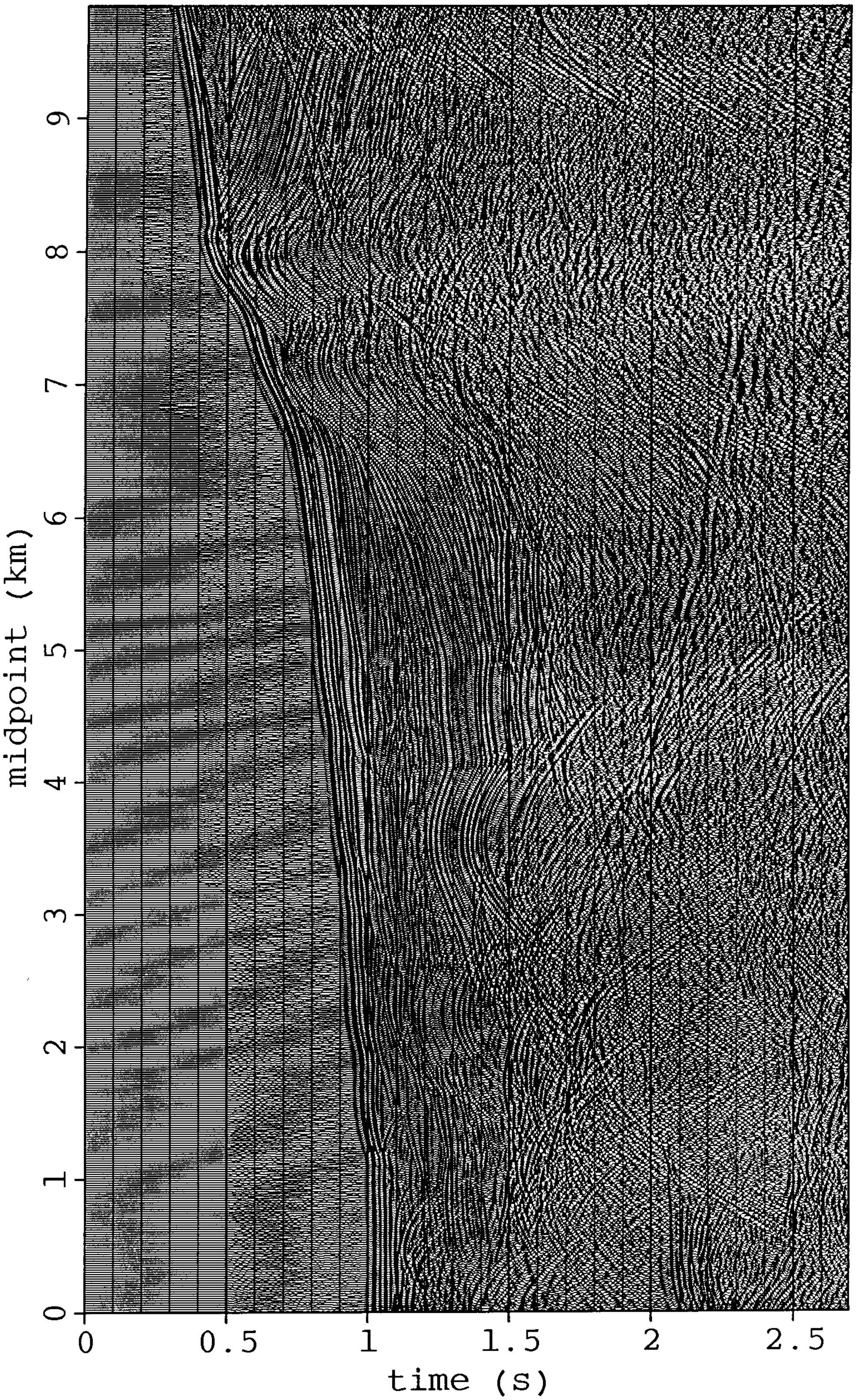


FIG. 2.13. Portion of a data set from offshore California, extracted from a suite of 61 constant-velocity stacks.

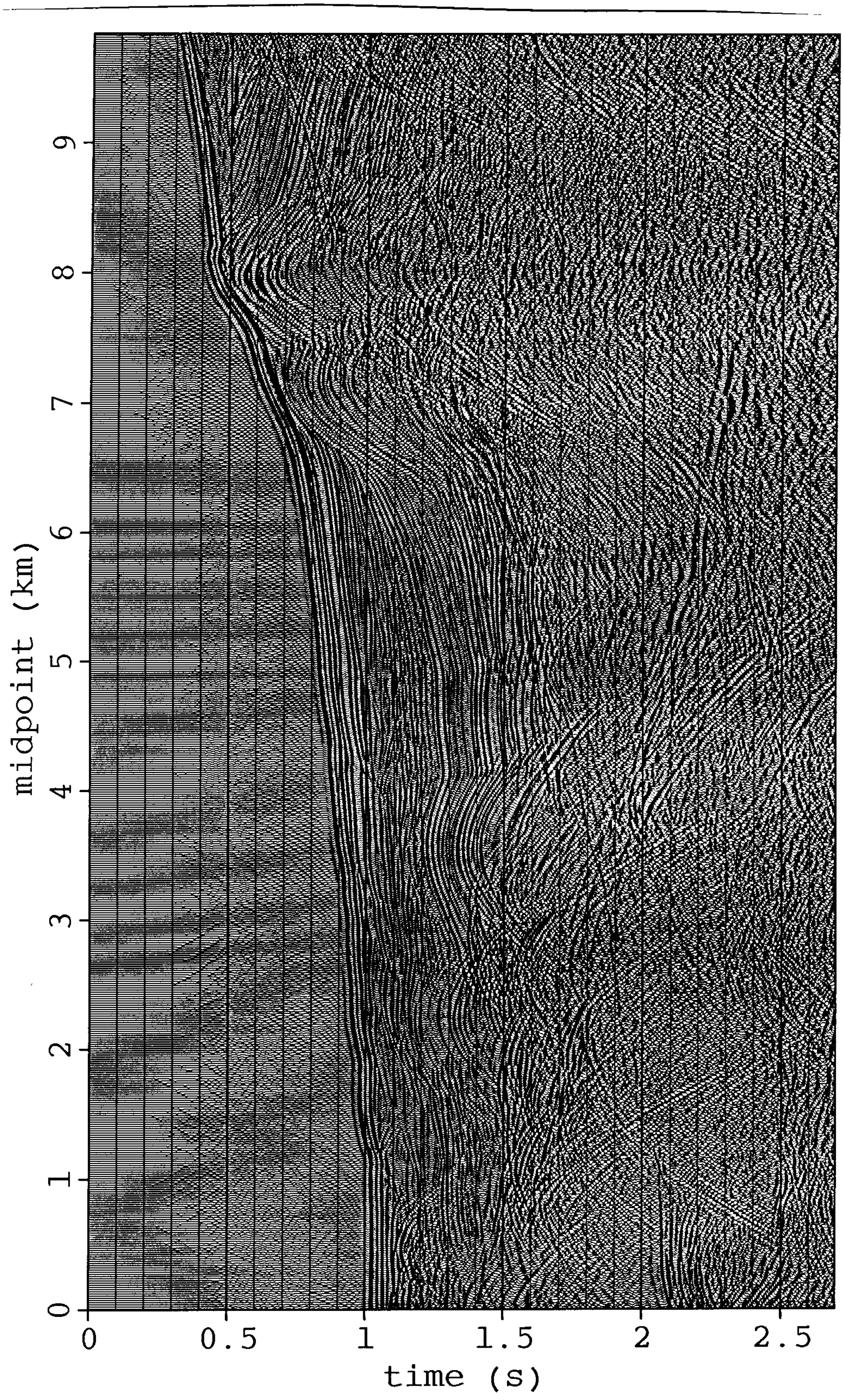


FIG. 2.14. Portion of a data set from offshore California, extracted from a suite of 81 constant-velocity stacks after DMO correction.

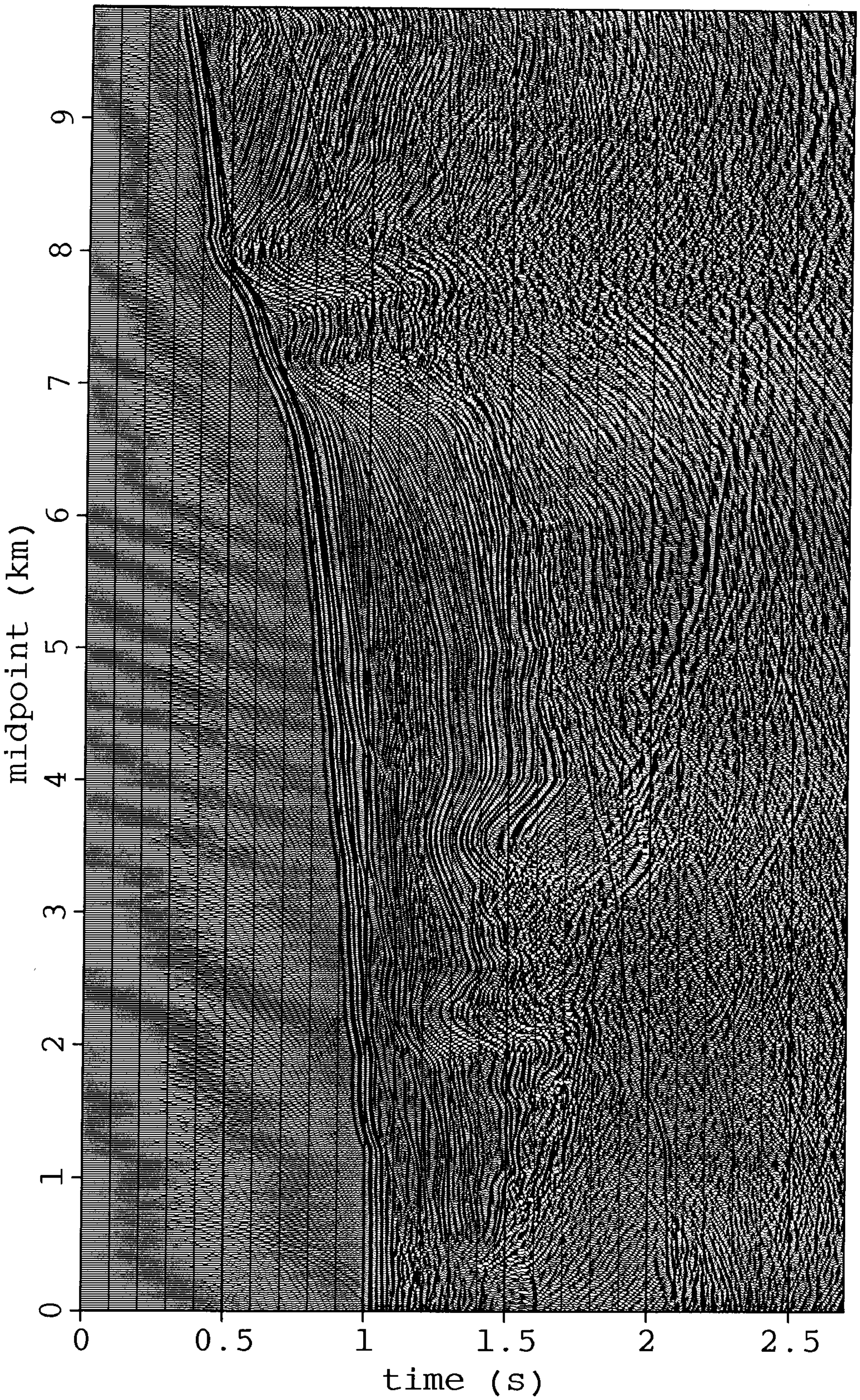


FIG. 2.15. Portion of a data set from offshore California, extracted from a suite of 81 constant-velocity stacks after DMO correction and migration.

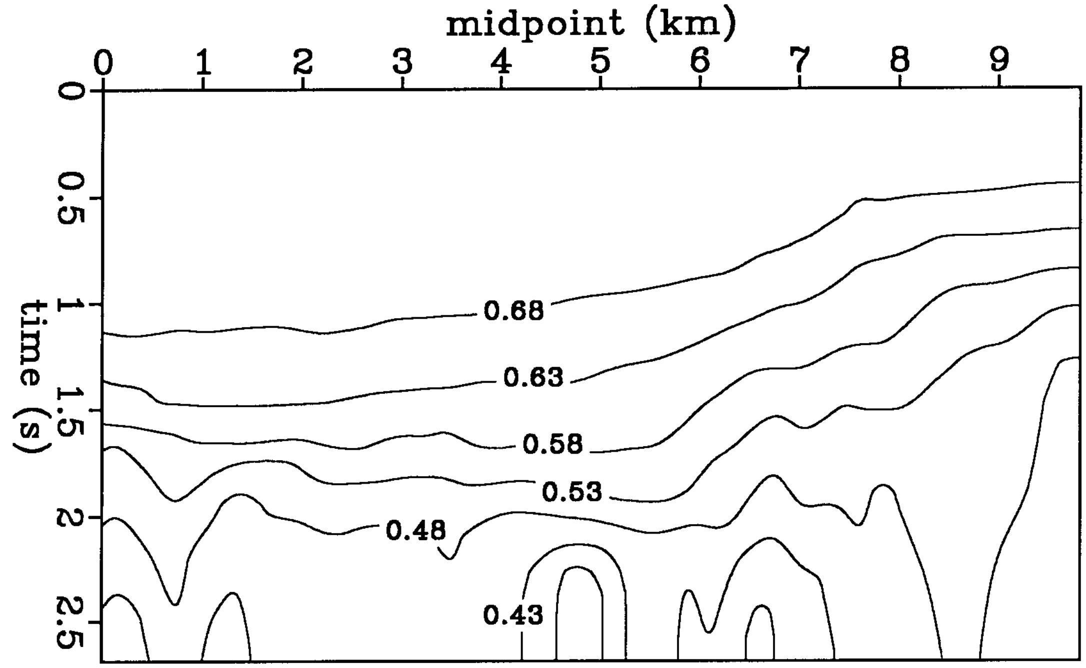


FIG. 2.16. Stacking velocity function for the offshore California data set. This velocity function was used to produce Figure 2.13. Contours are in slowness, with an increment of 0.05 s/km.

2.5 VELOCITY ANALYSIS USING VELOCITY-SPACE IMAGING

The velocity-space processing algorithms unify imaging and velocity analysis, so that DMO and migration are done *before* any commitment to a velocity function. Velocity analysis is performed on data that is already partially or fully imaged; the large discrepancies between stacking velocities and migration velocities that can arise because of dip are suppressed. Moreover, extraction of an image by interpolation is computationally trivial, so a wide variety of velocity functions can be tested, and the stacked or migrated images corresponding to each can be examined and compared.

DMO-corrected or migrated stacks can be used for velocity analysis just like constant-velocity stacks. Figure 2.18a shows the stacked traces for a range of velocities at the midpoint 3.35 km from the left end of the Gulf coast line shown in Figures 2.7 to 2.11. The velocity trend is clearly defined. To make the velocities easier to pick, the data values are squared and the resulting energy values are smoothed over a short time window and contoured for plotting. Such a plot derived from Figure 2.18a is shown in Figure 2.18b.

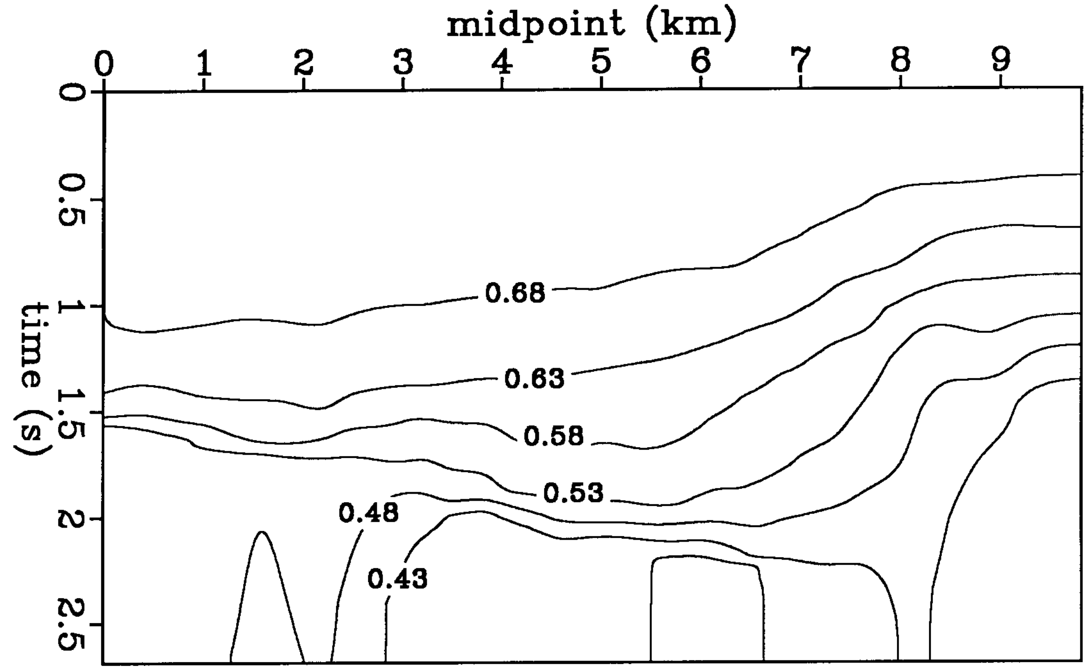


FIG. 2.17. Migration velocity function for the offshore California data set. This velocity function was used to produce Figure 2.15. Contours are in slowness, with an increment of 0.05 s/km.

Similar analyses may be produced equally readily for the DMO-corrected data (Figure 2.19) and for the DMO-corrected and migrated data (Figure 2.20). In each of these figures the flat beds define a clear velocity function. In Figure 2.18 a high velocity feature is visible at 2.2 seconds, separated from the dominant velocity trend. This event is caused by the dipping fault plane visible in Figure 2.10b. The application of DMO in Figure 2.19 moves this energy back to the velocity trend defined by the flat beds. In Figure 2.20 this fault plane energy is no longer visible; migration shifts it laterally, as seen in Figure 2.12b, so that it no longer contributes to the velocity analysis for this midpoint.

Velocity analysis in variable-velocity media

In a layered medium, the stacking velocities of flat beds will be close to the rms average of the interval velocities. For velocity that is a function of depth only, the DMO and migration velocities will also be close to the rms velocities, even for dipping beds. Figure 2.21 shows velocity analysis panels for stacking, DMO, and

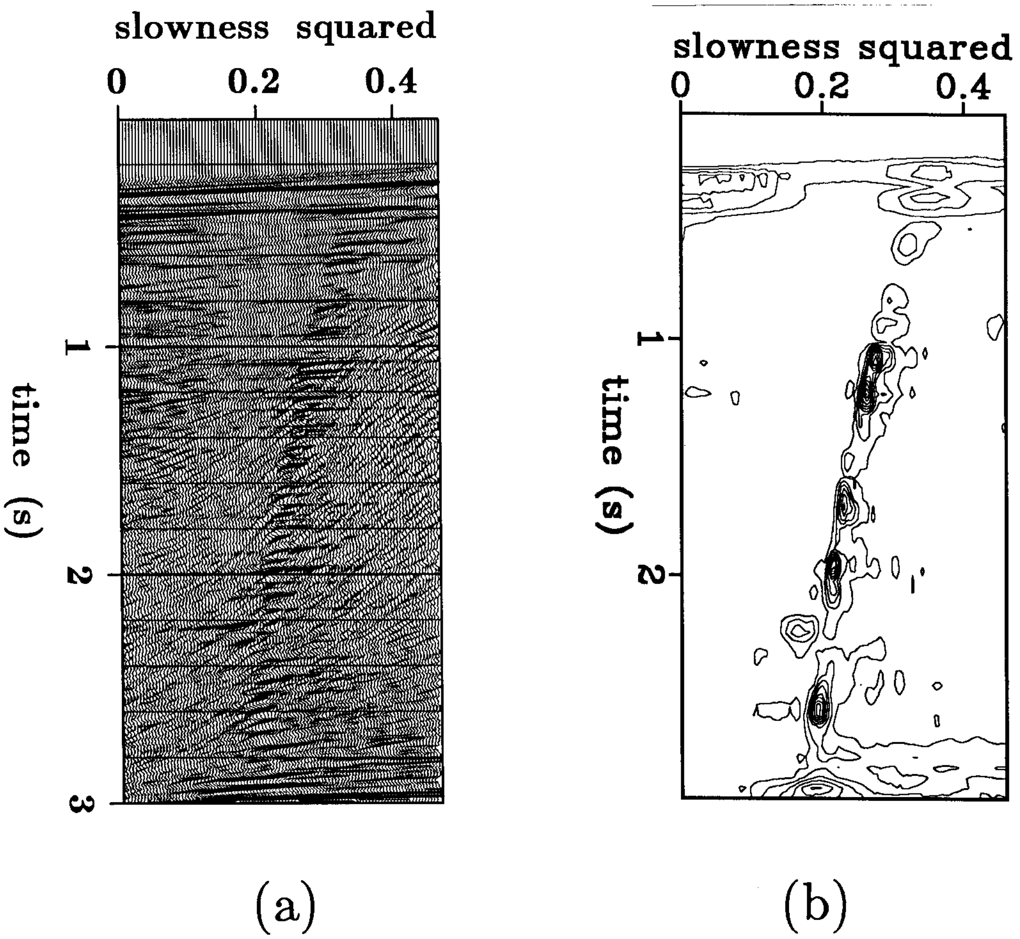


FIG. 2.18. (a) Stacked traces for a range of squared slownesses at the midpoint 3.35 km from the left side of Figure 2.7. (b) Contour plot of energy in Figure 2.18a. The stacking velocity function is easy to pick by connecting the energy peaks. The peak at 2.2 seconds at a higher velocity then the main trend is caused by a dipping fault plane reflection.

migration of the synthetic data shown in Figures 2.5 and 2.6. The velocity analyses are all at the point 2.2 km from the left edge. Overlaid on each panel is the rms velocity function. The flat bed at 2.05 seconds lies on the rms curve in all the panels. The 45 ° dipping bed yields a stacking velocity much too high, as expected. DMO correction gives a velocity slightly lower than the rms curve, and migration slightly higher.

Why are the DMO and migration velocities not exactly the same as the flat bed (rms) velocities? The velocity space algorithm can remove the large bias caused by

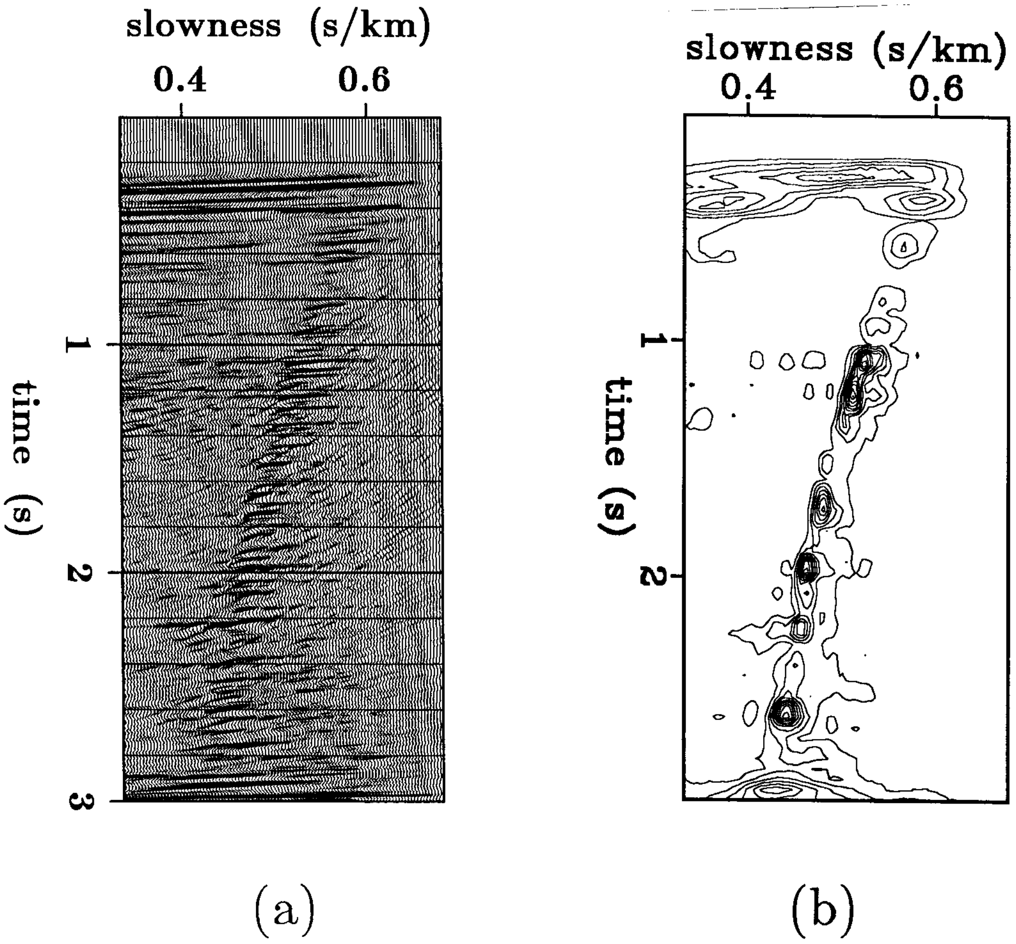
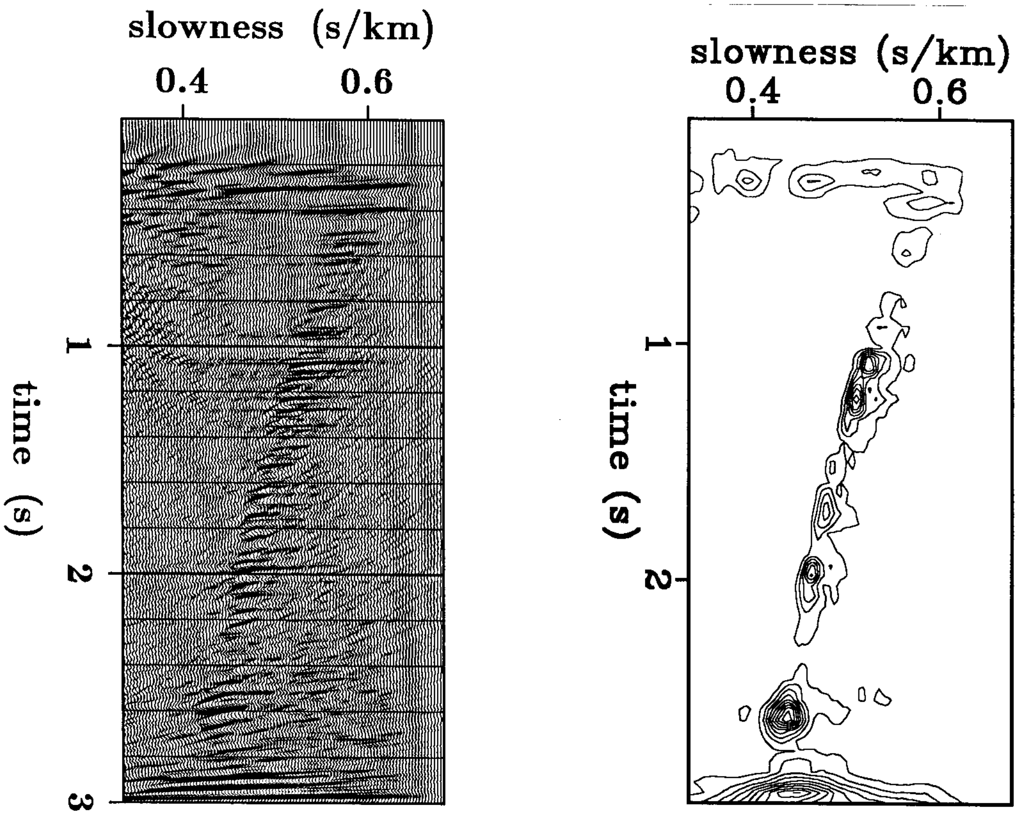


FIG. 2.19. (a) Stacked and DMO-corrected traces for a range of slownesses at the same midpoint as in Figure 2.18. (b) Contour plot of energy in Figure 2.19a. The fault plane reflection at 2.2 seconds now has the same stacking slowness as the flat beds.

structure, but it can not correct exactly for ray curvature effects. The dipping bed rays travel at shallower average angles than the flat bed ones, and so are more affected by ray curvature. The apparent dip (dt/dx) is measured from surface data, and ray curvature causes the apparent emergent angle to differ from that in a constant velocity medium. Also, the migration velocity will be higher than the DMO velocity because migration moves the velocity information carried by a reflector segment updip; the unmigrated velocity information belongs higher up, in a zone of lower velocity. None of these effects is usually large (Hale,1983), and the DMO-corrected or migrated velocity functions are far closer to being single valued than are



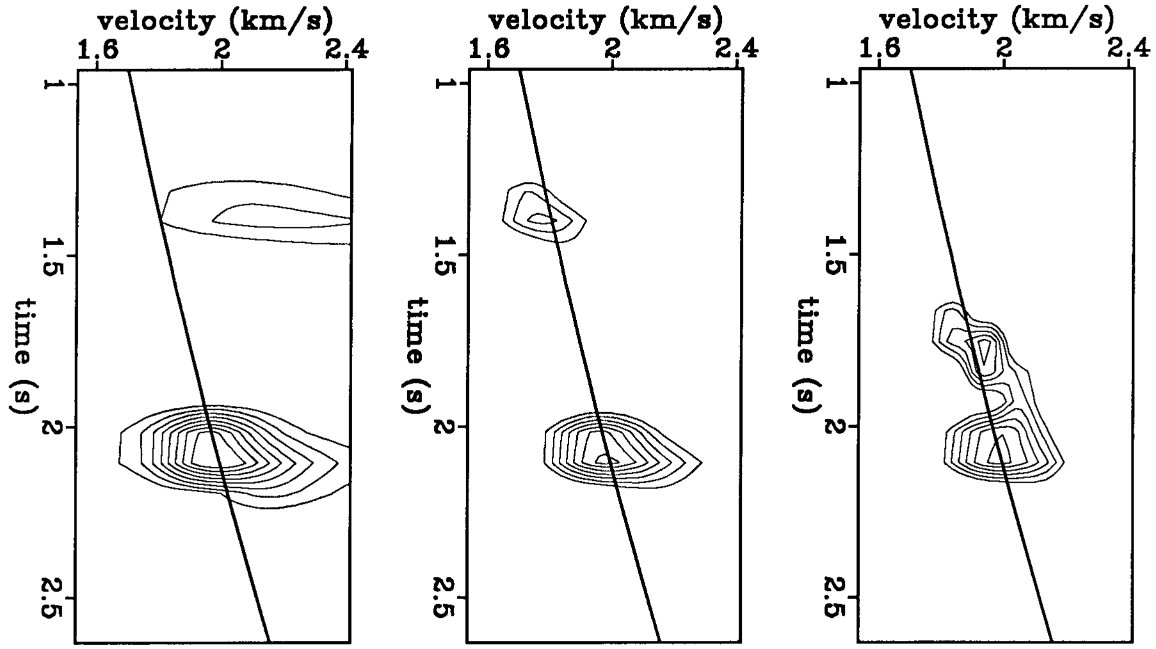
(a) (b)

FIG. 2.20. (a) Stacked, DMO-corrected, and migrated traces for a range of slownesses at the same midpoint as in Figures 2.18 and 2.19. (b) Contour plot of energy in Figure 2.20a. The fault plane reflection at 2.2 seconds is moved laterally by migration so that it no longer is visible in this velocity analysis.

the uncorrected stacking velocities.

Coherence measures for stacking

One practical problem with velocity analyses based on energy is the dominance of strong reflectors over weak, but equally coherent, events. A standard method for dealing with this problem is to replace measurement of energy in the stack with measurement of semblance. Let the NMO-corrected data for a particular midpoint be written as $q_{ij} = q(t_i, h_j)$. Then the energy in the stack at that midpoint is given by



(a) (b) (c)

FIG. 2.21. (a) Stacking velocity analysis for data of figure 2.5, at the 2.2 km midpoint. Solid curve is the rms average of the interval velocities. Contour lines indicate energy in the velocity analysis. Dipping bed event at 1.4 seconds is at far too high a velocity. (b) Same velocity analysis after DMO. Dipping bed event is now at slightly too low a velocity. (c) Same velocity analysis after DMO and migration. Dipping bed event is now at slightly too high a velocity.

$$E_i = \left(\frac{1}{N_h} \sum_{j=1}^{N_h} q_{ij} \right)^2 \tag{2.8}$$

where N_h is the number of offsets in the sum. If the energy is smoothed over a window extending L samples before and after a particular sample point, this becomes

$$E_i = \frac{1}{N_h^2(2L+1)} \sum_{k=i-L}^{i+L} \left(\sum_{j=1}^{N_h} q_{kj} \right)^2 . \tag{2.9}$$

Semblance normalizes this energy sum by dividing it by the sum of the energy in each input trace window:

$$S_i = \frac{E_i}{\frac{1}{N_h(2L+1)} \sum_{k=i-L}^{i+L} \sum_{j=1}^{N_h} q_{kj}^2} \tag{2.10}$$

$$= \frac{\sum_{k=i-L}^{i+L} \left(\sum_{j=1}^{N_h} q_{kj} \right)^2}{N_h \sum_{k=i-L}^{i+L} \sum_{j=1}^{N_h} q_{kj}^2} . \tag{2.11}$$

Because computing semblance rectifies and smooths the signal the semblance data cannot be DMO-corrected and migrated as the original data were. However, weighting the data before stack can provide many of the benefits of a semblance computation and still retain most of the waveform information. Let the weighted data q'_{ij} be given by

$$q_{ij}' = q_{ij} \left(\frac{1}{N_h} \sum_{k=1}^{N_h} q_{ik}^2 \right)^{-1/2} . \tag{2.12}$$

Then the energy, computed using equation (2.8) applied to these weighted data, will be equivalent to semblance if no smoothing over a time window is done. With smoothing over time, the two will be similar, although not identical. An even simpler weighting scheme is to apply automatic gain control (AGC) to the data before stack, weighting the data, for example, by the energy in a time window around each data point. Both of these weighting methods result in stacks in which equally coherent events contribute similarly to velocity analysis even if one is much weaker in the original data.

Data that has been weighted or AGC'ed may be DMO-corrected and migrated. Migration does not exactly commute with gain, so the resulting image can be degraded, but the velocity analysis is typically much sharper. If need be, the data can be run through the velocity-space DMO and migration twice, once with weighting for a sharper velocity analysis, and once without for an improved image.

Automatic velocity analysis

In conventional stacking-velocity analysis, only a few selected midpoints are examined. For the velocity-space algorithms, all midpoints are processed, so considerably more information is available. This extra information may be used for either automatic or interactive velocity-analysis methods.

For the velocity analyses in this chapter, I hand picked velocities for every fiftieth midpoints and interpolated and smoothed these picked values to generate velocity models. It is excessively time consuming and error prone for a person to pick velocities from all the midpoints; an automatic optimization algorithm can be used instead. For an objective function to maximize, one could use the cumulative energy in the resulting image, tempered by penalty terms that favor smooth velocity models. A more sophisticated approach might follow the lead of Toldi (1985), who incorporated an inversion for interval velocities into a nonlinear optimization algorithm for stacking-velocity analysis, putting constraints on the physically more significant interval velocities rather than on the stacking velocities themselves. To extend Toldi's methods to migration velocities, however, requires detailed knowledge of how these velocities are related to the underlying interval velocities. This relationship is examined in chapter 3, and its use for inversion is explored in chapter 4.

Interactive velocity analysis

Another way to use the velocity information incorporated in the velocity-space data cube is to examine the processed data interactively. Using currently available high-speed graphics workstations it is possible to scan rapidly either over midpoint, or over velocity, estimating velocities by the amplitude peaks and by the geological plausibility of the image. Such scanning is particularly important for migration velocities, because small changes in migration velocities cause movement of events, a motion that the human eye is exceptionally adept at detecting. Once a velocity function is chosen, the corresponding image can be extracted rapidly by interpolation, allowing interactive updating of the velocity field.

As an example of how direct examination of migrated images can be used for velocity analysis, I examine two portions from the offshore California data shown in section 2.4. I show in Figures 2.22 and 2.23 a window of these data, imaged at several velocities; this is the same data window shown in Figure 1.1 as an example of migrated data. The strata seen here are highly contorted and faulted, and the velocity changes rapidly. The interpretation of the shape of the folds, and of the locations of faults, depends critically on the imaging velocity used. The migration velocity field in Figures 1.3 and 2.17, based on smoothed interpolation between velocities picked every 200 meters, loses some of this detail. In conventional processing, an error in the migration velocities used could impede understanding the geology; here, no commitment to velocity has been made, and an interpreter can help choose the velocities based on his or her knowledge of the expected structural style.

These figures represent the wavefield using variable intensity plots, rather than the variable-area wiggle trace plots used in Figures 2.13 to 2.15. Wiggle trace plots have limited dynamic range; without the elimination of half the traces and the application of AGC and other non-linear gain, Figures 2.13 to 2.15 would have large areas in which no data could be seen. The variable intensity raster plots have much wider dynamic range, and are much better at showing how the brightness (peak amplitude) of events changes as velocity is varied. A raster graphics screen can have much higher resolution and even greater dynamic range, allowing for detailed comparison of velocity panels.

A second window of the offshore California data is shown in Figures 2.24 and 2.25. This portion of the data is problematic to interpret. It is also an area where it is hard to pick velocity peaks. The first panel (upper picture in Figure 2.25) is the lowest slowness (highest velocity) of the four. In it only the sediments on the right side of the figure around 0.9 seconds appear to be imaging at their peak amplitude. A very high amplitude "bright spot" is seen at 8 km, under a bump on the sea floor, but the upward turning "smiles" from overmigration suggest that the velocity is too high here. In the next panel the area of bright sediments on the right moves up, and the bright spot images better than at any other velocity. Also, the sediments between about 7.2 and 7.5 km image reasonably well below about 0.8 seconds. In the third panel, the uppermost sediments on the right and in the middle are imaged, as are strata around 1 second on the left. Diffraction tails appear from the bright spot at 8 km. The last panel is at water velocity. The sea floor becomes extremely bright, as do the upper sediments on the right. Because of the disparity between the velocities and the dip directions seen in the sediments on the left and right, I believe that a steep fault separates them. The exact location of the fault is uncertain, because of the almost total absence of coherent reflections in the region between 7.5 and 8 km; this whole gap may correspond to a fault zone, or there may be more than one fault, as suggested by the irregularity of the seafloor. Energy panels for this region show little or no coherence below the patina of seafloor sediments, so the velocity contours in Figure 2.17 are interpolated through this gap. This interpolation gives a velocity near the bright spot that is too low, as evidenced by the diffraction tails posing as structure in Figure 2.15.

The bright spot that stands out in the high-dynamic-range intensity plots of Figures 2.24 and 2.25 is not prominent in Figure 2.15. I do not know the cause of this small area of high reflectivity. It may correlate with the location of a large fault, as suggested by the seafloor bump above it. It may be caused by gas accumulation, and it may also arise in part from focusing effects, including out-of-plane 3-D reflections. I do not know the location of these data (beyond the general description that they are from off the southern California coast), so I cannot speculate further on

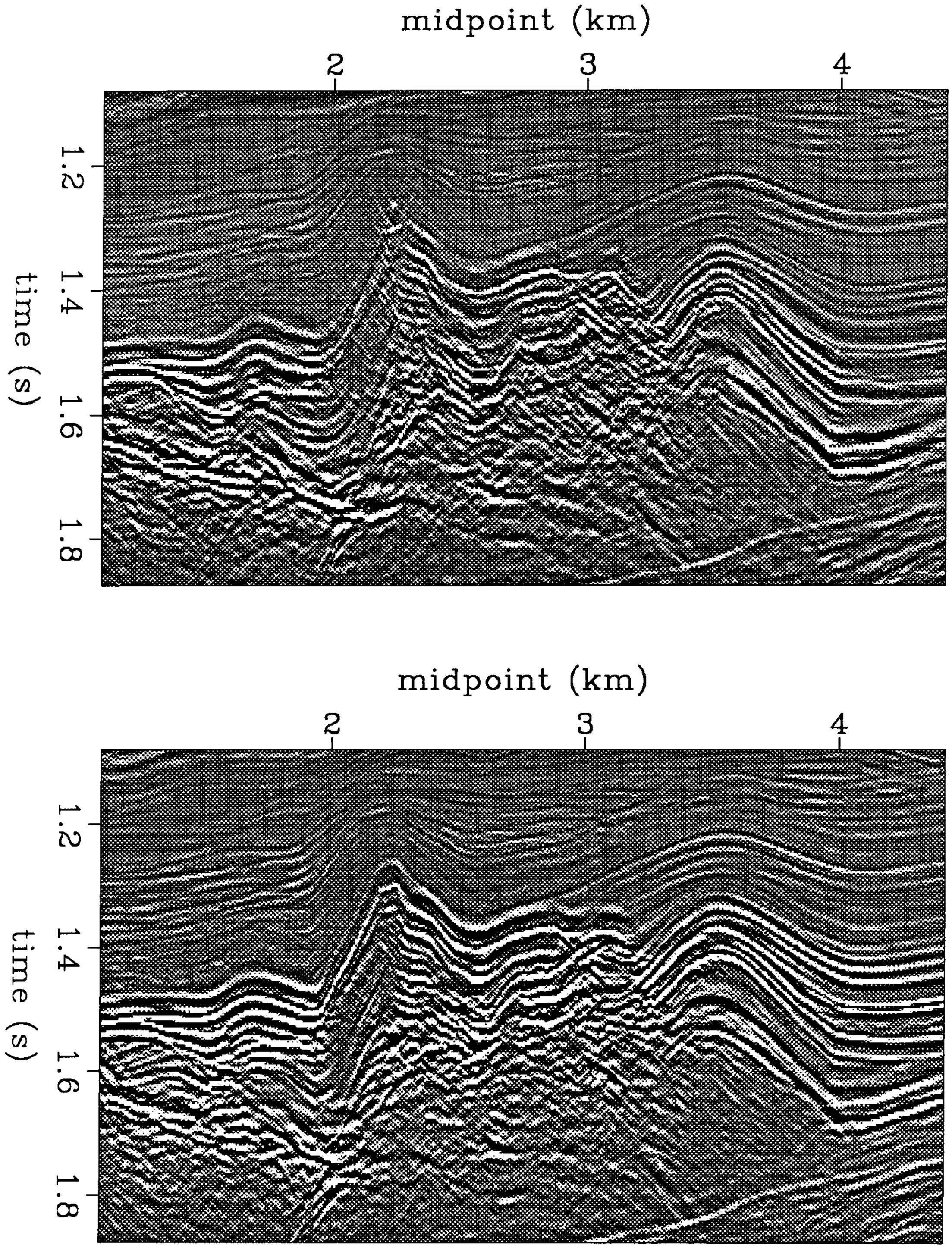


FIG. 2.22. DMO-corrected and migrated constant-velocity stacks of a portion of the offshore California data. The imaging slowness in the upper picture is 0.58 s/km and in the lower picture it is 0.61 s/km.

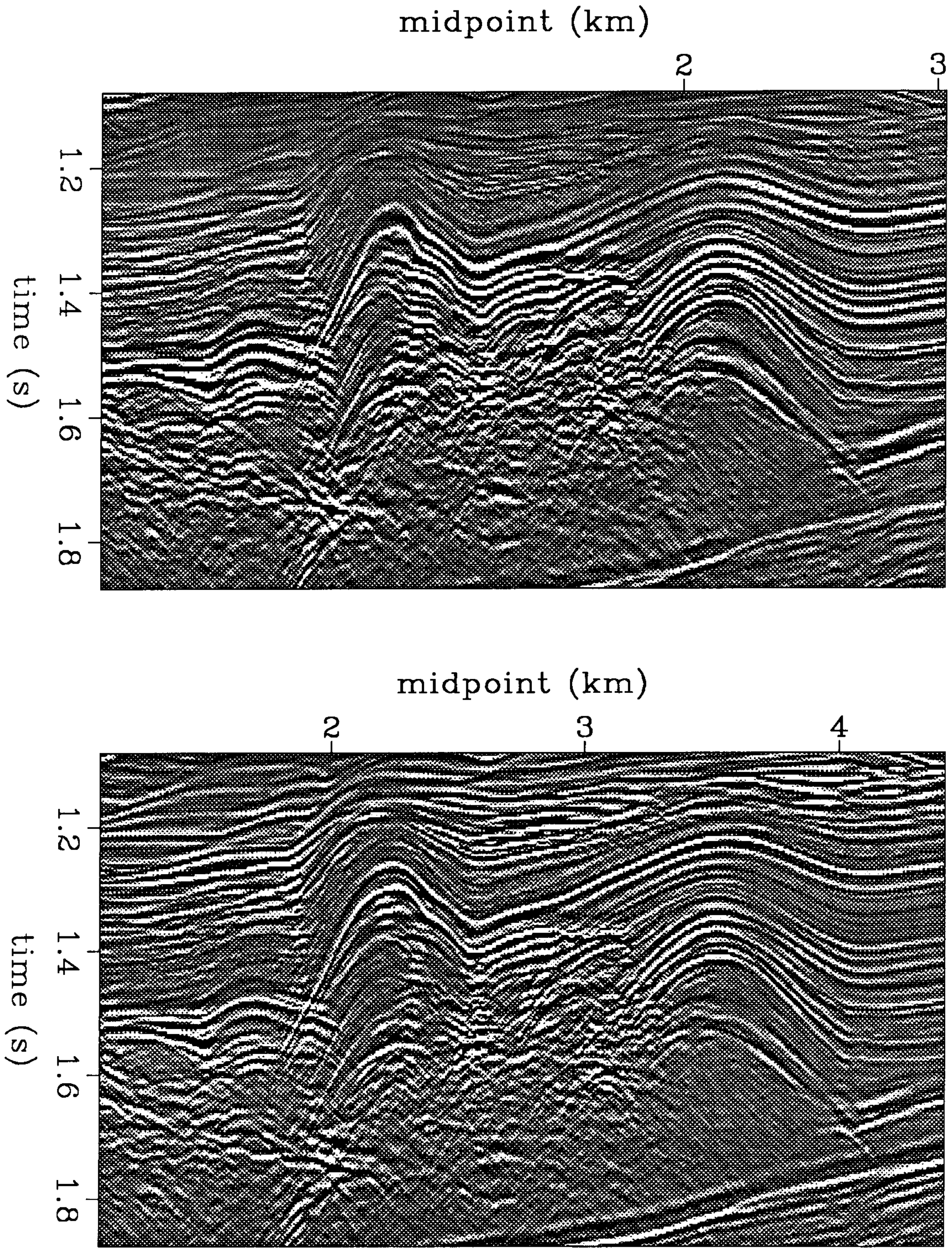


FIG. 2.23. DMO-corrected and migrated constant-velocity stacks of a portion of the offshore California data. The imaging slowness in the upper picture is 0.64 s/km and in the lower picture it is 0.67 s/km.

specific geologic interpretation.

As this last example of the offshore California data shows, interactive velocity analysis can use both focusing information (interpreted structure) and energy information (image brightness). In this example, velocity analysis using energy peaks gave little information about the area near 8 km in midpoint, but analysis based on focusing suggested a probable fault zone and a jump in velocity laterally. For some data, both forms of information might be useful, but might not agree. For data in which velocity varies substantially laterally, velocity analysis based on examining migrated images can result in velocities that differ from those chosen based on finding peak amplitude or energy, as noted in section 2.3. The velocity-space algorithm cannot directly compensate for these differences, since it uses time migration, not depth migration, and allows only a single velocity for both stacking and migration. However, if observed, these discrepancies may be used as an indicator that enough lateral velocity variation exists to warrant further efforts at careful interval velocity analysis and depth migration.

2.6 SAMPLING VELOCITY SPACE

The practicality of implementing the velocity-space algorithms depends directly on how many constant-velocity stacks must be processed. In this section I derive estimates of the number of stacks required.

Sampling strategy for constant-velocity stacks

Since velocity discrimination is much better for low velocities than for high ones, low velocities should be sampled at a denser rate than high velocities. Also, to be able to stack steeply dipping events, one needs to cover a velocity range that extends to an infinite velocity. For these reasons, I sample in slowness, the reciprocal of velocity, instead of velocity itself.

For a specified zero-offset time t_0 and a range of velocities v, normal moveout defines a suite of hyperbolic summation trajectories in offset-time space (h, t). One criterion for defining a good slowness sampling rate is to require that the time axis be sampled densely enough by this family of hyperbolas for all times and offsets. In terms of the slowness s the moveout equation (2.1) becomes

$$t^2 = t_0^2 + 4s^2h^2 \tag{2.13}$$

where h is half-offset, t_0 is zero-offset time, and t is time. All times are two-way travel times. For two moveout curves with the same zero-offset time t_0 but

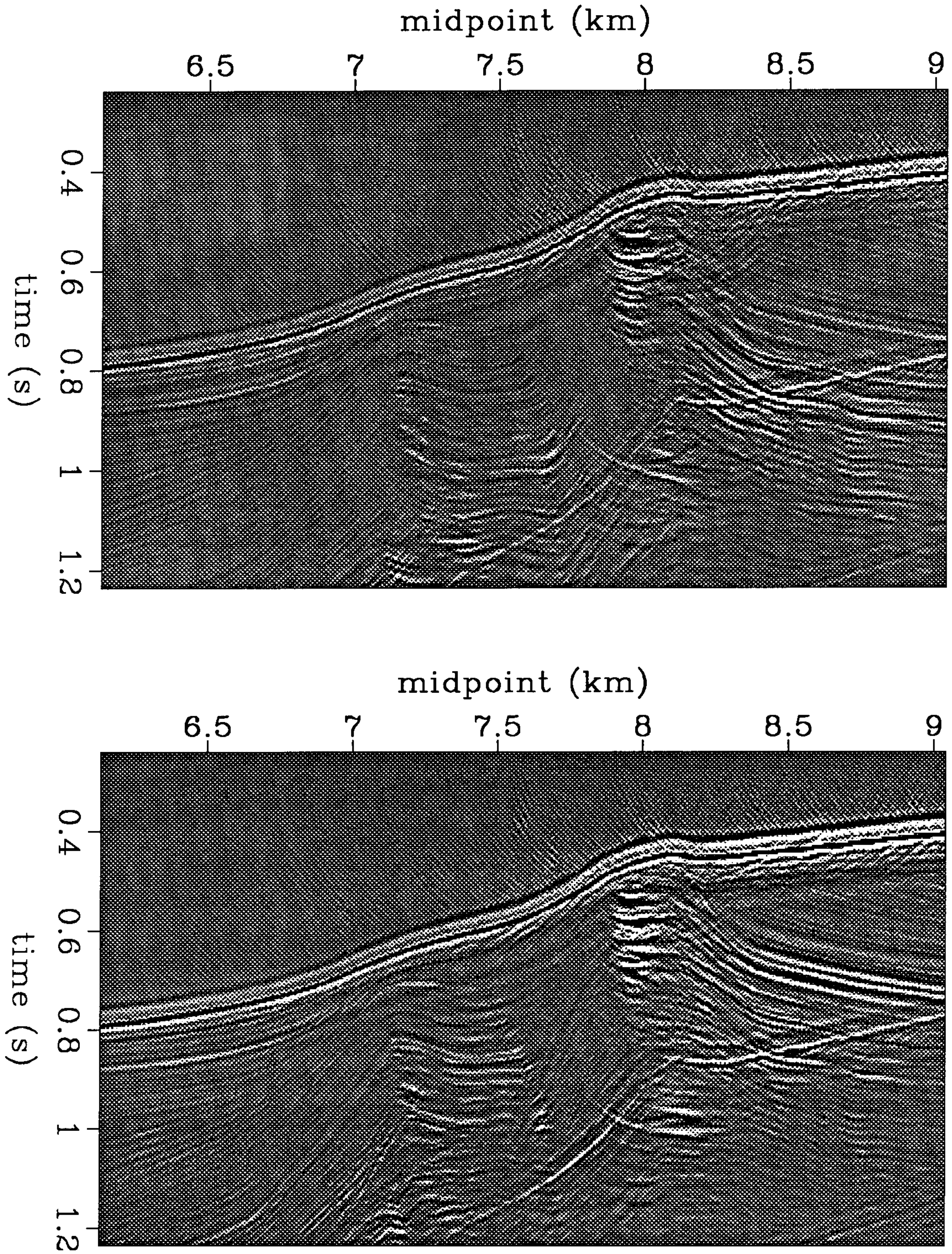


FIG. 2.24. DMO-corrected and migrated constant-velocity stacks of a portion of the offshore California data. The imaging slowness in the upper picture is 0.585 s/km and in the lower picture it is 0.62 s/km.

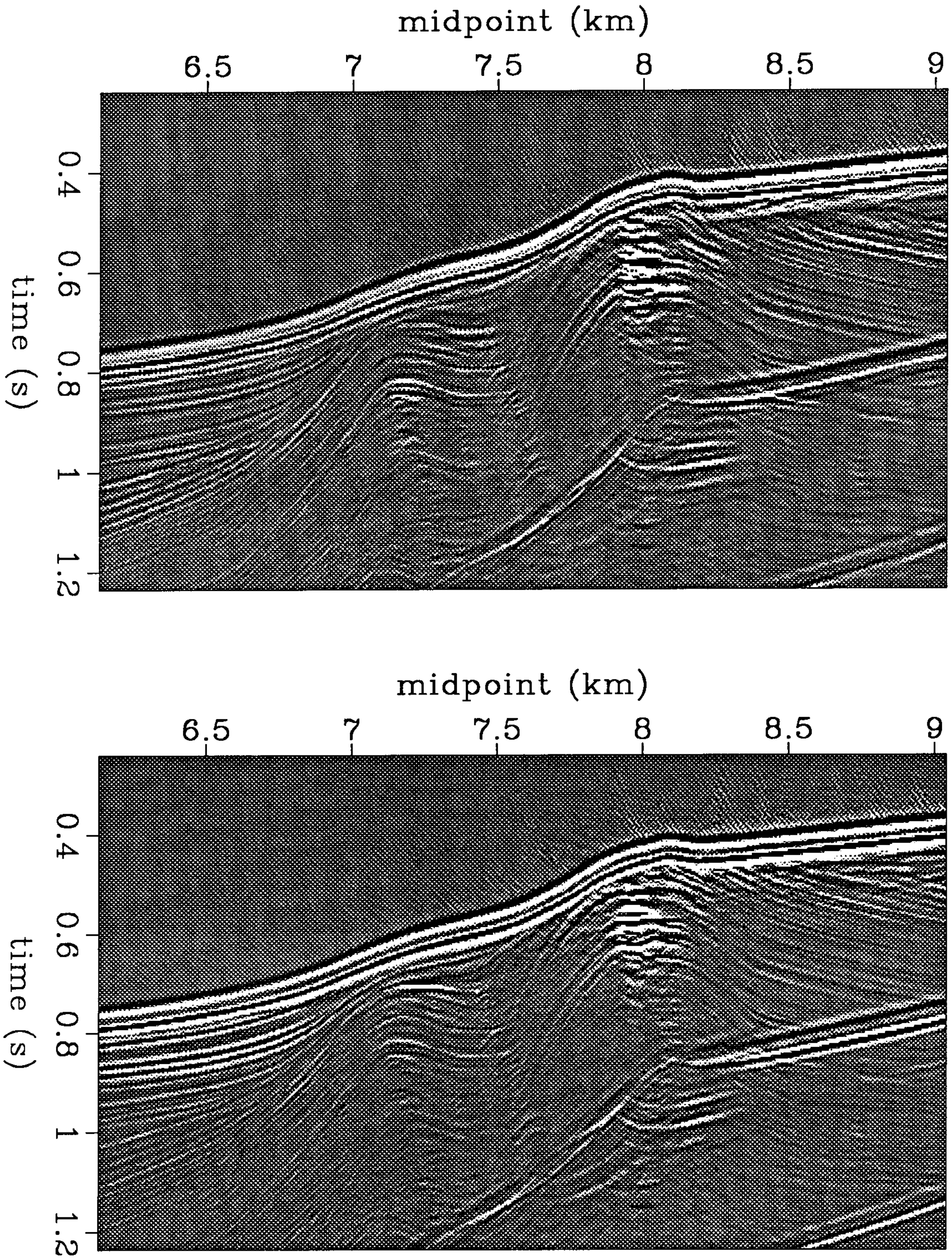


FIG. 2.25. DMO-corrected and migrated constant-velocity stacks of a portion of the offshore California data. The imaging slowness in the upper picture is 0.665 s/km and in the lower picture it is 0.685 s/km.

slownesses differing by Δs , one has

$$\Delta t \approx \frac{\partial t}{\partial s}\Delta s = \frac{4h^2 s}{t}\Delta s \ . \tag{2.14}$$

Note that for $hs >> t_0$, then $t \approx 2hs$ and equation (2.14) becomes

$$\Delta t \approx 2h\, \Delta s \ . \tag{2.15}$$

So for early times, large offsets, or low velocities, the time-axis sampling rate is proportional to the slowness sampling rate.

Sampling for dip moveout

Because velocity-space DMO shifts data only along the velocity axis, if velocity is sampled adequately in stacking, DMO should not require a higher sampling rate. In terms of slowness s the DMO equation (2.5) becomes

$$s = s_\theta \left(1 + \frac{k_y^{\,2}}{4 s_\theta^2\, \omega^2} \right)^{1/2} . \tag{2.16}$$

Note that the DMO operator can be used to interpolate the velocity axis in any fashion desired. The number and density of stacks can be changed during DMO, as can the sampling variable used (e.g., convert sampling in slowness to sampling in velocity, etc.) In particular, note that the high-velocity stacks are only needed to image steeply dipping events, and can be discarded after the DMO correction is applied. Thus, if some reasonable upper bound on expected velocities is known, fewer stacks need to be migrated or inverse Fourier transformed than are originally created.

Estimating sampling density for stacking

The difference between two hyperbolas with the same zero-offset time τ but different moveouts can be characterized by the time difference between them at a given offset. One wants, therefore, to ascertain how small Δs must be to keep the corresponding Δt below a given size.

For a specified ϵ, to ensure that $\Delta t < \epsilon$, equation (2.14) implies that an upper bound must be found for $4h^2 s / t$. This bound will exist, even for t approaching 0, since h will then also go to zero. For any non-zero offset h , the reflection data are muted before the direct arrival at the highest slowness s_{mute} , so $t/2h < s_{mute}$, and thus

$$\frac{4h^2 s}{t} \le \frac{4h^2 s_{mute}}{t} \le 2h_{\max} \ . \tag{2.17}$$

This gives

$$\Delta s \leq \frac{\epsilon}{2h_{\max}} \tag{2.18}$$

as an estimated bound on how finely s must be sampled to ensure that neighboring moveout hyperbolas differ in time at offset h by no more than a specified amount ϵ.

Sampling for migration

The amount events move during migration both in the lateral direction x and in migrated time τ also needs to be estimated. This movement Δx and $\Delta\tau$ can be expressed as a function of the sampling interval Δs :

$$\Delta x \approx \frac{\partial x}{\partial s}\Delta s = -\frac{t\sin\theta}{s^2}\Delta s \tag{2.19}$$

and

$$\Delta\tau \approx \frac{\partial\tau}{\partial s}\Delta s = \frac{t\sin\theta\tan\theta}{s}\Delta s \ . \tag{2.20}$$

For a derivation of these equations, see appendix F.

From equations (2.19) and (2.20), one gets the estimated bounds

$$|\Delta s| \leq \frac{s_{\min}^2}{t_{\max}\sin\theta}|\Delta x| \tag{2.21}$$

and

$$|\Delta s| \leq \frac{s_{\min}}{t_{\max}\sin\theta\tan\theta}|\Delta\tau| \ . \tag{2.22}$$

Note however, that migration of a dipping event also shifts the frequencies down by a factor of $\cos\theta$ (Stolt, 1978; Rothman et al., 1985), so that the allowable $\Delta\tau$ should be increased by a similar factor. Equation (2.22) then becomes

$$|\Delta s| \leq \frac{s_{\min}}{t_{\max}\sin^2\theta}|\Delta\tau| \ . \tag{2.23}$$

Sampling in squared slowness

Sometimes it is even more efficient to sample evenly in squared slowness, which samples low velocities more densely than does sampling evenly in slowness. Designate squared slowness by $\sigma \equiv s^2$. Then $\Delta\sigma = 2s\,\Delta s$, and the sampling estimates in equations (2.18), (2.21), and (2.23) become

$$\Delta\sigma \leq \frac{\epsilon s_{mute}}{h_{\max}} \tag{2.24}$$

$$|\,\Delta\sigma\,| \leq \frac{\sigma_{\min}}{t_{\max}\sin\theta}\,|\,\Delta x\,| \tag{2.25}$$

and

$$|\,\Delta\sigma\,| \leq \frac{\sqrt{\sigma_{\min}}}{t_{\max}\sin^2\theta}\,|\,\Delta\tau\,|\ . \tag{2.26}$$

Examples and discussion

To illustrate this section I use the same Gulf Coast data set used in Figures 2.7 through 2.9. The image was extracted from a series of stacks using even sampling in squared slowness to cover a range from 0.0 to 0.46 s^2/km^2 (water velocity to infinity). The acquisition parameters are: maximum offset $h_{\max}$ is 1.775 km, maximum time used here is 3 seconds, time-sampling interval is 4 ms, and midpoint spacing is 33.5 m. DMO was used to resample the slowness axis to even sampling in slowness. The range of slownesses used for migration is 0.333 s/km to 0.677 s/km, corresponding to a velocity range of 1.48 to 3.0 km/s. If the maximum allowable time shift Δt is set to the sampling interval, then data below 62 Hz will be shifted less than a quarter wavelength and will all add constructively during stacking. These data contain little useful information with higher frequency content. Applying the inequality estimate (2.24) with ϵ=0.004 suggests that one needs $\Delta\sigma \leq 0.00150$. To cover the slowness range from 0.0 to 0.67 with this sampling would require 300 stacks. However, this criterion is much too restrictive in practice. It is not necessary to stack every possible event perfectly, but only to be able to reconstruct events by good interpolation between stacks. A more reasonable bound is found by using ϵ=0.016 and $h = 0.5h_{\max}$, which corresponds to allowing the dominant frequencies in the data to shift up to a half wavelength at half the maximum offset. These parameters yield an estimate of 75 for the requisite number of stacks. Applying the migration estimates from equation (2.21) and allowing energy to move up to one trace laterally gives an estimate of 93 stacks. Similarly, applying equation (2.23) with $\Delta\tau$=0.016 s suggests using 194 stacks.

Figure 2.7 was generated using 101 stacks with a sampling rate of $\Delta\sigma$=0.0046 s/km. I now show that this is an adequate number for these data. Figure 2.26 shows a conventional NMO stack using the same variable velocity function used for interpolation in creating Figure 2.7. The data in this figure, and all subsequent figures in this section, are plotted with the same gain and clip parameters as Figure 2.7. Figure 2.27 shows the difference between Figure 2.7 and Figure 2.26, confirming that no more than 101 stacks are needed for these data to reproduce the quality of more conventional processing. The small differences that remain visible in Figure 2.27 are caused by differences in the weights, mutes, and interpolators used by the programs implementing the conventional NMO stack and the velocity space method. To verify in a different way that 100 stacks are adequate here, in Figure 2.28 I show the stacked section produced using twice the sampling density ($\Delta\sigma$=0.0023) as used for Figure 2.7. Figure 2.29 shows the difference between Figure 2.28 and Figure 2.7; using more stacks makes no improvement. Figure 2.30 shows the same data as Figure 2.7, extracted using the same velocity function, but created now by discarding every other stack before interpolation. Figure 2.31 shows that the difference between these two is minor, consisting mostly of weakened amplitude on a few events when fewer stacks are used. Those portions for which the velocity is farthest from any of the stacking velocities used are degraded most, but the total difference between the panels is slight.

These data were resampled during DMO to create 101 stacks evenly sampled at a rate of Δs =0.00344 to cover the range 0.333 to 0.677 s/km. Figure 2.8 shows the image extracted after DMO and migration using all 101 stacks, and Figure 2.32 shows the result using only every other stack; Figure 2.33 shows the difference between the two. Again, the difference is slight and I conclude that the lower sampling rate is adequate.

This data example suggests that the estimates derived using equations (2.18), (2.21), and (2.23) will be reasonable, but in practice too conservative. The worst errors caused by undersampling will be seen at early times in the section, and for steeply dipping events. Using 50 to 100 stacks provides an adequate sampling density for the data sets I have examined.

2.7 VELOCITY-SPACE IMAGING FOR THREE-DIMENSIONAL DATA

So far I have considered only two-dimensional data, collected along a single line that is assumed to be perpendicular to all structure. I now look at the practicality of extending these methods to three-dimensional data. For three-dimensional data, both the midpoint $\mathbf{y}=(y_1,y_2)$ and offset $\mathbf{h}=(h_1,h_2)$ become vectors with components in the x_1 and x_2 directions, as do their Fourier transforms, $\mathbf{k_y}$ and $\mathbf{k_h}$. The CVPM equation (2.7) becomes:

$$k_\tau = \frac{\omega}{2}\left[\left(1 - \frac{|\mathbf{k_y}+\mathbf{k_h}|^2 v^2}{4\omega^2}\right)^{1/2} + \left(1 - \frac{|\mathbf{k_y}-\mathbf{k_h}|^2 v^2}{4\omega^2}\right)^{1/2}\right]. \qquad (2.27)$$

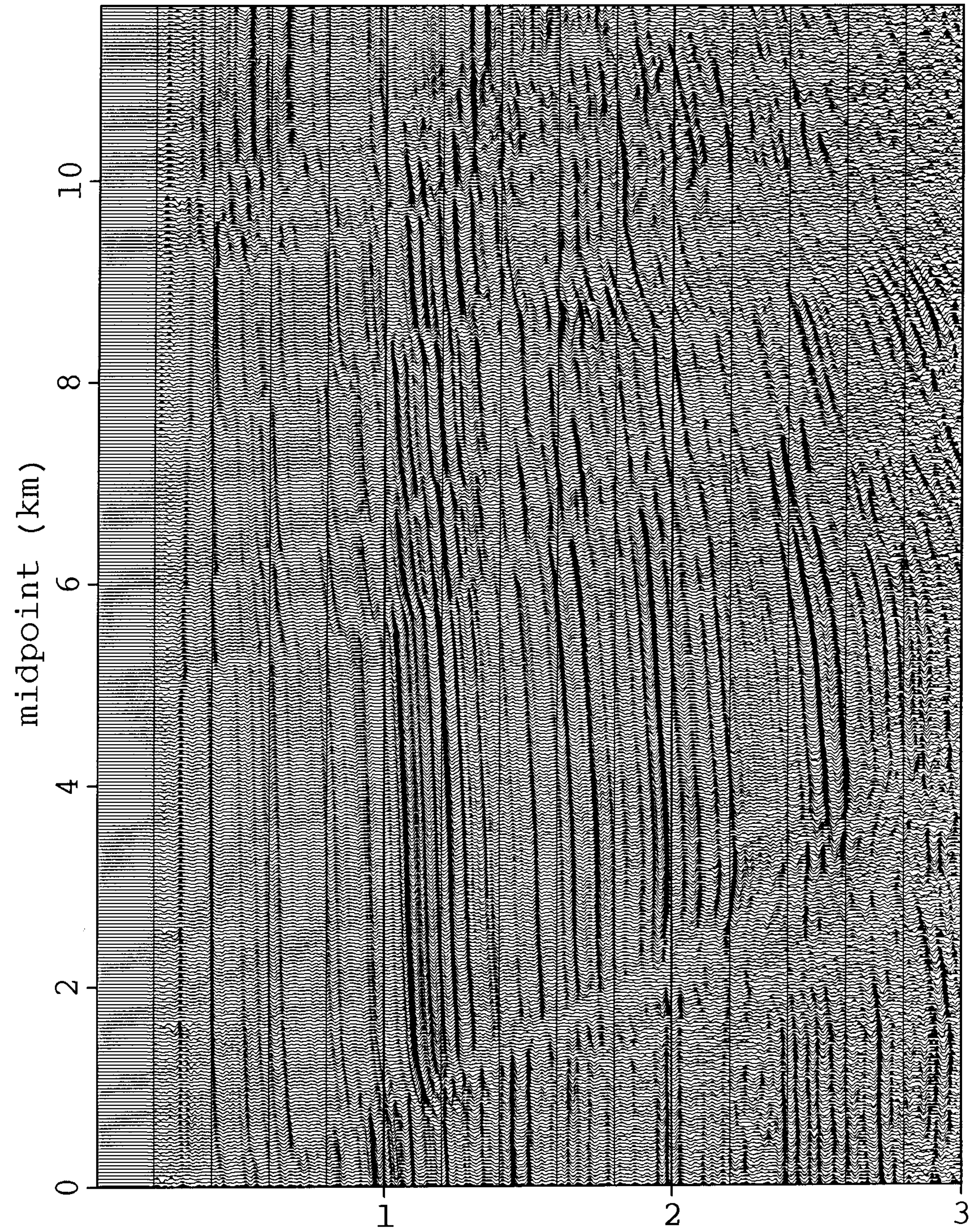


FIG. 2.26. Gulf coast data from Figure 2.7, generated using conventional NMO stacking.

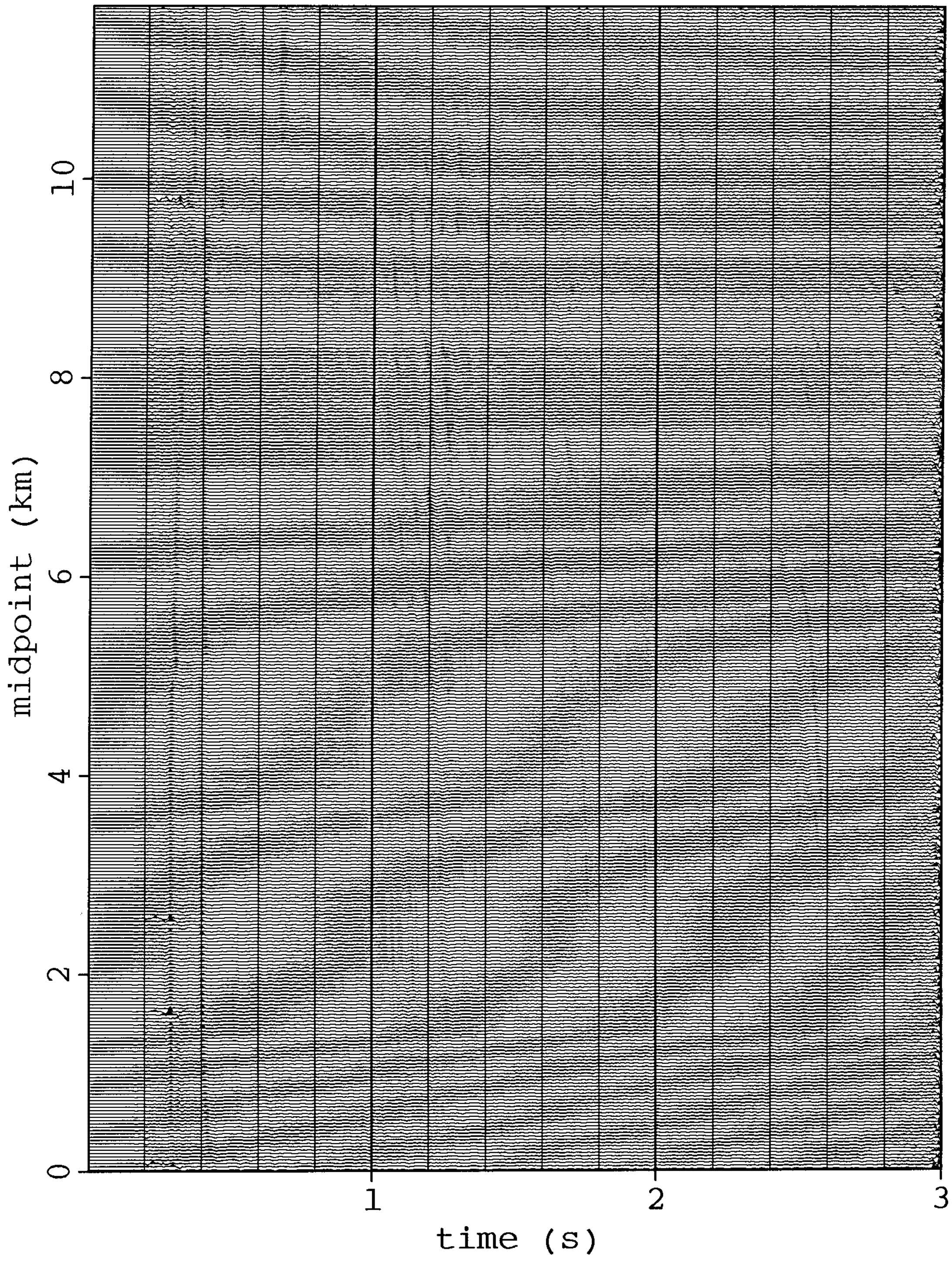


FIG. 2.27. Difference between Figure 2.7, which uses 101 stacks, and Figure 2.26, which uses conventional NMO stacking.

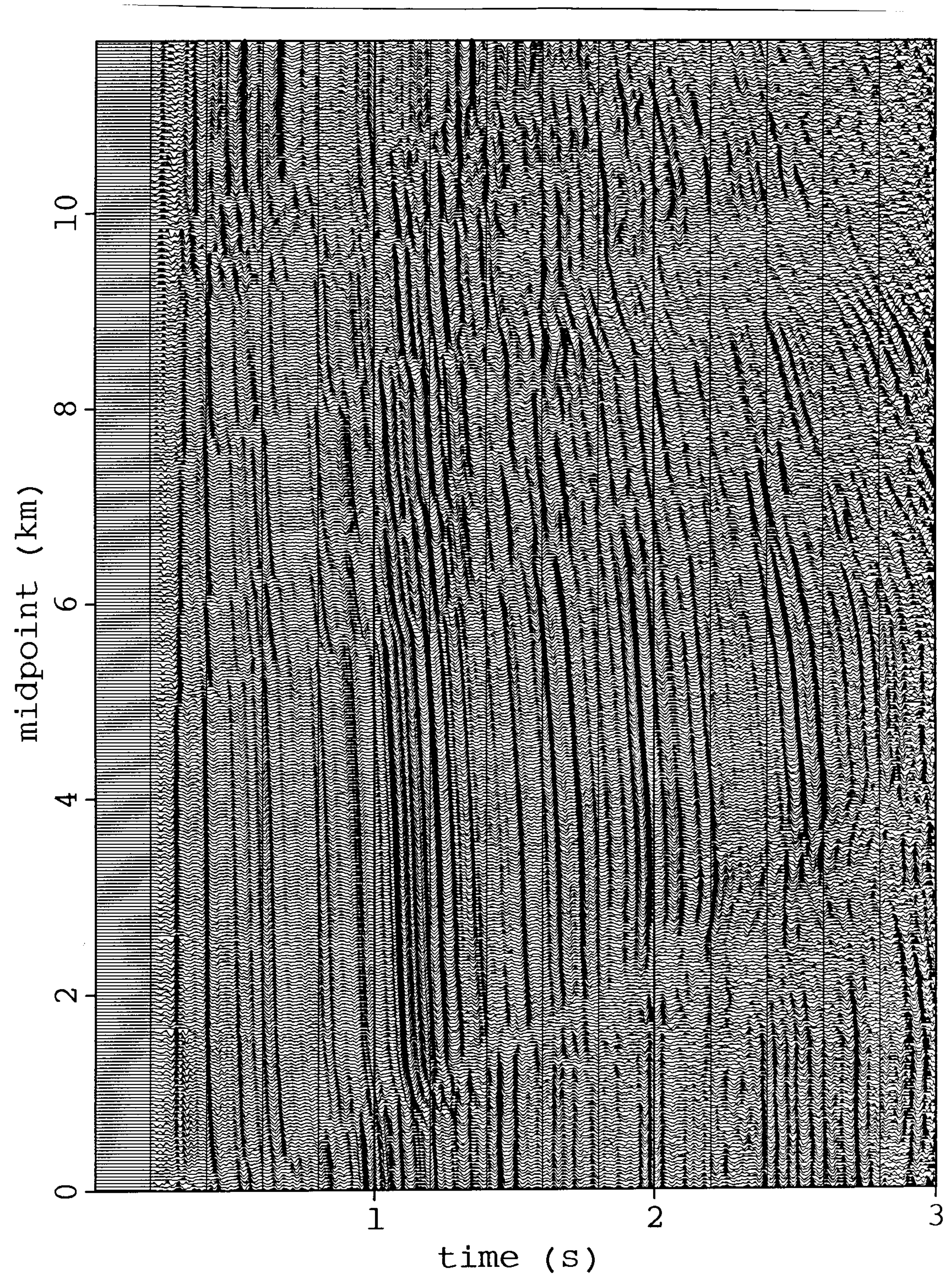


FIG. 2.28. Stacked section of Gulf coast data, extracted from a suite of 201 constant velocity stacks using the same velocity function as in Figures 2.7 and 2.26.

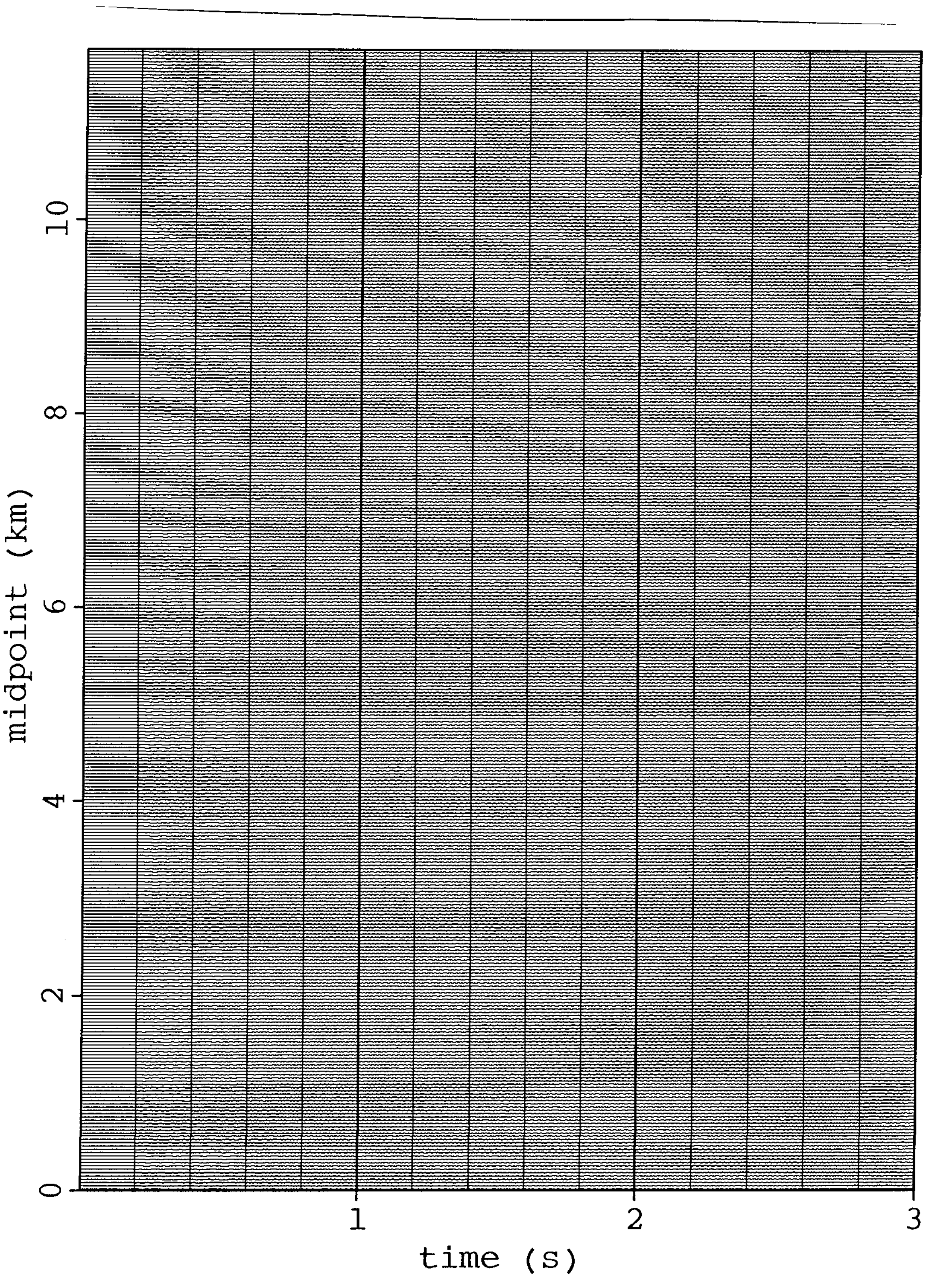


FIG. 2.29. Difference between Figure 2.7, which used 101 stacks, and Figure 2.28, which used 201 stacks.

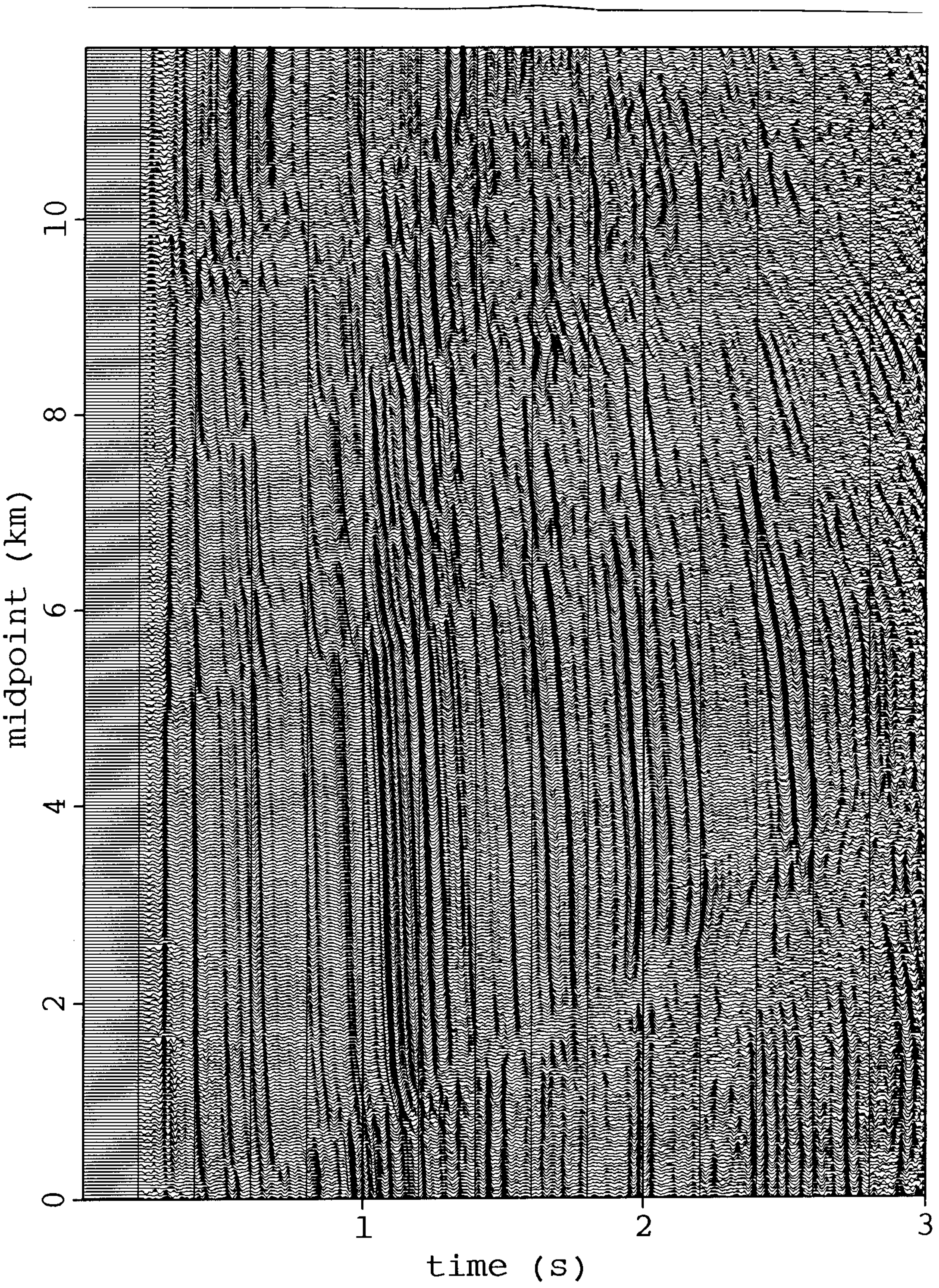


FIG. 2.30. Stacked section of Gulf coast data, extracted from a suite of 51 constant velocity stacks using the same velocity function as in Figures 2.7 and 2.26.

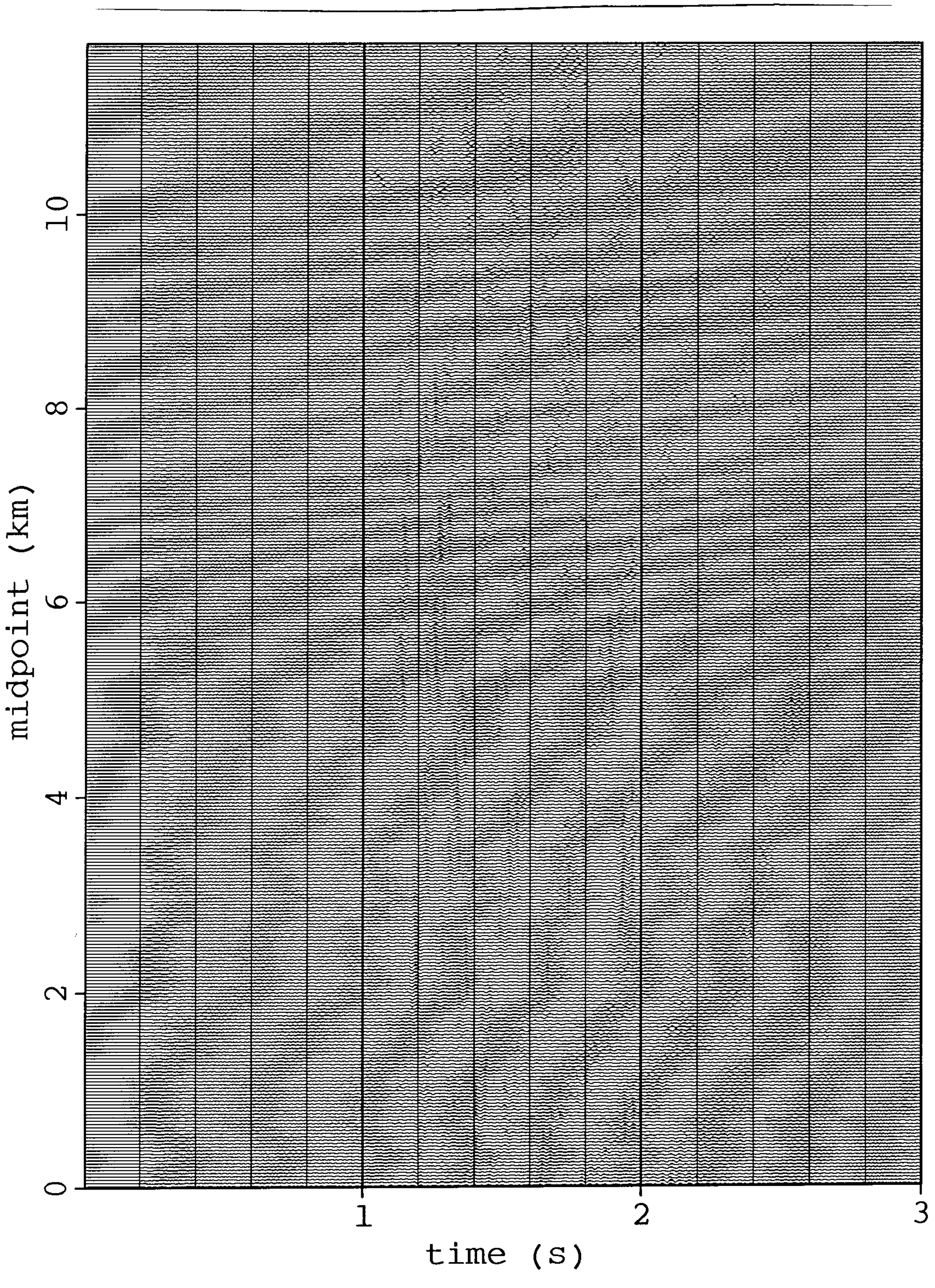


FIG. 2.31. Difference between Figure 2.7, which used 101 stacks, and Figure 2.30, which used 51 stacks.

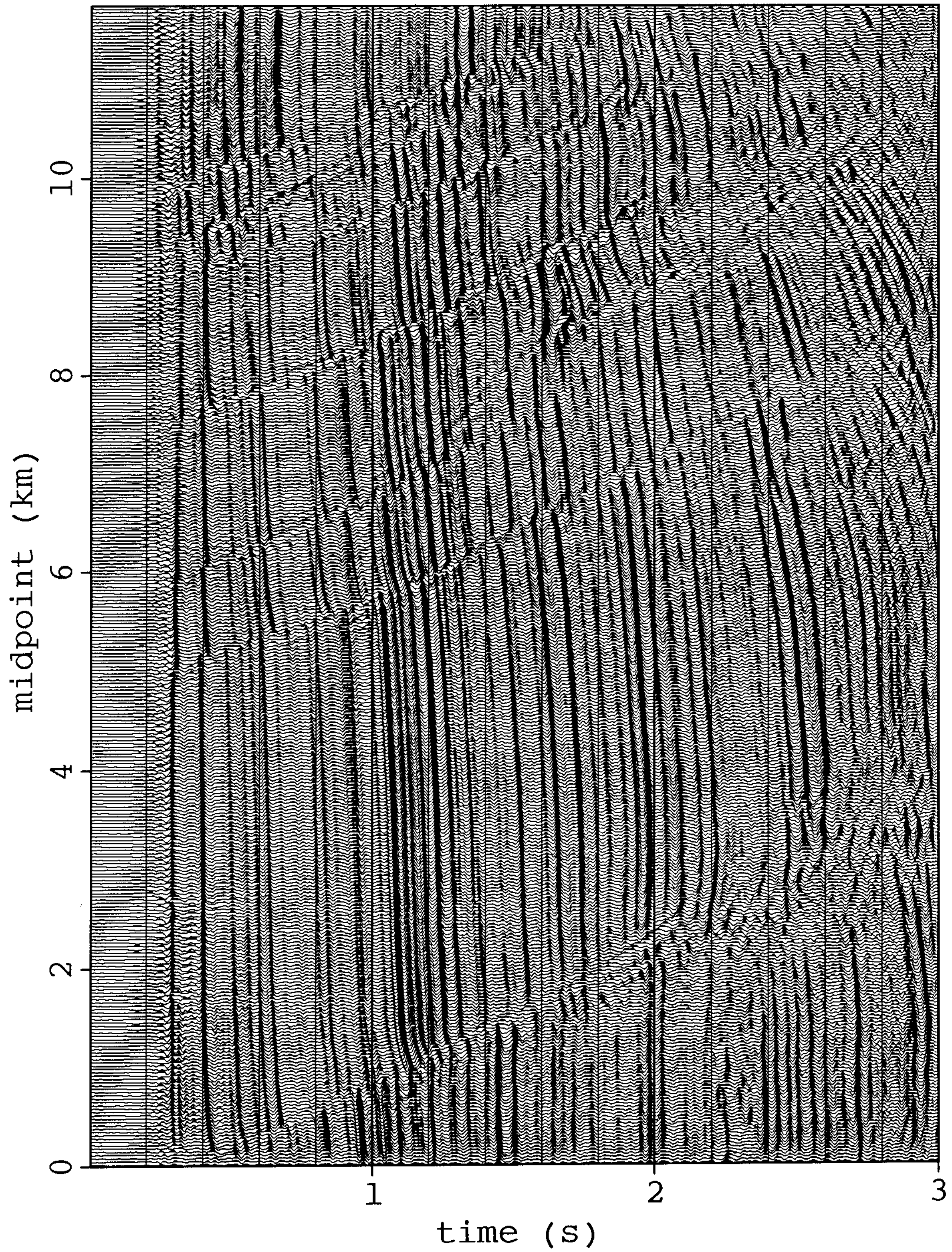


FIG. 2.32. Migrated Gulf coast data from Figure 2.9, extracted from a suite of 51 constant slowness stacks after DMO and migration using the same velocity function as in Figures 2.9. The slowness sampling rate is half that used in Figure 2.9.

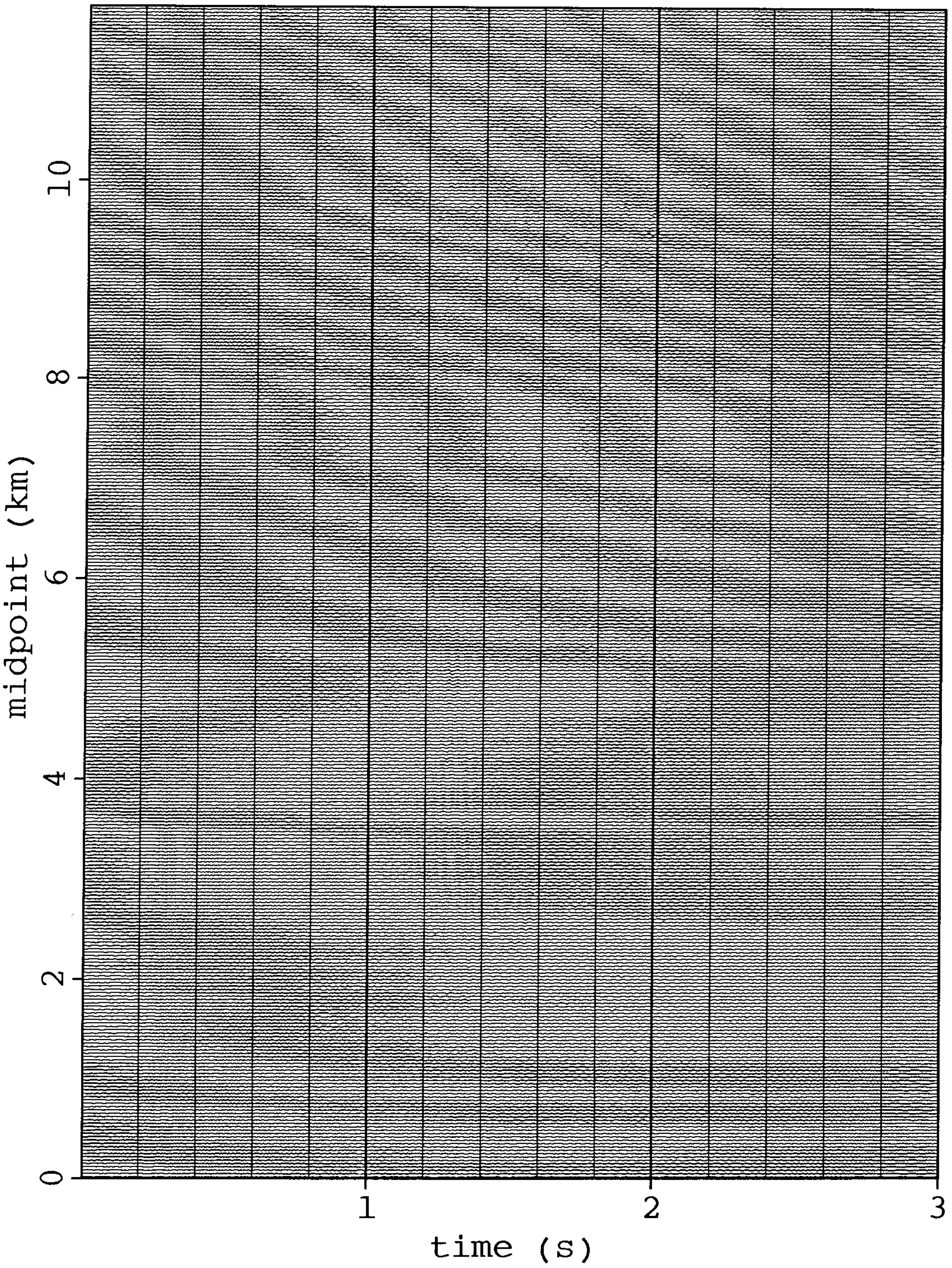


FIG. 2.33. Difference between Figure 2.9, which used 101 migrated stacks, and Figure 2.30, which used 51 migrated stacks.

As in the two-dimensional case, this equation can also be implemented in terms of shot and geophone coordinates instead of midpoint and offset.

For the CVS method, the transformation to velocity space in two dimensions exchanges the offset coordinate for a velocity coordinate. Generating constant-velocity stacks using all offsets for a given midpoint in three-dimensions collapses the two offset coordinates into a single velocity parameter. This loss of information is not readily recoverable. Instead, the stacks must be generated separately for different directions, trading the two offset coordinates now for the two parameters of velocity and azimuth.

Consider an upcoming plane wave reflected from a single dipping bed. Let θ again be the dip angle, and ϕ be the direction of dip, measured from the x_1 coordinate axis, as illustrated in Figure 2.34. Suppose an azimuthal direction ψ is chosen, again measured from the x_1 axis, and all traces with this shot to geophone offset azimuth are selected as a working data set. The apparent moveout velocity is (Levin, 1971)

$$v_{NMO} = v \left[1 - \sin^2\theta \cos^2(\phi-\psi)\right]^{-1/2} . \qquad (2.28)$$

This is the three-dimensional equivalent of equation (2.2). To apply it requires estimating both θ and ϕ from the data.

The Fourier domain expression for dip given in equation (2.3) needs only a simple change to allow for the dip direction; the equivalent relations are

$$\frac{vk_{y_1}}{2\omega} = \sin\theta \cos\phi \qquad (2.29)$$

and

$$\frac{vk_{y_2}}{2\omega} = \sin\theta \sin\phi . \qquad (2.30)$$

Solving for ϕ by dividing these two equations yields

$$\tan\phi = \frac{k_{y_2}}{k_{y_1}} . \qquad (2.31)$$

Thus, Fourier transforming the data over both x_1 and x_2 axes effectively decomposes the data into dip directions. ψ is known already, so the $\cos^2(\phi-\psi)$ term in equation (2.28) can be evaluated. Moreover, substituting into equation (2.31) for ϕ gives

$$\sin^2\theta = \frac{v^2 k_{y_2}^{\ 2}}{4\omega^2(k_{y_1}^{\ 2}+k_{y_2}^{\ 2})} . \qquad (2.32)$$

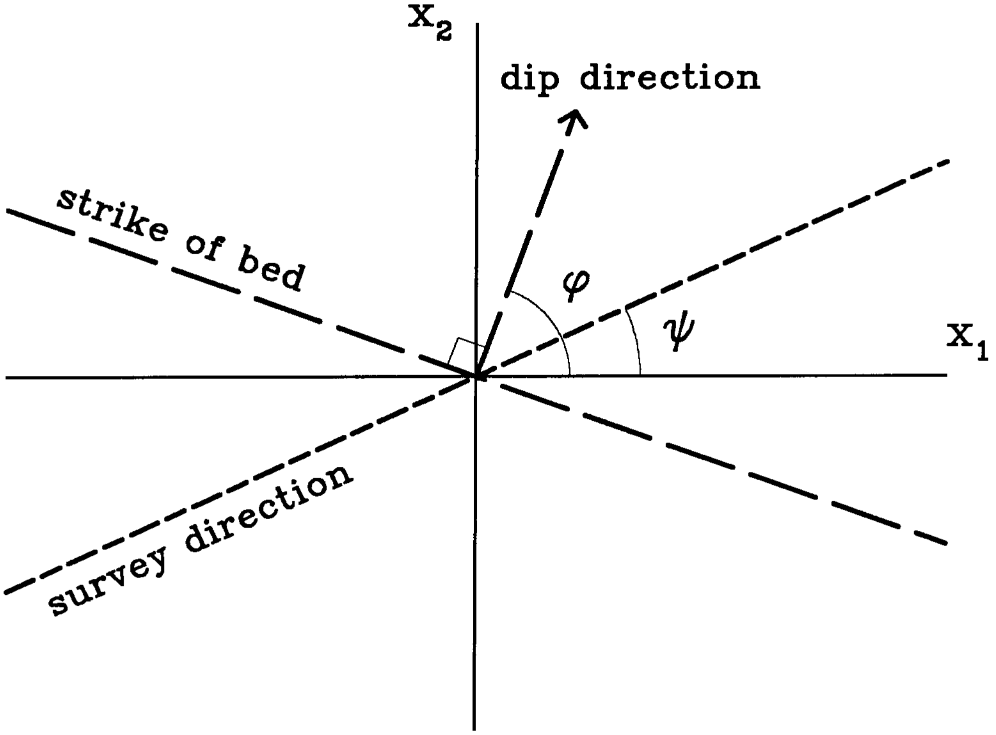


FIG. 2.34. Geometry of 3-D dip-moveout.

So the DMO equation (2.28) becomes

$$v_{NMO} = v \left[1 - \frac{v^2({k_{y_1}}^2 + {k_{y_2}}^2)}{4\omega^2} \cos^2(\phi - \psi) \right]^{-1/2} . \tag{2.33}$$

As a special case of note, if the survey parallels the x axis (ψ=0), equation (2.33) reduces to

$$v_{NMO} = v \left[1 - \frac{v^2 {k_{y_1}}^2}{4\omega^2} \right]^{-1/2} \tag{2.34}$$

which is just equation (2.4) again. If each azimuthal component of the data is Fourier transformed independently, the y_1 axis may be chosen in each case to be the azimuthal direction, so this last equation may be used generally. Note that equation (2.34) contains no contribution from the cross-line direction. DMO in three dimensions thus uses the same operator as in two dimensions, oriented along the appropriate offset axis, a point noted previously for other DMO implementations (Bolondi et al., 1982.; Hale, 1983; Jakubowicz et al., 1984). One practical consequence of this is that no Fourier transforms are required in the cross-line direction, so each line in a given shot-receiver azimuth direction may be processed for DMO independently.

This three-dimensional CVS algorithm requires that the data is binned into different azimuths, and that coverage in midpoint for each azimuth bin is dense enough to allow interpolation onto an even grid before Fourier transformation. There is no restriction on offset range, but velocity discrimination degrades for low fold coverage. The three-dimensional CVPM algorithm requires dense, evenly spaced sampling along both axes for both midpoint and offset (or shot and geophone) coordinates. In general requiring such densely sampled data is not realistic for current three-dimensional acquisition. However, many three-dimensional data, especially marine data, are collected as parallel two-dimensional lines. For such data, the azimuth range is limited, and the three-dimensional velocity space algorithms are practical. Without cable feathering there is only one azimuth present for parallel-line shooting, so one can simply use the two-dimensional DMO algorithm on each line, followed by three-dimensional migration. For small azimuthal deviation ψ, equation (2.33) may be approximated as

$$v_{NMO} = v \left[1 - \frac{v^2 k_{y_1}^2}{4\omega^2} \left(1 + 2\psi \tan\phi\right) \right]^{-1/2} . \tag{2.35}$$

Then if cable is feathering is present, or other azimuth binning is required, equation (2.35) can be used to estimate the effects of the feathering.

Migration may be incorporated in the three-dimensional CVS algorithm just as it is for two dimensions, except that equation (2.6) becomes

$$k_\tau = \omega \left(1 - \frac{v^2 (k_{y_1}^2 + k_{y_2}^2)}{4\omega^2} \right)^{1/2} . \tag{2.36}$$

This operator, unlike the DMO operator, requires cross-line information. However, for many practical applications, the three-dimensional migration can be split into two orthogonal passes of two-dimensional migration (Gibson et al., 1983). Thus it should be feasible to include the two-dimensional migration step in the DMO algorithm, followed by a cross-line two-dimensional poststack migration.

2.8 CONCLUSIONS

DMO and migration can be implemented practically in a space of constant-velocity stacks. DMO corrects the undesirable dip-filtering usually associated with NMO stacking. Including migration after DMO converts the stacks into prestack time-migrated sections. At each stage, an image may be extracted by interpolation between the stacks. Both DMO and migration substantially improve the quality of the resulting image.

The greatest advantage of the velocity-space imaging method is the potential improvement in quality of velocity analysis. Because the data is imaged before choosing velocities, the distorting effects geologic structure can have on velocity analysis are largely removed. Also, the commitment to a velocity function is postponed and may easily be changed after examining the stacked, DMO-corrected, or migrated section corresponding to any choice of velocities. The tremendous increase in velocity information available when using the velocity-space algorithms requires high-speed interactive graphics devices if one is to display and search the entire data space. Automatic optimization algorithms can also be used to search the data to improve velocity analyses.

These methods work best when lateral velocity variation is not large. When velocities do vary significantly laterally, coherent images are often still produced, but depth migration may be required to position all events correctly. For depth migration, interval velocities are required, so the next challenge is to understand how the patterns of stacking or migration velocities observed are related to the underlying interval velocities when these velocities vary laterally. This is the topic of chapters 3 and 4.